

ARDUINO PROGRAMLAMA



Yrd.Doç.Dr. Bülent ÇOBANOĞLU

Arduino Nedir? <http://arduino.cc/>

- Açık kaynaklı fiziksel programlama platformu
- Programlama dili : [*Processing / Wiring dili*](#)
- Arduino, açık kaynak kodlu yazılım ve donanıma sahip bir mikrodenetleyici platformudur. Açık kelimesi ile gerçek anlamda açık tasarımı ifade edilmektedir. Baskılı devresi, şematik tasarımı, pc üzerinde çalışan derleyicisi, kütüphaneleri ve tüm detayları ile internet ortamında paylaşılmaktadır.
- Arduino aynı zamanda mikro denetleyici cihazın adı olarak da kullanılmaktadır. Baş tasarımcılarının (Massimo Banzi ve David Cuartielles) İtalyan olmaları nedeniyle cihazın adı da doğal olarak İtalyancadan seçilmiş... Kelime “**Sıkı arkadaş**” anlamına gelen bir erkek ismi. Wikipedia kaynağına göre Arduino'ya ilham veren Wiring platformu, Ivrea Tasarım Enstitüsü'nde Hernando Barragan tarafından geliştirilmiş. Ivrea'lı Arduin ise bu enstitünün bulunduğu kasabaya ait tarihi bir karakterdir.

Processing & Wiring Dilleri

- **Processing;** Ben Fry ve Casey Reas tarafından 2001 de (ki her ikisi de John Maeda's öğrencileriydi, MIT Media Lab.de) geliştirildi.
- Grafiksel uygulamalar için kolay programlama ortamı oluşturur.
- <http://processing.org> Referans: <http://processing.org/reference/>
- **Wiring;** 2003 de Hernando Barragán tarafından başlatılan açık bir projedir. Gönüllü küçük bir ekip tarafından geliştirilmiştir.
- Wiring, tek bir MCU bordu + bir IDE + bir programlama dilinin birleşiminden oluşan açık kaynak kodlu elektronik prototip platformudur.
- <http://www.wiring.org.co/> Referans: <http://wiring.org.co/reference/>

Niçin Arduino?

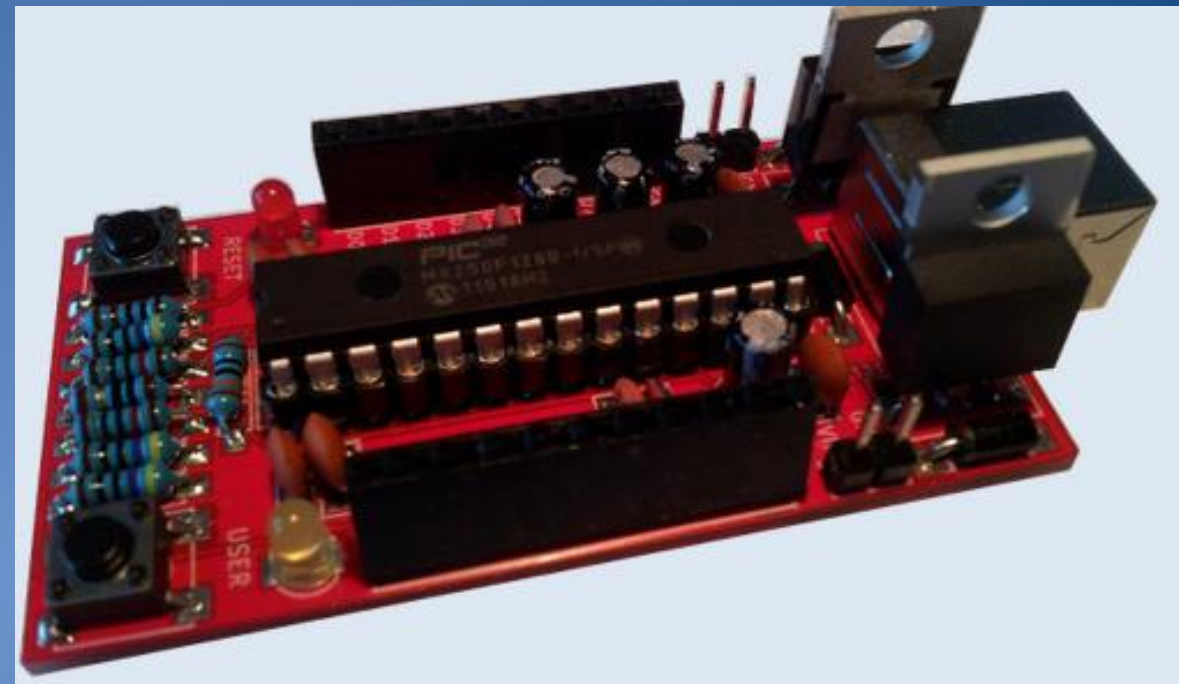
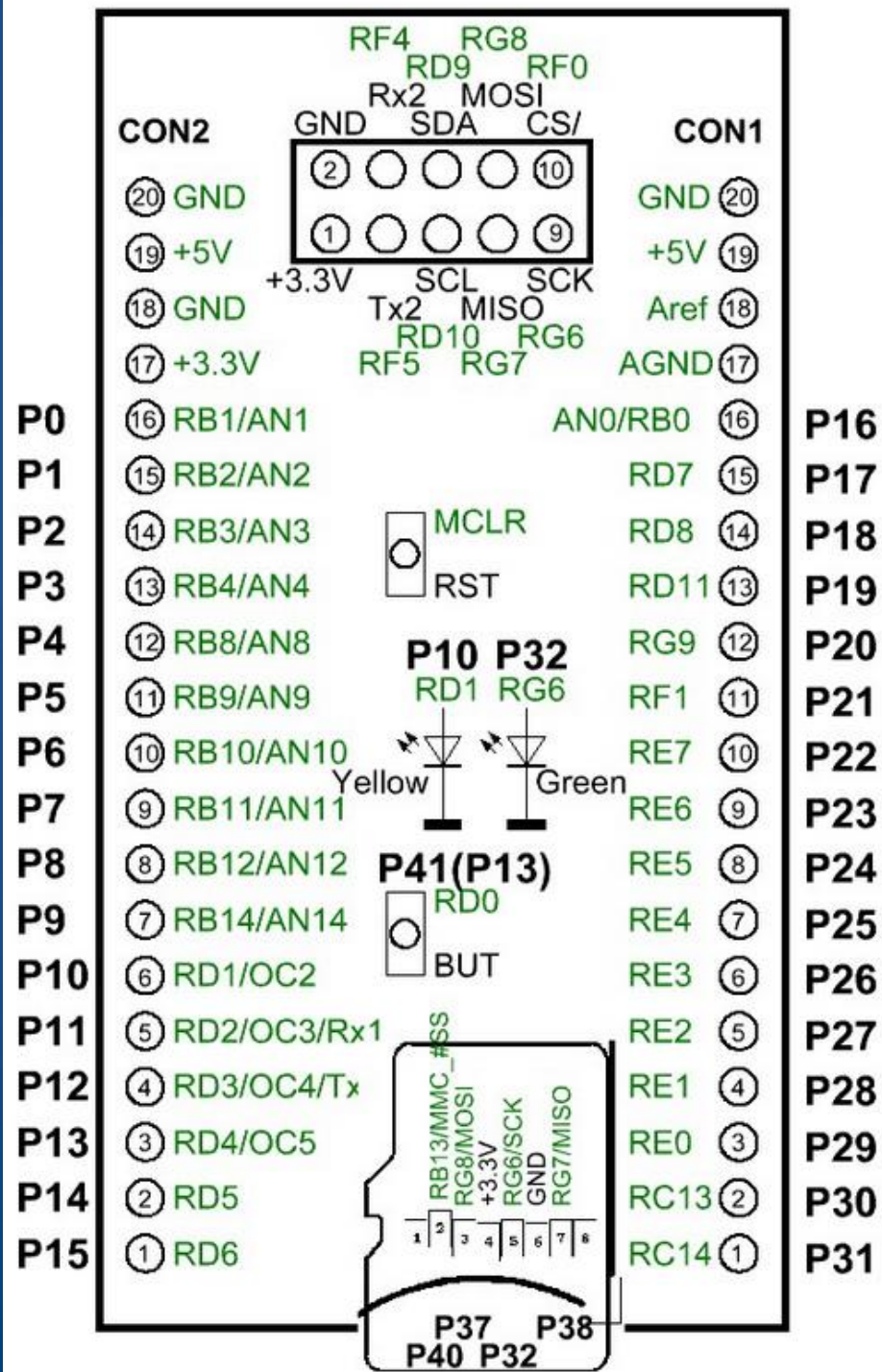
- Hem donanımı hem de yazılımı Açık kaynaklı,
- Ucuz
- Sadece USB veya Bluetooth ile iletişim
- İleri teknolojileri Boarda kolay entegre edebilirsiniz

PINGUNIO (<http://www.pinguino.cc/>)

Pinguino 32 Micro 30 mar 2012

P33P35P37P39

P34P36P38 P32



Arduino Boardları

- Fiziksel Giriş/Çıkış (I/O) bordu ile Programlanabilir Bütünleşik Devre (IC).



Arduino Uno



Arduino Leonardo



Arduino Due



Arduino Yun



LilyPad Arduino SimpleSnap



LilyPad Arduino



Arduino Robot



Arduino Esplora



Arduino Mega ADK



Arduino Ethernet

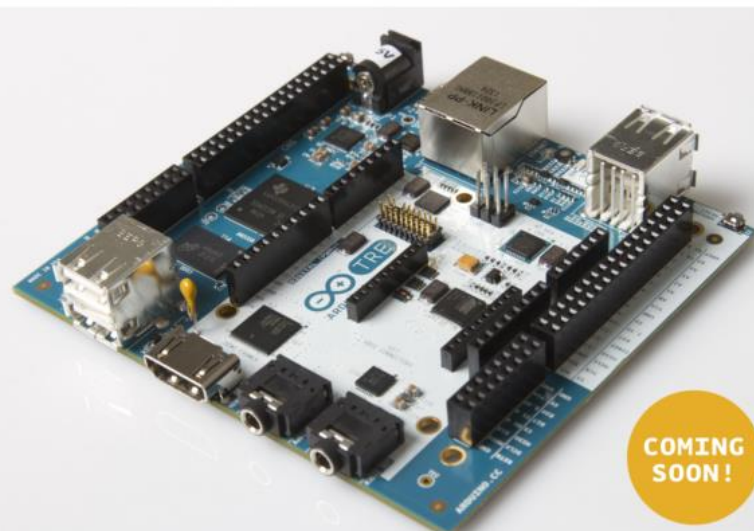


Arduino Nano



Arduino Pro Mini

Arduino Tre



Arduino Mega 2560



Arduino Micro



Arduino Pro



Arduino Fio



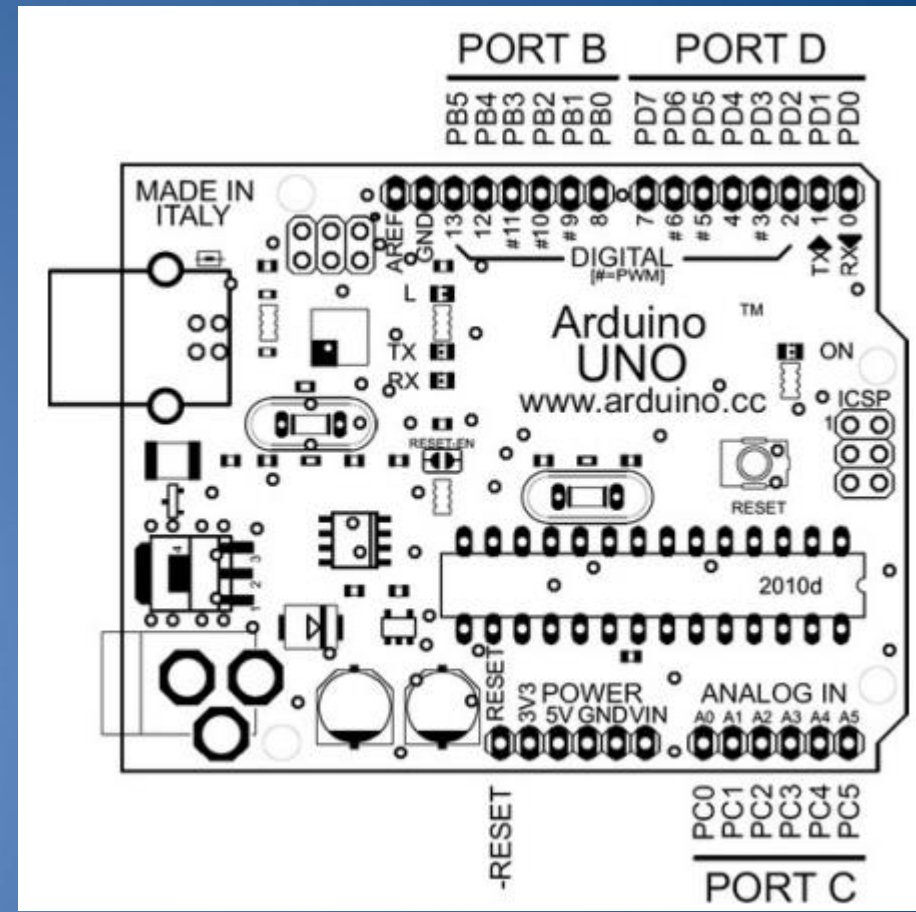
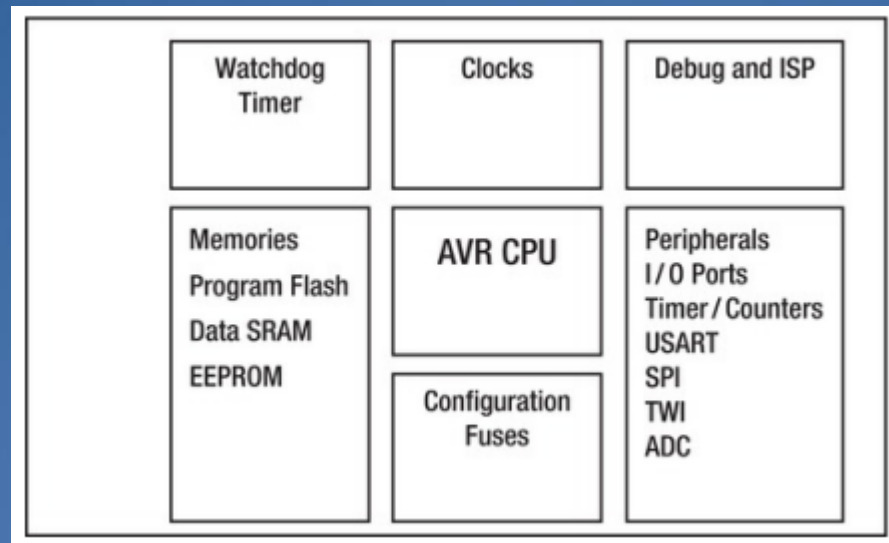
LilyPad Arduino USB



LilyPad Arduino Simple

Arduino Duo Donanimsal Yapisi

ATMEGA 328



Atmega168 Pin Mapping

Arduino function	ATmega168 Pin	ATmega168 Pin	Arduino function		
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM
ATmega48PA	4K Bytes	256 Bytes	512 Bytes
ATmega88PA	8K Bytes	512 Bytes	1K Bytes
ATmega168PA	16K Bytes	512 Bytes	1K Bytes
ATmega328P	32K Bytes	1K Bytes	2K Bytes

Table 3-9. Program Memory Page Size

Device	Page Size	Pages
ATmega328	64 words	256
ATmega2560	128 words	1,024

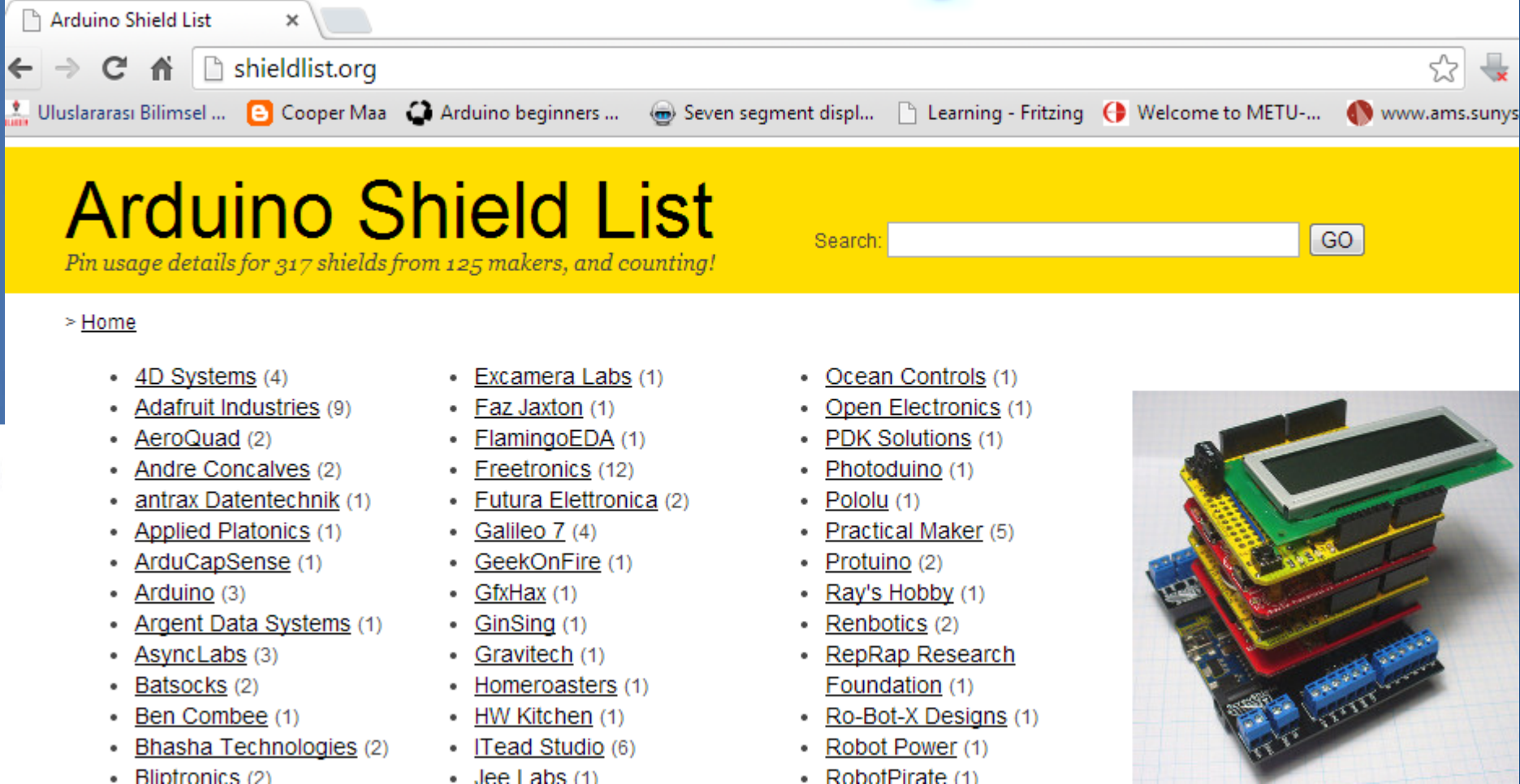
Arduino Duo Donanimals Yapısı

Pin Number		I/O Pin	Function	Description	Pin Number		I/O Pin	Function	Description
DIP-28	TQFP-32				DIP-28	TQFP-32			
14	12	PB0	D8	Digital pin 8	19	17	PB5	D13	Digital pin 13 (LED)
			CLKO	Divided system clock output				SCK	SPI bus master clock input (ISP)
			ICP1	Timer/counter 1 input capture input				PCINT5	Pin-change interrupt 5
			PCINT0	Pin-change interrupt 0					
15	13	PB1	D9	Digital pin 9 (PWM capable)	9	7	PB6	XTAL1	Quartz crystal or ceramic resonator input, external clock input
			OC1A	Timer/counter 1 output compare match A output				TOSC1	Timer/counter 2 oscillator input
			PCINT1	Pin-change interrupt 1				PCINT6	Pin-change interrupt 6
16	14	PB2	D10	Digital pin 10 (PWM capable)	10	8	PB7	XTAL2	Quartz crystal or ceramic resonator output
			OC1B	Timer/counter 1 output compare match B output				TOSC1	Timer/counter 2 oscillator output
			-SS	SPI bus master/slave select				PCINT7	Pin-change interrupt 7
			PCINT2	Pin-change interrupt 2					
17	15	PB3	D11	Digital pin 11 (PWM capable)	23	23	PC0	A0	ADC0, analog input 0
			OC2A	Timer/counter 2 output compare match A output				PCINT8	Pin-change interrupt 8
			MOSI	SPI bus master output, slave input (ISP)					
			PCINT3	Pin-change interrupt 3					
18	16	PB4	D12	Digital pin 12	24	24	PC1	A1	ADC1, analog input 1
			MISO	SPI bus master input, slave output (ISP)				PCINT9	Pin-change interrupt 9
			PCINT4	Pin-change interrupt 4					
					25	25	PC2	A2	ADC2, analog input 2
								PCINT10	Pin-change interrupt 10
					26	26	PC3	A3	ADC3, analog input 3
								PCINT11	Pin-change interrupt 11

Pin Number		I/O Pin	Function	Description
DIP-28	TQFP-32			
27	27	PC4	A4	ADC4, analog input 4
			SDA	I ² C/TWI serial bus data input/output line
			PCINT12	Pin-change interrupt 12
28	28	PC5	A5	ADC5, analog input 5
			SCL	I ² C/TWI serial bus clock line
			PCINT13	Pin-change interrupt 13
1	29	PC6	-RESET	Reset input, active low
			PCINT14	Pin-change interrupt 14
2	30	PD0	D0	Digital pin 0
			RXD	USART serial input
			PCINT16	Pin-change interrupt 16
3	31	PD1	D1	Digital pin 1
			TXD	USART serial output
			PCINT17	Pin-change interrupt 17
4	32	PD2	D2	Digital pin 2
			INT0	External interrupt 0
			PCINT18	Pin-change interrupt 18
5	1	PD3	D3	Digital pin 3 (PWM capable)
			OC2B	Timer/counter 2 output compare match B output
			INT1	External interrupt 1
			PCINT19	Pin-change interrupt 19
Pin Number		I/O Pin	Function	Description
DIP-28	TQFP-32			
6	2	PD4	D4	Digital pin 4
			XCK	USART external clock
			T0	Timer/counter 0 external counter input
			PCINT20	Pin-change interrupt 20
11	9	PD5	D5	Digital pin 5 (PWM capable)
			OC0B	Timer/counter 0 output compare match B output
			T1	Timer/counter 1 external counter input
			PCINT21	Pin-change interrupt 21
12	10	PD6	D6	Digital pin 6 (PWM capable)
			OC0A	Timer/counter 0 output compare match A output
			AIN0	Analog comparator input 0 (positive)
			PCINT22	Pin-change interrupt 22
13	11	PD7	D7	Digital pin 7
			AIN1	Analog comparator input 1 (negative)
			PCINT23	Pin-change interrupt 23

Shield- <http://arduino.cc/en/Main/ArduinoShields>

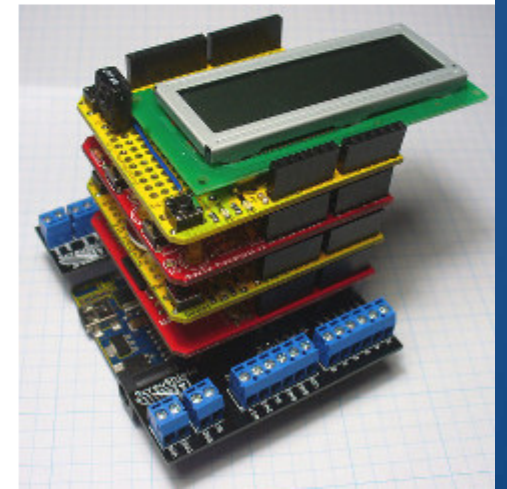
Shield: Arduino boardu üzerindeki devre eklentileridir.



The screenshot shows the website 'shieldlist.org' in a browser window. The page title is 'Arduino Shield List' and the subtitle is 'Pin usage details for 317 shields from 125 makers, and counting!'. There is a search bar with a 'GO' button. Below the search bar, there is a list of makers and their respective shield counts, organized in three columns.

> [Home](#)

- [4D Systems](#) (4)
- [Adafruit Industries](#) (9)
- [AeroQuad](#) (2)
- [Andre Concalves](#) (2)
- [antrax Datentechnik](#) (1)
- [Applied Platonics](#) (1)
- [ArduCapSense](#) (1)
- [Arduino](#) (3)
- [Argent Data Systems](#) (1)
- [AsyncLabs](#) (3)
- [Batsocks](#) (2)
- [Ben Combee](#) (1)
- [Bhasha Technologies](#) (2)
- [Blintronics](#) (2)
- [Excamera Labs](#) (1)
- [Faz Jaxton](#) (1)
- [FlamingoEDA](#) (1)
- [Freetronics](#) (12)
- [Futura Elettronica](#) (2)
- [Galileo 7](#) (4)
- [GeekOnFire](#) (1)
- [GfxHax](#) (1)
- [GinSing](#) (1)
- [Gravitech](#) (1)
- [Homeroasters](#) (1)
- [HW Kitchen](#) (1)
- [ITead Studio](#) (6)
- [Jee Labs](#) (1)
- [Ocean Controls](#) (1)
- [Open Electronics](#) (1)
- [PDK Solutions](#) (1)
- [Photoduino](#) (1)
- [Pololu](#) (1)
- [Practical Maker](#) (5)
- [Protduino](#) (2)
- [Ray's Hobby](#) (1)
- [Renbotics](#) (2)
- [RepRap Research Foundation](#) (1)
- [Ro-Bot-X Designs](#) (1)
- [Robot Power](#) (1)
- [RobotPirate](#) (1)





Farklı Platformlar

- Arduino- Labview Platformu

- <https://decibel.ni.com/content/groups/labview-interface-for-arduino>

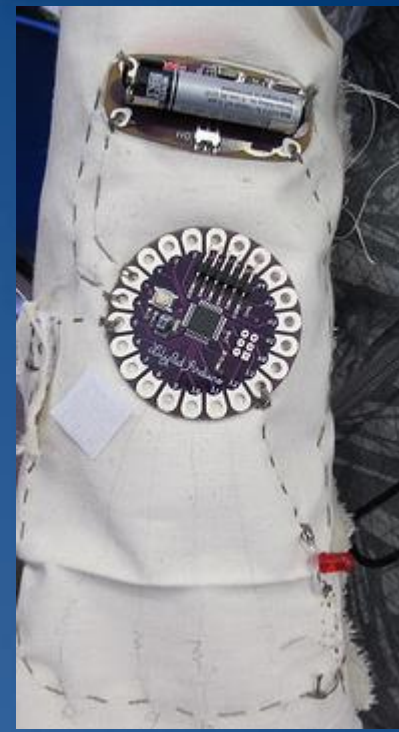
- <http://www.robishop.com/ArduinoLabVIEW-Bundle,PR-2855.html>

- Arduino- MATLAB ve Simulink

- <http://www.mathworks.com/hardware-support/arduino-matlab.html>

- WBAN ve Giyilebilir Teknolojiler

- <http://rainycatz.wordpress.com/2012/04/07/wearable-technology-bootcamp-with-lilypad-arduino-technocamps-aberystwyth/>



Arduino uygulamalarını Proteus ile Simüle Etmek

- <http://www.youspice.com/ys/gettingstartedwithproteus.3sp>
- <http://www.thinkcreate.org/index.php/debug-arduino-with-proteus/>
- <http://www.youtube.com/watch?v=sXdnvPtTZ7A>

Arduino Donanımlarını Nereden Temin Edebilirim?

- <http://arduino.cc/en/Main/Buy>
- <http://dx.com/s/Arduino>
- <http://www.ebay.com>
- <http://www.robotistan.com/> (Türkçe)
- <http://www.robotshop.com/arduino-2.html>
-

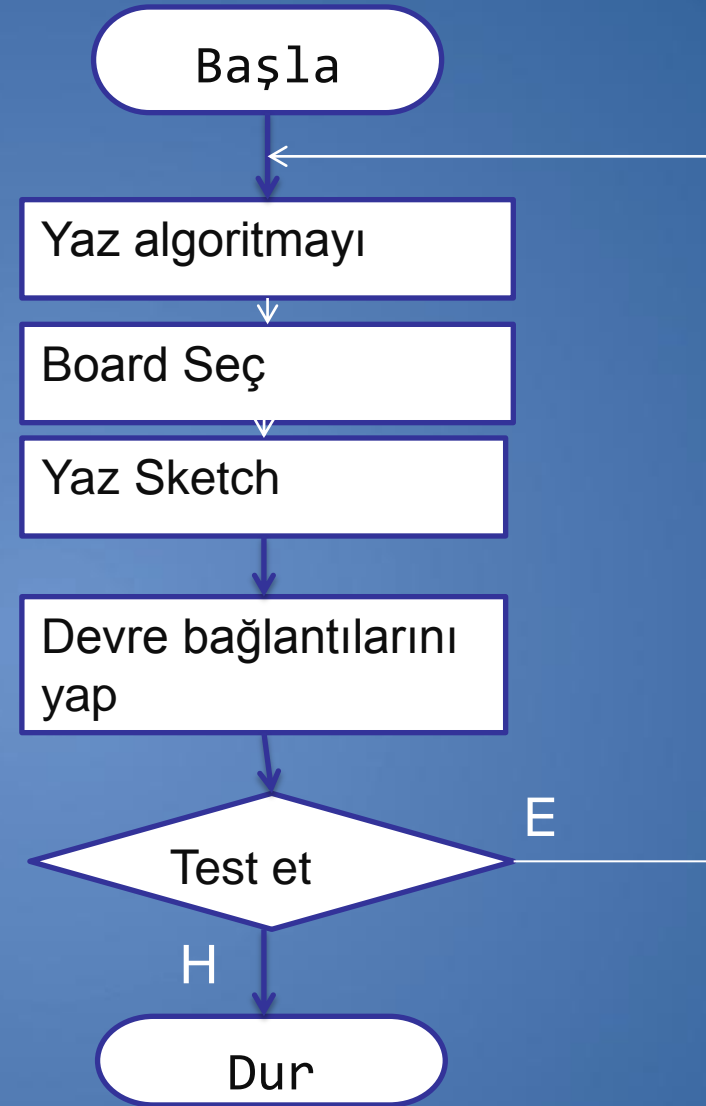
Arduino ile Nasıl Bir Proje/Ödev Yapabilirim?

1. Medikal Uygulamalar
2. Robotik Uygulamalar - Make an Arduino-Controlled Robot by Michael Margolis
 - İnsansı Robotlar - Humanoid Robots
3. Mekatronik Uygulamalar
4. Mobil Uygulamalar
5. Giyilebilir Uygulamalar
6. Kablosuz Haberleşme Uygulamaları - Building Wireless Sensor Networks by Robert Faludi
7. Algoritmik Uygulamalar
8. RFID Uygulaması - Getting Started with RFID by Tom Igoe
9. Ev Otomasyonu - Programming Your Home Automate with Arduino, Android, and Your Computer - Mike Riley

Arduino Referans Kaynakları

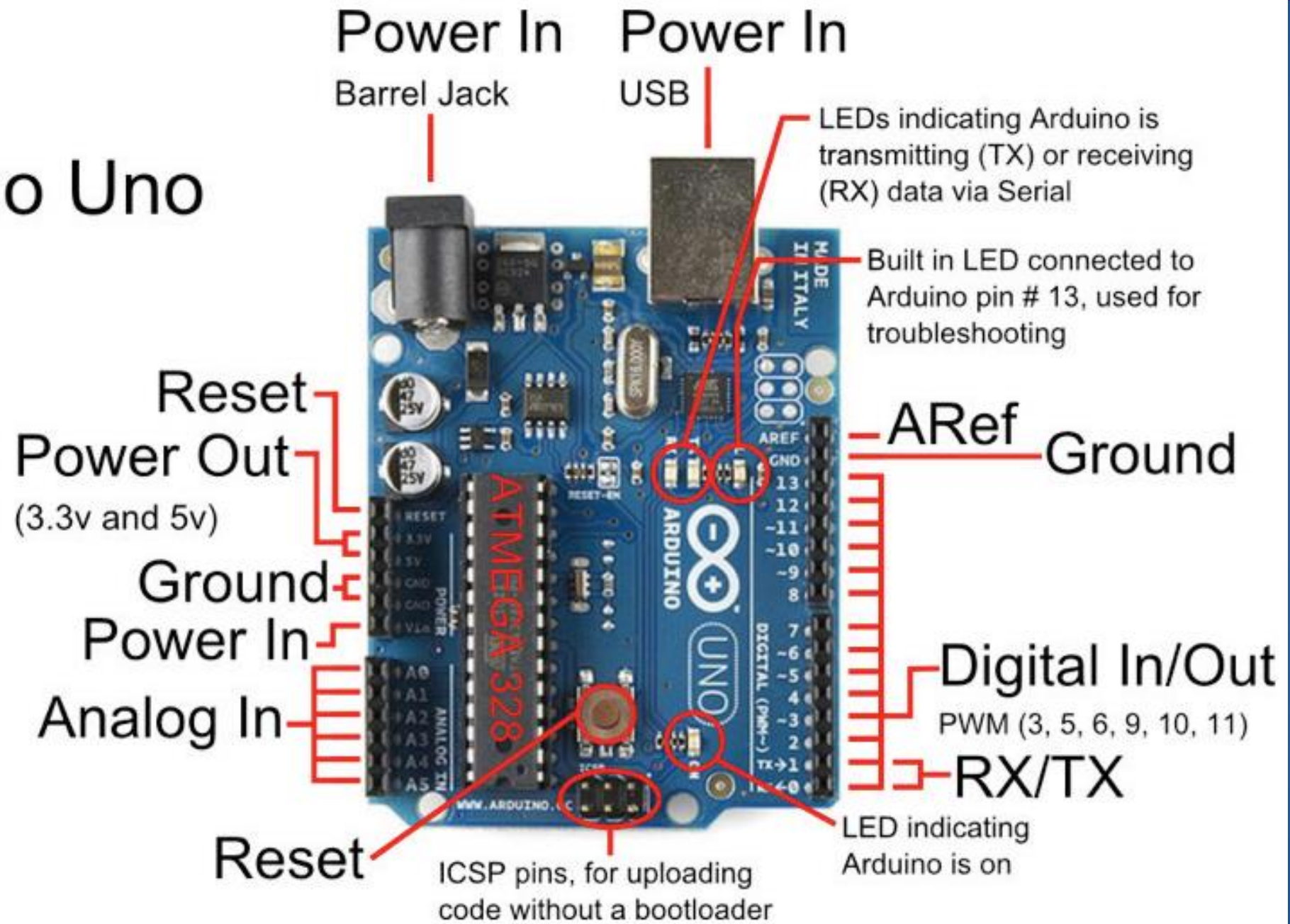
- <http://blog.robomore.com/?cat=6&paged=2>
- https://labitat.dk/wiki/Arduino_beginners_workshop
- <http://coopermaa2nd.blogspot.com/>
- <http://arduino.cc/en/Tutorial/HomePage>
- <http://shieldlist.org/>
- Fritzing- Çizim programı: <http://fritzing.org/download/>

Arduino Proje Geliştirme Adımları

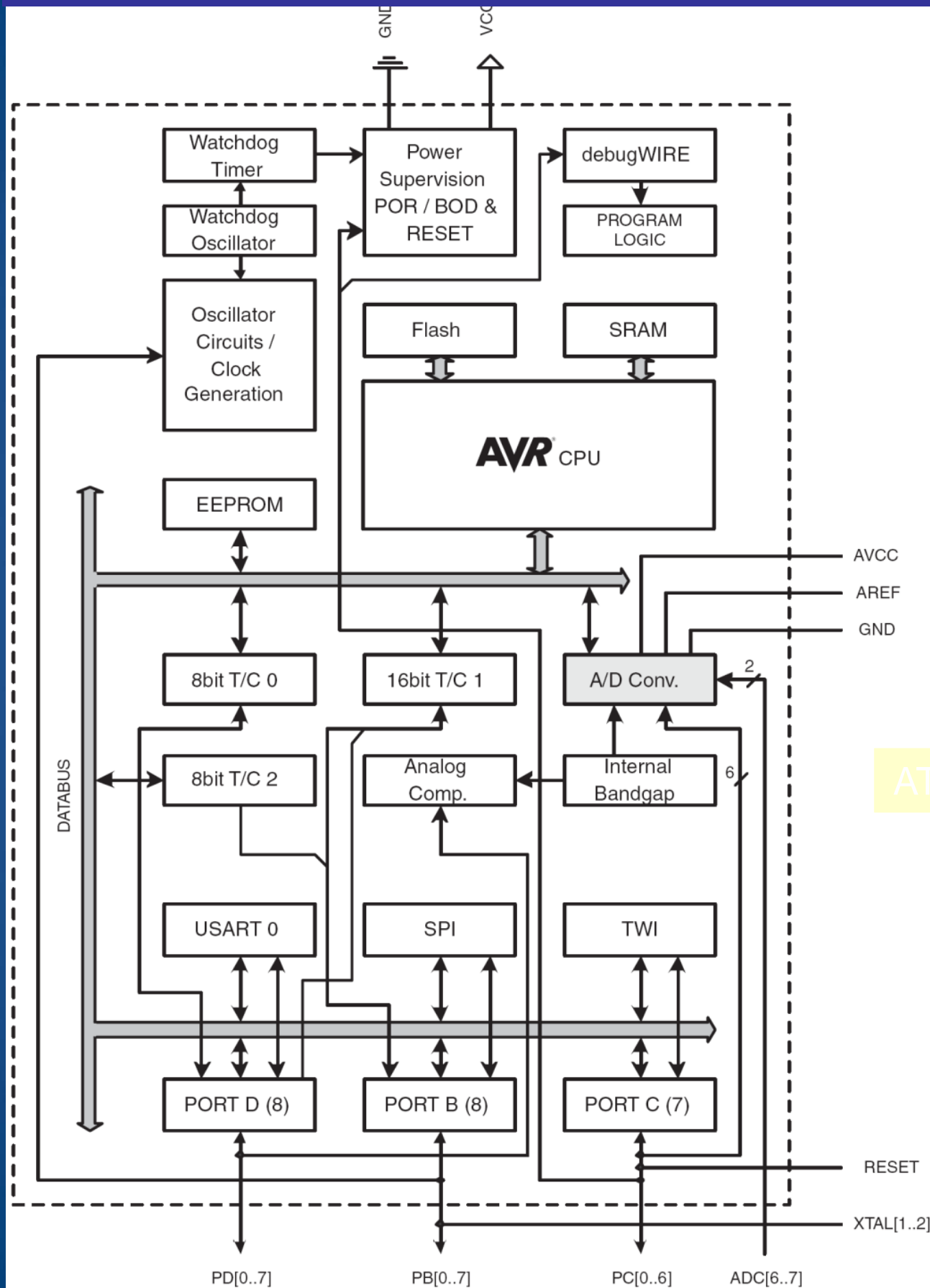


Arduino Uno Bağlantı Noktaları

Arduino Uno

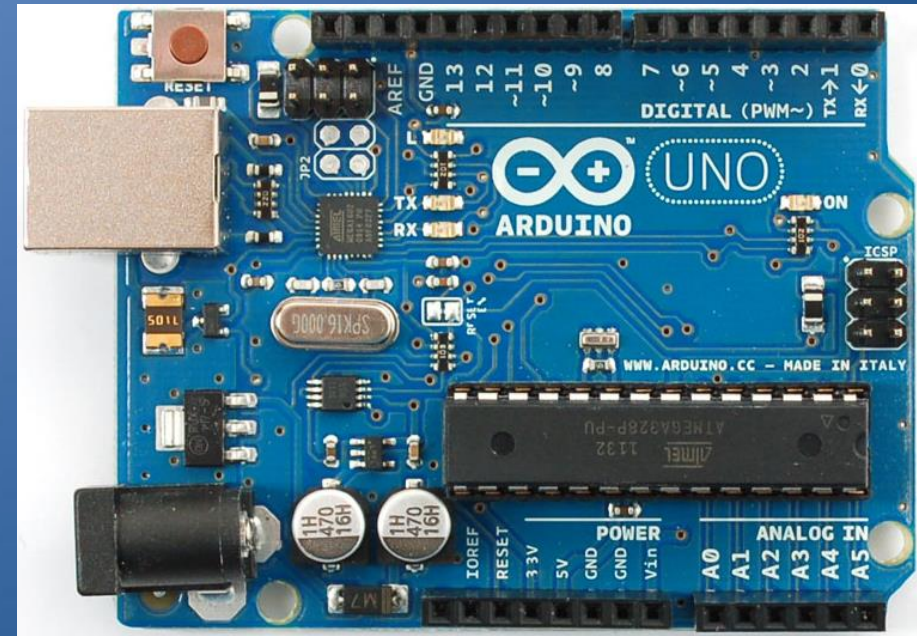


ATmega328 İç Mimarisi



(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

ATmega328 data sheet pp. 2, 5

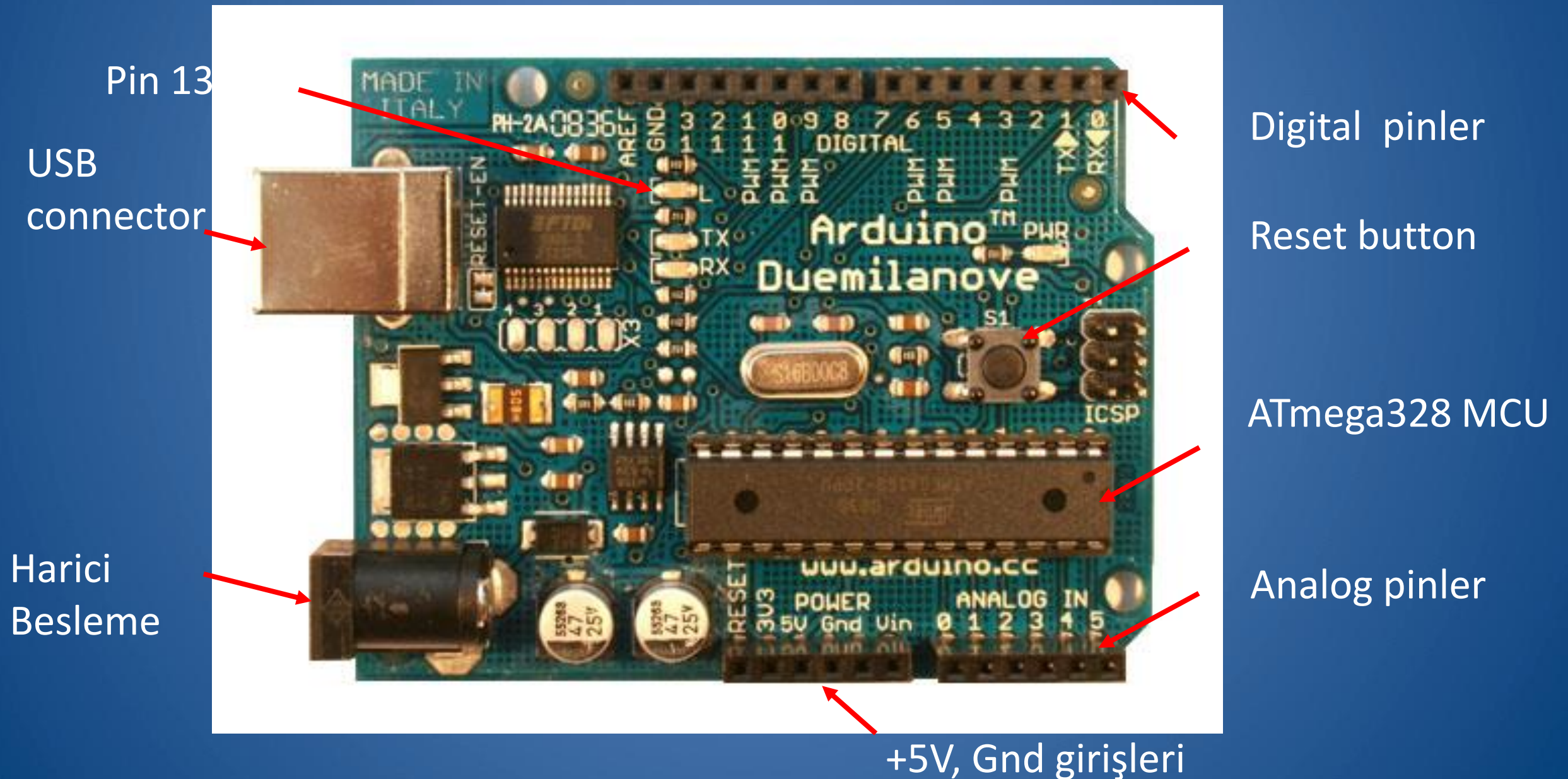


Arduino Duemilanove/Uno Özellikleri

Microcontroller	ATmega168/328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Programlanabilme Kapasitesi	10000 Flash, 100.000 EEPROM

Arduino Duemilanove

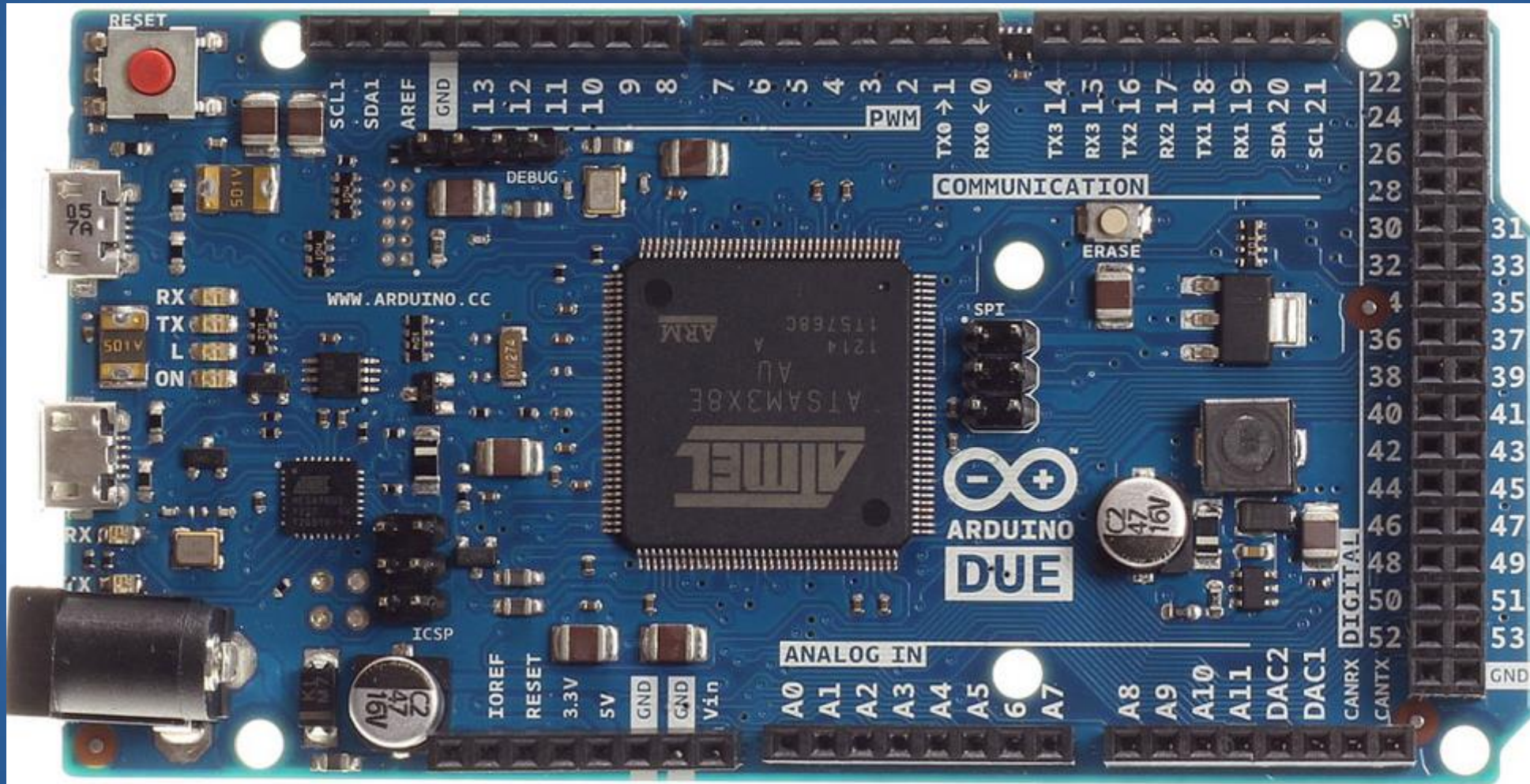
<http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>



<http://arduino.cc/en/uploads/Main/ArduinoDuemilanove.jpg>

Arduino Due

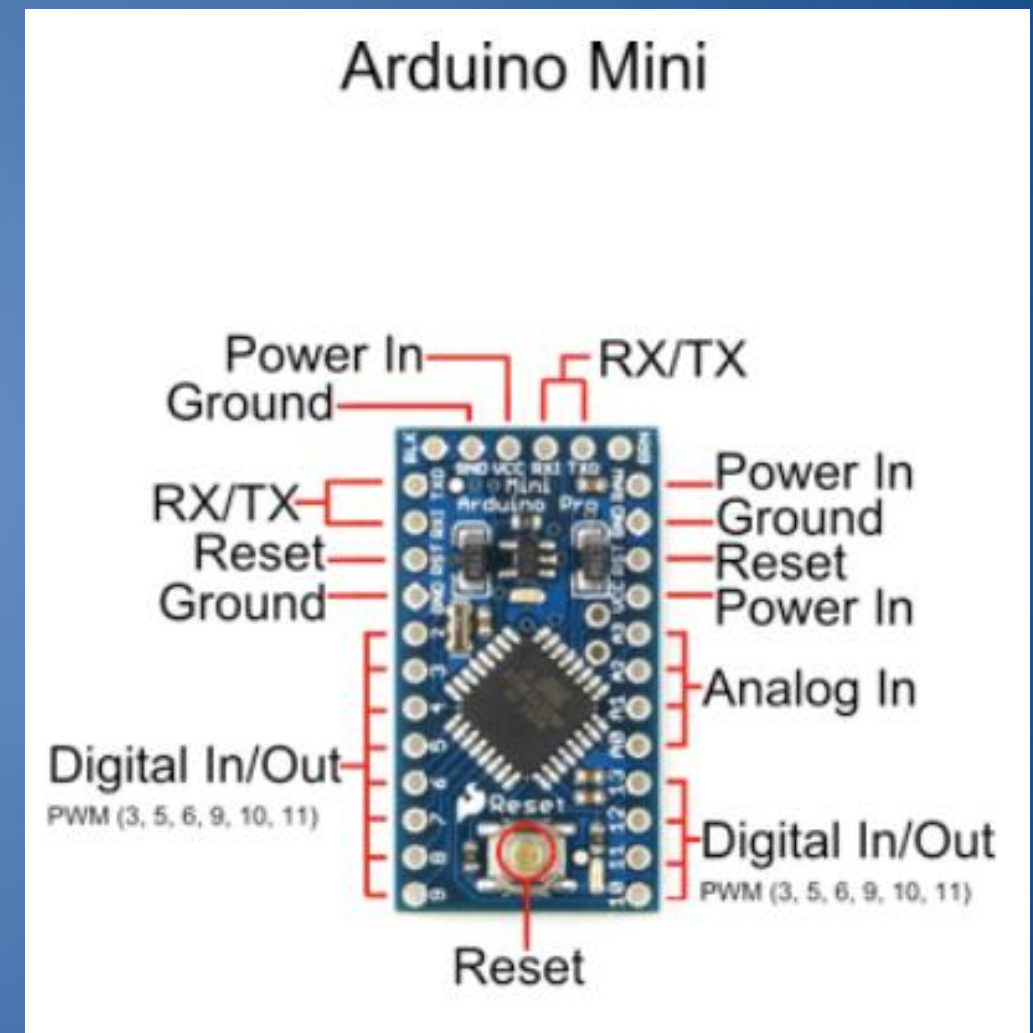
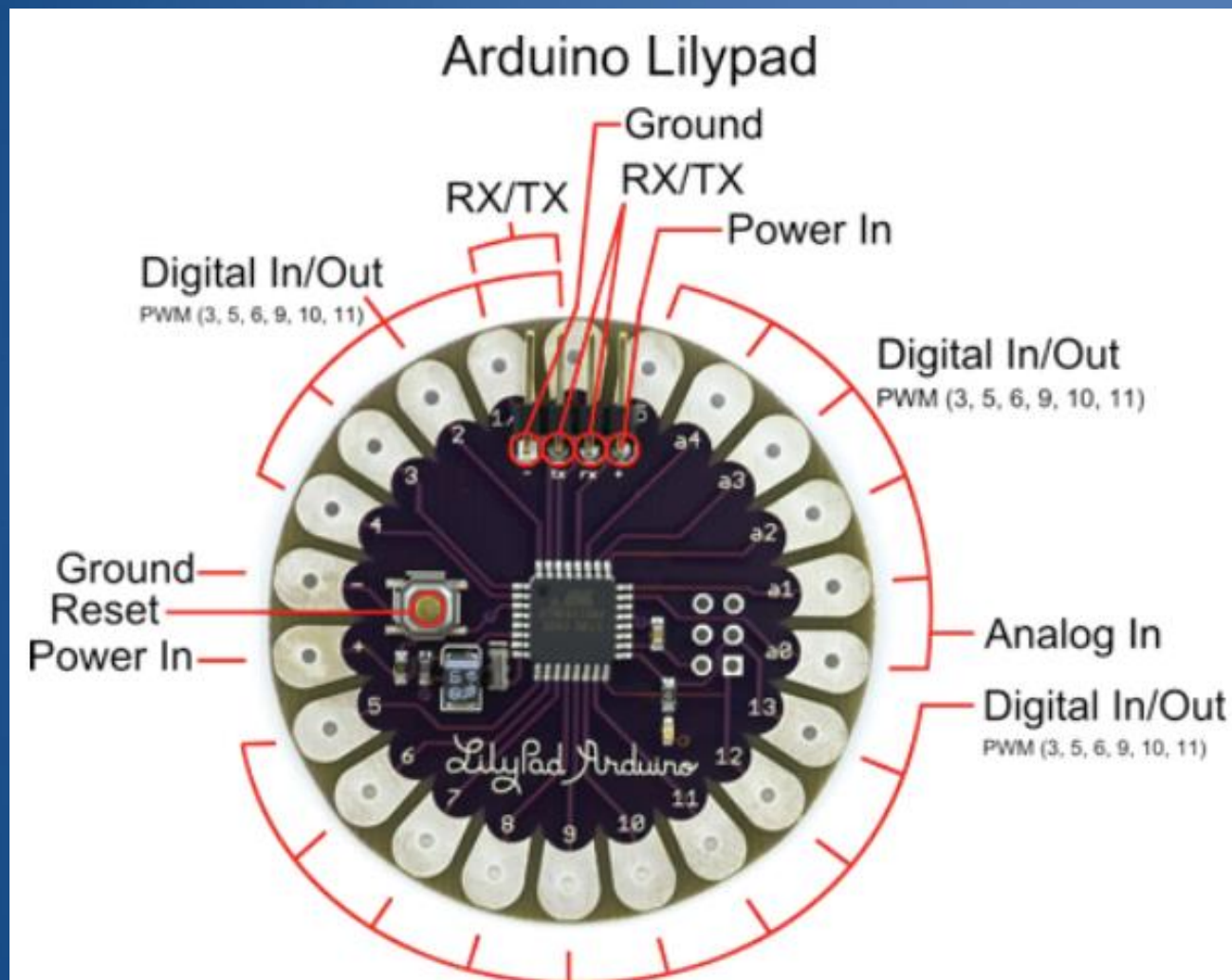
Atmel SAM3X8E processor (32 bit ARM Cortex M3 architecture, 84MHz)



http://www.adafruit.com/index.php?main_page=popup_image&pID=1076

<http://arduino.cc/en/Main/ArduinoBoardDue>

Arduino Lilypad – Arduino Mini

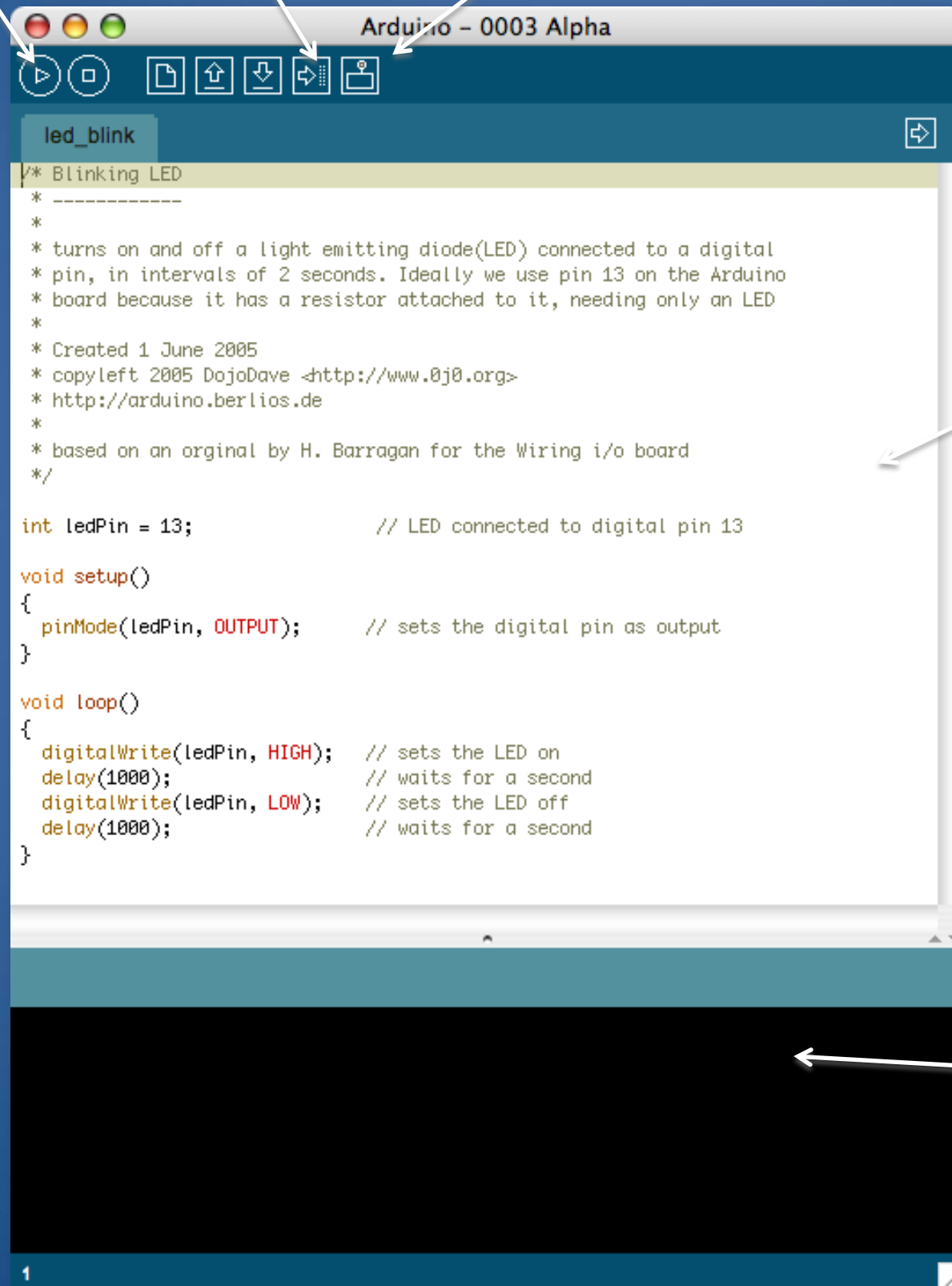


Compile

Upload to controller

Serial Monitor

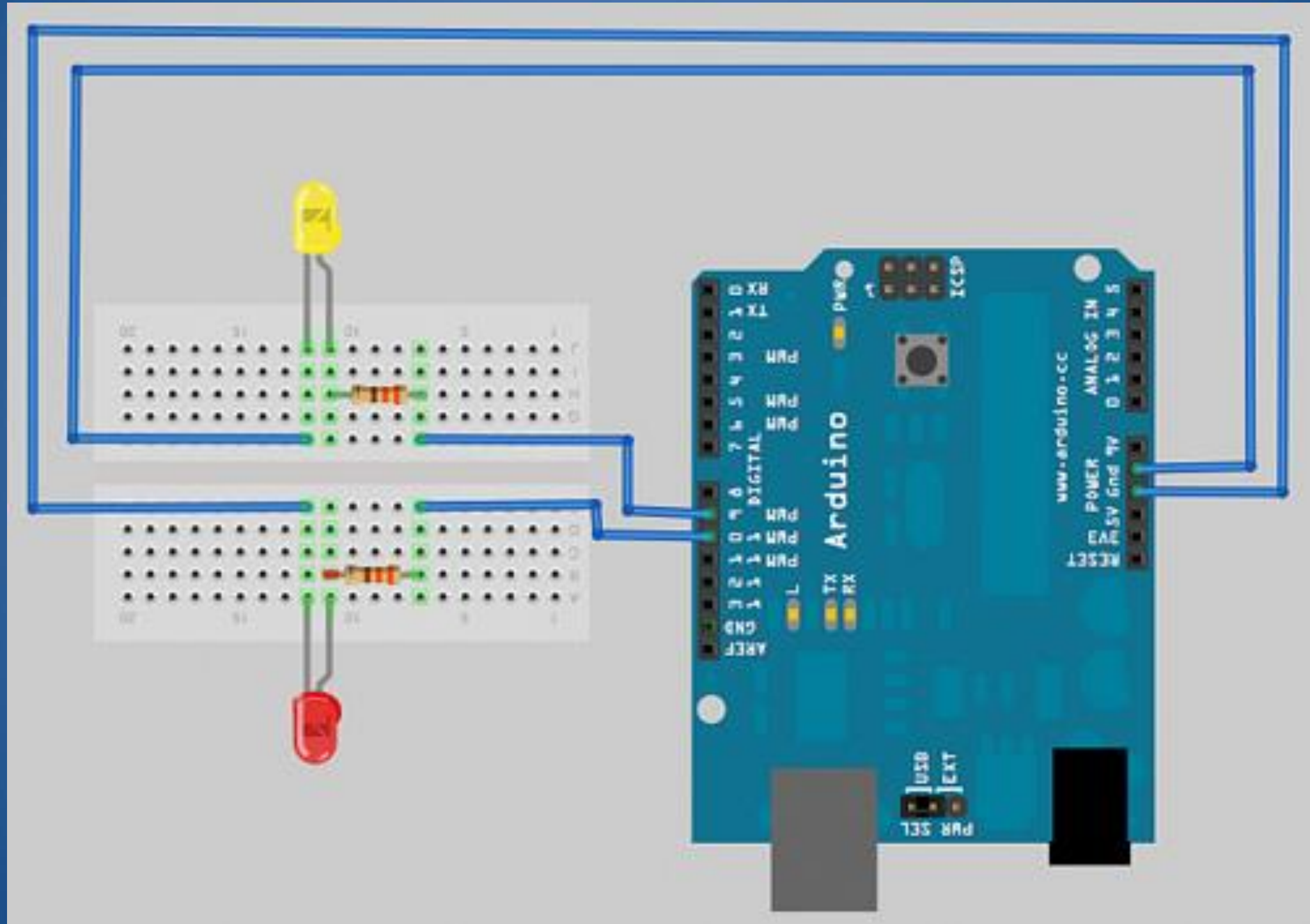
Arduino IDE Window



Code Editor

Text Output (Serial Data)

Donanımsal Bağlantı



Arduino Programlama Temelleri

- Arduino programları iki temel fonksiyon ile çalışır

```
void setup() {  
  
    //led pinleri, motorlar, sensorler, vs..  
  
}  
void loop() {  
  
    // sensorlerden bilgiyi al  
    // motorları kontrol et, veya ilgili ledi yak/söndür  
  
}
```

SETUP Fonksiyonu

```
void setup() {
```

```
  pinMode(9, OUTPUT);
```

```
}
```

port ismi

Giriş / Çıkış
Yönlendirmesi

pinMode komutu ile bir pini Giriş olarak yönlendirmek için INPUT, çıkış olarak yönlendirmek için ise OUTPUT deyimini kullanılır.

LOOP Fonksiyonu

```
void loop()
```

```
{
```

```
    digitalWrite(9, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(9, LOW);
```

```
    delay(1000);
```

```
}
```

1 sn bekle



LED yak, söndür



Giriş/Çıkış Pin Yönlendirme

- 0 ve 1 (PD0 and PD1). Pinleri giriş, ve bu pinlere değer atama

- Arduino yaklaşımı

```
pinMode(0, INPUT);  
pinMode(1, INPUT);  
digitalWrite(0, HIGH);  
digitalWrite(1, HIGH);
```

- 3, 5, ve 7 (PD3, PD5, and PD7) pinleri çıkış olarak ayarla

```
pinMode(3, OUTPUT);  
pinMode(5, OUTPUT);  
pinMode(7, OUTPUT);
```

Sayı sistemi tanımlamaları

- `int decimal=4711;`
- `int binary=B1001001100111;`
- `int octal=011147;`
- `int hexadecimal=0x1267;`
- **Ekrana farklı formatları yazdırma;**
 - `Serial.println(degisken, DEC);`
 - `Serial.println(degisken, HEX);`
 - `Serial.println(degisken, OCT);`
 - `Serial.println(degisken, BIN);`
 - `Serial.println(degisken, BYTE);`

Veri Tipleri

Table 6-2. Available Data Types and Their Ranges

Data Type	Number of Bits	Number of Bytes	Minimum Value	Maximum Value
unsigned char byte	8	1	0	255
signed char	8	1	-128	127
unsigned int word	16	2	0	65,535
int	16	2	-32,768	32,767
unsigned long	32	4	0	4,294,967,296
long	32	4	-2,147,483,648	2,147,483,647
unsigned long long	64	8	0	18,446,744,073,709,551,616
long long	64	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
float	32	4	-3.4028235E+38	3.4028235E+38
double	32	4	-3.4028235E+38	3.4028235E+38

Değişken Tanımlama

```
int val = 5;
```

Veri Tipi

Değişken adı

Atama
operatörü

Atanan değer

Gecikme –delay()

Time

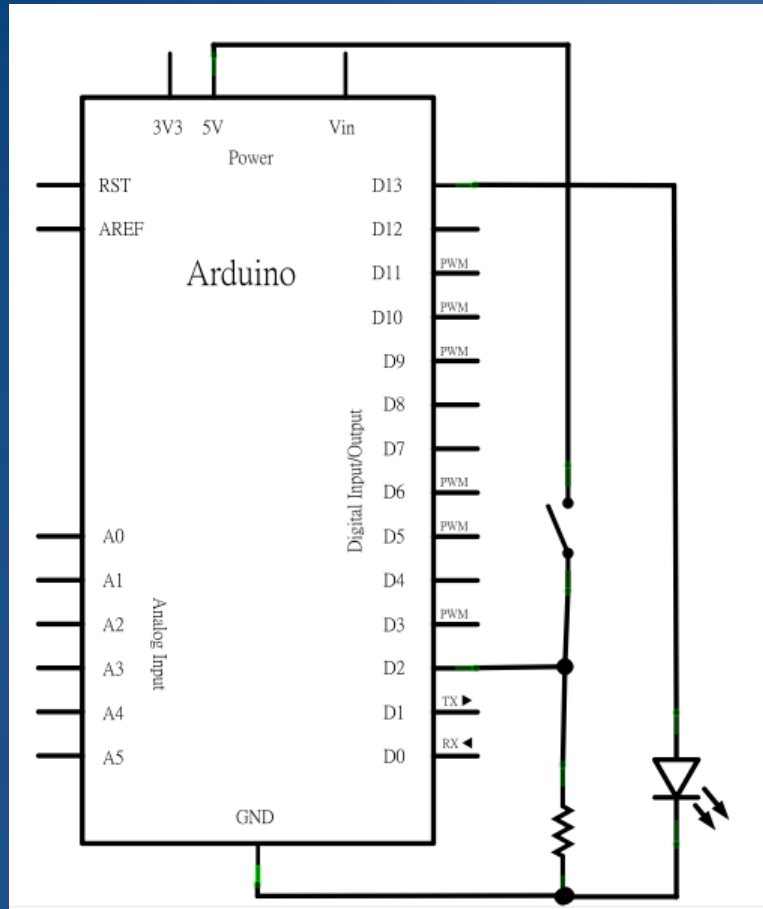
- unsigned long millis()
- delay(ms)
- delayMicroseconds(us)

unsigned long millis() : Board üzerindeki programın çalışmaya başlamasından bu ana kadar ki geçen zaman.

delay (ms) : unsigned long ms tipinde Milisaniye cinsinden gecikme

delayMicroseconds(us): Mikrosaniye cinsinden gecikme

Örnek Uygulama-1



```
const int buttonPin = 2; // pushbutton pin numarası
const int ledPin = 13; // LED pin numarası
int buttonState = 0; // pushbutton durumları okunacak
```

```
void setup()
{
// LED pinin bağlı olduğu Portu çıkış olarak yönlendir
pinMode(ledPin, OUTPUT);
// pushbutton pininin bağlı olduğu portu giriş olarak yönlendir
pinMode(buttonPin, INPUT);
}
```

```
void loop()
{
// pushbutton değerini / durumunu oku
buttonState = digitalRead(buttonPin);
// Eğer pushbuttona basıldı ise durumunu HIGH yap ve Ledi yak
if (buttonState == HIGH) {
digitalWrite(ledPin, HIGH);
}
else {
digitalWrite(ledPin, LOW);
}
}
```

```
butLed | Arduino 1.5.4
Dosya Düzenle Taslak Araçlar Yardım
butLed $
const unsigned int BUTTON_PIN=2;
const unsigned int LED_PIN =13;
void setup(){
pinMode(LED_PIN, OUTPUT);
pinMode(BUTTON_PIN,INPUT);
}
int led_state=LOW;

void loop(){
const int btnDurum = digitalRead (BUTTON_PIN);
if(btnDurum == HIGH){
led_state =(led_state == LOW)? HIGH : LOW;
digitalWrite(LED_PIN, led_state);
}
}
```


Seri iletişim (Giriş/Çıkış)

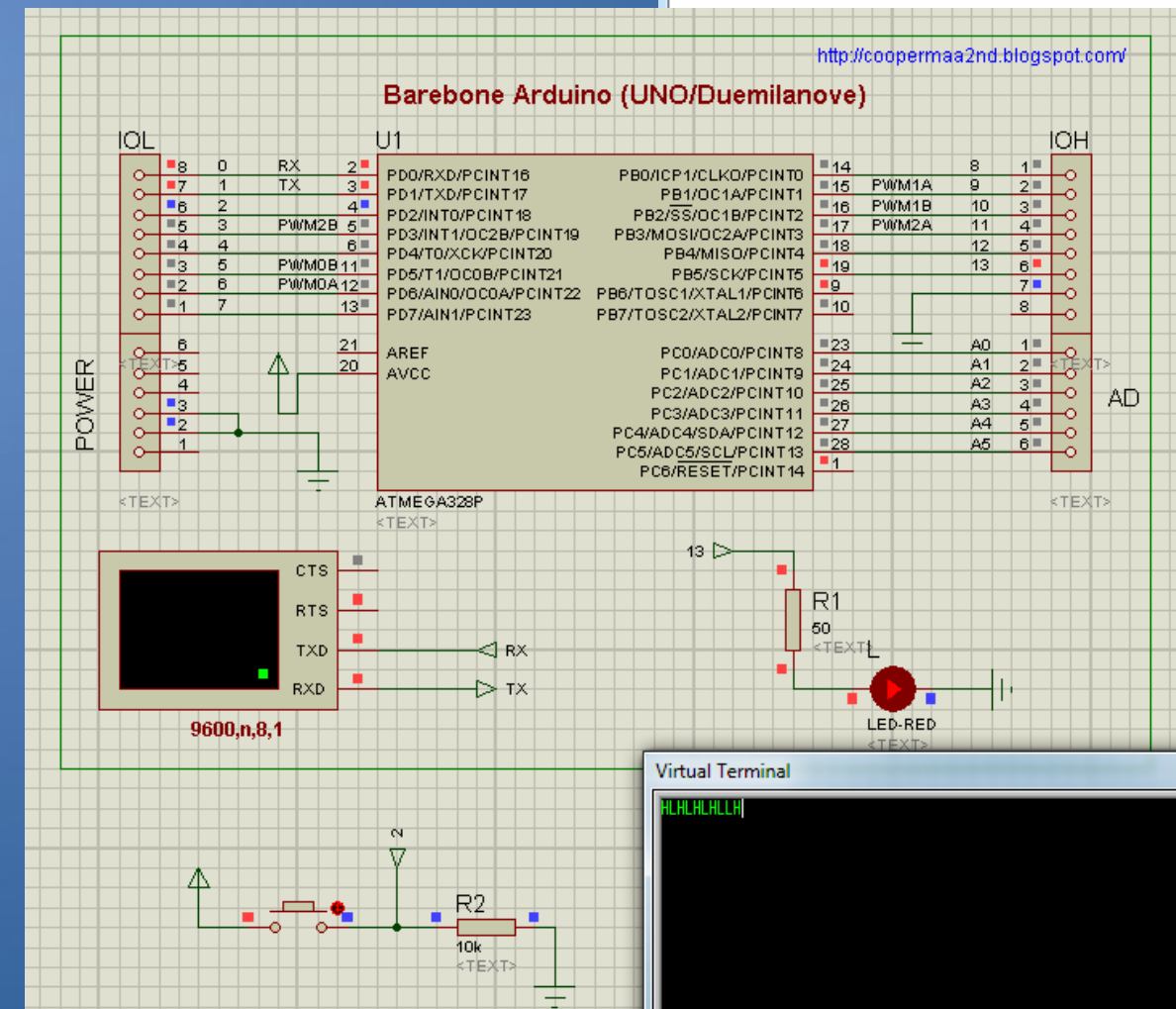
- Bilgisayar ile Arduino Board arasında iletişimde (USB, seri port) seri iletişim kullanılır. Bunun için digital pinlerden 0 (RX) ve 1 (TX) uçları kullanılır.
 - **Serial.begin(bps)**
 - **int Serial.available()**
 - **int Serial.read()**
 - **Serial.flush()**
 - **Serial.print(data)**
 - **Serial.println(data)**

Örnek Uygulama-2

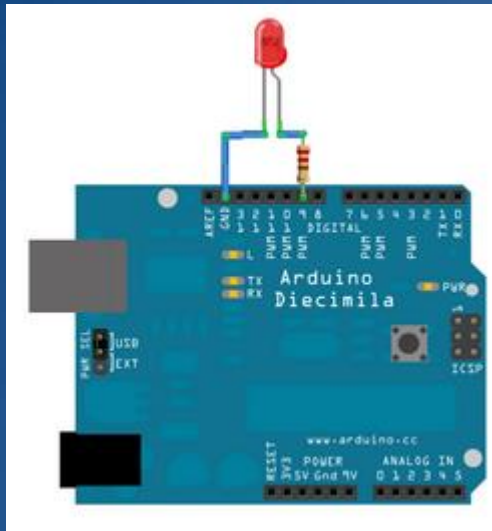
```
const int ledPin=13;
char gelen;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  if (Serial.available()>0)
    gelen=Serial.read();
  if(gelen=='H') digitalWrite (ledPin, HIGH);
  if(gelen=='L') digitalWrite (ledPin, LOW);
}
```

```
process_elipse | Arduino 1.5.4
Dosya Düzenle Taslak Araçlar Yardım
process_elipse
const int ledPin = 13;
int gelenByte;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // gelen bir veri varsa:
  if (Serial.available() > 0) {
    // FIFO (first in first out ) mantığı ile
    gelenByte = Serial.read();
    // gelen H (ASCII 72) ise LED i yak
    if (gelenByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // gelen L (ASCII 76) ise LED i söndür
    if (gelenByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

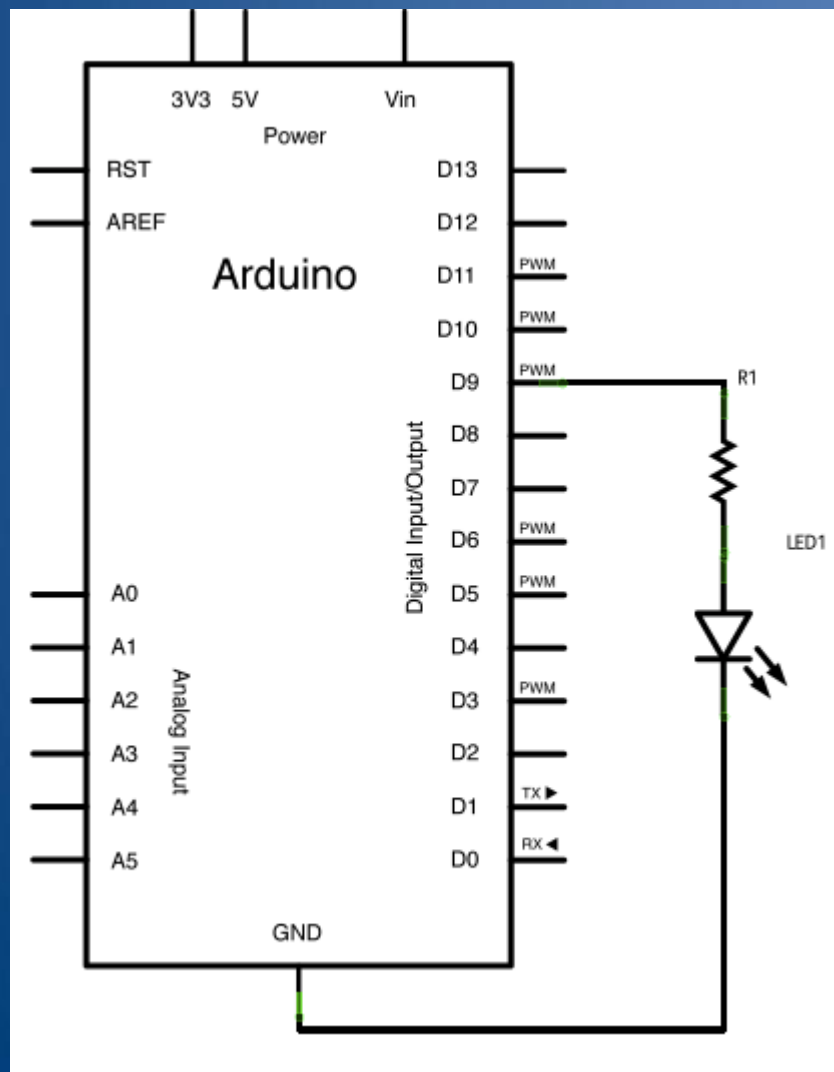


Analog Çıkış- Uygulama 3 (Dimmer)

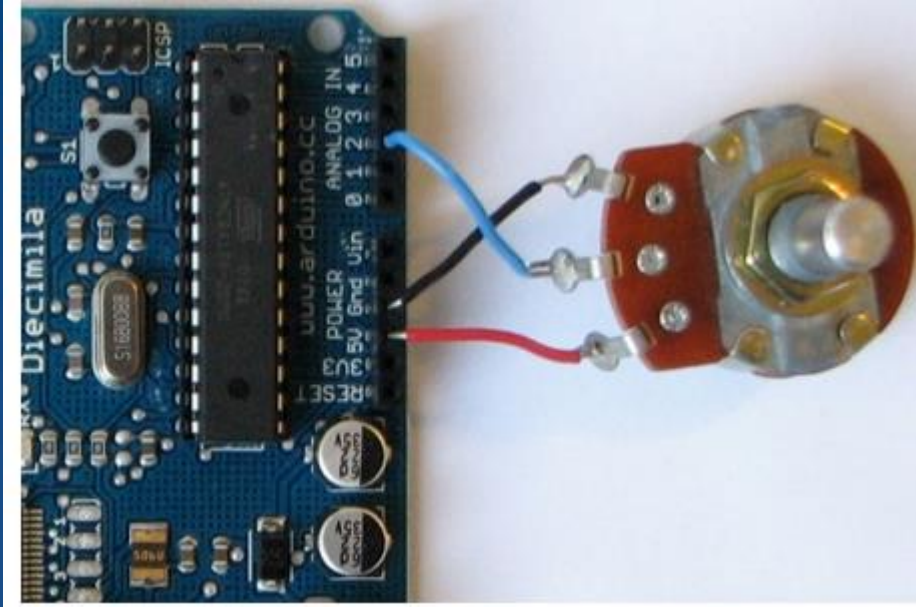


```
const int ledPin = 9;
void setup()
{
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  byte brightness;
  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    brightness = Serial.read();
    // set the brightness of the LED:
    analogWrite(ledPin, brightness);
  }
}
```



Analog Giriş- Uygulama 4 (Pot.)



```
int potPin = A2; // pot. pini seç
int ledPin = 13;
int val = 0; // sensordan gelen degeri tut

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

void loop() {
  val = analogRead(potPin); // oku sensorü
  digitalWrite(ledPin, HIGH);
  delay(val);
  digitalWrite(ledPin, LOW);
  delay(val);
}
```

Sıcaklık sensörü- Uygulama 5 (TMP36 sensörü)

Açıklama: Eğer 5V Arduino, kullanılıyorsa 10-bit analog donusumde Pin voltajı:
milliVolts = (reading from ADC) * (5000/1024)
// analog okunan değerler 0 - 1023 arasındadır .Bu formül 0-1023 arasındaki analog degeri 0-5000mV (= 5V) e donusturur. Sistem 5v ile beslendiğinden 5.0/1024 değeri ile okunan değeri çarparsak voltaj değeri bulunur

Eğer 3.3V Arduino, kullanılıyorsa 10-bit analog donusumde Pin voltajı:
milliVolts = (reading from ADC) * (3300/1024)
// Buda 0-3300mV (= 3.3V) arası donusum yapar

Sonrasında ise mV cinsinden Derece için;
Sicaklik= [(analog voltage in mV) - 500] / 10

// Ve okunan voltajı ekrana seri port üzerinden yaz.
Serial.println(voltage);

```
//TMP36 Pin bağlantısı
int sensorPin = 0;
//çözünürlük 10 mV / derece ve 500 mV offset değeri
void setup()
{
  Serial.begin(9600);
}

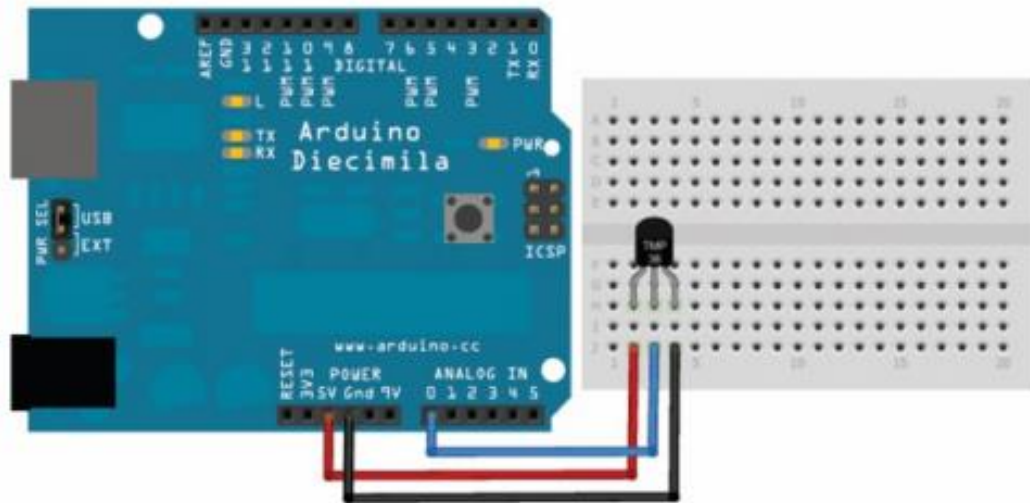
void loop()
{
  //sıcaklık sensöründen voltaj degerini oku
  int reading = analogRead(sensorPin);

  // okunan degeri voltaja donustur(3.3v arduino için 3.3 kullan)
  float voltage = reading * 5.0;
  voltage /= 1024.0;
  // voltajı yaz
  Serial.print(voltage); Serial.println(" volts");

  // sicaklığı yaz
  float temperatureC = (voltage - 0.5) * 100 ;
  //converting from 10 mv per degree wit 500 mV offset
  //to degrees ((voltage - 500mV) times 100)
  Serial.print(temperatureC); Serial.println(" C derece");

  // Fahrenheit a dönüştür
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF); Serial.println(" F derece");

  delay(1000);
}
```



Operatörler

Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- &= (compound bitwise and)
- |= (compound bitwise or)

Arithmetic Operators

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)
- ! (not)

```
if ( val > 10 && val < 20)
```

```
if ( val < 10 || val > 20)
```


Kontrol Komutlari

Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return

```
int counter = 0;
void setup() {
    Serial begin(9600);

    void loop() {

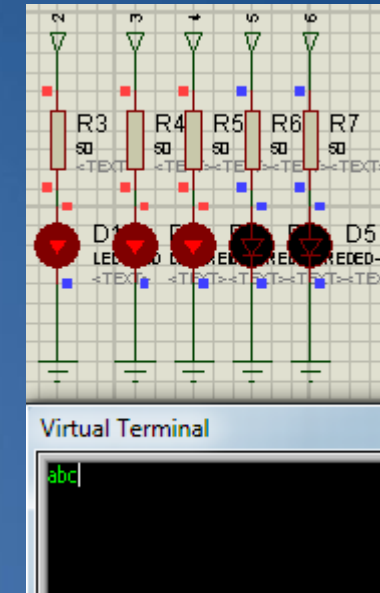
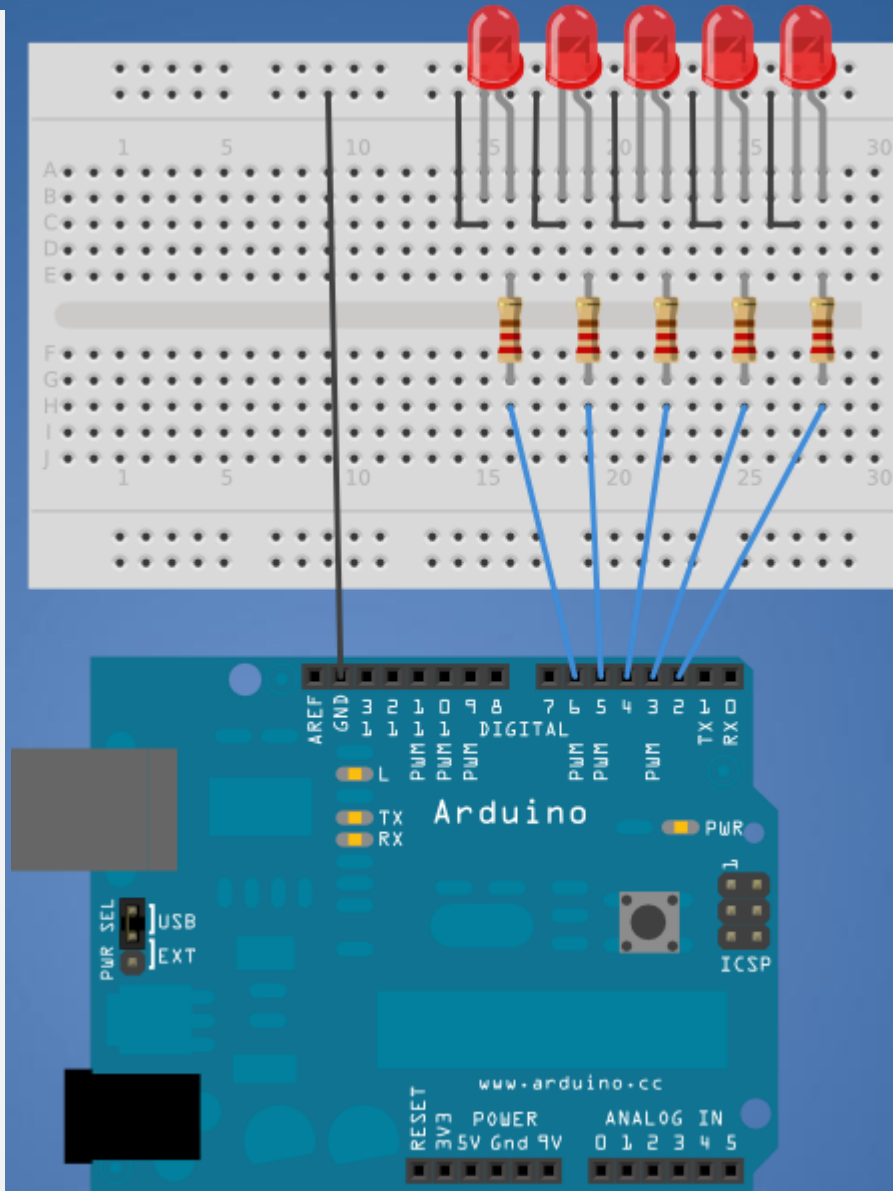
        if(counter < 10)
        {
            Serial println("less than 10");
        }
        else if (counter == 10)
        {
            Serial println("equal to 10");
        }
        else
        {
            Serial println("greater than 10");
            Serial end();
        }
        counter = counter + 1;
    }
}
```

ASCII Karakter Gönderme-Seri İletişim Uygulama-5

```
void setup() {
  Serial.begin(9600);
  // LED bağlanacak pinleri çıkış olarak ata
  for (int Pin = 2; Pin < 7; Pin++) {
    pinMode(Pin, OUTPUT);
  }
}

void loop() {
  // oku seri girisi:
  if (Serial.available() > 0) {
    int inByte = Serial.read();
    // ASCII değerler; 'a' = 97, 'b' = 98, vs..

    switch (inByte) {
      case 'a':
        digitalWrite(2, !digitalRead (2));
        break;
      case 'b':
        digitalWrite(3, HIGH);
        break;
      case 'c':
        digitalWrite(4, HIGH);
        break;
      case 'd':
        digitalWrite(5, HIGH);
        break;
      case 'e':
        digitalWrite(6, HIGH);
        break;
      default:
        // LEDler sönük
        for (int Pin = 2; Pin < 7; Pin++) {
          digitalWrite(Pin, LOW);
        }
    }
  }
}
```



Dizi (Array) Tanımlama- Uygulama 6

- Kullanım Şekli: `Tip DiziAdi [] = { Dizi Elemanlari,.... };`

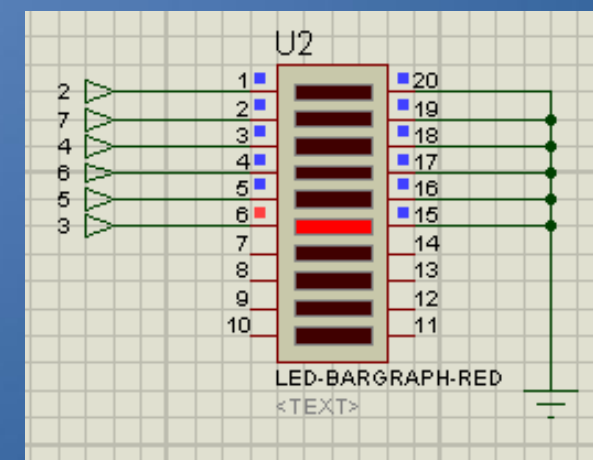
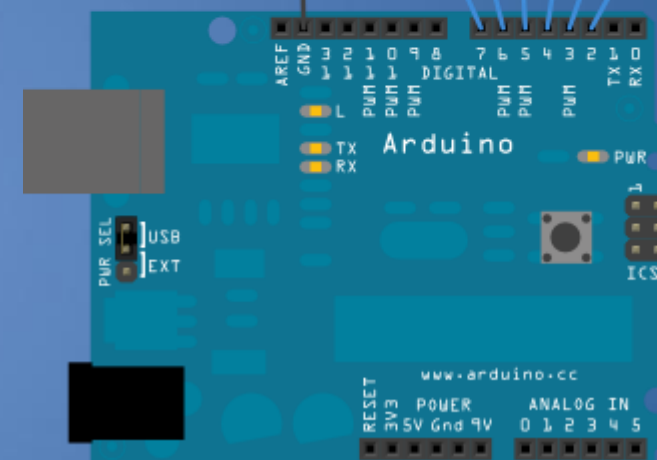
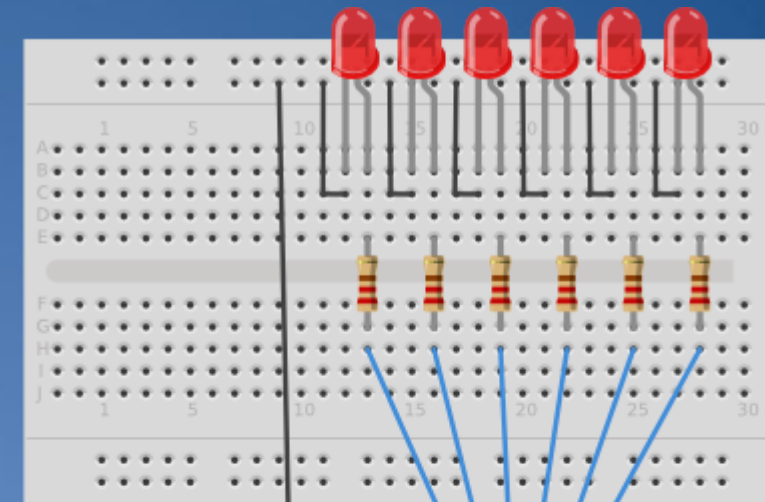
sketch_Dizi

```
int ledPins[] = { 2, 7, 4, 6, 5, 3 };
int pinCount = 6;

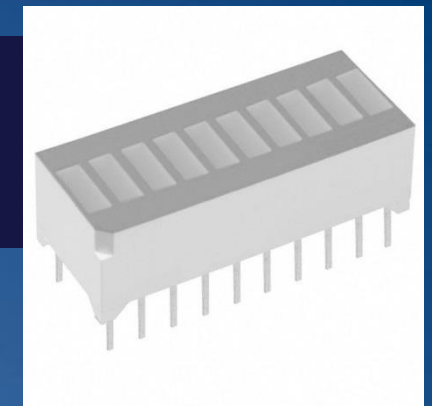
void setup()
{
  for (int thisPin = 0; thisPin < pinCount; thisPin++)
  {
    pinMode(ledPins[thisPin], OUTPUT);
  }
}

void loop()
{
  for (int thisPin = 0; thisPin < pinCount; thisPin++) {
    digitalWrite(ledPins[thisPin], HIGH);
    delay(100);
    digitalWrite(ledPins[thisPin], LOW);
  }

  for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {
    // turn the pin on:
    digitalWrite(ledPins[thisPin], HIGH);
    delay(100);
    // turn the pin off:
    digitalWrite(ledPins[thisPin], LOW);
  }
}
```



LED Bar Graph- Uygulama 7

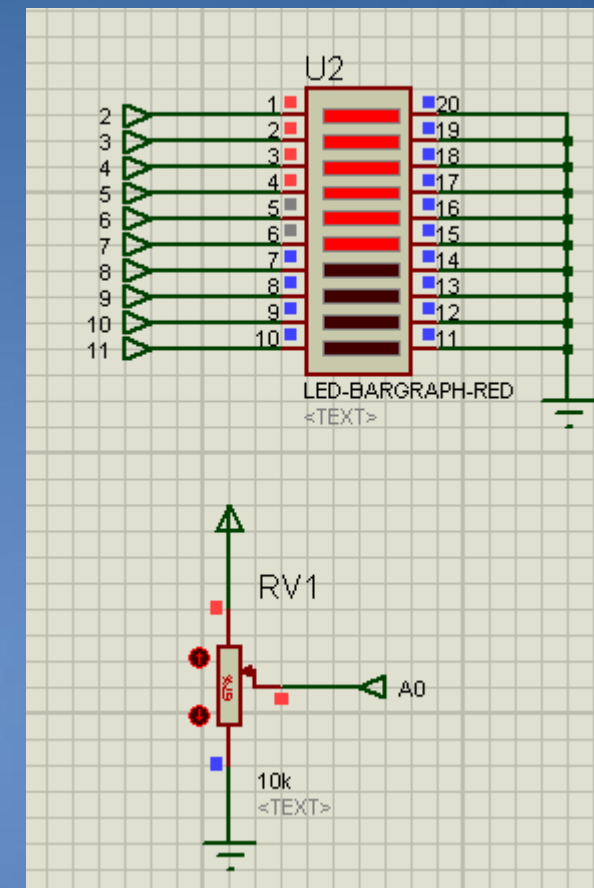


LEDbarGraph

```
// http://www.arduino.cc/en/Tutorial/BarGraph
const int analogPin = A0; // potentiometer bağlı pin
const int ledCount = 10; // Led sayısı
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };
void setup() {
  // Led bağlı pinler çıkış
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    pinMode(ledPins[thisLed], OUTPUT);
  }
}

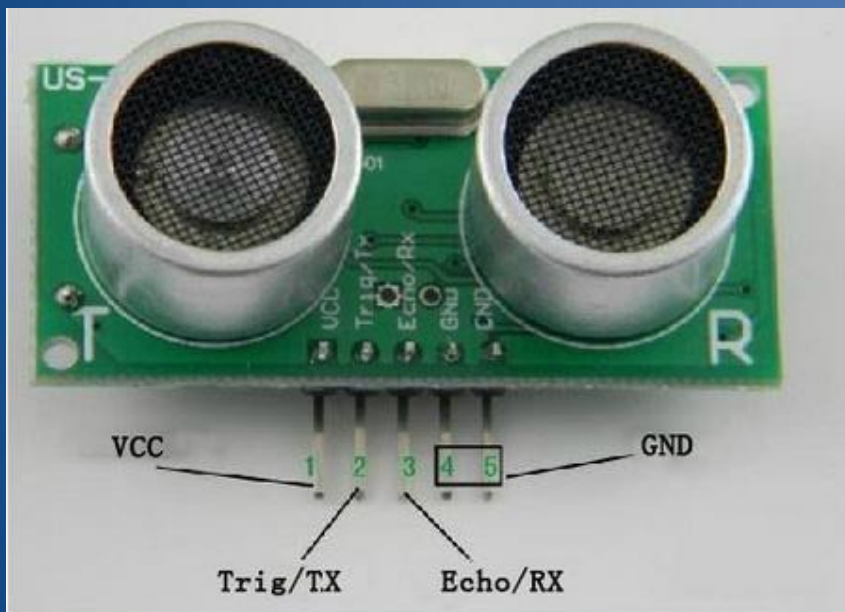
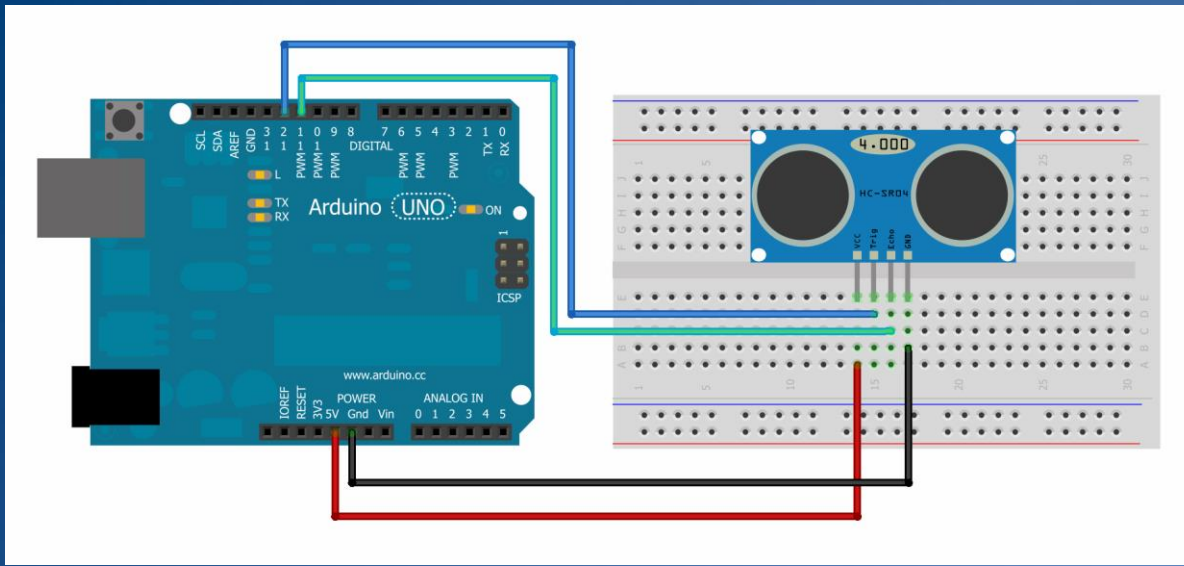
void loop() {
  // read the potentiometer:
  int sensorReading = analogRead(analogPin);
  // map the result to a range from 0 to the number of LEDs:
  int ledLevel = map(sensorReading, 0, 1023, 0, ledCount);

  // loop over the LED array:
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    // if the array element's index is less than ledLevel,
    // turn the pin for this element on:
    if (thisLed < ledLevel) {
      digitalWrite(ledPins[thisLed], HIGH);
    }
    // turn off all pins higher than the ledLevel:
    else {
      digitalWrite(ledPins[thisLed], LOW);
    }
  }
}
```



map(value, fromLow, fromHigh, toLow, toHigh)

Fonksiyon Tanımlama: Uygulama:Ultrasonic Sensor



sketch_oct25b §

```
int inputPin=4; // connect digital I/O 4 to the ECHO/Rx Pin
int outputPin=5; // connect digital I/O 5 to the TRIG/TX Pin

void setup()
{
  Serial.begin(9600);
  pinMode(inputPin, INPUT);
  pinMode(outputPin, OUTPUT);
}

unsigned long ping()
{
  digitalWrite(outputPin, LOW); // send low pulse for 2µs
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH); // send high pulse for 10µs
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW); // back to low pulse
  int distance = pulseIn(inputPin, HIGH); // read echo value
  // distance = pulseIn(inputPin, HIGH, 38000)
  int distancel= distance/29/2; // in cm
  //distancel = (distance / 58.138);
  return distancel;
}

void loop()
{
  int x = 0;
  x = ping();
  Serial.println(x);
  delay(250); //delay 1/4 seconds.
}
```

LCD- Uygulama 9 (Pot.)

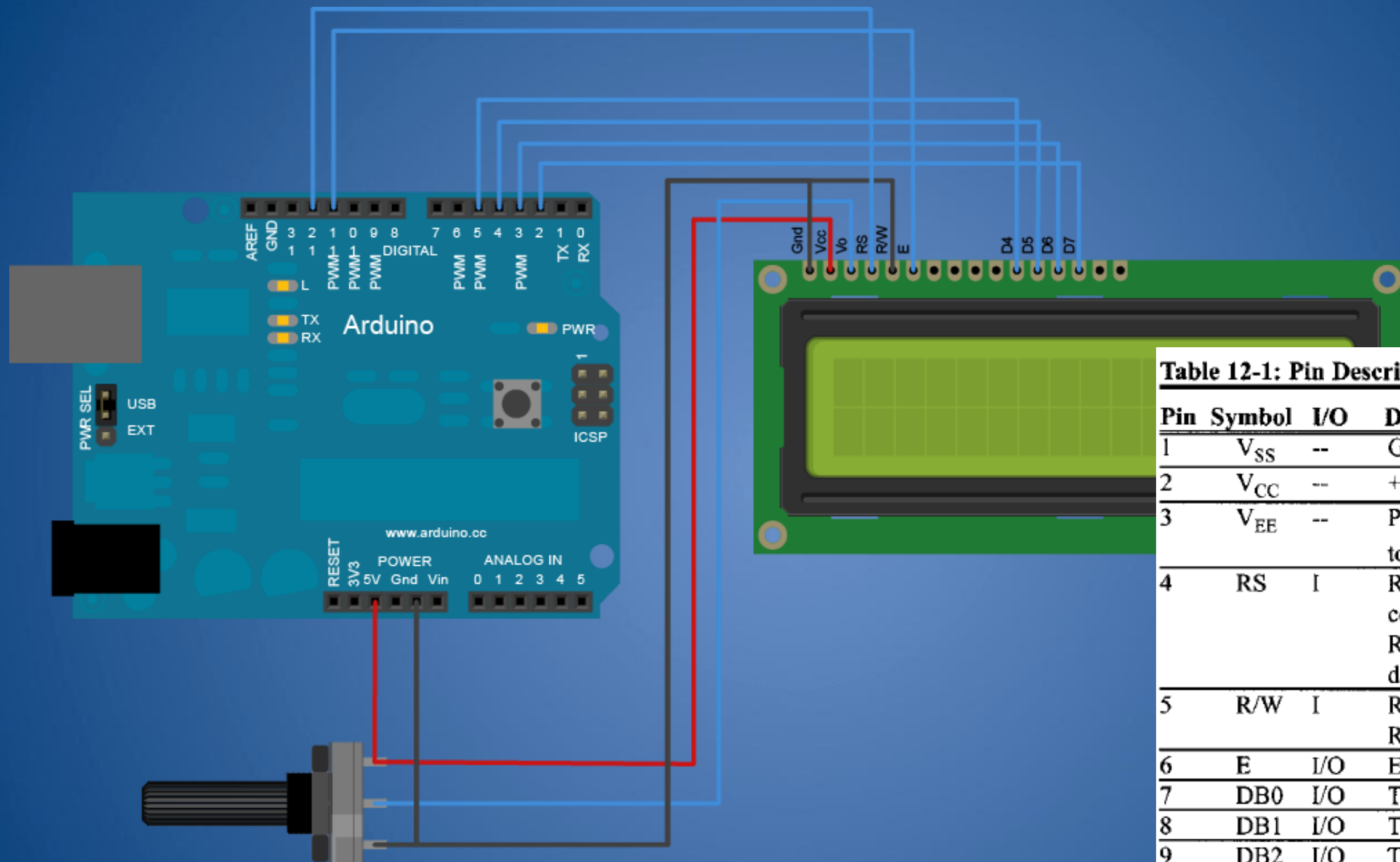
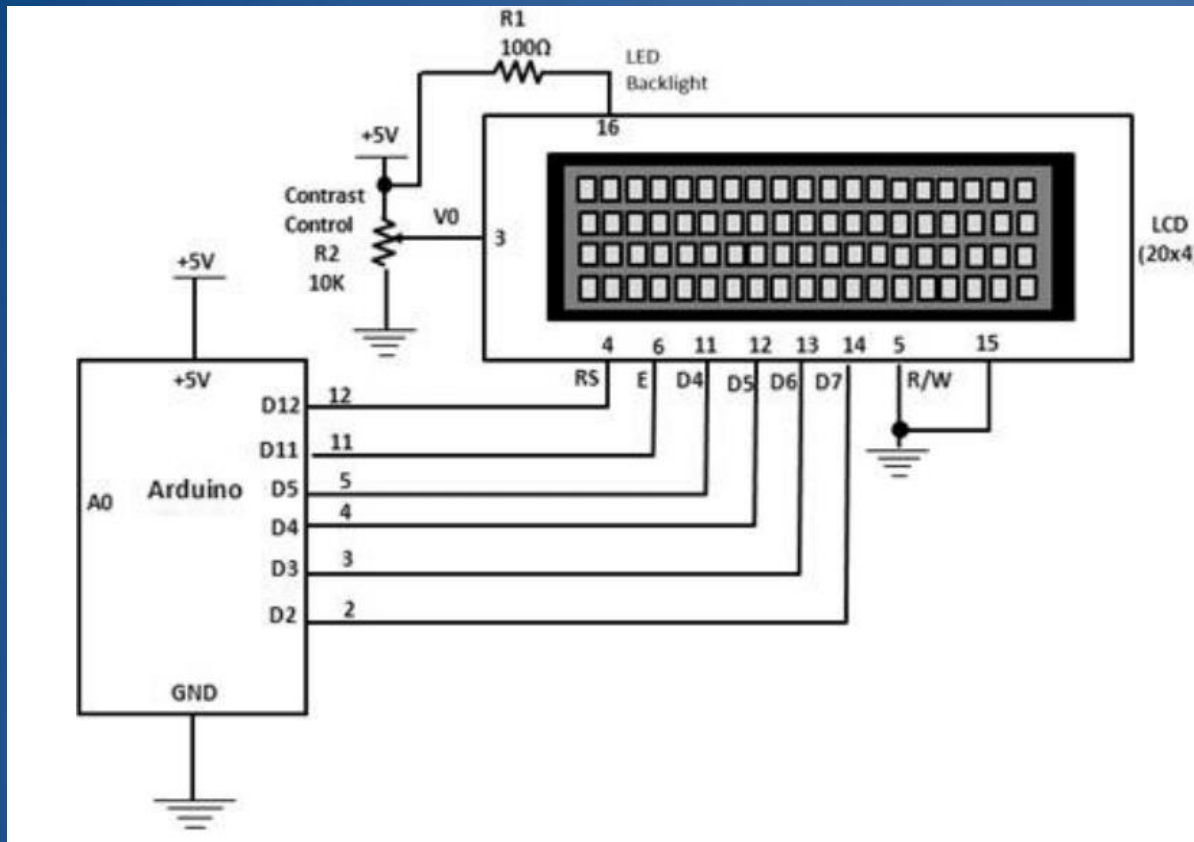


Table 12-1: Pin Descriptions for LCD

Pin	Symbol	I/O	Description
1	V _{SS}	--	Ground
2	V _{CC}	--	+5 V power supply
3	V _{EE}	--	Power supply to control contrast
4	RS	I	RS = 0 to select command register, RS = 1 to select data register
5	R/W	I	R/W = 0 for write, R/W = 1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

Paralel LCD- LiquidCrystal()



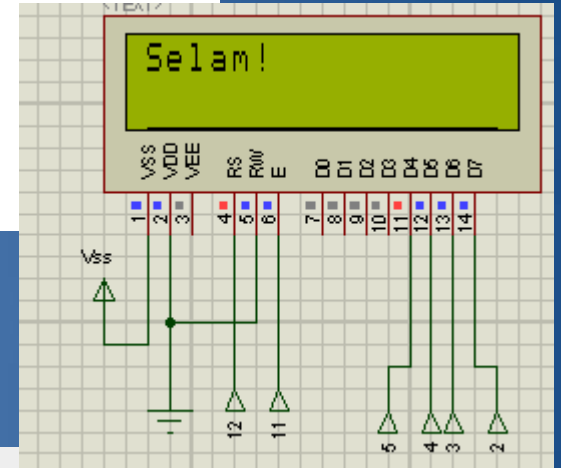
Örnek

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Selam!");
}

void loop() {}
```



Kullanım Şekli:

LiquidCrystal(rs, enable, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)

LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

Paralel LCD-Seri Port: Uygulama 10

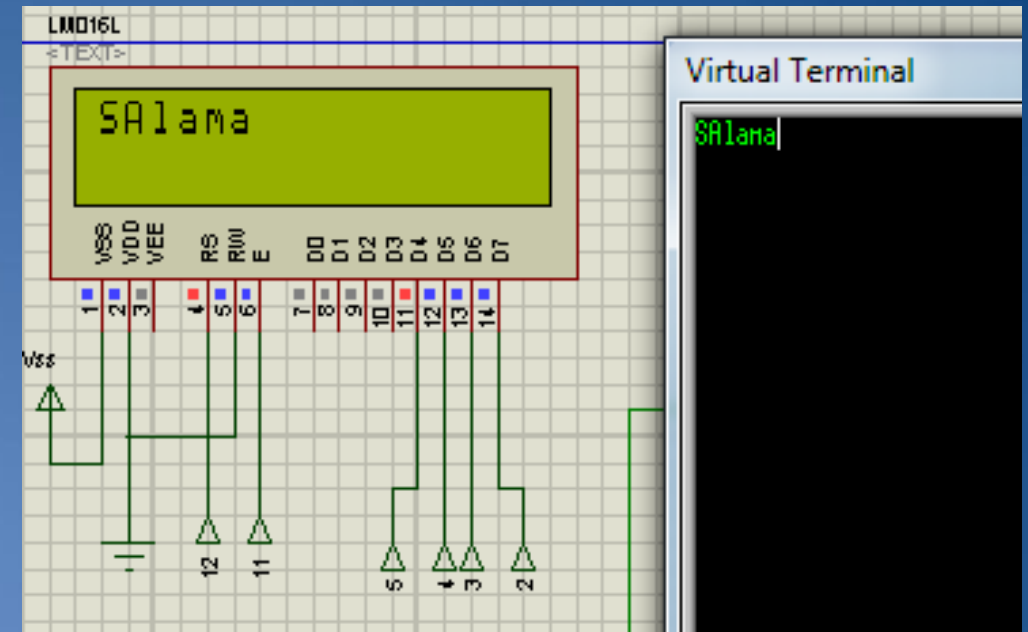
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

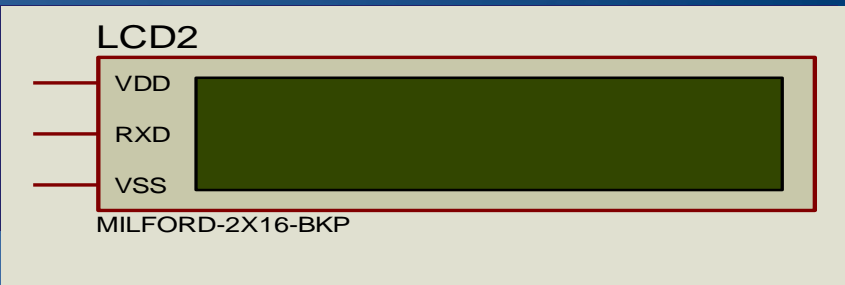
void setup(){

  lcd.begin(16, 2);
  // Seri iletişim hızı
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()) {
    // mesajı almak için bekle
    delay(200);
    // Ekranı temizle
    lcd.clear();
    // Girilen karakterleri oku
    while (Serial.available() > 0) {
      // Herbir karakteri LCD de göster
      lcd.write(Serial.read());
    }
  }
}
```

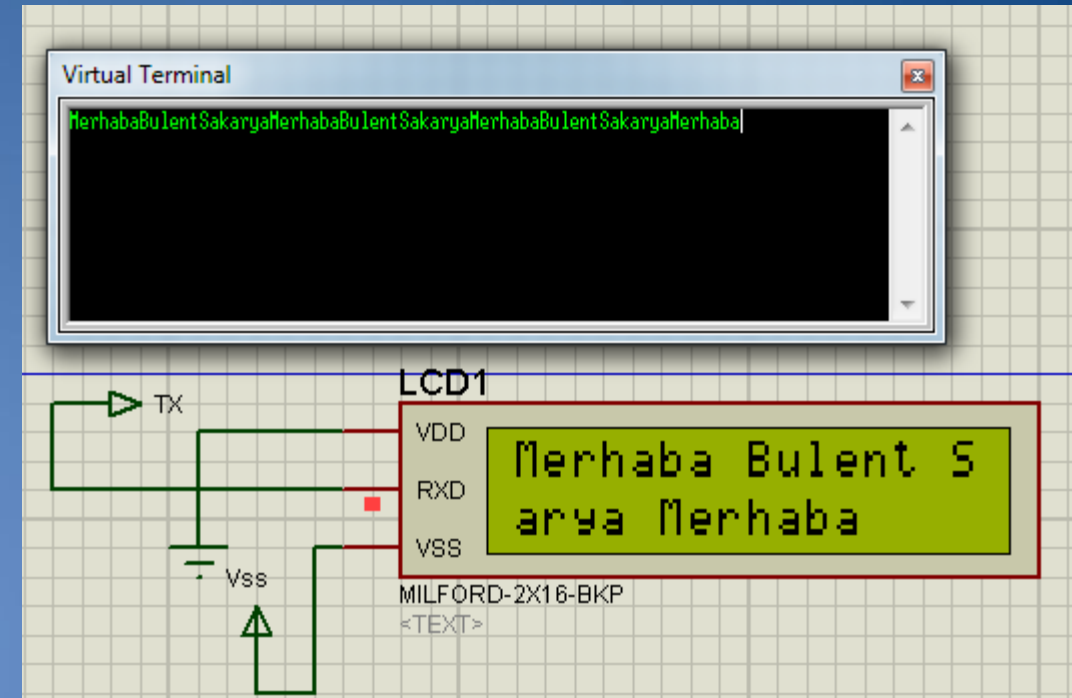


Seri LCD –Uygulama 11



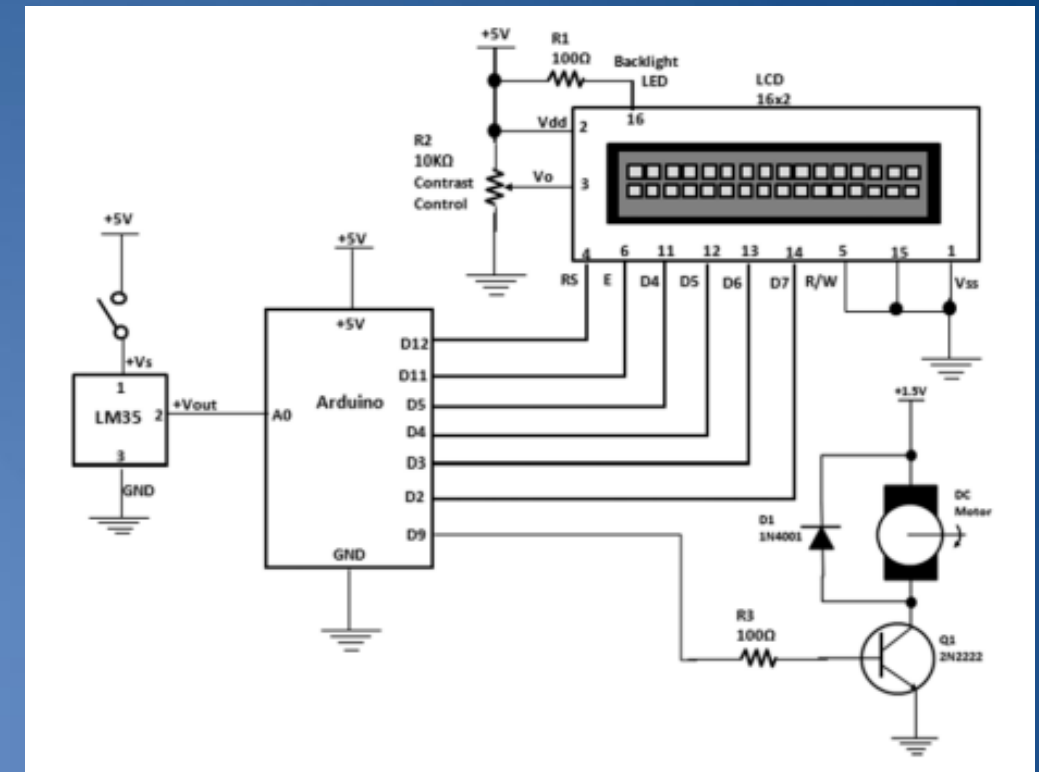
```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  delay(1000);
  Serial.print("Merhaba\n");
  delay(1000);
  Serial.print("Bulent\n");
  delay(1000);
  Serial.print("Sakarya\n");
}
```



Proje: Giyilebilir Fanlı LCD Ekranlı Şapka

```
int sensorPin = 0; // The analog pin the LM35's Vout is connected to.
const int motorPin = 9; // the number of the motor pin
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
/*
Initialize serial connection with the computer*/
void setup()
{
  Serial.begin(9600); // Begin serial connection with the computer
  lcd.begin(16,2);
  analogReference(INTERNAL);
  /* for Arduino Mega please use analogReference(INTERNAL1v1); */
  pinMode(motorPin, OUTPUT);
}
void loop()
{
  int reading = analogRead(sensorPin); // read data from LM35 using Arduino (A0) pin
  float voltage = reading * 5.0; // Convert sensor data to voltage
  voltage /= 1024.0;
  Serial.print(voltage); Serial.println("volts"); // Print voltage on serial monitor
  float temperatureC = ((100 * 1.1 * voltage) / 1024) * 100;
  float temperatureF = (temperatureC * (9.0 / 5.0)) + 32; // Convert voltage to temperature
  Serial.print(temperatureF); Serial.println(" degrees F"); // Print Temperature in C
  // display Temperature on LCD
  lcd.setCursor(0,0);
  lcd.print("Temperature=");
  lcd.setCursor(0,1);
  lcd.print(temperatureF); lcd.println(" degrees F ");
  //DC Motor control
  if(temperatureF > 58){
    digitalWrite(motorPin, HIGH);
  }
  else{
    digitalWrite(motorPin, LOW);
  }
  delay(10); // print data every 10 milliseconds
}
```



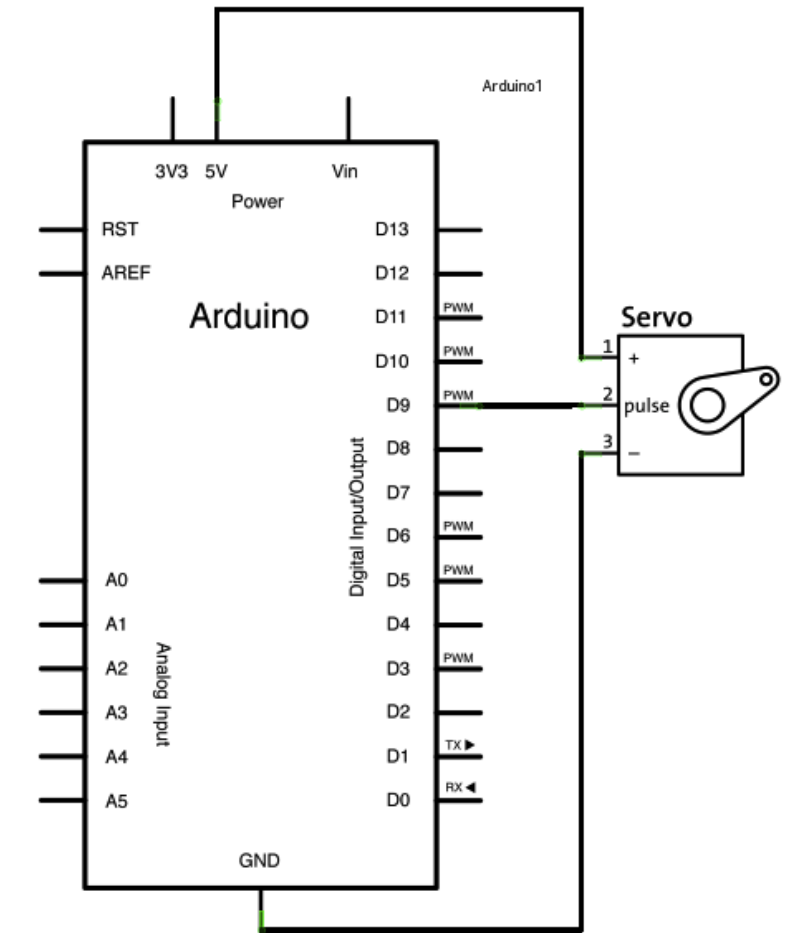
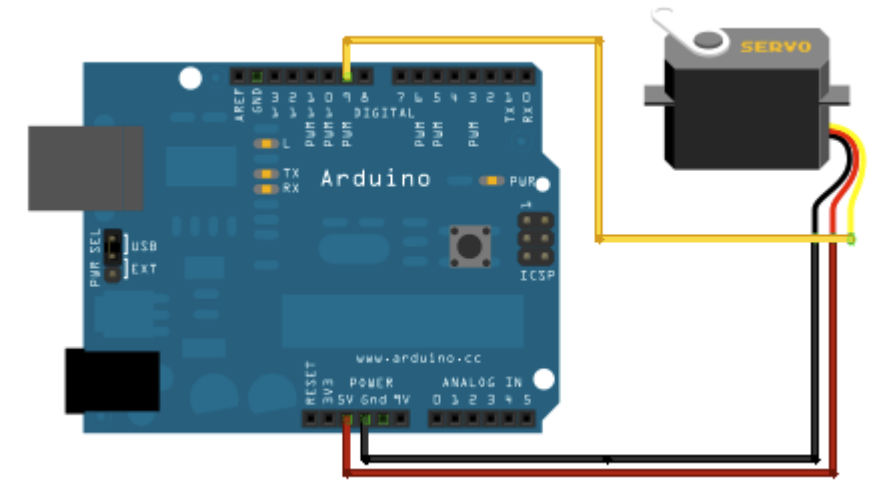
RC Servo Motor Örneği-1

```
#include <Servo.h>
```

```
Servo myservo; // Bir servo nesnesi oluşturuldu  
int pos = 0; // servo pozisyonunu tutan değişken
```

```
void setup()  
{  
  myservo.attach(9); // servonun bağlandığı pin  
}
```

```
void loop()  
{  
  for(pos = 0; pos < 180; pos += 1) // 0-180 derece  
  arası  
  {  
    myservo.write(pos);  
    delay(15);  
  }  
  for(pos = 180; pos >= 1; pos -= 1)  
  {  
    myservo.write(pos);  
    delay(15);  
  }  
}
```



RC Servo Motor Örneği-2

```
#include <Servo.h>
```

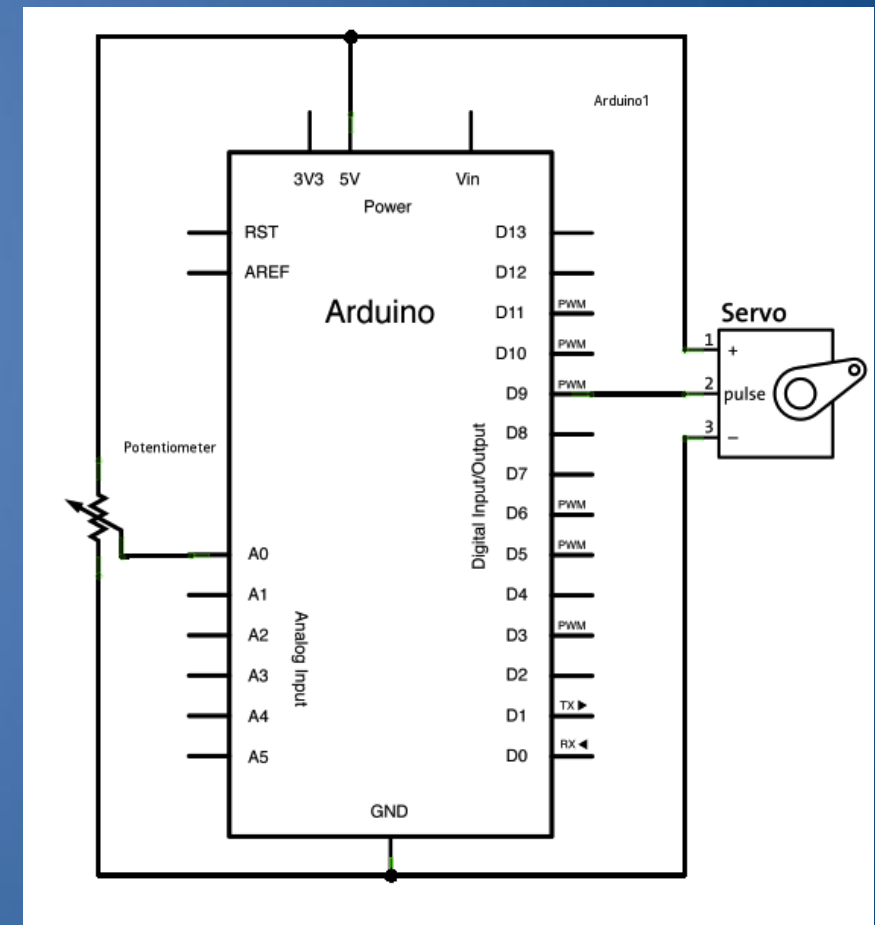
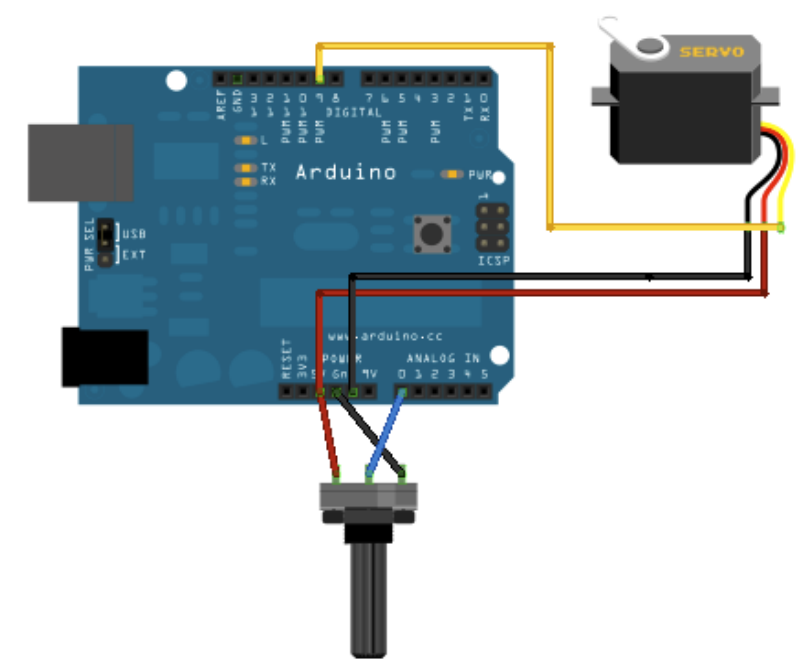
```
Servo myservo; // Bir servo nesnesi oluşturuldu  
int potpin = 0; // pot. Bağlananan analog pin  
int val; // analog pinden okunan degerleri tutan deg.
```

```
void setup()
```

```
{  
  myservo.attach(9); // servo bağlanan pin  
}
```

```
void loop()
```

```
{  
  val = analogRead(potpin); // oku potu ( 0 - 1023)  
  val = map(val, 0, 1023, 0, 179); // kalibre et  
  myservo.write(val); // poz. yaz  
  delay(15); // bekle  
}
```



Step Motor Örneği-1 (Kütüphane kullanmadan)

```
int motorPin1 = 8;  
int motorPin2 = 9;  
int motorPin3 = 10;  
int motorPin4 = 11;  
int delayTime = 500;
```

```
void setup() {  
  pinMode(motorPin1, OUTPUT);  
  pinMode(motorPin2, OUTPUT);  
  pinMode(motorPin3, OUTPUT);  
  pinMode(motorPin4, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(motorPin1, HIGH);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, HIGH);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, HIGH);  
  digitalWrite(motorPin4, LOW);  
  delay(delayTime);  
  digitalWrite(motorPin1, LOW);  
  digitalWrite(motorPin2, LOW);  
  digitalWrite(motorPin3, LOW);  
  digitalWrite(motorPin4, HIGH);  
  delay(delayTime);  
}
```

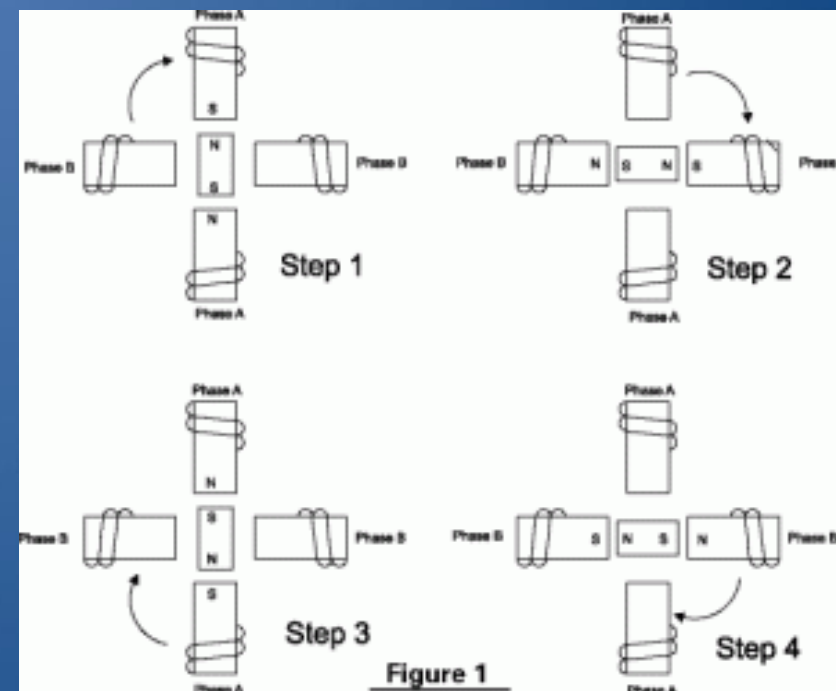
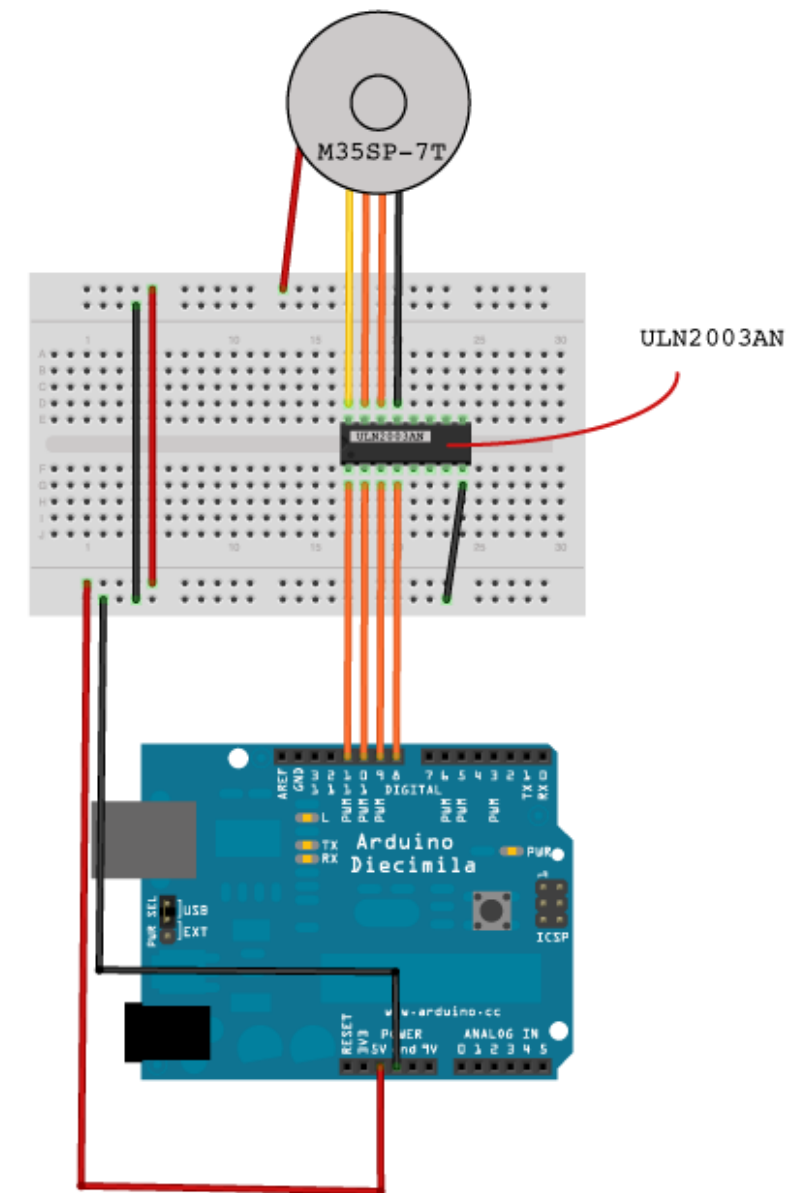


Figure 1

Step Motor Örneği-2

```
#include <Stepper.h>
```

```
int in1Pin = 12;  
int in2Pin = 11;  
int in3Pin = 10;  
int in4Pin = 9;
```

```
Stepper motor(100, in1Pin, in2Pin, in3Pin, in4Pin);
```

```
void setup()
```

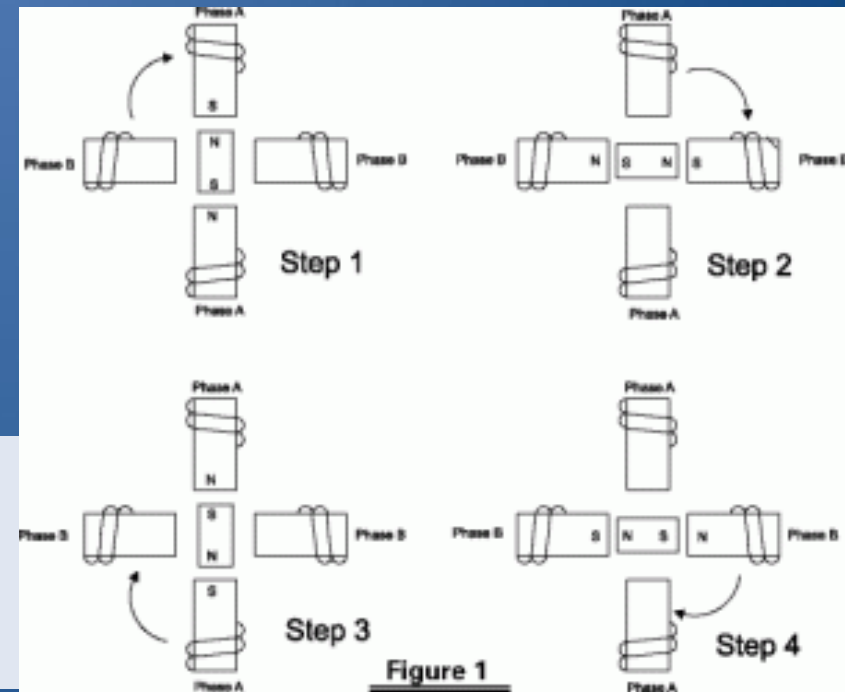
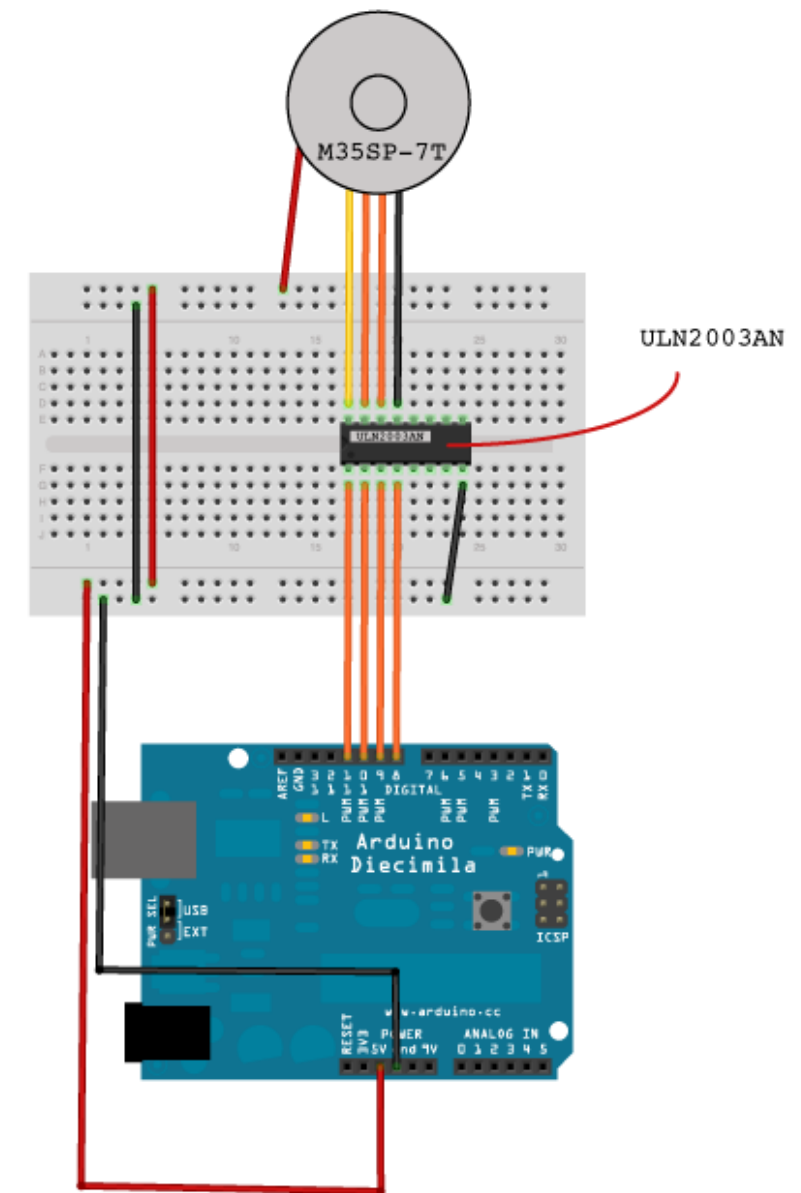
```
{  
  pinMode(in1Pin, OUTPUT);  
  pinMode(in2Pin, OUTPUT);  
  pinMode(in3Pin, OUTPUT);  
  pinMode(in4Pin, OUTPUT);  
}
```

```
while (!Serial);  
Serial.begin(9600);  
motor.setSpeed(20);  
}
```

```
void loop()
```

```
{  
  if (Serial.available())  
  {  
    int steps = Serial.parseInt();  
    motor.step(steps);  
  }  
}
```

steps: motorun bir tam turdaki adım sayısı.
Örneğin $360 / 3.6 = 100$ adım
pin1, pin2: motorun bağlandığı pinler
pin3, pin4: seçimlik motorun bağlandığı pinler



Step Motor Örneği-3

```
#include <Stepper.h>

#define STEPS 100 // step motor adım

Stepper stepper(STEPS, 8, 9, 10, 11); //pinler

int previous = 0;

void setup()
{
  stepper.setSpeed(30); // motor hızı 30 RPM
}

void loop()
{
  int val = analogRead(0);

  stepper.step(val - previous);

  previous = val;
}
```

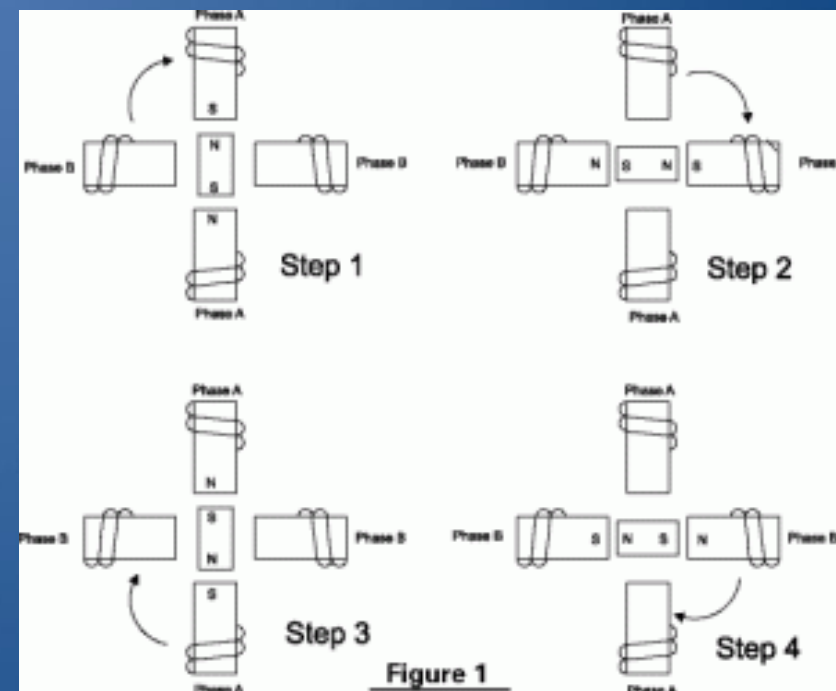
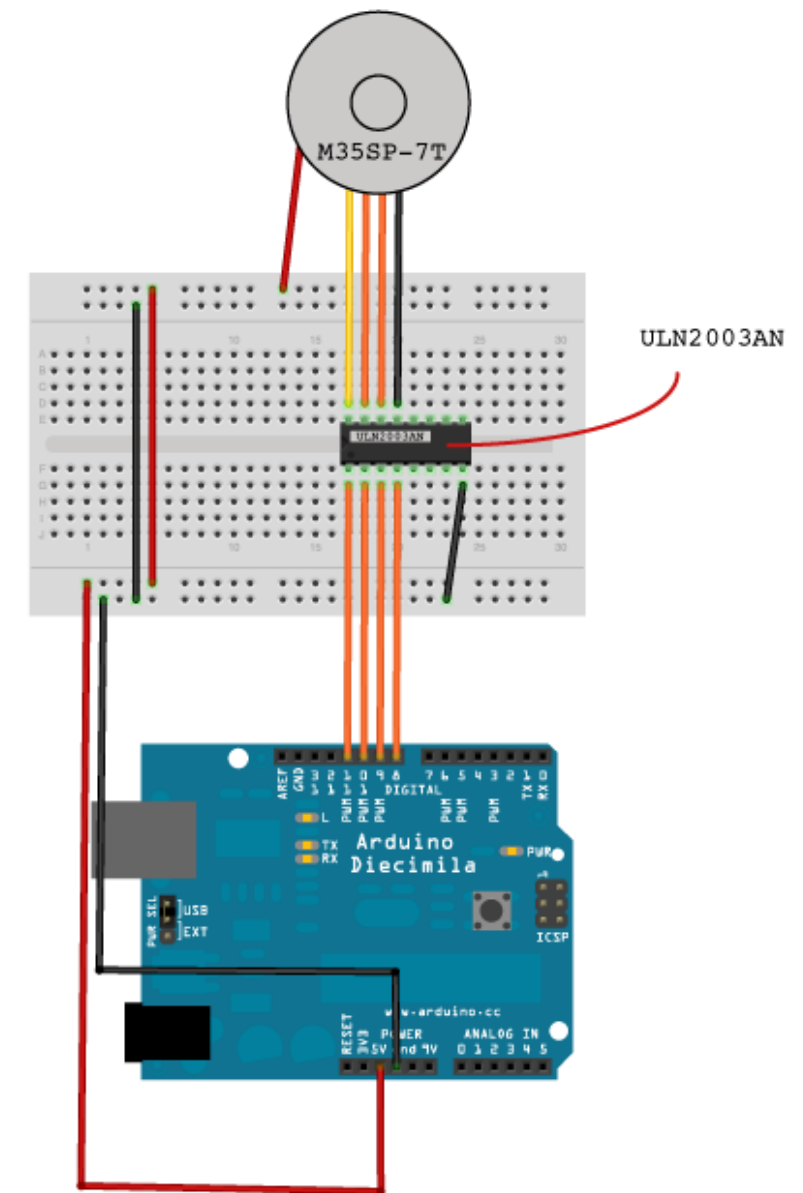
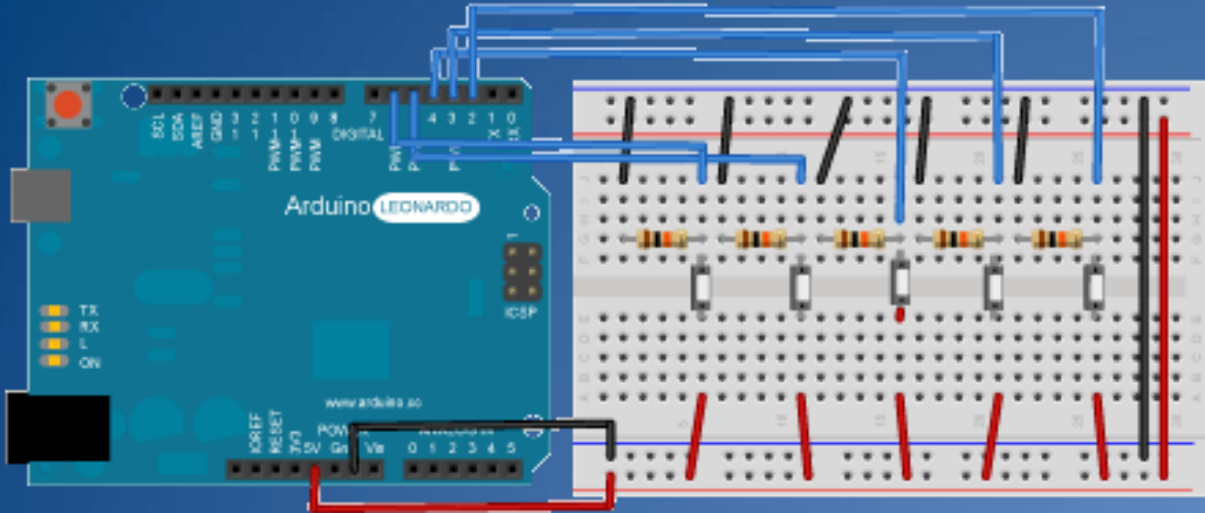


Figure 1

Proje: Sanal Klavye & Fare

Arduino; bilgisayarınızın klavyesi veya faresi olsun



3D Air Mouse projesi:

<http://www.instructables.com/id/3D-AIR-mouse-Arduino-Processing/>

```
const int upButton = 2;
const int downButton = 3;
const int leftButton = 4;
const int rightButton = 5;
const int mouseButton = 6;
```

```
void setup() { // initialize the buttons' inputs:
  pinMode(upButton, INPUT);
  pinMode(downButton, INPUT);
  pinMode(leftButton, INPUT);
  pinMode(rightButton, INPUT);
  pinMode(mouseButton, INPUT);
```

```
  Serial.begin(9600);
  // initialize mouse control:
  Mouse.begin();
  Keyboard.begin();
```

```
}
```

```
void loop() {
  // use serial input to control the mouse:
  if (Serial.available() > 0) {
    char inChar = Serial.read();

    switch (inChar) {
      case 'u':
        // move mouse up
        Mouse.move(0, -40);
        break;
      case 'd':
        // move mouse down
        Mouse.move(0, 40);
        break;
      case 'l':
        // move mouse left
        Mouse.move(-40, 0);
        break;
      case 'r':
        // move mouse right
        Mouse.move(40, 0);
        break;
      case 'm':
        // perform mouse left click
        Mouse.click(MOUSE_LEFT);
        break;
    }
  }
  // use the pushbuttons to control the keyboard:
  if (digitalRead(upButton) == HIGH) {
    Keyboard.write('u');
  }
  if (digitalRead(downButton) == HIGH) {
    Keyboard.write('d');
  }
  if (digitalRead(leftButton) == HIGH) {
    Keyboard.write('l');
  }
  if (digitalRead(rightButton) == HIGH) {
    Keyboard.write('r');
  }
  if (digitalRead(mouseButton) == HIGH) {
    Keyboard.write('m');
  }
}
```