

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO MÔN
**HỆ THỐNG NHÚNG
VÀ THIẾT KẾ
GIAO TIẾP NHÚNG**

Đề tài:

**THIẾT KẾ VÀ LẬP TRÌNH MODULE GIẢ
ĐIỀU KHIỂN TỪ XA ĐIỀU HÒA NHIỆT ĐỘ**

Nhóm thực hiện :	ĐOÀN QUANG LUU	20203884
	LÊ QUANG HUY	20200273
	ĐẶNG TRƯỜNG GIANG	20200183
	NGUYỄN BÁ TRUNG HIẾU	20200224

Lớp: Hệ thống Nhúng thông minh và IoT - K65
Giảng viên hướng dẫn: TS ĐÀO VIỆT HÙNG

Hà Nội, August 13, 2023

MỤC LỤC

DANH MỤC HÌNH VẼ	ii
DANH MỤC BẢNG BIỂU	iii
TÓM TẮT BÁO CÁO	iv
CHƯƠNG 1. CHƯƠNG MỞ ĐẦU	1
CHƯƠNG 2. PHÂN TÍCH ĐỀ TÀI	2
2.1 Cơ sở lý thuyết	2
2.1.1 Ánh sáng hồng ngoại (Infrared)	2
2.1.2 Truyền dữ liệu qua ánh sáng hồng ngoại	2
2.2 Linh kiện sử dụng	3
2.2.1 ESP8266	3
2.2.2 Kit Arduino Nano v3	5
2.2.3 LCD NOKIA5110	7
2.3 Chỉ tiêu sản phẩm	9
2.3.1 Chỉ tiêu chức năng	9
2.3.2 Chỉ tiêu phi chức năng	9
CHƯƠNG 3. THIẾT KẾ	10
3.1 Thu tín hiệu	10
3.1.1 Block Diagram	10
3.1.2 Sơ đồ mạch thu tín hiệu	10
3.1.3 Lập trình với Arduino IDE	11
3.1.4 Kết quả thu được	12
3.2 Giao thức LG	13
3.3 RawData	14
3.4 Phát tín hiệu	14

3.4.1	Block Diagram	14
3.4.2	Sơ đồ mạch phát tín hiệu	15
3.4.3	Thiết kế mạch PCB với Altium	16
3.4.4	Mạch sau khi hàn	17
3.4.5	Lập trình nút ON/OFF với Arduino IDE	17
3.4.6	Lập trình TEMP-, TEMP- với Arduino IDE	20
3.4.7	Lập trình hiển thị cho màn hình	22
3.4.8	Kết quả thu được	22
4	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	24
	TÀI LIỆU THAM KHẢO	25

DANH MỤC HÌNH VẼ

Hình 2.1	Bước sóng của một số loại ánh sáng	2
Hình 2.2	Truyền dữ liệu qua ánh sáng hồng ngoại	2
Hình 2.3	Module ESP8266	4
Hình 2.4	Sơ đồ chân ESP8266	5
Hình 2.5	Kit Arduino Nano v3	6
Hình 2.6	Sơ đồ chân Arduino Nano	7
Hình 2.7	LCD NOKIA5110	7
Hình 2.8	Sơ đồ chân LCD NOKIA5110	8
Hình 3.1	Block Diagram bộ thu tín hiệu IR	10
Hình 3.2	Mạch thu tín hiệu	10
Hình 3.3	Thư viện sử dụng trong mạch thu tín hiệu	11
Hình 3.4	Lập trình trong mạch thu tín hiệu	11
Hình 3.5	Dữ liệu thu được với mạch thu ở 23 độ	12
Hình 3.6	Block Diagram bộ phát tín hiệu IR	15
Hình 3.7	Block Diagram bộ phát tín hiệu IR	16
Hình 3.8	Lưu đồ thuật toán nút ON/OFF	18
Hình 3.9	Lập trình cho nút ON với Arduino IDE	19
Hình 3.10	Lưu đồ thuật toán cơ bản cho nút TEMP-/TEMP-	20
Hình 3.11	Lưu đồ thuật toán bước 2(gửi nhiệt độ)	21
Hình 3.12	Lập trình gửi nhiệt độ với Arduino IDE	22

DANH MỤC BẢNG BIỂU

Bảng 2.1	Linh kiện	3
Bảng 3.1	Mã hexacode thu được của điều khiển điều hòa LG	13
Bảng 3.2	Ý nghĩa của các ký tự hexa trong dữ liệu thu được	14

TÓM TẮT BÁO CÁO

Trong thời đại công nghệ phát triển, các thiết bị với những tính năng tuyệt vời được phát triển như: điều hòa nhiệt độ, tủ lạnh, ... Các thiết bị không chỉ được phát triển với tính năng cơ bản, hiện nay sự tiện lợi cũng được đặt lên song song với các tính năng tuyệt vời ấy. Điều khiển từ xa thiết bị là một ví dụ: mặc dù người dùng có thể điều khiển thiết bị bằng công tắc, tuy nhiên, một cách thuận tiện, người sử dụng có thể chỉ cần tại một vị trí có thể điều khiển được thiết bị mong muốn.

Tuy nhiên, đi cùng sự phát triển là có quá nhiều tính năng, từ đó dẫn tới phức tạp, thậm chí là thừa. Bởi vậy, để phù hợp với những người đơn giản, chỉ muốn thực hiện các chức năng cơ bản của thiết bị, nhóm nghiên cứu và thiết kế điều khiển ít nút bấm hơn.

Mục tiêu của bài tập là thiết kế và lập trình điều khiển từ xa điều hòa nhiệt độ đơn giản hơn bằng cách giới hạn chỉ 4 nút cơ bản bao gồm: Bật, tắt, tăng, giảm nhiệt độ. Từ đó đem lại trải nghiệm điều khiển thuận tiện và dễ dàng cho người sử dụng. Bộ giả lập điều khiển từ xa này sẽ giúp giảm bớt sự phức tạp và đảm bảo tính tương thích với nhu cầu thực tế của người dùng.

Bài báo cáo gồm 3 phần chính: mở bài, thực hiện sản phẩm và kết luận. Với sản phẩm được thiết kế sử dụng vi điều khiển 8 bit ATMEGA328P, phần lập trình được lập trình bằng Arduino IDE, bài tập được thiết kế để điều khiển điều hòa tại giảng đường đại học C9 - với điều hòa LG.

CHƯƠNG 1. CHƯƠNG MỞ ĐẦU

Trong thời đại công nghệ phát triển như hiện nay, sự tiện lợi và linh hoạt của việc điều khiển từ xa các thiết bị điện tử đã trở thành một yêu cầu ngày càng tăng của con người. Đặc biệt, khi mùa hè đến với những ngày oi bức, việc kiểm soát nhiệt độ trong không gian sống trở nên cực kỳ quan trọng. Để đáp ứng nhu cầu này, chúng em đã tiến hành thiết kế và lập trình một bộ giả lập điều khiển từ xa cho thiết bị điều hòa nhiệt độ, nhằm đơn giản hóa quá trình điều khiển và mang lại sự tiện ích cho người sử dụng.

Hiện nay, các thiết bị điều khiển từ xa đã được phát triển để đáp ứng nhu cầu điều khiển từ xa của người dùng. Tuy nhiên, với số lượng nút điều khiển ngày càng tăng, có thể không đến mức phức tạp, tuy nhiên với những người sử dụng đơn giản, việc nhiều nút bấm dẫn đến thừa, thậm chí bất tiện. Người dùng thường phải đổi mặt với việc điều khiển qua nhiều chế độ, những tính năng phức tạp mà họ thực sự không cần thiết. Điều này gây ra sự bất tiện và khó khăn trong việc sử dụng, đồng thời tăng nguy cơ gây nhầm lẫn trong quá trình điều khiển.

Với thực trạng đó, nhóm em đã nghiên cứu và thiết kế điều khiển từ xa điều khiển điều hòa nhiệt độ đơn giản với 4 nút: ON, OFF, TEMP-, TEMP-; điều khiển phù hợp với thực tế đồng thời đáp ứng chỉ tiêu thầy đặt ra:

- Sử dụng vi điều khiển 8 bit
- Lập trình bằng ngôn ngữ C hoặc tương đương, không sử dụng thư viện có sẵn
- Mã nguồn cho phép đổi chân I/O tùy ý bằng cách thay đổi code. Các chân chức năng được định nghĩa tại đầu chương trình bằng lệnh define

Nội dung của báo cáo gồm:

- Phần mở đầu giới thiệu đề tài.
- Cơ sở lý thuyết
- Thực hành và kết quả
- Kết luận.

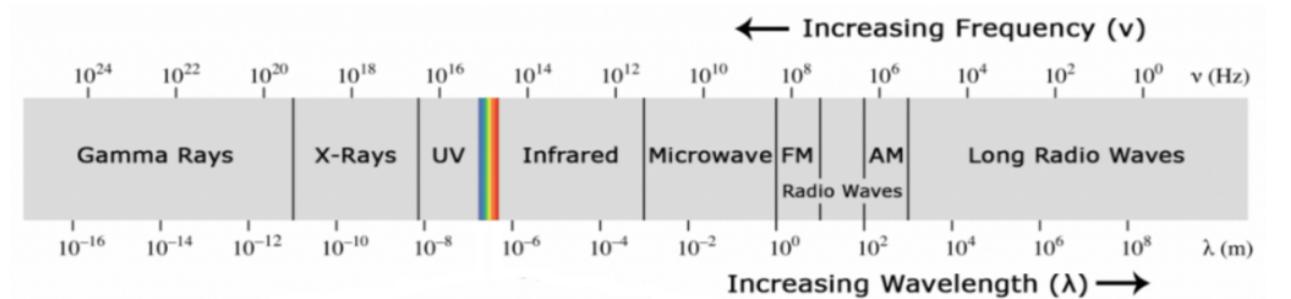
Chúng em xin được gửi lời cảm ơn đến thầy Đào Việt Hùng vì sự tận tình và niềm nhiệt huyết trong việc hướng dẫn bài tập lớn và quá trình dạy học. Sự giúp đỡ của thầy đã giúp chúng em hiểu rõ hơn về kiến thức và có thể áp dụng trong thực tế, đồng thời hoàn thành được một bài tập lớn. Trong quá trình làm việc, nhóm em có thể không tránh khỏi một số lỗi, vì vậy mong thầy có thể góp ý, chỉnh sửa để nhóm em có thể cải tiến và hoàn thiện sản phẩm trong tương lai.

CHƯƠNG 2. PHÂN TÍCH ĐỀ TÀI

2.1 Cơ sở lý thuyết

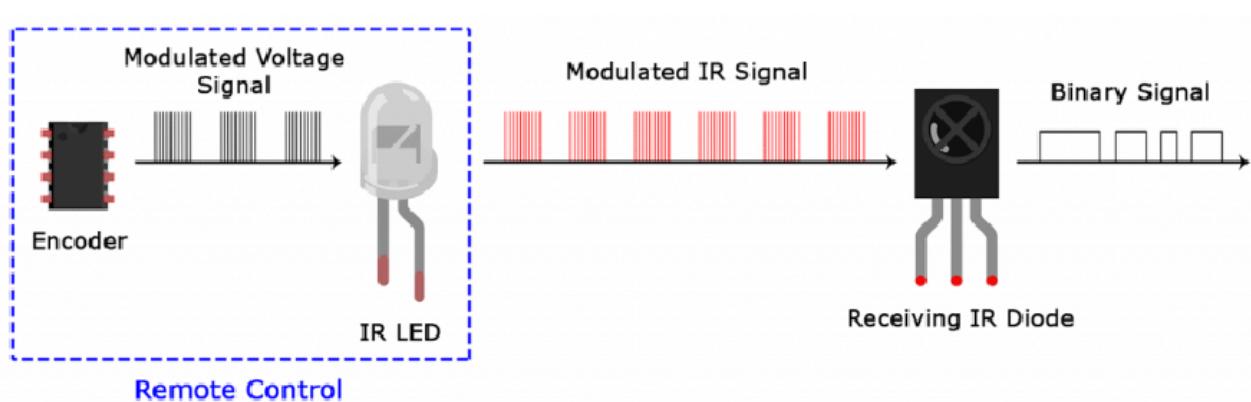
2.1.1 Ánh sáng hồng ngoại (Infrared)

Tia hồng ngoại là bức xạ điện từ có bước sóng dài hơn ánh sáng nhìn thấy nhưng ngắn hơn tia bức xạ vi ba. Vùng ánh sáng mà mắt người thông thường nhìn thấy, còn được gọi là "ánh sáng khả kiến", có bước sóng từ 380 nm đến 760 nm hay tần số 430-790 THz. Bức xạ hồng ngoại được định nghĩa có bước sóng từ 760 nm đến 1 mm. [1]



Hình 2.1 Bước sóng của một số loại ánh sáng

2.1.2 Truyền dữ liệu qua ánh sáng hồng ngoại



Hình 2.2 Truyền dữ liệu qua ánh sáng hồng ngoại

Nguyên lý cơ bản của loại điều khiển từ xa này là sử dụng ánh sáng hồng ngoại của quang phổ điện từ mà mắt thường không thấy được để chuyển tín hiệu đến thiết bị cần điều khiển [2]. Nó đóng vai trò như một bộ phát tín hiệu, sẽ phát ra các xung ánh sáng hồng ngoại mang một mã số nhị phân cụ thể. Khi ta ấn một nút phía bên ngoài thì sẽ vận hành một chuỗi các hoạt động khiến các thiết bị cần điều khiển sẽ thực hiện lệnh của nút bấm đó.

Quy trình này cụ thể như sau: Đầu tiên, khi ta nhấn vào một nút Mạch sẽ gửi tín hiệu đến đèn LED nằm phía trước. Từ đây, đèn LED sẽ phát ra một chuỗi các xung ánh sáng chứa các mã nhị phân (gồm những dãy số 1 và 0) tương ứng với lệnh “tăng âm lượng”. Mã lệnh này gồm nhiều mã con như khởi động, tăng âm lượng, mã địa chỉ thiết bị và ngừng lại khi thả nút ra.

Về phía bộ phận cần điều khiển, nó sẽ gồm một bộ thu ánh sáng hồng ngoại nằm ở mặt trước để có thể dễ dàng nhận được tín hiệu từ điều khiển từ xa. Sau khi đã xác minh mã địa chỉ này xuất phát đúng từ chiếc điều khiển của mình, chúng sẽ giải mã các xung ánh sáng thành các dữ liệu nhị phân để bộ vi xử lý của thiết bị có thể hiểu được và thực hiện các lệnh tương ứng. Điều khiển IR hoạt động dựa trên nguyên lý thu phát hồng ngoại: một bộ phát (transmitter) và một bộ thu (receiver)

2.2 Linh kiện sử dụng

Bảng 2.1 Linh kiện sử dụng trong sản phẩm

Linh kiện	Số lượng
Arduino nano v3	1
ESP8266	1
LCD NOKIA5110	1
Button	4
IR Transmitter	1
IR Receiver	1
Trở $100\Omega/4W$	1
Trở $1k\Omega/4W$	4
ESP8266	1

2.2.1 ESP8266

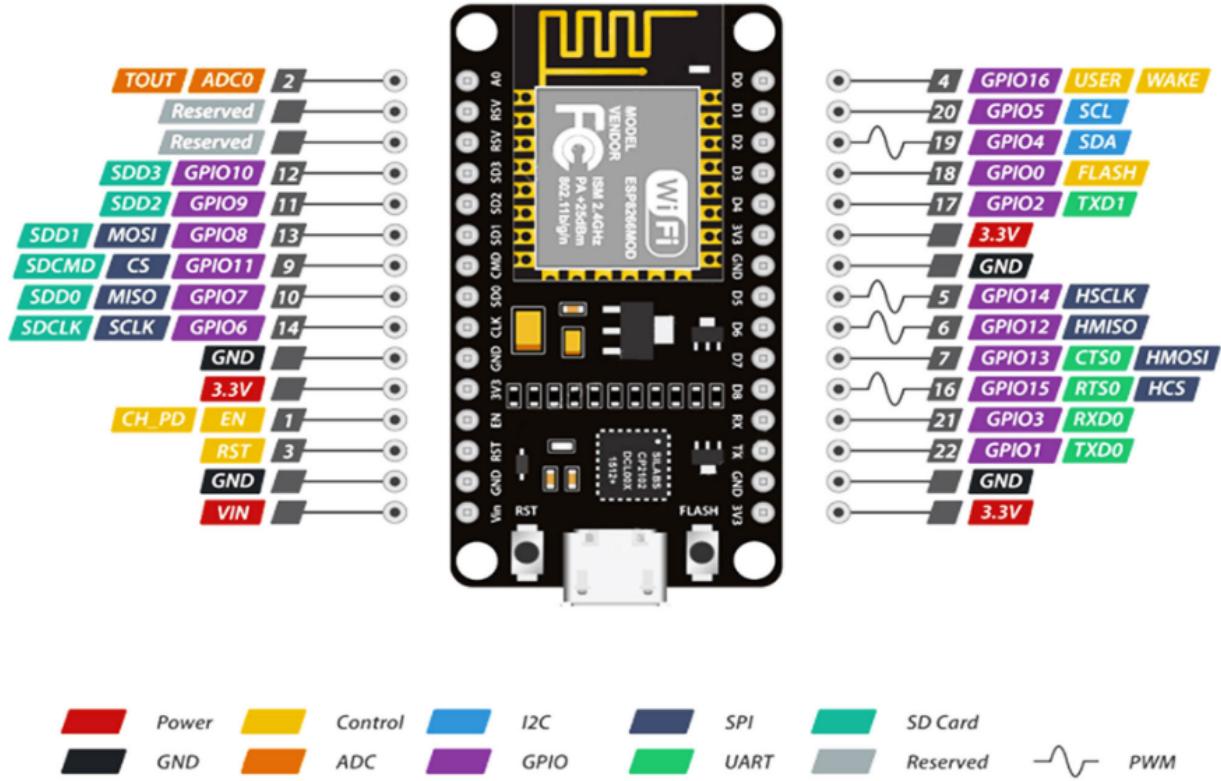
Vì mục tiêu của nhóm là thực hiện bộ phát tín hiệu IR, nhóm quyết định sử dụng ESP8266 để thuận tiện cho bộ thu.



Hình 2.3 Module ESP8266

-*Thông số kỹ thuật của ESP8266:* [3]

- Bộ nhớ: 32 KiB instruction, 80 KiB user data
- CPU: Tensilica Diamond Standard 106Micro (còn gọi là L106) @ 80 MHz (mặc định) hoặc 160 MHz
- Đầu vào: 17 chân GPIO
- Loại: Vi điều khiển 32-bit
- Năng lượng: 3.3 V DC
- Nhà chế tạo: Espressif Systems



Hình 2.4 Sơ đồ chân ESP8266

2.2.2 Kit Arduino Nano v3

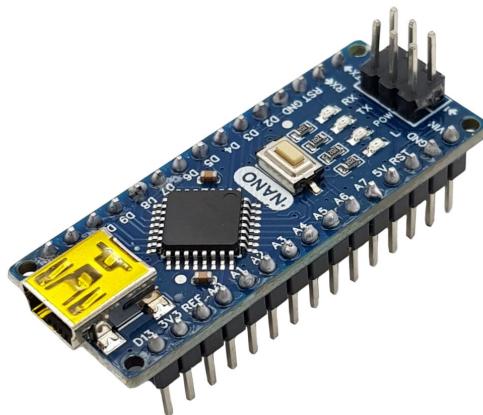
Arduino Nano v3 được sử dụng làm bộ phát.

- Một số thông tin cơ bản: [4]

- Mạch Arduino Nano CH340 có kích thước nhỏ gọn, có thiết kế và chuẩn chân giao tiếp tương đương với Arduino Nano chính hãng, tuy nhiên mạch sử dụng chip nạp chương trình và giao tiếp UART CH340 giá rẻ để tiết kiệm chi phí.
- Arduino Nano là phiên bản nhỏ gọn của Arduino Uno R3 , mọi tính năng hay chương trình chạy trên Arduino Uno đều có thể sử dụng trên Arduino Nano, một ưu điểm của Arduino Nano là Arduino Nano có thêm 2 chân Analog so với Arduino Uno.

- Thông số kỹ thuật của Arduino Nano v3:

- Vi điều khiển: ATmega328P (hộ 8 bit)
- Điện áp hoạt động: 5VDC

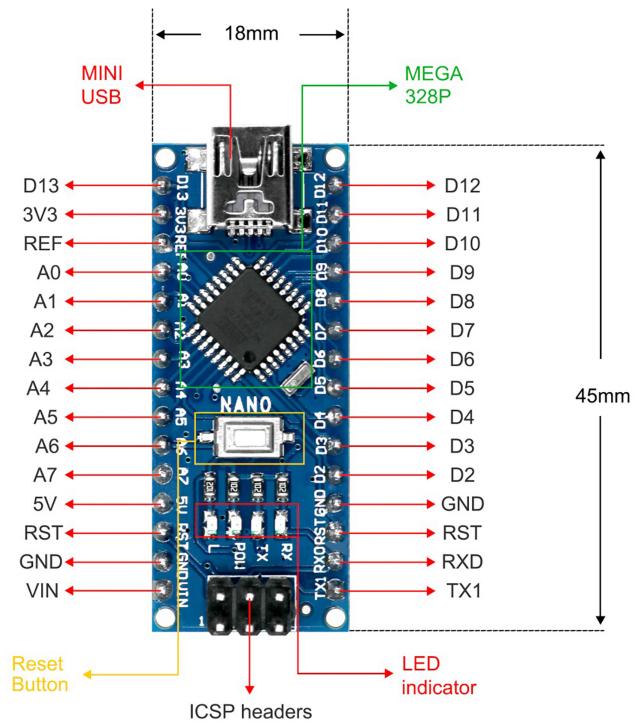


Hình 2.5 Kit Arduino Nano v3

- Điện áp khuyên dùng: 7 - 12 VDC
- Điện áp giới hạn: 6 - 20 VDC
- Tần số hoạt động: 16MHz
- Dòng tiêu thụ: 30mA
- Số chân Digital I/O: 14 chân (6 chân PWM)
- Số chân analog: 8 chân (độ phân giải 10 bit)
- Dòng tối đa trên mỗi chân I/O: 40mA
- Dòng ra tối đa (5V): 500mA
- Dòng ra tối đa (3.3V): 50mA
- Kích thước: 1.85cm x 4.3cm

- Khi kết nối với Arduino Nano:

- Không nối trực tiếp dòng 5V vào GND;
- Không cấp nguồn lớn hơn 5V cho bất cứ chân I/O nào;
- Dòng sử dụng trên pin I/O tối đa 40mA, khuyến cáo sử dụng ở 20mA;
- Tổng cường độ dòng điện cấp cho các I/O pin tối đa là 200mA;



Hình 2.6 Sơ đồ chân Arduino Nano

2.2.3 LCD NOKIA5110

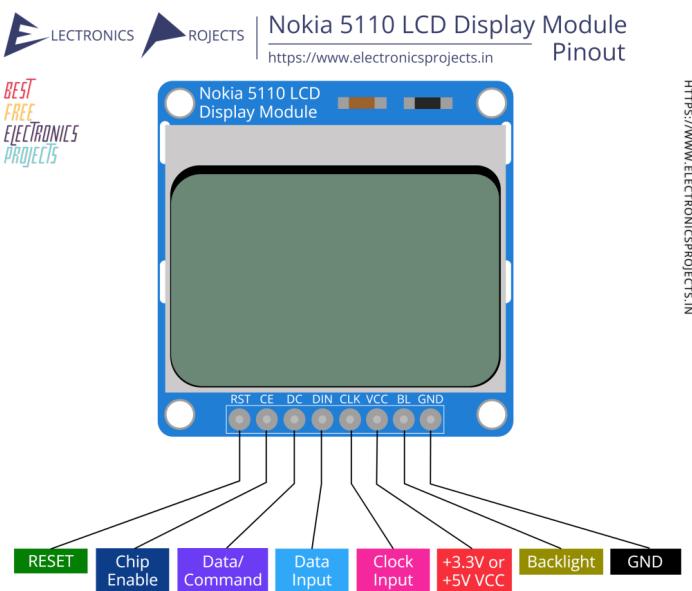
Màn hình Nokia 5110 LCD sử dụng IC điều khiển Philips PCD8544. Màn hình Nokia 5110 LCD rất thích hợp sử dụng cho các dự án Arduino vì kích thước nhỏ, khả năng hiển thị kí tự nhiều.



Hình 2.7 LCD NOKIA5110

Thông số NOKIA 5110: [5]

- Kích thước màn hình 84x48 (pixel)
- Giao thức SPI
- IC điều khiển PCD 8544
- Có card màn hình 504 byte Ram.
- 7pin (2 pin nguồn, 5 pin giao tiếp Spi)
- Điện áp nguồn: 3,5v
- Điện áp mức logic: 0-3,5v



Hình 2.8 Sơ đồ chân LCD NOKIA5110

2.3 Chỉ tiêu sản phẩm

2.3.1 Chỉ tiêu chức năng

- Nguồn: Mạch sử dụng sạc dự phòng làm nguồn cấp cho Arduino nano
Ôn định
- Hiển thị: Nhiệt độ được hiển thị qua LCD Nokia5110
- Chức năng: 4 nút ON, OFF, TEMP-, TEMP+
- Khoảng cách điều khiển: khoảng 1.5m

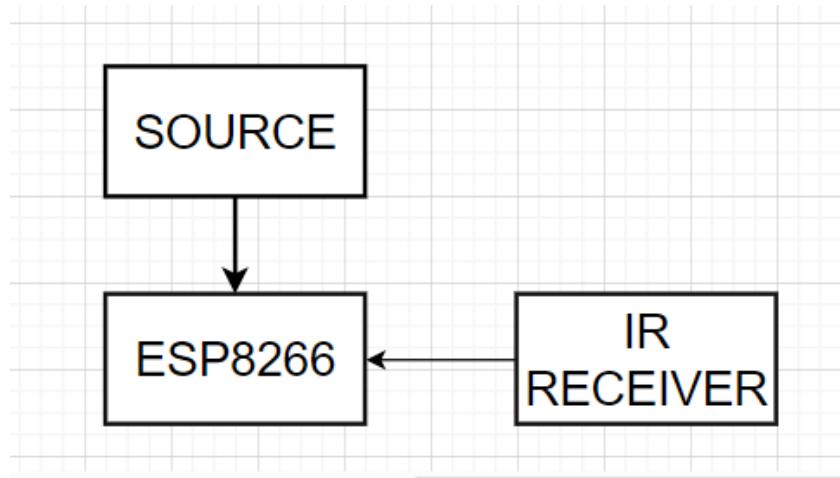
2.3.2 Chỉ tiêu phi chức năng

- Giá thành rẻ, dễ dàng sử dụng
- Kích thước sản phẩm: 7x10 cm
- Thiết kế, hàn mạch PCB

CHƯƠNG 3. THỰC HIỆN PROJECT

3.1 Thu tín hiệu

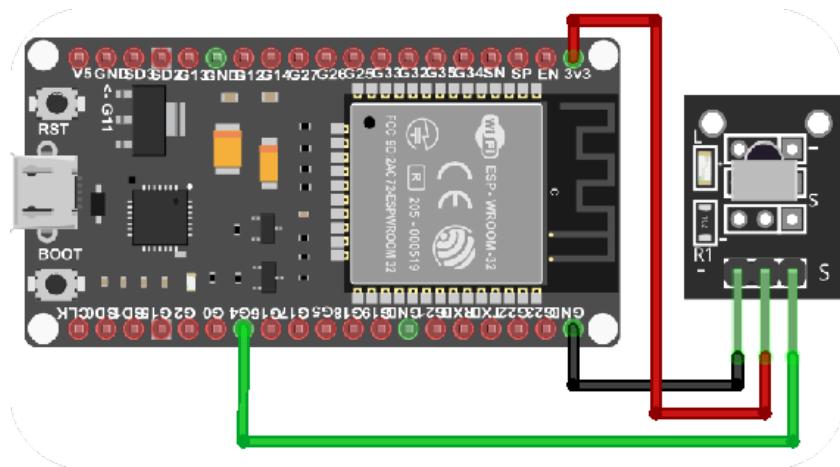
3.1.1 Block Diagram



Hình 3.1 Block Diagram bộ thu tín hiệu IR

ESP8266 được nối với Photodiode có tác dụng thu ánh sáng hồng ngoại. Kết quả được hiện lên Serial Monitor.

3.1.2 Sơ đồ mạch thu tín hiệu



Hình 3.2 Mạch thu tín hiệu

Module thu hồng ngoại có 3 chân: VCC, GND, và chân Signal. Nối 3 chân với ESP8266, ở đây chân Signal nối vào chân 11 của ESP8266.

3.1.3 Lập trình với Arduino IDE

Mục tiêu chính của nhóm trong project là mạch phát tín hiệu, vì vậy phần mạch thu là phần nhóm thực hiện thêm để kiểm tra tín hiệu và thực hiện project tiện lợi hơn. Ở phần code thu tín hiệu, nhóm sử dụng thư viện [IRremoteESP8266.h](#)

Thư viện được sử dụng:

```
#include <Arduino.h>
#include <assert.h>
#include <IRrecv.h>
#include <IRremoteESP8266.h>
#include <IRAc.h>
#include <IRtext.h>
#include <IRutils.h>
```

Hình 3.3 Thư viện sử dụng trong mạch thu tín hiệu

Phần lập trình cho mạch thu được lấy từ Examples của thư viện [IRremoteESP8266.h](#)

```
void loop() {
    // Check if the IR code has been received.
    if (irrecv.decode(&results)) {
        // Display a crude timestamp.
        uint32_t now = millis();
        Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n", now / 1000, now % 1000);
        // Check if we got an IR message that was to big for our capture buffer.
        if (results.overflow)
            Serial.printf(D_WARN_BUFFERFULL "\n", kCaptureBufferSize);
        // Display the library version the message was captured with.
        Serial.println(D_STR_LIBRARY " : v" _IRREMOTEESP8266_VERSION_STR "\n");
        // Display the tolerance percentage if it has been change from the default.
        if (kTolerancePercentage != kTolerance)
            Serial.printf(D_STR_TOLERANCE " : %d%%\n", kTolerancePercentage);
        // Display the basic output of what we found.
        Serial.print(resultToHumanReadableBasic(&results));
        // Display any extra A/C info if we have it.
        String description = IRAcUtils::resultAcToString(&results);
        if (description.length()) Serial.println(D_STR_MESGDESC " : " + description);
        yield(); // Feed the WDT as the text output can take a while to print.
#if LEGACY_TIMING_INFO
    // Output legacy RAW timing info of the result.
    Serial.println(resultToTimingInfo(&results));
    yield(); // Feed the WDT (again)
#endif // LEGACY_TIMING_INFO
```

Hình 3.4 Lập trình trong mạch thu tín hiệu

3.1.4 Kết quả thu được

Sau đây là một ví dụ về kết quả thu được, điều khiển sử dụng là điều khiển điều hòa LG, nhiệt độ là 23 độ C:

Timestamp : 002189.288	Thông tin thư viện
Library : v2.8.6	
Protocol : LG2	Giao thức sử dụng
Code : 0x880084C (28 Bits)	Hexa code Data gửi đi
Mesg Desc.: Model: 2 (AKB75215403), Power: On, Mode: 0 (Cool), Temp: 23C, Fan: 4 (Maximum)	
uint16_t rawData[59] = {3104, 9744, 476, 1564, 522, 516, 502, 514, 472, 570, 502, 1556, 504, 514, 502, 514, 484, 534, 472, 570, 502, 512, 504, 514, 502, 514, 502, 514, 546, 476, 522, 514, 472, 570, 502, 1558, 502, 512, 504, 512, 504, 514, 502, 512, 504, 1530, 472, 544, 472, 546, 526, 1532, 478, 1554, 474, 544, 498, 544, 500}; // LG2 880084C	RawData
uint32_t address = 0x88;	
uint32_t command = 0x84;	
uint64_t data = 0x880084C;	

Hình 3.5 Dữ liệu thu được với mạch thu ở 23 độ

Phần kết quả gồm thông tin thư viện, giao thức sử dụng, mã hexa tín hiệu, thông tin data gửi đi, và rawData (rawData là 1 mảng thời gian các xung được phát đi) - nhóm sẽ phân tích rawData để lấy ra mã hóa các lệnh trong điều khiển.

Tương tự, nhóm tiếp tục thu tín hiệu với các nhiệt độ khác nhau, có bảng sau:

Bảng 3.1 Mã hexacode thu được của điều khiển điều hòa LG

Chức năng	Hexacode
ON tại 23 độ	0x8808844 (28 Bits)
18 độ	0x880834F (28 Bits)
19 độ	0x8808440 (28 Bits)
20 độ	0x8808541 (28 Bits)
21 độ	0x8808642 (28 Bits)
22 độ	0x8808743 (28 Bits)
23 độ	0x8808844 (28 Bits)
24 độ	0x8808945 (28 Bits)
25 độ	0x8808A46 (28 Bits)
26 độ	0x8808B47 (28 Bits)
27 độ	0x8808C48 (28 Bits)
28 độ	0x8808D49 (28 Bits)
29 độ	0x8808E4A (28 Bits)
30 độ	0x8808F4B (28 Bits)
OFF	0x880084C (28 Bits)

3.2 Giao thức LG

Sau khi thu các tín hiệu cần thiết, nhóm phân tích và tìm hiểu về giao thức LG để làm dữ liệu thực hiện mạch phát.

Dựa vào kết quả tín hiệu thu được ở phần 3.1.4, nhận thấy dữ liệu thu được của giao thức LG có 28 bit tương ứng là 7 ký tự hexa. Mỗi tin nhắn được mã hóa theo cách sau [6] :

- Header bao gồm 1 xung cao dài $3300 \mu s$, và nghỉ $9900 \mu s$
- Đổi với bit 0, xung cao $480 \mu s$, nghỉ $1480 \mu s$
- Đổi với bit 1, xung cao $480 \mu s$, nghỉ $480 \mu s$
- Và phần xung kết thúc có độ dài $480 \mu s$

Mỗi một tin nhắn sẽ bao gồm 7 ký tự Hexa và lần lượt các ký tự từ trái qua phải đại diện cho một chức năng:

Bảng 3.2 Ý nghĩa của các ký tự hexa trong dữ liệu thu được

Ký tự số	Ý nghĩa
1	cố định
2	cố định
3	Chức năng đặc biệt (power, swing, air clean)
4	Mode
5	Nhiệt độ
6	Tốc độ gió
7	Checksum

Vì mục tiêu của nhóm là điều khiển nhiệt độ, nên thay vì tập trung vào các chức năng khác như tốc độ gió, mode hoặc các chức năng đặc biệt, nhóm sẽ tập trung vào phần liên quan tới nhiệt độ. Như vậy, dựa vào bảng 3.2, ta thấy ký tự hexa thứ 5 là phần quy định nhiệt độ, và ký tự thứ 7 là phần checksum.

3.3 RawData

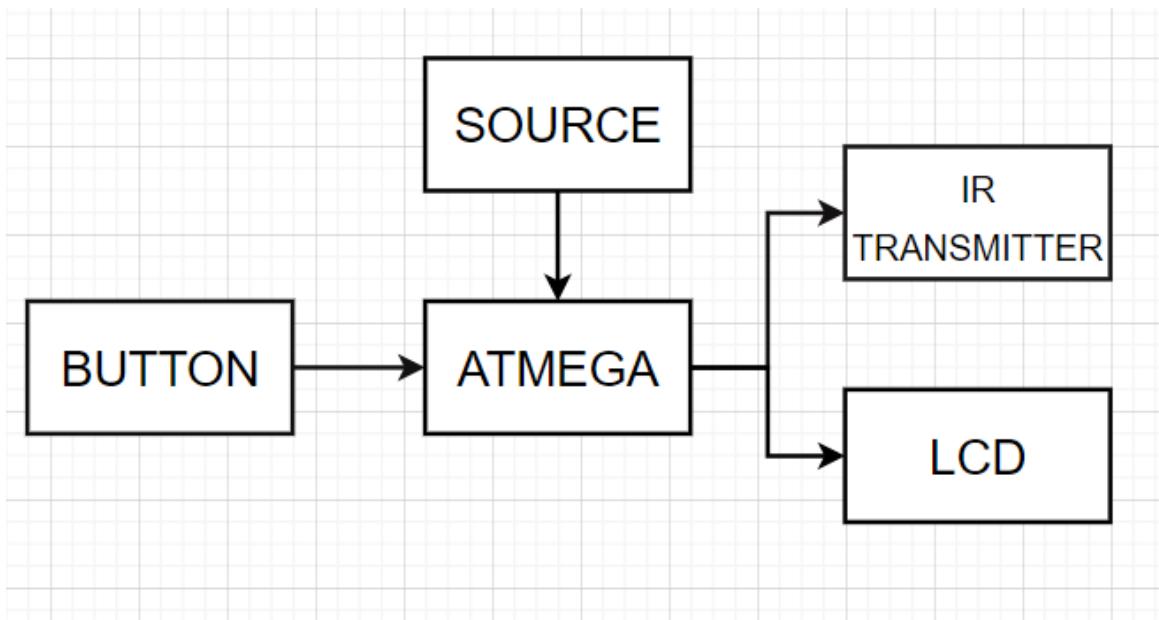
Sau khi phân tích dữ liệu LG, nhóm thực hiện phân tích rawData, đây là phần quy định thời gian ánh sáng hồng ngoại được phát.

Dựa vào hình 3.5, nhận thấy mảng có 59 số, tương ứng với 28 bit và 7 ký tự hexa. Với mảng này, các số đứng lẻ (1,3,5,7,...) là các xung cao, các số đứng chẵn (2,4,6,8,...) sẽ là xung nghỉ. Trong đó:

- Số thứ 1,2 là header với số thứ 1 là xung cao, số thứ 2 là thời gian nghỉ
- Số thứ 3 đến số thứ 58: lần lượt cứ 2 số tiếp theo đại diện cho 1 bit, với xung cao là số thứ nhất, số thứ hai là thời gian nghỉ
- Số cuối cùng (số thứ 59) là thời gian phát kết thúc

3.4 Phát tín hiệu

3.4.1 Block Diagram



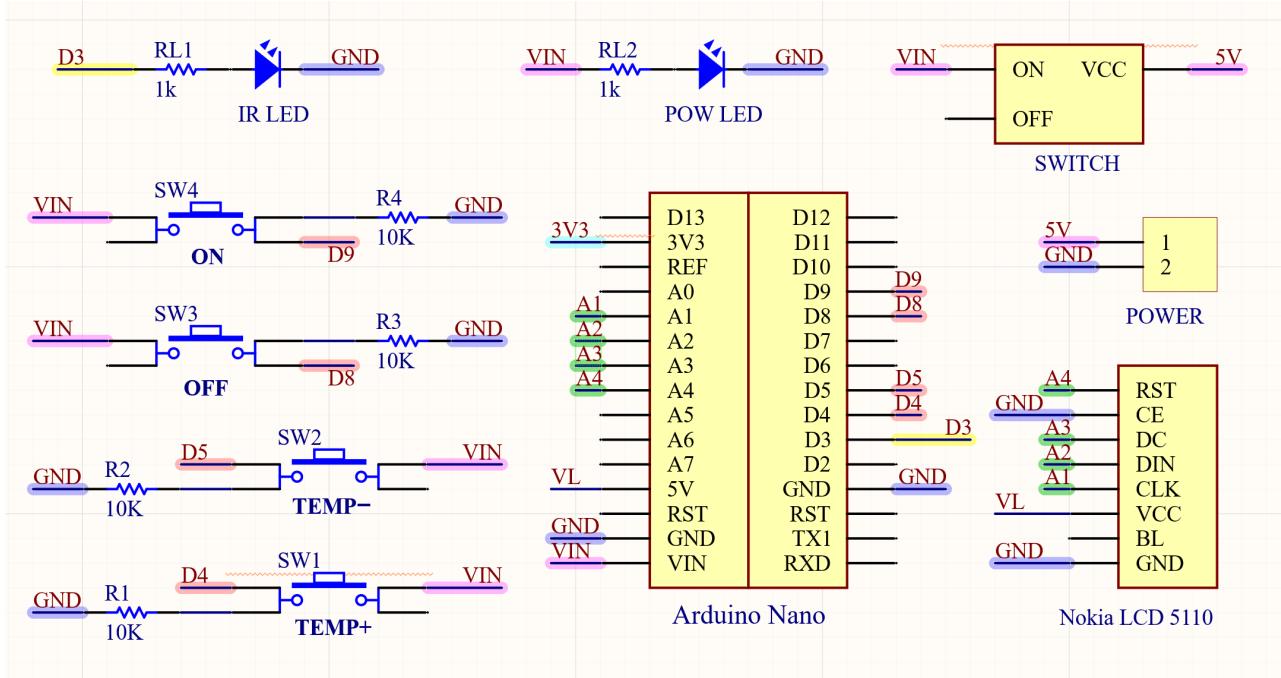
Hình 3.6 Block Diagram bộ phát tín hiệu IR

Mạch gồm các phần:

- BUTTON tương ứng với các nút bấm
- SOURCE: phần nguồn sử dụng sạc dự phòng
- Arduino Nano
- IR transmitter phát ánh sáng hồng ngoại
- Phần hiển thị

Về cơ bản thì, sau khi cấp nguồn cho Arduino nano, khi các nút bấm (button) được nhấn, IR transmitter sẽ được điều khiển phát ánh sáng hồng ngoại, đồng thời nhiệt độ tương ứng được hiển thị qua LCD.

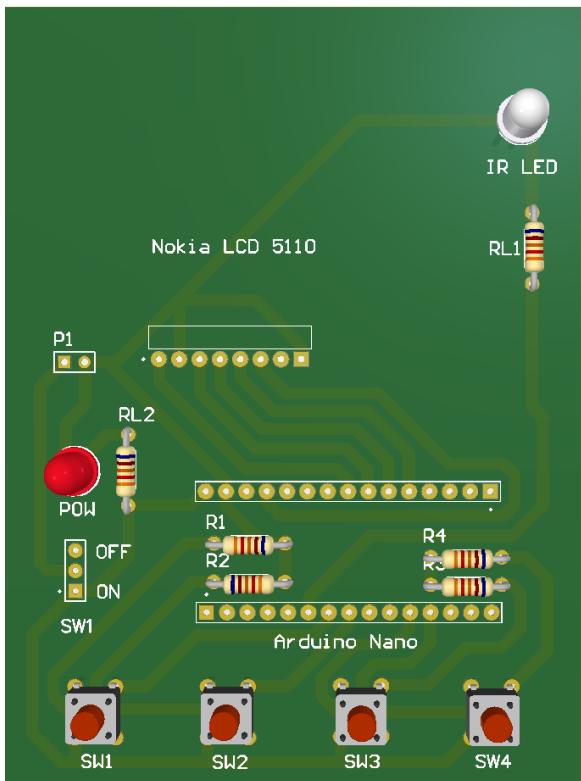
3.4.2 Sơ đồ mạch phát tín hiệu



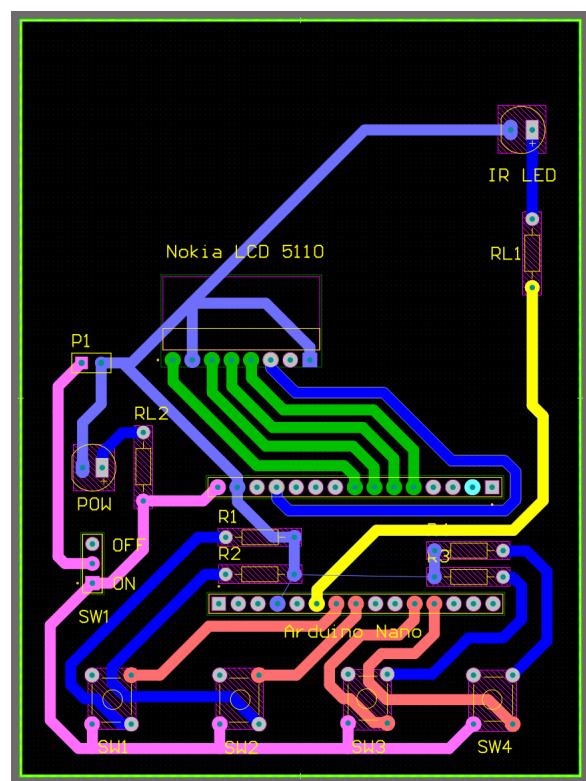
Hình 3.7 Block Diagram bộ phát tín hiệu IR

Các nút bấm được gắn vào trổ, và nối với chân D4, D5, D8, D9 tương ứng lần lượt là các chức năng TEMP-, TEMP-, OFF, ON. LCD SPI nối với chân A1, A2, A3, A4. IR transmitter được nối với chân D3. Ngoài ra, nhóm thiết kế công tắc khi sử dụng với pin, và đèn báo hoạt động.

3.4.3 Thiết kế mạch PCB với Altium



PCB mạch mặt trước

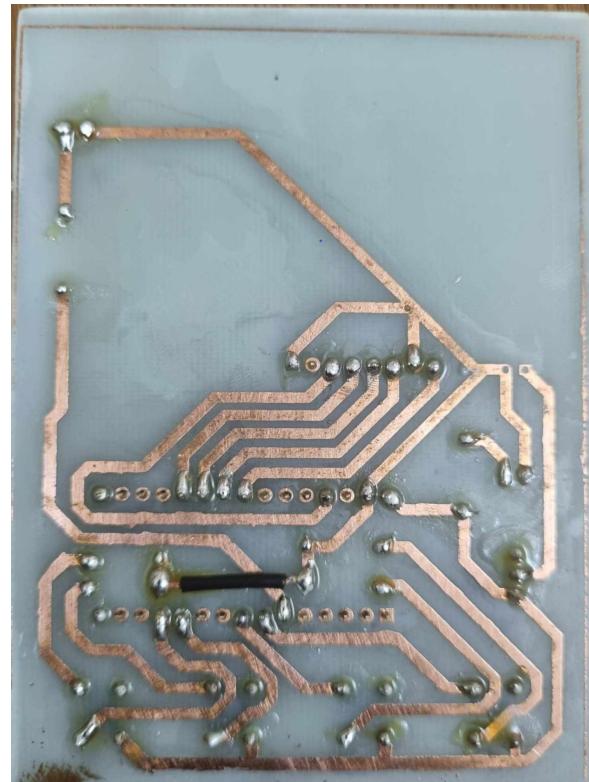


PCB mạch mặt sau

3.4.4 Mạch sau khi hàn

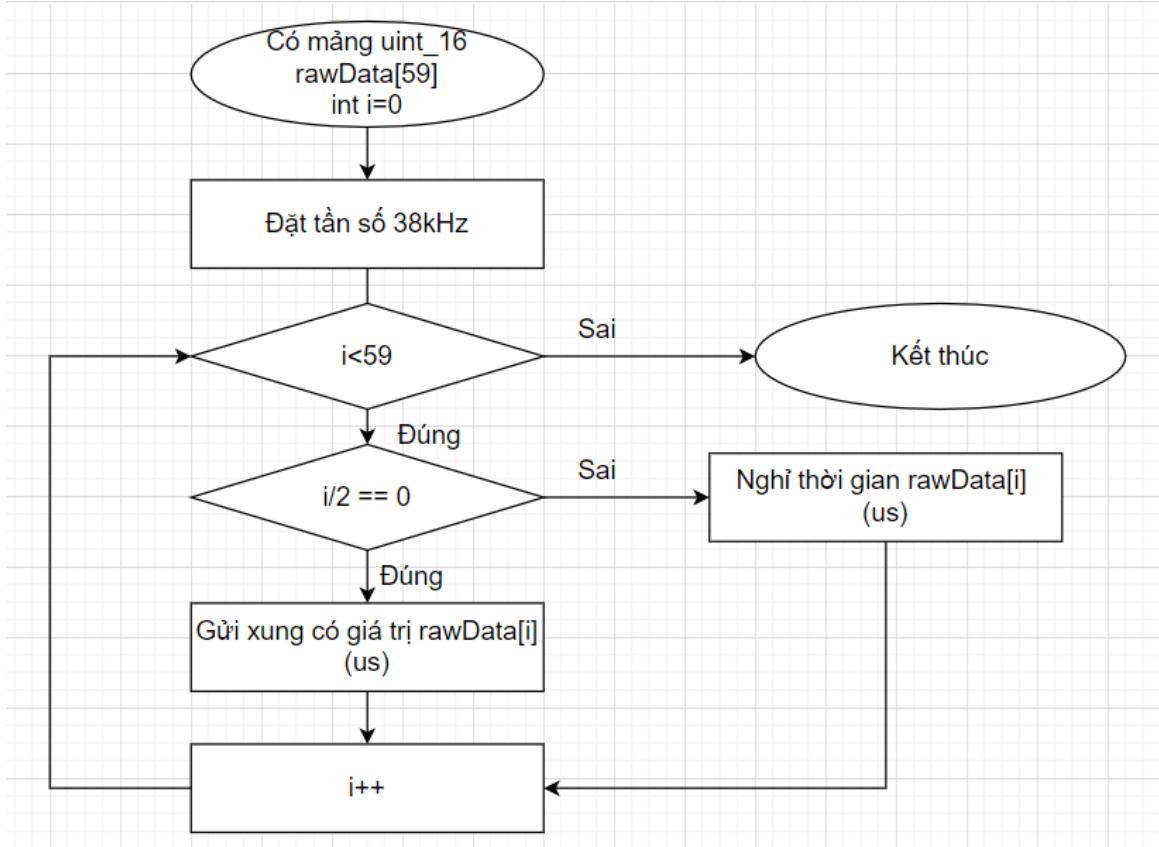


Mặt trước mạch



Mặt sau mạch

3.4.5 Lập trình nút ON/OFF với Arduino IDE



Hình 3.8 Lưu đồ thuật toán nút ON/OFF

Như đã nói ở phần 3.3 và phần 3.2, mảng rawData quy định thời gian gửi xung và nghỉ của IRtransmitter. Trong phần lập trình, có 2 mảng rawDataON[59], và rawDataOFF[59] chứa thông tin về rawData của ON và OFF điều khiển điều hòa LG. Với mỗi mảng, gửi lần lượt các giá trị tương ứng số đứng thứ 0,2,4,...,58 là xung cao và số đứng thứ 1,3,5,...,57 là thời gian nghỉ. Sau khi gửi hết 59 số trong mảng, ta đã hoàn thành phần gửi tín hiệu ON/OFF.

Phần lập trình nút ON cụ thể như sau, nút OFF có phần lập trình tương tự với mảng raw-DataOFF[59]:

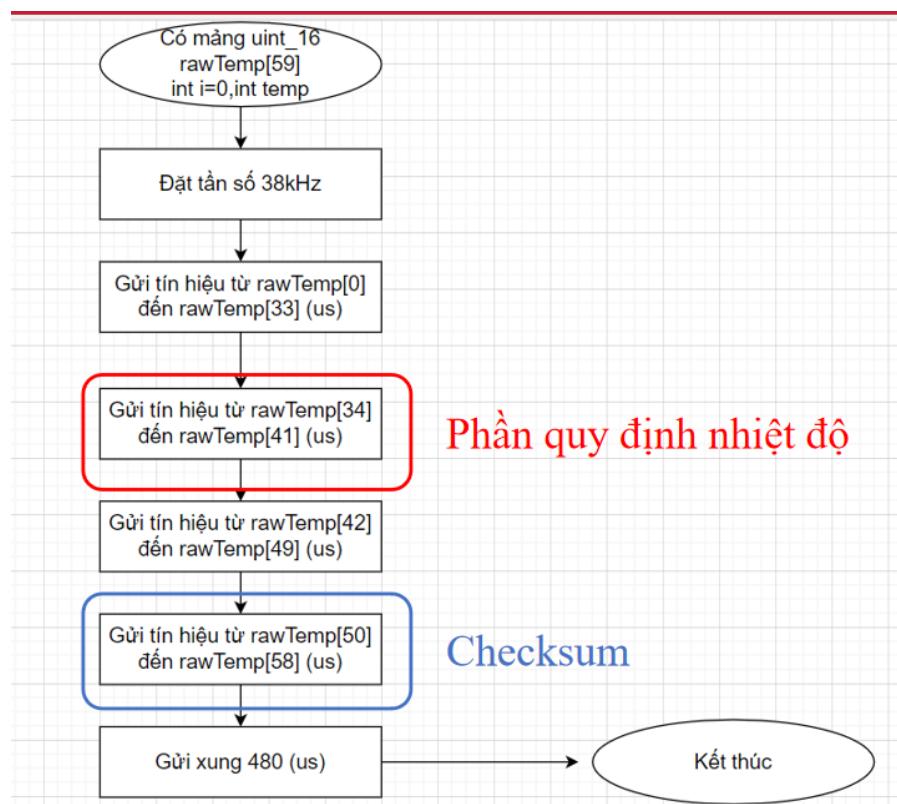
```
// Button On
if (digitalRead(bon) == HIGH) {
    temp=23;
    lcd.Asc_String(10,10,Asc("TEMPERATURE"), BLACK);
    lcd.Number_Long(20, 20,temp,1,BLACK);
    lcd.Asc_Char(50,20,'*O', BLACK);
    lcd.Asc_Char(60,20,67,2,BLACK);
    lcd.Display();
    timer = 0;
    delay(500);
    for (uint_fast16_t i = 0; i < 59; i++) {
        aMicroseconds = pgm_read_word(rawData + i);
        if(aMicroseconds <1000){
            aMicroseconds = 480;
        }
        if(aMicroseconds >1000 && aMicroseconds < 3000 ){
            aMicroseconds = 1480;
        }
        if (i & 1) {
            unsigned long start = micros() - (64 / clockCyclesPerMicrosecond());
            while (micros() - start < aMicroseconds) {}
        } else {
            analogWrite(3, ((256L * 100) / 30));
            unsigned long start = micros() - (64 / clockCyclesPerMicrosecond());
            while (micros() - start < aMicroseconds) {}
            analogWrite(3, 0);
        }
    }
}
```

Hình 3.9 Lập trình cho nút ON với Arduino IDE

3.4.6 Lập trình TEMP-, TEMP- với Arduino IDE

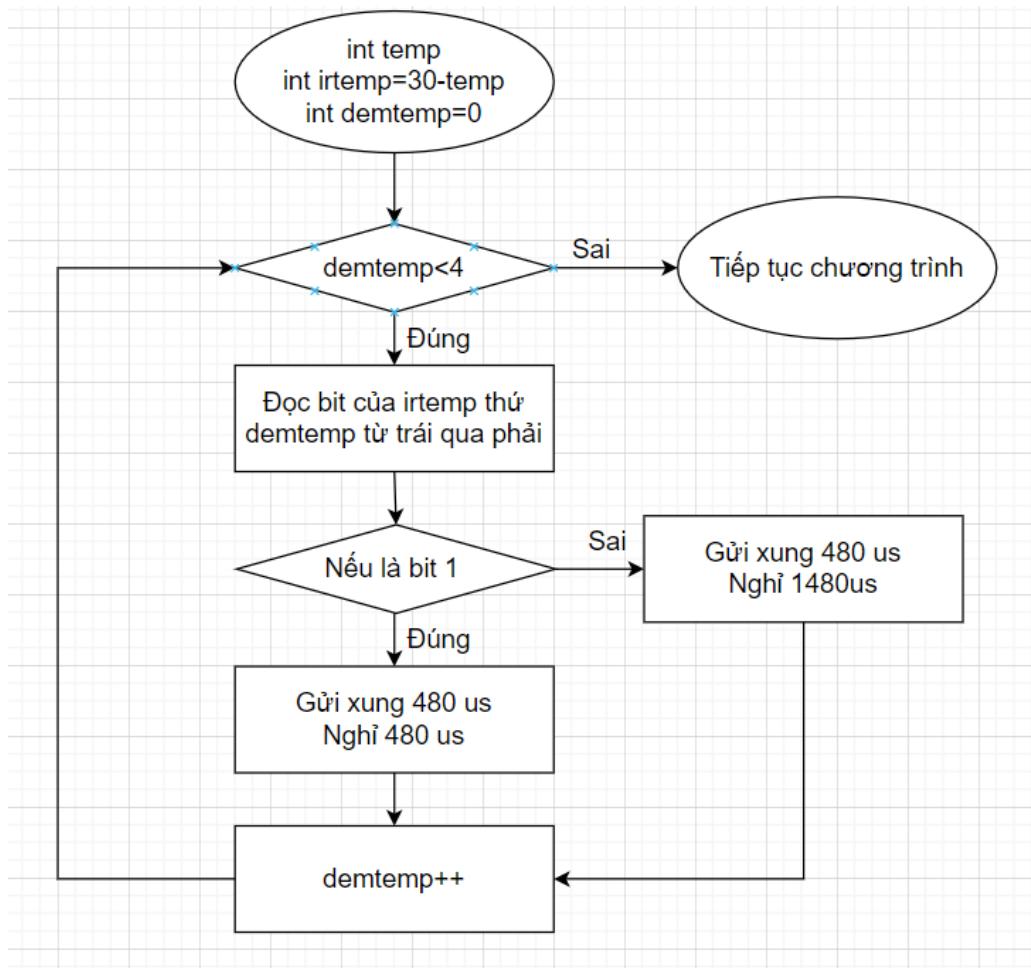
Nút TEMP-, và TEMP- thì có một chút thay đổi so với nút ON/OFF vì nút ON/OFF chỉ cần gửi giá trị tương ứng theo các mảng, còn nút TEMP-, TEMP- cần phải thay đổi theo từng mức nhiệt độ khác nhau, và các giá trị checksum khác nhau. Sau đây là thuật toán cơ bản cho nút TEMP-, và TEMP-. Thuật toán gồm 5 bước gửi tín hiệu (hình 3.11):

- Bước 1: Gửi tín hiệu từ rawTemp[0] đến rawTemp[33] (μs): bước này chỉ là gửi tín hiệu với thời gian tương ứng 34 số đầu trong mảng
- Bước 2: Gửi tín hiệu từ rawTemp[34] đến rawTemp[41] (μs): bước gửi nhiệt độ, phần lập trình nút TEMP-, TEMP- sẽ không gửi tín hiệu tương ứng với các giá trị rawTemp[34] đến rawTemp[41] của mảng, mà sẽ thay đổi tùy theo nhiệt độ được gửi
- Bước 3: Gửi tín hiệu từ rawTemp[42] đến rawTemp[49] (μs): tương tự bước 1
- Bước 4: Gửi tín hiệu từ rawTemp[50] đến rawTemp[57] (μs): bước gửi checksum, cũng tương tự như gửi nhiệt độ, thời gian gửi sẽ thay đổi tùy theo nhiệt độ thay đổi
- Bước 5: Gửi xung kết thúc 480 μs



Hình 3.10 Lưu đồ thuật toán cơ bản cho nút TEMP-/TEMP-

Phần lưu đồ cho bước 2 được làm rõ ở lưu đồ sau, bước 4 tương tự bước 2 với giá trị irchecksum=34-checksum :



Hình 3.11 Lưu đồ thuật toán bước 2(gửi nhiệt độ)

Lập trình gửi nhiệt độ như sau:

```
-----  
int dem;  
for (uint_fast16_t i = 0; i < 34; i++) {  
    aMicroseconds = pgm_read_word(rawDataTemp + i);  
    if(aMicroseconds <1000){aMicroseconds = 480;}  
    if(aMicroseconds >1000 && aMicroseconds < 3000 ){ aMicroseconds = 1480;}  
    sendDataTest(i,aMicroseconds);}  
for (uint_fast16_t i = 34; i < 42; i++) {  
    if(i%2==0){  
        dem = (40-i)/2;  
        if(bitRead(test,dem)!=0){sendDataTest(i,480); sendDataTest(i+1,480);}  
        else{sendDataTest(i,480); sendDataTest(i+1,1480);}}}  
for (uint_fast16_t i = 42; i < 50; i++) {aMicroseconds = pgm_read_word(rawDataTemp + i);  
    if(aMicroseconds <1000){aMicroseconds = 480;}  
    if(aMicroseconds >1000 && aMicroseconds < 3000 ){aMicroseconds = 1480;}  
    sendDataTest(i,aMicroseconds);}  
for (uint_fast16_t i = 50; i <58 ; i++) {  
    if(i%2==0){  
        dem = (56-i)/2;  
        if(bitRead(cheks,dem)!=0){sendDataTest(i,480); sendDataTest(i+1,480);}  
        else{ sendDataTest(i,480); sendDataTest(i+1,1480);}}}  
sendDataTest(58,480);  
}
```

Hình 3.12 Lập trình gửi nhiệt độ với Arduino IDE

3.4.7 Lập trình hiển thị cho màn hình

Cũng như phần thu, phần hiển thị không phải mục đích nhóm hướng tới, vì vậy nhóm quyết định sử dụng thư viện cho màn hình để thuận tiện.

Thư viện sử dụng: [include "PCD8544HOANGSA.h"](#)

Phần lập trình với màn hình (với temp là biến nhiệt độ):

```
lcd.AscString(10,10,Asc("TEMPERATURE"), BLACK);  
lcd.NumberLong(20, 20,temp,1,BLACK);  
lcd.AscChar(50,20,'*O', BLACK);  
lcd.AscChar(60,20,67,2,BLACK);  
lcd.Display();
```

3.4.8 Kết quả thu được

Sau khi hoàn thiện code, nhóm thực hiện thu kết quả với bộ thu và kiểm tra kết quả đạt được: kết quả tín hiệu phát ra từ mạch đúng với dữ liệu đã thu được từ điều khiển LG, khoảng cách điều khiển là khoảng 1,5m. Có thể tạm thời kết luận mạch đã đạt chức năng cơ bản. Để chắc chắn, nhóm sẽ thực hiện trực tiếp với điều hòa LG tại phòng C9-401 của trường.

Sau đây là video demo mạch phát và thu bằng ESP8266 của nhóm:

Video demo phát LG2

Nhóm đã thực hiện với bộ nguồn là pin Litium 18650 kết hợp với bộ ổn áp, tuy nhiên có 1 số trục trặc nên hiện tại sạc dự phòng vẫn được sử dụng, tương lai sẽ cố gắng phát triển mạch bằng việc sử dụng pin.

Video demo phát DAIKIN với nguồn là pin Litium 18650

4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Qua bài tập lớn Thiết kế và Lập trình Bộ giả lập điều khiển từ xa điều hòa nhiệt độ, chúng em đã có cơ hội áp dụng và hiểu sâu hơn về kiến thức lập trình và thiết kế giao tiếp nhúng trong môi trường thực tế. Bài tập lớn này đã đóng góp rất nhiều cho sự nắm bắt của chúng em về các khái niệm và kỹ thuật liên quan đến hệ thống nhúng và giao tiếp nhúng.

Chúng em đã học được cách thiết kế và lập trình một bộ giả lập điều khiển từ xa, từ việc xử lý tín hiệu giao tiếp, xây dựng giao thức truyền thông, đến tương tác với thiết bị điều khiển từ xa. Qua đó, chúng em đã nắm bắt được quy trình và phương pháp thiết kế hệ thống nhúng có khả năng giao tiếp từ xa.

Bài tập lớn cũng đã giúp chúng em rèn luyện kỹ năng làm việc nhóm, tăng cường khả năng phân tích và giải quyết các vấn đề liên quan đến lập trình, giao tiếp và trình bày kết quả một cách rõ ràng và logic.

Tuy chưa kịp hoàn thành nguồn với pin, và chưa đủ khả năng để sử dụng hoàn toàn chip thay cho Arduino Nano, tuy nhiên với sự cố gắng của cả nhóm. Sản phẩm đã đạt đủ chỉ tiêu ban đầu đặt ra, trong tương lai, nhóm sẽ phát triển thêm để hoàn thiện sản phẩm.

Chúng em muốn gửi lời cảm ơn chân thành đến thầy Đào Việt Hùng, người đã truyền đạt kiến thức một cách nhiệt tình và vui tính.

Sự hỗ trợ và giúp đỡ của thầy đã đóng vai trò quan trọng trong quá trình học tập của chúng em. Thầy đã giảng dạy chúng em một cách chi tiết và cụ thể về các khái niệm, kiến thức và kỹ thuật lập trình và thiết kế giao tiếp nhúng cần thiết để hoàn thành bài tập lớn này. Những kinh nghiệm và kiến thức mà thầy truyền đạt cũng đã giúp chúng em hiểu rõ hơn về quá trình lập trình và cách áp dụng nó vào thực tế.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Đào Việt Hùng và chúc thầy luôn khỏe mạnh và thành công trong công việc. Sự hướng dẫn và động viên từ thầy đã truyền cảm hứng và định hướng cho chúng em tiếp tục nghiên cứu và phát triển trong tương lai. Sản phẩm có thể còn một số điểm chưa hoàn thiện, mong thầy nhận xét và góp ý sửa đổi.

TÀI LIỆU THAM KHẢO

- [1] Wikipedia, “Tia hồng ngoại,” [Online]. Available : https://vi.wikipedia.org/wiki/Tia_hong_ngoai
- [2] DICOM, “Điều khiển từ xa bằng hồng ngoại hoạt động như thế nào?,” [Online]. Available : <https://dicom.vn/tin-tuc-su-kien/tin-tuc/dieu-khien-tu-xa-bang-hong-ngoai-hoat-dong-nhu-the-nao.html>.
- [3] Hshop, “Kit rf thu phát wifi esp8266 nodemcu lua cp2102,” [Online].Available:<https://hshop.vn/products/kit-rf-thu-phat-wifi-esp8266-nodemcu>.
- [4] Nshop, “Arduino nano v3.0 atmega328p,” [Online].Available:<https://nshopvn.com/product/arduino-nano-v3-0-atmega328p-khong-kem-day-cap-usb/>.
- [5] Arduino., “Nokia5110 | hướng dẫn sử dụng và chia sẻ thư viện hoÀng sa,” [Online]. Available : <http://arduino.vn/tutorial/1345-nokia5110-huong-dan-su-dung-va-chia-se-thu-vien-hoang-sa>.
- [6] chaeplin, “Lg ac,” [Online]. Available : <https://gist.github.com/chaeplin/7ad00d6d4d29d164f869?fb537lQ>.