



TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG
BỘ MÔN ĐIỆN TỬ VIỄN THÔNG

LƯU VĂN ĐẠI
LẠI NGUYỄN DUY

GIÁO TRÌNH **ĐIỀU KHIỂN THIẾT BỊ TỪ XA**

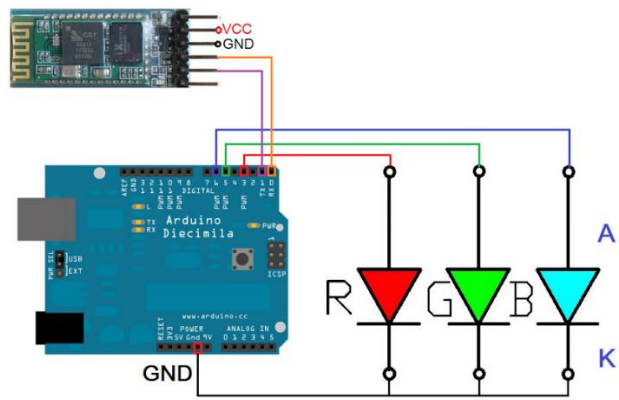
(GIÁO TRÌNH DÙNG CHO HỆ CAO ĐẲNG
NGÀNH CNKT ĐIỆN TỬ, TRUYỀN THÔNG
CHUYÊN NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG)



Android Device



Bluetooth Module



Arduino Board

RGB LEDs

TP. HỒ CHÍ MINH, 09 – 2019

MỤC LỤC

MỤC LỤC	2
CHƯƠNG 1. ĐIỀU KHIỂN TỪ XA BẰNG TIA HỒNG NGOẠI.....	3
1.1 GIỚI THIỆU.....	3
1.2 PHÂN CỨNG.....	3
1.3 GIẢI THUẬT VÀ LẬP TRÌNH GIẢI MÃ.....	5
1.4 KẾT QUẢ.....	12
1.5 KẾT LUẬN	14
1.6 CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1	14
CHƯƠNG 2. ĐIỀU KHIỂN TỪ XA BẰNG TẦN SỐ VÔ TUYẾN	15
2.1 REMOTE RF.....	15
2.2 GIẢI MÃ TÍN HIỆU RF 433MHZ VỚI ARDUINO.....	19
CHƯƠNG 3. ĐIỀU KHIỂN TỪ XA QUA BLUETOOTH.....	23
3.1 MODULE BLUETOOTH HC-06	23
3.2 CÁC THÀNH PHẦN	24
3.3 NỐI DÂY	25
3.4 CHƯƠNG TRÌNH.....	25
3.5 ỨNG DỤNG ĐIỀU KHIỂN TRÊN ĐIỆN THOẠI ANDROID.....	27
CHƯƠNG 4. ĐIỀU KHIỂN TỪ XA QUA INTERNET.....	30
4.1 MODULE ESP8266 WEMOS D1	30
4.1.1 Thông Số Kỹ Thuật	30
4.1.2 Sơ đồ pin digital.....	31
4.2 CÁC THÀNH PHẦN	31
4.3 NỐI DÂY	32
4.4 THƯ VIỆN	32
4.5 CHƯƠNG TRÌNH NẠP VÀO WEMOS D1	32
4.6 HƯỚNG DẪN SỬ DỤNG ỨNG DỤNG BLYNK.....	33
4.7 KẾT QUẢ THỰC HIỆN.....	38
CHƯƠNG 5. ĐIỀU KHIỂN TỪ XA BẰNG ĐIỆN THOẠI QUA SMS	40
5.1 TIN NHẮN SMS	40
5.2 MODULE SIM800A	41
5.3 KHẢO SÁT TẬP LỆNH AT CỦA MODULE SIM800A.....	43
5.3.1 Các thuật ngữ.....	43
5.3.2 Cú pháp lệnh AT.....	44
5.3.3 Các lệnh AT cơ bản	44
5.4 PHÂN CỨNG.....	46
5.5 SƠ ĐỒ KẾT NỐI	47
5.6 CHƯƠNG TRÌNH ARDUINO	47
5.7 CÚ PHÁP NHẮN TIN ĐIỀU KHIỂN	47
5.8 HƯỚNG PHÁT TRIỂN	48
TÀI LIỆU THAM KHẢO	50

CHƯƠNG 1. ĐIỀU KHIỂN TỪ XA BẰNG TIA HỒNG NGOẠI

1.1 GIỚI THIỆU

Ngày nay, con người đã và đang tìm các cách khác nhau, dựa trên những nền tảng của ngành công nghiệp điện tử để thực hiện công việc truyền thông và liên lạc sao cho nhanh chóng và hiệu quả nhất. Từ sử dụng dây cáp điện, cáp quang, tia hồng ngoại cho đến sóng vô tuyến cự ly ngắn, sóng vi ba và sử dụng cả vệ tinh. Chúng đã hầu như đáp ứng được mọi nhu cầu khác nhau của con người.

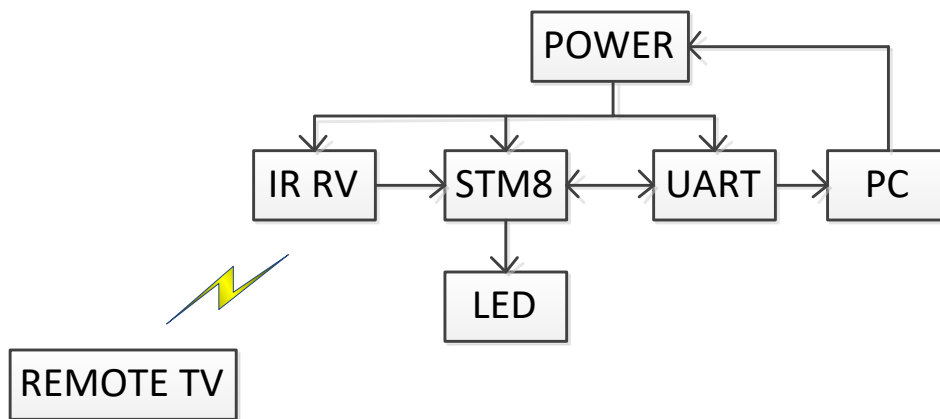
Trong đó có một phương thức truyền tải thông tin rất đặc biệt, đó chính là sử dụng hồng ngoại. Phương thức này có hạn chế là khoảng cách ngắn và tốc độ thấp. Tuy nhiên, nó cũng có những lợi thế chỉ của riêng mình mới có như: nó là một loại tia sáng nên không bị ảnh hưởng bởi môi trường điện từ, thiết bị phát và thu thì dễ thiết kế, lắp ráp, và chi phí thấp.

Chính vì những ưu điểm này, cho đến nay, hồng ngoại vẫn là phương thức truyền tải thông tin hữu ích, đặc biệt trong những công việc yêu cầu đơn giản – cự ly ngắn, tốc độ thấp, cần chi phí thấp, môi trường có nhiều điện từ. Đây là một phương thức hiệu quả và tiết kiệm, phù hợp để truyền tải thông tin trong mạng nhỏ. Ví dụ như từ máy tính sang máy tính, máy tính sang điện thoại, điện thoại với điện thoại...hoặc các thiết bị hiện đại gia dụng khác.

Chương này cung cấp kiến thức giúp người đọc dễ dàng thiết kế mạch ứng dụng điều khiển thiết bị điện trong nhà bằng remote TV. Có rất nhiều vi điều khiển có thể thực hiện việc giải mã, tuy nhiên tác giả chọn vi điều khiển STM8 của hãng ST vì tính kinh tế, giá thành thấp.

1.2 PHẦN CỨNG

1.2.1 Sơ đồ khối

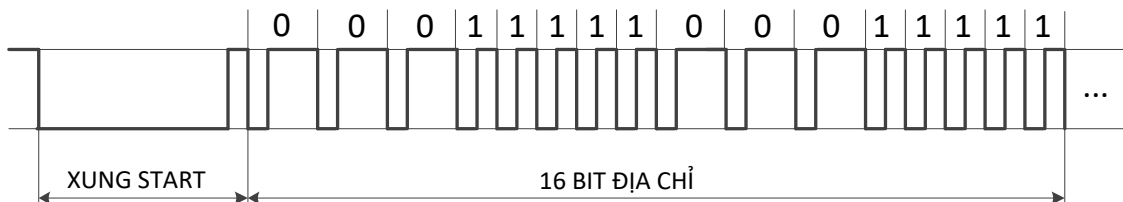


Hình 1.1 Sơ đồ khối

Mạch giải mã gồm mắt thu hồng ngoại gửi dữ liệu tới vi điều khiển STM8. Vi điều khiển giải mã và gửi dữ liệu lên máy vi tính qua UART. STM8 cũng điều khiển on off một LED khi giải mã xong.

1.2.2 Remote ti vi và mắt thu hồng ngoại

Remote là một thiết bị phát sóng hồng ngoại, sử dụng trong các mục đích điều khiển từ xa (tầm 10m). Remote nhận lệnh điều khiển từ người điều khiển thông qua các phím bấm, sau đó xuất ra một khung dữ liệu ứng với phím được bấm. Có rất nhiều loại remote được sử dụng như: Sony, LG,... mỗi loại có 1 cách mã hóa phím bấm khác nhau. Khi 1 nút nhấn được ấn, khung truyền dữ liệu có dạng như sau:



Hình 1.2 Khung truyền

Sau xung START và 16 bit địa chỉ là mã của nút nhấn được bấm.

- Bit 0: tín hiệu ở mức thấp trong 700 μ s và ở mức cao trong 500 μ s.
- Bit 1: tín hiệu ở mức thấp trong 700 μ s và ở mức cao trong 1600 μ s.

Mắt thu hồng ngoại gồm 3 chân: VCC, GND, OUT. Nó có chức năng chuyển sóng hồng ngoại phát ra từ remote thành đúng dạng tín hiệu số ứng với phím được bấm, nhưng tín hiệu ra có pha ngược với tín hiệu từ remote. Khi không có sóng tới, tín hiệu ra trên chân DATA ở mức cao.

1.3 GIẢI THUẬT VÀ LẬP TRÌNH GIẢI MÃ

1.3.1 Mã hoá remote hồng ngoại

Trang web <http://lirc.sourceforge.net/remotes> trình bày code của tất cả các loại remote. Qua đó rút ra kết luận sau: có 2 loại mã hóa khác nhau được tích hợp trên remote.

a. Loại điều chế độ rộng xung thấp là loại remote có bit 0 và bit 1 khác nhau ở độ rộng xung thấp, điển hình là điều khiển sony: Sony CD player RM-470 có số bit dữ liệu là 7 bit

Bảng 1.1 Bảng định nghĩa bit remote Sony CD player RM-470

	Xung thấp μs	Xung cao μs	
HEADER	2412	588	Bit khởi động
ONE	612	588	Bit 1
ZERO	1210	588	Bit 0
ptrail	1210		

Khi ấn 1 nút trên điều khiển thì sẽ có 1 xung khởi động được truyền đi với độ dài xung thấp 2412 μs và xung cao 588 μs (để đánh thức MCU), tiếp đó là 7 bit dữ liệu, bit 1 có 612 μs xung thấp và 588 μs xung cao, bit 0 có 1210 μs xung thấp và 588 μs xung cao. Cuối cùng là bit stop có 1210 μs xung thấp và xung cao dài hơn 2000 μs . Nếu giữ nút trên remote thì quá trình truyền mã lệnh cứ lặp lại như thế mãi cho đến khi nhả nút.

Bit 0 và bit 1 có phần xung cao bằng nhau (588 μs) để làm mốc đo phần xung thấp phân biệt giữa bit 0 và 1. Sóng mang của remote có tần số 38 KHz hoặc 40 KHz. Khi không nhấn nút trên remote tức là mắt thu không nhận tín hiệu gì thì chân ra của mắt thu hồng ngoại có mức logic 1. Trạng thái này gọi là trạng thái rỗi.

b. Loại điều chế độ rộng xung cao là loại remote có bit 0 và bit 1 khác nhau ở độ rộng xung cao, điển hình là điều khiển Samsung (đây là kiểu mã hóa phổ biến nhất). Ví dụ Samsung 00025G có 16 bit dữ liệu với các bit được định nghĩa ở bảng 2.

Bảng 1.2 Bảng định nghĩa bit remote Samsung 00025G

	Xung thấp μs	Xung cao μs	
HEADER	8985	4387	Bit khởi động

ONE	599	1626	Bit 1
ZERO	599	506	Bit 0
ptrail	599		

1.3.2 Sự giống nhau, khác nhau trong việc mã hoá các remote:

a. Sự khác nhau

- Số lượng bit dữ liệu được truyền đi khác nhau: có loại 7 bit (sony), loại 8 bit, 12 bit, 16 bit, 18 bit, 32 bit. Cùng 1 hãng điện tử ví dụ sony thì số bit cũng có thể khác nhau, điều khiển tivi sony có 7 bit, còn dàn âm thanh sony là 16 bit. Mỗi bit sẽ mã hóa được 2 trạng thái 0 và 1, vậy n bit sẽ mã hóa được 2^n trạng thái khác nhau, có nghĩa là với giao thức 7 bit thì có thể có $2^7 = 128$ lệnh tương ứng 128 nút trên remote, với giao thức 16 bit thì có thể có $2^{16} = 65\,536$ lệnh tương ứng 65 536 nút trên remote. Tất nhiên nếu sử dụng giao thức có số lượng bit nhiều thì khả năng bị trùng phím với điều khiển khác là rất ít nhưng cũng không cần thiết lắm, trong khi điều này lại làm giảm tuổi thọ pin remote.

- Xung khởi động khác nhau

- Phương thức mã hóa khác nhau: điều chế độ rộng xung thấp hoặc cao

b. Giống nhau

- Mỗi bit khởi động, dừng, 0, 1 đều có phần xung thấp, cao.

- Bit khởi động có phần xung thấp bao giờ cũng $> 2000\mu s$

- Bit 1 có tổng độ dài xung thấp và cao $< 1500\mu s$

- Bit 0 có tổng độ dài xung thấp và cao $> 1500\mu s$

- Bit dừng có xung cao $> 2000\mu s$

- Điều khiển nào có độ dài dữ liệu > 16 bit thì phần dữ liệu đầu tiên là giống nhau, 16 bit còn lại khác nhau.

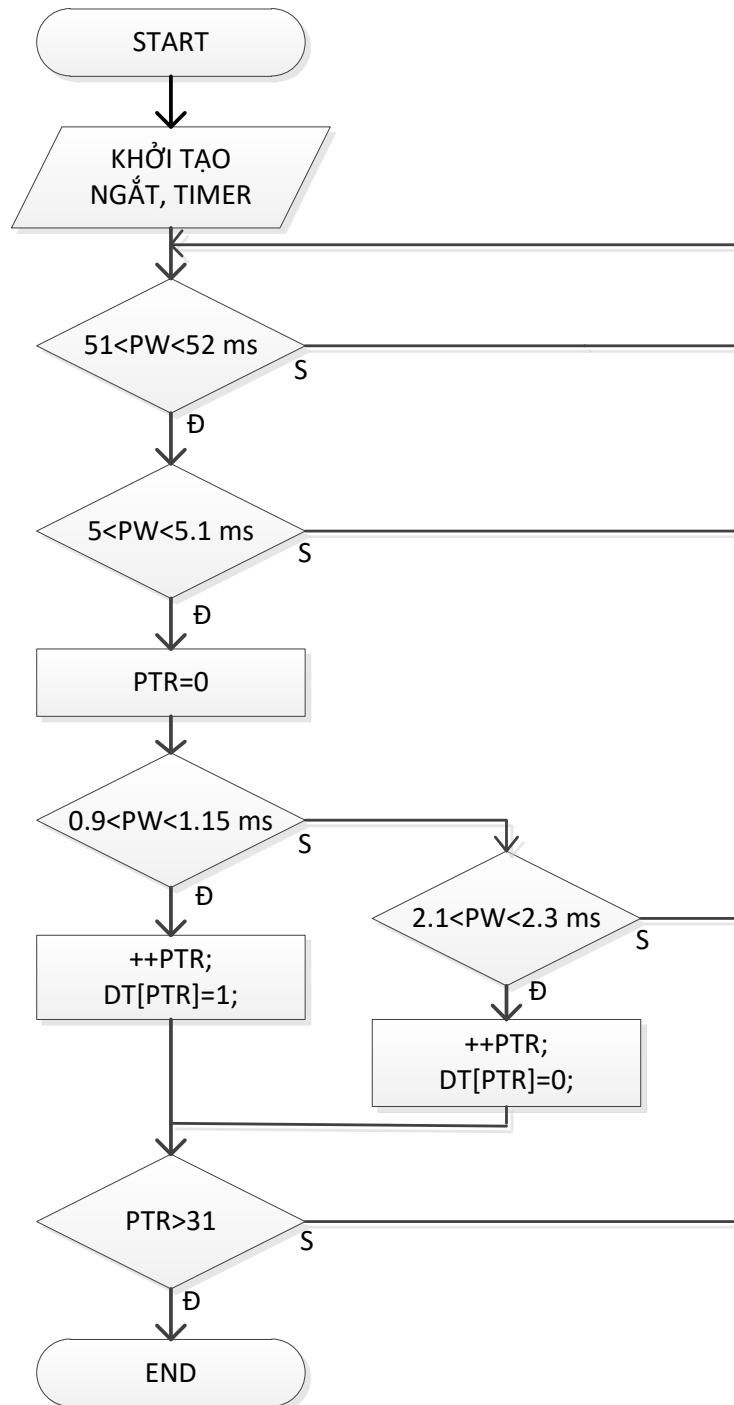
- Bit có giá trị cao nhất được truyền đi trước tiên, bit có giá trị thấp truyền sau cùng

1.3.3 Thuật toán giải mã

Nhóm tác giả đã đo kiểm dạng sóng tín hiệu tại chân ra của mắt thu hồng ngoại, sau phân tích và quyết định sử dụng capture cả cạnh lên và cạnh xuống của vi điều khiển để giải mã thành các bit nhị phân. Nhóm tác giả sử dụng timer và ngắt cạnh lên và cạnh xuống. Tức là mỗi khi gặp cạnh lên hay cạnh xuống thì sẽ ngắt và mỗi khi

ngắt thì cho timer chạy, từ đó dựa vào thời gian timer (thời gian xung cao, xung thấp) để giải mã thành các bit nhị phân.

Hình 3 là lưu đồ chương trình giải mã với các chữ viết tắt: DT là data, PTR là con trỏ nhận dữ liệu, PW là độ rộng xung. Sau khi giải mã xong dữ liệu sẽ được lưu trong mảng DT[] với 32 phần tử tương ứng với 32 bit dữ liệu được gửi từ remote.



Hình 1.3 Lưu đồ giải thuật

Các bước thực hiện giải mã:

- Bước 1: Đo độ rộng xung cao nếu $51000 < \text{độ rộng} < 52000 \mu\text{s}$. Nếu đúng thì nhảy sang bước 2. Nếu sai quay lại bước 1.
- Bước 2: Chờ HEADER. Đo độ rộng xung cao nếu $5000 < \text{độ rộng} < 5100 \mu\text{s}$. Nếu đúng cho con trỏ nhận dữ liệu =0 và nhảy sang bước 3. Nếu sai quay lại bước 1.
- Bước 3: Nhận dữ liệu:
 - + Nếu $900 < \text{độ rộng} < 1150 \mu\text{s}$ thì tăng con trỏ và lưu nhận bit 1
 - + Ngược lại nếu $2100 < \text{độ rộng} < 2300 \mu\text{s}$ thì tăng con trỏ và lưu nhận bit 0
 - + Ngược lại quay lại bước 1 và cho con trỏ nhận dữ liệu =0
 - + Nếu con trỏ nhận dữ liệu > 31 thì cho con trỏ nhận dữ liệu =0, cho cờ nhận dữ liệu =1 và nhảy sang bước 4
- Bước 4: Nếu cờ nhận dữ liệu sai thì quay lại bước 1. Nếu đúng thì xử lý dữ liệu thu được và chờ phím nhấn từ remote tivi.

Cuối cùng có 32 bit dữ liệu giải mã. Nhóm đã dùng thuật toán trên viết chương trình cho STM8 đọc mã từ remote, sau đó sẽ thực hiện: hiển thị mã đọc được và bật tắt LED.

1.3.4 Giải mã remote tivi samsung BN59-00891a (21 inch CRT)

a. Tín hiệu phát ra từ remote tivi Samsung bn59-00891a

Bảng 1.3 Bảng định nghĩa bit remote Samsung bn59-00891a

	Xung thấp (μs)	Xung cao (μs)	
START	4604	443	Bit khởi động
ONE	618	544	Bit 1
ZERO	618	1609	Bit 0
ptrail	618		

Khung truyền tương ứng với các nút trên remote:

16 bit phát đi đầu | 16 bit cuối cùng | Key

00011111 00011111 01110111 10001000 0

00011111 00011111 11011111 00100000 1

00011111 00011111 01011111 10100000 2

00011111 00011111 10011111 01100000 3

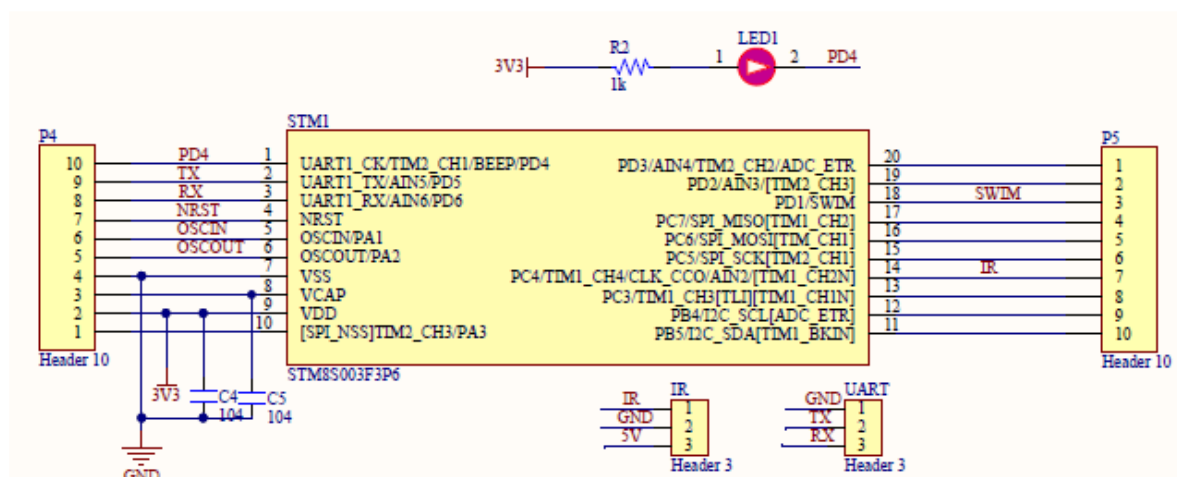
00011111 00011111 11101111 00010000 4

00011111 00011111 01101111 10010000 5

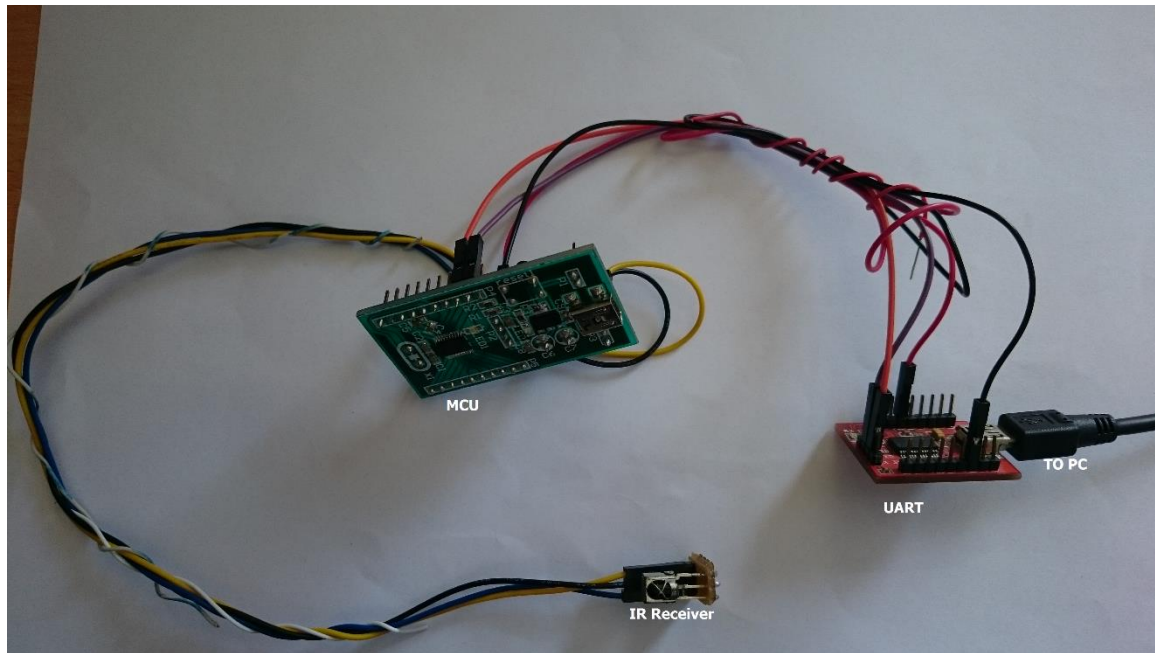
00011111 00011111 10101111 01010000 6
00011111 00011111 11001111 00110000 7
00011111 00011111 01001111 10110000 8
00011111 00011111 10001111 01110000 9
00011111 00011111 11101001 00010110 enter
00011111 00011111 01001011 10110100 exit
00011111 00011111 00000111 11111000 info
00011111 00011111 11110111 00001000 p-
00011111 00011111 10110111 01001000 p+
00011111 00011111 10100111 01011000 menu
00011111 00011111 00001111 11110000 mute
00011111 00011111 10111111 01000000 power
00011111 00011111 00110111 11001000 pre CH
00011111 00011111 01111111 10000000 source
00011111 00011111 00101101 11010010 tool
00011111 00011111 00101111 11010000 vol-
00011111 00011111 00011111 11100000 vol+

b. Sơ đồ nguyên lý mạch giải mã

Mạch gồm vi điều khiển STM8S003F3P6 kết nối tới LED, mắt thu hồng ngoại và mạch giao tiếp UART để kết nối với máy vi tính.



Hình 1.4 Sơ đồ nguyên lý



Hình 1.5 Mạch hoàn chỉnh

c. Chương trình giải mã

Nhóm nghiên cứu dựa vào thuật toán ở trên để giải mã tín hiệu nhận được từ mắt thu hồng ngoại. Khi nhấn nút bất kỳ trên remote thì mã của nút nhấn đó sẽ được hiện thị trên màn hình máy tính, đồng thời một LED được kết nối tới một chân vi điều khiển sẽ đảo trạng thái.

Đoạn chương trình giải mã tín hiệu thu được từ mắt thu hồng ngoại:

```
switch(SMRF)
{
    case WAIT_START_BIT:
        if(x_time > 5100 && x_time < 5200 && INPUT_RF)
        {
            SMRF = WAIT_HEADER;
        }
        break;
    case WAIT_HEADER:
        if(x_time > 500 && x_time < 510 && INPUT_RF);
        {
            SMRF = GET_DATA; Point_RF_Data = 0;
        }
        break;
    case GET_DATA:
        if(x_time > 90 && x_time < 115)
        {
            Point_RF_Data++; Buff_RF[Point_RF_Data] = 1;
        }

        else if(x_time > 210 && x_time < 230)
        {
            Point_RF_Data++; Buff_RF[Point_RF_Data] = 0;
        }
        else
        {
            SMRF = WAIT_START_BIT; Point_RF_Data = 0;
        }
        if(Point_RF_Data > 31)
        {
            Get_Data_OK = TRUE; SMRF = FINISH;
            Point_RF_Data = 0;
        }
        break;
    case FINISH:
        if(!Get_Data_OK) SMRF = WAIT_START_BIT;
        break;
    default: break;
}
x_time=0;
```

Hình 1.6 Chương trình giải mã

Đoạn chương trình điều khiển LED và hiện thị mã của nút nhấn lên màn hình máy vi tính:

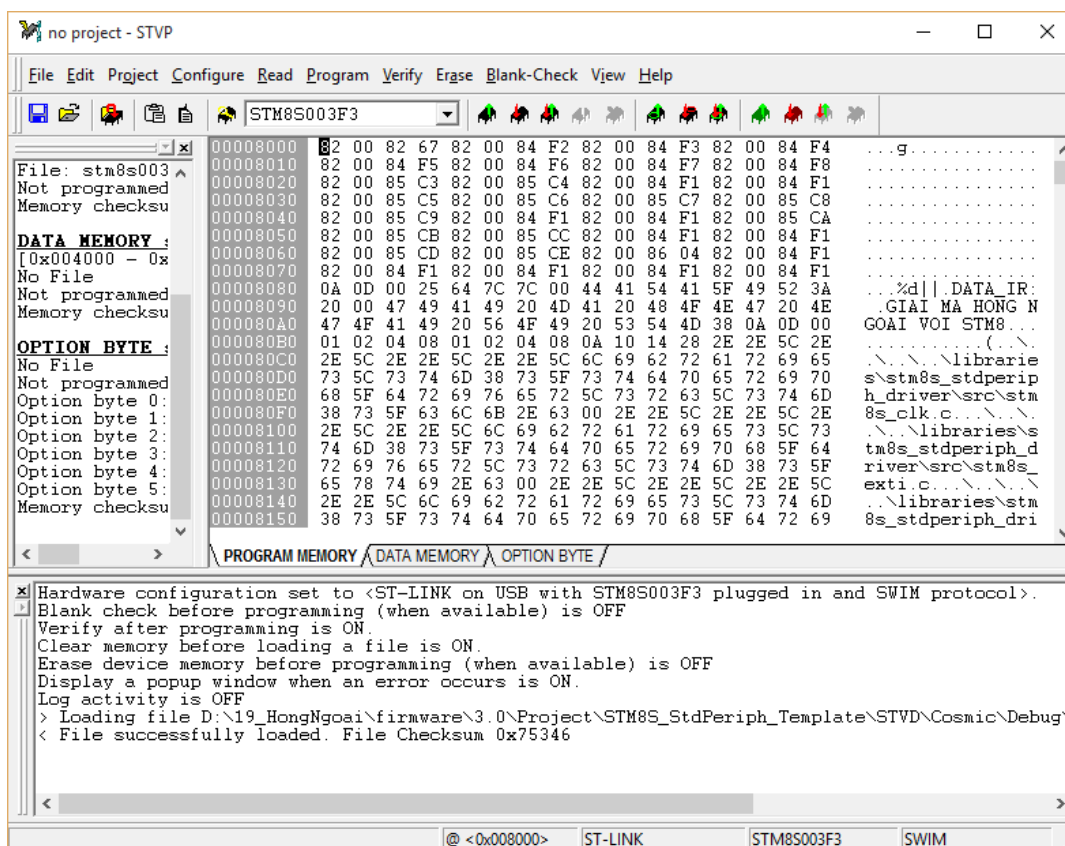
```
delay10us(50000);
GPIO_WriteReverse(GPIOC, GPIO_PIN_3);
if(Get_Data_OK)
{
    disableInterrupts();
    Get_Data_OK = FALSE;
    Data_2_Number();
}
```

```

printf("DATA_IR: ");
for (i=0;i<4;i++)
{
    printf("%d||",DATA_IR[i]);
}
printf("\n\r");
GPIO_WriteReverse(GPIOD, GPIO_PIN_4);
}

```

Biên dịch chương trình và nạp cho vi điều khiển.



Hình 1.7 Nạp cho vi điều khiển

1.4 KẾT QUẢ

Sau khi hoàn thành mạch, nhóm nghiên cứu tiến hành thực nghiệm để kiểm tra:

Bước 1. Cấp nguồn cho mạch

Bước 2. Nhấn nút trên remote tivi

Bước 3. Quan sát trạng thái LED và kiểm tra mã đọc được

Hình dưới là mã đọc được trên máy tính khi nhấn số 1, số 2 trên remote. So với bảng mã hoá nút số 1, số 2 ở trên thì kết quả hoàn toàn đúng. Để dễ dàng tác giả đã gom 8 bit nhị phân thành 1 số thập phân. Như vậy 32 bit nhị phân sẽ có 4 số thập phân. 00011111 00011111 11011111 00100000 đây là khung truyền khi phím 1 trên remote được nhấn, so với kết quả thu được là 31||31||223||32. Nếu đổi kết quả thu ra số nhị phân:

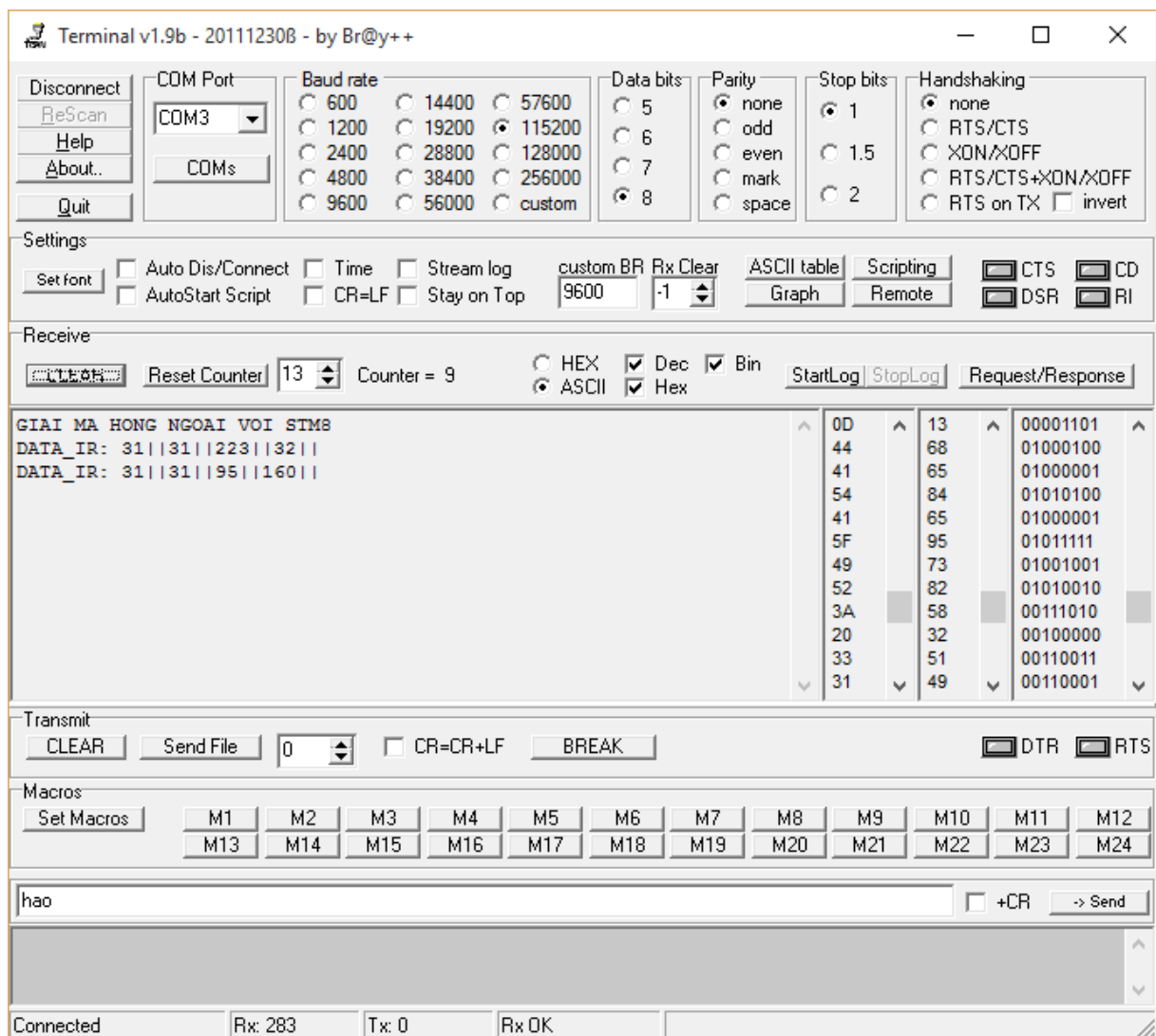
$$(31)_{10} = (00011111)_2$$

$$(31)_{10} = (00011111)_2$$

$$(223)_{10} = (11011111)_2$$

$$(32)_{10} = (00100000)_2$$

thì ta được chuỗi giống hoàn toàn như trên. Tiến hành kiểm tra các nút nhấn còn lại thì thu được kết quả hoàn toàn chính xác.



Hình 1.8 Hiện thị mã nút nhấn trên remote

1.5 KẾT LUẬN

Chương trình dựa vào thuật toán trên để giải mã tín hiệu thu được từ mắt thu hồng ngoại hoạt động tốt. Thuật toán đáp ứng được yêu cầu đề ra là đọc được mã của từng nút nhấn trên remote tivi samsung. Đây là cơ sở để có thể thiết kế các mạch điều khiển thiết bị điện tử từ xa bằng remote tivi. Hướng phát triển của đề tài tiến hành tìm cách lưu mã các nút nhấn của các loại remote khác nhau và phát ra mã đã thu được. Từ đó thiết kế ra remote đa năng thay thế tất cả các remote điều khiển tivi, quạt, đầu DVD...

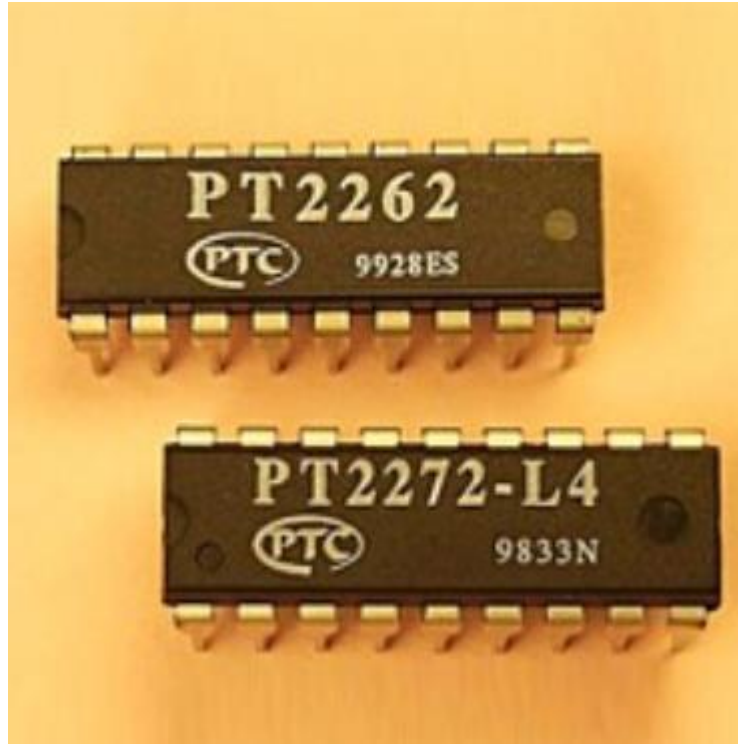
1.6 CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1

1. Thiết kế hệ thống điều khiển thiết bị điện như đèn, quạt bằng remote TV
2. Vẽ sơ đồ nguyên lý mạch giải mã tín hiệu hồng ngoại remote TV

CHƯƠNG 2. ĐIỀU KHIỂN TỪ XA BẰNG TẦN SỐ VÔ TUYẾN

2.1 REMOTE RF

PT2262/2272 là sản phẩm của Princeton Technology sử dụng trong các ứng dụng điều khiển từ xa.

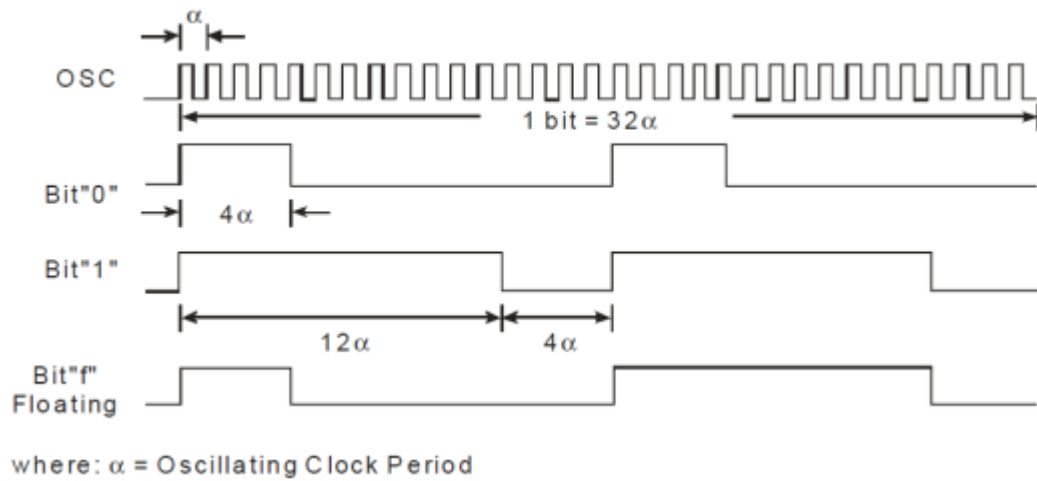


Hình 2.1 Cặp chip thu phát PT2262/2272

Remote phát sử dụng PT2262

Biểu diễn Bit code mã hóa

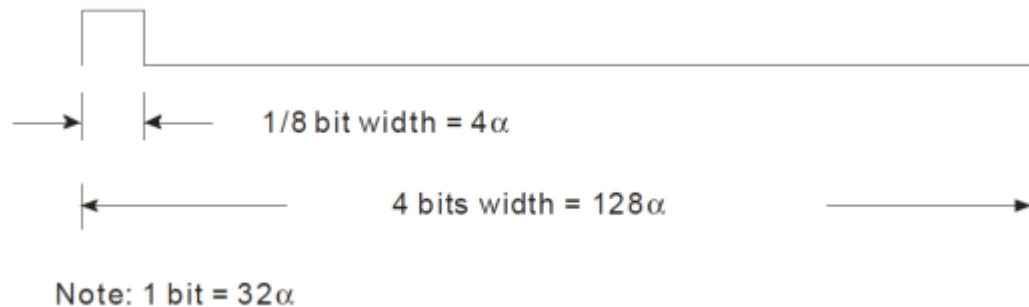
Cơ bản PT2262 sử dụng mỗi Bit gồm 3 trạng thái 0,1 và f. Mỗi trạng thái sẽ có 1 kiểu mã hóa Bit code khác nhau. Mỗi Bit code mã hóa chứa trong 32 chu kì tần số mã hóa của OSC(32α).



Hình 2.3 Waveform của các Bit code

Bit đồng bộ (Synchronous Bit)

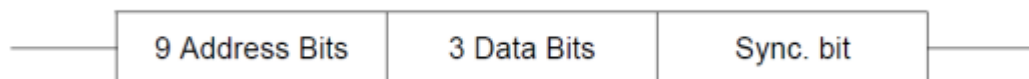
Là Bit được thêm vào trong 1 khung truyền để giúp đồng bộ hóa quá trình mã hóa/giải mã. Độ dài của Sync Bit là bằng 4 lần độ dài 1 bit địa chỉ/dữ liệu tức 128α .



Hình 2.4 Waveform của Sync Bit

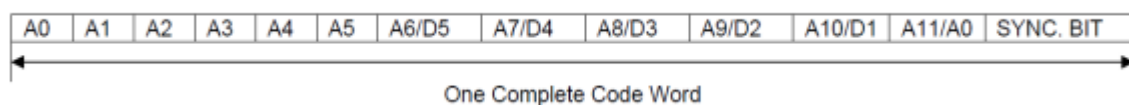
Từ mã (Code Word)

♣ Cấu trúc việc truyền các Bit địa chỉ/ dữ liệu và Bit đồng bộ của PT2262 được xác định như sau:



Hình 2.5 Cấu trúc 1 từ mã 9 bits địa chỉ/3 bits dữ liệu

Với dòng PT2262 sử dụng 6 bits địa chỉ và 6 bits dữ liệu thì từ mã này được biểu diễn:



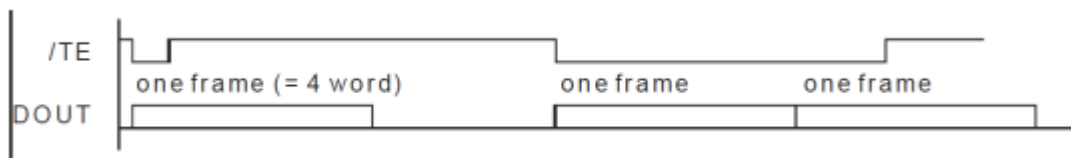
Như vậy với các dòng PT2262 với số đường địa chỉ/dữ liệu khác nhau ta có các Code Word khác nhau theo dạng sau:

0 Data:	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	Sync Bit
1 Data:	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	D0	Sync Bit
2 Data:	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	D1	D0	Sync.Bit
3 Data:	A0	A1	A2	A3	A4	A5	A6	A7	A8	D2	D1	D0	Sync.Bit
4 Data:	A0	A1	A2	A3	A4	A5	A6	A7	D3	D2	D1	D0	Sync.Bit
5 Data:	A0	A1	A2	A3	A4	A5	A6	D4	D3	D2	D1	D0	Sync.Bit
6 Data:	A0	A1	A2	A3	A4	A5	D5	D4	D3	D2	D1	D0	Sync.Bit

Hình 2.6 Các dòng PT2262 với số lượng bit địa chỉ/dữ liệu khác nhau

Khung truyền(Code Frame)

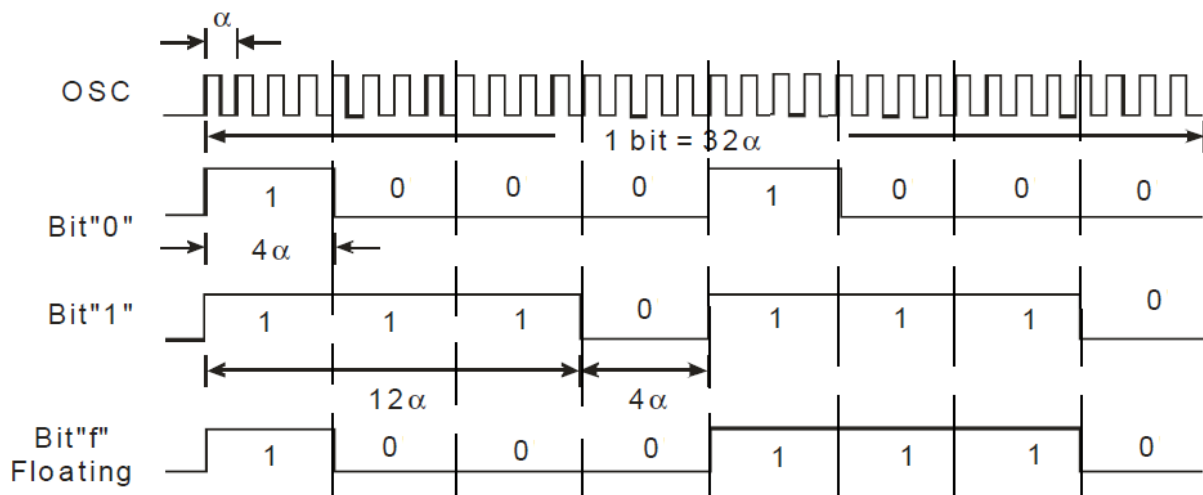
Một khung truyền bao gồm 4 từ mã (Code Word) phát liên tục nhau ra chân DOUT của PT2262 khi chân /TE là tích cực(/TE tích cực mức thấp).



Hình 2.7 Cấu trúc 1 khung truyền được gửi

Giải thuật

Mỗi bit sử dụng trong PT2262 có độ dài 32α thay thế bằng 8 bits 1,0 liên tục(1 Byte). Theo biểu diễn này ta có:



where: α = Oscillating Clock Period

Hình 2.8 Timing của phần mã hóa các bit

σ Bit 0 được thay thế bằng chuỗi 10001000

σ Bit 1 được thay thế bằng chuỗi 11101110

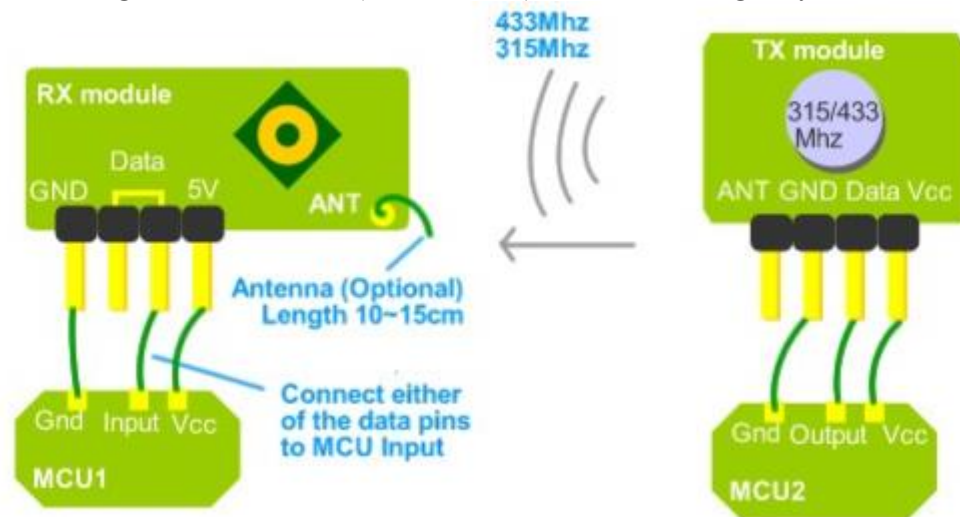
σ Bit f được thay thế bằng chuỗi 10001110

σ Sync Bit được thay thế bằng chuỗi : 10000000|00000000|00000000|00000000

♣ Như vậy các Bit Code khi phát đi sẽ được thay thế bằng chuỗi bit mã hóa nó. Ví dụ 1 từ mã Code Word PT2262 muốn phát là “11110000|1010| SyncBit” (8 bits địa chỉ|4 bits dữ liệu|1 Sync Bit) sẽ được mã hóa thành chuỗi 128 bit là:

11101110.11101110.11101110.11101110.10001000.10001000.10001000.10001000|11101110.10001000.11101110.10001000|11101110.00000000.00000000.00000000

♣ Thực hiện việc gọi 4 lần 1 từ mã(Code Word) để tạo 1 khung truyền.



Hình 2.9 Sử dụng MCU mã hóa/giải mã thay cho PT2262/PT2272

2.2 GIẢI MÃ TÍN HIỆU RF 433MHZ VỚI ARDUINO

2.2.1 Phần cứng:

- Arduino UNO



- 433 MHz RF Remote controlled



- 433 MHz receiver



- Jumper wires



Lưu ý Điều khiển hoạt động ở tần số 433 MHz

2.2.2 Installing the RC Switch Library

Với thư viện RC Switch cung cấp cho ESP8266, ESP32, hoặc Arduino, chúng ta dễ dàng thực hiện điều khiển thiết bị từ xa.

1. Vào <https://github.com/sui77/rc-switch/archive/master.zip> để download thư viện.
2. Giải nén file zip được folder **rc-switch-master**
3. Sửa tên folder thành **rc_switch**
4. Di chuyển folder **rc_switch** tới thư mục libraries của Arduino IDE
5. Sau đó mở Arduino IDE

Mở sketch ReceiveDemo Advanced:

File > Examples > RC_Switch > ReceiveDemo_Advanced

```
/*  
Example for receiving  
  
https://github.com/sui77/rc-switch/  
  
If you want to visualize a telegram copy the raw data and  
paste it into http://test.sui.li/oszi/  
*/  
  
#include <RCSwitch.h>  
  
RCSwitch mySwitch = RCSwitch();  
  
void setup() {  
  Serial.begin(9600);  
  mySwitch.enableReceive(0); // Receiver on interrupt 0 => that is pin #2  
}  
  
void loop() {  
  if (mySwitch.available()) {  
    output(mySwitch.getReceivedValue(), mySwitch.getReceivedBitlength(),  
mySwitch.getReceivedDelay(),  
mySwitch.getReceivedRawdata(),mySwitch.getReceivedProtocol());  
    mySwitch.resetAvailable();  
  }  
}
```

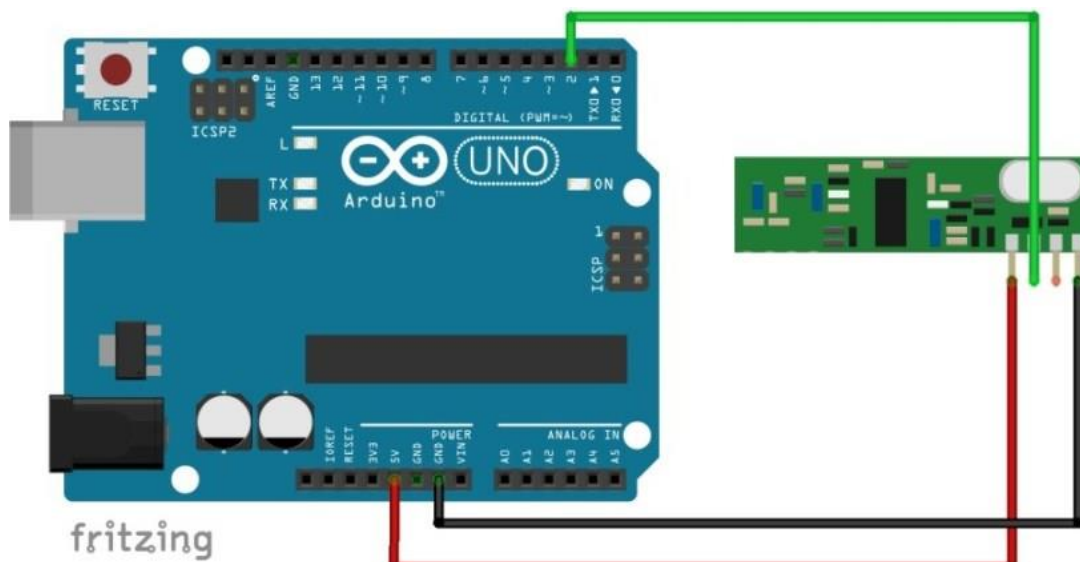
2.2.3 Nạp chương trình cho arduino

Kết nối board arduino với máy tính qua cáp USB và làm theo hướng dẫn sau:

1. Go to the **Tools** tab
2. Select **Arduino UNO** board
3. Select the **COM** port
4. Press the **Upload** button.

2.2.4 Sơ đồ kết nối

Kết nối module thu với arduino như hình



Mở Arduino IDE serial monitor và bắt đầu nhấn các nút nhấn. Bạn sẽ thấy mã nhị phân của các nút nhấn (it's highlighted in red):



Save your binary codes for each button press (you can also use the Decimal or Tri-State codes):

- Button 1 = (24Bit) Binary: 000101010101000101010101
- Button 2 = (24Bit) Binary: 000101010101000101010100
- Button 3 = (24Bit) Binary: 000101010101010001010101
- Button 4 = (24Bit) Binary: 000101010101010001010100

CHƯƠNG 3. ĐIỀU KHIỂN TỪ XA BẰNG ĐIỆN THOẠI QUA BLUETOOTH

3.1 MODULE BLUETOOTH HC-06

Module bluetooth HC-06 ra chân hoàn chỉnh giúp dễ dàng kết nối để thực hiện các thí nghiệm , module được thiết kế để cho hoạt động từ mức điện áp 3V3 => 5V . Khi kết nối với máy tính , HC-06 được sử dụng như 1 cổng COM ảo , việc truyền nhận với COM ảo sẽ giống như truyền nhận dữ liệu trực tiếp với UART trên module.

Lưu ý là khi thay đổi Baudrate cho COM ảo không làm thay đổi baudrate của AURT , baudrate UART chỉ có thể thay đổi bằng AT command trên module . Module HC-06 được setup mặc định là Slave không thể thay đổi được nên chỉ có thể giao tiếp với các thiết bị bluetooth ở dạng master như Smart phone , HC-06 master , hai module bluetooth được set là Slave không thể giao tiếp với nhau .



Hình 3.1 Board Bluetooth HC-06 thực tế

➤ **Thông số kỹ thuật :**

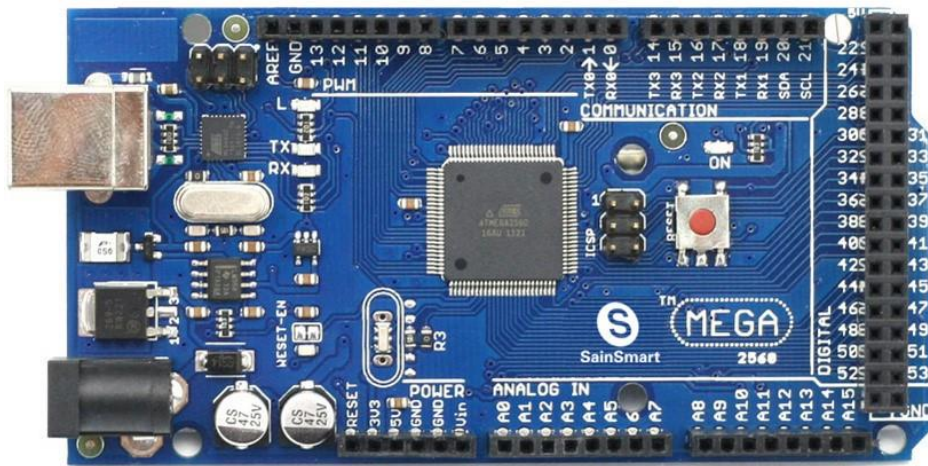
- Sử dụng CSR mainstream bluetooth chip , bluetooth V2.0 protocol standards.
- Điện thế hoạt động của UART 3.3-5V.
- Dòng điện khi hoạt động : khi Pairing 30mA , sau khi pairing hoạt động truyền nhận bình thường 8mA.
- Baudrate UART có thể chọn được : 1200 , 2400 , 4800 , 9600 , 19200 , 38400 , 57600 , 115200 .

- Kích thước của module chính : 28 mm x 15 mm x 2.35mm.
- Dải tần sóng hoạt động : 2.4GHz.
- Nhiệt độ lưu trữ : -40 to 85°C

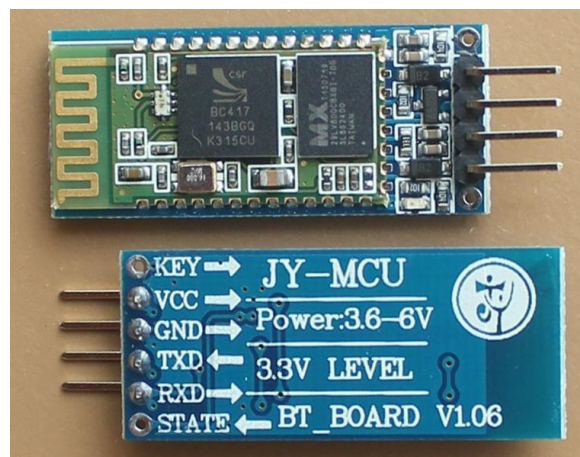
➤ **Thiết lập mặc định :**

- Baud rate : 9600 , N , 8 , 1,
- Pairing code : 1234.

3.2 CÁC THÀNH PHẦN



Board arduino MEGA



Module bluetooth HC 06



Mạch relay 4 kênh 5v

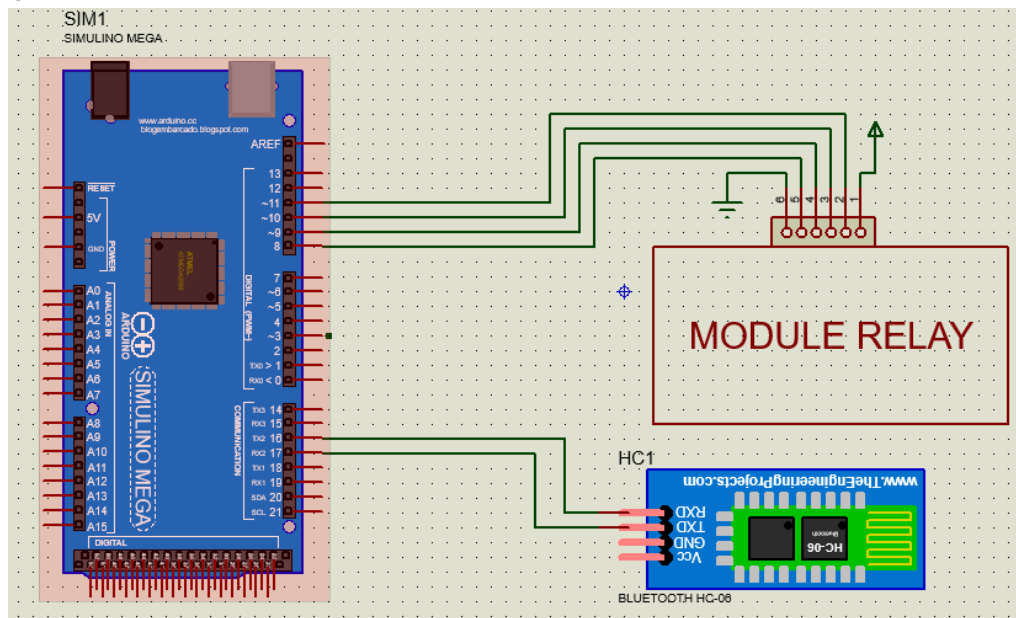
3.3 NỐI DÂY

- ❖ 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



3.4 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);
```

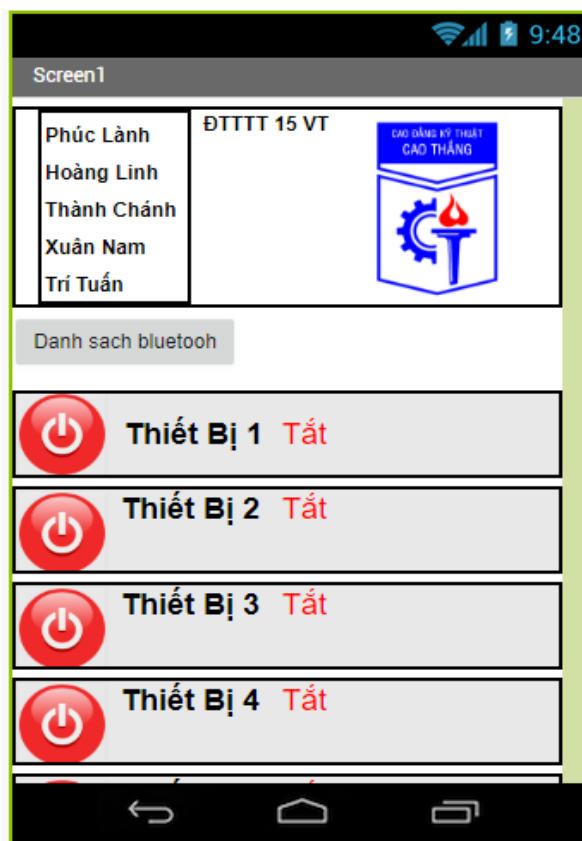
```
digitalWrite(10, LOW);  
digitalWrite(11, LOW);  
break;  
case 'Q':  
digitalWrite(8, HIGH);  
digitalWrite(9, HIGH);  
digitalWrite(10, HIGH);  
digitalWrite(11, HIGH);  
break;  
default:  
break;  
}  
}
```

3.5 ỨNG DỤNG ĐIỀU KHIỂN TRÊN ĐIỆN THOẠI ANDROID

Dùng app inventor để viết app điều khiển

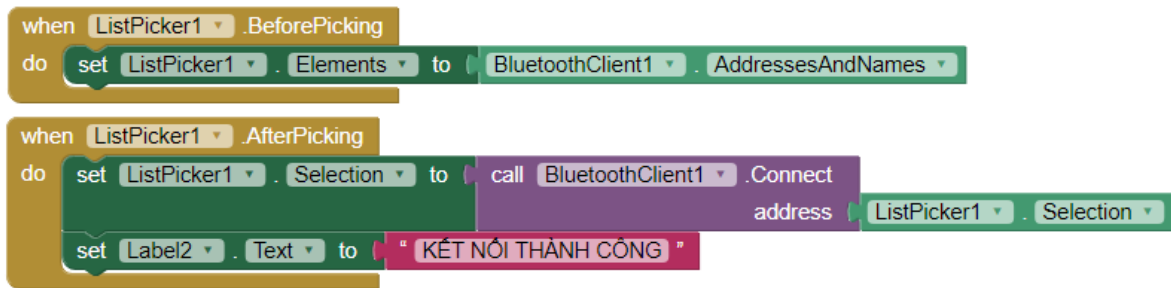
Chi tiết tại : <http://ai2.appinventor.mit.edu>

Giao diện :



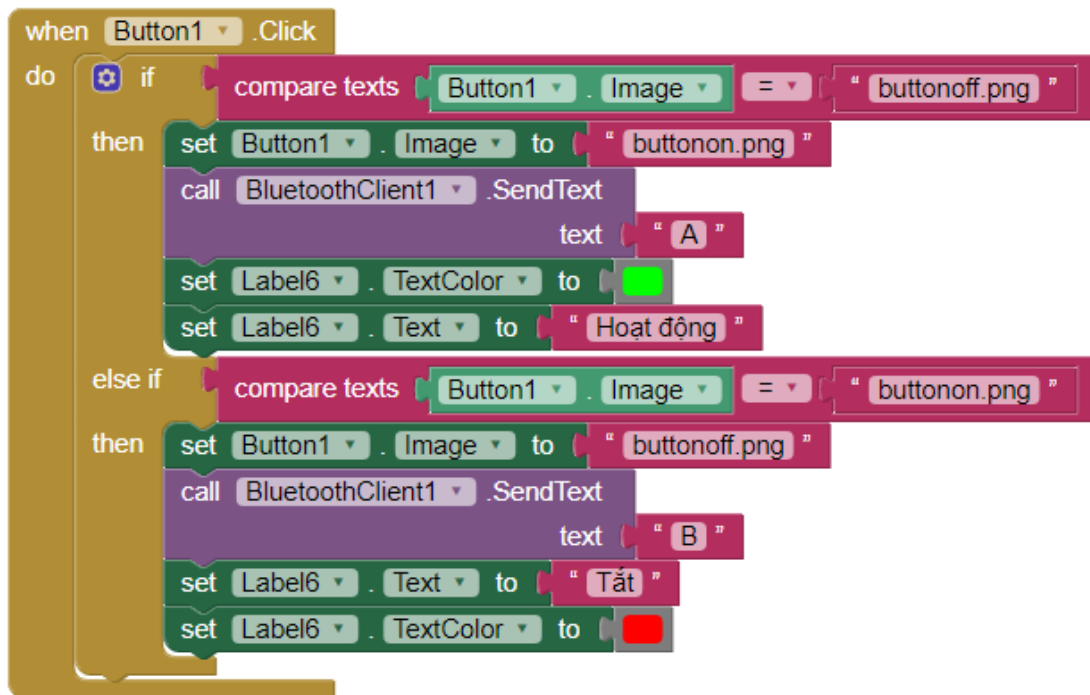
Các khối lệnh cơ bản:

Lệnh kết nối bluetooth:

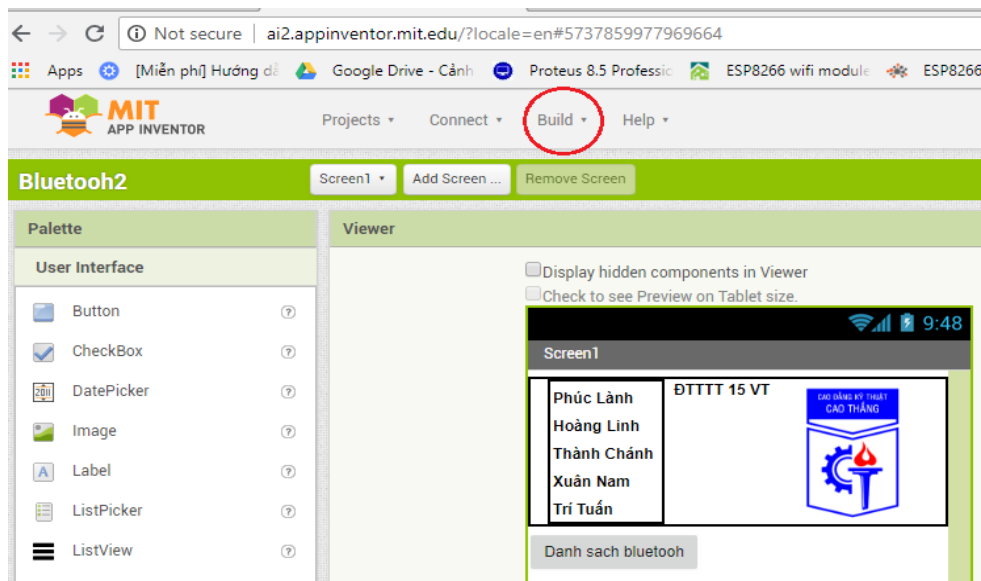


Nút nhấn:

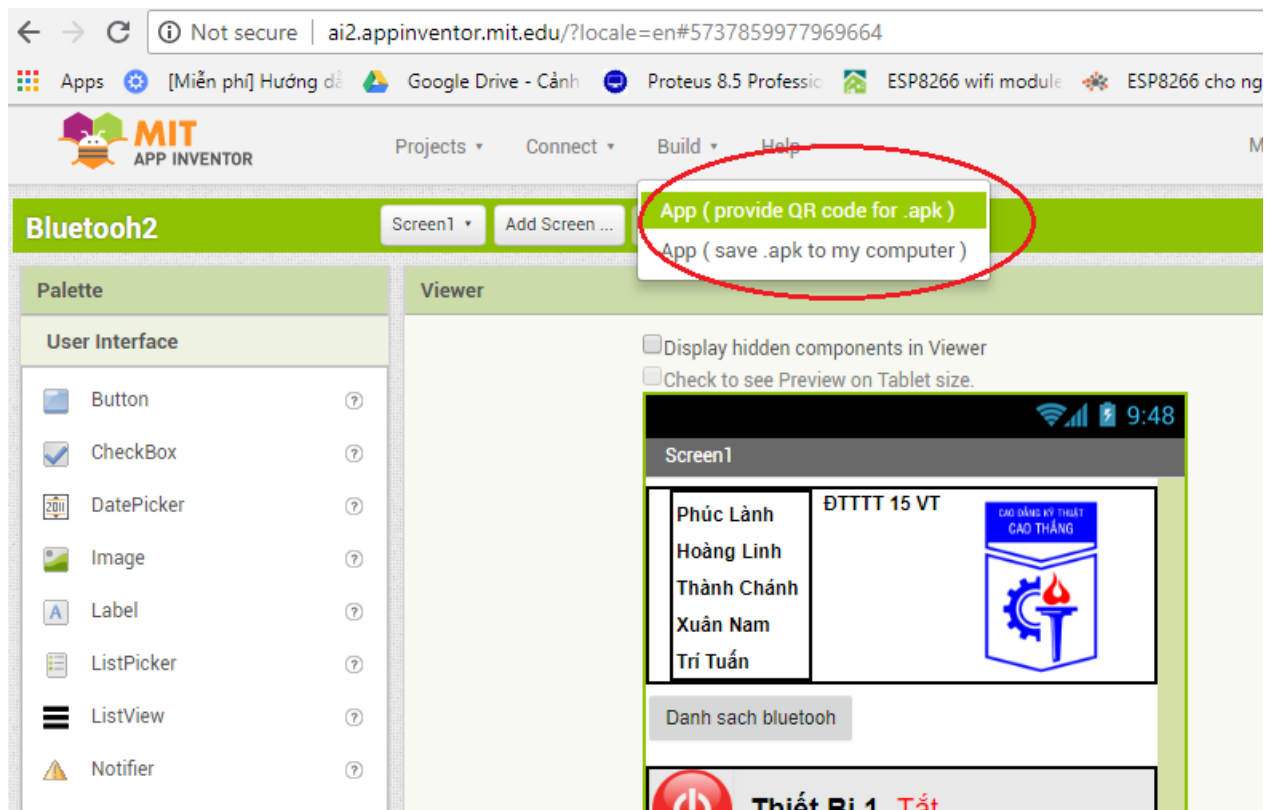
Khối lệnh của 1 nút các nút còn lại tương tự như vậy



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.

CHƯƠNG 4. ĐIỀU KHIỂN TỪ XA QUA INTERNET

4.1 MODULE ESP8266 WEMOS D1

4.1.1 Thông số kĩ thuật

WEMOS D1 R2 là kit phát triển phiên bản mới nhất từ WeMos, kit được thiết kế với hình dáng tương tự Arduino Uno nhưng trung tâm lại là module wifi Soc ESP8266EX được build lại firmware để có thể chạy với chương trình Arduino. Kit thích hợp và dễ dàng thực hiện các ứng dụng thu thập dữ liệu và điều khiển qua Wifi.



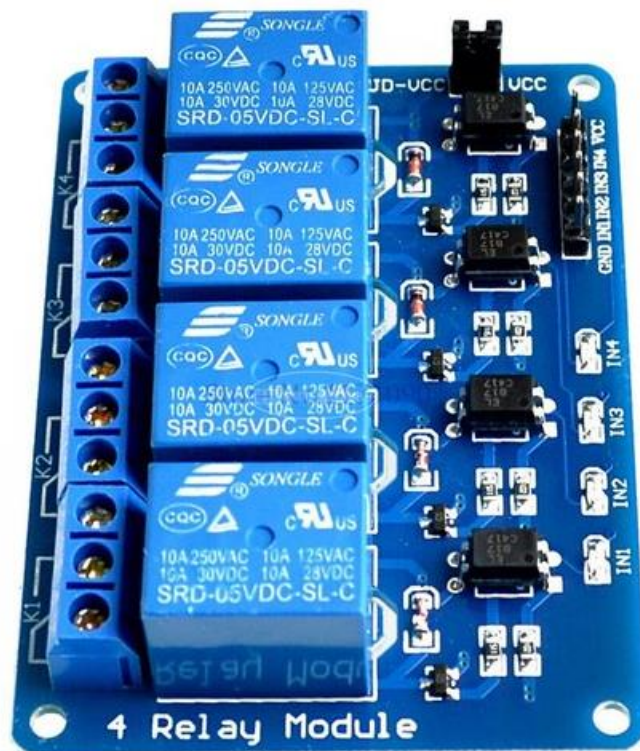
Vi điều khiển	ESP8266EX
Điện áp hoạt động	3V3
I/O Digital Pin	11
Analog Pin	1 (Max input=3V2)
Xung clock	80MHz/160MHz
Flash	4Mb
Khối lượng	25g
Kích thước	68.6mmX53.4mm

4.1.2 Sơ đồ pin digital

Wemos	ESP8266
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3,D15	GPIO0
D4,D14	GPIO2
D5,D13	GPIO14
D6,D12	GPIO12
D7,D11	GPIO13
D8	GPIO15
D9	GPIO3
D10	GPIO1

Lí do mình viết bảng này là để dễ dàng trong việc viết code. Bởi khi viết code bạn muốn xuất tín hiệu HIGH cho pin số 3 trên kit thì ko thể viết "digitalWrite(3,HIGH);" mà phải viết là "digitalWrite(0,HIGH);" (Theo trên bảng thì chân D3 trên kit là chân 0 của ESP). Hay nói cách khác là bạn điều khiển chân trên ESP8266. Và khi bạn code xuất chân D3 mức HIGH thì chân D15 cũng được xuất HIGH, lí do là vì: các chân từ D11=>D15 là các chân "giả" để kit trông giống arduino hơn (ESP chỉ có 11 pin digital), vì vậy các pin D11=>D15 được nối lần lượt với D7=>D3.

4.2 CÁC THÀNH PHẦN



Mạch relay 4 kênh 5v



Module wemos D1

4.3 NỐI DÂY

- ❖ Chúng ta cắm bốn dây tín hiệu của module relay vào bốn chân D1,D2,D3,D4 (lần lượt là GPIO5, GPIO4, GPIO0, GPIO2) trên wemos D1. Sau đó lắp các bóng đèn vào ổ cắm được nối sẵn với các relay.

4.4 THƯ VIỆN

- ❖ **ESP8266WiFi.h** - http://arduino.esp8266.com/stable/package_arduino_index.json
- ❖ **BlynkSimpleEsp8266.h** - <https://github.com/blynkkk/blynk-library>

4.5 CHƯƠNG TRÌNH NẠP VÀO WEMOS D1

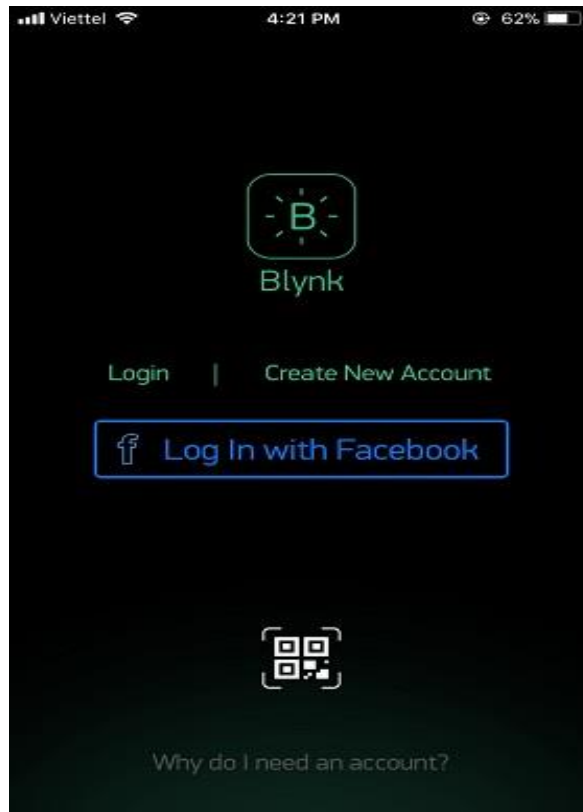
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "YourAuthToken"; //AuthToken copy ở Blynk Project
char ssid[] = "YourNetworkName"; //Tên wifi
char pass[] = "YourPassword"; //Mật khẩu wifi
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}
void loop()
{
}
```



```
Blynk.run();  
}
```

4.6 HƯỚNG DẪN SỬ DỤNG ỨNG DỤNG BLYNK

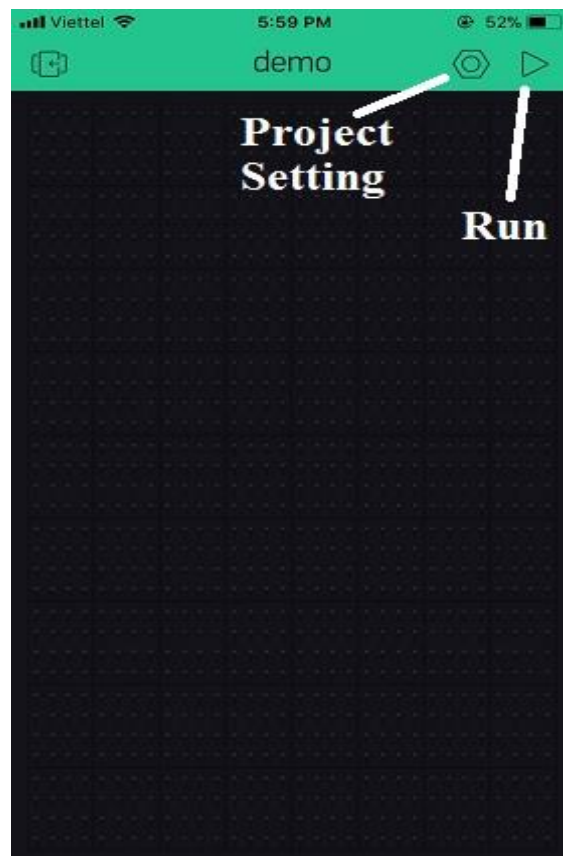
Trên điện thoại sử dụng iOS hoặc Android tải phần mềm Blynk về điện thoại. Sau khi cài đặt đăng ký một Account Blynk, lưu ý đăng ký bằng email thật sẽ giúp ích nhiều cho sau này.



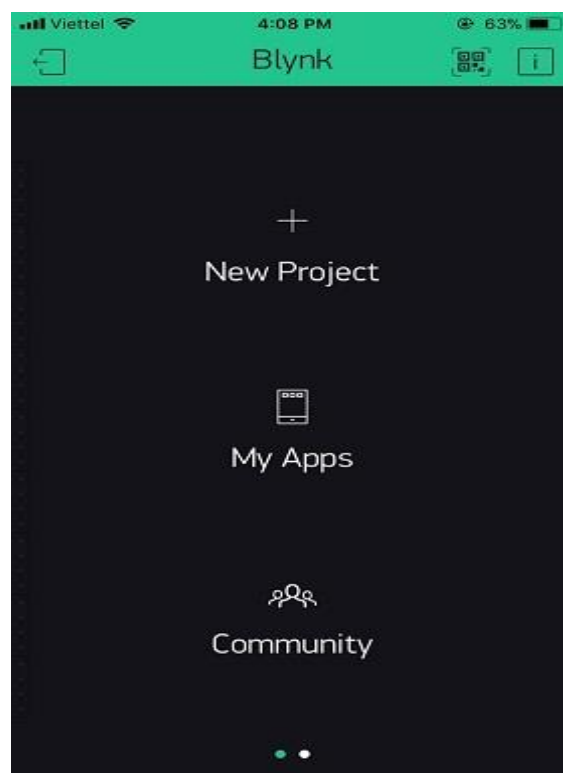
Biểu tượng ứng dụng Blynk trên điện thoại



Sau khi đăng nhập, màn hình hiện ra

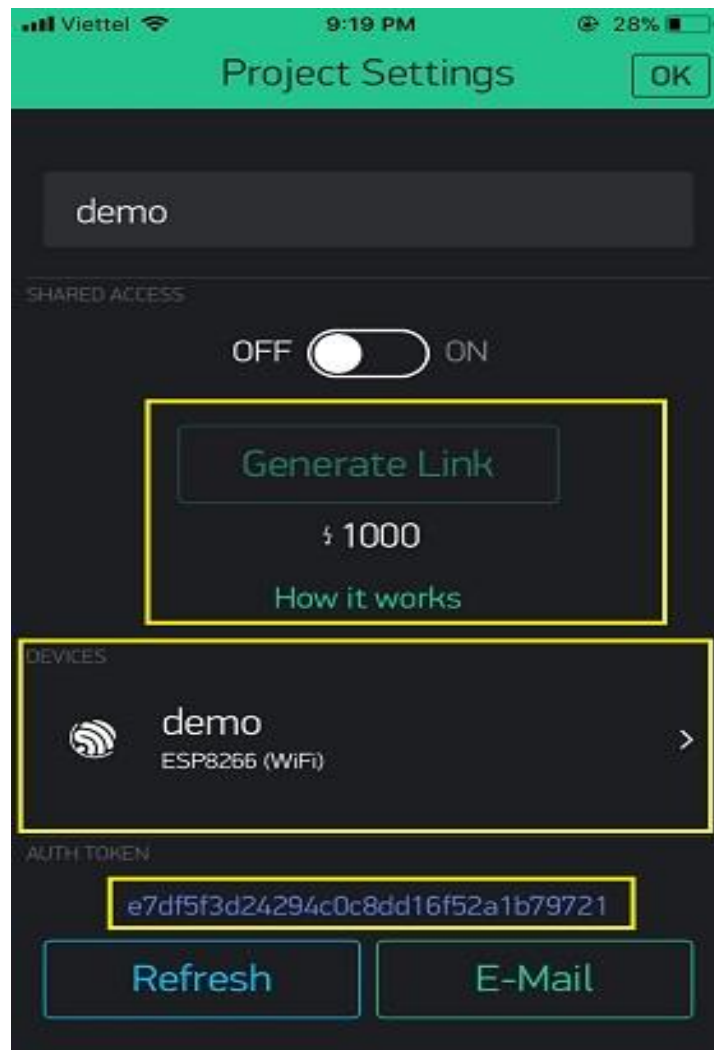


Kéo màn hình sang trái, giao diện tạo Project:



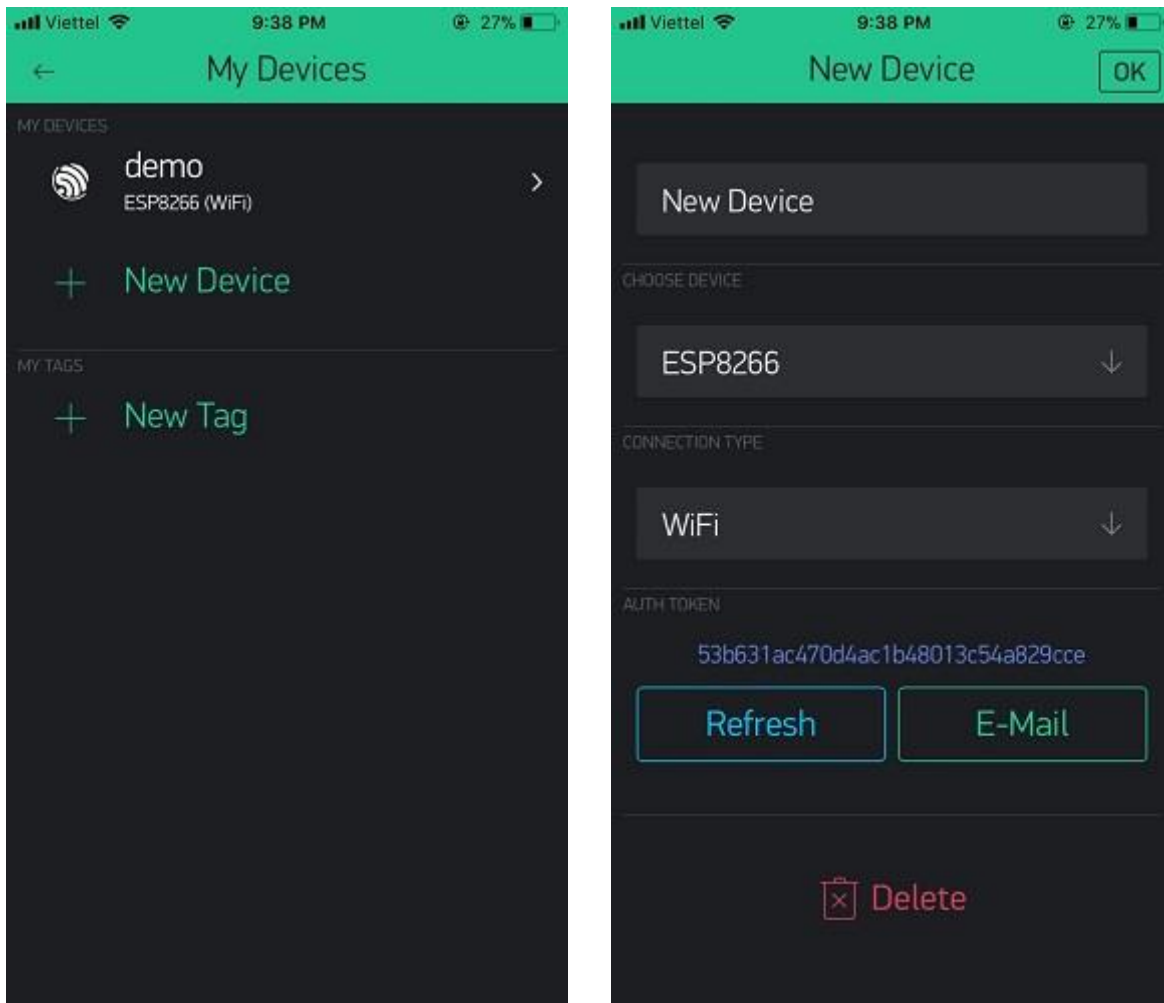
Chọn New Project, thiết lập tên Project, chọn thiết bị. Trong phần này sử dụng ESP8266 vì vậy sẽ chọn ESP8266 và kiểu kết nối là WiFi. Sau khi tạo Project, Blynk

app sẽ gửi Auth Token đến email đăng ký. Auth token được sử dụng để xác thực các thiết bị. Sau khi tạo xong project, vào phần Project Setting:



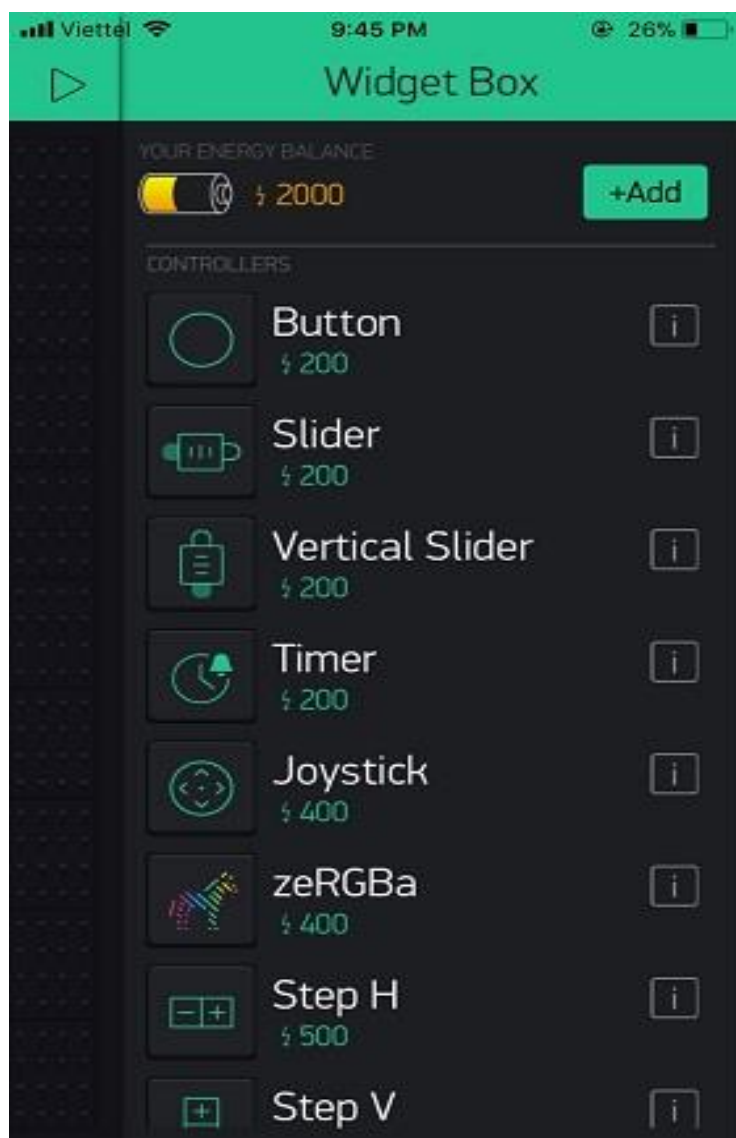
Mỗi account khi đăng ký sử dụng dịch vụ đám mây Blynk sẽ được cấp một số Energy. Như hình trên là 1000 Energy. Với mỗi Widget tạo ra trong Project sẽ tốn một số tài nguyên Energy, nếu người dùng muốn sử dụng thêm thì phải bỏ thêm tiền mua Energy. Blynk có mã nguồn mở, tuy nhiên để duy trì hoạt động và tái phát triển cho Blynk, dịch vụ đám mây Blynk cũng có thu phí theo nhu cầu của người sử dụng.

Khi kích vào Devices có thể thêm bớt các Device. Mỗi Device sẽ có một Auth Token dùng để xác thực thiết bị đến Blynk Server.

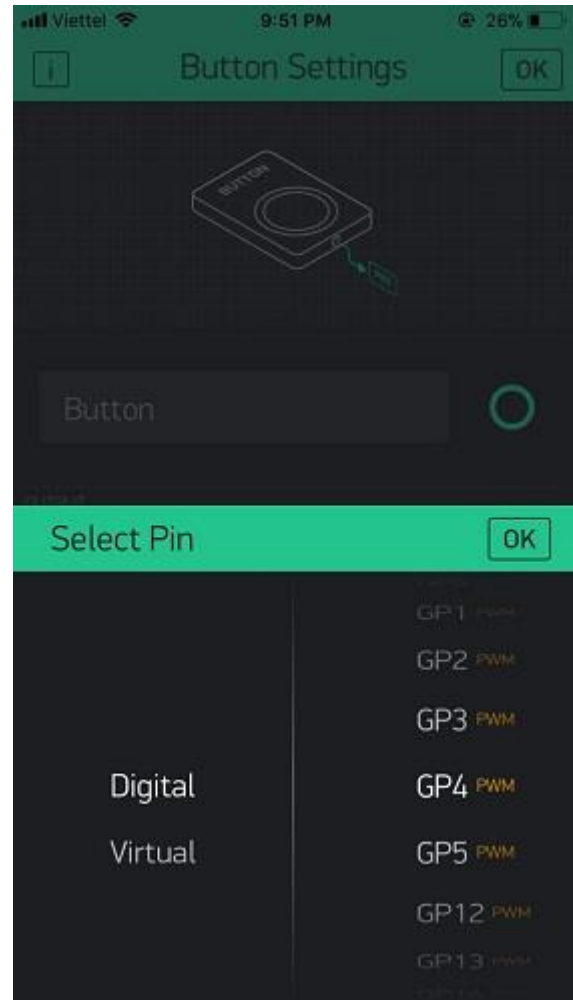
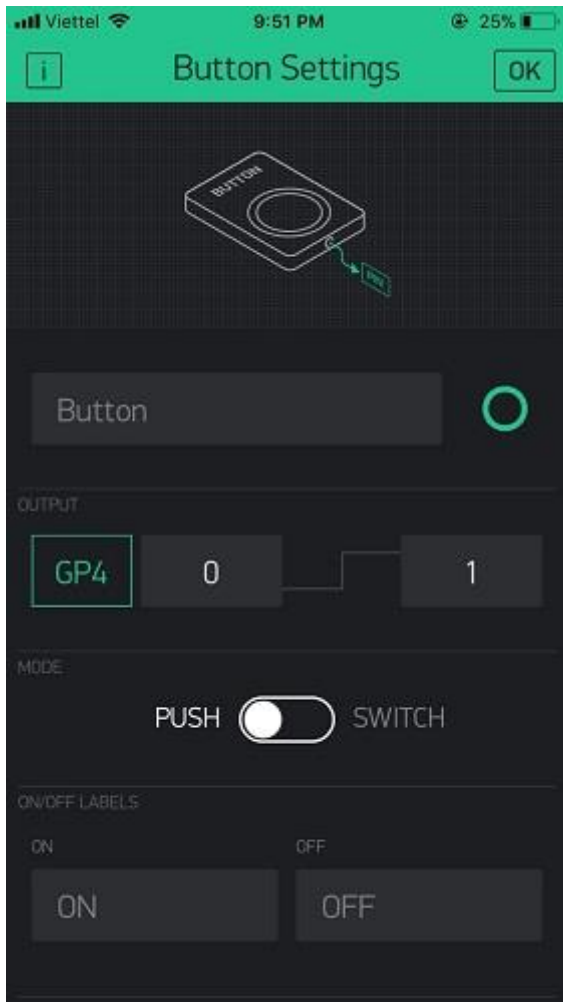


Thêm Widget

Ban đầu bảng vẽ là trống, kích vị trí bất kỳ trên bảng vẽ, hộp các Widget sẽ hiển thị ra. Thêm một Button vào bảng vẽ.



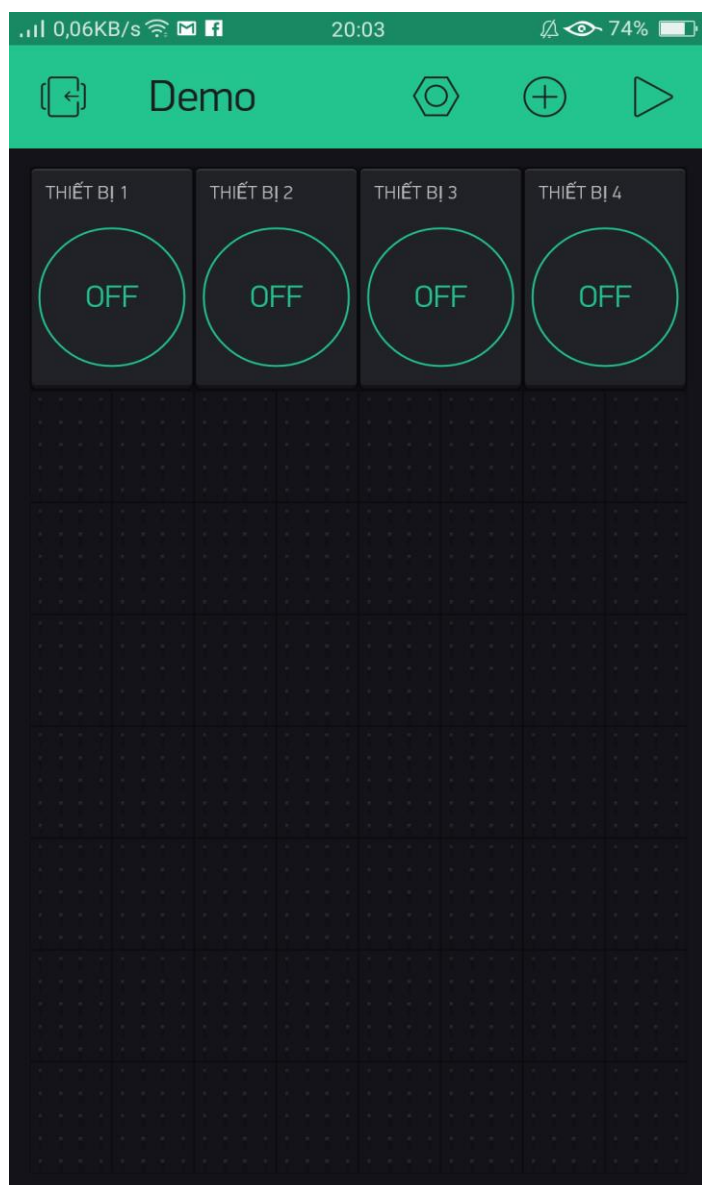
Sau khi thêm Button, có thể kích vào button, giữ và di chuyển button đến vị trí thích hợp. Kích vào Button để thiết lập, trong phần Output thiết lập là Digital và chọn GP5 tương ứng GPIO5 của ESP8266. Tương tự chúng ta tạo thêm các BUTTON khác để điều khiển các relay khác. Tùy vào loại relay chúng ta kích ở mức cao hay thấp mà chúng ta sẽ thiết lập trên nút bấm. Ở đây số 0 là mức thấp còn 1 là mức cao.



Sau khi thiết lập xong, thực hiện Run để bắt đầu hoạt động của Project. Khi sửa đổi Project thì cần STOP chương trình lại và có thể thêm các Widget khác.

4.7 KẾT QUẢ THỰC HIỆN

Sau khi hoàn thành thiết lập 4 nút chúng ta sẽ được giao diện 4 nút điều khiển. Sau khi bấm biểu tượng **Play** chúng ta sẽ có thể điều khiển Relay bằng 4 nút nhấn chúng ta vừa tạo.



CHƯƠNG 5. ĐIỀU KHIỂN TỪ XA BẰNG ĐIỆN THOẠI QUA SMS

5.1 TIN NHẮN SMS

– SMS là từ viết tắt của Short Message Service. Đó là một công nghệ cho phép gửi và nhận các tin nhắn giữa các điện thoại với nhau. SMS xuất hiện đầu tiên ở Châu Âu vào năm 1992. Ở thời điểm đó, nó bao gồm cả các chuẩn về GSM. Một thời gian sau đó, nó phát triển sang công nghệ wireless như CDMA và TDMA. Các chuẩn GSM và SMS có nguồn gốc phát triển bởi ETSI. ETSI là chữ viết tắt của European Telecommunications Standards Institute. Ngày nay thì 3GPP (Third Generation Partnership Project) đang giữ vai trò kiểm soát về sự phát triển và duy trì các chuẩn GSM và SMS.

– Như đã nói ở trên về tên đầy đủ của SMS là Short Message Service, từ cụm từ đó, có thể thấy được là dữ liệu có thể được lưu giữ bởi một tin nhắn SMS là rất giới hạn. Một tin nhắn SMS có thể chứa tối đa là 140 byte (1120 bit) dữ liệu. Vì vậy, một tin nhắn SMS chỉ có thể chứa :

+ 160 kí tự nếu như mã hóa kí tự 7 bit được sử dụng (mã hóa kí tự 7 bit thì phù hợp với mã hóa các kí tự latin chẳng hạn như các kí tự alphabet của tiếng Anh).

+ 70 kí tự nếu như mã hóa kí tự 16 bit Unicode UCS2 được sử dụng (các tin nhắn SMS không chứa các kí tự latin như kí tự chữ Trung Quốc phải sử dụng mã hóa kí tự 16 bit). Tin nhắn SMS dạng text hỗ trợ nhiều ngôn ngữ khác nhau. Nó có thể hoạt động tốt với nhiều ngôn ngữ mà có hỗ trợ mã Unicode , bao gồm cả Arabic, Trung Quốc, Nhật bản và Hàn Quốc. Bên cạnh gửi tin nhắn dạng text thì tin nhắn SMS còn có thể mang các dữ liệu dạng binary. Nó còn cho phép gửi nhạc chuông, hình ảnh cùng nhiều tiện ích khác ... tới một điện thoại khác.

– Một trong những ưu điểm nổi trội của SMS đó là nó được hỗ trợ bởi các điện thoại có sử dụng GSM hoàn toàn. Hầu hết tất cả các tiện ích cộng thêm gồm cả dịch vụ gửi tin nhắn giá rẻ được cung cấp, sử dụng thông qua sóng mang wireless. Không giống như SMS, các công nghệ mobile như WAP và mobile Java thì không được hỗ trợ trên nhiều model điện thoại. Sử dụng tin nhắn SMS ngày càng phát triển và trở lên rộng khắp :

– Các tin nhắn SMS có thể được gửi và đọc tại bất kỳ thời điểm nào. Ngày nay, hầu hết mọi người đều có điện thoại di động của riêng mình và mang nó theo người hầu như cả ngày. Với một điện thoại di động , bạn có thể gửi và đọc các tin nhắn SMS bất

cứ lúc nào bạn muốn, sẽ không gặp khó khăn gì khi bạn đang ở trong văn phòng hay trên xe bus hay ở nhà...

- Tin nhắn SMS có thể được gửi tới các điện thoại mà tắt nguồn.
- Nếu như không chắc cho một cuộc gọi nào đó thì bạn có thể gửi một tin nhắn SMS đến bạn của bạn thậm chí khi người đó tắt nguồn máy điện thoại trong lúc bạn gửi tin nhắn đó.
- Hệ thống SMS của mạng điện thoại sẽ lưu trữ tin nhắn đó rồi sau đó gửi nó tới người bạn đó khi điện thoại của người bạn này mở nguồn.
- Các tin nhắn SMS ít gây phiền phức trong khi bạn vẫn có thể giữ liên lạc với người khác Việc đọc và viết các tin nhắn SMS không gây ra ồn ào. Trong khi đó, bạn phải chạy ra ngoài khỏi rạp hát, thư viện hay một nơi nào đó để thực hiện một cuộc điện thoại hay trả lời một cuộc gọi. Bạn không cần phải làm như vậy nếu như tin nhắn SMS được sử dụng.
- Các điện thoại di động và chúng có thể được thay đổi giữa các sóng mang Wireless khác nhau. Tin nhắn SMS là một công nghệ rất thành công và trưởng thành. Tất cả các điện thoại mobile ngày nay đều có hỗ trợ nó. Bạn không chỉ có thể trao đổi các tin nhắn SMS đối với người sử dụng mobile ở cùng một nhà cung cấp dịch vụ mạng sóng mang wireless, mà đồng thời bạn cũng có thể trao đổi nó với người sử dụng khác ở các nhà cung cấp dịch vụ khác.
- SMS là một công nghệ phù hợp với các ứng dụng Wireless sử dụng cùng với nó.
- Nói như vậy là do:
 - + Thứ nhất, tin nhắn SMS được hỗ trợ 100% bởi các điện thoại có sử dụng công nghệ GSM. Xây dựng các ứng dụng wireless trên nền công nghệ SMS có thể phát huy tối đa những ứng dụng có thể dành cho người sử dụng.
 - + Thứ hai, các tin nhắn SMS còn tương thích với việc mang các dữ liệu binary bên cạnh gửi các text. Nó có thể được sử dụng để gửi nhạc chuông, hình ảnh, hoạt họa ...
- Thứ ba, tin nhắn SMS hỗ trợ việc chi trả các dịch vụ trực tuyến.

5.2 MODULE SIM800A

Phiên bản Sim900A xuất hiện từ hơn 10 năm trước và hiện tại hãng Simcom đã ngưng sản xuất. Sim900A hiện tại đã gần như được thay thế hoàn toàn bằng phiên bản nâng cấp Sim800A với thiết kế, cách sử dụng và bộ tập lệnh "y chang", ngoài ra còn có thêm 1 số tính năng mới như nhận dạng được key tone, hoạt động ổn định và bền bỉ hơn rất nhiều so với Sim900.



Hình 5.1 Mạch GSM GPRS Sim800A

Mạch GSM GPRS Sim800A tích hợp nguồn xung và ic đệm được thiết kế cho các ứng dụng cần độ bền và độ ổn định cao.

Mạch GSM GPRS Sim800A (SIM900A update) tích hợp nguồn xung và ic đệm được thiết kế nhỏ gọn nhưng vẫn giữ được các yếu tố cần thiết của thiết kế Module Sim như: Mạch chuyển mức tín hiệu logic sử dụng Mosfet, IC giao tiếp RS232 MAX232, mạch nguồn xung dòng cao, khe sim chuẩn và các đèn led báo hiệu, mạch còn đi kèm với Anten GSM.

Thông số kỹ thuật:

- Sử dụng module GSM GPRS Sim800A.
- Nguồn cấp đầu vào: 5 - 18VDC, lớn hơn 1A.
- Mức tín hiệu giao tiếp: TTL (3.3-5VDC) hoặc RS232.
- Tích hợp IC chuyển mức tín hiệu RS232 MAX232.
- Tích hợp nguồn xung với dòng cao cung cấp cho Sim800A.
- Sử dụng khe Micro Sim.
- Thiết kế mạch nhỏ gọn, bền bỉ, chống nhiễu.

Thứ tự các chân:

Header 1:

- VCC: Nguồn dương từ 5-18VDC, lớn hơn 1A
- GND: Mass, 0VDC.
- EN: Mặc định nối lên cao, chức năng dùng để khởi động (Enable) hoặc dừng hoạt động (Disable) Module Sim800, nếu muốn module Sim800 dừng hoạt động bạn có thể nối chân này xuống âm GND (0VDC).
- 232R: Chân nhận tín hiệu RS232.
- 232T: Chân truyền tín hiệu RS232
- GND: Mass, 0VDC.
- RXD: Kết nối với RX của MCU.
- TXD: Kết nối với TX của MCU (Chân nhận tín hiệu TTL 3.3V).

Header 2:

- BRXD: Thường không sử dụng, chân nhận tín hiệu, dùng để giao tiếp nạp Firmware cho Sim800, mức tín hiệu 3.3VDC.

- BTXD: Thường không sử dụng, chân truyền tín hiệu, dùng để giao tiếp nạp Firmware cho Sim800, mức tín hiệu 3.3VDC.
- GND: Mass, 0VDC.
- EPN: Ngõ ra loa Speaker âm
- EPP: Ngõ ra loa Speaker dương.
- MICP: Ngõ vào Micro dương.
- MICN: Ngõ vào Micro âm.

Để test hoạt động của module, bạn có thể dùng Vi điều khiển với giao tiếp UART hoặc tốt nhất và nhanh nhất là sử dụng mạch chuyển từ USB sang UART và truyền lệnh AT trực tiếp từ máy tính.

5.3 KHẢO SÁT TẬP LỆNH AT CỦA MODULE SIM800A

- Các lệnh AT viết tắt là Attention các hướng dẫn được sử dụng để điều khiển một modem. Bắt đầu mỗi dòng lệnh là “AT” hay “at”. Vì vậy mà các lệnh trong modem được gọi là lệnh AT.
- Từ các lệnh “AT” này, người lập trình có thể làm một số bước sau:
 - + Ban đầu đọc tin nhắn, viết tin nhắn và xóa tin nhắn.
 - + Thực hiện gửi tin nhắn SMS.
 - + Kiểm tra toàn bộ chiều dài nội dung tin nhắn
- Số lần gửi tin nhắn có thể thực hiện được trên một phút tương đối thấp, vào khoảng 6-10 tin nhắn trên 1 phút.
- Trong khuôn khổ của đề án này em chỉ tìm hiểu 1 số tập lệnh cơ bản phục vụ cho việc làm đề án của mình. Sau đây em xin giới thiệu 1 số tập lệnh cơ bản để cài đặt dùng cho dịch vụ SMS, bao gồm:
 - + Bước đầu tiên là công việc khởi tạo.
 - + Bước thứ hai là nhận và xử lý tin nhắn.
 - + Bước làm cuối cùng là gửi tin nhắn đi.

5.3.1 Các thuật ngữ

- <CR>: Carriage return (đọc dịch từ mã ASCII là \$0D).
- <LF>: Line Feed (đọc dịch từ mã ASCII là 0x0A)
- MT : Mobile Terminal – Thiết bị đầu cuối (ở đây là Module sim900).
- TE : Terminal Equipment – Thiết bị đầu cuối (ở đây là máy tính giao tiếp được dùng để giao tiếp với Module sim).

5.3.2 Cú pháp lệnh AT

- Lệnh khởi đầu : luôn là “AT” hoặc “at”
- Lệnh kết thúc là : ký tự <CR>(trong đồ án cần chuyển sang mã ASCII là \$0D).
- Thông thường sau mỗi lệnh AT là một đáp ứng, cấu trúc của đáp ứng này là : “<CR><LF><Response>“<CR><LF>”
- Cú pháp chính của lệnh AT có thể được phân chia thành 3 loại : cú pháp có cấu trúc cơ bản, cú pháp có cấu trúc tham số S, cú pháp có cấu trúc mở rộng.
- Với các cú pháp nêu trên thì các lệnh có thể hoạt động ở nhiều chế độ khác nhau. Các chế độ này được thống kê ở bảng bên dưới như sau :
- Một số chế độ của lệnh AT:

Bảng 5.1 Các chế độ lệnh AT

<Lệnh kiểm tra>	AT+<x>=?	Thống kê lại các tham số trong câu lệnh và các giá trị có thể thiết lập cho tham số.
<Lệnh đọc>	AT+<x>=?	Đọc nội dung tin nhắn được gửi đến, kiểm tra giá trị tin nhắn về mặt dữ liệu.
<Lệnh thiết lập>	AT+<x>=<...>	Đọc sử dụng để thiết lập các giá trị cho tham số
<Lệnh thực thi>	AT+<x>	Thực thi nội dung tin nhắn được tiến hành bên trong của Module sim

5.3.3 Các lệnh AT cơ bản

Lệnh ATZ

Lệnh ATZ dùng thiết lập lại (reset) tất cả các tham số hiện tại theo mẫu được người dùng định nghĩa. Lệnh trả về của modem là lệnh OK. Mẫu người dùng định nghĩa trước đó được lưu trên bộ nhớ cố định. Nếu không thiết lập lại được theo mẫu của người dùng định nghĩa thì nó sẽ reset lại theo đúng các tham số mặc định của nhà sản xuất. Bất cứ lệnh AT cộng thêm nào trên cùng một dòng với lệnh ATZ đều không được thực hiện.

Lệnh AT+CMGR : đọc nội dung tin nhắn

Lệnh AT+CMGR được dùng để đọc tin nhắn trên một ngăn nào đó trên sim điện thoại. Cấu trúc lệnh như sau: AT+CMGR=i, với i là ngăn bộ nhớ chứa tin nhắn trong sim. Đáp ứng trở về là lệnh OK nếu ngăn i có chứa tin nhắn. Nếu ngăn i không chứa

tin nhắn thì sẽ xuất hiện thông báo lỗi trả về ERROR. Ví dụ khi gõ lệnh AT+CMGR=1 thì sim900 sẽ đọc tin nhắn tại ngăn số 1 của bộ nhớ sim điện thoại gắn ngoài.

Lệnh AT+CMGS : gửi tin nhắn SMS

Lệnh AT+CMGS dùng để gửi tin nhắn SMS tới một số điện thoại cho trước. Cú pháp gửi tin như sau: 16 - AT+CMGS= “số điện thoại cần gửi” - Nội dung tin nhắn - ESC/Ctrl Z Số điện thoại cần gửi phải được đặt trong dấu ngoặc kép. Sau khi gõ xong số điện thoại thì cần thực hiện lệnh enter để xuống dòng và bắt đầu nội dung tin nhắn. Kết thúc lệnh ày bằng việc thực hiện lệnh Ctrl Z.

AT+CMGD : xóa tin nhắn SMS

Lệnh AT+CMGD dùng để xóa tin nhắn SMS trên sim. Cấu trúc lệnh như sau : AT+CMGD=i Với i là ngăn nhớ chứa tin nhắn cần xóa. Nếu ngăn i chứa tin nhắn thì đáp ứng trở về là OK, nếu việc thực hiện tin nhắn không thực hiện được như ngăn i không có tin nhắn, hoặc kết nối tới sim, lỗi sóng thì trả về sẽ là ERROR. Ví dụ xóa tin nhắn từ ngăn số 1 của sim: AT+CMGD=1.

Lệnh ATE : thiết lập chế độ lệnh phản hồi

- Lệnh này dùng để thiết lập chế độ lệnh phản hồi trở lại. Đáp ứng trả lại OK.
- Lệnh AT có hai tham số hoàn toàn khác nhau :
 - + ATE0 : tắt chế độ phản hồi.
 - + ATE1 : bật chế độ phản hồi.
- Khi giao tiếp Module sim900 với phần mềm putty trên máy tính, nếu ta dùng lệnh ATE0 thì khi gõ lệnh AT khác thì không nhìn thấy lệnh ta gõ mà chỉ nhìn thấy kết quả trả về của sim900. Ngược lại, khi dùng lệnh ATE1 thì sẽ nhìn thấy cả lệnh gõ lên và lệnh sim900 trả về.

Lệnh AT+W : lưu các tham số hiện tại vào mẫu người dùng

Lệnh AT+W được dùng để lưu cấu hình cài đặt được thiết lập bởi các lệnh ATE và AT+CLIP vào bộ nhớ (Lệnh AT+CLIP để cài đặt cuộc gọi). Đáp ứng trả về khi thực hiện lệnh này là OK.

Lệnh AT+CMGF : lựa chọn định dạng tin nhắn SMS

Lệnh AT+CMGF dùng để lựa chọn định dạng tin nhắn SMS, với hai chế độ là text và PDU [14], cụ thể như sau: AT+CMGF=1: lựa chọn sử dụng tin nhắn ở chế độ văn bản AT+CMGF=0: lựa chọn sử dụng tin nhắn ở chế độ PDU Đáp ứng trả về là “OK”

nếu như modem hỗ trợ, ngược lại, nếu modem không hỗ trợ chế độ định dạng tin nhắn là text hoặc PDU thì đáp ứng trả về sẽ là “ERROR”.

Lệnh AT+CNMI : thông báo có tin nhắn mới đến

Lệnh này dùng để thông báo có tin nhắn mới đến. Với các tham số khác nhau thì mỗi khi có tin nhắn, đáp ứng trả về cũng sẽ khác nhau. Ví dụ về các lệnh AT+CNMI khác nhau khi cùng nhận một tin nhắn SMS có nội dung giống nhau: AT+CNMI=1,1,0,0,0 sẽ trả về: +CMTI: "SM",10 AT+CNMI=2,2,0,0,0 sẽ trả về: +CMT: "+841675047778",",", "13/06/29,14:21:42+28" Abcdef. Như vậy, chúng ta thấy rằng, với trường hợp 1 thì nội dung tin nhắn được lưu trực tiếp vào ngăn số 10 của sim. Nội dung của tin nhắn này chỉ được đọc bằng lệnh AT+CMGR=10. Còn trong trường hợp số 2, nội dung tin nhắn được hiển thị ra cùng với thời gian và số điện thoại.

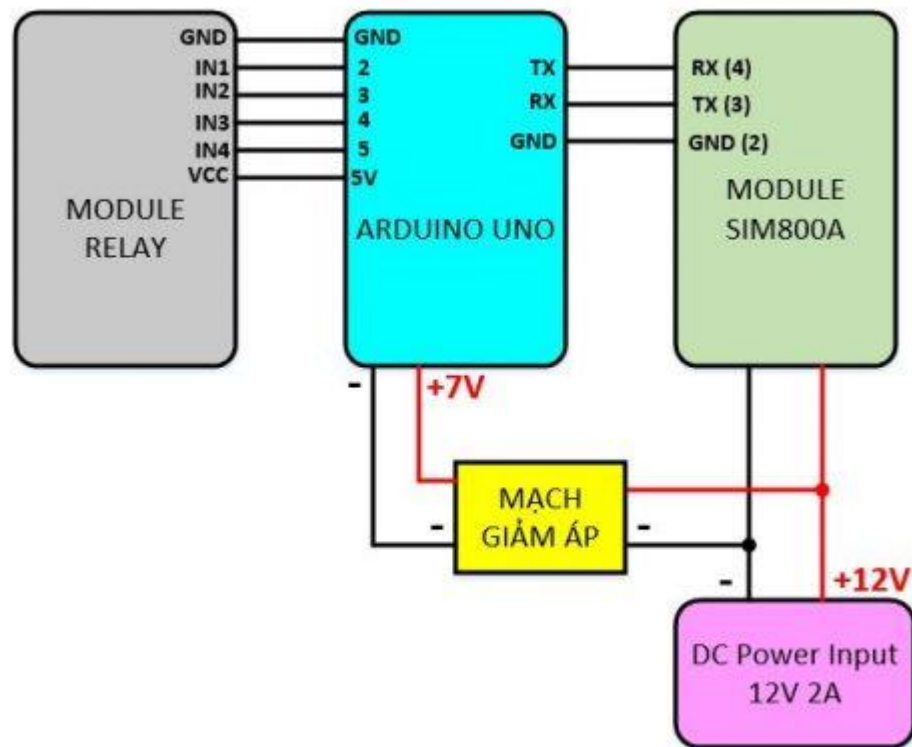
Lệnh AT+CSAS : Lưu các thiết lập tin nhắn SMS

Lệnh AT+CSAS dùng để lưu các thiết lập SMS do người dùng đã cài đặt trước đó. Lệnh này sẽ lưu trực tiếp các thông số đã cài đặt cho tin nhắn SMS như các lệnh AT+CMGF=1; AT+CNMI=2,2,0,0,0...và còn nhiều lệnh khác liên quan tới tin nhắn SMS đều được lưu lại bởi lệnh AT+CSAS này.

5.4 PHẦN CỨNG

1. Module SIM800A
2. Arduino Uno
3. Module Relay 5V 4 kênh
4. Dây kết nối

5.5 SƠ ĐỒ KẾT NỐI



Hình 5.2 Sơ đồ kết nối Arduino – ModuleSIM800A

5.6 CHƯƠNG TRÌNH ARDUINO

Điều khiển 4 relay, Có cuộc gọi từ module sim tới điện thoại báo hệ thống sẵn sàng, Có tính nhắn tin phản hồi, Có chức năng kiểm tra (check) trạng thái thiết bị #A.check*

5.7 CÚ PHÁP NHẮN TIN ĐIỀU KHIỂN

#A.light on*

#A.light off*

#A.tv on*

#A.tv off*

#A.fan on*

#A.fan off*

#A.all on*

#A.all off*

#A.switch on*

#A.switch off*

#A.check*

5.8 HƯỚNG PHÁT TRIỂN

Xử lý số điện thoại nhắn tin đến:

- Chỉ số điện thoại cài đặt mới điều khiển được. Hoặc cú pháp thêm mật khẩu: mật khẩu đúng mới điều khiển.
- Có thể đổi mật khẩu bằng nhắn tin. (Số ĐT admin mới thực hiện được).
- Chương trình trên chỉ nhắn tin phản hồi về 1 số ĐT cài trong code. Hướng phát triển là số ĐT nào điều khiển thì nhắn tin phản hồi về số ĐT đó.

NỘI DUNG KIỂM TRA – ĐÁNH GIÁ

1. Bài kiểm tra số 1:

Hình thức: Tự luận hoặc trắc nghiệm.

Nội dung:

- Trình bày được các phương pháp điều khiển từ xa.
- Trình bày được lịch sử của ngành điều khiển từ xa.
- So sánh giống nhau và khác nhau cơ bản của các phương pháp điều khiển từ xa

2. Bài kiểm tra số 2:

Hình thức: Tự luận hoặc trắc nghiệm.

Nội dung:

- Giải thích được hoạt động của mạch thu và mạch phát hồng ngoại, RF, Bluetooth
- Trình bày được các thông số cơ bản của mắt thu hồng ngoại, module RF, module Bluetooth
- Tính được các thông số cơ bản (dòng, áp, điện trở) trong một số mạch điều khiển từ xa sử dụng tia hồng ngoại, RF, Bluetooth

3. Bài kiểm tra số 3:

Hình thức: Tự luận hoặc trắc nghiệm.

Nội dung:

- Lựa chọn linh kiện thực hiện mạch điều khiển từ xa qua bluetooth.
- Thực hiện lập trình nhận và xử lý gói tin web gửi cho module GPRS.
- Lựa chọn linh kiện thực hiện mạch điều khiển từ xa bằng điện thoại qua SMS.
- Thực hiện lập trình nhận tin nhắn và xử lý tin nhắn

4. Bài thi kết thúc môn học:

Hình thức: Tự luận hoặc trắc nghiệm.

Nội dung:

- Trình bày được các phương pháp điều khiển từ xa.
- Giải thích được hoạt động của mạch thu và mạch phát hồng ngoại, RF, Bluetooth
- Trình bày được các thông số cơ bản của mắt thu hồng ngoại, module RF, module Bluetooth
- Tính được các thông số cơ bản (dòng, áp, điện trở) trong một số mạch điều khiển từ xa sử dụng tia hồng ngoại, RF, Bluetooth.
- Lựa chọn linh kiện thực hiện mạch điều khiển từ xa qua bluetooth.
- Thực hiện lập trình nhận và xử lý gói tin web gửi cho module GPRS.
- Lựa chọn linh kiện thực hiện mạch điều khiển từ xa bằng điện thoại qua SMS.
- Thực hiện lập trình nhận tin nhắn và xử lý tin nhắn

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thúc Hải, *Mạng máy tính và các hệ thống mở*, Nhà xuất bản Giáo Dục, 1999.
- [2]. Nguyễn Văn Thắng, Nguyễn Trọng Đức, *Ứng dụng Internet of things xây dựng ngôi nhà thông minh*, Đề tài nghiên cứu khoa học cấp trường (Trường Đại học Hàng Hải), 2016
- [3]. <http://arduino.vn>.