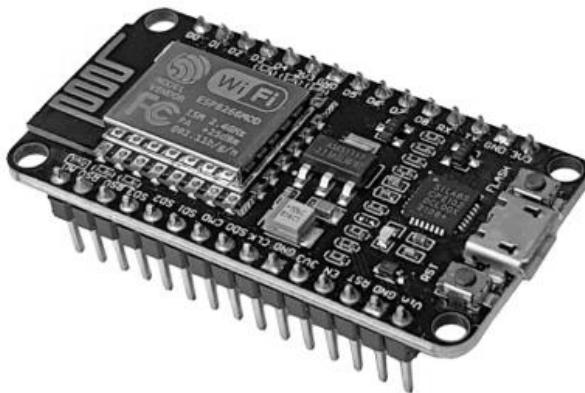


Chương 1: Giới thiệu

Phần 1: Phần cứng

Trong project này ta sử dụng cảm biến siêu âm SRF05 và NodeMCU ESF8266 là chính bên cạnh đó có thể thêm các led, loa thông báo và các dây kết nối các linh kiện với nhau.

1: NodeMCU ESF8266.



ESP8266, hay gọi đầy đủ là **ESP8266EX** là một vi mạch wifi giá rẻ có hỗ trợ bộ giao thức TCP/IP và có thể tích hợp vào thành phần của vi điều khiển, được sản xuất bởi hãng Espressif Systems ở Thượng Hải, Trung Quốc.

Chip ESP8226 lần đầu tiên được các nhà sản xuất phương Tây chú ý vào tháng 8 năm 2014 với **module ESP-01**, do nhà sản xuất bên thứ ba là Ai-Thinker sản xuất. Module này cho phép các vi điều khiển kết nối với mạng Wi-Fi và thực hiện các kết nối TCP/IP đơn giản bằng cách sử dụng các lệnh kiểu Hayes (tập lệnh AT). Tuy nhiên, ban đầu hầu như không có tài liệu tiếng Anh nào về chip và các tập lệnh của nó. Vì mức giá rất thấp với rất ít thành phần bên ngoài khác trên module, các module ESP8266 đã thu hút nhiều hacker khám phá nó và các phần mềm trên đó, cũng như việc dịch thuật các tài liệu tiếng Trung Quốc của chip.

ESP8285 là một chip ESP8266 với 1 MiB bộ nhớ flash được tích hợp, cho phép người dùng có thể sản xuất các thiết bị có khả năng kết nối với Wi-Fi chỉ với một chip đơn.

Các chân kết nối:

- Power supply (tạm dịch: khối nguồn)
- Power-on sequence and reset
- Flash
- Crystal oscillator
- RF
- External resistor (tạm dịch: điện trở ngoại)
- UART
-

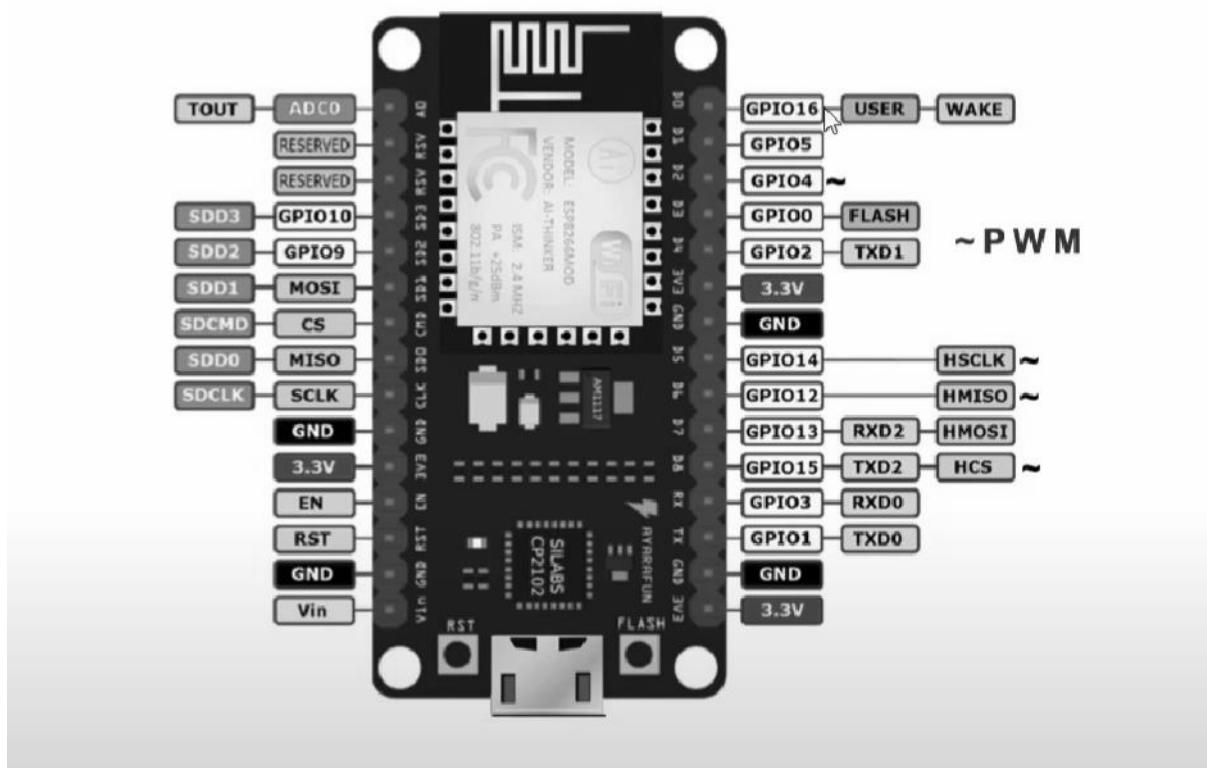
Tính năng:

Các tính năng của chip ESP8266EX bao gồm:

- Bộ xử lý: Lõi vi xử lý L106 32-bit RISC dựa trên Tiêu chuẩn Tensilica Xtensa Diamond, hoạt động ở tần số 80 MHz và SRAM trên chip.
- Bộ nhớ nội: Bộ nhớ nội của ESP8266EX bao gồm SRAM và ROM. ESP8266EX có thể truy xuất các bộ nhớ này thông qua các interface iBus, dBus, và AHB. Kích thước SRAM nhỏ hơn 36kB khi ESP8266EX hoạt động ở chế độ station. ROM của ESP8266EX không lập trình được nên chương trình được lưu trong Flash ngoại.
- Flash ngoại: ESP8266EX sử dụng flash ngoại để lưu trữ chương trình và boot trực tiếp chương trình từ bus SPI. Kích thước flash ngoại mặc định sẽ khác nhau với từng loại module ESP8266, như ESP-01 có kích thước flash ngoại là 1MB, ESP-12E có kích thước flash ngoại là 16MB. Nếu chương trình sử dụng kích thước lớn hơn kích thước mặc định thì bộ flash ngoại với kích thước lớn hơn phải được sử dụng tuy nhiên kích thước flash ngoại tối đa về mặt lý thuyết cho phép là 16 MB. Kích thước bộ nhớ flash nhỏ nhất có thể là 512 kB (tắt chế độ OTA) hoặc 1 MB (bật chế độ OTA).
- IEEE 802.11 b/g/n Wi-Fi:
 - Tích hợp TR switch, balun, LNA, bộ khuếch đại công suất và mạng kết hợp
 - Xác thực WEP hoặc WPA/WPA2 hoặc mạng mở không có password.
- 17 chân GPIO: Các chân có thể được dồn kênh cho các chức năng I²C, I2S, UART, PWM, IR - điều khiển từ xa,...
- 2 hardware timer 23 bit: FRC1 và FRC2.

- 3 SPI: general Slave/Master SPI, Slave SDIO/SPI và general Slave/Master HSPI.
- I²C: Hỗ trợ 1 I²C hoạt động được ở cả chế độ master và slave. Vì các GPIO đều có thể được cấu hình ở chế độ cực máng hở nên các GPIO đều có thể được cấu hình cho các chân SDA hay SCL của I²C bằng phần mềm.
- Giao diện I²S với DMA (sử dụng chung chân với GPIO)
- UART: Hỗ trợ 2 UART là UART0 và UART1. UART0 sử dụng GPIO3 (RX) và GPIO1 (TX). UART1 sử dụng GPIO2 (TX) và GPIO8 (TX). Tuy nhiên GPIO8 được dùng để kết nối bộ nhớ flash của chip nên UART1 chỉ sử dụng được GPIO2 để truyền dữ liệu.
- ADC 10 bit

Sơ đồ chân:

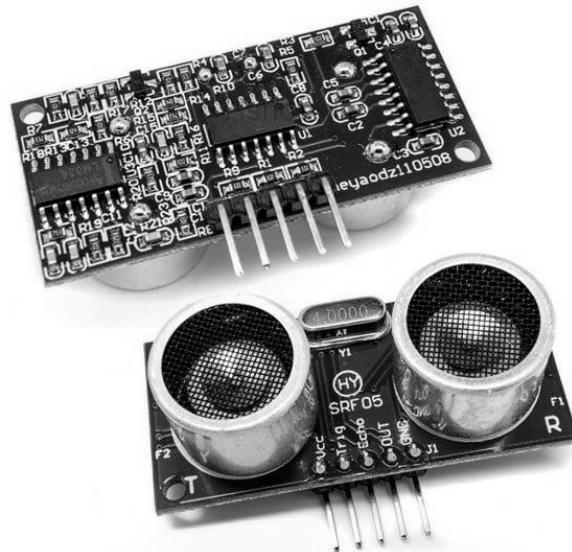


Thông số kỹ thuật:

- WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n

- Điện áp hoạt động: 5VDC thông qua cổng micro USB
- Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
- Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
- Bộ nhớ Flash: 4MB
- Giao tiếp: Cable Micro USB (tương đương cáp sạc điện thoại)
- Hỗ trợ bảo mật: WPA/WPA2
- Tích hợp giao thức TCP/IP
- Lập trình trên các ngôn ngữ: C/C++, MicroPython, Lua

2: Cảm biến siêu âm SRF05



Cảm biến siêu âm UltraSonic HY-SRF05 được sử dụng để nhận biết khoảng cách từ vật thể đến cảm biến nhờ sóng siêu âm, cảm biến có thời gian phản hồi nhanh, độ chính xác cao, phù hợp cho các ứng dụng phát hiện vật cản, đo khoảng cách bằng sóng siêu âm.

Cảm biến siêu âm UltraSonic HY-SRF05 có hai cách sử dụng là sử dụng cặp chân Echo / Trigger hoặc chỉ sử dụng 1 chân Out để phát và nhận tín hiệu, cảm biến được sử dụng phổ biến với vô số bộ thư viện và Code mẫu với Arduino.

Thông số kỹ thuật:

- Điện áp hoạt động: 5VDC
- Dòng tiêu thụ: 10~40mA
- Tín hiệu giao tiếp: TTL
- Chân tín hiệu: Echo, Trigger (thường dùng) và Out (ít dùng).
- Góc quét:<15 độ
- Tần số phát sóng: 40Khz
- Khoảng cách đo được: 2~450cm (khoảng cách xa nhất đạt được ở điều kiện lý tưởng với không gian trống và bề mặt vật thể bằng phẳng, trong điều kiện bình thường cảm biến cho kết quả chính xác nhất ở khoảng cách <100cm).
- Sai số: 0.3cm (khoảng cách càng gần, bề mặt vật thể càng phẳng sai số càng nhỏ).
- Kích thước: 43mm x 20mm x 17mm

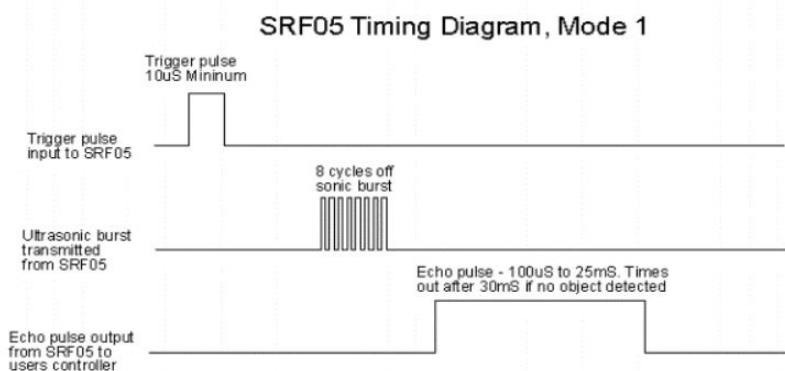
Sơ đồ chân: Có 5 chân

- VCC : 5V.
- Trig(T) : digital input.
- echo (R): digital output.
- OUT.
- GND.

Nguyên lý hoạt động:

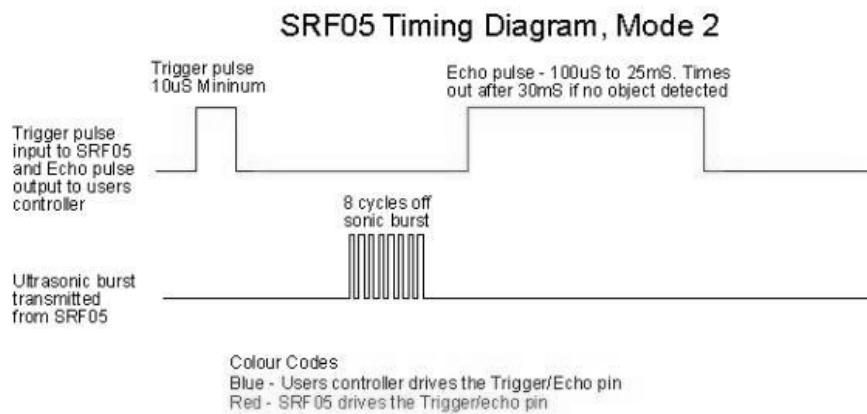
➤ **Ở chế độ 1: Tách biệt, kích hoạt và phản hồi**

Để đo khoảng cách, ta phát 1 xung rất ngắn (5 microSeconds) từ chân TRIG. Sau đó cảm biến sẽ tạo ra 1 xung HIGH ở chân ECHO cho đến khi nhận được xung phản xạ ở chân này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến quay trở lại. Tốc độ của âm thanh trong không khí là 340 m/s tương đương với 29,412 microSeconds/cm($1000000/(340*100)$). Khi đã tính được thời gian ta chia cho 29,412 để được khoảng cách cần đo.



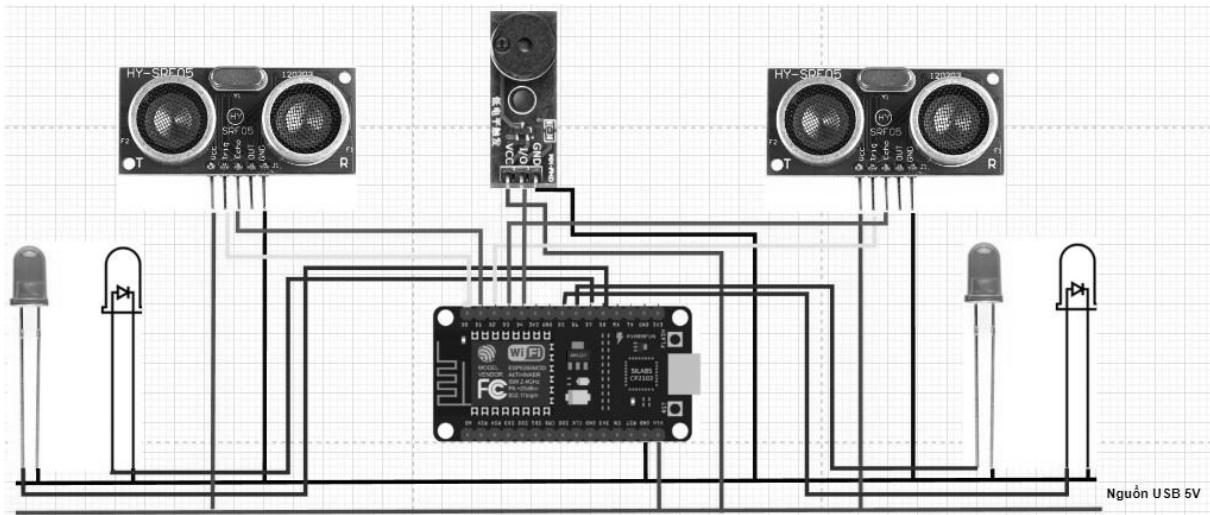
➤ **Ở chế độ 2: Dùng 1 chân cho cả kích hoạt và phản hồi**

Ta sử dụng chân OUT để nó vừa phát ra xung rồi nhận xung phản xạ về, chân chế độ thì nối đất. Tín hiệu hồi tiếp sẽ xuất hiện trên cùng 1 chân với tín hiệu kích hoạt. SRF05 sẽ không tăng dòng phản hồi cho đến 700 μ s sau khi kết thúc các tín hiệu kích hoạt và bạn đã có thời gian để kích hoạt pin xoay quanh và làm cho nó trở thành 1 đầu vào.



Do góc quét của cảm biến được mở rộng theo hình nón nên cảm biến càng xa sẽ không chính xác, ngoài ra đối với các bề mặt không phẳng hoặc bề mặt bị cong, xiên cũng làm ảnh hưởng đến độ chính xác của cảm biến.

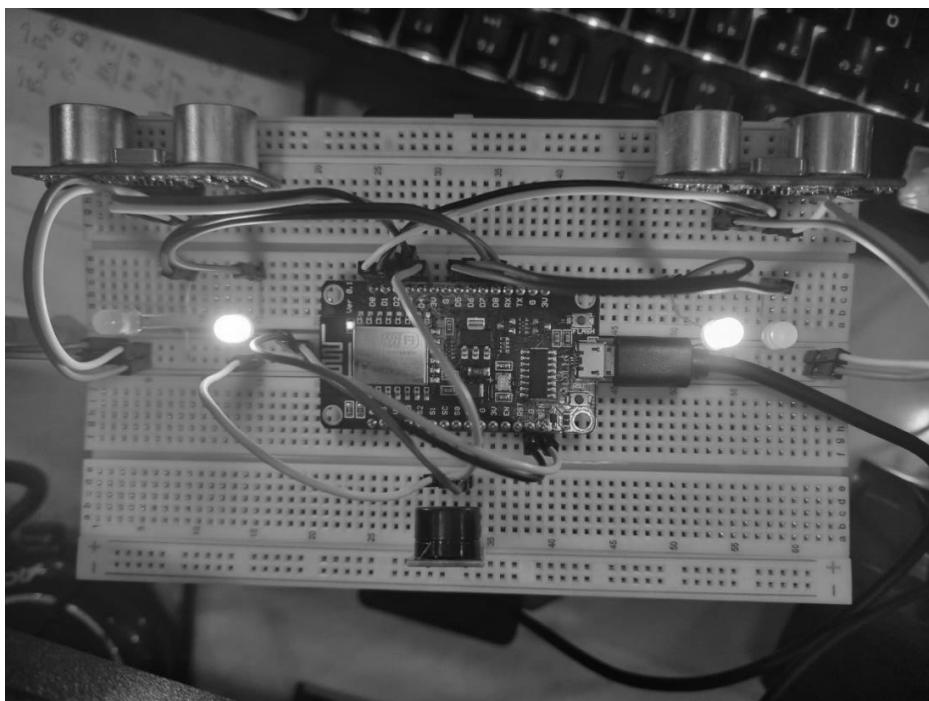
Phần 2: Sơ đồ mạch



Dạng sơ đồ mạch được kết nối giữa MCU và cảm biến như trên, ngoài ra nhóm em còn thêm các LED và loa để thông báo khi đạt khoảng cách nhất định của hệ thống đỗ xe.

1: Project của nhóm

Hình thực tế:



2: Nguyên tắc hoạt động

- Cảm biến siêu âm được sử dụng để tính khoảng cách giữa xe và các vật cản trong bãi đỗ xe.
- Khoảng cách giữa chướng ngại vật và xe sẽ được tín hiệu từ cảm biến gửi về MCU và được publish lên database của Blynk, từ ứng dụng này ta có thể xem được thông tin trực tiếp từ ứng dụng trên điện thoại di động hay là laptop...
- Khi data base nhận được tín hiệu từ cảm biến siêu âm, ta có thể truy cập lấy thông tin và nhận thông báo về điện thoại, laptop, ipad... của mình.
- Điều kiện để có thể nhận được tín hiệu là MCU và điện thoại kết nối với Blink phải có kết nối internet.
- MCU sẽ điều khiển LED và loa để phát ra cảnh báo cho người dùng phát hiện.

Phần 3: Ứng dụng Blynk (<https://blynk.io/>)

1: Giới thiệu về Blink



Blynk là một nền tảng với các ứng dụng điện thoại thông minh cho phép ta có thể dễ dàng tương tác với bộ vi điều khiển như: Arduino, Esp8266, Esp32 hoặc Raspberry qua Internet.

Blynk App là một bảng điều khiển kỹ thuật số cho phép ta có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget khác nhau mà nhà cung cấp thiết kế sẵn.

Blynk không bị ràng buộc với một số bo hoặc shield cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà ta lựa chọn. Cho dù Arduino hoặc Raspberry Pi của bạn được liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp kết nối và sẵn sàng cho các dự án IoT.

Blynk Server – chịu trách nhiệm về tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Có thể sử dụng Blynk Cloud hoặc chạy cục bộ máy chủ Blynk riêng của mình. Nó là mã nguồn mở, có thể dễ dàng xử lý hàng nghìn thiết bị và thậm chí có thể được khởi chạy trên Raspberry Pi.

Thư viện Blynk – dành cho tất cả các nền tảng phần cứng phổ biến – cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và lệnh đi.

Mỗi khi ta nhấn một nút trong ứng dụng Blynk, thông điệp sẽ truyền đến không gian của đám mây Blynk, và tìm đường đến phần cứng của bạn.

Mọi thứ mình cần để xây dựng và quản lý phần cứng được kết nối: cung cấp thiết bị, hiển thị dữ liệu cảm biến, điều khiển từ xa với các ứng dụng web và di động, cập nhật chương trình cơ sở qua mạng, bảo mật, phân tích dữ liệu, quản lý người dùng và truy cập, cảnh báo, tự động hóa và nhiều thứ khác hơn...

Đặc tính

- API và giao diện người dùng tương tự cho tất cả phần cứng và thiết bị được hỗ trợ
- Kết nối với đám mây bằng cách sử dụng:
 - Wifi
 - Bluetooth và BLE
 - Ethernet
 - USB (Nối tiếp)
 - GSM
 - ...
- Bộ Widget dễ sử dụng
- Thao tác ghim trực tiếp mà không cần viết mã
- Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng ghim ảo
- Theo dõi dữ liệu lịch sử qua tiện ích SuperChart
- Giao tiếp giữa thiết bị với thiết bị sử dụng Bridge Widget
- Gửi email, tweet, push notification...

Các tính năng mới liên tục được bổ sung!

Những ưu điểm của Blynk

- Có thể cập nhật OTA
- Có sẵn phần thêm ESP làm thiết bị và add wifi cho ESP sử dụng
- Các button có thể thêm icon, hình ảnh vào để cá nhân hóa, thể hiện trực quan hơn, người dùng có thể dễ dàng nhận biết button đó là để điều khiển thiết bị nào.
- Không cần thêm cầu kết nối data giữa các thiết bị khác nhau ở code nữa , chỉ cần chọn nó dùng data stream nào là được, ví dụ là dùng 2 thiết bị muốn ấn bật tắt trên 1 thiết bị , nó sẽ đồng bộ hóa ngay vs thiết bị kia để hiển thị trạng thái của cả 2 là giống nhau
- Sử dụng số lượng Widget box tùy ý không bị giới hạn Enzeny như xưa, ngày xưa muốn dùng nhiều sẽ phải làm bộ Pi server, vừa tốn tiền, vừa bị lag hơn server chính hãng, dùng máy server chia sẻ trên các nhóm cộng đồng thì mất bảo mật.

Những nhược điểm của app blynk

- Bản miễn phí chỉ dùng được 2 Devices trên 1 ứng dụng, Device ở đây tương tự như Project trên app cũ, app cũ có thể tạo được nhiều project
- Ngoài ra bản miễn phí còn 1 cái nữa là cái đồ thị theo dõi (Superchart) chỉ dùng được duy nhất 1 data stream ,muốn xem thêm cái khác thì phải thêm ô (Superchart) khác nữa và sẽ khó so sánh các thông số trực quan, bản cũ sẽ dùng được 4 cái stream sẵn có, bản mới muốn có phải xài bảng tốn phí
- Không có theo dõi GPS như bảng cũ

Như vậy, theo mình thấy app mới cho người dùng doanh nghiệp là chính, phức tạp hơn cho lập trình viên vì cách setup nhiều bước hơn tuy nhiên có nhiều tính năng mới, thiết kế đẹp, có OTA là 1 lợi thế. App cũ phù hợp cho các bạn sinh viên, những ai cần nghiên cứu, các đồ án, dự án nhỏ...

2: Cài đặt và sử dụng Blink với ESP8266

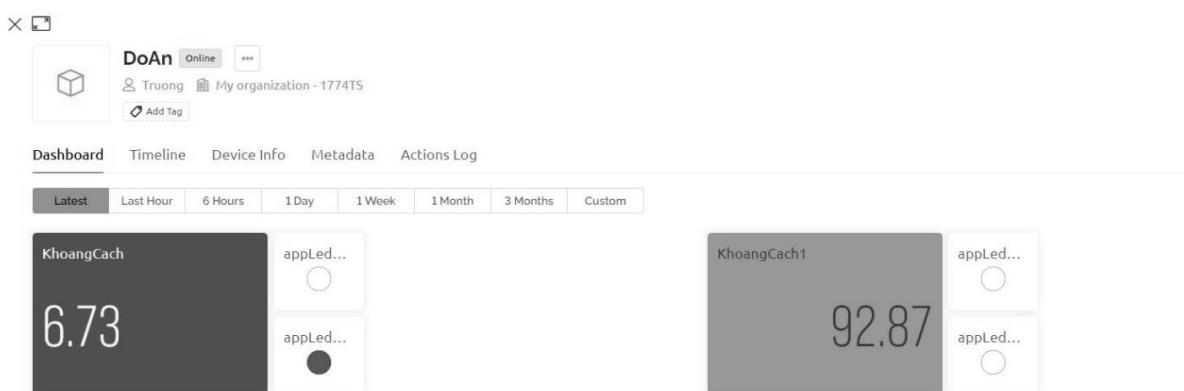
Đầu tiên yêu cầu ở phần mềm Arduino Ide đã cài board esp8266, thêm thư viện cho blynk mới, sau khi tải vào Arduino IDE chọn Sketch -> Include library -> Add .ZIP library và chọn file vừa tải về.

2.1: Thiết lập Blink trên máy tính:

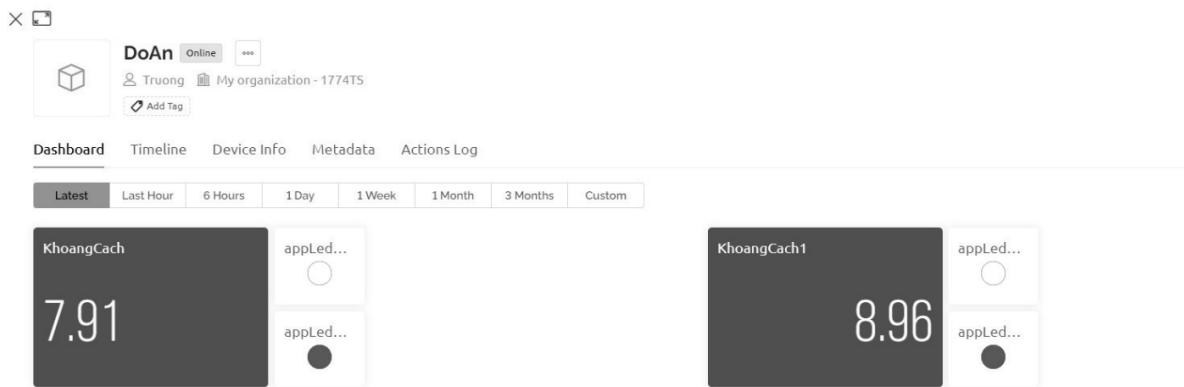
- Đầu tiên truy cập vào <https://blynk.cloud/> đăng nhập.
- Vào chọn New Template, nhập tên và chọn đầy đủ.

- Sau khi tạo xong sẽ hiện giao diện, ta copy mã Template để dán vào code.
- Tiếp theo, chọn Datastreams -> Virtual Pin -> nhập đầy đủ datastream của Pin -> Create.
- Sau khi tạo Pin xong, ta chọn Web Dashboard, kéo các Gauge bên trái qua để hiển thị nhiệt độ và độ ẩm, Switch để làm công tắc bật tắt Led, một biểu đồ chart để để hiển thị nhiệt độ (bản free chỉ hiển thị được 1 biểu đồ), nhấn biểu tượng cài đặt để chọn từng Pin hiển thị phù hợp.
- Tiếp theo, chọn biểu tượng Search -> New Device để chọn thiết bị từ From template.
- Chọn tên template mà bạn đã tạo -> Create, sau đó xem kết quả.

Sau khi thiết lập kết quả sẽ được như hình sau:



Khi ta đưa vật cần lại gần cảm biến siêu âm cảnh báo hiện lên màu đỏ như hình với khoảng cách cảnh báo được set tùy ý. Đèn led sẽ sáng và loa sẽ kêu lên để cảnh báo với người dùng.



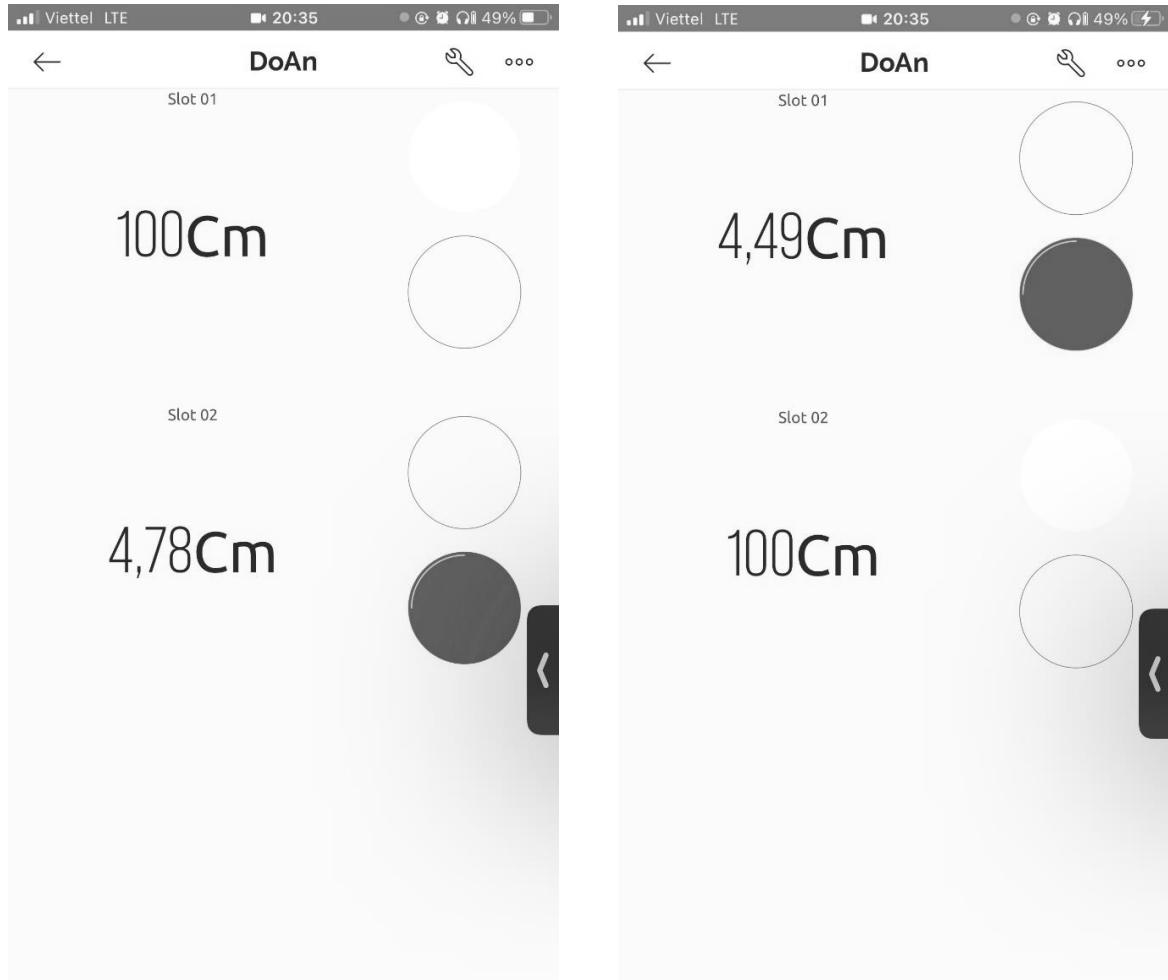
2.2: Thiết lập Blink trên điện thoại:

Trên điện thoại sau khi tải app Blynk mới về, mở lên sau đó đăng nhập tài khoản đã tạo bên web, tên thiết bị bạn tạo lúc nãy trên web sẽ được hiển thị sẵn:

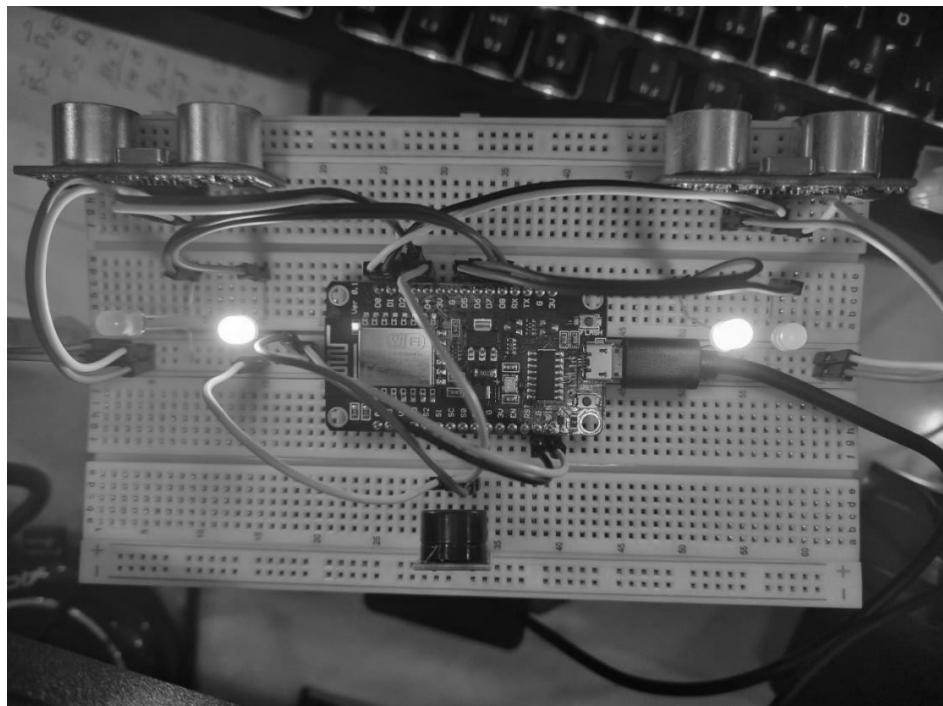
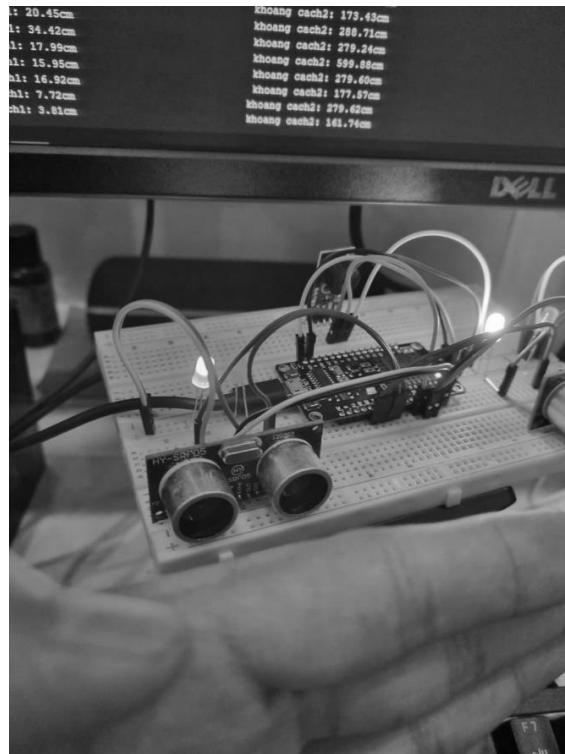
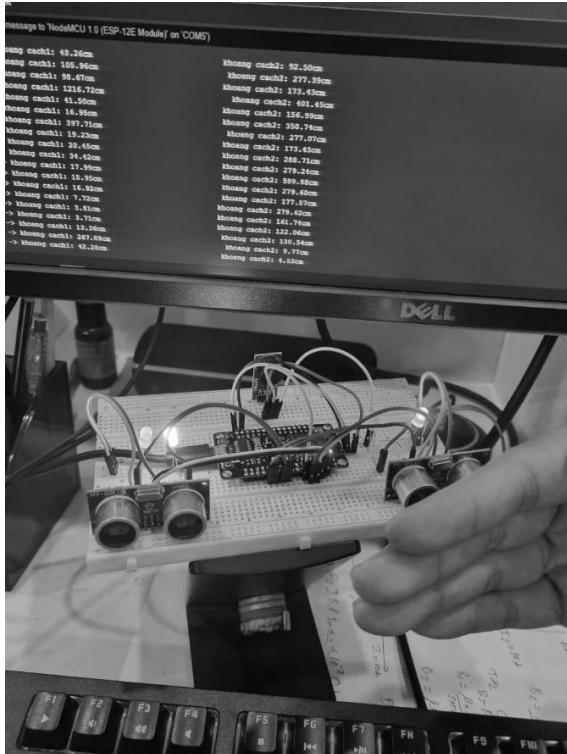
Chọn biểu tượng Button để điều khiển led, Value Display để hiển thị nhiệt độ, độ ẩm... nhớ chọn chân Pin cho từng mục.

Thành quả sau khi tạo xong, ta có thể thay đổi màu, thiết kế giao diện cho từng phần.

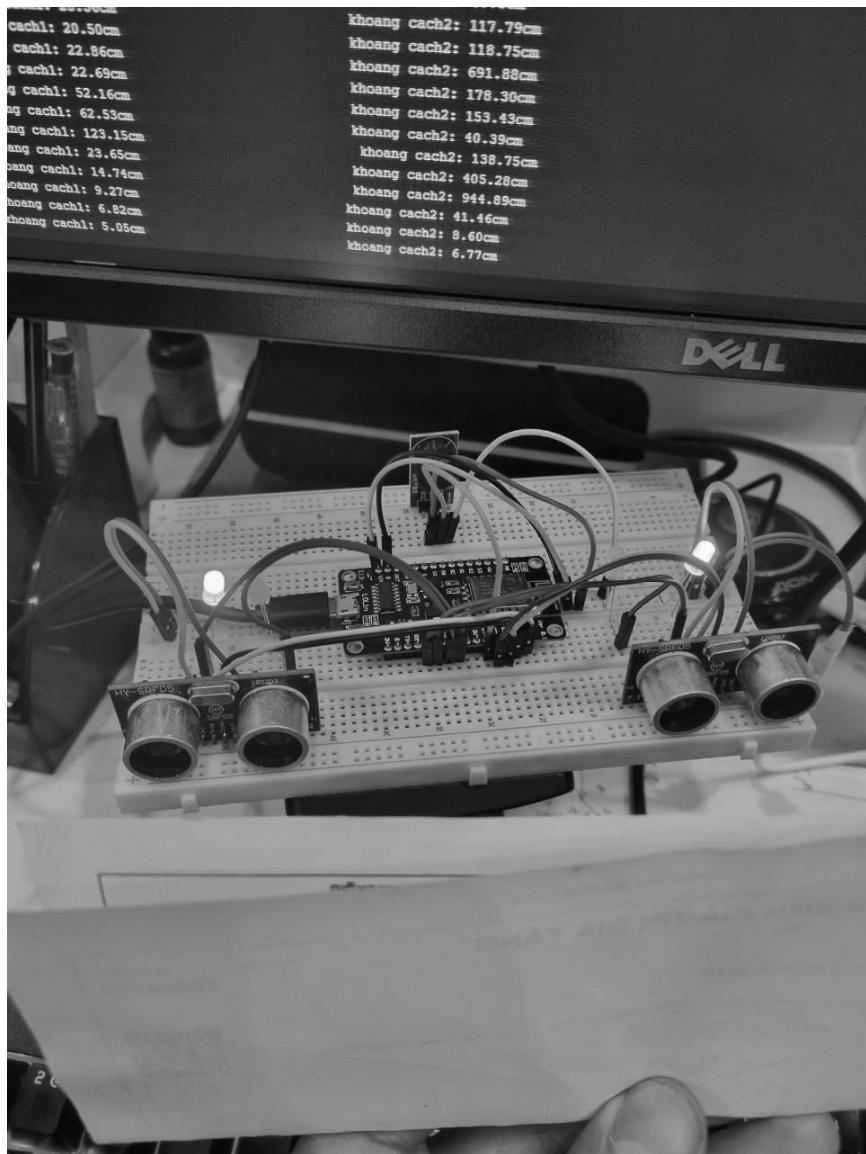
Hình ảnh minh họa:



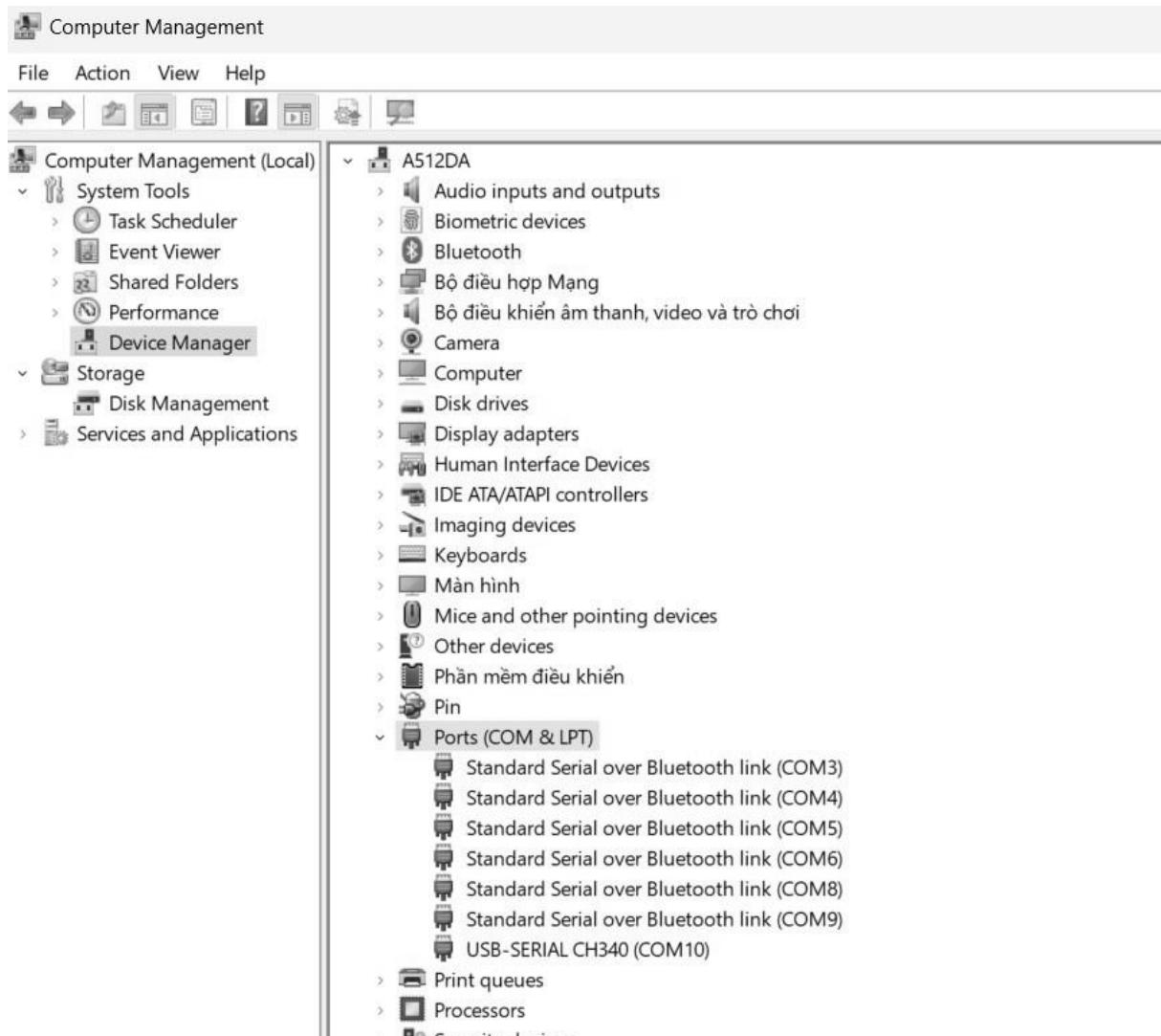
Phần 3: CHẠY DEMO

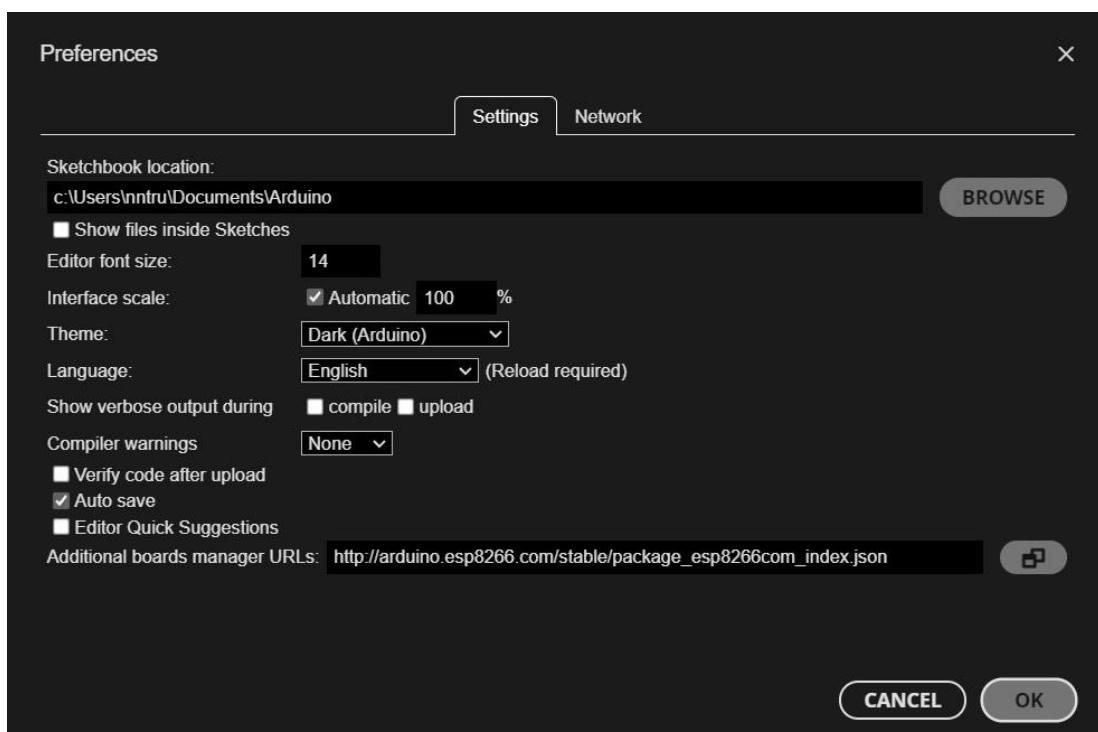
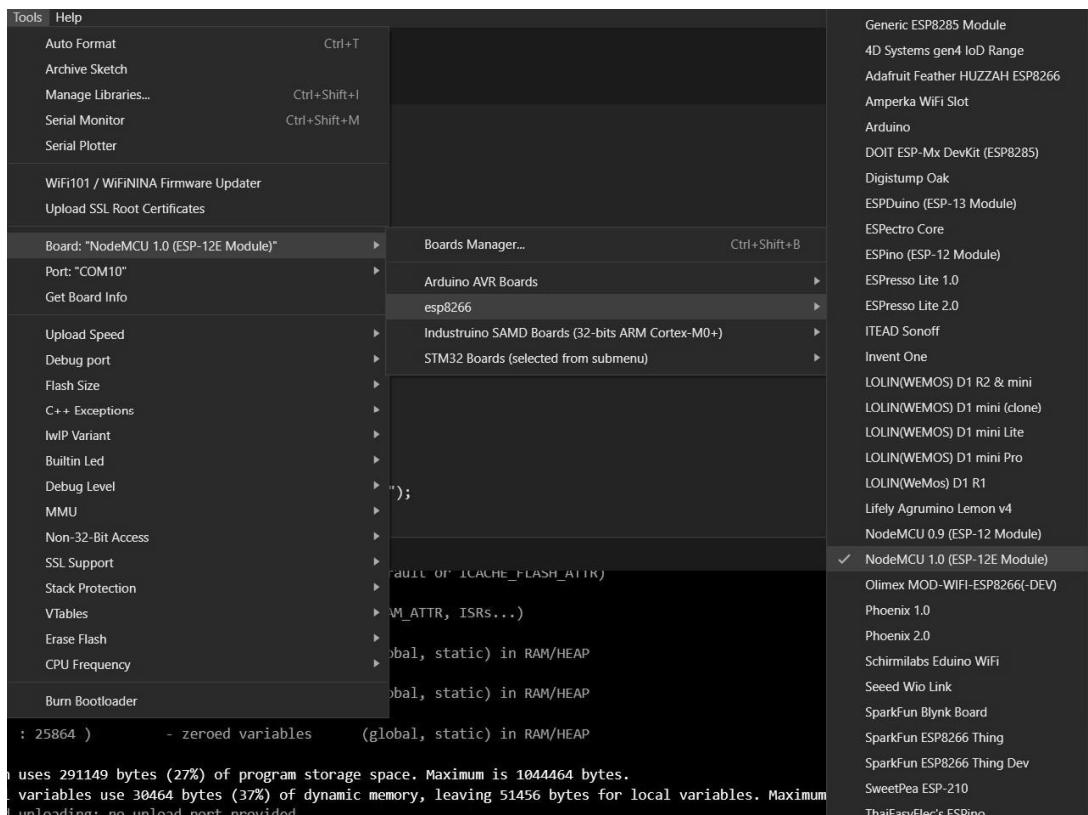


Khi phát hiện vật cản đèn Led sẽ sáng màu đỏ và loa phát ra tiếng kêu.



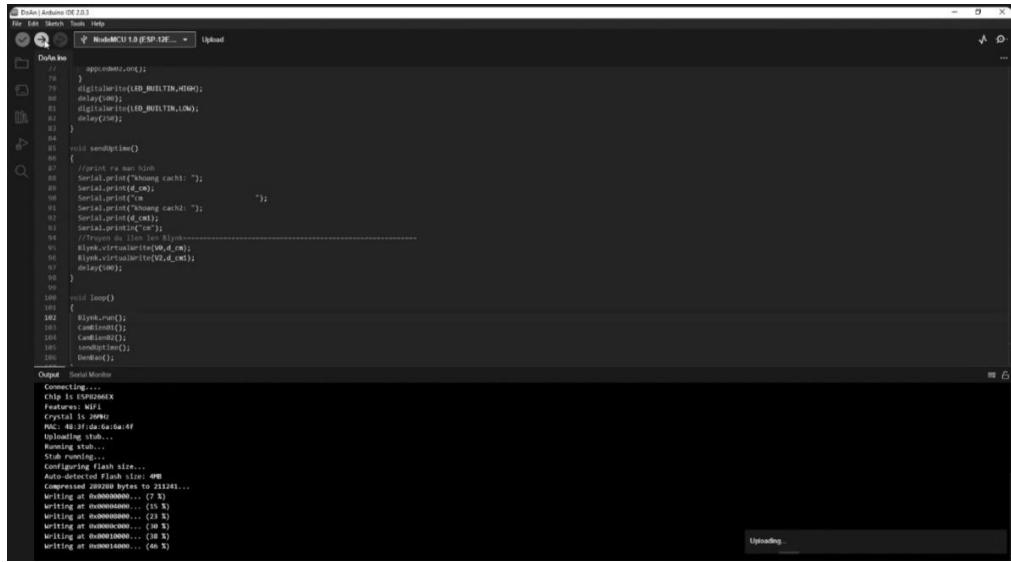
Kết nối các board lại với nhau:



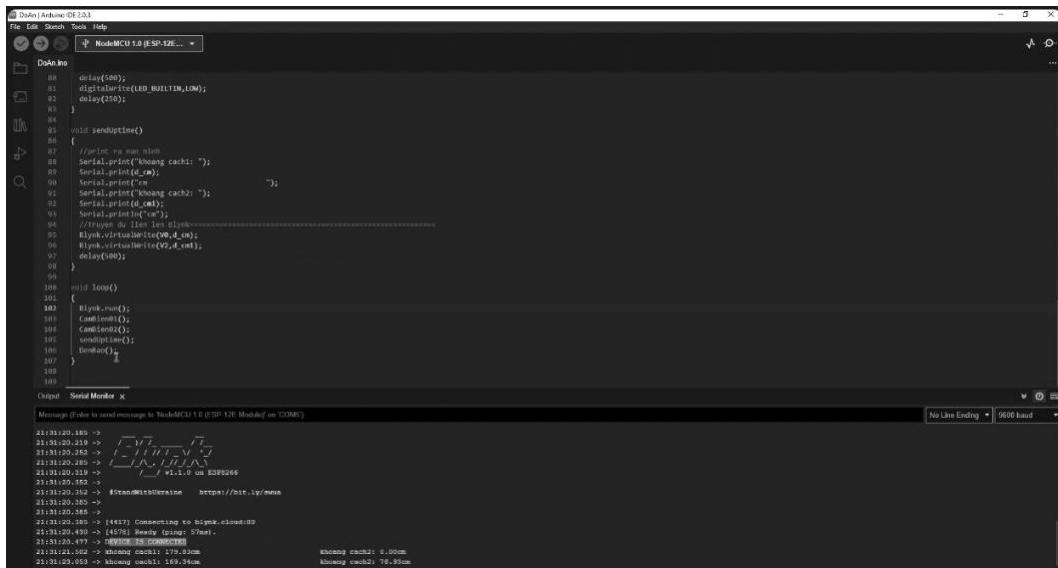


Phần 4: CODE

Code được nạp thành công:



Khoảng cách được đo và hiện thị bên tab Serial Monitor.



```

#define BLYNK_PRINT Serial
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL6QmC81r_"
#define BLYNK_DEVICE_NAME "DoAn"
#define BLYNK_AUTH_TOKEN "ytFTyiLrhZs1D9ugaxysckra7rLDP2xr"
#define trigPin D0
#define echoPin D1
#define trigPin1 D2
#define echoPin1 D3
#define LedW01 D6
#define LedR01 D5
#define LedW02 D7
#define LedR02 D8
#define Buzzer D4
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "ytFTyiLrhZs1D9ugaxysckra7rLDP2xr";
char ssid[] = "NhatTruong";
char pass[] = "tamsohai";
int duration;
float d_cm;
int duration1;
float d_cm1;
float sinVal;
int toneVal;
WidgetLED appLedW01(V3);
WidgetLED appLedR01(V4);
WidgetLED appLedW02(V5);
WidgetLED appLedR02(V6);
void setup()
{
    Serial.begin(9600); // Debug console
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);
    pinMode(LedW01, OUTPUT);
    pinMode(LedR01, OUTPUT);
    pinMode(LedW02, OUTPUT);
    pinMode(LedR02, OUTPUT);
    pinMode(Buzzer , OUTPUT);
    Blynk.begin(auth, ssid, pass);
    //Thong bao ket noi
    Serial.println("DEVICE IS CONNECTED");
}
void CamBien01()
{
    //cam bien song am 1
    digitalWrite(trigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
}

```

```

delayMicroseconds(10);
digitalWrite(trigPin,LOW);
duration = pulseIn(echoPin,HIGH); // doc gia tri canh len phan hoi ve mat
doc
d_cm = duration * 0.034/2.; // S = T 0.0343/2 Khoang cach tu vat den mat
}
void CamBien02()
{
    //cam bien song am 2
    digitalWrite(trigPin1,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin1,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin1,LOW);
    duration1 = pulseIn(echoPin1,HIGH); // doc gia tri canh len phan hoi ve
mat doc
d_cm1 = duration1 * 0.034/2.;// S = T 0.0343/2 Khoang cach tu vat den mat
}

void buzzer()
{
    for(int x=0; x<180; x++) {
        // convert degrees to radians then obtain value
        sinVal = (sin(x*(3.1412/180)));
        // generate a frequency from the sin value
        toneVal = 2000+(int(sinVal*1000));
        tone(Buzzer, toneVal);
        delay(2);
    }
}

void DenBao()
{
    if(d_cm<30) {
        appLedR01.on();
        appLedW01.off();
        digitalWrite(LedR01,HIGH);
        digitalWrite(LedW01,LOW);
    }
    else{
        appLedR01.off();
        appLedW01.on();
        digitalWrite(LedW01,HIGH);
        digitalWrite(LedR01,LOW);
    }
    if(d_cm1<30){
        appLedR02.on();
        appLedW02.off();
        digitalWrite(LedR02,HIGH);
        digitalWrite(LedW02,LOW);
    }
    else{
        appLedR02.off();
        appLedW02.on();
        digitalWrite(LedW02,HIGH);
        digitalWrite(LedR02,LOW);
    }
}

```

```

if(d_cm1<10 || d_cm<10)
{
    buzzer();
    delay(2500);
    noTone(Buzzer);
    delay(1500);
}
if(d_cm1>=10 && d_cm>=10)
{
    noTone(Buzzer);
}
}
void sendUptime()
{
    //print ra man hinh
    Serial.print("khoang cach1: ");
    Serial.print(d_cm);
    Serial.print("cm") ;
    Serial.print("khoang cach2: ");
    Serial.print(d_cm1);
    Serial.println("cm");
    //Truyen du lieu len Blynk
    Blynk.virtualWrite(V0,d_cm);
    Blynk.virtualWrite(V2,d_cm1);
    delay(500);
}
void loop()
{
    Blynk.run();
    CamBien01();
    CamBien02();
    sendUptime();
    DenBao();
}

```

Tài liệu tham khảo:

<https://dientuviet.com/huong-dan-lap-trinh-esp8266-nodemcu/>

<https://techatronic.com/smart-parking-system-using-iot/>

<https://blynk.io/>

<https://nshopvn.com/blog/huong-dan-cai-dat-va-su-dung-blynk-new-2-0-tren-arduino-ide-voi-esp8266/>

<https://dientuviet.com/huong-dan-lap-trinh-esp8266-nodemcu/>