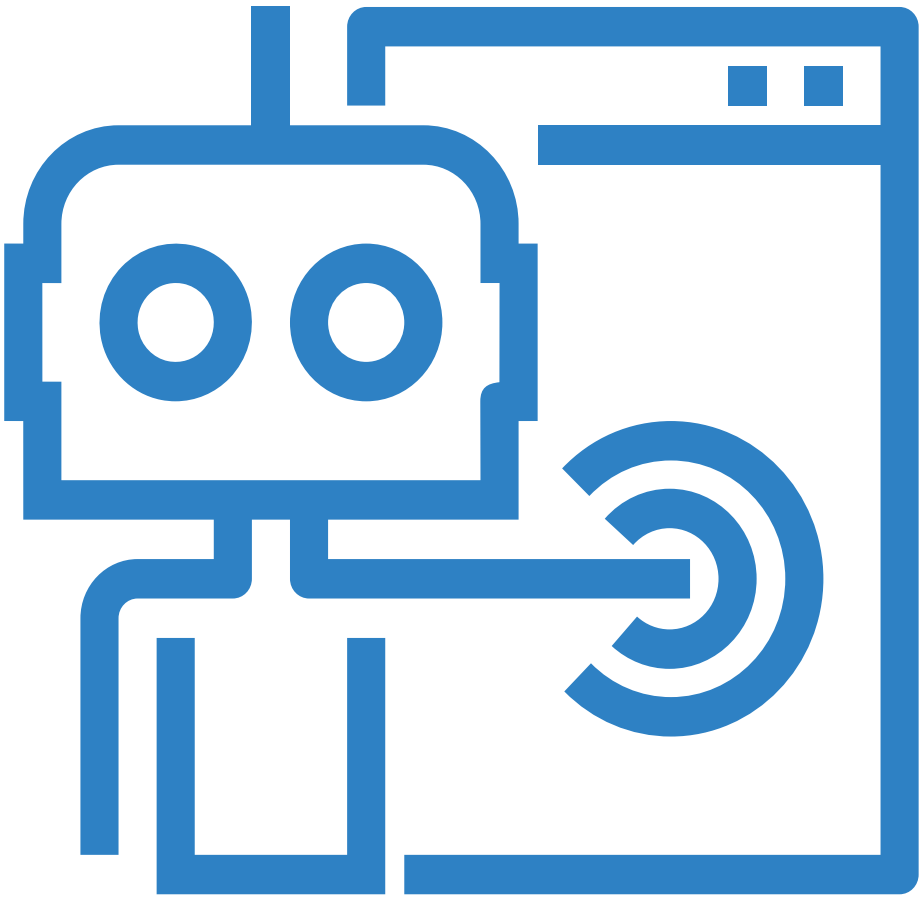
#MİLLİ TEKNOLOJİ HAMLESİ

DENEYAP TÜRKİYE

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ

ORTAOKUL



TÜBİTAK

TÜRKİYE TEKNOLOJİ TAKIMI

TÜBİTAK Yayınları

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ ORTAOKUL

Prof. Dr. İbrahim ÇETİN
Doç. Dr. Polat ŞENDURUR
Prof. Dr. Ercan TOP

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2021

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel malzemeler,
izin alınmadan tümüyle veya
kısmen yayımlanamaz.

ISBN 978-605-312-440-5
Yayıncı Sertifika No: 47703

Genel Yayın Yönetmeni: Fatma BAŞAR
Editör: İpek PİRPİROĞLU GENCER
Yardımcı Editör: Kübra BAL ÇETİNKAYA
Düzelti: Mustafa ORHAN
Telif İşleri Sorumlusu: Öznur KILIÇKAYA

TÜBİTAK
Bilim ve Toplum Başkanlığı
Popüler Bilim Kitapları Genel Yayın Yönetmenliği
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 95 21
e-posta: kitap@tubitak.gov.tr
esatis.tubitak.gov.tr

DENEYAP

Teknoloji Atölyeleri

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ

ORTAOKUL

Prof. Dr. İbrahim ÇETİN
Doç. Dr. Polat ŞENDURUR
Prof. Dr. Ercan TOP

TÜBİTAK

İçindekiler

ÖĞRETİM KILAVUZU **1**

GİRİŞ 1
GÖZLE UYGULA TASARLA ÜRET DEĞERLENDİR ÖĞRENME DÖNGÜSÜ 3
ETKİNLİK TABANLI PROGRAM VE SPİRAL ÖĞRENME 4
EŞLİ PROGRAMLAMA 5
GRUPLAR ARASI İLETİŞİM 6
DÖNEM PROJESİ 6
DERSTE KULLANILAN SET 6
LABORATUVAR KULLANIM YÖNERGESİ 7
SIKÇA SORULAN PEDAGOJİK SORULAR 7

1. BÖLÜM: ELEKTRONİK PROGRAMLAMAYA GİRİŞ **10**

1. GÖZLE VE UYGULA 11
1.1 Gözle- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç 11
1.1. Uygula- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç 11
1.2 Gözle ve Uygula- Basit Bir Devre Oluşturma 12
1.3 Gözle ve Uygula- Breadbord ve ArduinoUno ile Basit Bir Devre Oluşturma 13
1.3 Uygula- Basit Bir Devre Oluşturma ve Direnç Değerlerini Değiştirme 16
1.4 Gözle – Uygula ArduinoBlocks Geliştirme Ortamına Giriş 18
1.5 Gözle- Uygula Arduino ile Led Yakıp Söndürme 21
1.6 Uygula- Flip Flop 24
1.7 Uygula- Trafik Işığı 25
1.8 Uygula- Araba Yarışı Başlama Işıkları 26
2 TASARLA VE ÜRET 27
2.1 Tasarla- Kavşaktaki Trafik Lambaları 27
2.1 Üret- Kavşaktaki Trafik Lambaları 29
3- DEĞERLENDİR 31
4. İLAVE ETKİNLİK 32
4.1 Soldan-Sağa ve Sağdan-Sola Hareket Eden Led Uygulaması 32
4.2 Müzik ile Uyumlu Hareket Eden Led Uygulaması 33
4.3 Örüntü Tahmini Yarışması 34

2. BÖLÜM: TEMEL DEVRE KURULUMU, ARDUİNO UNO VE PROGRAMLAMAYA GİRİŞ **36**

1 GÖZLE VE UYGULA 38
1.1 Gözle- Arduino Uno'yu Tanıyorum 38
1.2 Gözle ve Uygula- Buzzer ile Melodiler 39
1.2 Uygula- Şarkı Çalan Buzzer 45
1.3 Gözle ve Uygula- Led Küpü Yapıyorum 45
1.4 Gözle ve Uygula- Led Küpünü Kodluyorum 47
1.5 Gözle- Led Küpünü Farklı Aralıklarla Yakıp Söndürüyorum 50
1.5 Uygula- Led Küpünü Farklı Aralıklarla Yakıp Söndürüyorum 51
1.5 Uygula- Katlar Yanıp Sönüyor 51

2. TASARLA VE ÜRET 54
2.1 Tasarla ve Üret- Şarkıyla Uyumlu Led Küpü 54
3. DEĞERLENDİR 55
4. İLAVE ETKİNLİK 55
4.1 Buzzerla Çalınan Şarkıyla Uyumlu Led Küpü 55
4.2 Led Küpü Animasyonu (Yarışma) 55

3. BÖLÜM-İLETKENLER, YALITKANLAR, YARI İLETKENLER VE ARDUİNO GİRİŞ PİNLERİ 57

1. GÖZLE VE UYGULA 59
1.1 Gözle-Yarı İletkenler 59
1.1 Uygula- Günlük Yaşamda Yarı İletkenler 59
1.2 Gözle ve Uygula- Diyot 59
1.3 Gözle- Elektronikte Kapılar 62
1.3 Uygula- Diyot ile VE Kapısı Yapıyorum 63
1.4 Gözle ve Uygula- Arduino'da Dijital Girdi- Push Buton ile Led yakma 64
1.5 Gözle ve Uygula- Değişkenler 68
1.6 Gözle ve Uygula- Kaç Buzzer'dan Ses Çıkıyor? 71
1.6 Uygula- Bir Buton Dört Hareketli Led 73
1.7 Gözle- RGB Led ile Farklı Renkler 74
1.8 Uygula- Ortak Ucu Belirleme 77
1.8 Uygula- Mor ve Turkuaz renkler 77
1.9 Gözle- ArduinoBlocks ile Ara Renkler 78
1.9 Uygula- Gökkuşağındaki 7 Temel Renk 78
2. TASARLA VE ÜRET 79
2.1 Tasarla- Diyot ile "VEYA" kapısı 79
2.1 Üret- Diyot ile VEYA Kapısı 80
2.2 Tasarla ve Üret- Yeniden Kaç Buzzer'dan Ses Çıkıyor? 80
3. DEĞERLENDİR 81
4. İLAVE ETKİNLİK 82
4.1 Notalara Göre Farklı Renk Veren Piyano 82
4.2 İki Butonla Bir Led'i Kontrol Etme 82
4.3 Butona Basınca Yanan Bir Daha Butona Basınca Sönen Led 83

4. BÖLÜM-DİJİTAL VE ANALOG GİRDİLER, POTANSİYOMETRE, TRANSİSTÖR VE KONDANSATÖRLER 84

1. GÖZLE VE UYGULA 86
1.1 Gözle- Potansiyometre ile Analog Girdi 86
1.1 Uygula- Potansiyometre ile Ses Ayarı 87
1.2 Gözle- LDR 89
1.2 Uygula- LDR ile Karanlıkta Yanan Led 90
1.3 Gözle- PWM ile Çıktı Üretme 91
1.3 Uygula- PWM ile Çıktı Üretme 92
1.3 Uygula- Yavaşça Yanıp Yavaşça Sönen Led 93
1.4 Gözle ve Uygula- Potansiyometre ile Led'in Parlaklığını Ayarlama 93
1.4 Uygula- LDR ile Işık Seviyesi Ayarlama 95
1.5 Gözle- Transistör 96
1.5 Uygula- Transistör ile Karanlıkta Yanan LED 98
1.6 Gözle ve Uygula- Transistör ile DC Motor Sürme 99
1.7 Gözle- Kondansatör ve Akım İlişkisi 101

1.7 Gözle ve Uygula- Transistör ve Kondansatör Kullanıyorum 103
1.7 Uygula- Led'in Yanma Süresini Değiştiriyorum 103
1.7 Uygula- Butona Basıldığında Sönüp Bir Süre Sonra Yanan Led 104
2. TASARLA VE ÜRET 105
2.1 Tasarla ve Üret- Sembolik El Kurutma Makinesi 105
2.1 Üret- Sembolik El Kurutma Makinesi 106
3. DEĞERLENDİR 109
4. İLAVE ETKİNLİK 110
4.1 Transistör ile Değil Kapısı Yapıyorum 110
4.2 DC Motorun Hızını Ayarlıyorum 111

5. BÖLÜM-SERİ PORTLAR, MULTİMETRE KULLANIMI, STEP MOTOR VE JOYSTİCK KULLANIMI 112

1. GÖZLE VE UYGULA 114
1.1 Gözle ve Uygula- Seri Port Kullanımı 114
1.2. Gözle ve Uygula- Multimetre ile Direnç, Gerilim ve Akım Ölçüyorum 117
1.3 Gözle- Rasgele Sayılar 119
1.3 Uygula- Butona Basıldığına Rasgele Yanan Led 119
1.4 Gözle ve Uygula- Step Motor ile Tahmin Oyunu 122
1.5 Gözle ve Uygula- Joystick ve Seri Port Kullanımı 127
1.6 Gözle ve Uygula- Joystick ile Servo Motor Kontrolü 130
1.6 Gözle ve Uygula- Robot Kol Yapımı 132
2. TASARLA VE ÜRET 136
2.1 Tasarla- Joystick ile Labirent Kontrolü 136
2.1 Üret- Joystick ile Labirent Kontrolü 137
3. DEĞERLENDİR 138
4. İLAVE ETKİNLİK 139
4.1 Robot Kol Yazı Yazıyor 139

6. BÖLÜM-DC MOTORLAR, KIZIL ÖTESİ VE MESAFE SENSÖRLERİ 140

1. GÖZLE VE UYGULA 142
1.1 Gözle ve Uygula- L298N'yi Kullanarak DC Motor Sürüyorum 142
1.1 Uygula- Motorları Ters Yönde Döndürüyorum 145
1.1 Uygula- Motorları Potansiyometre ile Farklı Hızlarda Döndürüyorum 145
1.2 Gözle ve Uygula- 2 Motorlu Araba Yapıyorum 147
1.3 Gözle ve Uygula- Mesafe Sensörünü Kullanıyorum 148
1.3 Uygula- Mesafe Sensörüne Bağlı Olarak Buzzer ve Led Kullanımı 149
1.3 Uygula- Ultrasonik Sayaç 150
1.4 Gözle- Ultrasonik Sensörü Arabaya Ekliyorum 150
1.4 Uygula- Yavaşlarken Bip Sesi Çıkaran Araba 151
1.5 Gözle ve Uygula- Engel Algılayan Kızılötesi Sensor 151
1.5 Uygula- Kızılötesi Sayaç 152
1.6 Gözle- Kızılötesi Sensör ile Siyah-Beyaz Ayrımı 153
1.7 Gözle ve Uygula- Kızılötesi Sensörleri Birlikte Kullanıyorum 154
1.8 Gözle ve Uygula- Eliptik Çizgi İzleyen Robot 156
1.9 Gözle- LCD ile Ekrana Yazı Yazma 158
1.9 Uygula- LCD'de Yazı Animasyonu 159
1.10 Gözle- LCD'de Sembol Kullanımı 160
1.10 Uygula- LCD'yi Arabaya Bağlıyorum 161
2. TASARLA VE ÜRET 161

2.1 Tasarla- Sembollerle Animasyon Yapıyorum 161
2.1 Üret- Sembollerle Animasyon Yapıyorum 162
3- DEĞERLENDİR 163
4. İLAVE ETKİNLİK 164
4.1 Otopark Sayaç 164
4.2 Dairenin İçerisinden Çıkmayan Robot 165
4.3 Arabayı Park Ediyorum 165
4.4 Üçüncü Çizgiyi Geçince Duran Robot 166

7. BÖLÜM-AKILLI EV TASARLIYORUM 167

1. GÖZLE VE UYGULA 169
1.1 Gözle ve Uygula- Nem Sensörü ve Dijital Saat 169
1.2 Gözle ve Uygula- Zaman, Sıcaklık ve Nem Bilgisini LCD'ye Yazıyorum 173
1.3 Gözle ve Uygula- Güvenlik Alarmı 176
1.4 Gözle ve Uygula- Yangın ve Gaz Alarmı 181
2. TASARLA VE ÜRET 185
2.1 Tasarla- Hayalimdeki Akıllı Ev 185
3. DEĞERLENDİR 188

8. BÖLÜM-NESNELERİN İNTERNETİNE GİRİŞ 189

1.GÖZLE VE UYGULA 190
1.1 Gözle ve Uygula- Uzaktan Kumandalı Yem Makinesi 190
1.2 Gözle ve Uygula- Kaç Kişi Geldi? 195
1.3 Gözle ve Uygula- Ev Gözlem Monitörü 207
2. TASARLA VE ÜRET 214
2.1 Tasarla-Su doluluk göstergesi 214
2.1 Üret: Su doluluk göstergesi 214
3- DEĞERLENDİR 215
4. İLAVE ETKİNLİK 216
4.1 Üret: Orman Yangın Alarmı 216

9. BÖLÜM-BLUETOOTH İLE BONİBON SAYIYORUM 217

1. GÖZLE VE UYGULA 219
1. Gözle- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı 219
1.1 Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı: Kapak Hazırlığı ve Montajı 221
1.2 Gözle ve Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı: Devre Kurulumu 223
1.2 Uygula: Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı- Bluetooth Bağlantısı 225
1.2 Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı- Sayı Göstergesi 230
1.3 Gözle ve Uygula- Mıknatıs Detektörü 232
2. TASARLA VE ÜRET 235
2.1 Tasarla ve Üret- Hangi Renkten Kaç Tane 235
3. DEĞERLENDİR 238
4. İLAVE ETKİNLİK 238
4.1 Bonibon Sayılarını Dosyaya Yazdırıyorum 238

10. BÖLÜM-METİN TABANLI ORTAMDA ELEKTRONİK PROGRAMLAMA 240

1. GÖZLE VE UYGULA 242
1.1 Gözle ve Uygula- Arduino IDE'ye Geçiş Yapıyorum 242
1.1 Uygula- Arduino Ortamında Flip Flop 246
1.2 Gözle- Tam Sayı Değişkenleri ve Döngüler ile Led Animasyonu 247
1.2 Uygula- Soldan Sağa Akan 5 Led'li Animasyon 250

1.3 Gözle ve Uygula- Soldan-Sağa ve Sağdan-Sola Akan 5'li Led 250
1.4 Gözle ve Uygula- Koşul ile Animasyonları Birleştiriyorum 250
1.5 Gözle- Arduino Ortamında PWM Pini Kullanımı 252
1.6 Gözle- PWM ile RGB Kullanımı 253
1.6 Uygula- Maviden Kırmızıya, Kırmızıdan Yeşile ve Yeşilden Maviye 254
1.7 Gözle ve Uygula- Basit Ambulans Sireni: Buzzer ve Hoparlör Kullanımı 255
1.8 Gözle ve Uygula- Gelişmiş Ambulans Sireni 256
1.9 Gözle- LCD Ekrana Yazı Yazdırıyorum 257
1.10 Gözle ve Uygula- Yanıp Sönen Yazı 259
1.11 Gözle- String (Metin) Değişken Tanımlıyorum 260
2. TASARLA VE ÜRET 262
2.1 Tasarla ve Üret- Döngüler ile İleri Led Animasyonu 262
3. DEĞERLENDİR 263
4. İLAVE ETKİNLİK 265
4.1 Basit Ambulans Sireni ve Lambası 265
4.2 Sırayla Akan TUBITAK Yazısı 265
4.3 Gelişmiş Ambulans Sireni ve Lambası 267
4.4 Sırayla Akan Yazının Genellemesi 268

11. BÖLÜM- JOYSTİCK, SERİ PORT EKRANI VE BLUETOOTH VERİLERİ 270

1. GÖZLE VE UYGULA 272
1.1 Gözle- Joystick Verilerini Okuyup Seri Ekranda Yazdırma 272
1.1 Uygula- Joystick Verisini Yönlere Göre Yazdırıyorum 273
1.2 Gözle- Dizileri Kullanarak LCD'ye Özel karakter Yazdırıyorum 274
1.3 Gözle ve Uygula- Gülen Yüzü Joystick Kullanarak Sağa Doğru Hareket Ettiriyorum 277
1.3 Uygula- Gülen Yüzü Joystick Kullanarak Sağa-Sola Hareket Ettiriyorum 278
1.3 Uygula- Gülen Yüz Kenarlar Arası Geçiş 279
1.4 Gözle- Seri Port Ekranından Gelen Bilgiyi LCD'ye Yazma 281
1.5 Gözle- Seri Port Ekranından Gelen Bilgiyi Okuyup Bütün Olarak LCD'ye Yazma 282
1.6 Gözle ve Uygula- Bluetooth ile Led Kontrolü 283
1.6 Uygula- Bluetooth ile Led Kontrolü ve LCD 286
2. TASARLA VE ÜRET 287
2.1 Tasarla ve Üret- Gülen Yüz Oyunu 287
3. DEĞERLENDİR 290
4. İLAVE ETKİNLİK 291
4.1 ledAcKapa Programını Kullanarak Gülen Yüzü Hareket Ettiriyorum 291
4.2 Gülen Yüz Oyununu Geliştiriyorum 291
4.3 Modifiye lcdAcKapa Programı ile Gülen Yüz Oyunu 292

12. BÖLÜM-KENDİ ARABAMI KODLUYORUM 293

1. GÖZLE VE UYGULA 295
1.1 Gözle ve Uygula- MPU 6050 ile Breadboard Görselleştirme 295
1.1 Uygula- MPU 6050 Açı Değerlerini Seri Port Ekranına Yazdırıyorum 298
1.1 Uygula- MPU 6050 ile Led Yakma 299
1.2 Gözle- Hız Sensörü ve Kesme (Interrupt) ile Sayaç Yapıyorum 300
1.2 Uygula- DC Motora Bağlı Tekerleğin Döndüğü Tur Sayısını Bulan Program 303
1.3 Gözle- Motor Gücü ile Dönen Tekerlek için Hız Sensörünün Saydığı Bölme Sayısını Bulan Program 304
1.3 Uygula- Arabayı Hareket Ettirirken Sol ve Sağ Teker İçin Bölme Sayısını Hesaplıyorum 307
1.4 Gözle- Hız Sensörü Yardımı ile Arabayı İleri Hareket Ettiriyorum 308

1.4 Uygula- Hız Sensörü Yardımı ile Arabayı İleri ve Geri Hareket Ettiriyorum 310
1.5 Gözle- Hız Sensörü Yardımı ile Arabayı Daha Dengeli İleri Hareket Ettiriyorum 312
1.4 Uygula- Hız Sensörü Yardımı ile Arabayı Daha Dengeli İleri ve Geri Hareket Ettiriyorum 314
2.TASARLA VE ÜRET 316
2.1 Tasarla ve Üret- Hayalimdeki Arabayı Oluşturuyorum 316
3. DEĞERLENDİR 317
EK-1-LED KÜP KURULUM YÖNERGESİ (BÖLÜM-2) 318
EK-2-ROBOT KOL MONTAJ KILAVUZU (BÖLÜM-5) 323
EK-3-ARABA KURULUM KILAVUZU RESİMLERİ 339
(BÖLÜM-6) 339
EK-4-BONİBONSEÇ KAYNAK KODU (BÖLÜM-9) 348
EK-5-MANYETİKKAYİT KAYNAK KODU (BÖLÜM-9) 349
EK-6-HC05 İSİM DEĞİŞİKLİĞİ (BÖLÜM-9) 351
EK-7-HC05 KONTROL KAYNAK KODU (BÖLÜM-12) 353
EK-8-LED AÇ-KAPA KAYNAK KODU (BÖLÜM-12) 354
EK-9-PROCESSİNGVİSUALİZATİONMPU6050.PDE KAYNAK KODU (BÖLÜM-12) 356
EK-10-PROJE HAZIRLIYORUM 358
1. PROJEM (GİRİŞ) 358
2 PROJEM (EMPATİ) 359
3. PROJEM (TANIMLAMA) 359
4. PROJEM (FİKİR ÜRETME) 360
5. PROJEM (GELİŞTİRME VE TEST ETME) 361
6. PROJEM (SUNUM) 361

Sunuş

Geleceğin bilim insanlarını, araştırmacılarını, teknoloji girişimcilerini, bilim ve teknoloji üretmeye meraklı gençleri yetiştirmek ülkemizi geleceğe taşımada en önemli görevlerimizden biridir. Teknolojik açıdan kendine yeterliliğinin en temel unsuru yetişmiş insan kapasitesinin geliştirilmesidir. Ülkemizin kalkınması için teknoloji üretme yetkinliği yüksek genç bireyler yetiştirmek, gençlere yaratıcılıklarını ortaya koyabilmeleri ve üretme kapasitelerinin farkına varmalarını sağlayacak ortamlar sunmak amacıyla Deneyap Teknoloji Atölyeleri kurulmaktadır.

Deneyap Teknoloji Atölyelerinde 36 ay süre ile ücretsiz olarak verilen proje geliştirme odaklı deneysel ve uygulamalı eğitimler ile bilim ve teknoloji alanında farkındalık yaratan, Ar-Ge becerileri gelişmiş, güncel teknolojilerle erken dönemde karşılaşmış, takım çalışması tecrübesi kazanmış, girişimci, yenilikçi gençler yetiştirilmesi hedeflenmektedir. Atölyelerde Tasarım ve Üretim, Robotik ve Kodlama, Elektronik Programlama ve Nesnelerin İnterneti, Yazılım Teknolojileri, İleri Robotik, Nanoteknoloji ve Malzeme Bilimi, Siber Güvenlik, Yapay Zekâ, Mobil Uygulama, Enerji Teknolojileri, Havacılık ve Uzay Teknolojileri derslerine ilişkin eğitimler verilmektedir. Verilen eğitimlerden biri de “Elektronik Programlama ve Nesnelerin İnterneti”dir.

Bu kitap ortaokul seviyesinde elektronik programlama ve nesnelerin interneti dersini vermek isteyen öğretmenler için hazırlanmıştır. Bu eserde elektronik programlama ve nesnelerin İnternetine yönelik içeriğin nasıl verilmesi gerektiği bir model doğrultusunda tasarlanmıştır. Bu iş için GUTÜD (Gözle Uygula Tasarla Üret Değerlendir) öğrenme döngüsü temel alınmıştır. İçeriğin tamamı bu modele göre kurgulanmıştır. GUTÜD öğrenme döngüsü (Çetin, Üçgül, Top ve Yükseltürk, 2021; Üçgül, Çetin, Yükseltürk ve Top, 2021) ilk defa kodlama (bilgisayar bilimi) eğitimi bağlamında geliştirilmiş bir modeldir. Kodlama eğitiminde içerik ve teknoloji bilgisi kadar pedagoji bilgisi de önemlidir. Öğretmenler sınıfa girdiklerinde ne yapmalıdırlar; ders için verilecek içerik bilgisi her şeyden önemli midir; dersi anlatırken en beğendikleri öğretmen gibi mi anlatmalıdırlar; anlattıkları derste amaç ne olmalıdır gibi cevaplanması gereken sorular bulunmaktadır. Günümüzde kodlama eğitimine yönelik birçok derste, genel olarak, öğrencilere bir içerik gösterilir ve öğrencilerin bu içeriği benzer koşullarda yinelemesi beklenir. Bu tür bir yaklaşım uygun bir öğretim yaklaşımı olmayacaktır. Bu ve benzeri öğretim ortamlarında öğrenciler düşük seviye bilişsel becerilerini kullanırlar ve öğrendikleri bilgileri ihtiyaçları olduğunda gerçek yaşam bağlamında kullanmakta sıkıntı çekerler. GUTÜD öğrenme döngüsü öğretmenlerin sınıfa girdiğinde konuyu dersin başından sonuna kadar nasıl işlemesi gerektiğini tarif eder. Bu döngünün öğrencilerin bilgi işlemsel düşünme, problem çözme ve takım olarak çalışma gibi ileri seviye becerilerini geliştirmelerine yardımcı olacağı düşünülmektedir. GUTÜD döngüsü ile öğrenciler öğretmenlerinin gösterdiği içeriği benzer ortamlarda tekrarlayan bireyler olmaktan öteye geçip ekip olarak çalışan, düşünen, tasarlayan, üreten, problem çözen ve değerlendiren bireyler olarak görülmektedir.

Kitap 14 hafta süren bir ders düşünülerek oluşturulmuştur. Her haftaya bir bölüm düşecek şekilde on iki haftalık içerik hazırlanmıştır. Öğrenciler dokuzuncu haftadan itibaren proje konuları ile ilgili yönlendirilebilir. On üçüncü ve on dördüncü haftalar ise tamamen proje

oluşturmaya ayrılabilir. Bu sırayı izlemek ve kitabın tamamını kullanmak istemeyen öğretmenler kitap içerisindeki ilgili bölümleri kendi dersleri için kullanabilirler. Kitapta temel elektronik devre elemanları, Arduino mikro denetleyicisi, ArduinoBlocks isimli programlama ortamı, sensörler, robotlar ve nesnelerin interneti konularına değinilmiştir. Bu konuların ötesinde, kitapta öğrencilerden bu temel bilgileri kullanarak karmaşık problemlere çözüm getirmeleri beklenmektedir.

Bu kitapta her bölüm kendi içerisinde bir bütünlük oluşturacak şekilde tasarlanmıştır. Yeni konular eski konuları kapsayacak şekilde oluşturulmuştur. Kitapta öğretmenlerin temel elektronik programlama bilgilerinin olduğu var sayılmıştır. Fakat yine de kitapta konular temelden ileri seviyeye gidecek şekilde anlatılmıştır. Bu alana yeni girmiş öğretmenler de kitapta anlatılan konulardan yararlanıp kendilerini geliştirebilirler. Bunun yanında temel programlama bilgisine sahip ileri seviye ortaokul öğrencileri ve lise öğrencileri de bu eserden faydalanabilirler. Bu durumda bir öğretmenden yardım alınmalı veya öğrenci yeterli olgunluğa erişene kadar kitap kullanılmamalıdır.

Kitap içerisinde öğretmenler için ilk bakışta kolay gibi görünen ama aslında zor olan birçok içerik bulunmaktadır. Öğretmenler ders öncesinde bu içerikler üzerinde hazırlık yapmazsa ders esnasında sıkıntı yaşayabilirler. Bunun yanında GUTÜD modelinin anlaşılıp uygulanması da ön çalışma gerektirebilir. Öğretmenlerin işleyecekleri konuyu bir hafta öncesinden incelemeleri önerilmektedir.

Sevgili öğretmenler kitabın size ve ülkemiz öğrencilerine faydalı olması dileğiyle!

Öğretim Kılavuzu

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ DERSİ ÖĞRETİM KILAVUZU

Dünya genelinde bilgisayar bilimi eğitimi ve enformatik eğitimi isimleriyle, Türkiye'de sıklıkla kodlama eğitimi ismiyle bilgi işlemin oluşturduğu gücün eğitimde farklı alanlarda kullanıldığı bir döneme tanıklık etmekteyiz. Aynı zamanda bilgi işlemin gücünün kendi mecrasında mühendisliğin, bilimin ve matematiğin çeşitli alanlarında robotik, sayısal modelleme, sosyal medya, yapay zekâ, insan genom projesi ve CERN araştırmaları gibi başlıklar altında kullanıldığı verimli bir dönemdeyiz. Fakat bu gelişmeler ile bilgisayar bilimi eğitiminde önemli bir problem ortaya çıkmıştır: Bilgisayar bilimi eğitimi nasıl verilmelidir? Bu problemin üzerinde yeteri kadar durulmadığını düşünmekteyiz. Görsel programlama, elektronik programlama, robotik kodlama, web programlama, metin tabanlı programlama, mobil programlama ve bunlar gibi isimler kullanılarak çeşitli dersler ve kursların açıldığını, kitapların yazıldığını gözlemekteyiz. İçerik üzerine yoğunlaşmak bir dereceye kadar önemlidir fakat bunun yanında içeriğin pedagojisine ve öğretimine de odaklanmak gereklidir. Bu bileşen eksik kaldığında istenilen verim alınamayabilir. Öğretmenler sınıflarına girdiğinde bahsi geçen konuları anlatmaktadırlar. Acaba öğretmenler sınıfın kapısını kapattıklarında nasıl bir öğretim yaklaşımı sergilemektedir ve hangi düşünme becerilerini hedeflemektedir? Sadece içeriğin verilmesi yeterli değildir. Aynı zamanda öğretimin tasarlanması da esas olmalıdır. Öğretim sistematik ve bütüncül bir etkinliktir. Tüm bunlara dayanarak bu kitapta deneyimsel öğrenmeye dayalı bir öğretim tasarımı oluşturulmuş ve kullanılmıştır. Bu tasarım GUTÜD öğrenme döngüsü (Çetin, Üçgül, Top ve Yükseltürk, 2021; Üçgül, Çetin, Yükseltürk ve Top, 2021) etkinlik tabanlı program, spiral öğrenme ve eşli programlamayı içermektedir.

GİRİŞ

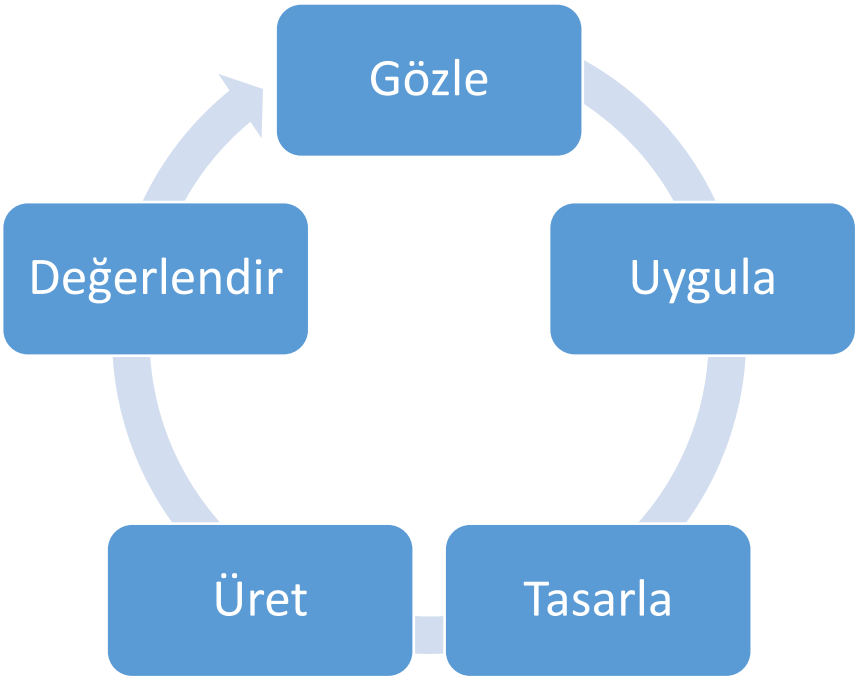
İçinde bulunduğumuz çağda öğrenme, öğrenenlerin sınıf içinde verilen kavram, prosedür ve temel bilgileri istendiği zaman belirli bir ölçüde sınav veya başka vasıtalar ile yeniden göstermesi olarak algılanmamaktadır. Öğrenenlerin öğrendiği kavram, prosedür ve temel bilgileri başka bağlamlara transfer etmesi ve bu bilgileri kendi amaçları doğrultusunda yeniden inşa etmesi gerekmektedir. Bu amaç doğrultusunda öğrenenler sadece belirli kavramları öğrenmekle yetinmemelidir. Bu kavramları öğrenirken problem çözme, soyutlama ve kritik düşünme gibi üst düzey düşünme becerilerini de geliştirmelidir.

Elektronik Programlama ve Nesnelerin İnterneti dersinde öğrencilerin temel elektronik, elektronik programlama ve nesnelerin interneti konularını kavrayıp bu kavramları farklı bağlamlarda kullanıp ileri seviye problem çözme etkinlikleri ile uğraşmaları gerekmektedir. Bu konu başlıkları ile sadece belirli bir mikrodenetleyicinin özellikleri, belirli devre elemanları ve sensörlerin kullanımı ve bir kodlama ortamının kullanılması yeterli değildir. Bunun ötesine geçilerek öğrencilerin bilgi işlemsel düşünme becerilerinin geliştirilmesi hedeflenmelidir. Cuny, Snider ve Wing (Cuny, Snyder ve Wing (2010)'dan akt., Wing, 2011, s.20) bilgi işlemsel düşünmeyi "Çözümlerin bir bilgi işleme birimi tarafından etkili şekilde yerine getirilebilecek formda sunulması amacıyla problemleri ve çözümleri formülleştirmeyi içeren düşünme süreci" olarak tanımlamışlardır. Bilgi işlemsel düşünme bir problem çözme sürecidir

ve beş bileşeni içerir: (i) soyutlama, (ii) algoritmik düşünme, (iii) problemi bileşenlerine ayırma, (iv) çözümü verim açısından analiz etme ve (v) bir çözümü farklı bağlamlar için genelleme.

Öğrencilerin elektronik programlama ve nesnelerin interneti konularını öğrenmeleri ve bilgi işlemsel düşünme becerilerini geliştirmeleri için farklı yaklaşımlar kullanılabilir. Bu amaç doğrultusunda Elektronik Programlama ve Nesnelerin İnterneti dersinde iki temel yaklaşım harmanlanarak kullanılacaktır. Bu yaklaşımlardan ilki, dersi rehber öğretmenin anlatımı üzerine kurar. Rehber öğretmen konuyu doğrudan anlatabilir, model olarak problemleri nasıl çözdüğünü gösterebilir, öğrencilerden gösterilen konu ve kavramları tekrarlamalarını isteyebilir. Bu yaklaşımda öğretmen asıl sorumluluğu alır ve öğrencilere konuyu aktarmayı hedefler. İkinci yaklaşım ise dersi öğrencinin keşfedeceği şekilde tasarlayıp öğrenci merkezli bir öğretim hedeflemektedir. Buna göre öğrenci süreçte aktif rol üstlenir. Öğretmen bilgiyi aktarmaktan ziyade, öğrencinin keşfetme veya oluşturma sürecinde ona yardım/rehberlik eden pozisyondadır.

Elektronik Programlama ve Nesnelerin İnterneti dersinde bahsi geçen iki yaklaşım harmanlanarak GUTÜD (Gözle, Uygula, Tasarla, Üret, Değerlendir) öğretim/öğrenme döngüsü oluşturulmuştur. Döngü adımları öğrenci penceresinden isimlendirilmiştir. Gözleyen, uygulayan, tasarlayan, üreten ve değerlendiren öğrencidir. GUTÜD öğrenme döngüsündeki Gözle ve Uygula adımları derste temel kavramların öğrenilmesi için kullanılacak ve aktarım yaklaşımı ön plana çıkaracaktır. Tasarla ve Üret adımlarında ise öğrencilerin verilen temel kavramları kullanarak ileri seviye zihinsel becerileri geliştirmeleri hedeflenmektedir. Bu aşamada keşfe dayalı yaklaşım ön plana çıkarılacaktır. Döngünün son adımı olan Değerlendir adımında öğrencilerden öğrenilen konuları pekiştirmesi, bu konuları basitçe günlük yaşam bağlamında düşünebilmesi ve öğrenme sürecinde yaşadığı deneyimler üzerine düşünerek öğrenme sürecini tamamlamaları beklenmektedir.



Şekil 1. GUTÜD Öğrenme/Öğretim Döngüsü

GÖZLE UYGULA TASARLA ÜRET DEĞERLENDİR ÖĞRENME DÖNGÜSÜ

Gözle: Bu bölüm iki kısımdan oluşur. Birinci kısım ders veya konu başlarında gerçekleştirilmelidir. Rehber öğretmen bu aşamada öğrencilerin geçmiş bilgilerini aktive etmek ve onların dikkat ve motivasyonlarını sağlamak ile görevlidir. Öğretmen öğrencilerin geçmiş bilgilerini aktive etmek ve onların dikkatini çekmek için bir önceki derste yapılan etkinlikleri / çalışmaları kısaca özetleyebileceği gibi günlük yaşamdan ilgi çekici örnekler de kullanabilir, örneğin otomasyona giriş yapmak için El-Cezerî'nin hayatından ve ürettiği cihazlardan bahsedilebilir.

Bu bölümün ikinci kısmı her Gözle etkinliği için geçerlidir. İkinci kısımda rehber öğretmen bir elektronik konusunu uygulamalı olarak (göstererek) anlatır. Gözle etkinliklerinde genel olarak temel hedef doğrudan konunun anlatılması değildir. Burada doğrudan anlatım yapmaktan ziyade rehber öğretmen bir görev üzerinden konuyu anlatır. Etkinlikte, uzmanın eldeki bir problemi nasıl çözeceği ve/veya o probleme nasıl yaklaşacağı gösterilir. Öğretmen bir problemin çözümünü bir uzmanın nasıl yapacağını gösterirken konuyu anlatmış olacaktır. Rehber öğretmen, çok sık olmamakla birlikte, bazen doğrudan anlatım yöntemini de kullanabilir. Gözle etkinliklerinde öğretmen daha aktiftir. Uygulamayı yaparken öğrencilere sorular sorabilir ve öğrencilerin sorularını yanıtlayabilir.

Gözle etkinliklerinde genel olarak öğrenciler gösterilen konu ile ilk defa karşılaşmış olacaktır. Bu yüzden buradaki etkinlikler diğer adımlarda bulunan etkinliklere göre daha basit olacaktır. Bu aşamada konu öğrenciler için dışsaldır. Öğrenciler konuyu içselleştirmenin ilk adımını atmaktadırlar. İçselleştirme sürecinde öğrencilerin zengin bir öğrenme deneyimi elde etmeleri ve bu deneyimleri üzerine düşünmeleri önemlidir. Öğrenciler bu aşamada bir görevi yerine getirirken onun her bir adımını açık bir şekilde yapmaya meyilli olacaktır. Görevin nasıl yerine getirileceğini hayal edemezler. Bu süreçte dışsal ipuçları önemli bir rol oynayabilir. Bu ipuçları fiziksel veya zihinsel olabilir. Bu duruma örnek olarak çocukların toplama işlemini öğrenirken parmaklarını kullanması verilebilir. Rehber öğretmen uygun olduğu durumlarda dışsal ipuçları vermelidir.

Uygula: Bu bölümde öğretmen öğrencilerden Gözle adımında gösterilen uygulamaların bir benzerini veya bazen aynısını, yapmalarını ister veya bazen birlikte yapar. Örneğin, bir elektronik devredeki ledlerin sırayla yakılması için gerekli elektronik devreyi oluşturmayı ve program kodlarını yazmayı gösteren rehber öğretmen, öğrencilerden bir trafik ışığını simule eden elektronik devre oluşturmayı ve program kodlarını yazmayı isteyebilir. Gözle ve uygula bölümü sonunda öğrenciler ilgili konular ve kavramlar hakkında bilgi sahibi ve çeşitli deneyimler kazanmış olurlar. Burada amaç öğrencilerin temel kavramlar ve konular hakkında deneyim kazanması, temel kavram ve konulara hâkim olması ve içselleştirmesidir.

Tasarla: Bu adımda öğrenciler aktif rol üstlenir. Öğretmen rehber pozisyonundadır. Öğretmen, öğrencilere takıldıkları noktalarda destek olacaktır. Fakat öğretmenin sağladığı destek gereğinden fazla olmamalıdır. Tasarla adımında öğretmen tarafından öğrencilere bir problem verilir. Öğrencilerden öncelikle bu problemin çözümünü tasarlamaları istenir. Tasarlama aşamasında öğrenciler temel itibariyle bilinenler ile istenenler arasındaki bağı kurarak bir çözüm planı üreteceklerdir.

Bu adımdaki önemli nokta, öğrencilerin çözüme doğrudan başlaması yerine önce çözüm hakkında düşünmesi ve bir çözüm planı üretmesidir. Öğrenciler her defasında tasarlama etkinliklerini yapmak istemeyebilir ve doğrudan çözüm üretmeye geçmek isteyebilir. Bu durumlarda, öğrencilerin adımları birebir uygulaması yönünde zorlamak yerine onlardan problemi doğrudan çözmeye başlamadan önce problem hakkında düşünmesi ve planlama yapması istenebilir.

Üret: Bu adımda da öğrenciler aktif rol üstlenir. Öğretmen yine rehber pozisyonundadır. Öğretmen öğrencilere problemlerin çözümünü vermez. Problemlerin öğrenciler tarafından çözülmesi için onlara

yardımcı olur. Üret adımında, öğrencilerden bir önceki adımda tasarladığı planı kullanarak probleme bir çözüm üretmesi istenir. Öğrenciler bilgisayar ve elektronik set başında çalışarak gerekli donanımsal ve algoritmik çözümleri geliştirirler.

"Tasarla ve Üret" adımları öğrencilerin "Gözle ve Uygula" bölümünde öğrendiği bilgi ve becerileri kullanarak problem çözme, soyutlama, algoritmik düşünme ve kritik düşünme gibi becerileri geliştirdikleri adımlardır. Bir başka ifade ile öğrencilerin öğrendikleri bilgilerin/becerilerin problem çözme sürecinde daha ileri seviyelerde kullanımını içermektedir. Bu yüzden "Tasarla ve Üret" aktivitelerinde öğrencilerin tasarımları ve çözümleri kendi başlarına yapmalarına izin verilmelidir. Öğrenciler üst seviye zihinsel becerilerini geliştirmeye çalıştığı durumlarda yardıma ihtiyaç duyabilirler. Burada yapılacak yardım öğrencilerin bahsi geçen becerileri farklı ortamlarda kendi başlarına bireysel olarak uygulayabilmelerini hedeflemelidir. Öğrenciler yardım kesildiğinde karmaşık görevler ile tek başlarına veya grup olarak baş etmeyi öğrenmelidir. Bu adımda öğrencilere problemin çözümünü doğrudan vermek onların tüm bu deneyimlerden eksik kalmasına sebep olabilir.

Değerlendir: GUTÜD öğretim döngüsünde değerlendirme ile anlatılmak istenen doğrudan öğrencinin başarısının puanlanması değildir. Buradaki temel hedefler, o hafta işlenen konunun pekiştirilmesi, öğrenilen bilgilerin günlük hayattaki kullanım alanlarının irdelenmesi ve öğrencinin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. İlk olarak, konunun pekiştirilmesi sürecinde içerikle ilgili görece klasik tabir edilebilecek sorularla öğrencilerin öğrendikleri bilgi ve beceriler hakkında düşünmeleri / tartışmaları planlanmaktadır. İkinci olarak öğrenilen içeriğin günlük hayatta kullanımı ile ilgili sorularla öğrencilerin farkındalığını ve öğrenilen bilginin aktarımını/transferini kolaylaştırmak hedeflenmektedir. Son olarak öğrenci öğrenme sürecini masaya yatırır. Öğrenme için belirli bir konu veya kavram ile ilgili deneyimlerin oluşturulması yeterli değildir. Elde edilen deneyimlerin üzerine düşünmek gerekmektedir. Değerlendirme adımında öğrenci öğrenme sürecinde yaşadığı deneyimler üzerine düşünür. Bu deneyim öğrenme sürecinin tamamlanması, öğrencinin kendini tanıması, öğrencinin gelecek öğrenmelerini daha verimli hale getirmesi için önemlidir.

ETKİNLİK TABANLI PROGRAM VE SPİRAL ÖĞRENME

Klasik programlar oluşturulurken konular birbirinden ayrılarak sıralanır ve oluşturulan sıraya göre öğretilir. Aslında bu mantıklı bir yaklaşımdır. Bu sayede konular basitten karmaşığa doğru ve organize bir şekilde öğrenciye sunulabilir. Her yaklaşımın olduğu gibi bu yaklaşımın da dezavantajları bulunur. Klasik yaklaşımda konular gerçek hayat bağlamından koparılmış olur. Öğrenilen konuların özümsenip bütünleştirilmesi çoğunlukla öğrencilere bırakılır. Bu kitapta etkinlik tabanlı program kullanılmıştır. Bu yaklaşımda program etkinlikler çerçevesinde oluşturulur. İlgili konular etkinlikler içerisinde verilir. Bu sayede öğrenciler yaptığı etkinlikte ayrık konulardan ziyade anlam bütünlüğü olan problemlerle uğraşabilir. Öğretmen bir uzman olarak bu problemlere nasıl yaklaştığını öğrencilere gösterebilir. Ayrıca öğrenciler ve öğretmen derslerinde kavramlara ayrık konular üzerinden gitmektense ürettikleri somut ürünler veya etkinlikler üzerinden ulaşmaya çalışır. Böylece tüm sınıfın paylaşabileceği, uygulayabileceği ve üzerinde düşünebileceği kolektif düşünme ürünleri/ortamları oluşur.

Hiçbir yaklaşım teknik veya yöntem bütün sorunları çözemez. Bir yaklaşımın temel alınması her koşulda onun uygulanması gerektiği anlamına gelmez. Bu bakış açısı ile oluşturulan program genel olarak etkinlik tabanlı olsa da bazen klasik yaklaşım da kullanılabilir. Örneğin bazı durumlarda etkinlik tabanlı programda bütünlüklü olarak ele alınan problemler öğrencilere çok karmaşık gelebilir. Burada klasik yöntemde olduğu gibi bu problem küçük parçalara ayrılarak parçaların öğretilmesi üzerinde durulabilir.

Nitekim çok sık olmamakla birlikte ve gerekli olduğu düşünüldüğünde klasik yöntem bu kitapta kullanılmıştır.

Etkinlik tabanlı programın sağladığı bir diğer avantaj konu sıralaması ile ilgilidir. GUTÜD döngüsü çerçevesinde verilen etkinlik tabanlı program sayesinde konular spiral şekilde verilmektedir. Buradaki spiralin üç bileşeni bulunmaktadır. Birinci bileşen basitten karmaşığa bir yönelimi belirtir. Gelişim genellikle basit, daha zor, daha zor … şeklinde ilerlemektedir. İkinci bileşen basitten karmaşığa doğru giden yapının ardışık olmadığını belirtir. Daha zor olan konu kendisinin daha kolay olan sürümünün verilmesinden sonraki ders/hafta verilmek zorunda değildir. Sıralama, birinci bileşenle çelişmediği müddetçe, etkinliklere göre yapılabilir. Üçüncü bileşen ise eski konuların yeni ve daha üst seviye olan bir düzlemde yeniden oluşturulmasına fırsat tanımayı işaret eder. "X" kavramının basit bir seviyede "Y" ve "Z" kavramlarının ise ileri seviyede verildiği var sayılsın. "X" kavramı ileri seviyede verildiğinde ileri seviye düzlemde bulunan Y ve "Z" kavramları ile entegre edilebilir. Böylece kavramlar arası ilişkiler oluşturulur. Bu durum "X" kavramının ileri bir düzlemde, o düzlemde bulunan kavramları dikkate alarak, yeniden oluşturulması anlamına gelmektedir. Bu programda bulunan etkinliklerde spiral öğrenmenin her bir bileşeni kullanılmıştır.

EŞLİ PROGRAMLAMA

Öğrenciler "Uygula ve Üret" adımlarını eşli programlama grupları içerisinde yapacaklardır. Eşli programlamada iki öğrenci, bir elektronik set ve bilgisayar karşısında yan yana otururak tasarım, elektronik devre, algoritma, kod yazma veya hata ayıklama için işbirlikli çalışır. Öğrencilerin "Uygula ve Üret" adımlarındaki görevleri tamamlayabilmeleri için elektronik devreyi oluşturmaları ve bilgisayar başında kodlama işlemini yapmaları gerekmektedir. Bir "Uygula veya Üret" adımında öğrencilerden birisi devreyi oluşturma görevinin sorumluluğunu alırken diğeri kod yazmanın sorumluluğunu üstlenir. Bir sonraki "Uygula veya Üret" etkinliğinde ise öğrenciler görev değiştirmelidir. Devre kuran öğrenci yeni etkinlikte kod yazmalı; kod yazan öğrenci ise yeni etkinlikte devre kurmalıdır. Her bir "Uygula/Üret" etkinliğinde sorumluluklar sırayla değiştirilerek etkinlikler tamamlanır. Bir öğrencinin sorumlu olduğu göreve diğer öğrenci müdahale etmez, ancak çözüm için fikir verebilir veya önerilerde bulunabilir. Görevden sorumlu olan öğrenci etkinliğin kendi bölümünü diğer öğrencinin önerilerini de göz önünde bulundurarak tamamlar. Bu dersteki eşli programlamada, elektronik devreyi oluşturan ve program kodunu yazan öğrenci düzenli olarak yer değiştirir. Öğrencilerin her iki görevden de öğreneceği şeyler vardır. Bu yüzden rol değiştirme çok önemlidir. Öğretmen öğrencilerin periyodik olarak görev değiştirmesini sağlamalıdır.

Eşli programlamada diğer bir önemli nokta, hangi iki öğrencinin eş olarak atanacağıdır. Burada, iyi bilen ve az bilen gibi, farklı bilgi veya beceri gruplarında öğrencilerin bir araya getirilmesi iyi bilenin az bilene öğretmesi açısından faydalı olarak görülebilir. Fakat pratikte bu fayda gerçekleşmemektedir. Bundan ziyade, iyi bilen bir müddet sonra diğerini kendine ayak bağı olarak görme eğilimi ve görece daha az bilen de iyi bilen ile iletişim kurmakta sıkıntı yaşayıp etkinliklerden kopma eğilimi gösterebilmektedir. Benzer bilgi ve beceri düzeyinde olan gruplar daha verimli çalışmaktadır. Bu yüzden benzer bilgi ve beceri düzeyinde olan öğrencilerin eş olarak belirlenmesi önemlidir. Bazı durumlarda eş olarak tayin edilen öğrenciler birbirleriyle ciddi anlaşmazlıklar ve çatışmalar yaşayabilirler. Çatışma yaşayan öğrencilerin aynı grupta kalması sağlıklı olmayacaktır. Bu durumu fark eden öğretmen, çatışma yaşayan eşleri farklı öğrencilerle yeniden eşlemelidir.

GRUPLAR ARASI İLETİŞİM

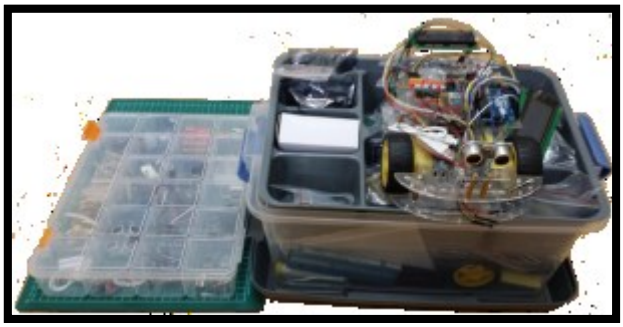
Etkinlikler boyunca gruplar arası bilgi alışverişine izin verilmelidir. Gruplar arasında paylaşımcı bir ortam oluşturulmalıdır. Fakat bu paylaşım, komple bir çözümün paylaşımı şeklinde olmamalıdır. Öğrenciler, çözüm yolları, stratejiler ve eksik bilgiler gibi konular için paylaşım yapabilirler. Fakat verilen bir problemin bütün çözümü paylaşılmamalıdır. Bu konu hakkında öğretmen, öğrencileri daha önceden bilgilendirmelidir ve onların ne tür bir paylaşım içerisinde olduğunu takip ederek gereğinden fazla olan paylaşımları engellemelidir.

DÖNEM PROJESİ

Bu derste öğrencilerin öğrendikleri bilgileri kullanarak özgün bir proje geliştirmeleri beklenmektedir. Projedeki amaç öğrencilerin yaratıcılıklarını tetiklemek, problem çözme, analitik düşünme ve grup halinde çalışma gibi bazı önemli becerilerini geliştirmelerine yardımcı olmaktır. Öğrenciler proje sürecinde elektronik devrelerini geliştirebildiklerini görecekler ve bu sayede kendilerine olan özgüvenleri de artacaktır. Proje geliştirme sürecinin, eğitimin son 6 haftasını kapsayacak şekilde planlanması önerilmektedir. Proje ile ilgili detaylı bilgi Ek 10'da (Proje Hazırlıyorum) verilmiştir.

DERSTE KULLANILAN SET

Bu ders kapsamında organizerlar, elektronik devre elemanları, mikrodenetleyiciler, kablolar, sensörler, modüller ve daha fazlasını içeren bir elektronik set kullanılmıştır. Bu set ortaokul ve lise kitabı yazarları tarafından laboratuvarlardaki ihtiyaçlar doğrultusunda oluşturulmuştur. Setin tüm içeriği liste halinde kitapta verilmemiştir. Fakat her etkinlik öncesinde kullanılan malzeme listesi verilmiştir. Aşağıdaki resimde derste kullanılacak set görülmektedir.



Resim 1. Derste Kullanılan Set

LABORATUVAR KULLANIM YÖNERGESİ

Elektronik Programlama ve Nesnelerin İnterneti dersi laboratuvar ortamında etkinlik bazlı gerçekleştirilecek bir derstir. Ders boyunca havya gibi yaralanmalara sebebiyet verme ihtimali olan ve transistor gibi yanma ihtimali olan devre elemanları kullanılacaktır. Bunun yanında derste kullanılacak devre elemanları düzensiz kullanım durumunda bozulabilir veya kaybolabilir. Bu durumlar göz önüne alındığında ilk haftanın ilk saatinden itibaren laboratuvar içerisinde bir düzen oturtulmalıdır. Rehber öğretmen ilk derste ve yeri geldikçe öğrencilere düzenli ve özenli çalışmaları gerektiği konusunda uyarılarda bulunmalıdır. Rehber öğretmen uyarılarda bulunurken çalıştığı öğrenci grubunun psikolojik alt yapısını dikkate almalıdır. Uyarılar rehber öğretmen tarafından itici veya sıkıcı kurallar olarak değil laboratuvarın sağlıklı ve verimli kullanımı için gerekli prosedürler olarak aksettirilmelidir.

Öğrenciler laboratuvarda devrelerini kesme matları üzerinde gerçekleştireceklerdir. Her grup kesme matı, küçük organizer kutu ve bilgisayarını düzenli bir şekilde masasına yerleştirmelidir. Öğrenciler büyük organizer kutularını masaya getirmemelidir. O hafta için gerekli olan devre elemanları öğretim kılavuzunun ilgili giriş bölümünde verilmiştir. Öğrenciler ders başında devre elemanlarını büyük organizer kutudan alıp küçük organizer kutuya aktarmalıdır ve masaya sadece küçük organizer kutuyu götürmelidir. İhtiyaç duyulduğunda büyük organizer kutudan yeni devre elemanı alınıp kullanılabilir.

Öğrenciler masalarda çalışırken devre elemanları, Arduino kartı ve bilgisayar düzenli bir şekilde durmalıdır. Gereğinden fazla dağınık masalarda çalışan veya bir başka grubun masasına taşan gruplar uygun bir dille uyarılmalıdır. Kaybolma ihtimali olan vida ve somun gibi gereçler ile çalışılırken büyük organizer kutusunun kapağı kullanabilir. Bu parçalar kapak üzerinde kullanılırsa düşüp kaybolma ihtimalleri azalır.

Öğrenciler aksi öğretim kılavuzunda belirtilmedikçe organizer kutularını aldıkları gibi ders sonunda teslim etmelidirler. Bunun için dersin son 10 dakikası mutlaka (veya rehber öğretmen tarafından uygun görülen bir süre) kurulan devrelerin sökülmesine ve devre elemanlarının organizer kutuları içerisinde istiflenmesine ayrılmalıdır. Rehber öğretmen öğrenciler hangi işle uğraşırlarsa uğraşsınlar dersin son 10 dakikasını mutlaka bu işlem için ayırmalıdır.

SIKÇA SORULAN PEDAGOJİK SORULAR

1. Öğrenciler Tasarla ve Üret ekinliklerini yaparken zorlanıyorlar ve/veya bitiremiyorlar. Bu durumlarda Tasarla ve Üret etkinliklerinin çözümlerini vermeli miyim?

Cevap: Rehber öğretmen hangi koşulda olursa olsun kesinlikle “Tasarla ve Üret” etkinliklerinin çözümlerini öğrencilere vermemelidir. “Tasarla ve Üret” etkinlikleri öğrencilerin “Gözle ve Uygula” etkinliklerinde öğrendikleri temel bilgileri kullanarak problem çözme ve soyutlama gibi ileri seviye zihinsel faaliyetlerde bulundukları etkinliklerdir. “Tasarla ve Üret” etkinliklerinde öğretmenler rehber pozisyonundadırlar. Rehber kişilerin sorularına çözüm üretmez. Onları sorularına çözüm bulmaları konusunda yönlendirir. Öğrencilere “Tasarla ve Üret” etkinliklerinin tasarlama ve çözümlerinin cevaplarını vermek öğrencilerin öğrendikleri bilgileri problem çözme ortamlarına aktarmaları durumlarını ortaya çıkarmamıza engel olabilir.

2. Öğrencilere “Gözle ve Uygula” etkinliklerinde soruların çözümlerini verdim. Bu yanlış bir uygulama mı?

Cevap: “Gözle ve Uygula” etkinliklerinde rehber öğretmen bir görev üzerinden öğrencilere bir uzmanın görevdeki amacı nasıl yerine getirdiğini uygulamalı olarak gösterir. Yani rehber öğretmen bir model pozisyonundadır. Burada görev temelli gidildiği için çoğu zaman doğrudan konunun anlatılmasından ziyade görev üzerinden konunun anlatılması gerçekleştirilir. Fakat yeri geldiğinde (çok sık olmamakla birlikte), rehber öğretmen konuları doğrudan da anlatabilir. Tüm bunlar göz önüne alındığında rehber öğretmenin Gözle ve Uygula etkinliklerinde bazı durumlarda öğrencilere soruların çözümünü vermesi normaldir.

3. Bu hafta sadece “Gözle ve Uygula” etkinliklerini bitirebileceğim. Zaman yetişmediği için hiç “Tasarla ve Üret” etkinliği yapamayacağım. Bu yaklaşım doğru mudur?

Cevap: Her hafta en az bir “Tasarla ve Üret” etkinliği mutlaka yapılmalıdır. “Tasarla ve Üret” etkinlikleri tamamen çıkarılamaz. “Tasarla ve Üret” etkinlikleri öğrencilerin öğrendikleri bilgileri problem çözme ortamlarına aktarma durumlarını ortaya çıkarmalarına yardımcı olmaktadır. Bu yüzden haftanın içeriğine hâkim olma durumlarını/ başka ortamlara aktarabilme seviyelerini görmelerine yardımcı olur. Sonuç olarak en az bir tane “Tasarla ve Üret” etkinliği mutlaka yapılmalıdır.

4. Bütün “Tasarla ve Üret” etkinliklerini ve “ilave etkinlikleri” bitiremiyorum. Öğrencilere yardımcı olup onları hızlandırsam nasıl olur?

Cevap: Rehber öğretmen bütün “Tasarla ve Üret” etkinliklerini ve “ilave etkinlikleri” bitirmek zorunda değildir. Ders boyunca farklı gruplar farklı hızlarda ilerleyebilir. Etkinlikler farklı gruplar için kademeli olarak tasarlanmıştır. Her grupta/sınıfta en az bir “Tasarla ve Üret” etkinliği yapıldıktan sonra rehber öğretmen konuları yetiştiremiyorum psikolojisine kapılmamalıdır. İlk “Tasarla ve Üret” etkinliği öğrencinin o günkü konu hakkında ileri seviye düşünme becerilerini geliştirmeleri için çok önemlidir. “İlave etkinlikler” bu becerilerin daha çok gelişmesine katkı sağlayacaktır. Bu yüzden öğrencilere yardımcı olup onları hızlandırmaya çalışmak doğru bir yaklaşım değildir.

5. Dersin sonuna geldim ve en az bir tane (veya daha fazla) “Tasarla ve Üret” ve “ilave etkinlik” yaptım. Dersin kalan kısmında ya “ilave etkinlik” ya da “Değerlendirme etkinliği” yapabilirim. Hangisini yapmalıyım.

Cevap: GUTÜD öğretim döngüsü öğrencilerin dersi daha iyi anlamaları, kavramları zengin bir şekilde oluşturmaları ve üst seviye bilişsel beceriler geliştirmeleri için oluşturulmuştur. Döngünün her bir adımı bu amaca ulaşmak için önemlidir. Değerlendirme etkinliklerinin atlanması önerilmez.

6. Derste gruplar arasında yarışmalar yaparak konuların daha verimli işleneceğini düşünüyorum. Bu tip yarışmalar yapmalı mıyım?

Cevap: Bu derste öğrenciler grup içi ve gruplar arası yardımlaşma ile ilerlemektedir. Bazı etkinlikler öğrencilerin tamamının katılmasını gerektirmektedir. Yani öğrenciler ortaklaşa üreterek öğrenmektedirler. Öğrenciler arasında yapılacak yarışmalar fazla olursa işbirlikli öğrenme ortamı zedelenebilir. Müfredatın dışına çıkıp yarışmalar yapmanız tavsiye edilmez.

7. Öğrenciler gruplar içerisinde işbirlikli çalışıyorlar. Eşli programlama prosedürleri gereksiz, sıkıcı veya uygulanması zor görünüyor. İşbirlikli çalışıp eşli programlama prosedürlerini uygulamasalar olmaz mı?

Cevap: Eşli programlama bu dersin tasarımı içerisinde temel bir yere sahip olan verimli bir yöntemdir. Eşli programlamadan vaz geçilemez.

8. Öğretim kılavuzunda verilen içerik yetersiz gibi duruyor. Bu durumlarda kendim konu ekleyip anlatmalı mıyım?

Cevap: Elektronik Programlama ve Nesnelerin İnterneti dersinde konuların işlenmesi sarmal bir şekilde tasarlanmıştır. Yani aynı konu farklı seviyelerde farklı haftalarda anlatılmış olabilir. Bu durumda gelecek haftalarda verilme ihtimali olan bir konunun rehber öğretmen tarafından erken anlatılması genel ders tasarımı ile ters düşebilir. Bu tip bir durumla karşılaşırsanız öğrencilerin öğrenmesinde ciddi sıkıntılar söz konusu değilse ileriki haftaların konularını gözden geçirip kararınızı vermeniz veya buna vaktiniz yok ise gelecek haftaları beklemeniz daha doğru olacaktır. Eğer öğrencilerin öğrenmesinde ciddi sıkıntılar yaşandığını düşünüyorsanız kendiniz yeni konular anlatabilirsiniz.

9. Öğretim kılavuzunu baştan sona inceledim devre yapımında ve kod yazımında modüler aritmetik ve seri-paralel bağlama gibi konular hiç anlatılmamış. Bunların gerekli olduğunu düşünüyorum. Bu konuları anlatmam uygun olur mu?

Cevap: Dirençlerin/kondansatörlerin/güç kaynaklarının seri veya paralel bağlanması, alternatif akım, Kirchoff kanunları, ivme, açısal hız ve modüler aritmetik gibi konular beşinci ve altıncı sınıf öğrencilerinin seviyelerinin üzerindedir. Bu konular derslerde henüz işlenmemiştir. Dersin tasarımında bu konular bilinçli olarak dışarıda bırakılmıştır. Rehber öğretmen tarafından anlatılması uygun olmaz.

10. Elektronik programlama ve nesnelerin interneti konularına fazlasıyla hâkimim. Öğretim kılavuzunu ders esnasında okumam yeterli. Hızlı bir şekilde anlayıp öğrencilere aktarabilirim. Bu şekilde ilerlemem nasıl olur?

Cevap: Öğretim kılavuzunda rehber öğretmenleri de zorlayacak seviyede problemler bulunmaktadır. Rehber öğretmen dersten bir hafta önce en azından ilgili haftanın öğretim kılavuzunu okumalıdır. Anlatılan devreleri ve programları deneme şansı varsa denemelidir. Aksi takdirde ders esnasında zor duruma düşebilir.

KAYNAKÇA

Çetin, İ., Üçgül, M., Top, E. &. Yükseltürk, E. (2021). Robotik Kodlama: Lise. Ankara: TÜBİTAK Popüler Bilim Kitapları.

Üçgül, M, Çetin, İ., Yükseltürk, E. & Top, E. (2021). Robotik Kodlama: Ortaokul. Ankara: TÜBİTAK Popüler Bilim Kitapları.

Wing, J. (2011). Research notebook: Computational thinking—What and why? The Link Magazine. Carnegie Mellon University, Pittsburgh. Erişim adresi: http://link.cs.cmu.edu/article.php?a=600

1. Bölüm: Elektronik Programlamaya Giriş

Ön bilgi:

- Öğrenciler robotik kavramını bilir.
- Öğrenciler robotik tasarım ve programlama becerisine sahiptir.

Bölümün Kazanımları:

- Öğrenciler gerilim, akım ve direnç tanımını yapabilir.
- Öğrenciler direncin çalışma prensiplerini bilir.
- Öğrenciler akım ve direnç arasındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamına giriş yapabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında basit kodlama yapabilir.
- Öğrenciler ArduinoBlocks ortamında geliştirdikleri programları Arduino'ya aktarabilirler.
- Öğrenciler Arduino pinlerinin ne işe yaradığını açıklayabilir.
- Öğrenciler breadboard çalışma prensiplerini bilir.
- Öğrenciler led ve direnç devre elemanlarının breadboard bağlantılarını yapabilir.

Bölümün Amacı:

Bu bölümün amacı, öğrencilerin direnç ve led gibi temel elektronik devre elemanlarının çalışma prensipleri anlamaları ve breadboard kullanılarak bu elemanların bağlantılarını yapabilmelerini sağlamaktır. İlerleyen süreçlerde diğer devre elemanlarının da kullanılması ve karmaşık devrelerin hazırlanması sırasında bağlantı kurma becerilerinin temelini oluşturmak hedeflenmektedir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Direnç (10 adet 220 ohm, bir adet 560 ohm, 1 adet 1 kohm), 10 adet kırmızı led, 10 adet yeşil Led, 10 adet sarı led, çeşitli tipte ve renkte bağlantı kabloları, breadboard, Arduino Uno, 9V pil (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Seçilen temel elektronik devre elemanlarının tanımlanması, kullanılması ve akım ve gerilim gibi temel kavramlar ile ilişkisinin örnekler üzerinde incelenmesi.

Uygula: Seçilen temel elektronik devre elemanlarının kullanıldığı örnek devrelerin ve programlarının oluşturulması.

Tasarla: Belirlenen senaryolar için gerekli elektronik devre elemanlarının belirlenmesi ve devrelerin tasarımının ve programının özelliklerinin ortaya konulması.

Üret: Belirlenen senaryolar için tasarla adımında planlanan elektronik devrelerin oluşturulması ve programlarının yazılarak hedeflere ulaşıldığının kontrol edilmesi.

Değerlendir: Yansıtma Etkinliği.

1. GÖZLE VE UYGULA

1.1 Gözle- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç

Günlük yaşamın hemen her alanında "elektrik" ile çalışan araçlar kullanılır. Örneğin yiyeceklerin tazeliğini koruyan buzdolapları, çizgi film izlenilen televizyonlar, oyun oynanan tablet bilgisayarlar ve cep telefonları çalışmak için elektriğe ihtiyaç duyar. Elektrik bir enerji türüdür. Bahsedilen bu araçlar pil, batarya ve duvar prizlerinden alınan şebeke elektriği gibi kaynaklar aracılığı ile elektrik enerjisini kullanır. Elektrik enerjisini nitelemek için iki temel kavram bilinmelidir. Bu kavramlar **gerilim** ve **akımdır**. Gerilim elektrik kaynağının iki ucu arasındaki potansiyel enerji farkıdır. Elektriğin kablo aracılığıyla bir noktadan başka bir noktaya akmasını sağlar. Gerilim birimi **Volt** olarak ifade edilir. Akım ise bir noktadan diğer noktaya olan elektrik akışının kendisidir. Aslında bir noktadan diğerine akan şey elektron olarak adlandırılır. Yani akım bir kablo içerisindeki elektronların hareketidir.

Dikkat

Bir noktadan diğer noktaya akan şey sadece elektron olmak zorunda değildir. Pozitif ve negatif yüklü parçacıklar (iyonlar) da hareket ederek akımı oluşturabilir. Elektrik devrelerinde akımı elektron akışı oluşturur. Dolayısı ile bu ders kapsamında elektrik devrelerinin kullanıldığı yerlerde akım elektron akışı olarak kabul edilecektir. Rehber öğretmen bu bilgiyi mutlaka vurgulamalıdır.

Akım birimi **Amperdir**. Elektronlar çok küçük oldukları için gözle görülemezler. Elektronların akışı bir benzetme ile daha iyi açıklanabilir. Bunun için öğrencilerden alt kısmında bir delik olan büyük bir su kovası hayal etmeleri istenir. Bu öyle bir kovadır ki delikten akan su anında kovanın içine geri gelmektedir ve suyun yüksekliği sabit kalmaktadır. Su seviyesi ne kadar yüksek olursa delikten akan suyun miktarı o kadar çok olur. Su seviyesi azaldıkça suyun akışı da aynı şekilde azalır. Bu durumda, gerilimi su kovasının içindeki suyun yüksekliğine, su akışı da akıma benzetilebilir. Yeniden tekrarlamak gerekirse elektronların akmasını sağlayan potansiyel enerjiye gerilim denilir. Elektronların kablo içerisinde akması ile oluşan elektrik olayına ise akım denilir. Gerilim ne kadar artarsa akım da o oranda artar.

Elektronlar bir noktadan başka bir noktaya akarken elektronun içinden aktığı malzemeler bu elektron akışına karşı direnç gösterebilir ve elektron akışını yavaşlatabilir. İşte elektron akışını yavaşlatan bu malzemelere **direnç** ismi verilir. Direnç miktarı **Ohm** cinsinden ifade edilir. Dirençler elektron akışına karşı koydukları için direnç ismini almışlardır. Su kovası benzetmesine geri dönülecek olursa kovanın altındaki delikten akan suyun akışına herhangi bir materyal aracılığı ile karşı konulduğunda suyun akışını azaltan bir direnç oluşturulur. Bu şekilde delikten akan su miktarı da azalmış olur. Elektrik akımını engelleyen dirençler de mevcuttur. Yani bir noktadan diğer noktaya akan elektrik akımına karşı olan, onu azaltan unsurlar bulunmaktadır.

1.1. Uygula- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç

Rehber öğretmen öğrencilere aşağıdaki soruları sorar ve öğrencilerin bu soruların cevabını sınıfça tartışmalarını sağlar.

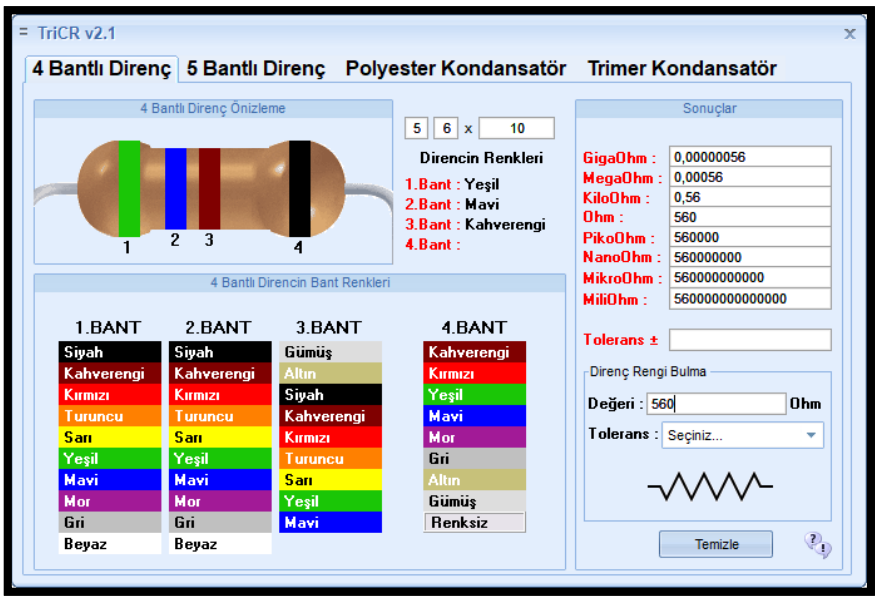
1. Yukardaki benzetmede suyun yüksekliği artırıldığında delikten akan suyun sabit kalması niçin ne yapılabilir?
2. Telefonların ve bilgisayar ekranlarının parlaklığı, bazı odalarda bir düğme çevirerek odanın aydınlığı veya bazı vantilatörlerin hızı ayarlanabilir. Sizce bu ayarlama işlemi nasıl gerçekleşmektedir.

1.2 Gözle ve Uygula- Basit Bir Devre Oluşturma

Malzeme Listesi:

9 volt pil
Bağlantı kabloları
Led
560 ohm direnç

Basit bir devre oluşturmak için en az üç temel unsura ihtiyaç vardır; güç kaynağı, bağlantı kabloları ve devre elemanları. Bu etkinlikte "led" yakan bir devre oluşturulacaktır, "led"in fazla akım çekip yanmaması için direnç kullanılır. Bu yüzden bu devrede 560ohm direnç kullanılacaktır. Direnç değerleri direnç üzerinde bulunan renkler aracılığıyla belirlenir. Bu derste direnç okuması öğretilmeyecektir. Bunun yerine pr.1 isimli uygulama kullanılacaktır (Bu uygulama İnternetten indirilebilir). "TriCR" uygulaması var olan bir direncin büyüklüğünü belirlemek veya istenilen büyüklükteki direncin renk kodlarını belirlemek için kullanılır. Direnci seçmek için öncelikle "TriCR" uygulaması açılır. Uygulamadaki "4 Bantlı Direnç" sekmesi seçilir ve "Değeri" hücresine 560 girilir. Aşağıdaki resimde direnç üzerinde oluşan renkler (yeşil, mavi, kahverengi) 560 ohm direnci göstermektedir. Diğer renklerden ayrı duran dördüncü bant direnç değeri ile ilgili değildir. Öğrenciler bu renkleri kullanarak devre için gerekli direnci seçer.

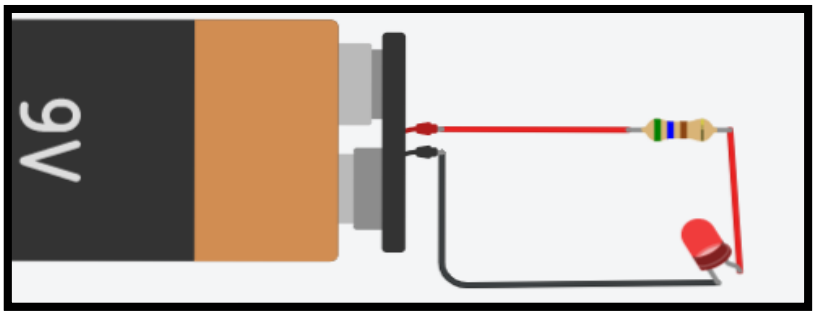


Resim 2. TriCR Programı Ekran Görüntüsü

Bu etkinlikteki devre oluşturulurken breadbord kullanılmayacaktır. Rehber öğretmen devreyi kurarken, öğrencilere doğru akım güç kaynaklarının bir artı ve bir eksi ucu olduğundan bahseder. Artı uç ve artı uç için kullanılan kablolarda genellikle kırmızı renk kullanılır. Fakat bu bir kural değildir. Kolaylık olsun diye belirlenmiş bir gelenektir. İstenildiğinde farklı renkler kullanılabilir. Güç kaynağından gelen eksi uç ve eksi uca bağlanan kablolarda genellikle siyah veya kahverengi tercih edilir. Bu da bir zorunluluk değildir. Bu bilgilerin ardından rehber öğretmen dirençlerin artı veya eksi ucunun olmadığından ve yönü fark etmeden istenilen şekilde devreye takılabileceğinden bahseder. Led için bu durum geçerli değildir. Led'lerin artı ve eksi bacakları bulunur. Aşağıdaki resimde görüldüğü gibi "led"de bulunan uzun bacak "artı" uca kısa bacak ise "eksi" uca gelecek şekilde bağlanmalıdır.

Resim 3. Led Bacakları

Rehber öğretmen devreyi kurduktan sonra devreye güç vererek çalıştırır ve çalışan devreyi öğrencilere gösterir. Bunu takiben rehber öğretmen öğrencilerden aşağıdaki resimde yer alan devreyi kurup çalıştırmalarını ister.



Resim 4. Devre Şeması

Rehber öğretmen bütün öğrenciler devreyi tamamladıktan sonra onlara aşağıdaki soruları sorar ve soruların sınıfça tartışılmasını sağlar. Eğer öğrencilerden doğru yanıtlar gelmezse öğretmen soruların yanıtını verir. Fakat rehber öğretmen ilk etapta soruların yanıtını vermemelidir.

i. Bu devrede gerilim arttırılırsa led'in verdiği ışık miktarı nasıl değişir? Sebepleriyle açıklayınız.

ii. Bu devrede yer alan direncin büyüklüğü arttırılırsa led'in verdiği ışık miktarı nasıl değişir? Sebepleriyle açıklayınız.

iii. Çevrenizde pille çalışan araçlar var mı? Bu araçların kaçar adet pile ihtiyaç duyduklarını biliyor musunuz? Bu araçlarda farklı sayıda pile neden ihtiyaç duyulur?

- Yanıt: piller birbirine paralel ve seri bağlanarak gerekli akım ve gerilim ayarlamaları yapılabilir.

1.3 Gözle ve Uygula- Breadbord ve ArduinoUno ile Basit Bir Devre Oluşturma

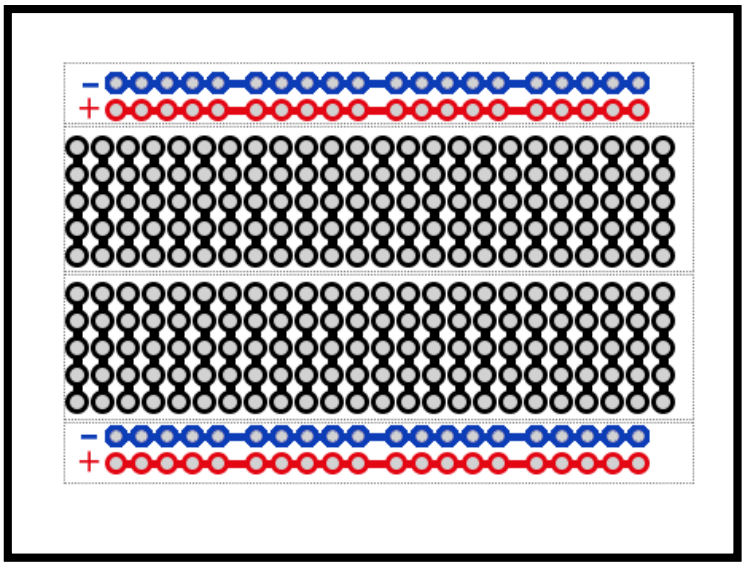
Malzeme Listesi:

ArduinoUno
Breadboard
Bağlantı kabloları
Led
560 ohm direnç

Uyarı

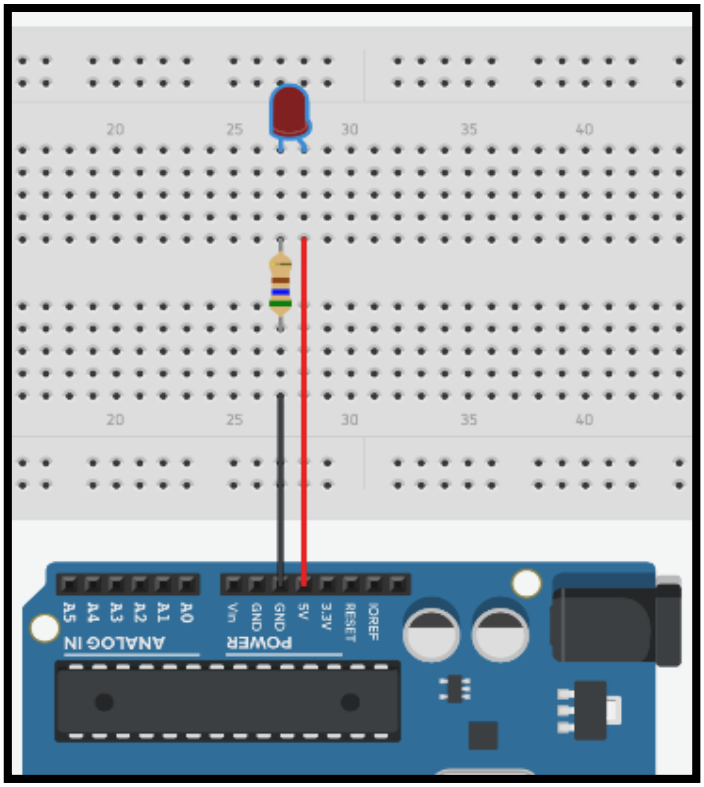
Rehber öğretmen öğrencileri bu ders boyunca direnç olmadan "led" kullanmamaları gerektiği konusunda uyarır.

Arduino Uno aslında programlanabilir bir işlemci ve diğer kart elemanlarını barındıran bir mikro denetleyicidir. Fakat bu etkinlikte mikro denetleyici özellikleri kullanılmayacaktır. Burada Arduino Uno yalnızca bir güç kaynağı olarak kullanılacaktır. Devreyi oluşturmak için gerekli bağlantıların yapılması için bir "breadboard"a ihtiyaç duyulur. Breadboard hızlı bir şekilde devre elemanları arasında bağlantı yapmak için kullanılır. Aşağıdaki resimde görüldüğü gibi "breadboard" üzerinde dikey ve yatay bağlantılar bulunur. Dikeyde ve yatayda belirtilen doğrular "breadboard"da birbirlerine bağlıdır. Rehber öğretmen öğrencilere dikey ve yatay bağlantıları anlatır. Yatay bağlantılarda yer alan artı ve eksi işaretleri aslında bir zorunluluk değildir. Güç kaynağından gelen bağlantıların artı ve eksi uçlarının karıştırılmaması için kolaylık sağlayan bir özelliktir.



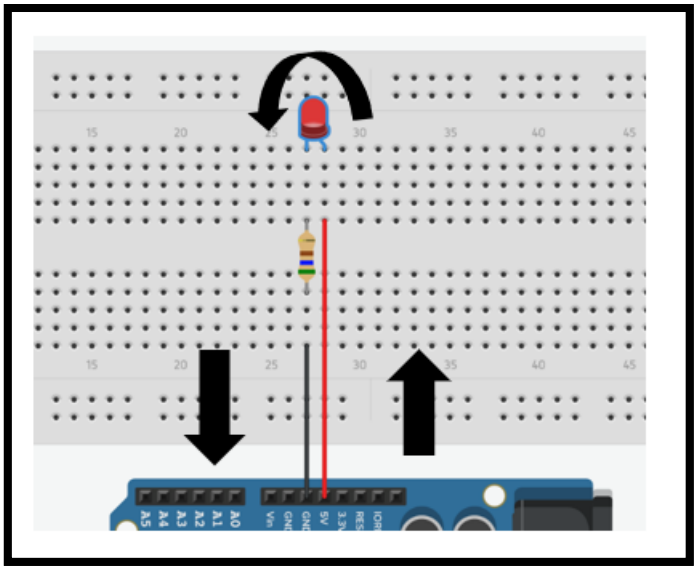
Resim 5. Breadboard İç Yapısı

Breadboard yardımıyla aşağıdaki resimde görülen devre kurulur ve bilgisayardan USB kablosu aracılığıyla güç verilerek çalıştırılır.



Resim 6. Devre Şeması

Devre kurulurken led'in eksi (katot) bacağının dirençle bağlantılı olduğu vurgusu yapılır. Led'in eksi bacağı direnç üzerinden GND'ye yani eksi (-) bağlantı noktasına ulaşmaktadır. Burada istenirse katot doğrudan eksiye bağlanırken, anot direnç üzerinden 5V'ye bağlanabilir. İki durumda da devreden geçen akım aynı olacaktır. Tek fark akımın önce geçeceği devre elemanının değişmesidir. Sonrasında bağlantı kablolarından biri, bir ucu kart üzerindeki 5V yuvasına diğer ucu da led'in artı bacağının (anot) bulunduğu sütuna takıldığı anlatılır. Son bağlantı kablosunun da direncin alt ucu ile kartın GND yuvasını bağlamak için kullanıldığı belirtilir. Bu şekilde devre tamamlanmış olur. Rehber öğretmen öğrencilere Arduino'nun GND (-) bacağından akımın akarak direnç ve led üzerinden geçtiğini ve ardından 5V (+) bacağına gelerek devreyi tamamladığını anlatır. Fakat devreler incelenirken akımın 5V (+) bacağından GND (-) bacağına hareket ettiğini kabul etmek gerektiğini vurgular. Akımın aslında bir yük değişimi olduğu ifade edilir. Bu nedenle elektrik akımını artıdan eksiye doğru gittiğini kabul etmenin gerektiği ayrıca vurgulanır. Bu durum aşağıdaki resimde gösterilmiştir.



Resim 7.Elektrik Akımının Kabul Edilen Yönü

Rehber öğretmen öğrencilerden uyarıdaki resimde gösterilen devreyi kurmalarını ister. Rehber öğretmen öğrencilere devre kurma konusunda gerektiğinde yardımcı olur. Bütün öğrenciler devreyi kurduktan sonra öğrencilere aşağıdaki soruları sorar ve öğrencilerin soruyu sınıfça tartışmalarını sağlar. Eğer öğrencilerden doğru yanıtlar gelmezse rehber öğretmen soruların yanıtını verir. Fakat rehber öğretmen, ilk etapta soruların yanıtını vermemelidir.

i. Devrede neden bir direnç kullanılmıştır?

ii. Devrede direnç kullanılmasaydı ne gibi sorunlar ile karşılaşılabilirdi?

iii. Devredeki "led"i ters bağlayalım. Devremiz hala çalışıyor mu?

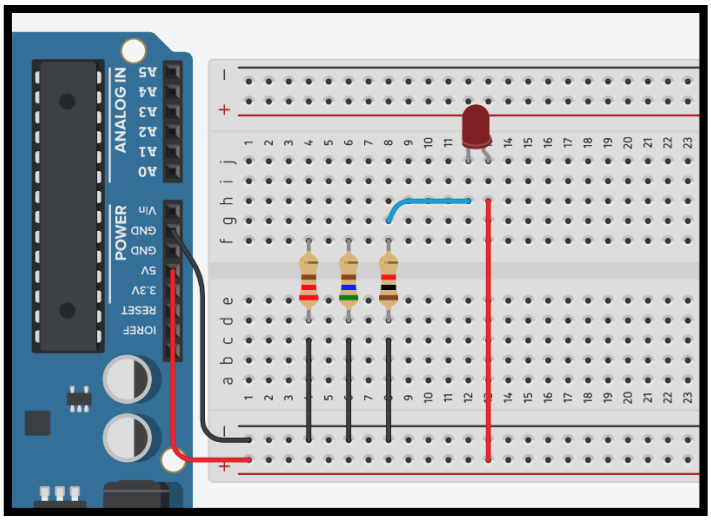
1.3 Uygula- Basit Bir Devre Oluşturma ve Direnç Değerlerini Değiştirme

Malzeme Listesi:

Arduino Uno
Breadboard
Bağlantı kabloları
Led
220 ohm, 560 ohm, 1 k ohm direnç

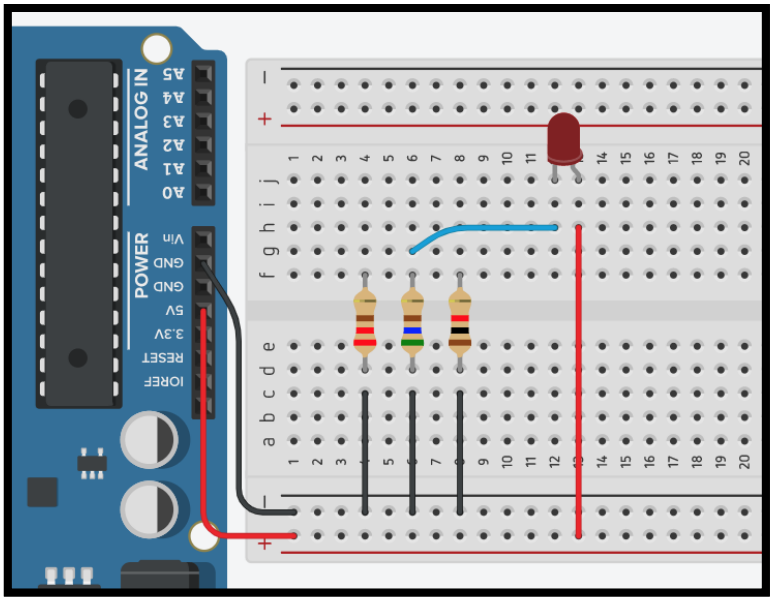
Bu etkinlikte amaç temsili şekilde bir odayı farklı ışık seviyelerinde aydınlatmaktır. Gerçekte odanın aydınlatılması gerçekleştirilmeyecektir. Bu devrede kullanılacak led odayı aydınlatacak lambayı temsil etmektedir. Led'in (gerçekte lamba) okuma yapmak için yüksek ışık, televizyon izlemek için orta düzeyde ışık ve dinlenmek için ise düşük seviyede ışık vermesi gerekmektedir. Rehber öğretmen öğrencilerden bu işlemi gerçekleştiren basit bir devre kurmalarını ister. Rehber öğretmen gerekirse öğrencilere farklı değerli dirençleri kullanarak devreden geçen akım miktarını ve dolayısı ile led'in parlaklık miktarını ayarlayabilecekleri hakkında ipucu verebilir. Fakat rehber öğretmen öncelikle öğrencilerin çözümü bulmak için düşünmelerine fırsat vermelidir. Hemen ipucunu söyleyerek öğrencilerin düşünme sürecini azaltmamalıdır. Rehber öğretmen ancak öğrencilerden çok zorlananlar olursa ipucu vermelidir.

Aşağıdaki resimde görülen devre şeması bu etkinlik için örnek bir çözümdür. Bu devrede bir adet led, 1 adet 220 ohm, bir adet 560 ohm, 1 adet 1 k ohm direnç, bağlantı kabloları, "breadboard" ve bir Arduino kart kullanılarak aşağıdaki devre oluşturulur. Belirtilen dirençleri belirlemek için "TriCR" uygulaması kullanılır.



Resim 8. Devre Şeması

Devredeki led odadaki lambayı temsil etmektedir. Üç farklı direncin birer uçları breadboardun alt kısmındaki yatay bağlantı kullanılarak Ardiuno kartının GND ucuna bağlanmıştır. Burada bilinmesi gereken nokta yatay bağlantıda yer alan sıralar bir uçtan diğer uca birbirine bağlıdır. Yani GND ucu ve dirençlerin bir uçları birbirine bağlı durumdadır. Bu yöntem fazla sayıda GND bağlantısı gerektiren durumlarda bu ucu çoğaltmak için kullanılmaktadır. Aynı durum 5V bağlantısı için de geçerlidir. İhtiyaç durumunda 5V de aynı şekilde çoğaltılabilir. Arduino kartının güç bağlantısı yapıldığında led yanar. Devre akımı 220 ohm değerindeki direnç ve led üzerinden geçerek tamamlanır. Sonra, en solda yer alan 220 Ohm değerindeki direncin led'in eksi (katot) ucuna bağlı kablosu çıkarılarak, kablo ortada yer alan 560 ohm değerindeki direncin ucuna bağlanarak devre oluşturulur. Bu şekilde, devre 220 ohm değerindeki direnç yerine 560 ohm değerindeki direnç üzerinden tamamlanmış olur. Devre şeması aşağıdaki resimde görülmektedir.



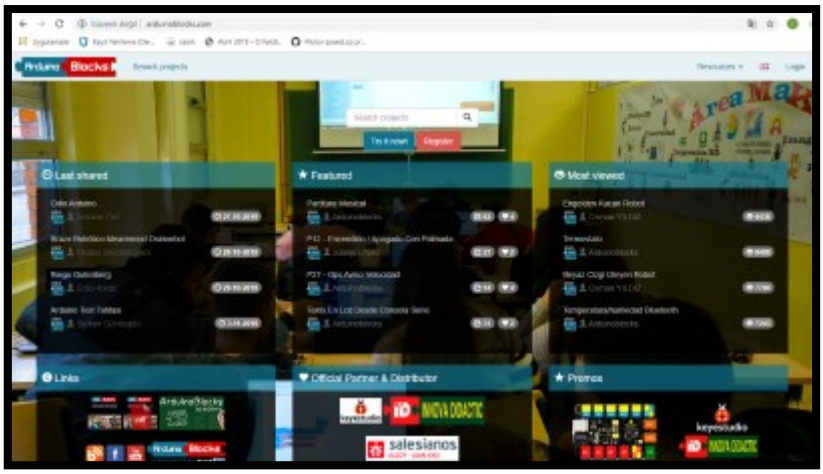
Resim 9. Devre Şeması - Farklı Direnç

Yanan led'in parlaklığı incelenir. Daha sonra aynı işlemler tekrarlanarak 1 k ohm değerindeki direncin bağlantısı yapılır ve parlaklık kontrol edilir.

1.4 Gözle – Uygula ArduinoBlocks Geliştirme Ortamına Giriş

Arduino programlanabilir bir mikrodenetleyicidir. Şimdiye kadar kurulan devrelerde Arduino sadece bir güç kaynağı olarak kullanılmıştır. Fakat Arduino bundan çok daha fazlasını yapabilir. Arduino'nun mikrodenetleyici özelliklerini kullanmak için bir programlama diline ihtiyaç vardır. Elektronik Programlama ve Nesnelerin İnterneti dersinde bu amaç doğrultusunda ArduinoBlocks kullanılacaktır. ArduinoBlocks blok tabanlı çevrimiçi bir Arduino geliştirme ortamıdır. Rehber öğretmen ArduinoBlocks geliştirme ortamının nasıl kullanılacağını projeksiyon ile yansıtarak aşağıdaki resimde öğrencilere adım adım anlatır:

i. Bir web tarayıcısı yardımıyla http://www.arduinoblocks.com/ adresi açılır.



Resim 10. ArduinoBlocks Sitesi

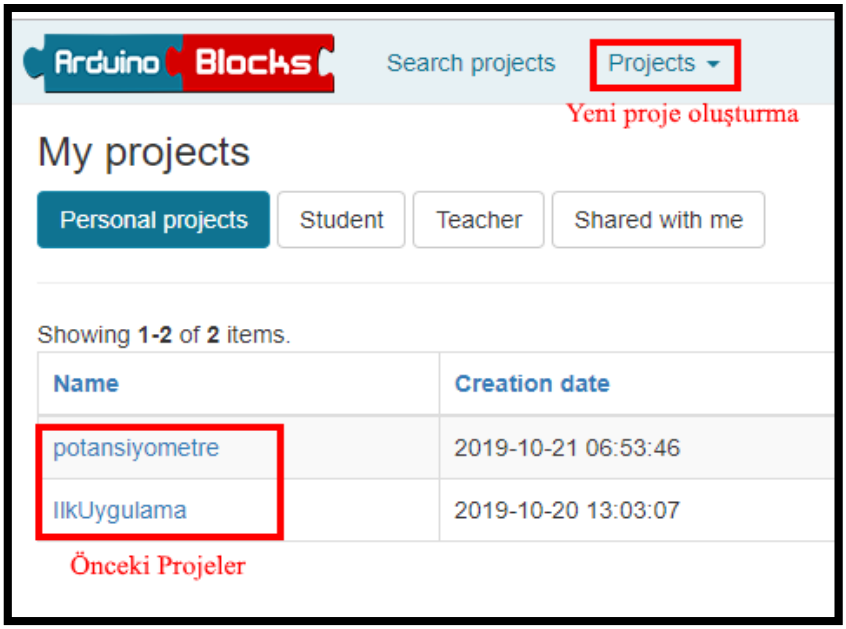
ii. ArduinoBlocks web sayfasının sağ üst köşesinde aşağıdaki resimde görüldüğü gibi "Login" bağlantısı bulunmaktadır. ArduinoBlocks geliştirme ortamını kullanabilmek için öncelikle sisteme giriş yapılmalıdır. Login bağlantısına tıklanır.

Resim 11. Siteye Giriş

iii. Login bağlantısı tıklandığında aşağıdaki resimde görünen arayüz tarayıcı ekranına gelir. Bu adımın gerçekleştirilebilmesi için rehber öğretmen ders öncesinde her bir öğrenci için bir "e-posta" hesabı alıp bu "e-posta" hesabı vasıtası ile öğrencileri ArduinoBlocks sitesine kaydetmiş olmalıdır. Ders esnasında rehber öğretmen öğrencilere e-posta hesaplarını ve şifrelerini (her bir öğrenci için alınmış olmalıdır) vererek ArduinoBlocks sistemine aşağıdaki resimde gösterildiği gibi girmelerini sağlar.

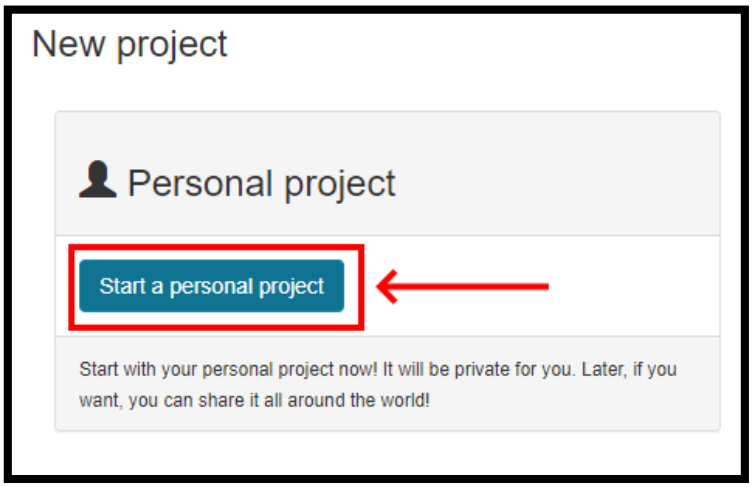
Resim 12. Giriş Ekranı

iv. Burada aşağıdaki resimde gösterildiği gibi gelen ekranda eğer varsa daha önceden oluşturulan projeler görülür. Yeni bir proje oluşturmak için Projects'e tıklanır ve açılan menüden New Projects seçilir.



Resim 13. Proje Ekranı

v. Aşağıdaki resimde görüldüğü gibi tarayıcı ekranına gelen seçeneklerden "Start a personal project" bağlantısına tıklanır.



Resim 14. Yeni Proje

vi. Açılan ekranda "Project Type" ArduinoUno olarak seçilir. Name metin kutusuna projeye verilmek istenen isim girilir. Sayfanın en alt kısmında bulunan "New Project" butonuna tıklanır.

vii. Projenin oluşturulacağı ekrana aşağıdaki resimde görüldüğü üzere ulaşılmıştır. Ekranın sol tarafındaki kod blokları öğrencilere kısaca gösterilir ve ardından ekranın orta kısmının (Setup ve Loop bloklarının bulunduğu kısım) program yazmak için kullanılacağı öğrencilere anlatılır. "Setup ve Loop" blokları Arduino programlarında mutlaka bulunmalıdır. "Setup" bloğunun içerisine yazılan kodlar Arduino'ya ilk yükleme esnasında bir kere çalıştırılır. Buraya genellikle ayarlar ile ilgili komutlar yazılır. "Loop" bloğu içerisindeki kodlar ise sürekli çalıştırılır. Asıl program kodları "Loop" bloğu içerisine yazılır.

Gruplardaki her iki öğrencinin de sisteme bir defa giriş yapması sağlanır.

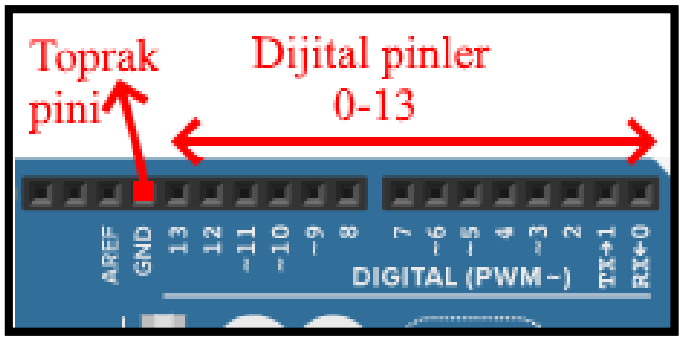
Resim 15. ArduinoBlocks Kodlama Ortamı

1.5 Gözle- Uygula Arduino ile Led Yakıp Söndürme

Malzeme Listesi:

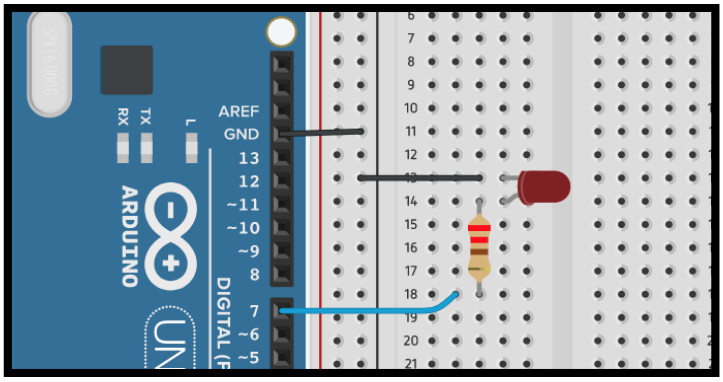
ArduinoUno
Breadboard
Bağlantı kabloları
Led
220 ohm direnç

Bu etkinlikteki amaç "ArduinoBlocks" ortamı ile periyodik olarak yanıp sönen "led" etkinliği yapmaktır. Bu etkinlikte öncelikle Arduino'nun dijital "pinleri" tanıtılmalıdır. Bu "pinler" yardımı ile hem dijital girdi alınabileceği hem de dijital çıktı verilebileceği vurgulanır. Bu "pinler" ile devre elemanlarına "5V" (ON-HIGH) veya "0V" (OFF-LOW) gönderilebilir. Bu "pinler" aynı zamanda dışarıdan gelen "5V" veya "0V" değerlerini algılamak için de kullanılır. Daha sonra "GND" pini gösterilir. Bunun İngilizce "ground" (toprak) kelimesinden geldiği vurgulanır ve herhangi bir kaynaktan sağlanan akımın GND üzerinden devreyi tamamlaması gerektiği vurgulanır. Aşağıdaki resimde Arduino Pinleri görülmektedir.



Resim 16. Arduino Pinleri

Rehber öğretmen nasıl kurulduğunu göstererek öğrencilerin aşağıdaki resimde görülen devreyi kurmalarını sağlar. Öğrencilere "ArduinoBlocks" ortamı sayesinde Arduino kartının 7. dijital "pininden" çıktı alınacağı ve alınan akımın GND üzerinden devreyi tamamlayacağı anlatılır.



Resim 17. Örnek Devre Şeması

ArduinoBlocks geliştirme ortamında aşağıdaki resimde görülen program yazılır ve programın nasıl yazıldığı projeksiyondan yansıtılarak öğrencilere gösterilir. Programı yazmak için "Input/Output" menüsündeki "Write Digital Pin" bloğu kullanılır.

Resim 18. Örnek Program

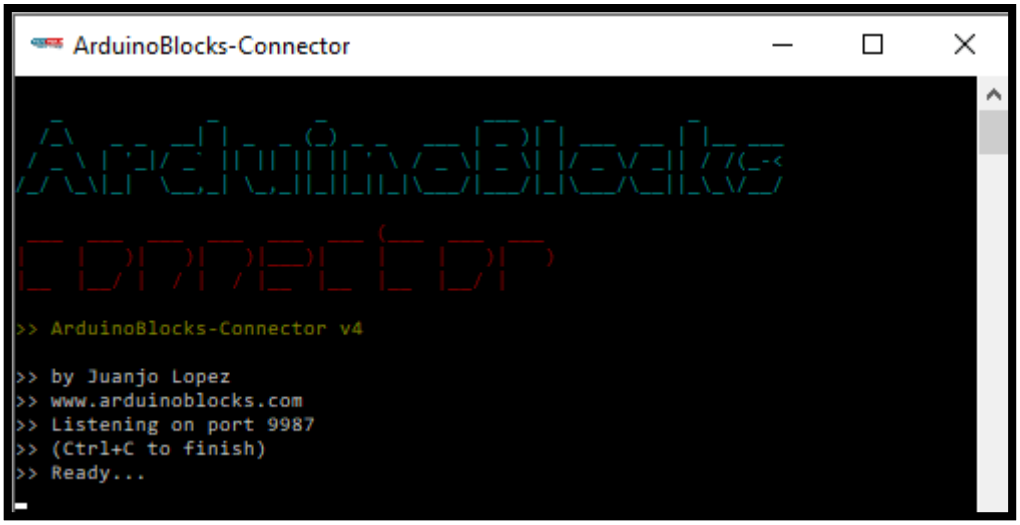
Arduino kartı bilgisayarın USB portuna kablo aracılığı ile bağlanır. ArduinoBlocks ortamında yazılan program aşağıdaki resimde göründüğü gibi Arduino'ya yüklenir ve bu yüklenme süreci öğrencilere adım adım gösterilir.

Resim 19. Programın Arduino Karta Yüklenip Çalıştırılması

Uyarı

Bağlantısı noktasının yanlış seçilmesi durumunda kodlar Arduino'ya yüklenemeyecektir.

Burada öğrencilere aşağıdaki resimde görünen ArduinoBlocks-Connector uygulamasının açık olmadan yazılan programların Arduino'ya yüklenemeyeceği vurgulanır.

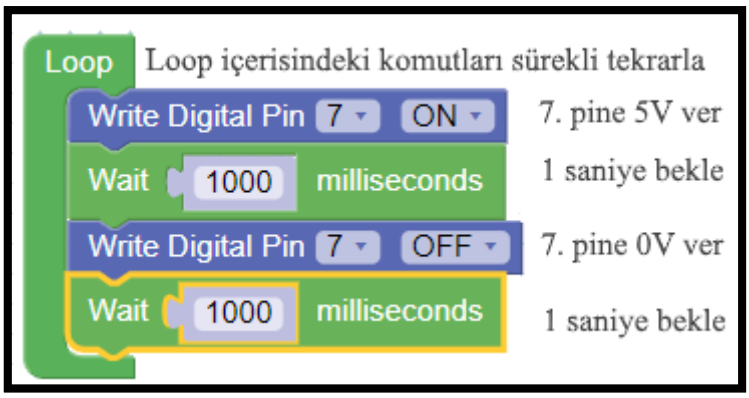


Resim 20. Arduino Blocks Connector Ekran Görüntüsü

Dikkat

ArduinoBlocks-Connector dersten önce öğrencilerin bilgisayarına rehber öğretmen tarafından yüklenmiş olmalıdır.

Rehber öğretmen, öğrencilerin yukarıdaki programı yazıp çalıştırmalarını sağlar. Bunun ardından öğrencilere, bu programın sadece 7. dijital pin'e bağlı olan bir led'i sürekli yaktığı söyler. Fakat etkinlikteki amaç bir led'i periyodik olarak yakıp söndüren bir program yazmaktır. Bu amaç doğrultusunda aşağıdaki resimde görülen program öğrencilere anlatılarak yazılır. Devamında öğrencilerin aynı kodu yazıp Arduino'ya yükledikten sonra çalıştırmaları sağlanır. "Wait" kod bloğu "Time" menüsündedir.



"

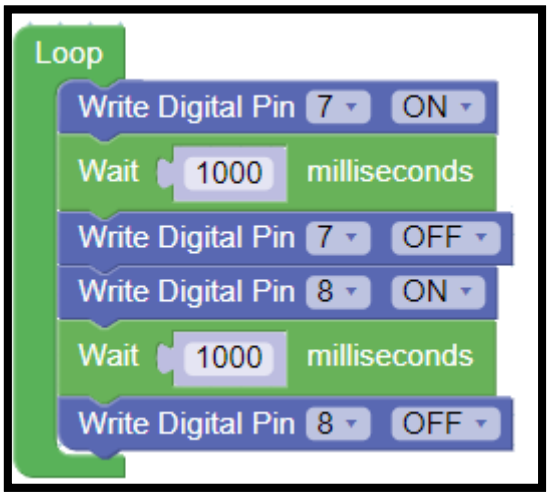
Resim 21. Örnek Program

1.6 Uygula- Flip Flop

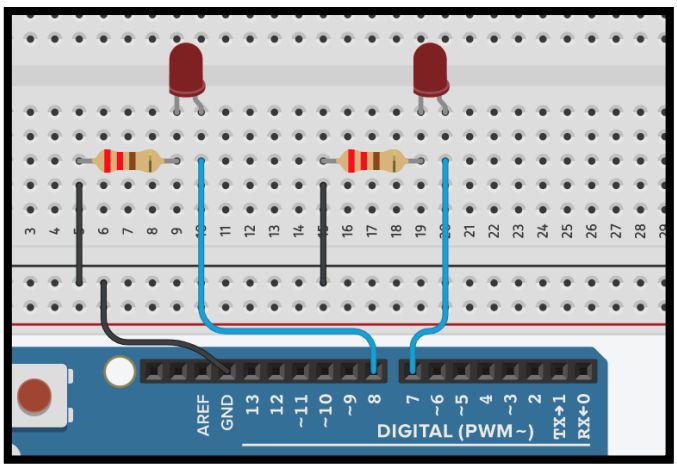
Malzeme Listesi:

ArduinoUno
Breadboard
Bağlantı kabloları
2 tane Led
2 tane 220 ohm direnç

Bu etkinlikteki amaç Arduino'nun 7. ve 8. dijital pinlerine bağlanmış iki led'i sırayla yakıp söndürmektir. Farklı bir ifade ile önce 7. pin'e bağlı led bir saniye boyunca yanmalıdır. Bu led söndükten hemen sonra 8. pin'e bağlı led bir saniye yanmalıdır. 8. pine bağlı led bir saniye sonra sönmektedir ve bu işlem sürekli tekrar etmelidir. Rehber öğretmen öğrencilerin bu uygulamayı yapmasını sağlar. Bu görev için örnek çözüm aşağıdaki resimlerde verilmiştir.



Resim 22. Örnek Program



Resim 23. Örnek Devre Şeması

Dikkat

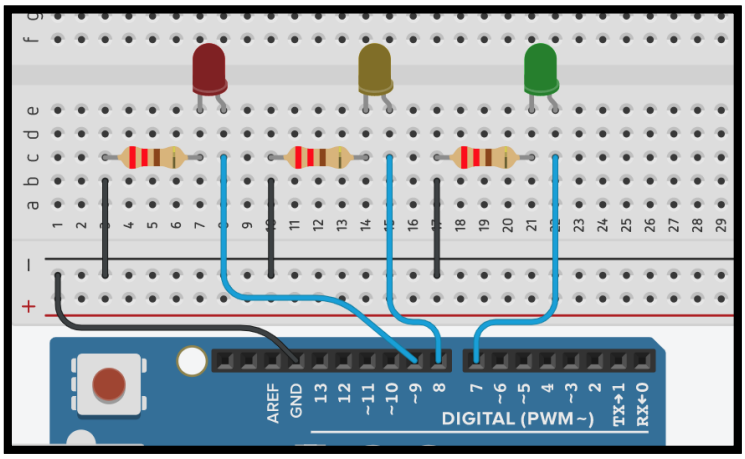
Bu etkinlikte led'ler direnç kullanmadan pin'lere bağlanırsa patlayabilir. Led bağlantısı için mutlaka 220 ohm dirençler kullanılmalıdır.

1.7 Uygula- Trafik Işığı

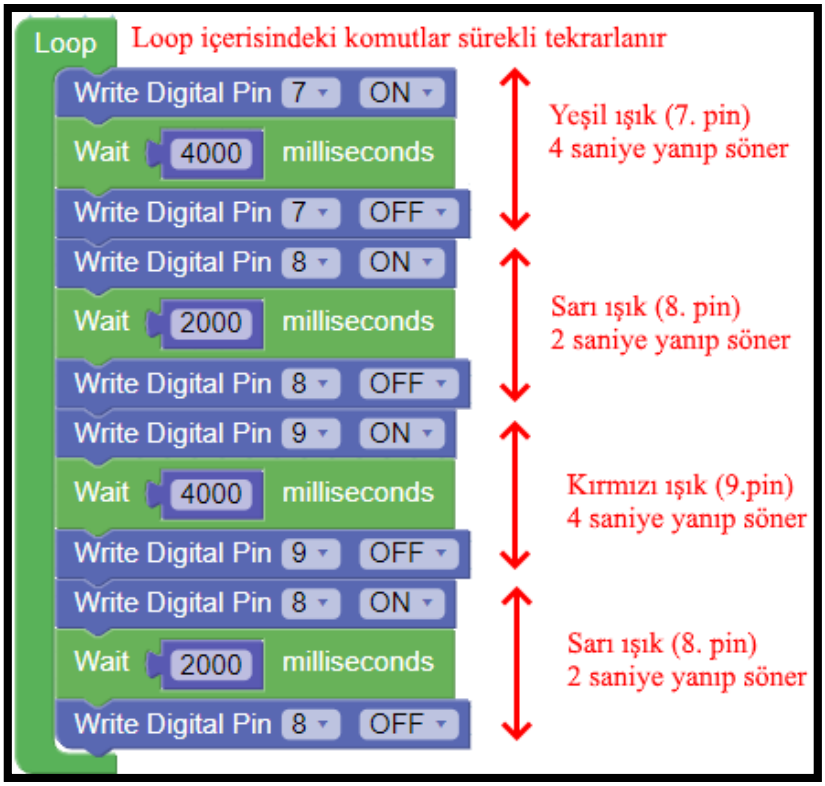
Malzeme Listesi:

ArduinoUno
Breadboard
Bağlantı kabloları
1 tane sarı, 1 tane kırmızı ve 1 tane yeşil Led
3 tane 220 ohm direnç

Bu etkinlikte yeşil, sarı ve kırmızı led'ler kullanılarak trafik lambasının bir örneğini yapmak hedeflenmektedir. Yeşil ışık 4 saniye, sarı ışık 2 saniye, kırmızı ışık 4 saniye yanmalıdır. Bu süreler trafik ışıklarında kullanılan sürelerden bağımsız olarak sadece bu etkinlik için belirlenmiştir. Rehber öğretmen öğrencilerin bu uygulamayı yapmasını sağlar. Bu görev için örnek bir çözüm aşağıdaki resimlerde verilmiştir.



Resim 24. Örnek Devre Şeması



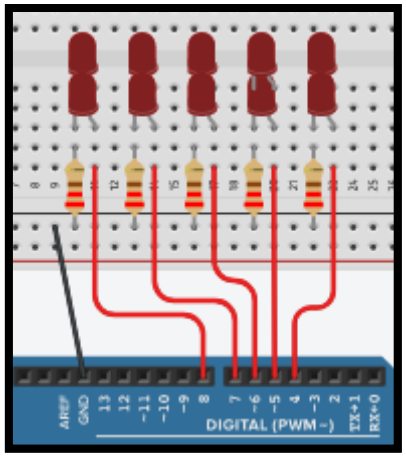
Resim 25. Örnek Program

1.8 Uygula- Araba Yarışı Başlama Işıkları

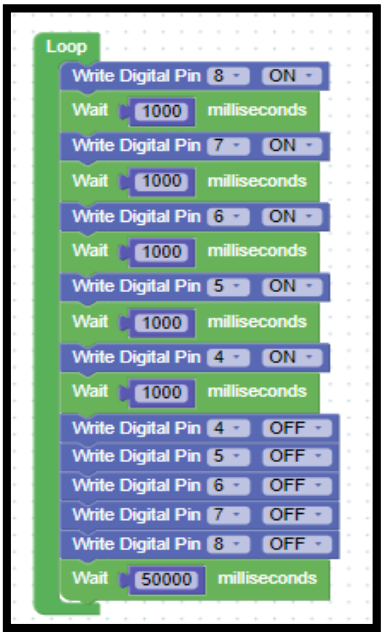
Malzeme Listesi:

Arduino Uno
Breadboard
Bağlantı kabloları
10 tane kırmızı led
10 tane 220 ohm direnç

Bir araba yarışında sürücüler başlama ışıklarını takip ederek yarışa başlarlar. Işıklar iki sıra halinde dizilmiş ve her bir sırada 5'er adet olacak şekilde 10 tane kırmızı led'den oluşur. Etkinlikte ilk olarak iki sıranın birinci led'leri, bir saniye sonra da ikici led'leri yakılır. Bu şekilde birer saniye arayla tüm led'ler yanana kadar örüntü devam ettirilir. Tüm led'ler yandıktan sonra aynı anda tamamı söndürülür. Yarış tüm led'ler sönünce başlar. Tüm aşamalar arasında bir saniye olmasına dikkat edilir. Konuyla ilgili örnek devre şeması ve örnek program aşağıdaki resimde görülmektedir.



Resim 26. Örnek Devre Şeması

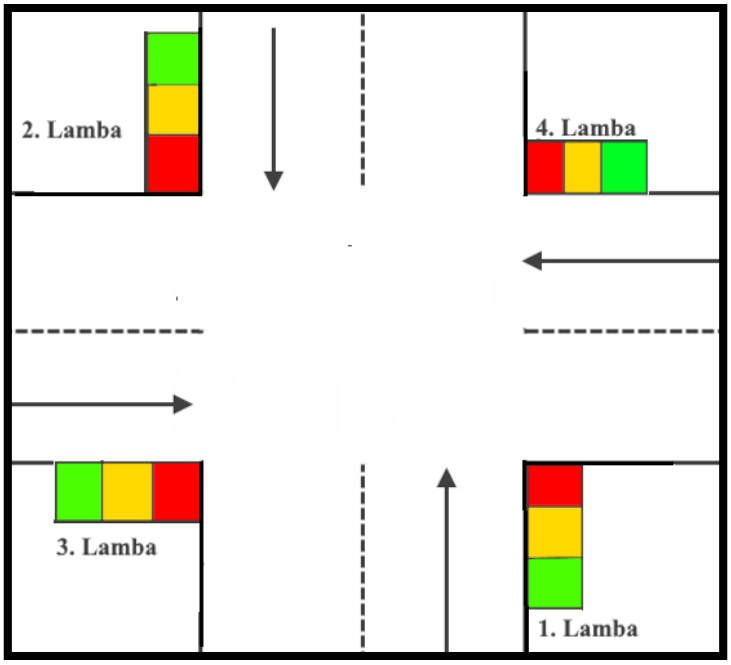


Resim 27. Örnek Program

2 TASARLA ve ÜRET

2.1 Tasarla- Kavşaktaki Trafik Lambaları

Bu etkinlikte amaç bir kavşaktaki dört farklı trafik lambasının birbirleriyle uyumlu bir şekilde çalışmasını sağlamaktır. Aşağıda resimde örnek bir kavşak gösterilmiştir.



Resim 28. Trafik Kavşağı

Öncelikle birinci lambadaki araçlar geçmelidir. Birinci lambadaki araçlar hareket halindeyken diğer lambalardaki araçlar durmalıdır. Daha sonra ikinci lambadaki araçlar geçmelidir. İkinci lambadaki araçlar hareket halindeyken diğer lambalardaki araçlar durmalıdır. Ardından üçüncü lambadaki araçlar geçmelidir. Üçüncü lambadaki araçlar hareket halindeyken diğer lambalardaki araçlar durmalıdır. Son olarak dördüncü lambadaki araçlar geçmelidir. Dördüncü lambadaki araçlar hareket halindeyken diğer lambalardaki araçlar durmalıdır. Öğrenciler "breadboard" kullanarak kavşaktaki trafik ışıklarını modellemelidirler. Gerektiğinde iki "breadboard" birlikte kullanılabilir.

Kavşaktaki trafik lambaları etkinliğini tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir. Aşağıdaki örnek süreç kesinlikle öğrencilere gösterilmemelidir veya başka bir örnek de sınıfça birlikte yapılmamalıdır. Rehber öğretmen sınıfta genel bir sorun görürse sınıfa açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Tanımlama: Öğrencilerin öncelikle kavşaktaki trafik lambaları etkinliğinin neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrenciler öncelikle gerekli işlemleri detaylı olarak maddeler hâlinde yazmalıdır.

Örnek:

i. Dört tane sarı, kırmızı ve yeşil led setinden oluşan devre tanımlanacak.

ii. Herhangi bir sette yeşil ışık yanıyorsa diğer setlerdeki kırmızı led'ler yanacak.

iii. Setteki yeşil led dört saniye yandıktan sonra sarı led yanacak. Sarı led'in yanmasıyla yeşil led kapatılacak.

iv. Sarı led iki saniye yandıktan sonra kırmızı led yanacak.

v. Kırmızı led yanarken, aynı anda sarı led kapatılacak. Aynı anda bir sonraki set'te sarı led yanacak, sarı led ışığı yanan setteki kırmızı led kapatılacak.

vi. Sarı led iki saniye sonra kapatılacak ve setteki yeşil led yanacak.

vii. iii, iv, v ve vi adımları tekrarlanacak.

Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

i. Devre için dört tane sarı, kırmızı ve yeşil led'den oluşan set breadboard'a yerleştirilir.

ii. Her bir led (direnç de eklenmelidir) Arduino'ya bağlanır ve devreleri tamamlanır (akım girişi - yani pin girişi- ve GND bağlantısı yapılır).

iii. Led setleri içerisinden biri aktif set olarak seçilir.

iv. Loop içerisinde aktif setteki yeşil led'in bağlı olduğu pin ON konumuna getirilir. Aynı zamanda diğer setlerdeki bütün kırmızı led'lerin bağlı olduğu pinler OFF konumuna getirilir.

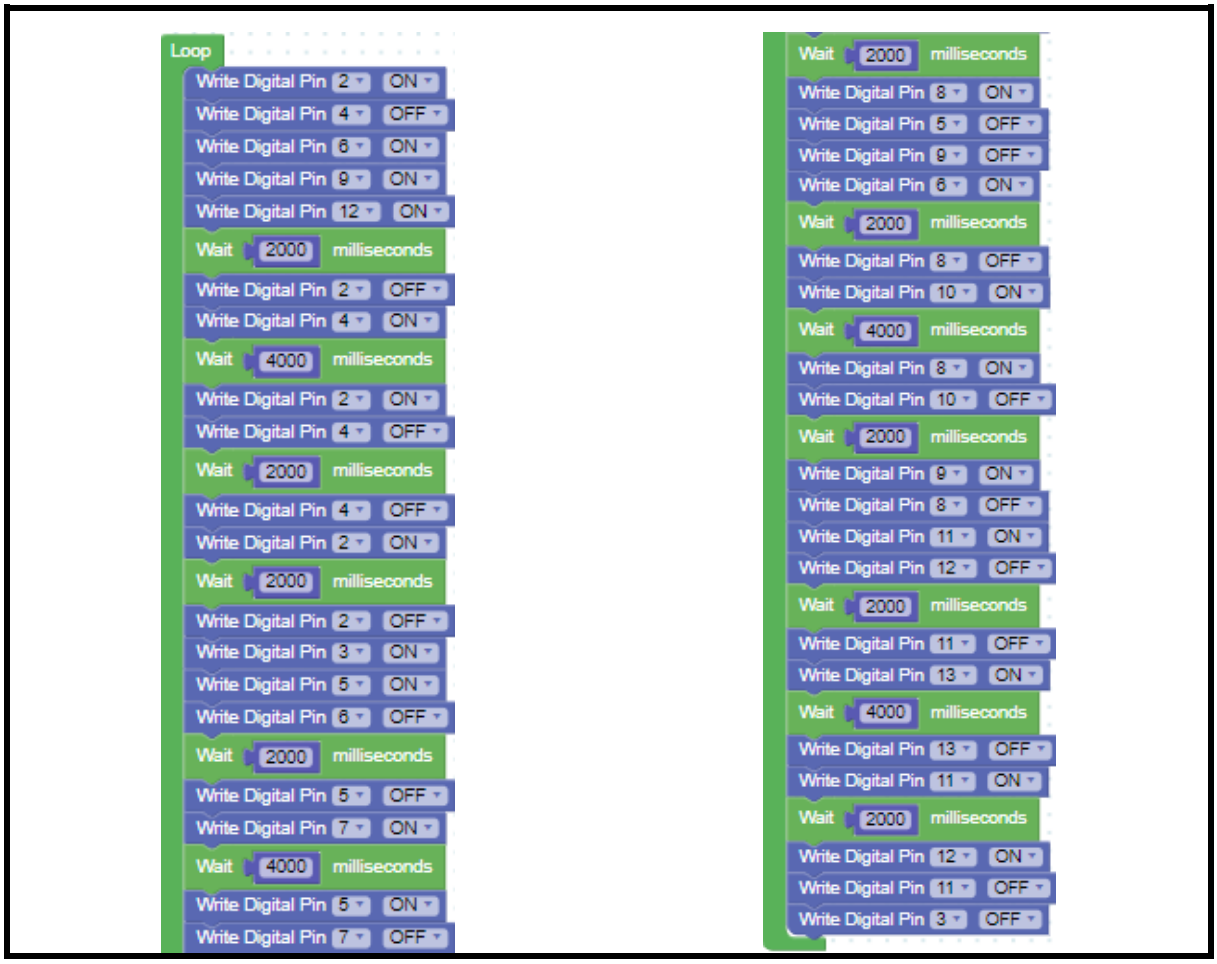
v. Dört saniye geçtikten sonra aktif setteki sarı led'in bağlı olduğu pin ON konumuna getirilir, yeşil led'in bağlı olduğu pin OFF konumuna getirilir.

vi. İki saniye sonra aktif setteki sarı led'in bağlı olduğu pin OFF konumuna getirilir ve kırmızı led'in bağlı olduğu pin ON konumuna getirilir. Bir sonraki set aktif set olarak belirlenir. Aktif sette sarı led'in bağlı olduğu pin ON konumuna getirilir ve bu setteki kırmızı led'in bağlı olduğu pin OFF konumuna getirilir.

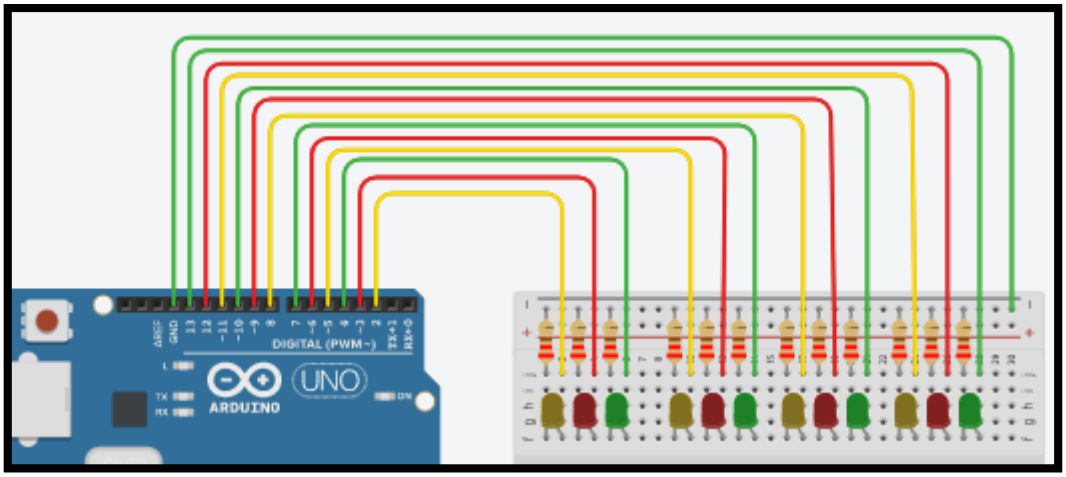
vii. İki saniye sonra iv adımla birlikte aktif setteki sarı led'in bağlı olduğu pin OFF konumuna getirilir ve diğer maddeler yapılmaya devam edilir.

2.1 Üret- Kavşaktaki Trafik Lambaları

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve örnek programa benzer bir içerik hazırlamaları beklenir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.



Resim 29. Örnek Program

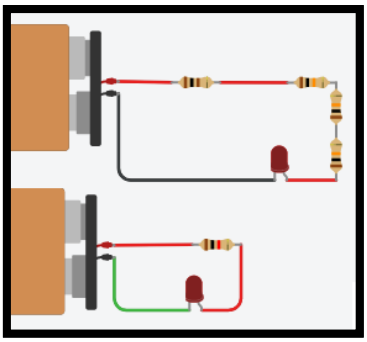


Resim 30. Örnek Devre Şeması

3- DEĞERLENDİR

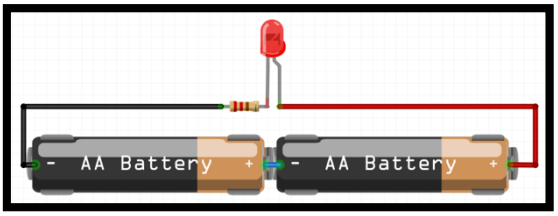
A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Gerilimin sabit olduğu bir devrede var olan direncin yanına yeni direnç/dirençler yan yana gelecek şekilde bağlanırsa akımda ne gibi değişiklikler olabilir? (**Dikkat:** Bu soruda rehber öğretmen dirençlerin aşağıdaki resimde görüldüğü gibi seri bağlı olduğunu vurgulamak için seri bağlı dirençleri tahtaya yansıtmalıdır/çizmelidir. Rehber öğretmen paralel bağlı dirençler konusunu tartışma dışı bırakmalıdır).



Resim 31. Seri Bağlı Dirençler

2. Aşağıdaki resimde görülen devrede gerilim miktarı nedir? Sebepleriyle açıklayınız.



Resim 32. Devre Şeması - Seri Bağlı Piller

3. Bu bölümde öğrendiğiniz devre elemanlarının günlük yaşam içindeki kullanımları nelerdir? Örneklerle açıklayınız.

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Verilen problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Akım, gerilim ve direnç ilişkisini anlamlandırırken zorlandınız mı?
    - Program ve mikrodenetleyici kullanarak devre elemanlarını yönetmede zorluklar yaşadınız mı?
    - Breadboard'da devre oluştururken zorlandınız mı?

- Kavşaktaki Trafik Lambaları etkinliğinde tasarım ve üretim aşamasının hangisinde daha çok zorlandınız?

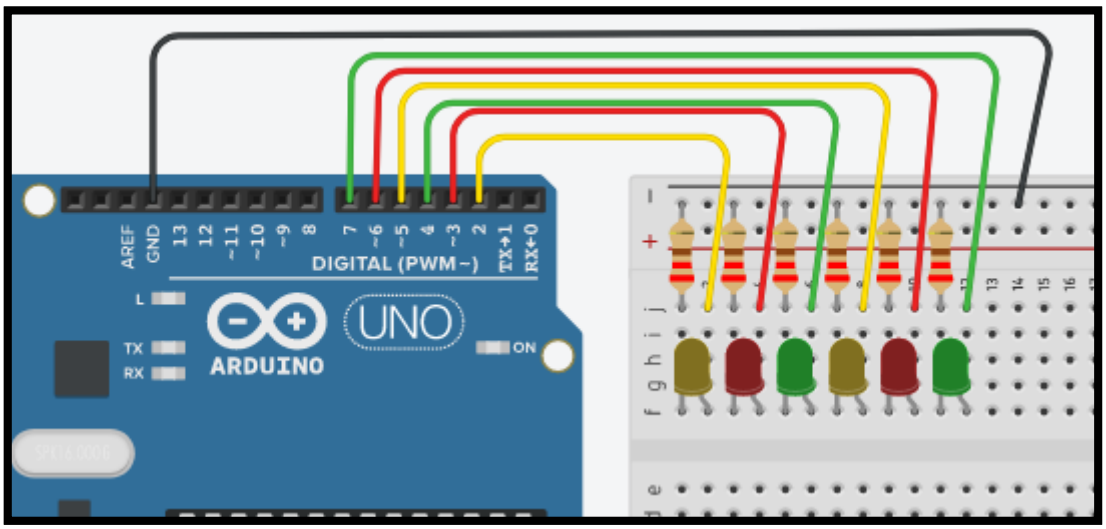
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

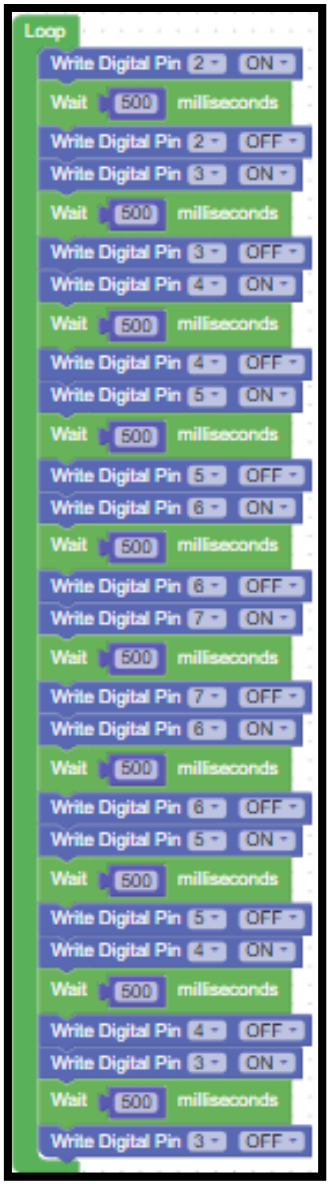
4.1 Soldan-Sağa ve Sağdan-Sola Hareket Eden Led Uygulaması

Bu etkinlikteki amaç led'leri kullanarak ışığın soldan sağa ve sağdan sola akan bir şekilde hareket ettiği hissini uyandırmaktır. Bunun için fiziksel olarak yan yana bulunan led'ler sırayla açılıp kapatılmalıdır. Ardışık led'lerdeki açılıp kapanma göz tarafından ışığın hareketi olarak algılanır. Işıktaki hareket birinci led'den başlayarak son led'e kadar gitmeli, oradan da geriye ilk led'e dönmelidir. Bu şekilde soldan sağa ve sağdan sola hareket işlemi sürekli devam etmelidir. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve programa benzer bir içerik hazırlamaları beklenir. Aynı devre, başlangıçta bütün led'ler açılarak sonrasında sırasıyla her bir led kapatılıp açılarak da programlanabilir.



Resim 33. Örnek Devre Şeması

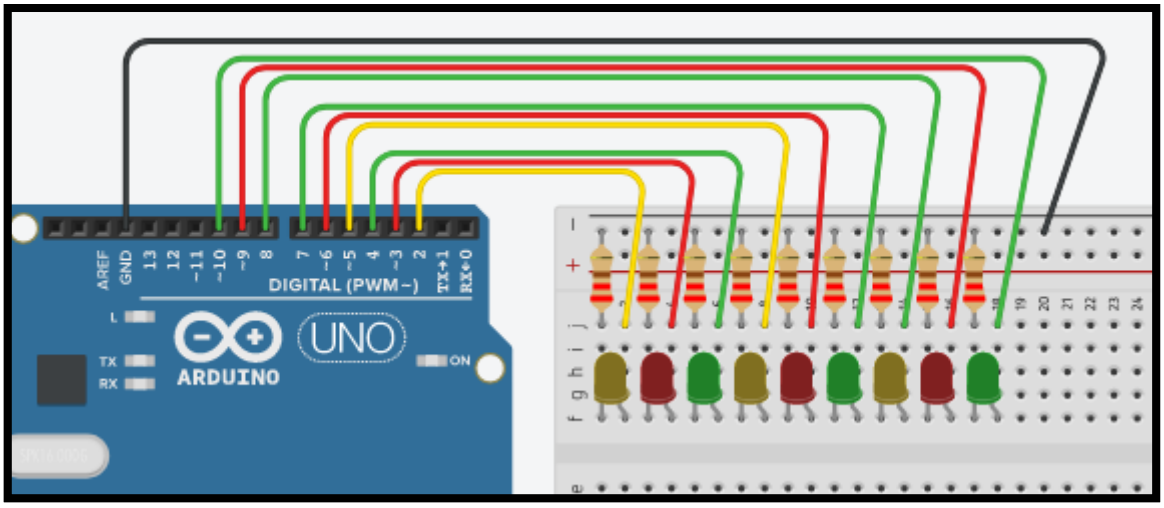


Resim 34. Örnek Program

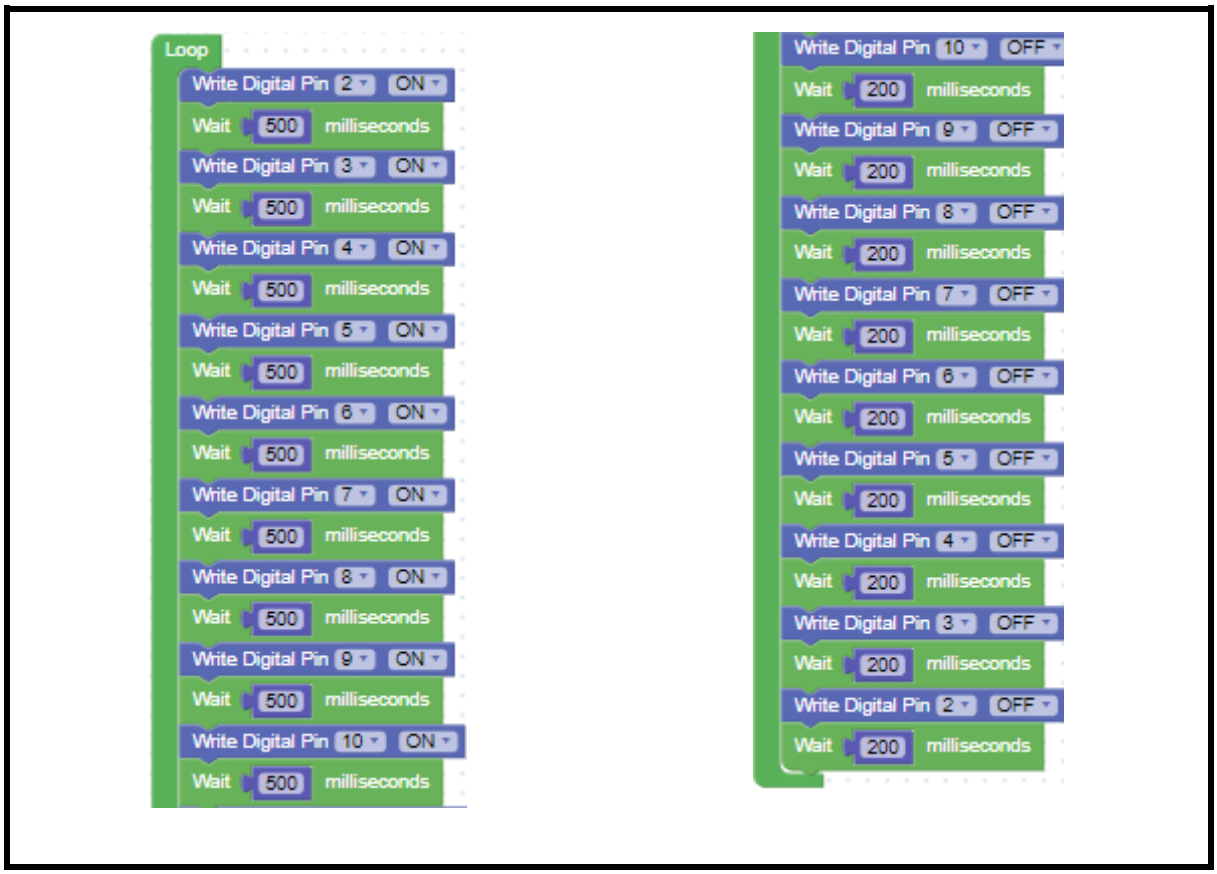
4.2 Müzik ile Uyumlu Hareket Eden Led Uygulaması

Bu etkinlikteki amaç müzik ritmine uygun bir örüntü bulup bu örüntüye bağlı olarak led hareketi gerçekleştirmektir. Etkinlik için öğrenciler bir şarkı seçecektir ve bu şarkı ile uyumlu olarak hareket eden led uygulaması yapacaklardır. Şarkı bilgisayar üzerinde çalıştırılacak Arduino ile yapılan led uygulaması ise bu şarkı ile uyumlu bir şekilde led hareketlerini sağlayacaktır. Işığın hareketi ve müziğin ritmi uyumlu olmalıdır. Kaç adet led kullanılacağı ve hangi şarkının seçileceği öğrenciye bağlıdır. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve programa benzer bir içerik hazırlamaları beklenir.



Resim 35. Örnek Devre Şeması

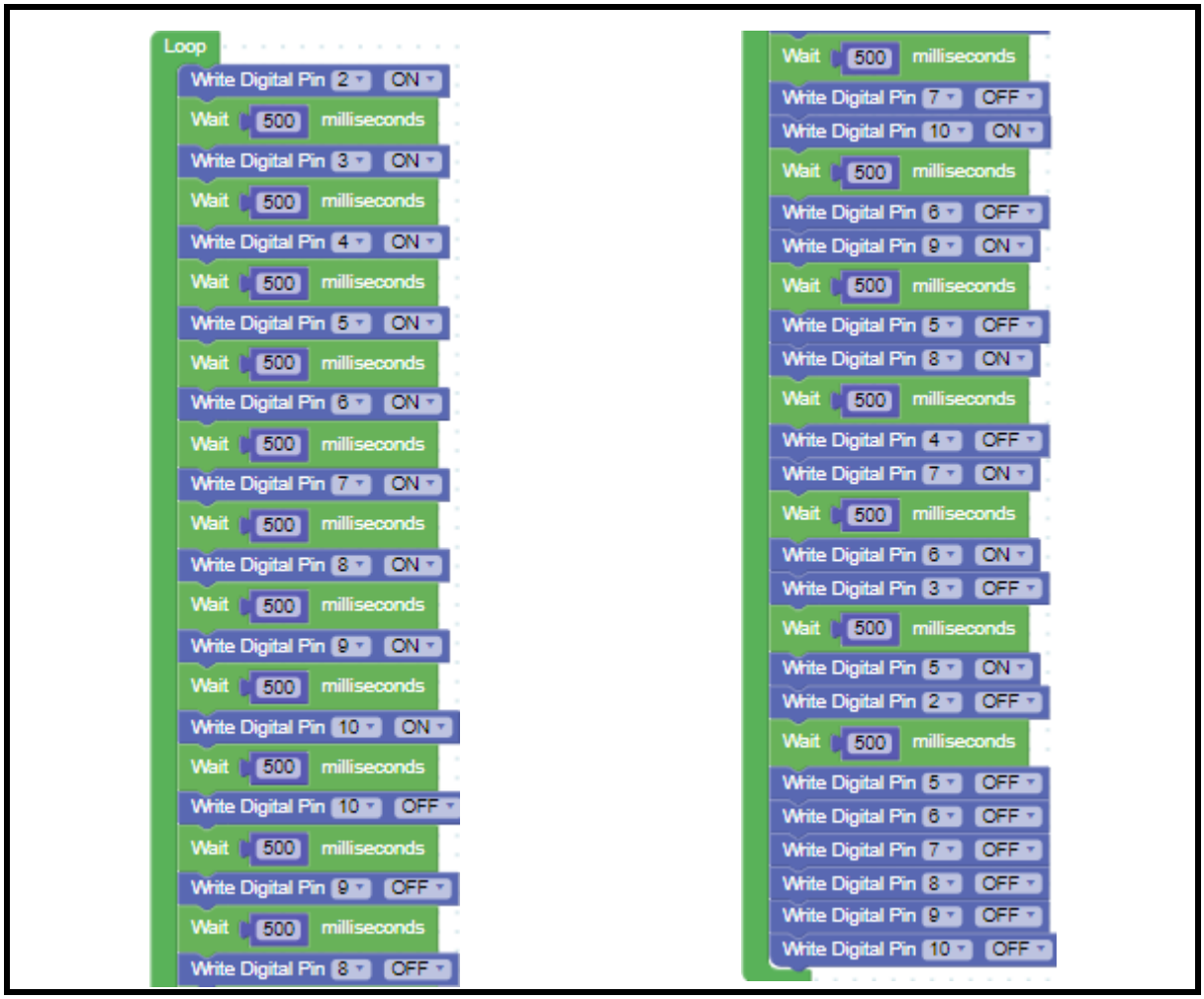


Resim 36. Örnek Program

4.3 Örüntü Tahmini Yarışması

Örüntüler basit anlamıyla belirli nesne veya olayların bir kural doğrultusunda sıralanması şeklinde tanımlanabilir. Bir problemdeki olası örüntüleri tespit etmek çözümü kolaylaştıran bir unsur olabilir. Bu etkinlikte her bir öğrenciye 5'er dakika verilerek yan yana konumlanmış led'leri belirli bir örüntü oluşturacak şekilde yakması istenir. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Daha sonra eşleştirilmiş iki grup birbirlerinin tasarımlarını incelerler. İnceleme tamamlandıktan sonra her grup inceledikleri diğer grubun örüntüsünü oluşturmaya çalışır. Belirlenen süre sonunda gruplar birbirlerinin çözümlerinin doğru olup olmadığını kontrol ederler. Yukarıdaki örnek bir devre için örnek bir örüntü programı aşağıdaki resimde verilmiştir.



Resim 37. Örnek Program

2. Bölüm: Temel Devre Kurulumu, Arduino UNO ve Programlamaya Giriş

Ön bilgi:

- Öğrenciler "gerilim", "akım" ve "direnç" kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında basit uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler "Arduino pinleri"ni, "breadboard" çalışma prensiplerini bilir ve "led/direnç/breadboard/Arduino" bağlantılarını yapabilir

Bölümün Kazanımları:

- Arduino kartın özelliklerini tanımlayabilir.
- Aktif ve pasif "buzzer" kullanarak elektronik devreler kurabilir.
- ArduinoBlocks ile oluşturulan "buzzer" devrelerini programlayabilir.
- ArduinoBlocks programlama ortamında değişken tanımlayıp kullanabilir.
- Lehim kullanarak led küpü oluşturabilir.
- ArduinoBlocks ortamında döngü kullanabilir.
- ArduinoBlocks ortamında fonksiyonlar kullanabilir.
- Oluşturulan led küpünü "ArduinoBlocks" ortamında döngü ve değişkenler kullanarak programlayabilir.

Bölümün Amacı

Bu bölümün ilk amacı, öğrencilerin "Arduino Uno" kartını tanıyarak, "buzzer" bağlantılarını yapmaları ve "ArduinoBlocks" ortamında programlayabilmelerini sağlamaktır. Bölümün diğer amacı ise, öğrencilerin lehim kullanmasını öğrenerek "led" küpü oluşturmalarını sağlamak, "led" küpünün Arduino kartıyla birlikte devresini oluşturmak ve "ArduinoBlocks" ortamında "led" küpünün programlanmasını gerçekleştirmektir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Direnç (3 adet 100 ohm, 2 adet 220 ohm, 3 adet 560 ohm, 3 adet 5 k ohm), 3m kalın elektrik kablosu, 30 adet beyaz led, erkek-dişi bağlantı kabloları, kaide için saydam materyaller (4 tane 10 cm, 4 tane 13 cm), Arduino Uno, aktif & pasif buzzer, breadboard, delikli pertinaks, lehim teli, havya, pasta. (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Arduino kartının, değişkenlerin, aktif ve pasif buzzer'ın tanımlanması ve kullanılması. Led küpünün lehim işlemlerinin yapılması, devrelerinin oluşturulması, döngülerin ve fonksiyonların anlatılması ve ArduinoBlocks ortamında led küpünün döngü ve fonksiyon kullanılarak programlanması.

Uygula: Arduino kart ve buzzer kullanılarak çeşitli melodilerin çalınması. Led küpünün lehim kullanılarak oluşturulması, led küpü devrelerinin oluşturulması ve led küpüne döngüler/fonksiyonlar kullanılarak farklı ışık efektlerinin uygulanması.

Tasarla: Bilgisayardan oynatılan bir şarkıyla uyumlu yanan led küpü programının tasarımının yapılması. Buzzer ile çalınan şarkıyla uyumlu yanan led küpü programının tasarımının yapılması.

Üret: Bilgisayardan oynatılan bir şarkıyla uyumlu yanan led küpünün programlanması. Buzzer ile çalınan şarkıyla uyumlu yanan led küpünün programlanması.

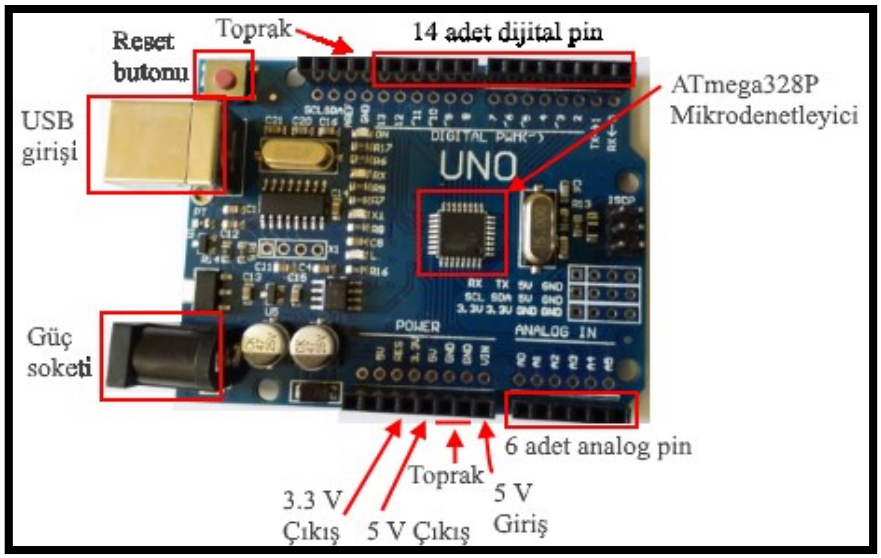
Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1 GÖZLE VE UYGULA

1.1 Gözle- Arduino Uno'yu Tanıyorum

Arduino Uno temelde bir mikrodenetleyici barındıran elektronik bir karttır. Üzerinde kontrol elemanlarını bağlamak için "pinler" ve "USB" bağlantısı da bulunmaktadır. Robotların programlanması gibi mikrodenetleyiciler de programlanabilir. Arduino'da robotlardan farklı olarak birçok elektronik donanım/devre elemanı Arduino'nun girdi ve çıktı pinleri kullanılarak devreye takılır ve yazılan bir program ile etkileşime geçebilir. Mikrodenetleyiciler ile akıllı evlerde kullanılan sistemlerden, robotlara; hırsız alarmından drone'lara kadar birçok tasarım yapılabilir.

Aşağıdaki resimde görüldüğü gibi Arduino UNO ATmega328P mikrodenetleyicisine sahiptir. Arduino için yapılan programlar "ATmega328P" içerisinde saklanır ve burada çalıştırılır. USB girişi ve Güç soketi aracılığı ile Arduino kartına elektrik sağlanabilir. Ayrıca "5V" giriş pini kullanılarak da Arduino karta elektrik sağlanabilir. Arduino kart üzerinde "5V" ve "3.3V" çıkış alınabilecek pinler bulunmaktadır. Bu pinler "Toprak" pinleriyle birlikte kullanılarak devre elemanlarına elektrik sağlar. Arduino Uno kartın üzerinde 14 adet "dijital pin" bulunmaktadır. Bu pinler ile hem girdi hem de çıktı işlemleri yapılabilir. Bu pinler aracılığıyla dijital devre elemanları Arduino karta takılıp mikrodenetleyici ile uyumlu bir şekilde çalıştırılabilir. Bunun yanında Arduino Uno kartı üzerinde 6 adet "analog pin" bulunmaktadır. Bu pinler analog devre elemanları ile iletişim için kullanılır. Aynı zamanda bu pinler dijital girdi/çıktı için de kullanılabilir. Bu noktada analog dijital farkı çok önemli değildir. İleriki derslerde bunlardan bahsedilecektir. Şimdilik bunların varlığından haberdar olup temel görevlerini bilmek yeterlidir. Son olarak Arduino Uno kartı üzerinde bir adet "Reset butonu" bulunmaktadır. Bu butona basılarak mikrodenetleyicideki program yeniden başlatılabilir. Aslında Arduino Uno üzerinde anlatılanların dışında başka elektronik malzemeler de bulunur fakat bu ders kapsamında şimdilik bahsedilen bilgiler yeterlidir.



Resim 38. Arduino Kartı

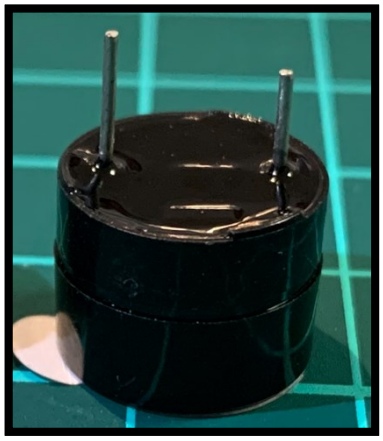
1.2 Gözle ve Uygula- Buzzer ile Melodiler

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet aktif buzzer
1 adet pasif buzzer
1 adet 100 Ohm direnç

Daha önce ışıklı devre elemanları işlenmişti. Işıklı devre elemanları gibi sesli devre elemanları da bulunur. Bunlar ses üretebilir. Buzzer da ses üretebilen bir devre elemanıdır. Günlük yaşamda şarkı ve ses kaydı gibi ses unsurlarını oynatmak için hoparlörler sıklıkla kullanılır. Buzzerlar mini hoparlörlere benzetilebilir. Buzzerlar farklı ses sinyalleri üretebilir. Fakat hoparlör kadar net ses üretemez. Sadece belirli notalarda sesler üretebilir. Örneğin farklı notalarda bip sesi üretebilir. Alarm ve fırın gibi günlük hayatta kullanılan birçok cihaz içerisinde kullanılabilir. Piyasada aktif ve pasif olmak üzere iki çeşit buzzer bulunmaktadır. Bunlar birbirine çok benzemektedir. Her iki buzzer çeşidinin de iki bacağı bulunur. Bu bacaklardan büyük olanı "+" küçük olanı ise "–" bacaktır. Aşağıda resimlerde pasif ve aktif buzzer görselleri bulunmaktadır. Pasif buzzerın alt kısmı açık iken, aktif buzzerın alt kısmı siyah renkte bir kaplama ile kapatılmıştır. Setlerdeki aktif ve pasif buzzerları ayırt etmek için bu bilgi kullanılabilir.

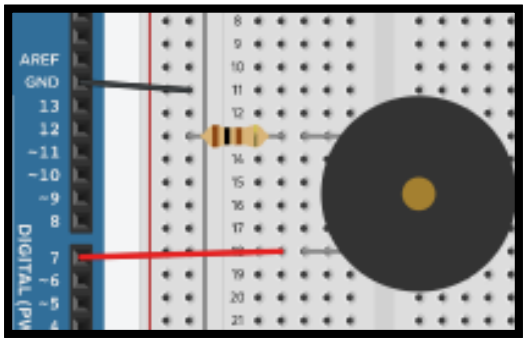
Resim 39. Pasif Buzzer



Resim 40. Aktif Buzzer

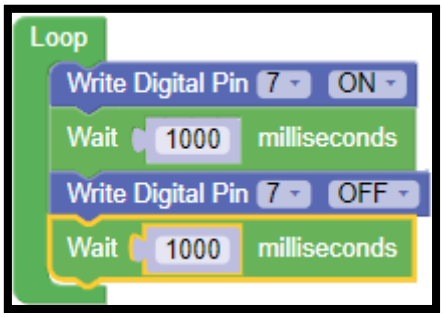
Aktif buzzer'a çalışma aralığı içerisinde bir elektrik akımı uygulandığında bip sesi üretir. Pasif buzzera doğru akım uygulandığında istenilen ses elde edilemez. Pasif buzzer'dan istenilen sesi elde etmek için ArduinoBlocks'un gerekli komutlarını kullanmak gerekir. Bu komutlar bazı aktif buzzer'lar ile de kullanılabilir fakat her zaman istenilen sesi üretmeyebilir. Bu yüzden aktif ve pasif buzzer'lar farklı şekilde kodlanır.

Aktif ve pasif buzzer'ın devre şeması veya bağlantıları değişmez. İkisi de aynı şekilde yapılır. Bu yüzden aşağıda bir devre şeması verilecektir fakat bu her iki buzzer çeşidi için de uygulanabilir. Aşağıdaki resimde görüldüğü gibi buzzer'ın artı bacağı Arduino Uno kartının yedinci dijital pinine bağlanmıştır. Buzzer'ın diğer bacağı ise "100 ohm" bir direnç üzerinden Arduino Uno kartının GND pinine bağlanmıştır.



Resim 41. Örnek Devre Şeması

Aşağıda resimde aktif buzzer için yazılmış bir kod bulunmaktadır. Bu kod sayesinde buzzer bir saniye "bip" sesi çıkarır, bir saniye bekler ve bu işlemi sürekli yapar.

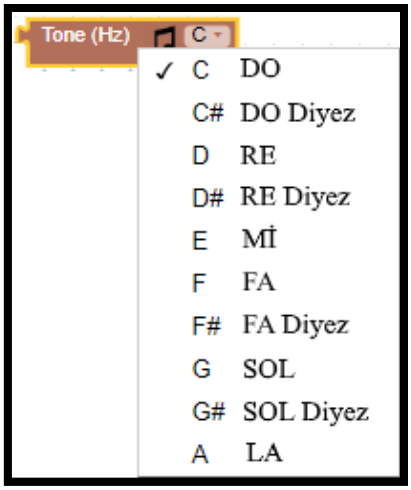


Resim 42. Örnek Program

Pasif buzzer ile (buzzer'ın desteklediği ölçüde) istenilen nota veya istenilen frekanstaki ses çalınabilir. Bunun için "Actuators" başlığındaki "Buzzer" ve "Tone" blokları kullanılır. Aşağıdaki resimde bu komutlar görülmektedir. Buzzer komutu ile istenilen Arduino dijital pinine bağlanmış (görselde ikinci pin) buzzer istenilen sürede (görselde resimde 500 mili saniye) istenilen frekansta (görselde resimde 1000 Hz) ses üretir.

Resim 43. Buzzer ve Tone Blokları

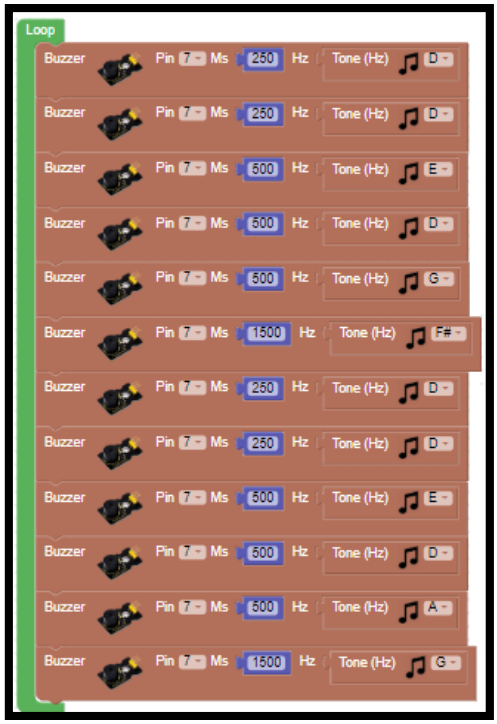
Eğer buzzer'dan bir nota çalınmak isteniyorsa "Tone" bloğu da kullanılabilir. "Tone" bloğu kullanılarak C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La) notaları çalınabilir. Bunların yanında ara notalar olan diyezler de çalınabilir. Nota seçimi aşağıdaki resimde görülmektedir.



Resim 44. Nota Seçimi

İstenilen notanın çalınabilmesi için "Tone" bloğu "Buzzer" bloğundaki frekans/nota kısmına yerleştirilir. Aşağıdaki resimde bulunan komut ile DO notası yarım saniyeliğine çalınır.

Resim 45. İstenen Notanın çalınması

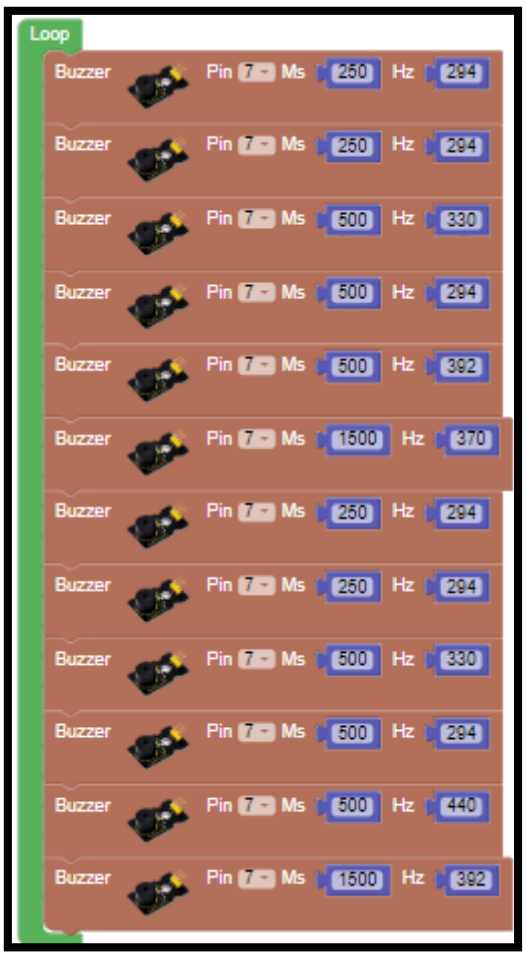
Aşağıdaki resimde görülen program "Mutlu Yıllar Sana" şarkısının bir kısmını oynatmaktadır. Rehber öğretmen bu programı yazıp, Arduino'da oynatarak öğrencilerin dinlemesini sağlar. Ardından rehber öğretmen bu programın aynısını öğrencilerin yapmasını ister.

Resim 46. Örnek Program

Sadece Tone bloğu ile notalar çalınmaz. Notaların frekans değerleri kullanılarak da notalar çalınabilir. Aşağıdaki tabloda notaların Hertz cinsinden frekans karşılıkları verilmiştir.

Notaların Frekans (Hz) Karşılıkları

DO	RE	Mİ	FA	SOL	LA	Sİ
262	294	330	349	392	440	494

Aşağıdaki resimde görülen programda çalınan "Mutlu Yıllar Sana" parçasının girişini aynı şekilde oynatmaktadır. Rehber öğretmen öğrencilerden bu programı yazmalarını ister. Öğrencilerin bu programı yazmaları için FA Diyez notasının frekansını bilmeleri gerekmektedir. Bu frekans değeri 370 Hz'dir.



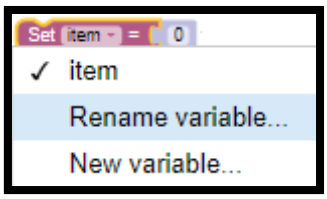
Resim 47. Örnek Program

Notaların frekans karşılıklarını akılda tutmak zordur. Bu yüzden programın başında notaların frekans değerleri değişkenlere bir defa atanırsa programda istenilen yerde frekans değerlerindense değişken isimleri akılda tutulup kod yazılabilir. Örneğin 262 değeri Do değişkenine atanırsa program yazarken

262 yerine Do ismi kullanılabilir. Değişken tanımlamak ve kullanmak için aşağıdaki resimde görüldüğü gibi Variables blok başlığı altındaki "Set item" ve "item" blokları kullanılır.

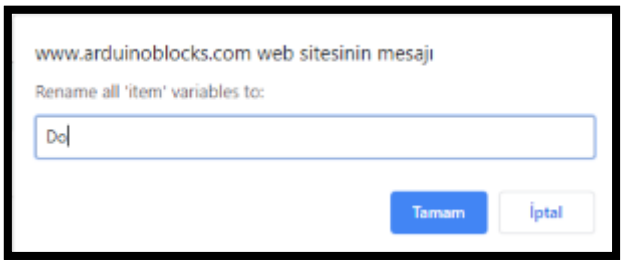
Resim 48. Değişkenler Blok Başlığı

Do değişkenini tanımlamak için aşağıdaki resimde görüldüğü gibi "Set item" bloğunda "item" üzerine tıklanır ve "Rename variable" seçeneği seçilir.



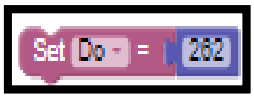
Resim 49. Değişken İsmi Belirleme - Adım 1

Açılan pencereden değişkenin ismi girilir ve Tamam tuşu tıklanır. Burada değişken ismi Do olacağı için aşağıdaki resimde görüldüğü gibi Do girilmiştir.



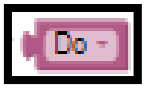
Resim 50. Değişken İsmi Belirleme - Adım 2

Değişken ismi verildikten sonra artık değişkenin ilk değeri atanmalıdır. Do'nun frekans karşılığı 262 olduğu için aşağıdaki resimde görüldüğü gibi eşitliğin sağ tarafında kalan ifade yerine 262 yazılır.



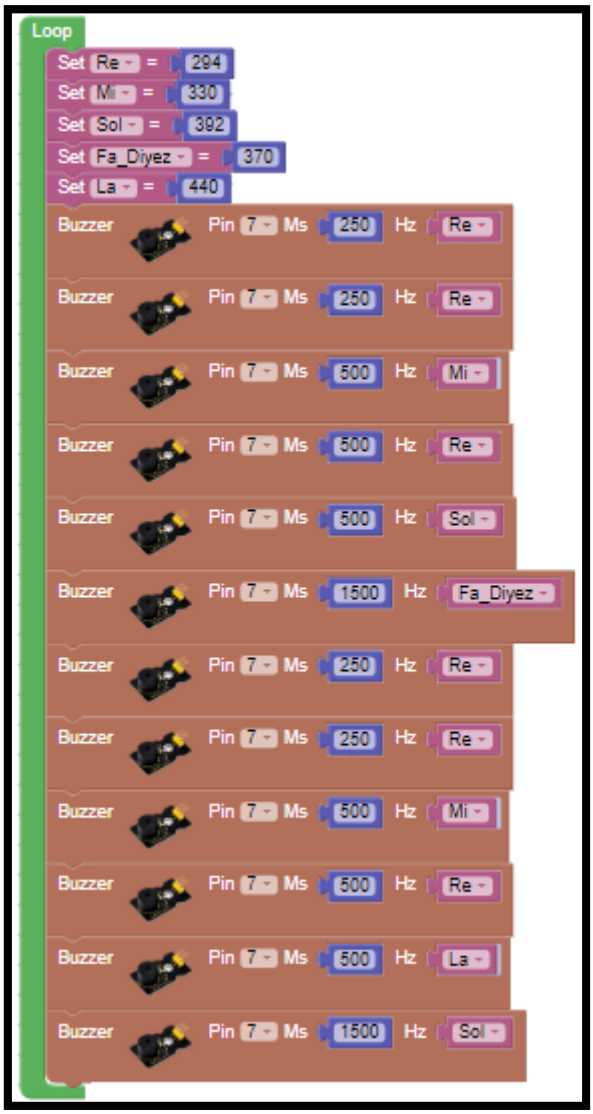
Resim 51. Değer Atama

Do değişkeni bir defa tanımlanıp ilk değeri atandıktan sonra programda kullanılabilir. Bir değişkeni programda kullanabilmek için "Variables" blok başlığı altındaki "item" kullanılır. "item" bloğundaki aşağı ok tıklanarak istenilen değişken program içerisinde kullanılabilir. Aşağıdaki resimde programda kullanılabilecek olan Do değişkeni gösterilmiştir.



Resim 52. Do Değişkeni

Bu şekilde RE, Mİ, SOL, FA Diyez ve LA notaları değişken olarak tanımlandığında program içerisinde bu değişkenler sayısal frekans değerleri yerine kullanılabilir. Aşağıdaki program bu kullanıma örnek olarak verilebilir. Aşağıdaki resimde verilen programda görüldüğü üzere notalar bir kere tanımlanmıştır ve program içerisinde ihtiyaç duyulduğunda kullanılmıştır.



Resim 53. Örnek Program

Dikkat

Bu programda değişkenleri "Loop" içerisinde tanımlamak programlama pratiği açısından en iyi yöntem olmayabilir. Fakat pedagojik açıdan burada tanımlanması dersin amaçları doğrultusunda uygun olacaktır.

1.2 Uygula- Şarkı Çalan Buzzer

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet aktif buzzer
1 adet 100 Ohm direnç

Bu etkinlikteki amaç öğrencilerin basit bir şarkıyı buzzer'dan çaldırmasıdır. Öğrenciler buldukları/bildikleri basit bir şarkının notalarını kullanarak gerekli kodu yazar. Rehber öğretmen öğrencilerin İnternet'ten nota aramasına izin verir. Fakat rehber öğretmen nota okumasını gerektiren veya nota uzunluğu gibi etkinliği yavaşlatıcı durumlarda öğrencilere yardımcı olmalıdır.

1.3 Gözle ve Uygula- Led Küpü Yapıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Direnç (3 adet 100 ohm, 2 adet 220 ohm, 3 adet 560 ohm, 3 adet 5 k ohm)
3m kalın elektrik kablosu
30 adet beyaz led
Kaide için saydam materyaller (4 tane 10 cm, 4 tane 13 cm
Delikli pertinaks
Lehim teli, havya ve pasta

Bu etkinlikte amaç 3x3x3 büyüklüğünde bir led küpü oluşturmaktır. Rehber öğretmen led küpünü sınıfta öğrencilerin karşısında onlara göstererek adım adım oluşturmalıdır. Öğrenciler de rehber öğretmeni takip ederek led küpünü yapmalıdır. Burada öğrenciler led küpünü yaparken fazla vakit kaybetmemelidir. Rehber öğretmen led küpünün hızlı bir şekilde yapılması için öğrencileri yönlendirmelidir.

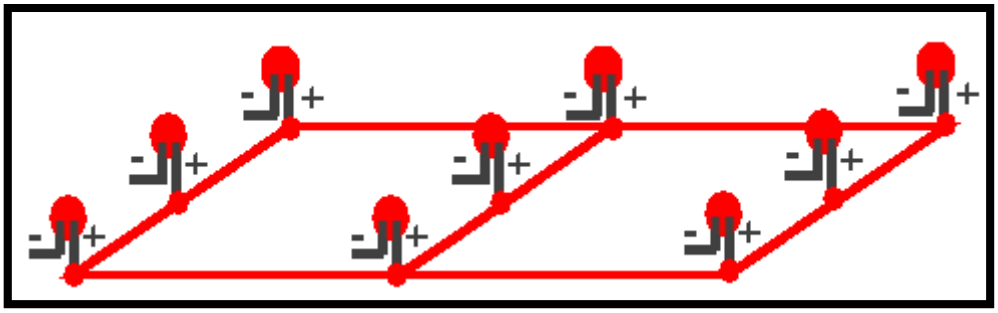
Dikkat

Oluşturulacak led küpü sonraki etkinliklerde yeniden kullanılacaktır. Bu yüzden rehber öğretmen, öğrencilerin led küpünü sağlam bir şekilde yapmasını ve etkinliğin ardından güvenli bir yerde saklamasını sağlamalıdır.

Etkinlikte havya kullanılacağı için rehber öğretmenler herhangi bir yanık olayı ile karşılaşmamak için gerekli özeni göstermelidir

Etkinliğin yapımını kolaylaştırmak için bir kaide seti oluşturulmuştur. Bu kaide seti, led küpü ve diğer devre elemanlarının delikli pertinaks üzerinde birleştirilip etkinlik için gerekli olan devrenin tamamlanması ve sabit kalması için kullanılacaktır. Devrenin son halinin resmi aşağıda verilecektir. Fakat ondan önce devre şeması üzerinde çalışarak devrenin nasıl kurulacağını anlamaları için öğrencilere yardımcı olunmalıdır.

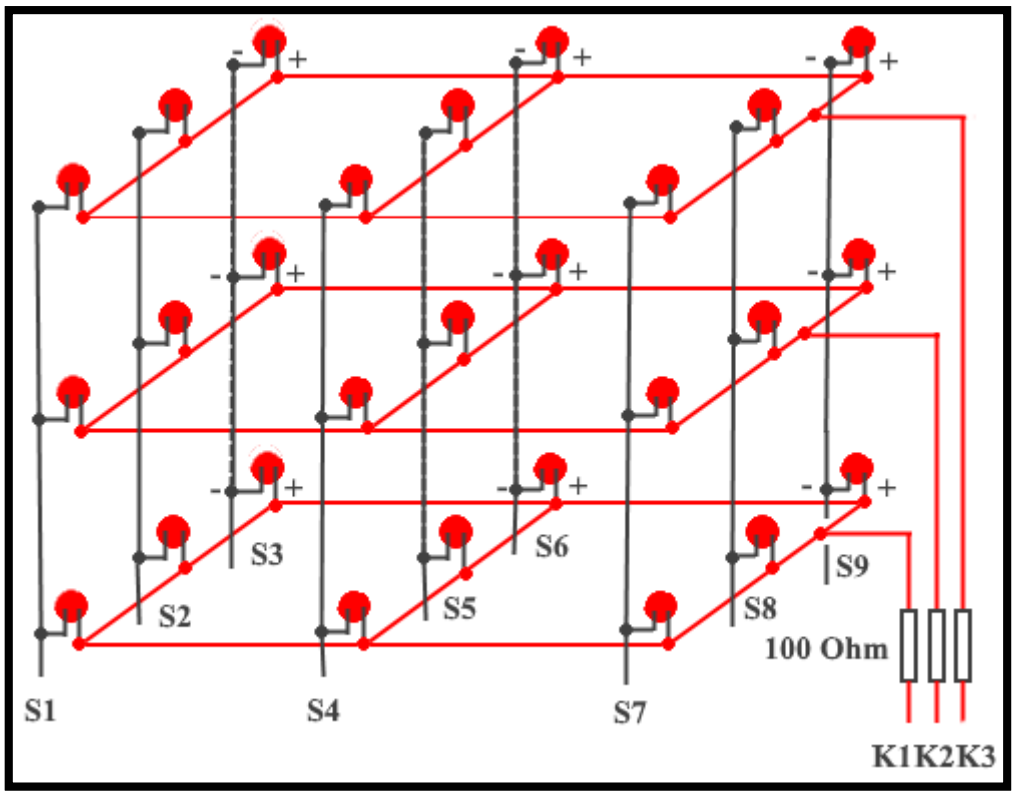
Bu led küpünde 3 kat bulunacaktır. Her bir kata dokuz (3x3) adet led bağlanacaktır. En üst katın bağlantı şeması aşağıdaki resimde verilmiştir. Resimde görüldüğü gibi ledlerin "anot" (+) bacakları iletken bir tel aracılığı ile birbirlerine bağlanmıştır. "Katot" (-) bacakları ise bir alt katmandaki led'ler ile birleştirilmek üzere yamultularak açık bırakılmıştır. Bu bacaklar alt katlardaki led'ler ile birleşerek sütunları oluşturacaktır.



Resim 54. Led Küpü Üst Kat

Bu şekilde üç farklı kat oluşturulacaktır. İkinci katın yapımına geçmeden önce birinci kattaki devre elemanlarında bir sıkıntı olup olmadığı kontrol edilmelidir. Bunun için bir multimetre (Avometre)/güç kaynağı kullanılabilir. Multimetre kısaca akım, gerilim ve direnç gibi değerleri ölçmek için kullanılan bir araçtır. Multimetre direnç ölçme moduna alınır. Multimetrenin "+" bacağı ledlerin "+" bacağının bağlı olduğu tele değdirilir. Multimetrenin "-" bacağı yakılmak istenen led'in - bacağına değdirilir. Ledlerin yanıp yanmadığı kontrol edilir ve tüm ledlerin yandığından emin olunur. Bu test için multimetre yerine bir güç kaynağı da kullanılabilir. Güç kaynağı kullanılacaksa rehber öğretmen öğrencileri test işleminde led'leri yakmamak için uygun bir direnç kullanmaları konusunda uyarır.

Bu şekilde devam edilerek ikinci ve üçüncü katlar da oluşturulur. Oluşturulan katlar aşağıdaki şemada görüldüğü gibi birleştirilir. Led'lerin boşta kalan (-) bacakları bir tel yardımıyla bir sütun oluşturacak şekilde birleştirilir. Her bir kattan çıkan tel "100 ohm" dirençler üzerinden Arduino'nun pinlerine bağlanacaktır. Bu dirençler led'lerin patlamasını önlemek içindir. Aşağıdaki resimde görülen şemada delikli pertinaks bulunmamaktadır fakat bu dirençler gerçek tasarımda delikli pertinaks üzerinde olacaktır.



Resim 55. Led Küpü Şeması

Şemadan da görüleceği gibi led küpünde üç kat (K1, K2, K3) ve dokuz sütun bulunmaktadır (S1, S2, …, S9). Bu sayede her bir led ayrı ayrı yakılabilir. Örneğin birinci sütunun (S1) en üst katında bulunan led'i yakmak için üçüncü kata (K3) 5V (ON) ve birinci sütuna (S1) 0V (OFF) verilmelidir. Rehber öğretmen öğrencilere farklı led'lerin ve led gruplarının nasıl yakılabileceğine örnekler gösterir. Led küpü tamamlandıktan sonra her bir led'in çalışıp çalışmadığı kontrol edilmelidir. Bunun için S1, S2, ..., S9 sütun bacakları ve K1, K2, K3 kat bacakları kullanılmalıdır.

Son olarak sütunlar sırasıyla Arduino'nun 1, 2, …, 9 numaralı dijital pinlerine (S1->1, S2->2, …, S9->9 olacak şekilde) ve katlar ise Arduino'nun A1, A2, A3 pinlerine (K1->A1, K2-> A2, K3->A3 olacak şekilde) bağlanır. Bu bağlantıların yapılmasının ardından artık led küp ile ışık oyunları yapılabilir.

Led küpü için hazırlanan kaide ve delikli pertinaks kullanılarak oluşturulmuş led küpünü inşa etmek için EK-1 (Led Küp Kurulum Yönergesi) kullanılabilir.

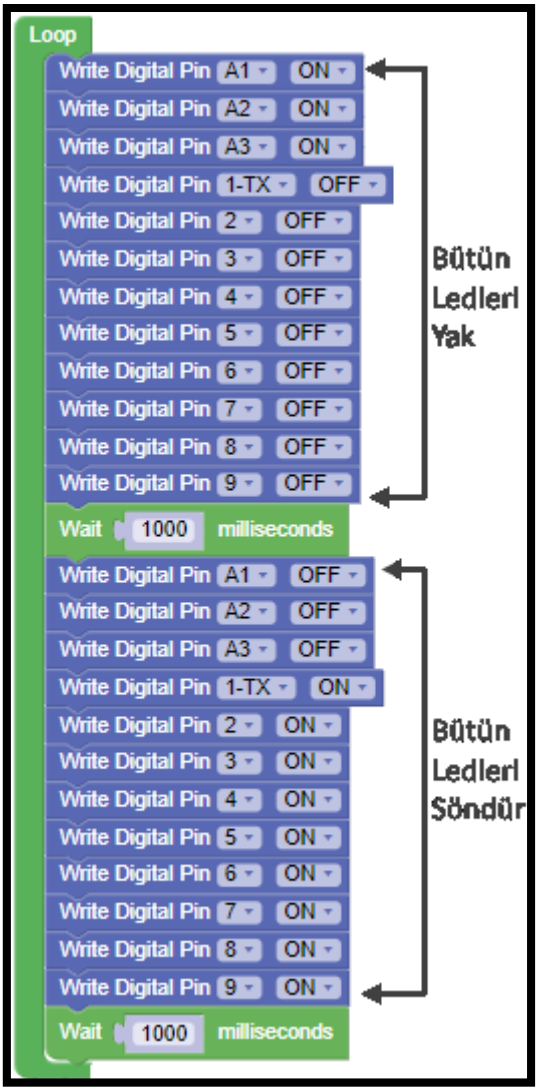
1.4 Gözle ve Uygula- Led Küpünü Kodluyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Led küpü

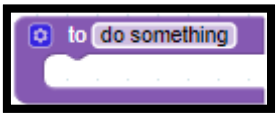
Bu etkinlikte 3x3x3 boyutunda bir led küpü kullanılarak ışık oyunları yapılacaktır. Led küpünün bağlantısı sütunlar sırasıyla Arduino'nun 1, 2, …, 9 numaralı dijital pinlerine karşılık gelecek şekilde (S1->1, S2->2, …, S9->9) ve katlar ise Arduino'nun A1, A2, A3 pinlerine karşılık gelecek şekilde (K1->A1, K2->, A3->K3) yapılmalıdır.

İlk ışık oyunu bütün ledlerin yakılıp söndürülmesi ile oluşturulacaktır. Bunun için önce katların hepsine ON (5V) ve sütunların hepsine OFF (0V) verilerek bütün led'ler yakılır. Bilindiği gibi ledlerin anot (+) bacağına "5V" ve katot (-) bacağına "0V" verildiğinde led yanacaktır. Bir saniye beklenir. Bunun ardından katların hepsine OFF (0V) ve sütunların hepsine ON (5V) verilerek bütün led'ler söndürülür. Burada led'leri yakmak için onların + ve - bacaklarına verilen değerler ters bir şekilde led bacaklarına uygulanmıştır (0V->+ bacak, 5V-> - bacak) değerlerin tersi led bacaklarına verilmiştir. Aşağıdaki resimde örnek program görülmektedir.



Resim 56. Örnek Program

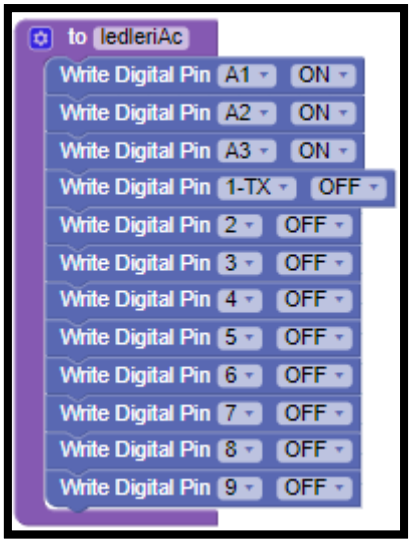
Görüldüğü gibi led'lerin tamamını yakıp söndürmek için uzun uzun kod yazmak gerekmektedir. "Loop" bloğunun içerisinde yeniden led'lerin tamamı yakılmak istenseydi 12 adet kod bloğu yazmak gerekecekti. Yine benzer şekilde led'lerin tamamının kapatılması istenseydi 12 adet kod bloğu daha yazılması gerekecekti. Bunun yerine fonksiyonlar kullanılabilirdi. Fonksiyonlar kabaca sık kullanılan komutların bir araya getirilerek isimlendirilmesidir. "ledleriKapat" ve "ledleriAc" isminde iki adet fonksiyon yazarak her defasında bu kodları tekrar tekrar yazmaktan kurtulmak mümkündür. Bunun yerine fonksiyonlar gerekli yerde çağırılır. Fonksiyon yazmak için "Functions" blok grubu altındaki "to do something" bloğu kullanılacaktır. Bu blok aşağıda resimde görülmektedir.



Resim 57. Fonksiyon Bloğu

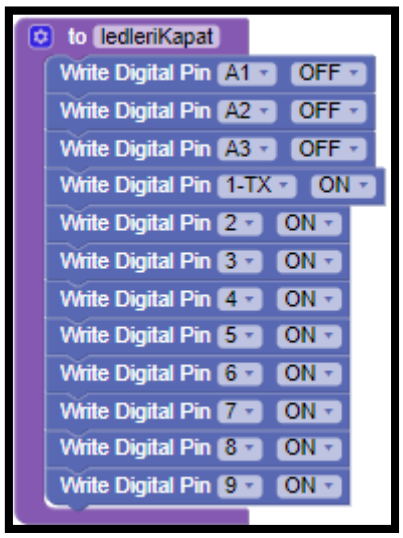
Bu bir fonksiyon oluşturma bloğudur. "do something" fonksiyonun ön tanımlı ismidir ve üzerine tıklanarak fonksiyon ismi değiştirilebilir. "do something" yazılı alana tıklanır ve bu alana "ledleriAc" yazılır. Bu şekilde oluşturulan fonksiyonun ismi tanımlanmış olur. Daha sonra bloğun içerisine gerekli

komutlar yerleştirilir. Aşağıdaki resimde "ledleriAc" fonksiyonunun isminin tanımlanmış ve komutlarının girilmiş hali görülmektedir.



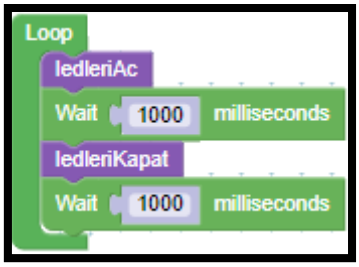
Resim 58. Ledleri Açma Fonksiyonu

Yine benzer şekilde aşağıdaki resimde görüldüğü gibi "ledleriKapat" isimli fonksiyon oluşturulur.



Resim 59. Ledleri Kapatma Fonksiyonu

Artık bu iki fonksiyon programda istenilen yerden çağrılıp kullanılabilir. Bu fonksiyonları çağırıp kullanmak için "Functions" blok grubu içerisinde fonksiyonların adı kullanılarak oluşturulmuş kod blokları istenilen yere sürüklenerek bırakılır. Aşağıdaki resimde görülen program da led'leri açıp kapama görevini yerine getirmektedir.



Resim 60. Fonksiyonların Programda Kullanımı

1.5 Gözle- Led Küpünü Farklı Aralıklarla Yakıp Söndürüyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Led küpü

Bundan önceki iki etkinlikte led küpündeki led'ler birer saniye aralıkla yakıp söndürülmüştür. Bu etkinlikte ise led küpünün led'lerinin yanıp sönmesi arasındaki süre bir saniye, iki saniye, üç saniye, dört saniye; bir saniye, iki saniye, üç saniye dört saniye; … aralıklarla tekrar edecek şekilde ayarlanmalıdır. Bunun için değişken ve döngü isimli iki yeni kavramı bilmek gerekmektedir.

Döngüler:

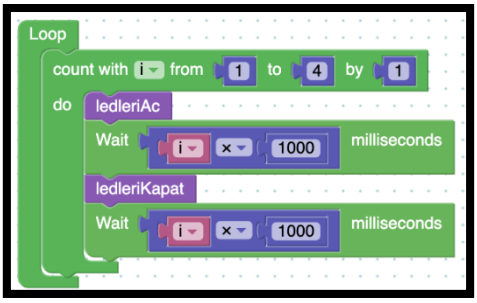
Programlamada tekrarlanan görevleri yerine getirmek için döngüler kullanılabilir. Farklı tiplerde döngüler bulunmaktadır. Bu etkinlikte "Control" blok grubu içerisindeki aşağıdaki resimde görülen kod bloğu kullanılacaktır.

Resim 61. For Tipi Döngü

Bu kod bloğu ile "i" isminde bir değişken oluşturulur ve bu değişkene ilk değer olarak 1 değeri atanır ve bu "i" değeri her defasında 1 artırılarak 10 değerine kadar bloğun içerisine konulacak kodları tekrar eder. Sonuç olarak bu kod bloğunun içerisine konulacak olan kod blokları 10 defa tekrar edilmiş olur. Bu döngünün grafiksel gösterimi aşağıdaki resimde görüldüğü gibidir.

Resim 62. Döngü Açıklaması

Bu etkinlikte led küpünün ledlerinin yanıp sönmesi arasındaki süre bir saniye, iki saniye, üç saniye, dört saniye; bir saniye, iki saniye, üç saniye dört saniye; … aralıklarla tekrar edecek şekilde ayarlanması istenmiştir. Bunun için aşağıdaki resimde görülen program tahtaya yansıtılır ve öğrencilere adım adım anlatılır.



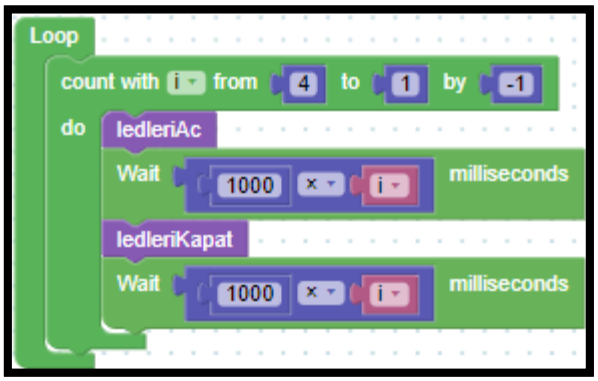
Resim 63. Örnek Program

1.5 Uygula- Led Küpünü Farklı Aralıklarla Yakıp Söndürüyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Led küpü

Bir önceki etkinlikten farklı olarak bu etkinlikte ledlerin yanıp sönmeleri arasındaki süre dört saniye, üç saniye, iki saniye, bir saniye … olacak şekilde değiştirilmelidir. Bu görev öğrenciler tarafından yapılacaktır. Öğrencilerden döngü içerisindeki ilk değer, son değer ve artış miktarını değiştirerek "i" değişkenini geriye doğru sayabileceğini keşfetmeleri beklenmektedir. Öğrencilere bu keşif için gerekli süre verilmelidir. Bu süre sonunda öğrenciler başarılı olamazsa rehber öğretmen programın nasıl yapılacağını açıklayabilir. Aşağıdaki resimde örnek bir kod verilmiştir.



Resim 64. Örnek Program

1.5 Uygula- Katlar Yanıp Sönüyor

Malzeme Listesi:

Arduino Uno
Breadboard
Led küpü

Bu etkinlikte öğrencilerden led küpünün yanıp sönmesini aşağıda belirtildiği şekilde kodlamaları istenir (Not: adım basamakları arasındaki süre öğrenciler tarafından belirlenmelidir):

(i) Birinci kattaki led'ler yanar,

(ii) Birinci kattaki led'ler söner ve ikinci kattakiler yanar,

(iii) İkinci kattaki ledl'er söner ve üçüncü kattakiler yanar,

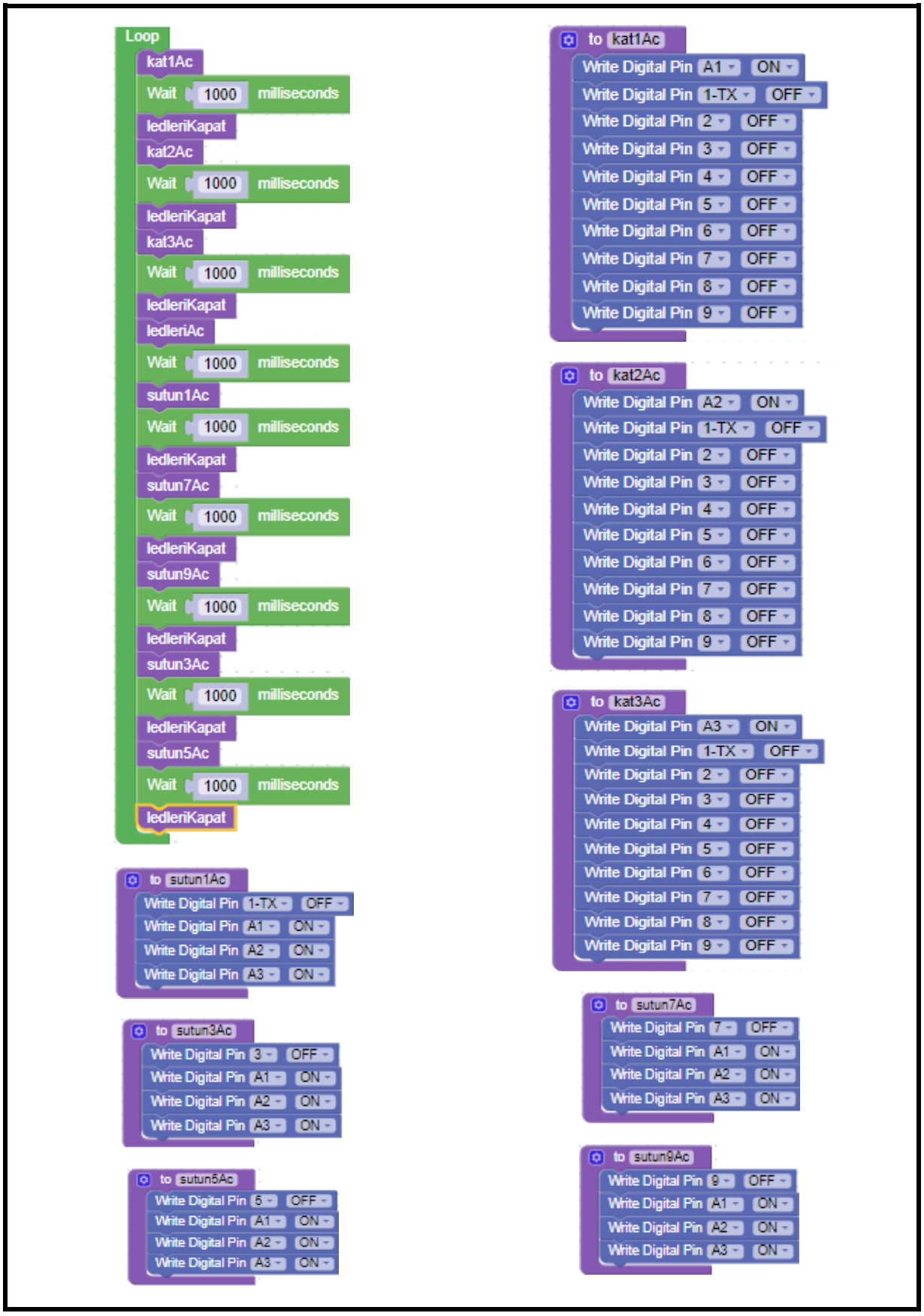
(iv) Küpün tamamı yanar,

(v) Küpün tamamı söner,

(vi) S1, S7, S9, S3 ve S5 sütunlarındaki led'ler sırasıyla yanar,

(vii) Bütün led'ler söner.

Bu etkinlik için örnek bir kod aşağıdaki resimde verilmiştir. Öğrenciler kendi kodlarını yazmalıdırlar. Rehber öğretmen bu etkinlikte öğrencileri kodlarını yazarken fonksiyon kullanmaları konusunda uyarmalıdır. Örnek kodun aşağıda görüldüğü gibi tasarlanmasının sebebi pedagojiktir. Bu kod programlama açısından bakıldığında farklı bir şekilde de tasarlanabilir. Aslında farklı fonksiyonların içerisinde tekrar eden kodların olması o fonksiyonların atomik olmadığını gösterir. Fakat aşağıdaki fonksiyonlar dersin amacı doğrultusunda yeterlidir.

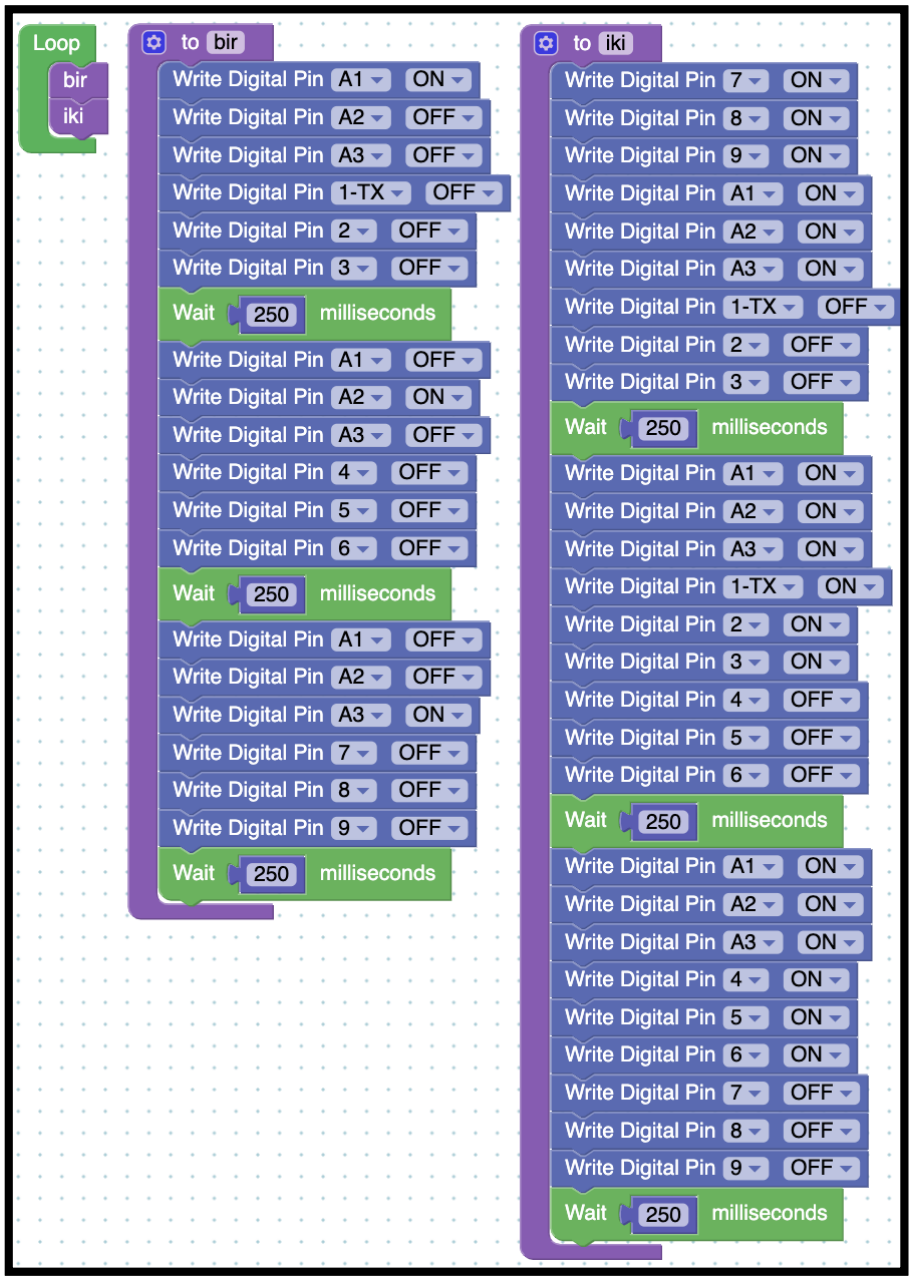


Resim 65. Örnek Program

2. TASARLA ve ÜRET

2.1 Tasarla ve Üret- Şarkıyla Uyumlu Led Küpü

Bu etkinlikte öğrenciler bir şarkı seçeceklerdir ve seçtikleri şarkının ritmi ile uyumlu led küpü animasyonu yapacaklardır. Öğrenciler istedikleri şarkıyı seçebilirler. Şarkı bilgisayardan oynatılırken led küpü de çalıştırılacaktır. Gruplar şarkı ve led küpünü senkronize bir şekilde sınıfta sunacaklardır. Programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Program kodu için aşağıdaki resimde görülene benzer bir kod oluşturabilirler.



Resim 66. Örnek Program

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Bu bölümde led küpü etkinliğinde küpün her bir katı oluşturulduğunda oluşturulan kattaki ledlerin çalışıp çalışmadığı kontrol edilmiştir ve katlar birbirine bağlandıktan sonra bütün led'ler kontrol edilmiştir. Bunun yerine led küpünün tamamı bitirildikten sonra ledler kontrol edilseydi ne gibi sıkıntılarla karşılaşılabilirdi.

2. Led küpü oluşturulurken büyük bir görev parçalara ayrılarak ve her bir parçanın ayrı ayrı çalışıp çalışmadığı kontrol edilerek ilerlenmiştir. Arkasından parçalar birleştirilmiştir ve oluşturulan bütün yeniden test edilmiştir. Bu şekilde ürün geliştirme günlük yaşamda nerelerde kullanılabilir. Siz daha önce bu şekilde herhangi bir ürün geliştirdiniz mi?

3. Bu bölümde ve geçen bölümde öğrenilen bilgiler ile günlük yaşamda kullanılmak üzere nasıl bir elektronik ürün oluşturabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Led küpünün katlarının oluşturulmasındaki mantığı anlamakta zorlandım.
    - Buzzer ile melodi oluştururken frekans değerinin ne olduğunu anlamakta zorlandım.
    - Led küpü ile şarkıyı uyum içerisinde kullanırken zorlandım.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Buzzerla Çalınan Şarkıyla Uyumlu Led Küpü

Bu etkinlikte öğrenciler bir şarkı seçeceklerdir. Seçtikleri bu şarkıyı buzzer ile çaldıracaklardır. Aynı zamanda öğrenciler buzzer'dan çalınan şarkının ritmi ile uyumlu bir led küpü animasyonu yapacaklardır. Her ikisini aynı zamanda çalıştırarak ses ve animasyondan oluşan bir şov gerçekleştireceklerdir.

4.2 Led Küpü Animasyonu (Yarışma)

Bu etkinlikte öğrenciler led küpü animasyonu yarışması yapacaklardır. Her bir grup kendi led küpü animasyonunu oluşturmalıdır. Animasyonun uzunluğu 1 dakikadan kısa olmamalıdır.1 dakikadan kısa

olan animasyonlar yarışmadan elenir. Öğrenciler animasyonu yaparken döngüler ve değişkenleri kullanmak zorundadır. Döngü ve değişken içermeyen kodlar kabul edilmeyecektir ve yarışmadan elenecektir. Bütün gruplar animasyonlarını tamamladıktan sonra sınıfta sunacaktır. Her bir grup sunan gruba 10 puan üzerinden not verecektir. Öğretmen 20 puan üzerinden not vermelidir. Sunumlar sonrasında puanlar toplanıp öğretmen tarafından birinci grup sınıfta ilan edilecektir.

3. Bölüm-İletkenler, Yalıtkanlar, Yarı İletkenler ve Arduino Giriş Pinleri

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler lehim kullanmayı bilir.

Bölümün Kazanımları:

- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir.
- Öğrenciler diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler diyot kullanarak VE/VEYA kapısı içeren devreler oluşturabilir.
- Öğrenciler VE/VEYA mantıksal ifadelerini kullanarak program yazabilir.
- Öğrenciler Arduino kartın dijital girdi pinlerini kullanarak program yazabilir.
- Öğrenciler RGB led'ler için gerekli elektronik devreleri oluşturabilir.
- Öğrenciler RGB led'leri içeren elektronik devreleri programlayabilir.
- Öğrenciler if/else karar yapısını kullanarak programlama yapabilir.

Haftanın Amacı:

Bu bölümün ilk amacı, öğrencilerin iletken, yalıtkan ve yarı-iletken kavramlarını öğrenerek (yarı iletkenlerden olan diyotları kullanarak) VE/VEYA kapıları içeren elektronik devreleri oluşturmaları ve ArduinoBlocks ortamında VE/VEYA ifadelerini kullanarak program yazmalarını sağlamaktır. Bu bölümün ikinci amacı öğrencilerin dijital girdi pinlerini kullanarak program yazmalarını sağlamaktır. Bölümün son amacı ise, öğrencilerin RGB led'leri kullanabilmek için gerekli elektronik devreleri oluşturmaları ve ArduinoBlocks ortamında RGB led'leri programlanmasını gerçekleştirmektir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Direnç (2 adet 100 ohm, 1 adet 220 ohm, 4 adet 320 ohm, 2 adet 10 Kohm), 2 adet aktif buzzer, 2 adet diyot, 4 adet led, 2 adet buton, 1 adet RGB led, Arduino Uno, breadboard (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir).

Bölümün İşlenişi:

Gözle: İletken, yalıtkan ve yarı-iletken kavramlarının açıklanması. VE/VEYA kapılarının tanımlanması ve kullanılması. Dijital girdi pinlerinin tanımlanması, elektronik devlerde kullanılması ve programlarının oluşturulması. RGB led'in tanımlanması ve kullanılması.

Uygula: İletken, yalıtkan ve yarı-iletken kavramlarının tartışılması. Yarı iletkenlerden olan diyot içeren devrelerin oluşturulması ve programlanması. VE/VEYA kapıları içeren devrelerin oluşturulması ve programlanması. Dijital girdi pinlerinin elektronik devrelerde kullanılması ve programlarının oluşturulması. RGB led devrelerinin oluşturulması ve programlanması.

Tasarla: Diyot ile "VEYA" kapısı oluşturma, mikrodenetleyici özelliklerini kullanmadan "VE/VEYA" kapısı oluşturma, iki butonla bir ledi yönetme, bir butonla bir ledi yönetme için tanımlama ve fikir üretme adımlarının geliştirilmesi.

Üret: Tasarla adımında geliştirilen Diyot ile "VEYA" kapısı oluşturma, mikrodenetleyici özelliklerini kullanmadan "VE/VEYA" kapısı oluşturma, iki butonla bir ledi yönetme, bir butonla bir ledi yönetme etkinliklerinin devrelerinin ve programlarının oluşturulması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE ve UYGULA

1.1 Gözle-Yarı İletkenler

Maddeler birçok şekilde sınıflandırılabileceği gibi akım iletme özelliğine göre de sınıflandırılabilir. ***İki ucuna potansiyel farkı uygulanan bir madde eğer elektrik akımını iletiyorsa buna iletken; elektrik akımını iletmiyorsa buna da yalıtkan denir***. Çevremizdeki birçok madde elektrik akımını iletir. Ancak bazı maddeler elektrik akımını o kadar az iletirler ki bu nedenle elektriği iletmedikleri kabul edilir ve "yalıtkan" olarak adlandırılır. Bununla birlikte, bazı maddeler ise elektrik akımına karşı neredeyse hiç direnç göstermez. Bu nedenle "iletken" olarak sınıflandırılır. Plastik maddeler ve tahta parçaları yalıtkanlara örnek olarak verilebilirler çünkü neredeyse elektriği hiç iletmezler. Metaller ve tuzlu su iletken olarak sınıflandırılabilirler çünkü elektrik akımına karşı çok az direnç gösterirler. Bu nedenle elektrik devrelerinde akımı bir yerden başka bir yere iletmek için genelde iyi iletken olduğu bilinen bakırdan yapılan teller kullanılır.

Bazı maddeler vardır ki bu iki türün de özelliklerini taşır. Belli durumlarda iletken gibi davranırken, belirli durumlarda ise yalıtkan gibi davranırlar. Örneğin yalıtkan gibi görünen bazı maddelere ısı verildiğinde bu maddelerin elektriği ilettiği görülür. Benzer şekilde bazı maddeler diğer hallerde yalıtkan gibi davranırken üzerine ışık düştüğünde elektrik akımını iletebilir. Birçok sensörün temel çalışma prensibi belli koşullar altında iletkenlikleri değişen maddeler üzerine inşa edilmiştir. İşte bu özellikleri taşıyan maddelere yarı iletken maddeler denir. Yarı iletken maddeler ile yapılmış bazı temel elektronik devre elemanları vardır. Bunlara örnek olarak diyot ve transistör sayılabilir.

1.1 Uygula- Günlük Yaşamda Yarı İletkenler

Rehber öğretmen aşağıdaki soruyu sınıfça tartıştırır:

(i) Evlerde doğalgaz/gaz kaçağı olduğu durumlarda alarm ile uyarı veren elektronik devreler bulunmaktadır. Bu devreler içerisinde gaz kaçağının olup olmadığını anlayan gaz sensörleri bulunur. Bu sensörler gaz kaçağını algıladığında elektrik akımını iletirler ve alarmın çalınmasını tetiklerler. Gaz sensörlerinin yapımında yarı iletkenler kullanılabilir mi? Cevabınızı sebepleriyle açıklayınız.

(ii) Günlük yaşamda kullanılan elektronik ürünlerde birçok yarı iletken kullanılmaktadır. Bu yarı iletkenlere örnekler veriniz.

1.2 Gözle ve Uygula- Diyot

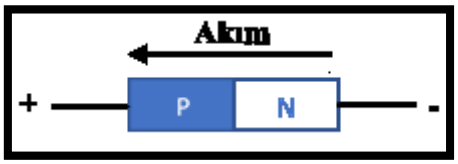
Malzeme Listesi:

Arduino Uno
Breadboard
1 adet diyot
1 adet aktif buzzer
1 adet 100 Ohm direnç

Diyot yarı iletken maddelerden üretilmiş, iki bağlantı noktası olan bir devre elemanıdır. Diyot P ve N tipi iki yarı iletken maddenin birleşmesinden oluşur. Aşağıdaki resimde bir diyotun içyapısı gösterilmiştir. Mavi alan P tipi yarı iletkeni beyaz alan ise N tipi yarı iletkeni temsil etmektedir. İçyapısı aşağıdaki resimde verilen bir diyotun iki tarafında birer bağlantı noktası vardır. Bağlantı noktalarından birine anot, diğerine ise katot adı verilir.

Resim 67. Diyot

Anot bağlantı noktasına "+" ve katot bağlantı noktasına "-" yüklü gerilim uygulandığına diyot iletken hale gelir ve akım geçirmeye başlar. Aşağıdaki resimde diyotun iletken olması durumu gösterilmiştir.



Resim 68. Diyotun İletken Olması

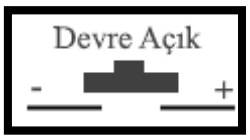
Bu durum aşağıdaki resimde gösterildiği gibi özetlenebilir. Anot ucuna "+" katot ucuna "–" yüklü gerilim uygulandığında diyot iletken gibi görev görür. Diyotu bir anahtar gibi düşünecek olursak, anahtar devreyi kapatır ve akım geçişine izin verir. Aslında diyotun iletken olması için iki ucuna verilen gerilimin eşik değerini aşmış olması gerekir. Diyotta kullanılan yarı iletken maddenin cinsine bağlı olmak üzere (Silisyum ~ 0.7 V veya Germanyum ~ 0.3 V) eşik değer aşıldığı an diyot iletken gibi hareket etmeye başlar. Bu eşik değerinin altında uygulanan gerilimlerde aslında iletim gerçekleşmez.

Resim 69. Devre Karşılığı

Ancak bunun tersi olduğunda yalıtkan gibi davranır. Yani anot bağlantı noktasına "-" ve katot bağlantı noktasına "+" yüklü gerilim uygulanırsa diyot akımı iletmez (aslında belirli bir gerilim değerinden sonra iletecektir fakat şimdilik pedagojik açıdan burada bahsedilmemelidir). Aşağıdaki resimde diyotun yalıtkan olma durumu gösterilmiştir.

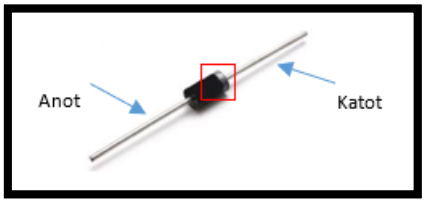
Resim 70. Diyotun Yalıtkan Oluşu

Bu durum aşağıdaki resimde olduğu gibi gösterilebilir. Diyotun anot bacağına "-" katot bacağına "+" uygulanırsa, diyot devreden akım geçmesine izin vermez.



Resim 71. Devre Karşılığı

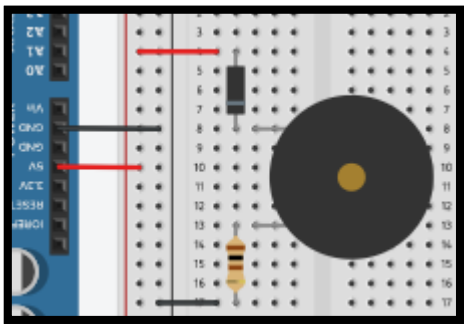
Birçok farklı türde diyot vardır. Aşağıda temel bir diyotun resmi yer almaktadır. Resimdeki gibi bir diyotun anot ve katot uçlarını ayırt etmek için üzerinde yer alan gri renkli şeride bakılır. Bu şerit hangi tarafa yakınsa o taraftaki bağlantı noktası katot, diğeri anottur.



Resim 72. Diyot Bağlantıları

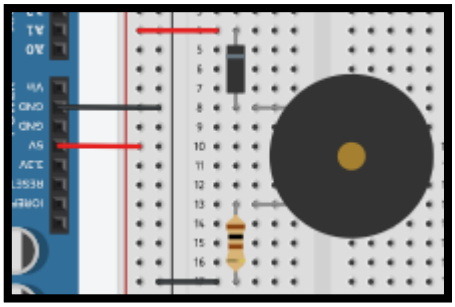
Resimdeki basit diyotun dışında farklı türde diyotlar da vardır. Aslında şimdiye kadar sıklıkla kullanılan led'ler de bir çeşit diyottur. Basit diyotun taşıdığı tüm özellikleri barındırır. Tek farkı iletken olduğu durumda basit led'den farklı olarak, aynı zamanda, bir ışık yayar. Led kelimesi "ligth emiting diyot (ışık yayan diyot)" kelimelerinin baş harflerinden oluşur.

Diyotların çalışma prensiplerini anlamak için aşağıdaki devre rehber öğretmen tarafından öğrencilere anlatılarak kurulur. Bu devrede Arduino 5V güç kaynağı olarak kullanılmıştır. Yani Arduino'nun mikrodenetleyici özellikleri kullanılmamıştır. Arduino yerine "5V" güç kaynağı kullanılırsa da devre aynı şekilde çalışır. Öğrencilere Arduino'nun güç kaynağı olarak kullanıldığı rehber öğretmen tarafından tekrar vurgulanır. Bu devrede diyot uygun şekilde bağlandığı için aktif buzzera akım gider ve buzzer bip sesini çalar. Aşağıdaki resimde diyotun iletken gibi davranması görülmektedir.



Resim 73. Diyotun İletken Gibi Davranması

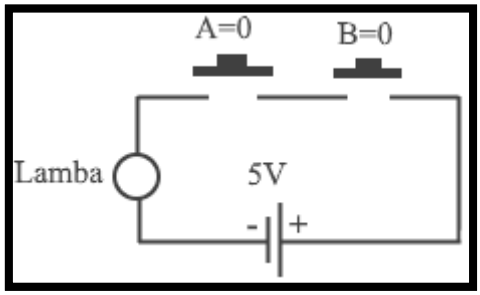
Rehber öğretmen öğrencilerden diyotun yönünü ters çevirerek devreyi yeniden kurmalarını ister. Aşağıdaki resimde diyotun yönü ters çevrilmiştir. Devreye yeniden güç verilir. Öğrenciler buzzerdan ses gelmediğini gözleyeceklerdir. Diyot ters bağlandığı için akımı iletmez ve devrede akım olmadığı için buzzer çalışmaz.



Resim 74. Diyotun Yalıtkan Gibi Davranması

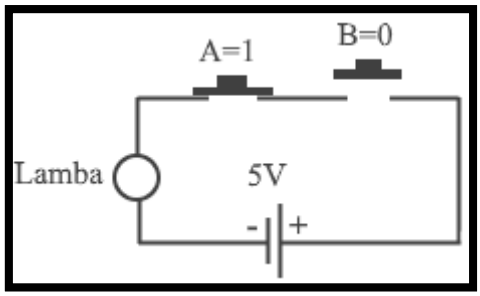
1.3 Gözle- Elektronikte Kapılar

Elektronikte aritmetik (toplama, çıkarma, çarpma ve bölme) ve mantıksal işlemler (VE VEYA ve DEĞİL) önemlidir. Bu işlemler kullanılarak karmaşık işlemler yapılabilir. Temel olarak Arduino mikrodenetleyicisi de bu işlemleri gerçekleştirir. Mantıksal işlemler elektronikte kapılar vasıtasıyla yapılır. "VE", "VEYA" ve "DEĞİL" üç temel kapıdır. Bu kapılar kullanılarak diğer bütün kapılar elde edilebilir. "VE" kapısı elektronikte kullanılan temel kapılardan birisidir. "VE" kapısı arka arkaya bağlanmış iki anahtardan oluşan bir devreye benzetilebilir. Aşağıdaki resimde görüldüğü gibi anahtarların her ikisi de açık konumdayken lamba yanmaz. Yani lamba üzerinden akım geçmediği için değeri "0" (OFF) olur.



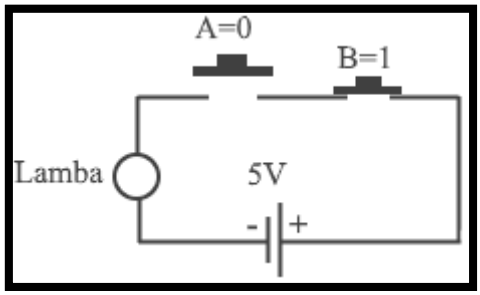
Resim 75. Ve Kapısı Durum 1

A anahtarı kapalı ve B anahtarı açıkken (A=1, B=0) Lamba yine yanmaz. Yani lamba üzerinde oluşan değer yine 0 olur. Aşağıdaki resimde "VE" kapısı durumu görülmektedir.



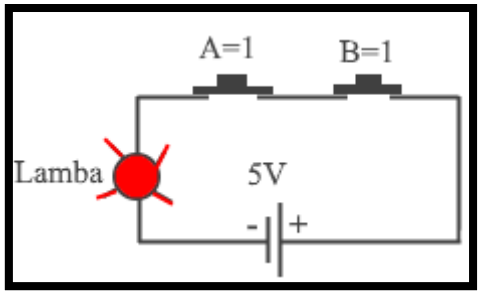
Resim 76. Ve Kapısı Durum 2

Aşağıdaki resimde görüldüğü gibi A anahtarı açık ve B anahtarı kapalı iken (A=0, B=1) lamba yine yanmaz. Yani lamba üzerinde oluşan değer yine 0 olur.



Resim 77. Ve Kapısı Durum 3

A ve B anahtarının her ikisinin kapalı oluğu durumdaysa (A=1, B=1) devreden akım geçer ve lamba yanar. Yani lambanın değeri aşağıdaki resimdeki gibi "1" (ON) olur.



Resim 78. Ve Kapısı Durum 4

Sonuç olarak söylemek gerekirse A ve B'nin "1" olduğu durumlar dışındaki tüm durumlarda sonuç "0" (0V) ve yalnızca A ve B'nin "1" olduğu durumda sonuç "1" (5V) olur. Bu durum aşağıdaki tabloda gösterilmiştir.

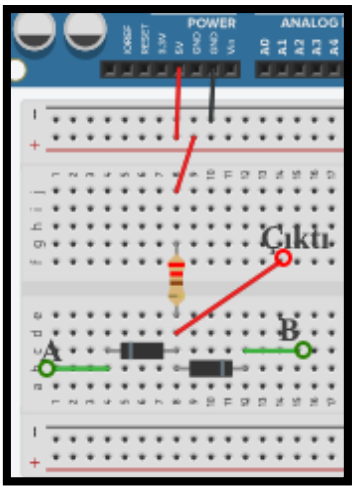
A	B	Çıktı
0	0	0
0	1	0
1	0	0
1	1	1

1.3 Uygula- Diyot ile VE Kapısı Yapıyorum

Malzeme Listesi:

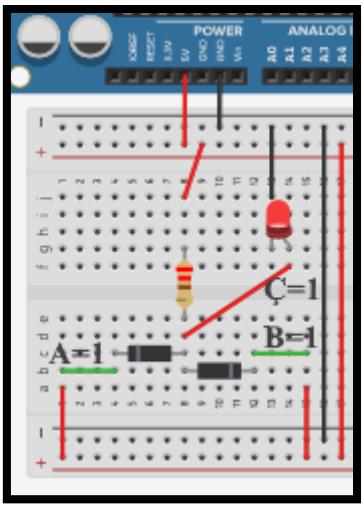
Arduino Uno
Breadboard
2 diyot
1 led
1 220 ohm direnç

Diyotlar kullanılarak VE kapısı oluşturulabilir. Aşağıdaki resimde A ve B "Girdi" uçları ve "Çıktı" ucu gösterilmiştir. Burada A ve B uçlarına verilen değerlere göre "Çıktı" ucundan alınacak değer değişir.



Resim 79. Basit Bir Ve Kapısı Oluşturma

Çıktı ucuna bir led takılırsa devrenin çalışması daha kolay anlaşılır. A ve B ucuna "1" (5V) değerleri verildiğinde led yanar, yani Çıktı değeri "1" (5V) olur. A ve B uçlarına "1" değeri verildiğinde her iki diyotta yalıtkan olur. Diyotlar akımı iletmediği için akımın iletileceği tek yol Çıktı'dır. Yani "Çıktı" değeri "1" olmuştur. Aşağıdaki resimde doğruluk tablosu kontrolü görülmektedir.



Resim 80. Doğruluk Tablosu Kontrolü

Rehber öğretmen öğrencilerden A ve B uçlarına doğruluk tablosundaki diğer değerleri verip "Çıktı" değerini kontrol etmesini ister. Sonucun neden bu şekilde gerçekleştiği sınıfça tartışılır. Eğer sınıf tartışmalarında istenilen yanıt elde edilemezse rehber öğretmen öğrencilere şu açıklamayı yapar: "Diğer bütün değerler için çıktı değeri 0 olacaktır. Çünkü diğer durumlarda diyotlardan bir tanesi mutlaka akımı iletir ve kısa devre oluşturur. Kısa devre olduğu için akım diyot üzerinden akacaktır ve Çıktı'ya ulaşamayacaktır. Böylece "Çıktı" değeri "0" olur."

Dikkat

A ve B girdi uçlarından girdi yapılmadığı durum burada göz ardı edilmelidir. VE kapısının doğruluk tablosuna bakıldığında A ve B mutlaka ya 0 ya da 1 değeri almalıdır.

1.4 Gözle ve Uygula- Arduino'da Dijital Girdi- Push Buton ile Led yakma

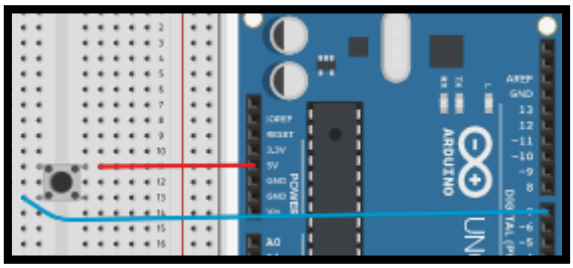
Malzeme Listesi:

Arduino Uno
Breadboard
1 adet buton
10 Kohm ve 220 Ohm direnç
Led

Arduino'nun pinlerinden "0 ve 1" olmak üzere dijital çıktı verilebildiği daha önce anlatılmıştı. Dijital pinler bunun yanında dijital girdi için de kullanılabilir. Yani dijital pinlere "0" (0V) veya "1" (5V) değerleri gönderip işlemler yapılabilir. Bu etkinlikteki amaç push butonuna basıldığında led yakan devreyi tasarlayıp programı yazmaktır. Bu iş için bir push buton kullanılacaktır. Push butonun aşağıdaki resimde gösterildiği gibi dört bacağı bulunur. Bunlardan karşılıklı ikisi (mavi çizgi ile gösterilmiştir) birbirlerine bağlıdır. Aynı şekilde karşıdaki iki bacak da birbirine bağlıdır. Fakat bu gruplar birbirine bağlı değildir. Butona basıldığında bu iki grup birbirine bağlanmış olur. Aşağıdaki resimde buton görülmektedir.

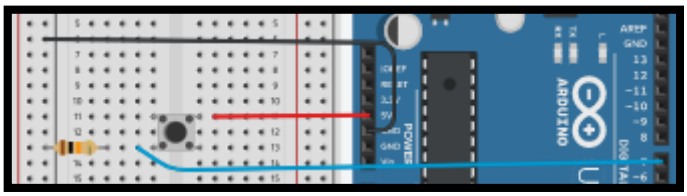
Resim 81. Buton

Aşağıdaki resimde botunun bir bacağına 5V verilmiştir diğer bacağı ise 7 numaralı pine bağlanmıştır. Butona basıldığında 7 numaralı pine "1" bilgisi iletilecektir. Bu pin'den gelen "1" değeri okutulabilir. Fakat bu bağlantının çeşitli sakıncaları bulunmaktadır. Arduino bu bağlantı ile butondan gelen değerleri yanlış okuyabilir. Yanlış okumaları engellemek için pull down direnç bağlantısı yapılmalıdır.



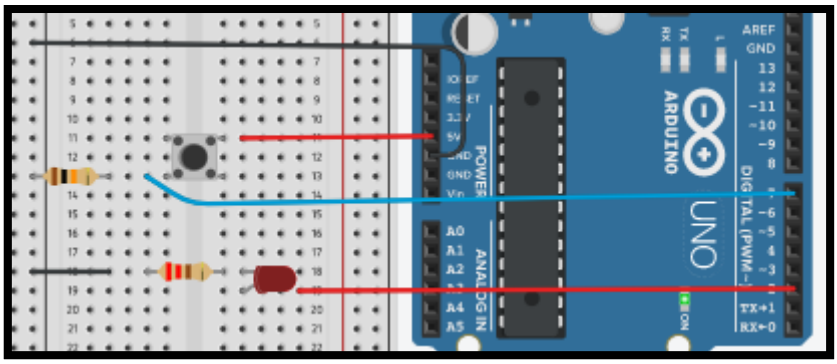
Resim 82. Buton Yanlış Bağlantı

Pull down direnç bağlantısı aşağıdaki resimde gösterilmiştir. Butonun bacağına 10K Ohm bir direnç ile GND bağlantısı yapılmıştır. Bu bağlantı sayesinde butona basılmadığında "0" değeri 7 numaralı pine gönderilecektir. Böylece butona basıldığında 7 numaralı pine "1"; butona basılmadığında 7 numaralı pine "0" bilgisi gönderilmiş olur.



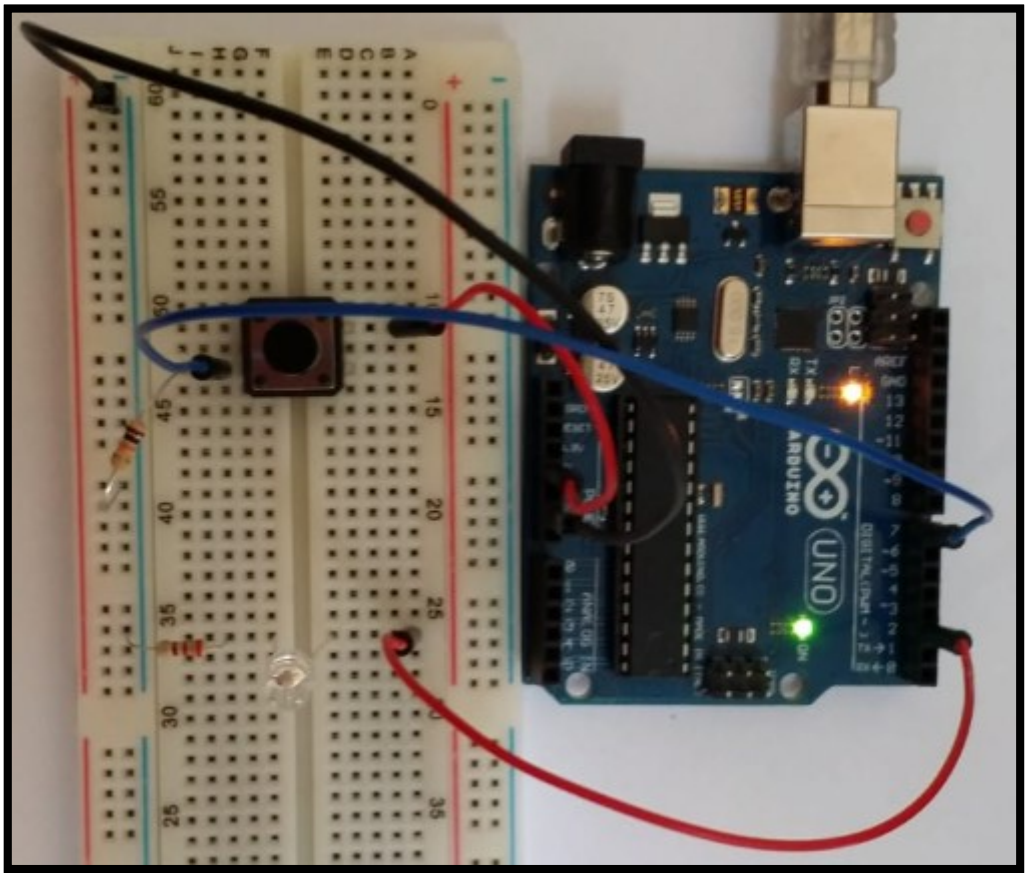
Resim 83. Pull Down Bağlantı

Aşağıdaki resimde devre butona basıldığında 2 numaralı dijital pine bağlı led'in yanması için tasarlanmıştır.



Resim 84. Örnek Devre Şeması

Bu devre gerçek yaşamda aşağıdaki resimde görüldüğü gibi olacaktır (buton'un breadboard'a takılı olduğu yere dikkat edilmelidir, orta kısım dışında başka bir yere takılırsa çalışmayacaktır).

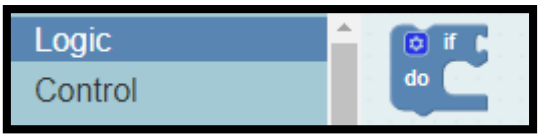


Resim 85. Gerçek Yaşam Devre Kurulumu

Butona basıldığında 7 numaralı pinde 1 değeri okunur. 7 numaralı pin'de 1 değeri okunduğunda 2 numaralı pin'den 1 (5V) bilgisi gönderilip led yakılmalıdır. Bu görev için aşağıdaki komutlar kullanılabilir. Burada if-else bloğu kullanılmıştır. 7 numaralı dijital pin'den okunan değer 1 ise (koşul doğruysa) "do" kısmında yazan komutlar çalıştırılır. Yani 2 numaralı dijital pine 1 gönderilir (led yanar). 7 numaralı dijital pin'den okunan değer 1 değilse (koşul yanlışsa) "else" kısmındaki kodlar çalıştırılır. Yani 2 numaralı pin'e 0 gönderilir (led söner). Aşağıdaki resimde örnek program görülmektedir.

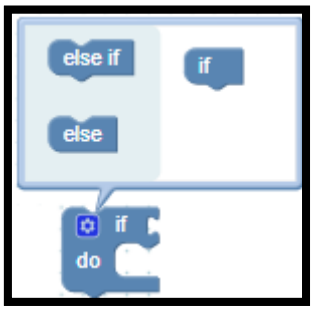
Resim 86. Örnek Program

"if-else" bloğu "Logic" blok başlığında bulunur. "Logic" blok başlığı aşağıdaki resimde gösterilmiştir.



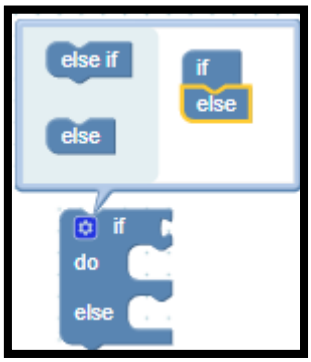
Resim 87. If-Else Bloğu

Bu blok başlığında bulunan "if" bloğu kodlama kısmına sürüklenip bırakılır. "if" bloğunun sol üst köşesindeki özellikler tıklanır. "İf" bloğu oluşturma adımları aşağıdaki resimlerde görülmektedir.



Resim 88. If-Else Bloğu Oluşturma Adım 1

Çıkan pencerede sol tarafta bulunan "else" bloğu sağ tarafta bulunan if'in altına taşınır. Böylece "if-else" bloğu elde edilmiş olunur. Burada if'in altına "else if" de eklenebilir. Bu şekilde sol tarafta bulunan bloklar sağ taraftaki if bloğunun altına eklenerek istenen koşul yapısı kullanılabilir.



Resim 89. If-Else Bloğu Oluşturma Adım 2

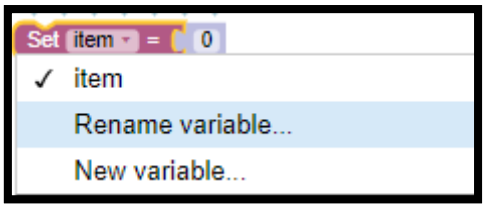
1.5 Gözle ve Uygula- Değişkenler

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet buton
10 Kohm ve 220 Ohm direnç
1 adet Led

Yukarıda yazılan kodda "if" kısmının koşul ifadesi yerine 7 numaralı dijital pin'den okunan değer konulmuştur. Bu şekilde kodda herhangi bir yanlışlık yoktur. Fakat bunun yerine 7 numaralı pin'den okutulan değer bir değişkene atanabilir ve bu değişken vasıtası ile koşul ifadesi kontrol ettirilebilir. Bunun için önce değişkenler incelenecek ve ardından yukarıda yazılan kod değişken kullanılarak yeniden yazılacaktır.

Programcılar programlarında çeşitli değerlerin hatırlanmasını ister. Örneğin bir öğrencinin aldığı haftalık harçlığı düşünülsün. Bir programcı programında kullanılmak üzere haftalık harçlık miktarının hatırlanmasını isteyebilir. Haftalık harçlık miktarının hatırlanması için değişken isimli programlama bileşeni kullanılmalıdır. Öğrencinin haftalık harçlığının 20 TL olduğu varsayılsın. Bu miktarın programda hatırlanması için "harclik" isimli bir değişken tanımlayıp içine 20 değerini aktarmak gerekir. ArduinoBlocks'da değişken tanımlamak için "Variables" blok grubu içerisindeki "Set *item*" bloğu kullanılır. Aşağıdaki resimde görüldüğü gibi "*item*" menüsü açıkken "*Rename variable*" metni üzerine tıklanarak değişkenin ismi "harclik" olarak belirlenir.



Resim 90. Değişken Oluşturma

Daha sonra "harclik" değişkenine 20 değeri atanır. Artık bu değişkenin içerisinde 20 değeri saklanacaktır. Bu işlem aşağıdaki resimde gösterilmiştir. Programcı artık programında istediği yerde "harclik" değişkeni içerisinde tutulan değeri kullanıp onun üzerinden işlem yapabilir.

Resim 91. Değişkene Değer Atama

Programlamadaki değişkenler matematikteki değişkenlerden farklıdır. Programlamadaki değişkenler bir seferde içerisinde sadece bir değer saklayabilen bir kutuya benzetilebilir. Yukarıda kod ile bilgisayarın belleğinde "harclik" isimli bir kutu oluşturulmuş ve içerisine 20 değeri konulmuştur. Bilgisayarda değişkenlerin saklandığı yere RAM ismi verilir.

Dikkat

Burada rehber öğretmen benzetmedeki kutunun bir seferde yalnızca bir değer saklayabildiğini vurgulamalıdır. Aksi takdirde öğrenciler değişkenlerin içerisinde aynı anda birden fazla değer saklanabileceğini düşünebilir.

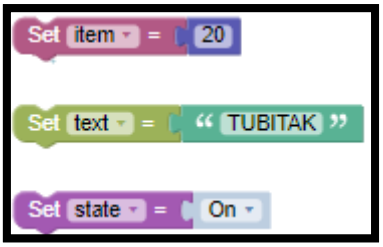
Değişken üzerinde bir işlem yapılmak istendiğinde gerekli bloklar kullanılarak istenilen işlem yapılabilir. Örneğin öğrencinin haftalık harçlığının, aldığı harçlığın yarısı kadar artırılmak istendiğini düşünelim. Bunun için "harclik" değişkeni ve matematiksel işlem blokları kullanılarak sonuç bulunabilir. Rehber öğretmen aşağıdaki resimde görülen kodu yazar ve öğrencilere yapılan işlemi açıklar.

Resim 92. Örnek Matematik İşlemi

Dikkat

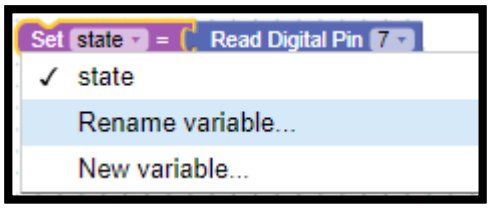
Burada öğrencilerden bazıları "harclik = harclik + harclik/2" komutunu matematikteki eşittir ile karıştırıp kavram karmaşası yaşayabilir. Bunun olmaması için programlamadaki eşittir işaretinin matematikteki eşittir işaretinden farklı olarak atama anlamına geldiği vurgulanır. Yani yapılan işlemde "harclik" değeri ile "harclik" değerinin 1,5 katı birbirine eşitlenmemiştir. Bunun yerine RAM'de bulunan "harclik" isimli kutuya yeni değeri olan "harclik+harclik/2" atanmıştır.

Değişkenler sayı cinsinden olmak zorunda değildir. ArduinoBlocks'da aşağıdaki resimde görüldüğü gibi sayı cinsinden değişkenlerin yanında metin (text) cinsinden ve doğru-yanlış (ON-OFF) cinsiden değişkenler (state) tanımlanabilir. Aşağıdaki resimde farklı tipteki değişkenler görülmektedir.



Resim 93. Farklı Tipteki Değişkenler

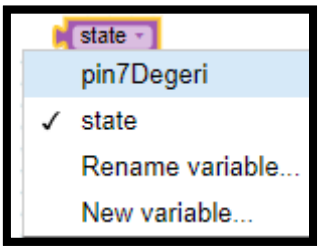
Butondan gelen girdiye bağlı olarak led yakılan program örneğine (1.4 Gözle ve Uygula) yeniden dönülür. 7 numaralı pin'den değer olarak 0 (OFF) veya 1 (ON) gelir. Bu tür değerleri okumak için state değişkeni oluşturulur. Aşağıdaki resimlerde bu türde bir değişken oluşturularak değişkene "pin7Degeri" ismi verilmiştir. Bu değişkene 7 numaralı pin'den okunan dijital değer aktarılmıştır. Artık 7 numaralı pin değeri için bu değişken kullanılabilir.



Resim 94. Değişken Oluşturma

Resim 95. İlk Değer Atama

"pin7Degeri" değişkenini kullanmak için "Variables" blok başlığı altından "state" tipi değişken seçilir. Açılan listeden önceden oluşturulmuş "pin7Degeri" isimli değişken seçilir. Bu işlem aşağıdaki resimde gösterilmiştir.

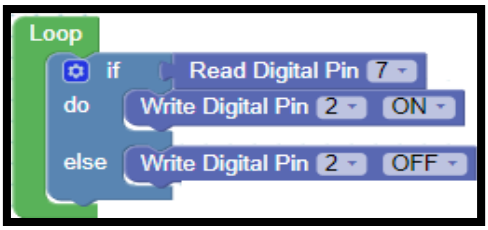


Resim 96. State Tipi Değişken

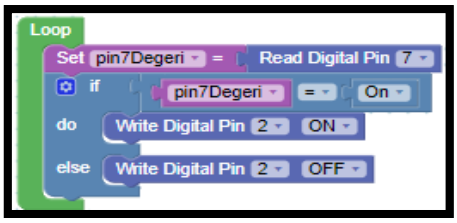
"pin7Degeri" değişkeninin ON yani 1 (5V) olup olmadığını kontrol etmek için aşağıdaki resimde görülen bloklar kullanılır. Değişken dışındaki iki blok "Logic" blok başlığı altında bulunmaktadır.

Resim 97. Koşul İfadesi

Sonuç olarak aşağıdaki resimlerde görülen kod kullanılarak da 7 numaralı pin'den girilen değere bağlı olarak 2 numaralı pine bağlı olan led yakılabilir.



Resim 98. Değişkensiz Kullanım



Resim 99.Değişkenli Kullanım

1.6 Gözle ve Uygula- Kaç Buzzer'dan Ses Çıkıyor?

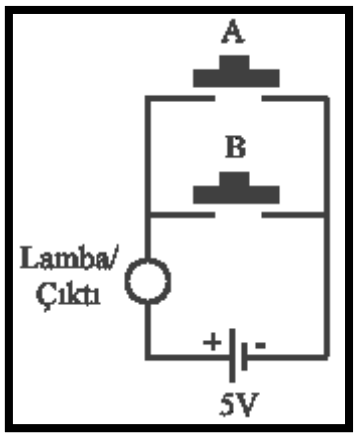
Malzeme Listesi:

Arduino Uno
Breadboard
2 adet buton
2 adet aktif buzzer
2 adet 10K Ohm, 2 adet 100 Ohm, 2 adet 330 Ohm direnç
2 adet led

Bu etkinlikte amaç ses çıkaran buzzer sayısına eşit sayıda led yakmaktır. Buzzerlardan sadece birisi (hangisinin olduğu önemli değil) bip sesi çaldığında sol led yanmalıdır. Her iki Buzzerdan da ses çıkıyorsa her iki led de yanacaktır.

Bu etkinliğin tamamlanması için "VE" ile "VEYA" operatörü kullanılacaktır. "VE" operatörü daha önceden anlatıldığı için burada "VEYA" operatörü anlatıldıktan sonra gerekli kod yazılacaktır. Rehber öğretmen aşağıda anlatıldığı şekliyle "VEYA" operatörünü anlatıp etkinliğin geri kalanını gerçekleştirmelidir.

Programlamada/elektronikte VE'nin yanında "VEYA" operatörü/kapısı da önemlidir. "VEYA" operatörü aşağıdaki resimde görülen devreye benzetilebilir.

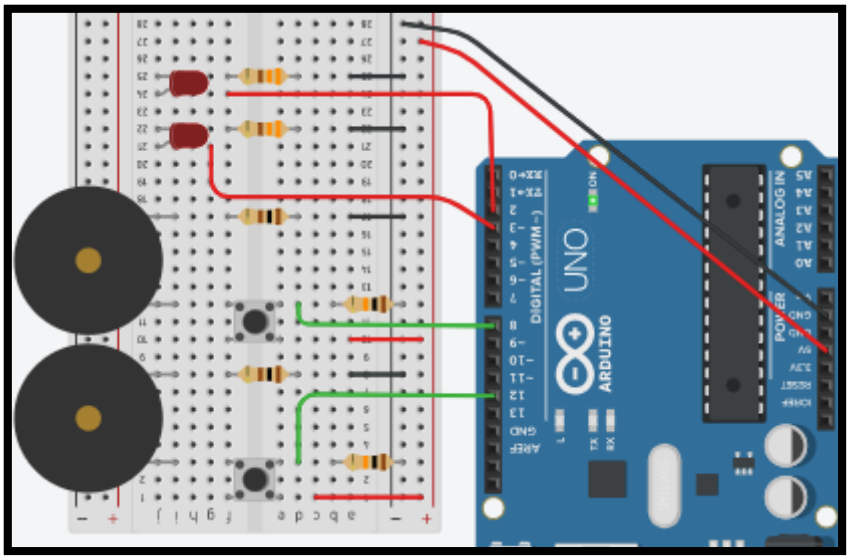


Resim 100. Veya Kapısının Modellenmesi

A=0 ve B=0 iken devreden akım geçmez ve böylece lamba yanmaz. Yani Çıktı "0" olmuştur. A=1, B=0 iken A butonu üzerinden akım geçerek lambayı yakacaktır. Yani Çıktı 1 olmuştur. Benzer şekilde A=0 ve B=1 iken lamba yanacaktır. Yani Çıktı "1" olacaktır. Son olarak A=1 ve B=1 olma durumunda da lamba yanacaktır. Yani sonuç "1" olacaktır. Sonuç olarak "VEYA" kapısının doğruluk tablosu aşağıdaki gibidir. Görüldüğü gibi "VEYA" kapısında sonucun "0" olması için her iki girdi değerinin de "0" olması gerekmektedir. Başka bir ifadeyle her iki anahtar da açık konumda olmalıdır. Girdilerden herhangi birisi veya her ikisi "1" olduğunda çıktı "1" olmaktadır. "VEYA" operatörünün doğruluk tablosu aşağıda verilmiştir.

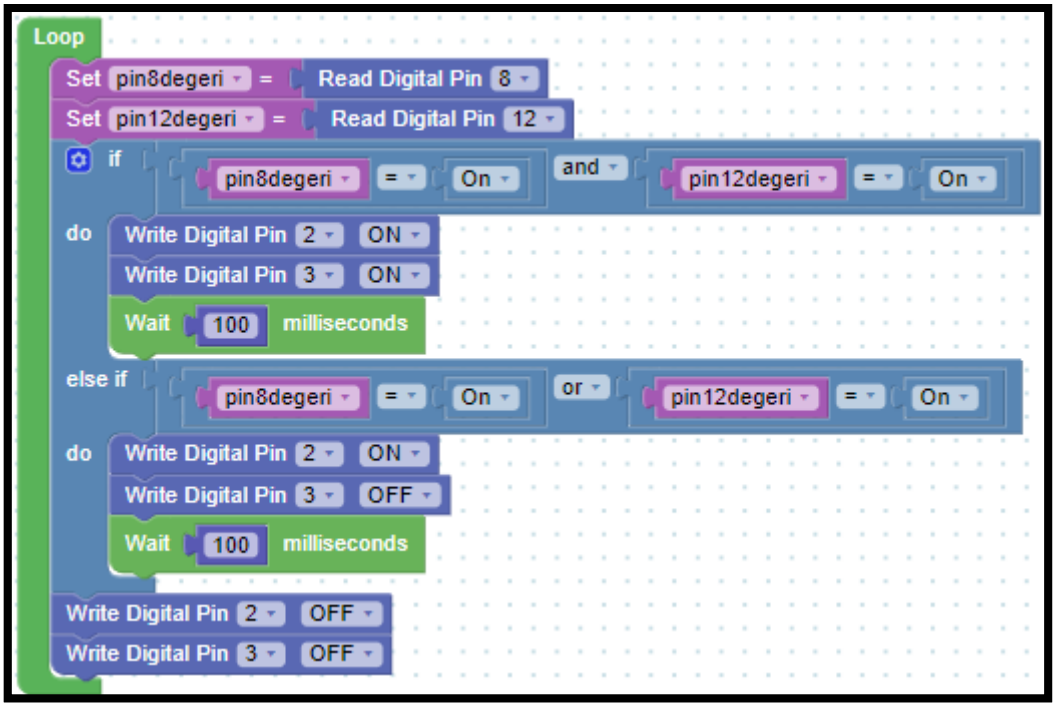
A	B	Çıktı
0	0	0
0	1	1
1	0	1
1	1	1

Bu etkinlikte kullanılacak devre tasarımı aşağıdaki resimde gösterilmiştir. Sol tarafta iki adet buton ve buzzer bulunmaktadır. “Pull down” direnç 10K Ohm seçilmiştir. Buzzerlar için kullanılan direnç 100 Ohm seçilmiştir. Led’ler için kullanılan direnç ise 330 Ohm seçilmiştir. Butonlar yardımıyla buzzerlardan bip sesi çıkarılır. Aynı zamanda alttaki butona basıldığında 12 numaralı dijital pine ve üstteki butona basıldığında “8” numaralı dijital pine “1” değeri gönderilmiş olunur. Butonlara basılmadığında bu pinlere “0” değeri gönderilir.



Resim 101. Örnek Devre Şeması

“12” ve “8” numaralı dijital pinlere gelen değerlere göre led’ler yakılacaktır. “12” numaralı pine ve “8” numaralı pine “1” bilgisi gittiğinde her iki led yanacaktır. Sadece “12” numaralı pine “1” değeri gittiğinde (8 numaralı giriş 0) üst taraftaki led yanacaktır. Sadece “8” numaralı pine “1” değeri gittiğinde (12 numaralı giriş 0) yine üst taraftaki led yanacaktır. Bu işlemi gerçekleştiren kod aşağıdaki gibidir. Rehber öğretmen bu kodu yazarak öğrencilere anlatır ve devrenin çalışmasını öğrencilere gösterir. Ardından öğrencilerden aynı devreyi kurarak ve kodu yazarak çalıştırmalarını ister. Aşağıdaki resimde örnek program görülmektedir.



Resim 102. Örnek Program

1.6 Uygula- Bir Buton Dört Hareketli Led

Malzeme Listesi:

Arduino Uno
Breadboard
4 adet led
4 adet 330 Ohm direnç, 1 adet 10K Ohm direnç
1 adet buton

Bu etkinlikte 4 adet led bir buton tarafından çalıştırılacaktır. Butona basılma şekline göre led'ler ile animasyon yapılacaktır. Çalışma şekli aşağıdaki gibi olmalıdır.

i. Butona birinci defa basıldığında 1, 2, 3 ve 4 numaralı led'ler sırasıyla yanıp söner. Devamında 1 numaralı led yanar ve yanık kalır.

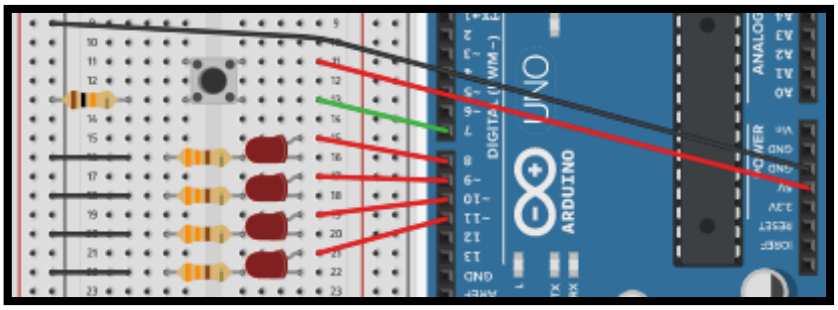
ii. Butona ikinci defa basıldığında 2, 3 ve 4 numaralı led'ler sırasıyla yanıp söner. Devamında 1 ve 2 numaralı led yanar ve yanık kalır.

iii. Butona üçüncü defa basıldığında 3 ve 4 numaralı led'ler sırasıyla yanıp söner. Devamında 1, 2 ve 3 numaralı led yanar ve yanık kalır.

iv. Butona dördüncü defa basıldığında 4 numaralı led yanıp söner. Devamında 1, 2, 3 ve 4 numaralı led yanar ve yanık kalır.

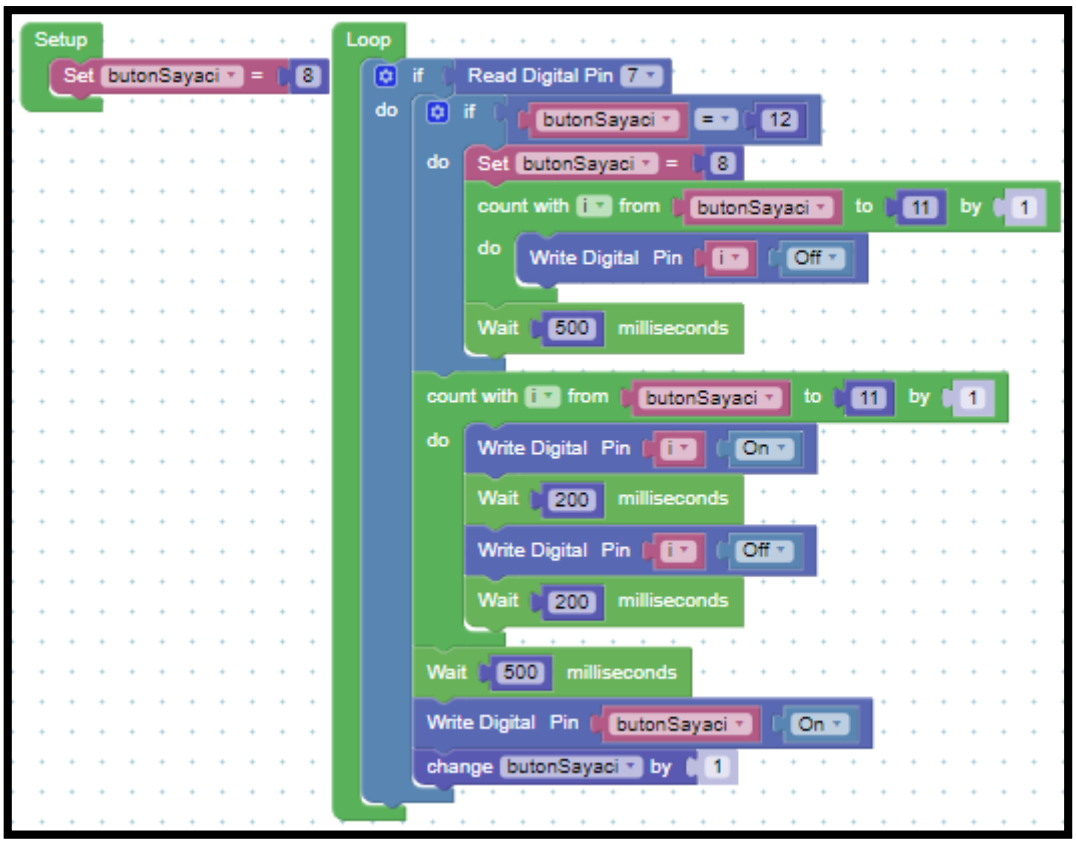
v. Butona beşinci defa basıldığında bütün led'ler söner ve butona bir defa basılmış gibi birinci adımdan yeniden başlanır.

Bu etkinlik için kullanılabilecek örnek devre tasarımı aşağıdaki resimde verilmiştir.



Resim 103. Örnek Devre Şeması

Bu etkinlik için kullanılabilecek örnek kodlar aşağıdaki resimde verilmiştir.



Resim 104.Örnek Program

1.7 Gözle- RGB Led ile Farklı Renkler

Malzeme Listesi:

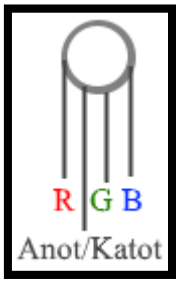
Arduino Uno
Breadboard
1 adet RGB led
3 adet 330 Ohm direnç

Kırmızı (Red), yeşil (Green) ve mavi (Blue) renkler farklı oranlarda karıştırılarak diğer renkler elde edilebilir. Bu şekilde üç temel renkten diğer renklerin elde edildiği modele RGB (**R**ed, **G**reen, **B**lue) modeli denir. Bu üç rengi aynı miktarda karıştırarak beyaz; kırmızı ve mavi rengi karıştırarak mor renk; kırmızı ile yeşil rengi karıştırarak sarı renk; mavi ile yeşil rengi karıştırarak Turkuaz rengi elde edilir.

Dikkat

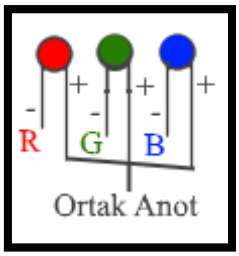
Günlük hayatta ana renkler kırmızı, sarı ve mavi olarak kabul edilmiştir. RGB sisteminde ana renkler farklıdır. Rehber öğretmen bu farkı vurgular.

Bir RGB led ile Kırmızı (Red), Yeşil (Green) ve Mavi (Blue) renkler oluşturabilir. Aynı zamanda bu renklerin karışımlarından oluşacak renkler de oluşturulabilir. Aşağıdaki resimde görüldüğü gibi RGB led'in dört bacağı bulunur. Bunlardan uzun olanı ortak anot veya ortak katot bacağını gösterir. Ortak anot/katot bacağının solunda kalan tek bacak ise kırmızı renk içindir. Ortak anot/katot bacağının sağında kalan ikili bacaktan soldaki yeşil ve sağdaki ise mavi renk içindir.



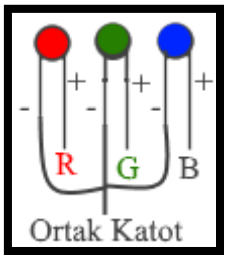
Resim 105. RGB Led

RGB led, kırmızı, yeşil ve mavi led'in birleşimi olarak düşünülebilir. Bu led'ler birleştirilirken eğer anot kısımları birbirine bağlandıysa bu ortak anot bir RGB olur. Bu durum aşağıdaki resimde gösterilmiştir.



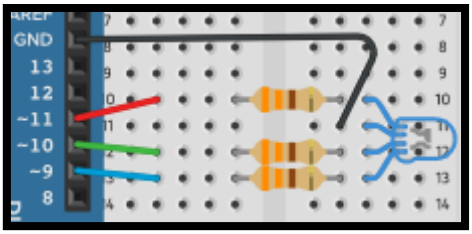
Resim 106. Anot RGB Led

Eğer led'ler birleştirilirken katot kısımları birbirine bağlandıysa bu ortak katot bir RGB olur. Bu durum aşağıdaki resimde gösterilmiştir.



Resim 107. Katot RGB Led

Ortak katot bir RGB kullanılacaksa bağlantılar aşağıdaki resimde görüldüğü gibi yapılmalıdır. Ortak katot ucuna GND, kırmızı bacağa 11 numaralı dijital pin, yeşil bacağa 10 numaralı pin ve mavi bacağa 9 numaralı pin gelecek şekilde bağlantı yapılır. Pinler ile bacaklar arasına 3 adet 330 Ohm direnç bağlanmıştır.



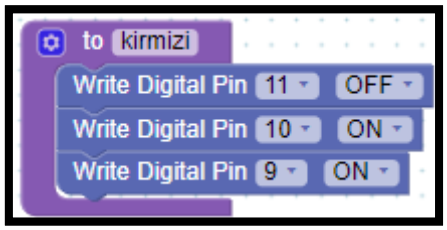
Resim 108. Ortak Katot RGB Örnek Devre Şeması

Ortak anot bir RGB kullanılacaksa, aşağıdaki resimde görüldüğü gibi bir bağlantı yapılmalıdır. Ortak anot ucuna 5V, kırmızı bacağa 11 numaralı dijital pin, yeşil bacağa 10 numaralı pin ve mavi bacağa 9 numaralı pin gelecek şekilde bağlantı yapılır. Pinler ile bacaklar arasına 3 adet 330 Ohm direnç bağlanmıştır.

Resim 109. Ortak Anot RGB Örnek Devre Şeması

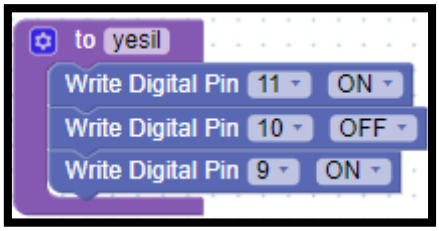
Yukarıdaki her iki bağlantıda da özellikle 9, 10 ve 11 numaralı pinler seçilmiştir. Bu pinlerin yanında "~" (tilde) işareti bulunur. Yanında "~" işareti bulunan pinlere PWM pini denilir. Şimdilik PWM'in ne anlama geldiğinden bahsedilmeyecektir. İlerleyen bölümlerde PWM anlatılacaktır. Fakat bütün renk tonlarının elde edilebilmesi için RGB bağlantısının PWM pinleri ile yapılması gerektiği bilinmelidir.

Ortak anot bir RGB'de kırmızı rengi yakmak için aşağıdaki resimde görüldüğü gibi "kirmizi" isimli bir fonksiyon tanımlanabilir. Ortak katot bir RGB led'in kullanıldığı durumlarda 11. pin için ON, 10. için OFF ve 9. için OFF değerleri verilir.



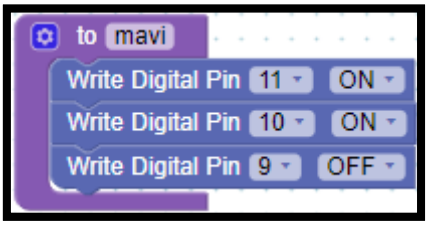
Resim 110. Ortak Anot Kırmızı Renk

Ortak anot bir RGB'de yeşil rengi yakmak için aşağıdaki resimde görüldüğü gibi bir fonksiyon tanımlanabilir. Ortak katot bir RGB led'in kullanıldığı durumlarda 11. pin için OFF, 10. için ON ve 9. için OFF değerleri verilir.



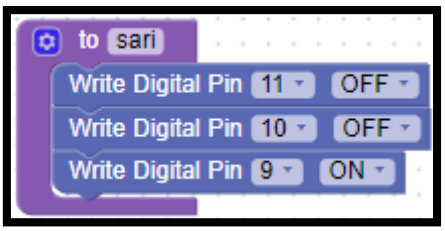
Resim 111. Ortak Anot Yeşil Renk

Ortak anot bir RGB'de mavi rengi yakmak için aşağıdaki resimdeki gibi bir fonksiyon tanımlanabilir. Ortak katot bir RGB led'in kullanıldığı durumlarda 11. pin için OFF, 10. için OFF ve 9. için ON değerleri verilir.



Resim 112. Ortak Anot Mavi Renk

Üç temel renk karıştırılarak istenilen renk elde edilebilir. Örneğin ortak anot bir RGB led'de sarı renk için aşağıdaki resimde verilen fonksiyon kullanılabilir.



Resim 113. Ortak Anot Sarı Renk

1.8 Uygula- Ortak Ucu Belirleme

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
3 adet 330 Ohm direnç

Rehber öğretmen öğrencilerden setlerinde bulunan RGB led'lerden birini alıp bu RGB led'in ortak ucun anot / katot olduğunu bulmalarını ister. Aşağıda örnek bir çözüm verilmiştir, öğrencilerin kendi çözümlerini bulmaları/denemeleri sağlanmalıdır. Öğrencilere direnç kullanmadan RGB led'leri güç kaynağına bağlamamaları gerektiği hatırlatılmalıdır.

Öncelikle RGB led'in ortak ucu olan uzun bacağı 330 Ohm bir direnç üzerinden 5V'a bağlanır. Diğer uçlardan herhangi biri GND'ye bağlanır. RGB led yanıyorsa led'in ortak anot olduğu anlaşılır. RGB led yanmıyorsa ya ortak katot bir led'dir ya da led bozuktur. Emin olmak için 5V ve GND'ye bağlı bacaklar yer değiştirilerek sağlama yapılır.

1.8 Uygula- Mor ve Turkuaz renkler

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
3 adet 330 Ohm direnç

Bu etkinlikteki amaç bir RGB led'le mor ve turkuaz renkleri yakmaktır. Rehber öğretmen öğrencilerden mor ve turkuaz renkler için fonksiyon yazmalarını ve bu fonksiyonları çağırarak çalıştırmalarını ister.

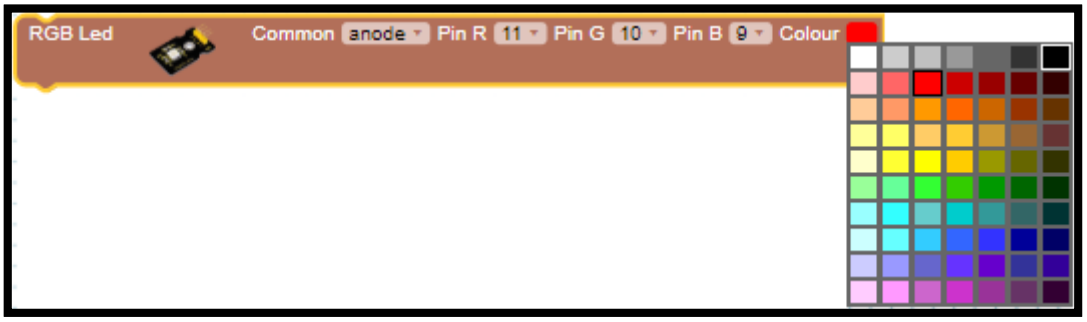
1.9 Gözle- ArduinoBlocks ile Ara Renkler

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
3 adet 330 Ohm direnç

Yukarıdaki örneklerde RGB modelinde üç ana renk olan kırmızı, yeşil ve mavi ve bunun yanında ara renkler olan sarı, mor ve turkuaz oluşturulmuştur. Aslında bu renkler dışında bulunan ara renkler de RGB ile yakılabilir. Bu iş için "ArduinoBlocks" ortamında iki farklı yöntem bulunur. Bunlar PWM yöntemi ile renkleri oluşturmak ve "ArduinoBlocks"un "RGB Led" bloğunu kullanarak renkleri oluşturmaktır. Bu etkinlikte "RGB Led" bloğu kullanılacaktır.

"RGB Led" bloğu "Actuators" blok başlığı altındadır. Bu blokta beş tane parametre bulunur. Birinci parametre RGB led'in ortak anot/katot olduğunu belirlemek içindir. İkinci parametre RGB'nin kırmızı bacağının hangi pine bağlı olduğunu belirtir. Üçüncü parametre RGB'nin yeşil bacağının hangi pine bağlı olduğunu belirtir. Dördüncü parametre RGB'nin mavi bacağının hangi pine bağlı olduğunu belirtir. Son parametre ise renk seçimidir. Var olan renklerden bir tanesi seçilir. RGB led bu rengi yakacaktır. Aşağıdaki resimde RGB led bloğu görülmektedir.



Resim 114. RGB Led Bloğu

Dikkat

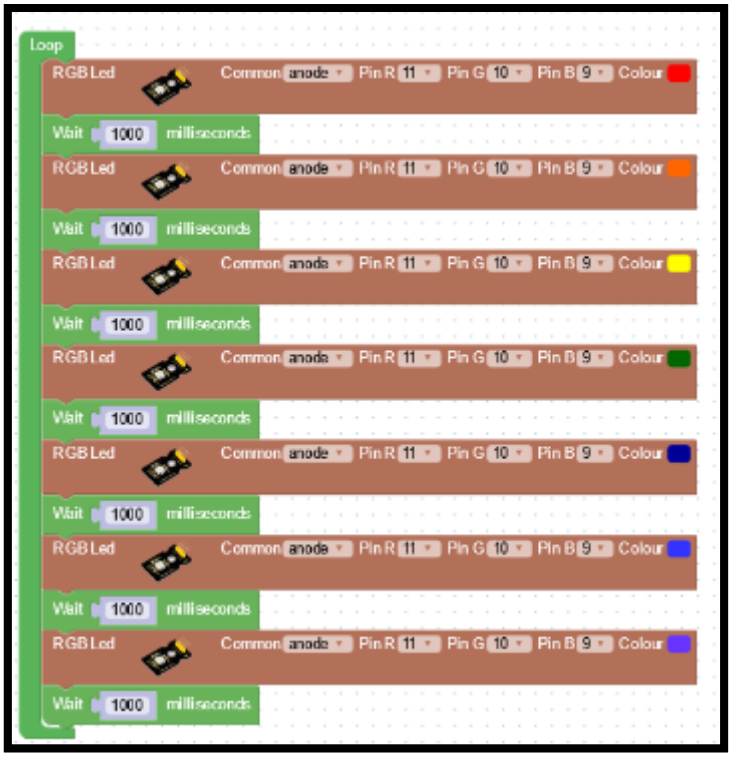
RGB ledin bacakları PWM pinlerine bağlanmalıdır. Aksi takdirde "RGB Led" bloğu istenildiği gibi çalışmayacaktır.

1.9 Uygula- Gökkuşağındaki 7 Temel Renk

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
3 adet 330 Ohm direnç

Bu etkinlikte amaç birer saniye arayla RGB led'in sırayla kırmızı, turuncu, sarı, yeşil, mavi, lacivert ve mor renklerini yakmasını sağlamaktır. Aşağıdaki resimde örnek program görülmektedir.



Resim 115. Örnek Program

2. TASARLA ve ÜRET

2.1 Tasarla- Diyot ile "VEYA" kapısı

Bu etkinlikte amaç diyot kullanarak "VEYA" kapısı oluşturmaktır. Öğrenciler "VEYA" kapısının doğruluk tablosunu bu derste öğrenmiştir. Öğrenciler diyot kullanarak bu tabloyu sağlayan bir devre tasarlamalıdır. Rehber öğretmen çözümü öğrencilere vermemelidir. Öğrenciler çözümü kendileri üretmelidir.

Diyot ile "VEYA" kapısı etkinliğini tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmeleri gerekir. Aşağıdaki örnek süreç kesinlikle öğrencilere gösterilmemelidir. Rehber öğretmen sınıfta genel bir sorun görürse sınıfa açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada rehber öğretmen sorusu olan gruplara yardımcı olabilir. Fakat öğrencilere hiçbir şekilde tam bir çözüm verilmemelidir.

Tanımlama: Öğrencilerin öncelikle diyot ile "VEYA" kapısı etkinliğinin neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrenciler öncelikle gerekli işlemleri detaylı olarak maddeler hâlinde yazmalıdır.

Örnek:

i. "VEYA" kapısında "A" ve "B"nin "0" olduğu durumlar dışındaki tüm durumlarda sonuç "1" (5V) ve yalnızca "A ve B"nin "0" olduğu durumda sonuç "0" (0V) olur.

ii. Çıktı ucuna devrenin istenildiği gibi çalıştığını kontrol etmek için bir led kullanılabilir.

iii. Diyotlardan en az birisinden akım geçtiğinde led yanmalıdır.

Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

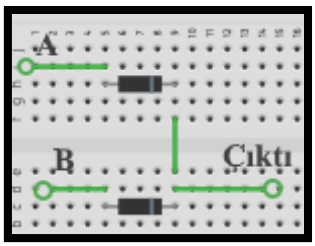
Örnek:

i. "VEYA" kapısının modellenmesinde birbirine bağlı paralel iki anahtar bulunmaktadır. Buna benzer şekilde diyotlar bağlanarak anahtar olarak kullanılabilir. Diyotlar paralel bağlandığında ayrı ayrı giriş sağlanamayacağı için diyotların anot uçları ayrık bırakılabilir, katot uçları da birleştirilebilir.

ii. "A ve B" uçlarına verilen gerilimler girdi olarak algılanır.

2.1 Üret- Diyot ile VEYA Kapısı

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Aşağıdaki resimde "VEYA" kapısının iki adet diyot ile yapılmış çözümü verilmiştir. "A" girişinden veya "B" girişinden "1" (5V) verildiği zaman "Çıktı 1" (5V) olur. "A ve B" girişinden "0" (0V) verildiği zaman "Çıktı 0" (0V) olur.



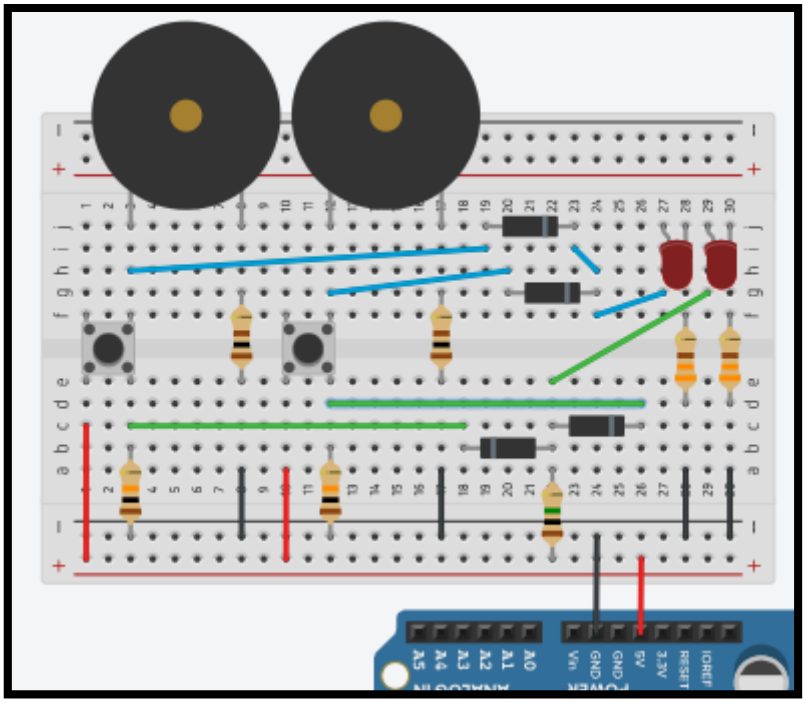
Resim 116. Diyot İle Veya Kapısı

2.2 Tasarla ve Üret- Yeniden Kaç Buzzer'dan Ses Çıkıyor?

Bu bölümde Kaç Buzzer'dan Ses Çıkıyor? (1.6 Gözle ve Uygula) etkinliğinde buzzerlardan çıkan ses sayısına göre bir veya iki led yakılmıştır. Bu görev için "ArduinoBlocks" ortamında kod yazılmıştır. Bu etkinlikte "ArduinoBlocks" ortamı kullanılmadan yani Arduino kartın mikrodenetleyici özellikleri kullanılmadan "VE" ile "VEYA" kapısı aracılığı ile aynı görevi yerine getiren devre tasarlanacaktır.

Devreyi oluşturmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

Bu devrede Arduino kartı sadece güç kaynağı olarak kullanılmalıdır. Bu iş için kullanılabilecek devre aşağıdaki resimde verilmiştir.



Resim 117. Örnek Devre Şeması

Dikkat

Bu devrede kullanılan direnç değerleri önemlidir. Yanlış direnç seçimi devrenin çalışmamasına sebep olabilir.

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Yarı iletkenler diyot ve led gibi birçok elektronik devre elemanının yapımında kullanılırlar. Elektronik ürünler açısından düşünüldüğünde yarı iletkenlerin temel görevi nedir?

2. Bu bölümde gördüğünüz konuları günlük yaşamda nerelerde kullanabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Butonun breadboard bağlantısı yanlış olmuş.
    - RGB'yi 3 direnç yerine ortak bir dirençle yakmaya çalıştım, neden çalışmıyor?

- RGB ile her rengi oluşturamıyorum, neden?

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Notalara Göre Farklı Renk Veren Piyano

Bu etkinlikte amaç buzzer, buton ve RGB kullanılarak, çalınan notalara göre farklı ışıklar üreten bir ışıklı piyano oluşturmaktır. Piyano yedi farklı buton ile "do, re, mi, fa, sol, la ve si" notalarını çalabilmelidir. Notalar buzzer aracılığı ile çalınacaktır. RGB her nota için farklı bir renk üretecektir (örneğin, do için mavi, re için yeşil, mi için kırmızı, fa için sarı, sol için pembe ve si için turuncu). Bunun yanında yapılacak bir değiştir butonu ile RGB'den notalar için verilen ışıklar değiştirilecektir. Bu buton ile beş farklı renk tepkisi üretilmelidir. Do notası için örnek vermek gerekirse:

i. Kod ilk defa yüklendiğinde, do notası çaldığında mavi renk yanar,

ii. Değiştir butonuna birinci defa basıldıktan sonra, do notasına basıldığında yeşil renk yanar,

iii. Değiştir butonuna ikinci defa basıldıktan sonra, do notasına basıldığında kırmızı renk yanar,

iv. Değiştir butonuna üçüncü defa basıldıktan sonra, do notasına basıldığında sarı renk yanar,

v. Değiştir butonuna dördüncü defa basıldıktan sonra, do notasına basıldığında mor renk yanar,

vi. Değiştir butonuna bir daha basılırsa, do notasına basıldığında yeniden ilk renk olan mavi renk yanar ve i-vi adımları sürekli tekrarlanır.

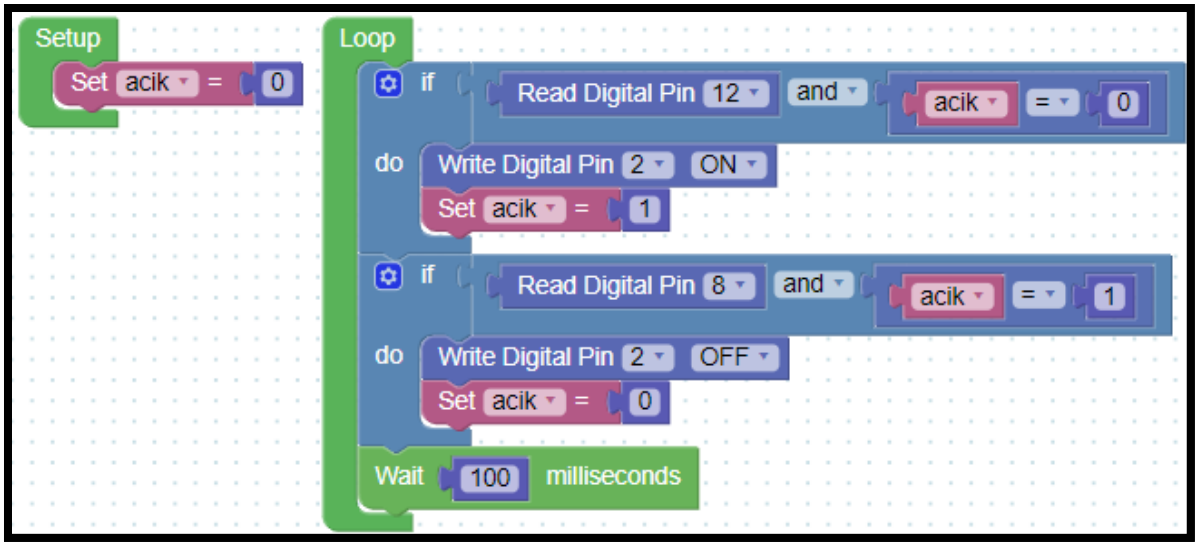
Değiştir butonu ile farklı renkler üretme etkisi bütün notalar için yapılmalıdır. Değiştir butonuna her basıldığında (beş farklı renk tepkisi olacak şekilde) do dışındaki notalar için üretilen renkler de değişmelidir.

4.2 İki Butonla Bir Led'i Kontrol Etme

Bu etkinlikte amaç iki buton kullanarak bir ledi kontrol etmektir. Bir butona basınca led yanmalıdır. Diğer butona basıldığında ise led sönmelidir.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

Bu etkinlik için aşağıdaki resimde görülen kod kullanılabilir.



Resim 118. Örnek Program

4.3 Butona Basınca Yanan Bir Daha Butona Basınca Sönen Led

Bu etkinlikte amaç bir buton ile bir ledi kontrol etmektir. Devrede bir buton ve bir led bulunmalıdır. Butona bir kere basınca led yanmalıdır. Bir daha basıldığında ise led sönmelidir. Butona bir daha basılırsa led yeniden yanmalıdır. Yani butona her bir basılışta led durum değiştirmelidir. Eğer led yanıksa sönmelidir; sönükse yanmalıdır.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

4. Bölüm-Dijital ve Analog Girdiler, Potansiyometre, Transistör ve Kondansatörler

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler "ArduinoBlocks" geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini/dijital pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler "VE/VEYA" kapısı içeren devreleri programlayabilir.
- Öğrenciler RGB led'leri içeren elektronik devreleri programlayabilir.

Bölümün Kazanımları:

- Dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Potansiyometre içeren elektronik devreler kurup programlayabilir.
- LDR içeren elektronik devreler kurup programlayabilir.
- PWM ile voltajı ayarlanabilen devreler kurup programlayabilir.
- Transistör içeren elektronik devreler kurup programlayabilir.
- Kondansatör ve akım ilişkisini açıklayabilir.
- Transistörü ve kondansatörü birlikte kullanan elektronik devreleri programlayabilir.

Bölümün Amacı:

Bu bölümün temel amacı, öğrencilerin gelen voltajı istenilen seviyelerde bölmek için gerekli kavramları öğrenmelerini, voltajı bölme için gerekli devre bağlantılarını yapmalarını ve "ArduinoBlocks" ortamında programlayabilmelerini sağlamaktır. Bölümün alt amaçları ise, öğrencilerin Potansiyometre, LDR, PWM, Transistör ve Kondansatör kullanımını öğrenmelerini ve elektronik devrelerde uygulayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, 3 adet led, 1 adet LDR, 1 adet 10K potansiyometre, 1 adet BC547, Direnç (1 adet 100 Ohm, 1 adet 330 Ohm, 1 adet 560 Ohm, 1 adet 1K Ohm, 3 adet 10 Kohm), 1 adet buton, 1 adet diyot, 1 adet 9V DC motor, kondansatör (1 adet 100 uF 16V, 1 adet 47 uF 16V, 1 adet 330 uF 16V), 1 adet 9V pil (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Elektronik devrelerde potansiyometre, LDR, PWM, transistör ve kondansatör kullanımının ve programlanmasının çeşitli şekillerde gösterilmesi.

Uygula: Elektronik devrelerde potansiyometre, LDR, PWM, transistör ve kondansatör kullanımının ve programlanmasının çeşitli şekillerde uygulanması.

Tasarla: El kurutma makinesinin, değil kapısının ve DC motorunun hızının ayarlanması için gerekli elektronik devrelerin ve program kodlarının tasarlama adımlarının oluşturulması.

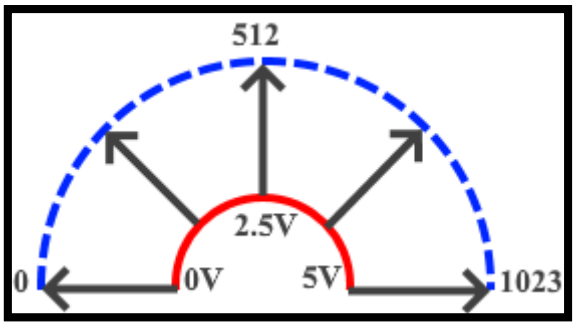
Üret: Tasarlanan el kurutma makinesi, değil kapısı ve hızı ayarlanabilen DC motoru elektronik devrelerinin oluşturulması ve program kodlarının yazılması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE ve UYGULA

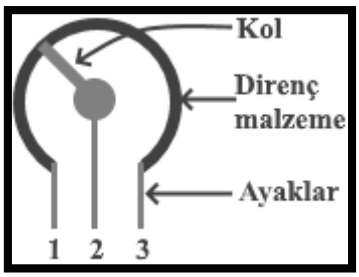
1.1 Gözle- Potansiyometre ile Analog Girdi

Bu bölümde potansiyometre ve analog girdi birlikte anlatılacaktır. Arduino Uno kartı üzerinde bulunan dijital girdi pinleri ile "0" (0V) ya da "1" (5V) girdisi yapılabilir ancak dijital pinlerden ara değerde girdiler mümkün değildir. Fakat bazı durumlarda "0V" (0) ile "5V" (1) arasındaki değerler de kullanılmak istenebilir. Bu yüzden Arduino Uno kart üzerinde analog giriş pinleri (A0-A5) bulunur. Analog pinler kullanılarak "0 ile 5V" arasında kalan değerler de ayırt edilebilir/kullanılabilir. Bu derste analog sinyalin ne olduğu işlenmeyecektir. Fakat yine de Arduino Uno kart üzerinde bulunan analog pinlerin tam olarak analog olmadığı bilinmelidir. Analog pinler kullanılarak 0V ile 5V arasındaki değerler 1024 parçaya ayrılır. "0V" 0'a karşılık gelir. "5V" ise 1023 değerine karşılık gelir. "0V ile 5V" arasındaki değerler ise 0 ile 1023 arasında bir değere dönüştürülür. Örneğin 2.5V'un alacağı değer 512'dir. Bu durum aşağıdaki resimde gösterilmiştir.



Resim 119. Dijital-Analog Dönüşümü

Şu ana kadarki etkinliklerde sadece dijital girdi (yani yalnızca 0V ve 5V) ile ilgili örnekler yapılmıştır. Ancak birçok devre elemanından gelen değer analog pinlerden okunur çünkü bu elemanlar 0V ile 5V arasındaki ara değerleri de üretirler. Potansiyometre bu elemanlardan bir tanesidir. Potansiyometrenin iki görevi bulunur. Bu görevler direnç ayarlama (ayarlı direnç) ve voltaj bölmedir. Aşağıdaki resimde potansiyometrenin içyapısı gösterilmiştir. Potansiyometrenin 10K Ohm olduğu düşünülsün. Potansiyometre ayarlı direnç olarak kullanılmak istenirse 1 ve 2 numaralı bacakları devreye bağlanır. Kol 1 numaralı bacağa yaklaştırıldıkça direnç azalır ve 0 Ohm değerine kadar iner. Kol 3 numaralı bacağa yaklaştırıldıkça direnç artar ve 10K Ohm değerine kadar çıkar. Aynı işlem 2 ve 3 numaralı bacaklarla da yapılabilir. Fakat bu durumda kol 3 numaralı bacağa yaklaştıkça direnç azalacaktır ve 1 numaralı bacağa yaklaştıkça direnç artacaktır. Bu durumun nedeni bağlı iki bacak arasındaki direnç oluşturan malzemenin miktarıdır. Miktar arttıkça direnç artar ve tersinde ise azalır.



Resim 120. Potansiyometre İç Görünüm

Potansiyometre voltaj bölücü olarak kullanılmak istenirse bir güç kaynağının (5V olduğu varsayılsın) "+" ucu 1 numaralı bacağa, "-" (GND) ucu ise 3 numaralı bacağa bağlanır. Çıktı değeri 2 numaralı bacaktan alınır. Potansiyometrenin kolu 1 numaralı bacağa yaklaştıkça çıktı değeri 0V'a doğru iner. Potansiyometrenin kolu 3 numaralı bacağa yaklaştıkça çıktı değeri güç kaynağından verilen voltaj (5V) değerine doğru ilerler. Böylece potansiyometre kullanılarak 5V'luk bir güç kaynağı kullanıldığında 0V-5V arası voltaj değerleri elde edilir. Aşağıdaki resimde 10K Ohm bir potansiyometre görülmektedir. Potansiyometrenin 10K olduğu üst kısmında bulunan yazı ile belirtilmiştir.

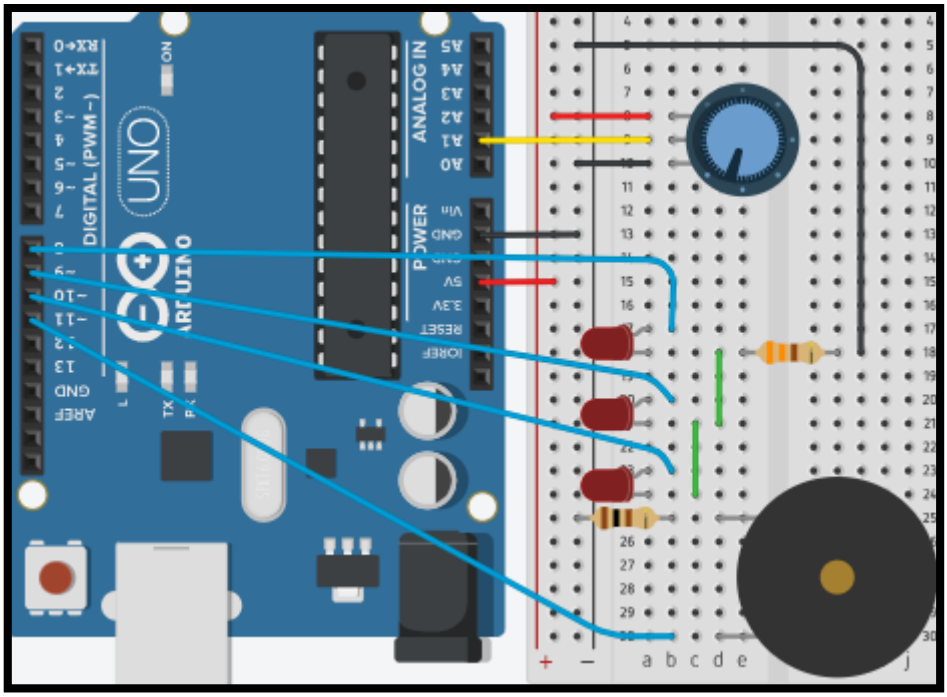
Resim 121. Potansiyometre

1.1 Uygula- Potansiyometre ile Ses Ayarı

Malzeme Listesi:

- Arduino Uno
- Breadboard
- 3 adet led
- 1 adet 10K potansiyometre
- 1 adet 330 Ohm direnç
- 1 adet 100 Ohm direnç
- 1 adet aktif buzzer

Bu etkinlikte amaç potansiyometreden gelen gerilim değerine göre led yakıp buzzerdan ses üretmektir. Potansiyometreden gelen gerilim değeri çok düşük olduğunda yan yana bulunan ledlerden hiçbiri yanmayacak; gelen gerilim değeri biraz artırıldığında sadece ilk led yanacak; gerilim değeri biraz daha artırılınca ilk iki led yanacak; son olarak potansiyometreden gelen gerilim değeri yüksek olduğunda üç led'in üçü de yanacaktır ve buzzerdan bir saniye aralıklarla uyarı sesi çalınacaktır. Bu görev için aşağıdaki resimde görülen devre kullanılabilir.

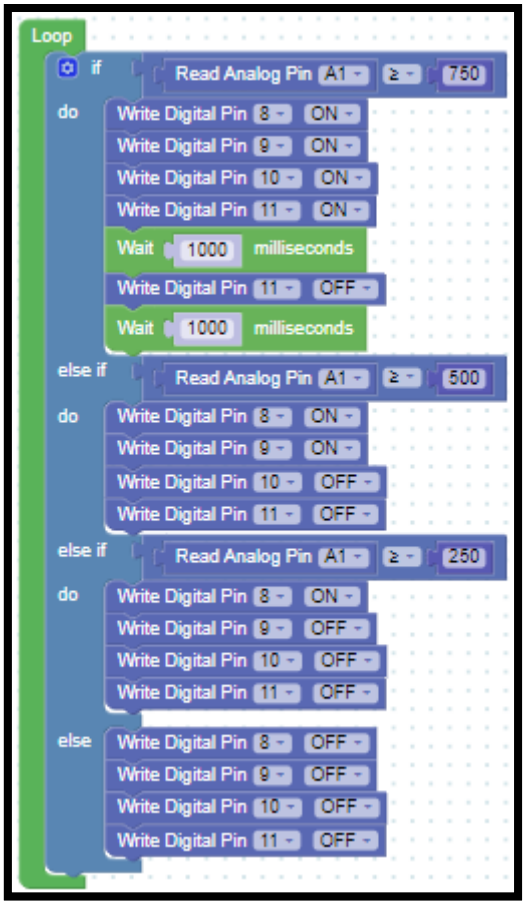


Resim 122. Örnek Devre Şeması

Bu görev için A1 analog pininden gelen değer okunmalıdır. ArduinoBlocks ortamında Analog pinlerden değer okumak için "Input/Output" blok başlığında bulunan "Read Analog Pin" bloğu kullanılır. Bu blok aşağıdaki resimde gösterilmiştir.

Resim 123. Analog Değer Okuma Bloğu

Analog pinden gelen değer 0-1023 arasında değişiklik gösterir. Eğer bu değer 750'den büyükse "üç" led yanar ve buzzer bir saniye aralıklarla "bip" sesi çıkarır. Eğer bu değer 500 ile 750 arasında ise ilk "iki" led yanar. Eğer bu değer 250 ile 500 arasında ise "ilk" led yanar. Eğer bu değer 0 ile 250 arasındaysa "hiçbir" led yanmaz. Bu görev için aşağıdaki resimde görülen kod kullanılabilir.

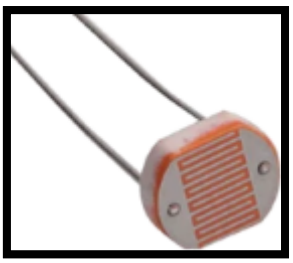


Resim 124. Örnek Program

Bu etkinlikte rehber öğretmen devreyi öğrencilere göstererek kurar; kodu öğrencilere anlatarak yazar, kodu çalıştırır ve potansiyometrenin farklı değerleri için kodun çalışmasını öğrencilerin gözlemlemesini sağlar. Ardından öğrencilerin aynı devreyi kurmalarını ve kodu yazarak çalıştırmalarını sağlar.

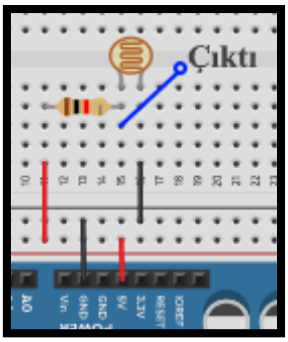
1.2 Gözle- LDR

LDR (Light Dependent Resistor) *kelime anlamı olarak ışık duyarlı direnç demektir*. Bazen LDR'ye foto direnç de denilmektedir. LDR'nin üretiminde yarı iletken maddeler kullanılmaktadır. LDR'nin üzerine ışık düştükçe direnci azalır; karanlık ortamlarda ise direnci artar. LDR'nin karanlıktaki direnci 1M Ohm (1M Ohm= 1.000.000 Ohm) seviyesine çıkarken; aydınlıktaki direnci 10 Ohm seviyesine kadar düşebilir. Bu değerler kullanılan LDR'nin cinsine göre değişiklik gösterebilir. Tipik bir LDR aşağıdaki resimde görüldüğü gibidir.



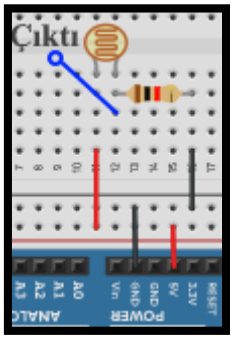
Resim 125. LDR

LDR ışık sensörü olarak kullanılabilir. Örneğin karanlıkta otomatik olarak çalışan lambalarda ve elin varlığını algılayıp otomatik işlemler gerçekleştiren cihazlarda kullanılabilir (aslında burada gerçekleşen işlem elin varlığını algılamaktan ziyade LDR'nin üzerine düzen ışığın el tarafından engellenmesidir). Ortamdaki ışığın yoğunluğuna göre LDR'den okunan direnç değeri düşünülerek devreler tasarlanabilir. LDR farklı amaçlar için kullanılabilir fakat potansiyometrede olduğu gibi voltaj/gerilim bölücü olarak da kullanılabilir. Bunun için LDR ile bir direnç (burada 1K Ohm kullanılmıştır) birbirine aşağıdaki resimde görüldüğü gibi bağlanır. Işık miktarı arttıkça LDR'nin direnci azalır ve böylece Çıktı gerilimi düşer. Işık miktarı ne kadar artarsa gerilim o kadar düşer. Işık miktarı ne kadar azalırsa gerilim o kadar artar.



Resim 126. LDR Bağlantı 1

Direnç ile LDR'nin yerleri değiştirilerek aşağıdaki resimde görüldüğü gibi bir devre de kurulabilir. Bu devrede ışık miktarı arttıkça LDR'nin direnci azalır ve Çıktı gerilimi artar (toplam direnç azalacağından devreden geçen akım miktarı artar ve direnç üzerine düşen gerilim artar). Işık miktarı azaldıkça çıktı gerilimi azalır.



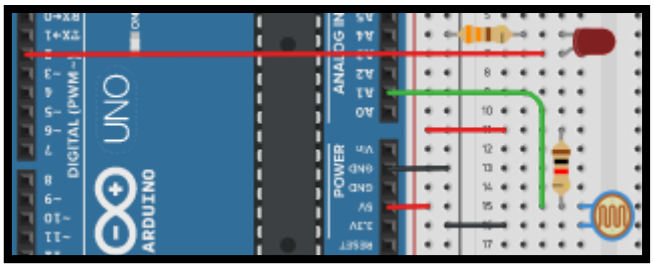
Resim 127. LDR Bağlantı 2

1.2 Uygula- LDR ile Karanlıkta Yanan Led

Malzeme Listesi:

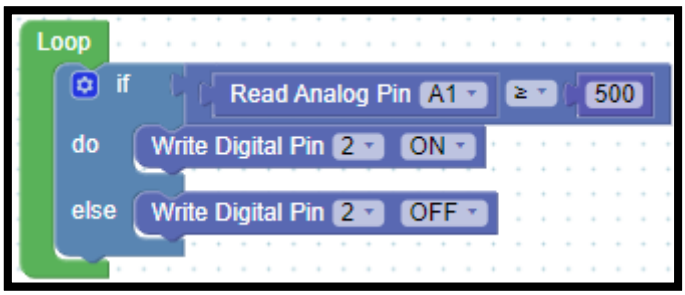
Arduino Uno
Breadboard
1 adet 1K Ohm direnç
1 adet 330 Ohm direnç
1 adet LDR

Bu etkinlikte amaç hava kararınca yanan led için devre tasarlayıp kodunu yazmaktır. Burada aşağıdaki resimde görülen devre kullanılabilir. Bu devrede LDR ve 1K Ohm direnç gerilim bölücü olarak kullanılmıştır. A1 analog pinine verilen gerilim, direnç ile LDR arasından alınmaktadır. Karanlık arttıkça LDR'nin direnci artar ve A1 pinine giden gerilim değeri de artar.



Resim 128. Örnek Devre Şeması

Devredeki led, 330 Ohm direnç üzerinden Arduino'nun 2 numaralı dijital pinine bağlanmıştır. A1 analog pininden gelen değer belirli bir değerin üzerine çıktığında (yani hava yeteri kadar karardığında) 2 numaralı dijital pine ON değeri gönderilirse led yanar. Sonuç olarak hava belirli bir karanlık seviyesine gelince led yanmış olur. Bu iş için gerekli kod aşağıdaki resimde görüldüğü gibidir.



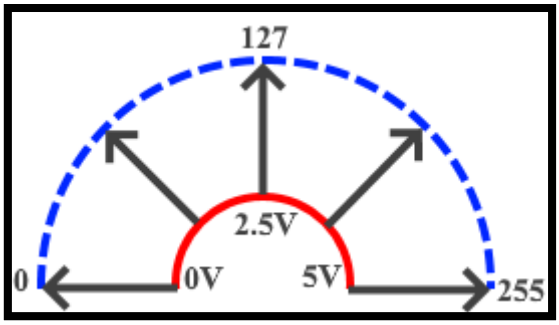
Resim 129. Örnek Program

Dikkat

Yukarıdaki kodda led'in yanması için eşik değer 500 olarak alınmıştır. Bu değer ortamdaki ışık miktarına göre değişiklik gösterebilir. Öğrenciler kendi ortamlarına göre eşik değeri belirlemelidirler.

1.3 Gözle- PWM ile Çıktı Üretme

Dijital pinlerden verilen çıktı değerleri daha önce de söylenildiği gibi ya "0" (0V) ya da "1" (5V) değerindedir. Fakat analog girdi de olduğu gibi bazı durumlarda "0V" ve "5V" arasındaki değerler çıktı olarak kullanılmak istenebilir. Örneğin bir motorun hızını belirlemek için ve RGB ledin farklı renklerde yanması için ara değerlere ihtiyaç duyulur. Arduino Uno kartta ara değerleri elde etmek için PWM denilen bir yöntem kullanılır. PWM yönteminin detayları burada anlatılmayacaktır. Öğrencilerin PWM yöntemi ile dijital pinlerden "0 ile 255" arasında 256 farklı değerin gönderilebileceğini bilmesi yeterlidir. Böylece "0V" 0'a, "5V" ise 255'e karşılık gelir. "0V" ile "5V" arasındaki değerler "0 ile 255" arasındaki değerlere dönüştürülür. Örneğin 2.5 volt 127 değerine karşılık gelir. Bu durum aşağıdaki resimde gösterilmiştir.



Resim 130. Dijital-PWM Dönüşümü

Arduino'nun 3, 5, 6, 9, 10, 11 dijital bacakları PWM çıktısı sağlayabilir. PWM pinlerinin yanında aşağıdaki resimde görüldüğü gibi tilde (~) işareti vardır.

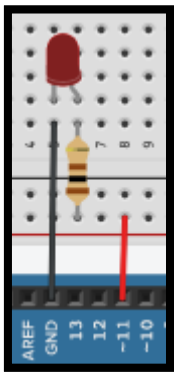
Resim 131. PWM Pinleri

1.3 Uygula- PWM ile Çıktı Üretme

Malzeme Listesi:

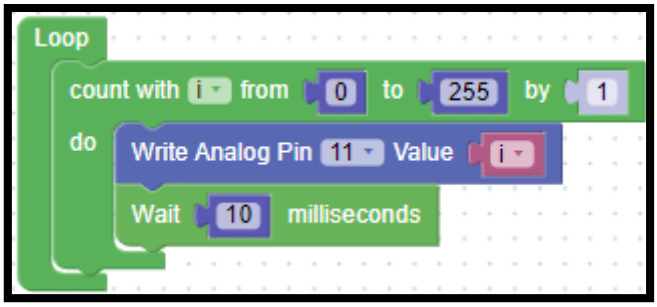
Arduino Uno
Breadboard
1 adet led
1 adet 100 Ohm direnç

Bu etkinlikte PWM pinleri kullanılarak yaydığı ışık miktarı sürekli değişen bir led devresi yapılacaktır. Bu devrenin şeması aşağıdaki gresimde görüldüğü gibidir.



Resim 132. Örnek Devre Şeması

Devrenin çalışması için gerekli kod aşağıdaki resimde görüldüğü gibidir. Aşağıdaki kod çalıştırıldığında 11 numaralı PWM pinine 10 milisaniye aralıklarla 0'dan başlayarak 255'e artacak şekilde değerler gönderilmektedir. 255 değerin ardından yeniden 0'dan başlanarak değerler gönderilmeye başlanır. Böylece led'in parlaklığı 255 değerine kadar sürekli artar ve gelen 0 değeriyle birlikte söner. Bu artarak yanıp sönme işlemi sürekli tekrar eder.



Resim 133. Örnek Program

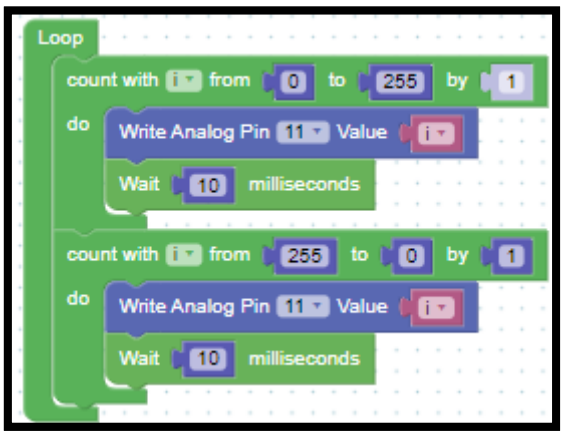
Rehber öğretmen devreyi öğrencilere göstererek kurar ve kodu öğrencilere göstererek açıklar. Ardından öğrencilerden aynı devreyi kurup, kodunu yazıp çalıştırmalarını sağlar.

1.3 Uygula- Yavaşça Yanıp Yavaşça Sönen Led

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet led
1 adet 100 Ohm direnç

Bir önceki etkinlikte led yavaş yavaş yanıp aniden söner. Sonra yeniden yavaş yavaş yanıp aniden söner ve bu sürekli tekrarlanır. Bu etkinlikte led'in yavaş yavaş yandıktan sonra yavaş yavaş sönmesi istenmektedir (aniden olmamalıdır). Bu iş için aşağıdaki resimde verilen kod kullanılabilir.



Resim 134. Örnek Program

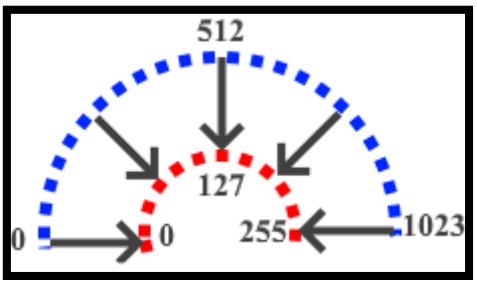
1.4 Gözle ve Uygula- Potansiyometre ile Led'in Parlaklığını Ayarlama

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet 10K potansiyometre
1 adet led
1 adet 100 Ohm direnç

Bu etkinlikte analog girdi ve PWM pinleri birlikte kullanılarak potansiyometre ile bir ledin parlaklığı kontrol edilecektir. Potansiyometreden gelen analog değere bağlı olarak PWM pinine bir değer gönderilecek ve böylece PWM pinine bağlı olan ledin parlaklığı ayarlanacaktır. Fakat analog girdi 0-

1023 arasında değer alırken PWM pini 0-255 arasında bir değer çıktı verebilmektedir. 0-1023 arasındaki değerin 0-255 arasındaki değere çevirmek gerekecektir. Bu durum aşağıdaki resimde gösterilmiştir.

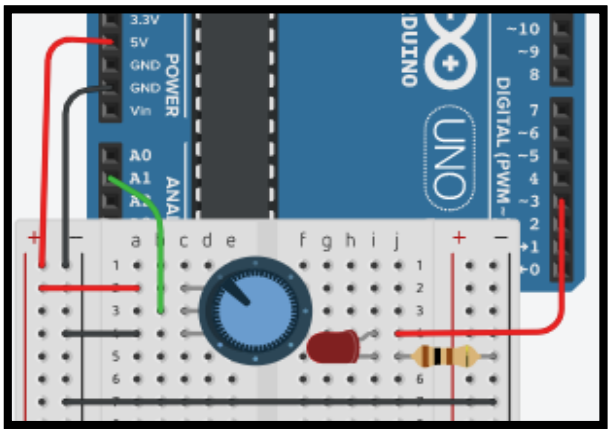


Resim 135. PWM Analog Dönüşümü

Bu iş için "ArduinoBlocks" ortamında "Math" blok başlığı altında "map" bloğu bulunmaktadır. Bu blok aşağıdaki resimde görüldüğü gibidir. Map bloğunun 5 parametresi bulunur. Birinci parametre dönüştürülecek değeri gösterir. Aşağıdaki komutta bu kısım boş bırakılmıştır. İkinci ve üçüncü parametre ilk değerin bulunduğu aralığı verir. Aşağıdaki resimde görüldüğü gibi bu aralık 0-1023 aralığıdır. Dördüncü ve beşinci parametre ise ilk değerin dönüştürüleceği aralığı verir. Aşağıdaki resimde görüldüğü gibi değerin dönüştürüleceği aralık 0-255 aralığıdır. Yani ilk parametre yerine girilecek değer 0-1023 aralığından 0-255 aralığına dönüştürülecektir.

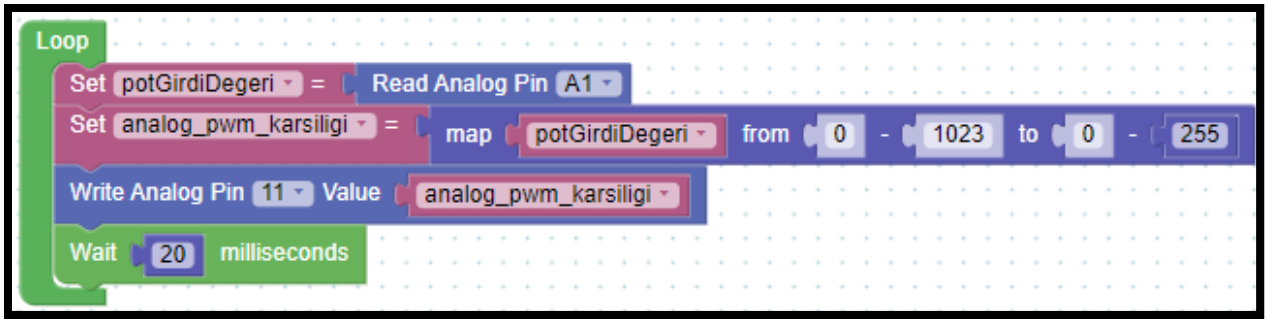
Resim 136. Map Bloğu

Bu etkinlik için aşağıdaki resimde verilen devre kullanılabilir. Görüldüğü gibi potyansiyometreden gelen değer A1 analog girdi pinine yönlendirilmiştir ve 3 numaralı PWM pininden çıkan değer ise led'e yönlendirilmiştir.



Resim 137. Örnek Devre Şeması

Aşağıdaki resimde verilen kodlar kullanılarak istenilen görev yerine getirilebilir. Aşağıdaki kod potansiyometreden gelen değeri 0-255 aralığına dönüştürüp 11 numaralı pinde bulunan led'e göndermektedir.



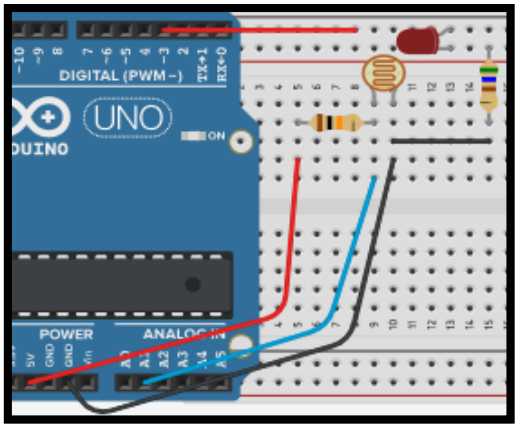
Resim 138. Örnek Program

1.4 Uygula- LDR ile Işık Seviyesi Ayarlama

Malzeme Listesi:

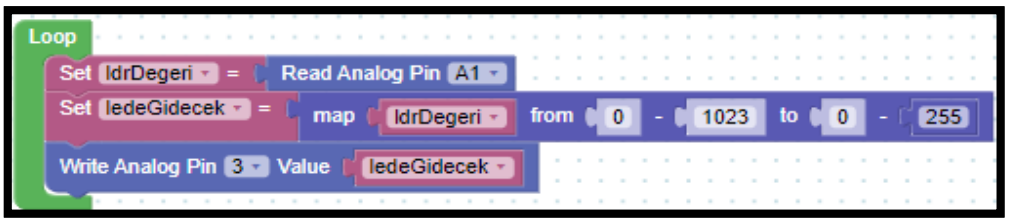
Arduino Uno
Breadboard
1 adet 1K Ohm direnç
1 adet 560 Ohm direnç
1 adet led
1 adet LDR

Bu etkinlikte amaç ortam karardıkça led'in parlaklığını artıran bir devre tasarlamaktır. Bu iş için aşağıdaki resimde verilen devre kullanılabilir. Karanlık arttıkça A1 analog pin'e LDR üzerinden gönderilen değer artar. Aydınlık arttıkça A1 analog pin'e LDR üzerinden gönderilen değer azalır. Bu değer dönüştürülüp 3 numaralı dijital pin'e PWM değeri olarak gönderilirse ortam ışığına bağlı olarak led'in parlaklığı ayarlanabilir.



Resim 139. Örnek Devre Şeması

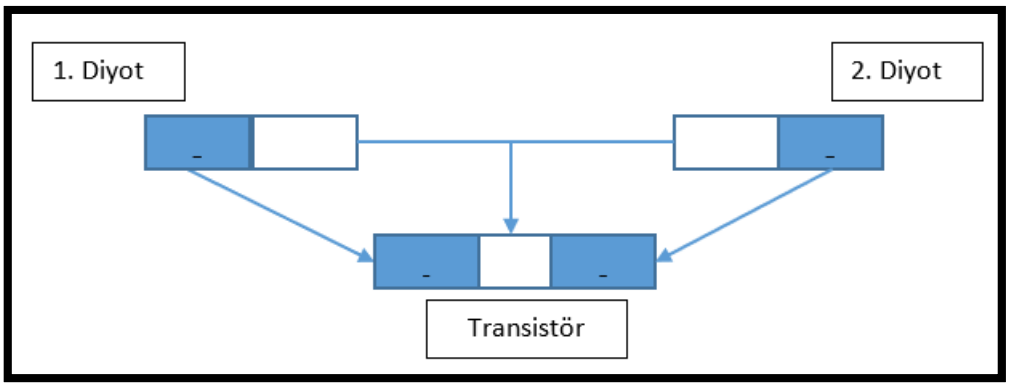
Bu iş için aşağıdaki resimde görülen kod kullanılabilir.



Resim 140. Örnek Program

1.5 Gözle- Transistör

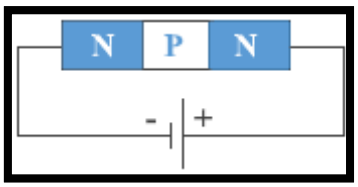
Bilgisayar da dâhil olmak üzere günümüzde kullanılan elektronik cihazların oluşturulmasında en büyük rol yarı iletken maddelerin kullanılması sonucu ortaya çıkan diyotlar ve transistörlerdir. Bir bilgisayarın içerisinde ve hatta uygulamalarda kullandığımız Arduino kartın üzerindeki çiplerin birçoğunun içerisinde çok küçük boyutlarda transistörler barındıran devreler bulunur. Transistörler de diyotlar gibi N ve P tipi yarı iletken maddelerin bir araya getirilmesi ile oluşturulur. Transistörler diyottan farklı olarak N-P-N veya P-N-P olacak şekilde üç bölümden oluşur. Daha önceden de belirtildiği gibi diyotlar P-N olacak şekilde iki kısımdan oluşmaktadır. Temelde bir transistör iki diyotun aynı uçlarının birbirine yapıştırılması şeklinde düşünülebilir. Transistörün içyapısı aşağıdaki resimde görülmektedir.



Resim 141. Transistör İç Yapısı

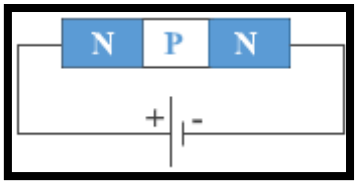
Yukarıdaki resimde iki diyotun N kısımları birbirine birleştirilerek oluşturulan PNP bir transistör yer almaktadır. Gerçekte bir transistör iki diyotun birleştirilmesi ile elde edilmez. Bu görsel transistörün çalışma şeklini açıklamak için oluşturulmuş bir benzetmedir. Benzer bir işlem iki diyotun P kısımları birleştirilerek de yapılabilir. Bu şekilde yapılan birleştirme sonucu NPN transistörler oluşturulur. PNP ve NPN tipi transistörlerin temel çalışma prensipleri aynı olmasına rağmen ters yükler ile çalıştırılan iki ayrı transistördür.

Burada ağırlıklı olarak NPN transistörler üzerinden konu anlatılacaktır. Aşağıdaki resimde bir güç kaynağının + ucu NPN transistörün N bacaklarından birine, "-" ucu ise diğerine bağlanmıştır. Bu resimde devreden akım geçmez. Birinci diyot gibi düşünülebilecek olan NP bloğu (sol diyot) akım geçirmek istese bile ikinci diyot olan PN (sağ diyot) bloğu akım geçirmez.



Resim 142. Bir Pil ile Transistör Bağlantısı

Aşağıdaki resimde görüldüğü gibi güç kaynağının yönünü ters çevrilip bağlantılar yeniden yapılırsa yine bir değişiklik olmayacaktır. Devreden akım geçmez. Bu sefer PN bloğu (sağ diyot) akım geçirmek istese bile NP bloğu (sol diyot) akım geçirmez.

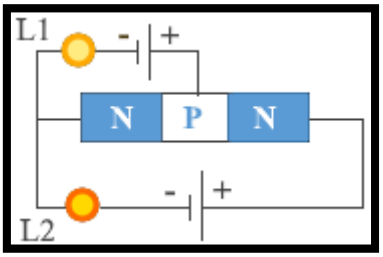


Resim 143. Bir Pil ile Transistör Ters Bağlantısı

Dikkat

Aslında yukarıdaki ifadeler tam olarak doğru değildir. Örneğin Esaki osilatör devresinde transistörler yukarıdaki resimdeki gibi bağlanır ve yeterli gerilim uygulandığında NP arasında bir akım meydana gelir. Pedagojik açıdan bu durumun anlatılmaması daha uygun olacaktır.

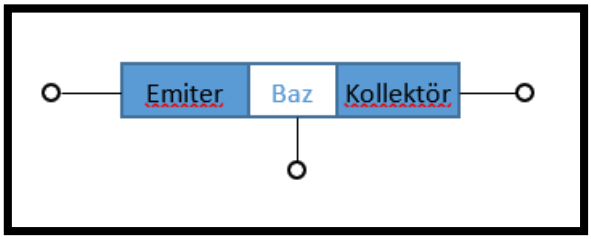
Yukarıdaki iki resimde gösterilen devrede de P kısmı güç kaynağının yönüne bağlı olarak ya soldan sağa olabilecek akım geçişini ya da sağdan sola olabilecek akım geçişini engellemektedir. Bu engeli ortadan kaldırmak için P kısmını iletken hale getirmek gerekir. Bunun için NP kısmına (soldaki diyot) bir gerilim uygulanması gerekmektedir (silisyum için yaklaşık 0.7 V). Bu durum aşağıdaki resimde gösterilmiştir. Bu kısma gerilim uygulandığında bir diyot gibi davranıp akım iletecektir ve L1 lambası yanacaktır. P kısmının gösterdiği aşırı direnç ortadan kalktığında artık transistörün iki ucu arasında da elektron akışı yani akım meydana gelir. Böylece L2 lambası da yanar. L2 lambası L1 lambasına göre daha parlak yanar. Burada dikkat edilmesi gereken iki nokta vardır. İlk olarak transistör bir anahtar şeklinde davranmıştır. P kısmına herhangi bir gerilim uygulanmadığında transistörün sol ve sağ bacağı arasında bir akım meydana gelmez. P kısmına uygun bir gerilim uygulandığında ise transistörün iki bacağı arasında akım oluşmuştur. İkinci olarak transistör bir yükselteç gibi davranmıştır. P kısmı üzerinden geçen küçük akım sağdaki N bacağı üzerinden geçen daha büyük bir akım olarak yükseltilmiştir. Bunun sonucu olarak L2 lambası L1 lambasına oranla daha parlak yanmıştır. Aşağıdaki resimde iki pil ile transistör bağlantısı görülmektedir.



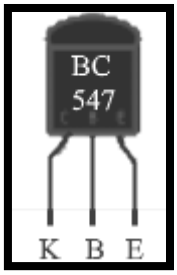
Resim 144. İki Pil ile Transistör Bağlantısı

Transistörlerin N ve P kısımları yaptıkları işe göre emiter, baz ve kollektör olarak isimlendirilmiştir. Bu durum aşağıdaki resimde gösterilmiştir. Emit İngilizce göndermek/yollamak ve collect ise toplamak anlamına gelmektedir. Bir baz gerilimi uygulandığında emiterden kollektöre elektron gönderilmektedir yani emiterden kollektöre doğru bir akım oluşur. Basit anlamda bir transistörün emiter ucundan kollektör ucuna olan akım geçişi baz ucu ile kontrol edilir. Burada transistörün anahtar görevi vurgulanmıştır. Transistörün ikinci görevi yükselteç olmasıdır. Bazdan geçen akım miktarına göre kollektörden daha yüksek miktarda akım elde edilir. Örneğin bir mikrofon aracılığıyla yapılan konuşmanın hoparlörlerde çok daha yüksek bir ses olması transistörler aracılığı ile gerçekleşir. Transistörler birçok devrede akım yükseltici görevini üstlenirler. Bu derste transistörün anahtar görevi daha çok kullanılacaktır.

Piyasada çok farklı özelliklere sahip farklı transistörler bulunmaktadır. Bu derste BC547 isimli transistör kullanılacaktır. Bu transistörün şekli aşağıdaki resimlerde olduğu gibidir. Sol bacak kollektör, orta bacak baz ve sağ bacak emiter bacağıdır.



Resim 145. Transistör İç Yapısı



Resim 146. BC 547 Transistör

Dikkat

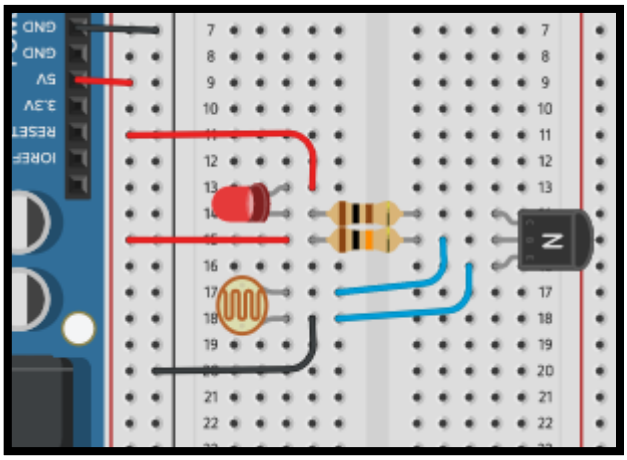
Transistörlerin bacak sıralamaları değişmektedir. Başka bir transistörde bacak sıralaması farklı olabilir. Transistörün kataloğundan bacakları bulunabilir.

1.5 Uygula- Transistör ile Karanlıkta Yanan LED

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet BC547
1 adet led
1 adet 100 Ohm direnç
1 adet 10 K Ohm direnç
1 adet LDR

Bu etkinlikte amaç transistör kullanarak karanlık olduğunda yanan bir led devresi tasarlamaktır. Bu devre aşağıdaki resimde gösterilmiştir.



Resim 147. Örnek Devre Şeması

Devrede Arduino sadece 5V güç kaynağı olarak kullanılmıştır. Arduino yerine herhangi bir 5V güç kaynağı da kullanılabilir. Rehber öğretmen yukarıdaki devreyi öğrencilere göstererek kurar ve çalışma mantığını öğrencilere anlatır. Devrenin çalışma mantığı şu şekildedir:

Bu devrede transistör anahtar olarak kullanılmaktadır. Aydınlık bir ortamda LDR'nin direnci çok düşük olduğundan 10K Ohm direnç üzerinden gelen akım transistörün baz bacağını tetikleyecek değere ulaşamadan LDR üzerinden GND'ye iletilir. Fakat karanlık bir ortamda LDR'nin direnci artacağından transistörün baz bacağında 0.7 V eşik değerini geçecek şekilde gerilim oluşur ve bu gerilim emiter ile kollektör arasında bir akım oluşmasına dolayısı ile kollektör bacağına bağlı olan ledin yanmasına sebep olur.

Öğretmen devrenin çalışmasını öğrencilere anlattıktan sonra aynı devreyi öğrencilerin kurmasını ister.

1.6 Gözle ve Uygula- Transistör ile DC Motor Sürme

Malzeme Listesi:

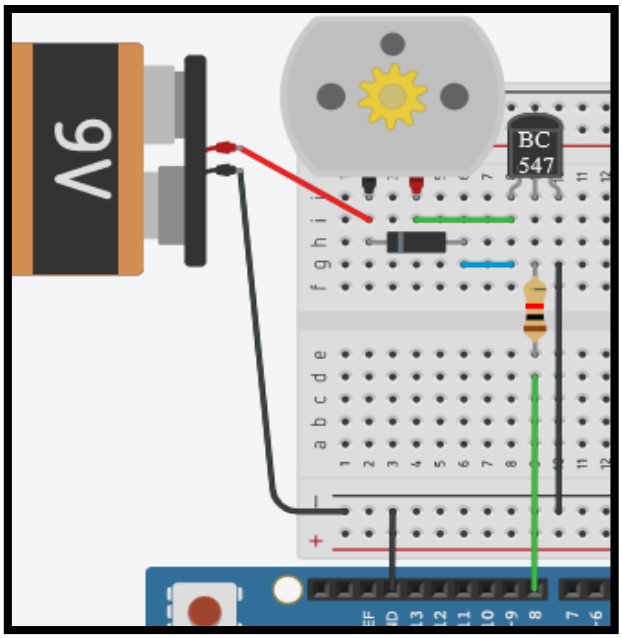
Arduino Uno
Breadboard
1 adet BC547
1 adet 9V DC motor
1 adet diyot,
1 adet 1K direnç,
1 adet 9V pil

Bu etkinlikteki amaç BC547 kullanarak öğrencilere basitçe DC motor sürmeyi göstermektir. DC motor 1 saniye çalışıp 1 saniye duracaktır ve bu işlemi sürekli tekrarlayacaktır.

Bu derste DC motorlar konu olarak odak noktası değildir. Öğrencilerin DC motorların uçlarına uygun miktarda gerilim uygulandığında döndüğünü bilmeleri yeterlidir. Arduino kartının I/O pinlerinden maksimum 40 mA akım kullanılabilir. Fakat bazı durumlarda DC motorlar 40 mA'den daha fazla akım çeker. Bu da Arduino kartın limitlerini aştığından ona zarar verebilir. DC motor gibi fazla akım çeken devre elamanlarını Arduino kart ile programlamak için transistörler anahtar olarak kullanılabilir. Aşağıdaki devre böyle bir uygulamaya örnektir. Bu devre şu şekilde çalışır:

Bu devrede motorun akımı 9V pil üzerinden sağlanmaktadır. Pilin + ucu motora, motorun diğer ucu ise BC547'nin kollektör bacağına bağlanmıştır. BC 547'nin baz bacağı bir direnç üzerinden Arduino'nun 8 numaralı dijital pinine bağlanmıştır. 9V pilin – bacağı ile Arduino'nun GND bacağı birleştirilerek BC 547'nin emiter bacağına bağlanmıştır. Bu durumdayken 8 numaralı pinden ON değeri gönderildiğinde

BC547 anahtar olarak görev görür ve 9V pilden aldığı akım kollektör bacağından akmaya başlar. Böylece motor döner. Devrede motora paralel olarak bağlı bulunan bir diyot bulunmaktadır. Bu diyotun görevi motordan kaynaklı oluşabilecek voltaj artışlarının transistöre ulaşmasını engelleyip transistörü korumaktır. Örnek devre şeması aşağıdaki resimde görülmektedir.



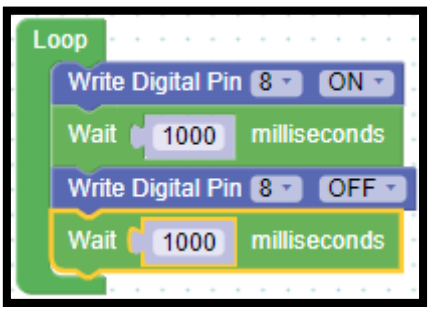
Resim 148. Örnek Devre Şeması

Dikkat

Burada kullanılacak olan direnç değeri motorun çalışması için önemlidir. Çok yüksek değerli direnç kullanıldığında motor çalışmaz, çok düşük değerli direnç kullanıldığında transistör yanar. Direnç değeri devrede kullanılacak olan 9V pilin özelliklerine göre değişiklik gösterebilir. Bu direnç değeri öncelikle 1K olarak denenmeli ardından çalışmazsa kademeli olarak 220 Ohm'a kadar düşürülmelidir.

Devrede kullanılan DC motor için araba kiti içerisindeki sarı motorların kullanılması önerilir.

Rehber öğretmen devreyi kurup çalışmasını öğrencilere anlattıktan sonra aşağıdaki kodu yazar ve öğrencilere göstererek kodu çalıştırır. Öğrenciler devrenin çalışmasını gözledikten sonra devreyi kurar ve kodu yazarak çalıştırır. Örnek program aşağıdaki resimde görülmektedir.



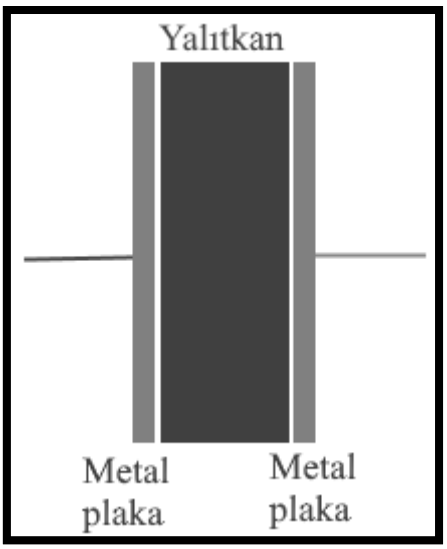
Resim 149. Örnek Program

Bütün öğrenciler görevi tamamladıktan sonra rehber öğretmen aşağıdaki soruyu öğrencilere sorar ve yanıtlar sınıfça tartışılır.

- Motorun bir ucu 8 numaralı pine bağlanıp diğer ucu GND'ye bağlansaydı ve yine aynı kod kullanılsaydı, DC motor aynı şekilde çalışmaz mıydı? Neden böyle bir yol tercih edilmiştir?

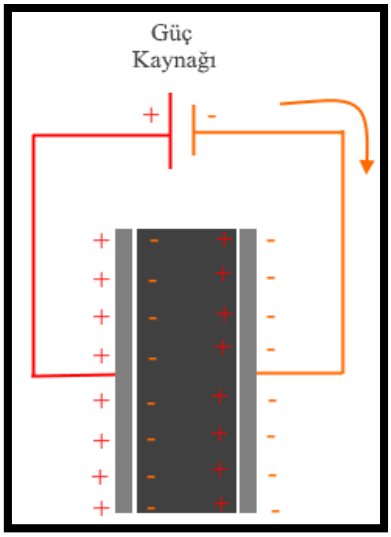
1.7 Gözle- Kondansatör ve Akım İlişkisi

Basit bir devre oluşturmak için en az üç temel unsura ihtiyaç duyulduğundan daha önce bahsedilmişti. Bunlar sırasıyla güç kaynağı, bağlantı kabloları ve devre elemanlarıdır. Kondansatör de direnç ve led gibi elektronik devrelerde kullanılan temel bir devre elemanıdır. Kondansatöre kapasitör de denilir. Bir kondansatör karşılıklı iki metal plaka ve plakalar arasında yer alan yalıtkan bir maddeden (bazen de boşluktan) oluşur. Kondansatörün iç yapısı aşağıdaki resimde görülmektedir.



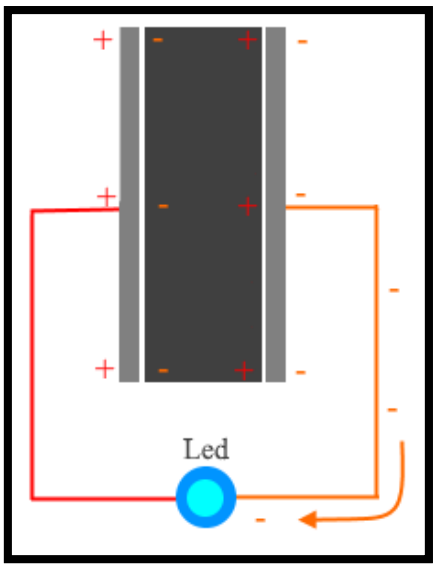
Resim 150. Kondansatör İç Yapısı

Kondansatörün plakaları arasında elektrik akımı yoktur. Yaptığı şey bir güç kaynağına bağlandığında plakaları arasında enerjiyi depolamaktır. Güç kaynağına bağlandığında bir plakasında artı (+) yükler, diğer plakasında ise eşit miktarda eksi (-) yükler toplanır. Bu durum aşağıdaki resimde gösterilmiştir.



Resim 151. Şarj Olma

Eksi (-) yükle yüklenen uçtaki elektronlar devrede buldukları bir yol üzerinden artı (+) yüklü uca ulaşmaya çalışır (aşağıdaki resimde devreye bağlanmış led üzerinden). Bu işi yaparken de devre üzerinde bir elektrik akımı oluşmasına neden olur. Kondansatör deşarj olduğunda bu akım kesilir.



Resim 152. Deşarj Olma

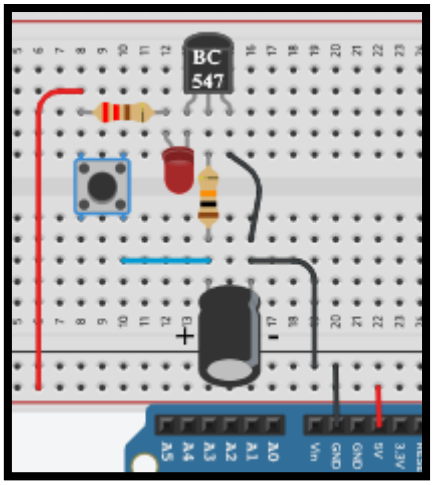
Kondansatörler bir çeşit şarj edilebilir pile benzer. Elektrik enerjisini depolayabilirler. Fakat pilden farklı olarak uzun süreli enerji sağlayamazlar. Hızlı bir şekilde şarj olurlar ve hızlı bir şekilde deşarj olurlar. Aslında bu bir avantajdır. Örneğin fotoğraf makinelerinin flaşları ani bir şekilde yüksek enerjiye ihtiyaç duyarlar çünkü kısa bir süre içerisinde yüksek aydınlık oluşturmaları gerekir. Flaşa bağlanan kondansatör öncelikle fotoğraf makinesine güç sağlayan pil tarafından doldurulur. Daha sonra flaşa basıldığı anda kondansatör üzerindeki elektrik enerjisi birden boşalarak ani yük akışı sağlar. Böylece anlık yüksek ışık elde edilmiş olur.

1.7 Gözle ve Uygula- Transistör ve Kondansatör Kullanıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet BC547
1 adet Buton
1 adet 10 Kohm direnç, 1 adet 330 Ohm direnç
1 adet 100 uF 16V kondansatör
1 adet led

Bu etkinlikteki amaç transistör ve kondansatörü kullanarak butona basıldığında ledi yakıp buton bırakıldığında yavaş yavaş ledin sönmesini sağlayan devreyi oluşturmaktır. Buna benzer bir devre arabalarda kullanılabilir. Hava karanlıkken arabanın kontağı kapatıldığında buton tetiklenip arabanın içerisindeki lambalar yanabilir ve bir müddet sonra arabanın ışıkları yavaş yavaş ve kendiliğinden sönebilir. Bu iş için kullanılabilecek devre aşağıdaki resimde gösterilmiştir.



Resim 153. Örnek Devre Şeması

Rehber öğretmen devreyi öğrencilere göstererek kurar ve çalışmasını anlatır. Ardından öğrencilerden aynı devreyi kurup çalıştırmasını ister. Devrenin çalışması aşağıda anlatılmıştır:

Anahtar açıkken kondansatör boştur ve transistörün bazına gerilim ulaşmaz. Butona basıldığında kondansatör şarj olur ve transistörün bazına yeterli gerilim ulaşır ve kollektör-emiter arasında akım oluşur. Böylece kollektör bacağına bağlı olan led'in ışığı yanar. Buton bırakıldığında kondansatör deşarj olmaya başlar ve baz bacağına bir müddet daha gerilim sağlamaya devam eder. Fakat bir müddet sonra kondansatördeki yük miktarı azalır ve baz bacağına yeterli gerilimi sağlayamaz. Baz bacağındaki akım azaldığı için kollektörden geçen akım da azalacaktır. En sonunda led ışımayı kesecektir.

1.7 Uygula- Led'in Yanma Süresini Değiştiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet BC547
1 adet Buton
1 adet 10 Kohm direnç, 1 adet 330 Ohm direnç

1 adet 330 uF 16V kondansatör
1 adet 47 uF 16V kondansatör
1 adet led

Yukarıda yapılan devrede butona basıldığında led bir müddet daha çalışıp sönmektedir. Bu etkinlikteki amaç led'in yanma süresini artırmak ve azaltmaktır. Rehber öğretmen öğrencilerden devrede değişiklik yaparak ledin yanma süresini değiştirmesini ister. Bunu yaparken öğrenciler Arduino'yu bir önceki devrede olduğu gibi güç kaynağı olarak kullanabilirler. Arduino'nun mikrodenetleyici özelliklerini kullanmamalıdırlar.

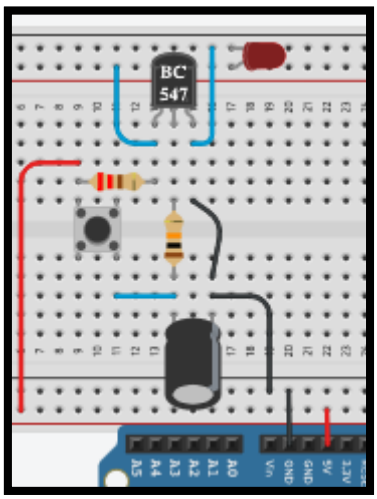
Bu görevi yerine getirmek için devredeki kondansatörü değiştirmek yeterli olur. Kondansatörün kapasitesi artırılırsa led'in yanma süresi artar. Kondansatörün kapasitesi azaltılırsa yanma süresi azalır.

1.7 Uygula- Butona Basıldığında Sönüp Bir Süre Sonra Yanan Led

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet BC547
1 adet direnç (BC547A için 10K, BC547B için 20K, BC547C için 30K),
1 adet 9V pil
1 adet 100 uF 16V kondansatör
1 adet led

Bu etkinlikteki amaç "1.7 Gözle ve Uygula- Transistör ve Kondansatörü Kullanıyorum" etkinliğindeki devrenin ters çalışanını yapmaktır. Butona basılmadığı durumda led yanmalıdır. Butona basıldığında ise bir müddet sönüp sonra yeniden yanmalıdır. Öğrenciler bu etkinlikte "1.7 Gözle ve Uygula- Transistör ve Kondansatörü Kullanıyorum" etkinliğindeki devre elemanlarının aynılarını kullanmak zorundadırlar. Yeni bir devre elemanı kullanamazlar veya Arduino'yu mikrodenetleyici olarak kullanamazlar. Bu sorunun örnek çözümü aşağıdaki resimde görüldüğü gibidir.



Resim 154. Örnek Devre Şeması

2. TASARLA ve ÜRET

2.1 Tasarla ve Üret- Sembolik El Kurutma Makinesi

Bu etkinlikte yapılan devrenin üç farklı görevi olacaktır. Birinci görevdeki amaç DC motora takılan pervane yardımıyla lavabolarda bulunan el kurutma makinesinin bir modelini oluşturmaktır. Kullanıcılar ellerini LDR'nin karşısına tuttuğunda pervane bir süre dönüp sonra duracaktır. Bu sayede el kurutma makinesi modellenmiş olacaktır. İkinci görevdeki amaç el kurutma makinesinin kaç defa kullanıldığını 5 adet led kullanarak ikilik tabanda bildirmektir. Kullanıcı ikilik tabanı ve yanan led'in 1 ve sönük led'in ise 0'a karşılık geldiğini bilmektedir. Devreye yerleştirilmiş olan butona basıldığında led'ler ikilik tabanın basamakları olacak şekilde kullanım sayısını bildirecektir. Kurutma makinesinin 4 defa çalıştırılmış olduğu var sayılsın. 4'ün ikilik tabandaki karşılığı 00100'dır (beş led olduğu için beş basamak kullanılmıştır). Bu durumda yan yana dizilmiş beş led'den sadece ortadakinin yanması gerekmektedir. Üçüncü görev ise kurutma makinesinin bakımı ile ilgilidir. 31 kullanımdan sonra kurutma makinesinin bakıma girmesi gerekmektedir. Bu yüzden 31. kullanımda kurutma makinesi belirli bir süre boyunca aralıklarla bip sesi çalar ve çalışmayı durdurur. Kullanıcı LDR'nin karşısına elini tuttuğunda pervane dönmez. Son olarak, son görevde amaç kurutma makinesinin çalışma sayısını sıfırlamaktır. İkinci bir butona basıldığında kurutma makinesinin çalışma sayısı 0'a düşürülür. Böylece kurutma makinesinin bakımını yapan teknisyen, bakım sonrasında bu butona basarak makinenin yeniden çalışır duruma gelmesini sağlamış olur.

Dikkat

5 led kullanıldığı için ledler en fazla 2^5=32, 32-1=31 kullanım bildirebilir. Bir led daha takılırsa 2^6=64, 64-1=63 kullanım bildirilebilir. İhtiyaca göre yeni led eklenebilir.

Sembolik el kurutma makinesi etkinliğini tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir. Aşağıdaki örnek süreç kesinlikle öğrencilere gösterilmemelidir. Rehber öğretmen sınıfta genel bir sorun görürse sınıfa açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Tanımlama: Öğrencilerin öncelikle sembolik el kurutma makinesi etkinliğinin neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrenciler gerekli işlemleri detaylı olarak maddeler hâlinde yazmalıdır.

Örnek:

i. LDR'nin önünde el algılandığında fan 2 saniye çalışmalıdır.

ii. Fanın çalışma sayısı sürekli sayılmalıdır.

iii. Fanın çalışma sayısı 2'lik tabana dönüştürülmelidir.

iv. 2'lik tabandaki sayılara göre ledler beş saniyeliğine yakılmalıdır.

v. Çalışma sayısı 31 olunca el algılandığında fan çalışmamalıdır, buzzer'dan ses çalınmalıdır.

vi. Butona basılınca fanın çalışma sayısı sıfırlanmalıdır.

Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Tanımlamada belirlenen her bir adımın gerçekleştirilebilmesi için gerekli fikirler belirlenmelidir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

Örnek:

i. LDR ile fanın önüne el geldiğini anlamak için LDR gerilim bölücü olarak kullanılır.

ii. Bir sayaç değişkeni oluşturulmalıdır ve her el algılandığında sayaç değişkenin değeri bir artırılmalıdır.

iii. El algılandığında fanın bağlı olduğu pine 2 saniye boyunca ON değeri verilmelidir.

   - Fan çalışırken mesafe sensörü ve LDR ile algılama yapılmamalıdır.

iv. Sayaç değişkeninin 2'lik tabandaki karşılığı hesaplanmalıdır. Bu işlem sayaç değişkeni sıfır olana kadar sayaç değişkeninden ikinin katları düzenli olarak çıkarılarak yapılabilir. Bu iş için yeni bir değişken oluşturulmalıdır.

v. Sayaç değişkeninin 2'lik tabandaki karşılığına denk gelen ledlere 2 saniye boyunca ON değeri verilmelidir sonrasında hepsi OFF yapılmalıdır.

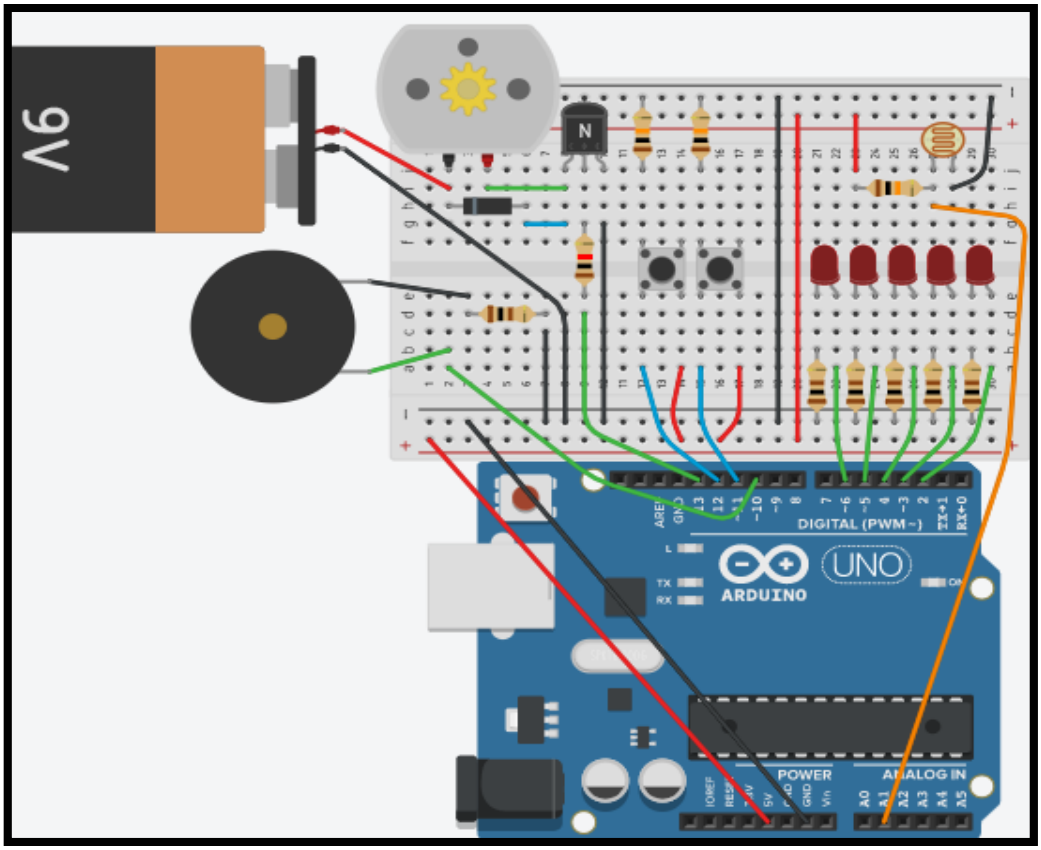
vi. Sayaç değişkeni 31 olduğunda buzzer aralıklarla ses çalmalıdır.

vii. Butona basıldığında sayaç değişkeni sıfırlanmalıdır.

2.1 Üret- Sembolik El Kurutma Makinesi

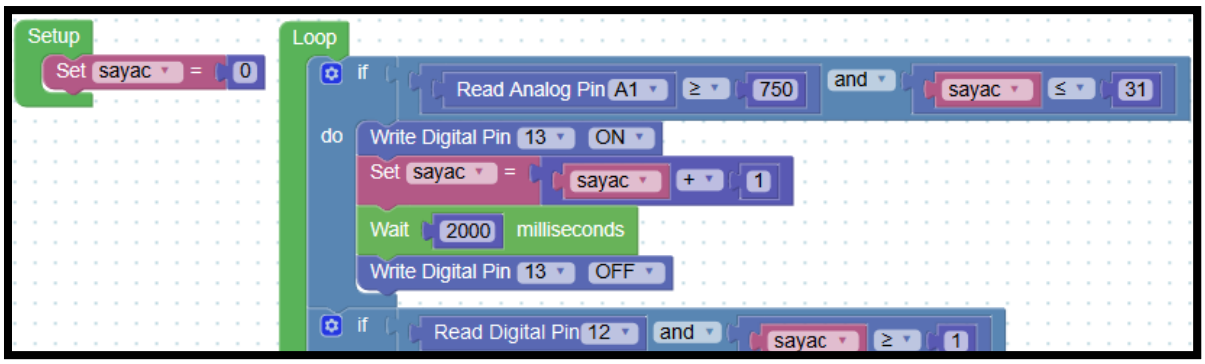
Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki devreye ve programa benzer bir içerik hazırlamaları beklenir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Öğrencilerin kurması gereken devreye örnek olarak aşağıdaki resimde görülen devre verilebilir.

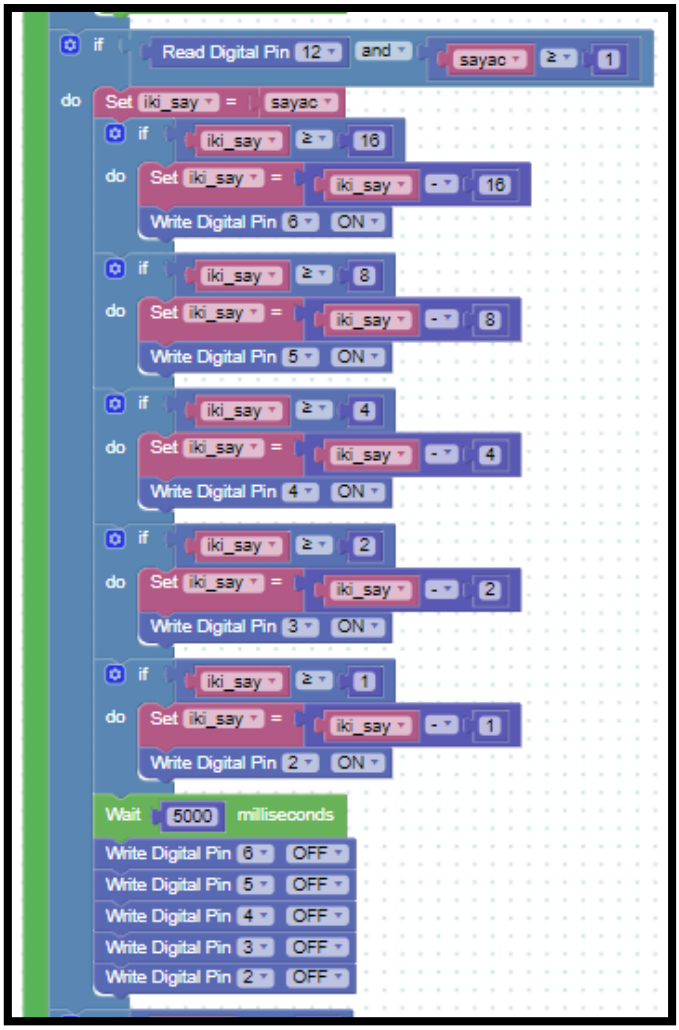


Resim 155. Örnek Devre Şeması

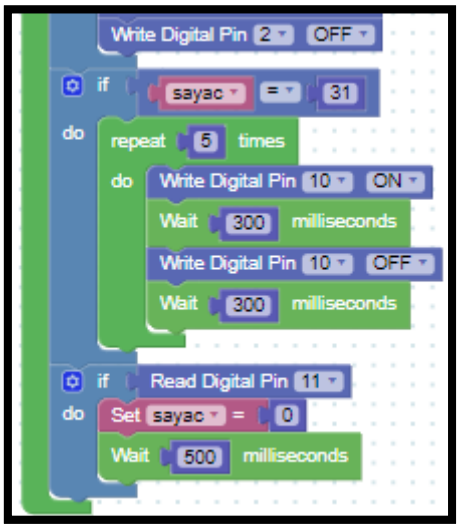
Örnek kod aşağıdaki resimlerde üç bölüm halinde verilmiştir. Bu kodlar arka arkaya birleştirilmelidir.



Resim 156. Örnek Program Birinci Bölüm



Resim 157. Örnek Program İkinci Bölüm



Resim 158. Örnek Program Üçüncü Bölüm

3. DEĞERLENDİR

Dikkat

Birinci değerlendirme sorusunun tartışılması çok önemlidir. Rehber öğretmenler bu bölümde mutlaka bu sorunun tartışılması için en az 10 dakika ayırmalıdır. Sorunun cevabı temel olarak yazılmıştır. Rehber öğretmen, öğrencilerin sınıfça soruyu tartışmalarını sağlamalıdır. Cevap doğrudan öğrencilere kesinlikle verilmemelidir. Tartışmada herhangi bir sonuca ulaşılamadığı durumda rehber öğretmen tartışmaya katılmalıdır. Öğrenciler bu sorudaki devreleri hatırlamayabilir. Rehber öğretmen iki devreyi de tahtaya yansıtır.

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Bugün iki farklı karanlıkta yanan ve aydınlıkta sönen led devresi yapıldı (“LDR ile Karanlıkta Yanan Led: 1.2 Uygula” ve “Transistör ile Karanlıkta Yanan LED: 1.5 Uygula” etkinlikleri). Birinde transistör kullanıldı; diğerinde LDR üzerinden analog giriş yapıldı. Bu iki devre arasındaki fark nedir?

Cevap

Birinde Arduino mikro işlemci olarak kullanılmıştır diğerinde ise sadece güç kaynağı olarak kullanılmıştır. Arduino’nun güç kaynağı olarak kullanıldığı devrede (transistörlü) devre elemanları karanlıkta ledin yanmasını sağlamıştır. Arduino’nun mikrodenetleyici olarak kullanıldığı devrede ise devre elemanları ArduinoBlocks ortamı vasıtası ile kodlanmıştır. Bu kod ile devre çalıştırılmıştır. Bu kodda yapılacak bir değişiklik ile devrenin çalışma şekli değiştirilebilir. Fakat Arduino’nun güç kaynağı olarak kullanıldığı devre kodlanamaz. Sabit bir görevi vardır ve yalnızca onu yerine getirir. Öte taraftan transistörlü devrenin maliyeti Arduino’nun mikrodenetleyici olarak kullanıldığı devreye göre ucuzdur. İhtiyaç ve maliyet dengesi gözetilerek koşula göre hangi devrenin kullanılacağına karar verilebilir.

2. Bazı elektrikli aletlerin fişi prizden çekildiği halde üzerindeki led prizden çekilir çekilmez sönmez ve bir müddet daha yanmaya devam eder. Sizce bunun nedeni ne olabilir?

3. Bu bölümde gördüğünüz devre elemanlarını günlük yaşamda nerelerde kullanabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - BC547 ile DC motor sürdüğüm devrede motor çok hızlı çalıştı ve yanık kokuları gelmeye başladı. Sebebi ne olabilir?
    - BC547 ile DC motor sürdüğüm devrede yüksek direnç kullandığım zaman motor çalışmadı. Sebebi ne olabilir?
    - Sembolik el kurutma makinesi etkinliğinde sayacı 10’luk tabandan ikilik tabana çevirmekte zorlandım. Siz bu duruma nasıl çözüm buldunuz?

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Transistör ile Değil Kapısı Yapıyorum

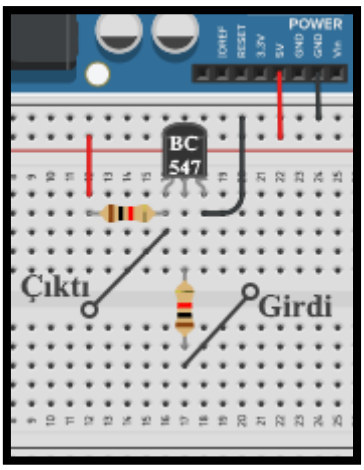
Malzeme Listesi:

Arduino Uno
Breadboard
2 adet 1K Ohm direnç
1 adet BC547 transistör

Bu etkinlikte amaç transistör kullanarak değil (NOT) kapısı oluşturmaktır. Değil kapısının veya değil operatörünün görevi girdinin tersini almaktır. Girdi olarak "1" (5V) girilirse çıktı olarak "0" (0V) verir. Girdi olarak "0" (0V) girilirse çıktı olarak "1" değeri (5V) üretir. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki devreye benzer bir içerik hazırlamaları beklenir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Aşağıdaki resimde "değil kapısı" için örnek bir devre verilmiştir. Girdi "1" verilirse BC547'nin baz ayağına yeterli gerilim ulaşır ve kollektör-emiter arası kısa devre olur (aslında tam olarak kısa devre olmaz fakat burada öyle kabul edilmiştir) yani akım çıktı yerine doğrudan kollektör-emiter üzerinden GND'ye ulaşır. Böylece Çıktı "0" olur. Girdi "0" verilirse BC547'nin baz ayağında gerilim oluşmaz ve güç kaynağından direnç ile gelen akım Çıktı'ya iletilir. Böylece Çıktı "1" olur.



Resim 159. Örnek Devre Şeması

4.2 DC Motorun Hızını Ayarlıyorum

DC motorun hızı uygulanan gerilim miktarına göre değişir. Sınırlar içerisinde olmak kaydı ile DC motor gerilim arttıkça hızlanır, gerilim azaldıkça yavaşlar. Bu derste BC547 ve 9V pil (1.6 Gözle ve Uygula) kullanılarak DC motor çalıştırılmıştır. Fakat bu devrede, gerilim sabit olduğu için, DC motor sabit hızla çalışmaktadır. Bu etkinlikte amaç potansiyometre ile DC motorun hızını kontrol eden devreyi yapıp programını yazmaktır. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

5. Bölüm-Seri Portlar, Multimetre Kullanımı, Step Motor ve Joystick Kullanımı

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, RGB led, PWM, transistor ve kondansatör kullanımını bilir ve gerekli kodlamaları yapabilir.

Bölümün Kazanımları:

- Seri portun nasıl çalıştığını açıklayabilir.
- Seri port ile devreden gelen veriye göre ArduinoBlocks ortamında programlar geliştirebilir.
- Kondansatörü elektronik devrelerde kullanabilir.
- İhtiyacına göre Multimetreyi uygun şekilde kullanabilir.
- Rastgele değer oluşturup buna uygun kod bloğu oluşturabilir.
- Step motor içeren elektronik devreler oluşturabilir.
- Step motor içeren elektronik devrelerin programlamasını gerçekleştirebilir.
- Joystick içeren elektronik devreler oluşturabilir.
- Joystick içeren elektronik devrelerin programlamasını gerçekleştirebilir.
- Robot kolun elektronik devrelerini gerçekleştirebilir.
- Robot kolun elektronik programlamasını oluşturabilir.

Bölümün Amacı:

Bu bölümün temel amacı, öğrencilerin seri port kavramını öğrenmelerini, seri portlar için gerekli devre bağlantılarını oluşturabilmelerini ve ArduinoBlocks ortamında programlayabilmelerini sağlamaktır. Bölümün alt amaçları ise; ArduinoBlocks ortamında rastgele değişkenini kullanabilme, multimetre kullanabilme ve elektronik devrelerde step motor, servo motor, joystick, potansiyometre, LDR, PWM, transistor ve kondansatör kullanabilme ve gerekli programları oluşturabilme olarak belirlenmiştir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino/5 V güç kaynağı, breadboard, 5 adet led, 1 adet RGB led, 1 adet LDR, 1 adet 10K potansiyometre, Multimetre, Step motor ve sürücü kartı (ULN2003), Kartondan yapılmış gösterge pini, 1 adet joystick, Robot kol seti, 4 adet servo motor seti, 4 adet potansiyometre, Direnç (5 adet 330 Ohm, 1 adet 10K Ohm 1 adet 1K Ohm), 1 adet buton (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Elektronik devrelerde seri port kullanımı, multimetre kullanımı, step ve servo motor kullanımı, joystick kullanımı ve robot kol yapımı sürecinin ve programlanmasının gösterilmesi. ArduinoBlocks ortamında rastgele değerlerinin oluşturulması ve ilgili elektronik devrelere yansımasının gösterilmesi.

Uygula: Elektronik devrelerde seri port kullanımı, multimetre kullanımı, rastgele sayıların kullanılması, step ve servo motor kullanımı, joystick kullanımı ve robot kol yapımının uygulanması ve gerekli programların oluşturulması. ArduinoBlocks ortamında rastgele değerlerinin oluşturulması ve ilgili elektronik devrelere yansımasının uygulanması.

Tasarla: Joystick ile labirent kontrol edilebilen ve belirlenen karakterleri yazan robot kol için gerekli elektronik devreleri ve programları tasarlamak için tanımlama ve fikir üretme adımlarının oluşturulması.

Üret: Joystick ile labirent kontrol edilebilen ve belirlenen karakterleri yazan robot kol etkinliği için gerekli elektronik devrelerin ve programların oluşturulması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE ve UYGULA

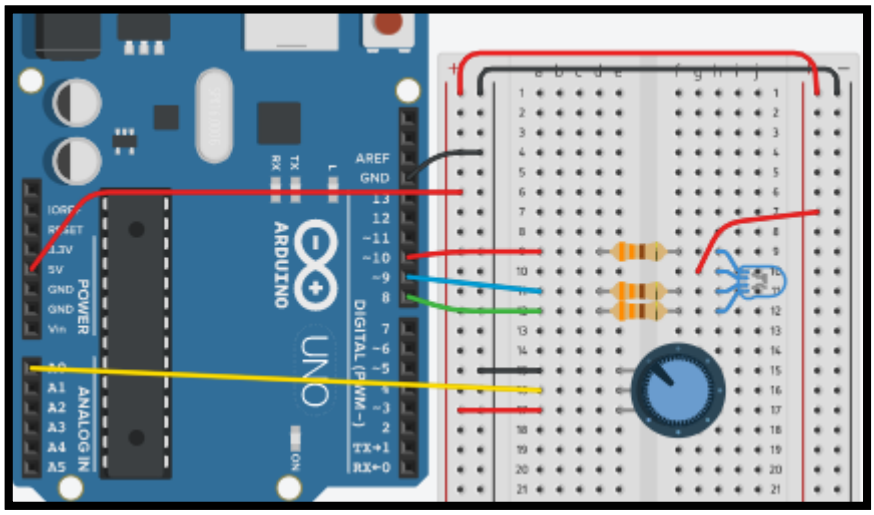
1.1 Gözle ve Uygula- Seri Port Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
1 adet 10K potansiyometre
3 adet 330 Ohm direnç

Arduino üzerindeki birçok bağlantı noktası üzerinden veri alışverişinde bulunabilir. Şu ana kadar dijital ve analog pinler aracılığı ile farklı şekillerde Arduino'ya bilgi gönderilmiş ve alınmıştır. Bu bilgi gönderme ve alma durumu Arduino ile bilgisayar arasında da gerçekleşebilir. Arduino ile seri bağlantı yaparak Arduino'dan alınan analog bir girdinin bilgisayar tarafından kullanılmasını gösterebilmek için aşağıdaki devre kurulur. Bu devrenin amacı potansiyometre ile analog girişe gönderilen gerilimin büyüklüğünü bilgisayar ekranında görüntülemek ve aynı zamanda bu büyüklüğü RGB led ile sınıflandırmaktır. Analog girişe gönderilen gerilim 1.5V'nin altında ise yeşil, 1.5V-3V arasında ise mavi, 3V'den büyük ise kırımız rengin RGB led'de görülmesi gerekmektedir.

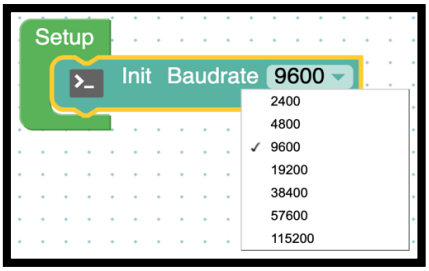
Bunun için öncelikle Arduino kartının 5V ucu breadboard'un "+" kısmına, GND ucu ise breadboard'un "–" ucuna bağlanır. İstenirse breadboard'un sağ ve sol tarafında yer alan "+ ve –" uçlar resimde görüldüğü gibi birbirine bağlanabilir. Bu şekilde devre elemanlarına gerektiğinde kolayca güç verilebilecektir. Daha sonra potansiyometre breadboard'a takılır ve sağ ucu "+", sol ucu ise "–"ye gelecek şekilde bağlantıları yapılır. Potansiyometrenin orta ucu ise analog giriş olarak kullanılacağı için A0 analog pinine takılır.



Resim 160. Örnek Devre Şeması

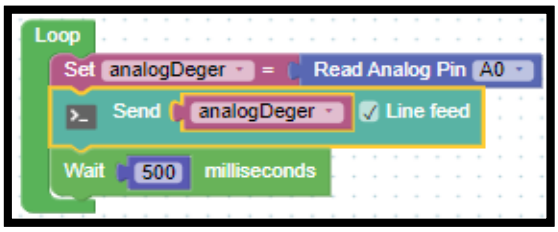
Daha sonra RGB led breadboarda takılır ve sırasıyla kırmızı ucu 10 numaralı, mavi ucu 9 ve yeşil ucu ise 8 numaralı dijital pine takılır. Ortak anot ucu ise "+" bağlantı noktasına takılır. Bu şekilde devre tamamlanmış olur. İstenen görevi yerine getirmeden önce A0 analog pininden okunan değeri seri port ekranında göstermek yerinde olacaktır. Bir adım sonra istenen görev yerine getirilebilir. Aşağıda resimde gösterildiği Setup bloğu içerisinde seri port ekranı ile iletişim hızı ayarlanır. *Init Bautrate* bloğu bilgisayara veri göndermeyi ya da almayı başlatmak için kullanılır. Aynı zamanda giriş ve çıkış okumaları için kullanılacak hız da bu blok ile belirlenir. Bloğun sağ tarafında yer alan açılır menüden

istenilen hız seçilir. Bu ders kapsamında yer alan etkinliklerin birçoğunda 9600 seçeneği yeterli olacaktır.



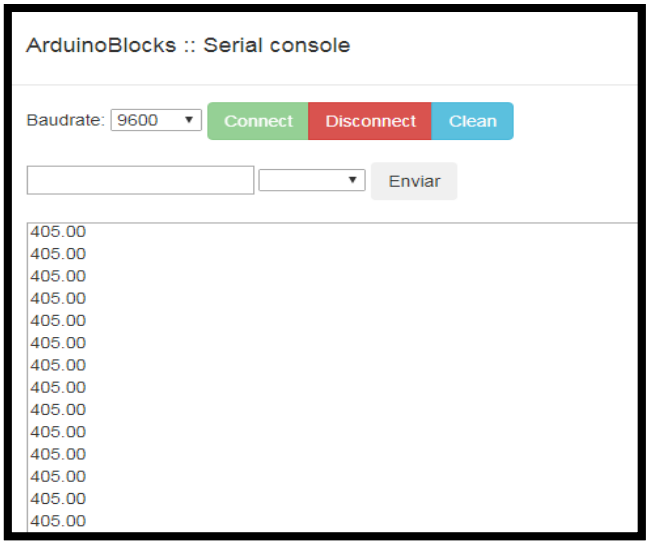
Resim 161. Baudrate Ayarı

Seri port kullanımı aktif edildikten sonra aşağıdaki resimde görülen kod yazılarak Arduino'ya yüklenerek çalıştırılır.



Resim 162. Örnek Program

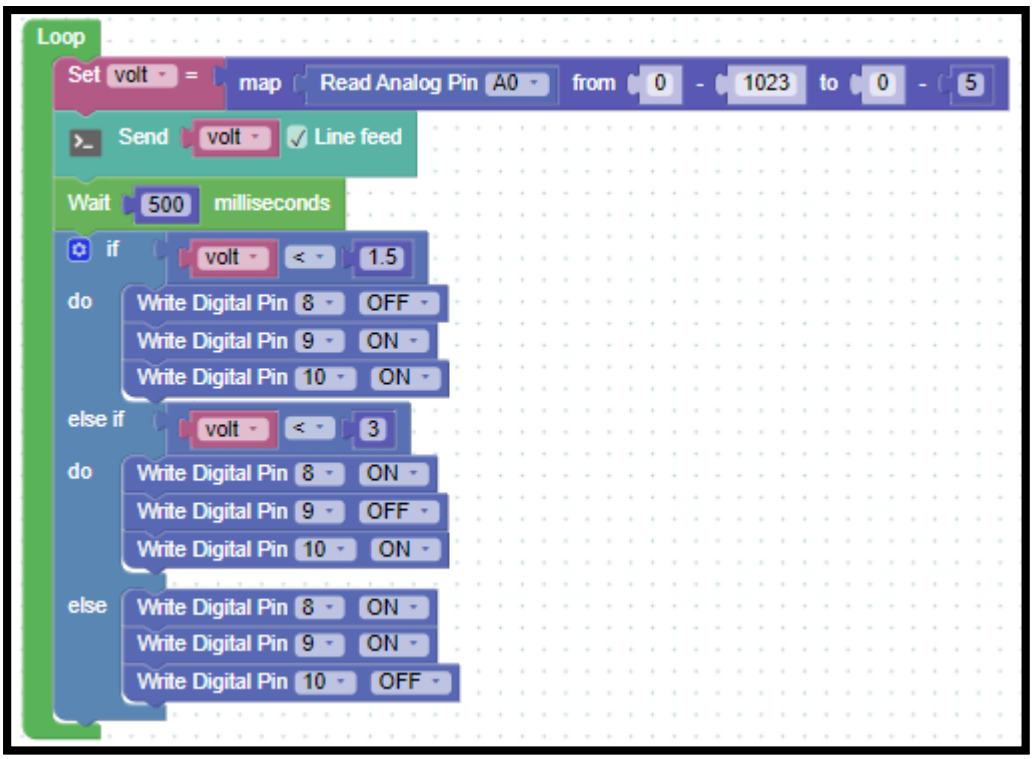
Yukarıdaki kod ile oluşturulan analogDeger değişkeninin içeriği A0 analog pinine gelen değer ile güncellenecektir. Kod bloğunda ayrıca Serial Port blok başlığında yer alan *Send – Line feed* bloğu da kullanılmıştır. Bu blok, üzerinde verilen değeri seri port ekranına yazdırma işini yerine getirmektedir. Line feed yazısının yanındaki onay işareti kaldırılırsa okunan değerler ekranda yan yana yazdırılacaktır. Onay işaretinin kullanılması ile okunan her bir değer yeni satıra yazdırılacaktır. Yukarıdaki örnekte seri port ekranına analogDeger değişkeninin içeriği gönderilmektedir. Ekrana gönderilen değerlerin görülebilmesi için yükleme işlemi yapıldıktan sonra ekranın sağ üst kısmında yer alan Console butonu tıklanır. Console butonu açıldıktan sonra Connect butonuna tıklanır. Aşağıdaki resimdekine benzer bir ekran görüntüsü elde edilir.



Resim 163. Seri Konsol Ekran Çıktısı

Açılan pencere, bağlantısı yapılan A0 analog girişindeki değerleri gösterecektir. Ancak potansiyometre en sağdan en sola döndürüldüğünde ekranda okunan değerlerin 0 ile 1023 sayıları arasında değiştiği gözlemlenecektir. 0V için ekranda 0 değeri görülür ve 5V için ekranda 1023 değeri görülür.

Bu adımda artık ekrana gerçek gerilim değerlerini gösteren ve RGB ledin söylendiği şekilde yanmasını sağlayan program yazılacaktır. Ekrana gerçek gerilim değerinin yazdırılması için "*map*" bloğu kullanılabilir. Map bloğu ile "analogDeger" gerçek gerilim değerine dönüştürülebilir. Aşağıdaki resimde örnek program görülmektedir.



Resim 164. Örnek Program

Yukarıdaki kod Arduino karta yüklenir ve hem ekrandaki sayılar hem de RGB ledin renkleri incelenerek renk değişimlerinin hangi noktalarda gerçekleştiği gözlemlenir. Bu etkinlikte Arduino bilgisayara seri bağlantı üzerinden bilgi göndermiştir. Aynı zamanda bilgisayar da benzer şekilde bilgi gönderebilir ve Arduino'ya bağlı farklı kartlar ile bilgi alışverişinde bulunabilir. Bunun için Arduino üzerinde yer alan 0 (RX) ve 1 (TX) numaralı pinler kullanılır. Bu pinlerin kullanımı ilerleyen IOT etkinliklerinde anlatılacaktır.

1.2. Gözle ve Uygula- Multimetre ile Direnç, Gerilim ve Akım Ölçüyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB led
1 adet 10K potansiyometre
1 adet 10K Ohm direnç
1 adet multimetre

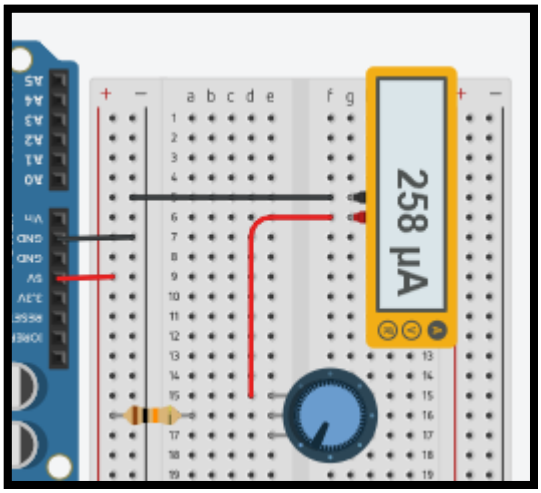
Bilgisayarda ölçülen değerlerin doğru şekilde ölçülüp ölçülmediği de multimetre isimli bir araç ile test edilebilir. Multimetre belirli noktalar arasındaki akım, gerilim ve direnç değerlerini ölçebilen bir araçtır. Aşağıdaki resimde multimetre görülmektedir.

Resim 165. Multimetre

Multimetre elektronik devrelerde farklı birçok amaçla ölçümler yapmak için kullanılır. Örneğin bir direncin değerinin üzerindeki renk kodları ile verilen değerde olup olmadığını multimetrenin direnç ölçme özelliği ile kontrol edilebilir. Bunun için multimetre "Ω" işareti ile gösterilen kademede ölçülecek direncin büyüklüğüne göre ayarlanır. Pozitif ve negatif uçlara bağlı problar (ölçüm uçları) direncin iki ucuna değdirilerek multimetrenin üzerindeki değer okunur. Örnek olarak devrede RGB ledlere takılı olan dirençlerden biri çıkarılır ve bahsedildiği şekilde direnç değeri ölçülür. Ölçülen değerin 330 ohm olup olmadığı kontrol edilir. Ölçüm ile gerçek değer arasında ufak farkların olabileceğinden bahsedilir. Direnç ölçüm özelliği birbirine bağlı olması gereken iki noktanın gerçekten bağlı olup olmadığını anlamak için de kullanılabilir. Birbirine bağlı olan iki noktanın arasında hiç direnç olmayacağı için ilgili noktalar arası multimetre ile ölçülürse 0 değerinin okunması gerekir.

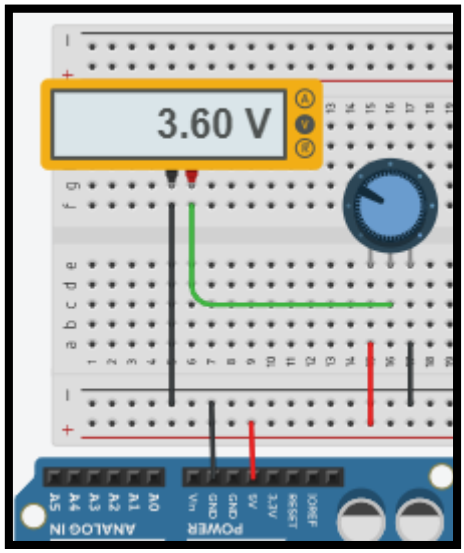
Akım ölçmek için multimetre "µA" kademesine getirilir. Multimetrenin bir noktadan geçen akımı ölçebilmesi için akımın multimetre üzerinden geçmesi gerekir. Öncelikli olarak multimetrenin "-" ucu

breadboardun GND bağlı şeridine bağlanır. Bu bağlantı için erkek-erkek jumper kablo ve krokodil kablo kullanılabilir. "+" uç ise potansiyometrenin sol ucuna bağlanır. Potansiyometrenin orta ucu 10K Ohm direnç üzerinden GND'ye bağlanır. Örnek devre aşağıdaki resimde gösterilmiştir. Ardından Arduino'ya güç verilir. Potansiyometrenin ayarı değiştirilerek multimetrenin okuduğu değerler gözlenir. Bu noktada devreden geçen akımın direnç değerine bağlı olduğundan bahsedilir ve potansiyometre ile direnç arttırıldığında devreden geçen akımın azaldığı, direnç düşürüldüğünde ise akımın arttığı vurgusu yapılır.



Resim 166. Akım Ölçme

Gerilim ölçmek için ise multimetre bir devrenin farklı iki noktasına bağlanır ve bu iki nokta arasındaki potansiyel fark ölçülür. Aşağıdaki resimde multimetre gerilim ölçmek için konumlandırılmıştır. Multimetre "V" ile gösterilen gerilim ölçme bölümünde uygun kademeye getirilir. Bu şekilde potansiyometrenin sol ucu ile orta ucu arasındaki gerilim farkı ölçülür. Devreye güç verilir. Multimetre üzerindeki değer okunur. Potansiyometrenin kolu çevrilerek multimetredeki değişim gözlemlenir.



Resim 167. Gerilim Ölçme

Akım, gerilim ve direnç ölçümlerinin hepsinde geçerli olmak kaydıyla ölçüm ayarının doğru konumlandırılması önemlidir. Örneğin bir direnç değeri ölçülecekse potansiyometrenin direnç değerinden büyük olmak kaydıyla olabilecek ön düşük kademede olması gerekir. Çok büyük kademelerde tam sonuç elde edilemezken, düşük kademelerde ise okuma hiç yapılmaz. Aşağıdaki

resimlerde 330 ohm değerindeki direncin yüksek, ideal ve düşük kademelerdeki ölçümlerine ait ekran görüntüleri yer almaktadır.

Resim 168. Yüksek Ayar

Resim 169. Doğru Ayar

Resim 170. Düşük Ayar

Yukarıdaki görsellerde de görüldüğü üzere doğru okuma ortadaki resimdeki gibi olmalıdır. Ortadaki ölçümde multimetre ayarı 2000 ohm şeklindedir. Soldaki resimde multimetre ayarı 2000 Kohm ve sağdaki ölçümde ise multimetre ayarı 200 ohm olarak ayarlanmıştır.

1.3 Gözle- Rasgele Sayılar

Programcılar zaman zaman herhangi bir kritere bağlı kalmadan sayı üretilmesine ihtiyaç duyabilirler. Bunun için programlama dillerinde rasgele sayı üretimini sağlayan bir komut yer alır. Arduino projelerinde de rasgele sayı üretmek için de bir komut yer almaktadır. *Random integer from* komutu başlangıç ve bitiş değerleri arasında herhangi bir koşula bağlı olmadan rastgele bir sayı üretir. Aşağıdaki resimde yer alan komut başlangıç değeri olan 1'den bitiş değeri 100'e kadar bir sayı üretir. Bu program içerisinde ilgili yerde kullanılabilir.

Aşağıdaki resimlerde görülen programda her defasında 1 ile 1000 arasında rasgele bir sayı üretilir ve üretilen sayı seri port ekranına yazılır.

Resim 171. Rasgele Sayı Üretme Bloğu

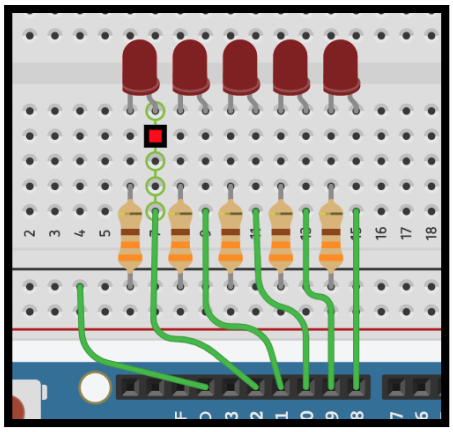
Resim 172. Örnek Program

1.3 Uygula- Butona Basıldığına Rasgele Yanan Led

Malzeme Listesi:

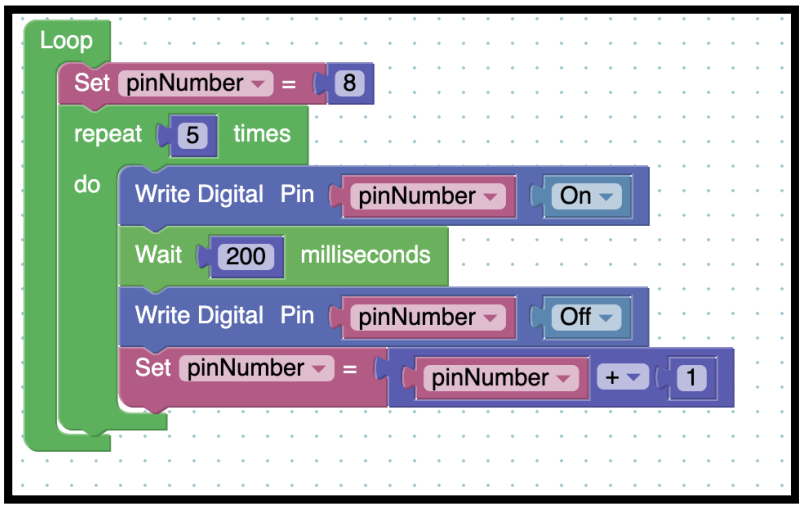
Arduino Uno
Breadboard
5 adet led
5 adet 330 Ohm direnç
1 adet buton

Bu etkinlikte amaç bir butona basıldığında sırayla yanıp sönmeye başlayan ledlerin rasgele bir ledde kalarak durmasını sağlamaktır. Beş farklı dijital çıkışa bağlanan ledler sırayla yanmalı ve rasgele bir ledde durmadan önce 10 kere başa dönmelidir. Bunun için öncelikle aşağıdaki resimde görülen devre kurulur.



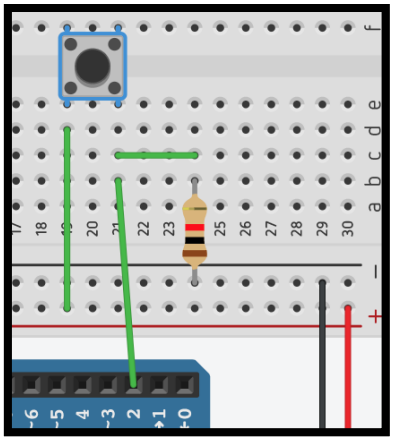
Resim 173. Örnek Devre Şeması

Resimdeki devrede seçilen 5 dijital uç birer lede ve ledlerin katot uçları da 330 ohm değerindeki dirençler üzerinden GND'ye bağlanmıştır. Bu ledleri sırayla yakmak için seçilen ilk dijital çıkış ucu ON/HIGH konumuna getirilip, belli bir süre OFF/LOW konumuna getirilmelidir. Ardından sıradaki lede geçilerek aynı işlem tekrarlanmalıdır. Her bir led için bu durum tekrarlanacağı için işlem bir döngü ile gerçekleştirilebilir. Bunun için aşağıdaki resimdekine benzer bir döngü kullanılabilir. Bu devrede çıkış için 8 ile 12 arasındaki pinler kullanılmış ve döngü de bu pinler dikkate alınarak oluşturulmuştur. İstenirse diğer çıkışlar da kullanılabilir.



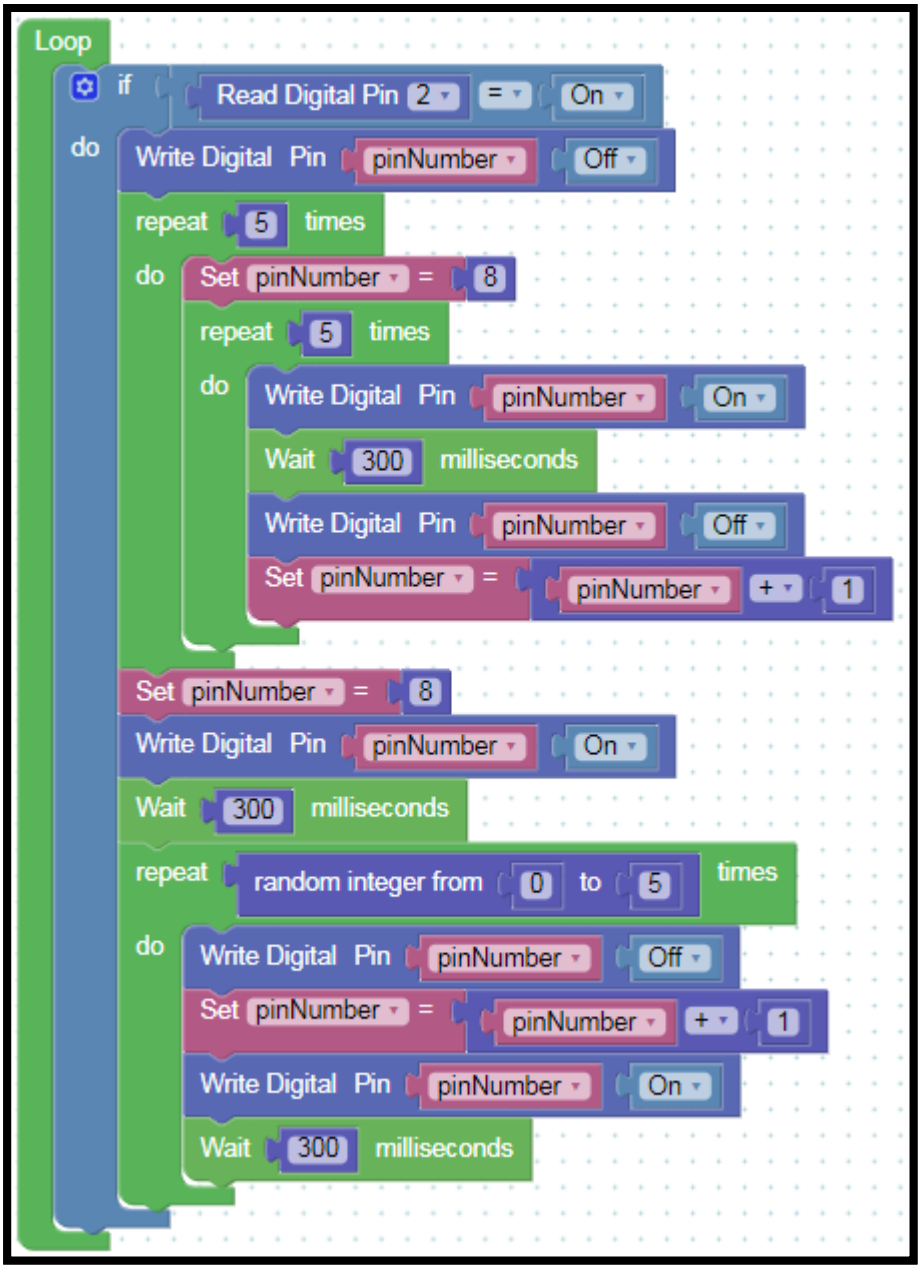
Resim 174. Örnek Program

Arduino kodu zaten sürekli bir döngü içinde olduğu için devre bu kodla çalıştırıldığında içeride oluşturulmuş olan döngü aslında sürekli başa dönmekte ve ilk led'den başlayarak ledleri sürekli yakıp söndürdüğü sırayı tekrarlamaktadır. Etkinliğin ilerleyen adımlarında bu durumun düzeltileceği öğrencilere aktarılmalıdır. Bu döngü kurulduktan sonra devreyi bir buton ile aktive etmeyi sağlamak için aşağıdaki resimde görülen bölüm devreye eklenmelidir.



Resim 175. Ek Devre Şeması

Devreye eklenen bu bölüm daha önceki derslerde kullanılan buton kullanma biçimlerinden pull-down şeklinde kullanılmıştır. Butona basıldığında 2 numaralı dijital girişe "1" bilgisini gönderirken, bırakıldığında ise 0 bilgisini gönderecektir. Bu noktada yukarıda verilen kod bloğunun bu buton basıldığında aktif olmasını sağlayan kod yazılacaktır ve bu butona basıldığında sırayla yanıp sönmeye başlayan ledlerin rasgele bir ledde kalarak durmasını sağlanacaktır. Bu kod aşağıdaki resimde verilmiştir. Ancak kodu sınıf ile paylaşmadan önce öğrencilerden bu durumu kendilerinin çözmesi beklenmeli ve öğrencilere rehberlik sağlanmalıdır. Kodu yazamayan gruplara yardım edilmelidir.



Resim 176. Örnek Program

Yukarıdaki kod bloğunda iç içe iki döngü kullanılmıştır. Bu döngülerden içteki ledleri sırasıyla yakıp söndürürken, dıştaki döngü bu olayın da 5 defa tekrarlanmasını sağlamaktadır. Ayrıca bu hareketin başlaması ise buton ile kontrol edilmiş ve bir koşullu ifade ile gerçekleştirilmiştir. Bunların dışında hareketin üçüncü bir döngü ile rastgele bir ledde durması sağlanmıştır. Döngü 0 ile 3 arasında rastgele bir sayı üzerinden döndürülerek bu işlem tamamlanmıştır.

1.4 Gözle ve Uygula- Step Motor ile Tahmin Oyunu

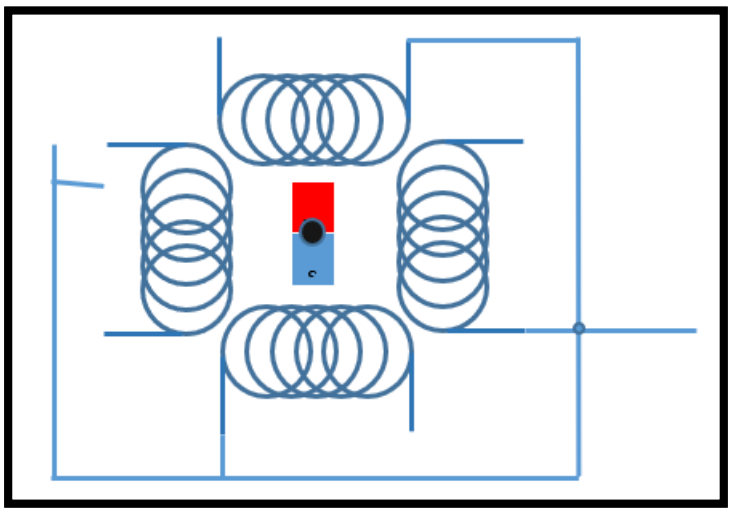
Malzeme Listesi:

Arduino Uno
Breadboard
Step motor ve sürücü kartı (ULN2003)
Kartondan yapılmış gösterge pini
2 adet led

1 adet buton
2 adet 330 Ohm, 1 adet 1 KOhm direnç

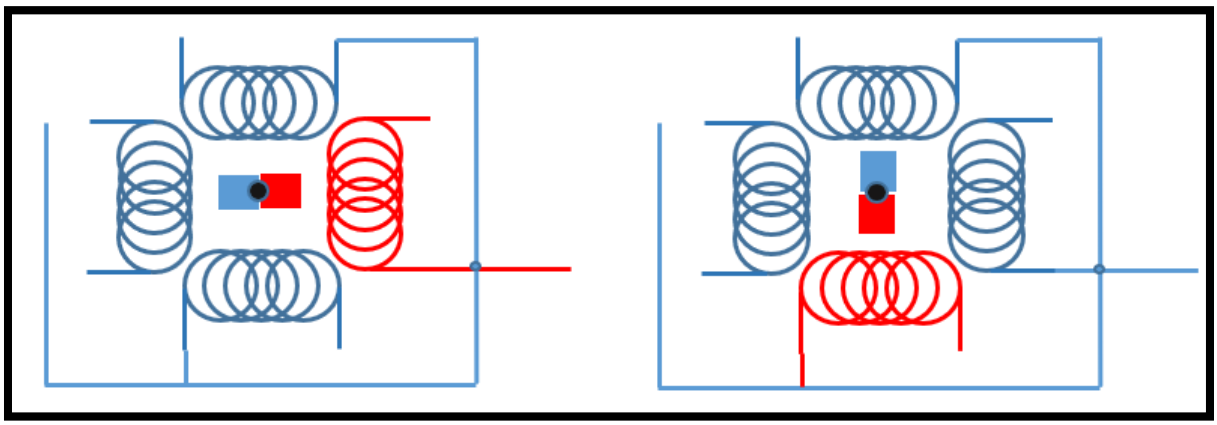
Bu etkinlikteki amaç rasgele bir sayı üreterek step motorun üretilen bu sayıya paralel olarak hareket etmesini sağlamaktır. Kullanıcı butona bastığında step motora bağlı gösterge pini hareket etmeye başlayıp, konumlandırılmış ledlerden birinde rastgele duracaktır. Bir ledde konumlanmadan önce en az dört kere iki led arasında gidip gelmeli ve işaret ettiği led de yanmalıdır.

Step motorların ortasında serbestçe dönebilen bir mıknatıs ve mıknatısın çevresine yerleştirilmiş farklı sayıda bobin yer almaktadır. Bobinlerin üzerinden akım geçtiğinde manyetik olarak aktif olurlar etraflarında yer alan metal ve mıknatısları çekerler. Step motorun içerisinde yer alan bu bobinler uygun şekilde sırayla aktif edildiğinde ortada yer alan mıknatıs istenilen tarafa döndürülebilir. Aşağıdaki resimde step motorun içi yapısı görülmektedir.



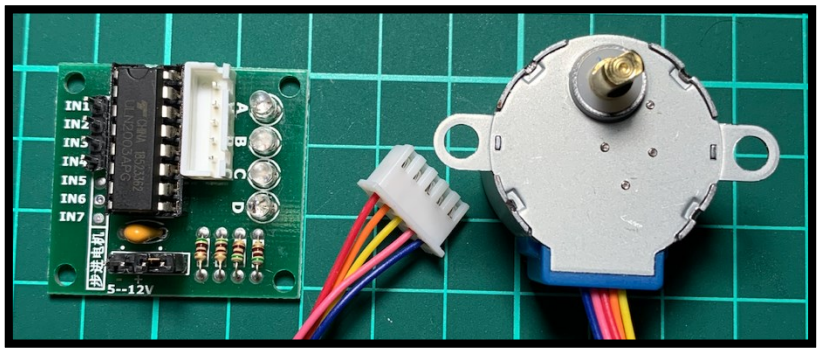
Resim 177. Step Motor İç Yapısı

Resimde işaret edildiği gibi her bir bobinin ikişer ucu vardır bunların birer tanesi birbirine bağlıdır. Yani bobinlerinin birer ucu da açıkta kalır. Ortak olan uç – gerilime bağlanır. Eğer bir bobinin boşta kalan ucuna + gerilim verilirse o bobin aktif olur ve motorun ortasında yer alan mıknatısı çeker. Bobinler sırayla gerilime bağlanıp ayrılırlarsa ortada yer alan mıknatıs bir dönme hareketi meydana getirir. Örneğin en sağdaki bobin aktif edildiğinde ortadaki mile bağlı mıknatıs bu bobine dönecektir. Alttaki bobin aktif olduğunda ise mıknatıs altta yer alan bobine dönecektir. Aşağıdaki resimde aktif sağ ve alt bobinler görülmektedir.



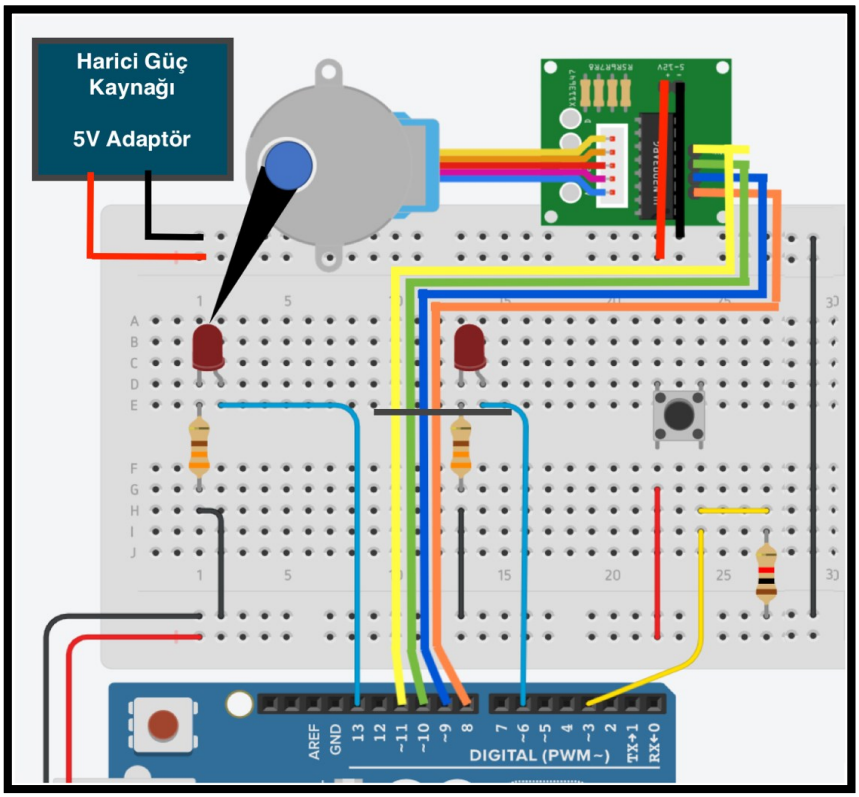
Resim 178. Aktif Sağ ve Alt Bobinler

Bu işlem arka arkaya tekrarlandığında ortadaki mil sürekli döndürülebilir. Etkinliklerde kullanacağımız step motorun bir örneği ve Arduino'ya bağlantısının yapılması için gerekli olan sürücü kartı (ULN2003) aşağıdaki resimde verilmiştir. Step motor çalışmak için Arduino kartın sağlayabileceği akımdan daha fazlasına ihtiyaç duyar. Bu nedenle doğrudan Arduino'ya bağlanırsa çalışmak için gerek duyduğu akımı karttan almaya çalışacak ve dolayısıyla Arduino'dan çok fazla akım geçişine neden olacaktır. Bu da Arduino'nun zarar görmesine neden olacaktır. Sürücü ise bu sorunu ortadan kaldırarak Arduino karttan gelen akımı kendi içinde step motoru çalıştıracak kadar yükseltmekte ve olası zararların önüne geçmektedir. Ancak yine de yüksek akım geçişi nedeniyle Arduino kartının zarar görme olasılığı mevcuttur. Bu nedenle Step motor ayrı bir güç kaynağı ile çalıştırılırsa daha sağlıklı olacaktır. Bunun için setlerde yer alan adaptör kullanılabilir.



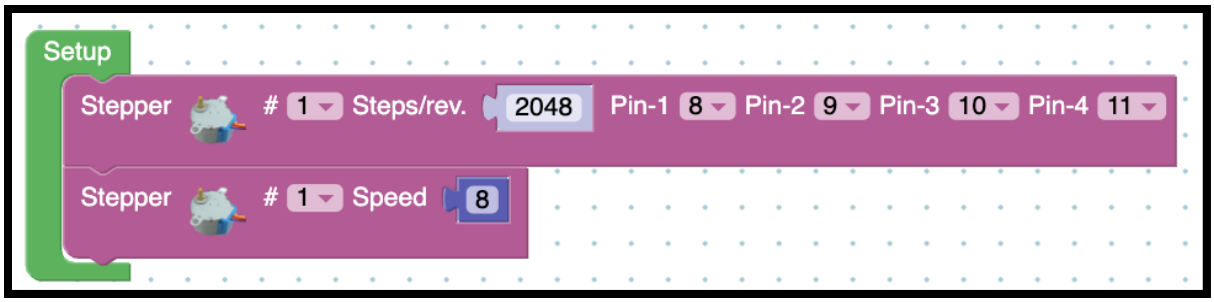
Resim 179. Step Motor ve Sürücüsü

Yukarıdaki resimde step motorun çıkış uçlarını bir arada tutan bir soket olduğu görülmektedir. Bu soket sürücü üzerindeki ilgili alana takılarak step motorun bağlantıları sürücü üzerinden yapılır. Etkinlikte istenilen devre aşağıdaki resimde görüldüğü gibi kurulmalıdır.



Resim 180. Örnek Devre Şeması

Devredeki butona basıldığında gösterge pini iki led arasında gidip gelmeli ve hangi lede yaklaşırsa o ledi yakmalıdır. Daha sonra ise iki led'den birinde rastgele durmalıdır. Bu işlem için öncelikle step motorun setup bloğu içerisinde tanıtılması ve dönüş hızının belirlenmesi gerekir. İlgili bloklar aşağıdaki resimde ifade edilmiştir.



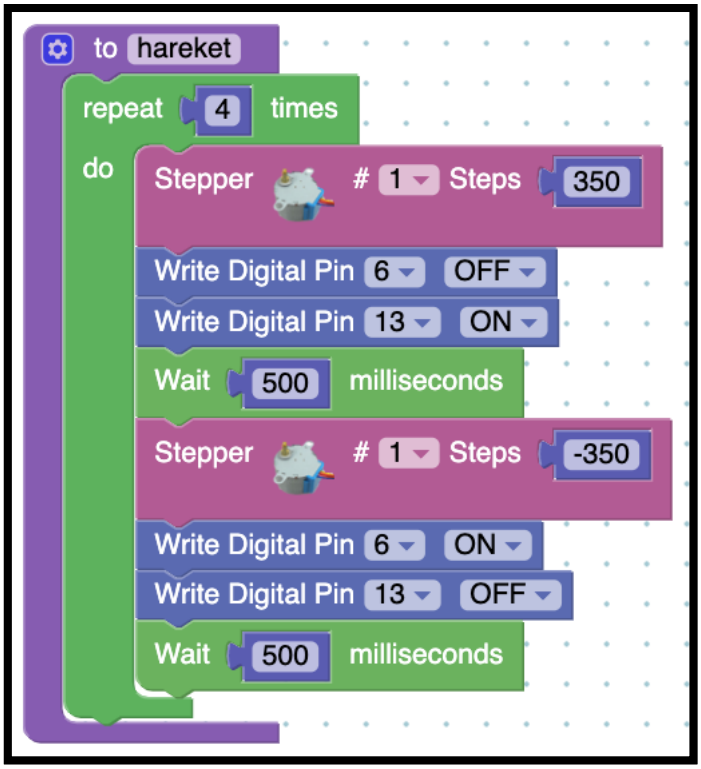
Resim 181. Step Motor Setup Kodları

Resimde görüldüğü üzere step motorun 8, 9, 10 ve 11 numaralı pinlere bağlı olduğu ilk komut ile belirlenmiştir. Bu komut içerisinde ayrıca Steps/rev etiketi ile belirlenen 2048 sayısıyla step motorun bir tur atmak için atması gereken adım ifade edilmektedir İkinci blok ise step motorun dönüş hızı için kullanılmıştır. Hız değeri arttıkça step motorun adımları arasındaki süre kısalmaktadır. Bu etkinlikte hız 8 olarak belirlenmiştir. Step motoru hareket ettirmek için Motor menüsü altında yer alan ve aşağıda resmi verilen komut kullanılır.

Resim 182. Stepper Bloğu

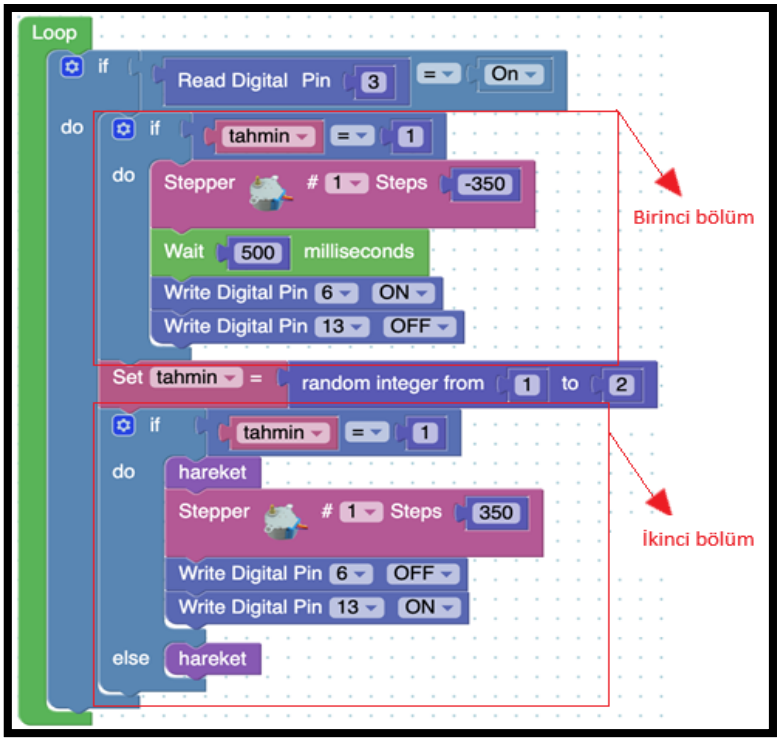
Bu komut Steps alanına girilen sayı kadar adım atılmasını sağlar. Setup bloğu içerisinde 2048 olarak belirlenen adım sayısının bir tur için gerekli adım olduğu durumda, bu komut içerisinde 2048 değeri bir tam tur atılmasını sağlayacaktır. Ayrıca eksi (-) değer girilmesi ters yönde dönüşü sağlayacaktır. Yani + değer saat yönünde, - değer ise saat yönünün tersinde hareket içindir. Bu şekilde motor iki yönlü hareket edecektir.

Butona basıldığında belirlenen ledde durmadan önce iki led arasında dört kere gidip gelinmesi gerekmektedir. Bu işlem için bir fonksiyon yazılmalıdır. Örnek kodlar aşağıdaki resimde görüldüğü gibi olmalıdır. Rehber öğretmen öncelikle fonksiyon içerisinde verilen kodu Loop bloğunun içinde yazıp öğrenciler ile test etmeli ve daha sonra fonksiyon haline dönüştürmelidir.



Resim 183. Hareket Fonksiyonu

Fonksiyon step motoru 350 adım saat yönünde döndürdükten sonra 13 numaralı pine bağlı ledi yakmaktadır. Yarım saniye bekledikten sonra saat yönünün tersinde 350 adım atarak 13 numaralı pine bağlı ledi söndürüp 6 numaralı pindeki ledi yakmaktadır. Bu işlem bir döngü içerisinde dört kere tekrarlanmaktadır. Fonksiyon yazıldıktan sonra aşağıdaki resimde görülen kod Arduino'ya yüklenmelidir.



Resim 184. Örnek Program

Bu kod iki bölümden oluşmaktadır. İlk olarak ikinci bölüm açıklanacaktır. İkinci bölüm oluşturulan tahmin isimli değişkene Math blok başlığı içerisinde yer alan *random integer from* komutu yardımıyla 1 ve 2 arasından rasgele bir sayı seçilmesini sağlayarak başlamaktadır. Seçilen sayı 1 ise oluşturulan hareket fonksiyonu çalıştırılır ve son olarak 350 adım saat yönünde bir hareket daha sağlanır. Ledlerden biri yakılır. Eğer tahmin 2 ise else bloğu çalıştırılır. Bu blokta ek 350 adımlık komut yer almaz. Bu nedenle diğer ledde durulacaktır. Kodun birinci bölümü ise step motoru başlangıç konumuna getirmek içindir. Eğer bir önceki butona basışta 1 sayısı üretilmişse else bloğunun motor 350 adım geri getirilerek başlangıç konumuna dönecektir.

Dikkat

Rehber öğretmen ledlerin arasındaki mesafe ve motorun konumlanmasına göre 350 adımlık hareketin arttırılıp azaltılabileceğini öğrenciler ile paylaşmalıdır.

1.5 Gözle ve Uygula- Joystick ve Seri Port Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick

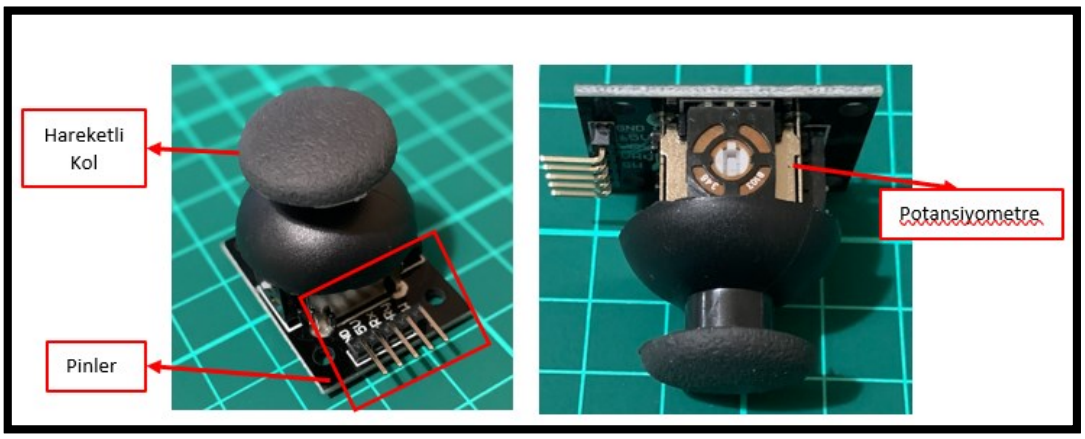
Bu etkinliğin amacı Joystick ile iki Analog girişe aynı anda 0 ile 1024 arasında değerler göndermek ve joysticki hareket ettirerek ile bu değişiklikleri bilgisayar ekranında gözlemlemektir.

Joystickler bilgisayar ve video oyunlarında sıklıkla kullanılan ancak günlük yaşamda da çok fazla karşılaşılan araçlarıdır. Bir diğer adı da kontrol çubuğudur. X ve Y koordinatlarında hareket eden bir çubuğa sahiptir. Beş ve altıncı sınıf öğrencileri derslerinde henüz koordinat eksenini görmemiş olabilir. X ve Y koordinatı yerine sağ-sol ve yukarı-aşağı ya da sağ-sol ve ileri-geri eksenlerinde de hareket ettiği

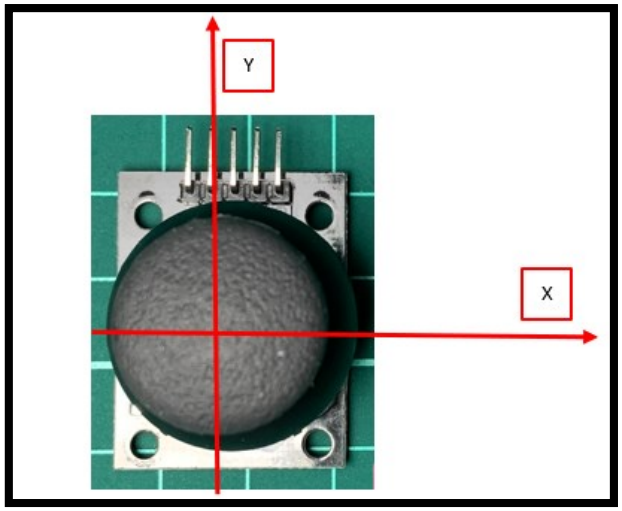
şeklide de ifade edilebilir. Aşağıda Arduino ile uyumlu olarak çalışan ve etkinliklerde kullanılacak olan joystick resmi yer almaktadır.

Aslında basit bir joystick içerisinde iki adet potansiyometre yer almaktadır. Bir potansiyometre "X" ekseni (sağ-sol), ikinci potansiyometre ise "Y" ekseni (yukarı-aşağı ya da ileri-geri) hareketi için kullanılır. Daha önceki etkinliklerde potansiyometre ile analog girişler kullanılarak 0 ile 1024 arasındaki değerler Arduino'ya gönderilmişti. Joystick de analog girişleri kullanarak aynı anda iki farklı girişe 0 ile 1024 arasında değerler gönderebilir. Aşağıda Arduino ile kullanılabilen bir Joystick resmi ve jostick'in içerisinde yer alan potansiyometrelerden biri yer almaktadır.

Aşağıdaki resimlerde Joystickin 5 adet pine sahip olduğu görülmektedir. Bu pinlerden solda yer alan iki pin GND ve 5V pinleridir ve joystickin güç kaynağına bağlanması için kullanılır. Arduino'nun GND ve 5V girişlerinden gelen güç, analog girişlere değer göndermek için gereklidir. Ortada yer alan ucu X ekseni hareketi (jostickin sağa ve sola hareketi) değerini okumak için kullanılır. Dördüncü uç ise joystick kolunun Y ekseni (yukarı-aşağı) hareketini okumak için kullanılır. Son pin ise joystickin buton özelliğini kullanmak için gereklidir. Daha önceki etkinliklerde gösterilen buton bağlantıları GND ucu ve 5 numaralı pin için yapılırsa hareketli kola basıldığında bu iki ucu iletken hale getirir. Bu etkinlikte jostickin buton özelliği kullanılmayacaktır.

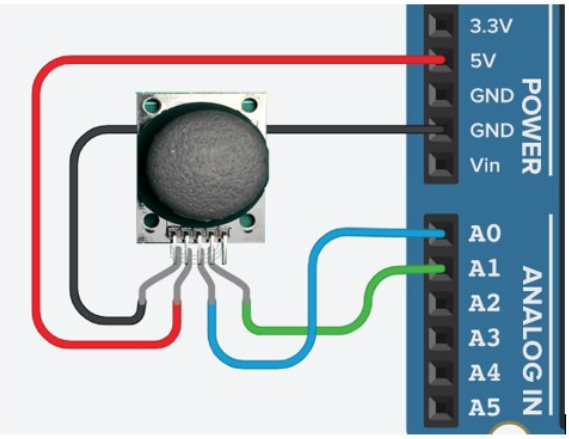


Resim 185. Joystick Önden ve Yandan Görünüm



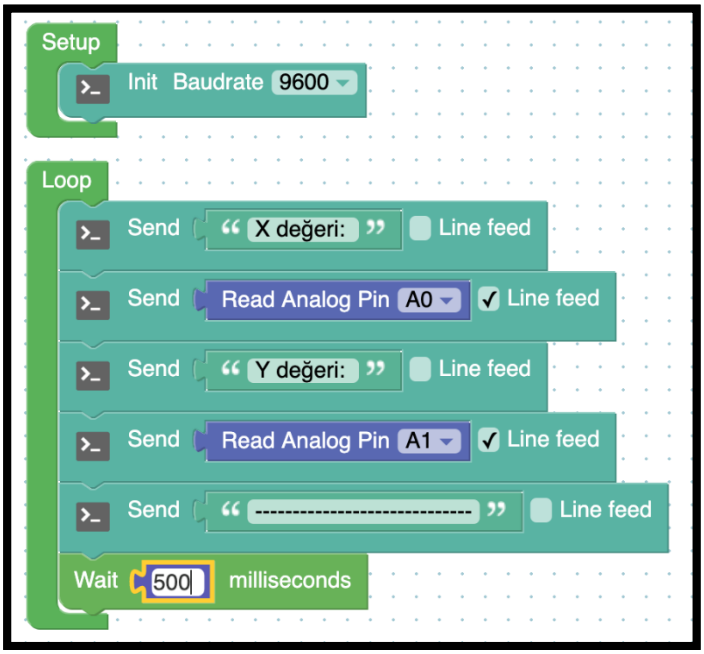
Resim 186. Joystick Hareket Koordinat Yapısı

İlk olarak aşağıdaki resimde verilen devre kurulur. Bu devrede joystickin GND ve 5V uçları Arduino'daki aynı isimli uçlara bağlanır. Daha sonra X ucu Arduino'nun A0 analog girişine, Y ucu ise Arduino'nun A1 analog girişine takılır. Bu şekilde devre bağlantıları kurulmuş olur.



Resim 187. Örnek Devre Şeması

Joystick bağlantıları yapıldıktan sonra kol hareketleri ile değişen değerleri bilgisayardan okuyabilmek için aşağıdaki resimde görülen kod yazılıp Arduino'ya yüklenmelidir. Burada seri port kullanımı yapılarak bu işlem gerçekleştirilecektir.



Resim 188. Örnek Program

Arduino üzerinden okunan değerlerin ekranda anlaşılır şekilde yer alması için ekrana S*end* komutu ile "X değer:" ve "Y değeri:" ifadeleri yazılmış ve ifadelerin her birinden sonra joystick tarafından gönderilen X ve Y değerleri okunarak ekrana yazdırılmıştır. Bu ifadelerin yanında yer alan "*Line feed*" kutucuklarındaki işaret kaldırılarak imlecin alt satıra inmesi engellenmiştir. Son olarak her bir okumayı birbirinden ayırmak için "-" işaretiyle okumalar arası bir çizgi oluşturulmuş ve okumalar arasında da yarım saniyelik bekleme eklenmiştir.

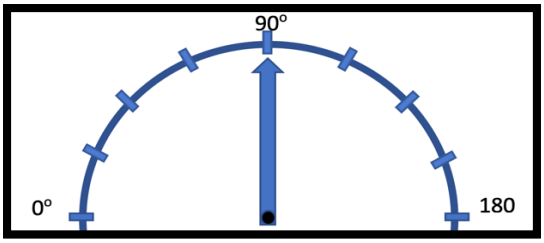
1.6 Gözle ve Uygula- Joystick ile Servo Motor Kontrolü

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick
1 adet servo motor

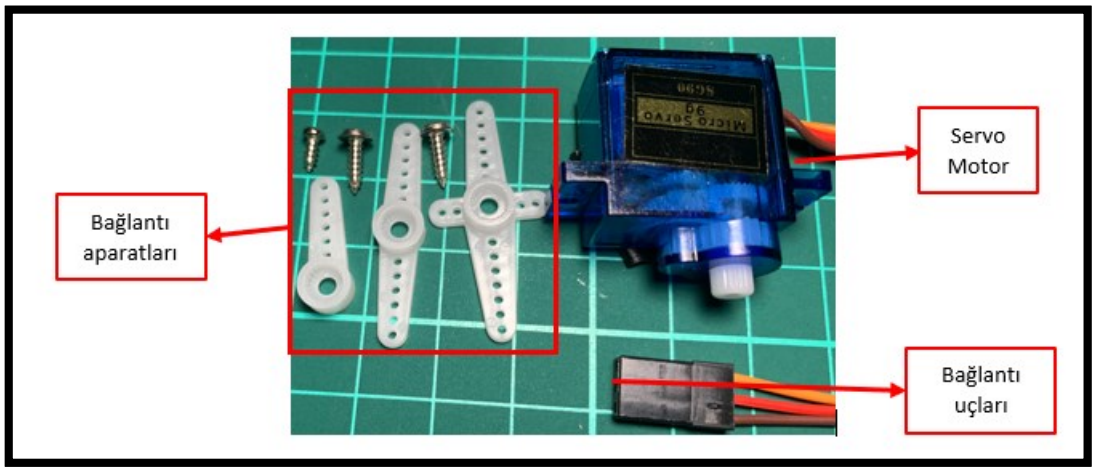
Bu etkinlikte amaç açısal dönüş özelliğine sahip bir motor olan servo motorun joystick yardımıyla istenilen derecede dönmesini sağlamaktır.

Servo motor genellikle 0 ile 180 derece arasında (bazı servolar 360 derece dönüş sağlayabilir) dönüş sağlayabilen bir motordur. 180 derece dönüşün karşılığı yarım turdur. Yarım tur içerisinde servo motor istenilen derecede durdurulabilir. Aşağıdaki resimde servo motor dönüş açıları görülmektedir.



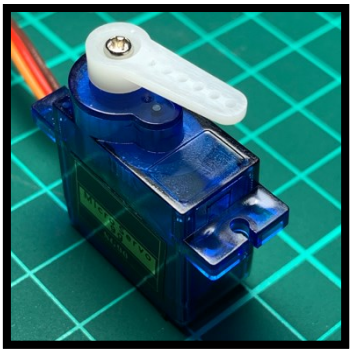
Resim 189. Servo Motor Dönüş Açıları

Servo motorların diğer bir özelliği ise ağır yükleri kaldırabilmeleridir. Bu nedenle robotik uygulamalarda robot kollarını hareket ettirmek için sıklıkla kullanılır. Elektronik programlama etkinliklerinde kullanılacak olan servo motorun bir örneği aşağıdaki resimde verilmiştir.



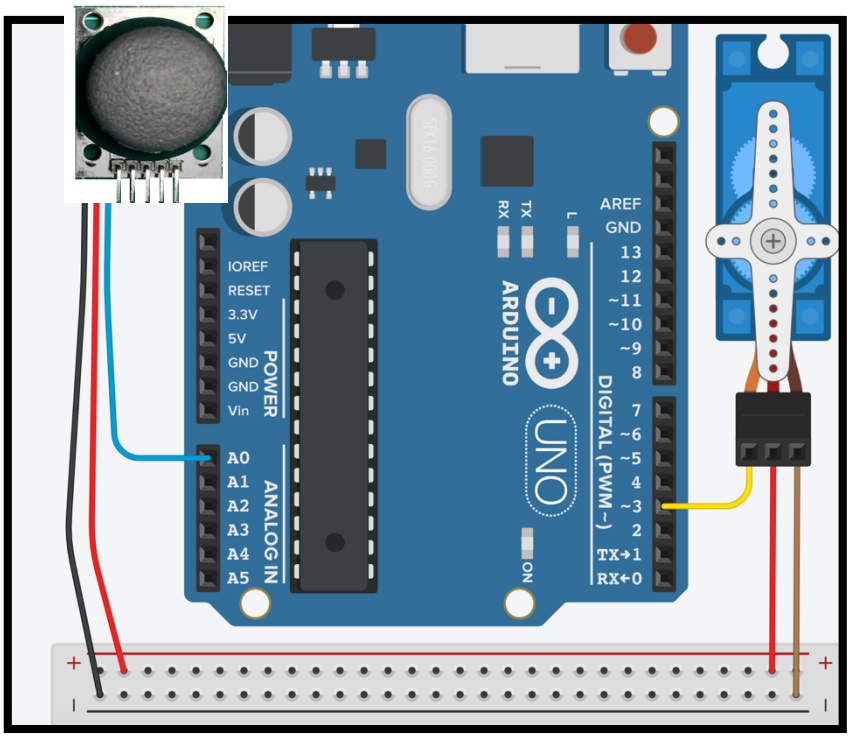
Resim 190. Servo Motor Seti

Servo motor ile hareket ettirilmek istenen nesneler servo motora bağlantı aparatları aracılığı ile bağlanır. Ayrıca Arduino ile servo motor arasındaki iletişim üç adet bağlantı ucu içeren pin ile sağlanır. Bağlantı uçlarından kahverengi olan GND, kırmızı olan ise 5V bağlantı uçlarıdır. Bu uçlar ile servo motora enerji verilir. Sarı uç ise sinyal ucudur. Sinyal ucu ile servo motorun dönme derecesi iletilir. Bu etkinlikte servo motor joystick ile kontrol edilecektir. Bunun için öncelikle aşağıdaki resimde görüldüğü gibi küçük bir vida yardımı ile bağlantı aparatlarından solda yer alan kol servo motora bağlanır.



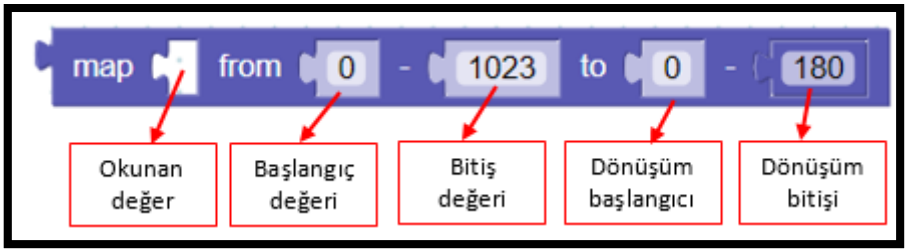
Resim 191. Aparat Bağlantısı

Servo motoru Arduino'ya bağlamak için dijital pinler kullanılabilir. Ancak dijital pinlerin belli bir bölümü bu bağlantılar için uygundur. Uygun olan pinler sırasıyla 3, 5, 6, 9, 10 ve 11 numaralı PWM pinleridir. Bu pinlerin yanında "~" işareti bulunur. Arduino tek seferde bu 6 adet pin aracılığı ile 6 tane servo motoru kontrol edebilir. Etkinlikteki devreyi kurmak için erkek-erkek bağlantı kablolarının birer uçları servo motora bağlanır. Sarı uca bağlı olan kablonun diğer ucu ise yukarıda bahsedilen numaralı pinlerden birine bağlanır. Diğer uçlardan kırmızı olanı 5V pinine, kahverengi olanı ise GND pinine bağlanır. Joystickin de GND ve 5V bağlantıları dişi-erkek kablolar yardımı ile Arduino'nun boşta kalan GND ve 5V pinlerine takılır. Bu etkinlikte jostickin X eksenindeki hareketi ile servo motor kontrol edilecektir. O nedenle joystickin X ucu analog uçlardan birine takılmalıdır. Devre aşağıdaki resimdeki gibi kurulmuş olmalıdır. Güç bağlantılarını yapmak için breadboard kullanılabilir.



Resim 192. Örnek Devre Şeması

Devrenin istenilen şekilde çalışabilmesi için joystickten okunan değerin açısal değere dönüştürülmesi gerekmektedir. Daha önce belirtildiği gibi analog girişler 0 ile 1023 arasında değerleri okumaktadır. Dolayısıyla joystickte bu aralıkta değerler gönderecektir. 0-1023 arasındaki değerleri 0-180 arasındaki değerlere dönüştürmek için Math blok başlığında yer alan *map* komutu kullanılacaktır. Aşağıdaki resimde map bloğu açıklaması görülmektedir.

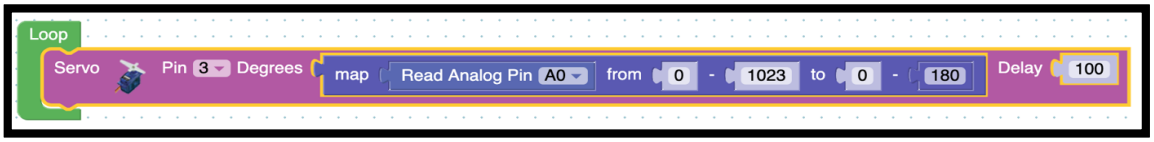


Resim 193. Map Bloğu Açıklaması

Map komutu ile analog girişten okunan 0 ile 1023 arasındaki değer 0-180 derece arasına dönüştürülür ve servo motora gönderilir. Servo motora açı değeri gönderip döndürmek için motor bloğu içindeki Servo komutu kullanılır. Aşağıdaki resimde servo bloğu görülmektedir.

Resim 194. Servo Bloğu

Servo komutunun "Degrees" alanına girilen açı değerine doğru servo motoru döndürülür. Joystickten gelen değerle servo motoru döndürmek için aşağıdaki resimde görülen kod bloğu Arduino'ya yüklenir.



Resim 195. Örnek Program

Öğrencilerin de devreyi kurması ve Arduino kartlarına ilgili kodu yazıp yüklemeleri sağlanır. Bu işlemden sonra öğrenciler ile ikinci bir servo motorun aynı joystick ile kontrol edilip edilemeyeceği sorulur ve sınıfça tartışılır.

1.6 Gözle ve Uygula- Robot Kol Yapımı

Malzeme Listesi:

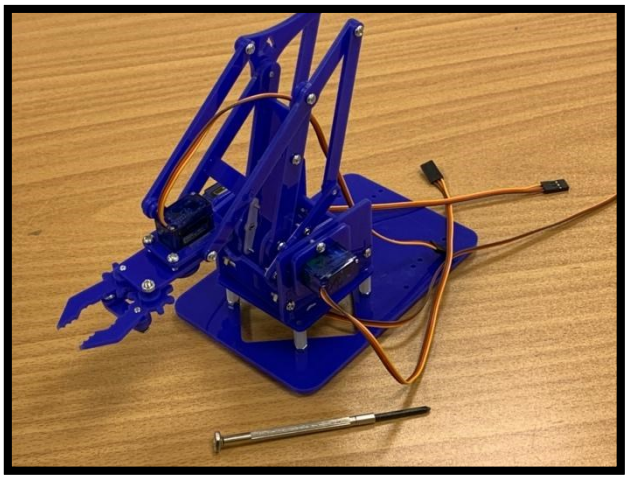
Arduino Uno
Breadboard
Robot kol seti
4 adet servo motor seti
4 adet potansiyometre

Günlük yaşamda kullandığımız birçok eşya ve araç aslında insan eli değmeden bazı işlemleri otomatik şekilde yerine getiren robotlar ile yapılmakta ve üretilmektedir. Robot kelimesi her ne kadar insana benzeyen ve kendi kendine hareket eden araçları çağrıştırsa da seri üretimi yapılan birçok şey pek insana benzemeyen araçlar ile üretilmektedir. Örneğin otomobiller, mobilyalar ve bilgisayarlar bunlara örnek olarak verilebilirler. Bu araçları ve eşyaları üreten robotların özelliği aynı işi hemen hemen yüksek hassasiyetle hiç yorulmadan yerine getirebilmeleridir. Bu etkinlikte amaç ilk olarak potansiyometreler ile kontrol edilen bir mekanik kol oluşturmak ve daha sonrasında belirli bir görevi kendi başına yerine getirebilecek şekilde kodlayarak bu aracı robot kola çevirmektir.

Bu işlem için ilk olarak servo motorların da içinde yer aldığı robot kol setinin montajı yapılmalıdır. Bunun için EK-2'deki Robot Kol Montaj Kılavuzu kullanılabilir. Robot kol montajını yaparken cıvata sıkma ve küçük parçalarla uğraşma gibi el becerileri gerekmektedir. Rehber öğretmen el becerisi yeteri kadar gelişmemiş öğrenciler bulunduğunu düşünüyorsa, bu öğrencileri grubundan ayırıp el becerisi gelişmiş gruplar ile oturtabilir. Bu etkinlik için gruptaki eleman sayısının bir sınırı yoktur. Rehber öğretmen istediği düzenlemeyi yapabilir. Rehber öğretmenler sınıflarında el becerisi yeteri kadar gelişmemiş fazla öğrenci bulunduğunu düşünüyorsa, bu öğrencilerin tamamını bir grup yapabilir. Bu grupta rehber öğretmenlerden biri ortada robot kolun montajını öğrencilere göstererek yapar ve öğrenciler rehber öğretmeni gözleyerek montajı gerçekleştirir. Aşağıdaki resimde robot kol görülmektedir.

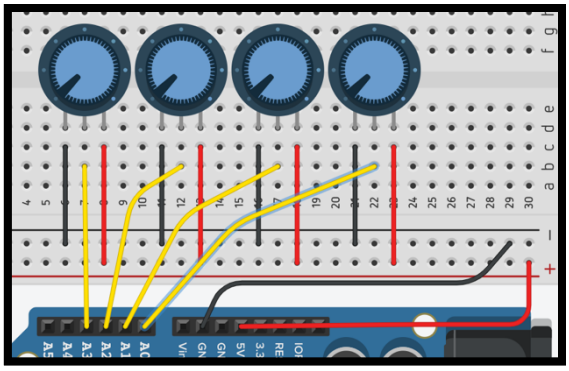
Dikkat

Öğrenciler robot kolun montajını yaparken veya montaj bittikten sonra robot koldaki servo motorları el kuvveti ile döndürmek durumunda kalabilir. Setlerde bulunan servo motorların çarkları plastik olduğundan zorlama durumunda bozulabilir. Rehber öğretmen etkinlik öncesinde ve etkinlik boyunca öğrencileri servo motorlar dönmüyorsa motorların sınır değerde bulunduğu ve zorlayarak döndürmeye çalışmaları durumunda bozulacağı konusunda uyarmalıdır.



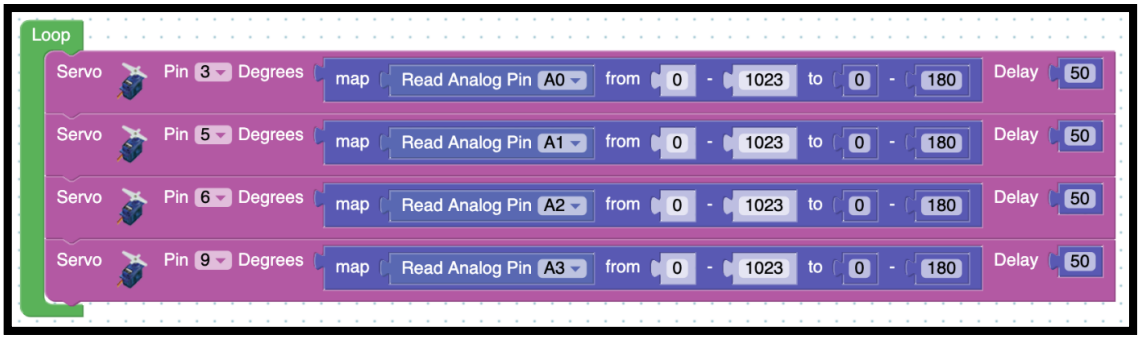
Resim 196. Robot Kol

Daha sonra potansiyometreler ve Arduino ile aşağıdaki resimde verilen devre kurulur. Bu devrede potansiyometrelerin sağ ve sol uçları breadboarda verilen GND ve 5V hatlarına bağlanır. Orta uçları ise Arduino'nun A0'dan A3'e kadar olan analog girişlere sırası ile bağlanır. Bu devrede yer alan her bir potansiyometre robot kolda yer alan bir servo motor ile eşleştirilecektir. Bu eşleştirme için robot koldaki servo motorların sarı bağlantı uçları 3, 5, 6 ve 9 numaralı dijital pinlerinden (PWM) birine takılır. Sıra ile takılması karmaşıklığı azaltır. Servo motorların kırmızı uçları erkek-erkek bağlantı kablolarıyla breadboard'un 5V hattına, kahverengi uçları ise GND hattına bağlanır. Bu şekilde tüm bağlantılar tamamlanmış olur. Örnek devre şeması aşağıdaki resimde görülmektedir.



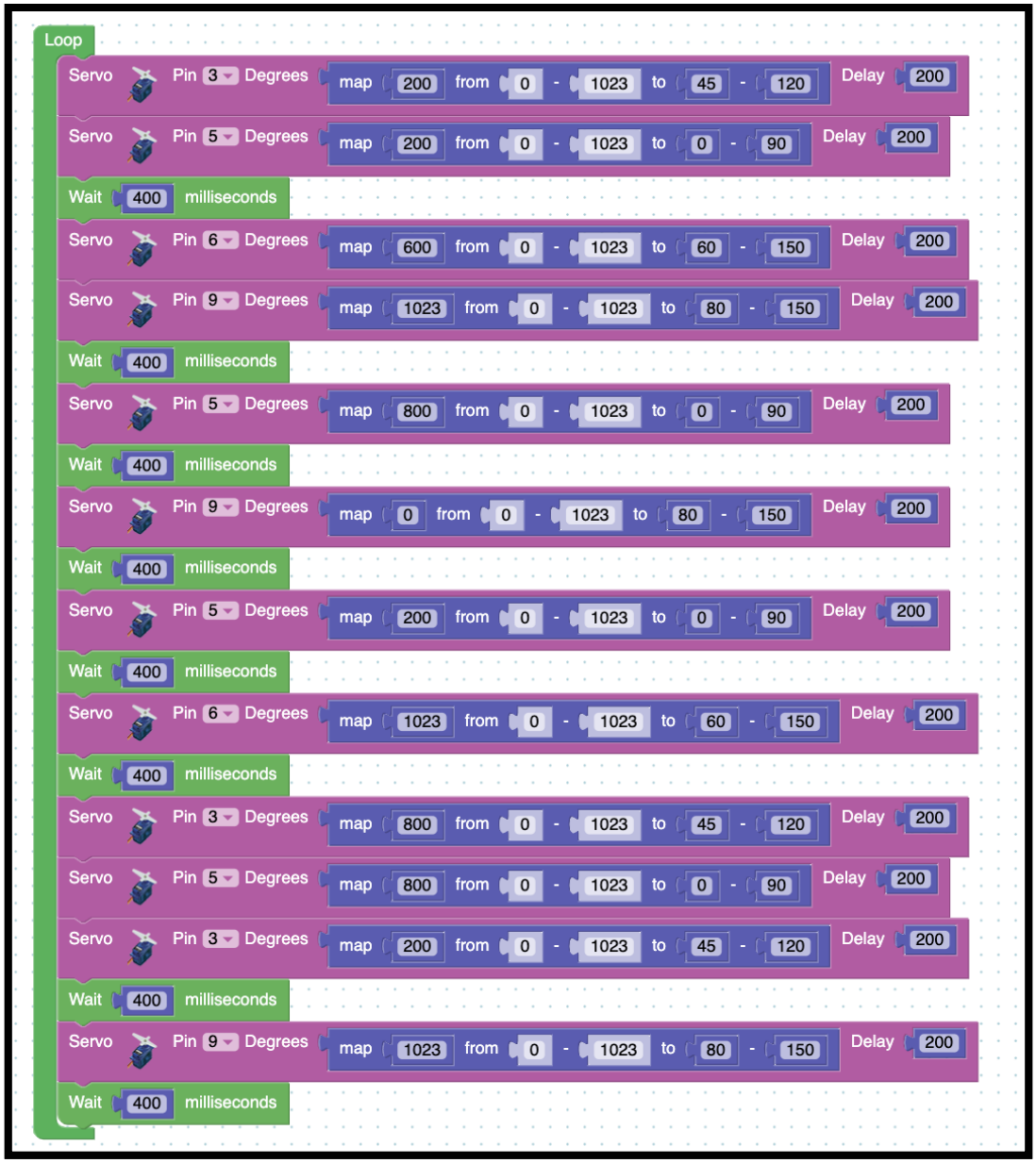
Resim 197. Örnek Devre Şeması

Devrede yer alan üç potansiyometre 1, 2 ve 3 numaralı servo motorların hareketini kontrol edecektir. Son potansiyometre ise robot kolun tutma mekanizmasını açık ve kapalı komuna getirmek için kullanılacaktır. Bunun için aşağıdaki resimde görülen kod yazılmalıdır.



Resim 198. Örnek Program 1

Kod ve devrenin nasıl oluşturulduğu açıklamalar ile öğrencilerle paylaşılmalıdır. Ardından öğrenciler oluşturdukları robot kol ve ilgili devreyi denemelidir. Daha sonra öğrencilere tasarlanan bu kolun neleri yapabilecek kabiliyetlere sahip olduğu sorulur ve sınıfla tartışılır. Tartışma sonunda öğrencilerden robot kolun otomatik olarak yapmasını istedikleri bir hareketi belirlemeleri istenir. Bu hareket öğrenciler tarafından öncelikli olarak potansiyometre ve buton ile gerçekleştirilir. Daha sonra öğrencilerden potansiyometre ve butonları kullanmadan robot kola tasarlanan hareketi yaptıran kodu yazmaları istenir. Aşağıdaki resimde yer alan kod örnek bir hareket için oluşturulmuştur.



Resim 199. Örnek Program 2

Bu süreçte öğrencilere süre verilerek denemeler yapmaları istenir. Gerekirse öğrencilere gerekli kodu yazmaları için yardım edilir. Tamamlayamayan öğrenciler olması durumunda yukarıdaki kod örnek olarak verilir ve öğrenciler tarafından test edilmesi istenir.

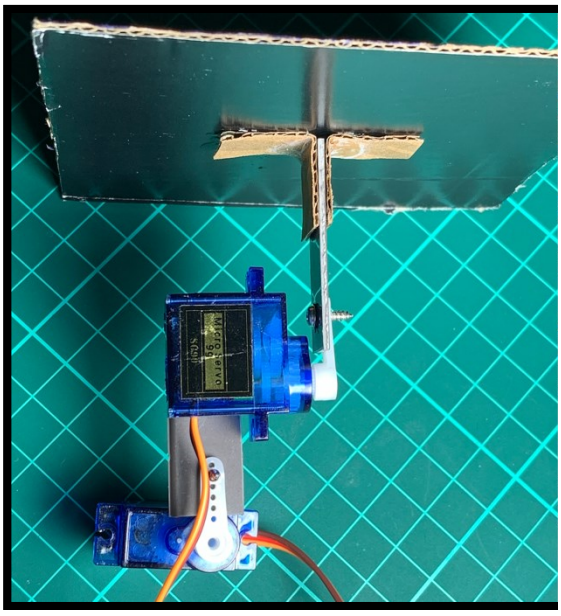
Dikkat

Servo motorlar farklı türlerde olabilir. Bazı servo motorlar 360 derecelik dönüşler yapabilirken bazıları ise daha dar aralıklarda dönüş yapabilirler. Etkinlikte kullanılan servo motorlar 0-180 derece arasında dönüş yapabildiğinden bu değerlerin dışına çıkılmaması önemlidir. Aksi takdirde servo motorlar hasar görebilir.

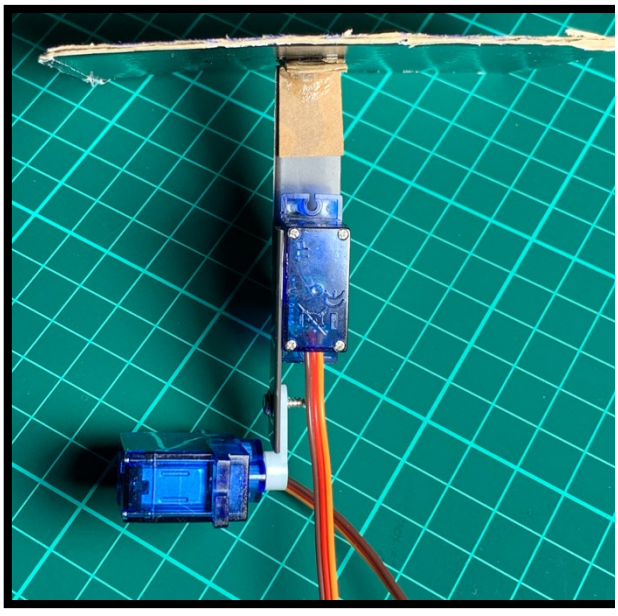
2. TASARLA ve ÜRET

2.1 Tasarla- Joystick ile Labirent Kontrolü

Bu etkinlikte amaç joystick yardımı ile labirent kontrol edilen bir oyun için gerekli yazılım ve donanımı üretmektir. Labirent içerisinde belirlenen başlangıç noktasında küçük bir top/bilye bulunacaktır (**Not**: Rehber öğretmen bu topun/bilyenin ders öncesinde temin edilmesini sağlamalıdır). Labirent sağa-sola veya ileriye-geriye yatırılarak top labirent içerisinde hareket ettirilecek ve bitiş/çıkış noktasına gelmesi sağlanacaktır. Labirentin ileri-geri ve sağ sol yatma hareketleri joystick ile kontrol edilecektir. Bu görev için öğrencilerden 10cm x 10cm kare bir labirent tasarlamaları ve bu labirenti aşağıdaki resimdeki gibi iki boyutta hareket etmesini sağlayacak bir düzenek kurmaları istenir. Aşağıdaki resim temsilidir ve üst kısmı boştur. Öğrencilerin üreteceği labirentte kartonun üst kısmında labirent bulunmalıdır. Aşağıdaki düzenekte iki plastik parça kullanılmıştır. Her bir plastik parça birer servo motor çarkına vidalanmıştır. Daha sonra bir servo motorun plastik parçası diğer servo motorun yan tarafına resimdeki gibi yapıştırılmıştır. Üstte yer alan 10cm x 10cm ebatlarındaki karton bir labirenti temsil etmektedir. Labirentte belirlenen başlangıç noktasına ufak bir bilye ya da top konularak labirentin diğer tarafındaki bir delikten düşmesini sağlayacak (bitiş noktasına gelmesini sağlayacak) hareketlerin de servo motorları kontrol eden bir joystick tarafından gerçekleştirilmesi gerekmektedir. Joystickin X eksen pini ile bir servo, Y eksen pini ile diğer servo motor kullanıldığı taktirde labirent sağa-sola ve ileri-geri eğimli olacak şekilde hareket ettirilebilecektir.



Resim 200. Önden Görünüş



Resim 201. Yandan Görünüş

Joystick ile labirent kolu etkinliğini tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir. Aşağıdaki örnek süreç kesinlikle öğrencilere gösterilmemelidir veya başka bir örnek de sınıfça birlikte yapılmamalıdır. Rehber öğretmen sınıfta genel bir sorun görürse sınıfa açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Tanımlama: Öğrencilerin öncelikle joystick ile labirent kolu etkinliğinin neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrenciler öncelikle gerekli işlemleri detaylı olarak maddeler hâlinde yazmalıdır. Bunun için aşağıdaki sorulara cevap aranabilir.

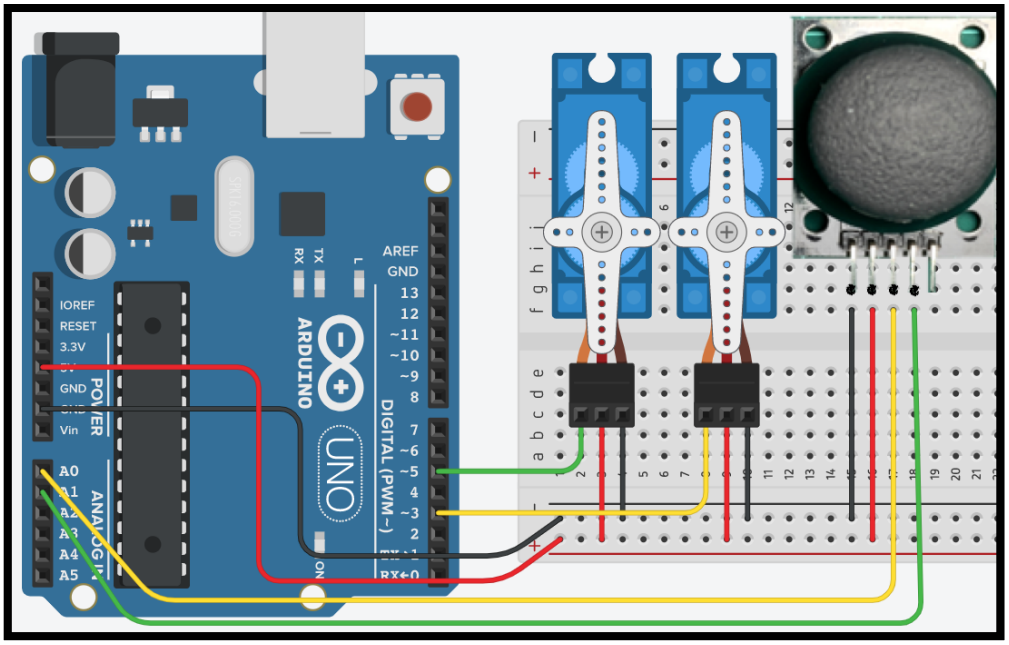
- Servo motorlar nasıl konumlanmalıdır? Hangi doğrultuda hareket etmelidir?
- Hareketli parçalar nasıl oluşturulabilir?
- Servo motorlar birbirlerine nasıl bağlanacak?
- Nasıl bir labirent tasarlanmalı?
- Servo motorlar labirente nasıl bağlanacak?
- Servo motorların hareket açıları ne olmalı?
- Tüm düzenek nasıl sabit tutulacak?

Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Tanımlamada belirlenen her bir adımın gerçekleştirilebilmesi için gerekli fikirler belirlenmelidir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

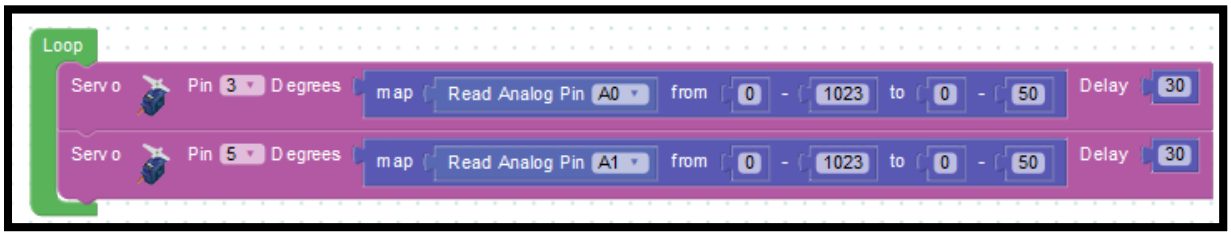
- Servo motorlar birbirine dik doğrultularda hareket etmeli.
- Bir servo motorun hareketli parçası diğer servo motorun gövdesine bağlanabilir.
- Labirent de hareketli kısmı boşta olan servo motora bağlanabilir. Bu şekilde bir servo diğer servo ve labirenti bir eksende hareket ettirir. Diğer servo da labirenti diğer eksende hareket ettirir.
- Bir joystick iki ekseni de kontrol edebilir. İki servo bir joystick ile hareket ederse labirent istenilen şekilde hareket eder.
- Servo motorlar 0 ile 180 derece arasında dönebiliyorlar. Bu çok geniş bir aralık. Bilyenin hareket etmesi için 40-50 derecelik bir hareket yeterli olacaktır.
- Çalışma alanında bant ya da yapıştırıcı ile geçici olarak monte edilebilir.

2.1 Üret- Joystick ile Labirent Kontrolü

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve programa benzer bir içerik hazırlamaları beklenir. Devrede yer alan iki servo motor hazırlanan düzeneğe takılı olan servolardır. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.



Resim 202. Örnek Devre Şeması



Resim 203. Örnek Program

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Potansiyometre gibi direnç seviyesi ayarlanabilen bir araç ile ne tür araçlar kontrol edilebilir? Etkinliklerde joysticklerin de aslında içerisinde potansiyometreler olduğu anlatılmıştır. Temel çalışma prensibi potansiyometre olan başka araçlar olabilir mi?

2. Robot kol etkinliği ile tasarla üret etkinliğindeki (joystick ile labirent kontrolü) benzer noktalar neler olabilir? Mekanik parçalar arasında benzerlikler var mıdır? Servo motorların görevleri arasındaki benzerlikler ve farklılıklar neler olabilir?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- Farklı motor türlerinin çalışma şekillerini, aralarındaki benzerlikleri ve farklılıkları ayırt ederken neler yaşadınız?
- Robot kol montajı sırasında zorlandınız mı? Yönergeleri takip etmek ve parçaları birbirinden ayırt etmek için ne gibi bir yol izlediniz?
- Robot kola otomatik bir hareket yaptırma etkinliğinde neler yaşadınız? Hareketleri belirlemek için belirli bir yöntem izlediniz mi?

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Robot Kol Yazı Yazıyor

Bu etkinlik için 1.6 Gözle ve Uygula başlığında oluşturulan robot kol kullanılacaktır. Bu tasarla-üret etkinliğinde her bir öğrenci oluşturdukları robot kol ile belirledikleri 3 harfi bir kâğıda yazdırabilecekleri kodu oluşturmalıdır. Bunun için robot kolun bir kalemi tutması gerektiği unutulmamalıdır. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler sonra bilgisayar ve robot kol başında çalışarak istenen görevi yerine getiren programı oluştururlar. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

6. Bölüm-DC Motorlar, Kızıl Ötesi ve Mesafe Sensörleri

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, RGB led, PWM, transistor, step motor, servo motor, joystick, robotik kol ve kondansatör kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.

Bölümün Kazanımları:

- DC motor sürmek için gerekli elektronik devreleri oluşturabilir.
- DC motor içeren elektronik devreleri programlayabilir.
- 2 DC motor içeren araba oluşturabilir.
- 2 DC motor içeren araba oluşturabilir programlayabilir.
- Mesafe sensörünü kullanabilmek için gerekli elektronik devreleri oluşturabilir.
- Mesafe sensörü içeren elektronik devreleri programlayabilir.
- Kızıl ötesi sensörü kullanabilmek için gerekli elektronik devreleri oluşturabilir.
- Kızıl ötesi sensörü içeren elektronik devreleri programlayabilir.
- LCD kullanmak için gerekli elektronik devreleri oluşturabilir.
- LCD içeren elektronik devreleri programlayabilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin DC motor, mesafe sensörü, kızıl ötesi sensör, LCD ve mobil robot platformu bileşenlerini elektronik devrelerde kullanabilmelerini ve bu elektronik devreleri ArduinoBlocks ortamında programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, L298N sürücü modülü, 2 adet DC motor, 9V pil, 1 adet 10K Potansiyometre, 4WD çok amaçlı mobil araba platformu, HC-SR04 mesafe sensörü, Mesafe Sensörü Montaj Aparatı Tip C, 1 adet buzzer, 3 adet kızılötesi sensör ve sensör kartı, Robotik kodlama dersindeki mat, 2x16 LCD ekran- I2C dönüştürücü kartı, 4x20 Mavi LCD ekran- I2C dönüştürücü kartı (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Elektronik devrelerde DC motor, mesafe sensörü, kızılötesi sensör, LCD kullanımının ve ArduinoBlocks ortamında programlanmasının gösterilmesi. 4WD çok amaçlı mobil robot platformu ile araba oluşturulması ve bu arabanın ArduinoBlocks ortamında çeşitli sensörlerle birlikte kullanımı için gerekli programlamanın gösterilmesi.

Uygula: Elektronik devrelerde DC motor, mesafe sensörü, kızılötesi sensör, LCD kullanımının ve ArduinoBlocks ortamında programlanmasının yapılması. 4WD çok amaçlı mobil robot platformu ile araba oluşturulması ve bu arabanın ArduinoBlocks ortamında çeşitli sensörlerle birlikte kullanımı için gerekli programlamanın yapılması.

Tasarla: LCD'de animasyon gösterimi, otoparkın katlarındaki araç sayısını gösteren temsili otopark devrelerinin ve programlarının yapılması, aracın belirli bir alana park etmesi, dairenin içinden çıkmayan arabanın yapılması ve üçüncü çizgiyi geçince duran araba etkinlikleri için gerekli tasarımın yapılması.

Üret: LCD'de animasyon gösterimi, otoparkın katlarındaki araç sayısını gösteren temsili otopark devrelerinin ve programlarının yapılması, aracın belirli bir alana park etmesi, dairenin içinden çıkmayan arabanın yapılması ve üçüncü çizgiyi geçince duran araba etkinlikleri için gerekli elektronik devrelerin ve programların yapılması

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

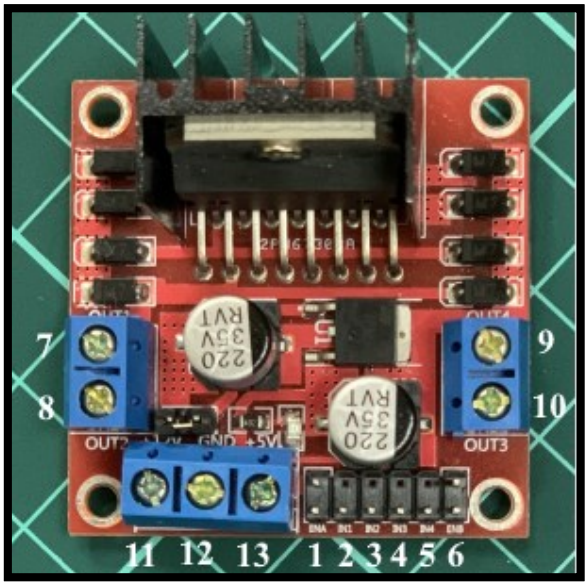
1. GÖZLE VE UYGULA

1.1 Gözle ve Uygula- L298N'yi Kullanarak DC Motor Sürüyorum

Malzeme Listesi:

Arduino Uno
Breadboard
L298N sürücü modülü
2 adet DC motor
Şarj Edilebilir 9 V 200 mAh Pil

Bu etkinlikte amaç L298N modülünü kullanarak iki adet DC motor sürmektir. Devre kurulmadan önce motor sürücü tanıtılacaktır. Daha önce DC motorları sürmek için doğrudan Arduino Uno kartın bacaklarının kullanılamayacağı anlatılmıştır ve DC motor sürmek için bir transistor devresi kullanılmıştır. DC motorları sürmek için çeşitli sürücü modülleri bulunmaktadır. Bu modüller kullanıcılar tarafından da üretilebilir. Fakat burada L298N isimli sürücü modülü kullanılacaktır. Bu modül kullanılarak hem DC hem de step motor sürülebilir. Fakat burada sadece DC motor sürme işleminden bahsedilecektir. Aşağıdaki resimde bir L298N modülü görülmektedir.

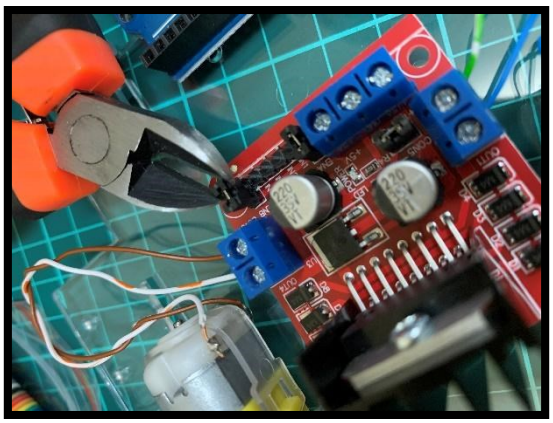


Resim 204. L298N Modülü

L298N modülü ile iki farklı çıkıştan DC motor sürülebilir. 7-8 ve 9-10 numaralı kanallar/bacaklar DC motora güç verilen bacaklardır. Bu kanallara DC motor bağlanabilir. Her bir kanal 2 Amper'e kadar akım sağlayabilir. Her iki kanal ayrı ayrı kullanılabileceği gibi aynı anda da kullanılabilir. Kanallardan birine bağlanmış DC motorunun bağlantıları ters yönde olacak şekilde değiştirilirse motorun dönme yönü de değişir. 1 (ENA), 2 (IN1) ve 3 (IN2) numaralı bacaklar sol DC motoru (7-8 kanalı) ve 4 (IN3), 5 (IN4), ve 6 (ENB) numaralı bacaklar sağ DC motoru (9-10 kanalı) sürmek için kullanılır. 2 ve 3 numaralı bacaklar sol motorun dönüş yönünü belirler. 2 numaralı bacak ON (5V) ve 3 numaralı bacak OFF (0V) iken DC motor ileri yönde döner. 2 numaralı bacak OFF ve 3 numaralı bacak ON iken DC motor geriye doğru döner. 4 ve 5 numaralı bacaklar aynı görevi sağ DC motor için üstlenir. 1 numaralı bacak sol motorun dönüş hızını belirlemek için kullanılır. Bu bacağa enable bacağı da denilir. Enable bacağı Arduino Uno'nun PWM pinlerinden birine bağlanmalıdır. Aksi halde motorun dönüş hızı ayarlanamaz. PWM bacağından gelen 0-255 arasındaki değere göre motorun dönüş hızı ayarlanır. Fakat belirli düşük değerler için motor dönmeyip bip sesi çıkarabilir. Bu değerler için motor döndürülemez. 6

numaralı bacak ise sağ motorda aynı görev (motor dönüş hızını belirleme) için kullanılır. Bu bacağa da enable bacağı denilir.

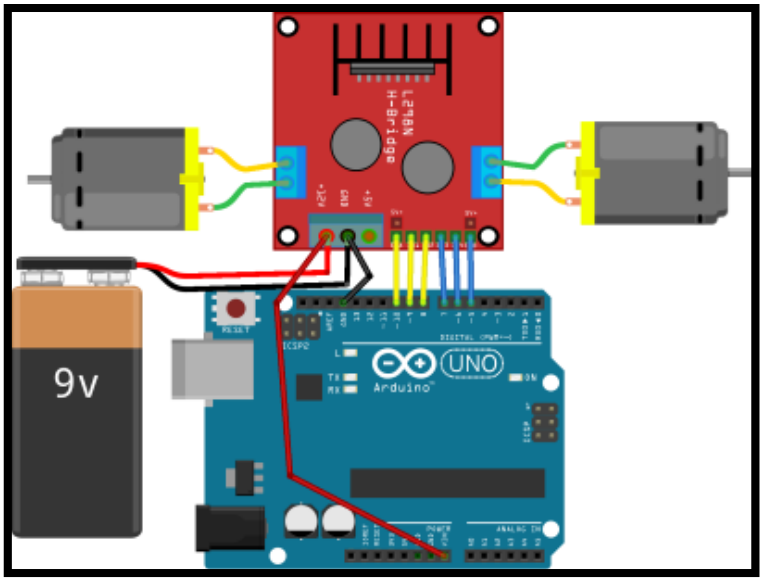
L298N ilk satın alındığında 1 ve 6 numaralı bacakta jumperlar bulunmaktadır. Bu durum aşağıdaki resimde gösterilmiştir. Bu jumperlar çıkarılmazsa motorlar için hız ayarlaması yapılamaz. Jumperlar takılı olduğunda motorlar sadece en yüksek hızda çalışır. Ara değerlerde hız üretilmez. Burada yapılacak etkinliklerde hız ayarlaması yapılacağı için jumperların ikisi de çıkarılmıştır. Rehber öğretmen öğrencilerden bu jumperları söküp kaybolmayacak şekilde organizerlarının içine koymasını sağlamalıdır.



Resim 205. Jumper Sökme

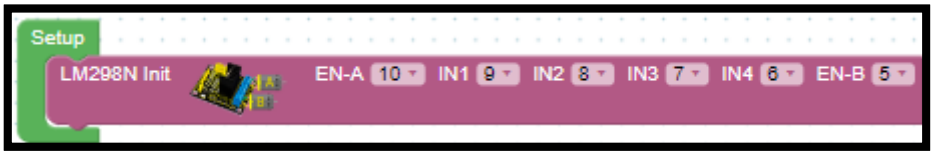
L298N modülünün 11 numaralı bacağına doğru akım güç kaynağının (burada 9V pil kullanılacaktır) + ucu bağlanır. 12 numaralı bacağa ise güç kaynağının - ucu bağlanır. Burada 4.8-24V arası güç kaynağı kullanılabilir. Bu değerler kullanılan L298N modülüne göre farklılık gösterebilir. Bazı modellerde 35V'a kadar güç kaynağını kullanılabilir. Modül için kullanılabilecek maksimum gerilim için modülün kılavuzuna başvurunuz. Son olarak 13 numaralı kanal 5V çıkış almak için kullanılabilir. Bu kanal vasıtasıyla Arduino kartı beslenebilir. Ayrıca bu kanal 5V giriş için de kullanılabilir. 13 numaralı kanaldan 5V giriş yapıldığında modül içerisindeki mantık devresi beslenilir. 13 numaralı kanal giriş için kullanılacaksa 11-12 kanalının arkasında bulunan jumper çıkarılmalıdır. Buradaki etkinlikler için bu jumper'ın çıkarılmasına gerek yoktur.

Aşağıdaki resimde L298N ile iki DC motorun sürülmesine yönelik devre gösterilmiştir. Rehber öğretmen devreyi anlatarak öğrenciler karşısında kurar.



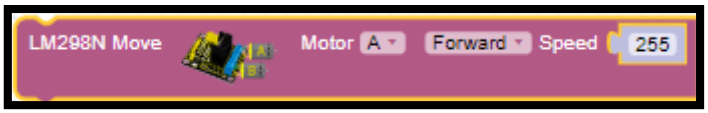
Resim 206. Örnek Devre Şeması

Bu devre için ArduinoBlocks'da kod yazmak için öncelikle L298N'nin iki motor için bağlantılarının nasıl yapıldığı belirtilmelidir. Devrede sol motorun ENA (1 numaralı), IN1 (2 numaralı) ve IN2 (3 numaralı) bacakları Arduino'nun 10, 9 ve 8 numaralı bacaklarına bağlanmıştır. Sağ motorun IN3 (4 numaralı), IN4 (5 numaralı) ve ENB (6 numaralı) bacakları ise Arduino'nun 7, 6 ve 5 numaralı bacaklarına bağlanmıştır. Bu durum ArduinoBlocks ortamında aşağıdaki resimde görüldüğü gibi ifade edilir. "LM298N Init" bloğu Motor blok başlığı altında bulunur.



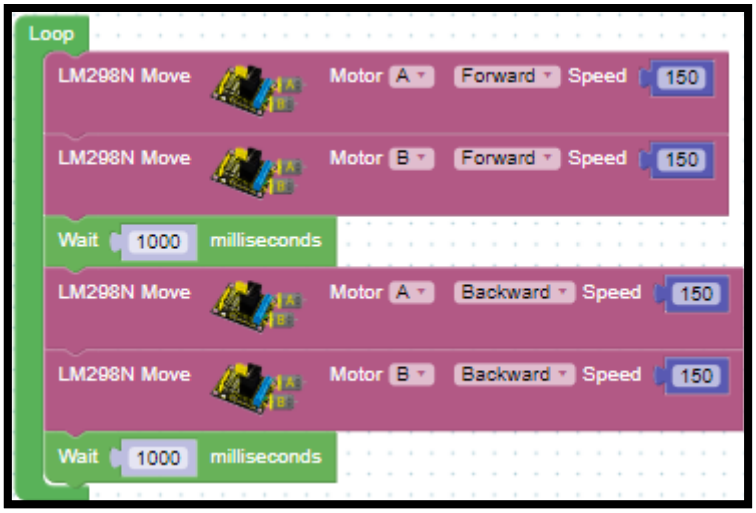
Resim 207. Setup Ayarı

Motorları hareket ettirmek için gerekli "*LM298N Move*" komutu Motor blok başlığı altında bulunmaktadır. Bu blokta üç parametre bulunur. Birinci parametre sol veya sağ motordan hangisinin kullanılacağını belirtir. İkinci parametre ileri (Forward) veya geri (Backward) hareket için kullanılır. Üçüncü parametre hızı belirlemek için kullanılır. Bu parametre için 0-255 arasında bir değer girilebilir. Fakat belirli değerlerin altındaki hızlar için motor dönmeyebilir. Aşağıdaki resimde move bloğu görülmektedir.



Resim 208. L298N Move Bloğu

Aşağıdaki resimde görülen kod ile sağ ve sol motorlar sürekli olarak 1 saniye boyunca ileri gidip ardından 1 saniye geriye doğru hareket etmektedir. Rehber öğretmen bu kodu öğrencilere anlatarak yazar.



Resim 209. Örnek Program

Rehber öğretmen devrenin çalışmasını öğrencilere gösterdikten sonra öğrencilerin aynı devreyi kurmalarını ve kodu yazıp çalıştırmalarını sağlar.

Dikkat

Yukarıdaki kod çalıştırıldığında motorlardan birisi ileri diğeri geri dönüyorsa bu durumu düzeltmek için 7-8 veya 9-10 kanallarından bir tanesinin bağlantısı ters yapılabilir veya "*LM298N Init*" bloğunda IN1 8, IN2 ise 9 yapılabilir. Bunun yerine IN3 ve IN4'e karşılık gelen Arduino bacak numaraları yer değiştirilebilir. Bu yapılacakların hepsi iki motorun aynı yönde hareket etmesini sağlar.

1.1 Uygula- Motorları Ters Yönde Döndürüyorum

Malzeme Listesi:

Arduino Uno
L298N sürücü modülü
2 adet DC motor

Şarj Edilebilir 9 V 200 mAh Pil

Bir önceki etkinlikte sağ ve sol motor aynı anda ileri ve geri hareket etmiştir. Bu etkinlikte amaç sağ ve sol motoru son hızda ters yönde hareket ettirmektir. Bu görev için aşağıdaki adımlar takip edilebilir.

i. Sol motor en yüksek hızda (255) ileriye; sol motor en yüksek hızda geriye doğru bir saniye boyunca hareket eder,

ii. Sol motor en yüksek hızda (255) geriye; sağ motor en yüksek hızda geriye doğru bir saniye boyunca hareket eder,

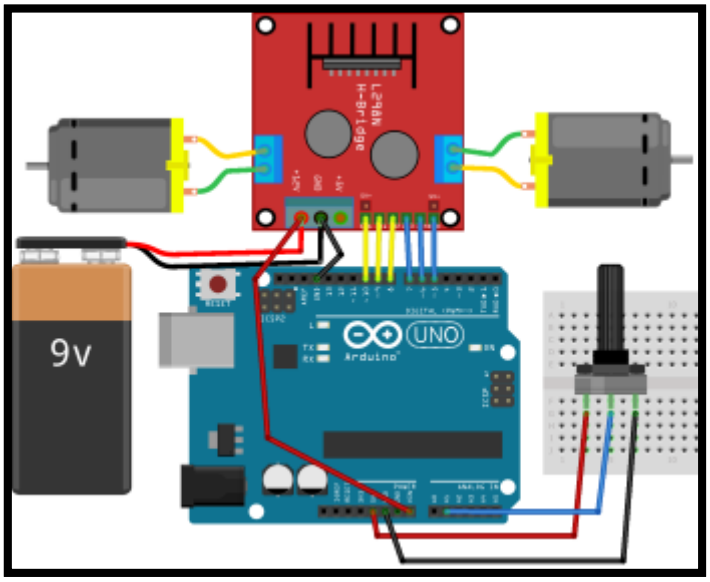
iii. i ve ii işlemleri sürekli tekrarlanır.

1.1 Uygula- Motorları Potansiyometre ile Farklı Hızlarda Döndürüyorum

Malzeme Listesi:

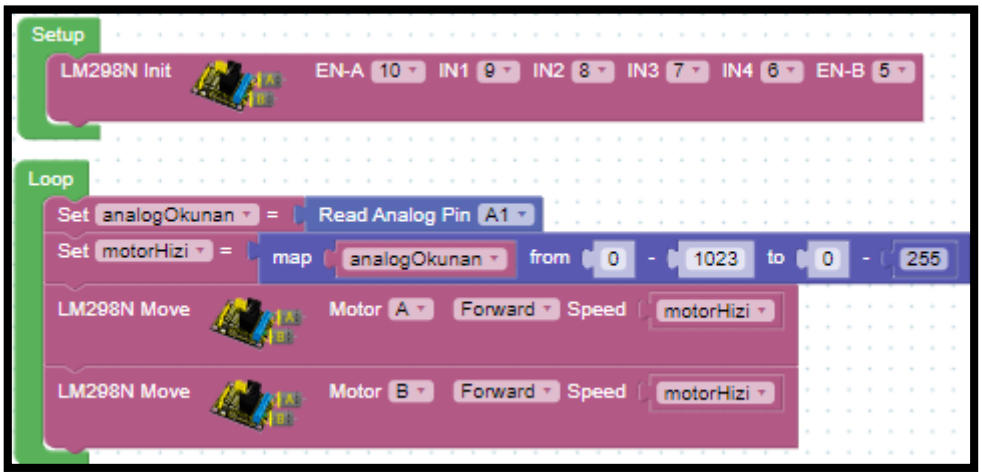
Arduino Uno
L298N sürücü modülü
2 adet DC motor
Şarj Edilebilir 9 V 200 mAh Pil
1 adet 10K Potansiyometre

Bu etkinlikteki amaç potansiyometre aracılığı ile motorların hızını belirlemektir. Potansiyometrenin kolu bir yöne çevrilince motorların hızı artmalıdır diğer yönde çevrildiğinde ise motorların hızı azalmalıdır. Bu etkinlik için aşağıdaki resimde görülen devre kullanılabilir.



Resim 210. Örnek Devre Şeması

Aşağıdaki resimde verilen kod bu görev için kullanılabilir. Potansiyometrenin orta bacağından gelen değer A1 analog pininden okunmuştur ve buradan okunan değer map komutu ile 0-255 aralığına aktarılmıştır. Böylece potansiyometreden okunan değer ile motor hızları ayarlanmıştır. Motorlara verilen hız belirli bir değerin altına düştüğünde motorlar hareket etmez ve bip benzeri bir ses çıkarır. Yapılan ölçümlerde motora verilen hız değeri 75-80 aralığından daha az olduğunda motorların dönmediği görülmüştür. Bu değerler rehber öğretmenlerin/öğrencilerin durumları için farklılık gösterebilir. Yazılan kodlarda bu durum göz önünde bulundurulmalıdır.



Resim 211. Örnek Program

1.2 Gözle ve Uygula- 2 Motorlu Araba Yapıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
L298N sürücü modülü
4WD çok amaçlı mobil araba platformu
Şarj Edilebilir 9 V 200 mAh Pil

Bu etkinlikte amaç 2 teker ve 1 sarhoş teker içeren 2 motorlu araba inşa etmek ve bu arabayı ileri-geri hareket ettirmektir. Arabanın nasıl oluşturulacağı EK-3'teki Araba Kurulum Kılavuzunda adım adım anlatılmıştır. Rehber öğretmen öğrencilere göstererek arabanın kurulumunu gerçekleştirir. Aynı zamanda öğrencilerin arabalarını kurmasını da sağlar. Bütün öğrenciler arabayı kurduktan sonra aşağıdaki resimde görülen kod ile araba ileri-geri hareket ettirilir.

Resim 212. Örnek Program

Dikkat

Kod çalıştırıldığında arabanın tekerlekleri farklı yöne dönüyor ise (bu durumda araba dönme hareketi yapacaktır) L298N'in motor bağlantılarından birisi ters yapılmıştır. Motor bağlantısı düzeltilebilir. Ya da "*LM298N Init*" komutunda IN1 ile IN2 bacaklarının yerleri değiştirilebilir (ya da IN3 ile IN4 bacaklarının yerleri değiştirilir).

Bazı arabalarda DC motorlardaki hata payından ötürü arabalar bir yöne doğru çekebilir. Bu durumda öncelikle araba sabit bir hızla ilerletilerek arabanın çektiği yön bulunur. Arabanın düz gitmesi için çektiği tarafın tersinde bulunan motora daha az hız verilebilir. Örneğin başlangıçta iki motora da 150 hız verildiği düşünülsün. Araba sağa doğru çekiyorsa sol motora verilen değer 135'e düşürülebilir. Bu değer örnek olarak verilmiştir. Her grup kendi durumuna göre ayarlama yapmalıdır.

1.3 Gözle ve Uygula- Mesafe Sensörünü Kullanıyorum

Malzeme Listesi:

Arduino Uno
HC-SR04 mesafe sensörü
Ultrasonik Sensör Montaj Aparatı Tip C

Mesafe sensörü adından da anlaşılacağı gibi karşısında bulunan bir nesnenin sensöre olan mesafesini bulmak için kullanılır. Mesafe sensörüne ultrasonik sensör de denilir. Mesafe sensörünün iki gözü bulunmaktadır. Mesafe sensörü bir gözden insan kulağının duyamayacağı ses dalgaları gönderir ve diğer gözden karşıda bulunan nesneden, eğer böyle bir nesne varsa, yansıyan ses dalgalarını alır. Ses dalgalarını gönderme ve yansıyan dalgaları alma arasında geçen süre hesaplanarak karşıdaki cismin mesafesi bulunur. Aşağıdaki resimde bir HC-SR04 ultrasonik sensör gösterilmiştir. Rehber öğretmen öğrencileri mesafe sensöründen hesaplanan değerin ortamın sıcaklık ve nem değerlerine bağlı olarak değişiklik gösterebileceği konusunda uyarır. Mesafe sensörü ile ölçülen mesafe değerinin kuru havada yaklaşık olarak oda sıcaklığı için geçerli olduğu bilgisi verilir. Eğer mesafe sensöründen gelen değer soğuk bir ortamda hesaplanırsa bulunacak değerin tam olarak doğru olmayacağı vurgulanır. Burada yapılan etkinlikler oda sıcaklığında olduğu için mesafe değeri genel olarak doğru hesaplanır. Aşağıdaki resimde mesafe sensörü görülmektedir.

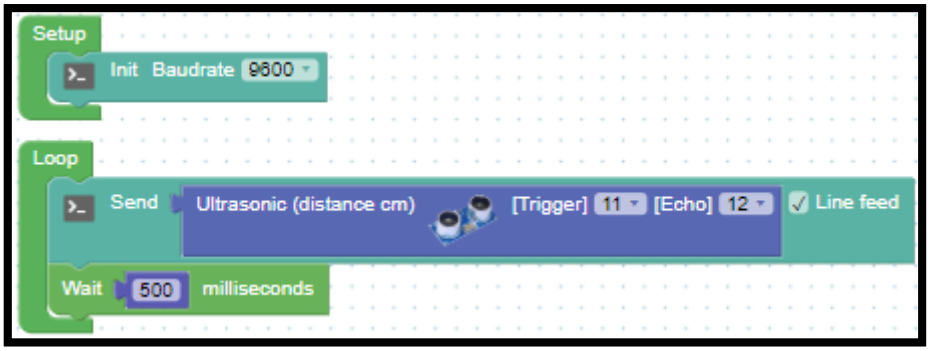
Resim 213. Mesafe Sensörü

Mesafe sensörünün 4 bacağı bulunur bunlar "VCC, Trig, Echo ve GND" bacaklarıdır. VCC bacağına Arduino'nun 5V çıkışı bağlanır. GND bacağına Arduino'nun GND bacağı bağlanır. Echo ve Trig bacakları Arduino'nun dijital pinlerinden herhangi birine bağlanabilir. Bu etkinlikte Echo 12 numaralı pine, Trig ise 11 numaralı pine bağlanmıştır. ArduinoBlocks ortamında bulunan "Ultrasonic (distance cm)" bloğu otomatik olarak mesafeyi hesaplar. ArduinoBlocks ortamında programcılar başka bir kod yazmadan (Arduino IDE'de programcılar bu değeri kendileri hesaplatmalıdır) mesafeyi hesaplamış olur. Ultrasonik sensör 2cm ve 400cm arasındaki değerleri ölçebilir. Sensör bu değerler dışında kalan ölçümleri yapamaz. Sensör eğer karşısında bir engel yoksa 0 değeri döndürür. Ultrasonik sensör genel olarak hassas hesap yapar. Fakat, her ne kadar kataloğunda belirtilmese de ultrasonik sensör çok yakın ve çok uzak değerlerde hatalı sonuçlar üretebilir. Bu yüzden alt ve üst sınıra yakın olan değerlerde daha dikkatli kullanılmalıdır. ArduinoBlocks ortamında mesafe hesaplamak için kullanılacak olan "Ultrasonic (distance cm)" bloğu "Sensors" blok başlığı altındadır. Aşağıda bu bloğun resmi verilmiştir. Bu blok ultrasonik sensörden gelen değeri alarak ultrasonik sensör ve cisim arasındaki mesafeyi cm cinsinden hesaplar.

Resim 214. Mesafe Ölçüm Bloğu

Bu etkinlikte basitçe ultrasonik sensörden alınan değerler seri porta aktarılacak ve değerlerin doğruluğu farklı uzaklıklar ve cisimler için test edilecektir. Bu görev için aşağıdaki resimde görülen kod

kullanılabilir. Bu kod ultrasonik sensör ile karşısında bulunan cismin arasındaki mesafeyi yarım saniye aralıklarla seri port ekranına yazmaktadır.



Resim 215. Örnek Program

Rehber öğretmen açıklayarak devreyi kurar ve kodu anlatır. Ardından öğrencilerin de devreyi kurup, kodu yazarak çalıştırmalarını sağlar. Farklı cisimler ve uzaklıklar için programın ürettiği değerlerin öğrenciler tarafından gözlemlenmesi sağlanır. Burada ortaya çıkan farklılıklara dikkat çekilir.

Dikkat

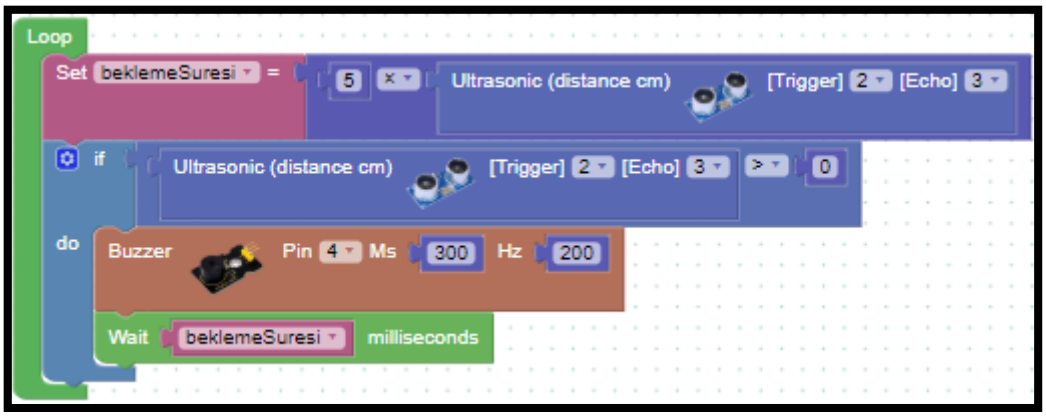
Bu etkinlik için Robotik Kodlama dersinde kullanılan mat üzerindeki metre kullanılabilir.

1.3 Uygula- Mesafe Sensörüne Bağlı Olarak Buzzer ve Led Kullanımı

Malzeme Listesi:

Arduino Uno
HC-SR04 mesafe sensörü
1 adet buzzer

Bu etkinlikte amaç ultrasonik sensörün karşısında bulunan cisme yaklaştıkça buzzerdan çıkan sesin sıklığını artırmaktır. Bu görev için aşağıdaki resimde görülen kod bloğu kullanılabilir.



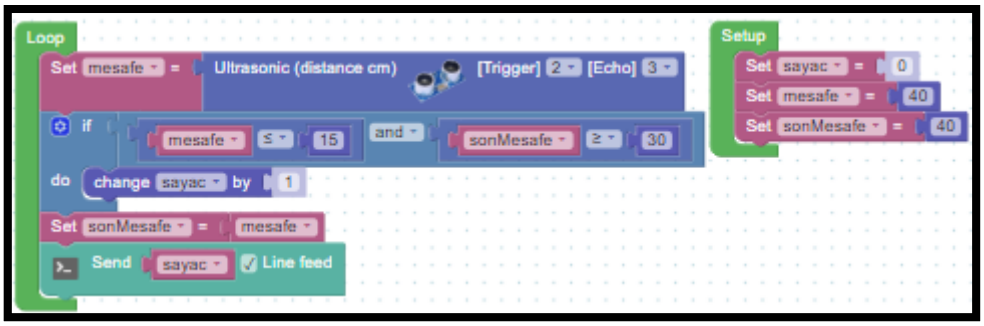
Resim 216. Örnek Program

1.3 Uygula- Ultrasonik Sayaç

Malzeme Listesi:

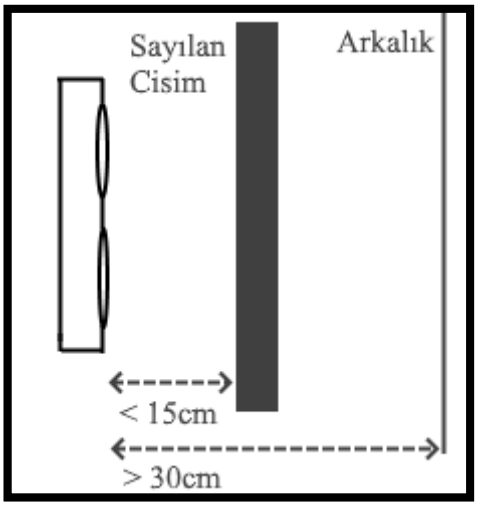
Arduino Uno
Breadboard
HC-SR04 mesafe sensörü
1 adet buzzer

Bu etkinlikte amaç ultrasonik sensörün karşısından geçen nesnelerin sayısını bulup seri port ekranına yazmaktır. Görevin yerine getirilebilmesi için cismin sensör ile bir arkalık arasından geçmesi gerekmektedir. Bu görev için bir sayaç oluşturulur ve her yeni gelen nesne için bu sayaç bir artırılır. Aşağıdaki resimde örnek bir çözüm verilmiştir (*Trig* 2 nolu ve *Echo* 3 nolu pine bağlanmıştır).



Resim 217. Örnek Program

Verilen kodun çalışması için ultrasonik sensör ile cisim arasındaki mesafe 15 cm'den kısa olmalıdır. Aşağıdaki resimde gösterildiği gibi konulacak olan arkalık da 30 cm'den biraz fazla olmalıdır.



Resim 218. Ortam Tasarımı

1.4 Gözle- Ultrasonik Sensörü Arabaya Ekliyorum

Malzeme Listesi:

Arduino Uno
Breadboard
L298N sürücü modülü
4WD çok amaçlı mobil araba platformu
Şarj Edilebilir 9 V 200 mAh Pil
1 adet PV pil
HC-SR04 mesafe sensörü

Rehber öğretmen öğrencilere anlatarak, EK-3'te (Araba Kurulum Kılavuzu) gösterildiği gibi, ultrasonik sensörü arabaya yerleştirir. Bu etkinlikte amaç arabanın karşısında bulunan duvara yavaşlayarak yaklaşmasını ve duvara 10 cm mesafe kalınca durmasını sağlamaktır. Bu görev için aşağıdaki resimde görülen kod kullanılabilir.

Resim 219. Örnek Program

Dikkat

Burada arabanın bir tarafa doğru çekmediği kabul edilmiştir. Çeken arabalar için hız ayarlaması yapılmalıdır.

1.4 Uygula- Yavaşlarken Bip Sesi Çıkaran Araba

Malzeme Listesi:

Arduino Uno
L298N sürücü modülü
4WD çok amaçlı mobil araba platformu
Şarj Edilebilir 9 V 200 mAh Pil
HC-SR04 mesafe sensörü
1 adet buzzer

Bu etkinlikte bir önceki etkinliğe ek olarak bir buzzer eklenecektir. Araba duvara yavaşlayarak yaklaşacaktır. Araba duvara yavaşlayarak yaklaşırken buzzerdan hızı artan bir şekilde bip sesi çıkacaktır. Son olarak duvara 10 cm kala araba duracaktır.

1.5 Gözle ve Uygula- Engel Algılayan Kızılötesi Sensor

Malzeme Listesi:

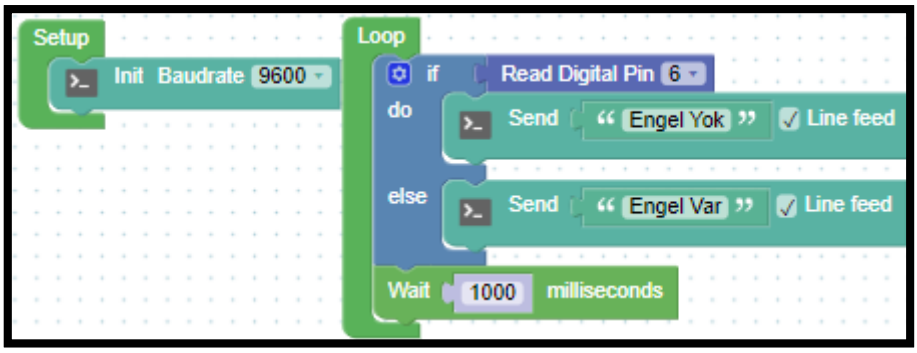
Arduino Uno
Breadboard
Kızılötesi sensör

Kızıl ötesi alıcı ve verici barındıran sensörler kullanılarak da karşıda bulunan engeli algılama görevi yerine getirilebilir. Kızılötesi engel sensörlerinin birden fazla çeşidi bulunur. Fakat bu sensörlerin genel

çalışma prensibi birbirlerine benzer. Kızılötesi verici kızıl ötesi sinyal gönderir ve kızılötesi alıcı karşıda bulunan cisimden yansıyan sinyalleri yakalar. Kızıl ötesi verici karşıya doğru sinyal gönderir, eğer karşıda bir engel varsa engelden yansıyan kızıl ötesi ışık alıcıya geri döner ve kızıl ötesi sensör 0 (OFF) değerini üretir. Eğer karşıda bir engel yoksa kızılötesi sensör 1 değerini üretir. Sensörden 1mm ile 10cm arasında bulunan uzaklıklarda bulunan cisimlere tepki verebilir (cismin rengi bu değerleri etkileyebilir). Aşağıdaki resimde kızılötesi sensör gösterilmiştir. Kızılötesi sensörün üç bacağı bulunur. Bunlar VCC (5V'a bağlanır), GND (GND'ye bağlanır) ve OUT (herhangi bir dijital girişe bağlanır). VCC ve GND bacaklarından 5V gerilim uygulanır ve sensör eğer engel varsa OUT bacağından 0 ve engel yoksa 1 değeri üretir.

Resim 220. Kızılötesi Sensör

Bu etkinlikte kızılötesi sensör kullanılarak karşıda bir engel yoksa seri port ekranında "Engel Yok" eğer engel varsa "Engel Var" yazan program yazılacaktır. Engelin renginin beyaz olması önerilir. Bu görev için aşağıdaki resimde görülen kod kullanılabilir. Bu kodun çalışması için sensörün OUT bacağı Arduino kartının 6 numaralı bacağına bağlanmış olmalıdır. 6 numaralı bacak yerine herhangi bir dijital giriş kullanılabilir.



Resim 221. Örnek Program

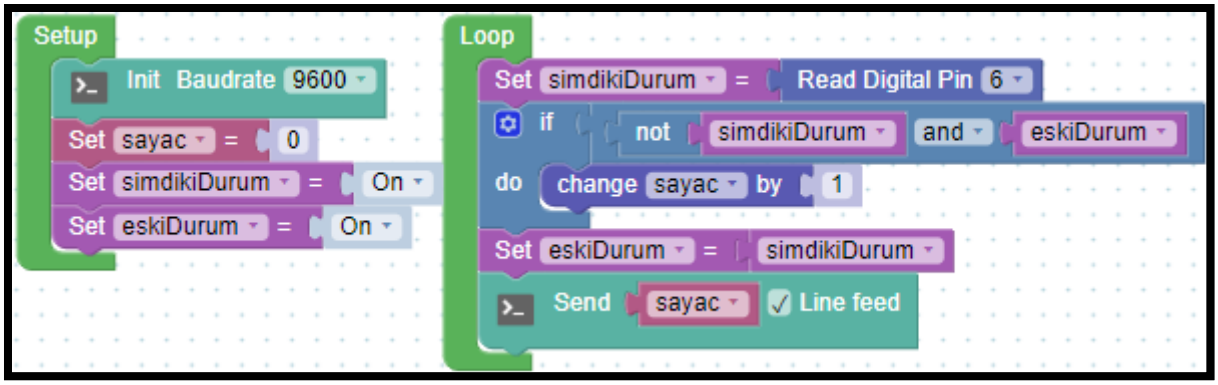
Rehber öğretmen öğrencilere göstererek devreyi kurar ve kodu öğrencilere anlatarak yazar. Kodu çalıştırıp sonuçlarını öğrencilere gözletir. Son olarak öğrencilerden de aynı görevi yerine getirmelerini ister.

1.5 Uygula- Kızılötesi Sayaç

Malzeme Listesi:

Arduino Uno
Breadboard
Kızılötesi sensör

Bu etkinlikte amaç kızılötesi sensör karşısından geçen nesnelerin sayısını bulup seri port ekranına yazmaktır. Bu görev için aşağıdaki resimde görülen kod kullanılabilir.



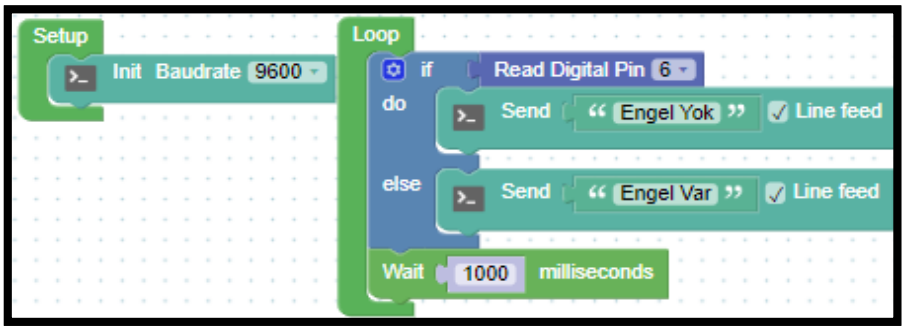
Resim 222. Örnek Program

1.6 Gözle- Kızılötesi Sensör ile Siyah-Beyaz Ayrımı

Malzeme Listesi:

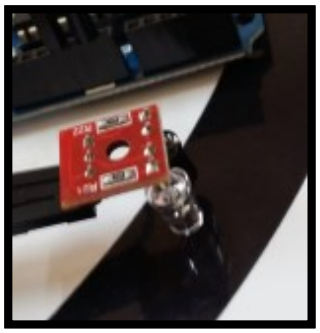
Arduino Uno
Breadboard
TCRT5000 kızılötesi sensör

Kızılötesi sensör kullanılarak siyah ile beyaz renk birbirinden ayırt edilebilir. Kızılötesi sensöre yakın bir şekilde duran (çok yakın olmalıdır aksi taktirde çalışmayabilir) siyah bir cisim olduğu var sayılsın. Siyah renk kendisine gönderilen ışıkları soğuracağı/tutacağı için kızılötesi vericinin gönderdiği ışık kızılötesi alıcıya ulaşmaz. Bu yüzden karşısında bir şey yokmuş gibi davranır. Rehber öğretmen aşağıdaki resimde görülen kodu Arduino karta yükler.

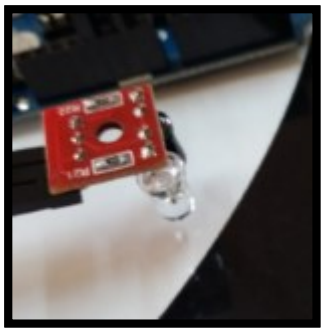


Resim 223. Örnek Program

Robotik programlamada kullanılan matlar üzerinde beyaz zeminde siyah çember bulunmaktadır. Rehber öğretmen kızıl ötesi sensörü yakın olacak şekilde çemberdeki siyah renk üzerinde tutar ve kodun çıktısını öğrencilere gösterir. Aynı işlemi beyaz zemin için de yapar. Aşağıdaki resimlerde siyah ve beyaz zeminler görülmektedir.



Resim 224. Siyah Zemin



Resim 225. Beyaz Zemin

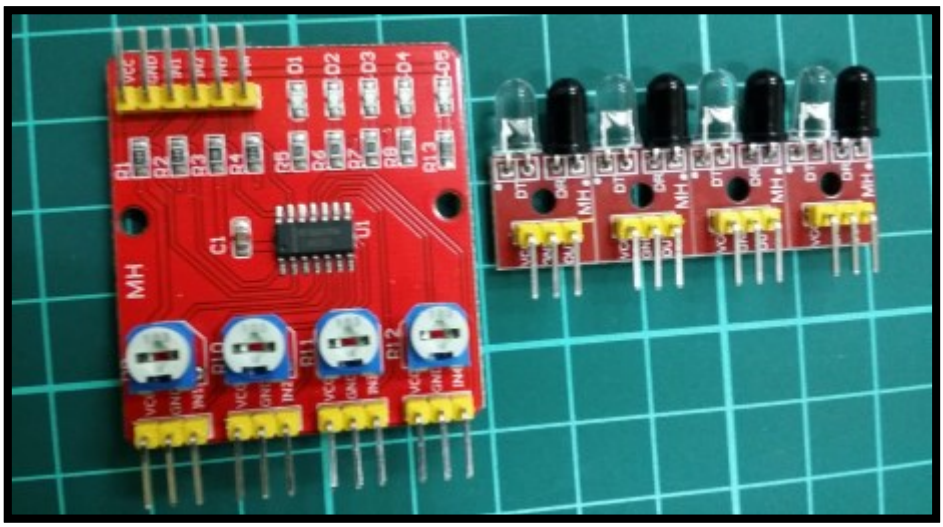
Yeteri kadar deneyimden sonra rehber öğretmen öğrencilere bunun sebebini sorar. Öğrencilere düşünmesi için vakit verir. Bu durum sınıfça tartışılır. Öğrenciler sebebi bulamazsa rehber öğretmen gerekli açıklamaları yapar.

1.7 Gözle ve Uygula- Kızılötesi Sensörleri Birlikte Kullanıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
3 kızılötesi sensör ve sensör devre kartı

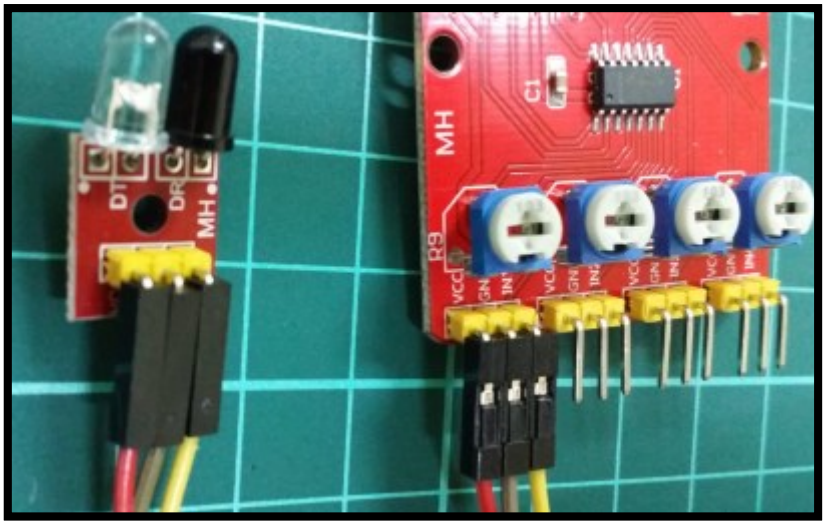
Bu etkinlikteki amaç üç sensörün bulunduğu yolda ilerleyen bir cismin yerinin hangi sensörün önünde olduğunu buldurmaktır. Bu görev için kızılötesi sensör kullanılacaktır. Robotun çizgiyi takip edebilmesi için üç kızılötesi sensöre ihtiyaç duyulur. Piyasada 4'lü kızılötesi sensörün ortak kullanımına izin veren modüller bulunmaktadır. Aşağıdaki resim bu modülü göstermektedir.



Resim 226. Kızılötesi Sensör Modülü

Yan yana bulunan infrared sensörler bir pense (el ile de yapılabilir) yardımıyla kolayca birbirinden ayrılabilir. Modül kartının sensörler ile bağlantısı kolaydır. Dişi-dişi jumper kablolar kullanılarak sensörün VCC bacağı modüldeki üçlülerden bir tanesinin VCC bacağına; sensörün GND bacağı kartın GND bacağına, sensörün OUT bacağı kartın kullanılan üçlüsünün IN bacağına (burada IN1 kullanılmıştır) bağlanır. Kart üzerinde bulunan mavi-beyaz elemanlar potansiyometredir. Bu

potansiyometreler kullanılarak uzaklık ayarı yapılabilir. Sensörün karta bağlantısı aşağıdaki resimde gösterilmiştir.

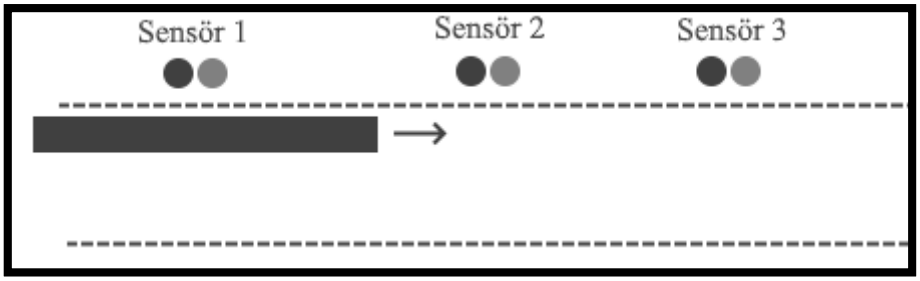


Resim 227. Sensör Kart Bağlantısı

Bu karta dörde kadar sensör bağlanabilir. Bu etkinlikte üç sensör kullanılacağı için IN2 ve IN3 girişi için de sensör bağlantıları yapılır.

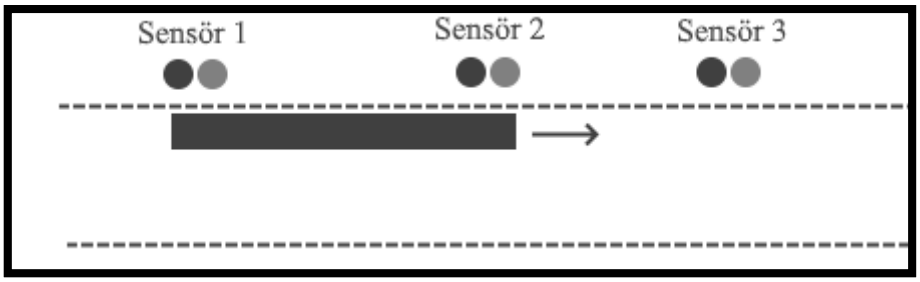
Kızılötesi sensör kartının üst kısmında bulunan bacaklar yardımıyla kartın Arduino bağlantısı kolayca yapılabilir. Sensör kartının VCC bacağı Arduino'nun 5V pinine, sensör kartının GND bacağı Arduino'nun GND pinine, sensör kartındaki kullanılan IN bacakları ise Arduino'nun dijital girdi pinlerine takılır. Üst tarafta bulunan IN bacakları aşağıdaki IN bacakları ile eşleştirilmiştir. Yani kartın altındaki IN1 bacağı kartın üstündeki IN1 bacağına karşılık gelir. Burada üç sensör kullanılacaktır. IN1 bacağı 13 numaralı dijital pine, IN2 bacağı 12 numaralı dijital pine ve IN3 bacağı 11 numaralı dijital pine bağlanacaktır. Bu şekilde üçlü sensöre arkadan bakıldığında sağdaki sensör 13 numaralı dijital pine, ortadaki sensör 12 numaralı dijital pine ve soldaki sensör 11 numaralı dijital pine bağlanmış olarak görülür. Bu bağlantılar böyle olmak zorunda değildir. Buradaki konular bu bağlantılara göre anlatılacağı için bağlantı detayları verilmiştir.

Yeniden tekrarlamak gerekirse bu etkinlikteki amaç üç sensörün bulunduğu yolda ilerleyen bir cismin yerinin hangi sensörün önünde olduğunu buldurmaktır. Bahsi geçen yol aşağıdaki resimde modellenmiştir.



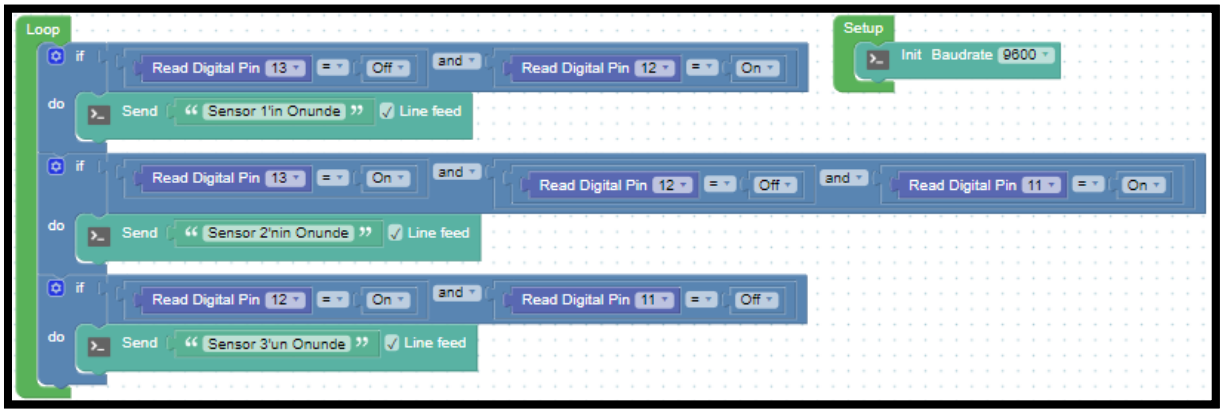
Resim 228. Yer Bildirimi

Sensörlerin önünden bir anda sadece bir araç geçebilir. Fakat araçların uzunluklarından ötürü araçlar kısa bir süreliğine de olsa her iki sensörün önünden geçebilmektedir. Bu durum aşağıdaki resimde gösterilmiştir. Fakat araçların yerleri sadece bir sensörün önünden geçerken bildirilmelidir. Her iki sensörün önünden geçtiğinde yer bildirimi yapılmayacaktır.



Resim 229. Yer Bildirimi Yok

Bu görev için aşağıdaki kod kullanılabilir. Bu kod Sensör 1'in 13 numaralı pine, Sensör 2'nin 12 numaralı pine ve Sensör 3'ün 11 numaralı pine bağlı olduğu durum için yazılmıştır. Rehber öğretmen anlatarak devreyi kurar ve aşağıdaki resimde görülen kodu öğrencilere açıklar. Son olarak aynı devre ve kodun öğrenciler tarafından da yapılmasını sağlar.



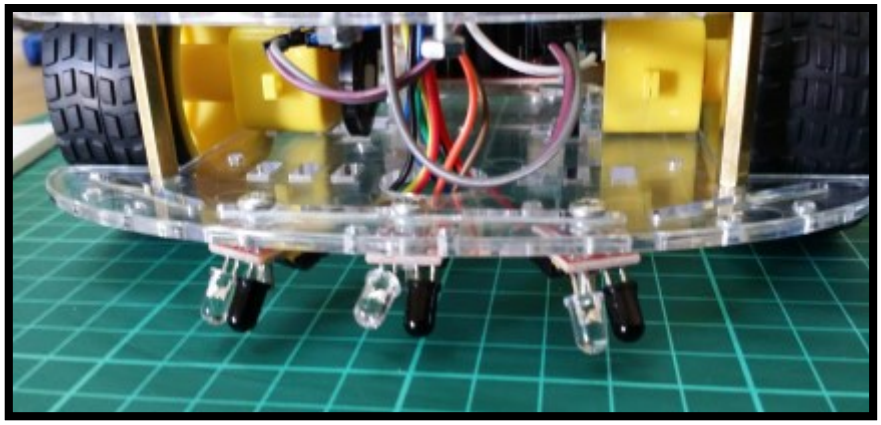
Resim 230. Örnek Program

1.8 Gözle ve Uygula- Eliptik Çizgi İzleyen Robot

Malzeme Listesi:

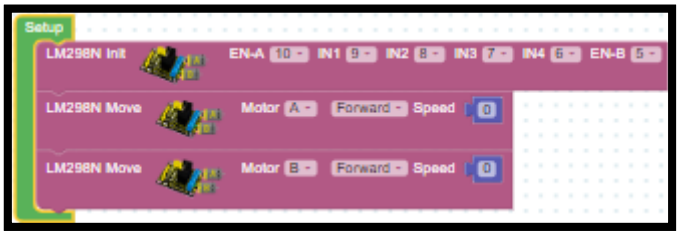
Arduino Uno
L298N sürücü modülü
4WD çok amaçlı mobil araba platformu
Şarj Edilebilir 9 V 200 mAh Pil
3 adet kızılötesi sensör ve sensör kartı

Bu etkinlikte amaç eliptik bir çizgiyi izleyen bir robot oluşturmak ve kodunu yazmaktır. Aşağıdaki resimde gösterildiği gibi robotun ön alt kısmına üç adet kızılötesi sensör yerleştirilir. Sensörlerin robota bağlantısı aşağıdaki resimde gösterilmiştir. Kızılötesi alıcı ve vericilerin baktığı yön önemlidir. Kızılötesi alıcı ve vericiler birbirlerine çok yakın ve/veya dik olarak aşağıya bakar şekilde olmamalıdır. Birbirlerine çok yakın iseler araları biraz açılmalıdır. Mat ile dik açı yapıyorlarsa karşıdan bakıldığında bakan kişiye doğru olacak şekilde küçük bir açı yapmalıdır.



Resim 231. Robot Üzerindeki Sensörler

Sensör kartı en üst kata Arduino kartın yanında iki taraflı bant yardımıyla yerleştirilir ve bağlantılar yapılır. Bağlantılar 13, 12 ve 11 numaralı pinlere yapılmalıdır. Bu görev için aşağıdaki kod kullanılabilir. Rehber öğretmen devre bağlantılarını öğrencilere anlatarak yapar. Aşağıdaki resimlerde görülen kodu öğrencilere açıklar. Son olarak öğrencilerin devreyi kurup kodu yazmalarını sağlar.



Resim 232. Örnek Setup Kodları

Resim 233. Örnek Program

Dikkat

Bu etkinlikte sensörlerin arasındaki mesafe ve baktıkları yön önemlidir. Rehber öğretmen kod çalışmazsa sensörler arasındaki mesafe ve baktıkları yönü değiştirerek kodun çalışmasın sağlamalıdır.

1.9 Gözle- LCD ile Ekrana Yazı Yazma

Malzeme Listesi:

Arduino Uno
Breadboard
2x16 LCD ekran- I2C dönüştürücü kartı
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Şimdiye kadar ekran tabanlı çıktı birimi olarak seri port ekranı kullanılmıştır. LCD'ler aracılığıyla da dışarıya veri gönderilebilir. Piyasada farklı LCD'ler bulunur. Aşağıdaki resimde 2x16 (alttaki) ve 4x20 (üstteki) LCD görülmektedir. 2x16 LCD'lerde 2 satır ve 16 sütun bulunur. 4x20 LCD'lerde 4 satır ve 20 sütun bulunur. Bir satır ve bir sütuna karşılık gelen birime hücre denilebilir. Her bir hücreye bir karakter yazılabilir.

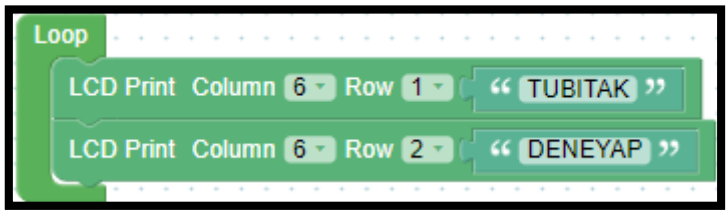
Resim 234. LCD Ekranlar

LCD'ler Arduino kartına doğrudan bağlanabileceği gibi I2C modülü aracılığıyla I2C bağlantısı üzerinden de bağlanabilir. Bu derste kullanılan LCD'lerde I2C modülü LCD'nin arkasına lehimlenmiş olarak gelmektedir. I2C üzerinden LCD bağlantısı yapmak oldukça kolaydır. I2C modülü üzerinde bulunan SCL bacağı Arduino kartının A5 pinine, SDA bacağı A4 pinine, VCC bacağı 5V pinine ve GND bacağı GND pinine bağlanır. SCL ve SDA bacakları A5 ve A4 pinleri dışındaki analog pinlere bağlanırsa LCD çalışmaz. A5 ve A4 pinleri Arduino kartının I2C bağlantı noktalarıdır (sonraki derslerde gösterileceği gibi başka pinler de mevcuttur). Bu yüzden I2C modülündeki SCL ve SDA bacakları bu pinlere bağlanmak *zorundadır*. 2x16 ve 4x20 LCD'lerin bağlantısı farklılık göstermez. İki çeşit LCD de aynı şekilde bağlanır.

ArduinoBlocks ortamında LCD ile ilgili komutlar "*LCD Screen*" blok başlığı altında bulunur. LCD'nin kullanılabilmesi için öncelikle "Setup" bloğu içerisinde "*LCD Begin (I2C)*" bloğu başlatılmalıdır. Aşağıdaki resimde LCD setup ayarları görülmektedir.

Resim 235. LCD Setup Ayarı

LCD'de yazılması istenen metin "LCD Print" bloğu kullanılarak yazılır. "LCD Print" bloğunda metnin nereden başlayacağı belirtilmelidir. Aşağıdaki resimde görülen komutta TUBITAK kelimesi 6. sütun (Column) ve 1. satırdan (Row) başlanarak yazılır. Aşağıdaki komut 4x20 LCD ekranına alt alta gelecek şekilde TUBITAK ve DENEYAP yazar.

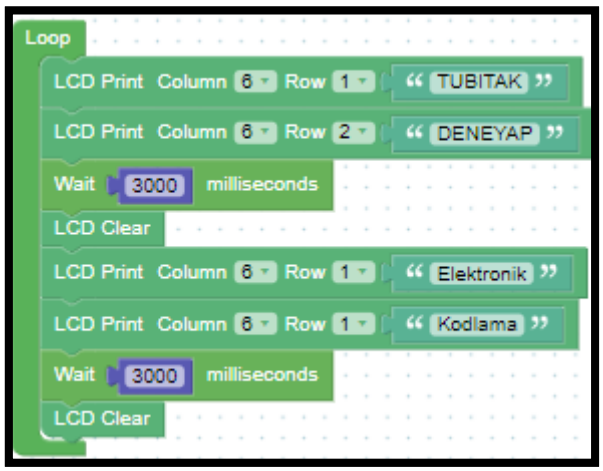


Resim 236. Örnek Yazı Yazma Programı

Dikkat

Bu komutu verildiğinde ekran ışığı yanıyor fakat yazı ekranda belirmiyorsa, I2C modülü üzerinde bulunan potansiyometre ile ayar yapılmalıdır.

LCD'de ekranı silip yeni bir metin yazmak için "LCD Clear" bloğu kullanılır. Aşağıdaki resimde ekranda üç saniye boyunca TUBITAK-DENEYAP yazısı görülecek ve ardından üç saniye boyunca Elektronik-Kodlama yazısı görülecektir (yazılar alt alta görülür, burada böyle yazılmıştır).



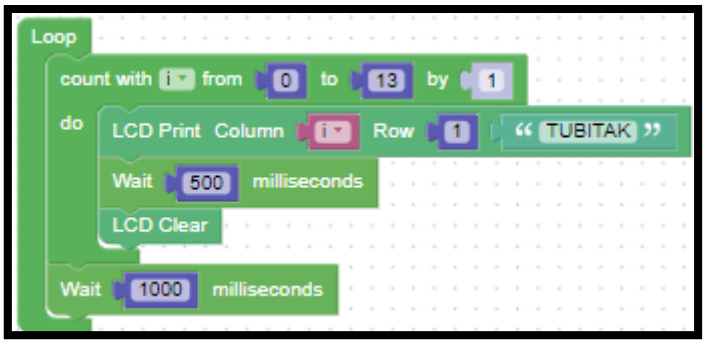
Resim 237. Örnek Yazı Animasyonu Programı

1.9 Uygula- LCD'de Yazı Animasyonu

Malzeme Listesi:

Arduino Uno
Breadboard
2x16 LCD ekran- I2C dönüştürücü kartı
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bu etkinlikte amaç ekranın en solundan başlayarak en sağına doğru birer birer ilerleyerek giden yazı animasyonu yapmaktır. TUBITAK yazısı ilk önce ekranın en solunda belirir ve arkasından akarak ekranın en sonuna kadar ilerler. TUBITAK yazısı ekrandan çıkmaz. Bu işlem sürekli tekrarlanır. Bu iş için aşağıdaki resimde görülen kod kullanılabilir.



Resim 238. Örnek Program

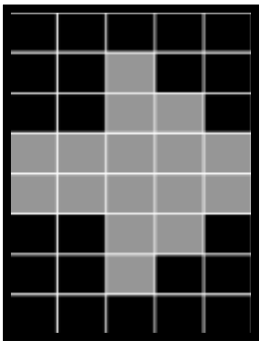
1.10 Gözle- LCD'de Sembol Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

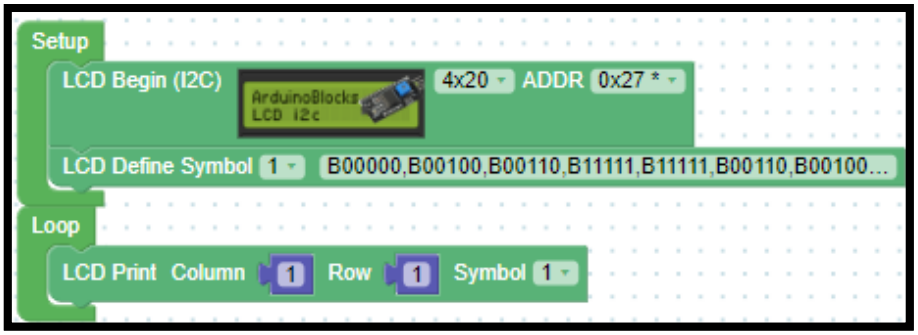
ArduioBlocks ortamında LCD'de kullanılmak üzere çeşitli semboller tanımlanabilir. Bu iş için şu bağlantı kullanılabilir: http://www.arduinoblocks.com/web/help/chareditor

Bağlantı tıklandığında açılan siteden sembol tasarlanabilir. Siyah hücrelerin üzerine tıklandığında tıklanan hücre yeşil renkli olur. Bu şekilde istenen sembol tasarlanabilir. Aşağıdaki resimde bir ok işareti tasarlanmıştır.



Resim 239. Ok Sembolü

Sembolün tasarımı bittiğinde altında oluşan kod kopyalanarak "LCD Define Symbol" bloğuna yapıştırılabilir. Böylece bu sembol LCD'de kullanılabilir. Aşağıdaki resimde görülen kod bloğunda LCD ekranına oluşturulmuş ok işareti yazdırılmaktadır.



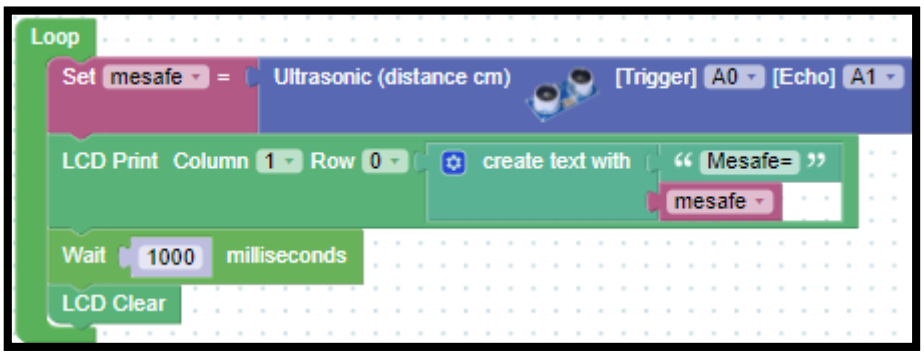
Resim 240. Örnek Program

1.10 Uygula- LCD'yi Arabaya Bağlıyorum

Malzeme Listesi:

Arduino Uno
L298N sürücü modülü
4WD çok amaçlı mobil robot platformu
Şarj Edilebilir 9 V 200 mAh Pil
3 adet TCRT5000 kızılötesi sensör
4x20 Mavi LCD ekran- I2C dönüştürücü kartı
HC-SR04 mesafe sensörü
HC-SR04 mesafe sensörü

Bu etkinlikte amaç araba üzerine monte edilmiş LCD ekranda mesafe sensöründen gelen uzaklık değerinin gösterilmesidir. LCD'nin arabaya monte edilişi EK-3'te (Araba Kurulum Kılavuzu) gösterilmiştir. Etkinlik için mesafe sensörünün "*Trig*" bacağı Arduino'nun "A0" pinine, *Echo* bacağı ise A1 pinine bağlanmıştır. Bu görev için aşağıdaki resimde görülen kod kullanılabilir.



Resim 241. Örnek Program

2. TASARLA VE ÜRET

2.1 Tasarla- Sembollerle Animasyon Yapıyorum

Bu etkinlikte amaç ekranın sağ tarafında bulunan bombaya benzer bir sembolün ekranın sol tarafından akarak gelen bir ok sembolü tarafından patlatılmasını gösteren bir animasyon yapmaktır.

Sembollerle animasyon yapıyorum etkinliğini tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir. Aşağıdaki örnek süreç kesinlikle öğrencilere gösterilmemelidir. Rehber öğretmen sınıfta genel bir sorun görürse sınıfa açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Bu

etkinlikte 1.10 Gözle (LCD'de Sembol Kullanımı) başlığında kullanılan elektronik devre kullanılacaktır.

Tanımlama: Öğrencilerin öncelikle sembollerle animasyon yapıyorum etkinliğinin neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrenciler öncelikle gerekli işlemleri detaylı olarak maddeler hâlinde yazmalıdır.

- Başlangıç ekranında ekranın sağ üst köşesinde bomba sembolü vardır.
- Ekranın sol üst köşesinden bomba sembolüne doğru bir ok sembolü her 0,2 saniyede hareket eder.
- Ok sembolü bomba sembolüne değdiğinde patlama efekti sağ üst köşeden diğer üç köşeye doğru akar.

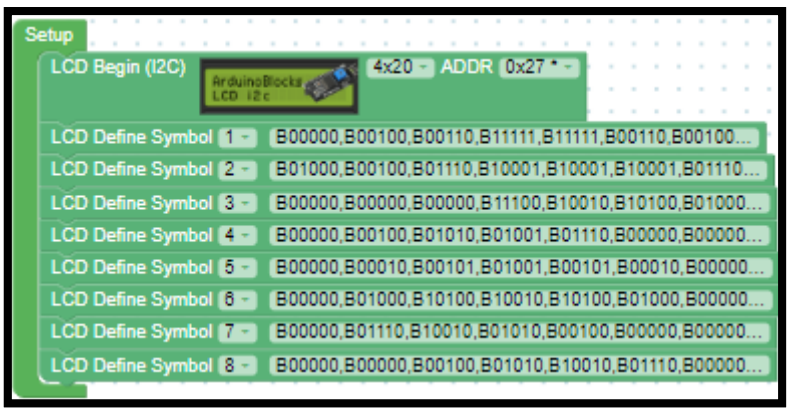
Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Tanımlamada belirlenen her bir adımın gerçekleştirilebilmesi için gerekli fikirler belirlenmelidir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

- http://www.arduinoblocks.com/web/help/chareditor adresinden ok, bomba ve patlama sembolleri oluşturulur.
- Oluşan kodlar kopyalanarak "LCD Define Symbol" bloğuna yapıştırılabilir.
- Bomba simgesi sağ üst köşeye yerleştirilir.
- Ok simgesi sol üst köşeden başlayarak bomba simgesine doğru her 0,2 saniyede hareket eder.
- Ok simgesi bomba simgesine değdiğinde patlama ekranda patlama sembolü/sembolleri görünür.

2.1 Üret- Sembollerle Animasyon Yapıyorum

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Bu etkinlik için aşağıdaki resimde görülen örnek kod kullanılabilir. Bu sorunun birden fazla çözümü vardır. Öğrenciler kendi animasyonlarını üretebilirler.

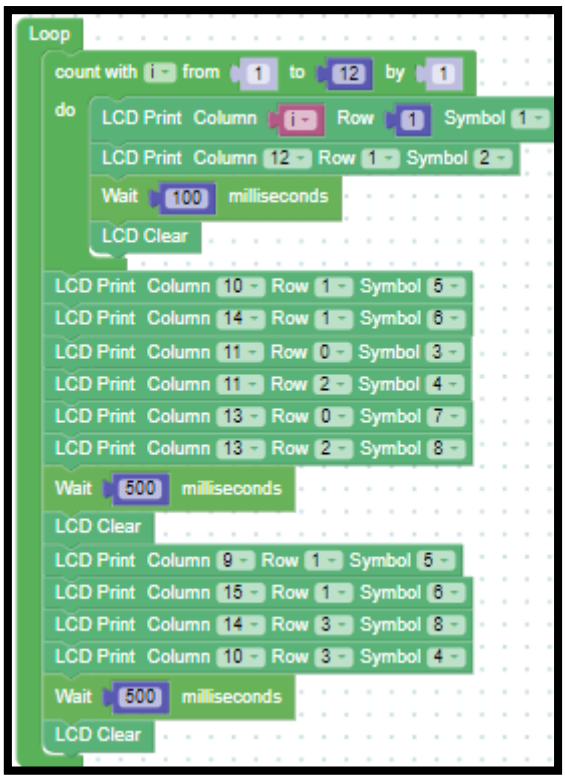


Resim 242. Örnek Program

Sembollerin tanımlandıkları kodlar sırasıyla aşağıdaki şekildedir:

```
1  B00000,B00100,B00110,B11111,B11111,B00110,B00100,B00000
2  B01000,B00100,B01110,B10001,B10001,B10001,B01110,B00000
3  B00000,B00000,B00000,B11100,B10010,B10100,B01000,B00000
4  B00000,B00100,B01010,B01001,B01110,B00000,B00000,B00000
5  B00000,B00010,B00101,B01001,B00101,B00010,B00000,B00000
6  B00000,B01000,B10100,B10010,B10100,B01000,B00000,B00000
7  B00000,B01110,B10010,B01010,B00100,B00000,B00000,B00000
8  B00000,B00000,B00100,B01010,B10010,B01110,B00000,B00000
```

Aşağıdaki kod ile tanımlanan sembollerden bir animasyon oluşturulmuştur.



Resim 243. Örnek Program

3- DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Mesafe sensörünün günlük yaşamda kullanılabileceği yerlere örnek veriniz.

2. Sizce aşağıdaki resimde görülen kod çalışır mı? Çalışırsa nasıl çalışır? Çalışmazsa neden çalışmaz? Sebepleriyle açıklayınız?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaşılan problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Arabadaki düz gitmeme sorununu nasıl çözdünüz?
    - Çizgi izleyen arabadaki kodum çalışmadı. Sebebi ne olabilir?
    - Arabaya ileri git komutu verdiğim halde olduğu yerde dönüyor. Sebebi ne olabilir?
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Otopark Sayaç

Üç katlı bir otopark bulunmaktadır. Otoparkın girişi birinci kattadır ve çıkışı üçüncü kattadır. Birinci, ikinci ve üçüncü kat girişlerinde birer adet kızılötesi sensör (sırasıyla K1, K2 ve K3) bulunmaktadır. Bu sensörler şu ana kadar otoparka girmiş ve her bir katta park etmiş toplam araç sayısını saymak için kullanılacaktır. Otoparkta ve katlarında anlık olarak bulunan araç sayısı önemli değildir. Şu ana kadarki sayılar istenmektedir.

Birinci kata gitmek isteyen bir sürücü K1 sensörünün önünden geçmek zorundadır. İkinci kata gitmek isteyen bir sürücü K1 ve K2 sensörlerinin önünden geçmek zorundadır. Üçüncü kata gitmek isteyen bir sürücü ise K1, K2 ve K3 sensörlerinin önünden geçmek zorundadır. Otoparkta alt katlardan üst katlara doğru gitmek mümkündür fakat üst katlardan alt katlara inilmesi mümkün değildir. Birinci kata park eden bir sürücü otoparktan çıkmak istediğinde K1, K2 ve K3'ün önünden geçmeden doğrudan çıkışa gitmektedir. İkinci kata park eden bir sürücü otoparktan çıkmak istediğinde K2 ve K3'ün önünden geçmeden doğrudan çıkışa gitmektedir. Üçüncü kata park eden bir sürücü otoparktan çıkmak istediğinde K3'ün önünden geçmeden doğrudan çıkışa gitmektedir. Bu koşullar göz önüne alındığında otoparka

giren ve her bir kata şu ana kadar park edilmiş toplam araç sayısını anlık olarak seri port ekranına yazdıran programı oluşturunuz.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Dikkat

Öğrenciler sorunun çözümü için basit otopark maketleri yapabilecekleri gibi maket benzeri bir şey oluşturmadan sensörleri aralıklı bir şekilde yan yana koyup otoparkı modelleyebilirler. Fakat öğrencilerin karmaşık otopark maketi yapmaları istenmemektedir. Bu etkinlikteki amaç elektronik malzemeleri bir programlama dili vasıtasıyla kodlamaktır. Karmaşık maketler yapmak öğrencilerin bu etkinlikteki asıl amaca daha az odaklanmasına sebep olabilir.

Öğrenciler çözüme yönelik elektronik devre ve program tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

4.2 Dairenin İçerisinden Çıkmayan Robot

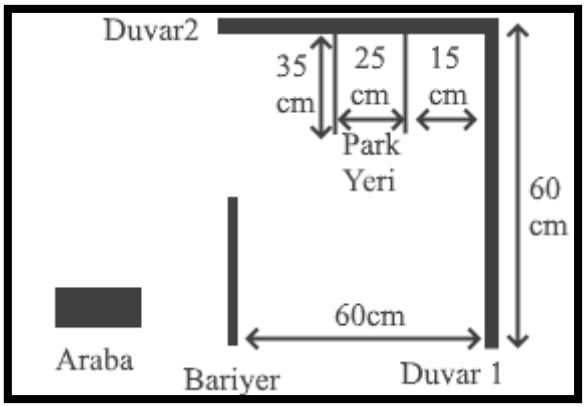
Bu etkinlikte amaç daire şeklinde beyaz zemin üzerine çizilmiş siyah şeridin içerisinden çıkmayan robot yapmaktır. Bu görevde Robotik Programlama dersinde kullanılan mat kullanılabilir. Robot sürekli dairenin içerisinde hareket etmelidir fakat dışarıya çıkmamalıdır.

Bu etkinlikte 1.8 Gözle ve Uygula başlığında hazırlanan araba kullanılacaktır. Programı yazmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler sonra bilgisayar, mat ve araba başında çalışarak istenilen görevi yerine getiren programı oluştururlar. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

4.3 Arabayı Park Ediyorum

Aşağıda bir park yerinin taslağı verilmiştir. Öğrenciler bu taslağın duvarlarını ve bariyeri mukavvalar yardımıyla oluşturmalıdır. Park yerinin girişinde bir bariyer bulunmaktadır. Bariyerden 60 cm sonra "Duvar" 1 bulunmaktadır. "Duvar" 1'in genişliği en az 60 cm olmalıdır. Araba bariyere yaklaştığında bariyer açılmalıdır. Sonra araba içeri girip belirlenen yere park etmelidir. Aşağıdaki resimde park yeri görülmektedir.



Resim 244. Park Yeri

Bu etkinlikte 1.8 Gözle ve Uygula başlığında hazırlanan araba kullanılacaktır. Programı yazmaya başlamadan önce grupların tasarla adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler sonra bilgisayar, park yeri ve araba başında çalışarak istenilen görevi yerine getiren programı oluştururlar. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

4.4 Üçüncü Çizgiyi Geçince Duran Robot

Bu etkinlikte amaç robotun üçüncü çizgiyi geçtikten sonra durmasını sağlamaktır. Etkinlik için Robotik Kodlama dersinde kullanılan mat üzerindeki üçlü çizgiden faydalanılabilir. Robot ilk iki çizgiyi geçecektir ve üçüncü çizginin hemen ardından durmalıdır.

Bu etkinlikte 1.8 Gözle ve Uygula başlığında hazırlanan araba kullanılacaktır. Programı yazmaya başlamadan önce grupların tasarlama adımı için yukarda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Öğrenciler bilgisayar, mat ve araba başında çalışarak istenilen görevi yerine getiren programı oluştururlar. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir.

7. Bölüm-Akıllı Ev Tasarlıyorum

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol ve kondansatör kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.

Bölümün Kazanımları:

- DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanılan elektronik devreleri oluşturabilir.
- DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanılan elektronik devreleri programlayabilir.
- Sensörlerden alınan verileri LCD'de görebilmek için gerekli elektronik devreleri oluşturabilir.
- Sensörlerden alınan verileri LCD'de görebilmek için gerekli programları oluşturabilir.

Bölümün Amacı:

Bu bölümün amacı DTH11 sıcaklık ve nem sensörünü, RTC DS3231 saat modülünü, PIR sensörünü, kapı alarm sensörünü, ses sensörünü, alev sensörünü, yanıcı gaz sensörünü, toprak nem sensörünü, yağmur sensörünü, tilt sensörünü ve tuş takımını elektronik devrelerde kullanabilmelerini ve bu elektronik devreleri ArduinoBlocks ortamında programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, 4x16 LCD ekran- I2C dönüştürücü kartı, 4x20 Mavi LCD ekran- I2C dönüştürücü kartı, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü,

tilt sensörü ve tuş takımı, buzzer, servo motor, led, (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: DTH11 sıcaklık ve nem sensörünün, RTC DS3231 saat modülünün, PIR sensörünün, kapı alarm sensörünün, ses sensörünün, alev sensörünün, yanıcı gaz sensörünün, toprak nem sensörünün, yağmur sensörünün, tilt sensörünün ve tuş takımının tanıtılması, elektronik devrelerde kullanılması ve bu elektronik devrelerin ArduinoBlocks ortamında programlanması.

Uygula: DTH11 sıcaklık ve nem sensörünün, RTC DS3231 saat modülünün, PIR sensörünün, kapı alarm sensörünün, ses sensörünün, alev sensörünün, yanıcı gaz sensörünün, toprak nem sensörünün, yağmur sensörünün, tilt sensörünün ve tuş takımının elektronik devrelerde kullanılması ve bu elektronik devrelerin ArduinoBlocks ortamında programlanması.

Tasarla: Toprak nem sensörü, yağmur sensörü ve tilt sensörüyle birlikte üç tane daha sensör kullanarak akıllı bir ev tasarımı için gerekli elektronik devrelerin oluşturulması ve programının yapılması için tanımlama ve fikir yürütme adımlarının gerçekleştirilmesi.

Üret: Toprak nem sensörü, yağmur sensörü ve tilt sensörüyle birlikte üç tane daha sensör kullanarak tasarlanan akıllı evin elektronik devrelerin ve programının oluşturulması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE VE UYGULA

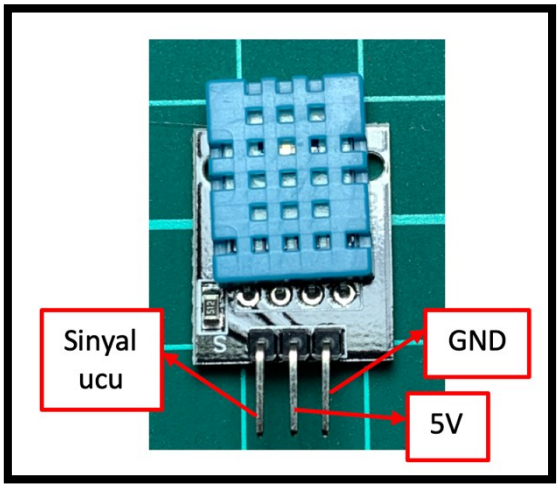
1.1 Gözle ve Uygula- Nem Sensörü ve Dijital Saat

Malzeme Listesi:

Arduino Uno
Breadboard
RTC DS3231 saat modülü
DTH11 sıcaklık ve nem sensörü

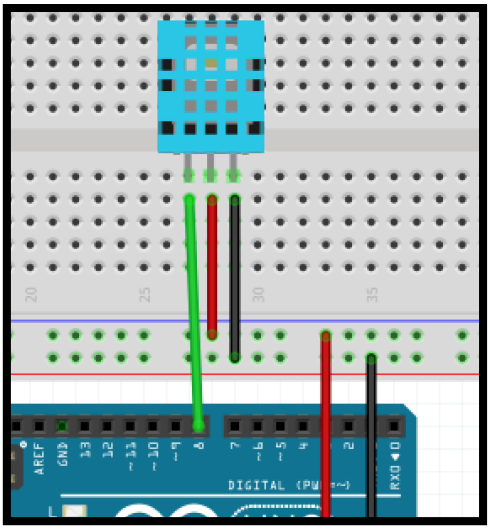
Bu etkinlikte amaç RTC DS3231 saat modülünü ve DTH11 sıcaklık ve nem sensörünü kullanarak seri port ekranında güncel sıcaklık, nem, saat ve tarih bilgilerini gösteren dijital bir saat yapmaktır. Öncelikli olarak sıcaklık ve nem sensörü tanıtılacak ve ilgili devreler tanıtım sonrası kurulacaktır.

Hava nemi havanın taşıdığı su buharı miktarı ile ilgilidir. Havanın taşıdığı nemi bilmek birçok alanda önemlidir. Örneğin yaz aylarında sıcaklık artışıyla birlikte havadaki nem miktarının düşmesi orman yangınlarına neden olabilir. Bazı tarım ürünlerinin verimli şekilde yetiştirilmesi havadaki nem oranına bağlıdır. Zaman zaman yüksek nem zararlı mikroorganizmaların çoğalmasına neden olabilir. Örneklerden de anlaşılacağı gibi havadaki nemi ölçmek birçok alanda önemlidir. Arduino ile havadaki nem kolaylıkla ölçülebilir. Bunun için DTH11 isimli hava nemini ölçen sensör kullanılacaktır. Bu sensör aşağıdaki resimde görülmektedir.



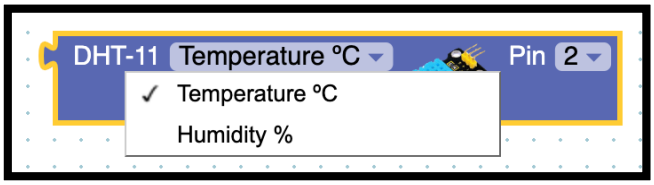
Resim 245. DTH11 Nem ve Sıcaklık Sensörü

DTH11 sensörü havadaki nemi %5 hata payı ile ölçmektedir. Ayrıca havadaki nemi ölçmenin yanında 0 ile 50 C° arasındaki hava sıcaklığını da %2'lik bir yanılma payı ile ölçebilmektedir. Bu da özellikle kapalı alanlardaki nem ve sıcaklık ile ilgili ölçümlerin yapılabilmesi için idealdir. DTH11 sensörü görselde görüldüğü gibi üç adet bağlantı pinine sahiptir. Bu pinlerden sağ ve ortada yer alanlar DTH11'in güç bağlantıları için kullanılmaktadır. Sağdaki uç – yani GND, ortadaki uç ise 5V bağlantısı içindir. Solda yer alan uç ise Arduino'ya sinyal göndermek için kullanılır. Bu sinyal ucu herhangi bir dijital pine bağlanabilir. Etkinliğin diğer adımlarına geçmeden önce DTH11 nem ve sıcaklık sensörünün çalışıp çalışmadığı kontrol edilmelidir. Bunun için aşağıdaki resimde görülen devre kurulmalıdır. Devrede DTH11 sensörünün güç bağlantıları yapıldıktan sonra solda yer alan sinyal ucu Arduino UNO'nun dijital pinlerinden herhangi birine takılabilir. Bu pin numarası kod yazarken kullanılacaktır.



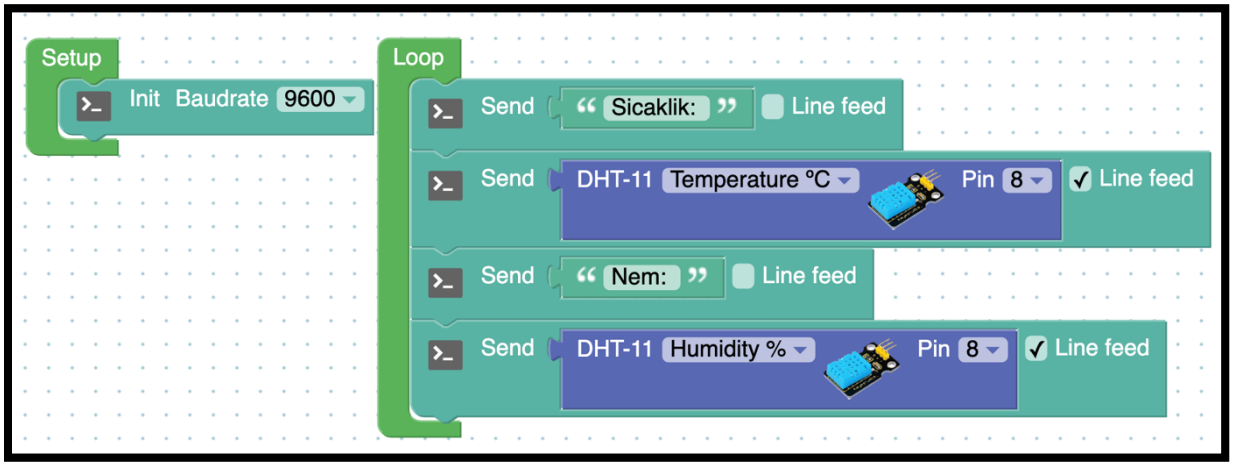
Resim 246. Örnek Devre Şeması

Devre kurulduktan sonra "*Sensors"* blok başlığının altında bulunan DTH11 bloğu kullanılarak sensörün çalışıp çalışmadığı kontrol edilmelidir. DTH11 bloğunun üzerinde sıcaklık ve nem bilgilerinden hangisinin okunacağının belirlendiği açılır bir menü bulunur. Bu menüden sıcaklık bilgisi ile ilgili veriler "*Temperature*" seçeneği ile alınır ve santigrat derece ölçü biriminde çıktı alınır. Aynı şekilde "*Humidity*" seçeneğiyle havadaki nemin yüzde bilgisi okunur. Bloğun sağ tarafındaki pin seçeneğinde sinyal ucunun bağlı olduğu pin numarası belirtilir. Aşağıdaki resimde sıcaklı ve nem okuma bloğu görülmektedir.



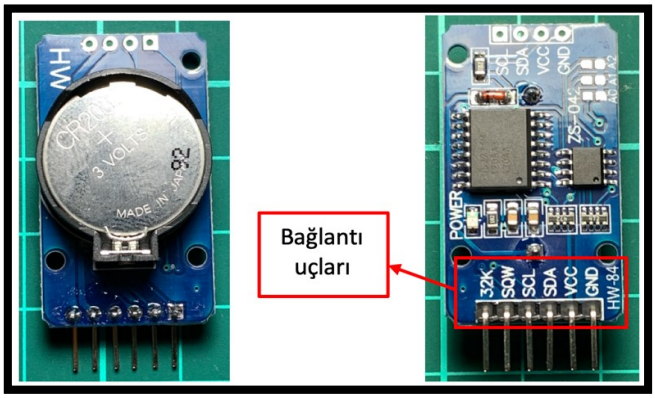
Resim 247. Sıcaklık ve Nem Okuma Bloğu

Sıcaklık ve nem bilgisini ölçmek ve ekrana yazdırmak için aşağıdaki resimde görülen kod kullanılır. Rehber öğretmen sıcaklık ve nem bilgilerinin okunduğundan ve her bir öğrenci tarafından ilgili kodun çalıştırıldığından emin olur.



Resim 248. Örnek Program

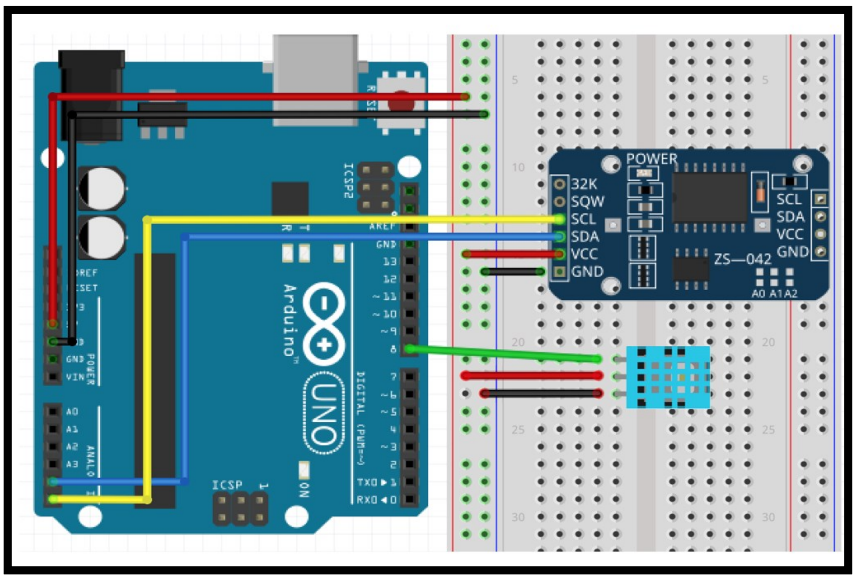
Etkinliğin tamamlanması için saat ve tarih bilgilerine ihtiyaç vardır. Bu görevi yerine getiren saat modülleri bulunur. Elektronik Programlama ve Nesnelerin İnterneti dersi kapsamında kullanılacak olan saat modülü aşağıda görseli verilen "RTC DS3231" saat modülüdür. Arduino ile kullanılabilen diğer saat modüllerinden farkı hata payının düşük olmasıdır. Özellikle sıcaklık değişimlerinden kaynaklı sapmalar bu modülde diğerlerine göre daha az görülür. Ayrıca içerisinde bir de EPROM (easily programmable read only memory/kolay programlanabilir sadece okunabilir bellek) bulunur. EPROM içerisine yazılan bilgileri hafızasında tutar. Enerji kesilse bile bu bilgileri kaybetmez. Aşağıdaki resimde RTC DS3231 saat modülü görülmektedir.



Resim 249. RTC DS3231 Saat Modülü

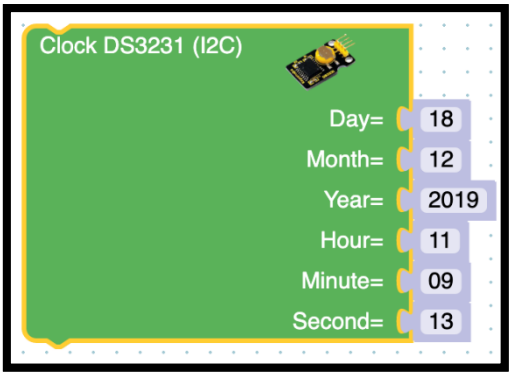
"RTC DS3231" saat modülü yıl, ay, hafta, gün, saat, dakika ve saniyeyi sayabilir. 28, 29, 30 ve 31 günden oluşan ayları ayırt edebilir. CR2032 kodlu saat pili ile başlangıç değeri ayarlanmış saat ve tarihi güncel olarak tutabilir. Kart üzerindeki VCC etiketli uç 5V bağlantısı ve GND etiketli uç ise GND bağlantısı içindir. SDA ve SCL uçları ise Arduino ile saat modülü arasındaki veri transferi içindir. Arduino UNO'da bu uçlar A4 ve A5 analog girişlerine karşılık gelir. Kalan iki uç, kart içerisindeki EPROM'un programlanması için kullanılır. Bu etkinliklerde EPROM programlanması yapılmayacaktır.

"RTC DS3231" saat modülü daha önce kurulmuş olan devreye aşağıdaki resimde görüldüğü şekliyle entegre edilir. GND ve VCC uçları sırası ile Breadboard üzerindeki 5V ve GND hatlarına bağlanır. SDA ve SCL uçları ise sırayla A4 ve A5 uçlarına takılır. Arduino ile saat modülü arasındaki iletişim bu uçlar aracılığı ile gerçekleştirilecektir.



Resim 250. Örnek Devre Şeması

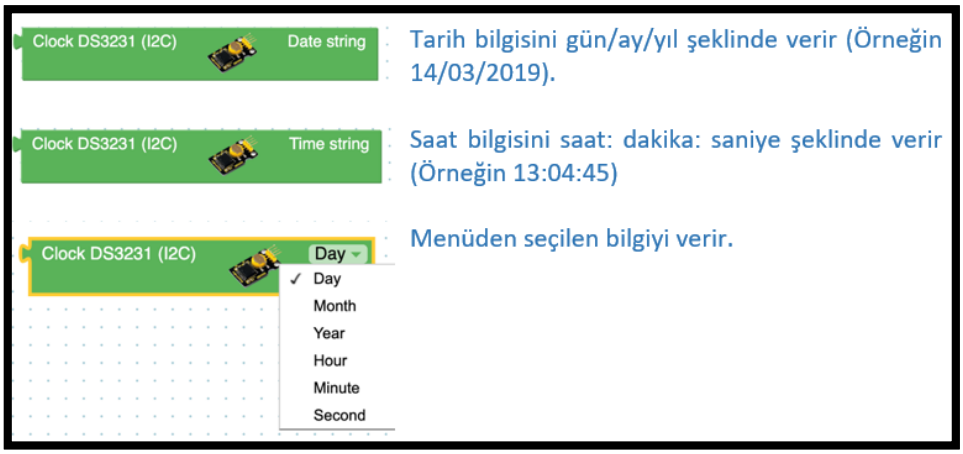
Devre kurulduktan sonra RTC DS3231 saat modülünün ilk ayarlarının yapılması ve çalışıp çalışmadığının kontrol edilmesi gerekmektedir. ArduinoBlocks'un "Clock RTC" blok başlığı içerisinde ilk olarak modülün ayarlanması için aşağıdaki görselde görülen blok kullanılır. Bu blok sayesinde saat modülünün ayarlaması yapılır. Saat modülü devreden ayrılarak güç bağlantısı kesilse bile üzerinde yer alan pilin enerjisi tükenmediği ya da pil yerinden çıkarılmadığı sürece güncel saat ve tarih bilgisi tutulacaktır. Aşağıdaki resimde clock RTC bloğu görülmektedir.



Resim 251. Clock RTC Bloğu

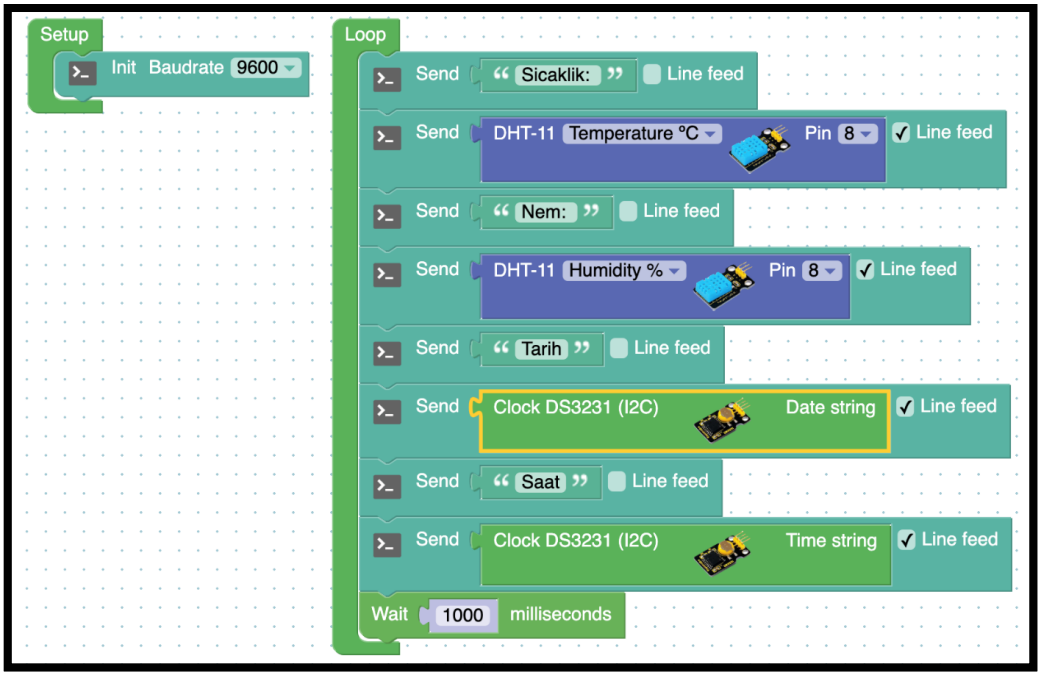
Görüldüğü gibi üstte yer alan ilk kutucuktan itibaren sırasıyla gün, ay, yıl, saat dakika ve saniye bilgilerinin girişi bu komut ile gerçekleştirilir.

Saat modülünün başlangıç ayarları yapıldıktan sonra istenirse tarih bilgilerinin (Yıl, gün, saat) ya da saat bilgilerinin (saat, dakika, saniye) tamamı toplu olarak okunabilirken, tüm bu bilgiler tek tek olacak şekilde de okunabilir. Bu bloklar aşağıdaki resimde gösterilmiştir.



Resim 252. Saat Modülü Blokları

Saat modülünün ilk tarih ayarlanması ve bu bilgilerin nasıl okunacağı rehber öğretmen tarafından aktarıldıktan sonra ana etkinlik devresinin kurulmasına devam edilir. Bu iş için aşağıdaki resimde görülen kod kullanılabilir. Kod sayesinde nem ve sıcaklık bilgilerinin yanında tarih ve saat bilgileri de okunarak ekrana yazılacaktır.



Resim 253. Örnek Program

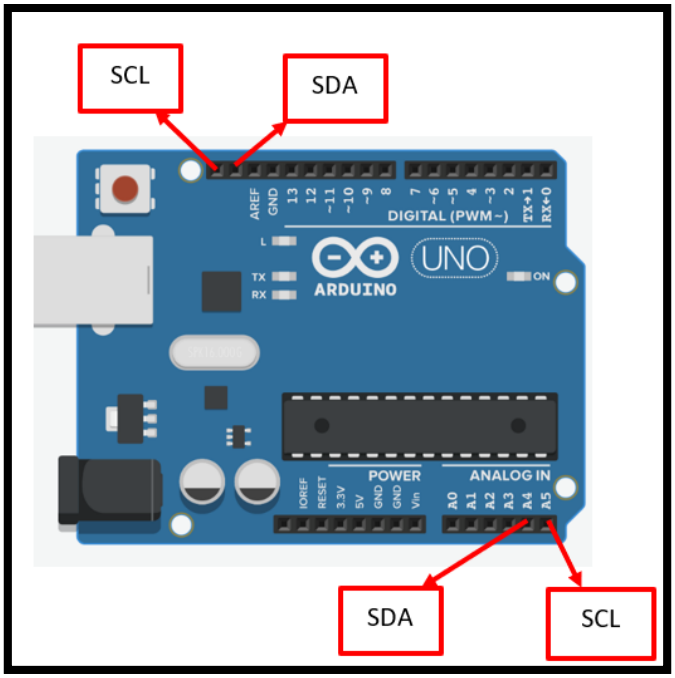
1.2 Gözle ve Uygula- Zaman, Sıcaklık ve Nem Bilgisini LCD'ye Yazıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
RTC DS3231 saat modülü
DTH11 sıcaklık ve nem sensörü
4x20 I2C LCD

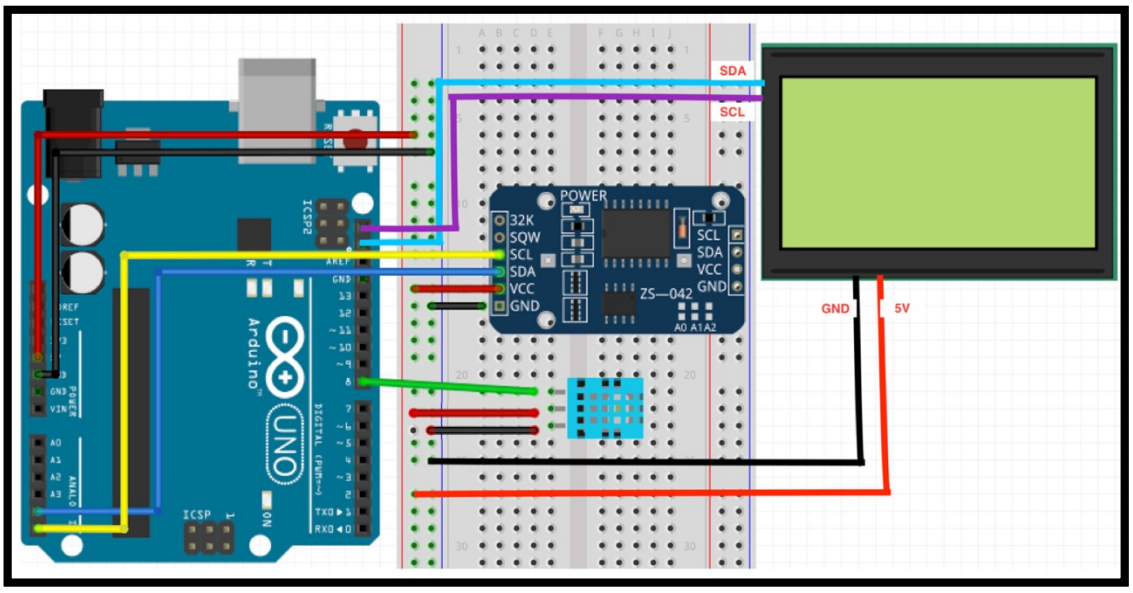
Bir önceki etkinlikte saat ve nem modüllerinden gelen bilgiler seri port ekranına yazdırılmıştır. Bu etkinlikte amaç bu bilgileri LCD'ye yazdırmaktır. Rehber öğretmen devre ile okunan bilgilerin LCD'ye yazılması için gerekli olan bağlantıların kurulumunu ve kodların yazılımını öğrencilere bırakmalıdır. Gerekli durumlarda aşağıdaki örnek devre ve kodlar kullanılarak öğrencilere yardım edilebilir.

LCD ekran ve saat modülü ortak I2C bağlantısı kullanır. Daha önceki derslerde I2C bağlantısının A4 ve A5 analog girişleri ile sağlandığından bahsedilmiştir. Ancak bu bağlantı uçlarına sadece bir adet I2C modülü bağlanabildiğinden ikinci bir I2C bağlantısına ihtiyaç olacaktır. Bunun için aşağıdaki resimde işaret edildiği gibi 13 numaralı dijital pinin sağında yer alan üçüncü ve dördüncü pinler kullanılır. Öğrenciler LCD bağlantısını yapmadan önce bu bilginin sınıf ile paylaşılması gerekmektedir.



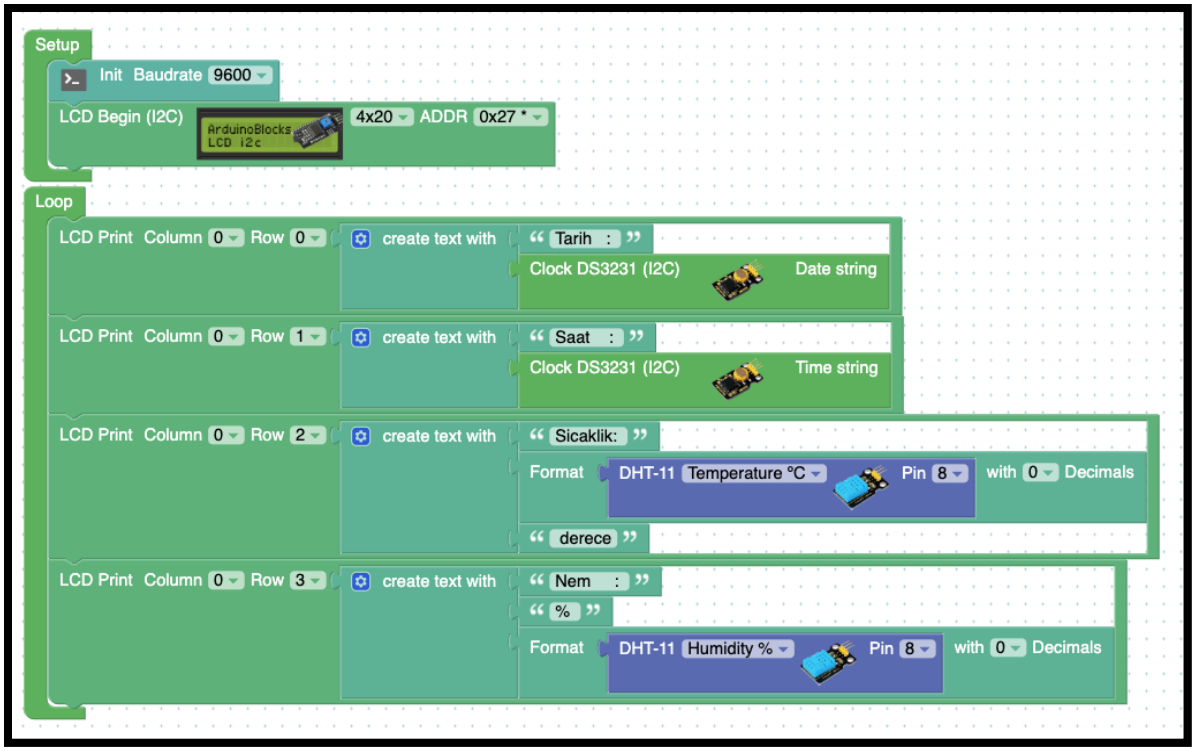
Resim 254. I2C Bağlantı Pinleri

LCD ekranın SDA pini ve SCL pini aşağıdaki resimde görüldüğü gibi Arduino'da ilgili pinlere takılır, GND ve 5V bağlantıları yapılır.



Resim 255. Örnek Devre Şeması

Bu bağlantılar yapıldıktan sonra tarih, saat, sıcaklık ve nem bilgilerinin LCD ekrana yazdırılması için örnek kod bloğu aşağıdaki resimde verilmiştir.



Resim 256. Örnek Program

LCD ekranda sensörden okunan değerin diğer bilgiler ile karıştırılmaması için "text" menüsünün içerisinde yer alan "*create text with*" bloğu kullanılmaktadır. Bu blok birden fazla kaynaktan gelen metni birleştirmek için kullanılır. Yukarıdaki kodda verildiği gibi "Tarih:" etiketi saat modülünden okunan tarih (*Date string*) bilgisiyle birleştirilmiş ve aynı satıra yazılmıştır. Benzer şekilde "Saat:" etiketi saat modülündeki zaman (*Time string*) bilgisi ile eşleştirilmiştir. Nem ve sıcaklık ile ilgili de benzer etiketlemeler yapılmıştır. Nem ve sıcaklık bilgilerinin birimleri de ekranda yazdırıldığı için iki yerine üç bilgi tek satırda birleştirilmiştir. Aşağıdaki resim LCD ekrandaki örnek bir çıktıyı göstermektedir.

Resim 257. Örnek LCD Ekran Çıktısı

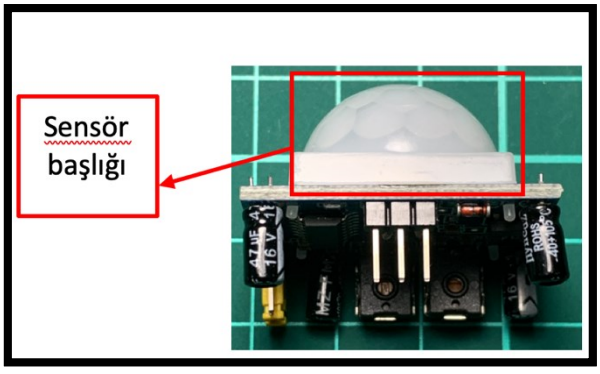
1.3 Gözle ve Uygula- Güvenlik Alarmı

Malzeme Listesi:

Arduino Uno
Breadboard
PIR sensörü
Kapı alarm sensörü
Buzzer
Led
Ses sensörü
Tuş takımı

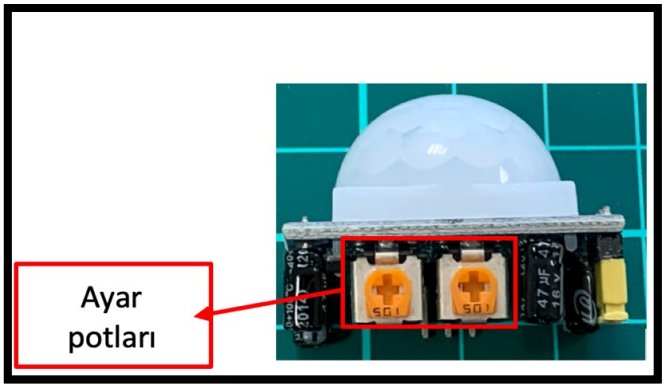
Bu etkinlikte evlerde sıklıkla kullanılan güvenlik alarmlarının bir modeli oluşturulacaktır. Giriş kapısı açıldığında, ayrıca hareket ve ses algıladığında alarmın aktif edilmesi; önceden belirlenen şifre girildiğinde ise alarmın pasif duruma getirilmesi gerekmektedir. Devrede hareket pasif kızılötesi sensör (PIR) ile, sesin algılanması ses sensörü ile ve izinsiz kapı açılışı ise kapı alarm sensörü ile gerçekleştirilecektir. Ayrıca devreye bağlanan tuş takımı ile şifre girişi sağlanarak aktif olan alarm pasif hale getirilecektir. Bahsedildiği üzere üç temel sensör kullanılacaktır. Bu sensörlerin açıklamaları aşağıdaki gibidir.

PIR sensörü (pasif kızılötesi sensör) ortamdaki hareketi algılayan bir sensördür. Çevremizde yer alan tüm nesneler, buna canlılar ve cansız varlıklar da dahil olmak üzere, bir ısı yayar ve bu ısı kızılötesi bir ışımaya neden olur. Kızılötesi ışımalar insan gözü tarafından algılanamaz. Kızılötesi ışımalar ile çalışan farklı araçlar ve sensörler vardır. Bunlardan bazıları ortama kendi kızılötesi ışınlarını gönderir ve bu ışınların çevreden yansımalarını yakalar. Yakaladığı ışımalara göre ortam hakkında bilgi verir. Ancak burada bahsettiğimiz PIR sensörü ortama herhangi bir ışıma yapmaz. Bu nedenle pasif kızıl ötesi bir sensör olarak nitelendirilir. Çalışmaya başladığında ortamdaki kızılötesi ışımaları algılar. Ortama bir nesne ya da canlı girdiğinde PIR sensörü bu değişikliği algılar ve bir sinyal üretir. Etkinliklerde kullanılan PIR sensörünün bir örneği aşağıdaki resimde verilmiştir.



Resim 258. PIR Sensörü

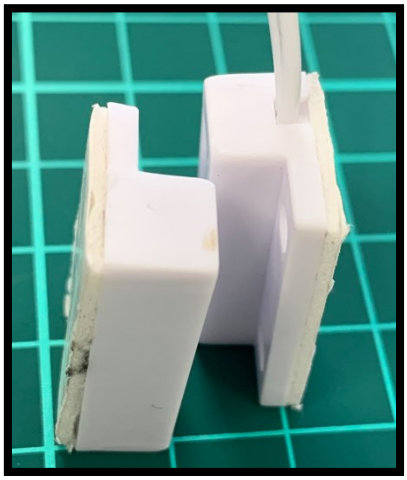
Resimde görüldüğü gibi PIR sensörünün üç adet bağlantı noktası mevcuttur. Bu bağlantı uçlarından iki tanesi sensöre güç vermek için kullanılır. Doğru uçları belirlemek için beyaz renkli sensör başlığı yerinden çıkarılır. Pinlerin üzerine gelen noktalardaki etiketlere göre bağlantılar yapılır. Bu bağlantı uçlarından biri GND bağlantısı için, diğeri ise 5V bağlantısı için kullanılır. Son uç ise sinyal ucudur. Bu uç hareket algılandığında 5V değerinde bir çıkış üretir. Hareket yoksa "0V" çıkış verir. Ara bir değer üretmez. Yani ya hareket var ya da hareket yok bilgisini verir. Bu nedenle Arduino'nun dijital pinleri ile kullanılır. Aşağıdaki resimde PIR sensörü port ayarları görülmektedir.



Resim 259 PIR. Sensörü Ayar Potları

Yukarıda yer alan resimde işaretlendiği gibi PIR sensörünün iki adet ayar potu mevcuttur. Bunlardan solda yer alan pot, hareket algılandığında üretilecek sinyalin süresini belirler. Hareket algılandığında istenirse uzun, istenirse kısa süreli sinyal üretmesi sağlanabilir. Sağda yer alan pot ise hareket algılama mesafesini ayarlamak içindir. Görece olarak yakın veya uzak mesafedeki hareketler algılanabilir.

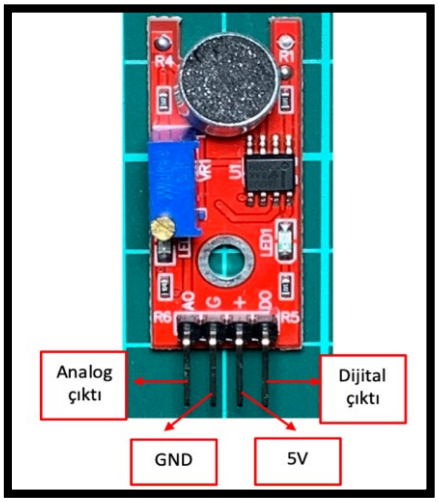
Kapı alarm sensörü aslında basit bir mıknatıs ve elektrik anahtarı gibi çalışan bir röleden (reed röle) oluşur. İki parça bir arada bulunduğunda mıknatısın yaydığı manyetik alan nedeniyle anahtar kapalı konumda kalır ve iki uç kısa devre halindedir. İki parça birbirinden ayrıldığında manyetik alan ortadan kalktığı için anahtar açık konuma geçer. Kapı alarm sensörünün bir örneği aşağıdaki resimde sunulmuştur.



Resim 260. Kapı Alarm Sensörü

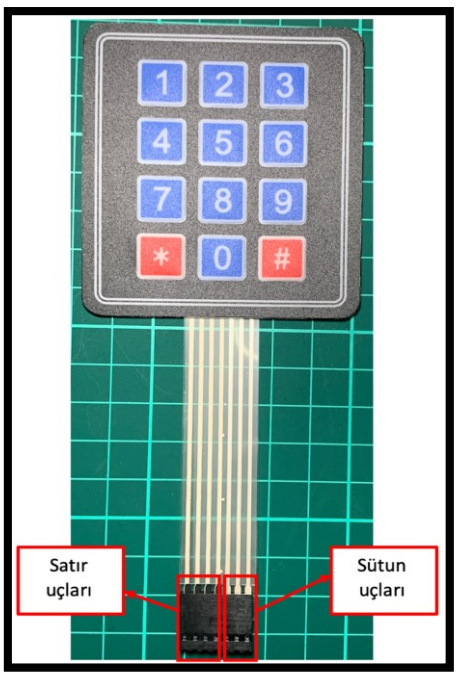
Ses sensörü ortamdaki sesleri algılayarak bu seslere göre sinyal üretir. Ses havada dalgalar halinde yayılır. Bu dalgalar hava basıncında değişikliklere neden olur. Mikrofonlar ve ses sensörleri havadaki bu basınç değişikliklerini algılarlar. Etkinlikte kullanılacak olan ses sensörü hem dijital hem de analog çıktılar üretebilir. Analog çıktı sayesinde ortamdaki ses düzeyi ölçülebilir. Dijital çıktı sayesinde ise sesin belli bir seviyenin üzerinde olup olmadığı tespit edilebilir. Bu seviyeyi ayarlamak için sensörün üzerinde bir ayar potu bulunur. Bu pot sayesinde çıktı vermesi istenen ses düzeyi ayarlanabilir. İstenirse düşük sesler göz ardı edilebilir ya da sensör düşük seslere de duyarlı hale getirilebilir. Aşağıda resmi verilen ses sensörünün üzerinde dört adet bağlantı pini bulunmaktadır. Bunlardan en sağda yer alan pin

üzerinde D0 etiketi taşır ve dijital çıktı elde etmek için kullanılır. En solda yer alan pin ise A0 etiketi taşır ve bu pin 0-1024 arasında ya da 0 ile 100 (%) arasında analog çıktı üretir. G etiketli pin GND, + etiketli pin ise 5V bağlantısı içindir.



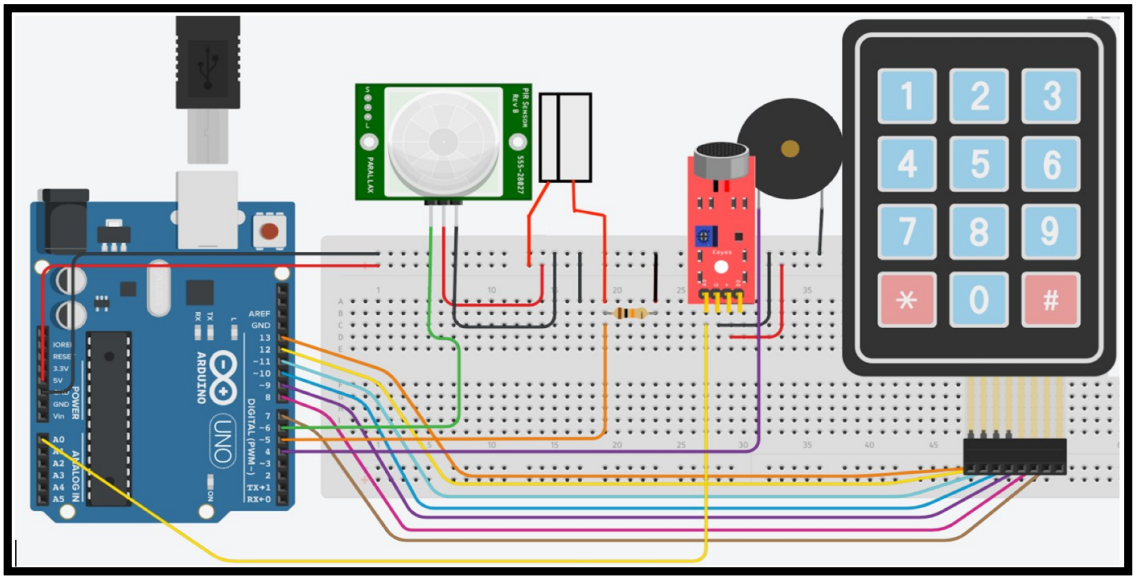
Resim 261. Ses Sensörü

Devrede yer alan sensörlerin dışında alarmı kapatmak için kullanılacak şifrenin girileceği tuş takımı yer alacaktır. Tuş takımı, üzerinde rakamların ve "*" ile "#" karakterlerinin bulunduğu bir tuş takımıdır. Üzerinde 16 tuş ve 12 tuş barındıran farklı tuş takımları olabilir. Bu etkinlikte kullanılacak tuş takımı 12 tuş barındırmaktadır. Tuşlar 4 satır ve 3 sütun oluşturacak şekilde dizilmişlerdir. Bunun yanında tuş takımının Arduino'ya bağlantısının yapılabilmesi için 7 adet çıkış ucu bulunmaktadır. Bu uçlar her bir tuşa değil satır ve sütunlara karşılık gelmektedir. Yani bir tuş basıldığında o tuşun bulunduğu satıra ve sütuna ait bağlantı noktaları aktif olur. Dolayısıyla ilgili kodlama işlemleri yapılırken tuşların satır ve sütun numaralarına dikkat edilmelidir. Aşağıdaki resimde 12 tuştan oluşan (4x3) bir tuş takımı gösterilmektedir.



Resim 262. 3x4 Tuş Takımı

Resimde görüldüğü gibi solda yer alan dört bağlantı noktası satır numaralarına karşılık gelir. Yani birinci uç birinci satıra, ikinci uç ise ikici satıra şeklinde sıralanmışlardır. Aynı şekilde diğer üç uç da sütunları niteler.



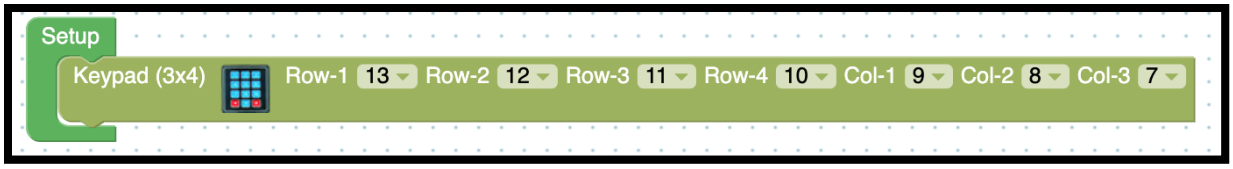
Resim 263. Örnek Devre Şeması

Ses, hareket ve kapı alarm sensörlerinin bağlantıları yukarıdaki görselde görüldüğü gibi gerçekleştirilir. Tuş takımının satır uçları 13, 12, 11 ve 10 numaralı dijital pinlere, sütun uçları ise 9, 8 ve 7 numaralı dijital pinlere takılır.

PIR sensörünün üzerinde yer alan kapak çıkarıldıktan sonra VCC ile işaretli uç dış breadboard'da + hattına, GND ile işaretli olan uç ise – hattına takılır. Ortada yer alan OUT etiketli uç ise sinyal ucudur ve 6 numaralı dijital pine takılır. Bağlantıları yapılan PIR sensörü sabit bir şekilde konumlandırılır. Daha sonra ses sensörünün bağlantılarına geçilir. Bu etkinlikte ses sensörünün analog çıktısı kullanılacaktır. Bu nedenle en solda yer alan A0 ucu bir dişi-erkek kablo yardımı ile Arduino'nun A0 girişine takılır.

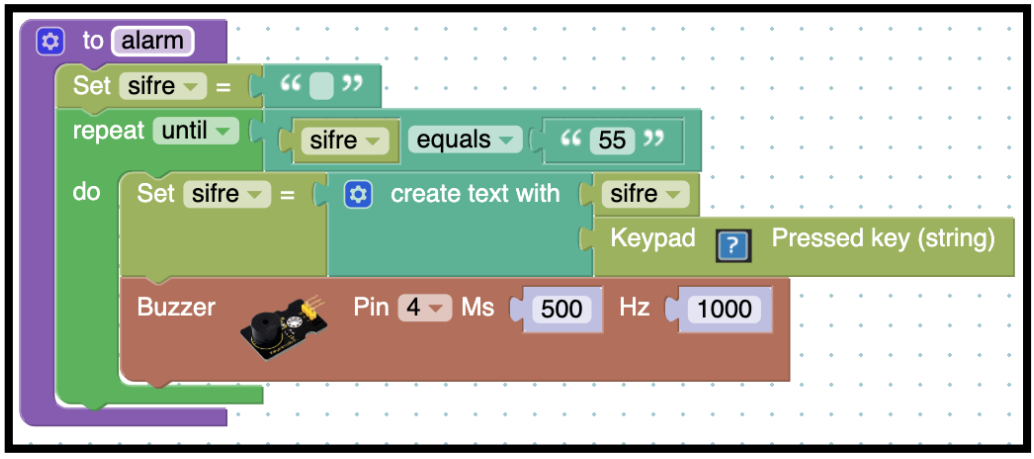
Etkinlikteki son sensör olan kapı alarm sensörünün bağlantısı da yapılmalıdır. Sensörün bir ucu Breadboard'daki artı hattına takılır. Diğer ucu ise devre şemasında görüldüğü gibi herhangi bir noktaya takılır. Bu nokta 1 kohm değerindeki bir direnç yardımı ile – hattına bağlanır. Direnç ile kapı sensörünün bağlı olduğu yerden ise Arduino'nun 5 numaralı dijital pine bir hat çekilir. Aslında bu bağlantı şekli daha önce buton bağlantısında kullanılan pull-down yönteminin aynısıdır. Sensörün iki parçası bir aradayken 5 numaralı dijital pine ON bilgisi, parçalar ayrıldığında is OFF bilgisi gider. Son olarak alarm sesinin çalması için bir buzzer kullanılacaktır. Buzzerın "–" ucu GND'ye, "+" ucu ise Arduino üzerindeki 4 numaralı dijital pine takılmalıdır.

Devre kurulduktan sonra öncelikle tuş takımı tanıtımı setup bloğu içinde yapılmalıdır. Bunun için ArduinoBlocks'ta tuş takımı bölümünde yer alan 3x4 tuş takımı bloğu "Setup"un içine aşağıdaki resimde görüldüğü gibi yerleştirilir ve pin tanıtımı yine gösterildiği gibi yapılır. Blok üzerinde yer alan "Row" kelimesi satır, "Col" kelimesi ise sütun anlamına gelmektedir.



Resim 264. 3x4 Tuş Takımı Setup Ayarı

Devamında ise alarmın çalması ve şifre girilince alarmın kapanmasını sağlayan fonksiyonun yazılması gerekmektedir. Fonksiyon kullanılmasının nedeni alarmın farklı sensörlerden gelen bilgiler ile aynı şekilde çalışacak olması nedeniyle kodları tekrarlamanın önüne geçmektir. Fonksiyon aşağıdaki resimde görüldüğü gibi olmalıdır.



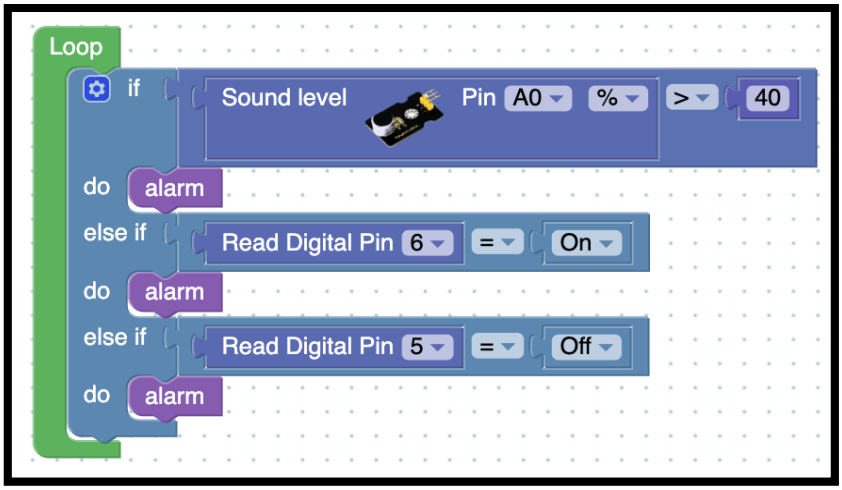
Resim 265. Örnek Alarm Fonksiyonu

Fonksiyona alarm ismi verildikten sonra şifre bilgisinin tutulacağı bir değişken oluşturulmuş ve içerisine boş bir değer atanmıştır. Bunun nedeni alarm her kapatıldığında ve bir sonraki çalışmasından önce yeni şifre girişine olanak sağlamaktır. Kod bloğunda görüleceği gibi şifre 55 olarak belirlenmiştir. "*Repeat until"* döngüsü ile şifre değişkeni 55 olana kadar döngü devam edecektir.

Dikkat

Şifre değişkeni sayısal değil metinsel bir bilgi tutar. Dolayısı ile içerisinde rakamlardan oluşan bir bilgi olsa bile bir metin olarak işlem görür. Bu nedenle şifre değişkeninin içeriğinin "55" olup olmadığı "=" yerine text menüsünün içerisinde yer alan "equals" bloğu ile yapılmıştır.

Daha sonra "Keypad? Pressed key (string)" komutu ile basılan tuş bilgisi metinsel bilgi olarak şifre değişkenine eklenir. Bu sırada 4 numaralı dijital pin "ON" yapıldığı için bağlı olan buzzer çalmaya başlar. Şifre değişkeni 55 olana kadar çalmaya devam eder. Şifre girilince "OFF" konumuna geçen dijital pin buzzer'ı kapatır. Alarm fonksiyonun devamında ana kod bloğunun hazırlanması gerekir. Bunun için aşağıdaki resimde görülen kod bloğu yazılmalıdır.



Resim 266. Örnek Program

Kod temelde bir "if" ile iki "else if" koşul satırından oluşur. if satırı ses sensörü içindir. Sensör bölümünün içinde yer alan ses sensörü bloğunun yakaladığı ses değeri 40'tan fazla ise alarmın çalması sağlanır. Burada değer yüzde olarak okunduğu için kesim noktası 40 olarak belirlenmiştir. İstenirse 0 ile 1024 arası bir değer de kullanılabilir. Devamında 6 numaralı pine bağlı olan PIR sensörünün ON olması kapı alarm sensörünün çalışmasını sağlayacaktır. Bunların dışında kapı sensörünün OFF olması durumu da alarm fonksiyonu çalıştıran bir diğer unsurdur.

Bu noktada öğrencilere fonksiyon kullanılmasaydı kodların nasıl değişeceği sorulmalı ve cevapları sınıfça tartışılmalıdır. Ayrıca fonksiyon yazarken öğrencilerin kendi şifrelerini belirlemeleri istenir.

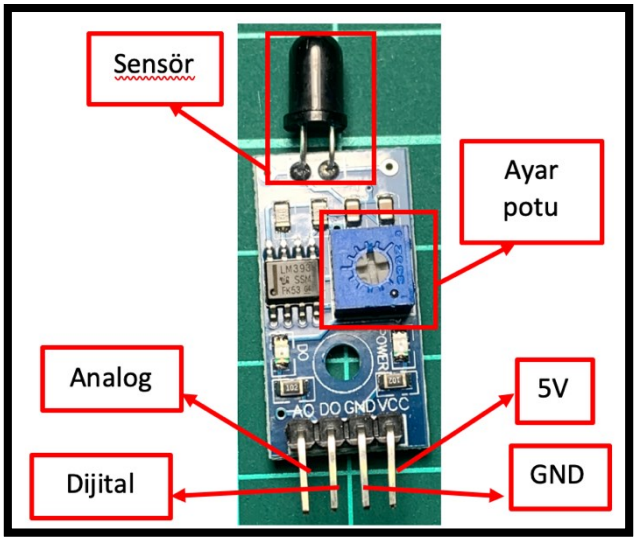
1.4 Gözle ve Uygula- Yangın ve Gaz Alarmı

Malzeme Listesi:

Arduino Uno
Breadboard
Alev sensörü
Yanıcı gaz sensörü
Buzzer
Servo motor

Bu etkinlikte amaç alev ve yanıcı gaz sensörlerini kullanarak, yanıcı gaz ya da gaz kaçağı algılandığında eve gelen doğal gazı kesen, yangın algıladığında ise alarm çalarak uyarı veren bir devre tasarlamaktır. Bunun için alev sensörü ve yanıcı gaz sensörü birlikte kullanılacaktır. Aşağıda bu sensörlerin çalışma prensipleri açıklanmıştır.

Alev sensörü, belirli bir noktada ortaya çıkmış olan yangın ya da alevi algılar. Alev aslında kendine has bir dalga boyunda ışık yayar. Alev sensörü bu dalga boyundaki ışığı yakalar, varlığı ve uzaklığı hakkında bilgi üretir. Olası yangınlar için oluşturulacak erken uyarı sistemlerinin vazgeçilmez bir parçasıdır. Elektronik programlama dersi etkinlikleri kapsamında aşağıda resmi sunulan ve kart bağlantılarının yapılmasını sağlayan modülüyle birlikte sunulmuş alev sensörü kullanılacaktır. Bu sensör 20 cm ile 100 cm uzaklık aralığında bulunan alevi tespit edebilir. Alev var ya da yok bilgisini dijital olarak verebildiği gibi, alev büyüklüğünü analog olarak da bildirebilir.

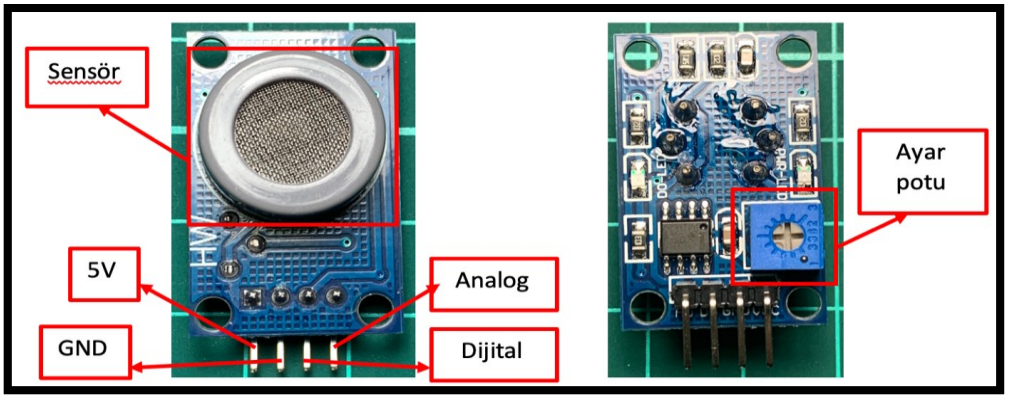


Resim 267. Alev Sensörü

Resimde yer alan uçlardan en sağdaki ve onun bir yanındaki uçlar alev sensörü modülünün güç bağlantılarını yapmak için kullanılır. Sağdaki uç 5V, diğer uç ise GND'ye bağlanmalıdır. GND ucunun solunda yer alan bağlantı ucu ise dijital çıkış ucudur. Alev algılandığında 5V bir çıkış üretir. Alev yok ise 0V şeklinde bir çıkış üretir. Yukarıda bahsedildiği gibi bu işlemi 100 cm'den daha yakın alev için gerçekleştirebilir. Üzerinde yer alan ayar potu ile hassasiyet ayarı yapılabilir. Bu ayar potu sayesinde istenirse en ufak alevi algılamak mümkün olabilirken belli bir büyüklüğe ulaşmayan alevler görmezden gelinebilir.

Bu etkinlikte 5V ve GND bağlantıları yapılmış breadboard üzerine alev sensörü takılır. GND ve 5V bağlantıları breadboard'un – ve + hatları ile bağlanır. Dijital çıktı üreten D0 pini ise Arduino'nun herhangi bir dijital pinine takılabilir. Örneğin 12 numaralı pin bu iş için kullanılabilir. Bu şekilde alev sensörünün bağlantısı tamamlanmış olur.

Gaz kaçaklarının algılanması için de yanıcı gaz sensörü MQ2 kullanılacaktır. Yanıcı gazların ve dumanların insanların solunum yaptığı ortama serbestçe yayılması büyük tehlikeler oluşturur. Bu gazların çoğu kokusuz olduğu için insan burnu tarafından da çok zor algılanabilir. Bu nedenle havadaki yanıcı gazların ve dumanın algılanması için sensörler ve bu sensörlere bağlı olarak çalışan uyarı ve alarm sistemleri birçok yerde bulunur. Elektronik programlama dersinde kullanılacak olan gaz sensörü aşağıda resmi verilen MQ2 yanıcı gaz sensörüdür. Bu sensör yanıcı gazların yanında dumanı da algılar. Ortamda gaz ve duman olduğunda sensör içerisindeki maddeler bu gazla kimyasal tepkimeye girer. Tepkime sonucu bir ısı açığa çıkar ve sensörün sıcaklığı artar. Sensör içerisindeki sıcaklığı ölçen başka bir araç da sıcaklık bilgisini çıkış uçlarına yönlendirir.



Resim 268. MQ2 Gaz Sensörü

Dikkat

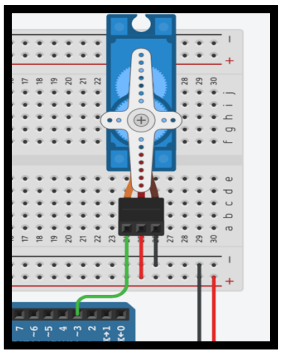
Sensör yanıcı gaz varlığını tespit etmek için kullanıldığında ısınır. Dolayısıyla bu işlem sonrasında elle temastan kaçınılmalıdır. Aksi takdirde yanık olayı gerçekleşebilir.

Resimde görüldüğü üzere diğer sensörlerdeki gibi güç girişi bağlantıları bulunmaktadır. Önden bakıldığında en solda yer alan uç 5V ve onun yanındaki uç ise GND bağlantısı içindir. Üçüncü uç dijital çıktı için kullanılır 0 ile 5V arasında bir değer üretir. Kart üzerinde ayar potu aracılığı ile belirlenmiş eşik değeri aşıldığında ON konumuna geçer; aşılmadığında ise OFF konumundadır. Son uç ise analog çıktı içindir.

Yukarıda belirtilen yanıcı gaz sensörünün devreye bağlanması için öncelikle sensör breadboard üzerinde uygun bir noktaya takılır. GND ve 5V bağlantıları aynı şekilde yapılır. Dijital çıktı üreten pini ise Arduino üzerinde herhangi bir dijital pine takılabilir. Örneğin bu etkinlikte 7 numaralı dijital pin kullanılabilir.

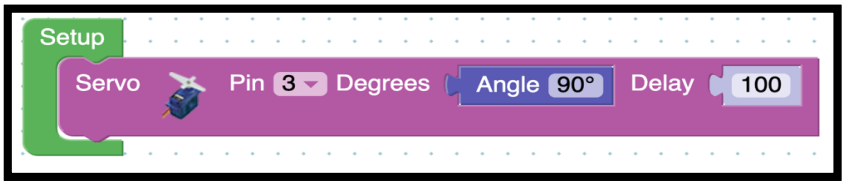
Sensör bağlantıları yapılan devrede alarm çalacak olan buzzer ve gazı kesecek olan servo motor bağlantılarının yapılması gerekir. Öncelikle buzzer breadboard'a takılır. Eksi ucu GND hattına, + ucu ise 100 Ohm direnç üzerinden bir dijital pine takılır. Bu etkinlikte buzzer için 11 numaralı dijital pin kullanılacaktır.

Son olarak servo motor bağlantıları yapılmalıdır. Servo motorun + ve – bağlantıları yapıldıktan sonra aşağıdaki görselde resimde görüldüğü gibi sinyal ucu "~" işaretli pinlerden (PWM) birine takılır. Bu etkinlikte kullanılacak olan pin 3 numaralı dijital pindir.



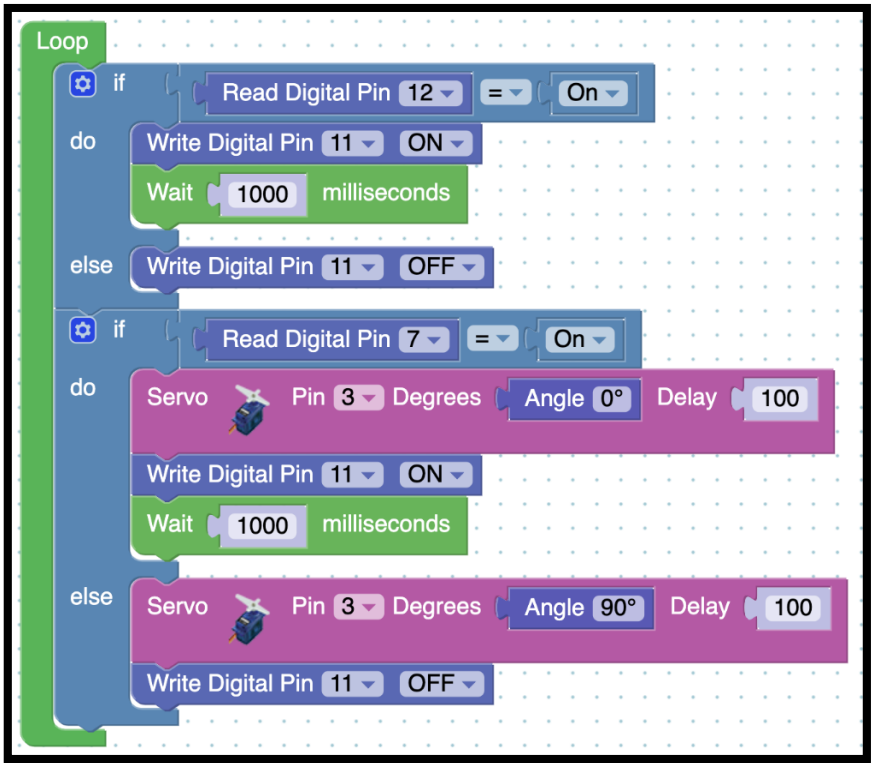
Resim 269. Servo Motor Bağlantısı

Devre için gerekli kodların yazımından önce servo motorun 90 dereceye ayarlanması gerekir. Bunun için aşağıdaki kodun yazılması ve servo motorun açısının ayarlanması gerekir. Ancak kodu çalıştırmadan önce servo motorun üzerindeki aparat çıkarılmalı ve açı ayarlandıktan sonra servo motora aşağıdaki resimde görüldüğü gibi yeniden takılmalıdır.



Resim 270. Servo Motor Ayarı

Rehber öğretmen servo motorun açısının tüm öğrenciler tarafından doğru şekilde ayarlandığından emin olduktan sonra kodlama aşamasına geçer. Bu etkinlikteki sensörlerin kullanımı bir önceki etkinlikte yer alan sensörlerin kullanımına benzediği için rehber öğretmen öğrencilere süre vererek kendi kodlarını yazmanlarını ister. Ancak kod yazma işlemine geçmeden önce servo motorun gaz kesme işlemini yapmakla görevli olduğu, mevcut konumunun gazın açık, 90 derece sağa dönmüş halinin ise kapalı anlamına geldiği öğrencilere aktarılmalıdır. Örnek kod aşağıdaki resimde görüldüğü gibi olmalıdır.



Resim 271. Örnek Program

Kod içinde yer alan birinci "if" bloğu yangın durumunda alarmın çalmasını sağlar. İkinci "if" bloğu ise yanıcı gaz algılanması durumunda hem alarmı çalar hem de gazı kesecek olan servo motoru 90 derece döndürür. Yukarıdaki kod öğrenciler ile paylaşılmalıdır. Yanıcı gaz sensörünün ve alev sensörünün çalışıp çalışmadığını kontrol etmek için basit bir çakmaktan faydalanılabilir.

Dikkat

Çakmak kullanımı sırasında öğrenciler mutlaka rehber öğretmen eşliğinde çalışmalıdır. Olası kazaların önüne geçmek için mutlaka gerekli önlemler rehber öğretmen tarafından alınmalıdır.

Devre ve kod bloğunun çalışması test edildikten sonra etkinliği erken tamamlayan öğrenciler isterlerse alarm sesinde farklı uyarı tonları elde edebilirler. Bunun için öğrencilere ek süre verilebilir.

2. TASARLA VE ÜRET

2.1 Tasarla- Hayalimdeki Akıllı Ev

Gerekli malzemeler: Aşağıda listelenen sensörlerden en az iki tanesi, şu ana kadar etkinliklerde kullanılmış olan sensör ve modüllerden en az üç tanesi, maket ev yapımı için karton, makas ve yapıştırıcı, Arduino UNO, breadboard, gerekli durumlarda kullanılmak üzere lehim ve havya, bağlantı kabloları

Bu etkinlikte öğrenciler hayallerindeki akıllı evi tasarlayacaklardır. Akıllı ev tasarlanırken bu bölümde kullanılmış olan nem-sıcaklık, PIR, ses, kapı alarm, alev ve yanıcı gaz sensörlerinden en az üç tanesini kullanmalıdır. Bunun yanında bu bölümde anlatılmayan ve pratik kullanımının nasıl olduğu öğrenciler tarafından çözülmesi beklenen ve aşağıda açıklamaları verilen sensörlerden de en az iki tanesinin kullanılması gerekmektedir. Öncelikle aşağıda bulunan, sensörlerin kullanımları ile ilgili hazırlanmış açıklamaların çıktısı alınıp öğrencilere verilir. Daha sonra öğrencilere bu sensörleri de dikkate aldığınızda "hayalinizdeki akıllı ev nedir?" sorusu yöneltilir ve öğrencilerin hayallerindeki evin özelliklerini maddeler halinde yazmaları istenir. Örneğin;

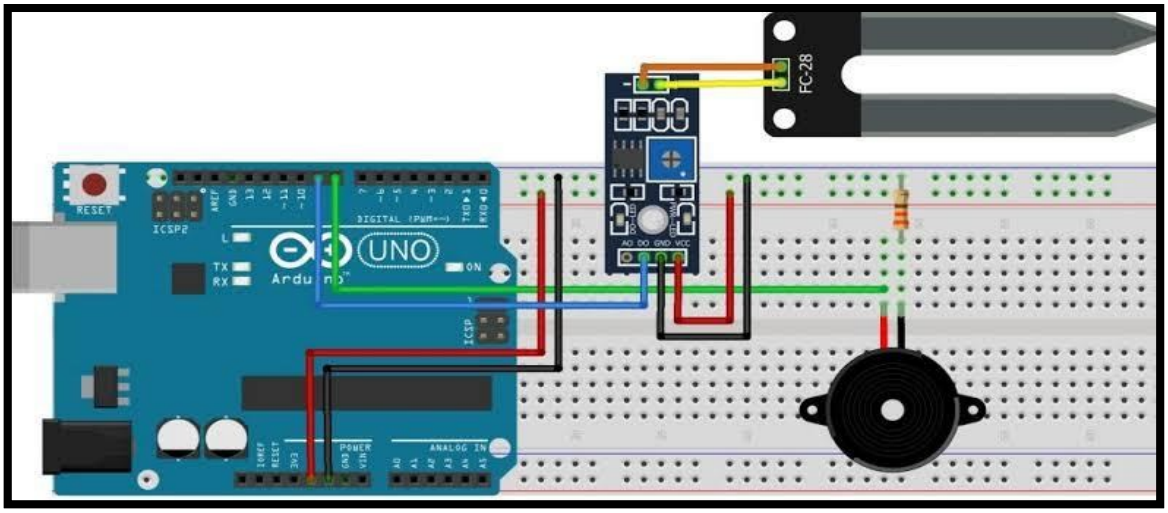
Hayalimdeki evde;

1. Kapı anahtarsız olarak şifre ile açılmalı,
2. Kapı şifre kullanılmadan zorla açıldığında sesli ve ışıklı alarm uyarısı olmalı,
3. Bitkiler otomatik sulanmalı,
4. Yağmur yağdığında balkonun üstü kapanmalı.

Daha sonra hayal ettikleri özelliklerin hangi sensör ve modüller ya da devre elemanları ile yapılabileceği üzerinde düşünmeleri istenir. Belirledikleri malzemeleri ilgili maddenin yanına yazmaları istenir. Arduino kartının sınırlı sayıda bağlantısı olması nedeniyle oluşabilecek durumlara dikkat etmeleri istenir. Gerekli durumlarda öğrencilere ikinci Arduino kartları temin edilebilir.

Kullanılabilecek sensörler ve açıklamaları aşağıdaki gibidir (çıktı alınıp öğrencilere verilecek):

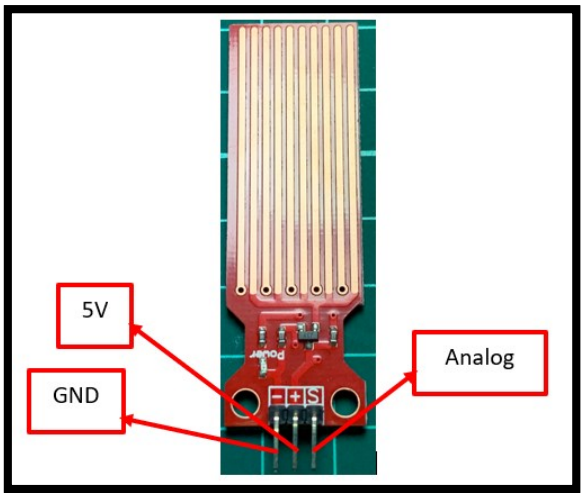
Toprak nem sensörü*:* Toprak nem sensörü toprağın taşıdığı nemi ölçmek için kullanılan bir sensördür. Tüm bitkiler gelişebilmek için kendilerine özgü bir toprak nemine ihtiyaç duyar. Bazı bitkiler çok fazla neme ihtiyaç duyarlar ve bu nedenle çokça sulanmaları gerekir. Bazıları ise çok daha az neme ihtiyaç duyarlar ve fazla sulandıklarında kökleri çürür ve ölürler. Doğru sulamanın yapılması bitkilerin gelişimi için önemlidir. Toprak nem sensörü de bu anlamda toprak nemini ölçerek sulama zamanının gelip gelmediğinin anlaşılmasını sağlayabilir. Etkinliklerde kullanılacak toprak nem sensörü aşağıdaki resimde verilmiştir. Resimde görüldüğü gibi toprak nem sensörü iki bölümden oluşmaktadır. Birinci bölüm toprağa batırılan bölümdür (sağda). Bu kısım görüldüğü gibi toprağa giren iki sivri bacaktan oluşmaktadır. Yaptığı işlem bu bacağın arasındaki direnci ölçmektir. Toprağa batırılmadığı durumda iki bacak arasındaki direnç çok yüksektir. Toprağa batırıldığında ise topraktaki nem miktarına göre direnci değişir. Topraktaki nem miktarı arttıkça iki bacak arasındaki direnç de azalır.



Resim 272. Toprak Nem Sensörü Bağlantısı

İkinci bölüm (soldaki) ise toprağın içinde yer alan ve özünde bir direnç ölçer olan parçadan gelen bilgileri dijitalleştirir. Yani Arduino tarafından okunabilir hale getirir. Bu kartın bir tarafında iki, diğer tarafında ise dört adet bağlantı ucu bulunur. İki adet bağlantı ucu olan taraftaki uçlar sensörün yukarıda tarif edilen birinci bölümüne bağlanır. Bu şekilde kart birinci bölümden gelen değerleri okuyabilir. Bağlantılar dişi-dişi kablolar aracılığıyla yapılır. Diğer tarafta yer alan dört kablodan iki tanesi güç bağlantıları içindir. Sol tarafta yer alan uç 5V bağlantısı, onun yanındaki ise GND bağlantısı içindir. Üçüncü uç dijital çıktı almak için kullanılır. Son uç ise analog çıktı içindir. Kartın üzerinde yer alan ayar potu dijital çıktının ON/OFF olma noktasını ayarlamak için kullanılır. Yani nemli-nemsiz durumun eşik noktasını ayarlamak içindir. Analog çıkış ise nemin miktarına göre 0V-5V arası değer üretir. Arduino'nun analog girişine bağlandığında 0 ile 1024 arasında bir değer okunur.

Yağmur sensörü: Yağmur sensörü, aynı toprak nem sensörü gibi sıvı ile temas ettiğinde oluşan direnç değişikliğini algılayan bir sensördür. Toprak nem sensöründen farklı olarak aşağıdaki resimde görüldüğü gibi üzerinde iletken şeritler bulunur. Yağmur suyu gibi bir sıvı ile temas ettiğinde şeritler arasında bir akım oluşur ve bu akım Arduino'ya analog bir sinyal olarak gönderilir. Bu özelliği sayesinde yağmurun yağıp yağmadığının tespitinin yanında 4 cm derinliğe kadar su seviyesi de ölçülebilir. Sensörün plakalarının suya batırılması ile aynı sıvı temasındaki gibi bir sinyal üretilir. Su ne kadar derin olursa sinyal de o kadar güçlenir.

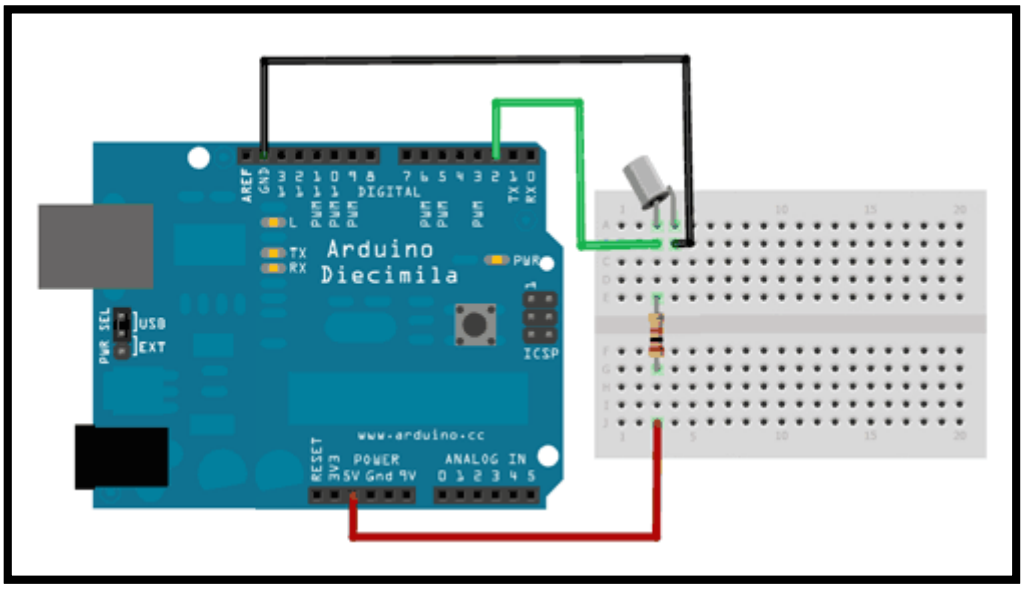


Resim 273. Nem Sensörü

Yağmur sensörünün üzerinde üç adet pin bulunur. Bunlardan solda yer alan iki tanesi sensörün çalışması için gerekli olan gücü sağlayan GND ve 5V bağlantı uçlarıdır. Son uç ise sinyal ucudur. Sinyal ucu Arduino kartı üzerindeki analog girişlerin herhangi birine bağlanabilir. 0 ile 1024 arasında değer üretir.

Tilt: Tilt sensörü bir diğer adı ile eğim sensörü aslında basit bir anahtar görevi görür. Küçük bir silindirin içerisinde metal bir bilye bulunur. Bu metal bilye sayesinde sensörün dik veya eğimli olduğu anlaşılır. Tilt sensörü dik durumda iken bilye sensör içerisindeki iki ucun birbirine temas etmesini sağlar ve akım iletilir. Ancak sensör bir tarafa doğru yatırılıp içerideki bilyenin hareket etmesi sağlanırsa bu sefer temas eden uçlar birbirinden ayrılmış olur ve akım iletilmez. Bu şekilde tilt sensörünün bir tarafa eğildiği ya da ters döndüğü anlaşılır. Tilt sensörü belli cisimlerin üzerine bağlanarak o cismin dengede olup olmadığı ya da düşüp düşmediği anlaşılabilir. Devreye bağlantısı standart push buton ile aynıdır. Aşağıdaki resimlerde tilt sensörü ve bağlantısı görülmektedir.

Resim 274. Tilt Sensörü



Resim 275. Tilt Sensörü Bağlantısı

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Bu bölümde görülen sensörlerin birçoğunda dijital ve analog çıktı bacakları bulunmaktadır. Dijital ve analog çıktı bacakları arasındaki fark nedir?

2. Bu öğrenilen konular günlük yaşamda nerelerde kullanılabilir?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaşılan problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Tuş takımının satır ve sütun bağlantılarını anlamakta ve kodlamakta zorlandım.
    - Alarmı kapatmak için gereken şifre kodunu yazmakta zorlandım.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız
- Grup arkadaşınızdan ne öğrendiniz?

8. Bölüm-Nesnelerin İnternetine Giriş

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve "led/direnç/buzzer/breadboard/Arduino" bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler "VE/VEYA" kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol, kondansatör, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.

Bölümün Kazanımları:

- Kızıl ötesi kumanda seti, "RC522 RFID NFC" kiti ve "ESP8266 wifi modülü" kullanılan elektronik devreleri oluşturabilir.
- Kızıl ötesi kumanda seti, "RC522 RFID NFC" kiti ve "ESP8266 wifi modülü" kullanılan elektronik devreleri programlayabilir.
- Kızıl ötesi kumanda ile bir aracı kumanda etmek gerekli elektronik devreyi oluşturup programlayabilir.
- Nesnelerin interneti (IOT) ile ilgili elektronik devreler oluşturup programlayabilir.
- Öğrenciler sensörlerden alınan verilerin internete aktarımını sağlamak için gerekli elektronik devreleri oluşturup programlayabilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin sensörler, kızıl ötesi modüller ve Wifi bileşenlerini elektronik devrelerde kullanabilmelerini ve bu elektronik devrelerin iletişimi için gerekli bileşenleri ArduinoBlocks ortamında programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, kızıl ötesi kumanda seti, 1 servo motor, boş pet şişe, RC522 RFID NFC kiti, direnç (1 adet 330 ohm, 3 adet 1 kohm), 1 adet led, ESP8266 01 wifi modülü, sıcaklık ve sem sensörü, 100 µF kondansatör (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Kızıl ötesi kumanda ile servo motor kontrolü, kart okuyucu bilgilerinin İnternete aktarılması ve ev ortamı ile ilgili çeşitli sensörlerden alınan bilgilerin İnternete aktarılması ve başka kullanıcılar tarafından görülebilmesi için gerekli elektronik devrelerin oluşturulması ve bu elektronik devrelerin ArduinoBlocks ortamında programlanmasının gösterilmesi.

Uygula: Kızıl ötesi kumanda ile servo motor kontrolü, kart okuyucu bilgilerinin İnternete aktarılması ve ev ortamı ile ilgili çeşitli sensörlerden alınan bilgilerin İnternete aktarılması ve başka kullanıcılar tarafından görülebilmesi için gerekli elektronik devrelerin oluşturulması ve bu elektronik devrelerin ArduinoBlocks ortamında programlanmasının gösterilmesi.

Tasarla: Su doluluk oranın tespit edilerek IOT ile internete aktarımı için gerekli elektronik devrelerin oluşturulması ve programının yapılması için tanımlama ve fikir yürütme adımlarının gerçekleştirilmesi.

Üret: Su doluluk oranın tespit edilerek IOT ile internete aktarımı için gerekli elektronik devrelerin oluşturulması ve programının yapılması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1.GÖZLE ve UYGULA

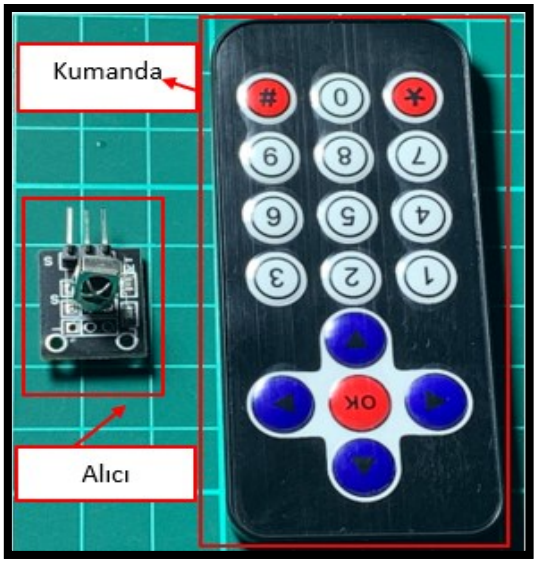
1.1 Gözle ve Uygula- Uzaktan Kumandalı Yem Makinesi

Malzeme Listesi:

Arduino Uno
Breadboard
Kızıl ötesi kumanda seti
Servo motor
Boş pet şişe

Bu etkinliğin amacı kızılötesi kumanda ile bir aracı kontrol etmektir. Etkinlikte bir servo motor kontrol edilerek motora bağlı olan bir şişe kapağı açılıp kapatılacaktır.

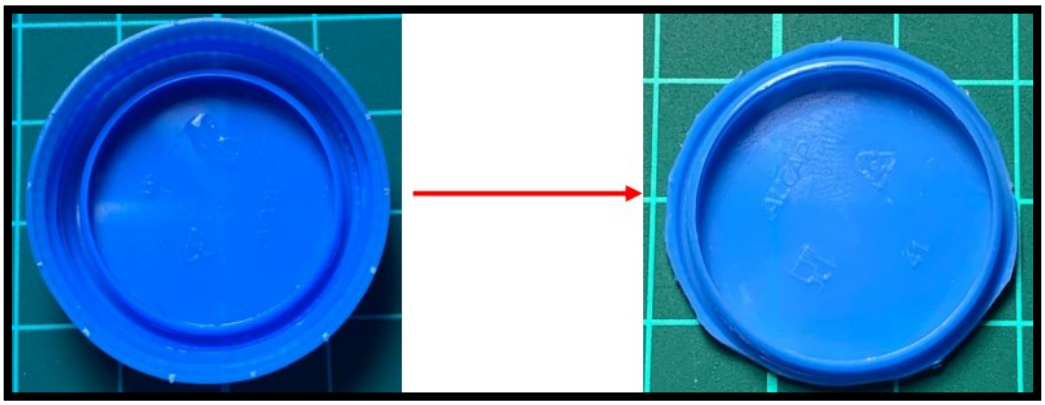
Kızılötesi kumanda kiti bir kızıl ötesi ışık kaynağı ve bu kaynaktan çıkan kızılötesi ışığı algılayan sensörden oluşur. Televizyonları, klimaları ve benzeri cihazları çalıştıran kumandalar ile aynı çalışma prensibine sahiptir. İstenirse çevrede bulunan diğer kumandalar da bu etkinlikte kullanılabilir. Temel çalışma prensipleri yaklaşık olarak daha önceki etkinliklerde kullanılan kızılötesi sensörler ile aynıdır. Kitte yer alan kumanda belirli bir frekansta kızılötesi ışık yayar. Başka bir deyişle çok hızlı bir şekilde ışığı yakıp söndürebilir. Işık yandığında 1, söndüğünde 0 bilgisi gönderilir. Kumanda üzerinde bir tuşa basıldığında, bu tuşa özel 0 ve 1 bilgilerinden oluşan bir kod gönderilir. Uzaktaki sensör bu 0 ve 1 bilgilerinde oluşan kodu aynı sıra ile algılar. Sonuç olarak bu sıralama Arduino tarafından okunur. Sonrasında okunan koda göre program yazılarak her tuşa bir görev atanabilir. Aşağıda kızılötesi sensörün bir resmi verilmektedir.



Resim 276. Kızıl Ötesi Kumanda

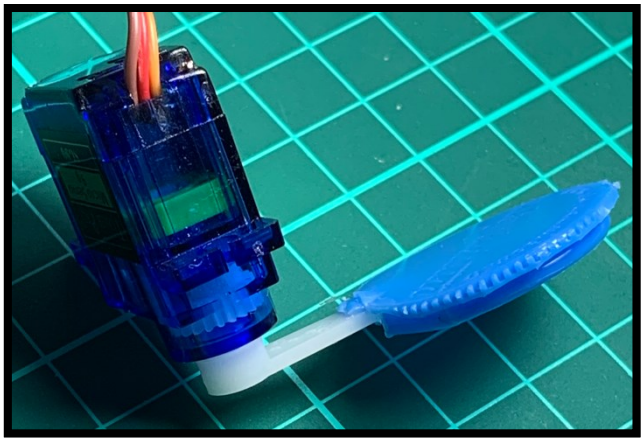
Kumandanın kızılötesi sinyal gönderebilmesi için CR 2025 kodlu saat piline ihtiyaç vardır. Setlerde bulunan kumandalar içerisinde saat pili bulunmalıdır. Pilin bulunmadığı kumandalarda setlerde bulunan saat pili kumandanın arkasında yer alan pil yuvasının içerisine yerleştirilmelidir. Setin diğer parçası ise alıcı sensördür. Yukarıdaki görselde görüldüğü gibi üç adet bağlantı ucu vardır. Bu uçlardan sağda yer alan 5V, ortada yer alan ise GND bağlantısı içindir. En soldaki uç ise sinyal ucudur. Yakından bakıldığında yanında S etiketi bulunur. Bu uç bir dijital pine takılarak kumandanın gönderdiği kızılötesi sinyaller okunur.

Etkinlikte ilk olarak kullanılacak malzemeler hazırlanmalıdır. Bunun için bir pet şişenin kapağının kenarı bir makas ya da bıçak yardımı ile aşağıdaki resimde görüldüğü gibi kesilmelidir. Rehber öğretmen kesme işlemi esnasında öğrencilerin dikkatli çalışmasını sağlamalıdır. Öğrencilerin kendilerine zarar verme ihtimali olduğunu düşünüyorsa ya birebir kendi gözetiminde kesme işlemini yapmalarını sağlamalı ya da kesme işlemini kendisi yapmalıdır. Fakat öğrencilerin yapabileceğini düşünüyorsa onlara fırsat tanımalıdır.



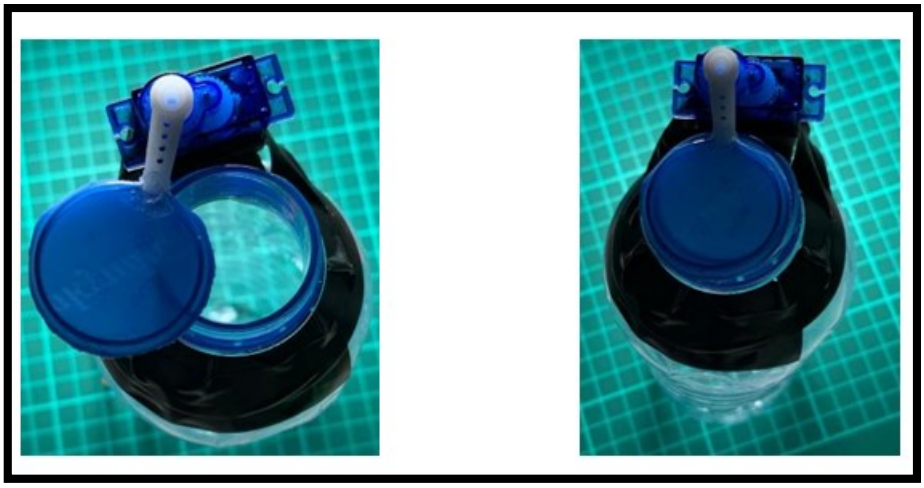
Resim 277. Kapak Kesim İşlemi

Daha sonra kesilerek hazırlanan pet şişe kapağı aşağıdaki resimde görüldüğü gibi servo motora vidalanır ya da yapıştırılır.



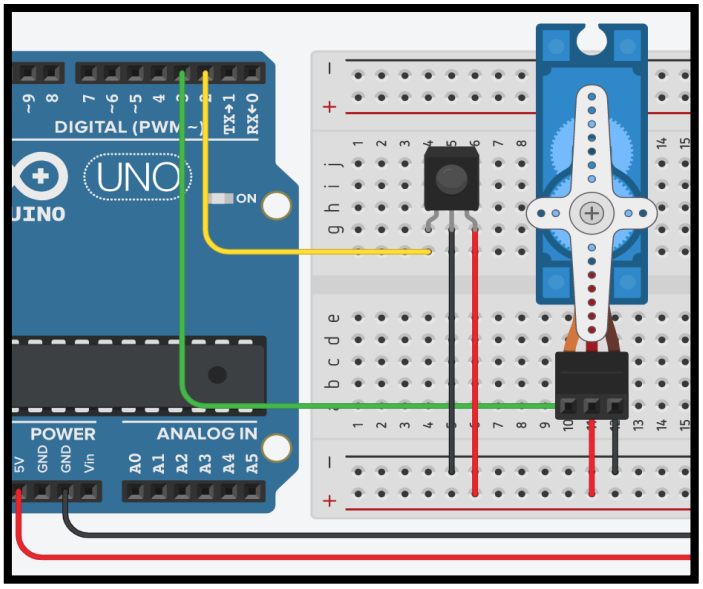
Resim 278. Servo motor kapak bağlantısı

Servo motor takılmadan önce açısı 90 dereceye ayarlanır ve sonrasında pet şişeye bant ya da yapıştırıcı yardımıyla aşağıdaki resimlerde gösterildiği gibi monte edilir. Monte etme işleminden önce pet şişenin taban kısmı kesilmelidir. Şişeyi doldurmak için bu kesik kısım kullanılır.



Resim 279. Servo Motor, Kapak ve Şişe Bağlantısı

Malzemeler hazırlandıktan sonra aşağıdaki resimde görülen devre kurulmalıdır. Devredeki görülen siyah eleman kızıl ötesi algılayıcıdır. Algılayıcının sinyal bacağı Arduino kartın iki numaralı dijital pinine takılmıştır.



Resim 280. Örnek Devre Şeması

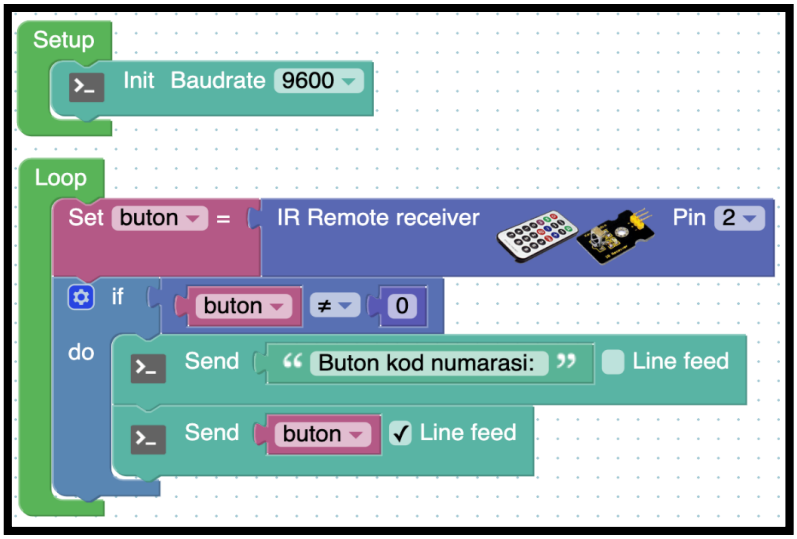
Dikkat

Yukarıdaki devrede servo motor şematik olarak gösterilmiştir ve gerçek uygulama ortamındaki şişeye takılı servo motoru sembolize etmektedir.

Devre kurulduktan sonra kodlamaya geçmeden önce kumandanın tuşlarının gönderdiği kodların belirlenmesi gerekmektedir. Bu etkinlikte kapağı çevirecek olan kumanda üzerindeki iki tuş belirlenmelidir. Akılda kalması açısından sağ ve sol ok tuşları seçilebilir. İsteyen öğrenciler farklı tuşları da kullanabilir. Tuş kodlarının belirlenmesi için en iyi yol okunan kodların seri port ekranına yazdırılmasıdır. Bunun için aşağıdaki resimde gösterilen Sensors blok başlığı içerisindeki "*IR Remote receiver"* komutu kullanılır.

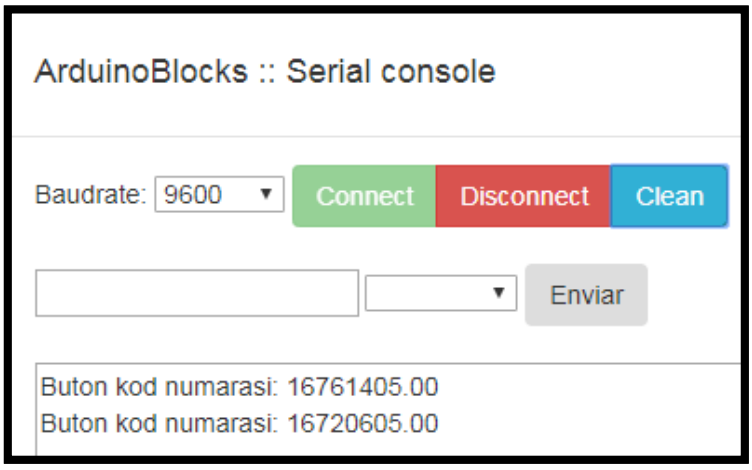
Resim 281. IR Remote Receiver Bloğu

Bu blok basılan tuşun kodunu döndürür. Aşağıda resimde verilen kod Arduino'ya yüklendikten sonra seri port ekranı açılır ve Connect tuşuna bastıktan sonra kumandanın sağ ve sol ok tuşlarına sırası ile basılır. Kodun doğru çalışması için alıcının takıldığı pinin doğru seçilmesi gerekir. Bu etkinlikte alıcı 2 numaralı dijital pine sinyal göndermektedir.



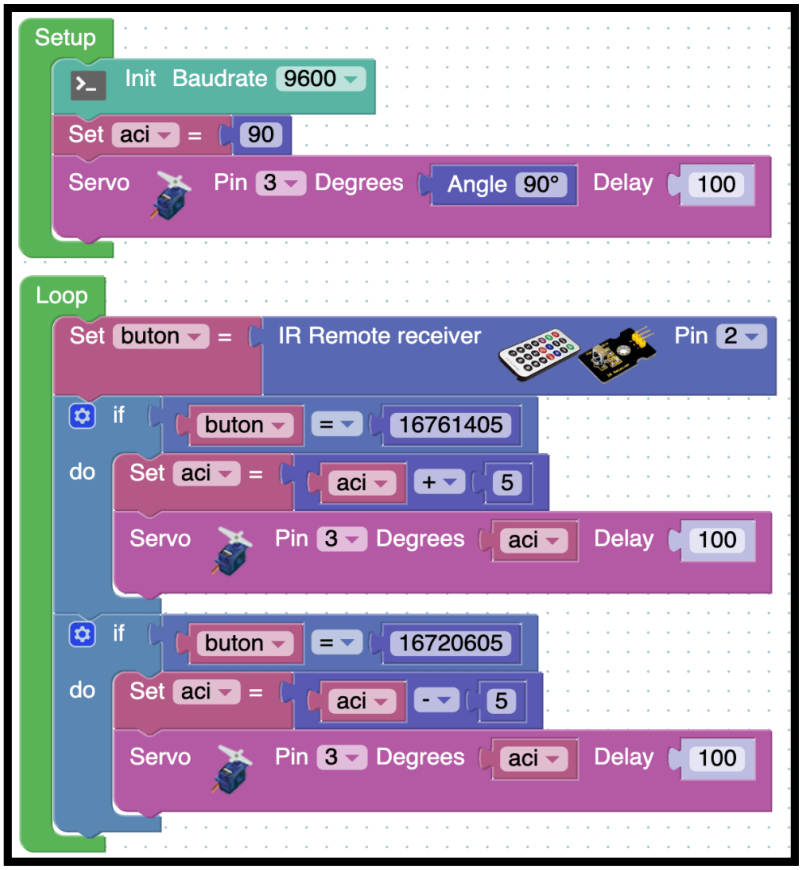
Resim 282. Kumanda Kod Okuma Programı

Seri port ekranında aşağıdaki resimde görülene benzer bir çıktı elde edilmesi gerekir. İlk sayı sağ ok tuşu, ikinci sayı ise sol ok tuşunun kodunu vermektedir. Herhangi bir butona basılmadığı takdirde *IR Remote receiver* komutu 0 değeri döndüreceğinden okunan değer ancak 0'dan faklı ise ekrana yazdırılmıştır.



Resim 283. Örnek Çıktı

Sağ ve sol ok tuşlarının kodu öğrenildikten sonra aşağıdaki resimde görülen kod Arduino'ya yüklenmelidir. Bu kod başlangıçta 3 numaralı pine takılı olan servo motorun açısını 90 dereceye ayarlar. Daha sonra kumandada yer alan sağ ok tuşuna basıldığında servo motorunun açısını 5 derece artırır. Sol tuşa basılınca ise işlemin tersi yapılır ve açı 5 derece azaltılır.



Resim 284. Örnek Program

Kod yüklendikten sonra öğrencilerin düzeneklerini test etmeleri istenir. Etkinliği erken bitiren öğrencilerden 1 tuşuna basılınca az, 2 tuşuna basılınca orta, 3 tuşuna basılınca ise çok yem dökülecek durumları kodlamaları istenir. 0 tuşunun ise kapalı duruma getirmek için kullanılması gerektiği söylenir. Burada öğrencilerin servo motorun dönüş açılarını kontrol etmeleri gerekecektir. Kapak tamamen kapalı olduğunda ve tamamen açık olduğunda servo motorun kaç derecede olduğuna bakılmalıdır. Tamamen açık ve tamamen kapalı olduğu durumlar arasındaki açı farkı alınır. Bu farkın üçte biri 1 tuşuna, üçte ikisi ise 2 tuşuna ve tamamı 3 tuşuna basıldığında servo motorun dönüşü için ayarlanmalıdır. Örneğin tamamen kapalı açısı 0 derece ve tamamen açık açısı ise 90 derece olduğu varsayılsın. Bu durumda 1 tuşuna basıldığında 30 derece, 2 tuşuna basıldığında 60 dereceye dönmelidir. 3 tuşuna basıldığında 90 ve 0 tuşuna basıldığında tamamen kapatmak için 0 dereceye dönmelidir.

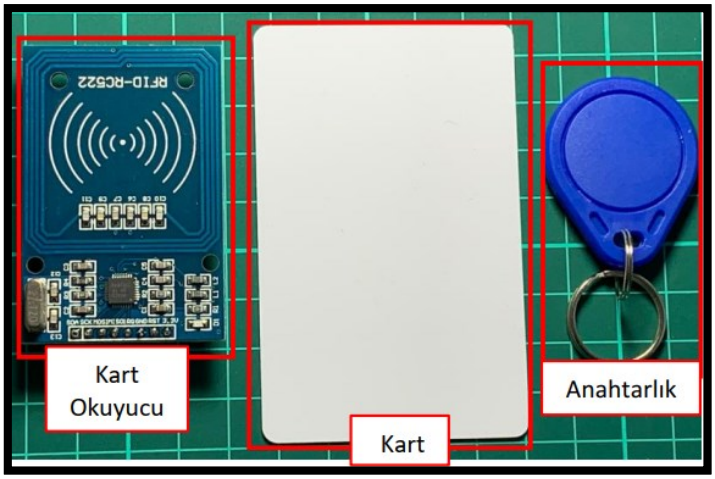
1.2 Gözle ve Uygula- Kaç Kişi Geldi?

Malzeme Listesi:

Arduino Uno
Breadboard
RC522 RFID NFC kiti
Direnç (1 adet 330 ohm, 3 adet 1 kohm)
1 adet led
ESP 8266 wifi modülü
100 µF kondansatör

Bu etkinlikteki amaç RC522 RFID NFC kiti ile bir yere giriş yapan kişilerin sayısının ve giriş zamanlarının tutulduğu ve bu bilgilerin internete bağlı uzak bir bilgisayardan kontrol edildiği internet nesneleri (Internet of Things- IOT) uygulaması yapmaktır.

RC522 RFID NFC kiti günlük yaşamda sıklıkla karşımıza çıkan ve genelde bir yere giriş-çıkışların takibi ve kontrolü için kullanılan elektronik karttır. Kit resimde görüldüğü gibi bir kart okuyucu ve bu okuyucunun okuyabileceği kartlar ve anahtarlıklardan oluşur. Her bir kart ya da anahtarlık içerisinde genellikle kendine özgü bir kod barındırır. Kart/anahtarlık, bağlantıları yapılmış okuyucuya yeterince yaklaştırıldığında okuyucu kartın içerisindeki bu kodu okuyabilir. Bu nedenle kişilere verilmiş ve içerisinde orijinal bir kod barındıran anahtarlıklar ve kartlar ile bir yere giren kişilerin kim olduğu, giriş izinlerinin olup olmadığı ya da kaç kişinin girdiği gibi veriler bu kart aracılığı ile yönetilebilir.



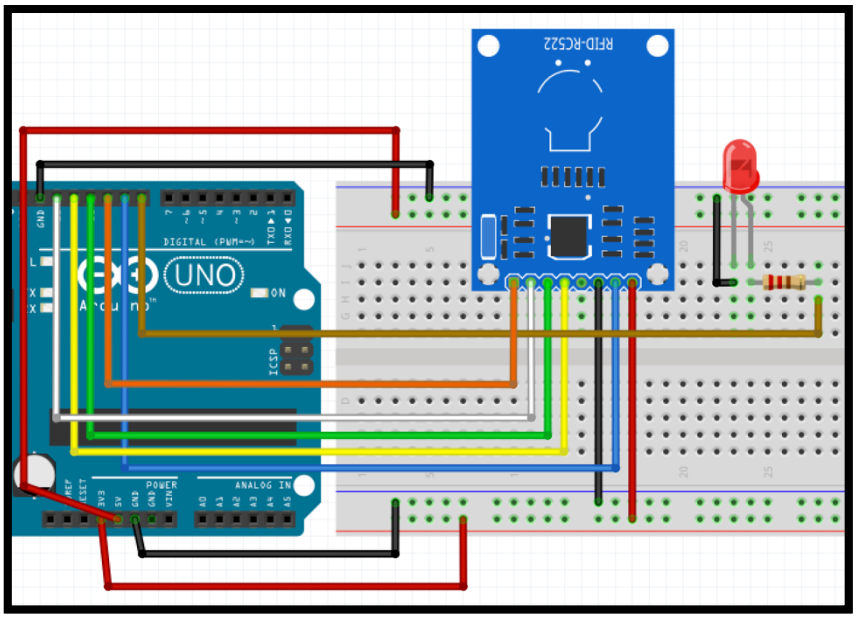
Resim 285. RC522 RFID NFC Kiti

RC522 RFID NFC kart okuyucunun Arduino ile çalışabilmesi için aşağıdaki resimde verilen kart pinlerinin Arduino'nun ilgili pinlerine takılması gerekir. Önceki işlemlerden farklı olarak RFID NFC kart okuyucu 5V ile değil 3.3V ile çalışmaktadır. Bu nedenle güç bağlantısı yapılırken 5V verilmediğinden emin olunmalıdır.

Resim 286. RC522 RFID NFC Kiti Bağlantı Uçları

Resim üzerinde verilen uçlardan "BOŞ" etiketi ile işaretlenen ucun devre kurulumu esnasında boş bırakılması gerekmektedir. Öğrencilere RC522 RFID NFC kart okuyucunun çalışma prensibi aktarılır. Bu elemanın olası kullanım alanlarına örnekler verildikten sonra sınıfa başka hangi yerlerde ve hangi

amaçlarla RC522 RFID NFC kart okuyucunun kullanılabileceği sorusu yöneltilir ve gelen cevaplar sınıfça tartışılır. Daha sonra aşağıdaki devre kurularak RC522 RFID NFC kart okuyucunun Arduino bağlantısı yapılır. Ayrıca resim görüldüğü gibi kart okuma işleminin yapıldığı sırada görsel uyarı vermesi amacı ile 8 numaralı dijital pine 220 Ohm direnç üzerinden LED bağlanır.



Resim 287. Örnek Devre Şeması

ArduinoBlocks ile RC522 RFID NFC kart okuyucuyu programlamak için öncelikle Setup bloğu içerisinde kart okuyucunun tanıtılması gerekmektedir. Bunun için RFID menüsünün içerisinden aşağıda resmi verilen MFRC522 Init (SPI) bloğu seçilir ve Setup bloğu içerisine yerleştirir. Bağlantıları yapılmış kart ile iletişim kart üzerindeki CS ve RESET pinleri aracılığı ile gerçekleştirilir. Bu kod bloğu üzerinde CS pini 10, RESET pini ise 9 numaralı pin olarak tanımlanmıştır. Bu etkinlikte de aynı pinler kullanılacağından değişiklik yapılmasına gerek yoktur.

Resim 288. MFRC522 Init (SPI) Bloğu

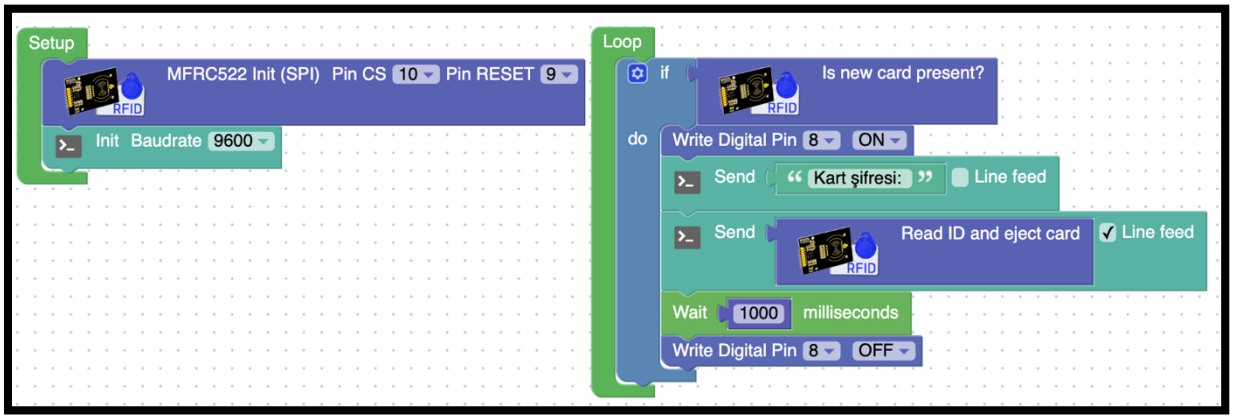
Kart okuyucunun "Setup" içerisinde tanıtılmasından sonra Loop bloğunun içerisinde kullanılmasını sağlayan iki kod bloğu bulunmaktadır. Bunlardan ilki aşağıdaki resimde görülen "*Is new card present?*" bloğudur. Bu blok kart okuyucuya bir kart ya da anahtarlık yaklaştırılınca "True/Doğru" bilgisi üretir. Mantıksal karşılaştırma yapan bloklar ile kullanılır. Örneğin "if" bloğuna doğrudan bağlanabilir. Ek bir karşılaştırma yapılmasına gerek yoktur.

Resim 289. "Is new card present?" Bloğu

Kart okuyucu ile çalışan ikinci kod bloğu ise aşağıdaki resimde görülen "*Read ID and eject card*" bloğudur. Bu blok iki işlemi aynı anda gerçekleştirir. Birinci görevi yaklaştırılan kartın ya da anahtarlığın içerisinde yer alan kodu okumaktır. İkincisi ise kısa okuma işlemi yapıldıktan sonra kısa bir süreliğine okumayı durdurmaktır. Eğer bu durdurma işlemi yapılmazsa okuma defalarca tekrar eder ve kodlama işlemi zorlaşır.

Resim 290. "Read ID and eject card" Bloğu

RC522 RFID NFC kart okuyucu ile ilgili blokların tanıtılmasından sonra kurulan devrede kart okuyucunun doğru çalışıp çalışmadığının test edilmesi gerekmektedir. Bu nedenle aşağıdaki resimde görülen kod yazılarak Arduino'ya yüklenir. Kod bloğunun amacı RC522 RFID NFC kart okuyucu ile anahtarlık ve kartın içerisindeki kodu okumak ve seri port ekranında bu kodları yazdırmaktır.

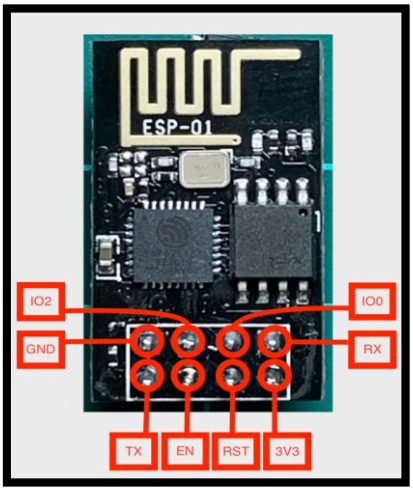


Resim 291. Kart Şifresi Belirleme Programı

Kod bloğunda görüldüğü gibi if komutunun yanına "*Is new card present?*" bloğu eklenmiş ve başka bir karşılaştırma yapılmamıştır. Kart yaklaştığında bu blok "True/doğru" çıktısı üretecek ve if bloğunun içindeki kodlar çalıştırılacaktır. Yani bir kart yaklaştığında 8 numaralı pine takılı led yanacak ve "Console" ekranına kartın şifresi gönderilecek ve daha sonra led sönecektir. Kodların ekranda çalıştırılmasından sonra sağ üstte yer alan "Console" butonu tıklanmalı ve açılan pencerede "Connect" butonuna tıklanarak bağlantı kurulmalıdır. Console ekranında kart ve anahtarlık okutulduktan sonra ekranda ilgili kartın ve anahtarlığın şifresinin yazıldığı görülmelidir. Tüm öğrencilerin setlerinde yer alan kartları anahtarlıkları okutmaları, "Console" ekranına yazdırmaları ve diğer grupların setlerindeki kart ve anahtarlıklar ile test etmeleri istenir. Daha sonra öğrencilere ekrana yazdırılan kodların birbirinden farklı olup olmadığı sorulur.

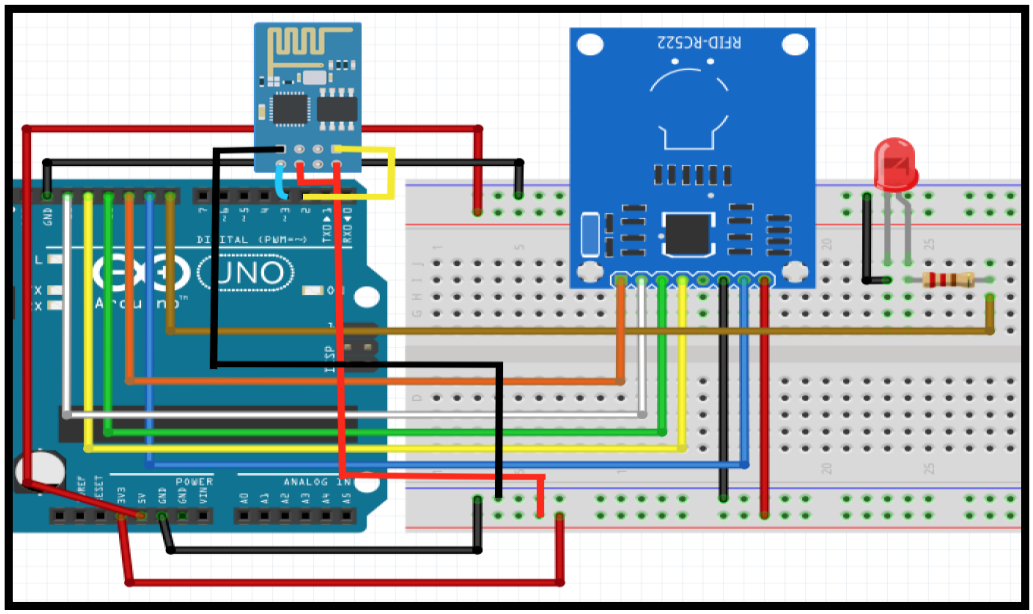
RC522 RFID NFC kart okuyucu kitinin doğru çalıştığından emin olduktan sonra sıradaki adım sayma işlemini İnternet üzerinden yaptırmaktır. Bu basit bir nesnelerin İnterneti (Internet of things-IOT) uygulamasıdır. IOT basitçe çevremizde yer alan cihazların bir kullanıcıya ihtiyaç duymadan ağ üzerinden veri gönderip alması veya birbirleri ile iletişime geçmesidir. Örneğin bir buzdolabının yumurta sayısı azaldığında iş yerinde olan ev sahibine mesaj göndermesi, dışarıdaki hava sıcaklığı belli bir derecenin üzerinde çıktığında ağ üzerinden komut gönderilerek ısıtma sisteminin kapatılması gibi işlemler IOT örnekleridir. Bu etkinlikte de kart okuyucu aracılığı ile bir yere giren kişi sayısı kablosuz internet (Wifi) üzerinden gönderilecek ve başka bir cihazda grafik olarak sunulacaktır. Arduino'un Wifi

ağına bağlanabilmesi için bir Wifi modülüne ihtiyaç vardır. Etkinlikte ESP 8266 01 modülü kullanılacaktır. Modülün görseli bağlantı uçlarının açıklaması ile birlikte aşağıdaki resimde sunulmuştur.



Resim 292. ESP 8266 01 Wifi Modülü

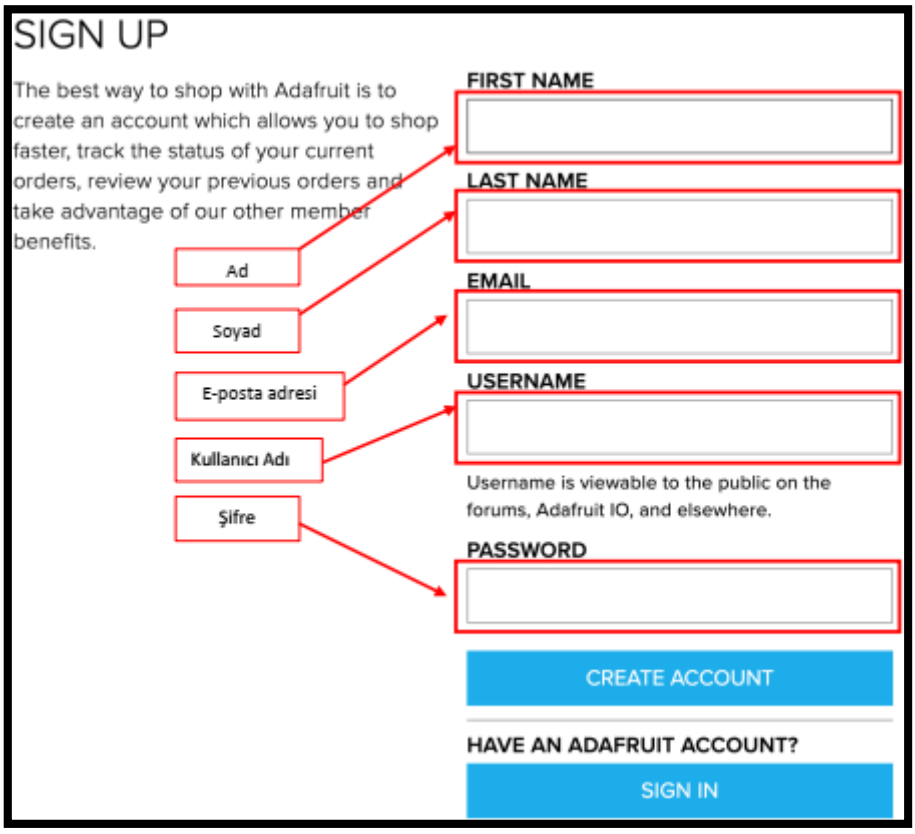
RC522 RFID NFC kart okuyucu gibi ESP 8266 wifi modülü de 3.3V ile çalışmaktadır. Resimde yer alan 3V3 etiketi 3.3 V bağlantısı, GND ucu da GND bağlantısı için kullanılır. RX ve TX uçları sırası ile veri alış ve verişi içindir. RST ucu modülü yeniden başlatmak için kullanılır. Bu bacak GND'ye bağlanırsa modül yeniden başlatılır. Elektronik Programlama ve Nesnelerin İnterneti dersindeki etkinlikler kapsamında EN ucu 3.3V bağlantısına takılır. Diğer uçlar ders kapsamındaki etkinliklerde kullanılmamaktadır. ESP 8266 modülü devreye aşağıdaki resimde olduğu gibi bağlanmalıdır.



Resim 293. Örnek Devre Şeması

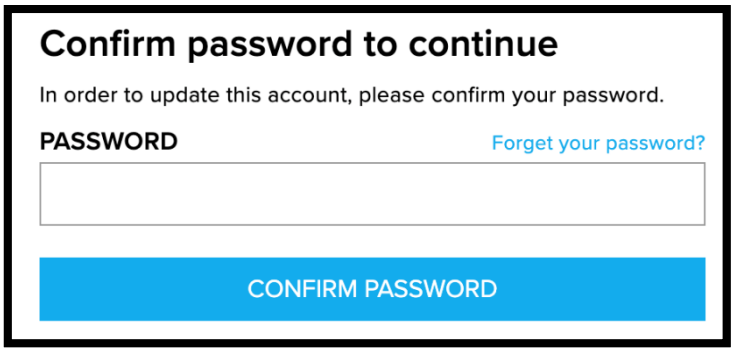
IOT kavramı farklı araçların bir ağ üzerinden birbiri ile iletişime geçmesi esasına dayanmaktadır. Bu etkinlikte kullanılacak ağ İnternet olacaktır. Bluetooth'un da ağ olarak kullanıldığı IOT etkinlikleri olabilmektedir. Kurulan devre ile iletişime geçmek için bir internet sunucusuna ihtiyaç vardır. Bu sunucu kullanıcı tarafından oluşturulabileceği gibi hazır olarak kullanıma açılmış sunucular da kullanılabilir. Bu etkinliklerde https://io.adafruit.com/ internet sitesinde ücretsiz olarak sunulan sunucu

hizmetinden yararlanılacaktır. Bu sunucudan yararlanmak için ilk olarak siteye kayıt yaptırarak üyelik olunması gerekmektedir. Üyelik için bir eposta adresi yeterli olacaktır. Aşağıda resimde adımları açıklanmış üyelik işlemlerini her gruptan en az bir öğrenci içim gerçekleştirilmelidir. İlk olarak "https://io.adafruit.com/" adresine girilir ve sağ üstte yer alan "Get Started For Free" bağlantısına tıklanır. Aşağıdaki resimde görülen sayfa açılacaktır.



Resim 294. AdaFruit Kullanıcı Kayıt İşlemleri

Ad, soyad, e-posta, kullanıcı adı ve şifre bilgileri belirlendikten ve ilgili alanlara yazıldıktan sonra "CREATE ACCOUNT" butonuna tıklanarak hesap oluşturulur. Bir sonraki ekranda "SAVE SETTINGS" butonu tıklanarak kayıt işlemleri tamamlanmış olur. Daha sonra ise aşağıdaki resimde yer alan kutucuğa belirlenen şifre girilir ve "CONFIRM PASSWORD" butonuna tıklanır.

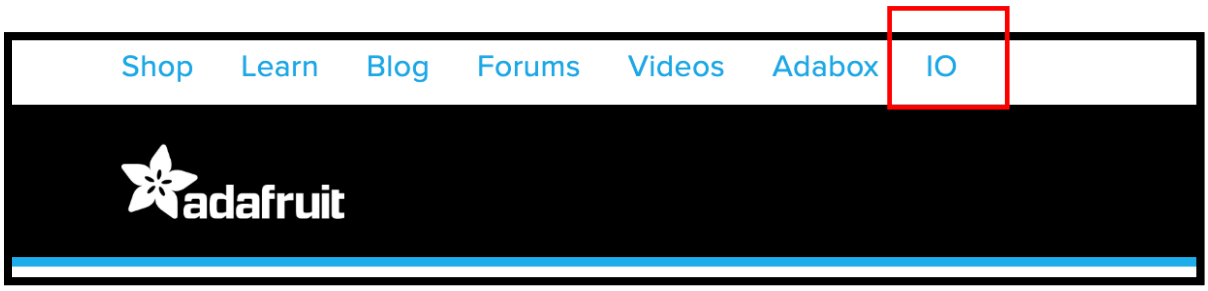


Resim 295. AdaFruit Kullanıcı Kayıt İşlemleri

Dikkat

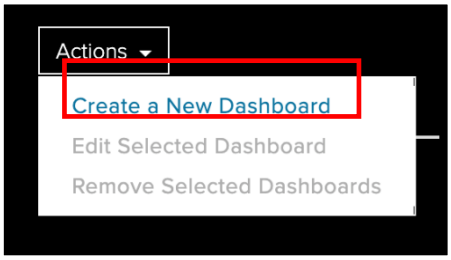
Öğrencilerin Adafruit sitesine üyelikleri ders öncesinde rehber öğretmen tarafından yapılmalıdır/yapılması sağlanmalıdır. Ders esnasında öğrenciler üyelik ile zaman kaybetmemelidir.

Bu işlemler tamamlandıktan sonra artık kurulan devre ile iletişime geçecek olan araçlar oluşturulacaktır. Bunun için ilk olarak aşağıdaki resimde görülen ekranın üstünde yer alan bağlantılardan "IO" yazılı bağlantıya tıklanır.



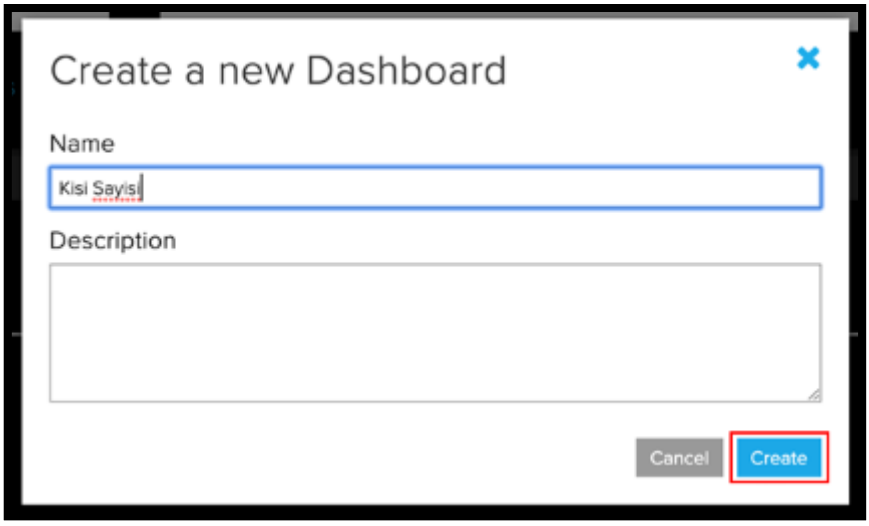
Resim 296 AdaFruit Girişi

Tıklandıktan sonra ilk olarak hazırlanacak veri gönderme ve alma araçlarının tutulacağı sayfanın oluşturulması gerekmektedir. Bu sayfalara "Dashboard" adı verilmektedir. İlerleyen süreçte farklı projeler için farklı sayfalar oluşturulabilir. Dashboard oluşturmak için aşağıdaki resimde görüldüğü gibi "Actions" yazılı açılır menü tıklanır ve menüden "Create a New Dashboard" seçeneği seçilir.



Resim 297. AdaFruit Yeni Dashboard Oluşturma

Açılan pencerede aşağıdaki resimde görüldüğü gibi "Name" kutucuğuna etkinliği tanımlayan bir isim girilir. Bu etkinlik için "Kisi Sayisi" ismi verilebilir. Türkçe karakter kullanılmamasına dikkat edilmelidir. Aynı pencerede yer alan "Descriptions" kutucuğu etkinlik ile ilgili bilgileri ve tanımlamaları kaydetmek içindir. Zorunlu bir alan değildir. Bu etkinlik kapsamında boş bırakılabilir. İsim girildikten sonra "Create" butonu tıklanır.

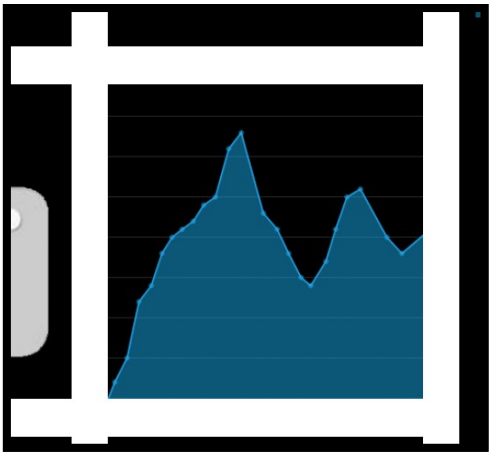


Resim 298. AdaFruit Yeni Dashboard Oluşturma

İşlem sonunda oluşturulan sayfa ekrandaki listeye eklenecektir. Listedeki "Kisi Sayisi" isimli sayfa bağlantısına tıklanır. Bu pencere etkinlik içerisinde oluşturulacak araçların yer alacağı sayfadır. Sayfaya araç eklemek için aşağıdaki resimde görüldüğü gibi sağ üstte yer alan "+" butonuna tıklanır.

Resim 299. AdaFruit Araç Ekleme

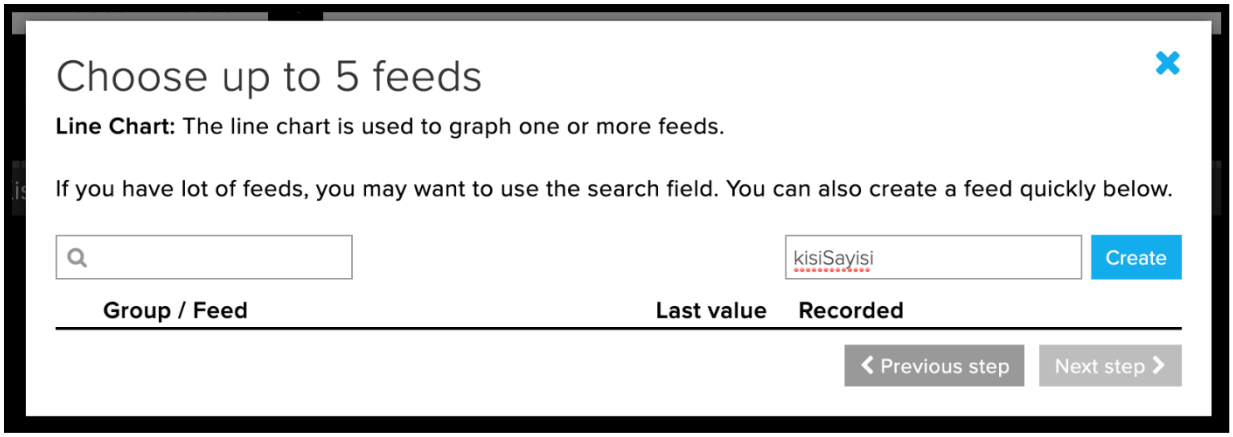
Açılan pencerede eklenebilecek araçlar listelenmektedir. Bu listeden yer alan araçlardan ve üzerine gelindiğinde "Line Chart" yazısı beliren araç kullanılacaktır. Bu araç devre ve yazılan kod aracılığıyla elde edilen kişi sayısını aşağıdaki resimde görüldüğü gibi grafik olarak yansıtacaktır.



Resim 300. AdaFruit Araç Ekleme

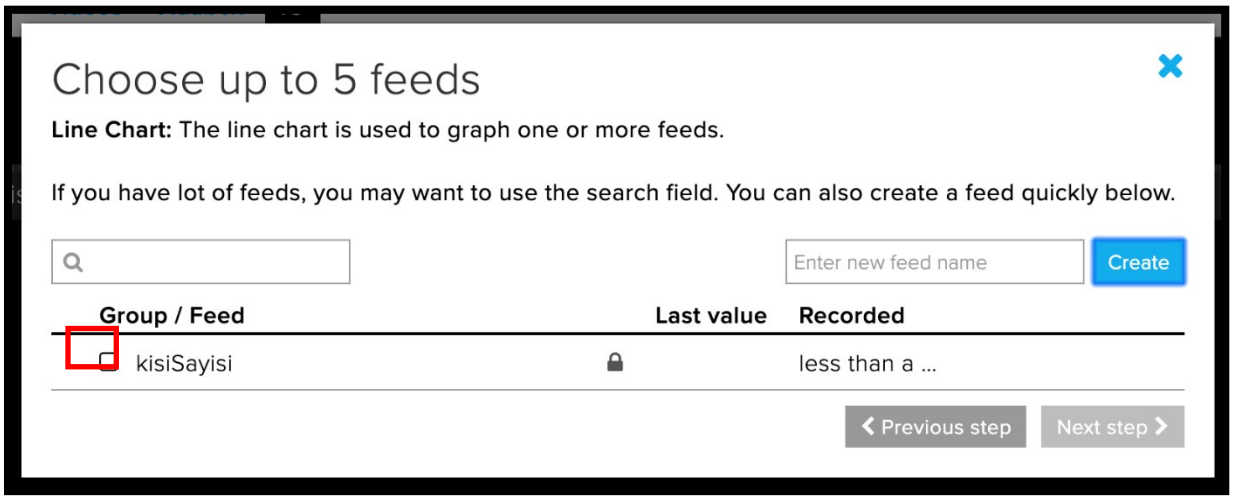
"Line Chart" aracı üzerine tıklanır. Aşağıdaki resimde görülen pencere açılacaktır. Bu pencerede oluşturulan aracın adı girilerek "Create" butonuna tıklanmalıdır. Araca verilen ad önemlidir. ArduinoBlocks ortamında yazılacak kod içerisinde bu isim yer alacaktır. O nedenle Türkçe karakter kullanılmadan isim oluşturulmalıdır. Bu etkinlik için "kisiSayisi" ismi kullanılabilir. Görselin sağ

tarafında yer alan ve içerisinde “Enter new feed name” yazan kutunun içerisine “kisiSayisi” yazılır ve “Create” butonuna tıklanır.



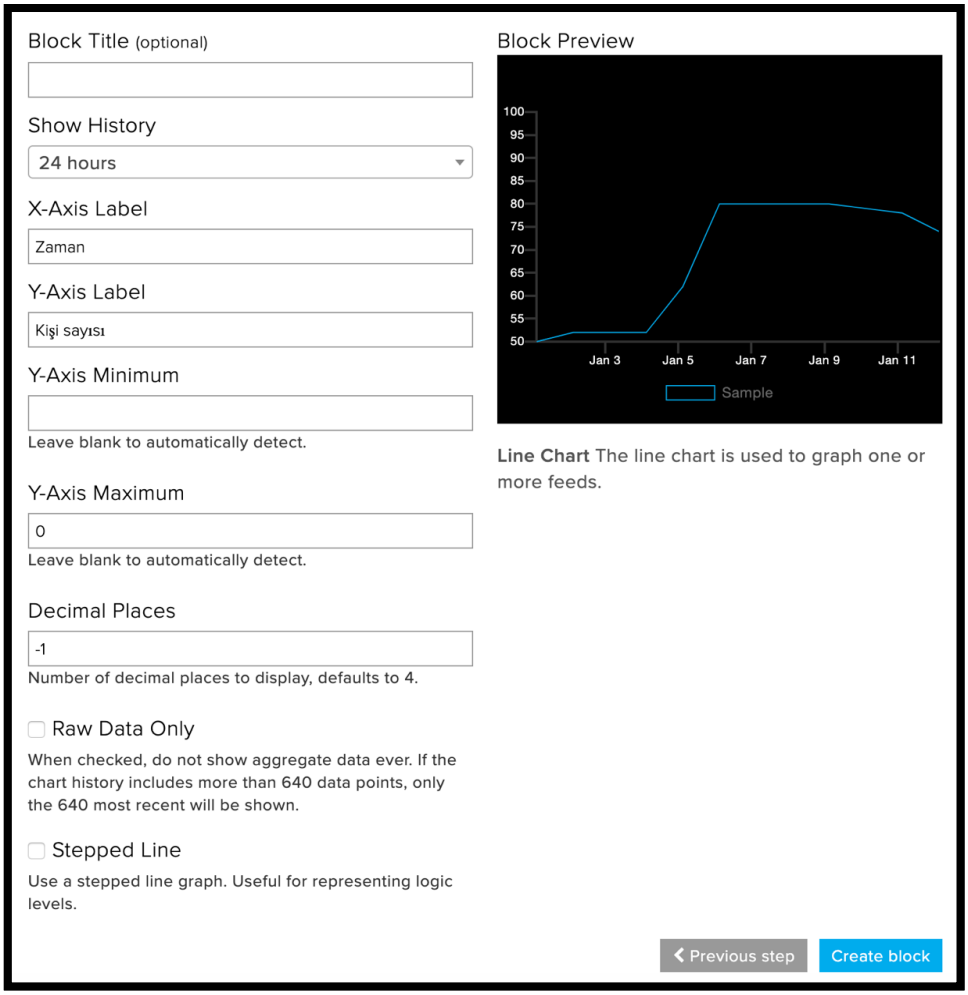
Resim 301. AdaFruit Araç Ekleme

Create butonuna tıklandıktan sonra aşağıdaki resimde görüldüğü gibi aynı pencerenin altında oluşturulan araç listelenir.



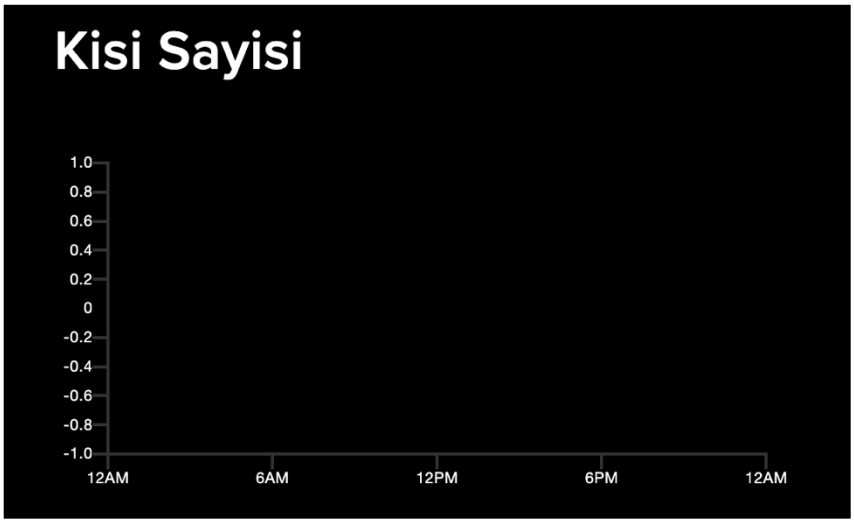
Resim 302. AdaFruit Araç Ekleme

Aracın ayarlarının yapılabilmesi için aracın isminin solunda bulunan kutucuk işaretlenir ve Next step butonuna tıklanır. İşlem sonucunda aşağıdaki gibi bir pencere elde edilir. Bilgiler resimdeki gibi doldurulur. “X-Axis Label” grafiğin yatay kısmının etiketidir. Bu kısım zaman bilgisini tutacaktır. “Y-Axis Label” ise kişi sayısı bilgisini verecektir. “Y-Axis” minimum kutusuna 0 girilmelidir. Bilgiler aşağıdaki resimdeki gibi doldurulduğunda “Create Block” butonuna tıklanır.



Resim 303. AdaFruit Araç Ayarı

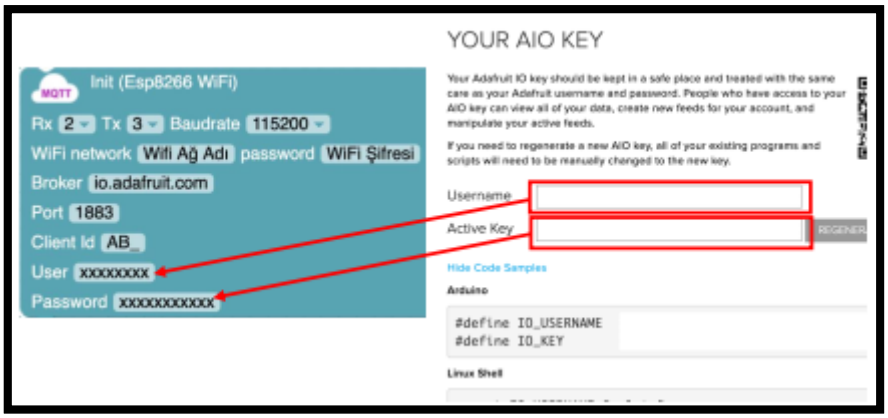
Bu işlemlerden sonra ekran aşağıdaki resimde görüldüğü gibi görünmelidir. Bu işlemden sonra ArduinoBlocks ile yazılan kodun ve eklenen grafik aracının iletişimini sağlamaktır.



Resim 304 AdaFruit Araç Çıktı Ekranı

İlk olarak ESP 8266 wifi modülünün bağlantı bilgileri girilmelidir. Bunu için MQTT blok başlığı içerisindeki *Init(Esp8266 WiFi)* bloğu Setup'ın içerisine yerleştirilir. Bu blok hem internet bağlantısı için hem de IOT etkinliğinin gerçekleştirildiği Adafruit bağlantsı için kullanılacaktır.

ESP 8366'nın Rx ve Tx uçlarının Arduino kartı üzerinde bağlı olduğu pinlerin bilgisi Rx ve Tx alanına girilir. Etkinlikte kurulan devrede Rx 2, Tx ise 3 numaralı pinlere takılıdır. Daha sonra *WiFi network* alanına atölyede var olan kablosuz ağın tam adı, *password* alanına ise ağın şifresi girilir. Bu bilgiler bilinmiyorsa yetkili personelden ders öncesi öğrenilmelidir. *Broker* alanına https://io.adafruit.com/ adresi girilir. *User* ve *Password* alanları Adafruit sitesinde bulunan ekranındaki bilgiler (sağdaki resim) ile doldurulacaktır. Bunun için Adafruit ekranının sağ üstünde yer alan AIO Key bağlantısı tıklanır. Buradaki bilgilerden *Username* bilgisi bloktaki User, *Active Key* bilgisi ise bloğun altındaki *Password* alanına girilir. Bilgi girişleri aşağıdaki resimde özetlenmiştir.

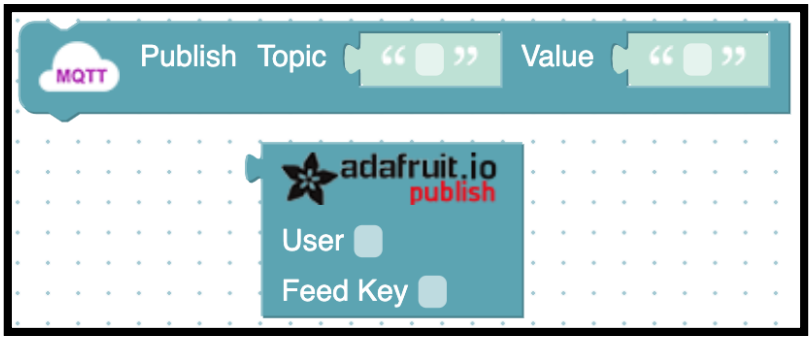


Resim 305 Wifi Ayarları ve AdaFruit Bağlantısı

Dikkat

Resimde bilgiler sembolik olarak girilmiştir. Bu bilgilerdeki en ufak yanlışlık hem internete bağlantıyı hem de Adafruit ile iletişimi engelleyecektir. Bu nedenle özellikle WiFi adı ve şifresinin her öğrenci tarafından görülebilecek bir yere yazılması hataları azaltacaktır.

Yukarıdaki blok doğru şekilde doldurulup Setup bloğunun içine yerleştirildiğinde hem internet bağlantısı kurulmuş hem de Adafruit ile ilgili bağlantı da hazır hale getirilmiş olacaktır. Bundan sonraki aşama Arduino üzerinden veri gönderme işlemidir. Bunun için MQTT başlığı içerisinden iki farklı blok beraber kullanılacaktır. Blokların resimleri aşağıda sunulmuştur.



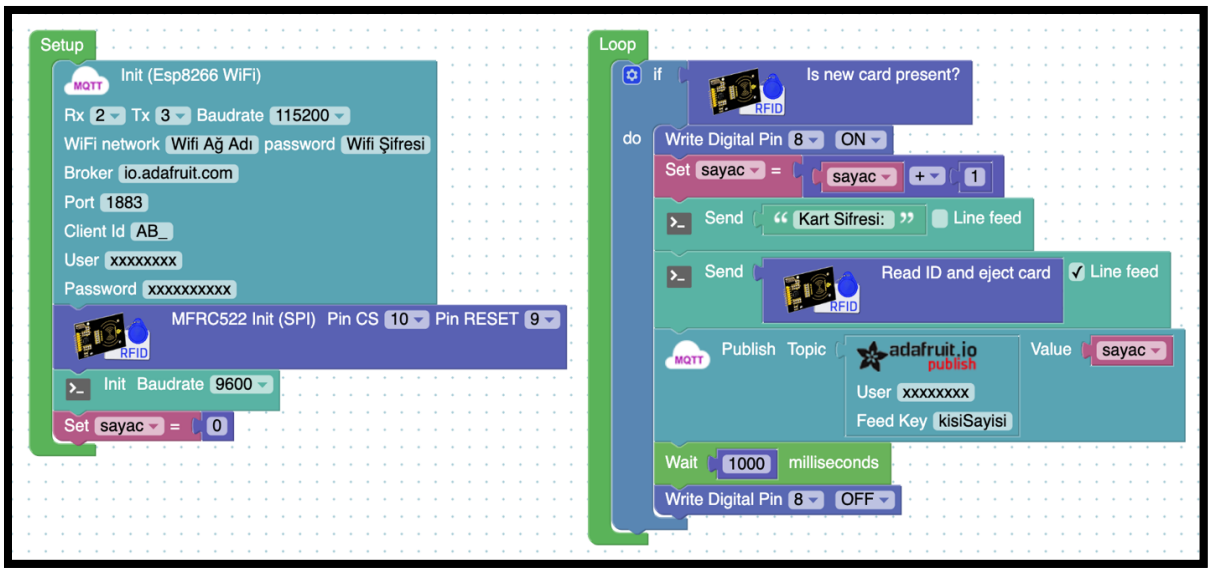
Resim 306. AdaFruit Bilgi Gönderme ve Alma

Publish Topic bloğundaki ilk boşluk bilginin gönderileceği yer, ikinci boşluk ise gönderilecek bilgi içindir. Adafruit'e bilgi göndermek için ikinci blokla birlikte kullanılmalıdır. İki blok aşağıdaki resimde

görüldüğü gibi birleştirilmeli ve "User" kısmına bir önceki bağlantı kurma işlemindeki kullanıcı adı girilmeli, "Feed Key" kısmına ise Adafruit'te oluşturulan aracın adı girilmelidir.

Resim 307. AdaFruit Bilgi Gönderme ve Alma

Bu blok çalıştırıldığında "Value" alanına girilen değer "Adafruit" ekranında oluşturulan ve "kisiSayisi" adı verilmiş olan grafiğe gönderilecektir. Gönderilen değer ve gönderilme zamanı grafik üzerinden okunabilecektir. Kart okuma ve seri port ekranına yazdırma işlemlerinin yapıldığı kod bloğu aşağıdaki resimde görüldüğü gibi güncellenerek kişi sayısı hem seri port ekranına hem de Adafruit üzerinde oluşturulan grafik aracına gönderilir.



Resim 308. Örnek Program

Resimde görüldüğü gibi Setup bloğu içerisinde ağ bağlantısı bilgileri girilmiş, RFID kart okuyucu ve seri port kullanımı başlatılmış ve son olarak da kişi sayısının tutulacağı sayaç değişkeni tanımlanarak başlangıç değeri olarak 0 atanmıştır. Loop bloğunun içerisinde ise kart/anahtarlık okunup okunmadığını kontrol eden if bloğunun içerisinde okuma işleminin başladığını göstermek için 8 numaralı pine bağlı led yakılmış, sayaç değişkeni arttırılmış ve okunan kart şifresi seri porta gönderilmiştir. Devamında ise sayac içerisindeki değer Adafruit'e gönderilmiş ve işlemin tamamlandığını ifade etmek için led bir saniye sonra söndürülmüştür. Ledin sönmesi aynı zamanda yeni kart okuma işleminin yapılabileceği anlamına gelmektedir.

Dikkat

Ağ bağlantısındaki sorunlar ve zaman zaman ESP 8266 Wifi modülündeki sıkıntılar nedeniyle okuma işlemlerinde aksaklıklar meydana gelebilmektedir. Böyle durumlarda modül doğru çalışana kadar beklenmelidir.

Dikkat

ESP 8266 WiFi modülü bağlı Arduino'ya kod yüklemesi yapıldığında Wifi modülünün ağa bağlanması ve veri transferine başlaması belirli bir süre almaktadır. Bu süre devreden devreye değişebilmektedir. Bu nedenle istenilen çıktıyı alamayan öğrencilerin en az iki dakika beklediklerinden emin olunmalıdır.

Öğrencilerden kodu yazıp Arduino'ya yükledikten sonra seri port ekranından okuttukları kartların şifrelerini kontrol etmeleri istenir. Daha sonra seri porttaki okuma sayısının Adafruit'te oluşturulan grafik aracını nasıl etkilediği kontrol edilir. Grafiklerde uzun süre değişiklik olmaması ya kodda ya da bağlantıda sorun olduğu anlamına gelebilir. Böyle durumlarda öğrenci devreleri ve kodları kontrol edilmelidir. Öğrenciler doğru çıktıları elde ettikten sonra rehber öğretmen grafiği öğrenciler için yorumlar. Ardından buna benzer başka ne tür bilgilerin Arduino üzerinden Adafruit'e gönderilebileceği konusu sınıfça tartışılır.

Dikkat

Beşinci ve altıncı sınıf öğrencileri grafik çizimini ve yorumlanmasını bilmemektedir. Rehber öğretmen grafiğin yorumlanmasını öğrencilere bırakmamalıdır. Basit bir şekilde grafikte Arduino karttan gelen sayısal bilginin görsel olarak ifade edildiği vurgulanarak anlatılmalıdır.

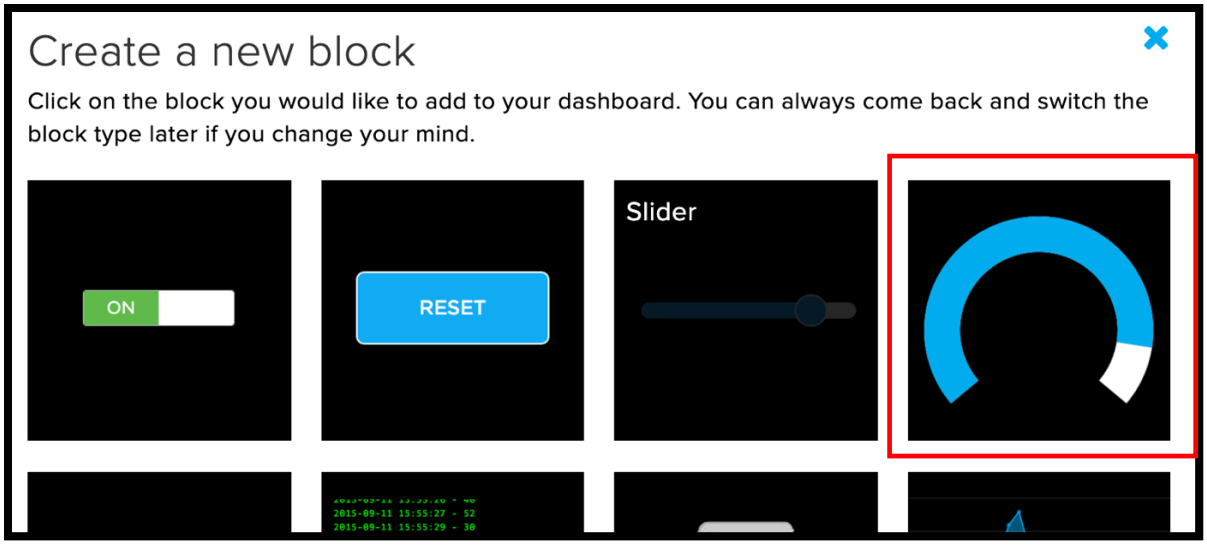
1.3 Gözle ve Uygula- Ev Gözlem Monitörü

Malzeme Listesi:

Arduino Uno
Breadboard
ESP8266 WiFi modülü
Sıcaklık ve Nem sensörü
RC522 RFID NFC kart okuyucu

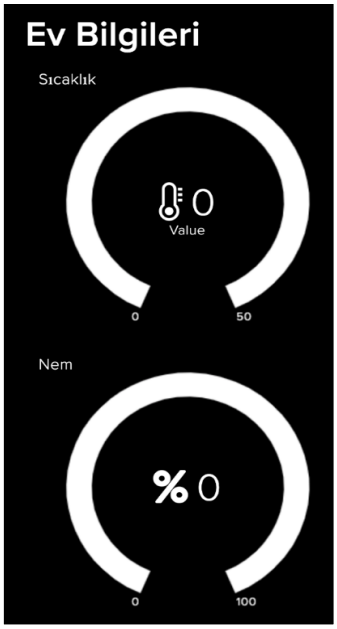
Bu etkinlikteki amaç dört kişilik bir ailenin yaşadığı bir ev ile ilgili bilgilerin internet üzerinden farklı bir noktaya gönderilmesi ve gözlemlenmesi için bir IOT uygulaması yapmaktır. Evin sıcaklığı, nem oranı ve evdeki aile üyelerinin kimler olduğu Adafruit üzerinde gösterilecektir.

Etkinlik için öncelikle Adafruit içerisinde yeni bir sayfa yani Dashboard oluşturulması gerekmektedir. Bunun için https://io.adafruit.com/ sitesine giriş yaptıktan sonra üstte yer alan Dashboard bağlantısı tıklanır ve açılan sayfadan "View all" bağlantısı tıklanır. Açılan sayfada varsa daha önce oluşturulmuş sayfalar listelenecektir. Bir önceki etkinlikte oluşturulan "Kisi Sayisi" isimli sayfa da burada listelenmiş olmalıdır. Yeni bir sayfa için bir önceki etkinlikte takip edilen aşamalar kullanılarak "Ev Bilgileri" isimli bir sayfa/dashboard oluşturulur. Oluşturulan sayfaya giriş yapılır. Sayfa içerisinde ilk önce sıcaklık ve nem bilgilerinin gösterileceği araçlar eklenmelidir. Bunun için "+" butonu tıklanır ve "Gauge" isimli araç seçilir ve Sicaklik ismi verilerek Create butonu tıklanır. Yeni araç ekleme resmi aşağıda verilmiştir.



Resim 309. AdaFruit Yeni Araç Ekleme

Araç listesinde "Sicaklik" ismi belirdikten sonra yanındaki kutu işaretliyken "Next step" butonu tıklanarak araç ayar sayfasına girilir. "Block Title" kısmına "Sıcaklık", "Gauge Max Value" kısmına ise 50 yazılır. "Gague Label" kutusuna da "derece" ifadesi yazılır. İstenirse Sayfanın altında yer alan "Show icon" kutusu işaretlenerek sıcaklık değerini sembolize eden bir ikon yerleştirilebilir. Kutu işaretlendikten sonra altında yer alan açılır menüden "temperature" figürü seçilerek bu işlem gerçekleştirilebilir. Tüm ayarlar tamamlandıktan sonra "Create Block" butonu tıklanır. Aynı işlemler Nem bilgisi için de tekrarlanır. Farklı olarak Gauge isminin ve "Block Title" bilgilerinin nemölçere göre düzenlenmesi ve "Gauge Max Value" değerinin de 100 olması gerekmektedir. "Gauge Label" kutusu boşaltılarak "Show icon" seçeneği tıklanır ve açılır menüden "percent" seçeneği seçilir. "Create Block" butonu tıklanır. Bu işlem sonunda ekran görüntüsü aşağıdaki resimde görülen gibi olmalıdır.

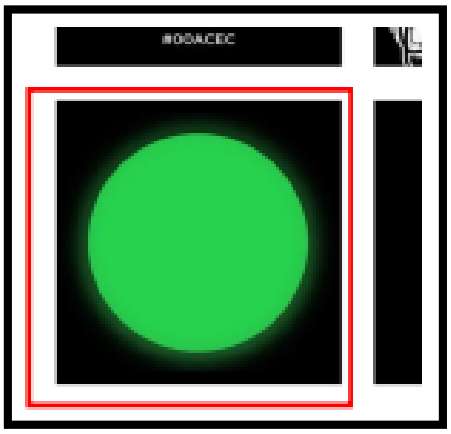


Resim 310. AdaFruit Ekran Alıntısı

Eve giren ve çıkan kişilerin bilgilerini görüntülemek için ise "Indicator" aracı kullanılacaktır. Bu araç kendi içerisinde bir karşılaştırma yaptıktan sonra karşılaştırmanın doğru olması durumunda belirlenen

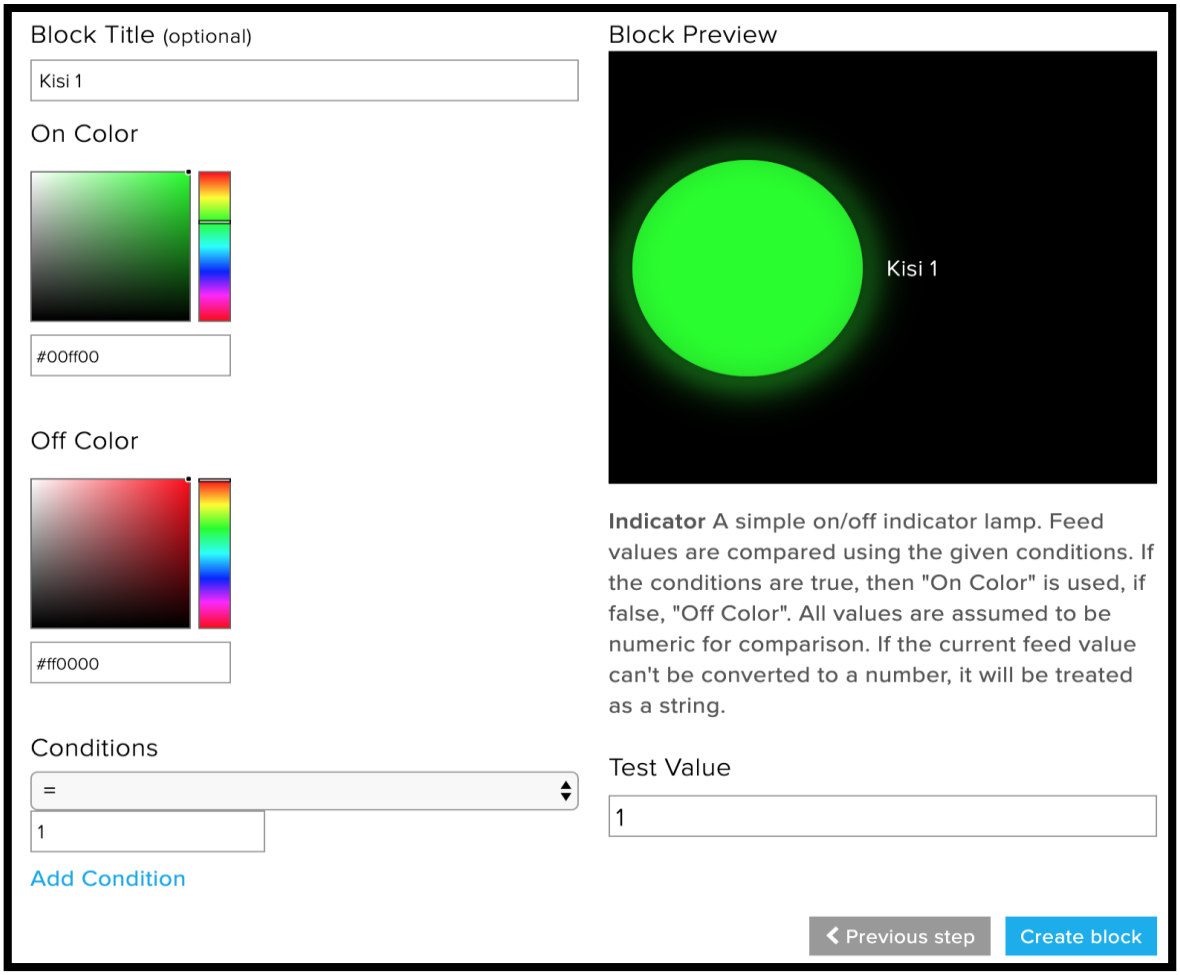
bir rengi, yanlış olması durumunda ise başka bir rengi görüntülemektedir. Etkinlikte eve giriş izni olan aile üyelerinin 4 kişi olduğu varsayılırsa bu araçtan dört adet eklenmeli ve araç isimlendirmeleri de buna göre yapılmalıdır. Burada aile üyeleri Kisi-1, Kisi-2, Kisi-3, Kisi-4 şeklinde ifade edilecektir. Ancak öğrenciler isimlendirmede Türkçe karakter kullanmamak kaydıyla kendi aile üyelerinin isimlerini kullanabilirler. Aşağıdaki resimde bir adet "Indicator" aracının eklenişi gösterilmektedir. Bu işlemin tüm aile üyeleri için tekrarlanması gerekmektedir.

Dashboard ekranında sağ üste yer alan "+" butonu tıklanır ve açılan pencereden "Indicator" aracı seçilir.



Resim 311. AdaFruit Araç Ekleme

Gelen pencerede Kisi-1 ismi verilerek "Create" butonuna tıklanır. Açılan yeni pencerede "Block Title" kısmına "Kisi 1" yazılır. Bu ekranda gösterilecek aracın başlığı olacaktır. "On Color" alanından kişi evde olduğunda gösterilecek renk belirlenir. Ön tanımlı olarak yeşil renk seçilidir. Bu kısımda diğer eklenecek "Indicator" nesneleri ile tutarlılığı sağlamak adına değişiklik yapılmaması daha uygun olacaktır. Aynı durum "Off Color" alanı için de geçerlidir. Son olarak yapılacak karşılaştırmanın bilgileri girilir. Arduino aracılığı ile gönderilecek bilgi bu karşılaştırmaya sokularak doğru renk belirlenecektir. Karşılaştırma sayısal veriler ile yapılmaktadır. Bu nedenle kişi evde ise "1", değil ise "0" bilgisi gönderilecektir. Dolayısıyla "Conditions" menüsünden "=" seçilir alttaki kutucuğa "1" yazılır. Bu durumda 1 değeri gönderildiğinde araç yeşil, başka bir değer gönderildiğinde ise kırmızı yanacaktır. Araç ekleme resmi aşağıda görülmektedir.

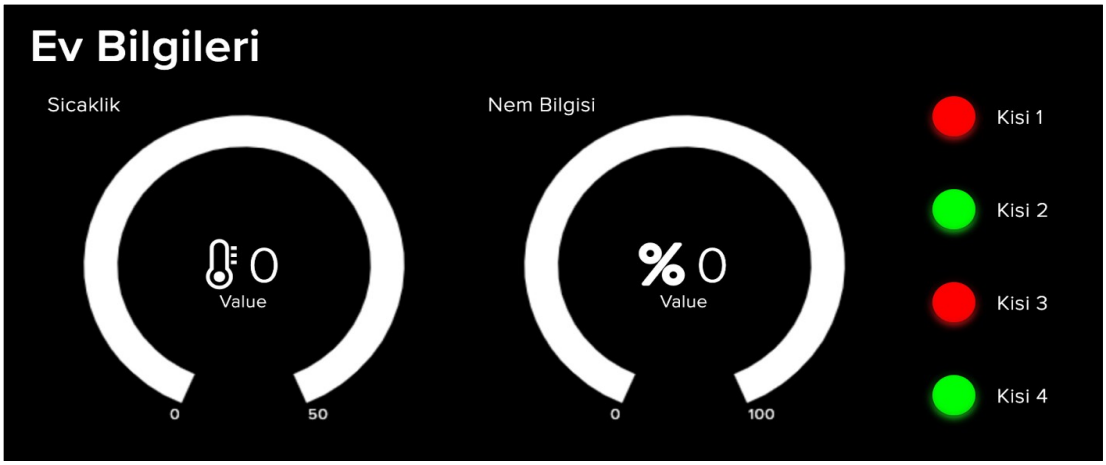


Resim 312. AdaFruit Araç Ekleme

Araçlar eklendikten sonra hepsinin alt alta sıralandığı görülecektir. Bu durum kullanımı zorlaştıracağından araçların yerleri düzenlenmelidir. Bunun için ekranın sağ üstünde yer alan butonlardan ayar butonu tıklanır. Ayar butonu tıklandıktan sonra resimde görüldüğü gibi araçlar istenildiği gibi sürüklenip uygun bir yere konumlandırılabilir.

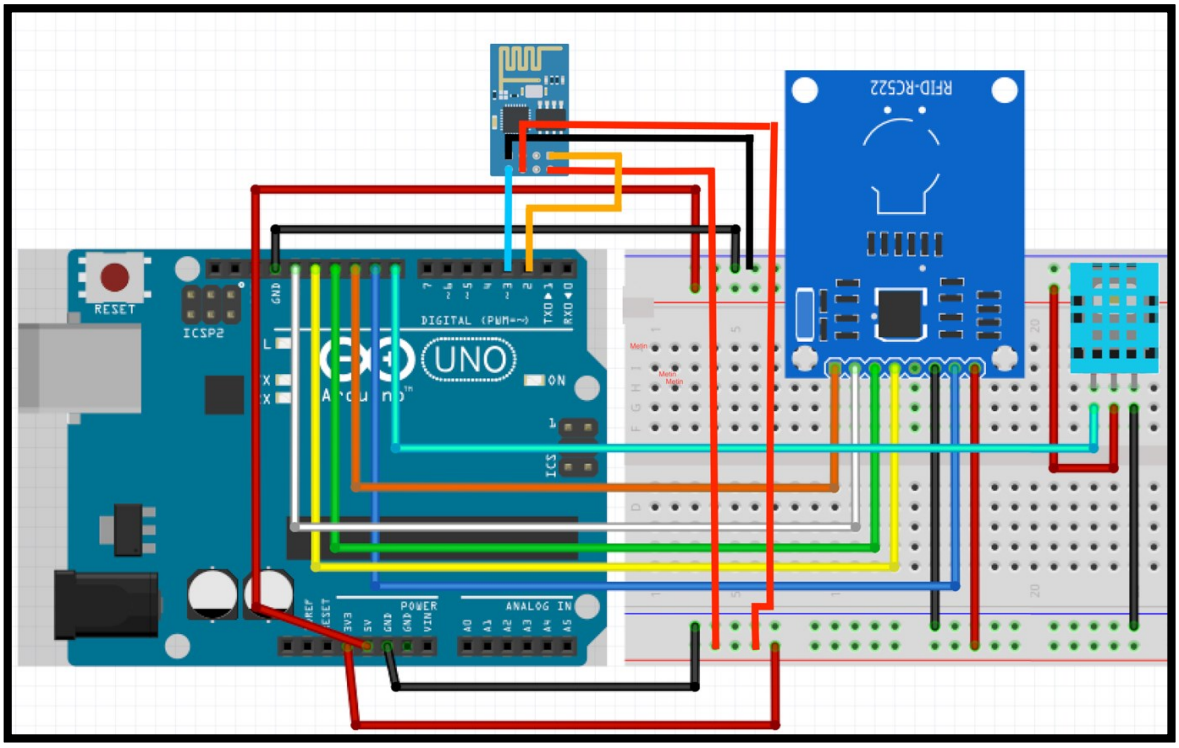
Resim 313. AdaFruit Araç Ayarı

Örnek bir Dashboard aşağıdaki resimde görüldüğü gibi olmalıdır.



Resim 314. Örnek Dashboard

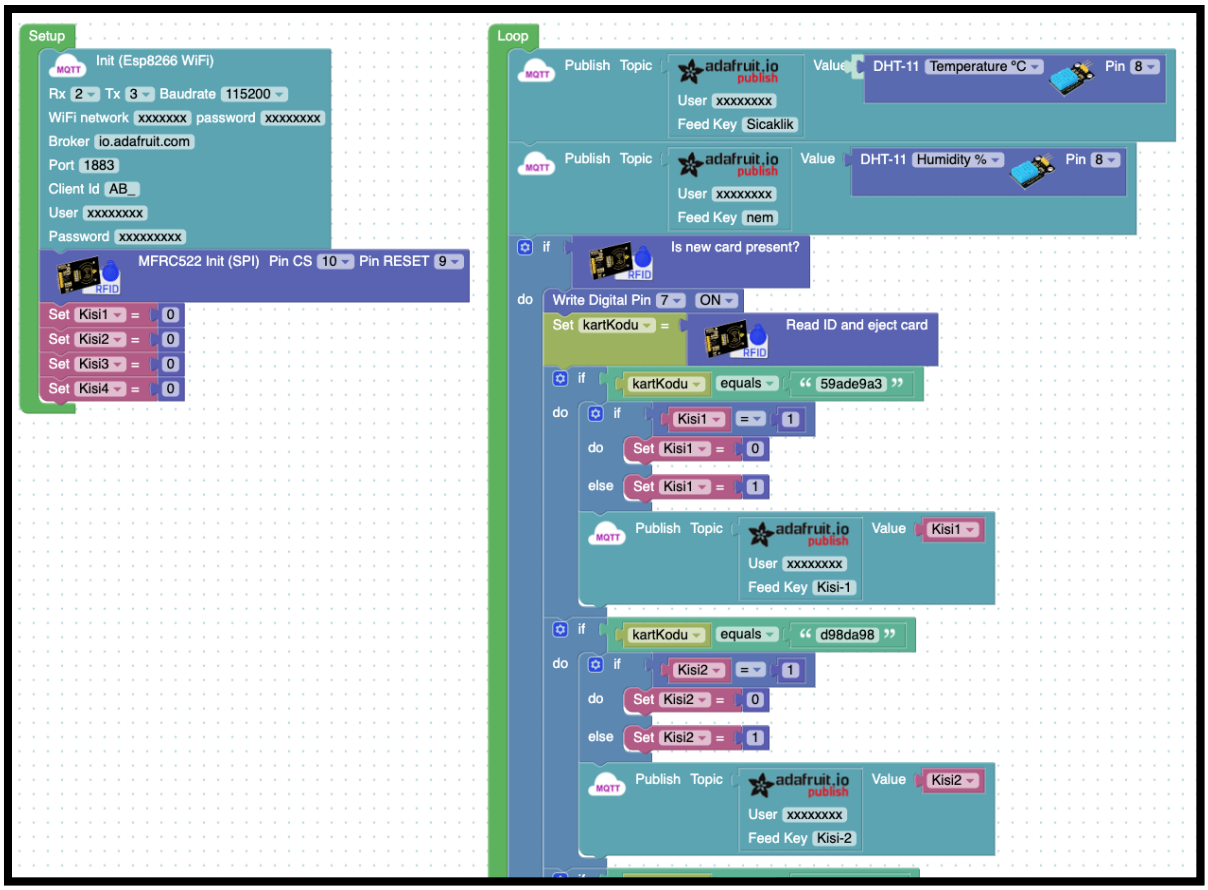
Adafruit araçlarının yerleştirilmesi ve Dasboard'a son halinin verilmesinden sonra devre kurulumuna geçilmelidir. Devre kurulumu aşağıdaki resimde görüldüğü gibi olmalıdır.



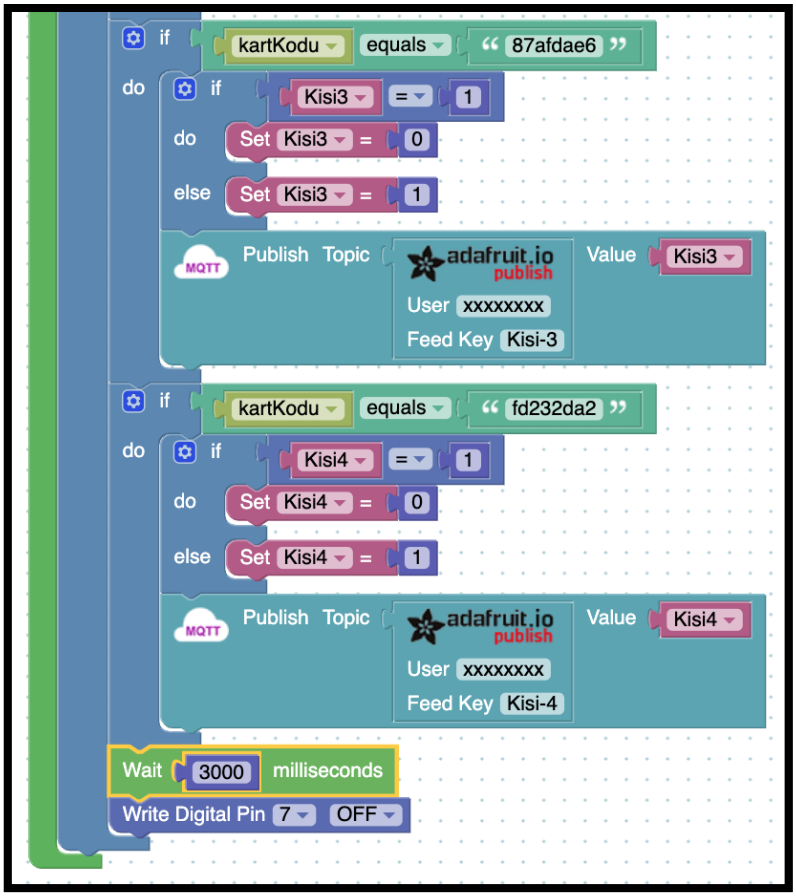
Resim 315. Örnek Devre Şeması

Devrede kart okuyucu ve Wifi modülü bağlantıları önceki etkinlikteki gibi, DTH11 ısı ve nem sensörü de 8 numaralı dijital pine bağlanarak devre kurulur. Breadboard'un alt kısmı 5V, üst kısmı ise 3.3V olacak görselde bağlantıların yapıldığına dikkat edilmelidir. Devre kurulduktan sonra kodların yazımına geçilmelidir. Ancak kodlamanın yapılabilmesi için anahtarlık ve kart şifrelerine ihtiyaç vardır. Bu nedenle "Kaç kişi gelmiş?" etkinliğinde yer alan ve kartın çalışıp çalışmadığının kontrolü için kullanılan kod kullanılır ve her grup kartlarının ve anahtarlıklarının kodlarını not eder. Ayrıca gruplarda bir kart ve bir anahtarlık olduğu için yan yana olan iki grup kart ve anahtarlık şifrelerini birbirleri ile paylaşarak kodlama esnasında kullanırlar. Test aşamasında birbirleri ile kartları ve anahtarlıklarını değiş tokuş

ederler. Bu işlem gerçekleştirildikten sonra aşağıdaki kod yazılır ve Arduino'ya yüklenir. Kart okuma işleminin gerçekleşip gerçekleşmediğini kontrol etmek için dijital pinlerden birine led takılarak kart okuma işlemi başladığında ON, bittiğinde OFF konumuna getirilebilir. Aşağıdaki resimde görülen kod, led takılmış durum için verilmiştir.

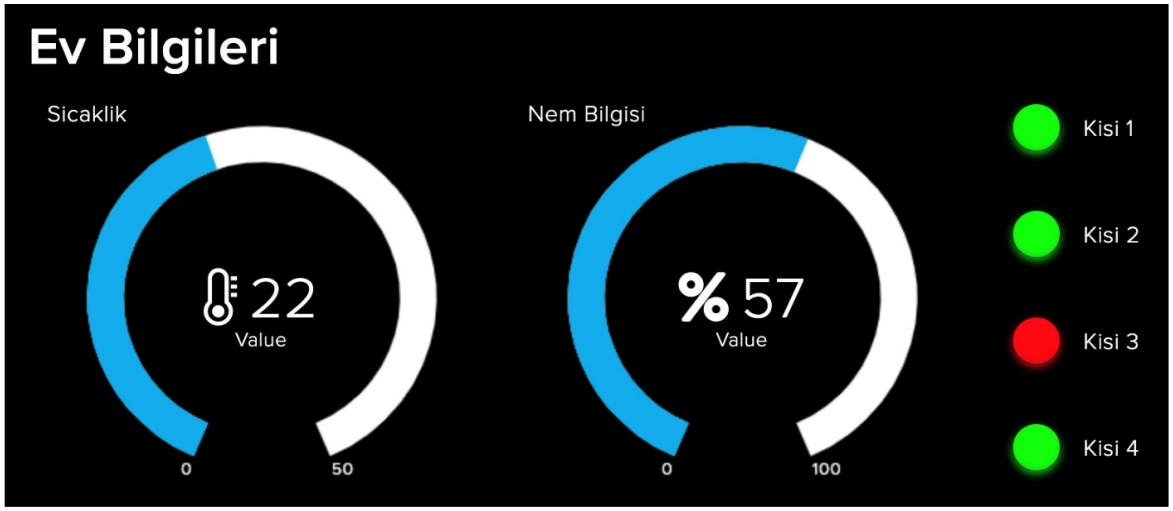


Resim 316. Örnek Program



Resim 317. Örnek Program (Devam)

Setup bloğunun içinde WiFi bağlantı bilgilerinin girişi yapılır. Devamında her bir kişi için oluşturulan değişkenlere 0 değeri atanır. Bu değer kart okuma yapıldığında işlemin eve giriş olarak tanımlanması içindir. Loop bloğunun içerisinde ilk olarak Adafruit ile oluşturulan Sıcaklık ve Nem araçlarına DTH 11'de okunan değerler gönderilir. Daha sonra kart okuma işlemi gerçekleştirilir. Evdeki 4 kişinin kendisine ait bir kartı olduğu varsayıldığı için 4 adet if bloğu kullanılmıştır. Okunan kart hangi kişiye aitse o if bloğu çalışır. Kişi eve girdiğinde ilgili değişken 1 duruma getirilir ve Adafruit'e 1 değeri gönderilir. Kişi dışarı çıkıyorsa ilgili değişken 0 yapılır ve Adafruit'e 0 değeri gönderilir. Adafruit'de kişinin evde olup olmama durumuna göre ilgili kişinin aracı yeşil veya kırmızı renge döner. Aşağıdaki resimde sıcaklık ve nem sensöründen okunan değerlerin sıcaklık için 22, nem için ise 57 olduğu görülmektedir. Kişi 1, Kişi 2 ve Kişi 4'ün evde, Kişi 3'ün ise ev dışında olduğu bilgisi renkler ile belirtilmektedir.



Resim 318. Dashboard Örnek Çıktısı

2. TASARLA ve ÜRET

2.1 Tasarla-Su doluluk göstergesi

Bu etkinlikte amaç farklı işlerde kullanılan su tanklarının ya da su haznelerinin doluluk seviyelerinin uzaktan gözlemlenebildiği bir devre ve IOT ekranı tasarlamaktır. Öğrenciler su seviyesini ölçecekleri durumu kendileri belirlemelidir. Her bir grup su seviyesi ölçümünü farklı amaçla tasarlayabilir.

Su doluluk göstergesi etkinliğini tamamlamak için öğrencilerin yağmur sensörünü aşağıdaki görselde verilen örnek devredeki gibi kurmaları, tercih ettikleri IOT araçları ile bir ekran tasarlamaları ve Arduino kartından elde ettikleri verileri bu ekranda okumaları gerekmektedir. Örnek kod verilmiştir. Ancak bu kod bloğu öğrenciler ile paylaşılmamalıdır.

Tanımlama: Tanımlama aşamasında öğrencilerin su seviyesi ölçülebilmesi ve ölçülen bilginin ne şekilde Adafruit ekranına gönderilebileceğini belirlemesi gerekir. Bu aşamada gerekli olan bilgiler detaylı olarak yazılmalıdır.

Örnek:

- Su seviyesindeki değişimi algılamak için analog ölçüm gerekir.
- Adafruit ekranında sayısal verilerin gösterilebildiği araçlar kullanılır.

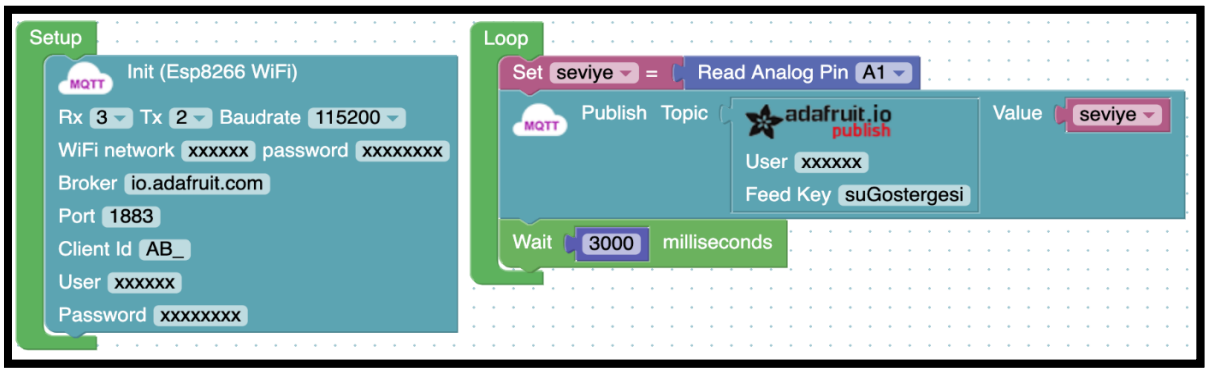
Fikir üretme: Bu aşamada öğrencilerin tanımlama adımında belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Tanımlamada belirlenen her bir adımın gerçekleştirilebilmesi için gerekli fikirler belirlenmelidir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

- Su seviyesini ölçmek için yağmur sensörü kullanılabilir. Ancak sadece 4 cm derinliğindeki su seviyesi ölçülebilir.
- Su seviyesini A0-A5 analog uçlarından birini kullanarak Arduino kartına bağlanmalıdır.
- Adafruit ekranında grafik aracı sayısal bilgileri gösterebilmektedir. Bu nedenle grafik aracı su seviyesini göstermek için kullanılabilir.

2.1 Üret: Su doluluk göstergesi

Öğrenciler önceki etkinlikte kullanılan ve Arduino kartını internete ve internet üzerinden Adafruite bağlayan kodları tekrar kullanabilirler. Bu bilgiler ışığında devrelerini tasarlayan öğrencilerin aşağıdaki

resimde görülene benzer bir kod oluşturmaları gerekir. Kod, tasarıma ve kullanılan Adafruit aracına ve yapılan bağlantıların pin numaralarına göre değişebilir. Aşağıdaki kod ESP8286 Wifi modülünün RX olarak D2, TX olarak D3 şeklinde bağlanması ve bunun yanında yağmur sensörünün de A1 pinine bağlanması durumuna göre oluşturulmuştur. Ayrıca yağmur sensörü test sırasında kullanılan suyun iletkenlik seviyesine göre farklı değerler üretebilir. Bu nedenle minimum seviye ve maksimum seviye durumlarında hangi değerlerin okunduğu mutlaka test edilmelidir.



Resim 319. Örnek Kod Bloğu

3- DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Sunucu ve istemci nedir, tartışınız (Not: Tartışmanın ardından rehber öğretmen öğrencileri sunucu ve istemci hakkına somut örnekler kullanarak bilgilendirmelidir).

2. Bu bölümde öğrenilen konular doğal/yapay felaket durumlarında nasıl kullanılabilir. Örneklendirerek açıklayınız.

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaşılan problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Adafruit sitesinde gönderdiğim değerleri okutamadım.
    - Adafruit sitesinin çalışma mantığını anlamadım.
    - ESP8266 istediğim gibi çalışmıyor.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Üret: Orman Yangın Alarmı

Bu etkinlikte amaç orman yangınlarının hızlı tespiti için oluşturulmuş bir IOT uygulaması yapmaktır. Öğrenciler Esp 8266 Wifi modülünü kullanarak internet erişimi olan ve yangın ya da alev algıladığında Adafruit üzerinde oluşturdukları ekrana alarm mesajı gönderen bir devre oluşturmalıdır. Burada öğrenciler Adafruit üzerinde istedikleri aracı kullanmakta özgürdürler. Ancak alarmın doğru şekilde okunabilmesi için dijital bilgi alışverişi yapılmalı ve alev sensörü dijital çıktı üretecek şekilde tasarlanmalıdır. Ayrıca alev sensörünün ormanda belirli bir noktaya yerleştirileceğinden mutlaka yerinin tespit edilmesi gerekmektedir. Bu nedenle öğrencilerin alarm sinyalinin yanında alarm durumunun yeri ile de bilgi göndermesi gerekmektedir. Bu durum için öğrencilerin özgün çözümler üretmesi teşvik edilmelidir.

9. Bölüm-Bluetooth ile Bonibon Sayıyorum

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini açıklayabilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol, kondansatör, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanabilir ve ArduinoBlocks ortamında programlama yapabilir.
- Öğrenciler Multimetre kullanabilir.
- Öğrenciler kızıl ötesi kumanda seti, RC522 RFID NFC kiti ve ESP8266 wifi modülü kullanarak IOT devreleri oluşturup programlayabilir.
- Öğrenciler sensörlerden alınan verilerin internete aktarımını sağlamak için gerekli elektronik devreleri oluşturup programlayabilir.
- Öğrenciler nesnelerin İnternetiyle ilgili temel kavramlarını bilerek uygulayabilir.

Bölümün Kazanımları:

- TP 223 dokunma sensörü, TCS34725 renk sensörü, 7 segmentli gösterge ve HC05 Bluetooth modülü kullanılan elektronik devreleri oluşturup gerekli kodları ArduinoBlocks ortamında oluşturabilir.
- Manyetik alan sensörü, SD kart modülü ve HC05 Bluetooth modülü kullanılan elektronik devreleri oluşturup gerekli kodları ArduinoBlocks ortamında oluşturabilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin çeşitli sensörlerden aldıkları verileri saklama ve gösterme için gerekli elektronik devreleri oluşturabilmelerini ve gerekli bileşenleri ArduinoBlocks ortamında programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, renkli bonibonlar, 2 adet servo motor, 1 adet TP 223 dokunma sensörü, 1 adet TCS34725 renk sensörü, 1 adet HC05 Bluetooth modülü, karton, yapıştırıcı, 1 adet saat modülü, 1 adet manyetik alan sensörü, 1 adet SD kart modülü ve hafıza kartı (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: TP 223 dokunma sensörü, TCS34725 renk sensörü, manyetik alan sensörü, SD kart modülü, 7 segmentli gösterge ve HC05 Bluetooth modülü kullanılan elektronik devrelerin oluşturulması ve sensörlerden gelen verilerin gösterimi/kaydedilmesi için ArduinoBlocks ortamında programların oluşturulmasının gösterilmesi.

Uygula: TP 223 dokunma sensörü, TCS34725 renk sensörü, manyetik alan sensörü, SD kart modülü, 7 segmentli gösterge ve HC05 Bluetooth modülü kullanılan elektronik devrelerin oluşturulması ve sensörlerden gelen verilerin gösterimi/kaydedilmesi için gerekli programların ArduinoBlocks ortamında oluşturulması.

Tasarla: Çeşitli sensörlerden gelen verilerin IOT ile internete aktarımı ve SD karta kaydedilmesi için gerekli elektronik devrelerin oluşturulması ve programının yapılması için tanımlama ve fikir yürütme adımlarının gerçekleştirilmesi.

Üret: Çeşitli sensörlerden gelen verilerin IOT ile internete aktarımı ve SD karta kaydedilmesi için gerekli elektronik devrelerin oluşturulması ve programının yapılması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE VE UYGULA

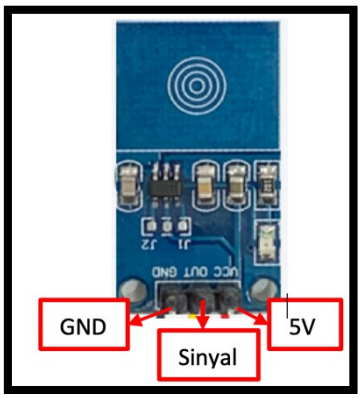
1. Gözle- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı

Malzeme Listesi:

1 adet TP 223 dokunma sensörü
1 adet TCS34725 renk sensörü
1 adet HC05 Bluetooth modülü

Bu etkinlikte amaç oluşturulan karton kutu üzerine konulan bonibonun rengine göre hareket eden; bonibon kırmızı ise kutunun içine alan, kırmızı değilse dışarı gönderen ve bu işlemi dokunma sensörü veya Bluetooth üzerinden gönderilen sinyalle gerçekleştiren bir devre oluşturmaktır. Ayrıca kutu içerisine alınan bonibon sayısı kutunun dışında bulunan 7 segmentli göstergeler ile 9' a kadar sayılacaktır. Devre kurulumunda daha önce kullanılmayan iki adet yeni sensör bulunmaktadır. Bu sensörler TP 223 dokunma sensörü ve TCS34725 renk sensörüdür. Ayrıca IOT etkinliklerinde kullanılan HC05 Bluetooth modülü ve 7 segmentli gösterge ilk defa kullanılacaktır. Bu elektronik elemanların açıklamaları aşağıda verilmiştir.

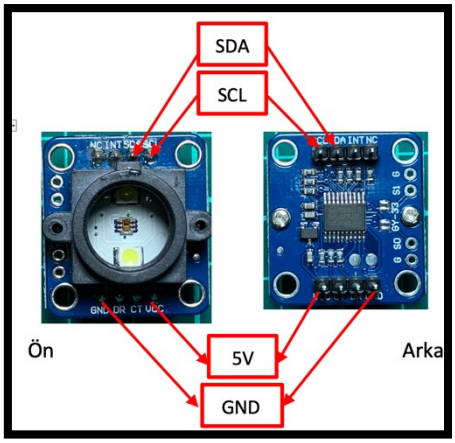
TP 223 Dokunma sensörü: dokunmayı algılayan ve bu şekilde dijital çıktı üreten bir sensördür. Bu tür sensörlere aynı zamanda kapasitif dokunma sensörü adı da verilmektedir. Bunun nedeni bir kapasitör yani kondansatör gibi çalışmasındandır. Bu sensör iki ayrı kutup içerir. Bu kutuplar bir kondansatörün iki plakasına benzetilebilir. Sensörün güç bağlantıları yapıldığında bir plakaya "+" diğer plakaya da "–" yükler yüklenir. Kapasitör üzerindeki dokunma alanına dokunulduğunda "+" ve "–" yüklerin hareket etmesine neden olur ve dolayısıyla iki plaka arasındaki yük farkı değişir. Bu da sensörün bir çıktı üretmesine neden olur. Etkinliklerde şu ana kadar kullanılan sensörlerin büyük çoğunluğu algılamayı direnç değişimi üzerinden gerçekleştirmektedir. Örneğin LDR ışık ile direnci değişen bir sensördür. Aynı şekilde toprak nem sensörü ve yağmur sensörü de direnç değişikliklerini algılayan sensörlerdir. Dokunma sensörü bunlardan farklı olarak kapasite değişikliğine duyarlıdır. Aşağıdaki resimde dokunma sensörü görülmektedir.



Resim 320. Dokunma Sensörü

TCS34725 renk sensörü: cisimlerden yansıyan ışığı algılayarak hangi renkte olduğunu tespit eden bir sensördür. Cisimler üzerlerine düşen ışığı yansıtırlar. Bazı renkleri de yansıtmadan soğururlar. Cisimlerin yansıttıkları ışık sahip oldukları rengi oluşturur. Bunun yanında PIR sensörü açıklamasında anlatıldığı gibi cisimler etraflarına kızılötesi ışınlar da yayarlar. Kızılötesi ışınlar insan gözü tarafından algılanmazlar. Ancak bu ışın türüne duyarlı araçlar tarafından algılanır. TCS34725 renk sensörü kızılötesi ışınları bloke eder. Sadece insan gözünün görebildiği renkleri algılar. Ayrıca üzerinde güneş ışığına yakın renkte ışık yayan bir led barındırır. Bu led ışığı rengi algılanacak cismin üzerine düşürerek

renginin en doğru şekilde algılanmasını sağlar. Aşağıdaki resimde TCS34725 renk sensörünün görseli ve bağlantı uçlarının açıklaması yer almaktadır.



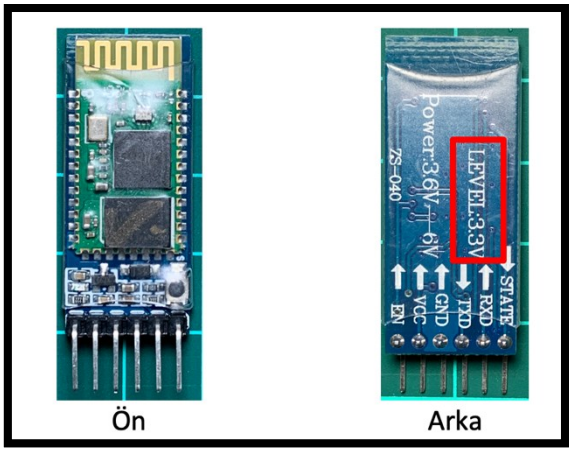
Resim 321. Renk Sensörü

Renk sensörünü kullanmak için ArduinoBlocks'ta "Sensors" blok başlığı içerisindeki "*Color" Sensor* (TCS34725) blokları kullanılır. Sensör ile ilgili üç adet blok vardır. Bu bloklar ve görevleri aşağıdaki resimde verilmiştir.

Resim 322. Renk Sensörü Blokları

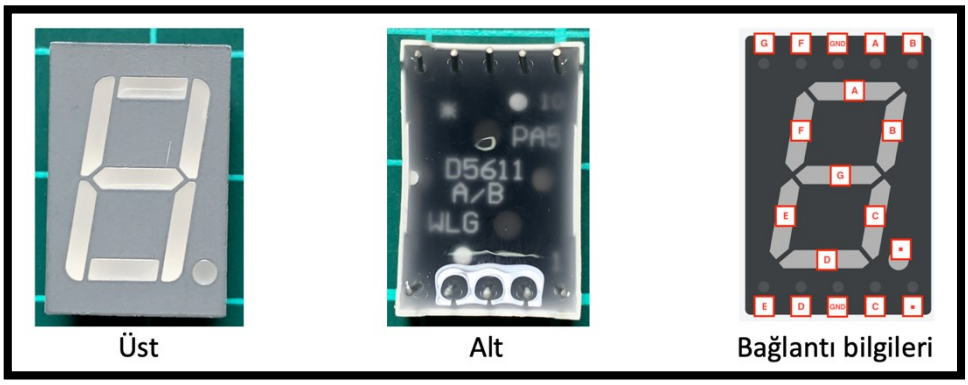
HC05 Bluetooth Modülü: IOT uygulamaları ESP8266 WiFi modülü ile internet üzerinden gerçekleştirilebileceği gibi HC05 Bluetooth modülü kullanılarak da gerçekleştirilebilir. Bluetooth bir kablosuz veri iletişim teknolojisidir. Kısa mesafeler içerisinde bir araçtan diğerine veri göndermek ve veri almak için kullanılır. Aslında kısa mesafeli radyo sinyalleri aracılığı ile iletişim gerçekleşir. Çevremizde yer alan birçok araç bu sistemi kullanabilecek donanıma sahiptir. Örneğin cep telefonları, bilgisayarlar, yeni nesi otomobiller ve hatta televizyonlar ve klimalar gibi ev gereçleri de Bluetooth bağlantısını kullanabilir. Bluetooth bağlantısına sahip araçlar ile veri iletiminin hızı araçlar arasındaki mesafeye ve gönderilecek verinin boyutuna göre değişebilir.

Etkinliklerde kullanılacak olan HC05 Bluetooth modülü ile Arduino projelerinde oluşturulan devrelere kolaylıkla Bluetooth bağlantı özelliği eklenebilir. HC05 Bluetooth modülü, ESP8266 WiFi modülü gibi veri alışverişini RX ve TX pinlerini kullanarak gerçekleştirir. HC05 Bluetooth modülünün görseli ve bağlantı pinleri aşağıdaki resimde sunulmuştur.



Resim 323. Bluetooth Modülü

7 Segmentli Gösterge: Temel olarak içerisinde 7 adet led bulundurur. Segmentlerin her biri bu ledlere karşılık gelir. Segmentler kullanılarak, yazdırılacak olan sayı için gerekli olan ledler yakılır. Bu sayede ekranda ilgili sayı belirecektir. 7 segmentli göstergeler iki türde olabilir. Bunlar Ortak Anot ve Ortak Katot türleridir. Göstergenin içerisinde yer alan ledlerin her birinin bir ucu birbirine bağlıdır. Eğer bağlı olan uçlar ledlerin katot uçları ise ortak katot, anot uçları ise ortak anot gösterge ismini alır. Arduino ile çalışırken ortak katot gösterge kullanılıyorsa ortak uç eksiye takılır ve dijital uçlardan yakılacak segmente ON bilgisi gönderilir. Ortak anotta ise yakılacak segmente dijital pinlerden OFF bilgisi gönderilir ve ortak anot uç 5V'ye takılır. Aşağıdaki resimlerde, setlerde bulunan 7 segmentli ortak katot gösterge ve göstergenin bağlantı bilgileri bulunmaktadır.



Resim 324. 7 Segmentli Gösterge

1.1 Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı: Kapak Hazırlığı ve Montajı

Malzeme Listesi:

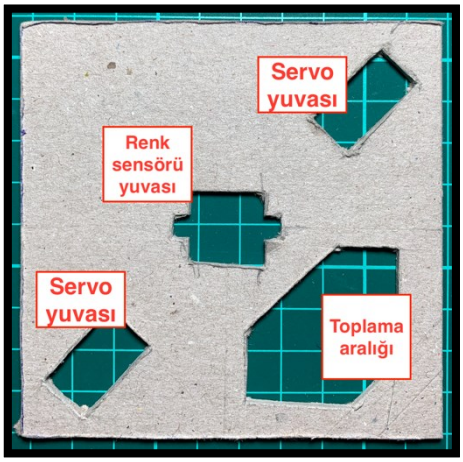
2 adet servo motor
1 adet TP 223 dokunma sensörü
1 adet TCS34725 renk sensörü
Karton
Yapıştırıcı

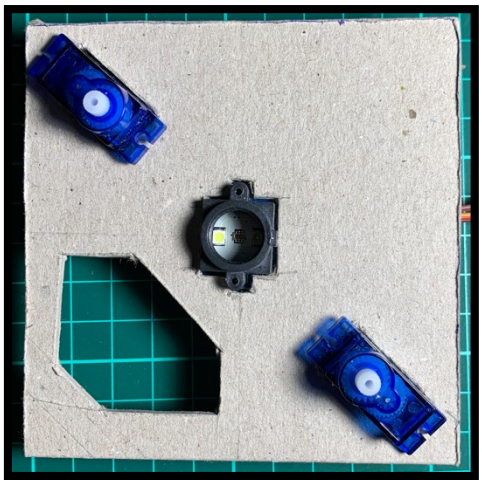
Bu etkinliklerdeki amaç kırmızı renkli bonibonları kutunun içerisine diğer renklerdeki bonibonları ise kutunun dışına otomatik olarak atan bir cihaz tasarlamaktır. Bonibonlar sırayla renk sensörünün üzerine konulacaktır. Konulduktan sonra dokunma sensörüne dokunulur. Eğer bonibon kırmızı ise cihaz onu

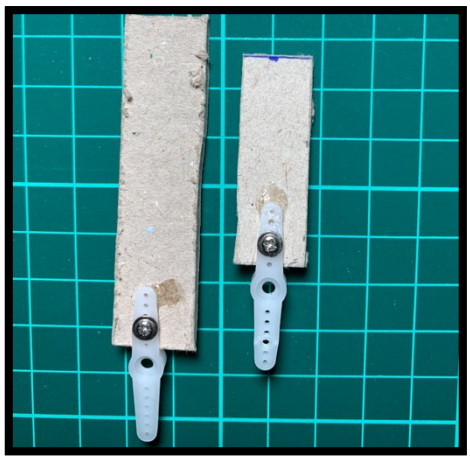
kutunun içerisine, eğer kırmızı değilse dışına atar. Bonibonların toplanması için öncelikle bir kutuya ihtiyaç vardır. Bu etkinlikte bir kutu oluşturulacaktır. Bu kutunun yüksekliği öğrenci isteğine göre değişebilmekle birlikte üst kısmında yer alan kapak bölümünün 10x10 cm boyutlarında bir kare olması gerekmektedir. Bu kare kapak bölümünün üzerine iki adet servo motor, bir adet renk sensörü yerleştirilecek ve aynı zamanda bonibonların kutunun içine düşmesini sağlayacak olan bir delik açılacaktır. Aşağıda motor yuvalarının, renk sensörünün, bonibonların düşeceği deliğin nasıl olması gerektiğini gösteren görseller bulunmaktadır.

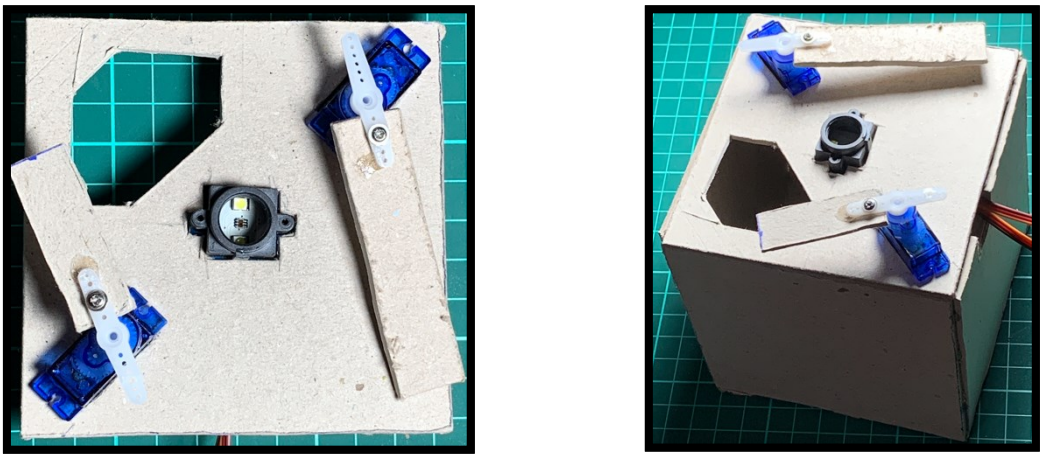
Servo motorların yerleşeceği kısımların dikdörtgen olması nedeniyle ebatları cetvelle ölçülür, karton üzerine çizilip kesilir. Renk sensörünün tam olarak karton kapağa oturması için plastik korumasının vidalarının sökülerek çıkarılması ve koruma parçasının kalıp olarak kullanılması faydalı olabilir. Koruma parçası kalıp olarak kullanılarak karton kapağın üst kısmına renk sensörü için açılacak deliğin şekli çizilebilir. Daha sonra koruma parçası yerine takılabilir. Karton çizilen kısımlardan kesilerek yuva oluşturulur. Kapağın motorlar ve sensörler takılmış hali aşağıdaki görseldeki gibi olmalıdır.

Oluşturulan tasarım ile ortada yer alan sensörün üzerine konulacak bonibonun rengi algılanacak ve buna göre kutunun içine alınacak veya dışarıya gönderilecektir. Algılama işlemi renk sensörü tarafından yapılırken bonibonu hareket ettirme işi servo motorlarla gerçekleştirilecektir. Bu nedenle servo motorlar için iki adet hareket kolu tasarlanmalıdır. Bu kolları öğrenciler dikdörtgen şekilde kesilmiş, boyu 5 cm ve eni 2 cm olan kartonlarla oluşturabilir. Bu kartonların boyutları isteğe göre değiştirilebilir, kesin ölçüler değildir. Öğrenciler kendi düzeneklerine uygun boyutu kendileri belirleyebilirler. Aşağıdaki resimlerde bu kolların takılı olduğu bir örnek verilmiştir.









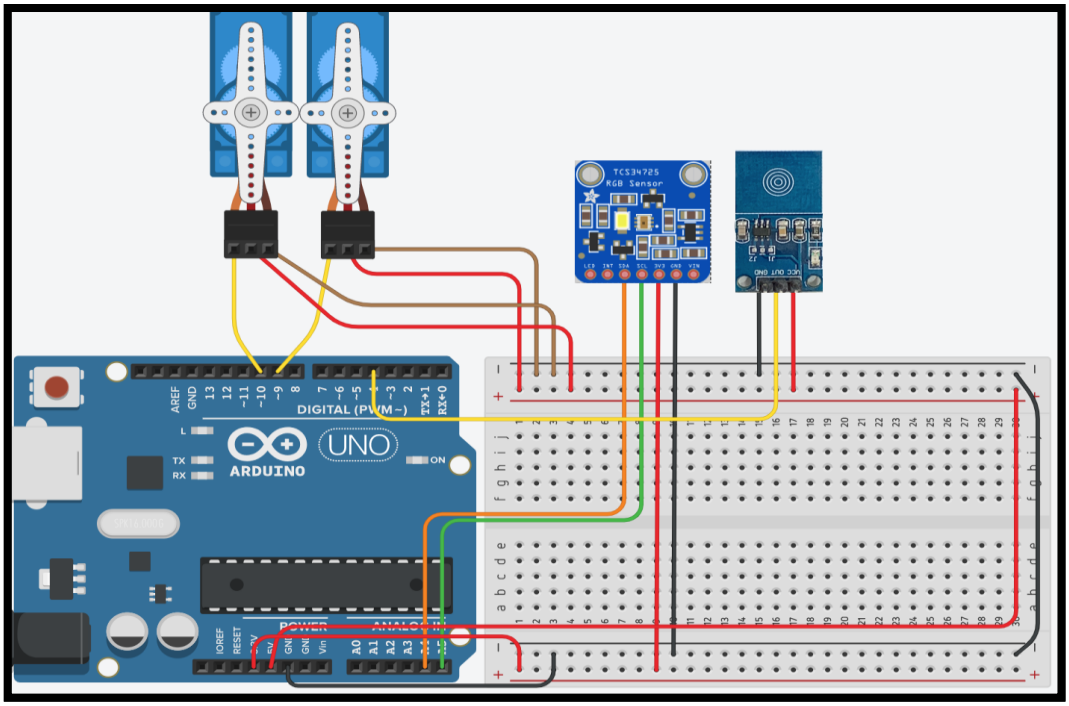
Resim 325. Bonibon Kutusu Oluşturma Aşamaları

1.2 Gözle ve Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı: Devre Kurulumu

Malzeme Listesi:

Arduino Uno
Breadboard
Renkli bonibonlar
2 adet servo motor
1 adet TP 223 dokunma sensörü
1 adet TCS34725 renk sensörü
Karton
Yapıştırıcı

Kapak ve kutu hazırlandıktan sonra aşağıdaki resimde görüldüğü gibi devrenin kurulması gerekir. Devre kurulumunda gösterilen servo motorlar ve renk sensörünün bir önceki aşamadaki kapakta yer alan servo motorlar ve renk sensörü olduğuna dikkat edilmelidir.

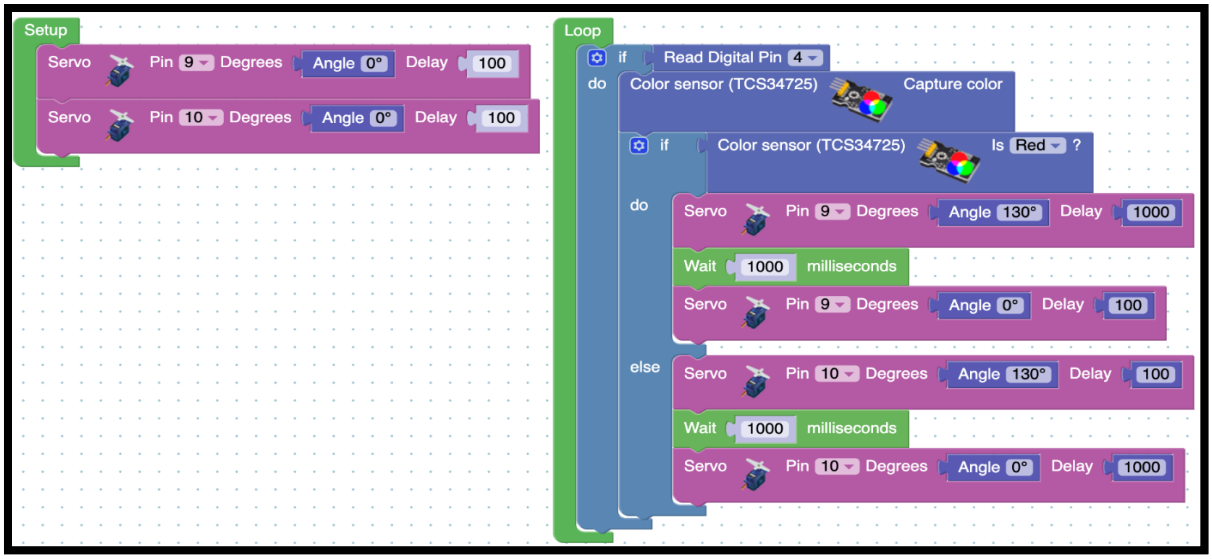


Resim 326. Örnek Devre Şeması

Dikkat

Resimde yer alan devre elemanları ile setlerden çıkan elemanlar arasında görünüş ve bağlantı uçlarının sıralaması açısından farklılıklar olabilir. Devre elemanlarının üzerindeki etiketler dikkatle okunmalıdır.

Rehber öğretmen öğrencilerin yukarıda verilen devreyi kurmalarına yardımcı olur. Rehber öğretmen özellikle servo motorların PWN pinlerine takıldığından emin olmalıdır. Rehber öğretmen daha sonra öğrencilerden servo motorların açısını "0" derece yapmalarını ister. Bunun nedeni, hareketli kolları takmadan ve kodlamaya geçmeden önce servo motor açılarını doğru konumlandırmaktır. Hazırlanan hareketli kollar servo motorlar "0" dereceye ayarlandıktan sonra takılabilir. Öğrencilerden bir bonibonun renk sensörünün üzerine konulduktan ve dokunma sensörüne basıldıktan sonra bonibon kırmızı ise açılan delikten kutunun içine düşmesini, değil ise kutunun dışına itilmesini sağlayan kodu yazmaları istenir. Farklı boyutta bonibonlar bulunmaktadır. Bu etkinlikte kullanılan bonibonlar standart boyutta olanlardır. Rehber öğretmen bonibonların boyut olarak renk sensörünün korumasına denk gelip gelmediğini etkinlik öncesinde kontrol etmelidir. Öğrenciler kırmızı yerine sevdikleri başka bir renk için de bu işlemi yapabilirler. Aşağıdaki resimde bu iş için kullanılabilecek bir kod örneği verilmiştir.



Resim 327. Örnek Program

Yukarıda verilen kod bloğunda, ilk olarak servo motorlar başlangıç konumu olan 0 dereceye ayarlanmaktadır. Bu işlem setup bloğu içinde gerçekleştirilir. Daha sonra 4 numaralı dijital pine takılı olan dokunma sensörü ON bilgisi gönderdiğinde renk okuma işlemi başlar. Burada 4 numaralı dijital pindeki değer herhangi bir karşılaştırmaya tabi tutulmamıştır. Çünkü dokunma sensörüne basıldığında ON bilgisi gelir bu da if bloğu tarafından "Doğru-True" olarak algılanır.

Renk okuma işleminde bu rengin kırmızı olup olmadığı bilgisi renk sensöründen alınır. Dokunma sensöründen gelen bilginin okunması sırasındaki işlem burada da uygulanır. Yani renk kırmızı ise renk sensöründen ON bilgisi geleceği için if bloğu tarafından "Doğru-True" olarak yorumlanır. Renk kırmızı değilse sönsörden OFF bilgisi geleceği için yanlış olarak yorumlanır. Servolar hareket ederken hızları da delay bölümüne girilen değerler ile kontrol edilmiştir. Çok hızlı ya da yavaş olmaları düzeneğin doğru çalışmasını etkileyebilir. Bu durumda öğrencilerden kurdukları düzeneğe uygun hızları belirlemeleri istenir.

Dikkat

TCS34725 renk sensörü çalışmak için 3.3V gerilime ihtiyaç duyar. 5V ile de sorunsuz çalışabilir. Ancak uzun vadeli kullanımlarda 3.3V tercih edilmelidir.

TCS34725 renk sensörü her zaman yüksek hassasiyetle çalışmayabilir. Renkleri tanıyamadığında RGB değerleri kullanılarak programcı tarafından o renk tanıtılabilir.

1.2 Uygula: Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı- Bluetooth Bağlantısı

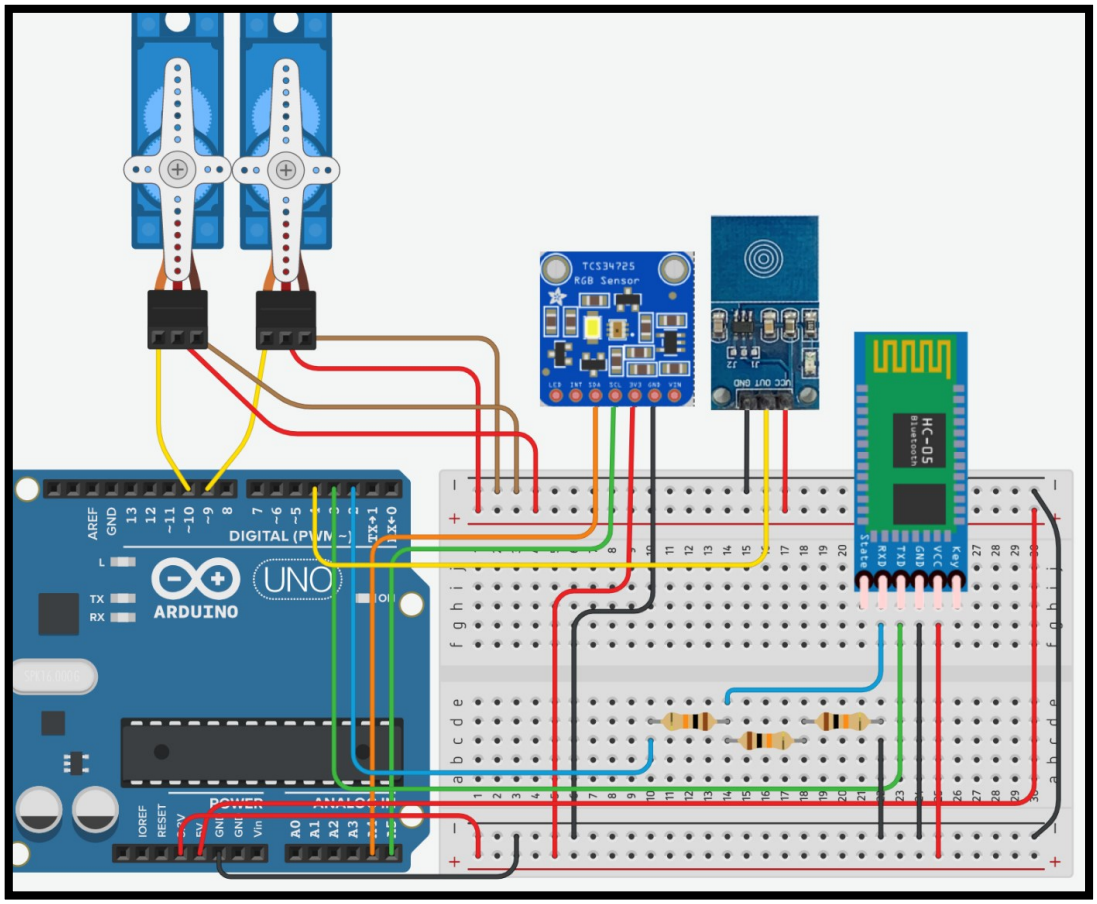
Malzeme Listesi:

Arduino Uno
Breadboard
Renkli bonibonlar
2 adet servo motor
1 adet TP 223 dokunma sensörü
1 adet TCS34725 renk sensörü
1 adet HC05 Bluetooth modülü
Karton ve Yapıştırıcı

Dikkat

Bluetooth bağlantıları kısa alanlar için etkili bir bağlantı türüdür. Etkinlikte kullanılan modüller bağlantı kurulumu esnasında aynı adı taşıyacaklarından öğrencilerin farklı grupların modüllerine bağlanma durumları olabilir. Bu yüzden HC 05 modüllerinin ön tanımlı isimleri değiştirilmelidir. Rehber öğretmenlerin etkinlik öncesinde öğrencilerin kullanacakları HC05 modüllerinin ismini EK-6'dan (HC05 İsim Değişikliği) yararlanarak değiştirmeleri gerekir.

Bu etkinlikte amaç bilgisayar ile Arduino kart arasında Bluetooth aracılığı ile bağlantı kurup oluşturulan düzeneği, dokunma sensörüne ek olarak, bilgisayarı kullanarak çalıştırmaktır. Devre bağlantıları yapılmadan önce öğrencilerden çevrelerinde gördükleri ve kablosuz bağlantı ile çalışan cihazlara örnek vermeleri istenir. Bunların hangi tür bağlantıları kullandıkları sınıfça tartışılır. Daha sonra düzeneğe aşağıdaki resimde verildiği şekilde HC05 Bluetooth modülü bağlantısı yapılır.



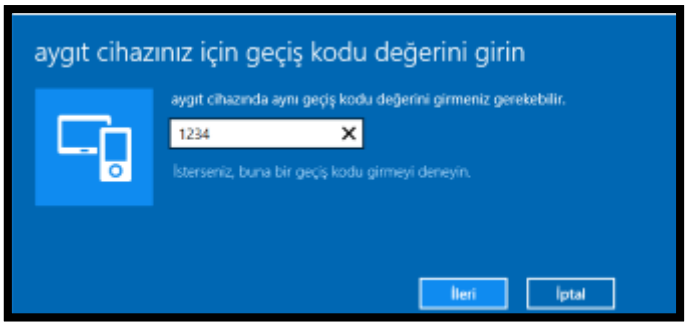
Resim 328. Örnek Devre Şeması

Bağlantılar yapıldıktan sonra öncelikle bilgisayar ile HC-05 Bluetooth modülü arasındaki bağlantı kurulmalıdır.

Dikkat

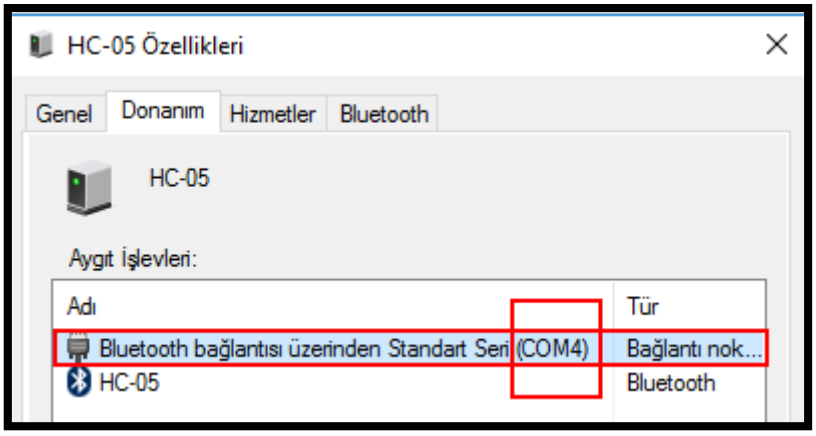
Bluetooth modülü 3.3V ile 6V arasındaki değerlerde sorunsuz çalışmaktadır. Bu durum güç bağlantısı için geçerlidir. Ancak modülün RX girişine gönderilen verinin 3.3V olması daha sağlıklı olacaktır. Uzun süreli kullanımlarda HC05 Bluetooth modülünün RX girişine gönderilen sinyalin voltajının düşürülmesi gerekmektedir. Bunun için eşdeğer üç direnç seri bağlanır. Bir ucu GND'ye takılırken diğer uca Arduino UNO'dan gelen sinyal ucu takılır. GND bağlı uçtan başlayarak 2. ve 3. direncin arasından RX ucuna bağlantı yapılır. Bu şekilde 5V olan Arduino çıkışının yaklaşık 3.3V'luk kısmı modüle gönderilir.

Bilgisayarın Bluetooth üzerinden HC-05'e bağlantısının yapılması için bilgisayarın Bluetooth seçeneklerinden (Bluetooth Ayarları/Bluetooth and Other Device Settings) HC-05 bulunur ve üzerine tıklanır (Eşleştir). Aşağıdaki resimde görüldüğü gibi açılan pencerede şifre/pin sorulacaktır. Şifre olarak "1234" girilir. Şifre girildikten sonra bağlantı gerçekleşmiş olur.



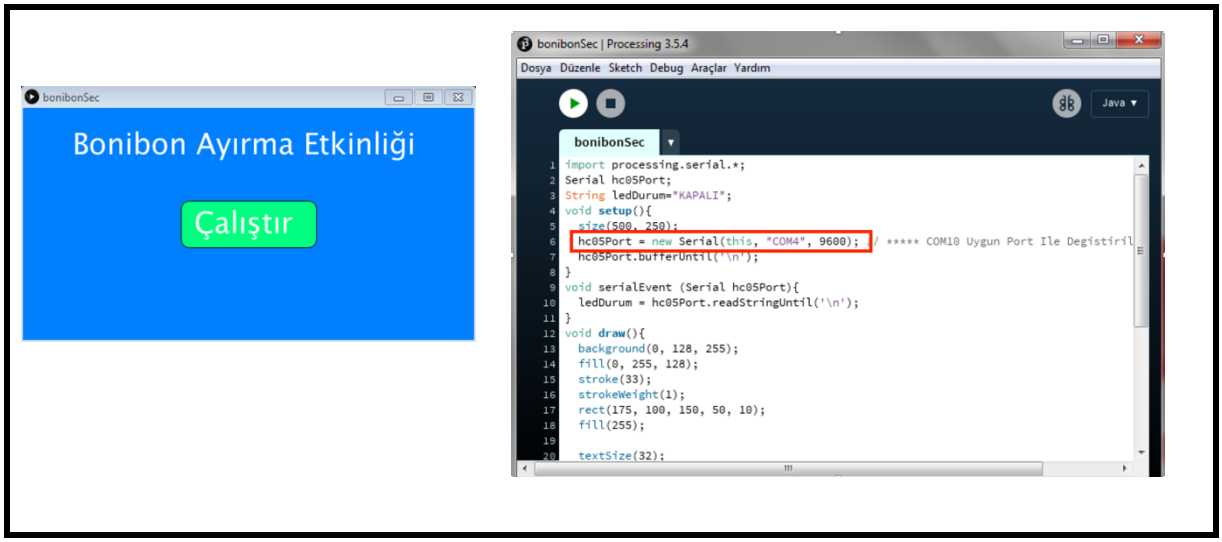
Resim 329. Bilgisayar Bluetooth Ayarı

Bluetooth bağlantısı kurulduktan sonra yerine getirilmesi gereken iki temel görev daha vardır. Bunlardan birincisi Arduino kartına yüklenecek kodun oluşturulması diğeri ise bilgisayar üzerinden Bluetooth aracılığıyla gönderilecek uygulamanın oluşturulmasıdır. Bilgisayardaki uygulama hazır olarak verilecektir. Bu uygulamanın çalışabilmesi için Processing IDE'nin bilgisayarda kurulu olması gerekir. İlgili programa "https://processing.org/download/" adresinden ulaşılabilir. Etkinlikle verilecek olan uygulamanın adı bonibonSec'dir. Bu programa EK-4 BonibonSeç Kaynak Kodu'ndan ulaşılabilir (bağlantı verilebilir). Uygulamanın kaynak kodlarında HC05'in bağlı olduğu port bilgisinin değiştirilmesi gerekebilir. Bunun için öncelikle HC05'in hangi COM portu üzerinden bilgisayara bağlandığı bulunmalıdır. "Denetim Masası" (Control Panel), "Donanım ve Ses" (Hardware and Sound), "Aygıtlar ve Yazıcılar" (Devices and Printers) penceresinin alt kısmında bulunan Belirtilmemiş (Unspecified) başlığını altından HC-05 bulunur (bunun için bilgisayara öncelikle HC-05'in bağlanmış olması gerekir). HC-05 üzerine çift tıklandığında açılan pencereden "Donanım" (Hardware) sekmesi seçilir. Aşağıdaki resimde görüldüğü gibi "Aygıt İşlevleri" bölümünde HC-05'in hangi COM portuna bağlı olduğu yazılıdır. Öğrencilerin HC05 modüllerinin isimleri etkinlik öncesi değiştirilmiştir. Bu durum dikkate alınarak öğrenciler rehber öğretmen tarafından COM portlarının bulunması hakkında bilgilendirilmelidir.



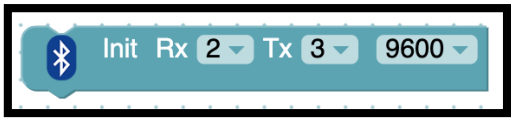
Resim 330 Bilgisayar Bluetooth Ayarları

Buradaki örnekte HC-05, COM4 portu üzerinden bilgisayara bağlanmıştır. bonibonSec programı Processing IDE ile açılır. Aşağıdaki resimde gösterildiği gibi, açılan kaynak kodda altıncı satırda bulunan COM portu bilgisi düzenlenir (buraya öğrenciler buldukları port bilgisini yazmalıdır). Daha sonra program kaydedilir (Dosya> Kaydet). Bu aşamada Arduino programı hazır olmadığı için bonibonSec programı çalışmayacaktır. Bir sonraki adımda Arduino programı yazılıp yüklendikten sonra bonibonSec programı çalıştırılmalıdır.



Resim 331. Bluetooth Uygulaması

Son aşama, ArduinoBlocks kodlarının oluşturulmasıdır. ArduinoBlocks, Bluetooth bağlantısı ile ilgili komutları yine aynı ismi taşıyan Bluetooth kod başlığı atında sunmaktadır. İlk yapılması gereken HC-05 Bluetooth modülünün RX ve TX bacaklarının hangi pinlere takıldığını belirlemektir. Yukarıda verilen devre şemasında RX 2 numaralı, TX ise 3 numaralı dijital pine takılmıştır. Bu bilgiler aşağıdaki resimde görülen komut aracılığı ile setup bloğu içerisine yerleştirilir.



Resim 332 Bluetooth Modülü Bağlantı Bloğu

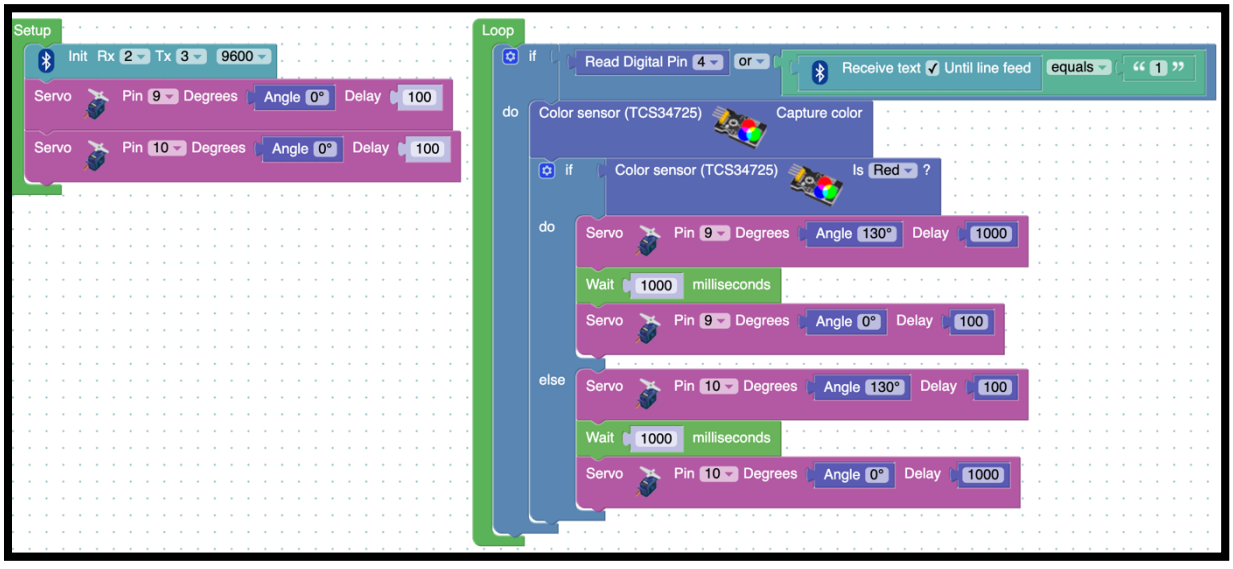
Dikkat

"RX" pini veri alınan (receive), "TX" pini ise veri gönderilen (transmit) pin anlamına gelir. Veri alışverişi yapan iki araç (burada Arduino kartı ve Bluetooth modülü örnek olarak verilebilir) birbirine bağlanırsa birinin "RX" pini diğerinin "TX" pini ile eşleşmelidir. Metin tabanlı kodlama esnasında bu çaprazlama işi kodlar aracılığı ile yapılmaktadır. ArduinoBlocks ise bu çaprazlama işlemini kendisi düzenlemektedir. Bu nedenle Bluetooth modülünün "RX" pini Arduino kartında hangi pine takılmışsa bu pin yine komutun "RX" kısmında belirtilmelidir. Aynı durum TX pini için de geçerlidir. Ayrıca, Arduino Kart üzerindeki donanımsal "RX" ve "TX" pinlerinin kullanılmaması kod yükleme sırasında oluşacak hataların da önüne geçecektir.

Bluetooth üzerinden veri göndermek ve veri almak için "*Send*" ve "*Receive text*" blokları kullanılır. Bu bloklar kullanılma amaçlarına göre değişiklik gösterirler. Bu etkinlikte bilgisayar üzerinden gönderilen bilgi okunacağı için "*Receive text*" bloğu kullanılacaktır. Okunan bilgi türü de metinsel olacağından, metinsel bilgi okumak için kullanılan blok seçilmiş ve resmi aşağıda sunulmuştur.

Resim 333. Receive Text Bloğu

"bonibonSec" programında bulunan Çalıştır butonuna tıkladığında "1" verisini metin olarak göndermektedir. Bu nedenle sayısal bir veri gibi görünmesine rağmen metinsel olarak işlem görecektir. Bluetooth ile de bonibon seçme işlemini yapabilmek için etkinliğin önceki aşamalarında yazılan kodlar aşağıdaki resimde güncellenmelidir.



Resim 334. Örnek Program

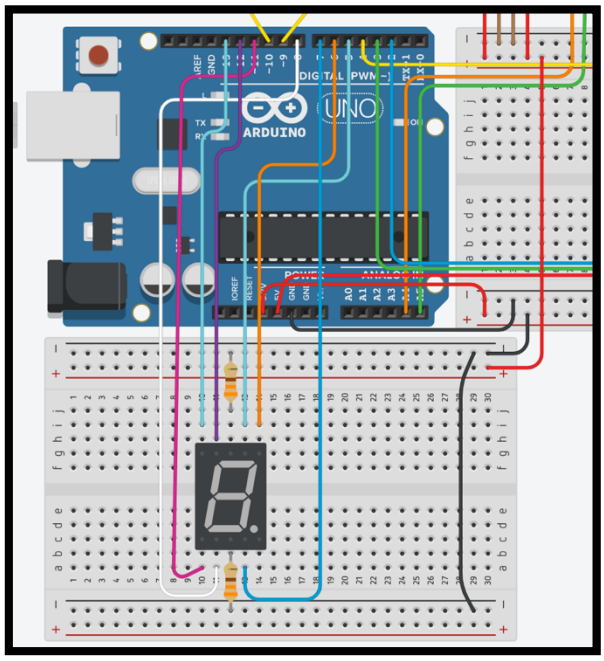
Bu programda önceki programa ek olarak, Setup bloğu içerisinde RX ve TX bağlantı pinleri belirlenmiştir. Ayrıca düzeneğin çalışması için tetikleme görevi gören dokunma sensörünün yanına Bluetooth üzerinden okunan bilgi de eklenmiş ve her iki koşulda da mekanizmanın çalışması sağlanmıştır. Arduino kartına program yüklendikten sonra bonibonSec programı çalıştırılır ve cihazın çalışması kontrol edilir.

1.2 Uygula- Bluetooth Kontrollü Kırmızı Renkli Bonibon Ayırıcı- Sayı Göstergesi

Malzeme Listesi:

Arduino Uno
Breadboard
Renkli bonibonlar
2 adet servo motor
1 adet TP 223 dokunma sensörü
1 adet TCS34725 renk sensörü
1 adet HC05 Bluetooth modülü
1 adet 7 segmentli gösterge
2 adet 330 ohm direnç
Karton ve Yapıştırıcı

Bu etkinlikte amaç Bluetooth ya da dokunma sensörü ile kontrol edilen düzenekteki bonibon sayısını 9'a kadar sayan ve bunu 7 segmentli göstergede gösteren devreyi oluşturmak ve kodlarını yazmaktır. Bunun için öncelikle devreye 7 segmentli göstergenin eklenmesi gerekmektedir. Sayfa düzeni nedeniyle ek bir breadboard kullanılarak eklenen 7 segmentli gösterge bölümü resmi aşağıdaki gibi verilmiştir. Ayrı bir breadboard yerine aynı breadboard kullanılabilir.

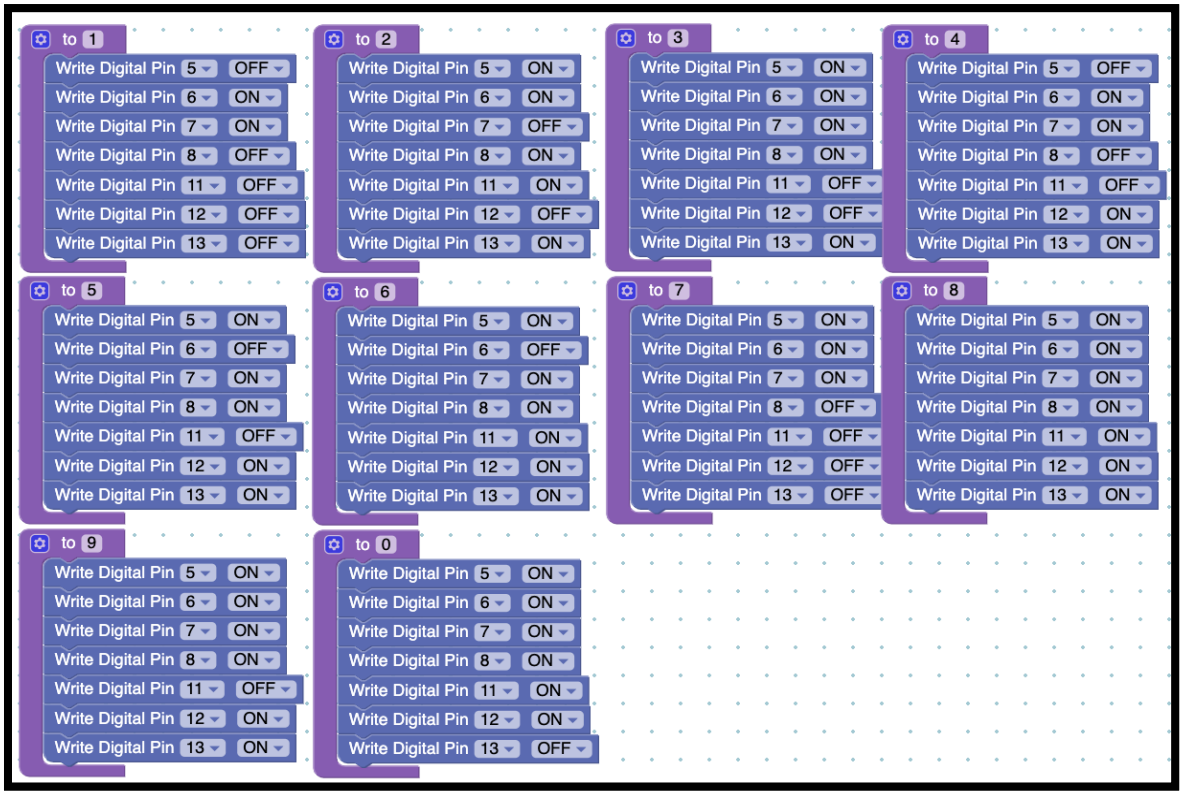


Resim 335. Örnek Devre Şeması

7 segmentli göstergenin ortak katot uçları 330 ohm değerindeki dirençlerle GND hattına bağlanmıştır. Bu dirençler göstergeyi yüksek akımlardan korumak içindir. Daha sonra A ucu 5 numaralı, B ucu 6 numaralı, C ucu 8 numaralı ve D ucu 9 numaralı dijital pinlere bağlanmıştır. E, F ve G uçları ise sırasıyla 11, 12 ve 13 numaralı uçlara bağlanmıştır.

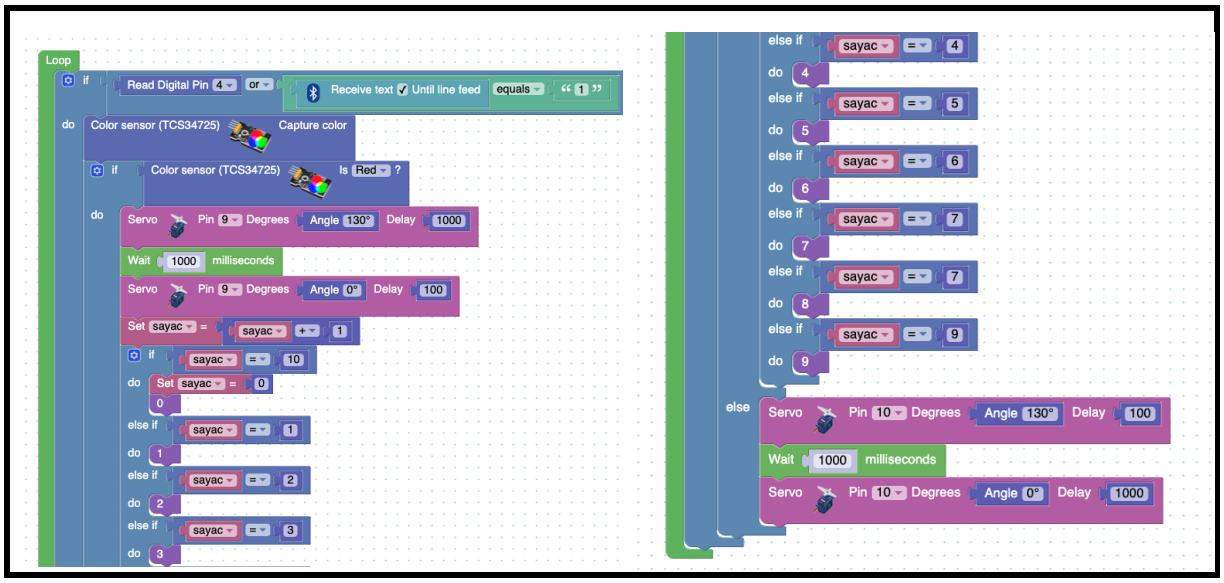
Devre kurulduktan sonra 7 segmentli göstergenin rakam gösterebilmesi için yakılacak (ON) olan segmentlerin belirlenmesi gerekir. Bunun için projeksiyon ile tahtaya etkinlik başında görseli verilen 7 segmentli gösterge yansıtılır. Öğrencilerden 0'dan 9'a kadar olan rakamları gösterebilmek için hangi segmentlerin yakılması (ON) gerektiğini belirlemeleri ve not almaları istenir. Tüm rakamlar belirlenmeden kodlama işlemine geçilmemelidir. Daha sonra her bir rakam için rakamın ismini taşıyan

bir fonksiyon oluşturulur. Bu fonksiyon çalıştığında ilgili rakam göstergede belirmelidir. Aşağıdaki resimde bu fonksiyonların oluşturulan devre şeması ile uyumlu çalışan kodları verilmiştir.



Resim 336 7 Segmentli Gösterge Fonksiyonları

Fonksiyonlar oluşturulduktan sonra ana kod bloğuna aşağıda gösterilen eklentiler yapılmalıdır. Görselde ifade edildiği gibi kutu içine düşen bonibonları sayan bir sayaç konulmuştur. Bu sayacın aldığı değere göre 7 segmentli gösterge için uygun olan rakamı yazdıran fonksiyon çalıştırılmıştır. Kırmızı bonibon sayısı 10'a ulaştığında sayaç değişkeni sıfırlanmıştır. Aşağıdaki resimde görülen kod bloğu yazılmadan önce oluşturulan sayaç değişkeni Setup bloğu içerisinde sıfırlanmalıdır.



Resim 337 Örnek Program

Görüldüğü gibi yukardaki kod bloğu aslında sayaca göre ilgili sayıyı göstergede yazdıran fonksiyonu çağırmaktadır. Sayaç değişkeninin kontrol edildiği if bloğu içerisinde ilk kontrol edilen durum sayacın 10'a ulaşıp ulaşmadığıdır. Bonibon sayısı 10'a ulaşmışsa sayaç sıfırlanır ve göstergede 0 rakamı görüntülenir. Burada öğrencilere 100'e kadar olan bonibon sayısını göstergede yazdırmak için ne yapılması gerektiği sorusu yöneltilmeli, öğrencilerin zihinlerinde oluşan algoritmalar sınıfça tartışılmalıdır.

1.3 Gözle ve Uygula- Mıknatıs Detektörü

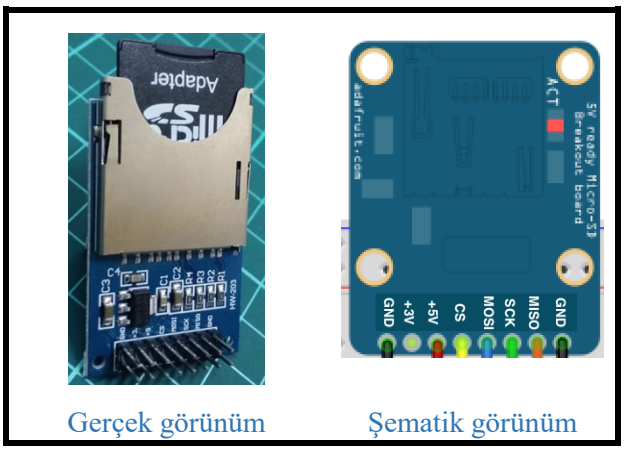
Malzeme Listesi:

Arduino Uno
Breadboard
1 adet saat modülü,
1 adet manyetik alan sensörü
1 adet SD kart modülü ve hafıza kartı
1 adet HC-05 Bluetooth modülü

Bu etkinlikte amaç, manyetik alan sensörü ile tespit edilen mıknatıs özelliği taşıyan bir nesne ile ilgili bilgileri SD Kart içerisinde oluşturulan bir metin dosyasına yazdırmaktır. Yazdırılacak bilgiler mıknatısın tespit edildiği tarih ve saat bilgileridir. Ayrıca mıknatısın tespit edilmesi durumunda yazdırma işlemi bilgisayardan Bluetooth üzerinden gönderilen bir komut aracılığı ile gerçekleştirilecektir. SD kart içerisine yazılan bilgiler ise kontrol edilerek doğru yazma işlemi yapılıp yapılmadığı tespit edilecektir.

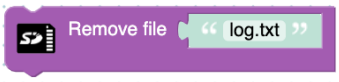
Bu etkinlikte daha önceki etkinliklerde kullanılmayan iki temel eleman kullanılacaktır. Bu araçlar SD kart ve manyetik alan sensörüdür. Araçların tanımları ve kullanım şekilleri aşağıda tanıtılmıştır.

SD kart ve kart modülü: SD kartlara bilgi yazmaya ve yazılı bilgileri okumaya yarayan bir modüldür. SD kart modülüne hafıza kartları doğrudan takılıp çalıştırılabilir. SD kart modülü 3.3 ve 5V gerilimler altında sağlıklı bir şekilde çalışabilir. Aşağıdaki görsel SD kart modülünün bağlantı uçlarını açıklamaktadır. Modül üzerinde 6 adet bağlantı pini vardır. Bu pinlerden iki tanesi güç bağlantıları (5V ve GND) içindir. Kalan dört pin ise veri transferi için kullanılır. SD kart modülü Arduino devrelerinde kullanılan veri iletim yöntemlerinden SPI veri transfer yöntemini kullanır. Arduino UNO SPI bağlantısı için 10, 11, 12 ve 13 numaralı pinleri kullanır. Bu pinlerden 10 numaralı pin SS bağlantısı, 11 numaralı pin ise MOSI bağlantısı içindir. 12 ve 13 numaralı pinler ise sırasıyla MISO ve SCK bağlantıları içindir. SD kart modülünün ilgili pinleri Arduino UNO üzerindeki ilgili pinlere bağlanarak kullanılır. Aşağıdaki resimde SD kart modülü görülmektedir.

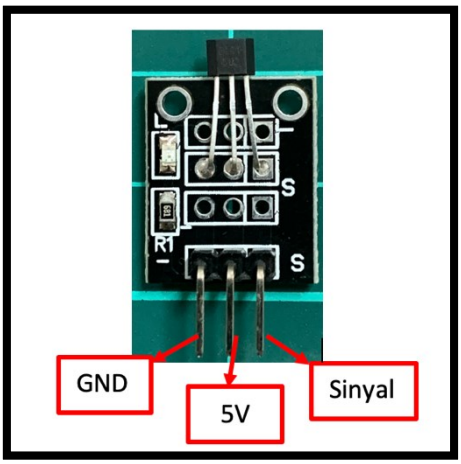


Resim 338. SD Kart Modülü

SD kart takılı modül ile ilgili belli başlı Arduino komutları ve açıklamaları aşağıda verilmiştir.

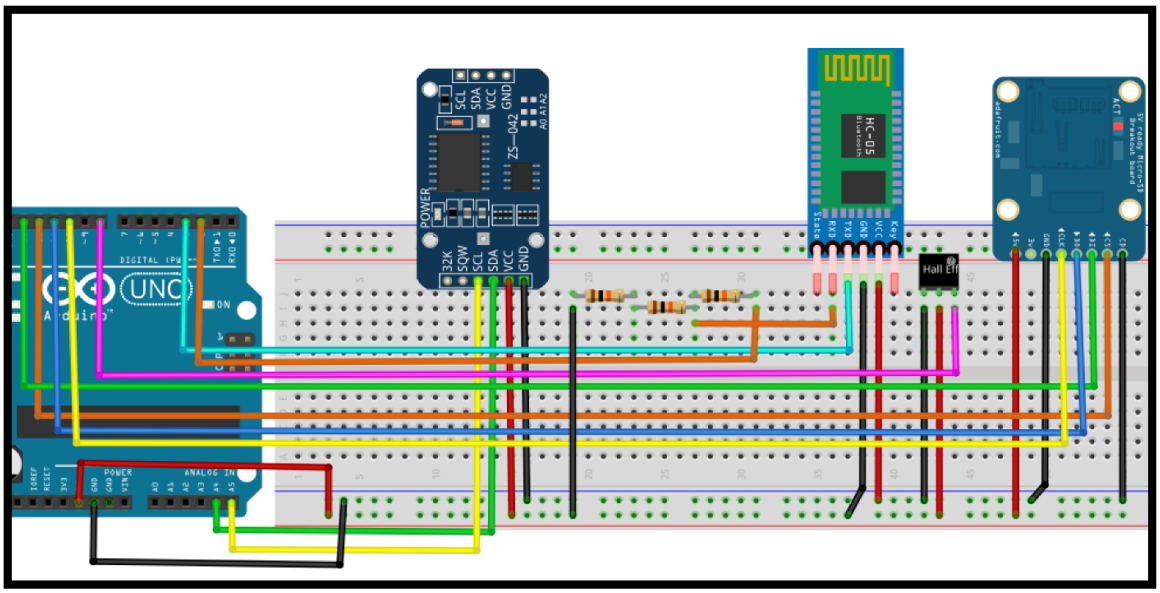
	Bu komut SD kart modülünün kullanılmaya başlamasını sağlar. Setup bloğu içine yerleştirilir ve pin numarası olarak SS (CS) pininin bağlı olduğu numara girilir.
	Bu komut SD kart içinde daha önceden oluşturulan bir dosya içine veri yazma işlemi için kullanılır. Eğer bu dosya önceden oluşturulmadıysa önce dosyayı oluşturur, daha sonra yazma işlemini gerçekleştirir. Komut içindeki ilk bölüme dosya ismi, ikinci bölüme ise dosya içine yazılacak veri girilir. Etkinlikte kullanılacak dosya tipi text dosyası olacağı için oluşturulan dosyanın uzantısı txt olmalıdır. *Line feed* seçeneği işaretlenirse dosyaya veri yazıldıktan sonra bir sonraki satıra geçilir. Bu seçenek işaretli değilse bir sonraki veri önceki ile aynı satırda yan yana yazılır.
	Oluşturulan bir dosyayı silmek için kullanılır.
	Oluşturulmuş bir dosyanın boyutunu öğrenmek için kullanılır.

Manyetik alan sensörü: Mıknatıs gibi manyetik olan bir cisme yaklaştırıldığında dijital çıkış üreten bir sensördür. Örneğin manyetik alan sensörü bir mıknatısa yaklaştırıldığında manyetik alandan etkilenir ve dijital bir çıktı üretir. Üç adet bağlantı noktasına sahiptir. Bunlardan ikisi güç bağlantıları içindir. Kalan uç ise dijital çıktı içindir. Manyetik alan sensörü herhangi bir manyetik alan yok ise (manyetik alan her yerde bulunur ancak çoğu zaman göz ardı edilecek kadar azdır) 5V çıktı üretirken, manyetik alan olan bir ortama yaklaşınca 0V değerinde bir çıktı üretir. Manyetik alan sensörünün bağlantı uçlarını içeren resim aşağıda görülmektedir.



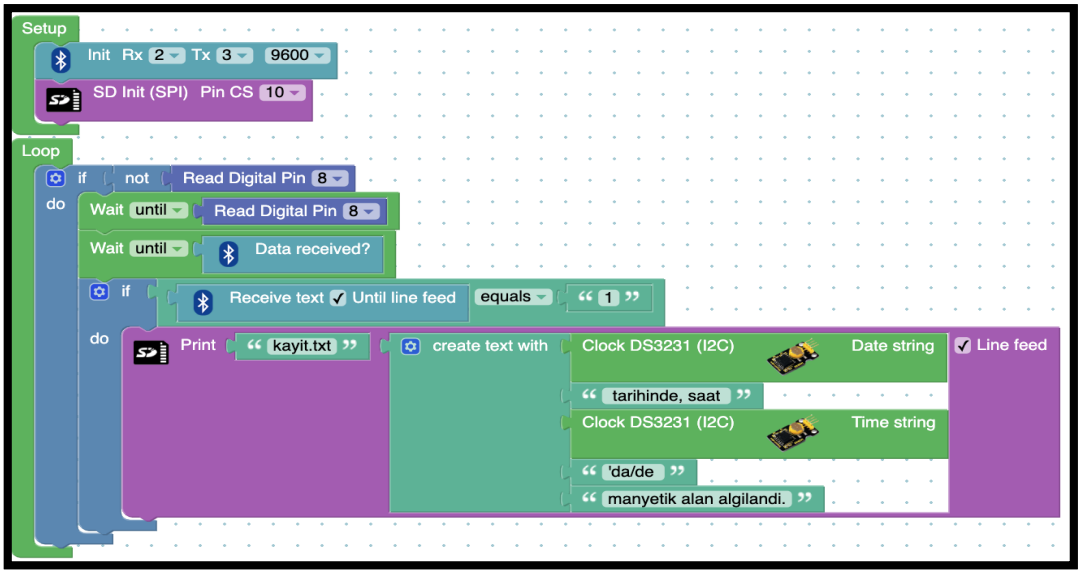
Resim 339. Manyetik Alan Sensörü

Devrede yer alan dört temel parça aşağıdaki resimde görüldüğü şekliyle Arduino kartına bağlanır. Şemada bulunan "*Hall Eff*" isimli eleman manyetik sensördür. Manyetik sensörün sinyal bacağı Arduino kartın sekiz numaralı dijital pinine bağlanmıştır. SD kart modülünün SPI bağlantı noktalarına ve saat modülünün I2C bağlantı noktalarına doğru şekilde bağlandığından emin olunmalıdır. Burada öğrencilere bu iki aracın kendilerine özel bağlantıları olduğu ve bu bağlantılara uygun noktalara bağlanmaları gerektiği vurgusu yapılır.



Resim 340. Örnek Devre Şeması

Devre kurulduktan sonra bir önceki etkinlikte anlatılan Bluetooth bağlantısı kurulma adımları tekrarlanır. Daha sonra etkinlik materyalleri arasında verilen "manyetikKayit" (EK-5 ManyetikKayıt Kaynak Kodu) (bağlantı verilebilir) isimli program Processing IDE kullanılarak açılır ve port numarası yine bir önceki etkinlikte olduğu gibi düzeltilir. İşlemler yapıldıktan sonra aşağıdaki resimde görülen kod bloğu Arduino karta yüklenir. Bu kod öğrencilere doğrudan verilebilir. Ancak kod verilirken SD kart ile ilgili komutların nasıl çalıştığı kod üzerinden anlatılmalıdır.



Resim 341. Örnek Program

Devre kurulup Arduino karta kod yüklendikten sonra "manyetikKayit" programı çalıştırılır. Aşağıdaki resimde olduğu gibi "manyetikKayit" programında iki ayrı buton yer almaktadır. Bu butonlardan soldaki "Evet", sağdaki ise "Hayır" butonudur. Bir manyetik cisim algılandıktan ve uzaklaştırıldıktan sonra, "Evet" butonuna tıklandığında Arduino kartına yüklenen kodun içindeki "if" bloğu çalışacak ve ilgili veri (manyetik alan okumasının yapıldığı tarih ve saat bilgisi) "kayit.txt" dosyasına yazdırılacaktır. "Hayır" butonuna basılırsa herhangi bir veri kaydı yapılmaz. Bir mıknatıs yardımıyla devre ve kod denenmelidir. Manyetik alan sensörüne bir mıknatıs yaklaştırıldığında sensör üzerinde yer alan led yanar. Mıknatıs geri çekilene kadar ışık yanmaya devam eder. Kod içinde yer alan birinci "Wait until" komutu, kodun çalışmasını mıknatıs uzaklaştırılana kadar bekletir. Mıknatıs uzaklaşınca ikinci "Wait until" komutu devreye girer. Bluetooth üzerinden bir komut gönderilene kadar kod yine komut bekletilir. Daha sonra yukarıda ifade edildiği gibi "Evet" butonu ile bilgiler dosyaya yazdırılır ya da "Hayır" butonu ile yazdırma işlemi iptal edilir. Aşağıdaki resimde ilgili bluetooth programı görülmektedir.

Resim 342. manyetikKayit Bluetooth Programı

Öğrenciler birkaç yazma işlemi yaptıktan sonra, SD kart modülden çıkarılmalı ve bilgisayara takılmalıdır. İçerisinde yer alan "kayit.txt" dosyası açılıp kayıt işlemi ile ilgili veriler öğrenciler tarafından kontrol edilmelidir. Kontrol işlemi tamamlandıktan sonra öğrencilere benzer şekilde hangi verilerin kaydının tutulabileceği sorusu yöneltilir ve sınıfta bir tartışma ortamı oluşturulur. Tartışmaya bu kayıt işlemlerinin ne işe yarayabileceği ve nerelerde kullanılabileceği soruları ile devam edilir.

2. TASARLA VE ÜRET

2.1 Tasarla ve Üret- Hangi Renkten Kaç Tane

Bu bölümdeki etkinliklerde bonibon rengine göre bonibonu kutunun içine alan veya bonibonu dışarı atan bir düzenek oluşturulmuştu. Bu etkinlikteki amaç ise aynı düzeneğin renkleri sayan ve bu renklerin sayısını Adafruit ekranı üzerinde yayınlayan devreyi ve kodları oluşturmaktır. Oluşturulan düzenek aşağıdaki özellikleri taşımalıdır;

i) Düzenek kırmızı, mavi, yeşil ve sarı renkli bonibonları kutunun içine almalıdır.

ii) Alınan bonibonların sayıları renklere göre sayılmalıdır. Yani kutu içerisinde hangi renkten kaç adet bonibon olduğu sayaçlar ile kontrol edilmelidir.

iii) Okuma işlemi Bluetooth üzerinden kontrol edilmelidir.

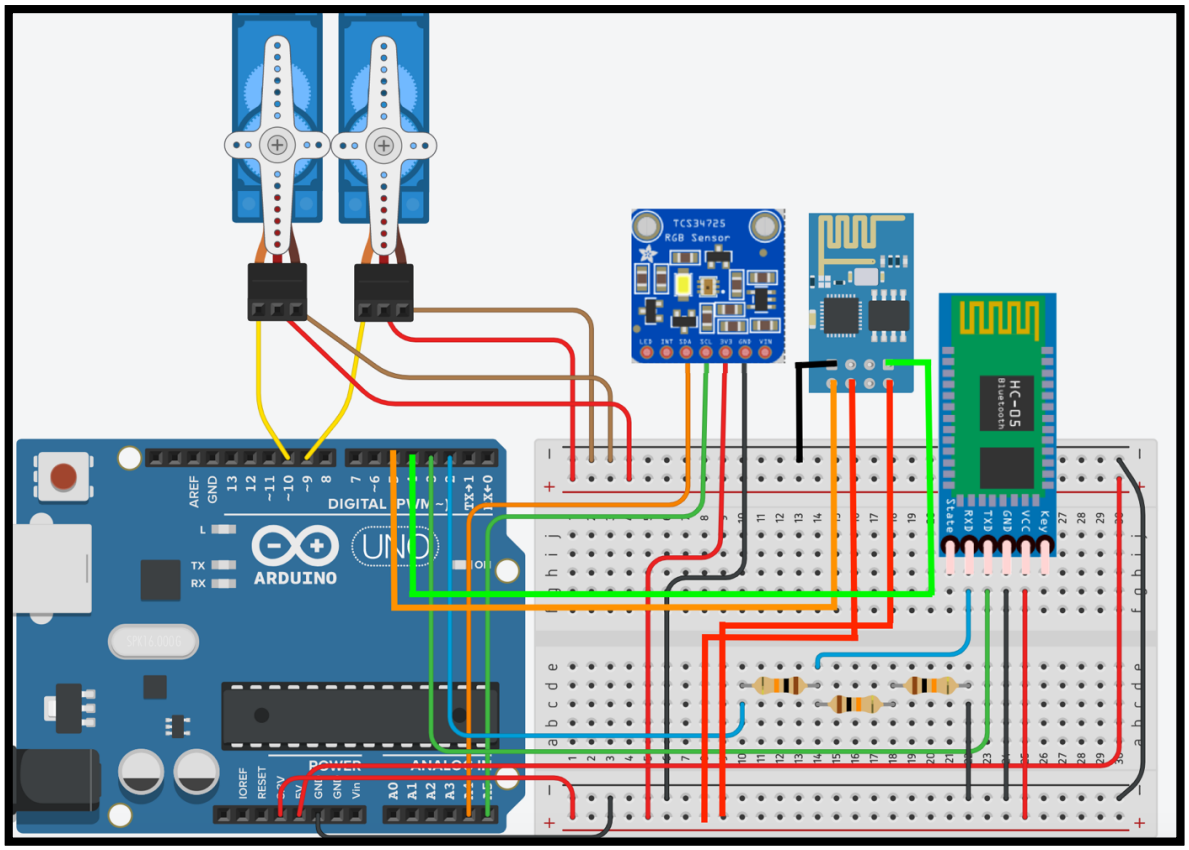
iv) Renklerin sayıları Adafruit ile oluşturulmuş ekranda ayrı ayrı gösterilmelidir.

v) Bluetooth aracılığı ile komut göndermek için bonibonSec programı kullanılmalıdır.

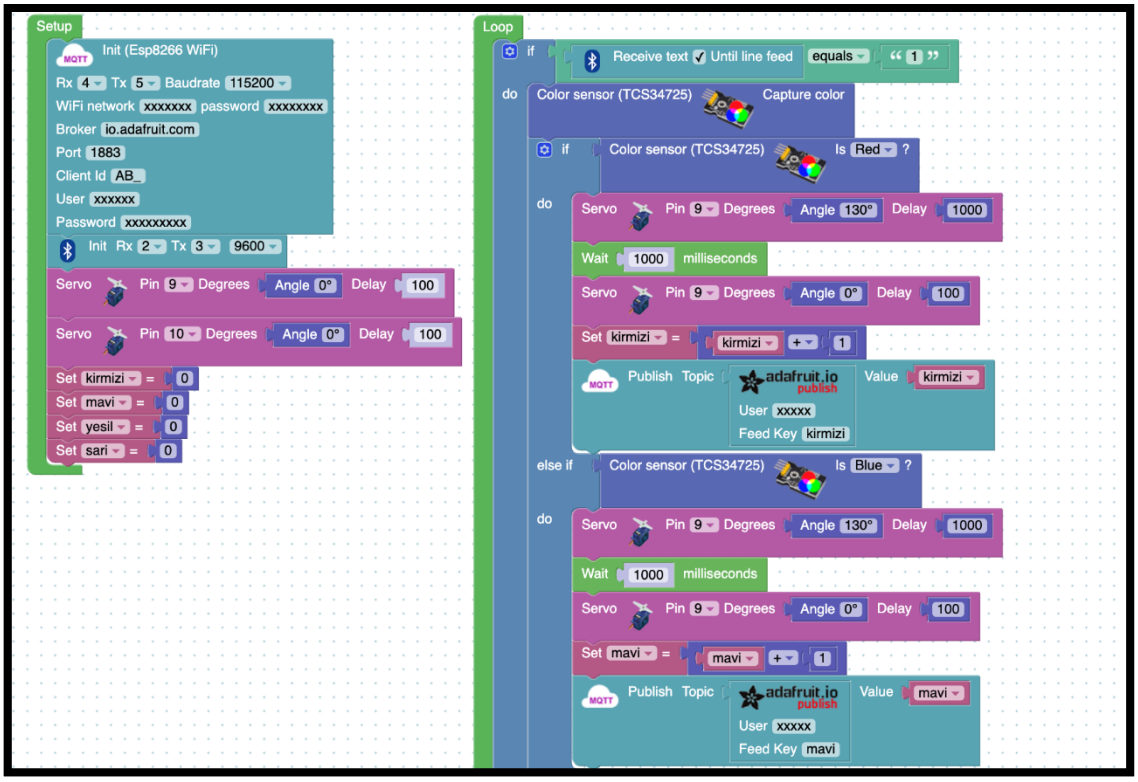
Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını

yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Bu görev için aşağıdaki resimde görülen örnek devre, program kodu ve Adafruit ekranı kullanılabilir.

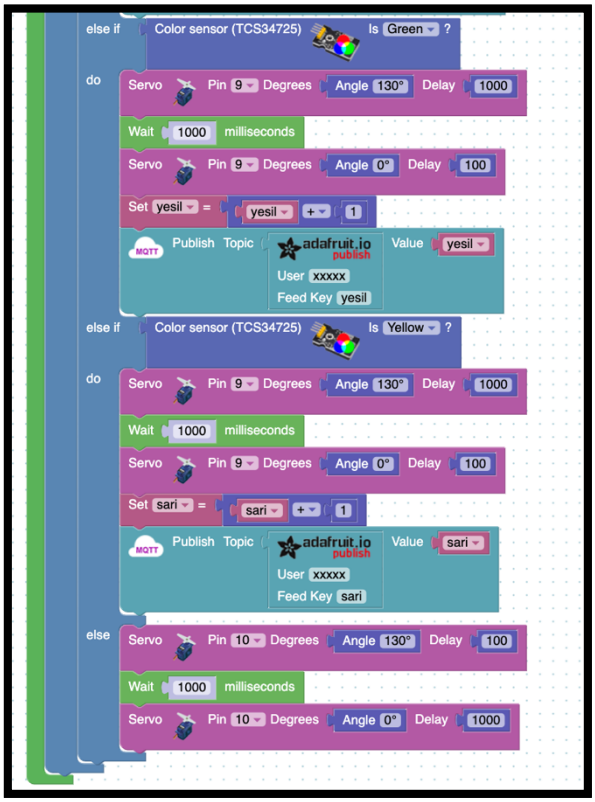
Örnek devre şeması:



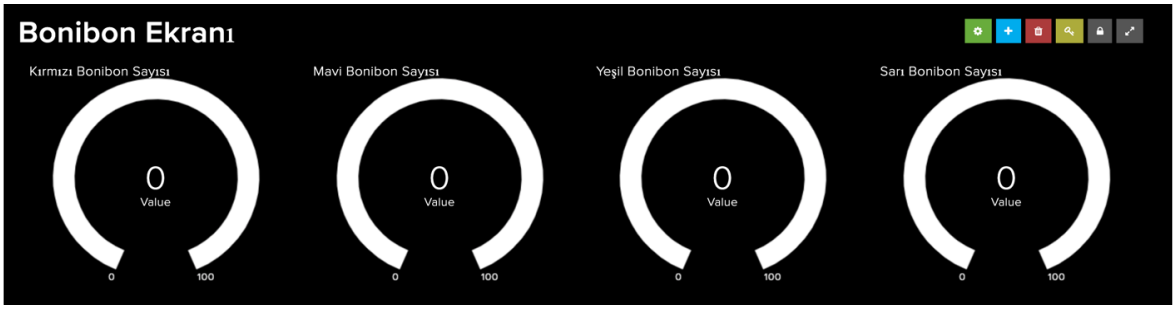
Resim 343. Örnek Devre Şeması



Resim 344. Örnek Program



Resim 345. Örnek Program (Devam)



Resim 346. Örnek AdaFruit Ekranı

3. DEĞERLENDİR

A. Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Bugün yapılan etkinliklerde bazı parçaların kendilerine özgü bağlantıları olduğunu gördük. Örneğin Saat modülü I2C bağlantısı, SD kart modülü ise SPI bağlantısını kullanmakta, Bluetooth modülü RX ve TX pinleri üzerinden bağlanmıştır. Bu bağlantılar ile özel bir bağlantı şekli olmayan araçlar arasında ne gibi farklılıklar var? Özel bağlantısı olan araçlar size bir sınırlama getiriyor ve işinizi zorlaştırıyor mu?

2. Renk sensörü ve manyetik alan sensörü hangi amaçlarla kullanılabilir? Günlük yaşamda kullanılabilecek alanlar nelerdir?

B. Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Bluetooth modülünün bağlantılarında zorlandım.
    - Servo motor ucuna takılan aparatları ayarlarken zorlandım.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Bonibon Sayılarını Dosyaya Yazdırıyorum

Bu etkinlikte amaç internet üzerinden Adafruit ekranına gönderilen bonibon sayılarını SD kart içinde oluşturulan bir text (metin) dosyasına yazdırmaktır. Her yeni bonibon eklenişinde tüm renklerdeki bonibon sayılarının güncel halinin dosyaya yazılması ve önceki bilgilerin silinmesi gerekmektedir. SD kart içindeki metin dosyası sadece güncel bilgileri tutmalıdır. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında

çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Proje Hazırlıyorum

Bu hafta itibariyle, proje çalışmalarına başlanması önerilmektedir. Projeye giriş süreci için planlamalar yapılmalıdır. Detaylar için "EK-10-Proje Hazırlıyorum' bölümü incelenebilir.

10. Bölüm-Metin Tabanlı Ortamda Elektronik Programlama

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol, kondansatör, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.
- Öğrenciler kızıl ötesi kumanda seti, RC522 RFID NFC kiti ve ESP8266 wifi modülü kullanarak IOT devreleri oluşturup programlayabilir.
- Öğrenciler sensörlerden alınan verilerin internete aktarımını sağlamak için gerekli elektronik devreleri oluşturup programlayabilir.
- Öğrenciler Nesnelerin İnternetinin temel kavramlarını bilerek uygulayabilir.

Bölümün Kazanımları:

- Öğrenciler Arduino kodlama ortamını kullanarak temel seviyede program yazıp çalıştırabilir.
- Değişkenleri, koşul ifadelerini, döngüleri ve iç içe döngüleri Arduino kodlama ortamında kullanabilir.
- Arduino kodlama ortamında PWM pinlerini kullanarak programlar oluşturabilir.
- Arduino kodlama ortamında RGB, buzzer ve LCD kullanımı ile ilgili komutları kullanabilir.
- Arduino kodlama ortamında RGB, buzzer ve LCD kullanarak açık uçlu problemleri çözebilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin oluşturdukları temel elektronik devreleri programlarken metin tabanlı Arduino Kodlama ortamını kullanabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, direnç (5 adet 220 ohm, 3 adet 330 ohm, 1 adet 100 ohm), 5 adet led, 1 adet RGB, 1 adet pasif buzzer/hoparlör, 4x20 Mavi LCD ekran- I2C dönüştürücü kartı (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Arduino kodlama ortamının tanıtılması ve Arduino blok ortamında değişken, koşul ifadesi, döngü ve iç içe döngü komutlarının çeşitli elektronik devreler üzerinden kullanımının gösterilmesi.

Uygula: Arduino kodlama ortamının kullanımı ve Arduino blok ortamında değişken, koşul ifadesi, döngü ve iç içe döngü komutlarının çeşitli elektronik devreler üzerinden kullanımı.

Tasarla: Arduino kodlama ortamında iç içe döngüler kullanılarak oluşturulacak led animasyonu için tanımlama ve fikir üretme adımlarının oluşturularak tasarımının yapılması.

Üret: Tasarımı yapılan led animasyonunun Arduino kodlama ortamı kullanılarak programlanması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

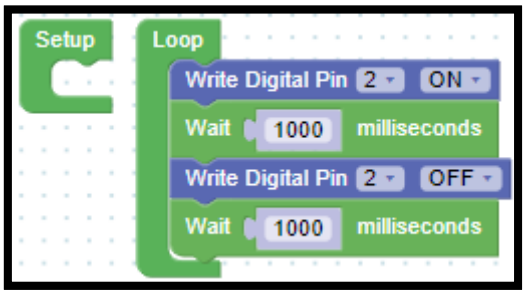
1. GÖZLE VE UYGULA

1.1 Gözle ve Uygula- Arduino IDE'ye Geçiş Yapıyorum

Malzeme Listesi:

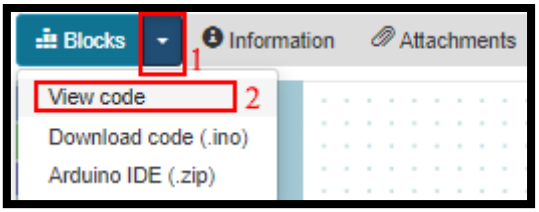
Arduino Uno
Breadboard
1 tane led
1 tane 220 Ohm direnç

Bu derste şimdiye kadar Arduino Uno kartı ArduinoBlocks ortamı ile kodlanmıştır. ArduinoBlocks ortamı çevrimiçi blok tabanlı bir ortamdır. Bu ortamda kodlar bloklar halinde hazır olarak sunulur ve programcı kodları kodlama alanına sürükleyerek ve düzenleyerek programlarını oluşturur. Bu tip ortamlar programlamaya giriş yapmak için ideal olsalar da belirli bir seviyeden sonra programcılar metin tabanlı kodlama ortamlarına ihtiyaç duyarlar. Her ne kadar ArduinoBlocks gibi blok tabanlı ortamlarda Arduino üzerinde programlar oluşturulabilse de Arduino'nun kendi programlama ortamı metin tabanlıdır. Dersin geri kalan kısmında Arduino kodlama ortamı kullanılarak programlar yazılacaktır. Arduino kodlama ortamına giriş yapmadan önce ArduinoBlocks ortamında yazılan kodların da aslında ArduinoBlocks tarafından metin tabanlı kodlara çevrildiğini göstermek yerinde olacaktır. Aşağıdaki resimde ArduinoBlocks ortamında iki numaralı pine takılı olan ledi bir saniye aralıklarla yakıp söndüren program görülmektedir.



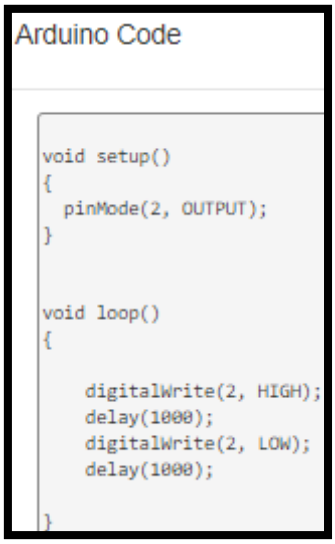
Resim 347. Örnek Program

Resimde görüldüğü gibi temel olarak iki blok vardır. Bunlar "Setup ve Loop" bloklarıdır. Bu örnekte "Setup" bloğunun içerisinde herhangi bir kod bulunmamaktadır. "Loop" bloğu içerisinde ise iki numaralı pine bağlanmış ledi yakıp söndürmek için gerekli kodlar bulunmaktadır. Setup bloğu programın Arduino kartına yüklendiğinde bir defaya mahsus olarak çalıştırılacak kodları içerir. Loop bloğu ise Setup bloğundaki kodların çalıştırılmasının ardından programın Arduino kartında çalıştığı müddetçe tekrarlanacak komutları içerir. Yani program Arduino Uno karta yüklendiğinde önce Setup bloğunda bulunan kodlar bir kere çalıştırılır ve ardından Loop bloğunda bulunan komutlar sürekli (karta enerji verildiği süre boyunca) tekrarlanır. Aslında ArduinoBlocks ortamında içerisi boş görünen Setup bloğunun içi tam anlamıyla boş değildir. Loop bloğu içerisinden yazma işlemi yapılan 2 numaralı pinin öncelikle yazma işlemi için tanımlanması gerekir. ArduinoBlocks ortamı, arka planda programcı için Loop bloğu içerisinde iki numaralı dijital pine ON ve OFF bilgisi gönderebilmek için Setup bloğu içerisinde bu pinin dijital çıktı (OUT) işlemleri için kullanılacağının tanımlamasını yapar. Bu durumu gözlemlemek için "Blocks" başlığının sağında bulunan açılır menü sembolü olan aşağı ok tuşuna tıklanır ve açılan menüden aşağıdaki resimde görüldüğü gibi "View code" tıklanır.



Resim 348. ArduinoBlocks Kodlarının Görüntülenmesi

Açılan pencerede ArduinoBlocks ortamında yazılan kodun Arduino karşılığı gösterilir. Aşağıdaki resimde görüldüğü gibi ArduinoBlocks ortamında yazılan kodun metin tabanlı karşılığı olan Arduino kodları otomatik olarak oluşturulmuştur. Bu kodlara bakıldığında, kodların temel olarak setup ve loop isimli kısımlardan oluştuğu görülür. Setup ve loop aslında birer fonksiyondur. Bunlara fonksiyon isminin verildiğini ve fonksiyonlarda kullanılacak komutların süslü parantezler içerisine yazıldığını bilmek şimdilik yeterli olacaktır. Fonksiyonların Arduino ortamında nasıl tanımlandığı ve kullanıldığı konusu şimdilik anlatılmamalıdır.



Resim 349. Setup ve Loop Fonksiyonları

ArduinoBlocks ortamında bulunan setup ve loop fonksiyonları Arduino kodlama kısmında da bulunmaktadır. Bir Arduino kodlama programında bu iki fonksiyon mutlaka bulunmalıdır. Arduinoblocks ortamında boş görünen setup fonksiyonu Arduino kodlama ortamında boş değildir. Aslında ArduinoBlocks ortamı buraya gerekli kodu programcı için arka planda yazmıştır. Setup fonksiyonu içerisinde "pinMode(2, OUTPUT);" komutu bulunmaktadır. Hatırlandığı gibi Arduino kartında bulunan pinler dijital çıktı, dijital girdi, analog girdi veya PWM çıktı (analog çıktı benzeri) için kullanılabilir. pinMode komutu Arduino kartında bulunan pinlerin hangi amaçla kullanılacağını bildirir. Örneğin pinMode(2, OUTPUT) komutu ile iki numaralı pinin çıktı (OUTPUT) için kullanılacağı bildirilmiştir. Her Arduino programında bir pin kullanılmadan önce o pinin hangi amaçla kullanılacağı pinMode komutuyla Arduino programlama ortamına bildirilmelidir. Görüldüğü gibi pinMode komut satırının sonunda ";" işareti bulunmaktadır. Arduino ortamında komutların sonuna ";" işareti konulur. Eğer bu işaret unutulursa kodlar çalışmaz ve Arduino kodlama ortamı hata mesajı verir.

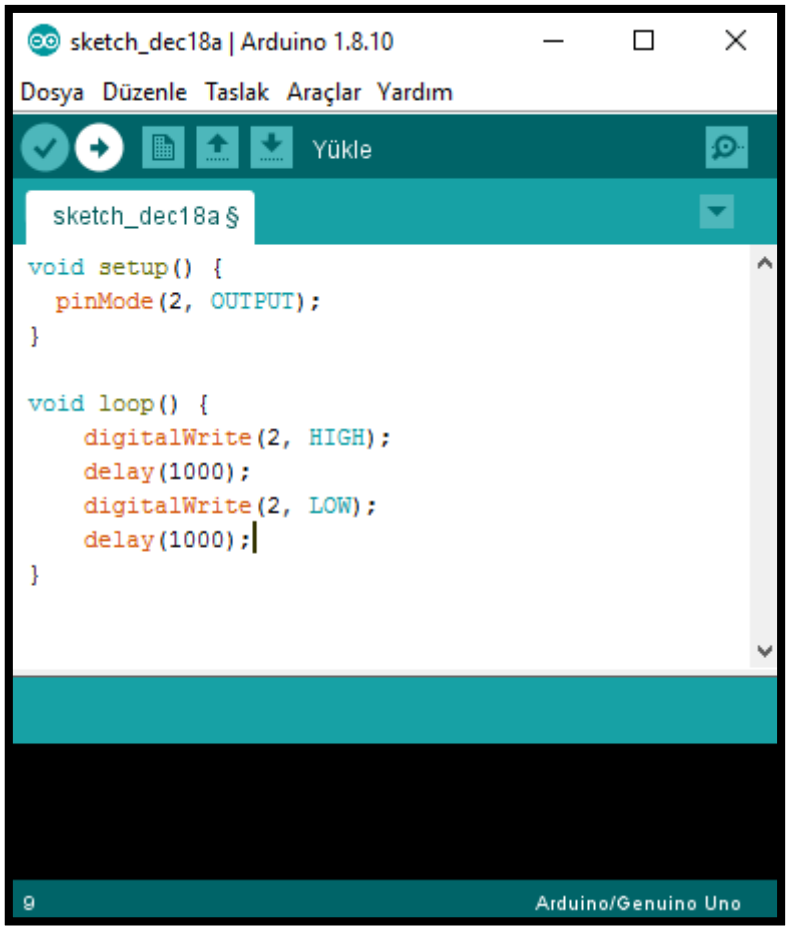
Arduino kartında sürekli çalışmasını istediğimiz kodlar loop bloğu içerisinde yazılır. Burada bulunan "digitalWrite(2, HIGH);" komutu iki numaralı pine dijital çıktı olarak HIGH (ON veya 5V) değerinin gönderileceğini bildirir. Yani burada bulunan led yakılacaktır. "delay(1000);" komutu ise bir saniye (1000 mili saniye) boyunca yeni bir işlem yapılmadan beklenmesini sağlar. "digitalWrite(2, LOW)"

komutu ise iki numaralı pine LOW (OFF veya 0V) değerinin gönderileceğini bildirir. Sonuç olarak bu kodlar çalıştırıldığında iki numaralı pine bağlı led bir saniye aralıklarla yanıp söner.

Dikkat

Arduino kodlama ortamında kod yazarken dikkat edilmesi gereken çeşitli hususlar bulunmaktadır. Bunlardan birisi kodların yazımında büyük harf ve küçük harflerin yer değiştiremeyeceğidir. "pinMode" komutu "pinmode" olarak yazılamaz. Benzer şekilde "digitalWrite" komutu "digitalwrite" veya "DigitalWrite" şeklinde yazılamaz.

ArduinoBlocks ortamında oluşturulan kodlar aynen kopyalanıp Arduino kodlama ortamında aktarılabilir ve burada çalıştırılabilir. Bu görevi yerine getirmek için Arduino kodlama ortamı açılmalıdır. Aşağıdaki resimde gösterildiği gibi setup ve loop fonksiyonları içerisindeki komutlar aktarılmıştır.

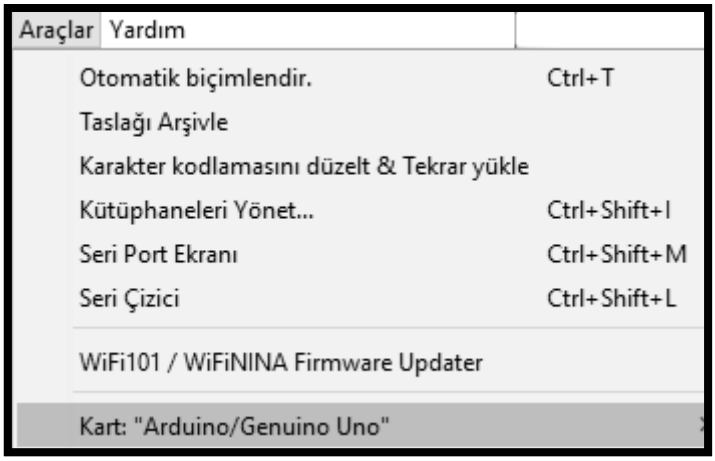


Resim 350. Arduino Kodlama Ortamı

Dikkat

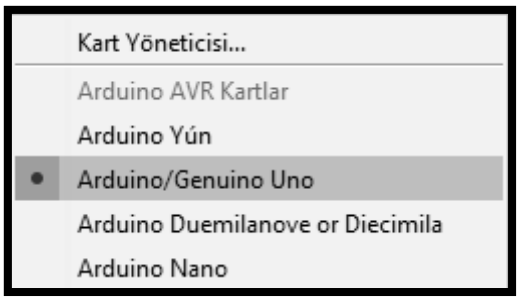
ArduinoBlocks ortamındaki belirli kütüphanelere ait komutları Arduino kodlama ortamında çalıştırmak için önce o kütüphanenin Arduino kodlama ortamına yüklenmiş olması gerekmektedir. Kütüphaneler yüklenmeden bu komutlar çalıştırılamaz. Çalıştırılmaya çalışılırsa Arduino kodlama ortamı hata verir.

Arduino kodlama ortamında yazılmış bir kod Arduino karta yüklenmeden önce kullanılan kartın çeşidi bildirilmelidir. Bunun için aşağıdaki resimde görüldüğü gibi "Araçlar" menüsündeki "Kart:" seçeneğine tıklanır.



Resim 351. Arduino Kart Ayarları

Açılan menü içerisinden aşağıdaki resimde görüldüğü gibi "Arduino/Genuino Uno" maddesi seçilir. Böylece Arduino ortamına Arduino Uno kartı tanıtılmış olur.

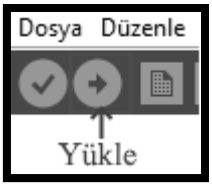


Resim 352. Arduino kart ayarları

Arduino kartının çeşidi Arduino kodlama ortamına bildirildikten sonra Arduino kartının hangi porta bağlı olduğu da bildirilmelidir. Bu iş için "Araçlar" menüsünden aşağıdaki resimde görüldüğü gibi "Port" maddesi kullanılarak Arduino kartın hangi porta bağlı olduğu seçilir.

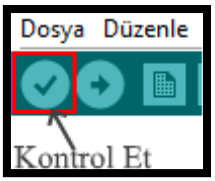
Resim 353. Arduino Port Ayarları

Artık Arduino komutları çalıştırılabilir. Arduino komutlarını çalıştırmak için aşağıdaki resimde görüldüğü gibi "Yükle" butonuna tıklanır. Bu butona tıklandığında kodlar Arduino karta yüklenir ve çalıştırılır.



Resim 354 Kod Yükleme Butonu

Arduino karta yazılan programı yüklemeden, programda hata olup olmadığı kontrol edilmek istendiğinde "Yükle" butonunun solunda bulunan ve resimde görülen "Kontrol Et" butonuna tıklanır. Bu butona tıklandığında yazılan programda eğer yazım yanlışı varsa Arduino kodlama ortamı tespit ettiği hatayı bildirir. Burada öğrencilere yazılan programın doğru olup olmadığının kontrolü değil, yazım hataları olup olmadığının kontrolünün yapıldığı ifade edilmelidir.



Resim 355. Kontrol Et Butonu

Rehber öğretmen öğrencilere yukarıda anlatılan süreci detaylı olarak gösterir. Rehber öğretmen süreci anlatırken öğrenciler anlatılanları Arduino kodlama ortamını kullanarak uygular.

1.1 Uygula- Arduino Ortamında Flip Flop

Malzeme Listesi:

Arduino Uno
Breadboard
2 adet led
2 adet 220 Ohm direnç

Bu etkinlikte amaç Arduino ortamında "flip-flop" kodu yazmaktır. "Flip flop" şu şekilde çalışır:

(i) Birinci led yanacaktır,

(ii) 1 saniye sonra birinci led sönecektir,

(iii) 1 saniye sonra ikinci led yanacaktır,

(iii) 1 saniye sonra ikinci led sönecektir,

(iv) 1 saniye beklenecektir

(v) i-iv adımları sürekli tekrar edecektir.

Bu görev için aşağıdaki kod kullanılabilir. Öğrenciler aşağıdaki kodu Arduino kodlama ortamında kendileri yazmalıdır. Zorlanan öğrenciler eğer kodu yazamazlarsa ArduinoBlocks ortamındaki "View Code" seçeneğinden faydalanabilirler. Öğrenciler bu özellikten faydalanarak kodu yazmaları durumunda aynı kodu yeniden "View Code" özelliğini kullanmadan kendileri yazmalıdır. Aşağıda verilen kodda "//" işaretinden sonra o satır ile ilgili açıklama yer almıştır. Bu açıklama programın çalışmasını etkilememektedir.

```
void setup() {
  pinMode(2, OUTPUT); //2 numaralı pin - birinci led - dijital çıktı
```

```
  pinMode(3, OUTPUT);  //3 numaralı pin - ikinci led - dijital çıktı
}
void loop() {
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   digitalWrite(3, HIGH);
   delay(1000);
   digitalWrite(3, LOW);
   delay(1000);
}
```

1.2 Gözle- Tam Sayı Değişkenleri ve Döngüler ile Led Animasyonu

Malzeme Listesi:

Arduino Uno
Breadboard
2 adet led
2 adet 220 Ohm direnç

Bu etkinlikte amaç birinci ledi 5 kere yakıp söndürmek ve bunun ardından ikinci ledi beş kere yakıp söndürmektir. İki ledin sırayla 5'er kere yanması işlemi başa dönerek sürekli tekrar edecektir. Bu iş için aşağıdaki kod kullanılabilir. Rehber öğretmen aşağıdaki kodu öğrencilere gösterir ve kodun çalışacağını fakat yazmasının uzun ve zahmetli olduğunu anlatır. Aşağıdaki kodda tekrarlanan bölümler bulunmaktadır. Rehber öğretmen öğrencilere bu tip durumlarda döngülerin kullanılabileceğini anlatır.

```
void setup() {
  pinMode(2, OUTPUT); //2 numaralı pin - birinci led
  pinMode(3, OUTPUT);  //3 numaralı pin - ikinci led
}
void loop() {
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
```

```
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(3,LOW);
  delay(1000);
}
```

Yukarıda yazılan kod döngüler kullanılarak yeniden yazılacaktır fakat önce kısaca değişken ve döngülerden bahsedilmesi gerekmektedir.

Değişkenler: Daha önce anlatıldığı gibi programcılar belirli değerleri mikrodenetleyici/bilgisayar hafızasında saklamak isterler. Bu değerler değişken ismi verilen hafıza birimlerinde saklanır. Farklı çeşitlerde değişkenler olabilir. Burada tam sayı değişkenlerinden bahsedilecektir. Tam sayı değişkeni tanımlamak için aşağıdaki iki yöntem de kullanılabilir. Birinci yöntemde önce değişken tanımlanmış ve ardından programın ilerleyen bölümlerinde değişkene ilk değeri atanmıştır. İkinci örnekte değişken tanımlanırken ilk değer ataması da yapılmıştır. Yani değişken tanımlama ve değişkene ilk değer atama aynı satırda yapılabileceği gibi farklı satırlarda da yapılabilir.

Farklı satırlarda	**Aynı satırlarda**

int deg_ismi;
…
deg_ismi=n;
 	
Örnek:
int sayac;
…
sayac=0;
	int deg_ismi=n;
	
	Örnek:
	int sayac=0;

For Döngüsü: Programcılar tekrarlanan görevleri yerine getirmek için döngüleri kullanır. Farklı çeşitte döngüler vardır. Burada "for" döngüsünden bahsedilecektir. For döngüsünde döngünün başlangıcı, bitişi ve başlangıç ile bitiş arasında kaç adım atılacağı bellidir. Örneğin bir ledin bir saniye aralıklarla beş defa yanıp sönmesi isteniyor olsun. Burada başlangıç bir, bitiş beş ve başlangıç ile bitiş arasında her seferde atılması gereken adım sayısı birdir. Bunun için aşağıdaki kod kullanılabilir. Kodda for döngüsü içerisinde öncelikle "i" isimli bir değişken tanımlanır ve buna ilk değer olarak 1 değeri verilir. For ifadesinden sonra gelen (int i=1; i<=5; i++) ifadesi, "i" değişkeninin 1'den başlayacağını (i=1), "i" değişkeninin beşe kadar ilerleyeceğini (i<=5) ve "i" değişkeninin her defasında birer artırılacağını (i++)

söylemektedir. Her bir "i" değeri için döngünün gövdesindeki komutlar çalıştırılır. Yani döngünün gövdesindeki komutlar i=1, i=2, i=3, i=4 ve i=5 için beş defa çalıştırılır.

```
for(int i=1; int<=5; i++){
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
}
```

Aynı kod aşağıdaki gibi de yazılabilir. Burada "i" değişkeni döngünün dışında tanımlanmıştır ve "i" değişkeni ilk değeri olan 1'i döngü içerisinde almıştır. İki kod da aynı şekilde çalışır. Fakat yukardaki koddaki "i" değişkenine döngü dışından erişilemezken. Aşağıdaki koddaki "i" değişkenine döngü dışından da erişilebilir. Bu fark şimdilik önemli değildir ve rehber öğretmen tarafından fazlaca vurgulanmamalıdır.

```
 int i;
 for (i=1;i<=5;i++){
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
   }
```

Etkinlikte aşağıda verilen program kullanılabilir. Görüldüğü gibi aynı görev çok daha az satır kod yazılarak yapılmıştır. Bu yüzden bu görevi döngüleri kullanarak yapmak daha yerinde olacaktır. İlk yazılan (döngüsüz) ve ikinci yazılan (döngülü) kod hız açısından karşılaştırıldığında aralarında fark bulunmamaktadır. Yani birinci ve ikinci kod aynı hızda çalışır. Birinin diğerinden daha kısa şekilde yazılması daha hızlı çalışacağı anlamına gelmez.

```
void setup() {
 pinMode(2, OUTPUT); //2 numaralı pin - birinci led
 pinMode(3,OUTPUT);  //3 numaralı pin - ikinci led
}
void loop() {
 int i;
 for (i=1; i<=5; i++){
   digitalWrite(2, HIGH);
   delay(1000);
   digitalWrite(2, LOW);
   delay(1000);
 }
 for(int j=1 ;j<=5; j++){
   digitalWrite(3,HIGH);
   delay(1000);
   digitalWrite(3,LOW);
   delay(1000);
 }
}
```

1.2 Uygula- Soldan Sağa Akan 5 Led'li Animasyon

Malzeme Listesi:

Arduino Uno
Breadboard
5 adet led
5 adet 220 Ohm direnç

Bu etkinlikte amaç yan yana bulunan 5 ledin soldan sağa doğru akıyormuş görüntüsünü veren animasyon kodunu yazmaktır. Bu iş için aşağıdaki kod kullanılabilir.

```
for (int i=1; i<=5; i++){
  digitalWrite(i,HIGH);
  delay(500);
  digitalWrite(i,LOW);
  delay(500);
  }
```

1.3 Gözle ve Uygula- Soldan-Sağa ve Sağdan-Sola Akan 5'li Led

Malzeme Listesi:

Arduino Uno
Breadboard
5 adet led
5 adet 220 Ohm direnç

Bu etkinlikte amaç soldan sağa ve ardından sağdan sola doğru akacak 5 led'li animasyon yapmaktır. Yukarıdaki etkinlikte soldan sağa doğru akan animasyon yapılmıştır. Sağdan sola akan animasyon yapmak için aşağıdaki kod kullanılabilir. Aşağıdaki kod ile görüldüğü gibi döngü değişkeni azalarak da ilerleyebilir. Aşağıdaki döngünün başlangıç değeri 5, bitiş değeri ise 1'dir. Değişkenin değerini birer birer azaltmak için "i—" (i=i-1) ifadesi kullanılmıştır. Burada "i--"ifadesinin solunda bulunan "i>=1" ifadesine de dikkat etmek gerekir. "i>=1" ifadesi döngü değişkeni olan i'nin 1'den büyük veya eşit olduğu sürece çalışması gerektiğini söyler. Bu sayede döngü değişkeninin değeri 5'ten başlar ve 1'e kadar devam eder.

```
for (int i=5; i>=1; i--){
  digitalWrite(i,HIGH);
  delay(500);
  digitalWrite(i,LOW);
  delay(500);
  }
```

Rehber öğretmen soldan sağa akan led kodunu sağdan sola akan led kodu ile birleştirerek devreyi çalıştırır. Öğrencilerin kodun çalışmasını gözlemesini sağlar. Ardından öğrencilerden bu kodu kendilerinin yeniden yazmasını ister.

1.4 Gözle ve Uygula- Koşul ile Animasyonları Birleştiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
5 adet led
5 adet 220 Ohm direnç

Bu etkinlikte 5 adet led kullanılarak birden fazla animasyon arka arkaya oluşturulacaktır. Bu animasyonlar aşağıdaki şekildedir.

(i) Soldan sağa akan animasyon (sırasıyla 1, 2, 3, 4, 5),

(ii) Bütün ledleri iki kere yanıp söndüren animasyon,

(iii) Soldan sağa tek ledlerde akan animasyon (sırasıyla 1, 3, 5),

(iv) Bütün ledleri iki kere yanıp söndüren animasyon,

(v) Soldan sağa akan animasyon (sırasıyla 1, 2, 3, 4, 5),

(vi) Bütün ledleri iki kere yanıp söndüren animasyon,

(vii) Soldan sağa çift ledlerde akan animasyon (sırasıyla 2, 4),

(viii) Bütün ledleri iki kere yanıp söndüren animasyon.

Bu görev için gerekli kod aşağıda verilmiştir. Bu kodda if - else if – else koşul yapısı kullanılmıştır. Rehber öğretmen kodu açıklamaya başlamadan önce bu yapıyı öğrencilere açıklar. Öğrencilere kodu anlattıktan sonra öğrencilerin kodu kendilerinin yazmasını sağlar.

```
int sayac=1;
void setup() {
  pinMode(1, OUTPUT); //1. led
  pinMode(2, OUTPUT); //2. led
  pinMode(3, OUTPUT); //3. led
  pinMode(4, OUTPUT); //4. led
  pinMode(5, OUTPUT); //5. led
}
void loop() {
  if (sayac==1 || sayac==5){
    for (int i=1; i<=5; i++){
      digitalWrite(i, HIGH);
      delay(200);
      digitalWrite(i, LOW);
      delay(200);
      }
    }
  else if(sayac==3){
    for(int i=1;i<=5;i=i+2){
      digitalWrite(i, HIGH);
      delay(200);
      digitalWrite(i, LOW);
      delay(200);
    }
    }
    else if(sayac==7){
    for(int i=2;i<=4;i=i+2){
      digitalWrite(i, HIGH);
      delay(200);
      digitalWrite(i, LOW);
      delay(200);
```

```
    }
    }
    else{
    for(int k=1;k<=2;k++){
      for(int i=1;i<=5;i++){
        digitalWrite(i,HIGH);
        }
      delay(200);
      for(int i=1;i<=5;i++){
        digitalWrite(i,LOW);
        }
      delay(200);
    }
  }
  sayac++;
  if(sayac==9){
    sayac=1;
  }
}
```

Dikkat

Bu kod mod operatörü kullanılarak da yapılabilir. Fakat ortaokul öğrencileri henüz modüler aritmetik dersini görmediği için bu şekilde tasarlanmıştır.

1.5 Gözle- Arduino Ortamında PWM Pini Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet led
1 adet 220 Ohm direnç

Daha önceden anlatıldığı gibi Arduino Uno kartlardan analog çıktı alınamaz. Fakat PWM pinleri sayesinde 0-255 arasındaki değerler çıktı olarak verilip analog çıktı benzeri bir etki yaratılabilir. Bu iş için Arduino kodlama ortamında analogWrite komutu kullanılır. analogWrite komutunun kullanımı şu şekildedir:

analogWrite(PWM_pini, gönderilecek değer);

Örnek: analogWrite(3, 127); // kod satırı ile 3 numaralı PWM pinine 127 bilgisi gönderilmiştir.

Aşağıdaki program ile üç numaralı PWM pinine bağlanmış bir ledin parlaklığı giderek artırılacak ve söndürülecektir. Bu işlem sürekli tekrar edilecektir.

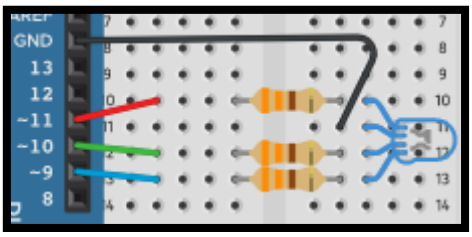
```
void setup() {
  pinMode(3,OUTPUT);
}
void loop() {
  for (int i=0;i<=255;i=i+5){
    analogWrite(3,i);
    delay(30);
    }
}
```

1.6 Gözle- PWM ile RGB Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB
3 adet 330 Ohm direnç

Bu etkinlikte amaç PWM pinlerini ve analogWrite komutunu birlikte kullanarak RGB'den farklı renkler elde etmektir. Hatırlanacağı gibi RGB led'lerin ortak "anot" veya ortak "katot" olmak üzere iki farklı çeşidi bulunmaktadır. Aşağıdaki resimde ortak katot bir RGB'nin bağlantısı görülmektedir. Görüldüğü üzere ortak bacak GND pinine ve kırmızı bacağı 11 numaralı pine, yeşil bacağı 10 numaralı pine, mavi bacağı ise 9 numaralı pine bağlanmıştır.



Resim 356. Örnek Devre Şeması

Yukarıda kullanılan her bir PWM pinine "analogWrite" komutu kullanılarak 0-255 arasında bir değer verilebilir. Örneğin RGB'den kırmızı renk almak için RGB'nin kırmızı bacağının bağlı olduğu 11 numaralı pine 255 ve diğer pinlere de 0 değeri gönderilir. Bu iş için aşağıdaki program kullanılabilir.

```
void setup() {
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop() {
  analogWrite(11,255);
  analogWrite(10, 0);
  analogWrite(9, 0);
}
```

Ortak katot bir RGB'de herhangi bir rengin koyuluğu artırılmak isteniyorsa o pine 255 veya 255'e yakın değerler gönderilir. O rengin koyuluğu azaltılmak isteniyorsa 0 veya 0'a yakın değerler gönderilir. Örneğin RGB ile oluşturulacak bir renkte kırmızı rengin fazla, yeşil rengin orta ve mavi rengin az koyulukta olması isteniyorsa kırmızı renk pinine 255'e yakın bir değer, yeşil renk pinine 127'ye yakın bir değer, mavi renk pinine 0'a yakın bir değer gönderilir. Bu iş için aşağıdaki kod kullanılabilir. Bu şekilde kırmızı, yeşil ve mavi renkler farklı tonlarda karıştırılarak istenilen renkler elde edilebilir.

```
  analogWrite(11, 255);
  analogWrite(10, 127);
  analogWrite(9, 30);
```

Ortak anot RGB'de bağlantı ve kullanılan parametreler ortak katot RGB'ye göre farklıdır. Aşağıdaki resimde ortak anot RGB'nin bağlantısı gösterilmiştir.

Resim 357. Örnek Devre Şeması

Ortak anot bir RGB'de renkler için kullanılan koyuluk değerleri ortak katot RGB'nin tersidir. Yani ortak anot RGB'de koyuluğu artırmak için 0 veya 0'a yakın değerler verilir. Örneğin kırmızı ışık yakmak için 11 numaralı pine 0 değeri verilirken, 10 ve 9 numaralı pinlere 255 değeri verilmelidir. Ortak anot RGB ile oluşturulacak bir renkte kırmızı rengin fazla, yeşil rengin orta ve mavi rengin az koyulukta olması isteniyorsa kırmızı renk pinine 0'a yakın bir değer, yeşil renk pinine 127'ye yakın bir değer, mavi renk pinine 255'e yakın bir değer gönderilir. Bu görev için aşağıdaki kod kullanılabilir.

```
analogWrite(11, 30);
analogWrite(10, 127);
analogWrite(9, 255);
```

1.6 Uygula- Maviden Kırmızıya, Kırmızıdan Yeşile ve Yeşilden Maviye

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet RGB
3 adet 330 Ohm direnç

Bu etkinlikte RGB kullanılarak önce mavi renk bir müddet yakılacaktır. Ardından mavi rengin yoğunluğu azaltılarak kırmızı rengin yoğunluğu artırılacaktır. Ara geçiş renklerinden sonra kırmızı renk bir süreliğine yakılacaktır. Ardından kırmızının yoğunluğu azaltılarak yeşil rengin yoğunluğu artırılacaktır. Yeşil renk bir müddet yanık kalacaktır. Ardından yeşil rengin yoğunluğu azaltılacak ve mavi rengin yoğunluğu artırılacaktır. Mavi renk tamamen yandığında bir müddet açık kalacaktır. Bu işlemler sürekli tekrarlanacaktır. Bu görev için aşağıdaki kod bloğu kullanılabilir (Bu kod için RGB'nin kırmızı bacağı 11, yeşil bacağı 10 ve mavi bacağı 9 numaralı dijital pine bağlanmıştır).

```
void loop() {
 analogWrite(11, 255);
 analogWrite(10, 255);
 analogWrite(9, 0);
 delay(1000);
 for (int i=0;i<=255;i++){
  analogWrite(9,i);
  analogWrite(11, 255-i);
  delay(30);
 }
 delay(1000);
 for(int i=0;i<=255;i++){
  analogWrite(11, i);
  analogWrite(10, 255-i);
  delay(30);
 }
```

```
  delay(1000);
  for(int i=0; i<=255; i++){
    analogWrite(10, i);
    analogWrite(9, 255-i);
    delay(30);
  }
}
```

1.7 Gözle ve Uygula- Basit Ambulans Sireni: Buzzer ve Hoparlör Kullanımı

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet 100 Ohm direnç
1 adet pasif buzzer/hoparlör

Önceki bölümlerde buzzer ile ilgili detaylar anlatılmıştı. Daha önce söylenildiği gibi pasif buzzerda farklı frekanslarda farklı süreler için sesler çalınabilir. Aynı işlem hoparlörler kullanılarak da yapılabilir. Hoparlörler buzzerlara göre daha kaliteli ses üretirler. Fakat hoparlörler buzzer'a göre boyut olarak büyüktür ve maliyeti daha fazladır. Maliyet, ses kalitesi ve boyut değişkenleri düşünülerek buzzer veya hoparlör seçimi yapılabilir. Arduino kodlama ortamında hoparlörden/buzzerdan farklı notalarda ses çalmak için tone komutu kullanılır. "noTone" komutu ise çalınan sesi durdurmak için kullanılır. "Tone" komutunun kullanımı aşağıdaki şekildedir:

tone(pin_numarasi, çalınacak_frekans);

Örnek:

tone (2,500); // 2 numaralı dijital pinden 500 Hertz ses çalar

Tone komutu yukarıda örneklendiği gibi pin numarası ve çalınacak sesin frekansı verilerek iki parametreli çalıştırıldığında yeni bir tone/noTone komutu kullanılana kadar veya program bitene kadar aynı frekansı çalmaya devam eder. Tone komutunun üç parametreli kullanımı da bulunmaktadır. Tone komutu üç parametreli kullanıldığında ise, noTone komutunu kullanmaya gerek kalmadan, istenilen müddet boyunca nota çalınıp durdurulabilir. Bu kullanım aşağıdaki gösterilmiştir:

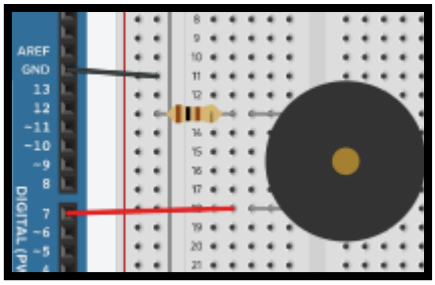
Tone(pin_numarası,çalınacak_frekans, çalma_süresi)

Örnek:

```
tone(2, 500, 500);
delay(1000);
```

Yukarıdaki örnekte 500 Hertz ses yarım saniye boyunca çalıp ardından yarım saniye boyunca susacaktır. Bu örnekte delay komutu 1 saniye boyunca bekleme yapmasına rağmen buzzer/hoparlör yarım saniye boyunca susmuştur. Burada bir hata yoktur. Delay komutunun yaptığı beklemenin içerisine tone komutunun çaldığı ses de dahildir. Başka bir deyişle tone komutu çalıştıktan hemen sonra daha çalma işlemi sürerken alt satırda yer alan delay komutu da çalışmaya başlar. Yani bekleme ve çalma toplamda bir saniye sürecektir. Çalma yarım saniye sürdüğü için beklemeye de yarım saniye kalır.

Bu etkinlikte basit bir ambulans sesi çalınacaktır. Bu görev için pasif buzzer aşağıda yer alan resimdeki gibi bağlanır. Buzzer/hoparlörle birlikte 100 Ohm direnç kullanılmıştır. Direncin görevi olası fazla akımın buzzera/hoparlöre zarar vermesini engellemektir.



Resim 358. Örnek Devre Şeması

Bu etkinlik için aşağıdaki kod kullanılabilir. Rehber öğretmen bu kodu öğrencilere anlatarak yazar. Ardından öğrencilerin aynı kodu kendilerinin yazmasını ve çalıştırmasını sağlar. Son olarak öğrenciler buzzer yerine hoparlör kullanarak aynı kodu çalıştırır. Hoparlörün devre bağlantısı buzzer ile aynıdır. Buzzer yerinden çıkarılarak hoparlörün kırmızı renkli kablosu 7 numaralı pine, siyah renkli kablosu ise direncin açık bulunan bacağına bağlanır.

```
void setup() {
  analogWrite(9,255);
}
void loop() {
  tone(7,500);
  delay(500);
  tone(6,800);
  delay(500);
}
```

Dikkat

3 ve 11 numaralı PWM pinleri tone komutu ile kullanıldığında hatalı sonuçlar üretebilir. Bu nedenle Tone komutu kullanıldığı zaman 3 ve 11 numaralı PWM pinlerinin kullanımı tercih edilmemelidir.

1.8 Gözle ve Uygula- Gelişmiş Ambulans Sireni

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet 100 Ohm direnç
1 adet hoparlör/buzzer

Bu etkinlikte amaç ambulansların çıkardığı sese daha yakın bir ses çıkaran kodu yazmaktır. Bu görev için aşağıdaki kod kullanılabilir. Rehber öğretmen aşağıdaki kodu açıklayarak tahtada yazar. Rehber öğretmen kodu anlatırken iç içe döngü kullanıldığından bahseder ve iç içe döngü kullanımının nasıl olması gerektiğini anlatır. Ardından öğrencilerin aynı kodu yazıp çalıştırmalarını sağlar (Not: bu etkinlikte hoparlör/buzzer 7 numaralı pine bağlanmıştır).

```
void loop() {
  for (int j=1;j<=2;j++){
    for (int i=500;i<=800;i++){
      tone(7,i);
      delay(15);
    }
    for(int i=800;i>=500;i--){
      tone(7,i);
```

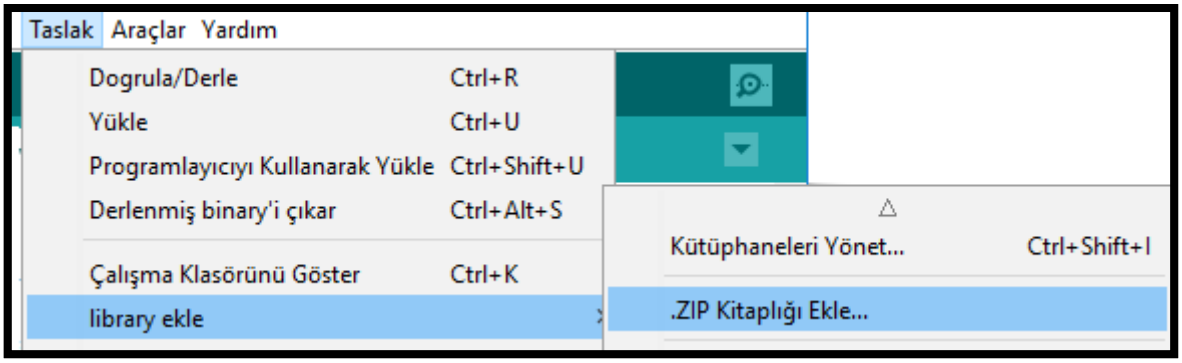
```
      delay(15);
    }
  }
  for(int j=1;j<=6;j++){
    tone(7,500);
    delay(500);
    tone(7,800);
    delay(500);
  }
}
```

1.9 Gözle- LCD Ekrana Yazı Yazdırıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

I2C modülü bağı bulunan LCD'lere yazı yazmak için LiquidCrystal_I2C ve Wire kütüphaneleri kullanılır. Bu kütüphaneler sayesinde I2C üzerinden bağlantı yapılıp LCD'ye komutlar gönderilir. Wire kütüphanesi Arduino kodlama ortamıyla birlikte gelir. Yeniden yüklenmesine gerek yoktur. Fakat "LiquidCrystal_I2C" kütüphanesi Arduino kodlama ortamıyla gelmemektedir. Bu kütüphane internetten indirilerek yüklenmelidir (https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library). İnternetten gerekli zip dosyası indirildikten sonra aşağıdaki resimde gösterildiği gibi "Taslak" menüsündeki "library ekle" seçeneği ve açılan menüden ".ZIP Kitaplığı Ekle" seçeneği tıklanır. Açılan pencereden seçilen zip dosyası seçilerek "Open" butonu tıklanır. Bu şekilde LiquidCrystal_I2C kütüphanesi yüklenmiş olur. Arduino kodlama ortamının kütüphaneyi algılayabilmesi için kapatılıp yeniden çalıştırılması gerekir. Arduino kodlama ortamı kapatılıp açıldığında kütüphane artık kullanıma hazırdır.



Resim 359. I2C Kütüphanesi Ekleme

Bu kütüphaneyi yüklemenin başka bir yolu daha vardır. İndirilen zip dosyası (LiquidCrystal_I2C-master) bir program kullanılarak açılır ve oluşan klasörün ismi "LiquidCrystal_I2C" olarak değiştirilir. Ardından bu klasör Arduino klasöründe bulunan "libraries" klasörünün içerisine kopyalanır. Arduino'nun "libraries" klasörünü bulmak için Arduino kodlama ortamının kısa yolunun üzerine sağ tıklanır. Açılan pencereden "Dosya konumunu aç" seçilir. Açılan Arduino klasörünün içerisinde "libraries" klasörü görülecektir.

I2C modülü kullanılarak LCD'de yazı yazmak için yukarıda bahsi geçen iki kütüphanenin include direktifi ile kodlama ortamına dahil edilmesi gerekir. Bunun için aşağıdaki komutlar kullanılır. Bu kütüphaneleri kodlama ortamına dahil etmek için gerekli komutlar el ile yazılabileceği gibi Taslak>library ekle>Wire ve Taslak>library ekle> LiquidCrystal_I2C seçenekleri tıklanarak bu kütüphanelerin Arduino kodlama ortamı tarafından otomatik olarak da eklenmesi sağlanabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

LCD'ye yazı yazmak için öncelikle bir LCD nesnesi oluşturulmalıdır. Nesne kavramının ne olduğu bu aşamada ortaokul öğrencileri için önemli değildir. Öğrencilerin bu nesnenin nasıl oluşturulduğunu bilmeleri yeterlidir. include direktiflerinin hemen altında aşağıdaki komut kullanılarak lcd isimli bir 20x4'lük lcd nesnesi aşağıdaki gibi oluşturulur:

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

Burada 16x2'lik bir LCD kullanılmak istenirse "LiquidCrystal_I2C lcd(0x27,16,2);" komutunun kullanılması gerekir. Nesne oluşturmak için kullanılan komutta 0x27 parametresi bulunur. Bu parametre adres ile ilgilidir. Bu derste kullanılan LCD'ler için kullanılması uygundur. Fakat başka LCD'lerde farklı değerler kullanılabilir.

LCD'ye yazı yazmadan önce LCD'nin başlatılması ve arka plan ışığının açılması gerekir. Başlatma ve arka plan ışığı açma işlemlerinin yalnızca bir kere program başlangıcında yapılması gerektiği için bu işlemler için kullanılacak komutlar setup fonksiyonu içerisinde yazılmalıdır. Bu komutlar aşağıdaki şekilde yazılabilir:

```
lcd.init();
lcd.backlight();
```

Kullanılacak LCD'de 20 sütun (0-19) ve 4 satır (0-3) bulunmaktadır. Yani LCD üzerinde toplamda 80 hücre vardır. Bu hücrelerden hangisinden başlayarak yazı yazılacağının Arduino kartına bildirilmesi gerekir. Bu iş için setCursor fonksiyonu kullanılır. Bu fonksiyonu kullanarak yazı yazmaya birinci sütun ve sıfırıncı satırdan başlamak için "lcd.setCursor(1, 0);" komutu kullanılır. LCD'de bilgi göstermek için

“lcd.display();” ve “lcd.write();” komutları kullanılır. “lcd.display();” ekrana bir metin yazar. “lcd.write()” ise ekrana bir karakter yazar.

Rehber öğretmen aşağıdaki kodu öğrencilere açıklayarak anlatır. Aynı komutu öğrencilerin yazmasını sağlar. Burada daha önceki derslerde anlatıldığı üzere aşağıdaki bağlantı yapılmıştır.

I2C modülü	Arduino Kartı
GND	GND
VCC	5V
SDA	A4
SCL	A5

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
  lcd.init();
  lcd.backlight();
}
void loop() {
  lcd.setCursor(6, 1);
  lcd.print("TUBITAK");
  lcd.setCursor(6, 2);
  lcd.print("DENEYAP");
}
```

Dikkat

Ortaokul öğrencileri koordinat ekseni konusunu bilmemektedir. LCD’nin satır ve sütunları anlatılırken koordinat ekseni kullanılmadan anlatılmalıdır.

1.10 Gözle ve Uygula- Yanıp Sönen Yazı

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

“lcd.clear();” komutu kullanılarak ekranda bulunan bilgi silinir. Aşağıdaki kod ekranda bir metnin yanıp sönmesini sağlar. Rehber öğretmen öğrencilere kodu açıklayarak anlatır ve ardından öğrencilerin aynı kodu yazmasını sağlar.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
  lcd.init();
  lcd.backlight();
}
void loop() {
  lcd.setCursor(6, 1);
  lcd.print("TUBITAK");
```

```
 lcd.setCursor(6, 2);
 lcd.print("DENEYAP");
 delay(500);
 lcd.clear();
 delay(500);
 }
```

1.11 Gözle- String (Metin) Değişken Tanımlıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Arduino'da String veri tipini tanımlamak için birden fazla yöntem bulunmaktadır fakat burada "String" sınıfı kullanılacaktır. Önceki konularda anlatıldığı gibi "int" veri tipi tam sayıları tanımlamak için kullanılır. String ise metin tipindeki değişkenleri tanımlamak için kullanılır. Örneğin "TUBITAK" bir string tipinde yani metinsel bir ifadedir. Arduino kodlama ortamında string ifadeler çift tırnak içerisinde gösterilir. Aşağıda string tipinde bir değişkenin tanımlanması ve ilk değer ataması örnek olarak gösterilmiştir. Sol taraftaki kodda atolye adlı değişkenin tanımlanması ve ilk değer ataması bir satırda yapılırken sağ tarafta bu işlem farklı iki satırda yapılmıştır. Her iki kullanım da geçerlidir.

```
String atolye= "DENEYAP";
```

```
String atolye;
…// Buraya başka kodlar yazılabilir
atolye="DENEYAP";
```

Aşağıdaki kod ile daha önce LCD ekranda yanıp sönen yazı bu defa string değişkeni kullanılarak yazılmıştır. Rehber öğretmen bu kodu öğrencilere açıklayarak anlatır.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
String kurum = "TUBITAK";
String atolye = "DENEYAP";
void setup() {
 lcd.init();
 lcd.backlight();
}
void loop() {
 lcd.setCursor(6, 1);
 lcd.print(kurum);
 lcd.setCursor(6, 2);
 lcd.print(atolye);
 delay(500);
 lcd.clear();
 delay(500);
 }
```

String'in Uzunluğu:

Bir string ifadenin uzunluğunu yani kaç karakterden oluştuğunu bulmak için length fonksiyonu kullanılır. Aşağıda bu kullanıma bir örnek verilmiştir:

```
String kurum= "TUBITAK";
```

int uzunluk=kurum.length() // uzunluk 7'dir

Aşağıdaki kod ile LCD'ye TUBITAK ve alt satırında TUBITAK kelimesinin uzunluğu olan 7 yazılır. Rehber öğretmen kodu açıklayarak öğrencilere anlatır.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
String kurum = "TUBITAK";
void setup() {
  lcd.init();
  lcd.backlight();
}
void loop() {
  lcd.setCursor(6, 1);
  lcd.print(kurum);
  lcd.setCursor(9, 2);
  lcd.print(kurum.length());
  delay(500);
  lcd.clear();
  delay(500);
  }
```

Metnin Alt Metnini Almak:

Bir metnin içinden sadece bir bölümü yani alt metnini almak için substring fonksiyonu kullanılır. Bu fonksiyonun kullanımı şu şekildedir: "metin_ismi.substring(başlangıç, bitiş);". Burada başlangıç alınan alt metne dahildir fakat bitiş alınan alt metne dahil değildir. Buna örnek olarak aşağıdaki kod verilebilir. Örnekte görüldüğü gibi metnin ilk karakteri sıfırıncı karakter olarak seçilmiştir. Yedi karakterli bir metnin ilk karakteri sıfırıncı karakterdir son karakteri ise altıncı karakterdir.

```
String kurum = "TUBITAK";
String solMetin=kurum.substring(0, 3); // TUB yazar
String sagMetin=kurum.substring(4, 7); //TAK yazar
```

Metindeki Bir Karakteri Almak:

Bir metinde bulunan bir karakteri almak için [] operatörü kullanılabilir. Örneğin içeriği "TUBITAK" olan kurum isimli metninin ortasındaki 'I' karakterini almak için şu kod kullanılabilir: "kurum[3];"

Aşağıdaki kod ile ekrana TUB I TAK metni yazılıp söndürülür. Rehber öğretmen bu kodu öğrencilere açıklayarak anlatır.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
String kurum = "TUBITAK";
String solMetin;
String sagMetin;
void setup() {
  lcd.init();
  lcd.backlight();
}
void loop() {
  solMetin=kurum.substring(0, 3);
  sagMetin=kurum.substring(4, 7);
```

```
lcd.setCursor(6, 1);
lcd.print(solMetin);
lcd.setCursor(10, 1);
lcd.print(kurum[3]);
lcd.setCursor(12, 1);
lcd.print(sagMetin);
delay(500);
lcd.clear();
delay(500);
}
```

2. TASARLA VE ÜRET

2.1 Tasarla ve Üret- Döngüler ile İleri Led Animasyonu

Malzeme Listesi:

Arduino Uno
Breadboard
5 adet led
5 adet 220 Ohm direnç

Bu etkinlikte amaç aşağıdaki şekilde çalışan led animasyonu yapmaktır:

(i) Program bütün ledler (beş adet) sönük olarak başlayacaktır,

(ii) Birinci led bir saniye yanacaktır ve sönmeyecektir,

(iii) İkinci led bir saniye yanacaktır ve sönmeyecektir,

(iv) Üçüncü led bir saniye yanacaktır ve sönmeyecektir,

(v) Dördüncü led bir saniye yanacaktır ve sönmeyecektir,

(vi) Beşinci led bir saniye yanacaktır,

(vii) Bütün ledler bir buçuk saniye içerisinde sönüp, yanıp, sönecektir,

(viii) Beşinci led bir saniye yanacaktır ve sönmeyecektir,

(ix) Dördüncü led bir saniye yanacaktır ve sönmeyecektir,

(x) Üçüncü led bir saniye yanacaktır ve sönmeyecektir,

(xi) İkinci led bir saniye yanacaktır ve sönmeyecektir,

(xii) Birinci led bir saniye yanacaktır,

(xiii) Bütün ledler bir buçuk saniye içerisinde sönüp, yanıp, sönecektir,

(xiv) i-xii adımları yinelenecektir.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Bu görev için aşağıdaki örnek program kodu kullanılabilir.

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  for (int i=2;i<=6;i++){ // Başlangıçta bütün ledler söndürülür
    digitalWrite(i,LOW);
    }
}
void loop() {
  for (int i=2;i<=6;i++){
    digitalWrite(i,HIGH);
    delay(1000);
  }
  for (int i=2;i<=6;i++){
    digitalWrite(i,LOW);
  }
  delay(500);
  for (int i=2;i<=6;i++){
    digitalWrite(i,HIGH);
  }
  delay(500);
  for (int i=2;i<=6;i++){
    digitalWrite(i,LOW);
  }
  delay(500);
  for(int i=6;i>=1;i--){
    digitalWrite(i,HIGH);
    delay(1000);
    }
  for (int i=2;i<=6;i++){
    digitalWrite(i,LOW);
  }
  delay(500);
  for (int i=2;i<=6;i++){
    digitalWrite(i,HIGH);
  }
  delay(500);
  for (int i=2;i<=6;i++){
    digitalWrite(i,LOW);
  }
  delay(500);
}
```

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Aşağıdaki kodu Arduino Uno'ya yükleyip çalıştırmadan, kodun LCD çıktısının nasıl olacağını bulup tanımlayınız.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
  lcd.init();
  lcd.backlight();
}
void loop() {
  for (int i=0;i<=3;i++){
    for(int j=4-i;j<=4+i;j++){
      lcd.setCursor(j, i);
      lcd.print("*");
    }
    lcd.delay(500);
  }
  delay(3000);
  lcd.clear();
  delay(500);
}
```

2. Bu derste gördüğünüz konuların tamamını birleştirerek bir proje yapacak olsanız, bu projeyi günlük yaşamda nerede kullanabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Kod yazarken noktalı virgül kullanmamak gibi yaptığım basit hatalardan ötürü sürekli hata mesajı aldım.
    - İç içe döngülerin mantığını anlamakta sıkıntı yaşadım.
    - Alışkın olduğum için ArduinoBlocks ortamında kod blokları ile kod yazmak istiyorum. Arduino kodlama ortamı bana yabancı geliyor.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?

- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 Basit Ambulans Sireni ve Lambası

Malzeme Listesi:

Arduino Uno
Breadboard
4 adet 100 Ohm direnç
1 adet RGB
1 adet pasif buzzer/hoparlör

Bu etkinlikte amaç basitçe bir ambulans sirenini çalmak ve buna paralel olarak RGB kullanarak kırmızı-mavi ışık yakarak ambulans lambasını modellemektir. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Bu görev için aşağıdaki örnek program kodu kullanılabilir.

Bu kod için RGB'nin kırmızı bacağı 10 numaralı pine, yeşil bacağı 9 numaralı pine ve mavi bacağı 6 numaralı pine bağlanmıştır. Burada RGB için analogWrite komutu tercih edilmiştir. Aynı işlem digitalWrite komutu kullanılarak da yapılabilir. Buzzer'ın veya hoparlörün + bacağı ise 7 numaralı pine bağlanmıştır.

```
void setup() {
  analogWrite(9,255);
}
void loop() {
  tone(7,500);
  analogWrite(10,0);
  analogWrite(6,255);
  delay(500);
  tone(7,800);
  analogWrite(10,255);
  analogWrite(6,0);
  delay(500);
}
```

4.2 Sırayla Akan TUBITAK Yazısı

Malzeme Listesi:

Arduino Uno

Breadboard

4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bu etkinlikte amaç TUBITAK yazısının her bir harfinin sırayla ekranın solundan sağına akmasını sağlayıp ekranın sağında TUBITAK yazısı tamamen belirdikten sonra bu yazının yanıp sönmesini sağlamaktır.

Bu görevin detayı aşağıdaki şekildedir:

(i) TUBITAK yazısı ekranın en solunda belirir ve ekranda 1 saniye kalır,

(ii) K harfi TUBITAK yazısından ayrılır, kayarak ekranın en sağına gider,

(iii) A harfi TUBITA yazısından ayrılır, kayarak K harfinin soluna gider,

(iv) T harfi TUBIT yazısından ayrılır, kayarak A harfinin soluna gider,

(v) I harfi TUBI yazısından ayrılır, kayarak T harfinin soluna gider,

(vi) B harfi TUB yazısından ayrılır, kayarak I harfinin soluna gider,

(vii) U harfi TU yazısından ayrılır, kayarak B harfinin soluna gider,

(viii) T harfi kayarak U harfinin soluna gider,

(ix) Ekranına en sağındaki TUBITAK yazısı birkaç kere yanıp söner ve ekrandan kaybolur,

(x) Yukarıdaki adımlar sırayla sürekli tekrarlanır.

Programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Bu görev için aşağıdaki örnek program kodu kullanılabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
String metin="TUBITAK";
String soldaKalan;
String sagaGiden;
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
}
void loop() {
  lcd.setCursor(0,0);
  lcd.print(metin);
  delay(1000);
  lcd.clear();
  for(int j=6;j>=0;j--){
    for (int i=j;i<=(13+j);i++){
      soldaKalan=metin.substring(0,j);
      lcd.setCursor(0,0);
      lcd.print(soldaKalan);
      lcd.setCursor(i,0);
      lcd.print(metin[j]);
      lcd.setCursor(14+j,0);
      lcd.print(sagaGiden);
      delay(200);
      lcd.clear();
```

```
  }
  sagaGiden=metin.substring(j,7);
  }
 lcd.setCursor(13,0);
 lcd.print(sagaGiden);
 delay(500);
 lcd.clear();
 delay(500);
 lcd.setCursor(13,0);
 lcd.print(sagaGiden);
 delay(500);
 lcd.clear();
 delay(500);
 lcd.setCursor(13,0);
 lcd.print(sagaGiden);
 delay(500);
 lcd.clear();
 delay(500);
 soldaKalan="";
 sagaGiden="";
 }
```

4.3 Gelişmiş Ambulans Sireni ve Lambası

Bu derste basit ambulans sireni ve lambası birlikte çalıştırılmıştır. Bu etkinlikte amaç gelişmiş ambulans siren sesi (2.2 Tasarla ve Üret – Basit Ambulans Sireni ve Lambası) ile ambulans lambasının birlikte çalışmasını sağlamaktır. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Bu görev için aşağıdaki örnek program kodu kullanılabilir. Ortaokulda modüler aritmetik konusu işlenmediği için kod içerisinde mod operatörü kullanılmamıştır. Rehber öğretmen öğrencileri yönlendirirken mod operatörünü kullanmamalıdır. Rehber öğretmen öğrencilere doğrudan kodu vermemelidir. Öğrenciler çözüm üzerinde uğraşırken onları çözümü vermeden yönlendirebilir.

```
void setup()
{
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
pinMode(5, OUTPUT);
}
void loop()
{

  for (int j=1;j<=2;j++){
    int sayac=0;
    for (int i=200;i<=300;i=i+1){
      sayac=sayac+1;
      tone(10,i*2);
      delay(25);
      if (sayac==1||sayac==2||sayac==7||sayac==8){
        digitalWrite(7,LOW);
```

```
      digitalWrite(6,HIGH);
      digitalWrite(5,HIGH);
     }
     else if (sayac==17||sayac==18||sayac==23||sayac==24){
      digitalWrite(7,HIGH);
      digitalWrite(6,HIGH);
      digitalWrite(5,LOW);
     }
     else {
      digitalWrite(7,HIGH);
      digitalWrite(6,HIGH);
      digitalWrite(5,HIGH);
     }
     if (sayac==32) sayac=0;
    }
    sayac=0;
    for (int i=200;i<=300;i++){
     sayac++;
     tone(10,1000-(i*2));
     delay(25);
     if (sayac==1||sayac==2||sayac==7||sayac==8){
      digitalWrite(7,LOW);
      digitalWrite(6,HIGH);
      digitalWrite(5,HIGH);
     }
     else if (sayac==17||sayac==18||sayac==23||sayac==24){
      digitalWrite(7,HIGH);
      digitalWrite(6,HIGH);
      digitalWrite(5,LOW);
     }
     else {
      digitalWrite(7,HIGH);
      digitalWrite(6,HIGH);
      digitalWrite(5,HIGH);
     }
     if (sayac==32) sayac=0;
    }
}
}
```

4.4 Sırayla Akan Yazının Genellemesi

Bu derste Sırayla Akan TUBITAK Yazısı etkinliği yapılmıştır. Bu etkinlikteki amaç bu akan yazının sadece TUBITAK için değil 10 karakterden küçük olan bütün metinler için çalışmasını sağlamaktır. Başka bir değişle TUBITAK yazısının "4.2 Sırayla Akan TUBITAK Yazısı" etkinliğinde aktığı gibi, 10 karakterden küçük herhangi bir metin verildiğinde bu metinde benzer şekilde akmalıdır. Fakat verilen metnin boyutu bilinmemektedir. Verilen metin 10 karakterden küçük herhangi bir metin olabilir. Programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

Proje Hazırlıyorum

Bu haftada, proje çalışmalarına devam edilmesi önerilmektedir. Projenin "empati" süreci için planlamalar yapılmalıdır. Detaylar için "EK-10-Proje Hazırlıyorum" bölümü incelenebilir.

11. Bölüm- Joystick, Seri Port Ekranı ve Bluetooth Verileri

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol, kondansatör, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.
- Öğrenciler kızıl ötesi kumanda seti, RC522 RFID NFC kiti ve ESP8266 wifi modülü kullanarak IOT devreleri oluşturup programlayabilir.
- Öğrenciler sensörlerden alınan verilerin internete aktarımını sağlamak için gerekli elektronik devreleri oluşturup programlayabilir.
- Öğrenciler Arduino kodlama ortamında değişkenleri, kontrol bloklarını, döngüleri, iç içe döngüleri ve PWM pinlerini kullanarak programlar oluşturabilir.

Bölümün Kazanımları:

- Arduino kodlama ortamında joystick verilerini programlarda kullanabilir.
- Arduino kodlama ortamında dizileri kullanarak LCD ekran görünümlerini biçimlendirebilir.
- Arduino kodlama ortamında joystick ve seri port verilerini kullanarak LCD ekran görünümlerini biçimlendirebilir.
- Arduino kodlama ortamında Bluetooth ile led ve LCD ekran kontrolü işlemlerini gerçekleştirebilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin Arduino Kodlama ortamında joystick, seri port ekranı ve Bluetooth verilerini kullanarak seri ekranı ve LCD'yi programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, joystick, direnç (1 adet 220 ohm, 3 adet 1K ohm, 1 adet 100 ohm), 1 adet led, 1 adet HC-05 Bluetooth modülü, 4x20 Mavi LCD ekran- I2C dönüştürücü kartı (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Arduino kodlama ortamında joystick verilerinin seri ekrana ve LCD'ye aktarılması, LCD'nin joystick hareketlerine göre biçimlendirilmesi, seri port ekranındaki verilerin LCD'ye aktarılması, Bluetooth ile led ve LCD'nin kontrolü işlemlerinin gösterilmesi.

Uygula: Arduino kodlama ortamında joystick verilerinin seri ekrana ve LCD'ye aktarılması, LCD'nin joystick hareketlerine göre biçimlendirilmesi, seri port ekranındaki verilerin LCD'ye aktarılması, Bluetooth ile led ve LCD'nin kontrolü işlemlerinin uygulanması.

Tasarla: Arduino kodlama ortamında Joystick ve LCD kullanarak geliştirilecek "Gülen Yüz" oyununun programlanması için tanımlama ve fikir üretme adımlarının oluşturularak tasarımının yapılması.

Üret: Tasarımı yapılan "Gülen Yüz" oyununun Arduino kodlama ortamı kullanılarak programlanması.

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1 Gözle- Joystick Verilerini Okuyup Seri Ekranda Yazdırma

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick

Bu etkinlikte amaç joystick'ten gelen X, Y ve buton değerlerini seri port ekranına yazdırmaktır. Daha önce anlatıldığı üzere joystick'te X ve Y eksenleri ve bir de buton bulunmaktadır. X ve Y eksenleri ile joystick'in sağ-sol ve yukarı-aşağı hareketleri algılanabilir. X ve Y eksenlerinden bir aralık içerisinde değişen değerler elde edildiği için bunların analog girdi pinleri üzerinden okunması gerekir. Bu iş için Arduino kodlama ortamında analogRead komutu kullanılır. Joystick'te bulunan butondan yalnızca 1 veya 0 değerleri gelebilir. Bu yüzden buton değerlerini okumak için digitalRead komutu kullanılır. digitalRead komutu dijital girdi almak için kullanılır. Daha önce anlatıldığı üzere dijital girdi almak için pull up veya pull down direnç kullanmak gerekir. Arduino kartın dijital girişlerinde dahili pull up direnç bulunmaktadır. Dışarıdan bir direnç takarak dijital girdi almak yerine Arduino Uno kartında bulunan pull up direnci kullanmak daha pratik olacaktır. Bir dijital girişteki pull up direnç kullanılmak isteniyorsa pinMode komutuyla önceden bildirilmelidir. pinMode komutunun girdi alacağı ve pull up direnç kullanacağı "pinMode(pin_numarası, INPUT_PULLUP);" şeklinde belirtilir.

Joystick'ten gelen değerlerin seri port ekranında gösterilmesi istenilmektedir. Seri port ekranını kullanmadan önce seri port bağlantısı Setup fonksiyonu içerisinde şu şekilde yapılır: "Serial.begin(9600);". Seri port ekranına bilgi yazdırmak için Serial.print(bilgi) veya Serial.println(bilgi) komutları kullanılabilir. Print komutu ile yazdırılan yazıdan sonra yeni yazılacak yazı aynı satırdan devam eder. println komutu ise bilgi yazdırıldıktan sonra satır başı yapılmasını ve bir alt satıra geçilmesini sağlar.

Aşağıda bu görev için kullanılabilecek kod verilmiştir. Rehber öğretmen kodu açıklayarak öğrencilere anlatır. Kodu anlatırken if koşul cümleciğinin nasıl kullanıldığını ve seri port ekranının nasıl açıldığını ayrıca vurgular. Kodu çalıştırır ve ardından öğrencilerin aynı devreyi kurarak kodu çalıştırmasını sağlar. Kod üzerindeki açıklamalar joystick pinlerinin Arduino kartında bağlı oldukları pinleri de göstermektedir. Örneğin kodun sekizinci satırında joystick'in x ekseni pininin Arduino Uno'da A0 numaralı pine takıldığı ifade edilmiştir.

```
int xEkseni=0;
int yEkseni=0;
int butonDurumu=0;
void setup() {
  pinMode(2, INPUT_PULLUP);
  Serial.begin(9600);
}
void loop() {
xEkseni=analogRead(A0); // x ekseni A0'a bağlı
yEkseni=analogRead(A1); // y ekseni A1'e bağlı
butonDurumu=digitalRead(2); // buton 2'ye bağlı
Serial.print("X ekseni: ");
Serial.println(xEkseni);
Serial.print("Y ekseni: ");
Serial.println(yEkseni);
if (butonDurumu==0){
  Serial.println("Buton Basildi");
```

```
}
else{
  Serial.println("Buton basili degil");
}
delay(1000);
}
```

Dikkat

Analog girdi için pinMode komutunun kullanılması zorunlu değildir. Fakat okuma kolaylığı açısından analog girdi ve çıktı için pinMode kullanılabilir.

1.1 Uygula- Joystick Verisini Yönlere Göre Yazdırıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick

Bu etkinlikte joystick'in yöneldiği yönün seri port ekranına yazdırılması amaçlanmaktadır. Joystick'in yerleştirilme şekline göre yönler değişebileceğinden öğretmen aşağıda verilen X ve Y değerlerinde değişiklik yapabilir. Bu görevin detayı aşağıdaki gibidir:

i) Joystick kuzeye yönelmiş ve son noktaya ulaşmışsa (X:1023, Y:495) ekrana "kuzey" yazılır,

(ii) Joystick güneye yönelmiş ve son noktaya ulaşmışsa (X:0, Y:495) ekrana "güney" yazılır,

(iii) Joystick doğuya yönelmiş ve son noktaya ulaşmışsa (X:490, Y:1023) ekrana "doğu" yazılır,

(iv) Joystick batıya yönelmiş ve son noktaya ulaşmışsa (X:490, Y:0) ekrana "batı" yazılır,

(v) Joystick ortada ilk pozisyonunda ise (X:490, Y:495) ekrana "yönsüz" yazılır,

(vi) Bu değerlerden başka bir değer gösterirse ekrana "ara yön" yazılır.

```
int xEkseni=0;
int yEkseni=0;
void setup() {
  Serial.begin(9600);
}
void loop() {
xEkseni=analogRead(A0); // x ekseni A0'a bağlı
yEkseni=analogRead(A1); // y ekseni A1'e bağlı
if (xEkseni==1023 && yEkseni==495){
  Serial.println("Kuzey");
}
else if(xEkseni==0 && yEkseni==495){
  Serial.println("Güney");
}
else if (xEkseni==490 && yEkseni==1023){
  Serial.println("Doğu");
}
else if (xEkseni==490 && yEkseni==1023){
  Serial.println("Batı");
}
else if(xEkseni==490 && yEkseni==495){
```

```
  Serial.println("Yönsüz");
}
else{
  Serial.println("Ara Yön");
}
delay(500);
}
```

Dikkat

Yukarıda X ve Y için verilen 490 ve 495 değerleri joystick'ten joystick'e göre değişkenlik gösterebilir. Öğrenciler kendi durumlarına göre bu değerleri ayarlamalıdır.

1.2 Gözle- Dizileri Kullanarak LCD'ye Özel karakter Yazdırıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

LCD'deki her bir hücre 5 sütun 8 satırdan oluşmaktadır. Yani her bir hücre aslında 40 alt hücreden oluşmaktadır. Burada bulunan alt hücreler kullanılarak özel karakterler (sembol) tasarlanabilir. Bu işlemi yapabilmek için öncelikle "diziler" ve "byte very" tipi bilinmelidir.

Diziler:

Diziler istenilen sayıda değişkenin bir araya getirilmesi ile oluşturulan bileşik değişkenlerdir. Daha önce değişkenler içerisinde bir anda bir birimlik bilgi taşıyabilen kutulara benzetilmişti. Diziler ise bu kutuların istenilen sayıda yan yana getirilmesi ile oluşan kutu dizisine benzetilebilir. Bu kutu dizisindeki her bir kutuya ayrı ayrı isim vermek yerine bunlara genel bir isim verilir. Burada aşağıdaki resimde görüldüğü gibi kutuların içerisinde 12 adet nota olduğu ve kutulara genel olarak nota ismi verildiği kabul edilir. Birinci kutudaki nota nota[0], ikinci kutudaki nota[1], üçüncü kutudaki notaya nota[2] ve son kutudaki notaya nota[11] isimleri kullanılarak ulaşılabilir.

nota[0]	nota[1]	nota[2]	nota[3]	nota[4]	nota[5]	nota[6]	nota[7]	nota[8]	nota[9]	nota[10]	ota[11]

Resim 360. Notalar

Bu notalara ilk değer ataması farklı şekillerde yapılabilir. Her dizinin bir veri tipi vardır. Bu yüzden dizi tanımlanırken önce dizinin veri tipi yazılır. Aşağıda örnekte görüldüğü gibi dizinin tanımlanması ve ilk değer ataması aynı satırda yapılmıştır (**Not**: Değer olarak notaların frekans karşılıkları yazılmıştır).

```
int nota[12]={294, 294, 330, 294, 392, 370, 294, 294, 330, 294, 440, 392};
```

Dizi tanımlandıktan sonra da değer ataması yapılabilir. Bunun için aşağıdaki kod kullanılabilir.

```
int nota[12];// içerisinde 12 kutunun olduğu dizi tanımlanmıştır
… //diğer kodlar
nota[0]=294;
nota[1]=294;
… // diğer nota değerleri burada belirlenmiştir
nota[11]=392;
```

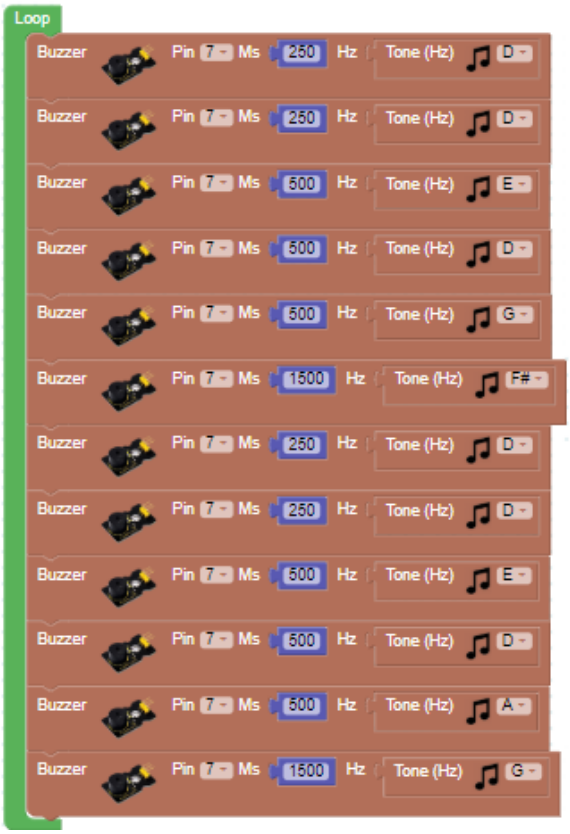
Tanımlanan bir dizi elemanı daha sonra değiştirilmek istenirse o elemana "indis" numarası ile ulaşılır ve yeni değeri atanır. Örneğin 11. nota 494 değeri ile değiştirilmek istenmiş olsun, bu işlem için şu komut kullanılabilir: "nota[11]=494;"

Bu dersin ikinci bölümünde aşağıdaki resimde görülen kod kullanılarak "Mutlu Yıllar Sana" şarkısı çalınmıştı. Burada kullanılan notaların frekansı ise aşağıdaki tabloda görüldüğü gibidir.

Notaların Frekans (Hz) Karşılıkları						
DO	RE	Mİ	FA	SOL	LA	Sİ
262	294	330	349	392	440	494

Resim 361. Nota Frekansları

ArduinoBlocks ortamında uzun uzun yazılan kod Arduino kodlama ortamında diziler kullanılarak daha az satırda yazılabilir (**Not:** Bu iş aslında ArduinoBlocks ortamında listeler (Lists) kullanılarak da yapılabilir). Bu görev için notalar ve notaların çalma süreleri iki farklı diziye sırayla aktarılmalıdır.



Resim 362. Örnek Program

Burada aşağıdaki kod kullanılabilir. Aşağıdaki kodda ilk defa buzzerın bağlandığı pin bir değişken olarak tanımlanmıştır. Rehber öğretmen kodu açıklarken bu durumdan da bahsetmelidir. Rehber öğretmen aşağıdaki kodu öğrencilere açıklar.

```
int nota[12]={294,294,330,294,392,370,294,294,330,294,440,392};
int calmaSuresi[12]={250,250,500,500,500,1500,250,250,500,500,500,1500};
int buzzerPini=2; //Buzzer'ın 2 numaralı pine bağlanacağını belirten değişken tanımlanmıştır.
void setup() {
  pinMode(buzzerPini,OUTPUT);
}
void loop() {
  for (int i=0;i<12;i++){
    tone(buzzerPini,nota[i]);
    delay(calmaSuresi[i]);
```

```
    noTone(buzzerPini);
    delay(10);
  }
}
```

Byte Tipi Değişken:

Program yazarken bazı durumlarda değişken için pozitif ve küçük olan tam sayı değerleri kullanmak gerekebilir. Burada int tipinde değişken tanımlanabilir fakat int tipi negatif sayılar ve büyük/küçük sayılar için de kullanılabildiği için gereğinden daha büyük bir hafıza kullanılmış olur. Bunun yerine bu değişken byte tipinde tanımlanabilir. Byte tipindeki değişkenler içerisinde 0-255 arasındaki tam sayı değerleri saklanabilir. Bu şekildeki değişkeni tanımlamak diğer değişken tanımları ile aynıdır. Aşağıda bir örnek verilmiştir.

```
byte haftaninGunleri=7;
```

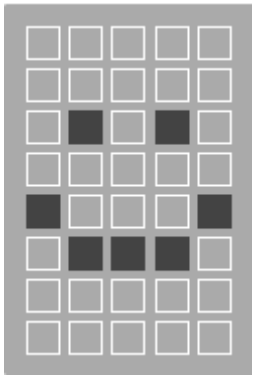
```
byte haftaninGunleri;
…//diğer kodlar
haftaninGunleri=7;
```

Özel Karakter Tanımlama:

LCD'de özel karakter yazmak için byte tipi diziler kullanılacaktır. Özel karakter tasarımı yapmak için farklı siteler vardır. Aşağıda bağlantısı verilen sitelerden herhangi birisi bu iş için kullanılabilir. Rehber öğretmen dizilerin nasıl tanımlandığını biliyorsa bu siteler yerine ArduinoBlocks karakter tanımlama sitesini de kullanabilir.

(i) https://maxpromer.github.io/LCD-Character-Creator/
(ii) https://omerk.github.io/lcdchargen/
(iii) https://mikeyancey.com/hamcalc/lcd_characters.php

Örnek özel karakter olarak gülen yüz tanımlanacaktır. Bunun için önce verilen sitelerden birisi kullanılarak aşağıdaki resimde görülen gülen yüz tasarımı yapılır.



Resim 363. Gülen Yüz Tasarımı

Gülen yüz tasarlandığında site otomatik olarak gülen yüz için gerekli olan diziyi oluşturur. Bu dizi aşağıda görülmektedir. Bu dizi aslında işaretlenen hücreleri göstermektedir. Her bir satırda yapılan yorumlar hücrelerin nasıl belirlendiğini göstermektedir.

```
byte gulenYuz[] = {
  B00000, //ilk satır boş
  B00000, // ikinci satır boş
  B01010, // üçüncü satırda ikinci ve dördüncü kolon işaretlendi
```

```
  B00000, // dördüncü satır boş
  B10001, // beşinci satırda birinci ve beşinci kolon işaretlendi
  B01110, // altıncı satırda ikinci, üçüncü ve dördüncü kolon işaretlendi
  B00000, // yedinci satır boş
  B00000 // sekizinci satır boş
};
```

Dizideki her bir satır, dizinin özel karakter tanımlanırken resimdeki bir satıra karşılık gelmektedir. Bu şekilde tanımlama okuma kolaylığı sağlamakla beraber yapılması zorunlu bir işlem değildir. Bu dizi aşağıda gösterildiği gibi tek bir satırda da tanımlanabilir.

```
byte gulenYuz[] = {B00000, B00000, B01010, B00000, B10001, B01110, B00000,B00000};
```

Dizi ile tanımlanan özel karakter ekrana yazılmadan önce "lcd.createChar(0, gulenYuz);" komutu ile tanımlanır ve ona bir numara atanır (burada 0 atanmıştır). Özel karaktere atanacak sayılar 0 ile 7 arasında olabilir. Bir başka ifadeyle toplamda sekiz farklı özel karakter tanımlanabilir.

Aşağıdaki kod ekrana gülen yüz yazmak için kullanılabilir. Rehber öğretmen kodu öğrencilere açıklar. Ardından aynı kodu öğrencilerin yazmasını sağlar.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,4,20);
byte gulenYuz[]={B00000,B00000,B01010,B00000,B10001,B01110,B00000,B00000};
void setup() {
  lcd.init();
  lcd.backlight();
  lcd.createChar(0,gulenYuz);
}
void loop() {
    lcd.setCursor(0,0);
    lcd.write(0);
}
```

1.3 Gözle ve Uygula- Gülen Yüzü Joystick Kullanarak Sağa Doğru Hareket Ettiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bu etkinlikte amaç gülen yüz karakterini joystick kullanarak sağa doğru hareket ettirmektir. Joystick Y ekseninde sağa doğru ittirildiğinde ekrandaki gülen yüz de sağa doğru hareket etmelidir. Bu görev için aşağıdaki kod kullanılabilir. Joystick'in x ekseni ucu A0, y ekseni ucu ise A1 numaralı pinlere takılmıştır. Rehber öğretmen aşağıdaki kodu öğrencilere açıklar. Daha sonra öğrencilerin aynı görevi yerine getiren bir kod yazmalarını sağlar.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 4, 20); // LCD A4 ve A5 (I2C) üzerinden bağlanmıştır
int xEkseni=0;
int yEkseni=0;
```

```
byte gulenYuz[]={B00000,B00000,B01010,B00000,B10001,B01110,B00000,B00000};
int gulenYuzKoordinat[2]={0,0};
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.createChar(0,gulenYuz);
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.write(0);
}
void loop() {
  xEkseni=analogRead(A0); // joystick VRx ->A0
  yEkseni=analogRead(A1); // joystick VRy ->A1
  if((480<xEkseni && xEkseni<510) && yEkseni>900){
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[0]=gulenYuzKoordinat[0]+1;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(100);
  }
}
```

Dikkat

Burada "X" ekseni için kullanılan 480 ve 510 değeri öğrencilerin kullandığı joystick'ler için farklı olabilir. Kullanılan bu değerler joystick'in "X" ekseninde yaklaşık olarak orta noktada bulunduğunu göstermektedir. Eğer orta nokta değerleri öğrencilerin joystick'lerinde farklılık gösteriyorsa. Yukarıdaki kod o değerlere göre düzenlenmelidir.

1.3 Uygula- Gülen Yüzü Joystick Kullanarak Sağa-Sola Hareket Ettiriyorum

Malzeme Listesi:

- Arduino Uno
- Breadboard
- 1 adet joystick
- 4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bu etkinlikte amaç gülen yüz karakterini joystick kullanarak sağa ve sola doğru hareket ettirmektir. Joystick "Y" ekseninde sağa doğru ittirildiğinde ekrandaki gülen yüz de sağa doğru hareket etmelidir. Joystick "Y" ekseninde sola doğru ittirildiğinde ekrandaki gülen yüz de sola doğru hareket etmelidir. Bu görev için aşağıdaki kod kullanılabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,4,20); //LCD ->A4, A5
int xEkseni=0;
int yEkseni=0;
byte gulenYuz[]={B00000,B00000,B01010,B00000,B10001,B01110,B00000,B00000};
int gulenYuzKoordinat[2]={0,0};

void setup() {
  Serial.begin(9600);
```

```
  lcd.init();
  lcd.backlight();
  lcd.createChar(0,gulenYuz);
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.write(0);
}

void loop() {
  xEkseni=analogRead(A0); //joystick VRx
  yEkseni=analogRead(A1); //joystick VRy
  if((480<xEkseni && xEkseni<505) && yEkseni>900){
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[0]=gulenYuzKoordinat[0]+1;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(100);
  }
    if((480<xEkseni && xEkseni<505) && yEkseni<100){
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[0]=gulenYuzKoordinat[0]-1;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(100);
  }
}
```

1.3 Uygula- Gülen Yüz Kenarlar Arası Geçiş

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet joystick
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bir önceki uygula etkinliğinde (1.3 Uygula- Gülen Yüzü Joystick Kullanarak Sağa-Sola Hareket Ettiriyorum) gülen yüz sağ ve sol kenarlardan taştığı zaman daha fazla ilerleyememektedir veya beklenmedik şekilde hareketler yapmaktadır. Bu etkinlikte amaç joystick yardımıyla sağa ve sola hareket eden gülen yüz, sağ kenardan taşacağı zaman onu aynı satırın en solundan; sol kenardan taşacağı zaman ise onu aynı satırın en sağından yeniden LCD'ye girmesini sağlamaktır. Bu görev için aşağıdaki kod kullanılabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,4,20);
int xEkseni=0;
int yEkseni=0;
byte gulenYuz[]={B00000,B00000,B01010,B00000,B10001,B01110,B00000,B00000};
int gulenYuzKoordinat[2]={0,0};

void setup() {
```

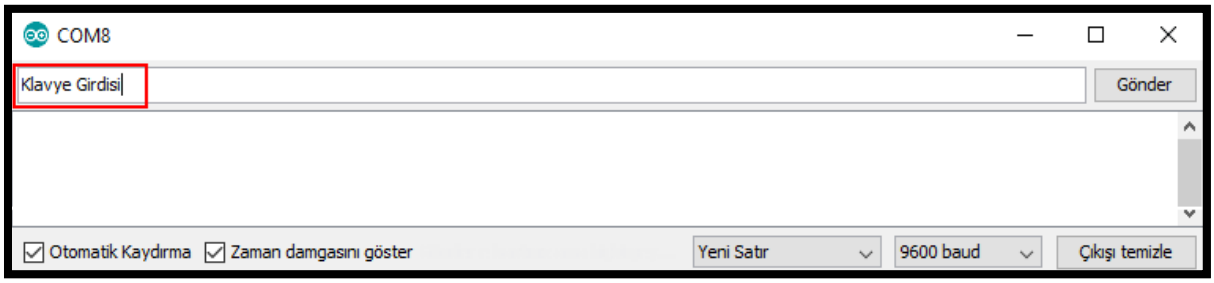
```
 lcd.init();
 lcd.backlight();
 lcd.createChar(0,gulenYuz);
 lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
 lcd.write(0);
}

void loop() {
 xEkseni=analogRead(A0);
 yEkseni=analogRead(A1);
  if((470<xEkseni &&xEkseni<520)&& yEkseni>900){
   if (gulenYuzKoordinat[0]<19){
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.print(" ");
     gulenYuzKoordinat[0]=gulenYuzKoordinat[0]+1;
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.write(0);
     delay(100);
   }
  else{
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[0]=0;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(100);
    }
  }
   if ((470<xEkseni &&xEkseni<520)&& yEkseni<100){
    if(gulenYuzKoordinat[0]>0){
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.print(" ");
     gulenYuzKoordinat[0]=gulenYuzKoordinat[0]-1;
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.write(0);
     delay(100);
    }
    else{
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.print(" ");
     gulenYuzKoordinat[0]=19;
     lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
     lcd.write(0);
     delay(100);
    }
  }
}
```

1.4 Gözle- Seri Port Ekranından Gelen Bilgiyi LCD'ye Yazma

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Daha önce seri port ekranına bilgi yazma ile ilgili etkinlikler yapılmıştı. Seri port ekranı kullanılarak ekrana bilgi çıktısı vermenin yanında seri port ekranından klavye ile bilgi girişi de sağlanabilir. Aşağıdaki resimde klavye girdisinin nereden yapılacağı gösterilmiştir.



Resim 364. Seri Port Ekranı Bilgi Girişi

Klavyeden girilmek istenen bilgi yazıldıktan sonra "Enter" tuşuna basılarak bilgi Arduino karta gönderilir. Seri port ekranından bilgi girişi olup olmadığını kontrol etmek için "Serial.available()" ifadesi kullanılabilir. Eğer seri port ekranından bilgi girişi yapıldıysa Serial.available() değeri sıfırdan büyük olur. Seri port ekranından yapılan girdiyi okumak için "Serial.read();" komutu kullanılabilir. "Serial.read()" komutu Serial.available() değeri sıfırdan büyük olduğu müddetçe girilen bilgiyi karakter okur. Örneğin seri port ekranından "TUBITAK" bilgisi girilmiş olsun. Bu durumda Serial.available() değeri 0'dan büyük olur. "Serial.read();" komutu baştaki 'T' karakterini okur. "UBITAK" bilgisi hala girdi olarak bulunduğu için Serial.available() değeri hala sıfırdan büyüktür. "Serial.read();" komutu ile bir okuma işlemi daha yapılırsa "BITAK" bilgisi hala bilgi olarak bulunacaktır ve Serial.available() değeri sıfırdan büyüktür. Bu şekilde "TUBITAK" bilgisinin tamamı "Serial.read();" komutu ile okunduğunda Serial.available() değeri sıfır olur.

Seri port ekranından bilgi okuyup LCD'ye yazmak için Serial.available() değeri sıfırdan büyük olduğu müddetçe "Serial.read();" komutu ile bilgi okunmalıdır. Okuma işlemi bir çeşit döngü ile yapılmalıdır çünkü okuma işlemi Serial.read() komutu ile tekrarlanır. Fakat burada for döngüsü kullanılamaz. For döngüsü bir döngünün ne kadar tekrar edeceği bilindiği durumlarda kullanılır. "TUBITAK" bilgisi yedi karakter içermektedir. For döngüsü ile yedi defa "Serial.read();" komutu tekrar edilip "TUBITAK" bilgisi okutulabilir. Fakat başka bir bilgi; örneğin "PROGRAMLAMA" girildiğinde for döngüsü gereken işlemi yapamayabilir çünkü "PROGRAMLAMA" bilgisi için döngünün dokuz kere dönmesi gerekir. Bu gibi döngünün kaç defa tekrar edeceği bilinmediği durumlarda, döngünün dönme sayısı bir koşula bağlanır. Yani bir koşul doğru olduğu müddetçe döngü çalışır. Koşul yanlış olduğunda ise döngü tekrar etmeyi bırakır. Bahsi geçen bu döngüye while döngüsü denilir. While döngüsü ile istenilen işlem aşağıdaki kodda gösterildiği şekilde yapılabilir. Rehber öğretmen kodu öğrencilere açıklar ve öğrencilerin aynı kodu yazıp çalıştırmalarını sağlar.

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,4,20);
void setup() {
  lcd.init();
  lcd.backlight();
```

```
  Serial.begin(9600);
}
void loop() {
  while (Serial.available()>0){ // Bu değer sıfırdan büyük olduğu müddetçe döngü yinelenir
    lcd.write(Serial.read());
     }
}
```

Dikkat

Bu komut ile LCD'ye bilgi yazıldığında bilginin sonunda istenmeyen bir karakter eklenmiştir. Bu "Enter" tuşu karakteridir. Seri port ekranında bilgiyi göndermek için "Enter" tuşuna basıldığında "Enter" karakteri de "Serial.read();" komutu tarafından okunmuştur. "Enter" karakterini yazmamanın çeşitli yolları vardır. Bir sonraki etkinlikte (Seri Port Ekranından Gelen Bilgiyi Okuyup Bütün Olarak LCD'ye Yazma) bu işlem gösterilecektir.

1.5 Gözle- Seri Port Ekranından Gelen Bilgiyi Okuyup Bütün Olarak LCD'ye Yazma

Malzeme Listesi:

Arduino Uno
Breadboard
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bir önceki etkinlikte (Seri Port Ekranından Gelen Bilgiyi LCD'ye Yazma) seri port ekranından okunan bilgi karakter LCD'ye yazılmıştır. Çoğu zaman bu işlem tercih edilmez. Bu etkinlikte gelen bilgi öncelikle bir string değişkenine aktarılacaktır ve ardından bu bilgi LCD'ye bütün olarak yazılacaktır. "Serial.read;" komutu bir anda bir karakter okuyabilir. Bu yüzden okunan her yeni karakter bir string değişkene eklenmelidir. Bu iş için "+" operatörü kullanılabilir. "+" operatörü iki string ifadeyi yan yana olacak şekilde birleştirmeyi sağlar. Seri port ekranında bir bilgi yazıldıktan sonra, bilgiyi göndermek için enter tuşuna basılır. Ancak Arduino enter tuşunu da bir karakter olarak algılar ve dolayısıyla gelen karakterler string değişkene eklenirken enter karakteri de eklenecektir. Enter karakteri Arduino kodlama ortamında '\n' olarak ifade edilir. Bir koşul ifadesi (örneğin "if (karakter!='\n')") kullanılarak enter karakteri dışarıda bırakılabilir. Aşağıdaki program bu etkinlik için kullanılabilir. Rehber öğretmen programı öğrencilere açıklar ve öğrencilerin aynı programı yazıp çalıştırmasını sağlar.

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,4,20);
char karakter;
String bilgi;
void setup() {
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
}
void loop() {
  if (Serial.available()>0){
    bilgi="";
    while (Serial.available()>0){
      karakter=Serial.read();
      if (karakter!='\n'){ // enter karakteri alınmaz
```

```
      bilgi=bilgi+karakter; // okunan yeni karakter bilgiye eklenir
    }
   delay(5);
   }
   lcd.clear();
   lcd.setCursor(1,1);
   lcd.print(bilgi);
   Serial.println(bilgi);
  }
}
```

1.6 Gözle ve Uygula- Bluetooth ile Led Kontrolü

Malzeme Listesi:

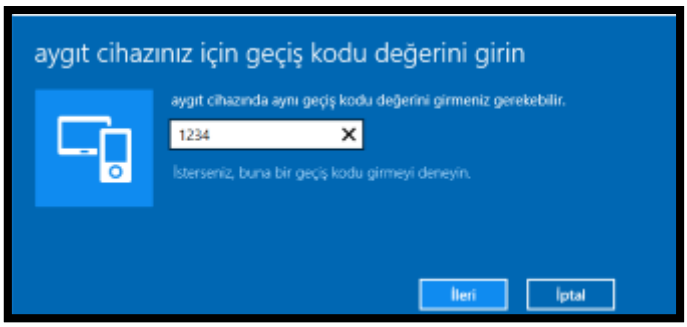
Arduino Uno
Breadboard
3 adet 1K Ohm direnç
1 adet 220 Ohm direnç
1 adet led
1 adet HC-05 Bluetooth modülü

Dikkat

Bu etkinlik öncesinde Processing IDE rehber öğretmenler tarafından bilgisayarlara yüklenmiş olmalıdır.

Bu etkinlikte amaç bilgisayar ile Arduino kart arasında Bluetooth aracılığı ile bağlantı kurup bilgisayarı kullanarak Arduino kart üzerindeki ledi yakıp söndürmektir. Bu etkinlikte üç temel bileşen bulunmaktadır. Bilgisayardaki program, Bluetooth modülü ve Arduino için program. Bilgisayardaki program öğrencilere hazır olarak verilecektir. Bu programın ismi ledAcKapa'dır (EK-8 Led Ac-Kapa Kaynak Kodu) (bağlantı verilebilir). Bluetooth bağlantısı için HC-05 kullanılacaktır. Arduino kodu rehber öğretmen/öğrenciler tarafından yazılacaktır.

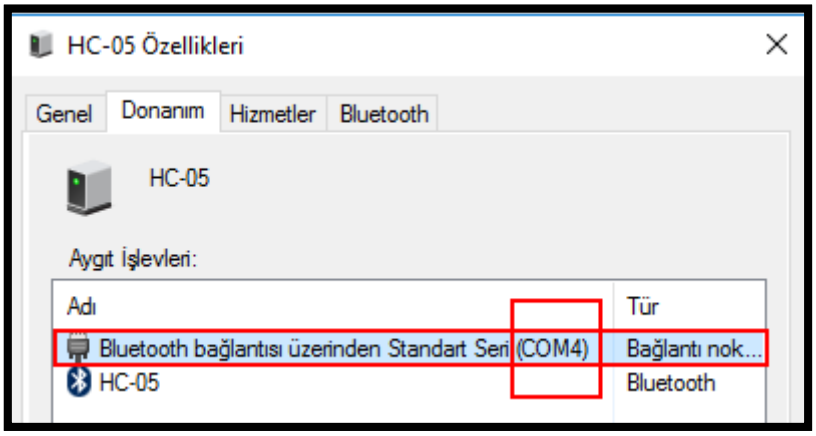
Bilgisayarın Bluetooth bağlantısı ile HC-05'e bağlanması için bilgisayarın Bluetooth seçeneklerinden (Bluetooth Ayarları/Bluetooth and Other Device Settings) HC-05 bulunur ve üzerine tıklanır (Eşleştir). Açılan pencerede aşağıdaki resimde görüldüğü gibi şifre/pin sorulacaktır. Şifre olarak "1234" girilir.



Resim 365. Bilgisayar Bluetooth Bağlantı Ayarı

"ledAcKapa" programı Processing IDE kullanılarak yazılmıştır. Kodun çalışması için HC-05'in hangi COM portu üzerinden bilgisayara bağlandığının bulunması gerekmektedir. Bunun için Denetim Masası (Control Panel)-> Donanım ve Ses (Hardware and Sound)->Aygıtlar ve Yazıcılar (Devices and Printers) penceresinin alt kısmında bulunan Belirtilmemiş (Unspecified) başlığı altında HC-05 bulunur (bunun

için öncelikle HC-05'in bağlanmış olması gerekir). HC-05 üzerine çift tıklandığında açılan pencereden Donanım (Hardware) sekmesi seçilir. Aşağıdaki resimde görüldüğü gibi Aygıt İşlevleri bölümünde HC-05'in hangi COM portuna bağlı olduğu yazılıdır.



Resim 366. Bluetooth Port ayarı

Buradaki örnekte HC-05, COM4 portu üzerinden bilgisayara bağlanmıştır. "lecAcKapa" programı Processing IDE ile açılır. Aşağıdaki resimde gösterildiği gibi, açılan kaynak kodda altıncı satırda bulunan COM portu bilgisi düzenlenir (buraya öğrenciler buldukları port bilgisini yazmalıdır). Daha sonra program kaydedilir (Dosya->Kaydet). Bu aşamada Arduino programı hazır olmadığı için "ledAcKapa" programı çalışmayacaktır. Bir sonraki adımda Arduino programı yazılıp yüklendikten sonra "ledAcKapa" programı çalıştırılmalıdır.

```
1 import processing.serial.*;
2 Serial hc05Port;
3 String ledDurum="KAPALI";
4 void setup(){
5   size(500, 250);
6   hc05Port = new Serial(this, "COM4", 9600); // ***** COM10 Uygun Port Ile Degistirilmeli ******
```

Resim 367. ledAcKapa Programı Port Düzenlemesi

"ledAcKapa" isimli program Bluetooth'un bağlandığı COM portu üzerinden Arduino kart ile iletişime geçer. Programdaki "LED'i Ac" butonuna tıklandığında COM portu üzerinden Arduino seri portuna '1' bilgisi gönderilmektedir. "LED'i Kapat" butonuna tıklandığında ise COM portu üzerinden '0' bilgisi gönderilmektedir. "ledAcKapa" programı bunun yanında seri port üzerinden Arduino tarafından gönderilen led'in durumu (ACIK/KAPALI) ile ilgili bilgiyi aşağıdaki resimde olduğu gibi ekranda gösterir.

Resim 368. ledAcKapa Programı Arayüzü

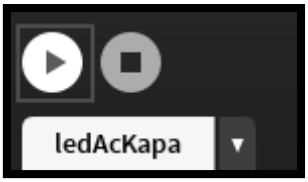
Arduino kartı "ledAcKapa" programı ile eşgüdümlü çalışabilmek için seri portu dinlemelidir. Seri porttan '1' değeri geldiğinde ledi yakmalı, '0' değeri geldiğinde ise ledi söndürmelidir. Arduino programlama ortamında kodu yazmadan önce HC-05 bağlantısı yapılmalıdır. HC-05'in VCC bacağı Arduino kartın 5V pinine; GND bacağı GND pinine, TX bacağı RX pinine (dijital 0) ve RX bacağı gerilim bölücü kullanılarak TX pinine (dijital 1) bağlanmalıdır. Gerilim bölücü olarak üç seri bağlı 1K Ohm direnç kullanılır. Dirençlerin bir ucu GND'ye bağlanır. Diğer ucuna Arduino kartının TX pini bağlanır. 1K - 2K bağlantı noktasından HC-05'in RX bağlantısı yapılır: (Arduino)TX - 1K- (HC-05)RX - 1K - 1K - (Arduino)GND. Arduino TX pininden 5V gerilim üretir fakat HC-05'in RX bacağı 3.3V ile çalışmaktadır. TX pini RX bacağına doğrudan bağlanırsa HC-05'i bozabilir. Bu yüzden TX pininden gelen 5V gerilim bölücü kullanılarak RX bacağına 3.3V olarak iletilmelidir. HC-05 bağlantısı tamamlanmıştır. Son olarak 13 numaralı dijital pine 220 Ohm direnç kullanılarak bir led bağlanır. Aşağıdaki kod Arduino programlama ortamında yazılarak Arduino Uno karta yüklenir.

Dikkat

Program yüklenirken TX ve RX pinlerindeki bağlantılar çıkarılmalıdır ve yükleme bittikten sonra bu bağlantılar yeniden yapılmalıdır.

```
int ledPin=13;
char ledDurum = ' ';
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600);
}
void loop() {
  if(Serial.available() > 0){
    ledDurum = Serial.read();
  }
  if (ledDurum == '0') {
   digitalWrite(ledPin, LOW);
   Serial.println("KAPALI");
   ledDurum = ' ';
  }
  else if (ledDurum == '1') {
   digitalWrite(ledPin, HIGH);
   Serial.println("ACIK");
   ledDurum = ' ';
  }
}
```

Bluetooth ile led kontrolü için artık ortam hazırlanmıştır. ledAcKapa programı ile Arduino kart üzerindeki led açıp kapatılabilir. Bunun için ledAcKapa kodu aşağıdaki resimde gösterilen çalıştır butonu aracılığı ile çalıştırılır. Bu program çalıştırıldığında her şey yolunda gittiyse HC-05 üzerinde bulunan led sürekli yanıp sönme modundan iki saniyede iki kere yanma moduna geçer. Ekranda açılan pencerede "LED'i Ac" butonuna tıklanınca Arduino kartta bulunan led yanar. "LED'i Kapat" butonuna tıklandığında Arduino kartta bulunan led söner.



Resim 369. ledAcKapa Programının Çalıştırılması

Yukarıda anlatılan etkinliği rehber öğretmen öğrencilere adım adım anlatır. Öğrencilere her adımda yardımcı olur. Öğrencilerin gerekli devreyi kurduktan sonra ayarlamaları yapıp programı çalıştırmalarını sağlar.

Dikkat

Yukarıda yazılan kod çalışmazsa bunun nedeni HC-05'in ön tanımlı Baud değeri olabilir. Bazı HC-05'lerde bu ön tanımlı değer 38400 olabilir. Bu durumda hem Arduino kodunda hem de Processing kodunda 9600 olan Baud değeri 38400 olarak değiştirilmelidir. Gerekli kod satırları aşağıda gösterilmiştir:

```
hc05Port = new Serial(this, "COM4", 38400); // Processing kodu

Serial.begin(38400);  // Arduino kodu
```

1.6 Uygula- Bluetooth ile Led Kontrolü ve LCD

Malzeme Listesi:

Arduino Uno
Breadboard
3 adet 1K Ohm direnç
1 adet 220 Ohm direnç
1 adet led
1 adet HC-05 Bluetooth modülü
4x20 Mavi LCD ekran- I2C dönüştürücü kartı

Bu etkinlik öncesinde öğrenciler breadbord ve Arduino üzerindeki bütün devre elemanlarını söker. Arduino'da yeni bir sayfa açar. Bir önceki etkinlikte yazılan hazır kod veya devreyi kullanmaz. Bir önceki etkinlikte ledAcKapa programındaki butonlar kullanılarak Arduino kart üzerinde bulunan led açıp kapatılmıştır. Bu etkinlikte yine program aracılığı ile led açılıp kapatılacaktır. Fakat buna ek olarak led açıldığında LCD'de "LED ACIK" ve kapandığında LCD'de "LED KAPALI" yazacaktır. Bu görev için aşağıdaki kod kullanılabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,4,20);
int ledPin=13;
char ledDurum = ' ';
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
}
void loop() {
```

```
 if(Serial.available() > 0){
   ledDurum = Serial.read();
}
if (ledDurum == '0') {
 digitalWrite(ledPin, LOW);
 Serial.println("KAPALI");
 ledDurum = ' ';
 lcd.clear();
 lcd.print("LED KAPALI");
}
else if (ledDurum == '1') {
 digitalWrite(ledPin, HIGH);
 Serial.println("ACIK");
 ledDurum = ' ';
 lcd.clear();
 lcd.print("LED ACIK");
}
}
```

2. TASARLA VE ÜRET

2.1 Tasarla ve Üret- Gülen Yüz Oyunu

Bu bölümde derste sağa ve sola doğru hareket edebilen gülen yüz etkinliği yapılmıştır. Bu etkinlikte amaç gülen yüz etkinliğini geliştirerek oyun haline getirmektir. Oyunun detayları aşağıdaki şekildedir:

(i) Gülen yüz sahnede joystick yardımıyla sağa-sola ve yukarı-aşağı hareket edebilir,

(ii) Gülen yüz bir satırdan taştığı zaman o satırın diğer ucundan yeniden sahneye girer,

(iii) Gülen yüz bir sütundan taştığı zaman o sütunun diğer ucundan yeniden sahneye girer,

(iv) Oyun başlar başlamaz ekranın rasgele iki hücresinde özel hazırlanmış ödül ve ceza karakterleri belirir,

(v) Gülen yüz ödül karakterinin üzerine gelirse “Tebrikler” – “Oyunu Kazandınız!” yazısı ekranda belirir, hoparlörden kazanma melodisi çalar (bu melodiyi öğrenci belirler) ve oyun biter.

(vi) Gülen yüz ceza karakterinin üzerine gelirse ekranda “Maalesef” – “Oyunu kaybettiniz” yazısı belirir, hoparlörden kaybetme melodisi çalar (bu melodiyi öğrenci belirler) ve oyun biter.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Bu görev için aşağıdaki örnek program kodu kullanılabilir.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,4,20);
int xEkseni=0;
int yEkseni=0;
int melodiPin=8;
byte gulenYuz[]={B00000,B00000,B01010,B00000,B10001,B01110,B00000,B00000};
```

```
byte ceza[8] = {B10000,B01000,B00100, B01010,B10001,B10001,B01010,B00100};
byte odul[8] = {B00000,B01010,B11111,B11111,B01110,B00100,B00000,B00000};
int kaybetmeMelodi[]={392,370,350,330};
int kaybetmeMelodiSure[]={500,500,500,750};
int kazanmaMelodi[]={262,262,262,392,262,392};
int kazanmaMelodiSure[]={250,125,125,375,125,1000};
int gulenYuzKoordinat[2]={0,0};
int cezaKoordinat[2];
int odulKoordinat[2];
int cezaBulundu=0;
int odulBulundu=0;

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  randomSeed(analogRead(A2));
  odulKoordinat[0]=random(20);
  odulKoordinat[1]=random(4);
  cezaKoordinat[0]=random(20);
  cezaKoordinat[1]=random(4);
  lcd.createChar(0,gulenYuz);
  lcd.createChar(1,odul);
  lcd.createChar(2,ceza);
  lcd.setCursor(0,0);
  lcd.print("Oyunu Baslatmak icin");
  delay(1000);
  lcd.clear();
  delay(500);
  lcd.print("Joystick'i saga itin");
  delay(1000);
  lcd.clear();
  lcd.setCursor(odulKoordinat[0],odulKoordinat[1]);
  lcd.write(1);
  lcd.setCursor(cezaKoordinat[0],cezaKoordinat[1]);
  lcd.write(2);
}

void loop() {
  xEkseni=analogRead(A0);
  yEkseni=analogRead(A1);
  if((gulenYuzKoordinat[0]==odulKoordinat[0]) && (gulenYuzKoordinat[1]==odulKoordinat[1])){
    lcd.clear();
    lcd.setCursor(6,1);
    lcd.print("Tebrikler");
    lcd.setCursor(1,2);
    lcd.print("Oyunu Kazandiniz!");
    for(int i=0;i<6;i++){
      tone(melodiPin,kazanmaMelodi[i]);
      delay(kazanmaMelodiSure[i]);
```

```
  noTone(melodiPin);
  delay(5);
 }
 while(1);
 }
 if((gulenYuzKoordinat[0]==cezaKoordinat[0]) && (gulenYuzKoordinat[1]==cezaKoordinat[1])){
  lcd.clear();
  lcd.setCursor(6,1);
  lcd.print("Malesef");
  lcd.setCursor(1,2);
  lcd.print("Oyunu Kaybettiniz!");
  for(int i=0;i<4;i++){
   tone(8,kaybetmeMelodi[i]);
   delay(kaybetmeMelodiSure[i]);
   noTone(8);
   delay(50);
  }
  while(1);
 }
 if((470<xEkseni &&xEkseni<520)&& yEkseni>900){
 if (gulenYuzKoordinat[0]<19){
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.print(" ");
  gulenYuzKoordinat[0]=gulenYuzKoordinat[0]+1;
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.write(0);
  delay(100);
 }
else{
 lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
 lcd.print(" ");
 gulenYuzKoordinat[0]=0;
 lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
 lcd.write(0);
 delay(100);
 }
}
 if ((470<xEkseni &&xEkseni<520)&& yEkseni<100){
 if(gulenYuzKoordinat[0]>0){
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.print(" ");
  gulenYuzKoordinat[0]=gulenYuzKoordinat[0]-1;
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.write(0);
  delay(100);
 }
 else{
  lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
  lcd.print(" ");
  gulenYuzKoordinat[0]=19;
```

```
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(100);
   }
 }
  if ((470<yEkseni &&yEkseni<520)&& xEkseni<100){
   if(gulenYuzKoordinat[1]<3){
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[1]=gulenYuzKoordinat[1]+1;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(200);
   }
   else{
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[1]=0;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(200);
   }
 }
  if ((470<yEkseni &&yEkseni<520)&& xEkseni>900){
   if(gulenYuzKoordinat[1]>0){
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[1]=gulenYuzKoordinat[1]-1;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(200);
   }
   else{
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.print(" ");
    gulenYuzKoordinat[1]=3;
    lcd.setCursor(gulenYuzKoordinat[0],gulenYuzKoordinat[1]);
    lcd.write(0);
    delay(200);
    }
 }
}
```

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Yazdığınız gülen yüz oyununda programlamada öğrendiğiniz fonksiyonların kullanılması uygun olur mu? Cevabınızı sebepleriyle açıklayınız.

2. Bugün gördüğünüz konuları kendi günlük yaşamınızı kolaylaştırmak için nerede kullanabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Gülen Yüz etkinliğindeki algoritmayı anlamakta sıkıntı yaşadım.
    - İçi içe "if" ifadelerini anlamakta zorlandım.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4. İLAVE ETKİNLİK

4.1 ledAcKapa Programını Kullanarak Gülen Yüzü Hareket Ettiriyorum

Bu etkinlikte amaç gülen yüzü sağa-sola hareket ettirmektir. Fakat daha önceki etkinliklerden farklı olarak ledAcKapa programı kullanılarak gülen yüzün hareketi sağlanacaktır. "LED'i Ac" butonu tıklandığında gülen yüz sola, "LED'i Kapat" butonu tıklandığında gülen yüz sağa hareket etmelidir. Etkinlik için gülen yüzün LCD kenarlarına geldiğinde nasıl davrandığı önemli değildir. Öğrenciler isterlerse kenarlara gelindiğinde aynı satırın başına dönme kodunu seçimli olarak yazabilirler. Fakat bunu yazmak zorunda değildirler. Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat öğrencilere hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

4.2 Gülen Yüz Oyununu Geliştiriyorum

Bu derste Gülen Yüz Oyunu etkinliği ile basitçe bir oyun kodu yazılmıştır. Bu etkinlikte amaç oyunu biraz daha geliştirmektir. Oyunun geliştirilmesi öğrencilerin hayal gücüne bağlıdır. Aşağıda maddeler halinde belirli öneriler verilmiştir. Öğrenciler bu önerilere uyarak oyunu geliştirebilir veya kendi hayal gücünü kullanarak farklı şekilde de oyunu geliştirebilir.

Öneriler:

(i) Gülen yüz yerine Pac Man gibi bir yüz ifadesi kullanılabilir,

(ii) Ceza ve ödül yenildikçe yeni ceza ve ödüller ekranda rasgele hücrelerde belirir,

(iii) Ekranın belirli bir köşesi puan alanı olarak ayrılır. Ödül ve cezayı yeme durumuna göre puan hesaplanır. Hesaplanan puan ekranda ayrılan köşede sürekli gösterilir. Pac Man bu köşeye ulaşamaz.

(iv) Oyunda birkaç seviye olur. İleri seviyelerde Pac Man'in ödüle ulaşması için hareketli olan cezalardan kurtulup ilerlemesi gerekir.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat öğrencilere hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

4.3 Modifiye lcdAcKapa Programı ile Gülen Yüz Oyunu

Bu etkinlikte amaç gülen yüz oyununda gülen yüzü modifiye ledAcKapa programı ile hareket ettirmektir. ledAcKapa programı öğrenciler tarafından incelenecek ve kod anlaşıldıktan sonra modifiye edilerek Sağ, Sol, Yukarı, Aşağı butonları oluşturulacaktır. Gülen yüz bu butonlar aracılığı ile hareket ettirilecektir. Yani modifiye ledAcKapa programında:

(i) Sol butonuna basıldığında LCD'de gülen yüz sola,

(ii) Sağ butonuna basıldığında LCD'de gülen yüz sağa,

(iii) Aşağı butonuna basıldığında LCD'de gülen yüz aşağıya,

(iv) Yukarı butonuna basıldığında LCD'de gülen yüz aşağıya hareket edecektir.

Daha önceden joystick ile oynan gülen yüz oyunu bu sayede modifiye ledAcKapa programı ile oynanmış olacaktır.

Devreyi oluşturmaya ve programı yazmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir. Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra set başında çalışarak istenilen görevi yerine getirir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada rehber öğretmen onlara yardımcı olabilir. Fakat öğrencilere hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

Proje Hazırlıyorum

Bu haftada, proje çalışmalarına devam edilmesi önerilmektedir. Projenin "tanımlama" süreci için planlamalar yapılmalıdır. Detaylar için "EK-10-Proje Hazırlıyorum" bölümü incelenebilir.

12. Bölüm-Kendi Arabamı Kodluyorum

Ön bilgi:

- Öğrenciler gerilim, akım ve direnç kavramlarını bilir ve aralarındaki ilişkiyi açıklayabilir.
- Öğrenciler ArduinoBlocks geliştirme ortamında değişken, döngü ve fonksiyon kullanarak uygulama geliştirip Arduino'ya aktarabilir.
- Öğrenciler Arduino pinlerini, breadboard çalışma prensiplerini bilir ve led/direnç/buzzer/breadboard/Arduino bağlantılarını yapabilir.
- Öğrenciler iletken, yalıtkan ve yarı-iletken kavramlarını açıklayabilir ve diyot içeren elektronik devreler oluşturabilir.
- Öğrenciler VE/VEYA kapısı içeren devreleri programlayabilir.
- Öğrenciler dijital ve analog girdi arasındaki farkları ve benzerlikleri açıklayabilir.
- Öğrenciler elektronik devrelerde potansiyometre, LDR, Mesafe Sensörü, LCD, kızılötesi sensör, RGB led, PWM, transistor, step motor, servo motor, DC motor, joystick, robotik kol, kondansatör, DTH11 sıcaklık ve nem sensörü, RTC DS3231 saat modülü, PIR sensörü, kapı alarm sensörü, ses sensörü, alev sensörü, yanıcı gaz sensörü, toprak nem sensörü, yağmur sensörü, tilt sensörü ve tuş takımı kullanımını bilir ve gerekli kodlamaları yapabilir.
- Öğrenciler elektronik devrelerde seri port kullanmayı ve ArduinoBlocks ortamında programlamayı bilir.
- Öğrenciler Multimetre kullanabilir.
- Öğrenciler kızıl ötesi kumanda seti, RC522 RFID NFC kiti ve ESP8266 wifi modülü kullanarak IOT devreleri oluşturup programlayabilir.
- Öğrenciler sensörlerden alınan verilerin internete aktarımını sağlamak için gerekli elektronik devreleri oluşturup programlayabilir.
- Öğrenciler Arduino kodlama ortamında değişkenleri, kontrol bloklarını, döngüleri, iç içe döngüleri ve PWM pinlerini kullanarak programlar oluşturabilir.
- Öğrenciler Arduino kodlama ortamında joystick, seri port ve Bluetooth verilerini kullanarak seri port ekranı ve LCD görünümlerini biçimlendirebilir.

Bölümün Kazanımları:

- MPU 6050 açı sensörü ile ilgili Arduino kodlama ortamında program kodu oluşturabilir.
- LM 393 kızılötesi hız sensörü ile ilgili Arduino kodlama ortamında program kodu oluşturabilir.
- Arduino kodlama ortamında LM 393 kızılötesi hız sensörü verilerini kullanarak arabayı daha dengeli hareket ettirmek için gerekli program kodlarını oluşturabilir.

Bölümün Amacı:

Bu bölümün amacı öğrencilerin öncelikle MPU 6050 açı sensörünü ve LM 393 kızılötesi hız sensörünü Arduino kodlama ortamında programlayabilmelerini sağlayarak, arabanın dengeli

hareket edebilmesi için LM 393 kızılötesi hız sensörü verilerini Arduino Kodlama ortamında programlayabilmelerini sağlamaktır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Arduino Uno, breadboard, 1 adet MPU 6050 sensör, 2 adet 220 ohm direnç, 2 adet led, 2 adet LM 393 kızılötesi hız sensörü, 1 adet 20 bölmeli enkoder disk, 1 adet sarı renkli DC motor,

Monte haldeki araba kiti (Tasarla & Üret bölümü için bazı gruplar farklı malzemelere ihtiyaç duyabilir.)

Bölümün İşlenişi:

Gözle: Arduino kodlama ortamında MPU 6050 açı sensörü ve LM 393 kızılötesi hız sensörü kullanımının, LM 393 kızılötesi hız sensörü verilerinin arabanın daha dengeli hareket etmesi için kullanımının gösterilmesi.

Uygula: Arduino kodlama ortamında MPU 6050 açı sensörü ve LM 393 kızılötesi hız sensörü kullanımının uygulanması, LM 393 kızılötesi hız sensörü verilerinin arabanın daha dengeli hareket etmesi için kullanımının uygulanması.

Tasarla: “Hayalimdeki arabayı oluşturuyorum” etkinliği için tanımlama ve fikir üretme adımlarının oluşturularak arabanın tasarımının yapılması.

Üret: Tasarımı yapılan “Hayalimdeki arabayı oluşturuyorum” etkinliğinin elektronik devresinin oluşturulması ve Arduino kodlama ortamı program kodunun oluşturulması

Değerlendir: Konu değerlendirmesi, günlük yaşam bağlamına transfer ve yansıtma etkinliği.

Ön Not:

Bu haftadan sonra öğrenciler fiili olarak projeleri üzerinde çalışmaya başlayacaklardır. Eğer rehber öğretmen öğrencilerin bundan önceki bölümlerde eksik konuları olduğunu düşünüyorsa bu bölümü Gözle ve Uygula etkinliklerini “1.4 Gözle- Hız Sensörü Yardımı ile Arabayı İleri Hareket Ettiriyorum” etkinliği ile sonlandırıp doğrudan Tasarla ve Üret (2.1 Tasarla ve Üret- Hayalimdeki Arabayı Oluşturuyorum) etkinliğine geçebilir. Buradan kazanacağı zaman ile rehber öğretmen öğrencilerin eksik olduğu noktaları vurgulayabilir veya öğrencilerin projeleri için onlara yardımcı olabilir. Eğer rehber öğretmen öğrencilerin bundan önceki bölümlerde eksik konuları olduğunu düşünmüyorsa, bu bölümde bulunan etkinlikleri zamanın elverdiği ölçüde bitirmeye çalışmalıdır.

1. GÖZLE VE UYGULA

1.1 Gözle ve Uygula- MPU 6050 ile Breadboard Görselleştirme

Malzeme Listesi:

Arduino Uno

Breadboard

1 adet MPU 6050 sensör

MPU 6050 altı eksenli sensör, ivme ve açısal hızölçer olarak görev yapar. MPU 6050'nin açısal hız ve ivme ölçme fonksiyonları kullanılarak x, y, z eksenlerindeki açısal hız ve yer çekimi ivmesi cinsinden ivme büyüklüğü değerleri bulunabilir. Bu büyüklükler kullanılarak x, y ve z eksenlerinde yapılan dönme hareketlerinin açı cinsinden miktarı gibi çeşitli hesaplamalar yapılabilir. Fakat beşinci ve altıncı sınıf öğrencileri hız, açısal hız, ivme ve üç boyutlu cisimlerin ekseni konularını henüz görmemiştir. Bu yüzden öğrencilere sadece MPU 6050'nin x, y ve z ekseni etrafındaki hareket için kullanılması anlatılacaktır. Öğrenciler x, y ve z eksenini henüz bilmediği için bu kavram onlara bir dikdörtgenler prizmasının veya küpün en, boy ve yüksekliği etrafındaki hareket olarak anlatılacaktır. Öğrencilerin bu hareketi somutlaştırması için https://www.youtube.com/watch?v=pQ24NtnaLl8 bağlantısında yer alan video kullanılabilir.

Bu etkinlikte amaç breadboard üzerine yerleştirilmiş MPU 6050'yi hareket ettirdikçe bilgisayar ekranındaki breadboardun görsel modelinin aynı hareketi yapmasını sağlamaktır. Bu görev için iki program gerekmektedir. Birinci program breadboardun görsel modelini ve onun hareketini sağlayan programdır. Bu programa EK-9 processingVisualizationMPU6050 Kaynak Kodu'ndan erişilebilir (bağlantı verilebilir). "processingVisualizationMPU6050" öğrencilere hazır olarak verilecektir. Program Processing programlama ortamında yazılmıştır. Öğrenciler programın detaylarını bilmek durumunda değildir. Öğrencilerin bu programın COM portu üzerinden Arduino kartından açı bilgilerini alıp bu bilgilere göre breadbordun görselleştirilmesini sağladığını bilmeleri yeterlidir. Etkinlikte kullanılacak ikinci program Arduino kartı içindir. Arduino kartı MPU 6050'den aldığı açı bilgilerini Arduino kartın bağlı bulunduğu COM portu üzerinden processingVisualizationMPU6050 isimli programa gönderecektir.

Programlar yazılmadan önce gerekli bağlantıların yapılması gerekmektedir. Rehber öğretmenler aşağıdaki bağlantıyı kullanabilir. Bağlantıdan da anlaşılacağı üzere MPU 6050, I2C'yi kullanmaktadır. Ayrıca MPU 6050'nin INT (interrupt) bacağı da Arduino kartın kesme/araya girme pini olan iki numaralı pinine bağlanmıştır. MPU 6050'nin kullanımı için bu bağlantı zorunlu değildir. Fakat bu derste kullanılacak kütüphane kesme/araya girme pinini kullandığı için bu bağlantı yapılmalıdır.

MPU 6050	Arduino Uno
VCC	5V
GND	GND
SCL	A5
SDA	A4
INT	2 (Dijital 2 pini)

Bu derste MPU 6050 ile kod yazmak için "MPU6050_tockn" kütüphanesi kullanılacaktır (https://github.com/Tockn/MPU6050_tockn). Bu kütüphane Arduino programlama ortamında yüklü olarak gelmemektedir. Rehber öğretmen ders esnasında bu kütüphanenin nasıl yüklendiğini öğrencilere göstermelidir ve onların kütüphaneyi yüklemelerini sağlamalıdır.

Bağlantılar yapıldıktan sonra asıl program çalıştırılmadan önce MPU 6050'nin kalibrasyonu gerçekleştirilmelidir. MPU 6050'nin kullanılmadan önce kalibrasyonunun yapılması cihazın daha doğru çalışmasını sağlar. Robotik Kodlama dersinde de benzer şekilde gyro sensörü kullanılmadan önce kalibre edilmiştir. MPU 6050'nin kalibrasyonu için üç adet parametreye ihtiyaç vardır. Bu iş için aşağıdaki program kullanılabilir.

```
#include <MPU6050_tockn.h>
#include <Wire.h>
MPU6050 mpu6050(Wire);
void setup() {
  Serial.begin(38400);
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
}
void loop() {
}
```

Program Arduino karta yüklenir ve ardından aşağıdaki resimde gösterildiği gibi seri port ekranı 38400 baud modunda açılır.

Resim 370. Seri Port Baud Rate Ayarı

Program çalıştırıldığında seri port ekranı aşağıdakine benzer bir çıktı verecektir. Burada bulunan X, Y ve Z değerleri kalibrasyon için kullanılacak parametrelerdir. Bu değerler kaydedilmelidir.

```
10:13:00.210 -> ========================================

10:13:00.243 -> Calculating gyro offsets

10:13:00.243 -> DO NOT MOVE MPU6050...

10:13:03.439 -> Done!

10:13:03.439 -> X : -7.94

10:13:03.439 -> Y : -1.51

10:13:03.439 -> Z : 0.39

10:13:03.439 -> Program will start after 3 seconds

10:13:03.473 -> ========================================
```

Artık Arduino programlama ortamında kod yazmaya başlanabilir. Yazılması gereken program ve açıklaması aşağıda verilmiştir. Bu programda MPU 6050'nin kalibrasyonu için kaydedilen değerler "mpu6050.setGyroOffsets(-7.94,-1.51,0.39);" komutu ile kalibrasyon için kullanılır.

```
#include <MPU6050_tockn.h> // MPU 6050 kütüphanesi
#include <Wire.h> //I2C için kullanılan kütüphane
MPU6050 mpu6050(Wire); // MPU 6050 nesnesi yaratılır

void setup() {
```

```
  Serial.begin(38400); //Seri bağlantı 38400 baud ile başlatılır
  Wire.begin(); //I2C başlangıç
  mpu6050.begin(); //MPU 6050 başlangıç
  mpu6050.setGyroOffsets(-7.94,-1.51,0.39); // MPU 6050 kalibrasyonu. Daha önce
}                                           // kaydedilen değerler kullanılır

void loop() {
  mpu6050.update(); // MPU 6050'den okunan değer güncellenir
  Serial.print(mpu6050.getAngleX()); // X ekseni – En için kullanılan açı
  Serial.print("<"); // processingVisualizationMPU6050 programı için gerekli
  Serial.print(mpu6050.getAngleY()); // Y ekseni – Boy için kullanılan açı
  Serial.print("<"); // processingVisualizationMPU6050 programı için gerekli
  Serial.println(mpu6050.getAngleZ()); // Z ekseni – Yükseklik için kullanılan açı
}
```

Programda önce gerekli kütüphaneler yüklenir ("MPU6050_tockn.h" ve "Wire.h") ve bir MPU 6050 nesnesi oluşturulur. Ardından setup içerisinde I2C, MPU 6050 ve seri bağlantı için gerekli ayarlar yapılır. Son olarak MPU 6050'den anlık olarak gelen açı değerleri (mpu6050.getAngleX(), mpu6050.getAngleY() ve mpu6050.getAngleZ()) seri port üzerinden processingVisualizationMPU6050 programına gönderilir. Gönderme işlemi yapılırken aşağıdaki yapı kullanılmıştır.

```
  Serial.print(mpu6050.getAngleX());
  Serial.print("<");
  Serial.print(mpu6050.getAngleY());
  Serial.print("<");
  Serial.println(mpu6050.getAngleZ());
```

Bu yapı sayesinde processingVisualizationMPU6050 programına gönderilen bilgi "XDeğeri<YDeğeri<ZDeğeri(Enter veya \n)" formatında olur. processingVisualizationMPU6050 programı verinin bu formatta geleceğini bilir ve ona göre veriyi XDeğeri, YDeğeri ve ZDeğeri olarak ayırır ve kullanır. Gönderilen veri farklı formatta ise processingVisualizationMPU6050 programı doğru çalışmaz.

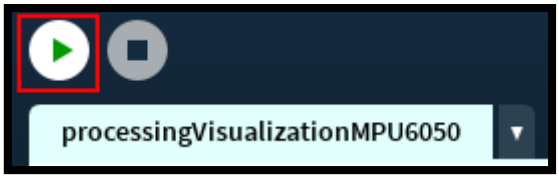
Dikkat

Öğrenciler X, Y, Z eksenlerini henüz derslerinde işlememiştir. Bunun yerine en, boy, yükseklik tabirinin kullanılması daha doğru olacaktır.

Bu program Arduino karta yüklendikten sonra processingVisualizationMPU6050 programı Processing IDE kullanılarak açılır. Program içerisinde bulunan aşağıda verilen satır düzenlenmelidir. COM8 yerine Arduino kartın bağlandığı COM portu yazılmalıdır.

```
arduinoPortu = new Serial(this, "COM8", 38400);
```

Gerekli port düzenlemesi yapıldıktan sonra aşağıdaki resimde gösterilen buton kullanılarak program çalıştırılır. Program ilk etapta hata verebilir. Bu durumda kapatılıp yeniden başlatılırsa sorun çözülecektir. Program doğru çalışana kadar aç-kapa işlemi gerçekleştirilir.

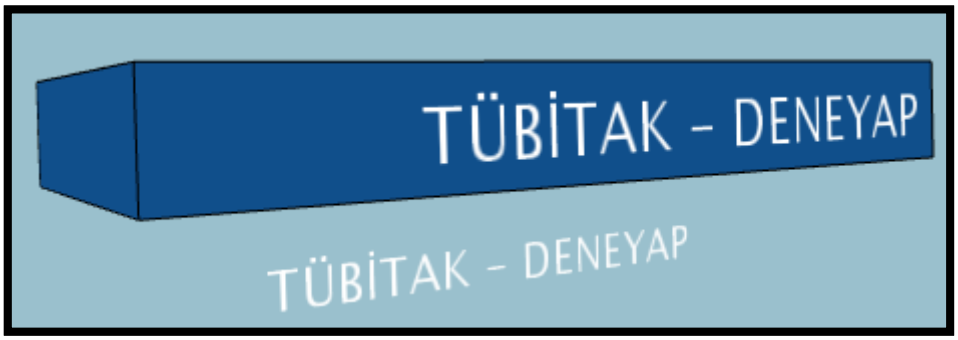


Resim 371. Program Çalıştırma Butonu

Dikkat

Önce Arduino programı yüklenip ardından "processingVisualizationMPU6050" çalıştırılmalıdır. Bu sıra tersine çevrilirse program çalışmaz.

Program çalıştırıldıktan sonra MPU 6050'nin üzerinde bulunduğu breadboard hareket ettirildiğinde ekrandaki breadboard modeli de hareket edecektir. Aşağıdaki resimde bu modelin ekran görüntüsü verilmiştir.



Resim 372. Model Ekran Görüntüsü

Dikkat

MPU 6050'nin breadboarda takılma yönü ters ise en-boy hareketi ters olarak görülebilir. MPU 6050'nin yönü düzeltildiğinde hareket de düzelir.

Rehber öğretmen devreyi kurarak tüm süreci öğrencilere anlatır. Öğrencilerin tüm bağlantıları sökmesini ve Arduino programını silmesini sağlar ve ardından öğrencilerin devreyi kurup programı çalıştırmasını sağlar. Son olarak öğrencilere programın ekranda çalışmasını deneyimlemeleri için gerekli zaman verilir. Öğrencilerin x, y, z eksenlerini henüz derslerinde görmediğinden bu deneyim onlar için önemlidir.

1.1 Uygula- MPU 6050 Açı Değerlerini Seri Port Ekranına Yazdırıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet MPU 6050 sensör

Bu etkinlikte amaç MPU 6050'den gelen X, Y ve Z açı değerlerini seri port ekranına yazdırmaktır. Bu iş için aşağıdaki kod kullanılabilir. Öğrenciler yazdıkları kodun çıktısını gözlemlemelidir. Öğrenciler breadbordu farklı yönlerde çevirerek açı değişikliklerini gözlemlerler.

```
#include <MPU6050_tockn.h>
```

```
#include <Wire.h>
MPU6050 mpu6050(Wire);
void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);// Burada daha önce bulunan değerler setGyroOffsets();
}                                    // komutu ile doğrudan kullanılabilir
void loop() {
  mpu6050.update();
  Serial.print("X: ");
  Serial.print(mpu6050.getAngleX());
  Serial.print(" Y:");
  Serial.print(mpu6050.getAngleY());
  Serial.print(" Z:");
  Serial.println(mpu6050.getAngleZ());
  delay(500);
}
```

Öğrenciler programı yazıp çalıştırdıktan sonra aşağıdaki soru sınıfça tartışılır.

Soru*:* Ekranda görülen X, Y ve Z değerlerinin dikdörtgenler prizmasındaki en, boy, yükseklik karşılıkları nasıl olur?

1.1 Uygula- MPU 6050 ile Led Yakma

Malzeme Listesi:

Arduino Uno
Breadboard
1 adet MPU 6050 sensör
2 adet led
2 adet 220 ohm direnç

Bu etkinlikte amaç MPU 6050 sağa doğru çevrildiğinde sağdaki ledi, sola doğru çevrildiğinde ise soldaki ledi yakmaktır. MPU 6050 ortaya doğru baktığında ise hiçbir led yanmamalıdır. Rehber öğretmen uygulama öncesi öğrencileri aşağıdaki konular hakkında uyarır:

(i) "calcGyroOffsets(true);" komutu MPU 6050'nin kalibrasyonunu otomatik olarak yapar fakat bu süreç her defasında yapıldığı ve uzun sürdüğü için zaman kaybına sebep olabilir,

(ii) Zaman kaybını engellemek için "setGyroOffsets();" komutu ile önceden bulunan değerler doğrudan kullanılabilir.

```
#include <MPU6050_tockn.h>
#include <Wire.h>
MPU6050 mpu6050(Wire);
int sagLed=12;
int solLed=13;

void setup() {
```

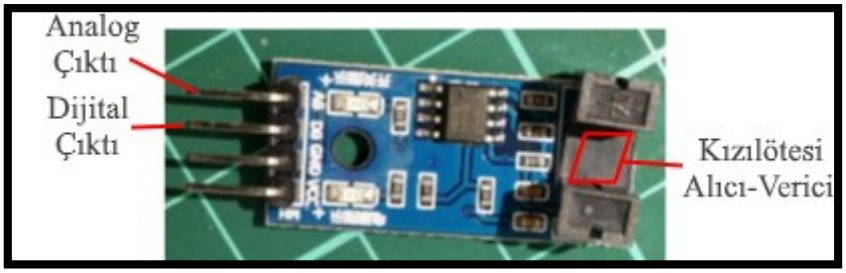
```
  Serial.begin(9600);
  Wire.begin();
  mpu6050.begin();
  mpu6050.setGyroOffsets(-7.86,-1.55,0.41); // önce calcGyroOffsets(true); komutu ile bulunmalı
  pinMode(sagLed,OUTPUT);
  pinMode(solLed,OUTPUT);
}

void loop() {
  mpu6050.update();
  Serial.println(mpu6050.getAngleZ());
  if (mpu6050.getAngleZ() > 30){
    digitalWrite(sagLed,HIGH);
    digitalWrite(solLed,LOW);
    }
  else if(mpu6050.getAngleZ() < -30){
    digitalWrite(sagLed,LOW);
    digitalWrite(solLed,HIGH);
    }
  else{
    digitalWrite(sagLed,LOW);
    digitalWrite(solLed,LOW);
    }
}
```

1.2 Gözle- Hız Sensörü ve Kesme (Interrupt) ile Sayaç Yapıyorum

Malzeme Listesi:

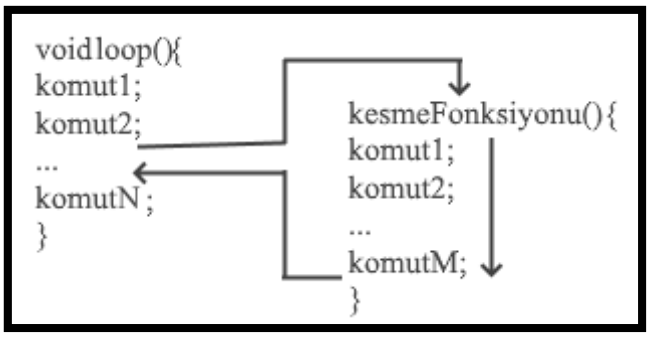
Arduino Uno
Breadboard
1 adet LM 393 kızılötesi hız sensörü

Bu etkinlikte amaç LM 393 kızılötesi hız sensörünü kullanarak sayaç yapmaktır (LM393 aslında modülde kullanılan entegredir). Aşağıdaki resimde görüldüğü gibi kızıl ötesi hız sensörünün kızılötesi alıcı ve vericisi bulunur. Alıcı verici arasında herhangi bir cisim bulunmadığında Dijital Çıktı bacağının değeri 0 olur. Alıcı ve vericinin arasına bir cisim girdiğinde Dijital Çıktı 1 değerini alır. Analog Çıktı da Dijital çıktıya benzer bir şekilde çalışır. Alıcı-verici arasında bir cisim yokken Analog Çıktı'dan yaklaşık olarak 0.15V değeri okunur. Alıcı-verici arasına bir cisim girdiğinde Analog Çıktı yaklaşık olarak 5V değer üretir.



Resim 373. Kızılötesi Sensör

Arduino kartın dijital pinleri kullanılarak, hız sensöründe durum değişikliği oluştuğunda (örneğin cisim var → cisim yok; cisim yok → cisim var) kızılötesi alıcı-verici arasından geçen cisim sayısı hesaplatılabilir. Fakat bu derste bu görevi yerine getirmek için Arduino kartın kesme/araya girme (Interrupt) pini kullanılacaktır. Arduino Uno kartın 2 ve 3 numaralı dijital pinleri kesme/araya girme pinleridir. Kesme pinleri ile ilişkilendirilmiş fonksiyonlar kesme pinlerinde meydana gelen voltaj değişimlerinde (0'dan 1'e veya 1'den 0'a) çalıştırılırlar. Daha net bir şekilde söylemek gerekirse kesme pinlerinde voltaj değişimi meydana geldiğinde ana program bulunduğu adımda bekletilir ve kesme pini ile ilişkilendirilmiş fonksiyonun çalışmasına öncelik verilir. Ana program bulunduğu yerden kesme pini ile ilişkilendirilmiş fonksiyona geçiş yapar ve bu fonksiyon içerisindeki komutları çalıştırır. Fonksiyon içerisindeki komutlar çalıştırıldıktan sonra ana program kaldığı yerden devam eder. Bu durum aşağıdaki resimde özetlenmiştir.



Resim 374. Kesme Fonksiyonu

Yukarıda bahsi geçtiği gibi kesme pinleri ile ilişkilendirmek üzere bir fonksiyon tanımlanmalıdır. Bu derste ArduinoBlocks ortamında fonksiyon tanımlanmıştır. Arduino programlama ortamını kullanarak da fonksiyon tanımlanabilir. Her fonksiyonun bir ismi bulunur. Burada sayaç yapılmak istendiğinden fonksiyonun ismi "cisimSayar" olarak belirlenebilir. Bir yapının fonksiyon olduğunu bildirmek için fonksiyon isminden sonra "()" ifadesi kullanılır. Yani fonksiyon ifadesi "cisimSayar()" olarak yazılır. Fonksiyon tanımı yapılırken fonksiyon ifadesinden önce bir anahtar kelime eklenmelidir. Farklı anahtar kelimeler kullanılabilir, fakat buradaki amaç doğrultusunda void anahtar kelimesi kullanılacaktır. Void ile birlikte fonksiyon ifadesi fonksiyon başlığını oluşturur. Fonksiyon başlığı yazıldıktan sonra fonksiyonun içerisinde olması istenen komutlar iki süslü parantez içerisine yazılır. Aşağıda yazılan kod bir fonksiyon örneğidir.

```
void cisimSayar(){ // Fonksiyon başlığı
sayac++; // Fonksiyon gövdesi

}
```

Arduino kodlama ortamında kullanılan setup ve loop da aslında bir fonksiyondur. Bu iki fonksiyon programın çalışması için zorunlu olarak tanımlanması gerekli fonksiyonlardır. Programcı amacı doğrultusunda istediği kadar fonksiyon tanımlayıp kullanabilir. cisimSayar fonksiyonu daha önceden tanımlanmış "sayac" değişkeninin değerini bir artırır (sayaç++;). "sayac" değişkeni yukarıdaki kodda gösterilmese de fonksiyonun dışında ve setup fonksiyonundan önce tanımlanmalıdır. Fonksiyon içerisinde de değişken tanımlanabilir. cisimSayar fonksiyonunda gerekli olmadığı için fonksiyon içerisinde değişken tanımlanmamıştır. Aşağıda "sayac" değişkeninin nerede tanımlandığı gösterilecektir.

Kesme pini ile ilişkilendirilecek olan cisimSayar fonksiyonu tanımlandıktan sonra kesme pini ile ilişkilendirme işlemi gerçekleştirilmelidir. Bu iş için aşağıdaki komut setup fonksiyonu içerisinde kullanılmalıdır.

attachInterrupt(digitalPinToInterrupt(2), cisimSayar, RISING);

Yukarıdaki komutta üç parametre bulunur. Birinci parametre Arduino kart üzerindeki iki numaralı kesme pininin kullanıldığını söyler. Üç numaralı kesme pini kullanmak istenilirse "digitalPinToInterrupt(3)" ifadesi yazılmalıdır. İkinci parametre ise kesme pini ile ilişkilendirilecek fonksiyonun ismidir. Üçüncü parametre ise kesme işleminin hangi durumda başlatılacağını bildirir. "RISING" (artarken) ifadesi iki numaralı kesme pinindeki değer 0'dan 1'e geçerken kesme fonksiyonunu çalıştırır. "RISING" yerine "FALLING" (azalırken) yazılırsa iki numaralı kesme pininde 1'den 0'a geçerken "cisimSayar" isimli fonksiyon çalıştırılır. Buradaki durumda "RISING" ile "FALLING" arasında bir fark bulunmamaktadır. Bu yüzden herhangi biri kullanılabilir.

Dikkat

Arduino Uno kartında sadece iki ve üç numaralı dijital pinler kesme işlemi için kullanılabilir.

Nesne sayma görevi için aşağıdaki kod kullanılabilir. Aşağıdaki kodda cisimSayar isimli bir fonksiyon tanımlanmıştır. Bu fonksiyon daha önceden tanımlanmış olan sayaç değişkeninin değerini bir artırır. Setup fonksiyonu içerisinde iki numaralı dijital kesme pini cisimSayar fonksiyonu ile ilişkilendirilmiştir. Yani iki numaralı pine gönderilen değer dijital 0'dan dijital 1 değerine dönerken cisimSayar isimli fonksiyon çalıştırılır.

```
int sayac=0;
void setup() {
  attachInterrupt(digitalPinToInterrupt(2),cisimSayar,RISING);
  Serial.begin(9600);
}
void loop() {
  Serial.println(sayac);
  delay(500);
}
void cisimSayar(){
  sayac++;
}
```

Yazılan Arduino kodunun çalıştırılması için uygun bağlantıların yapılması gerekmektedir. Bağlantılar aşağıdaki şekilde yapılabilir.

LM 393	Arduino Kartı
VCC	5V
GND	GND
A0	2 (iki numaralı dijital pin)

Dikkat

Bu etkinlikte LM393 hız sensörünün analog çıktı bacağı olan A0 bacağı kullanılmıştır. Bu etkinlikte dijital çıktı bacağı olan D0'ın kullanılması daha uygun olacağı düşünülebilir. Fakat yapılan denemelerde D0 bacağı kararsız çalıştığı için A0 bacağı tercih edilmiştir.

Rehber öğretmen öğrencilere açıklayarak bağlantıyı yapar ve kodu yazar. Kodu çalıştırıp seri port ekranında hız sensörünün kızılötesi alıcı-vericisi arasından her geçen cisim için sayacın bir arttığını gösterir.

1.2 Uygula- DC Motora Bağlı Tekerleğin Döndüğü Tur Sayısını Bulan Program

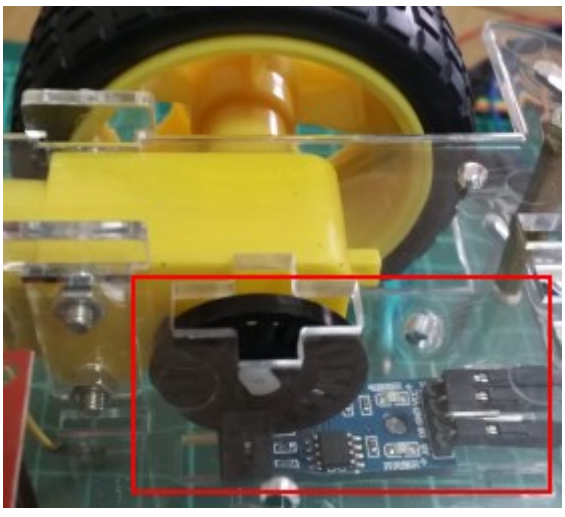
Malzeme Listesi:

Arduino Uno
Breadboard
1 adet LM393 hız sensörü
1 adet 20 bölmeli enkoder disk
1 adet sarı renkli DC motor

Bu etkinlikte amaç LM393 hız sensörü ve 20 bölmeli enkoder disk kullanarak DC motora takılı tekerleğin kaç tur döndüğünü seri port ekranına yazmaktır. Aşağıdaki resimde 20 bölmeli enkoder disk görülmektedir.

Resim 375. 20 Bölmeli Enkoder Disk

Öğrenciler bu etkinliği gerçekleştirebilmek için hız sensörünü aşağıdaki resimlerde görüldüğü gibi daha önce inşa ettikleri arabaya eklemelidirler. Aslında bu etkinlik enkoder ve hız sensörü arabaya eklenmeden de yapılabilir, fakat bir sonraki etkinlikte araba için enkoder ve hız sensörleri kullanılacağı için öğrencilerin iki enkoder ve hız sensörünü arabalarına eklemeleri daha uygun olacaktır. Aşağıdaki resimlerde hem arabanın üstünden hem de arabanın altından hız sensörü ve enkoderin nasıl yerleştirildiği gösterilmiştir. Hız sensörü şasinin üst katına çift taraflı bant ile yapıştırılmıştır. Aynı zamanda hız sensörünün vida deliğine karşılık gelecek şekilde şasinin üst katında bir delik açılmıştır. Rehber öğretmen daha uygun olacağını düşündüğü durumlarda öğrencilere böyle bir delik açtırarak vida ve somun aracılığı ile hız sensörünün monte edilmesini sağlayabilir.



Enkoder-Hız Sensörü Alttan Çekim

Enkoder-Hız Sensörü Üstten Çekim

Resim 376. Enkoder-Hız Sensörü Şasi Bağlantısı

Bu etkinlik için aşağıdaki kod kullanılabilir. Kodun tekerleğin tur sayısını bulması için öğrencilerin tekerleği elleriyle döndürmesi gerekir.

```
int sayac=0;
int turSayisi=0;
void setup() {
  attachInterrupt(digitalPinToInterrupt(2),bolmeSayar,RISING);
  Serial.begin(9600);
}
void loop() {
  if (sayac>=20){
    turSayisi=turSayisi+1;
    sayac=0;
    }
  Serial.print("Sayac: ");
  Serial.println(sayac);
  Serial.print("Tekerlek ");
  Serial.print(turSayisi);
  Serial.println(" defa dondu");
  delay(500);
}
void bolmeSayar(){
  sayac++;
}
```

Dikkat

Öğrenciler tekerleği döndürürken fazla titretirlerse kod doğru çalışmayabilir.

1.3 Gözle- Motor Gücü ile Dönen Tekerlek için Hız Sensörünün Saydığı Bölme Sayısını Bulan Program

Malzeme Listesi:

Arduino Uno
Breadboard
2 adet LM393 hız sensörü
Monte haldeki araba kiti
9V pil

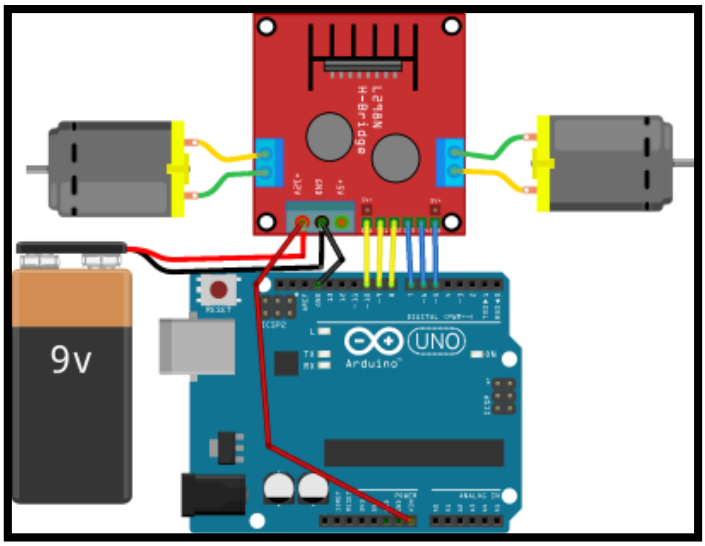
Bir önceki etkinlikte (DC Motora Bağlı Tekerleğin Döndüğü Tur Sayısını Bulan Program) tekerleğin kaç tur döndüğü hesaplanırken öğrenciler tekerleği el ile döndürmüşlerdir. Bu etkinlikte teker motor gücüyle dönerken hız sensörünün kaç adet bölme saydığı seri port ekranına yazılacaktır.

Hatırlanacağı üzere DC motor sürücüsünün üzerinde ENA, IN1, IN2, IN3, IN4 ve ENB bacakları bulunmaktadır. Bunlardan ENA, IN1 ve IN 2 sol motorun yönünün ve hızının ayarlanması için ENB, IN3 ve IN4 ise sağ motorun yönünün ve hızının ayarlanması için kullanılır. Bu etkinlikte sadece sol motor kullanılacağı için sağ motordan bahsedilmeyecektir. Fakat sol motor için anlatılanlar sağ motor için de geçerlidir.

ENA bacağı motorun dönüş hızını belirlemek için kullanılır ve bu yüzden Arduino kartın PWM pinlerinden birine bağlanmalıdır. IN1 ve IN2 bacakları sol motorun dönüş yönünü belirler. IN1 HIGH (1) ve IN2 LOW (0) iken motor ileriye doğru; IN1 LOW(0) ve IN2 HIGH (1) iken motor geriye doğru

döner. Pratikte kullanırken ileriye veya geriye değerlerini ezberlemektense 0-1 ve 1-0 değerlerinin zıt yönlü hareketi sağladığını bilmek yeterlidir. Yazılacak programda veya bağlantılarda yapılacak küçük değişikliklerle yön belirlenebilir. L298N motor sürücüsü ile motor freni yapmak için ENA HIGH, IN1 HIGH ve IN2 HIGH olarak ayarlanmalıdır (başka şekillerde de fren yapılabilir, fakat şimdilik bu yeterlidir).

Aşağıdaki resimde iki motor için bağlantı şeması verilmiştir. Bu etkinlikte sadece sol motor kullanılacaktır fakat bir sonraki etkinlikte iki motor da (öğrencilerin araba oluşturacağı etkinlikte) kullanılacağı için öğrencilerin arabayı aşağıdaki bağlantılara göre şimdiden oluşturmaları daha pratik olacaktır.



Resim 377 Örnek Devre Şeması

Bu etkinlikte motorun ileri, geri hareketi ve freni fonksiyonlar kullanılarak yapılacaktır. Bunun için parametre alan fonksiyonların nasıl tanımlandığını bilmek gerekmektedir. Bazı durumlarda fonksiyonlara parametre göndermek gerekir. Örneğin motorun ileri doğru hareket etmesini sağlayan bir fonksiyon tanımlamak istersek, ileriye doğru gidilecek hızın ne olduğunu fonksiyona bildirmemiz gerekir. Hız için 0-255 arasında değerler verilebilir (düşük hız değerleri için motor dönmeyecektir). Hız parametresi fonksiyonunun parantezleri içerisinde fonksiyona bildirilir. Aşağıda bu duruma örnek verilmiştir. Fonksiyona parametre olarak gönderilecek olan hiz'in tipi de parantez içerisinde belirtilmelidir.

```
void ileriGit(int hiz){
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite(ENA,hiz);
  }
```

Daha sonra bu fonksiyonu programda herhangi bir yerden 150 değeri ile çağırmak için, yani motorun 150 hızıyla ileriye doğru hareket etmesini sağlamak için "ileriGit(150);" komutu verilir. 150 yerine istenen değer yazılarak hız ayarlaması yapılabilir. Bu etkinlikte aşağıdaki kod kullanılabilir. Rehber öğretmen açıklayarak devreyi kurar. Ardından kodu açıklayarak yazar ve çalıştırır. Son olarak öğrencilerin aynı devreyi kurup kodu yazmasını sağlar.

```
int ENA=10; //PWM olmalı
```

```
int IN1=9;
int IN2=8;
int sayac=0;

void setup() {
  attachInterrupt(digitalPinToInterrupt(2),bolmeSayar,RISING);
  Serial.begin(9600);
  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
}
void loop() {
  ileriGit(240); //240 hızı ile
  delay(1000); //1 saniye ileri gider
  frenYap();
  Serial.print("Sayac: ");
  Serial.println(sayac);
  delay(500);
  geriGit(100); // 100 hızı ile
  delay(1000); // 1 saniye geri gider
  frenYap();
  Serial.print("Sayac: ");
  Serial.println(sayac);
  delay(500);
}
void ileriGit(int hiz){
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite(ENA,hiz);
  }
void geriGit(int hiz){
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,hiz);
  }
void frenYap(){
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,255);
  }
void bolmeSayar(){
  sayac++;
}
```

Dikkat

Bu etkinlikte 9V pil kullanılacağı için pillerin önceden şarj edilmesi gerekmektedir.

1.3 Uygula- Arabayı Hareket Ettirirken Sol ve Sağ Teker İçin Bölme Sayısını Hesaplıyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Monte haldeki araba kiti
2 adet LM393 hız sensörü
9V pil

Bu etkinlikte amaç arabanın her iki motoru ileri ve geri hareket ederken hız sensörünün saydığı bölme sayısını ekrana yazmaktır. Bu görev için aşağıdaki kod kullanılabilir. Kodun çalışması için ENB bacağı Arduino kartın 5 numaralı dijital pinine; IN4 bacağı Arduino kartın 6 numaralı dijital pinine ve IN3 bacağı Arduino kartın 7 numaralı dijital pinine bağlanmalıdır. Buna ek olarak sağ tekere monte edilen hız sensörünün analog bacağından alınan çıktı Arduino kartın 3 numaralı kesme/dijital pinine bağlanmalıdır.

```
int ENA=10; //PWM olmalı
int IN1=9;
int IN2=8;
int ENB=5; //PWM olmalı
int IN4=6;
int IN3=7;
int sayacSol=0;
int sayacSag=0;

void setup() {
  attachInterrupt(digitalPinToInterrupt(2),bolmeSayarSolTeker,RISING);
  attachInterrupt(digitalPinToInterrupt(3),bolmeSayarSagTeker,RISING);
  Serial.begin(9600);
  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENB,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
}
void loop() {
  ileriGit(150);
  delay(1000);
  frenYap();
  Serial.print("Sol Teker Sayac: ");
  Serial.println(sayacSol);
  Serial.print("Sag Teker Sayac: ");
  Serial.println(sayacSag);
  delay(500);
  geriGit(150);
  delay(1000);
  frenYap();
  Serial.print("Sol Teker Sayac: ");
```

```
  Serial.println(sayacSol);
  Serial.print("Sag Teker Sayac: ");
  Serial.println(sayacSag);
  delay(500);
}
void ileriGit(int hiz){
  digitalWrite(IN1,HIGH); // Sol Motor
  digitalWrite(IN2,LOW);
  analogWrite(ENA,hiz);
  digitalWrite(IN3, HIGH); // Sag Motor
  digitalWrite(IN4,LOW);
  analogWrite(ENB,hiz);
  }
void geriGit(int hiz){
  digitalWrite(IN1,LOW); // Sol Motor
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,hiz);
  digitalWrite(IN3, LOW); // Sag Motor
  digitalWrite(IN4,HIGH);
  analogWrite(ENB,hiz);
  }
void frenYap(){
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,255);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4,HIGH);
  analogWrite(ENB,255);
  }
void bolmeSayarSolTeker(){
  sayacSol++;
}
void bolmeSayarSagTeker(){
  sayacSag++;
}
```

1.4 Gözle- Hız Sensörü Yardımı ile Arabayı İleri Hareket Ettiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Monte haldeki araba kiti
2 adet LM393 hız sensörü

Bu etkinlikte amaç hız sensörü ile arabayı istenilen hızda ve istenilen enkoder bölme sayısında (enkoderde 20 bölme bulunmaktadır) ileri doğru hareket ettirmektir. Bu etkinlik için iki parametreli fonksiyon kullanılacaktır. Bu yüzden öncelikle iki parametreli fonksiyonların nasıl tanımlandığı üzerinde durulmalıdır. Aşağıda iki parametre alan ileriBolmeGit isimli fonksiyon örnek olarak verilmiştir. Fonksiyon parametreleri virgül ile ayrılmalıdır ve her parametrenin cinsi belirtilmelidir. ileriBolmeGit fonksiyonu belirlenen hız değerinde sol ve sağ motoru verilen bölme (enkoderdeki çubuk)

sayısı kadar ileri götürür ve motorları durdurur. Motorlar durduktan sonra sol ve sağ motordaki dönme sayısı bölme cinsinden ekrana yazılır.

```
void ileriBolmeGit(int bolmeSayisi, int hiz){
  sayacSol=0;
  sayacSag=0;
  digitalWrite(IN1,HIGH); //Sol motor ileri
  digitalWrite(IN2,LOW);

  digitalWrite(IN3, HIGH); //Sag motor ileri
  digitalWrite(IN4,LOW);
  while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
     analogWrite(ENA,hiz); //istenen bölme sayısı kadar
     analogWrite(ENB,hiz); // ve istenen hızda ilerle
  }
  analogWrite(ENA,0); // Sol motor dur
  analogWrite(ENB,0); // Sag motor dur
  Serial.print("Sol Sayac: ");
  Serial.println(sayacSol);
  Serial.print("Sag Sayac:");
  Serial.println(sayacSag);
  sayacSol=0;
  sayacSag=0;
}
```

Bu programın her zaman doğru çalışması için sayacSol ve sayacSag değişkenlerinin int olarak tanımlanması yeterli değildir. Kesme (Interrupt) pini ile ilişkilendirilen değişkenlerin “volatile” olarak tanımlanması gerekir. Bu tanımlama şekli aşağıdaki kodda örneklendirilmiştir. Kod arabayı 150 hızıyla 60 enkoder bölmesi (yani 3 tam tur – 3x20) kadar ilerletir. Rehber öğretmen kodu öğrencilere açıklayarak çalıştırır. Ardından öğrencilerin aynı kodu yazarak çalıştırmalarını ve kodun çıktısını gözlemelerini sağlar.

```
int ENA=10; //PWM olmalı
int IN1=9;
int IN2=8;
int ENB=5; //PWM olmalı
int IN4=6;
int IN3=7;
volatile int sayacSol=0;
volatile int sayacSag=0;

void setup() {
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(2),bolmeSayarSolTeker,RISING);
  attachInterrupt(digitalPinToInterrupt(3),bolmeSayarSagTeker,RISING);
  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENB,OUTPUT);
```

```
 pinMode(IN3,OUTPUT);
 pinMode(IN4,OUTPUT);
}
void loop() {
 ileriBolmeGit(60,150);
 while(1);
}
void ileriBolmeGit(int bolmeSayisi, int hiz){
 sayacSol=0;
 sayacSag=0;
 digitalWrite(IN1,HIGH); //Sol motor ileri
 digitalWrite(IN2,LOW);
 digitalWrite(IN3, HIGH); //Sag motor ileri
 digitalWrite(IN4,LOW);
 while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
    analogWrite(ENA,hiz); //istenen bölme sayısı kadar
    analogWrite(ENB,hiz); // ve istenen hızda ilerle
 }
 analogWrite(ENA,0); // Sol motor dur
 analogWrite(ENB,0); // Sag motor dur
 Serial.print("Sol Sayac: ");
 Serial.println(sayacSol);
 Serial.print("Sag Sayac:");
 Serial.println(sayacSag);
 sayacSol=0;
 sayacSag=0;
}
void bolmeSayarSolTeker(){
 sayacSol++;
}
void bolmeSayarSagTeker(){
 sayacSag++;
}
```

1.4 Uygula- Hız Sensörü Yardımı ile Arabayı İleri ve Geri Hareket Ettiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Monte haldeki araba kiti
2 adet LM393 hız sensörü

Bir önceki etkinlikte (Hız Sensörü Yardımı ile Arabayı İleri Hareket Ettiriyorum) araba sadece ileri hareket ettirilmiştir. Bu etkinlikte arabanın geri hareket etmesini sağlayacak fonksiyon tanımlanacaktır ve arabanın ileri ve geri hareket etmesi sağlanacaktır. Bu görev için aşağıdaki kod kullanılabilir.

```
int ENA=10; //PWM olmalı
int IN1=9;
int IN2=8;
int ENB=5; //PWM olmalı
```

```
int IN4=6;
int IN3=7;
volatile int sayacSol=0;
volatile int sayacSag=0;

void setup() {
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(2),bolmeSayarSolTeker,RISING);
  attachInterrupt(digitalPinToInterrupt(3),bolmeSayarSagTeker,RISING);
  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENB,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
}
void loop() {
  ileriBolmeGit(60,150);
  delay(1000);
  geriBolmeGit(60,150);
  while(1);
}
void ileriBolmeGit(int bolmeSayisi, int hiz){
  sayacSol=0;
  sayacSag=0;
  digitalWrite(IN1,HIGH); //Sol motor ileri
  digitalWrite(IN2,LOW);
  digitalWrite(IN3, HIGH); //Sag motor ileri
  digitalWrite(IN4,LOW);
  while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
     analogWrite(ENA,hiz); //istenen bölme sayısı kadar
     analogWrite(ENB,hiz); // ve istenen hızda ilerle
  }
  analogWrite(ENA,0); // Sol motor dur
  analogWrite(ENB,0); // Sag motor dur
  Serial.print("Sol Sayac: ");
  Serial.println(sayacSol);
  Serial.print("Sag Sayac:");
  Serial.println(sayacSag);
  sayacSol=0;
  sayacSag=0;
}
void geriBolmeGit(int bolmeSayisi, int hiz){
  sayacSol=0;
  sayacSag=0;
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
```

```
  digitalWrite(IN3, LOW);
  digitalWrite(IN4,HIGH);
  while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
     analogWrite(ENA,hiz);
     analogWrite(ENB,hiz);
  }
  analogWrite(ENA,0);
  analogWrite(ENB,0);
  Serial.print("Sol Sayac: ");
  Serial.println(sayacSol);
  Serial.print("Sag Sayac:");
  Serial.println(sayacSag);
  sayacSol=0;
  sayacSag=0;
}
void bolmeSayarSolTeker(){
  sayacSol++;
}
void bolmeSayarSagTeker(){
  sayacSag++;
}
```

1.5 Gözle- Hız Sensörü Yardımı ile Arabayı Daha Dengeli İleri Hareket Ettiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Monte haldeki araba kiti
2 adet LM393 hız sensörü

Bundan önceki etkinlikte hız sensörü kullanılmasına rağmen bazı arabalar sağa veya sola çekerek ilerlemiştir. Bu etkinlikte amaç arabanın daha dengeli bir şekilde ileriye doğru hareket etmesini sağlamaktır. Bu görevi gerçekleştirmek için hız sensöründen gelen değer kullanılarak motorlardan hızlı gidenin hızı yavaşlatılacaktır. Böylece araba daha dengeli hareket edecektir.

Bu etkinlik için fark isminde yeni bir değişken tanımlanmalıdır. Bu değişken ile sol sayaç ile sağ sayaç arasındaki fark bulunup (fark=sayacSol-sayacSag), daha hızlı giden tekerin bu fark oranında hızı azaltılmalıdır. Fark değişkeninin değeri pozitifse yani sol teker daha hızlı gidiyorsa fark ile motorun güncel hızı oranında sol teker yavaşlatılmalıdır. Fark değişkeninin değeri negatifse yani sağ teker daha hızlı gidiyorsa fark ile motorun güncel hızı oranında sağ teker yavaşlatılmalıdır. Bu görev için aşağıdaki kod kullanılabilir. Aşağıdaki kod bu iş için kullanılabilir

```
fark=sayacSol-sayacSag;
   if(fark>0){
     analogWrite(ENA,hiz-fark*(hiz/5));
     }
   else{
     analogWrite(ENA,hiz);
       }
   if(fark<0){
```

```
    analogWrite(ENB,hiz+fark*(hiz/5));
    }
   else{
    analogWrite(ENB,hiz);
    }
```

Bu kodda fark değişkeni “hiz/5” ile çarpılarak gerekli teker yavaşlatılmıştır. Bunun nedeni güncel hıza bağlı olarak oluşan farkı kapatma çabasıdır. Burada hız 5 yerine başka bir sayıya da bölünebilir. Öğrenciler bu değeri kendi durumlarına göre ayarlayabilirler. Aslında burada hız/5 yerine 10 gibi bir sabit de kullanılabilir. Öğrenciler bir sabit kullanmak istiyorsa bu sabiti kendi durumlarına göre ayarlayıp bulmalıdırlar. Bu etkinlik için aşağıdaki kod kullanılabilir. Rehber öğretmen aşağıdaki kodu öğrencilere yazarak açıklar ve aynı kodu öğrencilerin yazıp çalıştırmasını sağlar.

```
int ENA=10; //PWM olmalı
int IN1=9;
int IN2=8;
int ENB=5; //PWM olmalı
int IN4=6;
int IN3=7;
int fark=0;
volatile int sayacSol=0;
volatile int sayacSag=0;

void setup() {
 attachInterrupt(digitalPinToInterrupt(2),bolmeSayarSolTeker,RISING);
 attachInterrupt(digitalPinToInterrupt(3),bolmeSayarSagTeker,RISING);
 pinMode(ENA,OUTPUT);
 pinMode(IN1,OUTPUT);
 pinMode(IN2,OUTPUT);
 pinMode(ENB,OUTPUT);
 pinMode(IN3,OUTPUT);
 pinMode(IN4,OUTPUT);
}
void loop() {
 ileriBolmeGit(120,150);
 while(1);
}
void ileriBolmeGit(int bolmeSayisi, int hiz){
 sayacSol=0;
 sayacSag=0;
 fark=0;
 digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3, HIGH);
 digitalWrite(IN4,LOW);
 while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
  fark=sayacSol-sayacSag;
  if(fark>0){
   analogWrite(ENA,hiz-fark*(hiz/5));
```

```
    }
   else{
    analogWrite(ENA,hiz);
     }
   if(fark<0){
    analogWrite(ENB,hiz+fark*(hiz/5));
    }
   else{
    analogWrite(ENB,hiz);
    }
 }
  analogWrite(ENA,0);
  analogWrite(ENB,0);
  sayacSol=0;
  sayacSag=0;
  fark=0;
}
void bolmeSayarSolTeker(){
 sayacSol++;
}
void bolmeSayarSagTeker(){
 sayacSag++;
}
```

1.4 Uygula- Hız Sensörü Yardımı ile Arabayı Daha Dengeli İleri ve Geri Hareket Ettiriyorum

Malzeme Listesi:

Arduino Uno
Breadboard
Monte haldeki araba kiti
2 adet LM393 hız sensörü

Bir önceki etkinlikte araba sadece ileri doğru hareket etmiştir. Bu etkinlikte öğrenciler arabanın geri doğru hareket etmesi için gerekli kodu yazacaklardır ve arabayı ileri ve geri doğru hareket ettireceklerdir.

```
int ENA=10; //PWM olmalı
int IN1=9;
int IN2=8;
int ENB=5; //PWM olmalı
int IN4=6;
int IN3=7;
int fark=0;
volatile int sayacSol=0;
volatile int sayacSag=0;
void setup() {
 attachInterrupt(digitalPinToInterrupt(2),bolmeSayarSolTeker,RISING);
 attachInterrupt(digitalPinToInterrupt(3),bolmeSayarSagTeker,RISING);
 pinMode(ENA,OUTPUT);
```

```
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENB,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
}
void loop() {
  ileriBolmeGit(120,150);
  delay(3000);
  geriBolmeGit(120,150);
  delay(3000);
  while(1);
}
void ileriBolmeGit(int bolmeSayisi, int hiz){
  sayacSol=0;
  sayacSag=0;
  fark=0;
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4,LOW);
  while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
    fark=sayacSol-sayacSag;
    if(fark>0){
      analogWrite(ENA,hiz-fark*(hiz/5));
      }
    else{
      analogWrite(ENA,hiz);
        }
    if(fark<0){
      analogWrite(ENB,hiz+fark*(hiz/5));
      }
    else{
      analogWrite(ENB,hiz);
      }
  }
    analogWrite(ENA,0);
    analogWrite(ENB,0);
    sayacSol=0;
    sayacSag=0;
    fark=0;
}
void geriBolmeGit(int bolmeSayisi, int hiz){
  sayacSol=0;
  sayacSag=0;
  fark=0;
  digitalWrite(IN1,LOW);
```

```
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4,HIGH);
  while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi){
    fark=sayacSol-sayacSag;
    if(fark>0){
      analogWrite(ENA,hiz-fark*(hiz/5));
      }
    else{
      analogWrite(ENA,hiz);
    }
    if(fark<0){
      analogWrite(ENB,hiz+fark*(hiz/5));
      }
    else{
      analogWrite(ENB,hiz);
    }
  }
    analogWrite(ENA,0);
    analogWrite(ENB,0);
    sayacSol=0;
    sayacSag=0;
    fark=0;
}
void bolmeSayarSolTeker(){
  sayacSol++;
}
void bolmeSayarSagTeker(){
  sayacSag++;
}
```

Dikkat

Arabanın daha dengeli hareket etmesi için kodda çeşitli iyileştirmeler yapılabilir. Burada kullanılan basit bir iyileştirme algoritmasıdır. Fakat bu ders kapsamında yeterlidir. Buna ek olarak öğrencilere sarhoş tekerin kalitesinin arabanın daha dengeli hareket etmesi için önemli olduğu vurgulanır.

2.TASARLA VE ÜRET

2.1 Tasarla ve Üret- Hayalimdeki Arabayı Oluşturuyorum

Bu etkinlikte amaç öğrencilerin yaratıcılıklarını kullanarak bir araba ve bu araba için bir kod tasarlamasıdır. Öğrenciler etkinliğin sınırını kendileri belirleyecektir. Öğrenciler araba malzemesi ve istedikleri diğer devre elemanlarını kullanarak hayalindeki aracı tasarlayabilir. Birden fazla grup bir araya gelebilir.

3. DEĞERLENDİR

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. "1.5 Gözle- Hız Sensörü Yardımı ile Arabayı Daha Dengeli İleri Hareket Ettiriyorum" etkinliğinde program içerisinde aşağıdaki komutta ve operatörü (&&) kullanılmıştır:

while (sayacSol<bolmeSayisi && sayacSag<bolmeSayisi)

Bunun yerine program içerisinde veya operatörü (||) kullanılmış olsaydı kod nasıl çalışırdı?

while (sayacSol<bolmeSayisi || sayacSag<bolmeSayisi)

2. Bu bölümde gördüğünüz konuları üretim sektöründe/sanayide nerede kullanabilirsiniz?

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
    - Hız sensörünün kullanımını anlamakta sıkıntı yaşadım.
    - Arabanın daha dengeli gitmesini sağlayamadım.
    - Bu bölümde araba ile ilgili algoritmaları anlamakta sıkıntı yaşadım.
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

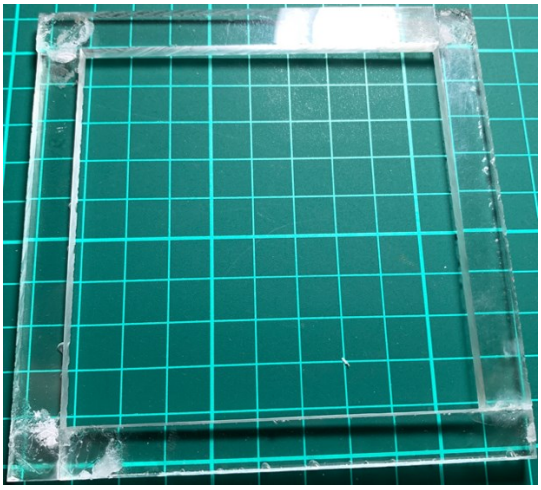
Proje Hazırlıyorum

Bu haftada, proje çalışmalarına devam edilmesi önerilmektedir. Projenin "fikir üretme" süreci için planlamalar yapılmalıdır. Detaylar için "EK-10-Proje Hazırlıyorum" bölümü incelenebilir.

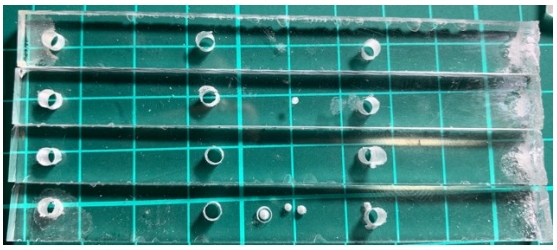
Sonraki iki hafta boyunca sadece proje çalışmalarına devam edilmesi önerilmektedir. Projenin "geliştirme ve test etme" ve "sunum" süreci için planlamalar yapılmalıdır. Detaylar için "EK-10-Proje Hazırlıyorum" bölümü incelenebilir.

EK-1-LED Küp Kurulum Yönergesi (Bölüm-2)

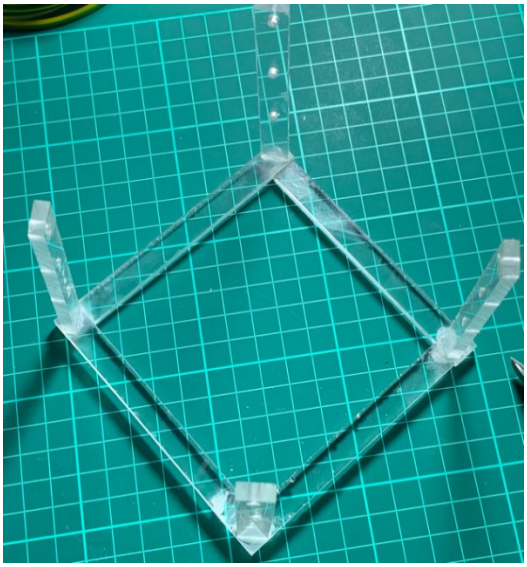
1. Led küp düzeneğinin kaidesini oluşturacak sekiz adet 10 cm uzunluğunda şeffaf pleksi parça bulunmaktadır. Bu parçalardan dördü aşağıdaki resimde gösterildiği şekilde birleştirilir ve birleşim noktalarından birbirlerine yapıştırılır. Bu bölüm led küpü kaidesinin tabanını oluşturacaktır.



2. Kalan dört parça ile kaidenin kolonları oluşturulacaktır. Kaide kullanılarak üç katlı bir led küpü oluşturulacağından her bir kolon üzerinde üç adet delik açılmalıdır. Aşağıdaki resimde gösterildiği gibi delikler açılır. Delikler açılırken ilk deliğin sol/üst/uç kısma yakın olması ve diğer iki deliğin üçer cm ara ile açılmasına dikkat edilir.

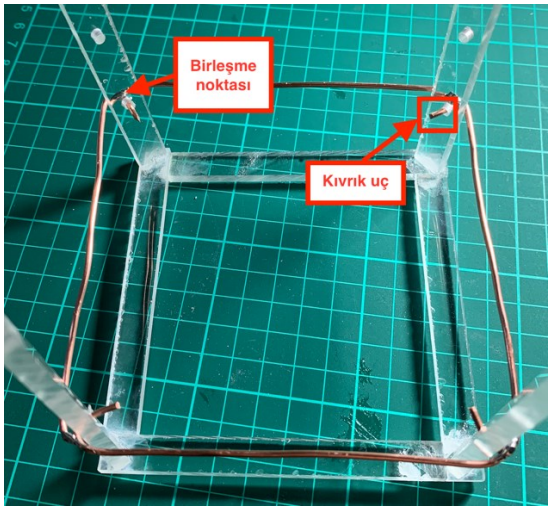


3. Taban ve kolonlar resimde gösterildiği gibi yapıştırılır. Yapıştırma işlemini yaparken kolonların uç kısmına açılan deliğin kaidenin üst tarafında olmasına dikkat edilmelidir.

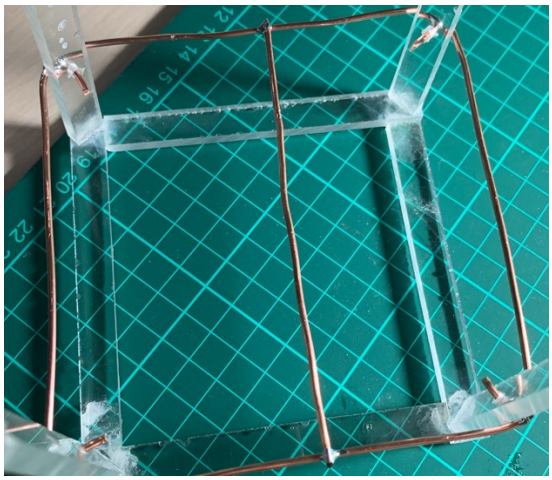


4. Aşağıdaki sol resimde gösterildiği gibi yaklaşık olarak 11-12 cm uzunluğunda kalın elektrik telleri kesilir. Daha sonra bu tellerin uçları sağ resimde gösterildiği gibi kıvrılır. Burada yer alan teller bir kat içindir. Üç kat için aynı işlemin ve yapılması ve tellerin hazırlanması gerekmektedir.

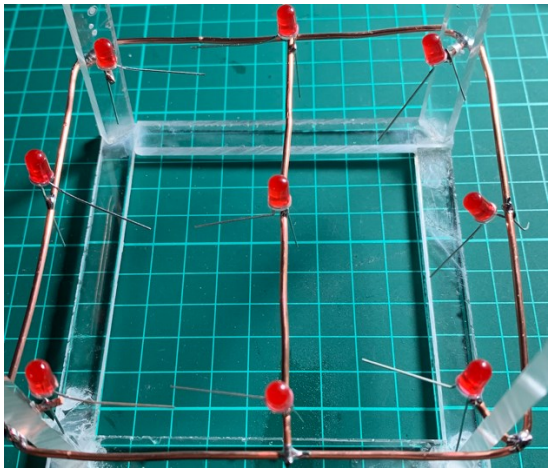
5. Uçları kıvrılan teller en alt sırada yer alan deliklerden geçirilir. Birleşme noktalarından lehimlenir. Kıvrık uçların deliklerden geçirilerek kaidenin iç kısmında konumlanmasına dikkat edilmelidir.



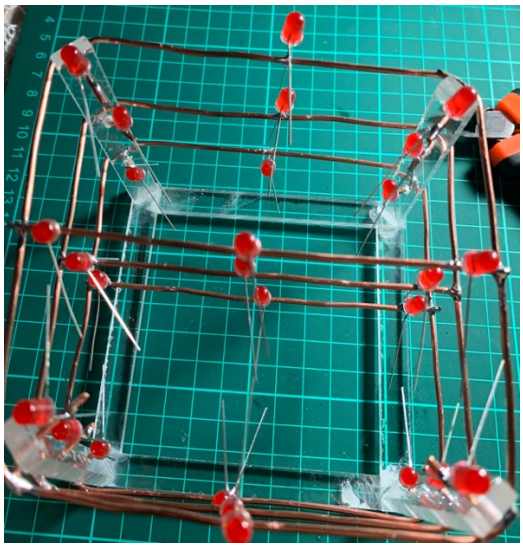
6. 11 cm uzunluğunda başka bir tel kesilerek, resimde gösterildiği gibi, kaide üzerine daha önceden yerleştirilmiş tellerin iki tanesinin orta noktalarından birleşecek şekilde lehimlenir.

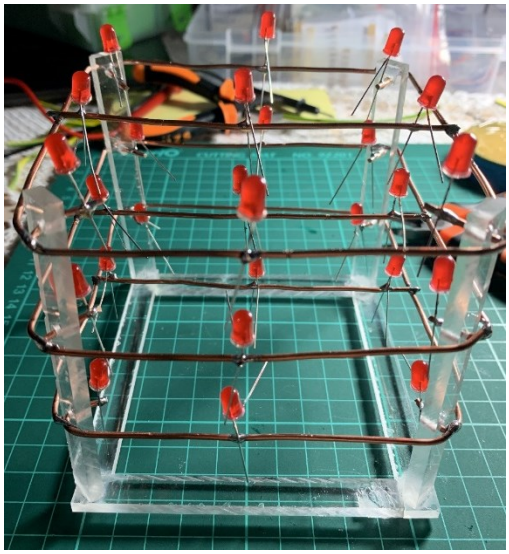


7. 9 adet led, anot uçlarından kaidenin birinci katına resimdeki gibi lehimlenir. Resimde kırmızı ledler kullanılmıştır. Setler içerisinde en fazla beyaz led bulunmaktadır (atmış adet). Bu yüzden resimde kullanılanın aksine kırmızı yerine beyaz led kullanılmalıdır.

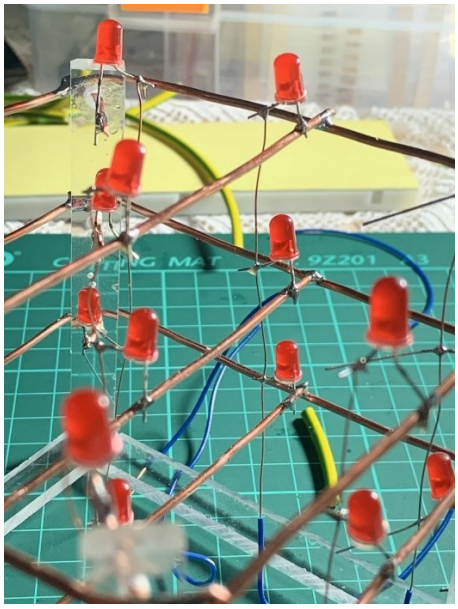


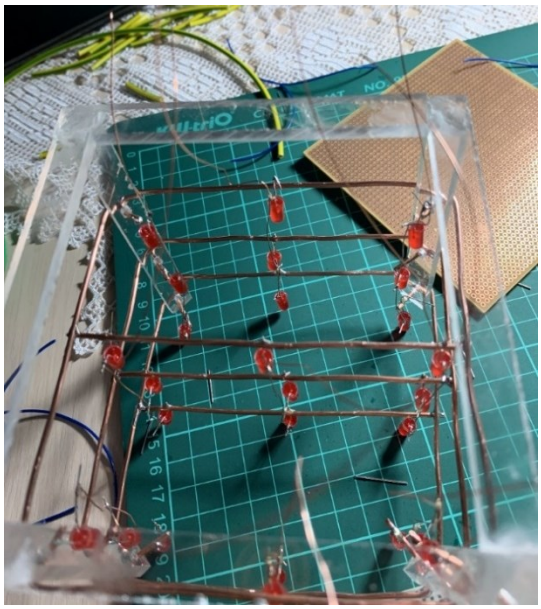
8. Dört, beş, altı ve yedinci adımlar ikinci ve üçüncü katlar için tekrarlanır. Tüm katlar tamamlandığında resimlerdeki gibi bir yapı elde edilmelidir.



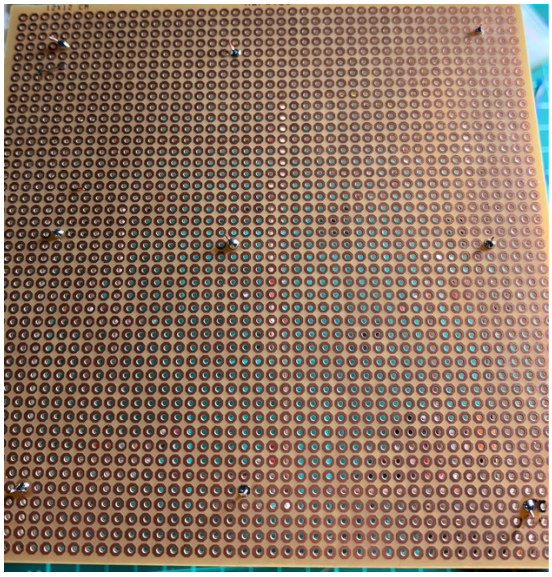


9. Aynı kolonda bulunan 3 ledin katot (-) bacakları bir tel yardımıyla resimlerde görüldüğü gibi birbirlerine bağlanır. Bu işlem bütün kolonlar için gerçekleştirilir. Kolonları oluşturmak için kullanılan teller kaidenin altından delikli pertinaksa bağlanacağı için bu teller delikli pertinaksa yetişecek uzunlukta olmalıdır.



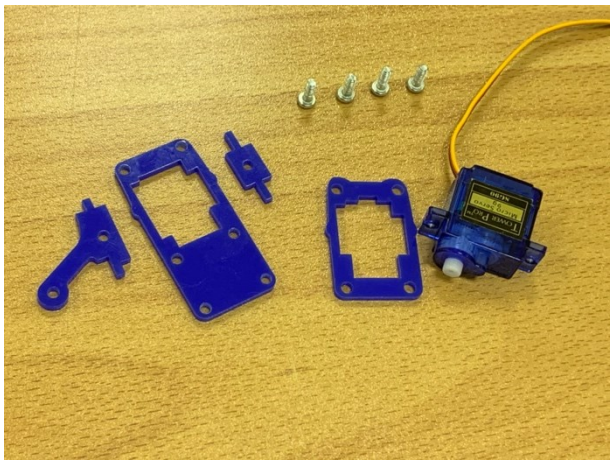


10. Aşağıdaki resimde yer alan delikli pertinaksta dokuz adet lehlimli nokta bulunmaktadır. Bu noktalar sütunları birleştiren teller ile bağlantılıdır. Her bir uca zil teli kablosu (erkek-erkek jumper kablo da olabilir) lehimlenerek (pertinaksın altından veya üstünden), Arduino bağlantısı için uç çıkarılır.

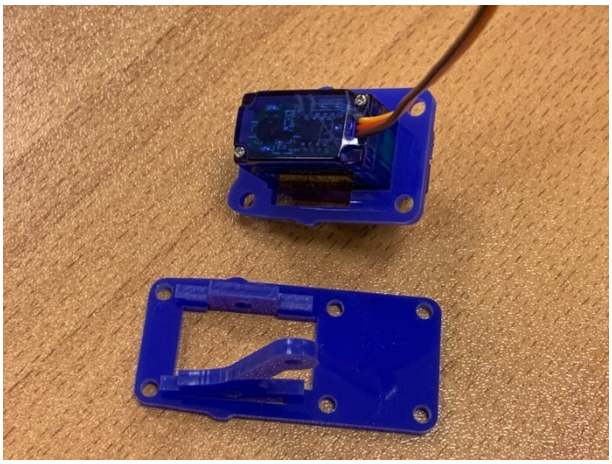


EK-2-Robot Kol Montaj Kılavuzu (Bölüm-5)

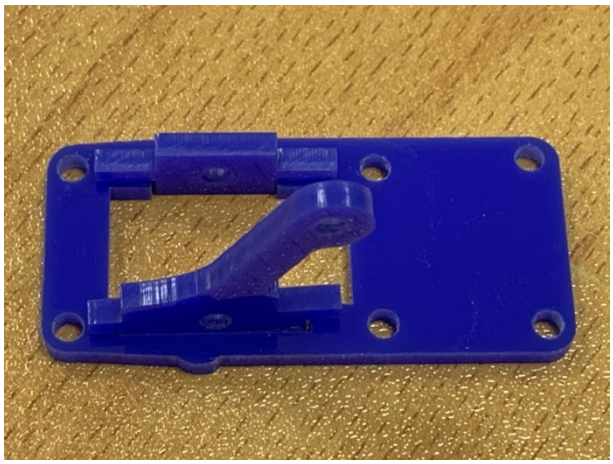
A. Öncelikle robot kolun kavrama kısmı yapılacaktır. Aşağıdaki adımlar sırasıyla takip edilerek montaj yapılabilir.



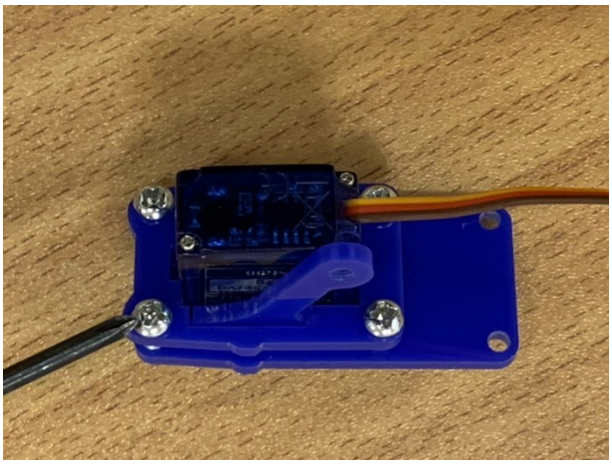
Bağlanacak elemanlar resimde gösterilmiştir.



Yan elemanları resimdeki gibi yerleştirmeye dikkat ediniz.

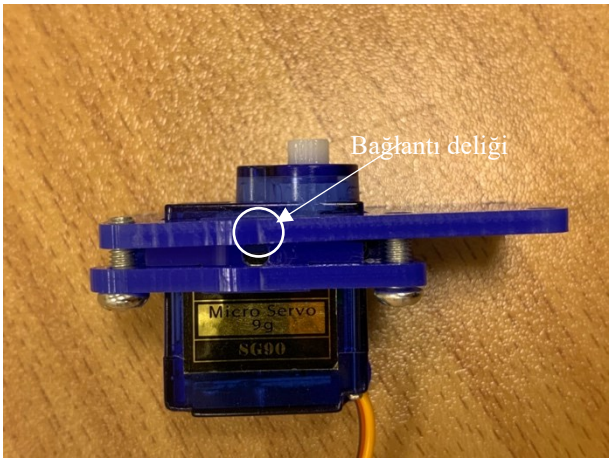


Yan elemanlar yakından çekim.



Cıvatalar sıkılır.

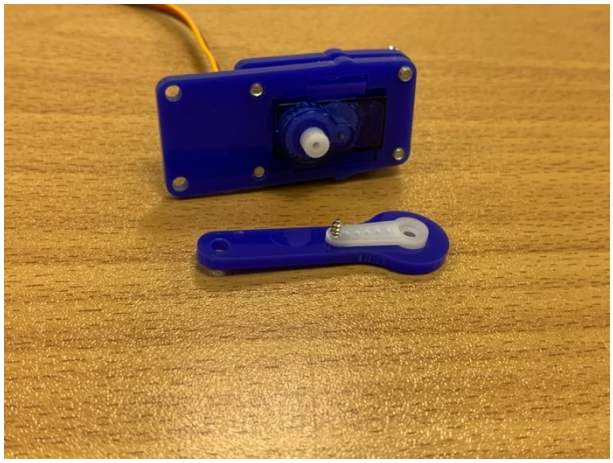
Bağlantı sonrası yandan çekim.



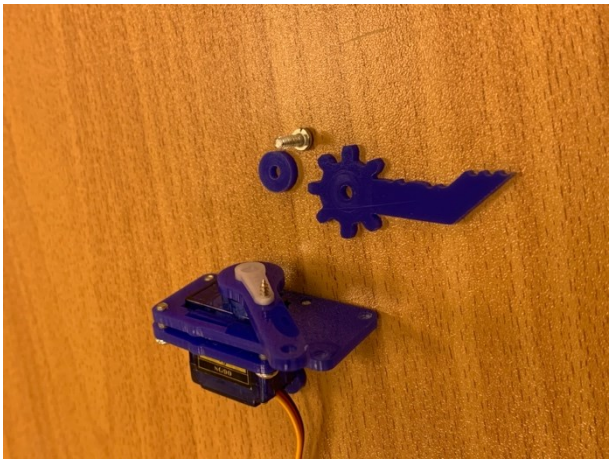
Bağlantı sonrası diğer yandan çekim.

Motor karşılığı olacak parçalar.

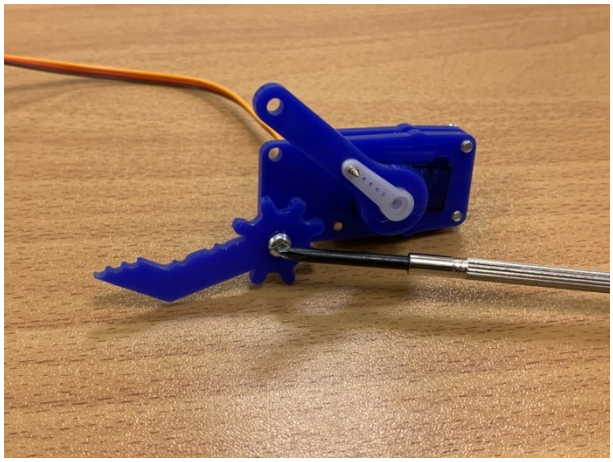
Vida sıkılır.



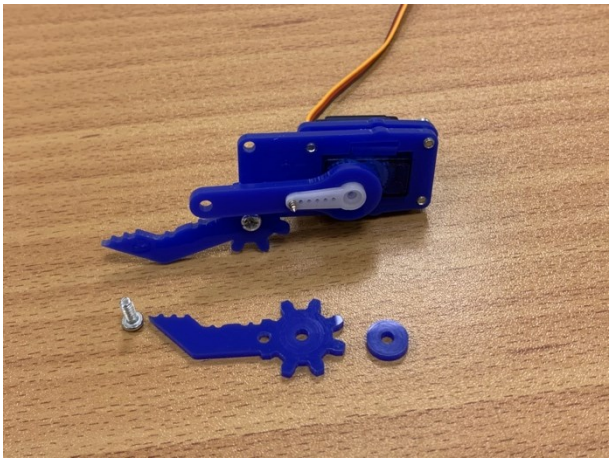
Motor çarkı karşılığına yerleştirilir.



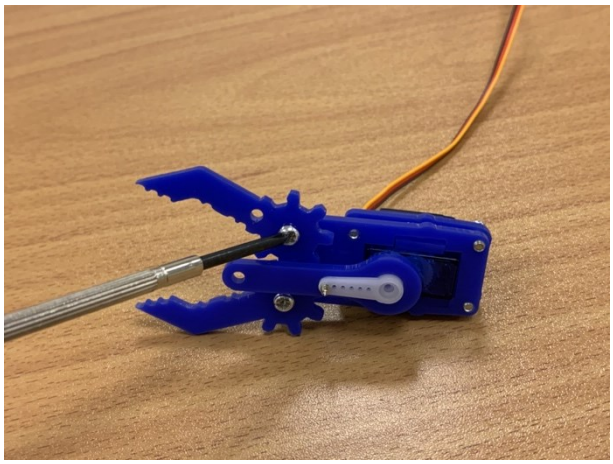
Monte edilecek yeni parçalar.



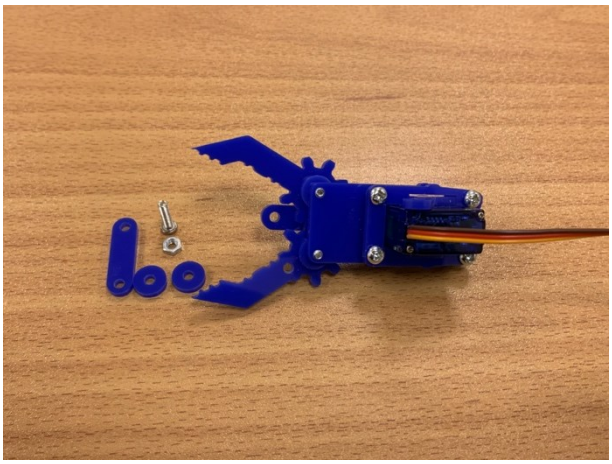
Pulu çarkın altına koymayı unutmayınız.



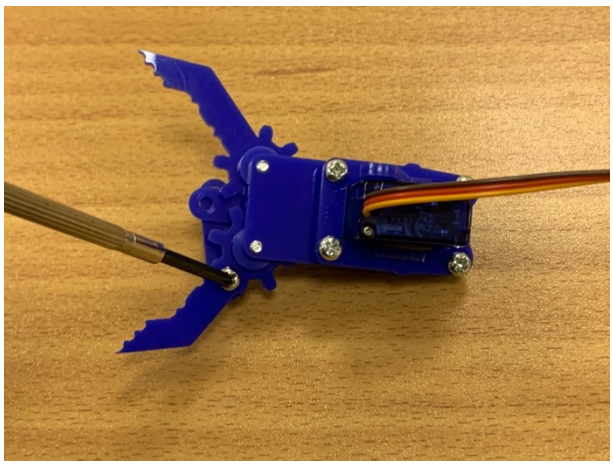
Monte edilecek yeni parçalar.



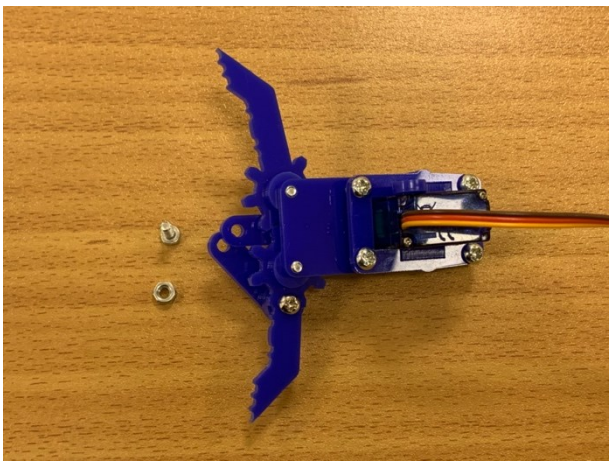
Pulu çarkın altına koymayı unutmayınız.



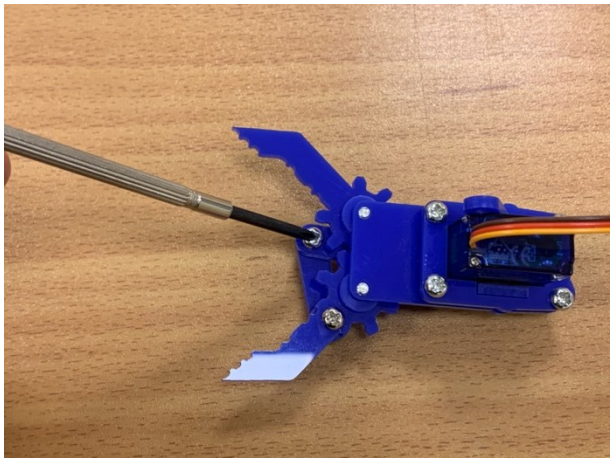
Monte edilecek yeni parçalar.



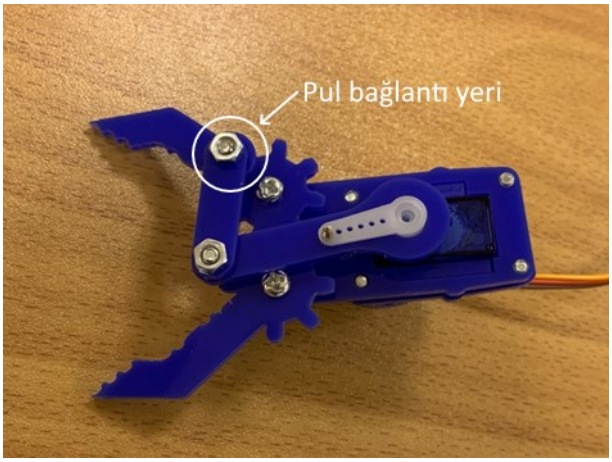
İki adet pulu bağlantı için yerleştirmeyi unutmayınız.



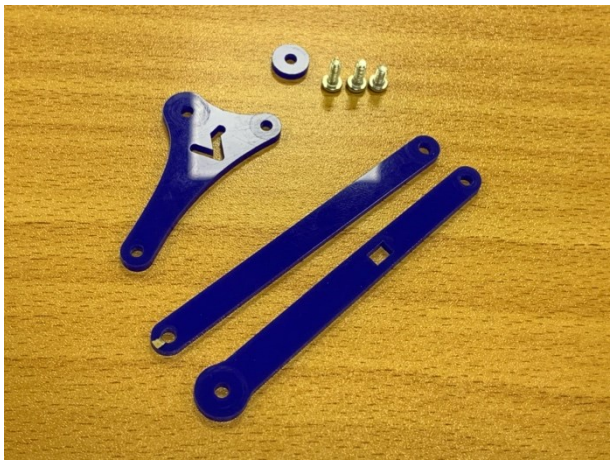
Son cıvata da bağlanır.



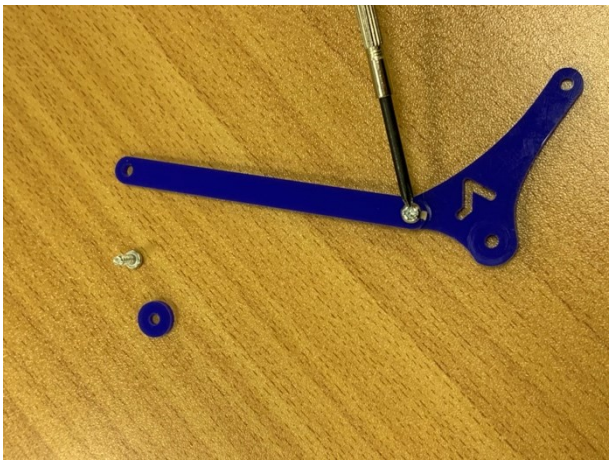
Bağlantı sonrası üstten görünüm.



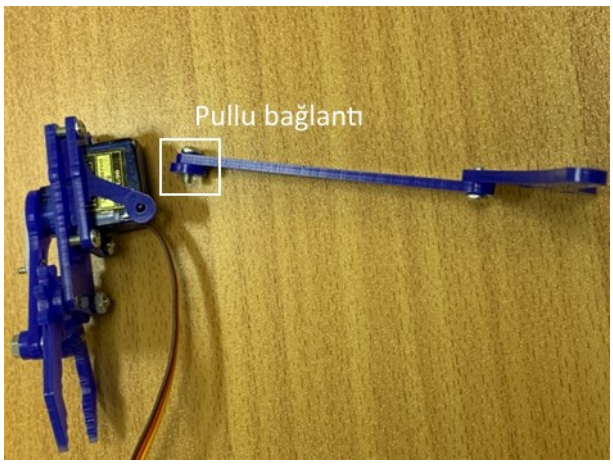
Bağlantı sonrası alttan görünüm.



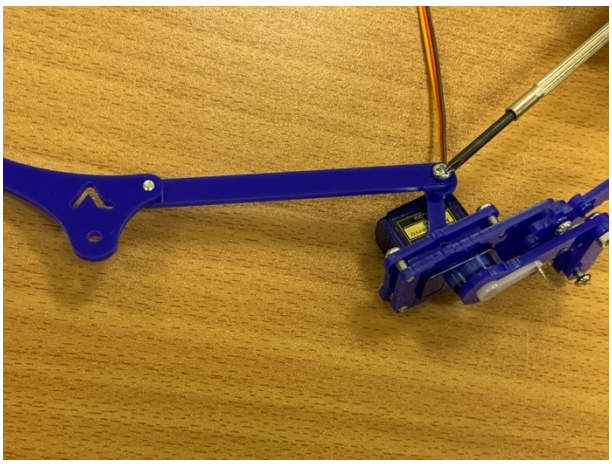
Monte edilecek yeni parçalar.



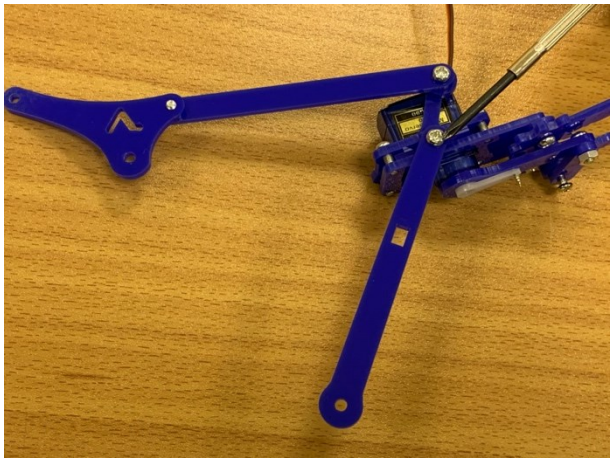
Cıvata sıkılır.



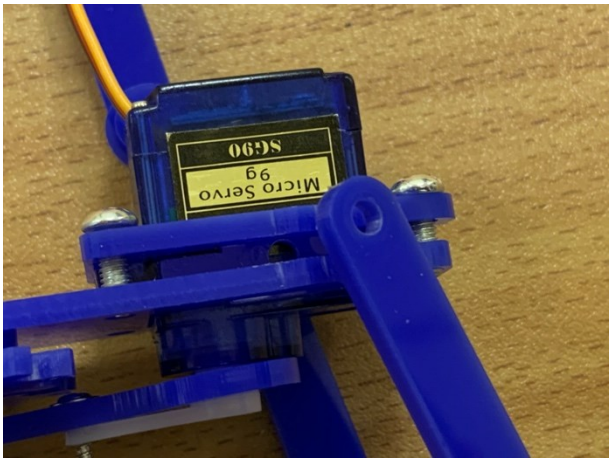
Pul kullanmayı unutmayınız.



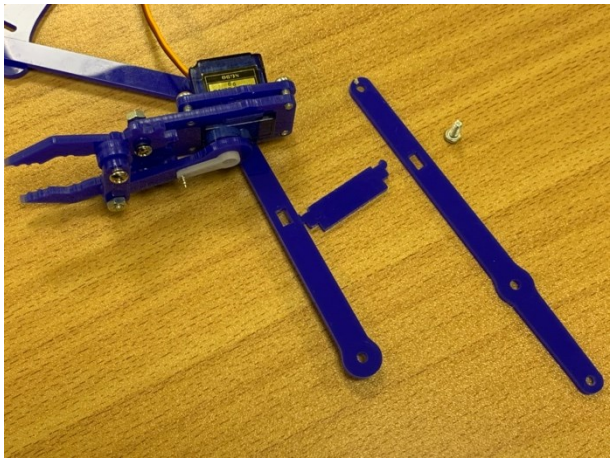
İki parça birbirine şekildeki gibi monte edilir.



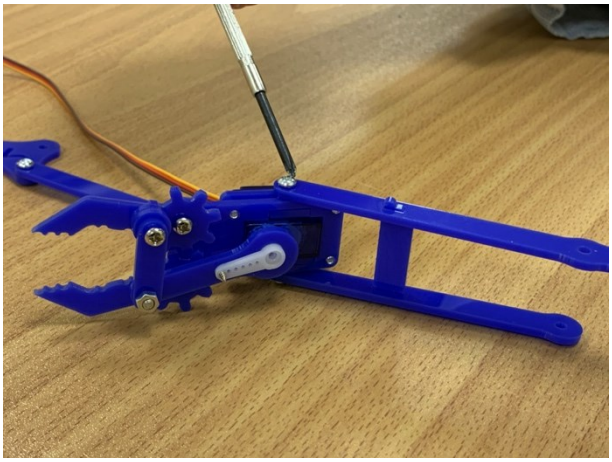
Yeni parça monte edilir. Monte edilen kısım sağdaki resimde büyütülmüştür.



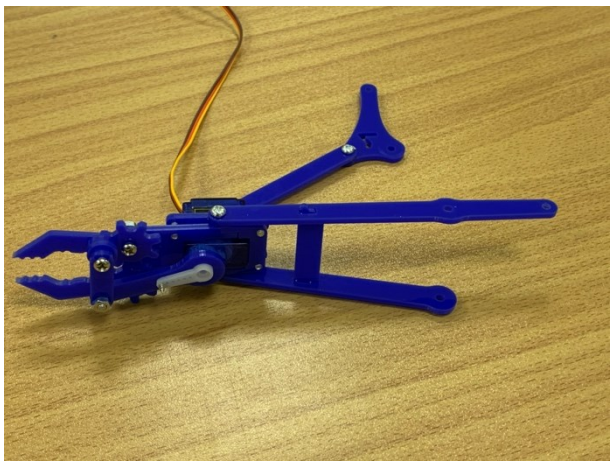
Monte edilen yeni parçanın montaj öncesi bağlantı deliği görüntüsü.



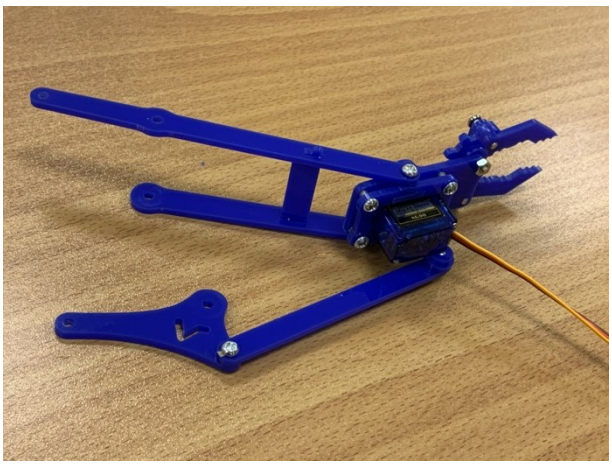
Monte edilecek yeni parçalar.



Orta bağlantı çubuğu takıldıktan sonra cıvata sıkılır.

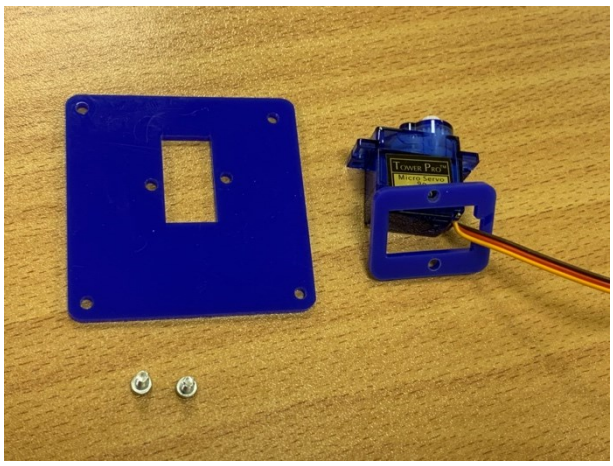


Bağlantı sonrası yandan görünüm.

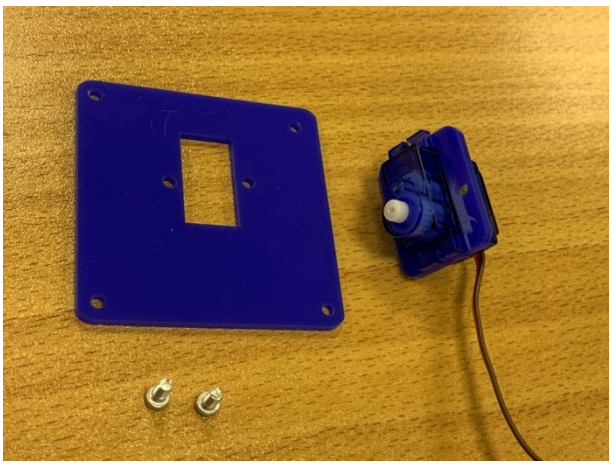


Bağlantı sonrası diğer yandan görünüm.

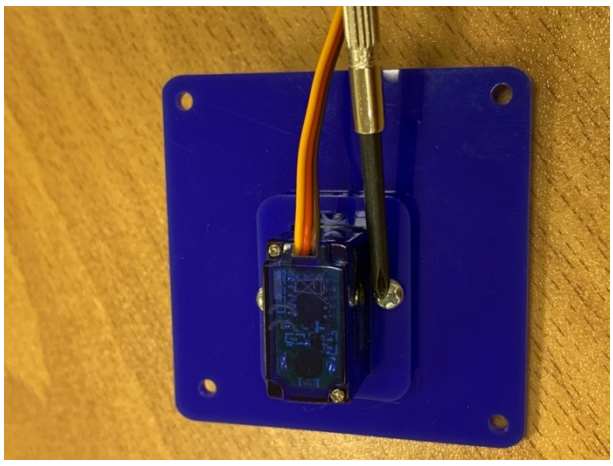
B. Robot kolun kavrama kısmı hazırlanmıştır. Sırada robot kolun hareketli kaidesini oluşturmak bulunmaktadır.



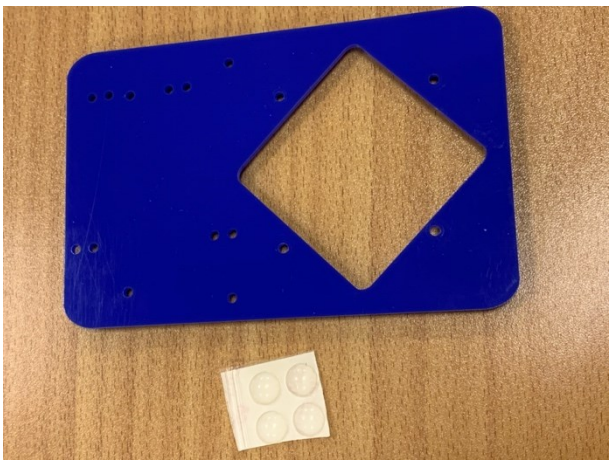
Monte edilecek parçalar.



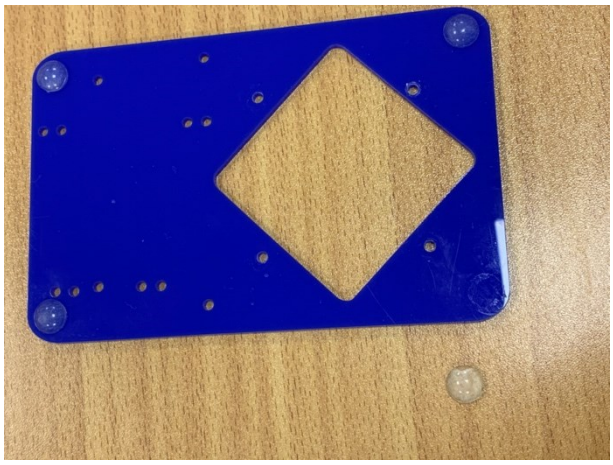
Dörtgensi bölge şekildeki gibi takılır.



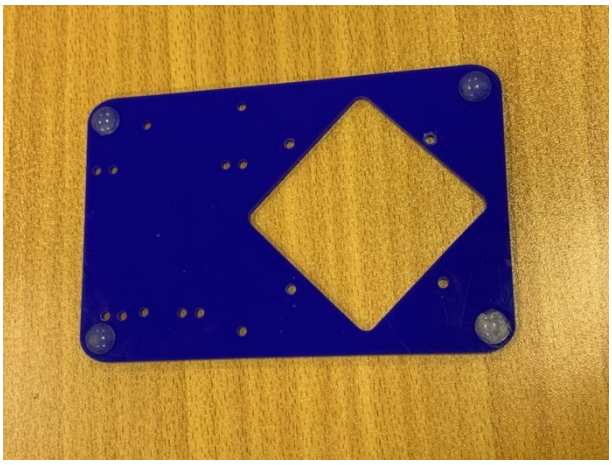
Cıvatalar sıkılır.



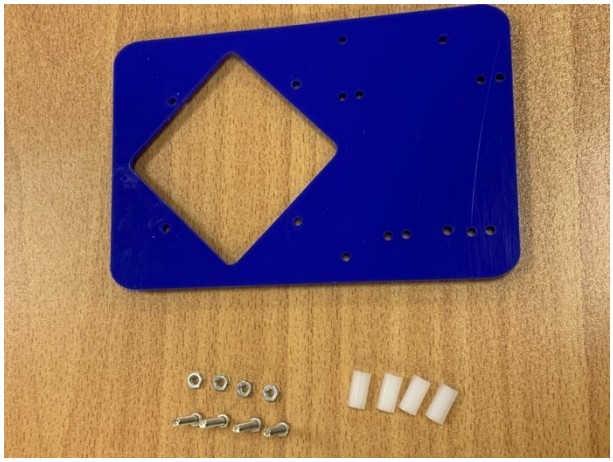
Monte edilecek yeni parçalar



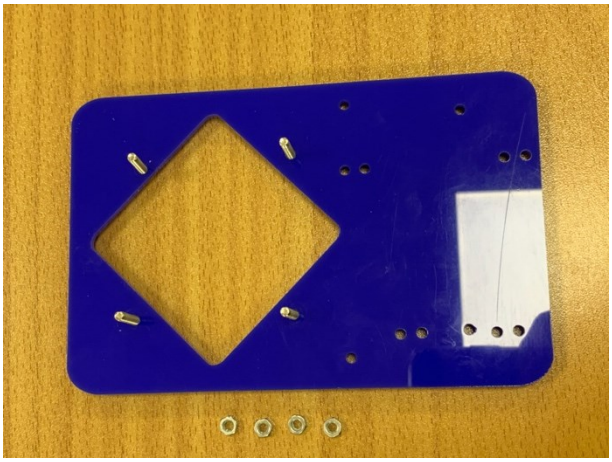
Alt ayaklar monte edilir.



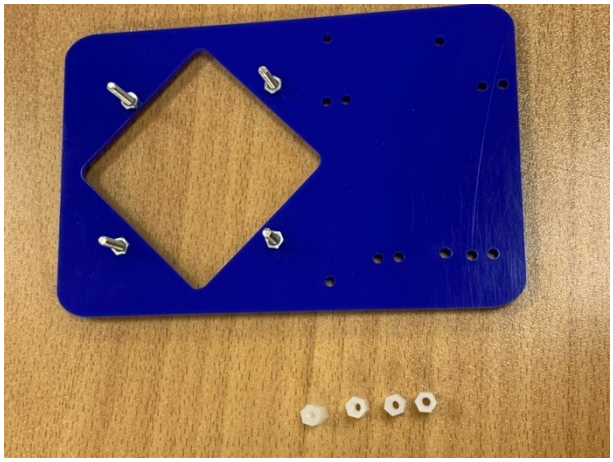
Montaj sonrası görünüm.



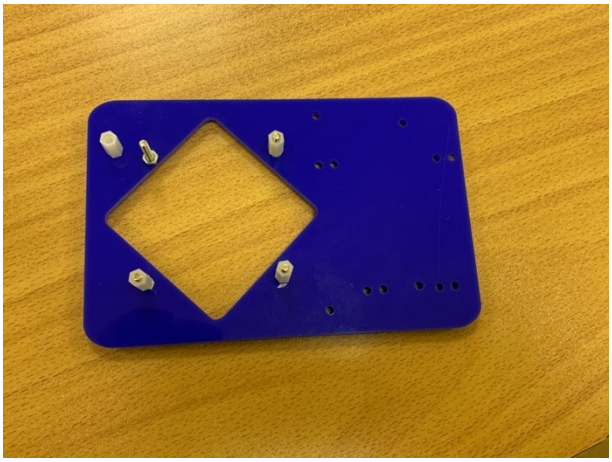
Monte edilecek yeni parçalar (not: ayaklar altta bulunmaktadır).



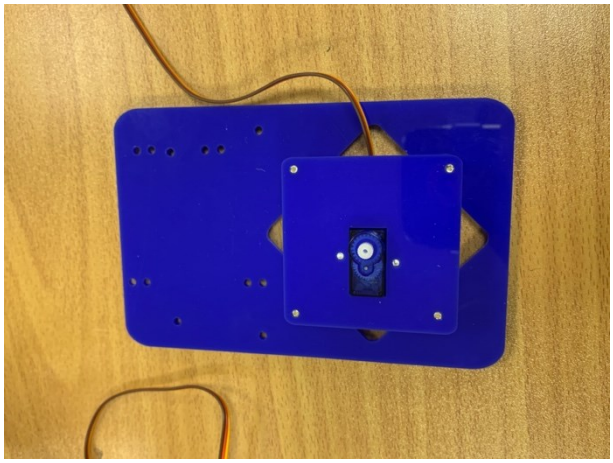
Cıvatalar yerine takılır.



Somunlar sıkılır.

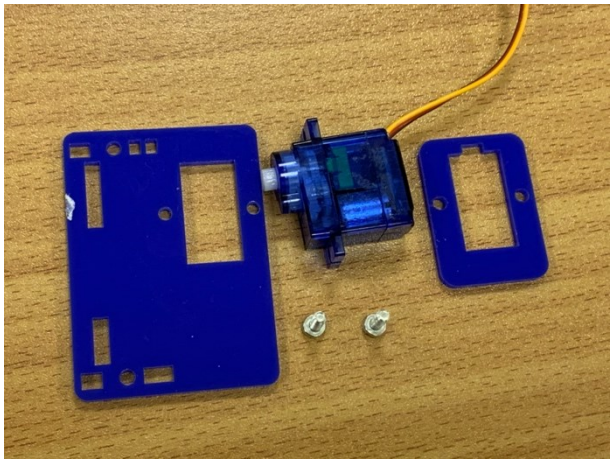


Aralayıcılar cıvatalara monte edilir.

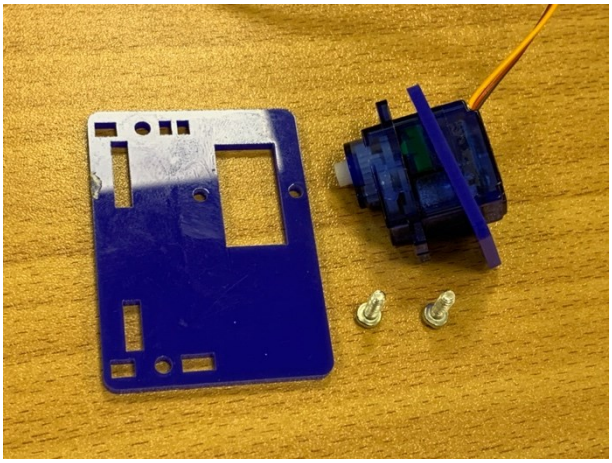


Montaj sonrası kaide, üstten görünüm.

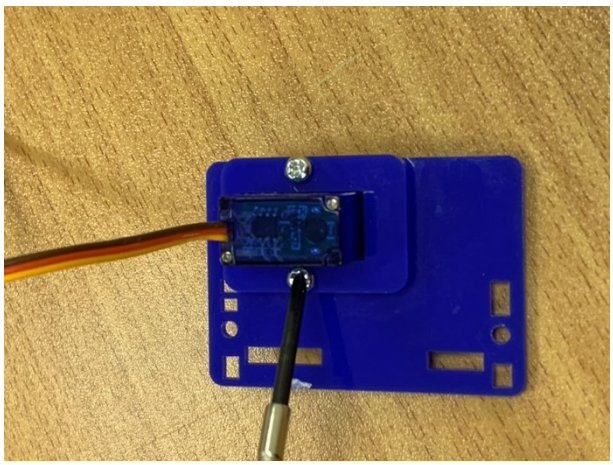
C. Robotun kaidesi hazırlanmıştır. Sırada robotun gövdesini oluşturmak bulunmaktadır.



Monte edilecek parçalar.



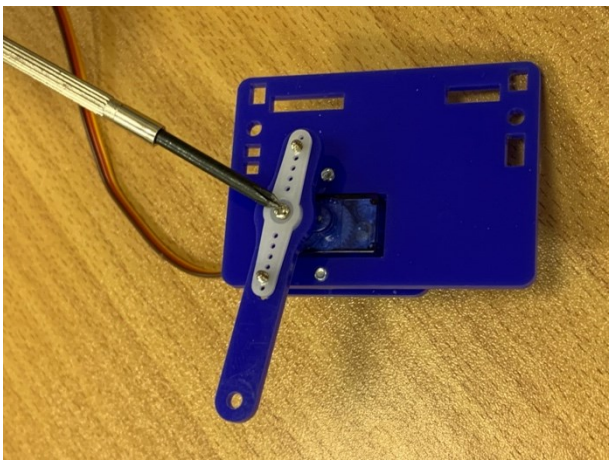
Dörtgensi bölge yerleştirilir.



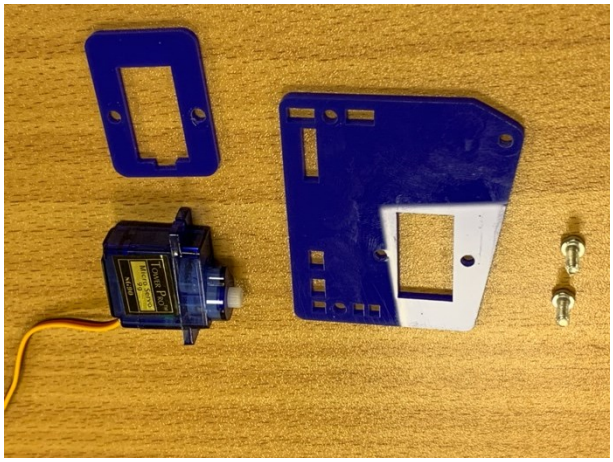
Cıvatalar sıkılır.

Motor karşılığı için kullanılacak yeni elemanlar.

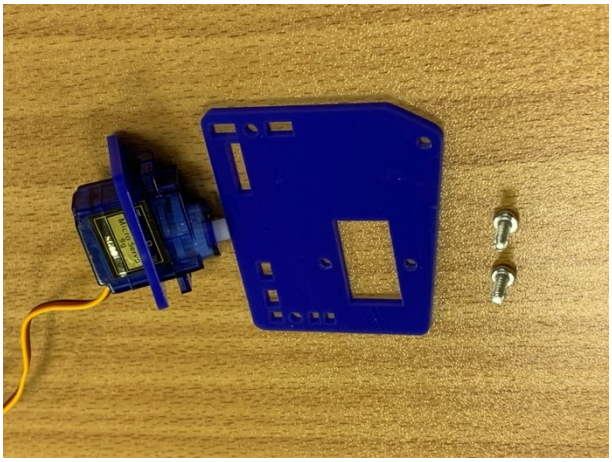
Vidalar sıkılarak motor karşılığı oluşturulur.



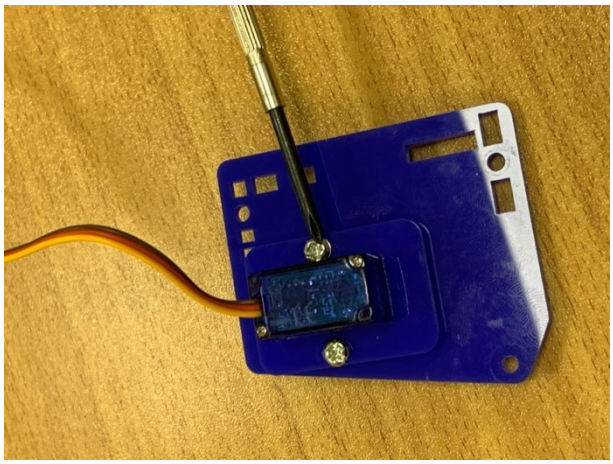
Motor karşılığı motora yerleştirilir ve cıvata sıkılır.



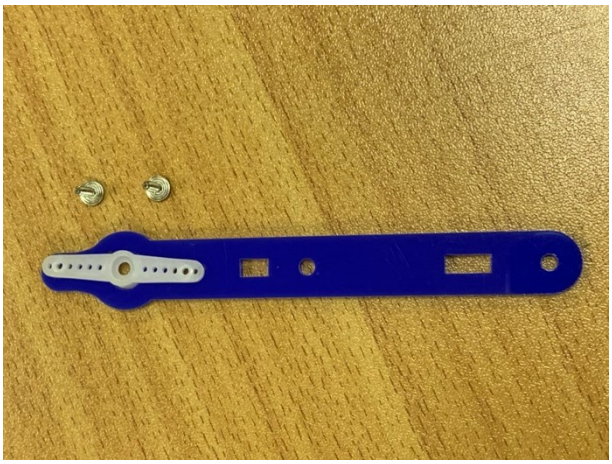
Monte edilecek yeni parçalar.



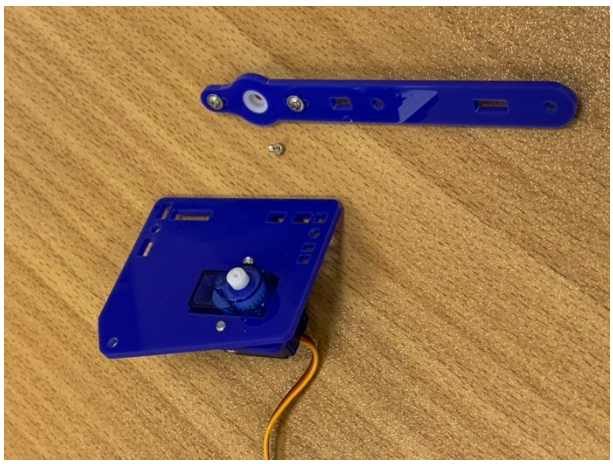
Dörtgensi bölge yerleştirilir.



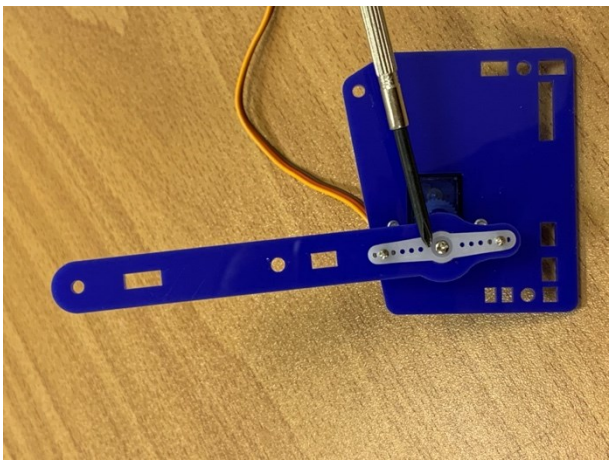
Cıvatalar sıkılır.



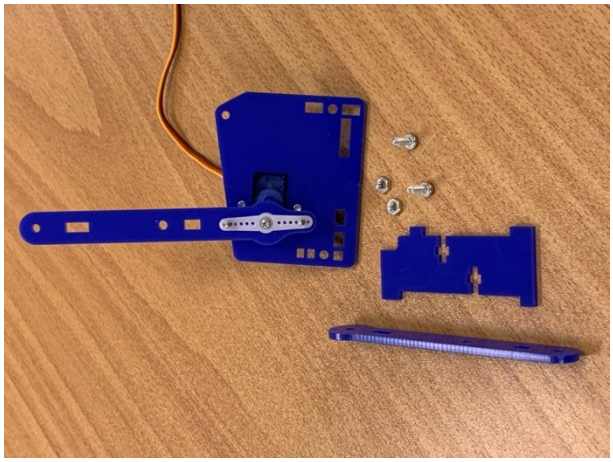
Motor karşılığı için yeni parçalar.



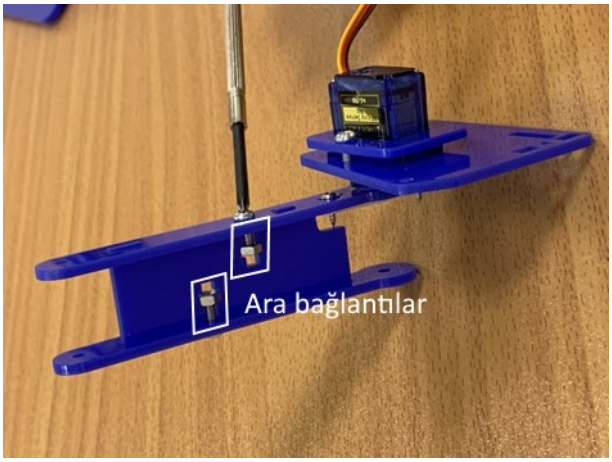
Motor karşılığı vidalar sıkılarak oluşturulur.



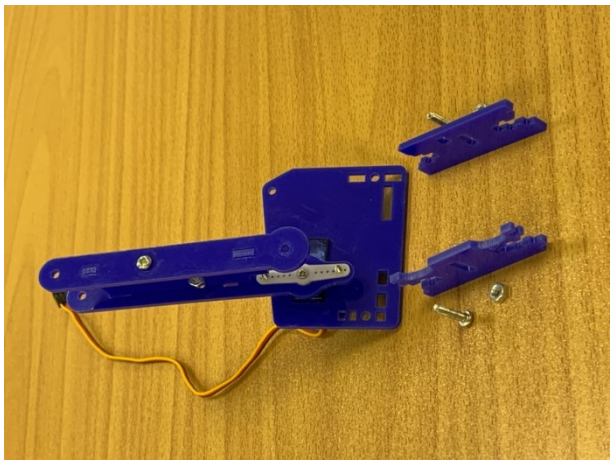
Motor karşılığı motora monte edilir ve cıvatası sıkılır.



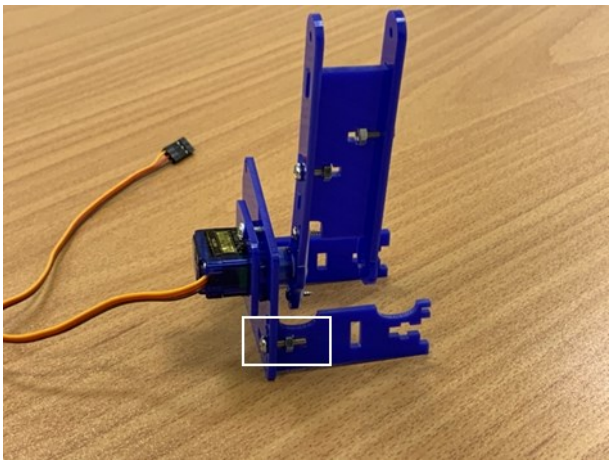
Monte edilecek yeni parçalar.



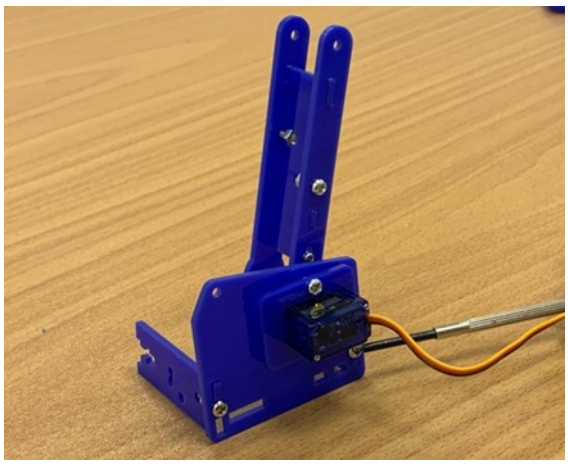
İki çubuğu birleştiren ara bağlantılar yapılır.



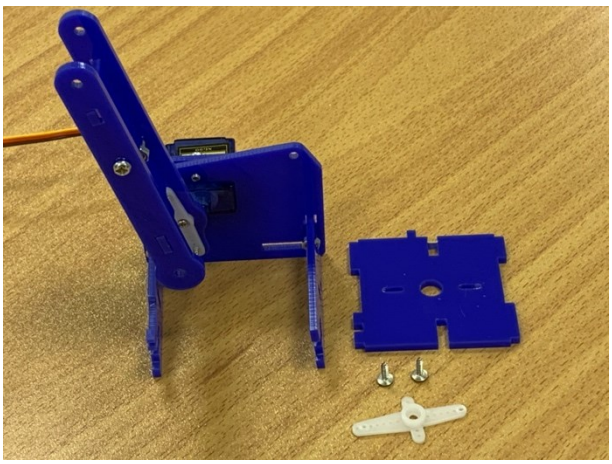
Monte edilecek yeni parçalar.



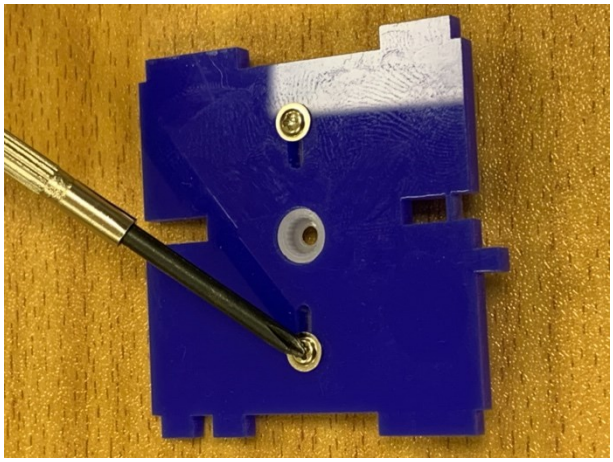
Cıvata ve somun görseldeki gibi bağlanır.



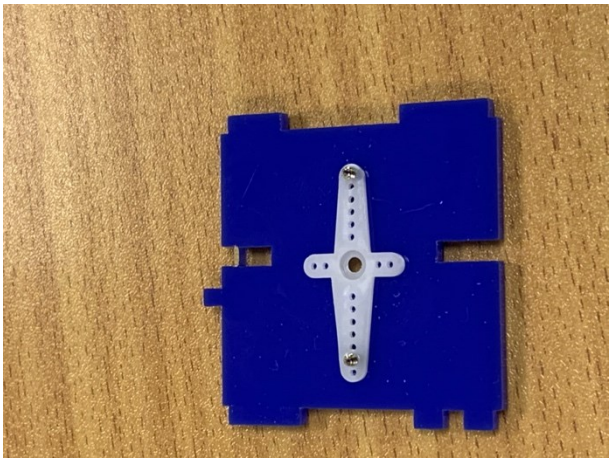
Diğer parça da bağlanır.



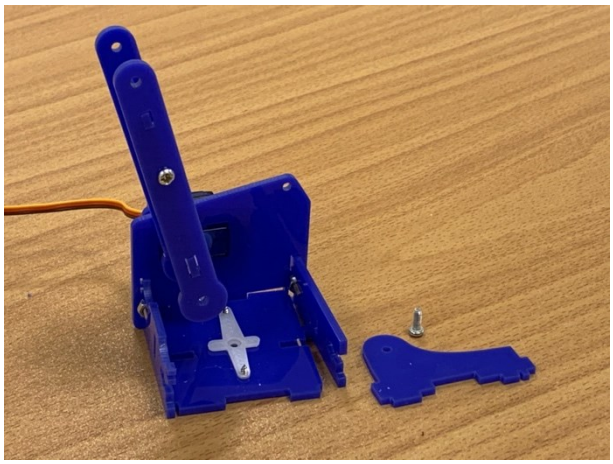
Monte edilecek yeni parçalar.



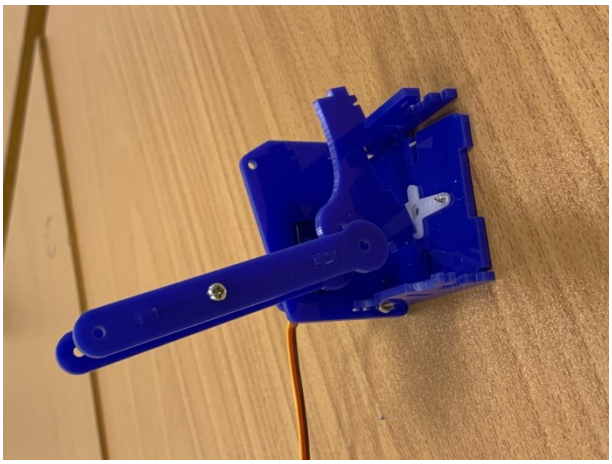
Motor karşılığı takılır ve vidalar sıkılır.



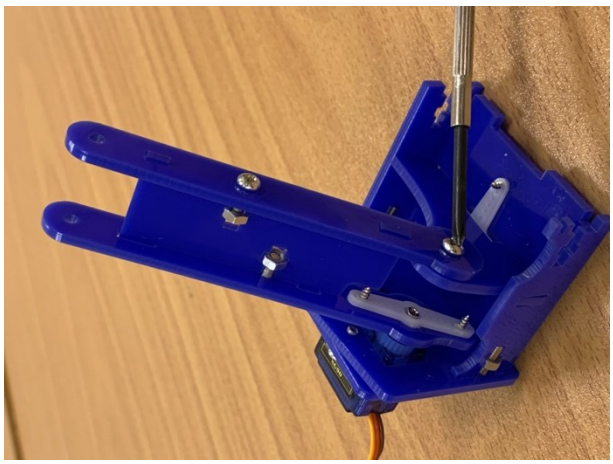
Bağlantı sonrası görünüm.



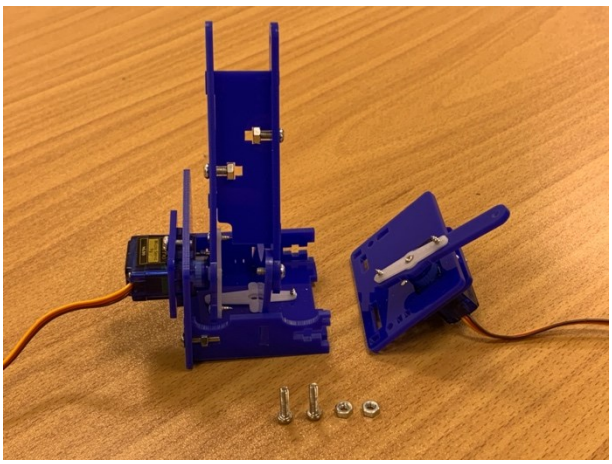
Monte edilecek parçalar.



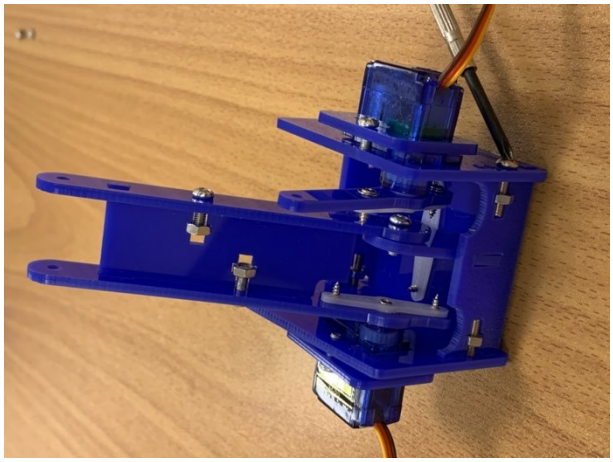
Parçalar yerine yerleştirilir.



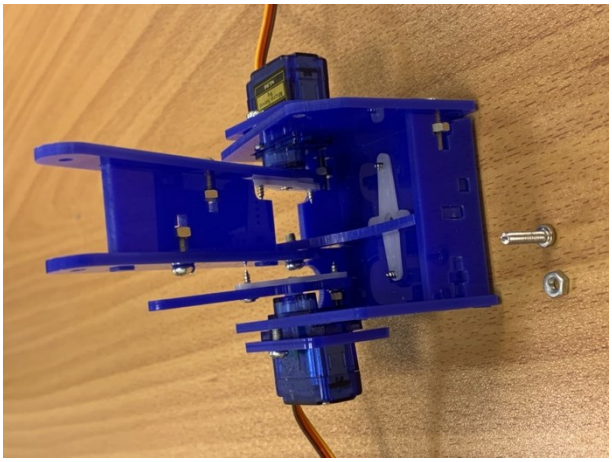
Cıvata sıkılır.



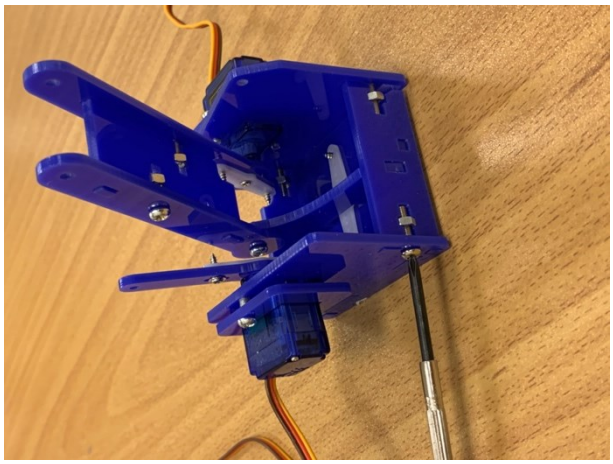
Önceden oluşturulmuş parça ve yeni cıvata ve somunlar.



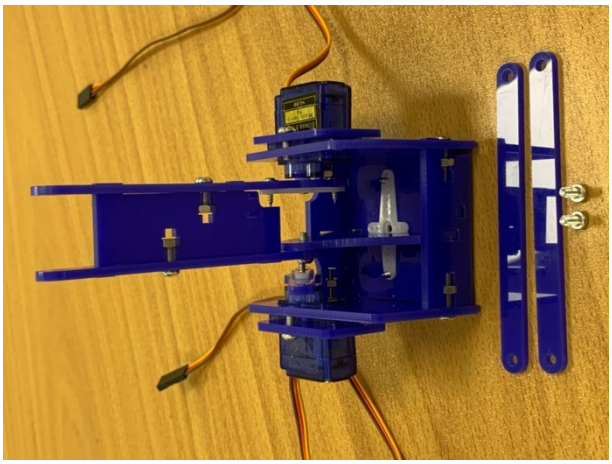
Görseldeki gibi birinci bağlantı yapılır.



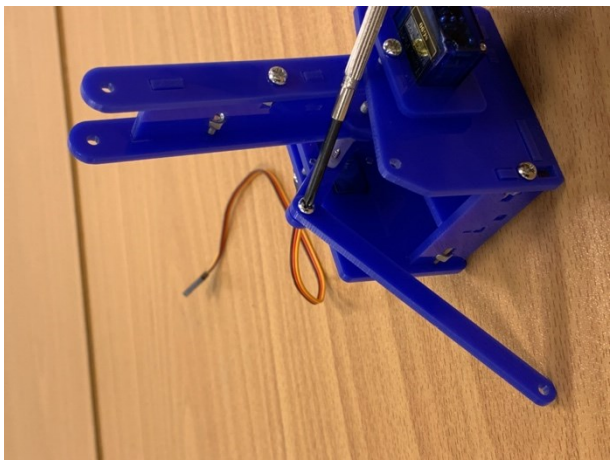
Birinci bağlantı sonrası görünüm.



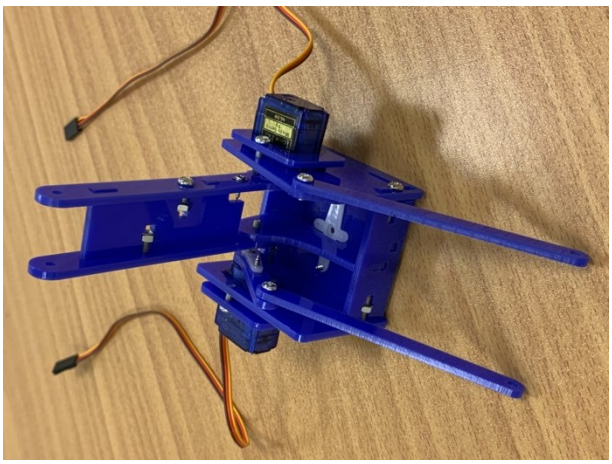
Resimdeki gibi ikinci bağlantı yapılır.



Monte edilecek yeni parçalar.

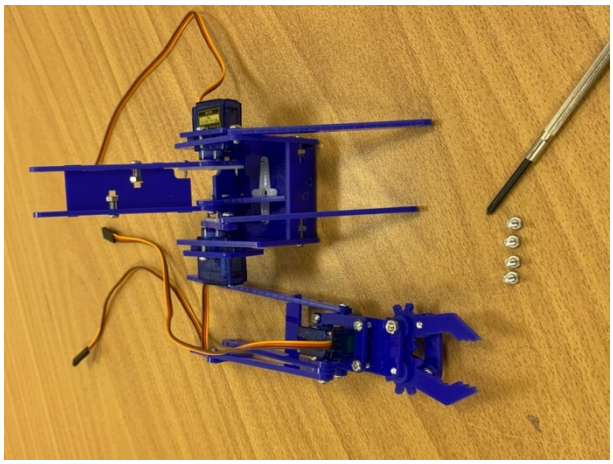


Birinci cıvata sıkılır.

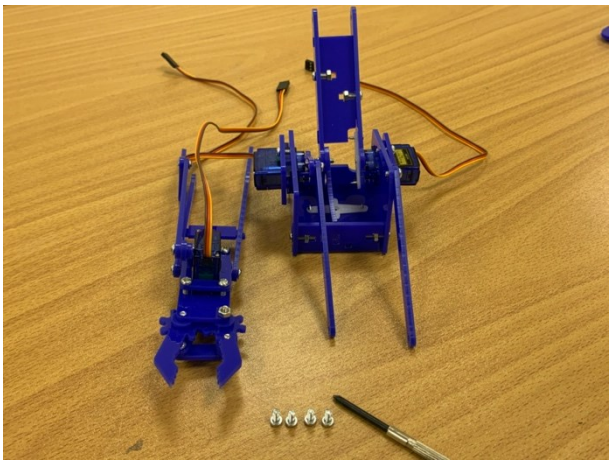


İkinci cıvata sıkılır.

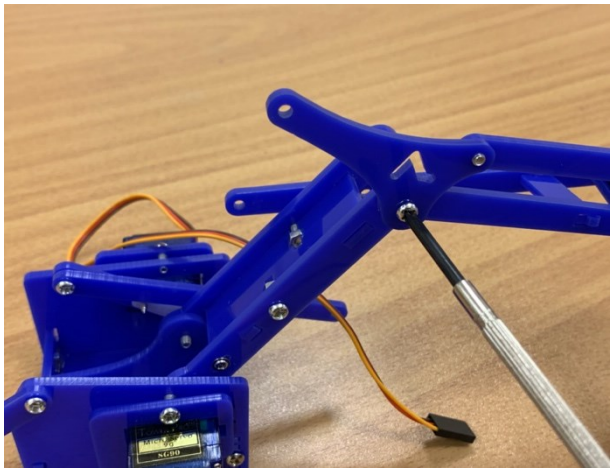
D. Robotun kavrama kolu, gövdesi ve kaidesi hazırlanmıştır. Aşağıdaki resimlerde gösterildiği gibi kavrama kolu ve gövde birleştirilerek robot kolun üst kısmı oluşturulur.



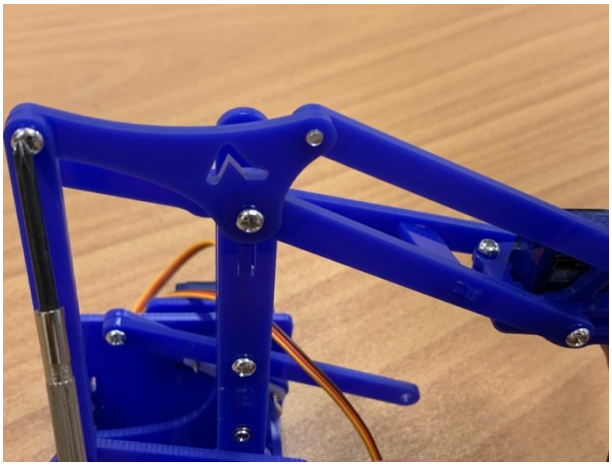
Bağlantısı yapılacak kavrama kolu, gövde ve yeni cıvatalar.



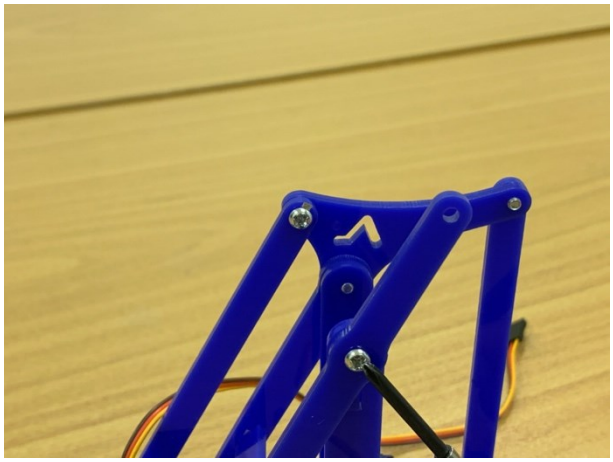
Bağlantı öncesi önden görünüm.



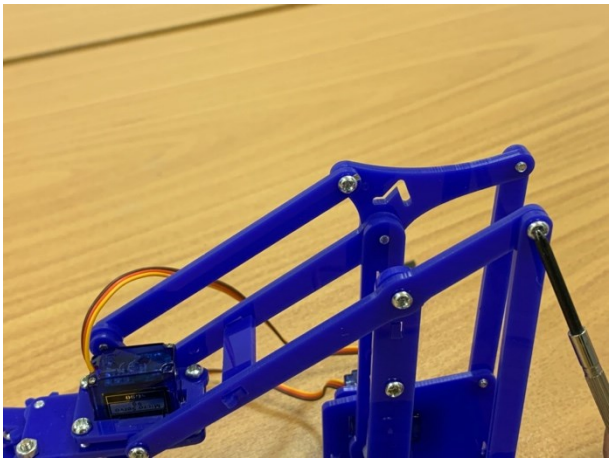
Birinci cıvata sıkılır.



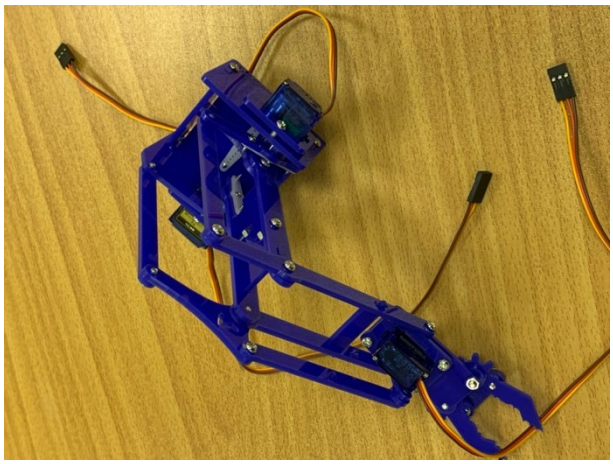
İkinci cıvata sıkılır.



Üçüncü cıvata sıkılır.

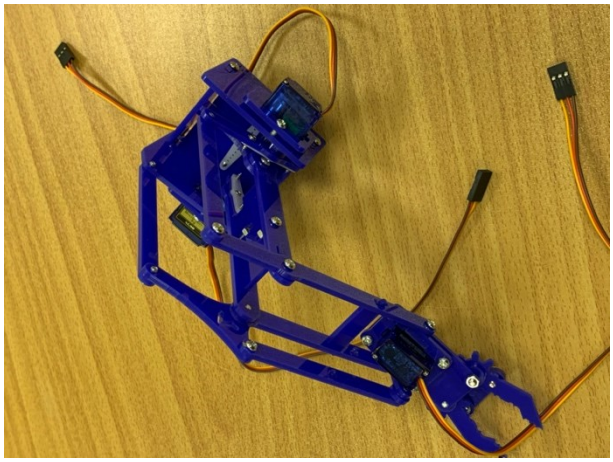


Dördüncü cıvata sıkılır.

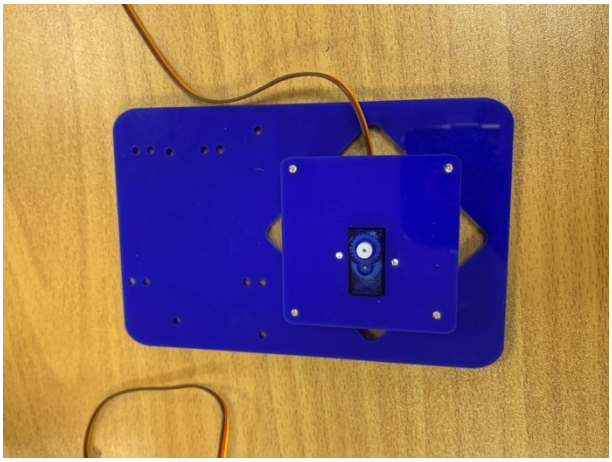


Montaj sonrası üstten görünüm.

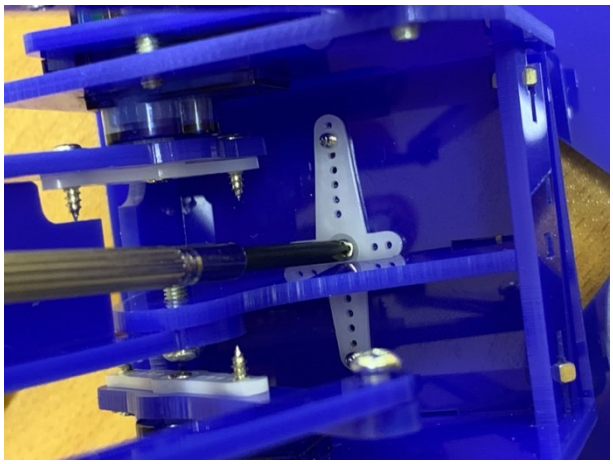
E. Bu aşamaya kadar robot kolun kavrama kısmı ve gövdesi birleştirilip robot kolun üst kısmı hazırlanmıştır. Bunun yanında robot kolun kaidesi de hazırlanmıştır. Burada robot kolun oluşturulan üst kısmı ve kaidesi birleştirilecektir. Bunun için kaidedeki servo motor üst kısımda bulunan servo motor parçasına yerleştirilir ve vidası takılarak sıkılır.



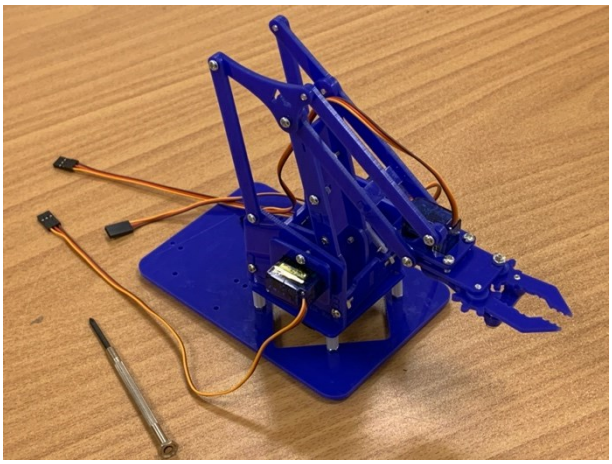
Önceden oluşturulmuş üst kısım.



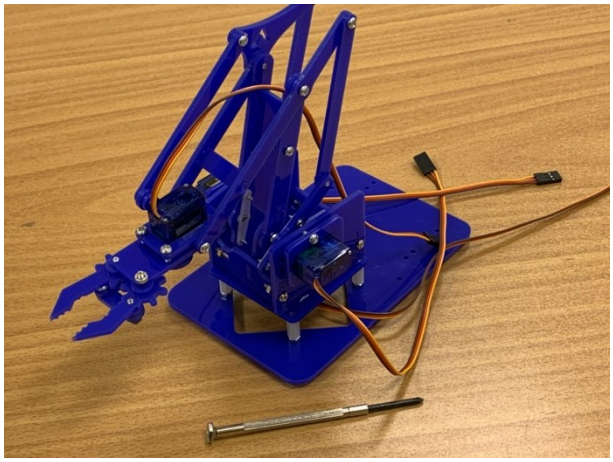
Önceden oluşturulmuş kaide.



Üst kısımda bulunan motor karşılığı kaidede bulunan motora takılır ve cıvata sıkılır.



Montaj sonrası yandan görünüm.



Montaj sonrası diğer yandan görünüm.

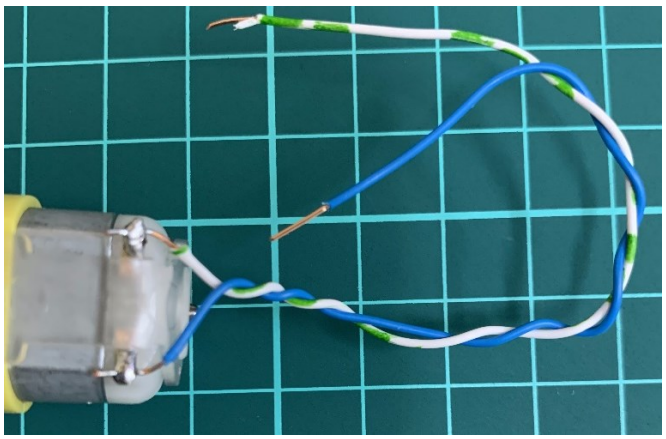
F. Son Ayarlar: Artık robot kol hazırdır. Arduino kart bağlantıları yapılarak robot kol kullanmaya başlanabilir. Fakat bu aşamada motorların durma açıları doğru olmayabilir. Bu durumda servo motorların başlangıç açılarını ayarlamak gerekmektedir. Bağlantılar yapılıp kod yüklendikten sonra robot kol istenilen yöne doğru yöneliyor fakat beklendiği gibi ilerliyemiyorsa muhtemelen başlangıç açıları yanlıştır. Örneğin başlangıç değeri 170 derecede olan bir servo motor 50 derece daha döndürülmek istenirse 10 derece döndükten sonra duracaktır. Bu durumda servo motorun başlangıç açısı, duruma göre, 0 veya 90 olarak ayarlanır.

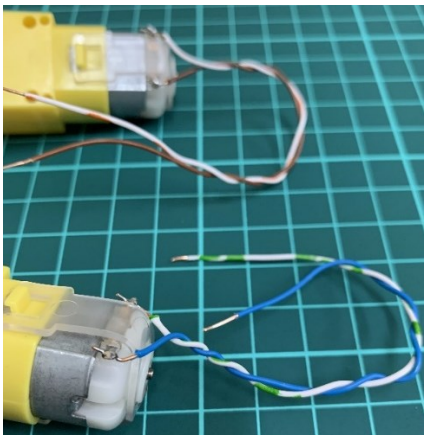
Önemli Uyarı

Montaj esnasında servo motoru çevirmek gerekebilir. Öğrenciler servo motoru çevirirken dikkatli olmalıdırlar. Sette kullanılan servo motorların çarkları plastikten yapılmıştır. Servo motor dönmüyorsa, onu zorlayarak döndürmek isteyen öğrenciler az bir zorlama ile çarkları kırılabilir. Bu yüzden öğrenciler etkinlik öncesinde ve etkinlik boyunca servo motoru döndürürken zorlamamaları konusunda uyarılmalıdır. Rehber öğretmen uyarıları yaparken, tüm sınıfın kendisini dinlediğini ve uyarıları anladığına kanaat getirmezse uyarıları yeniden yapmalıdır.

EK-3-Araba Kurulum Kılavuzu Resimleri (Bölüm-6)

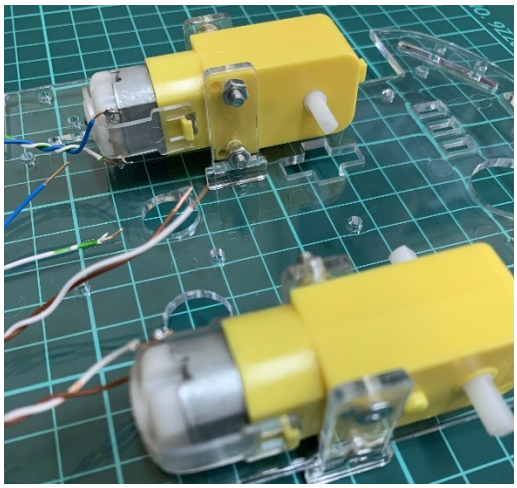
1) Öncelikle DC motorların elektrik kabloları lehimli değilse kablolar DC motorlara lehimlenir.

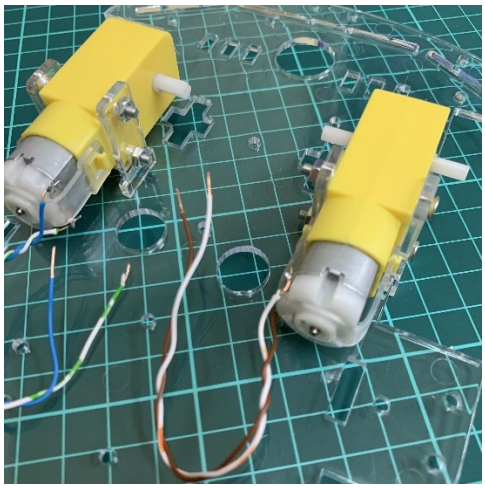




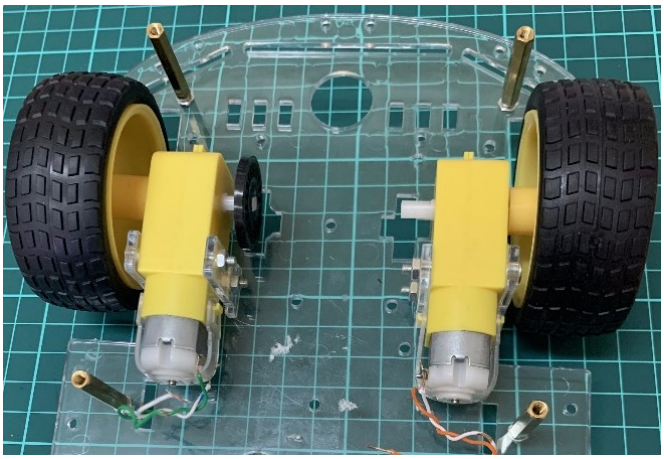
2) Alt pleksi gövdeye DC motorları sabitlemek için kullanılacak ayaklar monte edilir.

3) Motor sabitleme ayakları, vidalar ve somunlarla DC motorlar şasiye sabitlenir.



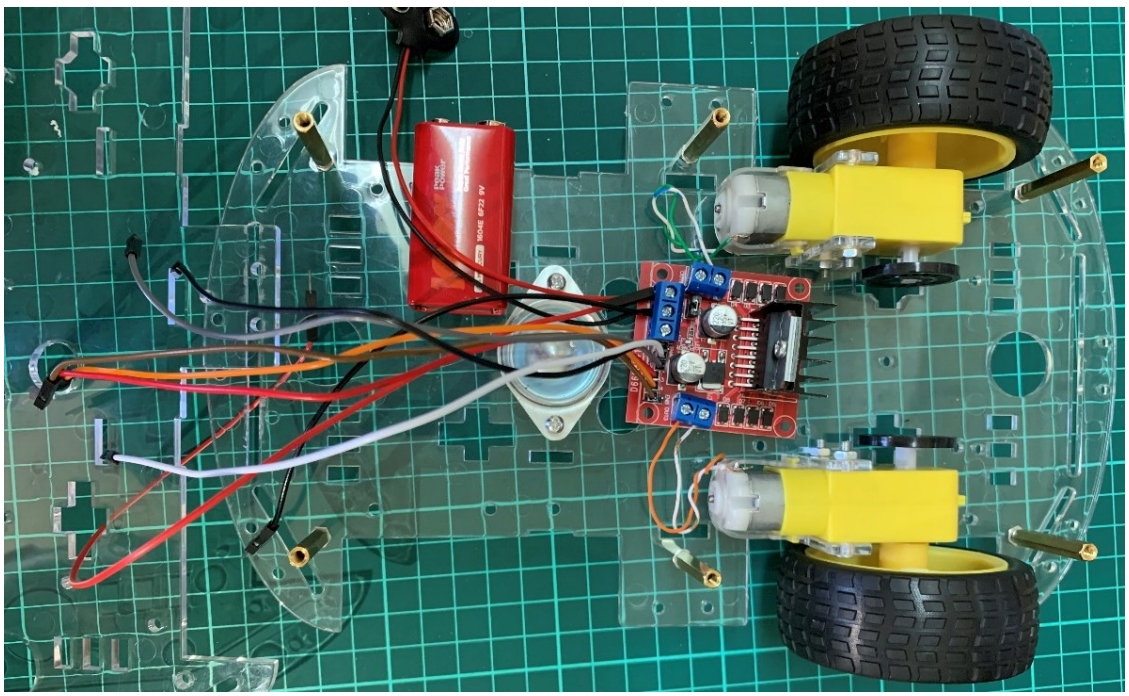


4) Sonraki adımda DC motorlara tekerler ve enkoder monte edilir. Enkoder tek yönlü olarak motora monte edilecek şekilde tasarlanmıştır. Enkoder girmiyorsa zorlamayıp diğer yön denenmelidir.

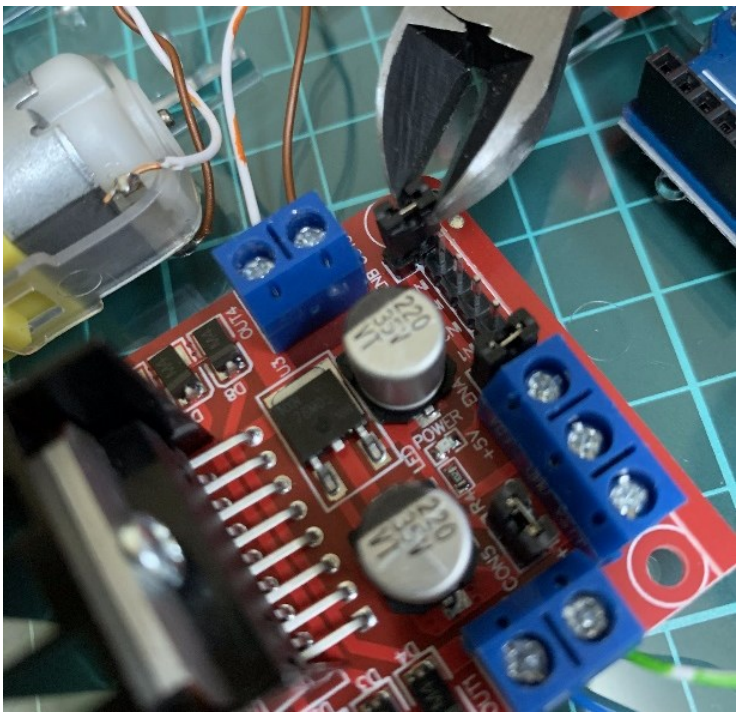


5) Delikli metal sarhoş teker pleksi şasiye görseldeki gibi yeni delikler açılarak monte edilir.

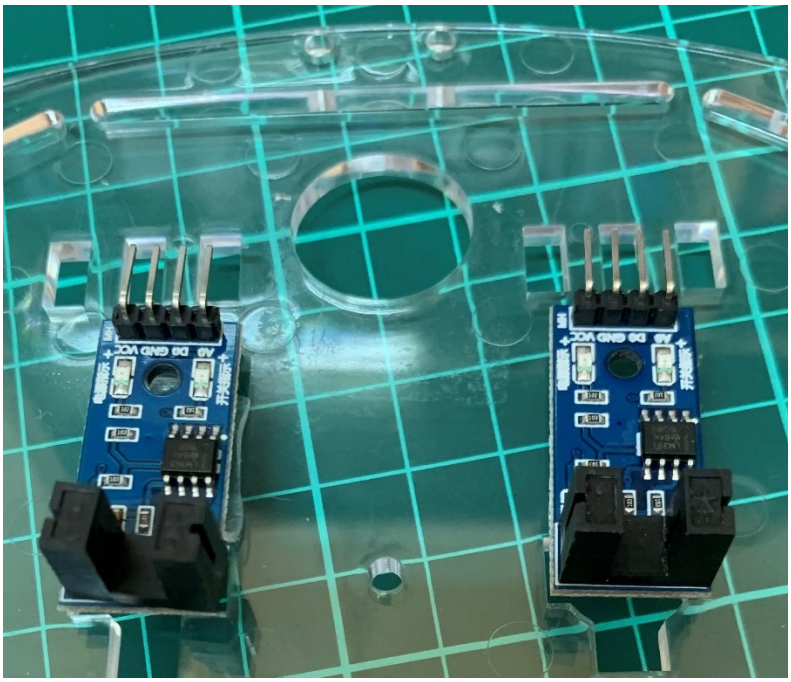
6) L298N sürücü modülü ve pil iki taraflı bant ile pleksi şasiye yapıştırılır. DC motor bağlantıları, pil bağlantı uçları ve pin kabloları L298N modülüne takılır.



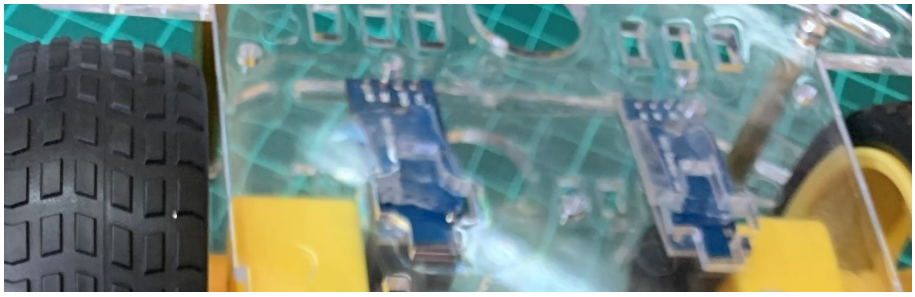
Dikkat: L298N ilk satın alındığında 1 ve 6 numaralı bacaklarda jumperlar bulunmaktadır. Bu durum aşağıdaki görselde gösterilmiştir. Bu jumperlar çıkarılmalıdır.



7) Kızılötesi hız sensörü resimdeki gibi üst pleksiye monte edilir. Hız sensörünün enkoderin bölmelerine denk gelecek şekilde ayarlanması önemlidir. Hız sensörü altıncı bölümde kullanılmayacaktır. Hız sensörü onikinci bölümde kullanılacaktır. Fakat araba oluşturulurken monte edilmesi ileride kolaylık sağlayacaktır.

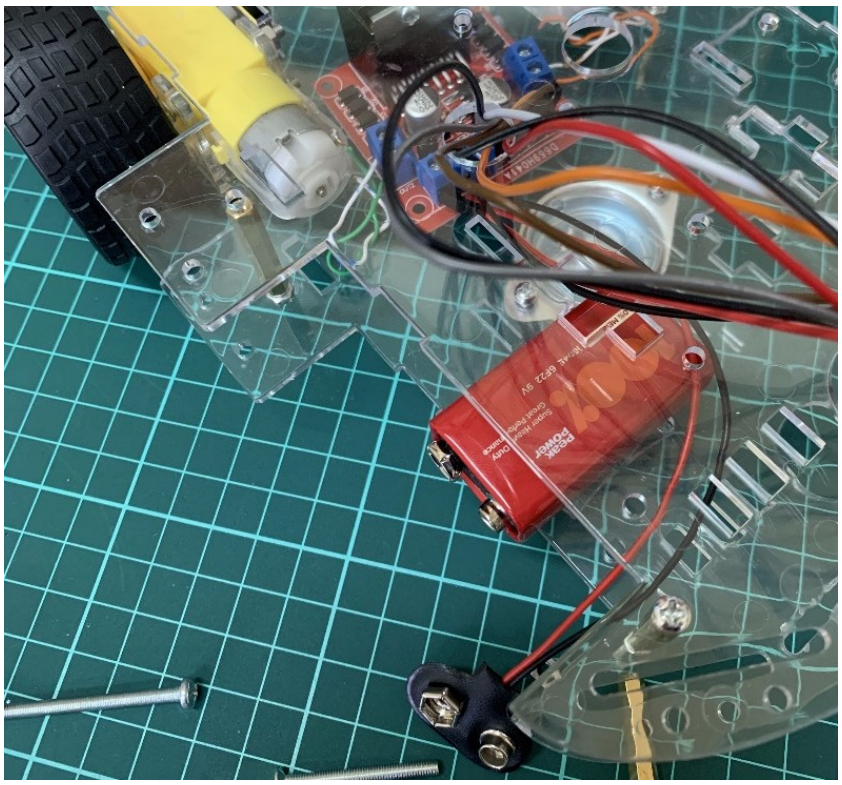


Hız Sensörü Alttan Görünüm

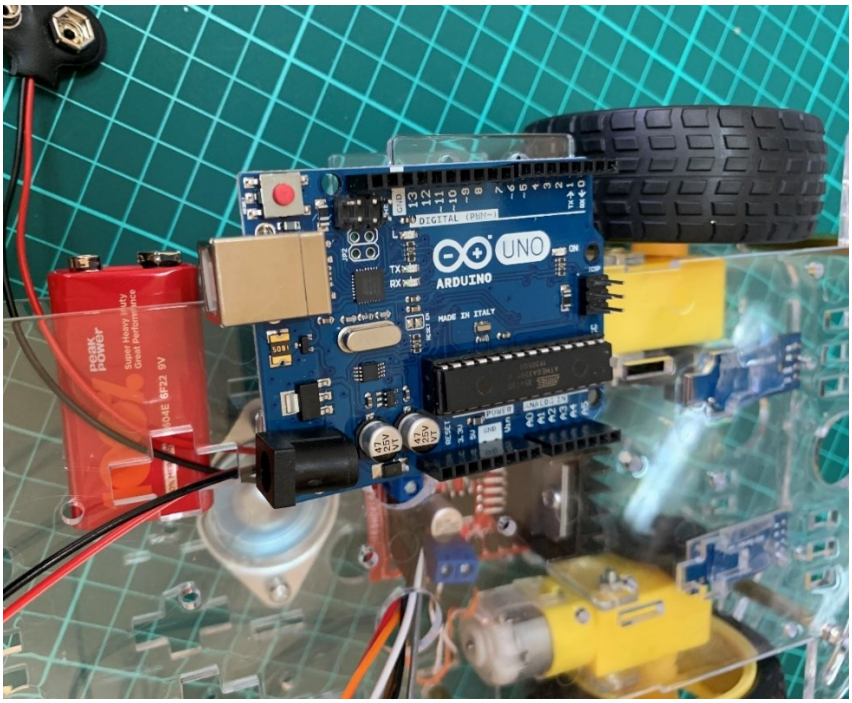


Hız Sensörü Üstten Görünüm

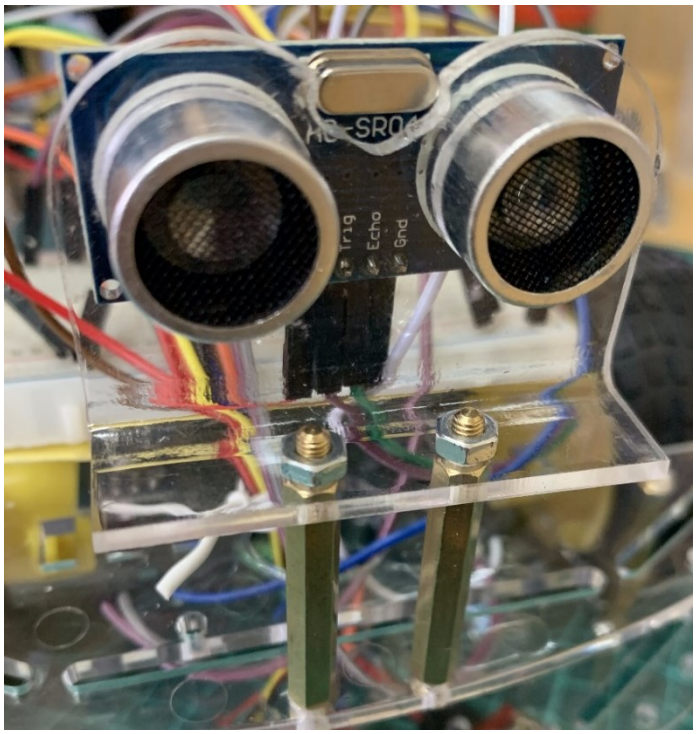
8) Üst pleksi gövde somun, aralayıcı ve vidalar kullanılarak alt gövde ile birleştirilir. Arkada ortada ve önde ikişer aralayıcı kullanılarak üs gövdenin esnemesi önlenebilir. Fakat araba tamamlanana kadar aralayıcıların sadece iki tanesinin somununun takılması önerilir.



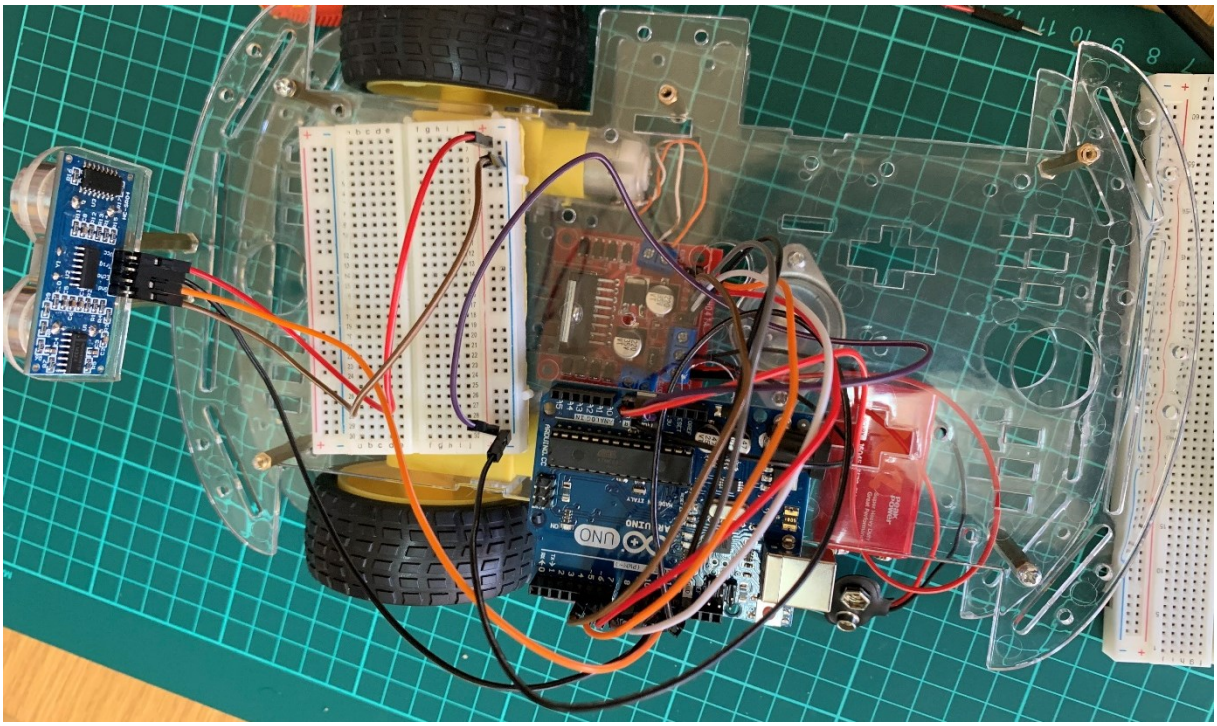
9) Arduino resimdeki gibi iki taraflı bant ile üst pleksiye yapıştırılır.



10) HC-SR04 mesafe sensörü sensör tutucu ve aralayıcı ile yükseltilerek üst pleksi gövdeye monte edilir. Burada öğrenciler aralayıcı kullanmak zorunda değildir. İhtiyaca göre başka şekillerde tasarımlar da yapılabilir.

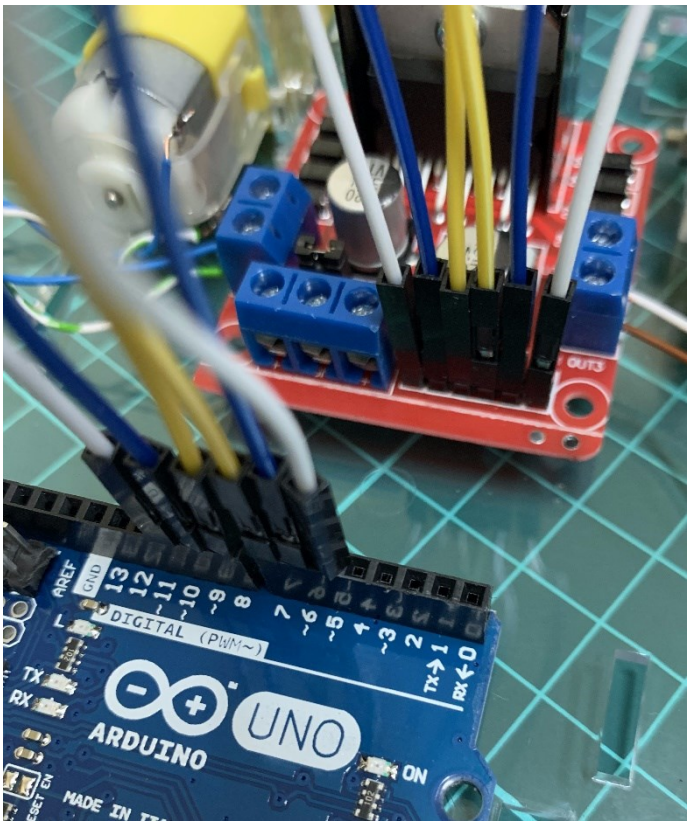


Ultrasonik Sensör Karşıdan Görünüm

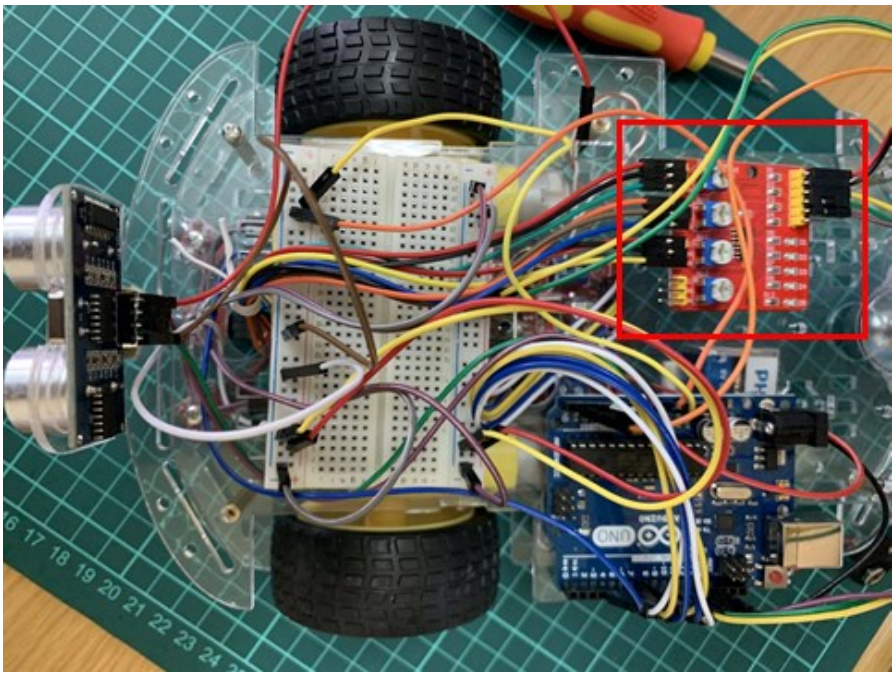


Ultrasonik Sensör Genel Görünüm

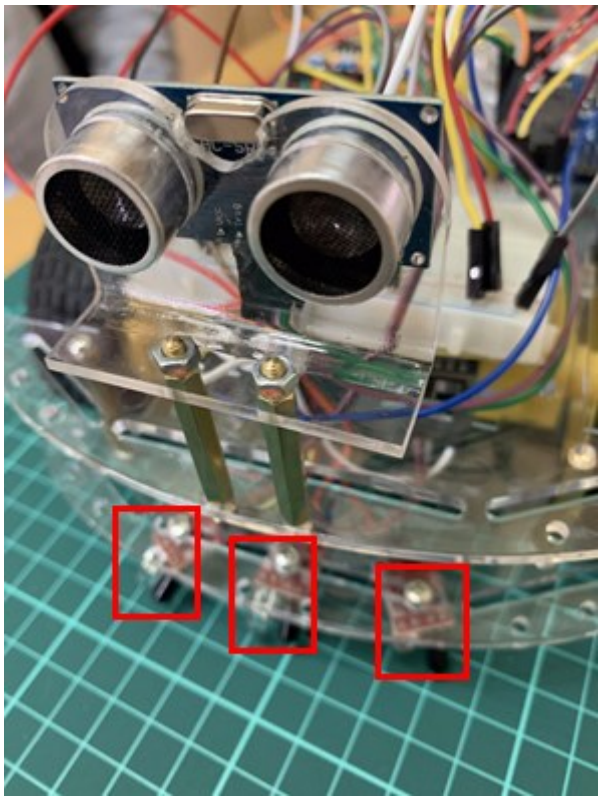
11) Breadboard ve Arduino Uno iki tarafı yapışkan bant ile üst pleksiye yapıştırıldıktan sonra kablo bağlantıları yapılır. L298N sürücü modülü ve Arduino Uno kablo bağlantıları yapılırken aşağıdaki resim yardımcı olabilir.



12) Kızılötesi sensörler ve sensör kartı arabaya yerleştirilir.

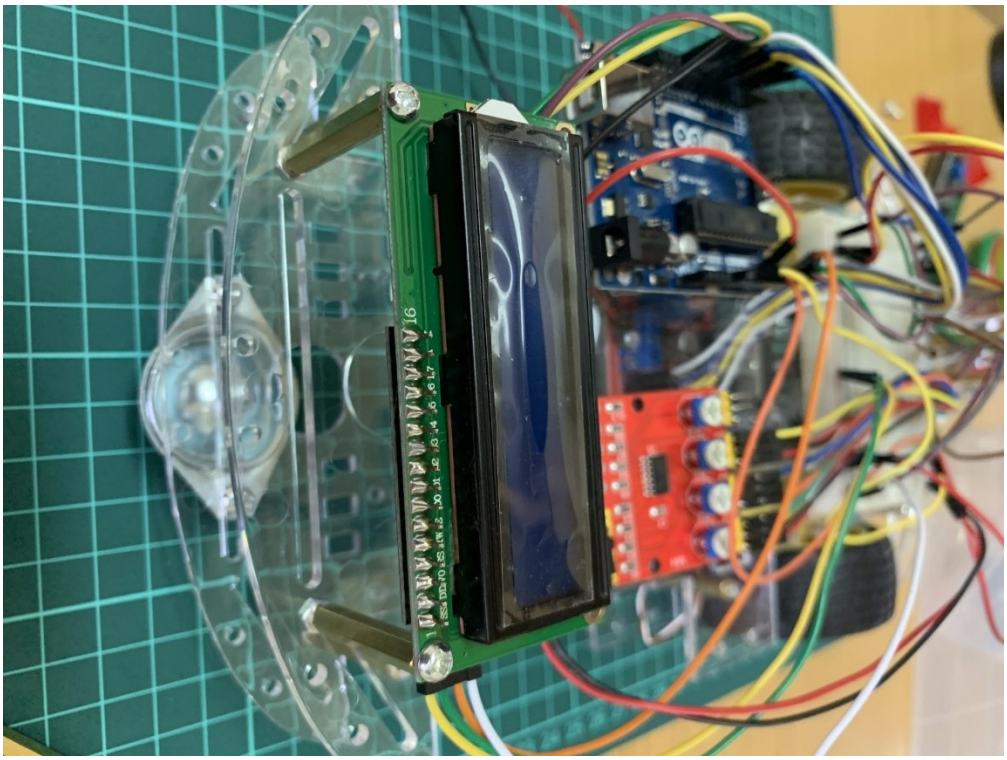


Kızılötesi Sensör Kartı Genel Görünüm



Kızılötesi Sensörler Önden Görünüm

13) Son olarak 2x16 veya 4x20 LCD arabaya aşağıdaki resimde gösterildiği gibi yerleştirilir.



EK-4-BonibonSeç Kaynak Kodu (Bölüm-9)

```
import processing.serial.*;
Serial hc05Port;
String ledDurum="KAPALI";
void setup(){
  size(500, 250);
  hc05Port = new Serial(this, "COM4", 9600); // ***** COM10 Uygun Port Ile Degistirilmeli ******
  hc05Port.bufferUntil('\n');
}
void serialEvent (Serial hc05Port){
  ledDurum = hc05Port.readStringUntil('\n');
}
void draw(){
  background(0, 128, 255);
  fill(0, 255, 128);
  stroke(33);
  strokeWeight(1);
  rect(175, 100, 150, 50, 10);
  fill(255);
  textSize(32);
  text("Bonibon Ayırma Etkinliği", 55, 50);
  text("Çalıştır",190, 135);
  textSize(24);
  if(mousePressed && mouseX>175 && mouseX<325 && mouseY>100 && mouseY<150){
    hc05Port.write('1');
    stroke(255,255,255);
    strokeWeight(4);
    noFill();
    rect(175, 100, 150, 50, 10);
    delay(1000);
  }
}
```

EK-5-ManyetikKayit Kaynak Kodu (Bölüm-9)

```
import processing.serial.*;
Serial hc05Port;
String ledDurum="KAPALI";
void setup(){
  size(500, 250);
  hc05Port = new Serial(this, "COM40", 9600); // ***** COM10 Uygun Port Ile Degistirilmeli ******
  hc05Port.bufferUntil('\n');
}
void serialEvent (Serial hc05Port){
  ledDurum = hc05Port.readStringUntil('\n');
}
void draw(){
  background(0, 128, 255);
  fill(0, 255, 128);
  stroke(33);
  strokeWeight(1);
  rect(50, 100, 150, 50, 10);
  rect(250, 100, 150, 50, 10);
  fill(255);

  textSize(32);
  text("MIKNATIS DETEKTÖRÜ", 55, 50);
  text("EVET",80, 135);
  text("HAYIR", 265, 135);
  textSize(24);
  text("Kayıt yapılsın mı?", 120, 90);


  if(mousePressed && mouseX>50 && mouseX<200 && mouseY>100 && mouseY<150){
    hc05Port.write('1');
    stroke(255,255,255);
```

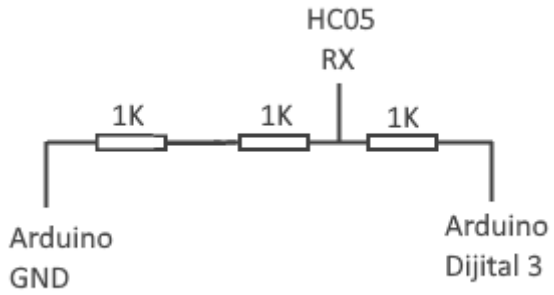
```
    strokeWeight(4);
    noFill();
    rect(50, 100, 150, 50, 10);
    delay(1000);
  }

  if(mousePressed && mouseX>250 && mouseX<450 && mouseY>100 && mouseY<150){
    hc05Port.write('2');
    stroke(255,255,255);
    strokeWeight(4);
    noFill();
    rect(250, 100, 150, 50, 10);
    delay(1000);
  }
}
```

EK-6-HC05 İsim Değişikliği (Bölüm-9)

A. Öncelikle HC05 ile Arduino Uno arasındaki bağlantılar yapılacaktır. Bağlantılar için aşağıdaki şema kullanılabilir.

HC05	**Arduino Uno**
VCC	5V
GND	GND
TX	2 numaralı dijital pin
RX	Voltaj bölücüden dijital 3'e
STATE	4 numaralı dijital pin

A.1. Şemada da söylendiği gibi HC05'in RX bacağı bağlantısı voltaj bölücüsü üzerinden yapılmalıdır. Bu iş için 3 adet 1K Ohm direnç ile aşağıdaki bağlantı kullanılabilir.



B. Aşağıdaki kod Arduino IDE kullanılarak Arduino Uno'ya yüklenir.

```
#include <SoftwareSerial.h>

SoftwareSerial hc05baglanti(2,3);


int statePini=4; //STATE bacağı 4 nolu dijital pine bağlanır


void setup() {
  pinMode(statePini,OUTPUT);
  digitalWrite(statePini,HIGH);
  Serial.begin(9600);
  hc05baglanti.begin(38400);
```

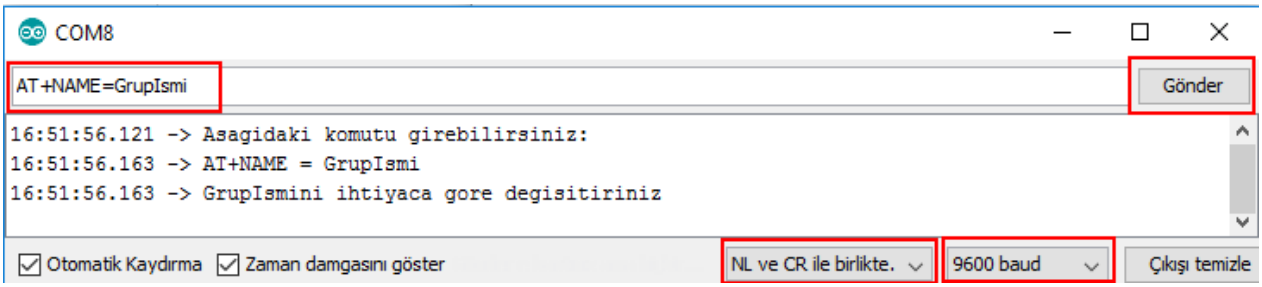
```
  Serial.println("Asagidaki komutu girebilirsiniz:");
  Serial.println("AT+NAME = GrupIsmi");
  Serial.println("GrupIsmini ihtiyaca gore degisitiriniz");
}

void loop() {
  if (hc05baglanti.available()){
    Serial.write(hc05baglanti.read());
  }
  if (Serial.available()){
    hc05baglanti.write(Serial.read());
  }
}
```

C. Aruino kartın güç bağlantısı kesilir. HC05 üzerinde EN pininin üzerinde bulunan butona basılı tutulur. Arduino karta yeniden güç verilir. HC05 üzerindeki led 3 saniye aralıklarla yanıyorsa yapılan işlemler doğrudur. Buton bırakılır ve bir sonraki adıma geçilir. Led sık yanıp sönüyorsa yapılan işlemler geçersizdir. B ve/veya A adımları yeniden tekrarlanır.

D. Arduino IDE'nin seri port ekranı açılır. Aşağıda seri port ekranını açmak için kullanılan menü gösterilmiştir.

Açılan ekranda aşağıdaki komutu girilir (grup ismi her öğrenci grubu için farklı verilir):

AT+NAME=GrupIsmi



EK-7-HC05 Kontrol Kaynak Kodu (Bölüm-12)

```
int ledPin=13;
char ledDurum = ' ';
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600);
}
void loop() {
  if(Serial.available() > 0){
    ledDurum = Serial.read();
 }
 if (ledDurum == '0') {
  digitalWrite(ledPin, LOW);
  Serial.println("KAPALI");
  ledDurum = ' ';
 }
 else if (ledDurum == '1') {
  digitalWrite(ledPin, HIGH);
  Serial.println("ACIK");
  ledDurum = ' ';
 }
}
```

EK-8-LED Aç-Kapa Kaynak Kodu (Bölüm-12)

```
import processing.serial.*;
Serial hc05Port;
String ledDurum="KAPALI";
void setup(){
  size(500, 250);
  hc05Port = new Serial(this, "COM4", 9600); // ***** COM10 Uygun Port Ile Degistirilmeli ******
  hc05Port.bufferUntil('\n');
}
void serialEvent (Serial hc05Port){
  ledDurum = hc05Port.readStringUntil('\n');
}
void draw(){
  background(0, 128, 255);
  fill(0, 255, 128);
  stroke(33);
  strokeWeight(1);
  rect(50, 100, 150, 50, 10);
  rect(250, 100, 200, 50, 10);
  fill(255);

  textSize(32);
  text("LED AC/KAPAT ETKINLIGI", 55, 50);
  text("LED'i Ac",60, 135);
  text("LED'i Kapat", 265, 135);
  textSize(24);
  text("LED DURUMU:", 120, 200);
  text(ledDurum, 285, 200);

  if(mousePressed && mouseX>50 && mouseX<200 && mouseY>100 && mouseY<150){
    hc05Port.write('1');
```

```
    stroke(255,255,255);
    strokeWeight(4);
    noFill();
    rect(50, 100, 150, 50, 10);
  }

  if(mousePressed && mouseX>250 && mouseX<450 && mouseY>100 && mouseY<150){
    hc05Port.write('0');
    stroke(255,255,255);
    strokeWeight(4);
    noFill();
    rect(250, 100, 200, 50, 10);
  }
}
```

EK-9-processingVisualizationMPU6050.pde Kaynak Kodu (Bölüm-12)

```
//************** DİKKAT ******************
//Error, disabling serialEvent() for COMX null hatası alınırsa
//Program kapatılıp yeniden çalıştırılır

import processing.serial.*;
Serial arduinoPortu;
String arduinoGelenVeri="";
float xAcisi=0, yAcisi=0,zAcisi=0;
long simdi=0;
int a=1;
void setup() {
  size (800, 600, P3D);
  arduinoPortu = new Serial(this, "COM8", 38400); // COM8 Arduinonun Bağlı Olduğu Port ile Değiştirilecektir
  arduinoPortu.bufferUntil('\n');
}
void draw() {

 translate(width/2, height/2, 0);
 background(154,192,205);

 rotateX(radians(yAcisi));
 rotateZ(radians(xAcisi));
 rotateY(radians(zAcisi));

 textSize(30);
 fill(16, 79, 139);
 box (400, 50, 200); // Draw box

 textSize(25);
```

```
 fill(255, 255, 255);
 text("TÜBİTAK - DENEYAP", -125, 10, 150);

 fill(255, 255, 255);
 textSize(25);
 text("TÜBİTAK - DENEYAP", -125, 80);
 }

void serialEvent (Serial arduinoPortu) {
if (a==1){
  while(millis()-simdi<1000){}
  a=2;}
    arduinoGelenVeri = arduinoPortu.readStringUntil('\n');
    if (arduinoGelenVeri != null) {
      arduinoGelenVeri = trim(arduinoGelenVeri);
    String acilar[] = split(arduinoGelenVeri, '<');
    if (acilar.length > 1) {
      xAcisi = float(acilar[0]);
      if(Float.isNaN(xAcisi)){xAcisi=0;}
      yAcisi = float(acilar[1]);
      if(Float.isNaN(yAcisi)){yAcisi=0;}
      zAcisi = float(acilar[2]);
      if(Float.isNaN(zAcisi)){zAcisi=0;}
    }
  }
}
```

EK-10-Proje Hazırlıyorum

Öğrencilerin öğrendiklerini grup halinde uygulayabileceği proje geliştirme sürecinin, kursun son 6 haftasında planlanması beklenmektedir. Projelerin tanımlama, hazırlık, empati, fikir üretme, geliştirme-test etme ve sunum gibi etkinliklerle yapılması ve rehber öğretmenler gözetiminde ders dışı etkinliklerle de destek olunması önerilmektedir.

1. Projem (Giriş)

Etkinliğin Amacı: Gruplara açık uçlu bir proje vererek öğrencilerin yaratıcılıklarını tetiklemek, problem çözme, analitik düşünme ve grup halinde çalışma becerilerini artırmaktır. Öğrenciler bu etkinlikte kendi elektronik devrelerini geliştirebildiklerini görecekler ve bu sayede de kendilerine olan özgüvenleri artacaktır.

Etkinliğin Konusu: Proje çalışması

Proje çalışması: Bu etkinlikten itibaren üç hafta boyunca gruplar projelerini tanımlama ve tasarlama üzerinde çalışacaklardır. Grupların son iki hafta tanımlayıp tasarladıkları elektronik devreleri oluşturup gerekli programları geliştirmeleri beklenmektedir. Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip sonra bu probleme kendi tasarladıkları elektronik devreler ile çözüm üreteceklerdir. Öğrencilerin problem üzerinde sistematik bir şekilde çalışmaları için grupların problemi analiz etmeleri ve kendi çözüm önerilerini tasarlamaları, geliştirmeleri ve değerlendirmeleri istenecektir.

Bu süreçte karşılaşılacak elektronik devre tasarımı ve program oluşturma ile ilgili problemlerde eğitmenler, öğrencilere uygun zaman ve miktarda destek olacaklardır. Bu destek öğrencilerin ihtiyacından ne az ne de fazla olmalıdır.

Öğrencilere proje çalışmasında son ürünü geliştirmek için Tasarım ve Üretim dersinde uygulanan tasarım odaklı düşünmenin beş aşamasının uygulanacağı söylenir.

1. Empati kurma: Hedef kullanıcı kitlenin ihtiyaç ve beklentilerini irdeleme. (1. hafta)
2. Tanımlama: Kullanıcılardan alınan bilgilerden çıkan sonuçları yorumlayıp ihtiyacı öngörme. (2. Hafta)
3. Fikir üretme: Çözüm için çeşitli, yaratıcı ve yenilikçi fikirler üretme. (3. Hafta)
4. Prototip geliştirme: Fikirlerin hızlıca uygulandığı prototipler hazırlama. (4. Hafta)
5. Test etme: Ortaya çıkan ürünü yeniden hedef kitle ile değerlendirme. (5. Hafta)

Sonraki haftaya hazırlık: Bir sonraki hafta (birinci hafta) için öğrencilerin öncelikle çevrelerinde ve günlük hayatlarında elektronik devreler kullanılarak çözülebilecek bir problemi belirlemeleri gerektiği iletilir. Problemi belirlemeden önce Empati kurma adımını gerçekleştirmeleri söylenir. Uygun bir problem sorusu belirlemeden önce problemi anlamak gerekir. Belirlenen problem projenin zorluklarını ve hedeflerini tanımlamaya yardımcı olur. Sonrasında problemle ilgili yoğun bir araştırma ve gözlem yapılması gerekir. Süreç projenin tamamlanması için gerekli bilgiler sağlar ve ortam hakkında bilgiler sunar. Süreç sonunda çok vakit ve emek harcanarak geliştirilen elektronik devrelerin, insanların işine yaramasını, bir problemi çözmesini ve hayatlarını kolaylaştırmasını isteriz. Empati süreci sonunda

tasarlanacak elektronik devrelerin kullanılabilmesi için planlanan /öngörülen ortamlardaki ihtiyaçların ve beklentilerin iyi belirlenmesi gerekmektedir.

Birinci haftada öğrencilerin proje konularını ve gerekçelerini ifade etmelerinin isteneceği belirtilmelidir.

2 Projem (Empati)

Etkinliğin Amacı: Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları elektronik devreler ile çözüm üreteceklerdir.

Sunum: Grupların elektronik devre projesi için yaptıkları empati çalışması sonucu belirledikleri problem tanımlarını sınıfta açıklamaları istenir. Grupların açıklamalarında aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati çalışması sonucunu sunmaya başlamadan önce hazırlanma, gözden geçirme ve tekrar etme için 10 dakika süre verilir.

- Problem cümlesi: elektronik devrenin ne yapması isteniyor?
- Elektronik devrenin kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Elektronik devrenin kullanılması planlanan ortamdaki beklentiler nelerdir?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Rehber öğretmene Not: Öğrenci tarafından belirlenen elektronik devrenin görevinin gerçekleştirilebilir olması önemlidir. Laboratuvarlarda bulunan imkânlarla bu görevin başarılabiliyor olması gerekmektedir. Bu safhada gerçekleştirilemeyecek bir görevin seçilmesi veya öğrencinin bu şekilde bir projeye yönlendirilmesi, proje sürecinde öğrencilerin vakit kaybetmelerine ve motivasyonlarının olumsuz şekilde etkilenmesine neden olabilir. Bu yüzden rehber öğretmenin proje önerilerini çok iyi incelemesi/irdelemesi ve proje önerilerinin gerçekleştirilebilir/yapılabilir olduğundan emin olması gerekir.

Sonraki haftaya hazırlık: Gruplara ikinci hafta için belirledikleri projeler için tanımlama adımlarını gerçekleştirmeleri istenir. Tanımlama adımında, empati adımında toplanan bilgiler bir bütün haline getirilir. Süreç esnasında çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir.

3. Projem (Tanımlama)

Etkinliğin Amacı: Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir elektronik devre ile çözüm üreteceklerdir.

Sunum: Grupların belirledikleri elektronik devre projesi için yaptıkları empati çalışması ve sonrasında gerçekleştirdikleri tanımlama adımı sonucunu sınıfta açıklamaları istenir. Grupların açıklamalarında aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati ve tanımlama çalışması sonuçlarını sunmaya başlamadan önce hazırlanma, gözden geçirme ve tekrar etme için gruplara 10 dakika süre verilir.

- Problem cümlesi: elektronik devrenin ne yapması isteniyor?
- Elektronik devrenin kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Elektronik devrenin kullanılması planlanan ortamdaki beklentiler nelerdir?
- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?
- Elektronik devrenin belirlenen hedefleri gerçekleştirebilmesi için nasıl bir tasarımı olmalıdır?

- Elektronik devrenin belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Sonraki haftaya hazırlık: Gruplara üçüncü hafta için, belirledikleri işlemlerin gerçekleştirilebilmesi için fikir üretme aşamasında olası çözümleri belirlemeleri ve uygun olanları seçmeleri gerektiği söylenir. Fikir üretme aşamasında, empati ve tanımlama adımında toplanan bilgiler bir bütün haline getirilir ve çözüm önerisi taslak olarak üretilir. Süreç esnasında çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir.

Rehber öğretmen için tanım: Fikir üretme aşamasında çok sayıda fikir ve çözüm üretmek önemlidir. Bu çalışmayı 'Beyin Fırtınası' tekniği ile gerçekleştirebilirsiniz! Bunun dışında problem alanını genişleterek olayların dışından bakmak önerilir. Bu şekilde sorunu incelemek ve çözüm yolu bulmak için alternatif yollar aranmaya başlanır.

4. Projem (Fikir Üretme)

Etkinliğin Amacı: Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir elektronik devre ile çözüm üreteceklerdir.

Sunum: Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir. Bu hafta, empati ve tanımlama adımında toplanan bilgiler bir bütün haline getirilir ve çözüm önerisi taslak olarak üretilir. Alternatif çözüm yaklaşımlarının önerilmesi gereklidir, bunlar çok detaylı ve tamamlanmış fikirler olmak zorunda değildir. Fakat, öneriler, gerçekçi ve üretilmeye uygun çözüm fikirleri olmalıdır.

Grupların belirledikleri elektronik devre projesi için yaptıkları fikir üretme çalışması sonucunu söz alarak sınıfta açıklamaları istenir. Grupların açıklamalarında aşağıdaki bilgileri içermesi gerektiği belirtilir. Bundan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için 10 dakika süre verilir.

- Problem cümlesi: elektronik devrenin ne yapması isteniyor?
- Elektronik devrenin kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Elektronik devrenin kullanılması planlanan ortamdaki beklentiler nelerdir?
- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?
- Elektronik devrenin belirlenen hedefleri gerçekleştirebilmesi için nasıl bir tasarımı olmalıdır?
- Elektronik devrenin belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?
- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?
- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?
- Seçilen programlama çözümünün şematik veya maddeler halinde gösterimi (algoritması) nasıldır?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Sonraki haftaya hazırlık: Gruplara dördüncü hafta için, üretilen fikirlere bağlı olarak elektronik devrenin tasarlanıp programlanması aşamasına geçileceği söylenir.

5. Projem (Geliştirme ve Test Etme)

Etkinliğin Amacı: Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir elektronik devre ile çözüm üreteceklerdir.

Prototip geliştirme: Grupların belirledikleri elektronik devre projesi için yaptıkları fikir üretme çalışması sonucunda karar verdikleri elektronik devreyi oluşturup ve programa başlamaları gerekir.

Grupların iki hafta boyunca fikir üretme aşamasında belirledikleri çözümlerini uygulamaları, elektronik devre tasarımı ve programını geliştirmeleri istenir. Çözüm için geliştirilen elektronik devrelerde ve programlarda bazı aksaklıkların / problemlerin olması doğal bir durumdur. İlk denemede planlanan elektronik devrenin hemen oluşturulması ve geliştirilen programın sorunsuz bir şekilde çalışması çok karşılaşılan bir durum değildir. Bu gibi sorunların olmasının / ortaya çıkmasının doğal olduğu rehber öğretmen tarafından öğrencilere söylenmeli ve/veya hissettirilmelidir. Ortaya çıkan sorunlar için çözüme yönelik bir plan hazırlanıp uygulanmalıdır. Çözümün sağlanması için gruplara aşağıdakilere benzer bir süreç önerilebilir:

a. Problemler maddeler halinde listelenmeli (beklenilen elektronik devre davranışı ile gerçekleşen arasındaki farklar),
b. Problemler arasındaki ilişkiler belirlenmeli,
c. Öncelik sırası belirlenmeli,
d. Öncelik sırasına göre çözümler geliştirilmeli (çözüm için hipotezler oluşturulmalı),
e. Geliştirilen çözümler (hipotezler) denenmeli ve
f. Elektronik devrenin her koşulda istenilen şekilde çalışması için gerekli düzenlemeler yapılmalı.

Bütün grupların hazırladıkları prototipleri (tam olarak çalışmasa bile) rehber öğretmene göstermeleri sağlanır. Bu sayede rehber öğretmenin geliştirilen ürünü test etmesi / görmesi, çeşitli çözüm önerileri sunması ve/veya gözden kaçan noktaları ortaya çıkarması sağlanabilir.

Test etme: Ortaya çıkan son ürünü hedef kitle ile / hedef ortamda değerlendirmek geliştirilen çözümlerin kusursuz bir şekilde çalışması için önemlidir. Hedef kitleye sınıfta ulaşmak her zaman mümkün olmayabilir. Bu yüzden grupların ortaya çıkardıkları ürünleri diğer öğrencilerin kendi başlarına deneyimlemelerine izin verilmesi gözden kaçan unsurların ortaya çıkarılmasına yardımcı olabilir.

6. Projem (Sunum)

Sunum: Grupların proje sunumları için hazırlanmaları gerekiyor. Sunum yapacak gruplar projelerini bir senaryo içerisinde tiyatral olarak sunabilir.

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ

ORTAOKUL

DENEYAP TÜRKİYE