



## ROBOT TEKNOLOJİSİ

**ÖĞR. GÖR. NECATİ AKSOY**

14. Hafta

# ROBOTİKTE HABERLEŞME

- Bluetooth
- Wi-Fi
- GPS

# BLUETHOOTH

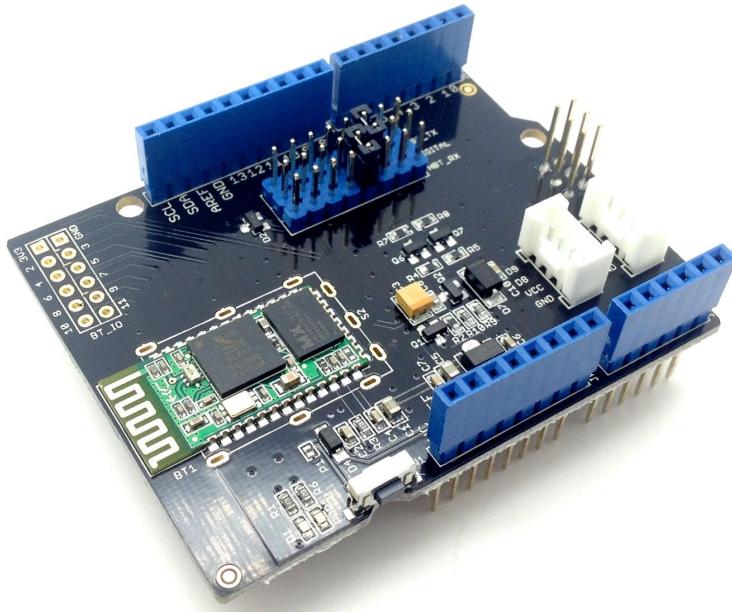
**Bluetooth, yakın mesafelerde işe yarayan bir teknolojidir.** Bluetooth'un sağlıklı bir iletişim alışverişini gerçekleştirmesi için 10 metre uzaklık olması gereklidir.

Eğer iki cihaz arasındaki mesafe yaklaşık 10 metreden fazlaysa Bluetooth ile veri ve bilgi paylaşımı yapmak mümkün olmuyor. Ancak farklı teknolojilerle desteklenen teknolojiler (örnek beacon teknolojisi) ile bu mesafe arttırılabilir.

Kablo bağlantısına ihtiyacı olmayan **Bluetooth'un diğer özellikleri** aşağıdaki gibidir:

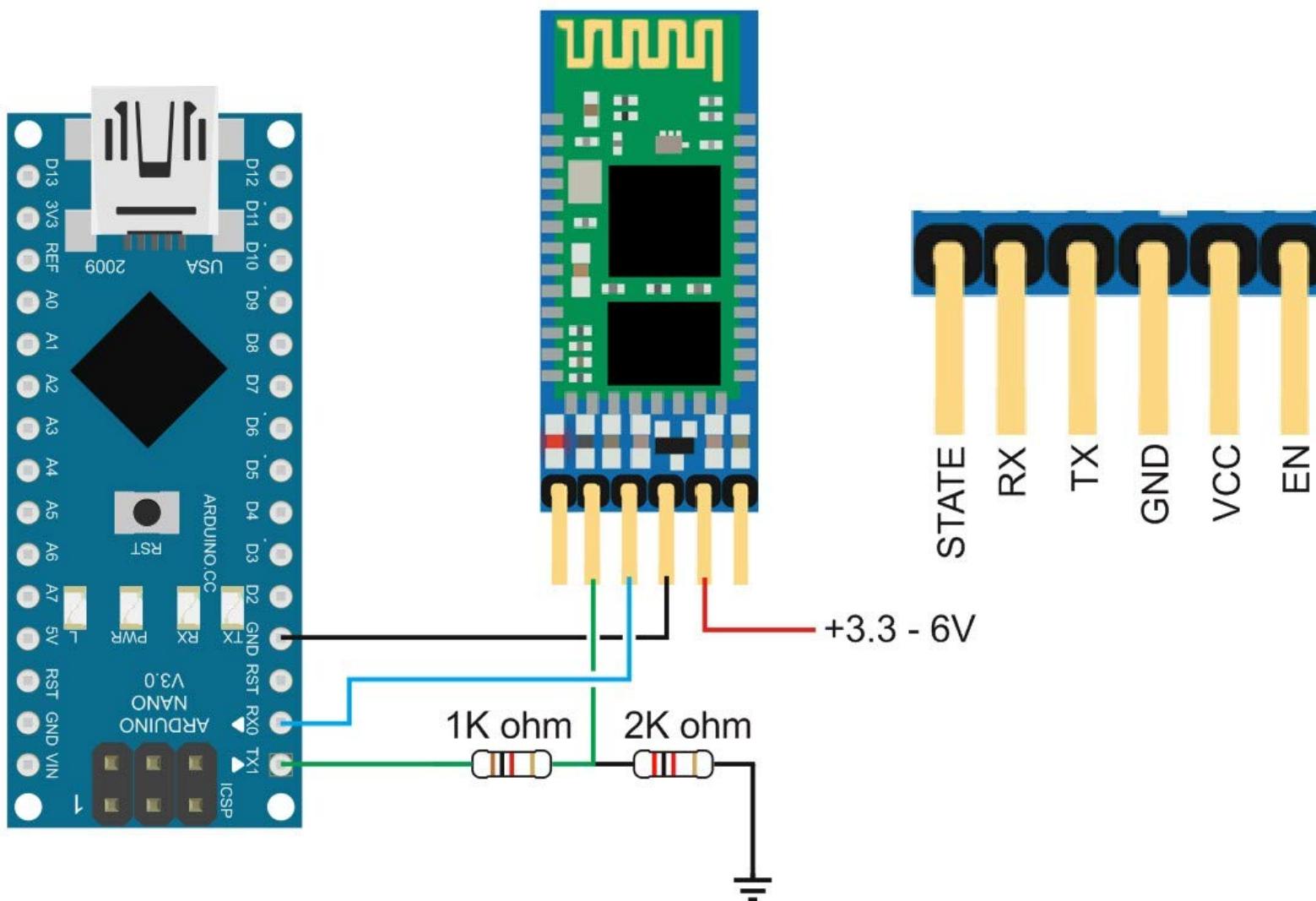
- Kısa mesafe radyo frekansları ile iletişim kurar.
- 2.4 GHZ hızıyla iletişimini sağlar.
- Veri aktarım hızı 1 mbps ve 721 mbps aralığındadır.
- Ses ve görüntü gibi farklı dosyaları aktarır.
- Güçlü bir iletişim kaynağıdır ve enerji tüketimi düşüktür.

# Bluetooth Modülleri



# BlueTooth Bağlantı Şekli

HC-05



# BlueTooth Kodlamasi

```
char Incoming_value = 0; //Variable for storing Incoming value
void setup()
{
  Serial.begin(9600); //Sets the data rate in bits per second (BPS)
  pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
}
void loop()
{
  if(Serial.available() > 0)
  {
    Incoming_value = Serial.read(); //Read the incoming data and store it in Incoming_value
    Serial.print(Incoming_value); //Print value of Incoming_value
    Serial.print("\n"); //New line
    if(Incoming_value == '1') //Checks whether value of Incoming_value is 1
      digitalWrite(13, HIGH); //If value is 1 then LED turns ON
    else if(Incoming_value == '0') //Checks whether value of Incoming_value is 0
      digitalWrite(13, LOW); //If value is 0 then LED turns OFF
  }
}
```

```
#include <SoftwareSerial.h>

SoftwareSerial bt_iletisim(7, 6); // RX,TX

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

void loop()
{
  if (bt_iletisim.available())
  {
    char data = bt_iletisim.read();
    Serial.println(data);
  }
}
```

# Wi-Fi

Wi-Fi (İngilizce: Wireless Fidelity, Türkçe: Kablosuz Bağlantı Alanı) kişisel bilgisayar, video oyunu konsolları, dijital ses oynatıcıları ve akıllı telefonlar gibi cihazların kablosuz olarak birbirlerine bağlanmasını sağlayan teknolojidir.

Wi-Fi ürünlerin kablosuz bağlantı sağlayabildiğini gösteren bir uyumluluk göstergesidir ve IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n ve IEEE 802.11ac standartlarına göre belirlenir.

Wi-Fi dizüstü bilgisayarlar, PDAlar ve diğer taşınabilir cihazların yakınlarındaki kablosuz erişim noktaları aracılığıyla yerel alan ağına bağlanabilmesini sağlar. Bağlantı, kablosuz erişim noktaları ve cihazın ortak desteklediği, IEEE 802.11 protokolüne bağlı olarak 2.4 GHz veya 5 GHz radyo frekansında gerçekleştirilir. Veri, CSMA/CA (Carrier sense multiple access with collision avoidance) protokolüne uygun gönderilip alınır ve böylece paketlerin iletimi sırasında hata oluşması sorunu çözülür.

Wi-Fi; kişisel bilgisayar, akıllı telefon, oyun konsolu ve dijital işitsel cihazların kablosuz ağ sahası içerisinde internete bağlanmasını sağlayan bir teknolojidir. Wi-Fi'ın erişim noktaları, bir oda gibi küçük alanlardan, birkaç milkarelik geniş alanlara kadar etki edebilir, buralarda internet erişimini sağlayabilirler. Ev ve ofislerdeki özel kullanımının yanında kamuya ait alanlarda da ücretsiz olarak veya diğer ticari şekillerde kullanılmaktadır. Havaalanları, otel ve restoran gibi yerlerde genel olarak bu hizmet ücretsiz olarak sağlanmaktadır. 2008'den bu yana 300 metropolite Wi-Fi (Muni-Fi) projesi başlamıştır.

# Wi – Fi Özellikleri

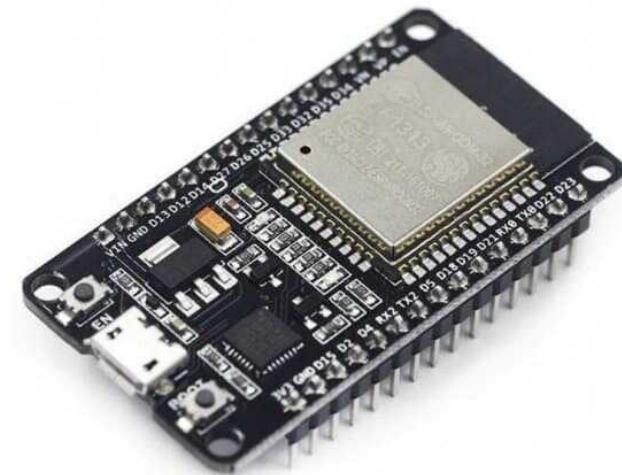
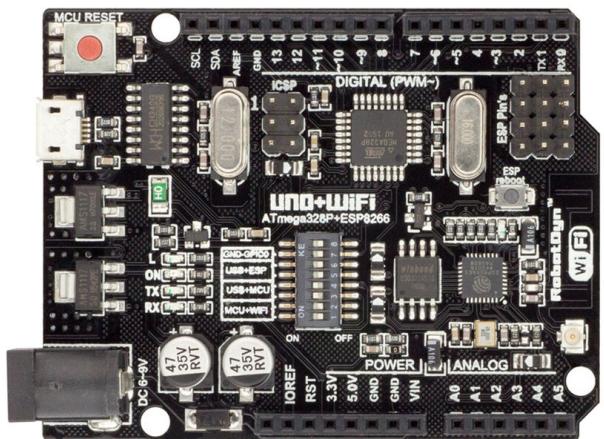
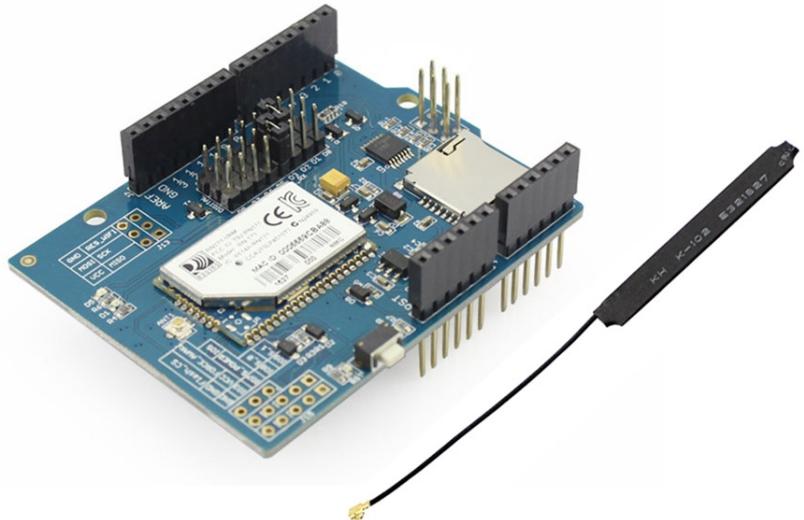
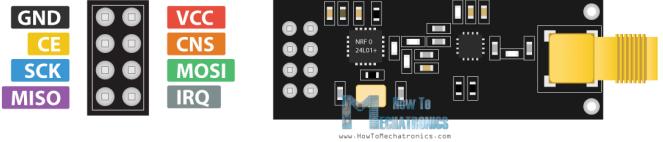
- Lisans gerektirmeyen frekanslarda çalışır
- Ağ için kablolama gereksinimi yoktur, böylece kablo çekilemeyecek binalarda veya binalararası bağlantınlarda kolaylık sağlar.
- Diğer kablosuz çözümlere göre çok daha ucuz ve kolay alınıp kurulabilir.
- Birden çok kablosuz erişim noktası kullanılan ağlarda kablosuz dolaşım ile kablosuz iletişim kesilmeden bir erişim noktasından diğerine geçiş yapılabilir.
- WEP, WPA ve benzeri kablosuz şifreleme yöntemleri veya IEEE 802.1x gibi yetkilendirme yöntemleriyle çeşitli güvenlik seçenekleri sunar.
- Wi-Fi Global bir standart kümesidir, dünyanın her yerinde aynı şekilde çalışır.

# Wi – Fi Modülleri

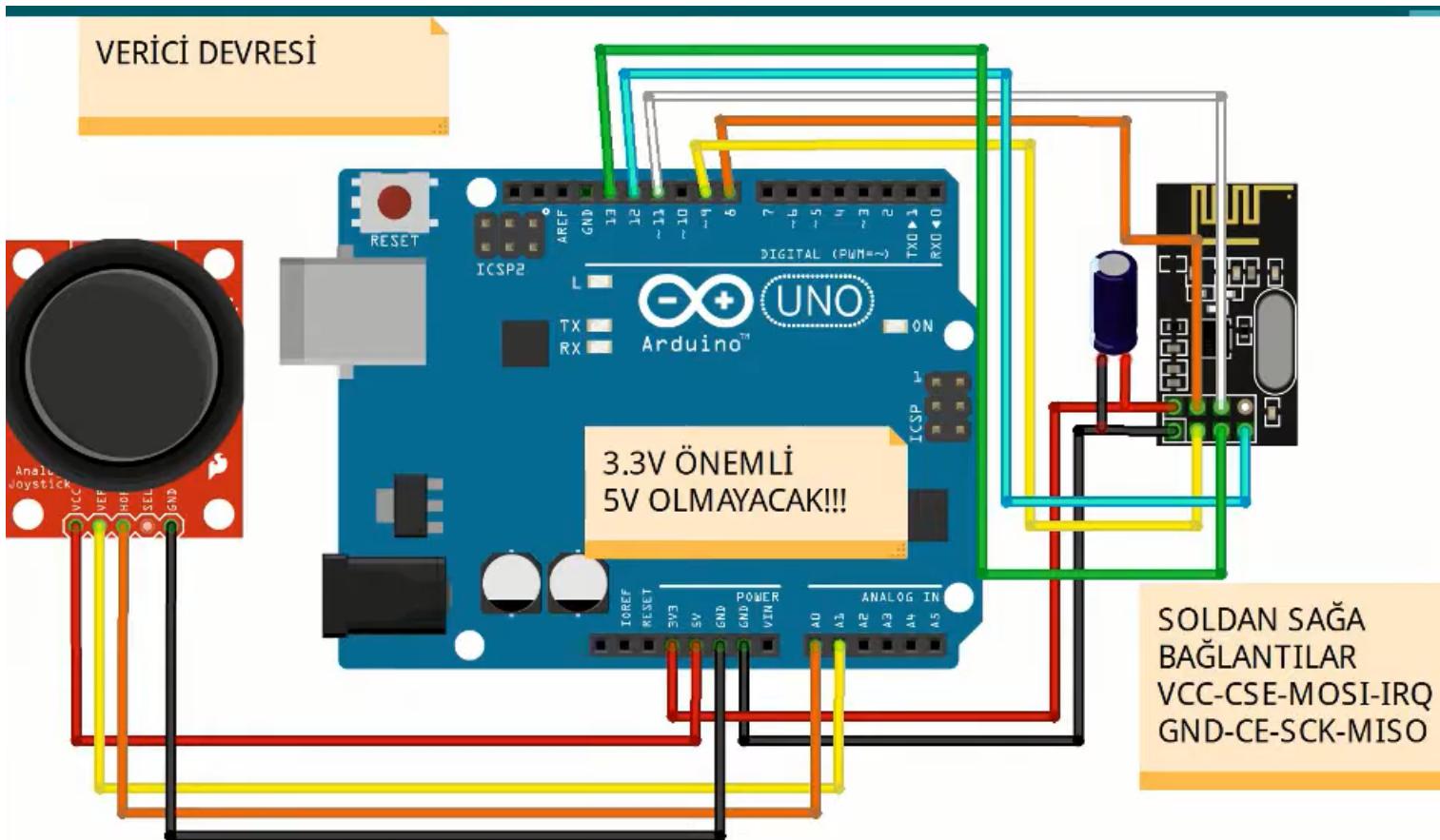
NRF24L01 Pinout



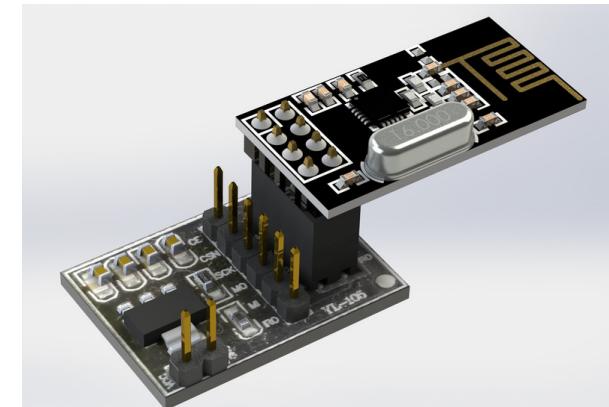
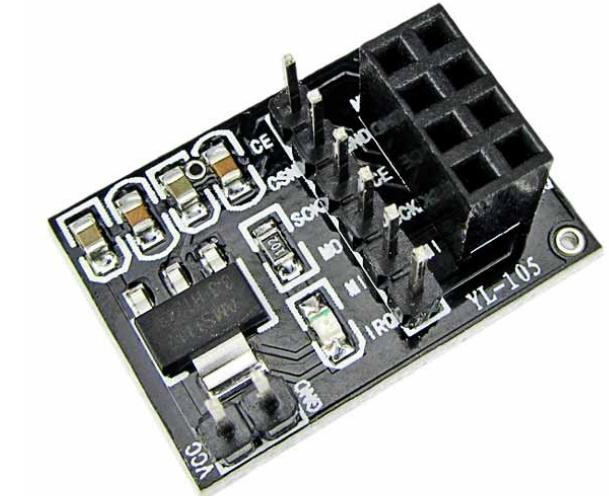
NRF24L01+ PA/LNA Pinout



# Nrf24 Örneği



fritzina



# Nrf24 Transmitter Kodu

```
/*-----( Gerekli Kütüphaneleri Yükliyoruz )-----*/
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
/*-----( Değişkenlerimizi tanımlıyoruz )-----*/
#define CE_PIN 9
#define CSN_PIN 8
#define YEKSEN A1
#define XEKSEN A0

// NOT: Sondaki "LL" "LongLong" tipi demek
const uint64_t pipe = 0xE8E8F0F0E1LL; // yayın adresi diyebiliriz iki tarafta da aynı olmalı ki haberleşecek

/*-----( RF24 kütüphanesi ile radio adında bir haberleşme nesnesi yaratıyoruz )-----*/
RF24 radio(CE_PIN, CSN_PIN); // radio yaratıldı
/*-----( Girdileri tutacağımız diziyi tanımlayalım )-----*/
int verilerim[2]; //

void setup() //***** SETUP: 1 defa çalışır *****/
{
    Serial.begin(57600);
    radio.begin(); //İletişimi başlattık
    //kumandaladaki pinleri tanımlıyoruz
    pinMode(YEKSEN, INPUT);
    pinMode(XEKSEN, INPUT);

    radio.openWritingPipe(pipe); //adrese bağlanıyoruz
}//--(setup kısmı bitimi )---

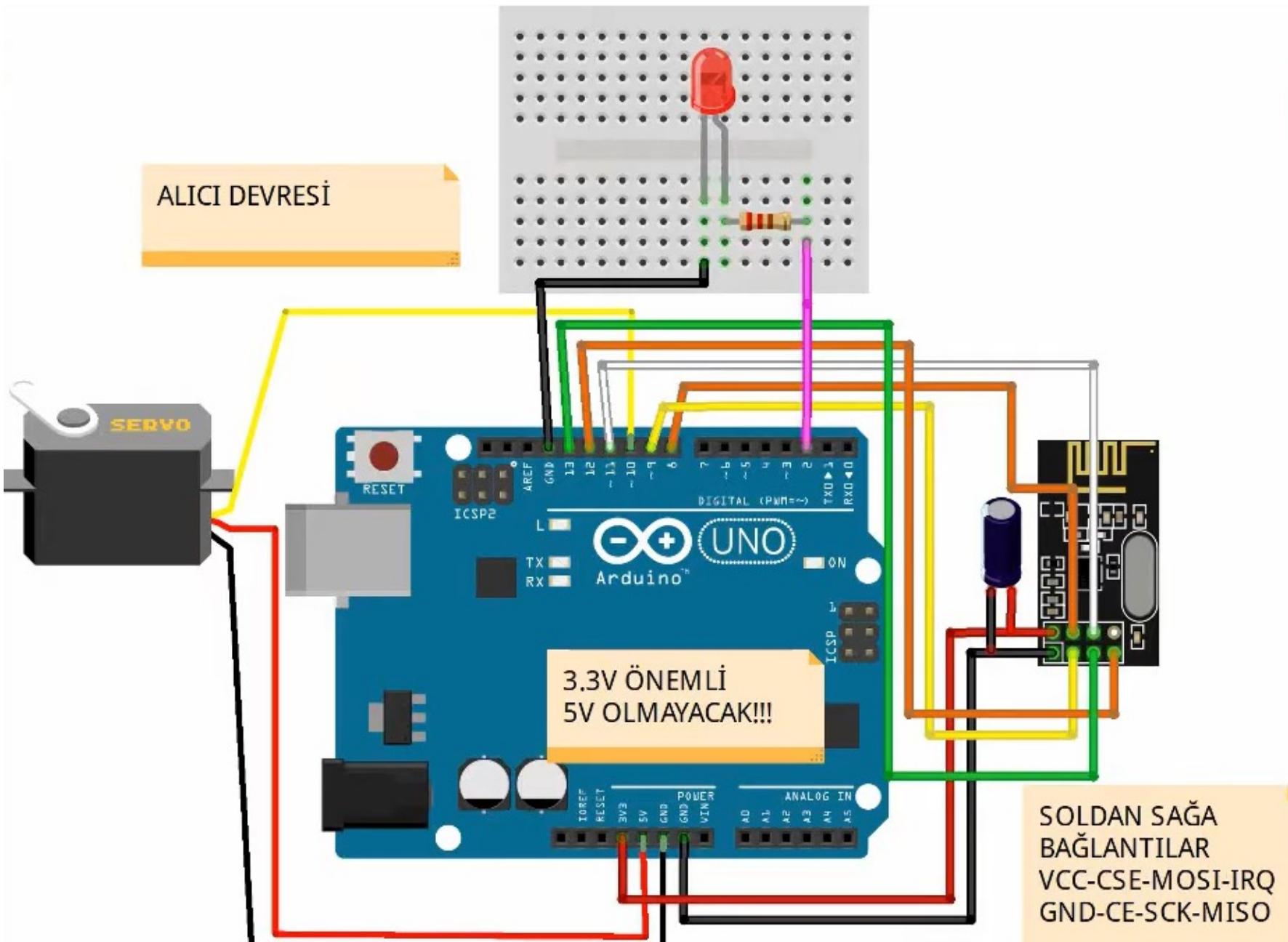


void loop() //***** LOOP: sürekli çalışıyor *****/
{
    verilerim[0]=analogRead(A0); //kumandanın verileri okuyup dizimize yazdırıyoruz
    verilerim[1]=analogRead(A1);

    radio.write( verilerim, sizeof(verilerim) ); //verileri gönderilmek üzere çıkışa yazdırıyoruz

    // loop sonu

    //
    //BİTTİ
```



# Nrf24 Recevier Kodu

```
#define KEY1 0 //diziden gelecek değişkenler için tanımladık (bu örnekte 2 veri alacağız)
#define KEY2 0

/*
-----( kütüphaneleri yükleyelim )-----
#include <Servo.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

//RF modülün pinleri
#define CE_PIN 9
#define CSN_PIN 8

const uint64_t pipe = 0xE8E8F0F0E1LL; // adres

/*
-----( radio nesnesi yaratılıyor)-----
RF24 radio(CE_PIN, CSN_PIN); //
/*
-----( değişkenler tanımlanıyor )-----

// radiodan gelen 5 adet veriyi almak için bir dizi oluşturuyoruz
int verilerim[] = {KEY1, KEY2}; // keylerimiz
int servodegeri, leddegeri;
Servo servoNesnesi;//servomuz için bir nesne tanımladık.

int led=2; //ledi Arduino'nun 2.pinine
//Bu ledi kumanda ile iletişim kurma başarılı ise yakıp kumandayı söndürüyoruz

void setup()
{
    Serial.begin(57600);
    delay(1000);

    pinMode(led, OUTPUT);
    servoNesnesi.attach(10);//servomuzun sinyal kablosu 10. pwm kanala bağlı
    /////////////////////////////////
    //radio ile iletişim kuruyoruz
    Serial.println("Nrf24L01 Alıcı Başlatılıyor");
    radio.begin();
    radio.openReadingPipe(1, pipe);
    radio.startListening();
}

void loop() //***** LOOP: sürekli çalışır *****/
{
    if ( radio.available() ) //eğer radyo ile temas kurabilmisse
    {
        // done değişkeni false yapıyoruz ve while içinde sonsuz döngü oluşturarak verileri toplayacağız
        bool done = false;
        while (!done)
        {
            // veriyi alıyoruz.
            done = radio.read( verilerim, sizeof(verilerim) ); //ne kadar veri varsa al demek
            digitalWrite(led, LOW); //ledi söndürdük
            servodegeri=map[verilerim[0],0,1023,0,179];//stickten gelen bilgiyi servo için 0-179 aralığına uyarladık
            leddegeri=map[verilerim[1],0,1023,0,255];//stickten gelen bilgiyi 0-255 aralığına uyarladık
            //serial monitörden verileri görmek için;
            Serial.print("KEY1 = ");
            Serial.print(servodegeri);
            Serial.print(" KEY2 = ");
            Serial.print(leddegeri);
            Serial.println();

            servoNesnesi.write(servodegeri);

            if ((leddegeri > 130) || (leddegeri < 120)) // sadece ileri komutu veriliyor ileri gidecek
            {
                digitalWrite(led, HIGH);
            }
            else
            {
                digitalWrite(led, LOW);
            }

            } //while
        } //ilk if

        //eğer radyo bulunamazsa
        else
        {
            Serial.println("Verici Bulunamadı");
            verilerim[0] = 0;
            verilerim[1] = 0;

            servoNesnesi.write(90);//servoyu ortalıyor
            digitalWrite(led, LOW); //ledimizi de söndürüyoruz
        }
    } //---(main loop sonu )---
}
```

# ESP8266

## ESP8266 WiFi Serial Modül

Nesnelerin interneti alanında geliştirdiğiniz projeler için oldukça popüler bir ürün olan ESP8266 WiFi Serial Modülü. Bu ürün sayesinde kablosuz ağlara kolayca bağlanabilir, projelerinizde istediğiniz uygulamaları rahatlıkla gerçekleştirebilirsiniz. Ekonomik fiyatı ile dikkat çeken ürün, Arduino ile de uyumlu bir kullanım sunuyor. Modül üzerinde yer alan dahili anten wi-fi ağlara kolayca bağlanabiliyor. WiFi ESP8266 Serial Modülü uygun fiyatlarla satın almak için hemen siparişinizi oluşturun!

Oldukça entegre bir çip olan WiFi Serial Modülü nesnelerin interneti projelerinde sıkılıkla kullanılmaktadır. TTL (seri haberleşme) ile kablosuz internet ağına bağlanabilen modül, kendi internet ağını yayarak diğer cihazların da bu ağa bağlanabilmesini sağlamaktadır. Tüm Wi-Fi ağ işlevlerini başka bir uygulama işlemcisinden yönlendirebilmesine olanak tanır.

### ESP8266 WiFi Serial Modül Özellikleri

Gelişen teknoloji ile birlikte artık birçok nesnenin çevre ile etkileşeme gereken daha akıllı sistemler oluşturduğunu görüyoruz. Siz de projelerinizde akıllı sistemler tasarlıyor ve wi-fi bağlantınızı ekonomik ve kullanışlı bir ürün ile çözmek istiyorsanız Wi-fi Serial Modülü projelerinize dahil edebilirsiniz.

11 b/g/n

Wi-Fi Direct (P2P), soft-AP

Entegre TCP/IP protocol stack

Entegre TR switch, balun, LNA, power amplifier ve network eşleyici

Entegre PLLs, regülatörler DCXO and güç dağıtım uniteleri

11b modunda +19.5dBm çıkış gücü verir

Power down leakage akım <10uA

1MB Flash Bellek

Düşük güç tüketimli 32-bit CPU işlemci uygulamalarda da kullanılabilir

SDIO 1.1 / 2.0, SPI, UART

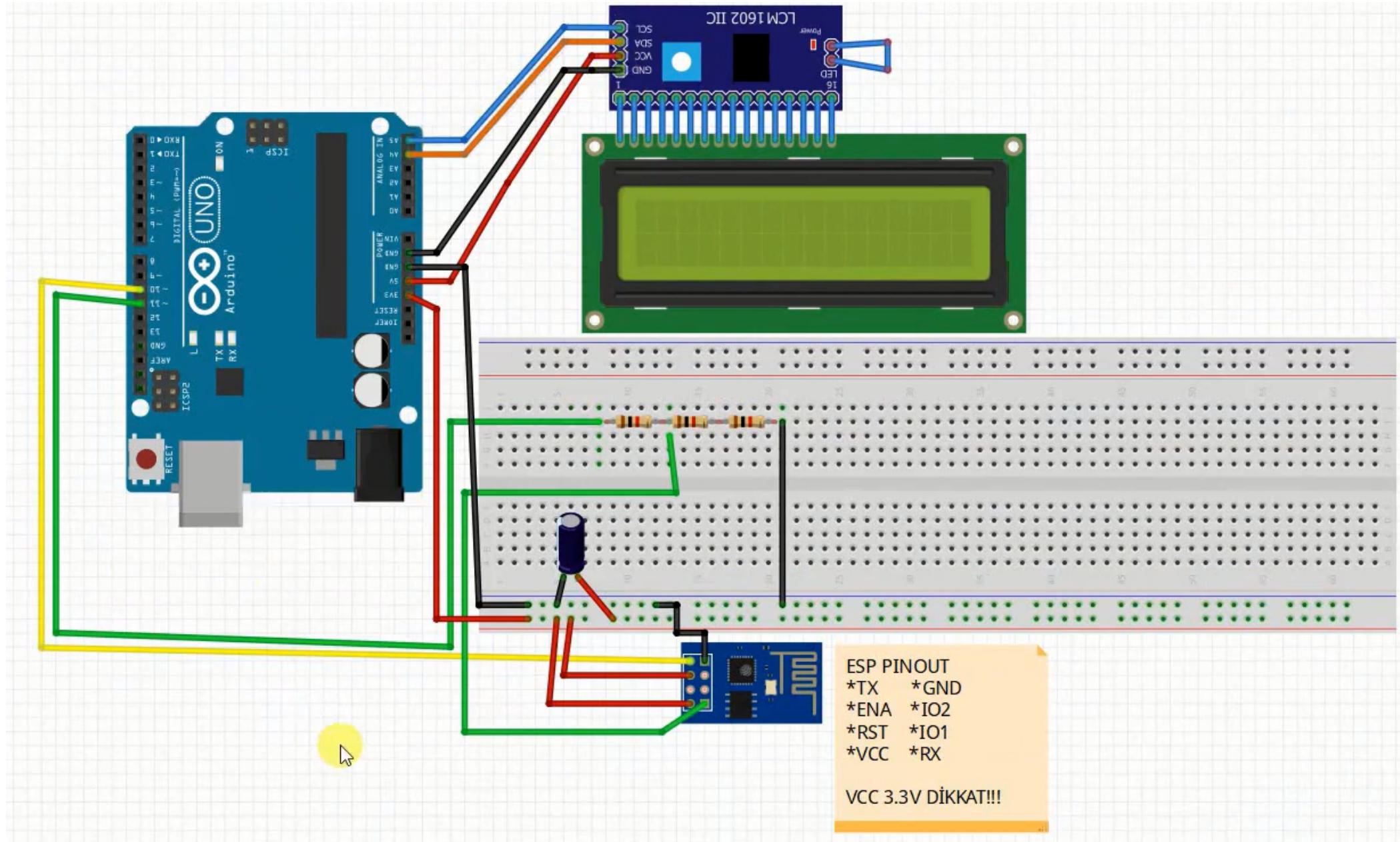
STBC, 1x1 MIMO, 2x1 MIMO

A-MPDU & A-MSDU aggregation & 0.4ms guard interval

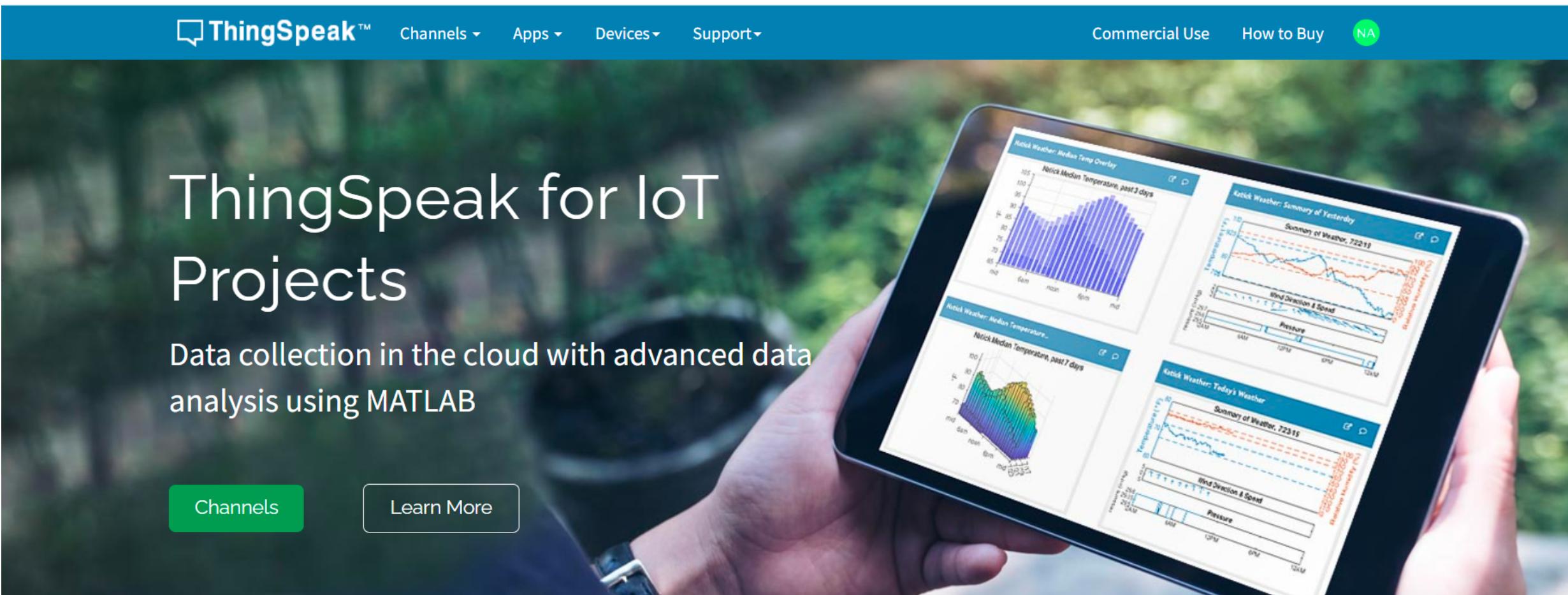
Açılış ve paket transfer süresi < 2ms

Standby güç tüketimi < 1.0mW (DTIM3)

# ESP8266 Uygulaması



# ESP8266 Uygulamasi



The image shows the homepage of ThingSpeak, a cloud service for IoT projects. At the top, there's a navigation bar with links for "ThingSpeak™", "Channels", "Apps", "Devices", "Support", "Commercial Use", "How to Buy", and a "NA" button. The main title "ThingSpeak for IoT Projects" is displayed prominently on the left, with a subtitle "Data collection in the cloud with advanced data analysis using MATLAB". Below the title are two buttons: "Channels" (green) and "Learn More" (white). On the right, a person is holding a tablet displaying a dashboard for "Katik Weather". The dashboard includes several charts and graphs showing temperature, wind direction, and pressure data over different time periods (3 days, 7 days, yesterday, today). The background of the page features a blurred outdoor scene with greenery.



# ESP8266 Uygulaması

```
#include <SoftwareSerial.h>           //Gerekli kütüphaneleri ekliyoruz.  
  
#include <Wire.h>  
  
#include <LiquidCrystal_I2C.h>  
  
String agAdi = "*****";             //Kablosuz Ağ adını buraya yazıyoruz.  
String agSifresi = "*****";         //Ağ şifresini buraya yazıyoruz.  
  
int rxPin = 10;                    //ESP8266 TX pinı  
int txPin = 11;                    //ESP8266 RX pinı  
  
String ip = "184.106.153.149";      //Thingspeak sitesinin ip adresi  
  
#define DEBUG true                 //çalışmanın düzgün olduğunu takip etmek için kullandım  
  
SoftwareSerial esp(rxPin, txPin);   //Seri haberleşme için pin ayarlarını yapıyoruz.  
LiquidCrystal_I2C lcd(0x27, 16, 2); //ekranın adresi, sütun sayısı, satır sayısı  
  
void setup() {  
    lcd.begin();                  //LCD ekranı başlatıyoruz  
    Serial.begin(9600);            //Seri port haberleşmesini başlattık  
    Serial.println("Başlatılıyor");  
    //esp.begin(115200);           //ESP8266 ile seri haberleşmeyi başlatıyoruz. İlk çalışmada 115200 sonra 9600 yapıcıaz  
    esp.begin(9600);              //ESP8266 ile seri haberleşmeyi başlatıyoruz.  
    esp.println("AT");            //AT komutu ile modül kontrolünü yapıyoruz.  
    Serial.println("AT Gonderildi");  
  
    Serial.println("AT Gonderildi");  
    while(!esp.find("OK")){        //Modül hazır olana kadar bekliyoruz.  
        esp.println("AT");  
        Serial.println("ESP8266 ile iletişim kurulamadı.");  
    }  
    //esp.println("AT+UART_DEF=9600,8,1,0,0");      //ilk seferde sadece 1 kereliğine çalıştırıp baud rate değişt  
    Serial.println("OK Komutu Alındı");  
    esp.println("AT+CWMODE=1");       //ESP8266 modülünü client olarak ayarlıyoruz.  
    while(!esp.find("OK")){          //Ayar yapılanca kadar bekliyoruz.  
        esp.println("AT+CWMODE=1");  
        Serial.println("Modül Ayarı Yapılıyor....");  
    }  
    Serial.println("Modül Client olarak ayarlandı");  
    Serial.println("Ağa Baglanıyor...");  
    esp.println("AT+CWJAP=\""+agAdi+"\",\""+agSifresi+"\""); //Kablosuz Ağa bağlanıyoruz.  
    while(!esp.find("OK"));          //Ağa bağlanana kadar bekliyoruz.  
    Serial.println("Ağa Bağlandı.");  
    delay(1000);  
  
    lcd.clear();                  //LCD ekranı temizledik  
    lcd.home();                   //Kursörü başa aldık  
}  
  
void loop() {  
    esp.println("AT+CIPSTART=\"TCP\",\""+ip+"\",80"); //Thingspeak sitesine TCP ile bağlanıyoruz.  
    if(esp.find("Error")){           //Bağlantı hatası kontrolü yapıyoruz.  
        Serial.println("AT+CIPSTART Error");  
    }  
}
```



# ESP8266 Uygulaması

```
void loop() {  
  
esp.println("AT+CIPSTART=\"TCP\",\"+ip+,80"); //Thingspeak sitesine TCP ile bağlanıyoruz.  
  
if(esp.find("Error")){ //Bağlantı hatası kontrolü yapıyoruz.  
  
Serial.println("AT+CIPSTART Error");  
  
}  
  
lcd.home();  
  
lcd.print("ALTIN CANLI GR");  
  
lcd.setCursor(4, 1);  
  
lcd.print(altin());  
  
Serial.println("Baglanti Kapatildi.");  
  
esp.println("AT+CIPCLOSE"); //Bağlantıyı kapatıyoruz  
  
delay(1000);  
  
}  
  
}
```



```
String Time()  
{  
  
String rest = "AT+CIPSEND=90";  
rest += "\r\n";  
  
sendData(rest, 2000, 0); //Gönderilecek Karakter Sayısı. ("AT+CIPSEND=90")  
  
}
```

```
String hostt = "GET /apps/thinghttp/send_request?api_key=DOS4J9JD9IR1CADD";  
hostt += "\r\n";  
hostt += "Host:api.thingspeak.com";  
  
String hostt = "GET /apps/thinghttp/send_request?api_key=RHLEEPPLGG17UK8TJ";  
hostt += "\r\n";  
hostt += "Host:api.thingspeak.com";  
hostt += "\r\n\r\n\r\n\r\n\r\n\r\n\r\n";  
  
String Altin = sendData(hostt, 2000, 1); // GET request ( GET /apps/thinghttp/send_request?api_key=XXXXXXXXXXXXXX  
// Host: Host_server_name )  
  
***** gelen verinin içinden sadece saatle ilgili bölümü alıyoruz.*****  
int baslangic=Time.indexOf(':');  
Time=Time.substring(baslangic+1,baslangic+9);  
  
return (Time);  
  
}  
  
String altın()  
{  
  
String rest = "AT+CIPSEND=90";  
rest += "\r\n";  
  
sendData(rest, 2000, 0); //Gönderilecek Karakter Sayısı. ("AT+CIPSEND=90")  
  
}
```

# ESP8266 Uygulaması

```
***** gelen verinin içinden sadece ilgili bölümü alıyoruz. *****  
int baslangic=Altin.indexOf(':');  
Altin=Altin.substring(baslangic+1,baslangic+9);
```

```
return (Altin);
```

```
}
```

```
String covid()
```

```
{
```

```
String rest = "AT+CIPSEND=90";
```

```
rest += "\r\n";
```

```
sendData(rest, 3000, 0); //Gönderilecek Karakter Sayısı. ("AT+CIPSEND=90")
```

```
String hostt = "GET /apps/thinghttp/send_request?api_key=GF15NKHSJR0PZOPJ";
```

```
hostt += "\r\n";
```

```
hostt += "Host:api.thingspeak.com";
```

```
hostt += "\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n";
```

```
String Nufus = sendData(hostt, 3000, 1); // GET request ( GET /apps/thinghttp/send_request?api_key=XXXXXXXXXXXXXX
```

```
// Host: Host_server_name )
```

```
int baslangic=Nufus.indexOf(':');
```

```
Nufus=Nufus.substring(baslangic+1,baslangic+11);
```

```
return (Nufus);
```

```
}
```

```
String sendData(String komut, const int zamangecen, boolean debug)
```

```
{
```

```
String response = "";
```



```
esp.print(komut); // komut satırını esprin çıkışına yazdırıyoruz
```

```
long int Zaman = millis();
```

```
while ( (Zaman + zamangecen) > millis() )
```

```
{
```

```
while (esp.available())
```

```
{
```

```
if (debug)
```

```
{
```

```
Serial.print(response);
```

```
}
```

```
while (esp.available())
```

```
{
```

```
// esp'nin bağlantısından gelen veri var ise
```

```
char c = esp.read(); //her bir karakteri okuyup c ye yazdırıyoruz.
```

```
response += c; // her karakteri response değişkeninin sonuna ekliyor ve alacağımız tekst oluyoruz
```

```
}
```

```
}
```

```
return response;
```

```
}
```



# GPS Nedir?

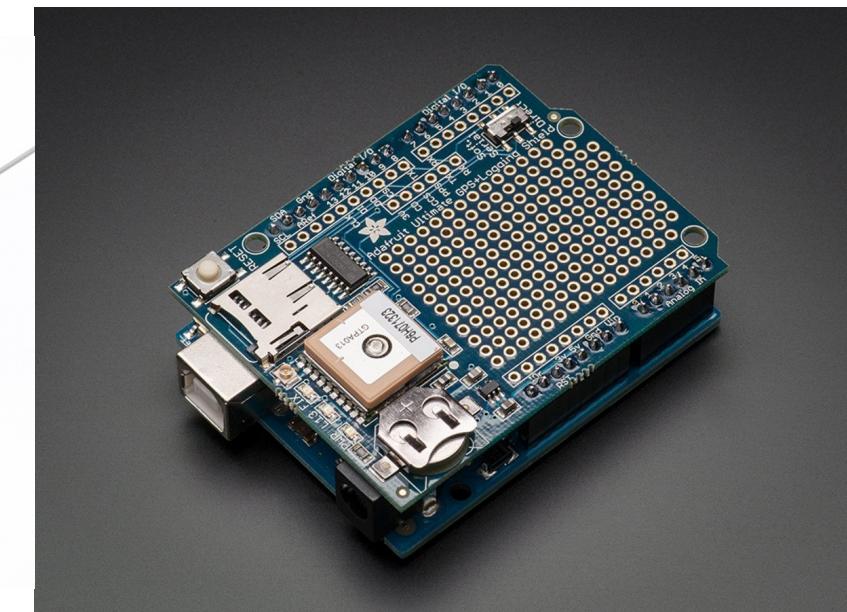
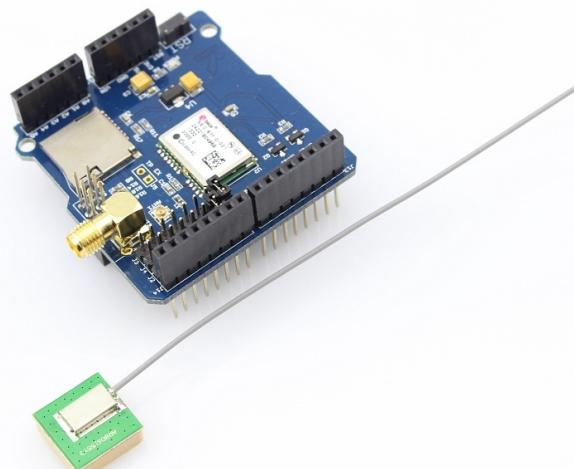
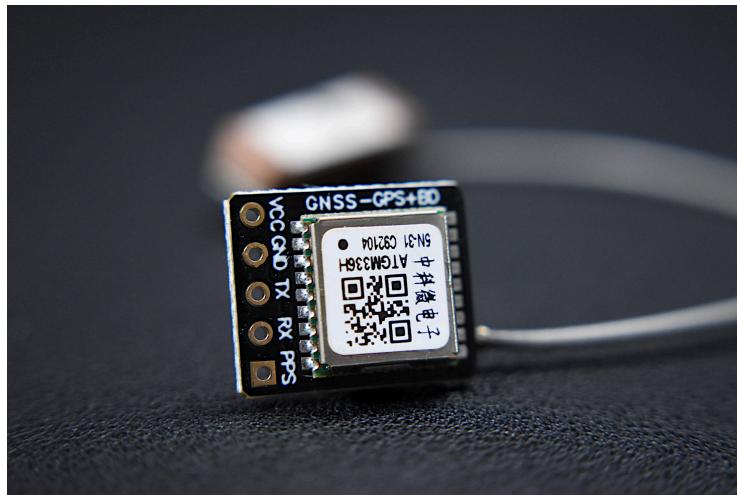
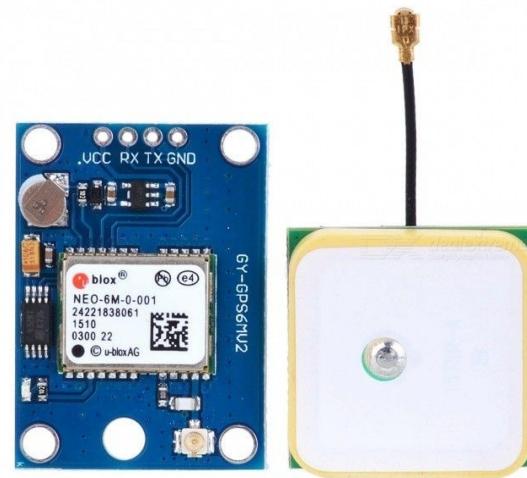
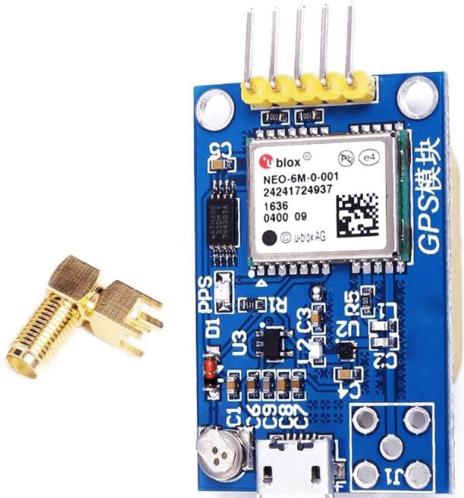
Global Positioning System (kısaca GPS veya Türkçe karşılığıyla Küresel Konumlama Sistemi), ABD hükümetine ait ve ABD Uzay Kuvvetleri tarafından yönetilen uydu tabanlı radyonavigasyon sistemidir.[1] Dünya'daki ve Dünya yakınındaki GPS alıcılarına, en az dört GPS uydusunu görebilmeleri şartıyla coğrafi konum ve saat bilgisi sağlayan küresel uydu navigasyon sistemlerinden biridir. Uydular bir tür radyo sinyali yayarlar ve yeryüzündeki GPS alıcıları bu sinyalleri alıp yorumlayarak konum belirlenmesini gerçekleştirir.

GPS projesi, öncelindeki navigasyon sistemlerinin kısıtlı işlevselliklerini aşabilmek amacıyla[2] 1960'lardan gelen bir dizi gizli mühendislik çalışması da dahil olmak üzere ilk denemelerde ortaya çıkan birkaç görüşün de bütünlendirilmesi ile, 1973 yılında geliştirilmiştir. GPS, ABD Savunma Bakanlığı (DoD) tarafından esas olarak 24 uydı ile çalışacak şekilde tasarlanıp yapılmış ve devreye alınmıştır. 1994 yılında tam olarak işler hale gelmiştir. Sistem, Bradford Parkinson, Roger L. Easton ve Ivan A. Getting'in icatları ile güçlendirildi.

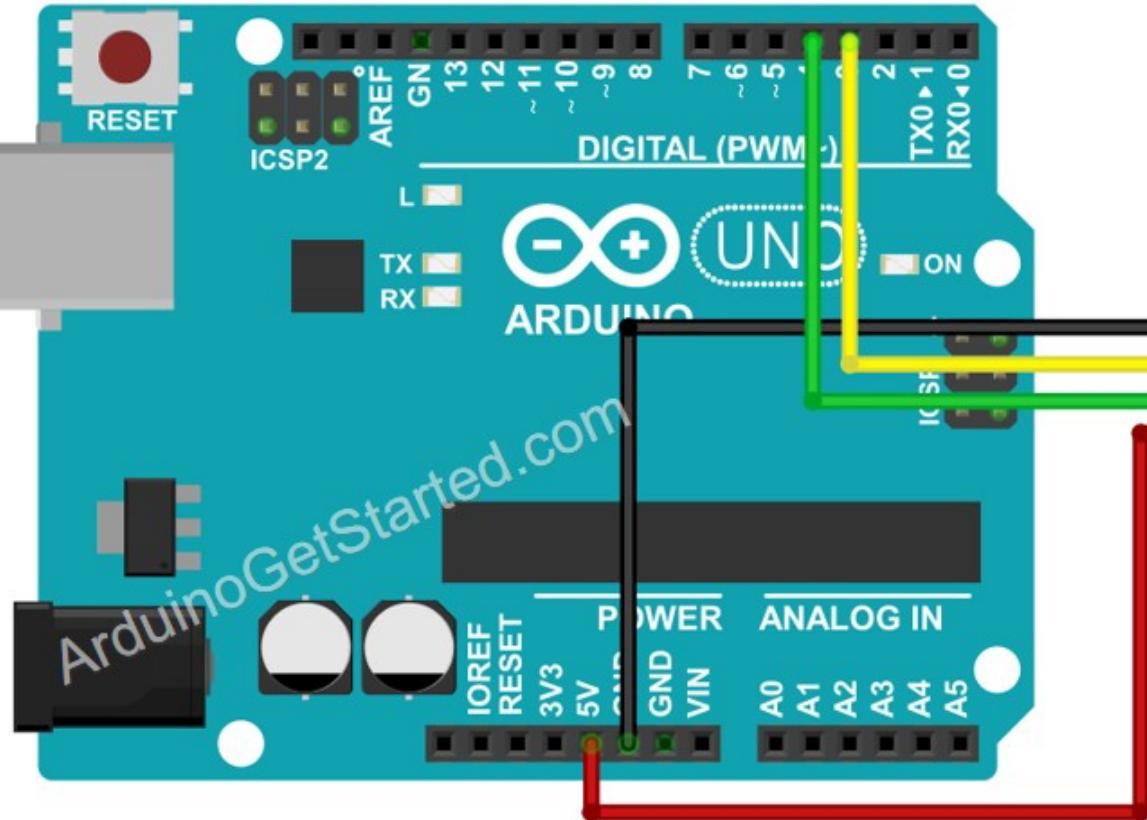
GPS sistemi, var olan sistem üzerinde teknoloji ilerlemeleriyle ve yeni taleplerle artık yenileştirilme ve Yeni Nesil Operasyonel Kontrol Sistemi (OCX) destekli, gelişmiş GPS III uydularının hayatı geçirilmesi çabalarına yol açmıştır.[3] Beyaz Saray ve Başkan Yardımcısı Al Gore 1998 yılında duyurular ile bu değişimi başlattı. 2000 yılından beri, GPS III yenileştirmeleriyle ilgili kararlarda ABD Kongresi yetkilidir.

Ek olarak diğer sistemlerin kullanımında GPS geliştirilme aşamasındadır. Rus navigasyon sistemi GLONASS, GPS ile birlikte çağrıdaşı olarak geliştirilmektedir; ama O, 2000'li yılların ortalarına kadar dünyayı tam olarak kapsamadan çalışmıştır.[4] GPS'in yanı sıra AB tarafından geliştirilen Galileo, Çin tarafından geliştirilen Compass ve Hindistan tarafından geliştirilen IRNSS adlı konumlandırma sistemleri de vardır.

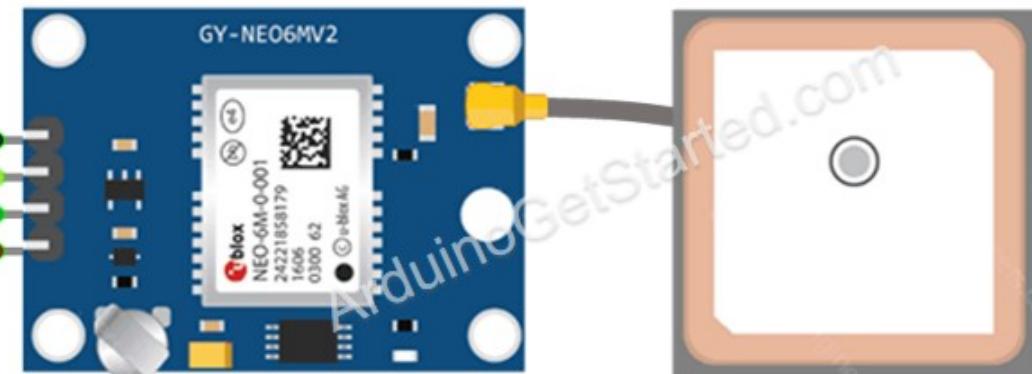
# GPS Modülleri



# GPS Uygulamasi



NEO-6M GPS Module



 **ArduinoGetStarted.com**

# GPS Uygulamasi

```
#include <SoftwareSerial.h>

SoftwareSerial GPS(10, 11); // Rx, Tx

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

void loop() {
  String veri = GPS.readStringUntil('\r');
  Serial.print(veri);
}
```

# GPS Uygulaması

# GPS Uygulamasi

```
#include <SoftwareSerial.h>

SoftwareSerial GPS(10, 11); // Rx, Tx

String saat = ""; // hhmmss.00
String enlem = ""; // DDMM.MMMM Kuzey/Güney N/S
String boylam = ""; // DDMM.MMMM Doğu/Batı E/W
String irtifa = ""; // metre

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

void loop() {
  GPSdinle();
  Serial.print("Saat: ");
  Serial.println(saat);
  Serial.print("Enlem: ");
  Serial.println(enlem);
  Serial.print("Boylam: ");
  Serial.println(boylam);
  Serial.print("irtifa: ");
  Serial.println(irtifa);
  Serial.println();
}

void GPSdinle() {
```



```
void GPSdinle() {

  // $GPGGA arıyoruz
  if ( GPS.find("$GPGGA,") ) {

    // Gelen veriyi parçalıyoruz
    saat = GPS.readStringUntil(',');
    enlem = GPS.readStringUntil(',');
    enlem.concat(GPS.readStringUntil(',');
    boylam = GPS.readStringUntil(',');
    boylam.concat(GPS.readStringUntil(','));

    // irtifaya kadar olan kısmını atlıyoruz
    for ( int i = 0; i < 3; i++ ) {
      GPS.readStringUntil(',');
    }

    // irtifa verisini parçala
    irtifa = GPS.readStringUntil(',');
    irtifa.concat(GPS.readStringUntil(','));

    // Verinin geri kalanını atlıyoruz
    GPS.readStringUntil('\r');
  }
}
```