

Ölçme Kontrol ve Otomasyon Sistemleri

12

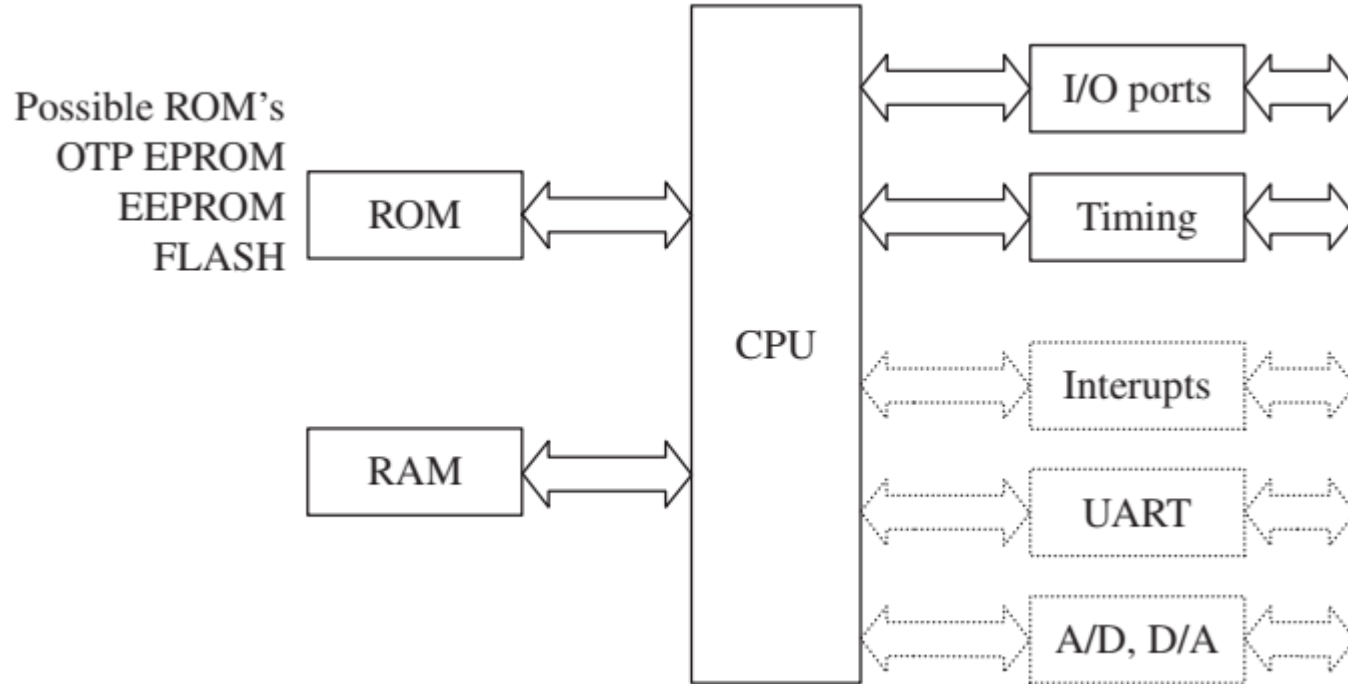
Dr. Mehmet Ali DAYIOĞLU

Ankara Üniversitesi Ziraat Fakültesi
Tarım Makinaları ve Teknolojileri Mühendisliği Bölümü

Microcontroller

Mikro denetleyici mimarisi

Mikrodenetleyici temel olarak çip üzerindeki bir bilgisayardır. Bir işlem birimi ROM, RAM, seri iletişim bağlantı noktaları, ADC'ler vb. içerir. Aslında, mikro denetleyici monitör, klavye ve fare olmayan bir bilgisayardır.



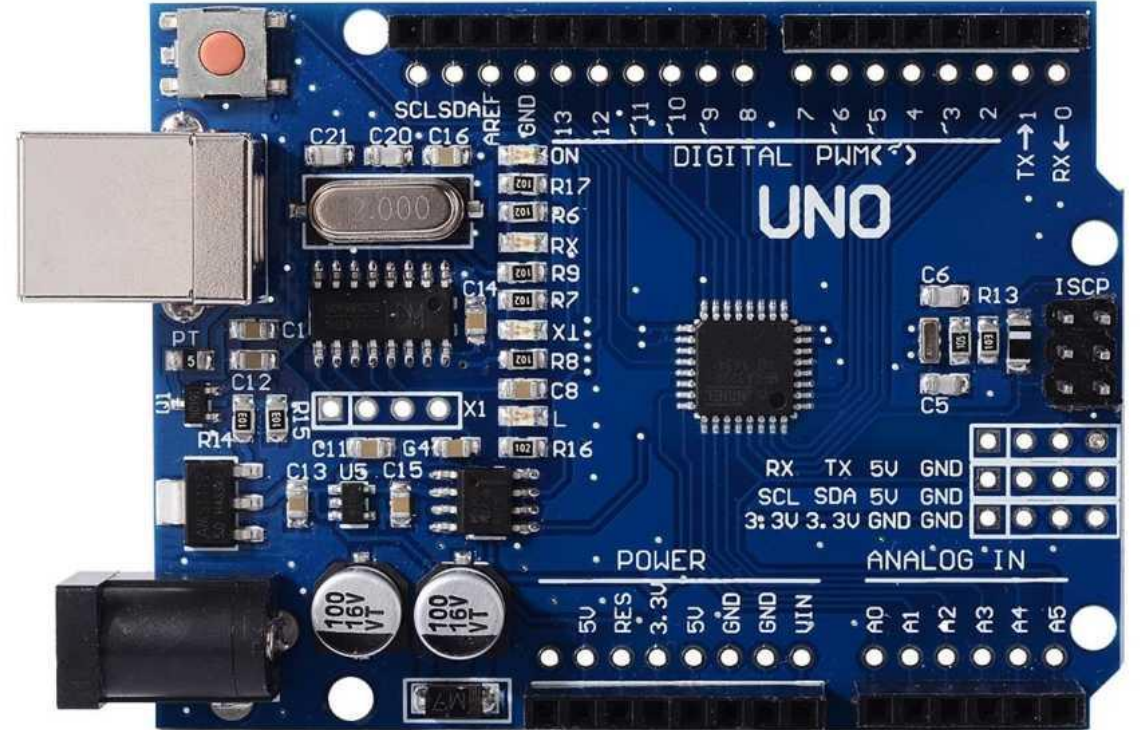
Arduino

Arduino, kullanımı kolay, açık kaynak kodlu yazılım ve donanıma sahip bir **mikrodenetleyici** prototipleme platformudur. Arduino mikrodenetleyici geliştirme kartı ve IDE içerir. IDE, Mac, Linux ve Windows bilgisayarları için mevcut olup, kullanımı kolaydır.

Orjinal

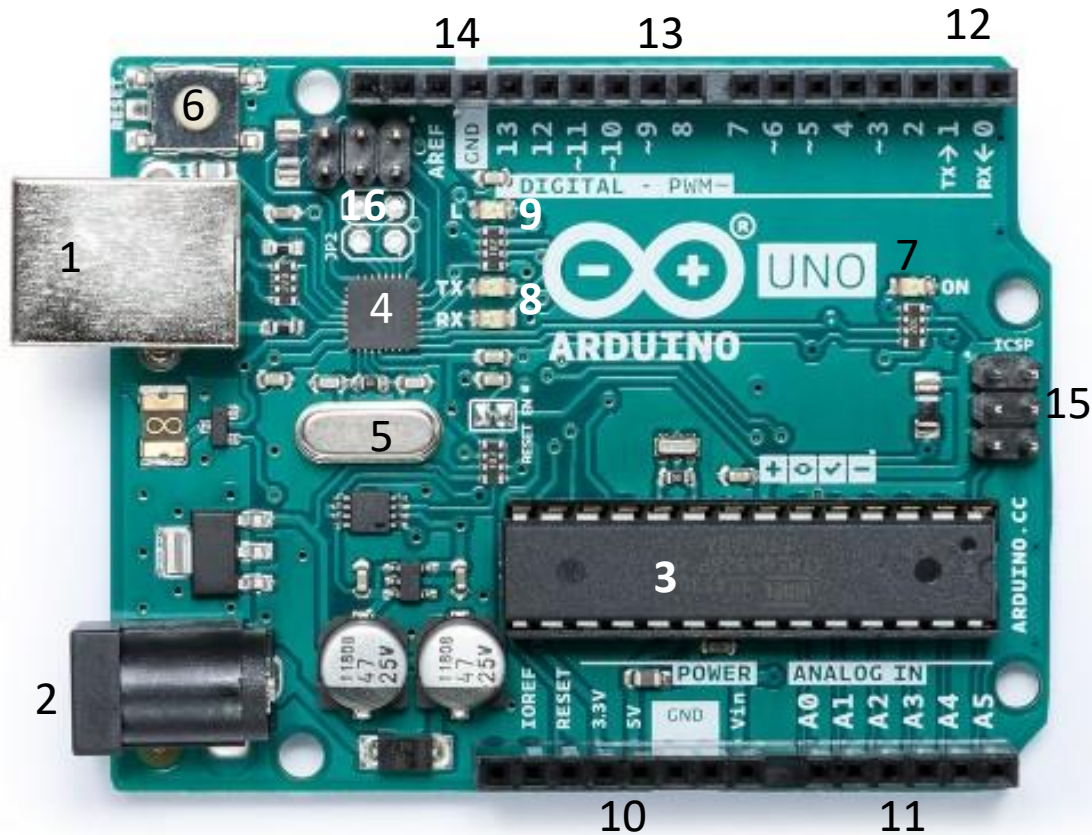


Klon



Arduino UNO

- Mikroişlemci: ATmega328 (8-bit)
- SRAM: 2KB
- Flash Memory: 32KB
- Saat Hızı: 16MHz
- Analog Giriş: 6 Adet
- Sayısal Giriş: 14 (6 tanesi PWM)



- 1 : USB jakı
- 2 : Power jakı (7-12 V DC)
- 3 : Mikrodenetleyici ATmega328
- 4 : Haberleşme çipi
- 5 : 16 MHz kristal
- 6 : Reset butonu
- 7 : Power ledi
- 8 : TX / NX ledleri
- 9 : Led
- 10 : Power pinleri
- 11 : Analog girişleri
- 12 : TX / RX pinleri
- 13 : Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)
- 14 : Ground ve AREF pinleri
- 15 : ATmega328 için ICSP
- 16 : USB arayüzü için ICSP

Arduino IDE

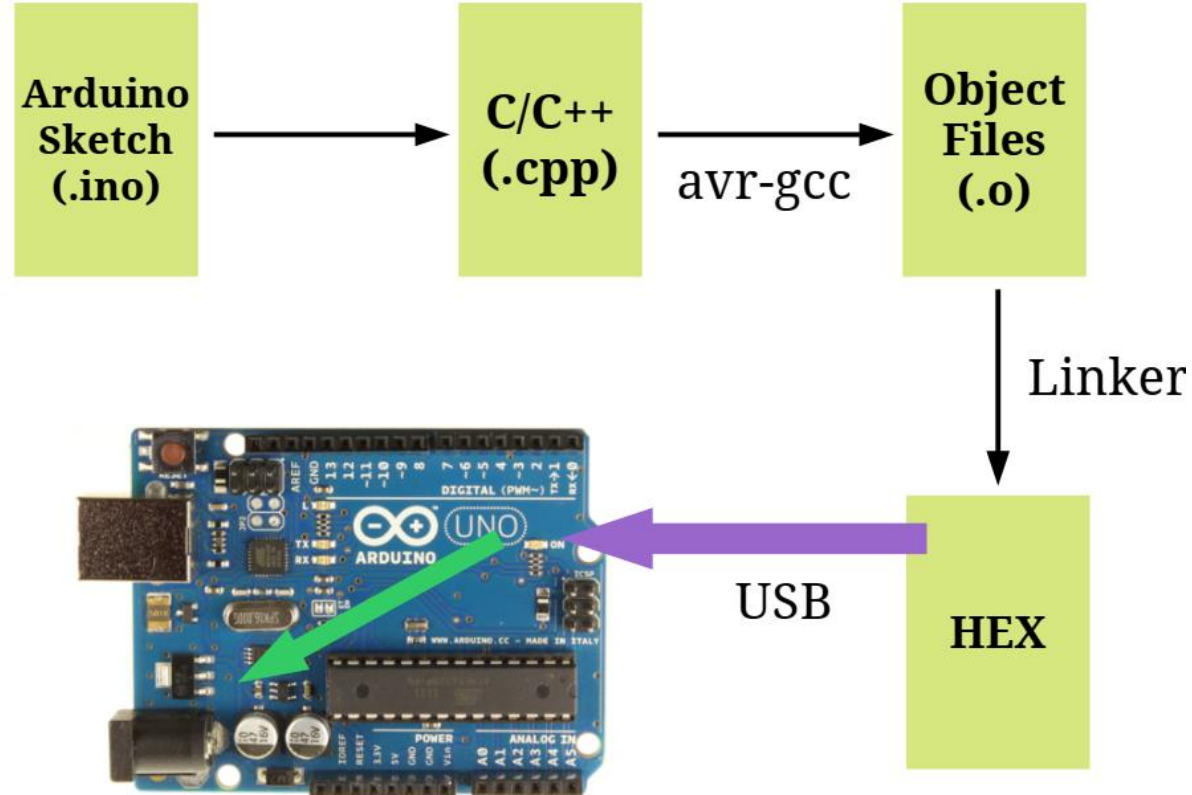
Platform bağımsız: Linux, Mac, Windows

- Alt seviyede gcc derleyicisini kullanıyor,
 - IDE Java ile geliştiriliyor,
 - Açık kaynaklı:
- <https://github.com/arduino/Arduino>

Arduino Programlama Dili

Arduino, Wiring tabanlı C/C++'a çok yakın bir dil ile programlanıyor,

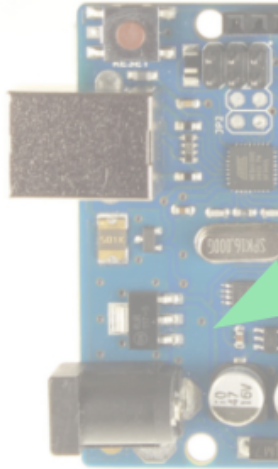
- <http://wiring.org.co>
- Arduino IDE ortamı ise Processing temel alınarak geliştiriliyor,
 - <http://www.processing.org>
- Arduino kaynak kod dosyalarına (*.ino) Sketch (taslak) adı veriliyor.



Arduino IDE

Arduino
Sketch
(.ino)

Tek tuş!



```
Blink
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Arduino IDE

• void setup ()

- ✓ Pinlerin Giriş mi Çıkış mı Olacağını Buraya Yazarız ve de Serial Monitör Ekranını Ekleyebiliriz.
- ✓ İlk Değerler Ve İlk Başta Bir Kere Çalışacak Kodlar Yazılır.

• void loop ()

- ✓ Sürekli Tekrarlanarak Sırasıyla Çalışacak Kodları Burada Yazıyoruz.

• digitalWrite

- ✓ Bu Komut ile Pinlerimize 0V ya da 5V Güç Veririz. Örnek Olarak;

- ❖ **Kullanımı** ---> **digitalWrite(değişken, HIGH); ya da (değişken, LOW);**
- ❖ digitalWrite(LED, HIGH); ---> Burada LED İsimli Değişkene Güç Verdik.
- ❖ digitalWrite(motor, LOW); ---> Burada İse Motor İsimli Değişkeni Durdurduk.

• analogWrite

- ✓ Bu Komutla İse 0 ve 5V Arası Değerleri Vermemizi Sağlıyor. Gelen Ve Giden Elektrik Akımı Her Zaman 0-255 arasındır. Bu Komutun Yararı İse Bazı Şeylerin Hızıyla Oynamamızı Sağlıyor. Örneğin Motorun Yavaş, Orta Hızlı Veya Çok Hızlı Olarak Sürmemizi Sağlıyor.

- ❖ **Kullanımı** ---> **analogWrite(değişken, LOW)**
- ❖ digitalWrite(motor, 255); ---> Burada Motora 5V Verdik.
- ❖ digitalWrite(motor, 204); ---> Burada Motora 4V Verdik.

• digitalRead

- ✓ Bu Komut ile Gelen 0 ve 5V Değerlerini Okumamızı Sağlar.
- ✓ Dijital olarak 5V = 1'e 0V ise = 0'a Eşittir.

- ❖ **Kullanımı** ---> **digitalRead(değişken);**
- ❖ digitalRead(blueetooth); ---> Burada Bluetooth'dan Gelen Verileri Okuduk.
- ❖ digitalRead(potansiyometre); ---> Burada Potansiyometre'den Gelen Verileri Okuduk.

• analogRead

- ✓ Bu Komut ile Gelen 0-5V arasındaki Değerlerini Okumamızı Sağlar.

- ❖ **Kullanımı** ---> **analogRead(değişken);**
- ❖ analogRead(blueetooth); ---> Burada Bluetooth'dan Gelen Verileri Okuduk.
- ❖ analogRead(potansiyometre); ---> Burada Potansiyometre'den Gelen Verileri Okuduk.

• delay();

- ✓ Bu Komut ile Bekleme Süresini Ayarlıyoruz.

- ❖ **Kullanımı** ---> **delay(beklenecek zaman);**
- ❖ delayMicroseconds(100); ---> 100 Mili Saniye Bekliyor.
- ❖ delaySeconds(100); ---> 100 Saniye Bekliyor.
- ❖ delay(100); ---> Eğer "Seconds, Microseconds vb." şeyleri yazmazsak Arduino bunu MS olarak anlar.

• Serial.begin(haberleşme_hızı);

- ✓ Bu Komut ile Seri Haberleşmeyi Açıyoruz. Bu Serial Haberleşmeden Metin, Bilgi, Sayı Gönderebilir ya da Alabiliriz. Genelde Haberleşme Hızı 9600'dur.

- ❖ **Kullanımı** ---> **Serial.begin(haberleşme_hızı);**
- ❖ Serial.begin(9600); ---> 9600 Baund Hızında Seri Haberleşmeyi Başlattık.

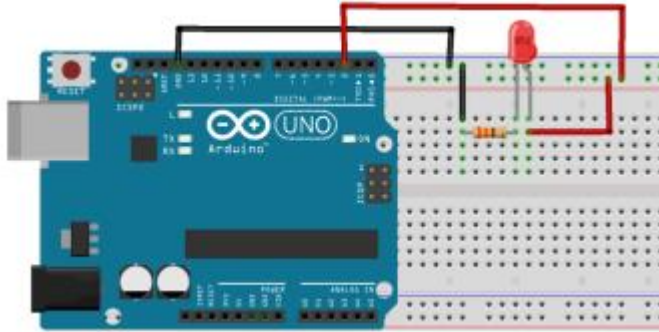
• pinMode();

- ✓ Bu komut ile Pinlerin Giriş mi Yoksa Çıkış mı Olacağına Karar Veriyoruz.
- ❖ Çıkış(**OUTPUT**) Olarak Kullanılanlar ---> LED, Motor, LCD Ekran
- ❖ Giriş(**INPUT**) Olarak Kullanılanlar ---> Potansiyometre, Buton,
- ❖ **Kullanımı** ---> **pinMode(Pin Numarası ya da Değişken Adı , OUTPUT ya da INPUT);**
- ❖ **Örnek** ---> pinMode(motor, OUTPUT); ---- pinMode(11, INPUT);

• Serial.print();

- ✓ Bu Komut ile Rastgele şeyler ya da Sensörlerden Gelen Verileri Ekran Yazdırabiliriz.
- ❖ **Kullanımı** ---> **Serial.print(değişken); Serial.print("Merhaba Dünya!");**
- ❖ Serial.print(blueetooth); ---> Bluetooth'dan Gelen Verileri Yan yana Ekran Yazdırıyoruz.
- ❖ Serial.println(blueetooth); ---> Bluetooth'dan Gelen Verileri Alt alta Yazdırıyoruz.

Proje: LED



```
int led = 2;
void setup()
{
  pinMode(led,OUTPUT);
}
void loop()
{
  digitalWrite(led,HIGH);
  delay(100);
  digitalWrite(led,LOW);
  delay(100);
}
```


Kaynaklar (References)

1. M. Nacar, 2015. Elektrik – Elektronik Ölçmeleri ve İş Güvenliği, Ankara Ofset Matbaacılık
2. J. P. Holman, 2012. Experimental methods for engineers —8th ed., McGraw-Hill series in mechanical engineering
3. S. Monk , P. Scherz, 2016. Practical Electronics for Inventors,Yayınevi : McGraw-Hill Education
4. D. J. Curtis, 2014. Process Control Instrumentation Technology, Pearson, Eighth Edition
5. M. A. Dayıođlu, 2017. 6. Ünite: Seralarda Bilişim ve Otomasyon Teknolojisi, Sayfa: 102 – 134, Kitap Adı: Örtüaltı Üretim Sistemleri, 3. BaskıAnadolu Üniversitesi Yayın No: 2275
6. M. W. Birimicombe, M.A. D. Phil, 2000. Introduction electronic systems, Nelson
7. H. Pastacı, 2017. Elektrik ve Elektronik Ölçmeleri, 11. Baskı, Nobel Yayıncılık, Ankara
8. W. C. Dunn, 2005. Fundamentals of Industrial Instrumentation and Process Control, McGraw-Hill
9. J. Fraden, 2010. Handbook of Modern Sensors Physics, Designs, and Applications, Fourth Edition, Springer