



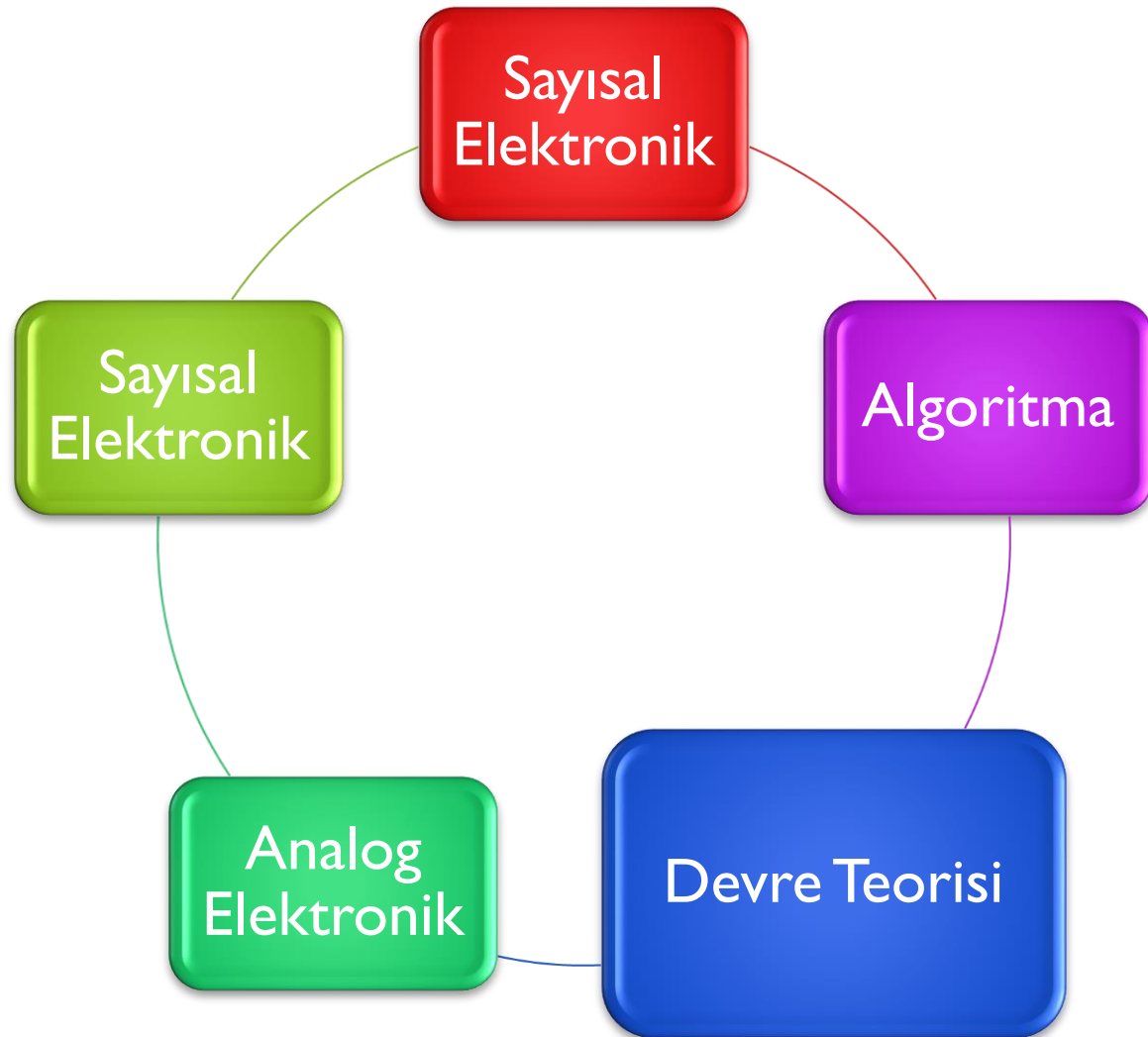
MİKRODENETLEYİCİLER (ARDUINO)

1. Hafta

Hazırlayan: Öğ. Gör. Sabri Uzuner

https://www.researchgate.net/profile/Sabri_Uzuner

Sahip olunması gereken yeterlilikler



Ders Değerlendirme

- Proje 100%
- Not: Kesin değerlendirme yöntemi ve konular ileriki haftalarda belirlenecek.

Course Staff

Sabri UZUNER - Lecturer

- **Contacts**

- Room: Dr. Engin PAK Cumayeri MYO
- Phone: 0 380 735 40 48 - 51 99 Dahili: 7522
- E-mail: sabriuzuner@duzce.edu.tr

- Webpage:

https://www.researchgate.net/profile/Sabri_Uzuner

- **Office hours:** Çarşamba 9:30-17:00 (MYO) ve Salı 9:30-15:00 (Mühendislik)

Hafta	Konular	Kullanılan Öğretim Yöntemi ve Teknikleri	Ön Hazırlık
1. Hafta	Temel kavramlar: sayılar, bilgisayarlar ve mikrodenetleyiciler		
2. Hafta	Elektronik: dirençler, gerilim, akım ve Ohm Kanunu		
3. Hafta	Sayısal Mantık: transistörler, flip flop ve mantık fonksiyonları		
4. Hafta	C programlamaya giriş		
5. Hafta	Mikrodenetleyici Giriş / Çıkış		
6. Hafta	Tasarım ve Geliştirme Süreci		
7. Hafta	Anahtarlar ve LED'ler Arayüz		
8. Hafta	Arasınav		
9. Hafta	Diziler ve Fonksiyonel ayıklama		
10. Hafta	Sonlu Durum Makineleri		
11. Hafta	UART - Seri Arabirim, I / O Senkronizasyon		
12. Hafta	kesmeler		
13. Hafta	DAC ve Ses		
14. Hafta	ADC ve Veri Toplama		

MOTİVASYON

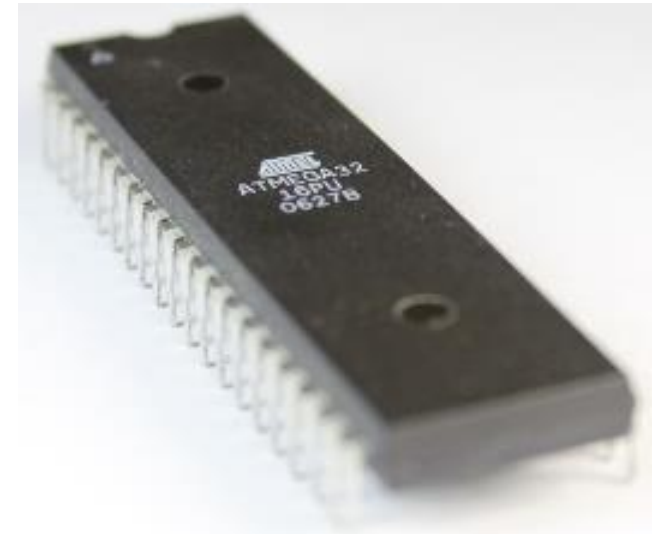
İnsansız hava araçları, robotlar, akıllı ev sistemleri gibi projeleri uygulamalı olarak yapabilmek.

Son yıllarda çok fazla yaygın olarak kullanılan ARDUINO mikrodenetleyicisi ile popüler projeleri rahatlıkla yapmak.

Devre sistemlerine ve burada kullanılan elektronik malzemelerine aşina olmak

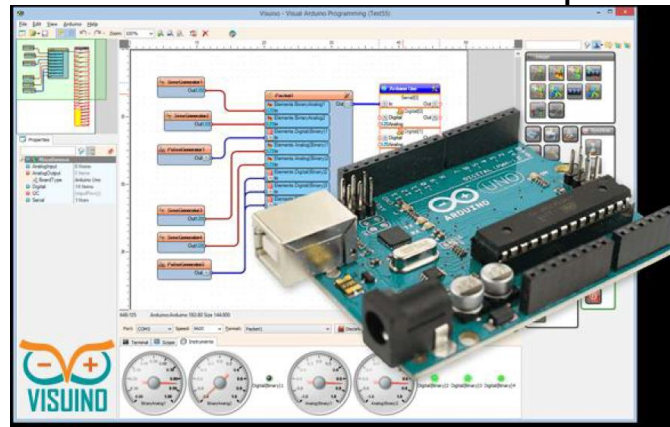
What is Microcontroller??

- ❑ Microcontrollers are small computers integrated into a single chip
- ❑ They contain
 1. Processing core
 2. Flash Memory for program
 3. I/O peripherals
 4. RAM
 5. Peripherals such as clocks, timers, PWM etc
- ❑ Microprocessors are used for general purpose applications, while microcontrollers are self sufficient and are used for specific tasks.
- ❑ Microcontrollers are an example of embedded systems.



What is ARDUINO??

- ❑ Arduino is an open-source physical computing platform.
- ❑ It is a small microcontroller board with a USB plug.
- ❑ Based on a simple i/o board and a development environment that implements the Processing/writing language.
- ❑ Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.
- ❑ Easy-to-use hardware and software.
- ❑ It's intended for students, artists, designers, hobbyists and anyone who tinker with technology.
- ❑ It is programmed in Arduino Programming language(APL) similar to C/C++.
- ❑ Way more easy to program compared to other microcontroller packages.
- ❑ The Arduino is a microcontroller development platform(not a microcontroller....)
- ❑ It is the winner of “**worlds best interaction award 2012**” sponsored by google



Why Arduino?

- ❑ It is Open Source, both in terms of **Hardware** and **Software**.
- ❑ It is cheap, (about \$20, the cost of going out for pizza)
- ❑ USB connectivity (MacBooks don't have serial ports)
- ❑ More powerful than a BASIC stamp (it costs around \$180)
- ❑ Simple and easy to use by someone without formal electronics training. Editing and rewriting is often easier than writing from scratch. That's where the BeagleBoard falls short. "It has virtually no example application that you can just copy and hack to learn from".
- ❑ We know simple is attractive.



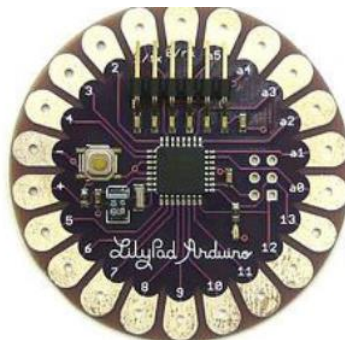
Arduino aynı zamanda İtalyancada dost anlamına gelmektedir.

Different flavours!!!

- ❑ There are many versions of Arduino board.versions differ by size,microcontroller,etc



MEGA



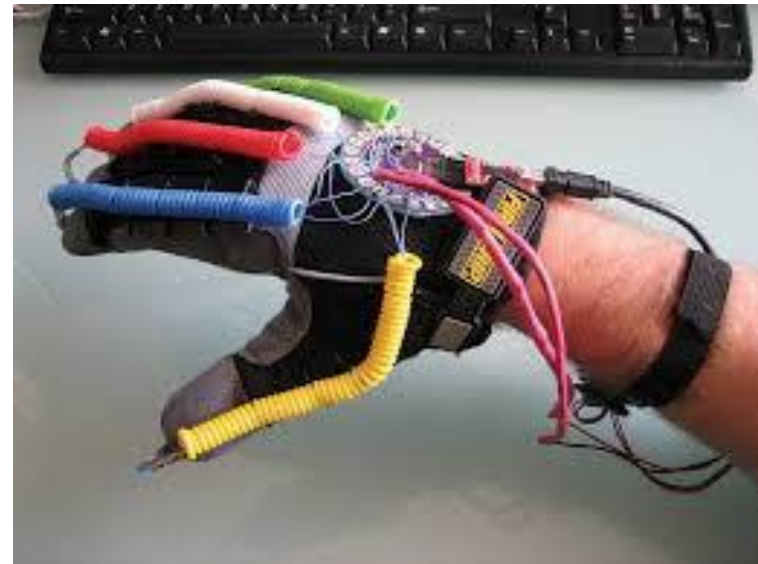
LILYPAD



MINI



NANO 43mm x 18mm



Arduino Pro Mini, Uno'dan farklı olarak fazladan 2 analog girişe sahiptir ve doğrudan USB kablosu üzerinden programlanmamaktadır. Pinleri lehimsiz olarak gelmektedir. Arduino Mini kartı ise Pro Mini ile neredeyse aynı olup sadece Breadboard üzerinde kablolama imkanı sunmaktadır.

Arduino DUE kartı diğer kartlardan çok daha farklı olup 32 Bit işlemciye sahip ilk Arduino modelidir. Üzerinde ARM mimarisine sahip 32 Bitlik işlemci vardır. Bu, diğer Arduino kartlarına göre çok daha fazla veri işleyebildiği anlamına gelmektedir. Yapı olarak Arduino Mega ile aynıdır.

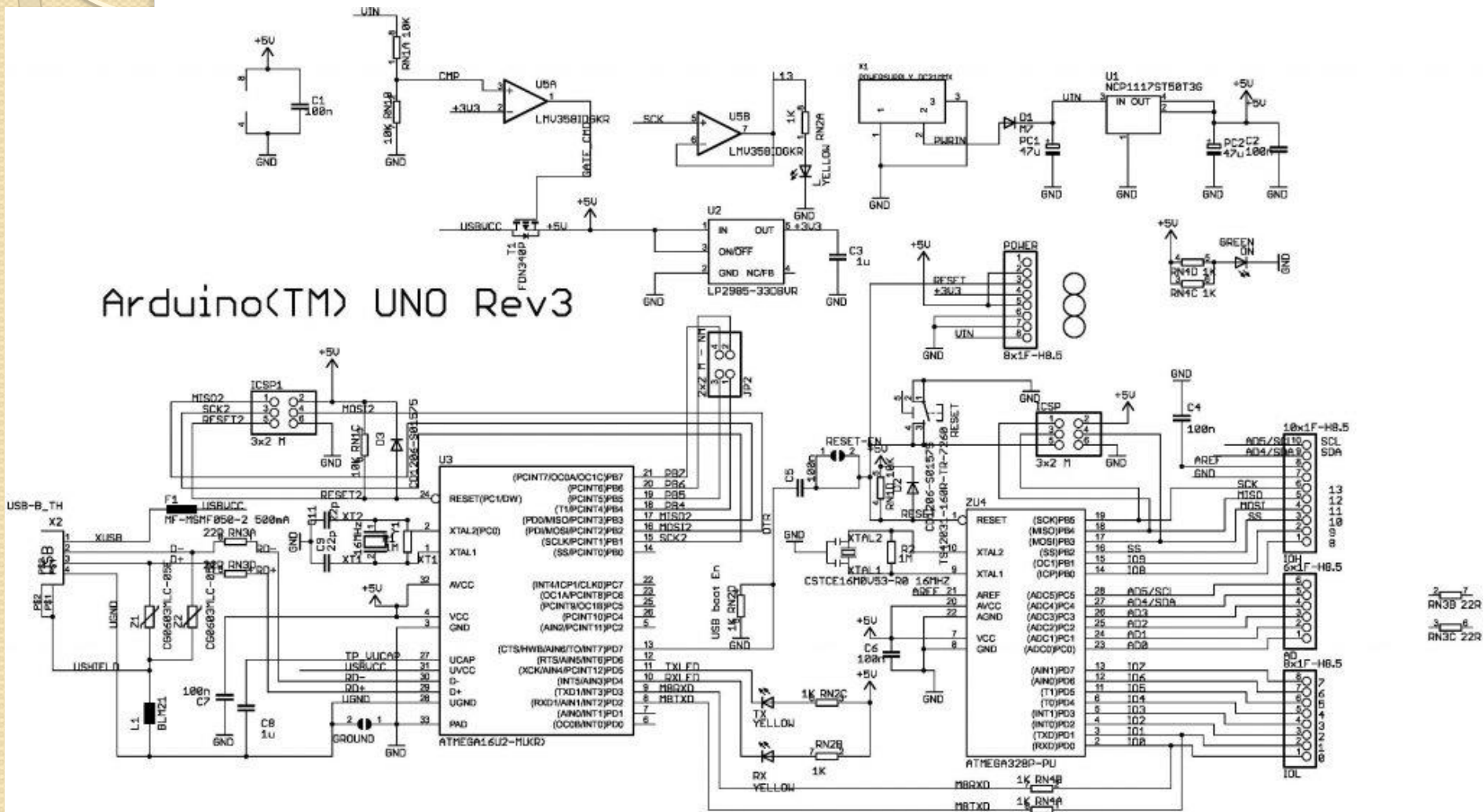
Arduino Leonardo modeli pin yapısı olarak Uno'ya benzemektedir. Uno'dan en büyük farkı üzerinde dahili bir serial dönüştürücü olmamasıdır. Bunun sebebi Leonardo üzerinde bulunan Atmega32u4 işlemcisinin kendi pinleri vardır. Kullanım olarak Uno'dan bir farkı yoktur.

Arduino Esplora yapı olarak diğer modellerden farklıdır, biraz daha kumandaya benzemektedir. Esplora'nın yapılış amacı Arduino'ya yeni başlayanlara çeşitli sensörlerin kullanımını öğretmektir. Üzerinde çeşitli sensörler, Joystick ve farklı elektronik komponentler bulunmaktadır.

Arduino YUN, üzerinde dahili olarak hem ethernet hem de WiFi modülü içermektedir. Kartın üzerinde hem bir Atmega mikroişlemci hem de Linux mikroişlemci bulunmaktadır. Host özellikli, Linux işlemciye bağlı ikinci bir USB portu bulunmaktadır. Kartın hafızası harici bellek ile artırılabilir.

Schematic Diagram

Arduino(TM) UNO Rev3



Arduino Uno Kartının Özellikleri

- ◆ Mikrodenetleyici: ATmega328
- ◆ Besleme Voltajı: 7-12V
- ◆ Çalışma Voltajı: 5V (Arduino UNO kaç voltta çalışır?)
- ◆ Giriş-Çıkış Pinleri Sayısı: 14 adet dijital pini mevcuttur.
Bunlardan 6 tanesi PWM olarak kullanılabilir. 6 pinden 8 bitlik analog sinyal çıkışı elde edilebilmektedir.
- ◆ Pinlerdeki Akım: 40mA(5V), 50mA(3.3V)
- ◆ FLASH: 32 KB (2 KB Bootloader tarafından kullanılıyor)
1 KB EEPROM
2 KB SRAM
- ◆ Çalışma Saat Hızı: 16MHz
- ◆ İletişimi de USB üzerinden sağlayabilmektedir.

Arduino Mega Kartının Özellikleri

Mikrodenetleyici : ATmega2560

Çalışma gerilimi : +5 V DC

Tavsiye edilen besleme gerilimi : 7 - 12 V
DC

Besleme gerilimi limitleri : 6 - 20 V

Dijital giriş / çıkış pinleri : 54 tane (15 tanesi PWM çıkışını destekler)

Analog giriş pinleri : 16 tane

Giriş / çıkış pini başına düşen DC akım :
40 mA

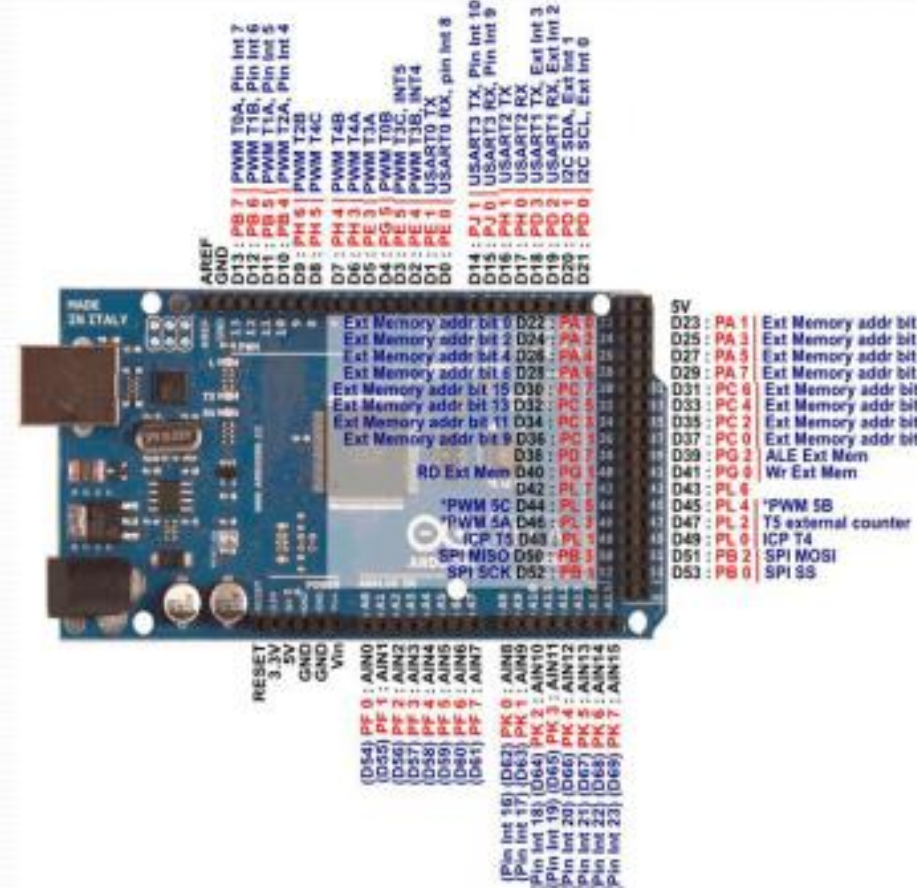
3,3 V pini için akım : 50 mA

Flash hafıza : 256 KB (8 KB bootloader için kullanılır)

SRAM : 8 KB

EEPROM : 4 KB

Saat frekansı : 16 MH



Arduino Nano Kartının Özellikleri

Mikrodenetleyici : Arduino Nano V3 te ATmega328
(önceki versiyonlarda ATmega168)

Çalışma gerilimi : +5 V DC

Tavsiye edilen besleme gerilimi : 7 - 12 V DC

Besleme gerilimi limitleri : 6 - 20 V

Dijital giriş / çıkış pinleri : 14 tane (6 tanesi PWM çıkışını destekler)

Analog giriş pinleri : 8 tane

Giriş / çıkış pini başına düşen DC akım : 40 mA

Flash hafıza : ATmega328 için 32 KB, ATmega168 için 16 KB (2 KB bootloader için kullanılır)

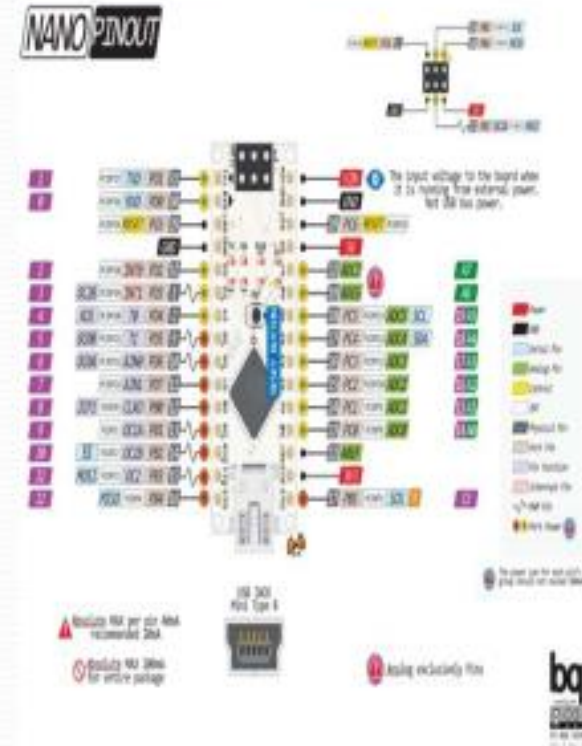
SRAM : ATmega328 için 2 KB, ATmega168 için 1 K

EEPROM : ATmega328 için 1 KB, ATmega168 için 512 byte

Saat frekansı : 16 MHz

Boyutları : 18 mm x 45 mm

Ağırlık : 5 g



Peki Projem İçin Hangi Kart Uygun ?

1-Projenin amacı belirlenmeli, bu amaçlar doğrultusunda kullanılacak çevre birimlerin belirlenmeli.

2-Proje için bütçe durumu belirlenmeli.

Yanıtlar doğrultusunda uygun geliştirme kartı seçilmelidir.

Arduino IDE sini nasıl yüklerim ?

https://www.youtube.com/watch?v=_2JJ29kmwRU



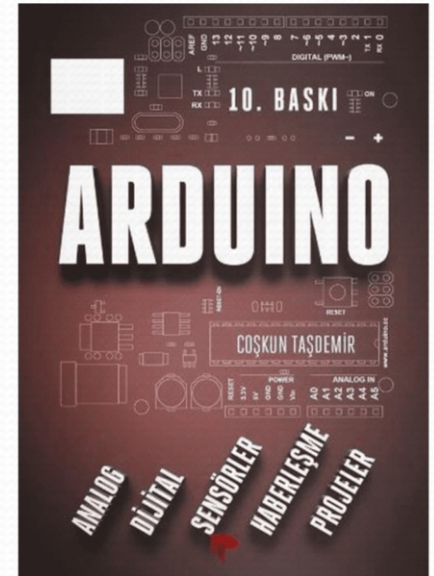
Arduino için kaynaklar

- Arduino – Coşkun TAŞDEMİR
- YOUTUBE === >>>>LEZZETLİ ROBOT TARİFLERİ

İNTERNET



Lezzetli Robot Tarifleri
82,5 B abone



ARDUINO:SOFTWARE

- ❑ Arduino boards can be controlled Using an implementation of Wiring, Which is a version of Processing developed specifically for electronic I/O.
- ❑ Arduino looks like Processing, but is actually built in C, so there are a few differences to look out for.
- ❑ Arduino.IDE can be downloaded from <http://www.arduino.cc>
- ❑ Currently used version is 004.

A screenshot of the Arduino IDE interface. The title bar reads "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area displays the "Blink" sketch. The code is as follows:

```
/*  
 * Blink  
 *  
 * The basic Arduino example. Turns on an LED on for one second,  
 * then off for one second, and so on... We use pin 13 because,  
 * depending on your Arduino board, it has either a built-in LED  
 * or a built-in resistor so that you need only an LED.  
 *  
 * http://www.arduino.cc/en/Tutorial/Blink  
 */  
  
int ledPin = 13;           // LED connected to digital pin 13  
  
void setup()               // run once, when the sketch starts  
{  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output  
}  
  
void loop()               // run over and over again  
{  
  digitalWrite(ledPin, HIGH); // sets the LED on  
  delay(1000);               // waits for a second  
  digitalWrite(ledPin, LOW);  // sets the LED off  
  delay(1000);               // waits for a second  
}  

```

The status bar at the bottom shows "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)". The page number "22" is visible in the bottom right corner.

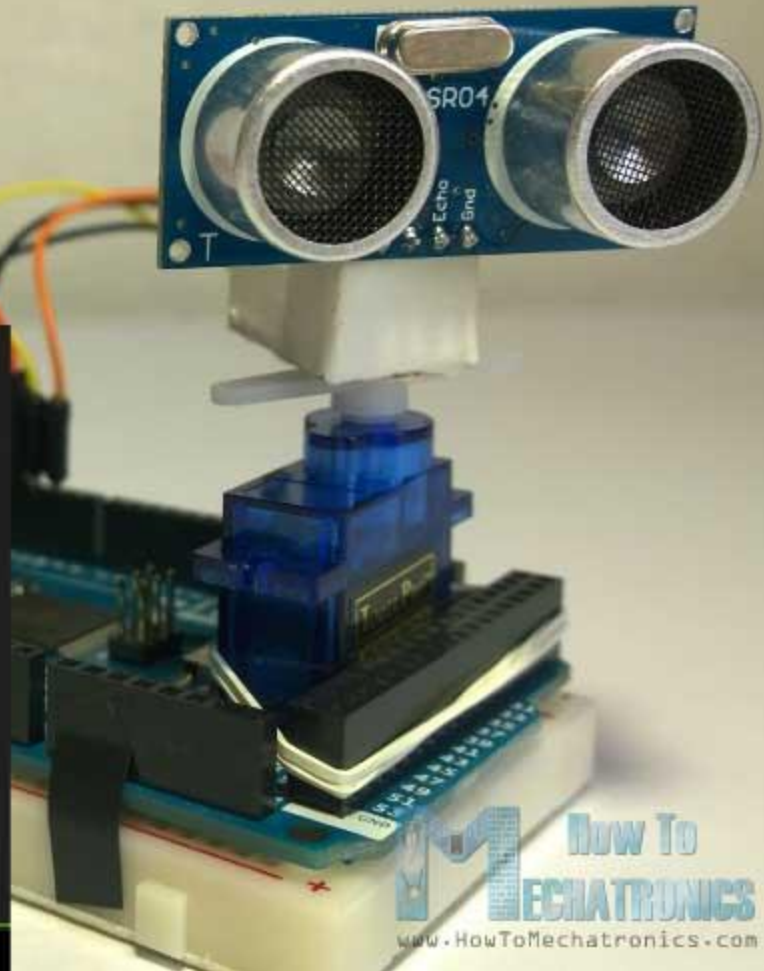
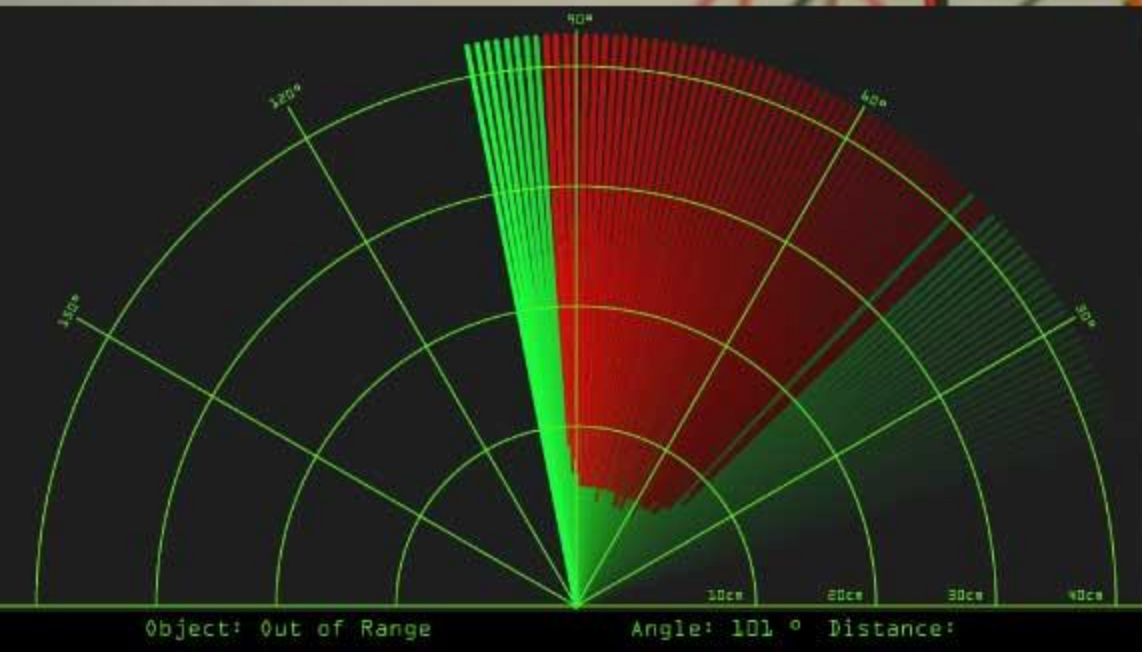
THE ARDUINO IDE

- ❑ The Arduino is programmed in C language.
- ❑ The language is very simple and provides many abstraction for simplicity of reading and writing powerful applications.
- ❑ It provides a serial monitor to see the serial data from the USB virtual COM port.
- ❑ Allows one click compiling, verification and burning of code onto the Arduino.



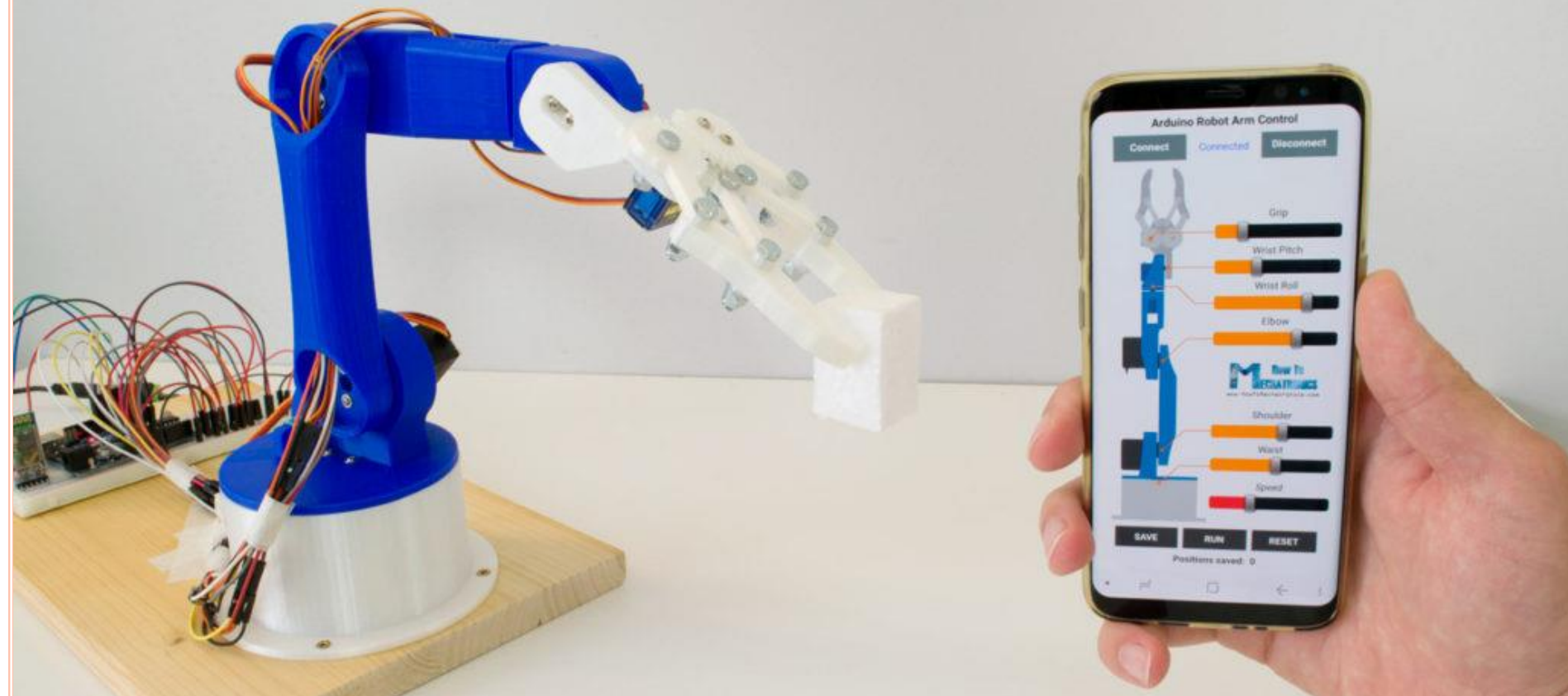
ARDUINO PROJECTS WITH DIY INSTRUCTIONS

How To Make an Arduino Radar



For this project you just need two components along with an Arduino board, and that's an ultrasonic sensor and small servo motor. The range of the radar can be adjusted to up to 4 meters with 180 degrees rotation.

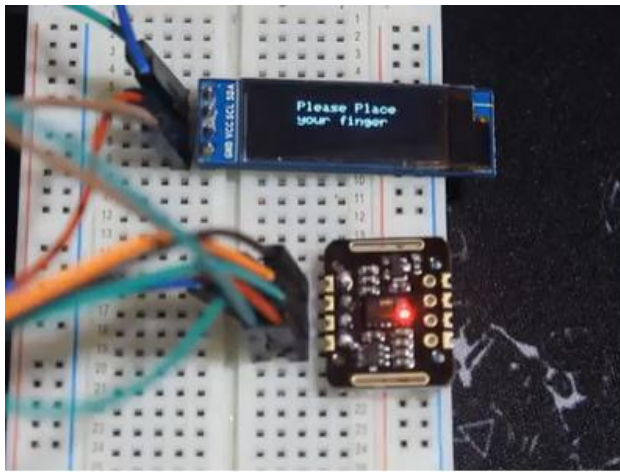
DIY Arduino Robot Arm



<https://howtomechatronics.com/arduino-projects/>







Measure Heart Rate and SpO2 with MAX30102

Project tutorial by SurtrTech



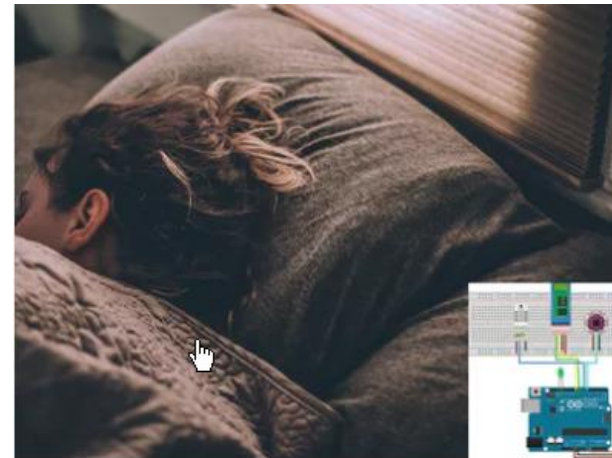
Purdue ExoMIND Glove

Project showcase by Purdue MIND



Alzheimer's Assistant

Project tutorial by Abdullah Sadiq



Sleeping Phases Determination

Project tutorial by Diana Khalipina

Biomedical projects

<https://create.arduino.cc/projecthub/projects/tags/health?page=2>





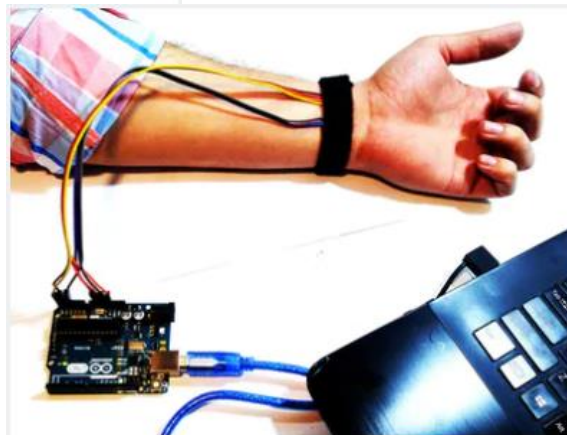
Flowers - Arduino Nano, CrazyCircuits, DFRobot

Project tutorial by Kitty Yeung



Heart Beat Indicator Using ECG

Project tutorial by Dmitry Dziuba



Arduino Polygraph Machine (Lie Detector)

Project tutorial by ElectroPeak

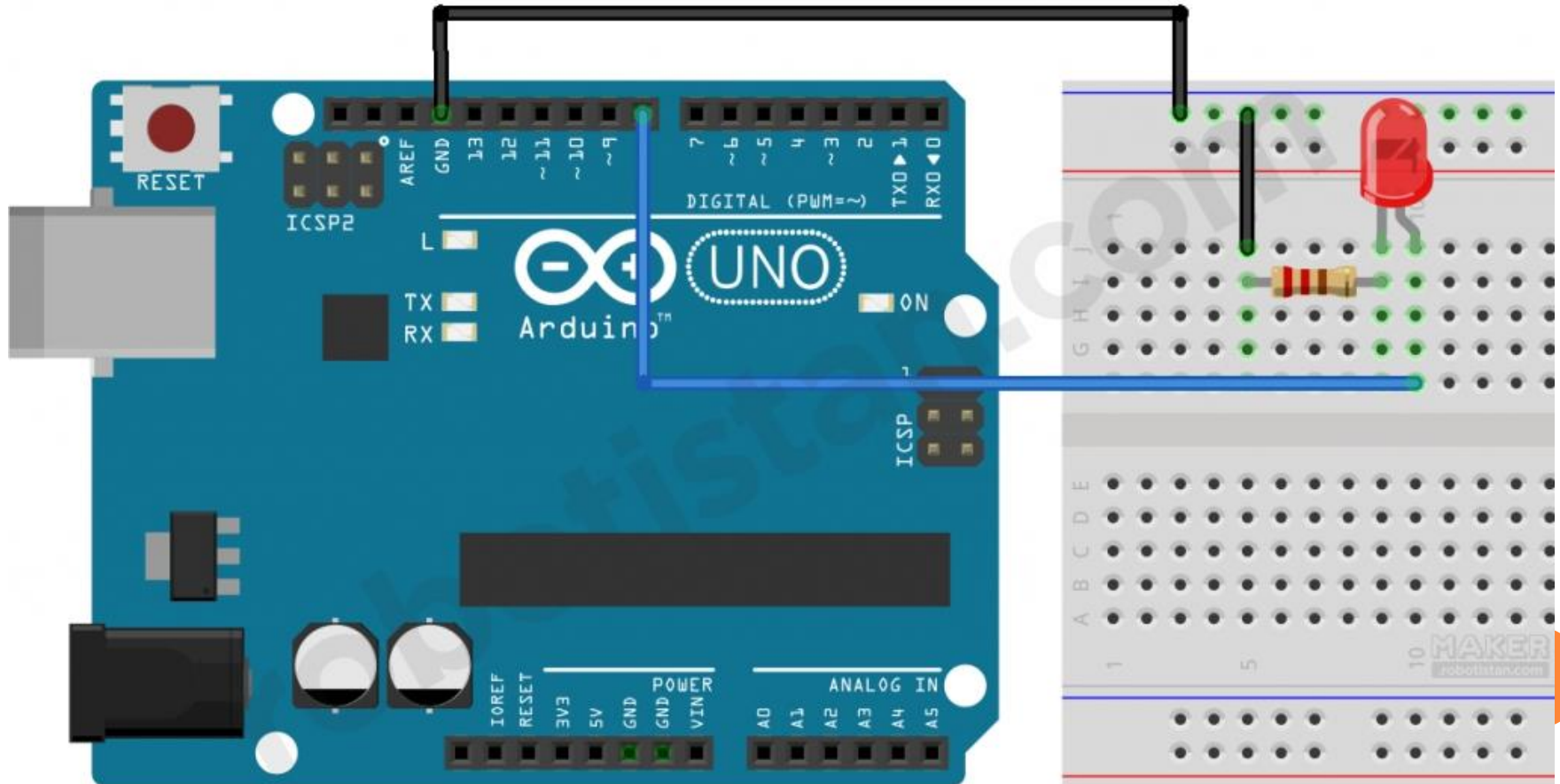


Malzeme Listesi

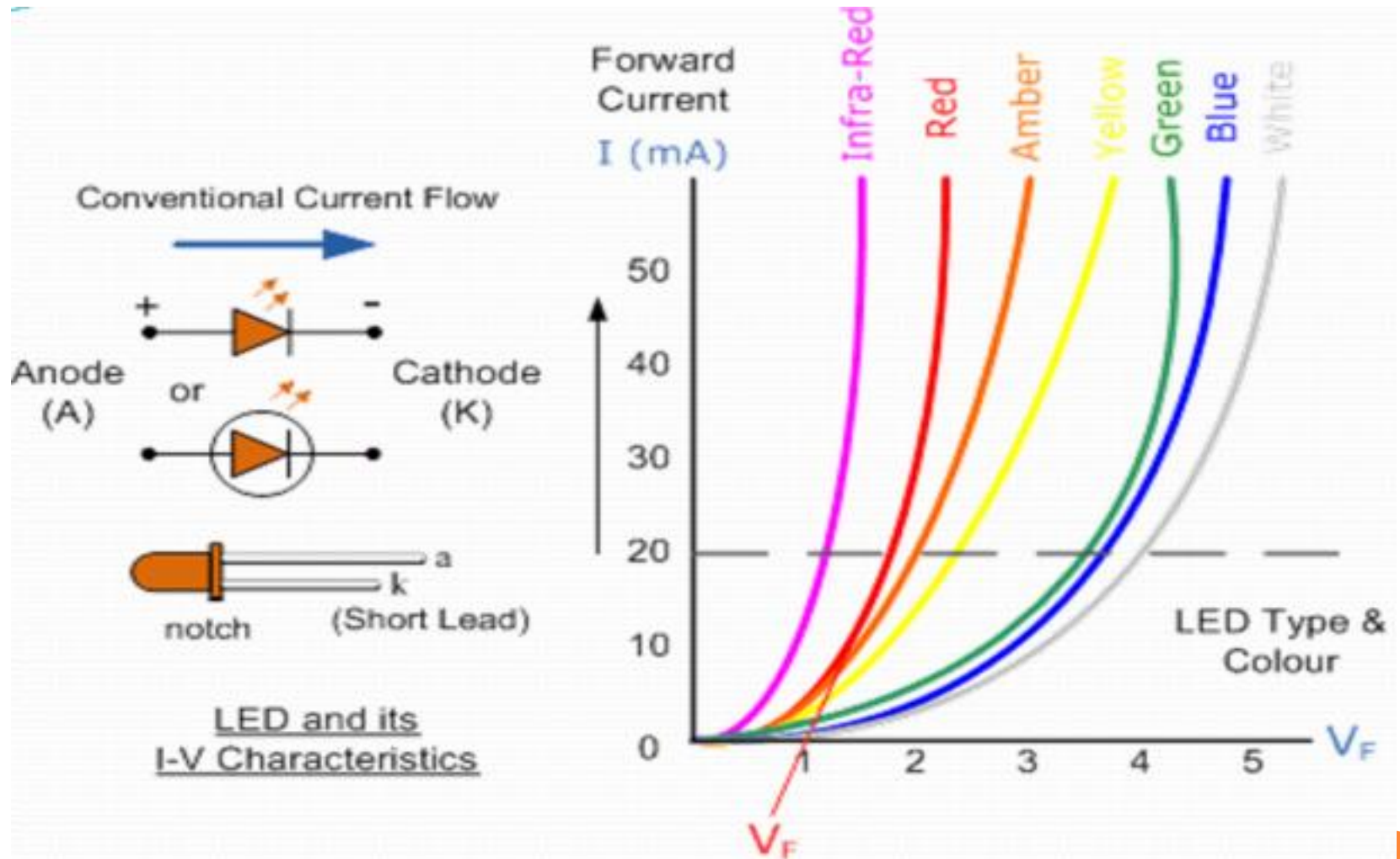
- Arduino Uno
- Breadboard
- LED
- 220 Ω direnç
- İki ucu erkek jumper kablo

1. UYGULAMA Arduino LED Yakma (Blink)

https://www.robotistan.com/arduino-baslangic-seti?utm_source=maker&utm_medium=banner

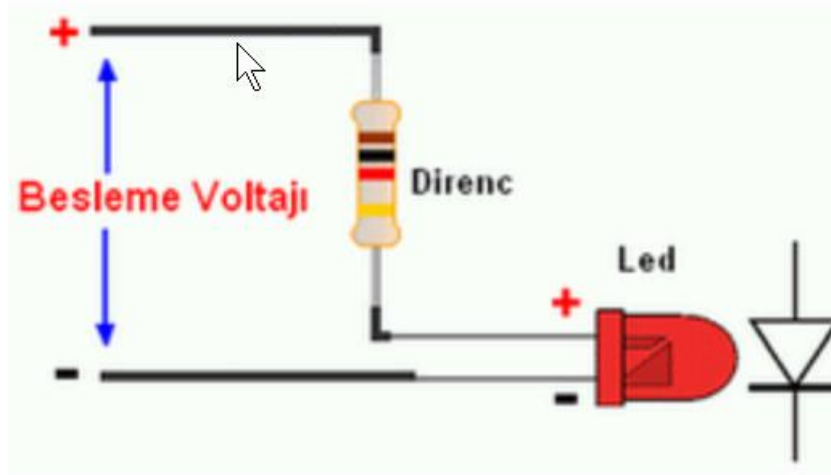


LED KORUMA DİRENCİ NE OLMALIDIR



LED KORUMA DİRENCİ NE OLMALIDIR

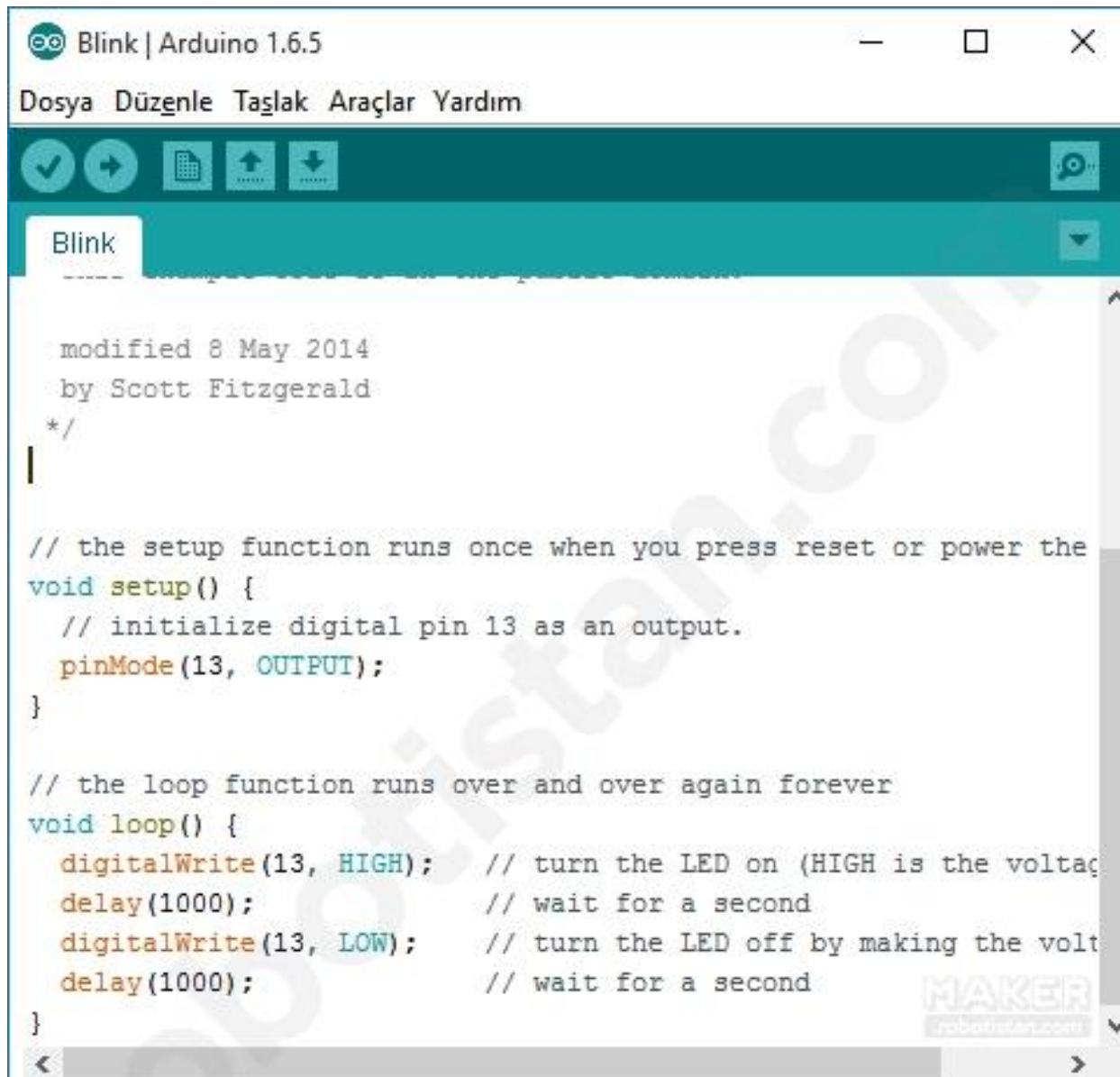
$$\text{Kullanılabilir Minimum Direnç Değeri} = \frac{\text{Besleme Voltajı} - (\text{LED'in Katalog Voltaj Değeri} \times \text{Seri Bağlı LED sayısı})}{\text{LED'in Katalog Akım Değeri}}$$



SORU: Besleme gerilimi için 12V gerilim kaynağı ve kırmızı led kullanılırsa koruma için kullanılacak direnç ne olmalıdır?



Program Kodu



The screenshot shows the Arduino IDE interface with the 'Blink' program loaded. The title bar reads 'Blink | Arduino 1.6.5'. The menu bar includes 'Dosya', 'Düzenle', 'Tağlak', 'Araçlar', and 'Yardım'. The toolbar contains icons for checking, undo, redo, saving, and uploading. The file name 'Blink' is shown in the tab. The code editor displays the following code:

```
modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

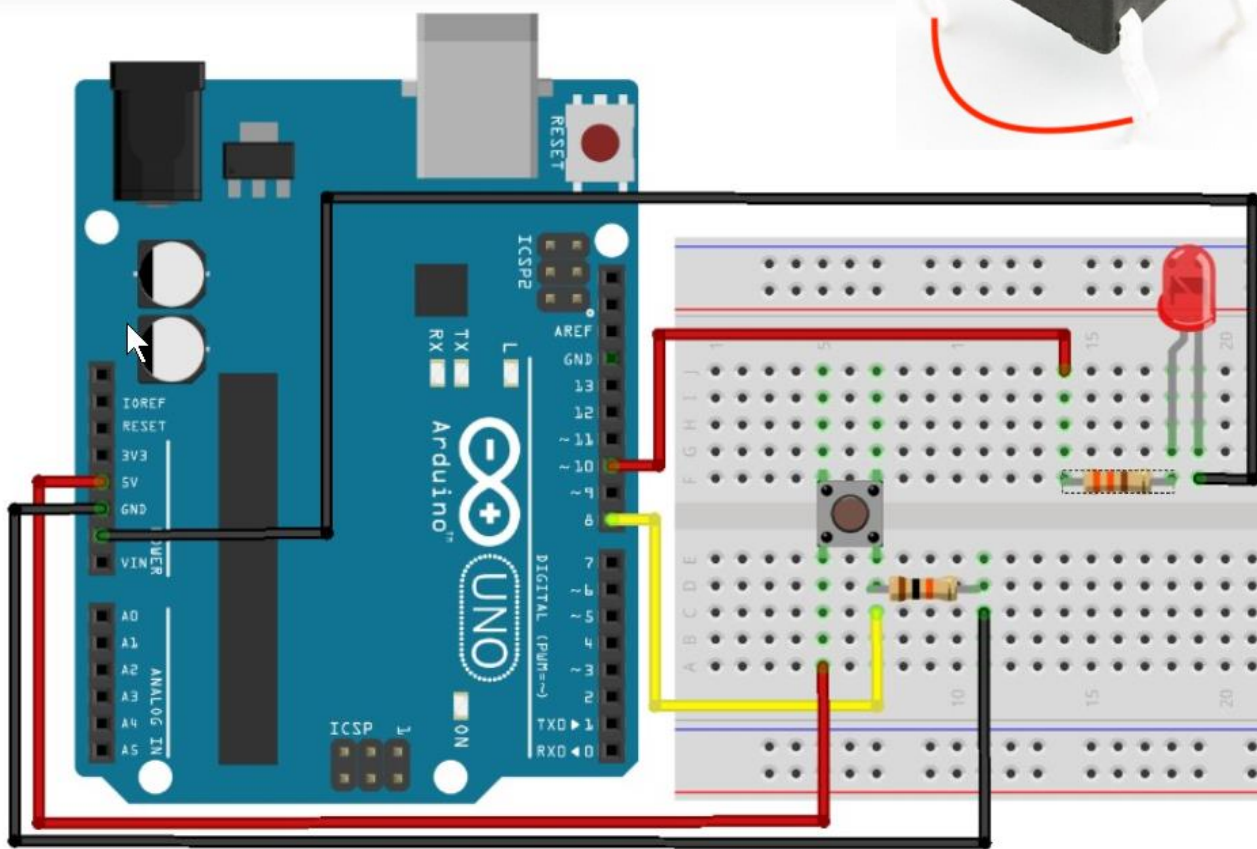
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

A watermark 'robotistan.com' is visible diagonally across the code. A 'MAKER robotistan.com' logo is in the bottom right corner of the code editor. A vertical scrollbar is on the right, and a horizontal scrollbar is at the bottom.

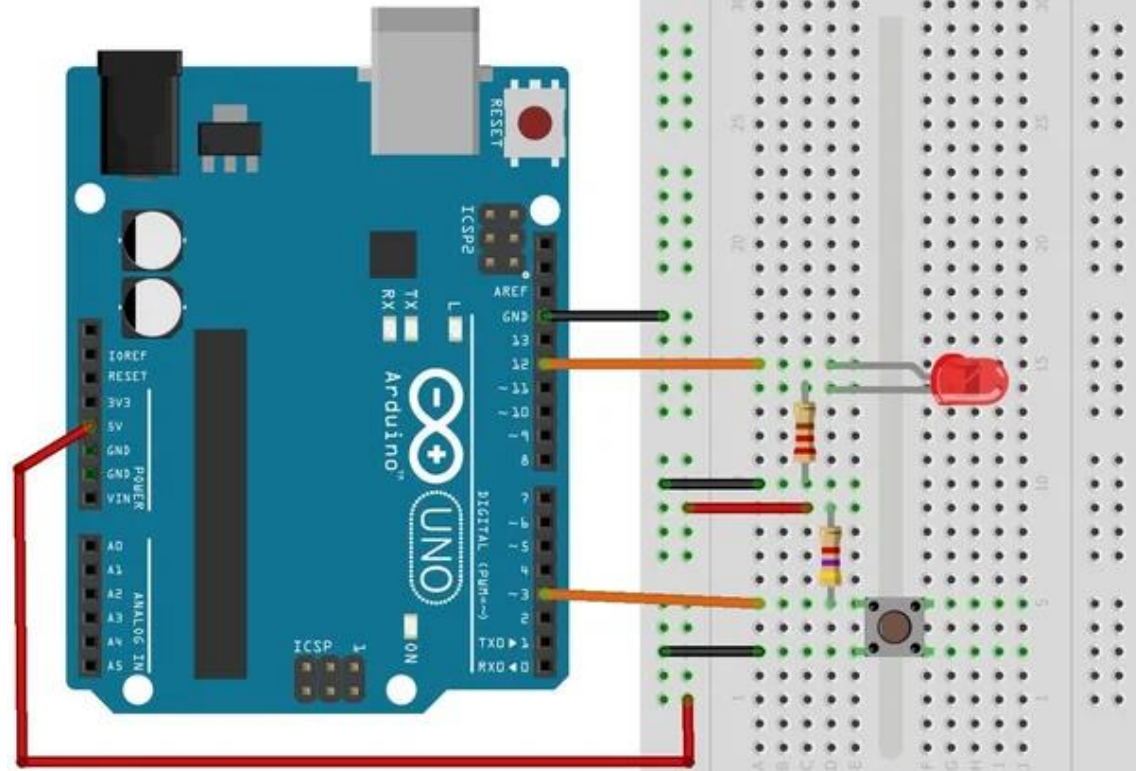
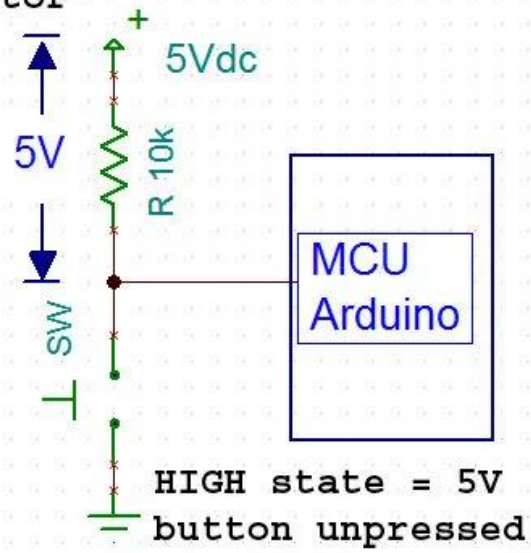
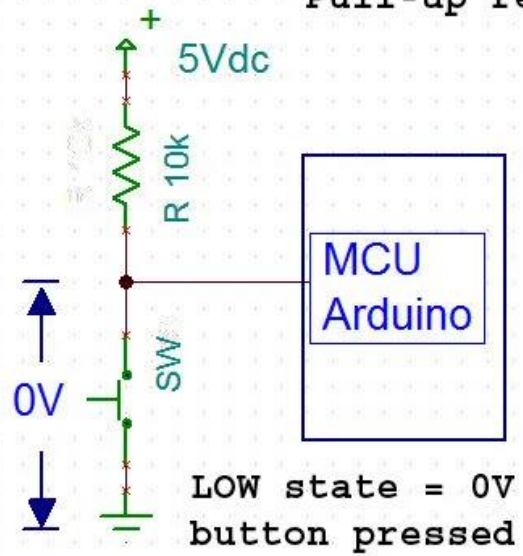
Malzeme Listesi

- Arduino UNO
- Breadboard
- 1 adet 330Ω direnç
- 1 adet 10kΩ direnç
- 1 adet push-buton
- İki ucu erkek jumper kablo

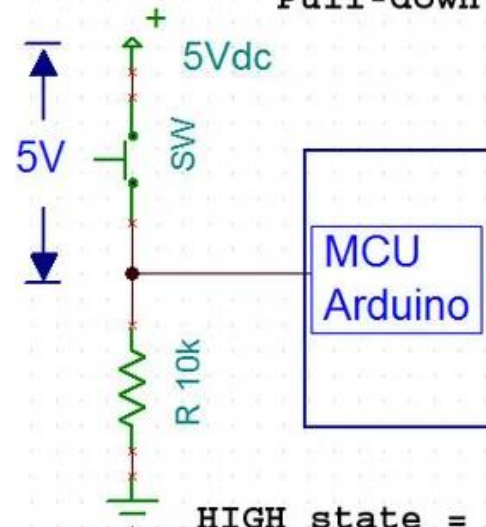
2. UYGULAMA Push Buton ile LED Yakma



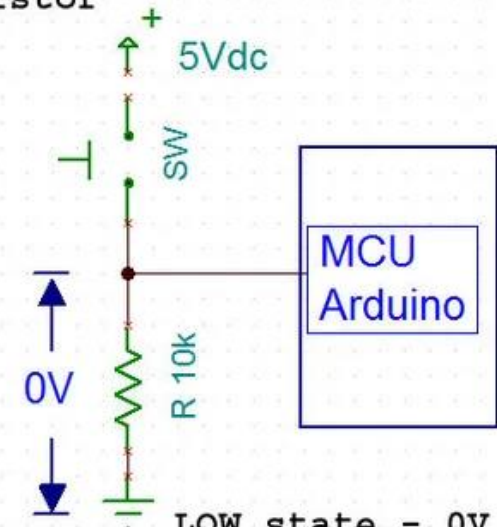
Pull-up resistor



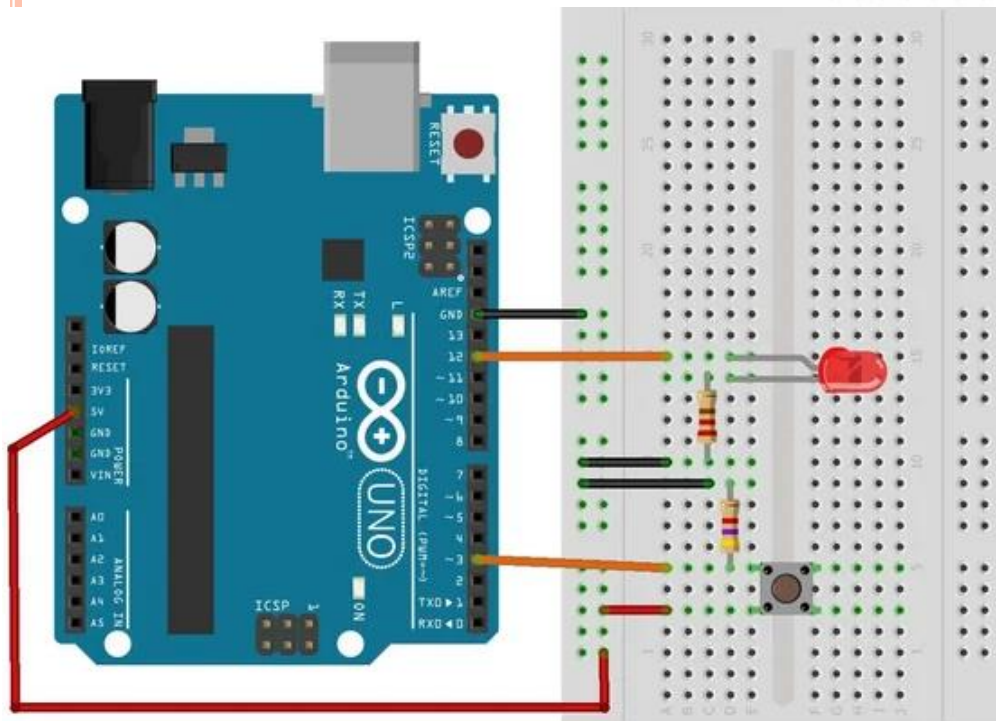
Pull-down Resistor



HIGH state = 5V
Button pressed



LOW state = 0V
Button unpressed



fritzing



Program Kodu

```
#define Buton 8
#define Led 10
void setup()
{
    pinMode(Buton, INPUT);
    pinMode(Led, OUTPUT);
}

void loop()
{
    if (digitalRead(Buton) == 1)
        digitalWrite(Led,HIGH);
    else
        digitalWrite(Led,LOW);
}
```

```
#define buton 8
#define led 9
int buton_durumu=0;
void setup() {
    pinMode(buton, INPUT);
    pinMode(led, OUTPUT);
}
void loop() {
    buton_durumu=digitalRead (buton);
    if (buton_durumu==1) {
        digitalWrite(led,HIGH);
    }
    else{
        digitalWrite(led,LOW);
    }
}
```

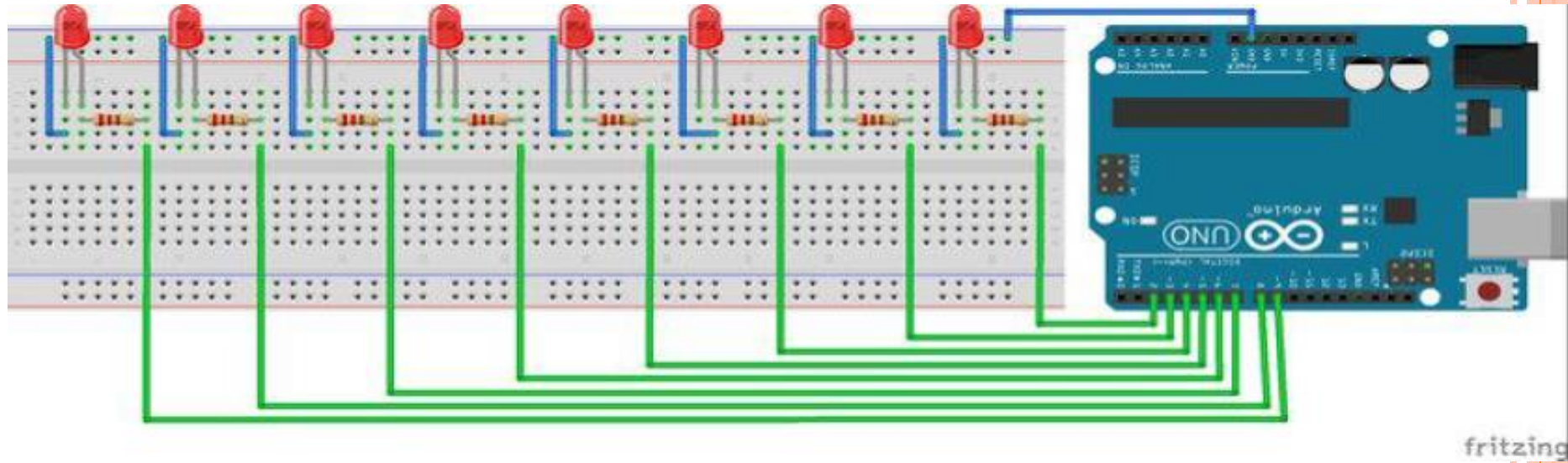
Arduino Data Types	Value Assigned	Value Ranges
boolean	8 Bit	True or False
byte	8 Bit	0 to 255
char	8 Bit	-127 to 128
unsigned char	8 Bit	0 to 255
word	16 Bit	0 to 65535
unsigned int	16 Bit	0 to 65535
int	16 Bit	-32768 to 32767
long	32 Bit	-2,147,483,648 to 2,147,483,647
float	32 Bit	-3.4028235E38 to 3.4028235E38
www.TheEngineeringProjects.com		



3. UYGULAMA

KARA ŞİMŞEK PROJESİ





Malzeme Listesi

- Arduino UNO
- Breadboard
- 8 adet 330Ω direnç
- 8 adet Led
- İki ucu erkek jumper kablo

```
const int LEDdizisi[] = {2,3,4,5,6,7,8,9};
```

void setup () { /* void setup kısmı arduino kartımızın temel kurulum işlemlerini yaptığımız yerdir yani ilk başta arduino kartımız başlarken hangi ayarları yapması gerekiyorsa o ayarları yaptığımız yerdir !!! void setup her çalışmada sadece 1 kez çalışır ve biter.*/

```
for(int i=0; i<8 ;i++)  
{ /* For dongusuyle LEDdizisi elemanlarına ulasiyoruz */  
  pinMode(LEDDizisi[i], OUTPUT); /* LED pinleri cikis olarak ayarlandi */  
}
```

**void loop() {/*
 void loop bölümü arduino kartımızın çalışıp void setup ayarlarını yaptıktan sonra SÜREKLİ OLARAK ÇALIŞTIĞI YERDİR WHILE(1) komutu ile AYNI MANTIKLA ÇALIŞIR.**

```
*/  
for(int i=0; i<8; i++){ /* Tum LEDleri sirayla 50 milisaniye yakip sonduruyoruz */  
  digitalWrite(LEDDizisi[i],HIGH);  
  delay(50);  
  digitalWrite(LEDDizisi[i],LOW);  
}
```

```
for(int j=7;j>-1; j--)  
{ /* LEDleri geri yonde 50 milisaniye yakip sonduruyoruz */  
  digitalWrite(LEDDizisi[j],HIGH);  
  delay(50);  
  digitalWrite(LEDDizisi[j], LOW);  
}  
}
```

```
const int LEDdizisi[] = {2,3,4,5,6,7,8,9};
```

int sonradan değiştirilebilir değerler için kullanılır. eğer **const int** tanımladığın değerler sonradan değişmez.

Arduino Diziler

Diziler, birden fazla bilgiyi ayrı ayrı hafızasında tutan değişken yapılarıdır. Örnek olarak 5 elemanlı bir dizi 5 adet farklı değişkeni hafızasında tutabilmektedir.

Meyveler Dizisi	
İndis No	Atanan Değer
0	Elma
1	Armut
2	Üzüm
3	Çilek

Normal bir değişken bir kutu gibidir ve içine bir tür malzeme koyulabilir. Diziler ise çok çekmeceli / raflı bir dolap gibidir. Her çekmecesine ayrı malzemeler yerleştirilebilir ve bu bilgiler birbirine karışmazlar.

```
char myChar = 'A';  
char myChar = 65; // both are equivalent  
Sözdizimi; char diziAdı[indis sayısı];  
char dizi[4]; // 4 elemanlı bir dizi oluşturduk
```

Yukarıdaki örnek kodda dizi adında 4 elemanlı bir dizi oluşturduk. “char” komutu ile dizi içerisinde tutacağımız veri türünü belirledik, ardından dizi ismi girdik ve köşeli parantezler. Yani bu dizi içerisine 4 adet değer atayabiliriz.

Eğer dizi oluştururken değer atayacaksa, dizi eleman sayısını girmemize gerek yoktur.

```
int dizi[] = {1,3,5,7,9} // 5 elemanlı dizi oluşturduk
```

Yukarıda oluşturduğumuz dizinin elemanlarına erişmek için ise dizi adını yazıp köşeli parantez içerisinde erişmek istediğimiz elemanın indis numarasını gireriz. İndis numaralarının **sıfırdan** başladığını unutmayalım.

```
1 int dizi[] = {1,3,5,7,9} // 5 elemanlı dizi oluşturduk
2
3 // seri porta dizinin sıfırıncı elemanı olan 1 sayısını yazmak için
4 Serial.println(dizi[0]); // seri porta 1 sayısını gönderir
5
6 // seri porta dizinin üçüncü elemanı olan 7 sayısını yazar
7 Serial.println(dizi[3]); // seri porta 7 sayısını gönderir
```

Eğer istersek dizi elemanlarını değişkene de atayabiliriz;

```
1 int dizi[] = {1,3,5,7,9};
2
3 int sayi = dizi[2]; // sayi değişkeni değeri 5 olur
```



Dizinin herhangi bir indisindeki elemanı değiştirmek istersek, yeni atayacağımız değer o indis numarasına atanır;

```
1 int dizi[] = {1,3,5,7,9}; // "dizi" adında 5 elemanlı bir dizi oluşturduk
2
3 // şimdi yukarıdaki dizimizin üçüncü indisindeki 7 değerini 6 yapalım
4 dizi[3] = 6; yukarıdaki dizimizdeki 7 değeri 6 oldu
```

Eğer bir dizinin tüm elemanlarını listelemek istersek bunu döngü ile yapabiliriz;

```
1 int dizi[] = {1,3,5,7,9};
2
3 for(int i = 0; i<5; i++){
4     Serial.println(dizi[i]);
5 }
```



Thank you for listening

