ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN - ĐIỆN TỬ

BÁO CÁO

BÀI TẬP LỚN HỌC PHẦN

HỆ THỐNG NHÚNG

THIẾT KẾ GIAO TIẾP NHÚNG

Thiết kế và lập trình mô đun giải mã tín hiệu điều khiển từ xa của máy chiếu

Sinh viên thực hiện:	**Nguyễn Phương Linh**	**20206203**
	Trần Trung Hiếu	**20200230**
	Nguyễn Ngọc Dương	**20200122**
Người hướng dẫn:	**TS. Đào Việt Hùng**	

Hà Nội, 1-2024

MỤC LỤC

PHÂN CÔNG CÔNG VIỆC

Nhiệm vụ		Thành viên
Tìm hiểu lý thuyết	Tìm hiểu lý thuyết thu phát sóng hồng ngoại	Trần Trung Hiếu
	Lựa chọn linh kiện sử dụng, tìm hiểu về datasheet linh kiện sử dụng	Nguyễn Phương Linh Trần Trung Hiếu Nguyễn Ngọc Dương
	Tìm giao thức, bộ mã của YT-150	Nguyễn Phương Linh
	Phân rã mô-đun	Nguyễn Phương Linh
Lập trình	Lập trình thư viện IRLib	Nguyễn Phương Linh Trần Trung Hiếu
	Lập trình thư viện ReadWriteLib	Nguyễn Phương Linh
	Lập trình hàm main, xử lý lệnh và in ra LCD	Trần Trung Hiếu Nguyễn Ngọc Dương
Viết báo cáo	1. Mở đầu	Nguyễn Ngọc Dương
	2. Truyền thông bằng tín hiệu hồng ngoại	Trần Trung Hiếu
	3. Đặc tả hệ thống, phân tích tín hiệu hồng ngoại và phân rã mô-đun	Nguyễn Phương Linh
	4. Mô-đun giải mã tín hiệu hồng ngoại	Nguyễn Phương Linh
	5. Mô-đun hiển thị lên LCD	Trần Trung Hiếu
	6. Triển khai, kiểm thử, gỡ lỗi và làm sản phẩm	Nguyễn Ngọc Dương

	7. Kết quả và kết luận	Nguyễn Ngọc Dương
Hoàn thiện sản phẩm	Kiểm tra hệ thống với điều khiển mô phỏng, điều khiển thực tế	Trần Trung Hiếu
	Gỡ lỗi phát sinh và kiểm tra lại hệ thống	Nguyễn Phương Linh
	Hàn mạch, làm vỏ hộp cho hệ thống	Nguyễn Ngọc Dương
Chuẩn bị slide	Nội dung	Nguyễn Phương Linh Trần Trung Hiếu Nguyễn Ngọc Dương
	Làm slide	Nguyễn Ngọc Dương

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC TỪ VIẾT TẮT (NẾU CẦN)

Viết tắt	Dạng đầy đủ	Giải nghĩa
PCB	printed circuit board	Mạch in
I/O	Input/Output	Vào/Ra
LCD	Liquid-crystal display	Màn hình tinh thể lỏng
LED	Light-emitting diode	Đi-ốt phát sáng
IR	Infrared Radiation	Hồng ngoại
PDM	Pulse-Duration Modulation	Điều chế thời gian xung
FSM	Finite State Machine	Máy trạng thái hữu hạn
ISR	Interrupt Service Routine	Trình phục vụ ngắt
GPIO	General Purpose Input Output	Cổng vào ra chung

1. MỞ ĐẦU

1.1 Giới thiệu tổng quan

Trong bối cảnh công nghệ ngày càng phát triển, các hệ thống truyền thông không dây đang trở thành một lĩnh vực quan trọng, đặc biệt là khi áp dụng vào việc điều khiển các thiết bị từ xa. Vì vậy, việc tìm hiểu và nghiên cứu thông tin được truyền từ các bộ điều khiển từ xa sẽ giúp chúng em hiểu sâu hơn về lĩnh vực này. Và để phù hợp và có thể thực hiện được thì đề tài của chúng em tập trung vào việc thiết kế và lập trình mô-đun giải mã tín hiệu điều khiển từ xa của máy chiếu. Bên cạnh đó, hệ thống giải mã của nhóm cần đáp ứng được các yêu cầu sau :

1. Giải mã bản tin (ứng với ít nhất 4 lệnh) thu từ photodiode hồng ngoại khi điều khiển bằng bộ điều khiển từ xa của máy chiếu có sẵn trong giảng đường.
2. Sử dụng vi điều khiển 8 bit.
3. Lập trình bằng ngôn ngữ C hoặc tương đương, không sử dụng thư viện có sẵn.
4. Mạch điện hàn trên PCB tự thiết kế hoặc sử dụng kit có bán sẵn, tuyệt đối không sử dụng breadboard và cắm dây.
5. Mã nguồn cho phép đổi chân I/O tùy ý bằng cách thay đổi code. Các chân chức năng được định nghĩa tại đầu mỗi chương trình bằng lệnh #define.

1.2 Các nhiệm vụ cần thực hiện

Từ yêu cầu bài toán được đặt ra, nhóm nhận định được các đầu việc lớn cần thực hiện như sau:

- Nghiên cứu phát thu tín hiệu hồng ngoại
- Lựa chọn linh kiện, ví dụ như mô-đun thu phát hồng ngoại để nhận tín hiệu, vi điều khiển/vi xử lý phù hợp và thỏa mãn yêu cầu (là vi xử lý 8

bit) để giải mã cũng như điều khiển các hoạt động tương ứng của tải, tải để thể hiện, mô phỏng các trạng thái tương ứng của máy chiếu.

- Để giải mã được tín hiệu từ điều khiển máy chiếu ở giảng đường (điều khiển Casio YT-150), ta cần biết ***giao thức truyền thông*** được sử dụng và bảng mã các nút của điều khiển
- Lập trình vi điều khiển ...
- Thử nghiệm gỡ lỗi ...
- Làm mạch, hộp,...

1.3 Mục tiêu đặt ra

Với các nhiệm vụ cần phải thực hiện, thì mục tiêu của nhóm cần phải đạt được như sau:

- Nghiên cứu phát thu tín hiệu hồng ngoại: Hiểu rõ về cơ sở lý thuyết và cách hoạt động của tín hiệu hồng ngoại. Xác định các yếu tố ảnh hưởng đến chất lượng thu và phát tín hiệu.
- Lựa chọn linh kiện: Chọn mô-đun thu phát hồng ngoại phù hợp, có khả năng nhận diện và phát tín hiệu một cách chính xác. Lựa chọn vi điều khiển/vi xử lý 8 bit có hiệu suất cao để giải mã và điều khiển máy chiếu.
- Tìm hiểu giao thức truyền thông và bảng mã nút: Nghiên cứu và xác định giao thức truyền thông được sử dụng trong điều khiển Casio YT-150. Tìm hiểu bảng mã các nút để hiểu cách mã hóa các lệnh điều khiển.
- Lập trình vi điều khiển 8 bit: Phát triển mã nguồn cho vi điều khiển 8 bit sao cho có thể giải mã tín hiệu hồng ngoại và điều khiển máy chiếu theo các lệnh tương ứng. Đảm bảo mã nguồn linh hoạt và dễ hiểu.
- Thử nghiệm và gỡ lỗi: Thực hiện các bước thử nghiệm để đảm bảo tính ổn định của hệ thống. Gỡ lỗi để xác định và khắc phục mọi vấn đề phát sinh trong quá trình thực hiện.
- Làm mạch, hộp: Xây dựng mạch điện trên PCB và lắp đặt vào hộp để tạo thành một sản phẩm hoàn chỉnh và đảm bảo tính thẩm mỹ và an toàn của sản phẩm.

1.4 Tóm tắt bố cục của báo cáo

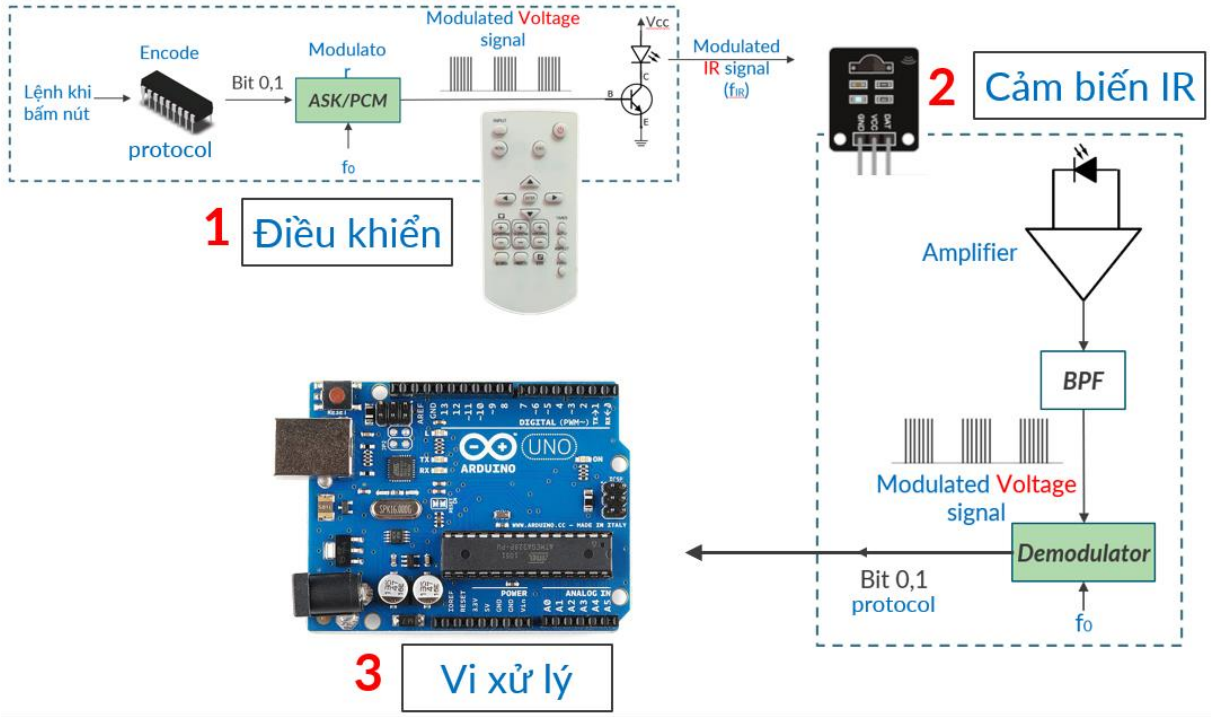
Báo cáo của nhóm chúng em bao gồm những phần chính sau:

- Tổng quan truyền thông bằng tín hiệu hồng ngoại của hệ thống (về sơ đồ tổng quan, sóng hồng ngoại, cách thu phát tín hiệu bằng hồng ngoại)
- Spec hệ thống và hệ thống con (đề xuất hệ thống, lựa chọn linh kiện, giải mã tín hiệu hồng ngoại của YT-150, Mô-đun hóa, phân rã mô-đun)
- Module giải mã tín hiệu hồng ngoại
- Module hiển thị lên LCD
- Triển khai, kiểm thử và làm sản phẩm
- Kết quả và Kết luận

2. Tổng quan truyền thông bằng tín hiệu hồng ngoại của hệ thống

2.1 Sơ đồ tổng quan

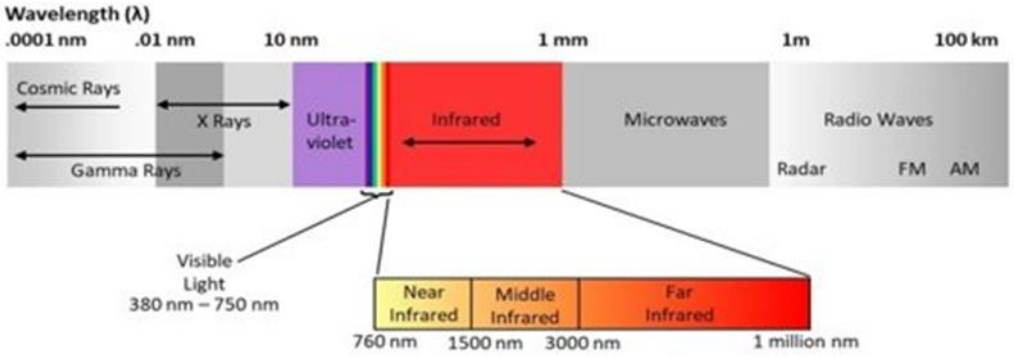
Sơ đồ tổng quan về thu phát tín hiệu hồng ngoại của hệ thống được thể hiện trong Hình 2.1.1



Hình 2.1.1. Sơ đồ tổng quan về hệ thống thu phát hồng ngoại.

- Điều khiển từ xa (1): Khi người dùng bấm nút, sẽ tạo tín hiệu điện được mã hóa cho nút bấm (theo giao thức (protocol) được chọn). Tín hiệu điện này sẽ được điều chế và làm sáng LED theo mức logic của nó. LED sáng tắt theo tín hiệu được điều chế mang theo thông tin.
- Cảm biến hồng ngoại (IR Sensor) (2): Thu tín hiệu hồng ngoại được điều chế, lọc với tần số sóng mang (f_0) mà bên phát phát đi để thu tín hiệu điện được điều chế. Sau đó, thực hiện giải điều chế ra được dòng bit 0,1 mà bên phát phát đi.
- Vi xử lí (3): Thu được dòng bit 0,1 từ cảm biến hồng ngoại, thực hiện giải mã dòng bit trên theo giao thức (protocol) mà bên phát mã hóa, thu được mã lệnh. Vi xử lí sẽ đối chiếu với bảng mã lệnh để thu được lệnh mà điều khiển yêu cầu.

2.2 Sóng hồng ngoại:



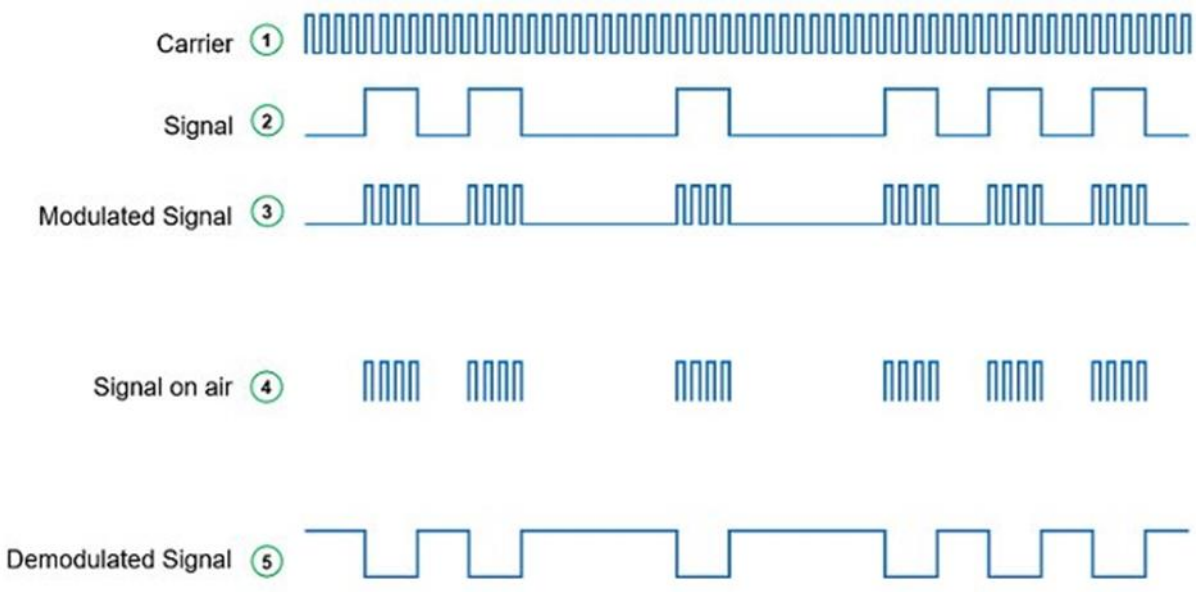
Hình 2.2.1. Thang sóng điện từ.

Sóng hồng ngoại là sóng điện từ có bước sóng 700 nm đến 1000 nm, tương ứng với tần số từ 300 GHz đến 430 GHz.

Sóng hồng ngoại có các đặc điểm: an toàn với con người, chi phí thấp, khả năng truyền thông đa dạng (có thể điều chế biên độ (Amplitude Modulation) hoặc điều chế xung (Pulse-Width Modulation) nên được ứng dụng rộng rãi trong các ứng dụng truyền thông không dây (remote controls), camera giám sát,...

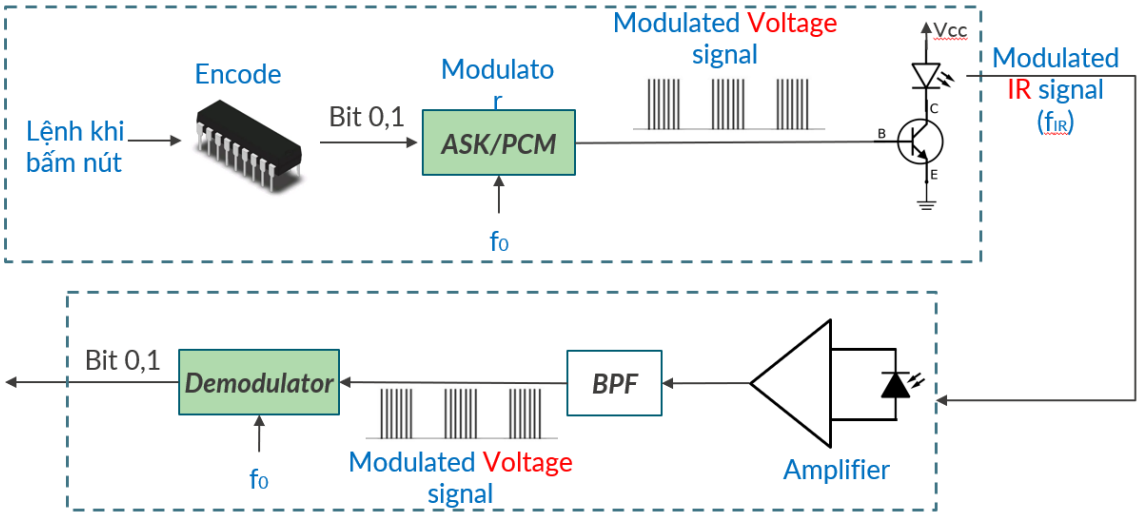
Tuy nhiên nhược điểm của việc ứng dụng sóng hồng ngoại là chỉ sử dụng được trong phạm vi hẹp, dễ hấp thụ và không thể xuyên qua các vật cản đục. Đặc biệt, do mọi vật có thể phát ra ánh sáng hồng ngoại nên khi thu phát tín hiệu hồng ngoại có thể bị nhiễu. Do đó để điều chế khi thu phát sóng hồng ngoại để giảm nhiễu và đảm bảo bên thu thu đúng tín hiệu.

2.3 Cách thu phát tín hiệu hồng ngoại



Hình 2.3.1. Điều chế và giải điều chế tín hiệu hồng ngoại.

Sóng hồng ngoại có thể được điều chế bằng nhiều cách. Một trong những cách phổ biến là nhân tín hiệu phát (signal ở bước 2 Hình 2.3.1) với sóng mang xung vuông (f ~ 38kHz) (carrier ở bước Hình 2.3.1) trước khi phát. Khi đó, tín hiệu được điều chế (modulated signal ở bước 3 Hình 2.3.1) có dạng sóng mang (với các tín hiệu phát mức logic cao) và có dạng tín hiệu mức thấp (với các tín hiệu phát mức logic thấp).



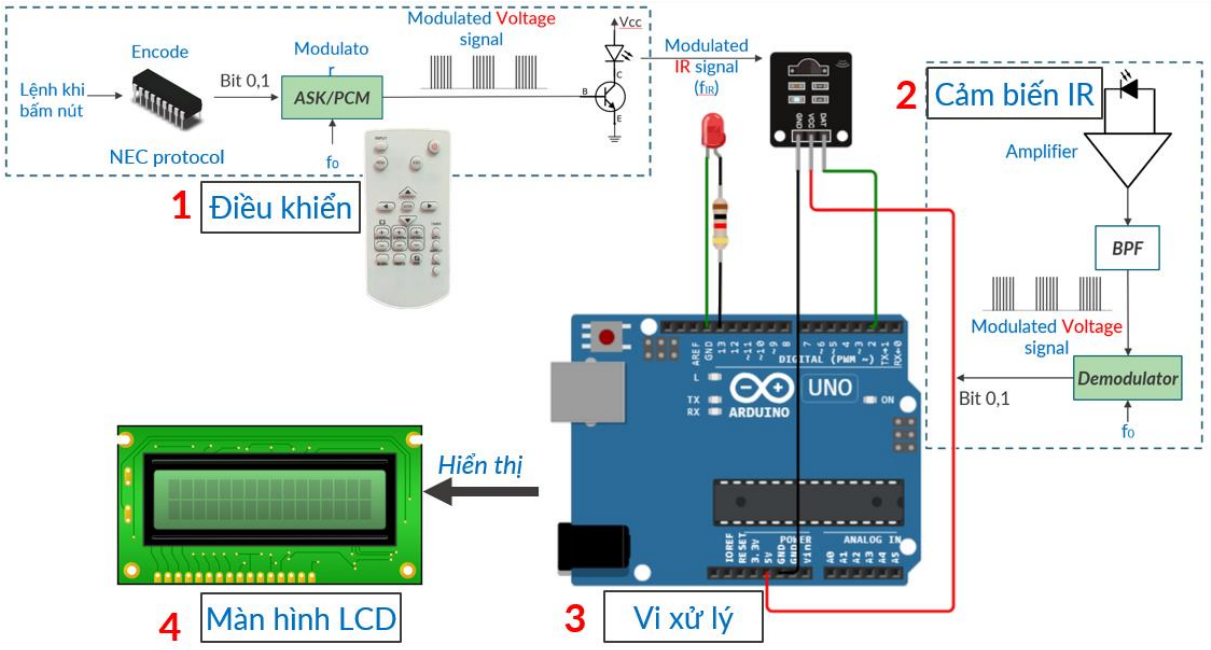
Hình 2.3.2. Hệ thống điều chế và giải điều chế tín hiếu hồng ngoại.

Tín hiệu điện này được đưa vào LED hồng ngoại. LED sẽ sáng ở những mức logic cao và tắt ở mức logic thấp, từ đó truyền thông tin tới bên thu bằng ánh sáng hồng ngoại. Ở bên thu (Hình 2.3.2) do cảm biến hồng ngoại thực hiện cảm biến thu hồng ngoại chuyển đổi tín hiệu dạng ánh sáng sang tín hiệu dạng điện chính là tín hiệu sau khi được điều chế (modulated voltage signal). Bên thu sẽ giải điều chế tín hiệu điện đó, thu được dòng bit 0,1. Khi biết được giao thức mã hóa, bên thu sẽ khôi phục được thông tin bên phát truyền đi.

3. Đặc tả hệ thống, phân tích tín hiệu hồng ngoại và phân rã mô-đun

3.1 Đề xuất hệ thống

Để giải quyết bài toán được đặt ra, nhóm đề xuất một hệ thống như sơ đồ được trình bày Hình 3.1.1. Cụ thể, mô hình hệ thống nhóm đề xuất gồm:



Hình 3.1.1. Sơ đồ hệ thống đề xuất.

- Điều khiển từ xa (1): Khi người dùng bấm nút, sẽ tạo tín hiệu điện được mã hóa cho nút bấm (theo một giao thức đã được quy định). Tín hiệu điện này sẽ được điều chế và làm sáng LED theo mức logic của nó. LED sáng tắt theo tín hiệu được điều chế mang theo thông tin. Điêu khiển từ xa này

chính là điều khiển trên giảng đường (YT-150) hoặc là điều khiển tương tự được điện thoại thông minh mô phỏng (đã kiểm thử).

- Cảm biến hồng ngoại (IR Sensor) (2): Thu tín hiệu hồng ngoại được điều chế, lọc với tần số sóng mang mà bên phát phát đi để thu tín hiệu điện được điều chế. Sau đó, thực hiện giải điều chế ra được dòng bit 0,1 mà bên phát phát đi.
- Vi xử lí (3): Thu được dòng bit 0,1 từ cảm biến hồng ngoại, thực hiện giải mã dòng bit trên theo giao thức đã quy định và thu được mã lệnh. Vi xử lí sẽ đối chiếu với bảng mã lệnh để thu được lệnh mà điều khiển yêu cầu và tiến hành điều khiển tải tương ứng, giúp mô phỏng hoạt động của máy chiếu
- Mô đun hiển thị (4): Vi điều khiển sẽ điều khiển để mô đun hiển thị hiển thị các trạng thái tương ứng của máy chiếu cho người dùng sau mỗi nút bấm.

3.2 Lựa chọn linh kiện

Cân nhắc yêu cầu bài toán đặt ra và đặc điểm của điều khiển YT-150 cũng như một vài yếu tố khác, nhóm tiến hành lựa chọn linh kiện cần sử dụng để triển khai hệ thống do nhóm đề xuất:

1. ***Cảm biến thu hồng ngoại***: Do điều khiển YT-150 sử dụng sóng mang có tần số ~38kHz để phát tín hiệu hồng ngoại nên nhóm cần phải chọn cảm biến thu có tần số sóng mang tương ứng. Sau khi khảo sát độ khả dụng, phổ biến của các cảm biến thu IR, nhóm quyết định sử dụng cảm biến ***1838T***
2. ***Vi điều khiển***: Khi lựa chọn vi điều khiển, ta cần quan tâm tới một vài yếu tố như là vi điều khiển 8 bit (theo yêu cầu của giảng viên), tốc độ xung nhịp, số chân I/O, bộ nhớ và trình quản lý ngắt (đặc biệt trong bài toán giải mã tín hiệu). Do thành viên trong nhóm có sẵn board Adruino Uno R3 nên nhóm quyết định sử dụng board ***Adruino Uno R3*** với các thông số:

- Chip ***ATMega328P*** 8 bit, có tần số hoạt động lên tới 16MH
- 16 chân IO
- Hỗ trợ 2 timer (8 bit và 16 bit)
- Bộ nhớ: 32KB Flash, 2KB SRAM và 1KB EEPROM

3. ***Mô-đun hiển thị***: Khi chọn mô đun hiển thị, nhóm cân nhắc về giá cả và tính phổ biến của linh kiện nên nhóm quyết định sử dụng mô-đun ***LCD 16x2***, một mô đun phổ biến cho sinh viên khi lập trình nhúng. Đi kèm theo đó là mô-đun I2C để kết nối LCD với vi điều khiển qua ít chân hơn (tiết kiệm số chân IO của vi điều khiển)

3.3 Phân tích tín hiệu hồng ngoại của YT-150

3.3.1 Giao thức và bảng mã của YT-150

Để phân tích, tìm ra giao thức cũng như bảng mã của điều khiển YT-150; nhóm có đề xuất các giải pháp như sau và tiến hành khảo sát:

Tìm kiếm các datasheet và tài liệu chính thức của YT-150: Nhóm đã tiến hành khảo sát và tìm kiếm trong các tài liệu được CASIO công bố nhưng không tìm được thông tin về giao thức điều khiển sử dụng

Phân tích tín hiệu hồng ngoại bằng logic analyzer: Nhóm tiến hành khảo sát các bước sử dụng logic analyzer để phân tích tín hiệu, Tuy nhiên nhóm nhận thấy logic analyzer có giá trên thị trường khoảng 180.000 VNĐ tới 220.000 VNĐ nên nhóm quyết định chuyển sang cách thứ ba.

Sử dụng vi điều khiển và các thư viện có sẵn các protocol thông dụng để thử giải mã. Nếu không giải mã được (tức không phải các protocol thông dụng), nhóm sẽ tiến hành sử dụng ngắt để theo dõi chân GPIO kết nối với IR Receiver và in ra từng khoảng thời gian khi giá trị logic trên chân IO thay đổi, qua đó thu được dữ liệu thô và tiến hành phân tích.

Sau khi tiến hành phân tích theo cách 3, nhóm thu được kết quả là điều khiển YT-150 sử dụng giao thức ***NEC*** (cụ thể là NEC2) và thử nghiệm thu được bảng mã như Bảng 3.3.1. Bảng mã lệnh của YT-150

3.3.

Command (DEC)	**Command (HEX)**	**Nút bấm**
10	0x0A	INPUT
11	0x0B	POWER
12	0x0C	MENU
14	0x0E	ESC
42	0x2A	KEYSTONE +
43	0x2B	D-ZOOM +
44	0x2C	VOLUME +
45	0x2D	KEYSTONE -
46	0x2E	D-ZOOM -
47	0x2F	VOLUME -
58	0x3A	BLANK
59	0x3B	FREEZE
60	0x3C	ECO
74	0x4A	UP
75	0x4B	DOWN
76	0x4C	ENTER
77	0x4D	LEFT
78	0x4E	RIGHT

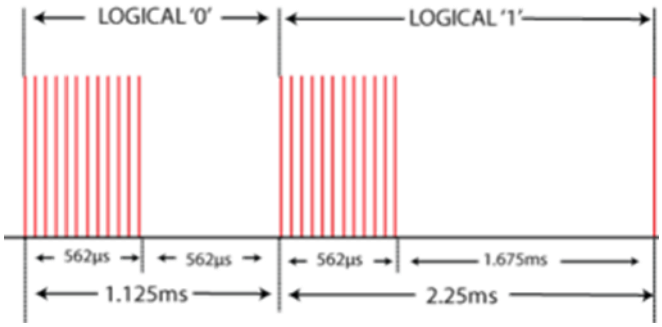
90	0x5A	TIMER
91	0x5B	AUTO
92	0x5C	ASPECT
93	0x5D	FUNC

Bảng 3.3.1. Bảng mã lệnh của YT-150

3.33.3

3.3.2 Lý thuyết về giao thức NEC

Đối với việc thực hiện thu phát tín hiệu từ điều khiển YT-150 sử dụng sóng hồng ngoại, tín hiệu sẽ được mã hóa theo giao thức NEC, được điều chế khoảng cách xung (PDM), rồi được phát, thu, giải điều chế. Phía thu do cảm biến hồng ngoại thực hiện, tín hiệu thu bởi diode thu hồng ngoại sẽ được khuyếch đại trước khi qua bộ lọc thông dải để loại bỏ nhiễu. Bộ lọc thông dải sẽ lọc những tín hiệu trong dải tần của tần số sóng mang mà bên phát đã phát (~38kHz), cho ra tín hiệu điện đã được điều chế. Tín hiệu đó qua bộ giải điều chế sẽ thu được dữ liệu (dòng bit 0,1).



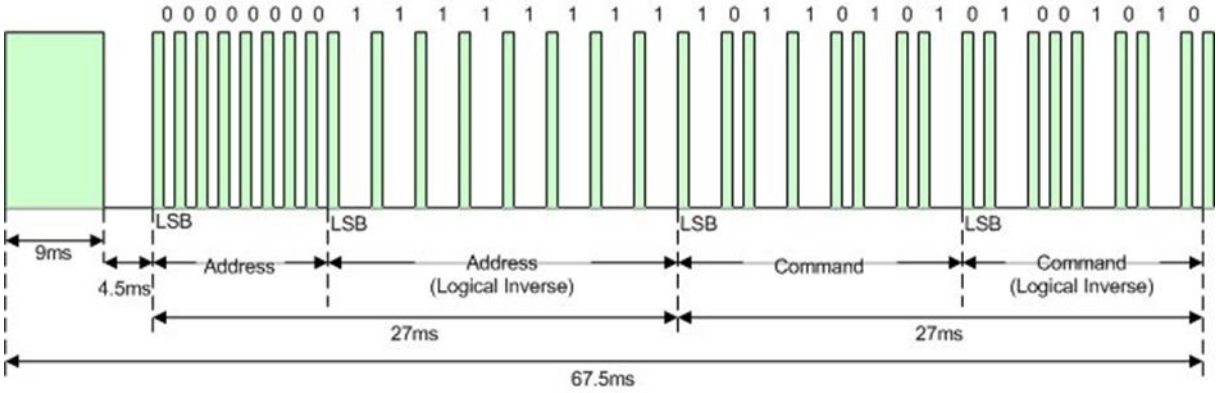
Hình 3.3.1. Tín hiệu điều chế

Qua việc nghiên cứu và tìm kiếm tài liệu, công cụ có sẵn, nhóm tìm ra được giao thức mà điều khiển máy chiếu module YT-150 sử dụng là giao thức NEC.

Giao thức NEC sử dụng điều chế khoảng cách xung (Pulse Distance Encoding). Trong đó 1 burst (1 khoảng xung / 1 khoảng trống) có độ dài 562.5 micro giây. Mức logic thấp (mức ‘0’) được điều chế thành 1 burst xung (pulse burst) và sau đó là 1 burst trống (space burst). Mức logic cao (mức ‘1’) được điều chế thành 1 burst xung và sau đó là 3 burst trống (1.675 ms).

START	ADDRESS	~ADDRESS	COMMAND	~COMMAND	STOP

Hình 3.3.2. Cấu trúc của 1 khung truyền



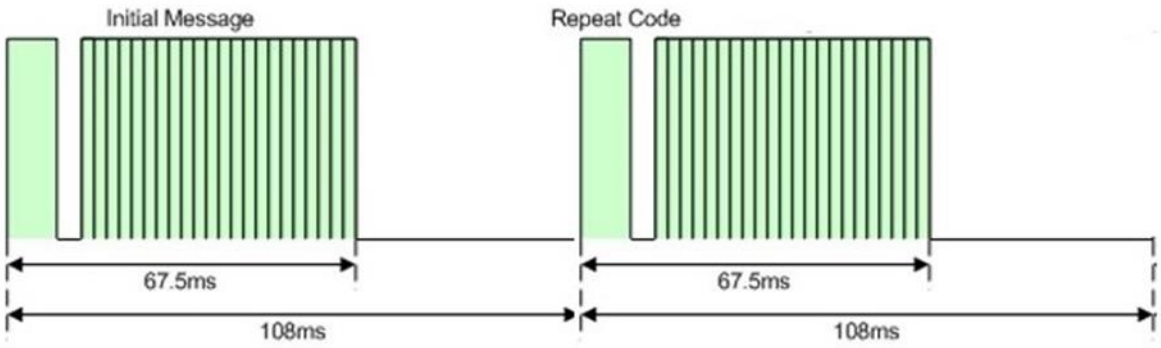
Hình 3.3.3.Ví dụ về 1 khung truyền

Trong một khung truyền (1 frame), giao thức NEC sẽ truyền Vùng bắt đầu (Start), vùng địa chỉ (address), vùng địa chỉ đảo (inverse address), vùng lệnh (command), vùng lệnh đảo (inverse command) và 1 pulse burst (0.562 ms) kết thúc (Stop) như hình vẽ. Cụ thể:

- Vùng bắt đầu (Start): Truyền 9ms các pulse burst, sau đó truyền 4.5 ms các space burst
- Vùng địa chỉ (Address): truyền địa chỉ phía phát (nếu có) dưới dạng các logical ‘0’, logical ‘1’ như **Error! Reference source not found.**.
- Vùng địa chỉ đảo (Inverse Address): truyền các logical ‘0’, logical ‘1’ đảo ngược của vùng trên. Ví dụ ở Vùng địa chỉ truyền các logical: 0 1 0 thì

Vùng địa chỉ đảo truyền: 1 0 1. Tổng thời gian vùng địa chỉ và vùng địa chỉ đảo là 27ms.

- Vùng lệnh (Command): Truyền lệnh dưới dạng các logical ‘0’, ‘1’
- Vùng lệnh đảo (Inverse Command); truyền các logical ‘0’, logical ‘1’ đảo ngược của vùng trên. Tổng thời gian vùng lệnh và vùng lệnh đảo là 27ms.
- Tổng thời gian 1 khung truyền: 9 +4.5 + 27 + 27 = 67.5 ms (bỏ qua Stop)

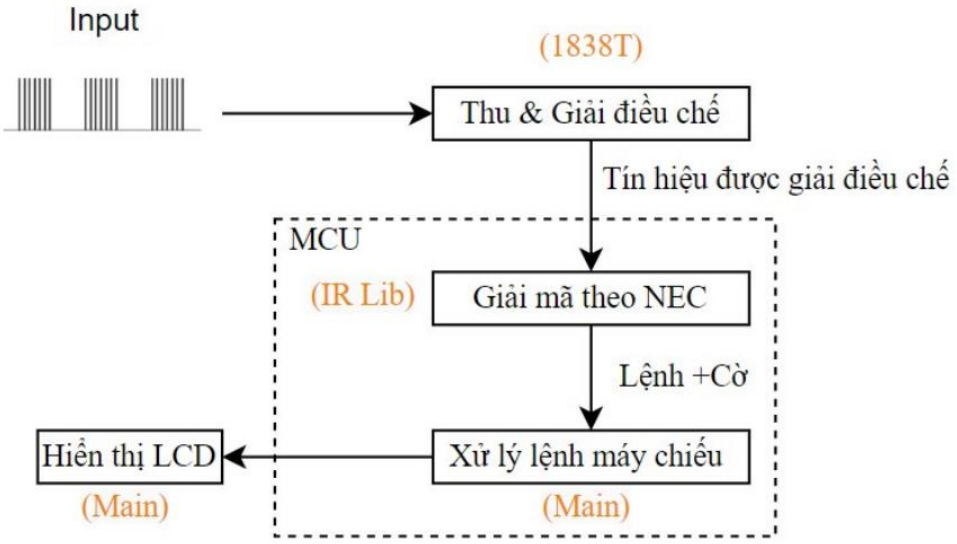


Hình 3.3.4. Khung truyền lặp

Khi truyền lặp lại một lệnh (ví dụ khi người dùng giữ phím tăng giảm âm lượng), khung truyền sẽ kéo dài ~ 108ms, trong đó truyền ~40 ms mức logic thấp (“0”). Với NEC2, khung truyền phía sau được lặp lại giống hệt khung truyền phía trước nếu tiếp tục truyền cùng một lệnh như hình Hình 3.3.4.

3.4 Mô-đun hóa, phân rã mô-đun

Để thực hiện các chức năng và hệ thống như đã đề xuất, nhóm tiến hành phân rã mô-đun để thực hiện các chức năng như trong Hình 3.4.1.



Hình 3.4.1. Sơ đồ phân rã chức năng.

a. Để thực hiện chức năng thu và giải điều chế, ta sử dụng cảm biến 1838T để thu và giải mã hồng ngoại luôn.

- Đầu vào: Tín hiệu đã được điều chế từ điều khiển lan truyền trong không khí.
- Đầu ra: xung tín hiệu số đã được giải điều chế, với 2 mức High và Low.
    - High: 4.7V-5V.
    - Low: 0.4V (mức low).
- Đầu ra này được nối tới chân IO của vi điều khiển và chân IO đó sẽ được cấu hình là Input PIN và được theo dõi để gọi ngắt khi thay đổi giá trị.

b. Việc giải mã tín hiệu thu được sau điều chế theo protocol NEC sẽ được lập trình trong một thư viện tên ***IRLib.h*** (thư viện này sẽ được đề cập kỹ hơn ở phần 4).

- Đầu vào: Tín hiệu sau giải điều chế, đầu ra của cảm biến 1838T.
- Đầu ra: Mã lệnh, địa chỉ và các cờ ứng với tín hiệu vừa thu.
- Tổng quan hoạt động: Từ đầu vào được theo dõi bằng ngắt, dựa vào mô hình FSM để giải mã ra phần header, nội dung của bản tin, qua đó

lưu các trường thông tin giải mã được vào trong một cấu trúc/biến để vi xử lý sau đó.

c. Việc xử lý tín hiệu sau khi thu được mã lệnh và cờ từ mô-đun giải mã hồng ngoại cũng như điều khiển để hiển thị lên LCD sẽ được thực hiện bởi một mô-đun con được viết thẳng vào trong ***main.cpp*** (phần này sẽ được trình bày kỹ hơn ở phần 5).
    - Đầu vào: Mã lệnh và cờ, thông tin sai khi được giải mã.
    - Đầu ra: Lệnh hiển thị lên LCD. LCD nhận được lệnh và sẽ hiển thị trạng thái tương ứng của máy chiếu
    - Hoạt động: Dựa vào trạng thái hiện tại của máy chiếu (LCD) và nút vừa được bấm mà hiển thị trạng thái kế tiếp tương ứng, mô phỏng đúng với máy chiếu. Lưu ý ở mô-đun này cần kiểm soát được việc các nút nào có thể giữ và nút nào dù giữ cũng chỉ hoạt động như 1 lần ấn. Ví dụ, khi máy chiếu đang tắt, dù giữ nút POWER thì cũng chỉ nhận là 1 lần bấm POWER để bật máy chứ không bật tắt liên tục. Tuy nhiên khi giữ nút VOLUMN+ thì cần nhận ra là giữ nút để tăng âm lượng liên tục.

4. Mô-đun giải mã tín hiệu hồng ngoại

Như đã giới thiệu sơ qua ở mục 3, mô-đun giải mã tín hiệu hồng ngoại sẽ được viết trong thư viện ***IRLib.h***, có nhiệm vụ nhận tín hiệu đầu vào là đầu ra của cảm biến thu và giải điều chế hồng ngoại và dựa vào độ rộng các khoảng 0 (space) và 1 (pulse) để giải mã thành các bit tương ứng theo giao thức NEC.

Để sử dụng thư viện này, người dùng chỉ cần thêm (include) thư viện trong hàm main, định nghĩa (define) các macro yêu cầu (như về chân vi xử lý sẽ sử dụng để nối với đầu ra của cảm biến IR). Các thông tin của frame vừa truyền sẽ được lưu trong biến ***ir_receive_data***. Biến này được khai báo ngay trong thư viện IRLib.h và có kiểu *ir_receiver_data_cb_t* (kiểu dữ liệu này dùng để lưu thông tin sau giải mã, sẽ được đề cập kỹ hơn sau).

4.1 Các macro trong IRLib

Bảng 4.1.1 thể hiện các macro và ý nghĩa các macro được sử dụng trong thư viện. Trong đó các macro có hậu tố *_S* là các state của máy trạng thái hữu hạn (FSM) mà nhóm sử dụng để giải mã, *_FLG* là các macro cho cờ của tín hiệu (như cờ lỗi, cờ lặp, ...) và tiền tố *NEC_* là các macro dựa theo độ rộng xung của tín hiệu theo giao thức NEC (tính theo micro giây) để tiện sử dụng khi lập trình, dễ đọc và bảo trì sau này.

Macro	Giá trị	Mô tả
Giao thức NEC		
NEC_BURST	562	Độ rộng 1 khoảng xung (burst) của NEC tính theo micro giây
NEC_ADDRESS_LEN	16	Số bit địa chỉ (cả bit đảo) trong 1 khung truyền của NEC
NEC_COMMAND_LEN	16	Số bit lệnh (cả bit đảo) trong 1 khung truyền của NEC
NEC_DATA_LEN	32	Tổng số bit trong vùng data (tính là địa chỉ và lệnh) trong 1 khung truyền của NEC
NEC_START_PULSE	16*562	Độ rộng của xung AGC báo hiệu bắt đầu 1 khung truyền NEC (xấp xỉ 9000 micro giây)
NEC_START_SPACE	8*562	Độ rộng của khoảng trống sau AGC (xấp xỉ 4500 micro giây)
NEC_BIT_PULSE	562	Duration of a single bit pulse in a NEC2 frame.
NEC_ONE_SPACE	3*562	Duration of the space for logical 1 in a NEC2 frame (nearly 1690 microseconds).
NEC_ZERO_SPACE	562	Duration of the space for logical 0 in a NEC2 frame.
NEC_REPEAT_INTERVAL	110000	Interval between repeated frames (110000 microseconds).

Các trạng thái của FSM		
RECEIVER_IDLE_S	0	Trạng thái chờ (khi chưa có tín hiệu nào hoặc sau khi giải mã xong 1 khung). Khi ở trạng thái này, FSM sẽ chờ nhận được 1 sườn âm ở đầu vào (tức data nhận được là bit 1) để chuyển trạng thái.
RECEIVER_WAIT_START_SPACE_S	1	Trạng thái sau khi nhận được sườn âm (để bắt đầu AGC), chờ tiếp sườn dương để kết thúc AGC và bắt đầu khoảng trống (dài 4.5 micro giây) sau đó.
RECEIVER_WAIT_FIRST_DATA_PULSE_S	2	Trạng thái sau khi nhận được sườn dương từ trạng thái trước đó (từ trạng thái số 1, đã bắt đầu vùng space ở header), chờ tiếp sườn âm để kết thúc phần space ở header và bắt đầu xung burst đầu tiên của phần data sau đó.
RECEIVER_WAIT_DATA_SPACE_S	3	Trạng thái sau khi nhận được sườn âm từ trạng thái trước đó (báo hiệu đã nhận xong phần burst của data), chờ tiếp sườn dương để kết thúc phần burst và bắt đầu vùng space của một ký hiệu.
RECEIVER_WAIT_DATA_PULSE_S	4	Trạng thái sau khi nhận được sườn dương từ trạng thái trước đó (báo hiệu đã nhận xong phần space của data), chờ tiếp sườn âm để kết thúc phần space và bắt đầu vùng burst của một ký hiệu.
Các cờ của khung truyền		
IR_EMPTY_FLG	0	Cờ E, báo hiệu chưa truyền khung tín hiệu nào

IR_NEW_CODE_FLG	1	Cờ N, báo hiệu nút bấm mới mới được bấm lần đầu (sau một khoảng thời gian bấm lại 1 nút vẫn sẽ có cờ N)
IR_REPEAT_CODE_FLG	2	Cờ R, là cờ lặp báo hiệu nút vừa giải mã là nút đang được người dùng giữ
IR_PARITY_ERROR_FLG	3	Cờ E, là cờ báo hiệu nút vừa giải mã bị lỗi khi kiểm tra chẵn lẻ phần dữ liệu

Bảng 4.1.1. Các macro được sử dụng trong thư viện IRLib.

4.2 Các cấu trúc và hàm trong IRLib

Trong thư viện, nhóm sử dụng các cấu trúc (struct, union) tự định nghĩa để lưu trữ dự liệu như Hình 4.2.1.

```
typedef union ir_data_t
{
    struct
    {
        uint8_t address;
        uint8_t inv_address;
        uint8_t command;
        uint8_t inv_command;
    } ir_data_field;
    uint32_t data;
}ir_data_t;
```

```
typedef struct ir_receiver_t {
    uint32_t last_sym_change_micro;
    uint8_t current_state;
    uint8_t current_bit_index;
    uint8_t flag;
    ir_data_t raw_data;
}ir_receiver_t;
```

```
typedef struct ir_receiver_data_cb_t {
    uint8_t address;
    uint8_t command;
    uint8_t last_command;
    uint8_t flag;
    bool data_valid;
}ir_receiver_data_cb_t;
```

Hình 4.2.1. Các cấu trúc lưu trữ được định nghĩa.

- Union *ir_data_t* thể hiện cấu trúc của phần dữ liệu thô (phần data) thu được của 1 frame NEC sau khi giải mã, bao gồm 4 trường như đã trình bày ở mục *3.2.2* và trên hình, mỗi trường 8 bit. Mục đích sử dụng struct lồng vào union tạo nên bản đồ trường bit, giúp lập trình viên có thể truy cập và thay đổi cả thanh ghi hoặc từng vùng cụ thể.
- Struct *ir_receiver_t* giúp định nghĩa kiểu cấu trúc lưu trữ trạng thái hiện tại của tín hiệu đang được giải mã, gồm các trường như độ dài của 1 khoảng trông hoặc khoảng pulse gần nhất, trạng thái hiện tại, chỉ số của bit đang giải mã, các cờ và dữ liệu thô (thuộc kiểu *ir_data_t*). Biến

của kiểu struct này sẽ được cập nhật liên tục trong quá trình giải mã, được sử dụng như bộ đệm để lưu dữ liệu trong FSM.

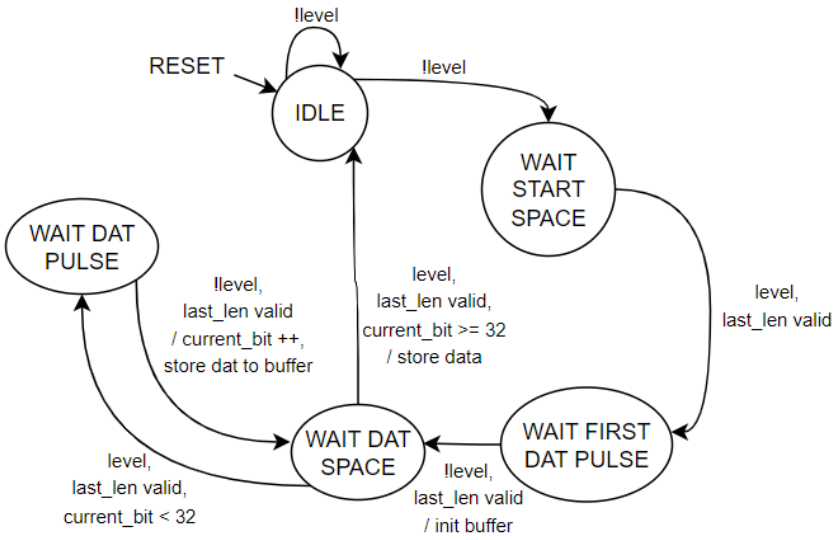
- Struct *ir_receiver_data_cb_t* giúp định nghĩa kiểu cấu trúc cho dữ liệu đã được giải mã hoàn tất của 1 khung dữ liệu NEC bao gồm các trường như địa chỉ, mã lệnh, mã lệnh trước đó, cờ và tín hiệu báo hiệu dữ liệu đã sẵn sàng để sử dụng. Các rường thông tin này sẽ được cập nhật từ giá trị của biến thuộc kiểu *ir_receiver_t* ngay sau khi giải mã thành công một khung và trước khi gọi hàm xử lý ngắt. Dữ liệu của biến có kiểu cấu trúc này sẽ được sử dụng trong các chương trình ứng dụng của hàm main để xử lý mã lệnh nhận được.

Bảng 4.2.1 thể hiện các hàm, đầu vào, đầu ra và chức năng các hàm có trong thư viện.

Hàm	**Mô tả**	**Tham số**	**Trả về**
ir_recv_init	Hàm giúp khởi tạo và cấu hình các IO cần thiết (như chân IO nối từ cảm biến tới vi điều khiển) và bật hàm xử lý ngắt	None	Kiểu Boolean, true nếu thành công và ngược lại sẽ trả về false.
isr_enable	Bật ISR	None	None
isr_disable	Tắt ISR	None	None
user_ir_isr_handle	Hàm callback để xử lý ngắt do người dùng tự định nghĩa	None	None
ir_isr_handler_cb	Hàm callback được sử dụng trong thư viện IRLib để giải mã tín hiệu hồng ngoại dựa theo FSM. Mỗi khi interrupt xảy ra, hàm này sẽ được gọi để giải mã tín hiệu.	None	None

Bảng 4.2.1. Mô tả các hàm được sử dụng trong thư viện IRLib.

4.3 Mô hình máy trạng thái hữu hạn (FSM) sử dụng để giải mã



Hình 4.3.1 Sơ đồ chuyển trạng thái cho IRLib để giải mã tín hiệu

Hình 4.3.1 mô tả sơ đồ chuyển trạng thái của FSM (FSM giúp đưa đầu ra dựa theo trạng thái hiện tại và đầu vào của FSM) nhóm sử dụng để giải mã tín hiệu hồng ngoại. Nhóm sử dụng 5 trạng thái như đã trình bày ở Bảng 5.1.1. FSM này sẽ được lập trình và chính là hàm *ir_isr_handle_cb* được gọi mỗi khi xảy ra ngắt khi mức logic trên chân GPIO nối với cảm biến.

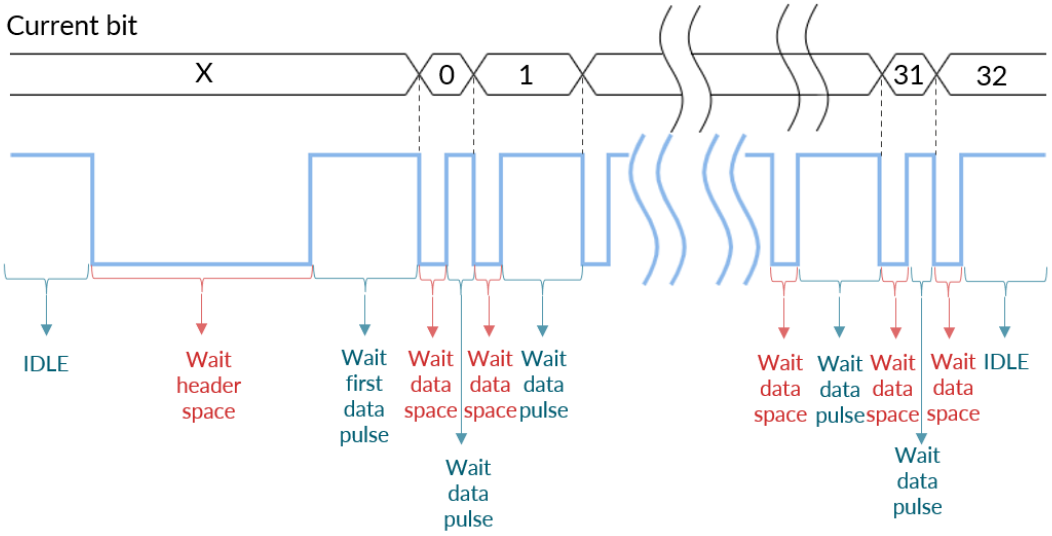
Biến *level* thể hiện mức điện áp trên chân IO. Lưu ý những khoảng mức high khi được điều chế và truyền đi, sau khi giải điều chế sẽ thu được mức low và ngược lại. Do đó khi *level* = 1 tức là phần phát đi khi chưa điều chế thực tế là một khoảng space và ngược lại.

Biến *last_len_valid* để kiểm tra xem độ rộng của xung/khoảng trống vừa nhận có hợp lệ không mỗi khi chuyển trạng thái, dựa vào trạng thái hiện tại. Khi hợp lệ mới chuyển sang trạng thái kế tiếp, nếu không thì sẽ quay trở về trạng thái *IDLE*. Ví dụ, khi trạng thái hiện tại là *WAIT_START_SPACE*, khi chuyển trạng thái sang *WAIT_FIRST_DATA_PULSE*, tức mới hoàn thành xong phần AGC. Lý tưởng thì phần AGC có độ rộng 9ms, nên để hợp lệ thì độ rộng của xung mà vi điều khiển nhận hiết được trong thực tế phải nằm trong khoảng 75%

tới 125% của độ rộng lý tưởng. Độ rộng của xung gần nhất sẽ được tính bằng hàm *micros()* của adruino.

Biến current_bit là chỉ số của bit đang xủe lý hiện tại (đếm từ 0 kể từ khi bắt đầu pulse đầu tiên của phần dữ liệu). Khi current_bit < 32, nghĩa là vẫn chưa hoàn tất 1 khung truyền, còn current_bit = 32 báo hiệu 1 khung truyền đã hoàn tất.

Các đầu ra được kí hiệu trên sơ đồ chuyển trạng thái như *init buffer* và *store data to buffer* thể hiện việc khởi tạo (khởi tạo các trường về 0) và lưu dữ liệu, các cờ vào biến có kiểu *ir_receiver_t*. Riêng quá trình *store data* thể hiện việc chuyển tất cả dữ liệu sau giải mã 1 khung từ biến kiểu *ir_receiver_t* sang biến kiểu *ir_receiver_data_cb_t* để người dùng đọc và xử ký tương ứng trong các hàm ứng dụng.



Hình 4.3.2 Ví dụ quá trình giải chuyển trạng thái trong một khung truyền

Hình 4.3.2 là một ví dụ quá trình chuyển trạng thái của FSM trong một khung truyền. Khi không nhận tín hiệu hoặc mới reset lại hệ thống, FSM sẽ ở trạng thái *IDLE* (đoạn màu xanh đầu tiên trong hình). Lúc này, FSM sẽ chờ sườn âm để chuyển trạng thái. Sau khi phát hiện một sườn âm, trạng thái chuyển sang *WAIT_HEADER_SPACE* và tiến hành kiểm tra độ dài của khoảng *IDLE* giúp

gán cờ phù hợp (cờ N hay cờ R). Sau đó, khi nhánh được một sườn dương, FSM kiểm tra độ rộng của khoảng pulse/space gần nhất có hợp lệ không, nếu hợp lệ chuyển trạng thái sang *WAIT_FIRST_DATA_PULSE* và chờ tiếp sườn âm. Sau khi nhận được sườn âm và kiểm tra thấy độ rộng hợp lệ, FSM chuyển trạng thái sang *WAIT_DATA_SPACE* và bắt đầu khởi tạo buffer, gán giá trị 0 cho các biến như *current_bit* và trường dữ liệu *raw_data*. Tiếp tục quá trình này, lưu ý mỗi khi chuyển từ trạng thái *WAIT_DATA_PULSE* sang *WAIT_DATA_SPACE*, phần mềm sẽ từ độ dài khoảng trống vừa thu được để quyết định xem là bit thu được là 0 hay 1, sau đó dịch bit đó vào trong *raw_data,* chi số *current_bit* được tăng lên 1. Khi từ trạng thái *WAIT_DATA_SPACE* chuyển sang trạng thái khác, sẽ phải kiểm tra *current_bit* xem đã hết khung truyền chưa để quyết định trạng thái kế tiếp.

4.4 Thư viện ReadWriteLib giúp đọc ghi trực tiếp baremetal

Thư viện ReadWriteLib được nhóm sử dụng để đọc ghi nhanh một chân GPIO dựa theo số pin, được sử dụng trong hàm xử lý ngắt (bởi vì trong hàm xử lý ngắt các tác vụ yêu cầu thực hiện nhanh nhất có thể).

PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Hình 4.4.1. Các thanh ghi quan trọng của GPIO

Trong ATMega328P, có ba thanh ghi quan trọng ứng với 1 chân GPIO, lần lượt như **Error! Reference source not found.** là ví dụ cho các thanh ghi của GPIO ở PORT B.

1. Thanh ghi DDR: Là thanh ghi giúp cấu hình chế độ cho chân GPIO là chân đầu vào (input) hay đầu ra (output).
2. Thanh ghi PORT: Khi GPIO được cấu hình là đầu vào thì thanh ghi này sẽ giúp cấu hình việc pull up của chân GPIO. Khi chân GPIO được cấu hình là đầu ra, giá trị trong thanh ghi này chính là mức logic đầu ra của chân GPIO.
3. Thanh ghi PIN: Là thanh ghi chỉ đọc, khi GPIO được cấu hình là đầu vào thì thanh ghi này giúp đọc trạng thái logic hiện tại trên chân GPIO. Khi cố tình viết bit 1 vào một bit trong thanh ghi này, bit tương ứng trong thanh ghi PORT sẽ bị thay đổi (toggle).

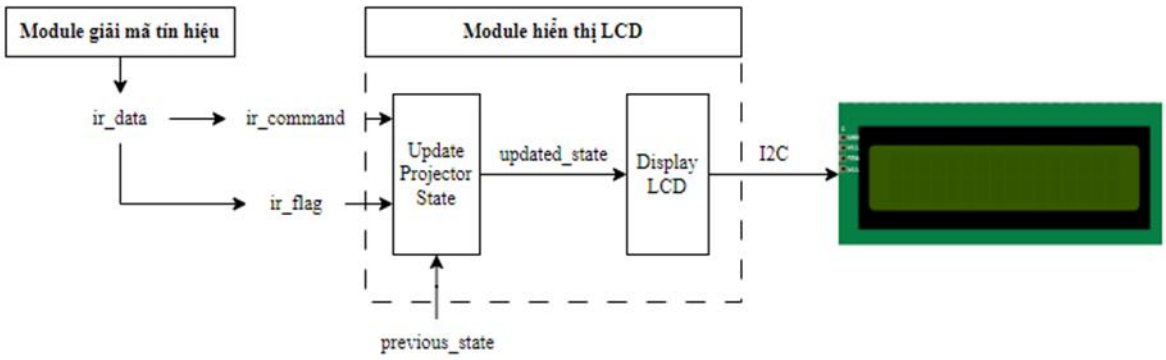
Để đọc và ghi vào thanh ghi, nhóm em sử dụng các toán tử bit-wise để chuyển từ số pin thành chỉ só bit tương ứng ở các thanh ghi DDR, PORT, PIN và để set, clear các bit trong từng thanh ghi.

5. Mô-đun hiển thị lên LCD

Như đã giới thiệu sơ qua ở mục 3, mô-đun hiển thị lên LCD sẽ được viết trong hàm ***main.cpp***, có nhiệm vụ nhận tín hiệu đầu vào là mã lệnh, cờ của biến ir_receiver_data (lưu thông tin của frame vừa truyền). Đầu ra là 4 pin nối với LCD để hiển thị trạng thái máy chiếu lên LCD. Về nhiệm vụ của module, module sẽ dựa vào trạng thái hiện tại của máy chiếu và nút vừa bấm để hiển thị trạng thái tiếp theo của máy chiếu.

5.1 Thiết kế module:

Dựa trên yêu cầu đó, nhóm thiết kế module như hình vẽ dưới đây:



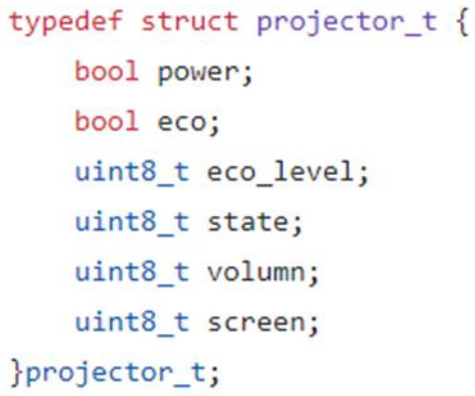
Hình 5.1.1. Sơ đồ module hiển thị lên LCD.

Trong Hình 5.1.1 biến *ir_data* lưu thông tin về frame vừa truyền (địa chỉ, mã lệnh, cờ,...). Trong đó mã lệnh (*ir_command*) và các cờ lệnh (*ir_flag*) được đưa vào **module hiển thị LCD.**

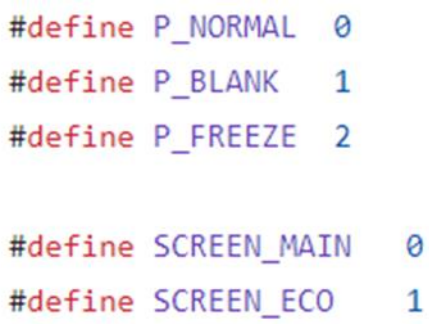
Biến *previous_state* lưu các trạng thái của máy chiếu trước khi có tín hiệu được giải mã từ **module giải mã tín hiệu**. Biến update_state lưu trạng thái của máy chiếu sau khi có tín hiệu được giải mã. Khối **Display LCD** nhận trạng thái mới của máy chiếu và hiển thị lên LCD qua giao tiếp I2C.

5.2 Các định nghĩa, cấu trúc trong module hiển thị LCD

Trong hàm ***main.cpp***, nhóm sử dụng các cấu trúc (struct, union) tự định nghĩa để lưu trữ dữ liệu như Hình 5.2.1.



Hình 5.2.1. Cấu trúc lưu trữ dữ liệu.



Hình 5.2.2. Các trạng thái của *state* & *screen.*

Để theo dõi và cập nhật, hiển thị trạng thái của máy chiếu, struct *projector*_t được sử dụng để lưu thông tin về các chế độ, trạng thái của máy chiếu. Các biến: *power, eco* được cập nhật trạng thái khi người dùng ấn phím tương ứng. Biến *state* để cập nhật trạng thái của máy chiếu: trạng thái bình thường (P_NORMAL), trạng thái tắt hình tạm thời (P_BLANK) hay trạng thái dừng hình (P_FREEZE). Biến *screen* để cập nhật máy chiếu đang chiếu màn hình chính *(SCREEN_MAIN)* hay màn hình ở chế độ Eco *(SCREEN_ECO).* Các biến *eco_level, volumn* lưu số mức tiết kiệm năng lượng và số mức âm lượng. Để dễ đọc, sửa và bảo trì, các giá trị của *state* và *screen* được khai báo trước Hình 5.2.2.

Tóm lại, máy chiếu sẽ hiển thị 1 trong 2 màn hình: Màn hình chính hoặc màn hình trong chế độ Eco, được quy định bởi biến *screen.* Trong màn hình chính *(SCREEN_MAIN),* máy chiếu ở 1 trong 3 trạng thái: *P_NORMAL* (bình thường), *P_BLANK* (tắt hình tạm thời) và *P_FREEZE* (dừng hình). Ở màn hình chính, âm lượng có thể tùy chỉnh và cập nhật bởi biến *volumn.* Máy chiếu sẽ kiểm tra biến *power* để hiển thị hoặc tắt. Tương tự máy chiếu sẽ kiểm tra biến *eco* để chuyển sang màn hình Eco hoặc về màn hình chính. Trong màn hình Eco, mức tiết kiệm

năng lượng có thể tùy chỉnh và cập nhật bởi biến *eco_level* **khi trạng thái Eco được bật.**

5.3 Khối cập nhật và hiển thị lên LCD

5.3.1 Khối cập nhật trạng thái máy chiếu

- Khối cập nhật trạng thái (**UpdateProjectorState**) nhận đầu vào là các lệnh được giải mã từ **module giải mã tín hiệu hồng ngoại** và trạng thái hiện tại của máy chiếu. Đầu ra trạng thái của máy chiếu sau khi được cập nhật.
- Hoạt động:
    - Bước 1: Kiểm tra nút Power được bấm hay không để cập nhật lại trạng thái *power*
    - Bước 2: Nếu *power* bằng 1 tức máy chiếu được bật, kiểm tra xem đang ở màn hình chính *(SCREEN_MAIN*) hay màn hình Eco *(SCREEN_ECO)*
        - Bước 3.1: Nếu ở màn hình chính, kiểm tra mã lệnh nhận được, đối chiếu với bảng **Error! Reference source not found.** để cập nhật lại trạng thái tương ứng. Chú ý kiểm tra cờ của lệnh, với một số mã lệnh tĩnh (POWER, ESC, BLANK, FREEZE, ECO,UP, DOWN, LEFT, RIGHT) trong bảng **Error! Reference source not found.** thì việc giữ phím để truyền lại 1 frame không có tác dụng. Đối với các mã lệnh động (VOLUMN+, VOLUMN-) thì ngược lại.
        - Bước 3.2: Nếu ở màn hình Eco, tương tự ta kiểm tra các mã lệnh nhận được, đối chiếu với bảng **Error! Reference source not found.** để cập nhật lại trạng thái tương ứng. Chú ý ở màn hình Eco, trạng thái Eco bật hay tắt phụ thuộc vào cả biến *eco* và chỉnh trạng thái của người dùng. VD khi *eco* = 1, người dùng ấn phím UP thì trạng thái Eco bật, người dùng ấn

phím DOWN thì trạng thái Eco tắt. Chỉ khi trạng thái Eco bật, người dùng có thể tăng giảm mức tiết kiệm năng lượng (*eco_level*) bởi 2 phím LEFT, RIGHT.

5.3.2 Khối hiển thị trạng thái máy chiếu lên LCD

- Khối hiển thị trạng thái máy chiếu nhận đầu vào là trạng thái máy chiếu sau khi được cập nhật và thực hiện in trạng thái lên màn hình LCD. Việc in lên màn hình LCD được thực hiện thông qua thư viện <***LiquidCrystal_I2C.h***> và <***Wire.h***> có sẵn. Nhóm thực hiện gọi các hàm của thư viện và truyền vào tham số là các trạng thái của máy chiếu.
- Hoạt động:
    - Bước 1: Xóa màn hình LCD
    - Bước 2: Kiểm tra trạng thái màn hình của máy chiếu là màn hình chính hay màn hình Eco
        - Bước 3.1: Nếu ở màn hình chính, hiển thị trạng thái (Normal, Blank hay Freeze) và hiển thị âm lượng (*volumn*)
        - Bước 3.2: Nếu ở màn hình Eco, hiển thị trạng thái Eco (bật hay tắt) và nếu trạng thái Eco bật, hiển thị mức tiết kiệm năng lượng (*eco_level*)

6. Triển khai, kiểm thử, gỡ lỗi và làm sản phẩm

6.1 Triển khai

Sau khi bắt đầu quá trình nghiên cứu chi tiết về các giao thức điều khiển máy chiếu hiện đại và quyết định triển khai mạch, nhóm em thực hiện theo các bước sau đây:

1. Chuẩn bị linh kiện:
   Xác định và chuẩn bị các linh kiện cần thiết để triển khai, bao gồm vi điều khiển, các thành phần điện tử, và các dây nối theo như Bảng 6.1.1.

Tên phần tử	Số lượng
Mạch vi điều khiển Arduino uno R3	1
Đèn LED	1
Diode Thu Hồng Ngoại 1838T(IR Receiver)	1
Điện trở 1k	1
Màn Hình LCD	1

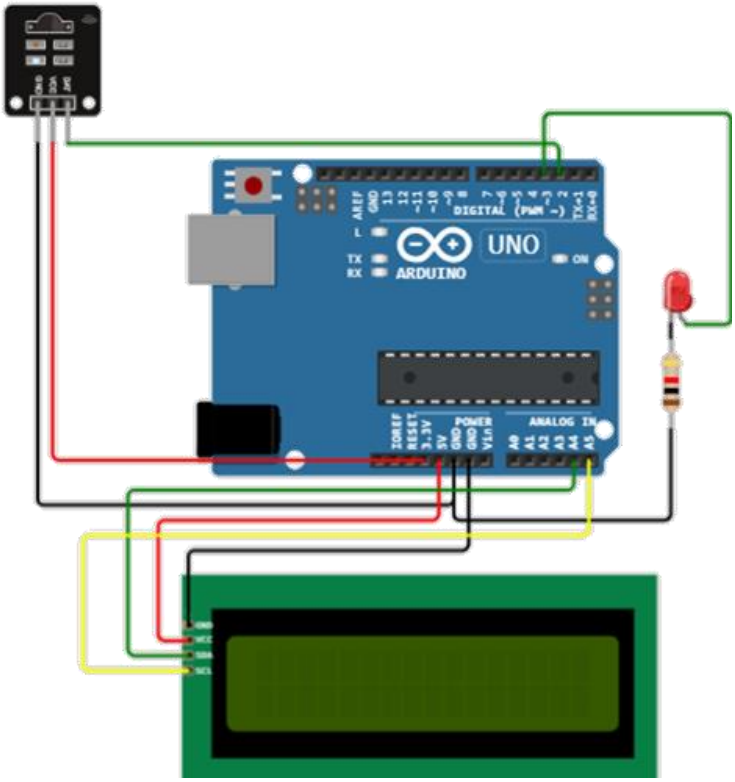
Bảng 6.1.1. Các linh kiện được sử dụng.

2. Triển khai mạch :

Nhóm thực hiện nối các chân của các linh kiên với mạch theo Bảng 6.1.2và Hình 6.1.1.

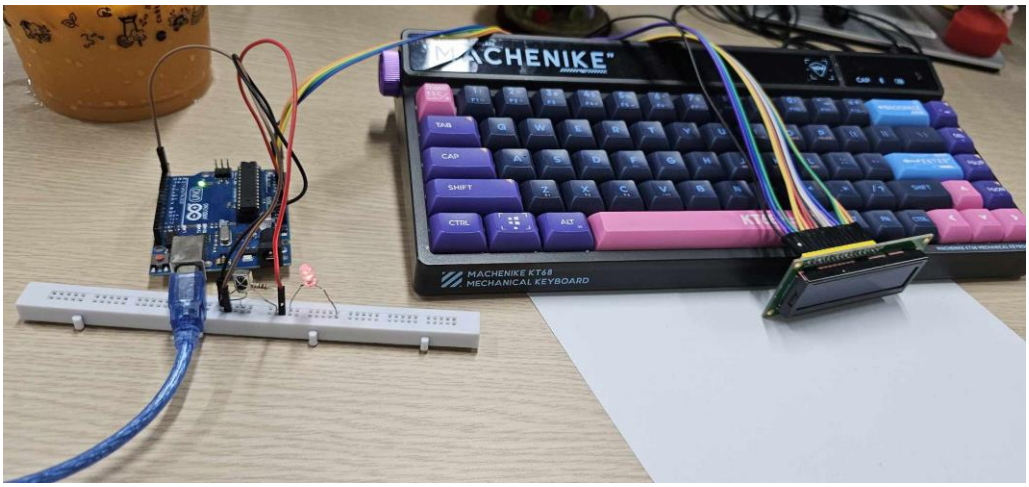
Arduino Uno Pin	I2C LCD Pin	IR Receiver
PIN3		DAT
3.3V(5V)		VCC
GND		GND
GND	GND	
5V	VCC	
PIN A4	SDA	
PIN A5	SCL	

Bảng 6.1.2. Chân được kết nối với nhau của các linh kiện.



Hình 6.1.1. Sơ đồ kết nối các linh kiện.

3. Mạch kiểm thử:



Hình 6.1.2. Mạch lắp thực tế trước khi hàn mạch.

6.2 Kiểm thử

Sau khi triển khai mạch, nhóm tiến hành kết nối nguồn của mạch qua USB từ máy tính cá nhân thực để hiện kiểm thử hệ thống mạch với điều khiển máy chiếu trên phòng học với các nút/chức năng theo Bảng 6.2.1.

Nút/Chức năng	Mô tả ngắn gọn	Kết quả
POWER	Bật/tắt máy chiếu.	Đạt
VOLUME +	Tăng âm lượng (1 bước/nhấn hoặc giữ liên tục).	Đạt
VOLUME -	Giảm âm lượng (1 bước/nhấn hoặc giữ liên tục).	Đạt
FREEZE	Chuyển máy chiếu sang chế độ FREEZE.	Đạt
BLANK	Chuyển máy chiếu sang chế độ BLANK.	Đạt
ESC	Thoát khỏi màn hình/chế độ/chức năng sang màn hình chính.	Đạt
ECO	Chuyển sang ECO SCREEN, dùng để điều chỉnh cài đặt tiết kiệm năng lượng.	Đạt
LEFT	Nút bên trái của điều khiển từ xa, dùng để thay đổi trạng thái tiết kiệm năng lượng hoặc giảm mức tiết kiệm năng lượng.	Đạt
RIGHT	Nút bên phải của điều khiển từ xa, dùng để Thay đổi trạng thái tiết kiệm năng lượng hoặc tăng mức tiết kiệm năng lượng.	Đạt
DOWN	Nút xuống của điều khiển từ xa, dùng để chọn các chức năng trong màn hình (ví dụ: điều chỉnh mức tiết kiệm năng lượng trong ECO SCREEN).	Đạt

UP	Nút lên của điều khiển từ xa, dùng để chọn các chức năng trên màn hình.	Đạt

Bảng 6.2.1. Kiểm thử các chức năng.

6.3 Sản phẩm hoàn thiện

Khi đã kiểm thử đạt hết các chức năng nhóm đã thực hiện hàn mạch và thiết kế và làm hộp cho mạch để tạo tạo thành sản phẩm hoàn thiện.

Hình 6.3.1. Sản phẩm hoàn thiện.

7. Kết quả và kết luận

7.1 Kết quả

Chúng em đã thực hiện theo yêu cầu và đưa sản phẩm đến giai đoạn hoàn thiện, với khả năng giải mã bản tin từ photodiode hồng ngoại khi điều khiển

bằng bộ điều khiển từ xa của máy chiếu trong giảng đường. Dưới đây là tổng quan về nhưng kết quả đã đạt được theo yêu cầu:

1. Giải mã bản tin với ít nhất 4 lệnh : Chúng em thực hiện giải mã 11 lệnh khác nhau từ bản tin thu được từ photodiode hồng ngoại của điều khiển từ xa của máy chiếu trên giảng đường.
2. Sử dụng vi điều khiển 8 bit : Vi điều khiển ATmega328 đã được lựa chọn để giải mã và hiển thị tín hiệu truyền từ điều khiển.
3. Lập Trình Bằng Ngôn Ngữ C : Để giải mã các tín hiệu, nhóm chúng em đã thực hiện viết thư viện IRLib. Đây là thư viện được thiết kế để phát hiện và giải mã các khung giao thức NEC2 dựa trên các ngắt thay đổi logic trong chân của ATmega328P.
4. Mã Nguồn Cho Phép Đổi Chân I/O :Chúng em đã thực hiện để cho phép người dùng đổi chân I/O tùy ý bằng lệnh "#define IR_RECEIVER_PIN ".

7.2 Kết luận

Nhóm đã tiến hành thiết kế và lập trình mô-đun giải mã tín hiệu điều khiển từ xa của máy chiếu. Tuy đạt kết quả đã đạt được theo yêu cầu và nhóm cũng học được nhiều bài học trong suốt quá trình thực hiện dự án, nhưng nhóm em nhận thấy cần có nhưng cải thiện về mặt hình thức của hệ thống cũng như tối ưu hóa và có thể thực hiện nhiều chức năng hơn của 1 hệ thống giải mã tín hiệu trên trong tương lai.

Để hoàn thành báo cáo này, chúng em xin cảm ơn sự hướng dẫn của TS. **Đào Việt Hùng** cũng như các tài liệu tham khảo (đề cập cuối báo cáo).

Tuy đã tìm hiểu và cố gắng song chúng em không tránh khỏi những thiếu sót, em mong nhận được những lời nhận xét và góp ý của thầy để hoàn thiện hơn trong tương lai.

Nhóm 6

TÀI LIỆU THAM KHẢO

[1] “Microcontrollers ATmega328P Datasheet,” [Trực tuyến]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

[2] “XJ-V1 Projector User's guide,” [Trực tuyến]. Available: https://www.projectorcentral.com/pdf/projector_manual_9089.pdf.