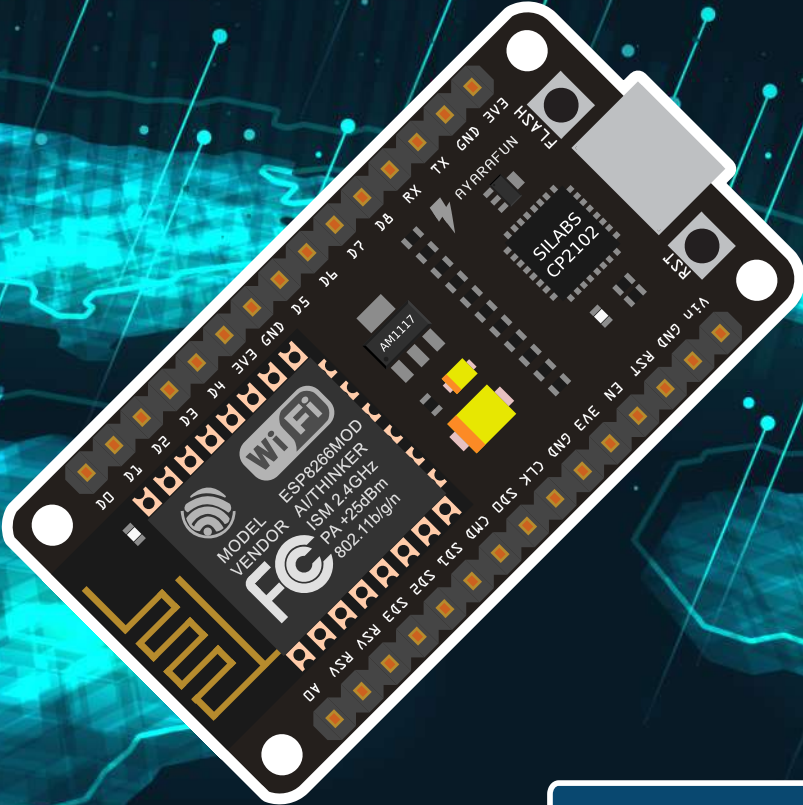




NodeMCU IoT Cloud Uygulamaları



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 |
| Arduino IoT Cloud'a Giriş..... | 10 |
| Arduino IoT Cloud Kullanımı..... | 12 |
| IoT Cloud ve Amazon Alexa ile LED Kontrolü..... | 16 |
| DHT11 ile Sıcaklık ve Nem Ölçümü..... | 25 |
| RGB LED Kontrolü..... | 30 |
| Bitki Sulama Sistemi..... | 34 |
| Servo Motor Kontrolü..... | 39 |
| Duman-Gaz Dedektörü..... | 44 |

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 mikrodenetleyiciler ile kullanılabilen bir röle kartıdır. Mikrodenetleyiciden tetik sinyali sırasında 20mA'lık bir akım çekmektedir. Çeşitli hobi, endüstriyel ve robotik wprojelere sıklıkla kullanılır. 30VDC veya 220VAC gerilimde 10A'e kadar akımı anahtarlayabilmektedir. Her bir röle için kontrol ledleri bulunmaktadır.



DHT11 Sıcaklık ve Nem Sensörü

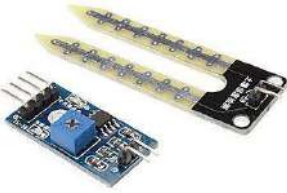
DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçer.

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 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.



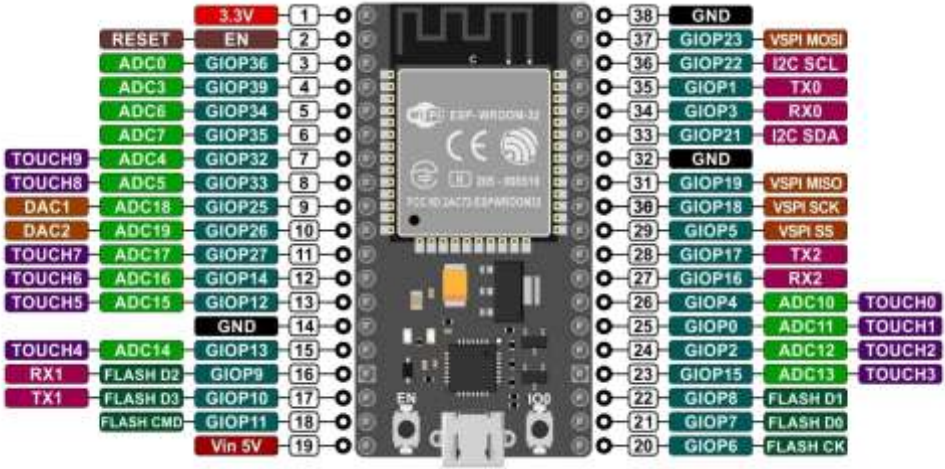
Buzzer

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 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.

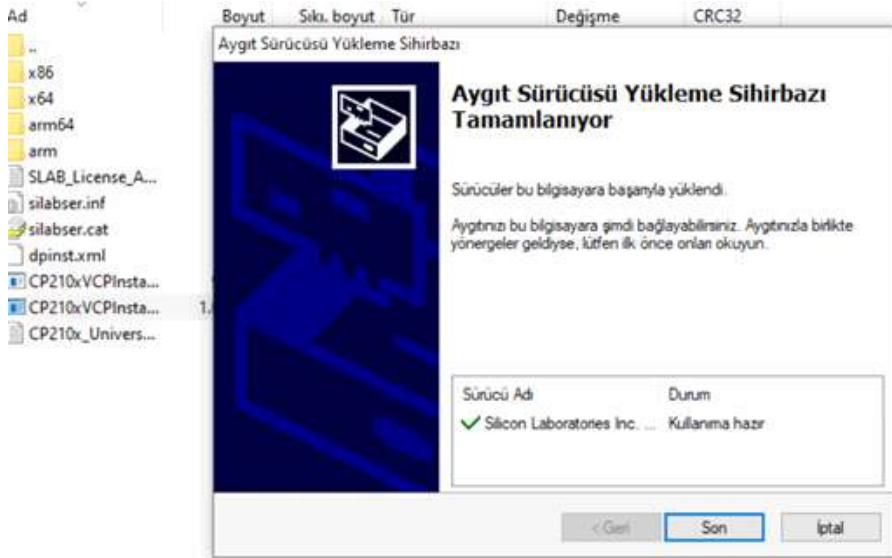


NodeMCU Driver Yüklemesi

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

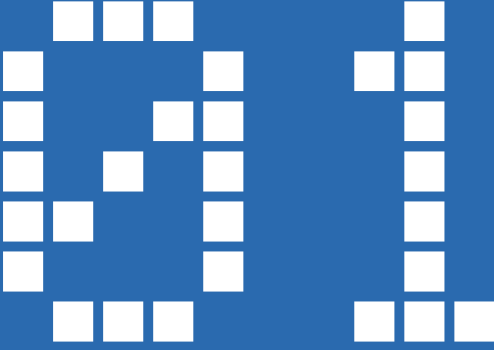
https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip

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/>)



Arduino IoT Cloud 'a Giriş

BLYNK IoT
(IOS)



PROJELER LİNKİ



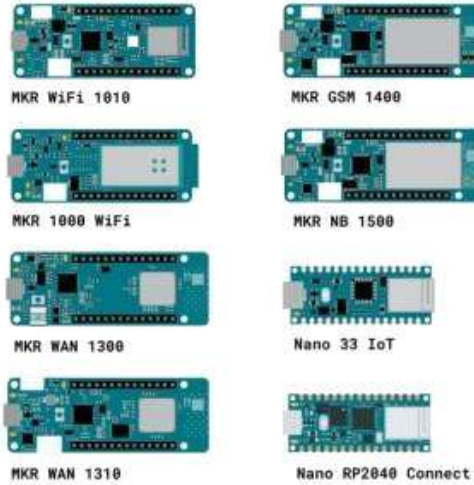
BLYNK IoT
(ANDROID)



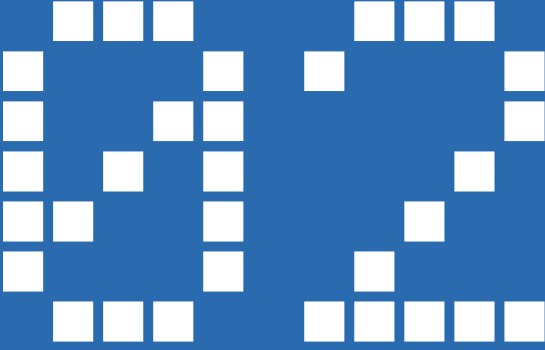
github.com/Robotistan/IoTCloud

Arduino IoT Cloud, Arduino tarafından geliştirilen bir bulut sistemidir. Özel tasarlanmış Arduino kartları, ESP8266, ESP32 ve LoRawan ile IoT projelerini bu bulut platform üzerinden gerçekleştirme imkanına sahip olabilirsiniz. Web arayüzü ve mobil uygulama gibi özellikleri, OTA protokolüne sahip olması ve en önemlisi ücretsiz olması benzer platformlar arasında kesinlikle bir tercih sebebi oluşturmaktadır. Ancak bu kısımda ufak bir sorun bulunmaktadır. Arduino Cloud'un ücretsiz versiyonunda sadece veri görselleştirme yapabiliyoruz. Çünkü Arduino Cloud'un ücretsiz erişilebilir olan paketi bazı özellikleri kısıtlıyor.

Uyumlu Kartlar:



1. MKR 1000 WiFi
2. MKR WiFi 1010
3. MKR WAN
4. Nano RP2040 Connect
5. Nano 33 IoT
6. ESP32
7. ESP8266
8. LoRaWAN



Arduino IoT Cloud Kullanımı

BLYNK IoT
(IOS)



PROJELER LİNKİ

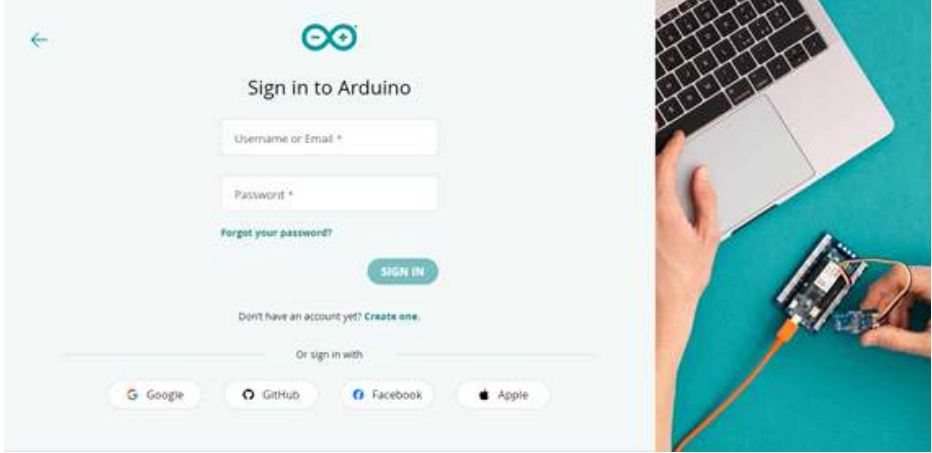


BLYNK IoT
(ANDROID)

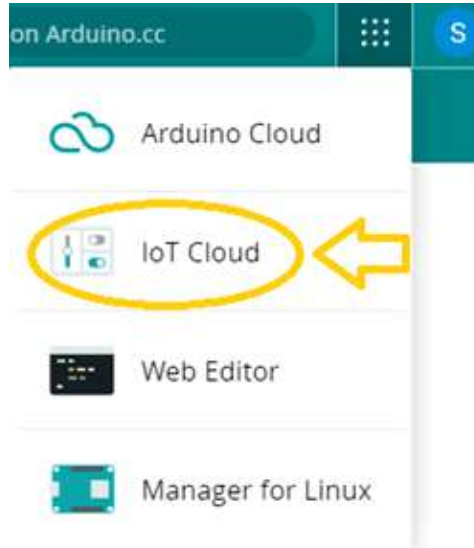


github.com/Robotistan/IoTCloud

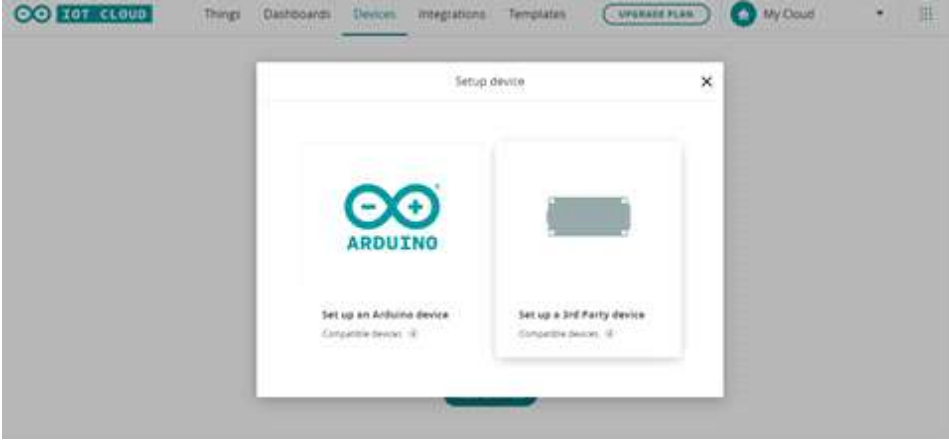
1. İlk olarak Arduino kendi sitesine giderek bir hesap oluşturmamız gerekiyor. Kayıt ekranında farklı şekillerde kayıt olma seçeneği mevcuttur. Size uygun olanı seçerek gerekli adımlardan sonra kaydı tamamlayabilirsiniz. (<https://www.arduino.cc/>)



2. Kayıt işlemi tamamlandıktan sonra sağ üst köşedeki dört nokta menüsüne giderek Arduino IoT Cloud'a tıklayalım. Ya da sayfa içeriğinde bulunan CLOUD kısmına tıklayarak <https://cloud.arduino.cc/home/> sayfasına gidilebilir.



3. Ardından açılan sekmeden Device kısmından kullanacağımız cihazı eklemeliyiz.



4. Cihaz ekleme işleminde 3rd Party Device seçeneğini seçmeliyiz. Seçimden sonra ESP8266 seçip, kullandığımız NodeMCU cihazı seçmeliyiz. Bu aşamadan sonra size oluşturduğunuz cihazın ID'sini ve Secret Key'ini verecektir. Bu bilgileri kayıt etmelisiniz. Özellikle Secret Key'i daha sonradan bulmanız imkansız. Onu kayıt edip projemizde gerekli yerlerde kullanmalıyız. Bu anahtarı kayıp ederseniz cihazı silip tekrardan oluşturmalısınız.



Setup device
X

Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF](#).

Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID

ccb6e569-591b-4915-82bf-53f042cd407f

Secret Key

LXCWYICQ8FILPTQ3GYBF

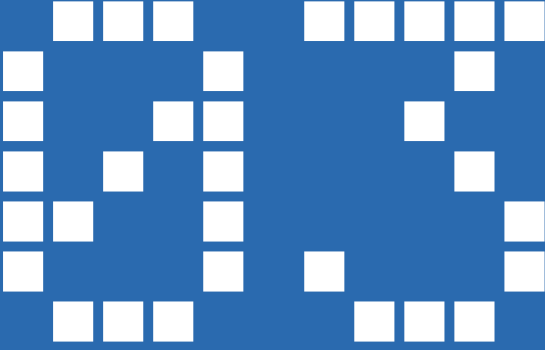
Secret key cannot be recovered

Please keep it safe, if you lose it you will have to delete and setup your device again.

☒ I saved my device ID and Secret Key
 CONTINUE

5. Kartınıza Web üzerinden kodları atabilmek için bilgisayarınızda Arduino Agent'in kurulu olması gerekiyor. Linkten indirerek kurulumunu gerçekleştirebilirsiniz.

https://create.arduino.cc/getting-started/plugin/welcome?_gl=1*_oz7s11*_ga*MTAyMzMzMjAzNS4xNjI2MDY3NDYw*_ga_NEXN8H46L5*MTYyODIzNjY3Mi4xOC4xLjE2MjgyMzcwODMuMA



IoT Cloud ve Amazon Alexa ile LED Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



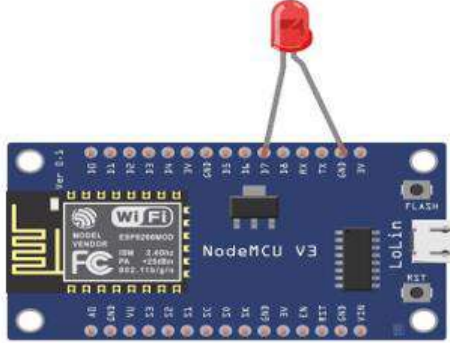
BLYNK IoT
(ANDROID)



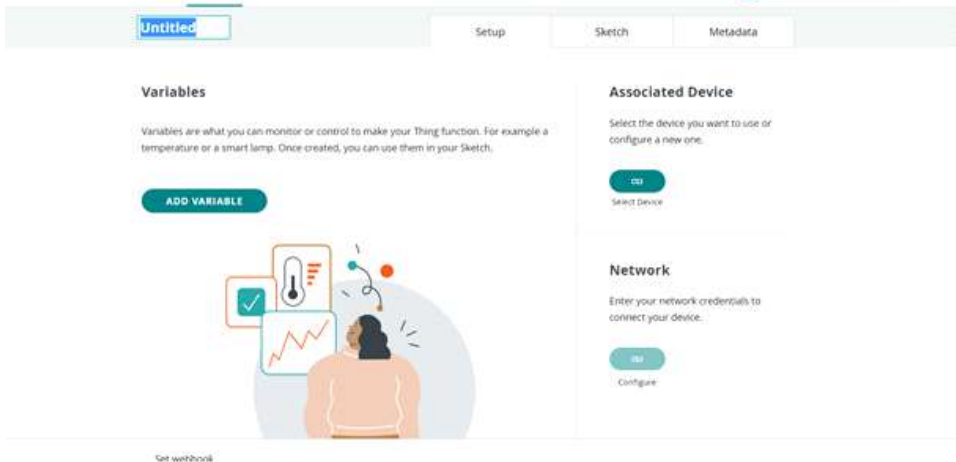
github.com/Robotistan/IoTCloud

1. Bu projede IoT Cloud ve Amazon Alexa Uygulaması ile LED kontrolünü yapacağız. İlk olarak devremizin kurulumu ile başlayalım.

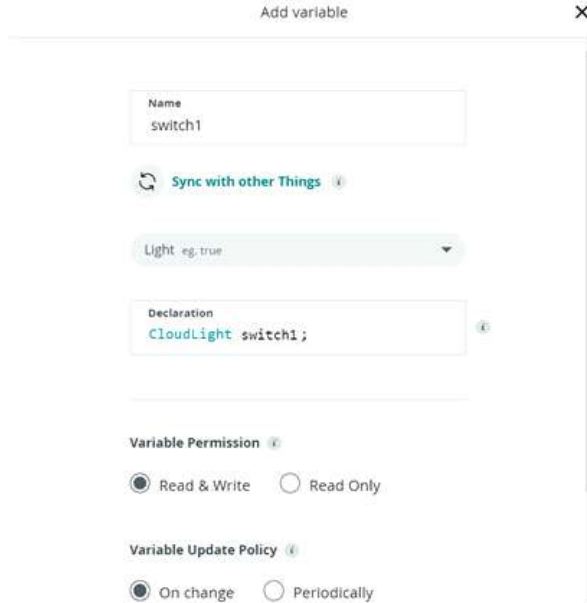
| Arduino IoT cloud ile led kontrolü | |
|------------------------------------|-----|
| LED(+) | D7 |
| LED(-) | GND |



2. Daha önceki adımlarda device oluşturmayı öğrenmiştik, yeni bir device oluşturup things kısmından create things sekmesine giriyoruz. Daha sonrasında associated device kısmına girip oluşturduğumuz device'ı eklemeliyiz. Network kısmından ise projemizin ID'sini ve WiFi adı ve şifremizi eklemeliyiz. Untitled yazan yere istersek projemiz için isim ekleyebiliriz.



3. Daha sonrasında add variable seçeneğine tıklayıp değişkenimizi oluşturmamızdır. Değişkenimize switch1 ismini verdik siz farklı bir isim verebilirsiniz, biz bu projede LED yakıp söndüreceğimiz için Arduino IoT Cloud bizim için light isminde bir değişken eklemiş biz de onu bulup seçiyoruz. Light, bir boolean değişkeni yani 0 ve 1 değerlerini alabiliyor, siz isterseniz direkt bir boolean değişkeni de oluşturabilirsiniz. Daha sonrasında IoT Cloud bizim için aşağıdaki seçenekleri otomatik dolduruyor. LED kullanacağımız için read and write seçeneğini seçebiliriz ve sonrasında add variable seçeneğini seçerek değişkenimizi oluşturuyoruz.



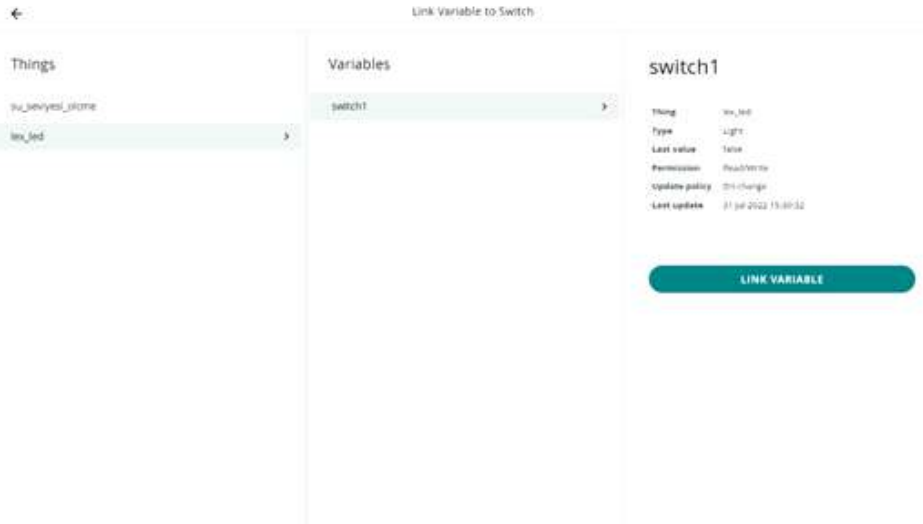
The screenshot shows the 'Add variable' dialog box in the IoT Cloud interface. It contains the following fields and options:

- Name:** switch1
- Sync with other Things:** A toggle switch that is currently turned on.
- Light:** A dropdown menu showing 'Light eg. true'.
- Declaration:** A text area containing the code `CloudLight switch1;`.
- Variable Permission:** Two radio buttons: 'Read & Write' (selected) and 'Read Only'.
- Variable Update Policy:** Two radio buttons: 'On change' (selected) and 'Periodically'.

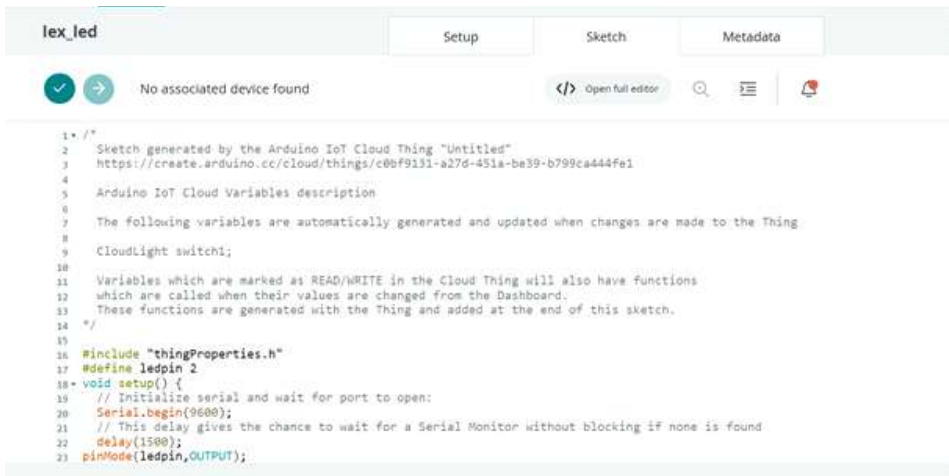
4. Sonrasında Dashboard kısmından build dashboard seçeneğine tıklayarak anahtarımızı projemize eklemeliyiz.



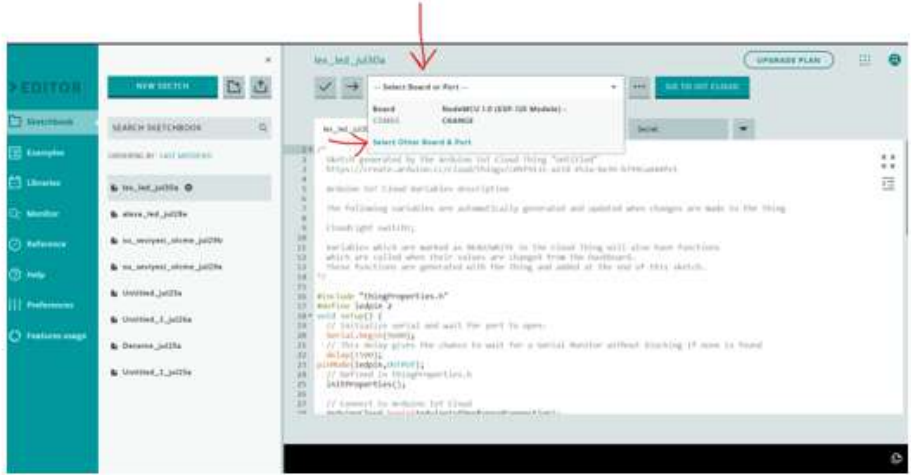
5. Add seçeneğinden switch yazan kısmı seçip, oluşturduğumuz switch1 değişkenini anahtarımıza atamalıyız. Bu aşamada oluşturduğumuz değişken soluk bir renkteyse kullandığımız widget ile değişken türümüz uyuşmuyordur. Örneğin değişkenimizi 6.integer türünde seçseydik IoT Cloud anahtarımıza değişkenimizi atamamıza izin vermeyecektir. Link variable seçeneğine tıklayarak değişkenimizi anahtarımıza atıyoruz.



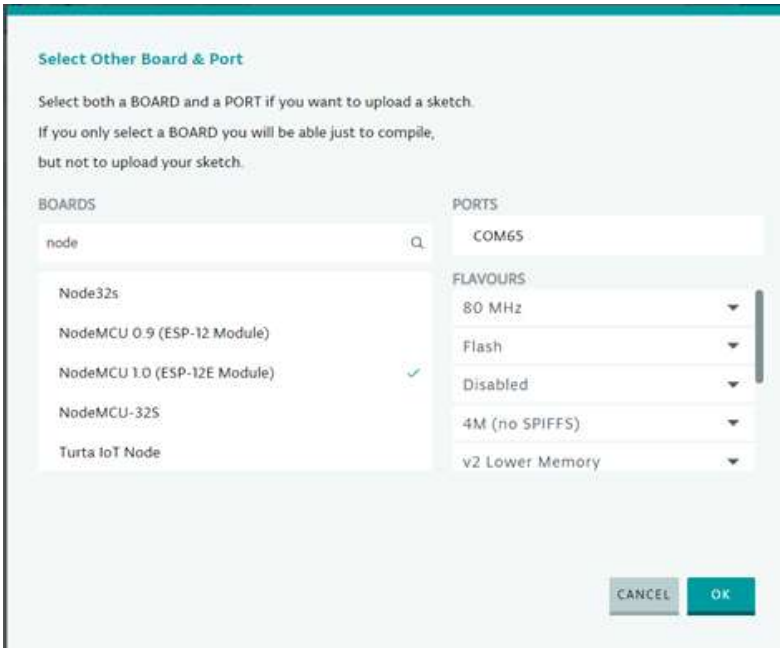
6. Sonrasında things kısmından projemize girip, sketch kısmına gelip kodlarımızı ayarlamalıyız. Bu aşamada open full editor kısmına tıklayıp daha kapsamlı bir kod yazma yerine girebiliriz.



7. Sonrasında NodeMCU kartımızı bilgisayarımıza bağlayıp port seçmeliyiz, işaretlediğimiz yerden select other port and port kısmına girmeliyiz.



8. Sonrasında kartımızı bulup ok seçeneğine tıklamalıyız.



9. Bu aşamadan sonra sadece kodumuzu yazmalıyız. Hadi şimdi kodumuza bakalım.

```

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/c0bf9331-a27d-451a-be39-b799ca44afe1

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

Cloudlight switch1;

variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
#define ledpin 13
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(ledpin, OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  the default is 0 (only errors).
  Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
}

/*
Since Switch1 is READ_WRITE variable, onSwitch1Change() is
executed every time a new value is received from IoT Cloud.
*/

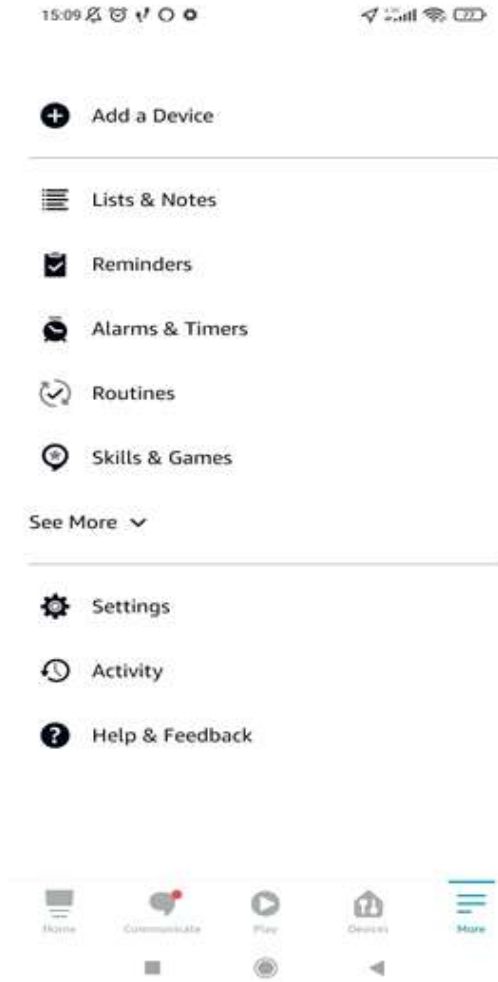
void onSwitch1Change() {
  if (switch1 == 1)
  {
    digitalWrite(ledpin, HIGH);
    Serial.println("Device1 ON");
  }
  else
  {
    digitalWrite(ledpin, LOW);
    Serial.println("Device1 OFF");
  }
}

```

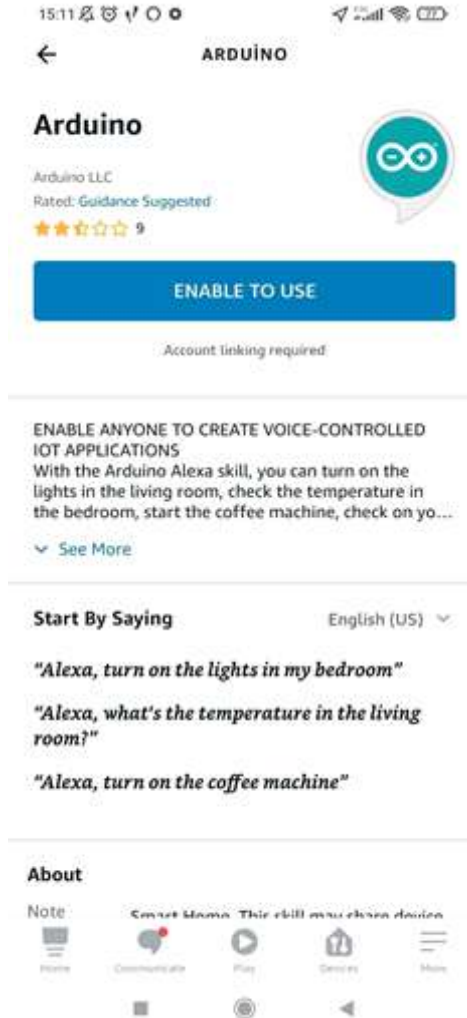
10. Kodumuza baktığımızda değişkenimizin atamasını IoT Cloud'un otomatik yaptığını görüyoruz, fonksiyonunu da kendisi oluşturuyor. LED'imizin pinini atayıp output olduğunu belirttik. Switch1Change fonksiyonumuz içerisinde anahtarımız 1 olduğunda LED'imizin yanıp, seri ekranda device1 on yazmasını, eğer anahtarımız kapalıysa LED'imizin sönmesini ve seri ekranda device1 off yazmasını istedik.

11. Seri ekranda IoT Cloud'un internetimize bağlandığından emin olalım. Connected to "internet isminiz" yazısını görmediğiniz sürece bağlanmamış demektir. Farklı bir internetten denemelisiniz, seri ekranımızda gördüğümüz gibi device on ya da device off yazılarını görmekteyiz.

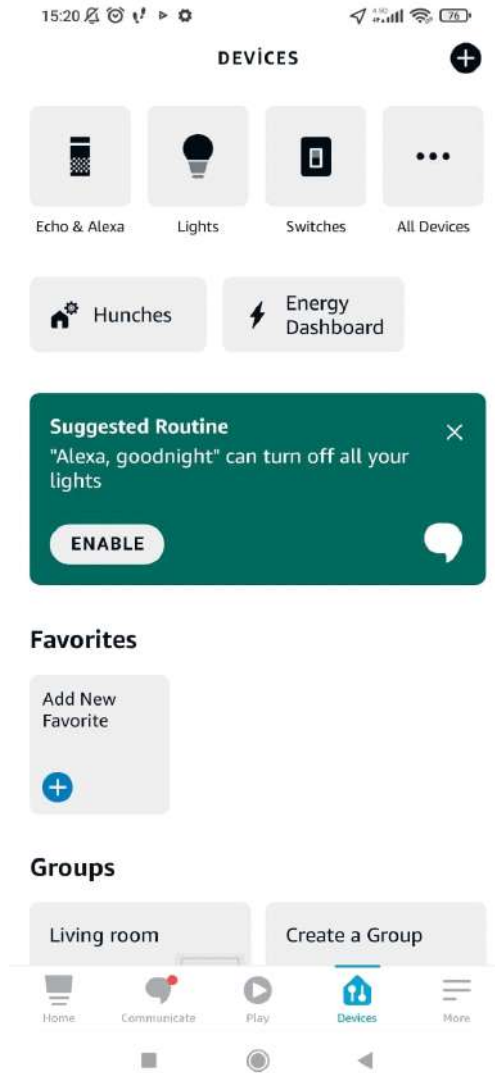
12. Şimdi de Alexa ile kontrolü inceleyeceğiz. Öncelikle Alexa uygulamasını telefonumuza yükleyip bize söylenen adımları takip ediyoruz. Bu ekrana geldiğimizde Skill & Games seçeneğine giriyoruz.



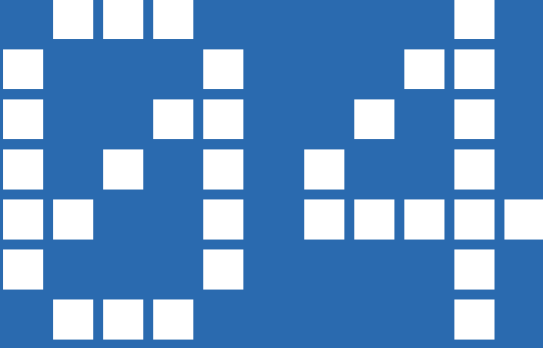
13. Daha sonrasında Arduino yazıp enable to use seçeneğine tıklıyoruz. Bu aşamadan sonra Alexa bizim device'mızı arayıp bulacaktır.



14. Sonrasında device kısmından switch seçeneğine girip oluşturduğumuz device'ı göreceğiz.



15. Artık Alexa ile LED'imizi yakıp söndürebiliriz. Yukarıdaki mavi kısma tıklayıp Alexa turn on the switch one dersek LED'imiz yanacaktır eğer Alexa turn off the switch one dersek LED'imiz sönecektir.



DHT11 İle Sıcaklık ve Nem Ölçümü

BLYNK IoT
(IOS)



PROJELER LİNKİ



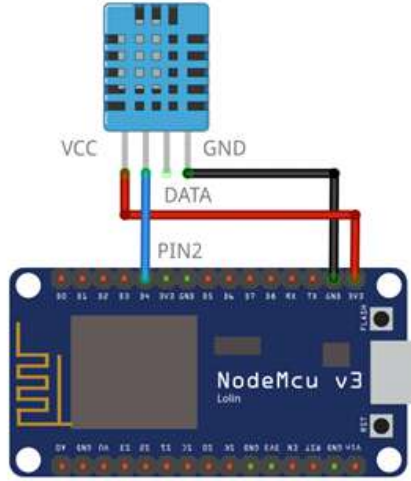
BLYNK IoT
(ANDROID)



github.com/Robotistan/IoTCloud

Bu proje tamamlandığında herhangi bir ortamın, sıcaklık ve nem değerlerini Arduino IoT Cloud uygulaması üzerinden takip edebileceksiniz.

1.Uygulamada kullanacağımız devre elemanlarını hazırlayarak, devre şemamızı oluşturalım.
(NodeMCU WiFi geliştirme kartı, DHT11 sıcaklık nem sensörü, breadboard, jumper kablo)



2. Önceki uygulamamızdaki adımları yeniden yapacağız. DHT11 adında bir device oluşturup projemize başlamalıyız. Daha sonrasında add variables seçeneğinden sıcaklık ve nem değişkenlerimizi eklemeliyiz. İki değişkenimizin de read only seçeneğinde olduğundan emin olalım.

nem

Declaration
CloudRelativeHumidity nem

Type
Relative humidity

Variable Permission
Read Only

Send Values
On change threshold: 0

ID
01168c80-cb41-4c3c-afbe-674acb7a80ab

Last Value
66

Last Update
02 Aug 2022 11:08:47

sıcaklık

Declaration
CloudTemperatureSensor sıcaklık

Type
Temperature sensor (°C)

Variable Permission
Read Only

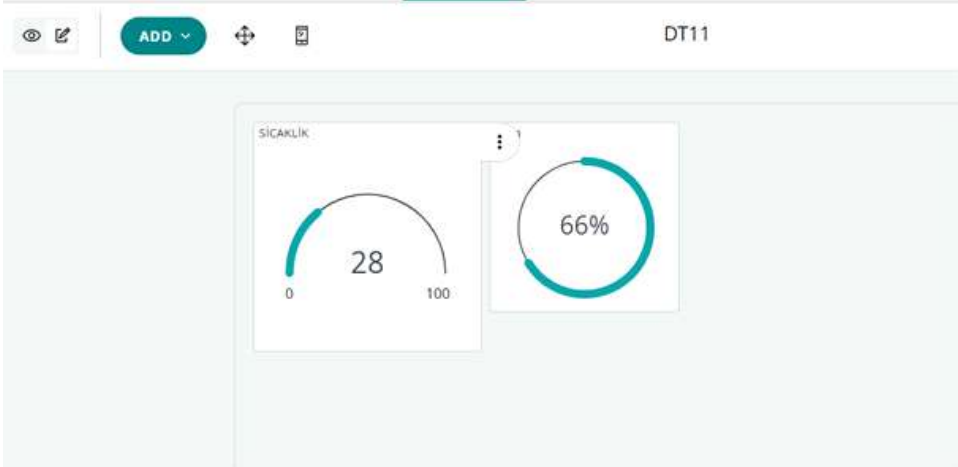
Send Values
On change threshold: 0

ID
7499ef7a-0e93-43d0-993e-1aab9b070134

Last Value
28

Last Update
02 Aug 2022 11:08:44

3. Daha sonrasında dashboards kısmına gelip sıcaklık için bir gauge, nem için bir tane percentage ekliyoruz ve değişkenlerimizin atamasını yapıyoruz.



4. Daha sonrasında sketch kısmında kodumuzu yazacağız. DHT11 kullandığımız için DHT kütüphanemizi eklemeliyiz.

DHT SENSOR LIBRARY

Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors [More info](#)

> Examples

```
// DHT sensor library - Version: Latest
#include <DHT.h>
#include <DHT_U.h>

/*
 * Sketch generated by the Arduino IoT Cloud Thing "Untitled"
 * https://create.arduino.cc/cloud/things/e9e1879e-8306-407e-bef4-2e8760700ea8
 *
 * Arduino IoT Cloud Variables description
 *
 * The following variables are automatically generated and updated when changes are made to the Thing:
 * CloudTemperatureSensor sicaklik;
 * CloudRelativeHumidity nem;
 *
 * Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
 * which are called when their values are changed from the Dashboard.
 * These functions are generated with the Thing and added at the end of this sketch.
 */

#include "thingProperties.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  dht.begin();
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
   * The following function allows you to obtain more information
   * related to the state of network and IoT Cloud connection and errors.
   * the higher number the more granular information you'll get.
   * The default is 0 (only errors).
   * Maximum is 4
   */
  setDebugLogLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  DHT_SENSOR_READ();
}

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

  Serial.println(t);

  Serial.print("Temperature : ");
  Serial.print(t);
  Serial.print(" Humidity : ");
  Serial.println(h);

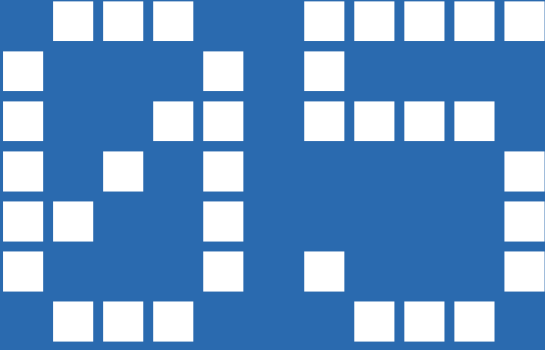
  delay(1000);
}

```

5. Seri ekranımızda ve widgetlar'ımızda değerlerimizi görebiliyoruz.

[illegible]

6. Kodumuzda önce DHT digital pinimizin atamasını yapıyoruz, daha sonrasında DHT_SENSOR_READ adında bir fonksiyon oluşturuyoruz. Float türünde sıcaklık ve nem değerimizi tutacak 2 değişken oluşturuyoruz. Sensör hatasında ekrana yazı yazması için bir if bloğu yazıyoruz. Hata olmadığı zamanlar için ise seri ekranımıza ve widget'larımıza veri göndermek için birkaç satır kod daha yazıyoruz.



RGB LED Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



BLYNK IoT
(ANDROID)



github.com/Robotistan/IoTCloud

1. Bu projede IoT Cloud uygulaması ile RGB diyot LED kontrolünü yapacağız. Gece lambası ya da aksesuar gibi bir projeye çevirerek, evinizde güzel bir ortam elde edebilirsiniz.
2. Öncelikle device kısmından RGB isminde projemizi oluşturuyoruz. Daha sonrasında değişkenlerimizi projemize ekleyeceğiz. V0, V1, V2 adında bool türünde üç adet değişken oluşturacağız.

The screenshot shows the IoT Cloud interface. On the left, the 'Declaration' section shows a variable named 'v0' of type 'bool'. Below it, the 'Variable Permission' is set to 'Read & Write'. The 'Send Values' section is set to 'On change'. The 'ID' is '5388cc00-5b29-4408-bd34-18334df8a288'. The 'Last Value' is 'false' and the 'Last Update' is '03 Aug 2022 12:56:17'. On the right, a table lists three variables: 'v0', 'v1', and 'v2', all of type 'bool'. Each variable has a 'false' value and a timestamp of '03 Aug 2022 12:56:17'.

| Variable Name | Type | Value | Last Update |
|---------------|------|-------|----------------------|
| v0 | bool | false | 03 Aug 2022 12:56:17 |
| v1 | bool | false | 03 Aug 2022 12:52:47 |
| v2 | bool | false | 03 Aug 2022 12:55:04 |

3. Daha sonra ise dashboards kısmından kırmızı, yeşil ve mavi ayaklarımızı kontrol etmek için 3 adet switch ekliyoruz ve değişkenlerimizi atıyoruz.

The screenshot shows the IoT Cloud dashboard for a project named 'new_rgb'. It features three toggle switches labeled 'v0', 'v1', and 'v2'. Each switch is currently in the 'OFF' position.

4. Daha sonrasında sketch kısmına gelip kodlarımızı yazmalıyız.

```

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/98dab5a3-a34d-4c2f-a521-cd3211bd25f9

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

bool v0;
bool v1;
bool v2;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

#define RED D6
#define GREEN D7
#define BLUE D8

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
  */
  setDebugLogLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  onV0Change();
  onV1Change();
  onV2Change();
}

/*
Since v0 is READ_WRITE variable, onV0Change() is
executed every time a new value is received from IoT Cloud.
*/

```



```

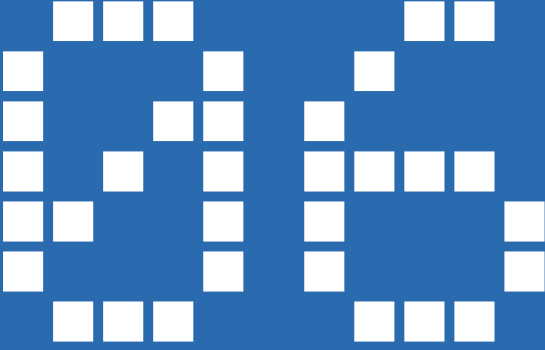
void onV0Change() {
  if(v0 == 1){
    digitalWrite(RED,LOW);
  }
  else
  {
    digitalWrite(RED,HIGH);
  }
}

/*
  Since v1 is READ WRITE variable, onV1Change() is
  executed every time a new value is received from IoT Cloud.
*/
void onV1Change() {
  if(v1 == 1){
    digitalWrite(GREEN,LOW);
  }
  else
  {
    digitalWrite(GREEN,HIGH);
  }
}

void onV2Change() {
  if(v2 == 1){
    digitalWrite(BLUE,LOW);
  }
  else
  {
    digitalWrite(BLUE,HIGH);
  }
}

```

5. Kodumuzda öncelikle kırmızı, yeşil ve mavi ayaklarımızın atamasını yapıyoruz. Daha sonrasında switch'lerimiz 1 olduğu zaman ayaklarımıza low voltage gönderiyoruz. Çünkü bizim kullandığımız RGB ortak anotlu bir RGB'Ydi. Eğer siz ortak katotlu bir RGB kullanıyorsanız low ve high kısımlarının yerlerini değiştirmelisiniz.



Bitki Sulama Sistemi

BLYNK IoT
(IOS)



PROJELER LİNKİ



BLYNK IoT
(ANDROID)

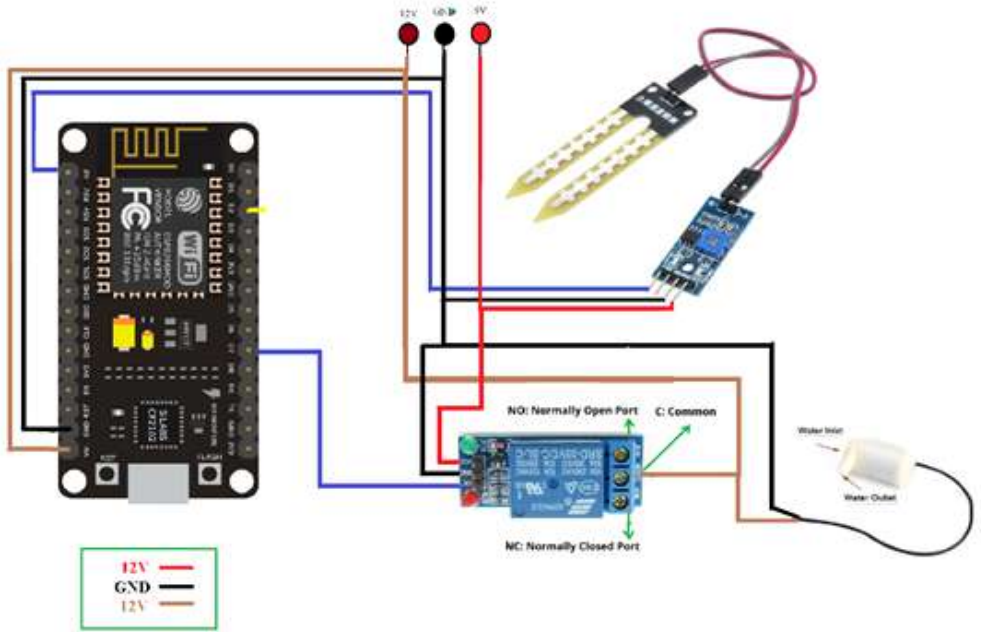


github.com/Robotistan/IoTCloud

Bu projemizde toprak nem sensöründen aldığımız veriler ile rölemiz açılıp kapanacak. Bu olay sonucunda mini dalgıç su pompası toprağımızı sulamış olacaktır.

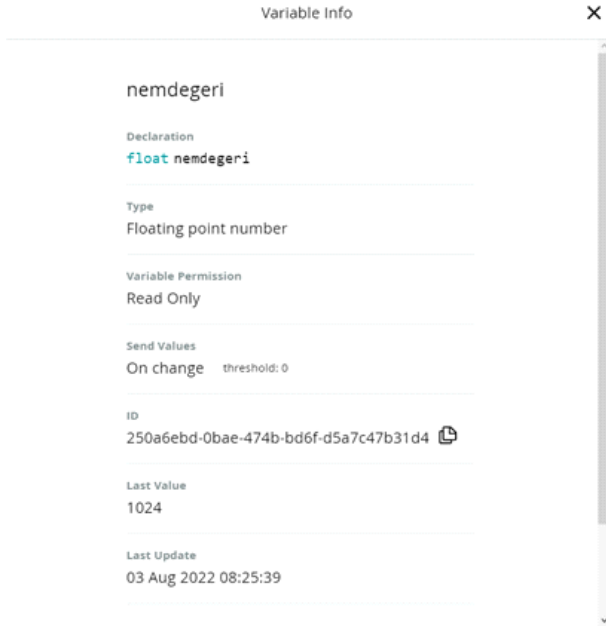
1. İlk olarak devre kurulumu ile başlayalım.

- NodeMCU
- Breadboard
- Jumper Kablolar
- 2 Kanal Röle Modülü
- Toprak-Nem Sensörü
- Mini Dalgıç Pompa



| BİTKİ SULAMA SİSTEMİ | |
|----------------------|-----|
| SENSÖR PİN | A0 |
| SENSÖR VCC | VİN |
| (-) | GND |
| Röle | D7 |

2. Öncelikle device'ımızı oluşturup things kısmına eklemeliyiz. Sonrasında nemdegeri isminde bir float değişkeni oluşturacağız ve read only seçeneğini işaretleyeceğiz.



3. Ardından dashboards kısmına gelip nem değerimizi göreceğimiz widget'ı ekleyeceğiz ve oluşturduğumuz değişkeninin atamasını yapacağız.



4. Sonrasında things sketch kısmından kodlarımızı yazmaya geçeceğiz.

```

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled 2"
https://create.arduino.cc/cloud/things/532295c2-b8c2-4f44-990d-e92a4f8d0939

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

float nemdegeri;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

int nemPin = A0;
#define relay1 D0

int nemEsikdegeri = 550;

void setup() {
  // Initialize serial and wait for port to open:
  pinMode(nemPin, INPUT);
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  pinMode(relay1, OUTPUT);
  digitalWrite(relay1, HIGH);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
  */
  setDebugLogLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  nemdegeri_READ();
}

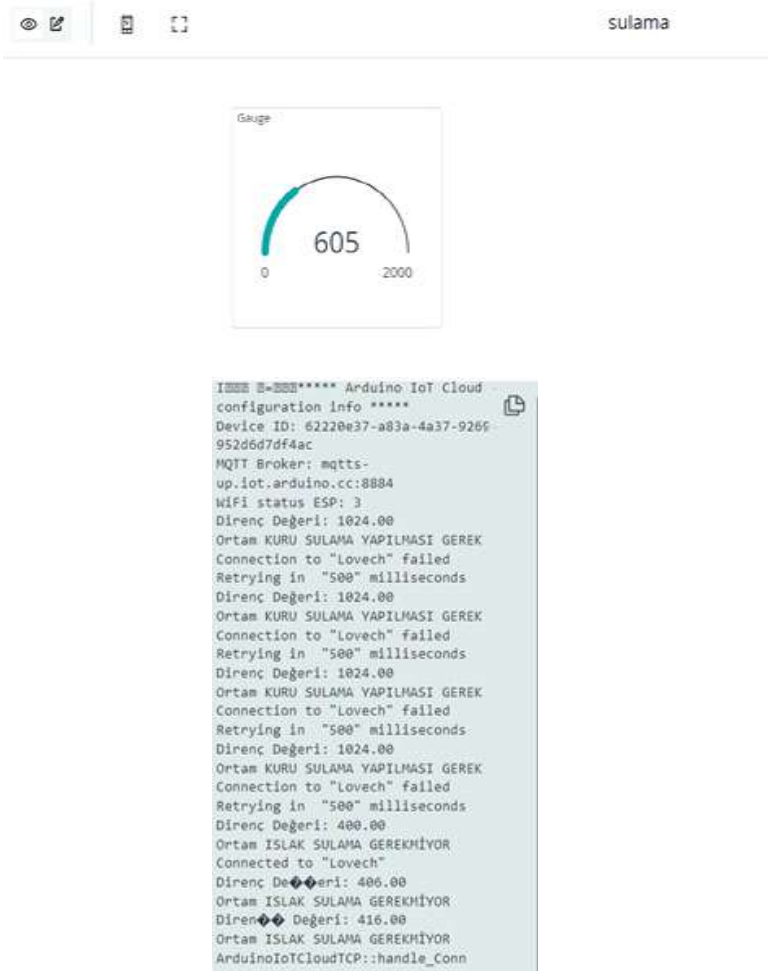
void nemdegeri_READ()
{
  nemdegeri = analogRead(nemPin);

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

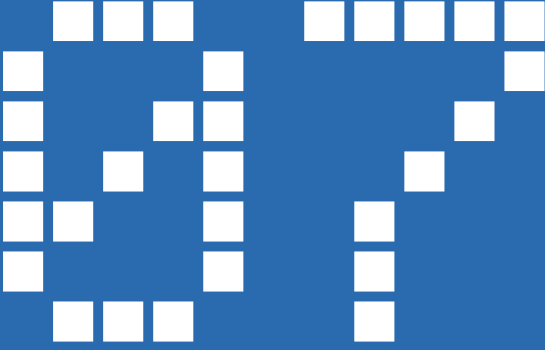
  if(nemdegeri < nemEsikdegeri)
  {
    Serial.println("Ortam ISLAK SULAMA GEREKMİYOR");
    digitalWrite(relay1, LOW);
  }
  else
  {
    Serial.println("Ortam KURU SULAMA YAPILMASI GEREK");
    digitalWrite(relay1, LOW);
  }
  delay(1000);
}

```

5. Seri ekranımızda gördüğümüz gibi sensörümüz boştayken default 1024 değerini veriyor. Suya soktuğumuzda ise yaklaşık olarak 550'nin altında bir değer veriyor bu yüzden eşik değerini böyle ayarladık. Gauge ekranımızda da değerlerimizi görebiliyoruz, internet hızınıza göre gelmesi geç olabilir arada sayfayı yenilemenizi öneririz.



6. Kodumuzda öncelikle nemPin isminde bir değişken oluşturup analog girişimize atamasını yapıyoruz ve input olarak da kartımıza tanıtıyoruz. Eşik değerimizi 550 olarak belirleyip onun da atamasını yapıyoruz. Rölemizin de atamasını yapıp output olarak kullanıyoruz, kodumuz yüklenirken de high olmasını istediğimiz için kodu böyle yazdık. Sonrasında ölçtüğümüz değer ile eşik değerimizi karşılaştırıp sulama yapılıp yapılmaması gerektiğine karar veriyoruz. Sulama yapılacağı zaman rölemiz açılıp pompamız çalışacak ve sulama işlemi gerçekleşecektir.



Servo Motor Kontrolü

BLYNK IoT
(IOS)



PROJELER LİNKİ



BLYNK IoT
(ANDROID)



github.com/Robotistan/IoTCloud

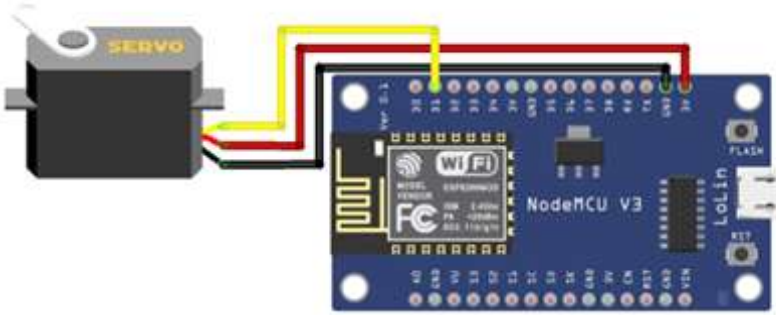
Bu projede WiFi üzerinden Arduino IoT Cloud uygulamasıyla servo motor kontrolü projesi yapacağız. Servo motorun pervanesini taktığımızda kaç derece döndürebildiğimizi rahat bir şekilde görebilecek duruma geleceğiz.

1. İlk olarak devre kurulumu ile başlayalım

Uygulamada Kullanılan Devre Elemanları:

- 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 GND'ye bağlayıp devre kurulumunu tamamlıyoruz.



| SERVO MOTOR KONTROLÜ | |
|----------------------|------------|
| SERVO PİN | D1 |
| VCC | <u>Vin</u> |
| (-) | GND |

2. Öncelikle device'ımızı oluşturup things kısmından ayarlarımızı yapıyoruz, daha sonrasında ise integer türünde değişkenimizi oluşturuyoruz.

haktan

Declaration

`int haktan`

Type

Integer number

Variable Permission

Read & Write

Send Values

On change threshold: 0

ID

d86ddaa9-8192-461f-837c-0128049a1ede

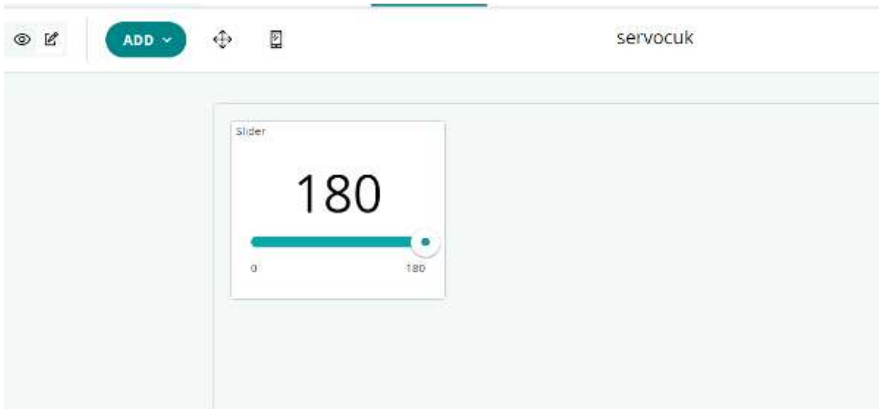
Last Value

180

Last Update

31 Jul 2022 16:21:02

3. Ardından dashboards kısmına gelip servomuzu kontrol edeceğimiz slider'ımızı ekliyoruz ve değişkenimizin atamasını yapıyoruz, 180 derece döndürebileceğimiz için max değerine 180 giriyoruz.



4. Sonrasında things sketch kısmına gelip kodumuzu yazacağız. Öncelikli olarak servo kütüphanemizi eklemeliyiz.

```
// Servo - Version: Latest
#include <Servo.h>

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/64146fea-a6dd-4f22-9cdd-2d86e973d451

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

int haktan;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
Servo servo;

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
  servo.attach(5);
  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

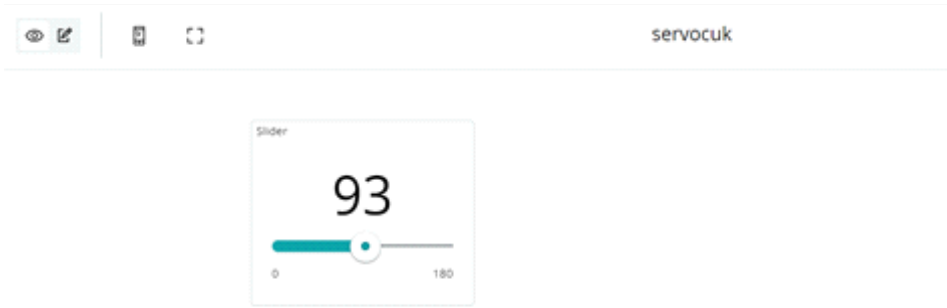
  /*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here

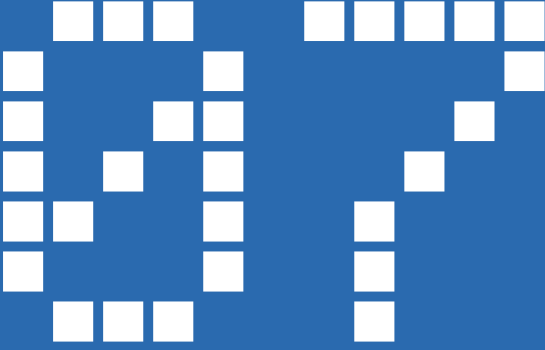
}

/*
Since Haktan is READ_WRITE variable, onHaktanChange() is
executed every time a new value is received from IoT Cloud.
*/
void onHaktanChange() {
  Serial.println(haktan);
  servo.write(5);
}
```

5. Slider'ımızı hareket ettirerek servomuzu kontrol ediyoruz.



6. Kodumuz da önce servo kütüphanemizi ekliyoruz daha sonrasında değişkenimizi atayıp servomuzu TX pinimize atıyoruz. Fonksiyonumuzun içerisinde seri ekranda gözükmesi için ve servonun çalışması için gerekli kodu yazıyoruz.



Duman - Gaz Dedektörü

BLYNK IoT
(IOS)



PROJELER LİNKİ



BLYNK IoT
(ANDROID)



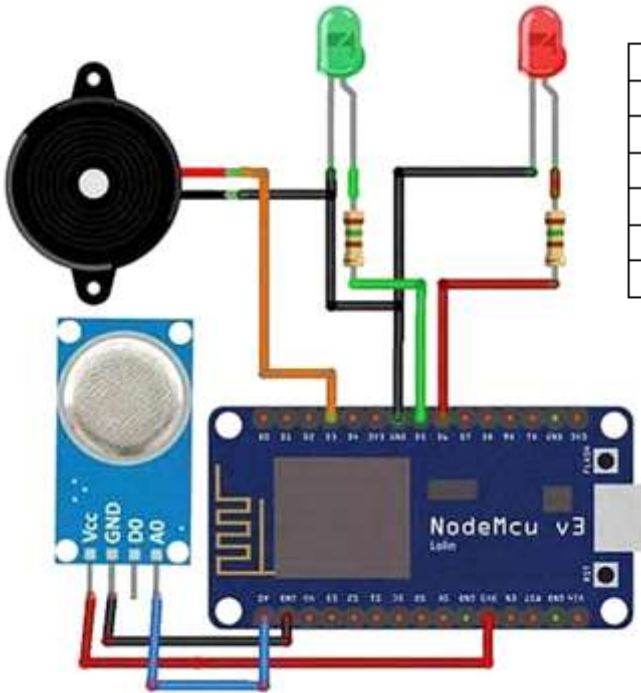
github.com/Robotistan/IoTCloud

Bu projede Arduino IoT Cloud programını kullanarak WiFi ağı üzerinden NodeMCU kartı sayesinde gaz sensöründen gelen veriler ile ortamın gaz ve duman seviyesini ölçebileceksiniz. Veriler belli bir seviye üzerine çıktığında telefonunuza bildirim gelecek ve buzzer çalışacaktır. Böylelikle ortamdaki havanın ne kadar temiz olduğuna karar verebileceksiniz.

Uygulamada Kullanılan Devre Elemanları:

- NodeMCU
- 2 adet LED
- 2 adet 150 ohm Direnç
- MQ2 Gaz Sensörü
- Buzzer
- Breadboard
- Jumper Kablolar

1. Devre şemasında gözüktüğü gibi bağlantıları yapalım. LED'leri D5 ve D6 pinlerine, buzzer'ı 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 PİN | A0 |
| BUZZER PİN | D3 |
| KIRMIZI LED PİN | D6 |
| YEŞİL LED PİN | D5 |
| VCC | 3.3V |
| (-) | GND |

2. Öncelikle device'ımızı oluşturup things kısmında atama işlemini ve WiFi işlemlerini yapıyoruz. Sonrasında değişkenimizi oluşturuyoruz ve readonly seçeneğini işaretliyoruz.

Variable Info

sensorValue

Declaration

`float sensorValue`

Type

Floating point number

Variable Permission

Read Only

Send Values

On change threshold: 0

ID

e89745a8-8aa1-4174-a9f8-a7e148f9ea4d

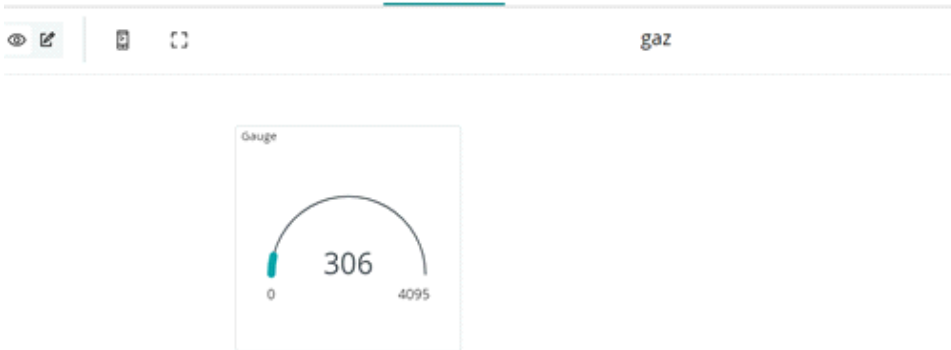
Last Value

306

Last Update

04 Aug 2022 10:20:45

3. Dashboard kısmına gelip gauge ekliyoruz ve değişkenimizi atıyoruz. Max value olarak 4095 giriyoruz. 1000'den fazla girmenizde fayda var.



4. Daha sonrasında things sketch kısmından kodlarımızı yazıyoruz.

```

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/62b3541b-9578-4c22-bf52-1e61e9bb9464

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

float sensorValue;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

#define buzzerPin 0
#define GreenLed 14
#define RedLed 12

#define MQ2pin (A0)

int n,m;
void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking
  delay(1500);
  Serial.println("Gas sensor warming up!");
  delay(5000);
  pinMode(GreenLed, OUTPUT);
  pinMode(RedLed, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(16, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  digitalWrite(GreenLed, LOW);
  digitalWrite(RedLed, LOW);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  void sensorValue_READ(){
    ArduinoCloud.update();
    sensorValue = analogRead(MQ2pin);

    Serial.println("");
    Serial.print("Sensor Value: ");
    Serial.print(sensorValue);
    Serial.println("");

    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);
    }
  }
}

```

5. Seri ekranımızda normaldeki değerleri ve çakmak gazı tuttuğumuzdaki değerleri görebilmekteyiz. Gauge ekranımızda da değerimizi görüyoruz.

```
Sensor Value: 257.00
Sensor Value: 257.00
Sensor Value: 256.00
Connected to Arduino IoT Cloud
Thing ID: 62b3541b-9578-4c22-bf52-
1e61e9bb9464
Sensor Value: 256.00
Sensor Value: 255.00
Sensor Value: 255.00
Sensor Value: 662.00
663
Sensor Value: 665.00
666
Sensor Value: 645.00
645
Sensor Value: 610.00
610
Sensor Value: 570.00
Smoke detected!
Sensor Value: 453.00
Smoke detected!
```



6. Kodumuzu yazarken öncelikle yeşil, kırmızı LED'lerimizin ve buzzer'ın değişken atamalarını ve kullandığımız analog pinin makro atamasını yapıyoruz. Değişkenlerimizi çıkış ya da giriş olacak şekilde ayarlıyoruz. Read only seçeneğini seçtiğimiz için IoT Cloud bizim için fonksiyon oluşturmuyor. Kendimiz sensorValue_READ isminde bir fonksiyon oluşturuyoruz. Oluşturduğumuz if else blokları ile projemizi kontrol ediyoruz.

```
/*
The following function allows you to obtain more information
related to the state of network and IoT Cloud connection and errors
the higher number the more granular information you'll get.
The default is 0 (only errors).
Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

void loop() {
  sensorValue_READ();
}
```




youtube.com/robotistan

FORUM

robotistan



forum.robotistan.com

BLOG

robotistan



maker.robotistan.com

Robotistan Elektronik Ticaret A.Ş.

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