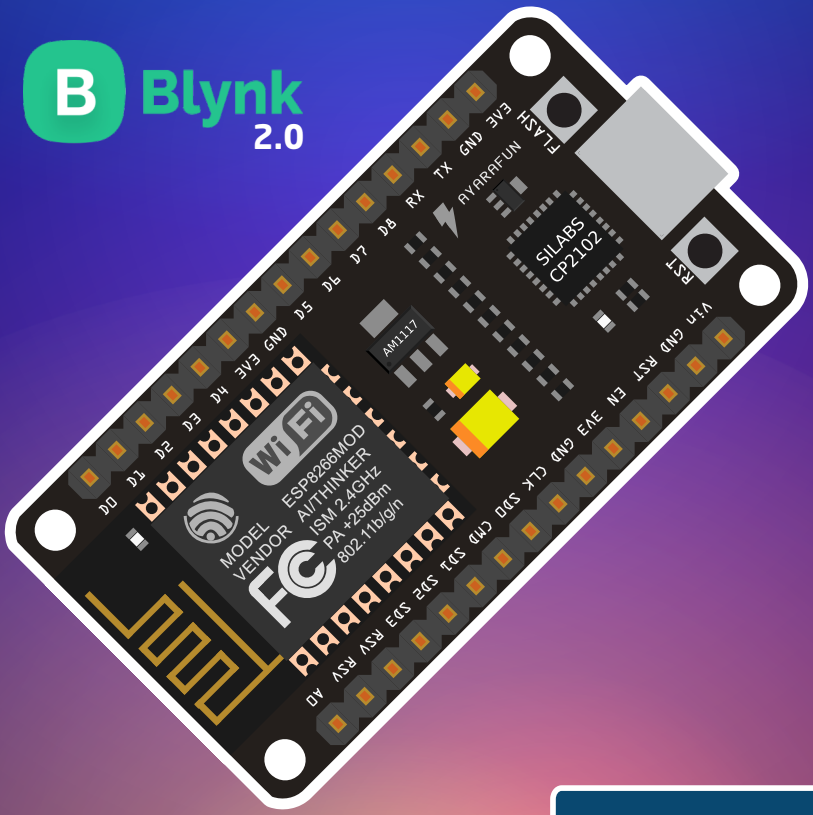
NodeMCU IoT Proje Geliştirme Kitabı

Blynk 2.0



robotistan

Elektronik ve Kodlama dünyasına hoşgeldiniz. Bu kitabı açtığınıza göre siz de merak denizinde yüzüp, yeni şeyler öğrenmeye heveslisiniz demektir. Bu tür konularda yeni şeyler öğrenmek zor gibi düşünülse de adım adım ve doğru uygulamalar ile ilerlerseniz çok basit olduğunu fark edeceksiniz. İlk aşamalarda uygulamaları yaptıkça oturmayan anlamsız gelen yerler olacaktır. Bu sorunu uygulama yaptıkça aşacaksınız. Sadece biraz sabır gerekli...Kolay ve doğru yol haritası ile Arduino programlamayı öğrenebilmeniz için uygulamalar kolaydan başlayarak, daha komplekse doğru ilerlemektedir.

Uygulamaların daha detaylı videolu anlatımlarını izlemek isterseniz kitabın arka kısmındaki QR kodu taratarak YouTube kanalımıza gidebilirsiniz. Uygulamalara dijital ortamda erişmek isterseniz http://maker.robotistan.com blog sayfamızda da bulunmaktadır. Kitapçık içerisinde yazılan kodlara hem ilgili videoların açıklama kısmından hem de blog sayfamızdan ulaşabilirsiniz.

Bu kitap Robotistan Elektronik A.Ş bünyesinde yazılmıştır. Yazılış amacı ise Arduino'ya kolay ve doğru yoldan başlamak isteyenlere rehber olmasıdır. Umudumuz bu içeriklerin herkese faydalı olması ve sizlerin öğrenme sürecini kolaylaştırıp hızlı şekilde proje yapmanızı sağlamaktır.

Set içerikleri, uygulamalar, videolarımız ve aklınıza takılan ter türlü öneri ve sorularınız için info@robotistan.com e-mail adresinden bize iletebilirsiniz.

Robotistan Ekibi

İçindekiler

Set İçeriğini Tanıyalım.....4

NodeMCU nedir?.....7

Tarayıcı Üzerinden Led Kontrolü.....9

Rgb Led Kontrolü.....17

Dht11 İle Sıcaklık Ve Nem Ölçümü.....24

Uzaktan Röle Kontrolü.....31

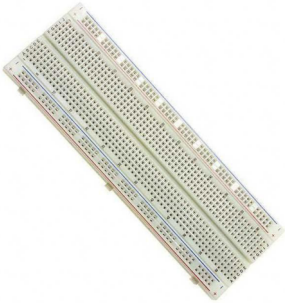
Toprak-Nem sensörü uygulaması.....36

Servo Motor Kontrolü.....41

Duman - Gaz Dedekötrü.....46

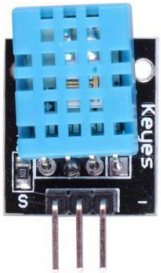
Set İçeriğini Tanıyalım

NodeMCU LoLin ESP8266 Geliştirme Kartı
Üzerinde NodeMCU firmware yüklü ESP8266 WiFi modül barındıran bir geliştirme kartıdır. ESP8266 SDK'sı kullanılarak geliştirildiğinden, ekstradan bir mikrokontrolcüye ihtiyaç duymadan GPIO, PWM, IIC, 1- Wire ve ADC bağlantılarını destekler.



Breadboard Nedir?
Breadboard üzerinde devrelerimizi test ettiğimiz araçtır. Kurduğumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sağlar. Tasarladığımız devreleri baskı devre veya delikli plaketler üzerine aktarmadan önce denememize olanak sağlar.

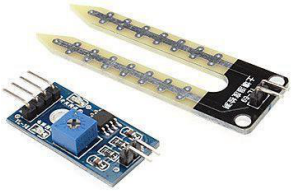
2 Kanal Role Kontrol Kartı
5V ile kontakların kontrol edilebildiği, Arduino veya diğer başka mikrodenetleyeciler ile kullanılabilen bir röle kartıdır. Mikrodenetleyeciden tetik sinyali sırasında 20mA'lik bir akım çekmektedir. Çeşitli hobi, endüstriyel ve robotik wprojelerde sıklıkla kullanılır. 30VDC veya 220VAC gerilimde 10A'e kadar akımı anahtarlayabilmektedir. Her bir role için kontrol ledleri bulunmaktadır.



DHT11 Sıcaklık ve Nem Sensörü
DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimidir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçer.

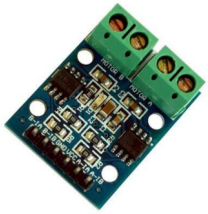
Set İçeriğini Tanıyalım

Mini Dalgıç Pompa
Çok sessiz çalışan düşük güç tüketimine sahip olan su geçirmez su motorumuz ile yağ e su gibi sıvıları saate 120 litreye kadar aktarabilirsiniz. IP68 standarta sahip bu ürün su ve toz geçirmemektedir.



Toprak Nem Sensörü
Toprağın içerisindeki nem miktarını veya ufak ölçekte bir sıvının seviyesini ölçmek için kullanabileceğiniz bir sensördür. Nem ölçer probları ölçüm yapılacak ortama batırılarak kullanılır. Toprağın veya içine batırılan sıvının meydana getirdiği dirençten dolayı, prob uçları arasında bir gerilim farkı oluşur.

Diyot LED
Işık yayan diyot (LED), adından da anlaşılacağı gibi enerji verildiği zaman görülebilir bir ışık yayan diyottur. Genel olarak kırmızı, sarı ve yeşil olmak üzere üç değişik renkte yapılırlar. Çalışma akımları 5 mA ile 50 mA arasındadır.



Motor Sürücü Kartı
Kart üzerinde L9110 motor sürücü entegresi bulunmaktadır. 2.5-12V arası giriş geriliminde çalışan sürücü kartı ile her iki yönde iki ayrı DC motor veya 4 kablolu 2 fazlı step motor kontrol edilebilir.
Sürücü çıkış gerilimi 7,6V'tur. Bu sayede 6V'luk motorlar ile kullanımı oldukça kolaydır. Kanal başına sürekli olarak 800mA ve anlık olarak ise 2A'e kadar akım verebilmektedir.

Set İçeriğini Tanıyalım

RGB LED Diyot
İçerisinde Kırmızı, Yeşil ve Mavi olmak üzere üç farklı renkte LED barındırmaktadır. Bu ledleri ayrı ayrı veya beraber yakarak gökkuşağının bütün renklerini elde edebilirsiniz.

Yanıcı Gaz ve Sigara Dumanı Sensör Kartı
Ortamda bulunan ve konsantrasyonu 300 ile 10,000 ppm arasında değişen yanıcı ve patlayıcı gaz ve/veya dumanı algılayan bu yarıiletken gaz sensörü, -20 ile 50°C arasında çalışabilir ve 5V'ta sadece 150mA akım çeker. Analog çıkışı sayesinde algılanan gaz konsantrasyonu kolayca okunabilir.

Sg90 Mini Servo Motor
Tower Pro SG90 küçük mekanizmalarınız için ideal bir servo motordur. Her marka uzaktan kumanda alıcılarına tam uyumlu olup RC araçlarınızda kullanabilirsiniz. Bunun yanı sıra birçok mikrodenetleyiciden alabileceğiniz PWM sinyali ile kendi yaptığınız robot projelerinizde de kolaylıkla kullanabilirsiniz.

Buzzer
Buzzer; mekanik, elektromekanik ya da piezoelektrik prensiplerine bağlı olarak çalışan işitsel ikaz cihazı çeşididir. Kullanım alanları oldukça fazla olan buzzerlar, genel itibarıyla piezoelektrik prensibiyle çalışmaktadırlar. Buzzerlar, kullanım alanlarına da bağlı olarak alarm, zamanlayıcı, onaylama cevap ikazı gibi işlevlerde kullanılabilmektedirler.

NodeMCU Nedir?

NodeMCU üzerinde ESP8266 modülü bulunduran açık kaynak kodlu, ufak boyutlu elektronik geliştirme kartıdır.

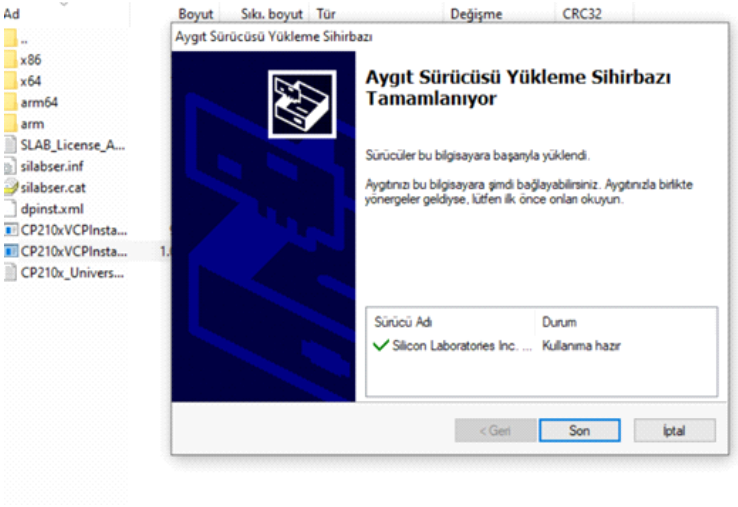
Ucuz olmasına rağmen çok stabil çalışan bir karttır. Kullanım alanı oldukça geniştir. Üzerinde bulunan ESP8266 Wifi modülü sayesinde internete kolay bir şekilde bağlanabiliyor, bu özelliği sayesinde uzaktan kontrol ve IOT projelerinde çok fazla kullanılır. Ayrıca düşük güç tükettiği için, güç tüketimi önemli olan projelerde de çok tercih edilir.

NodeMCU Driver Yüklemesi

NodeMCU V3 kartları genelde CP2102 versiyonu kullanır. Bu drive versiyonunu kurmak için zip dosyasınını indirelim.

https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip

Cp2102 driveri bilgisayarınıza yükledikten sonra, dosyanın içinde bulunan exe uzantılı dosyayı çalıştırıyoruz. Daha sonra açılan setup penceresinde ileri butonuna tıklıyoruz. Son butonuna tıkladıktan sonra setup penceresi otamatik kapanacaktır ve driverınız yüklenmiş olacaktır. Artık NodeMCUnuza Arduino IDE üzerinden sorunsuz bir şekilde kod yükleyebilirsiniz.



Eğer NodeMCU klon ise Ch340g driver'i kurmamız gerekli. Maker blog sayfamızda bu konuyla ilgili detaylı anlatımı mevcuttur.

(https://maker.robotistan.com/nodemcu-esp8266/)

01

Tarayıcı Üzerinden LED Kontrolü

BLYNK IoT
(IOS)

PROJELER LİNKİ

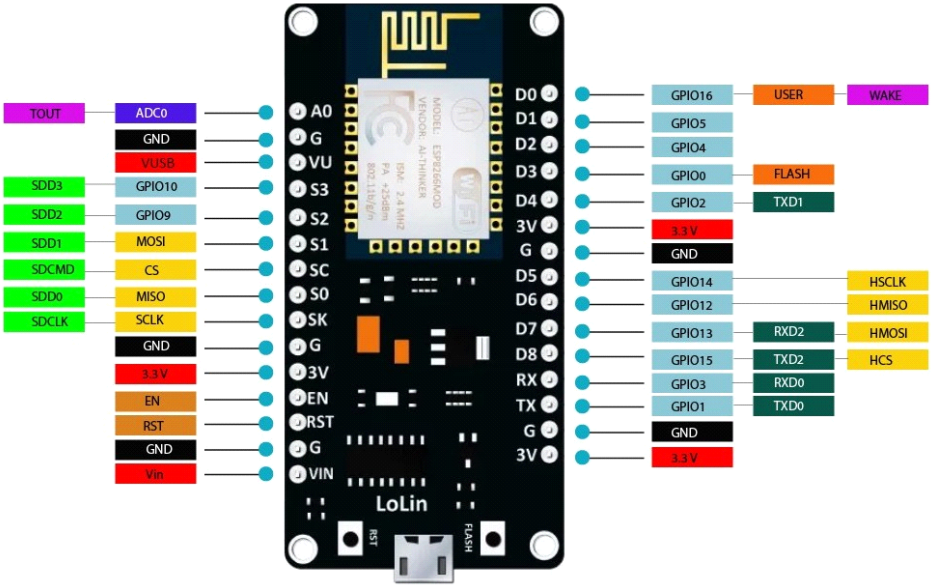
BLYNK IoT
(ANDROID)

http://rbt.ist/nodemcu

Maker'larin çok kullandığı cümleyle ilk projemize başlıyoruz. "Her proje led yakmakla başlar". İlk yapacağımız proje tarayıcı üzerinden led kontrol etmek olacak, IOT projelerine yeni başlayan birisi için oldukça ilgi çekici bir projedir.

NodeMCU geliştirme kartı ile proje yaparken, Arduino ide programını kullanacağız. Eğer bilgisayarınızda Arduino ide programı yüklü değilse, arduino.cc sitesine girerek indirebilirsiniz. Ya da indirilmesi gerekenler sayfasından indirebilirsiniz.

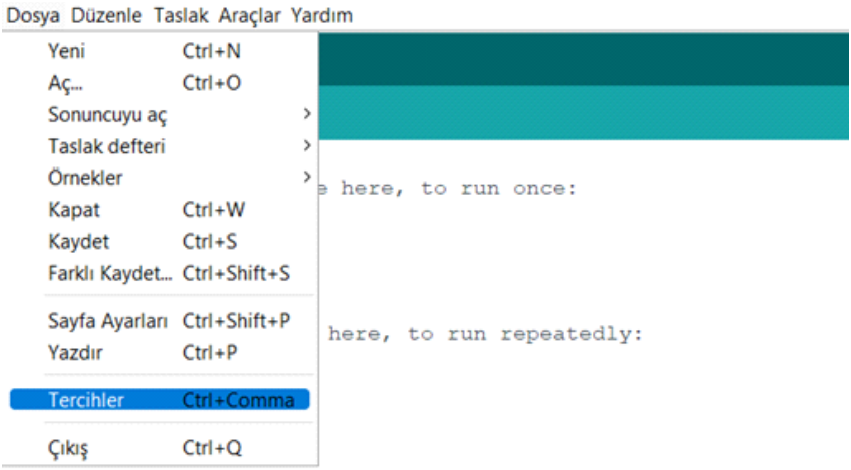
Gerekli Malzemeler
- NodeMCU Kartı
- 1 adet LED
- Breadboard
- Yeteri kadar jumper kablo (dişi-dişi)



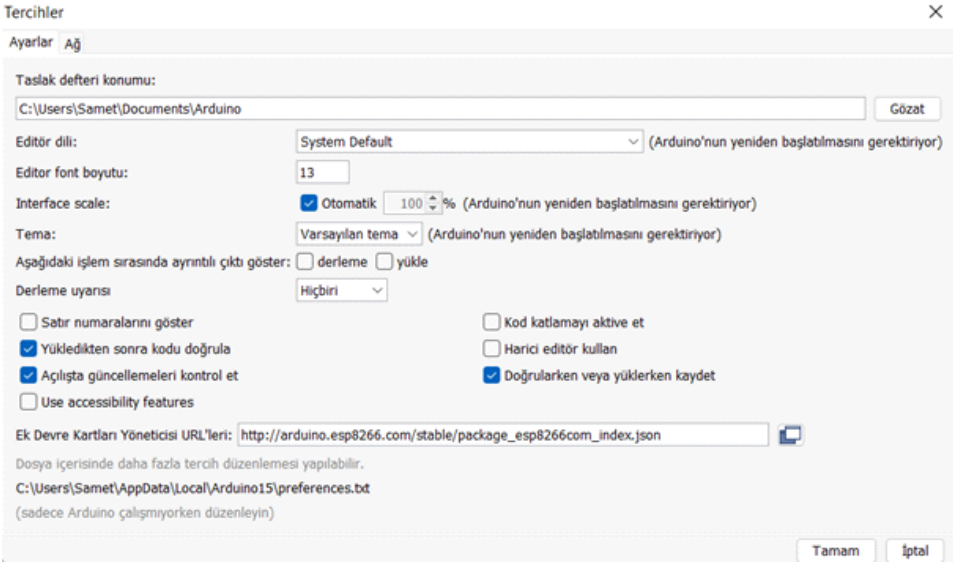
Yukarıdaki pinout(pin dağılımı) tablosu, yazılımı Arduino ide ile yazacağımız için NodeMCU unun pinlerini bu ara yüzde nasıl kullanacağımızı bilmemiz gerekiyor. Örneğimizde kullanacağımızgibi; mesela GPIO13 pini=D7 pinine karşılık geliyor. Yani biz yazılımda 13. pini kullandığımızda pinout tablosundan da bakacak olursak NodeMCU modülün de D7 pinini kullanmış oluyoruz.

DigitalWrite(13,HIGH); komutunu kullandığımız zaman NodeMCU modülündeki D7 pinine lojik 1 bilgisini göndermiş oluruz. Biz ledi D7 pinine bağladık, siz isterseniz yazılımda belirtmek şartıyla başka pinlere de bağlayabilirsiniz.

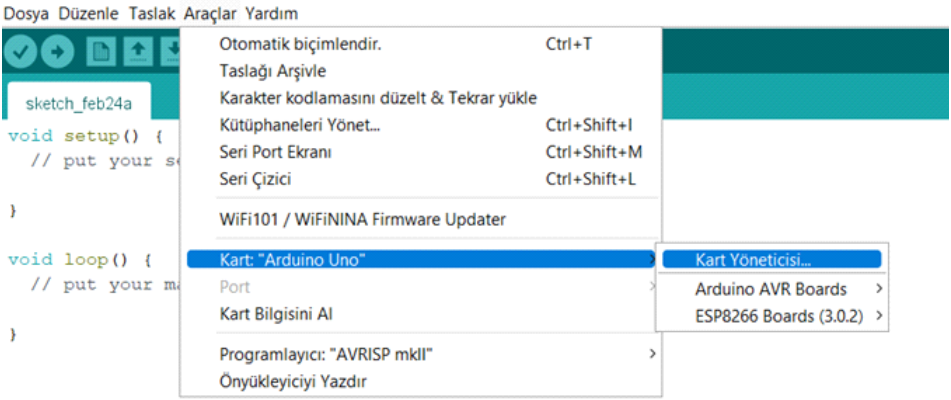
1) Yazılımı yazmadan önce arduino ide üzerinde birkaç ayar yapmamız gerekiyor. Arduino ara yüzünde varsayılan olarak NodeMCU modülü bulunmuyor. Modülü entegre etmek için, sol üst kısımda bulunan "**dosya menüsünden tercihler kısmına**"giriyoruz.



2) Açılan sekmede bulan ek devre kartları yöneticisi kısmına, bu linki el ile manuel şekilde girelim. "**http://arduino.esp8266.com/stable/package_esp8266com_index.json**" ve tamam diyelim.

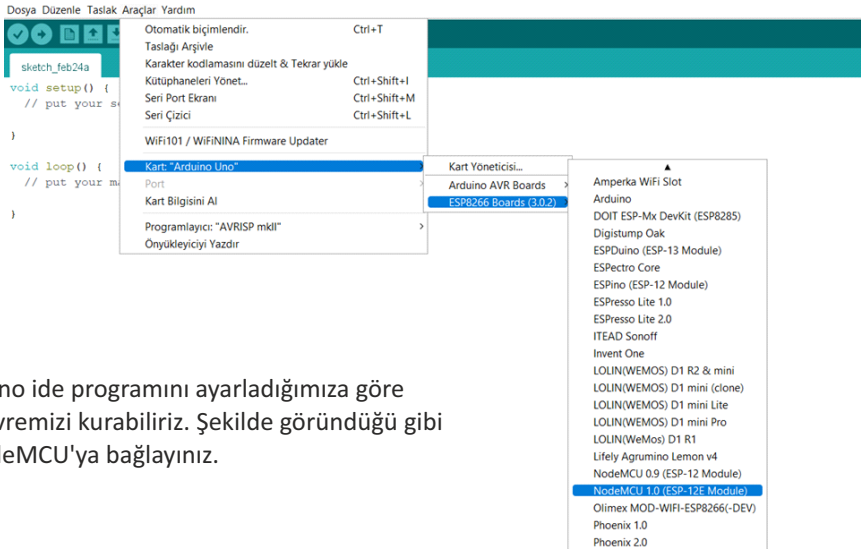


3) Daha sonra araçlar menüsünden kart yöneticisini açıyoruz.

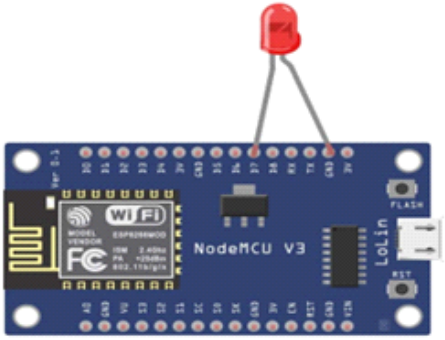


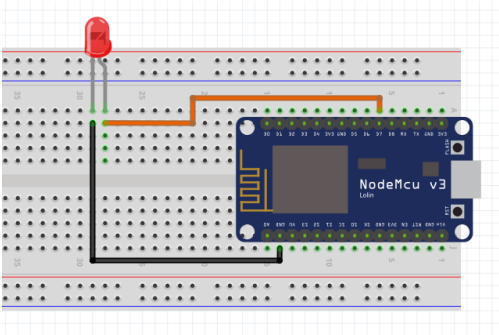
4) Açılan pencerede arama kısmına, esp8266 yazıp arama yapıyoruz. Daha sonra güncel versiyonunu seçip kuruyoruz.

5) Daha sonra kart(board) kısmından NodeMCU 1.0 (ESP-12E module) seçebiliriz. Bu ayarlamaları yaptıktan sonra artık kartımıza yazılım yükleyebiliriz.



6) Arduino ide programını ayarladığımıza göre artık devremizi kurabiliriz. Şekilde göründüğü gibi ledi NodeMCU'ya bağlayınız.





Tarayıcı Üzerinden Led Kontrolü	
LED(+)	D7
LED(-)	GND

Eğer NodeMCU klon ise Ch340g driver'i kurmamız gerekli. Maker blog sayfamızda bu konuyla ilgili detaylı anlatımı mevcuttur.

7) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz.

```
#include <ESP8266WiFi.h>

const char* ssid = "WİFİ ADI";
const char* password = "WİFİ ŞİFRESİ";

int ledPin = 13; // GPIO13 ucu D7 ye karsılık gelir
WiFiServer server(80);

void setup() {
Serial.begin(115200);
delay(10);

pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, LOW);

// Wifi bağlantısı
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
// server baslatılıyor
server.begin();
Serial.println("Server started");

// baglantı saglandıktan sonra serial monitorde bize ip adresini gösterecek
Serial.print("Use this URL to connect: ");
Serial.print("http://");
Serial.print(WiFi.localIP());
Serial.println("/");

}

void loop() {
// bir clien istemci baglı olup olmadığını kontrol ediyoruz
WiFiClient client = server.available();
if (!client) {
return;
}

// client ın bir data gondermesini bekliyoruz
Serial.println("new client");
while(!client.available()){
delay(1);
}

// gelen istekleri okuyoruz
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
```

```
client.flush();
int value = LOW;
if (request.indexOf("/LED-ACIK") != -1) {
digitalWrite(ledPin, HIGH);
value = HIGH;
}
if (request.indexOf("/LED-KAPALI") != -1) {
digitalWrite(ledPin, LOW);
value = LOW;
}
// bu kısımda html kodlarını internet arayüzüne yazıdırıyoruz.
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<html>");

client.print("Led suanda: ");
if(value == HIGH) {
client.print("ACIK");
} else {
client.print("KAPALI");
}
client.println("");
client.println("<a href=\"/LED-ACIK\"\"><button>led on </button></a>");
client.println("<a href=\"/LED-KAPALI\"\"><button>led off </button></a>");
client.println("</html>");

delay(1);
Serial.println("Client disonnected");
Serial.println("");}
```

8) Programı kartımıza yükledikten sonra serial monitörden takip edelim. Kod yüklendikten kısa süre sonra kartımızın wifi bağlantısı kuruldu ve aldığı IP numarasını bize gösterdi.

```
Connecting to blue
..
WiFi connected
Server started
Use this URL to connect: http://192.168.2.184/
```

Autoscroll | No line ending | 115200 baud | Clear output

9) Serial monitörde çıkan ip adresini internet tarayıcımıza yazıyoruz. Karşımıza gelen ekran da bulunan butonlar sayesinde ledi açıp kapatabilirsiniz

Bu projemizde yerel ağ üzerinden NodeMCU kartı ile bir ledi nasıl kontrol edebileceğimiz öğrenmiş olduk. Önümüzdeki projeler için ilk adımı atmış olduk.

02

RGB LED Kontrolü

BLYNK IoT
(IOS)

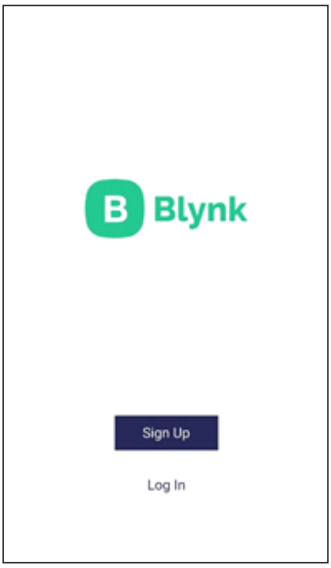
PROJELER LİNKİ

BLYNK IoT
(ANDROID)

http://rbt.ist/nodemcu

Bu projede Blynk uygulaması ile RGB diyot led kontrolünü yapacağız. Gece lambası ya da aksesuar gibi bir projeye çevirerek, evinizde güzel bir ortam elde edebilirsin. İlk olarak Blynk uygulamasının kurulumu ile başlayalım.

1)

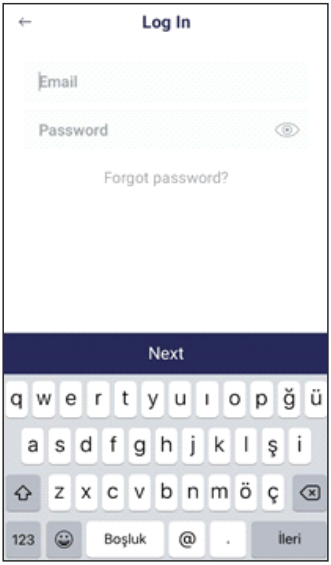


Uygulamayı telefonunuza indirdikten sonra ilk olarak karşınıza bu sayfa çıkıyor.

Buradan hızlıca bir hesap açabilir ya da Facebook hesabınız ile giriş yapabilirsiniz.

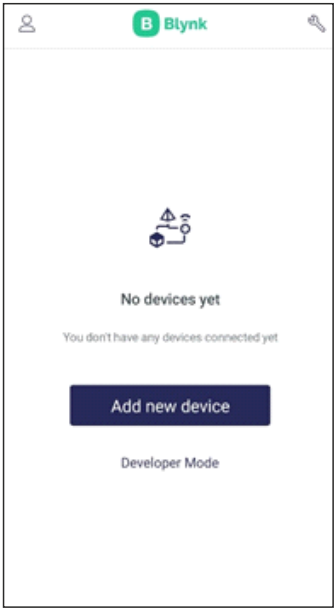
Ben hesap açmanızı tavsiye ederim çünkü oldukça kolay bir işlem.

2)



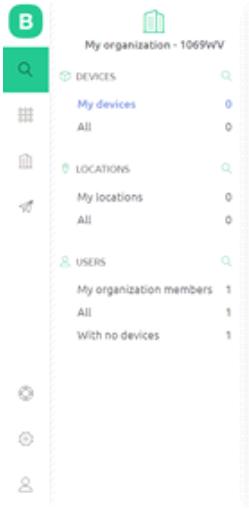
Hemen bir hesap oluşturuyoruz. Ve next butonuna dokunuyoruz.

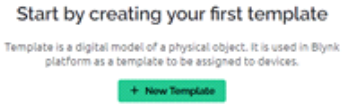
3)



2-) Hemen bir hesap oluşturuyoruz. Ve next butonuna dokunuyoruz.

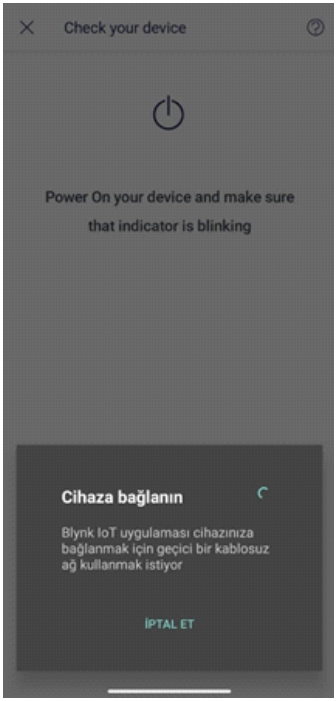
4) İlk projemizi oluşturmak için bilgisayarımızdan Google'a Blynk.Console yazıp aratıyoruz. Ardından sayfaya daha önce oluşturmuş olduğumuz hesap ile giriş yaparak New Template yazan kısma tıklayarak ilk projemizi başlatmış oluyoruz.



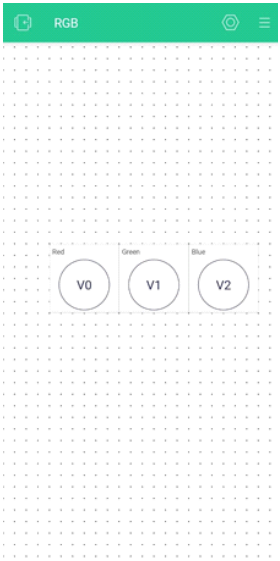


5) NodeMCU kartımız için ESP8266'yı seçiyoruz ve gerekli isimlendirmeleri yapıyoruz. Ardından Datastreams kısmından RGB ledimizin üç bacağı için gerekli pin atamalarını yapıp renk ve isim veriyoruz.

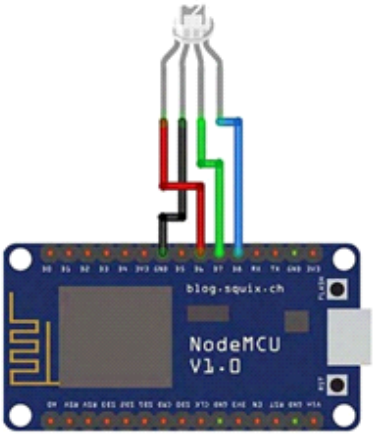
6) NodeMCU kartı ile telefonu eşleştirmek için telefonumuzdan Blynk uygulamasını açıp yeni cihaz ekle diyoruz. Ardından cihazımızı ve telefonumuzu Wİ-Fİ yardımı ile eşleştiriyoruz.

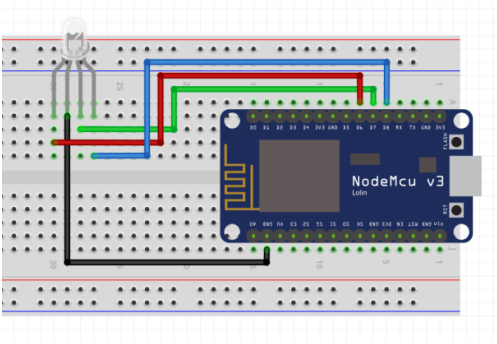


7) Eşleşmeyi tamamladıktan sonra Blynk uygulaması üzerinde buton tanımlamalarını yapıyoruz.



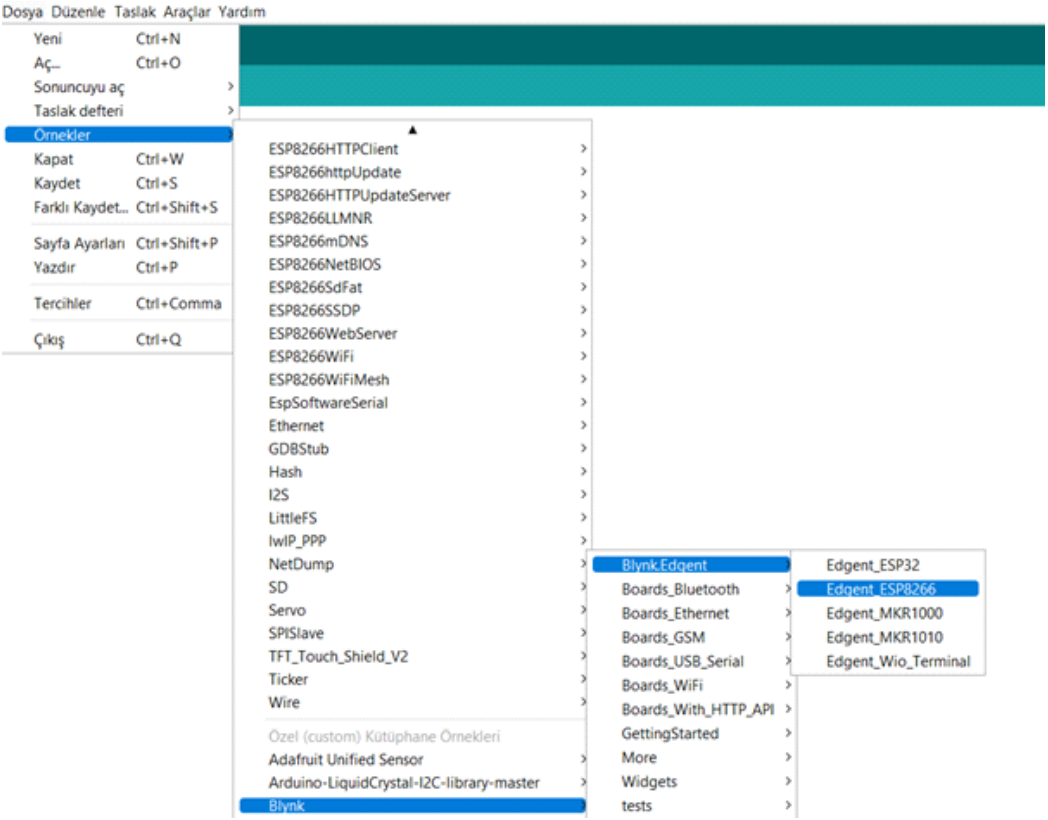
8) Eşleştirme işleminin tamamlanmasından sonra devre kuruluma geçiyoruz. Burada önemli olan ledin eksi ucun karıştırmamamız gerektiğidir. LED'in kırmızı renk pinini D6, yeşili D7, maviyi D8 pinlerine bağlayacağız.





RGB Led Kontrolü	
RGB Kırmızı	D6
RGB Yeşil	D7
RGB Mavi	D8
RGB (-)	GND

9) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.
(Kütüphaneyi " **https://github.com/blynkkk/blynk-library** " bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)



10) Kod kısmında bulunan BLYNK TEMPLATE ID ve DEVICE NAME herkesin kendine özgü olup Google üzerinden Blynk sitesine girilerek info kısmından alınır. Kodu yükledikten sonra proje tamamlanmış olacaktır.

rgbled § | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
#define BLYNK_TEMPLATE_ID "TMPL1gF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#define APP_DEBUG
#include "BlynkEdgent.h"
int ledpin  = 13;
int ledpin1 = 12;
int ledpin2 = 14;
  BLYNK_WRITE(V19){
    int pinvalue = param.asInt();
    digitalWrite(ledpin,pinvalue);
}
  BLYNK_WRITE(V20){
    int pinvalue = param.asInt();
    digitalWrite(ledpin1,pinvalue);
}
  BLYNK_WRITE(V21){
    int pinvalue = param.asInt();
    digitalWrite(ledpin2,pinvalue);
}
void setup() {
  Serial.begin(115200);
  delay(100);
  pinMode(ledpin, OUTPUT);
  pinMode(ledpin1, OUTPUT);
  pinMode(ledpin2, OUTPUT);
  BlynkEdgent.begin();
}
void loop() {
  BlynkEdgent.run(); }
```

DHT11 ile Sıcaklık ve Nem Ölçümü

BLYNK IoT
(IOS)

PROJELER LİNKİ

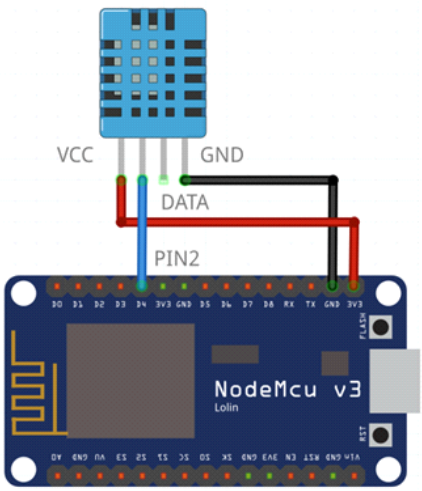
BLYNK IoT
(ANDROID)

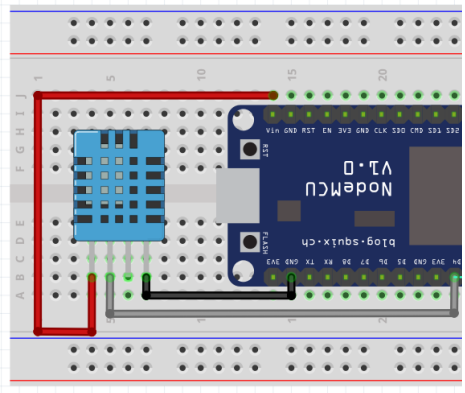
http://rbt.ist/nodemcu

Bu proje tamamlandığında herhangi bir ortamın, sıcaklık ve nem değerlerini blynk uygulaması üzerinden takip edebilecekseniz.
Arduino içinde varsayılan olarak NodeMCU modülü gelmemektedir. NodeMCU modülünü arduino ide'ye entegre etmek için, kitabın ilk projesinde anlatmıştık. Oradan bakarak NodeMCU'yu nasıl entegre edebileceğinizi öğrenebilirsiniz.

1)Projemize ilk olarak devre şemasını kurarak başlayalım.

Gerekli Malzemeler:
- NodeMCU wifi geliştirme kartı
- Dht11 sıcaklık nem sensörü
- Breadboard
- Yeteri kadar jumper kablo

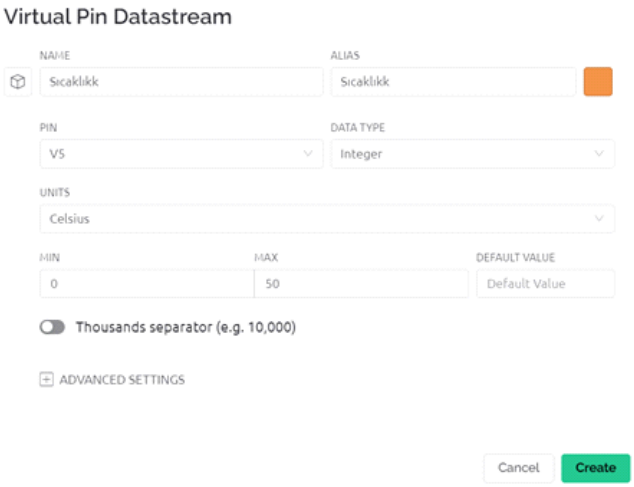




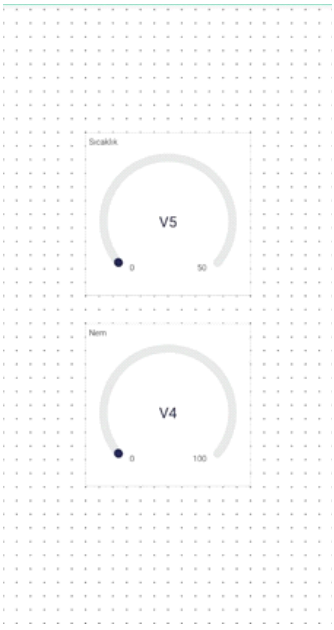
DHT11 İle Sıcaklık Ve Nem Ölçümü	
DHT Pin	D4
VCC	3.3V
(-)	GND

Bağlantı şemasında da gözüktüğü üzere sensörün data pini D4 e bağlıyoruz.

2) Google'a Blynk Console yazıp açılan sayfaya hesabımızla giriş yapalım. Ardından DataStreams kısmından sıcaklık ve nem için bir tanımlama yapalım.



3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş yapıp nem ve sıcaklık için Gauge ekliyoruz. Doğru pinleri eşleştirdiğimizi unutmayalım.

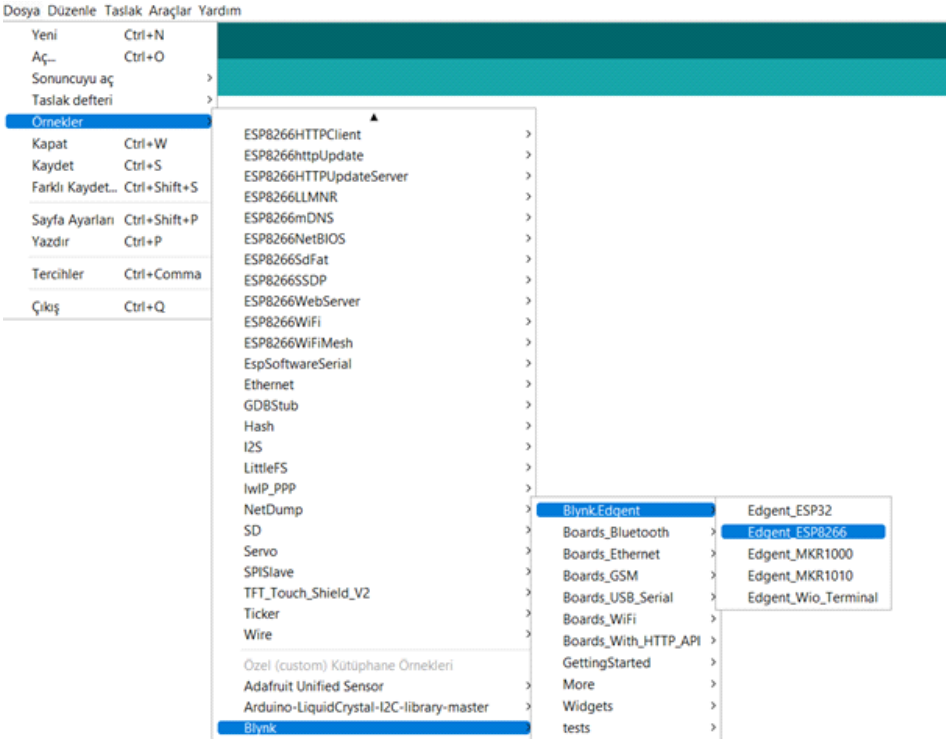


Nem sensörümüz bize normalde 0-1023 arası bir değer verecektir. Gauge de eklediğimizde karşımıza bu sayı aralığı çıkacaktır. Nem aralığı olarak bu aralığı görmek istemediğimiz için yazılımda bunu istediğimiz değer aralığına dönüştüreceğiz. Bu 0-1023 ile 0-50 arasındaki ilişkiyi yazılım kısmında yapacağız ve V4 pininden okuma yapacağız.
Aynı işlemleri sıcaklık gaugesi içinde yapıyoruz. Yalnız buradaki pinimizi V5 olarak ayarlamayı unutmuyoruz.

NOT: Blynk uygulamasında daha gelişmiş widgetlar kullanılabilir fakat bu widgetları kullanabilmek için uygulamayı pro sürümüne yükseltmeniz gerekmektedir.

4) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi " **https://github.com/blynkkk/blynk-library** " bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)



5) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz.

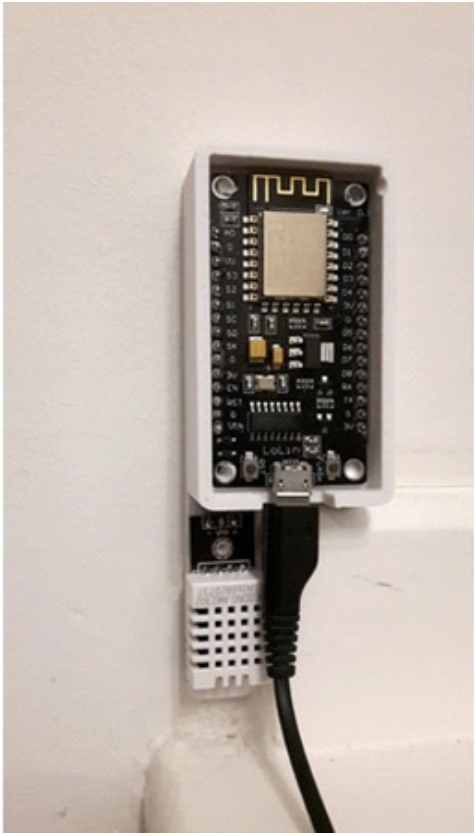
dht11 § | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h

```
#define BLYNK_TEMPLATE_ID "TMPLlgF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
#define APP_DEBUG
#define DHTPIN 12 //D4 pin of Node MCU
#define DHTTYPE DHT11
#define MODE
#include "BlynkEdgent.h"
#include <DHT.h>

int value = 200;

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer1;
float h , t;

void sendSensor()
{
 h = dht.readHumidity();
 t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V2, h);
  Blynk.virtualWrite(V1, t);
}
```

```
void setup()
{
  Serial.begin(115200);
  delay(100);
  dht.begin();
  BlynkEdgent.begin();
  timer1.setInterval(2000L, sendSensor);
}

void loop()
{

 BlynkEdgent.run();
 timer1.run();

  h = dht.readHumidity();
  t = dht.readTemperature();

  if (t>0 && t<15)
  {
    Serial.print("COLD   ");

  }
  else if (t>14 && t<30)
  {
    Serial.print("WARM   ");

  }

  else if (t>30 && t<45)
  {
    Serial.print("HOT   ");
    ;
  }

  else
  {
    Serial.print("VERY HOT  ");

  }

  if (h>0 && h<30)
  {
    Serial.println("LOW HUMIDITY");

  }

  else if (h>30 && h<50)
  {
    Serial.println("Normal HUMIDITY");

  }
  else
  {
    Serial.println("HIGH HUMIDITY");

  }
  delay(800);
}
```

Bu projeyi bitirdikten sonra, güzel bir kutuya koymak istemez misiniz?

İşte bu yüzden 3d yazıcı ile basılabilecek bir stl dosyasının linkini sizlerle paylaştık. 3d yazıcı ile bu kutuyu basıp projenizi güzel bir şekilde muhafaza edebilirsiniz. Ayrıca bu çizimi isterseniz kendinize göre revize edebilirsiniz, belki kendi markanızı bu kutunun üstüne basmak isteyebilirsiniz.
“**www.thingiverse.com**“adresindeki arama kısmına 2201956 yazarak çizim dosyalarına ulaşabilirsiniz.



04

Uzaktan Röle Kontrolü

BLYNK IoT (IOS)

PROJELER LİNKİ

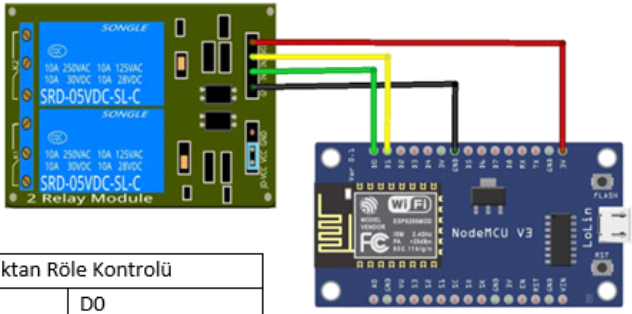
BLYNK IoT (ANDROID)

http://rbt.ist/nodemcu

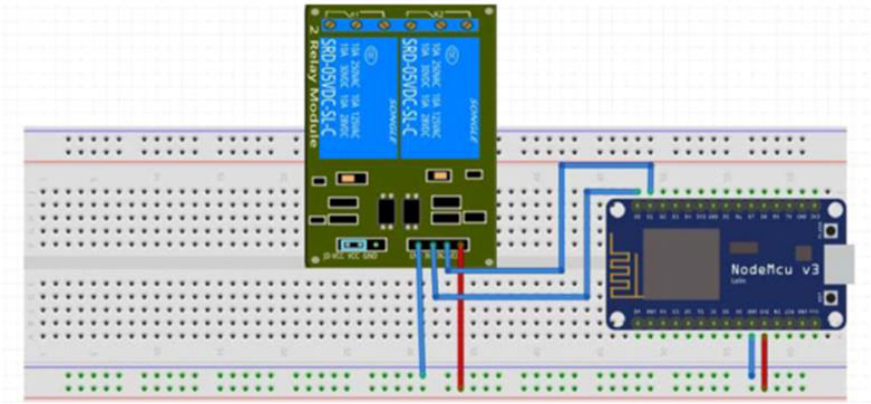
Bu projede “Blynk” uygulaması ile röle açıp kapatma uygulamamızı gerçekleştireceğiz. Bu rölenin uçlarına ne bağlarsanız (klima, lamba, motor) evinizden, iş yerinizden kontrol edebileceksiniz. Bu projeyi yaparken röleye bağlayacağınız yüksek voltaj ile çalışan cihazların bağlantılarını yaparken dikkatli olunuz.

Gerekli Malzemeler:
- NodeMCU Wifi geliştirme kartı
- 2 kanal röle modülü
- Breadboard
- Yeteri kadar jumper kablo

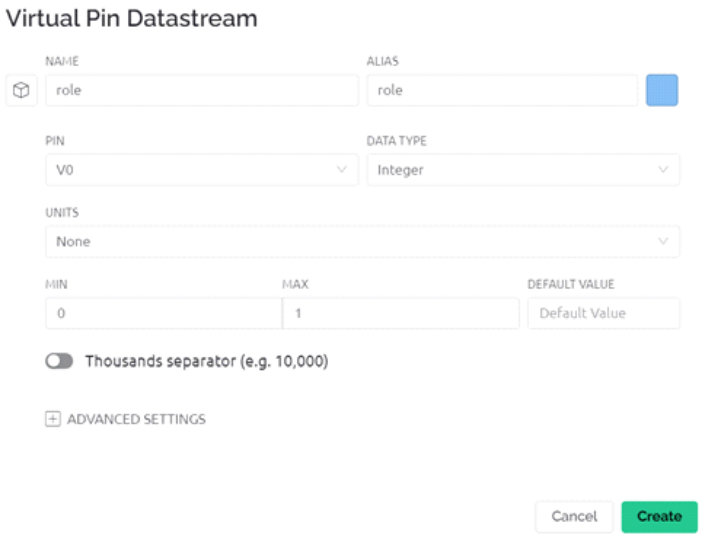
1) İlk olarak bağlantı şemamızı oluşturalım. D0 ve D1 pinlerimizi röleye bağlantı için kullanacağız. VCC kısmını 3.3V ile birleştirerek devremizi kurmuş oluyoruz.



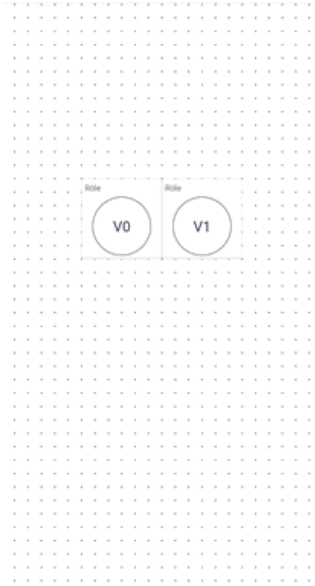
Uzaktan Röle Kontrolü	
Röle Pin 1	D0
Röle Pin 2	D1
VCC	3.3V
(-)	GND



2) Google'a Blynk Console yazıp açılan sayfaya hesabımızla giriş yapalım. Ardından DataStreams kısmından rölemiz(V0 ve V1 Pinleri) için bir tanımlama yapalım.

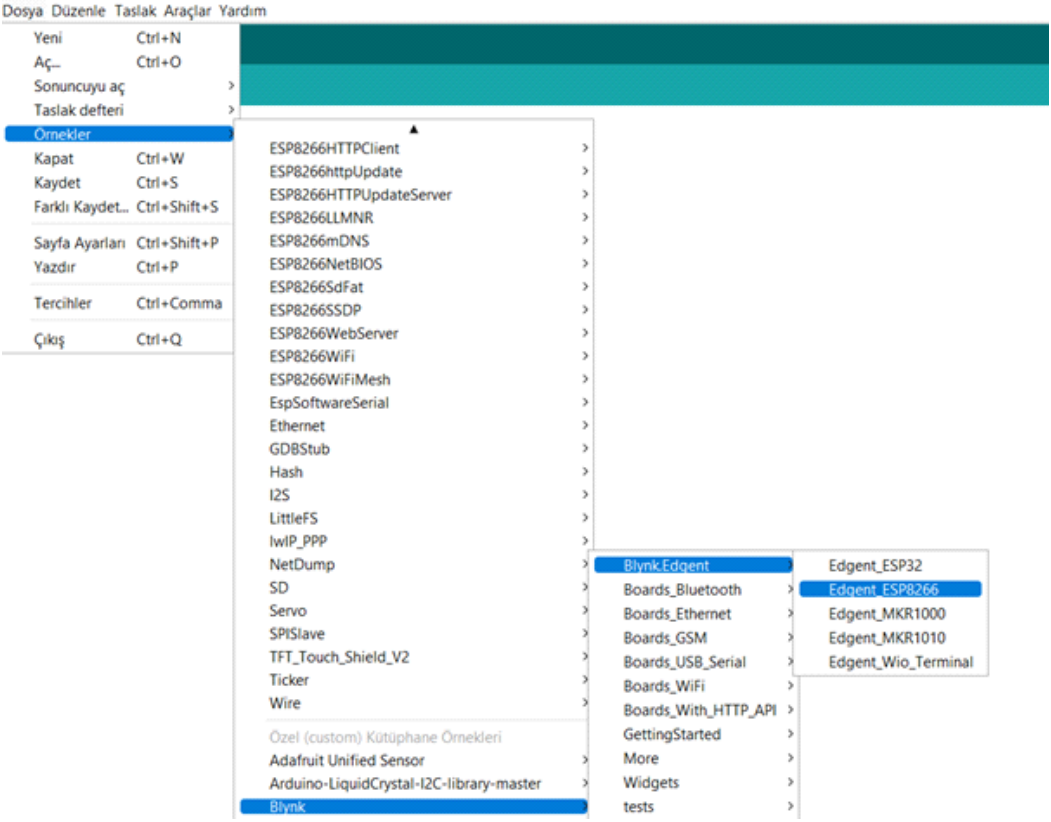


3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş yapıp röle için butonlar ekliyoruz. Doğru pinleri(V0 ve V1) eşleştirdiğimizi unutmayalım.



4) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi " **https://github.com/blynkkk/blynk-library** " bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)

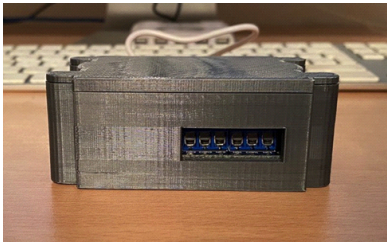


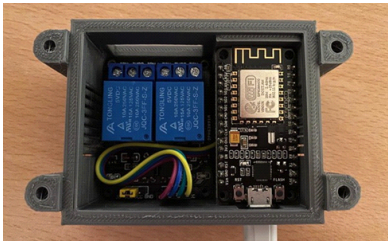
5) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz. Kodumuza yükleme yapmadan önce kendimize özgü name ve ID kısmını girmeyi unutmayalım.

```
r_le § | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h
#define BLYNK_TEMPLATE_ID "TMPLlgF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#define APP_DEBUG
#include "BlynkEdgent.h"

BLYNK_WRITE(V1)
{
  int pinValue = param.asInt();
  digitalWrite(5,pinValue);
}
BLYNK_WRITE(V0)
{
 int pinValue = param.asInt();
  digitalWrite(4,pinValue);
}
void setup()
{
  Serial.begin(115200);
  delay(100);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  BlynkEdgent.begin();
}
void loop() {
  BlynkEdgent.run();
}
```

Projeyi güzel bir kutuya koyabilmeniz için 3d yazıcı ile basılabilecek bir stl dosyasının linkini sizlerle paylaştık. 3d yazıcı ile bu kutuyu basıp projenizi güzel bir şekilde muhafaza edebilirsiniz. Ayrıca bu çizimi isterseniz kendinize göre revize edebilirsiniz, belki kendi markanızı bu kutunun üstüne basmak isteyebilirsiniz.

www.thingiverse.com adresindeki arama kısmına **4093462** kodunu yazarak cizim dosyalarına ulaşabilirsiniz.





05

Toprak-Nem Sensörü Uygulaması

BLYNK IoT
(IOS)

PROJELER LİNKİ

BLYNK IoT
(ANDROID)

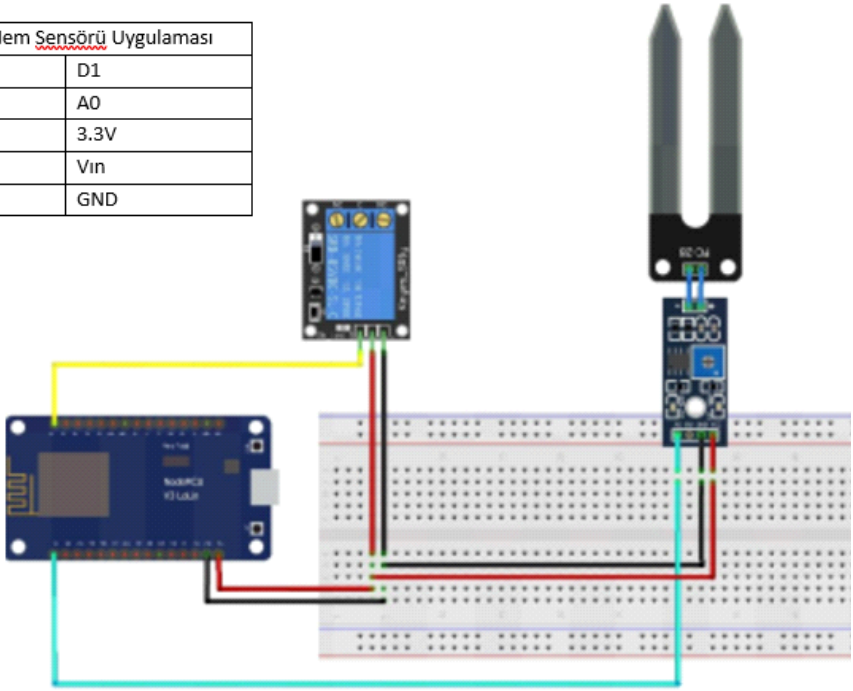
http://rbt.ist/nodemcu

Bu projede Wifi üzerinden Blynk uygulaması ile toprak-nem sensörünün okuduğu değeri görebileceğiz. Bu gördüğümüz değere bağlı olarak röle ile birlikte çıktı alabileceğiz.

1) İlk olarak devre kurulumu ile başlayalım.

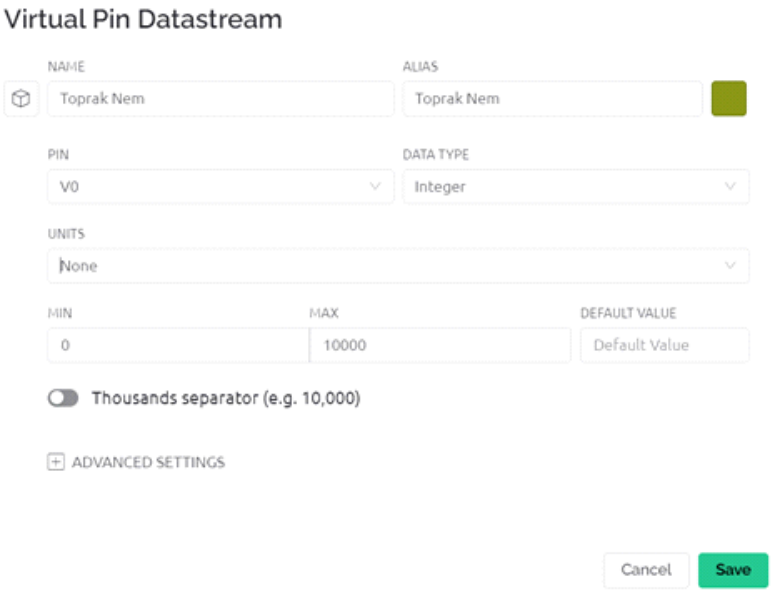
Gerekli Malzemeler :
- NodeMCU
- Breadboard
- Jumper kablolar
- 2 kanal röle modülü
- Toprak-Nem sensörü

Toprak-Nem Sensörü Uygulaması	
Röle Pin	D1
Sensör Pin	A0
Röle VCC	3.3V
Sensör VCC	Vın
(-)	GND

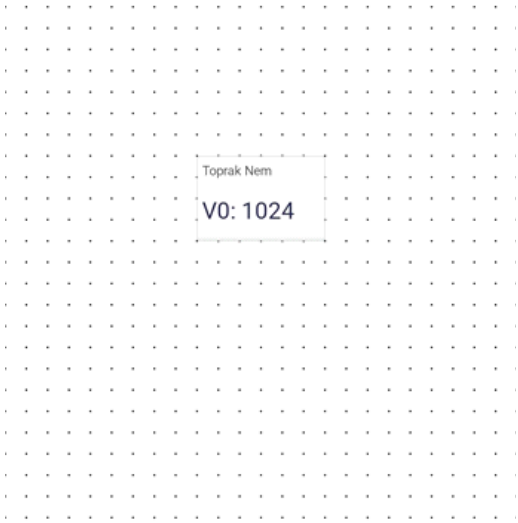


Röle bağlantısı için sinyal pinini D1 kısmına, VCC kısmını ise 3.3V pinine bağlıyoruz. Röle bağlantısı tamamlandıktan sonra toprak nem sensörü için sinyal pinini anlog pin olan A0 pinine, VCC kısmını ise Vin pinine bağlayarak devre kurulumunu tamamlamış oluyoruz.

2) Google'a Blynk Console yazıp açılan sayfaya hesabımızla giriş yapalım. Ardından DataStreams kısmından V0 pini için bir tanımlama yapalım.

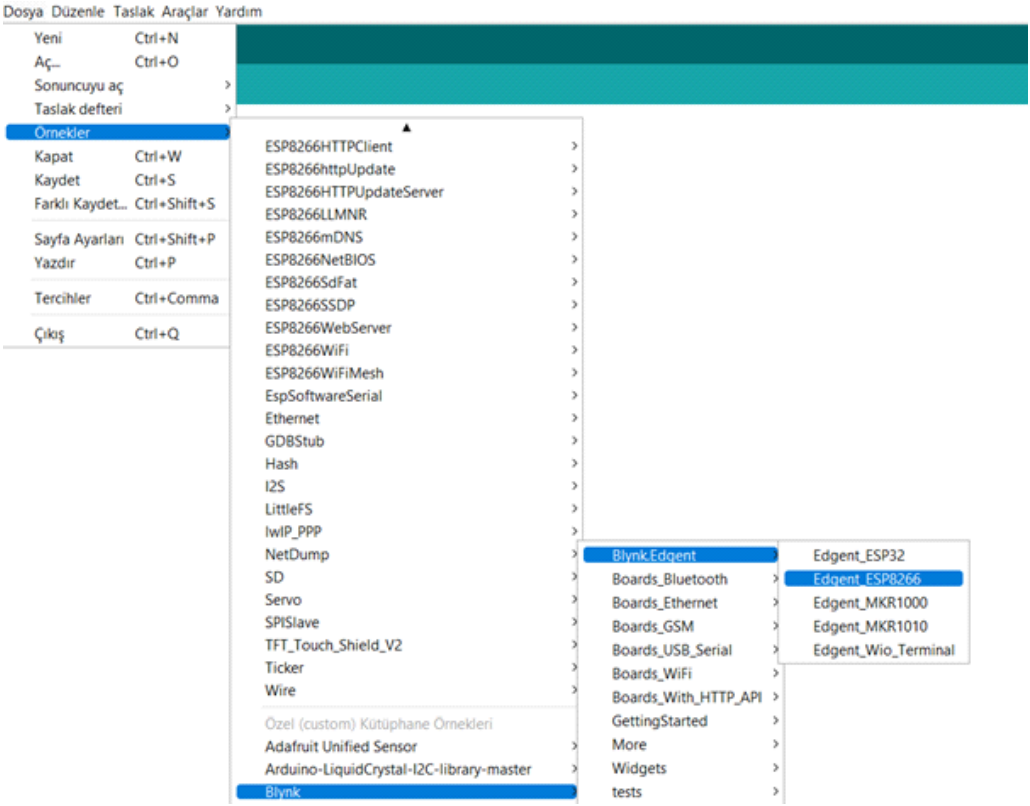


3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş yapıp toprak nem sensörünün değerini görmek için labeled value ekliyoruz.



4) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi " **https://github.com/blynkkk/blynk-library** " bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)



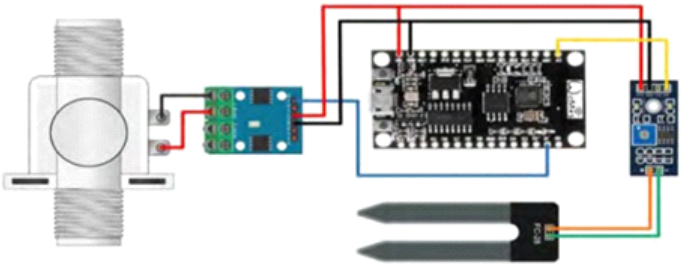
5) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz. Kodumuza yükleme yapmadan önce kendimize özgü name ve ID kısmını girmeyi unutmayalım.

Edgent_ESP8266 § | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
#define BLYNK_TEMPLATE_ID "TMPLlgF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION        "0.1.0"

#define BLYNK_PRINT Serial
#define BLYNK_DEBUG

#define APP_DEBUG
#include "BlynkEdgent.h"

int soilmoisture;
int analogInput = A0;
int relay = D1;

BLYNK_WRITE(V0)
{
  int pinValue = param.asInt(); // assigning incoming value from pin A0 to a variable
}
void setup() {

pinMode(analogInput, INPUT);
pinMode(relay, OUTPUT);
Serial.begin(115200);
  delay(100);

  BlynkEdgent.begin();
}

void loop() {
Blynk.run();
soilmoisture = analogRead(analogInput);
Serial.println(soilmoisture);
Blynk.virtualWrite(V0,soilmoisture);
BlynkEdgent.run();
if(soilmoisture>950)
{
  digitalWrite(relay,HIGH);
}
else
{
  digitalWrite(relay,LOW);
}
}
```

NOT: Toprak nem sensörü ve röle ile birçok proje geliştirilebilir. Rölenin çıkış pinlerine motor sürücü, led, buzzer vb. gibi elektronik elemanlar bağlanıp yeni projeler elde edilebilir.

Örnek vermek gerekirse setimizin içinden çıkan L9110 Çift motor sürücü ve dalgıç pompa ile çiçek sulama projesi yapılabilir. Dilerseniz buzzer ve led gibi elektronik devre elemanlarıyla devremizi şekillendirip telefon üzerinden Blynk uygulaması ile kontrol edebilirsiniz.



Toprak-Nem Sensörü Uygulaması	
Röle Pin	D1
Sensör Pin	A0
Röle VCC	3.3V
Sensör VCC	Vın
(-)	GND

06

Servo Motor Kontrolü

BLYNK IoT (IOS)

PROJELER LİNKİ

BLYNK IoT (ANDROID)

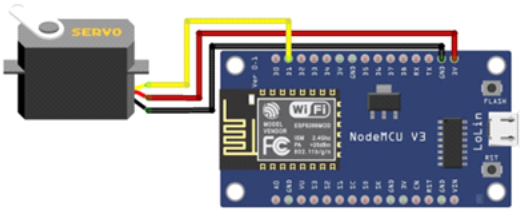
http://rbt.ist/nodemcu

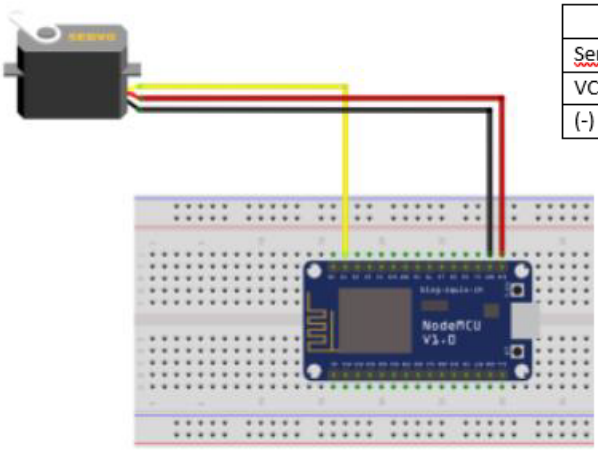
Bu projede wifi üzerinden Blynk uygulamasıyla servo motor kontrolü projesi yapacağız. Servo motorun pervanesini taktığımızda kaç derece döndürebildiğimizi rahat bir şekilde görebilecek duruma geleceğiz.

1) İlk olarak devre kurulumu ile başlayalım.

Gerekli Malzemeler :
- NodeMCU
- Servo Motor
- Breadboard
- Jumper Kablo(Dişi-Erkek)

Servo motor 5v ile çalıştığı için artı ucunu NodeMCU'nun VIN girişine bağlıyoruz. Bu kısıma dikkat etmek gereklidir çünkü 3.3V pinine bağlanır ise servo motorumuz çalışmayacaktır. Sinyal ucunu D1, eksi ucunu gndye bağlayıp devre kurulumunu tamamlıyoruz.

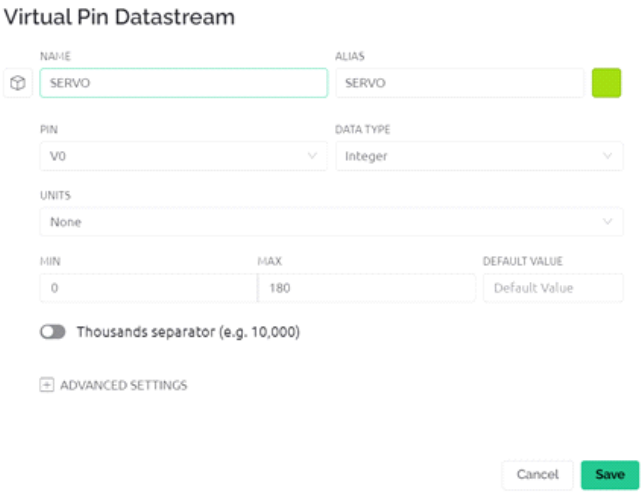




Servo Motor Kontrolü	
Servo Pin	D1
VCC	Vın
(-)	GND

NOT: Servo Motorumuzun pervane montajını yapmayı unutmayalım.

2) Google'a Blynk Console yazıp açılan sayfaya hesabımızla giriş yapalım. Ardından DataStreams kısmından servo motorumuz(V0 Pini) için bir tanımlama yapalım.

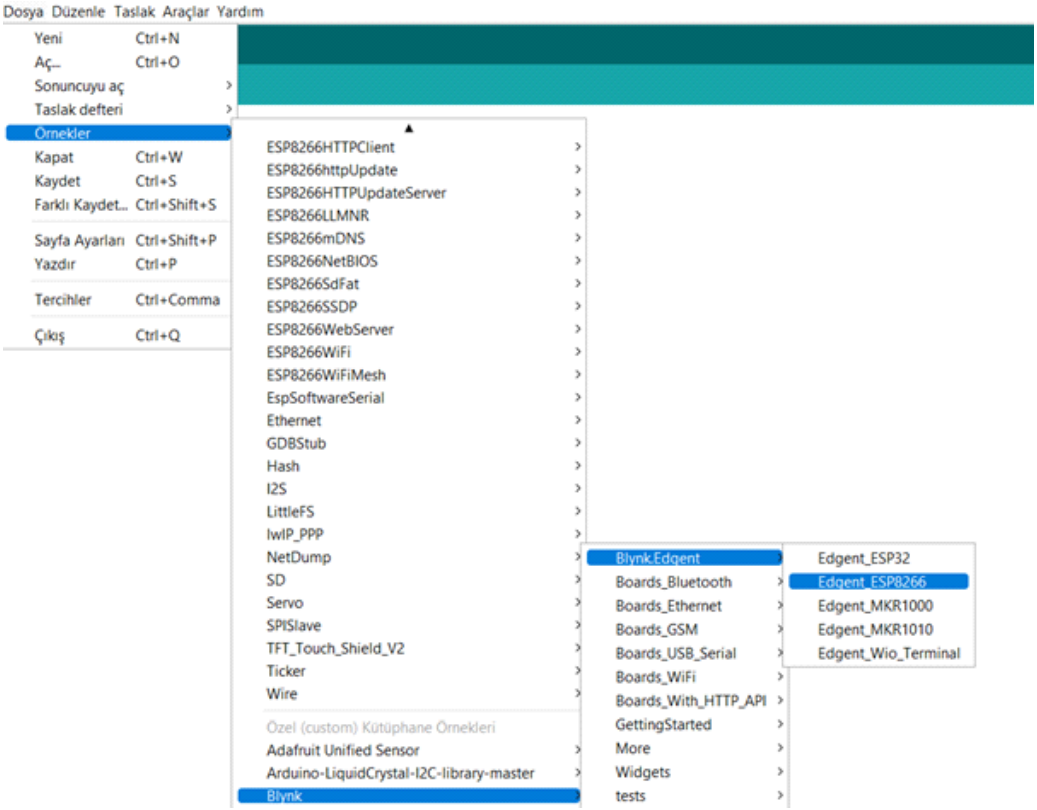


Servo motorun yapısı gereği 180° dönebildiği için max kısmına olabilecek en yüksek dereceyi verdik. Siz tanımlama yaparken 90° veya istediğiniz bir açıda döndürme işlemi yapabilirsiniz.

3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş yapıp servo motoru telefonumuzdan döndürebilmek için Slider widgetını ekliyoruz. Daha önceki projelerde bahsettiğimiz gibi Blynk uygulamasını pro moda yükselterek daha farklı widgetlara erişebilirsiniz. Doğru pinleri(V0) eşleştirdiğimizi unutmayalım.

4) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi " **https://github.com/blynkkk/blynk-library** " bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)



5) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz. Kodumuza yükleme yapmadan önce kendimize özgü name ve ID kısmını girmeyi unutmayalım.

servo1 § | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
#define BLYNK_TEMPLATE_ID "TMPLlgF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#define APP_DEBUG

#include "Servo.h"
#include "BlynkEdgent.h"

Servo servo;

BlynkTimer timer;

BLYNK_WRITE(V0){
  servo.write(param.asInt());
}

void setup()
{
   Serial.begin(115200);
   servo.attach(D1);
   BlynkEdgent.begin();
}

void loop() {
  BlynkEdgent.run();


}
```

07

Duman - Gaz Dedektörü

BLYNK IoT (IOS)

PROJELER LİNKİ

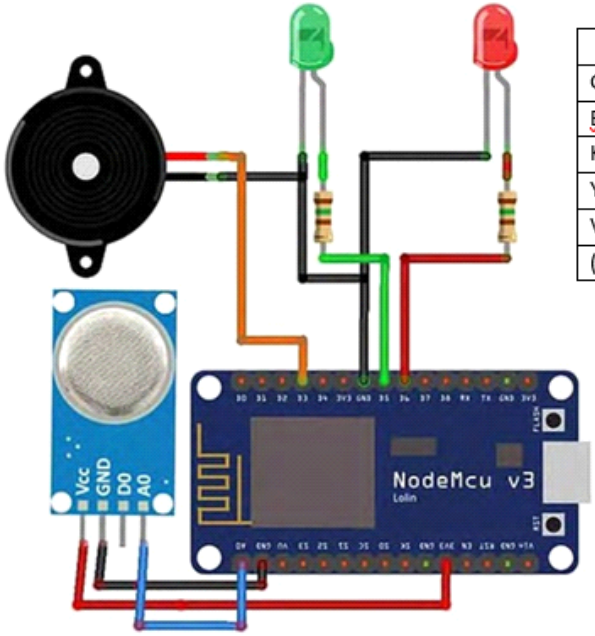
BLYNK IoT (ANDROID)

http://rbt.ist/nodemcu

Bu projede telefonunuzdan Blynk programını kullanarak wifi ağı üzerinden NodeMCU kartı sayesinde Gaz sensöründen gelen veriler ile ortamın gaz ve duman seviyesini ölçebileceksiniz, belli seviye üzerinde telefonumuza bildirim gelecek ve buzzer çalışacaktır. Böylelikle ortamdaki havanın ne kadar temiz olduğuna karar verebileceksiniz.

Gerekli Malzemeler:
- NodeMCU
- 2 adet Led
- 2 adet 150 ohm Direnç
- MQ2 Gaz Sensörü
- Buzzer
- Breadboard
- Jumper Kablolar

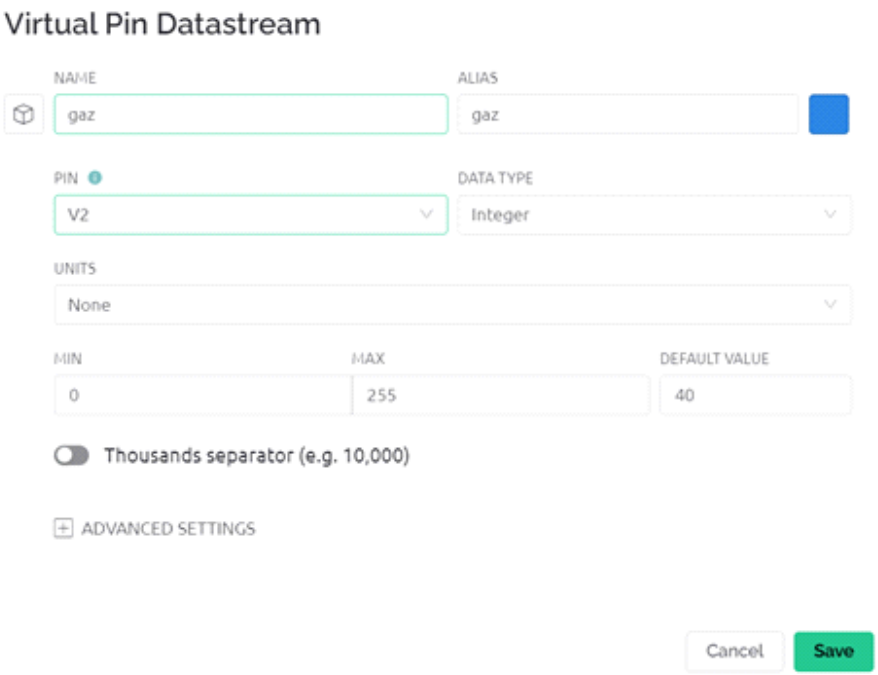
1) Devre şemasında gözüktüğü gibi bağlantıları yapalım. Ledleri D5 ve D6 pinlerine, buzzerı ise D3 pinine bağlayalım. Ayrıca gaz sensörünü de analog A0 pinine bağlayalım. Gaz sensörünü Analog bir pine bağlamaya dikkat edelim.



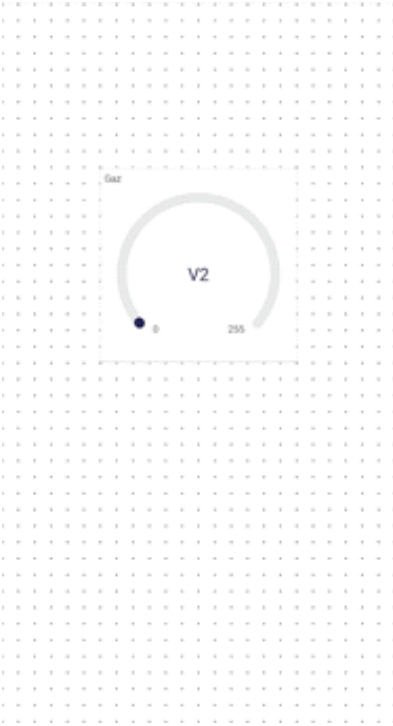
Duman – Gaz Dedektörü	
Gaz Sensörü Pin	A0
Buzzer Pin	D3
Kırmızı Led Pin	D6
Yeşil Led Pin	D5
VCC	3.3V
(-)	GND

Devre kurulumu tamamladıktan sonra Blynk uygulamasındaki ayarlamalara geçeceğiz. Daha önceden Blynk uygulamasının ayarlarını yapmıştık. Eğer herhangi bir sorun ile karşılaşıyorsanız ilk projemizde nasıl kurulum yapılacağından bahsetmiştik.

2) Google'a Blynk Console yazıp açılan sayfaya hesabımızla giriş yapalım. Ardından DataStreams kısmından gaz sensörmüz(V2 Pini) için bir tanımlama yapalım.



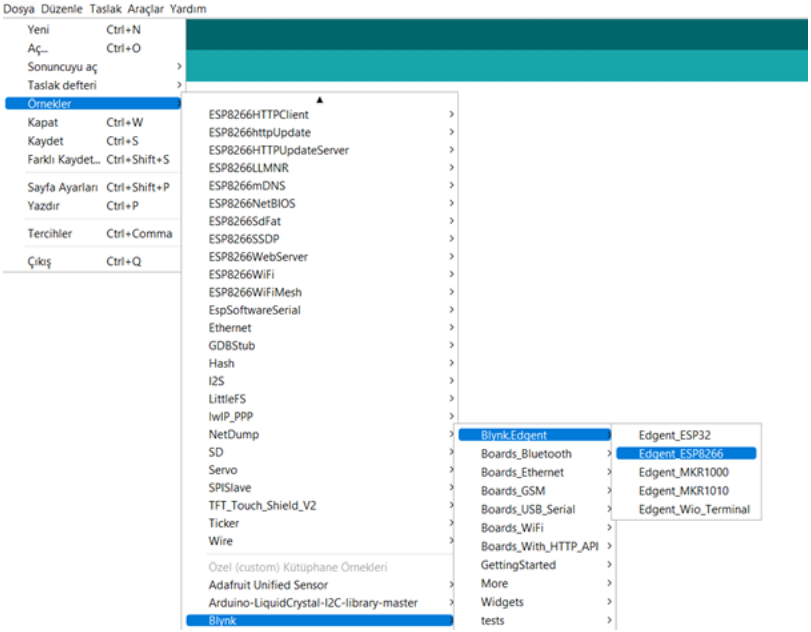
3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş gas sensörümüz için gauge ekliyoruz.. Daha önceki projelerde bahsettiğimiz gibi Blynk uygulamasını pro moda yükselterek daha farklı widgetlara erişebilirsiniz. Doğru pinleri(V2) eşleştirdiğimizi unutmayalım.



4) Blynk uygulamasının bize bilidirim göndermesi için notification'u ekliyoruz. Bütün ayarları tamamladıktan sonra sağ üst köşede bulunan play butonuna basarak gaz sensöründen bilgi almaya başlayabiliriz. Kod kısmına geçmeden önce devre ve Blynk tasarımlarınızı kendinize göre şekillendirebilirsiniz.

5) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklere gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi **https://github.com/blynkkk/blynk-library** bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)



6) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz. Kodumuza yükleme yapmadan önce kendimize özgü name ve ID kısmını girmeyi unutmayalım.

gas_sebs_r_§ | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
#define BLYNK_TEMPLATE_ID "TMPLlgF2jGe-"
#define BLYNK_DEVICE_NAME "blink led"
#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#define APP_DEBUG
#include "BlynkEdgent.h"

BlynkTimer timer;

int GaSSensor = A0;
int data = 0;
#define GreenLed 14
#define RedLed 12
int buzzer = 0;

void setup()
{
   Serial.begin(115200);
  pinMode(GreenLed, OUTPUT);
  pinMode(RedLed, OUTPUT);
  timer.setInterval(1000L, getSendData);

  BlynkEdgent.begin();
}
void loop() {
  BlynkEdgent.run();
  timer.run();

}
void getSendData()
{
data = analogRead(GasSensor);
  Blynk.virtualWrite(V2, data);

 if (data > 300)
  {
    Blynk.notify("Attention !!! Your Gas Sensor Worked");
    digitalWrite(GreenLed, HIGH);
    digitalWrite(RedLed, LOW);
    tone(buzzer, 1000, 200);
  }

  else
  {
    digitalWrite(GreenLed, LOW);
    digitalWrite(RedLed, HIGH);
    noTone(buzzer);
  }
}
```

7) Arduino IDE'nin serial port ekranından gas sensörümüzün çalışıp çalışmadığı kontrol edebiliriz.

COM14

```
09:07:44.713 -> Dikkat!!! Gaz Sensörünüz Çalıştı
```

YouTube

youtube.com/robotistan

FORUM

robotistan

forum.robotistan.com

BLOG

robotistan

maker.robotistan.com

Robotistan Elektronik Ticaret A.Ş.

Hazırlayanlar: Samet SÜLÜN (İçerik) - Yasin TAŞCIOĞLU - Mehmet AKÇALI (Editör) - Mehmet Nasır KARAER (Grafik)
info@robotistan.com - www.robotistan.com
Tel: 0850 766 0 425