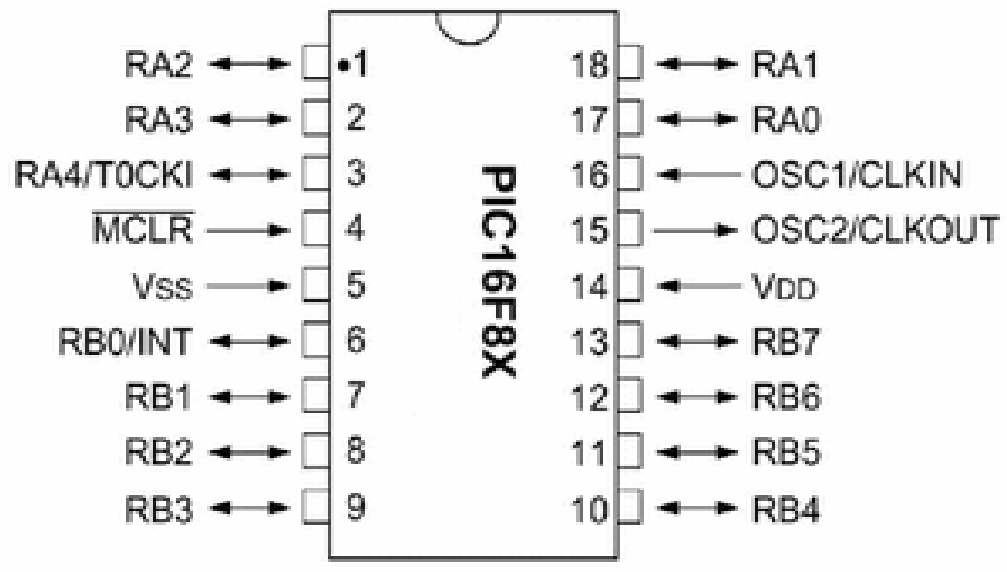
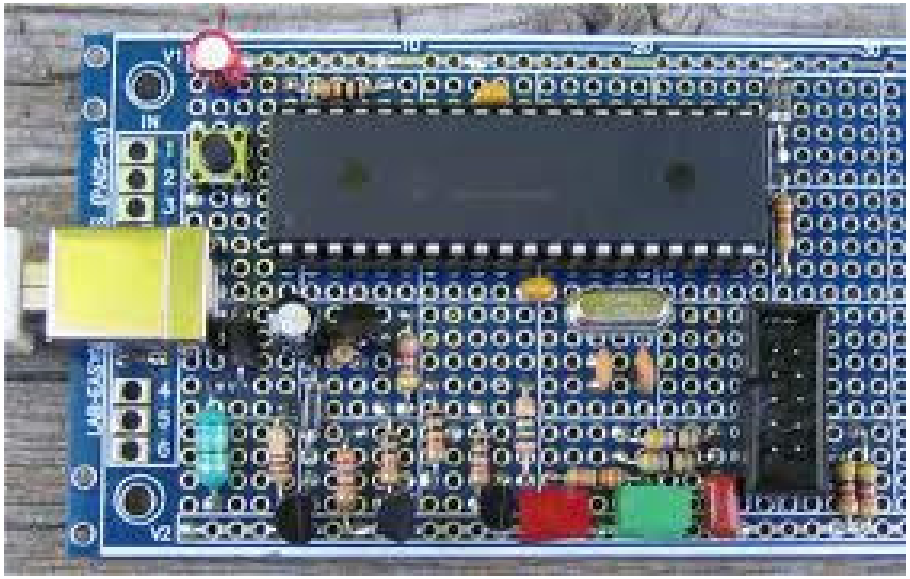
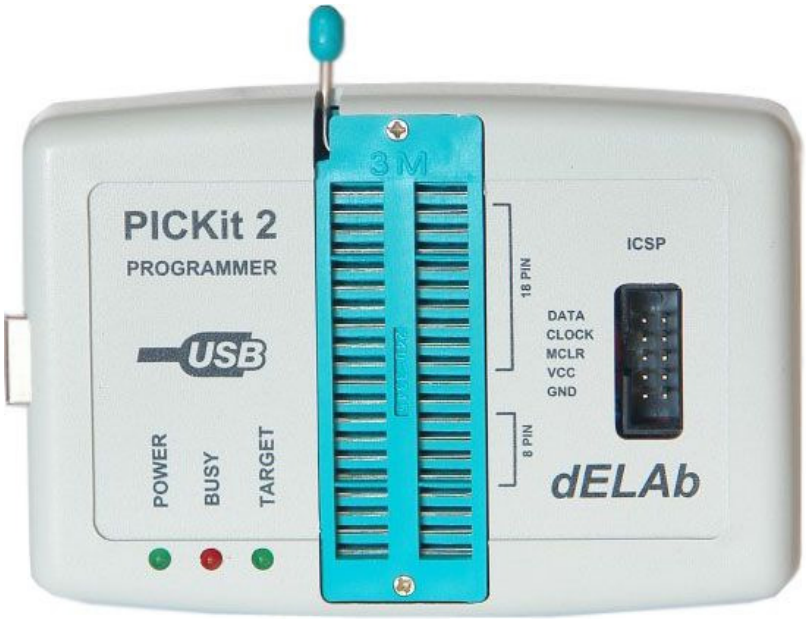
1. Debi
2. İvme
3. Hız-Devir
4. Uzunluk
5. Açı-eğim
6. Kuvvet
7. Basınç
8. Moment
9. Seviye
10. Sıcaklık
11. Nem
12. Konum
13. Kütle
14. Işık
15. Ses
16. Temas
17. Renk
18. Isı Akısı
19. Gaz kaçağı

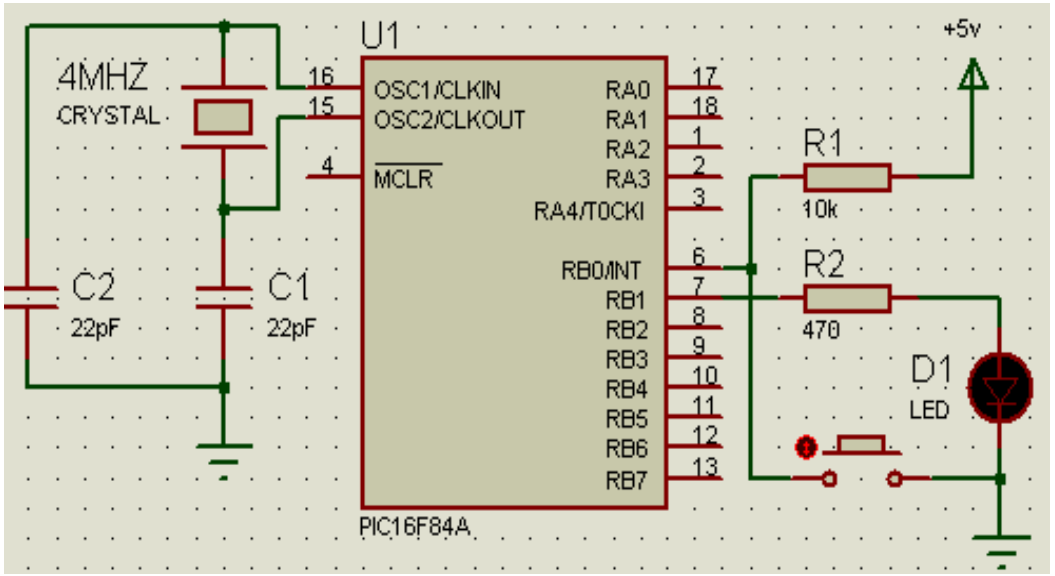
1. Direnç
2. Akım
3. Gerilim
4. Kapasitans
5. Endüktans
6. Frekans

PIC'ler (PERIPHERAL INTERFACE CONTROLLER)







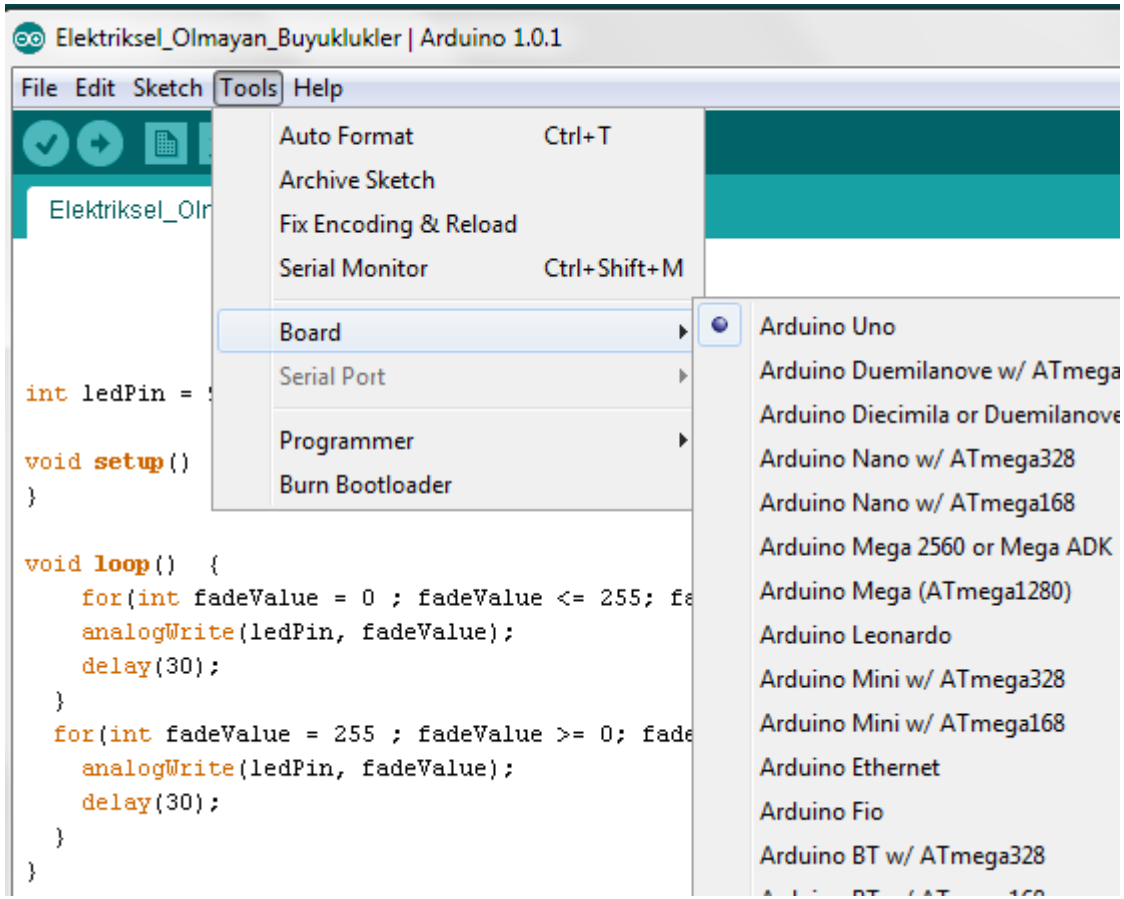


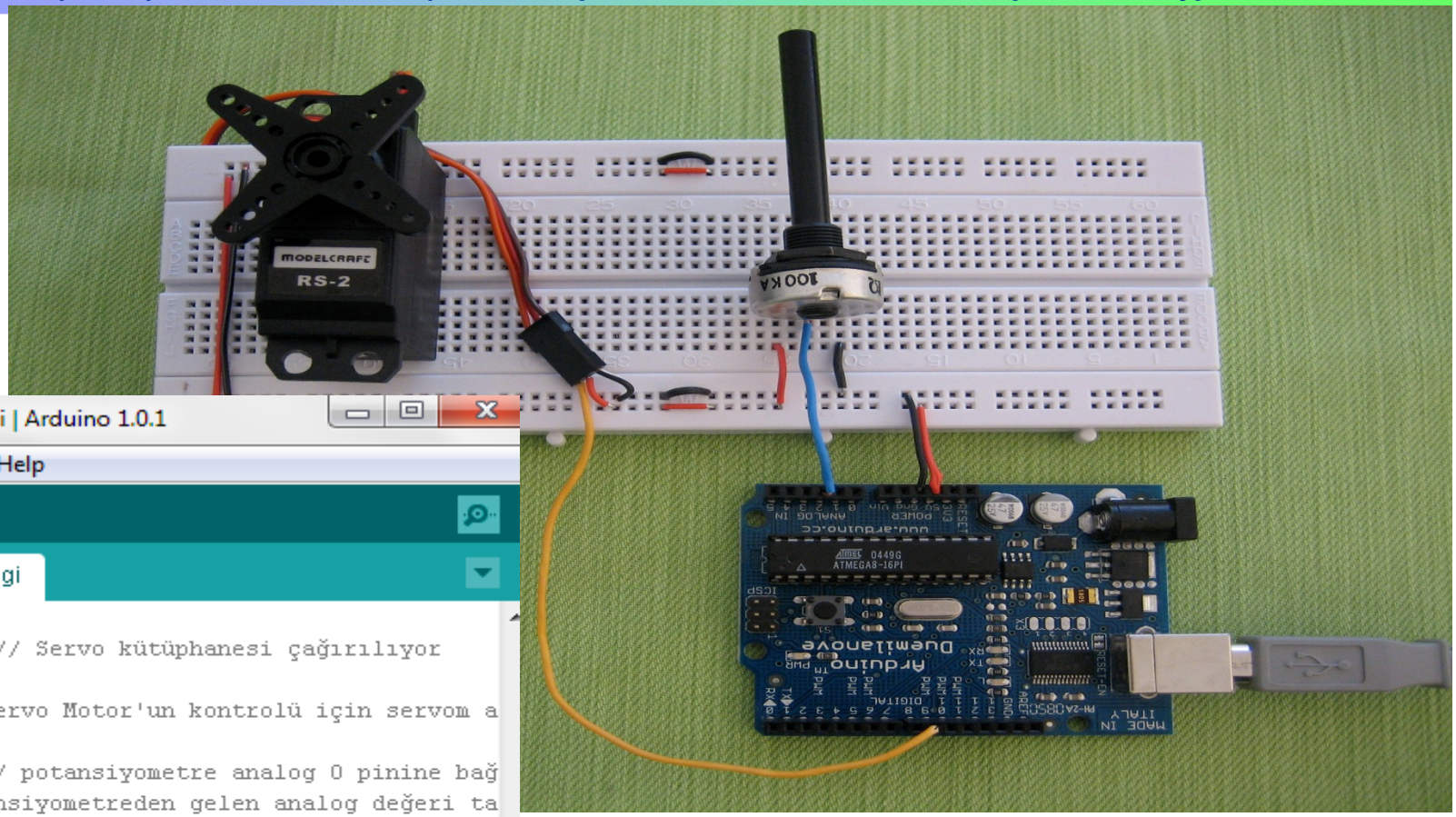
Arduino


MADE
IN ITALY
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN


Sıcaklık 22.2°C
Rutubet 45.5%


MQ-3US


Elektriksel_Olmayan_Buyuklukler | Arduino 1.0.1
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
Arduino Uno
Arduino Duemilanove w/ ATmega
Arduino Diecimila or Duemilanove
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
int ledPin =
void setup()
}
void loop() {
for(int fadeValue = 0 ; fadeValue <= 255; fa
analogWrite(ledPin, fadeValue);
delay(30);
}
for(int fadeValue = 255 ; fadeValue >= 0; fade
analogWrite(ledPin, fadeValue);
delay(30);
}
}

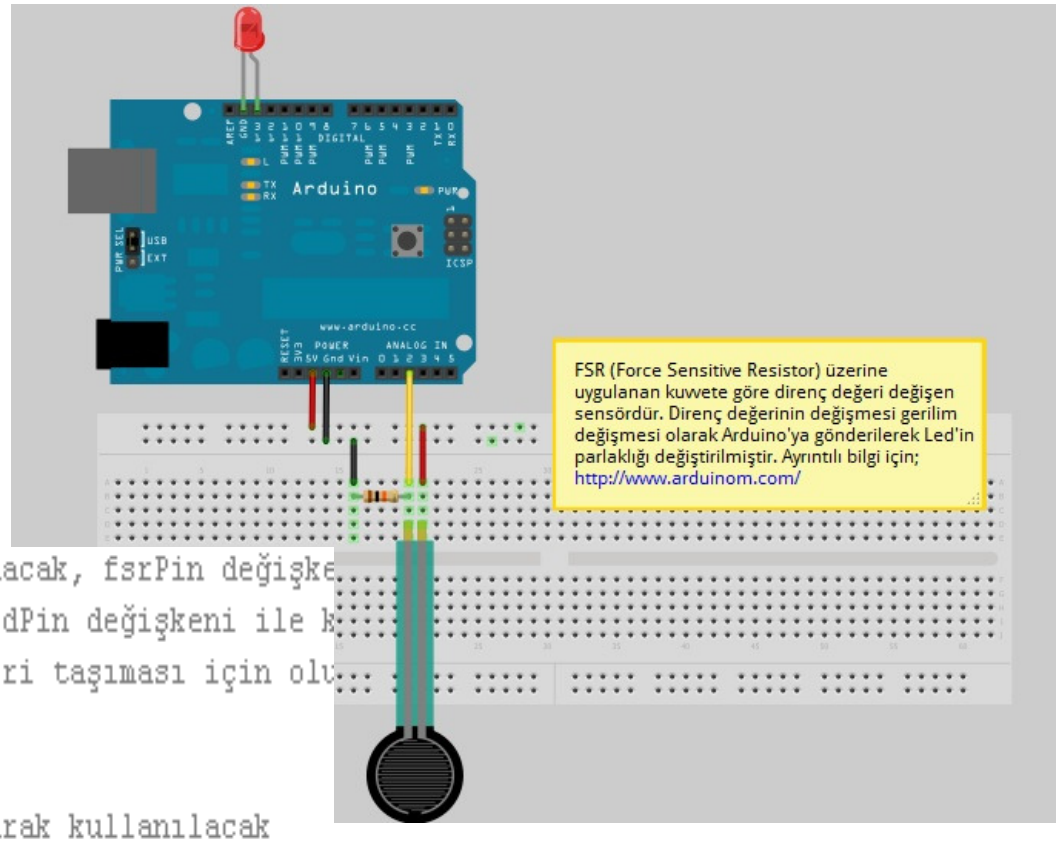


Elektriksel_Ders_Ornegi | Arduino 1.0.1

File Edit Sketch Tools Help

Elektriksel_Ders_Ornegi

```
#include <Servo.h> // Servo kütüphanesi çağırılıyor

Servo servom;  // Servo Motor'un kontrolü için servom a

int potpin = A0;  // potansiyometre analog 0 pinine bağ
int val;    // potansiyometreden gelen analog değeri ta

void setup()
{
  servom.attach(9);  // Servo Motor 9. pine bağlanıyor
}

void loop()
{
  val = analogRead(potpin);            // potansiyometr
  val = map(val, 0, 1023, 0, 179);     // değişkenin sa
  servom.write(val);                   // Servo Motor'u
  delay(15);                           // 15 milisaniye
}
```




```
int fsrPin = A2;     // FSR sensörü 2. pine bağlanacak, fsrPin değişke
int ledPin = 13;    // LED 13. pine bağlanacak, ledPin değişkeni ile k
int val = 0;        // sensörden gelen analog değeri taşıması için olu

void setup() {
  pinMode(ledPin, OUTPUT);  // ledPin'i çıkış olarak kullanılacak
}

void loop() {
  val = analogRead(fsrPin);    // sensörden aldığın değeri val değişk
  digitalWrite(ledPin, HIGH);  // LED'i yak
  delay(val);                  // val değişkeninin değeri kadar bekle
  digitalWrite(ledPin, LOW);   // LED'i söndür
  delay(val);                  // val değişkeninin değeri kadar bekle
}
```

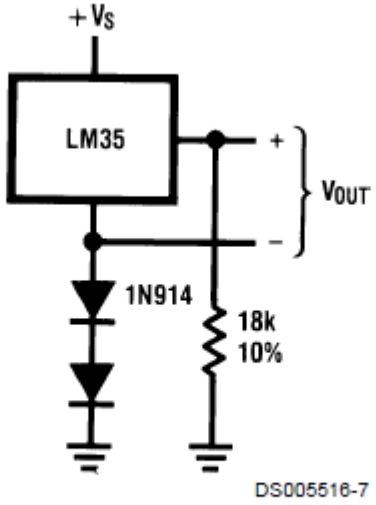
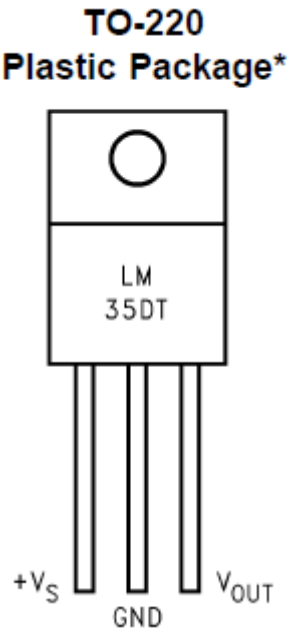
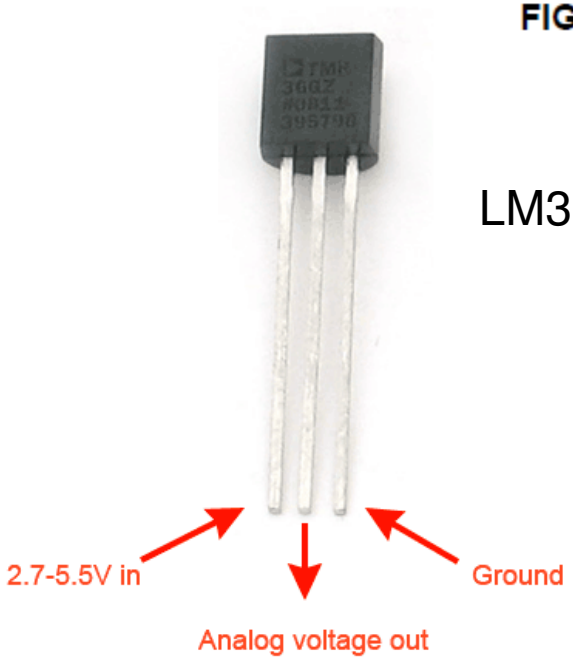


FIGURE 7. Temperature Sensor, Single Supply, −55° to +150°C





LM35 sensörü için: **Sıcaklık °C = (Vout mV cinsinden) / 10**

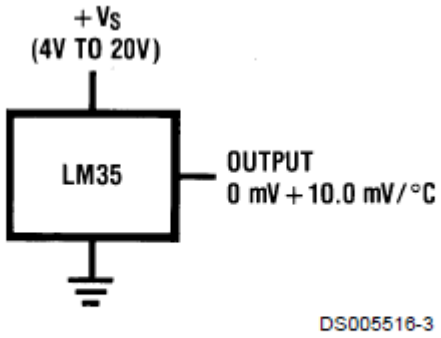


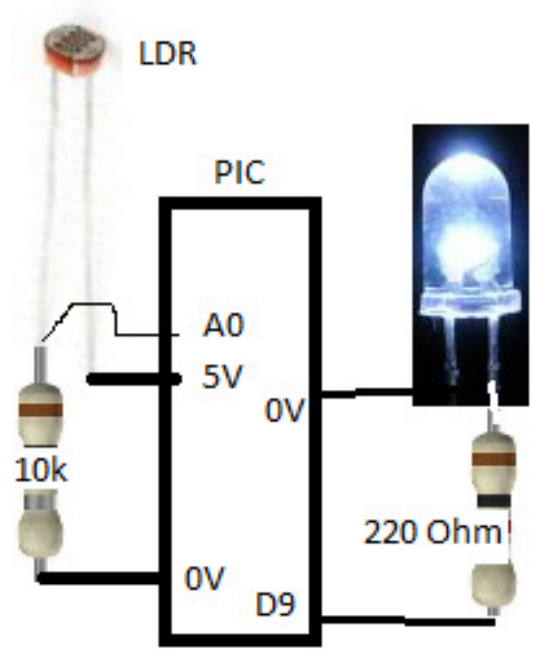
FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

LM35_ders_ornegi §

```
float SensorVerisi;
int AnalogGiris = 0;

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

void loop()
{
  SensorVerisi = analogRead(AnalogGiris);

// Sensörün datasheet'ine göre "Sıcaklık °C = (Vout mV cinsinden) / 10" formulü ile hesaplanır
// 1V -->1000mV olduğu için, yukarıdaki formüle göre 1000/10=100, yani Volt*100 olacaktır.
// analog girişten - sensörden okunan gerilim 10 bitlik değer 0..5V için 0...1024 olarak değişkene gelir-okunur
// bunun için 0...5v arası bir değere çevirmek için bu okunan değer 5/1024 ile çarpılarak dönüşüm yapılır

SensorVerisi = (SensorVerisi * 1000.0/10.0)*5.0/1024.0; // analog veri Sıcaklığa dönüştürülür
Serial.print("SICAKLIK -->" ); Serial.print( (byte)SensorVerisi); Serial.println(" C" ); //PC de görüntüle
delay(1000);                          // yeni veriyi almadan önce 1 saniye bekle
}
```

Ders_ornegi_fotoselIli_fader §

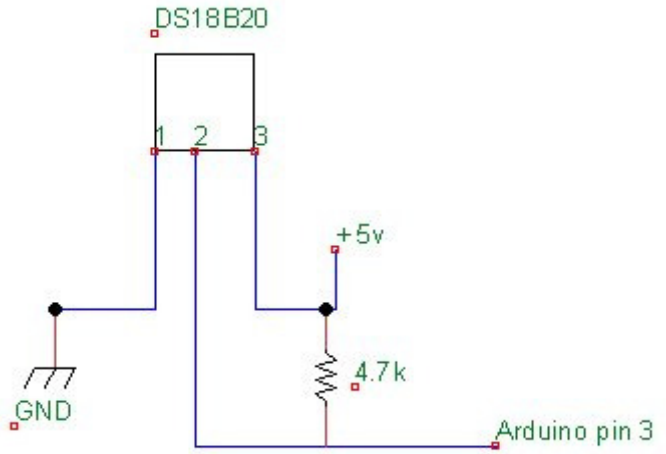
```
int ledPin = 9;
int i = 0;
float analog;

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

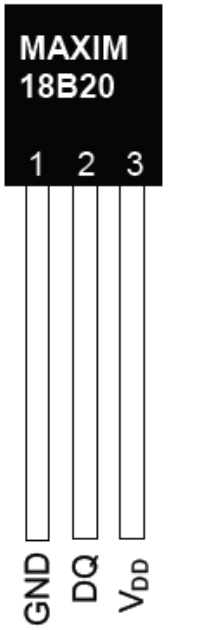



```
void loop()
{
 analog= analogRead(A0);
 analog=analog*5/1024;
if (analog < 3.5) //LDR den gelen değeri bundan küçükse
  {
    for (i = 0; i<=255; i+=1)
    {
        analogWrite(ledPin, i);
        delay(5);
    }
    delay(1000);

    for (i = 255; i>=0; i-=1)
    {
       analogWrite(ledPin, i);
       delay(5);
    }
    delay(1000);
  }
else
  {
  digitalWrite(ledPin,HIGH);
  delay(500);
  digitalWrite(ledPin,LOW);
  delay(500);
  }
}
```

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)

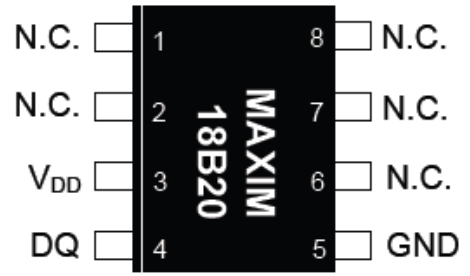
Datasheet





(BOTTOM VIEW)

TO-92
(DS18B20)



SO (150 mils)
(DS18B20Z)

µSOP
(DS18B20U)

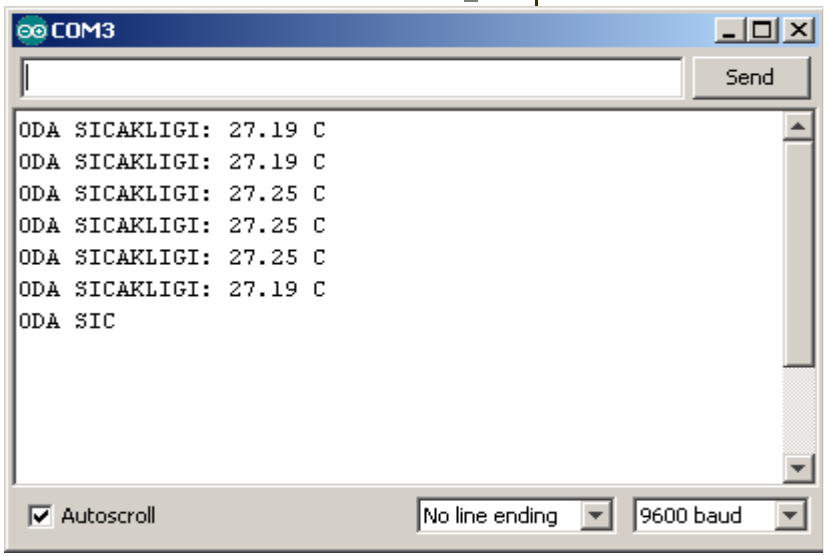
Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)
+125	0000 0111 1101 0000
+85*	0000 0101 0101 0000
+25.0625	0000 0001 1001 0001
+10.125	0000 0000 1010 0010
+0.5	0000 0000 0000 1000
0	0000 0000 0000 0000
-0.5	1111 1111 1111 1000
-10.125	1111 1111 0101 1110
-25.0625	1111 1110 0110 1111
-55	1111 1100 1001 0000

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20

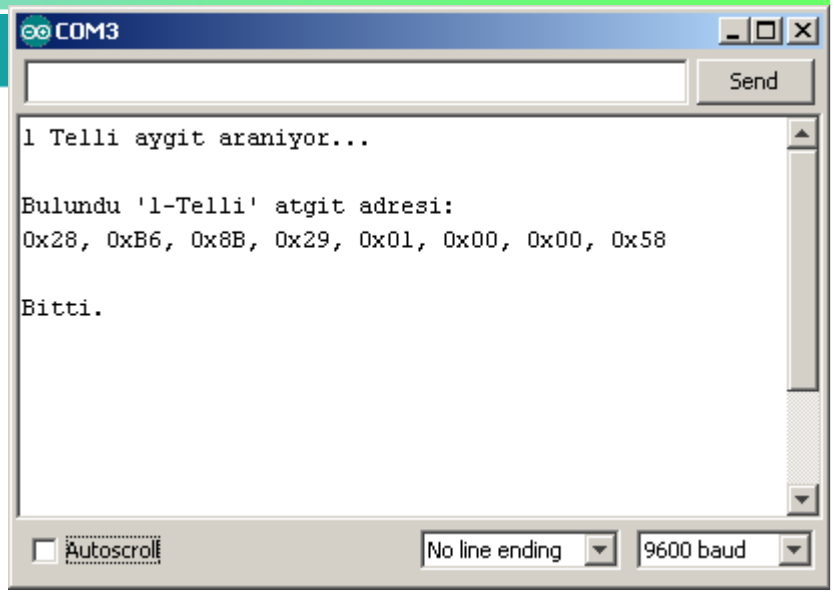
ds1820_ile_sicaklik_basit§

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define Veri_pini 10 // Data ayağı port 10'a bağlı

OneWire Tek_hat(Veri_pini); // tek kablo iletişimi için tanımlama
DallasTemperature sensor(&Tek_hat);//sensor nesnesi tanımlanıyor

void setup(void)
{
  Serial.begin(9600);
  sensor.begin(); // Kütüphaneyi başlat
}

void loop(void)
{
  sensor.requestTemperatures(); // Sıcaklığı okutma komutu
  Serial.print("ODA SICAKLIGI: ");
  Serial.print(sensor.getTempCByIndex(0));
  Serial.println(" C");
 delay(1000);
}
```



ds1820_adres_bulma §

```
#include <OneWire.h>
OneWire  Veri_pini(10);  // Pin 10'a bagla datayi

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

void AdresiBul(void) {
  byte i;  byte Adres[8];
  Serial.print("1 Telli aygit araniyor...\n\r");
  while(Veri_pini.search(Adres)) {
    Serial.print("\n\rBulundu \'1-Telli\' atgit adresi:\n\r");
    for( i = 0; i < 8; i++) { Serial.print("0x");
      if (Adres[i] < 16) {Serial.print('0');}
      Serial.print(Adres[i], HEX);
      if (i < 7) {Serial.print(", ");}
    } //for sonu
    if ( OneWire::crc8( Adres, 7) != Adres[7]) {
        Serial.print("Adres mevcut degil!\n"); return;}
  }
  Serial.print("\n\r\n\rBitti.\r\n");
  Veri_pini.reset_search(); return;
} //AdreiBul procedürün sonu

void loop(void) {
/*burasi kullanilmiyo */ }
```

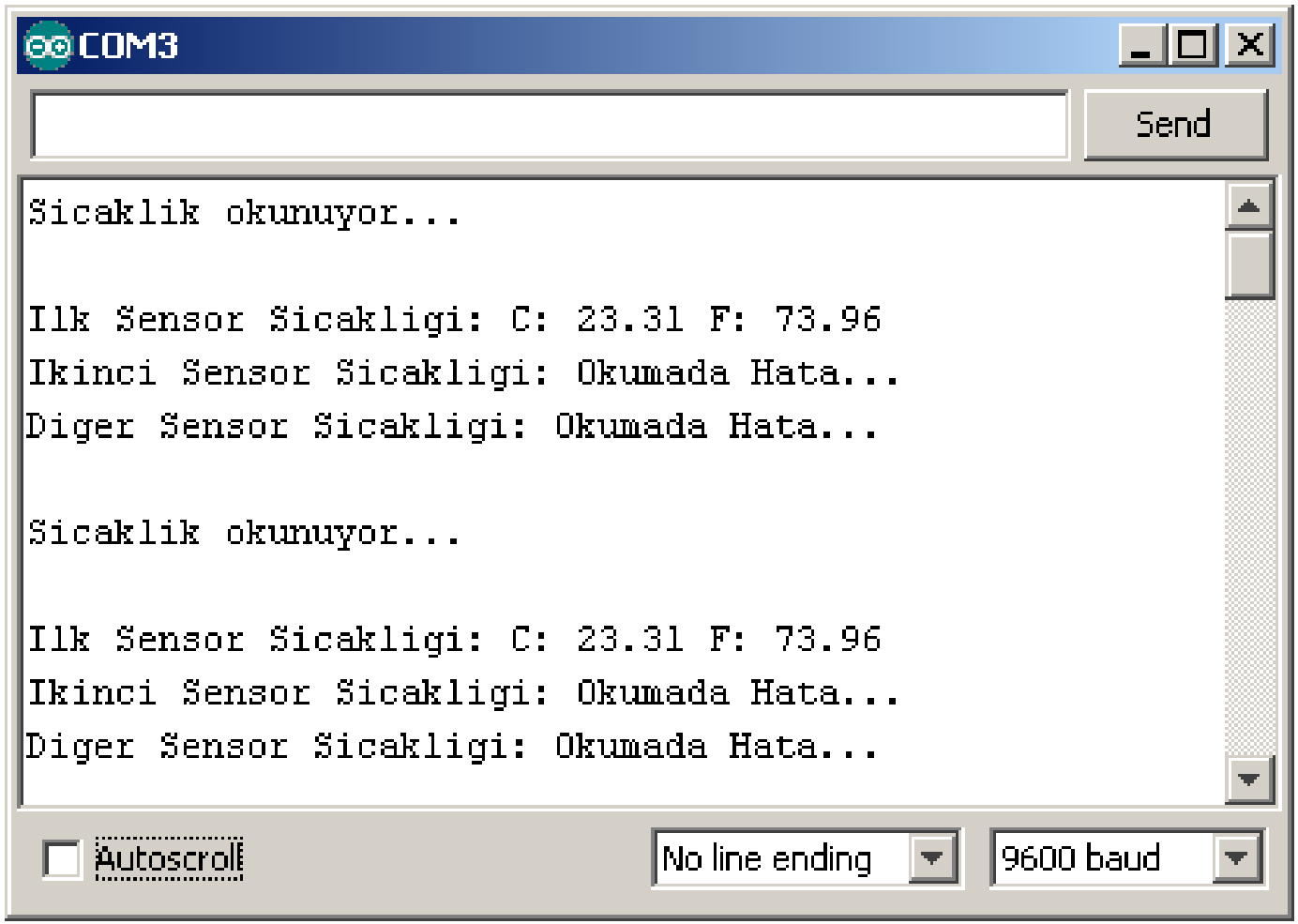


Aynı hatta birden fazla sensor

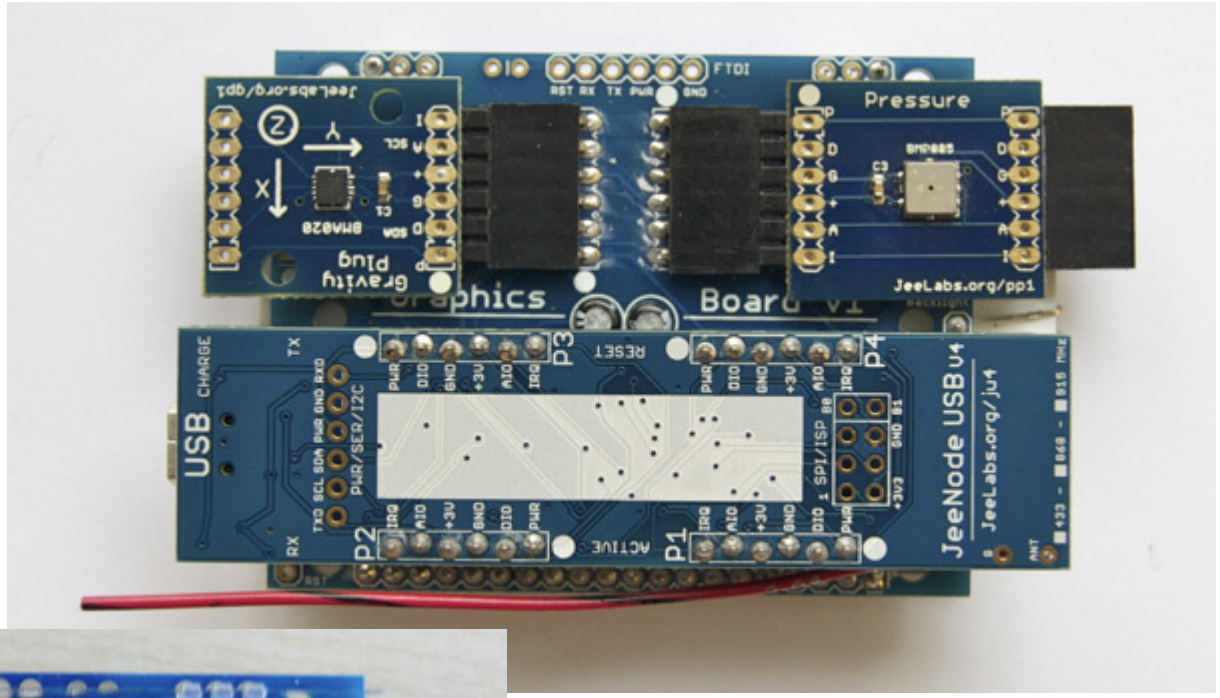
sicaklik_3_sensor_tek_hat§

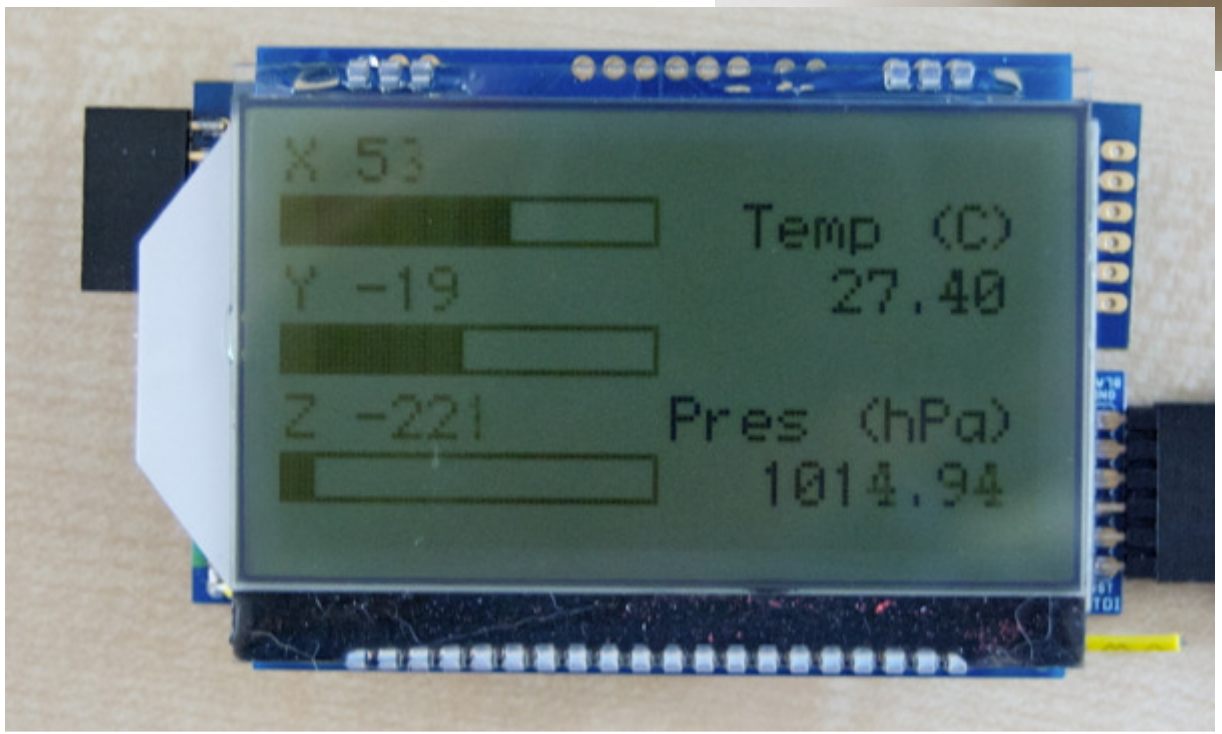
```
#define Veri_Ayagi 10 // Data pini 10'a bagla

OneWire TekHat(Veri_Ayagi);// Tek hat uzerinden iletisim hazirlaniyor
DallasTemperature sensorler(&TekHat);// sensorler nesnesi tanımlanıyor

// Aynı Hatta baglı sensör adresleri bildiriliyor
DeviceAddress Sensor_1 = { 0x28, 0xB6, 0x8B, 0x29, 0x01, 0x00, 0x00, 0x58 };
DeviceAddress Sensor_2 = { 0x28, 0x6B, 0xDF, 0xDF, 0x02, 0x00, 0x00, 0xC0 };
DeviceAddress DigerSensor = { 0x28, 0x59, 0xBE, 0xDF, 0x02, 0x00, 0x00, 0x9F };

void setup(void)
{
Serial.begin(9600);
sensorler.begin(); // Sensör kütüphanesini başlatıyor
// sensorlerin (9-12 bit arası) çözüunürlük değeri ayarlanıyor
sensorler.setResolution(Sensor_1, 12);
sensorler.setResolution(Sensor_2, 10);
sensorler.setResolution(DigerSensor, 10);
}
```

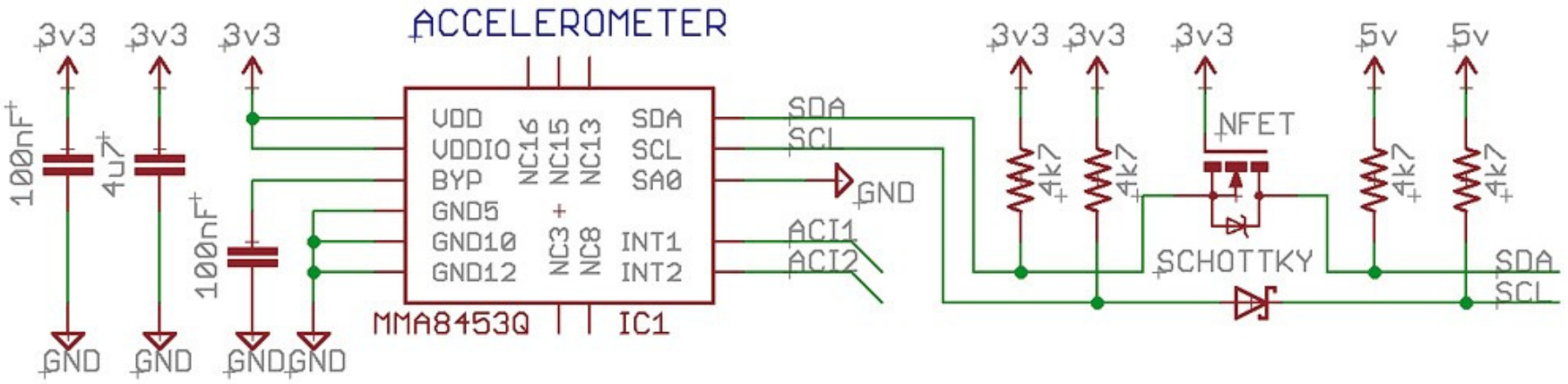
```
void SicakligiYazdir(DeviceAddress cihazAdresi)
{
float tempC = sensorler.getTempC(cihazAdresi);
if (tempC == -127.00) {
Serial.println("Okumada Hata...");
} else { Serial.print("C: "); Serial.print(tempC);
Serial.print(" F: ");
// float fahren = tempC*1.8 + 32.0; Serial.print(fahren);
Serial.println(DallasTemperature::toFahrenheit(tempC));}}

void loop(void) {
Serial.println("Sicaklik okunuyor...\n");
sensorler.requestTemperatures();

Serial.print("Ilk Sensor Sicakligi: ");SicakligiYazdir(Sensor_1);
Serial.print("Ikinci Sensor Sicakligi: ");SicakligiYazdir(Sensor_2);
Serial.print("Diger Sensor Sicakligi: ");SicakligiYazdir(DigerSensor);
Serial.print("\n"); delay(2000);}
```


COM3
Send
Sicaklik okunuyor...
Ilk Sensor Sicakligi: C: 23.31 F: 73.96
Ikinci Sensor Sicakligi: Okumada Hata...
Diger Sensor Sicakligi: Okumada Hata...
Sicaklik okunuyor...
Ilk Sensor Sicakligi: C: 23.31 F: 73.96
Ikinci Sensor Sicakligi: Okumada Hata...
Diger Sensor Sicakligi: Okumada Hata...
Autoscroll
No line ending
9600 baud

Gravity + Pressure + GLCD

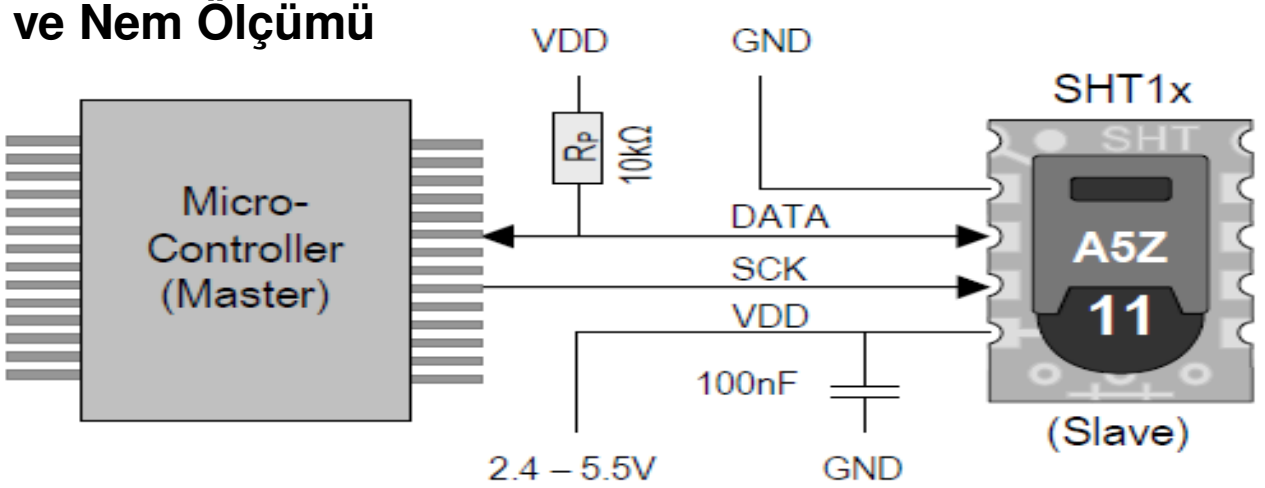







```
#include <I2C.h>
#include <MMA8453_n0m1.h>

MMA8453_n0m1 accel;

accel.dataMode(true, 2);


accel.update();

Serial.println(accel.x());
Serial.println(accel.y());
Serial.println(accel.z());
```

Sıcaklık ve Nem Ölçümü



Temperature

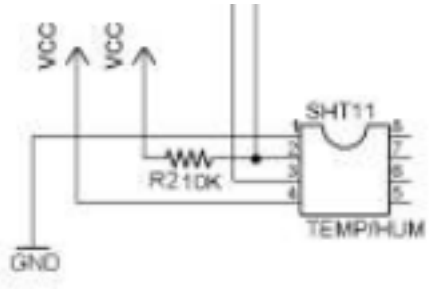
Parameter	Condition	min	typ	max	Units
Resolution [1]		0.04	0.01	0.01	°C
		12	14	14	bit
Accuracy [2] SHT10	typical		±0.5		°C
	maximal	see Figure 3			
Accuracy [2] SHT11	typical		±0.4		°C
	maximal	see Figure 3			
Accuracy [2] SHT15	typical		±0.3		°C
	maximal	see Figure 3			
Repeatability			±0.1		°C
Operating Range		-40		123.8	°C
		-40		254.9	°F
Response Time [6]	τ (63%)	5		30	s
Long term drift			< 0.04		°C/yr

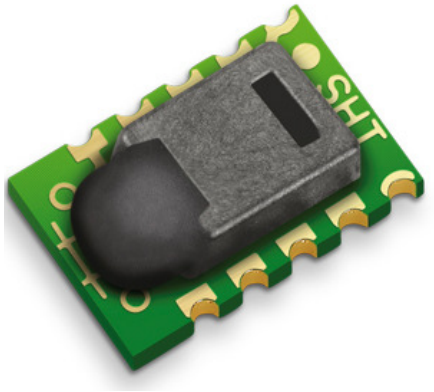
Relative Humidity

Parameter	Condition	min	typ	max	Units
Resolution [1]		0.4	0.05	0.05	%RH
		8	12	12	bit
Accuracy [2] SHT10	typical		±4.5		%RH
	maximal	see Figure 2			
Accuracy [2] SHT11	typical		±3.0		%RH
	maximal	see Figure 2			
Accuracy [2] SHT15	typical		±2.0		%RH
	maximal	see Figure 2			
Repeatability			±0.1		%RH
Hysteresis			±1		%RH
Non-linearity	linearized		<<1		%RH
Response time [3]	τ (63%)		8		s
Operating Range		0		100	%RH
Long term drift [4]	normal		< 0.5		%RH/yr

SHT11_ders_ornegi_ino §

Sıcaklık ve Nem Ölçümü






```
#include <SHTlx.h>
#define veriPini  10
#define saatPini 11
SHTlx shtlx(veriPini, saatPini);
void setup()
{
   Serial.begin(9600);
   Serial.println("Ders Orneii SHT11");
   Serial.println("Veri Okuma 2 saniyede bir ");
 }

void loop()
{
  float temp;
   float nem;
  temp = shtlx.readTemperatureC();
  nem = shtlx.readHumidity();

  Serial.print("SICAKLIK: "); Serial.print(temp ); Serial.print("C -");
  Serial.print(" NEM: "); Serial.print(nem);Serial.println("%");
  delay(2000);
}
```

2 Interface Specifications

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial Data, bidirectional
3	SCK	Serial Clock, input only
4	VDD	Source Voltage
NC	NC	Must be left unconnected

Table 1: SHT1x pin assignment, NC remain floating.

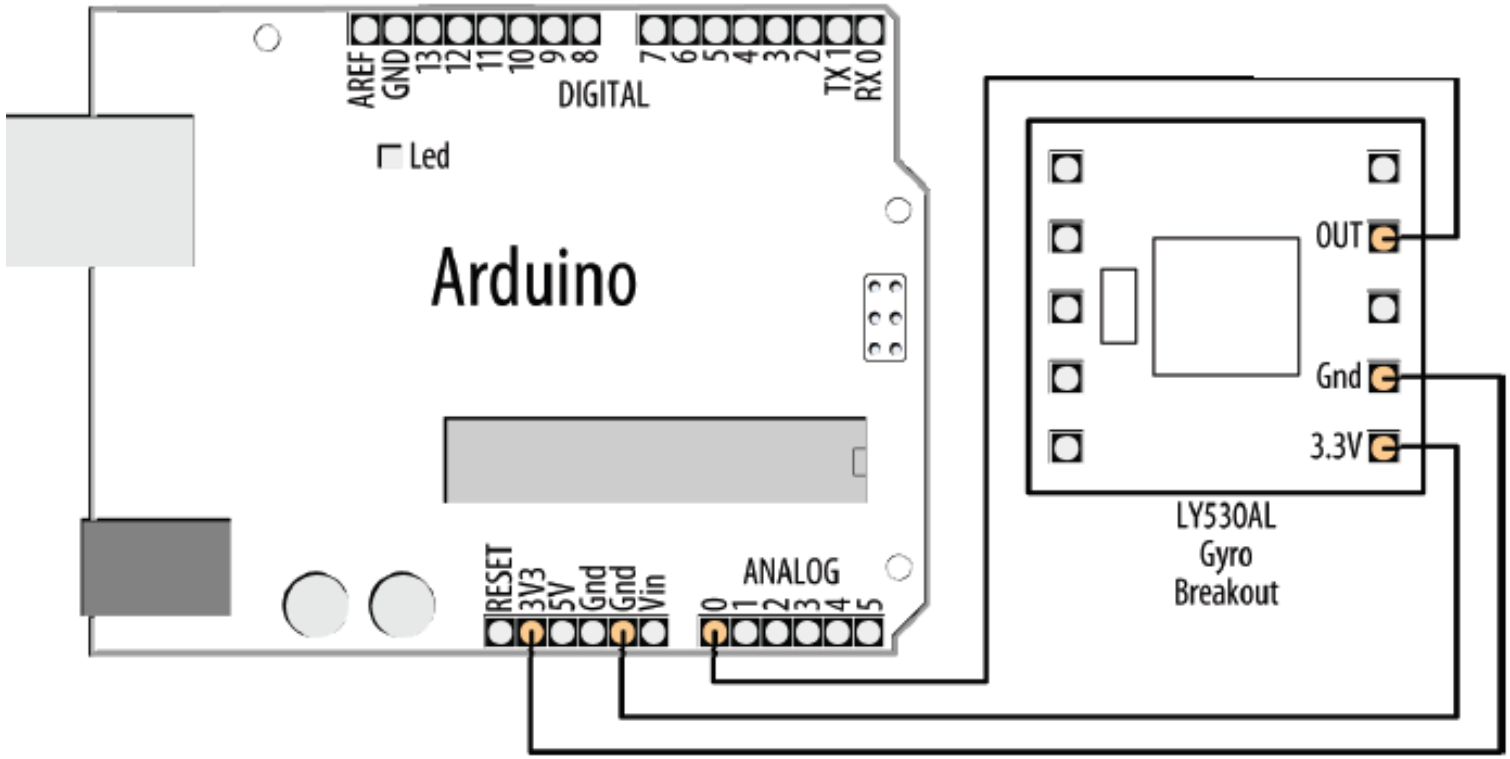
```
//--------------------------------- histerezis yaratacak kodlar ------
if (temp >=30) {
 digitalWrite(Sicaklik_Rolesi,HIGH); Histerezis_T=1;
}
if (temp <= 25 && Histerezis_T == 1){
 digitalWrite(Sicaklik_Rolesi,LOW); Histerezis_T=0;
}
//---------------------------------
```

Ders_ornegi_Jiroskop§

Jiroskop ile dönmeyi görüntüleme

```
//Dönme oranını serial monitörde görüntüler
const int inputPin = 0;
int rotationRate = 0;
void setup()
{
Serial.begin(9600);
}
void loop()
{
rotationRate = analogRead(inputPin);
Serial.print("Dönme orani: ");
Serial.println(rotationRate);
delay(100); // Bir sonraki okuma için 100ms bekle
}
```

- 2.7 to 3.6VDC power supply
- Yaw rotation sensing
- 1x and 4x amplified outputs
- Low power consumption
- All necessary filtering components included
- Access to power-down, self-test, and high-pass filter reset pins

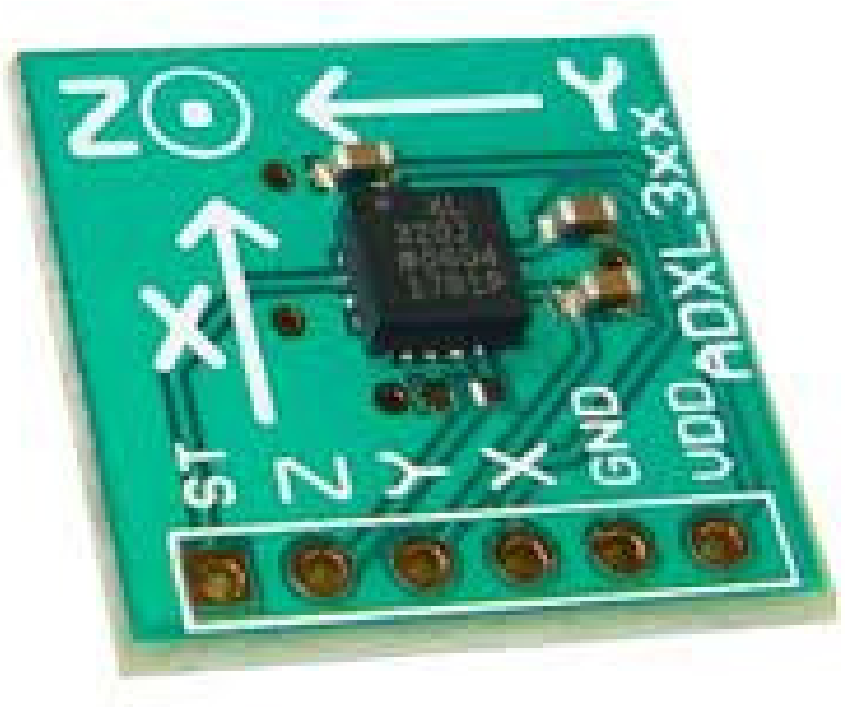


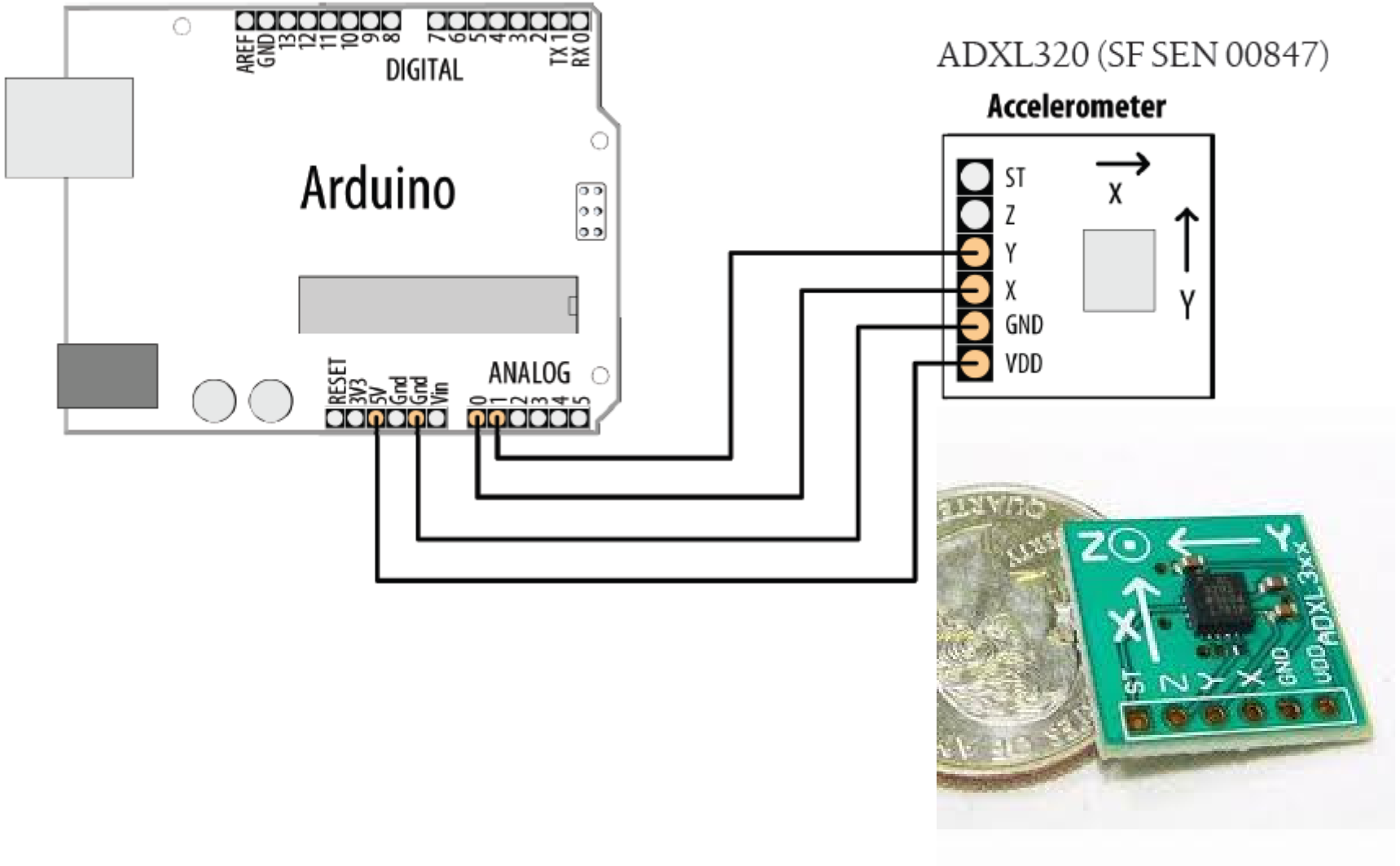
LY530AL gyro connected using 3.3V pin

ders_ornegi_ivme_olcme

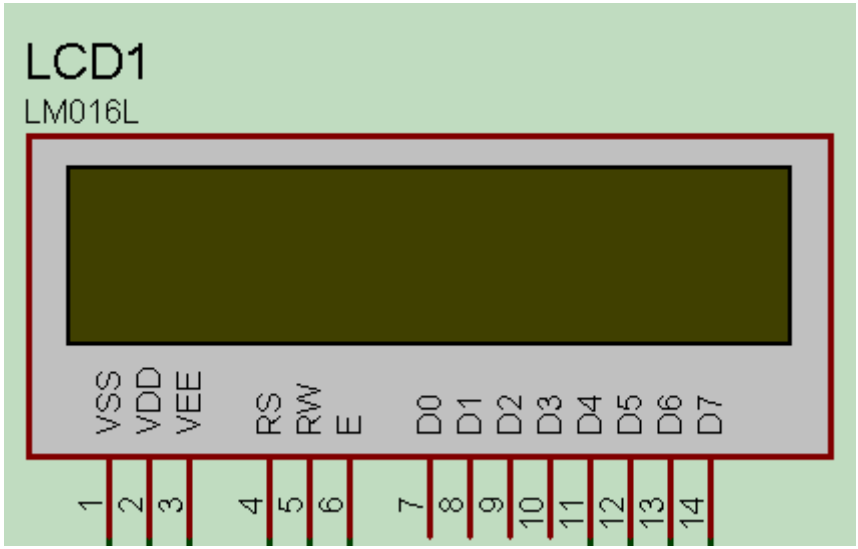
İvme ölçer ile İvme Ölçme

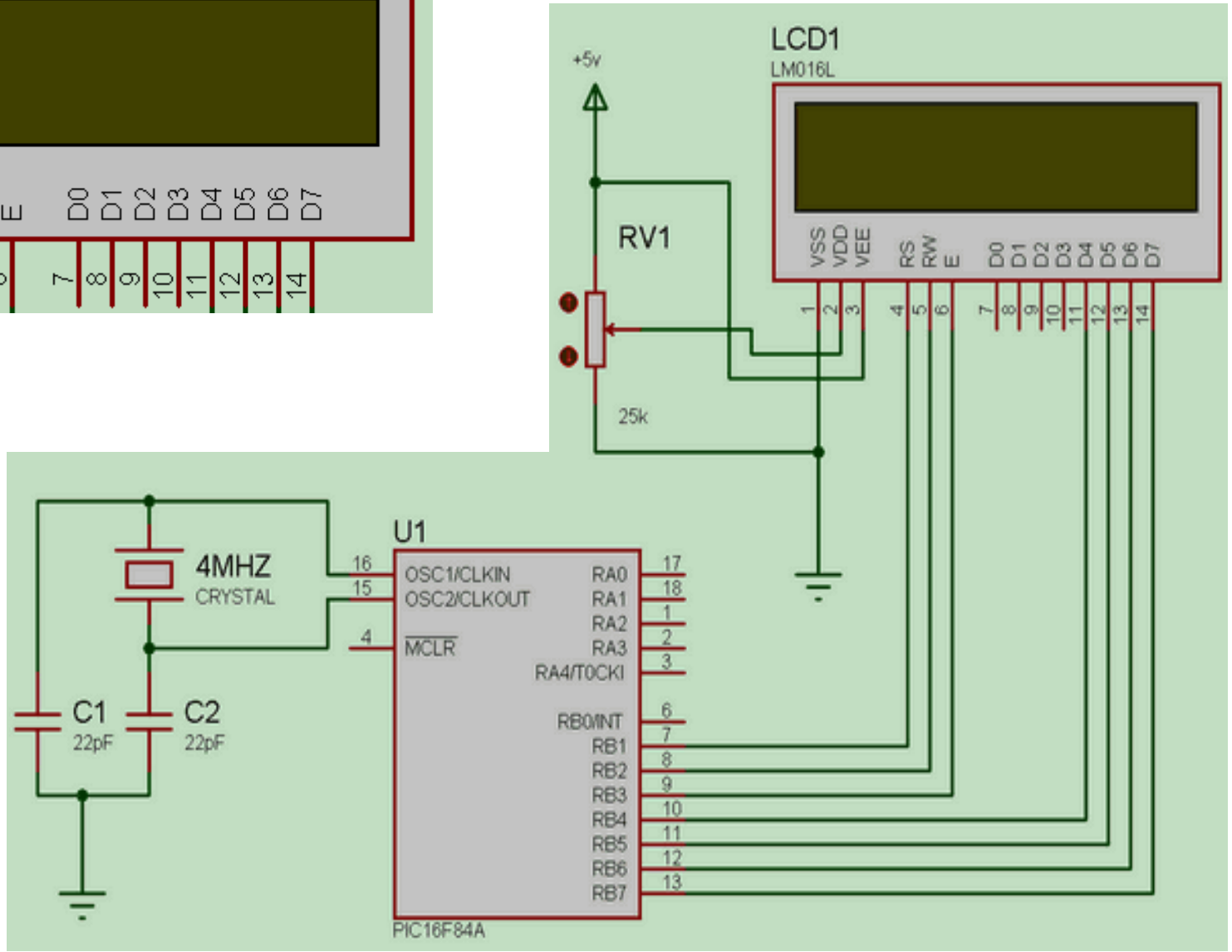
```
const int xPin = 0; // analog input pinleri
const int yPin = 1;
void setup()
{
Serial.begin(9600);
}
void loop()
{
int xValue;
int yValue;
xValue = analogRead(xPin);
yValue = analogRead(yPin);
Serial.print("X value = ");
Serial.println(xValue);
Serial.print("Y value = ");
Serial.println(yValue);
delay(100);
}
```

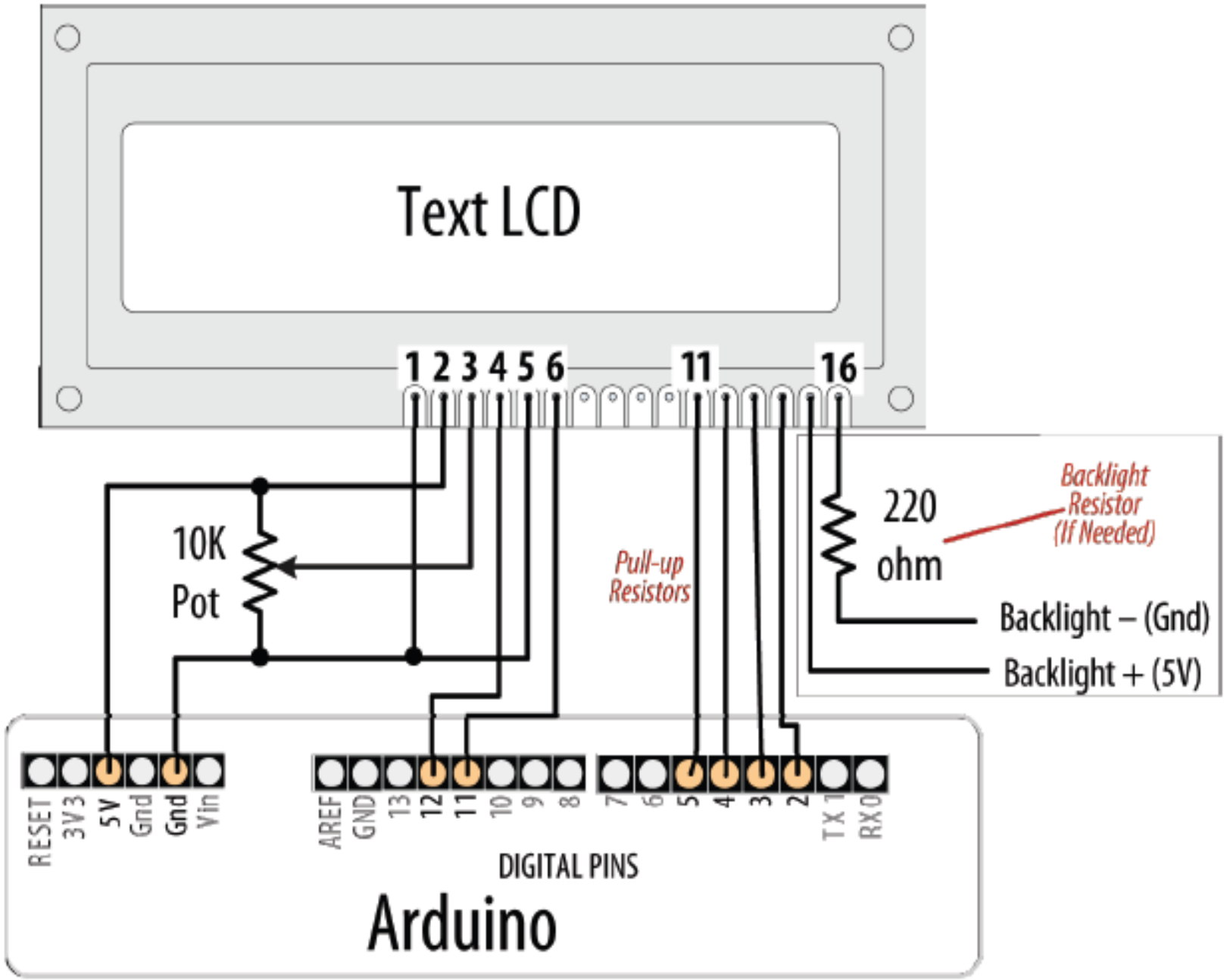



AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX 1
RX 0
DIGITAL
Arduino
RESET
3V3
5V
Gnd
Gnd
Vin
ANALOG
0
1
2
3
4
5
ADXL320 (SF SEN 00847)
Accelerometer
ST
Z
Y
X
GND
VDD
X
Y
Z
X
Y
ST
Z
Y
X
GND
VDD
ADXL 3xx

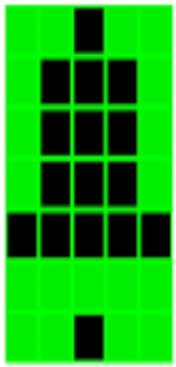
LCD Kullanımı


LCD1
LM016L
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14


LCD1
LM016L
+5v
RV1
25k
U1
4MHZ
CRYSTAL
C1
22pF
C2
22pF
OSC1/CLKIN
OSC2/CLKOUT
MCLR
RA0
RA1
RA2
RA3
RA4/T0CKI
RB0/INT
RB1
RB2
RB3
RB4
RB5
RB6
RB7
PIC16F84A


Text LCD
1 2 3 4 5 6
11
16
10K
Pot
Pull-up
Resistors
220
ohm
Backlight
Resistor
(If Needed)
Backlight – (Gnd)
Backlight + (5V)
RESET
3V3
5V
Gnd
Gnd
Vin
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX 1
RX 0
DIGITAL PINS
Arduino

Custom Pattern	Decimal	Hex
Row 1:	4	0x04
Row 2:	14	0x0E
Row 3:	14	0x0E
Row 4:	14	0x0E
Row 5:	31	0x1F
Row 6:	0	0x00
Row 7:	4	0x04



```
0.karakter $40 4.karakter $60

1.karakter $48 5.karakter $68

2.karakter $50 6.karakter $70

3.karakter $58 7.karakter $78
```

```
void LCD_build(){
        LCD_command(0x48);      //Load the location where we want
to store
        LCD_senddata(0x04);     //Load row 1 data
        LCD_senddata(0x0E);     //Load row 2 data
        LCD_senddata(0x0E);     //Load row 3 data
        LCD_senddata(0x0E);     //Load row 4 data
        LCD_senddata(0x1F);     //Load row 5 data
        LCD_senddata(0x00);     //Load row 6 data
        LCD_senddata(0x04);     //Load row 7 data
        LCD_senddata(0x00);     //Load row 8 data
}
```

Pinout

Pin	Symbol	Level	Function
1	VSS	L	Power Supply 0V (GND)
2	VDD	H	Power Supply +5V
3	VEE	-	Contrast adjust. (about 0V)
4	RS	H / L	H=Command, L=Data
5	R/W	H / L	H=Read, L=Write
6	E	H	Enable (falling edge)
7	D0	H / L	Display Data, LSB
8	D1	H / L	Display Data
9	D2	H / L	Display Data

Pin	Symbol	Level	Function
10	D3	H / L	Display Data
11	D4 (D0)	H / L	Display Data
12	D5 (D1)	H / L	Display Data
13	D6 (D2)	H / L	Display Data
14	D7 (D3)	H / L	Display Data, MSB
15	-	-	NC (see EA DIP122-5N)
16	-	-	NC (see EA DIP122-5N)
17	A	-	LED B/L+ Resistor required
18	C	-	LED B/L -

Ders_ornegi_LCD1 §

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Selamlar...");
}

void loop() {
  // set the cursor to column 0, line 1
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Function

- LiquidCrystal()
- begin()
- clear()
- home()
- setCursor()
- write()
- print()
- cursor()
- noCursor()
- blink()
- noBlink()
- display()
- noDisplay()
- scrollDisplayLeft()
- scrollDisplayRight()
- autoscroll()
- noAutoscroll()
- leftToRight()
- rightToLeft()
- createChar()

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

lcd.begin(cols, rows)

lcd.clear()

lcd.setCursor(col, row)

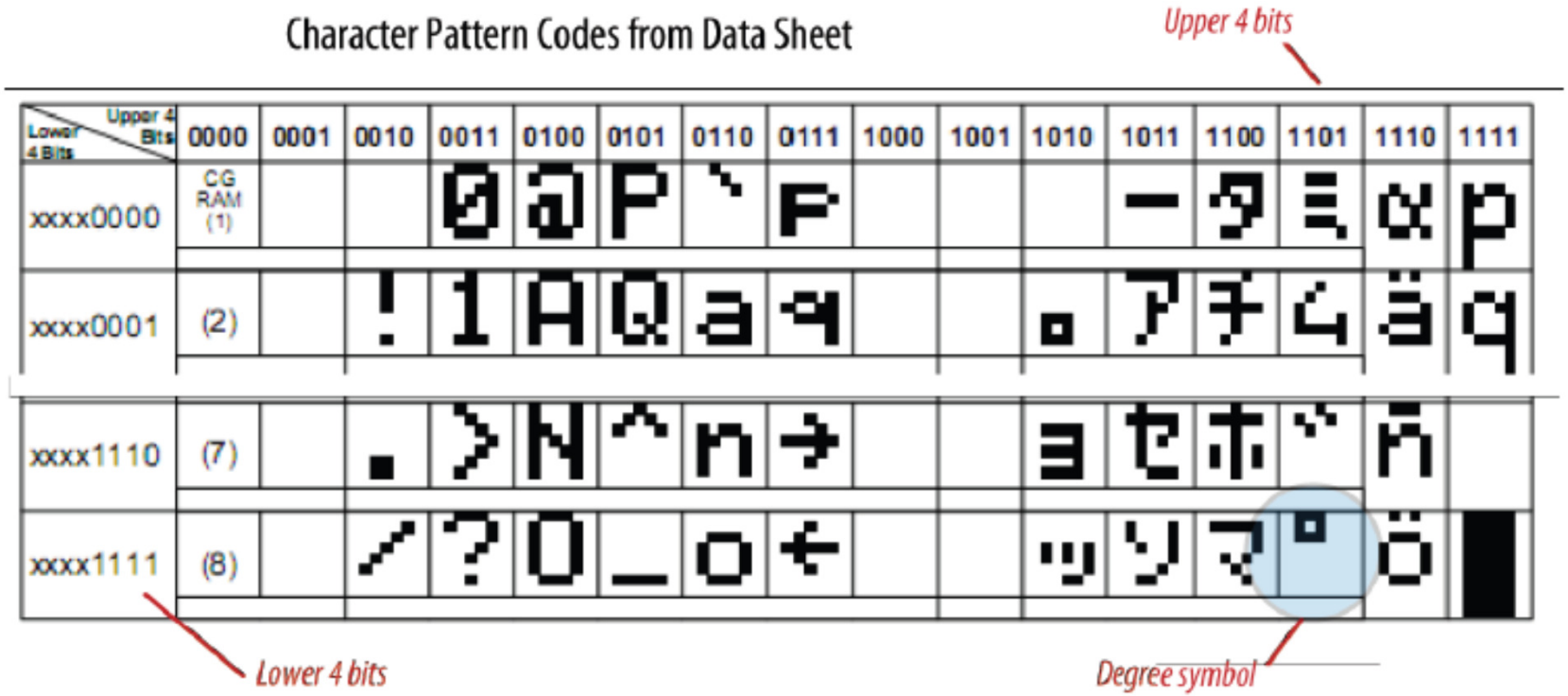
lcd.print(data) data: the data to print (char, byte, int, long, or string

LCD data sheet'inden ilgili karakterin koduna bakıp byte tipinde sembolü tanımlayıp lcd.write() ile ekrana yazdırıyoruz.

```
const byte degreeSymbol = B11011111;
const byte piSymbol     = B11110111;
const byte centsSymbol  = B11101100;
const byte sqrtSymbol   = B11101000;
const byte omegaSymbol  = B11110100;
```

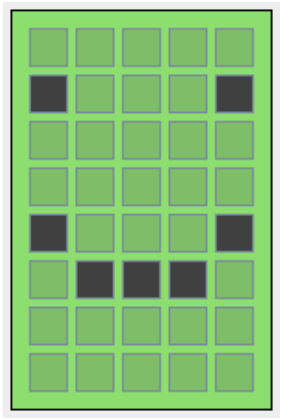
```
lcd.write(symbol);
```

Character Pattern Codes from Data Sheet

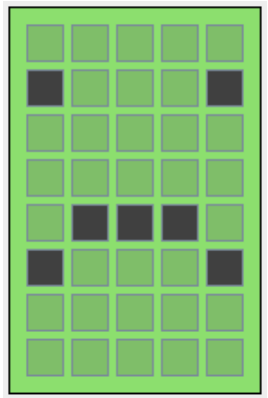


Data sheet'te olmayan özel karakter yazdırma

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte happy[8] =
{
  B00000,
  B10001,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000,
  B00000
};
```

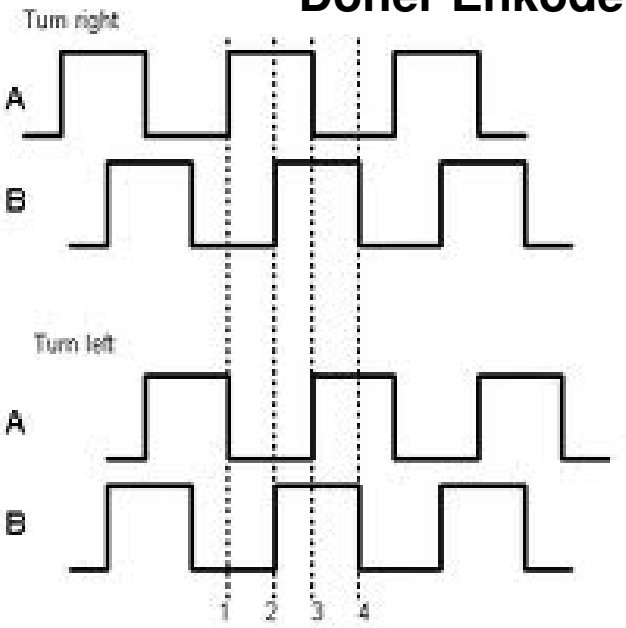


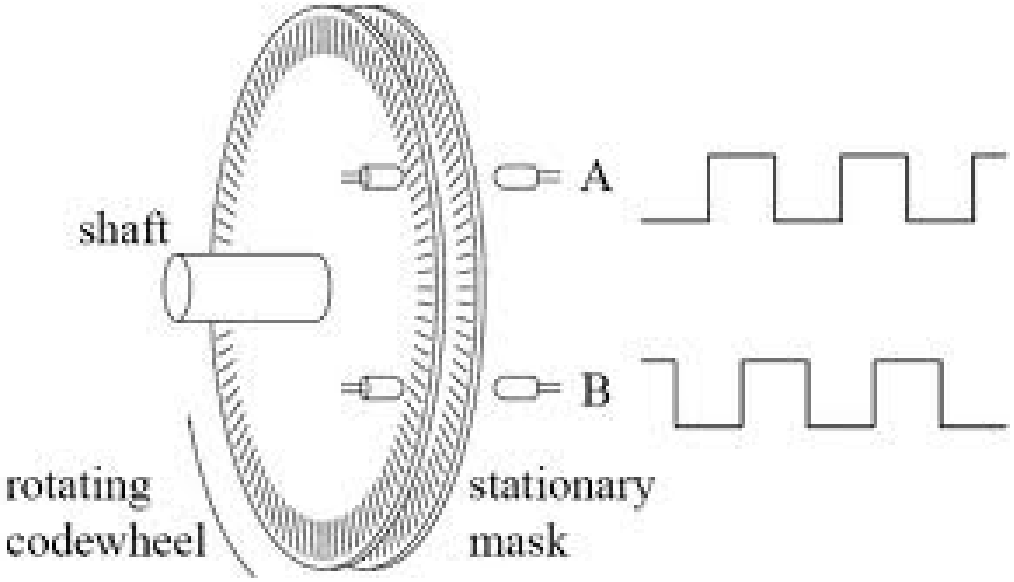
```
byte saddy[8] =
{
  B00000,
  B10001,
  B00000,
  B00000,
  B01110,
  B10001,
  B00000,
  B00000
};
```



```
void setup() {
  lcd.createChar(0, happy);
  lcd.createChar(1, saddy);
  lcd.begin(16, 2);

}

void loop() {
  for (int i=0; i<2; i++)
  {
    lcd.setCursor(0,0);
    lcd.write(i);
    delay(500);
  }
}
```

Döner Enkoder Okuma (Hız ve Yön)

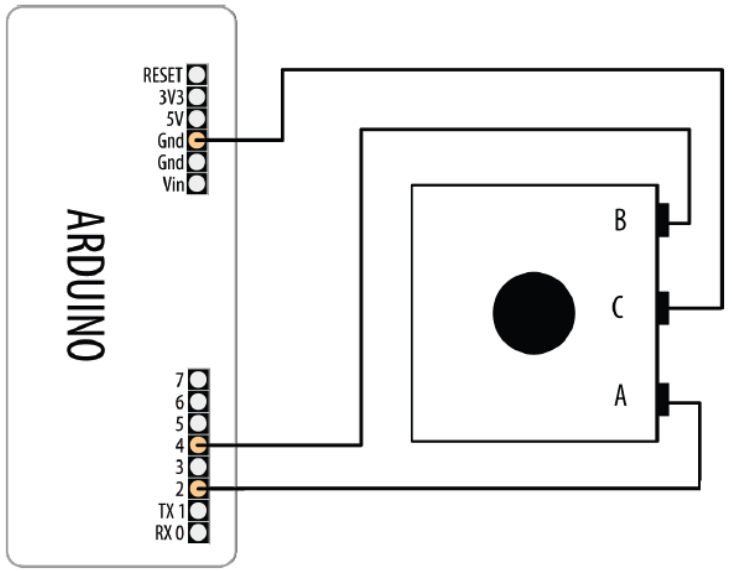


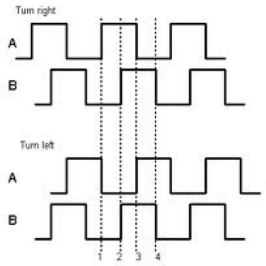


Döner Enkoder Okuma (Hız ve Yön)

Doner_Enkoder_ders_ornegi

```
const int encoderPinA = 4;
const int encoderPinB = 2;
const int encoderStepsPerRevolution=16;
int angle = 0; int val; int encoderPos = 0;
boolean encoderALast = LOW; // remembers the previous pin state
void setup()
{
pinMode(encoderPinA, INPUT);
pinMode(encoderPinB, INPUT);
digitalWrite(encoderPinA, HIGH);
digitalWrite(encoderPinB, HIGH);
Serial.begin (9600);
}
void loop()
{
boolean encoderA = digitalRead(encoderPinA);
if ((encoderALast == HIGH) && (encoderA == LOW))
{
if (digitalRead(encoderPinB) == LOW) {encoderPos--;}else{encoderPos++;}
angle=(encoderPos % encoderStepsPerRevolution)*360/encoderStepsPerRevolution;
Serial.print (encoderPos);
Serial.print (" ");
Serial.println (angle);
}
encoderALast = encoderA;
}
```



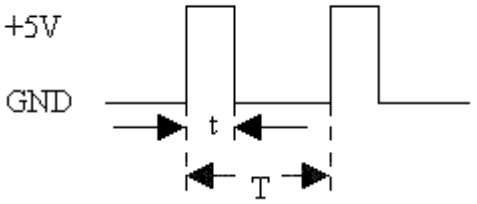


t = 0.9mS
T = 20mS
Angle = 0°

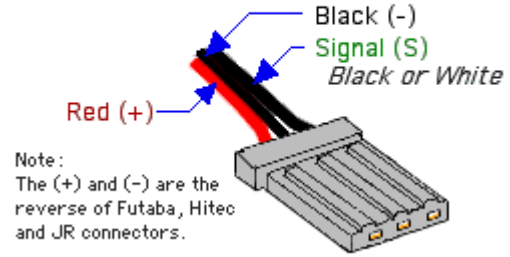
t = 1.5mS
T = 20mS
Angle = 90°

t = 2.1mS
T = 20mS
Angle = 180°

PWM
T = 20mS
t = 0.9 S - -> 2.1 S
1 / 0.02 Saniye= 50Hz

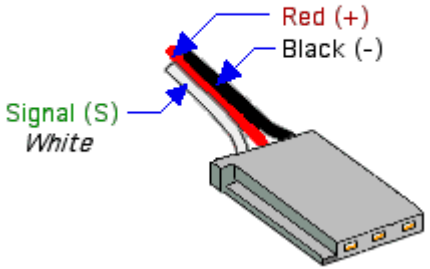


AIRTRONICS

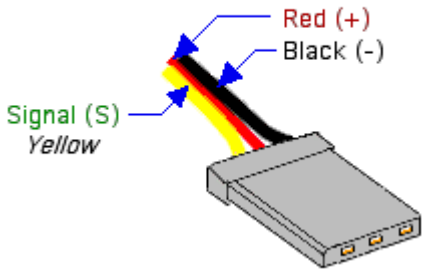


Note:
The (+) and (-) are the reverse of Futaba, Hitec and JR connectors.

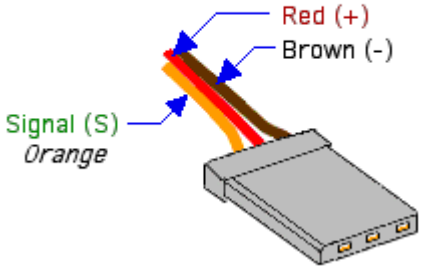
Futaba
"J" Type Connector



hitec



JR Radios

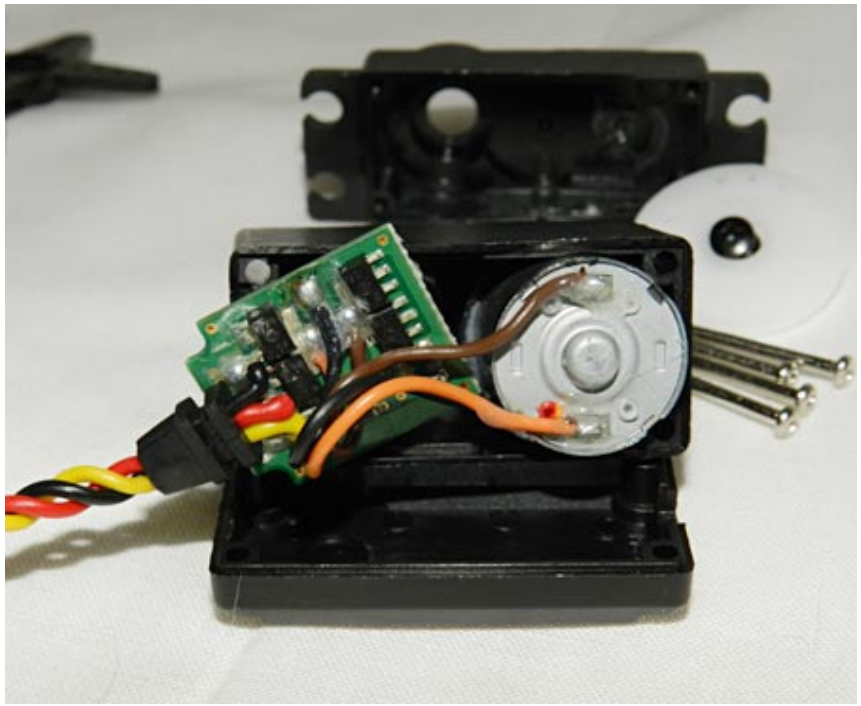


Servo'yu orta konuma getirir

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(9);
  myservo.writeMicroseconds(1500);
}
void loop() {}
```

Önceki sayfadaki servo için
- 900 sola
- 1500 orta
- 2100 sağa

Not: Bu değerler servo tipine bağlı olarak değişebilir.

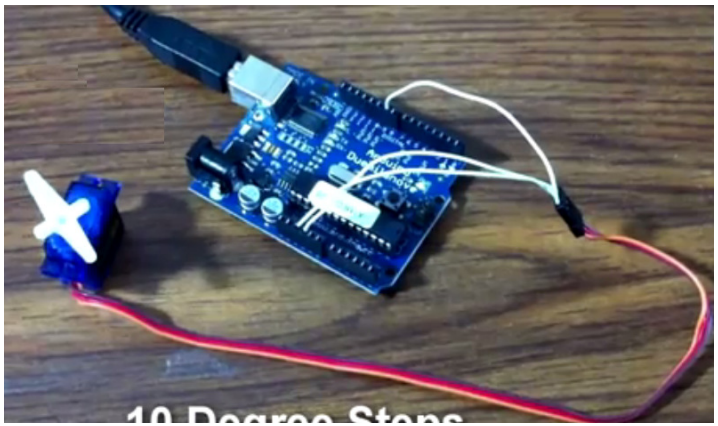


Ders_ornegi_servo1 §

```
#include <Servo.h>
Servo myservo;  // Bir servo kontrol  nesnesi oluşturuluyor
                // maksimum 8 servo nesnesi tanımlanabilir
int pos = 0;    // konumu belirlemede kullanılacak değişken

void setup()
{
  myservo.attach(9);  // aservonun bağlanacak pin tanımlanır
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // 0 dan 180 derece döngüsü
  {                                 // 1 derece artış yapılacak
    myservo.write(pos);             // servoyu pos konumuna getir
    delay(15);      //ilgili konuma gidene kadar  bekletilir
  }
  for(pos = 180; pos>=1; pos-=1)
  {
    myservo.write(pos);
    delay(15);
  }
}
```

Functions

- attach()
- write()
- writeMicroseconds()
- read()
- attached()
- detach()



Ders_ornegi_servo2

```
#include <Servo.h>
Servo myservo;
int potpin = 0;
int val;
void setup(){myservo.attach(9); }

void loop()
{
val = (analogRead(potpin)*3);
// servoya uygun olçeklendirme yapılıyor
val = map(val, 0, 1023, 50, 0);
myservo.write(val);
delay(15);
}
```