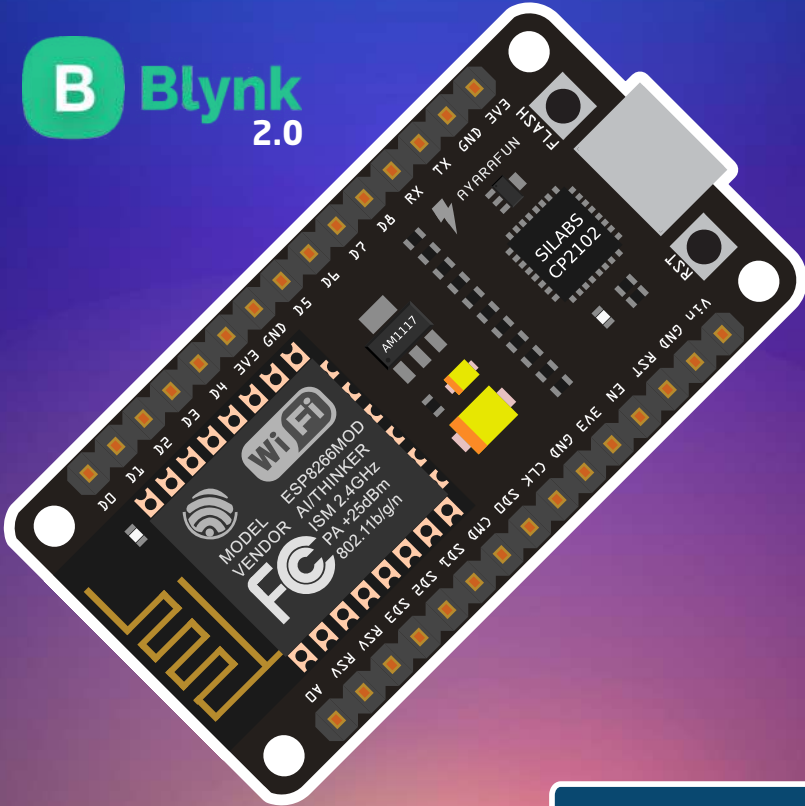




NodeMCU IoT Proje Geliştirme Kitabı

B 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 açacaksınız. 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.....	5
NodeMCU nedir?.....	8
Tarayıcı Üzerinden Led Kontrolü.....	9
Rgb Led Kontrolü.....	17
Dht11 İle Sıcaklık Ve Nem Ölçümü.....	29
Uzaktan Röle Kontrolü.....	34
Toprak-Nem sensörü uygulaması.....	40
Servo Motor Kontrolü.....	46
Duman - Gaz Dedektörü.....	51

Set İçeriğini Tanıyalım



NodeMCU LoLin ESP8266 Geliştirme Kartı

Üzerinde NodeMCU firmware yüklü ESP8266 WiFi modül barındıran bir geliştirme kartıdır. ESP8266 SDK'sı kullanılarak geliştirildiğinden, ekstrasından 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 plakette üzerine aktarmadan önce denememize olanak sağlar.



2 Kanal Röle Kontrol Kartı

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



DHT11 Sıcaklık ve Nem Sensörü

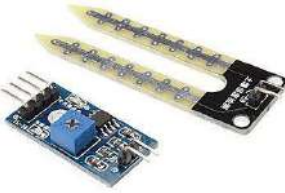
DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçer.

Set İçeriğini Tanıyalım



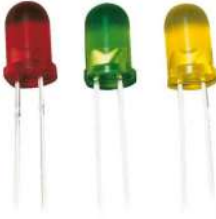
Mini Dalgıç Pompa

Çok sessiz çalışan düşük güç tüketimine sahip olan su geçirmez su motorumuz ile yağ e su gibi sıvıları saate 120 litreye kadar aktarabilirsiniz. IP68 standartta 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 problemleri 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.



RGB LED Diyet

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 kullanılabilirler.

NodeMCU Nedir?

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

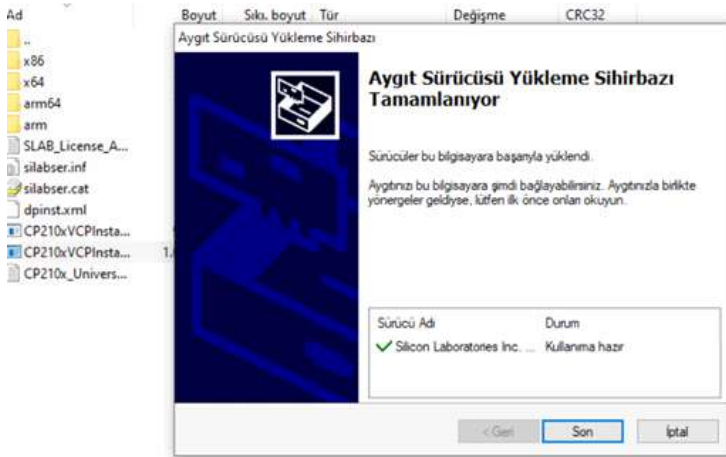
Ucuz olmasına rağmen çok stabil çalışan bir karttır. Kullanım alanı oldukça geniştir. Üzerinde bulunan ESP8266 Wifi modülü sayesinde internete kolay bir şekilde bağlanabiliyor, bu özelliği sayesinde uzaktan kontrol ve IOT projelerinde çok fazla kullanılır. Ayrıca düşük güç tükettiği için, güç tüketimi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

NodeMCU Driver Yüklemesi

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

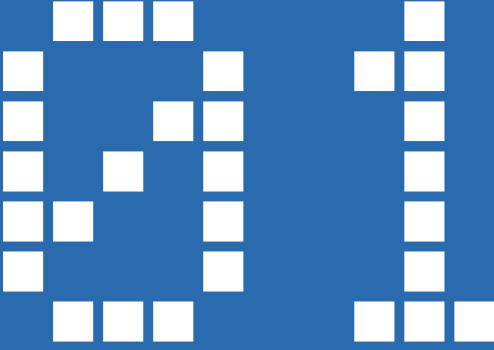
https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip

Cp2102 driverı bilgisayarınıza yükledikten sonra, dosyanın içinde bulunan exe uzantılı dosyayı çalıştırıyoruz. Daha sonra açılan setup penceresinde ileri butonuna tıklıyoruz. Son butonuna tıkladıktan sonra setup penceresi otomatik 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/>)



Tarayıcı Üzerinden LED Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)

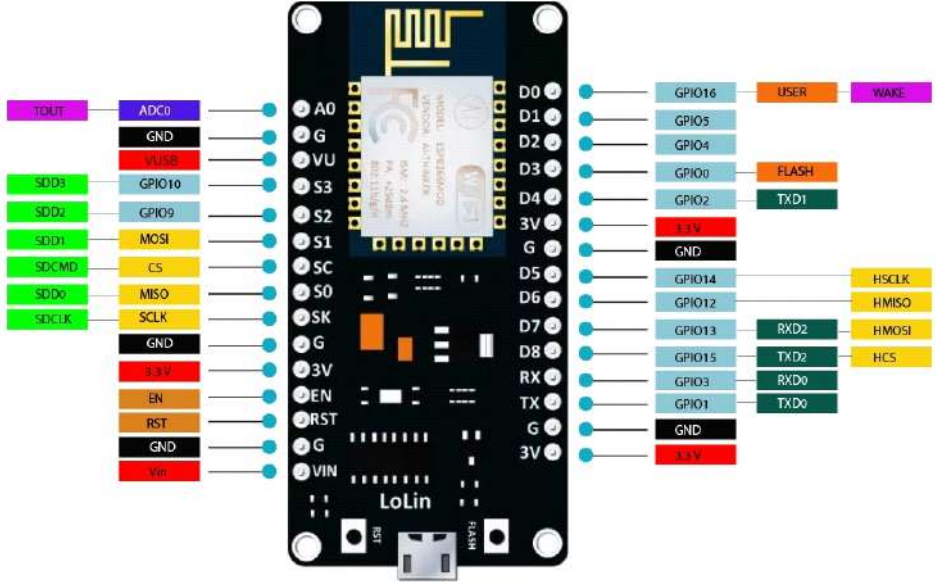


Maker'ların çok kullandığı cümleyle ilk projemize başlıyoruz. "Her proje led yakmakla başlar". İlk yapacağımız proje tarayıcı üzerinden led kontrol etmek olacak, IOT projelerine yeni başlayan birisi için oldukça ilgi çekici bir projedir.

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

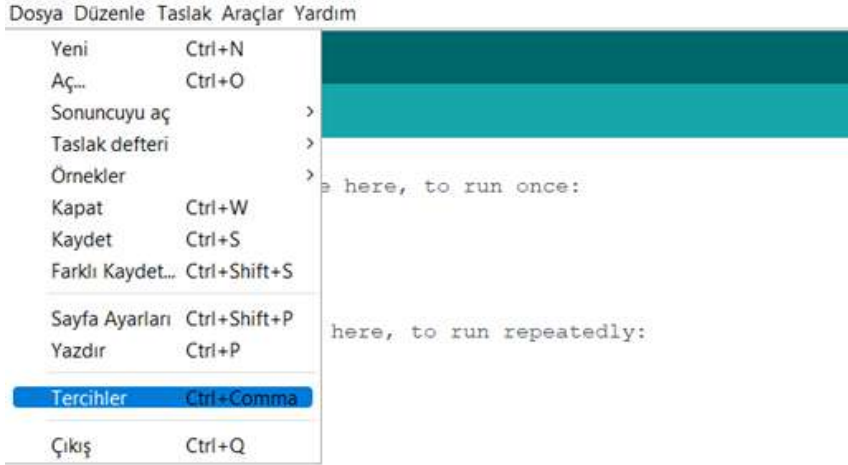
- NodeMCU Kartı
- 1 adet LED
- Breadboard
- Yeteri kadar jumper kablo (dişi-dişi)



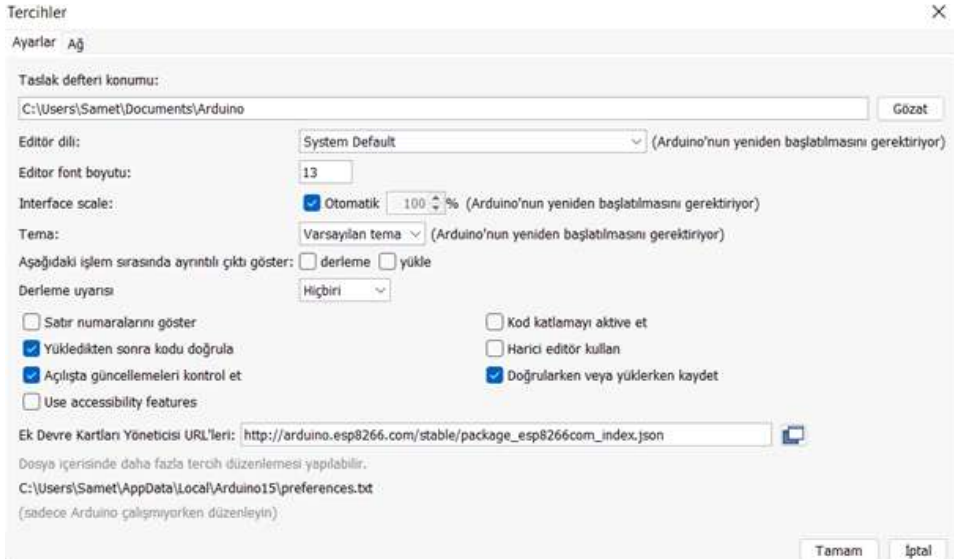
Yukarıdaki pinout(pin dağılımı) tablosu, yazılımı Arduino ide ile yazacağımız için NodeMCU unun pinlerini bu ara yüzde nasıl kullanacağımızı bilmemiz gerekiyor. Örneğimizde kullanacağımız gibi; mesela GPIO13 pini=D7 pinine karşılık geliyor. Yani biz yazılımda 13. pini kullandığımızda pinout tablosundan 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.

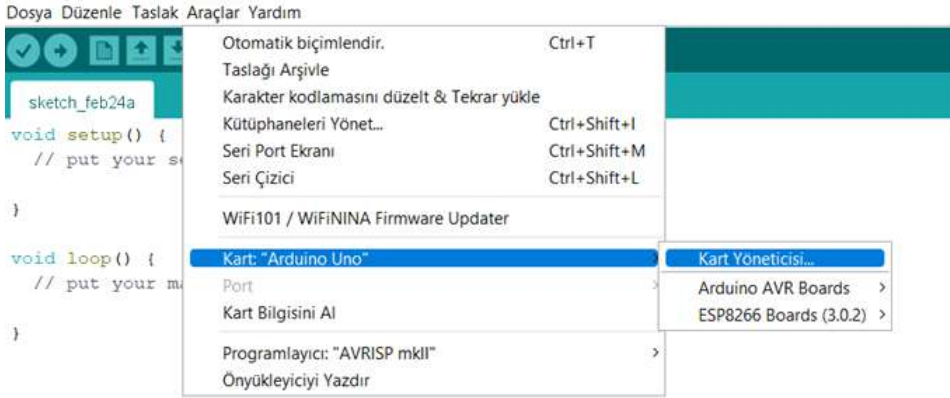
1) Yazılımı yazmadan önce arduino ide üzerinde birkaç ayar yapmamız gerekiyor. Arduino ara yüzünde varsayılan olarak NodeMCU modülü bulunmuyor. Modülü entegre etmek için, sol üst kısımda bulunan "**dosya menüsünden tercihler kısmına**"giriyoruz.



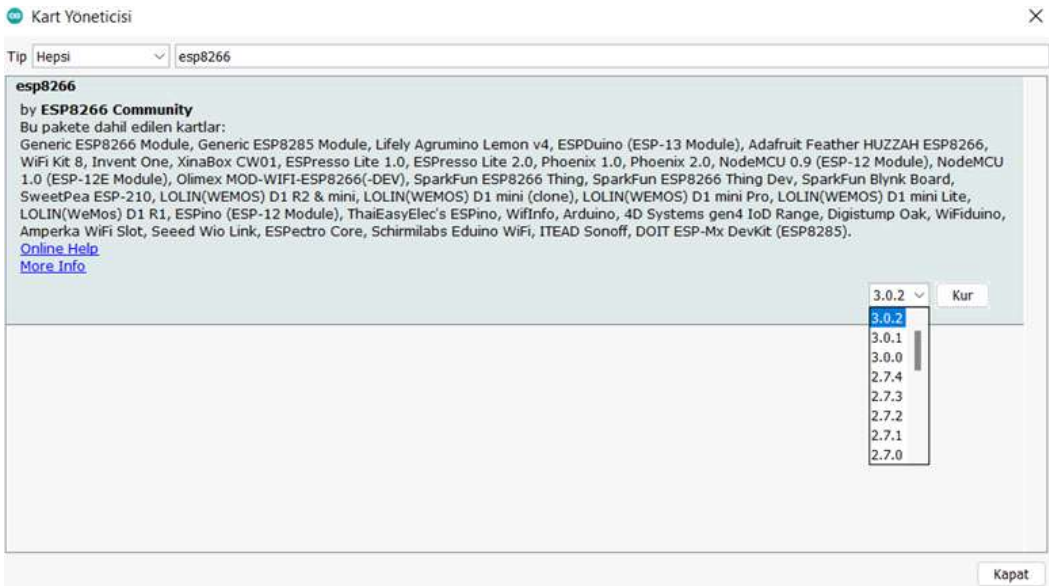
2) Açılan sekmede bulan ek devre kartları yöneticisi kısmına, bu linki el ile manuel şekilde girelim. "http://arduino.esp8266.com/stable/package_esp8266com_index.json" ve tamam diyelim.



3) Daha sonra araçlar menüsünden kart yöneticisini açıyoruz.



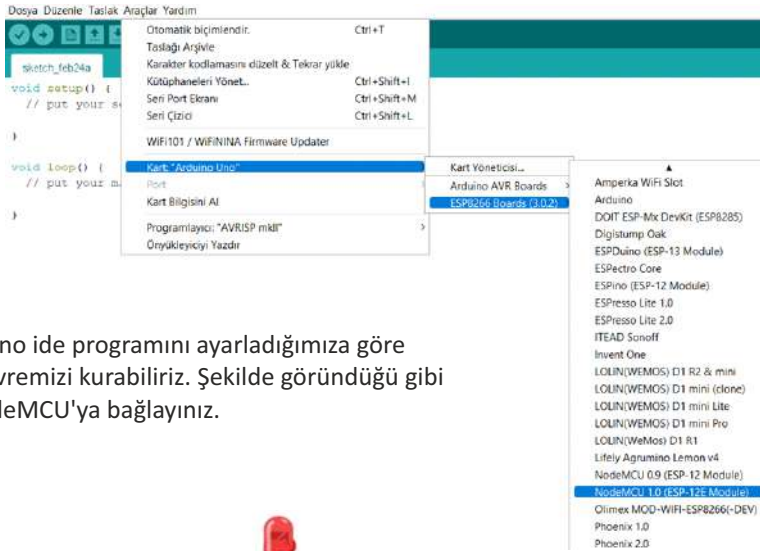
4) Açılan pencerede arama kısmına, esp8266 yazıp arama yapıyoruz. Daha sonra güncel versiyonunu seçip kuruyoruz.



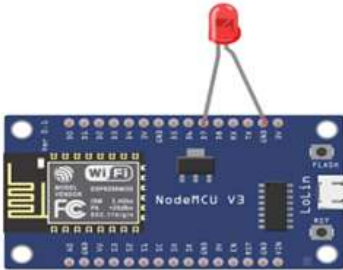
5) Ayrıca Blynk kütüphanesininde güncel versiyonunu yüklemeliyiz.



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



7) Arduino ide programını ayarladığımıza göre artık devremizi kurabiliriz. Şekilde görüldüğü 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.

8) Buraya kadar her şey tamamsa kodumuzu yüklemeye geçebiliriz.

```

#include <ESP8266WiFi.h>

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

int ledPin = 13; // GPIO13 ucu D7 ye karşılık 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 başlatılıyor
  server.begin();
  Serial.println("Server started");

  // bağlantı sağlandıktan sonra serial monitörde 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 bağlı olup olmadığını kontrol ediyoruz
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // client ın bir data göndermesini 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 yazdırıyoruz.
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 disconnected");
Serial.println("");

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.



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

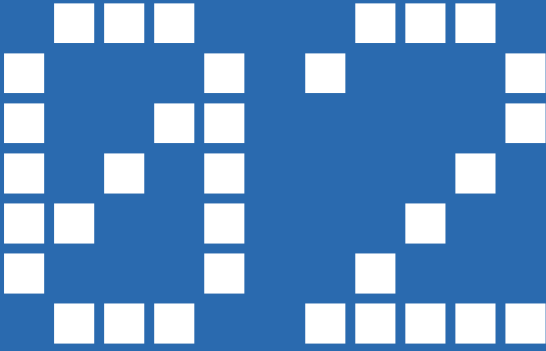
```
Connecting to blue
..
Wifi connected
Server started
Use this URL to connect: http://192.168.2.184/
```

A screenshot of a serial monitor application. The text area shows the following output: "Connecting to blue", "..", "Wifi connected", "Server started", and "Use this URL to connect: http://192.168.2.184/". At the bottom, there are controls: a checked "Autoscroll" checkbox, a "No line ending" dropdown menu, a "115200 baud" dropdown menu, and a "Clear output" button.

10) 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.



RGB LED Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



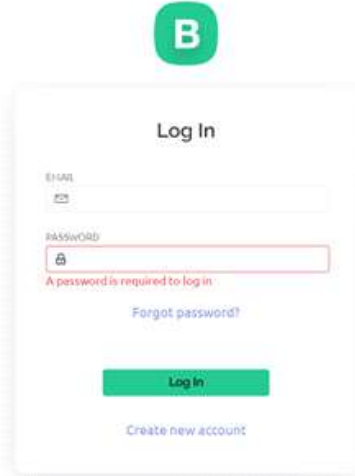
<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)



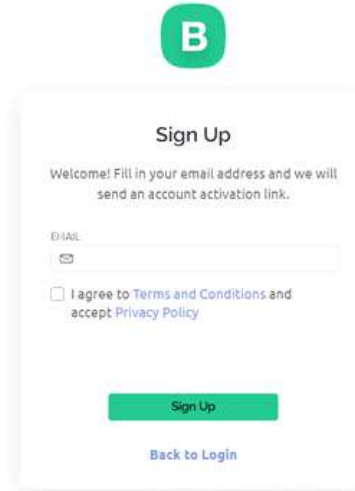
Bu projede Blynk uygulaması ile RGB diyot led kontrolünü yapacağız. Gece lambası ya da aksesuar gibi bir projeye çevirerek, evinizde güzel bir ortam elde edebilirsiniz. İlk olarak Blynk uygulamasının kurulumu ile başlayalım.

1)Öncelikle Web sitesinden Blynk uygulamasına girip üye olmamız gerekiyor.



The image shows a Blynk login form. At the top, there is a green circle with a white letter 'B'. Below it, the title 'Log In' is centered. The form contains two input fields: 'EMAIL' and 'PASSWORD'. The 'PASSWORD' field has a red border and a red error message below it: 'A password is required to log in'. Below the password field is a link 'Forgot password?'. At the bottom, there is a green 'Log In' button and a link 'Create new account'.

2) Email'imize girip gelen linke tıklamalıyız.



The image shows a Blynk sign up form. At the top, there is a green circle with a white letter 'B'. Below it, the title 'Sign Up' is centered. The form contains a welcome message: 'Welcome! Fill in your email address and we will send an account activation link.' Below this is an 'EMAIL' input field. Under the input field, there is a checkbox and the text 'I agree to Terms and Conditions and accept Privacy Policy'. At the bottom, there is a green 'Sign Up' button and a link 'Back to Login'.

3) Daha sonra gelen ekran da şifremizi oluşturmaliyiz.

Create Password

Create a password which is hard to guess.

PASSWORD

50-50

- Make it at least 8 symbols long
- Other tips:
 - Use uncommon words
 - Use non-standard uPPerCaSiNg
 - Use creatif spelllllllling
 - Use non-obvIoU\$ number\$ & symbot\$

[Log In](#) [Next](#)

4) Bizi hoşgeldin sayfası karşılıyor. Şimdilik geçelim.

Blynk Tour

1 Welcome 2 Platform 3 Folders 4 Devices 5 Template 6 Template components 7 Features 8 Business

Hi Blynk!

You've just joined the community of more than 500,000+ developers building amazing IoT products and projects.

With Blynk you can connect your devices to the internet and create mobile and web dashboards to control your devices from anywhere in the world.

Let's save your learning time with a few quick steps.



[Skip](#) [Let's go!](#)

5) Template bölümüne gelemim.

B

My organization - 303BPV

DEVELOPMENT TOOLS

Generate

Users

Locations

Billing

Test

ACCESS

Roles and permissions

DEVELOPERS

Workbooks

Billing

You are using Free plan

0/1000 1 of 5 0 of 5 [Get More](#)

Plan Comparison

Monthly [Yearly \(Save 20%\)](#)

Free	Plus	PRO	Business
50 months Billed yearly	14.99 months Billed yearly	547 months Billed yearly	1499 months Billed yearly
Devices 10 10	Devices 40 100 200		
web App	web App	web App	web App
One App	One App	One App	One App
Upgrade	Upgrade	Upgrade	Upgrade

Your current plan

6) Bu ekranda New Template seçip, yeni template oluşturacağız.



7) Oluşturduğumuz şablona bir isim verelim. Diğer ayarlar default gelecektir. Görünmüyorsa görseldeki gibi düzenleyebilirsiniz.

Create New Template

NAME

Use letters, digits and spaces only

HARDWARE

CONNECTION TYPE

DESCRIPTION

19 / 128

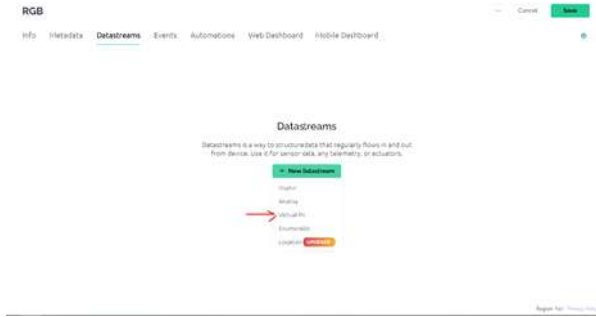
Cancel

Done

8) Açılan ekranda Datastream seçip değişkenlerimizi oluşturalım.



9) Sanal pinler oluşturacağımızdan görseldeki gibi “virtual pin” seçerek devam edelim.



10) 3 tane değişken oluşturmamız gerekli RED, GREEN ve BLUE. İlk olarak RED oluşturacağız, pin olarak istediğimizi seçebiliriz. Biz VO'ı seçiyoruz. Integer olarak kullanacağız, sadece basıp çekme işlemini yapacağımız için 0 ve 1 değerleri bizim işimize yarayacaktır.

Virtual Pin Datastream

NAME ● ALIAS

Use letters, digits, spaces, and underscores only

PIN DATA TYPE

UNITS

MIN MAX DEFAULT VALUE

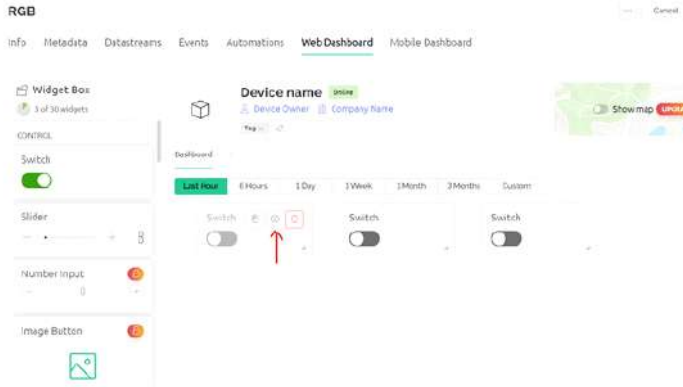
ADVANCED SETTINGS

Cancel Create

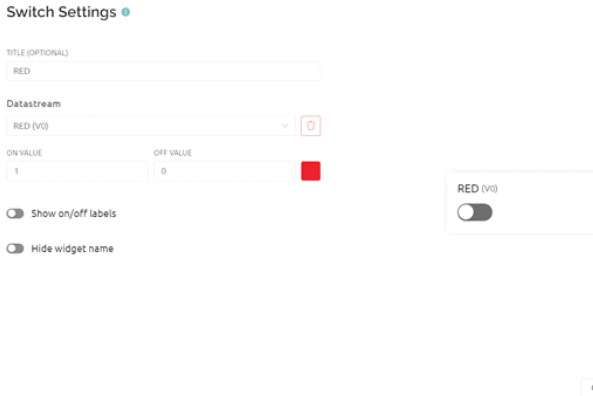
11) Bunu 2 kere daha tekrar edip 3 tane değişken elde ediyoruz.



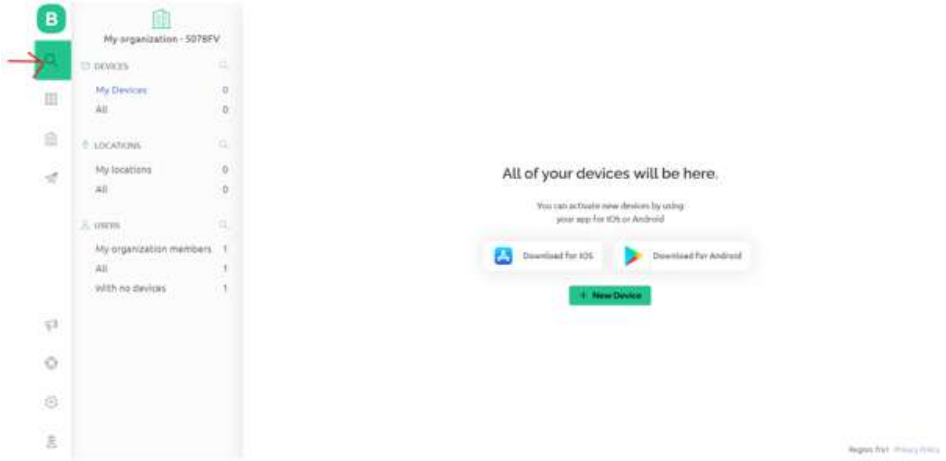
12) Daha sonrasında Web Dashboard ekranına gelip butonlarımızı eklemeliyiz. Web sitesinde buton olmayabilir, switch ekleyip buton olarak kullanabilirsiniz biz bu projede switch kullanacağız. Eklendikten sonra okla gösterdiğimiz yerden değişkenlerimizi switchlerimize atmalıyız.



13) İlk switchimize RED(V0) değişkenimizi atıyoruz ve save diyoruz. Bunu geriye kalan 2 değişkenimiz için de yapıyoruz.



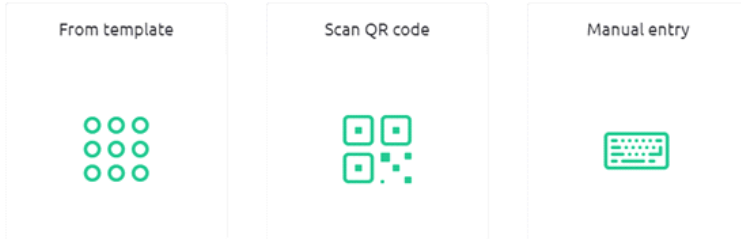
14) Artık yeni device dosyamızı oluşturmamız, okla gösterdiğim yere tıklayıp ekranın ortasındaki new device seçeneğine girmeliyiz.




15) Gelen ekranda new template seçeneğine tıklamalıyız ve template kısmında ismini verdiğimiz RGB seçeneği önümüze gelecek onu seçip ilerlemeliyiz.

New Device

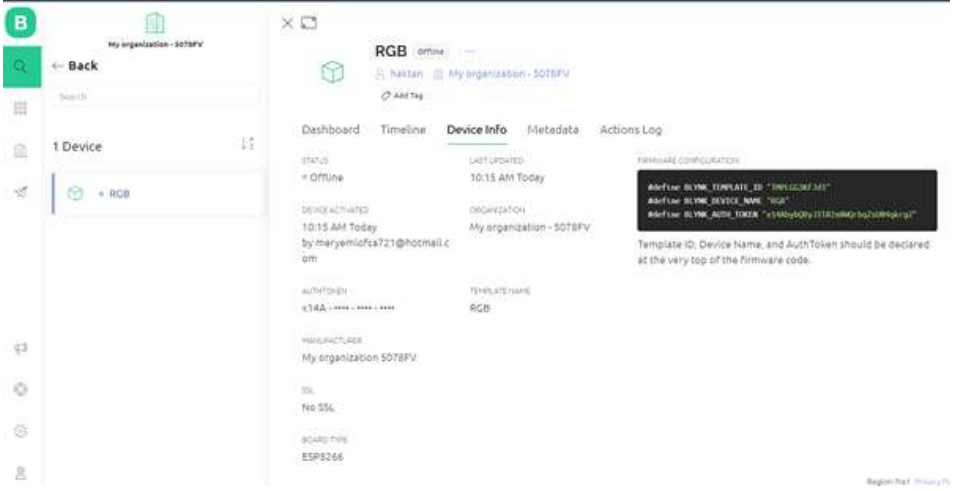
Choose a way to create new device



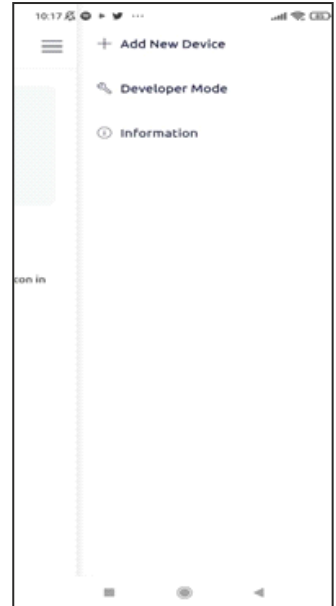
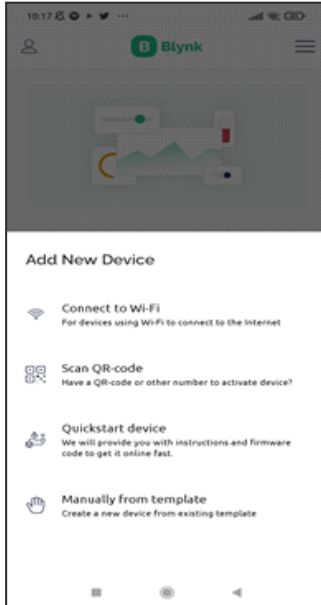
 Point on the cards to see instructions

Cancel

16) Artık TOKEN, ID ve NAME atamaları oluşturuldu bunu kodlarımızda kullanacağız.



17) Daha sonrasında BLYNK uygulamasını telefona indirip giriş yapmalıyız. Gelen ekranda sağ üst köşedeki seçeneğe girip ADD NEW DEVICE kısmını seçmeliyiz.

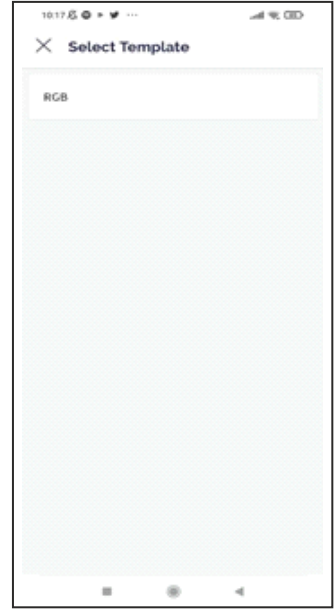


18) Gelen ekranda en aşağıdaki manually from template seçeneğini seçmeliyiz.

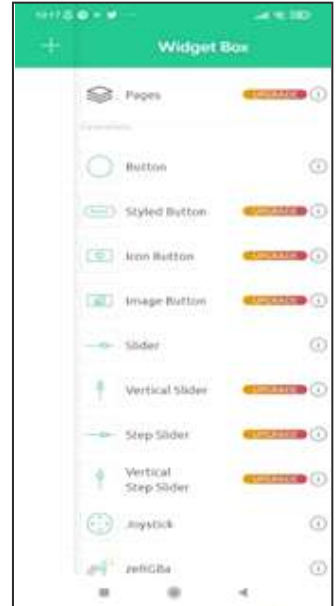
19) Burada oluşturduğumuz template'i görmemiz gerekiyor. Ya da ID girerek ulaşmak mümkün. Eğer site üzerinde oluşturduğumuz şablona halen erişim sağlayamadıysanız, adımları krarlamanızda fayda olacaktır.



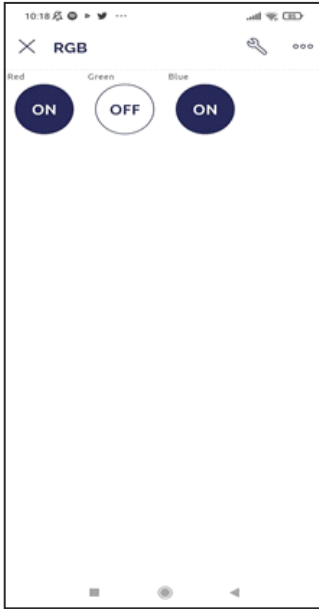
21) Şimdi artı işaretine tıklayıp button'larımızı eklemeliyiz.



20) Artık template'imizi eklemiş bulunmaktayız. Bu ekranda anahtar seçeneğine tıklayıp button'larımızı eklemeliyiz.



22) Eklediğimiz butona tıklayıp Web sayfasında oluşturduğumuz datastream'i eklemeliyiz, oluşturduğumuz datalar burada görünecektir. Switch kullanacağımız için seçimimizi yapalım ve bu adımları diğer 2 button için de gerçekleştirelim.

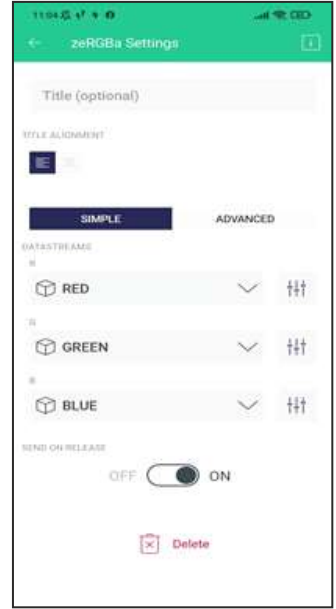


23) Artık butonlarımız da hazır bu aşamadan sonra kodlarımızı yüklemeliyiz.

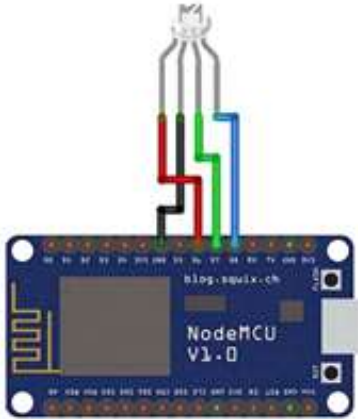


24) Dilerseniz görseldeki gibi zeRGB ekleyerek, uygulamayı daha eğlenceli bir hale getirelim.

25) Bu ekranda 3 datamızı da zeRGB'ya eklemeliyiz. Artık zeRGB ile de LED'imizi kontrol edebiliriz.



26) RGB LED'imizin bağlantılarını yapmalıyız. Kırmızı ayağını D6, yeşil ayağını D7 ve mavi ayağını D8'e bağlamalıyız. LED'iniz ortak anotluysa devrenin pozitif hattına , ortak katotlu ise eksi hattına bağlantı yapmalısınız.



RGB LED KONTROLÜ	
RGB KIRMIZI	D6
RGB YEŞİL	D7
RGB MAVİ	D8
RGB(-)	GND

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.

```

RGB §
#include <Blynk.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define RED D4
#define GREEN D7
#define BLUE D6

#define BLYNK_AUTH_TOKEN "YourAuthToken"

char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

BLYNK_WRITE(V0){
  int pinvalue = param.asInt();
  digitalWrite(RED,pinvalue);
}
BLYNK_WRITE(V1){
  int pinvalue = param.asInt();
  digitalWrite(GREEN,pinvalue);
}
BLYNK_WRITE(V2){
  int pinvalue = param.asInt();
  digitalWrite(BLUE,pinvalue);
}

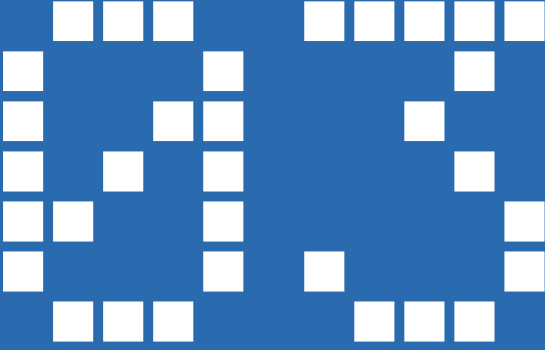
void setup() {
  Serial.begin(115200);
  delay(100);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.





DHT11 ile Sıcaklık ve Nem Ölçümü

BLYNK IoT
(IOS)



PROJELER LİNKİ



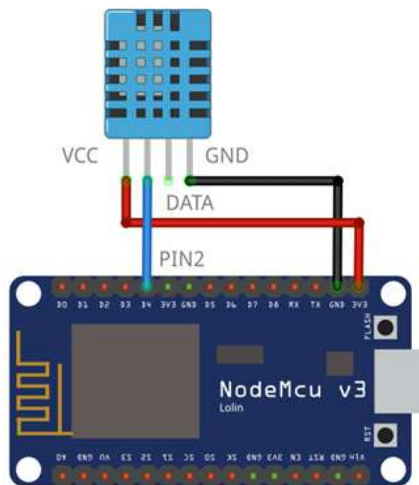
BLYNK IoT
(ANDROID)



<http://rbt.ist/nodemcu>

Bu proje tamamlandığında herhangi bir ortamın, sıcaklık ve nem değerlerini blynk uygulaması üzerinden takip edebilecekseniz.

1) Uygulamada kullanacağımız devre elemanlarını hazırlayarak, devre şemamızı oluşturalım. (NodeMCU WiFi geliştirme kartı , Dht11 sıcaklık nem sensörü , Breadboard , jumper kablo)



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

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

Virtual Pin Datastream

NAME

Sıcaklık

ALIAS

Sıcaklık

PIN

V5

DATA TYPE

Integer

UNITS

Celsius

MIN

0

MAX

50

DEFAULT VALUE

Default Value

☐ Thousands separator (e.g. 10,000)

☐ ADVANCED SETTINGS

Cancel

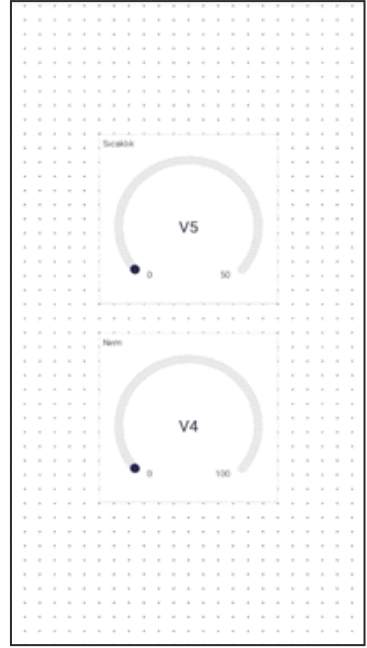
Create

3) Daha önceden Web sayfasından device oluşturup app ile eşleştirme işlemini yapmış olduğumuz için Blynk uygulamasına giriş yapıp manuel template kısmından template'imizi ekliyoruz. 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 ve V5 pinlerinden değer okuması yapacağız.

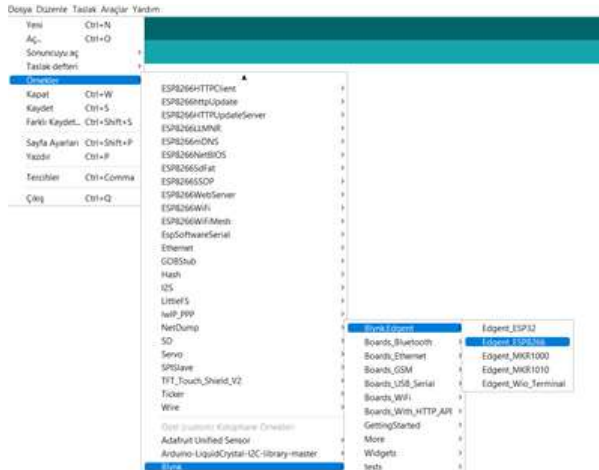
Virtual Pin Datastream üzerinden V5 pinindeki Sıcaklık için verilen değer aralığını 0-50 arasında ayarlarken V4 pinindeki Nem değerini 0-100 arasını gösterecek şekilde widget ayarlarını yapıyoruz.

NOT: Blynk uygulamasında daha gelişmiş widget'lar kullanılabilir fakat bu widget'ları 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. Fakat DHT 11 kütüphanesinin bilgisayarınızda kurulu olması gerekmektedir. Kurmak için linki kullanabilirsiniz. Ya da Arduino IDE kütüphaneler kısmından en güncel halini manuel olarak indirmesini gerçekleştirebilirsiniz.

(<https://github.com/adafruit/DHT-sensor-library>)

```

EdgeT_ESP8266 | BlynkEsp8266 | BlynkState | ConfigMode | ConfigStore | Console | Indicator | OTA | ResetButton | Settings
#define BLYNK_TEMPLATE_ID "YourTemplateID"
#define BLYNK_DEVICE_NAME "dht11"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

void sendSensor() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.println(t);
  Blynk.virtualWrite(V4, h);
  Blynk.virtualWrite(V5, t);
  Serial.print("Temperature : ");
  Serial.print(t);
  Serial.print(" Humidity : ");
  Serial.println(h);

  if (t > 30) {
    Blynk.email("your email", "Alert", "Temperature over 28C!");
    Blynk.logEvent("temp_alert", "Temp above 30 degree");
  }
}

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(2500L, sendSensor);
}

void loop() {
  Blynk.run();
  timer.run();
}

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.



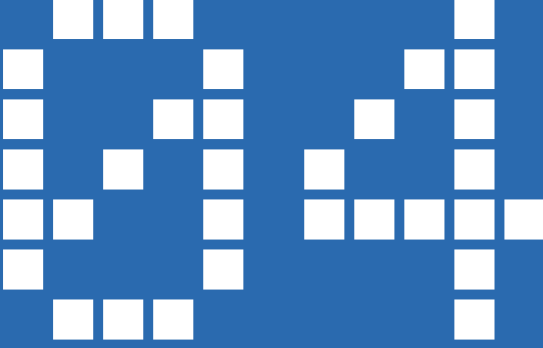


Uygulamayı tamamladıktan sonra projeleştirmeyi düşünüyor ve 3D ile de ilgileniyorsanız aşağıdaki örneği inceleyebilirsiniz.

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.





Uzaktan Röle Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)

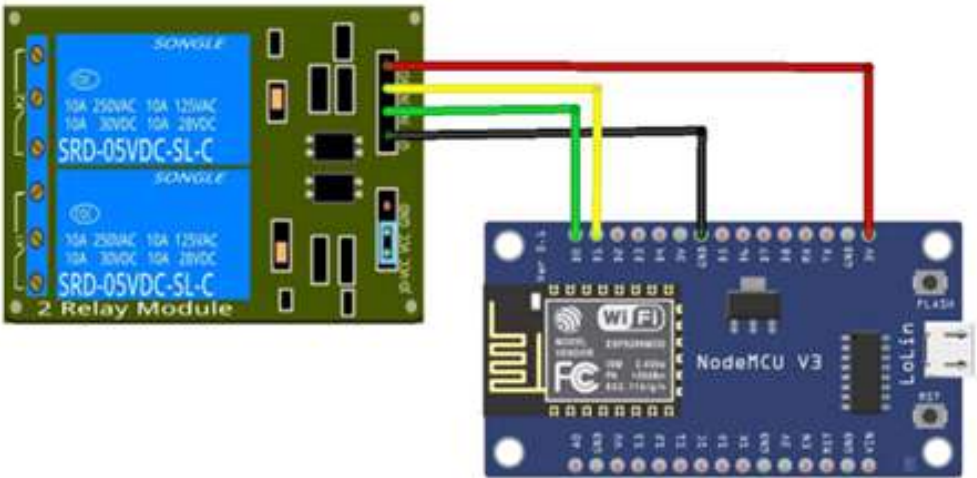


Bu projede “Blynk” uygulaması ile röle açıp kapatma uygulamamızı gerçekleştireceğiz. Bu rölenin uçlarına ne bağlarsanız (klima, lamba, motor) evinizden, iş yerinizden kontrol edebileceksiniz. Bu projeyi yaparken röleye bağlayacağınız yüksek voltaj ile çalışan cihazların bağlantılarını yaparken dikkatli 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 <u>Pin 1</u>	D0
Röle <u>Pin 2</u>	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.

Virtual Pin Datastream

NAME: role ALIAS: role

PIN: V0 DATA TYPE: Integer

UNITS: None

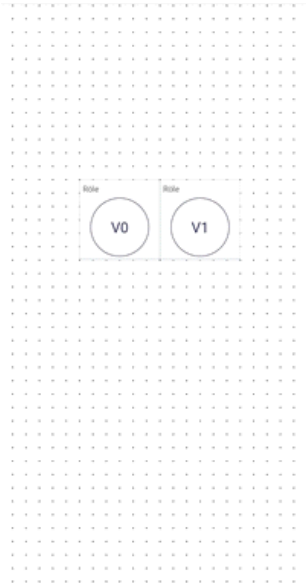
MIN: 0 MAX: 1 DEFAULT VALUE: Default Value

☐ Thousands separator (e.g. 10,000)

☐ ADVANCED SETTINGS

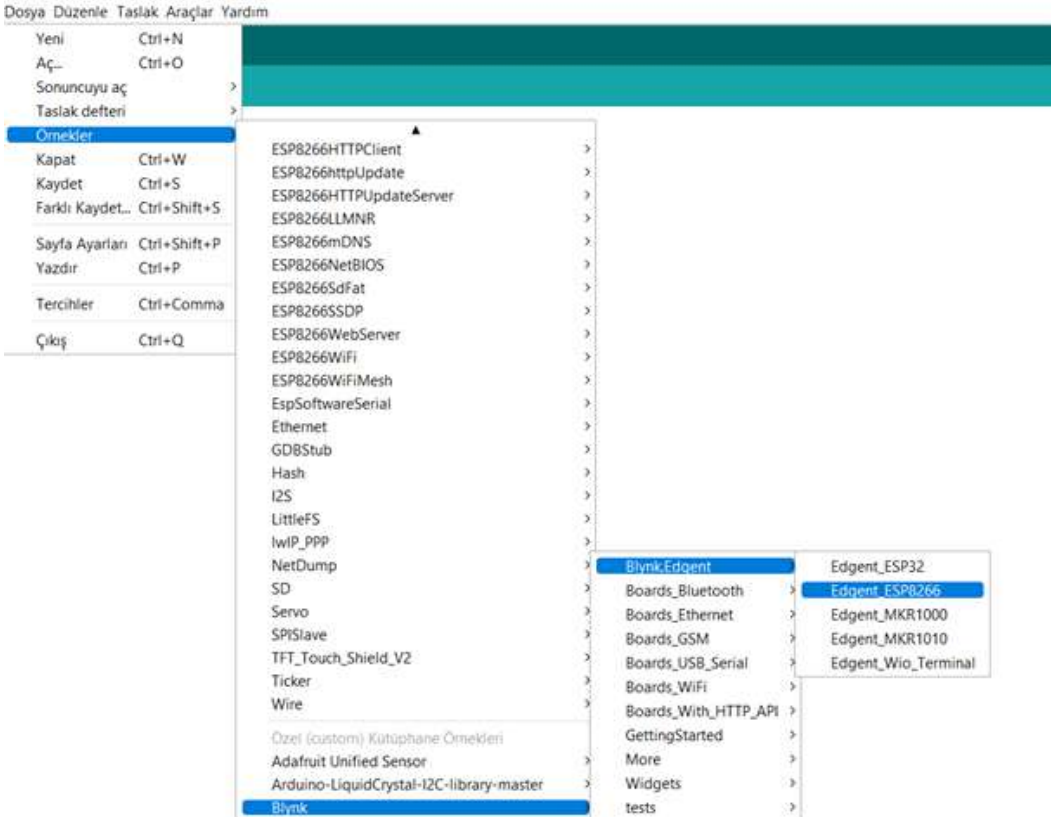
Cancel Create

3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemi 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.

```

relay_kontrol $
#include <Blynk.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define relay1 D0
#define relay2 D1
#define BLYNK_AUTH_TOKEN "YourAuthToken"

char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

BLYNK_WRITE(V0) {
  bool value1 = param.asInt();
  |
  if (value1 == 1) {
    digitalWrite(relay1, LOW);
  } else {
    digitalWrite(relay1, HIGH);
  }
}

BLYNK_WRITE(V1) {
  bool value2 = param.asInt();

  if (value2 == 1) {
    digitalWrite(relay2, LOW);
  } else {
    digitalWrite(relay2, HIGH);
  }
}

void setup() {

  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);

  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}

void loop() {

  Blynk.run();
}

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.



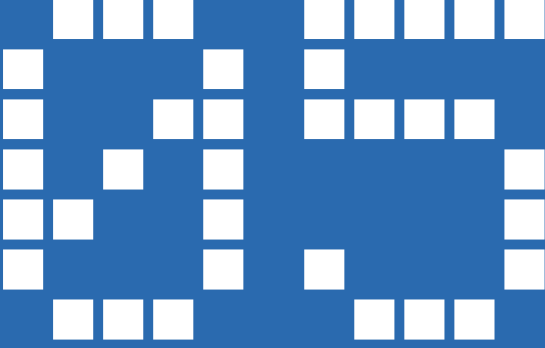


Uygulamayı tamamladıktan sonra projeleştirmeyi düşünüyor ve 3D ile de ilgileniyorsanız aşağıdaki örneği inceleyebilirsiniz.

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 çizim dosyalarına ulaşabilirsiniz.





Toprak-Nem Sensörü Uygulaması

BLYNK IoT
(IOS)



PROJELER LİNKİ



<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)

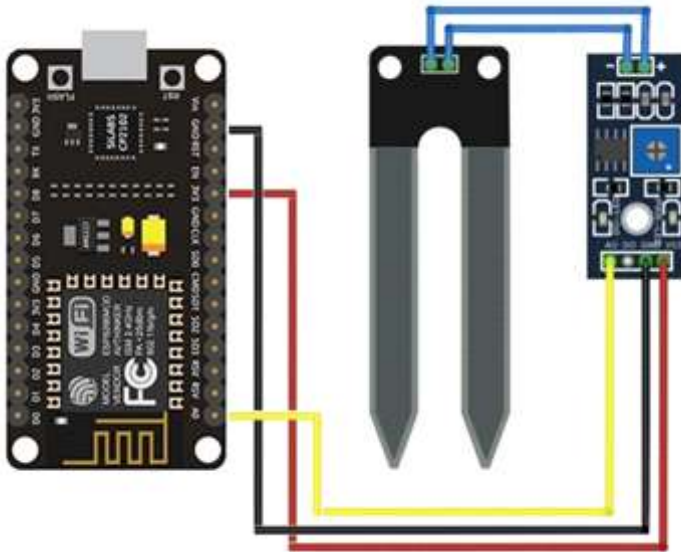


Bu projede Wifi üzerinden Blynk uygulaması ile toprak-nem sensörünün okuduğu değeri görebileceğiz. Bu gördüğümüz değere bağlı olarak röle ile birlikte çıktı alabileceğiz.

1) İlk olarak devre kurulumu ile başlayalım.

Gerekli Malzemeler :

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



Toprak nem sensörünün A0 pinini NodeMCU'nun A0 pinine,GND pinini GND pini ile VCC pinini de 3.3V kısmına bağlıyoruz.Sensörün eksi ve artı hatlarının bağlantısını da yaptıktan sonra devre bağlantımız bitmiş oluyor.

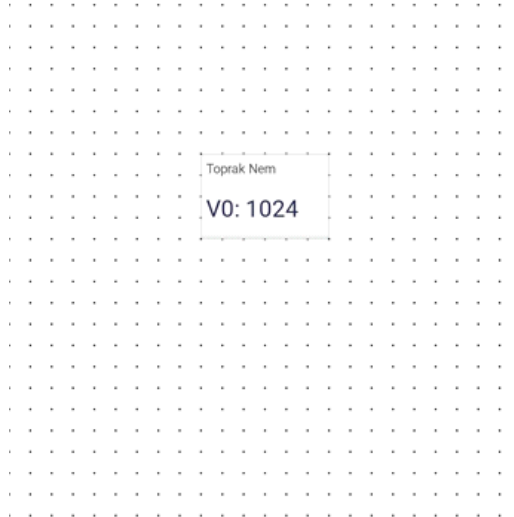
TOPRAK-NEM SENSÖRÜ UYGULAMASI	
SENSÖR PİN	A0
SENSÖR VCC	VİN
(-)	GND

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

Virtual Pin Datastream

NAME	ALIAS	
<input type="text" value="Toprak Nem"/>	<input type="text" value="Toprak Nem"/>	
PIN	DATA TYPE	
<input type="text" value="V0"/>	<input type="text" value="Integer"/>	
UNITS		
<input type="text" value="None"/>		
MIN	MAX	DEFAULT VALUE
<input type="text" value="0"/>	<input type="text" value="10000"/>	<input type="text" value="Default Value"/>
<input checked="" type="checkbox"/> Thousands separator (e.g. 10,000)		
<input type="button" value="+ ADVANCED SETTINGS"/>		
<input type="button" value="Cancel"/>		<input type="button" value="Save"/>

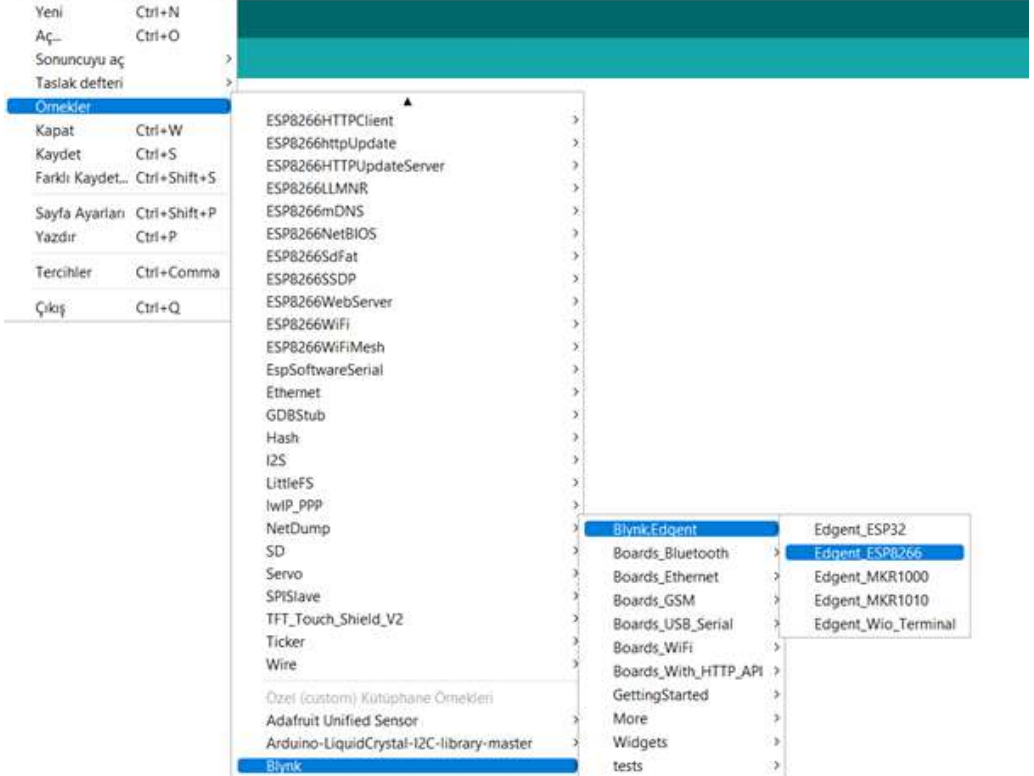
3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemini yapmış olduğumuz Blynk uygulamasına giriş yapıp toprak nem sensörünün değerini görmek için labeled value ekliyoruz.



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.)

Dosya Düzenle Taslak Araçlar Yardım



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

```

SULAMA | BlynkEsp8266.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Console.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "YourTemplateID"
#define BLYNK_DEVICE_NAME "YourDeviceName"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;
int nemPin = A0;

int nemdegeri;
int nemEsikdegeri = 270;

char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  pinMode(nemPin, INPUT);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(1000L, sendData);
  Serial.begin(115200);
}

void sendData()
{
  nemdegeri = analogRead(nemPin);

  Serial.print("Direnç Değeri: ");
  Serial.println(nemdegeri);

  if(nemdegeri < nemEsikdegeri)
  {
    Serial.println("Ortam ISLAK SULAMA GEREKMİYOR");
  }
  else
  {
    Serial.println("Ortam KURU SULAMA YAPILMASI GEREK");
  }
  delay(1000);
  Blynk.virtualWrite(V0, nemdegeri);
}

void loop()
{
  Serial.println(nemdegeri);
  delay(200);
}

Blynk.run();
timer.run();
}

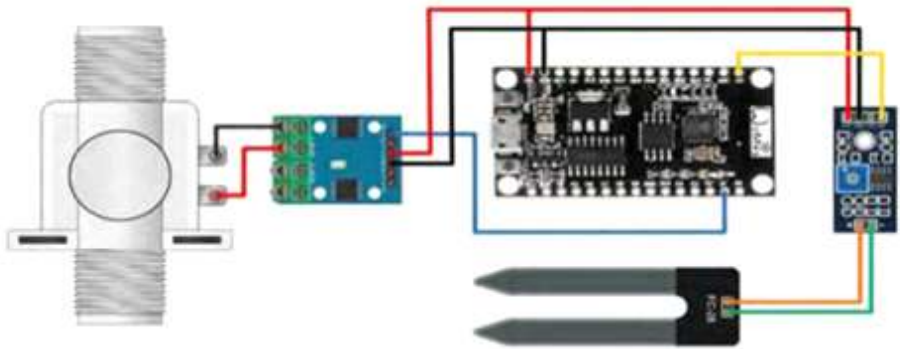
```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.

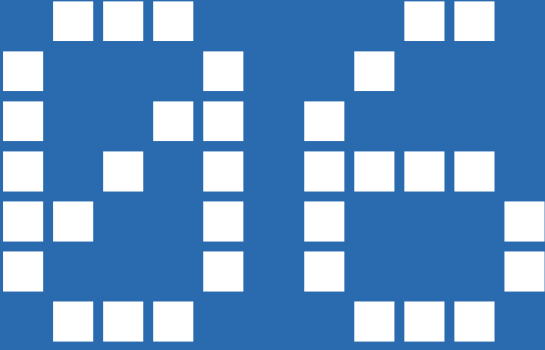


NOT: Toprak nem sensörü ve röle ile birçok proje geliştirilebilir. Rölenin çıkış pinlerine motor sürücü, led, buzzer vb. gibi elektronik elemanlar bağlanıp yeni projeler elde edilebilir.

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	Vin
(-)	GND



Servo Motor Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)



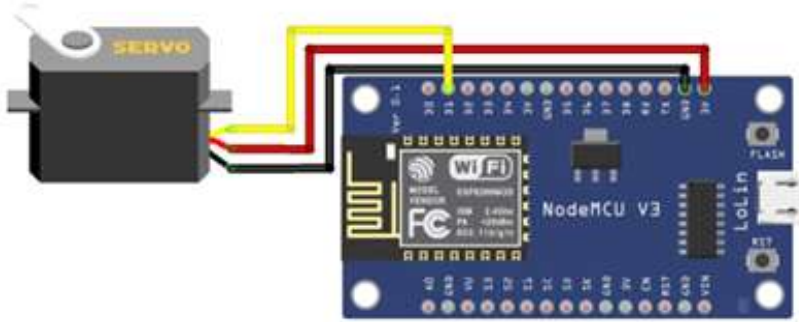
Bu projede wifi üzerinden Blynk uygulamasıyla servo motor kontrolü projesi yapacağız. Servo motorun pervanesini taktığımızda kaç derece döndürebildiğimizi rahat bir şekilde görebilecek duruma geleceğiz.

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 kısma 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	
<u>Servo Pin</u>	D1
VCC	Vin
(-)	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.

Virtual Pin Datastream

NAME SERVO	ALIAS SERVO
PIN V0	DATA TYPE Integer
UNITS None	
MIN 0	MAX 180
DEFAULT VALUE Default Value	
<input checked="" type="checkbox"/> Thousands separator (e.g. 10,000)	
<input type="checkbox"/> ADVANCED SETTINGS	
Cancel	Save

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.

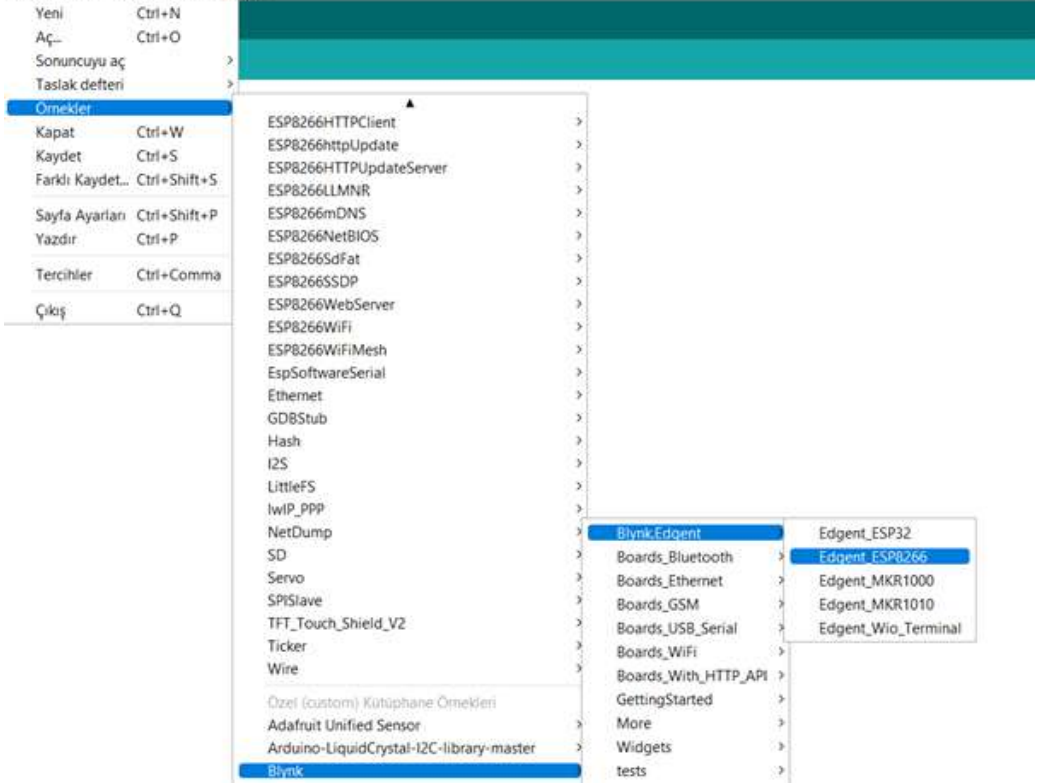
3) Daha önceden telefonumuza indirip ve kart ile eşleştirme işlemi yapmış olduğumuz Blynk uygulamasına giriş yapıp servo motoru telefonumuzdan döndürebilmek için Slider widgetını ekliyoruz. Daha önceki projelerde bahsettiğimiz gibi Blynk uygulamasını pro moda yükselterek daha farklı widgetlara erişebilirsiniz.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

Dosya Düzenle Taslak Araçlar Yardım



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

```

servo $ BlynkEdgent.h BlynkState.h ConfigMode.h ConfigStore.h Console.h Indicator.h OTA.h ResetButton.h Settings.h

#include <Servo.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "YourTemplateID"
#define BLYNK_DEVICE_NAME "YourDeviceName"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
#define USE_NODE_MCU_BOARD
// #include "BlynkEdgent.h"
Servo servo;
char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

BLYNK_WRITE(V0) {
  int s0 = param.asInt();
  servo.write(s0);
  Blynk.virtualWrite(V0, s0);
}

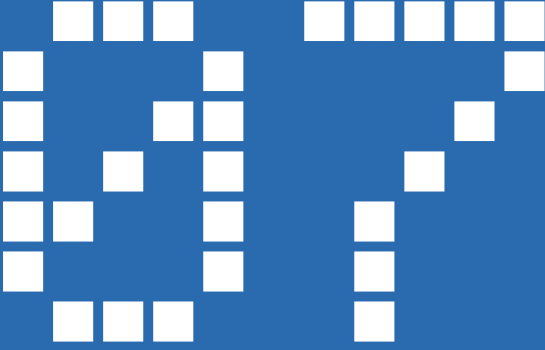
void setup() {
  Serial.begin(9600);
  servo.attach(D1);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}

void loop() {
  Blynk.run();
}

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.





Duman - Gaz Dedektörü

BLYNK IoT
(IOS)



PROJELER LİNKİ



<http://rbt.ist/nodemcu>

BLYNK IoT
(ANDROID)

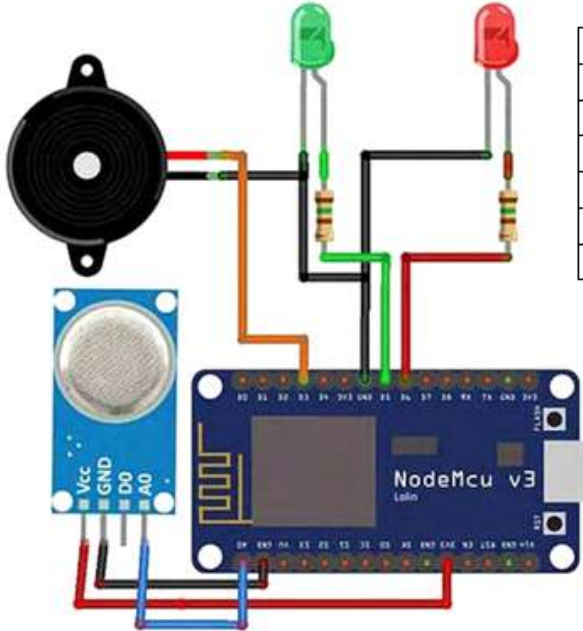


Bu projede telefonunuzdan Blynk programını kullanarak wifi ağı üzerinden NodeMCU kartı sayesinde Gaz sensöründen gelen veriler ile ortamın gaz ve duman seviyesini ölçebileceksiniz, belli seviye üzerinde telefonumuza bildirim gelecek ve buzzer çalışacaktır. Böylelikle ortamdaki havanın ne kadar temiz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

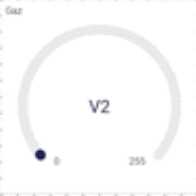
Devre kurulumu tamamladıktan sonra Blynk uygulamasındaki ayarlamalara geçeceğiz. Daha önceden Blynk uygulamasının ayarlarını yapmıştık. Eğer herhangi bir sorun ile karşılaşıyorsanız ilk projemizde nasıl kurulum yapılacağından bahsetmiştik.

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

Virtual Pin Datastream

NAME	ALIAS	
<input type="text" value="gaz"/>	<input type="text" value="gaz"/>	
PIN ⓘ	DATA TYPE	
<input type="text" value="V2"/>	<input type="text" value="Integer"/>	
UNITS		
<input type="text" value="None"/>		
MIN	MAX	DEFAULT VALUE
<input type="text" value="0"/>	<input type="text" value="255"/>	<input type="text" value="40"/>
<input type="checkbox"/> Thousands separator (e.g. 10,000)		
<input type="checkbox"/> ADVANCED SETTINGS		
<input type="button" value="Cancel"/> <input type="button" value="Save"/>		

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 bildirim 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.

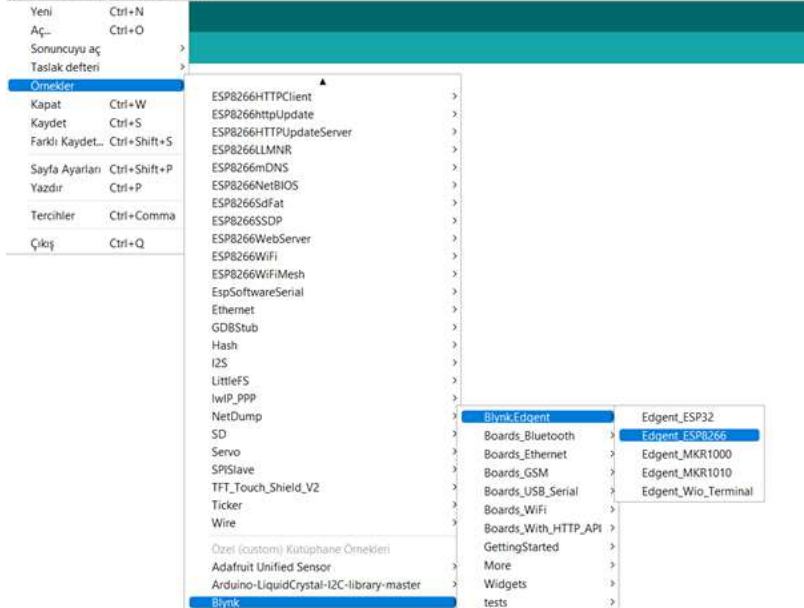
5) Devre kurulumumuz ve Blynk uygulaması üzerinde ayarlamalar bittiğine göre, NodeMCU kartına kodu yükleyebiliriz. Yazılımı yazmadan önce gerekli kütüphaneler için dosyamızı açalım. Dosya kısmından örneklerle gelelim. Özel kütüphanelerden BlynkEdgent içinden ESP8266'yı seçelim. Ardından kodumuza geçelim.

(Kütüphaneyi

<https://github.com/blynkkk/blynk-library>

bu link üzerinden indirip Arduino IDE'mize eklememiz gerekmektedir.)

Dosya Düzenle Taslak Araçlar Yardım



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

```
GAZ$ BlynEsp.h BlynSerial.h ConfigMode.h ConfigStore.h Console.h InetClient.h OTA.h ReadBuffer.h Settings.h

#define BLYNK_TEMPLATE_ID "YourTemplateID"
#define BLYNK_DEVICE_NAME "YourDeviceName"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
#define GreenLed 14
#define RedLed 12

#define MQ2pin (A0)
#define BLYNK_PRINT Serial
#define buzzerPin 0
#define GreenLed 14
#define RedLed 12

#include <SPI.h>
#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;
char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

float sensorValue;

int n,m;

void setup()
{
    Serial.begin(115200);
    Serial.println("Gas sensor warming up!");
    delay(5000);

    Blynk.begin(auth, ssid, pass);

    pinMode(GreenLed, OUTPUT);
    pinMode(RedLed, OUTPUT);
    pinMode(12, OUTPUT);

    pinMode(16, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    digitalWrite(GreenLed, LOW);
    digitalWrite(RedLed, LOW);
    timer.setInterval(1000L, sendUptime);
}

void sendUptime()
{
    Blynk.virtualWrite(V2, sensorValue); //V1 is for Gas
}

void loop()
{
    sensorValue = analogRead(MQ2pin); // read analog input pin 0

    Serial.println("");
}
```

```

Serial.print("Sensor Value: ");
Serial.print(sensorValue);
Serial.println("");
Serial.println("");
Serial.print("smoke: ");
Serial.print(A0);
Serial.println("");
Blynk.virtualWrite(V0, sensorValue);
if(sensorValue > 300 && sensorValue < 600)
{
    Serial.print("Smoke detected!");
}

n=analogRead(A0);

if (n>600)
{
    n==m;
    m=analogRead(A0);
    Serial.println(m);
}
if (n>300 && n<600)
{
    digitalWrite(D3,LOW);
    digitalWrite(D6,HIGH);
    digitalWrite(D5,LOW);
    tone(buzzerPin,800,80);
    delay(2000);
}
if (n<300)
{
    digitalWrite(D5,HIGH);
    digitalWrite(D3,LOW);
    digitalWrite(D6,LOW);
    delay(2000);
    noTone(buzzerPin);
}

Blynk.run();
}

```

Projenin kodlarına yandaki
QR kodu okutarak ulaşabilirsiniz.





FORUM



forum.robotistan.com

BLOG



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