



BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MKT0311_Mikroişlemci Tabanlı Sistem Tasarımı
(BAHAR 2022)

Dönem Projesi

Değerlendirme Kriterleri:

1. Projede konu belirlenmesi ve projenin tamamlanması işlemleri belirlenen zamanda gerçekleştirildi mi? **(5p)**
2. Raporda kullanılan sensör ve elemanların ayrıntılı tanıtımı yapıldı mı? **(5p)**
3. Raporda donanım bağlantı devresi Fritzing gibi programda çizildi ve çalışma mantığı anlatıldı mı? **(10p)**
4. Raporda yazılım akış şeması yapılmış mı? Anlatıldı mı? Ve kodda açıklamalar comment olarak yan tarafa yapılmış mı? **(10p)**
5. Tüm sistem fiziksel olarak çalıştırıldı ve sorunsuz çalışıyor mu? **(30p)**
6. Örnek seçilen projeden ne farklılıklar var ayrıntılı gösterilip açıklandı mı? **(10p)**
7. Yapılan proje proteus simülatöründe aynı sensörlerle sorunsuz çalıştırıldım mı? **(20p)**
8. Sunum ve gösteri esnasında sorulan sözlü sorulara yeterince cevap verildi mi? **(10p)**
9. Raporda grup üyeleri iş bölümünü ayrıntılı olarak açıklamış mı? **(Bu ayırım yapılmamış rapor değerlendirilmeyecektir)**

Deneyi Yapanın

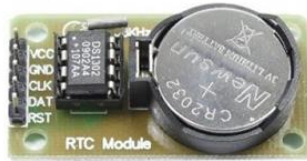
Grup üyeleri Ad ve Soyadı:	Ceyda Tekin, Fatih Es, Mehmet Salih Önder, Muhamed Saltuk Yaşar
Öğrenci Numaraları:	18360859025, 18360859021, 18360859028, 18360859016
Rapor Teslim Tarihi:	
Rapor Sunum Tarihi:	

1- PROJE İSMİ: Akıllı Saksı

2- **PROJE AMACI:** Fotosentez verimini arttıran, havanın koşullarını iyileştiren, ekran yardımıyla iletişim kuran, kendi kendisini ilaçlayabilen ek olarak birçok konfigrasyonu bulunan akıllı saksı kullanıcıların bitkilerinin daha uzun süre yaşamalarını ve gereksinimlerini insanlara aktarmaktadır.

3- KULLANILAN MALZEMELERİN TANITIMI:

a) Saat Modülü - DS1302/1307- RTC Modül:

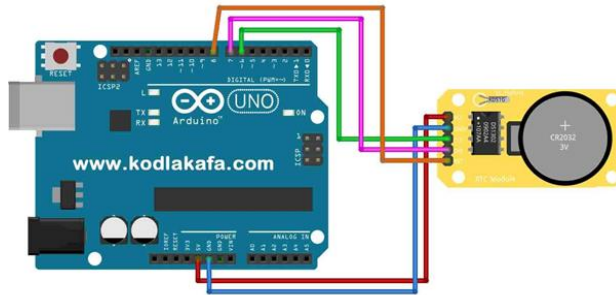


RTC'nin açılımı Real Time Clock yani Gerçek Zamanlı Saat'tir. RTC modülünün içerisinde gerçek zamanlı saat devrelerinde bulunan DS1302 entegresi bulunur. Bu modül sayesinde saat bilgisini anlık ve sürekli olarak okuyabilmek mümkündür. RTC modülü yıl, ay, gün, saat, dakika ve saniye bilgilerini sürekli olarak içerisinde barındırmaktadır.

RTC modülünün üzerinde 3V'luk bir CR2032 pili takabilmek için yuva bulunmaktadır. Bu pil DS1302 entegresinin gücünü sağlayarak zamanın sürekli olarak güncel kalmasını sağlamaktadır.

Saat modülünün 5 ayağı bulunmaktadır. Bu ayakların biri 5V enerji alışından sorumlu, bir diğeri GND yani devrenin topraklanmasından sorumludur. Geriye kalan CLK, DATA ve RST pinleri ise arduino uno üzerindeki dijital pinlere bağlanır.

RTC çalışmak için modüle uygun RealTimeClockDS1307.h ya da virtuabotixRTC.h gibi kütüphaneye ihtiyaç duymaktadır.



```
#include <virtuabotixRTC.h>
```

```
//RTC bağlantı pinleri  
//CLK -> 6, Data -> 7, RST -> 8  
virtuabotixRTC RTCSaat(6, 7, 8);
```

```
void setup() {  
  Serial.begin(9600);
```

```
// RTC üzerindeki tarih ve saati aşağıdaki kod ile ayarlayabilirsiniz.  
// saniye, dakika, saat, haftanın günü, ayın günü, ay, yıl (haftanın günü pazartesi için 1 yazıyoruz.)  
RTCSaat.setDS1302Time(00, 10, 12,1, 13, 01, 2020);
```

Kodları açıklayacak olursak;

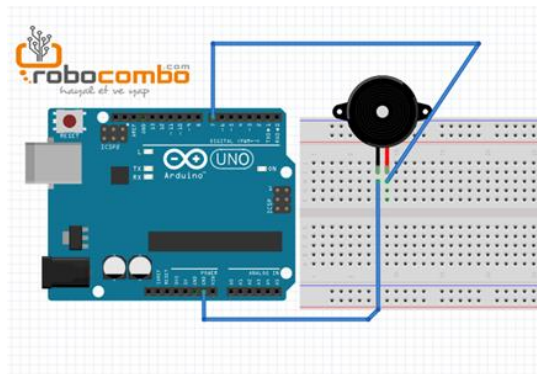
virtuabotixRTC RTCSaat(6, 7, 8); komutu ile RTCSaat isminde bir RTC nesnesi oluşturuyoruz. Parantez içindeki rakamlar ise sırasıyla CLK, DAT ve RST pinlerinin arduino bağlantılarını ifade etmektedir. RTCSaat.setDS1302Time(00, 10, 12, 1, 13, 01, 2020); komutu ile bir sefer RTC modülünün saat ve tarih bilgisini ayarlıyoruz. Bu komutta parantez içindeki sayılar sırasıyla saniye, dakika, saat, haftanın günü, ayın günü, ay ve yıl bilgisini ifade etmektedir. Haftanın günü olarak pazartesi için 1 baz alınmaktadır. Bu kod satırını ilk defa yükleme yaptıktan sonra başına // koyarak kodu açıklama satırına dönüştürmeyi ya da silmeyi unutmayın. Aksi taktirde kodu arduinoya her yüklemede saat ve tarih bilgisini yeniden yüklemiş olursunuz.

- Loop döngüsü içindeki RTCSaat.updateTime(); komutu ile RTC deki kayıtlı tarih ve saat bilgilerini çekiyoruz.
- RTCSaat.dayofmonth komutu ile ayın gün bilgisini,
- RTCSaat.month komutu ile ay bilgisini,
- RTCSaat.year komutu ile yıl bilgisini,
- RTCSaat.hours komutu ile saat bilgisini,
- RTCSaat.minutes komutu ile dakika bilgisini,
- RTCSaat.seconds komutu ile saniye bilgisini çekerek seri ekrana yazdırıyoruz.

b) Buzzer:



Buzzer Arduino devrelerinde ses elde etmek amacıyla kullanılan bir ekipmandır. Buzzerı günlük hayatta bize uyarı veren birçok yerde görebiliriz. Örneğin araçlardaki park sistemlerinde, alarmlarda vb. alanlarda görme şansımız vardır. Kullanımı oldukça kolay olduğu için çok kullanışlı bir ekipmandır.



```
sketch_mar24a $
int buzzerPin=7;

void setup() {
  pinMode(buzzerPin,OUTPUT);
}

void loop() {
  digitalWrite(buzzerPin,HIGH);
  delay(2000);
  digitalWrite(buzzerPin,LOW);
  delay(2000);
}
```

Gördüğünüz üzere kodlaması gerçekten çok basittir. İlk önce `int buzzerPin=7;` komutu ile buzzerımızın giriş pinini 7 numaralı pin olarak seçiyoruz. Daha sonra `pinMode(buzzerPin,OUTPUT);` komutu ile de bizim buzzerımızın bir OUTPUT kaynak olduğunu belirtiyoruz. `digitalWrite(buzzerPin,HIGH); delay(2000); digitalWrite(buzzerPin,LOW); delay(2000);` Burada ise öncelikle buzzerımızı HIGH yani çalışır durumuna getiriyoruz. Daha sonra delay komutu sayesinde 2 saniye çalışır halde beklemesini belirtiyoruz. Daha sonra buzzerımızı LOW durumuna getirip 2 saniye durmasını belirtiyoruz.

c) LCD Ekran + I2C Modülü



LCD Ekran Nedir?

LCD (Liquid Crystal Display), sıvı kristal ekran elektrik ile kutuplanan sıvının ışığı tek fazlı geçirmesi ve önüne eklenen bir kutuplanma filtresi ile gözle görülebilmesi ilkesine dayanan bir görüntü teknolojisidir. LCD ekranlar düşük enerji tüketimleri ve küçük boyutları sayesinde eski tüplü ekranların yerini almıştır. LCD ekranların keşfinden sonra ekran teknolojisinde bir çağ açılmıştır.

I2C Modülü Faydaları

LCD Ekranların arduino vb. kartlarla bağlantılarını yapmak için çok fazla kabloya ve pine ihtiyaç duyulmaktadır. Bu modül sayesinde bu ihtiyaç 4 pin ile çözülmektedir. Ayrıca modül üzerinde bulunan ekran parlaklık ayarı ve kontrast ayarı sayesinde ekranınızın ayarlarını kontrol etmek için fazladan bir devre yapmanıza gerek kalmaz.

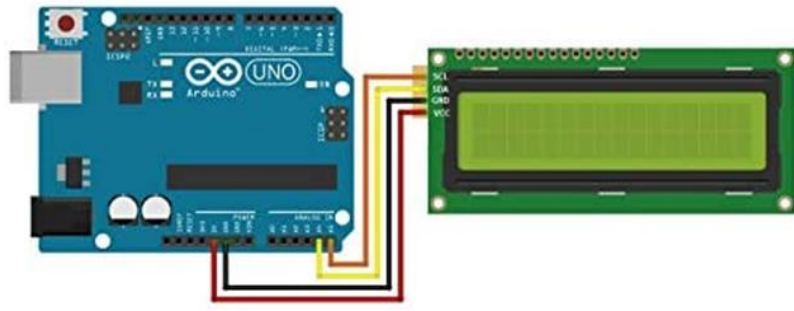


Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

Not: Klon Arduino Modellerinde SDA ve SCL Pinlerine bağlantı yapılır.

I2C Modülü üzerinde 4 adet pin bulunmakta.

- GND = Arduino GND pinine
- VCC = Arduino 5V
- SDA = Klon Arduino SDA Pinine (Orjinal Arduino A4)
- SCL = Klon Arduino SCL Pinine (Orjinal Arduino A5)



Arduino ile I2C Modülü sayesinde ekran arasındaki bağlantıları bu şekilde gerçekleştirdikten sonra artık kodlama kısmına geçilebilir.

Öncelikle arduino için LCD Ekran kütüphanesini indirmemiz gerekmekte

Kütüphaneyi indirdikten sonra IDE programını açıyoruz ve aşağıdaki adımları takip ederek kütüphane kurulumunu yapıyoruz.

Taslak > library ekle(kütüphane ekle) > .ZIP Kitaplığı ekle (İndirdiğiniz zip dosyasını seçiyorsunuz)

Kütüphane kurulumu başarıyla gerçekleşmiş oluyor. Artık uygulamalara geçebilirsiniz.

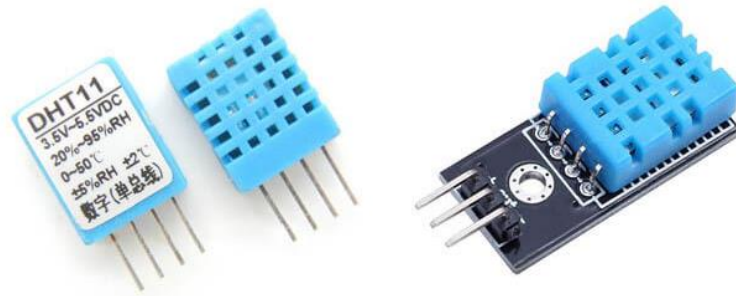
Aşağıdaki uygulamada ilk satır sabit dururken ikinci satırın kayan yazı olması sağlanacaktır.

```
#include <LiquidCrystal I2C.h>
LiquidCrystal I2C lcd(0x27,16,2); // Bu kodu kullanırken ekranda yazı çıkmaz ise 0x27 yerine 0x3f yazınız !!

void setup() {
  lcd.begin();
}

void loop(){
  int i;
  lcd.setCursor(3,0); // ilk satırın başlangıç noktası
  lcd.print("ROBOLINK");
  for(i=0; i<17; i++)
  {
    lcd.setCursor(i,1); // ikinci satırın başlangıç noktası. for döngüsü sayesinde bu sütun sürekli kavraklır.
    lcd.print("TEKNOLOJİ");
    delay(300);
    lcd.setCursor(i,1);
    lcd.print(" ");
  }
}
```

d) Hava Isı/Nem Sensörü



DHT11, ortamdaki sıcaklığı ve nemi ölçmek amacıyla kullanılan bir sensördür. Ortam sıcaklığını çeşitli ısı birimlerinde (celcius, farhenayt, kelvin) elde etmemizi ve ortamdaki nemi (humidity) % cinsinden ölçmek için kullanılmaktadır.

Arduino ile yapılan bir çok projede DHT11 sıcaklık ve nem sensörü tercih edilmektedir. Özellikle akıllı ev vb projelerde çoğunlukla kullanılır.

DHT11 sıcaklık ve nem sensörü tek sensör olarak veya bir devre kartı üzerine entegre edilmiş olarak satılmaktadır. İki sensöründe arduino bağlantısı ve kullanımı aynı olmasına rağmen aralarında sadece fiyat açısından farklılık vardır.

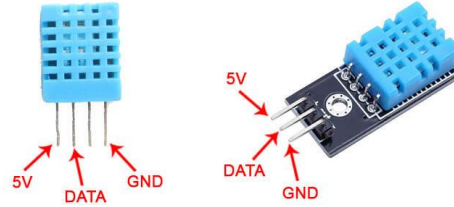
Bu modül sayesinde hazırlayacağınız arduino projesi ile evinizdeki sıcaklık durumuna göre kombinizi kontrol edebileceğiniz veya ortamdaki nem oranı düştüğünde evinizdeki bir nemlendiriciyi otomatik çalıştırabileceğiniz ileri düzey projeler geliştirebilirsiniz.

Birçok modülde olduğu gibi arduino ile DHT11 kullanmak için bir kütüphane dosyası gereklidir. DHT11 kütüphanesi arduino program kurulumunda otomatik olarak eklenen bir kütüphane değildir. Bu nedenle sonradan eklenmesi gerekir.

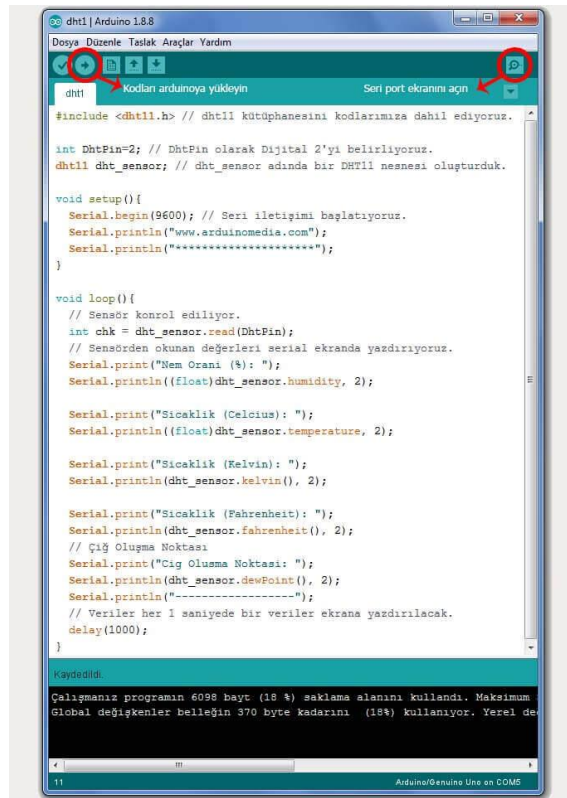
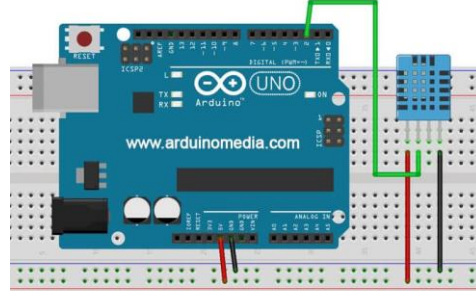
Bu işlem için öncelikle DHT11 arduino kütüphanesini bilgisayarınıza indirin.

Kütüphaneyi indirdikten sonra, dht11 dosyasını, arduino programının kurulu olduğu klasördeki libraries klasörünün içine kopyalayın. Bu işlemden sonra dht11 sıcaklık ve nem sensörünü arduino ile kodlayabilirsiniz.

DHT11 sıcaklık ve nem sensörü bacak bağlantıları aşağıdaki görseldeki gibi yapılmalıdır.



Arduino ile DHT11 sıcaklık ve nem ölçümü için aşağıdaki arduino devresini hazırlayalım.



Programın sağ üst köşesindeki büyüteç butonuna basarak seri port ekranını açarak arduino DHT11 ile ölçülen sıcaklık değerlerini görebilirsiniz.

e) LDR Işık Sensörü



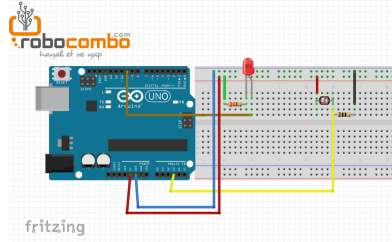
Arduino ile LDR Kullanımı için Gerekli Malzemeler:

- Arduino Uno
- Breadbord
- LDR Işık Sensörü
- 1 Adet LED
- 330 Ω Direnç
- 100 Ω Direnç
- Jumper Kablolar

LDR (Light Dependent Resistor) yani Türkçe 'de "Işığa Bağımlı Direnç" anlamına gelmektedir. LDR ışığa duyarlı bir devre elemanıdır. Ters orantılı bir çalışma prensibine sahiptir. Yani üzerine düşen ışık değeri arttıkça sahip olduğu direnç değeri azalırken, üzerine düşen ışık değeri artıkça sahip olduğu direnç artar. Bu özelliği sebebiyle LDR devrede bir ON-OFF şeklinde anahtarlama görevinde bulunur. Günlük hayatta LDR sensörü alarmlarda, anahtarlama cihazlarında, sokak aydınlatmalarında kullanılabilirler. Devremiz de kullandığımız LDR sensörümüz bir Analog Girişli Sensördür.

Arduino da ADC (Analog To Digital Converter) özelliği mevcuttur. Bu özellik sayesinde dış dünyadan gelen verileri sayısal değerlere dönüştürebiliriz. Arduino UNO kartı üzerinde A0-A5 olmak üzere 6 adet analog giriş bulunur. Analog giriş sayısı modele göre değişebilir ama işlevi aynıdır.

LDR'nin Ne Olduğunu Öğrendiğimize Göre Devre Şemasına ve Kodlamasına Bakalım.



```
int ledPin = 10;
int LDRPin = A3;
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  int isik_degeri = analogRead(LDRPin);
  Serial.println(isik_degeri);
  delay(100);
  if(isik_degeri<200)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

int ledPin = 10; // İlk olarak LED'imizin ve LDR sensörümüzün pinlerini belirliyoruz.
int LDRPin = A3; pinMode(ledPin, OUTPUT); // LED'imizin bir OUTPUT olduğunu belirtiyoruz.

Serial.begin(9600); // Seri iletişimi başlatıyoruz. Loop Kısımına indiğimiz zaman ise;

int isik_degeri = analogRead(LDRPin); // LDR sensörümüzün ölçmüş olduğu ışık değerini int tipinde bir değişkene atıyoruz.

Serial.println(isik_degeri); // Gelen değeri ise yeni değişkenimizde okumasını istiyoruz.
delay(100); // Saniyenin 1/10 kadarını beklemesini söylüyoruz.

if(isik_degeri<200) // if döngüsü başlatıyoruz gelen ölçülen değerın bizim belirlediğimiz değerin altında veya üstünde olma durumunu kontrol ediyoruz.

```
{ digitalWrite(ledPin, HIGH);  
// Eğer ki gelen ışık düzeyi 200 de fazla ise LED'imizi aktif ediyoruz. }  
else {
```

```
digitalWrite(ledPin, LOW); // Eğer ki gelen ışık düzeyi 200 den fazla ise LED'imizi pasif durumuna  
çekiyoruz. }
```

f) Arduino

Arduino, elektronik donanım ve yazılım temelli bir geliştirme platformudur. Arduino ile öğrenciler de profesyoneller de çok detaylı programlama ve elektronik bilgiye sahip olmadan, temel bilgiler ile hobi amaçlı, eğitim amaçlı veya profesyonel anlamda projeler yapabilirler. Arduino Uno, en yaygın kullanılan ve en çok bilinen modelidir. 2010 yılında kullanıma sunulmuştur.



Arduino Uno ne işe yarar sorusuna tam olarak cevap verelim. Arduino Uno ile çeşitli sensörlerden fiziksel bilgi alabilir, bu bilgiler ile çeşitli deneyler yapabilirsiniz. Ayrıca motor, LED, buzzer gibi uyarıcılardan bir çıktı elde edebilirsiniz. Bu gibi elektronik komponentleri Arduino Uno kartına bağlayarak kontrol etmek için temel bir programlama bilgisi yeterlidir. Projelerin seviyesine göre gerekli olan elektronik ve programlama bilgisi seviyesi de artacaktır. Boyut olarak çok daha küçük ve çok daha büyük modeller olsa da Arduino Uno'nun boyutu projelere göre en standart olanıdır. 14 adet dijital çıkış pini bulunması 14 farklı dijital sensörün ve uyarıcının kontrol edilebileceği anlamına gelmektedir. Bu da birçok proje için yeterli bir sayıdır. Bu dijital çıkışlardan 5 tanesi PWM çıkışıdır. Motorların hızı, LED'lerdeki parlaklık seviyeleri gibi analog olarak kontrol edilmesi istenen uyarıcılar bu PWM pinlerine bağlanarak kontrol edilir. Arduino Uno'daki 6 tane analog giriş ise analog giriş sinyali alabildiğimiz sensörler içindir.

Arduino Uno ile LED yakıp söndürmek gibi en temel uygulamalardan drone, robot, akıllı ev otomasyonu, hırsız alarm sistemi, park sensörü gibi daha gelişmiş projeler de yapabilirsiniz. Bu tamamen ne yapmak istediğinizle alakalıdır. Kısacası Arduino Uno, standart boyutlarda bir kontrol kartı olup, basitten zora birçok uygulamada elektronik devreleri kontrol etmenizi sağlamaktadır.

Arduino Uno Özellikleri Nelerdir?

Haberleşme:

Arduino Uno birçok şekilde haberleşme işlemini gerçekleştirebilir. RX ve TX pinleri ile seri haberleşme imkanı mümkündür. Atmega16u2 USB-seri dönüştürücü de bilgisayarda sanal bir seri port açarak Atmega328 ile bilgisayar arasında haberleşmeyi sağlar. Arduino IDE içerisinde yer alan seri monitör ile Arduino ile bilgisayar arasında metin temelli bilgilerin gönderilip alınmasını sağlar. Arduino ile bilgisayar arasında USB üzerinden bir haberleşme olduğunda Arduino üzerindeki RX ve TX yazan LED'ler yanar. Arduino Uno'da normalde bir tane seri port bulunmaktadır fakat SoftwareSerial kütüphanesi kullanılarak bu sayı yazılımsal olarak artırılabilir.

Atmega328 ayrıca I2C ve SPI portları da sağlamaktadır. Arduino IDE içerisinde yer alan Wire kütüphanesi I2C kullanımını, SPI kütüphanesi de SPI haberleşmesini sağlamak için kullanılır.

Programlama:

Arduino Uno da diğer tüm Arduino'lar gibi Arduino IDE ile programlanır. Detaylı Arduino Kurulumu yazımızı inceleyerek programı kurabilirsiniz.

Bootloader yazılımı bypass edilerek kart doğrudan mikrodenetleyicinin ICSP header kullanılarak ISP programlayıcı ile programlanabilir (*).

USB Kısa Devre ve Aşırı Akım Koruması:

Arduino Uno üzerinden bulunan resetlenebilir sigorta, bilgisayarınızın USB portunu kısa devrelerden veya aşırı akım tüketimi durumlarından korumaktadır. Kart, USB portu üzerinden 500mA'den fazla akım çektiğinde otomatik olarak USB'den aldığı gücü koruma amacıyla kesmektedir. Fazla akım durumu veya kısa devre ortadan kaldırıldığında sigorta normal konuma döner ve tekrar bağlantı kurulur.

Arduino Uno'nun genel anlamda özellikleri aşağıdaki gibidir:

- Mikrodenetleyici: ATmega328
- Çalışma Gerilimi: 5V
- Giriş Gerilimi (önerilen): 7-12V
- Giriş Gerilimi (limit): 6-20V
- Dijital G/Ç Pinleri: 14 (6 tanesi PWM çıkışı)
- Analog Giriş Pinleri: 6
- Her G/Ç için Akım: 40 mA
- 3.3V Çıkış için Akım: 50 mA
- Flash Hafıza: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Saat Hızı: 16 MHz
- Uzunluk: 68.6 mm
- Genişlik: 53.4 mm
- Ağırlık: 25 g
-

Arduino Uno Pinout ve Datasheet

Güç:

Arduino Uno, gücünü USB üzerinden veya adaptör girişinden alabilir. Yani bilgisayarınızın USB girişinden veya bilgisayarınızdan bağımsız olarak bir adaptör veya bataryadan güç elde edebilirsiniz. Doğrudan Vin (+) ve GND (-) pinlerinden de besleyebilirsiniz.

Harici güç kaynağı olarak 6-20V arası limit değerleri bulunmaktadır. Önerilen harici besleme gerilimi ise 7-12 V arasındır. Bunun sebebi 7V altındaki gerilimin stabil çalışmayıp, 12V üzeri gerilimin de aşırı ısınma sebebi olabilemesidir. Kart üzerinde bulunan regülatör sayesinde 7-12V arası gerilim 5V'a düşürülür ve kart bu şekilde çalışır.

- Vin: Harici güç kaynağı için kullanılan pin.
- 5V: Regülatörden çıkan 5V çıkış gerilimini sağlar.
- 3V3: Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
- GND: Toprak (-) pinleridir.

Giriş/Çıkış (I/O):

14 adet dijital, 6 adet analog giriş/çıkış pini bulunmaktadır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maksimum 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir:

Seri Haberleşme- 0 (RX) ve 1 (TX): TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bilgisayardan karta program yüklenirken veya bilgisayar-UNO arasında karşılıklı haberleşme yapılırken de bu pinlerden faydalanılır. Bu sebeple, karta program yüklendiği esnada veya kartla bilgisayar arası iletişim kurulduğunda bu pinleri kullanmamak gerekir.

Harici Kesme (Interrupt)- 2 ve 3: Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için attachInterrupt() fonksiyon sayfasını inceleyebilirsiniz.

PWM- 3,5,6,9,10 ve 11: 8-bit çözünürlükte PWM çıkış pinleridir.

SPI- 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK): SPI haberleşmesi için bu pinler kullanılır.

LED- 13: Kart üzerinde dahili bir LED bulunmaktadır (L harfi ile gösterilmiş). Bu LED 13.pine bağlıdır. HIGH yapıldığında LED yanacak, LOW yapıldığında ise sönecektir.

Analog- A0,A1,A2,A3,A4,A5: 6 adet 10-bit çözünürlüğünde analog giriş pini bulunmaktadır. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'tur. AREF pini ve analogReference() fonksiyonu kullanılarak alt limit yükseltip, üst limit düşürülebilir.

I2C- A4 veya SDA pini ve A5 veya SCL pini: I2C haberleşmesi için bu pinler kullanılır.

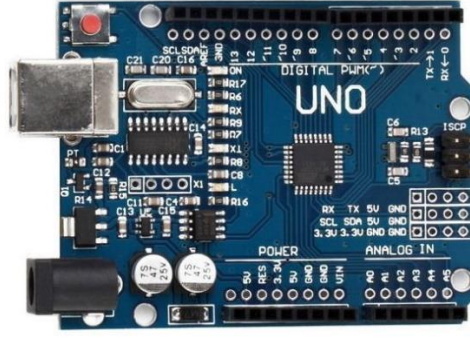
AREF: Analog girişler için ölçüm referansı pini.

Reset: Resetleme işlemi için bu pin LOW yapılır. Bunun yerine kartta bulunan Reset butonuna da basılabilir.

Not: Arduino'da harici kesmede diğer pinler kullanılamaz duruma gelir.

Arduino Uno PWM Pin

Yukarıda da belirttiğimiz gibi PWM pinleri 3,5,6,9,10 ve 11 numaralı pinlerdir. PWM çıkışları ile motor hız kontrolü, ışık parlaklık gibi analog kontrollerin yapılabilmesi sağlanır.



Arduino Uno'yu Bilgisayara Tanıtma

Arduino Uno ile programlamaya başlamadan önce bilgisayarınızda Arduino'nun kurulmuş olması gerekiyor. Detaylı Arduino kurulumuna buradan ulaşabilirsiniz. Kurulum işlemlerini tamamladıktan sonra Arduino Uno'yu bilgisayara tanıtmak için aşağıdaki işlemleri sırasıyla uygulayın:

- Araçlar>Kart>Arduino Uno seçerek kartımızı belirtiyoruz.
- Araçlar>Port seçtikten sonra Arduino Uno'nun bağlı olduğu portu seçiyoruz.
- Böylelikle Arduino Uno, bilgisayarımıza tanımlanmış oluyor. Yükle butonuna basarak programı karta yüklüyoruz ve projemiz çalışmaya başlıyor.

Arduino Uno vs. Mega

Uno da Mega da Arduino'nun en popüler modellerinden ikisi. Temelinde iki kart da aynı yazılım arayüzünde programlanıyorlar fakat tamamen ihtiyaca göre her ikisinin de tercih yerleri değişebiliyor. Çalışma gerilimi, giriş gerilim limitleri gibi özellikleri aynı olsa da temel olarak iki kart arasında bir boyut farkı vardır. Bunun haricinde ise en belirgin özelliklerden bazıları:

Arduino Uno 14 Dijital I/O pine sahipken Arduino Mega tam 54 Dijital I/O pine sahiptir.

Arduino Uno'da bu pinlerin 6 tanesi PWM çıkışı iken Arduino Mega'da 15 adet PWM çıkışı vardır.

Arduino Uno 6 adet Analog giriş pinine sahiptir, Arduino Mega ise 16 adet analog giriş pinine sahiptir.

g) MZ80 Cisim Sensörü

MZ80 kızılötesi cisim algılama sensörü oldukça geniş mesafe ölçüm aralıklarına sahip olan bir kızılötesi mesafeleri ölçüm sensörüdür. Ayrıca görünür ışık ile karışma olasılığı da oldukça düşüktür. Bu sayede de dış faktörlerden dolayı genel olarak yanlış sonuç vermesi çok düşüktür.

Arka tarafındaki vida aracılığıyla hangi mesafeler aralığında ölçüm yapılacağı belirlenir. Bu sistem ayara bağlı dijital bir çıkış verir. Cisim sensörünün çıktısı mesafe değeri değildir. Bir cisim sensörde ayarlanan değerlerde sensörün önüne geldiği zaman dijital bir çıkış verir.

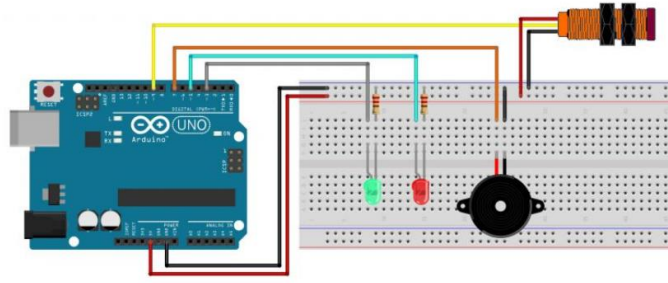
Kızılötesi sensör(IR) bir verici devresi ve bir alıcı vardır, yan yana yerleştirilmiştir. Bir nesne veya kişi sensörün önüne geçtiğinde, IR sinyali alıcı tarafından yansıtılır ve algılanır, bu da çıkış pinini düşük (LOW) sinyale çeker ve istenilen özellik çalıştırılır.

MZ80 kızılötesi cisim algılama sensörü, optik mesafeler sensörlerinden biridir. MZ-80 sensörü en az 2 cm en fazla ise 80 cm arasında yer alan cisimleri algılayabilme kabiliyetine sahiptir. Oldukça basit kurulumu, yüksek hassasiyeti nedeni ile kullanım alanı çok geniştir.

Kullanılabilecek alanlar çok fazla olup bunlar sumo robot, otomasyon sistemleri, engel tespit etme, güvenlik sistemleri, cisim sayma ve de labirent robotu olmaktadır. Mesafeden kaçınmakta olan robot, labirent robotu yapabilmek mümkündür. Bunların yanında takometre ya da enkoder yaparak herhangi tekerin dönüş sayısını ve de hızını hesaplayabilmek mümkündür. Üretim bandından geçmekte olan ürünlerin sayımı gibi bazı projelerde de yaygın kullanılabilmektedir.

- Besleme voltajı: 5v
- Akım Tüketimi : > 25mA (min) ~ 100mA (max)
- Boyutlar : 1.7cm (diameter) x 4.5cm (length)
- Kablo uzunluğu : 45cm
- Nesne algılaması : saydam ya da mat
- Ölçüm aralığı : 3cm – 80cm (engelin yüzeyine bağlı olarak değişebilir)
- NPN çıkış (normal 1)
- Çalışma sıcaklık aralığı : -25 °C ~ 55 °C
- Sensörde sadece 3 çıkış bulunur bunlar: Kırmızı(Vcc) , Siyah(GND) ve Sarı(sinyal)

Bir örnek ile MZ80 Arduino Uno üzerinde inceleyelim. Örneğimizde sensörün önünde hareket algılandığında bu sensörü bir varlık detektörü olarak kullanacağız, bir led'i açıp kapatacağız. Ayrıca kısa bir süre için buzzer tetikleyeceğiz.



```
int YESIL_LED = 3;
int KIRMIZIL_LED = 5;
int BUZZER = 7;
int MZ80_PINI = 9;
int SENSOR_DURUMU = 0;
void setup()
{
  // SENSOR PININI GIRIS OLARAK AYARLA
  pinMode(MZ80_PINI, INPUT);
  // PINLERI CIKIS OLARAK AYARLA
  pinMode(YESIL_LED, OUTPUT);
  pinMode(KIRMIZIL_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
}
void loop()
{
  SENSOR_DURUMU = digitalRead(MZ80_PINI);
  if (SENSOR_DURUMU == 0)
  {
    // EĞER SENSORE YAKLAŞILMIŞSA
    digitalWrite(YESIL_LED, LOW);
    digitalWrite(KIRMIZIL_LED, HIGH);
    // BUZZER SINYAL VER VE KIRMIZI LEDI YAK
    for (int i = 1; i <= 3; i++)
    {
      digitalWrite(BUZZER, HIGH);
      delay(100);
      digitalWrite(BUZZER, LOW);
      delay(100);
    }
    delay(100);
  }
  else
  {
    // SENSORDE SINYAL YOKSA KIRMIZI, YESIL LEDI VE BUZZERI DURDUR
    digitalWrite(YESIL_LED, HIGH);
    digitalWrite(KIRMIZIL_LED, LOW);
    digitalWrite(BUZZER, LOW);
  }
}
```

h) Servo Motor

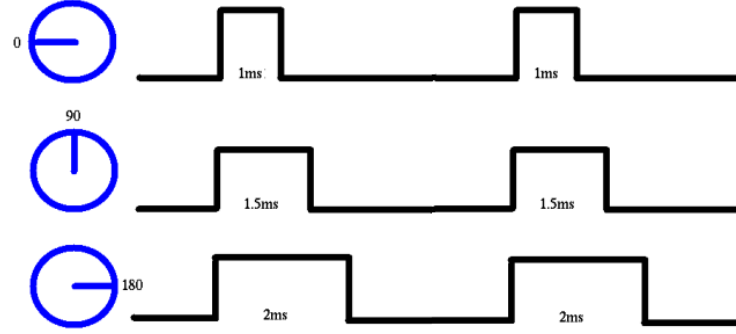
Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzeneştir. Servo motorlar, robot teknolojilerinde en çok kullanılan motor çeşidi olmakla birlikte, RC (Radio Control) uygulamalarda da kullanılmaktadırlar. RC Servo Motorlar ilk olarak uzaktan kumandalı model araçlarda kullanılmışlardır. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.

İçerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüyü ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolcünden veya uzaktan kumandadan sağlanabilmektedir. Servo, her 20 ms içerisinde bir pals değeri okumaktadır. Pals uzunluğu motorun dönüşünü belirlemektedir. Örnek olarak 1.5 ms'lik bir pals, motorun 90

derece pozisyonunu almasını sağlayacaktır (Nötr Pozisyon). Servolar hareket etmeleri için bir komut aldıklarında önce istenilen pozisyona hareket ederler, sonrasında ise o pozisyonda kalırlar. Servolar bulundukları pozisyonu korurken kendilerine dışarıdan bir güç uygulandığında bu güce direnirler. Bulundukları konumu sonsuza kadar koruyamazlar, pozisyonlarını koruyabilmeleri için palsin tekrar edilmesi gerekebilir. Hareket etmeleri için gereken pals genişliklerinin minimumları ve maksimumları vardır ve bu değerler değişkendir. Fakat genellikle minimum pals genişliği 1 ms, maksimum pals genişliği ise 2 ms'dir.

Aşağıdaki şekilde genellikle sahip oldukları PWM değerleri vardır. 1 ms duty cycle değerinde 0° , 2 ms değerinde ise 180° pozisyonunu almış olur. (Daha önce belirttiğim gibi bu limit bazen 360° de olabilmektedir.)



Servo Motor Çeşitleri

AC-DC Servo:

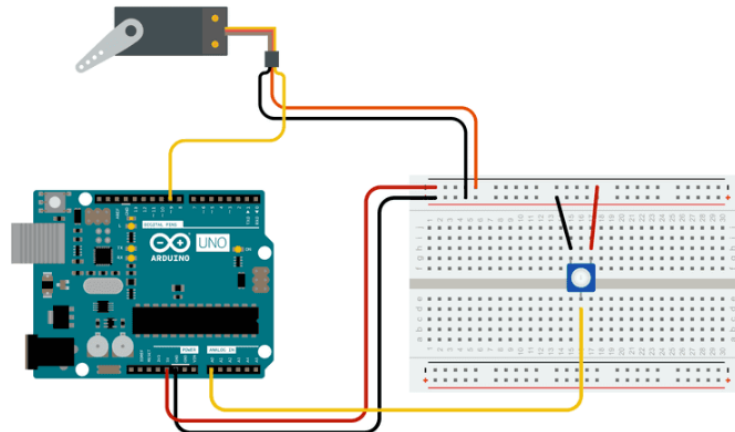
Servo motorlar genel olarak AC Servo ve DC Servo olarak ikiye ayrılırlar. AC Servo Motorlar endüstride kullanılmaktadır. Bu yazımızda bahsettiğimiz RC Servolar, DC Motorlardır.

Dijital-Analog Servo:

RC Servolar devre yapılarına göre Analog Servo ve Dijital Servo olmak üzere ikiye ayrılmaktadırlar. Dijital servolar, analog servolara göre daha yüksek frekansta çalışırlar. Bu sayede dijital servolar komutlara daha net ve hızlı tepki verirler, daha iyi bir tutma torku elde ederler. Hızlanmaları daha yumuşak gerçekleşir. Analog servolara göre dezavantajları ise daha fazla enerji harcaması sebebiyle pil ömrünü daha çabuk tüketmeleridir.

RC Servolar, içlerindeki mekanizmada bulunan motorlara göre de değişkenlik göstermektedir. Çekirdeksiz motorlar yapılarında mıknatıs bulundurmadan kablolar yardımıyla manyetik alan oluşturmaktadır. Bu sebeple hafiftirler, daha çabuk tepki verirler ve daha yumuşak hareket ederler. Fırçasız motorların avantajı ise daha verimli ve daha çok güç üretebilmeleridir.

Servo motorların üç kablosu vardır: güç, toprak ve sinyal. Güç kablosu tipik olarak kırmızıdır ve Arduino kartındaki 5V pinine bağlanmalıdır. Topraklama kablosu tipik olarak siyah veya kahverengidir ve kart üzerindeki bir topraklama pimine bağlanmalıdır. Sinyal pini tipik olarak sarı veya turuncudur ve kart üzerindeki PWM pinine bağlanmalıdır. Bu örneklerde, pin numarası 9'dur. Düğme örneği için, potansiyometreyi, iki dış pimi güce (+5V) ve toprağa bağlanacak ve orta pimi de ağı bağlanacak şekilde kablolayın.A0gemide. Ardından servo motoru +5V, GND ve pin 9'a bağlayın.



```

#include <Servo.h>
Servo myservo;
int potpin = 0;
int val;

void setup() {
myservo.attach(9);
}
void loop() {
val = analogRead(potpin);           //0 ile 1023 deęerleri arasında deęerimizi okuruz
val = map(val, 0, 1023, 0, 180);    // servo motoru 0 ile 180 arasına map ederiz
myservo.write(val);                 // Servo motorun pozisyonunu ayarlanır
delay(15);                          // Servo motor beklemekte
}

```

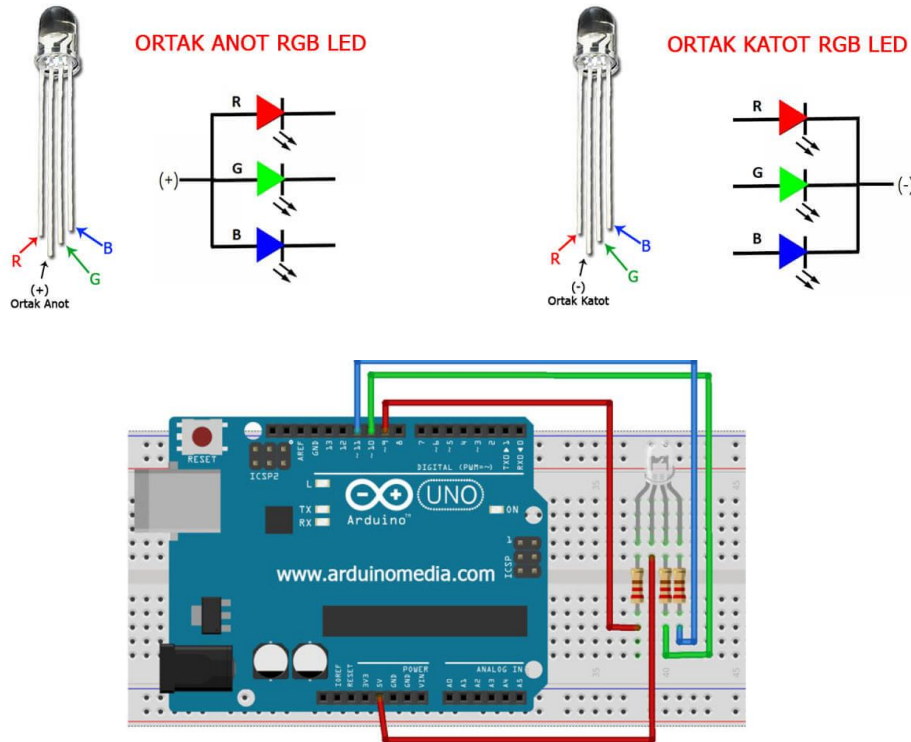
i) RGB Led

RGB led, ierisinde kırmızı (Red), Yeşil (Green) ve Mavi (Blue) olmak üzere 3 adet led bulunan Led çeşitidir. İsmi bu 3 rengin ingilizce baş harflerinden alır. Red-Green-Blue (RGB). RGB Led ierisinde bulunan 3 renk led sayesinde bu 3 rengin dıřında bir ok renkte ıřık verme ellięine sahiptir. RGB Led renk deęiřtirmek iin bu 3 rengin karıřımlarını kullanır. Bu nedenle bir ok arduino projesinde kullanılmaktadır. RGB Led arduino baęlantısı yapılırken renk bacakları arduino PWM zellikli 3,5,6,8,9,10,11 numaralı pinlere baęlanmalıdır.

RGB Led 4 bacaklı led řeklinde veya basit bir devre kartı üzerine entegre edilmiř halde satılmaktadır. İki tr RGB led kullanımı aynıdır.

RGB led kullanımında dikkat edilmesi gereken dięer bir konu ise kullandığınız RGB ledin anot ya da katot olması durumudur. RGB led anot ya da katot olma durumuna gre arduino baęlantısı ve arduino kodları farklılık gstermektedir.

Peki kullandığımız RGB ledin ortak anot ya da ortak katot zellięini nasıl anlarız? Burada yapacağınız basit bir iřlem var. Biraz sonra anlatacağıımız baęlantı řemasını ve kodları oluřturduğunuzda eęer RGB led hi ıřık vermiyorsa sadece uzun bacaęını 5V ise Gnd, Gnd ise 5V olarak deęiřtirmeniz yeterli olacaktır. RGB led uzun bacaęı 5V pinine baęlı iken yanıyorsa ortak anot RGB led, Gnd pinine baęlı iken yanıyorsa ortak katot RGB led demektir.



Ortak anot RGB led arduino baęlantısı yapılırken en uzun bacak arduino 5V pinine baęlanmalıdır. Dięer renk bacakları ise arduino pwm pinlerinden birine baęlanmalıdır. Bu rnek iin 9,10 ve 11 numaralı pwm pinlerini kullanacağız.

```

int R=9;
int G=10;
int B=11;
void setup() {
pinMode(R,OUTPUT);
pinMode(G,OUTPUT);
}

```



```
pinMode(B,OUTPUT);  
}  
void loop() {  
  //Kırmızı renk elde etmek için...  
  analogWrite(R,0);  
  analogWrite(G,255);  
  analogWrite(B,255);  
  delay(500);  
  //Yeşil renk elde etmek için...  
  analogWrite(R,255);  
  analogWrite(G,0);  
  analogWrite(B,255);  
  delay(500);  
  //Mavi renk elde etmek için...  
  analogWrite(R,255);  
  analogWrite(G,255);  
  analogWrite(B,0);  
  delay(500);  
}
```

Kodlarda dikkat etmemiz gereken durum ise; ortak anot RGB led kullandığımız için yanmasını istediğimiz renk pinine 0 pwm sinyalini, yanmasını istemediğimiz renk pinine 255 pwm sinyalini gönderiyoruz. Farklı renkler elde etmek için tüm renk pinlerine 0-255 arasında değişen farklı pwm sinyalleri gönderebilirsiniz.

4- GERÇEKLEŞTİRİLEN PROJENİN FARKLI AÇILARDAN FİZİKSEL RESİMLERİ





5- ÖRNEK SEÇİLEN PROJE NE İDİ? BU PROJEDE ÖRNEK PROJEDEN NE GİBİ FARKLILIKLAR YAPILDI? HEM DONANIM HEM DE YAZILIM AÇISINDAN FARKLILIKLARI VE ÖZGÜN KATKILARINIZI AYRINTILI BELİRTİNİZ

Örnek proje için [tıklayınız](#).

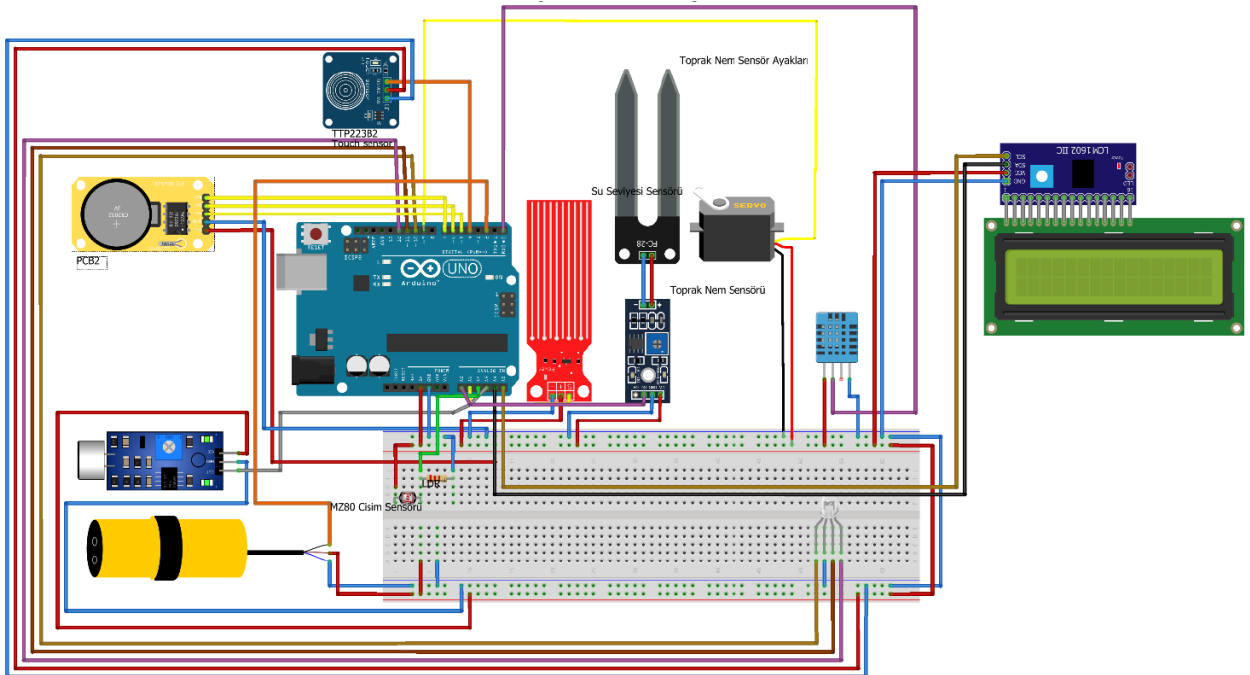
Eklediğimiz kısımlar:

- Saat modülü ile gerçek zaman takibi
- Lcd ekran ile bireylerle iletişim
- Hava nem sensörü ile gerçek zamanlı olarak havanın nem ve sıcaklık değerlerini okumak
- Buzzer eklentisi ile bitki susuz kaldığında uyarı sistemi
- Mz80 sistemi ile cisim algılayıp, lcd ekrana mesajlar yazdırma
- Dokunma sensörü ile lcd ekrana farklı içerikler yazdırma imkanı
- Ldr ile ortamın karanlık olmasına bağlı olarak Rgb ledi yakmak
- Buton eklentisi ile sistemi kapatmak

Eksik kısımlar:

- Hazırladığımız projenin, linkte olan projeye göre eksik kısmı yoktur.

6- DONANIM BREADBORD - ARDUİNO BAĞLANTI ŞEMASI VE AÇIKLAMASI:



fritzing

Yukarıda belirtilen şemada mavi kabloların hepsi GND yani topraklama uçlarını temsil ederken görünen tüm kırmızı kablolar VCC yani güç alınan kabloyu temsil etmektedir.

Şemanın sol altından bulunan MZ80 Kızılötesi cisim sensörü önünde bir engel olması durumunda dijital 1 değerini arduinoya iletir. Böylelikle sakı aslında bir insanın önünden geçtiğini fark edecektir.

MZ80 sensörünün hemen üstünde ses kartı bulunmaktadır. Ses kartıyla insanlara gerektiği zamanlarda sesli uyarı yapmayı hedeflemektedir. Dijital pin olarak bağlanan ses kartı HIGH olması durumunda ses çıkaracaktır. Arduino unonun hemen solunda bulunan saat modülü sayesinde gün/saat/dakika/tarih gibi değerleri elde edebileceğiz. Böylelikle akıllı saksı aynı zamanda bir dijital saat işlevi de görecektir.

Arduino unonun üzerinde bulunan dokunma sensörü bir buton gibi çalışır. Ama gerek kullanım kolaylığı gerekse gelişen teknolojiyle birlikte dokunma sensörü kullanmak bir ürünün kullanımını kolaylaştırır. Dokunma sensörü sayesinde her basıldığında LCD ekran üzerinden sırasıyla sıcaklık, nem, toprak nem miktarı, havanın kalitesi gibi değerleri görüntüleme imkanı sunulur.

Şemada arduino unonun sağında bulunan kırmızı su seviyesi sensörü sayesinde akıllı saksının su deposundaki su azaldığında bu kişiye bildirilir, böylelikle kullanıcı su deposunu doldurabilecektir.

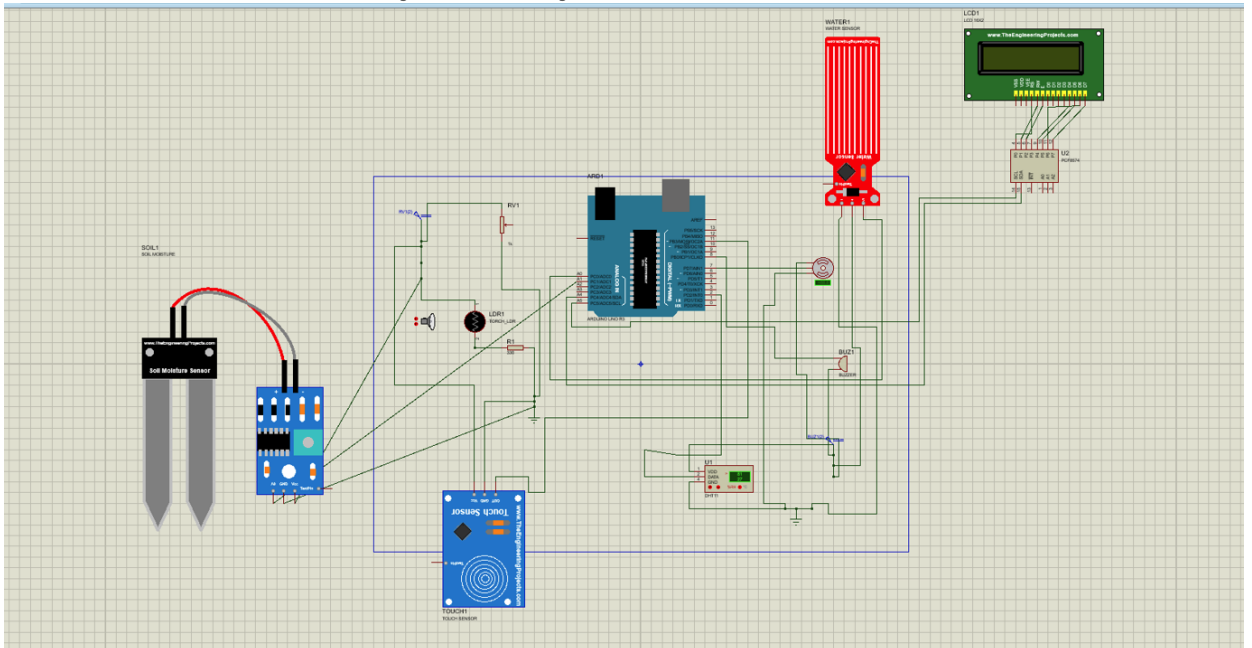
Hemen onun sağında bulunan toprak nem sensörü, toprağın su miktarı yani nem miktarı azaldığında otomatik olarak sağ tarafında şematize edilmiş servo motor ile iletişime geçerek su deposundan saksıya su akmasını sağlayacaktır. Böylelikle bitki için ideal ortam sağlanmış olur. Toprak nem sensörünün sağ altında mavi bir hava kalitesi sensörü şematize edilmiş olarak bulunur. Bu sensör sayesinde su sıcaklığı, nemi gibi değerleri elde edebiliriz. Bu hem kişi için bilgi verici içerik olacak aynı zamanda kişi bu veriler uyarınca bitkinin konumunu değiştirme gibi aksiyonlar alabilir. Şemanın en sağında LCD ekran ve I2C modülü bulunur. Akıllı saksı bu parçalar ile kişiler ile iletişim kuracak, sat gösterecek ve bilgiler verecektir.

Breadbord üzerinde LDR yani ışık sensörüne ve RGB Led'e yer verilmiştir. LDR ile havanın yetersiz ışık aldığı yani akşam olduğunu anlarda RGB led ile sağlanan mor ışık bitkinin daha çok fotosentez yapmasını sağlayacak böylelikle eğer ki bitkimiz bir meyve/sebze üretiyorsa daha hızlı şekilde olgunlaşıp, büyüyecektir.

7- DONANIM SİSTEMİNİN ÇALIŞMA MANTIĞINI ANLATINIZ:

Donanım sistemi batarya tarafından beslenmektedir. LCD ekran ile iletişim kurmak üzere I2C haberleşme modülü kullanılmaktadır. Analog olarak su seviyesi ölçen sensör A0 pinine bağlanmıştır, toprak nemini ölçen sensör A1 pinine, hava nem ve sıcaklığını ölçen sensör dijital 2 pinine, buzzer dijital 3 pinine, servo motorun data ucu dijital 4 pinine, mz 80 cisim sensörü dijital 5 pinine, RGB ledin sırasıyla kırmızı-yeşil-mavi ayakları 5-6-7. Dijital pinlere, saat modülünün CLK değerini taşıyan ucu dijital 8, saat modülünün data değeri taşıyan pini dijital 9, saat modülünün reset değerini taşıyan ucu dijital 10, dokunma sensörü dijital 11 ve analog olan LDR ışık sensörü A2 pinine bağlanmıştır. LDR sensörü sorunsuz çalışabilmesi için 330 ohm direnç devreye eklenmiştir. Devrenin tamamı mikro denetleyici olan arduino udo tarafından aşağıda belirtilen kodlara göre yönlendirilecek şekilde dizayn edilmiştir.

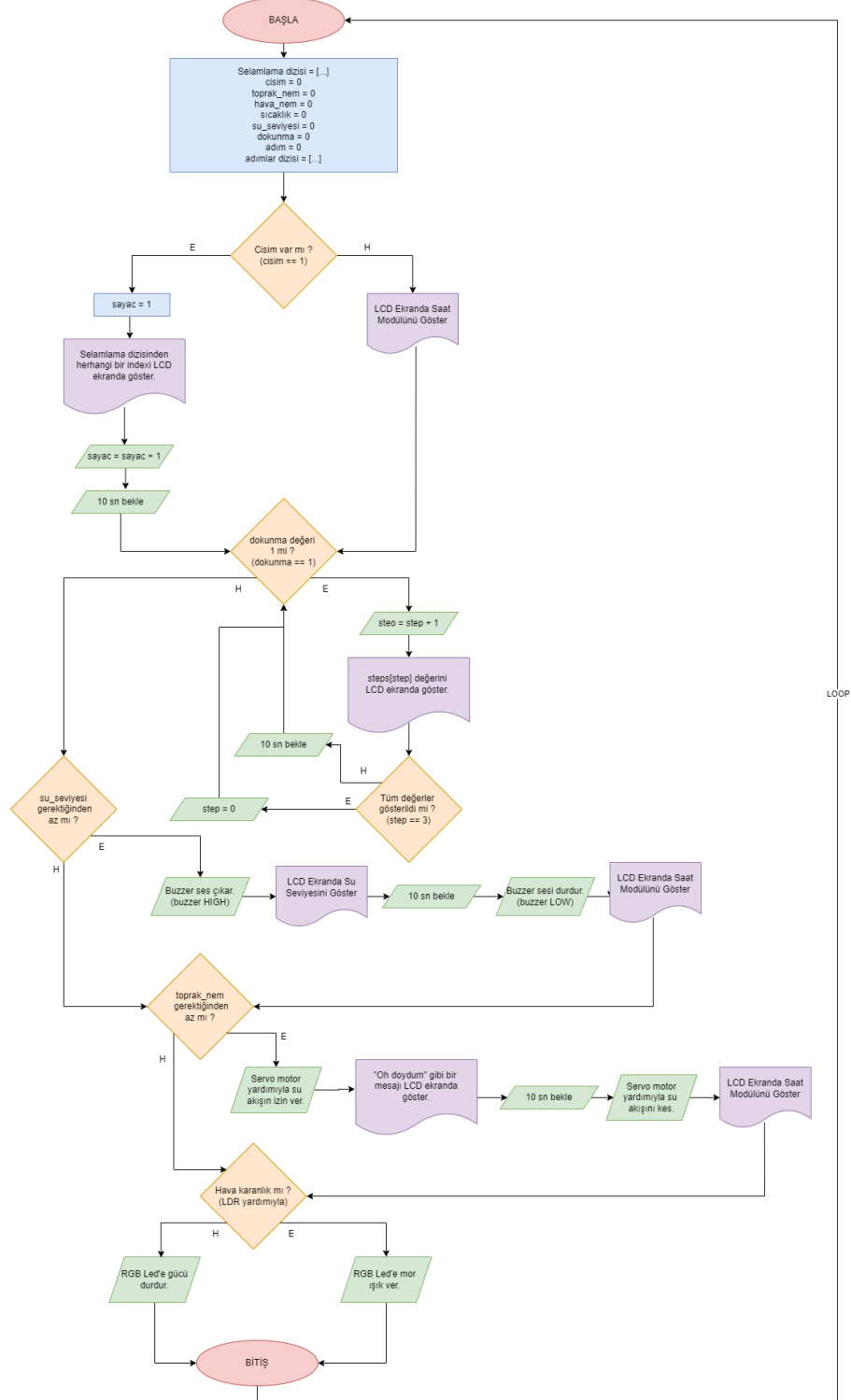
8- PROTEUS - ARDUİNO BAĞLANTI ŞEMASI VE AÇIKLAMASI:



Öncelikle projenin daha karışık olmaması sebebiyle iki tane Dc gerilim kaynağı ve iki tane de ground çizilmiştir. Daha sonra 6 tane sensörümüzden 4 tanesi Proteusda bulunmamaktadır. 2 tanesi zaten Proteusda bulunmamaktadır. MZ80 mesafe sensörü bulunmamamız sebebiyle kullanamadık. Su seviyesi A0 pinine bağlanmıştır. Su seviyesini + ve - kısımları sağdaki Dc kaynağına ve sağdaki ground kısmına bağlanmıştır. Soldaki toprak nem sensörü ise sola daha yakın olduğu için soldaki Dc kaynağı ve ground'a bağlanmıştır.

Toprak nem sensörü analog çalışan bir sensör olduğundan Arduino'muzun A1 girişine bağladık. Sol altta gördüğümüz üzere Touch sensörü ise yine soldaki ground ve Dc kaynağına bağlanmıştır. Dijital pinimizi ise 11'e bağladık. Sağ alt kısımda ise Hava nem/sıcaklık sensörü bulunmaktadır. Bu sensörümüzü ise sağdaki ground ve Dc kaynağına bağladık. Arduino'dan dijital pinlerden 2 numara pine bağladık. Yukarı sağ bölümde ise Lcd ekranımız vardır. Lcd ekranını bağlamamız için çok fazla pin ihtiyacımız olduğundan bu ihtiyacımızdan kurtulmak için pcf8574 i2c modülü kullanılmıştır. Pcf8574 i2c modülünün SCL pini Arduino'numuzun A5 pinine bağlıyoruz. Daha sonra SDA pini Arduino'numuzun A4 pinine bağlıyoruz. Buzzer'ın bir ucu Dc kaynağına bağlanmıştır. Buzzer'ın diğer ucu 8 numaralı pine bağlanmıştır. Yine Servo motorumuzu da sağdaki ground ve Dc kaynağına bağladık. Diğer bacağını ise 7 numaralı pine bağladık.

9- ARDUİNO YAZILIM PROGRAMININ AKIŞ DİYAGRAMI VE AÇIKLAMASI:



MZ80 cisim sensörü ile bir engel yani bu senaryo için bir kişi algıladığında selamla içerisinde kişiyi selamlayacağı farklı cümlelerden birini seçip bunu ekranda yazdıracaktır. Böylelikle kişinin hedefi sadece ordan geçmek dahi olsa saksıyı kontrol ederek bir gerekliliği var mı kontrol edebilir.

Eğer ki bir cisim görmemiş ise dijital saat gibi çalışarak ekranda saat bilgisine yer verecektir. Dokunma sensörü sayesinde ilk dokunma da değer 1 arttırılacaktır. Toplam da 4 farklı bilgi vereceği için kullanıcı butona üç kere bastığında tekrar saati göstermeye dönecektir. Eğer ki dokunma değeri yok ise devre su seviyesi kontrolüne geçecektir. Eğer ki su deposundaki seviye yetersiz ise buzzer ile kullanıcıyı uyararak LCD ekran üzerinden bu uyarıyı belirtecektir. Su seviyesinin yeterli olduğu durumda ise toprak nem seviyesini kontrol edecektir. Toprak neminin yeterli olmaması durumunda servo motor yardımıyla depodan saksıya su geçişi sağlayacak ve toprak neminin ideal seviyeye dönmesini sağlayacaktır. Nem seviyesi yeterli ise havanın karanlık olup/olmasına bakacaktır. Işığın yetersiz olması durumunda RGB lede güç vererek bitkiye ışık sağlayacaktır. Bitkiler en hızlı fotosentezi mor ışıktaki yaptığından dolayı mor ışık verecektir. Eğer hava aydınlanması iyiye ki bu senaryoda sabah saatleri olur lede güç vermez ve tekrar şemanın başına dönerken bu işlemi bir loop olacak şekilde kontrol eder.

10- YAZILIM PROGRAMI VE HER BİR SATIRDA AÇIKLAMALAR (COMMENTS):

```
// Servo Motorun Kütüphanesi Dahil Edilir
#include <Servo.h>

// Saat Modülü Kütüphanesi Dahil Edilir.
#include <virtuabotixRTC.h>

// LCD ve I2C Modülü Kütüphanesi Dahil Edilir.
#include <LiquidCrystal_I2C.h>

// Servo'dan bir obje üretilir
Servo ServoNesnesi;

// Arduino uno üzerine takılı tüm pinler tanımlanır
#define SU_SEVIYE A0
#define TOPRAK_NEM A1
#define HAVA_NEM 2
#define BUZZER 3
#define SERVO 4
#define MZ80 5
int RGB_R = 6;
int RGB_G = 7;
int RGB_B = 12;
#define SAAT_CLK 8
#define SAAT_DAT 9
#define SAAT_RST 10
#define DOKUNMA 11
#define LDR A2

// LCD üzerinde cisim görüldüğünde yazacak yazı
String SELAMLA = "HOSGELDIN";

// LCD den 'lcd' adında bir nesne üretilmesi
LiquidCrystal_I2C lcd(0x27,16,2);

// saat modülünden Saat adında bir nesne üretilmesi
virtuabotixRTC Saat(8, 9, 10);

// hava nem-sıcaklık sensörünün kütüphanesinin dahil edilmesi.
#include <dht11.h>

//dht11 kütüphanesinden obje oluşturulması
dht11 dht_sensor;

// dokunma değerini kontrol barındıran değişken
int touch_count = 0;
void setup() {

// Lcd başlat
```

```
lcd.begin();

// saat modülüne saat dakika saniye gün ay yıl gibi değerleri ver.
Saat.setDS1302Time(15, 57, 18, 7, 12, 06, 2022);

// servo motorun hazırlanması
ServoNesnesi.attach(SERVO);

// bileşenlerin (dolayısıyla pinlerin) girdi veya çıktı şeklinde tanımlanması
pinMode(SU_SEVIYE, INPUT);
pinMode(TOPRAK_NEM, INPUT);
pinMode(BUZZER, OUTPUT);
pinMode(SERVO, OUTPUT);
pinMode(MZ80, INPUT);
pinMode(RGB_R, OUTPUT);
pinMode(RGB_G, OUTPUT);
pinMode(RGB_B, OUTPUT);
pinMode(DOKUNMA, INPUT);
pinMode(LDR, INPUT);

// seri monitörden izlemek için Serial monitörün başlatılması
Serial.begin(9600);
}

void loop() {

    // START CODE
    // mz80 sensörü değer okur.
    int MZ80_STATUS = digitalRead(MZ80);
    // su seviyesi ölçülür
    int su_seviyesi_deger = analogRead(SU_SEVIYE);
    // hava ve nem değerleri ölçülür
    int chk = dht_sensor.read(HAVA_NEM);
    // toprak nem değeri ölçülür
    int toprak_deger = analogRead(TOPRAK_NEM);
    // dokunma sensörüne dokunulup/dokunulmadığı ölçülür.
    int dokunma_check = digitalRead(DOKUNMA);
    // LDR sensöründen ıřık değeri ölçülür
    int isik_degeri = analogRead(LDR);
    // 0.5 sn bekleme
    delay(500);
    // MZ80 cisim görüyor mu? Evet ise değeri 0 olur.
    if(MZ80_STATUS == 0) {
        // Cisim var ise lcd ekran temizlenir ve SELAMLA değeri olan 'HOřGELDİN' ekrana yazılır
        lcd.clear();
        lcd.setCursor(0,0); // İlk satırın başlangıç noktası
        lcd.print(SELAMLA);
        // 5 sn bekleme
        delay(5000);
    } else {
        // SAAT YAZDIR
        lcd.clear();
        lcd.setCursor(0,0); // İlk satırın başlangıç noktası

        // SAat değışkeni yardımıyla saatin ekrana yazılması
        Saat.updateTime();
        lcd.print(Saat.dayofmonth);
        lcd.print("/");
        lcd.print(Saat.month);
        lcd.print("/");
        lcd.print(Saat.year);
    }
}
```

```

    lcd.print(" ");
    lcd.setCursor(0,1);
    lcd.print(Saat.hours);
    lcd.print(":");
    lcd.print(Saat.minutes);
    // 3 sn beklenir
    delay(3000);
}
// Dokunma sensörüne dokunma var mı kontrol edilir.
if(dokunma_check == 1) {
    // var ise dokunma değeri bir arttırılır
    touch_count += 1;
    // her zaman mod 4'ü alınır çünkü dokunmaya bağlı olarak 4 farklı bilgi lcd ekrandan
    verilmektedir.
    // mod 4 çünkü ilk değer sıfır (0-1-2-3-4 her bir değer için farklı bir bilgi lcd
    ekranda gösterilir.)
    touch_count = touch_count % 4;

    // Touch değerini takip etmek için geliştirme sürecinde kullanılan kodlar
    Serial.print("ICERI : ");
    Serial.println(touch_count);

}

// dokunma sensörüne bir kere basıldıysa
if(touch_count == 1) {
    // lcd temizle
    lcd.clear();
    lcd.setCursor(0,0); // İlk satırın başlangıç noktası
    lcd.print("SU SEVIYESI: ");
    lcd.setCursor(0,1); // İkinci satırın başlangıç noktası
    lcd.print(su_seviyesi_deger); // su seviyesi değerini ekrana yazdır
    // 3 sn bekle
    delay(3000);
    // su seviyesi az ise ve mız80 bir cisim görmüyor ise aciktim işlevi çağrılır
    if (su_seviyesi_deger < 165 && MZ80_STATUS == 1){
        aciktim();
    }
} else if (touch_count == 2) { // dokunma sensörüne iki kere dokunulmuşsa
    // lcd temizle
    lcd.clear();
    lcd.setCursor(0,0); // İlk satırın başlangıç noktası
    lcd.print("Nem Orani: ");
    lcd.print((float)dht_sensor.humidity, 2); // nem oranını yazdır
    lcd.setCursor(0,1); // İkinci satırın başlangıç noktası
    lcd.print("Sicaklik: ");
    lcd.print((float)dht_sensor.temperature, 2); // sıcaklık değerini yazdır
    // 3 sn bekle
    delay(3000);
    // su seviyesi az ise ve mız80 bir cisim görmüyor ise aciktim işlevi çağrılır
    if (su_seviyesi_deger < 165 && MZ80_STATUS == 1){
        aciktim();
    }
} else if (touch_count == 3) { // Dokunma sensörüne üç kere dokunulmuşsa
    // lcd ekranı temizle
    lcd.clear();
    lcd.setCursor(0,0); // İlk satırın başlangıç noktası

    Saat.updateTime(); // güncel zamanı al
    lcd.print(Saat.dayofmonth); // günü yazdır
    lcd.print("/");
    lcd.print(Saat.month); // ayı yazdır
    lcd.print("/");

```



```

    lcd.print(Saat.year); // yılı yazdır
    lcd.print(" ");
    lcd.setCursor(0,1); // İkinci satırın başlangıç noktası
    lcd.print(Saat.hours); // Saati yazdır
    lcd.print(":");
    lcd.print(Saat.minutes); // dakikayı yazdır

    delay(3000);
    // su seviyesi az ise ve mz80 bir cisim görmüyor ise acıktım işlevi çağrılır
    if (su_seviyesi_deger < 165 && MZ80_STATUS == 1){
        acıktım();
    }
}

// su seviyesi az ise ve mz80 bir cisim görmüyor ise acıktım işlevi çağrılır
if (su_seviyesi_deger < 165 && MZ80_STATUS == 1){
    acıktım();
}

// Toprak nem seviyesi düşük mü ?
if(toprak_deger< 1500){
    ServoNesnesi.write(90); /* Motorun mili 90. dereceye donuyor ve su akmaya başlar*/
    delay(4000); // dört saniye boyunca su akar
    ServoNesnesi.write(270); /* Motor mili -90. dereceye donuyor ve su akışı durur*/
    delay(1000); // bir saniye bekle
}

// Işık değeri idealden az mı ? Hava karanlık mı ?
if(isik_degeri<200){
    renkAyarla(80, 0, 80); // Renk ayarla işlevi çağrılır
} else {
    // Hava aydınlık ise RGB led'in her renk değeri için LOW verilir ve ışık söner.
    analogWrite(RGB_R, LOW);
    analogWrite(RGB_G, LOW);
    analogWrite(RGB_B, LOW);
}

}

// veirlen kırmızı yeşik ve mavi oranına göre renk ayarlayan işlev
void renkAyarla(int kırmızı, int yesil, int mavi){
    kırmızı = 255-kırmızı;
    yesil = 255-yesil;
    mavi = 255-mavi;
    analogWrite(RGB_R, kırmızı);
    analogWrite(RGB_G, yesil);
    analogWrite(RGB_B, mavi);
}

// Çarıldığında buzzer çalıştıran ve ekrana acıktım yazdıran işlev
void acıktım(){
    //tone(BUZZER, 1000); // Send 1KHz ses sinyali çıkar
    delay(1000); // 1 saniye boyunca çak
    lcd.clear(); // lcd temizle
    lcd.setCursor(0,0); // İlk satırın başlangıç noktası
    lcd.print("ACIKTIM !!"); // acıktım yaz
    noTone(BUZZER); // Buuzer durdur
    delay(2000); // 2 sn bekle
}

```

11- GRUP ÜYELERİNİN İŞ BÖLÜMÜNÜ AYRINTILI YAZINIZ. KİMLER HANGİ İŞLERİ YAPTI İSİM İSİM BELİRTİNİZ.

Muhammed Saltuk Yaşar (18360859016):

- Proje Montajı
- Doküman Hazırlama
- Arduino ve Arduino çeşitleri ve farkları araştırılması
- Hava ısı/nem sensörü araştırılması ve örnek kod yapısı
- Ldr ışık sensörü araştırılması ve örnek kod yapısı
- Toplantılara katılım %100
- Malzemelerin Temini
- Akış şeması çizim
- İş bölümü yazımı
- Arduino kod yazımı

Mehmet Salih Önder (18360859028):

- Proje Montajı
- Doküman Hazırlama
- Yağmur sensörü araştırılması ve örnek kod yapısı
- Toprak-nem sensörü araştırılması ve örnek kod yapısı
- Jumper ve direnç araştırılması ve kullanılması
- Fritzing çizimi
- Toplantılara katılım %100
- Akış şeması çizim
- Proteus – Arduino bağlantı şeması ve açıklaması
- Arduino kod yazımı
- Malzeme temini

Fatih Es (18360859021):

- Proje Montajı
- Doküman Hazırlama
- Fritzing çizimi
- RTC(saat modülü) araştırılması ve örnek kod yapısı
- Buzzer araştırılması ve örnek kod yapısı
- Lcd ekran ve I2C Modülü araştırılması ve örnek kod yapısı
- Toplantılara katılım %100
- Akış şeması çizim
- Arduino kod yazımı
- Yazılım programı ve anlatımı yorum satırları
- Gerekli durumlarda lehim işlemi
- Buton eklentisi yapımı

Ceyda Tekin (18360859025):

- Proje Montajı
- Doküman Hazırlama
- MZ80 cisim sensörü araştırılması ve örnek kod
- Servo motor araştırılması ve örnek kod
- RGB led araştırılması ve örnek kod yapısı
- Toplantılara katılım %100
- Akış şeması çizim
- Arduino kod yazımı
- Gerçekleştirilen projenin fotoğraf çekimi
- Donanım sisteminin çalışma mantığı