BỘ TRUYỀN NHẬN HỒNG NGOẠI

GROUP 4

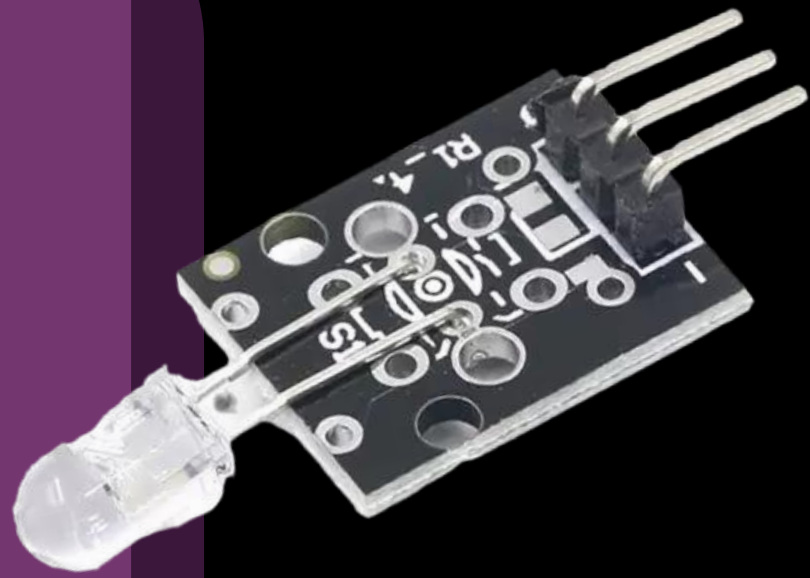
▶ MEMBER TASK

	THUYẾT TRÌNH	SOẠN NỘI DUNG	SLIDE	CODE
PHAN ĐÌNH QUÝ	x		x	x
TRẦN ĐÌNH BẢO	x	x		
TRƯƠNG PHƯỚC THỊNH	x	x		x

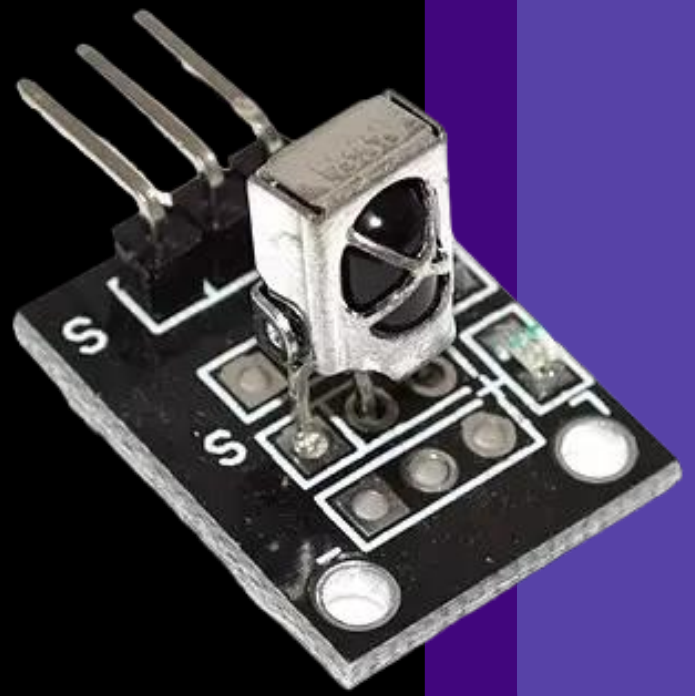
I. GIỚI THIỆU

1. BỘ TRUYỀN HỒNG NGOẠI

- Bộ truyền hồng ngoại gồm có một dãy LED hồng ngoại, có chức năng như một bộ phát tín hiệu hồng ngoại, và một mối nối PN được chế tạo đặc biệt với sự bức xạ hồng ngoại cao.
- Khi một dòng điện đi qua mối nối PN, nó có thể kích thích một nguồn ánh sáng hồng ngoại. Năng lượng của bức xạ ánh sáng hồng ngoại tỉ lệ với dòng điện. Tuy nhiên, trong trường hợp dòng điện bức xạ vượt quá giới hạn lớn nhất cho phép, năng lượng ánh sáng hồng ngoại có thể sụt giảm khi dòng điện tăng.

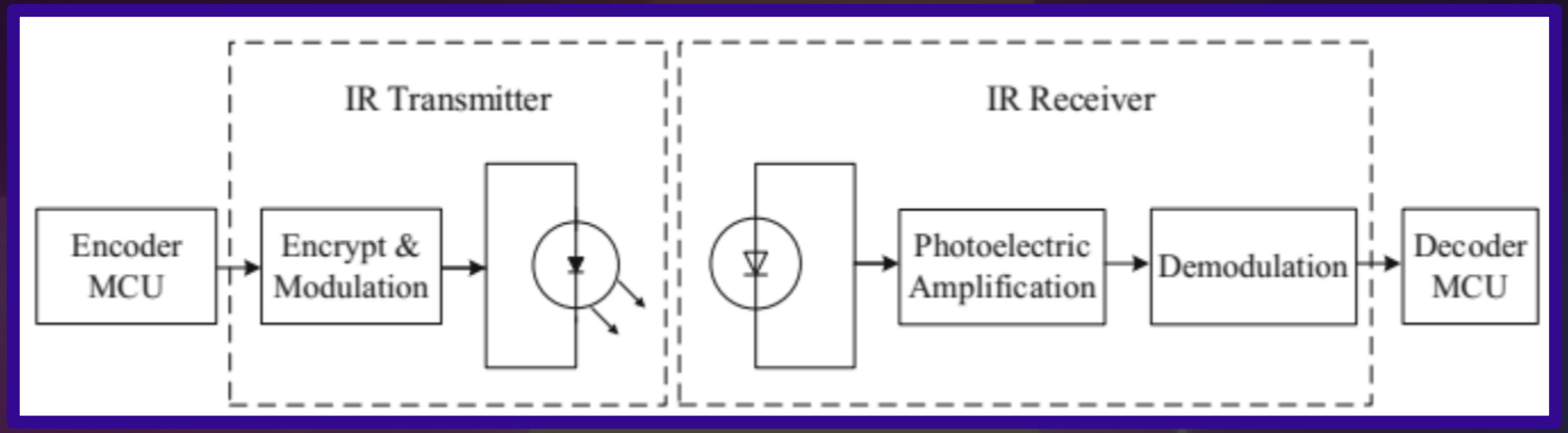


▶▶ 2. BỘ NHẬN HỒNG NGOẠI



- Bộ nhận hồng ngoại là một thiết bị bán dẫn sử dụng bộ chuyển đổi ánh sáng hồng ngoại sang tín hiệu điện.
- Phần linh kiện chính của thiết bị là mối nối PN được chế tạo từ một vật liệu đặc biệt. Mối nối PN có cấu trúc khác với một diode thông thường, cho phép nhận nhiều ánh sáng hồng ngoại.
- Khi cường độ của ánh sáng hồng ngoại tăng, dòng điện sẽ được sinh ra. Sau đó, dòng điện được xử lý để trích xuất thông tin được truyền.

II. SƠ ĐỒ NGUYÊN LÝ

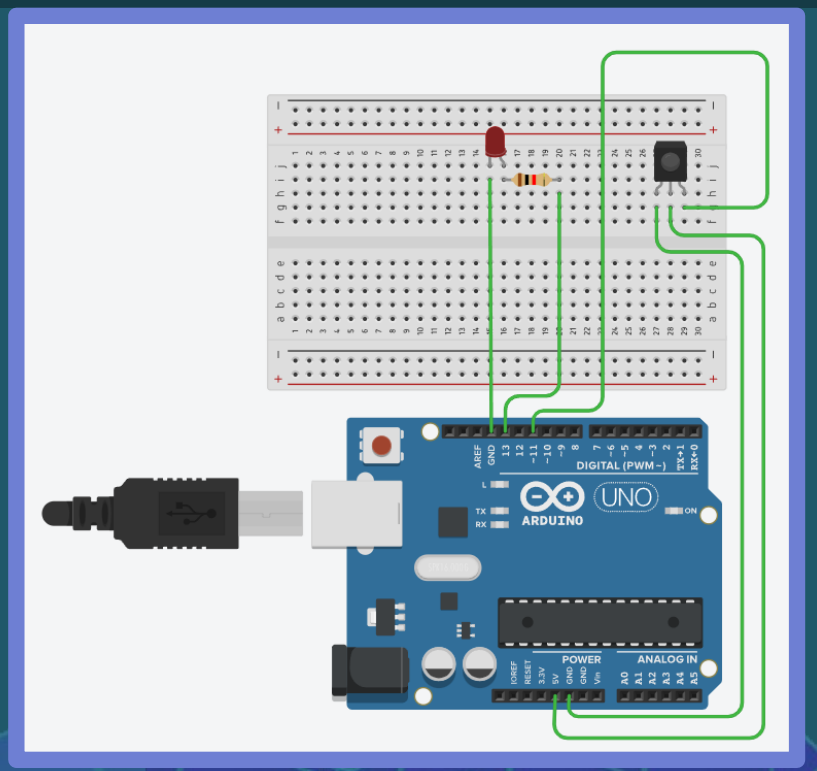


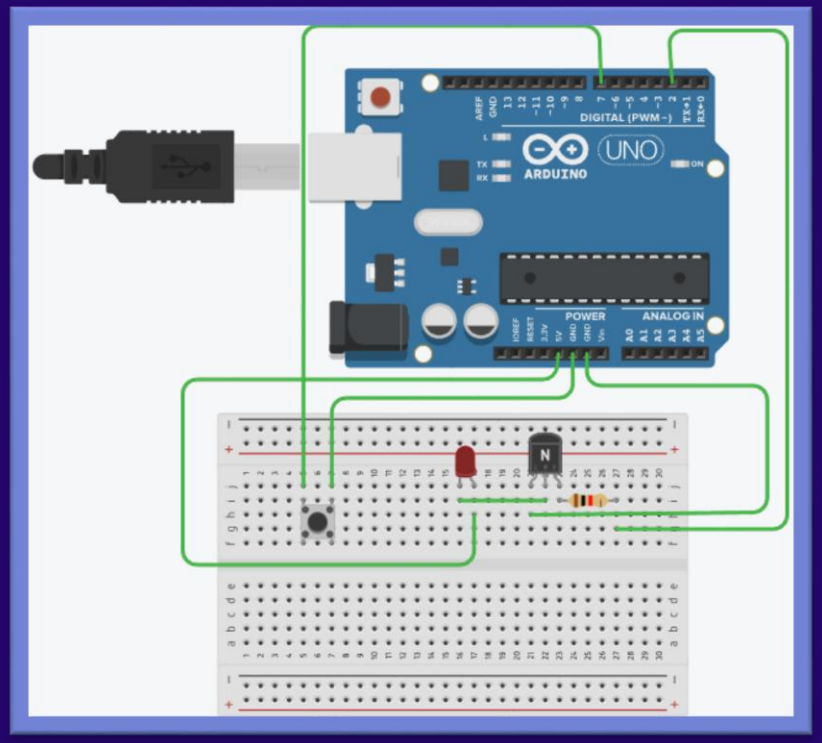
Trong module truyền nhận hồng ngoại (IR transmitter/receiver), dữ liệu được truyền được bắt đầu và kết thúc bằng các bit start và stop. Các bit start thông báo đến bộ nhận rằng dữ liệu sắp đến và bit stop chỉ rằng việc truyền đã kết thúc.

III. THIẾT LẬP PHẦN CỨNG

Bộ nhận hồng ngoại

(IR Reciver)





Bộ truyền hồng ngoại

(IR Transmitter)

IV. CODE

Truyền hồng ngoại
IR Transmitter

```
#include <IRremote.h>

// Define the pins
int irLedPin = 13;  // IR LED anode (+) connected to the collector
int bjtBasePin = 2; // BJT base pin
int buttonPin = 7;

IRsend irsend(2);

void setup() {
  // Set the pin modes
  pinMode(irLedPin, OUTPUT);
  pinMode(bjtBasePin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(buttonPin) == LOW){
  // Turn on the BJT to allow current to flow to the IR LED
  digitalWrite(bjtBasePin, HIGH);

  // Send Sony infrared signal with data code 0x1 and 12 bits
  irsend.sendSony(0x1, 12);
  delay(200);

  // Turn off the BJT to stop the current flow to the IR LED
  digitalWrite(bjtBasePin, LOW);

  delay(200);
  }
}
```

```ino
#include <IRremote.h>

const int RECV_Pin = 11;  // IR Sensor pin
const int LED_Pin = 13;   // LED pin

IRrecv irrecv(RECV_Pin);
decode_results results;

void setup() {
  Serial.begin(9600);   // configure the baud rate
  irrecv.enableIRIn();  // start the receiver
  pinMode(LED_Pin, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.print("Received value: 0x");
    Serial.println(results.value, HEX);
    if (results.bits > 0) {
      Serial.println("available");
      // Modify this part based on the codes sent by the transmitter
      if (results.value == 0x1) {  // If IR signal received is 0x1 (toggle LED )
        toggleLED();
      }
    }

    irrecv.resume();  // prepare to receive the next value
  }
}

void toggleLED() {
  int state = !digitalRead(LED_Pin);
  // Toggle LED state
  digitalWrite(LED_Pin, state);
  Serial.println("Toggling LED");
  delay(200);
}
```

Nhận hồng ngoại IR Receiver

BỘ NHẬN ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC

```ino
#include <IRremote.h>
#include <Stepper.h>

const int RECV_Pin = 11;  // IR Sensor pin
const int STEPPER_PIN1 = 8;  // Stepper motor control pin 1
const int STEPPER_PIN2 = 9;  // Stepper motor control pin 2
const int STEPPER_PIN3 = 10; // Stepper motor control pin 3
const int STEPPER_PIN4 = 12; // Stepper motor control pin 4

IRrecv irrecv(RECV_Pin);
decode_results results;

Stepper stepper(512, STEPPER_PIN1, STEPPER_PIN3, STEPPER_PIN2, STEPPER_PIN4);  // 512 steps per revolution

bool rotateClockwise = true;

void setup() {
  Serial.begin(9600);   // configure the baud rate
  irrecv.enableIRIn();  // start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.print("Received value: 0x");
    Serial.println(results.value, HEX);

    // Modify this part based on the codes sent by the transmitter
    if (results.value == 0x1) {  // If IR signal received is 0x1
      rotateClockwise = !rotateClockwise;  // Toggle the rotation direction
    }

    irrecv.resume();  // prepare to receive the next value
  }

  // Rotate the stepper motor continuously
  int rotationSteps = rotateClockwise ? 512 : -512;
  stepper.setSpeed(20);  // Adjust the speed as needed
  stepper.step(rotationSteps);  // Adjust the steps as needed
}
```