<http://www.instructables.com/id/Quick-Start-to-Nodemcu-ESP8266-on-Arduino-IDE/> Bu adreste kaynak var.

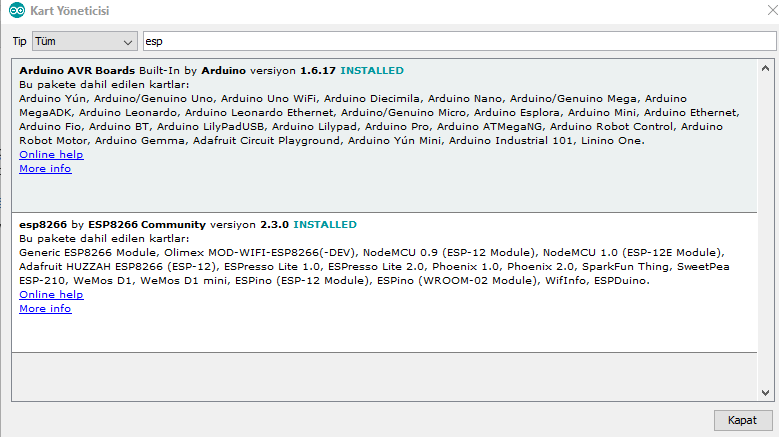
1. Tlf a MQTT dashboard kur
2. Github
3. 1-Arduino ide

2- Pyton/pie platformio ikisinden birini kurunuz.

1. arduino-1.8.1-windows kurduk
2. dosya tercihler aşağıdaki linki yapıştır. (ESP kütüphanesini almsı için)
3. <http://arduino.esp8266.com/stable/package_esp8266com_index.json> adresini yapıştır.

3-Araçlar/Kart/Kart yöneticisi (en üstte) gir ve

ESP arattır.



Alttaki kütüphaneyi kur. (ESP8266)

4-kart içinden ESP modülünü seç. (Biz NODEMCU 09 ESP yi seçtik)

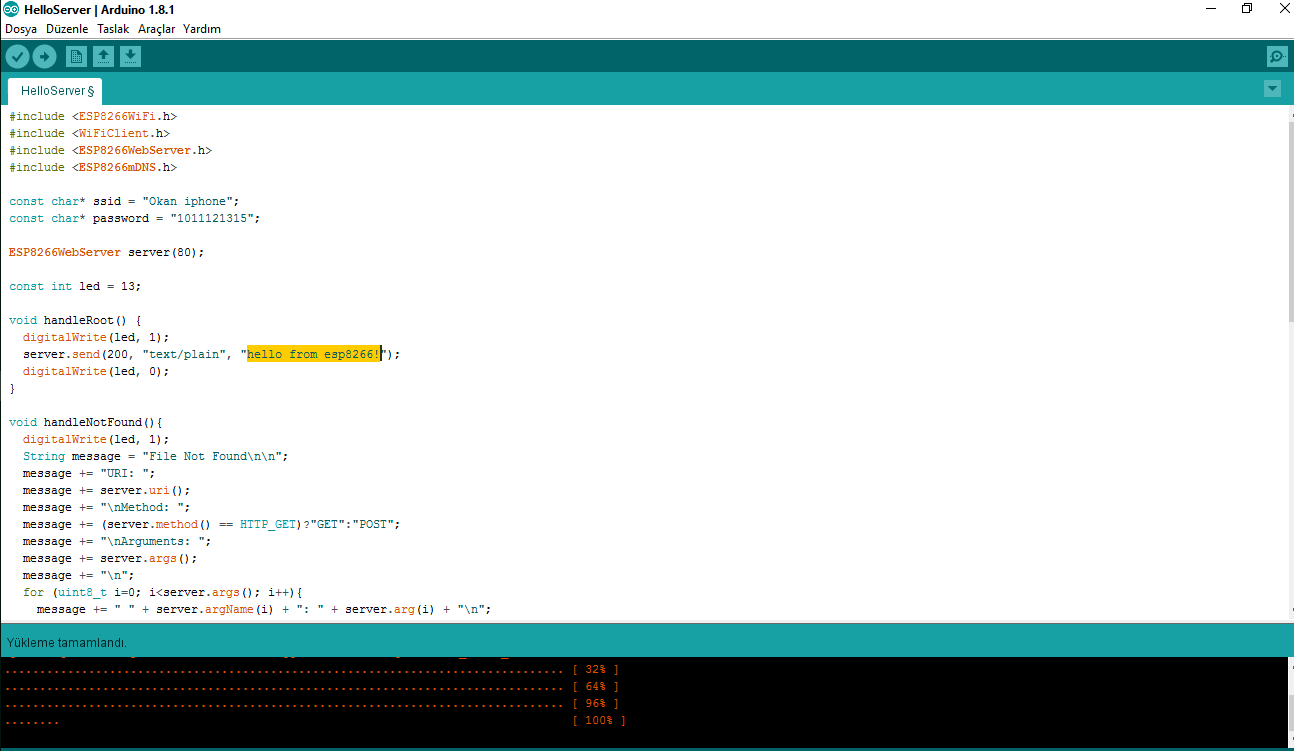
ROS Server, AMAZON

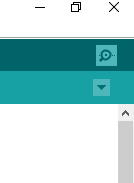
5.ESP Takınca port görmedi (Windows da arduino prog port4)

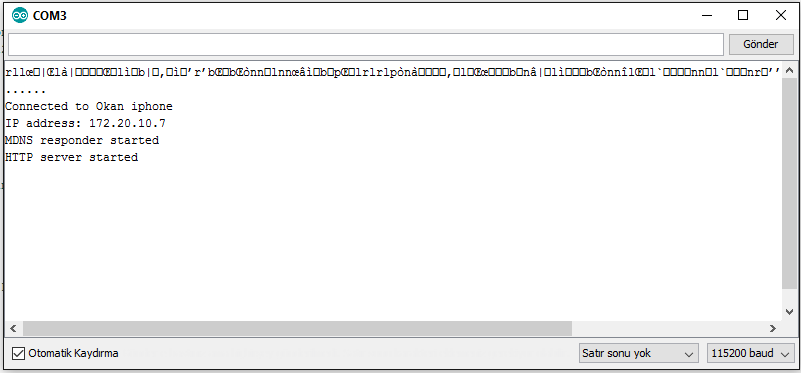
ch340g driver isimli driver indirip kurduk problem kalktı.

6.arduino içinden örnekler içindeki esp altından espweb server seçtik. Açılan kod içine

Telefondan ssıd ayarla (Telefon internet paylaşımı) hello worl u göreceksin



7.Sağ üstteki  den önizle. Açılan ip yi web serwera yaz . jello World mesajını göreceksin.

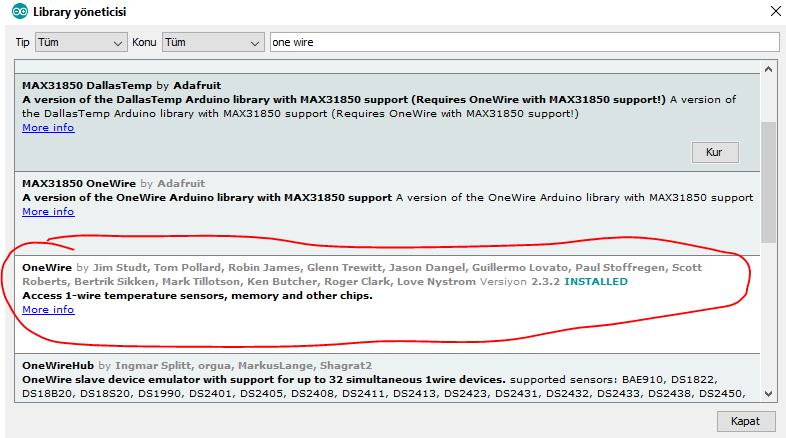


8.örnek2. [esp\_temperature](https://github.com/karakaplanm/iot/tree/master/esp_temperature) örneği

[https://github.com/karakaplanm/iot](https://mail.akdeniz.edu.tr/owa/redir.aspx?C=HDafwubJBI6E2gCah1MpMN0L3XqDH69JzFcub0przdvyVTsn_0zUCA..&URL=https%3a%2f%2fgithub.com%2fkarakaplanm%2fiot) adresinden kodu alalım.

|  |
| --- |
| #include <ESP8266WiFi.h> //Esp8266 Wifi kütüphanesi |
|  | #include <PubSubClient.h> //MQTT için Publisher ve Subscriber Kütüphanesi |
|  | #include <OneWire.h> // OneWire kütüphanesini ekliyoruz. |
|  |  |
|  | #define wifi\_ssid "trmn" |
|  | #define wifi\_password "adem6060" |
|  | #define mqtt\_server "54.89.221.74" |
|  | #define mqtt\_user "ab2017" |
|  | #define mqtt\_password "abx" |
|  | #define topic1 "" |
|  | #define topic2 "temperature" |
|  | #define topic3 "speed" |
|  | #define readSize 6 |
|  |  |
|  |  |
|  | void callback(char\* topic, byte\* payload, unsigned int length); |
|  | WiFiClient espClient; |
|  | PubSubClient client(mqtt\_server, 1883, callback, espClient); |
|  |  |
|  | // Ledi 12. pine bağlıyoruz |
|  | int ledPin = 12; |
|  | // Alınan veriyi 0. pine bağlıyoruz |
|  | int receivedData=0; |
|  | // Sıcaklık sensörünü bağladığımız dijital pini 2 olarak belirliyoruz. |
|  | int DS18S20\_Pin = 2; |
|  | // Sıcaklık Sensörü Giriş-Çıkışı |
|  | OneWire ds(DS18S20\_Pin); // 2. Dijital pinde. |
|  |  |
|  | void setup() { |
|  |  |
|  | // Ledi bağladığımız 12. pin için çıkış tanımlaması |
|  | pinMode(ledPin, OUTPUT); |
|  | //Arduino üzerinde kullandığımız Serial portun Baud Rate 'ini 115200 olarak ayarlıyoruz |
|  | Serial.begin(115200); |
|  | // Wifi kurulumu için oluşturduğumuz fonksiyonu çağırıyoruz |
|  | setup\_wifi(); |
|  | // Topic3 ten gelen veriyi subscribe ediyor |
|  | client.subscribe(topic3); |
|  | } |
|  |  |
|  | // Alınan veriyi yazmak için callback fonksiyonu tanımlıyoruz. |
|  | void callback(char\* topic, byte\* payload, unsigned int length) { |
|  | char rxBuffer[readSize]; //Gelen datayı okumak için bir Buffer dizisi tanımlıyoruz. |
|  | Serial.println(); |
|  | Serial.print("Incoming data: "); //Gelen datayı seri port üzerinden yolla. |
|  | //Serial.println(topic); |
|  | for(int i=0; i<readSize; i++){ //Gelen datayı diziye kaydet. |
|  | if(rxBuffer[i-2]!='.'){ //Gelen datayı parse etmek için kullanılan koşul. |
|  | rxBuffer[i]=payload[i]; |
|  | } |
|  | else i=readSize-1; |
|  | } |
|  | //Serial.print(rxBuffer); |
|  |  |
|  | receivedData = atoi(rxBuffer); //Buffer daki datayı int e çevir. |
|  | Serial.println(receivedData); //Parse edilmiş gelen datayı seri port üzerinden yolla. |
|  | } |
|  |  |
|  | void setup\_wifi() { // WIFI ye bağlanmak için oluşturulmuş fonksiyon. |
|  | delay(10); |
|  | Serial.println(); |
|  | Serial.print("Connecting to "); |
|  | Serial.println(wifi\_ssid); |
|  | WiFi.begin(wifi\_ssid, wifi\_password); |
|  | while (WiFi.status() != WL\_CONNECTED) { |
|  | delay(500); |
|  | Serial.print("."); |
|  | } |
|  | Serial.println(""); |
|  | Serial.println("WiFi connected"); |
|  | Serial.println("IP address: "); |
|  | Serial.println(WiFi.localIP()); |
|  |  |
|  | } |
|  |  |
|  | void reconnect() { //Eğer cihaz server a bağlanamazsa sürekli bağlanmayı denemesi için yazılmış fonksiyon. |
|  | while (!client.connected()) { |
|  | Serial.print("Attempting MQTT connection..."); |
|  | if(client.connect("TestMQTT", mqtt\_user, mqtt\_password)) { |
|  | Serial.println("connected"); |
|  | } |
|  | else { |
|  | Serial.print("failed, rc="); |
|  | Serial.print(client.state()); |
|  | Serial.println(" try again in 5 seconds"); |
|  | delay(5000); |
|  | } |
|  | } |
|  |  |
|  | } |
|  |  |
|  | void pubMQTT(String topic,float topic\_val){ //MQTT üzerinden bir datayı yayımlamak için gerekli fonksiyon. |
|  | Serial.print("Newest topic " + topic + " value:"); |
|  | Serial.println(String(topic\_val).c\_str()); |
|  | client.publish(topic.c\_str(), String(topic\_val).c\_str(), true); |
|  | } |
|  |  |
|  | long lastMsg = 0; |
|  |  |
|  | int receivedLast=0; |
|  |  |
|  | void loop() { // Sürekli döngü. |
|  | if (!client.connected()) { //Eğer cihaz MQTT yer bağlı değilse bağlanıp okumayı dene. |
|  | reconnect(); |
|  | client.subscribe(topic3); |
|  | } |
|  |  |
|  | client.loop(); |
|  |  |
|  | //2 seconds |
|  | long now = millis(); |
|  | if (now - lastMsg > 2000) { |
|  | lastMsg = now; |
|  | //Eğer gelen data bir önceki gelen data ile aynı olup olmadığını karşılaştır. |
|  | if(receivedData!=receivedLast){ |
|  | analogWrite(ledPin, receivedData); |
|  | receivedLast=receivedData; |
|  | } |
|  | // temperature değişkenini sıcaklık değerini alma fonksiyonuna bağlıyoruz. |
|  | float temperature = getTemp(); |
|  | // Sensörden gelen sıcaklık değerini Serial monitörde yazdırıyoruz. |
|  | Serial.print("Sicaklik: "); |
|  | Serial.println(temperature); |
|  | //Publish Values to MQTT broker |
|  | pubMQTT(topic2,temperature); |
|  |  |
|  | } |
|  | } |
|  |  |
|  | // Aşağıdaki fonksiyon DS18B20 sıcaklık sensörümüzden gelen verileri |
|  | // Celcius cinsinden sıcaklık değerlerine çevirmek için kullanılıyor. |
|  | // Herhangi bir değişiklik yapmamız gerekmiyor. |
|  |  |
|  | float getTemp(){ |
|  | //returns the temperature from one DS18S20 in DEG Celsius |
|  |  |
|  | byte data[12]; |
|  | byte addr[8]; |
|  |  |
|  | if ( !ds.search(addr)) { |
|  | //no more sensors on chain, reset search |
|  | ds.reset\_search(); |
|  | return -1000; |
|  | } |
|  |  |
|  | if ( OneWire::crc8( addr, 7) != addr[7]) { |
|  | Serial.println("CRC is not valid!"); |
|  | return -1000; |
|  | } |
|  |  |
|  | if ( addr[0] != 0x10 && addr[0] != 0x28) { |
|  | Serial.print("Device is not recognized"); |
|  | return -1000; |
|  | } |
|  |  |
|  | ds.reset(); |
|  | ds.select(addr); |
|  | ds.write(0x44,1); // start conversion, with parasite power on at the end |
|  |  |
|  | byte present = ds.reset(); |
|  | ds.select(addr); |
|  | ds.write(0xBE); // Read Scratchpad |
|  |  |
|  | for (int i = 0; i < 9; i++) { // we need 9 bytes |
|  | data[i] = ds.read(); |
|  | } |
|  |  |
|  | ds.reset\_search(); |
|  |  |
|  | byte MSB = data[1]; |
|  | byte LSB = data[0]; |
|  |  |
|  | float tempRead = ((MSB << 8) | LSB); //using two's compliment |
|  | float TemperatureSum = tempRead / 16; |
|  |  |
|  | return TemperatureSum; |
|  |  |
|  | } |

Arduino dan Taslak gir, library ekle/ library düzenle oraya one wire yazarak kütüphaneyi (OneWire by Jim Studt….) kuruyoruz. Bu bir sensör haberleşmesi için kütüphanedir.



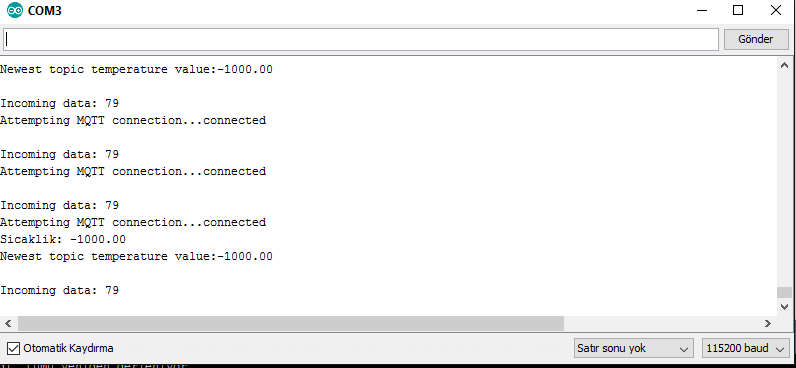
Tempeture sensörü için kütüphane linki

<https://github.com/knolleary/pubsubclient/releases/tag/v2.6>

source code.zip dosyasını indirip ekliyoruz.

Taslak/library ekle den zip ekle diyerek dosyayı(pubsubclient-2.6.zip) ekliyoruz.

Üstteki kodu yükleyip, telefon wifi sini kod içine yazıp derleyince aşağıdaki sonuç ekranını görüyoruz.

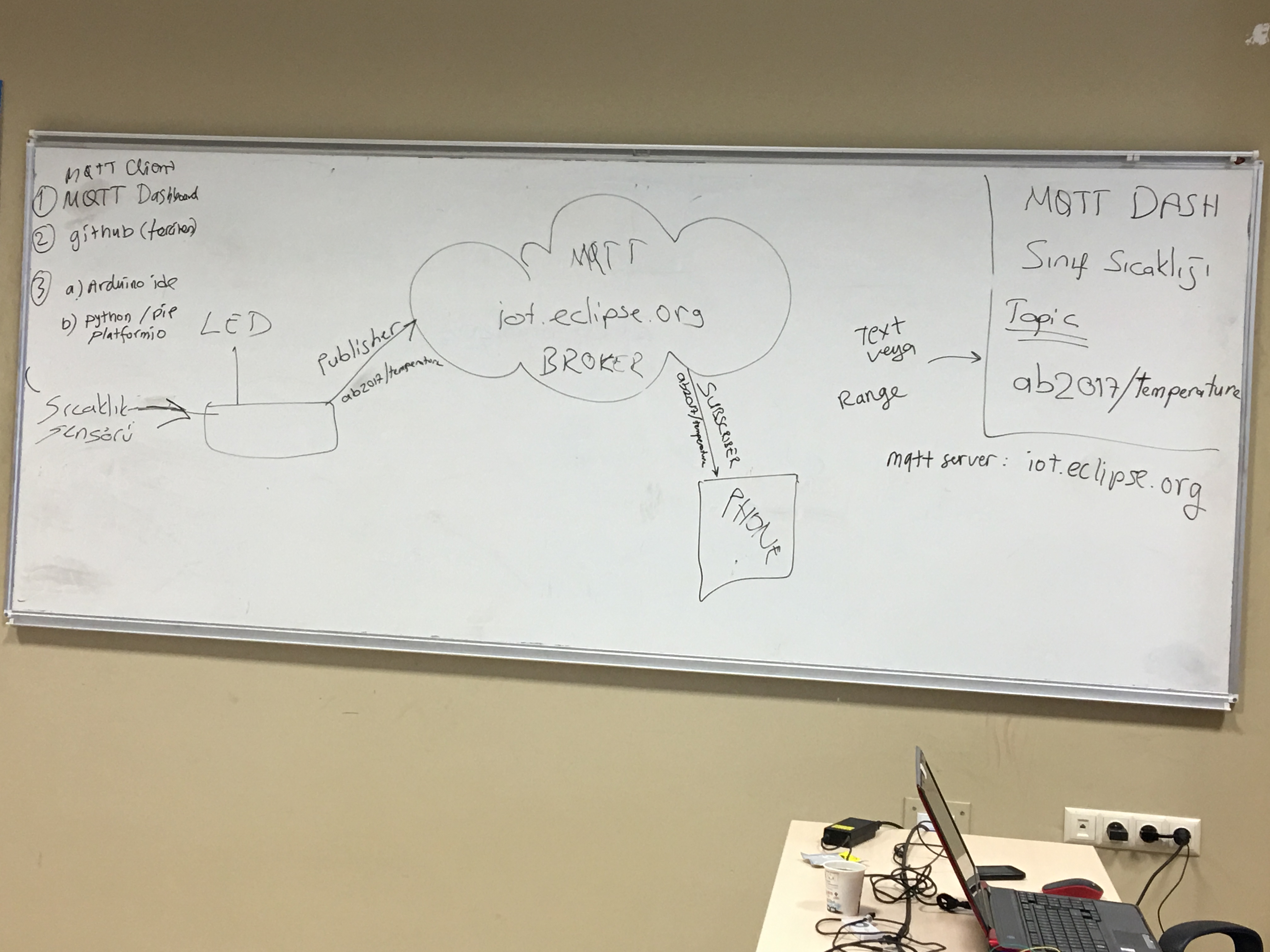


\*\*\*Seri portta bişey 115200 bout ile çalışır.

Örnek2

LED

ESP



Sıcaklık sensörü

#define topic1 "ab2017/okan" topic 1 in komutunu yandaki gibi değiştirdik.

#define topic2 "ab2017/temperature"