**KULLANILAN MALZEMELER**

**a) Arduino UNO**

**Arduino UNO Genel Özellikleri**

♦ Mikrodenetleyici: ATmega328

♦ Besleme Voltajı: 7-12V

♦ Çalışma Voltajı: 5V

♦ 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

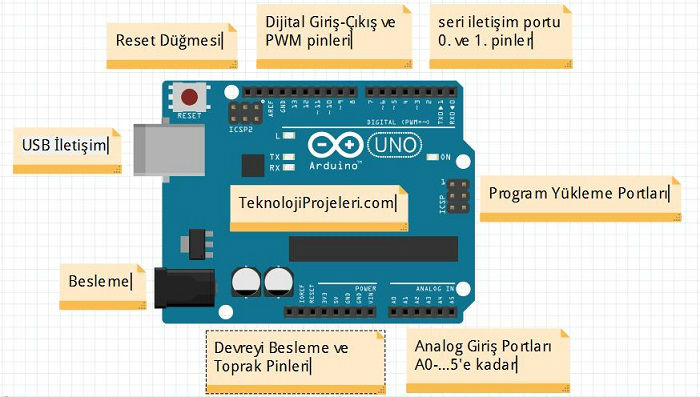
♦ SRAM: 1 KB

♦ EEPROM: 1 KB

♦ Çalışma Saat Hızı: 16MHz

♦ İletişimi de USB üzerinden sağlayabilmektedir.

Arduino pin özelliklerini aşağıdaki çizilen resimden görebilirsiniz. Ayrıca resmin altında detaylı açıklamalarda mevcuttur;



♦ Vin yazan Pin; Arduino normalde USB ile beslenmektedir. Ayrıca bu pin üzerinden de 7-12V vererek besleyebilirsiniz.

♦ 5V yazan Pin; Regülesi sayesinde 5V veren pindir. Böylelikle bu çıkış pininden devrenizi 5V besleyebilirisiniz.

♦ 3.3V yazan Pin; Yine aynı şekilde regüleli 3.3V veren bu çıkış piniyle devrenizi 3.3V besleyebilirsiniz.

♦ GND yazan Pin; Ground kısaltılmasıdır. Yani Toprak pinidir. Devrenizdeki elemanların eksi (-) ucu buraya gelecektir.

♦ IOREF yazan Pin; Arduino kart eklentileri için besleme voltajı seçimini sağlamaktadır.

♦ RX-TX yazan Pinler; Seri iletişim pinleridir. TX pini ile veri gönderebilirsiniz, RX pini ile de verileri alabilirsiniz.

♦ 2 ve 3. Pinler; Bu pinler özellikli bir pindir. Bu pinler ile harici kesinti yapabilirsiniz.

♦ PWM Pinleri; Bu pinler 3,5,6,9,10,11 yani üstünde ~ simgeyi gördüğünüz pinler PWM pinleridir. analogWrite() komutu ile 8 bitlik analog sinyal çıkışı elde edebilirsiniz.

♦SPI Pinleri; 10,11,12,13. Pinlerle SPI kütüphanesini kullanarak, SPI iletişimi yapabilirsiniz.

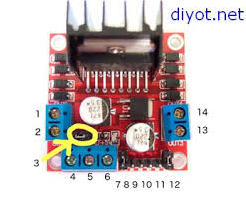
♦TWI veya I2C Pinleri; A4 ve A5 pinleri ile TWI veya I2C olarak bilinen protokolü kullanarak iletişim sağlayabilirsiniz.

♦ AREF Pini; Analog girişlerde referans voltajı olarak kullanılır.

♦ Reset Butonu; Mikrodenetleyiciye reset atmanızı sağlar.

**b) L298N Motor Sürücü Kartı**

L298N motor sürücü kontrol modülü tipik olarak 24V’a kadar olan motorların hız ve yön kontrolünde kullanılmaktadır. İki kanallı bir yapıya sahiptir. Kanal başına 2A akım vermektedir. Kart üzerinde L298N motor sürücü entegresi kullanılmıştır. DC motorlardan ayrı olarak step motor kontrolüne de imkan sağlamaktadır. Bu devre akımı hem polarite hem de PWM ile kontrolünde sürer.



**1)**DC Motor 1 "+" veya step motor "A+"

**2)**DC Motor 1 "-" veya step motor "A-"

**3)**12 V jumper (besleme gerilimi 12 V'dan yüksek olduğunda kaldırılır.)

**4)**Motor beslemesi (maksimum 34 V)

**5)**GND

**6)**12 V jumper takılı olduğunda 5 V çıkış verir. Arduino veya diğer devreler beslenebilir.

**7)**PWM çıkışı ile 1. DC motorda hız kontrolü

**8)**IN1 1. Motor hız belirleme pini, 1. girişi

**9)**IN2 1. Motor hız belirleme pini, 2. girişi

**10)**IN3 2. Motor hız belirleme pini, 1. girişi

**11)**IN4 2. Motor hız belirleme pini, 2. girişi

**12)**PWM çıkışı ile 2. DC motorda hız kontrolü

**13)**DC Motor 2 "+" veya step motor "B+"

**14)**DC Motor 2 "-" veya step motor "B-"

**c) DC Motor**

DC motor, düz akım elektrik enerjisini mekanik enerjiye dönüştüren makinedir. Motorun içinde yer alan sargılara elektrik akımı uygulandığında, yine motorun içerisinde bulunan sabit mıknatıslara zıt yönde oluşan manyetik kuvvetin etkisi ile hareket etme prensibine dayanır. Bu akımın yönünün, sürekli olarak sabit mıknatısa ters manyetik alan oluşturacak şekilde değiştirilmesi gereklidir. Bu değişim, fırçalı motorlarda motorun sarımlarına temas eden fırçalar ile, fırçasız motorlarda ise elektronik hız kontrol devresi tarafından yapılır.

Fırçasız motorların:

Artıları: Yüksek performans, bakım kolaylığı

Eksileri: Sürücü ile kullanma zorunluluğu, maliyet



**d) Servo Motor**

Servo motorlar, ilk olarak uzaktan kumandalı model araçlarda kullanılmıştır. Arabanın direksiyonunu, uçağın kanatlarını ve helikopterlerin swashplate dediğimiz mekanizmasını hareket ettirmede kullanılırlar. Genellikle 180° gibi çalışma açılarına sahiptirler. İstenilen konuma gitmesi ve o herhangi başka bir komut gelmediği sürece o konumda sabit kalması düşünülerek tasarlanmışlardır.

Servo motorlar, aslında içlerinde yine bir fırçalı veya fırçasız DC motor barındırırlar. Motorun dönüş hareketi bu motor ile sağlanır. Çoğunlukla çalışma gelirimleri 5V’tur. Bu motora ek olarak, servo motorların içinde bir dişli mekanizması, motor milinin dönüş miktarını ölçen bir potansiyometre ve motor sürme ve potansiyometreden konum bilgisini almak için bir devre bulunur. Servo içindeki motor hareket ettikçe potansiyometre döner ve kontrol devresi, istenilen konum ile motorun o anki konumunu karşılaştırarak motor sürme işlemini yapar.

Servo motorlar PWM sinyal ile çalışırlar. Bu PWM sinyal, bir model uzaktan kumanda setinden gelebileceği gibi, Arduino veya benzeri mikrokontrolcüler ile de sağlanabilir. PWM ile gelen sinyalin duty cycle değeri 1000 µs ise bu servo motorun 0° konumuna gitmesi anlamına gelir. Bu değer 2000 µs olduğunda ise motor hareket limitinin sınırına (çoğu servo için bu 180°’dir) ulaşmış olur.

Servo motorların:

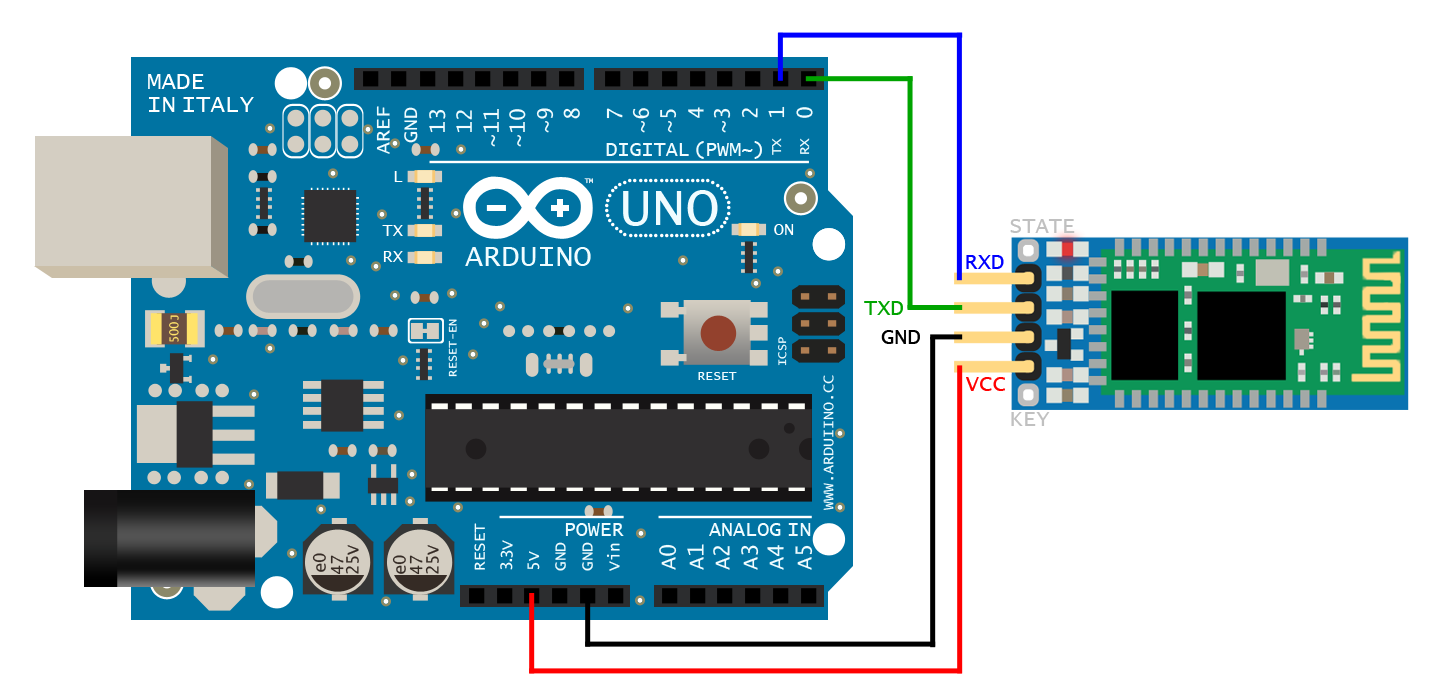
Artıları: hassas pozisyon kontrolü, sürücüye ihtiyaç duymaması

Eksileri: Kısıtlı hareket imkanı



**e) HC-06 Bluetooth Modülü**

Bluetooth SSP(Serial Port Standart) kullanımı ve kablosuz seri haberleşme uygulamaları için yapılmıştır. Üzerinde bulunan pinler yardımı ile kolay kullanım sağlar. Bu kart bluetooth 2.0'ı destekleyen, 2.4GHz frekansında haberleşme yapılmasına sağlar. Açık alanda yaklaşık 10 metre büyüklüğünde bir haberleşme mesafesine sahiptir. Çalışma mantığı ise seri haberleşme esasına dayanır. Tx ucu verici ucudur, Rx ise alıcı ucudur. 3.3V ve 5V arasındaki gerilimlerde çalışabilir. Ancak veri alış verişini 3.3V ile yapmaktadır.



**HC-06 BLUETOOTH KONTROLÜNDE KULLANILAN ANDROİD UYGULAMA**

<https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor>

**KOD**

#include<Servo.h>

int DonmeHizi = 175;

int derece = 80;

int DonmeHizi1 = 125;

int derecea = 150;

int yon = 137;

const int sagileri = 9;

const int saggeri = 8;

const int solileri = 12;

const int solgeri = 13;

const int solenable = 11;

const int sagenable = 10;

const int enable = 6;

const int input1 = 4;

const int input2 = 5;

Servo motorA;

Servo motorB;

void ileri(int hiz)

{

digitalWrite(sagileri,HIGH);

digitalWrite(saggeri,LOW);

digitalWrite(solileri, HIGH);

digitalWrite(solgeri,LOW);

}

void sagaDon(int hiz)

{

analogWrite(sagenable, hiz);

digitalWrite(sagileri,LOW);

digitalWrite(saggeri,HIGH);

analogWrite(solenable, hiz);

digitalWrite(solileri, HIGH);

digitalWrite(solgeri,LOW);

}

void solaDon(int hiz)

{

analogWrite(sagenable, hiz);

digitalWrite(sagileri,HIGH);

digitalWrite(saggeri,LOW);

analogWrite(solenable, hiz);

digitalWrite(solileri, LOW);

digitalWrite(solgeri,HIGH);

}

void geri(int hiz)

{

analogWrite(sagenable, hiz);

digitalWrite(sagileri,LOW);

digitalWrite(saggeri, HIGH);

analogWrite(solenable, hiz);

digitalWrite(solileri, LOW);

digitalWrite(solgeri, HIGH);

}

void M\_Ileri ()

{

digitalWrite(input1,HIGH);

digitalWrite(input2,LOW);

digitalWrite(enable,HIGH);

}

void M\_Geri ()

{

digitalWrite(input1,LOW);

digitalWrite(input2,HIGH);

digitalWrite(enable,HIGH);

}

void M\_Kapat()

{

digitalWrite(input1,LOW);

digitalWrite(input2,LOW);

digitalWrite(enable,LOW);

}

void dur()

{

digitalWrite(sagileri, HIGH);

digitalWrite(saggeri, HIGH);

digitalWrite(solileri, HIGH);

digitalWrite(solgeri, HIGH);

}

void setup()

{

Serial.begin(9600);

motorA.write(150);

motorB.write(137);

motorB.attach(3);

motorA.attach(2);

pinMode(enable,OUTPUT);

pinMode(input1,OUTPUT);

pinMode(input2,OUTPUT);

pinMode(sagileri,OUTPUT);

pinMode(saggeri,OUTPUT);

pinMode(solileri,OUTPUT);

pinMode(solgeri,OUTPUT);

pinMode(sagenable,OUTPUT);

pinMode(solenable,OUTPUT);

}

void loop(){

if (Serial.available() > 0)

{ char tus = (char)Serial.read();

if( tus == 'w' )

ileri(DonmeHizi);

if( tus == 's' )

M\_Kapat();

dur();

if( tus == 'a' )

solaDon(DonmeHizi1);

if( tus == 'd' )

sagaDon(DonmeHizi1);

if( tus == 'x' )

geri(DonmeHizi);

if( tus == 'k' )

motorA.write(derece);

if( tus == 'u' )

motorA.write (derecea);

if( tus == 'l' )

yon = yon + 5;

motorB.write(yon);

if( tus == 'r' )

yon = yon - 5;

motorB.write(yon);

if( tus == 'p' )

M\_Ileri();

if( tus == 'o' )

M\_Geri();

}

}