

DENEYAP

Teknoloji Atölyeleri

ELEKTRONİK PROGRAMLAMA VE NESNELERİN İNTERNETİ

LİSE

Dr. Öğr. Üyesi Erman UZUN

Doç. Dr. İlker YAKIN

Dr. Öğr. Üyesi Ali GÖK



TÜBİTAK Deneyap Kitapları 6

**ELEKTRONİK PROGRAMLAMA VE
NESNELERİN İNTERNETİ**
LİSE

Dr. Öğr. Üyesi Erman UZUN
Doç. Dr. İlker YAKIN
Dr. Öğr. Üyesi Ali GÖK

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2021

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel materyaller, TÜBİTAK'tan yazılı izin alınmadan
tümüyle veya kısmen çoğaltılamaz ve yayımlanamaz.
Kitabın PDF formatındaki elektronik nüshasına
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi> adresinden ulaşılabilir.
TÜBİTAK Deneyap Kitapları DENEYAP TÜRKİYE Projesi kapsamında hazırlanmıştır.

ISBN 978-605-312-441-2
Yayıncı Sertifika No: 47703

Yayın Tarihi: 2023

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editörler: İpek PİRİROĞLU GENCER, Kübra BAL ÇETİNKAYA
Düzeltili: Mustafa ORHAN
Telif İşleri Sorumlusu: Havva Hilal KAÇAR

TÜBİTAK Bilim ve Toplum Başkanlığı
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 96 50
e-posta: deneyap@tubitak.gov.tr
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi>

İçindekiler

İÇİNDEKİLER.....	iii
SUNUŞ.....	viii
ÖĞRETİM KILAVUZU	1
Giriş.....	1
<i>Gözle-Uygula-Tasarla-Üret-Değerlendir Öğrenme Döngüsü</i>	2
ETKİNLİK TABANLI PROGRAM VE SİRAL ÖĞRENME	4
EŞLİ PROGRAMLAMA	5
GRUPLAR ARASI İLETİŞİM.....	5
DÖNEM PROJESİ	6
DERSTE KULLANILAN SET	6
LABORATUVAR KULLANIM YÖNERGESİ.....	7
SIKÇA SORULAN PEDAGOJİK SORULAR	7
1. BÖLÜM - ELEKTRONİK PROGRAMLAMAYA GİRİŞ	10
1. ADIM: GÖZLE VE UYGULA	11
1.1 <i>Gözle - Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç</i>	11
1.2 <i>Uygula- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç</i>	12
1.3 <i>Gözle ve Uygula - Basit Bir Devre Oluşturma (Öğrenci 1)</i>	12
1.4 <i>Gözle ve Uygula - Breadboard ve Arduino Uno ile Basit Bir Devre Oluşturma (Öğrenci 2)</i>	14
1.5 <i>Uygula- Basit Bir Devre Oluşturma ve Direnç Değerlerini Değiştirme (Öğrenci 1)</i> 16	
1.6 <i>Gözle - Arduino UNO ve Arduino IDE Geliştirme Ortamına Giriş</i>	18
1.7 <i>Gözle - Uygula Arduino UNO ile LED Yakıp Söndürme (Öğrenci 1)</i>	21
1.8 <i>Uygula- Flip Flop (Öğrenci 2)</i>	23
1.9 <i>Uygula - Trafik Işığı (Öğrenci 1)</i>	25
1.10 <i>Uygula- Araba Yarışı Başlama Işıkları (Öğrenci 2)</i>	27
2. ADIM: TASARLA VE ÜRET	29
2.1 <i>Tasarla - Kavşaktaki Trafik Lambaları</i>	29
2.2 <i>Üret- Kavşaktaki Trafik Lambaları</i>	30
2.3 <i>Tasarla - Müzik ile Uyumlu Hareket Eden LED Uygulaması</i>	32
2.4 <i>Üret- Müzik ile Uyumlu Hareket Eden LED Uygulaması</i>	32
3. ADIM: DEĞERLENDİR.....	33
4. İLAVE ETKİNLİK	33
4.1 <i>Transit Geçişli Bir Kavşaktaki Trafik Lambaları</i>	33
2. BÖLÜM – TEMEL DEVRE ELEMANLARI İLE ARDUİNO UNO VE KODLAMA ARACI ARDUİNO IDE.....	34
1. ADIM: GÖZLE VE UYGULA	35
1.1 <i>Gözle ve Uygula - Breadboard, Seri/Paralel Bağlantı, Analog/Dijital Veri ve Pinler</i>	35
1.2 <i>Uygula- LED Parlaklığı (Öğrenci 1)</i>	39
1.3 <i>Gözle - Direnç, Direnç Okuma</i>	40
1.4 <i>Gözle ve Uygula - Buzzer</i>	41

1.5 Uygula – Şarkı Çalan Buzzer (Öğrenci 1).....	43
1.6 Gözle ve Uygula - Potansiyometre ile LED parlaklığını ayarlama ve Seri Haberleşme	44
1.7 Gözle ve Uygula – Buton Kontrollü LED Yakma	45
1.8 Uygula – Buton ile Soldan-Sağa ve Sağdan-Sola Hareket Eden LED Uygulaması... ..	47
2. ADIM: TASARLA VE ÜRET	48
2.1 Tasarla- Bir Buton Dört Hareketli LED Uygulaması.....	48
2.2 Üret- Bir Buton Dört Hareketli LED Uygulaması.....	50
2.3 Tasarla- Butonlar ile Piyano Yapımı.....	50
2.4 Üret- Butonlar ile Piyano Yapımı.....	52
3. ADIM: DEĞERLENDİR.....	52
3. BÖLÜM – DİYOT VE LED UYGULAMALARI	53
1. ADIM: GÖZLE VE UYGULA	54
1.1 Gözle – Yarı İletkenler	54
1.2 Gözle – Diyot	54
1.3 Gözle ve Uygula - Avometre ile direnç, gerilim ve akım ölçüyorum (Öğrenci 1)	56
1.4 Uygula- Diyot (Öğrenci 2).....	57
1.5 Gözle ve Uygula – Buzzer diyot uygulaması (Öğrenci 1).....	58
1.6 Gözle ve Uygula - Diyot ile “VE” Kapısı Yapıyorum (Öğrenci 2).....	59
1.7 Gözle ve Uygula- Kaç Buzzer’dan Ses Çıkıyor? (Öğrenci 1).....	62
1.8 Gözle: RGB LED ile Farklı Renkler	65
1.9 Uygula- Ortak Ucu Belirleme (Öğrenci 2)	70
1.10 Gözle - Uygula: Arduino IDE ile Ara Renkler (Öğrenci 1).....	70
1.11 Uygula- Gökkuşağındaki 7 Temel Renk (Öğrenci 2).....	71
1.12 Gözle ve Uygula – 3 potansiyometre ile RGB LED kontrolü (Öğrenci 1).....	72
1.13 Gözle ve Uygula- LED Küpü Yapıyorum (Öğrenci 2).....	74
1.14 Gözle ve Uygula- LED Küpünü Kodluyorum (Öğrenci 1).....	76
1.15 Gözle- Uygula LED Küpünü Farklı Aralıklarla Yakıp Söndürüyorum (Öğrenci 2)	77
1.16 Uygula- LED Küpünü Farklı Aralıklarla Yakıp Söndürüyorum (Öğrenci 1).....	78
1.17 Uygula- Katlar Yanıp Sönüyor (Öğrenci 2).....	79
2. ADIM: TASARLA VE ÜRET	81
2.1 Tasarla - Şarkıyla Uyumlu LED Küpü.....	81
2.2. Üret - Şarkıyla Uyumlu LED Küpü.....	81
2.3 Tasarla - Buzzerla Çalınan Şarkıyla Uyumlu LED Küpü.....	81
2.4 Üret- Buzzerla Çalınan Şarkıyla Uyumlu LED Küpü.....	82
3. ADIM: DEĞERLENDİR.....	82
4. İLAVE ETKİNLİK	82
4.1 LED Küpü Animasyonu (Yarışma).....	82
4. BÖLÜM - TRANSİSTÖR, KONDANSATÖR, ULTRASONİK MESAFE VE SES SENSÖRÜ	84
1. ADIM: GÖZLE VE UYGULA	85
1.1 Gözle ve Uygula – LDR ile LED Uygulaması (Öğrenci 1).....	85
1.2 Gözle- Uygula: Transistör	87
1.3 Uygula– Transistörün Anahtarlama için Kullanımı.....	91
1.4 Gözle – Uygula: Transistör ile Değil Kapısı Yapıyorum (Öğrenci 2).....	92
1.5 Gözle-Uygula Transistör ile Karanlıkta Yanan LED (Öğrenci 1).....	93
1.6 Gözle-Uygula: Kondansatör ve Akım İlişkisi (Öğrenci 2).....	94
1.7 Uygula: Kondansatör Değeri Okuma ve Ölçümü (Öğrenci 1).....	96

1.8 Gözle: Transistör ve Kondansatörü Kullanıyorum (Öğrenci 2).....	97
1.9 Uygula: LED'in Yanma Süresini Değiştiriyorum (Öğrenci 1).....	98
1.10 Uygula: Butona Basıldığında Sönüp Bir Süre Sonra Yanan LED (Öğrenci 2).....	98
1.11 Gözle-Uygula: HC-SR04 Ultrasonik Mesafe Sensörü ile Mesafe Ölçüyorum (Öğrenci 1).....	99
1.12 Gözle-Uygula: 8 LED'li Park Sensörü (Öğrenci 2).....	102
1.13 Gözle-Uygula: KY-038 Ses Sensörü ile Gürültü Dedektörü (Öğrenci 1).....	103
2. ADIM: TASARLA VE ÜRET.....	105
2.1 Tasarla-Üret: Kapalı Otopark Park Destek Sistemi.....	105
2.2 Üret- Kapalı Otopark Park Destek Sistemi.....	105
3. ADIM: DEĞERLENDİR.....	107
4. İLAVE ETKİNLİK.....	108
4.1 Ses Sensörünü kullanarak kendi Ekolayzırımı yapıyorum.....	108
4.2 Ambulans Sirenine Duyarlı Kavşak.....	109
5.BÖLÜM - EKРАН VE GÖSTERGE ARAÇLARININ KULLANIMI.....	110
1. ADIM: GÖZLE VE UYGULA.....	111
1.1 Gözle – Ekranlar (Gösterim Araçları).....	111
1.2 Gözle – 7 Bölümlü Gösterge (7 Segment Display).....	112
1.3 Gözle ve Uygula - 7 Segment Gösterge'nin ortak anot veya ortak katot olduğunu tespit etme (Öğrenci 1).....	113
1.4 Gözle ve Uygula - 7 Segment Gösterge LED'leri yakıp söndürme (Öğrenci 2).....	115
1.5 Gözle ve Uygula - 7 Segment Gösterge ile rakamları oluşturma (Öğrenci 1).....	117
1.6 Uygula- 7 Segment Gösterge ve Buzzer ile geri sayım (Öğrenci 2).....	121
1.7 Gözle ve Uygula - 7 Segment Gösterge ve Buton ile Sayaç Yapımı (Öğrenci 1).....	123
1.8 Gözle – LCD (Liquid Crystal Display) Ekranlar.....	126
1.9 Gözle – Arduino IDE'de Kütüphane (Library).....	129
1.10 Gözle ve Uygula – LCD Ekranı İlk Yazımı Yazıyorum (Öğrenci 2).....	131
1.11 Gözle ve Uygula – LCD Ekran için Kullanılan Komutlar (Öğrenci 1).....	133
1.12 Gözle ve Uygula – LCD Ekranında Sağa-Sola Hareketli Yazı (Öğrenci 2).....	135
1.13 Gözle ve Uygula – LCD Ekranı ve Ekran Işığını Açma-Kapatma (Öğrenci 1).....	136
1.14 Gözle ve Uygula – LCD Ekranında Özel Karakter Oluşturma ve Gösterme (Öğrenci 2).....	137
1.15 Gözle ve Uygula– Arduino UNO'ya Birden Fazla I2C Modüllü LCD Ekran Bağlama ve LCD Ekranların Bağlantı Adreslerini Değiştirme (Öğrenci 1).....	140
2. ADIM: TASARLA VE ÜRET.....	144
2.1 Tasarla - 3 Ekran, 1 Yedi Segment Display ve 8 Butonlu Basketbol Skorboard Yapımı.....	144
2.1 Üret- 3 Ekran, 1 Yedi Segment Display ve 8 Butonlu Basketbol Skorboard Yapımı.....	146
3. ADIM: DEĞERLENDİR.....	149
6.BÖLÜM - DC, SERVO VE STEP MOTORLARIN KULLANIMI.....	150
1. ADIM: GÖZLE VE UYGULA.....	151
1.1 Gözle – DC Motor Nedir?.....	151
1.2 Gözle-Uygula: Butona basınca çalışan vantilatör (Öğrenci 1).....	153
1.3 Gözle ve Uygula - Transistör ile DC Motor Sürme (Öğrenci 2).....	154
1.4 Gözle– DC Motor Sürücülerini Kullanma.....	156
1.5 Gözle ve Uygula – Merdaneli Çamaşır Makinesi (Öğrenci 1).....	158
1.6 Gözle ve Uygula - Servo Motorlar (Öğrenci 1).....	161

1.7 Uygula – Hareketli Sonar ile Mesafe Ölçme (Öğrenci 2)	163
1.8 Uygula – Joystick Kontrollü Hareketli Sonar ile Sesli ve Görsel Uyarı Sistemi (Öğrenci 1).....	164
1.9 Gözle ve Uygula – Step Motor (Öğrenci 2)	167
2. ADIM: DEĞERLENDİR.....	175
3. ADIM: EK ETKİNLİK -	176
7. BÖLÜM - KÂĞIT UÇAK FIRLATMA RAMPASI	177
2. ADIM: TASARLA VE ÜRET	177
2.1 Tasarla – Kâğıt Uçak Fırlatma Rampası.....	178
2.2 Üret- Kâğıt Uçak Fırlatma Rampası	183
3. ADIM: DEĞERLENDİR.....	184
4. ADIM: EK ETKİNLİK - YARIŞMA	184
5. PROJEME BAŞLIYORUM (GİRİŞ).....	184
8.BÖLÜM- BLUETOOTH KONTROLLÜ 3 TEKERLİ MOBİL ROBOT.....	187
1. ADIM: GÖZLE VE UYGULA	188
1.1 Gözle - L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı	188
1.2 Gözle ve Uygula – 3 Tekerlekli Mobil Robot Yapımı	190
1.3 Uygula – Mobil Robotu Kare Şeklinde Hareket Ettirme (Öğrenci 1)	195
1.4 Uygula- Mobil Robotu Farklı Geometrik Şekillerde Hareket Ettirme (Öğrenci 2)..	196
1.5 Uygula- Mobil Robotu Daire Şeklinde Hareket Ettirme (Öğrenci 1).....	199
1.6 Gözle ve Uygula – Bluetooth Kablosuz İletişim Modülü	200
2. ADIM: TASARLA VE ÜRET	206
2.1 Mobil Robot Yarışı.....	206
2.2 Tasarla - Mobil Robot Yarışı.....	206
3. ADIM: DEĞERLENDİR.....	207
4. PROJE GELİŞTİRMEDE EMPATİ	208
9.BÖLÜM – ÇİZGİ İZLEYEN MOBİL REHBER ROBOT	211
1. ADIM: GÖZLE VE UYGULA	212
1.1 Gözle ve Uygula – SD Kart Modülü	212
1.2 Gözle ve Uygula – Kızılötesi (Infrared) Sensör	214
1.3 Uygula – Çizgi İzleyen Mobil Robot (Öğrenci 1)	216
2. ADIM: TASARLA VE ÜRET	218
2.1 Tasarla - Müzeler için Mobil Rehber Robot.....	218
2.2 Tasarla - Mobil Rehber Robot Tasarımı.....	220
2.3 Üret- Mobil Rehber Robot Tasarımı.....	221
3. ADIM: DEĞERLENDİR.....	224
4. PROJEYİ TANIMLIYORUM	225
10.BÖLÜM - SICAKLIK, NEM SENSÖRÜ VE Wİ-Fİ MODÜLÜNÜN KULLANIMI	229
1. ADIM: GÖZLE VE UYGULA	230
1.1 Gözle ve Uygula – DHT11 Sıcaklık ve Nem Sensörü Kullanımı	230
1.2 Gözle ve Uygula – Wi-Fi Modülü	231
2. ADIM: TASARLA VE ÜRET	235
2.1 Tasarla- Mobil Uzay Keşif Robotu	235
2.2 Tasarla - Mobil Uzay Keşif Robotu Tasarımı.....	235
2.3 Üret- Mobil Uzay Keşif Robotu Tasarımı.....	236

3. ADIM: DEĞERLENDİR.....	240
4. PROJEYE YÖNELİK FİKİR ÜRETME	240
11.BÖLÜM- SENSÖRLER İLE IOT UYGULAMALARINA GİRİŞ	245
1. ADIM: GÖZLE VE UYGULA	246
1.1 Gözle ve Uygula – Eğim (Tilt) Sensörü	246
1.2 Gözle ve Uygula – Hareket (PIR) Sensörü (Öğrenci 2)	248
1.3 Gözle ve Uygula – RFID Kart Okuyucu ile Elektronik Kapı Kilidi Oluşturma (Öğrenci 1).....	252
1.4 Gözle ve Uygula – Keypad Kullanımı (Öğrenci 2).....	256
1.5 Gözle ve Uygula – Su Seviyesi/Yağmur Sensörü Kullanımı	260
1.6 Gözle ve Uygula – Gaz Sensörü Kullanımı.....	262
2. ADIM: TASARLA VE ÜRET	264
2.1 Tasarla- Nükleer Santraller için Kaza Simülasyonu ve Güvenlik Protokolü	264
2.2 Tasarla – Nükleer Santraller için Güvenlik Protokolü Tasarımı	265
2.3 Üret- Nükleer Santraller için Güvenlik Protokolü Tasarımı	266
3. ADIM: DEĞERLENDİR.....	267
4. PROJEYE YÖNELİK PROTOTİP GELİŞTİRME	268
12.BÖLÜM – PROJİYİ TEST ETME	274
13.BÖLÜM – PROJE DEFTERİNİN TESLİM EDİLMESİ	281

Sunuş

Her gün gelişen teknolojiyi takip etmek ve hatta gelişmelere yön vermek insanoğlunun 21. yüzyıldaki en önemli önceliklerinden biri olmuştur. İnternet ve mobil teknolojilerin insan hayatını zaman ve konum fark etmeden etkilemektedir. Özellikle internetin bilgiye ulaşmada sağladığı kolaylık ve özgürlük sayesinde insanlığın son yıllarda oluşturduğu bilgi miktarında da önemli artışlar olmuştur. Bu noktada bilgi toplumu olan ülkeler geleceğe yön vermeye başlamıştır. Tüm bunlara ek olarak katma değeri yüksek teknoloji üretimi yapan ülkeler de diğer ülkeleri kendilerine bağımlı hale getirmişlerdir. Bu noktada ülkemizin de teknoloji tüketen bir toplum durumunda kalmasının sıkıntıları hissedilmektedir. Bu sıkıntıyı aşabilmek adına teknoloji tüketen bir toplum olmak yerine teknoloji üreten bir toplum olabilmek adına yeni nesillerin teknolojiyi karşılaştıkları problemleri çözmek için kullanan bireyler olarak gerekli bilgi ve beceriye sahip olarak yetişmeleri gerekmektedir. İşte tam da burada ülkemizde DENEYAP TÜRKİYE ve benzeri hareketlerin üstlendiği rol önem arz etmektedir.

DENEYAP TÜRKİYE atölyelerinde eğitim alan öğrencilerin dünyada Maker Hareketi veya Do It Yourself (DIY) olarak öne çıkan bakış açısına sahip bireyler olarak ortaokul ve lise yıllarında farklı fikir ve projelerini hayata geçirmeleri sağlanabilmektedir. Bu atölyeler ile geleceğin teknolojilerini üretebilecek gençlere bu hareketi destekleyecek uygun ortam ve alt yapının sağlanması ile teknoloji üreten bir toplumun oluşmasında önemli bir adım atılmıştır. En basit bakış açısı ile konuyu ele aldığımızda; kendi uzaktan kumandalı arabasını yapan çocuklardan evindeki çiçekleri uzaktan sulayabilen çocuklara, okullarda TÜBİTAK tarafından desteklenen Bilim Fuarlarına katılan öğrencilerden Teknofest'e proje hazırlayan öğrencilere ve dahası yüksek teknoloji üreten bireylerin yetişmesi için ülke geneline yayılan bir hareket olarak DENEYAP TÜRKİYE atölyeleri ülkemizin geleceği adına anahtar bir potansiyele sahiptir.

Ülkemizin kalkınması için teknoloji üretme yetkinliği yüksek genç bireyler yetiştirmeyi hedef alan Deneyap Teknoloji Atölyelerinin eğitim modeli kapsamında 36 ay süre ile ücretsiz olarak Tasarım ve Üretim, Robotik ve Kodlama, Elektronik Programlama ve Nesnelerin İnterneti, Yazılım Teknolojileri, İleri Robotik, Nanoteknoloji ve Malzeme Bilimi, Siber Güvenlik, Yapay Zeka, Mobil Uygulama, Enerji Teknolojileri, Havacılık ve Uzay Teknolojileri derslerine ilişkin eğitimler verilmektedir. Verilen eğitimlerden biri de "Elektronik Programlama ve Nesnelerin İnterneti"dir.

Elektronik Programlama ve Nesnelerin İnterneti dersi kapsamında öğrencilere bir elektrik devresindeki gerilim, direnç ve akım ilişkisini gibi basit düzeyde elektrik bilgisinden başlayarak diyot, kondansatör vb devre elemanlarının yanı sıra ultrasonik mesafe sensörü, ısı ve sıcaklık sensörü, gaz sensörü, yağmur sensörü gibi çeşitli sensörleri kullanarak elektronik devrelerin kullanılacağı bir eğitim oluşturulmuştur. Yine bu elektronik devre elemanlarının istenen şekilde çalışabilmeleri için nasıl kodlama yapabilecekleri de bu ders kapsamında öğrencilere verilecektir. Devrelerde elde edilecek verilerin öncelikle farklı gösterge ve ekranlarda sunulması, farklı motorların çalışma ve kullanım alanları, prototip uzay aracı gibi farklı görevleri yerine getirebilecek otonom araçların tasarlanması, çizgi izleyen araçlardan kendi aralarında yarıştıracabilecekleri arabaların oluşturulması ve yine farklı sensörleri kullanarak akıllı ev teknolojilerini üretmek bu derste öğrencilere aktarılacak konular olarak öne çıkmaktadır. Bunlara ek olarak İnternete bağlı sistemler aracılığı ile sensörlerden gelen verilerin anlamlı bir

şekilde kullanılmasını ve uzaktan kontrol edilebilen cihaz ve robotların tasarlanabileceği bir yapı ile Nesnelerin İnterneti kavramı da eğitimin içerisine yerleştirilmiştir. Ders içeriğinde ilerleyen bölümlerde öğrencilere proje hazırlamaları için çeşitli görevler ve yönlendirmeler de yapılmış olup, bu süreçlerini detaylı olarak raporlayabilecekleri “Proje Kayıt Defteri” hazırlamaları için ilgili bölümler de oluşturulmuştur. Eğitim sonunda Proje Yarışması ve Sunumları ile eğitim sonlandırılmıştır.

Bu bağlamda Elektronik Programlama ve Nesnelerin İnterneti eğitimi kapsamında lise öğrencilerine yapacakları teknolojilerde öncelikle mekanik tasarımın oluşturulması; sonrasında elektronik devre elemanları ve bağlantıların yapılması; ve en son olarak da gerekli kodlamaların yapıldığı programların yazılması ile istenen işlevi yerine getirebilecek teknolojik alet, cihaz ve robotların oluşturulması hedeflenmiştir. Öğrencilerin üniversite eğitimlerinden önce bu eğitimi almaları ve kendilerinde gördükleri potansiyel bağlamında mühendislik alanlarına yönelmeleri ve daha verimli bir üniversite eğitim süreci geçirerek teknoloji üreten bir toplum olma yolunda en yüksek azim ve gayretle çalışmaları bu dersin en büyük kazanımı olacaktır.

Öğretim Kılavuzu

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 birlikte 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 Gözle-Uygula-Tasarla-Üret-Değerlendir (GUTÜD) öğrenme döngüsü, etkinlik tabanlı program, spiral öğrenme ve eşli programlamayı içermektedir.

Giriş

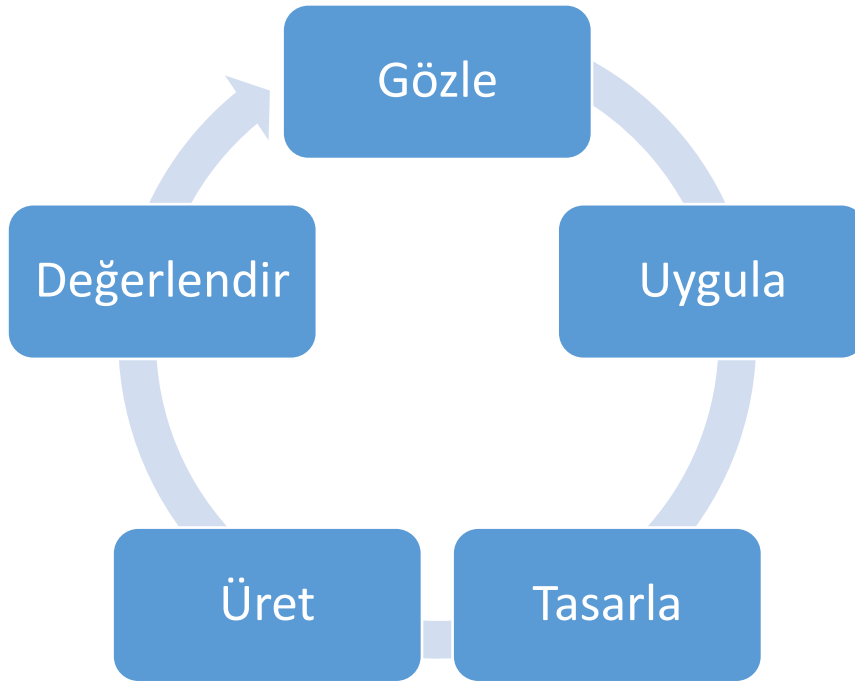
İçinde bulunduğumuz çağda öğrenme, öğrenenlerin sınıf içinde verilen kavram, prosedür ve temel bilgileri istendiği zaman belirli bir ölçüde sınav veya başka vasıtalar ile yeniden göstermesi olarak algılanmamaktadır. Öğrenenlerin öğrendiği kavramı, prosedür ve temel bilgileri başka bağlamlara transfer 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 (2010) 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 eğitmenin anlatımı üzerine kurur.

Eğitmen 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 yaklaşımda 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 pozisyonundadır.

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” adımı öğrencilerden öğrenilen konuları pekiştirmesi, bu konuları basitçe günlük yaşam bağlamında düşünebilmesi ve öğrenme sürecinde yaşadığı deneyimler üzerine düşünerek öğrenme sürecini tamamlamaları beklenmektedir.



Resim 1: GUTÜD Öğrenme/Öğretim Döngüsü

Gözle-Uygula-Tasarla-Üret-Değerlendir Öğrenme Döngüsü

Gözle: Bu bölüm iki kısımdan oluşur. Birinci kısım ders veya konu başlarında gerçekleştirilmelidir. Eğitmen 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 / çalışmalarını kısaca özetleyebileceği gibi günlük yaşamdan ilgi çekici örnekler de kullanabilir, örneğin otomasyona giriş yapmak için El-Cezeri'nin hayatından ve ürettiği cihazlardan bahsedilebilir.

Bu bölümün ikinci kısmı her Gözle etkinliği için geçerlidir. İkinci kısımda eğitmen bir elektronik konusunu uygulamalı olarak (göstererek) anlatır. Gözle etkinliklerinde genel olarak temel hedef doğrudan konunun anlatılması değildir. Burada doğrudan anlatım yapmaktan ziyade eğitmen 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. Eğitmen, 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 işlemi öğrenirken parmaklarını kullanması verilebilir. Eğitmen uygun olduğu durumlarda dışsal ipuçları vermelidir.

Uygula: Bu bölümde öğretmen öğrencilerden Gözle adımıyla 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 eğitmen, öğrencilerden bir trafik ışığını simüle 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 adımıyla öğretmen tarafından öğrencilere bir problem verilir. Öğrencilerden öncelikle bu problemin çözümünü tasarlama istenir. Tasarlama aşamasında öğrenciler temel itibarıyla 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 isteyebilir 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 adımıyla, 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 düşünceleri/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 adımı öğrenci öğrenme sürecinde yaşadığı deneyimler üzerine düşünür. Bu deneyim öğrenme sürecinin tamamlanması, öğrencinin kendini tanıması ve öğrencinin gelecek öğrenmelerini daha verimli hale getirmesi için önemlidir.

Etkinlik Tabanlı Program ve Spiral 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 özümsemiş bütünlüğüne öğrencilere bırakılır. Bu kitapta etkinlik tabanlı program kullanılmıştır. Bu yaklaşımda program etkinlikler çerçevesinde oluşturulur. İlgili konular etkinlikler içerisinde verilir. Bu sayede öğrenciler yaptığı etkinlikte ayrı konulardan ziyade anlam bütünlüğü olan problemlerle uğraşabilir. Öğretmen bir uzman olarak bu problemlere nasıl yaklaştığını öğrencilere gösterebilir. Ayrıca öğrenciler ve öğretmen derslerinde kavramlara ayrı konular üzerinden gitmekten 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

Eşli Programlama

Öğrenciler “Uygula” ve “Üret” adımlarını eşli programlama grupları içerisinde yapacaklardır. Eşli programlamada iki öğrenci, bir elektronik set ve bilgisayar karşısında yan yana oturarak 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” adımı 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 diğeri öğrenci müdahale etmez, ancak çözüm için fikir verebilir veya önerilerde bulunabilir. Görevden sorumlu olan öğrenci etkinliğin kendi bölümünü diğeri 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 diğeri 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 diğeri 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 Arası İletişim

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.

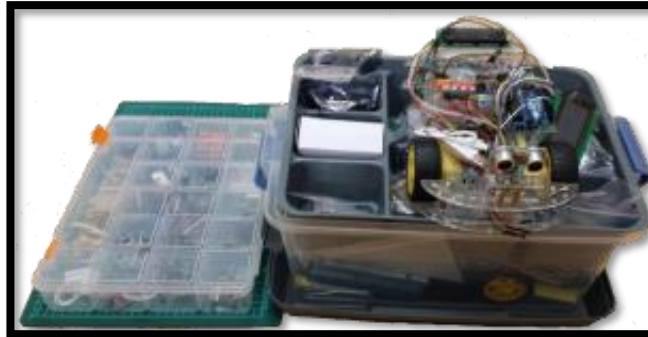
Dönem Projesi

Bu derste öğrencilerin öğrendikleri bilgileri kullanarak özgün bir proje geliştirmeleri beklenmektedir. Projedeki amaç öğrencilerin yaratıcılıklarını teşvik etmek, problem çözme, analitik düşünme ve grup halinde çalışma gibi önemli becerilerini geliştirmelerine yardımcı olmaktır. Öğrenciler proje sürecinde elektronik devreler geliştirebildiklerini görecekler ve bu sayede kendilerine olan özgüvenleri de artacaktır. Proje geliştirme sürecinin, eğitimin son 7 haftasını kapsayacak şekilde planlanması önerilmektedir. Proje ile ilgili detaylı bilgiler “7. Bölüm” ile başlayarak ilgili bölümlerin sonunda ayrı bir başlık altında detaylı olarak açıklanmaktadır. Kısaca ifade etmek gerekirse, grupların ve proje ekiplerinin 7. haftada belirlenmesiyle birlikte öğrencilerin ilk üç proje haftası boyunca (8., 9. ve 10. haftalar) projelerini tanımlama ve tasarlama etkinlikleri üzerinde çalışmaları, 11. haftada belirledikleri problemi çözmek için tanımladıkları ve tasarladıkları çözümü geliştirme (prototip oluşturma) ve 12. ve 13. haftalarda ise geliştirmiş oldukları çözümü test etmeleri ve sürece yönelik proje kayıt defterini teslim etmeleri beklenmektedir.

Bu kapsamda; proje sergisinin yapılacağı son haftaya kadar grupların 7. haftadan başlayarak; “Proje Kayıt Defteri”ni doldurmaları ve ilgili haftalar boyunca yapılacak işlemleri bu deftere adım adım yazmaları istenecektir. Her hafta adım adım doldurulacak proje kayıt defterinde; test etme basamağından önce gruplar daha önceki haftalarda girilmiş olan bilgileri güncelleyebilir ve belirli oranda değiştirilebilirler. Bu yüzden kitabın ilgili bölüm sonlarında proje kayıt defterindeki başlıklar tekrar olacak şekilde verilmiş; bir sonraki haftanın etkinliklerini kapsayacak şekilde ilerleyerek -istendiğinde güncelleme ve değişikliklerin yapılabilmesi için-tasarlanmıştır. Proje kayıt defterinin son ve tamamlanmış versiyonu ise 13. hafta sonunda tüm gruplardan istenecektir. Ayrıca proje sürecinde öğrenciler tüm proje süreçlerini (toplantılar, tasarım çalışmaları, kod geliştirme aşamaları, prototip geliştirme süreçleri, test işlemleri vb.) “Video” kaydına alacaklar ve son hafta proje kayıt defterinde tüm süreçleri aktaran video linki teslim edilmiş olacaktır. Hazırlanan videonun proje sergisinde kullanılması yerinde olacaktır.

Derste Kullanılan Set

Bu ders kapsamında organizeler, 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 2: Derste Kullanılan Set

Laboratuvar Kullanım Yönergesi

Elektronik Programlama ve Nesnelerin İnterneti dersi laboratuvar ortamında etkinlik bazı gerçekleştirilecek bir derstir. Ders boyunca havya gibi yaralanmalara sebebiyet verme ihtimali olan ve transistör 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. Öğretmen ilk derste ve yeri geldikçe öğrencilere düzenli ve özenli çalışmalarını gerektiği konusunda uyarılarda bulunmalıdır. Öğretmen uyarılarda bulunurken çalıştığı öğrenci grubunun psikolojik alt yapısını dikkate almalıdır. Uyarılar öğretmen tarafından itici veya sıkıcı kurallar olarak değil laboratuvarın sağlıklı ve verimli kullanımı için gerekli prosedürler olarak aksettirilmelidir.

Öğrenciler laboratuvarında devrelerini kesme matları üzerinde gerçekleştireceklerdir. Her grup kesme matı, küçük organizör kutu ve bilgisayarını düzenli bir şekilde masasına yerleştirmelidir. Öğrenciler büyük organizör kutularını masaya getirmemelidir. O hafta için gerekli olan devre elemanları öğretim kılavuzunun ilgili giriş bölümünde verilmiştir. Öğrenciler ders başında devre elemanlarını büyük organizör kutudan alıp küçük organizör kutuya aktarmalıdır ve masaya sadece küçük organizör kutuyu götürmelidir. İhtiyaç duyulduğunda büyük organizör 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 çalışırken büyük organizör kutusunun kapağı kullanabilir. Bu parçalar kapak üzerinde kullanılırsa düşüp kaybolma ihtimalleri azalır.

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

Sıkça Sorulan Pedagojik Sorular

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

Cevap: 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 buldukları etkinliklerdir. “Tasarla ve Üret” etkinliklerinde öğretmenler rehber pozisyonundadırlar. Öğretmen 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 uygulamaya mı?

Cevap: “Gözle ve Uygula” etkinliklerinde öğretmen bir görev üzerinden öğrencilere bir uzmanın görevdeki amacı nasıl yerine getirdiğini uygulamalı olarak gösterir. Yani 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), eğitmen konuları doğrudan da anlatabilir. Tüm bunlar göz önüne alındığında eğitmenin “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 aktarmalarına 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” ve “ilave etkinlikleri” bitiremiyorum. Öğrencilere yardımcı olup onları hızlandırsam nasıl olur?

Cevap: Eğitmen bütün “Tasarla ve Üret” 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 eğitmen 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 üretirken 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 eğitmen 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. 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 eğitmenleri de zorlayacak seviyede problemler bulunmaktadır. Eğitmen 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.

1. Bölüm - Elektronik Programlamaya Giriş

Ön bilgi:

- Öğrenciler algoritma mantığını bilir.
- Öğrenciler temel programlama kavramlarını bilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.

Bölüm Kazanımları:

- Öğrenciler gerilim tanımını yapabilir.
- Öğrenciler akım tanımını yapabilir.
- Öğrenciler direnç tanımını yapabilir.
- Öğrenciler dirençin çalışma prensiplerini bilir.
- Öğrenciler akım ve direnç arasındaki ilişkiyi açıklayabilir.
- Öğrenciler temel elektronik devre elemanlarının breadboard bağlantılarını yapabilir.
- Öğrenciler Arduino mikrodenetleyici kartını ve üzerindeki bileşenleri bilir.
- Öğrenciler Arduino IDE yazılımını ve arayüzünü kullanabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin Arduino UNO mikrodenetleyicisini kullanarak direnç ve LED gibi temel elektronik devre elemanlarının çalışma prensipleri anlamaları ve breadboard kullanılarak bu elemanların bağlantılarını yapabilmelerini ve Arduino IDE yazılımı ile bu devrenin istenen şekilde programlanabilmesini 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:

Malzeme Listesi	
Arduino UNO	Breadboard
9 V pil	Kırmızı LED
220 ohm direnç	Mavi LED
330 ohm direnç	Sarı LED
560 ohm direnç	
1 K ohm direnç	

Haftanın İşlenişi:

Gözle: Arduino UNO ve yazılım arayüzü olan Arduino IDE ile seçilen temel elektronik devre elemanlarının tanımlanması, kullanılması ve akım-gerilim gibi temel kavramlar ile ilişkisinin örnekler üzerinde incelenmesi.

Uygula: Arduino UNO ve yazılım arayüzü olan Arduino IDE ile seçilen temel elektronik devre elemanlarının kullanıldığı örnek devrelerin oluşturulması ve programlanması.

Tasarla: Devre tasarımı öncesinde Arduino UNO üzerinde ilgili problemi çözmek için gerekli devrenin şematik olarak hazırlanması ve devreyi uygun şekilde çalıştıracak algoritmanın tasarlanması.

Üret: Arduino UNO mikrodenetleyicisinin dijital çıkış pinlerinin devre tasarımlarının yapılması. Farklı LED uygulamalarının Arduino IDE kullanılarak kodlanması.

Değerlendir: Yansıtma Etkinliği.

1. ADIM: 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 izlenen televizyonlar, oyun oynanılan tablet bilgisayarlar ve cep telefonları çalışmak için elektriğe ihtiyaç duyarlar. Elektrik bir enerji türüdür. Bahsedilen bu araçlar pil, batarya ve duvar prizlerinden alınan şebeke elektriği gibi kaynaklar aracılığı ile elektrik enerjisini kullanırlar. 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. Sıklıkla evlerimizde kullandığımız AAA kalem pillerde iki uçtan biri 0 V diğer ise 1.5 V’luk potansiyeli temsil eder. Devrede bu pilin uçları bağlandığında 1.5 V’luk bir gerilim oluşur.

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. Akım birimi **Amperdir**.

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. Eğitimci bu bilgiyi mutlaka vurgulamalıdır.

Elektronlar çok küçük oldukları için gözle görülemezler. Elektronların akışı bir benzetme ile daha iyi açıklanabilir. Bunun için öğrencilerden alt kısımlarına 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 veya elektronların kablo içerisinde ittirilmesini sağlayan potansiyel enerjiye gerilim denilir. Elektronların kablo içerisinde akması ile oluşan elektrik olayına ise akım denilir. Gerilim, yani elektronlar üzerine uygulanan itirme gücü 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.

Ohm Yasası: Akım, gerilim ve direnç arasındaki bağlantıdır ve direnç ile akım ters orantılıdır. Devredeki gerilim sabitken direnç arttıkça, devredeki akım azalır. Direnç sabitken, gerilim ile

akım ise doğru orantılıdır. Yani gerilim artarsa akım artar. Yukarıdaki su kovası benzetmesine dönecek olursak su yüksekliği aynı olan iki kova düşünelim (gerilimleri aynı). Kovalardan birindeki deliği daha çok kapatırsak (direnç artarsa) bu kovadaki suyun akış miktarı (akım) diğerine oranla azalır.

$$V = I \times R$$

V: Gerilim (Volt ile ölçülür)

I: Akım (Amper ile ölçülür)

R: Direnç (Ohm ile ölçülür)

1.2 Uygula- Elektrik Devrelerinde Elektrik, Akım, Gerilim ve Direnç

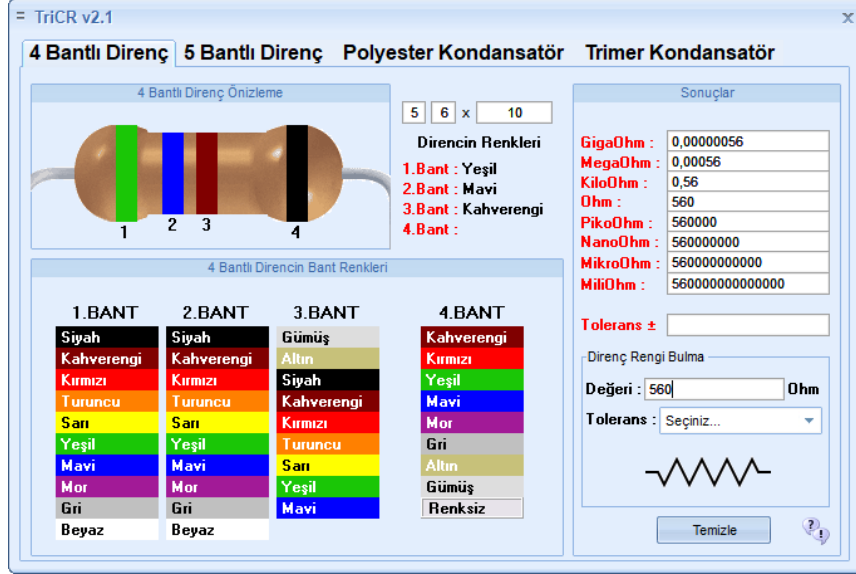
Eğitmen öğrencilere aşağıdaki soruları sorar ve öğrencilerin bu soruların cevabını sınıfça tartışmalarını sağlar.

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

1.3 Gözle ve Uygula - Basit Bir Devre Oluşturma (Öğrenci 1)

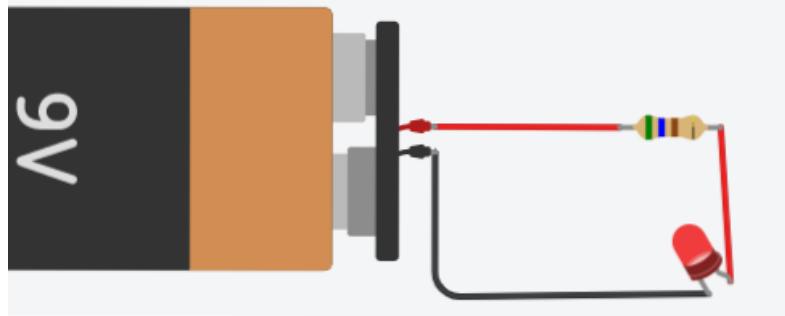
Malzeme Listesi
9 V pil
Bağlantı kabloları
330 ohm direnç
560 ohm direnç
1 K ohm direnç

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 zarar görmemesi için direnç kullanılır. Bu yüzden ilk devrede 560ohm direnç kullanılacaktır. Direnç değerleri direnç üzerinde bulunan renkler aracılığıyla belirlenir. Bu derste direnç okuması öğretilmeyecektir. Direnç okuma etkinlikleri bir sonraki ders içeriğinde daha ayrıntılı yer alacaktır. Bu derste direnç renk kodları TriCR isimli uygulama ile belirlenecektir (Cep telefonları için farklı uygulamalar da mevcut, "Direnç Hesaplayıcısı", "Direnç Oku" gibi). Direnci seçmek için öncelikle TriCR uygulaması açılır. Uygulamadaki "4 Bantlı Direnç" sekmesi seçilir ve "Değeri" hücrelerine 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.



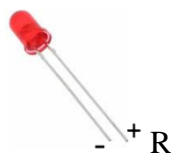
Resim 1.1: TriCR Programı Arayüzü

Öğrenciler bu renkleri kullanarak devre için gerekli direnci seçer. Bu devre oluşturulurken breadbord kullanılmayacaktır. Devre şeması aşağıdaki resimde verilmiştir. Devre aşağıdaki resimde gösterildiği şekilde oluşturulup öğrencilere gösterilmelidir.



Resim 1.2: Gözle-Uygula Örnek Devre Şeması

Devreyi kurarken eğitmen öğrencilere doğru akım güç kaynaklarının bir artı ve bir eksi ucu olduğundan bahseder. Güç kaynağından gelen artı uç ve artı uca bağlanan devre elemanlarında genellikle **kırmızı** renk kablolar 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 kablolar da genellikle kahverengi ve siyah tercih edilir. Bu da bir zorunluluk değildir. Bunun ardından eğitmen dirençlerin artı veya eksi ucunun olmadığından ve yönü fark etmeden istenilen şekilde devreye takılabileceğinden bahseder. LED ve diğer birçok devre elemanı için bu durum geçerli değildir. Örneğin 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 1.3: LED Ayak Yapısı

Eğitmen devreyi kurduktan sonra devreye güç vererek çalıştırır ve çalışan devreyi öğrencilere gösterir. Bunu takiben eğitmen öğrencilerden aynı devreyi farklı dirençlerle (330 ohm, 560 ohm ve 1K'lık dirençlerle) kurup çalıştırmalarını ister. Farklı dirençlerle oluşturulan her devreyi gözlemleyerek farklılıklara dikkat etmelerini ister.

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

- Bu devrede gerilim arttırılırsa LED'in verdiği ışık miktarı nasıl değişir? Sebepleriyle açıklayınız.
- Bu devrede yer alan direncin büyüklüğü artırıldığında LED'in verdiği ışık miktarı nasıl değişir? Sebepleriyle açıklayınız.
- Çevrenizde pille çalışan cihazlar var mı? Bu cihazların kaçar adet pile ihtiyaç duyduklarını biliyor musunuz? Bu cihazlarda farklı sayıda pile neden ihtiyaç duyulur?

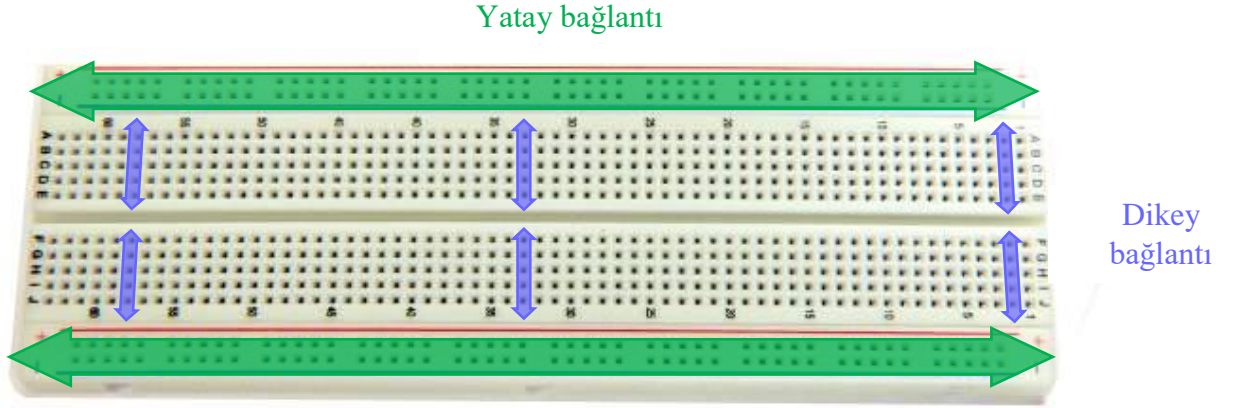
1.4 Gözle ve Uygula - Breadboard ve Arduino Uno ile Basit Bir Devre Oluşturma (Öğrenci 2)

Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
560 ohm direnç
Kırmızı LED

Dikkat

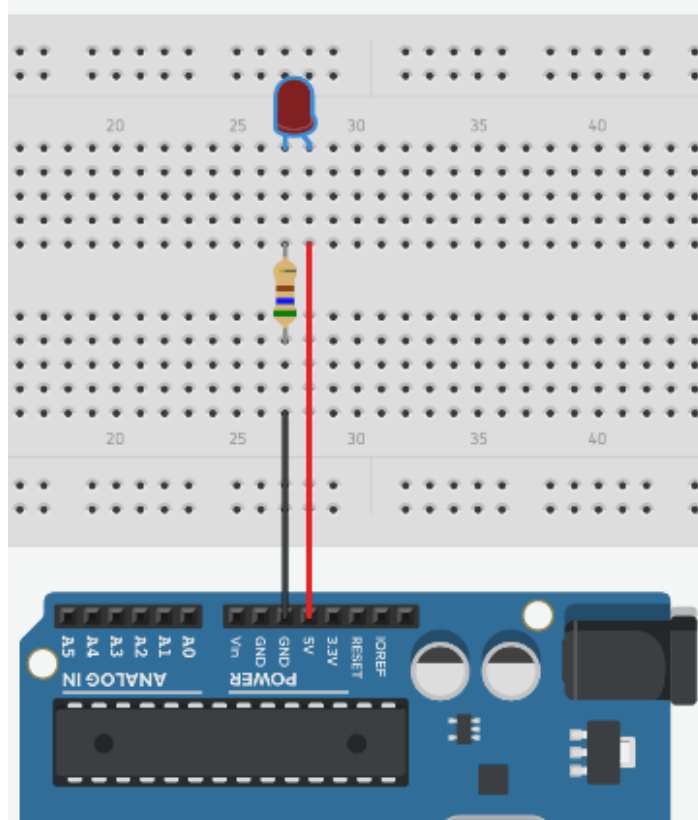
Eğitmen öğrencileri bu ders boyunca direnç olmadan LED kullanmamaları gerektiği konusunda uyarır.

Arduino Uno aslında programlanabilir bir işlemci, giriş-çıkış pinleri 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 (devre tahtası) 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. Eğitmen 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 1.4: Breadboard Bağlantıları

Breadboard yardımıyla aşağıdaki resimde görülen devre kurulumu ve bilgisayardan USB kablosu aracılığıyla güç verilerek Arduino UNO çalıştırılır.



Resim 1.5: Gözle-Uygula Örnek Devre Şeması

Burada vurgulanması gereken nokta LED'in "anot", "katot" bağlantılarının yukarıda açıklandığı gibi yapılmasıdır. Sonrasında bağlantı kablolarından biri, bir ucu kart üzerindeki 5 V yuvasına diğer ucu da LED'in artı bacağına (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. Eğitimci öğrencilere Arduino'nun GND (-)

bacağından akımın akarak direnç ve LED üzerinden geçtiği ve ardından 5 V (+) bacağına gelerek devreyi tamamladığı anlatılır. Fakat devreler incelenirken akım 5 V (+) bacağından GND (-) bacağına hareket etmektedir.

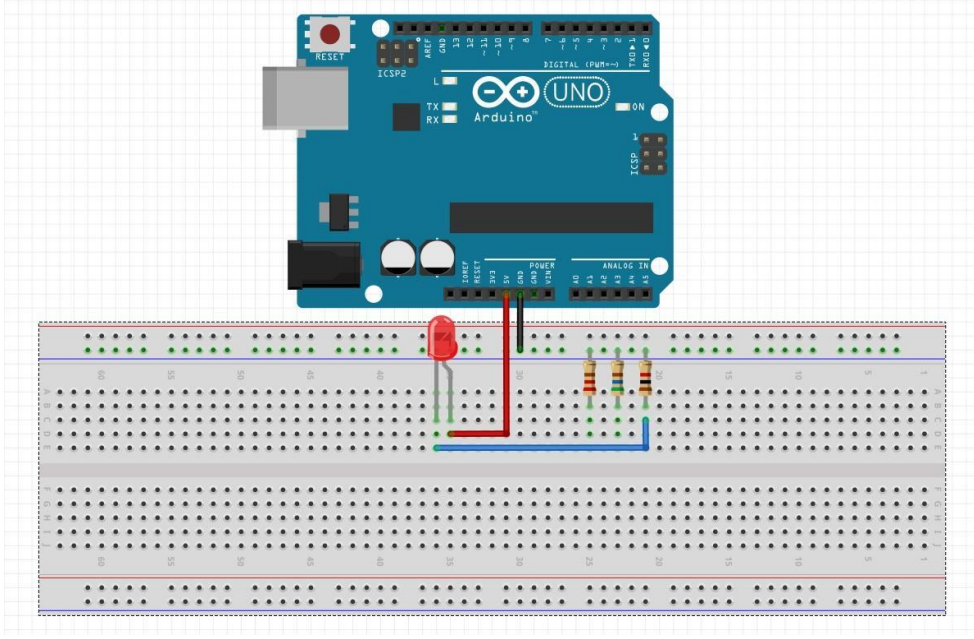
Eğitmen öğrencilerden yukarıdaki resimde gösterilen devreyi kurmalarını ister. Eğitmen öğrencilere devre kurma konusunda gerektiğinde yardımcı olur. Bütün öğrenciler devreyi kurduktan sonra öğrencilere aşağıdaki soruları sorar ve öğrencilerin soruyu sınıfça tartışmalarını sağlar. Eğer öğrencilerden doğru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

- i. Devrede neden bir direnç kullanılmıştır?
- ii. Devrede direnç kullanılmazaydı ne gibi sorunlar ile karşılaşılabilirdi?
- iii. ABD’den alınan bir elektronik cihazların adaptörünün Türkiye’de kullanılabilmesi için nelere dikkat etmeliyiz? Ohm yasasından faydalanarak bunun sebebini açıklayınız. (Türkiye’de 220 V kullanılırken, Amerika’da 110 V kullanılmaktadır)

1.5 Uygula- Basit Bir Devre Oluşturma ve Direnç Değerlerini Değiştirme (Öğrenci 1)

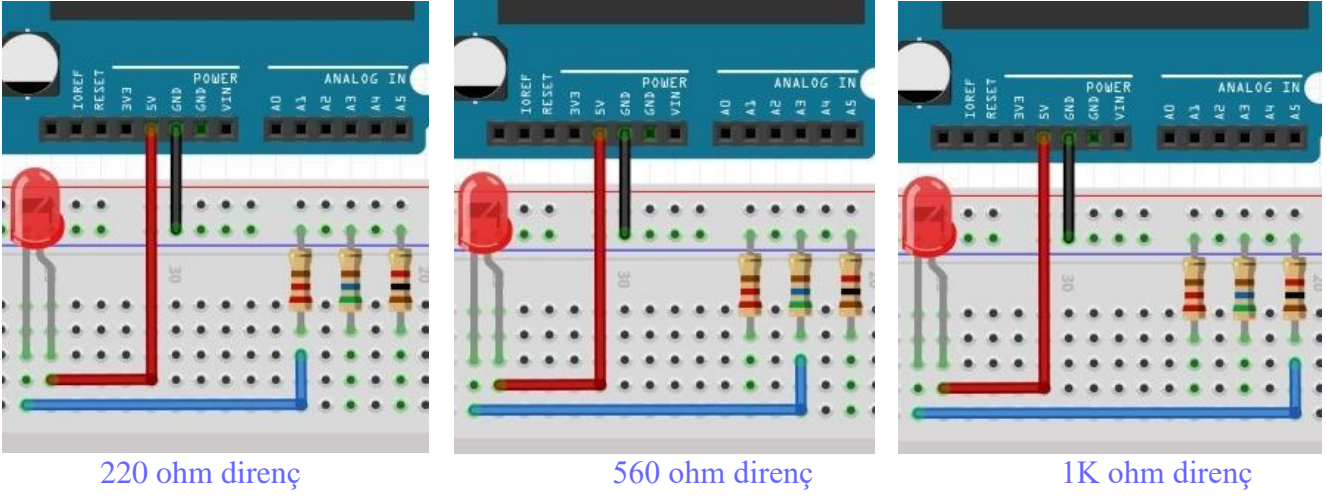
Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
220 ohm direnç
560 ohm direnç
1K ohm direnç
Kırmızı LED

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. Eğitmen öğrencilerden bu işlemi gerçekleştiren basit bir devre kurmasını ister. Eğitmen gerekirse öğrencilere farklı değerli dirençleri kullanarak devreden geçen akım miktarını ve dolayısı ile LED’in parlaklık miktarını ayarlayabilecekleri hakkında ipucu verebilir. Fakat eğitmen öncelikle öğrencilerin çözümü bulmak için düşünmelerine fırsat vermelidir. Hemen ipucunu söyleyerek öğrencilerin düşünme sürecini azaltmamalıdır. Eğitmen ancak öğrencilerden çok zorlananlar olursa ipucu verebilir. Aşağıdaki resimde görülen devre şeması bu etkinlik için örnek bir çözümdür. Belirtilen dirençleri belirlemek için “TriCR” uygulaması kullanılır.



Resim 1.6: Uygula Etkinliđi Örnek Devre Şeması

Devredeki LED odadaki lambayı temsil etmektedir. Üç farklı direncin birer uçları breadboardun alt kısmındaki yatay bağlantı kullanılarak Ardıuno 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. Eğitmenin burada vurgulaması gereken nokta devre tahtasının yatay hattını kullanarak GND'nin çoğaltılmasıdır. Ardıuno kartının güç bağlantısı yapıldığında LED yanar. Devre akımı 1 Kohm değerindeki direnç ve LED üzerinden geçerek tamamlanır. Sonra, en sağda yer alan 1 Kohm 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 1 Kohm değerindeki direnç yerine, 560 ohm değerindeki direnç üzerinden tamamlanmış olur. Devre şeması aşağıdaki resimde görülmektedir.



Resim 1.7: Uygula Etkinliği Örnek Devre Şeması

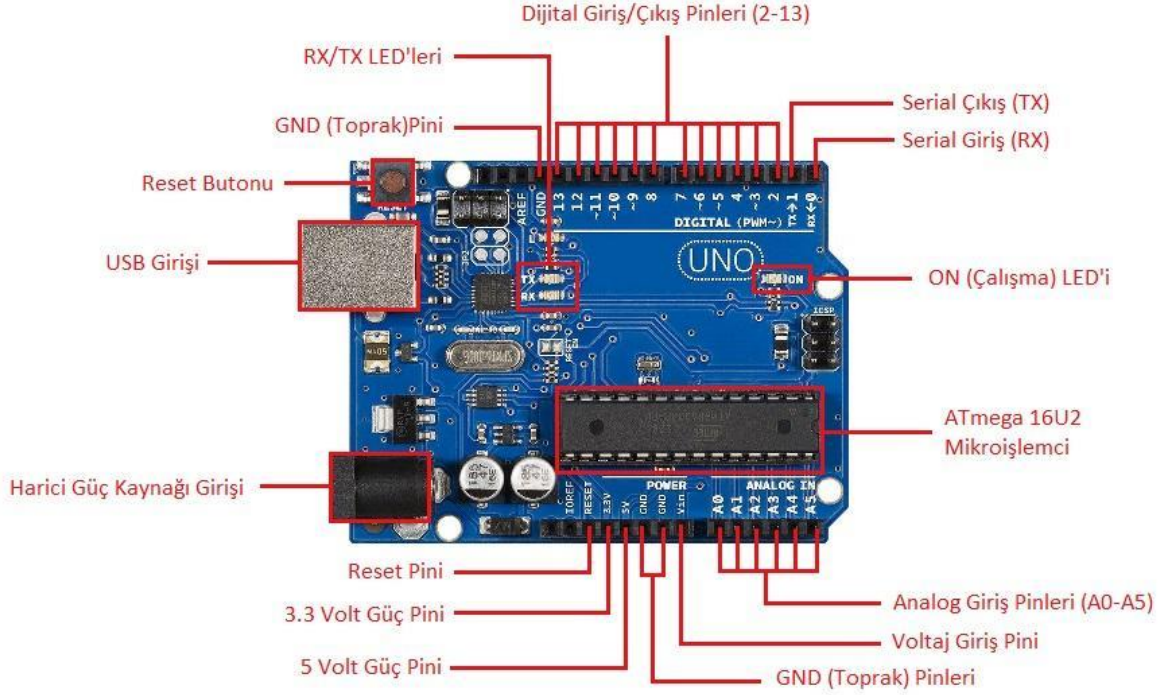
Karta güç yeniden verildikten sonra yanan LED'in parlaklığı incelenir. Devre 220 ohm değerindeki direnç yerine 560 ohm değerindeki direnç üzerinden tamamlanır. Daha sonra aynı işlemler tekrarlanarak 1K ohm değerindeki direncin bağlantısı yapılır ve parlaklık kontrol edilir.

1.6 Gözle - Arduino UNO ve Arduino IDE 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. Mikrodenetleyici; kendi işlemcisi, belleği ve giriş-çıkış pinlerine sahip olan programlanabilir kartlara verilen genel bir addır. Arduino aslında donanım ve yazılımın entegre şekilde kolayca kullanılmasına imkan veren açık kaynak kod temelli ve ATmega mikroişlemcisine sahip bir mikrodenetleyicidir. Arduino; üzerine dijital ve analog pinler aracılığıyla sensör, motor ve diğer elektronik devre elemanlarını bağlayıp programlayarak çevresi ile iletişime geçebilen fiziksel programlama platformudur. Arduinonun farklı amaçlar için üretilen farklı versiyonları vardır. Bunlardan bazıları:

- Arduino Uno
- Arduino Mega
- Arduino Lilypad'dir.

Bu ders kapsamında biz Arduino UNO üzerinde uygulamalar gerçekleştireceğiz. Aşağıdaki resimde Arduino'nun yapısı görülmektedir.



Resim 1.8: Arduino'nun Yapısı

Arduino UNO aslında farklı güç kaynağı girişlerine sahiptir. Bunlar USB port, harici güç kaynağı girişi ve kart üzerindeki pinlerdir. Bu ders kapsamında Arduino UNO'ya USB port üzerinden bilgisayara bağlayarak güç verilecektir. USB portu aynı zamanda Arduino UNO'nun bilgisayarla iletişim kurmasını ve yazdığımız kodları aktarmamızı sağlar. Bu haberleşmeyi yine Arduino UNO üzerindeki "ATmega 16U2" mikroişlemci gerçekleştirir. Arduino UNO üzerinde bir de "TX, RX" LED'leri bulunmaktadır. Bu LED'ler Arduino UNO ile bilgisayar arasında veri alışverişi olup olmadığını gösterir. "TX", veri gönderirken; "RX" ise veri alırken yanar. Ayrıca bu iki LED'in üzerinde "L" LED'i bulunur. Bu LED aslında Arduino UNO'nun 13. pinine bağlıdır ve bu pin üzerinde yazılan kodlarla kontrol edebildiğimiz bir LED'dir. Ayrıca Arduino UNO üzerinde kartın çalışıp çalışmadığını gösteren "ON" LED'i bulunmaktadır. Arduino UNO'nun bir diğer bileşeni de reset butonudur. Bu buton Arduino UNO üzerine aktardığımız kodu en baştan başlatmamızı sağlar.

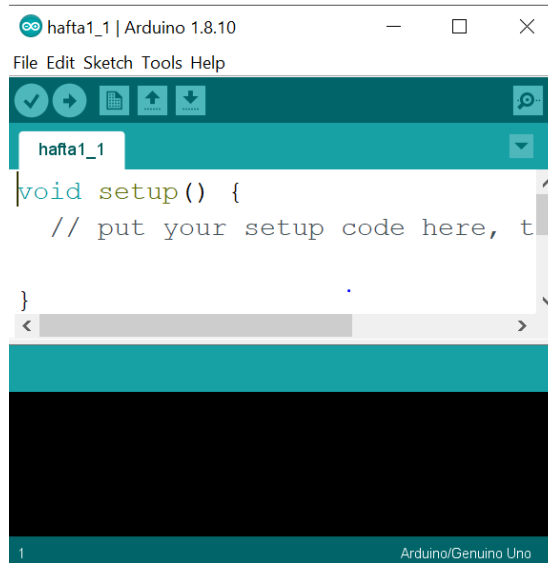
Arduino'nun mikrodenetleyici özelliklerini kullanmak için bir programlama diline ihtiyaç vardır. Arduino'nun kendine ait bir yazılımı vardır. Elektronik Programlama ve Nesnelerin İnterneti dersinde bu amaç doğrultusunda **ArduinoIDE** kullanılacaktır. ArduinoIDE metin tabanlı ve C benzeri bir kod dizimi (syntax) sahip olan bir Arduino geliştirme ortamıdır. Eğitimci ArduinoIDE geliştirme ortamının nasıl kullanılacağını projeksiyon ile yansıtarak aşağıdaki resimde öğrencilere adım adım anlatır:

- i. Bir web tarayıcısı yardımıyla <https://www.arduino.cc/> adresi açılır. Burada **Software** menüsünden **Downloads**'a tıklanır.



Resim 1.9: Arduino IDE Web Sayfası

- ii. ArduinoIDE geliştirme ortamını kullanabilmek için öncelikle ilgili program indirilip kurulmalıdır. Bu süreç eğitmen tarafından öğrencilere gösterilmelidir ama eğitmen derslerden önce ArduinoIDE’yi atölyelerdeki bilgisayarlara kurmakla görevlidir. Aşağıdaki resimde Arduino IDE arayüzünün görüntüsü verilmiştir.



Resim 1.10: Arduino IDE Arayüzü

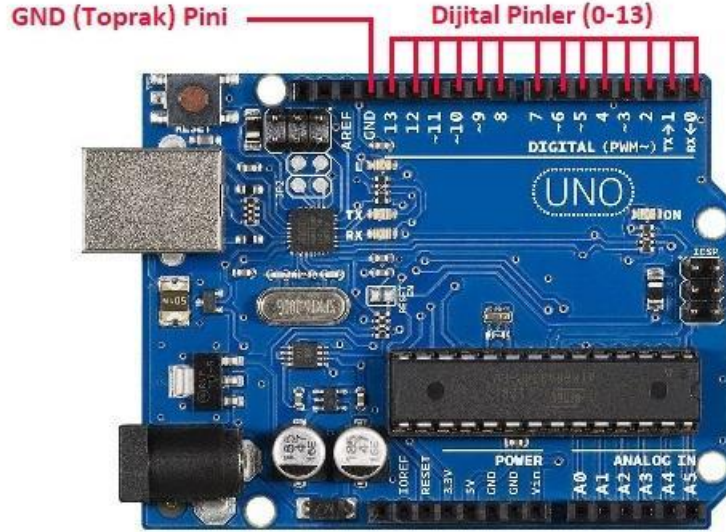
Dikkat

“Arduino Web Editörü” hakkında eğitmen tarafından üyelik ve kullanım bilgileri öğrencilere gösterilir.

1.7 Gözle - Uygula Arduino UNO ile LED Yakıp Söndürme (Öğrenci 1)

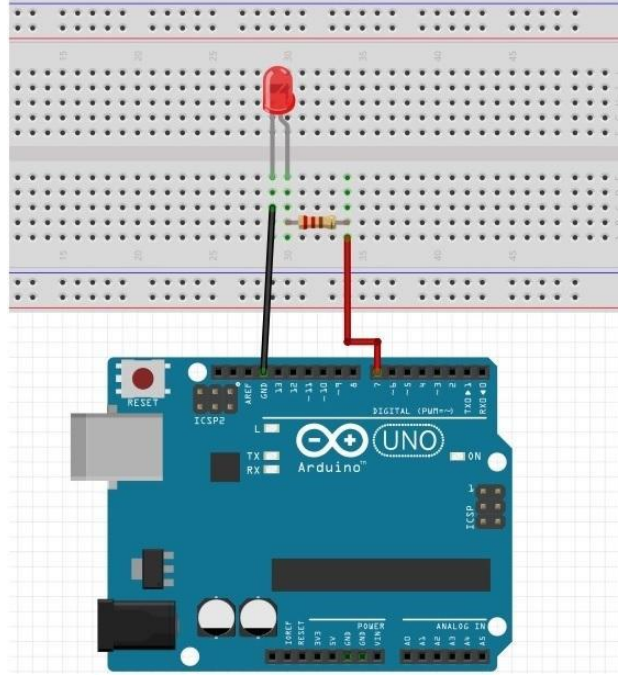
Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
220 ohm direnç
Kırmızı LED

Bu etkinlikteki amaç Arduino IDE yazılımı ile periyodik olarak yanıp sönen LED etkinliği yapmaktır. Bu etkinlikte öncelikle aşağıdaki resimde görülen Arduino'nun dijital pinleri tanıtılmalıdır. Bu pinler yardımı ile hem dijital girdi alınabileceği hem de dijital çıktı verilebileceği vurgulanır. Bu pinler ile devre elemanlarına "5 V" (ON-HIGH) veya "0 V" (OFF-LOW) gönderilebilir. Bu pinler aynı zamanda dışarıdan gelen "5 V" veya "0 V" 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.



Resim 1.11: Arduino UNO Dijital Pinleri

Eğitmen devrenin nasıl kurulduğunu göstererek öğrencilerin aşağıdaki resimde görülen devreyi kurmasını sağlar. Öğrencilere Arduino IDE'yi kullanarak Arduino UNO kartının 7. dijital pininden çıktı alınacağı ve alınan akımın GND üzerinden devreyi tamamlayacağı anlatılır.



Resim 1.12: Gözle-Uygula Etkinliği Örnek Devre Şeması

Arduino IDE geliştirme ortamında aşağıdaki resimde görülen program yazılır ve programın nasıl yazıldığı projeksiyondan yansıtılarak öğrencilere gösterilir.

```

led_yakma_sondurma | Arduino 1.8.10
File Edit Sketch Tools Help
led_yakma_sondurma §
1 void setup() {
2 pinMode(7, OUTPUT);
3 }
4
5 void loop() {
6 digitalWrite(7,HIGH);
7 delay(1000);
8 digitalWrite(7,LOW);
9 delay(1000);
10 }
11
Done Saving

```

Resim 1.13: Gözle-Uygula Etkinliği Örnek Kod

Arduino UNO kartı bilgisayarın USB portuna kablo aracılığı ile bağlanır. Arduino IDE ortamında yazılan program Arduino UNO'ya yüklenir ve bu yüklenme süreci öğrencilere adım adım gösterilir.

Dikkat

Bağlantı noktasının yanlış seçilmesi durumunda kodlar Arduino'ya yüklenemeyecektir. Arduino IDE ile yazılan kodu Arduino UNO'ya atabilmek için öncelikle Araçlardan Arduino UNO'nun bağlı olduğu Port girilmelidir.

Yukarıdaki LED yakıp söndürme örneği için yazılan programda görüldüğü gibi Arduino IDE'yi ilk açtığımızda bizi **iki fonksiyon (setup ve loop)** karşılar. Programlama dillerinde fonksiyonlar genellikle geliştiriciler tarafından kod yazma sürecini kolaylaştırmak için kullanılır. Arduino IDE'de hazırlanan program Arduino UNO kartına yüklendiğinde ilk çalışan fonksiyon setup fonksiyonudur. Setup fonksiyonu sadece bir defa çalıştırılır ve bu fonksiyon içerisinde genellikle tanımlamalar yapılır (pinler ve pinlerin çalışma modları vb.). Loop fonksiyonu ana fonksiyon olup içerisindeki komutlar dışarıdan müdahale olmadığı sürece sonsuz bir döngü içindedir ve Arduino UNO çalıştığı sürece tekrar eder.

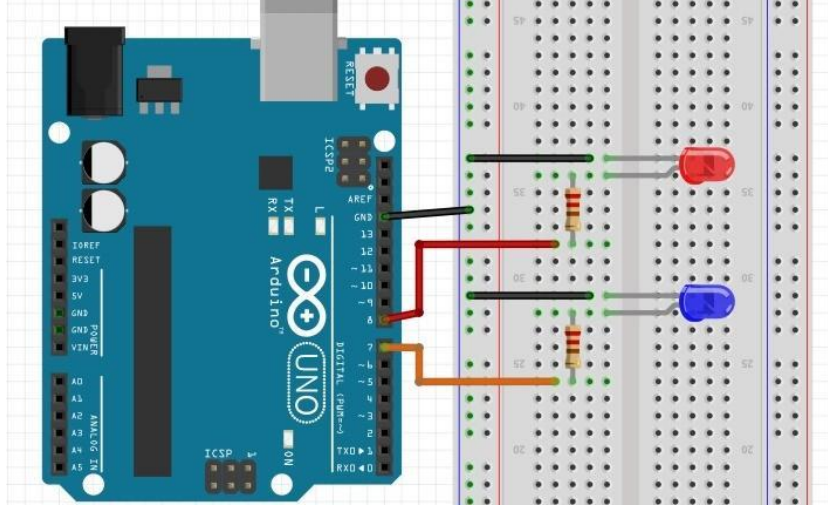
Eğitmen tarafından öğrencilerin yukarıdaki resimde görülen programı yazıp çalıştırmaları sağlanır. Bunun ardından öğrencilere bu programın sadece 7. dijital pin'e bağlı olan bir LED'i yakıp söndürdüğü söylenir. Bu etkinlikteki amaç bir LED'i periyodik olarak yakıp söndüren bir program yazmaktır. Bütün öğrenciler Arduino IDE'ye programı yazdıktan sonra öğrencilere aşağıdaki soruları sorar ve öğrencilerin soruyu sınıfça tartışmalarını sağlar. Eğer öğrencilerden doğru yanıtlar gelmezse eğitmen soruların yanıtını verir. Fakat ilk etapta soruların yanıtını vermemelidir.

- i. Sizce Arduino IDE'nin farklı bir pinini çıkış olarak kullanmak istersek programda ne tür bir değişiklik yapılması gerekir?
- ii. Sizce yazılan programda LED'in yanıp sönmeye süresini nasıl değiştirebiliriz?

1.8 Uygula- Flip Flop (Öğrenci 2)

Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
2 adet 220 ohm direnç
1 adet Kırmızı LED
1 adet Mavi LED

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 ve bu işlem sürekli tekrar etmelidir. Eğitmen öğrencilerin bu uygulamayı yapmasını sağlar. Bu görev için örnek çözüm aşağıdaki resimlerde verilmiştir.



Resim 1.14: Uygula Etkinliği Örnek Devre Şeması

```

File Edit Sketch Tools Help
flip_flop_LED
1 void setup() {
2   pinMode(7, OUTPUT);
3   pinMode(8, OUTPUT);
4 }
5
6 void loop() {
7
8   digitalWrite(7, HIGH);
9   digitalWrite(8, LOW);
10  delay(1000);
11
12  digitalWrite(7, LOW);
13  digitalWrite(8, HIGH);
14  delay(1000);
15 }

```

Resim 1.15: Uygula Etkinliği Örnek Kod

Dikkat

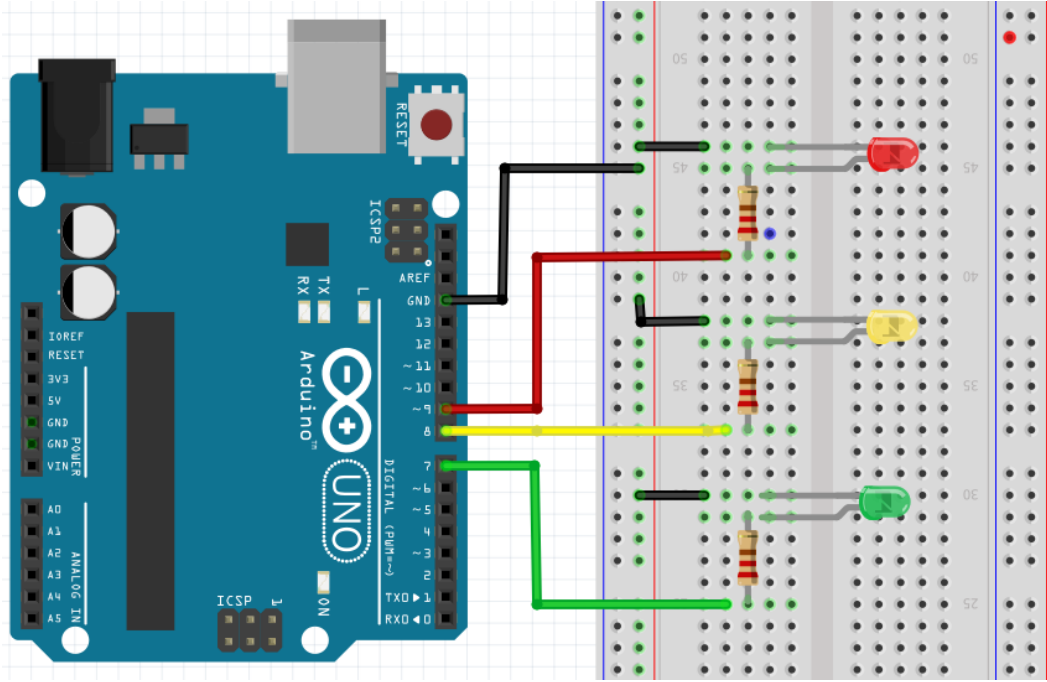
Arduino üzerinde bulunan pinleri çıkış veya giriş amacıyla kullanılabilir. Programın en başında (Setup içerisinde), kullanılacak pinler ayarlanmalıdır.
pinMode(7,OUTPUT);

Bu etkinlikte LED'ler direnç kullanmadan pin'lere bağlanırsa LED'ler patlayabilir. LED bağlantısı için mutlaka 220 ohm (veya yeterli büyüklükte) dirençler kullanılmalıdır.

1.9 Uygula - Trafik Işığı (Öğrenci 1)

Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
3 adet 220 ohm direnç
1 adet Kırmızı LED
1 adet Sarı LED
1 adet Yeşil LED

Bu etkinlikte **kırmızı**, **sarı**, **yeşil** LED'ler kullanılarak trafik lambasının bir örneğini yapmak hedeflenmektedir. **Kırmızı ışık** 4 saniye, **sarı ışık** 1 saniye, **yeşil ışık** 4, tekrar **sarı ışık** 1 saniye yanmalıdır. Bu süreler trafik ışıklarında kullanılan sürelerden bağımsız olarak sadece bu etkinlik için belirlenmiştir. Eğitimci öğrencilerin bu uygulamayı yapmasını sağlar. Bu görev için örnek bir çözüm aşağıdaki resimlerde verilmiştir.



Resim 1.16: Uygula Etkinliği Örnek Devre Şeması


```

kirmizi_isik | Arduino 1.8.10
File Edit Sketch Tools Help
kirmizi_isik $
1 #define kirmizi_led 9
2 #define sari_led 8
3 #define yesil_led 7
4
5 void setup() {
6   pinMode(kirmizi_led,OUTPUT);
7   pinMode(sari_led,OUTPUT);
8   pinMode(yesil_led,OUTPUT);
9 }
10
11 void loop() {
12   digitalWrite(kirmizi_led,HIGH);
13   digitalWrite(sari_led,LOW);
14   digitalWrite(yesil_led,LOW);
15   delay(4000);
16   digitalWrite(kirmizi_led,LOW);
17   digitalWrite(sari_led,HIGH);
18   digitalWrite(yesil_led,LOW);
19   delay(1000);
20   digitalWrite(kirmizi_led,LOW);
21   digitalWrite(sari_led,LOW);
22   digitalWrite(yesil_led,HIGH);
23   delay(4000);
24   digitalWrite(kirmizi_led,LOW);
25   digitalWrite(sari_led,HIGH);
26   digitalWrite(yesil_led,LOW);
27   delay(1000);
28 }
Done compiling.
Sketch uses 1038 bytes (3%) of program storage space

```

Resim 1.17: Uygula Etkinliđi Örnek Kod

Dikkat

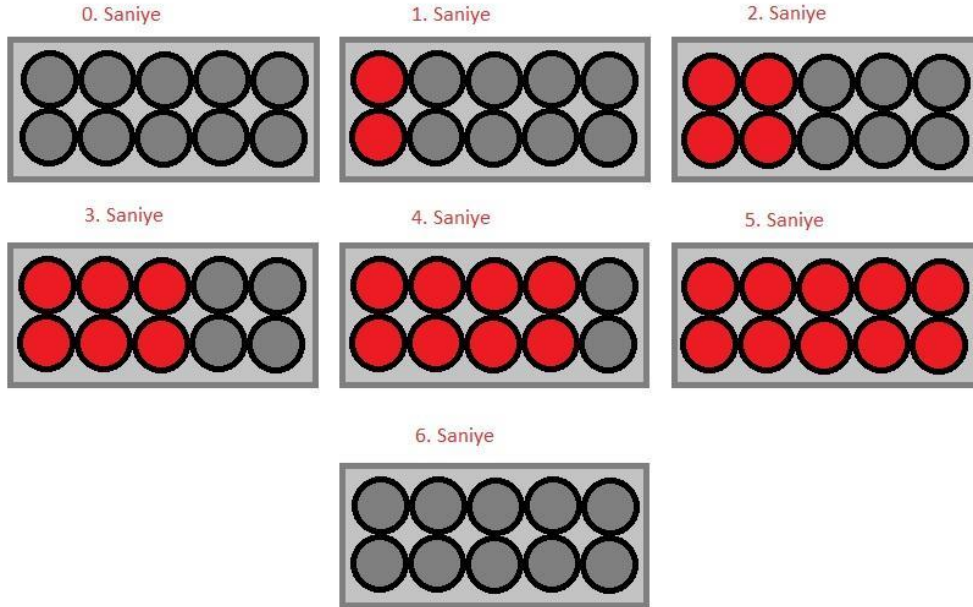
Arduino’da özellikle fazla sayıda “pin” kullanıldığında ve çok fazla kod satırından oluşan programlarda hangi “pin”e hangi elektronik devre elamanının bağlandığını kod satırları yazılırken daha kolay hatırlanması ve kullanılabilmesi için pinlere bağlanan devre elemanlarına uygun olarak isimlendirme (tanımlama) yapılır. Arduino IDE’de bu tanımlama “#define” komutu ile yapılır. Bu tanımlama “setup” ve “loop” fonksiyonlarının dışında yapılabilir. Satıra #define komutu ile başlandıktan sonra pine verilecek isim yazılır ve sonrasında “pin” numarası yazılarak (sonunda “;” kullanmadan) alt satıra geçilir. Eğitmen yukarıdaki kod bloğunda #define komutunun kullanımı ve işlevi hakkında öğrencileri bilgilendirir.

Ayrıca, eğitmen bu tanımlama işleminin daha sonraki derslerde detaylı olarak anlatılacak değişken atama yöntemi (int) ile de yapılabileceğini öğrencilere açıklar.

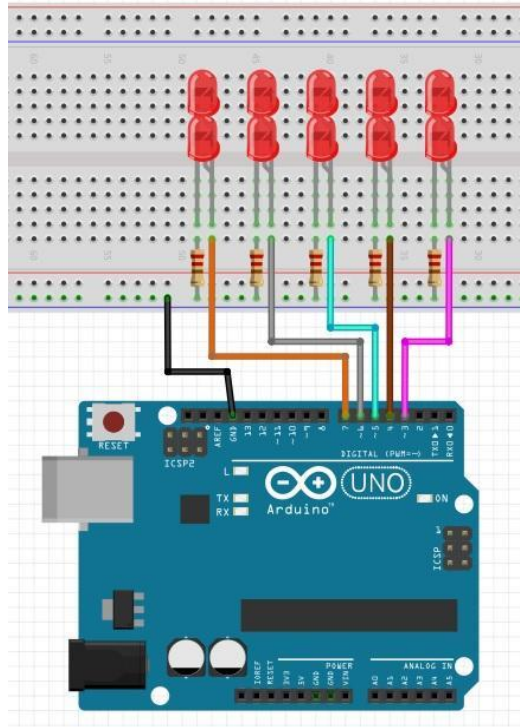
1.10 Uygula- Araba Yarışı Başlama Işıkları (Öğrenci 2)

Malzeme Listesi
Arduino Uno
Breadboard
Bağlantı kabloları
10 adet 220 ohm direnç
10 adet Kırmızı LED

Bir araba yarışında sürücüler başlama ışıklarını takip ederek yarışa başlarlar. Bu ışıklar iki sıra halinde dizilmiş ve her bir sırada 5'er adet olacak şekilde 10 tane kırmızı lambadan oluşur. Bu etkinlikte öğrenciler araba yarışlarının başlatılmasında kullanılan ışık uyarısının bir benzerini yapacaklardır. Bunun için ilk olarak iki sıranın birinci LED'leri, bir saniye sonra da ikinci 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 LEDler sönünce başlar. Tüm aşamalar arasında bir saniye olmasına dikkat edilir. Uygula etkinliği ile ilgili örnek görsel, devre şeması ve kod aşağıdaki resimlerde görülmektedir.



Resim 1.18: Uygula Etkinliği Örnek Görsel



Resim 1.19: Uygula Etkinliđi Örnek Devre Şeması

```

araba_yaris_start | Arduino 1.8.10
araba_yaris_start $
void setup() {
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
}

void loop() {
  digitalWrite(7,HIGH);
  delay(1000);
  digitalWrite(7,HIGH);
  digitalWrite(6,HIGH);
  delay(1000);
  digitalWrite(7,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(5,HIGH);
  delay(1000);
  digitalWrite(7,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(4,HIGH);
  delay(1000);
  digitalWrite(7,HIGH);
  digitalWrite(6,HIGH);
  digitalWrite(5,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(3,HIGH);
  delay(1000);
  digitalWrite(7,LOW);
  digitalWrite(6,LOW);
  digitalWrite(5,LOW);
  digitalWrite(4,LOW);
  digitalWrite(3,LOW);
  delay(5000);
}

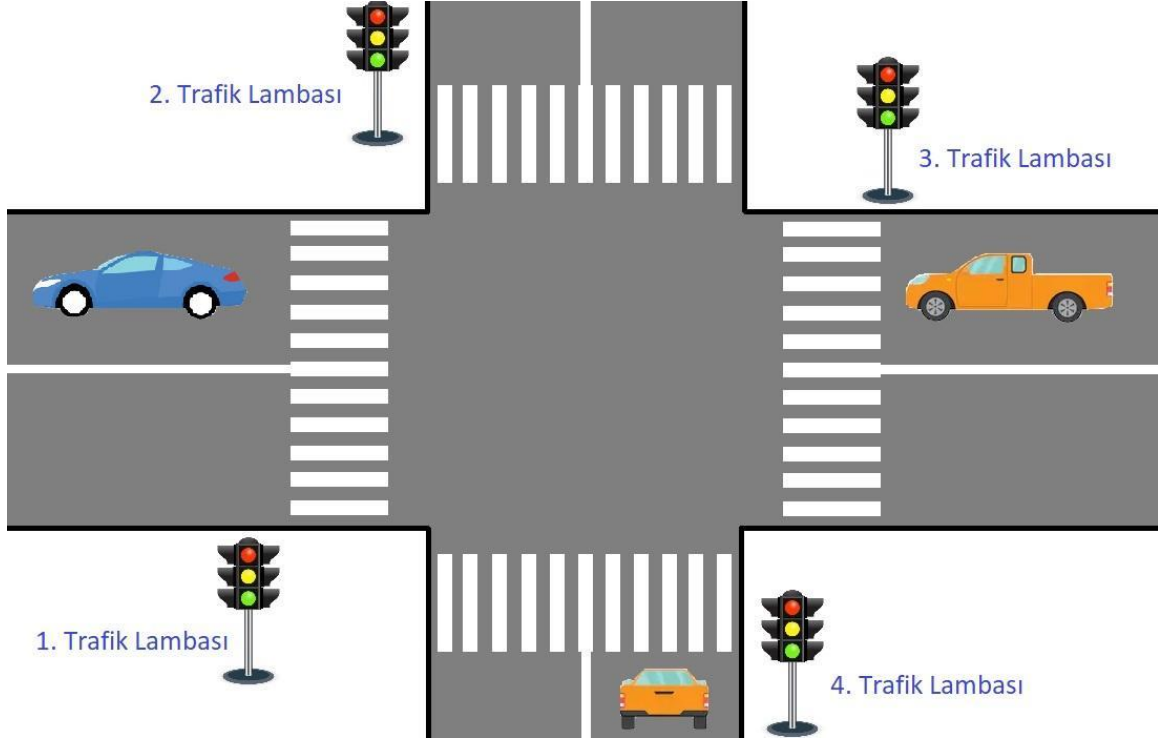
```

Resim 1.20: Uygula Etkinliđi Örnek Kodu

2. ADIM: 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 1.21: Tasarla Etkinliği Örnek Görsel

Öncelikle birinci lambadaki araçlar geçmelidir. Dolayısıyla birinci lambada ilk olarak **yeşil ışık** yanması gerekmektedir. Bu 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 sınıfta birlikte yapılmamalıdır. Eğitimci sınıfta genel bir sorun görürse öğrencilere açıklama yapabilir, bireysel sorular için gruplara açıklama yapılabilir. Gerektiği noktada eğitimci 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:

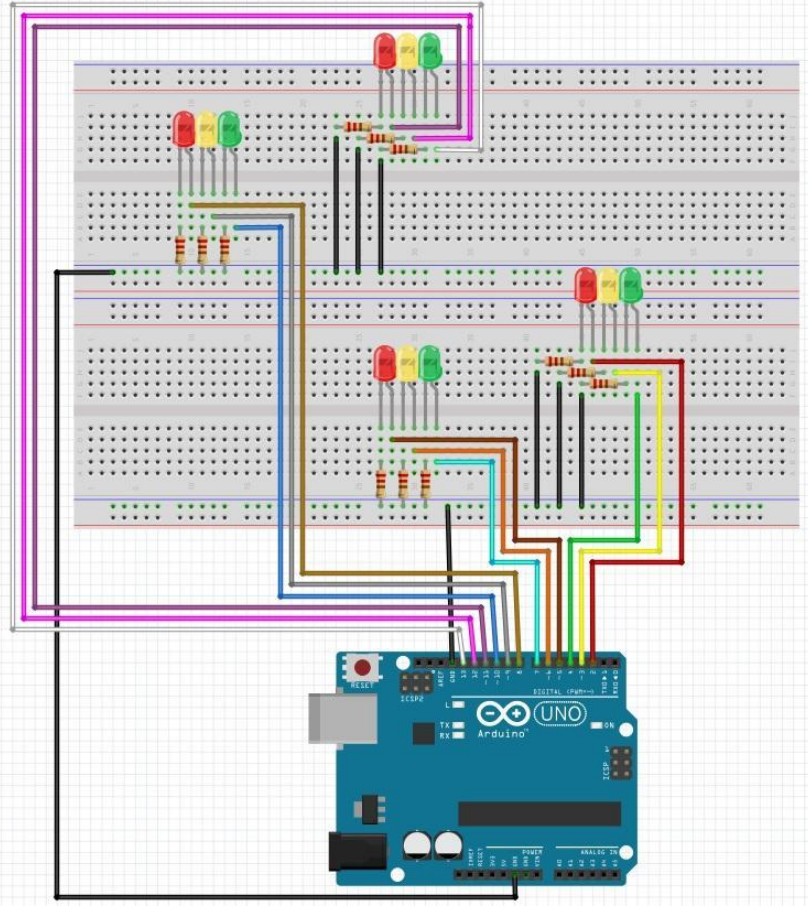
- Dört tane sarı, kırmızı ve yeşil LED setinden oluşan devre tanımlanacak.
- Herhangi bir sette yeşil ışık yanıyorsa diğer setlerdeki kırmızı LED'ler yanacak.
- Setteki yeşil LED dört saniye yandıktan sonra sarı LED yanacak. Sarı LED'in yanmasıyla yeşil LED kapatılacak.
- Sarı LED bir saniye yandıktan sonra kırmızı LED yanacak.
- 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.
- Sarı LED bir saniye sonra kapatılacak ve setteki yeşil LED yanacak.
- 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.

- Devre için dört tane sarı, kırmızı ve yeşil LED'den oluşan set breadboard'a yerleştirilir.
- Her bir LED (direnç de eklenmelidir) Arduino UNO'ya bağlanır ve devreleri tamamlanır (akım girişi -yani pin girişi- ve GND bağlantısı yapılır).
- LED setleri içerisinde biri aktif set olarak seçilir.
- Loop fonksiyonu 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.
- Dört saniye geçtikten sonra aktif setteki sarı LED'in bağlı olduğu pin ON konumuna getirilir, yeşil LED'in bağlı olduğu pin OFF konumuna getirilir.
- Bir 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.
- Bir saniye sonra iv adım ile birlikte aktif setteki sarı LED'in bağlı olduğu pin OFF konumuna getirilir ve diğer maddeler yapılmaya devam edilir.

2.2 Üret- Kavşaktaki Trafik Lambaları

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



Resim 1.22: Tasarla-Üret Etkinliği Örnek Devre Şeması

```

trafik_lamp | Arduino 1.8.10
File Edit Sketch Tools Help
trafik_lamp
1 #define k1 2
2 #define s1 3
3 #define y1 4
4 #define k2 5
5 #define s2 6
6 #define y2 7
7 #define k3 8
8 #define s3 9
9 #define y3 10
10 #define k4 11
11 #define s4 12
12 #define y4 13
13
14
15 void setup() {
16   pinMode(2, OUTPUT);
17   pinMode(3, OUTPUT);
18   pinMode(4, OUTPUT);
19   pinMode(5, OUTPUT);
20   pinMode(6, OUTPUT);
21   pinMode(7, OUTPUT);
22   pinMode(8, OUTPUT);
23   pinMode(9, OUTPUT);
24   pinMode(10, OUTPUT);
25   pinMode(11, OUTPUT);
26   pinMode(12, OUTPUT);
27   pinMode(13, OUTPUT);
28
29 }
30
31 void loop() {
32   digitalWrite(s4, LOW);
33   digitalWrite(y4, HIGH);
34   digitalWrite(k1, HIGH);
35   digitalWrite(k2, HIGH);
36   digitalWrite(k3, HIGH);
37   delay(4000);
38   digitalWrite(y4, LOW);
39   digitalWrite(k4, HIGH);
40   digitalWrite(k1, LOW);
41   digitalWrite(s1, HIGH);
42   digitalWrite(k2, HIGH);
43   digitalWrite(k3, HIGH);
44   delay(1000);
45   digitalWrite(k4, HIGH);
46   digitalWrite(s1, LOW);
47   digitalWrite(y1, HIGH);
48   digitalWrite(k2, HIGH);
49   digitalWrite(k3, HIGH);
50   delay(4000);
51   digitalWrite(k4, HIGH);
52   digitalWrite(y1, LOW);
53   digitalWrite(k1, HIGH);
54   digitalWrite(k2, LOW);
55   digitalWrite(s2, HIGH);
56   digitalWrite(k3, HIGH);
57   delay(1000);
58   digitalWrite(k4, HIGH);
59   digitalWrite(k1, HIGH);
60   digitalWrite(s2, LOW);
61   digitalWrite(y2, HIGH);
62   digitalWrite(k3, HIGH);
63   delay(4000);
64   digitalWrite(k4, HIGH);
65   digitalWrite(k1, HIGH);
66   digitalWrite(y2, LOW);
67   digitalWrite(k2, HIGH);
68   digitalWrite(k3, LOW);
69   digitalWrite(s3, HIGH);
70   delay(1000);
71   digitalWrite(k4, HIGH);
72   digitalWrite(k1, HIGH);
73   digitalWrite(k2, HIGH);
74   digitalWrite(s3, LOW);
75   digitalWrite(y3, HIGH);
76   delay(4000);
77   digitalWrite(y3, LOW);
78   digitalWrite(k4, LOW);
79   digitalWrite(s4, HIGH);
80   digitalWrite(k1, HIGH);
81   digitalWrite(k2, HIGH);
82   digitalWrite(k3, HIGH);
83   delay(1000);
84 }
Done compiling
Sketch uses 1434 bytes (4%) of program storage space.
Global variables use 9 bytes (0%) of dynamic memory.
12 Arduino/Genuino Uno on COM4

```

Resim 1.23: Tasarla-Üret Etkinliği Örnek Kodu

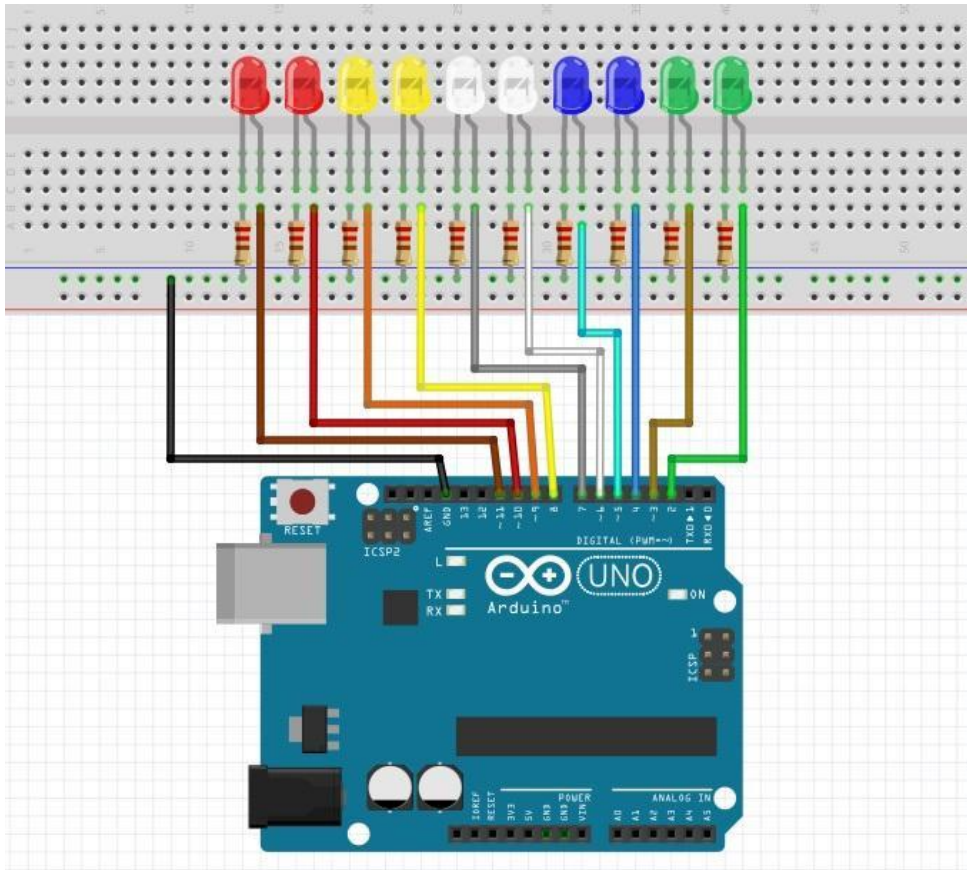
2.3 Tasarla - Müzik ile Uyumlu Hareket Eden LED Uygulaması

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 UNO 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 yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekmektedir.

2.4 Üret- Müzik ile Uyumlu Hareket Eden LED Uygulaması

Öğrenciler yukarıda çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerektiği noktada eğitmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir. Öğrencilerin aşağıdaki resimde görüldüğü gibi bir devre tasarımı yapmaları beklenir.



Resim 1.24: Üret Etkinliği Örnek Devre Şeması

3. ADIM: DEĞERLENDİR

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, eğitmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Breadboard'da devre oluştururken zorlandınız mı? Çözümlerinizi hakkında bilgi veriniz.
 - Akım, gerilim ve direnç ilişkisini anlamlandırırken zorlandınız mı?
 - Program ve mikrodenetleyici kullanarak devre elemanlarını yönetmede zorluklar yaşadınız mı?
 - Bu hafta öğrendiğiniz devre elemanlarının günlük yaşam içindeki kullanımları nelerdir? Örneklerle açıklayınız.
 - Kavşaktaki trafik lambaları etkinliğinde tasarım ve üretim aşamasının hangisinde daha çok zorlandınız? Çözüm için yazdığınız programın daha az kod satırı kullanarak yazılması mümkün müdür? Nasıl?
- 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?
- Arduino çeşitlerini internetten araştırarak en az 3 tanesi için bileşenlerinin karşılaştırma tablosunu hazırlayınız. (Hazırlanacak tabloda mikrodenetleyici modelleri, giriş-çıkış pinleri ve dahili modüllerin sayısı, boyut, çalışma gerilimleri ve kullanım alanlarına yönelik bilgileri yer almalıdır.)

4. İLAVE ETKİNLİK

4.1 Transit Geçişli Bir Kavşaktaki Trafik Lambaları

Bu etkinlikte amaç bir kavşaktaki dört farklı trafik lambasının birbirleriyle uyumlu bir şekilde çalışmasını sağlamaktır. Daha önce devresini tasarlayıp programını yazdığınız kavşaktaki trafik ışıklarını araçlar için karşılıklı transit geçecek şekilde tekrar programlayınız. Burada dikkat etmeniz gereken nokta birbirine karşı olan yollardaki trafik lambalarının aynı anda kırmızı, aynı anda sarı ve aynı anda yeşil yanmasını sağlamanız gerekeceğidir. Bu etkinlik için diğer kurallar bir kavşaktaki trafik lambaları etkinliği ile aynı olup arabaların kavşakta sola dönmeyecekleri varsayılacaktır.

2. Bölüm – Temel Devre Elemanları ile Arduino UNO ve Kodlama Aracı Arduino IDE

Ön bilgi:

- Öğrenciler “gerilim”, “akım”, “direnç” tanımlarını bilir.
- Öğrenciler direncin çalışma prensibini bilir ve akım ile arasındaki ilişkiyi açıklayabilir.
- Öğrenciler “Arduino UNO” kartı bileşenlerini bilir ve “Arduino IDE” arayüzünü kullanabilir.

Bölüm Kazanımları:

- Öğrenciler “breadboard” çalışma prensiplerini bilir.
- Öğrenciler “Seri/Paralel Bağlantı”, “Analog/Dijital veri” ve “pinler” hakkında bilgi sahibidir.
- Öğrenciler dirençler üzerindeki farklı renklerden dirençlerin ohm değerlerini hesaplayabilir.
- Öğrenciler “buzzer” devre elemanının çalışma prensiplerini bilir.
- Öğrenciler “potansiyometre” devre elemanının çalışma prensiplerini bilir.
- Öğrenciler “buton” devre elemanının çalışma prensiplerini bilir.
- Öğrenciler “potansiyometre” ile LED parlaklığını ayarlama ve seri haberleşme uygulamalarını gerçekleştirebilir.
- Öğrenciler “Buton” kontrollü LED yakma ve “buton” ile hareketli LED Uygulamalarını gerçekleştirebilir.
- Öğrenciler LED’ler, “buton”lar ve “buzzer” ile farklı uygulamaların devrelerini kurabilir ve “Arduino IDE” program kodlarını yazabilirler.
- “If/else” karar yapısı, “for” döngüsü, dizi değişkenini kullanarak programlama yapabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle “breadboard”, “seri/paralel bağlantı”, “Analog/Dijital veri” ve “Arduino UNO” üzerindeki pinler hakkında bilgi sahibi olmalarını hedeflemektedir. Ayrıca, öğrenciler “direnç renk kodları”, “potansiyometre”, “buton”, ve “buzzer” devre elemanlarının özelliklerini öğrenme ve bu devre elemanları ile farklı uygulamalar gerçekleştirebilmeleri amaçlanmaktadır. Bu kapsamda programlamada farklı kontrol yapıları, sabitler ve giriş-çıkışlar gibi konuların uygulamalar içerisindeki kullanımlarını da deneyimlemeleri hedefler içerisinde.

Göze ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	Erkek-Dişi bağlantı kablosu
Breadboard	Buzzer
220 ohm direnç	Hoparlör
560 ohm direnç	Kırmızı LED
100 ohm direnç	
10 Kohm Potansiyometre	

Haftanın İşlenişi:

Gözle: Breadboard, Seri/Paralel Bağlantı, Analog/Dijital Veri ve Pinler, direnç renk okuma, Buzzer, potansiyometre ve buton tanımlarının yapılması ve özelliklerinin verilmesi.

Uygula: Buzzer, potansiyometre ve buton devre elemanlarının kullanıldığı örnek devrelerin oluşturulması ve programlanması.

Tasarla: Bir buton ile dört hareketli LED uygulaması ve butonları kullanarak piyano tasarımının gerçekleştirilmesi.

Üret: Bir buton ile dört hareketli LED uygulaması ve butonları kullanarak piyano tasarımını gerçekleştirme; devrelerini oluşturma ve Arduino IDE içerisinde program kodlarının yazılması.

Değerlendir: Yansıtma Etkinliği.

1. ADIM: GÖZLE ve UYGULA

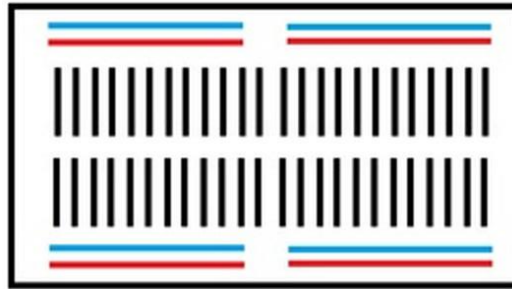
1.1 Gözle ve Uygula - Breadboard, Seri/Paralel Bağlantı, Analog/Dijital Veri ve Pinler

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
330 ohm direnç
Kırmızı LED

Breadboard

Breadboard elektronik devre elemanlarını bir arada tutan ve gerekli ara bağlantıları daha sağlıklı şekilde gerçekleştirmemizi sağlayan araçtır. Breadboard üzerinde aşağıdaki resimde de görüldüğü gibi yatay ve dikeyde ilerleyen çizgiler aslında kısa devredir, yani bu çizgiler bir kablo üzerinden bağlantılıdır diye düşünülebilir.

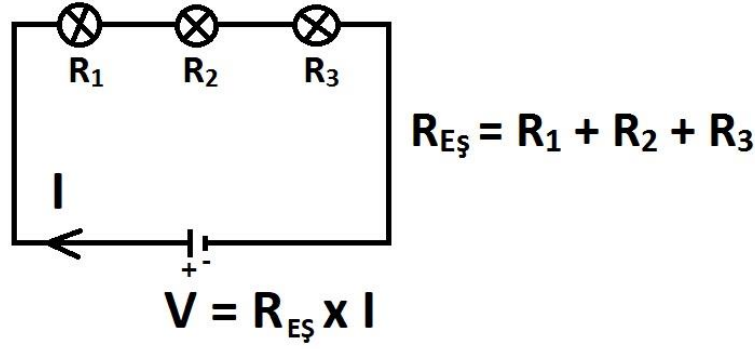
Breadboard'da yanlardan dikey şekilde uzanan hatlar genellikle güç bağlantılarını yapmak için kullanılır.



Resim 2.1: Breadboard Bağlantıları

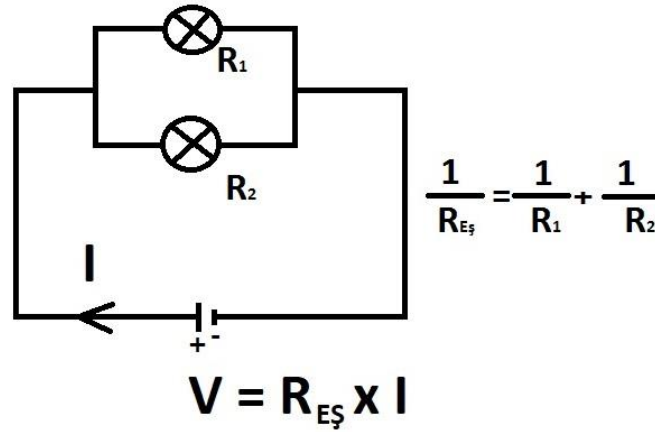
Seri-Paralel Bağlantı

Elektrik devreleri bağlantı şekillerine göre paralel ve seri olmak üzere ikiye ayrılırlar. Seri devrelerde akım devredeki elemanlardan tek bir hat üzerinden geçerler. Akım her bir eleman üzerinden geçer ve bu elemanlar akıma karşı direnç gösterirler. Bu nedenle devrede oluşan direnç artar. “Ohm kanunu” üzerinden düşünecek olursak “ $V=IxR$ ”de voltaj sabitken, direnç artacağı için devre üzerinden geçecek akım azalacaktır. Aşağıdaki resimde dirençlerin seri bağlanması görülmektedir.



Resim 2.2: Dirençlerin Seri Bağlanması

Paralel devrelerde ise devre elemanları birbirine paralel olarak bağlanır ve buna bağlı olarak direnç azalır. Gerilim sabitken direncin azalması sonucunda devre üzerinden geçecek akım artacaktır. Aşağıdaki resimde dirençlerin paralel bağlanması görülmektedir.



Resim 2.3: Dirençlerin Paralel Bağlanması

Arduino UNO ve Giriş – Çıkış Pinleri

Arduino'nun farklı versiyonlarında farklı sayıda ve türde giriş – çıkış pinleri bulunmaktadır. Bu ders kapsamında Arduino UNO üzerindeki giriş-çıkış pinlerinden bahsedilecektir. Bu pinler aslında Arduino UNO'ya bağladığımız sensör ve devre elemanlarıyla veri alışverişini için kullanılır. Arduino UNO'da analog ve dijital giriş-çıkış pinleri; toprak ve güç giriş-çıkışları bulunmaktadır. Bu pinleri kısaca açıklayacak olursak:

GND: Sensörlerin ve devre elemanlarının GND veya “-“ uçları bu pine bağlanır ve topraklama sağlar.

3,3 V: Bu pin 3,3 V'luk besleme gerektiren cihazlara bağlanır ve enerji çıkış pinidir.

5 V: Bu pin 5 V'luk besleme gerektiren cihazlara bağlanır ve enerji çıkış pinidir.

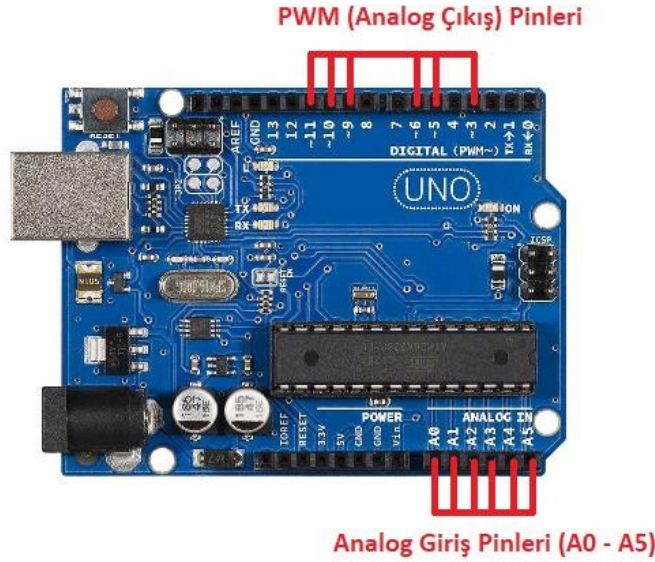
Vin: Arduino UNO kartının “5 V”, “9 V” ve “12 V” gerilimle beslenmesini sağlayan pindir. Burada dikkat edilmesi gereken bir nokta ise Arduino UNO Vin pini üzerinden beslendiğinde güç kaynağının GND ucu Arduino UNO'nun herhangi bir GND pinine bağlanmalıdır.

Analog- Dijital Veri ve Pinler

Analog ve dijital veri kavramlarını kısaca açıklamak gerekirse analog veri zamana göre sonsuz bir değer alan veri türü iken dijital veri genel olarak “var” (1) veya “yok” (0) bilgisinden oluşur. Örneğin bir arabanın hareketli ya da hareketsiz olması dijital bir veri gibi düşünülebileceken arabanın seyir halindeki hızı ise analog bir veridir.

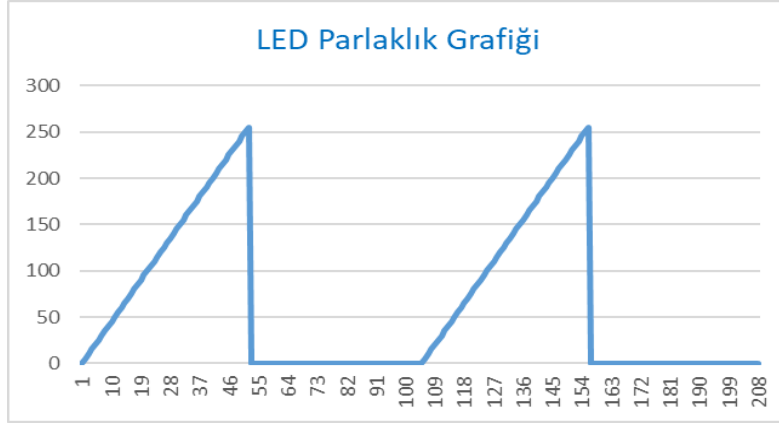
Arduino UNO üzerindeki “A0, A1, A2, A3, A4 ve A5” analog giriş pinleridir. Bu pinler, bazı sensörler ve diğer devre elemanlarından gelen analog veriyi Arduino UNO’ya 0 ile 1023 arasında bir değer olarak aktarırlar. Fakat Arduino UNO’da direk analog çıkış alabileceğimiz pinler yoktur.

PWM: Arduino UNO üzerinde analog çıkış verebilmek için dijital pinlerden bazıları (“~” işareti olan 3, 5, 6, 9, 10, 11 nolu pinler) “0 ile 255” arasında değerler olarak ve dijital çıkışın frekansını arttırarak analog bir pin gibi davranırlar. Bu çıkışlara “PWM” çıkışları denir. Bu “PWM” pinleri analog çıkış dışında, dijital giriş-çıkış pinleri olarak da kullanılabilir. Aşağıdaki resimde analog giriş – çıkış pinleri görülmektedir.

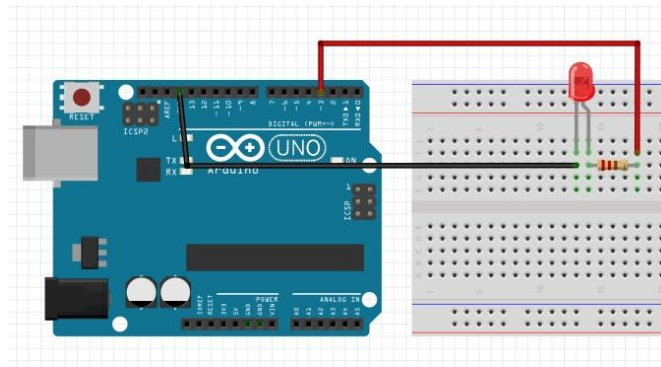


Resim 2.4: Arduino UNO Analog Giriş-Çıkış Pinleri

PWM pinlerinin kullanımına ilişkin bir örnek verilecek olursa normalde bu pinler dijital çıkış (0-5 V, HIGH-LOW vb.) üretmesine karşın bu pinlere bağlı olan LED’lerin parlaklığı analog çıkış verecek şekilde ayarlanabilir. Aşağıda öğrencilerle LED parlaklığını 5 sn içerisinde maksimum parlaklığa ulaştırıp sönen ve “5 sn” sonra tekrar yanan bir uygulama gerçekleştirilecektir. LED parlaklığının 5 sn içerisinde maksimum parlaklığa ulaşması “0” değerinden başlayarak “255” değerine kadar ulaşmasını gerektirmektedir. “5 sn” içinde sönüp tekrar yanması da bu uygulamanın bir döngü içerisinde devam edeceğini göstermektedir. Aşağıdaki resimlerde led parlaklık grafiği ve etkinlik örnek devre şeması görülmektedir.



Resim 2.5: LED Parlaklık Grafiği

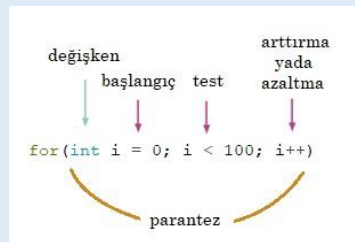


Resim 2.6: Gözle-Uygula Etkinliği Örnek Devre Şeması

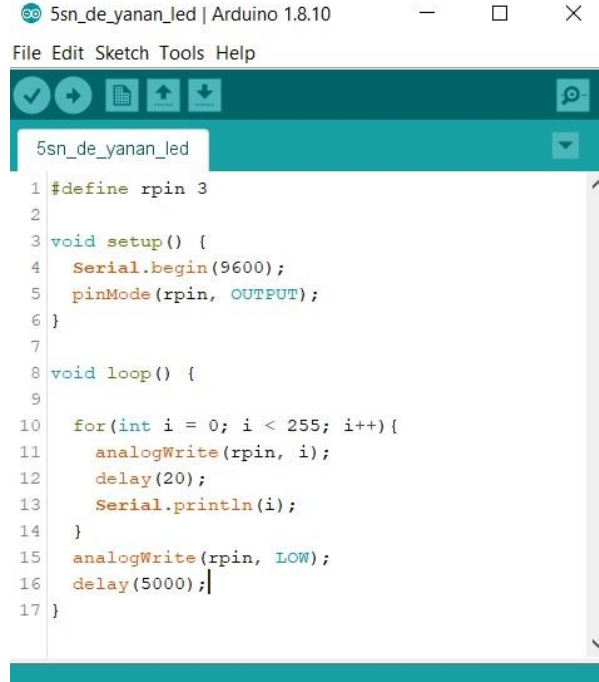
Dikkat

Eğitmen devrenin fiziksel kurulumunu ve programın yazımından önce kontrol yapılarından döngüler hakkında öğrencileri kısaca bilgilendirmelidir. Özellikle **“for”** döngüsü hakkında **“for”** deyiminin küme parantezi içine alınmış bir deyim bloğunu tekrarlamak için kullanıldığı; döngüyü artırmak ve sonlandırmak için içerisinde genellikle bir *artış sayacı (değişken)* kullanıldığını özellikle belirtip aşağıdaki söz dizimini öğrencilere göstermelidir. **“for”** ifadesinin tekrar eden işlemler için yararlı olduğu ve genellikle veri / pin topluluğu üzerinde çalışmak için dizilerle birlikte kullanıldığı öğrencilere iletilmelidir.

Arduino IDE içerisinde yazılacak program kodlarında **“i”**, **“j”**, **“k”**, **“l”**, **“m”** ve **“x”** gibi değişken isimlerinin sıklıkla kullanılacağı konusunda öğrenciler bilgilendirilmelidir. Özellikle aşağıdaki söz diziminde parantez içerisindeki **“i”** değişkeninin Void fonksiyonlarından önceki bölümde tanımlanmadığı için parantez içerisinde **“int”** olarak yazılıp başlangıç değerinin **“0”** olarak tanımlandığına dikkat çekilmelidir.



Uygulamanın devamında devre şeması ve daha sonra da devreye ilişkin ‘‘Arduino IDE’’ programı eğitmen tarafından tüm öğrencilere gösterilecektir. Bu aşamada LED parlaklığının 5 sn içerisinde hangi aralıkta değişim gösterdiğini öğrencilere gösterebilmek için Arduino IDE’nin sağ üst köşesindeki simgeye tıklanarak seri port ekranı açılmalıdır. LED parlaklık değeri değişiminin gözlemleneceği bu ekranın sağ alt köşesinde yer alan değeri (baud değeri) Arduino IDE programında kullanılan değeri ile aynı olmalıdır (örnek: 9600 baud). Aşağıdaki resimde etkinlik için örnek kod görülmektedir.



```

1 #define rpin 3
2
3 void setup() {
4   Serial.begin(9600);
5   pinMode(rpin, OUTPUT);
6 }
7
8 void loop() {
9
10  for(int i = 0; i < 255; i++){
11    analogWrite(rpin, i);
12    delay(20);
13    Serial.println(i);
14  }
15  analogWrite(rpin, LOW);
16  delay(5000);
17 }

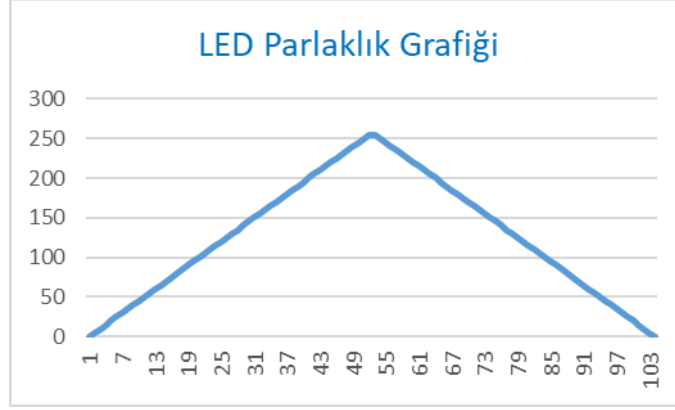
```

Resim 2.7: Gözle-Uygula Etkinliği Örnek Kodu

1.2 Uygula- LED Parlaklığı (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
330 ohm direnç
Kırmızı LED

Bu etkinlikteki amaç öğrencilerin Arduino UNO’nun 6 nolu pinini (PWM pinlerinden biri) kullanarak bu pine bağlı olan LED’in parlaklığını 3 sn de maksimuma ulaştırıp 3 sn’de de söndüren uygulamayı gerçekleştirmektir. Eğitmen bu uygulamaya ilişkin öğrencilere aşağıdaki voltaj seviyesini gösteren grafiğe uygun şekilde devreyi kurmalarını ve Arduino IDE içerisinde programı yazmalarını isteyecektir. Aşağıdaki resimlerde led parlaklık grafiği ve programın örnek bir kodu verilmiştir.



Resim 2.8: LED Parlaklık Grafiği

```

led_pwm_3_sn §

int rpin=11;

void setup() {
  Serial.begin(9600);
  pinMode(rpin,OUTPUT);
}

void loop() {
  for(int i=0; i<=255; i++){
    analogWrite(rpin,i);
    delay(12);
    Serial.println(i);
  }
  for(int k=255; k>0; k--){
    analogWrite(rpin,k);
    delay(12);
    Serial.println(k);
  }
}

```

Resim 2.9: Uygula Etkinliği Örnek Kodu

1.3 Gözle - Direnç, Direnç Okuma

Direnç

Daha önceki derste **Ohm kanunundan** ($V = I \times R$) bahsedilmişti. Bu konuyu tekrar hatırlatmak gerekirse sabit bir gerilim altında akımı azaltmak için devrede yer alan direnci değiştirebiliriz. Yani devrede akımı ayarlamak için direnç değerinde değişiklik yapılabilir. Bu durumu gerçek hayattan bir örnek ile açıklamak gerekirse altı şeritli bir otoyolda gerçekleşen bir kaza sonucu 2 şeridin iptal olması durumunda (direnç artımı) trafik akışı yavaşlayacaktır (akım azalacaktır).

Direnç renk okuma:

Dirençlerin üzerinde “ohm” değerleri direk olarak yazılmaz. Bunun yerine direnç üzerindeki renk kodları ile ohm cinsinden değerleri gösterilir. Dirençlerin üzerinde hassasiyetlerine bağlı olarak 4, 5 ya da 6 farklı renk bulunur. Bu renklerden sonuncusu direncin tolerans değerini belirtir. Direncin toleransı direncin üretimden kaynaklanan hata payını gösterir. Örnek vermek gerekirse 100 ohmluk direnç eğer %5’lik bir toleransa sahipse bu dirençten 95-105 ohm arasında direnç değeri elde edilebilir. Dolayısıyla direncin toleransı düştükçe daha hassas devreler kurulabilir. Diğer renkler ise aşağıda verilen formül kullanılarak direncin ohm

değerinin hesaplanmasında kullanılır. Formülde yer alan şeritlerin renk değerleri ise şu şekildedir:

Renklerin değerleri: Siyah = 0, Kahverengi = 1, Kırmızı = 2, Turuncu = 3, Sarı = 4, Yeşil = 5, Mavi = 6, Mor = 7, Gri = 8, Beyaz = 9.

$$\text{Direncin Ohm Değeri} = (10 \times (\text{ilk şeridin renk değeri}) + 1 \times (\text{ikinci şeridin renk değeri})) \times 10^{(\text{üçüncü şeridin renk değeri})}$$

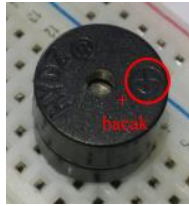
Dikkat

Dirençlerin renk sayı değerlerini kolayca akılda tutmak için “Sokakta Sayamam Gibi” ifadesi ezberlenebilir.

0	1	2	3	4	5	6	7	8	9								
S	o	K	a	K	T	a	S	a	Y	a	M	a	M	G	i	B	i
i	a	l	u	a	e	a	o	r	r	i	r	i	y	e	z	a	z
y	h	r	r	r	ş	v	r	i	h	u	l	u	h	u	l	u	h
a	v	i	u	i	i	i	i	i	v	e	n	g	i	i	i	i	i
h	e	r	m	n	c	u	n	g	i	i	i	i	i	i	i	i	i
	r	e	n	z	i												
	g	i															
	i																

1.4 Gözle ve Uygula - Buzzer

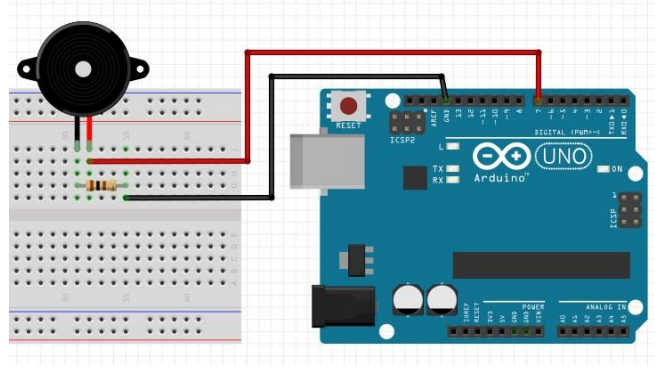
Daha önce ışıklı devre elemanları işlenmişti. Işıklı devre elemanları gibi sesli devre elemanları da bulunur. Bunlar ses üretebilirler. Bu uygulamada kullanılacak elektronik devre elemanı olan **Buzzer** da ses üretebilen bir devre elemanıdır. Günlük yaşamda şarkı ve ses kaydı gibi ses unsurlarını oynatmak 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 “+” (artı) küçük olanı ise “-” (eksi) bacaktır. Aşağıda resimlerde iki adet buzzer görülmektedir. Bu resimlerde + bacaklar gösterilmiştir.



Resim 2.10: Buzzer Bacak Bağlantıları

Aktif buzzer’a çalışma aralığı içerisinde bir doğru akım uygulandığında bip sesi üretir. Pasif buzzera doğru akım uygulandığında istenilen ses elde edilemez. Pasif buzzer’dan istenilen sesi elde etmek için Arduino IDE içerisinde gerekli komutları 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 yapılıdır. Bundan dolayı 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 7. dijital pinine bağlanmıştır. Buzzer’ın diğer bacağı ise 100 ohm bir direnç üzerinden Arduino UNO kartının GND pinine bağlanmıştır.



Resim 2.11: Gözle-Uygula Etkinliği Örnek Devre Şeması

Aşağıda resimde aktif buzzer için Arduino IDE’de yazılmış bir kod bulunmaktadır. Arduino UNO kartının 7. dijital pinine bağlanan buzzer bu kod sayesinde bir saniye bip sesi çıkarır, bir saniye bekler ve bu işlemi sürekli yapar.

```

buzzer | Arduino 1.8.10
File Edit Sketch Tools Help
buzzer §
1 #define buzzerPin 7
2
3 void setup() {
4   pinMode(buzzerPin, OUTPUT);
5 }
6
7
8
9 void loop() {
10
11   digitalWrite(buzzerPin, HIGH);
12   delay(1000);
13   digitalWrite(buzzerPin, LOW);
14   delay(1000);
15 }
16 }
17

```

Resim 2.12: Gözle-Uygula Etkinliği Örnek Kodu

Bildiğimiz gibi hemen hemen her sesin kendine ait bir notası vardır. Pasif buzzer ile (buzzer’ın desteklediği ölçüde) istenilen nota veya istenilen frekanstaki ses çalınabilir. Notaların her biri için belirli olan bir Hertz cinsinden frekans değeri vardır. Aşağıdaki tabloda notaların Hertz cinsinden frekans karşılıkları verilmiştir.

Notaların Frekans (Hz) Karşılıkları							
DO	RE	Mİ	FA	SOL	LA	Sİ	DO (ince)
262	294	330	349	392	440	494	523

Arduino IDE içerisinde her bir notanın frekans olarak karşılığı girildiğinde **tone** komutu ve gerekli kontrol yapıları kullanılarak C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C_ (ince Do) notaları çalınabilir. Frekans değerleri bilindiğinde ara notalar olan diyezler de

çalınabilir. Aşağıda devre şeması ve Arduino IDE ile yazılmış program C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C_ (İnce Do) notalarını sırayla çalıp duran bir buzzer uygulamasını içermektedir. Uygulamada buzzer'ın artı ucu 13 numaralı dijital pine bağlanmış ve bir kez çalışması amacıyla program setup fonksiyonu altına yazılmıştır. Buzzer ile çalınacak notalar aşağıdaki resimde sıra ile bir dizi içine alınmış; nota sayısını bir arttırarak for döngüsü içerisinde tanımlanmıştır.

```

buzzer_nota

int buzzerPin=13;
int notasayisi=8;
int C=262;
int D=294;
int E=330;
int F=349;
int G=392;
int A=440;
int B=494;
int C_=523;
int notalar[]={C,D,E,F,G,A,B,C_};

void setup() {

  for(int i=0; i<notasayisi; i++)
  {
    tone(buzzerPin, notalar[i]);
    delay(500);
    noTone(buzzerPin);
    delay(20);
  }
  noTone(buzzerPin);
}

void loop() {
}

```

Resim 2.13: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Eğitmen devrenin fiziksel kurulumunu ve programın yazımından önce **diziler** hakkında öğrencileri kısaca bilgilendirmelidir. Özellikle dizilerin aynı türden bilgileri saklamak için kullanılan veri yapıları olduğu ve belirli sayıda aynı türden veriyi bellekte saklamak için kullanılan değişkenler listesinden oluştuğu vurgulanmalıdır.

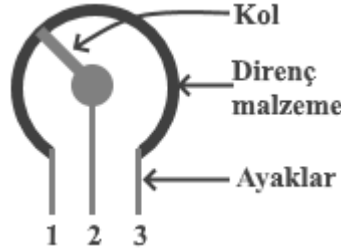
Ayrıca dizilerin tanımlanmasında köşeli parantezlerin ([]) kullanıldığı; dizi içerisindeki elemanlara yine köşeli parantezler ile erişim sağlandığı ve dizilerde ilk elemanın sayımının **sıfırdan (0)** başladığı öğrencilere iletilmelidir. Dizilere başlangıç değeri atamasının küme parantezleri ile yapılabileceğine dikkat çekilmelidir.

1.5 Uygula – Şarkı Çalan Buzzer (Öğrenci 1)

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

1.6 Gözle ve Uygula - Potansiyometre ile LED parlaklığını ayarlama ve Seri Haberleşme

Yukarıda bahsedilen analog pinlerden gelen değerlerin okunmasında birçok devre elemanından yararlanır. 0 V ile 5 V arasındaki ara değerleri de üreten bu devre elemanlarından biri de potansiyometredir. **Potansiyometre**, dışarıdan gelen fiziksel müdahaleler ile değeri değiştirilebilen dirençlerdir. Potansiyometrenin iki görevi bulunur. Bu görevler direnç ayarlama (ayarlı direnç) ve voltaj bölmedir. Aşağıdaki resimde potansiyometrenin iç yapısı gösterilmiştir. Potansiyometrenin “10 K Ohm” olduğunu düşünelim. Potansiyometre ayarlı direnç olarak kullanılmak istenirse 1 ve 2 numaralı bacakları kullanılarak devreye bağlanır. Kol 1 numaralı bacağına yaklaştırıldıkça direnç azalır ve “0 Ohm” değerine kadar iner. Kol 3 numaralı bacağına yaklaştırıldıkça direnç artar ve “10K Ohm” değerine kadar çıkar. Aynı işlem 2 ve 3 numaralı bacaklarla da yapılabilir. Fakat bu durumda kol 3 numaralı bacağına yaklaştıkça direnç azalacaktır ve 1 numaralı bacağına yaklaştıkça direnç artacaktır. Bu durumun nedeni bağlı iki bacak arasındaki direnç oluşturan malzemenin miktarıdır. Miktar arttıkça direnç artar ve tersinde ise azalır. Aşağıdaki resimde potansiyometrenin ayak yapısı görülmektedir.



Resim 2.14: Potansiyometre Ayak Yapısı

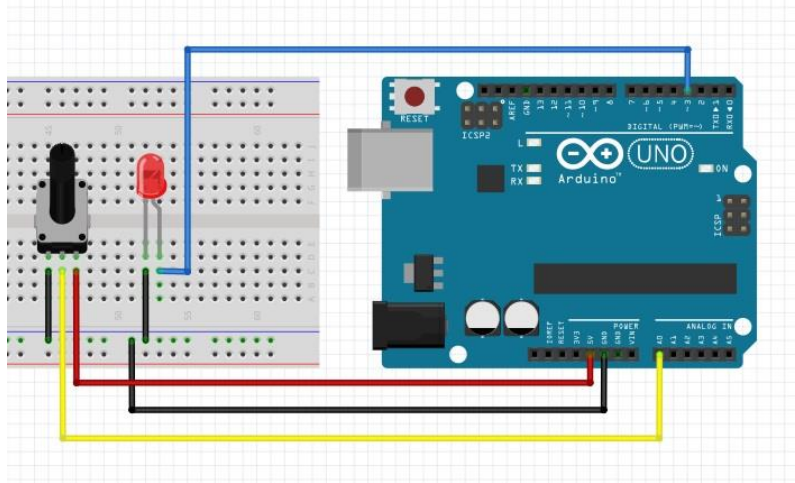
Potansiyometre voltaj bölücü olarak kullanılmak istenirse bir güç kaynağının (5 V olduğu varsayalım) “+” ucu 1 numaralı bacağına, “-“ (GND) ucu ise 3 numaralı bacağına bağlanır. Çıktı değeri 2 numaralı baktan alınır. Potansiyometrenin kolu çevrilerek çıkış gerilimi değiştirilebilir. Böylece potansiyometre kullanılarak 5 V’luk bir güç kaynağı kullanıldığında “0 V-5 V” arası voltaj değerleri elde edilir. Aşağıdaki resimde “10K Ohm” bir potansiyometre görülmektedir. Potansiyometrenin “10K” olduğu üst kısmında bulunan yazı ile belirtilmiştir. Aşağıdaki resimde potansiyometre görülmektedir.



Resim 2.15: Potansiyometre

Aşağıdaki resimlerde örnek olarak verilen devre şeması ve Arduino IDE ile yazılmış program potansiyometre kullanılarak bir LED’in parlaklığını değiştirme uygulamasını içermektedir. Potansiyometrenin sol bacağı GND, sağ bacağı 5 V ve orta bacağı A0 analog pinine bağlandığı devrede LED’in bir bacağı 3. pine, diğer bacağı da GND pinine bağlanmalıdır. Analog girdi 0-1023 arasında değer alırken pwm pini (3. pin) 0-255 arasında bir değer çıktı verebilmektedir. 0-1023 arasındaki değerin 0-255 arasındaki değere çevirmek için programda analog pinden

alınan değer “deger” isimli bir değişkene atılır ve bu değer 4’e bölünerek LED’e aktarılmak suretiyle parlaklık kontrolü yapılır. Eğitmen devre şemasını kurulumunu öğrencilere gösterir ve kurar; Arduino IDE ile yazılmış programın çalışmasını öğrencilere gösterir. Uygulama öğrencilere gösterilirken seri port ekranı açılıp öğrencilerin potansiyometre ile değerini değiştirdiğinde LED’in parlaklığının değişimini gözlemlemeleri sağlanır. Ardından öğrencilerin aynı devreyi kurmalarını ve kodu yazarak çalıştırmalarını sağlar.



Resim 2.16: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

potansiyometre_kontrollu_led | Arduino 1.8.10
File Edit Sketch Tools Help
potansiyometre_kontrollu_led $
1 #define pot A0
2 #define led 3
3
4 void setup() {
5   Serial.begin(9600);
6   Serial.println("Potansiyometreden okunan analog değer:");
7 }
8
9 void loop() {
10
11   int deger = analogRead(pot);
12   deger = deger / 4;
13   analogWrite(led, deger);
14   Serial.println(deger);
15   delay(100);
16
17 }
18

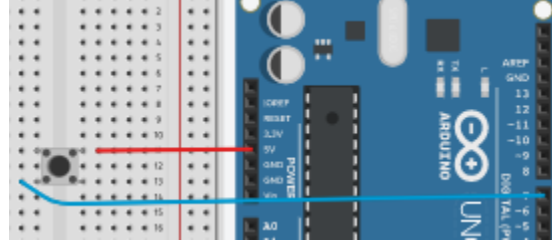
```

Resim 2.17: Gözle-Uygula Etkinliği Örnek Kodu

1.7 Gözle ve Uygula – Buton Kontrollü LED Yakma

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” (0 V) veya “1” (5 V) değerleri gönderip işlemler yapılabilir. Bu etkinlikteki amaç bir push

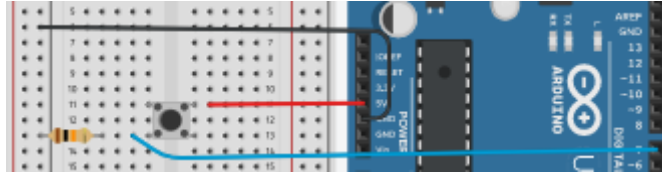
butona basıldığında bir LED yakan devreyi tasarlayıp programı yazmaktır. Bu iş için bir push buton kullanılacaktır. Push butonun aşağıdaki şekilde gösterildiği gibi dört bacağı bulunur. Bunlardan karşılıklı ikisi (mavi çizgi ile bağlı gösterilenler) birbirlerine bağlıdır. Aynı şekilde karşıdaki iki bacak da birbirine bağlıdır. Fakat bu gruplar birbirine bağlı değildir. Butona basıldığında bu iki grup birbirine bağlanmış olur.



Resim 2.18: Buton Bağlantı Örneği

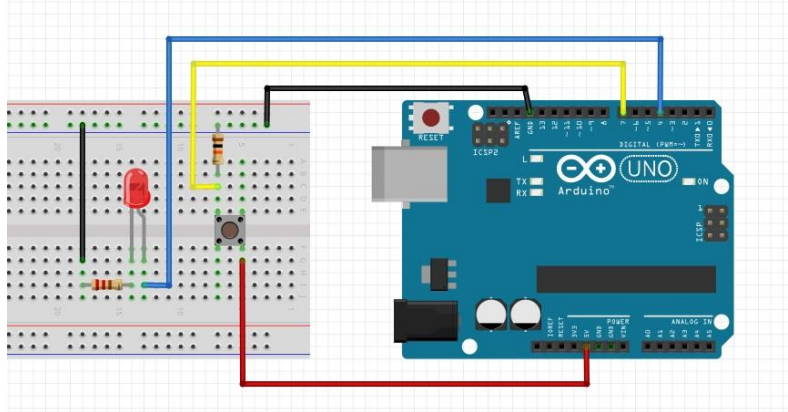
Yukarıdaki devre resminde butonun bir bacağına “5 V” verilmiştir, diğer bacağı ise “7” numaralı pine bağlanmıştır. Butona basıldığında “7” numaralı pine “1” bilgisi iletilecektir. Bu pinden gelen “1” değeri okutulabilir. Fakat bu bağlantının çeşitli sakıncaları bulunmaktadır. Arduino bu bağlantı ile butondan gelen değerleri yanlış okuyabilir. Yanlış okumaları engellemek için pull down direnç bağlantısı yapılmalıdır.

Pull down direnç bağlantısı aşağıdaki resimde gösterilmiştir. Butonun bacağına “10K Ohm” bir direnç ile GND bağlantısı yapılmıştır. Böylece butona basıldığında “7” numaralı pine “1”; butona basılmadığında “7” numaralı pine “0” bilgisi gönderilmiş olur.



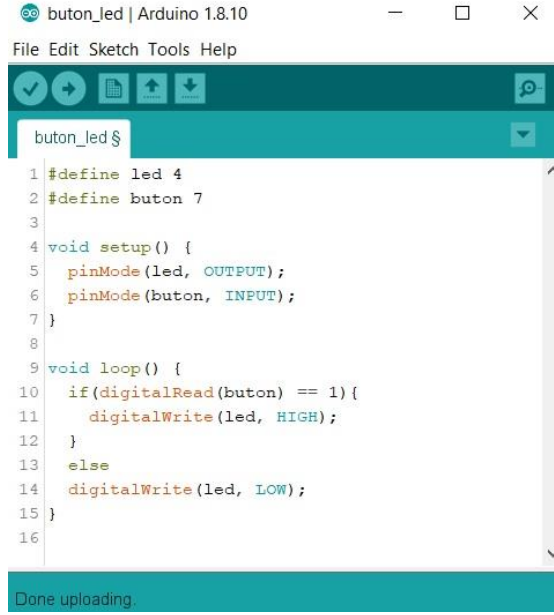
Resim 2.19: Buton Bağlantı Örneği

Aşağıdaki resimde görülen devre butona basıldığında “4” numaralı dijital pine bağlı LED’in yanması için tasarlanmıştır.



Resim 2.20: Gözle-Uygula Etkinliği Örnek Devre Şeması

Butona basıldığında “7” numaralı pinde “1” değeri okunur. “7” numaralı pin’de “1” değeri okunduğunda “4” numaralı pin’den “1” (5 V) bilgisi gönderilip LED yakılmalıdır. Yukarıda devre şeması verilen uygulamanın Arduino IDE’de yazılmış program örneği aşağıdaki resimde verilmiştir. Program kodunda buton değerinin “1” olması şartıyla (if kontrol yapısı) LED’in yanma eylemi gerçekleştirmesi; diğer durumlarda ise (else komutu) LED’in sönme ifadesi eklenmiştir. Eğitim devre şemasını kurulumunu öğrencilere gösterir ve kurar; Arduino IDE ile yazılmış programın çalışmasını öğrencilere gösterir. Ardından öğrencilerin aynı devreyi kurmalarını ve kodu yazarak çalıştırmalarını sağlar.



```

1 #define led 4
2 #define buton 7
3
4 void setup() {
5   pinMode(led, OUTPUT);
6   pinMode(buton, INPUT);
7 }
8
9 void loop() {
10  if(digitalRead(buton) == 1){
11    digitalWrite(led, HIGH);
12  }
13  else
14    digitalWrite(led, LOW);
15 }
16
Done uploading.

```

Resim 2.21: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

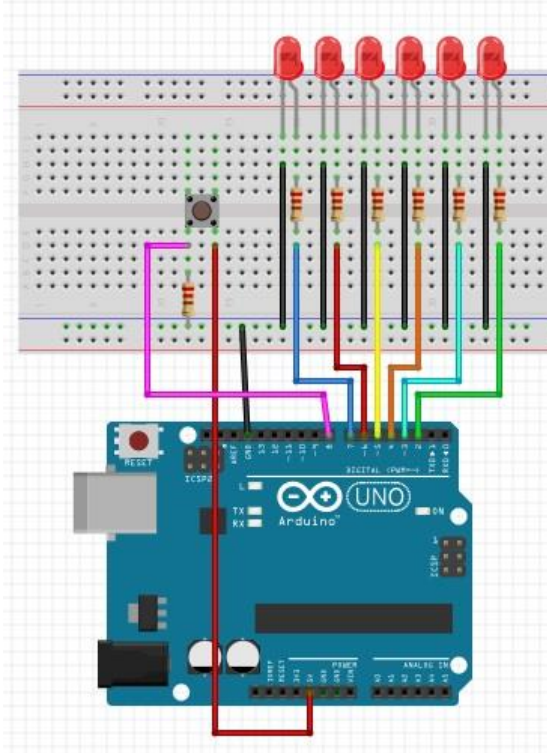
Eğitmen Arduino IDE içerisinde programın kodlarını öğrencilere gösterirken “#if/else” deyiminin koşullu ifadeleri yürütmek için kullanıldığını ve temel kod akışı üzerinde daha fazla denetim sağlamak için tercih edildiğini öğrencilere aktarmalıdır. “#if/else” deyiminde “if”in eğer, “else”in ise değil anlamına geldiği belirtilmelidir. Ayrıca “if” ve “else” in program içerisinde birlikte kullanıldığı; “else” in tek başına kullanılmadığı uyarısı yapılmalıdır.

1.8 Uygula – Buton ile Soldan-Sağa ve Sağdan-Sola Hareket Eden LED Uygulaması

Bu uygulamada amaç LED’leri kullanarak ışığın soldan sağa ve sağdan sola kayan bir şekilde hareket ettiği hissini uyandırmaktır. Bunun için fiziksel olarak yan yana bulunan LED’ler buton yardımıyla sırayla açılıp kapatılmalıdır. Ardışık LED’lerdeki açılıp kapanma göz tarafından ışığın hareketi olarak algılanır. Butona bastıktan sonra ışıktaki 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. Bu uygulamada öğrencilerin bilgisayar ve set başında çalışarak istenilen görevi yerine getirmeleri beklenmektedir. Öğrencilere eğitimci tarafından uygulamayı tamamlamak için yeterli süre verilir. Gerekli durumlarda eğitimci öğrencilere rehberlik etmelidir. Öğrencilerin aşağıdaki resimlerde görülen devreye ve programa benzer bir program hazırlamaları beklenir.

Dikkat

Eğitmen Arduino IDE içerisinde programın kodlarını öğrencilere gösterirken **#return** deyiminin bir fonksiyonun işlemi bittikten sonra çağrıldığı yere değer döndürmesi anlamı taşıdığı ve o fonksiyonun çalışmasını durduğunu hatırlatılması yerinde olacaktır. Uygulamada butona basılmadığı durumda “else” ifadesi ile LED’lerin başlangıç durumuna geleceği belirtilmelidir.



```
buton_ile_sagdan_sola | Arduino 1.8.10
buton_ile_sagdan_sola §
int ledler[] = {2,3,4,5,6,7};
int buton=8;

void setup() {
  for(int i = 0; i<6; i++){
    pinMode(ledler[i], OUTPUT);
  }
  pinMode(buton, INPUT);
}

void loop() {
  if (digitalRead(buton)==1){
    for(int j = 0; j<6; j++){
      digitalWrite(ledler[j], HIGH);
      delay(100);
      digitalWrite(ledler[j], LOW);
    }
    for(int k = 5; k>-1; k--){
      digitalWrite(ledler[k], HIGH);
      delay(100);
      digitalWrite(ledler[k], LOW);
    }
  }
  else
  return;
}

Yükleme tamamlandı.
```

Resim 2.22: Uygula Etkinliği Örnek Devre Şeması ve Kodu

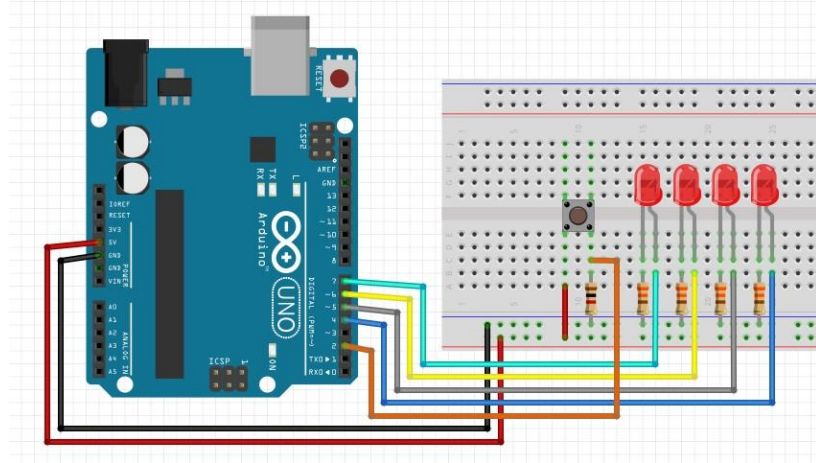
2. ADIM: TASARLA ve ÜRET

2.1 Tasarla- Bir Buton Dört Hareketli LED Uygulaması

Bu etkinlikte 4 adet LED bir buton tarafından çalıştırılacaktır. Devrede butonun basılma şekline göre LED’ler ile animasyon yapılacaktır. Devrenin çalışma şekli aşağıdaki gibi olmalıdır:

- (i) Butona birinci defa basıldığında 1, 2, 3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 1 numaralı LED yanar ve yanık kalır.
- (ii) Butona ikinci defa basıldığında 1, 2, 3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 2 numaralı LED yanar ve yanık kalır.
- (iii) Butona üçüncü defa basıldığında 1,2,3 ve 4 numaralı LED’ler sırasıyla yanıp söner. Devamında sadece 3 numaralı LED yanar ve yanık kalır.
- (iv) Butona dördüncü defa basıldığında 1,2,3 ve 4 numaralı LED yanıp söner. Devamında sadece 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.

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışır. Eğitimci gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir. Aşağıdaki resimde etkinlik örnek devre şeması görülmektedir.



Resim 2.23: Tasarla Etkinliği Örnek Devre Şeması

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Devrenin fiziksel olarak kurulumunda buton ve LED'ler için kullanılacak olan dirençlerin ohm değerleri neler olmalıdır?
- Devrenin fiziksel olarak kurulumunda LED'lerin Arduino UNO pinlerine bağlantılarında nasıl bir sıra izlenmelidir?
- Arduino IDE içinde yazılacak olan kodlarda hangi tür kontrol yapıları kullanılmalıdır?
- Arduino IDE içerisinde yazılacak olan kodlarda butonların sıra ile basılması durumlarında LED'lerin yanma ve yanık kalma durumlarının algoritması nasıl olmalıdır?

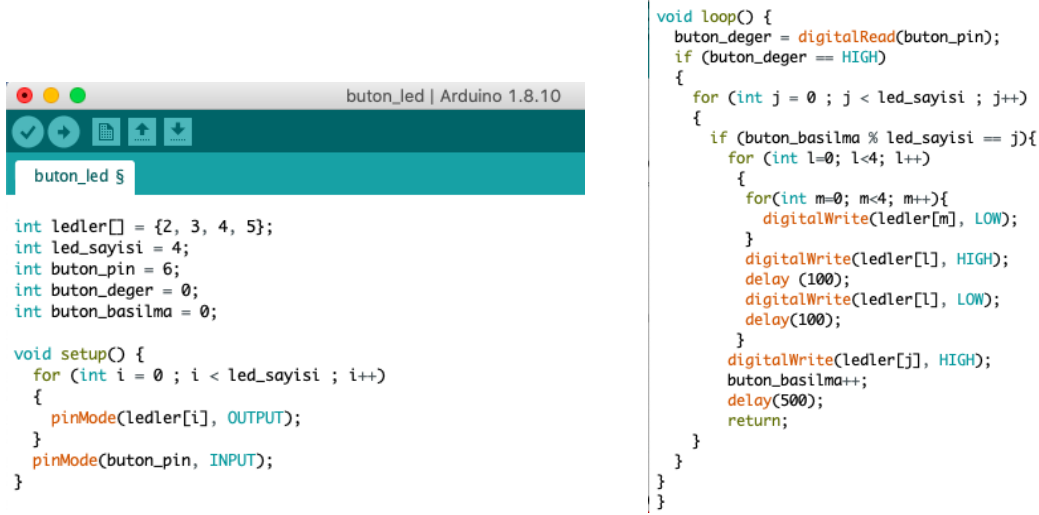
Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir.

- Devrenin fiziksel kurulumu için 220 ve 10K ohm'luk dirençler kullanılmalı ve Arduino UNO üzerindeki dijital pin bağlantıları ile Void fonksiyonları öncesindeki tanımlamalar bağlantılardaki pin değerleri ile örtüşmelidir.
- Başlangıçta LED'lerin sönmesi ve daha sonra sıra ile yanma ve butonun basılma sayısı ile yanık kalacak LED'lerin sıralaması iç içe planlanmalıdır.
- Butonun basılma sayısı ve butonun basıldığında aktif çalışması durumlarının gerektireceği ifadelerin if kontrol yapısı ile tanımlanabileceği düşünülebilir.
- Butonun basılma sayısı ile LED'leri yakma ve söndürme işlemleri için for döngü yapıları kullanılmalıdır.

Bu etkinlik için kullanılacak devre tasarımı yukarıda verilmiştir.

2.2 Üret- Bir Buton Dört Hareketli LED Uygulaması

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle “Arduino IDE” içerisinde yazılacak program kodlarında iç içe döngülerin tasarlanmasında ve çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.



```

int ledler[] = {2, 3, 4, 5};
int led_sayisi = 4;
int buton_pin = 6;
int buton_deger = 0;
int buton_basilma = 0;

void setup() {
  for (int i = 0 ; i < led_sayisi ; i++)
  {
    pinMode(ledler[i], OUTPUT);
  }
  pinMode(buton_pin, INPUT);
}

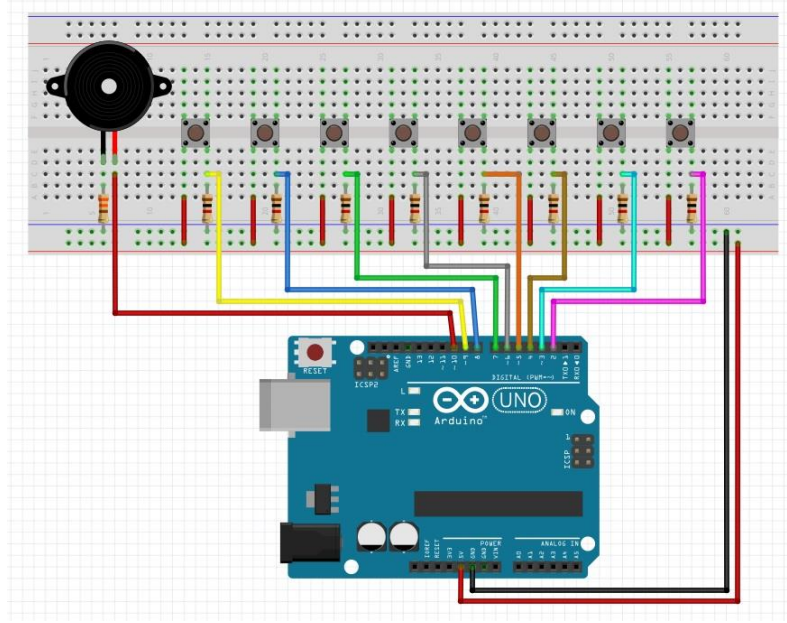
void loop() {
  buton_deger = digitalRead(buton_pin);
  if (buton_deger == HIGH)
  {
    for (int j = 0 ; j < led_sayisi ; j++)
    {
      if (buton_basilma % led_sayisi == j){
        for (int l=0; l<4; l++){
          for(int m=0; m<4; m++){
            digitalWrite(ledler[m], LOW);
          }
          digitalWrite(ledler[l], HIGH);
          delay (100);
          digitalWrite(ledler[l], LOW);
          delay(100);
        }
        digitalWrite(ledler[j], HIGH);
        buton_basilma++;
        delay(500);
        return;
      }
    }
  }
}

```

Resim 2.24: Üret Etkinliği Örnek Kod

2.3 Tasarla- Butonlar ile Piyano Yapımı

Bu etkinlikteki amaç setler içerisinde yer alan butonları ve Buzzer'ı kullanarak bir piyano tasarlamaktır. Bunun için C (Do), D (Re), E (Mi), F (Fa), G (Sol), A (La), B (Si), C_ (ince Do) notalarının her birinin frekans değerlerini çalacak şekilde butonlar ve Buzzer ile Arduino UNO bağlantıları gerçekleştirilerek devre kurulur. Aşağıdaki resimde örnek bir devre şeması verilmiştir. Devrenin kurulumunda öğrencilerin butonlar ve Buzzer için GND ve dijital pin değerlerinin bağlantısını doğru bir şekilde gerçekleştirmeleri önemlidir. Arduino IDE içerisinde öğrencilerin yazacakları kodlarda değişken tanımlarına, if kontrol yapısının kurulmasına ve tone fonksiyonunun doğru şekilde kullanılmasına dikkat edilmelidir.



Resim 2.25: Tasarla Etkinliği Örnek Devre Şeması

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışırlar. Eğitimci gerekli noktalarda yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat çözümü öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Devre oluşturulurken butonların dirençleri ve “5 V” güç bağlantıları Breadboard ve Arduino UNO arasında nasıl yapılmalıdır?
- Devre oluşturulurken “Buzzer” bağlantısı “Breadboard” ve “Arduino UNO” arasında nasıl yapılmalıdır?
- Arduino IDE içerisinde “Void” fonksiyonlarından önce Buzzer, buton bağlantı pinleri ve notaların ilk değer değişkenlerinin tanımlamaları nasıl yazılmalıdır?
- Setup fonksiyonu içerisinde hangi değişkenler giriş ve çıkış olarak tanımlanmalıdır?
- Loop fonksiyonu içerisinde notaların frekanslarını doğru şekilde Buzzer’da çalınmasını sağlayacak kontrol yapısı nasıl yazılmalıdır?
- tone fonksiyonu değerleri neler olmalıdır?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Devre için buton bağlantıları Breadboard üzerinde Arduino UNO içerisindeki pinlere sıralı şekilde bağlanmalıdır.
- Arduino IDE içerisinde nota değerlerinin başlangıç değerleri tanımlanmalıdır.
- Arduino IDE programında her bir buton için if kontrol yapısı her bir nota değeri için butonlardan okunmalıdır.
- Arduino IDE programında her bir butona karşılık olarak tone fonksiyonu ilgili Hertz cinsinden frekans değeri vermelidir.

2.4 Üret- Butonlar ile Piyano Yapımı

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin yukarıdaki devreye ilişkin olarak aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.

```

piano
#define dob 2
#define reb 3
#define mib 4
#define fab 5
#define solb 6
#define lab 7
#define sib 8
#define idob 9
#define buzzer 10

int drm_do = 0;
int drm_re = 0;
int drm_mi = 0;
int drm_fa = 0;
int drm_sol = 0;
int drm_la = 0;
int drm_si = 0;
int drm_ido = 0;

void setup() {
  pinMode(dob, INPUT);
  pinMode(reb, INPUT);
  pinMode(mib, INPUT);
  pinMode(fab, INPUT);
  pinMode(solb, INPUT);
  pinMode(lab, INPUT);
  pinMode(sib, INPUT);
  pinMode(idob, INPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  drm_do = digitalRead(dob);
  drm_re = digitalRead(reb);
  drm_mi = digitalRead(mib);
  drm_fa = digitalRead(fab);
  drm_sol = digitalRead(solb);
  drm_la = digitalRead(lab);
  drm_si = digitalRead(sib);
  drm_ido = digitalRead(idob);

  if (drm_do == HIGH){
    tone(buzzer, 262, 100);
  }
  if (drm_re == HIGH){
    tone(buzzer, 294, 100);
  }
  if (drm_mi == HIGH){
    tone(buzzer, 330, 100);
  }
  if (drm_fa == HIGH){
    tone(buzzer, 349, 100);
  }
  if (drm_sol == HIGH){
    tone(buzzer, 392, 100);
  }
  if (drm_la == HIGH){
    tone(buzzer, 440, 100);
  }
  if (drm_si == HIGH){
    tone(buzzer, 494, 100);
  }
  if (drm_ido == HIGH){
    tone(buzzer, 523, 100);
  }
}

```

Resim 2.26: Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün tasarlayıp geliştirdikleri soldan sağa-sağdan sola yanıp sönen LED ve butonlar ile piyano uygulamaları için sınıf içerisinde öğrenciler seçilir ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

- Devreler oluşturulurken neden farklı dirençler kullanıldı?
- Devrelerin fiziki kurulumunda en fazla nerelerde zorlandınız?
- Program algoritmalarını oluştururken hangi süreçlerde zorlandınız?
- Arduino IDE içerisinde yazılan kodlarda “#for” deyiminin kullanılmasının avantajları nelerdir?
- Arduino IDE içerisinde yazılan kodlarda “#if-else” deyiminin kullanılmasının avantajları nelerdir?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşmaya kadar devam ettirilebilir.

3. Bölüm – Diyot ve LED Uygulamaları

Ön bilgi:

- Öğrenciler algoritma mantığını bilir.
- Öğrenciler temel programlama kavramlarını bilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.
- Öğrenciler Arduino UNO'yu tanır ve bunu kullanarak devreleri kurup kodlarında düzenleme yapabilirler.

Bölüm Kazanımları:

- Öğrenciler diyotun çalışma prensiplerini bilir.
- Öğrenciler “ve (and)” kapısını ve veya (or)” kapısını bilir ve bunlarla devre tasarlayabilir.
- Öğrenciler avometre ile direnç, akım ve voltaj değerlerini ölçebilir.
- Öğrenciler avometre ile bir diyotun sağlamlığını kontrol edebilir.
- Öğrenciler LED'in çalışma mantığını bilir ve bunu kullanarak özgün tasarımlar yapabilir.
- Öğrenciler RGB LED'in çalışma mantığını bilir ve bu devre elamanını kullanarak devre tasarlayabilir.
- Öğrenciler Arduino IDE ile değişkenleri, döngüleri ve fonksiyonları kullanabilir.
- Öğrenciler temel elektronik devre elemanlarının bağlantılarını yapabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin Arduino UNO mikrodenetleyicisini kullanarak diyotun çalışma prensipleri anlamaları ve bu devre elamanını kullanarak dijital elektronikteki temel kavramlardan olan kapı devrelerine yönelik bağlantıları yapabilmelerini ve Arduino IDE yazılımı ile bu devrenin istenen şekilde programlanabilmesini sağlamaktır. Ardından daha karmaşık LED ve RGB LED devrelerin tasarlanması ve Arduino IDE ile programlanması sayesinde yazılımsal temellerini geliştirmek hedeflenmektedir.

Göze ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	Kırmızı LED
Breadboard	Diyot
100 ohm direnç	Bağlantı kablosu
220 ohm direnç	Avometre
330 ohm direnç	Potansiyometre
560 ohm direnç	Buzzer
10 Kohm direnç	Buton
RGB LED	

Haftanın İşlenişi:

Göze: Diyotun ve RGB LED'in çalışma mantığını farklı uygulamalar ve açıklamalarla zenginleştirerek bunu Arduino UNO ve yazılım arayüzü olan Arduino IDE ile bütünleşik olarak farklı amaçlar için kullanılması hedeflenmektedir.

Uygula: LED ve RGB LED kullanarak farklı amaçlarla tasarlanacak olan örnek devrelerin oluşturulması ve programlanması. Bu aşamada kodlama sürecinde değişken, döngü ve fonksiyon kullanımı pekiştirilmelidir.

Tasarla: Devre tasarımı öncesinde Arduino UNO üzerinde ilgili problemi çözmek için gerekli devrenin şematik olarak hazırlanması ve devreyi uygun şekilde çalıştıracak algoritmanın tasarlanması, kodunun yazılması.

Üret: Arduino UNO mikrodenetleyicisinin dijital çıkış pinlerinin devre tasarımlarının yapılması. Farklı LED uygulamalarının Arduino IDE kullanılarak kodlanması.

Değerlendir: Yansıtma Etkinliği.

1.ADIM: GÖZLE ve UYGULA

1.1 Gözle – Yarı İletkenler

Maddeler birçok şekilde sınıflandırılabilir gibi akım iletme özelliğine göre de sınıflandırılabilir. İki ucuna potansiyel farkı uygulanan bir madde eğer elektrik akımını iletirse buna “iletken”; eğer akımı iletmiyor ise 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 adlandırılırlar. Öte yandan bazı maddeler ise elektrik akımına karşı neredeyse hiç direnç göstermezler. Bu nedenle “iletken” olarak sınıflandırılırlar. Plastik ve tahta parçası gibi bazı maddeler neredeyse elektriği hiç iletmedikleri için yalıtkanlara örnek olarak verilebilirler. Metaller ve tuzlu su iletken olarak sınıflandırılabilirler, çünkü elektrik akımına karşı çok az direnç gösterirler ve elektriği iletirler. 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 taşırlar. Belli durumlarda iletken gibi davranırken, belirli durumlarda ise yalıtkan gibi davranırlar. Örneğin yalıtkan gibi görünen bazı maddelere ısı verildiğinde bu maddelerin elektriği iletmediği görülür. Benzer şekilde bazı maddeler diğer hallerde yalıtkan gibi davranırken üzerine ışık düştüğünde elektrik akımını iletirler. Arduino UNO kartları ile de kullanılabilen 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. Bunlar arasında diyot ve transistörü sayabiliriz. Bu hafta diyot uygulamaları ile çalışmalarımıza devam edeceğiz. Gelecek hafta ise transistörü tanıyıp onunla ilgili temel uygulamalar yapacağız.

1.2 Gözle – Diyot

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 iç yapısı gösterilmiştir. Mavi alan “P” tipi yarı iletkeni beyaz alan ise “N” tipi yarı iletkeni temsil etmektedir. İç yapısı şekilde verilen bir diyotun iki tarafında birer bağlantı noktası vardır. Bağlantı noktalarından birine anot, diğerine ise katot adı verilir.



Resim 3.1: Diyot Bağlantı Noktaları

Anot bağlantı noktasına “+” ve katot bağlantı noktasına “-” yüklü gerilim uygulandığında diyot iletken hale gelir ve akım geçirmeye başlar. Aşağıdaki resimde diyot akım yönü görülmektedir.



Resim 3.2: Diyot Akım Yönü

Bu durum yukarıdaki 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 olarak (Silisyum~0.7 V veya Germanyum~0.2 V) eşik değeri aşıldığı an diyot iletken gibi hareket etmeye başlar. Bu eşik değerinin altında uygulanan gerilimlerde aslında iletim gerçekleşmez. Aşağıdaki resimde diyot kapalı anahtar akım yönü görülmektedir.



Resim 3.3: Diyot Kapalı Anahtar Akım Yönü

Ancak bunun tersi olduğunda yalıtkan gibi davranır. Yani anot bağlantı noktasına “-” ve katot bağlantı noktasına “+” yüklü gerilim uygulanırsa diyot akımı iletmeyecektir. Aşağıdaki resimde diyot akım iletmeme durumu görülmektedir.



Resim 3.4: Diyot Akım İletmeme Durumu

Bu durum aşağıdaki resimde olduğu gibi gösterilebilir. Diyotun anot bacağına “-”, katot bacağına “+” uygulanırsa, diyot devreden aşağıdaki resimdeki gibi akım geçmesine izin veremeyecektir.



Resim 3.5: Diyot Açık Anahtar (Akım İletmez)

Diyotun tek yönde akım geçirmesi ve diğer yönde akım geçirmemesi özelliği bu devre elemanının sağlamlık kontrolünde kullanılmaktadır. Diyotların **avometre** (ya da multimetre; bu ders kapsamında ikisi de dönüşümlü olarak kullanılacaktır) ile sağlamlık kontrolünü yapabilmek için avometre diyot kademesine alınır. Avometrenin **kırmızı ucu** diyotun anotuna, **siyah ucu** ise diyotun katotuna bağlandığında çok düşük bir direnç değeri okunmalıdır. Tam tersi şekilde bağlandığında ise avometre bize 700-800 ohm civarı bir değer göstermelidir. Eğer bu şekilde ölçüm yapılırsa o diyot sağlam demektir. Sağlamlık testi kurulan bazı devrelerde beklenen sonuçlar alınmazsa devre elemanlarının tek tek ölçülerek sağlamlığına bakılması

gerekir. Örneğin, eski dijital göstergeli arabaların kadranlarında ya da eski dijital display barındıran aletlerde bütün karakterler hafif görünür hale gelmesinin temel nedeni elektronik kartta bulunan diyotların zamanla bozularak ters yönde de akım geçirmesidir.

1.3 Gözle ve Uygula - Avometre ile direnç, gerilim ve akım ölçüyorum (Öğrenci 1)

Malzeme Listesi
Avometre
Kırmızı LED
330 ohm direnç
Potansiyometre
Breadboard

Avometre belirli noktalar arasındaki akım, gerilim ve direnç değerlerini ölçebilen bir araçtır. Aşağıdaki resimde avometre görülmektedir.



Resim 3.6: Avometre

Avometre elektronik devrelerde farklı amaçlarla ölçümler yapmak için kullanılır. Örneğin bir direncin değerinin üzerindeki renk kodları ile verilen değerde olup olmadığını avometrenin direnç ölçme özelliği ile kontrol edilebiliriz. Bunun için eğitmen aşağıdaki açıklamaları yaparak avometre ile ölçümleri önce kendisi yapar, ardından öğrenciler tarafından yapılmasına yardımcı olur. Avometre “Ω” işareti ile gösterilen kademede ölçülecek direncin büyüklüğüne göre ayarlanır. Pozitif ve negatif uçlara bağlı probalar (ölçüm uçları) direncin iki ucuna değiştirilerek avometrenin üzerindeki değer okunur. Örnek olarak devrede LED’lere takılı olan dirençlerden biri çıkarılır ve bahsedildiği şekilde direnç değeri ölçülür. Ölçülen değer 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 avometre ile ölçülürse 0 değerinin okunması gerekir. Akım ölçmek için avometre “ μA ” kademesine getirilir. Avometrenin bir noktadan geçen akımı ölçebilmesi için akımın avometre üzerinden geçmesi gerekir yani devreye seri bağlanması gerekmektedir. Öncelikli olarak multimetrenin “-“ ucu breadboardun GND bağlı şeridine bağlanır. “+” uç ise potansiyometrenin sol ucuna bağlanır. Potansiyometrenin sol ucunun GND bağlantısı yapılır. Ardından Arduino’ya güç verilir. Potansiyometrenin ayarı değiştirilerek avometrenin okuduğu değerler gözlenir.

Gerilim ölçmek için ise avometre bir devrenin iki noktasına bağlanır ve bu iki nokta arasındaki potansiyel farkı ölçer. Bunu ölçebilmek için avometre devreye paralel bağlanmalıdır. Devrede avometre bağlantısı değiştirilmeden “V” ile gösterilen gerilim ölçme ayarına getirilir. Bu şekilde potansiyometrenin sol ucu ile orta ucu arasındaki gerilim farkı ölçülecektir. Devreye güç verilir. Avometre üzerindeki değer okunur ve ölçümler üzerine konuşulur.

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 en düşük kademede olması gerekir. Çok büyük kademelerde tam sonuç elde edilemezken, düşük kademelerde ise okuma hiç yapılamaz. Aşağıdaki resimlerde 330 ohm değerindeki direncin yüksek, ideal ve düşük kademelerdeki ölçümlerine ait ekran görüntüleri yer almaktadır.



Resim 3.7: Avometre Direnç Ölçümü

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

1.4 Uygula- Diyot (Öğrenci 2)

Malzeme Listesi

Avometre

Diyot

Bu etkinlikteki amaç öğrencilerin ellerindeki setlerde var olan diyotlardan birinin sağlamlık testini gerçekleştirmesidir. Bu uygulamada eğitmen öğrencilerden bir diyot almalarını ve Anot –Katot uçları arasındaki direnci avometre ile ölçmelerini ister. Bu ölçüm ile öğrencilerin diyotun Anot-Katot uçlarını belirlemeleri ve aynı zamanda sağlamlık kontrolünü yapmaları beklenir. Gerekli durumlarda eğitmen öğrencilerin ölçüm süreçlerini değerlendirerek her grubun uygulamayı başarılı bir şekilde gerçekleştirilmesine yardımcı olur.

1.5 Gözle ve Uygula – Buzzer diyot uygulaması (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Diyot
Buzzer
100 ohm direnç
Breadboard
Bağlantı kabloları

Birçok farklı türde diyot vardır. Aşağıda temel bir diyotun görseli 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.



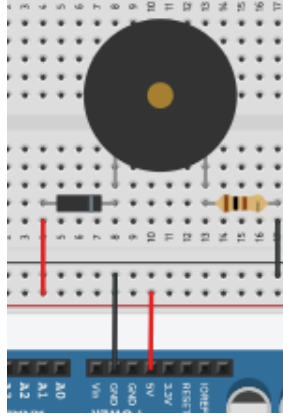
Resim 3.8: Diyot Bağlantı Noktaları

Resimdeki basit diyotun dışında farklı türde diyotlar da vardır.

- Kristal diyot
- Zener diyot
- Tünel diyot
- LED
- Foto Diyot
- Ayarlanabilir Kapasiteli Diyot.

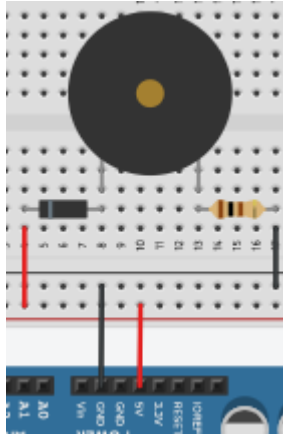
Aslında şimdiye kadar sıklıkla kullanılan LED'ler de bir çeşit diyottur. Basit diyotun taşıdığı tüm özellikleri barındırır. Tek farkı iletken olduğu durumda basit diyottan farklı olarak, aynı zamanda, bir ışık yayar. **LED kelimesi "light emitting diode" (ışık yayan diyot) kelimelerinin baş harflerinden oluşmaktadır.** Önceki derslerimizdeki LED'li devreler hatırlanacak olursa, LED'imizi ters bağladığımızda devremizin çalışmamasının sebebi de aslında LED'in bir diyot olarak işlev görmesi, yani akımı tek yönde iletmesidir.

Diyotların çalışma prensiplerini anlamak için aşağıdaki devre eğitmen tarafından öğrencilere anlatılarak kurulur. Bu devrede Arduino "5 V" güç kaynağı olarak kullanılmıştır. Yani Arduino'nun mikrodenetleyici özellikleri kullanılmamıştır. Arduino yerine 5 V güç kaynağı kullanılırsa da devre aynı şekilde çalışacaktır. Öğrencilere Arduino'nun güç kaynağı olarak kullanıldığı eğitmen tarafından tekrar vurgulanır. Bu devrede diyot uygun şekilde bağlandığı için aktif buzzera akım gitmiş ve buzzer bip sesini çalmıştır. Buzzerdan akım geçerken eğitmen avometre ile buzzerdan geçen akımı ölçer ve öğrencilerin gözlemlemesini sağlar. Daha sonra öğrencilerden devreyi kurarak ölçüm yapmalarını ister. Akım ölçümünü için avometre devreye seri bağlanmalıdır. Bu işlemde setlerdeki krokodiller de kullanılabilir. Aşağıdaki resimde örnek devre şeması görülmektedir.



Resim 3.9: Gözle-Uygula Etkinliği Örnek Devre Şeması

Bu uygulamadan sonra eğitmen öğrencilerden kurmuş oldukları devrede diyotun yönünü ters çevirerek devreyi yeniden düzenlemelerini 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. Bu şekilde buzzerdan akım geçmez eğitmen öğrencilerden avometre ile buzzerdan geçen akımı ölçmelerini ister.



Resim 3.10: Gözle-Uygula Etkinliği Örnek Devre Şeması

1.6 Gözle ve Uygula - Diyot ile “VE” Kapısı Yapıyorum (Öğrenci 2)

Malzeme Listesi

Arduino UNO

Diyot

Kırmızı LED

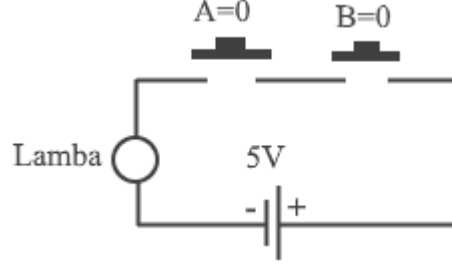
220 ohm direnç

Breadboard

Bağlantı kabloları

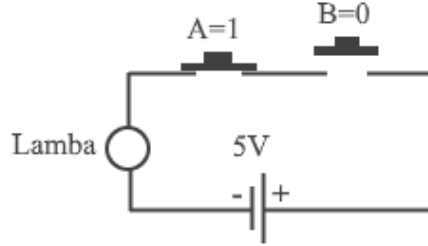
Elektronikte aritmetik (toplama, çıkarma, çarpma ve bölme) ve mantıksal (“VE”, “VEYA” ve “DEĞİL”) işlemler önemlidir. Bu işlemler kullanılarak karmaşık işlevler gerçekleştirilebilir. Temel olarak Arduino mikrodenetleyicisi de bu işlemleri gerçekleştirir. Mantıksal işlemler elektronikte kapılar vasıtasıyla yapılır. “VE”, “VEYA” ve “DEĞİL” üç temel kapıdır. Bu

kapılar kullanılarak diğer bütün kapılar elde edilebilir. **VE** kapısı elektronikte kullanılan temel kapılardan birisidir. VE kapısını arka arkaya bağlanmış iki anahtardan oluşan devreye benzetebiliriz. Aşağıdaki resimde görüldüğü gibi anahtarların her ikisi de açık konumdayken lamba yanmaz. Yani lamba üzerinde akım geçmediği için değeri **“0” (OFF)** olur.



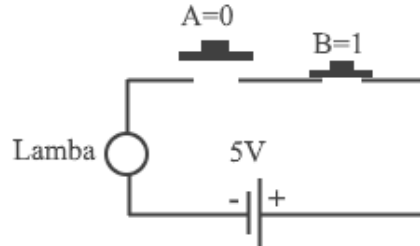
Resim 3.11: "VE" Kapısı Devresi

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 devresi görülmektedir.



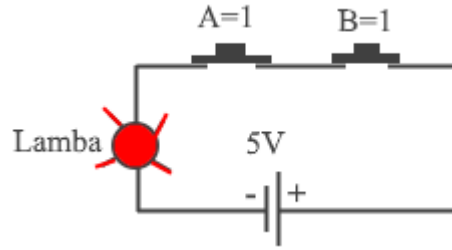
Resim 3.12: "VE" Kapısı Devresi

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



Resim 3.13: "VE" Kapısı Devresi

Aşağıdaki resimde görüldüğü gibi A ve B anahtarının her ikisinin kapalı olduğu durumdaysa ($A=1, B=1$) devreden akım geçer ve lamba yanar. Yani lambanın değeri **“1” (ON)** olur.

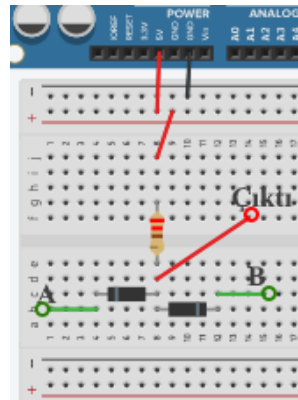


Resim 3.14: "VE" Kapısı Devresi

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

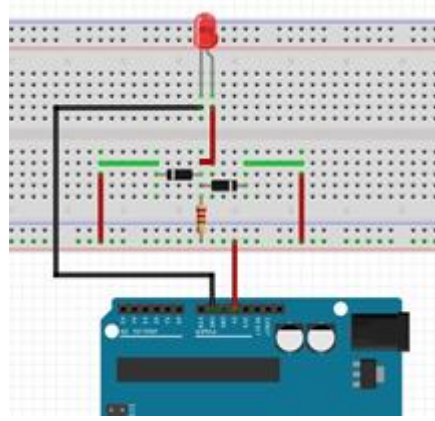
A	B	Çıktı
0	0	0
0	1	0
1	0	0
1	1	1

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 değişecektir.



Resim 3.15: "VE" Kapısı Devre Şeması

Çıktı ucuna bir LED takılırsa devrenin çalışması daha kolay anlaşılır. A ve B ucuna "1" (5 V) değerleri verildiğinde LED yanar, yani Çıktı değeri "1" (5 V) olur. Bunun nedeni A ve B uçlarına "1" değeri verildiğinde her iki diyot da yalıtkan olur. Diyotlar akım iletmediği için akımın iletileceği tek yol "Çıktı"dır. Yani "Çıktı" değeri "1" olmuştur. Aşağıdaki resimde "VE" kapısı devre şeması görülmektedir.



Resim 3.16: "VE" Kapısı Devre Şeması

Eğitmen 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 eğitmen 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."

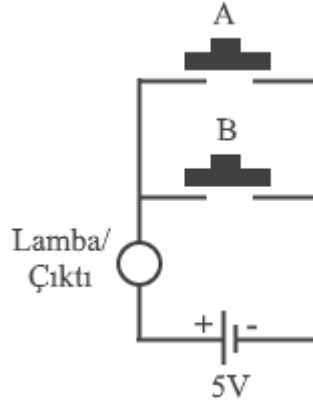
1.7 Gözle ve Uygula- Kaç Buzzer'dan Ses Çıkıyor? (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Buton
Aktif buzzer
Kırmızı LED
100 Ohm Direnç
330 Ohm Direnç
10K Ohm Direnç
Breadboard
Bağlantı kabloları

Bu etkinlikte amaç ses çıkan 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 işlem aşağıdaki şekilde gösterilen devre ile yapılacaktır.

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

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

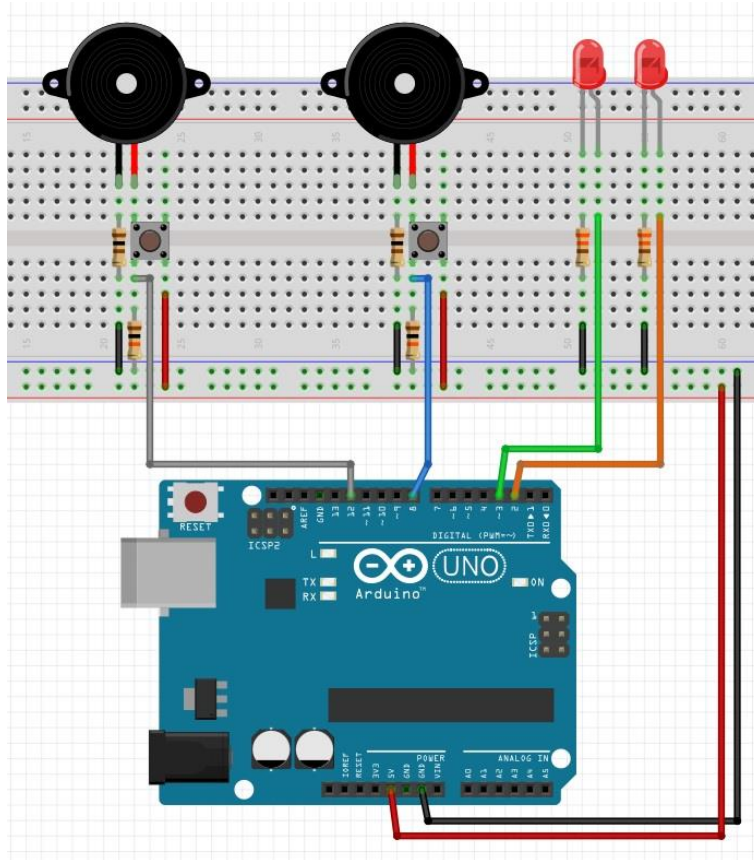


Resim 3.17: "VEYA" Kapısı Devresi

A=0 ve B=0 iken devreden akım akmaz ve böylece lamba yanmaz. Yani çıktı "0" olmuştur. A=1, B=0 iken A butonu üzerinden akım akarak 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 devre tasarımı aşağıdaki resimde gösterilmiştir. Sol tarafta iki adet buton ve iki adet 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 sol butona basıldığında 12 numaralı dijital pine ve sağ butona basıldığında 8 numaralı dijital pine 1 değeri gönderilmiş olunur. Butonlara basılmadığında bu pinlere 0 değeri gönderilir.



Resim 3.18: Gözle-Uygula Etkinliği Örnek Devre Şeması

12 ve 8 numaralı dijital pinlere gelen değerlere göre LED'ler yakılacaktır. 12 numaralı pin ve 8 numaralı pinin her ikisine birden 1 bilgisi gittiğinde her iki LED yanacaktır. 12 numaralı pin veya 8 numaralı pinden sadece bir tanesinden (hangisi olduğu farketmeksizin) 1 değeri gittiğinde sol taraftaki LED yanacaktır. Yani sadece 12 numaralı pinden veya 8 numaralı pinden 1 değeri geldiğinde sol LED yanacaktır. Eğer her iki pinden de 1 değeri gelmezse (0 değeri geldiğinde) LED'lerin ikisi de yanmayacaktır. Bu işlemi gerçekleştiren kod aşağıdaki resimde görüldüğü şekildedir. Eğitmen bu kodu yazarak öğrencilere anlatır ve devrenin çalışmasını öğrencilere gösterir. Burada eğitmenin öğrencilere “ve” yapısının “&&” işareti ile sağlandığını; “veya” yapısının ise kod içerisinde “||” işareti ile sağlandığını belirterek bu kodların nasıl çalıştığını açıklamalıdır. Ardından öğrencilerden aynı devreyi ve kodu yazarak devreyi çalıştırmalarını ister.

```

1 const int BUTTON1 = 12;
2 const int BUTTON2 = 8;
3 const int LED1 = 3;
4 const int LED2 = 2;
5 int BUTTONstate1 = 0;
6 int BUTTONstate2 = 0;
7
8 void setup() {
9   pinMode(BUTTON1, INPUT);
10  pinMode(BUTTON2, INPUT);
11  pinMode(LED1, OUTPUT);
12  pinMode(LED2, OUTPUT);
13
14 }
15
16 void loop() {
17   BUTTONstate1 = digitalRead(BUTTON1);
18   BUTTONstate2 = digitalRead(BUTTON2);
19   if(BUTTONstate1 == HIGH && BUTTONstate2 == HIGH) {
20     digitalWrite(LED1, HIGH);
21     digitalWrite(LED2, HIGH);
22   }
23   if(BUTTONstate1 == HIGH || BUTTONstate2 == HIGH) {
24     digitalWrite(LED1, HIGH);
25     digitalWrite(LED2, LOW);
26   }
27   else{
28     digitalWrite(LED1, LOW);
29     digitalWrite(LED2, LOW);
30   }
31 }
32

```

Derleme tamamlandı.

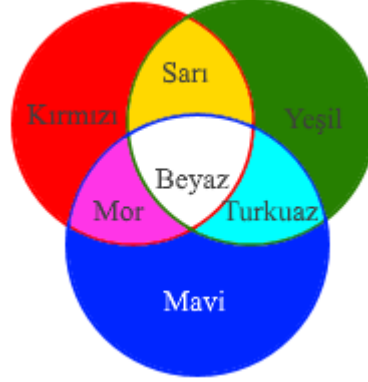
Resim 3.19: Gözle-Uygula Etkinliği Örnek Kodu

1.8 Gözle: RGB LED ile Farklı Renkler

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 denilir. Bu üç rengin üçü de aynı miktarda karıştırarak beyaz, kırmızı ve mavi rengi karıştırarak mor renk, kırmızı ile yeşili karıştırarak sarı renk, mavi ile yeşili karıştırarak Turkuaz rengi elde edilebilir. Bu renkler aşağıda resimde gösterilmiştir.

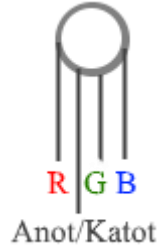
Dikkat

Günlük hayatta ana renkler kırmızı, sarı ve mavi renkler olarak kabul edilmiştir. Ancak ışık renkleri olarak RGB sisteminde ana renkler farklıdır. Eğitmen bu farkı vurgular.



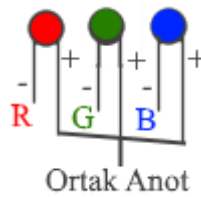
Resim 3.20: Ana ve Bazı Ara Renkler

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



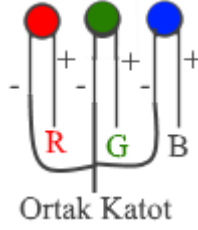
Resim 3.21: RGB LED Ayak Yapısı

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.



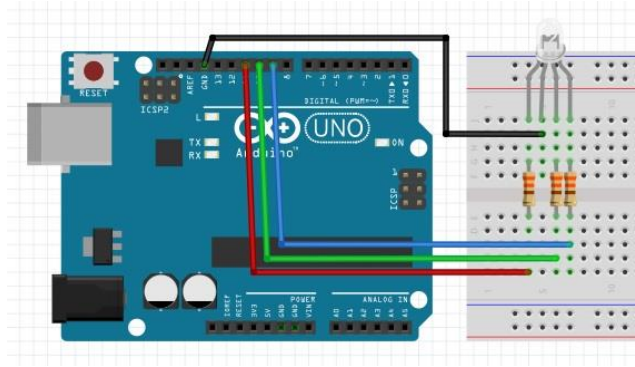
Resim 3.22: Ortak Anot RGB LED

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



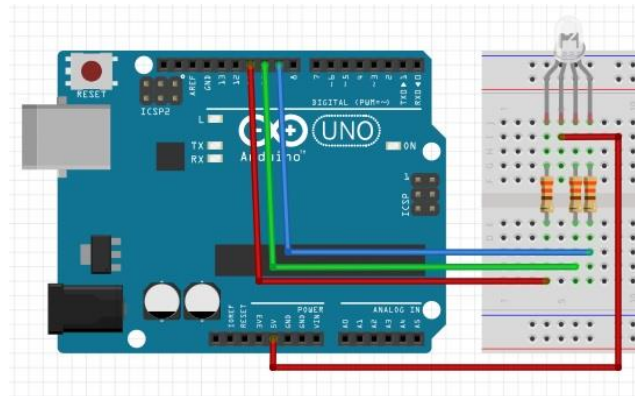
Resim 3.23: Ortak Katot RGB LED

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



Resim 3.24: Gözle Etkinliği Örnek Devre Şeması

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



Resim 3.25: Gözle Etkinliği Örnek Devre Şeması

Yukarıdaki her iki bağlantıda da özellikle 9, 10 ve 11 numaralı pinler seçilmiştir. Bu pinlerin yanında “~” (tilde) işareti bulunmaktadır. Yanında “~” işareti bulunan pinlere önceki haftalarda da anlatıldığı gibi PWM pini denilir. Bütün renk tonlarının elde edilebilmesi için RGB bağlantısının PWM pinleri ile yapılması gerektiği bilinmelidir. Bu eğitim tarafından vurgulanmalıdır.

Ortak **katot** bir RGB’de kırmızı rengi yakmak için kod kullanılabilir. Ortak **anot** bir RGB LED’in kullanıldığı durumlarda 11. pin için “OFF”, 10. pin için “ON” ve 9. pin için “ON” değerleri verilir.

Dikkat

Bu örnekte pinlerin “0” ya da “1” olması, “OFF” ya da “ON” olarak açıklanmıştır. Bundan sonraki konularda ve derslerde “0, LOW” ya da “OFF” ve “1”, “HIGH” ya da “ON” dönüşümlü olarak kullanılacaktır.

Aşağıdaki resimde örnek kod görülmektedir.

```

RGB_LED_1
1  int red_light_pin= 11;
2  int green_light_pin = 10;
3  int blue_light_pin = 9;
4  void setup() {
5      pinMode(red_light_pin, OUTPUT);
6      pinMode(green_light_pin, OUTPUT);
7      pinMode(blue_light_pin, OUTPUT);
8  }
9  void loop() {
10     digitalWrite(red_light_pin, 1);
11     digitalWrite(green_light_pin, 0);
12     digitalWrite(blue_light_pin, 0);
13 }
14 |

```

Resim 3.26: Gözle Etkinliği Örnek Kodu

Ortak **katot** bir RGB’de yeşil rengi yakmak için aşağıdaki resimde görüldüğü gibi bir kod kullanılabilir. Ortak **anot** bir RGB LED’in kullanıldığı durumlarda 11. pin için ON, 10. pin için OFF ve 9. pin için ON değerleri verilir.

```

RGB_LED_2
1 int red_light_pin= 11;
2 int green_light_pin = 10;
3 int blue_light_pin = 9;
4 void setup() {
5   pinMode(red_light_pin, OUTPUT);
6   pinMode(green_light_pin, OUTPUT);
7   pinMode(blue_light_pin, OUTPUT);
8 }
9 void loop() {
10  digitalWrite(red_light_pin, 0);
11  digitalWrite(green_light_pin, 1);
12  digitalWrite(blue_light_pin, 0);
13 }
14 |

```

Resim 3.27: Gözle Etkinliği Örnek Kodu

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

```

RGB_LED_3
1 int red_light_pin= 11;
2 int green_light_pin = 10;
3 int blue_light_pin = 9;
4 void setup() {
5   pinMode(red_light_pin, OUTPUT);
6   pinMode(green_light_pin, OUTPUT);
7   pinMode(blue_light_pin, OUTPUT);
8 }
9 void loop() {
10  digitalWrite(red_light_pin, 0);
11  digitalWrite(green_light_pin,0);
12  digitalWrite(blue_light_pin, 1);
13 }
14 |

```

Resim 3.28: Gözle Etkinliği Örnek Kodu

Üç temel renk karıştırılarak istenilen renk elde edilebilir. Örneğin ortak katot bir RGB LED'de sarı renk için aşağıdaki resimde görülen kod dizimi kullanılabilir.

```

RGB_LED_4
1 int red_light_pin= 11;
2 int green_light_pin = 10;
3 int blue_light_pin = 9;
4 void setup() {
5   pinMode(red_light_pin, OUTPUT);
6   pinMode(green_light_pin, OUTPUT);
7   pinMode(blue_light_pin, OUTPUT);
8 }
9 void loop() {
10  digitalWrite(red_light_pin, 1);
11  digitalWrite(green_light_pin,1);
12  digitalWrite(blue_light_pin, 0);
13 }

```

Resim 3.29: Gözle Etkinliği Örnek Kodu

1.9 Uygula- Ortak Ucu Belirleme (Öğrenci 2)

Eğitmen 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 çözümleri bulmaları/denemeleri sağlanmalıdır. Öğrencilere direnç kullanmadan RGB LED’leri güç kaynağına bağlamamaları gerektiği hatırlatılmalıdır.

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

1.10 Gözle - Uygula: Arduino IDE ile Ara Renkler (Öğrenci 1)

Malzeme Listesi
Arduino UNO
RGB LED
330 Ohm Direnç
Breadboard
Bağlantı kabloları

Yukarıdaki örneklerde RGB LED üzerinde üç ana renk olan kırmızı, yeşil ve mavi ve bunun yanında ara renkler olan sarı oluşturuldu. Aslında bu renkler dışında bulunan ara renkler de RGB LED ile yakılabilir. Bunun için PWM pinlerden analog sinyal alma yöntemi ile renkleri oluşturmak mümkündür. Bu amaçla aşağıdaki örnekteki gibi PWM çıkışları olan 9,10 ve 11. pinler analog çıkış almak için “OUTPUT” olarak tanımlanır. Böylece her bir pine bağlı olan RGB LED’in ayaklarına gelen analog veri değiştirilerek renk kombinasyonları elde edilebilir.

Bu etkinlikteki amaç bir RGB LED’le mor ve turkuaz ışık renkleri oluşturmaktır. Eğitmen öğrencilerden mor ve turkuaz renkler için fonksiyon yazmalarını ve bu fonksiyonları çağırarak çalıştırmalarını ister. Aşağıdaki resimde görülen örnek kod ortak katot için oluşturulmuş bir örnektir. Öğrenciler isterlerse ortak anotlu devre için de kod yazabilirler.

```

RGB_LED_5 §
1 int red_light_pin= 11;
2 int green_light_pin = 10;
3 int blue_light_pin = 9;
4 void setup() {
5     pinMode(red_light_pin, OUTPUT);
6     pinMode(green_light_pin, OUTPUT);
7     pinMode(blue_light_pin, OUTPUT);
8 }
9 void loop() {
10
11
12     purple ();
13     delay(1000);
14     turquois();
15     delay(1000);
16 }
17
18 void purple()
19 {
20     analogWrite(red_light_pin,128);
21     analogWrite(green_light_pin,0);
22     analogWrite(blue_light_pin,128);
23 }
24 void turquois()
25 {
26     analogWrite(red_light_pin,64);
27     analogWrite(green_light_pin,224);
28     analogWrite(blue_light_pin,208);
29 }

```

Resim 3.30: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

RGB LED'in bacakları **PWM** pinlerine bağlanmalıdır. Aksi takdirde analog çıkış alınamayacağı için RGB LED'in ayaklarına istenilen veri aktarılamaz.

1.11 Uygula- Gökkuşağındaki 7 Temel Renk (Öğrenci 2)

Malzeme Listesi
Arduino UNO
RGB LED
330 Ohm Direnç
Breadboard
Bağlantı kabloları

Bu etkinlikteki amaç birer saniye arayla RGB LED'in sırayla kırmızı, yeşil, mavi, sarı, yeşil, lacivert, mor ve beyaz renkleri yakmasını sağlamaktır. Bu amaç "RGB_color" isimli bir fonksiyon tanımlayarak ve loop fonksiyonu içerisinde bu fonksiyona değer göndererek yazılabilir. Örnek kod aşağıdaki resimde görülmektedir.

RGB_LED_6

```

1 int red_light_pin= 11;
2 int green_light_pin = 10;
3 int blue_light_pin = 9;
4 void setup() {
5   pinMode(red_light_pin, OUTPUT);
6   pinMode(green_light_pin, OUTPUT);
7   pinMode(blue_light_pin, OUTPUT);
8 }
9 void loop() {
10  RGB_color(255, 0, 0); // Kırmızı
11  delay(1000);
12  RGB_color(0, 128, 0); // Yeşil
13  delay(1000);
14  RGB_color(0, 0, 255); // Mavi
15  delay(1000);
16  RGB_color(255, 255, 0); // Sarı
17  delay(1000);
18  RGB_color(0, 128, 0); // yeşil
19  delay(1000);
20  RGB_color(0, 0, 128); // lacivert
21  delay(1000);
22  RGB_color(80, 0, 80); // mor
23  delay(1000);
24  RGB_color(255, 255, 255); // Beyaz
25  delay(1000);
26 }
27 void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
28 {
29   analogWrite(red_light_pin, red_light_value);
30   analogWrite(green_light_pin, green_light_value);
31   analogWrite(blue_light_pin, blue_light_value);
32 }

```

Resim 3.31: Uygula Etkinliği Örnek Kodu

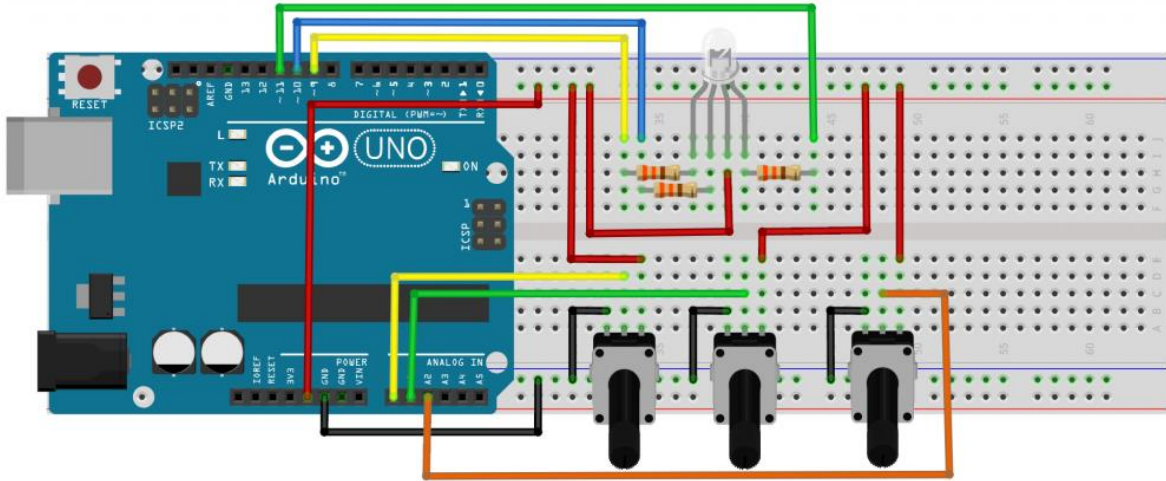
1.12 Gözle ve Uygula – 3 potansiyometre ile RGB LED kontrolü (Öğrenci 1)

Malzeme Listesi
Arduino UNO
RGB LED
10K ohm Direnç
220 ohm Direnç
Breadboard
Bağlantı kabloları

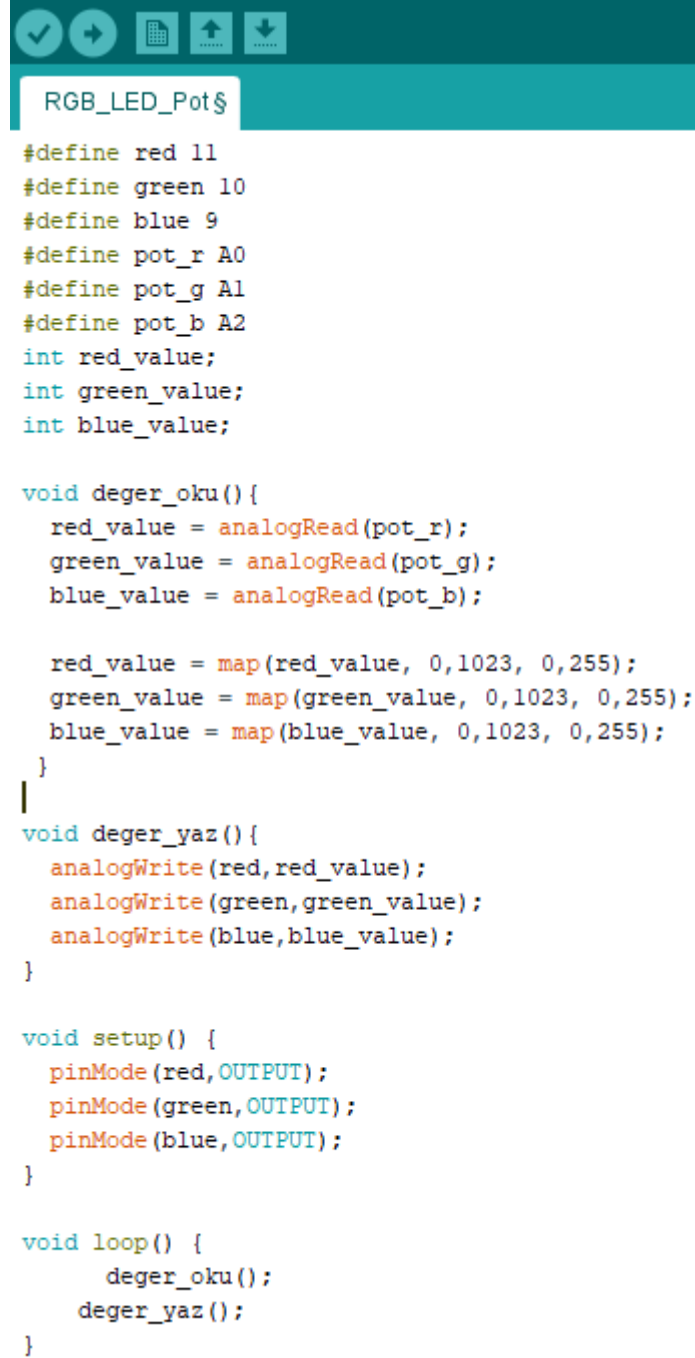
Bu uygulamada 3 adet potansiyometre kullanarak RGB LED'in renk değişimini kontrol edeceğiz. 3 potansiyometreden 1. potansiyometre ile kırmızı rengin'i, 2. potansiyometre ile yeşil rengin ve 3. potansiyometre ile de mavi rengin yanma seviyesini ayarlayarak istenen rengin RGB LED'de oluşturulması sağlanabilir. Örnek devre şeması ve kod aşağıdaki resimlerde görülmektedir.

Dikkat

Bunu yaparken “deger_oku” ve “deger_yaz” fonksiyonları tanımlanacak ve “loop” içerisinde bu fonksiyonlar çağırılarak bu işlemler gerçekleştirilecektir. Deger_oku fonksiyonu potansiyometrelerden gelen değerlerin okunması için tanımlanır. Ayrıca “map” fonksiyonunu ile 0-1023 arası okunan değerler 0-255 arası değerlere dönüştürülür. Değer_yaz fonksiyonu ise potansiyometrelerden alınan değerleri RGB LED'e aktarmak için tanımlanır.



Resim 3.32: Gözle-Uygula Etkinliği Örnek Devre Şeması



```

RGB_LED_Pot§

#define red 11
#define green 10
#define blue 9
#define pot_r A0
#define pot_g A1
#define pot_b A2
int red_value;
int green_value;
int blue_value;

void deger_oku() {
  red_value = analogRead(pot_r);
  green_value = analogRead(pot_g);
  blue_value = analogRead(pot_b);

  red_value = map(red_value, 0,1023, 0,255);
  green_value = map(green_value, 0,1023, 0,255);
  blue_value = map(blue_value, 0,1023, 0,255);
}

void deger_yaz() {
  analogWrite (red,red_value);
  analogWrite (green,green_value);
  analogWrite (blue,blue_value);
}

void setup() {
  pinMode (red,OUTPUT);
  pinMode (green,OUTPUT);
  pinMode (blue,OUTPUT);
}

void loop() {
  deger_oku();
  deger_yaz();
}

```

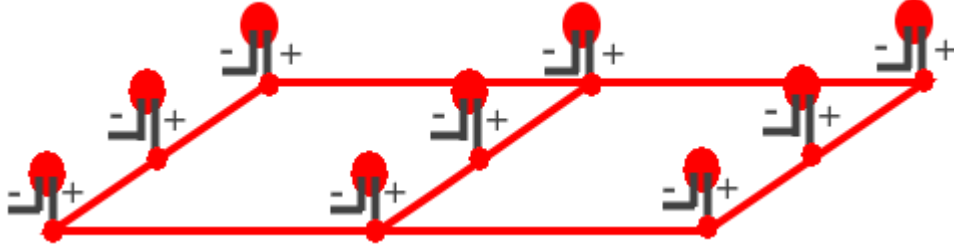
Resim 3.33: Gözle-Uygula Etkinliği Örnek Kodu

1.13 Gözle ve Uygula- LED Küpü Yapıyorum (Öğrenci 2)

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

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 şekilde verilmiştir. Şekilde görüldüğü gibi LED'lerin 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 3.34: Gözle-Uygula Etkinlik Şeması

Dikkat

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

Dikkat

Etkinlikte kullanılacak LED'lerin sağlamlıkları set ile verilen avometrelerin diyot sağlamlık ölçümü kısmını kullanarak test edilmeli ve sağlam olan LED'ler kullanılarak LED küpünün montajı yapılmalıdır.

Dikkat

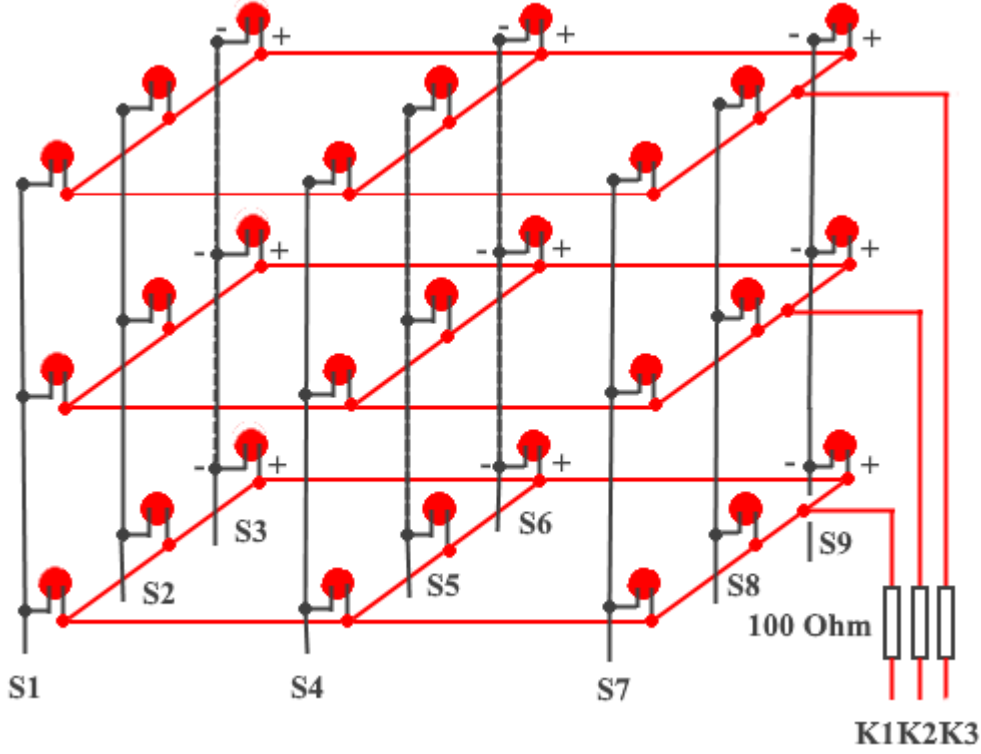
Etkinlikte **havya** kullanılırken eğitmenler herhangi bir yanık olayı ile karşılaşmamak için gerekli özeni göstermelidir.

Dikkat

Etkinlikte eğitmen gerek duyduğu takdirde LED'lerin birbiri ile olan bağlantılarını yapmaları için setlerle birlikte gelen kaideleri kullanabilir. Ama öncelikle LED küpü devresinin kurulumu LED'ler birleştirildikten sonra **delikli pertinaks** üzerine yapılacaktır.

Bu şekilde üç farklı kat oluşturulacaktır. İkinci katın yapımına geçmeden önce birinci kattaki devre elemanlarında bir sıkıntı olup olmadığı kontrol edilmelidir. Bunun için bir avometre (multimetre) veya güç kaynağı kullanılabilir. Avometre direnç ölçme moduna alınır. Avometrenin + bacağı LED'lerin + bacağına bağlı olduğu tele değdirilir. Avometrenin - bacağı yakılmak istenen LED'in - bacağına değdirilir. LED'lerin yanıp yanmadığı kontrol edilir ve tüm LED'lerin yandığından emin olunur. Bu test için avometre yerine bir güç kaynağı da kullanılabilir. Güç kaynağı kullanılacaksa eğitmen 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 boşa 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 yanmasını önlemek içindir. Şemada delikli pertinaks bulunmamaktadır fakat bu dirençler gerçek tasarımda delikli pertinaks üzerinde olacaktır.



Resim 3.35: Gözle-Uygula Etkinlik Şeması

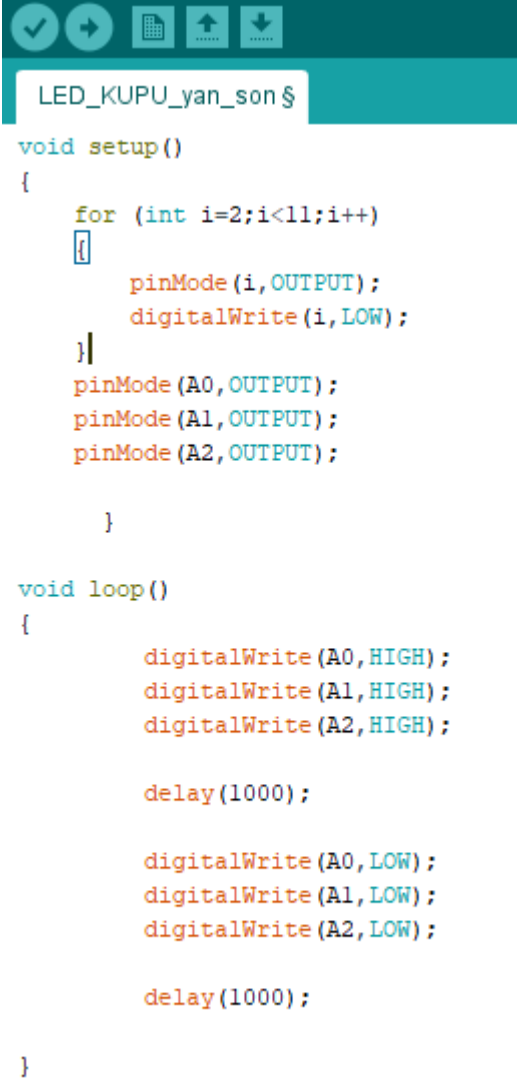
Şemadan da görüleceği gibi LED küpünde üç kat (K1, K2, K3) ve dokuz sütun bulunmaktadır (S1, S2, ..., S9). Bu sayede her bir LED ayrı ayrı yakılabilir. Örneğin birinci sütunun (S1) en üst katında bulunan LED'i yakmak için üçüncü kata (K3) 5 V (ON) ve birinci sütuna (S1) 0 Vt (OFF) verilmelidir. Eğitmen öğrencilere farklı LED'lerin ve LED gruplarının nasıl yakılabileceğine örnek 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 2, 3, ..., 10 numaralı dijital pinlerine (S1→2, S2→3, ..., S9→10 olacak şekilde) ve katlar ise Arduino'nun A1, A2, A3 pinlerine (K1→A0, K2→A1, K3→A2 olacak şekilde) bağlanır. Bu bağlantıların yapılmasının ardından artık LED küp ile ışık oyunları yapılabilir.

1.14 Gözle ve Uygula- LED Küpünü Kodluyorum (Öğrenci 1)

Bu etkinlikte 3x3x3 boyutunda bir LED küpü kullanılarak ışık oyunları yapılacaktır. LED küpünün bağlantısı sütunlar sırasıyla Arduino'nun 2, 3, ..., 10 numaralı dijital pinlerine karşılık gelecek şekilde (S1→2, S2→3, ..., S9→10) ve katlar ise Arduino'nun A0, A1, A2 pinlerine karşılık gelecek şekilde (K1→A0, K2→A1, K3→A2) yapılmalıdır.

İlk ışık oyunu bütün LEDlerin yakılıp söndürülmesi ile oluşturulacaktır. Bunun için önce katların hepsine “ON” (5 V) ve sütunların hepsine “OFF” (0 V) verilerek bütün LED’ler yakılır. Bilindiği gibi LED’lerin anot (+) bacağına 5 V ve katot (-) bacağına “0 V” verildiğinde LED yanacaktır. Bir saniye beklenir. Bunun ardından katların hepsine “OFF” (0 V) ve sütunların hepsine “ON” (5 V) verilerek bütün LED’ler söndürülür. Burada LED’leri yakmak için onların “+” ve “-“ bacaklarına verilen değerler ters bir şekilde LED bacaklarına uygulanmıştır (0 V → + bacak, 5 V → - bacak)değerlerin tersi LED bacaklarına verilmiştir. Örnek kod aşağıdaki resimde görülmektedir.



```

void setup()
{
  for (int i=2;i<11;i++)
  {
    pinMode(i,OUTPUT);
    digitalWrite(i,LOW);
  }
  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A2,OUTPUT);
}

void loop()
{
  digitalWrite(A0,HIGH);
  digitalWrite(A1,HIGH);
  digitalWrite(A2,HIGH);

  delay(1000);

  digitalWrite(A0,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);

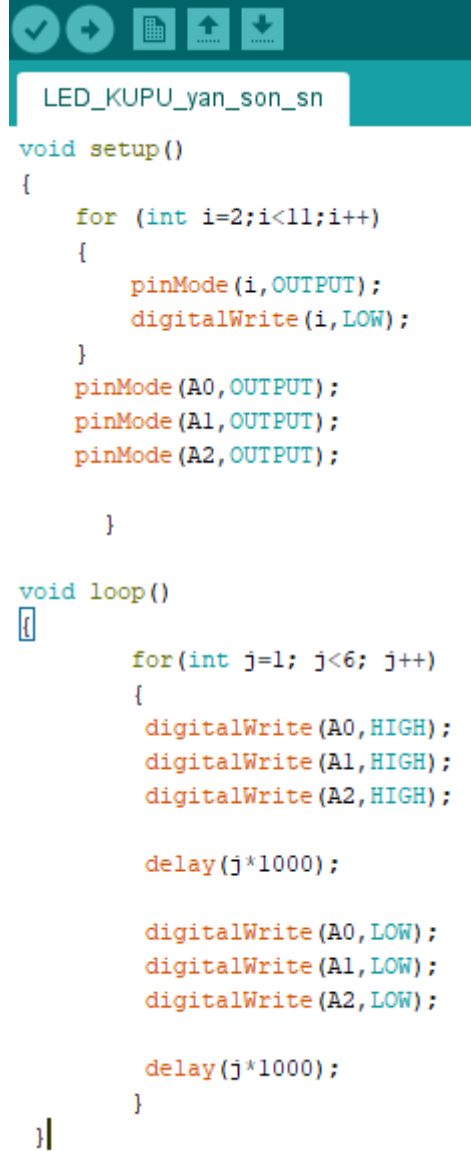
  delay(1000);
}

```

Resim 3.36: Gözle-Uygula Etkinliği Örnek Kodu

1.15 Gözle- Uygula LED Küpünü Farklı Aralıklarla Yakıp Söndürüyorum (Öğrenci 2)

Bundan önceki iki etkinlikte LED küpündeki LED’ler birer saniye aralıkla yakıp söndürüldü. Bu etkinlikte ise LED küpünün LED’lerinin yanıp sönmeleri arasındaki süre bir saniye, iki saniye, üç saniye, dört saniye; beş saniye, aralıklarla tekrar edecek şekilde ayarlanmalıdır. Örnek kod aşağıdaki resimde görülmektedir.



```

void setup()
{
  for (int i=2;i<11;i++)
  {
    pinMode(i,OUTPUT);
    digitalWrite(i,LOW);
  }
  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A2,OUTPUT);

}

void loop()
{
  for(int j=1; j<6; j++)
  {
    digitalWrite(A0,HIGH);
    digitalWrite(A1,HIGH);
    digitalWrite(A2,HIGH);

    delay(j*1000);

    digitalWrite(A0,LOW);
    digitalWrite(A1,LOW);
    digitalWrite(A2,LOW);

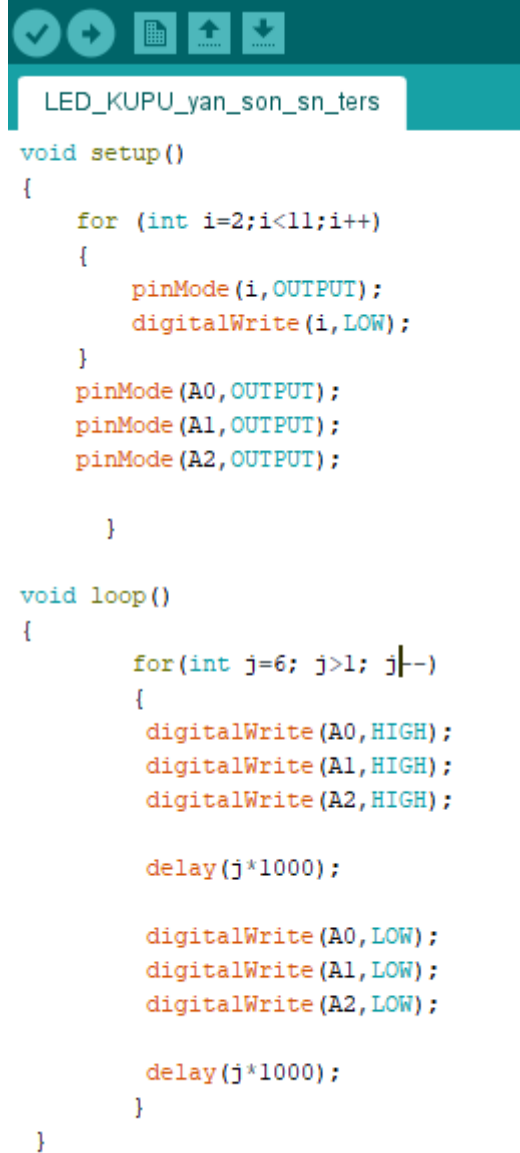
    delay(j*1000);
  }
}

```

Resim 3.37: Gözle-Uygula Etkinliği Örnek Kodu

1.16 Uygula- LED Küpünü Farklı Aralıklarla Yakıp Söndürüyorum (Öğrenci 1)

Bir önceki etkinlikten farklı olarak bu etkinlikte LED'lerin yanıp sönmeleri arasındaki süre altı saniye, beş saniye, dört saniye, üç saniye, iki saniye olacak şekilde değiştirilmelidir. Bu görev öğrenciler tarafından yapılacaktır. Öğrencilerden döngü içerisindeki ilk değer, son değer ve artış miktarını değiştirerek “j” değişkenini geriye doğru sayabileceğini keşfetmeleri beklenmektedir. Öğrencilere bu keşif için gerekli süre verilmelidir. Bu süre sonunda öğrenciler başarılı olamazsa eğitmen programın nasıl yapılacağını açıklayabilir. Aşağıdaki resimde örnek kod görülmektedir.



```

void setup()
{
  for (int i=2;i<11;i++)
  {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
}

void loop()
{
  for(int j=6; j>1; j|--)
  {
    digitalWrite(A0, HIGH);
    digitalWrite(A1, HIGH);
    digitalWrite(A2, HIGH);

    delay(j*1000);

    digitalWrite(A0, LOW);
    digitalWrite(A1, LOW);
    digitalWrite(A2, LOW);

    delay(j*1000);
  }
}

```

Resim 3.38: Uygula Etkinliđi Örnek Kodu

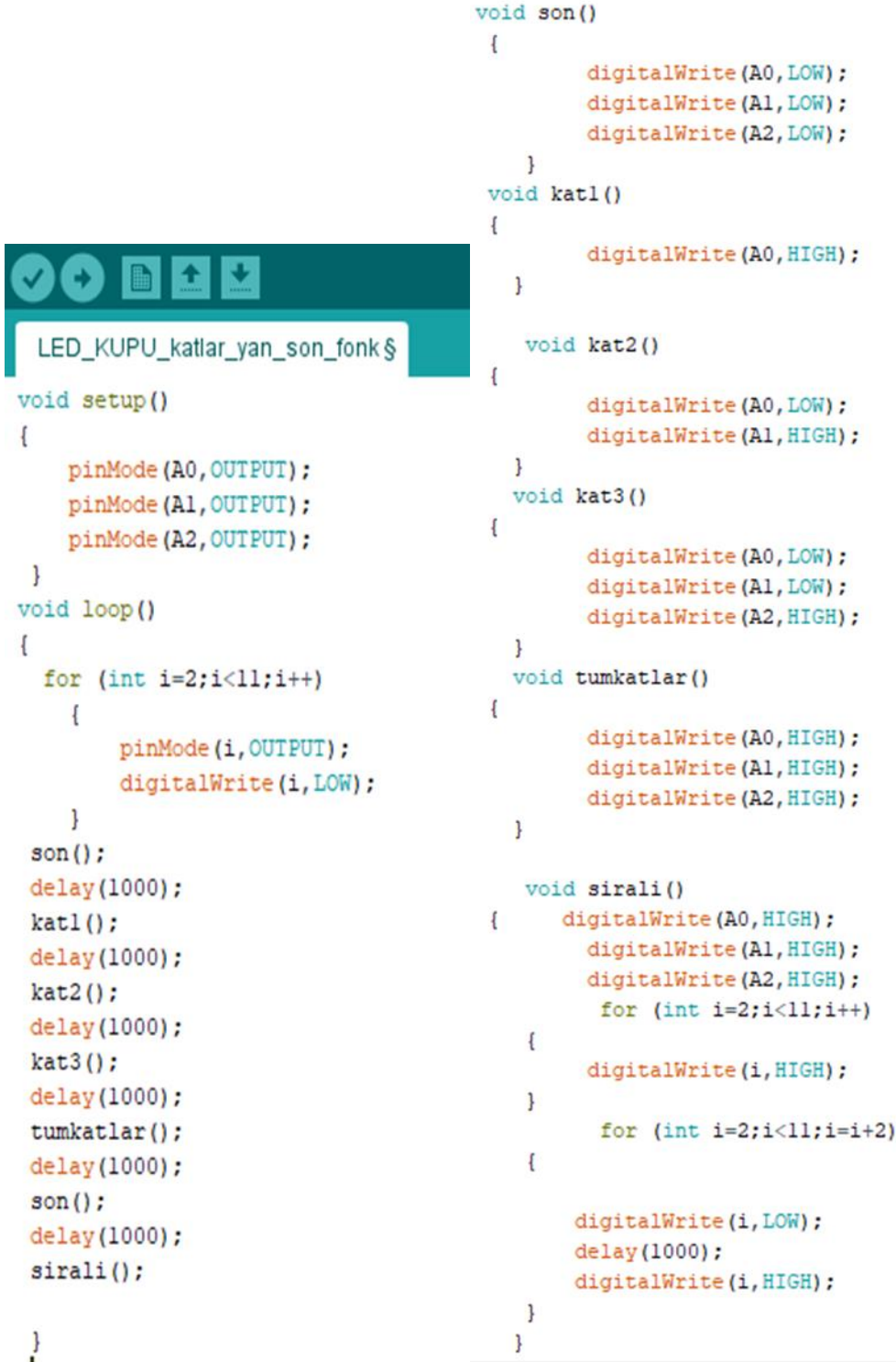
1.17 Uygula- Katlar Yanıp Sönüyor (Öğrenci 2)

Bu etkinlikte öğrencilerden LED küpünün yanıp sönmesini aşağıda belirtildiđi 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 LED'ler 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 aşağıda verilmiştir. Öğrenciler kendi kodlarını yazmalıdırlar. Eğitimci bu etkinlikte öğrencileri kodları yazarken fonksiyon kullanmaları konusunda uyarmalıdır. Örnek kodun aşağıda görüldüğü gibi tasarlanmasının sebebi pedagojiktir. Bu kod

programlama açısından bakıldığında farklı bir şekilde de tasarlanabilir. Fakat aşağıdaki resimde görülen fonksiyonlar dersin amacı doğrultusunda yeterlidir.



```

void son()
{
    digitalWrite(A0, LOW);
    digitalWrite(A1, LOW);
    digitalWrite(A2, LOW);
}
void kat1()
{
    digitalWrite(A0, HIGH);
}
void kat2()
{
    digitalWrite(A0, LOW);
    digitalWrite(A1, HIGH);
}
void kat3()
{
    digitalWrite(A0, LOW);
    digitalWrite(A1, LOW);
    digitalWrite(A2, HIGH);
}
void tumkatlar()
{
    digitalWrite(A0, HIGH);
    digitalWrite(A1, HIGH);
    digitalWrite(A2, HIGH);
}
void sirali()
{
    digitalWrite(A0, HIGH);
    digitalWrite(A1, HIGH);
    digitalWrite(A2, HIGH);
    for (int i=2; i<11; i++)
    {
        digitalWrite(i, HIGH);
    }
    for (int i=2; i<11; i=i+2)
    {
        digitalWrite(i, LOW);
        delay(1000);
        digitalWrite(i, HIGH);
    }
}

void setup()
{
    pinMode(A0, OUTPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, OUTPUT);
}
void loop()
{
    for (int i=2; i<11; i++)
    {
        pinMode(i, OUTPUT);
        digitalWrite(i, LOW);
    }
    son();
    delay(1000);
    kat1();
    delay(1000);
    kat2();
    delay(1000);
    kat3();
    delay(1000);
    tumkatlar();
    delay(1000);
    son();
    delay(1000);
    sirali();
}

```

Resim 3.39: Gözle-Uygula Etkinliği Örnek Kodu

2. ADIM: TASARLA ve ÜRET

2.1 Tasarla - 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. Öğrenciler animasyonu yaparken döngüler ve değişkenleri kullanmak zorundadır. Gruplar seçtikleri şarkılarla uyumlu yanan LED küpünü şarkı ile senkronize bir şekilde sınıfta sunacaklardır.

Program kodlarının yazımında ve algoritmanın oluşturulmasında eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Bu uygulama için nasıl bir algoritma tasarlanabilir?
- Şarkı seçiminde nelere dikkat etmek gerekir?
- Şarkıya uyumlu animasyonu nasıl tasarlayabiliriz?
- Ne tür fonksiyonlar oluşturulabilir?
- Ne amaçla tasarlanacak fonksiyonlara ihtiyaç duyulabilir?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Şarkı belirlendikten sonra şarkı kısımlara ayrılmalı.
- LED küpü animasyonları belirlenmeli.
- LED küpü animasyonları için uygun fonksiyonlar yazılır.
- Şarkı ile uyumlu animasyonlar ve saniyeleri belirlenir.

2.2. Üret - Şarkıyla Uyumlu LED Küpü

Öğrenciler yukarıda çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirebilecek devreyi yukarıda hazırladıkları şekilde kurar ve kodlar. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerekli olduğu noktada eğitmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

2.3 Tasarla - 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.

Program kodlarının yazımında ve algoritmanın oluşturulmasında eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Buzzer ile çalınabilecek nitelikte uygun müzik seçimi yapılmalı.
- Buzzerda uygun notların seçimi için notalar belirlenmeli.
- Buzzerdan gelen sesin pick yaptığı anlarda ya da sesin azaldığı zamanlarda LED küpünde nasıl bir animasyon oluşturulacağı planlanmalı.

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Şarkı belirlendikten sonra şarkı buzzerda çalınabilecek şekilde kısımlara ayrılmalı.
- Buzzer için notalar ve bunların kodda nasıl yazılacağı planlanmalı.
- Buzzer ve LED küpü için uygun fonksiyonlar yazılmalı

2.4 Üret- Buzzerla Çalınan Şarkıyla Uyumlu LED Küpü

Öğrenciler yukarıda çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirebilecek devreyi yukarıda hazırladıkları şekilde kurar ve kodlar. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Gerekli noktada eğitmen onlara yardımcı olabilir. Fakat hiçbir şekilde tam bir çözüm verilmemelidir. Gruplar çözümlerini kendileri üretmelidir.

3. ADIM: DEĞERLENDİR

A) Eğitmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmalarını sağlar.

1. Bu hafta LED küpü etkinliğinde küpün her bir katı oluşturulduğunda oluşturulan kattaki LED'lerin çalışıp çalışmadığı kontrol edilmiştir ve katlar birbirine bağlandıktan sonra bütün LED'ler kontrol edilmiştir. Bunun yerine LED küpünün tamamı bitirildikten sonra LED'ler 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 hafta ve geçen hafta 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:

- 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, eğitmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
- 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 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 uzun 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 sunumunu yapan diğer gruplara 10 puan üzerinden not verecektir. Öğretmen 20 puan üzerinden not vermelidir. Sunumlar sonrasında puanlar toplanıp öğretmen tarafından birinci grup sınıfta ilan edilecektir.

4.Bölüm - Transistör, Kondansatör, Ultrasonik Mesafe ve Ses Sensörü

Ön bilgi:

- Öğrenciler algoritma mantığını bilir.
- Öğrenciler Arduino IDE’de temel programlama kavramlarını bilir.
- Öğrenciler daha önceki haftalarda kullanılan devre elemanlarını bilir ve devre tasarımında kullanabilir.
- Öğrenciler devre tasarlayıp buna uygun Arduino IDE kodunu yazabilir.

Bölüm Kazanımları:

- Öğrenciler transistörün çalışma prensiplerini bilir ve devrelerde kullanabilir.
- Öğrenciler kondansatörün çalışma prensiplerini bilir ve devrelerde kullanabilir.
- Öğrenciler ultrasonik mesafe sensörünün çalışma prensiplerini bilir ve bu devre elamanını Arduino UNO’ya bağlayarak devre tasarlayabilir.
- Öğrenciler ultrasonik mesafe sensörünü Arduino IDE’de yazdığı kodla kontrol edebilir.
- Öğrenciler ses sensörünün çalışma prensiplerini bilir ve bu devre elamanını Arduino UNO’ya bağlayarak devre tasarlayabilir.
- Öğrenciler ses sensörünü Arduino IDE’de yazdığı kodla kontrol edebilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin Arduino UNO mikrodenetleyicisini kullanarak transistör, kondansatör, ultrasonik mesafe sensörü ve ses sensörünün çalışma prensiplerini anlamalarıdır. Bu devre elamanlarını kullanarak devre tasarlayıp ya da tasarlanmış devreleri anlayabilmeleri istenmektedir. Ayrıca, Arduino IDE yazılımı ile bu devre elemanları ile kurulan devrelerin istenen şekilde programlanabilmesini sağlamaktır. Böylece sensörlerin kullanıldığı devrelerin tasarlanması ve Arduino IDE ile programlanması sayesinde öğrencilerin yazılımsal temelini geliştirmek hedeflenmektedir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi
Arduino UNO
LDR
BC547
HC-SR04 ultrasonik mesafe sensörü
100 Ohm direnç
220 Ohm direnç
330 Ohm direnç
1K direnç
2K direnç
Kırmızı LED
Breadboard
Bağlantı kabloları

Haftanın İşlenişi:

Gözle: Transistör, kondansatör, ultrasonik mesafe sensörü ve ses sensörünün çalışma prensiplerini ve kontrolünü farklı uygulamalar ve açıklamalarla zenginleştirerek bunu Arduino UNO ve yazılım arayüzü olan Arduino IDE ile bütünleşik olarak farklı amaçlar için kullanılması hedeflenmektedir.

Uygula: Transistör, kondansatör, ultrasonik mesafe sensörü ve ses sensörünü farklı amaçlarla tasarlanacak olan örnek devrelerin oluşturulması ve programlanması. Bu aşamada kodlama sürecinde değişken, döngü, dizi ve fonksiyon kullanımı pekiştirilmelidir.

Tasarla: Devre tasarımı öncesinde Arduino UNO kullanarak ilgili projeyi gerçekleştirebilmek için gerekli devrenin şematik olarak hazırlanması ve devreyi uygun şekilde çalıştıracak algoritmanın tasarlanması.

Üret: Arduino UNO mikrodenetleyicisinin dijital ya da analog çıkış pinlerine transistör, kondansatör, ultrasonik mesafe sensörü ve ses sensörünün doğru bağlanarak devre tasarımlarının yapılması. Farklı uygulamalarının Arduino IDE kullanılarak kodlanması.

Değerlendir: Yansıtma Etkinliği.

1. ADIM: GÖZLE ve UYGULA

1.1 Gözle ve Uygula – LDR ile LED Uygulaması (Öğrenci 1)

Malzeme Listesi
Arduino UNO
LDR
330 Ohm direnç
Kırmızı LED
Breadboard
Bağlantı kabloları

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 değen ışığın şiddeti arttıkça 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 4.1: LDR

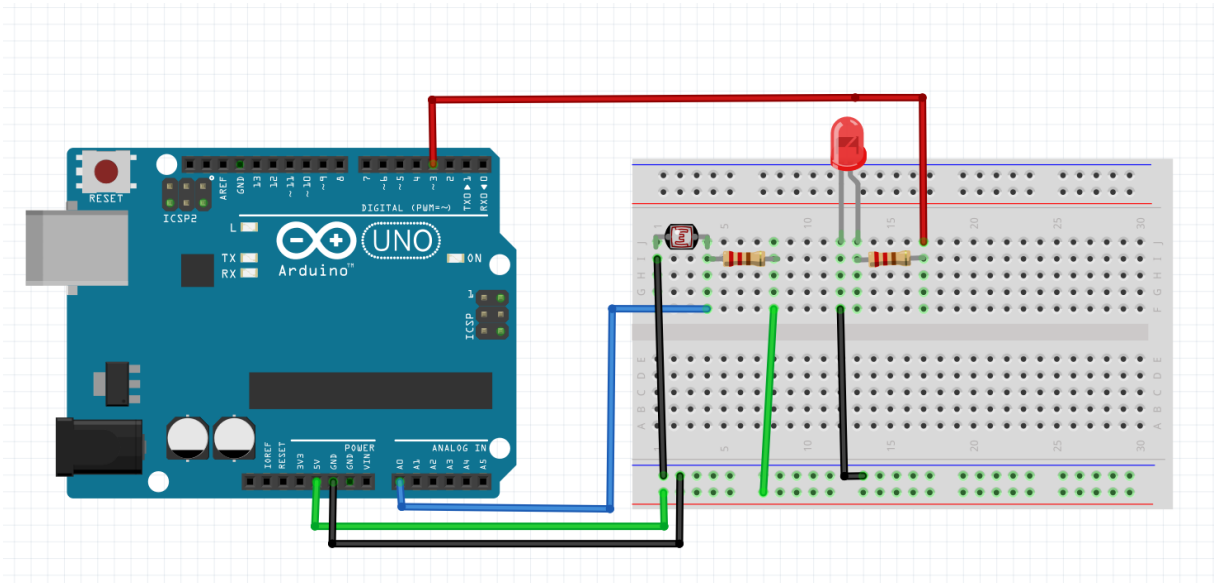
LDR'nin sağlamlık kontrolünü avometre ile gerçekleştirebiliriz. Avometreyi kullanarak LDR'nin direncini ışıklı ve karanlık ortamlarda ölçersek görürüz ki; ortam ışığı arttıkça direnç değeri azalacak, ışık azaldıkça ise bu değer daha yüksek bir direnç değeri olacaktır.

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 düşen ışığın el tarafından engellenmesi ve gölge oluşmasıdır). 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 birbirine bağlanır ve ışık miktarı arttıkça LDR'nin direnci azalır ve böylece çıktı gerilimi düşer. Işık miktarı ne kadar artarsa LDR'nin üzerindeki gerilim o kadar azalacaktır. Işık miktarı ne kadar azalırsa gerilim o kadar artacaktır. Aşağıda devre şeması ve Arduino IDE'de yazılmış programı verilen uygulamada LED 330 Ohm direnç üzerinden Arduino'nun 3 numaralı (PWM) pinine; LDR'nin bir bacağı da A0 analog pinine bağlanmıştır. Arduino IDE programı incelendiğinde LED ve LDR pin tanımlarından sonra "ldrdeger" isimli bir değişken tanımlandığı görülecektir. LDR ile A0 analog pininden gelen 0-1023 arasındaki değer "ldrdeger" değişkeni atamasıyla map() fonksiyonu ile 0-255 arasında ölçeklenmekte ve bu çıkışı kullanarak elde edilen değer LED'e aktarılmaktadır. Böylece ortamın parlaklığına göre (örneğin karanlık arttıkça LDR'nin direnci artacaktır) LED'in parlaklığı da değişmiş olacaktır. Bu uygulamada eğitmen devreyi kurar ve Arduino IDE'de programın çalışmasını öğrencilere gösterir. Bu aşamada LDR üzerindeki parlaklık değerinin değişimiyle LED'in parlaklığının değişiminin vurgusu yapılır. Daha sonra öğrencilerden devreyi kurmalarını ve Arduino IDE'de programı yazmalarını ister. Aşağıdaki resimlerde örnek devre şeması ve kod görülmektedir.

Dikkat

Yukarıdaki açıklamada PWM pinlerinden 3 nolu pin kullanılmıştır. Devreyi öğrenciler kurarken isterlerse 3,5,6,9,10 ve 11 nolu PWM pinlerini kullanabilirler.



Resim 4.2: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

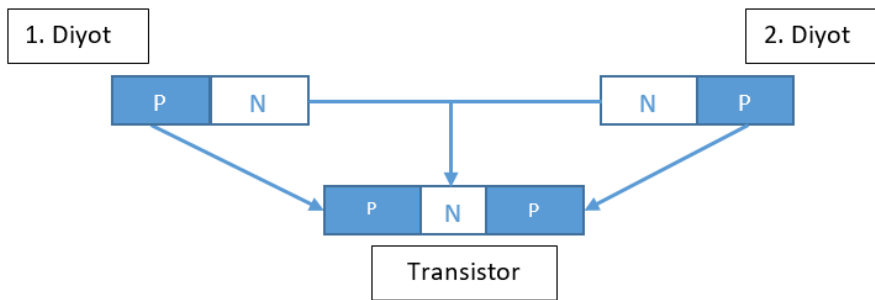
ldr_sensor §
1 int led=3;
2 int ldr=A0;
3 int ldrDeger;
4
5 void setup() {
6   pinMode(led, OUTPUT);
7   pinMode(ldr, INPUT);
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   ldrDeger=analogRead(ldr);
13   ldrDeger=map(ldrDeger, 0, 1023,0, 255);
14   Serial.println(ldrDeger);
15   analogWrite(led, ldrDeger);
16   delay(100);
17 }

```

Resim 4.3: Gözle-Uygula Etkinliği Örnek Kodu

1.2 Gözle- Uygula: 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 mikrodenetleyicisinin 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. Aşağıdaki resimde transistörün şeması görülmektedir.



Resim 4.4: PNP Transistör

Yukarıdaki resimde iki diyotun N kısımları birbirine birleştirilerek oluşturulan PNP bir transistör yer almaktadır. Gerçekte bir transistör iki diyotun birleştirilmesi ile elde edilmez. Bu resim 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 tipli transistörlerin temel çalışma prensipleri aynı olmasına rağmen ters yükler ile çalıştırılan iki ayrı transistördür.

Burada ağırlıklı olarak NPN transistörler üzerinden konu anlatılacaktır. Aşağıdaki şekilde bir güç kaynağının “+” ucu NPN transistörün N bacaklarından birine “-“ ucu ise diğerine bağlanmıştır. Bu şekilde 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 4.5: NPN Transistör Devre Bağlantısı

Aşağıdaki şekilde 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. Aşağıdaki resimde NPN transistör devre bağlantısı görülmektedir.



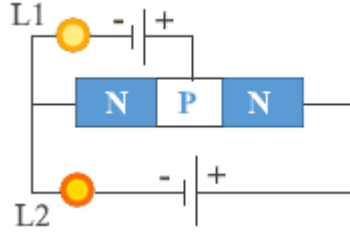
Resim 4.6: NPN Transistör Devre Bağlantısı

Dikkat

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

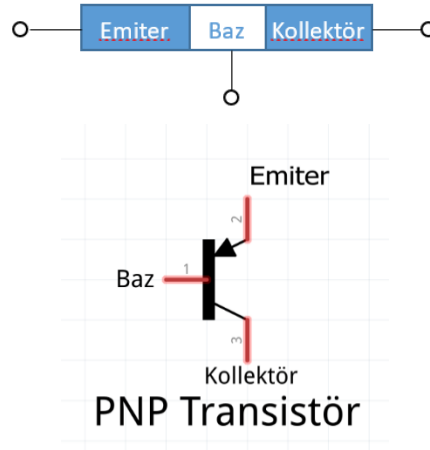
Yukarıdaki iki resimde gösterilen devrede de “P” kısmı güç kaynağının yönüne bağlı olarak ya soldan sağa olabilecek akım geçişini ya da sağdan sola olabilecek akım geçişini engellemektedir. Bu engeli ortadan kaldırmak için “P” kısmını iletken hale getirmek gerekir. Bunun için “NP” kısmına (soldaki diyot) bir gerilim uygulanması gerekmektedir (silisyum için 0.7 V). Bu durum aşağıdaki şekilde gösterilmiştir. Bu kısma gerilim uygulandığında bir diyot gibi davranıp akım ileticektir 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 NPN transistör devre bağlantısı görülmektedir.



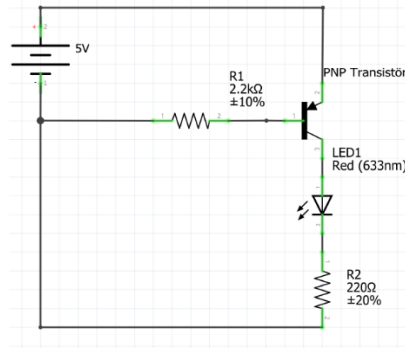
Resim 4.7: NPN Transistör Devre Bağlantısı

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.



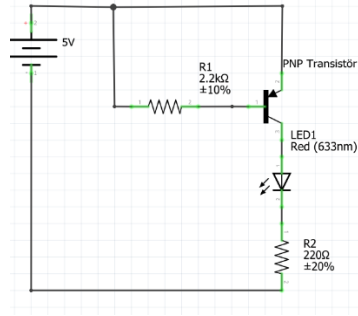
Resim 4.8: PNP Transistör

Aşağıdaki resimde PNP transistörün bazına “-” bir gerilim uygulanmıştır. Böylece emiterden baza doğru küçük bir akım akacak ama emiterden kollektöre yüksek bir akım geçecektir.



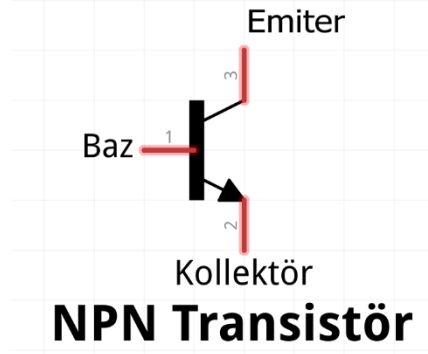
Resim 4.9: PNP Transistör Devre Bağlantısı

Yine aynı PNP transistörün bazına “+” gerilim uygulanırsa bu durumda emiterden kollektöre akım geçişi duracaktır. Aşağıdaki resimde PNP transistör devre bağlantısı görülmektedir.

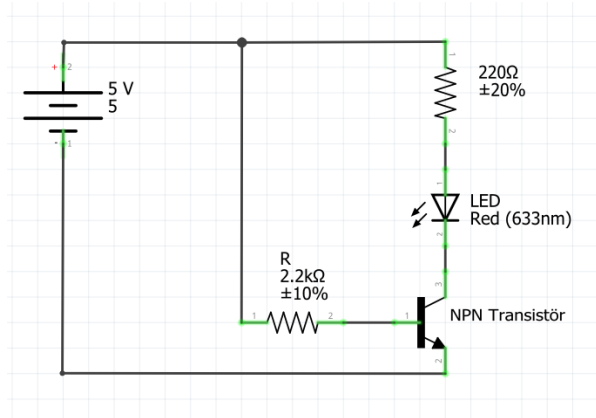


Resim 4.10: PNP Transistör Devre Bağlantısı

Bu ders kapsamında inceleyeceğimiz bir başka transistör türü ise NPN transistördür. Bu transistör de tetikleme amacıyla kullanılacak olursa transistörün bazına – bir gerilim uygulanırsa, kollektörden emitere bir akım akışı olmaz. Fakat Transistörün bazına “+” bir gerilim uygulanması durumunda ise bazdan emitere küçük bir akım geçişi gözlenirken, kollektörden emitere büyük bir akım geçişi yapılabilir. Aşağıdaki resimlerde NPN transistör vev devre bağlantısı görülmektedir.

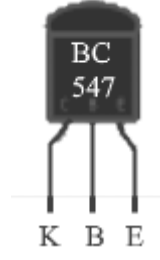


Resim 4.11: NPN Transistör



Resim 4.12: NPN Transistör Devre Bağlantısı

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



Resim 4.13: BC 547 NPN Transistör Ayak Yapısı

Dikkat

Transistörlerin bacak sıralamaları değişmektedir. Başka bir transistörde bacak sıralaması farklı olabilir. Transistörün katalogundan bacakları bulunabilir. Ayrıca, transistörün bazına bağlanacak dirençler bu devre elemanının aşırı akım çekerek bozulmasını önleyecektir.

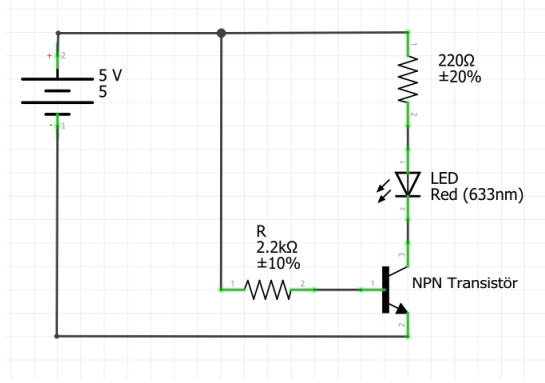
Dikkat

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.

1.3 Uygula– Transistörün Anahtarlama için Kullanımı

Malzeme Listesi
Arduino UNO
BC547
220 Ohm direnç
2 K Ohm direnç
Breadboard
Bağlantı kabloları

Eğitmen transistörün bir anahtarlama elemanı olarak da kullanılabileceğini yukarıdaki örnekler üzerinden bir kere daha anlatır. Ardından öğrencilerden BC 547 NPN transistörü kullanarak aşağıdaki resimde görülen devreyi kurmasını ister. Devreyi kurduktan sonra avometre ile transistörün baz-emiter akımı ve kollektör-emiter arasındaki akım ölçülmelidir.



Resim 4.14: Uygula Etkinliği Devre Şeması

Bu ölçüm daha sonra transistörün bazına – gerilim uygulanarak tekrar edilir ve ölçümler not edilerek öğrencilerin bu konu üzerinde tartışmalara sağlanır. Bu ölçümlerden elde edilen değerler NPN transistörün çalışma mantığını daha iyi anlatmak için kullanılır. Bu süreci desteklemek için ilgili BC547'nin bu dokümanla birlikte yollanacak katalogları eğitmen tarafından açıklanmalıdır.

Dikkat

Transistörlerin kataloglarında yer alan voltaj düşümünü de hesap ettirilebilir. Yani, 5 V verdiğimizde transistörden çıkan voltaj 4,5 V olacaktır. Bazı devreler kurulurken buna da dikkat etmeliyiz.

Arduino UNO üzerinden devre kurup pinleri kontrol ettiğimizde bu pinler maksimum 40mA akım gönderdiği için daha yüksek akıma ihtiyaç duyan devreleri kontrol etmek mümkün olmayacaktır. Fakat bu tür devrelerde transistörler kullanılabilir ve daha yüksek akıma ihtiyaç duyan devreler kontrol edilebilir. Transistörleri anahtarlama için ise yine Arduino UNO kullanılabilir ve bu devreyi harici bir güç ile besleyerek kontrol edebiliriz.

1.4 Gözle – Uygula: Transistör ile Değil Kapısı Yapıyorum (Öğrenci 2)

Malzeme Listesi

Arduino UNO

BC547

1 K Ohm Direnç

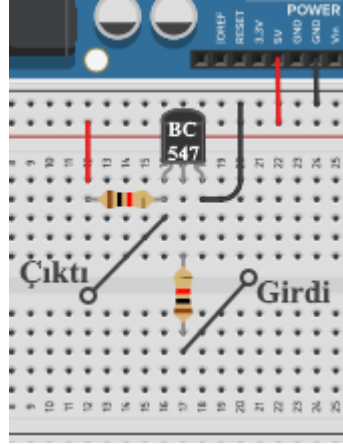
Breadboard

Bağlantı kabloları

Bu etkinlikte amaç transistör kullanarak bir değil (not) kapısı oluşturmaktır. “Değil kapısı”nın veya değil operatörünün görevi girdinin tersini almaktır. Girdi olarak 1 (5V) girilirse çıktı olarak 0 (0 V) verir. Girdi olarak 0 (0V) girilirse çıktı olarak 1 değeri (5 V) üretir.

Giriş	Çıkış
1	0
0	1

Aşağıda 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” veilirse BC547’nin baz ayağında gerilim oluşmaz ve güç kaynağından direnç ile gelen akım Çıktı’ya iletilir. Böylece Çıktı “1” olur.

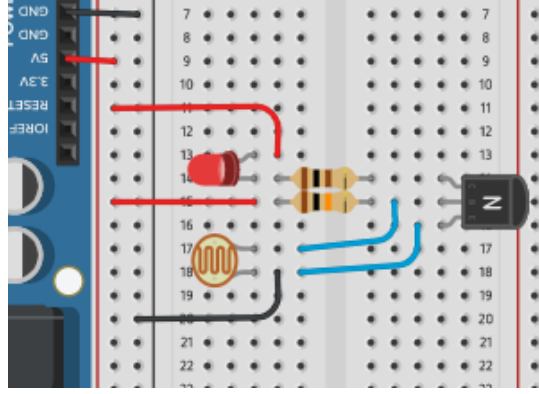


Resim 4.15: Gözle-Uygula Etkinliği Örnek Devre Şeması

1.5 Gözle-Uygula Transistör ile Karanlıkta Yanan LED (Öğrenci 1)

Malzeme Listesi
Arduino UNO
BC547
Kırmızı LED
100 Ohm Direnç
1 LDR
Breadboard
Bağlantı kabloları

Bu etkinlikteki amaç transistör kullanarak karanlık olduğunda yanan bir LED devresi tasarlamaktır. Bu devre aşağıdaki resimde gösterilmiştir.



Resim 4.16: Gözle-Uygula Etkinliği Örnek Devre Şeması

Devrede Arduino sadece 5 V güç kaynağı olarak kullanılmıştır. Arduino yerine herhangi bir 5 V güç kaynağı da kullanılabilir. Eğitimci 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 100K Ohm direnç üzerinden gelen akım transistörün baz bacağına tetikleyecek değere ulaşmadan LDR üzerinden GND'ye iletilir. Fakat karanlık bir ortamda LDR'nin direnci artacağından transistörün baz bacağına 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 LED'in yanmasına sebep olur.

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

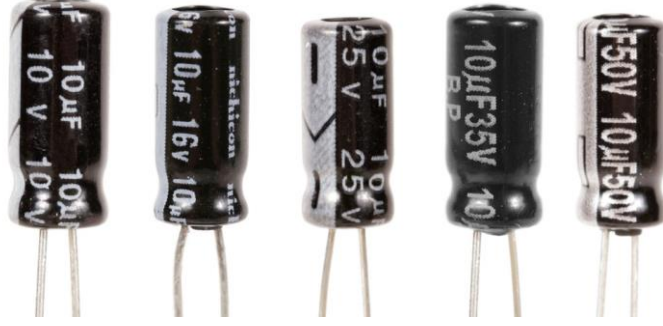
1.6 Gözle-Uygula: Kondansatör ve Akım İlişkisi (Öğrenci 2)

Kondansatörler çok basit anlamda çok küçük pillerdir. Kendilerine sağlanan elektriği içerisindeki metal plakalar arasında depolayıp, ihtiyaç duyulduğunda devreye geri sağlarlar. Çok düşük miktarda elektrik depolayabilirler ama doğru kullanılırsa sayısız defa şarj edilebilir. Kapasitör ya da sığaç gibi farklı isimlerle de adlandırılabilirler. Aşağıdaki resimde kondansatör örnekleri görülmektedir.



Resim 4.17: Kondansatör Örnekleri

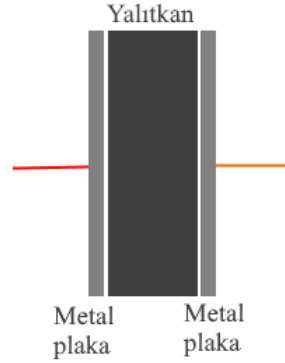
Kondansatörleri devre içerisinde kullanırken farad cinsinden kapasitelerine bakılır. Kondansatörün farad cinsinden değeri ne kadar yüksekse kondansatör o kadar çok enerji depolayabilir. Ayrıca kondansatörlerin üzerinde maksimum voltaj değeri de yer almaktadır. Bu da devredeki voltajdan birkaç volt yüksek olmalı ki kondansatörümüz devrede sorunsuz, bozulmadan çalışsın. Kondansatörler kaç volt ile yüklenirlerse o voltaj değerini üzerlerinde tutarlar. Aşağıdaki resimlerde farklı voltaj değerindeki kondansatörler görülmektedir.



Resim 4.18: Farklı Voltaj Değerindeki Kondansatörler

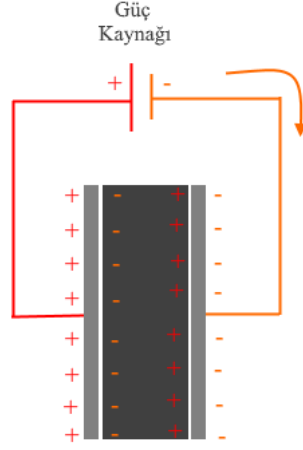
Farad	MicroFarad (µF)	NanoFarad (nF)	PicoFarad (pF)
1	10^6	10^9	10^{12}
10^{-6}	1	10^3	10^6
10^{-9}	10^{-3}	1	10^3
10^{-12}	10^{-6}	10^{-3}	1

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. Aşağıdaki resimde kondansatörün iç yapısı görülmektedir.



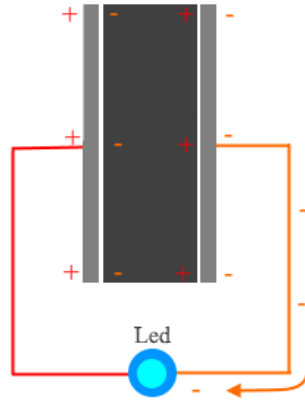
Resim 4.19: Kondansatör İç Yapısı

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 4.20: Kondansatör Güç Kaynağı Bağlantısı

Güç bağlantısı kesildiğinde ise eksi (-) yükle yüklenen uçtaki elektronlar devrede buldukları bir yol üzerinden artı (+) yüklü uca ulaşmaya çalışır (aşağıdaki şekilde devreye bağlanmış led üzerinden). Bu işi yaparken de devre üzerinde bir elektrik akımı oluşmasına neden olur. Kondansatör deşarj olduğunda bu akım kesilir. Aşağıdaki resimde kondansatör LED bağlantısı görülmektedir.

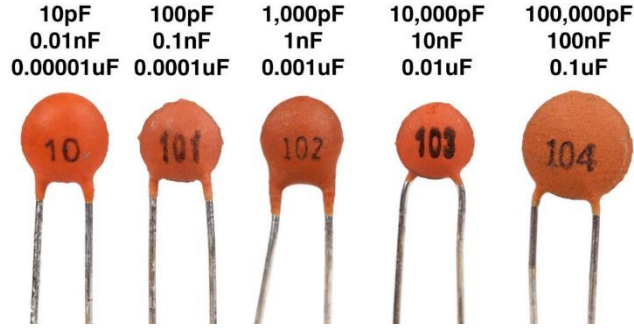


Resim 4.21: Kondansatör LED Bağlantısı

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 Uygula: Kondansatör Değeri Okuma ve Ölçümü (Öğrenci 1)

Eğitmen setlerin içinden çıkan farklı kapasitelerdeki kondansatörleri alarak değerlerini okumalarını ister. Burada set içerisinde yer alan mercimek tipi kondansatörler de kullanılacaktır. Aşağıdaki resimlerde mercimek kondansatörler görülmektedir.

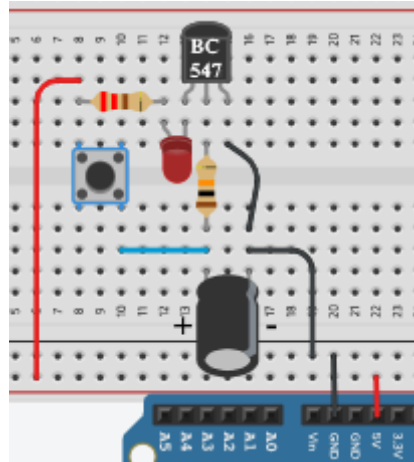


Resim 4.22: Mercimek Kondansatör

Öncelikle avometre voltmetre konumuna alınır. Daha sonra setten alınan bir kondansatörün avometre ile gerilimi ölçülür. Öğrencilerle birlikte bu işlem gerçekleştirilir ve ölçüm sonucunda kondansatörün gerilimi 0 V çıkacaktır. Ardından Arduino UNO'nun 3.3 V çıkışı ve 5 V çıkışı kullanılarak bir kondansatöre kısa süre, öncelikle 3.3 V verilir. Burada kondansatörün “+” ucuna Arduino Uno'nun 3.3 V çıkışı, kondansatörün “-” ucuna da Arduino UNO'nun GND'si bağlanır. Burada kondansatörün bacaklarına bağladığımız krokodiller yardımı ile Arduino UNO'dan besleme yapılabilir. Zaten kondansatör çok kısa sürede dolacaktır. Yine aynı krokodiller yardımı ile kondansatördeki gerilimin avometrenin voltaj ölçüm modunda ölçülmesi istenir. Bu ölçümü hızlı yapabilmek için avometre kondansatöre bağlıyken, kondansatör Arduino UNO'dan şarj edilir. Kondansatörün şarj olması ve voltaj düşümünü buradan gözlemlenebilir, çünkü kondansatör kısa sürede dolacak ve hızla boşalacaktır. Bu gözlemi yapabilmenin bir başka yolu ise yine aynı düzenele kondansatöre bağlı krokodiller yardımı ile kondansatörü şarj etmek ardından ayaklarına bir kırmızı LED bağlayarak ne kadar yanacağını gözlemlemektir. Kondansatör LED'i kısa süre yakacaktır.

1.8 Gözle: Transistör ve Kondansatörü Kullanıyorum (Öğrenci 2)

Bu etkinlikteki amaç transistör ve kondansatörü kullanarak butona basıldığında LED'i yakıp buton bırakıldığında yavaş yavaş LED'in 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 kullanılacak devre aşağıdaki resimde gösterilmiştir.



Resim 4.23: Gözle Etkinliği Örnek Devre Şeması

Eğitmen 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.

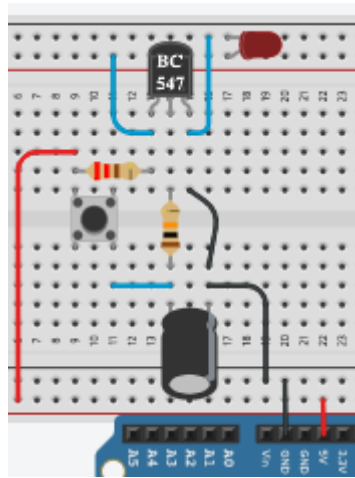
1.9 Uygula: LED'in Yanma Süresini Değiştiriyorum (Öğrenci 1)

Yukarıda yapılan devrede butona basıldığında LED bir müddet daha çalışıp sönmektedir. Bu etkinlikteki amaç LED'in yanma süresini artırmak ve azaltmaktır. Eğitmen öğrencilerden devrede değişiklik yaparak LED'in yanma süresini değiştirmesini ister. Fakat bunu yaparken öğrenciler Arduino'yu bir önceki devrede olduğu gibi güç kaynağı olarak kullanabilirler. Arduino'nun mikrodenetleyici özelliklerini kullanmamalıdır.

Bu görevi yerine getirmek için devredeki kondansatörü değiştirmek yeterli olacaktır. Kondansatörün kapasitesi artırılırsa LED'in yanma süresi artar. Kondansatörün kapasitesi azaltılırsa yanma süresi azalacaktır.

1.10 Uygula: Butona Basıldığında Sönüp Bir Süre Sonra Yanan LED (Öğrenci 2)

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



Resim 4.24: Uygula Etkinliği Örnek Devre Şeması

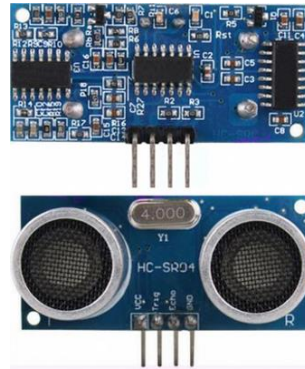
1.11 Gözle-Uygula: HC-SR04 Ultrasonik Mesafe Sensörü ile Mesafe Ölçüyorum (Öğrenci 1)

Malzeme Listesi
Arduino UNO
HC-SR04 ultrasonik mesafe sensörü
Breadboard
Bağlantı kabloları

Ultrasonik mesafe sensörü aslında denizaltılarda kullanılan sonar benzeri bir yapı ile mesafeyi ölçmektedir. Doğada ise yarasalar ve yunuslar ultrasonik ses dalgaları ile iletişim kurar ve hareket edebilirler. Elimizdeki setlerde HC-SR04 ultrasonik mesafe sensörü vardır ve ileride farklı projelerde bu devre elemanı kullanılacaktır.HC-SR04 üzerinde 4 adet pin bulunur. Bunlar:

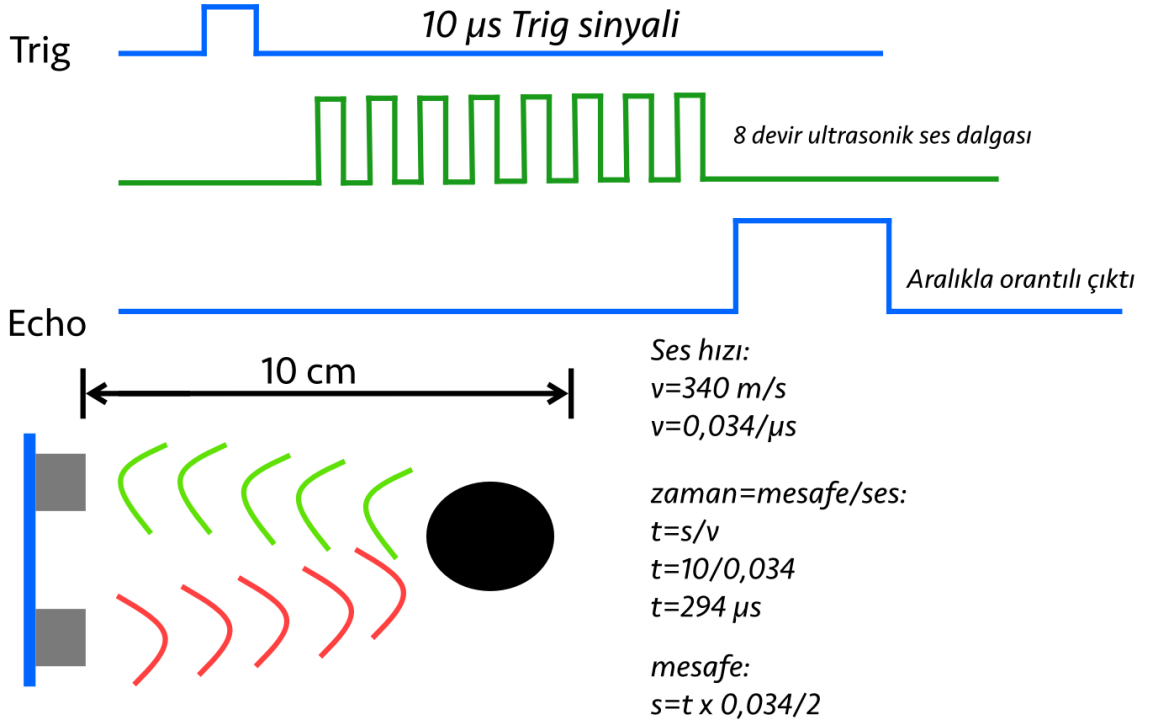
1. VCC
2. Trig(T)
3. Echo(R)
4. GND

Pinleridir. VCC ve GND pinleri Arduino UNO'nun 5 V ve GND çıkışlarına bağlanır. Trig ve Echo pinleri ise Arduino UNO üzerindeki dijital giriş-çıkış pinlerine bağlanır. Aşağıdaki resimlerde ultrasonik mesafe sensörü görülmektedir.



Resim 4.25: HC-SR04 Ultrasonik Mesafe Sensörü

Ölçüm yapılabilmesi için Trig pinine en az $10 \mu s$ 5V luk bir sinyal almalıdır. Bu sinyalin ardından HC-SR-04 sensörü 40000 Hz ile ortama 8 devir ultrasonik ses dalgaları yayar ve bu dalgalar ortamda bir engelle karşılaşırsa bu objeye çarparak yine sensörün alıcısı tarafından algılanır. Yansıyan bu ses dalgası algılandığında Echo pini 0 V'dan 5 V'a geçer. Echo pininin 5 V olduğu süre bize HC-SR04 ten çıkan ortamdaki objeye çarparak yansıyan dalgaların gidiş ve geri dönüş süresini verir. Bizde devrelerimizde buradan yola çıkarak HC-SR04 ile objenin uzaklığını hesap edebiliriz. Aşağıdaki resimlerde ultrasonik mesafe sensörünün çalışma prensibi görülmektedir.

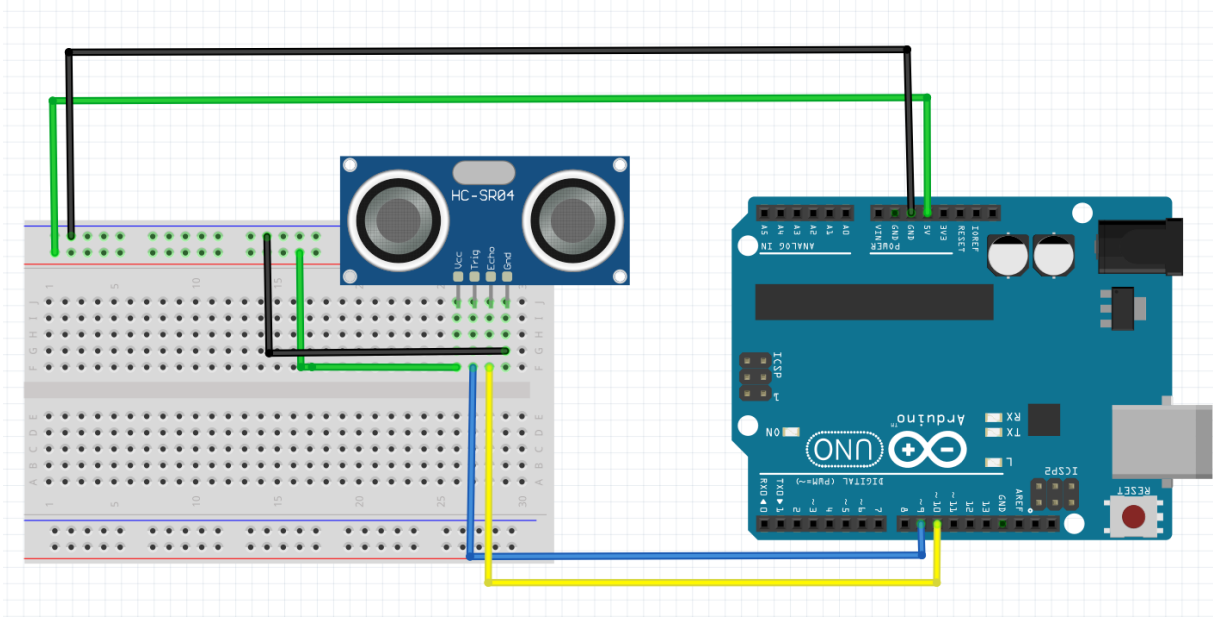


Resim 4.26: HC-SR04 Ultrasonik Mesafe Sensörünün Çalışma Prensipleri

Örneğin, 10 cm uzaktaki bir obje için bir hesaplama yapmak gerekirse ses hızı $0,034 \text{ cm}/\mu s$ 'dir. 10 cm uzaktadaki bir objeye ses dalgası $294 \mu s$ 'de çarpacaktır ve $294 \mu s$ 'de de dönecektir. Bizde bu nedenle hesaplamalarımızda bu formülü kullanırız.

- Mesafe (cm cinsinden) = $t \times 0,034/2$ ya da $t/58$

Bu etkinlikte amaç HC-SR04 ultrasonik mesafe sensörünün bağlantılarını ve çalışma mantığını anlayıp ses dalgaları üzerinden mesafe hesaplama sürecini açıklamaktır. Bu aşamada eğitmen aşağıdaki devreyi öğrencilerle birlikte kurar. Bu iş için kullanılacak devre aşağıdaki resimde gösterilmiştir. Devreyi kurma sürecinde eğitmen öğrencilere yardımcı olur. Ardından Arduino IDE'de aşağıdaki resimde görülen kod öğrencilerle birlikte yazılır ve açıklanır.



Resim 4.27: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

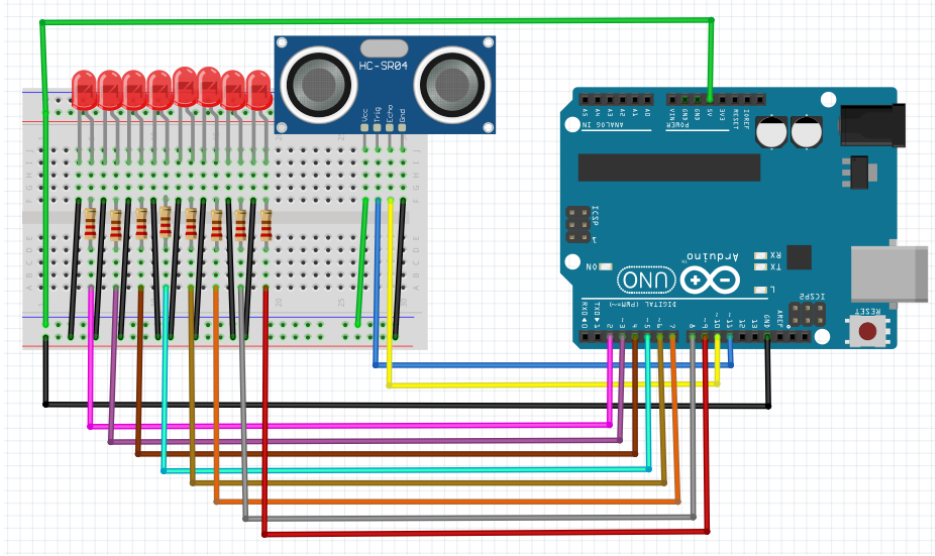
Yükle
ultrasonic_serial_2.ino
1 #define trigPin 9
2 #define echoPin 10
3 void setup() {
4   Serial.begin(9600);
5   pinMode(trigPin, OUTPUT);
6   pinMode(echoPin, INPUT);
7 }
8 void loop() {
9   int sure, mesafe;
10  digitalWrite(trigPin, LOW);
11  delayMicroseconds(2);
12  digitalWrite(trigPin, HIGH);
13  delayMicroseconds(10);
14  digitalWrite(trigPin, LOW);
15  sure = pulseIn(echoPin, HIGH);
16  mesafe = sure/58; // yada sure*0,034/2 :)
17  if (mesafe >= 200 || mesafe <= 0){
18    Serial.println("Güvenli mesafe");
19  }
20  else {
21    Serial.print("Uzaklık: ");
22    Serial.print(mesafe);
23    Serial.println(" cm");
24  }
25  delay(500);
26 }
27

```

Resim 4.28: Gözle-Uygula Etkinliği Örnek Kodu

1.12 Gözle-Uygula: 8 LED'li Park Sensörü (Öğrenci 2)

Bu aşamada öğrencilere 8 LED ile mesafeyi gösterecek bir devre kurlmaları istenir. Ardından Arduino UNO için Arduino IDE'de kodu yazarken dikkat edilmesi gereken noktalar eğitmen tarafından belirtilir. Bu devrede LED'lerin pin tanımlamalarının diziler üzerinden yapılacağı vurgulanır. 5 cm yakında bir obje varsa tek led yanıp sönecek ve her 5 cm de bir LED eklenerek toplamda 40 cm ye kadar bu şekilde ölçüm yapılarak LED'ler ile objenin uzaklığı gösterilecektir. Eğitmen kodu yazar ve öğrencilere açıklamalarda bulunur. Sonra devreyi kuran öğrenciler kodu yazarak düzeneğin doğru çalışıp çalışmadığını kontrol eder. Eğitmen bu süreçte öğrencilere destek olur. Etkinlik için örnek devre şeması ve örnek kod aşağıdaki resimde görülmektedir.



Resim 4.29: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

LED_ultrasonic
1  const int trigPin = 11;
2  const int echoPin = 10;
3  const int LEDdizisi[] = {2,3,4,5,6,7,8,9};
4  long duration;
5  float distance;
6
7  void setup()
8  {
9    pinMode(trigPin, OUTPUT);
10   pinMode(echoPin, INPUT);
11
12   for(int i=0; i<8 ;i++)
13   {
14     pinMode(LEDdizisi[i], OUTPUT);
15   }
16   Serial.begin(9600);
17 }
18
19 void loop()
20 { digitalWrite(trigPin, LOW);
21   delayMicroseconds(2);
22   digitalWrite(trigPin, HIGH);
23   delayMicroseconds(10);
24   digitalWrite(trigPin, LOW);
25   duration = pulseIn(echoPin, HIGH);
26   distance = 0.034*(duration/2);
27   if (distance <= 5)
28   { digitalWrite(LEDdizisi[0],HIGH);
29     delay(50);
30     digitalWrite(LEDdizisi[0],LOW);
31   }
32   else if (distance <= 10 && distance > 5)
33   {
34     for(int i=0; i<1; i++)
35     { digitalWrite(LEDdizisi[i],HIGH);
36       delay(50);
37       digitalWrite(LEDdizisi[i],LOW);
38     }
39   }

```

```

39     else if (distance <= 15 && distance > 10)
40     {
41     for(int i=0; i<2; i++)
42     { digitalWrite(LEDdizisi[i],HIGH);
43     delay(50);
44     digitalWrite(LEDdizisi[i],LOW);
45     }
46     }
47 else if (distance <= 20 && distance > 15)
48     {
49     for(int i=0; i<3; i++)
50     { digitalWrite(LEDdizisi[i],HIGH);
51     delay(50);
52     digitalWrite(LEDdizisi[i],LOW);
53     }
54     }
55 else if (distance <= 25 && distance > 20)
56     {
57     for(int i=0; i<4; i++)
58     { digitalWrite(LEDdizisi[i],HIGH);
59     delay(50);
60     digitalWrite(LEDdizisi[i],LOW);
61     }
62     }
63     else if (distance <= 30 && distance > 25)
64     {
65     for(int i=0; i<5; i++)
66     { digitalWrite(LEDdizisi[i],HIGH);
67     delay(50);
68     digitalWrite(LEDdizisi[i],LOW);
69     }
70     }
71 else if (distance <= 35 && distance > 30)
72     {
73     for(int i=0; i<6; i++)
74     { digitalWrite(LEDdizisi[i],HIGH);
75     delay(50);
76     digitalWrite(LEDdizisi[i],LOW);
77     }
78     }
79 else
80     {
81     for(int i=0; i<7; i++)
82     { digitalWrite(LEDdizisi[i],HIGH);
83     delay(50);
84     digitalWrite(LEDdizisi[i],LOW);
85     }
86     Serial.print("Mesafe:");
87     Serial.println(distance);
88     }
89     }

```

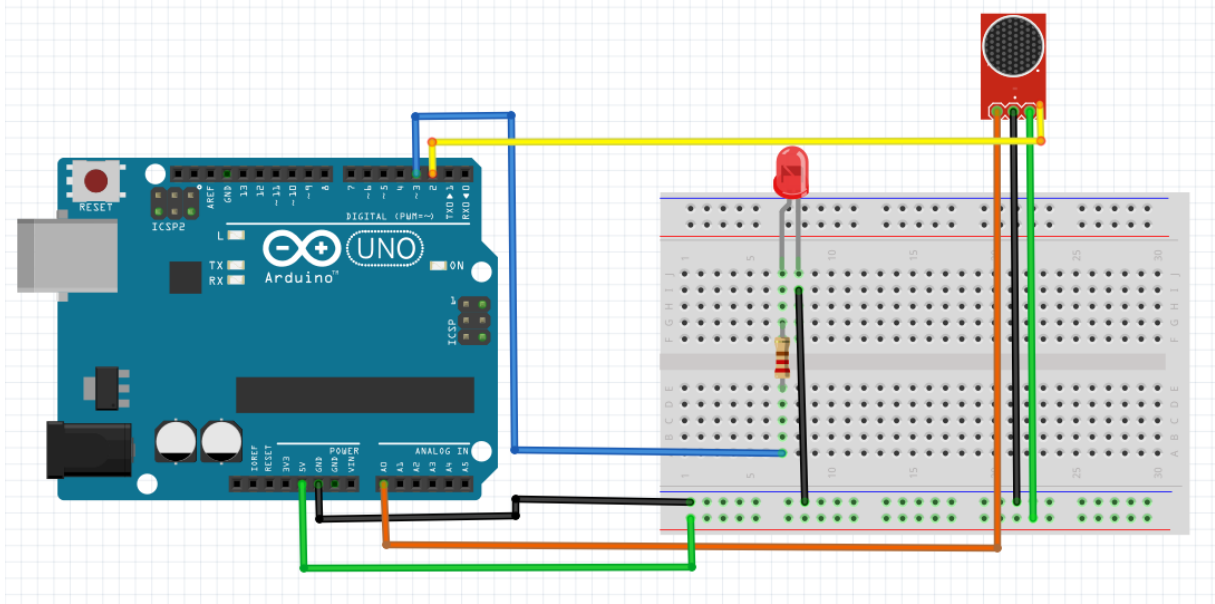
Resim 4.30: Gözle-Uygula Etkinliği Örnek Kodu

1.13 Gözle-Uygula: KY-038 Ses Sensörü ile Gürültü Dedektörü (Öğrenci 1)

Ses sensörü modülü olarak setlerde KY-038 bulunmaktadır. Bu modül aslında 3 fonksiyonel kısımdan oluşmaktadır. Bunlardan ilki modülün ucunda yer alan ve ses düzeyini ölçen sensördür. Bu sensör analog sinyali yükselticiye gönderir. Ardından, ses sensörü üzerinde yer alan potansiyometre ayarına göre modülün bu kısmına ulaşan ses sinyali güçlendirilir. Yine modülün bu kısmında, analog pine de ses seviyesi verisi gönderilir, eğer ölçülen ses seviyesi potansiyometre ile ayarlanan eşik değerinden büyük ise sensörün dijital pinine veri gönderilir.

- A0 Analog pin
- D0 Dijital pin
- GND GND
- VCC 5 V

Bu etkinlikte eğitmen öğrencilere ses sensörünün çalışma mantığını ve kullanılan pinleri anlattıktan sonra belirli bir ses seviyesinin üzerine çıktığında LED'i yakarak uyarı veren bir devrenin tasarımı üzerinde öğrencilere bilgi verir. Bunun için aşağıdaki resimde görülen devre şemasındaki devre eğitmen ve öğrenciler tarafından kurulur. Ardından aşağıdaki resimde görülen kod eğitmen tarafından yazılır ve öğrencilere açıklanır. Sonra öğrencilerin de bu kodu yazmaları istenir.



Resim 4.31: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

gurultu_olcer
int soundSensor = 2;
int Analog_Pin = A0;
int LED = 3;
void setup()
{
  pinMode (soundSensor, INPUT);
  pinMode(Analog_Pin, INPUT);
  pinMode (LED, OUTPUT);
  Serial.begin (9600);
}

void loop()
{
  int statusSensor = digitalRead (soundSensor);
  float Analog;
  Analog = analogRead (Analog_Pin);
  Serial.print("Gürültü düzeyi :");
  Serial.print(Analog, 4);
  Serial.print("Gürültü düzeyi :");

  if (statusSensor == 1)
  {
    digitalWrite(LED, HIGH);
    Serial.println(" AŞILDI ");
  }
  else
  {
    digitalWrite(LED, LOW);
    Serial.println(" AŞILMADI ");
  }
}

```

Resim 4.32: Gözle-Uygula Etkinliği Örnek Kodu

2. ADIM: TASARLA VE ÜRET

2.1 Tasarla-Üret: Kapalı Otopark Park Destek Sistemi

Bu etkinlikte evlerdeki kapalı otoparklarda kullanılmak üzere park yerinin dolu olup olmadığını kontrol eden, eğer boşsa yeşil LED yakan, doluyrsa kırmızı LED yakan bir uygulama gerçekleştirilecektir. Ayrıca, park etme sürecinde de park manevrasını kolaylaştırmak için sürücüyü yeşil, sarı, kırmızı LED'lerle ve park yerindeki ses sistemine üzerinden sesli uyarı verebilecek bir park sensörü uygulaması yapmanız gerekmektedir. Bunun için devrede 2'şer tane yeşil, sarı ve kırmızı LED bulunmalıdır. Park alanı boşken yeşil LED'ler yanıp sönmeli, doluyken kırmızı LED'ler ışık vermelidir. Park manevrası sırasında da geliştirilen park sistemi hem sesli, hem de ışıklı uyarı vermelidir. Işıklı uyarıda araç park sırasında yaklaşırken önce yeşil, sonra sarı çok yaklaşıncaya da kırmızı ışıkla yanıp sönerak uyarmalıdır. Bu etkinlikte devrede sadece buzzer kullanacak olsa da, devrede buzzer transistör ile tetikleneceği için devre gerekirse hoparlör bağlanabilecek şekilde de tasarlanabilir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

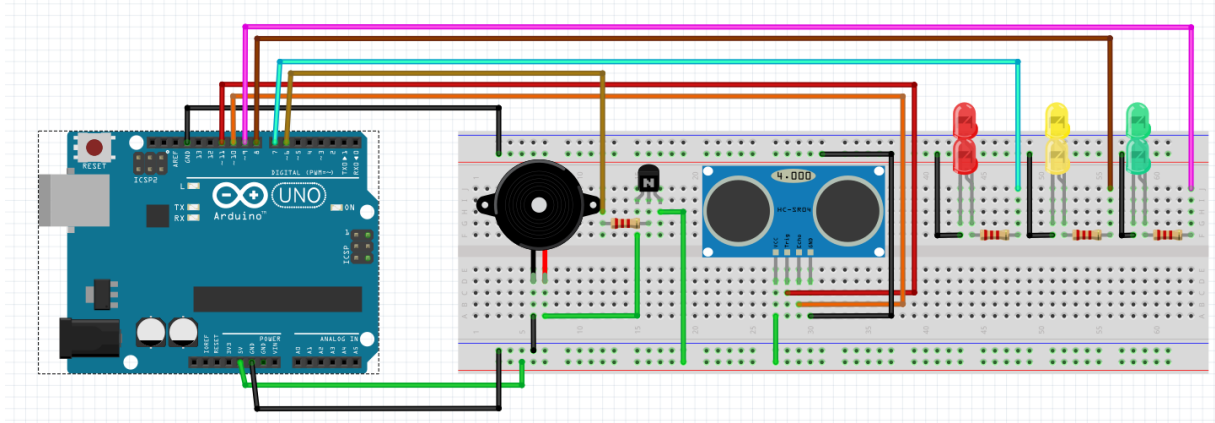
- Hangi devre elemanları ve sensörler kullanılabilir?
- Seçilen bu devre elemanları ve sensörler için uygun devre tasarımı nasıl olabilir?
- Bu devreyi kurabilmek için Arduino UNO'nun hangi pinleri nasıl kullanılabilir?
- Yazılacak Arduino IDE programının algoritması nasıl kurgulanabilir?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Devre elemanları ve sensörler üzerinde rahat çalışabilmek için breadboard üzerinde bağlantılar gerçekleştirilebilir.
- Sensörden gelen veriye göre LED'lerin ve hoparlörün çalışması if-else yapısı ile kontrol edilebilir.
- Hoparlör bağlantısı için transistör ile tetikleme yapılabilir.

2.2 Üret- Kapalı Otopark Park Destek Sistemi

Öğrenciler bu etkinlikteki yönergeleri tamamlayabilecek şekilde tasarımlarını yaptıktan sonra bilgisayar başında Arduino IDE üzerinde kodları yazarlar. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında farklı sensörlere ait kodların kullanımında ve çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.



Resim 4.33: Üret Etkinliği Örnek Devre Şeması

```

kapali_otopark_2
1  const int trigPin = 11;
2  const int echoPin = 10;
3  const int buzzPin = 6;
4  const int KLed = 7;
5  const int SLed = 8;
6  const int YLed = 9;
7  long duration;
8  long timer;
9  float distance;
10
11 void setup()
12 {
13   pinMode(trigPin, OUTPUT);
14   pinMode(echoPin, INPUT);
15   pinMode(buzzPin, OUTPUT);
16   Serial.begin(9600);
17 }
18 void loop()
19 {
20   digitalWrite(trigPin, LOW);
21   delayMicroseconds(2);
22   digitalWrite(trigPin, HIGH);
23   delayMicroseconds(10);
24   digitalWrite(trigPin, LOW);
25   duration = pulseIn(echoPin, HIGH);
26   distance = 0.034*(duration/2);
27
28   if (distance <= 10)
29   {
30     digitalWrite(buzzPin, HIGH);
31     digitalWrite(KLed, HIGH);
32   }
33   else if (distance <= 25 && distance > 10)
34   {
35     digitalWrite(buzzPin, HIGH);
36     digitalWrite(KLed, HIGH);
37     delay(50);
38     digitalWrite(buzzPin, LOW);
39     digitalWrite(KLed, LOW);
40     timer = distance* 10;
41     delay(timer);
42   }

```



```

42     else if (distance <= 45 && distance > 25)
43     {
44         digitalWrite(buzzPin, HIGH);
45         digitalWrite(SLed, HIGH);
46         delay(50);
47         digitalWrite(buzzPin, LOW);
48         digitalWrite(SLed, LOW);
49         timer = distance* 10;
50         delay(timer);
51     }
52     else
53     {
54         digitalWrite(buzzPin, LOW);
55         digitalWrite(YLed, HIGH);
56         delay(50);
57         digitalWrite(YLed, LOW);
58     }
59 }

```

Resim 4.34: Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Dikkat: Birinci değerlendirme sorusunun tartışılması çok önemlidir. Eğitimci bu hafta mutlaka bu sorunun tartışılması için en az 10 dakika ayırmalıdır. Sorunun cevabı temel olarak yazılmıştır. Fakat eğitimci öğrencilerin sınıfça soruyu tartışmalarını sağlamalıdır. Cevap doğrudan öğrencilere **kesinlikle** verilmemelidir. Tartışmada herhangi bir sonuca ulaşamadığı durumda eğitimci devreye girmelidir. Öğrenciler bu sorudaki devreleri hatırlamayabilir. Eğitimci devreleri tahtaya yansıtabilir.

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

Cevap: Birinde Arduino mikro işlemci olarak kullanılırken, diğerinde ise sadece güç kaynağı olarak kullanıldı. Arduino’nun güç kaynağı olarak kullanıldığı devrede (transistörlü) devre elemanları karanlıkta LED’in yanmasını sağladı. Arduino’nun mikrodenetleyici olarak kullanıldığı devrede ise devre elemanları Arduino IDE ortamı vasıtasıyla kodlandı. Bu kod ile devre çalıştı. Sadece 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. Fakat öte taraftan transistörlü devrenin maliyeti Arduino’nun mikrodenetleyici olarak kullanıldığı devreye göre daha ucuzdur. İhtiyaç ve maliyet dengesi gözlemlenerek koşula göre hangi devrenin kullanılacağına karar verilebilir.

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

3. PNP transistörün bağlı olduğu bir devrede Arduino UNO ile 12 V luk bir LED aydınlatmasının kontrol edilmesini istiyoruz. Ne tür bir sorunla karşılaşırız?
4. Aynı senaryoda NPN transistör kullanılırsa ne olur? Bu durumda LED'in parlaklığı nasıl ayarlanabilir?
5. Öğrencilerden seçilen sensörlerin ve devre elemanlarının özelliklerini anlatmaları ve bu sensörlerin hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
6. Bu hafta gördüğünüz devre elemanlarını günlük yaşamda nerelerde kullanabilirsiniz?
7. Tasarla-Üret etkinliklerinde seçilen sensörlerin ve devre elemanlarının Arduino IDE içerisindeki kod yazımlarında karşılaşılan güçlükler nelerdi?
8. Öğrencilerden şu aşamaya kadar öğrenilen tüm sensörlerin bir arada kullanılabileceği proje önerileri tartışmaları istenir.
9. Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilir.

4. İLAVE ETKİNLİK

4.1 Ses Sensörünü kullanarak kendi Ekolayzırımı yapıyorum

Bu etkinlikte öğrenciler 6 veya daha fazla LED kullanarak ses sensöründen aldıkları ses seviyesine bağlı olarak daha fazla LED'in yandığı bir devre tasarlayarak bunun Arduino IDE kodunu yazacaklardır. Burada amaç öğrencilerin cep telefonlarından açacakları müzikle uyumlu şekilde ses seviyesini LED'ler üzerinde gösteren bir ekolayzır devresi kurması ve kodlamasıdır. Öğrenciler ekolayzırlardaki gibi farklı renkte ve sayıda LED kullanabilirler. Etkinlikle ilgili örnek kod aşağıdaki resimde görülmektedir.

```

Ses_Sensoru_eko
1 int D0 = 1;
2 int DA = A0;
3 int ses1 =979;
4 int ses2 =980;
5 int ses3 =981;
6 int ses4 =982;
7 int ses5 =983;
8 int ses6 =984;
9
10 int sesdegeri = 0;
11
12 void setup() {
13   pinMode(2, OUTPUT);
14   pinMode(3, OUTPUT);
15   pinMode(4, OUTPUT);
16   pinMode(5, OUTPUT);
17   pinMode(6, OUTPUT);
18   pinMode(7, OUTPUT);
19 }
20 void loop() {
21   sesdegeri = analogRead(DA);
22   if (sesdegeri >= ses1){
23     digitalWrite(2, HIGH);
24   }
25   else{
26     digitalWrite(2, LOW);
27   }
28   if (sesdegeri >= ses2){
29     digitalWrite(3, HIGH);
30   }
31   else{
32     digitalWrite(3, LOW);
33   }
34   if (sesdegeri >= ses3){
35     digitalWrite(4, HIGH);
36   }
37   else{
38     digitalWrite(4, LOW);
39   }
40   if (sesdegeri >= ses4){
41     digitalWrite(5, HIGH);
42   }
43   else{
44     digitalWrite(5, LOW);
45   }
46   if (sesdegeri >= ses5){
47
48     digitalWrite(6, HIGH);
49   }
50   else{
51
52     digitalWrite(6, LOW);
53   }
54   if (sesdegeri >= ses6){
55     digitalWrite(7, HIGH);
56   }
57   else{
58     digitalWrite(7, LOW);
59   }
60 }

```

Resim 4.35: İlave Etkinlik Örnek Kodu

4.2 Ambulans Sirenine Duyarlı Kavşak

Bu etkinlikte öğrencilerden 2'li bir kavşakta trafik ışığı tasarlamasını istenmektedir. Fakat bu kavşakta yer alan sensörler ambulans sireni algıladığında kavşakta hangi taraf yeşil olursa olsun ambulansın geldiği yöndeki trafik ışığını uygun bir süre içerisinde yeşil yapmalıdır. Diğer yoldaki trafik ışığı ise kırmızıya dönmelidir

5.Bölüm - Ekran ve Gösterge Araçlarının Kullanımı

Ön bilgi:

- Öğrenciler algoritma mantığını bilir.
- Öğrenciler temel programlama kavramlarını bilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.
- Öğrenciler Arduino UNO'yu tanır ve bunu kullanarak devreleri kurup kodlarında düzenleme yapabilirler.
- Öğrenciler Arduino IDE arayüzü ile değişkenler, if/else, for döngüsü, dizi vb. yöntemleri ile kod yazarak elektronik devre elemanlarını yönetebilir.

Bölüm Kazanımları:

- Ekranların görevini ve çalışma prensiplerini bilir.
- 7 Segment Display gösterge aracının çalışma prensibini bilir ve bu aracı kullanabilir.
- Arduino IDE arayüzünde "kütüphane" kavramını bilir ve kod yazımında kullanabilir.
- LCD ekranların çalışma prensiplerini bilir ve uygulamalarda çıktı elemanı olarak kullanabilir.
- LCD ekranlar için I2C devresinin görevini ve çalışma prensibini bilir.
- 7 Segment Display ve LCD ekran araçları ile farklı uygulamalarda devreler kurabilir ve Arduino IDE arayüzü ile kodlar yazarak bu devre elemanlarını projelerde kullanabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle 7 segment gösterge, LCD ekranlar, I2C modülü ve protokolü, kütüphane kavramı ve kullanımı, farklı yapılarda I2C modülü entegreli LCD ekranlar hakkında bilgi sahibi olmasıdır. Ayrıca elektronik gösterge araçlarıyla ve Arduino UNO kartıyla farklı etkinlikler ve projelerde kullanıcılara ve dış dünyaya verilerin gösterilmesinde kullanabileceği yeterliliklerin öğrencilere kazandırılması amaçlanmaktadır. Bu kapsamda programlamada farklı kontrol yapıları, tek ve çift boyutlu diziler ve switch-case gibi konuların uygulamalar içerisindeki kullanımlarını da deneyimlemeleri hedefler içerisinde.

Göze ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	8 Segment Gösterge
Breadboard	I2C Entegreli 2x16 LCD Ekran
220 ohm direnç	I2C Entegreli 4x20 LCD Ekran
330 ohm direnç	Erkek-Erkek bağlantı kablosu
Buzzer	Erkek-Dişi bağlantı kablosu
Hoparlör	Buton

Haftanın İşlenişi:

Gözle: Ekranlar (Gösterim araçları), 7 segment gösterge, I2C modülü entegreli LCD Ekranlar ile kütüphane yapılarının tanımlarının yapılması ve özelliklerini verilmesi.

Uygula: 7 segment gösterge ve I2C modülü entegreli LCD ekranlar ile örnek devrelerin oluşturulması ve programlanması.

Tasarla: I2C modülü entegreli 2x16 ve 4x20 LCD ekranlar, 7 segment gösterge ve butonları kullanarak Basketbol Skorboard uygulamasını tasarımının gerçekleştirilmesi.

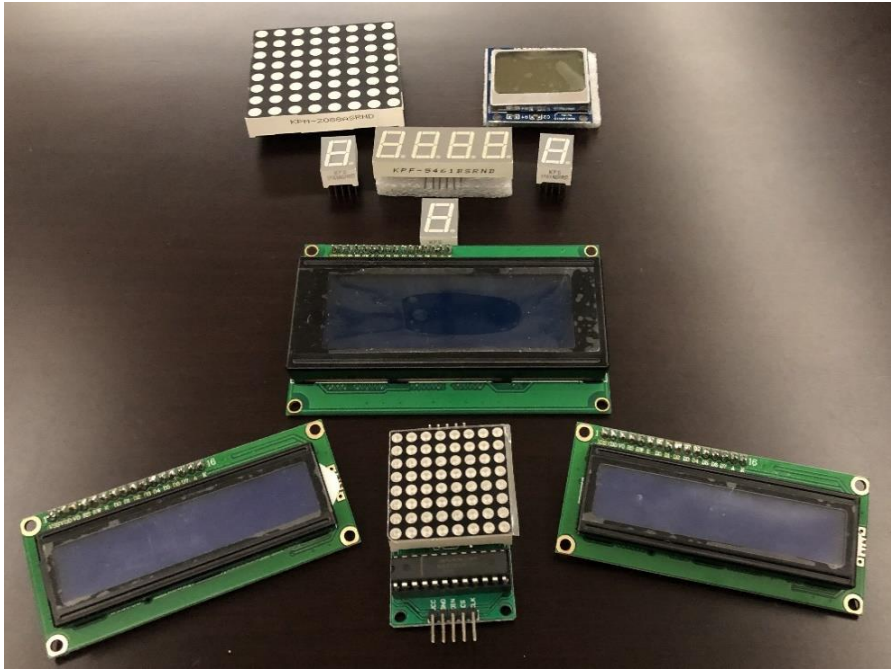
Üret: I2C modülü entegreli 2x16 ve 4x20 LCD ekranlar, 7 segment gösterge ve butonları kullanarak Basketbol Skorboard tasarımını gerçekleştirme; devrelerini oluşturma ve Arduino IDE içerisinde program kodlarının yazılması.

Değerlendir: Yansıtma Etkinliği.

1. ADIM: GÖZLE ve UYGULA

1.1 Gözle – Ekranlar (Gösterim Araçları)

Günümüzde çevremizdeki birçok elektronik cihazda üretilen veya sensörlerden gelen verilerin kullanıcılara iletilmek istendiğinde çıktı birimleri olarak ekranlardan yani gösterim araçlarından faydalanılır. En basit manada dijital termometrelerden mutfak robotlarına, numaratorlerden saatlere, telefonlardan bilgisayarlara her türlü elektronik cihazda çeşitli ekran göstergeler kullanılmaktadır. Elektronik Programlama ve Nesnelerin İnterneti dersi kapsamında Arduino UNO ile gerçekleştirilecek elektronik devre ve projelerde bu göstergelerden 7 segment gösterge, LCD ekran ve 8x8 nokta LED göstergelerin kullanılacağı eğitmen tarafından öğrencilere açıklanarak sırasıyla birer parça gösterilir. Bu haftaya kadar Arduino UNO ile birçok uygulamada veri ve çıktıları göstermek için Arduino IDE uygulamasındaki “Serial Port” aracılığı ile bilgisayar ekranının kullanıldığı, bu hafta ise gösterim araçları devre elemanlarının ve programlanmasının öğrenilmesi ile daha sonraki uygulamalarda ve projelerde bu ekranların kullanılabilceği açıklanacaktır. Aşağıdaki resimde farklı ekran türleri görülmektedir.

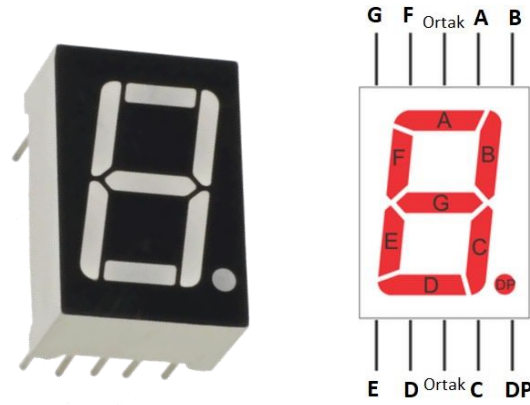


Resim 5.1: DENEYAP Setlerinde Yer Alan Farklı Ekran Türleri

1.2 Gözle – 7 Bölümlü Gösterge (7 Segment Display)

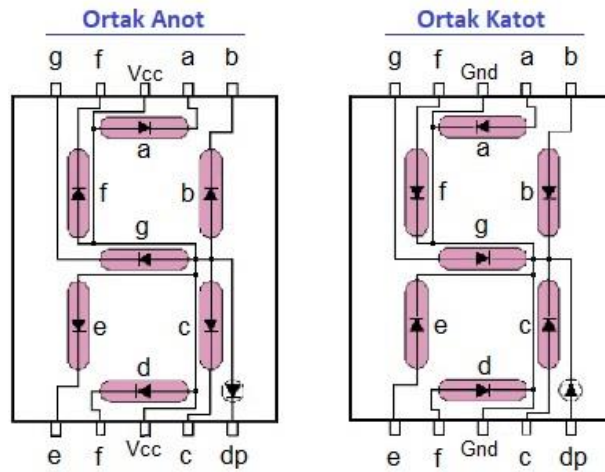
7 Segment Gösterge

Birçok teknolojik cihazda özellikle rakam, sayı ve saatlerin gösteriminde 7 Bölümlü Gösterge (7 Segment Display) kullanılmaktadır. Bu göstergeler genel olarak üzerinde ayrı ayrı kullanılabilen 7 şerit (çubuk) LED'den oluştuğu için 7 Segment Display olarak adlandırılmaktadır. 7 çubuk LED'e ek olarak sağ alt kısmında ondalık sayıların gösterilebilmesi için bir nokta LED daha bulunmaktadır. Yani bu gösterim araçlarında 8 ayrı LED devresi bulunmaktadır. Aşağıdaki resimde de gösterildiği gibi her LED birer harf ile isimlendirilerek hangi pinin hangi LED'i kontrol ederek yakacağı açıklanmıştır. Bu pinler aracılığıyla kontrol edilen LED'ler çeşitli kombinasyonlarla yakılarak rakamlar oluşturulmaktadır.



Resim 5.2: 7 Segment Gösterge ve Ayak Yapısı

7 Segment Göstergelerde bulunan 8 devre; daha az pin kullanılabilmesi ve daha işlevsel kontrol edilebilmesi için ortak artı (+) veya eksi (-) pinleri birleştirilmiştir. Ortak pinlerin birleştirilme biçimine göre **ortak anot** veya **ortak katot** olarak adlandırılan 2 çeşit 7 segment gösterge bulunmaktadır. Eğer LED devrelerinin “+” uçları birleştirilip alt ve üst sıralardaki pinlerin ortasında yer alan ortak pinlerde bağlanıyorsa ortak anot 7 segment gösterge, “-“ uçlar birleştirilip alt ve üst ortak pinde birleştiriliyorsa ortak katot 7 segment gösterge olarak isimlendirilmektedir. Aşağıdaki resimde 7 segment göstergelerde ortak anotsa orta pinlere 5 V enerji verilirken diğer pinler GND'ye bağlanmalıdır. Ortak katot 7 segment göstergelerde ise orta pinler GND'ye bağlanırken, diğer pinlere 5 V enerji verilmelidir.



Resim 5.3: Ortak Anot ve Katot 7 Segment Gösterge Ayak Yapısı

1.3 Gözle ve Uygula - 7 Segment Göstergenin ortak anot veya ortak katot olduğunu tespit etme (Öğrenci 1)

Malzeme Listesi

Arduino UNO

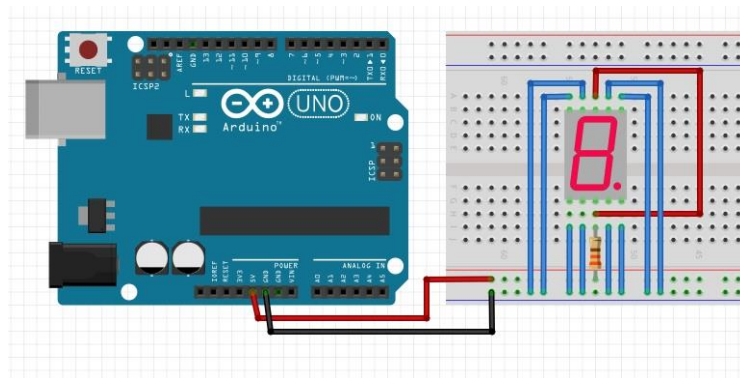
Breadboard

Bağlantı kabloları

330 ohm direnç

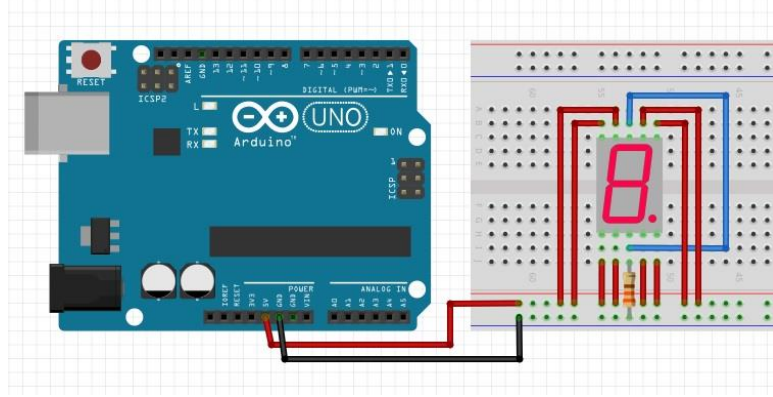
7 Segment gösterge

Ortak anot ve ortak katot 7 segment göstergelerde devreleri tamamlanan her LED yanarak rakamları oluşturmaktadır. Rakamları oluşturmaya başlamadan önce 7 segment göstergenin ortak anot mu, yoksa ortak katot bir 7 segment gösterge olduğunu anlamak için iki farklı devre kurulacaktır. Öncelikle 7 segment göstergenin ortak anot bir devre elemanı olduğunun tespiti için aşağıdaki resimlerdeki ilk devre kurulur. Bu devre ortak anot 7 segment gösterge için kurulmuştur ve devreye enerji verildiğinde göstergenizdeki LED'ler yanıyor ise göstergeniz ortak anot bir 7 segment göstergedir.



Resim 5.4: Gözle-Uygula Etkinliği Örnek Devre Şeması (Ortak Anot)

Eğer ışık yanmıyor ise, göstergenizin ortak katot bir gösterge olabilir ve bunun tespiti için de aşağıdaki resimde görülen devre kurulumu. Devreye enerji verildiğinde göstergedeki LED'ler yanmıyorsa 7 segment gösterge ortak katot bir devre elemanıdır.



Resim 5.5: Gözle-Uygula Etkinliği Örnek Devre Şeması (Ortak Katot)

Eğitmen kendi setindeki 7 segment displaylerin ortak anot mu ortak katot mu olduğunu öğrencilerin görebileceği şekilde yukarıdaki iki devreyi kontrol edip belirledikten sonra öğrencilerin de setlerindeki 7 segment göstergelerin türünü belirlemelerini isteyecek ve bu süreçte öğrencilere ihtiyaç duymaları halinde yardımcı olacaktır. Genellikle setteki 7 segment göstergeler ortak katot bir devre elemanı olmasına rağmen, bu bilgiyi öğrencilerle paylaşmayıp, öğrencilerden ellerindeki göstergeleri ayrı ayrı test ederek türlerini belirlemelerini sağlamalıdır.

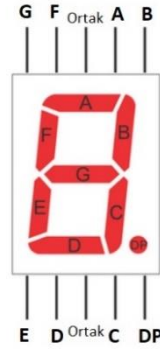
Dikkat

7 Segment Göstergelerde LED'lerin daha verimli çalışması için her LED devresine uygun bir direnç bağlanması gerektiği öğrencilere eğitmen tarafından hatırlatılır. Burada ortak katot 7 Segment Göstergelerde her LED'e ayrı bir direnç bağlamak yerine ortak GND pinine bir direnç(220 ohm veya 330 ohm) bağlanabileceği öğrencilere açıklanabilir. Aynı şekilde ortak anot 7 Segment Göstergelerde de ortak anot pinlerine direnç bağlanabileceği belirtilir.

1.4 Gözle ve Uygula - 7 Segment Gösterge LED'leri yakıp söndürme (Öğrenci 2)

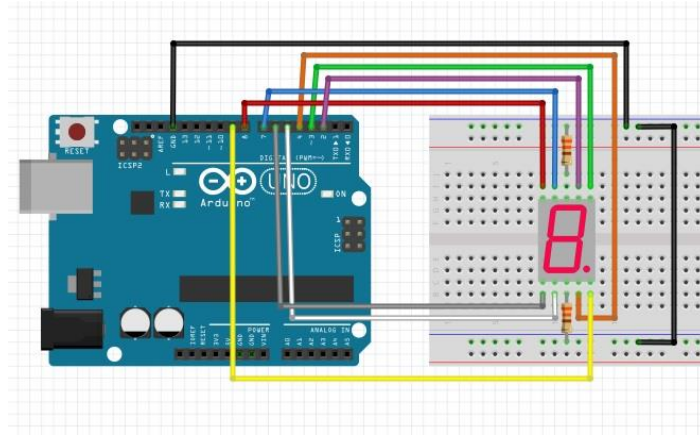
Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
330 ohm direnç
7 Segment gösterge

7 Segment Göstergenin ortak anot mu, ortak katot mu olduğunun belirlenmesinden sonra bu etkinlikte öğrencilerle 7 segment göstergelerde tüm LED'leri yakma ve söndürme kontrol işlemleri gerçekleştirilecektir. Gösterge üzerindeki 7 LED'in "A", "B", "C", "D", "E", "F" ve "G" harfleri ile isimlendirildiği, ayrıca nokta LED'in de "DP" olarak ifade edildiği aşağıdaki görsel veya benzeri aracılığı ile öğrencilere açıklanır. Burada bu LED'lerin oluşturulmak istenen rakama göre yanacağı veya söneceği ve bunun kontrolünün de Arduino UNO üzerindeki dijital pinler aracılığıyla kontrol edileceği öğrencilere aktarılır. Aşağıdaki resimde segment gösterge ayak yapısı görülmektedir.



Resim 5.6: 7 Segment Gösterge Ayak Yapısı

7 segment gösterge bağlantılarıyla ilgili bilgilendirme yapıldıktan sonra tüm LED'lerin öncelikle sırayla 1 saniye yanıp 1 saniye sönmeleri ve sonrasında tüm LED'lerin aynı anda 3 defa 1 saniye yanıp 1 saniye sönmelerini sağlamak için; aşağıdaki resimde görülen devre eğitmen tarafından kurulur ve öğrencilerin de daha sonra bu devreyi kendi setlerindeki malzemelerle oluşturmaları istenir.



Resim 5.7: Gözle-Uygula Etkinliği Örnek Devre Şeması

Dikkat

7 Segment Gösterge ile gerçekleştirilecek uygulama ve etkinliklerde genel olarak setlerdeki 7 segment göstergelerin ortak katot olduğu varsayıldığından tüm devre çizimlerinde sadece ortak katot 7 segment gösterge temel alınarak devre gösterimleri yapılmıştır. Eğitimci bu konuda öğrencileri uyarıp ellerindeki göstergelerin ortak katot veya ortak anot olma durumlarına dikkat ederek devrelerini oluşturmalarını isteyecektir.

Yine Arduino IDE ile hazırlanan kodlar da ortak katot 7 segment göstergeler için hazırlanmış olup, eğitimci ortak anot gösterge kullanıldığında örnek olarak verilen kod satırlarında gerekli değişiklikleri yapmalıdır.

Yukarıdaki devre kurulduktan sonra istenen görevi yerine getiren (tüm LED'ler öncelikle sırayla 1 saniye yanıp 1 saniye sönecek, sonra da tüm LED'lerin aynı anda 3 defa 1 saniye yanıp 1 saniye sönecek) kodlar Arduino IDE'de yazılarak öncelikle eğitimci tarafından öğrencilere gösterilir, sonrasında da öğrencilerin kendi bilgisayarlarında bu kodları yazarak Arduino UNO ile kurdukları devrelerine yüklemeleri ve 7 segment göstergelerinin istenen şekilde yanıp söndüğünü gözlemlemeleri istenir. Etkinlikle ilgili örnek kod aşağıdaki resimde görülmektedir.

```

7_segment_LED_yakma$
1 #define a 2
2 #define b 3
3 #define c 4
4 #define d 5
5 #define e 6
6 #define f 7
7 #define g 8
8 #define dp 9
9
10 void setup() {
11   pinMode(a, OUTPUT);
12   pinMode(b, OUTPUT);
13   pinMode(c, OUTPUT);
14   pinMode(d, OUTPUT);
15   pinMode(e, OUTPUT);
16   pinMode(f, OUTPUT);
17   pinMode(g, OUTPUT);
18   pinMode(dp, OUTPUT);
19 }
20
21 void loop() {
22   digitalWrite(a,HIGH);
23   delay(1000);
24   digitalWrite(a,LOW);
25   delay(1000);
26   digitalWrite(b,HIGH);
27   delay(1000);
28   digitalWrite(b,LOW);
29   delay(1000);
30   digitalWrite(c,HIGH);
31   delay(1000);
32   digitalWrite(c,LOW);
33   delay(1000);
34   digitalWrite(d,HIGH);
35   delay(1000);
36   digitalWrite(d,LOW);
37   delay(1000);
38   digitalWrite(e,HIGH);
39   delay(1000);
40   digitalWrite(e,LOW);
41   delay(1000);
42   digitalWrite(f,HIGH);
43   delay(1000);
44   digitalWrite(f,LOW);
45   delay(1000);
46   digitalWrite(g,HIGH);
47   delay(1000);
48   digitalWrite(g,LOW);
49   delay(1000);
50   digitalWrite(dp,HIGH);
51   delay(1000);
52   digitalWrite(dp,LOW);
53   delay(1000);
54
55   for(int i=0; i<3; i++){
56     digitalWrite(a,HIGH);
57     digitalWrite(b,HIGH);
58     digitalWrite(c,HIGH);
59     digitalWrite(d,HIGH);
60     digitalWrite(e,HIGH);
61     digitalWrite(f,HIGH);
62     digitalWrite(g,HIGH);
63     digitalWrite(dp,HIGH);
64     delay(1000);
65     digitalWrite(a,LOW);
66     digitalWrite(b,LOW);
67     digitalWrite(c,LOW);
68     digitalWrite(d,LOW);
69     digitalWrite(e,LOW);
70     digitalWrite(f,LOW);
71     digitalWrite(g,LOW);
72     digitalWrite(dp,LOW);
73     delay(1000);
74   }
75 }
76
Done uploading.
Sketch uses 1310 bytes (4%) of program storage space.
Global variables use 9 bytes (0%) of dynamic memory.
76 Arduino/Genuino Uno on COM4

```

Resim 5.8: Gözle-Uygula Etkinliği Örnek Kodu

1.5 Gözle ve Uygula - 7 Segment Gösterge ile rakamları oluşturma (Öğrenci 1)

Malzeme Listesi

Arduino UNO

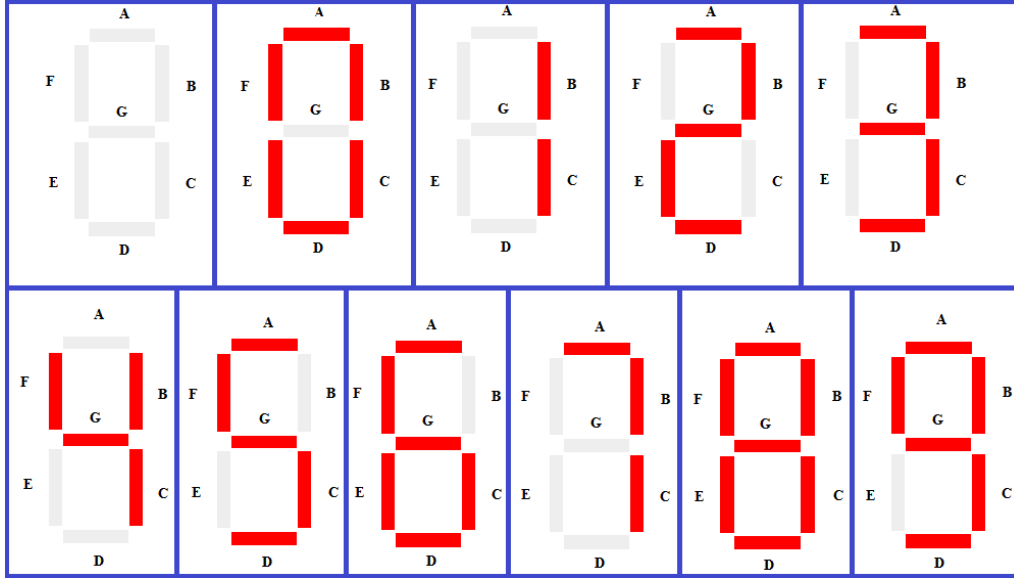
Breadboard

Bağlantı kabloları

330 ohm direnç

7 Segment gösterge

Bu etkinlikte öncelikle öğrencilere 7 segment göstergelerde rakamların oluşturulma mantığı açıklanacaktır. Gösterge üzerindeki 7 LED'in ("A", "B", "C", "D", "E", "F" ve "G" harfleri ile isimlendirilen) oluşturulmak istenen rakama göre yanacağı veya söneceği ve bunun kontrolünün de Arduino UNO üzerindeki dijital pinler aracılığıyla kontrol edileceği öğrencilere aktarılır. Aşağıdaki resimde ise 7 Segment Göstergede her rakam için hangi LED'lerin yanması gerektiğini açıklanmaktadır. Buradan da anlaşılacağı gibi A'dan G'ye kadar olan LED'ler Arduino IDE üzerindeki dijital pinlere bağlandığında ve bu LED'lerin yanıp sönmeye kontrol edildiğinde kullanıcıya verilmek istenen bilgi 7 segment gösterge üzerinde rakam olarak gösterilebilmektedir.

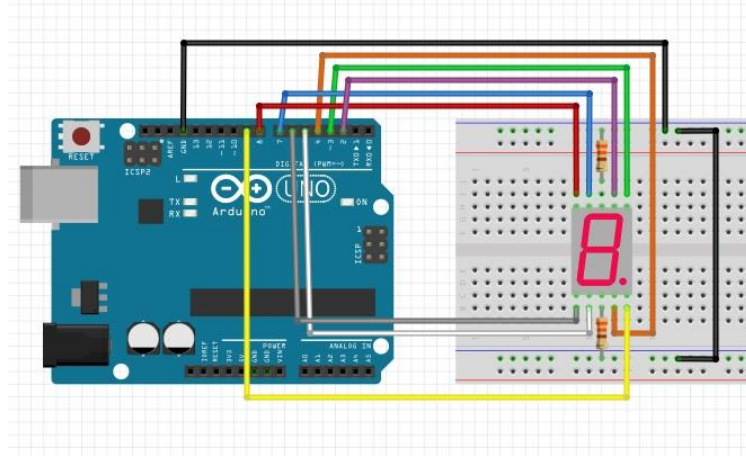


Resim 5.9: 7 Segment Göstergede Rakamların Gösterimi

Örnek olarak ortak katot 7 segment göstergede “0” rakamı gösterilmek isteniyorsa A, B, C, D, E, F ve G LED’lerine Arduino UNO üzerinde bağlanılan dijital pinlerden **5 V (HIGH)** gerilimi verilecekken, G LED’ine **0 V (LOW)** gerilim verilmelidir. Yine aşağıda her rakam için hangi LED’lerin “HIGH” olarak kodlanacağı, hangi LED’lerin ise “LOW” olarak kodlanacağını gösteren tablo öğrencilere açıklanmalıdır.

Rakam	LED Durumu	Rakam	LED Durumu	Rakam	LED Durumu	Rakam	LED Durumu
0	A – HIGH B – HIGH C – HIGH D – HIGH E – HIGH F – HIGH G - LOW	1	A – LOW B – HIGH C – HIGH D – LOW E – LOW F – LOW G - LOW	2	A – HIGH B – HIGH C – LOW D – HIGH E – HIGH F – LOW G – HIGH	3	A – HIGH B – HIGH C – HIGH D – HIGH E – LOW F – LOW G - HIGH
4	A – LOW B – HIGH C – HIGH D – LOW E – LOW F – HIGH G – HIGH	5	A – HIGH B – LOW C – HIGH D – HIGH E – LOW F – HIGH G - HIGH	6	A – HIGH B – LOW C – HIGH D – HIGH E – HIGH F – HIGH G – HIGH	7	A – HIGH B – HIGH C – HIGH D – LOW E – LOW F – LOW G - LOW
8	A – HIGH B – HIGH C – HIGH D – HIGH E – HIGH F – HIGH G - HIGH	9	A – HIGH B – HIGH C – HIGH D – HIGH E – LOW F – HIGH G - HIGH				

Daha sonra eğitmen 7 segment göstergede rakamların 0'dan 9'a sırayla 1 saniye yanıp, 1 saniye söndükten sonra nokta LED'inin de 3 defa 1 saniye yanıp 1 saniye söneceği bir devreyi Arduino UNO ile kurarak ve Arduino IDE'de bu görevi yerine getirecek kodlamayı yapar ve öğrencilerin de aynı devre ve kodlamayı yapmalarını ister. Aşağıdaki resimde etkinlikle ilgili örnek devre şeması görülmektedir.



Resim 5.10: Gözle-Uygula Etkinliği Örnek Devre Şeması

Yukarıda yer alan şekildeki devre bir önceki etkinlik kapsamında kurulmuş olduğu için aynı devre bu uygulama için de kullanılabilir. Rakamları sırayla yakıp söndürecek program için her rakamı ve nokta LED'ini yakacak bir fonksiyon tanımlanabilir. Ayrıca yine tüm LED'lerin söneceği bir fonksiyon da tanımlanabilir. Aşağıdaki resimde bu görevi yerine getiren fonksiyonlardan oluşan Arduino IDE kodları sunulmuştur. Aşağıdaki kod satırlarında özellikle uzun kod yazımı kullanılmış olup, öğrencilere 7 segment göstergeyi ilk etapta en uzun yöntemle programlama öğrencilere gösterilecektir, çünkü daha sonra farklı kod yazma teknikleri ile daha az satırda elektronik devre elemanları programlamanın önemi vurgulanacaktır.


```

7_segment_sayim
1 #define a 2
2 #define b 3
3 #define c 4
4 #define d 5
5 #define e 6
6 #define f 7
7 #define g 8
8 #define dp 9
9
10 void setup() {
11   pinMode(a, OUTPUT);
12   pinMode(b, OUTPUT);
13   pinMode(c, OUTPUT);
14   pinMode(d, OUTPUT);
15   pinMode(e, OUTPUT);
16   pinMode(f, OUTPUT);
17   pinMode(g, OUTPUT);
18   pinMode(dp, OUTPUT);
19 }
20
21 void loop() {
22   bir();
23   off();
24   iki();
25   off();
26   uc();
27   off();
28   dort();
29   off();
30   bes();
31   off();
32   alti();
33   off();
34   yedi();
35   off();
36   sekiz();
37   off();
38   dokuz();
39   off();
40   nokta();
41   off();
42   nokta();
43   off();
44   nokta();
45   off();
46 }
47 void bir() {
48   digitalWrite(a, LOW);
49   digitalWrite(b, HIGH);
50   digitalWrite(c, HIGH);
51   digitalWrite(d, LOW);
52   digitalWrite(e, LOW);
53   digitalWrite(f, LOW);
54   digitalWrite(g, LOW);
55   digitalWrite(dp, LOW);
56   delay(1000);
57 }
58 void iki() {
59   digitalWrite(a, HIGH);
60   digitalWrite(b, HIGH);
61   digitalWrite(c, LOW);
62   digitalWrite(d, HIGH);
63   digitalWrite(e, HIGH);
64   digitalWrite(f, LOW);
65   digitalWrite(g, HIGH);
66   digitalWrite(dp, LOW);
67   delay(1000);
68 }
69 void uc() {
70   digitalWrite(a, HIGH);
71   digitalWrite(b, HIGH);
72   digitalWrite(c, HIGH);
73   digitalWrite(d, HIGH);
74   digitalWrite(e, LOW);
75   digitalWrite(f, LOW);
76   digitalWrite(g, HIGH);
77   digitalWrite(dp, LOW);
78   delay(1000);
79 }
80
81 void dort() {
82   digitalWrite(a, LOW);
83   digitalWrite(b, HIGH);
84   digitalWrite(c, HIGH);
85   digitalWrite(d, LOW);
86   digitalWrite(e, LOW);
87   digitalWrite(f, HIGH);
88   digitalWrite(g, HIGH);
89   digitalWrite(dp, LOW);
90   delay(1000);
91 }
92 void bes() {
93   digitalWrite(a, HIGH);
94   digitalWrite(b, LOW);
95   digitalWrite(c, HIGH);
96   digitalWrite(d, HIGH);
97   digitalWrite(e, LOW);
98   digitalWrite(f, HIGH);
99   digitalWrite(g, HIGH);
100  digitalWrite(dp, LOW);
101  delay(1000);
102 }
103 void alti() {
104   digitalWrite(a, HIGH);
105   digitalWrite(b, LOW);
106   digitalWrite(c, HIGH);
107   digitalWrite(d, HIGH);
108   digitalWrite(e, HIGH);
109   digitalWrite(f, HIGH);
110   digitalWrite(g, HIGH);
111   digitalWrite(dp, LOW);
112   delay(1000);
113 }
114 void yedi() {
115   digitalWrite(a, HIGH);
116   digitalWrite(b, HIGH);
117   digitalWrite(c, HIGH);
118   digitalWrite(d, LOW);
119   digitalWrite(e, LOW);
120   digitalWrite(f, LOW);
121   digitalWrite(g, LOW);
122   digitalWrite(dp, LOW);
123   delay(1000);
124 }
125
126 void sekiz() {
127   digitalWrite(a, HIGH);
128   digitalWrite(b, HIGH);
129   digitalWrite(c, HIGH);
130   digitalWrite(d, HIGH);
131   digitalWrite(e, HIGH);
132   digitalWrite(f, HIGH);
133   digitalWrite(g, HIGH);
134   digitalWrite(dp, LOW);
135   delay(1000);
136 }
137 void dokuz() {
138   digitalWrite(a, HIGH);
139   digitalWrite(b, HIGH);
140   digitalWrite(c, HIGH);
141   digitalWrite(d, HIGH);
142   digitalWrite(e, LOW);
143   digitalWrite(f, HIGH);
144   digitalWrite(g, HIGH);
145   digitalWrite(dp, LOW);
146   delay(1000);
147 }
148
149 void nokta() {
150   digitalWrite(a, LOW);
151   digitalWrite(b, LOW);
152   digitalWrite(c, LOW);
153   digitalWrite(d, LOW);
154   digitalWrite(e, LOW);
155   digitalWrite(f, LOW);
156   digitalWrite(g, LOW);
157   digitalWrite(dp, HIGH);
158   delay(1000);
159 }
160
161 void off() {
162   digitalWrite(a, LOW);
163   digitalWrite(b, LOW);
164   digitalWrite(c, LOW);
165   digitalWrite(d, LOW);
166   digitalWrite(e, LOW);
167   digitalWrite(f, LOW);
168   digitalWrite(g, LOW);
169   digitalWrite(dp, LOW);
170   delay(1000);
171 }
172
Done Saving
160 Arduino/Genuino Uno on COM4

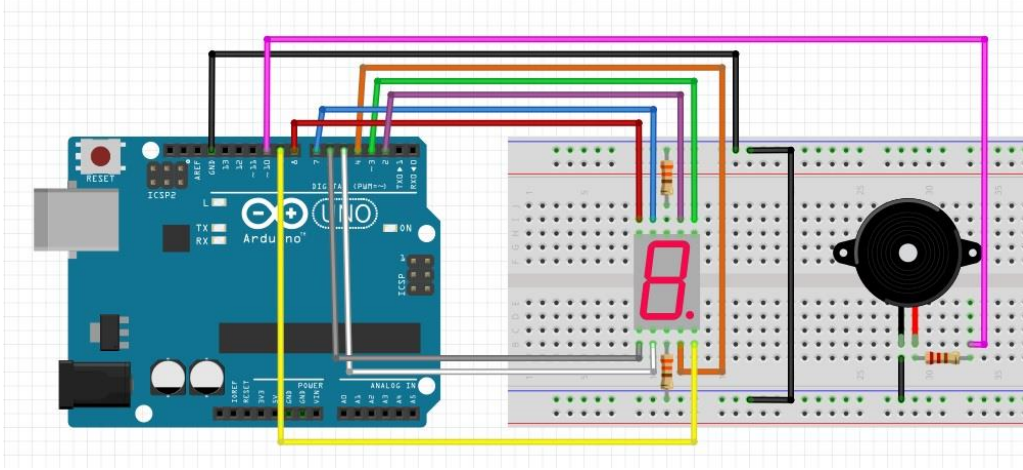
```

Resim 5.11: Gözle-Uygula Etkinliği Örnek Kodu

1.6 Uygula- 7 Segment Gösterge ve Buzzer ile geri sayım (Öğrenci 2)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
330 ohm direnç
220 ohm direnç
7 Segment gösterge
Buzzer veya Hoparlör

Bu uygulamada ise Elektronik Programlama ve Nesnelerin İnterneti dersinin öğretim tasarımı modelindeki eşli programlama mantığına uymak için bir önceki uygulamanın tam tersine geri sayım yapan bir uygulamayı devreye bir Buzzer veya Hoparlör bağlayarak diğer öğrencinin yapması sağlanacaktır. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde verilmiştir.



Resim 5.12: Uygula Etkinliği Örnek Devre Şeması

Bu uygulamada kurulan devre Arduino IDE’de aşağıdaki kuralları yerine getirecek şekilde kodlanacaktır:

- 9’dan 0’a sayılar bir saniye yanarak azalacak,
- Her sayı arasında 7 segment gösterge 0,25 saniye yanmayacak,
- 0’dan sonra nokta LED ve beraberinde buzzer (hoparlör) önce 0,5 saniye, sonra 0,25 saniye, sonra 0,15 saniye, sonra 0,10 saniye, sonra 0,05 saniye aktif (1) olup yine aynı sürelerde pasif (0) olacaktır.
- Daha sonra ise nokta LED ve buzzer (hoparlör) 3 defa 0,05 saniye aktif (1) olup 0,05 saniye pasif (0) olacaktır.
- En son olarak ise nokta LED ve buzzer (hoparlör) 3 defa 2,5 saniye aktif (1) olup 2,5 saniye pasif (0) olacaktır.

Etkinlikle ilgili örnek kodlar aşağıdaki resimde görülmektedir.

```

7_segment_gerisayim §
1 #define a 2
2 #define b 3
3 #define c 4
4 #define d 5
5 #define e 6
6 #define f 7
7 #define g 8
8 #define dp 9
9 #define buzzerPin 10
10
11 void setup() {
12   pinMode(a, OUTPUT);
13   pinMode(b, OUTPUT);
14   pinMode(c, OUTPUT);
15   pinMode(d, OUTPUT);
16   pinMode(e, OUTPUT);
17   pinMode(f, OUTPUT);
18   pinMode(g, OUTPUT);
19   pinMode(dp, OUTPUT);
20   pinMode(buzzerPin, OUTPUT);
21 }
22
23 void loop() {
24   dokuz();
25   off();
26   delay(250);
27   sekiz();
28   off();
29   delay(250);
30   yedi();
31   off();
32   delay(250);
33   alti();
34   off();
35   delay(250);
36   bes();
37   off();
38   delay(250);
39   dort();
40   off();
41   delay(250);
42   uc();
43   off();
44   delay(250);
45   iki();
46   off();
47   delay(250);
48   bir();
49   off();
50   delay(250);
51   sifir();
52   delay(1000);
53   off();
54   delay(250);
55   nokta();
56   tone(buzzerPin, 440);
57   delay(1000);
58   off();
59   noTone(buzzerPin);
60   delay(500);
61   nokta();
62   tone(buzzerPin, 440);
63   delay(500);
64   off();
65   noTone(buzzerPin);
66   delay(250);
67   nokta();
68   tone(buzzerPin, 440);
69   delay(250);
70   off();
71   noTone(buzzerPin);
72   delay(150);
73   nokta();
74   tone(buzzerPin, 440);
75   delay(150);
76   off();
77   noTone(buzzerPin);
78   delay(100);
79   nokta();
80   tone(buzzerPin, 440);
81   delay(100);
82   off();
83   noTone(buzzerPin);
84   delay(50);
85   nokta();
86   tone(buzzerPin, 440);
87   delay(50);
88   off();
89   noTone(buzzerPin);
90   delay(50);
91   nokta();
92   tone(buzzerPin, 440);
93   delay(50);
94   off();
95   noTone(buzzerPin);
96   delay(50);
97   nokta();
98   tone(buzzerPin, 440);
99   delay(50);
100  off();
101  noTone(buzzerPin);
102  delay(50);
103  nokta();
104  tone(buzzerPin, 440);
105  delay(2500);
106  off();
107  noTone(buzzerPin);
108  delay(2500);
109 }
110
111 void bir() {
112   digitalWrite(a, LOW);
113   digitalWrite(b, HIGH);
114   digitalWrite(c, HIGH);
115   digitalWrite(d, LOW);
116   digitalWrite(e, LOW);
117   digitalWrite(f, LOW);
118   digitalWrite(g, LOW);
119   digitalWrite(dp, LOW);
120   delay(1000);
121 }
122
123 void iki() {
124   digitalWrite(a, HIGH);
125   digitalWrite(b, HIGH);
126   digitalWrite(c, LOW);
127   digitalWrite(d, HIGH);
128   digitalWrite(e, HIGH);
129   digitalWrite(f, LOW);
130   digitalWrite(g, HIGH);
131   digitalWrite(dp, LOW);
132   delay(1000);
133 }
134
135 void uc() {
136   digitalWrite(a, HIGH);
137   digitalWrite(b, HIGH);
138   digitalWrite(c, HIGH);
139   digitalWrite(d, HIGH);
140   digitalWrite(e, LOW);
141   digitalWrite(f, LOW);
142   digitalWrite(g, HIGH);
143   digitalWrite(dp, LOW);
144   delay(1000);
145 }
146
147 void dort() {
148   digitalWrite(a, LOW);
149   digitalWrite(b, HIGH);
150   digitalWrite(c, HIGH);
151   digitalWrite(d, LOW);
152   digitalWrite(e, LOW);
153   digitalWrite(f, HIGH);
154   digitalWrite(g, HIGH);
155   digitalWrite(dp, LOW);
156   delay(1000);
157 }
158
159 void bes() {
160   digitalWrite(a, HIGH);
161   digitalWrite(b, LOW);
162   digitalWrite(c, HIGH);
163   digitalWrite(d, HIGH);
164   digitalWrite(e, LOW);
165   digitalWrite(f, HIGH);
166   digitalWrite(g, HIGH);
167   digitalWrite(dp, LOW);
168   delay(1000);
169 }
170
171 void alti() {
172   digitalWrite(a, HIGH);
173   digitalWrite(b, LOW);
174   digitalWrite(c, HIGH);
175   digitalWrite(d, HIGH);
176   digitalWrite(e, HIGH);
177   digitalWrite(f, HIGH);
178   digitalWrite(g, HIGH);
179   digitalWrite(dp, LOW);
180   delay(1000);
181 }
182
183 void yedi() {
184   digitalWrite(a, HIGH);
185   digitalWrite(b, HIGH);
186   digitalWrite(c, HIGH);
187   digitalWrite(d, LOW);
188   digitalWrite(e, LOW);
189   digitalWrite(f, LOW);
190   digitalWrite(g, LOW);
191   digitalWrite(dp, LOW);
192   delay(1000);
193 }
194
195 void sekiz() {
196   digitalWrite(a, HIGH);
197   digitalWrite(b, HIGH);
198   digitalWrite(c, HIGH);
199   digitalWrite(d, HIGH);
200   digitalWrite(e, HIGH);
201   digitalWrite(f, HIGH);
202   digitalWrite(g, HIGH);
203   digitalWrite(dp, LOW);
204   delay(1000);
205 }
206
207 void dokuz() {
208   digitalWrite(a, HIGH);
209   digitalWrite(b, HIGH);
210   digitalWrite(c, HIGH);
211   digitalWrite(d, HIGH);
212   digitalWrite(e, LOW);
213   digitalWrite(f, HIGH);
214   digitalWrite(g, HIGH);
215   digitalWrite(dp, LOW);
216   delay(1000);
217 }
218
219 void sifir() {
220   digitalWrite(a, HIGH);
221   digitalWrite(b, HIGH);
222   digitalWrite(c, HIGH);
223   digitalWrite(d, HIGH);
224   digitalWrite(e, HIGH);
225   digitalWrite(f, HIGH);
226   digitalWrite(g, LOW);
227   digitalWrite(dp, LOW);
228 }
229
230 void nokta() {
231   digitalWrite(a, LOW);
232   digitalWrite(b, LOW);
233   digitalWrite(c, LOW);
234   digitalWrite(d, LOW);
235   digitalWrite(e, LOW);
236   digitalWrite(f, LOW);
237   digitalWrite(g, LOW);
238   digitalWrite(dp, HIGH);
239 }
240
241 void off() {
242   digitalWrite(a, LOW);
243   digitalWrite(b, LOW);
244   digitalWrite(c, LOW);
245   digitalWrite(d, LOW);
246   digitalWrite(e, LOW);
247   digitalWrite(f, LOW);
248   digitalWrite(g, LOW);
249   digitalWrite(dp, LOW);
250 }
Done uploading.

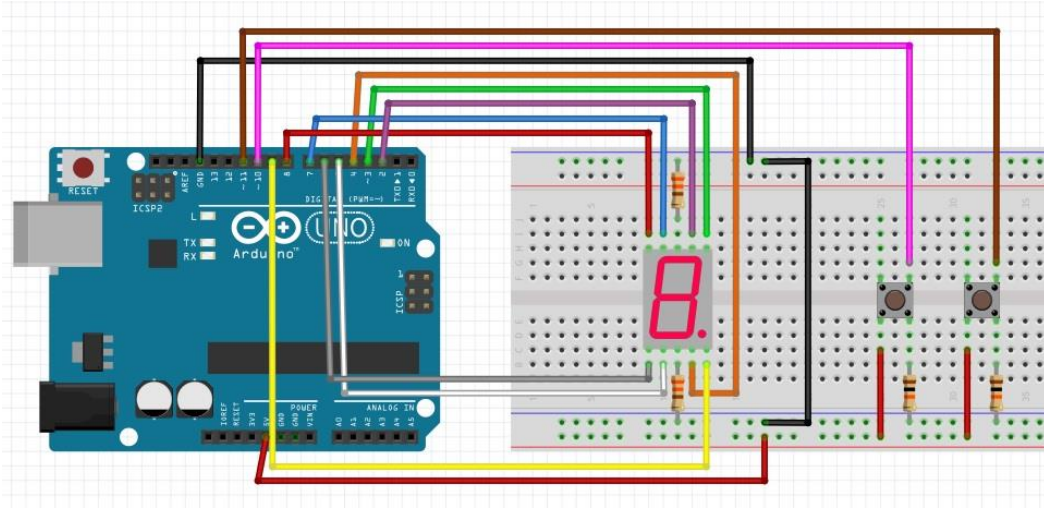
```

Resim 5.13: Gözle-Uygula Etkinliği Örnek Kodu

1.7 Gözle ve Uygula - 7 Segment Gösterge ve Buton ile Sayaç Yapımı (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
330 ohm direnç
10K ohm direnç
7 Segment gösterge
Buton

Bu etkinlikte eğitmen 7 segment göstergedeki sayıyı iki butonla azaltan ve arttıran devreyi tasarlayarak Arduino IDE’de kodlarını yazarak bu devrenin ve kodlarının mantığını öğrencilere aktardıktan sonra öğrencilerden de bu devreyi kurup kodları çalışan bir uygulama gerçekleştirmelerini isteyecektir. Öncelikle eğitmen daha önceki uygulamalarda kullanılan 7 segment gösterge devresini kuracaktır. Bu devreye daha sonra aşağıdaki resimde görüldüğü gibi iki pull-down buton devresi ekleyecektir.



Resim 5.14: Gözle-Uygula Etkinliği Örnek Devre Şeması

Bu devrede 7 segment gösterge sayaç başlangıçta 0 değerini gösterecek olup, soldaki butona her basıldığında sayaçtaki rakam 1 artacak, sağdaki butona her basıldığında ise sayaç bir azalacak şekilde Arduino IDE’de programlanacaktır. 7 segment göstergedeki değer 0 iken sağdaki butona basıldığında sayaçtaki değer tekrar 9 olacaktır. Yine 7 segment göstergedeki değer 9 olduğunda soldaki butona basıldığında ise sayaçtaki değer tekrar 0 olacaktır. Butona devamlı basılıyor mantığını kırmak için de kodun en sonuna 250 milisaniye bekleme süresi eklenecektir. Aşağıdaki resimde bu şekilde çalışan kodlar verilmiş olup, bu kodların yazımında değişken, for döngüsü, if/else döngüsü ve switch case kod yapısı kullanılmıştır.

```

7_segment_buton_sayac
1 #define a 2
2 #define b 3
3 #define c 4
4 #define d 5
5 #define e 6
6 #define f 7
7 #define g 8
8 #define dp 9
9 #define buton_arti 10
10 #define buton_eksi 11
11
12 int sayac = 0;
13
14 void setup() {
15     for(int i=2; i<10; i++){
16         pinMode(i, OUTPUT);
17     }
18     pinMode(buton_arti, INPUT);
19     pinMode(buton_eksi, INPUT);
20 }
21
22 void loop() {
23     if(digitalRead(buton_arti)==1){
24         sayac++;
25     }
26     if(digitalRead(buton_eksi)==1){
27         sayac--;
28     }
29     if(sayac > 9){
30         sayac=0;
31     }
32     if(sayac<0){
33         sayac=9;
34     }
35
36     switch(sayac){
37         case 0:
38             digitalWrite(a,HIGH);
39             digitalWrite(b,HIGH);
40             digitalWrite(c,HIGH);
41             digitalWrite(d,HIGH);
42             digitalWrite(e,HIGH);
43             digitalWrite(f,HIGH);
44             digitalWrite(g,LOW);
45             break;
46
47         case 1:
48             digitalWrite(a,LOW);
49             digitalWrite(b,HIGH);
50             digitalWrite(c,HIGH);
51             digitalWrite(d,LOW);
52             digitalWrite(e,LOW);
53             digitalWrite(f,LOW);
54             digitalWrite(g,LOW);
55             break;
56
57         case 2:
58             digitalWrite(a,HIGH);
59             digitalWrite(b,HIGH);
60             digitalWrite(c,LOW);
61             digitalWrite(d,HIGH);
62             digitalWrite(e,HIGH);
63             digitalWrite(f,LOW);
64             digitalWrite(g,HIGH);
65             break;
66
67         case 3:
68             digitalWrite(a,HIGH);
69             digitalWrite(b,HIGH);
70             digitalWrite(c,HIGH);
71             digitalWrite(d,HIGH);
72             digitalWrite(e,LOW);
73             digitalWrite(f,LOW);
74             digitalWrite(g,HIGH);
75             break;
76
77         case 4:
78             digitalWrite(a,LOW);
79             digitalWrite(b,HIGH);
80             digitalWrite(c,HIGH);
81             digitalWrite(d,LOW);
82             digitalWrite(e,LOW);
83             digitalWrite(f,HIGH);
84             digitalWrite(g,HIGH);
85             break;
86
87         case 5:
88             digitalWrite(a,HIGH);
89             digitalWrite(b,LOW);
90             digitalWrite(c,HIGH);
91             digitalWrite(d,HIGH);
92             digitalWrite(e,LOW);
93             digitalWrite(f,HIGH);
94             digitalWrite(g,HIGH);
95             break;
96
97         case 6:
98             digitalWrite(a,HIGH);
99             digitalWrite(b,LOW);
100            digitalWrite(c,HIGH);
101            digitalWrite(d,HIGH);
102            digitalWrite(e,HIGH);
103            digitalWrite(f,HIGH);
104            digitalWrite(g,HIGH);
105            break;
106
107         case 7:
108            digitalWrite(a,HIGH);
109            digitalWrite(b,HIGH);
110            digitalWrite(c,HIGH);
111            digitalWrite(d,LOW);
112            digitalWrite(e,LOW);
113            digitalWrite(f,LOW);
114            digitalWrite(g,LOW);
115            break;
116
117         case 8:
118            digitalWrite(a,HIGH);
119            digitalWrite(b,HIGH);
120            digitalWrite(c,HIGH);
121            digitalWrite(d,HIGH);
122            digitalWrite(e,HIGH);
123            digitalWrite(f,HIGH);
124            digitalWrite(g,HIGH);
125            break;
126
127         case 9:
128            digitalWrite(a,HIGH);
129            digitalWrite(b,HIGH);
130            digitalWrite(c,HIGH);
131            digitalWrite(d,HIGH);
132            digitalWrite(e,LOW);
133            digitalWrite(f,HIGH);
134            digitalWrite(g,HIGH);
135            break;
136     }
137     delay(250);
138 }
139

```

Resim 5.15: Gözle-Uygula Etkinliği Örnek Kodu

Eğitmen yukarıdaki kod yazımında öncelikle sayaç değişkeninin kullanılma mantığını öğrencilerin daha önceki kodlama bilgilerinden faydalanarak kendilerinin bulmalarını bekleyecektir. Bu kısımda öğrencilere kodları göstermeden, onların kendi çözümlerini bulmalarını sağlayacaktır. Eğer öğrenciler bu mantığı kurmakta zorlanırlarsa daha önceki uygulamalardan ipuçları vererek onlara rehberlik edebilir.

Dikkat

Arduino IDE’de setup bölümünde 7 segment göstergenin LED’lerine ait pinleri “OUPUT” olarak tanımlarken, tek tek kod yazmak yerine for döngüsünden yararlanabileceklerini ve daha az kod satırı ile aynı işlemi yapabileceklerini öğrencilere (direk söylemeden) keşfetmelerini sağlamalıdır.

Dikkat

Yukarıdaki devrede butonlara basıldığında sayaç değişkenin artırılması ve eksiltilmesi “if” karar yapısı ile kontrol edilmiştir. Değişen sayaç değerine göre 7 segment göstergede rakamların gösterilmesi ise “Switch-Case” karar yapısı ile kontrol edilmiştir. “Switch-Case” karar yapısının mantığı ve kullanımı eğitmen tarafından öğrencilere aşağıdaki anlatıma göre aktarılmalıdır.

Switch-Case karar yapısı bir değişkene göre farklı durumlar için farklı kod bloklarının çalışmasını sağlayan bir yapıdır. “Switch-Case” karar yapısı; “switch(değişken_adi)” yazıldıktan sonra değişkenin alacağı değerlere göre “case değer:” yazıldıktan sonra altındaki kod bloğunu “break;” koduna kadar çalıştıran bir yapıda kullanılmaktadır. Bu yapının değişkenin alabileceği her değer için yazılan bir case bloğu ile sürdürülmesi gerektiği öğrencilere açıklanmalıdır.

Bu uygulamanın programlaması daha az kod satırı ile yazılabilmesi için **çift düzey dizi** kullanılabilir. Geçen haftalarda dizilerin kullanımı öğrencilere gösterildiği için eğitmen öncelikle iki boyutlu dizilerle ilgili öğrencilere açıklamalarda bulunmalıdır. Daha sonra yukarıdaki kodların daha az kod satırı ile yapılabileceğini öğrencilere söyledikten sonra öğrencilerin bu görevi yerine getiren ve en az kod yazımı ile yapılması için süre vermelidir. Burada öğrencilere iki boyutlu dizilerin kullanılabilmesini hatırlatabilir. Eğer öğrenciler çözümü bulmakta zorlanırlarsa aşağıdaki resimde verilen kodları öğrencilere yazarak göstermeli ve kod mantığını açıklamalıdır.

```

7_segment_buton_arti_eksi_az_kod
1 #define a 2
2 #define b 3
3 #define c 4
4 #define d 5
5 #define e 6
6 #define f 7
7 #define g 8
8 #define dp 9
9 #define buton_arti 10
10 #define buton_eksi 11
11
12 int sayac = 0;
13
14 byte dogruluk_tablosu[10] [7]{{1,1,1,1,1,1,0}, //0
15                               {0,1,1,0,0,0,0}, //1
16                               {1,1,0,1,1,0,1}, //2
17                               {1,1,1,1,0,0,1}, //3
18                               {0,1,1,0,0,1,1}, //4
19                               {1,0,1,1,0,1,1}, //5
20                               {1,0,1,1,1,1,1}, //6
21                               {1,1,1,0,0,0,0}, //7
22                               {1,1,1,1,1,1,1}, //8
23                               {1,1,1,1,0,1,1}, //9
24                               };
25
26 void setup() {
27   for(int i=2; i<10; i++){
28     pinMode(i, OUTPUT);
29   }
30   pinMode(buton_arti, INPUT);
31   pinMode(buton_eksi, INPUT);
32 }
33
34 void loop() {
35   if(digitalRead(buton_arti)==1){
36     sayac++;
37   }
38   if(digitalRead(buton_eksi)==1){
39     sayac--;
40   }
41   if(sayac > 9){
42     sayac=0;
43   }
44   if(sayac<0){
45     sayac=9;
46   }
47
48   for(int pin=2; pin<9; pin++){
49     digitalWrite(pin, dogruluk_tablosu[sayac] [pin-2]);
50   }
51   delay(250);
52 }

```

Resim 5.16: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Yukarıdaki kod yazımını öğrenciler bulamazsa, öğretmen öğrencilere öncelikle kod yazımında dizi mantığını ve kullanımını açıklar. **Dizi** kavramının programlamada aynı türden birden fazla değişkeni tutmak için kullanıldığını anlatır. Programlamada kullanılırken ise öncelikle değişken olarak bir dizi adının tanımlanmasını ve bir veya birden fazla boyutta kaç elemandan oluştuğunu belirtir.



Eğitmen ayrıca dizinin elemanlarını çağırırken ilk elemanın 0. eleman olduğu noktasında öğrencileri uyarır.

Yukarıdaki tek boyutlu dizi kavramını öğrencilere açıkladıktan sonra uygulamanın programlamasında kullanılan iki boyutlu bir dizi olan “dogruluk_tablosu” dizisini açıklar.

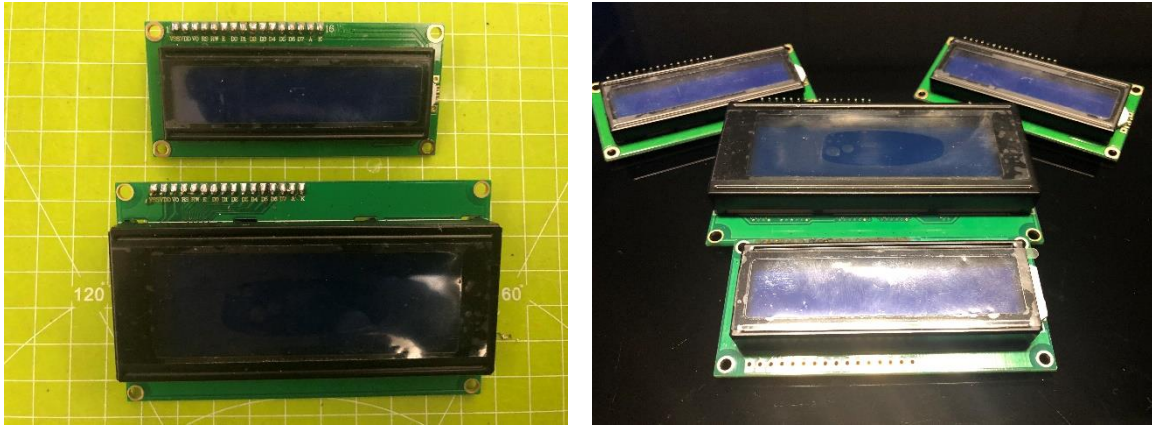
1.8 Gözle – LCD (Liquid Crystal Display) Ekranlar

LCD ekranlar günümüzde en fazla kullanılan görüntü ve veri çıktısı alınan gösterim teknolojileri olarak öne çıkmaktadır. İsminden de anlaşılacağı gibi “**LCD – Liquid Crystal**

Display – Sıvı Kristal Ekran”; hücrelerden oluşan ve hücrelerin içerisinde bulunan sıvının ışığı tek fazlı geçirerek üzerinde bulunan kutuplanma filtresi ile görünebilmesi mantığı ile çalışan görüntü teknolojileridir. Çevremizde televizyonlardan akıllı tahtalara, tabletlerden dijital saatlere kadar birçok teknolojik araçlarda kullanıldığı gibi Arduino ile gerçekleştirilen uygulama ve projelerde de en fazla kullanılan gösterim aracı olarak öne çıkmaktadır.

LCD Ekran

Arduino setlerinde genellikle kullanılan LCD tip 2 satır 16 karakterden oluşan 2x16 LCD ekrandır. Ayrıca daha fazla satır ve karakter gösterebilen 4 satır ve 20 karakterden oluşan 4x20 LCD ekran da set içerisinde bulunmaktadır. LCD ekranlar tek veya iki işlemcili LCD ekranlar olmasına rağmen; genellikle bu LCD ekranlarda HD44780 tipi tek işlemci bulunur ve LCD ekrana gelen mesajlar bu işlemci ile işlenerek kontrol edilir. LCD ekran projelerde basit manada veri ve bilgi paylaşımında kullanılsa da LCD ekranlarla ilgili daha ileri düzeyde data yazma, talimat yazma, karakter oluşturma, busy flag (ekran okuma), display data ram gibi işlemlerde de kullanılmaktadır. Aşağıdaki resimde LCD ekranlar görülmektedir.

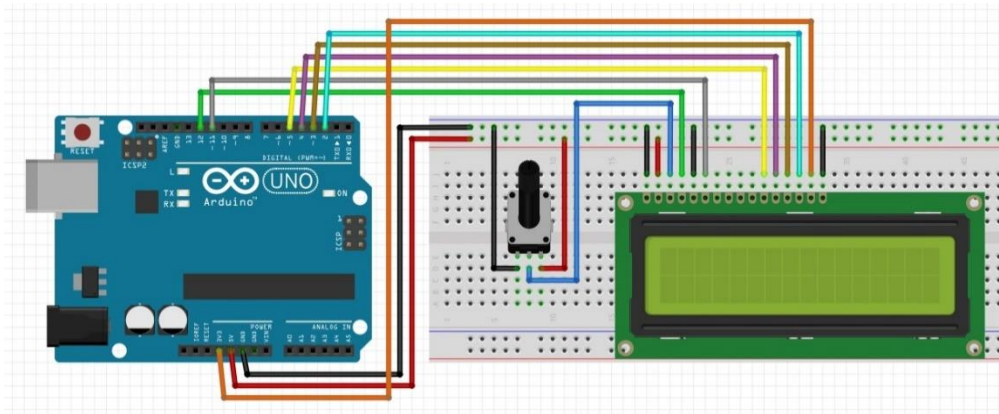


Resim 5.17: LCD Ekran

DENEYAP Atölyelerinde yürütülen Elektronik Programlama ve Nesnelerin İnterneti dersi için kullanılan setlerde farklı bir LCD ekran ve bağlantı şekli kullanılsa da; genel olarak LCD ekranların Arduino UNO kartına bağlantısı için 16 pin bulunmaktadır. Bu pinlerden 14 tanesi ekran ile Arduino UNO arasındaki bağlantıyı gerçekleştiriyorken, 15. ve 16. pinler LCD ekranın aydınlatması için “+” ve “-“ hattın bağlandığı pinlerdir. LCD ekran üzerindeki pinlerin isimleri ve görevleri aşağıdaki tabloda detaylı olarak verilmiştir.

LCD Ekran Pinleri			
Pin No	LCD Pin	Arduino Pin	Görevi
1	VSS	GND	Toprak Hattı
2	VDD	5 V	Giriş Gerilimi (Enerji)
3	V0	Potansiyometre	Kontrast Ayarı (Ekran parlaklık ayarı için)
4	RS	12. pin – Dijital Pin	Register Select – Komut (0) veya veri (1) arasında geçiş yapmak için kullanılır.
5	RW	GND	Read/Write – LCD ekrana yazma (0) / okuma (1) seçimi yapmak için kullanılır.
6	E	11. pin – Dijital Pin	Enable Pin – Register'lara yazmayı sağlayan etkinleştirme pini.

7	D0	Boş	8 bitlik veri yolu kullanılmak istendiğinde dijital pinlere bağlanır.
8	D1	Boş	8 bitlik veri yolu kullanılmak istendiğinde dijital pinlere bağlanır.
9	D2	Boş	8 bitlik veri yolu kullanılmak istendiğinde dijital pinlere bağlanır.
10	D3	Boş	8 bitlik veri yolu kullanılmak istendiğinde dijital pinlere bağlanır.
11	D4	5. pin – Dijital Pin	4 bitlik veri aktarımı
12	D5	4. pin – Dijital Pin	4 bitlik veri aktarımı
13	D6	3. pin – Dijital Pin	4 bitlik veri aktarımı
14	D7	2. pin – Dijital Pin	4 bitlik veri aktarımı
15	A	3.3 V	LCD ekran ışığı için enerji
16	K	GND	LCD ekran ışığı için toprak hattı



Resim 5.18: Gözle Etkinliği Örnek Devre Şeması

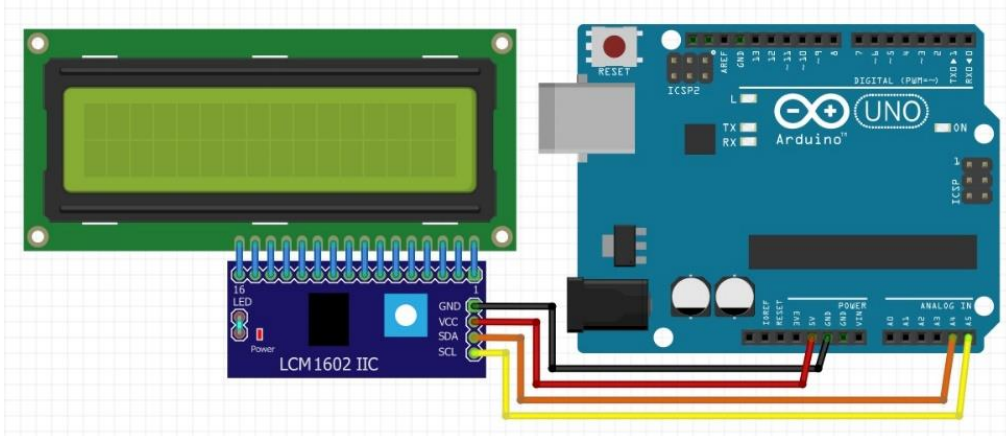
Yukarıdaki devre görseli setlerde bulunan LCD ekranların kontrast ayarını yapabilmek için devreye bir adet potansiyometre eklenir ve üstteki tabloda da belirtildiği üzere potansiyometrenin orta bacağı LCD ekranın 3. pini olan V0 pinine bağlanırken, sağ ve sol bacakların bir 5 V diğeri de GND pinine bağlanır. Ancak DENEYAP atölyelerindeki Elektronik Programlama ve Nesnelere İnterneti dersleri için kullanılan setlerde yukarıdaki bağlantı şeması verilen LCD ekranlar yerine I2C modüllü LCD ekranlar bulunmaktadır.

LCD Ekran I2C Modüllü

LCD ekranlar 16 pinden oluşan yapılarından dolayı projelerde kullanıldığında Arduino UNO kartında çok fazla pin girişini meşgul etmektedirler. 4 bitlik veri yolu kullanıldığında bile LCD ekranlar yukarıdaki görselde görüldüğü gibi Arduino UNO üzerinde 9 pine bağlanmaktadır. Projelerde LCD ekranların fazla pin kullanma problemini aşmak için LCD ekranlara **I2C modüllü** eklenerek 9 pin yerine sadece 4 pin ile LCD ekranlar Arduino UNO'ya bağlanabilmektedir. I2C modülü aslında bir seri haberleşme türlerinden sekron haberleşme prtokolüdür ve ismini **Inter-Integrated Circuit** ifadesinden alır. Bu haberleşme protokolü

genellikle kısa mesafeli ve düşük veri aktarım hızında tercih edilir, çünkü hat sayısının fazla olması nedeni ile uzun mesafeli haberleşmelerde kullanılamaz. Haberleşme için ise toprak hattı (GND) ve enerji (5 V) dışında SDA ve SCL pinlerini kullanmaktadır. Arduino UNO’da SDA bağlantısı A4 analog pini ile yapılırken, SCL bağlantısı ise A5 analog pininden sağlanmaktadır. **SDA (Serial Data Line)** çift yönlü veri aktarımının sağlandığı kanal iken, **SCL (Serial Clock)** ise kanalda aktarılan verilerin senkronizasyonu ve saat sinyaline göre veri akışının düzenlenmesi sağlar. Etkinlikle ilgili devre şeması aşağıdaki resimde görülmektedir.

I2C Modülü Pinleri			
Pin No	LCD Pin	Arduino UNO Pin	Görevi
1	GND	GND	Toprak Hattı
2	VCC	5 V	Giriş Gerilimi (Enerji)
3	SDA	A4	Çift Yönlü Veri Aktarımı
4	SCL	A5	Veri Aktarımının Senkronizasyonu



Resim 5.19: Gözle Etkinliği Örnek Devre Şeması

DENEYAP atölyelerinde Elektronik Programlama ve Nesnelerin İnterneti dersi için kullanılan setlerde üzerine I2C modülü lehimlenmiş 1 adet 4x20 LCD Ekran ile yine üzerlerine I2C modülü lehimlenmiş 2 adet 2x16 LCD ekran bulunmaktadır. Kullanım açısından, bağlantı ve kodlama açısından 4x20 LCD ekranlar ile 2x16 LCD ekranlar arasında önemli farklılıklar bulunmamaktadır. Bundan sonraki etkinliklerin çoğunda daha rahat çalışabilmek adına genellikle üzerine I2C modülü lehimlenmiş 4x20 LCD ekran kullanılacaktır.

Dikkat

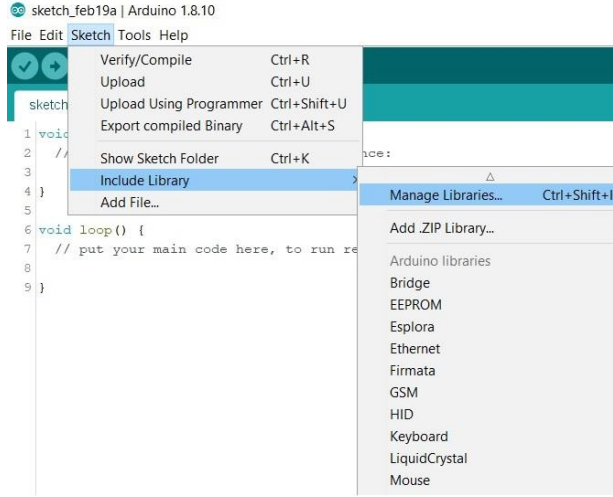
LCD ekranlar ister tek başına kullanılsın, isterse de I2C modülleri ile kullanılsın; Arduino IDE arayüzünde kodlayabilmek için bazı **kütüphanelere (library)** ihtiyaç duyulmaktadır.

1.9 Gözle – Arduino IDE’de Kütüphane (Library)

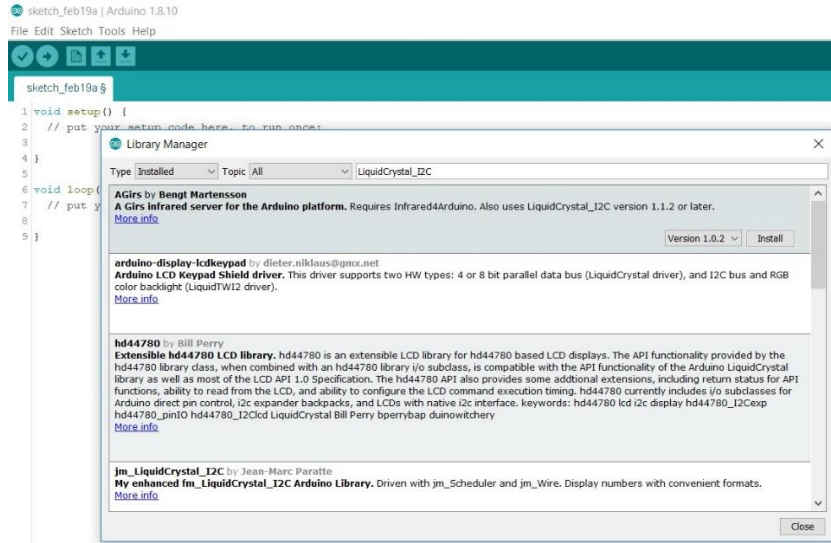
Arduino UNO kartları bazı devre elemanları ile Arduino IDE’de kodlanmak istediğinde daha kolay ve etkili bir şekilde kontrol edilebilmesi için bazı kütüphanelerden faydalanılmaktadır.

Yani, birçok programlama platformunda olduğu gibi Arduino IDE’de kodlama yapılırken kütüphaneler kullanılmaktadır. Özellikle Arduino UNO kartına takılan sensör ve diğer devre elemanlarının kendi özel kod ve fonksiyonlarını kullanarak daha az kod yazımı yapılması sağlanır. Kütüphaneler; bileşenlerle ilgili bilgiler ile birlikte çeşitli fonksiyonları içerirler ve bu sayede kod karmaşasının önlenmesini sağlarlar. Arduino IDE arayüzünde bu sensör ve devre elemanlarına ait kütüphanelerin öncelikle Arduino IDE’de bulunması ve kod bloğuna eklenmesi gerekmektedir. Bu kütüphanelerden bazıları Arduino IDE’de standart olarak bulunmakta iken; ihtiyaç duyulan diğer kütüphaneler indirilerek veya oluşturularak Arduino IDE arayüzüne eklenebilir. Arduino IDE’ye kütüphaneler iki farklı şekilde eklenebilir. Bunlar:

- ⇒ **1. Seçenek: Arduino IDE yazılımı içerisinde menüden ekleme** – Arduino IDE yazılımı açıldıktan sonra aşağıdaki resimde görüldüğü gibi “Taslak” menüsünden “Kütüphane Ekle”, “Kütüphaneleri Yönet” menülerini takip ederek “Kütüphane Yöneticisi” penceresi açılır. Bu pencerede arama kısmına istenen kütüphane ismi yazılarak arama yapılır. Listedeki istenen seçenek için “KUR” butonuna tıklanarak kütüphane Arduino IDE’nin “Kütüphane Ekle” menüsündeki kütüphaneler arasına eklenir.



Resim 5.20: Arduino IDE’ye Kütüphane Ekleme



Resim 5.21: Arduino IDE’ye Kütüphane Ekleme

- ⇒ **2. Seçenek:** Arduino klasöründeki kütüphane klasörüne ekleme - Bunun için öncelikle internette istenen kütüphane bulunur ve “zip” veya “rar” dosya uzantısı ile sıkıştırılmış bir formatta indirilir. Daha sonra bu sıkıştırılmış klasörden çıkarılır ve Arduino IDE yazılımının yüklü olduğu “Libraries” klasörünün içerisine yapıştırılır.

Dikkat

Arduino IDE’de bulunan veya sonradan eklenen kütüphaneler kod bloğuna **#include** komutu ile “<” ve “>” işaretleri arasında yazılarak dahil edilir. Aşağıda buna birkaç örnek verilmiştir.

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <DHT11.h>
```

Arduino IDE’de kod bloğuna kütüphaneler eklendikten sonra kütüphanelere dahil olan kodlar ve fonksiyonlar daha etkili kod yazımı için kullanılabilir.

1.10 Gözle ve Uygula – LCD Ekranı İlk Yazımı Yazıyorum (Öğrenci 2)

Malzeme Listesi

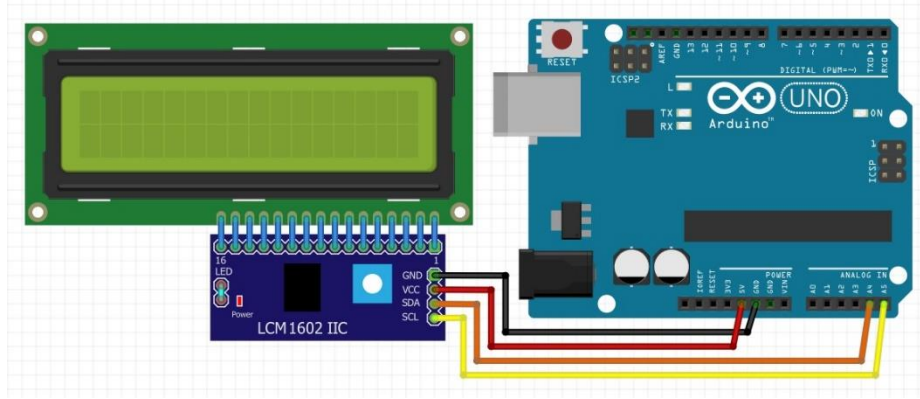
Arduino UNO

Bağlantı kabloları

2x16 LCD Ekran (I2C

Modüllü)

Bu etkinlikte I2C modülü entegreli 2x16 LCD ekranımıza ilk yazılar yazılacak olup, I2C kütüphanesi eklenerek, bu kütüphanedeki bazı komutlar kullanılacaktır. Öncelikle I2C entegreli LCD ekran Arduino UNO kartına bağlanacaktır. Arduino IDE yazılımında ise önce “Wire.h” ve “LiquidCrystal_I2C.h” kütüphanelerini kod bloğuna eklenecek, sonrasında LCD ekran LiquidCrystal_I2C kütüphanesi komutları kullanılarak bir nesne olarak tanımlanacaktır. En son olarak da “setup” kod bloğunda LCD ekranla önce haberleşme başlatılacaktır ve ekrana yazdırılmak istenen yazı (MERHABA DÜNYA) gönderilecektir. Aşağıda bu uygulamaya ait bulunan kodları eğitmen öncelikle öğrencilere kendisi açıklayarak anlatacak ve sonrasında öğrencilerin kendilerinin de LCD ekranlarına “MERHABA DUNYA” ifadesini yazmalarını istenecektir. Etkinlikle ilgili örnek devre şeması ve kod aşağıdaki resimlerde görülmektedir.



Resim 5.22: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

lcd_I2C_ekran_3 | Arduino 1.8.10
File Edit Sketch Tools Help
lcd_I2C_ekran_3
1 #include <Wire.h> //LCD için gerekli haberleşme kütüphanesini ekleniyor.
2 #include <LiquidCrystal_I2C.h> //I2C modüllü LCD ekran kütüphanesi (LiquidCrystal_I2C) ekleniyor
3
4 LiquidCrystal_I2C lcd (0x27, 16, 2); //LiquidCrystal_I2C kütüphanesinden "lcd" adında 16 sütun 2 satırlık ve 0x27 adresinde bir ekran tanımlanıyor.
5
6 void setup() {
7
8   lcd.begin(); // Tanımlanan lcd ekran bağlatılıyor.
9   lcd.print("MERHABA DUNYA"); // lcd ekranın ilk satır ve sütunundan başlatılarak "MERHABA DUNYA" yazılıyor.
10
11 }
12
13 void loop() {
14
15 }
16
Done Saving

```

Resim 5.23: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Eğer LCD ekranda yazı görünmüyorsa, bunun bir sebebi I2C modülündeki mavi parça içerisindeki ekran kontrast ayarı ile yazının görünmesi sağlanabilir.

Dikkat

LCD ve I2C kütüphanelerinde **print** fonksiyonu ile ekrana metin yazdırılırken;

- Metin çift tırnak (“ ”) arasında girilmelidir.
- Yazdırılmak istenen metinde Türkçe karakter kullanılmamalıdır.

1.11 Gözle ve Uygula – LCD Ekran için Kullanılan Komutlar (Öğrenci 1)

Malzeme Listesi

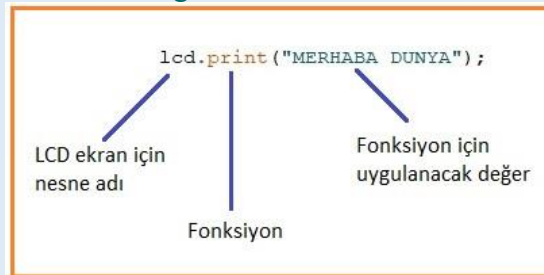
Arduino UNO
Bağlantı kabloları
4x20 LCD Ekran (I2C Modüllü)

LCD ekranlar aracılığıyla istenen veri ve mesajları dış dünya ile paylaşmada kütüphanelerden faydalandığından bahsedilmiştir. Bu uygulamada LCD ekranları kullanırken LCD ekran kütüphanelerinde bulunan ve en fazla kullanılan komut ve fonksiyonlar listelenecek ve bazıları örnek olarak kullanılacaktır. Aşağıdaki tabloda LCD ekranlarla ilgili en fazla kullanılan komut ve fonksiyonlar ve işlevleri bulunmaktadır.

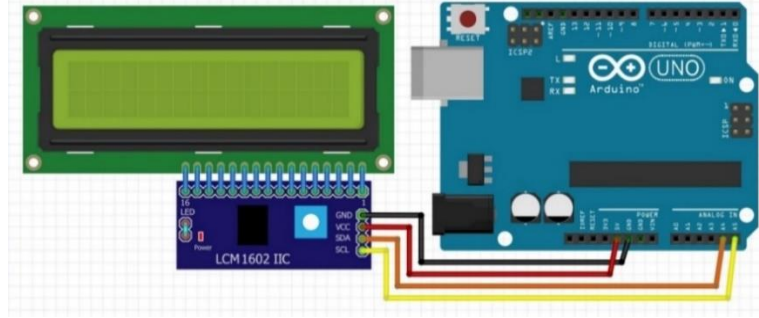
LCD Kütüphane Fonksiyonları	
Fonksiyon Adı	Görevi
begin	LCD ekran ile haberleşmeyi başlatır.
Home	İmleci en üst sol konuma (0,0) getirir.
Clear	LCD ekrandaki yazıları temizler
Print	Ekrana yazı, metin ve rakamları yazdırır.
setCursor	İmlecin konumunu belirtilen satır ve sütuna getirir.
cursor	Alt çizgili imleci açar.
noCursor	Alt çizgili imleci kapatır.
Blink	Yanıp sönen imleci açar.
noBlink	Yanıp sönen imleci kapatır.
display	LCD ekranı açar.
noDisplay	LCD ekranı kapatır.
backlight	Arka planın ışığını açar.
noBacklight	Arka planın ışığını kapatır.
scrollDisplayLeft	LCD ekranda imleci sola kaydırır.
scrollDisplayRight	LCD ekranda imleci sağa kaydırır.
createChar	Özel karakter oluşturur.
write	Özel karakter yazar.

Dikkat

Arduino IDE yazılımında LCD ekran için bu fonksiyonlar kullanılmak istendiğinde, öncelikle LCD_I2C.h kütüphanesinde oluşturulan LCD ekran nesnesinin ismi, nokta ve sonrasında komutlar yazılır. Komuttan sonra parantez içerisinde de varsa değişken veya değerler yazılır, sonuna da noktalı virgül eklenerek sonlandırılır.



Bu uygulamada ise yukarıdaki tablolardaki fonksiyonları kullanarak Arduino UNO'ya bağlı I2C modülü entegreli 4x20 LCD ekranda 1.satırda ilk sütundan başlayarak "DENEYAP", 2.satırda 4. sütundan başlayarak "ATOLYELERI", 3.satırda 8.sütundan başlayarak "LISE" ve 4.satırda 10.sütundan başlayarak "EGITIMLERI" kelimeleri yazılacaktır. Ayrıca son kelimenin sonuna yanıp sönen bir imleç eklenecektir. Etkinlikle ilgili örnek devre şeması, kod ve etkinliğin ekran çıktısı aşağıdaki resimlerde görülmektedir.



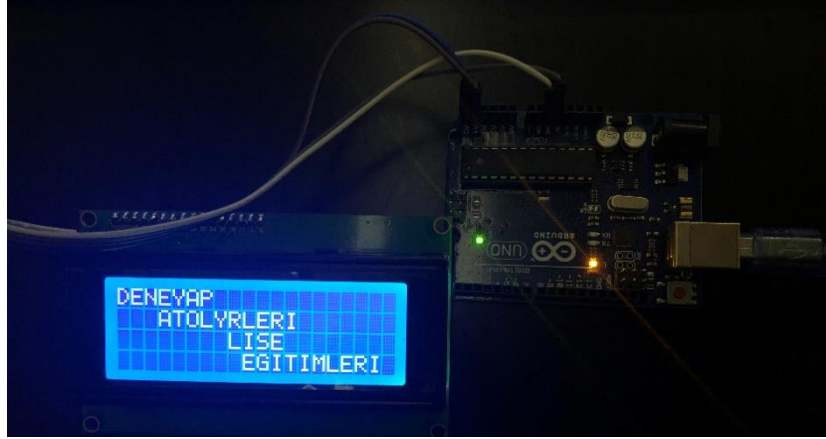
Resim 5.24: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

4_20_lcd_ekran $
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5
6 void setup() {
7   lcd.begin();
8 }
9
10 void loop() {
11   lcd.home();
12   lcd.print("DENEYAP");
13   lcd.setCursor(3,1);
14   lcd.print("ATOLYRLERI");
15   lcd.setCursor(8,2);
16   lcd.print("LISE");
17   lcd.setCursor(9,3);
18   lcd.print("EGITIMLERI");
19   lcd.blink();
20   delay(5000);
21 }
22
Done uploading.

```

Resim 5.25: Gözle-Uygula Etkinliği Örnek Kodu



Resim 5.26: Gözle-Uygula Etkinliği Ekran Çıktısı

Dikkat

Ekranında yazılması gereken metinler yazdırıldıktan sonra sonuna yanıp sönen imleç blink fonksiyonu ile eklenecektir. Bu imlecin yanıp söndüğünün görünebilmesi için de; kodun en sonuna delay fonksiyonu ile belirli bir süre (5 saniye gibi) bekleme eklenecektir.

1.12 Gözle ve Uygula – LCD Ekranda Sağa-Sola Hareketli Yazı (Öğrenci 2)

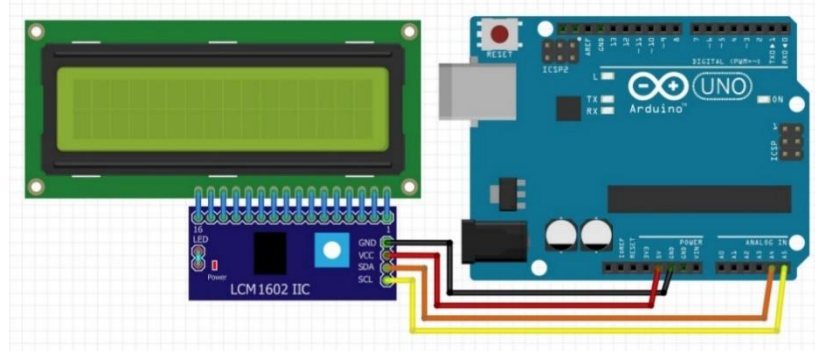
Malzeme Listesi

Arduino UNO

Bağlantı kabloları

4x20 LCD Ekran (I2C Modüllü)

LCD ekranlarda çeşitli fonksiyonlar farklı algoritma yapıları kullanılarak gösterilen metinler hareket ettirilebilir ve sağa veya sola kaydırılabilir. Bu uygulamada eğitimci bir önceki uygulamada öğrencilere bahsettiği LCD ve I2C kütüphanelerindeki fonksiyonları tanıtmaya devam edecektir. Eğitimci özellikle “**scrollDisplayRight**” ve “**scrollDisplayLeft**” komutları ile ekrandaki görselleri hareket ettirebileceklerini anlatarak, öğrencilere Arduino UNO’ya bağlanmış bir I2C modülü entegreli 4x20 LCD ekranın ikinci satırında “DENEYAP” yazısının, üçüncü satırında ise “ATOLYELERI” ifadesinin sağa ve sola hareket ederek gösterildiği bir algoritmayı Arduino IDE ile kodlayarak gösterilmesini öğrencilerden gerçekleştirmelerini ister. Burada eğitimci öğrencilere sadece kullanacakları fonksiyonların görevlerini ve nasıl kullanılacaklarını basit örneklerle gösterebilir. Ancak kesinlikle öğrencilere yukarıda istenen etkinliğin nasıl yapılacağını göstermemeli ve kodları öğrencilere vermemelidir. Bunun yerine eğitimci öğrencilerin istenen algoritmayı kurarak fonksiyonları doğru şekilde kullanmaları için rehber olarak onlara yol gösterebilir. Aşağıdaki resimde etkinlikle ilgili örnek devre şeması ve kod görülmektedir.



Resim 5.27: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

4_20_lcd_ekran_kayan_yazi $
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5
6 void setup() {
7   lcd.begin();
8 }
9
10 void loop() {
11   lcd.clear();
12   lcd.setCursor(1,1);
13   lcd.print("DENEYAP");
14   lcd.setCursor(0,2);
15   lcd.print("ATOLYELERI");
16
17   for(int i=1; i<11; i++){
18     lcd.scrollDisplayRight();
19     delay(250);
20   }
21   for(int j=1; j<11; j++){
22     lcd.scrollDisplayLeft();
23     delay(250);
24   }
25 }
26
Done uploading.

```

Resim 5.28: Gözle-Uygula Etkinliği Örnek Kodu

1.13 Gözle ve Uygula – LCD Ekranı ve Ekran Işığını Açma-Kapatma (Öğrenci 1)

Malzeme Listesi

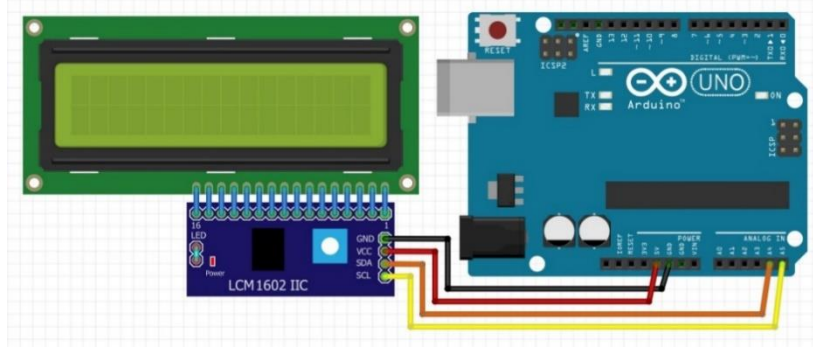
Arduino UNO

Bağlantı kabloları

4x20 LCD Ekran (I2C Modüllü)

Bu etkinlikte LCD ekran ve I2C kütüphanesinde bulunan ekranı ve ışığını açma-kapatma fonksiyonları ile birlikte imleci gösterme ve göstermeme fonksiyonlarını içeren bir kod yazıp, LCD ekran üzerinde görüntülenmesi sağlanacaktır. Öğretmen aşağıdaki kod dizimindeki

senaryoya uygun (veya kendisi bir senaryo oluşturarak) “clear, home, setCursor, blink, noBlink, backlight, noBacklight, print, display ve noDisplay” fonksiyonlarının kullanılacağı bir ekran uygulaması oluşturacaktır. Daha sonra da öğrencilerin buna benzer bir uygulamayı kendi Arduino UNO ve LCD ekranları için Arduino IDE’de gerçekleştirmelerini isteyecektir. Burada yine eğitmen öğrencilere kodu direkt vermeyecek, onların fonksiyonları kullanarak verilen senaryoyu yerine getiren bir uygulamayı yapabilmeleri için rehber olarak onlara gerektiğinde yardımcı olacaktır. Aşağıdaki resimde etkinlikle ilgili örnek devre şeması ve kod görülmektedir.



Resim 5.29: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

lcd_I2C_ekran_isik_acma_kapama
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 LiquidCrystal_I2C lcd(0x27 , 16 , 2);
5
6 void setup() {
7   lcd.begin();
8 }
9
10 void loop() {
11   lcd.noBacklight();
12   delay(2500);
13   lcd.backlight();
14   lcd.display();
15   lcd.home();
16   lcd.clear();
17   delay(1000);
18   lcd.print("Ekran aciliyor");
19   delay(500);
20   for(int i=1; i<4; i++){
21     lcd.print(".");
22     delay(500);
23   }
24   lcd.blink();
25   delay(2500);
26   lcd.noBlink();
27   lcd.clear();
28   delay(1000);
29   lcd.setCursor(7,1);
30   lcd.print("DENEYAP");
31   delay(2500);
32   for(int j=1; j<6; j++){
33     lcd.noBacklight();
34     delay(1000);
35     lcd.backlight();
36     delay(1000);
37   }
38   lcd.clear();
39   delay(1000);
40   lcd.print("Ekran kapaniyor!");
41   delay(1500);
42   lcd.noDisplay();
43   delay(1000);
44   lcd.noBacklight();
45   delay(2500);
46 }
47
Done uploading.

```

Resim 5.30: Gözle-Uygula Etkinliği Örnek Kodu

1.14 Gözle ve Uygula – LCD Ekranda Özel Karakter Oluşturma ve Gösterme (Öğrenci 2)

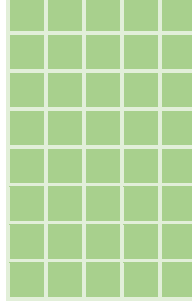
Malzeme Listesi

Arduino UNO

Bağlantı kabloları

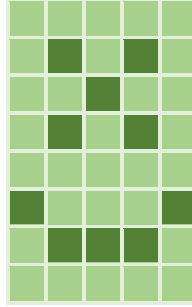
4x20 LCD Ekran (I2C Modüllü)

Aslında LCD ekranda gösterilen her harf, rakam ve diğer görseller gösterildiği alandaki 5 sütun ve 8 satırdan oluşan noktaların farklı kombinasyonları ile gösterilmektedir. Ekranda gösterilmek istenen her görsel için 5x8 pixel yapısındaki her noktanın yanıp sönmesi tek tek kodlanarak kontrol ediliyor ve istenen görsel LCD ekranda oluşuyor. Aşağıdaki resimde LCD ekran yapısı görülmektedir.



Resim 5.31: LCD Ekran Yapısı (5x8)

Yukarıdaki 5x8'lik alana bir “gülen surat” çizdirilmek istendiğinde hangi pixellerin yanacağı, hangi pixellerin sönük olacağı belirlenerek ve bunu Arduino IDE’de kodlayarak LCD ekran üzerinde görüntüleyebiliriz. Aşağıdaki resimde LCD ekranda gülen surat görülmektedir.



Resim 5.32: LCD Ekranda Gülen Surat

Yukarıdaki şekilde oluşturulmak istenen gülen yüz satır satır incelendiğinde pixellerin;

- 1.satırda: 0, 0, 0, 0, 0
- 2.satırda: 0, 1, 0, 1, 0
- 3.satırda: 0, 0, 1, 0, 0
- 4.satırda: 0, 1, 0, 1, 0
- 5.satırda: 0, 0, 0, 0, 0
- 6.satırda: 1, 0, 0, 0, 1
- 7.satırda: 0, 1, 1, 1, 0
- 8.satırda: 0, 0, 0, 0, 0

şeklinde kodlanması gerektiği anlaşılmaktadır. Bu satırların tek tek kodlanmasında daha önce de kullanılan **dizi** yöntemi kullanılacaktır. Burada kullanılan tek boyutlu dizide; oluşturulmak istenen şekle ait her satırdaki noktaların (pixellerin) koyu renk mi (1) yoksa açık renk mi (0) yanacağına yönelik kodlama yapılacaktır. Eğitimci; bu yöntemi kullanarak “gülen surat”, “ok” ve “kalp” görsellerinin oluşturulacağı ve kullanılacağı bir uygulamayı aşağıdaki şekilde öğrencilere öncelikle Arduino IDE arayüzü aracılığıyla gösterir. Sonrasında ise öğrencilerden

de bu uygulamayı yapmalarını ister. En son olarak ise öğrencilerin de kendi istedikleri üç farklı karakter oluşturmalarını ve bunları I2C modülü entegreli LCD ekrana Arduino IDE yazılımı ile gösteren bir kod yazımı yazmalarını sağlar. Burada eğitmen öğrencilerin karakterlerini oluşturabilmeleri için onlara rehber olarak yardımcı olur ama çözümü direk göstermez. Uygulamayla ilgili örnek kod aşağıdaki resimde görülmektedir.

```

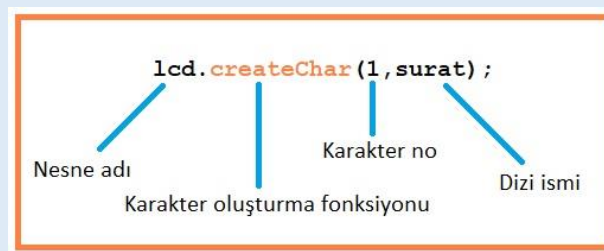
lcd_I2C_ozel_karakter §
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27, 20, 4);
4
5 byte surat[8] = {
6     B00000,
7     B01010,
8     B00100,
9     B01010,
10    B00000,
11    B10001,
12    B01110,
13    B00000,
14    };
15 byte ok[8] = {
16    B00000,
17    B00100,
18    B00010,
19    B11111,
20    B00010,
21    B00100,
22    B00000,
23    B00000,
24    };
25 byte kalp[8] = {
26    B00000,
27    B01010,
28    B11111,
29    B11111,
30    B11111,
31    B01110,
32    B00100,
33    B00000,
34    };
35
36 void setup() {
37
38     lcd.begin();
39
40     lcd.createChar(1,surat);
41     lcd.createChar(2,ok);
42     lcd.createChar(3,kalp);
43 }
44
45 void loop() {
46     lcd.clear();
47     delay(1000);
48     lcd.setCursor(4,1);
49     lcd.write(byte(1));
50     delay(1000);
51     lcd.setCursor(7,1);
52     lcd.print("DENEYAP");
53     delay(1000);
54     lcd.setCursor(16,1);
55     lcd.write(byte(1));
56     delay(2500);
57     lcd.setCursor(0,2);
58     for(int i=1; i<4; i++){
59         lcd.write(byte(2));
60         delay(500);
61     }
62     lcd.setCursor(6,2);
63     lcd.print("ATOLYELERI");
64     delay(2500);
65     lcd.setCursor(10,3);
66     lcd.write(byte(3));
67     delay(5000);
68 }
69

```

Resim 5.33: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Yukarıdaki kod satırlarında *byte* değişken tipinde “surat”, “ok” ve “kalp” adında üç dizi tanımlandıktan sonra, “**createChar**” fonksiyonu ile LCD ekranın 8 tane olan hafızasına (yani 0’dan 7’ye kadar karakter olarak) kaydedilebilir.



“createChar” fonksiyonu kullanılırken önce nesne adı ve noktadan sonra, createChar fonksiyonu yazılır, sonrasında ise parantez içerisinde karakter numarası ve virgülden sonra dizi ismi yazılır ve satır sonuna noktalı virgül konur.

Dikkat

“createChar” fonksiyonu ile oluşturulan karakterler daha sonra **write** fonksiyonu ile LCD ekran üzerine yazdırılır.



“write” fonksiyonu kullanılırken önce nesne adı ve noktadan sonra, write fonksiyonu yazılır, sonrasında ise parantez içerisinde önce tanımlanan değişkenin türü (byte) ve yine parantez içerisinde karakter numarasından sonra satır sonuna noktalı virgül konur.

1.15 Gözle ve Uygula– Arduino UNO’ya Birden Fazla I2C Modüllü LCD Ekran Bağlama ve LCD Ekranların Bağlantı Adreslerini Değiştirme (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
2x16 LCD Ekran (I2C Modüllü)
4x20 LCD Ekran (I2C Modüllü)

Arduino UNO ile projeler yaparken birden fazla LCD ekran kullanılmak istendiğinde I2C modülü entegre edilmemiş LCD ekranlarda oldukça zor olabilmektedir. Çünkü sadece bir LCD ekranın Arduino UNO’ya bağlarken Arduino UNO üzerindeki dijital pinlerden en az 6 tanesini kullanmak durumunda kalınmaktadır. Buna ek olarak diğer sensör ve devre elemanlarının da Arduino UNO kartı üzerindeki dijital pinlere bağlandığını düşündüğümüzde ciddi bir pin sayısı sıkıntısı yaşanabilir. Bunun için daha önceki Gözle ve Uygula kısımlarında I2C modülü kullanılarak sadece A4 (SDA) ve A5 (SCL) pinleri ile birlikte 5 V ve GND girişleri kullanıldığı ve belirli bir adresleme ile (0x27) LCD ekran üzerinde istenen verinin yazılabildiği anlatılmıştı. Tam da bu noktada, Arduino UNO ile gerçekleştirilen projelerde birden fazla LCD ekranın kullanılması gerektiğinde I2C modüllü entegreli LCD ekranları kullanarak ve I2C modüllerinin *adreslemeleri* ile ilgili ayarlamalar yapılarak; 8 adete kadar I2C modüllü LCD ekran Arduino UNO üzerinde başka bir pin kullanılmadan ve yine sadece A4 ve A5 pinleri ile birlikte 5 V ve GND pinleri kullanılarak Arduino UNO kartına bağlanabilir.

Normal olarak I2C modülü entegreli LCD ekranların adres bilgisi “0x27” olarak ayarlanmaktadır. Birden fazla I2C modülü entegreli LCD ekranı Arduino UNO’ya bağlarken ve Arduino IDE’de kodlarken iki noktada yapılacaktır. Öncelikle daha fazla I2C modülü entegreli LCD ekranı kullanmak istediğimizde I2C modülünde bulunan adresleme noktalarını değiştirerek I2C modülü entegreli LCD ekranın adresini “0x20” ile “0x27” arasında bir değer olarak değiştirilebilir. Bu adresleme noktaları I2C modülünün arka yüzünde “A0”, “A1” ve “A2” ifadeleri ile belirtilen ikişer noktadan oluşmaktadır. Aşağıdaki resimde I2C modülü resmi görülmektedir.



Resim 5.34: I2C Modüllü

I2C modülü üzerindeki adresleme noktalarından her çiftin kısa devre olup olmasına göre modülün adres bilgisi değişmektedir. Aşağıdaki tabloda kısa devre için “1” ifadesi kullanılırken, kısa devre olmama durumu ise “0” ile belirtilmiş olup, 8 farklı kombinasyona göre I2C modülünün alacağı adres bilgileri verilmiştir.

I2C Modülü Adres Bilgileri			
A0	A1	A2	Adres Bilgisi
0	0	0	0x27
1	0	0	0x26
0	1	0	0x25
1	1	0	0x24
0	0	1	0x23
1	0	1	0x22
0	1	1	0x21
1	1	1	0x20

Buradaki adres bilgilerine göre A0, A1 ve A2 hizasındaki çift noktalarının hangisinin kısa devre yapılacağı belirlendikten sonra havya ve lehim kullanılarak bu noktalar kısa devre haline gelmeleri için lehimlenmelidir.

Aşağıdaki görselde I2C modülü entegreli LCD ekranın üzerindeki I2C modülünün A1 ve A2 nokta çiftleri lehimlenerek I2C modülü entegreli 2x16 LCD ekranın adresi “0x21” olarak ayarlanmıştır.



Resim 5.35: I2C Modüllü LCD Ekran

Dikkat

I2C modülündeki **A0**, **A1** ve **A2** hizalarındaki çift noktalar birbirine lehimlenmesinde sadece üst üste olan noktaların birbirleri ile lehimlenmesine dikkat edilmelidir. Her çift sadece birbirleri ile lehimlenecek diğer noktalara lehimlenmeyecektir.

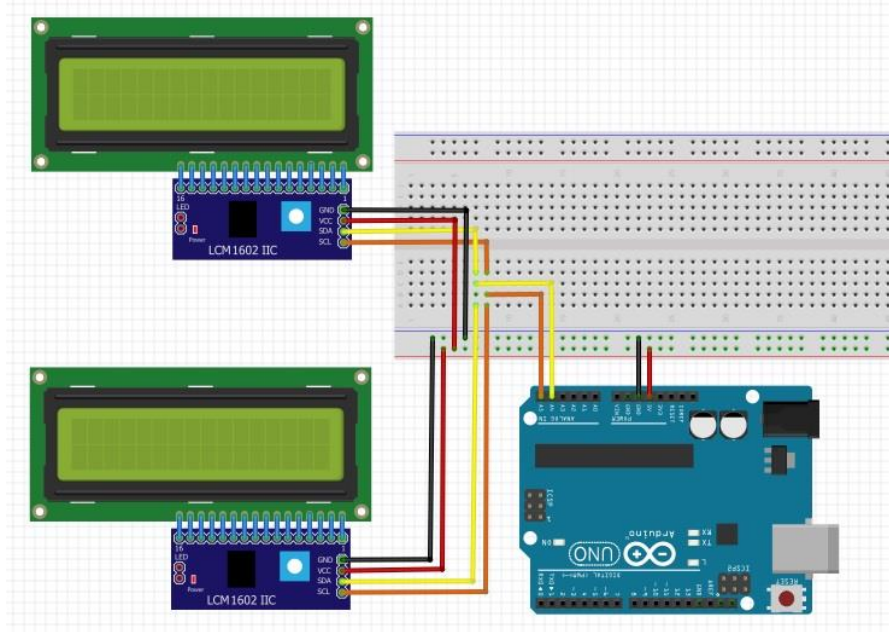
Bu noktaların lehimlenmesi çok hassas ve ince bir iş olduğu için eğitmen öğrencilere lehimleme için yardım edebilir.*

* Öğrencilerin seviyelerine ve becerilerine göre eğitmen I2C modülündeki adresleme noktalarını öğrencilerin sağlığı ve parçaların zarar görmemesi açısından dersten önce kendisi lehimleyerek I2C modülü entegreli LCD ekranları hazır hale getirebilir.

Dikkat

I2C modülü entegreli LCD ekranlardan 4x20 veya 2x16 satır ve sütunlu olması adresleme ayarları bakımından bir farklılık göstermemektedir.

I2C modülü entegreli LCD ekranların adresleme ayarları yapılarak 1 adet 2x16 I2C modülü entegreli LCD ekran ve 1 adet 4x20 I2C modülü entegreli LCD ekranı tek bir Arduino UNO kartına bağlayarak her ekranda farklı veriler yazdırabileceğimiz bir uygulamaya geçilecektir. Bu uygulamada ekranlardan birinde “*MERHABA DUNYA*” yazdırılacakken, diğer ekranda “*HELLO WORLD*” yazdırılacaktır. 4x20 I2C modülü entegreli LCD ekranın adresinde herhangi bir değişiklik yapılmadan adresi “0x27” olarak kullanılacak olup; 2x16 I2C modülü entegreli LCD ekranın A1 ve A2 adresleme çift noktaları lehimlenecek ve adresi “0x21” olarak ayarlanmış olacaktır. Kullanılacak I2C modülü entegreli LCD ekranların adresleme noktalarının lehimleme işlemi gerçekleştirildikten sonra Arduino UNO kartı üzerindeki aynı A4, A5, 5 V ve GND pinleri breadboard üzerinde çoğaltılarak LCD ekranlar I2C modülleri aracılığıyla Arduino UNO’ya bağlanır. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 5.36: Gözle-Uygula Etkinliği Örnek Devre Şeması

Devre kurulduktan sonra I2C modülü entegreli 4x20 LCD ekranda “MERHABA DUNYA”, I2C modülü entegreli 2x16 LCD ekranda ise “HELLO WORLD” yazısını yazdıran kodlar Arduino IDE’de yazılacaktır. Eğitmen bu uygulamayı kendisi yapmayıp, öğrencilerin yapmasını sağlayacaktır. Örnek kod aşağıdaki resimde görülmektedir.

```

lcd_i2c_cift_ekran §
1 #include <Wire.h> //LCD için gerekli haberleşme kütüphanesini ekleniyor.
2 #include <LiquidCrystal_I2C.h> //I2C modüllü LCD ekran kütüphanesi (LiquidCrystal_I2C) ekleniyor
3
4 LiquidCrystal_I2C lcd1 (0x27, 20, 4); //LiquidCrystal_I2C kütüphanesinden "lcd1" adında 20 sütun 4 satırlık ve 0x27 adresinde bir ekran tanımlanıyor.
5 LiquidCrystal_I2C lcd2 (0x21, 16, 2); //LiquidCrystal_I2C kütüphanesinden "lcd2" adında 16 sütun 2 satırlık ve 0x21 adresinde bir ekran tanımlanıyor.
6
7 void setup() {
8
9   lcd1.begin(); // Tanımlanan lcd1 ekran başlatılıyor.
10  lcd2.begin(); // Tanımlanan lcd2 ekran başlatılıyor.
11  lcd1.clear(); // lcd1 ekranı temizleniyor.
12  lcd2.clear(); // lcd2 ekranı temizleniyor.
13  lcd1.home(); //lcd1 ekranında imleç "0,0" satır ve sütununa getiriliyor.
14  lcd2.home(); //lcd2 ekranında imleç "0,0" satır ve sütununa getiriliyor.
15  lcd1.print("MERHABA DUNYA"); // lcd1 ekranına "MERHABA DUNYA" yazılıyor.
16  lcd2.print("HELLO WORLD"); // lcd2 ekranına "HELLO WORLD" yazılıyor.
17 }
18
19 void loop() {
20
21 }
22
Done uploading

```

Resim 5.37: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Eğitmen çift LCD ekranda yazı yazdırılması uygulamasında öğrencilere sadece LCD ekranların adres tanımlanmasında ve iki LCD ekranın Arduino IDE yazılımında farklı nesnelere olarak tanıtılmasında yardımcı olabilir. Bunun dışındaki konularda öğrencilerin kendi kodlarını yazmalarını sağlamalı, onlara takıldıkları noktalarda direk kodu vermek yerine rehber olarak yol göstermelidir.

2. ADIM: TASARLA ve ÜRET

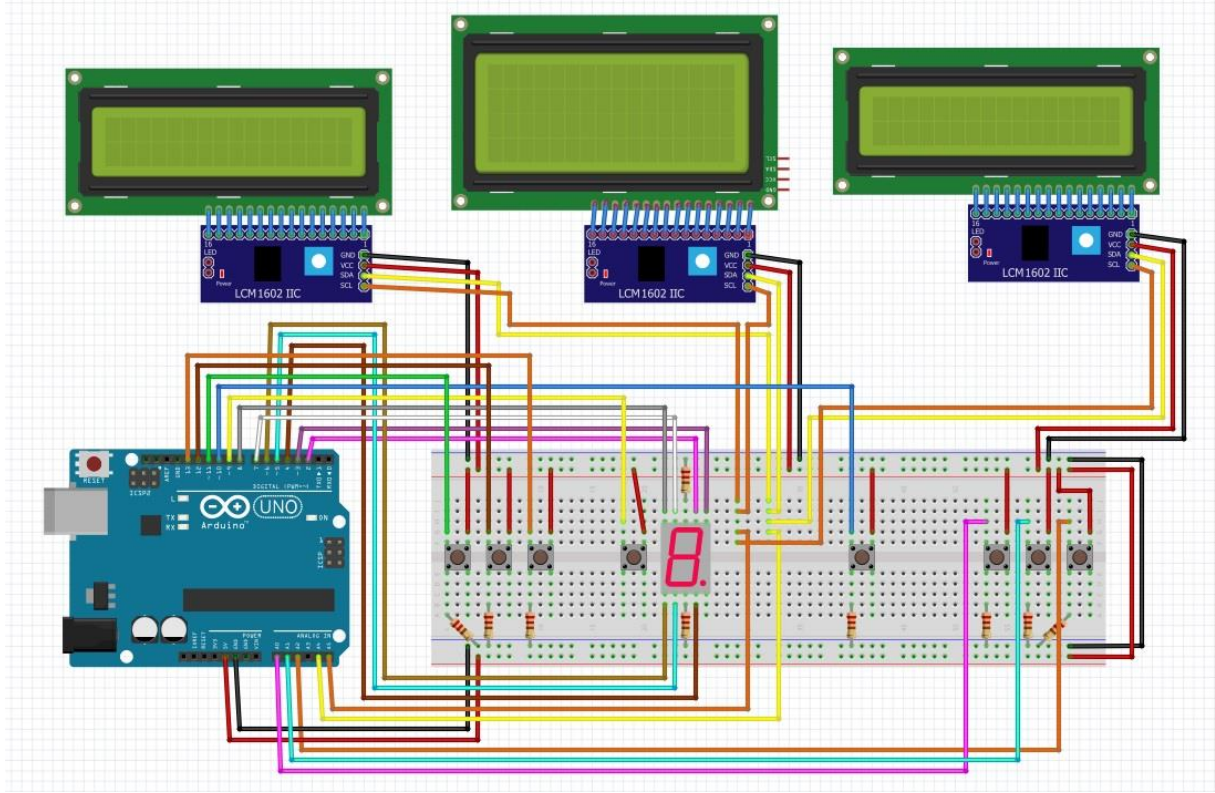
2.1 Tasarla - 3 Ekran, 1 Yedi Segment Display ve 8 Butonlu Basketbol Skorboard Yapımı

Bu etkinlikte 1 adet I2C modülü entegreli 4x20 LCD ekran ile beraber 2 adet I2C modülü entegreli 2x16 LCD ekran, 1 adet 7 segment display ve 8 butonun kullanıldığı bir basketbol skorboardu yapılacaktır. Skorboard aşağıdaki özelliklere göre tasarlanmalıdır.

- (i) I2C modülü entegreli 4x20 LCD ekran (ana ekran) basketbol maçı ile ilgili bilgileri ve süreyi göstermek için kullanılmalıdır. Ana ekranda maç süresince başka gösterim yapılmadıkça 1.satırda ortalı olarak “TBF 1.LIG MACI” yazacaktır.
- (ii) I2C modülü entegreli 2x16 LCD ekranlardan bir tanesi ev sahibi takımın adını ve skorunu göstermek için kullanılacakken, diğeri misafir takımın adını ve skorunu göstermek için kullanılacaktır.
- (iii) 7 segment display basketbol maçının kaçınıcı periyotta olduğunu göstermek için kullanılacaktır. Göstergede gösterilen rakam 1, 2, 3 ve 4 olarak basketbol maçında oynanan periodu gösterecektir.
- (iv) 8 buton ve görevleri ise aşağıdaki eşleşmeye göre tasarlanacaktır.
 - a. 1.buton; ev sahibi takımın skoruna 1 sayı ekler.
 - b. 2.buton; ev sahibi takımın skoruna 2 sayı ekler.
 - c. 3.buton; ev sahibi takımın skoruna 3 sayı ekler.
 - d. 4.buton; misafir takımın skoruna 1 sayı ekler.
 - e. 5.buton; misafir takımın skoruna 2 sayı ekler.
 - f. 6.buton; misafir takımın skoruna 3 sayı ekler.
 - g. 7.buton basketbol maçının period göstergesini bir artırır.
 - h. 8.buton tüm ekranları ve period göstergesini resetler.
- (v) Reset butonuna tıkladığında öncelikle Ana ekranda 2.satırda “Skorboard”, 3.satırda ise “sıfırlanıyor” yazdıktan 1 saniye sonra yarım saniye arayla 3 nokta yanacak ve ana ekran tekrar başlangıç haline dönerken (“TBF 1.LIG MACI” yazarken), ev sahibi takım ve misafir takımın skorları sıfırlanacak, period göstergesindeki rakam ise “1” olacaktır.
- (vi) Period butonuna tıkladığında öncelikle Ana ekranda 3.satırda ortalı olarak “PERIOD”, 3.satırda ise yine ortalı olarak “BASLIYOR” yazdıktan 1 saniye sonra yarım saniye arayla 3 nokta yanacak ve ana ekran tekrar başlangıç haline dönerken (“TBF 1.LIG MACI” yazarken), period göstergesindeki rakam 1 artacaktır.
- (vii) Sayı butonlarına basıldığında ise ana ekranda 3 satırda ortalı olarak
 - a. 1 sayı butonuna basıldı ise; “! 1 SAYI !”
 - b. 2 sayı butonuna basıldı ise; “!! 2 SAYI !!”
 - c. 3 sayı butonuna basıldı ise; “!!! 3 SAYI !!!”
 yazıları 3 defa 1 saniye görünüp, 0,5 saniye görünmeyecek ve sonrasında basıldığı takımın skoruna sayı eklenecektir. Ana ekran tekrar başlangıç haline dönecektir (“TBF 1.LIG MACI” yazacaktır).

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışırlar. Eğitimci gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin

kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir. Etkinliğin örnek şeması aşağıdaki resimde görülmektedir.



Resim 5.38: Tasarla Etkinliği Örnek Devre Şeması

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Devrenin kurulumunda hangi LCD ekranlar, gösterge, butonlar, dirençler ve ne kadar kablo kullanılmalıdır?
- Devrenin fiziksel olarak kurulumunda buton ve 7 segment gösterge için kullanılacak olan dirençlerin ohm değerleri neler olmalıdır?
- Devrenin fiziksel olarak kurulumunda I2C modülü entegreli LCD ekranların, 7 segment göstergenin ve butonların Arduino UNO pinlerine bağlantılarında nasıl bir sıra izlenmelidir?
- Özellikle I2C modülü entegreli 4x20 ve 2x16 LCD ekranların adresleme ayarları ile ilgili ne yapılması gerektiğini belirler.
- Kurulan devrenin Arduino IDE arayüzünde sorunsuz olarak çalışabilmesi için gereken kütüphaneleri belirler.
- Arduino IDE içinde yazılacak olan kodlarda hangi tür kontrol yapıları kullanılmalıdır?
- Özellikle her butonun görevini net olarak tanımlar ve nasıl bir kod yapısı kullanacağını belirler.
- Arduino IDE içerisinde yazılacak olan kodlarda her butona basılma durumunda LCD ekranlarda ve göstergede gösterilmesi gereken veriler için oluşturulması gereken algoritma nasıl olmalıdır?
- Arduino IDE içerisinde kullanılması gereken değişkenleri belirler.

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir.

- Devrenin fiziksel kurulumu için 220 ve 330 ohm'luk dirençler kullanılmalı ve Arduino UNO üzerindeki dijital pin bağlantıları ile Void fonksiyonları öncesindeki tanımlamalar bağlantılardaki pin değerleri ile örtüşmelidir.
- Devredeki ekipmanlar için kullanması gereken kütüphanelerin Arduino IDE'de bulunma durumunu kontrol eder, yok ise internetten veya Arduino IDE'nin kütüphane bölümünden indirir ve Arduino IDE'ye ekler.
- Arduino IDE içerisinde period sayısı, ev sahibi takımın skoru ve misafir takımın skoru için kullanması gereken değişkenleri belirler.
- Devredeki ekranları Arduino IDE içerisinde nesne olarak tanımlaması ile ilgili yapılara kadar verir.
- 7 segment göstergenin kodlamasında switch case yapısı ile ilgili planlama yapar.

Bu etkinlik için kullanılacak devre tasarımı yukarıda verilmiştir.

2.1 Üret- 3 Ekran, 1 Yedi Segment Display ve 8 Butonlu Basketbol Skorboard Yapımı

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında değişkenler, döngüler ve switch case yapıları ile çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.

Basketbol_Skorboard

```

1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 LiquidCrystal_I2C lcdS (0x27, 20, 4);
5 LiquidCrystal_I2C lcdH (0x21, 16, 2);
6 LiquidCrystal_I2C lcdA (0x22, 16, 2);
7
8 #define a 2
9 #define b 3
10 #define c 4
11 #define d 5
12 #define e 6
13 #define f 7
14 #define g 8
15
16 #define buton_arti 9
17 #define buton_reset 10
18
19 #define H1S 13
20 #define H2S 12
21 #define H3S 11
22
23 #define A1S A2
24 #define A2S A1
25 #define A3S A0
26
27 int period = 1;
28 int H_Skor = 0;
29 int A_Skor = 0;
30
31 void setup() {
32
33   for(int i=2; i<10; i++){
34     pinMode(i, OUTPUT);
35   }
36   pinMode(buton_arti, INPUT);
37   pinMode(buton_reset, INPUT);
38
39   pinMode(H3S, INPUT);
40   pinMode(H2S, INPUT);
41   pinMode(H1S, INPUT);
42
43   pinMode(A3S, INPUT);
44   pinMode(A2S, INPUT);
45   pinMode(A1S, INPUT);
46
47   lcdS.begin();
48   lcdH.begin();
49   lcdA.begin();
50
51   lcdS.clear();
52   lcdH.clear();
53   lcdA.clear();
54 }
55
56 void loop() {
57
58   if(digitalRead(buton_arti)==1){
59     period++;
60     lcdS.setCursor(7,2);
61     lcdS.print("PERIOD");
62     lcdS.setCursor(6,3);
63     lcdS.print("BASLIYOR");
64     delay(1000);
65     for (int j=1; j<4; j++){
66       lcdS.print(".");
67       delay(500);
68     }
69     lcdS.clear();
70     lcdS.setCursor(3,0);
71     lcdS.print("TBF 1. LIG MACI");
72   }
73
74   if(period > 4){
75     period = 1;
76   }
77
78   if(digitalRead(buton_reset)==1){
79     lcdS.setCursor(0,1);
80     lcdS.print("Skorboard");
81     lcdS.setCursor(0,2);
82     lcdS.print("Sifirlaniyor");
83     delay(1000);
84     for (int j=1; j<4; j++){
85       lcdS.print(".");
86       delay(500);
87     }
88     period = 1;
89     H_Skor = 0;
90     A_Skor = 0;
91
92     lcdS.clear();
93     lcdH.clear();
94     lcdA.clear();
95     lcdS.setCursor(3,0);
96     lcdS.print("TBF 1. LIG MACI");
97
98     lcdH.setCursor(2,0);
99     lcdH.print("ANADOLU EFES");
100    lcdH.setCursor(7,1);
101    lcdH.print(H_Skor);
102    lcdA.setCursor(3,0);
103    lcdA.print("KARSIYAKA");
104    lcdA.setCursor(6,1);
105    lcdA.print(A_Skor);
106  }
107
108  if(digitalRead(H1S)==1){
109    for (int j=1; j<4; j++){
110      lcdS.setCursor(5,2);
111      lcdS.print("! 1 SAYI !");
112      delay(1000);
113      lcdS.setCursor(0,2);
114      lcdS.print("
115      delay(500);
116    }
117    H_Skor = H_Skor + 1;
118  }
119

```

```

120 if(digitalRead(H2S)==1){
121     for (int j=1; j<4; j++){
122         lcdS.setCursor(4,2);
123         lcdS.print("!! 2 SAYI !!");
124         delay(1000);
125         lcdS.setCursor(0,2);
126         lcdS.print("                ");
127         delay(500);
128     }
129     H_Skor = H_Skor + 2;
130 }
131
132 if(digitalRead(H3S)==1){
133     for (int j=1; j<4; j++){
134         lcdS.setCursor(3,2);
135         lcdS.print("!!! 3 SAYI !!!");
136         delay(1000);
137         lcdS.setCursor(0,2);
138         lcdS.print("                ");
139         delay(500);
140     }
141     H_Skor = H_Skor + 3;
142 }
143
144 if(digitalRead(A1S)==1){
145     for (int j=1; j<4; j++){
146         lcdS.setCursor(5,2);
147         lcdS.print("! 1 SAYI !");
148         delay(1000);
149         lcdS.setCursor(0,2);
150         lcdS.print("                ");
151         delay(500);
152     }
153     A_Skor = A_Skor + 1;
154 }
155
156 if(digitalRead(A2S)==1){
157     for (int j=1; j<4; j++){
158         lcdS.setCursor(4,2);
159         lcdS.print("!! 2 SAYI !!");
160         delay(1000);
161         lcdS.setCursor(0,2);
162         lcdS.print("                ");
163         delay(500);
164     }
165     A_Skor = A_Skor + 2;
166 }
167
168 if(digitalRead(A3S)==1){
169     for (int j=1; j<4; j++){
170         lcdS.setCursor(3,2);
171         lcdS.print("!!! 3 SAYI !!!");
172         delay(1000);
173         lcdS.setCursor(0,2);
174         lcdS.print("                ");
175         delay(500);
176     }
177     A_Skor = A_Skor + 3;
178 }
179
180 lcdS.setCursor(3,0);
181 lcdS.print("TBF 1. LIG MACI");
182
183 lcdH.setCursor(2,0);
184 lcdH.print("ANADOLU EFES");
185 lcdH.setCursor(7,1);
186 lcdH.print(H_Skor);
187 lcdA.setCursor(3,0);
188 lcdA.print("KARSIYAKA");
189 lcdA.setCursor(6,1);
190 lcdA.print(A_Skor);
191
192 switch(period){
193     case 1:
194         digitalWrite(a,LOW);
195         digitalWrite(b,HIGH);
196         digitalWrite(c,HIGH);
197         digitalWrite(d,LOW);
198         digitalWrite(e,LOW);
199         digitalWrite(f,LOW);
200         digitalWrite(g,LOW);
201         break;
202
203     case 2:
204         digitalWrite(a,HIGH);
205         digitalWrite(b,HIGH);
206         digitalWrite(c,LOW);
207         digitalWrite(d,HIGH);
208         digitalWrite(e,HIGH);
209         digitalWrite(f,LOW);
210         digitalWrite(g,HIGH);
211         break;
212
213     case 3:
214         digitalWrite(a,HIGH);
215         digitalWrite(b,HIGH);
216         digitalWrite(c,HIGH);
217         digitalWrite(d,HIGH);
218         digitalWrite(e,LOW);
219         digitalWrite(f,LOW);
220         digitalWrite(g,HIGH);
221         break;
222
223     case 4:
224         digitalWrite(a,LOW);
225         digitalWrite(b,HIGH);
226         digitalWrite(c,HIGH);
227         digitalWrite(d,LOW);
228         digitalWrite(e,LOW);
229         digitalWrite(f,HIGH);
230         digitalWrite(g,HIGH);
231         break;
232 }
233 delay(250);
234 }

```

Done Saving.

Resim 5.39: Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün tasarlayıp geliştirdikleri basketbol skorboard etkinliği için sınıf içerisinde öğrenciler seçilir ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

- 7 segment göstergenin bağlantı şeklini ve kontrolünü açıklayınız.
 - 7 segment göstergenin ortak anot ve ortak katot olması bağlantısını nasıl etkiler?
- I2C modülü LCD ekranlarda neden kullanılmaktadır?
- I2C modülü entegreli LCD ekranların arka kısmında lehimleme neden yapıldı?
- Devrelerin fiziki kurulumunda en fazla nerelerde zorlandınız?
- Arduino IDE yazılımında “kütüphane” kullanımının sebepleri ve avantajları nelerdir?
- Program algoritmalarını oluştururken hangi süreçlerde zorlandınız?
- Arduino IDE içerisinde yazılan kodlarda tek boyutlu ve çift boyutlu dizi kavramının kullanılmasının avantajları nelerdir?
- Özellikle buton ile 7 segment göstergede arttırma ve eksiltme işleminin yazımında dizi kavramının kullanılmasının sebepleri ve avantajları nelerdir?
- Arduino IDE içerisinde yazılan kodlarda switch-case yapısının kullanılmasının avantajları nelerdir?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşmaya kadar devam ettirilebilir.

6.Bölüm - DC, Servo ve Step Motorların Kullanımı

Ön bilgi:

- Öğrenciler algoritma mantığını bilir.
- Öğrenciler temel programlama kavramlarını bilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.
- Öğrenciler Arduino UNO'yu tanır ve bunu kullanarak devreleri kurup kodlarında düzenleme yapabilirler.
- Öğrenciler Arduino IDE arayüzü ile değişkenler, "if/else, for" döngüsü, dizi vb. yöntemleri ile kod yazarak elektronik devre elamanlarını yönetebilir.

Bölüm Kazanımları:

- Öğrenciler DC motorların çalışma prensiplerini bilir.
- Arduino IDE üzerinde DC motorları çalıştırabilecek ve kontrol edebilecek programı yazabilir.
- Öğrenciler servo motorların çalışma prensiplerini bilir.
- Öğrenciler Arduino IDE üzerinde servo motorları çalıştırabilecek programı yazabilir.
- Öğrenciler servo motorlar ile ultrasonik sensör, HC-SR04 sensör tutucu, Buzzer ve LCD ekran gibi devre elemanlarının yer aldığı devreleri kurabilir ve program kodlarını yazabilir.
- Öğrenciler step motorların çalışma prensiplerini bilir.
- Öğrenciler Arduino IDE üzerinde step motorları kontrol edebilecek programı yazabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle DC, servo ve step motorların çalışma prensipleri hakkında bilgi sahibi olmalarıdır. Bu motorların ve sürücü devrelerinin tasarımı ve bağlantıları konusunda bilgi verilecektir. Motorların farklı devre elemanlarıyla bir arada kullanımı ve uygun devre tasarım süreçleri hakkında uygulamalar yapılacaktır. Ayrıca kurulan bu devreleri Arduino IDE ile kodlanması ve buna uyumlu farklı devre tasarımları üzerinde çalışmalar gerçekleştirilecektir.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	DC motor
Breadboard	DC Motor sürücü kartı
Bağlantı kabloları	Step Motor
Pervane	Step Motor sürücü kartı
Buzzer	Servo motor
Buton	BC547 transistör
Diyot	9 V pil
Potansiyometre	Ultrasonik sensör ve tutucu
Joystick	Direnç

Haftanın İşlenişi:

Gözle: DC motor, servo motor ve step motorun çalışma prensibinin verilmesi ve ilgili devrelerin kurulması.

Uygula: DC motor, servo motor ve step motor ile örnek devrelerin oluşturulması ve programlanması.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

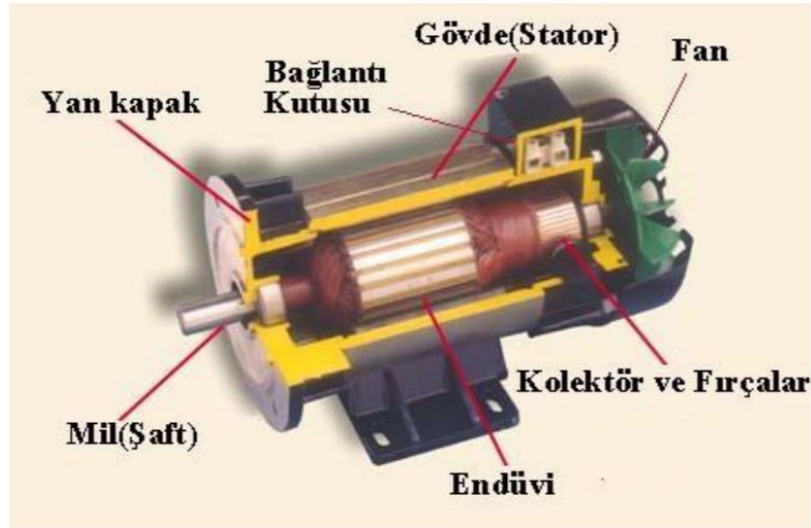
1. ADIM: GÖZLE ve UYGULA

1.1 Gözle – DC Motor Nedir?

DC motorlar doğru akımla çalışan ve elektrik enerjisini mekanik enerjiye dönüştüren araçlardır. Genellikle devrelerimizde hareket vermek istediğimiz mekanizmalara bağlanırlar. Motor türleri üzerine düşünürken motorların yapısal farklılıklarından ziyade hangi alanda, ne amaçla kullanılabileceği üzerine düşünmek ve planlama yapmak gerekmektedir. Öncelikle bu ders kapsamında sıklıkla duyacağınız bazı DC motor türleri aşağıda verilmiştir.

- Fırçalı DC motorlar
 - Redüktörlü DC motorlar
 - Redüktörsüz DC motorlar
- Fırçasız DC motorlar
- Servo Motorlar
- Step Motorlar

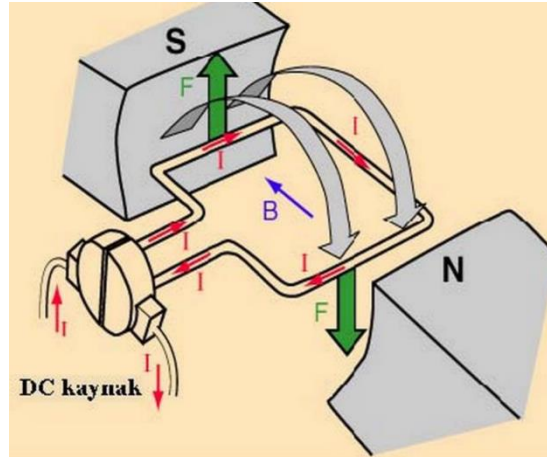
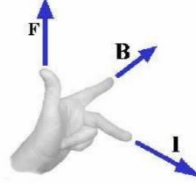
Fırçalı DC motor: Bu motor türünde manyetik alan motorun miline hareket vermek için kullanılır. Rotor (endüvi) üzerindeki sarımlara akım verilerek, gövde üzerine yerleştirilmiş sabit mıknatısları itip çekmesi prensibinden yola çıkarak hareket elde edilir. Burada motorun hareketli parçası olan rotor üzerindeki sarımlara gelecek akım kolektör ve fırça denen bir mekanizma ile kontrol edilir. Fırçalar rotorda sarılı bobinlere sırasıyla elektrik vererek gövdedeki sabit mıknatıslarla arasında oluşan manyetik alandan faydalanırlar. Fırçalı motorlar ekstra herhangi bir sürücü devresine ihtiyaç duymadıkları için tercih edilirler. Fakat fırçaların aşınmasıyla birlikte özelliklerini yitirecekleri için belirli aralıklarla değiştirilmeleri gerekmektedir. Aşağıdaki resimde fırçalı motor örneği görülmektedir.



Resim 6.1: Fırçalı Motor Örnek Göreseli

İletkenden geçen akım yön değiştirirse itilme yönü de değişir. İtilme yönünün değişmesi motorun dönüş yönünü de değiştirir. Bunu daha iyi anlayabilmek için sol el kuralından bahsetmekte fayda vardır. Bu kurala göre işaret parmağı yönünde bir manyetik alanın (B) olduğu ortamda, bu alan üzerine orta parmak yönünde bir akım (I) geçen bir iletken dahil

edilirse, iletkene baş parmak yönünde bir kuvvet (F) etkilediği görülür. Bu kurala göre aşağıdaki resimde görüldüğü gibi akımın yönüne bağlı olarak motorun hareket yönü de değişecektir.

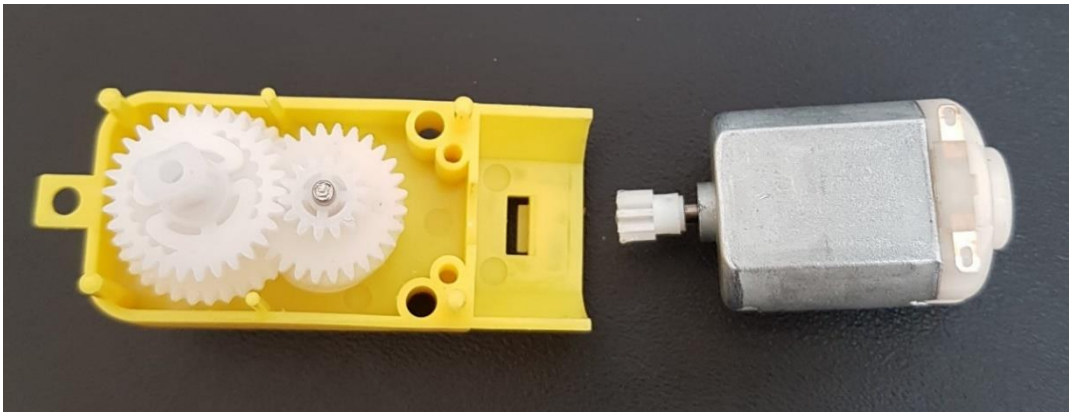


Resim 6.2: DC Motorda Akım, Manyetik Alan ve Dönüş Yönü İlişkisi

Fırçalı DC motorlar, “redüktörlü” ve “redüktörsüz” olmak üzere ikiye ayrılır.

Redüktörlü DC Motorlar:

Motorun yüksek devirde çalışmasından ziyade, torka (dönme momenti) ihtiyaç duyduğu durumlarda kullanılır. Miller ve dişliler aracılığıyla torku daha yüksek dönüş elde etmek amacıyla kullanılan bir mekanizmadır. Aşağıdaki resimde redüktörlü bir motorun iç yapısı görülmektedir. Bu mekanizma sayesinde motor hızı ve yönü motor gövdesi içerisinde ayarlanır ve torku (dönme kuvveti) daha yüksek motorlar elde edilmiş olur.



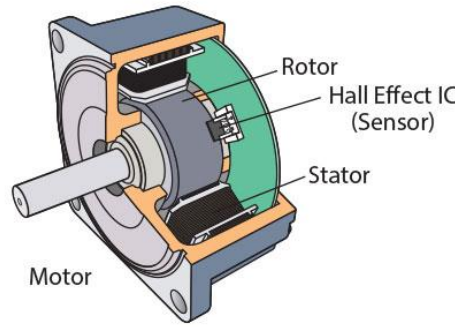
Resim 6.3: Redüktörlü DC Motor İç Yapısı

Redüktörsüz DC Motorlar:

Bu motorlarda DC motorun hızı daha ön plandadır ve motorun milinin yüksek devirde dönmesi istenen durumlarda redüktörsüz DC motorlar tercih edilir.

Fırçasız DC Motorlar:

Fırçalı DC motorlardan temel farkı bu motorlarda bobinler sabittir ve statora bağlıdır. Mıknatıslar ise rotora bağlıdır. Bu motorda rotor ve stator yer değiştirmiştir, böylece fırça ve kollektör sisteminin kalkması bu ikiliden kaynaklı elektrik/mekanik kayıpları ortadan kaldırmaktadır. Fakat konum belirleyici bir sensör ile rotor konumuna bağlı olarak statordaki uygun bobinlere akım yollanarak yüklenir ve hareket sağlanır. Bu motor yüksek hız ve tork sağlaması ile avantajlı olsa da harici güç kaynağı gerektirmesi ve maliyet nedeniyle kullanım alanları farklılaşmaktadır. Aşağıdaki resimde fırçasız DC motorun iç yapısı görülmektedir.



Resim 6.4: Fırçasız DC Motor İç Yapısı

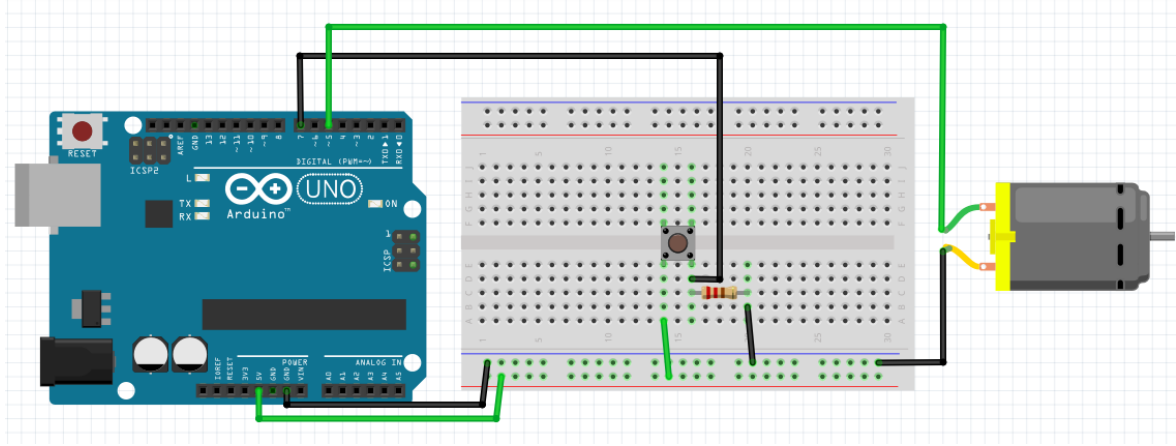
1.2 Gözle-Uygula: Butona basınca çalışan vantilatör (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
DC motor
Pervane
Buton
10 K Ohm Direnç

Burada amaç butona basınca çalışmaya başlayan, tekrar basıldığında duran bir vantilatör tasarlamaktır. Bunu yapabilmek için eğitmen öncelikle ihtiyaç duyulacak devre elemanlarını hazırlar. Buton daha önceki haftalardaki etkinliklerde zaten kullanılmıştı ama yine de eğitmen kısa bir hatırlatma yaparak butonların devreye nasıl bağlanması gerektiğini gösterir. Bu devrede butona basılıp basılmadığı 7 nolu pinden kontrol edilecektir. Miline pervane bağlanan motor ise 5 nolu pinden kontrol edilecektir. Eğitmen bu açıklamaları yaptıktan sonra, devre kurulumunu önce kendisi yapar ve sonrasında öğrencilere yardımcı olur. Devre kurulumu tamamlandıktan sonra öğrencilerle beraber aşağıdaki kodu yazar, açıklar ve öğrencilere yazım sürecinde destek olur.

Bu devre kurulduktan ve kod yazım süreci tamamlandıktan sonra devrede motora gelen bağlantı uçlarının yönü değiştirilir. Yani GND ucu ve 5 nolu pine giden uçlar yer değiştirilir. Böylece motorun dönüş yönü motorun girişi değiştirilerek kontrol edilmiş olur. Eğitmen bu konuda

öğrencileri bilgilendirdikten sonra öğrencilerin uygulamaları konusunda yardımcı olur. Gerekli açıklamaları yapar. Etkinlikle ilgili örnek devre şeması ve kod aşağıdaki resimde görülmektedir.



Resim 6.5: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

Kontrol Et
dc1
1 #define MOTOR 5
2 #define BUTTON 7
3 int deger = 0;
4 int eski_deger = 0;
5 int state = 0;
6
7 void setup() {
8   pinMode(MOTOR, OUTPUT);
9   pinMode(BUTTON, INPUT);
10 }
11 void loop() {
12   deger = digitalRead(BUTTON);
13   if ((deger == HIGH) && (eski_deger == LOW)){
14     state = 1 - state;
15   }
16   eski_deger = deger;
17   if (state == 1) {
18     analogWrite(MOTOR, 255);
19   }
20   else {
21     analogWrite(MOTOR, 0);
22   }
23 }

```

Resim 6.6: Gözle-Uygula Etkinliği Örnek Kodu

1.3 Gözle ve Uygula - Transistör ile DC Motor Sürme (Öğrenci 2)

Malzeme Listesi

Arduino UNO

Breadboard

Bağlantı kabloları

9 V DC motor

BC547 transistör

10 K Ohm Direnç

Diyot

9 V pil

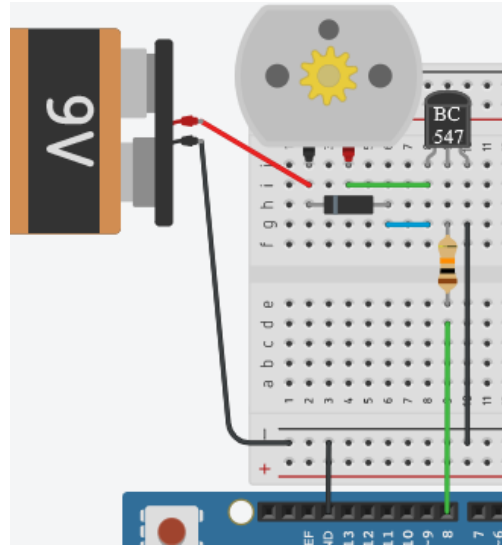
Bu etkinlikteki amaç BC547 transistörünü kullanarak öğrencilere DC motor sürmeyi göstermektir. Motor sürücülerine neden ihtiyaç duyulduğuna yönelik bilgi vererek bu etkinliğe başlamakta fayda olabilir.

Yukarıda anlatıldığı gibi aslında basit anlamda DC motorlar uçlarına uygun miktarda gerilim uygulandığında dönerler fakat bazı durumlarda DC motorlar 40 mA'den daha fazla akım çeker. Buna karşın Arduino UNO kartının I/O pinlerinden maksimum 40 mA akım çekilebilir. Bu da Arduino UNO'nun limitlerini aştığından ona zarar verebilir. DC motor gibi fazla akım çeken devre elamanlarını Arduino kart ile programlamak ve kontrol etmek için transistörler anahtar olarak kullanılabilir. Aşağıdaki devre böyle bir uygulamaya örnektir.

Bu devre şu şekilde çalışır:

Bu devrede motorun akımı 9 V pil üzerinden sağlanmaktadır. Pilin '+' ucu motora, motorun diğer ucu ise BC547'nin "kollektör" bacağına bağlanmıştır. BC547'nin "baz" bacağı bir direnç üzerinden Arduino UNO'nun 8 numaralı dijital pinine bağlanmıştır. 9 V pilin '-' bacağı ile Arduino UNO'nun GND bacağı birleştirilerek BC547'nin emiter bacağına bağlanmıştır. Bu durumdayken 8 numaralı pinden açık (ON) değeri gönderildiğinde BC547 anahtar görevi görür ve 9 V 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.

Bu devre tasarımı öğrencilere eğitmen tarafından açıklandıktan sonra devredeki DC motorun 1 sn hareket etmesi ve 1 sn durması istenmektedir. Buna yönelik olarak devre önce eğitmen tarafından kurulur ve ilgili kod yazılır. Ardından eğitmen, öğrencilerin devreyi kurum ve kod yazım süreçlerine destek olur. Etkinlikle ilgili örnek devre şeması ve kod aşağıdaki resimlerde görülmektedir.



Resim 6.7: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

1 #define motor 8
2 void setup() {
3   pinMode(motor, OUTPUT);
4 }
5 void loop() {
6   digitalWrite(motor, LOW);
7   delay(1000);
8   digitalWrite(motor, HIGH);
9   delay(1000);
10 }

```

Resim 6.8: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Burada kullanılacak olan direnç değeri önemlidir. Bu direnç değeri BC547'nin hangi çeşidinin kullanıldığına bağlı olarak değişim gösterir. BC547 üzerinde BC547A yazıyorsa 10 K Ohm, BC547B yazıyorsa 20K Ohm direnç, BC547C yazıyorsa 30K Ohm direnç kullanılmalıdır. Aksi halde BC547 yanabilir. Bu konuda öğrenciler uyarılmalıdır ve eğitmen öğrencilerin doğru direnci kullanmalarını sağlamalıdır. Setler içerisinde 20K Ohm ve 30K Ohm direnç yoktur. 10K Ohm dirençler seri bağlanarak 20K Ohm veya 30K Ohm direnç elde edilebilir. Bu bağlantılar aşağıdaki resimde olduğu gibidir.

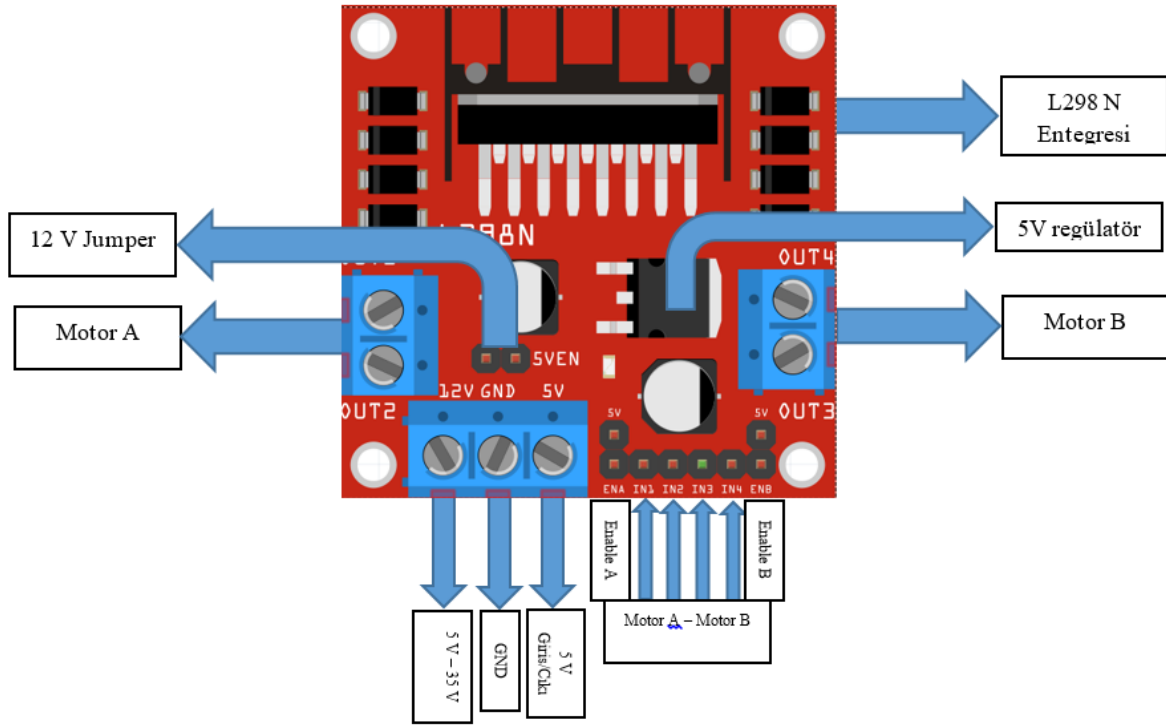


Eğitmen devreyi kurup çalışmasını öğrencilere anlattıktan sonra aşağıdaki kodu yazar ve öğrencilere göstererek kodu çalıştırır. Öğrenciler devrenin çalışmasını gözledikten sonra kendileri devreyi kurar ve kodu yazarak çalıştırır.

1.4 Gözle– DC Motor Sürücülerini Kullanma

Sürücü devreleri ile DC motorların hızları ve dönüş yönleri denetlenebilir. Fırçalı DC motorların en büyük avantajı, motora uygulanan gerilimin büyüklüğü ve yönü değiştirilerek çok kolay bir biçimde hız ve yön kontrolünün yapılabilmesidir. Daha önce de anlatıldığı gibi gerilimi değiştirilerek devri ayarlamak mümkündür. Endüviye uygulanan akımın yönü değiştirilerek de dönüş yönünün değiştirilmesi sağlanmaktadır. İşte motor sürücü devreleri de motora giden sinyallerde bu ayarlamaları yaparak motorun hız ve yönünü kontrol etmeye yardımcı olmaktadır.

Motor sürücülerinin motorların kontrolünde kullanılmasının temel nedeni, az önce de açıklandığı gibi Arduino UNO'nun çıkış pinlerinden maksimum 40 mA'e kadar çıkış elde edilebilmesidir. Arduino üzerinden doğrudan motor sürmek istersek kullanacağımız motorlar çok daha fazla akım çekeceğinden Arduino UNO'ya zarar verebiliriz. Motorlar; 12 V ve yüksek akım çektikleri için motorların Arduino üzerinden kontrolü problem yaratabilir. Bu durumu engellemek ve motorları kontrol etmek için motor sürücü devreleri kullanılmaktadır. Çok farklı amaç ve kapasitelerde motor sürücü devreleri olmasına karşın bu ders kapsamında **L298N sürücü modülü** kullanılacaktır. Aşağıdaki resimde “L298N DC motor sürücüsü” görülmektedir.



Resim 6.9: L298N DC Motor Sürücü

L298N sürücü modülünün en önemli bileşeni L298N entegresidir. Bu entegre 46 V'a kadar gerilimde çalışabilmekte ve 4 A'e kadar da akım verebilmektedir. Böylece kanal başına (motor A ve Motor B'ye) 2 A akım verebilir. Bu aralıktaki farklı motorlar L298N sürücüsüyle kontrol edilebilir.

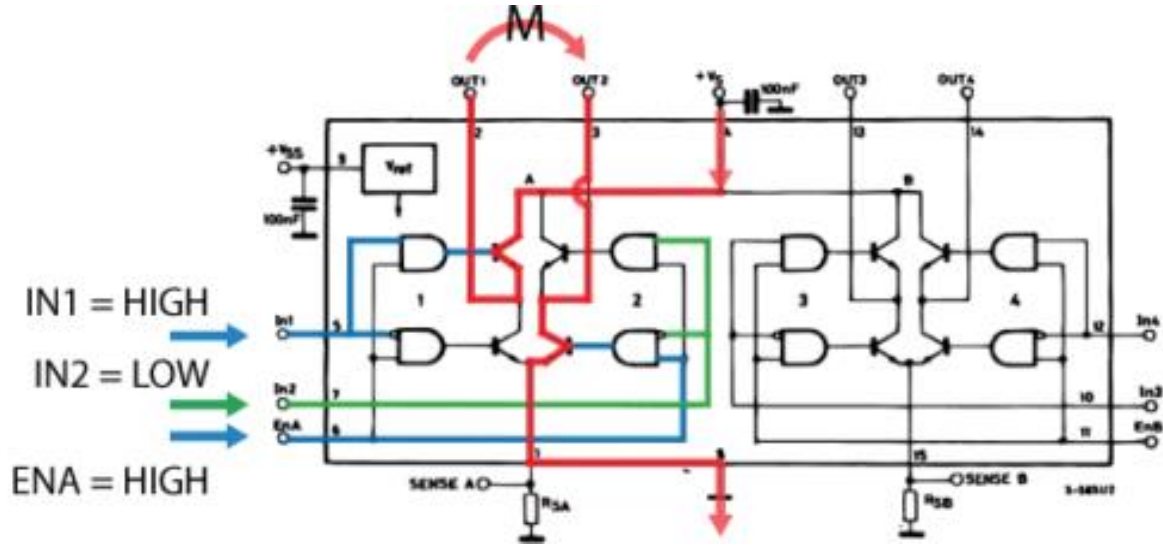
Ayrıca motor sürücüsünün bir diğer işlevi de ters akım korumasıdır. Motora gerilim verildiğinde motorda bir akım indüklenecektir. Motor durduğunda da motorda indüklenen bu akım bir yere gitmek isteyecektir. Bu da motor sürücü modülü üzerinden olacaktır. Motordan gelen bu ters akımı durdurmak için L298N modülü üzerinde diyotlarla koruma sağlanmıştır.

L298N motor sürücüsü üzerinde bulunan pinler ve açıklamaları:

- Sürücü modülü 12 V ile beslenir. Ayrıca 5 V çıkış da verebilir.
- Sürücü modülünün sağ ve sol kenarlarında yer alan portlar sürücü üzerinden kontrol edilecek motorların bağlanacağı çıkışlardır.
- IN1, IN2, IN3 ve IN4 pinleri arasındaki pinler Arduino UNO'ya bağlanacak olan giriş pinleridir. Bu pinlere gelen sinyali kontrol ederek sürücüye bağlı olan motorların dönüş yönü kontrol edilebilir.
- ENA ve ENB pinlerine ise PWM sinyali uygulayarak, motorların hızı kontrol edilebilir. Bu pinler kullanılmak istenirse üzerindeki jumperlar çıkarılmalıdır. Jumperlar çıkarılmazsa motorlar sabit bir hızda dönerler.

Bu sürücü üzerinde yer alan L298N entegresinde aslında temel olarak 2 tane H köprüsü vardır. H köprüsü 4 transistör ve kapılardan oluşan bir devredir. Bu şekilde sürücünün iki tarafına bağlanmış motorlar bu entegre içinde yer alan iki adet H kapısı ile kontrol edilebilmektedir. Motorlar fazla akım çeken devre elemanları olduğu için motorların yönü transistörler yardımıyla kontrol edilmektedir. Böylece motorların sağa veya sola dönüşlerini kontrol edebilmek mümkün olmaktadır.

Aşağıdaki resimde 2 H köprüsü görülmektedir.



Resim 6.10: 2 H köprüsü

1.5 Gözle ve Uygula – Merdaneli Çamaşır Makinesi (Öğrenci 1)

Malzeme Listesi

Arduino UNO

Breadboard

Bağlantı kabloları

L298 sürücüsü

Potansiyometre

Buton

10 K Ohm Direnç

DC Motor

9 V Pil

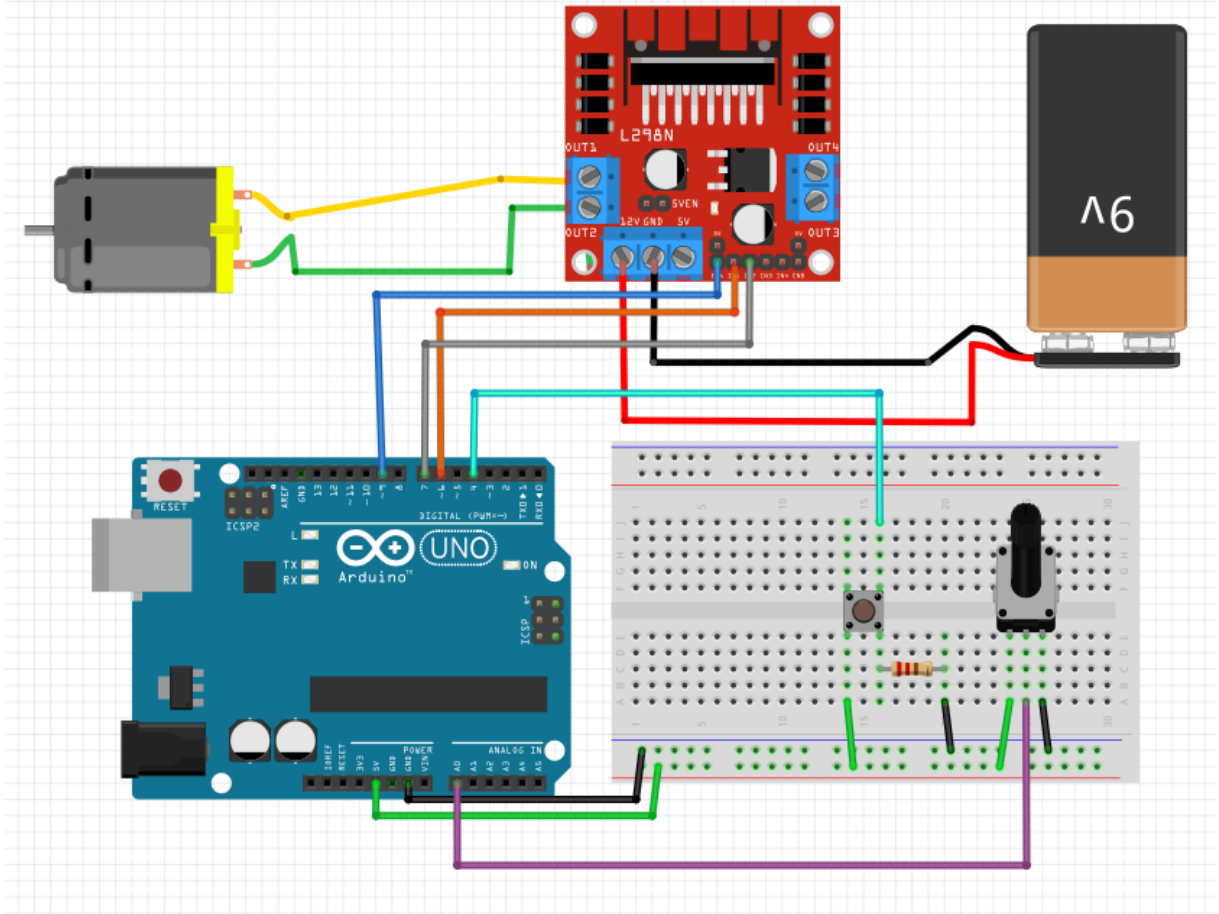
Bu etkinlikte amaç potansiyometre ile dönüş hızı artırılabilen ve açma kapama düğmesi bulunan bir merdaneli çamaşır makinesi tasarlamaktır. Bu çamaşır makinesinde, tuşa basıldığında içerisindeki DC motora bağlı olan merdane 5 sn sağa döner, 3 sn durur, ardından 5 sn sola döner ve yine 3 sn durur. Ayrıca makinenin dönüş hızı da devreye bağlı bir potansiyometre ile ayarlanabilmelidir.

Dikkat

Motordaki bu dönüş hareketini gözlemleyebilmek için DC motorun ucuna tekerlek bağlanabilir. Tekerleğin bir ucuna konacak işaret ile de dönüş yönü ve süresi test edilebilir.

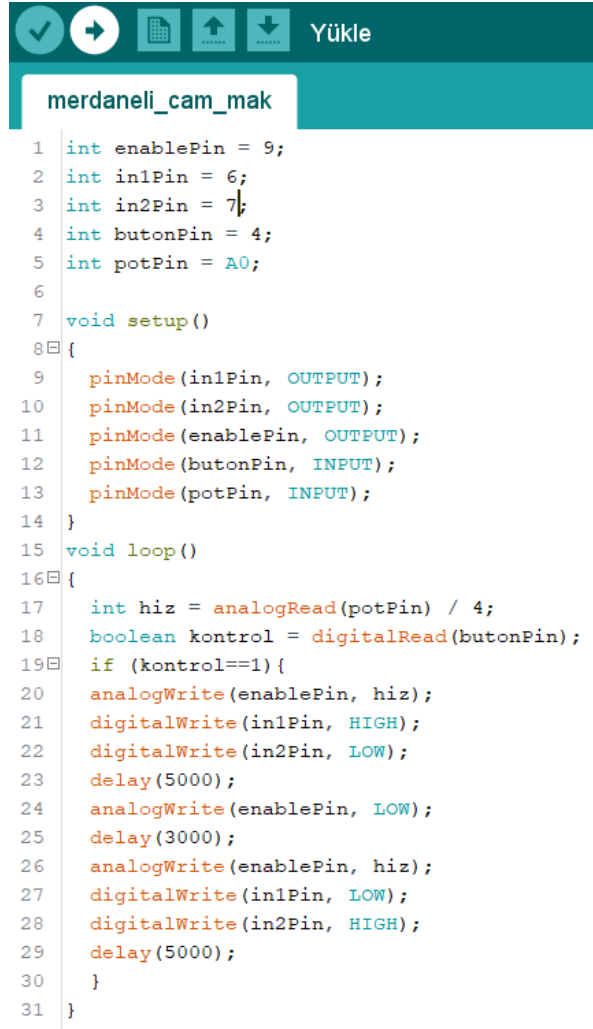
Eğer DC motorların girişlerinde kablolar lehimli değilse etkinliğe başlamadan önce öğrencilerin motorun uçlarındaki bağlantı kablolarını lehimlemeleri gerekmektedir.

Bunun için öncelikle eğitmen devrede ihtiyaç duyulacak devre elemanlarının belirlenmesi için öğrencilerle birlikte akıl yürüterek karar verir. Bunlar belirlendikten sonra devre bağlantıları hakkında öğrenciler ile beraber devre şeması üzerinde açıklamalar yapılır. Ardından eğitmen devreyi öğrencilerle birlikte kurar ve öğrencilere yol gösterir. Örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 6.11: Gözle-Uygula Etkinliği Örnek Devre Şeması

Kodlama kısmına geçmeden önce eğitmen bütün öğrencilerin ilgili devre kurulumunu gerçekleştirdiğinden emin olmalıdır. Devre kurulumundan sonra kodlama sürecinde kullanılacak olan merdaneli çamaşır makinesinin çalışma prensibini anlatır. Örnek kod aşağıdaki resimde görülmektedir.



```

1 int enablePin = 9;
2 int in1Pin = 6;
3 int in2Pin = 7;
4 int butonPin = 4;
5 int potPin = A0;
6
7 void setup()
8 {
9   pinMode(in1Pin, OUTPUT);
10  pinMode(in2Pin, OUTPUT);
11  pinMode(enablePin, OUTPUT);
12  pinMode(butonPin, INPUT);
13  pinMode(potPin, INPUT);
14 }
15 void loop()
16 {
17   int hiz = analogRead(potPin) / 4;
18   boolean kontrol = digitalRead(butonPin);
19   if (kontrol==1){
20     analogWrite(enablePin, hiz);
21     digitalWrite(in1Pin, HIGH);
22     digitalWrite(in2Pin, LOW);
23     delay(5000);
24     analogWrite(enablePin, LOW);
25     delay(3000);
26     analogWrite(enablePin, hiz);
27     digitalWrite(in1Pin, LOW);
28     digitalWrite(in2Pin, HIGH);
29     delay(5000);
30   }
31 }

```

Resim 6.12: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

“in1” - LOW, “in2” - HIGH olursa motor bir yöne döner ve “in1” - HIGH, “in2” - LOW olursa motor ters yöne döner.

Yukarıdaki kodda motoru durdurmak için “enable” pini ile hızı 0 yapılarak motorun durması sağlanmıştır. Motoru durdurmak için kullanılacak bir diğer yöntem de “in1” ve “in2” ye gelen Arduino çıkışları ikisi de LOW veya ikisi de HIGH olursa motor durur.

```

analogWrite(enablePin, hiz); //hızı bir kez yazıyoruz
if (kontrol==1){
  digitalWrite(in1Pin, HIGH); //motoru sağa çevir
  digitalWrite(in2Pin, LOW);
  delay(5000);
  digitalWrite(in1Pin, LOW); //motoru durdur
  digitalWrite(in2Pin, LOW);
  delay(3000);
  digitalWrite(in1Pin, LOW); //motoru sola çevir
  digitalWrite(in2Pin, HIGH);
  delay(5000);
}

```


1.6 Gözle ve Uygula - Servo Motorlar (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
Servo Motor

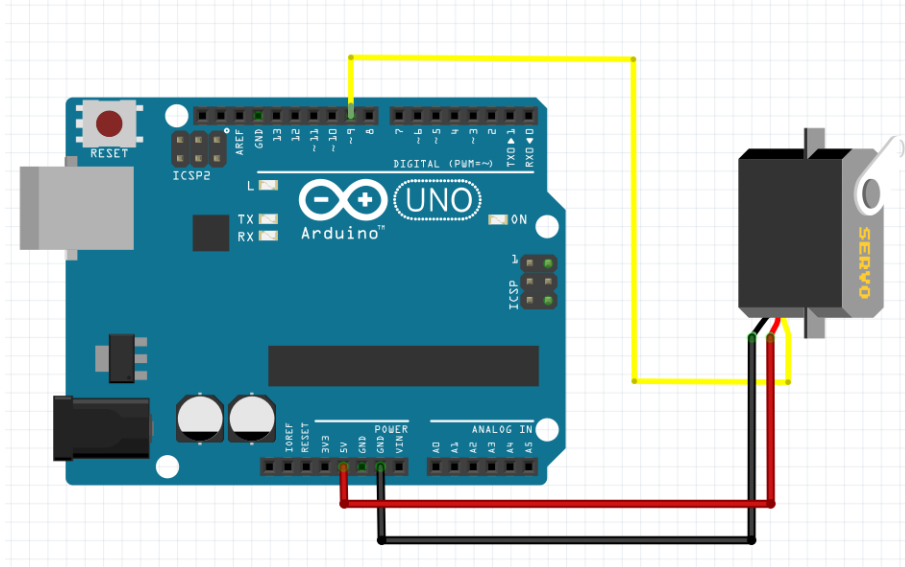
Servo motorların içinde bir potansiyometre veya kontrol devresi bulunmaktadır. AC veya DC adım motor içeren bir servo motorun 3 bağlantı ucu vardır. Uçlardan ikisi motorun çalışmasını sağlarken diğer uç giriş birimine motor shaftının konumunu bildirir. Özet olarak servo motorlar ile diğer motorlar arasındaki temel fark servo motorların çok hızlı bir şekilde ivmelendirilebilmeleri ve frenleme yapılabilmesidir. Ayrıca servo motorlarda tork kontrolü de yapılabilmektedir. Dolayısıyla servo motorlarda devir sayıları hızlı ve düzgün bir şekilde değiştirilebilmektedir. Bu durum da servo motorlarda küçük boyuttan büyük boyuta moment elde edilebilmesini sağlamaktadır. Aşağıdaki resimde servo motor görülmektedir.



Resim 6.13: Servo Motor

Bu özellikleriyle servo motorlar RC (radio controlled – uzaktan kumandalı) arabalar, helikopterler ve uçaklarda kullanılır. Arabalarda direksiyon, helikopterlerde pervane ve uçaklarda ise kontrol yüzeylerinin hareketlerini gerçekleştirmek için kullanılan servo motorlar genellikle 0-180 derece arasındaki açılarda çalışırlar. PWM sinyal ile bir servo motor 0 dereceden 180 derece arasında istenilen konumda ayarlanabilir.

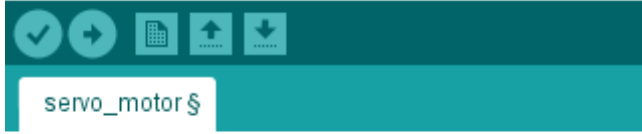
Aşağıda resimlerde servo motorun birer derecelik açılar ile 180 derece içinde dönmesine yönelik uygulama yer almaktadır. Eğitmenin öncelikle devrenin kurulumunu daha sonra Arduino IDE programını öğrenciler ile birlikte değerlendireceği uygulama için Arduino UNO üzerinde 9 numaralı PWM pininin seçildiği öğrencilere gösterilmelidir. Ayrıca Arduino IDE üzerinde yazılan kod içerisinde “#include” ile Servo kütüphanesinin programa eklendiği ve void fonksiyonlarından önce myservo isimli bir nesnenin tanımlandığı öğrencilere vurgulanmalıdır. Void setup fonksiyonu içerisinde “myservo.attach(9)” ile servonun “9.” pine eklendiği; loop fonksiyonu içerisinde de “myservo.write(pos)” ile servoya pos değerindeki pozisyona gitme komutunun verildiği belirtilmelidir.



Resim 6.14: Gözle-Uygula Etkinliği Örnek Devre Şeması

servo_motor | Arduino 1.8.9 (Windows Store 1.8.21.0)

Dosya Düzenle Taslak Araçlar Yardım



```
#include <Servo.h>
Servo myservo;
int pos=0;

void setup() {
  myservo.attach(9);
}
void loop() {
  for (pos=0;pos<=180;pos+=1)
  {
    myservo.write(pos);
    delay(15);
  }
  for(pos=180; pos>=0; pos-=1)
  {
    myservo.write(pos);
    delay(10);
  }
}
```

Resim 6.15: Gözle-Uygula Etkinliği Örnek Kodu

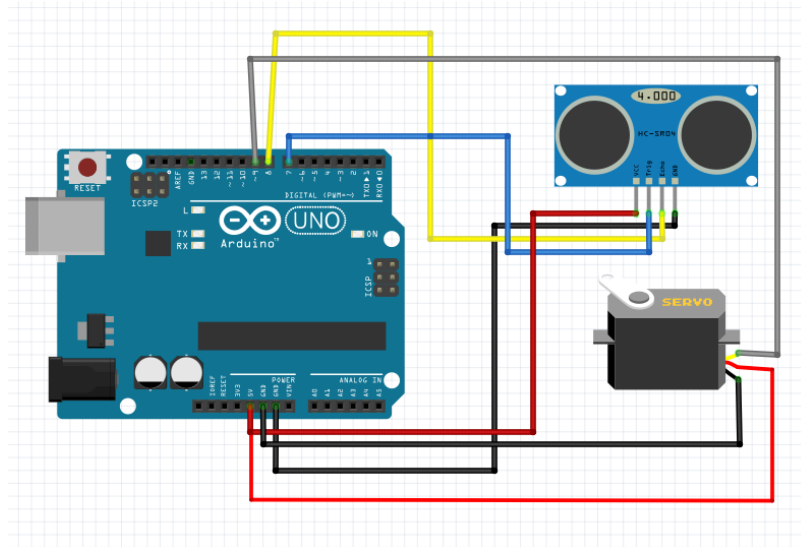
Dikkat

Motorun dönüş hareketi yukarıdaki kodda şu şekilde gerçekleşmektedir.

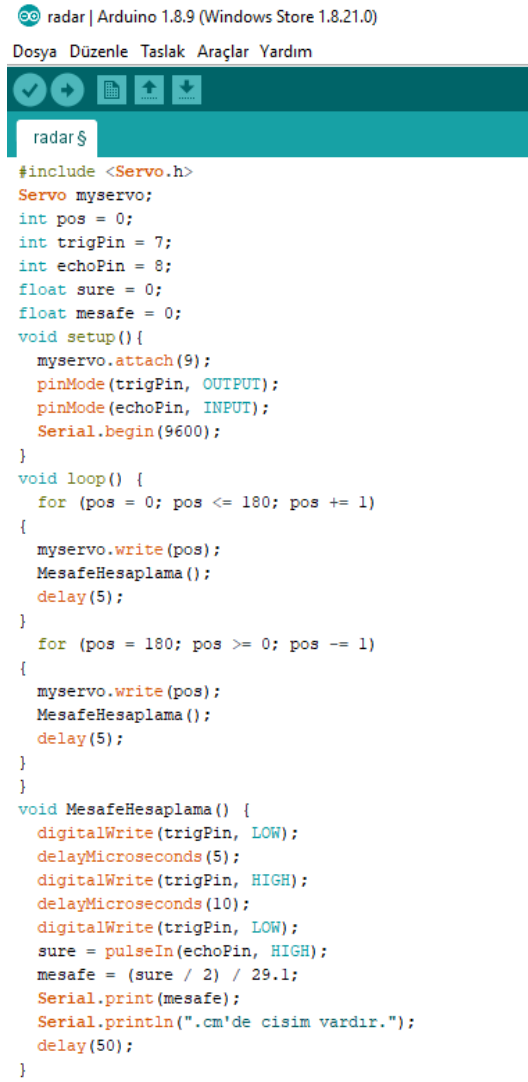
Motor önce bir tarafa 180 derece döner, sonra ters yöne dönerek başlangıç noktasına (0 dereceye) gelir. Bu dönüşünü “for” döngüsüyle sağlar ve motor her bir derecelik dönüşten sonra bir yönde 15ms, diğer yönde 10ms bekler.

1.7 Uygula – Hareketli Sonar ile Mesafe Ölçme (Öğrenci 2)**Malzeme Listesi****Arduino UNO****Breadboard****Bağlantı kabloları****Servo Motor****Ultrasonik sensör****HC-SR04 sensör tutucu**

Bu etkinlikteki amaç servo motor ve ultrasonik mesafe sensörü kullanarak taranan bir alandaki nesnelerin uzaklıklarını seri port ekranına yazdırmaktır. Öğrencilerin bir önceki uygulamadaki devreyi ve Arduino IDE programını kullanmaları süre açısından yerinde olacaktır. Devre kurulumunda öğrenciler tarafından Breadboard üzerinde servo motor bağlantılarının yanısıra ultrasonik mesafe sensör bağlantıları da gerçekleştirilmelidir. Ayrıca hem ultrasonik sensör hem de servo motor *HC-SR04 sensör tutucu* ile birleştirilmelidir. Eğitimci bu uygulama etkinliğinde öğrencilere uygun şekilde devreyi kurmalarını ve Arduino IDE içerisinde programı yazmalarını isteyecektir. Aşağıdaki resimde örnek devre şeması ve programın örnek bir kodu verilmiştir. Örnek kod içerisinde görülebileceği gibi daha önceki derslerde ultrasonik mesafe sensör kullanımına yönelik Arduino IDE programındaki kodlar burada ‘MesafeHesaplama()’ isimli fonksiyon altında kullanılmıştır. Bu uygulamada uygulama kodu void loop fonksiyonu içerisinde servo açı derecesinin yazdırılmasından sonra ‘MesafeHesaplama()’ fonksiyonunun çağırılmasından ibarettir.



Resim 6.16: Uygula Etkinliği Örnek Devre Şeması



```

radar | Arduino 1.8.9 (Windows Store 1.8.21.0)
Dosya Düzenle Taslak Araçlar Yardım
radar$
#include <Servo.h>
Servo myservo;
int pos = 0;
int trigPin = 7;
int echoPin = 8;
float sure = 0;
float mesafe = 0;
void setup() {
  myservo.attach(9);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1)
  {
    myservo.write(pos);
    MesafeHesaplama();
    delay(5);
  }
  for (pos = 180; pos >= 0; pos -= 1)
  {
    myservo.write(pos);
    MesafeHesaplama();
    delay(5);
  }
}
void MesafeHesaplama() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  sure = pulseIn(echoPin, HIGH);
  mesafe = (sure / 2) / 29.1;
  Serial.print(mesafe);
  Serial.println(".cm'de cisim vardır.");
  delay(50);
}

```

Resim 6.17: Gözle-Uygula Etkinliği Örnek Kodu

1.8 Uygula – Joystick Kontrollü Hareketli Sonar ile Sesli ve Görsel Uyarı Sistemi (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
Servo Motor
Ultrasonik sensör
HC-SR04 sensör tutucu
Joystick
Buzzer
LCD Ekran

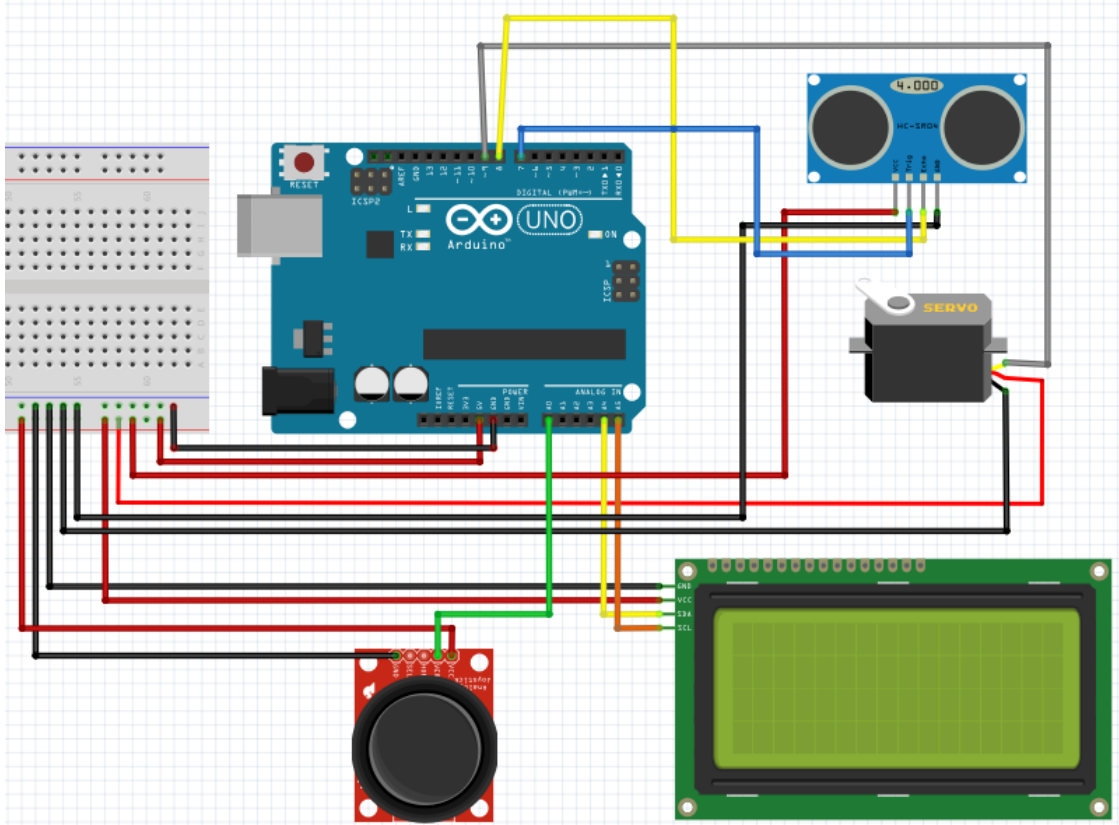
Bu etkinlikteki amaç servo motorun yönünü joystickten aldığı konuma göre (tek bir eksen kullanılarak) hareket ettirmek ve ultrasonik mesafe sensörü kullanarak taranan bir alandaki nesnelerin 5 cm'den daha yakın olması durumunda Buzzer ile uyarı sesi vererek LCD ekrana nesnenin mesafesini yazdırmaktır. Bu uygulamayı gerçekleştirmek için öğrencilerin kurulmuş olan bir önceki devre üzerine joystick, LCD ekran ve Buzzer devre elemanlarını eklemeleri ve Arduino IDE programı üzerinde gerekli kodları yazmaları gerekmektedir. Eğitimci bu aşamada daha önce öğrenilmiş olan bu becerileri öğrencilere göstermemeli; uygulamanın tamamını öğrencilerin gerçekleştirmesini sağlamalıdır. Gerekli yerlerde eğitimci öğrencilere rehberlik etmelidir.

Bu etkinliğe başlamadan önce bu etkinlikte ilk defa kullanılacak olan joystick ve onun çalışma mantığı öğrencilere açıklanır. Aşağıdaki resimde joystick görülmektedir.



Resim 6.18: Joystick

Joystick modülü üzerinde 5 pin bulunmaktadır. Bu pinlerden “VRx” yatay eksendeki sinyalleri, “VRy” dikey eksendeki sinyalleri, “SW” pini ise joystick tıklama buton pin değerini okumayı sağlar. Joystick modülü yatay ve dikey ekseninde 0 ile 1023 arasında analog değerler üretir. Bu nedenle joystickten aldığımız bu çıkışlar Arduino UNO'nun analog pinlerine bağlanmalıdır. Bu etkinlikte servo motor bir eksenle kontrol edileceğinden, joystickin bir eksenini Arduino UNO'nun A0 pinine bağlanacaktır. Aşağıdaki resimlerde etkinlikle ilgili örnek devre şeması ve kod görülmektedir.



Resim 6.19: Uygula Etkinliği Örnek Devre Şeması

```

21 void loop() {
22   deger = analogRead(A0);
23   pos = map(deger, 0,1023,0,180);
24   myservo.write(pos);
25   MesafeHesaplama();
26 }
27 if (mesafe < 5.00) {
28   digitalWrite(buzzerpin, HIGH);
29 }
30 else
31   digitalWrite(buzzerpin, LOW);
32   lcd.home();
33   lcd.print(mesafe);
34   lcd.setCursor(10,0);
35   lcd.println(".cm mesafede cisim vardır");
36   delay(100);
37 }
38 void MesafeHesaplama() {
39   digitalWrite(trigPin, LOW);
40   delayMicroseconds(5);
41   digitalWrite(trigPin, HIGH);
42   delayMicroseconds(10);
43   digitalWrite(trigPin, LOW);
44   sure = pulseIn(echoPin, HIGH);
45   mesafe = (sure / 2) / 29.1;
46   Serial.print(mesafe);
47   Serial.println(".cm'de cisim vardır.");
48   delay(50);
49 }

```

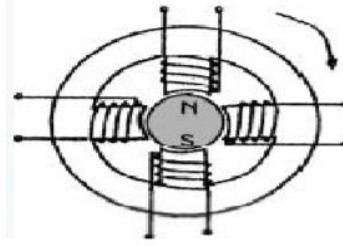
Resim 6.20: Uygula Etkinliği Örnek Kodu

1.9 Gözle ve Uygula – Step Motor (Öğrenci 2)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Step Motor (28BYJ-48) ve ULN2003A sürücü

Step motor adım adım dönen fırçasız bir DC motordur. Gönderilen sinyallere bağlı olarak oluşan adımlar sayılarak motor milinin konumu bir sensöre ihtiyaç duyulmadan hesaplanabilmektedir. Step motorun çalışma mantığını anlayabilmek için öncelikle iç yapısını bilmekte fayda vardır. Step motorlar rotorunda bir mıknatıs barındırır ve etrafında stator sargılarıyla çevrilidir. Bu sargılara uygun sinyaller gönderilerek rotorun açısal kontrolü adım adım gerçekleştirilir. Bu adım adım kontrolde motorun içindeki rotorun ne kadar döneceği motorun yapısına bağlı olarak değişir. Mesela 200 adımlı bir motorun her bir adımı 1,8 derecedir ve 360 derecelik bir turu 200 adımda gerçekleştirebilir. Motora uygulanacak sinyallerin frekansı değiştirilerek motorun hızı da kontrol edilebilir.

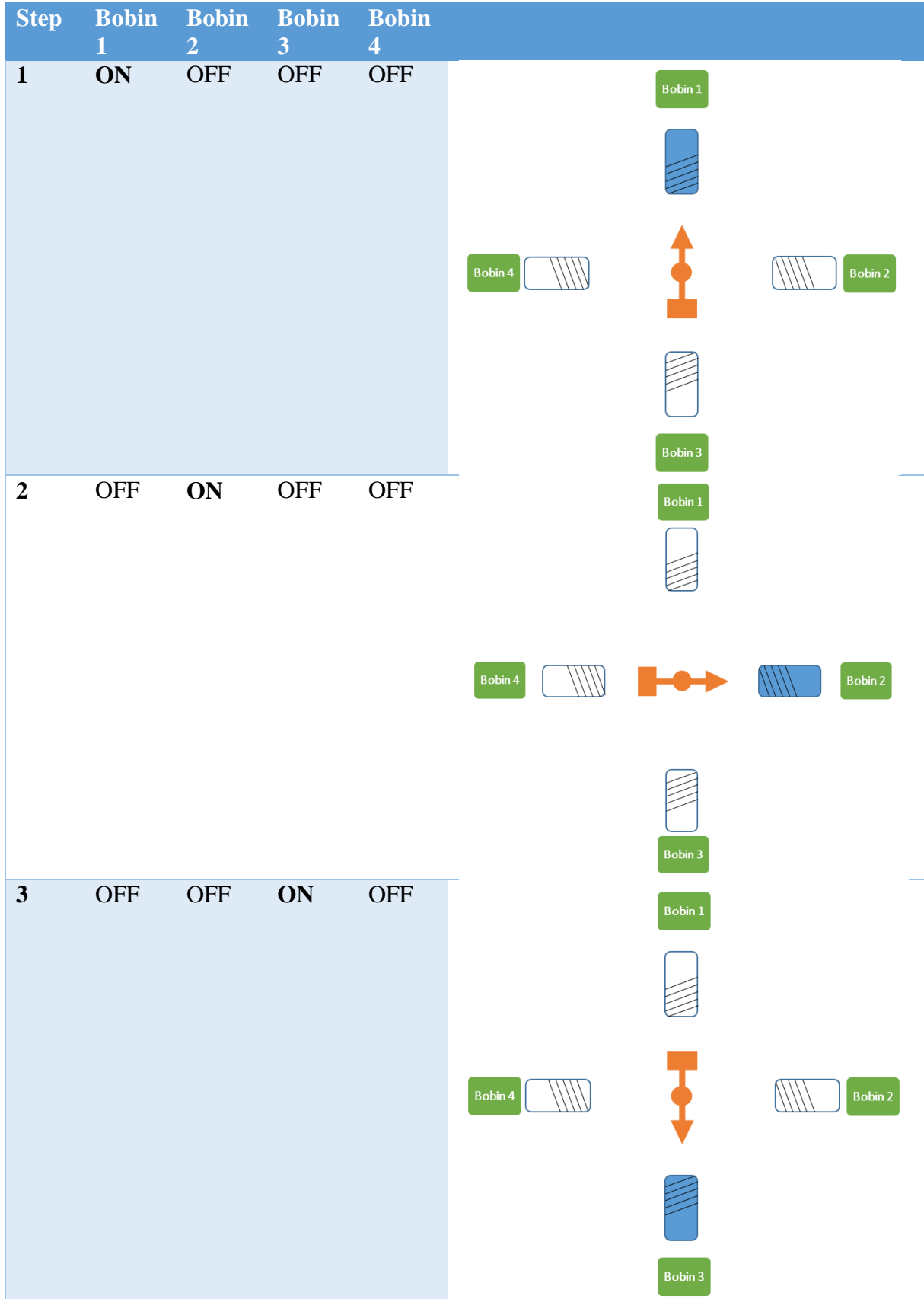
Aşağıdaki resimde gösterildiği gibi step motorların statorunda da 4 tane kutup vardır ve sırayla bütün kutuplara sinyal uygulanır. Bu uygulanan sinyaller sayesinde bir manyetik alan oluşur ve rotora hareket verilir.

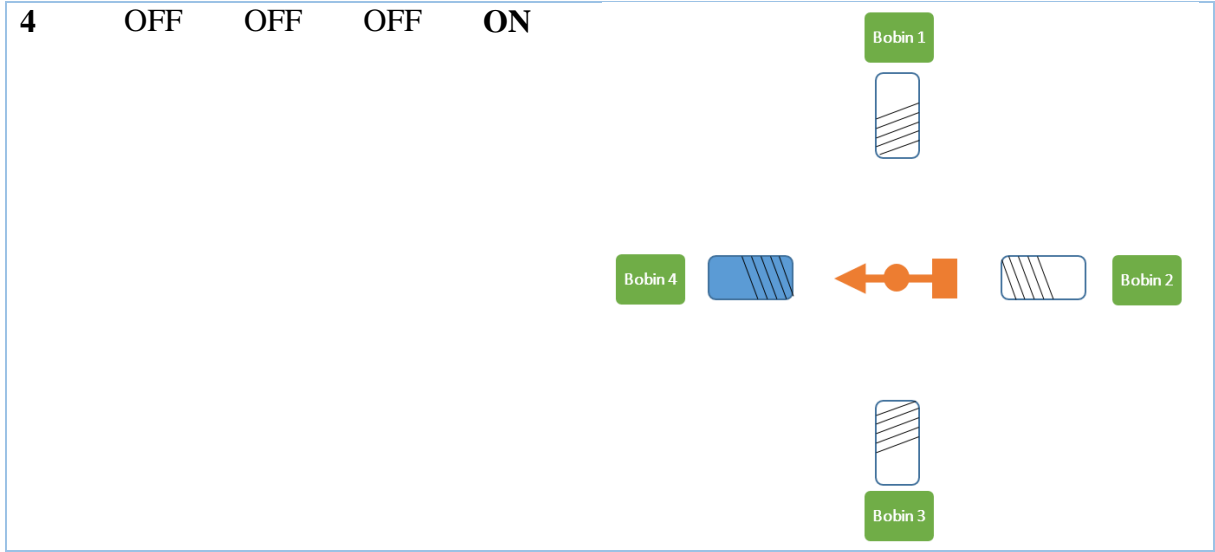


Resim 6.21: Step Motor İç Yapısı

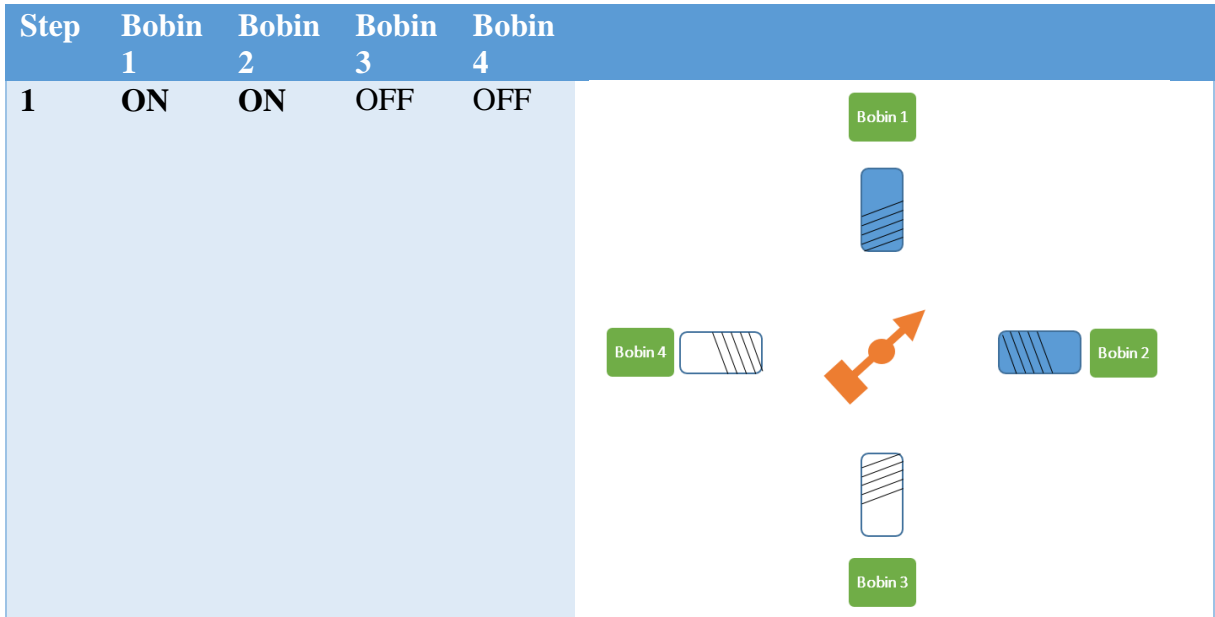
Adım motorlara hareket verirken 3 farklı yöntemle impuls verilebilir.

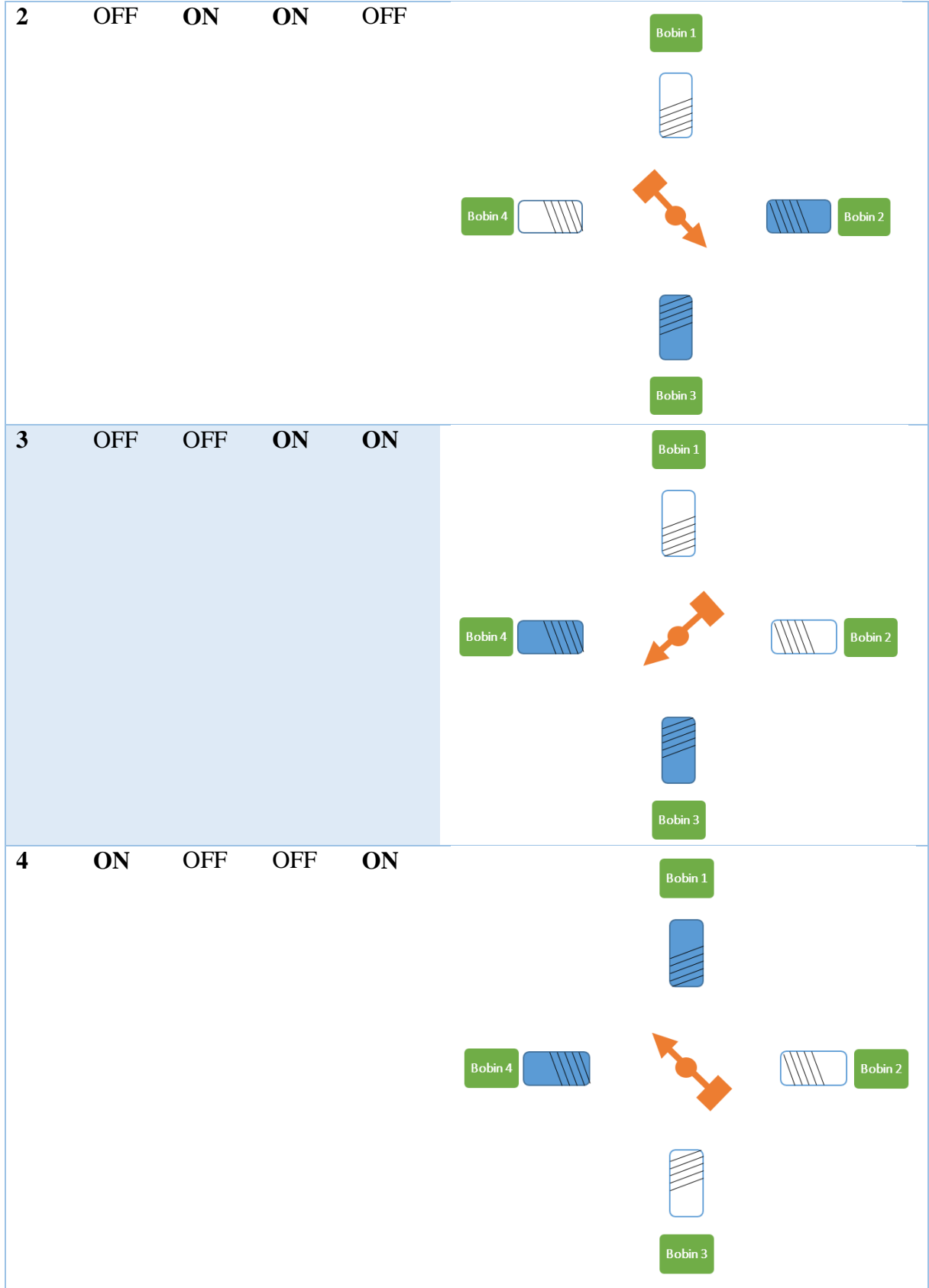
Tek fazlı impulsda statordaki sargılar sırayla ve tek tek uyarılır. Rotor da buna göre manyetik alandan dolayı yön değiştirir. Aşağıdaki tabloda bunun nasıl gerçekleştirildiğini anlatan bir açıklama vardır.



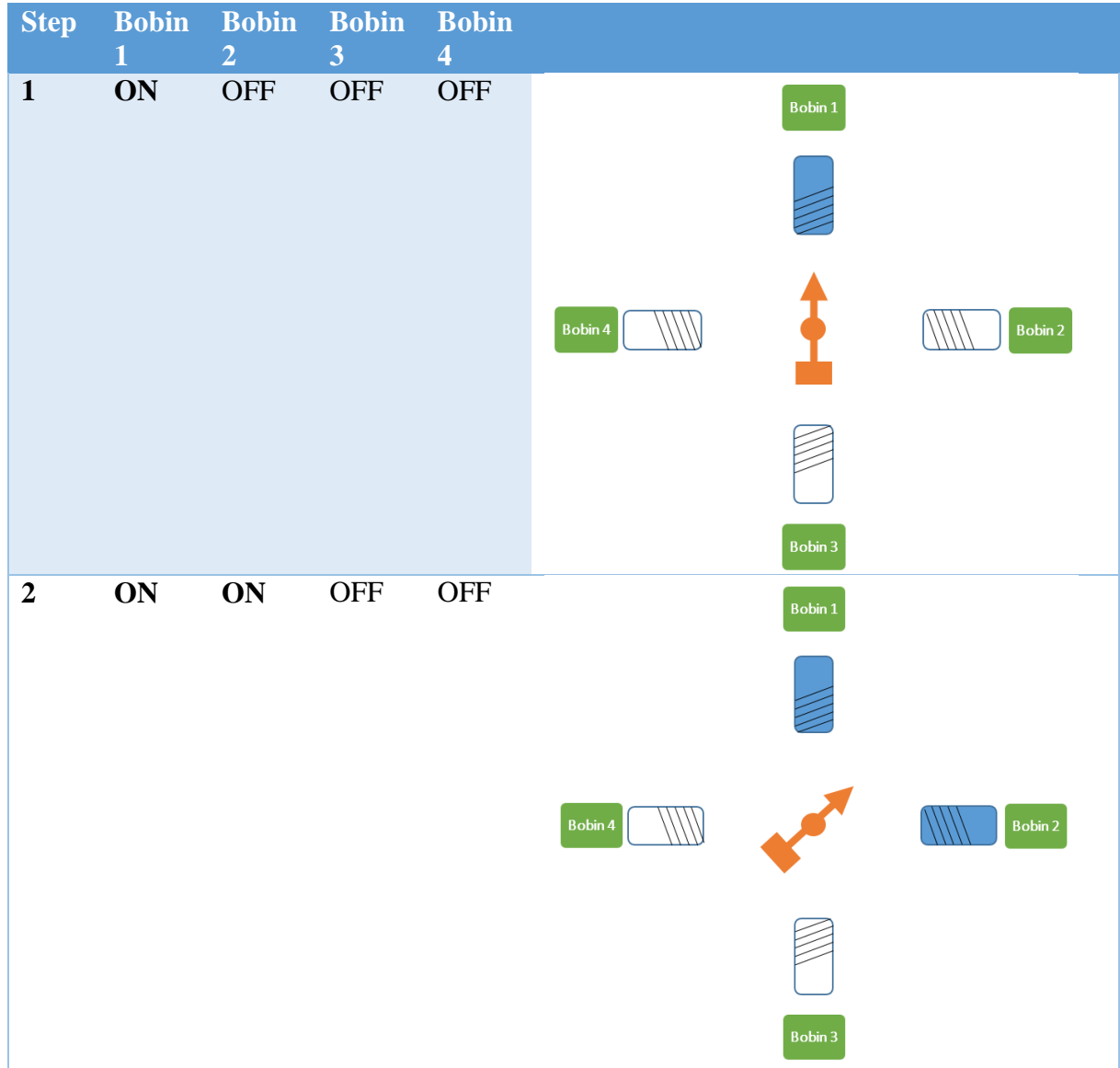


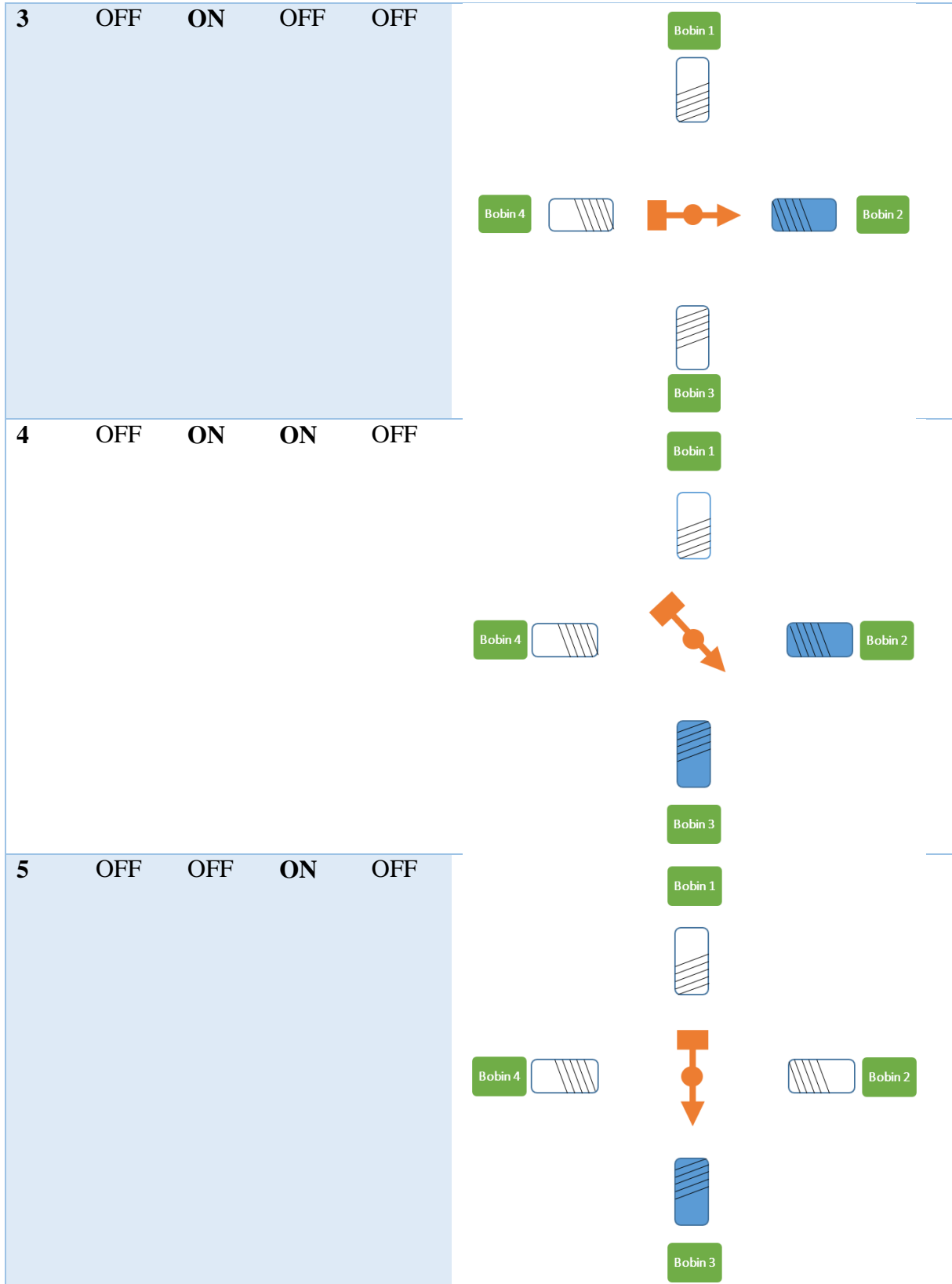
İki fazlı impulsta ise motor sargılarının ikisi aynı anda uyarılır ve iki stator bobini uyarıldığında oluşan manyetik alandan dolayı rotor konumu değişir. Bu tür kullanımlarda rotorun tepkisi, tek-faz impulsa göre daha hızlıdır fakat çekilen güç, iki katına çıkmaktadır. Aşağıda bu step motorun çalışmasına yönelik bir tablo yer almaktadır.

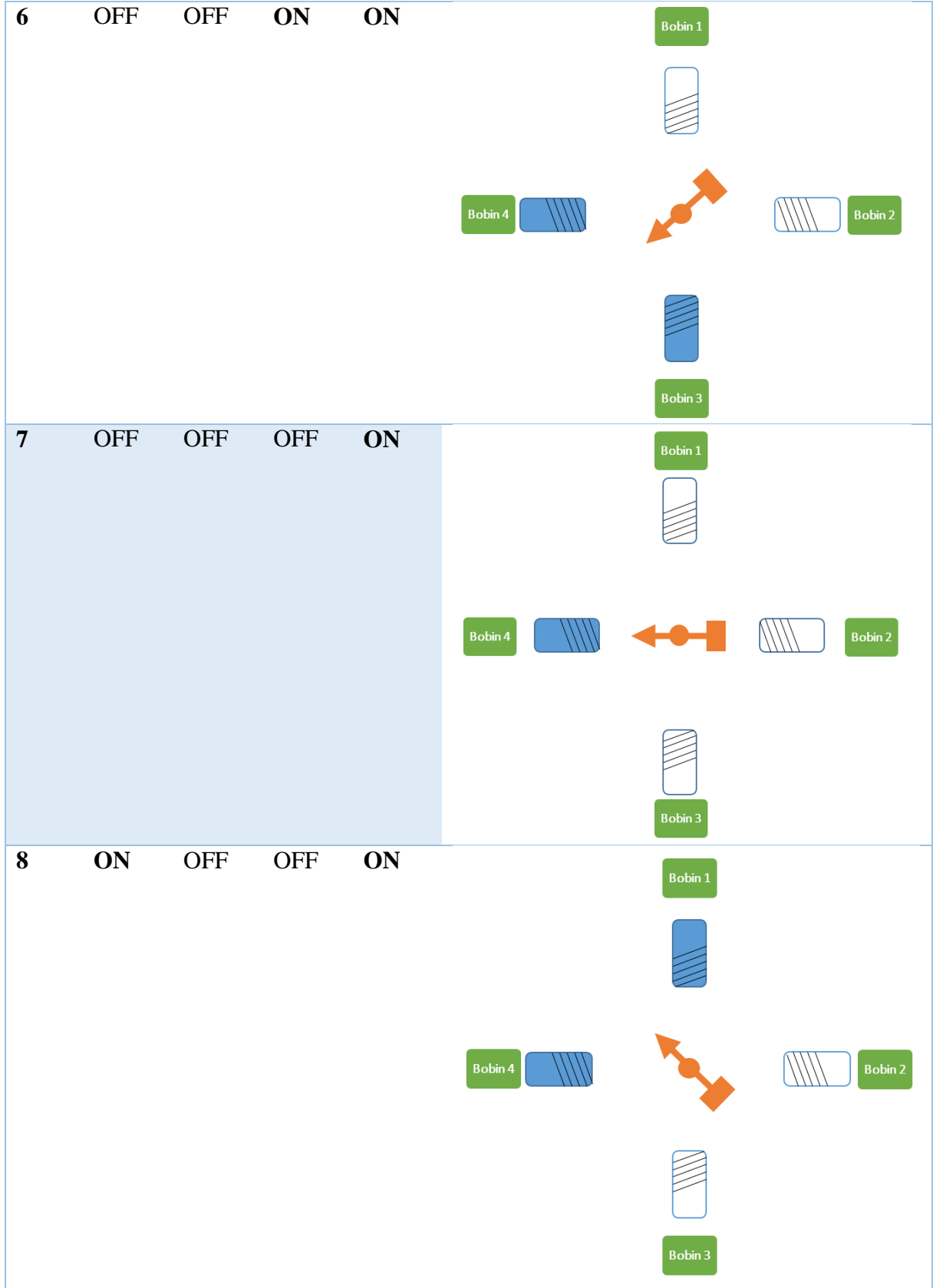




Karma impulsta ise motor sargıları tek tek ya da ikisi aynı anda uyarılır ve stator bobinleri uyarıldığında oluşan manyetik alandan dolayı rotor konumu değişir. Böylece rotorun adım açısı çok daha hassas şekilde ayarlanabilmektedir.





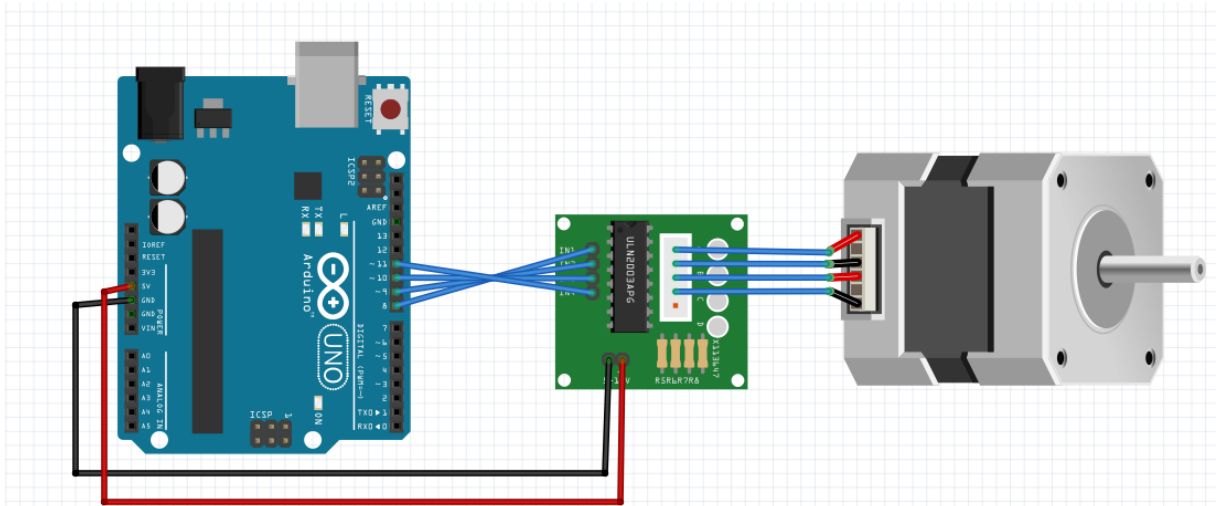


Step motorlar hassas konum kontrolü istenen devrelerde çok sıklıkla kullanılır (**Fakat bu motorlar devrelerde sürekli bir hareket sağlayamaz, hareket daha çok darbelidir**). Arduino UNO ile sırayla bobinlere sinyal verilerek step motor kontrol edilebilir. Fakat step motorun her bir bobinine 100mA daha fazla bir akım uygulanması gerekir. Bu da Arduino UNO'nun verebileceği akımın üzerinde olduğu için bir sürücü devresine ihtiyaç vardır. Elimizdeki setlerde step motorla beraber “**ULN2003**” çipli bir sürücü kartı bulunmaktadır. Bu devrede dört giriş bulunur. Her giriş bir bobini kontrol eder. Ayrıca sürücü kartında 5-12 V arasında besleme verilmesini sağlayan güç girişi de bulunmaktadır. Sürücü kartının üzerindeki LED'ler ise hangi bobine sinyal gittiğini göstermek için kullanılmaktadır. Aşağıdaki resimde step motor görülmektedir.



Resim 6.22: Step Motor

Step motordan gelen soket öncelikle sürücüye bağlanmalıdır. Ardından sürücü ile Arduino UNO bağlantısı yapılmalıdır. Bu etkinlikte amaç step motorun 1 tam tur (360 derece) saat yönünde belirli bir süratte dönmelerini sağlamak, ardından 1 tam tur (360 derece) saat yönünün tersine seri port ekranından girilen süratte dönmelerini sağlamaktır. Bunun için eğitmen öğrencilerle beraber aşağıdaki resimdeki devreyi kurar, bağlantıları uygun şekilde gerçekleştirir ve kodu yazar. Ardından devreyi öğrencilerin kurmasını ve kodlamasını ister. Bu süreçte eğitmen öğrencilere devre kurulumu ve kodlama süreçlerinde destek olur.



Resim 6.23: Gözle-Uygula Etkinliği Örnek Devre Şeması



```

stepper_1_1
1 #include <Stepper.h>
2
3 const int adim = 200;
4 int hiz;
5 Stepper myStepper(adim, 8, 9, 10, 11);
6
7 void setup() {
8
9     Serial.begin(9600);
10 }
11 void loop() {
12     if(Serial.available()>0){
13         hiz=Serial.read();
14         Serial.print("Step Motor İçin Hız Girişi Yapın:");
15         Serial.println(hiz);
16     }
17     |
18     myStepper.setSpeed(hiz);
19     Serial.println("Saat Yönünde Dönüş");
20     myStepper.step(adim);
21     delay(500);
22     Serial.println("Saat Yönü Tersine Dönüş");
23     myStepper.step(-adim);
24     delay(500);
25 }

```

Resim 6.24: Gözle-Uygula Etkinliği Örnek Kodu

2. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün yaptıkları gözle-uygula etkinlikleri için sınıf içerisinde öğrenciler seçilir ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

- Motor türleri ve kullanım alanları neler olabilir?
- Motorlarda sürücü kartları neden kullanılır?
- Bu derste gördüğümüz motor sürücülerinin çalışma mantığını anlatınız?
- Motorların çalışma mantıklarını sırasıyla açıklayınız?
- Joystick ile gözle-uygula etkinliklerinde Arduino UNO'ya nasıl veri aktarıldığını ve motorların nasıl kontrol edildiğini açıklayınız.
- Tasarım süreciyle ilgili ne tür sorunlar yaşadınız, nasıl çözdünüz?
- Devrelerin fiziki kurulumunda en fazla nerelerde zorlandınız?
- Arduino IDE yazılımında “kütüphane” kullanımının sebepleri ve avantajları nelerdir?
- Program algoritmalarını oluştururken hangi süreçlerde zorlandınız?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşmaya kadar devam ettirilebilir.

3. ADIM: EK ETKİNLİK -

Öğrencilerden bu etkinlik kapsamında 2 step motorun X ve Y ekseninde joystick ile kontrolüne yönelik bir devre tasarımları ve buna uygun kodu yazmaları istenmektedir.

Dikkat

Bu haftanın tasarla-üret etkinliği gelecek hafta uygulanacaktır. Bu etkinlik kapsamında gelecek hafta öğrenciler tarafından kâğıttan yapılmış bir uçak fırlatma rampası tasarlanacaktır. Gelecek haftaya kadar öğrenciler bu konuda tasarımlar üzerinde çalışmalıdırlar. Kâğıt uçak fırlatmak için elimizdeki setlerdeki malzemeleri (özellikle motorları) kullanarak farklı fırlatma rampası tasarımları oluşturmaları istenecektir. Öğrencilerden gelecek hafta uçak rampa tasarımında kullanılması için ayakkabı kutusu getirmeleri istenmelidir. Ayrıca fırlatma rampasında kullanılacak olan kâğıt uçak tasarımları da önemli olduğu için haftaya kadar farklı katlama yöntemleri kullanılarak kâğıt uçak tasarımları hazırlamaları da gerektiği vurgulanmalıdır.

7. Bölüm - Kâğıt Uçak Fırlatma Rampası

Ön bilgi:

- Öğrenciler DC, step ve servo motorların çalışma prensiplerini ve sürücü bağlantılarını bilir.
- Arduino IDE üzerinde DC, step ve servo motorları çalıştırabilecek ve kontrol edebilecek kodları yazabilir.
- Öğrenciler joystick'in çalışma prensiplerini bilir.
- Öğrenciler Arduino UNO'yu tanır ve bunu kullanarak devreleri kurup kodlarında düzenleme yapabilir.
- Öğrenciler Arduino IDE arayüzü ile değişkenler, “if/else, for” döngüsü, dizi vb. yöntemleri ile kod yazarak elektronik devre elamanlarını yönetebilir.

Bölüm Kazanımları:

- Öğrenciler DC motorların çalışma prensiplerini ve sürücüsünün bağlantılarını bilir.
- Arduino IDE üzerinde DC motorları çalıştırabilecek ve kontrol edebilecek programı yazabilir.
- Öğrenciler servo motorların çalışma prensiplerini bilir.
- Arduino IDE üzerinde servo motorları çalıştırabilecek programı yazabilir.
- Öğrenciler step motorların çalışma prensiplerini bilir.
- Arduino IDE üzerinde step motorları kontrol edebilecek programı yazabilir.
- Öğrenciler joystickin çalışma prensiplerini bilir.
- Arduino IDE üzerinde joystick kullanarak dc, step ve servo motorları kontrol edebilecek programı yazabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle joystick, DC, servo ve step motorların çalışma prensipleri hakkında bilgi sahibi olmalarıdır. Bu haftaki “Tasarla-Üret” etkinliğinde motorların ve sürücü devrelerinin tasarımı ve bağlantılarını konusunda uygulama yaptırılacaktır. Motorların farklı tasarımlarda bir arada kullanımı ve buna uygun devre tasarım süreçleri pekiştirilecektir. Ayrıca kurulan bu devrelerin Arduino IDE ile kodlanması sağlanacaktır.

Haftanın İşlenişi:

Tasarla: DC motor, servo motor ve step motorun çalışma prensibi kullanılarak verilen senaryoya uygun devre tasarımının oluşturulmasıdır.

Üret: DC motor, servo motor ve step motor ile örnek devrelerin oluşturulması ve programlanması.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

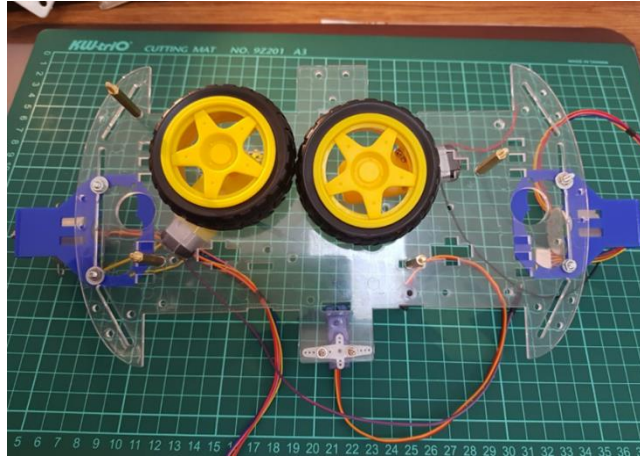
2. ADIM: TASARLA ve ÜRET

Bu tasarla-üret etkinliği motorlar ile ilgili olarak geçen hafta gerçekleştirilen “Gözle-Uygula” etkinliklerinin devamı niteliğindedir. Bu etkinlik kapsamında “kâğıt uçak fırlatma rampası” tasarlanacaktır. Kâğıt uçak fırlatmak için elimizdeki setlerdeki malzemeleri (özellikle motorları) kullanarak öğrencilerin farklı fırlatma rampası tasarımları oluşturmaları istenecektir (gerekirse yanlarında getirdikleri kutu ve atölyelerde yer alan mukavvalar da bu tasarımlarda kullanılabilir). Ayrıca fırlatma rampasında kullanılacak olan kâğıt uçak tasarımları da önemli olduğu için farklı katlama yöntemleri kullanılarak kâğıt uçak tasarımları hazırlamaları da gerekmektedir.

2.1 Tasarla – Kâğıt Uçak Fırlatma Rampası

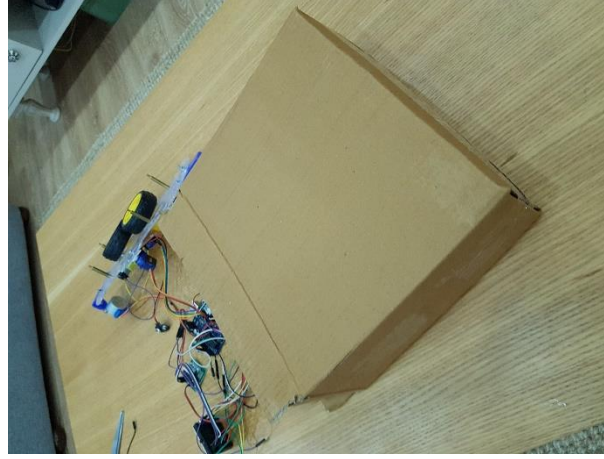
Bu etkinlikte 2 adet DC motor, 2 adet tekerlek, 2 step motor, 1 servo motor, 1 joystick, araba platformu, ayakkabı kutusu ve setteki diğer bağlantı aparatlarını kullanarak bir kâğıt uçak fırlatma rampası tasarlanacaktır. Kâğıt uçak fırlatma rampası aşağıdaki özelliklere göre tasarlanmalıdır.

1. Bütün bağlantılar araba platformu üzerinde gerçekleştirilebilir.
 2. Tasarım sürecinde öğrenciler serbest olmakla birlikte DC motorlar ve bunlara bağlı olan tekerlerin arasına kâğıt uçaklar sıkıştırılarak DC motorların dönüşüyle uçak fırlatma işlemi gerçekleştirilecektir.
 3. Servo motor fırlatma platformunun yatay ekseninde hareketini kontrol edecektir.
 4. Step motor fırlatma platformunun dikey ekseninde hareketini kontrol edecektir. Joystick devrede yer alan motorların tümünü kumanda edecektir. DC motorlar joystickte tıklandığında harekete geçeceklerdir.
 5. Setlerdeki farklı aparatlar bu işlem için kullanılabilir. Stabil bir platform olması açısından araba platformunun bir ya da birkaç katı boyutta olması ve aralayıcıların kullanımı önerilmektedir.
 6. Bu platformu sabitlemek için öğrenciler yanlarında getirdiği ayakkabı kutularını da kullanabilirler.
- Aşağıdaki resimde örnek tasarım görülmektedir.



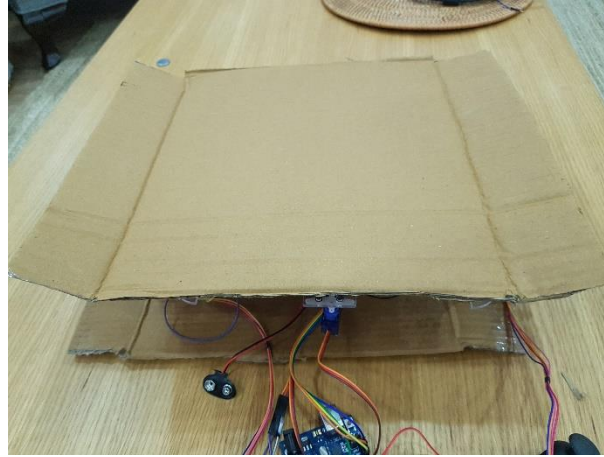
Resim 7.1: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Aşağıdaki resimlerde görünen motor yerleşimi örnek olarak verilmiştir. Öğrenciler kendi tasarımlarında tamamen özgürdür. Bu nedenle öğrencilerin yaptıkları tasarımlara göre Arduino IDE ortamında kodlama süreçleri de farklılık gösterecektir.



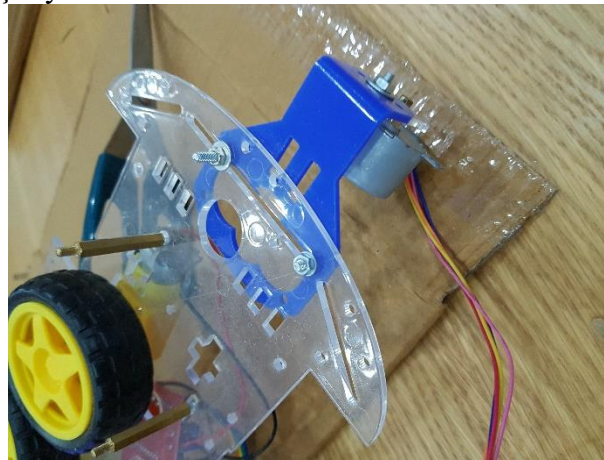
Resim 7.2: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Bu etkinlikte öncelikle kutumuzun alt ve üst zeminlerini keserek 2 parçaya ayırıyoruz. Bunlardan bir tanesi zemini oluşturacak, diğeri ise üst kısmı oluşturacaktır.



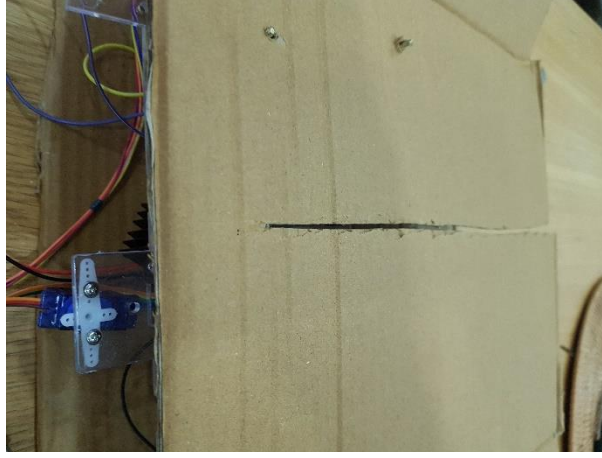
Resim 7.3: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Rampa tasarımı yaklaşık yukarıdaki resimdekine benzer olacaktır.



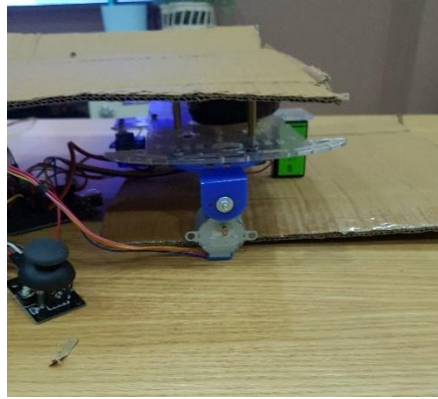
Resim 7.4: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Fırlatma rampası mekanizmasını zemin olarak kullanacağımız karton levhaya silikon tabanca ya da çift taraflı bant ile iyice sabitliyoruz. Bu işlemi yaparken step motor zemin olarak kullanılabilir.



Resim 7.5: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Ardından uçak fırlatma rampasının üst zeminini için platformdaki tekerleklerin arasına gelecek şekilde kartonda yaklaşık 2-3 mm genişliğinde bir kesik açılır. Bu kısım kâğıt uçakların içine yerleştirileceği kısım olacaktır ve tam tekerleklerin arasına girmelidir. Buraya yerleştirilen kâğıt uçağın yüksekliği ve yönü joystick ile ayarlandıktan sonra üzerindeki butona tıklanarak motorlar çalıştırılacak ve uçak fırlatma işlemi gerçekleştirilecektir.



Resim 7.6: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım



Resim 7.7: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Platformun alt ve üst zeminlerinin yerleşimi yukarıdaki resimlerdeki gibi olmalıdır. Üst zemin yerleştirildikten sonra silikon tabancayla platforma sabitlenir.

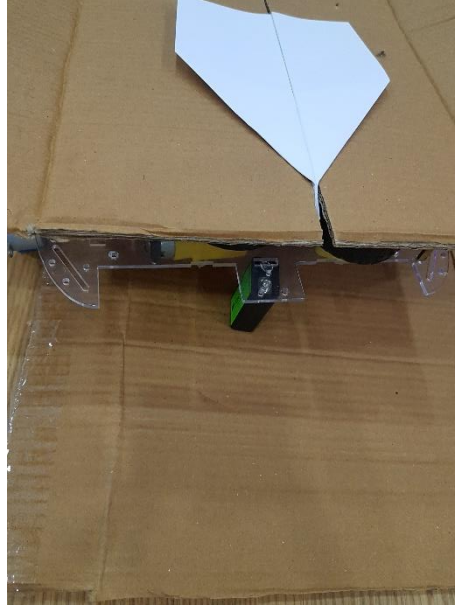


Resim 7.8: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım

Platform sabitlendikten sonra kullanıma hazırdır. Gerekli bağlantılar test edildikten sonra kod yazımına geçilebilir.



Resim 7.9: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım



Resim 7.10: Uçak Fırlatma Rampası Oluşturma Örnek Tasarım



Resim 7.11: Uçak Fırlatma için Örnek Kâğıt Uçaklar

Farklı kağıt uçak tasarımlarıyla sistem test edilebilir. Uçağın ağırlık merkezi ve uzunluğu bu testlerde değerlendirilmelidir.

Dikkat

Bu platform tasarımı örnek olarak verilmiştir. Setlerde farklı parçalar kullanılarak özgün tasarımların yapılması istenmelidir. Tasarıma göre kullanılacak devre elemanları ve bağlantıları değişeceği için bir devre şeması ve kodlar verilmemiştir.

Öğrenciler gruplar halinde oluşturacakları devreyi ve programı tartışırlar. Eğitimci gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Devrenin kurulumunda hangi motorlara, hangi sürücülere ve hangi devre elemanlarına ihtiyaç duyulacaktır?
- Devrenin fiziksel olarak kurulumundan önce hangi motorlar, hangi devre elemanları nasıl bir bağlantı şeması ile bağlanmalıdır?
- Devrenin fiziksel olarak kurulumunda bu devre elemanları setlerdeki ne tür aparatları kullanarak bağlanabilir?
- Fırlatma rampasının tasarımında nelere dikkat etmek gerekir?
- Kurulan devrenin Arduino IDE arayüzünde sorunsuz olarak çalışabilmesi için gereken kütüphaneler nelerdir?
- Arduino IDE içinde yazılacak olan kodlarda hangi tür kontrol yapıları kullanılmalıdır?
- Özellikle joystick kontrolünde motorların kontrolü net olarak tanımlanmalı ve pin bağlantıları ve devre şeması çizilmelidir.
- Arduino IDE içerisinde yazılacak olan kodda joystick kullanarak motorların kontrolü için oluşturulması gereken algoritma nasıl olmalıdır?
- Arduino IDE içerisinde kullanılması gereken değişkenler nelerdir?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir.

- Devre tasarımı sürecinde eldeki malzemeler fikir üretme sürecinde önemli olduğundan setteki farklı aparatlar ve bağlantı gereçleri incelenerek uygun tasarım fikri üzerinde çalışılır.
- Tasarlanan düzenek üzerinde ekip arkadaşı ile beraber devre elemanlarının yerleşimi ve bağlantısı ile ilgili özgün fikirler üretebilirler.
- Devrenin fiziksel kontrolü için joystick kullanılmalı ve Arduino UNO üzerindeki dijital pin bağlantıları ile fonksiyonları öncesindeki tanımlamalar bağlantılardaki pin değerleri ile örtüşmelidir.
- Devredeki ekipmanlar için kullanılması gereken kütüphanelerin Arduino IDE’de bulunma durumunu kontrol eder, yok ise internette veya Arduino IDE’nin kütüphane bölümünden indirir ve Arduino IDE’ye ekler.
- Arduino IDE içerisinde kullanılması gereken değişkenleri belirler.
- Devre tasarımı ve kod dışında kağıt uçak tasarımları da bu fırlatma rampası etkinliğinde önemli olduğu için bu konuda farklı fikirler denenmelidir.

2.2 Üret- Kağıt Uçak Fırlatma Rampası

Öğrenciler çözüme yönelik tasarımlarını yaptıktan sonra bilgisayar ve set başında çalışarak istenilen görevi yerine getirir. Öğrencilere eğitimci tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında değişkenler,

fonksiyonlar, döngüler, if-else yapıları ile çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır.

3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün yapılan “Tasarla-Üret” etkinliği ile ilgili ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

- Motor türleri ve kullanım alanları nelerdir?
- Motorların kontrolünde neden sürücüler kullanılır?
- Motorların çalışma mantıklarını sırasıyla açıklayınız.
- Joystick ile tasarla-üret etkinliklerinde Arduino UNO’ya nasıl veri aktarıldı ve motorları nasıl kontrol ettiniz?
- Tasarım süreciyle ilgili ne tür sorunlar yaşadınız, nasıl çözdünüz?
- Devrelerin fiziki kurulumunda en fazla nerelerde zorlandınız?
- Arduino IDE içerisinde yazılan kodlarda switch-case yapısının kullanılmasının avantajları nelerdir?
- Arduino IDE yazılımında “kütüphane” kullanımının sebepleri ve avantajları nelerdir?
- Arduino IDE yazılımında “kütüphane” kullanımının sebepleri ve avantajları nelerdir?
- Program algoritmalarını oluştururken hangi süreçlerde zorlandınız?
- Öğrencilerden seçilen motorların özelliklerini anlatmaları ve bu motorların hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Bu hafta “Tasarla-Üret” etkinliğinde kullandığımız devre elemanlarının günlük yaşam içindeki kullanım alanları nelerdir? Örneklerle açıklayınız.

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşmaya kadar devam ettirilebilir.

4. ADIM: EK ETKİNLİK - Yarışma

Öğrencilerden oluşturulan takımlar arasında en uzağa kâğıt uçak fırlatan takımın seçilmesi amacıyla bir turnuva düzenlenir. Sınıftaki öğrenci sayısına bağlı olarak 2’şer ya da 3’erli gruplar halinde turnuva düzenlenir. Bu amaçla her ekibe 3 defa uçak fırlatma hakkı verilir ve ölçülen mesafeler kaydedilerek en uzak atışlar kayıt altına alınarak turnuvanın o aşamasının kazanımı belirlenir(Grupların en uzak atışları burada referans alınır). Kazananlar daha sonra kendi aralarında tekrar aynı şekilde gruplandırılır ve yine kazananlar üst tura geçerek turnuvaya devam eder. Bu aşamalarda üst tura çıkan takımlara kendi platformunda ya da uçaklarında iyileştirmeler ve değişiklikler yapma hakkı verilir.

5. PROJEME BAŞLIYORUM (GİRİŞ)

Bu haftadan itibaren Elektronik Programlama ve Nesnelerin İnterneti dersi kapsamında öğrenci grupları oluşturup; öğrenci gruplarından birer proje konusu belirlemeleri istenerek, öğrencilerin ders kapsamında edindikleri bilgileri yaratıcı, çözüm üreten ve sonraki öğrenmelerine destek olabilecek şekilde bir tasarım ile sonlandırması beklenmektedir. Bu kapsamda öğrencilerin üst düzey düşünme becerilerine yönelik gelişim ve grup ile çalışma ve çözüm üretme becerilerinin artırılması hedeflenmektedir.

Öğrenci projeleri çerçevesinde öğrencilerden günlük hayat deneyimlerinde ve çevrelerinde belirledikleri bir sorun için Arduino tabanlı elektronik bir çözüm tasarımları ve bu tasarımları programlamaları beklenmektedir. Bu hafta ile birlikte öğrencilerin gelecek üç hafta boyunca (8., 9. ve 10. haftalar) projelerini tanımlama ve tasarlama etkinlikleri üzerinde çalışmaları, 11. haftada belirledikleri problemi çözmek için tanımladıkları ve tasarladıkları çözümü geliştirme (prototip geliştirme) ve 12. ve 13. haftalarda ise geliştirmiş oldukları çözümü test etmeleri ve sürece yönelik proje kayıt defterinin teslim etmeleri beklenmektedir. Yukarıda belirtilen haftalar kapsamında son çözümü geliştirmek için Tasarım ve Üretim dersinde uygulanan tasarım odaklı düşünmenin beş aşamasının uygulanacağı öğrencilere iletilmelidir. Başlıklar halinde proje sergisi haftasına kadar gerçekleştirilecek işlemler aşağıda özetlenmiştir:

1. **Empati kurma:** Belirlenen günlük hayat problemi için hedef kullanıcı kitlenin ihtiyaç ve beklentilerini irdeleme. (8. Hafta)
2. **Tanımlama:** Kullanıcılardan alınan bilgilerden çıkan sonuçları yorumlayıp ihtiyacı öngörme. (9. Hafta)
3. **Fikir üretme:** Çözüm için çeşitli, yaratıcı ve yenilikçi fikirler üretme. (10. Hafta)
4. **Prototip geliştirme:** Fikirlerin hızlıca uygulandığı prototipler hazırlama. (11. Hafta)
5. **Test etme:** Ortaya çıkan ürünün yeniden hedef kitle ile değerlendirme. (12. -13. Hafta)

Proje sergisinin yapılacağı son haftaya kadar gruplar bu haftadan başlayarak;

- **“Proje Kayıt Defteri”** doldurmaları ve ilgili haftalar boyunca yapılacak işlemleri bu deftere adım adım yazmaları istenecektir. Her hafta adım adım doldurulacak olan proje kayıt defterinde *test etme basamağından önce* gruplar daha önceki haftalarda girilmiş olan bilgileri güncelleyebilir ve belirli oranda değiştirilebilirler. Bu yüzden kitabın ilgili bölüm sonlarında proje kayıt defterindeki başlıklar tekrarlı şekilde verilmiş; bir sonraki haftanın etkinliklerini kapsayacak şekilde ilerleyerek (istendiğinde güncelleme ve değişikliklerin yapılabilmesi için) tasarlanmıştır. Eğitimci gruplara proje kayıt defterinde test etme haftasına kadar değişiklik yapabileceklerinin bilgisini hatırlatmalıdır. Proje kayıt defterinin son ve doldurulmuş versiyonunun 13. hafta sonunda tüm gruplardan isteneceği öğrencilere iletilmelidir.
- Öğrencilerden tüm proje süreçlerini (toplantılar, tasarım çalışmaları, kod geliştirme aşamaları, prototip geliştirme süreçleri, test işlemleri vb.) **“Video”** kaydına almaları gerekmektedir. Son hafta proje kayıt defterinde tüm süreci aktaran bu video linkinin isteneceği ve bu videonun proje sergisinde kullanılacağı öğrencilere hatırlatılmalıdır.

Bu Hafta ve Gelecek Hafta Gerçekleştirilecek Etkinlikler (8. Hafta): Öğrencilerden bu hafta öncelikle gruplarına bir isim bulmaları ve proje gruplarını oluşturarak dokümanın sonunda verilen proje kayıt defterlerine yazmaları istenecektir. Ayrıca eğitimci öğrencilerden gelecek hafta derse gelirken proje kayıt defterinin **EMPATİ** başlığı ile verilen bölümlerini de doldurmaları gerektiği vurgulanmalıdır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki beklentiler nelerdir?

8.Bölüm- Bluetooth Kontrollü 3 Tekerli Mobil Robot

Ön bilgi:

- Öğrenciler Arduino UNO kartı bileşenlerini bilir ve Arduino IDE arayüzünü kullanabilir.
- Öğrenciler Arduino UNO üzerinde elektronik devre elemanları arasında fiziksel bağlantı kurabilir ve Arduino IDE üzerinde program yazabilir.
- Öğrenciler değişken, döngüler ve fonksiyonlar ile ilgili temel kavramları bilir.
- Öğrenciler elektronik devre elemanları arasında fiziksel bağlantıları gerçekleştirebilir.

Bölüm Kazanımları:

- Öğrenciler voltaj regülatörlü çift motor sürücü kartının çalışma prensibi ve kullanımı hakkında genel bilgilere sahiptir.
- Öğrenciler voltaj regülatörlü çift motor sürücü kartı, Arduino UNO ve DC motorlar arasındaki fiziksel bağlantıları yapabilirler.
- Öğrenciler mobil robot platformu içerisindeki parçaları belirli bir sıra düzeninde bir araya getirebilirler.
- Öğrenciler mobil robot tasarımlarını Arduino UNO ile bir arada kullanarak robotun temel hareketlerini gerçekleştirebilecek Arduino IDE programlarını fonksiyonlar kullanarak yazabilirler.
- Öğrenciler tasarladıkları mobil robotun basit geometrik şekilleri takip edecek şekilde hareketler gerçekleştirmesine yönelik Arduino IDE program kodlarını yazabilirler.
- Öğrenciler Bluetooth kablosuz iletişim modülü hakkında temel bilgileri ve çalışma prensibini bilirler.
- Öğrenciler Bluetooth kablosuz iletişim modülü ile Arduino UNO, motor sürücü kartı ve DC motorlar arasındaki fiziksel devre bağlantısını yapabilir ve robot platformu üzerinde montajı gerçekleştirebilir.
- Öğrenciler Bluetooth kablosuz iletişim modülünü kullanarak mobil robotun telefon ya da tablet üzerinden kumanda edilebilmesi için gerekli uygulamayı indirebilir, uygulama üzerinde tuş düzenlemelerini yapabilir ve uygulama üzerinde kullanımı gerçekleştirebilir.
- Öğrenciler Arduino IDE üzerinde mobil robotun temel hareketleri, hızlanma, yavaşlama ve durma gibi kontrolleri gerçekleştirebilmek için gerekli program kodlarını yazabilirler.
- Öğrenciler farklı amaçlar için (örneğin yarışma) mobil robot tasarımını gerçekleştirebilir, farklı tasarım ilkelerini kullanabilir ve ideal robot için fiziksel düzenlemeleri yapabilir.
- Öğrenciler Arduino IDE programı üzerinde mobil robotun temel hareketleri, hızlanma, yavaşlama ve durma gibi kontrolleri farklı amaçlar doğrultusunda değiştirebilir, yeniden düzenleyebilir ve kullanabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle L298N voltaj regülatörlü çift motor sürücü kartı hakkında bilgi sahibi olmaları, robot platformu parçalarını birleştirerek 3 tekerlekli mobil robot yapımı ve mobil robotta temel hareketleri gerçekleştirebilecek Arduino IDE program kodlarını yazabilmektir. Ayrıca Bluetooth kablosuz iletişim modülü hakkında bilgi sahibi olmaları ve modülü geliştirilmiş mobil robot bağlantısını yaparak telefon ya da tablet üzerinden indirilecek uygulama ile kontrol edebilmeleri yine bu haftanın amaçları içerisinde yer almaktadır. Öğrencilerin ders kapsamında düzenlenecek yarış ile ideal mobil robot tasarımı yapmaları ve Arduino IDE üzerinde bu amaca yönelik program kodlarını yazmaları da ders hedefleri arasındadır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	DC motor
Breadboard	Tekerlek
Bağlantı kabloları	9 V pil
Robot platformu	Her yöne dönebilen metal tekerlek
L298N voltaj regülatörlü çift motor sürücü kartı	Mobil robot kontrolü için Android tabanlı telefon veya tablet
Bluetooth kablosuz iletişim modülü	

Haftanın İşlenişi:

Gözle: L298N voltaj regülatörlü çift motor sürücü kartı ve Bluetooth kablosuz iletişim modülü tanımlarının yapılması ve özelliklerinin verilmesi ile mobil robot tasarımının fiziksel olarak geliştirilmesi.

Uygula: Motor sürücü kartı, Bluetooth kablosuz iletişim modülü devre elemanlarının kullanıldığı örnek devrelerin oluşturulması ve programlanması ile 3 tekerlekli mobil robot montajı, tasarımı, geliştirilmesi ve programlanması.

Tasarla: Mobil robot yarışmasının düzenlenmesi.

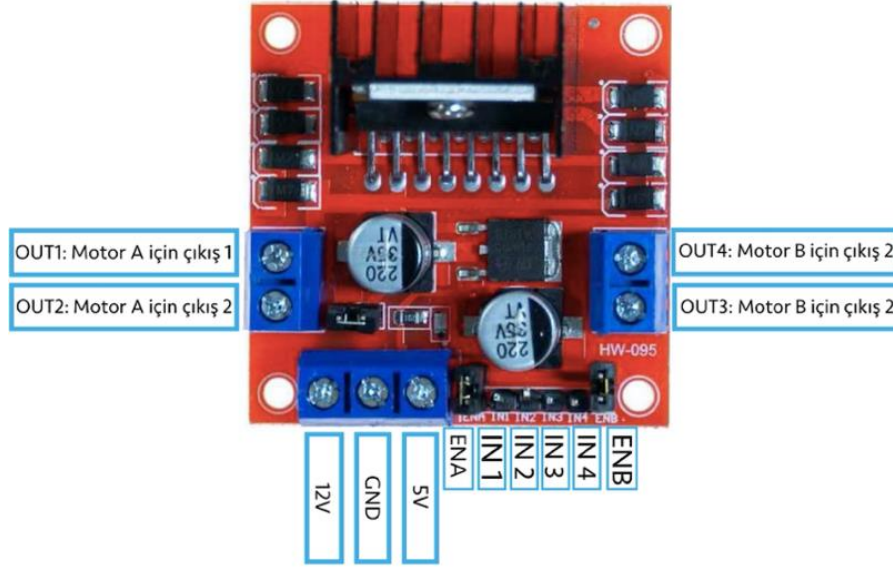
Üret: Mobil robot yarışması için ideal tasarımın ve program kodlarının yazımı ve kullanımı.

Değerlendir: Yansıtma etkinliği.

1. ADIM: GÖZLE ve UYGULA

1.1 Gözle - L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı

“L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı”, fırçalı DC motorları ve step motorları çalıştırmak için kullanılabilen çift H-köprülü motor sürücü kartıdır. Genel olarak motorların hız ve yönlerini kontrol etmek için kullanılan motor sürücü kartı, 2 adet DC motoru aynı anda 2 ampere kadar kontrol edebilme imkanı vermektedir. 24 V’a kadar olan motorları kullanabilme imkanı sağlayan modül; kanal başına 2A akım sağlayabildiğinden 2 motoru ileri veya geri tam güçte ya da istenilen hızda hareket ettirebilir. Aşağıdaki resimde voltaj regülatörlü çift motor sürücü kartı görülmektedir.



Resim 8.1: L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı

Yukarıda görseli verilmiş olan motor sürücü kartı için pin açıklamaları aşağıda verilmektedir. Eğitmen motor sürücü kartı üzerindeki pinleri öğrencilere açıklamalı ve öğrencilerin motor sürücü kartı üzerinde ilgili pin ve işlevleri görmeleri sağlanmalıdır.

L298N Motor Sürücü Kartı Pinleri			
Pin Adı	Görevi	Pin Adı	Görevi
Out 1	Motor A çıkış 1	EnA	Motor A için PWM sinyalini devreye alır
Out 2	Motor A çıkış 2	In1	Motor A çıkış 1 için giriş sinyali
Out 3	Motor B çıkış 1	In2	Motor A çıkış 2 için giriş sinyali
Out 4	Motor B çıkış 2	In3	Motor B çıkış 1 için giriş sinyali
GND	Toprak	In4	Motor B çıkış 2 için giriş sinyali
5 V	5 V enerji girişi	EnB	Motor B için PWM sinyalini devreye alır
12 V	12 V enerji girişi		

Motor sürücü kartı üzerinde sağ ve solda DC motoru bağlamak için “Out 1, Out 2, Out 3 ve Out 4” portları bulunmakta olup “Out 1 ve Out 2” şekilde “Motor A” ile gösterilen bağlantı için bir DC motor ile; “Out 3 ve Out 4” ise “Motor B” ile gösterilen diğer motor ile bağlantıyı sağlar. IN1 ile IN4 pinleri arasındaki pinler Arduino’ya bağlanacak olan giriş pinleridir. Bu pinler “Motor A” ve “Motor B” şeklinde gösterilen iki DC motor için giriş sağlamaktadır. EnA’ya ve EnB’ye PWM sinyali uygulanarak motorların dönüş hızları kontrol edilir. 12 V ile 0 V arasında bir güç verilerek kart beslenebilir. Kart üzerinde bulunan 5 V terminali ise gerektiğinde Arduino’ya veya başka bir modüle güç sağlamak için kullanılabilen çıkıştır.

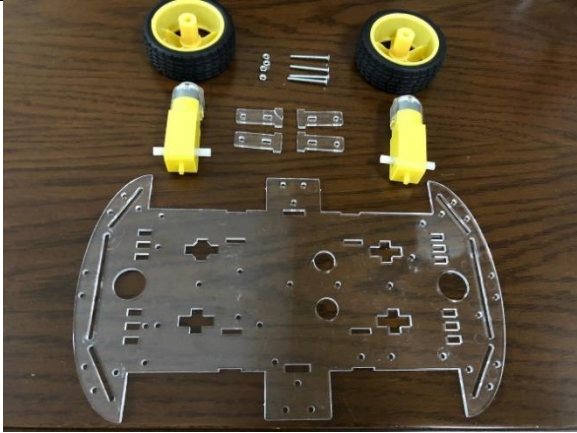
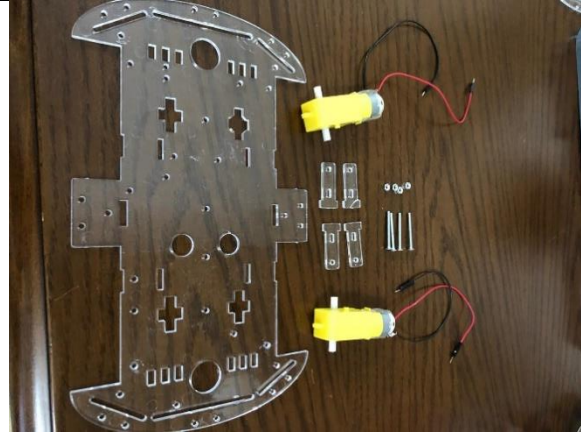
Dikkat

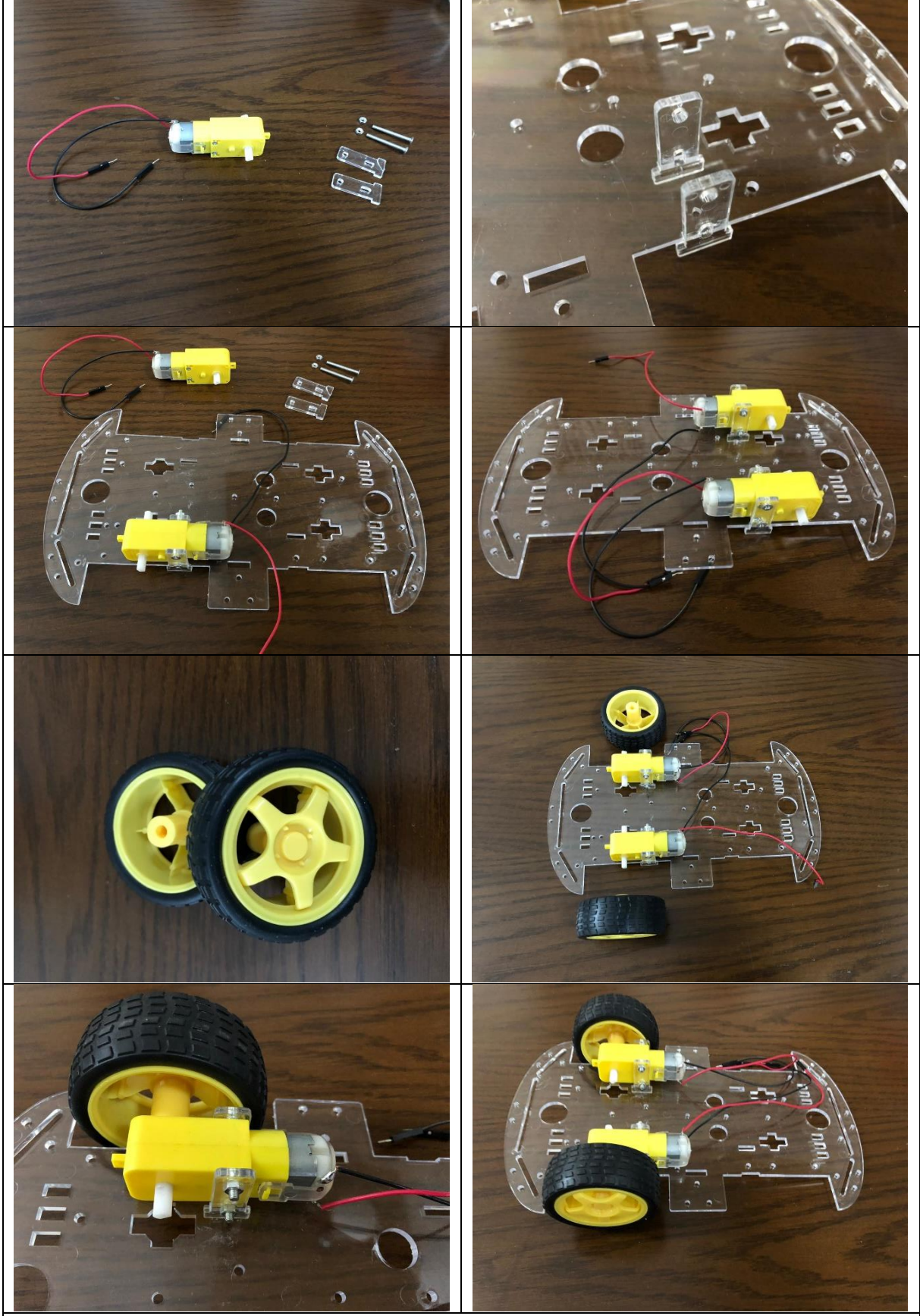
Motor yönü Arduino pinlerinde oluşturulan “HIGH” veya “LOW” sinyalleri ile kontrol edilmektedir. Örneğin, Motor A için In1: HIGH, In2:LOW sinyaliyle motor bir yönde dönerken motorun diğer yönde dönmesi istenirse bu sinyaller In1:LOW ve In2:HIGH şeklinde düzenlenmelidir.

1.2 Gözle ve Uygula – 3 Tekerlekli Mobil Robot Yapımı

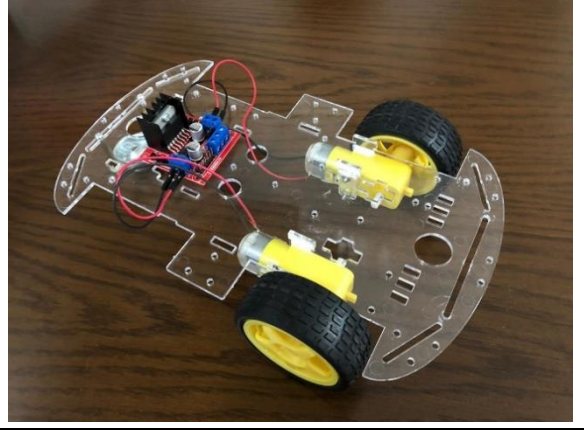
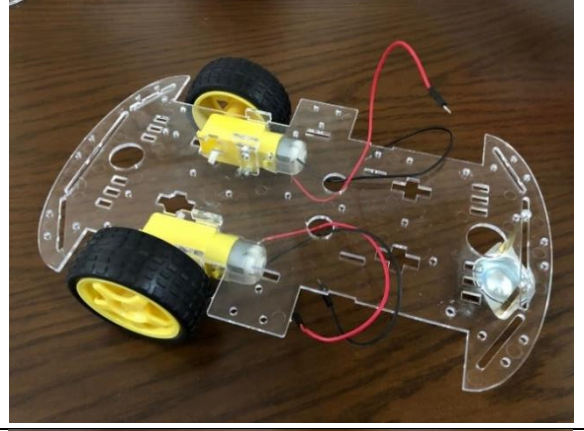
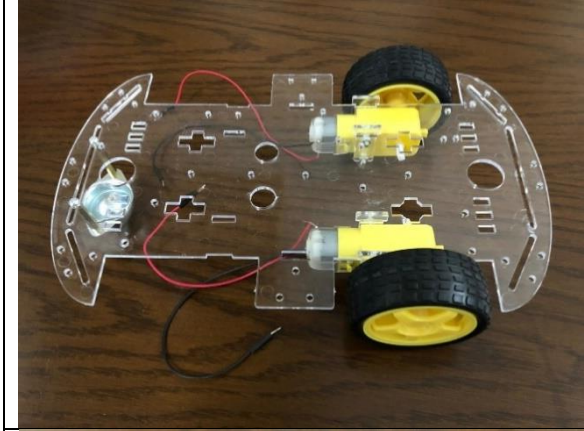
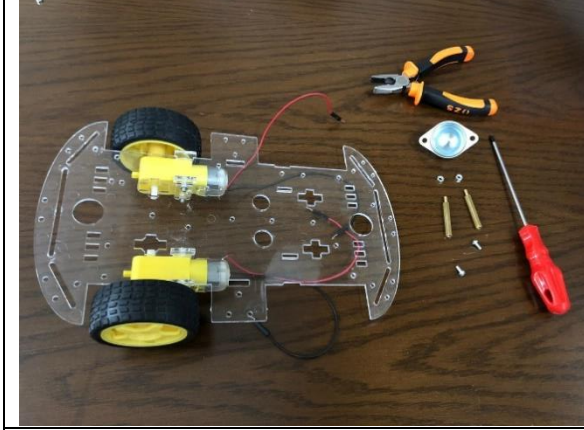
Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Her yöne dönebilen metal tekerlek
Mobil Robot Platformu
Motor Sürücü Kartı

Mobil robotlar, çok amaçlı robot platformları üzerinde elektronik devre elemanları ile birlikte programlanabilen robotik uygulamalarda farklı amaçlarda sıklıkla kullanılır. Mobil robotlar kullanım amaçlarına göre farklı şekilde tasarlanabilirler. Bu derste öncelikle 3 tekerlekli mobil robot tasarımı gerçekleştirilecektir. Bunun için öncelikle setler içerisinde çıkan robot platformunun montajının doğru şekilde gerçekleştirilmesi gereklidir. Bu kapsamda aşağıdaki yapım kılavuzundaki adımlar sıra ile izlenmeli ve mobil robot kullanıma hazır hale getirilmelidir. Eğitimci bu aşamada ilk olarak öğrencilere set içerisinde çıkan malzemeleri tanıtmalı ve sıra ile işlemleri gerçekleştirip, öğrencilerin yapması için gerekli zamanı onlara vermelidir. Yapım kılavuzunda ilgili yerlerde de belirtildiği gibi robot platformundaki bazı malzemelerin (örneğin motorların kablolarla lehimlenmesi işlemi) platforma monte edilmeden önce hazırlanması gerekmektedir. Eğitimci bu işlemler için de öğrencilere gerekli süreyi ayırmalı ve tehlike yaratabilecek süreçlerde öğrenci sağlığını koruyacak şekilde tedbirler almalıdır.

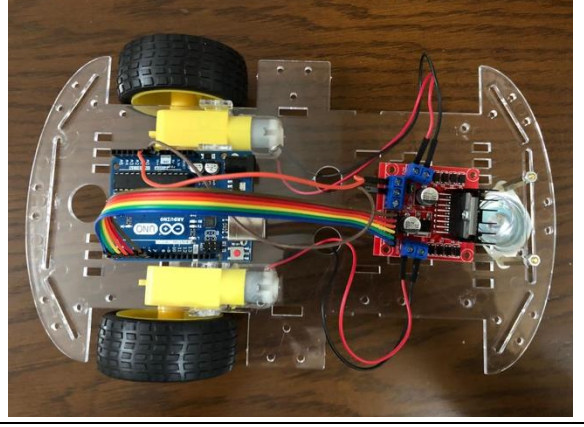
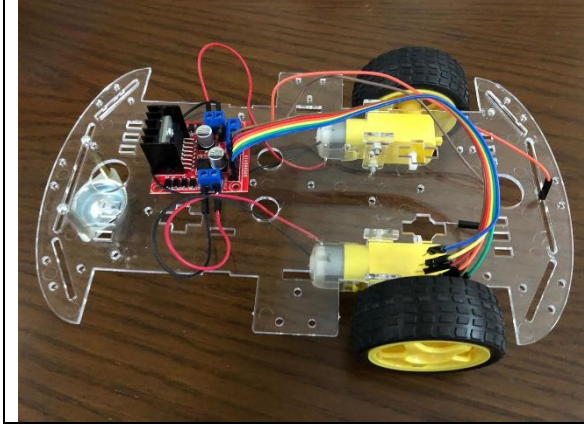
3 Tekerlekli Mobil Robot Yapım Kılavuzu	
Aşağıda resimlerde verilen adımlar sıra ile izlenerek 3 tekerlekli mobil robot montajı yapılmalıdır.	
	
Eğer lehimli değilse bu aşamada motorlara ileride yaşanabilecek karışıklıkları engellemek için tercihen siyah ve kırmızı kablolar şeklindeki gibi lehimlenmelidir.	



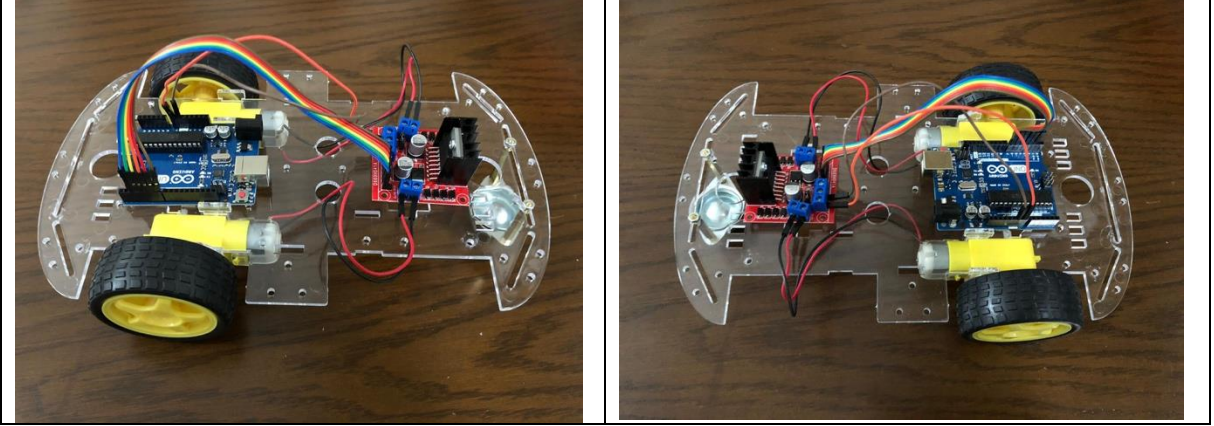
Motorların ve tekerleklerin montajı yapıldıktan sonra kablolar uygun şekilde üstten yukarıya doğru yönlendirilir.



Motor sürücü kartı mobil arabanın üzerine yerleştirilerek gerekli kablo bağlantıları yapılır.

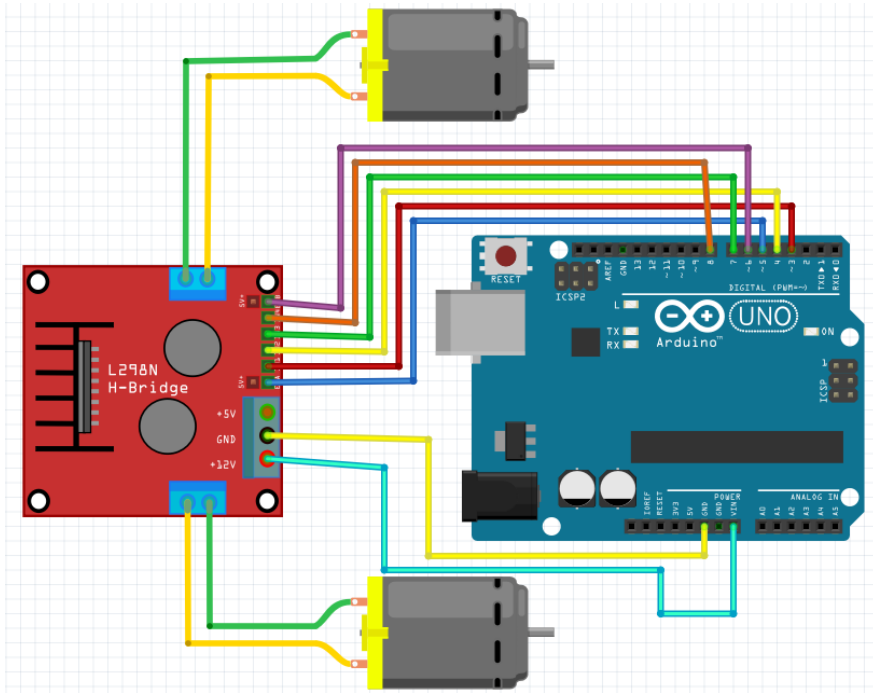


Motor sürücü kartından sonra Arduino Uno bağlantıları gerçekleştirilerek mobil robot harekete hazır hale getirilir.



Resim 8.2: 3 Tekerlekli Mobil Robot Yapım Aşamaları

Mobil robot yapım kılavuzunda belirtilen Arduino Uno, motor sürücü kartı ve tekerlekler arasındaki devre bağlantısı aşağıdaki resimde detaylı olarak gösterilmiştir.



Resim 8.3: Gözle-Uygula Etkinliği Örnek Devre Şeması

Dikkat

Mobil robotun tekerlekleri ve sürücü kartı arasındaki bağlantılarda aynı renkteki kabloların (üstteki devre görselindeki sarı ve yeşil kablolar) çapraz olarak bağlanması yerinde olacaktır. Mobil robotun ileri-geri hareketi sonucunda ters bağlantı yapılmış dahi olsa Arduino IDE içerisindeki pin değerlerinin değiştirilmesi ile her iki motorun ortak şekilde hareketi sağlanmış olacaktır.

Tüm öğrencilerin mobil robotun montajı ve devre bağlantılarını gerçekleştirmesinden sonra eğitimci mobil robotun belirli sürelerde ve sabit 100 birim hızında temel ileri-geri-sağ-sol hareketleri yapması için Arduino IDE’de yazılmış olan aşağıdaki resimde verilen örnek programı öğrencilere göstererek kodları açıklar. Program kodlarının açıklanmasında mobil robot için tüm temel hareketlerin ayrı birer fonksiyon olarak yazıldığı vurgulanmalıdır. Özellikle temel hareket fonksiyonları içerisinde durdurulması istenen motorların hızlarının “0” yapılarak işlemin gerçekleştirildiği üzerinde durulmalıdır. Eğitimci tüm öğrencilerin tasarladıkları mobil robotların temel hareketleri doğru şekilde gerçekleştirmesi için yeterli süreyi vermelidir.

```

iki_tekerli_araba_temel_hareketler
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 150;
8
9 void setup() {
10  pinMode(MotorR1,OUTPUT);
11  pinMode(MotorR2,OUTPUT);
12  pinMode(MotorL1,OUTPUT);
13  pinMode(MotorL2,OUTPUT);
14 }
15
16 void loop() {
17  ileri();
18  delay(2000);
19  dur();
20  delay(2000);
21  geri();
22  delay(1500);
23  dur();
24  delay(2000);
25  sag();
26  delay(1000);
27  dur();
28  delay(1000);
29  ileri();
30  delay(2500);
31  dur();
32  delay(1000);
33  sol();
34  delay(1000);
35  dur();
36  delay(1000);
37 }
38
39 void ileri() {
40  digitalWrite(MotorR1,HIGH);
41  digitalWrite(MotorR2,LOW);
42  analogWrite(MotorRE,hiz);
43  digitalWrite(MotorL1,HIGH);
44  digitalWrite(MotorL2,LOW);
45  analogWrite(MotorLE,hiz);
46 }
47
48 void sag() {
49  digitalWrite(MotorR1,HIGH);
50  digitalWrite(MotorR2,LOW);
51  analogWrite(MotorRE,0);
52  digitalWrite(MotorL1,HIGH);
53  digitalWrite(MotorL2,LOW);
54  analogWrite(MotorLE,hiz);
55 }
56
57 void sol() {
58  digitalWrite(MotorR1,HIGH);
59  digitalWrite(MotorR2,LOW);
60  analogWrite(MotorRE,hiz);
61  digitalWrite(MotorL1,HIGH);
62  digitalWrite(MotorL2,LOW);
63  analogWrite(MotorLE,0);
64 }
65
66 void dur() {
67  digitalWrite(MotorR1,HIGH);
68  digitalWrite(MotorR2,LOW);
69  analogWrite(MotorRE,0);
70  digitalWrite(MotorL1,HIGH);
71  digitalWrite(MotorL2,LOW);
72  analogWrite(MotorLE,0);
73 }
74
75 void geri() {
76  digitalWrite(MotorR1,LOW);
77  digitalWrite(MotorR2,HIGH);
78  analogWrite(MotorRE,hiz);
79  digitalWrite(MotorL1,LOW);
80  digitalWrite(MotorL2,HIGH);
81  analogWrite(MotorLE,hiz);
82 }
Done Saving

```

Resim 8.4: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Eğitimci Arduino IDE içerisinde mobil robotun temel hareketlerinin yazılı olduğu programın kodlarını öğrencilere gösterirken fonksiyonların belirli bir görevi gerçekleştirmek için tasarlanmış kod blokları olduğu ve herhangi bir eylem sonucunda

çağrıldıklarında yürütülmeye başladıkları vurgulanmalıdır. Örnek uygulamada da loop fonksiyonu içerisinde temel hareketleri gerçekleştirecek her bir fonksiyonun istenilen zamanda çağrılarak çalıştırıldığı bilgisi öğrenciler ile paylaşılmalıdır.

1.3 Uygula – Mobil Robotu Kare Şeklinde Hareket Ettirme (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Mobil Robot
Motor Sürücü Kartı

Bu etkinlikteki amaç öğrencilerin tasarladıkları mobil robota sabit 100 birim hızında kare şeklinde bir rotayı çizdirmeleridir. Eğitimci aşağıdaki resimde Arduino IDE içerisinde yazılmış örnek kodu verilen programı öğrencilerden yazmasını isteyecek ve tüm öğrencilerin mobil robotunun kare şeklinde bir rotayı izleyecek şekilde programlanması için yeterli zamanı verecektir. Bu kapsamda eğitimci öğrencilerin örnek kod üzerindeki bekleme sürelerinde, mobil robotun tekerlekler ile zemin arasındaki sürtünme katsayılarını ve motorların dönme hızlarını dikkate almalarını hatırlatmalı ve öğrencilerin Arduino IDE yazılımında kendi kod satırlarında bu değişkenlere göre düzenleme yapabileceklerini söylemelidir. Sonuç olarak öğrencilerin mobil robotları kendi kod satırlarındaki düzenlemelerden sonra düzgün bir kare şekli çizmelidir.

```

iki_tekerli_araba_kare §
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 100;
8
9 void setup() {
10  pinMode(MotorR1,OUTPUT);
11  pinMode(MotorR2,OUTPUT);
12  pinMode(MotorL1,OUTPUT);
13  pinMode(MotorL2,OUTPUT);
14 }
15
16 void loop() {
17  for (int i=0; i<4; i++){
18    ileri();
19    delay(1500);
20    dur();
21    delay(1500);
22    sag();
23    delay(1000);
24    dur();
25    delay(1500);
26  }
27 }
28
29 void ileri(){
30  digitalWrite(MotorR1,HIGH);
31  digitalWrite(MotorR2,LOW);
32  analogWrite(MotorRE,hiz);
33  digitalWrite(MotorL1,HIGH);
34  digitalWrite(MotorL2,LOW);
35  analogWrite(MotorLE,hiz);
36 }
37
38 void sag(){
39  digitalWrite(MotorR1,HIGH);
40  digitalWrite(MotorR2,LOW);
41  analogWrite(MotorRE,0);
42  digitalWrite(MotorL1,HIGH);
43  digitalWrite(MotorL2,LOW);
44  analogWrite(MotorLE,hiz);
45 }
46
47 void sol(){
48  digitalWrite(MotorR1,HIGH);
49  digitalWrite(MotorR2,LOW);
50  analogWrite(MotorRE,hiz);
51  digitalWrite(MotorL1,HIGH);
52  digitalWrite(MotorL2,LOW);
53  analogWrite(MotorLE,0);
54 }
55
56 void dur(){
57  digitalWrite(MotorR1,HIGH);
58  digitalWrite(MotorR2,LOW);
59  analogWrite(MotorRE,0);
60  digitalWrite(MotorL1,HIGH);
61  digitalWrite(MotorL2,LOW);
62  analogWrite(MotorLE,0);
63 }
64
65 void geri(){
66  digitalWrite(MotorR1,LOW);
67  digitalWrite(MotorR2,HIGH);
68  analogWrite(MotorRE,hiz);
69
70  digitalWrite(MotorL1,LOW);
71  digitalWrite(MotorL2,HIGH);
72  analogWrite(MotorLE,hiz);
73 }
74

```

Resim 8.5: Gözle-Uygula Etkinliği Örnek Kodu

1.4 Uygula- Mobil Robotu Farklı Geometrik Şekillerde Hareket Ettirme (Öğrenci 2)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Mobil Robot
Motor Sürücü Kartı

Bu etkinlik için eğitmen öğrencilere aşağıdaki dört maddeden bir tanesini atayarak maddede yazılı görevi yerine getirecek mobil robotun program kodunu yazmalarını isteyecektir.

- Kenar uzunluğu 40 cm olan bir eşkenar üçgen şeklinde hareket et, her köşede 1'er saniye bekle ve şekil tamamlandığında 5 saniye bekle.
- Kenar uzunluğu 40 cm olan bir beşgen şeklinde hareket et, her köşede 1'er saniye bekle ve şekil tamamlandığında 5 saniye bekle.
- Kenar uzunluğu 40 cm olan bir altıgen şeklinde hareket et, her köşede 1'er saniye bekle ve şekil tamamlandığında 5 saniye bekle.
- Uzun kenarı 80 cm, kısa kenarı 40 cm olan bir dikdörtgen şeklinde hareket et, her köşede 1'er saniye bekle ve şekil tamamlandığında 5 saniye bekle.

Dikkat

Yukarıdaki geometrik şekillere göre hareket edecek mobil robotların Arduino IDE program kodları yazılırken, istenen en uygun ölçekli şekli takip edebilmesi için eğitmen öğrencilere;

- Motorların dönme sürelerini ve hızlarını,
- Tekerleklerin çevre uzunluklarını,
- Zemin ile tekerlekler arasındaki sürtünme derecesini,
- Zemindeki diğer faktörleri (eğim, engebe vb.)

dikkate alarak program kodlarında tekrar tekrar değişiklikler yapmaları için uyarılarda bulunur.

Etkinlik için örnek kod aşağıdaki resimlerde görülmektedir.

```
iki_tekerli_araba_ucgen
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 100;
8
9 void setup() {
10   pinMode(MotorR1, OUTPUT);
11   pinMode(MotorR2, OUTPUT);
12   pinMode(MotorL1, OUTPUT);
13   pinMode(MotorL2, OUTPUT);
14 }
15
16 void loop() {
17   for (int i=0; i<3; i++){
18     ileri();
19     delay(2000);
20     dur();
21     delay(1000);
22     sag();
23     delay(1250);
24     dur();
25     delay(1000);
26   }
27   dur();
28   delay(5000);
29 }
30
31 void ileri() {
32   digitalWrite(MotorR1, HIGH);
33   digitalWrite(MotorR2, LOW);
34   analogWrite(MotorRE, hiz);
35   digitalWrite(MotorL1, HIGH);
36   digitalWrite(MotorL2, LOW);
37   analogWrite(MotorLE, hiz);
38 }
39
40 void sag() {
41   digitalWrite(MotorR1, HIGH);
42   digitalWrite(MotorR2, LOW);
43   analogWrite(MotorRE, 0);
44   digitalWrite(MotorL1, HIGH);
45   digitalWrite(MotorL2, LOW);
46   analogWrite(MotorLE, hiz);
47 }
48
49 void dur() {
50   digitalWrite(MotorR1, HIGH);
51   digitalWrite(MotorR2, LOW);
52   analogWrite(MotorRE, 0);
53   digitalWrite(MotorL1, HIGH);
54   digitalWrite(MotorL2, LOW);
55   analogWrite(MotorLE, 0);
56 }
57
Done Saving.
```

Resim 8.6: Gözle-Uygula Etkinliği Örnek Kodu

8.Bölüm: Bluetooth Kontrollü 3 Tekerli Mobil Robot

```
iki_tekerli_araba_besgen
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 100;
8
9 void setup() {
10  pinMode(MotorR1,OUTPUT);
11  pinMode(MotorR2,OUTPUT);
12  pinMode(MotorL1,OUTPUT);
13  pinMode(MotorL2,OUTPUT);
14 }
15
16 void loop() {
17  for (int i=0; i<5; i++){
18    ileri();
19    delay(2000);
20    dur();
21    delay(1000);
22    sag();
23    delay(700);
24    dur();
25    delay(1000);
26  }
27  dur();
28  delay(5000);
29 }
30
31 void ileri() {
32  digitalWrite(MotorR1,HIGH);
33  digitalWrite(MotorR2,LOW);
34  analogWrite(MotorRE,hiz);
35  digitalWrite(MotorL1,HIGH);
36  digitalWrite(MotorL2,LOW);
37  analogWrite(MotorLE,hiz);
38 }
39
40 void sag() {
41  digitalWrite(MotorR1,HIGH);
42  digitalWrite(MotorR2,LOW);
43  analogWrite(MotorRE,0);
44  digitalWrite(MotorL1,HIGH);
45  digitalWrite(MotorL2,LOW);
46  analogWrite(MotorLE,hiz);
47 }
48
49 void dur() {
50  digitalWrite(MotorR1,HIGH);
51  digitalWrite(MotorR2,LOW);
52  analogWrite(MotorRE,0);
53  digitalWrite(MotorL1,HIGH);
54  digitalWrite(MotorL2,LOW);
55  analogWrite(MotorLE,0);
56 }
57
Done Saving.
```

Resim 8.7: Gözle-Uygula Etkinliği Örnek Kodu

```
iki_tekerli_araba_altigen
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 100;
8
9 void setup() {
10  pinMode(MotorR1,OUTPUT);
11  pinMode(MotorR2,OUTPUT);
12  pinMode(MotorL1,OUTPUT);
13  pinMode(MotorL2,OUTPUT);
14 }
15
16 void loop() {
17  for (int i=0; i<6; i++){
18    ileri();
19    delay(2000);
20    dur();
21    delay(1000);
22    sag();
23    delay(600);
24    dur();
25    delay(1000);
26  }
27  dur();
28  delay(5000);
29 }
30
31 void ileri() {
32  digitalWrite(MotorR1,HIGH);
33  digitalWrite(MotorR2,LOW);
34  analogWrite(MotorRE,hiz);
35  digitalWrite(MotorL1,HIGH);
36  digitalWrite(MotorL2,LOW);
37  analogWrite(MotorLE,hiz);
38 }
39
40 void sag() {
41  digitalWrite(MotorR1,HIGH);
42  digitalWrite(MotorR2,LOW);
43  analogWrite(MotorRE,0);
44  digitalWrite(MotorL1,HIGH);
45  digitalWrite(MotorL2,LOW);
46  analogWrite(MotorLE,hiz);
47 }
48
49 void dur() {
50  digitalWrite(MotorR1,HIGH);
51  digitalWrite(MotorR2,LOW);
52  analogWrite(MotorRE,0);
53  digitalWrite(MotorL1,HIGH);
54  digitalWrite(MotorL2,LOW);
55  analogWrite(MotorLE,0);
56 }
57
Done Saving.
```

Resim 8.8: Gözle-Uygula Etkinliği Örnek Kodu


```

iki_tekerli_araba_dikdortgen
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7 int hiz = 100;
8
9 void setup() {
10 pinMode(MotorR1, OUTPUT);
11 pinMode(MotorR2, OUTPUT);
12 pinMode(MotorL1, OUTPUT);
13 pinMode(MotorL2, OUTPUT);
14 }
15
16 void loop() {
17 for (int i=0; i<2; i++){
18   ileri();
19   delay(2000);
20   dur();
21   delay(1000);
22   sag();
23   delay(1000);
24   dur();
25   delay(1000);
26   ileri();
27   delay(4000);
28   dur();
29   delay(1000);
30   sag();
31   delay(1000);
32   dur();
33   delay(1000);
34 }
35 dur();
36 delay(5000);
37 }
38
39 void ileri(){
40   digitalWrite(MotorR1,HIGH);
41   digitalWrite(MotorR2,LOW);
42   analogWrite(MotorRE,hiz);
43   digitalWrite(MotorL1,HIGH);
44   digitalWrite(MotorL2,LOW);
45   analogWrite(MotorLE,hiz);
46 }
47
48 void sag(){
49   digitalWrite(MotorR1,HIGH);
50   digitalWrite(MotorR2,LOW);
51   analogWrite(MotorRE,0);
52   digitalWrite(MotorL1,HIGH);
53   digitalWrite(MotorL2,LOW);
54   analogWrite(MotorLE,hiz);
55 }
56
57 void dur(){
58   digitalWrite(MotorR1,HIGH);
59   digitalWrite(MotorR2,LOW);
60   analogWrite(MotorRE,0);
61   digitalWrite(MotorL1,HIGH);
62   digitalWrite(MotorL2,LOW);
63   analogWrite(MotorLE,0);
64 }
65
Done uploading.

```

Resim 8.9: Gözle-Uygula Etkinliği Örnek Kodu

1.5 Uygula- Mobil Robotu Daire Şeklinde Hareket Ettirme (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Mobil Robot
Motor Sürücü Kartı

Bu etkinlikte eğitmen öncelikle öğrenci gruplarına iki motorun dönme hızlarının birbirinden farklı ayarlanarak mobil robotun dairesel hareketler çizebileceğini açıklar. Bunu da yukarıda çalışılmış olan Arduino IDE program kodlarında yer alan “MotorLE” ve “MotorRE” değişkenlerinin değerlerini birbirinden farklı yazarak oluşturabileceklerini anlatır ve öğrencilerden mobil robotun yarıçapı 40 cm olan bir çember çizerek hareket etmesini sağlayan programın kodlarını yazmalarını ister. Etkinlikle ilgili örnek kod aşağıdaki resimde görülmektedir.

Dikkat

Mobil robotun dairesel hareketlerde Arduino IDE program kodları yazılırken, istenen en uygun ölçekli şekli takip edebilmesi için eğitmen öğrencilere;

- Motorların dönme sürelerini ve hızlarını,
- Tekerleklerin çevre uzunluklarını,
- Zemin ile tekerlekler arasındaki sürtünme derecesini,
- Zemindeki diğer faktörleri (eğim, engebe vb.)

dikkate alarak program kodlarında tekrar tekrar değişiklikler yapmaları için uyarılarda bulunur.

```

iki_tekerli_araba_daire
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4 #define MotorL1 8
5 #define MotorL2 7
6 #define MotorLE 6
7
8 void setup() {
9   pinMode(MotorR1, OUTPUT);
10  pinMode(MotorR2, OUTPUT);
11  pinMode(MotorL1, OUTPUT);
12  pinMode(MotorL2, OUTPUT);
13 }
14
15 void loop() {
16   digitalWrite(MotorR1, HIGH);
17   digitalWrite(MotorR2, LOW);
18   analogWrite(MotorRE, 125);
19   digitalWrite(MotorL1, HIGH);
20   digitalWrite(MotorL2, LOW);
21   analogWrite(MotorLE, 75);
22 }
23
Done Saving.

```

Resim 8.10: Gözle-Uygula Etkinliği Örnek Kodu

1.6 Gözle ve Uygula – Bluetooth Kablosuz İletişim Modülü

Bluetooth kablosuz iletişim modülleri, Bluetooth SPP (Serial Port Profile) kullanımı ve kablosuz seri haberleşme uygulamaları için tasarlanmıştır. Arduino kartlar ile uzaktan yönetilebilir uygulamalar gerçekleştirebilmek için kullanılır. Aşağıdaki resimlerde de görülebileceği üzere hızlı prototiplemeye imkan sağlaması, breadboard, arduino ve çeşitli devrelerde rahatça kullanılabilmesi için gerekli pinler devre kartı sayesinde dışarıya alınmıştır.

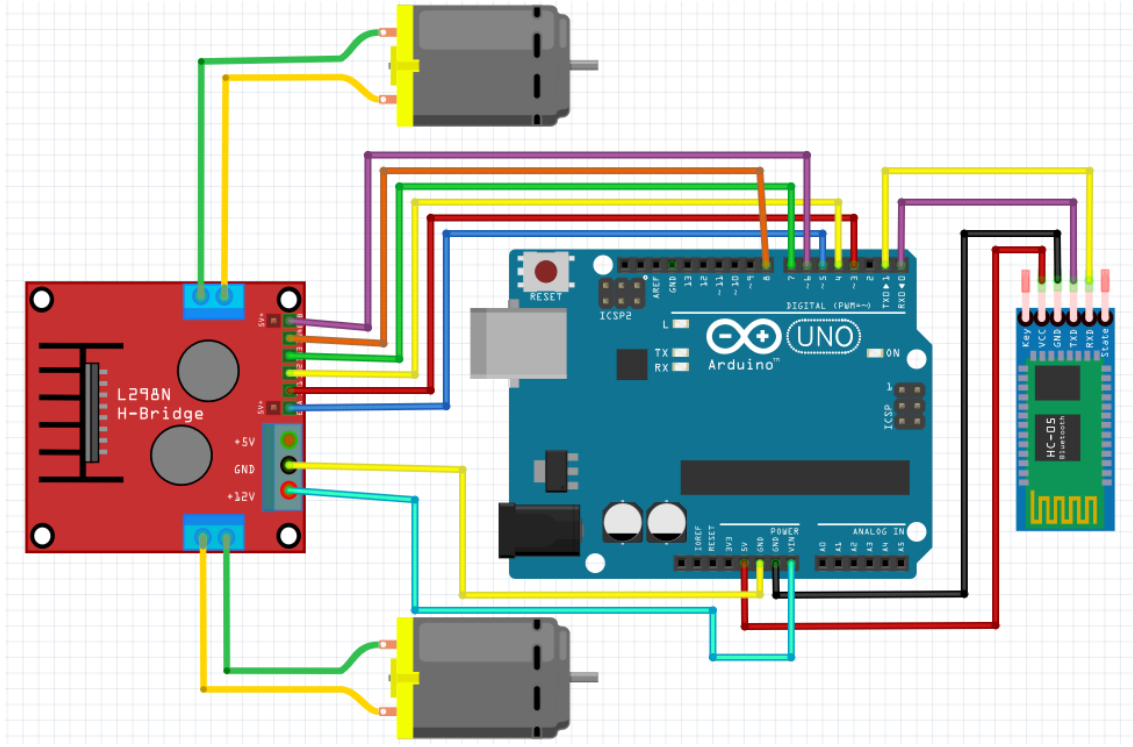


Resim 8.11: Bluetooth Modülü

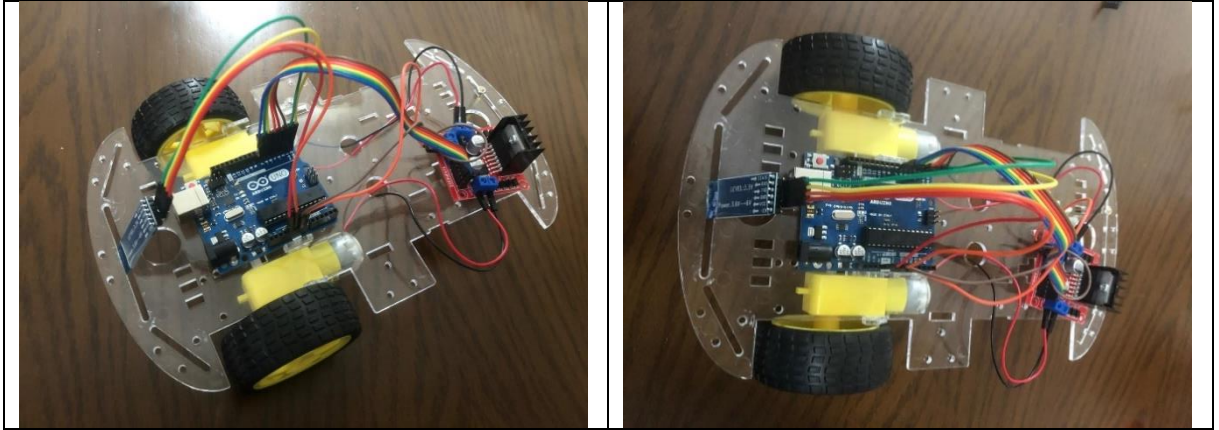
Arduino projelerinde genellikle “**HC-05**” veya “**HC-06**” Bluetooth modülleri kullanılır. Bu modüllerin genel olarak özellikleri aynı olmasına rağmen “HC-05” kartı kendisine yönlendirilen kablosuz bağlantı isteklerine cevap verebilme özelliğine sahipken aynı zamanda diğer kablosuz cihazlara bağlantı isteği gönderebilir. “HC-06” ise sadece kendisine gönderilen isteklere cevap verebilir. Daha detaylı açıklanacak olursa “HC-05” hem master (yönetici) hem de slave (köle) özelliğinde isteklere cevap verebilirken HC-06 sadece slave (köle) olarak işlem görür. Bluetooth 2.0’ı destekleyen modüller 2.4 GHz frekansında haberleşme yapılmasını sağlar ve yaklaşık 10 metre haberleşme menziline sahiptir.

Bu ders kapsamında gerçekleştirilecek uygulamalarda Bluetooth modülü üzerinde bulunan **RXD**, **TXD**, **GND** ve **VCC** giriş/çıkışları kullanılacaktır. Genel olarak Bluetooth modülü Arduino UNO’nun haberleşme pinleri olan TX ve RX üzerinden haberleşme sağlamaktadır. Burada dikkat edilecek olan nokta uygulamalarda Arduino UNO üzerinde TX pini ile Bluetooth modülü üzerindeki RXD; RX pini ile de TXD girişinin bağlanmasıdır. Diğer bir ifadeyle veri iletişimde alıcı ve verici pinleri karşılıklı olarak birbirine **çapraz** bağlanmalıdır. Bluetooth modülü üzerindeki VCC girişi Arduino UNO üzerinde 5 V ile GND ise Arduino UNO üzerindeki GND pinlerinden biriyle bağlanmalıdır.

Temel bilgilerden sonra eğitmen tasarlanan mobil robot üzerine Bluetooth modülünün bağlantı devresini ve aşağıdaki resimlerde görüldüğü gibi robot üzerine yerleştirilmesini öğrencilere gösterir. Öğrencilerden de kendi mobil robotları üzerinde bu işlemleri gerçekleştirmeleri için yeterli süreyi verir.



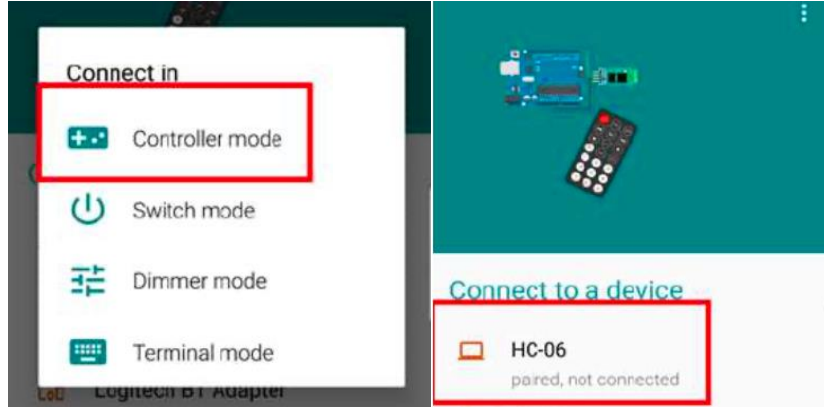
Resim 8.12: Gözle-Uygula Etkinliği Örnek Devre Şeması



Resim 8.13: Gözle-Uygula Etkinliği Örnek Devre Kurulumu

Gerekli bağlantıların yapılmasından sonra mobil robotun telefon ya da tablet üzerinden kumanda edilebilmesi için Arduino Bluetooth kontrol uygulamalarından birinin indirilmesi gerekecektir. Daha detaylı şekilde ifade edilecek olursa; telefon ya da tablet üzerinden belirli tuşlar ile mobil robotun temel hareketleri (ileri, geri, sağa dön, sola dön, hızlanma, hız azaltma ve durma) tanımlanacaktır. Bu amaç doğrultusunda bu tanımlamanın yapılması için etkinlikte “**Arduino Bluetooth Controller**” isimli uygulama kullanılacaktır. Telefon ya da tabletlere bu uygulama indirilmeli ve Bluetooth modülü ile eşleştirme işleminin yapılması gerekmektedir. Program kodunun bilgisayara yüklenmesinden sonra telefon ya da tabletin Bluetooth ayarlarına girilip “HC-06” ya da “HC-05” isimli Bluetooth modülü ile bağlantı gerçekleştirilmelidir. Bu aşamada şifre istenmesi durumunda “0000” ya da “1234” standart şifrelerden birinin girilmesi

gerekecektir. Şifre girildikten sonra “HC-06” ya da “HC-05” seçildikten sonra aşağıdaki resimde görüldüğü gibi “Controller Mode” bölümüne tıklanmalıdır.



Resim 8.14: Arduino Bluetooth Controller Bağlantısı

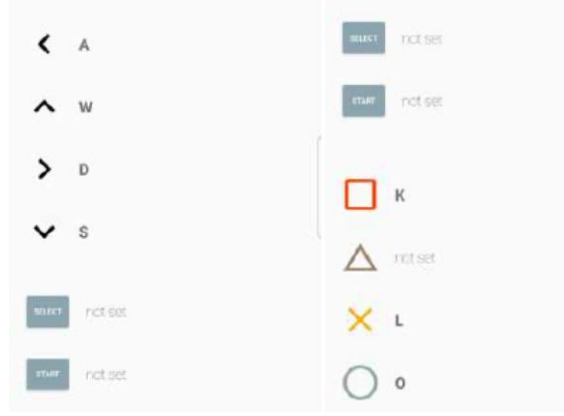
“Controller Mode” bölümüne girildikten sonra aşağıdaki resimde de görüldüğü gibi ekranın sağ üst köşesinde yer alan “Ayarlar” simgesine tıklanarak mobil robotun temel hareketleri, hızlanma, hız azaltma, sağa dönme, sola dönme ve durma gibi kontrolleri hangi tuşlar ile yapılacağı tanımlanmalıdır.



Resim 8.15: Arduino Bluetooth Controller Arayüzü Ayarları

Bu kapsamda aşağıdaki resimde de gösterildiği gibi;

- Mobil robota sola dönüş hareketini yaptırmak üzere “A”;
- Mobil robota ileriye gitme hareketini yaptırmak üzere “W”;
- Mobil robota sağa dönüş hareketini yaptırmak üzere “D”;
- Mobil robota geri gitme hareketini yaptırmak üzere “S”;
- Mobil robotun hızını arttırmak üzere “K”;
- Mobil robotun hızını azaltmak üzere “L”;
- Mobil robotu durdurmak üzere “O” tuşları tanımlanmalıdır.



Resim 8.16: Arduino Bluetooth Controller Kısayol Atamaları

Mobil robotun Bluetooth kablosuz iletişim modülü ile kumanda edilmesi için aşağıdaki resimde Arduino IDE içerisinde yazılmış örnek program kodu verilmiştir.

```

bluetooth_kontrollu_arac
1 #define MotorR1 3
2 #define MotorR2 4
3 #define MotorRE 5
4
5 #define MotorL1 8
6 #define MotorL2 7
7 #define MotorLE 6
8
9 int hiz = 100;
10 char veri;
11
12 void setup() {
13   pinMode(MotorR1, OUTPUT);
14   pinMode(MotorR2, OUTPUT);
15   pinMode(MotorRE, OUTPUT);
16   pinMode(MotorL1, OUTPUT);
17   pinMode(MotorL2, OUTPUT);
18   pinMode(MotorLE, OUTPUT);
19   Serial.begin(9600);
20 }
21
22 void loop() {
23
24   while(Serial.available()){
25     veri = Serial.read();
26
27     switch(veri) {
28       case 'W': //ileri
29       {
30         digitalWrite(MotorR1, HIGH);
31         digitalWrite(MotorR2, LOW);
32         analogWrite(MotorRE, hiz);
33
34         digitalWrite(MotorL1, HIGH);
35         digitalWrite(MotorL2, LOW);
36         analogWrite(MotorLE, hiz);
37         break;
38       }
39       case 'S': //geri
40       {
41         digitalWrite(MotorR1, LOW);
42         digitalWrite(MotorR2, HIGH);
43         analogWrite(MotorRE, hiz);
44
45         digitalWrite(MotorL1, LOW);
46         digitalWrite(MotorL2, HIGH);
47         analogWrite(MotorLE, hiz);
48         break;
49       }
50       case 'A': //Sol
51       {
52         digitalWrite(MotorR1, HIGH);
53         digitalWrite(MotorR2, LOW);
54         analogWrite(MotorRE, hiz);
55
56         digitalWrite(MotorL1, LOW);
57         digitalWrite(MotorL2, LOW);
58         analogWrite(MotorLE, hiz);
59         break;
60       }
61       case 'D': //Sag
62       {
63         digitalWrite(MotorR1, LOW);
64         digitalWrite(MotorR2, LOW);
65         analogWrite(MotorRE, hiz);
66
67         digitalWrite(MotorL1, HIGH);
68         digitalWrite(MotorL2, LOW);
69         analogWrite(MotorLE, hiz);
70         break;
71       }
72       case 'K': //Hiz Artirma
73       {
74         if(hiz<255) {
75           hiz=hiz+5;
76           break;
77         }
78       }
79       case 'L': //Hiz Azaltma
80       {
81         if(hiz>0) {
82           hiz=hiz-5;
83           break;
84         }
85       }
86       case 'O': //Durma
87       {
88         digitalWrite(MotorR1, LOW);
89         digitalWrite(MotorR2, LOW);
90         analogWrite(MotorRE, hiz);
91
92         digitalWrite(MotorL1, LOW);
93         digitalWrite(MotorL2, LOW);
94         analogWrite(MotorLE, hiz);
95         break;
96       }
97     }
98   }
99 }
100
Done compiling

```

Resim 8.17: Gözle-Uygula Etkinliği Örnek Kodu

Eğitmen Arduino IDE içerisinde yazılmış örnek kodu yansıtarak tüm öğrencilere göstermeli ve kod içeriklerini aktarmalıdır. Program kodu içerisinde yer alan “**while**” döngüsü içerisinde yer alan veri değişkeninin sinyal alındığı sürece programı çalıştıracakları üzerinde durulmalıdır. Ayrıca öğrencilere **switch – case** yapısının kullanım durumu da hatırlatılmalıdır. Daha sonra eğitmen tüm öğrencilere tasarlamış oldukları mobil robotları Bluetooth kablosuz iletişim modülü ile kumanda edebilmeleri için yeterli zamanı vermelidir.

2. ADIM: TASARLA ve ÜRET

2.1 Mobil Robot Yarışı

Robotik yarışmalar lise öğrencilerinin mezuniyetlerinden önce günümüzün küresel düzeydeki zorlu görevleri için çözüm stratejilerini geliştirme, birlikte çalışma ve öğrenme, yaratıcı fikir tasarlama, üretme ve uygulama gibi becerileri tecrübe etmelerini sağlayarak onları cesaretlendiren etkinliklerdir. Bu etkinlikte amaç öğrencilerin ders kapsamında geliştirdikleri mobil robotlar ile düzenlenecek yarışma için ideal robotu tasarlamak ve Arduino IDE içerisinde yazılacak program kodlarıyla kumanda düzenlemelerini yarışta kullanabilmeleridir.

Eğitmen öncelikle mobil robot pistini oluşturmalı ve yarışma kurallarını öğrencilere aktarmalıdır. Pist ve yarışma kuralları şu şekilde tasarlanabilir:

- Yarışma için sınıf içerisinde ya da derslik koridorunda mobil robotların yarışmaya başlayacağı bir *başlangıç ve bitiş çizgisi* belirlenmelidir.
- Pist üzerinde mobil robotların manevra yapabilmelerini sağlayacak şekilde *engeller* (çöp kovaları, kutular, çiçek saksıları vs.) yer almalıdır. Öğrencilerin mobil robot tasarımları çerçevesinde engellerin zorluk dereceleri arttırılabilir veya azaltılabilir.
- Yarışmaya hazırlanmaları ve pist üzerinde deneme sürüşleri yapmaları için öğrenci gruplarına yeterli süre verilmelidir. Bu aşamada öğrencilerin mobil robot tasarımlarında ve program kodlarında değişiklikler yapmalarına izin verilmelidir.
- Yarışma esnasında mobil robotların birbirlerini engelleme hakları vardır. Bu süreçte centilmenlik kurallarına uymaları gerektiği öğrencilere hatırlatılmalıdır.
- Ders süresi kapsamında tüm grupların yarışıp dereceye girenlerin belirlenebileceği tek bir yarış düzenlenebileceği gibi ilk yarışta ilk üçe giren grupların mobil robotlar ve program kodları üzerinde son bir değişiklik yapmalarına izin verilip ikinci ve son bir yarış da düzenlenebilir.
- Pisti ilk olarak tamamlayan gruplar bitirme derecelerine göre sıralanır.

2.2 Tasarla - Mobil Robot Yarışı

Öğrencilerin yarışma için fiziksel olarak mobil robot tasarımlarında ve Arduino IDE içerisinde program kodlarını yazmadan önce “tanımlama” ve “fikir üretme” aşamalarını gerçekleştirmeleri gerekmektedir. Bu iki süreç yarış kazanmak için öğrenci gruplarının mobil robot tasarımında ve program kodlarında ne gibi değişiklikler yapmaları gerektiğine yönelik yaratıcı çözümler getirmelerini sağlayacaktır.

Tanımlama süreci: Öğrencilerin öncelikle ders kapsamında geliştirdikleri mobil robotun temel hareketleri gözleyerek yarış pistine uygun şekilde tasarımda ne gibi değişiklikler yapmaları gerektiğini belirlemeleri gerekmektedir. Bu aşamada daha dayanıklı, hızlı ve yarışma için ideal bir robot olması için eklenecek, çıkarılacak veya yerleri değiştirilecek parçaları ortaya koymaları beklenmektedir. Burada öğrencilerin; mobil robotların ağırlık merkezi ve denge ayarları, tekerlerin sürtünme durumları, motorlara uygulanacak güç gibi ayarlamalar ile ilgili tanımlamalar yapmaları beklenmektedir. Ayrıca ders kapsamında yazılmış olan Arduino IDE kodlarında da yarış için gerekli değişiklikleri belirlemeleri ve yapacakları seçimleri tanımlamaları istenmektedir.

Fikir üretme süreci: Bu aşamada öğrencilerin yukarıda belirlenen mobil robot tasarımını ve Arduino IDE programına yönelik kod yazım işlemlerini nasıl gerçekleştirilebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler hem robot tasarımı hem de program kodlarında aşağıdakilere benzer fikirler üretebilir:

- Mobil robotun ağırlık merkezini belirleyip fiziksel tasarımda devre elemanlarının araba üzerindeki konumlarını değiştirme,
- Tekerleklerin dönme hızları ve performanslarına dayalı olarak yeni tasarım olasılıkları,
- Her yere dönebilen metal tekerleğin araba üzerindeki konumu,
- Mobil robot üzerine ağırlık ekleme,
- Eğer yarış pistinin bulunduğu zemin fazla kaygan ise tekerleklere lastik takma,
- Ders kapsamında geliştirilen robotun Arduino IDE içerisindeki kodlarındaki başlangıçtaki standart hızı olan 100 değerini değiştirme,
- Ders kapsamında geliştirilen robotun Arduino IDE içerisindeki kodlarındaki hız artırımını olan +5 değerini değiştirme,
- Ders kapsamında geliştirilen robotu telefon ya da tablet üzerinden kontrol edecek tuşlara Arduino IDE kodunda yeni fonksiyonlar ekleyerek daha dengeli, hızlı ve stabil hale getirmek.

Öğrencilerin önce tanımladıkları ve hakkında fikir ürettikleri robot tasarımının son halini vermeleri gerekmektedir. Mobil robotun ideal yarış hızında ve dayanıklılığında sorunsuz hareket ettiği test edilmelidir. Sorunlar varsa gözden geçirilip düzeltilmelidir. Sonrasında ise programı tamamlayıp, kontrol edip eksiklikleri giderilip, yarışa hazır hale getirilmelidir.

3. ADIM: DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat bulurlar. 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 eğitmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- Dairesel hareket ettirilen mobil robotta, robotun dönüşleri gerçekleştirebilmek için neler yaptınız?
 - Yarış için ideal mobil robot tasarımınız için fiziksel hangi değişiklikleri yaptınız? Sonuçları ne oldu?
 - Mobil robot yarışı için Arduino IDE içerisindeki program kodlarında ve telefon/tablet için tuş düzenlemelerinde ne gibi değişiklikler yaptınız? Yarış kapsamında başarılı değişiklikler olduğunu düşünüyor musunuz?
 - Kullandığımız yöntemler bu sıkıntıları gidermekte başarılı oldu mu?
 - Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
 - Grup arkadaşınızdan ne/neler öğrendiniz?
- Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

4. PROJE GELİŞTİRMEDE EMPATİ

Bu Hafta Gerçekleştirilecek Etkinlikler: Dersin sonunda eğitmen öğrencilerin proje kayıt defterlerinde bu haftaya ilişkin çözülmesi istenen empati bölümüne ait soruları cevaplayıp cevaplamadıklarını kontrol eder. Bu aşamada problemi belirlemeden önce öğrenci gruplarının empati kurma adımını gerçekleştirip gerçekleştirmedikleri kontrol edilmelidir. Proje süreci sonunda büyük emek harcanarak geliştirilecek olan teknolojik çözümün günlük hayattaki problemi çözmesi ve bireylerin hayatlarını kolaylaştırması istendiğinden planlanan/öngörülen ortamlardaki ihtiyaçların ve beklentilerin iyi belirlenmesi ve proje kayıt defterine kayıt altına alınması önemlidir.

Bir Sonraki Haftaya Hazırlık (9. Hafta): Gruplara gelecek hafta için belirledikleri projeler için tanımlama adımlarına yönelik proje kayıt defterindeki ilgili yerleri doldurarak derse gelmeleri istenir. Tanımlama adımında, empati adımında toplanan bilgiler bir bütün hâline getirilir. Süreç esnasında çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözüme yönelik tasarımların ve teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerektiğine yönelik soruların cevaplanmasının önemi vurgulanır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki beklentiler nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI (Tanımlama – 9. HAFTA)**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

9.Bölüm – Çizgi İzleyen Mobil Rehber Robot

Ön bilgi:

- Öğrenciler Arduino UNO kartı bileşenlerini bilir ve Arduino IDE arayüzünü kullanabilir.
- Öğrenciler robot kitini kullanarak mobil robot tasarımı yapabilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.
- Öğrenciler Arduino IDE arayüzü ile değişkenler, if/else, for döngüsü, dizi vb. yöntemleri ile kod yazarak elektronik devre elemanlarını yönetebilir.
- Öğrenciler birçok sensörü ve devre elemanını ortak bir proje çerçevesinde bir arada kullanabilir ve ilgili yönergeleri tamamlayabilecek şekilde Arduino IDE programını yazabilir.

Bölüm Kazanımları:

- Öğrenciler SD kart modülünün çalışma prensibini ve üzerindeki pinlerin işlevlerini bilir.
- Öğrenciler Arduino IDE üzerinde SD kart içindeki bir ses dosyasını çalıştırabilecek program kodunu yazabilir.
- Öğrenciler kızılötesi (infrared) sensörlerin çalışma prensiplerini bilir.
- Öğrenciler 4'lü çizgi izleyen sensör setindeki sensörlerin çalışma prensibini bilir ve uygulamalarda kullanır.
- Öğrenciler 4'lü çizgi izleyen sensör seti ile çizgi izleyen bir mobil robot tasarımı yapabilir.
- Öğrenciler 4'lü çizgi izleyen sensör seti ile çizgi izleyen bir mobil robotun istenilen işlevleri gerçekleştirebilecek şekilde Arduino IDE içerisinde program kodlarını yazabilir.
- Öğrenciler SD kart modülü, 4'lü çizgi izleyen sensör seti, mobil robot kiti ve LCD ekranlar ile ilgili gerçekleştirilecek bir etkinliğin devresini kurabilir ve Arduino IDE programını yazabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle SD kart modülü ile SD kart içerisindeki ses dosyalarını çalıştırabilmelerini, kızılötesi (infrared) sensörlerin çalışma prensipleri hakkında bilgi sahibi olmalarını, 4'lü çizgi izleyen sensör seti ile çizgi izleyen bir mobil robot tasarımı yapabilmelerini sağlamaktır. Böylece gerçekleştirilen bu uygulamalara LCD ekran ekleyerek yeni bir etkinlikte kullanabilme becerilerini kazandırmaktır.

Göze ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	Mobil robot kiti
Breadboard	I2C Entegreli 4x20 LCD Ekran
SD kart modülü	Erkek-Erkek bağlantı kablosu
4'lü çizgi izleyen sensör	Erkek-Dişi bağlantı kablosu
Hoparlör	

Haftanın İşlenişi:

Gözle: SD kart modülü, kızılötesi (infrared) sensörler ve 4'lü sensör seti hakkında temel bilgi ve özelliklerinin verilmesi.

Uygula: SD kart modülünü kullanarak SD kart içerisindeki ses dosyalarının çalıştırılması ve çizgi izleyen mobil robot tasarımı için örnek devrelerin oluşturulması ve programlanması.

Tasarla: Mobil rehber robot tasarımı

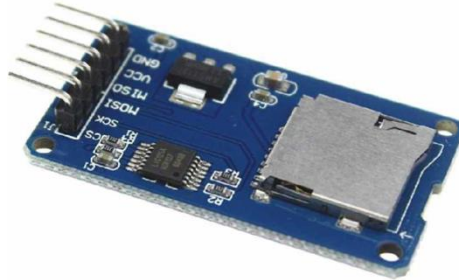
Üret: Mobil rehber robotunun istenilen görevleri yerine getirebilmek için devre bağlantılarının yapılması ve Arduino IDE içerisinde program kodlarının yazılması.

Değerlendir: Yansıtma Etkinliği.

1. ADIM: GÖZLE ve UYGULA**1.1 Gözle ve Uygula – SD Kart Modülü**

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
Breadboard
SD Kart Modülü
SD Kart
Hoparlör

SD kart modülü, SPI (Serial Peripheral Interface) protokolü üzerinden SD kartlar ile bilgi saklama, okunan bilgileri hafıza kartına yükleme ve sürekli gelen bilgileri kayıt altına alma gibi işlemlerin yapılabildiği bir modüldür. Üzerinde dahili 3.3 V voltaj regülatörü de bulunan modül ile 3.3 V'luk ve 5 V'luk sistemlerle uygulamalar yapılabilmektedir. Aşağıdaki resimde SD kart modülü görülmektedir.



Resim 9.1: SD Kart Modülü

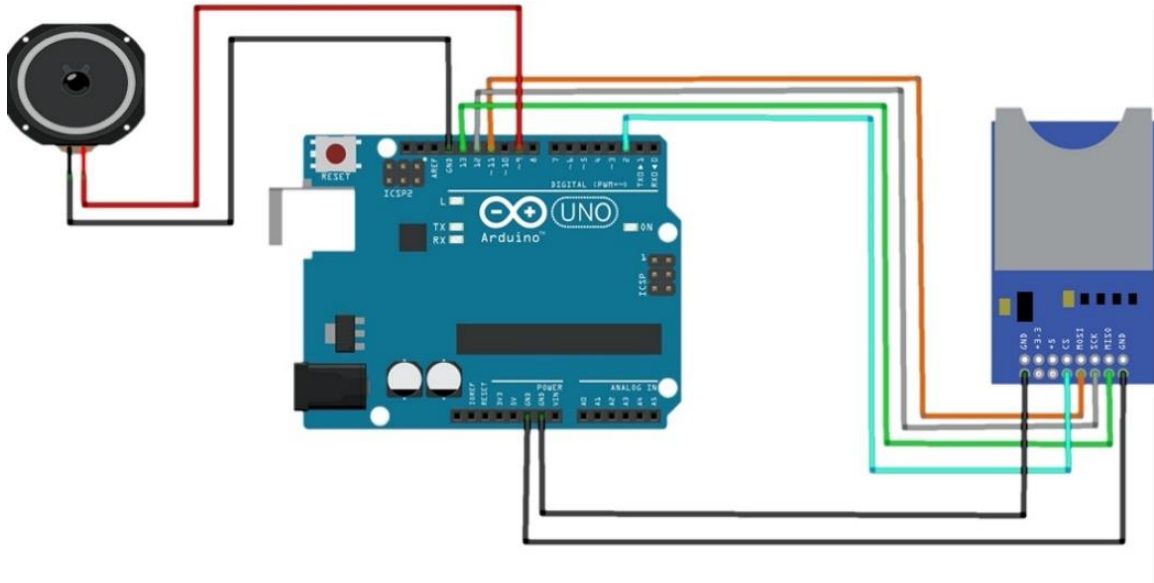
SD kart modülünün kütüphanelerce kullanılması için SD veya Micro SD kartların öncelikle formatlanması gerekmektedir. Bu yüzden kartın 2 GB'a kadar Fat16, 2 GB'dan büyük kapasiteler için ise Fat32 formatında formatlanması gerekecektir. SD kart modülünün bağlantısında 6 pin kullanılmaktadır. (1) **VCC** pini SD kart modülüne güç sağlayan pin olup Arduino UNO'da 5 V pinine bağlanır. (2) **GND** pini Arduino UNO'da GND pinine bağlanır. (3) **MISO** pini modülden SPI çıkışını sağlar. (4) **MOSI** pini modüle SPI girişidir. (5) **SCLK** pini Arduino UNO ile haberleşmeyi sağlayıp verilerin iletimini sağlar. (6) **CS** pini SPI protokolü ile belirli cihazları etkinleştirmek ve devre dışı bırakmak için kullanılır.

Temel bilgilerden sonra eğitmen aşağıdaki resimde devre şeması verilen uygulamada SD kart modülünü kullanarak SD kart içerisindeki ses dosyasının çalınması uygulamasını öğrencilere gösterip uygulamalarını isteyecektir. Öncelikle bu uygulama için SD kart içerisindeki ses dosyasının WAV formatında ve 8 bit, 8000 Hz yada 16000 Hz Stereo özelliğinde olması gerekmektedir. Uygulama öncesinde hazır bir WAV ses dosyası kullanılabileceği gibi dönüştürücü programlar ya da web siteleri kullanılarak bir mp3 dosyası da yukarıdaki özelliklere uygun olarak dönüştürülerek kullanılabilir (Bu çalışmadaki ses kayıtları için All2Wav programı kullanılmıştır). Örnek uygulamaya yönelik aşağıdaki resimde verilen program kodunda yer alan “1.wav”, “ses.wav” ve “6.wav” ses dosyaları örnek amaçlı yer almaktadır. Eğitmen, gerçekleştirilecek olan uygulamada ders süresini göz önüne alarak kullanılacak ses dosyası sayısına karar vermelidir.

Dikkat

SD kart içerisinde yer alacak olan ses dosyalarının isimlendirilmesinde Türkçe karakter kullanılmaması ve ses dosyalarına verilecek isimlerin kısa olmasını öneririz.

Ayrıca örnek program kodunda da yer aldığı gibi birden fazla ses kaydının kullanılacağı bir uygulama için kayıtların süresi kadar “delay” kodu eklenmesi ve kod içerisinde yer alan “volume” ile sesin sadece açılıp kapatılabileceğinin bilgilerinin öğrencilere hatırlatılması yerinde olacaktır.



Resim 9.2: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

sdkart_ses
1
2 #include <SD.h>
3 #define SD_ChipSelectPin 2
4 #include <TMRpcm.h>
5 #include <SPI.h>
6
7 TMRpcm tmrpcm;
8
9 void setup() {
10
11     tmrpcm.speakerPin = 9;
12
13     if (!SD.begin(SD_ChipSelectPin)) {
14         return;
15     }
16     tmrpcm.volume(1);
17 }
18
19 void loop() {
20
21     tmrpcm.play("1.wav");
22     delay(1000);
23     tmrpcm.play("ses.wav");
24     delay(13000);
25     tmrpcm.play("6.wav");
26     delay(1000);
27 }
28
Done compiling

```

Resim 9.3: Gözle-Uygula Etkinliği Örnek Kodu

Öğrencilere program kodlarında “*TMRpcm*”, “*SD*” ve “*SPI*” kütüphanelerinin ekleneceği ve “*tmrpcm*” isimli bir nesnenin ekleneceği özellikle vurgulanır. Eğitimci daha sonra öğrencilerden aynı devreyi kurmalarını ve program kodlarını yazmalarını ister.

1.2 Gözle ve Uygula – Kızılötesi (Infrared) Sensör

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
4'lü Çizgi İzleyen Sensör Seti

Ultrasonik sensörler gibi **kızılötesi (infrared)** sensörler de endüstri alanında gerekli süreç ve kontrol amaçlı işlemlerde uzaklık ölçümü için yaygın şekilde kullanılan sensörlerdendir. Özellikle robotik uygulamalarda kızılötesi sensörlerin farklı duyarlılık düzeylerinde ölçüm yapabilme özelliklerinden yararlanılmaktadır. Ultrasonik sensörlere oranla daha ucuz olması ve hızlı tepki vermesi robotik uygulamaların geliştirilmesinde büyük bir katkı sağlamıştır.

Genel olarak ifade edilecek olursa kızılötesi sensörler kızılötesi dalga boyunu kullanarak çalışırlar. Öncelikle içerisindeki lens aracılığı ile kızılötesi LED tarafından dar huzmeli ışık yayılır ve cisimlerden yansıyan ışık, ikinci bir lens aracılığıyla konuma duyarlı foto algılayıcı tarafından alınır. Bu aşamada foto algılayıcının iletkenliği kullanılarak bir gerilim üretilir ve bu gerilim ile uzaklık hesaplama işlemi gerçekleştirilir. Yani sensörün içinde bulunan LED, kontrol etmek istediğimiz bilgi ile aynı dalga boyuna sahip bir ışın üretir. Cismin üzerine düşen

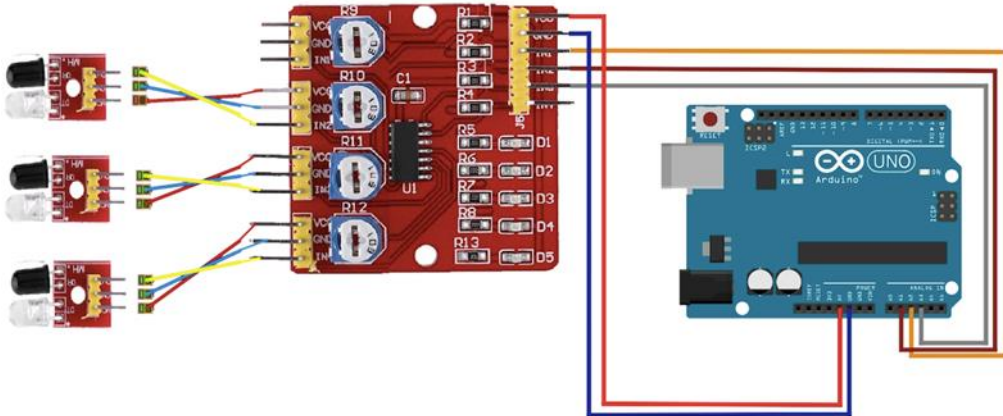
ve geri yansıyan kızılötesi ışınları bir foto algılayıcı denetler ve sensör geri dönen ışın sinyaline göre algılama işini gerçekleştirir. Buradaki önemli noktalardan biri kızılötesi sensörlerin çalışma prensiplerinde elde edilen çıkış ile uzaklığın ters orantılı şekilde çalışmasıdır. Diğer bir ifadeyle sensörün önünde yer alan cismin uzaklığı arttıkça kızılötesi sensörün gerilimi düşmektedir. Aşağıdaki resimde kızılötesi sensör görülmektedir.



Resim 9.4: Kızılötesi Sensör

Set içerisinde yer alan 4'lü çizgi izleyen sensör seti, üzerinde 4 adet kızılötesi alıcı verici modül bulunduran ve bu modüller yardımı zeminin rengindeki koyuluk ve açıklık farkına algılayarak çizgi bilgisine erişebilen kittedir. Set içerisindeki sette bir çizgi sensörü bir sayı hücresinden oluşur ve her hücre bir gönderen ve bir alıcıdan oluşmaktadır. Tüm sensörlerin hassasiyet ve mesafe ayarı kontrol kartı üzerinde yer alan potansiyometrelerden yapılabilmektedir.

Temel bilgilerden sonra eğitmen aşağıdaki resimde örnek devre şeması verilen devreyi kurar ve Arduino IDE programı yardımıyla 4'lü çizgi izleyen sensör setindeki 3 sensörün siyah ve beyaz çizgiler gördüklerinde "0" ve "1" değerlerini serial port ekranına yazdırılması ile sensörlerin çalışma mantığını öğrencilere aktarır. Bu uygulamada her bir sensör (sol, sağ ve orta sensörler) için *siyah çizgi gören sensörlerin "1" değerini; beyaz çizgi gören sensörlerin ise "0" değerini* ekrana yazdırdıkları hatırlatılmalıdır. Eğitmen daha sonra öğrencilerden aynı devreyi kurmalarını ve program kodlarını yazmalarını ister. Etkinliğin örnek kodu aşağıdaki resimde görülmektedir.



Resim 9.5: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

cizgi_izleme_mantik §
1 #define SensorSol A1
2 #define SensorOrta A2
3 #define SensorSag A3
4
5 void setup() {
6
7   pinMode(SensorSol, INPUT);
8   pinMode(SensorOrta, INPUT);
9   pinMode(SensorSag, INPUT);
10  Serial.begin(9600);
11  Serial.println("Cizgi Sensoru Deneme");
12  delay(2000);
13  }
14
15 void loop() {
16
17   Serial.print(digitalRead(SensorSol));
18   Serial.print(",");
19   Serial.print(digitalRead(SensorOrta));
20   Serial.print(",");
21   Serial.println(digitalRead(SensorSag));
22   delay(300);
23
24   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 0){
25   }
26
27   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 1){
28   }
29
30   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
31   }
32
33   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 1){
34   }
35
36   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
37   }
38 }
39

```

Resim 9.6: Gözle-Uygula Etkinliği Örnek Kodu

1.3 Uygula – Çizgi İzleyen Mobil Robot (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Bağlantı kabloları
4'lü Çizgi İzleyen Sensör Seti
Mobil Robot

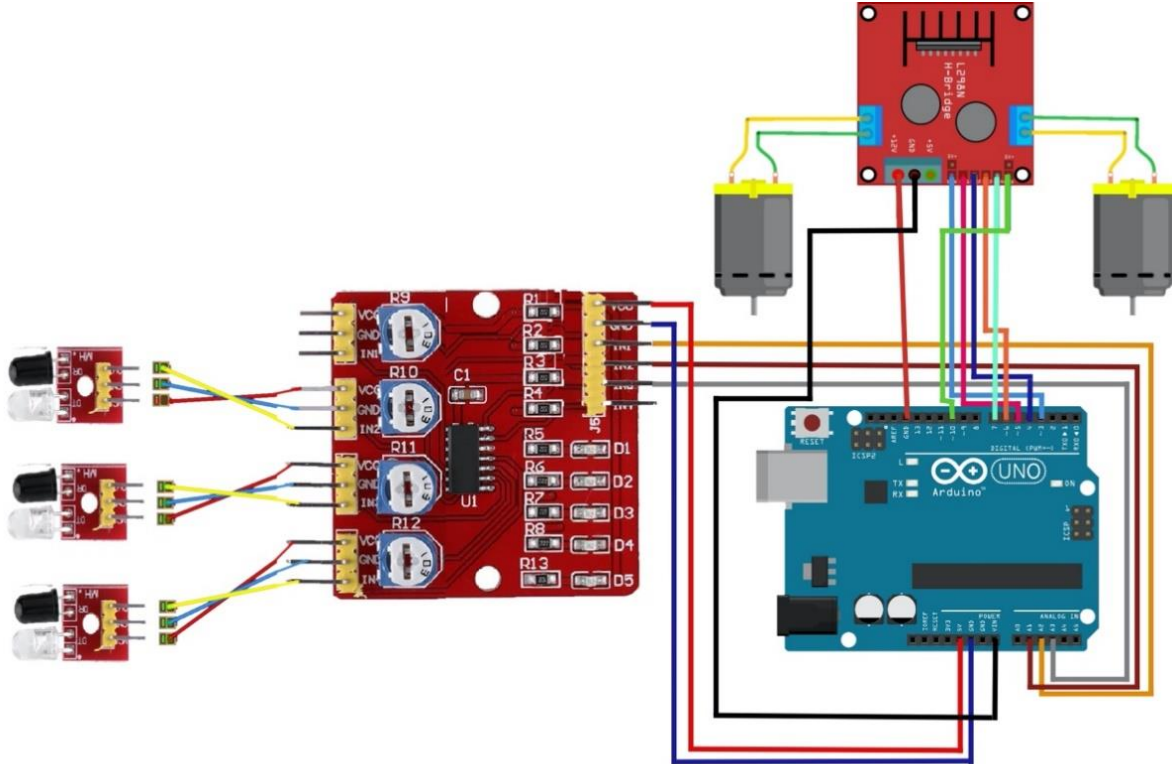
Bu etkinlik için eğitmen öğrencilerden geçen hafta kurulumu gerçekleştirilen mobil robotun aşağıdaki kurallar çerçevesinde çizgi takibini yapacak devreyi kurmalarını ve Arduino IDE içerisinde programı yazmalarını isteyecektir.

Kurallar:

- Mobil robot başlangıçta 75 hızda ilerleyecek,
- Mobil robot sadece ortadaki sensör siyah (sağ ve sol sensörler beyaz) çizgi gördüğünde düz ilerleyerek siyah çizgide kalmaya devam edecek,

- Mobil robot eğer sağdaki sensörde siyah (sol ve ortadaki sensörler beyaz) çizgi görürse sağ taraftaki tekerlek hızı 150'ye çıkararak robotun ortadaki sensörde siyah çizgiye dönmesi ve düz gitmesi sağlanacak,
- Mobil robot eğer soldaki sensörde siyah (sağ ve ortadaki sensörler beyaz) çizgi görürse sol taraftaki tekerlek hızı 150'ye çıkararak robotun ortadaki sensörde siyah çizgiye dönmesi ve düz gitmesi sağlanacak,
- Eğer her üç sensör de beyaz çizgi görürse mobil robot duracak.

Etkinlikle ilgili örnek devre şeması ve kod aşağıdaki resimlerde verilmiştir.



Resim 9.7: Uygula Etkinliği Örnek Devre Şeması


```

cizgi_izleyen_son
1 #define SensorSol A1
2 #define SensorOrta A2
3 #define SensorSag A3
4
5 #define MotorR1 4
6 #define MotorR2 5
7 #define MotorRE 3
8
9 #define MotorL1 7
10 #define MotorL2 6
11 #define MotorLE 10
12
13 int hiz = 75;
14
15 void setup() {
16
17   pinMode(SensorSol, INPUT);
18   pinMode(SensorOrta, INPUT);
19   pinMode(SensorSag, INPUT);
20   Serial.begin(9600);
21   Serial.println("Cizgi Sensoru Test");
22   delay(2000);
23
24   pinMode(MotorR1, OUTPUT);
25   pinMode(MotorR2, OUTPUT);
26
27   pinMode(MotorL1, OUTPUT);
28   pinMode(MotorL2, OUTPUT);
29
30 void loop() {
31
32   Serial.print(digitalRead(SensorSol));
33   Serial.print(",");
34   Serial.print(digitalRead(SensorOrta));
35   Serial.print(",");
36   Serial.println(digitalRead(SensorSag));
37   delay(300);
38
39   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 0){
40     ileri();
41   }
42
43   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 1){
44     sag();
45   }
46
47   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
48     sol();
49   }
50
51   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 1){
52     dur();
53   }
54
55   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
56     dur();
57   }
58 }
59
60 void ileri() {
61   digitalWrite(MotorR1, HIGH);
62   digitalWrite(MotorR2, LOW);
63   analogWrite(MotorRE, hiz);
64
65   digitalWrite(MotorL1, HIGH);
66   digitalWrite(MotorL2, LOW);
67   analogWrite(MotorLE, hiz);
68 }
69
70 void sag() {
71   digitalWrite(MotorR1, HIGH);
72   digitalWrite(MotorR2, LOW);
73   analogWrite(MotorRE, 75);
74
75   digitalWrite(MotorL1, HIGH);
76   digitalWrite(MotorL2, LOW);
77   analogWrite(MotorLE, 150);
78 }
79
80 void sol() {
81   digitalWrite(MotorR1, HIGH);
82   digitalWrite(MotorR2, LOW);
83   analogWrite(MotorRE, 150);
84
85   digitalWrite(MotorL1, HIGH);
86   digitalWrite(MotorL2, LOW);
87   analogWrite(MotorLE, 75);
88 }
89
90 void dur() {
91   digitalWrite(MotorR1, HIGH);
92   digitalWrite(MotorR2, LOW);
93   analogWrite(MotorRE, 0);
94
95   digitalWrite(MotorL1, HIGH);
96   digitalWrite(MotorL2, LOW);
97   analogWrite(MotorLE, 0);
98 }

```

Resim 9.8: Uygula Etkinliđi Örnek Kodu

2. ADIM: TASARLA ve ÜRET

2.1 Tasarla - Müzeler için Mobil Rehber Robot

Müzeler, zamanla veya insanların ilerlemesinden dolayı ortadan kalkma tehlikesi ile karşı karşıya kalan eserleri muhafaza etmeye yarayan ve aynı zamanda nesnelerin uzun tarihini insanlara öğretecek dinamik kurumlardır. Çağdaş anlamda müzeler, koleksiyonunda barındırdığı eserlerin anlamını, yararını ve değerini kavratarak, bilgisini geliştirme yollarını öğreten bir eğitim kurumu görevini üstlenir. Dünyanın pek çok müzesinde engellilerin müzeyi rahat gezebileceđi ve eğitimlere katılabileceđi projeler üretilmektedir. Bu hedefler doğrultusunda müzelerde; iç mekan tasarımları, müzeyi ve eserleri tanıtan eğitim araçları deđişmiş ve rehberler kullanılmaya başlanmıştır. Bu tür girişim ve projeler müzelerin hizmet

kalitesini arttırarak engelli bireylerin müzeleri kullarımlarını daha kullanışlı düzeye getirilmesini sağlamaktadır.

Bu etkinlikte öğrencilerden fiziksel engelli bireylerin müzeleri daha rahat gezebilmelerini sağlayacak ve müzedeki eserler hakkında bilgi verecek şekilde onlara eşlik edecek bir “**mobil rehber robot**” tasarımları istenmektedir. Mobil rehber robot eserler hakkında engelli bireylere hem sesli hem de görsel olarak bilgilendirme yapacaktır.

Dikkat

1. Öğrenciler tasarımlarına geçmeden önce eğitmen tarafından seçilecek bir müze için (örneğin: Çanakkale Troya Müzesi) oradaki 2 önemli eserin (örneğin: Kybele Heykelciği ve Polyksena Lahdi) isimleri, dönemleri ve önemi hakkında bilgi içeren, değişen sürelerde ve “wav” formatında 2 farklı ses dosyası hazırlamaları istenmelidir. Bu tür bir hazırlık tasarlanacak mobil rehber robotun görsel bilgilendirme metni için de kolaylık sağlayacaktır.
2. Aşağıda örnek Arduino IDE programında görsel bilgilendirme için eser isimleri ilgili yerlerde “1.eser bilgisi ve 2. eser bilgisi” olarak yer alacak olup, ses dosyaları da “1.wav” (1. eser ses dosyası) ve “2.wav” (2. eser ses dosyası) şeklinde ifade edilmiştir. Öğrenciler kendi tasarımlarında dosya içerik ve isimlendirme işlemlerinde istedikleri tercihi yapabilirler.
3. Öğrenciler tasarımlarına geçmeden önce eğitmen tarafından mobil rehber robotun etkinlik kapsamında izleyeceği rotayı hazırlaması gerekmektedir. Yukarıda ders kapsamında çizgi izleyen mobil robot için kullanılan materyal bu bölüme uyarlanabilir.

Tasarlanacak mobil rehber robotun çalışmaya başladıktan sonra aşağıdaki işlem adımlarını gerçekleştirmesi beklenmektedir.

1. Mobil rehber robot çalıştığında fiziksel engelli bireye eşlik eder şekilde yürüyüş yolu kenarındaki özel pistinde siyah çizgiyi takip ederek 75 hızda ilerlemelidir.
2. Mobil rehber robot sadece üç sensör de siyah çizgiyi okuduğunda (rehberlik edilirken bilgi verilecek eser sırası geldiğinde) ve beyaz rengi okuduğunda (mobil rehber robot siyah çizgiden çıktığında) durmalıdır.
3. Hakkında sesli ve görüntülü bilgi verilecek eser ile karşılaşıldığında mobil rehber robot için hazırlanan yolda üç siyah şerit olacaktır.
4. Mobil Rehber Robot çalışmaya başladıktan sonra fiziksel engelli bireye eşlik ederken; hakkında sesli ve görüntülü bilgi verilecek ilk eser ile karşılaşıldığında mobil rehber robot durmalı;
 - a. LCD ekran ile haberleşme başlamalı ve arka plan ışığı açılmalı. Daha sonra 1. eserin bilgisi LCD ekranda gösterilmeli, yanıp sönen imleç açılmalı ve sonrasında ise 1.eser ses dosyası çalmalıdır.
 - b. Daha sonra 1. eser ses dosyası süresi bittikten sonra yanıp sönen imleç kapatılmalı, ekran temizlenmeli, LCD ekran ve arka plan ışığı kapatılmalıdır.
 - c. Daha sonra mobil rehber robot 1.5 sn ileriye giderek tekrar siyah çizgide ilerlemeye başlamalıdır.

5. Mobil rehber robot ilerlemesini sürdürürken hakkında sesli ve görüntülü bilgi verilecek ikinci eser ile karşılaşıldığında mobil rehber robot yine durmalı;
 - a. LCD ekran ile haberleşme başlamalı ve arka plan ışığı açılmalı. Daha sonra 2. eserin bilgisi LCD ekranda gösterilmeli, yanıp sönen imleç açılmalı ve sonrasında ise 2.eser ses dosyası çalmalıdır.
 - b. Daha sonra 2. eser ses dosyası süresi bittikten sonra yanıp sönen imleç kapatılmalı, ekran temizlenmeli, imleç üst sol konuma (0,0) getirilmeli ve ekrana “Güle güle...” yazdırılmalıdır. Sonrasında da LCD ekran ve arka plan ışığı kapatılmalıdır.

2.2 Tasarla - Mobil Rehber Robot Tasarımı

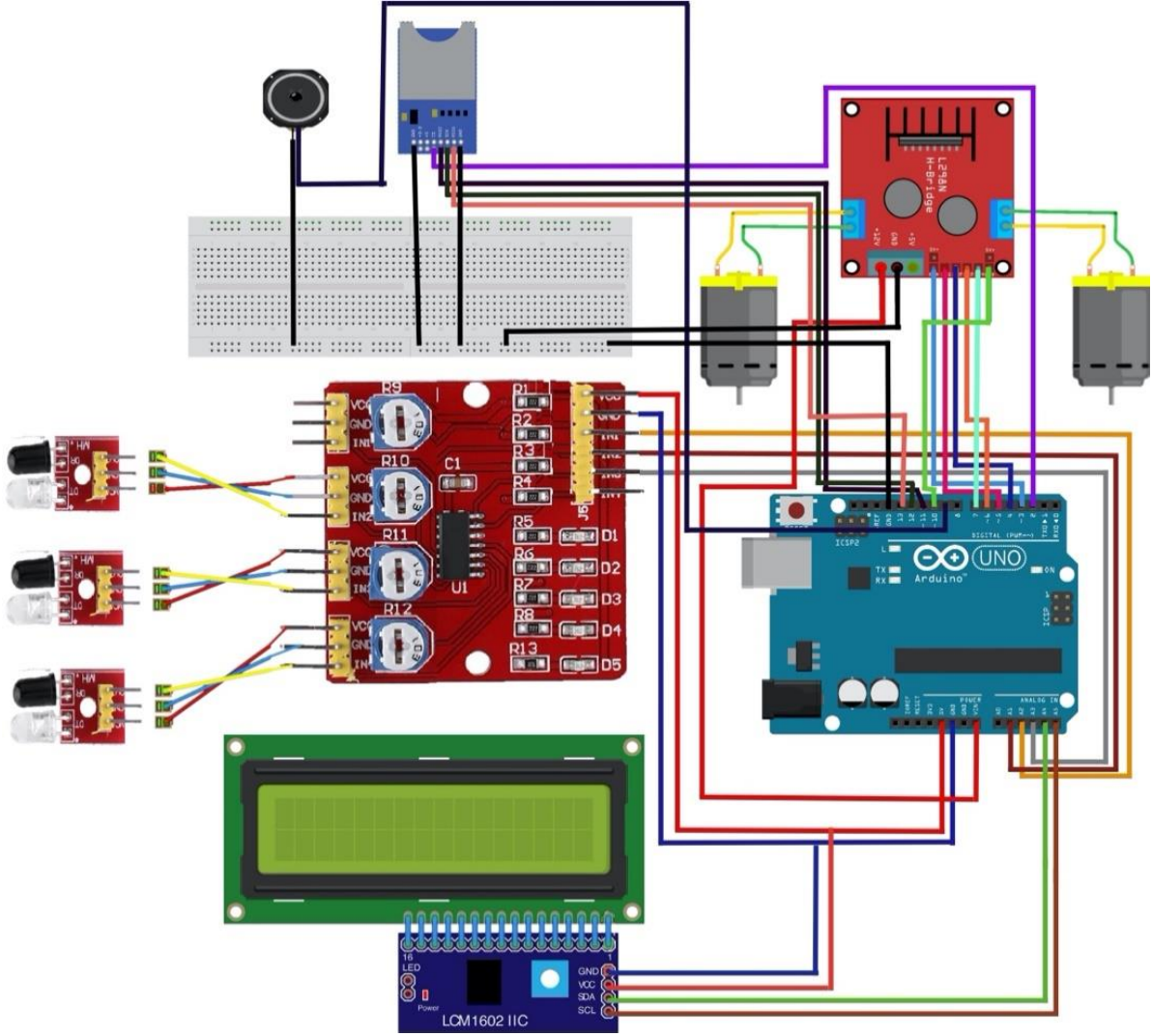
Öğrenciler geçen hafta oluşturmuş oldukları mobil robotun üzerine kuracakları devreleri gruplar halinde tartışır. Program kodlarının yazımında ve algoritmanın oluşturulmasında eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Etkinlik kapsamında hangi müzeden sesli ve görüntülü bilgi verilecek eserler seçilmeli ve içerikleri neler olmalı?
- Mobil rehber robotun hem görsel hem de işitsel bilgi vereceği için çok fazla modül ve sensör kullanılacağından Arduino UNO üzerindeki PWM ve analog pin yerleşimleri nasıl yapılmalıdır?
- Arduino UNO üzerinde kaç tane GND ve 5 V pini çoklanmalıdır?
- Mobil rehber robotun fiziksel tasarımı nasıl olmalıdır? Farklı devre elemanları ve sensörler mobil rehber robot üzerinde nasıl yerleştirilmelidir?
- Arduino IDE program algoritması nasıl kurulmalıdır? Hangi kütüphaneler, nesnelere, fonksiyonlar ve değişkenler kullanılmalıdır?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Mobil rehber robotun istenilen işlevleri gerçekleştirebilmesi için Arduino UNO üzerindeki pinlerin ekonomik kullanımı önemli olmalıdır. LCD ekran için ve çizgi izleyen sensörler için analog pinlerin kullanımı düşünülmelidir.
- Mobil rehber robotun fiziksel tasarımında çizgi takibinde ve durma işlemlerinde sorunsuz çalışabilmesi için bazı devre elemanlarının (örneğin LCD, breadboard, SD kart modülü vb.) üstte yer alması gerekirken bazılarının (örneğin kızılötesi sensör kartı vb.) ise aradaki katmanda yerleştirilmesi şeklinde tercihlerde bulunulabilir.
- Arduino IDE programında sesli ve görüntülü bilgi verilecek eserler ile bilgi verilmesi istenen kod bloğunda ses kaydı süresi göz önünde bulundurulmalıdır.
- Arduino IDE programında robotun gerçekleştirmesi istenen işlemlerden özellikle çizgi takibinde if döngü yapısı kullanılmalı ve eser hakkında bilgi verilmek istendiğinde diğer devre elemanlarına ilişkin kodlar yer almalıdır. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 9.9: Tasarla Etkinliği Örnek Devre Şeması

2.3 Üret- Mobil Rehber Robot Tasarımı

Öğrenciler mobil rehber robotunun yönergesini tamamlayabilecek şekilde tasarımlarını yaptıktan sonra bilgisayar başında Arduino IDE üzerinde kodları yazarlar. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında kodların, kütüphanelerin, değişkenlerin ve kontrol yapılarının kullanımında ve çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.

mobil_rehber_robot_son

```
1 #define SensorSol A1
2 #define SensorOrta A2
3 #define SensorSag A3
4
5 #define MotorR1 4
6 #define MotorR2 5
7 #define MotorRE 3
8
9 #define MotorL1 7
10 #define MotorL2 6
11 #define MotorLE 10
12
13 #include <Wire.h>
14 #include <LiquidCrystal_I2C.h>
15 LiquidCrystal_I2C lcd (0x27, 20, 4);
16
17 #include <SD.h>
18 #define SD_ChipSelectPin 2
19 #include <TMRpcm.h>
20 #include <SPI.h>
21
22 TMRpcm tmrpcm;
23
```

```
24 int hiz = 75;
25 int kosul = 0;
26
27 void setup() {
28   pinMode(SensorSol, INPUT);
29   pinMode(SensorOrta, INPUT);
30   pinMode(SensorSag, INPUT);
31   Serial.begin(9600);
32   Serial.println("Cizgi Sensoru Testi");
33   delay(2000);
34
35   pinMode(MotorR1, OUTPUT);
36   pinMode(MotorR2, OUTPUT);
37   pinMode(MotorL1, OUTPUT);
38   pinMode(MotorL2, OUTPUT);
39
40   tmrpcm.speakerPin = 9;
41
42   if (!SD.begin(SD_ChipSelectPin)) {
43     return;
44   }
45   tmrpcm.volume(1);
46 }
47
```



```

48 void loop() {
49   Serial.print(digitalRead(SensorSol));
50   Serial.print(",");
51   Serial.print(digitalRead(SensorOrta));
52   Serial.print(",");
53   Serial.println(digitalRead(SensorSag));
54   delay(300);
55
56   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 0){
57     ileri();
58   }
59
60   if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 1){
61     sag();
62   }
63
64   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
65     sol();
66   }
67
68   if(digitalRead(SensorSol) == 1 && digitalRead(SensorOrta) == 1 && digitalRead(SensorSag) == 1){
69     dur();
70     kosul=kosul+1;
71     if (kosul==1){
72       lcd.begin();
73       lcd.backlight();
74       lcd.print("1. eser bilgisi");
75       lcd.blink();
76       tmrpcm.play("1.wav");
77       delay(30000);
78       lcd.noBlink();
79       lcd.clear();
80       lcd.noDisplay();
81       delay(2000);
82       lcd.noBacklight();
83       delay(2000);
84       ileri();
85       delay(1500);
86     }
87     if (kosul>1){
88       lcd.begin();
89       lcd.backlight();
90       lcd.print("2. eser bilgisi");
91       lcd.blink();
92       tmrpcm.play("2.wav");
93       delay(40000);
94       lcd.noBlink();
95       lcd.clear();
96       lcd.home();
97       lcd.print("Gule gule...");
98       delay(2000);
99       lcd.noDisplay();
100      delay(2000);
101      lcd.noBacklight();
102      delay(2000);
103    }
104  }
105  if(digitalRead(SensorSol) == 0 && digitalRead(SensorOrta) == 0 && digitalRead(SensorSag) == 0){
106    dur();
107  }
108 }
109

```



```

110 void ileri() {
111   digitalWrite(MotorR1, HIGH);
112   digitalWrite(MotorR2, LOW);
113   analogWrite(MotorRE, hiz);
114
115   digitalWrite(MotorL1, HIGH);
116   digitalWrite(MotorL2, LOW);
117   analogWrite(MotorLE, hiz);
118 }
119
120 void sag() {
121   digitalWrite(MotorR1, HIGH);
122   digitalWrite(MotorR2, LOW);
123   analogWrite(MotorRE, 75);
124
125   digitalWrite(MotorL1, HIGH);
126   digitalWrite(MotorL2, LOW);
127   analogWrite(MotorLE, 150);
128 }
129
130 void sol() {
131   digitalWrite(MotorR1, HIGH);
132   digitalWrite(MotorR2, LOW);
133   analogWrite(MotorRE, 150);
134
135   digitalWrite(MotorL1, HIGH);
136   digitalWrite(MotorL2, LOW);
137   analogWrite(MotorLE, 75);
138 }
139
140 void dur() {
141   digitalWrite(MotorR1, HIGH);
142   digitalWrite(MotorR2, LOW);
143   analogWrite(MotorRE, 0);
144
145   digitalWrite(MotorL1, HIGH);
146   digitalWrite(MotorL2, LOW);
147   analogWrite(MotorLE, 0);
148 }
149

```

Resim 9.10: Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat bulurlar. 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 eğitmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- Çizgi izleyen mobil robot tasarımında robotunuzun siyah çizgide ilerlemesi için Arduino IDE kodunda ne gibi algoritmalar kullandınız?
 - Mobil rehber robot tasarımınızda PWM ve analog pinlerin kullanımında nasıl stratejiler izlediniz?
 - Mobil rehber robotun program kodunun algoritmasını oluştururken en çok nerede zorlandınız? Nasıl çözüm ürettiniz?
 - Mobil rehber robotun müze içerisinde daha fazla sayıda eser hakkında bilgi verebilmesi için tasarımınızda ne gibi değişiklikler yapmayı düşünürsünüz? Neden?
 - Kullandığımız yöntemler bu sıkıntıları gidermekte başarılı oldu mu?
 - Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
 - Grup arkadaşınızdan ne/neler öğrendiniz?
- Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

4. PROJEYİ TANIMLIYORUM

Bu Hafta Gerçekleştirilecek Etkinlikler: Dersin sonunda eğitimci öğrencilerin proje kayıt defterlerinde bu haftaya ilişkin tanımlama bölümüne ait soruları cevaplayıp cevaplamadıklarını kontrol eder. Öğrenci gruplarının empati çalışması sonrası gerçekleştirdikleri tanımlama bölümünde geliştirilecek teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı ve özelliklerinin açık şekilde yazılması ve belirlenmesi gerekmektedir. Proje kayıt defterindeki tanımlama bölümünde öğrenci gruplarının yazdıkları özellikler ve tasarıma ilişkin öngörülerin gerçekleştirilebilir olmasına dikkat edilmeli ve her grubun çıkarımları ortak şekilde değerlendirilmelidir.

Ayrıca öğrencilere proje kayıt defterlerinde son hafta proje sergisi günü teslim edilecek proje ayrıntıları bölümüne “**video**” kaydı için tüm çalışmalarının dijital kayıtlarını almalarının gerekliliği hatırlatılmalıdır.

Bir Sonraki Haftaya Hazırlık (10. Hafta): Gruplara gelecek hafta için belirledikleri projeler için fikir üretme adımlarına yönelik proje kayıt defterindeki ilgili yerleri doldurarak derse gelmeleri istenir. Fikir üretme aşamasında empati ve tanımlama bölümleriyle belirlenen problem durumu ve problem durumuna ilişkin hedeflerin gerçekleştirilecek olan özelliklere ilişkin üretilecek fikirlerin önemi eğitimci tarafından öğrencilere hatırlatılmalıdır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki beklentiler nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI (Tanımlama – 9. HAFTA)**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabildiği gerekmektedir?

○ **ÇÖZÜM ÖNERİSİ TASLAĞI (Fikir Üretme – 10. HAFTA)**

- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?

- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?



- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?



10.BÖLÜM - Sıcaklık, Nem Sensörü ve Wi-Fi Modülünün Kullanımı

Ön bilgi:

- Öğrenciler Arduino IDE içerisine kütüphane eklemesini ve kullanmasını bilir.
- Öğrenciler direncin çalışma prensibini bilir ve akım ile arasındaki ilişkiyi açıklayabilir.
- Öğrenciler Arduino UNO kartı bileşenlerini bilir ve Arduino IDE arayüzünü kullanabilir.

Bölüm Kazanımları:

- Öğrenciler DHT11 sıcaklık ve nem sensörünün çalışma prensibini bilir ve Arduino IDE üzerinde ortam sıcaklığını ve nemini ölçen programı yazabilir.
- Öğrenciler Wi-Fi modülünün çalışma prensiplerini ve Arduino IDE üzerinde DHT11 sıcaklık ve nem sensörünü de kullanarak ortam sıcaklığını ve nemini ölçen programı yazabilir.
- Öğrenciler Nesnelerin İnterneti (IOF) platformuna üye olabilir ve orada ilgili bölümleri açıp, yönetebilir.
- Öğrenciler ölçümler ile elde ettikleri verileri Nesnelerin İnterneti (IOF) platformunda görüntüleyebilir.
- Öğrenciler birçok sensörü ve devre elemanını ortak bir proje çerçevesinde bir arada kullanabilir ve ilgili yönergeleri tamamlayabilecek şekilde Arduino IDE programını yazabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin öncelikle “DHT11” sıcaklık ve nem sensörü ve Wi-Fi modülü gibi devre elemanlarının özelliklerini öğrenme ve bu devre elemanları ile farklı uygulamalar gerçekleştirebilmeleri amaçlanmaktadır. Ayrıca sensör ve devre elemanlarının birlikte yer aldığı ortak etkinlikleri gerçekleştirebilmeleri ve programlarını yazabilmeleri de hedeflenmektedir. Bu kapsamda programlamada farklı kütüphanelerin, nesnelerin, kontrol yapılarının ve değişkenlerin uygulamalar içerisindeki kullanımlarını da deneyimlemeleri hedefler içerisinde yer almaktadır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	Mobil robot kiti
Breadboard	Bilgisayar / Tablet
DHT 11 sıcaklık ve nem sensörü	Erkek-Erkek bağlantı kablosu
Wi-Fi modülü	Erkek-Dişi bağlantı kablosu

Haftanın İşlenişi:

Gözle: DHT11 sıcaklık ve nem sensörü ve Wi-Fi modülü tanımlarının yapılması ve özelliklerinin verilmesi.

Uygula: DHT11, Wi-Fi modülü, Buzzer, LED ekran, ultrasonik sensör gibi devre elemanlarının kullanıldığı örnek devrelerin oluşturulması ve programlanması.

Tasarla: Mobil uzay keşif robotu tasarımının yapılması.

Üret: Mobil uzay keşif robotunun istenilen görevleri yerine getirebilmesi için devre bağlantılarının yapılması ve Arduino IDE içerisinde program kodlarının yazılması.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

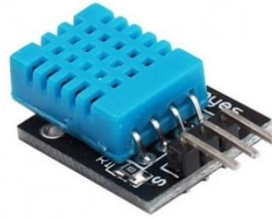
1. ADIM: GÖZLE ve UYGULA

1.1 Gözle ve Uygula – DHT11 Sıcaklık ve Nem Sensörü Kullanımı

Malzeme Listesi	
Arduino UNO	DHT11
Bağlantı kabloları	Breadboard

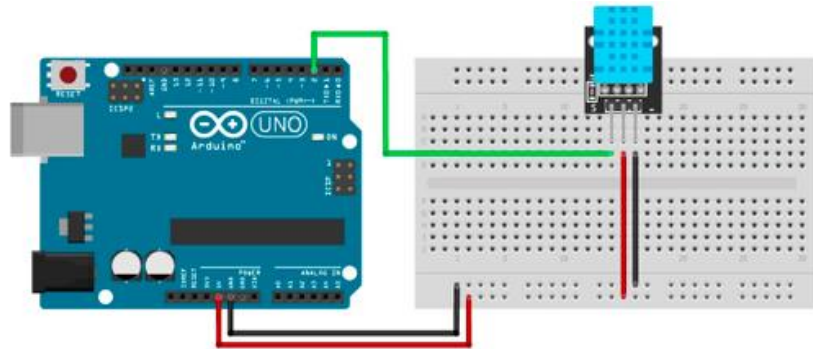
DHT11 Sıcaklık ve Nem Sensörü

DHT11 sıcaklık ve nem sensörü ortamdaki sıcaklık ve nem değerlerini ölçerek dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. DHT11 sıcaklık ve nem sensörü “ ± 2 °C” hassasiyetle 0-50°C arasında sıcaklık ölçebilmektedir. %20-90 RH arasında da %5 hata payı ile nem ölçebilme özelliğine sahip olan sensör 8 bit mikroişlemciye sahiptir. Bu özellikleriyle DHT11 hızlı ve kaliteli veri transferi gerçekleştirerek yüksek güvenilirlikte ve uzun dönem kullanıma uygun bir birimdir. Aşağıdaki resimde görülen DHT11 sıcaklık ve nem sensörünün **VCC**, **DATA**, **NC** ve **GND** pinleri bulunmaktadır.



Resim 10.1: DHT11 Sıcaklık ve Nem Sensörü

Eğitmen aşağıdaki resimde görülen örnek devre şeması ile verilen devreyi kurar ve Arduino IDE programı üzerinden ortam sıcaklığını ve nemini ölçerek öğrencilere gösterir. Öğrencilere program kodlarında dht11 kütüphanesinin ekleneceği ve DHT11 isimli bir nesnenin ekleneceği özellikle vurgulanır. Eğitmen daha sonra öğrencilerden aynı devreyi kurmalarını ve aşağıdaki resimde görülen örnek program kodlarını yazmalarını ister.



Resim 10.2: Gözle-Uygula Etkinliği Örnek Devre Şeması

```

DHT11_serial
1 #include <dht11.h>
2
3 int dht11Pin = 2;
4 float sicaklik, nem;
5 dht11 DHT11;
6
7 void setup() {
8
9 }
10
11 void loop() {
12
13     Serial.begin(9600);
14     Serial.println("olcum:");
15     DHT11.read(dht11Pin);
16     sicaklik = (float)DHT11.temperature;
17     nem = (float)DHT11.humidity;
18     Serial.println(sicaklik);
19     Serial.println(nem);
20     delay(1000);
21 }

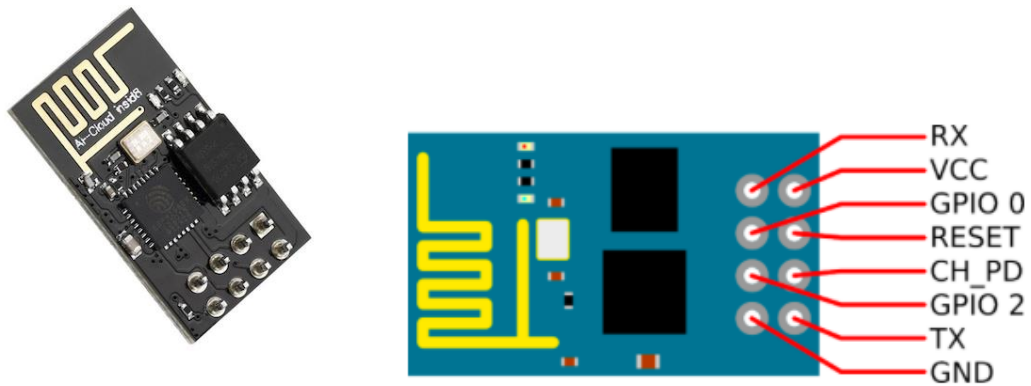
```

Resim 10.3: Gözle-Uygula Etkinliği Örnek Kodu

1.2 Gözle ve Uygula – Wi-Fi Modülü

Malzeme Listesi	
Arduino UNO	Wi-Fi modülü
Bağlantı kabloları	DHT11
Breadboard	

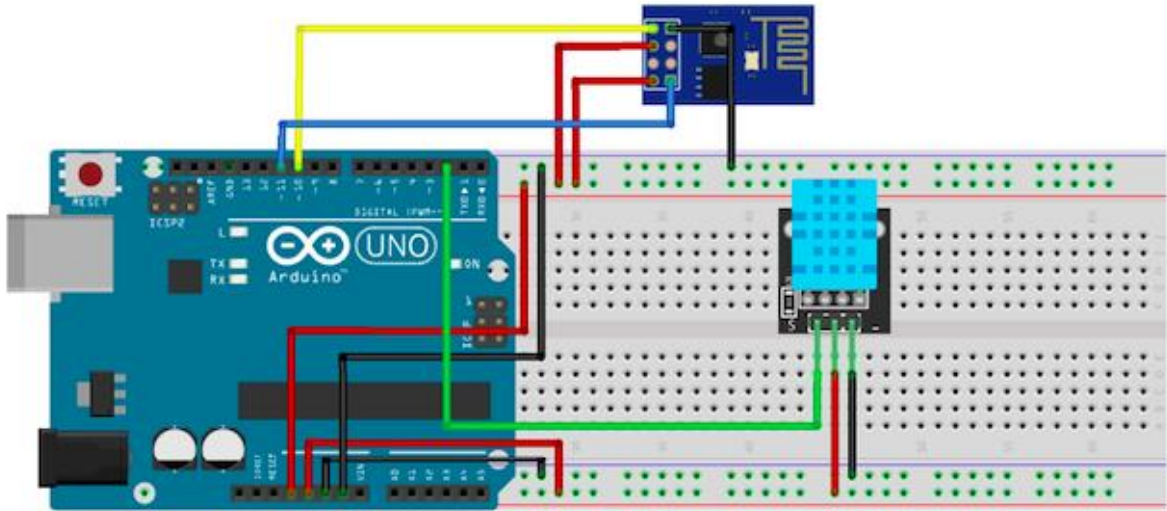
Üzerinde dahili anten bulunan “**ESP8266 Wi-Fi modülü**”, kablosuz ağlara kolaylıkla bağlanabilen ve veri paketlerini alıp gönderme özelliğine sahip bir modüldür. TCP/IP protokolünü destekleyen ESP8286 ile Nesnelerin İnterneti (IOT) platformunda en çok tercih edilen Wi-Fi modülü olarak içerisinde 32-bitlik CPU ve üzerinde yer alan GPIO pinleri ile hem kablosuz ağlara bağlanmak hem de kablosuz erişim noktası kurmak mümkündür. Dolayısıyla ESP8266 herhangi bir mikro denetleyici veya mikroişlemciye bağlanmadan bağımsız olarak da çalışabilir. Aşağıdaki resimde ESP 8266 Wi-Fi modülü ve ayak yapısı görülmektedir.



Resim 10.4: ESP 8266 Wi-Fi Modülü ve Ayak Yapısı

ESP8266 Wi-Fi Modülü Pinleri	
Pin İsmi	Görevi
RX	Modüle veri alma
VCC	3.3 V Enerji ile besleme
GPIO 0	Dijital G / Ç programlama (boş)
RESET	Wi-Fi modülü resetleme (bir buton ile) (boş)
CH_PD	3.3 V Enerji ile besleme
GPIO 2	Dijital I / O programlama (boş)
TX	Modülden veri gönderme
GND	Toprak hattı

ESP8266 Wi-Fi modülü üzerinde toplam 8 Pin bulunmaktadır. Birçok Arduino UNO etkinliğinde ESP866 Wi-Fi modülü üzerindeki 5 pin kullanılmaktadır. Yukarıdaki tabloda bu pinlerin isimleri ve görevleri belirtilmiştir. Devre bağlantılarında dikkat edilmesi gereken nokta VCC ve CH_PD ile belirtilmiş olan besleme gerilimlerinin 3.3 V olmasının gerekliliğidir. Aşağıda ortam sıcaklığı ve nemini ölçüp “ThingSpeak” yardımıyla elde edilen değerleri grafik halinde internet üzerinden elde etme etkinliği gerçekleştirilecektir. Eğitmen aşağıdaki resimde örneği verilmiş olan devreyi kurar ve devre elemanlarının bağlantılarını öğrencilere gösterir.



Resim 10.5: Gözle-Uygula Etkinliği Örnek Devre Şeması

Daha sonra Nesnelerin İnterneti (IoT) platformu kapsamında ölçümden elde edilecek ve Wi-fi modülü ile iletilecek verilerin görsel olarak gösterileceği “**ThingSpeak**” sitesine (<https://www.thingspeak.com>) üyelik adımları gerçekleştirilmelidir. Kayıt işlemleri sonunda ana sayfada bulunan ve aşağıdaki resimde görülen “Channels” bölümüne; daha sonra da “New Channel” butonuna tıklandığında yeni bir kanal açma bölümüne giriş yapılır.

New Channel

Name

Description

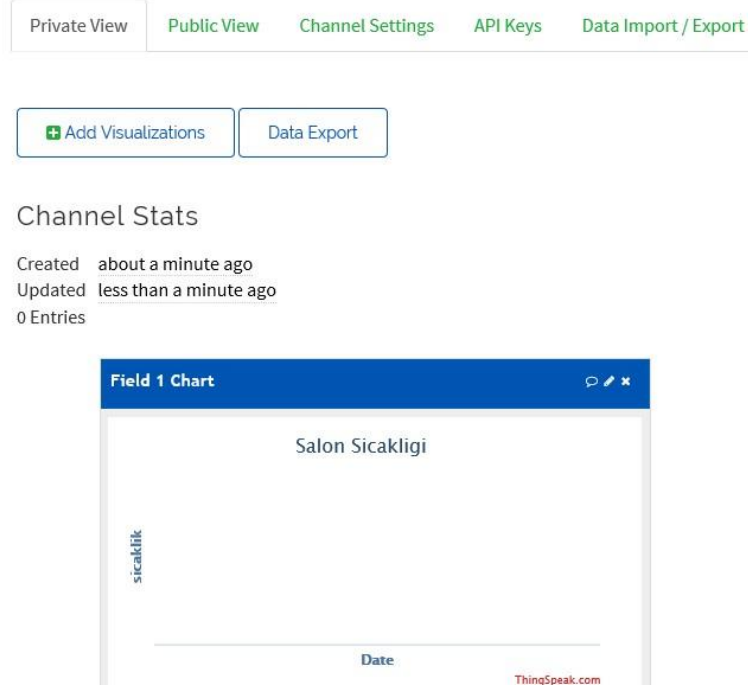
Field 1

Field Label 1



Resim 10.6: ThingsSpaeak IoT Platformuna Yeni Cihaz Ekleme

Bu bölümde yeni açılacak kanala “Name” yazan yere Türkçe karakterler kullanmadan bir isim verilir ve Field 1 kısmında “sicaklik”; Field 2 bölümüne ise “nem” yazıldıktan sonra “Save Channel” butonuna tıklanarak aşağıdaki resimde görülen verileri görüntüleyeceğimiz kanal oluşturulur.



Resim 10.7: ThingsSpaeak IoT Platformunda Cihaz Bilgilerini Görüntüleme

Yönlendirilen sayfada “Private View” bölümünde bir önceki adımda oluşturduğumuz sıcaklık ve nem grafiklerini Field1 ve Field 2 bölümlerinde görebiliriz. Bu sayfada oluşturulan kanal ile ilgili ayarlar, veri alımı ve aktarımı gibi işlemlerin yapılabileceği butonlar dışında “**API Keys**” isimli daha sonra Arduino IDE kodunda kullanılacak bölüm yer almaktadır. Bu sayfanın sol alt bölümünde yer alan “API Requests” başlığı altındaki “Write a Channel Feed” altındaki link ile Arduino UNO ve ESP8266 modülünün haberleşmesi sağlanacak olup bu API key açılan hesaba özel olarak Arduino IDE kodunda kullanılacaktır.

Devre bağlantıları kurulduktan ve “ThingSpeak” içerisinde kanal oluşturulduktan sonra Arduino IDE içerisinde yazılmış olan aşağıdaki resimde görülen kod öğrencilere gösterilir ve yazılmış kodlar üzerinde bilgi verilir. “**SoftwareSerial**” kütüphanesinin ekleneceği ve kütüphane içerisinde yazılacak kodlar hakkında açıklamalar yapılır. Ayrıca bu aşamada kodların altındaki tablodaki açıklamalar kullanılabilir.

```

DHT11_wifi
1 #include <SoftwareSerial.h>
2 #include <dht11.h>
3
4 String agismi = "baglanilacak_ag_ismi";
5 String agsifresi = "baglanilacak_ag_sifresi";
6
7 int rxPin = 10;
8 int txPin = 9;
9 int dht11Pin = 2;
10
11 String ip = "184.106.153.149";
12 float sicaklik, nem;
13
14 dht11 DHT11;
15 SoftwareSerial esp(rxPin, txPin);
16
17 void setup() {
18
19   Serial.begin(9600);
20   Serial.println("Started");
21   esp.begin(115200);
22   esp.println("AT");
23   //Serial.println("AT Yollandı.");
24   while(!esp.find("OK")){
25     esp.println("AT");
26     Serial.println("ESP8266 Bulunamadı.");
27
28     DHT11.read(dht11Pin);
29     sicaklik = (float)DHT11.temperature;
30     nem = (float)DHT11.humidity;
31     Serial.println(sicaklik);
32     Serial.println(nem);
33
34   }
35   Serial.println("OK Komutu Alındı");
36   esp.println("AT+CWMODE=1");
37   while(!esp.find("OK")){
38     esp.println("AT+CWMODE=1");
39     Serial.println("Ayar Yapılıyor...");
40   }
41   Serial.println("Client olarak ayarlandı");
42   Serial.println("Aga Bağlanılıyor...");
43   esp.println("AT+CWJAP="+agismi+"\","+agsifresi+"");
44   while(!esp.find("OK"));
45   Serial.println("Aga Bağlandı.");
46   delay(1000);
47 }
48
49 void loop() {
50
51   esp.println("AT+CIPSTART=\"TCP\", \""+ip+"\", 80");
52   if(esp.find("Error")){
53     Serial.println("AT+CIPSTART Error");
54   }
55
56   DHT11.read(dht11Pin);
57   sicaklik = (float)DHT11.temperature;
58   nem = (float)DHT11.humidity;
59
60   String veri = "GET https://api.thingspeak.com/update?api_key=F4620NI38X840KXC";
61   veri += "&field1=";
62   veri += String(sicaklik);
63   veri += "&field2=";
64   veri += String(nem);
65   veri += "\r\n\r\n";
66   esp.println("AT+CIPSEND=");
67   esp.println(veri.length()+2);
68   delay(2000);
69   if(esp.find(">")){
70
71     esp.print(veri);
72     Serial.println(veri);
73     Serial.println("Veri gönderildi.");
74     delay(1000);
75
76   }
77
78   Serial.println("Bağlantı Kapatıldı.");
79   esp.println("AT+CIPCLOSE");
80   delay(1000);
81 }
82

```

Resim 10.8: Gözle-Uygula Etkinliği Örnek Kodu

Wi-Fi Kodlarının Açıklaması	
Kullanılan Kodlar	Görevi
String agismi	Wi-Fi modülünün bağlanacağı ağ ismi
String agsifresi	Bağlanılacak olan ağ şifresi
String ip	ThingSpeak sitesinin ip adresi
“AT”	“AT” komutu modülün çalışıp çalışmadığını ve hazır olup olmadığını kontrol eder.
“AT+CWMODE=?”	“AT+CWMODE=?” modülü ile yapılacak uygulama türünün seçilmesini sağlar. Uygulamada STA (istasyon) modunda (1) çalışacak şekilde ayarlanacaktır. Böylelikle Wi-Fi modülü başka ağlara bağlanabilir.
“AT+CWJAP=”	“AT+CWJAP=”Wi-Fi adı”, ”Wi-Fi şifresi” ağa bağlanılmasını sağlar. Uygulamada programın başında tanımlanan ağ ismi ve ağ şifresi kullanılarak ağa bağlantı gerçekleştirilmektedir.
“AT+CIPSTART=”	TCP ya da UDP sunucusuna bağlanmak için kullanılır. Uygulamada 80 numaralı portu kullanarak “ThingSpeak” sitesinin ip adresi ile bağlantı gerçekleştirilir.
“AT+CIPSEND=”	Verilerin gönderilmesi için kullanılır.
“AT+CIPCLOSE”	Bağlantının sonlandırılmasını sağlar.

2. ADIM: TASARLA ve ÜRET

2.1 Tasarla- Mobil Uzay Keşif Robotu

Keşif robotları, mevcut ya da yeni keşfedilen gezegenlerin yüzeylerinde gezip, bilgi toplamak ve bu bilgileri Dünya'ya göndermek amacıyla tasarlanmaktadır. Şu ana kadar Ay'a ve Mars'a pek çok yolculuk yapan bu keşif robotları diğer gezegenlerde yaşam olup olmadığının anlaşılmasında büyük bir rol oynamaktadırlar ve gelecekte de önemli görevler gerçekleştireceklerdir. Temel görev olarak keşif robotları, araştırma yapacakları gezegene indikten sonra gezegenin jeolojisini, iklimini ve yüzeydeki kaya ve toprak formlarını toplamakta ve bunların analizini yapıp yaşama dair ipuçları bulmaya çalışmaktadır.

Bu etkinlikte öğrencilerden yeni keşfedilmiş bir gezegene gönderilen ve oradaki yaşam formlarını inceleyip dünyaya veri olarak gönderecek bir mobil uzay keşif robotu tasarlamaları istenmektedir. Tasarlanacak robotun çalışmaya başladıktan sonra şu adımları gerçekleştirmesi beklenmektedir:

1. Keşif robotu başlangıçta Wi-Fi modülü üzerinden dünya ile bağlantı kurar. Bu aşamada robotun “ESP 8266” üzerinden sadece ağa bağlanması yeterlidir.
2. Keşif robotu durarak önündeki ultrasonik sensör 90 dereceye kendini konumlandırır.
3. Keşif robotu önünde 20 cm'den yakın bir engel olmadığı durumda ilerlemeye başlar. Robotun önüne 20 cm'den daha yakın bir engel çıktığında öncelikle durup daha sonra sırasıyla, önce ultrasonik sensör sağa, sonra sola dönerek uzaklık ölçümü yaparak eğer varsa sağ ya da sola dönerek engelden kaçır. Eğer sağda ve solda engel var ise yarım saniye geriye gider.
4. Engel ve ilerleme sürecinden bağımsız olarak keşif robotu her 15 saniyede bir durarak Wi-Fi modülü aracılığıyla dünya ile bağlantı kurup bulunduğu gezegenin o andaki sıcaklık ve nem değerini gönderir.
5. Veri gönderim işlemi bittikten sonra keşif robotu önündeki engel durumuna göre 3. maddedeki şartları kontrol ederek ilerler.

2.2 Tasarla - Mobil Uzay Keşif Robotu Tasarımı

Öğrenciler geçen haftalarda oluşturmuş oldukları mobil robotun üzerine kuracakları devreleri gruplar halinde tartışır. Program kodlarının yazımında ve algoritmanın oluşturulmasında eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Uzay keşif robotunun fiziksel tasarımı nasıl olmalıdır? Farklı devre elemanları ve sensörler optimum sürüş ve veri alışverişini sağlamak için mobil robot üzerinde nasıl konumlandırılmalıdır?
- Keşif robotu birçok sensörden oluşacağı için devre kurulurken Arduino UNO üzerindeki PWM pinlerinin kullanımı nasıl sağlanmalıdır?
- Arduino IDE program algoritması nasıl kurulmalıdır? Hangi kütüphaneler ve nesnelere kullanılmalıdır?
- Arduino IDE içerisinde hangi fonksiyonlar kullanılabilir?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütülmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Keşif robotunun fiziksel tasarımında Arduino UNO ve diğer büyük devre elemanları üstte yer almalıdır.
- Hem devre bağlantıları için hem de Arduino UNO üzerinde GND ve 5 V girişlerini çoğaltmak için robotun üzerinde yer alacak şekilde küçük breadboard kullanılmalıdır.
- Arduino IDE içerisinde yazılacak programda veri gönderim işleminden sonra robotun fiziksel hareketinin başlayacağı algoritma içerisinde düşünülmelidir.
- Keşif robotunun engel ile karşılaştığında gerçekleştirmesi gerekli adımlar if-else yapısı ile kontrol edilebilir.
- Keşif robotunun temel hareketlerinde ve ultrasonik sensör ile uzaklık ölçümünde fonksiyonların kullanılması iş gücünü kısıltacaktır.

2.3 Üret- Mobil Uzay Keşif Robotu Tasarımı

Öğrenciler keşif robotunun yönergesini tamamlayabilecek şekilde tasarımlarını yaptıktan sonra bilgisayar başında Arduino IDE üzerinde kodları yazarlar. Öğrencilere eğitimci tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında farklı sensörlere ait kodların ve kütüphanelerin kullanımında ve çözüm için algoritmanın oluşturulma sürecinde eğitimcinin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenir.

Uzay_Araci

```

1 #include <Servo.h>
2 #include <SoftwareSerial.h>           //SoftwareSerial kütüphanimizi ekliyoruz.
3 #include <dht11.h>                   //DHT11 sensör kütüphanemizi ekliyoruz.
4
5 Servo myservo;
6 int Echo=A4;
7 int Trig=A5;
8 int in1=6;
9 int in2=7;
10 int in3=8;
11 int in4=4;
12 int ENA=5;
13 int ENB=11;
14 int hiz=150;
15 int saguzaklik=0, soluzaklik=0, ortauzaklik=0;
16
17
18 String agAdi = "Erman_note7";       //Ağımızın adını buraya yazıyoruz.
19 String agSifresi = "ylyo0157";     //Ağımızın şifresini buraya yazıyoruz.
20 int rxPin = 10;                     //ESP8266 RX pini
21 int txPin = 9;                      //ESP8266 TX pini
22 int dht11Pin = 2;
23 String ip = "184.106.153.149";     //Thingspeak ip adresi
24 float sicaklik, nem;
25 dht11 DHT11;
26
27 long previousMillis = 0;
28
29
30
31 SoftwareSerial esp(rxPin, txPin);    //Seri haberleşme pin ayarlarını yapıyoruz.
32
33 void ileri()
34 {
35   analogWrite(ENA,hiz);
36   analogWrite(ENB,hiz);
37   digitalWrite(in1,HIGH);
38   digitalWrite(in2,LOW);
39   digitalWrite(in3,LOW);
40   digitalWrite(in4,HIGH);
41 }
42
43 void geri()
44 {
45   analogWrite(ENA,hiz);
46   analogWrite(ENB,hiz);
47   digitalWrite(in1,LOW);
48   digitalWrite(in2,HIGH);
49   digitalWrite(in3,HIGH);
50   digitalWrite(in4,LOW);
51 }
52
53 void soladonus()
54 {
55   analogWrite(ENA,hiz);
56   analogWrite(ENB,hiz);
57   digitalWrite(in1,HIGH);
58   digitalWrite(in2,LOW);
59   digitalWrite(in3,HIGH);
60   digitalWrite(in4,LOW);
61 }
62
63 void sagadonus()
64 {
65   analogWrite(ENA,hiz);
66   analogWrite(ENB,hiz);
67   digitalWrite(in1,LOW);
68   digitalWrite(in2,HIGH);
69   digitalWrite(in3,LOW);
70   digitalWrite(in4,HIGH);
71 }
72

```

```

73 void durma() {
74   analogWrite(ENA, LOW);
75   analogWrite(ENB, LOW);
76 }
77
78 int uzaklik_testi() {
79   digitalWrite(Trig, LOW);
80   delayMicroseconds(2);
81   digitalWrite(Trig, HIGH);
82   delayMicroseconds(20);
83   digitalWrite(Trig, LOW);
84   float uzaklik=pulseIn(Echo, HIGH);
85   uzaklik=uzaklik/58;
86   return(int)uzaklik;
87 }
88 /*put your setup code here, to run once*/
89 void setup() {
90
91   Serial.begin(9600); //Seri port ile haberleşmemizi başlatıyoruz.
92   Serial.println("Started");
93   esp.begin(115200); //ESP8266 ile seri haberleşmeyi başlatıyoruz.
94   esp.println("AT"); //AT komutu ile modül kontrolünü yapıyoruz.
95   Serial.println("AT Yollandı.");
96   while(!esp.find("OK")){ //Modül hazır olana kadar bekliyoruz.
97     esp.println("AT");
98     Serial.println("ESP8266 Bulunamadı.");
99   }
100  Serial.println("OK Komutu Alındı");
101  esp.println("AT+CWMODE=1"); //ESP8266 modülünü client olarak ayarlıyoruz.
102  while(!esp.find("OK")){ //Ayar yapılana kadar bekliyoruz.
103    esp.println("AT+CWMODE=1");
104    Serial.println("Ayar Yapılıyor...");
105  }
106  Serial.println("Client olarak ayarlandı");
107  Serial.println("Aga Bağlanılıyor...");
108  esp.println("AT+CWLAP=\"" +agAdi+"\", \"" +agSifresi+"\""); //Ağımıza bağlanıyoruz.
109  while(!esp.find("OK")); //Ağa bağlanana kadar bekliyoruz.
110  Serial.println("Aga Bağlandı.");
111  delay(1000);
112
113
114  myservo.attach(3);
115  pinMode(Echo, INPUT);
116  pinMode(Trig, OUTPUT);
117  pinMode(in1, OUTPUT);
118  pinMode(in2, OUTPUT);
119  pinMode(in3, OUTPUT);
120  pinMode(in4, OUTPUT);
121  pinMode(ENA, OUTPUT);
122  pinMode(ENB, OUTPUT);
123  durma();
124 }
125 /*put your main code here, to run repeatedly*/
126 void loop() {
127
128  unsigned long currentMillis = millis();
129
130
131
132  if (currentMillis - previousMillis > 15000)
133  {
134    previousMillis = currentMillis;
135    durma();
136    esp.println("AT+CIPSTART=\"TCP\", \"" +ip+"\", 80"); //Thingspeak'e bağlanıyoruz.
137    if(esp.find("Error")){ //Bağlantı hatası kontrolü yapıyoruz.
138      Serial.println("AT+CIPSTART Error");
139    }
140    DHT11.read(dht11Pin);
141    sicaklik = (float)DHT11.temperature;
142    nem = (float)DHT11.humidity;
143

```

```

144 String veri = "GET https://api.thingspeak.com/update?api_key=VUR8JGUVHPZQ3GYX"; //Thingspeak komutu.
145 veri += "&field1=";
146 veri += String(sicaklik);
147 veri += "&field2=";
148 veri += String(nem); //Göndereceğimiz nem değişkeni
149 veri += "\r\n\r\n";
150 esp.print("AT+CIPSEND="); //ESP'ye göndereceğimiz veri uzunluğunu veriyoruz.
151 esp.println(veri.length()+2);
152 delay(2000);
153 if(esp.find(">")){ //ESP8266 hazır olduğunda içindeki komutlar çalışıyor.
154     esp.print(veri); //Veriyi gönderiyoruz.
155     Serial.println(veri);
156     Serial.println("Veri gönderildi.");
157     delay(1000);
158 }
159 Serial.println("Bağlantı Kapatıldı.");
160 esp.println("AT+CIPCLOSE"); //Bağlantıyı kapatıyoruz
161 delay(1000); //Yeni veri gönderimi için 1 dakika bekliyoruz.
162
163
164 }
165
166 myservo.write(90);
167 delay(50);
168 ortauzaklik=uzaklik_testi();
169
170 if(ortauzaklik<=20){
171     durma();
172     delay(500);
173
174     myservo.write(5);
175     delay(1000);
176     saguzaklik=uzaklik_testi();
177     delay(500);
178
179
180     myservo.write(180);
181     delay(1000);
182     soluzaklik=uzaklik_testi();
183
184
185     delay(500);
186     myservo.write(90);
187     delay(1000);
188
189 if(saguzaklik>soluzaklik){
190     sagadonus();
191     delay(500);
192 }
193 else if(saguzaklik<soluzaklik)
194 {
195     soladonus();
196     delay(500);
197 }
198 else if((saguzaklik<=20)|| (soluzaklik<=20))
199 {
200     geri();
201     delay(2000);
202 }
203 else
204 {
205     ileri();
206 }
207 }
208 else
209 ileri();
210 }

```

Resim 10.9: Tasarla-Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Keşif robotunda kullanılan sensör ve devre elemanlarının isimleri kâğıtlara yazılır ve tombala oyunu misali öğrencilerden bir kâğıt seçmeleri istenir:

- Öğrencilerden seçilen sensörlerin ve devre elemanlarının özelliklerini anlatmaları ve bu sensörlerin hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Keşif robotu etkinliğinde seçilen sensörlerin ve devre elemanlarının Arduino IDE içerisindeki kod yazımlarında karşılaşılan güçlükler nelerdi?
- Öğrencilerden şu aşamaya kadar öğrenilen tüm sensörlerin bir arada kullanılabileceği proje önerileri tartışmaları istenir.
- Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşmaya kadar devam ettirilir.

4. PROJEYE YÖNELİK FİKİR ÜRETME

Bu Hafta Gerçekleştirilecek Etkinlikler: Dersin sonunda öğretmen öğrencilerin proje kayıt defterlerinde bu haftaya ilişkin fikir üretme bölümüne ait soruları cevaplayıp cevaplamadıklarını kontrol eder. Öğrenci gruplarının bu bölümde yazmış oldukları tasarım ve programlamaya yönelik geliştirdikleri algoritmaların çözüme yönelik prototip öncesi önemli adım olduğu hatırlatılmalı ve grupların fikirleri tartışılmalıdır. Özellikle çözüme yönelik olası seçimlerin nasıl yapıldığı grup üyeleri tarafından eğitime aktarılmalıdır.

Ayrıca öğrencilere proje kayıt defterlerinde son hafta proje sergisi günü teslim edilecek proje ayrıntıları bölümüne “**video**” kaydı için tüm çalışmalarının dijital kayıtlarını almalarının gerekliliği hatırlatılmalıdır.

Sonraki Haftalara Hazırlık (11. Hafta): Gruplara gelecek hafta için belirledikleri projeler için prototip geliştirme adımının gerçekleştirileceği hatırlatılmalıdır. Proje kayıt defterinde ilgili yerlerin geliştirilecek prototipin özelliklerini içermesinin ve çözümlerine yönelik hedef kitle değerlendirmelerinin açıkça yazılmasının gerekliliği vurgulanmalıdır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki beklentiler nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI (Tanımlama – 9. HAFTA)**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

○ **ÇÖZÜM ÖNERİSİ TASLAĞI (Fikir Üretme – 10. HAFTA)**

- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?

- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?

- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?

○ **PROTOTİP GELİŞTİRME (Prototip Geliştirme – 11. HAFTA)**

- Geliştirdiğiniz prototip tanımlanmış olan problemin hangi özelliklerini çözmektedir?

- Geliştirdiğiniz prototip üzerinde değiştirmek/çözümünüzde geliştirmek istediğiniz noktalar nelerdir?

11.Bölüm- Sensörler ile IOT Uygulamalarına Giriş

Ön bilgi:

- Öğrenciler algoritma mantığını ve temel programlama kavramlarını bilir.
- Öğrenciler basit metin tabanlı programlamaya yönelik söz dizimi denetimi, yazımı, okuması ve hata ayıklaması yapabilir.
- Öğrenciler Arduino UNO'yu tanır ve bunu kullanarak devreleri kurup kodlarında düzenleme yapabilirler.
- Öğrenciler Arduino IDE arayüzü ile değişkenler, "if/else, for" döngüsü, dizi vb. yöntemleri ile kod yazarak elektronik devre elemanlarını yönetebilir.

Bölüm Kazanımları

- Tilt, hareket (PIR), yağmur, gaz sensörleri ile "RFID" kart okuyucu ve Keypad devre elemanlarının görevlerini ve çalışma prensiplerini bilir.
- Tilt, hareket (PIR), yağmur, gaz sensörleri ile "RFID" kart okuyucu ve Keypad devre elemanları ile geliştirilecek uygulamaları Arduino IDE üzerinde programlayabilir.
- Tilt, hareket (PIR), yağmur, gaz sensörleri ile "RFID" kart okuyucu ve Keypad devre elemanlarını geliştirilecek projelerin devresini kurabilir ve Arduino IDE üzerinde programlayabilir.

Haftanın Amacı

Bu haftanın amacı, öğrencilerin öncelikle tilt, hareket (PIR), yağmur, gaz sensörleri ile "RFID" kart okuyucu ve "Keypad" devre elemanları hakkında bilgi sahibi olmasıdır. Ayrıca bu hafta ile öğrencilere yukarıda belirtilen sensörlerin ve devre elemanlarının farklı etkinlikler ve projelerde kullanılması ile ilgili gerekli yeterliliklerin öğrencilere kazandırılması amaçlanmaktadır. Bu kapsamda programlamada farklı kontrol yapıları, değişken ve değer kullanımları konularının uygulamalar içerisindeki deneyimlemeleri de hedefler içerisinde yer almaktadır.

Gözle ve Uygula Adımlarında Kullanılacak Malzemeler:

Malzeme Listesi	
Arduino UNO	LDR Işık Sensörü
Breadboard	I2C Entegreli 2x16 LCD Ekran
Eğim (Tilt) Sensörü	Kırmızı LED
Gaz Sensörü	Buzzer
RC522 RFID NFC Kiti	Erkek-Erkek bağlantı kablosu
Yağmur Sensörü	Erkek-Dişi bağlantı kablosu
PIR Hareket Sensörü	330 ohm direnç
3x4 Keypad	220 ohm direnç

Haftanın İşlenişi:

Gözle: Tilt, hareket (PIR), yağmur, gaz sensörleri ile RFID kart okuyucu ve Keypad devre elemanlarının tanımlarının yapılması ve özelliklerinin verilmesi.

Uygula: Tilt, hareket (PIR), yağmur, gaz sensörleri ile RFID kart okuyucu ve Keypad devre elemanları ile örnek devrelerin oluşturulması ve programlanması.

Tasarla: Tilt, hareket (PIR), yağmur, gaz sensörleri ile RFID kart okuyucu ve Keypad devre elemanlarını kullanarak nükleer santraller için güvenlik protokolü tasarımının gerçekleştirilmesi.

Üret: Tilt, hareket (PIR), yağmur, gaz sensörleri ile RFID kart okuyucu ve Keypad devre elemanlarını kullanarak nükleer santraller için güvenlik protokolü tasarımının gerçekleştirilme; devrelerini oluşturma ve Arduino IDE içerisinde program kodlarının yazılması.

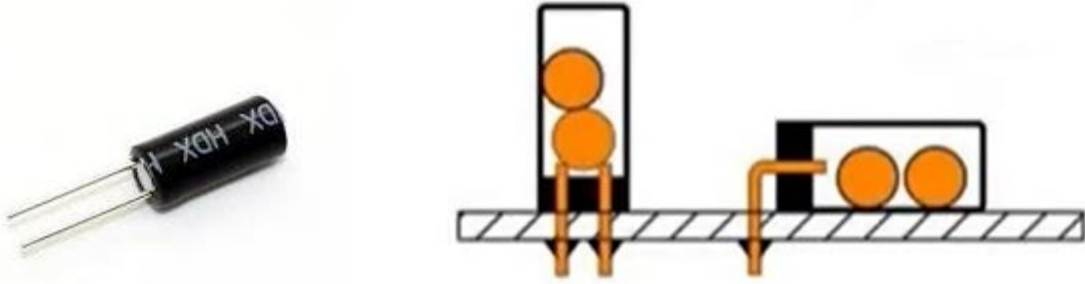
Değerlendir: Yansıtma Etkinliği.

1. ADIM: GÖZLE ve UYGULA

1.1 Gözle ve Uygula – Eğim (Tilt) Sensörü

Malzeme Listesi	
Arduino UNO	Tilt (Eğim) Sensörü
Breadboard	Buzzer
Bağlantı kabloları	Kırmızı LED
220 ohm direnç	

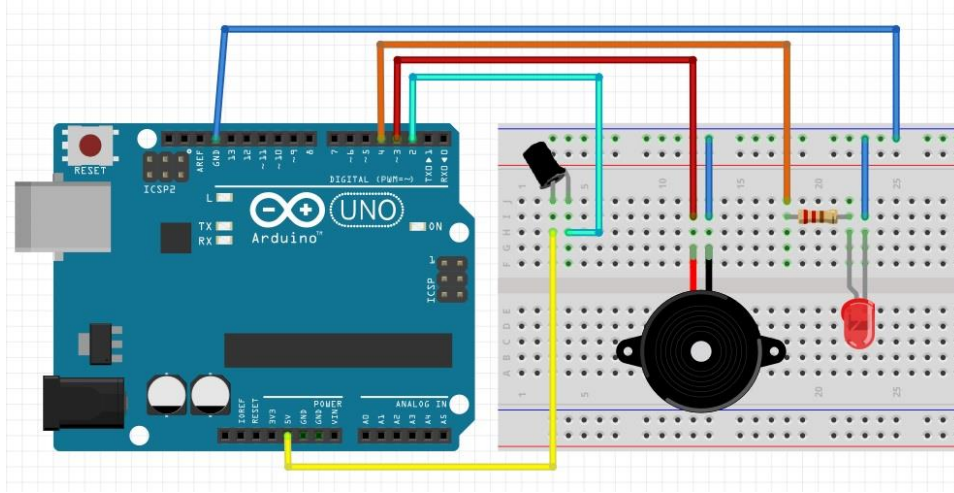
Tilt sensörü (eğim anahtarı) eğimin yönünü tespit etmek için kullanılan basit ve kullanışlı bir sensördür. Bu sensörler düşük maliyetli ve düşük güç tüketimlidirler. Basit yapıları nedeniyle oyuncaklar ve basit uygulamalarda sıklıkla tercih edilirler. Arduino başta olmak üzere birçok mikrodenetleyici platformu ile kullanılabilir. Dijital çıkış veren bir sensördür. Çoğunlukla “civa anahtarları”, “tilt anahtarları” ya da “döner top sensörleri” olarak adlandırılırlar. Aşağıdaki resimde tilt sensörü ve iç yapısı görülmektedir.



Resim 11.1: Tilt Sensörü ve İç Yapısı

Tilt sensörleri genellikle içi oyuk silindirik şekillerde olurlar. Silindirik olmayan şekilleri de vardır. Bu oyğun içinde serbest hareket eden iletken bir madde bulunur. Bu madde bir civa damlası veya döner bir top olabilir. Oyğun bir ucunda iki adet iletken oluşmuş kutuplar bulunur. Sensör dik hale geldiğinde içindeki küçük bilye çiftleri birbirleri ile temas ederek sensördeki iki uç arasındaki iletimi sağlayarak devrenin tamamlanmasını sağlar. Bu şekilde 5V’luk dijital sinyal mikrodenetleyiciye ulaşır ve sensörün dik durumda olduğu bilgisi alınmış olur. Sensör eğildiğinde yani yatay hale geldiğinde ise iletken hareketli maddeler uç kısımdan uzaklaşarak iki iletken ayrılırlar ve akımın geçmemesine sebep olurlar. Sensör içindeki bilye çok hareketli olduğu için yatay konumlandırmadan ziyade açılı veya dikey halde kullanılması önerilmektedir.

Eğitmen Tilt (eğim) sensörünü, çalışma prensibini ve yapısını öğrencilere tanıttıktan sonra; öğrencilerle beraber Tilt (eğim) sensörünü kullanarak bir eğim alarm devresi kurarak ve buna uygun kodu yazacakları gözle ve uygula etkinliğine devam edeceklerdir. Bu etkinlikte eğim sensörü dikey konumdan eğik konuma getirildiğinde sesli ve ışıklı uyarı veren bir sistem oluşturulacaktır. Sensörün takıldığı zemin eğik duruma geçtiğinde devreye bağlanan buzzer ve LED 1 saniye yanıp 0,5 saniye sönen bir döngü ile uyarı verecektir. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 11.2: Gözle-Uygula Etkinliği Örnek Devre Şeması

Yukarıdaki devreyi eğitmen önce kendisi kurar ve sonrasında devrenin çalışma mantığını öğrencilere aktararak onların da aynı devreyi kurmalarını sağlar. Eğitmen öğrencilerin kendi devrelerini kurmaları için yeterli süreyi verir. Bu süreçte öğrencileri gözlemler, gerektiğinde öğrencilerin devreyi kurmalarına yardımcı olur.

Devre kurulumu tamamlandıktan sonra aşağıdaki resimde örneği verilen kodlar ile devrede sensör eğildiğinde 1 saniye yanan 0,5 saniye sönen bir döngü ile alarm veren bir eğim sensör uygulaması tamamlanmış olur. Eğitmen öğrencilerin de kendi kod bloklarını yazmalarını sağlar. Burada eğitmen tarafından kodların öğrencilere direk yansıtılması yerine, öğrencilerin kendi bilgi ve becerileri ile uygun kod satırlarını yazmaları sağlanmalıdır. Eğitmen gerekli noktalarda öğrencilere direk çözümü vermeden yardımcı olabilir. Son olarak yazılan kod satırları Arduino UNO kartlarına yüklenerek eğim alarmı uygulamasının sorunsuz bir şekilde çalıştığı kontrol edilir.


```

tilt_egim_sensor
1 #define egim_sensor 2
2 #define buzzer 3
3 #define kirmizi_LED 4
4
5
6 void setup() {
7   pinMode(egim_sensor, INPUT);
8   pinMode(buzzer, OUTPUT);
9   pinMode(kirmizi_LED, OUTPUT);
10 }
11
12 void loop() {
13   int deger = digitalRead(egim_sensor);
14   if( deger==LOW){
15     digitalWrite(buzzer,HIGH);
16     digitalWrite(kirmizi_LED, HIGH);
17     delay(1000);
18     digitalWrite(buzzer,LOW);
19     digitalWrite(kirmizi_LED, LOW);
20     delay(500);
21   }
22   else{
23     digitalWrite(buzzer,LOW);
24     digitalWrite(kirmizi_LED, LOW);
25   }
26 }
27

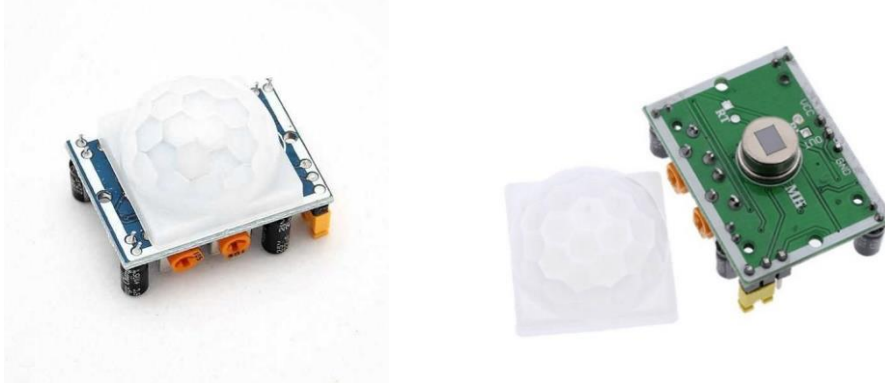
```

Resim 11.3: Gözle-Uygula Etkinliği Örnek Kodu

1.2 Gözle ve Uygula – Hareket (PIR) Sensörü (Öğrenci 2)

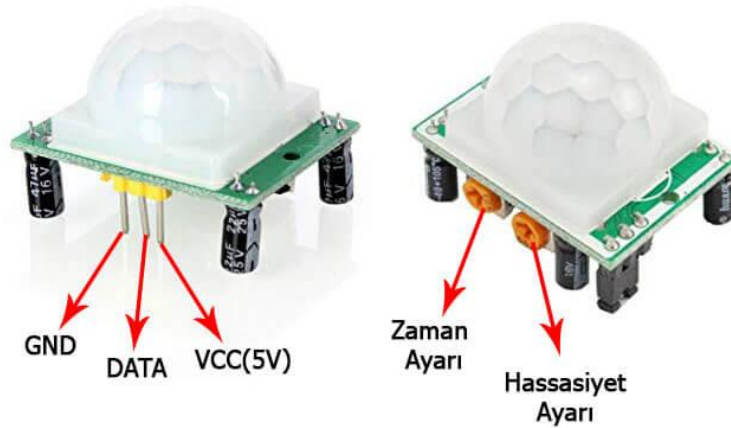
Malzeme Listesi
Arduino UNO
Breadboard
Hareket (PIR) Sensörü
Bağlantı kabloları
Kırmızı LED
220 ohm direnç

Günümüzde apartmanların merdiven boşluklarındaki harekete duyarlı aydınlatmalarda, güvenlik sistemlerinde ve garaj kapılarında sıklıkla karşılaştığımız Hareket (PIR) sensörlerinin bir türü olan “**HC-SR501 Hareket Sensörü**” (**PIR - Passive Infra-Red**) düşük güç tüketimli ve düşük maliyetli, oldukça sağlam, geniş bir lens yelpazesine sahip bir elektronik devre elemanıdır. Sensörün isimlendirilmesinde kullanılan PIR ismi Passive Infra-Red (Pasif Kızılötesi) kelimelerinin kısaltmasından oluşmaktadır. İsminden de anlaşılacağı gibi sensör kızılötesi dalgalarla çalışmaktadır. PIR sensörü termal enerjiye duyarlı kızılötesi sinyallerle ortamdaki ısı farkını algılar ve canlı bir cisim hareket ettiğinde ısı farkından dolayı hareketi belirler. Aşağıdaki resimde PIR sensörü görülmektedir.



Resim 11.4: PIR Sensörü

PIR hareket sensörleri; *Pyro elektrik sensör* ve kızılötesi sinyalleri pyro elektrik sensöre odaklayan *Fresnel (yakın plan) Lens* adı verilen özel bir lens olmak üzere iki ana bölümden oluşur. Fresnel lens sayesinde ortamdaki nesnelere gelen ışınlar sensörün odaklanmasını sağlar. Ortamda bir dalga değişimi olduğunda sensör algılama işlemi gerçekleştirir. PIR hareket sensörleri güç girişi (**VCC**), toprak hattı (**GND**) ve dijital çıkış (**DATA/OUT**) pinleri aracılığıyla Arduino UNO kartlarına bağlanırlar. Aşağıdaki resimde PIR sensörü ayak yapısı ve ayarları görülmektedir.

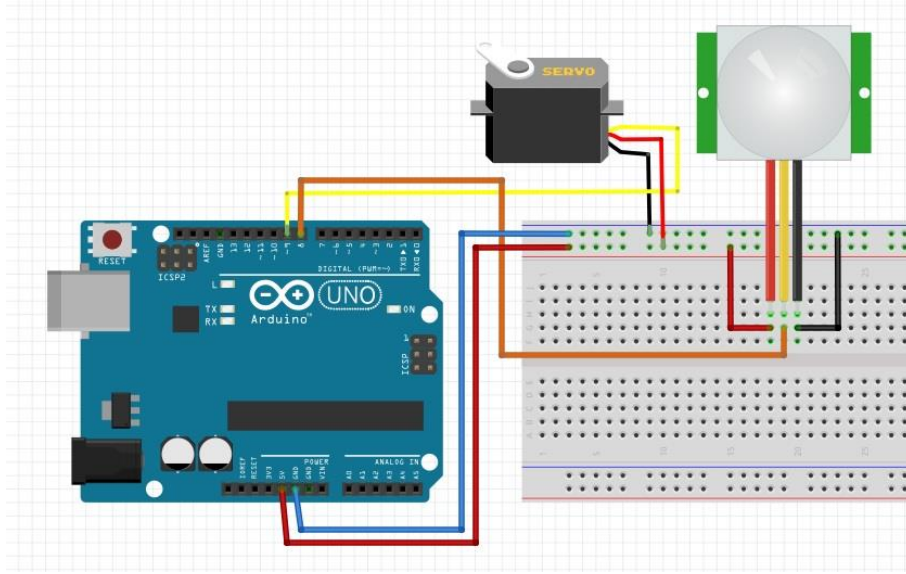


Resim 11.5: PIR Sensörü Ayak Yapısı ve Ayarları

Ayrıca PIR Hareket sensörünün üzerinde *hassasiyet* ve *zaman* ayarlarının yapılabileceği iki adet potansiyometre bulunmaktadır. Hassasiyet ayarı ile hareketin algılanabileceği maksimum mesafeyi ayarlar. Bu ayarlama ile 3 metre ile 7 metre arasında hareketin algılanacağı mesafe ayarlanabilir. Zaman ayarı ile ise çıkış sinyalinin hareketin algılanmasından sonra HIGH kalacağı zamanı belirler. 3 saniye ile 300 saniye arasında bir ayarlama yapılabilir.

Eğitmen PIR (hareket) sensörünü, çalışma prensibini ve yapısını öğrencilere tanıttıktan sonra; öğrencilerle beraber PIR (hareket) sensörünü kullanarak harekete duyarlı bir otomatik kapı devresi kuracak ve buna uygun kodu yazacakları gözle ve uygulama etkinliğine devam edecektir. Bu etkinlikte hareket sensörü kullanılarak herhangi bir nesne algılanmadığında servo motora bağlı kapının yatay konumda (kapalı) kalacağı ve Serial ekranda “Kimse yok...” mesajının yazdırılacağı; hareket sensörü bir nesne/hareket algıladığında ise servo motora bağlı kapının dikey konumda (açık) olacağı ve Serial ekranda “Birisini var...” mesajının yazdırılacağı bir

sistem oluşturulacaktır. Kapının her harekette kapalı ve açık kalacağı süre ise 2,5 saniye olarak ayarlanmalıdır. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 11.6: Gözle-Uygula Etkinliği Örnek Devre Şeması

Yukarıdaki devreyi eğitmen önce kendisi kurar ve sonrasında devrenin çalışma mantığını öğrencilere aktararak onların da aynı devreyi kurmalarını sağlar. Eğitmen öğrencilerin kendi devrelerini kurmaları için yeterli süreyi verir. Bu süreçte öğrencileri gözlemler, gerektiğinde öğrencilerin devreyi kurmalarına yardımcı olur.

Devre kurulumu tamamlandıktan sonra aşağıdaki resimde örneği verilen kodlar ile devrede sensör hareket algıladığında kapının açılacağı ve 2,5 saniye süre boyunca hareket algılanmazsa kapının kapanacağı bir sistem tamamlanmış olur. Eğitmen öğrencilerin kendi kod bloklarını yazmalarını sağlar. Burada eğitmen tarafından kodların öğrencilere direkt yansıtılması yerine, öğrencilerin kendi bilgi ve becerileri ile uygun kod satırlarını yazmaları sağlanmalıdır. Eğitmen gerekli noktalarda öğrencilere direkt çözümü vermeden yardımcı olabilir. Son olarak yazılan kod satırları Arduino UNO kartlarına yüklenerek harekete duyarlı otomatik kapı uygulamasının sorunsuz bir şekilde çalıştığı kontrol edilir.

```

PIR_Hareket_Sensor
1  #include <Servo.h>
2
3  #define PIR_pin 8
4  #define servo_pin 9
5
6  int hareket;
7
8  Servo kapi;
9
10 void setup() {
11     Serial.begin(9600);
12     kapi.attach(servo_pin);
13     pinMode(PIR_pin, INPUT);
14 }
15
16 void loop() {
17
18     hareket = digitalRead(PIR_pin);
19     if(hareket == HIGH) {
20         kapi.write(120);
21         Serial.println("Biris var...");
22         delay(2500);
23     }
24     else{
25         kapi.write(50);
26         Serial.println("Kimse yok...");
27         delay(2500);
28     }
29 }

```

Resim 11.7: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

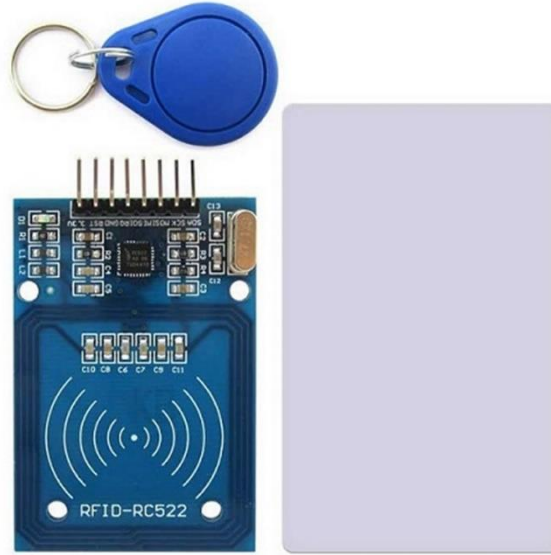
PIR Hareket Sensörü üzerindeki bağlantı pinlerinin isimleri sensör üzerindeki Fresnel lensi çıkarılarak görülebilir.

Ayrıca, PIR Hareket Sensörü üzerindeki hassasiyet ve zaman ayar potansiyometreleri ile sistemin doğru çalışması için gerekli ayarlamalar yapılmalıdır.

1.3 Gözle ve Uygula – RFID Kart Okuyucu ile Elektronik Kapı Kilidi Oluşturma (Öğrenci 1)

Malzeme Listesi
Arduino UNO
Breadboard
Bağlantı kabloları
RFID Kiti (Okuyucu, Kart ve Anahtarlık)
Servo Motor

“Radio Frequency Identification” – Radyo Frekansı Tanımlama (RFID) teknolojisi günümüzde elektronik kapılarda, toplu taşıma biletlerinde, işyeri ve okul girişlerindeki turnikelerde sıklıkla karşımıza çıkmaktadır. Radyo dalgalarının kullanıldığı RFID teknolojilerinde verileri kısa mesafelerde aktarmak için elektromanyetik alanlar kullanılır. RFID teknolojileri ile kişileri tanımlamak, tanımlı olmayan kişilerin depo gibi yerlere girmesini engellemek, giriş – çıkış saatlerini kayıt altında tutmak gibi birçok işlem gerçekleştirilebilir. Arduino ile RFID kitlerini kullanarak kapı kilidi, bir ortamda bulunan kişi sayısı bilgisinin alınması, bilgisayar oturum kilidi, alarm sistemi gibi projeler yapılabilir. Deneyap setlerinde “**RC522 RFID Kiti**” (Okuyucu, kart ve anahtarlık) bulunmaktadır. Bu kart okuyucu modülü Arduino, Raspberry Pi gibi birçok mikrodenetleyici ile sorunsuz bir şekilde çalışabilmektedir. RFID sistemi iki ana bileşenden oluşur. Bunlar aşağıdaki resimde görüldüğü gibi sorgulayıcı (okuyucu) olarak bilinen bir Alıcı-Verici devre parçası ve tanımlanacak bir nesneye bağlı bir aktarıcı (kart veya anahtarlık) olarak sıralanabilir.



Resim 11.8: RC522 RFID Kiti

RFID okuyucu, bir Radyo Frekansı modülü ve yüksek frekanslı elektromanyetik alan üreten bir antenden oluşur. RFID kart veya anahtarlık ise aslında pasif bir RFID cihazdır. Enerjisiz çalışmasına en olanak tanıyan özelliği bilgiyi depolayan ve işleyen bir mikroçip ve bir sinyal

almak ve iletmek için bir anten içermesidir. “RC522 RFID” modülü 13,56 MHz NFC Kart ve 13,56 MHz NFC anahtarlık ile yakın algılama özelliği kullanılarak mikrodenetleyiciye bağlı birçok elektronik devre elamanı kontrol edilebilir. RFID okuyucu, maksimum 10Mbps veri hızına sahip 4 pinli Seri Çevresel Arabirim (SPI) üzerinden bir mikro denetleyici ile iletişim kurabilir. Ayrıca I2C ve UART protokolleri üzerinden de iletişimi destekler.

RC522 RFID Kart Okuyucu Özellikleri:

- Giriş Voltajı : 3.3 V
- Frekans : 13,56 MHz
- Bağlantı Şekli : SPI, I2C, UART
- Mak. Çalışma Akımı : 13-26 mA
- Okuma Aralığı : 5 cm

RFID kitlerinde bulunan her kart ve anahtarlık nesnelere ait UID isimli bir numarası vardır. Okuyucumuza kartımızı veya anahtarlığımızı yaklaştırdığımızda bu numara okunarak işlem yapılır. Bu numara, her nesne için farklıdır. RFID kart okuyucuları bu nesnelere kart numaralarını elektromanyetik alanları kullanarak kısa mesafelerde okur ve bu numarayı mikrodenetleyiciye aktararak gerekli karşılaştırmaların yapılması sonucu istenen işlemlerin yapılıp yapılmamasını sağlar.

Dikkat

RC522 RFID kart okuyucunun Arduino IDE arayüzünde çalışabilmesi için güncel kütüphanesinin Arduino IDE’ye eklenmesi gerekmektedir. Daha önceki derslerde çeşitli elektronik devre elemanlarının kütüphanesinin Arduino IDE arayüzüne eklenmesi anlatıldığı gibi, RC522 RFID kit için **MFRC522 kütüphanesi** arayüzden veya internetten bulunarak eklenmelidir.

İnternette indirilen MFRC522 kütüphanesini aşağıdaki sıralamayı takip ederek Arduino IDE’ye ekleyebilirsiniz.

- *Taslak>library ekle>.ZIP Kitaplığı Ekle>ZIP dosyasını bilgisayarda indirdiğiniz yeri seçme>Aç*

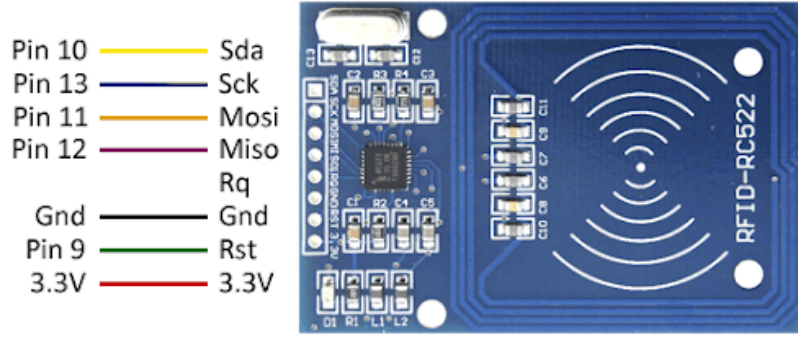
Aşağıdaki tabloda RFID kart okuyucu üzerindeki pinlerin görevleri ve Arduino UNO üzerinde hangi pinlere bağlanması gerektiği ile ilgili bilgilere yer verilmiştir. Eğitmen bunları uygulamaya geçildiğinde öğrencilere açıklamalarda bulunarak aktarabilir.

RC522 RFID – Arduino UNO Pin Bağlantı Şeması		
RC522 RFID Pini	İşlevi	Arduino UNO Pini
SDA	Sinyal girişi	Dijital (10)
SCK	SPI seri saat	Dijital (13)
MOSI	SPI girişi	Dijital (11)
MISO	SPI çıkışı	Dijital (12)
IRQ	Kesme	-
GND	Toprak Hattı	GND
RST	Sıfırlama / Kapatma	Dijital (9)
3.3 V	Enerji / Güç	3.3 V

Dikkat

RC522 RFID kiti 3,3 V ile çalışır. Bu sebeple daha yüksek voltaj vermemeniz gerekir.

Aşağıdaki resimde kitin ayak bağlantıları görülmektedir.

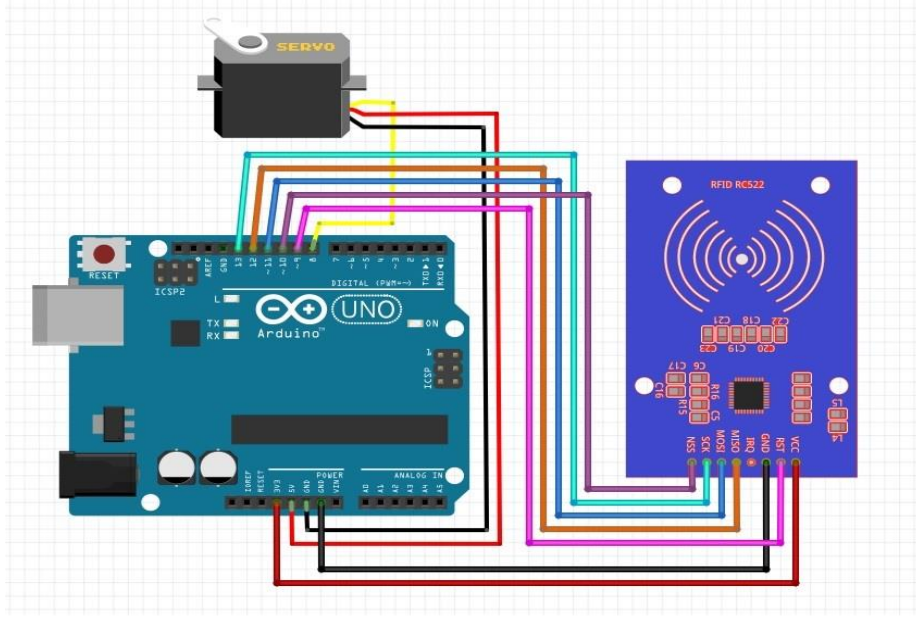


Resim 11.9: RC522 RFID Kiti Ayak Bağlantıları

Bu etkinlikte RFID okuyucuya kart veya anahtarlık okutulduğunda;

- eğer yetki verilmiş numaraya sahip kart veya anahtarlık ise servo motora bağlı kapının açılacağı ve Serial ekranda “Kapı açılıyor...” mesajının yanısıra kart veya anahtarlık numarasının da yazdırılacağı,
- eğer okutulan kart veya anahtarlık yetki verilmemiş kart veya anahtarlık ise servo motora bağlı kapının kapalı konumda kalmaya devam edeceği ve Serial ekranda “!!! Yanlış Kart !!!” mesajının altında kart veya anahtarlık numarasının da yazdırılacağı bir sistem oluşturulacaktır.

Doğru kart veya anahtarlık okutulduktan sonra kapı 5 saniye süre ile açık kalmalı ve bu süre sonunda otomatik olarak kapanmalıdır. Ayrıca her kart okutulduktan sonra 1 saniye beklemeden başka bir kart okunmamalıdır. Etkinlikle ilgili örnek devre şeması aşağıdaki resimde görülmektedir.



Resim 11.10: Gözle-Uygula Etkinliği Örnek Devre Şeması

Yukarıdaki devreyi eğitmen önce kendisi kurar ve sonrasında devrenin çalışma mantığını öğrencilere aktararak onların da aynı devreyi kurmalarını sağlar. Eğitmen öğrencilerin kendi devrelerini kurmaları için yeterli süreyi verir. Bu süreçte öğrencileri gözlemler, gerektiğinde öğrencilerin devreyi kurmalarına yardımcı olur.

Devre kurulumu tamamlandıktan sonra aşağıda resimde örneği verilen kodlar ile doğru kart veya anahtarlık okutulduğunda kapının açılacağı ve 5 saniye süre sonunda kapının otomatik olarak kapanacağı bir sistem tamamlanmış olur. Eğitmen öğrencilerin de kendi kod bloklarını yazmalarını sağlar. Burada eğitmen tarafından kodların öğrencilere direk yansıtılması yerine, öğrencilerin kendi bilgi ve becerileri ile uygun kod satırlarını yazmaları sağlanmalıdır. Eğitmen gerekli noktalarda öğrencilere direk çözümü vermeden yardımcı olabilir. Son olarak yazılan kod satırları Arduino UNO kartlarına yüklenerek RFID Kart veya anahtarlık ile otomatik kapı uygulamasının sorunsuz bir şekilde çalıştığı kontrol edilir.

```

rfid_otomatik_kapi
1 #include <MFRC522.h> //RFID okuyucu kütüphanesini ekliyoruz.
2 #include <Servo.h> //Servo motor kütüphanesini ekliyoruz.
3 #include <SPI.h> //SPI haberleşme kütüphanesini ekliyoruz.
4
5 #define RST_PIN 9 //RFID okuyucunun Reset pinini 9. pine tanımlıyoruz.
6 #define SS_PIN 10 //RFID okuyucunun SDA pinini 10. pine tanımlıyoruz.
7 #define Servo_PIN 8 //Servo motorun veri pinini 8. pine tanımlıyoruz.
8
9 Servo kapi; //Servo motoru kapi adında bir nesne olarak tanımlıyoruz.
10 MFRC522 rfid(SS_PIN, RST_PIN); //RFID okuyucuyu rfid adında bir nesne olarak tanımlıyoruz.
11 byte ID[4] = {196,254,80,131}; //Doğru kartın numarasını byte türünde ID dizisine tanımlıyoruz.
12
13 void setup() {
14   kapi.attach(Servo_PIN); //Servo motorumuzu attach komutu ile hangi pinde olduğunu atıyoruz.
15   Serial.begin(9600); //Serial monitörde kart numaralarını görebilmemiz için serial haberleşmeyi başlatıyoruz.
16   SPI.begin(); //Arduino ile RFID okuyucunun haberleşmesi için SPI haberleşmeyi başlatıyoruz.
17   rfid.PCD_Init(); //RFID okuyucuyu başlatıyoruz.
18 }
19
20 void loop() {
21
22   if(!rfid.PICC_IsNewCardPresent()) //RFID okuyucuya kart veya anahtarlık gösterilmediğinde programın başa dönmesini sağlıyoruz.
23     return;
24   if(!rfid.PICC_ReadCardSerial()) //RFID okuyucu okuma yapmadığında programın başa dönmesini sağlıyoruz.
25     return;
26   if(rfid.uid.uidByte[0] == ID[0] && //RFID okuyucuya okutulan kart veya anahtarlığın numarasını kontrol ediyoruz.
27       rfid.uid.uidByte[1] == ID[1] &&
28       rfid.uid.uidByte[2] == ID[2] &&
29       rfid.uid.uidByte[3] == ID[3]) {
30     Serial.println("Kapi acildi..."); //Eğer doğru kart okutuldu ise Serial Monitöre kapi açıldı yazısı yazdırıyoruz.
31     ekranayazdir(); //Okunan kart veya anahtarlığın numarasını aşağıda tanımlanan ekrana yazdır fonksiyonunu ile yazdırıyoruz.
32     kapi.write(180); //Kapi isimli servo motorumuzu 180 dereceye getirerek kapımızı açıyoruz.
33     delay(5000); //Kapiyı 5 saniye süre ile açık bekletiyoruz.
34     kapi.write(0); //Kapiyı tekrar kapatıyoruz.
35     Serial.println("Kapi kapaniyor..."); //Serial Monitöre kapi kapanıyor yazdırıyoruz.
36     delay(1000); //Yeni kart okumadan 1 saniye bekleme yapmasını sağlıyoruz.
37   }
38   else{
39     Serial.println("!!! Yanlis Kart !!!"); //Eğer doğru kart okutulmadı ise Serial Monitöre yanlış kart yazısını yazdırıyoruz.
40     ekranayazdir(); //Okunan kart veya anahtarlığın numarasını aşağıda tanımlanan ekrana yazdır fonksiyonunu ile yazdırıyoruz.
41   }
42   rfid.PICC_HaltA(); //RFID okuyucuyu yeni kart okuyana kadar durduruyoruz.
43 }
44
45 void ekranayazdir() { //ekranayazdir fonksiyonumuzu oluşturuyoruz.
46   Serial.print("ID Numarasi: "); //Serial Monitöre ID Numarası yazdırıyoruz.
47   for(int i=0; i<4; i++){ //Anahtarlık veya kartın numarasını 4 bölümden oluştuğu için for döngüsünde 4 defa aynı işlemi yaptırıyoruz
48     Serial.print(rfid.uid.uidByte[i]); //Anahtarlık veya kartın numarasının her bölümünü yazdırıyoruz.
49     Serial.print(" "); //Her byte arasına boşluk yazdırıyoruz.
50   }
51   Serial.println(""); //Her kart okumasından sonra da alt satıra geçmesini sağlıyoruz.
52 }
53

```

Resim 11.11: Gözle-Uygula Etkinliği Örnek Kodu

Dikkat

Doğru kart veya anahtarlık numarasının belirlenebilmesi için öncelikle kod satırlarında karşılaştırma için doğru numara “0,0,0,0” yazılabilir. Sonrasında doğru kabul edilecek kart veya anahtarlık RFID okuyucuya yaklaştırılarak Serial Monitörden numarasına bakılarak kod satırlarındaki karşılaştırma yapılan numara doğru kart veya anahtarlığın numarası yazılarak güncellenmelidir.

1.4 Gözle ve Uygula – Keypad Kullanımı (Öğrenci 2)

Malzeme Listesi

Arduino UNO

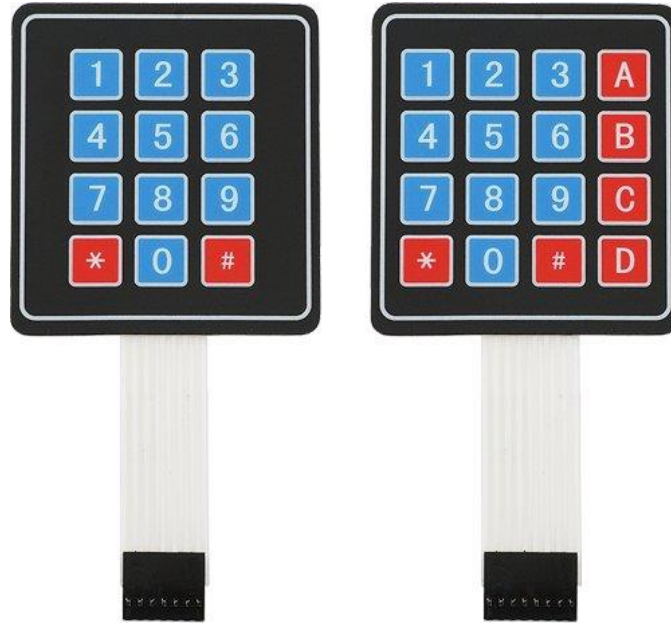
Breadboard

Bağlantı kabloları

4x3 Keypad

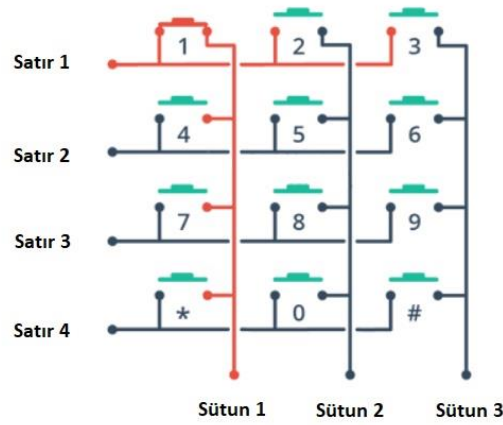
Şifre ile çalışan birçok ürün ve teknolojide **keypad (tuş takımı)** kullanılmaktadır. Özellikle günümüzde kurumların ve apartmanların otomatik kapılarının şifre ile açılmasında şifrenin kullanıcılar tarafından sistemlere girilmesinde tuş takımları kullanılmaktadır. Ayrıca sistemlere numerik bilgi girişi yapmak gerektiğinde de kullanılmaktadırlar.

Arduino ile kullanılan tuş takımlarından en çok aşağıdaki resimlerde görülen 4 satır “- 4” sütundan oluşan 16 tuşlu (4x4) ve 4 satır “- 3” sütundan oluşan 12 tuşlu (4x3) membran çeşitleri projelerde tercih edilmektedir. Membran tuş takımları çoğu proje için idealdir. Çünkü incedirler ve arka kısımlarındaki yapışkanlı yüzey sayesinde birçok yere adapte edilebilirler. Deneyap Atolyeleri Arduino setlerinde 4x3 tuş takımı bulunduğundan dolayı burada 4x3 tuş takımının yapısı ve kullanımı anlatılacaktır. Ancak 4x4 tuş takımının yapısında ve kullanımında çok büyük farklılıklar bulunmamaktadır.



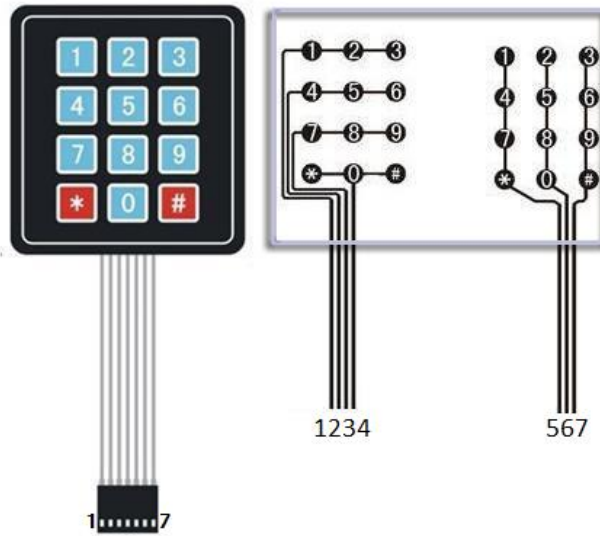
Resim 11.12: 4x3 ve 4x4 Tuş Takımları

4x3 tuş takımları üzerinde “0-9” rakamlarının yanında “*” ve “#” karakterleri de yer almaktadır. Aşağıdaki görselde 4x3 tuş takımının içyapısı gösterilmektedir. Satırlar için 4 pinden gelen sinyal, sütunlar için ayrılan 3 pine aktarılma kombinasyonuna göre çalışmaktadır. Örneğin aşağıdaki resimde Satır 1 pininden gönderilen sinyal 1 numaralı tuşa basıldığında Sütun 1 numaralı pinden mikrodenetleyiciye ulaşacaktır ve 1 numaralı tuşa basıldığı mikrodenetleyici tarafından algılanmış olacaktır.



Resim 11.13: 4x3 Tuş Takımının İç Yapısı

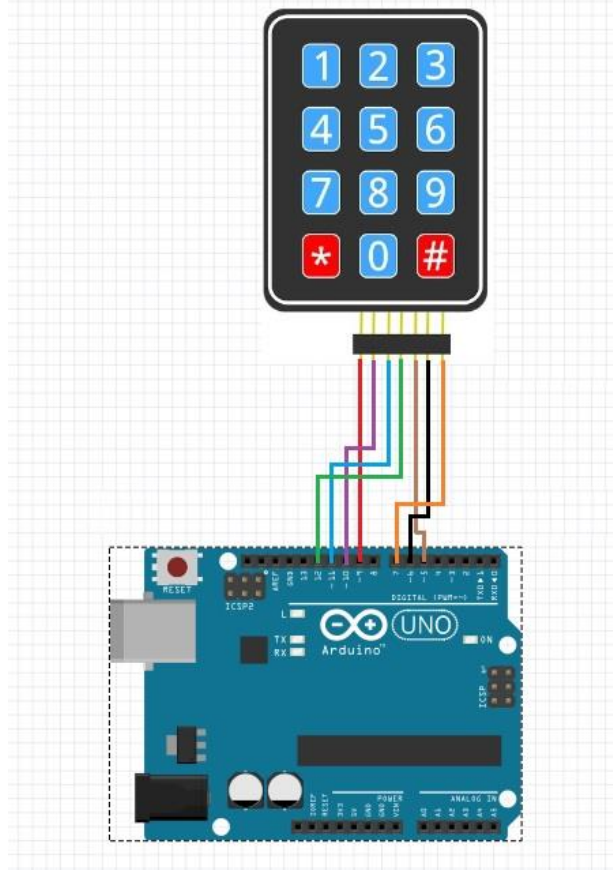
Keypad devre elemanı Arduino Uno mikrodenetleyicisine her satır ve her sütun için ayrılmış toplam 7 pin ile bağlanmaktadır. Bu pinlerin sırasıyla soldan ilk 4 tanesi satırlar için bağlanırken, yine sırası ile geri kalan 3 tanesi sütunlar için bağlanmaktadır. Aşağıdaki resimde 4*3 ve 7 pinli tuş takımının iç yapısı görülmektedir.



Resim 11.14: 4x3 ve 7 Pinli Tuş Takımının İç Yapısı

4x3 Keypad Pinleri		
Pin No	Arduino UNO Pin	Görevi
1	9	Satır 1
2	10	Satır 2
3	11	Satır 3
4	12	Satır 4
5	5	Sütun 1
6	6	Sütun 2
7	7	Sütun 3

Bu uygulamada tuş takımının çalışma mantığının kavranabilmesi için tuş takımında basılan tuşun değerinin “Serial Monitörde” görülebileceği basit bir kod yazılacaktır. Öncelikle eğitmen yukarıdaki pin eşleştirmelerine göre tuş takımını Arduino UNO’ ya bağlamalıdır. Bunun için örnek devre şeması aşağıdaki resimde verilmiştir.



Resim 11.15: Gözle-Uygula Etkinliği Örnek Devre Şeması

Yukarıdaki devreyi eğitmen önce kendisi kurar ve sonrasında devrenin çalışma mantığını öğrencilere aktararak onların da aynı devreyi kurmalarını sağlar. Eğitmen öğrencilerin kendi devrelerini kurmaları için yeterli süreyi verir. Bu süreçte öğrencileri gözlemler, gerektiğinde öğrencilerin devreyi kurmalarına yardımcı olur.

Devre kurulumu tamamlandıktan sonra aşağıdaki resimde örneği verilen kodlar ile devrede tuş takımından herhangi bir tuşa basıldığında Serial Monitörde numaranın görüneceği program tamamlanmış olur. Ancak, Arduino UNO ile keypad kullanımı için ayrıca bir kütüphane kullanmamız gerekmektedir. Keypad kütüphanesi arduino kütüphane klasörüne eklenmelidir. Burada tuş takımı ile ilgili kütüphanenin öğrenciler tarafından indirilip, kod satırlarının ilk kısmına eklenmesini sağlar. Eğitmen öğrencilerin de kendi kod satırlarını yazmalarını sağlar. Eğitmen tarafından kodların öğrencilere direkt yansıtılması yerine, öğrencilerin kendi bilgi ve becerileri ile uygun kod satırlarını yazmaları sağlanmalıdır. Eğitmen gerekli noktalarda öğrencilere direkt çözümü vermeden yardımcı olabilir. Son olarak yazılan kod satırları Arduino UNO kartlarına yüklenerek tuş takımı uygulamasının sorunsuz bir şekilde çalıştığı kontrol edilir.


```

keypad_01
1 #include <Keypad.h>
2
3 const byte satir = 4;
4 const byte sutun = 3;
5
6 char tus;
7 char tuslar [satir][sutun] = { ('1', '2', '3'),
8                               ('4', '5', '6'),
9                               ('7', '8', '9'),
10                              ('*', '0', '#')};
11
12 byte satir_pinleri[satir]={9,10,11,12};
13 byte sutun_pinleri[sutun]={5,6,7};
14
15 Keypad tus_nesnesi = Keypad(makeKeymap(tuslar), satir_pinleri, sutun_pinleri, satir, sutun);
16
17 void setup() {
18
19     Serial.begin(9600);
20
21 }
22
23 void loop() {
24
25     tus = tus_nesnesi.getKey();
26     if(tus){
27         Serial.println(tus);
28     }
29
30 }
31

```

Resim 11.16: Gözle-Uygula Etkinliği Örnek Kodu

1.5 Gözle ve Uygula – Su Seviyesi/Yağmur Sensörü Kullanımı

Malzeme Listesi	
Arduino UNO	Su seviyesi/Yağmur sensörü
Bağlantı kabloları	Breadboard
330 ohm direnç	Buzzer

Yağmur Sensörü

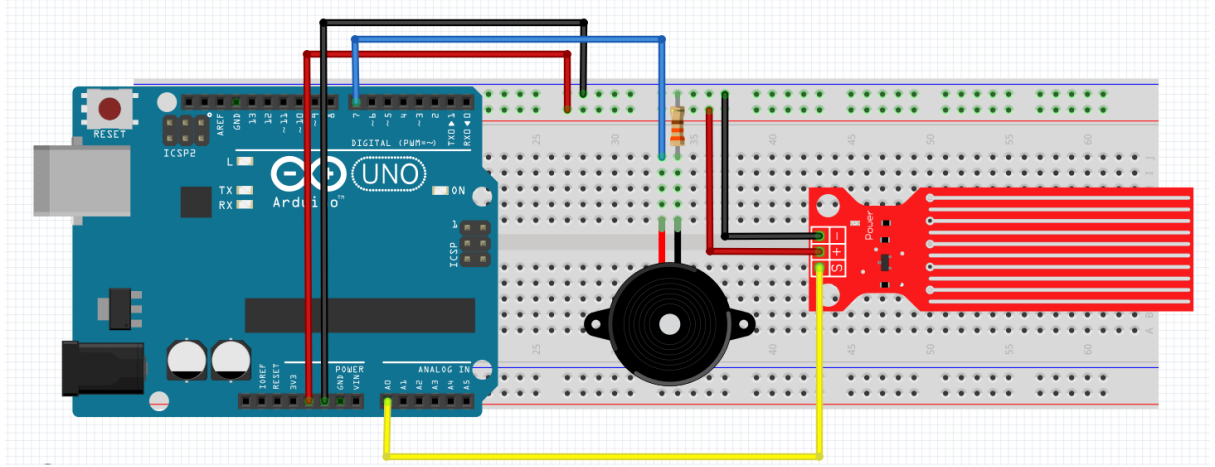
Bu sensör sığ su seviyelerinde ve yağmurlu ortamlarda kullanılan bir sensördür. Farklı türleri bulunmasına karşın bizim deney setlerimizde yer alan sensör 40 mm su seviyesine kadar ölçüm yapabilmektedir. Bunu gerçekleştirmek için üzerinde yer alan paralel olarak yerleştirilmiş iletken hatlardan faydalanır, bu iletken hatlara su temas ettikçe iletkenliği değişmektedir. Bu iletken hatlardan elde edilen veriyi kullanarak sensör çıkışlarına 0 ile 1023 arasında analog çıkış vermektedir. Bu sensörün 3 pini bulunmaktadır. Bunlardan biri GND ve 5 V girişleriyken, S pini analog data çıkış pinidir. Aşağıdaki resimde yağmur sensörü görülmektedir.



Resim 11.17: Yağmur Sensörü

Bu gözle uygula etkinliğinde eğitimci öğrencilerle beraber su seviyesi/yağmur sensörünü kullanarak bir yağmur alarm devresi kuracak ve buna uygun kodu yazacaktır. Burada devreyi kurmak için gerekli olan devre elemanları ve sensör yukarıda belirtilmiştir.

Bu devrede amacımız yağmur seviyesi belirli bir eşik değerini geçtiğinde buzzerın alarm vermesidir. Eğitimci öğrencilerle beraber board üzerine aşağıda verilen devreyi kurar, sensör ve çalışma prensibi hakkında bilgi verir. Devrenin çalışma mantığını öğrencilere anlatır, gerekirse öğrencilere devreyi kurmaları konusunda yardımcı olur. Etkinlikle ilgili örnek şema aşağıdaki resimde görülmektedir.



Resim 11.18: Gözle-Uygula Etkinliği Örnek Devre Şeması

Eğitimci devre kurulumu tamamlandıktan sonra devrenin çalışması için aşağıdaki resimde verilen kodu yazar ve kod ile ilgili açıklamalar yapar. Ardından bu kodu öğrencilerin yazmasını ister ve Arduino UNO'ya yüklemelerini ister. Gerekğinde öğrencilere yardımcı olur.

```

yagmur
1 int sensorPin = A0;
2 int yagmurseviyesi = 100;
3 int buzzerPin = 7;
4 int deger;
5
6 void setup() {
7   pinMode(buzzerPin, OUTPUT);
8 }
9 void loop() {
10  deger = analogRead(sensorPin);
11  if(deger > yagmurseviyesi){
12    digitalWrite(buzzerPin, HIGH);
13    delay(100);
14    digitalWrite(buzzerPin, LOW);
15    delay(100);
16  }
17  else{
18    digitalWrite(buzzerPin, LOW);
19  }
20 }

```

Resim 11.19: Gözle-Uygula Etkinliği Örnek Kodu

1.6 Gözle ve Uygula – Gaz Sensörü Kullanımı

Malzeme Listesi	
Arduino UNO	Gaz Sensörü Kartı
Bağlantı kabloları	Breadboard
330 ohm direnç	Buzzer

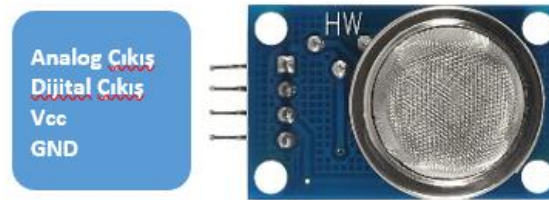
Gaz Sensörü (MQ-2)

Bu sensör breadboard üzerine yerleştirilmeye uygun olmadığı için bir kart üzerine yerleştirilmiş halde kullanılır. MQ-2'nin görünümü aşağıdaki resimde görüldüğü gibidir. Bu sensör uçan ve yanıcı olan gaz çeşitlerini (lpg, alkol, propan, hidrojen, metan, karbonmonoksit ve duman da dahil) konsantrasyonlarını algılamak için kullanılır. MQ serisinde pekçok farklı sensör tipi vardır ve bunlar farklı gazların, farklı konsantrasyon aralıklarında seviyesini ölçmek için kullanılır.



Resim 11.20: MQ-2 Gaz Sensörü

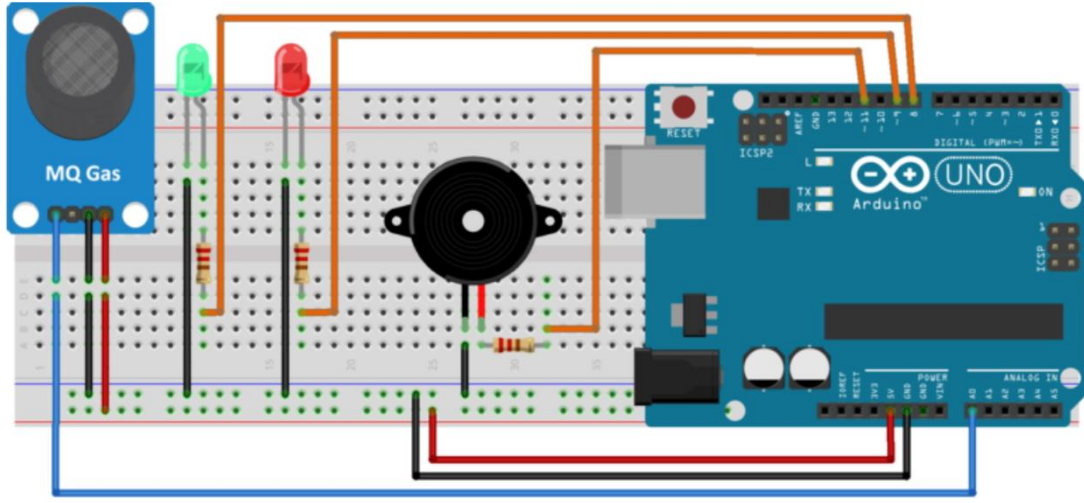
MQ-2'nin tek başına breadboarda bağlantısı yapılamadığı için bir kart üzerinden bağlantılarının gerçekleştirileceğini belirtmiştik. Kart üzerinde Vcc ve Gnd pinleri dışında, dijital ve analog çıkış alabileceğimiz 2 pin daha vardır. Analog çıkışta ortamdaki gaz oranı arttıkça, analog çıkış voltajı artacaktır (0 V - 5 V arasında bir değer alır). Dijital çıkışın eşik değeri ise bu kart üzerine yerleştirilmiş ayarlı potansiyometre ile belirlenebilmektedir. Bu çıkışın kalibre edilmesi için ise sensör board üzerinde bağlıken belirli bir miktar gaza maruz bırakılarak, kırmızı LED yanana kadar potansiyometre ile ayarlama yapılmaktadır. Hassasiyeti deneyerek belirlenmektedir. Aşağıdaki resimde MQ-2 gaz sensörünün pin bağlantıları görülmektedir.



Resim 11.21: MQ-2 Gaz Sensörü Pin Bağlantıları

Bu gözle uygula etkinliğinde eğitmen öğrencilerle beraber MQ-2 gaz sensörünü kullanarak bir gaz alarm devresi kuracak ve buna uygun kodu yazacaktır. Burada devreyi kurmak için gerekli olan devre elemanları ve sensör yukarıda belirtilmiştir.

Bu devrede amacımız gaz seviyesi normalden yeşil LED'in yanması, belirli bir eşik değerini geçtiğinde buzzerın alarm vermesi ve kırmızı LED'in yanmasıdır. Ayrıca bütün bu süreç içerisinde 100 ms aralıkla serial monitör ekranında Analog sensörden gelen değerin görülmesi gerekmektedir. Eğitmen öğrencilerle beraber board üzerine aşağıdaki resimde verilen devreyi kurar, sensör ve çalışma prensibi hakkında bilgi verir. Devrenin çalışma mantığını öğrencilere anlatır, gerekirse öğrencilere devreyi kurmaları konusunda yardımcı olur.



Resim 11.22: Gözle-Uygula Etkinliği Örnek Devre Şeması

Eğitmen devre kurulumu tamamlandıktan sonra devrenin çalışması için aşağıdaki resimde verilen kodu yazar ve kod ile ilgili açıklamalar yapar. Ardından bu kodu öğrencilerin yazmasını ister ve Arduino UNO'ya yüklemelerini ister. Gerekğinde öğrencilere yardımcı olur.

```

gazsensoru
1 int KLed = 8;
2 int YLed = 9;
3 int buzzer = 11;
4 int duman = A0;
5 int sensorseviye = 400;
6 void setup() {
7   pinMode(KLed, OUTPUT);
8   pinMode(YLed, OUTPUT);
9   pinMode(buzzer, OUTPUT);
10  pinMode(duman, INPUT);
11  Serial.begin(9600);
12 }
13 void loop() {
14   int analogSensor = analogRead(duman);
15   Serial.print("Sensörden gelen değer: ");
16   Serial.println(analogSensor);
17
18   if (analogSensor > sensorseviye)
19   {
20     digitalWrite(KLed, HIGH);
21     digitalWrite(YLed, LOW);
22     tone(buzzer, 1000, 200);
23   }
24   else
25   {
26     digitalWrite(KLed, LOW);
27     digitalWrite(YLed, HIGH);
28     noTone(buzzer);
29   }
30   delay(100);
31 }

```

Resim 11.23: Gözle-Uygula Etkinliği Örnek Kodu

2. ADIM: TASARLA ve ÜRET

2.1 Tasarla- Nükleer Santraller için Kaza Simülasyonu ve Güvenlik Protokolü

Nükleer santrallerde, nükleer maddelerin çevreye bırakılmamasını ve aynı zamanda nükleer reaksiyon sonucunda oluşan ısının her durumda reaktörden alınmasını garantiye alacak şekilde birçok güvenlik önlemi alınmıştır. Nükleer güç santrallerinde fisyon işlemi sonucunda büyük miktarlarda radyoaktivite ortaya çıktığı için, meydana gelebilecek herhangi bir kaza durumunda diğer nükleer tesislerde oluşabilecek kazalardan daha büyük hasar oluşturacak potansiyele sahiptir. Örneğin 11 Mart 2011 tarihinde meydana gelen 9.0 büyüklüğündeki deprem ve onu takip eden tsunami sonucu, nükleer santral tarihinin en büyük ikinci kazası Japonya'nın Fukushima şehrinde meydana gelmiştir. Kaza ile ilgili <https://www.afad.gov.tr/kbrn/fukushima-daiichi-nukleer-santral-kazasi> sitesinde detaylı bilgiler elde edilebilir. Kaza süreçleri incelenecek olursa farklı sebeplerden dolayı güvenlik önlemlerinin etkisiz hale geldiği görülebilmektedir.

Nükleer santrallerde kazaların önüne geçebilmek için tasarlanmış güvenlik sistemleri ile radyoaktivitenin serbest kalmasını engelleyen birçok bariyerler ve protokoller mevcuttur. Bu

etkinlikte öğrencilerden bir nükleer santral için tanımlanmış güvenlik protokolü çerçevesinde kademeli olarak ilerleyecek bir kaza simülasyonu ve simülasyona yönelik olarak da olası güvenlik önlemlerini tasarımları istenmektedir. Beş madde halinde tasarlanacak olan güvenlik protokolü ve önlemlerin sıra ile şu adımları içermesi beklenmektedir:

1. Kaza simülasyonun aynı Fukushima’da olduğu gibi bir deprem ile başladığı,
 - Depremin tilt sensörünün hareket ettirilmesiyle başladığı varsayılacaktır ve depremin algılanması ile birlikte kırmızı bir LED’in yanıp sönmeye başlamesi gerekmektedir. Depremsel aktivite yokken bu LED yanmayacaktır.
2. Kaza simülasyonunun 2. evresinde reaktörlerin gaz salınımına başladığı,
 - Gaz sensörüne gaz verilmesi ile sensörün devreye girip buzzer ile uyarı sinyali vermesi gerekmektedir.
3. Kaza simülasyonunun 3. evresinde Fukushima’daki kazada olduğu gibi dışarıdan ya da içeriden tehlikeli bölümlere su veya yağmur suyunun dolmaya başladığı,
 - Yağmur sensörünün devreye girip Buzzer ile bir önceki maddeden farklı bir uyarı sinyali vermesi gerekmektedir.
4. Simülasyonun 4. evresinde güvenlik protokolüne dayanarak artık nükleer santral içerisindeki personelin tahliyesinin başladığı,
 - RFID kullanarak özel korumalı acil çıkış kapılarının bir servo motor ile açılmasının sağlanması gerekmektedir.
5. Kaza simülasyonunun son evresinde ise nükleer santral acil çıkış koridorları içerisinde hiçbir personelin kalmaması için,
 - PIR sensörünün hareketli bir nesne yoksa senaryo gereği tüm sistemin durdurulması/kapatılması gerekmektedir. Bunun içinde Arduino üzerinde bir LED ile tesisin güvenli şekilde kapatıldığı gösterilir.

2.2 Tasarla – Nükleer Santraller için Güvenlik Protokolü Tasarımı

Bu aşamada öğrenciler ders kapsamında ve daha önceki haftalarda öğrendikleri devre elemanlarını kullanarak güvenlik protokolü tasarımını gerçekleştireceklerdir. Program kodlarının yazımında ve algoritmanın oluşturulmasında eğitmen gerekli noktalarda öğrencilere yönlendirici sorular sormalı ve önerilerde bulunmalıdır. Fakat devrenin kurulumu ve programın kodlarını öğrencilere hazır olarak vermemelidir.

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşğıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Güvenlik protokolünü içerek şekilde devrenin kurulumunda kullanılacak pin’lerin breadboard üzerindeki yerleşimleri ve konumlandırılmaları nasıl olmalıdır?
- Kaza simülasyonunun adım adım ilerleyecek olmasının
- Arduino IDE program algoritması nasıl kurulmalıdır? Bu algorithmada beş adımda gerçekleştirilecek olan adımlar kodlar içerisinde nasıl yer almalıdır?
- Arduino IDE içerisinde hangi karar yapıları kullanılmalıdır? Neden?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen devre ve Arduino IDE programına yönelik işlemlerin nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Güvenlik protokolü sıra ile gerçekleştirilecek beş adımdan oluştuğu için fiziksel tasarımda kullanılacak sensörlerin breadboard üzerinde uygun şekilde yerleştirilmesi gerekmektedir. Özellikle yağmur ve gaz sensörünün aktif durumda kullanılabilmesi için fiziksel olarak erişilebilir şekilde konumlandırılmalıdır.
- Hem devre bağlantıları için hem de Arduino UNO üzerinde GND ve 5 V girişlerini çoğaltmak ve bazı devre elemanlarının fiziksel yerleşimlerinin düşünülmesi gerekmektedir. Örneğin gaz ve yağmur sensörünün çalışabilecek şekilde; RFID ve keypad'in kullanılacak şekilde breadboard üzerinde konumlandırılması gerekmektedir.
- Arduino IDE içerisinde yazılacak programın algoritması kontrol yapılarının kaza simülasyonuna uygun şekilde cevap verebilecek şekilde tasarlanması gerekmektedir. Bu tasarımın farklı şekillerde yazılabilecek olması fikir üretimi aşamasında öğrenciler için önemli aşamalardan biridir.

2.3 Üret- Nükleer Santraller için Güvenlik Protokolü Tasarımı

Öğrenciler bir nükleer santral için öngörülen kaza simülasyonuna uygun şekilde geliştirecekleri güvenlik protokolü tasarımlarını yaptıktan sonra bilgisayar başında Arduino IDE üzerinde kodları yazarlar. Öğrencilere eğitmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Özellikle Arduino IDE içerisinde yazılacak program kodlarında farklı sensörlere ait kodların ve çözüm için algoritmanın oluşturulma sürecinde eğitmenin rehberliği öğrencilerin etkinliği gerçekleştirmesinde önemli olacaktır. Etkinlik için örnek kod aşağıdaki resimde verilmiştir.

```

Nukleer_Guvenlik
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <Servo.h>
4 #define egim_sensor 2
5 #define SS_PIN 10
6 #define RST_PIN 9
7 #define Servo_PIN 8
8 #define PIR_PIN 7
9 #define LED_deprem 3
10 #define LED_gaz 4
11 #define LED_acilcikis 5
12 #define LED_sistemkapali 6
13
14 int buzzer = 11;
15 int duman = A0;
16 int sensorseviye = 400;
17 int sensorPin = A1;
18 int yagmurseviyesi = 100;
19 int deger;
20 int hareket;
21
22 MFRC522 rfid(SS_PIN, RST_PIN); // Create MFRC522 instance.
23 byte ID[4] = {196,254,80,131};
24 Servo kapi;
25
26 void setup() {
27
28   pinMode(egim_sensor, INPUT);
29   pinMode(LED_deprem, OUTPUT);
30   pinMode(LED_gaz, OUTPUT);
31   pinMode(LED_acilcikis, OUTPUT);
32   pinMode(LED_sistemkapali, OUTPUT);
33   pinMode(buzzer, OUTPUT);
34   noTone(buzzer);
35   pinMode(duman, INPUT);
36   pinMode(PIR_PIN, INPUT);
37   Serial.begin(9600);
38   SPI.begin();
39   rfid.PCD_Init();
40   kapi.attach(Servo_PIN);
41   kapi.write(0); //servo start position
42 }
43
44 void loop() {
45
46   int deger = digitalRead(egim_sensor);
47   if(deger==LOW) {
48     digitalWrite(LED_deprem,HIGH);
49     delay(1000);
50     digitalWrite(LED_deprem,LOW);
51     delay(1000);
52
53     int analogSensor = analogRead(duman);
54     if (analogSensor > sensorseviye)
55     {
56       digitalWrite(LED_gaz, HIGH);
57       tone(buzzer, 1000, 200);
58       deger = analogRead(sensorPin);
59       if(deger > yagmurseviyesi)
60
61       {
62         digitalWrite(buzzer, HIGH);
63         delay(100);
64         digitalWrite(buzzer, LOW);
65         delay(100);
66
67         if (rfid.uid.uidByte[0]==ID[0] &&
68             rfid.uid.uidByte[1]==ID[1] &&
69             rfid.uid.uidByte[2]==ID[2] &&
70             rfid.uid.uidByte[3]==ID[3] )
71
72         {
73           kapi.write(180);
74           digitalWrite(LED_acilcikis, HIGH);
75           tone(buzzer, 500);
76           delay(5000);
77           noTone(buzzer);
78           kapi.write(0);
79           digitalWrite(LED_acilcikis, LOW);
80           delay(1000);
81
82           hareket = digitalRead(PIR_PIN);
83           if (hareket == HIGH){
84             digitalWrite(LED_sistemkapali, LOW);
85             delay(1000);
86           }
87           else {
88             digitalWrite(LED_sistemkapali, HIGH);
89             delay(10000);
90           }
91
92           else {
93             tone(buzzer, 300);
94             delay(1000);
95             noTone(buzzer);
96             digitalWrite(LED_acilcikis, LOW);
97             delay(1000);
98           }
99
100          }
101          else{
102            digitalWrite(buzzer, LOW);
103            delay(1000);
104          }
105        }
106        else
107        {
108          digitalWrite(LED_gaz, LOW);
109          noTone(buzzer);
110          delay(100);
111        }
112      }
113    }
114  }
115  digitalWrite(LED_deprem,LOW);
116  delay(100);
117 }
118 }

```

Resim 11.24: Tasarla-Üret Etkinliği Örnek Kodu

3. ADIM: DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Nükleer santraller için güvenlik protokolü tasarımında kullanılan sensör ve devre elamanlarının isimleri kağıtlara yazılır ve tombala oyunu misali öğrencilerden bir kağıt seçmeleri istenir:

- Öğrencilerden seçilen sensörlerin ve devre elemanlarının özelliklerini anlatmaları ve bu sensörlerin başka hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Nükleer santraller için güvenlik protokolü etkinliğinde seçilen sensörlerin ve devre elemanlarının Arduino IDE içerisindeki kod yazımlarında karşılaşılan güçlükler nelerdi?

- Öğrencilerden şu aşamaya kadar öğrenilen tüm sensörlerin bir arada kullanılabileceği proje önerileri tartışmaları istenir.
- Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilir.

4. PROJEYE YÖNELİK PROTOTİP GELİŞTİRME

Bu Hafta Gerçekleştirilecek Etkinlikler: Dersin sonunda eğitmen öğrencilerin proje kayıt defterlerinde bu haftaya ilişkin prototip geliştirme bölümüne ait soruları cevaplayıp cevaplamadıklarını kontrol eder. Öğrenci gruplarının bu bölümde yazmış oldukları tasarım ve programlamaya yönelik geliştirdikleri algoritmaların çözüme yönelik test etme öncesi önemli adım olduğu hatırlatılmalı ve grupların fikirleri tartışılmalıdır. Özellikle çözüme yönelik olası seçimlerin nasıl yapıldığı grup üyeleri tarafından eğitmene aktarılmalıdır.

Ayrıca öğrencilere proje kayıt defterlerinde son hafta proje sergisi günü teslim edilecek proje ayrıntıları bölümüne “**video**” kaydı için tüm çalışmalarının dijital kayıtlarını almalarının gerekliliği hatırlatılmalıdır.

Sonraki Haftalara Hazırlık (12. - 13. Haftalar): Gruplara gelecek iki hafta süresince geliştirdikleri prototipleri test etme adımının gerçekleştirileceği hatırlatılmalıdır. Özellikle geliştirdikleri prototipin test işlemi sonrası tanımlanan problemi çözebilme becerisinin sonuçlarının doğru şekilde irdelenmesinin gerekeceği önemlidir. Proje sergisi öncesi proje kayıt defterinin son bölümlerinin de doldurulmasının gerektiği öğrencilere aktarılmalıdır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI (Tanımlama – 9. HAFTA)**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

○ **ÇÖZÜM ÖNERİSİ TASLAĞI (Fikir Üretme – 10. HAFTA)**

- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?

- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?

- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?

○ **PROTOTİP GELİŞTİRME (Prototip Geliştirme – 11. HAFTA)**

- Geliştirdiğiniz prototip tanımlanmış olan problemin hangi özelliklerini çözmektedir?

- Geliştirdiğiniz prototip üzerinde değiştirmek/çözümünüzde geliştirmek istediğiniz noktalar nelerdir?

○ **TEST ETME (Test Etme – 12. HAFTA)**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüze yönelik olarak hedef kitlenin değerlendirmesi nasıl oldu?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzü hedef kitlenin kullanması ile hangi sorunlar ile karşılaştı?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzün hedef kitle kullanımı sonrası tanımlamış olduğunuz problemi çözebilme başarısı nedir?

12.Bölüm – Projeyi Test Etme

Bu Hafta Gerçekleştirilecek Etkinlikler: Dersin sonunda eğitimci öğrencilerin proje kayıt defterlerinde bu haftaya ilişkin test etme bölümüne ait soruları cevaplayıp cevaplamadıklarını kontrol eder. Bu aşamada öğrencilerin gerçekleştirdikleri işlemlerin çözüme yönelik önemli adımlar olduğu hatırlanmalıdır. Bu kapsamda test işleminin uygulanacağı hedef kitlenin geliştirilmiş olan teknolojik çözüme yönelik değerlendirmeleri ve yorumları proje kayıt defterine işlenip işlenmediği kontrol edilmelidir.

Sonraki Haftalara Hazırlık (13. Hafta): Gruplara gelecek hafta derse gelirken test etme bölümü için tüm test işlemlerinin tamamlanması ve proje kayıt defterinde ilgili alanların doldurulması gerektiği öğrencilere hatırlatılmalıdır. Ayrıca gelecek hafta tüm proje süresince kayıt altına almış oldukları videoyu düzenleyerek proje kayıt defterine diğer bölümler ile birlikte eklemeleri gerektiği vurgulanmalıdır. Özellikle haftaya proje kayıt defterinin bitmiş ve son halinin teslim edileceği hatırlatılmalıdır.

PROJE KAYIT DEFTERİ

- **GRUP ADI (Giriş Haftası – 7. HAFTA)**

- **PROJE EKİBİ (Giriş Haftası – 7.HAFTA)**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

- **PROJE İSMİ (Empati – 8. HAFTA)**

- **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI (Empati – 8. HAFTA)**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI (Tanımlama – 9. HAFTA)**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

○ **ÇÖZÜM ÖNERİSİ TASLAĞI (Fikir Üretme – 10. HAFTA)**

- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?

- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?

- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?

○ **PROTOTİP GELİŞTİRME (Prototip Geliştirme – 11. HAFTA)**

- Geliştirdiğiniz prototip tanımlanmış olan problemin hangi özelliklerini çözmektedir?

- Geliştirdiğiniz prototip üzerinde değiştirmek/çözümünüzde geliştirmek istediğiniz noktalar nelerdir?

○ **TEST ETME (Test Etme – 12. HAFTA)**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüze yönelik olarak hedef kitlenin değerlendirmesi nasıl oldu?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzü hedef kitlenin kullanması ile hangi sorunlar ile karşılaştı?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzün hedef kitle kullanımını sonrası tanımlamış olduğunuz problemi çözebilme başarısı nedir?

○ **PROJEDE KULLANDIĞINIZ DEVRE ELEMANLARI VE DİĞER KAYNAKLAR (13. HAFTA)**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzde size set ile verilen devre elemanlarından, sensörlerden, modüllerden vb. hangilerini kullandınız?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzde yararlandığınız basılı ve görsel kaynaklar nelerdir?

○ **PROJENİN AYRINTILARI (13. HAFTA)**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzün yapım aşamalarını aşağıdaki bölümde fotoğraflar ile anlatmanız beklenmektedir. Grup çalışmalarınız, devre çizimleriniz, kod yazım aşamalarınız, aktarmak istediğiniz notlar vb. yani proje geliştirme sürecinizi ayrıntılı şekilde yazınız.

- Proje sürecinizi özetleyen ve çözüme ulaşma aşamalarınızı gösteren video linkinizi yazınız.

13.Bölüm – Proje Defterinin Teslim Edilmesi

PROJE KAYIT DEFTERİ

○ **GRUP ADI**

○ **PROJE EKİBİ**

ADI – SOYADI	GÖREVİ
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

○ **PROJE İSMİ**

○ **ÇÖZÜLMESİ İSTENEN PROBLEM TANIMI**

- Problem Cümlesi: Geliştirilecek çözüm ile ne yapılmak isteniyor?

- Geliştirilecek çözümün kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?

○ **ÇÖZÜMÜN TANIMLANMASI**

- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?

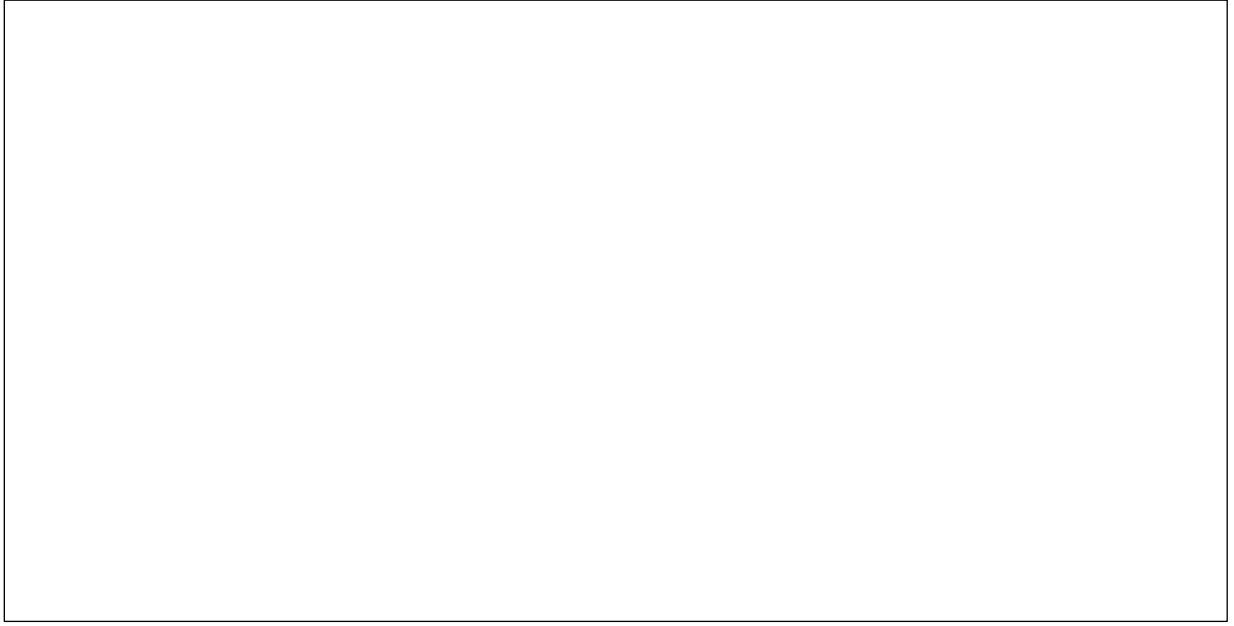
- Teknolojik çözümün belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

○ **ÇÖZÜM ÖNERİSİ TASLAĞI**

- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?

- Alternatif tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?

- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?



○ **PROTOTİP GELİŞTİRME**

- Geliştirdiğiniz prototip tanımlanmış olan problemin hangi özelliklerini çözmektedir?

- Geliştirdiğiniz prototip üzerinde değiştirmek/çözümünüzde geliştirmek istediğiniz noktalar nelerdir?

○ **TEST ETME**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüze yönelik olarak hedef kitlenin değerlendirmesi nasıl oldu?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzü hedef kitlenin kullanması ile hangi sorunlar ile karşılaşıldı?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzün hedef kitle kullanımı sonrası tanımlamış olduğunuz problemi çözebilme başarısı nedir?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzde size set ile verilen devre elemanlarından, sensörlerden, modüllerden vb. hangilerini kullandınız?

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzde yararlandığınız basılı ve görsel kaynaklar nelerdir?

○ **PROJENİN AYRINTILARI**

- Geliştirdiğiniz Arduino tabanlı teknolojik çözümünüzün yapım aşamalarını aşağıdaki bölümde fotoğraflar ile anlatmanız beklenmektedir. Grup çalışmalarınız, devre çizimleriniz, kod yazım aşamalarınız, aktarmak istediğiniz notlar vb. yani proje geliştirme sürecinizi ayrıntılı şekilde yazınız.

- Proje sürecinizi özetleyen ve çözüme ulaşma aşamalarınızı gösteren video linkinizi yazınız.