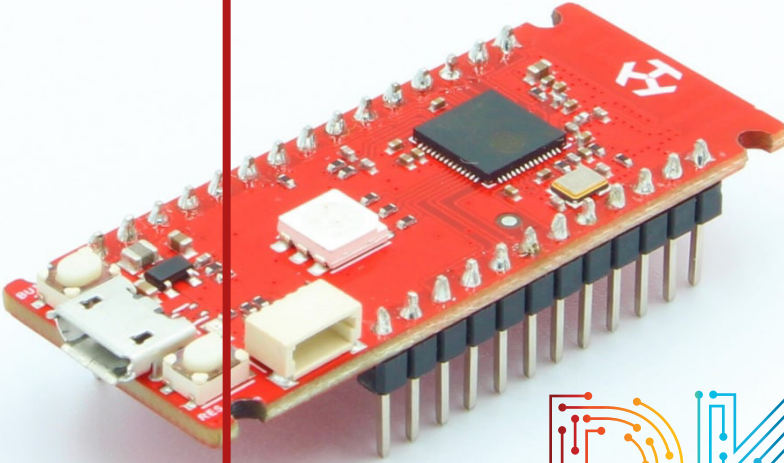


DENEYAP MiNi

UYGULAMA KİTABI

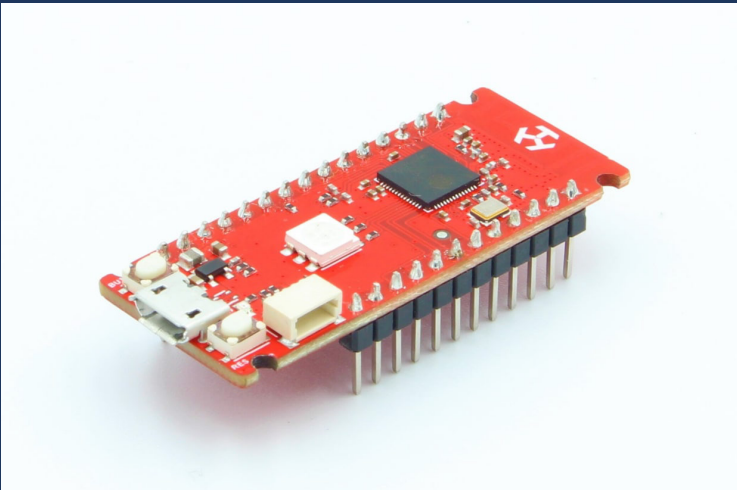


Deneyap Kart ile Milli Teknoloji Hamlesi yolculuđuna hoř geldiniz. Türkiye Teknoloji Takımı Vakfı olarak ölkemizin mühendislik kaynakları kullanılarak üretilen Deneyap Mini ile elektronik ve kodlama alanına ilgisi olan sizler için bir yol haritası oluşturmak istedik. Elinizdeki uygulama kitapçığı ile bu dünyaya adım atabilirsiniz. Kitapçık içerisinde başlangıç seviyesinden ileri seviyeye kadar uygulamalar adım adım anlatılmıştır. Sizler bu adımları uygularken hem öğrenecek hem de hayalini kurduğunuz yapay zeka ve nesnelerin interneti sistemlerini kendiniz yapmaya hazır olacaksınız.

Sorularınız ve önerileriniz için iletisim@deneyapkart.org adresinden bizimle iletişime geçebilirsiniz. Paylaşmak istediđiniz projelerinizi mth.tc/L8 adresi üzerinden web sitemize yükleyebilirsiniz.

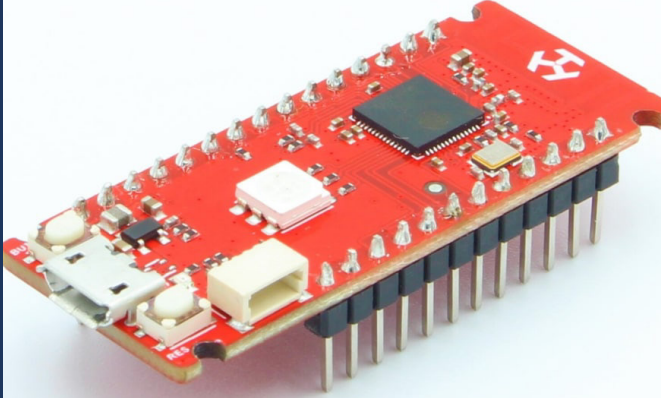
Teknoloji Geliřtiren Bir Türkiye Hedefiyle..

T3 Vakfı Deneyap Kart Ekibi



İÇİNDEKİLER

Arduino Ide Uygulamasının Kurulumu	1
Deneyap Mininin Arduino Ideye Eklenmesi	5
Proje No: 1 Deneyap Mini ile Led Yakma	10
Proje No: 2 Deneyap Mini Buton ile Led Yakma	14
Proje No: 3 Deneyap Mini ile Analog Okuma-Seri Haberleşme	17
Proje No: 4 Deneyap Mini ile Dahili RGB Led Yakma	22
Proje No: 5 Deneyap Mini Potansiyometre ile Led Yakma	26
Proje No: 6 Deneyap Mini ile Karaşimşek Uygulaması	29
Proje No: 7 Deneyap Mini LDR ile Işık Seviyesi Ayarlama	32
Proje No: 8 Deneyap Mini ile Park Sensörü Uygulaması	35
Proje No: 9 Deneyap Mini ile Gaz Kaçağı Alarmı	39
Proje No: 10 Deneyap Mini ile Toprağın Nemini Ölçme	43
Proje No: 11 Deneyap Mini Hareket Sensörü ile Servo Motor Kontrolü	46
Proje No: 12 Deneyap Mini ile Yağmur Alarmı Uygulaması	50
Proje No: 13 IOT Projesi	53

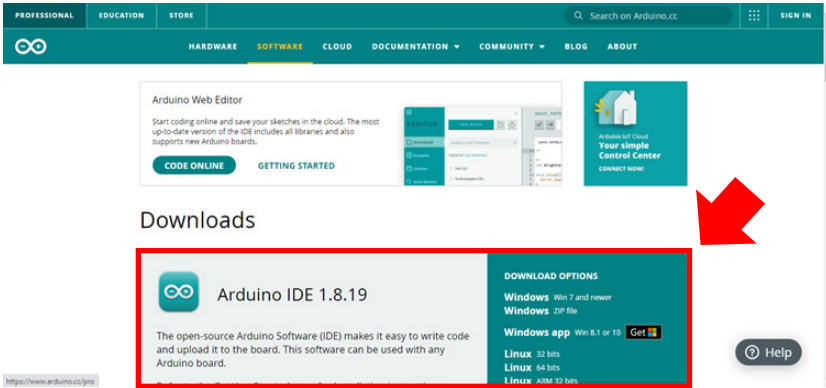


ARDUINO IDE UYGULAMASININ KURULUMU

Bu kısımda Arduino IDE (derleyicisinin) kurulumunu açıklayacağız. Arduino IDE'yi Arduino veya desteklenen diğer mikrokontrolcü kartlarını programlamak için kullanıyoruz. Arduino IDE'nin arayüzü oldukça kolay olduğundan programlamaya ilk adım için mükemmel bir platform sunmaktadır.

Geliştireceğimiz projelerin kodlarını bu arayüz yardımıyla derleyerek mevcut kartımıza atıyoruz. Derleyici bizim o dilde yapmış olduğumuz yazım ve imla hatalarını fark etmemizi sağlar ve bu doğrultuda bize dönüt verir. Fakat derleyiciler algoritma hatalarını veya başka bir deyişle mantık hatalarını fark edemezler. Şimdi kurulumla geçelim.

- İlk olarak Arduino derleyicisini indirmek için resmi Arduino sitesine giriyoruz. Menüdeki **Software** bloğunu seçtikten sonra **Downloads** yazan yerden bilgisayarımızla uyumlu olan kurulum dosyasını indiriyoruz.

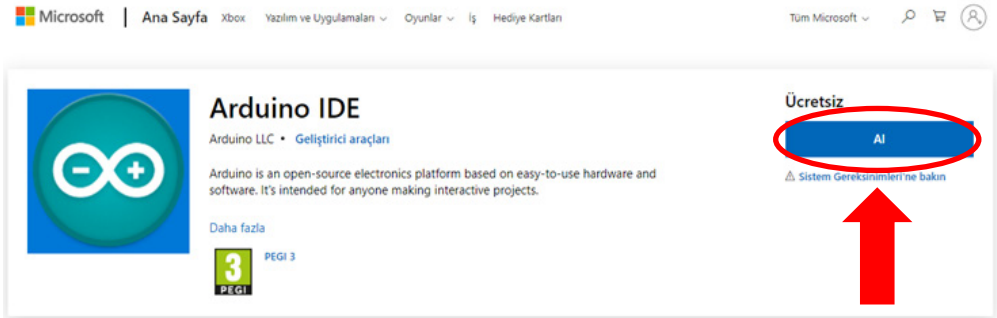


The screenshot shows the Arduino website's 'Downloads' section. The page has a teal header with navigation links: PROFESSIONAL, EDUCATION, STORE, HARDWARE, SOFTWARE, CLOUD, DOCUMENTATION, COMMUNITY, BLOG, ABOUT. A search bar and 'SIGN IN' button are also visible. Below the header, there are three main sections: 'Arduino Web Editor', 'Downloads', and 'Your simple Control Center'. The 'Downloads' section is highlighted with a red box and a red arrow. It features the Arduino logo and the text 'Arduino IDE 1.8.19'. Below this, it states: 'The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.' To the right, under 'DOWNLOAD OPTIONS', there are three options: 'Windows Win 7 and newer' (797 file), 'Windows app Win 8.1 or 10' (with a 'Get' button), 'Linux 32 bits' (64 bits), and 'Linux ARM 32 bits'.

- Bilgisayarımıza uygun olan kurulumu seçtikten sonra karşımıza bağlı ekranı çıkıyor. Bu aşamada bağlı yapmak isterseniz **Contribute Download** ile, istemezseniz **Just Download** seçeneği ile devam edebilirsiniz.

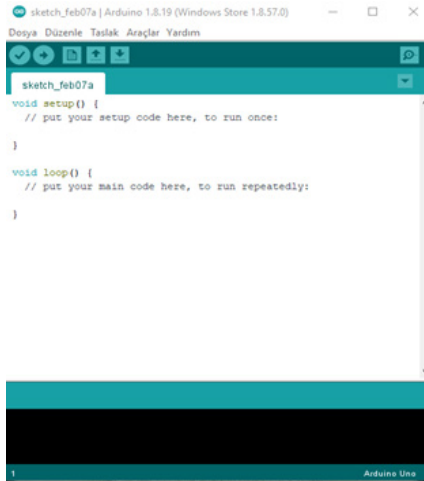


- Devam ettiğimizde **Microsoft Store**'a yönlendiriliyor ve uygulamayı indiriyoruz.

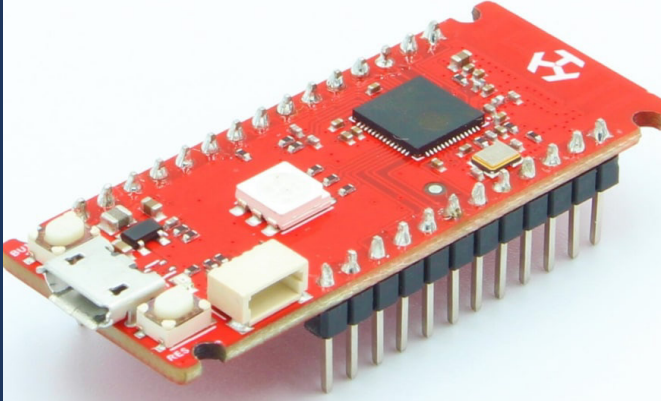




- Yüklemeyi başlatmak için **AI** butonuna tıklıyoruz. Yükleme bittikten sonra buton **Aç** olarak değişecektir. Artık, Arduino derleyicisini bilgisayarımıza kurmuş bulunuyoruz. Arama sekmesine Arduino yazarak derleyiciyi çalıştırabiliriz.



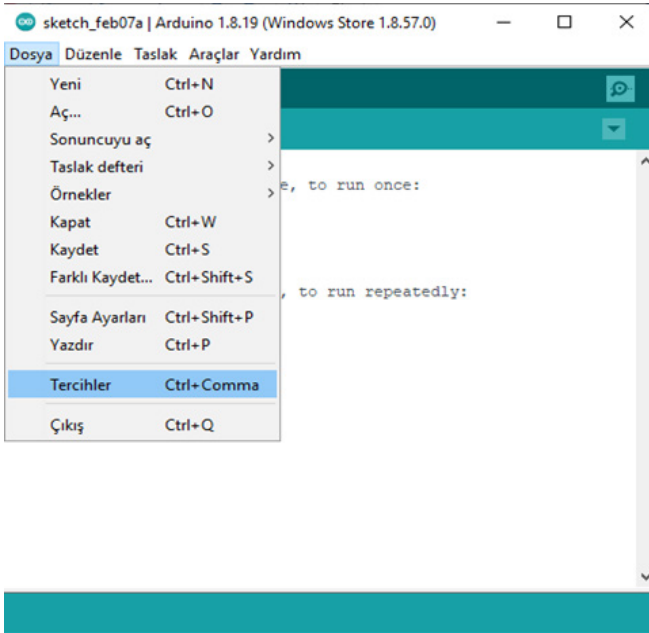
- Açılan bu sayfa **Arduino** derleyici açılış sayfasıdır. Bu sayfada uygulamaların yazılım geliştirmelerini ve konfigürasyon ayarlarını yapabilirsiniz.

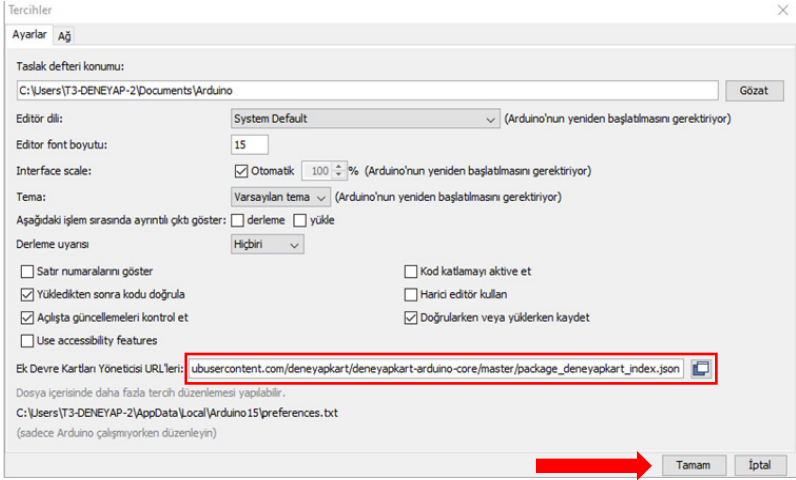


DENEYAP MiNi'NiN ARDUINO IDE'YE EKLENMESİ

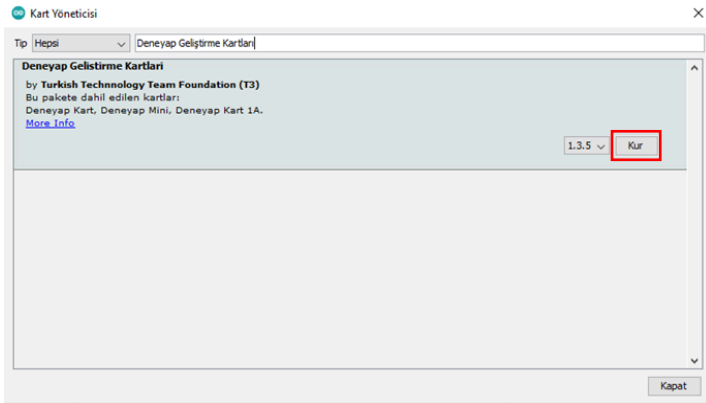
Kodlamaya başlamadan önce Deneyap Mini'yi Arduino IDE'ye tarif edildiği şekilde eklememiz gerekmektedir.

- Arduino IDE'de **Dosya (File)** → **Tercihler (Preferences)** adımlarını takip ederek açılan pencerede **Ayarlar (Settings)** sekmesinde bulunan **Ek Devre Kartları Yöneticisi URL'leri (Additional Boards Manager URLs)** kısmına, aşağıda paylaşılan JSON dosyasına ait adresi kopyalayınız.
- mth.tc/j3z

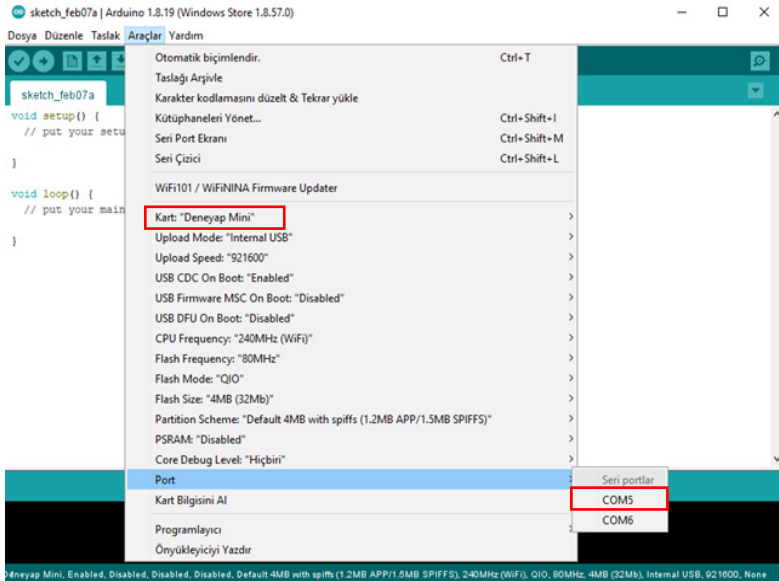
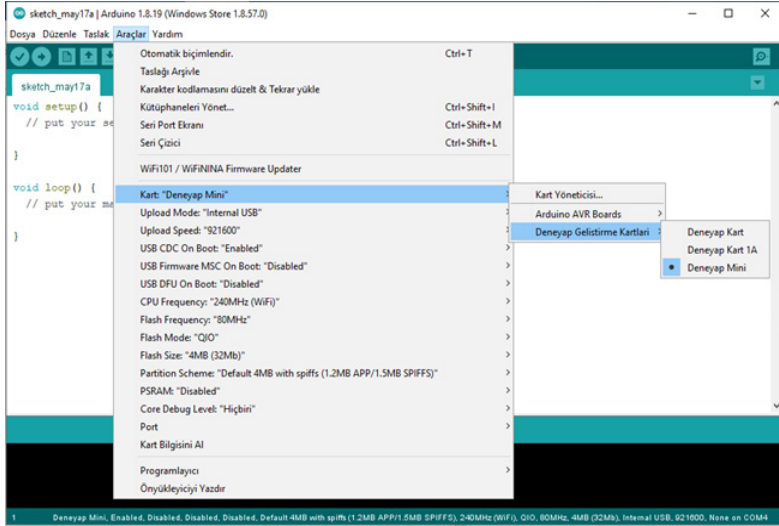




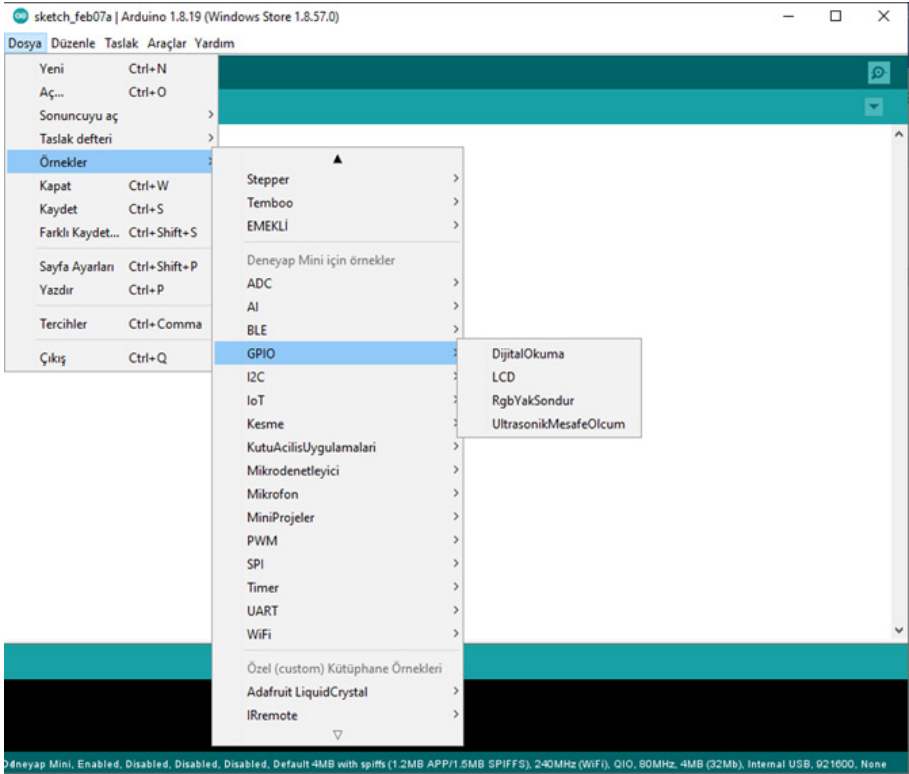
- Ardından, **Araçlar (Tools) → Kart (Board) → Kart Yöneticisi (Boards Manager)** adımlarını takip ederek açılan ekrandaki arama kısmına **“Deneyap Geliştirme Kartları”** yazınız. Son sürüm, varsayılan olarak seçilir ve **Kur (Install)** butonuna tıklayarak yüklemeyi gerçekleştirebilirsiniz.

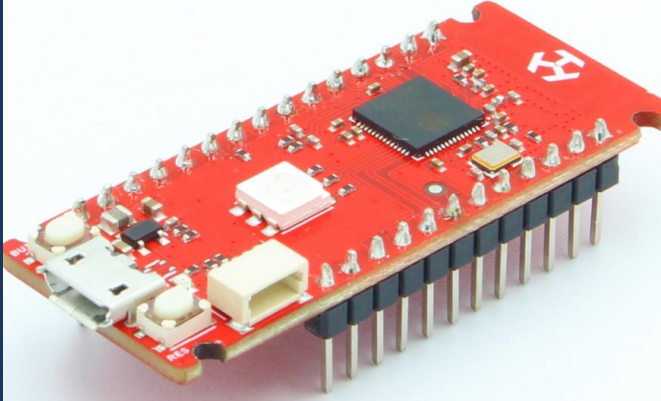


- En son aşama olarak **Araçlar (Tools) → Kart (Board)** adımı ile **“Deneyap Mini”** ve kartınızın bağlı olduğu **Port**'u seçip kodlama adımına geçebilirsiniz.



- Örnek kodlara erişmek için **Dosya (Files)** → **Örnekler (Examples)** başlığı altında bulunan **“Deneyap Mini için Örnekler (Examples for Deneyap Mini)”** yazan kısmı takip edebilirsiniz.





PROJE NO: 1

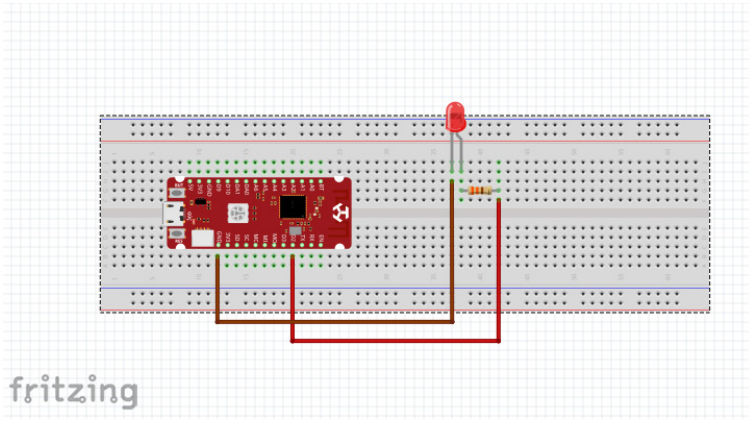
DENEYAP Mini İLE LED YAKMA BLINK UYGULAMASI



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet LED
- 1 adet 330 Ω Direnç
- 2 adet Erkek-Erkek Jumper Kablo

Devre Şeması:



Proje Kodları:

```
sketch_feb02a | Arduino 1.8.15
Dosya Düzenle Taslak Araçlar Yardım
sketch_feb02a $
#define led D2
void setup()
{
  pinMode(led,OUTPUT);
}
void loop()
{
  digitalWrite(led,HIGH);
  delay(1000);
  digitalWrite(led,LOW);
  delay(1000);
}
```

Daha önce indirmiş olduğumuz Arduino derleyicimizi açıyoruz ve sol üst köşede bulunan **Dosya** bölümündeki **Yeni** sekmesine tıklıyoruz. Karşımıza yeni bir Arduino sayfası açılacaktır. Açmış olduğumuz bu yeni sayfada void setup() ve void loop() gibi kod blokları hazır bir şekilde gelmektedir. Bu fonksiyonlar içerisine kodlarımızı yazmaya başlayabiliriz.

Void setup() fonksiyonuna girmeden önce başta **“#define led D2”** komutu ile LED adlı değişkenimize Deneyap Mini’deki dijital D2 pinini atıyoruz.

Void setup() fonksiyonu içerisine **“pinMode(led,OUTPUT)”** komutu ile LED’imizin bağlı olduğu pinin dijital çıkış (OUTPUT) olarak tanımlıyoruz.

“BİLGİ?”

Void setup() Fonksiyonuna programımız için gerekli ayarlamaları, konfigürasyonları ve Mini’yi çalıştırdığımızda sadece bir kez çalışmasını istediğimiz bölümleri iki süslü parantez ({}) arasına yazarız ve void setup() tüm bunları kapsamış olur.

“BİLGİ?”

pinMode() komutu parantezleri içine önce pin adı ardından arasına virgül ekleyerek pinlerimizin giriş ya da çıkış (**INPUT-OUTPUT**) olduğunu belirtir.
pinMode(led, OUTPUT);

Void loop() fonksiyonu içerisinde “**digitalWrite(led,HIGH);**” kodu ile LED’imizin yanmasını istiyoruz. “**delay(500);**” ile ise programın 500 milisaniye hiçbir işlem yapmadan beklemesini söylüyoruz. “**digitalWrite(led,LOW);**” komutu ile daha önceden lojik HIGH seviyesinde olan LED pinimizi lojik LOW seviyesine çekmiş oluyoruz. Tekrar “**delay(500);**” ile hiçbir işlem yapmadan 500 milisaniye beklemesini belirtiyoruz.

“BİLGİ?”

Void loop() fonksiyonu iki süslü parantez ({}) arasına yazmış olduğumuz kod bloklarını Deneyap Mini de enerji olduğu sürece sonsuz bir döngü şeklinde çalıştırmaktadır.



”

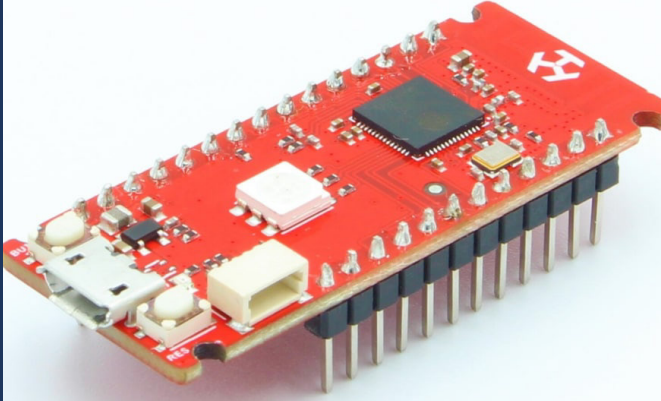
“BİLGİ?”

digitalWrite() adından da anlayacağınız üzere dijital yazma işlemi yapmaktadır. Parantezleri içerisine önce yazmak istediğimiz pin adı ardından virgül ile ayırarak **HIGH-LOW** yazılır. **digitalWrite(led, HIGH);** **delay()** komutu ile programam beklemesi gerektiği belirtilir. Parantezleri içerisine yazılan sayı milisaniye cinsinden hesaplanır.



”

Port numarasını ve kartımızı seçip kodlarımızı kartımıza yükleyelim. Özet olarak bu örneğimizde void loop() fonksiyonunda tanımlamış olduğumuz LED’in 500 milisaniye yanmasını ve 500 milisaniye sönmesini programlamış olduk. Bu işlem, kartımızdaki enerji kesilene kadar devam edecektir.



PROJE NO: 2

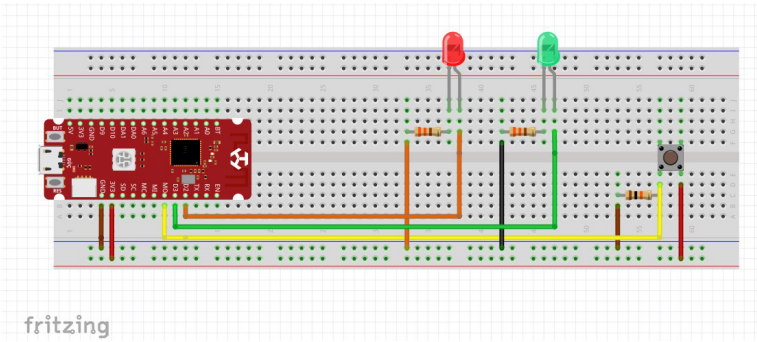
DENEYAP Mini BUTON İLE LED YAKMA



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet Kırmızı LED
- 1 adet Yeşil LED
- 1 adet Push Buton
- 2 adet 330 Ω Direnç
- 1 adet 10k Ω Direnç
- 9 adet Erkek-Erkek Jumper kablo

Devre Şeması:



fritzing

Proje Kodları:

```
sketch_0602a | Arduino 1.8.19 | Windows Store 1.8.37.0
Devre Düzene Tasın Araçlar Yardım
sketch_0602a
#define kırmızı_led D2
#define yeşil_led D3
#define buton D4
int buton_durum = 0;
void setup()
{
  pinMode(kırmızı_led, OUTPUT);
  pinMode(yeşil_led, OUTPUT);
  pinMode(buton, INPUT);
}
void loop()
{
  buton_durum = digitalRead(buton);
  if (buton_durum == 1)
  {
    digitalWrite(yeşil_led, HIGH);
    digitalWrite(kırmızı_led, LOW);
  }
  else
  {
    digitalWrite(yeşil_led, LOW);
    digitalWrite(kırmızı_led, HIGH);
  }
}
```

Bu örneğimizde butona basıldığında Yeşil LED'in, basılmadığında ise Kırmızı LED'in yanmasını sağlayacağız.

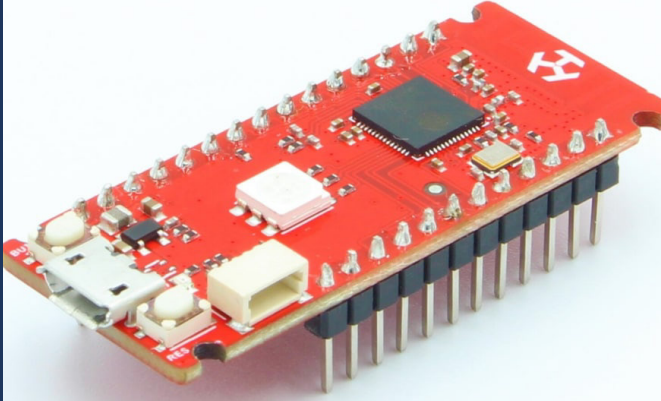
Void setup() fonksiyonundan önce #define ile LED ve butonlarımızın hangi pin'e bağlı olduğunu belirtiyoruz. **"int=buton_durum=0;"** komutu ile İnt türünde buton_durum adında başlangıç değeri 0 olan değişkeni tanımlıyoruz.

void setup() fonksiyonu içerisine kirmizi_led ve yesil_led adlı pinlerimizi çıkış (OUTPUT) ve buton adlı pinimizi giriş (INPUT) olarak tanımlıyoruz.

void loop() fonksiyonu içerisinde **"buton_durum=digitalRead(buton);"** komutu ile buton adlı pin'in dijital durumunu buton_durum adlı değişkene atıyoruz.

Butondan gelen veriyi "if-else" koşulu içerisinde **"if(buton_durum==1);"** komutu ile kontrol ediyoruz. Eğer buton durumu 1 (HIGH) ise **"digitalWrite(yesil_led, HIGH);"** komutu ile Yeşil LED'i yakmasını, eğer değilse else bloğu içerisinde **"digitalWrite(kirmizi_led, LOW);"** komutu ile Kırmızı LED'i söndürmesini söylüyoruz.

```
if (koşul) {  
    //koşul doğruysa yapılacaklar  
}  
else {  
    //koşul yanlışsa yapılacaklar  
}
```



PROJE NO: 3

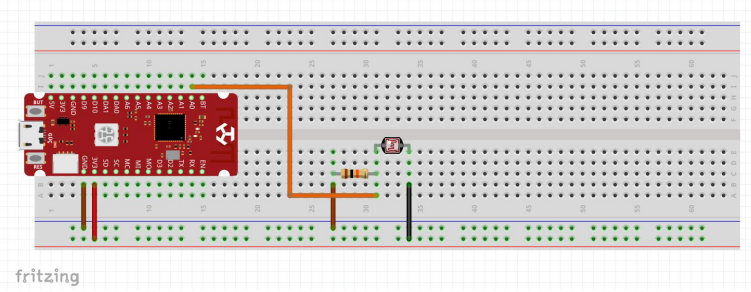
DENEYAP Mini İLE ANALOG OKUMA SERİ HABERLEŞME



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet Kırmızı LED
- 1 adet Yeşil LED
- 1 adet Push Buton
- 2 adet 330 Ω Direnç
- 1 adet 10k Ω Direnç
- 9 adet Erkek-Erkek Jumper kablo

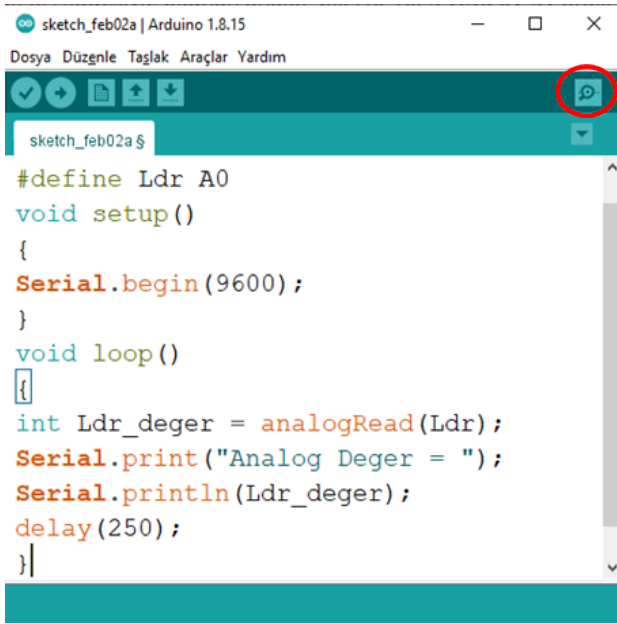
Devre Şeması:



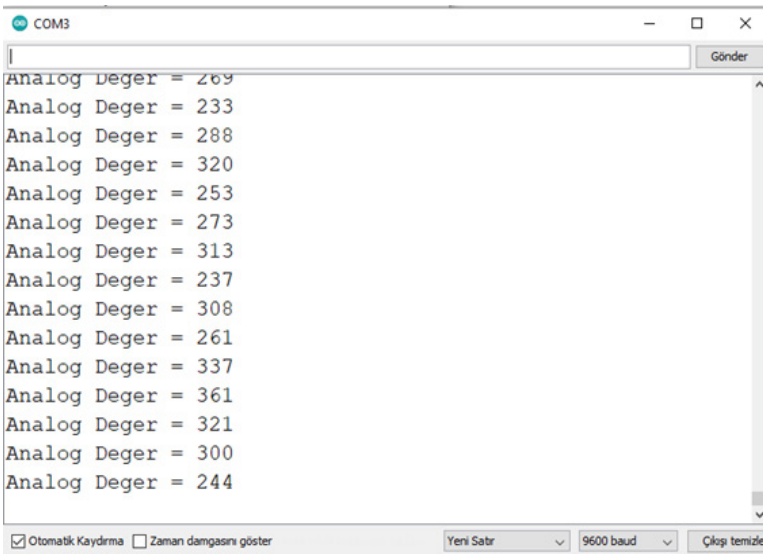
LDR (Light dependent Resistor): Kelime anlamı olarak ışığa duyarlı direnç demektir. Foto direnç olarak da adlandırılmaktadır. LDR'nin üzerine ışık düştükçe direnç azalır, karanlık ortamlarda ise direnç artar. LDR ışık sensörü voltaj/gerilim bölücü olarak kullanılabilir.



Proje Kodları:



```
sketch_feb02a | Arduino 1.8.15
Dosya Düzenle Tağlak Araçlar Yardım
sketch_feb02a $
#define Ldr A0
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int Ldr_deger = analogRead(Ldr);
  Serial.print("Analog Deger = ");
  Serial.println(Ldr_deger);
  delay(250);
}
```



```
COM3
Analog Deger = 269
Analog Deger = 233
Analog Deger = 288
Analog Deger = 320
Analog Deger = 253
Analog Deger = 273
Analog Deger = 313
Analog Deger = 237
Analog Deger = 308
Analog Deger = 261
Analog Deger = 337
Analog Deger = 361
Analog Deger = 321
Analog Deger = 300
Analog Deger = 244
 Otomatik Kaydırma  Zaman damgasını göster
Yeni Satır 9600 baud Çıkış temizle
```

“**#define Ldr A0**” ile A0 pinini LDR isimli deęiřkene atıyoruz. Daha sonra void setup() fonksiyonları ierisine “**Serial.begin(9600);**” ile seri haberleřmemizi baud 9600 olacak řekilde bařlatıyoruz.

“**BİLGİ?**”



Serial.begin(baud rate) komutu ile seri haberleřme bařlatılır. Paranteleri iine yazılan baud deęeri ile seri haberleřmenin hızı belirtilir. **Serial.begin(9600);**

”

Burada analog okuma yapacaęımız iin pin durumunu ıkıř (OUTPUT) veya giriř (INPUT) olarak tanımlamamız gerekmiyor.

void loop() fonksiyonunda “**int=Ldr_Deger= analogRead(Ldr);**” ifadeyle okunan analog bilgiyi (0-8191 veya 0V-3,3V) int trnde olarak tanımladıęımız Ldr_Deger deęiřkenine atıyoruz.

“**BİLGİ?**”



analogRead() komutu analog pinden veri okumamıza imkan saęlar. Parantezleri ierisine analog pin numarası ya da pine en bařta tanımladıęımız isim yazılır.

analogRead(Ldr);

Deneyap Mini'nin okuduęu analog deęerler 0-8191 veya 0V-3.3V'dir

”

“**Serial.print(“Analog Deger= ”);**” ifadesiyle serial ekran üzerinde “Analog Deger=” yazmasını istiyoruz.

“**Serial.println(Ldr_deger);**” ifadesiyle okumuş olduğumuz analog değeri serial ekrana yazdırıp imleci alt satıra geçiriyoruz ve 250 milisaniye beklemesini istiyoruz. Kodlarımızı, kartımızı ve portumuzu seçerek Deneypap Mini’ye yüklüyoruz. Gelen verilere bakmak için Serial ekranı açmamız gerekiyor. Bunun için Arduino derleyicisinde sağ üstte bulunan büyüteç simgesine tıkladığımızda seri haberleşme ekranı karşımıza çıkıyor.

“BİLGİ?”



Serial.print() komutu ile parantezleri içerisine ekranda yazmasını istediğimiz söz dizimi tırnak içinde yazılır ve imleç yanda kalarak yazmaya devam eder.

Serial.print(“Analog Değer=”);

Serial.println() komutu ile parantezleri içerisine ekranda yazılmasını istediğimi söz dizimi tırnak içinde yazılır. imleç alt satıra geçerek yazmaya devam edilir.

Sayısal değer için tırnak işareti kullanılmaz.

Serial.println(Ldr_deger);

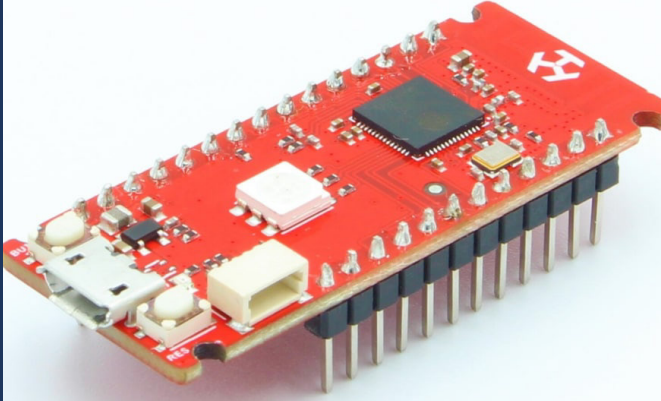
”

“DİKKAT!”



Deneyap Mini'de **RX-TX** pinleri seri haberleşmeyi sağlayan pinlerdir. ayrıca bu pinler kart ile bilgisayarın haberleşmesini sağlayan USB hattında bağlı. Eğer kod yüklenme esnasında bu pinlere herhangi bir bağlantı yapılmış olursa kod yükleme işlemi gerçekleşmeyecektir. Bunun sebebi ise kartımızın bilgisayara USB hattı ile haberleşmeyi sağlayamamasıdır.

”



PROJE NO: 4

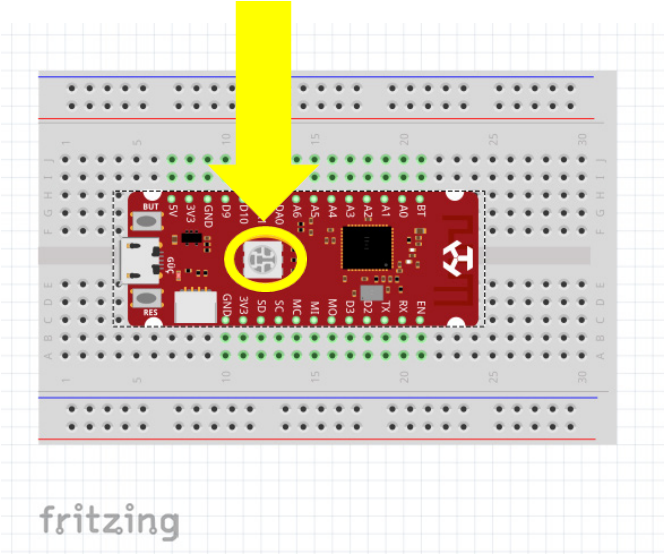
DENEYAP MiNi İLE DAHİLİ RGB LED KULLANIMI



Gerekli Malzemeler:

- Deneyap Mini

Devre Şeması:



“**DIKKAT!**”

Bu Uygulamada Deneyap Mini üzerinde yer alan dahili RGB led kullanılmıştır.



”

Proje Kodları:



```
sketch_feb02a | Arduino 1.8.19 (Windo... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_feb02a
#define beklemeSuresi 500
void setup() {
  pinMode(LEDR, OUTPUT);
  pinMode(LEDG, OUTPUT);
  pinMode(LEDB, OUTPUT);
}
void loop() {

  digitalWrite(LEDR, LOW);
  digitalWrite(LEDG, HIGH);
  digitalWrite(LEDB, HIGH);
  delay(beklemeSuresi);

  digitalWrite(LEDR, HIGH);
  digitalWrite(LEDG, LOW);
  digitalWrite(LEDB, HIGH);
  delay(beklemeSuresi);

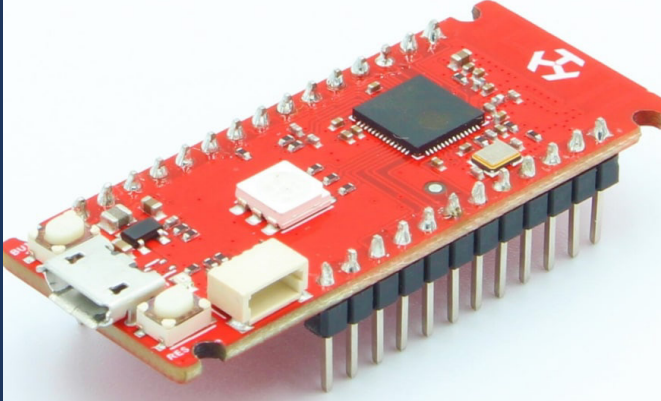
  digitalWrite(LEDR, HIGH);
  digitalWrite(LEDG, HIGH);
  digitalWrite(LEDB, LOW);
  delay(beklemeSuresi);

  digitalWrite(LEDR, LOW);
  digitalWrite(LEDG, LOW);
  digitalWrite(LEDB, HIGH);
  delay(beklemeSuresi);
}
Yükleme tamamlandı.
```

Deneyap Mini kartımız üzerinde dahili olarak bulunan LED'i belirli aralıklarla yakıp söndüreceğiz. RGB LED diğer LED'lerden farklı olarak 3 ana rengi içerisinde barındırır. Bu renkler kırmızı, yeşil ve mavidir. RGB LED ile farklı renk tonlamaları elde edebiliriz.

void setup() fonksiyonu içerisinde pin tanımlamalarını gerçekleştirelim. Bunun için pinMode() komutunu kullanıyoruz. Kırmızı için **LEDR**, yeşil için **LEDG**, mavi için ise **LEDB** isimli pinlerin çıkış olduğunu **OUTPUT** ile belirtiyoruz.

void loop() fonksiyonu içerisinde ise sırayla tüm renklerin 500 milisaniye boyunca yanmasını isteyeceğiz. Yakmak istediğimiz renk için olan pin'e lojik 1, diğerlerine ise lojik 0 değerini digitalWrite() komutu ile yazıp 500 milisaniye beklemesini delay() komutu ile belirtiyoruz. RGB LED'den sarı renk elde etmek için sadece kırmızı ve yeşil LED'i aktif edebilirsiniz.



PROJE NO: 5

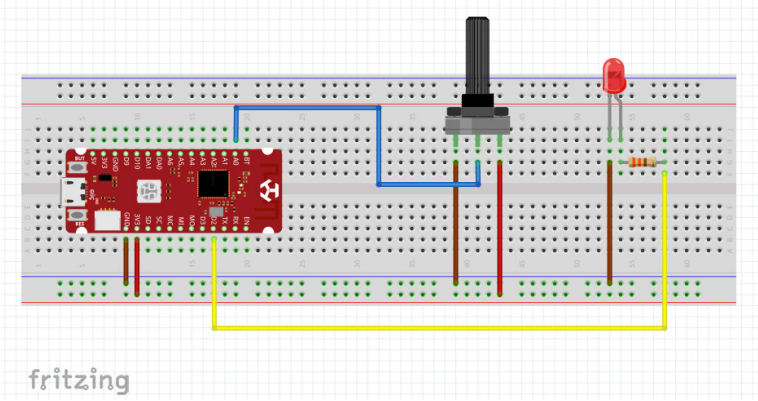
DENEYAP MİNİ POTANSİYOMETRE İLE LED IŞIĞINI AYARLAMA



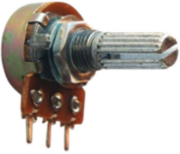
Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 10K Ω Potansiyometre
- 1 adet LED
- 1 adet 330 Ω Direnç
- 7 adet Erkek-Erkek Jumper Kablo

Devre Şeması:



“BİLGİ?”



Potansiyometre: Dışarıdan gelen fiziksel müdahaleler ile değeri değiştirebilen 3 bacaklı dirençlerdir. Potansiyometrenin yapı itibarıyla bir sirenç olduğunu düşünürsek elektronik devrelerde akım ve gerilim kontrolü potansiyometre ile sağlanabilir.



”

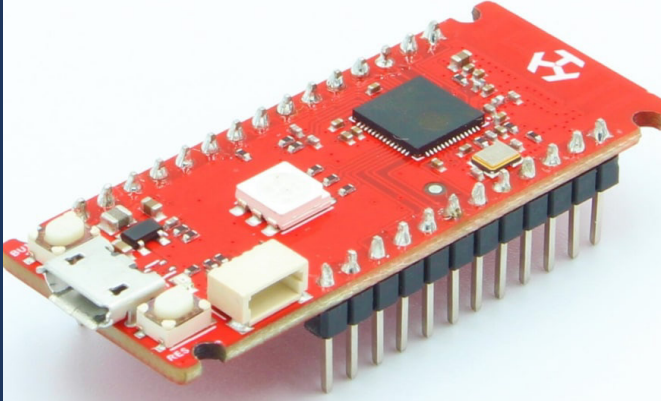
Proje Kodları:

```
sketch_feb02a | Arduino 1.8.19 (Windows Store 1.8.5... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_feb02a
#include <analogWrite.h>
#define led D2
#define pot A0
void setup() {
  pinMode(led,OUTPUT);
  pinMode(pot,INPUT);
}
void loop() {
  int deger = analogRead(pot);
  deger = map(deger, 0,8191,0,255);
  analogWrite(led,deger);
}
Derleme tamamlandı.
```

Bu örneğimizde, potansiyometreden okunan değere göre LED'imizin ışık şiddetini ayarlayacağız. Bunun için PWM pinlerini kullanacağız. PWM (Pulse with Modulation), sinyal genişlik modülasyonun kısaltmasıdır. Bu özellik Deneyap Mini üzerinde D0 ve D1 pinlerine bağlıdır. Tüm dijital giriş-çıkış pinleri (D0-D13) PWM pini olarak kullanılabilir.

İlk olarak LED ve pot adında değişkenler oluşturup bağlı oldukları pinleri belirtiyoruz. Sonrasında void setup() fonksiyonu içerisinde pinMode() komutu ile LED ve potansiyometremizin bağlı olduğu pinlerin giriş ve çıkış olduğunu belirtiyoruz.

void loop() fonksiyonunda, potansiyometreden **“analogRead(pot);”** komutu ile veriyi okuyup değer adındaki değişkenimize atıyoruz. Aldığımız bu veriyi Deneyap Kartımızdaki çıkışa göre yazdırmamız gerekiyor. Bunun için ise **“deger=map(deger,0,8191,0,255);”** komutunu kullanarak 0 ile 8192 arasında gelen değeri 0 ile 255 arasına oranlayıp sonucu değer değişkenimize atıyoruz. Analog okumayı 13 bit (2¹³=8192) çözünürlükte yaparken, analog yazmayı 8 bit (2⁸=256) çözünürlükte yapabiliyoruz. Map() komutu yerine derseniz direkt olarak 32'ye de bölebilirsiniz. Oranlama işleminden sonra **“analogWrite(led,deger);”** komutu ile LED'imize değer değişkenindeki veriyi yazdırıyoruz.



PROJE NO: 6

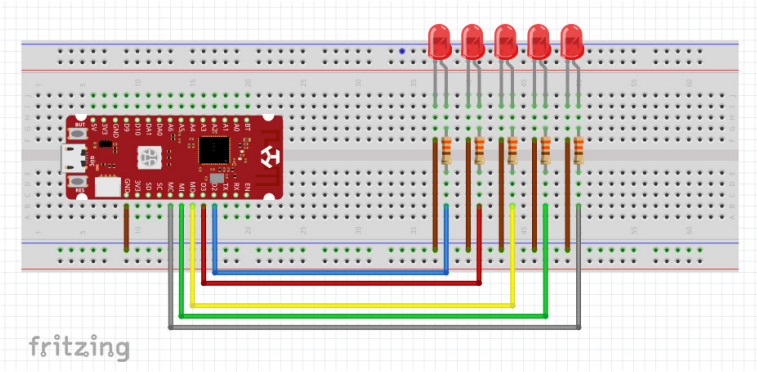
DENEYAP Mini İLE KARAŞİMŞEK UYGULAMASI



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 5 adet LED
- 5 adet 330 Ω Direnç
- 11 adet Erkek- Erkek Jumper Kablo

Devre Şeması:



Proje Kodları:

```
sketch_mar04a | Arduino 1.8.19 (Windo... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_mar04a $
int ledler[] = {D2, D3, D4, D5, D6};
void setup() {
  for(int i=0; i<6; i++){
    pinMode(ledler[i], OUTPUT);
  }
}
void loop() {
  for(int i=0; i<6; i++){
    digitalWrite(ledler[i], HIGH);
    delay(100);
    digitalWrite(ledler[i], LOW);
  }
  for(int x=5; x>-1; x--){
    digitalWrite(ledler[x], HIGH);
    delay(100);
    digitalWrite(ledler[x], LOW);
  }
}
```

Bu örneğimizde, art arda yapılması gereken işlemlerde kullandığımız for döngüsünü kullanacağız. Uygulamamızda 5 adet LED'e tek tek çıkış vermemiz ve onları sırasıyla yakmamız gerekecek.

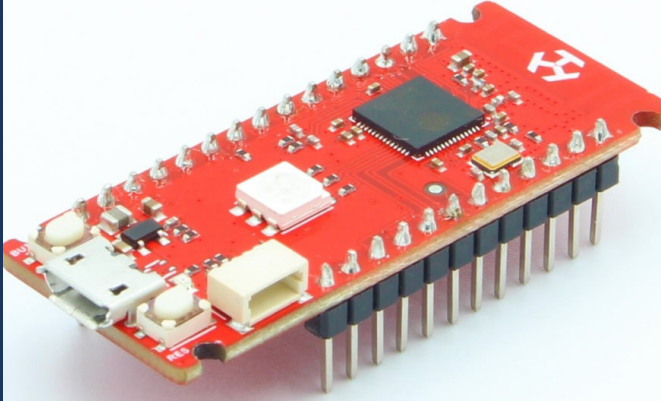
"int ledler[]=(D2,D3,D4,D5,D6)" ile int tam sayı türünde LED'ler adında bir dizi tanımlıyoruz. Dizi içerisine virgüller ile bağlı olan dijital pinleri yazıyoruz.

Void setup() fonksiyonu içerisinde kullandığımız LED'lerin OUTPUT olmasını istiyoruz. Bunun için tek tek yazmak yerine for döngüsü içerisinde **"pinMode(ledler[i],OUTPUT)"** LED'lerin çıkış olduğunu belirtiyoruz.

```
for (int i = 0; i < 10; i++)  
{  
    Komut Bloğu  
}
```

Örneğin, i değişkeninin değeri 3 ise pinMode(ledler[3],OUTPUT) dediğimizde yazılım bunu pinMode(3, OUTPUT) şeklinde algılıyor.

Void loop() fonksiyonu içerisinde de aynı mantık ile for döngüsünü kullanıyoruz. **"digitalWrite(ledler[i], HIGH);"** ile LED'lerin sıra ile yanmasını, **"delay(100);"** ile 100 milisaniye beklemesini ve **"digitalWrite(ledler[i], LOW);"** ile sönmelerini sağlıyoruz.



PROJE NO: 7

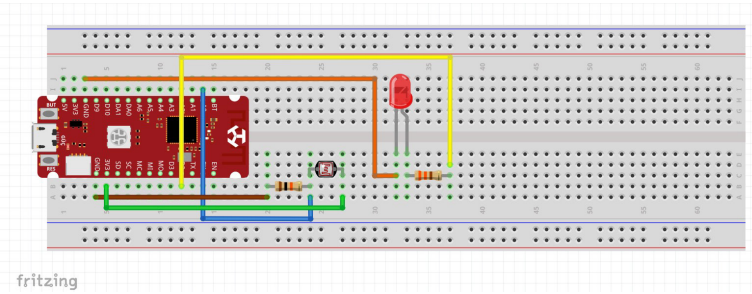
DENEYAP Mini LDR İLE IŞIK SEVİYESİ AYARLAMA



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet LED
- 1 adet LDR
- 5 adet Erkek-Erkek Jumper Kablo
- 1 adet 330 Ω Direnç
- 1 adet 10K Ω Direnç

Devre Şeması:



Proje Kodları:

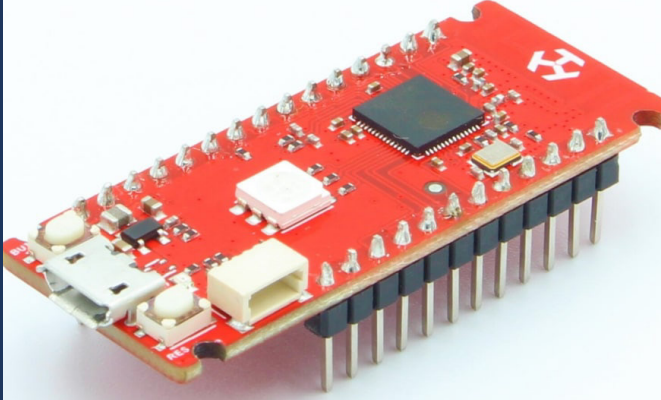
```
sketch_mar23a | Arduino 1.8.19 (Window... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_mar23a $
#define led D2
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  int deger=analogRead(A0);
  if (deger>1000){
    digitalWrite(led, LOW);
  }
  else{
    digitalWrite(led, HIGH);
  }
}
```

Bu uygulamamızda, ortamdaki ışığı algılayabilen LDR'den veri okuyup bu veriye göre LED'imizi yakıp söndüreceğiz. LDR, ortamdaki ışık miktarına göre direnci değiştirir. Bu değişimi Deneyap Mini Kart ile algılayabilir ve bu sayede ortam karanlık olduğunda LED'i yakıp ortam aydınlık olduğunda LED'i söndürebiliriz.

"#define led D3" ile LED adlı değişkenimize Deneyap Mini'deki dijital D3 pinini atıyoruz. Void setup() fonksiyonu içerisinde LED'imizin çıkış pini olduğunu **"pinMode(led, OUTPUT);"** komutları ile tanımlıyoruz.

Ardından void loop() fonksiyonları içerisinde **"int deger=analogRead(A0);"** komutları ile A0 pinine bağlı olan LDR direncimizin değerini analogRead komutu ile okuyup int türündeki değişkenimize aktarıyoruz.

"if(deger>900);" komutu ile LDR'den aldığımız değer 900'den büyük olup olmadığını kontrol ediyoruz. Değer büyük ise **"digitalWrite(led,-LOW);"** komutu ile LED'in sönmelerini, değilse else parantezleri içinde **"digitalWrite(led,HIGH);"** komutu ile LED'in yanmasını sağlıyoruz.



PROJE NO: 8

DENEYAP MiNi İLE PARK SENSÖRÜ UYGULAMASI



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 8 adet Dişı-Erkek Jumper Kablo
- 6 adet Erkek-Erkek Jumper Kablo
- Buzzer
- 330 Ω Direnç
- HC-SR04 Ultrasonik Sensör
- 1 adet Seviye Dönüştürücü

Devre Şeması:

“**DIKKAT!**”

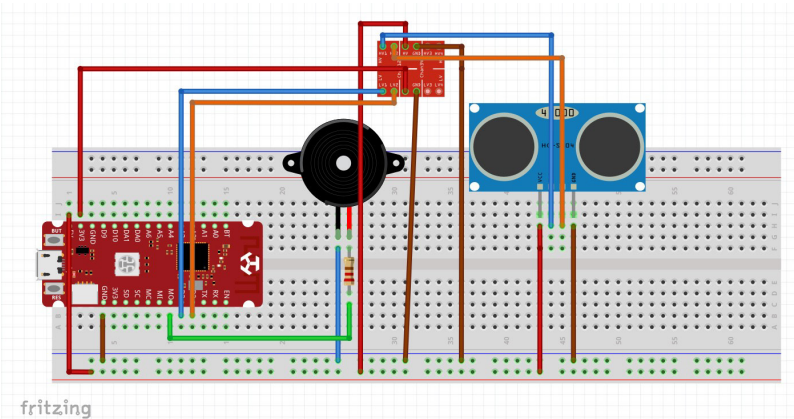


Deneyap Mini kartımız 3.3 ile çalışmaktadır. Bazı sensörlerin çalışma voltajı 5V olduğundan dolayı aradaki voltaj farkını dönüştürmek için seviye dönüştürücü kullanılır.

Bu uygulamadan sonra tüm projelerimizde Seviye Dönüştürücü kullanılmıştır.

Kullandığımız sensörler bizim kullandıklarımızdan farklı olabilir.

”



fritzing

“BİLGİ?”



Ultrasonik Mesafe Sensörü: Mesafe sensörünün iki gözü bulunmaktadır. Mesafe sensörü bir gözden insan kulağının duyamayacağı ses dalgaları gönderir ve diğer gözden karşıda bulunan nesneden, eğer böyle bir nesne varsa, yansıyan ses dalgalarını alır. Ses dalgalarını gönderme ve yansıyan dalgaları alma arasında geçen süre hesaplanarak karşıdaki cismin mesafesi bulunur.



Proje Kodları:

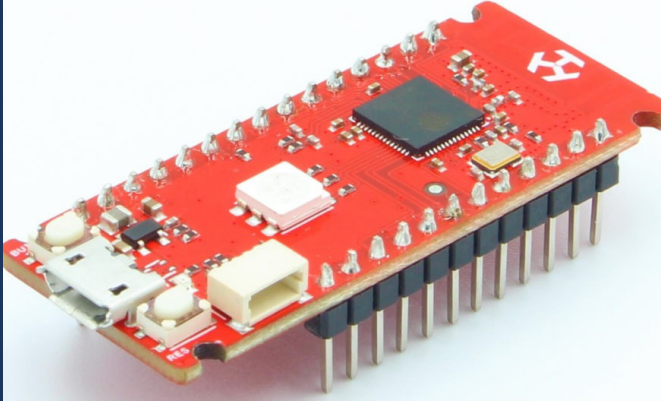
```
sketch_mar24e | Arduino 1.8.19 (Windows Store 1.8.57.0)
Dosya Düzenle Taslak Araçlar Yardım
sketch_mar24e $
#include "hcsr04ESP32.h"
#define trigPin D3
#define echoPin D2
#define buzzer D4
UltrasonicDistanceSensor distanceSensor(trigPin, echoPin);
void setup() {
  pinMode(buzzer, OUTPUT);
}
void loop() {
  float mesafe = distanceSensor.measureDistanceCm();
  if(mesafe<20){
    digitalWrite(buzzer, 1);
    delay(50);
    digitalWrite(buzzer, 0);
  }
  else if(mesafe<50){
    digitalWrite(buzzer,1);
    delay(100);
    digitalWrite(buzzer,0);
  }
  else if (mesafe<80){
    digitalWrite(buzzer,1);
    delay(150);
    digitalWrite(buzzer,0);
  }
  else{
    digitalWrite(buzzer,0);
  }
}
```


Bu projemizde HC-SR04 ultrasonik sensör yardımıyla mesafeyi ölçüp alınan değere göre buzzer ile geri bildirim alacağız.

İlk olarak **"#define"** ile sensörümüze ait olan pinleri ve buzzer'ımızın pin tanımlamalarını gerçekleştiriyoruz. voidsetup() fonksiyonu içerisinde **"pinMode();"** ile tanımladığımız pinlerin giriş-çıkış olduğunu belirtirken, **"Serial.begin(115200);"** ile seri haberleşmeyi başlatıyoruz.

Void loop() fonksiyonu içerisinde sensör ile ölçülen mesafeyi float türünde mesafe adındaki değişkenimize atıyoruz ve seri ekrana **"Serial.println(mesafe);"** komutu ile yazdırıyoruz.

Ardından if koşulu ile mesafeyi kontrol ediyoruz. Buzzer'ımıza **"digitalWrite();"** komutu ile lojik 1 ve lojik 0 değerlerini ve **"delay();"** komutu ile mesafeye göre bekleme süresini belirleyerek uygulamamızı tamamlıyoruz.



PROJE NO: 9

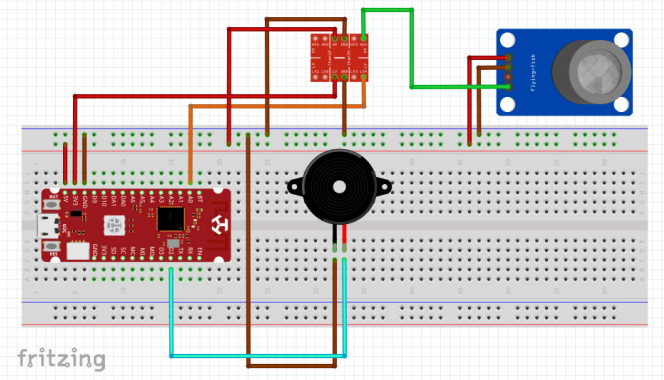
DENEYAP MiNi İLE GAZ KAÇAĞI ALARMI YAPIMI



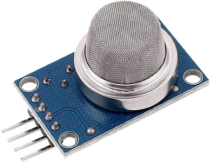
Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet MQ-2 Gaz Sensörü
- 1 adet Seviye Dönüştürücü
- 1 adet Buzzer
- 4 adet Erkek-Erkek Jumper Kablo
- 7 adet Erkek-Dişi Jumper Kablo
- 1 adet Dişi-Dişi Jumper Kablo

Devre Şeması:



“BİLGİ?”



Gaz Sensörü: Ortamdaki yanıcı gaz ve sigara dumanını algılayabilen bir gaz sensörüdür. Ortamdaki gaz yoğunluğuna göre analog çıkış, ya da gaz varlığına göre dijital çıkış verir. Duman algılama aralığı 200 - 10000 ppm'dir. MQ2 gaz sensörü aynı zamanda kimyasal direnç olarak da bilinir. Gazla temas ettiğinde direnci değişen bir algılama malzemesi içerir. Direnç değerindeki bu değişiklik gazın tespiti için kullanılır.



Proje Kodları:

```
sketch_mar23a | Arduino 1.8.19 (Window... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_mar23a $
int esikdegeri=5000;
#define buzzerPin D2
void setup() {
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  deger=analogRead(A0);
  if(deger>esikdegeri){
    digitalWrite(buzzerPin, HIGH);
    delay(100);
  }
  else{
    digitalWrite(buzzerPin, LOW);
  }
  Serial.println(deger);
}
```

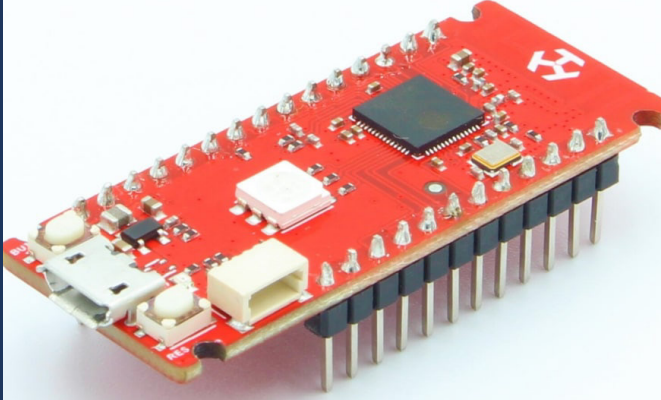
Bu örneğimizde gaz sensörü kullanarak gaz alarmı yapacağız. Ortamdaki hava ölçüldüğünde algılanan gaz eğer belirlenen eşik değerini geçerse alarm çalacak.

İlk olarak, **“int esikdegeri=400;”** tanımı ile int türünde esikdegeri isimli değişken oluşturup buna 400 değerini atıyoruz. **“#define buzzerPin D2”** dediğimizde buzzer’ımızın D2 pinine bağlı olduğunu belirtiyoruz. Ve gaz sensöründen gelen veriyi atamak için **“int deger;”** isimli bir değişken oluşturuyoruz.

Void setup() fonksiyonu içerisinde buzzer’ımızın bağlı olduğu pini **“pinMode(buzzerPin, OUTPUT);”** komutları ile çıkış pini olarak tanımlıyoruz. Ve seri haberleşme ekranından sensörün ölçtüğü değeri görebilmek için **“Serial.begin(9600);”** komutu ile seri haberleşme ekranını başlatıyoruz.

Void loop() fonksiyonu içerisinde sensörün sürekli olarak havanın durumunu ölçmesini ve bunu eşik değeri ile karşılaştırmasını istiyoruz. Bunun için sensör ile ölçülen havayı **“deger= analogRead(A0);”** komutu ile A0 pinine bağlı olan sensördeki değeri okuyarak değer değişkenine atıyoruz. Daha sonra if koşul döngüsü ile değer değişkeninin içindeki veriyi, tanımda belirttiğimiz eşik değeri ile karşılaştırıyoruz. Bunun için **“if(deger>e-sikdegeri)”** komutunu kullanıyoruz. Eğer ölçülen değer eşik değerinden büyükse if parantezleri içinde **“digitalWrite(buzzerPin, HIGH);”** komutu ile buzzer’ın çalmasını ve **“delay(100);”** ile 100 milisaniye beklemesini istiyoruz.

Else komutu içerisinde ise ölçülen değer eşik değerinden küçük olduğu durumda buzzer’ın susmasını **“digitalWrite(buzzerPin, LOW);”** komutu ile sağlıyoruz. Fonksiyonun sonunda da **“Serial.println(deger);”** ile değer değişkenindeki değeri seri port ekranında gösteriyoruz.



PROJE NO: 10

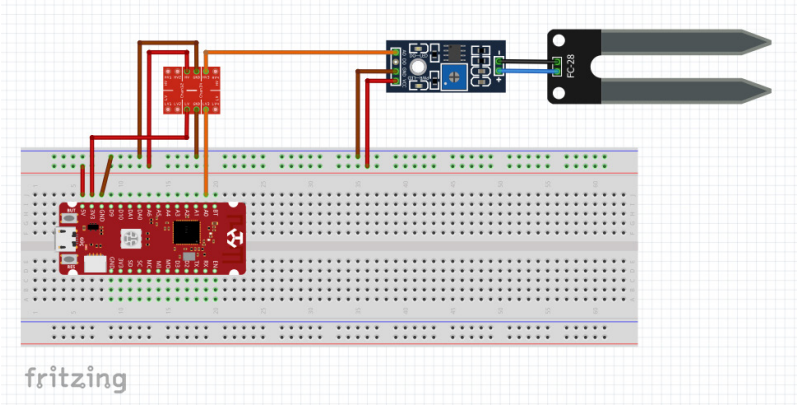
DENEYAP Mini İLE TOPRAĞIN NEMİNİ ÖLÇME



Gerekli Malzemeler:

- Deneyap Mini
- 7 adet Erkek-Dişi Jumper Kablo
- 3 adet Dişi-Dişi Jumper Kablo
- 2 adet Erkek-Erkek Jumper Kablo
- Breadboard
- 1 adet Toprak Nem Sensörü
- 1 adet Seviye Dönüştürücü

Devre Şeması:



“BİLGİ?”

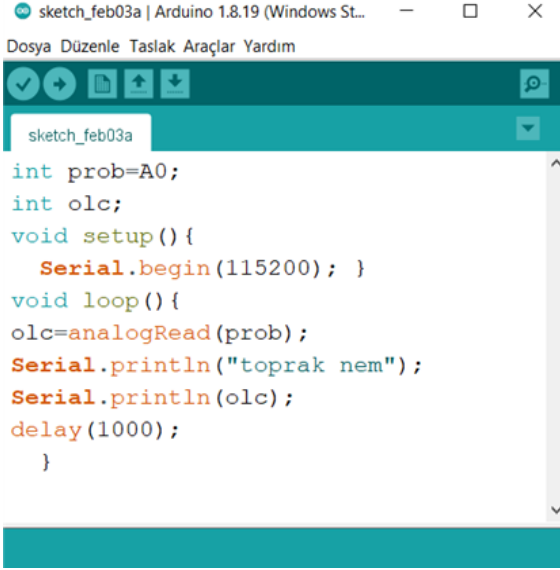


Toprak Nem Sensörü: Toprak nem sensörü toprağın içindeki nem miktarını ölçmek için kullanılan sensördür. Nem ölçer problar ölçüm yapılacak toprağa batırılır toprağın veya batırılan sıvının meydana getirdiği dirençten dolayı, prob uçları arasında bir gerilim farkı oluşur. Bu gerilim farkının büyüklüğüne göre de nem miktarı ölçülür.



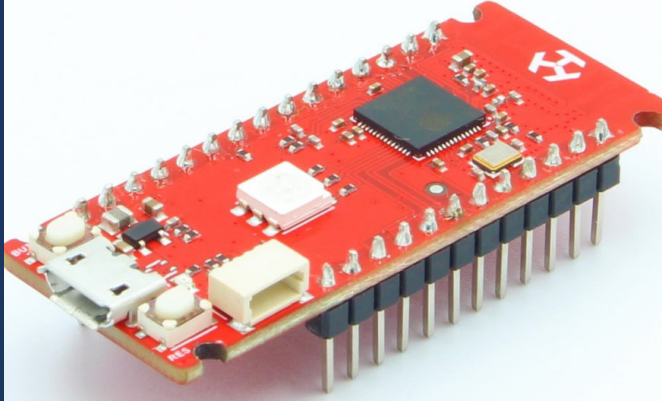
”

Proje Kodları:



```
sketch_feb03a | Arduino 1.8.19 (Windows St... - □ ×
Dosya Düzenle Taslak Araçlar Yardım
sketch_feb03a
int prob=A0;
int olc;
void setup() {
    Serial.begin(115200); }
void loop() {
    olc=analogRead(prob);
    Serial.println("toprak nem");
    Serial.println(olc);
    delay(1000);
}
```

Bu örneğimizde, toprak nem sensörü kullanarak toprağın nem değerini okumayı göreceğiz. İlk olarak void setup(); fonksiyonu öncesinde sensörün bağlı olduğu pini ve olc adındaki değişkenimizi tanımlıyoruz. Ardından void setup(); fonksiyonu içerisinde **“Serial.begin(115200);”** ile serial haberleşmeyi başlatıyoruz. Void loop(); fonksiyonu içerisinde sensörden gelen veriyi değişkenimize aktarmak için **“olc=analogRead(prob);”** komutunu kullanırken, aldığımız veriyi ekranda yazdırmak için Serialprintln(); komutunu kullanıyoruz. Ekranı yazdırdıktan sonra 1000 milisaniye beklemesini söylüyoruz.



PROJE NO: 11

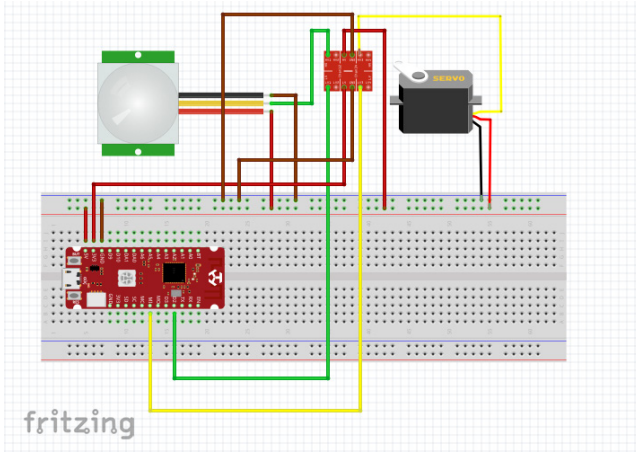
DENEYAP Mini HAREKET SENSÖRÜ İLE SERVO KONTROLÜ



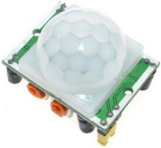
Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet Hareket Sensörü
- 1 adet Servo Motor
- 4 adet Erkek-Erkek Jumper Kablo
- 9 adet Dişi-Erkek Jumper Kablo
- 1 adet Dişi-Dişi Jumper Kablo
- 1 adet Seviye Dönüştürücü

Devre Şeması:



“BİLGİ”



PIR Hareket Sensörü: Sensör kızılötesi dalgalarla çalışmaktadır. PIR sensörü termal enerjiye duyarlı kızılötesi sinyallerle ortamdaki ısı farklılığını algılar ve canlı bir cisim hareket ettiğinde ısı farkından dolayı hareketi belirler. Üzerindeki lens sayesinde ortamdaki nesnelere gelen ışınlar sensörün odaklanmasını sağlar. Ortamda bir dalga değişimi olduğunda sensör algılama işlemi gerçekleştirir. Algıladığı bu hareketi dijital sinyal olarak çıkış verir. Kullanacağınız uygulamalara göre üzerinde bulunan potansiyometreleri bir tornavida yardımıyla ayarlayarak ölçüm mesafeni ve hassasiyetini ayarlayabilirsiniz.



“BİLGİ?”



Servo Motor: Mekanizmalar içerisinde hareket kontrolünü servo motor aracılığıyla gerçekleştirebilir. Genellikle 0 ile 180 arasında (bazıları 360 derece) dönüş sağlayabilen motordur. 180 derece dönüşün karşılığı yarım turdur. Yarım tur içerisinde istenilen derece servo motor durdurulabilir.



”

Proje Kodları:

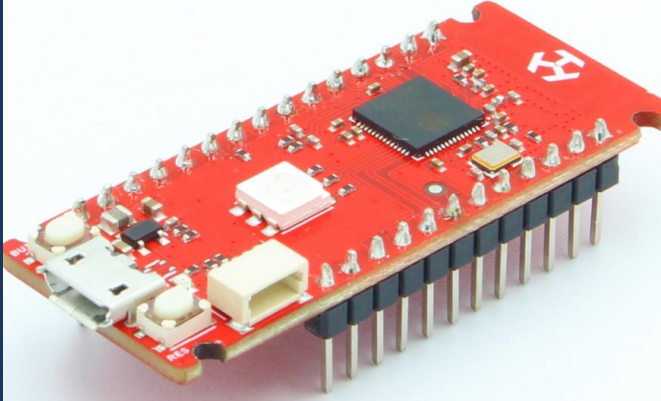
```
sadad | Arduino 1.8.15
Dosya Duzenle Tutarik Araclar Yardim

sadad$

#include <ESP32_S2_FastTimerInterrupt.h>
#include <ESP32_S2_ISR_Servo.h>
#include <ESP32_S2_ISR_Servo_Debug.h>
int sensor = D2;
int hareket;
ISR_servo_t servol =
{
  -1, D5
};
void setup() {
  delay(200);
  ESP32_ISR_Servos.useTimer(USE_ESP32_TIMER_NO);
  servol.servoIndex = ESP32_ISR_Servos.setupServo(servol.servoPin, MIN_MICROS, MAX_MICROS);
  pinMode(sensor, INPUT);
}
void loop() {
  hareket= digitalRead(sensor);
  if(hareket==HIGH) {
    ESP32_ISR_Servos.setPosition(servol.servoIndex, 45); //pozisyon kodu
    delay(500);
    ESP32_ISR_Servos.setPosition(servol.servoIndex, 60); //pozisyon kodu
    delay(500);
    ESP32_ISR_Servos.setPosition(servol.servoIndex, 120); //pozisyon kodu
    delay(1000);
  }
}
```

Bu örneğimizde, hareket sensörü ile birlikte servo motor kontrol edeceğiz. Hareket algılandığında servo motorumuz belirlenen açılar ile dönecek. Bunun için öncelikle gerekli olan servo motor kütüphanelerini kütüphane ekleme kısmından `#include` komutu ile ekliyoruz. `Void setup();` fonksiyonundan önce pin tanımlamasını, değişkeni ve servo motor tanımlamasını `servo1` adında D5 pininde olduğunu belirterek gerçekleştiriyoruz.

`Void setup();` fonksiyonu içerisinde servo motor tanımlamalarını ve sensörün giriş modunda olduğunu belirtiyoruz. `Void loop();` fonksiyonu içerisinde ise sensörden gelen değeri alıp hareket adındaki değişkenimize atıyoruz. Ardından koşul döngüsünü kullanarak `"if(hareket== HIGH)"` komutu ile eğer hareket varsa servo motorun `"ESP32_ISR_Servos.setPosition(servo1.servoIndex, 45);"` kodu ile D5 pinine bağlı olan `servo1` adındaki servo motorumuzun 45 derecelik açı olarak hareket etmesini ve 500 milisaniye beklemesini istiyoruz. Bunu istediğiniz kadar ve farklı açılarla tekrar edebilirsiniz.



PROJE NO: 12

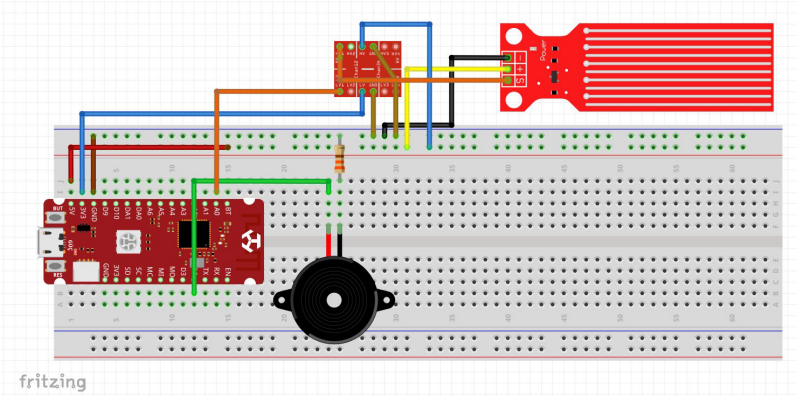
DENEYAP MiNi İLE YAĞMUR ALARMI YAPIMI



Gerekli Malzemeler:

- Deneyap Mini
- Breadboard
- 1 adet Dişi-Dişi Jumper Kablo
- 7 adet Dişi-Erkek Jumper Kablo
- 3 adet Erkek-Erkek Jumper Kablo
- Yağmur Sensörü
- Buzzer
- 1 adet Seviye Dönüştürücü

Devre Şeması:



“BİLGİ?”



Su Seviye Sensörü: Üzerine temas eden sıvıya göre sinyal çıkışı değiştirebilen sensör üzerindeki şerit şerit görümlü metal yüzeyler sayesinde ölçüm yapabilmektedir. Bu şeritlerin üzerinde halka ve ortasında boşluk bulunanlar güç şeritleri diğer şeritler ise ölçüm şeritleridir. Bu iki şerit arasında herhangi bir sıvı teması gerçekleştiğinde şeritler arası iletkenlik sayesinde bir sinyal çıkışı almaktayız.



”

Proje Kodları:

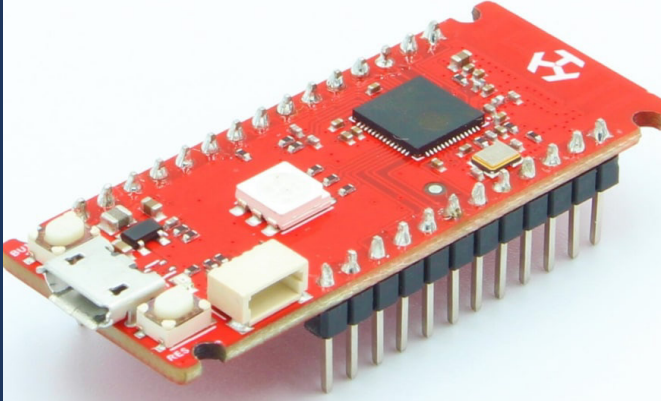
```
sketch_feb03a | Arduino 1.8.19 (Windows Store 1.8.57.0)
Dosya Düzenle Taslak Araçlar Yardım
sketch_feb03a.g
#define su A0
#define buzzer D2
int esikdegeri= 500;
int veri;
void setup() {
  pinMode(su, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(115200);
}
void loop() {
  veri = analogRead(su);
  if(veri>esikdegeri)
  {
    digitalWrite(buzzer, HIGH);
  }
  else{
    digitalWrite(buzzer, LOW);
  }
  Serial.print("Su Seviyesi : ");
  Serial.println(veri);
  delay(500);
}
```

Bu örneğimizde, yağmur sensörünün üzerindeki su miktarı belli bir eşik değerini geçerse bize sinyal gönderecek ve devre üzerindeki buzzer ses çıkarmaya başlayacak. Devremizi analog pin üzerinden aldığımız sinyale göre alarm verecek şekilde hazırladık.

İlk olarak sensörü, buzzer'ı, esikdegeri ve veri adlı değişkenlerimizi oluşturuyoruz. Voidsetup(); içerisinde sensörün giriş, buzzer'ın ise çıkış olduğunu pinMode(); ile belirtiyoruz. Ardından Serial.begin() ile seri haberleşmeyi başlatıyoruz.

Void loop(); fonksiyonu içerisinde ise su sensöründen gelen değeri veri değişkenine **"veri=analogRead(su);"** komutu ile atıyoruz. Daha sonra aldığımız değeri, eşik değeri ile if koşulu içerisinde kontrol ediyoruz. Eğer gelen veri eşik değerinden büyükse buzzer'ın çalmasını, küçükse çalmasını **"digitalWrite();"** komutu içerisinde belirtiyoruz.

Ardından seri ekrana yazdırma işlemlerini yapıp örneğimizi tamamlıyoruz.



PROJE NO: 13

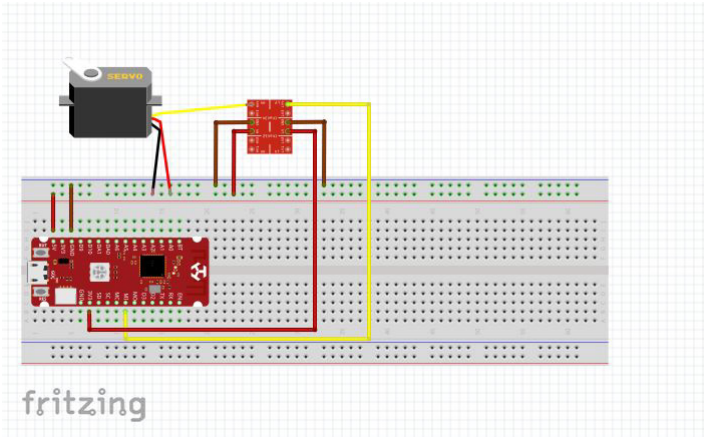
DENEYAP Mini WiFi İLE LAMBA AÇIP KAPATMA



Gerekli Malzemeler:

- Deneyap Mini
- SG90 Servo Motor
- 4 adet Erkek-Erkek Jumper Kablo
- 6 adet Erkek-Dişi Jumper Kablo
- BreadBoard
- 1 adet Seviye Dönüştürücü

Devre Şeması:



Proje Kodları:

```
Arduino 1.8.19 (Windows Store 1.8.57.0)
Doğru Düzenle Taslak Araçlar Yardım

_0t$
#include <ESP32_S2_FastTimerInterrupt.h>
#include <ESP32_S2_ISR_Servo.h>
#include <ESP32_S2_ISR_Servo_Debug.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
ISR_servo_t servo=
{
  -1, D5
};
const char *ssid = "*****";
const char *password = "*****";
WiFiServer server (80);
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.print("WIFI AĞINA BAĞLANILIYOR");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.println("Bağlanılıyor...");
    delay(300);
  }
  Serial.println("IP adresine bağlandı: ");
  Serial.print(WiFi.localIP());
  server.begin();
}
```

ot|Arduino 1.8.19 (Windows Store 1.8.57.0)
Doğa Düzenle Taslak Araçlar Yardım

```
Serial.println("Server başlatıldı");  
ESP32_ISR_Servos.useTimer(USE_ESP32_TIMER_NO);  
servo.servoIndex=ESP32_ISR_Servos.setupServo(servo.servoPin, MIN_MICROS, MAX_MICROS);  
ESP32_ISR_Servos.setPosition(servo.servoIndex,0);  
  
}  
  
void loop() {  
  WiFiClient client = server.available();  
  
  if(client){  
    {  
      Serial.println("Yeni istemci talebi geldi.");  
      String currentLine = "";  
      while(client.connected()){  
        {  
          if(client.available())  
          {  
            char c = client.read();  
            if (c == '\n')  
            {  
              if(currentLine.length() == 0)  
                <
```

ot|Arduino 1.8.19 (Windows Store 1.8.57.0)
Doğa Düzenle Taslak Araçlar Yardım

```
client.println("HTTP/1.1 200 OK");  
client.println("Content-type:text/html");  
client.println();  
client.print("<center>font-size:20' color:'blue'>Doğaygı Mıni Wifi ile Lamba Kontrolu/</center><br/>");  
client.print("<center>a href='\"/\"'/>Buton type='button'; style='width:50px;height:110px;font-size:30px'; Lambayı Aç/</center></a></center></br/>");  
client.print("<center>a href='\"/\"'/>Buton type='button'; style=' width:50px;height:110px;font-size:30px';Lambayı Kapat/</center></a></center></br/>");  
client.println();  
  
  break;  
  }  
  else  
  {  
    currentLine = "";  
  }  
  else if(c != '\n')  
  {  
    currentLine += c;  
  }  
  
  if(currentLine.endsWith("GET /H")          // Istemciden gelen talebi kontrol et, "GET /H" ya da "GET /L":  
  {  
    <
```

ot|Arduino 1.8.19 (Windows Store 1.8.57.0)
Doğa Düzenle Taslak Araçlar Yardım

```
    if(currentLine.endsWith("GET /H"))  
    {  
      ESP32_ISR_Servos.setPosition(servo.servoIndex,-45);  
      delay(1000);  
      ESP32_ISR_Servos.setPosition(servo.servoIndex,0);  
    }  
    if(currentLine.endsWith("GET /L"))  
    {  
      ESP32_ISR_Servos.setPosition(servo.servoIndex,90);  
      delay(1000);  
      ESP32_ISR_Servos.setPosition(servo.servoIndex,0);  
    }  
  }  
  }  
  }  
  
  client.stop();  
  Serial.println("Client Disconnected.");  
}  
  
}<
```

Bu örneğimizde, Deneyap Mini üzerinde bulunan wifi modülü ile internet üzerinden kartımıza bağlı olan servo motoru kontrol edeceğiz.

İlk olarak servo ve wifi için gerekli olan kütüphane tanımlamalarını gerçekleştiriyoruz. Daha sonra servo motorumuzu D5 pinine tanımlıyoruz. Ardından ssid ve şifre ile kartımızın bağlanacağı wifi ağının adını ve şifresini tırnak işaretleri (" ") içerisinde belirtiyoruz ve **“Wifiserver”** ile server tanımlamasını gerçekleştiriyoruz.

Void setup içerisinde seri ekranımızı başlatıyoruz. **“WiFi.begin(ssid, password);”** ile wifi ağına önceden yazılan adres adı ve şifreyi kullanarak bağlanıyoruz. Sonrasında seri ekrana yazdırıyoruz. While döngüsü içerisinde wifi ağına bağlanana kadar ekranda **“Serial.println(“Bağlanılıyor...”);”** yazmasını ve 300ms beklemesini söylüyoruz.

```
while (koşul)
{
  DENEYAP KART
  // kod veya döngünün gövdesi
}
```

Ağa bağlandıktan sonra seri ekrana ip adresini yazdırıyoruz. Ardından servo motor tanımlamalarını gerçekleştirip program başladığında servonun 0 derecede konumlanmasını **“ESP32_ISR_Servos.setPosition(servo.servoIndex,0);”** komutu ile söylüyoruz.

Void loop fonksiyonu içerisinde **“WiFiClient client = server.available();”** komutu ile bağlantı taleplerini dinliyoruz. **“if(client)”** istemci talebinin varlığını kontrol ediyoruz. Ardından veri kontrolü gerçekleştirip veriyi okuyarak eğer istemci talepte bulunmuşsa bir HTML sayfası oluşturup başlık ve 2 adet buton yerleştiriyoruz.

Son olarak istemciden gelen talebi kontrol ediyoruz.

Eğer “Lambayı Aç” butonuna tıklandı ise;

```
“ESP32_ISR_Servos.setPosition(servo.servoIndex,-45);  
delay(1000);
```

ESP32_ISR_Servos.setPosition(servo.servoIndex,0);” komutları ile servoyu lambanın anahtarını açacak şekilde hareket ettirerek tekrar 0 konumuna getiriyoruz.

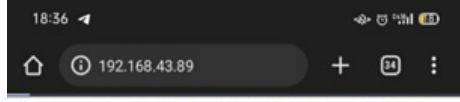
“Lambayı Kapat” butonuna tıklandı ise;

```
“ESP32_ISR_Servos.setPosition(servo.servoIndex,90);  
delay(1000);
```

ESP32_ISR_Servos.setPosition(servo.servoIndex,0);” komutları ile lambanın anahtarını kapatacak biçimde konumlandırmasını yapıp tekrar 0 konumuna gelmesini belirtiyoruz. Ardından **“client.stop();”** komutu ile bağlantıyı durdurup yeni talep bekliyoruz.

Kodların yazımı tamamlandıktan sonra kartımıza yüklüyoruz ve hazırladığımız devreyi evimizde kontrol etmek istediğimiz lambanın anahtarına denk gelecek şekilde yerleştiriyoruz. Ardından seri ekranında çıkan ip adresini telefonumuzdaki tarayıcıya yazıp hazırlamış olduğumuz html sayfasına gidiyoruz.

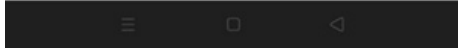
```
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
Bağlanılıyor...  
IP adresine bağlandı:  
192.168.43.89Server baslatildi
```



Deneyap Mini Wifi ile Lamba Kontrolu

Lambayi Ac

Lambayi Kapa



Sayfadaki butonlar ile lamba kontrolü artık cebimizde!



DENEYAP

Mini

UYGULAMA KİTABI

