

Topic 5

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

Hiii



Group 6

Nguyen Phuong Linh

20206203

Tran Trung Hieu

20200230

Nguyen Ngoc Duong

20200122

Phân công nhiệm vụ

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

Phân công nhiệm vụ (Tiếp)

Nhiệm vụ		Thành viên
Viết báo cáo	Mở đầu	Nguyễn Ngọc Dương
	Truyền thông bằng tín hiệu hồng ngoại	Trần Trung Hiếu
	Đặc tả hệ thống, phân tích tín hiệu hồng ngoại và phân rã mô-đun	Nguyễn Phương Linh
	Mô-đun giải mã tín hiệu hồng ngoại	Nguyễn Phương Linh
	Mô-đun hiển thị lên LCD	Trần Trung Hiếu
	Triển khai, kiểm thử, gỡ lỗi và làm sản phẩm	Nguyễn Ngọc Dương
	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

Mục lục

01

Sơ lược yêu cầu

02

Lập trình vi điều khiển

03

Triển khai & Kết quả

1

Sơ lược yêu cầu

NEC Protocol

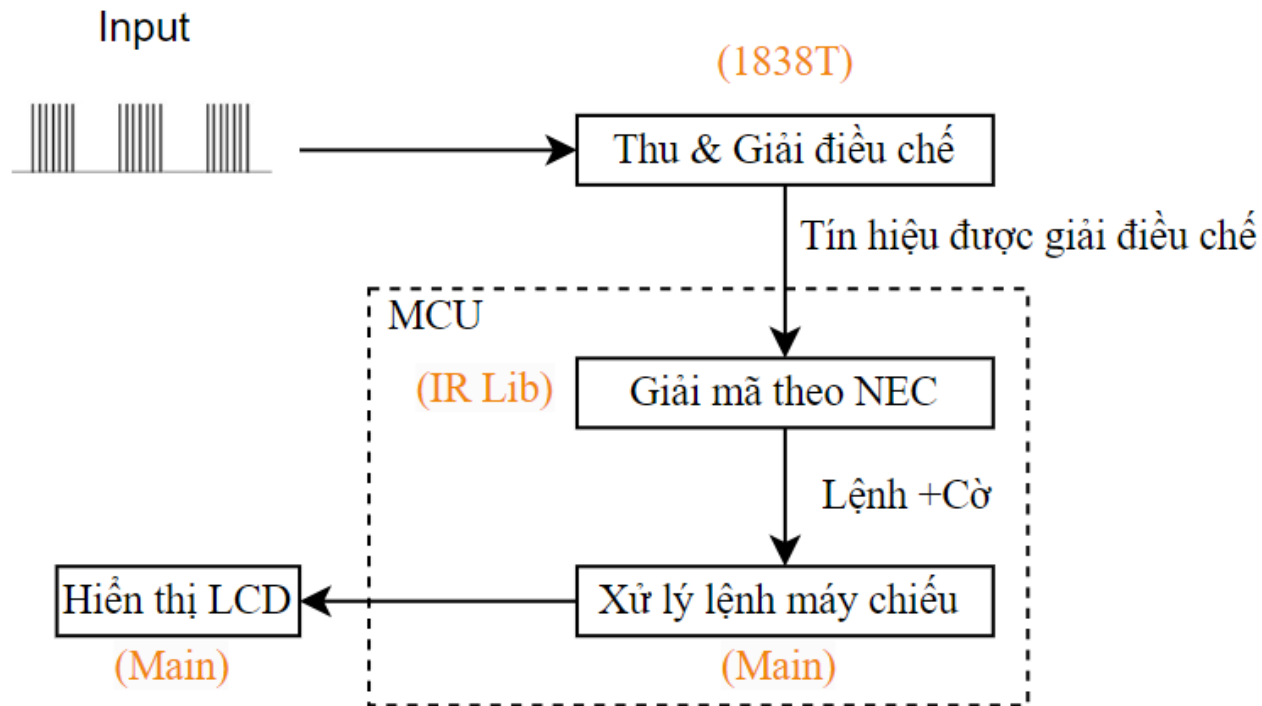
- Được điều chế với tần số sóng mang 38kHz
- Sử dụng **pulse distance encoding**

Nút bấm	Mã	Nút bấm	Mã	Nút bấm	Mã
INPUT	10	D-ZOOM -	46	LEFT	77
POWER	11	VOLUME -	47	RIGHT	78
MENU	12	BLANK	58	TIMER	90
ESC	14	FREEZE	59	AUTO	91
KEystone +	42	ECO	60	ASPECT	92
D_ZOOM +	43	UP	74	FUNC	93
VOLUME +	44	DOWN	75		
KEystone -	45	ENTER	76		

YT-150



Hệ thống đề xuất





2.1

**Chương trình
thực thi**

Các nút mô phỏng

- BẬT/TẮT nguồn: Nút POWER
- Điều chỉnh âm lượng: Nút VOLUME+ và VOLUME-
- Thay đổi chế độ: Nút FREEZE, BLANK và ESC
- Tiết kiệm năng lượng: Nút ECO, TRÁI, PHẢI, LÊN, XUỐNG và ESC

Struct

```
typedef struct projector_t {  
    bool power;  
    bool eco;  
    uint8_t eco_level;  
    uint8_t state;  
    uint8_t volumn;  
    uint8_t screen;  
}projector_t;
```

Thiết lập chân nối IR,
chân nối LED, sử dụng
hàm xử lý ngắt cho người
dùng ,, , bằng #define.



Flowchart

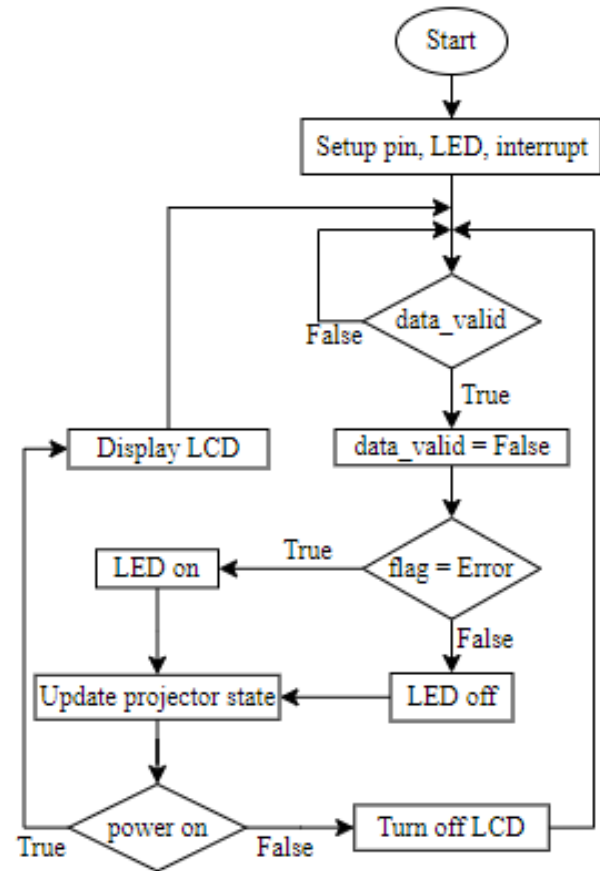
data_valid: kiểm tra xem tín hiệu IR đã nhận được có hợp lệ không

flag: parity_error_flag = 1 khi tín hiệu giải mã không hợp lệ

flag & data_valid: được cập nhật bởi mô-đun IR (IRLib)

Update projector state: kiểm tra loại "IR command" và **flag** để cập nhật trạng thái máy chiếu. Lệnh IR và **flag** được phát hiện và giải điều chế bởi mô-đun giải mã tín hiệu IR.

Display LCD: sử dụng chức năng có sẵn (thư viện <LiquidCrystal_I2C.h> để in trạng thái máy chiếu lên màn hình





2.2

**Chức năng
Read/Write**

Thanh ghi GPIO

Bit-wise: để
setup thanh ghi

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	

Input mode: config pull up

Output mode: đặt giá trị output

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	

Thiết lập chế độ
Input/ Output
cho cổng IO

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	

Dùng để đọc giá trị hiện tại
trên chân GPIO tương ứng
→ Nếu write 1 thì toggle
giá trị đang lưu trong
thanh ghi PORT

2.3

Mô-đun IR(IR Lib)

Macro

```
#define NEC_BURST          562
#define NEC_ADDRESS_LEN   16
#define NEC_COMMAND_LEN   16
#define NEC_DATA_LEN      (NEC_ADDRESS_LEN + NEC_COMMAND_LEN)
#define NEC_START_PULSE   (16 * NEC_BURST)    // nearly 9000
#define NEC_START_SPACE   (8 * NEC_BURST)      // nearly 4500
#define NEC_BIT_PULSE     NEC_BURST
#define NEC_ONE_SPACE      (3 * NEC_BURST)      // nearly 1690
#define NEC_ZERO_SPACE    NEC_BURST
#define NEC_REPEAT_SPACE   (4 * NEC_BURST)      // nearly 2250
#define NEC_REPEAT_PULSE   (16 * NEC_BURST)     // nearly 9000
#define NEC_REPEAT_INTERVAL 110000
#define NEC_MAX_REPEAT_SPACE (NEC_REPEAT_INTERVAL - NEC_START_PULSE
- NEC_START_SPACE - 65LL*NEC_BURST)
```

Struct/union

Union thể hiện Struct của dữ liệu IR nhận được

```
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;
```

Struct/union

Struct thể hiện trạng thái hiện tại, bit nhận được từ bộ thu IR (được cập nhật trong giai đoạn giải mã và được sử dụng trong FSM để giải mã).

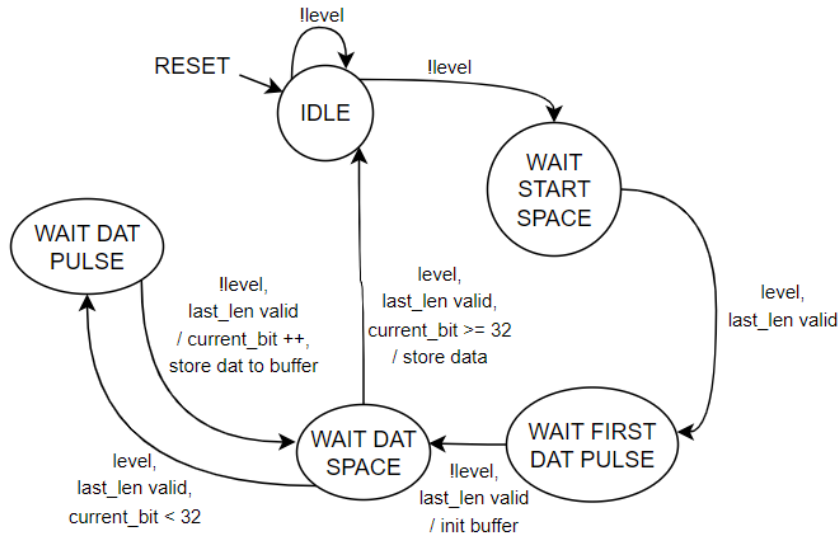
```
typedef struct ir_receiver_t {  
    uint32_t last_sym_change_micro;  
    uint8_t current_state;  
    uint8_t current_bit_index;  
    uint8_t flag;  
    ir_data_t raw_data;  
}ir_receiver_t;
```


Struct/union

Struct lưu trữ dữ liệu được giải mã. Nó sẽ được cập nhật ngay sau khi giải mã thành công một khung và trước khi gọi lại người dùng để chuyển dữ liệu nhận được vào vòng lặp chính để xử lý tiếp.

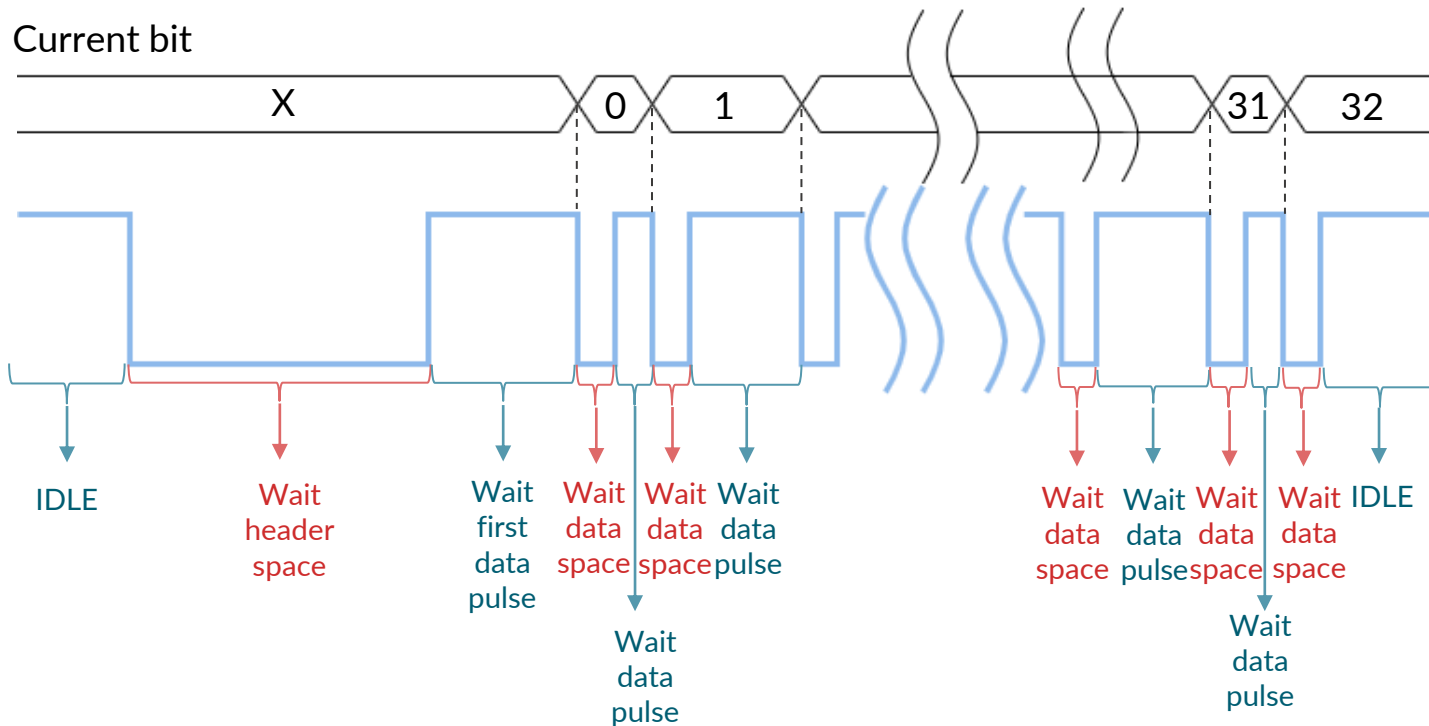
```
typedef struct ir_receiver_data_cb_t {  
    uint8_t address;  
    uint8_t command;  
    uint8_t last_command;  
    uint8_t flag;  
    bool data_valid;  
}ir_receiver_data_cb_t;
```

Sơ đồ chuyển trạng thái



Tín hiệu:

- **level:** Mức điện áp thu được trên IR RECEIVER PIN. Khi level = 1, tương ứng với bit 0 ở data.
- **last_len_valid:** Biến giúp kiểm tra độ dài của pulse/space mới đọc là hợp lệ hay không. Nếu hợp lệ last_len_valid = 1, ngược lại sẽ = 0.
- **current_bit:** Số thứ tự bit hiện tại, khi current_bit có giá trị nhỏ hơn 32 thì sẽ lưu trữ dữ liệu vào buffer, và khi đạt 32 thì dữ liệu trong buffer sẽ được đưa vào struct lưu nội dung sau giải mã của tín hiệu hồng ngoại.



Ngắt

`ir_rcv_init`: Khởi tạo bộ thu IR

`isr_enable`: Bật quy trình dịch vụ ngắt (ISR)

`ir_isr_handler_cb`: Hàm xử lý ngắt → được nối ngắt tương ứng trong bảng

vec-tơ xử lý ngắt bằng hàm `ISR()`

```
# if defined(INT0_VALID)
ISR(INT0_vect)
# elif defined(INT1_VALID)
ISR(INT1_vect)
# elif defined(PCINT0_VALID)
ISR(PCINT0_vect)
# elif defined(PCINT1_VALID)
ISR(PCINT1_vect)
# elif defined(PCINT2_VALID)
ISR(PCINT2_vect)
```

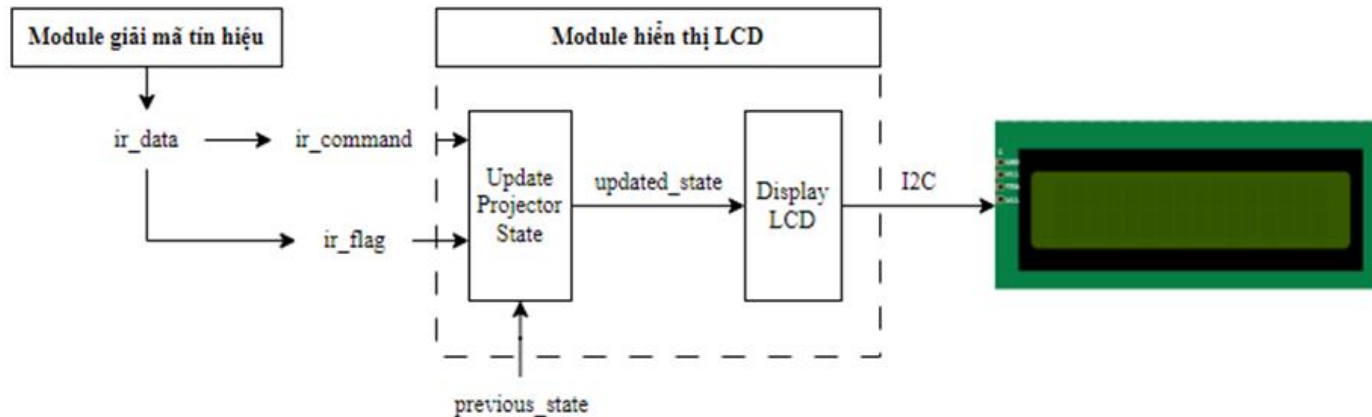


2.4

Module hiển thị LCD

Module hiển thị LCD

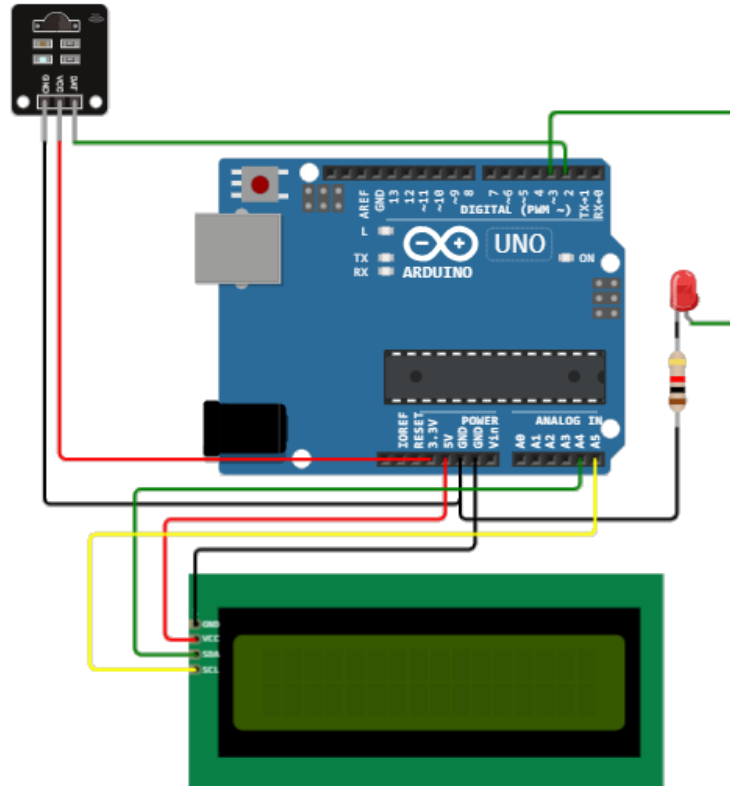
- `ir_data` chứa thông tin về frame truyền (địa chỉ, mã lệnh(`ir_command`) và các cờ lệnh(`ir_flag`)).
- `previous_state`: lưu trạng thái trước đó của máy chiếu
- `update_state`: lưu trạng thái mới của máy
- Giao tiếp I2C: hiển thị trạng thái máy mới lên màn hình.



03

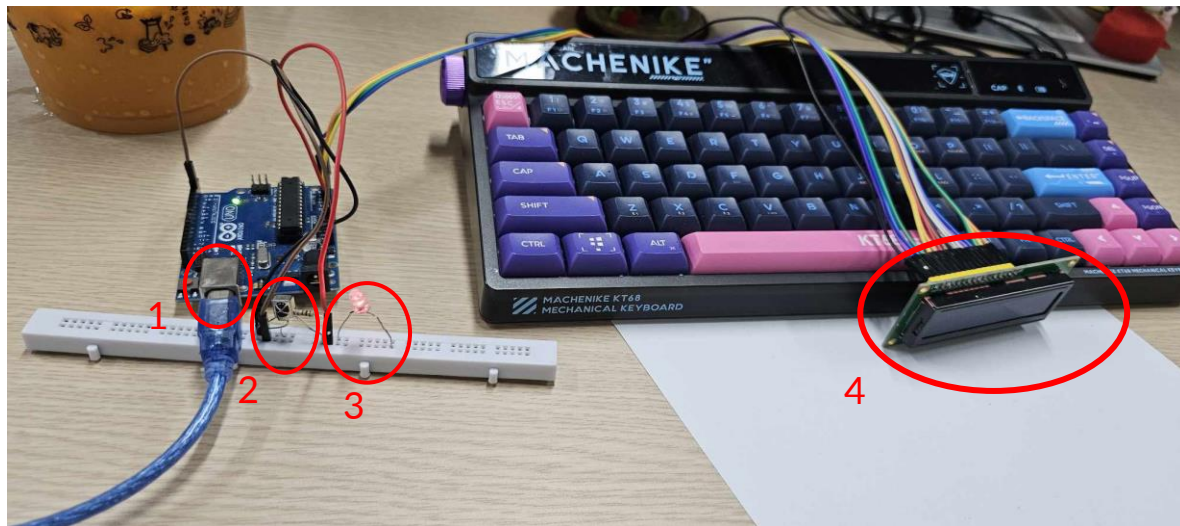
Triển Khai & Kết Quả

Sơ đồ lắp đặt



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

Lắp đặt thực tế trước khi hàn mạch



- 1: Cổng USB để nạp code và cấp nguồn từ máy tính
- 2: cảm biến thu tín hiệu IR
- 3: Led (sáng khi có lỗi xảy ra)
- 4: màn hình LCD hiển thị

Video kiểm tra các chức năng trước khi hoàn thiện



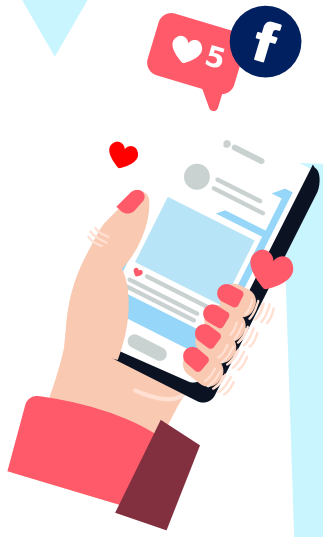
Sản phẩm hoàn thiện



Kết Luận

Nhóm đã thực hiện theo đúng các yêu cầu đề ra khi giải mã bản thu từ cảm biến hồng ngoại nhận tín hiệu từ bộ điều khiển từ xa của máy chiếu có sẵn trong giảng đường:

- ✓ Giải các mã bản tin ứng với ít nhất 4 lệnh : **11 Lệnh**
 - ✓ Sử dụng vi điều khiển 8 bit: **AT mega 328**
 - ✓ Lập trình bằng ngôn ngữ C hoặc tương đương, không sử dụng thư viện có sẵn: **IRLib, ReadWrite**
 - ✓ 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: **#define IR_RECEIVER_PIN 2**
- Nhóm đã hoàn thiện thành 1 sản phẩm hoàn chỉnh có thể thu, giải mã, hiển thị các chức năng từ điều khiển máy chiếu trên giảng đường.
- “Future work” : hoàn thiện thêm các chức năng, tối ưu hệ thống, ...



Thank you for watching

Group 6