****TRƯỜNG ĐẠI HỌC HẠ LONG**

**KHOA CÔNG NGHỆ THÔNG TIN**

-------------o0o------------



**BÁO CÁO BÀI TẬP LỚN**

**PHÁT TRIỂN ỨNG DỤNG IOT**

**TÊN ĐỀ TÀI: PHÒNG HỌP THÔNG MINH**

Lớp: Khoa học máy tính k6b

Học phần: Phát triển ứng dụng iot

Sinh viên thực hiện: Trần Xuân Thành

Nguyễn Văn Thạch

Phạm Minh Thiên

Nguyễn Mạnh Hùng

Giảng viên hướng dẫn: TS. Nguyễn Tài Tuyên

*Quảng Ninh, tháng 5 năm 2023*

*Thái Nguyên, ngày 24 tháng 1 năm 2014*

**

**MỤC LỤC**

[**LỜI CẢM ƠN** 4](#_Toc149467326)

[A. GIỚI THIỆU 5](#_Toc149467327)

[1. Lý do chọn đề tài này 5](#_Toc149467328)

[B. CƠ SỞ LÝ THUYẾT 6](#_Toc149467329)

[1. Thực trạng của phòng họp 6](#_Toc149467330)

[2. Biện pháp khắc phục 6](#_Toc149467331)

[3. Công cụ lập trình 7](#_Toc149467332)

[C. THIẾT KẾ HỆ THỐNG 8](#_Toc149467333)

[1. Đầu vào (cảm biến) 8](#_Toc149467334)

[2. Đầu ra (cơ cấu chấp hành) 8](#_Toc149467335)

[3. Khối xử lý điều khiển 9](#_Toc149467336)

[D. LINH KIỆN VẬT TƯ 13](#_Toc149467337)

[E. GHÉP NỐI THỬ NGHIỆM 14](#_Toc149467338)

[1. Sơ đồ ghép nối 14](#_Toc149467339)

[2. Xây dựng chương trình 14](#_Toc149467340)

[F. KẾT LUẬN 24](#_Toc149467341)

[G. TÀI LIỆU THAM KHẢO 25](#_Toc149467342)

# **LỜI CẢM ƠN**

Kính gửi TS. Nguyễn Tài Tuyên

Chúng em xin gửi lời cảm ơn sâu sắc và lòng biết ơn chân thành đến thầy về môn học "Phát triển ứng dụng IoT" trong thời gian học vừa qua. Được học môn này với đề tài "Phòng họp thông minh" đã mang lại cho chúng em một trải nghiệm học tập thú vị và hữu ích.

Thầy là người hướng dẫn tận tâm, kiên nhẫn và có kiến thức sâu rộng về lĩnh vực IoT. Dưới sự hướng dẫn của thầy, chúng em đã có cơ hội thực hành và áp dụng kiến thức lý thuyết vào thực tế, từ việc lập trình các thiết bị IoT đến việc mô phỏng phòng họp thông minh với khả năng đếm số người và điều chỉnh nhiệt độ một cách tự động.

Môn học này đã giúp chúng em hiểu rõ hơn về cách IoT có thể ứng dụng trong nhiều lĩnh vực, đặc biệt là trong quản lý không gian và tối ưu hóa sử dụng năng lượng. Chúng em đã học được cách làm việc nhóm hiệu quả, phát triển kỹ năng lập trình, và tạo ra sản phẩm thực tế có khả năng cải thiện cuộc sống hàng ngày.

Chúng em cảm ơn thầy rất nhiều về sự dẫn dắt và khả năng giảng dạy xuất sắc của thầy trong suốt thời gian học vừa qua. Môn học này đã mang lại cho chúng em kiến thức và kỹ năng quý báu, và chúng em sẽ luôn nhớ đến những bài học quý giá mà thầy đã truyền đạt.

Chúng em chân thành cảm ơn và kính chúc thầy luôn mạnh khỏe, hạnh phúc và thành công trong công việc giảng dạy cũng như trong cuộc sống cá nhân của mình.

THAY MẶT CÁC THÀNH VIÊN NHÓM

NHÓM TRƯỞNG

Thành

**Trần Xuân thành**

1. GIỚI THIỆU

1. Lý do chọn đề tài này

Chúng em đã chọn đề tài "Phòng họp thông minh với chức năng đếm số người và điều chỉnh nhiệt độ phù hợp" với mục tiêu nghiên cứu và phát triển ứng dụng IoT dựa trên các lý do quan trọng sau đây:

Thời đại số và vai trò của IoT: Chúng em đang sống trong thời đại số, và IoT (Internet of Things) đã trở thành một công nghệ quan trọng trong việc kết nối và tự động hóa các hệ thống. IoT đã thay đổi cách chúng em tương tác với môi trường xung quanh, từ nhà ở, công việc, đến các công cộng và thương mại. Việc sử dụng IoT để cải thiện cuộc sống và nâng cao hiệu quả làm việc đã trở thành một xu hướng không thể bỏ qua.

Quản lý không gian làm việc: Trong môi trường công nghiệp và thương mại, việc quản lý không gian làm việc đòi hỏi sự hiệu quả và tiết kiệm. Phòng họp, như một phần quan trọng của môi trường làm việc, thường gặp phải các thách thức về việc quản lý sự hiện diện của người dùng, cũng như điều kiện môi trường như nhiệt độ và ánh sáng. Sử dụng công nghệ IoT để tự động hóa quá trình này có thể giúp tiết kiệm thời gian và tài nguyên và đảm bảo sự thoải mái cho người dùng.

Đảm bảo môi trường thoải mái và tiết kiệm năng lượng: Phòng họp thông minh có khả năng đếm số người ra vào và điều chỉnh nhiệt độ, điều hòa, ánh sáng một cách tự động. Điều này không chỉ tạo ra một môi trường thoải mái cho người dùng mà còn giúp tiết kiệm năng lượng. Điều này rất quan trọng trong bối cảnh tăng cường tầm nhìn về bảo vệ môi trường và tối ưu hóa sử dụng năng lượng.

Từ những lý do trên, chúng em tin rằng đề tài này có tiềm năng mang lại giá trị thực sự trong việc ứng dụng công nghệ IoT để tạo ra các phòng họp thông minh, mang lại lợi ích về hiệu quả quản lý không gian làm việc và đảm bảo môi trường làm việc tối ưu cho người dùng.

1. CƠ SỞ LÝ THUYẾT

1. Thực trạng của phòng họp

Phòng họp, không chỉ là nơi các cuộc họp quan trọng diễn ra, mà còn là một phần quan trọng trong việc tạo ra môi trường làm việc hiệu quả. Hiện nay, phòng họp thường trang bị nhiều thiết bị và cơ sở hạ tầng như điều hòa, chiếu sáng, quạt, và hệ thống điện. Tuy nhiên, cách quản lý và điều khiển các thiết bị này vẫn còn tồn đọng một số vấn đề:

Số lượng người ra vào phòng họp: Việc quản lý số lượng người ra vào phòng họp có thể gặp khó khăn, đặc biệt là trong các tình huống khẩn cấp hoặc đối với các phòng có sức chứa hạn chế.

An toàn cháy nổ và an toàn điện: Đảm bảo an toàn trong phòng họp, đặc biệt là về mặt cháy nổ và điện, là một ưu tiên quan trọng.

Quản lý vận hành: Hiện nay, việc quản lý và vận hành các thiết bị trong phòng họp thường dựa vào con người và có thể không hiệu quả.

1. Biện pháp khắc phục

Ứng dụng công nghệ 4.0 (IoT) trong quản lý vận hành:

Để giải quyết các vấn đề trên, chúng em đề xuất ứng dụng công nghệ IoT (Internet of Things). Công nghệ này có thể tạo ra một hệ thống thông minh để quản lý phòng họp và cải thiện hiệu suất vận hành:

Đếm số lượng người: Sử dụng cảm biến chuyển động hoặc các công nghệ nhận dạng để đếm số lượng người ra vào phòng họp.

Điều khiển nhiệt độ và điều hòa: Các thiết bị điều khiển nhiệt độ và điều hòa có thể được kết nối và điều khiển từ xa thông qua mạng IoT để đảm bảo rằng phòng luôn duy trì nhiệt độ thoải mái.

Điều khiển ánh sáng: Các hệ thống chiếu sáng có thể tự động điều chỉnh dựa trên mức độ ánh sáng tự nhiên và số lượng người trong phòng.

Gửi dữ liệu lên cloud: Dữ liệu về số lượng người và điều kiện phòng họp có thể được gửi lên cloud để theo dõi và phân tích từ xa.

Phân tích dữ liệu: Dữ liệu thu thập được từ phòng họp có thể được phân tích để hiểu hơn về cách sử dụng và cải thiện hiệu suất.

Gửi thông báo về thiết bị cầm emy: Thông qua ứng dụng di động hoặc thiết bị cầm emy, người dùng có thể nhận thông báo về tình trạng của phòng họp và điều khiển các thiết bị.

1. Công cụ lập trình

Trong dự án này, chúng em sử dụng Arduino IDE để lập trình và điều khiển các thiết bị IoT. Arduino IDE cung cấp một môi trường lập trình dễ sử dụng cho việc phát triển các ứng dụng IoT.

1. THIẾT KẾ HỆ THỐNG

1. Đầu vào (cảm biến)

- Cảm biến hồng ngoại:

+ Giới thiệu: Cảm biến hồng ngoại là một loại cảm biến dùng để phát hiện và đo lượng sóng hồng ngoại phát ra từ các đối tượng.

+ Sơ đồ chân:

VCC (Nguồn): Chân này được kết nối đến nguồn cung cấp điện tích (VD: 5V).

GND (Đất): Chân này được kết nối đến đất (GND) của nguồn cung cấp.

OUT (Ra/Out): Chân này là đầu ra của cảm biến, nó sẽ cung cấp một tín hiệu hoặc điện áp thay đổi tùy thuộc vào điều kiện sự phát hiện của cảm biến.

+ Chức năng:

Cảm biến hồng ngoại hoạt động bằng cách phát ra sóng hồng ngoại và sau đó đo lượng sóng hồng ngoại phản xạ hoặc phản chiếu lại từ các đối tượng xung quanh.

Khi có sự thay đổi trong môi trường (như sự hiện diện hoặc chuyển động của đối tượng), cảm biến hồng ngoại sẽ phát hiện sự thay đổi này và tạo ra một tín hiệu đầu ra tương ứng.

Tín hiệu đầu ra có thể là một tín hiệu kỹ thuật số (ví dụ: 0V hoặc 5V) hoặc tín hiệu tương tự (điện áp biến đổi từ giá trị thấp đến giá trị cao hoặc ngược lại) tùy thuộc vào loại cảm biến cụ thể.

Cảm biến hồng ngoại thường được sử dụng để phát hiện sự hiện diện của đối tượng hoặc để đo khoảng cách đến đối tượng.

2. Đầu ra (cơ cấu chấp hành)

**Relay tắt và mở đèn chiếu sáng**

Đèn tuýp LED 1,2m/220V là một loại đèn chiếu sáng tiết kiệm năng lượng và có tuổi thọ cao, thường được sử dụng trong nhà. Chúng cung cấp ánh sáng tự nhiên và đồng đều, tiết kiệm điện và không chứa thủy ngân. Đèn tuýp LED 1,2m/220V dễ dàng lắp đặt và là một giải pháp chiếu sáng hiệu quả cho các không gian lớn trong nhà.- điều hòa:

**Module phát tín hiệu điều khiển điều hòa**

Điều hòa là thiết bị sử dụng để điều chỉnh nhiệt độ và độ ẩm của không khí trong một không gian nhằm tạo ra môi trường thoải mái. Các chức năng bao gồm làm lạnh, điều chỉnh nhiệt độ, độ ẩm, quạt và cài đặt thời gian. Điều hòa có thể được điều khiển từ xa thông qua remote hoặc ứng dụng di động, cung cấp sự tiện lợi và tiết kiệm năng lượng cho người dùng.

3. Khối xử lý điều khiển

**Modul esp32**

***Giới thiệu***

Esp32 là một dòng sản phẩm của Espressif Systems, là một vi xử lý tích hợp Wi-Fi và Bluetooth dựa trên vi điều khiển Xtensa 32-bit. Esp32 được phát triển để hỗ trợ một loạt các ứng dụng IoT (Internet of Things) và có khả năng kết nối với Internet thông qua Wi-Fi hoặc Bluetooth. Nó có một bộ xử lý mạnh mẽ, hỗ trợ nhiều giao tiếp và tích hợp nhiều tính năng hữu ích, làm cho nó trở thành một lựa chọn phổ biến cho các dự án phát triển IoT.

***Sơ đồ chân***

A computer chip with many different colored labels

Description automatically generated

Hình 1. Sơ đồ chân esp32

Chúng em đã tìm hiểu về sơ đồ chân ESP32. Bây giờ chúng ta hãy tập trung vào một số thiết bị ngoại vi quan trọng của ESP32 và các chân liên kết của chúng. Vi điều khiển ESP32 có:

34 GPIO có thể lập trình

18 kênh ADC 12 bit

2 kênh DAC 8-bit

16 kênh PWM

3 giao diện UART

3 giao diện SPI

2 Giao diện I2C

2 Giao diện I2S

10 GPIO cảm ứng điện dung

16 GPIO RTC

**GPIO**

Thiết bị ngoại vi được sử dụng phổ biến nhất là GPIO. ESP32 có 34 chân GPIO với mỗi chân thực hiện nhiều hơn một chức năng (chỉ một chân hoạt động). Bạn có thể cấu hình chân dưới dạng GPIO hoặc ADC hoặc UART trong chương trình.

Các chân ADC và DAC được xác định trước và bạn phải sử dụng các chân do nhà sản xuất chỉ định. Nhưng các chức năng khác như PWM, SPI, UART, I2C, ... có thể được gán cho bất kỳ chân GPIO nào thông qua chương trình.

**RTC GPIO**

ESP32 có 16 GPIO RTC, là một phần của hệ thống con RTC Low-Power. Các chân này có thể được sử dụng để đánh thức ESP32 khỏi chế độ ngủ sâu làm nguồn đánh thức bên ngoài.

**ADC**

ESP32 có hai module chuyển đổi analog sang kỹ thuật số SAR 12 bit với 8 kênh và 10 kênh mỗi module. Vì vậy, khối ADC1 và ADC2 kết hợp với nhau có 18 kênh ADC 12-bit.

Với độ phân giải 12-bit, các giá trị kỹ thuật số đầu ra sẽ nằm trong khoảng 0 - 4093.

**DAC**

Bộ vi điều khiển ESP32 có hai kênh chuyển đổi kỹ thuật số sang analog 8 bit độc lập để chuyển đổi các giá trị kỹ thuật số sang tín hiệu điện áp analog. DAC có mạng điện trở bên trong và sử dụng nguồn điện làm điện áp tham chiếu đầu vào.

Hai chân GPIO sau được liên kết với các chức năng của DAC

DAC1 - GPIO25

DAC2 - GPIO26

**GPIO cảm ứng điện dung**

SoC ESP32 có 10 GPIO cảm ứng điện dung, có thể phát hiện các biến thể về điện dung trên chân cắm do chạm hoặc tiếp cận chân GPIO bằng ngón emy hoặc bút stylus. Các GPIO cảm ứng này có thể được sử dụng để triển khai các miếng cảm ứng điện dung mà không cần bất kỳ phần cứng bổ sung nào.

**SPI**

Chip Wi-Fi ESP32 có ba khối SPI (SPI, HSPI và VSPI) ở cả chế độ master và chế độ slave. SPI được sử dụng để giao tiếp với bộ nhớ Flash. Vì vậy, bạn có hai giao diện SPI.

**I2C**

Có hai giao diện I2C trong ESP32 với sự linh hoạt hoàn toàn trong việc gán các chân, tức là, người dùng có thể gán các chân SCL và SDA cho cả hai giao diện I2C trong chương trình.

Nếu bạn đang sử dụng Arduino IDE, thì các chân I2C mặc định là

SDA - GPIO21

SCL - GPIO22

**PWM**

Bộ điều khiển PWM trong ESP32 có 16 kênh dạng sóng PWM độc lập với tần số và chu kỳ nhiệm vụ có thể định cấu hình. Dạng sóng PWM có thể được sử dụng để điều khiển động cơ và LED. Bạn có thể cấu hình tần số tín hiệu PWM, kênh, chân GPIO và cả chu kỳ nhiệm vụ.

1. LINH KIỆN VẬT TƯ

**Cảm biến**

1. Module ESP32

2. Cảm biến hồng ngoại

3. Còi báo

4. Relay

1. GHÉP NỐI THỬ NGHIỆM

1. Sơ đồ ghép nối

**A diagram of a computer system

Description automatically generated**

Hình 2. Sơ đồ ghép nối

2. Xây dựng chương trình

**Lưu đồ thuật toán**

A diagram of a flowchart

Description automatically generated

Hình 3. Lưu đồ thuật toán

**Viết code (viết chương trình)**

#include <Wire.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include <DHT.h>

#include "ThingSpeak.h"

const char\* WIFI\_NAME = "thành";

const char\* WIFI\_PASSWORD = "123456789";

const int myChannelNumber = 2281283;

const char\* myApiKey = "A2UQZ9N7ISHSMQLA";

const char\* server = "api.thingspeak.com";

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

WiFiClient client;

int camBienVao = 14;

int camBienRa = 12;

int giaTriVao;

int giaTriRa;

int den = 13;

int peopleCount = 0;

float temp = 28;

void setup()

{

Serial.begin(115200);

delay(10);

dht.begin();

WiFi.begin(WIFI\_NAME, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Wifi not connected");

}

Serial.println("Wifi connected !");

Serial.println("Local IP: " + String(WiFi.localIP()));

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

pinMode(camBienVao, INPUT);

pinMode(camBienRa, INPUT);

pinMode(den, OUTPUT);

digitalWrite(den, LOW);

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

else{

Serial.print("Nhiệt độ không khí");

Serial.println(t);

Serial.print("Độ ẩm: ");

Serial.println(h);

ThingSpeak.setField(3,t);

ThingSpeak.setField(4,h);

ThingSpeak.writeFields(myChannelNumber, myApiKey);

}

giaTriVao = digitalRead(camBienVao); //Đọc giá trị digital từ cảm biến và gán vào biến giaTriVao

giaTriRa = digitalRead(camBienRa); //Đọc giá trị digital từ cảm biến và gán vào biến giaTriVao

if (giaTriVao == 1)

{

digitalWrite(den, LOW);

}

else

{

peopleCount++;

if(temp>16){

temp--;

}else{

temp = 16;

}

Serial.print("Phat hien chuyen dong! So nguoi: ");

Serial.println(peopleCount);

Serial.print("Nhiệt độ điều hòa: ");

Serial.println(temp);

digitalWrite(den, HIGH);

delay(500);

digitalWrite(den, LOW);

ThingSpeak.setField(1,peopleCount);

ThingSpeak.setField(2,temp);

ThingSpeak.writeFields(myChannelNumber, myApiKey);

}

if (giaTriRa == 1){

digitalWrite(den, LOW);

}

else {

if(peopleCount>0){

peopleCount--;

}

else{

peopleCount = 0;

}

if(temp<28){

temp++;

}else{

temp = 28;

}

Serial.print("Phat hien chuyen dong! So nguoi: ");

Serial.println(peopleCount);

Serial.print("Nhiệt độ điều hòa: ");

Serial.println(temp);

digitalWrite(den, HIGH);

delay(500);

digitalWrite(den, LOW);

ThingSpeak.setField(1,peopleCount);

ThingSpeak.setField(2,temp);

ThingSpeak.writeFields(myChannelNumber, myApiKey);

}

delay(150);

}

**3. Nạp chương trình module phần cứng**

Sketch uses 740805 bytes (56%) of program storage space. Maximum is 1310720 bytes.

Global variables use 44392 bytes (13%) of dynamic memory, leaving 283288 bytes for local variables. Maximum is 327680 bytes.

esptool.py v4.5.1

Serial port COM3

Connecting.......

Chip is ESP32-D0WDQ6 (revision v1.0)

Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None

Crystal is 40MHz

MAC: 30:c6:f7:30:7f:c4

Uploading stub...

Running stub...

Stub running...

Configuring flash size...

Flash will be erased from 0x00001000 to 0x00005fff...

Flash will be erased from 0x00008000 to 0x00008fff...

Flash will be erased from 0x0000e000 to 0x0000ffff...

Flash will be erased from 0x00010000 to 0x000c6fff...

Compressed 17568 bytes to 12204...

Writing at 0x00001000... (100 %)

Wrote 17568 bytes (12204 compressed) at 0x00001000 in 1.4 seconds (effective 99.6 kbit/s)...

Hash of data verified.

Compressed 3072 bytes to 146...

Writing at 0x00008000... (100 %)

Wrote 3072 bytes (146 compressed) at 0x00008000 in 0.1 seconds (effective 302.9 kbit/s)...

Hash of data verified.

Compressed 8192 bytes to 47...

Writing at 0x0000e000... (100 %)

Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.2 seconds (effective 361.6 kbit/s)...

Hash of data verified.

Compressed 746560 bytes to 479541...

Writing at 0x00010000... (3 %)

Writing at 0x0001bc48... (6 %)

Writing at 0x00027808... (10 %)

Writing at 0x00033a07... (13 %)

Writing at 0x00038e37... (16 %)

Writing at 0x0003e3ac... (20 %)

Writing at 0x000439d4... (23 %)

Writing at 0x00048b7c... (26 %)

Writing at 0x0004dccf... (30 %)

Writing at 0x00052f76... (33 %)

Writing at 0x0005832e... (36 %)

Writing at 0x0005d4de... (40 %)

Writing at 0x000629fe... (43 %)

Writing at 0x00067c4c... (46 %)

Writing at 0x0006d7e4... (50 %)

Writing at 0x00072fa8... (53 %)

Writing at 0x0007817d... (56 %)

Writing at 0x0007d627... (60 %)

Writing at 0x000827fb... (63 %)

Writing at 0x00087fe0... (66 %)

Writing at 0x0008d74a... (70 %)

Writing at 0x0009345d... (73 %)

Writing at 0x00098f21... (76 %)

Writing at 0x0009eb10... (80 %)

Writing at 0x000a7f24... (83 %)

Writing at 0x000aedae... (86 %)

Writing at 0x000b4395... (90 %)

Writing at 0x000b9ba9... (93 %)

Writing at 0x000bf1e8... (96 %)

Writing at 0x000c4a87... (100 %)

Wrote 746560 bytes (479541 compressed) at 0x00010000 in 42.8 seconds (effective 139.6 kbit/s)...

Hash of data verified.

Leaving...

Hard resetting via RTS pin...

**4. Thử nghiệm hệ thống**

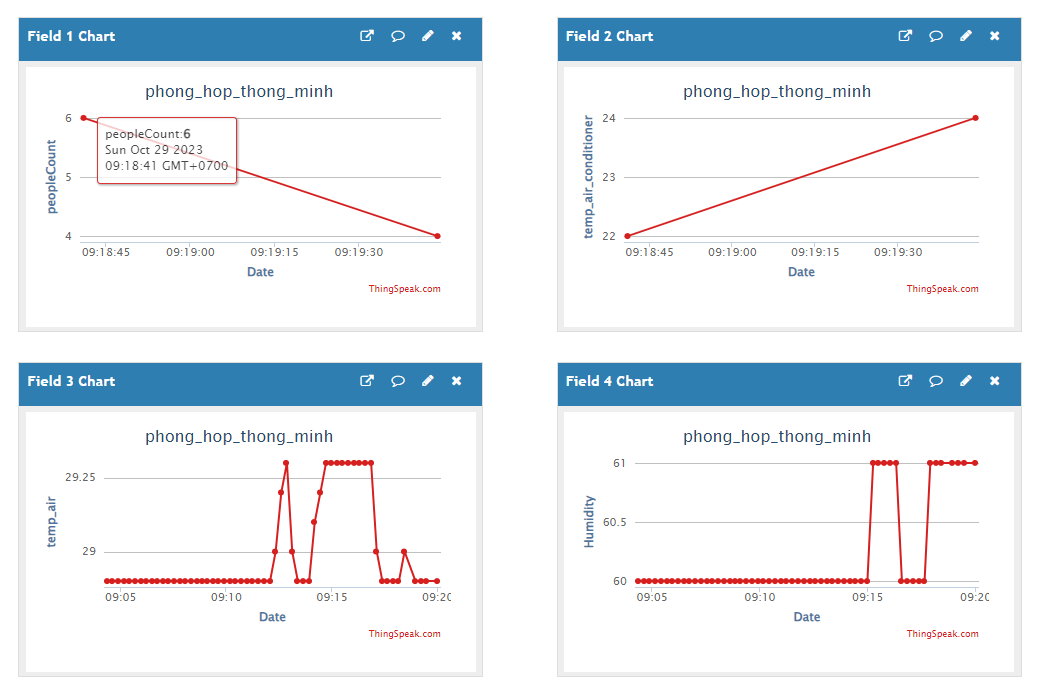
**Ghép nối thử trực tiếp trên board với máy tính**

A circuit board with many colored wires

Description automatically generated

Hình 4. Ghép nối thử trực tiếp trên board với máy tính

**Thử với hệ thống hoàn thiện với các điều kiện môi trường thực**



Hình 5. Thử với hệ thống hoàn thiện với các điều kiện môi trường thực

1. KẾT LUẬN

**Kết quả đạt được**

Dự án phòng hợp thông minh đã đạt được một số kết quả quan trọng trong quá trình phát triển. Chúng em đã thành công trong việc xây dựng một hệ thống giám sát và đo lường, cho phép đếm và ghi nhận số lượng người ra vào phòng một cách chính xác. Điều này cung cấp thông tin quan trọng về lư lượng người trong phòng, có thể hữu ích cho quản lý sự kiện hoặc giám sát an ninh.

Ngoài ra, hệ thống cũng có khả năng đo lường nhiệt độ và độ ẩm trong phòng, giúp theo dõi và duy trì môi trường phù hợp cho các hoạt động bên trong. Tuy nhiên, còn một số tính năng mà chúng tôi chưa thể hoàn thành. Điều này bao gồm khả năng điều chỉnh điều hòa không khí và chiếu sáng thông qua relay, điều quan trọng để tối ưu hóa sự thoải mái và tiết kiệm năng lượng. Chúng em đang tiếp tục phát triển để hoàn thiện các tính năng này và cải thiện tích hợp của hệ thống phòng hợp thông minh.

**Đề xuất hướng phát triển tiếp theo**

Hướng phát triển cho dự án phòng hợp thông minh có thể tập trung vào việc tích hợp hệ thống điều khiển thông minh cho điều hòa không khí và chiếu sáng thông qua relay. Điều này giúp cải thiện sự tiện lợi và hiệu suất tiết kiệm năng lượng cho người sử dụng. Đồng thời, chúng tôi có thể nghiên cứu và phát triển tích hợp trí tuệ nhân tạo (AI) và học máy để tự động điều chỉnh các thông số như nhiệt độ, độ ẩm, và chiếu sáng dựa trên mô hình sử dụng và thời tiết, tạo ra một hệ thống phòng hợp thông minh ngày càng thông minh và tiết kiệm năng lượng.

1. TÀI LIỆU THAM KHẢO

[1] <https://dientutuonglai.com/so-do-chan-esp32.html> , mục 5.