NHẬP MÔN KỸ THUẬT ĐTVT

TUẦN 8: CƠ SỞ KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG (PHẦN 3)

BO MẠCH TỰ LÀM

- Bo mạch tự làm gồm có 2 phần như sau
- Phần 1: Mạch tạo xung 1s dùng linh kiện 555, đèn chớp tắt trong chu kỳ 1s
- Phần 2: Mạch khuếch đại âm thanh, ngõ vào là âm thanh, ngõ ra nối ra loa (nên dùng loa nhỏ 3W, 4Ω), âm thanh sẽ được khuếch đại.

BO MẠCH TỰ LÀM

- Linh tải về:

  General/Files/Bo mạch tự làm và code ESP8266

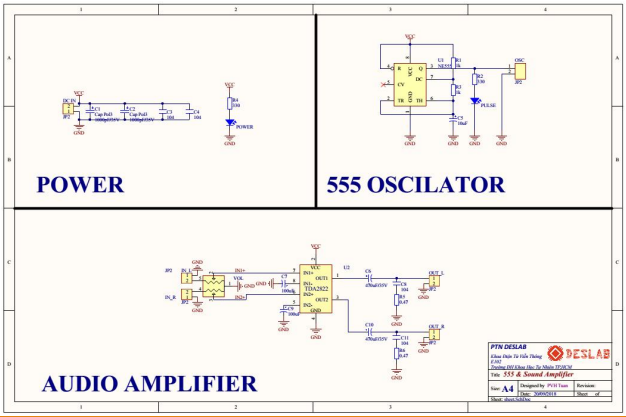
- Toàn bộ thiết kế (schematics và layout) dùng phần mềm Altium v.15
- Tập tin layout dạng PDF (in ra để ủi mạch)
- Tập tin hình bo mạch ở thiết kế 3D
- Tập tin gắn linh kiện (lớp silkscreen)

BO MẠCH TỰ LÀM

- Các linh kiện sau (BOM):
  - 555: SLx1
  - TDA2822: SLx1
  - LED: SLx2
  - Điện trở
  - Tụ điện
- Xem thêm các linh kiện trong tập tin **General/Files/Bo mạch tự làm và code ESP8266/BOM_ds_linh_kiện.xlsx**

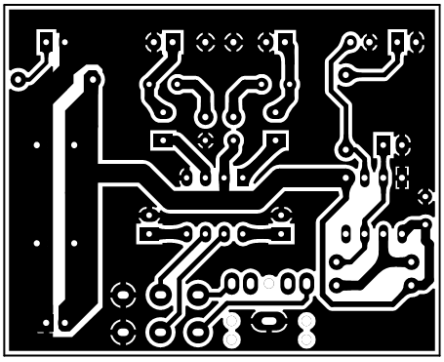
BO MẠCH TỰ LÀM

- Thiết kế schematics



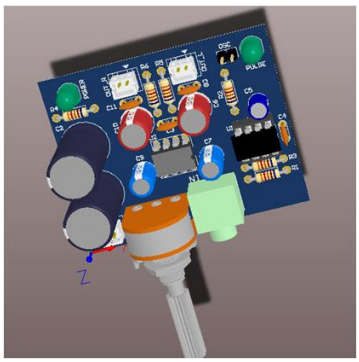
BO MẠCH TỰ LÀM

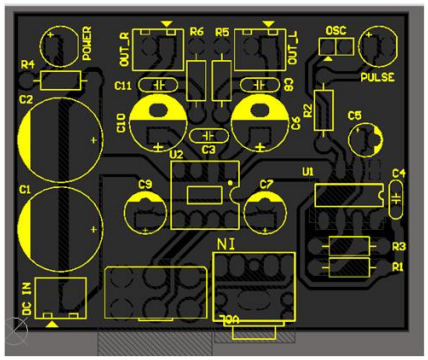
- Thiết kế layout của mạch (bản in)



BO MẠCH TỰ LÀM

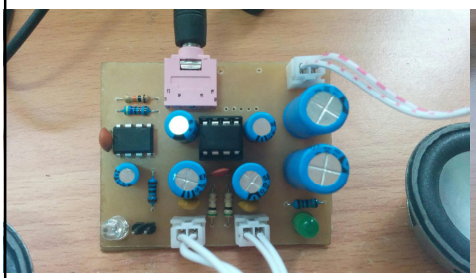
- Hình mạch dạng 3D và lớp linh kiện





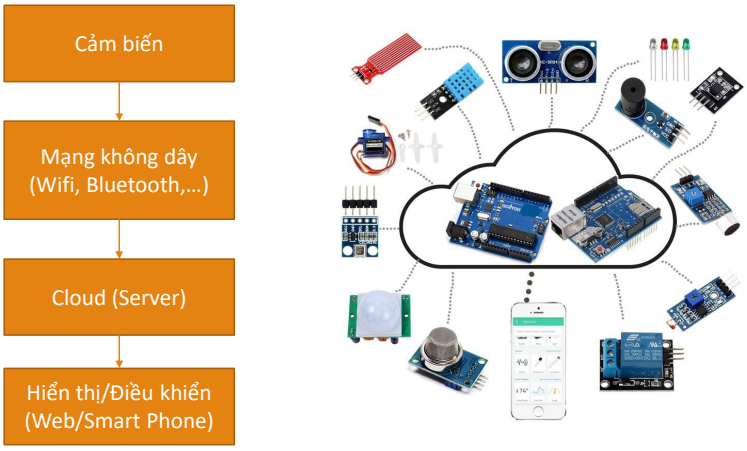
BO MẠCH TỰ LÀM

- Hình mạch sau khi hoàn thành



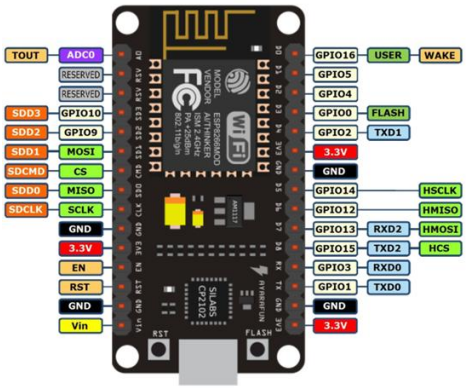
- **Video mạch:** [Mạch tự làm]

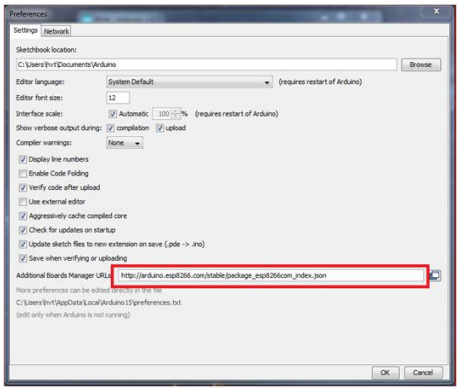
HỆ THỐNG IOT CƠ BẢN



LẬP TRÌNH ESP8266 BẰNG ARDUINO IDE

- Node MCU (ESP8266)
- Node MCU là một IoT platform mã nguồn mở được phát triển dựa trên chip WIFI ESP8266EX bên trong module ESP8266 v12
- Có chức năng kết nối Wifi hoặc làm Access point

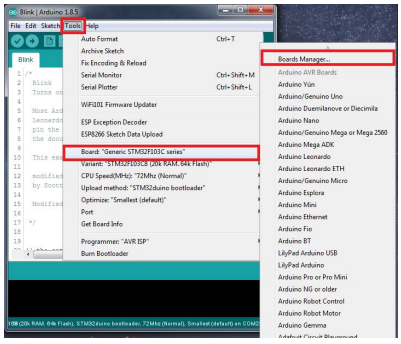


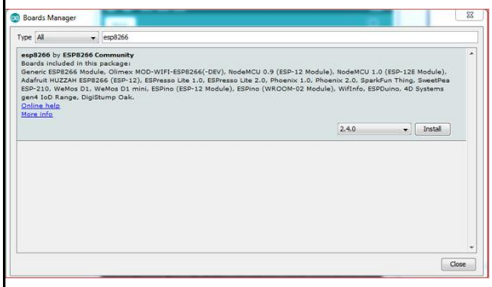


LẬP TRÌNH ESP8266 BẰNG ARDUINO IDE

- Cài đặt
- Đã cài Arduino IDE: https://www.arduino.cc/en/Main/Software
- Sau khi cài xong mở Arduino IDE lên, vào phần **File->Preferences**. Tại phần **Additional Boards Manager URLs** thêm đường dẫn sau và và ấn OK :

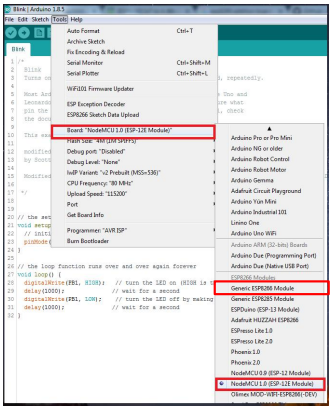
http://arduino.esp8266.com/stable/package_esp8266com_index.json.





LẬP TRÌNH ESP8266 BẰNG ARDUINO IDE

- Vào **Tools->Board->Boards Manager…**
- Tìm module ESP8266 và ấn install.
- Sau khi install xong nhấn Close để đóng cửa sổ Boards Manager lại và dùng cáp kết nối Node MCU với máy tính.



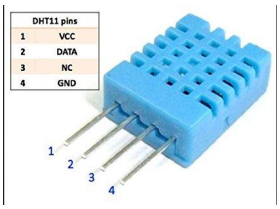
LẬP TRÌNH ESP8266 BẰNG ARDUINO IDE

- Chạy demo
- Vào Tools/Boards chọn module Node MCU (ESP8266 V12e Module).
- Code Blink LED tại chân D4 (pin 2)

```
int LED = 2;
 void setup() {
 pinMode(LED, OUTPUT);
 }
 void loop() {
 digitalWrite(LED, HIGH); // turn the LED on (HIGH
is the voltage level)
 delay(1000); // wait for a second
 digitalWrite(LED, LOW); // turn the LED off by
making the voltage LOW
 delay(1000); // wait for a second
 }
```

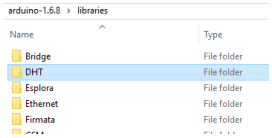

ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Cảm biến DHT11 là cảm biến đo nhiệt độ độ ẩm phổ biến do giá thành rẻ và tính ứng dụng cao.
- Sử dụng chuẩn giao tiếp one-wire (sử dụng 1 dây data để truyền dữ liệu) vì vậy khi giao tiếp với vi điều khiển cần phải chú ý cài đặt thời gian timer để dữ liệu đọc không bị lỗi.



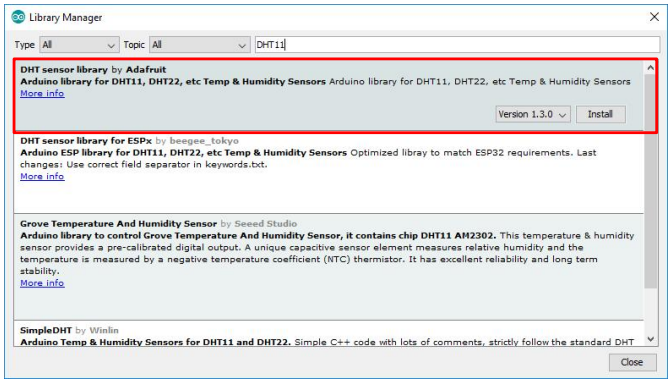
ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Đặc điểm cảm biến DHT11
  - Điện áp hoạt động: 3-5VDC.
  - **Dải nhiệt độ đo: 0-50°C. Sai số ±2°C.**
  - **Dải độ ẩm đo: 20-80%. Sai số 5%.**
  - Tần số lấy mẫu: 1Hz(mỗi giây một lần).
  - Để dữ liệu chính xác thì khoảng cách dây từ vi điều khiển đến cảm biến không quá 20m.
- Cảm biến DHT22
  - **Dải nhiệt độ đo: -40 → +80°C. Sai số ±0.5°C.**
  - **Dải độ ẩm đo: 0-100%. Sai số 2-5%.**
- Thư viện DHT11 cho Arduino
  - Link: https://github.com/adafruit/DHT-sensor-library
  - Tạo thư mục DHT chứa DHT.cpp và DHT.h trong thư mục cài Arduino/libraries



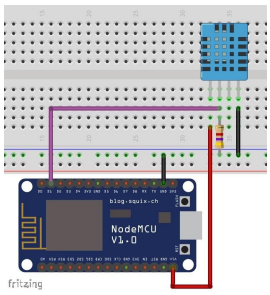
ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Hoặc vào Tools – Library Manager – Gõ DHT 11 – Chọn DHT Sensor library by Adafruit – Install



ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Nối dây giữa ESP8266 và DHT 11
- Linh kiện gồm
  - 1x Breadboard
  - 1x ESP8266, 1x R = 4.7KΩ
  - Dây cắm
  - 1x DHT11
    - Nếu dùng DHT11 4 chân thì cắm như sơ đồ bên (Chân 1: VCC, chân 2: Data, Chân 3: không cắm, Chân 4: GND)
    - Nếu dùng bo DHT có 3 chân thì Chân 1: VCC, chân 2: Data, Chân 3: GND



ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Chạy code sau trên Arduino IDE

```
#include <ESP8266WiFi.h>
#include "DHT.h"

#define DHTTYPE DHT11   // DHT 11

// Thay bằng tên Wifi và địa chỉ truy cập Wifi
const char* ssid = "YOUR_NETWORK_NAME";
const char* password = "YOUR_NETWORK_PASSWORD";

// Web Server ở port 80
WiFiServer server(80);

// Cảm biến DHT
const int DHTPin = 5; // Chân data DHT11 nối vào GPIO5
// Khởi tạo DHT
DHT dht(DHTPin, DHTTYPE);

static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];
```

```
// Chỉ chạy 1 lần khi khởi động
void setup() {
  // Khởi tạo Serial port
  Serial.begin(115200);
  delay(10);

  dht.begin();

  // Kết nối vào mạng WiFi
  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");

  // Bắt đầu web server
  server.begin();
  Serial.println("Web server running. Waiting for the ESP IP...");
  delay(10000);

  // Xuất địa chỉ IP của ESP8266
  Serial.println(WiFi.localIP());
}
```

ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Chạy code sau trên Arduino IDE

```
void loop() {
// Listenning for new clients
WiFiClient client = server.available();
if (client) {
 Serial.println("New client");
 // bolean to locate when the http request ends
  boolean blank_line = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      if (c == '\n' && blank_line) {
// Thời gian cảm biến đọc dữ liệu có thể mất 2s
       float h = dht.readHumidity();
// Đọc nhiệt độ giai Celsius (mặc định)
       float t = dht.readTemperature();
// Đọc nhiệt độ giai Fahrenheit (isFahrenheit = true)
       float f = dht.readTemperature(true);
// Kiểm tra đọc bị lỗi không
     if (isnan(h) || isnan(t) || isnan(f)) {
     Serial.println("Failed to read from DHT sensor!");
     strcpy(celsiusTemp,"Failed");
     strcpy(fahrenheitTemp, "Failed");
     strcpy(humidityTemp, "Failed");
    }
```

```
else {
// Tính nhiệt độ Celsius + Fahrenheit và độ ẩm
        float hic = dht.computeHeatIndex(t, h, false);
        dtostrf(hic, 6, 2, celsiusTemp);
        float hif = dht.computeHeatIndex(f, h);
        dtostrf(hif, 6, 2, fahrenheitTemp);
        dtostrf(h, 6, 2, humidityTemp);
// Dòng lệnh sau để xuất ra monitor cosole để debug
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t Temperature: ");
        Serial.print(t);
        Serial.print(" *C ");
        Serial.print(f);
        Serial.print(" *F\t Heat index: ");
        Serial.print(hic);
        Serial.print(" *C ");
        Serial.print(hif);
        Serial.print(" *F");
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t Temperature: ");
        Serial.print(t);
        Serial.print(" *C ");
        Serial.print(f);
        Serial.print(" *F\t Heat index: ");
        Serial.print(hic);
        Serial.print(" *C ");
        Serial.print(hif);
        Serial.println(" *F");
      }
```


ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Chạy code sau trên Arduino IDE

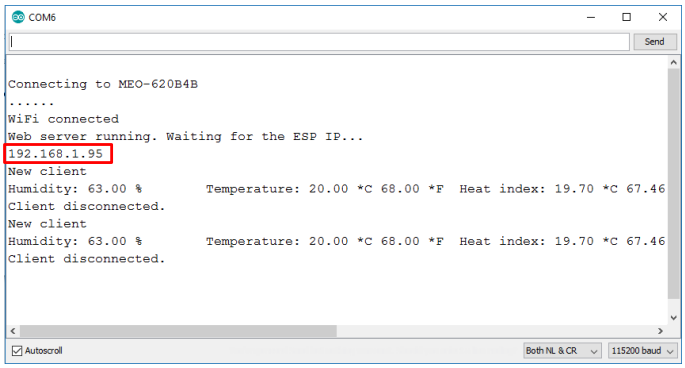
```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println();
// Trang Web hiển thị giá trị nhiệt độ, độ ẩm
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head></head><body><h1>ESP8266 -
Temperature and Humidity</h1><h3>Temperature in Celsius:
");
client.println(celsiusTemp);
client.println("*C</h3><h3>Temperature in Fahrenheit:
");
client.println(fahrenheitTemp);
client.println("*F</h3><h3>Humidity: ");
client.println(humidityTemp);
client.println("%</h3><h3>");
client.println("</body></html>");
break;
  }
```

```
if (c == '\n') {
// when starts reading a new line
  blank_line = true;
        }
else if (c != '\r') {
 // when finds a character on the current line
   blank_line = false;
       }
   }
}
// Đóng kết nối
    delay(1);
    client.stop();
    Serial.println("Client disconnected.");
  }
}
```

CODE CÓ SẴN TRÊN MICROSOFT TEAMS

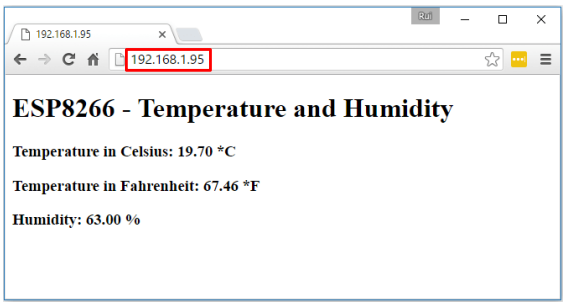
ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Quan sát giá trị nhiệt độ, độ ẩm trên Monitor console của Arduino IDE



ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN TRANG WEB VỚI ESP8266

- Trên Monitor console, địa chỉ IP là **192.168.1.95**
- Đây là địa chỉ IP của ESP8266. Mở trình duyệt web gõ vào địa chỉ trên
- Kết quả nhiệt độ, độ ẩm trên Web server chạy local trên ESP8266

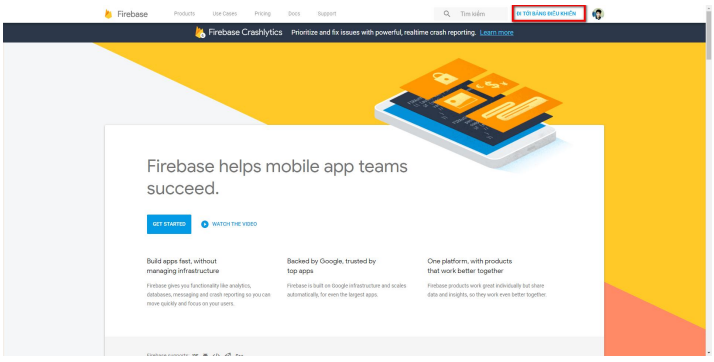


ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.
- Với Firebase có thể tạo ra các ứng dụng IoT là điều khiển thiết bị hoặc hiển thị kết quả theo thời gian thực.
- Firebase cung cấp cho người dùng các chức năng như:
  - **Realtime Database :** Cơ sở dữ liệu đám mấy noSQL data được lưu trữ dưới dạng Json. Dữ liệu được đồng bộ hóa với tất cả client trong thời gian thực.
  - **Firebase Authentication:** Hệ thống xác thực quyền.
  - **Firebase Hosting:** Giúp tạo một ứng dụng nền web với tên miền *firebaseapp.com
- Ở phần real-time database, Firebase cho chúng ta sử dụng miễn phí 10GB data download, 1GB storage và 100 truy xuất đồng thời trong một tháng

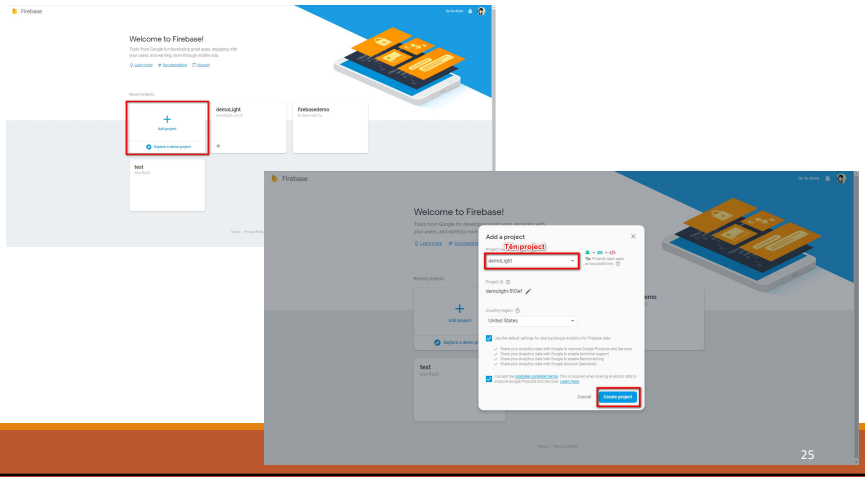
ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Bước 1: Vào https://firebase.google.com/ , đăng nhập với tài khoản Gmail



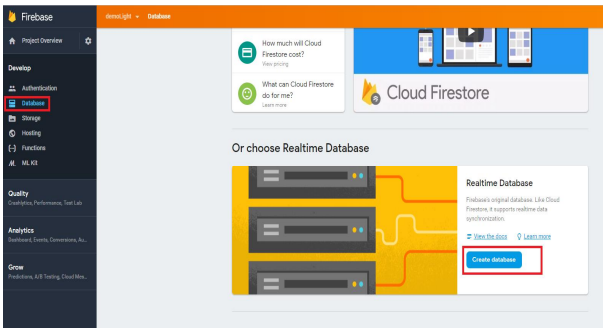
ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

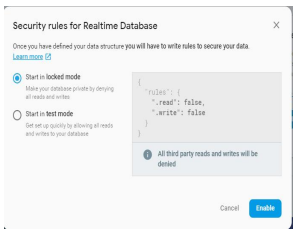
- Bước 2: Tạo 1 project mới, chọn tên Project và Create



ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

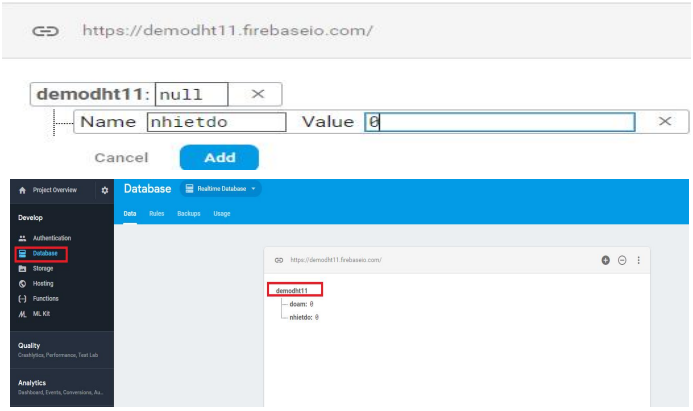
- Bước 3: Tạo database trong project





ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

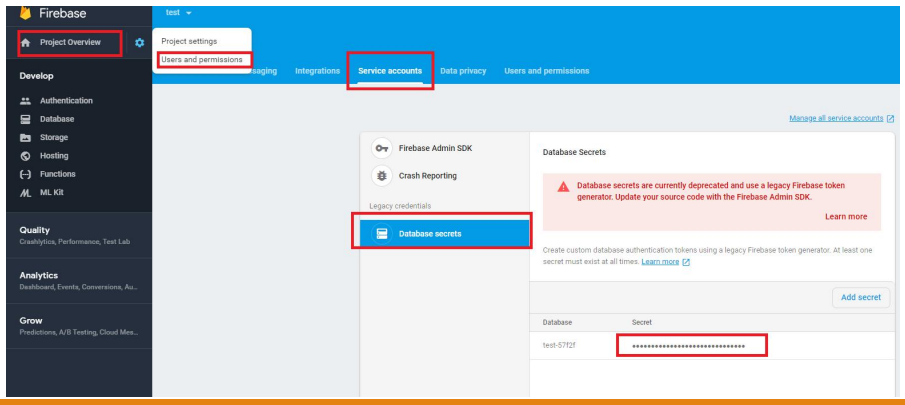
- Bước 4: Tạo dữ liệu trong database



27

ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Bước 5: Lấy token xác thực cho ESP8266



28

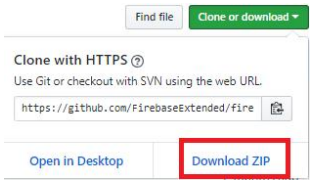
ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Bước 6: Cài đặt thư viện Firebase cho Arduino

+ Thư viện Arduino-firebase:

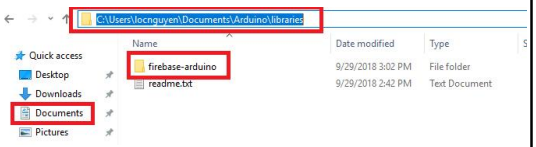
Tải thư viện tại :

https://github.com/FirebaseExtended/firebase-arduino



Giải nén các tất cả các file vào đường dẫn thư viện

Arduino (C:\Users<USER_NAME>\Documents\Arduino\libraries)



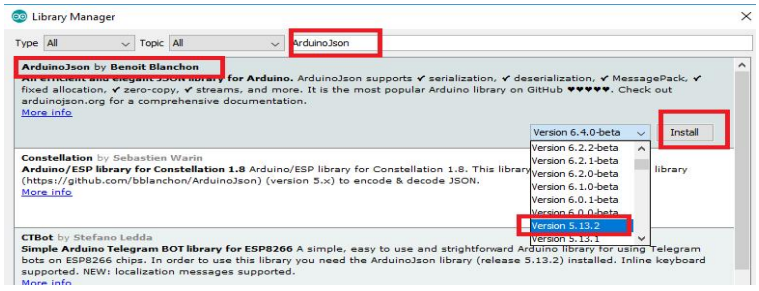
29

ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Bước 6: Cài đặt thư viện Firebase cho Arduino

+ Thư viện Arduino Json (Hỗ trợ cho thư viện Arduino-FireBase):

Cài đặt tương tự như th ư viện DHT. Lưu ý chỉ cài đặt phiên bản ổn định, không cài bản beta



30

ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

- Bước 7: Chạy code sau trên Arduino IDE

```
#include <FirebaseArduino.h>
#include <ESP8266WiFi.h>
#include "DHT.h"

#define DHTTYPE DHT11   // DHT 11

// Thay bằng tên Wifi và địa chỉ truy cập Wifi
const char* ssid = "YOUR_NETWORK_NAME";
const char* password = "YOUR_NETWORK_PASSWORD";
#define FIREBASE_HOST
"<ProjectName>.firebaseio.com"
#define FIREBASE_AUTH  "<TOKEN>"
// Cảm biến DHT
const int DHTPin = 5; // Chân data DHT11 nối
vào GPIO5
// Khởi tạo DHT
DHT dht(DHTPin, DHTTYPE);

static char celsiusTemp[7];
static char fahrenheitTemp[7];
static char humidityTemp[7];
```

```
// Chỉ chạy 1 lần khi khởi động
void setup() {
  // Khởi tạo Serial port
  Serial.begin(115200);
  delay(10);
  dht.begin();

  // Kết nối vào mạng WiFi
  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");
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```


ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

```
void loop() {

// Thời gian cảm biến đọc dữ liệu có thể mất 2s
float h = dht.readHumidity();
// Đọc nhiệt độ giai Celsius (mặc định)
float t = dht.readTemperature();
// Đọc nhiệt độ giai Fahrenheit (isFahrenheit =
true)
float f = dht.readTemperature(true);
// Kiểm tra đọc bị lỗi không
if (isnan(h) || isnan(t) || isnan(f)) {
 Serial.println("Failed to read from DHT
sensor!");
 strcpy(celsiusTemp,"Failed");
 strcpy(fahrenheitTemp, "Failed");
 strcpy(humidityTemp, "Failed");
 }
else {
// Tính nhiệt độ Celsius + Fahrenheit và độ ẩm
float hic = dht.computeHeatIndex(t, h, false);
dtostrf(hic, 6, 2, celsiusTemp);
float hif = dht.computeHeatIndex(f, h);
```

```
dtostrf(hif, 6, 2, fahrenheitTemp);
dtostrf(h, 6, 2, humidityTemp);
// Dòng lệnh sau để xuất ra monitor cosole để
debug
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t Temperature: ");
        Serial.print(t);
        Serial.print(" *C ");
        Serial.print(f);
        Serial.print(" *F\t Heat index: ");
        Serial.print(hic);
        Serial.print(" *C ");
        Serial.print(hif);
        Serial.print(" *F");
        Serial.print("Humidity: ");
        Serial.print(h);
        Serial.print(" %\t Temperature: ");
        Serial.print(t);
        Serial.print(" *C ");
        Serial.print(f);
        Serial.print(" *F\t Heat index: ");
        Serial.print(hic);
        Serial.print(" *C ");
        Serial.print(hif);
        Serial.println(" *F");
```

ESP8266 ĐỌC NHIỆT ĐỘ ĐỘ ẨM TỪ CẢM BIẾN DHT11

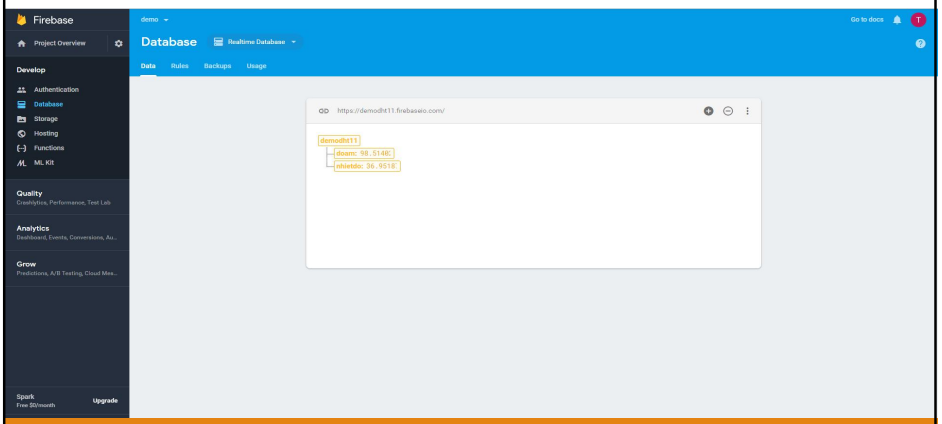
```
// Đưa dữ liệu lên firebase
      Firebase.setFloat("nhietdo",hic);
      if (Firebase.failed())
          Serial.println("Gui du lieu bi loi:");
      Firebase.setFloat("doam", hif);
      if (Firebase.failed())
        Serial.print("Gui du lieu bi loi:");
    }
    delay(2000);
}
```

CODE CÓ SẴN TRÊN MICROSOFT TEAMS

- Vào **General/Files/Bo mạch tự làm và code ESP8266**
  - esp8266_dht11.ino: Code Arduino ESP8266 và DHT11
  - esp8266_dht11_firebase.ino: Code Arduino ESP8266, DHT11 và Firebase

ĐỌC NHIỆT ĐỘ ĐỘ ẨM LÊN FIREBASE VỚI ESP8266

- Bước 8: Quan sát kết quả dữ liệu trên Firebase



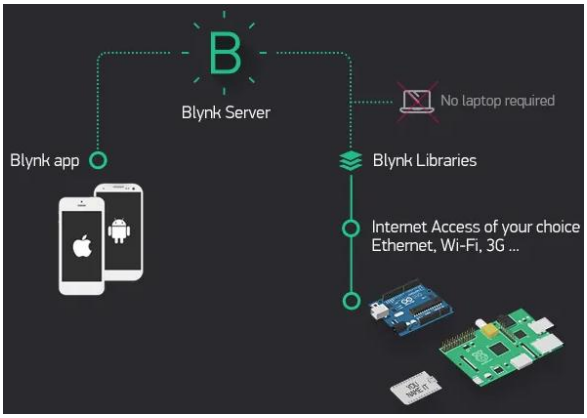
Lập trình IoT với ESP8266 và Blynk

- **Giới thiệu Blynk:** Blynk là một phần mềm mã nguồn mở được thiết kế cho các ứng dụng IoT(Internet of Things). Ứng dụng giúp người dùng điều khiển phần cứng từ xa , có thể hiển thị dữ liệu cảm biến , lưu trữ dữ liệu , biến đổi dữ liệu hoặc làm nhiều việc khác.
- Nền tảng Blynk có ba phần chính:
  - Blynk App – Ứng dụng Blynk cho phép khởi tạo giao diện cho các dự án của mình
  - Blynk Server – Chịu trách nhiệm giao tiếp qua lại hai chiều giữa điện thoại và phần cứng. Bạn có thể sử dụng server của Blynk nhưng sẽ bị giới hạn điểm Enegry. Trong các hướng dẫn sau này mình sẽ sử dụng Server riêng của mình! Và bạn cũng có thể sử dụng nó
  - Blynk Library – Thư viện chứa các nền tảng phổ biến , giúp việc giao tiếp phần cứng với Server dễ dàng hơn

35

Lập trình IoT với ESP8266 và Blynk

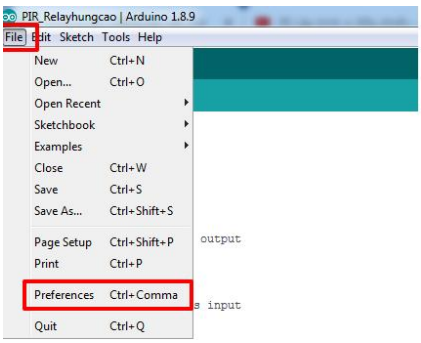
- Giới thiệu Blynk:



36

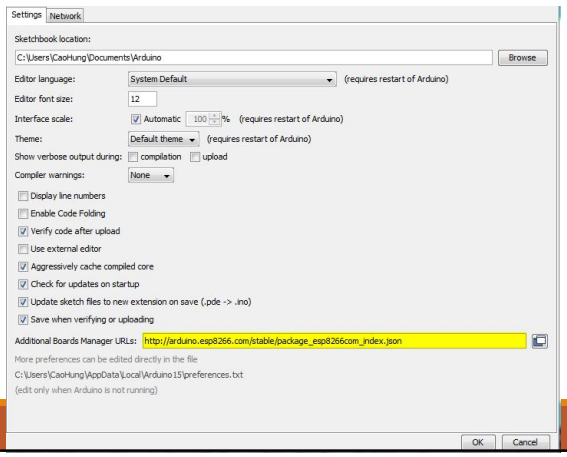
Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn cài đặt NodeMCU ESP8266 WIFI:** Khởi động Arduino IDE, click vào **File** trên thanh công cụ chọn **Preferences.**



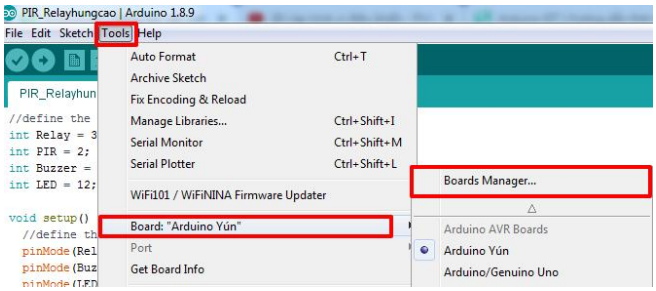
Lập trình IoT với ESP8266 và Blynk

- **Chèn đường link sau vào:**
  http://arduino.esp8266.com/stable/package_esp8266com_index.json



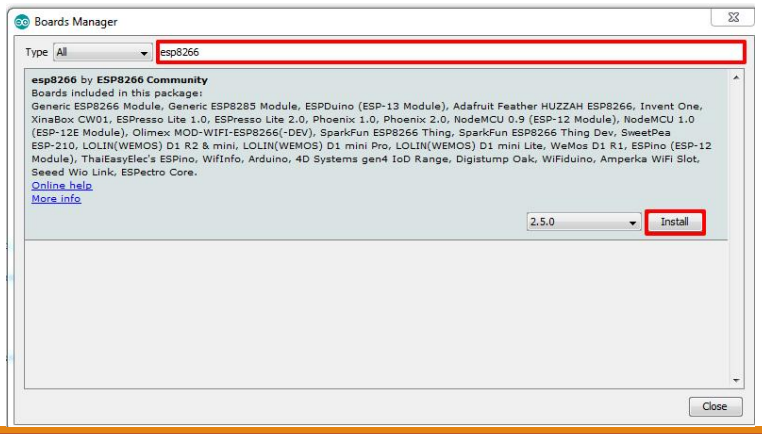
Lập trình IoT với ESP8266 và Blynk

- Vào **Tools → Board → Boards Manager**



Lập trình IoT với ESP8266 và Blynk

- Trong ô search gõ esp8266, rồi nhấn Install



Lập trình IoT với ESP8266 và Blynk

- **Cài thư viện Blynk cho Arduino IDE.**
- Vào **Sketch → Manage Libraries:** Gõ Blynk rồi nhấn Install

- Hoặc vào **Sketch → Include Library → Add. ZIP Library**, tải file ZIP từ Github hoặc từ link sau: https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk_Release_v0.6.1.zip

41

Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn bật/tắt LED trên Blynk**
- Tải ứng dụng Blynk trên thiết bị di động:

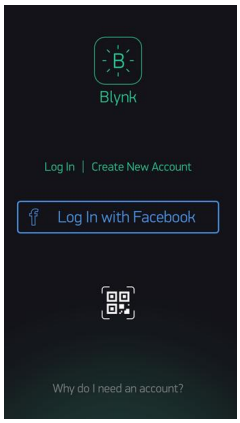
GET IT ON Google Play — https://play.google.com/store/apps/details?id=cc.blynk&hl=en_US

Download on the App Store — http://itunes.apple.com/us/app/blynk-iot-for-arduino-esp32/id808760481?mt=8

42

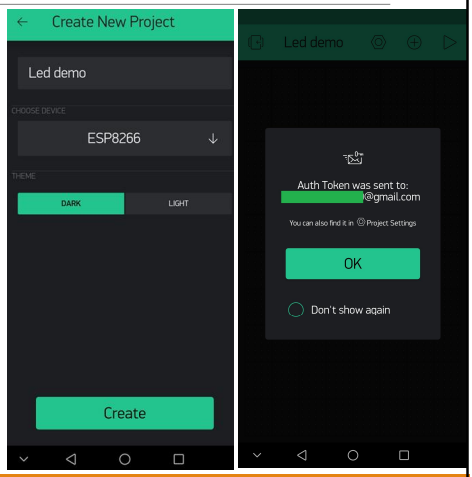
Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn bật/tắt LED trên Blynk**
- Tạo tài khoản Blynk: Tạo tài khoản mới trên Blynk hoặc đăng ký với tài khoản Facebook.
- Để đăng ký mới chọn Create New Account
- Điền địa chỉ Email và Password rồi nhấn Sign Up



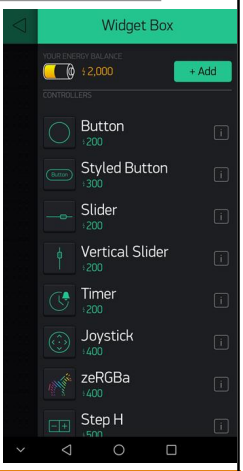
Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn bật/tắt LED trên Blynk**
- Tạo một Project mới: chọn Create New Project (hoặc nhấn + trên thanh công cụ) → Clock vào Create để hoàn tất tạo Project mới.
- Blynk sẽ cấp cho các bạn một **mã Token**, mã này sẽ gửi trực tiếp vào Email đã đăng ký.
- Mã Token này dùng để chèn vào Code Example của Blynk.



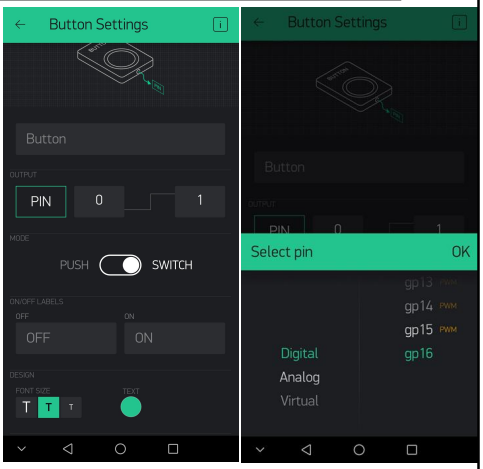
Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn bật/tắt LED trên Blynk**
- Để Bật/Tắt Led chúng ta cần có một nút bấm để điều khiển. **Vào Widget Box (+) > Click vào Button** để lấy nút nhấn ra.
- Click vào Button để tiến hành cài đặt thông số.
- Ở mục Button: Đặt tên cho nút nhấn.



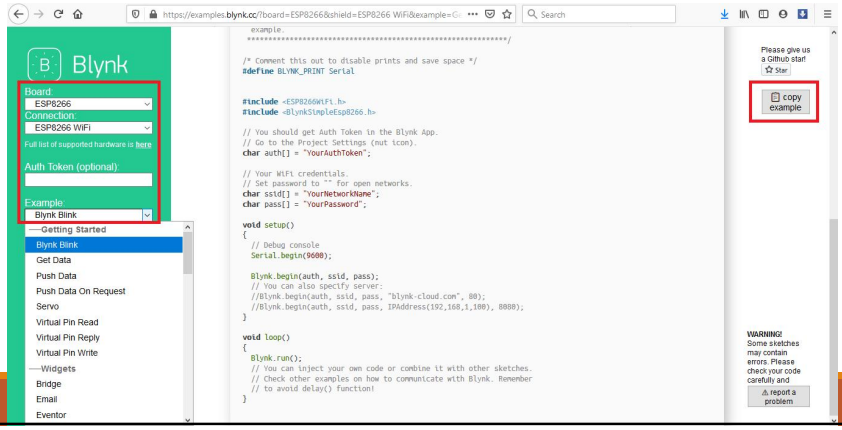
Lập trình IoT với ESP8266 và Blynk

- **Hướng dẫn bật/tắt LED trên Blynk**
- Click vào Button để tiến hành cài đặt thông số.
  - OUTPUT: Cấu hình cho PIN cần kết nối, ở đây mình chọn Digital > gp16 (chân này có LED được tích hợp trên board mạch sẵn).
  - MODE: có 2 chế độ PUSH (nhấn thả) và SWITCH (nhấn giữ).
  - ON/OFF LABELS: Thay đổi chế độ hiển thị cho nút nhấn (Vd: Các bạn không thích để ON/OFF có thể thay bằng BẬT/TẮT hoặc một cái tên nào mà bạn muốn).
  - DESIGN: Ở phần này các bạn có thể điều chỉnh màu sắc của nút nhấn.



Lập trình IoT với ESP8266 và Blynk

- **Nạp code vào NodeMCU ESP8266 Wifi**
- Vào https://examples.blynk.cc (công cụ tạo code cho Blynk trên Web)



Lập trình IoT với ESP8266 và Blynk

- **Nạp code vào NodeMCU ESP8266 Wifi**
- Chú ý cấu hình ở ô bên trái:
  - Chọn Board: ESP8266, ESP32, NodeMCU, v.v.
  - Hình thức kết nối (Wifi, Ethernet, Bluetooth, Cellular, v.v.)
  - Auth. Token là mã đã gửi đến Email đã đăng ký
  - Example: Chọn các ví dụ muốn được tạo ra tự động
- Sau khi cấu hình xong, ở ô bên phải chọn copy example để chép code từ trên Web bỏ vào Arduino IDE.

Lập trình IoT với ESP8266 và Blynk

- **Nạp code vào NodeMCU ESP8266 Wifi**
- Trong code chú ý:

```
char auth[] = "YourAuthToken";
// Mã Authentication Code
char ssid[] = "YourNetworkName";
// Tên mạng Wifi
char pass[] = "YourPassword";
// Password truy cập Wifi
```

- Chọn Board: **NodeMCU 1.0 (ESP-12E Module)**.
- Chọn cổng COM → Upload code

```
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "c2a162bafaf744009a19df8063a2abee";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```


Lập trình IoT với ESP8266 và Blynk

- **Nạp code vào NodeMCU ESP8266 Wifi**
- Link youtube: https://youtu.be/_5u1NXWFvQc
- Trong quá trình Upload mà bị báo lỗi #include <BlynkSimpleEsp8266.h> thì chúng ta khắc phục như sau:
- Vào đường dẫn:

C:\Users\tên_user_trên_máy\Documents\Arduino\libraries\Blynk_Release_v0.6.1\libraries

Copy toàn bộ thư viện vào đường dẫn sau:

C:\Users\ tên_user_trên_máy\Documents\Arduino\libraries

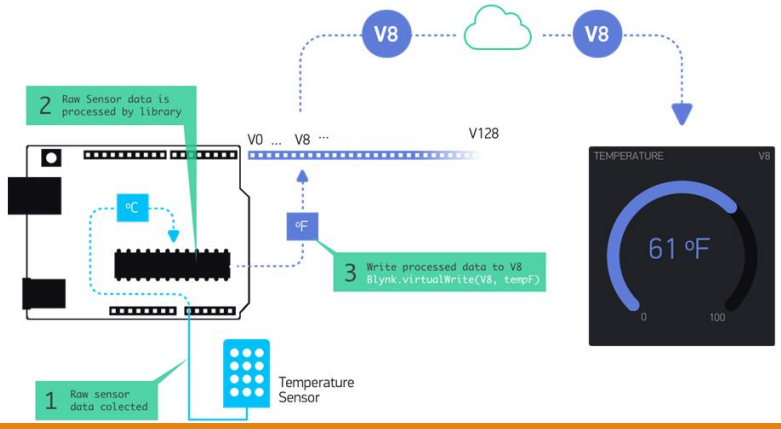
Có thể do Arduino IDE không đọc được thư viện.

Lập trình IoT với ESP8266 và Blynk

- Ứng dụng giám sát nhiệt độ, độ ẩm với NodeMCU ESP8266 và ứng dụng Blynk
- Sử dụng cảm biến DHT11 đo nhiệt độ và độ ẩm
- Link: https://youtu.be/HSwnCpXzA8A

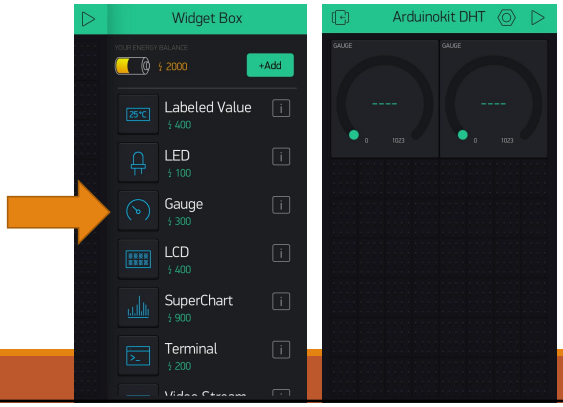
Lập trình IoT với ESP8266 và Blynk

- Sơ đồ khối hệ thống truyền dữ liệu lên Blynk:



Lập trình IoT với ESP8266 và Blynk

- **Cấu hình các thông số cảm biến DHT11 trên App Blynk**
- Để hiển thị các dữ liệu đo chúng ta vào Widget Box (+) →Click vào GAUGE để lấy các thang đo dữ liệu ra.

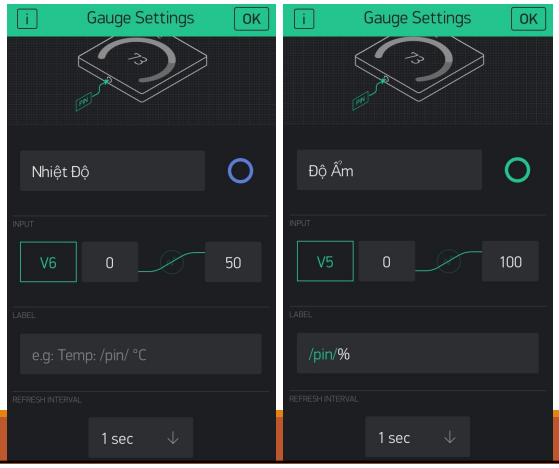


Lập trình IoT với ESP8266 và Blynk

- **Cài đặt các thông số chi tiết nhiệt độ, độ ẩm**
- Click vào GAUGE để tiến hành cài đặt thông số.
- **Cài đặt thông số nhiệt độ**
  - Ở mục GAUGE 1: Đặt tên cho thang đo nhiệt độ (temperature).
  - INPUT: Cấu hình cho PIN cần kết nối, ở đây chọn Virtual → V6 (0 - 1023) đổi thành (0 - 50).
  - LABELS: e.g: Temp:/pin/độ C.
  - REFRESH INTERVAL: Vào Push chọn tốc độ đọc cảm biến nhiệt độ (sec).
- **Cài đặt thông số độ ẩm**
  - Ở mục GAUGE 2 : Đặt tên cho thang đo độ ẩm (humidity).
  - INPUT: Cấu hình cho PIN cần kết nối, ở đây mình chọn Virtual → V5 (0 - 1023) đổi thành (0 - 100).
  - LABELS: e.g: Temp:/pin/% (Độ ẩm đơn vị là %).
  - REFRESH INTERVAL: Vào Push chọn tốc độ đọc cảm biến độ ẩm (sec).

Lập trình IoT với ESP8266 và Blynk

- **Cài đặt các thông số chi tiết nhiệt độ, độ ẩm**

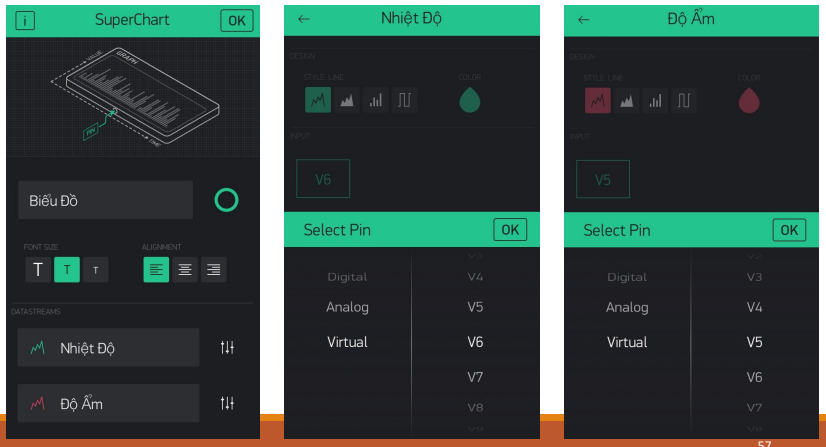


Lập trình IoT với ESP8266 và Blynk

- **Biểu đồ nhiệt độ, độ ẩm DHT11**
- Tiếp tục vào Widget Box (+) → Click vào SUPPERCHART để **biểu đồ hiển thị nhiệt độ, độ ẩm** ra.
- Ở mục SuperChart : Đặt tên biểu đồ.
- DATASTREAMS: Đặt tên cho từng biểu đồ Nhiệt độ và Độ Ẩm.
- Virtual là các chân dùng để trao đổi dữ liệu giữa phần cứng với App Blynk ở đây mình chọn (V6 cho Nhiệt Độ), (V5 cho Độ Ẩm).

Lập trình IoT với ESP8266 và Blynk

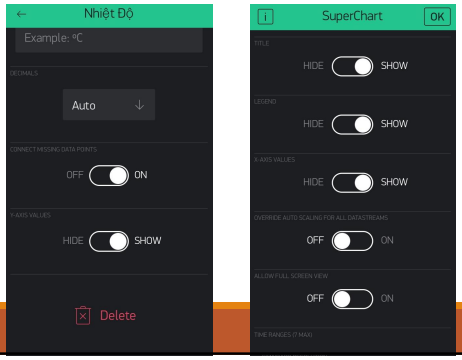
- **Biểu đồ nhiệt độ, độ ẩm DHT11**



57

Lập trình IoT với ESP8266 và Blynk

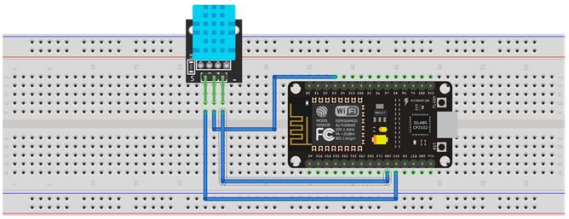
- **Biểu đồ nhiệt độ, độ ẩm DHT11**
- Để hiển thị vectơ các mức giá trị đo trên biểu đồ, vào mục Y-AXIS VALUES chuyển về nút SHOW (hiển thị) cho cả Nhiệt Độ và Độ Ẩm.
- Để hiển thị thời gian cập nhật vào mục X-AXIS VALUES chuyển về nút SHOW.



58

Lập trình IoT với ESP8266 và Blynk

- Sơ đồ kết nối DHT11 và NodeMCU ESP8266 và kết quả trên Blynk



NodeMCU ESP8266	Cảm biến nhiệt độ, độ ẩm DHT11
3V3	VCC (5V)
GND	GND
D4	OUT

Lập trình IoT với ESP8266 và Blynk

- Code (Arduino IDE)
- Các thư viện: DHT.h, SimpleTimer.h, BlynkSimpleEsp8266.h

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHT.h"
#include <SimpleTimer.h>
#define DHTTYPE DHT11
#define dht_dpin 2          //Chân GPIO2 tức D4 đọc data từ DHT11
DHT dht(dht_dpin, DHTTYPE);
SimpleTimer timer;
char auth[] = "49d330c27974424a8d4e897eec57dfd1";
char ssid[] = "Wifi SSID";          // Tên Wifi
char pass[] = "Wifi Password";   // Passwor Wifi
float t;
float h;

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(2000, sendUptime);
}

void sendUptime()
{
  float h = dht.readHumidity();        // Đọc độ ẩm từ DHT11
  float t = dht.readTemperature();   // Đọc nhiệt độ từ DHT11
  // Hiển thị lên Serial Monitor
  Serial.println("Humidity and temperature\n\n");
  Serial.print("Current humidity = ");
  Serial.print(h);
  Serial.print("%  ");
  Serial.print("temperature = ");
  Serial.print(t);
  Blynk.virtualWrite(V6, t);   // Trao đổi dữ liệu nhiệt độ với Blynk
  Blynk.virtualWrite(V5, h);   // Trao đổi dữ liệu độ ẩm với Blynk
}

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