

CHƯƠNG 3

PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS

Chương này cung cấp cho sinh viên kiến thức:

- Các phát triển ứng dụng IoT vào thực tế, xây dựng giải pháp ứng dụng
- Cách điều khiển thiết bị từ xa qua Bluetooth, Wifi, Internet, GSM,...

TT	Chuẩn đầu ra của chương	CĐR HP
1	Phân tích hệ thống IoT thực tế	5
2	Ứng dụng điều khiển thiết bị từ xa qua Bluetooth, GSM	6
3	Ứng dụng điều khiển thiết bị từ xa qua Internet, Wifi	8

3.1 SỬ DỤNG ARDUINO IDE LẬP TRÌNH ESP8266 NODEMCU

Song song với các phiên bản Arduino khác nhau như UNO R3, Tiny, ESP8266 NodeMCU là một trong những mạch tích hợp phổ biến trong việc phát triển các dự án IoT. Với ESP8266 NodeMCU, ngoài việc sử dụng ngôn ngữ lập trình Lua ta còn có thể lập trình sử dụng ngôn ngữ C/C++ thông qua Arduino IDE.

Ưu điểm của mạch ESP8266 NodeMCU là module wifi được tích hợp sẵn và sử dụng một vi điều khiển mạnh mẽ hơn so với Arduino nguyên thủy.

Điều khiển không dây giúp giảm bớt sự phức tạp của lắp đặt và sử dụng trong các xưởng sản xuất, các nhà máy hỗ trợ giảm hao phí trong vận hành vì các vấn đề mà hệ thống có dây gây ra.

Với ESP8266 NodeMCU, ngoài việc sử dụng ngôn ngữ lập trình Lua ta còn có thể lập trình sử dụng ngôn ngữ C/C++ thông qua Arduino IDE. Bài viết này sẽ hướng dẫn việc sử dụng Arduino IDE để tích hợp thư viện hỗ trợ cho việc lập trình ESP8266 NodeMCU cũng như cách biên dịch, nạp code và kiểm tra.

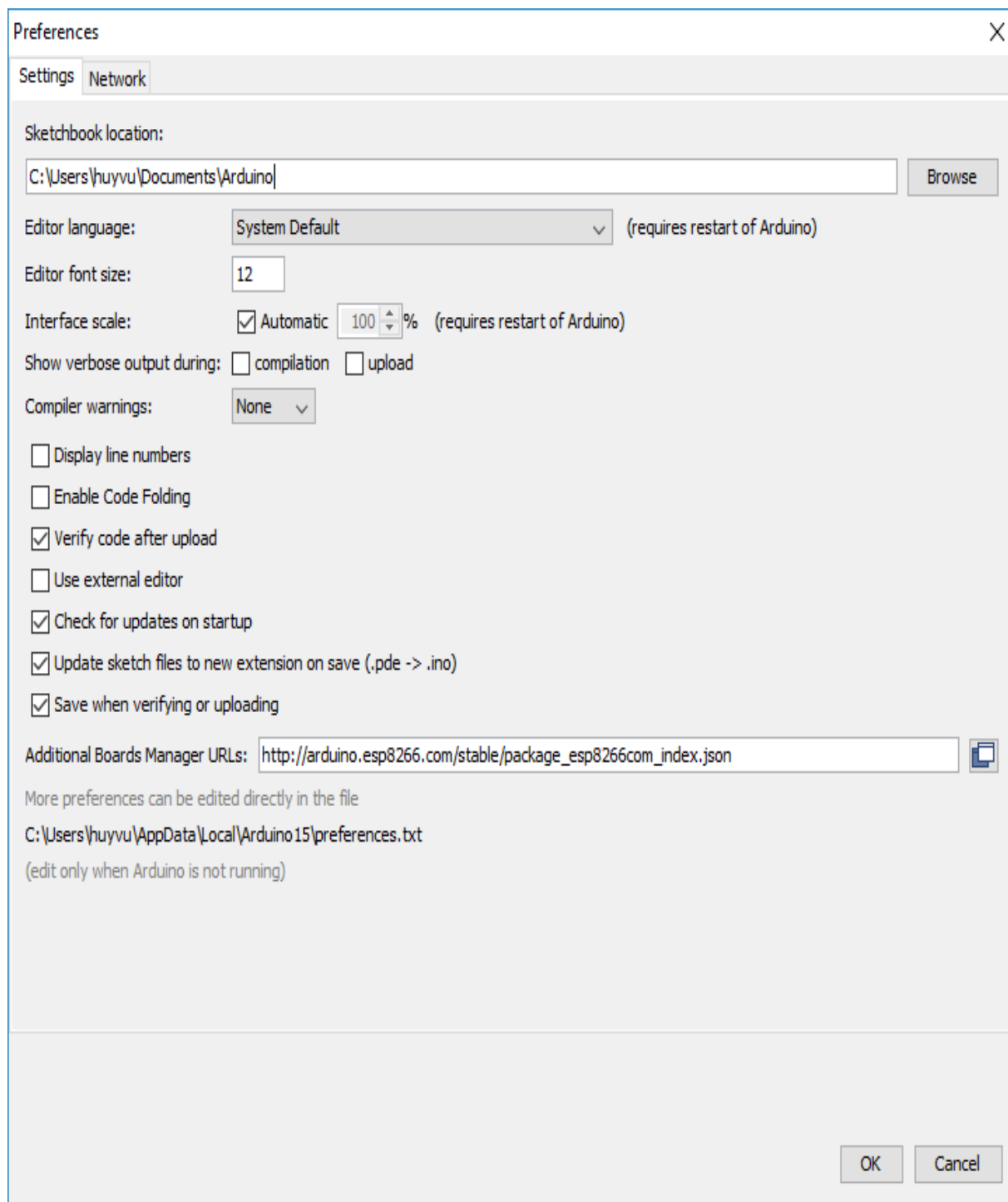
3.1.1 Tích hợp thư viện hỗ trợ ESP8266 NodeMCU

Để tích hợp thư viện hỗ trợ cho việc lập trình mạch ESP8266 NodeMCU. Ta lần lượt thực hiện các bước sau:

Bước 1: Thêm đường dẫn để tải các package cho NodeMCU vào Arduino IDE.

Khởi động Arduino IDE, từ màn hình chính chọn **File** → **Preferences**. Ta thêm đường dẫn bên dưới vào mục **Addition Boards Manager URLs**.

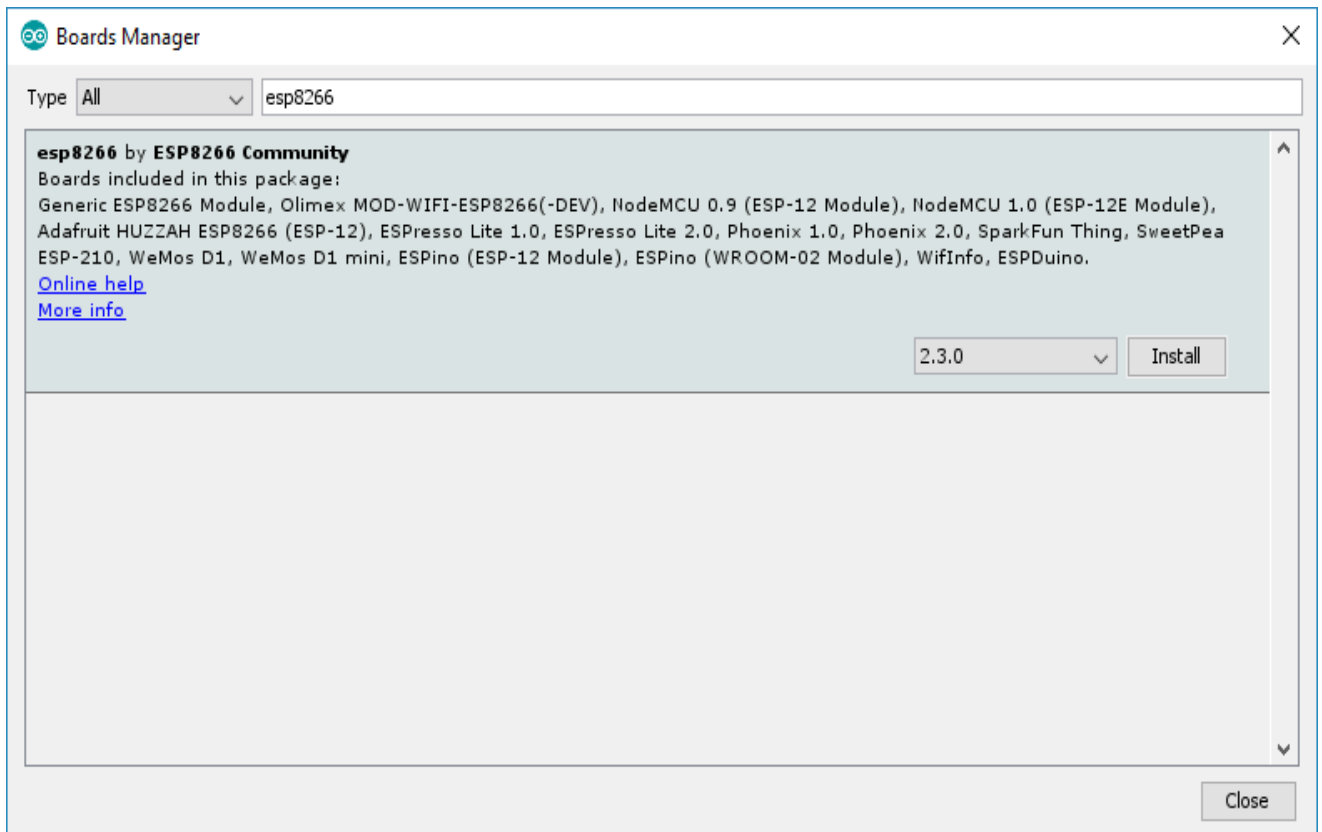
-
1. http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json
-



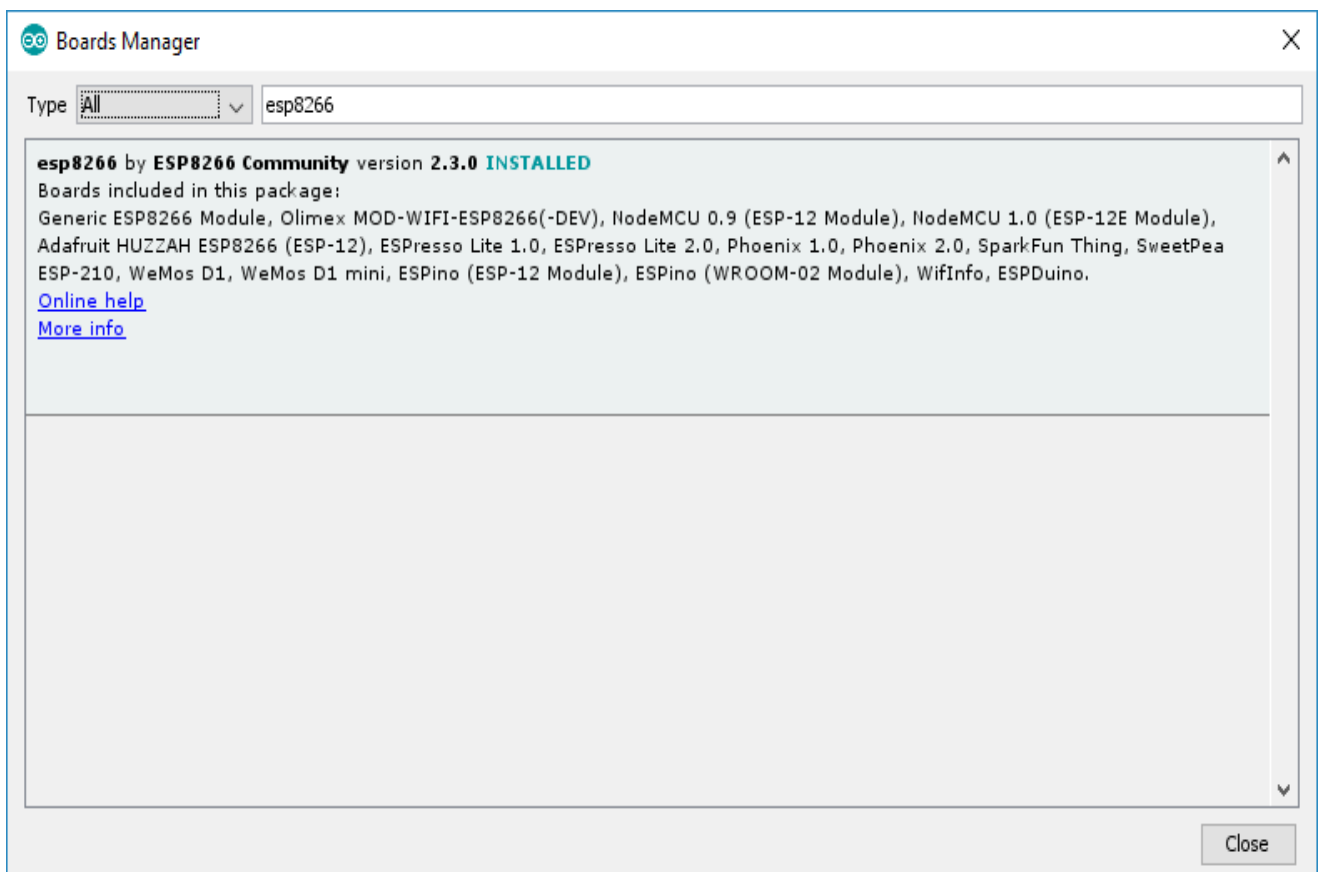
Chọn **OK** để xác nhận việc thêm vào.

Bước 2: Tải thư viện hỗ trợ

Từ giao diện chính của Arduino IDE, chọn **Tools** → **Board** → **Board Managers ...** Tại thanh tìm kiếm của hộp thoại Board Managers ta nhập vào *esp8266*, chọn **Install** để tiến hành tải, cài đặt thư viện.



Cài đặt thành công, giao diện của Board Managers sẽ trở nên như hình bên. Đến đây ta đã hoàn tất việc cài đặt thư viện.



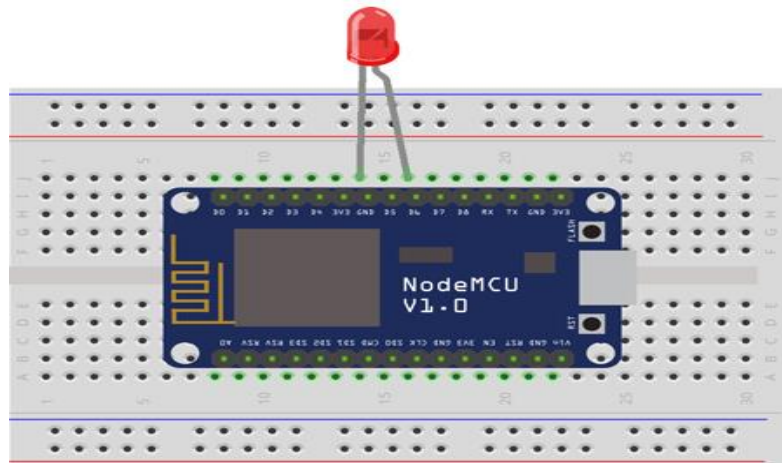
3.1.2 Lập trình cho ESP8266 NodeMCU

Do đây là một board Arduino-compatible, cấu trúc của một chương trình dành cho mạch này sẽ tuân theo cấu trúc của một chương trình viết cho mạch Arduino bao gồm có 2 phần chính:

- Hàm `setup()`: được gọi một lần duy nhất khi mạch được khởi động.
- Hàm `loop()`: được gọi lặp lại trong suốt quá trình hoạt động của mạch.

Bước đầu làm quen, ta sẽ viết một chương trình cho ESP điều khiển một đèn LED nhấp nháy theo chu kì 1 giây. Linh kiện cần chuẩn bị bao gồm 1 mạch ESP8266 NodeMCU và 1 đèn LED 5mm.

Sơ đồ mạch:



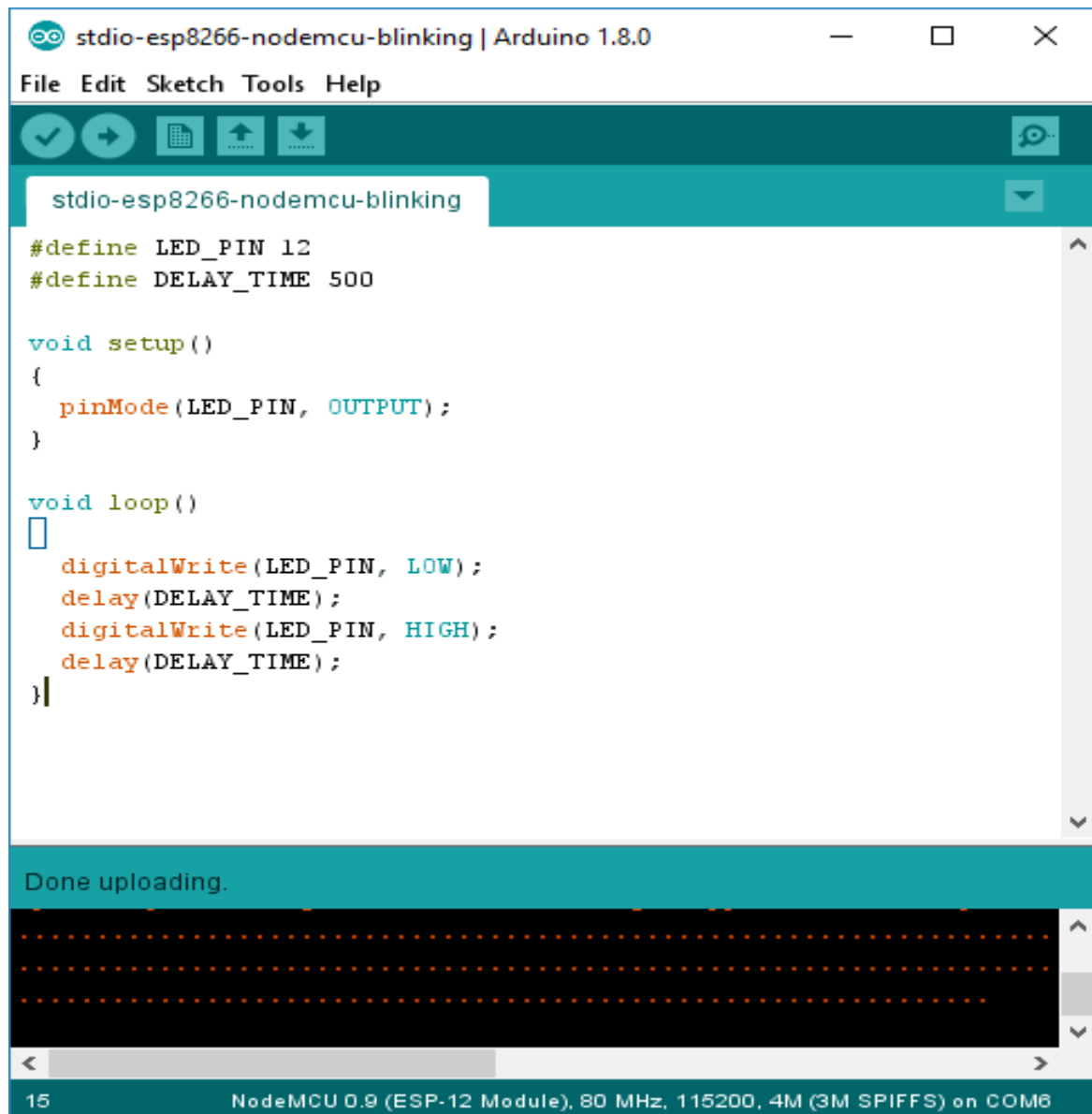
Lập trình:

Đoạn code sau minh họa việc điều khiển đèn LED chớp theo chu kì 1 giây.

```
1.  #define LED_PIN 12
2.  #define DELAY_TIME 500
3.  void setup()
4.  {
5.      pinMode(LED_PIN, OUTPUT);
6.  }
7.  void loop()
8.  {
9.      digitalWrite(LED_PIN, LOW);
10.     delay(DELAY_TIME);
11.     digitalWrite(LED_PIN, HIGH);
12.     delay(DELAY_TIME);
13. }
```

Nạp code:

Thao tác nạp code cho mạch ESP8266 NodeMCU cũng tương tự như nạp cho mạch Arduino thông thường. Tuy nhiên, cần lưu ý phải chọn phiên bản phù hợp với board đang sử dụng bằng menu Tools → Board. Do mạch của tôi là ESP8266 NodeMCU (ESP-12 module) do đó tôi cần chọn **NodeMCU 0.9**.



Sau khi nạp code thành công, ta sẽ thấy đèn LED nhấp nháy theo chu kì định sẵn.

3.2. BẬT TẮT ĐÈN QUA WEB VỚI ESP8266

Hướng dẫn điều khiển đèn học thông qua website với ESP8266. Khác với cách thông thường là tạo một server để điều khiển từ xa thì chúng ta sẽ sử dụng ESP8266 để tạo ra một web server nhỏ trên chip, sao đó tạo ra giao diện web để có thể điều khiển đèn

3.2.1 Phần cứng

Cần chuẩn bị phần cứng như:

- Modul ESP8266v12 hoặc NodeMCU.
- Nguồn 3.3V

- Bộ chuyển đổi USB2UART
- Board test và dây cắm
- Modul Relay 5V-220 hoặc modul tương tự
- Đèn học

Kết nối phần cứng:

Phần nối nguồn và chân để nạp chương trình cho ESP vẫn như các bài trước, ở đây có khác một chút là kết nối chân điều khiển relay với chân GPIO5 và kết nối giữa relay và đèn học

3.2.2 Chương trình

Chúng ta sẽ đi luôn vào chương trình lập trình điều khiển đèn. ESP8266 sẽ nhận yêu cầu từ website, trên giao diện web này sẽ có 2 nút nhấn (ON/OFF) để điều khiển tắt mở đèn.

Đầu tiên vẫn là cấu hình cho đúng tên wifi và passwifi, sau đó là tạo server

```
// Thông số WiFi nhà bạn
const char* ssid = "mang_wifi";
const char* password = "pass_wifi";

// Tạo server
WiFiServer server(80);
```

Khai báo GPIO5 điều khiển relay là Output

```
// Khai báo GPIO5
pinMode(output_pin, OUTPUT);
digitalWrite(output_pin, 0);
```

Sau khi kết nối và khởi tạo server trong vòng lặp loop(), sẽ được dùng để lắng nghe các kết nối và yêu cầu ở port 80.

Khi nhận được dữ liệu thực hiện kiểm tra xem có yêu cầu bật tắt (on/off) gì không ? Nếu có thì thực hiện thay đổi trạng thái của output thông qua lệnh

```
// Kiểm tra yêu cầu là gì
if (req.indexOf("/on") != -1){
    digitalWrite(output_pin, 1);
}
else if (req.indexOf("/off") != -1) {
    digitalWrite(output_pin, 0);
}
```

Tạo giao diện để có thể truy cập vào board thông qua trình duyệt dùng HTML

```
// Chuẩn bị thông tin phản hồi
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
s += "<head>";
s += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">";
```

```
s += "<script src=\"https://code.jquery.com/jquery-2.1.3.min.js\"></script>";  
s += "<link rel=\"stylesheet\" href=\"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css\">";  
s += "</head>";
```

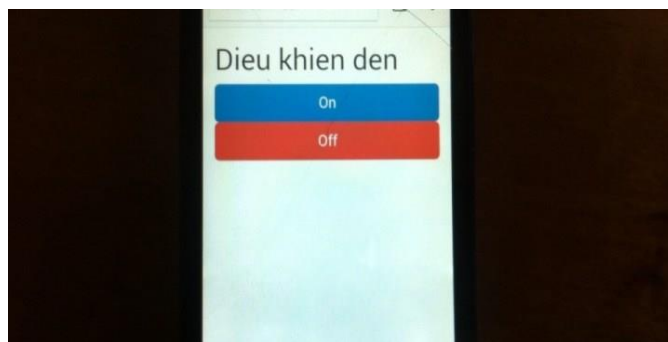
Tạo 2 nút nhấn trên giao diện và một số lệnh JavaScript để kiểm tra khi nào 2 nút được ấn

```
s += "<div class=\"container\">";  
s += "<h1>Dieu khien den</h1>";  
s += "<div class=\"row\">";  
s += "<div class=\"col-md-2\"><input class=\"btn btn-block btn-lg btn-primary\" type=\"button\" value=\"On\" onclick=\"on()\"></div>";  
s += "<div class=\"col-md-2\"><input class=\"btn btn-block btn-lg btn-danger\" type=\"button\" value=\"Off\" onclick=\"off()\"></div>";  
s += "</div></div>";  
s += "<script>function on() {$.get(\"/on\");}</script>";  
s += "<script>function off() {$.get(\"/off\");}</script>";
```

Cuối cùng là biên dịch và nạp chương trình xuống ESP8266.

3.2.3 Kết quả

Giao diện quản lý và điều khiển khi quan sát trên máy tính và điện thoại



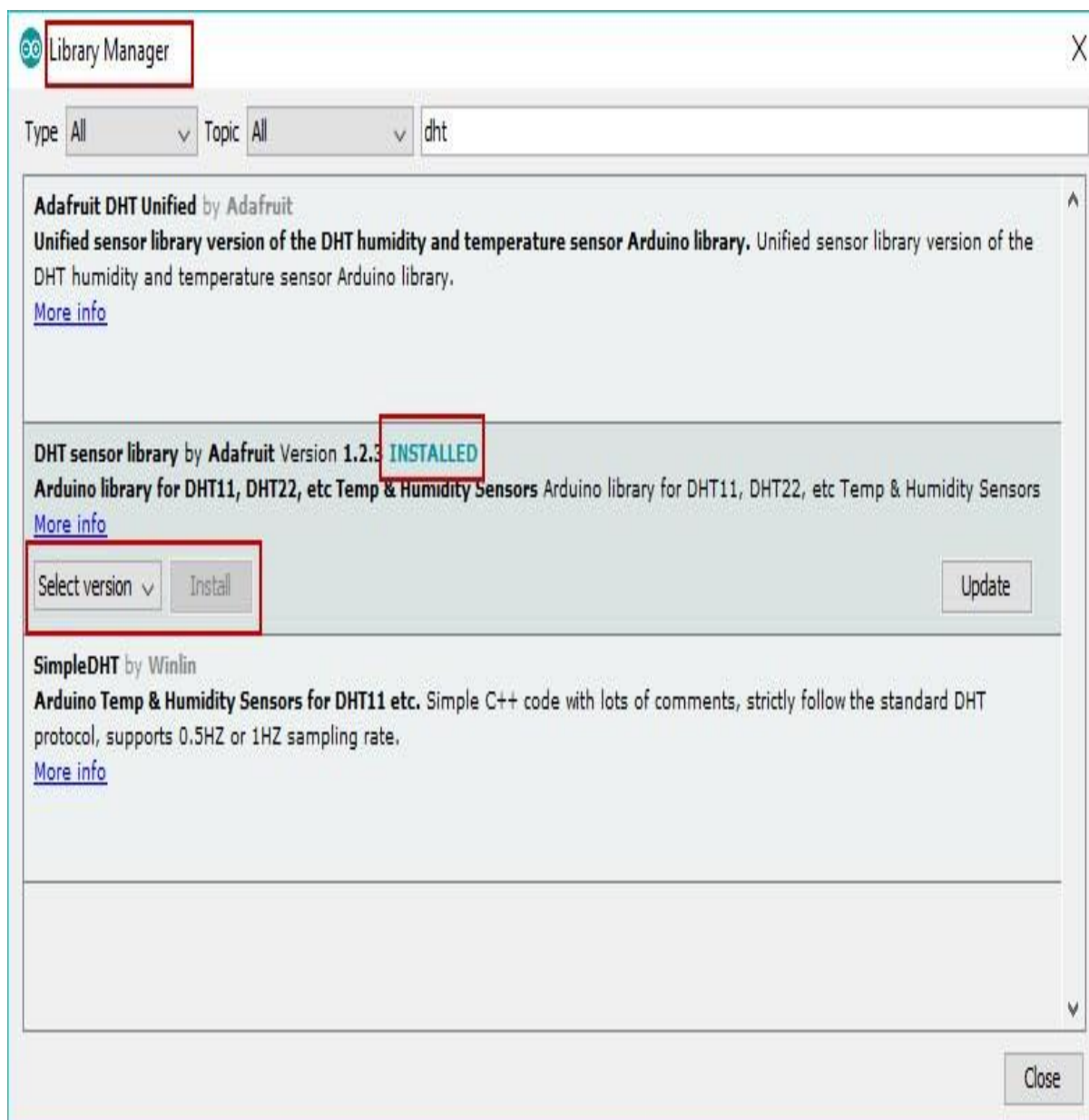
Thế là chúng ta đã có thể điều khiển thiết bị qua web một cách dễ dàng, bạn có thể phát triển thành điều khiển nhiều thiết bị khác ở trong nhà của mình.

3.3 WEBSERVER THU THẬP DỮ LIỆU DHT11 VỚI ESP8266

Ở bài trước chúng ta đã làm quen với việc lấy nội dung trang web thì hôm nay chúng ta sẽ quay trở lại với việc sử dụng ESP8266 đọc dữ liệu nhiệt độ độ ẩm môi trường từ cảm biến DHT11, sau đó sẽ thực hiện lập trình để có thể quan sát được thông tin về nhiệt độ độ ẩm thông qua wifi.

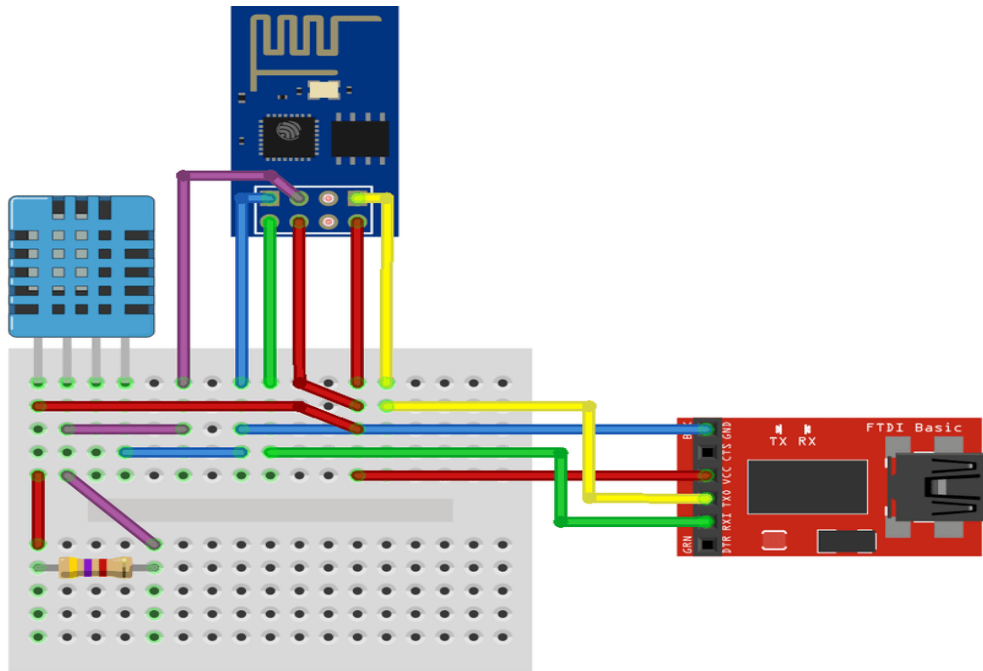
3.3.1 Đọc nhiệt độ độ ẩm với DHT11

Đầu tiên để có thể giao tiếp được với DHT11 thì cần tải thư viện bằng cách chọn Sketch > Include Library > Manage Libraries hoặc bạn có thể tải và cài thủ công thư viện tại đây



Tiếp theo là kết nối theo sơ đồ tương tự như hình với ESP8266 v1 (dùng GPIO2) và ESP8266 v7/v12 (dùng GPIO5).

Lưu ý: Nguồn cấp cho DHT11 có thể dùng 3.3 – 5V.



fritzing

Chương trình kiểm tra nhiệt độ và độ ẩm

```
// Chương trình đọc nhiệt độ, độ ẩm từ cảm biến DHT
// Written by ladyada, public domain
// Chỉnh sửa cho ESP8266 bởi hocARM.org
#include "DHT.h"
#define DHTPIN D1    // Chân DATA nối với chân D1
// Uncomment loại cảm biến bạn sử dụng, nếu DHT11 thì uncomment DHT11 và
// comment DHT22
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

// Kết nối
// DHT    | ESP8266
// -----
// VCC(1)  | 3.3V
// DATA(2) | D3
// NC(3)   | x
// GND(4)  | GND
// Kết nối chân 1 của DHT với 3.3V
// Nối trở 10k giữa chân 1 và chân 2
```

```
// Khởi tạo cảm biến
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Khởi tạo cổng serial baud 115200
  Serial.begin(115200);
  Serial.println("DHTxx test!");
  // Bắt đầu đọc dữ liệu
  dht.begin();
}

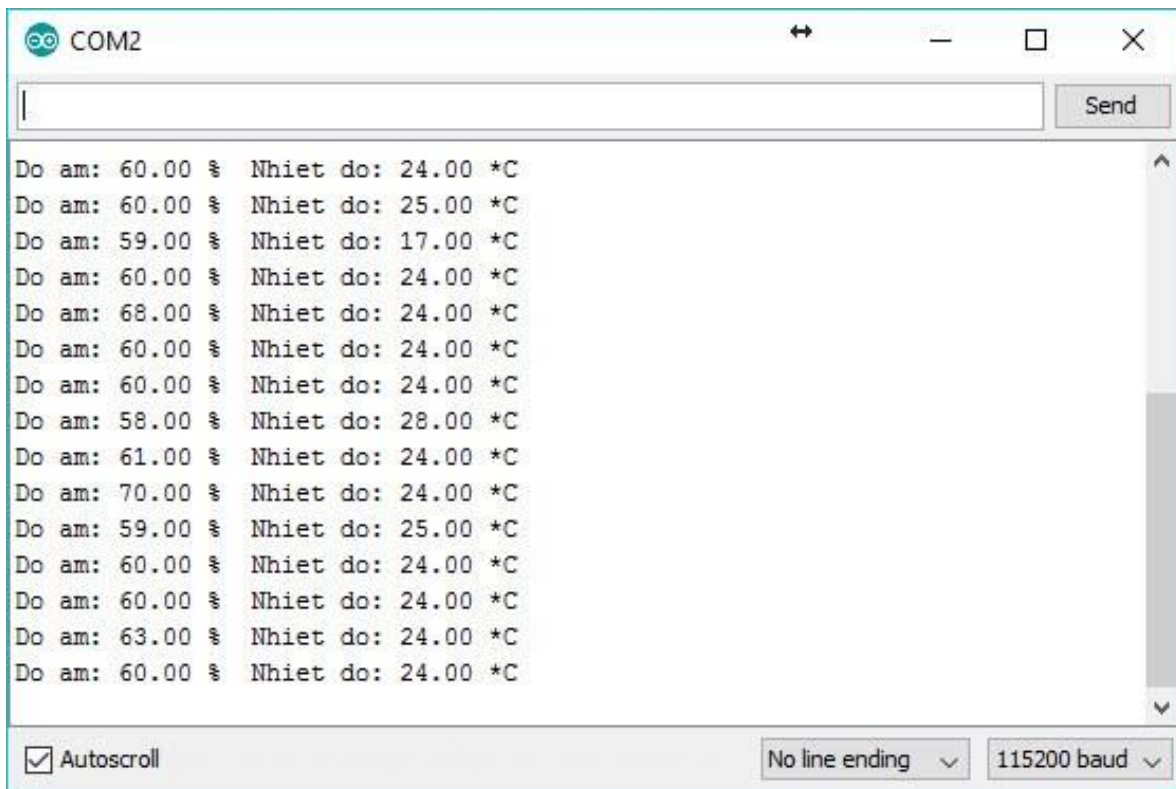
void loop()
{
  // Delay chuyển đổi dữ liệu khoảng 2s
  delay(2000);
  float h = dht.readHumidity();
  // Đọc giá trị nhiệt độ C (mặc định)
  float t = dht.readTemperature();
  // Đọc giá trị nhiệt độ F(isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Kiểm tra quá trình đọc thành công hay không
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // IN thông tin ra màn hình
  Serial.print("Do am: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Nhiệt độ: ");
  Serial.print(t);
  Serial.println(" *C ");

}
```

Sau khi nạp xuống ta quan sát qua terminal sẽ thấy được thông tin như sau:



3.3.2 Theo dõi nhiệt độ độ ẩm qua mạng Wifi

Ta đã đọc được nhiệt độ và độ ẩm rồi thì làm thế nào để gửi nó lên mạng wifi trong nhà ? Có cần phải tạo web gì không ? Thật may mắn là ESP8266 có thể tạo ra một server và web ở trên địa chỉ sẵn có của nó, ta chỉ cần đăng nhập vào địa chỉ (ví dụ 192.168.1.xx) là có thông tin về nhiệt độ, độ ẩm rồi.

Giải thích một số ý chính trong code:

```
// Thông số WiFi nhà bạn
const char* ssid = "mang_wifi";
const char* password = "pass_wifi";
```

Trước hết là cần thay đổi chỗ mang_wifi và pass_wifi thành tên và pass wifi nhà bạn là được. Tiếp tục là tạo một web server với port là 80 bằng lệnh:

```
// Tạo server
WiFiServer server(80);
```

Tạo xong rồi ta thực hiện kết nối ESP8266 vào mạng:

```
// Kết nối tới mạng WiFi
Serial.println();
Serial.println();
Serial.print("Kết nối tới mạng ");
```

```
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("Da ket noi WiFi");
```

Sau đó là khởi động server và in địa chỉ IP ra

```
// Khoi dong server
server.begin();
Serial.println("Khoi dong Server");
// In ra dia chi IP
Serial.println(WiFi.localIP());
```

Trong vòng lặp vô tận chúng ta sẽ thực hiện kiểm tra có client nào kết nối tới không bằng lệnh `WiFiClient client = server.available()`; đồng thời đọc nhiệt độ và độ ẩm

```
// Doc do am
float h = dht.readHumidity();
// Doc nhiet do o do C
float t = dht.readTemperature();
```

Tiếp tục chờ yêu cầu (request) từ client (máy tính hoặc điện thoại khác cùng mạng wifi)

```
// Doc dong dau tien cua yeu cau gui len.
String req = client.readStringUntil('\r');
Serial.println(req);
client.flush();
```

Coi như đã xong bước nhận yêu cầu, giờ là chuẩn bị câu trả lời cho yêu cầu từ client, ở đây ta sẽ tạo ra một trang web HTML sử dụng Bootstrap CSS để tự động reload trang sau mỗi một phút

```
// Chuan bi tao web de phan hoi
String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
s += "<head>";
```

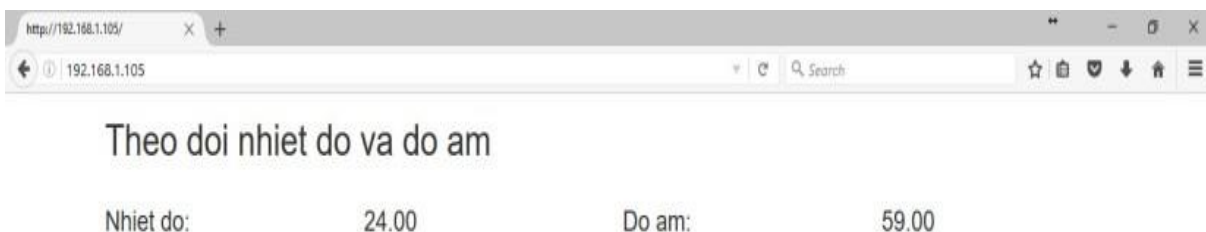
```
s += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">";
s += "<meta http-equiv=\"refresh\" content=\"60\" />";
s += "<script src=\"https://code.jquery.com/jquery-2.1.3.min.js\"></script>";
s += "<link rel=\"stylesheet\" href=\"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css\">";
s += "<style>body{font-size: 24px;} .voffset {margin-top: 30px;}</style>";
s += "</head>";
```

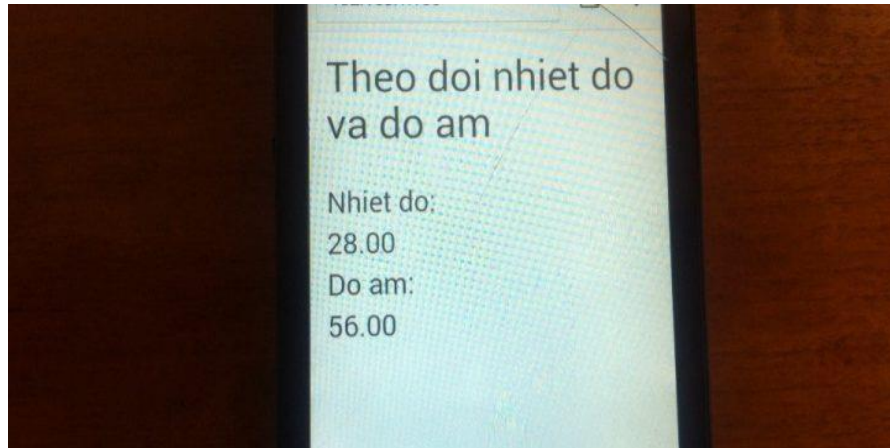
Cuối cùng là hiển thị nhiệt độ và độ ẩm đọc được từ DHT11 lên và đợi cho tới khi client ngắt kết nối với ESP8266

```
s += "<div class=\"container\">";
s += "<h1>Theo doi nhiet do va do am</h1>";
s += "<div class=\"row voffset\">";
s += "<div class=\"col-md-3\">Nhiet do: </div><div class=\"col-md-3\">" + String(t) + "</div>";
s += "<div class=\"col-md-3\">Do am: </div><div class=\"col-md-3\">" + String(h) + "</div>";
s += "</div>";

// Gui phan hoi toi client (o day la giao dien web)
client.print(s);
delay(1);
Serial.println("Client da thoat");
```

Kết quả hoạt động tốt khi theo dõi qua laptop và điện thoại:





Thế mà chúng ta đã thực hiện được giao tiếp với DHT11 để đọc nhiệt độ và độ ẩm, sau đó tạo webserver để gửi dữ liệu đã đọc được khi có yêu cầu từ laptop hoặc điện thoại, đây là cách đơn giản nhất để bạn có thể giám sát được thông tin, chẳng hạn như độ ẩm, nhiệt độ của đất cây trồng chẳng hạn. Cách này vẫn còn bị giới hạn chỉ những client trong cùng một mạng wifi mới có thể quan sát được thông tin.

3.4 ĐIỀU KHIỂN THIẾT BỊ TỪ XA

3.4.1 Giới thiệu kit

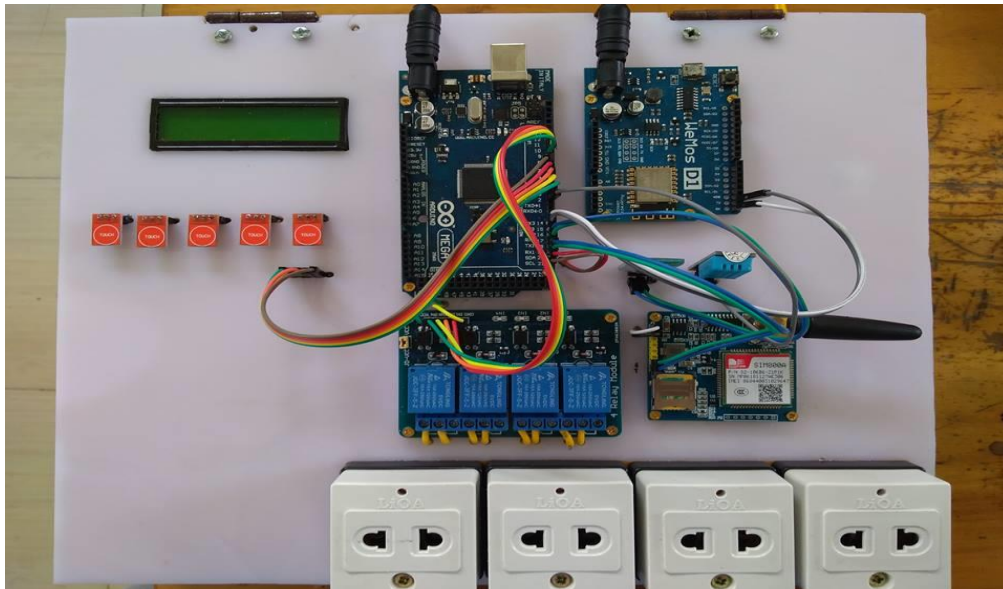


- Kit gồm hầu hết các phương pháp điều khiển từ xa hiện nay đó là điều khiển qua tin nhắn điện thoại, bluetooth, wifi, internet.
- Kit gồm các khối đã được cấp nguồn sẵn.
- Kit còn có phần cứng cơ bản như nút nhấn, cảm biến nhiệt độ, độ ẩm, LCD hiển thị giúp sinh viên thực hành các bài cơ bản trực quan nhất.
- Sinh viên tìm hiểu lý thuyết về từng phương pháp điều khiển từ xa sau đó có thực nghiệm trên kit.

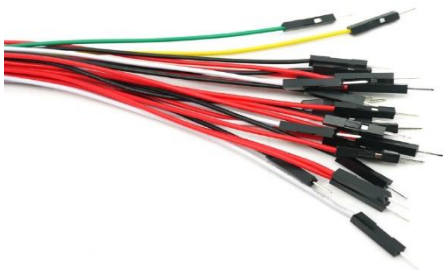
3.4.2 Điều khiển bằng điện thoại qua Bluetooth

Công tác chuẩn bị:

- Kít thực hành điều khiển thiết bị:



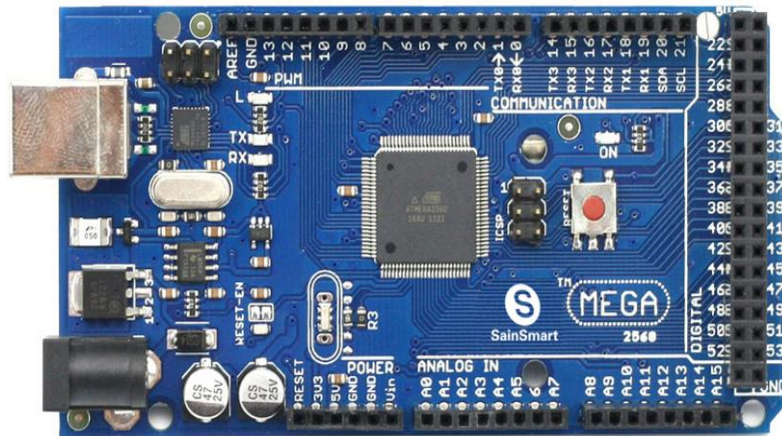
- Dây cắm và bóng đèn



Tiến hành thực hiện:

Chúng ta sẽ sử dụng các thiết bị được hỗ trợ sẵn trên Kit mô hình để thực hiện bài thực hành như: Ổ cắm, các board arduino, dây cắm,...

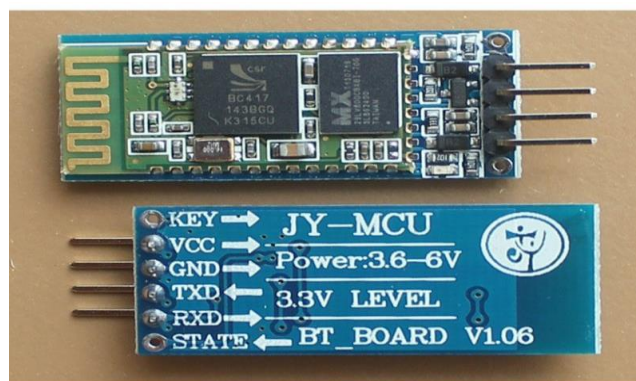
- Các thành phần sử dụng trên kit:



Board arduino MEGA



Mạch relay 4 kênh 5v



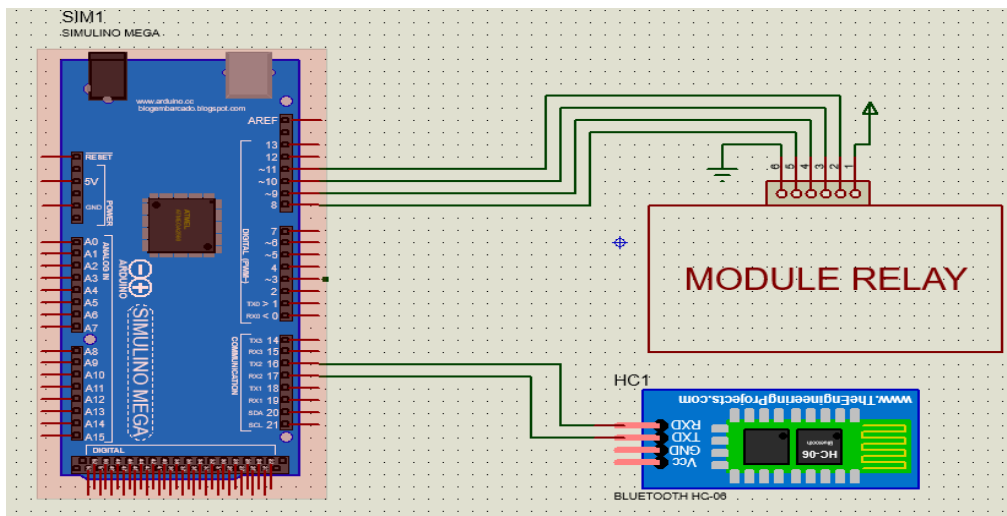
Module bluetooth HC 06

- Nối dây: Nối dây giữa module HC 06 với Arduino

HC 06	ARDUINO
TX	RX2
RX	TX2

Chú ý: Có thể thay đổi 2 chân tín hiệu RX, TX cắm vào Board arduino nhưng phải thay đổi trong phần lập trình

- Nối các dây tín hiệu của module relay và các chân 8, 9, 10, 11 của vi điều khiển Arduino.
- Các mạch trên kit đã được cấp nguồn sẵn nên không cần phải cắm dây nguồn.



Chương trình:

```
char state;
void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);
  Serial.begin(9600);
  Serial2.begin(9600); // Kết nối bluetooth module ở tốc độ 9600
}
void loop() {
  if(Serial2.available() > 0){
    // Đọc giá trị nhận được từ bluetooth
    state = Serial2.read();
    Serial.println(state);
  }
  else
    state = 0;
  switch (state) {
    case 'A':
      digitalWrite(8, LOW);
      break;
    case 'B':
      digitalWrite(8, HIGH);
      break;
    case 'C':
      digitalWrite(9, LOW);
```

```

    break;
case 'D':
    digitalWrite(9, HIGH);
    break;
case 'E':
    digitalWrite(10, LOW);
    break;
case 'F':
    digitalWrite(10, HIGH);
    break;
case 'G':
    digitalWrite(11, LOW);
    break;
case 'H':
    digitalWrite(11, HIGH);
    break;
case 'P':
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    break;
case 'Q':
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(11, HIGH);
    break;
default:
    break;
}
}

```

Ứng dụng điều khiển trên điện thoại Android:

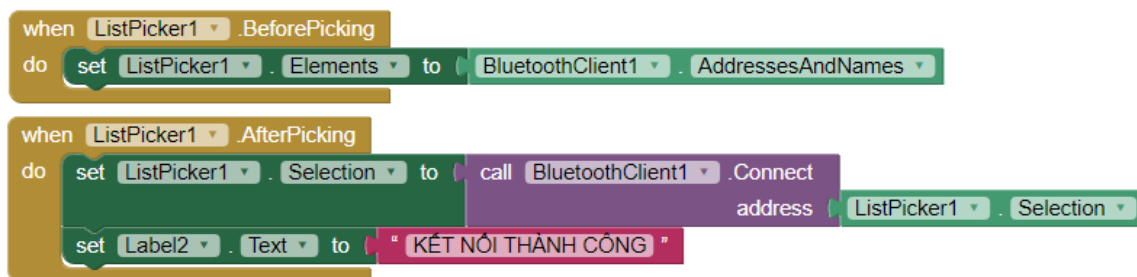
Ở đây nhóm dùng app inventor để viết app điều khiển, xem: <http://ai2.appinventor.mit.edu>

Giao diện :

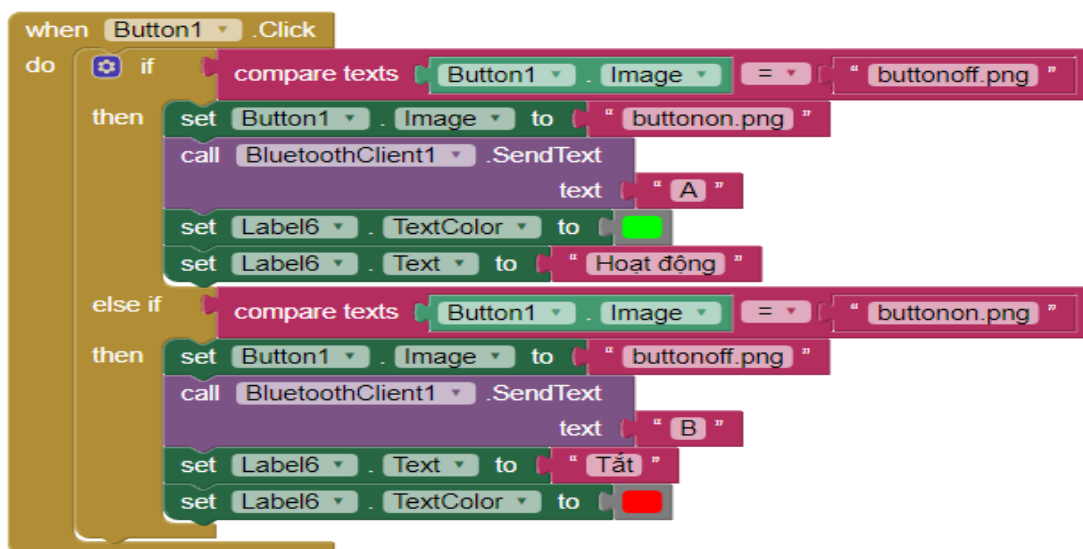


Các khối lệnh cơ bản:

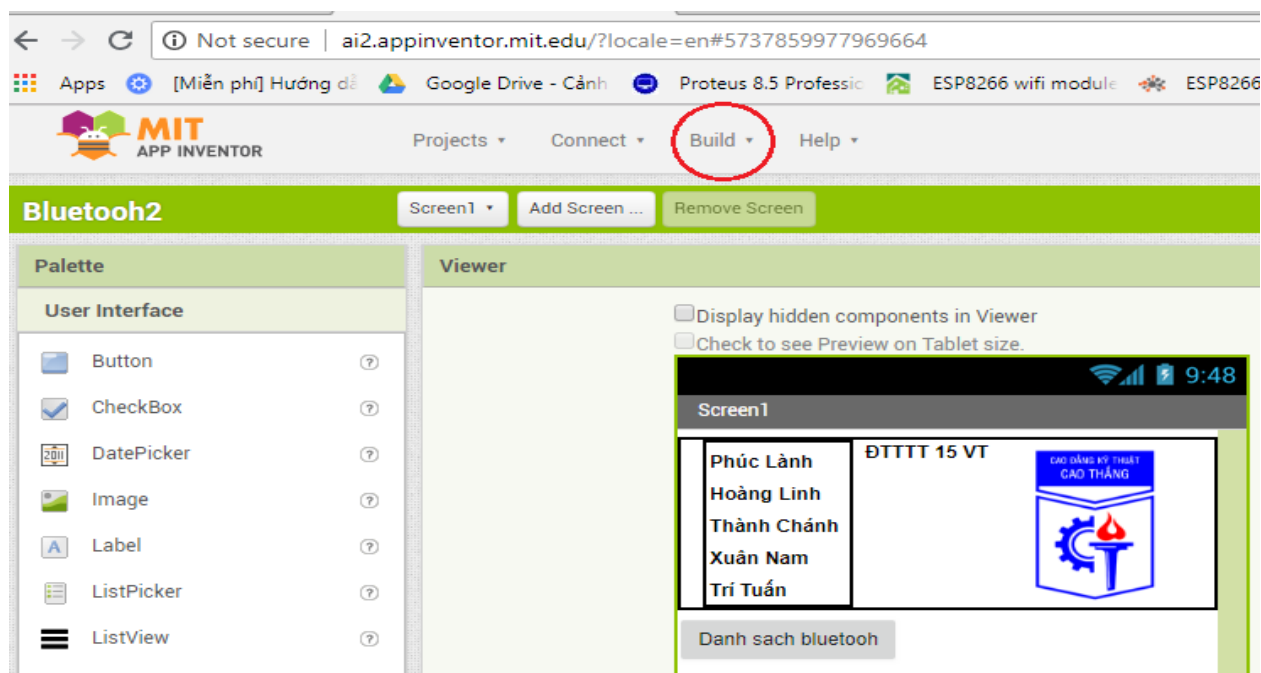
Lệnh kết nối bluetooth:



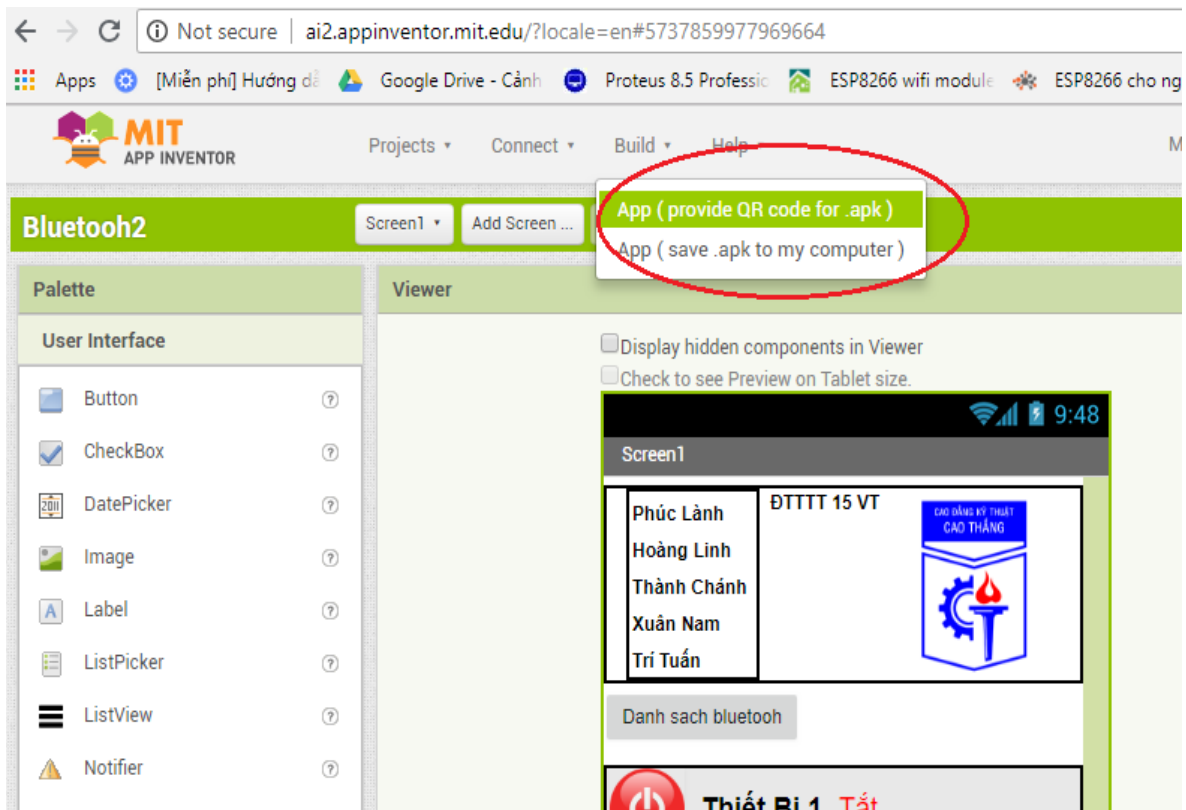
Nút nhấn: Khối lệnh của 1 nút các nút còn lại tương tự như vậy



Khi tạo xong các khối lệnh chúng ta tiến hành Build bằng cách click vào **build** trên mit app:



Khi bạn chưa chắc chắn có thể build bằng cách chọn quét mã QR:



Dùng phần mềm quét mã QR trên điện thoại để cài đặt app và tiến hành điều khiển thử.

Kết quả thực hiện:

Sau khi tải ứng dụng đã viết và cài đặt, chúng ta thực hiện kết nối bluetooth với module HC06. Sau khi kết nối thành công, chúng ta bấm vào các nút trên ứng dụng để điều khiển các thiết bị cắm vào ổ cắm 220v, ở đây là bóng đèn.

3.4.3 Điều khiển bằng điện thoại qua tin nhắn SMS với module SIM800A



Module SIM800

Tiến hành thực hiện: