

# ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

------

BÁO CÁO BÀI TẬP LỚN

Đề tài: Robotix-Friend trên ESP32

# Giảng viên hướng dẫn: ThS. Nguyễn Đức Tiến

Sinh viên - MSSV: Phạm Thành Dũng 20204644

# Mai Trung Kiên 20200301

Lớp: CTTN - Khoa học máy tính

# Học phần: Hệ nhúng

Mã học phần: IT4210

# Mã lớp: 143800

Khóa: K65

# Hà Nội, 01/2024

## Giới thiệu

* 1. **Đề tài**

Bài tập lớn này triển khai làm Robot vẫy tay thông qua nhận tín hiệu hồng ngoại trên ESP32 sử dụng kiến trúc Adruino. Nhóm lựa chọn đề tài này do áp dụng được nhiều kiến thức đã học, bên cạnh đó tạo ra được một người bạn đáng yêu luôn sẵn sàng vẫy tay chào đón mỗi lúc ta cần đến. Các chức năng chính bao gồm:

* Điều khiển từ xa: Người dùng bấm điều khiển để gửi tín hiệu tới người bạn Robot
* Màn hình OLED: Biểu cảm được cảm xúc của Robot trên màn hình OLED 1.3 inches
* Động cơ Sevro: Thể hiện được động tác vẫy tay của người bạn Robot

## Phân công nhiệm vụ thành viên

|  |  |
| --- | --- |
| **Thành viên** | **Nhiệm vụ** |
| Phạm Thành Dũng (nhóm trưởng) | Load ảnh bitmap trên SSD1306; cấu hình các chân trên ESP32; thiết kế, lắp mạch; viết báo cáo; quay video demo. |
| Mai Trung Kiên | Thiết kế các trạng thái của Robot; lập trình code logic Robot |

1. **Cấu hình mạch**

## Danh sách thiết bị

## Mạch tích hợp ESP32

## Màn hình SSD1306 OLED 1.3 inches

## Cảm biến hồng ngoại

## Động cơ Sevro

## Điện trở nhỏ

## Sơ đồ mạch

Màn hình SSD1306 OLED nối với ESP32 bằng giao thức I2C (SCL trên SSD1306 nối với chân 22 của ESP32, SDA trên SSD1306 nối với chân 21 của ESP32).

Động cơ Sevro gồm có 3 chân, chân dương được nối với nguồn dương của ESP32, chân ground được cắm với đất và chân điều khiển được cắm vào chân PWM 25 của ESP32.

Thiết bị cảm biến hồng ngoại cũng có 3 chân, chân âm và dương cũng cắm tương tự như động cơ, chân tín hiệu được cắm với cổng 27 trên vi xử lý để vi xử lý đọc dữ liệu truyền tới.

## Robotix-Friend trên ESP32

## Load ảnh dưới dạng bitmap

* Bước 1: Thiết kế biểu cảm gương mặt cho Robot sử dụng Paint cho kích thước tương ứng với màn hình 128x64.
* Bước 2: Sử dụng trang web: [*https://mischianti.org/ssd1306-oled-display-draw-images-splash-and-animations-2/*](https://mischianti.org/ssd1306-oled-display-draw-images-splash-and-animations-2/)để convert ảnh sang mã 16bit.
* Bước 3: Khi có ma trận 16bit tương ứng với ảnh trên, khởi tạo các màn hình tương ứng trong chương trình sử dụng thư viện Adafruit để đọc bitmap tương ứng.

## Cấu trúc chương trình, thuật toán

## 1. Khởi tạo các chân, và 2 screen sử dụng bitmap

*const* uint8\_t PROGMEM sc1[] = {

  …

}

*const* uint8\_t PROGMEM sc2[] = {

}

*const* int irReceiverPin = 27;

*const* int ledPin = 2;

Servo myservo;

int pos = 0;

int servoPin = 25;

## 

## 2. Bật khối Timer để chạy động cơ Sevro sử dụng PWM

  ESP32PWM::allocateTimer(0);

  ESP32PWM::allocateTimer(1);

  ESP32PWM::allocateTimer(2);

  ESP32PWM::allocateTimer(3);

  myservo.setPeriodHertz(50); *// standard 50 hz servo*

  myservo.attach(servoPin, 1000, 2000);

## 3. Vòng lặp nếu nhận được tín hiệu hồng ngoại thì Robot sẽ vẫy tay và có animation cho biểu cảm

void loop() {

  if (IrReceiver.decode()) {

    uint16\_t command = IrReceiver.decodedIRData.command;

    Serial.println(command);

    if (command>100){

    delay(100);

    screen1();

    for (pos = 0; pos <= 180; pos += 1) { *// goes from 0 degrees to 180 degrees*

*// in steps of 1 degree*

      myservo.write(pos); *// tell servo to go to position in variable 'pos'*

      delay(8); *// waits 15ms for the servo to reach the position*

    }

    animation();

    for (pos = 180; pos >= 0; pos -= 1) { *// goes from 180 degrees to 0 degrees*

      myservo.write(pos); *// tell servo to go to position in variable 'pos'*

      delay(8); *// waits 15ms for the servo to reach the position*

    }

    screen1();

    for (pos = 0; pos <= 180; pos += 1) { *// goes from 0 degrees to 180 degrees*

*// in steps of 1 degree*

      myservo.write(pos); *// tell servo to go to position in variable 'pos'*

      delay(8); *// waits 15ms for the servo to reach the position*

    }

    animation();

    for (pos = 180; pos >= 0; pos -= 1) { *// goes from 180 degrees to 0 degrees*

      myservo.write(pos); *// tell servo to go to position in variable 'pos'*

      delay(8); *// waits 15ms for the servo to reach the position*

    }

    }

    IrReceiver.resume();

  }

}

## 4. Code chương trình và tài liệu tham khảo

## Link github

* Code github cùng hướng dẫn cài đặt: https://github.com/ThanhDungCR7/Robotix-Friend

## Tài liệu tham khảo

* Slide, tài liệu môn học Hệ nhúng - IT4210.