

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN



ĐỒ ÁN I

Đề tài:

**MẠCH PHẦN CỨNG VÀ  
TRUYỀN THÔNG RF CHO HỆ THỐNG  
THU THẬP DỮ LIỆU PIR**

Nhóm sinh viên: Nguyễn Thái Sơn 20212951  
Nguyễn Văn Mừng 20212894

Giảng viên hướng dẫn: PGS.TS Nguyễn Quốc Cường

*Chữ kí của GVHD*

Hà Nội, 6-2024

# LỜI NÓI ĐẦU

Trong thời đại số hóa ngày nay, sự phát triển của công nghệ đã mở ra không gian rộng lớn cho các ứng dụng nhúng và thiết bị thông minh, với mục đích tối ưu hóa các quy trình và tăng cường tiện ích trong cuộc sống hàng ngày. Trong ngữ cảnh này, cảm biến hồng ngoại chủ động (PIR - Passive Infrared Sensor) đã nổi lên như một công nghệ vô cùng quan trọng và có ứng dụng rộng rãi trong nhiều lĩnh vực Smart Home.

PIR không chỉ là một thiết bị cảm biến thông thường mà còn là một công cụ linh hoạt và hiệu quả trong việc phát hiện chuyển động dựa trên các dòng nhiệt hồng ngoại. Nhờ tính nhạy cao và khả năng hoạt động đáng tin cậy, PIR giúp giảm tải công việc của con người trong các quy trình điều khiển tự động, đồng thời mang lại sự tiện lợi và an toàn cho người dùng.

Trong báo cáo này, chúng tôi đặt ra mục tiêu xây dựng một hệ thống bao gồm mạch cứng và phần mềm để thu phát, lưu trữ và thao tác trên dữ liệu từ cảm biến hồng ngoại chủ động (PIR). Mạch cứng PIR Array được thiết kế với 5 cảm biến PIR, có khả năng thu thập tín hiệu hồng ngoại từ môi trường xung quanh và trả về giá trị ADC trong khoảng từ 0 đến 4095. Đồng thời, hệ thống phần mềm được phát triển để đọc và lưu trữ dữ liệu từ PIR Array, cung cấp công cụ để thu thập dữ liệu và thực hiện việc gán nhãn. Mục tiêu của đề tài là nghiên cứu và triển khai giải pháp toàn diện để xử lý và phân tích dữ liệu từ PIR Array, nhằm hỗ trợ các ứng dụng trong các lĩnh vực nhúng và các hệ thống thông minh. Báo cáo sẽ tập trung vào phân tích cấu trúc, hoạt động của hệ thống cảm biến PIR và các kỹ thuật xử lý dữ liệu để áp dụng vào các ứng dụng thực tế.

***Team PIR HARDWARE***

# Contents

<b>Chương 1. Giới thiệu chung</b>	<b>1</b>
1.1 Lý do chọn đề tài . . . . .	1
1.2 Mục đích của đề tài . . . . .	1
1.3 Mục tiêu cụ thể . . . . .	1
<b>Chương 2. Mô tả bài toán</b>	<b>3</b>
2.1 Vấn đề cần giải quyết . . . . .	3
2.1.1 Mạch cứng PIR Array . . . . .	3
2.1.2 Vi điều khiển . . . . .	3
2.1.3 Server và Database. . . . .	3
2.1.4 Phần mềm quản lý . . . . .	4
2.2 Lựa chọn phương pháp . . . . .	4
2.2.1 Vi điều khiển . . . . .	5
2.2.2 Server và Database . . . . .	6
2.2.3 Desktop Application: . . . . .	7
2.3 Công cụ triển khai . . . . .	7
2.3.1 Phần cứng và vi điều khiển . . . . .	7
2.3.2 Mô hình . . . . .	7
2.3.3 Phần mềm . . . . .	7
<b>Chương 3. Kết quả đạt được</b>	<b>9</b>
3.1 Phần cứng và vi điều khiển . . . . .	9
3.2 Phần mềm . . . . .	11
3.3 Thiết lập bối cảnh thu data . . . . .	13
<b>Chương 4. Vấn đề còn tồn đọng và khó khăn</b>	<b>15</b>
<b>Chương 5. Kết luận chung và hướng phát triển</b>	<b>16</b>

5.1	Tóm tắt nội dung nghiên cứu . . . . .	16
5.2	Ý nghĩa và ứng dụng thực tiễn . . . . .	16
5.3	Hướng phát triển trong tương lai . . . . .	16
5.4	Lời cảm ơn . . . . .	17

# Chương 1. Giới thiệu chung

Trong thời đại số hóa hiện nay, công nghệ nhúng và cảm biến thông minh đang ngày càng phát triển mạnh mẽ, đem lại nhiều lợi ích trong các ứng dụng công nghiệp và Smart Home. Cảm biến hồng ngoại chủ động (PIR) với tính nhạy cao và giá thành hợp lý được coi là một công nghệ quan trọng trong việc phát hiện chuyển động. Dự án này nhằm xây dựng một hệ thống bao gồm mạch cứng PIR Array và phần mềm để thu phát, lưu trữ và xử lý dữ liệu từ các cảm biến PIR. Mục tiêu là nghiên cứu và triển khai giải pháp nhằm hỗ trợ các ứng dụng thực tế, đặc biệt là trong lĩnh vực nhúng và hệ thống Smart Home.

## 1.1 Lý do chọn đề tài

Hiện nay, nhu cầu về các thiết bị nhúng cũng như cảm biến thông minh ngày càng lớn do khả năng ứng dụng rộng rãi của nó: trong các quá trình điều khiển công nghiệp, trên xe cộ cũng như các hệ thống Smart Home. Một cảm biến thông minh là cảm biến có thể đo đạc được các chỉ số của môi trường xung quanh để tự đưa ra những quyết định điều khiển đúng đắn, giúp giảm tải sự can thiệp của con người và đem đến các trải nghiệm tiện lợi.

PIR (Passive infrared sensor) là cảm biến hồng ngoại phát có thể phát hiện chuyển động của vật thể trong 1 phạm vi nhất định. Với giá thành rẻ và tính nhỏ gọn, việc áp dụng PIR để làm cảm biến thông minh định vị được vị trí vật thể sẽ có tính ứng dụng rất cao trong các hệ thống nhúng nói chung và thiết bị Smart Home nói riêng.

## 1.2 Mục đích của đề tài

Mục tiêu của đề tài này là đưa ra được một hệ thống mạch cứng và phần mềm để thu phát, lưu trữ và thao tác trên dữ liệu PIR bao gồm:

- Một mạch cứng PIR Array bao gồm 5 PIR, thu tín hiệu hồng ngoại của môi trường xung quanh và trả về giá trị ADC trong khoảng từ 0 đến 4095
- Hệ thống phần mềm để đọc và lưu trữ dữ liệu từ PIR Array, đồng thời cung cấp công cụ để lấy dữ liệu và gán nhãn.

## 1.3 Mục tiêu cụ thể

Các thành phần cụ thể của hệ thống cần đạt được bao gồm:

- Mạch cứng PIR Array.
- Vi điều khiển để đọc giá trị mạch cứng theo thời gian thực và gửi dữ liệu giá trị đó

đi.

- Server với Database để nhận và lưu trữ dữ liệu từ vi điều khiển, đồng thời cung cấp các API để lấy các dữ liệu
- Ứng dụng Desktop để: điều khiển bật tắt quá trình ghi dữ liệu, quay video tương ứng của môi trường ghi, đồng thời gán nhãn cho dữ liệu.

Yêu cầu về dữ liệu được ghi: Là dữ liệu của cả 5 con PIR trong PIR Array, được ghi theo thời gian thực với chu kỳ 10ms. Mỗi giây sẽ gửi server một gói dữ liệu bao gồm 500 giá trị của 5 PIR.

## Chương 2. Mô tả bài toán

Đề tài này tập trung phát triển hệ thống cảm biến hồng ngoại chủ động (PIR) gồm mạch cứng, vi điều khiển ESP32, server và phần mềm quản lý. Mạch cứng với 5 PIR thu thập tín hiệu và trả về giá trị ADC từ 0-4095. Vi điều khiển đọc và gửi dữ liệu qua Wi-Fi mỗi 10ms. Server sử dụng C++ và SQLite để lưu trữ và xử lý dữ liệu, trong khi ứng dụng desktop được phát triển bằng NodeJS để quản lý, trực quan hóa và gán nhãn dữ liệu. Hệ thống đảm bảo thu thập dữ liệu ổn định, xử lý nhanh chóng và dễ mở rộng.

### 2.1 Vấn đề cần giải quyết

Các vấn đề cần giải quyết ở từng thành phần bao gồm:

#### 2.1.1 Mạch cứng PIR Array

PIR Array: gồm 5 PIR cùng cách mạch khuếch đại, chạy liên tục trong thời gian dài, giá trị voltage trả về phải nằm trong khoảng từ 0 – 4095.

#### 2.1.2 Vi điều khiển

- Phải chạy song song được 2 tác vụ là đọc dữ liệu từ PIR Array và đóng gói, gửi dữ liệu tới Server.
- Dữ liệu ghi phải đều, đúng chu kỳ 10ms
- Vi điều khiển gửi dữ liệu qua đường truyền không dây, cụ thể với dự án này là qua Wifi.

#### 2.1.3 Server và Database.

- Nhỏ gọn về dung lượng.
- Xử lý dữ liệu nhanh với khả năng chịu tải đủ tốt phù hợp với chu kỳ gửi 1 giây / lần cùng lượng dữ liệu lớn được gửi đến.
- Đưa ra giao thức phù hợp, dễ sử dụng để gửi và đọc dữ liệu
- Chạy ổn định trong thời gian dài mà không xảy ra sập, ngắt server.
- Có tính bảo trì, mở rộng cao.

### 2.1.4 Phần mềm quản lý

- Điều khiển ghi dữ liệu cùng quay video theo thời gian thực.
- Trực quan hóa dữ liệu bằng đồ thị theo thời gian thực.
- Hỗ trợ gán nhãn, xuất ra file CSV.

## 2.2 Lựa chọn phương pháp

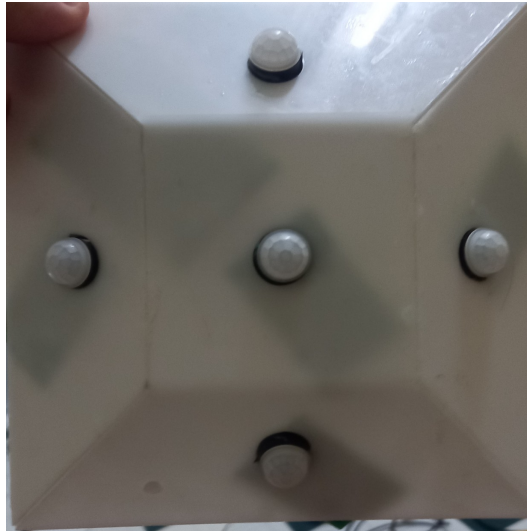


Figure 1: Mạch cứng PIR

Định nghĩa: Cảm biến PIR là một mạch cảm biến điện tử cảm ứng ánh sáng hồng ngoại phát ra từ các vật thể trong trường nhìn của nó. Về mặt kỹ thuật, PIR là một cảm biến nhiệt điện có thể phát hiện các mức bức xạ hồng ngoại khác nhau. PIR có thể phát hiện chuyển động của người, động vật trong phạm vi yêu cầu một cách thụ động, bởi yêu cầu của thông số kỹ thuật cụ thể

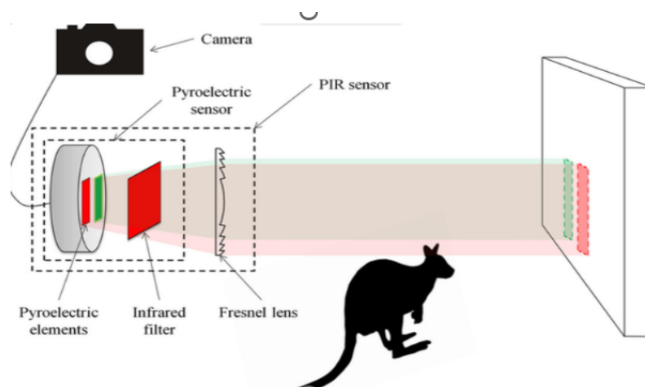


Figure 2: Cảm biến hồng ngoại PIR Sensor



Khi nguồn phát chuyển động vào vùng phát hiện (vùng có cả 2 kính hội tụ) thì tín hiệu sin nhảy mức cho điện áp ra. Sau đó, nguồn chuyển động khỏi vùng phát hiện, mạch trở về sẽ lưu giữ tín hiệu sau đó mới mất hẳn

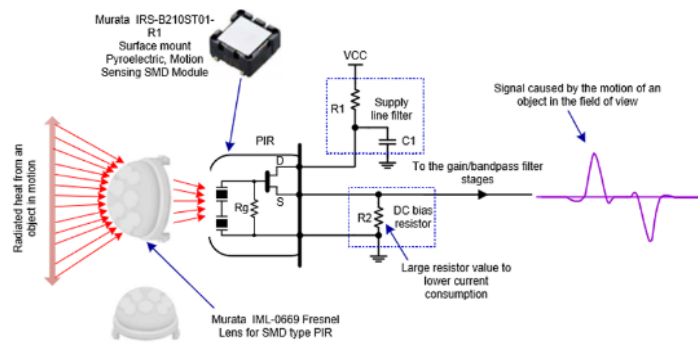


Figure 3: Nguyên lý hoạt động

### 2.2.1 Vi điều khiển

Sử dụng vi điều khiển ESP32 với ADC độ phân giải 12 bit, dải giá trị đầu ra 0-4095. ESP32 là một vi điều khiển mạnh mẽ và linh hoạt, thường được sử dụng trong các ứng dụng IoT và các dự án nhúng nhờ khả năng kết nối Wi-Fi và Bluetooth tích hợp sẵn. ADC (Analog to Digital Converter) trên ESP32 cho phép chuyển đổi tín hiệu analog sang tín hiệu số với độ phân giải 12 bit. Điều này có nghĩa là tín hiệu analog được chuyển đổi thành một giá trị số trong khoảng từ 0 đến 4095.

Ví dụ, nếu kết nối một cảm biến nhiệt độ analog vào ESP32, tín hiệu analog từ cảm biến có thể được chuyển đổi thành các giá trị số để vi điều khiển có thể xử lý.

ADC trên ESP32 có một số kênh, cho phép đọc tín hiệu từ nhiều nguồn khác nhau. Các kênh ADC này có thể được cấu hình và sử dụng một cách độc lập.

Một số điểm cần lưu ý khi sử dụng ADC trên ESP32:

- **Độ phân giải:** Với độ phân giải 12 bit, có thể nhận được 4096 mức giá trị khác nhau từ 0 đến 4095.
- **Điện áp tham chiếu:** Điện áp tham chiếu mặc định cho ADC trên ESP32 là từ 0V đến 3.3V. Điều này có nghĩa là giá trị 0 tương ứng với 0V và giá trị 4095 tương ứng với 3.3V.
- **Tốc độ lấy mẫu:** ESP32 có thể lấy mẫu tín hiệu analog ở tốc độ cao, phù hợp cho các ứng dụng yêu cầu cập nhật dữ liệu nhanh.

- **Hiệu chỉnh:** ADC trên ESP32 có thể cần được hiệu chỉnh để đảm bảo độ chính xác, do có thể có sai số do đặc điểm phần cứng và môi trường.

### 2.2.2 Server và Database

#### Về giao thức:

- Giao thức tầng transport sử dụng TCP/IP, tầng application sử dụng HTTP.
- Ngôn ngữ là C++, xử lý giao thức sử dụng C++ Socket

Phát triển một HTTP Module, tức là một thư viện HTTP Server, để xử lý giao thức HTTP.

#### Mô hình MVC:

- Model: Chứa các hàm ORM cho SQLite Database
- View: là các trang HTML đơn giản cung cấp thông tin về Server, đồng thời là các API cung cấp cho các bên khác đọc, ghi dữ liệu.
- Controller: Chứa các Middleware để xử lý các Request khác nhau

APIs do Server cung cấp cũng được triển khai theo mô hình RESTful với các thao tác đọc, ghi, sửa và xóa.

#### HTTP Module:

- Được phát triển từ C++ Socket để xử lý HTTP.
- Tận dụng khả năng về tốc độ và bộ nhớ của ngôn ngữ C++.
- Có thể đưa ra cú pháp trong sáng để tạo HTTP Server như các thư viện của các ngôn ngữ khác, xử lý được các Content-Type phổ biến.
- Có tính sử dụng lại cao khi cần xây dựng các dự án khác sử dụng C++ (điển hình như là làm microservices), có tính tùy biến cao.

#### Nền tảng cho Server:

- Chạy trên Linux (Debian based systems).
- Ngôn ngữ: C++.
- Sử dụng HTTP\_Module xử lý HTTP.
- Database: SQLite database.

### 2.2.3 Desktop Application:

Sử dụng NodeJS để phát triển ứng dụng Desktop.

Ưu điểm của NodeJS đối với dự án:

- Đa nền tảng, dễ cài đặt và sử dụng.
- Là môi trường phát triển phổ biến, tiện lợi, dễ tiếp cận.
- Sử dụng JavaScript, cung cấp nhiều API và các thư viện hữu ích giúp tiết kiệm thời gian phát triển và dễ dàng triển khai nhiều chức năng. Dễ bảo trì, mở rộng
- Nguồn tài liệu rộng rãi, có cộng đồng support lớn.

## 2.3 Công cụ triển khai

### 2.3.1 Phần cứng và vi điều khiển

- Layout mạch trên Altium
- Lập trình firmware trên ESPIDF

### 2.3.2 Mô hình

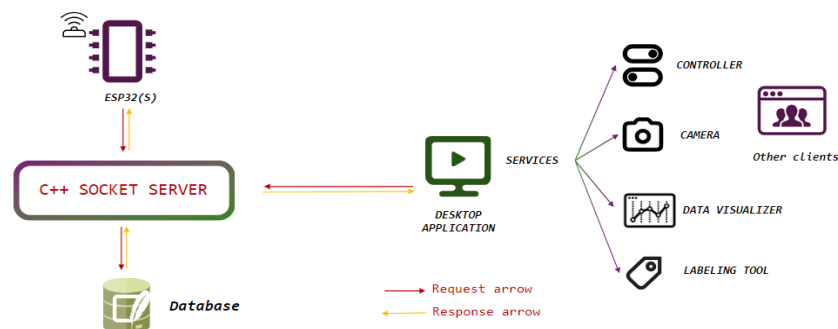


Figure 4: Mô hình cho hệ thống

### 2.3.3 Phần mềm

#### Phần mềm

- Text editor: Visual Code.

- Language: C++.
- Database: SQLite database.
- Build tool: CMake.
- Source Control Systems: Git & Github.

### **Desktop Application:**

- Text editor: Visual Code
- Environment: Nodejs.
- Language: TypeScript.
- Dependencies: ElectronJS, ReactJS, Tailwind CSS, Flowbite-React, ...

### Chương 3. Kết quả đạt được

Nguyên lý hoạt động của cảm biến vật cản hồng ngoại thụ động (PIR) dựa trên việc sử dụng một cặp cảm biến nhiệt điện để phát hiện năng lượng nhiệt từ sóng hồng ngoại trong môi trường xung quanh. Hai cảm biến này được đặt cạnh nhau và khi có sự chênh lệch về tín hiệu giữa chúng (ví dụ: khi một người di chuyển vào phòng), cảm biến hồng ngoại sẽ kích hoạt. Thiết bị có thể kích hoạt báo động, thông báo cho cơ quan chức năng (qua loa báo, đèn báo...) hoặc tự động bật/tắt đèn, quạt.

Bức xạ hồng ngoại được tập trung vào mỗi cảm biến nhiệt điện thông qua một loạt thấu kính Fresnel, được thiết kế như vỏ của cảm biến. Những thấu kính này mở rộng vùng cảm nhận của cảm biến, cho phép nó bao quát hầu hết khu vực xung quanh mắt cảm biến hồng ngoại. Thấu kính Fresnel giúp tối ưu hóa khả năng phát hiện, đảm bảo rằng ngay cả những chuyển động nhỏ cũng được ghi nhận, nâng cao hiệu quả và độ chính xác của cảm biến.

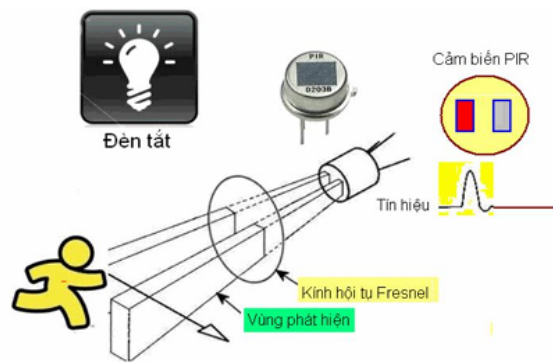


Figure 5: Mô hình cho hệ thống

#### 3.1 Phần cứng và vi điều khiển

##### \*Mạch cứng

Sơ đồ nguyên lý của mạch PIR bao gồm các thành phần chính như cảm biến PIR, bộ khuếch đại để tăng cường tín hiệu từ cảm biến, bộ lọc để loại bỏ các nhiễu không mong muốn và đảm bảo rằng chỉ các tín hiệu có ý nghĩa mới được xử lý, cùng với các linh kiện bổ sung khác như điện trở, tụ điện và vi điều khiển để đảm bảo toàn bộ hệ thống hoạt động ổn định và chính xác.

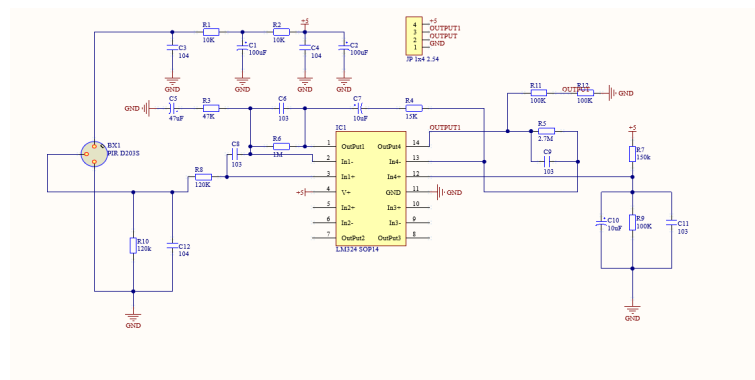
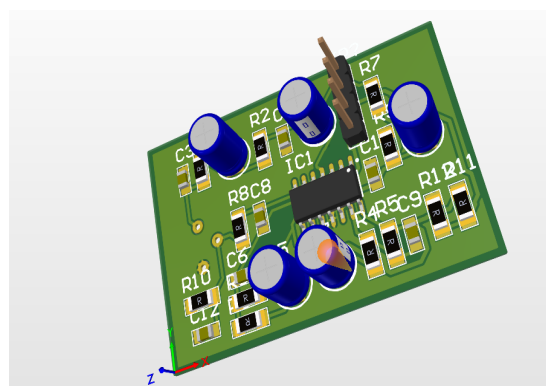
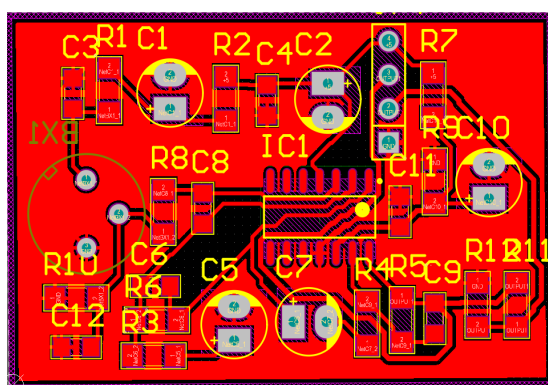


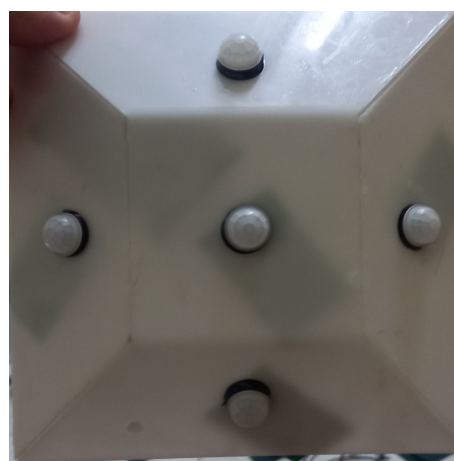
Figure 6: Mạch nguyên lý



(a) PCB

(b) 3D

Figure 7: Mạch in 3D



(a) Mạch thật

(b) PIR ARRAY

Figure 8: Sản phẩm

\*Thu dữ liệu

### **Sử dụng ESP32:**

ESP32 được sử dụng làm vi điều khiển chính cho hệ thống. ESP32 có tích hợp Wi-Fi và Bluetooth, giúp dễ dàng thu thập và truyền dữ liệu từ cảm biến PIR đến các thiết bị khác hoặc server.

### **Môi trường phát triển ESP IDF:**

ESP-IDF (Espressif IoT Development Framework) là môi trường phát triển chính thức của Espressif dành cho ESP32. Nó cung cấp các công cụ và thư viện cần thiết để phát triển ứng dụng trên ESP32.

### **Sử dụng FreeRTOS:**

FreeRTOS là một hệ điều hành thời gian thực được sử dụng trên ESP32 để quản lý các tác vụ và tài nguyên của hệ thống. Nó giúp đảm bảo rằng các tác vụ xử lý dữ liệu từ cảm biến PIR và truyền thông với server được thực hiện một cách hiệu quả và đáng tin cậy.

## **3.2 Phần mềm**

### **-Phát triển HTTP\_Module xử lý HTTP:**

HTTP\_Module được phát triển để xử lý các yêu cầu HTTP, bao gồm gửi và nhận dữ liệu từ server. Module này đảm bảo rằng dữ liệu từ cảm biến PIR có thể được truyền tới server một cách an toàn và hiệu quả.

### **-Hoàn thành xong Server với các API cơ bản:**

Server được triển khai với các API cơ bản để nhận, xử lý và lưu trữ dữ liệu từ các cảm biến PIR. Các API này cung cấp các chức năng cần thiết để giao tiếp với ESP32 và quản lý dữ liệu.

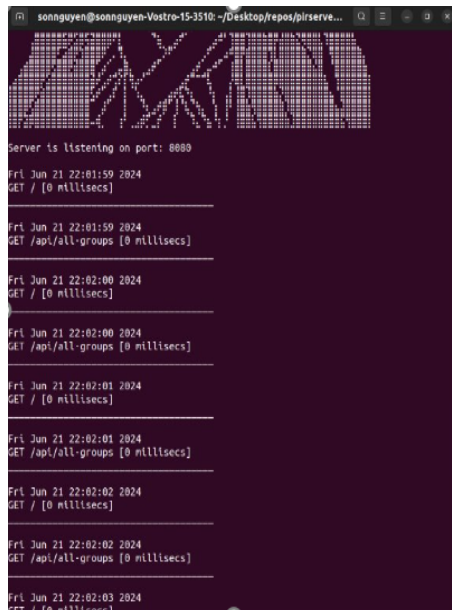


Figure 9: Server đang thu dữ liệu từ ESP32 gửi lên

## -Hoàn thành cơ chế quay, trực quan hóa dữ liệu và gán nhãn cho Desktop App:

Desktop App được phát triển để trực quan hóa dữ liệu thu thập từ cảm biến PIR. App này cung cấp các chức năng như quay lại dữ liệu, hiển thị trực quan thông tin và gán nhãn dữ liệu để phân tích và theo dõi.

Nhóm đã hoàn thành các bước từ phát triển phần cứng, vi điều khiển, thu thập dữ liệu, cho đến phần mềm xử lý và trực quan hóa, tạo nên một hệ thống hoàn chỉnh và hiệu quả trong việc phát hiện và xử lý dữ liệu từ cảm biến PIR.

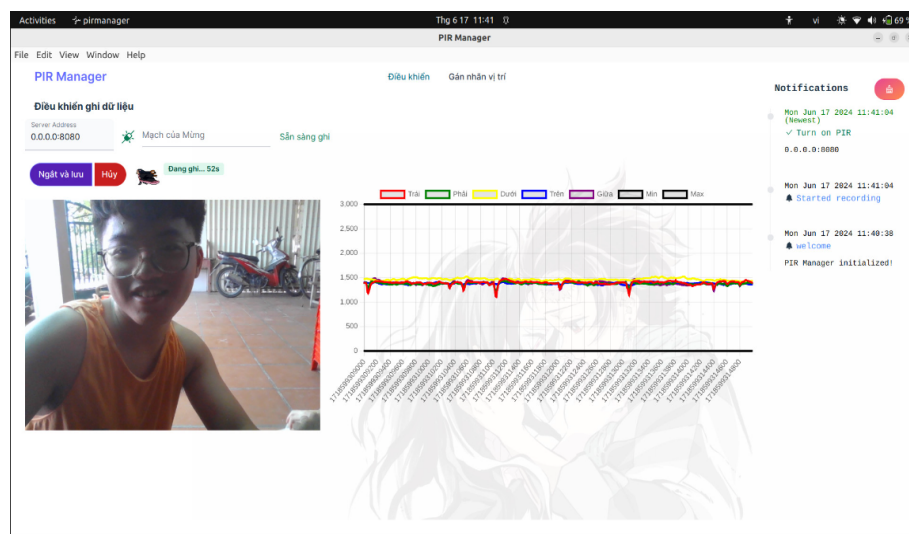


Figure 10: Giao diện app



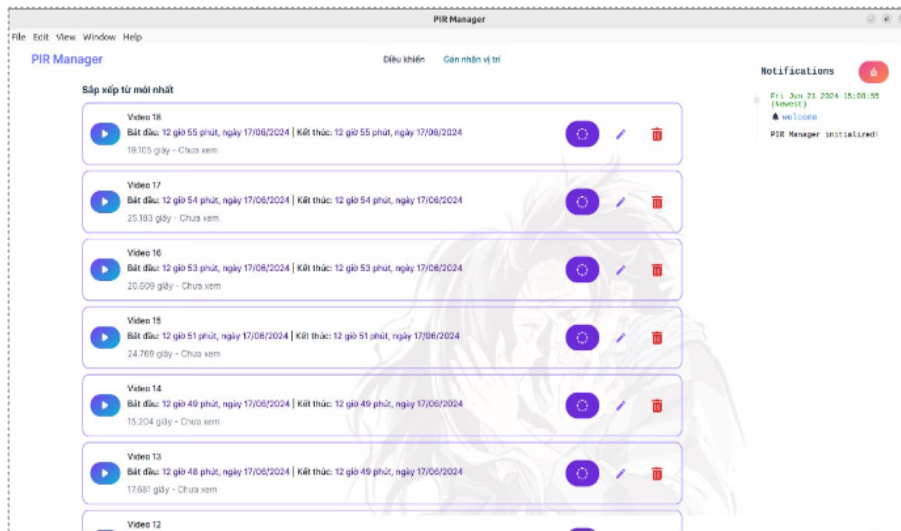


Figure 11: Chức năng gán nhãn phục vụ cho bên AI

### 3.3 Thiết lập bối cảnh thu data

#### Thử nghiệm:

- Nguồn cấp cho vi điều khiển: chân USB
- Vị trí: ngoài trời
- Nhiệt độ: 30 độ C
- Diện tích thử nghiệm: 3 x 3 m<sup>2</sup>

1	1	2	2	3	3
1	2	1	2	1	2
1	1	2	2	3	3
3	4	3	4	3	4
4	4	5	5	6	6
1	2	1	2	1	2
4	4	5	5	6	6
3	4	3	4	3	4
7	7	8	8	9	9
1	2	1	2	1	2
7	7	8	8	9	9
3	4	3	4	3	4

(a) Chia thành các vùng nhỏ



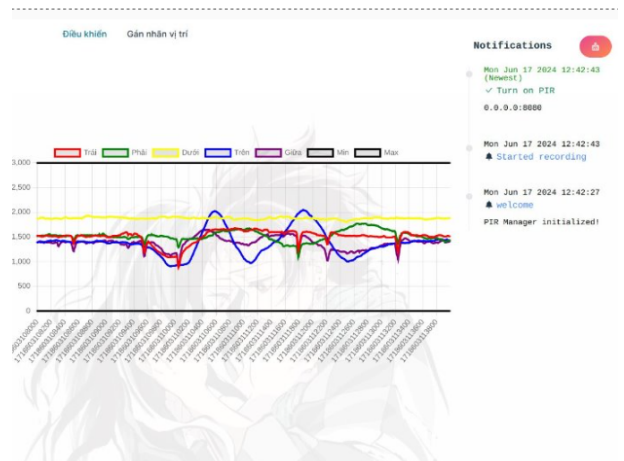
(b) HIện trường

Figure 12: Bối cảnh thu data

#### Kết quả:



(a) Dữ liệu khi không có chuyển động



(b) Đi từ ô 11 đến 32 (Hàng đầu tiên)

Figure 13: Đồ thị thể hiện giá trị ADC các cảm biến

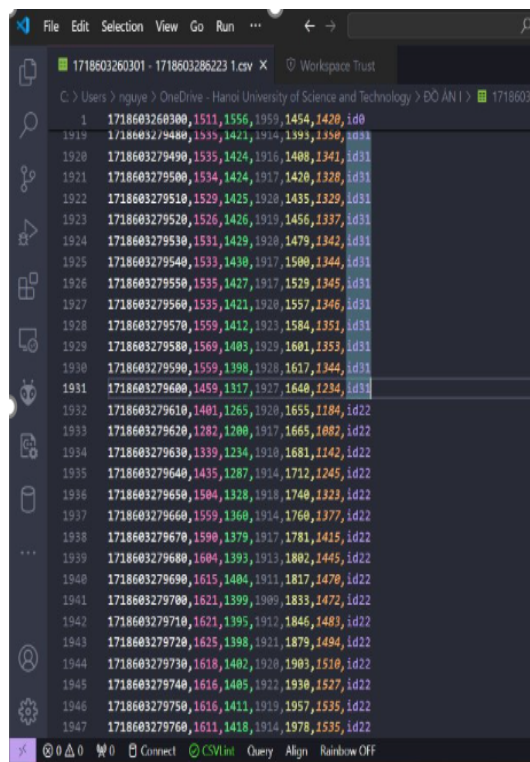
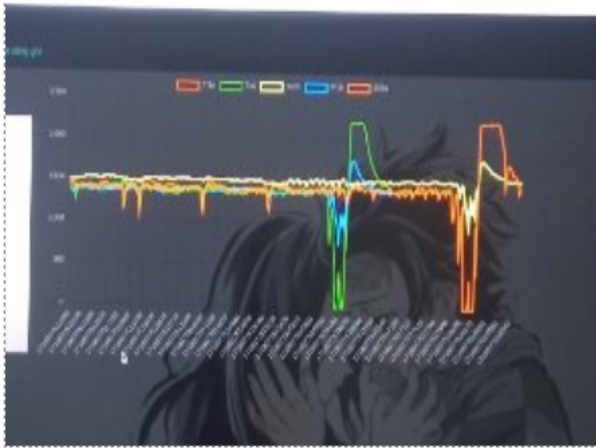


Figure 14: Một mẫu csv đầu ra, đã được gán nhãn.

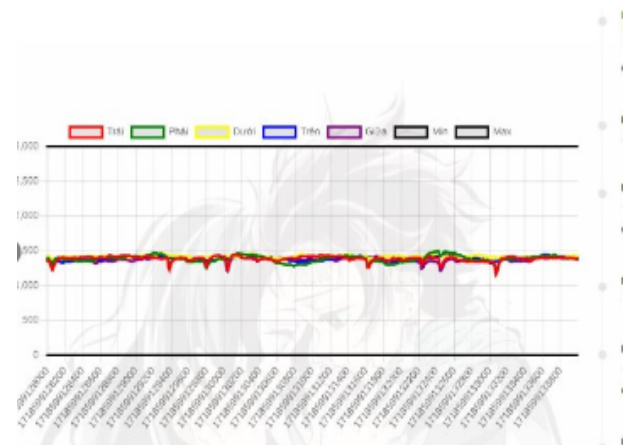
## Chương 4. Vấn đề còn tồn đọng và khó khăn

### Nhiều trong quá trình đọc ADC:

Trong quá trình thực hành thử nghiệm phát hiện việc kết nối Wifi và mở cổng Socket có ảnh hưởng đến số ADC mà ESP32 đọc được. Cụ thể, tốc độ kết nối Wifi càng yếu, thời gian mở cổng Socket càng lâu thì điện áp đọc được càng bị kéo xuống thấp, thậm chí là về 0 trong 1 khoảng thời gian.



(a) Dữ liệu kết nối kém



(b) Dữ liệu kết nối tốt hơn

Figure 15: Nhiều trong quá trình đọc ADC

### Dự đoán về nguyên nhân:

Dù chưa kịp có những thử nghiệm chi tiết, nhưng về mặt dự đoán, nhiều lớn này gây ra có thể bởi một trong các lý do sau:

- Việc sử dụng Wifi làm ảnh hưởng đến đầu đọc ADC hoặc phần hàm để lấy dữ liệu ADC.
- Việc sử dụng Wifi đã lấy mất phần năng lượng để cung cấp điện áp cho chân nguồn 3.3V, dẫn tới sụt nguồn cho toàn bộ 5 con PIR.

Trong đó, nguyên nhân thứ hai là nguyên nhân khả quan hơn.

## **Chương 5. Kết luận chung và hướng phát triển**

### **5.1 Tóm tắt nội dung nghiên cứu**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, việc ứng dụng các công nghệ cảm biến thông minh vào đời sống và sản xuất trở nên ngày càng cần thiết. Dự án "Mạch phần cứng và truyền thông RF cho hệ thống thu thập dữ liệu PIR" đã thành công trong việc thiết kế và triển khai một hệ thống hoàn chỉnh, bao gồm phần cứng, vi điều khiển, server và phần mềm quản lý.

Chúng tôi đã tiến hành thiết kế một mạch cứng với 5 cảm biến PIR, cho phép thu thập tín hiệu hồng ngoại từ môi trường xung quanh và chuyển đổi chúng thành giá trị số. Hệ thống vi điều khiển ESP32 được lập trình để đọc và truyền tải dữ liệu qua kết nối Wi-Fi đến một server được xây dựng bằng ngôn ngữ C++, lưu trữ trong cơ sở dữ liệu SQLite. Phần mềm quản lý trên desktop, phát triển bằng NodeJS, cung cấp giao diện người dùng để điều khiển, trực quan hóa và gán nhãn dữ liệu.

### **5.2 Ý nghĩa và ứng dụng thực tiễn**

Hệ thống thu thập dữ liệu PIR này không chỉ là một giải pháp kỹ thuật hoàn chỉnh mà còn mở ra nhiều cơ hội ứng dụng thực tiễn trong đời sống và công nghiệp:

- Ứng dụng trong nhà thông minh (Smart Home): Hệ thống có thể được tích hợp vào các thiết bị và hệ thống nhà thông minh để phát hiện chuyển động, điều khiển ánh sáng, an ninh, và nhiều ứng dụng khác.
- Giám sát an ninh: Với khả năng phát hiện chuyển động chính xác, hệ thống có thể được sử dụng trong các hệ thống giám sát an ninh để phát hiện xâm nhập trái phép.

### **5.3 Hướng phát triển trong tương lai**

Dù đã đạt được nhiều kết quả đáng khích lệ, dự án vẫn còn nhiều tiềm năng phát triển:

- Nâng cao độ chính xác và khả năng nhận diện: Tích hợp các thuật toán học máy để phân tích dữ liệu từ cảm biến, giúp nâng cao độ chính xác và khả năng nhận diện các đối tượng và hành vi phức tạp hơn.
- Mở rộng số lượng cảm biến: Tăng số lượng cảm biến trong mạch cứng để mở rộng phạm vi giám sát và thu thập dữ liệu.
- Tối ưu hóa hiệu suất: Cải thiện hiệu suất của vi điều khiển và server để xử lý dữ liệu nhanh hơn và tiết kiệm năng lượng hơn.

#### **5.4 Lời cảm ơn**

Chúng em xin chân thành cảm ơn PGS.TS Nguyễn Quốc Cường đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài này. Chúng em cũng xin cảm ơn các thầy cô và bạn bè đã luôn động viên và giúp đỡ chúng em hoàn thành dự án này. Dự án không chỉ là kết quả của những nỗ lực cá nhân mà còn là sự hợp tác và hỗ trợ từ nhiều phía.

Dự án "Mạch phần cứng và truyền thông RF cho hệ thống thu thập dữ liệu PIR" đã mang lại cho chúng em nhiều kiến thức và kinh nghiệm quý báu, giúp chúng em hiểu sâu hơn về công nghệ cảm biến và hệ thống nhúng, mở ra nhiều cơ hội cho các nghiên cứu và ứng dụng trong tương lai.

## References

- [1] DB203B Datasheet
- [2] K. Ngamakeura, S. Yongchareon, J. Yu, and S. Islam, “Passive infrared sensor dataset and deep learning models for device-free indoor localization and tracking,” *Journal/Conference Name*, 2021.
- [3] L. Wu and Y. Wang, “Stationary and moving occupancy detection using the SLEEP-IR sensor module and machine learning,” *Journal/Conference Name*, 2021.