

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN**  
**IOT CƠ BẢN**

# **THIẾT KẾ HỆ THỐNG THEO DÕI THÔNG TIN MÔI TRƯỜNG**

Giảng viên hướng dẫn: **THẦY ĐỖ TRÍ NHỰT**

Sinh viên thực hiện: **TÔ VĨNH KHANG – 51800408**

**ĐỖ VŨ QUỐC TÙNG – 51800323**

**LÂM VĂN BẢO - 51702063**

Nhóm Lớp : **05**

Khoá : **22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN**  
**IOT CƠ BẢN**

# **THIẾT KẾ HỆ THỐNG THEO DÕI THÔNG TIN MÔI TRƯỜNG**

Giảng viên hướng dẫn: **THẦY ĐỖ TRÍ NHỰT**

Sinh viên thực hiện: **TÔ VĨNH KHANG – 51800408**

**ĐỖ VŨ QUỐC TÙNG – 51800323**

**LÂM VĂN BẢO - 51702063**

Nhóm Lớp : **05**

Khoá : **22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

## LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Khoa Công nghệ thông tin và Trường Đại học Tôn Đức Thắng đã tạo điều kiện cho chúng em được học tập trong suốt thời gian qua. Chân thành cảm ơn Thầy Đỗ Trí Nhựt đã giúp chúng em có thêm kiến thức về IOT, có cái nhìn tổng quan rộng mở hơn về thế giới xung quanh thông qua các thiết bị thông minh. Cho chúng em được thử sức với một phạm trù môn học mới mẻ, đầy tính thực tiễn và tiềm năng trong tương lai gần.

Trong quá trình thực hiện bài báo cáo này nhóm vẫn khó tránh khỏi những sai sót không mong muốn, kính mong thầy có thể góp ý và giúp đỡ chúng em. Nhóm xin chân thành cảm ơn Thầy.

## **ĐỒ ÁN NÀY ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng em xin cam đoan đây là sản phẩm của riêng chúng em được sự hướng dẫn của Thầy Đỗ Trí Nhựt. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính chúng em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài báo cáo này còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 01 tháng 12 năm 2021

Tác giả

(ký tên và ghi rõ họ tên)

Tô Vĩnh Khang

(Trưởng nhóm)

Đỗ Vũ Quốc Tùng

Lâm Văn Bảo

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm

(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm

(kí và ghi họ tên)

## TÓM TẮT

Ở bài báo cáo này, chúng em đã tìm hiểu tổng quan về Hệ thống theo dõi thông tin môi trường để thu thập thông tin về nhiệt độ, độ ẩm và chuyển động thông qua các cảm biến tạo nên một thiết bị thông minh nhỏ gọn tiện lợi nhằm mục đích giúp cho những người dân vùng nông thôn, vùng quê nghèo đang có các hoạt động canh tác chăn nuôi nhỏ lẻ tại khu vực sân vườn của họ.

Qua việc tìm hiểu từ nhiều nguồn, nhóm đã đúc kết và tiến hành mô hình hóa hệ thống. Chọn các thành phần phần cứng cần thiết gồm vi điều khiển giá rẻ, năng lượng thấp có hỗ trợ WiFi mang tên ESP32 cùng cảm biến chuyển động, cảm biến nhiệt độ, độ ẩm và một số linh kiện điện tử liên quan. Đồng thời nghiên cứu nguyên lý hoạt động của chúng, cùng nhau trao đổi để làm mô phỏng trên phần mềm Cisco Packet Tracer để bảo đảm mọi thứ hoạt động được.

Tiếp theo đó, nhóm sử dụng công cụ thiết kế mạch điện tử trực tuyến Circuito.io để đảm bảo các mạch được nối dây đúng đắn, không còn lỗi phát sinh. Khi thiết bị đã có thể hiển thị dữ liệu môi trường trên màn hình LCD cũng như báo còi khi có chuyển động thành cộng, nhóm tiến hành đưa dữ liệu thu thập được lên Dịch vụ nền tảng phân tích ThingSpeak để trực quan hóa dữ liệu.

## MỤC LỤC

LỜI CẢM ƠN.....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN.....	iii
TÓM TẮT.....	iv
MỤC LỤC.....	1
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT.....	4
DANH MỤC CÁC HÌNH VÀ BẢNG.....	5
CHƯƠNG 1 – GIỚI THIỆU CHUNG.....	7
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT.....	9
2.1 Phần cứng.....	9
2.1.1 Vi điều khiển ESP32 DevKitC.....	11
2.1.2 Cảm biến nhiệt độ, độ ẩm DHT-11.....	12
2.1.3 Cảm biến chuyển động PIR HC-SR501.....	12
2.1.4 Màn hình LCD 1602 Tích hợp chuyển đổi I2C.....	13
2.1.5 Đèn LED.....	14
2.1.6 Điện trở cấm.....	14
2.1.7 Còi Buzzer 5V.....	15
2.1.8 Bóng bán dẫn NPN BC337.....	15
2.2 Các tiện ích bổ sung.....	16
2.2.1 Công cụ mô phỏng trực quan đa nền tảng - Cisco Packet Tracer..	16
2.2.2 Công cụ thiết kế mạch điện tử trực tuyến - Circuito.io.....	16
2.2.3 Môi trường phát triển tích hợp - Arduino IDE.....	17
2.2.4 Dịch vụ nền tảng phân tích - ThingSpeak.....	18
CHƯƠNG 3 - PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	19
3.1 Mô hình hóa.....	19
3.1.1 Sơ đồ Tổng quát hệ thống.....	19

3.1.2 Sơ đồ Hoạt động chính.....	19
3.1.2.1 Thu thập thông tin về Nhiệt độ và Độ ẩm.....	19
3.1.2.2 Thu thập thông tin về Chuyển động.....	20
3.2 Mô phỏng.....	21
3.2.1 Tạo các kết nối cho mô phỏng.....	21
3.2.2 Lập trình khối cho mạch mô phỏng.....	21
3.2.3 Chạy mô phỏng.....	22
3.3 Mô hình mạch khi triển khai thực tế.....	23
CHƯƠNG 4 - TRIỂN KHAI MẠCH THỰC TẾ.....	24
4.1 Lập trình lệnh cho mạch thực tế.....	24
4.1.1 Khai báo các thư viện sử dụng.....	24
4.1.2 Khai báo các biến.....	24
4.1.3 Khởi tạo các hàm con.....	25
4.1.3.1 Kết nối với internet.....	25
4.1.3.2 Bật tắt còi và đèn khi phát hiện chuyển động.....	26
4.1.3.3 Hiển thị nhiệt độ và độ ẩm ra màn hình LCD.....	26
4.1.3.4 Đưa dữ liệu lên dịch vụ ThingSpeak.....	27
4.1.4 Khởi chạy hệ thống.....	28
4.1.5 Vòng lặp chương trình.....	29
4.2 Kết quả đạt được.....	30
4.2.1 Kết quả trên Màn hình Serial.....	30
4.2.2 Kết quả trên Mạch thực tế.....	31
4.2.3 Kết quả trên ThingSpeak.....	33
CHƯƠNG 5 – TỔNG KẾT.....	34
5.1 Nhận xét kết quả.....	34
5.2 Thuận lợi và khó khăn.....	34
5.3 Định hướng phát triển.....	34



PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ.....	35
TÀI LIỆU THAM KHẢO.....	37

## **DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

IOT	Internet of Things
LED	Light Emitting Diode
PIR	Passive Infra-Red
WiFi	Wireless Fidelity
LCD	Liquid Crystal Display
DC	Direct Current
USB	Universal Serial Bus
I2C	Inter Integrated Circuit
IDE	Integrated Development Environment

## DANH MỤC CÁC HÌNH VÀ BẢNG

### DANH MỤC HÌNH

Hình 1.1 Hình ảnh về Mô tả tương tác của mạng lưới thiết bị kết nối Internet.....	7
Hình 1.2 Hình ảnh về Một số ứng dụng IOT hàng đầu hiện nay.....	8
Hình 2.1 Hình ảnh về Vi điều khiển ESP32 DevKitC.....	11
Hình 2.2 Hình ảnh về Thông tin các cổng của Vi điều khiển ESP32 DevKitC.....	11
Hình 2.3 Hình ảnh về Cảm biến nhiệt độ, độ ẩm DHT-11.....	12
Hình 2.4 Hình ảnh về Cảm biến chuyển động PIR HC-SR501.....	12
Hình 2.5 Hình ảnh về Màn hình LCD 1602 Tích hợp chuyển đổi I2C.....	13
Hình 2.6 Hình ảnh về Một số màu đèn LED.....	14
Hình 2.7 Hình ảnh về Điện trở cấm.....	14
Hình 2.8 Hình ảnh về Còi Buzzer 5V.....	15
Hình 2.9 Hình ảnh về Bóng bán dẫn NPN BC337.....	15
Hình 2.10 Hình ảnh về Công cụ Cisco Packet Tracer.....	16
Hình 2.11 Hình ảnh về Công cụ Circuito.io.....	16
Hình 2.12 Hình ảnh về Arduino.....	17
Hình 2.13 Hình ảnh về Giao diện ArduinoIDE.....	17
Hình 2.14 Hình ảnh về Nguyên lý hoạt động của ThingSpeak.....	18
Hình 3.1 Hình ảnh về Sơ đồ Mô hình hóa tổng quát hệ thống.....	19
Hình 3.2 Hình ảnh về Sơ đồ hoạt động Thu thập thông tin nhiệt độ, độ ẩm.....	20
Hình 3.3 Hình ảnh về Sơ đồ hoạt động Thu thập thông tin về chuyển động.....	20
Hình 3.4 Hình ảnh về Tạo các kết nối cho việc mô phỏng.....	21
Hình 3.5 Hình ảnh về Lập trình khối cho mạch mô phỏng.....	22
Hình 3.6 Hình ảnh về Kết quả khi chạy mô phỏng hệ thống.....	22
Hình 3.7 Hình ảnh về Các kết nối chân linh kiện.....	23
Hình 4.1 Hình ảnh về Các thông tin thu thập được trên màn hình Serial.....	30

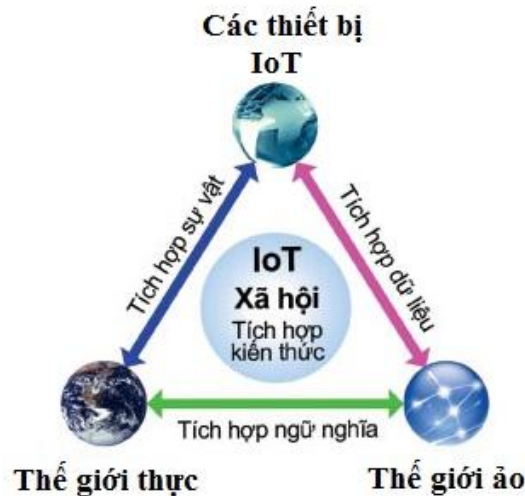
Hình 4.2 Hình ảnh về Tổng quan sản phẩm mạch thực tế.....	31
Hình 4.3 Hình ảnh về Số liệu nhiệt độ và độ ẩm hiển thị trên màn hình LCD.....	31
Hình 4.4 Hình ảnh về Sự thay đổi số liệu khi có tác động bởi môi trường bên ngoài...	31
Hình 4.5 Hình ảnh về Trạng thái khi không phát hiện chuyển động.....	32
Hình 4.6 Hình ảnh về Trạng thái khi có phát hiện chuyển động.....	32
Hình 4.7 Hình ảnh về Trực quan hóa dữ liệu Nhiệt độ trên ThingSpeak.....	33
Hình 4.8 Hình ảnh về Trực quan hóa dữ liệu Độ ẩm trên ThingSpeak.....	33

## **DANH MỤC BẢNG**

Bảng 2.1 Bảng các thông tin chi tiết phần cứng của hệ thống.....	10
--	----

## CHƯƠNG 1 – GIỚI THIỆU CHUNG

Ngày nay, công nghệ đã và đang đóng vai trò quan trọng trong xã hội. Nó giúp con người làm việc thoải mái hơn, dễ dàng hơn. Và Internet Vạn Vật (Tiếng anh: Internet of Things - IOT) là một trong số đó.



Hình 1.1 Hình ảnh về Mô tả tương tác của mạng lưới thiết bị kết nối Internet

Nó là một liên mạng, trong đó các thiết bị được nhúng với các bộ phận điện tử, phần mềm, cảm biến, cơ cấu chấp hành cùng với khả năng kết nối mạng máy tính giúp cho các thiết bị này có thể thu thập và truyền tải dữ liệu. Những vật vô tri vô giác trước đây như tay nắm cửa, bóng đèn,... giờ đây có thể được trang bị một bộ cảm biến thông minh có thể thu thập và truyền dữ liệu vào mạng. Người ta ước tính có khoảng 3 triệu thiết bị mới được kết nối với Internet mỗi tháng và đến năm 2025, sẽ có hơn 75 tỷ thiết bị được kết nối trên toàn thế giới, 2/3 trong số đó sẽ là cảm biến, thiết bị truyền động và các thiết bị thông minh mới được phát minh để giám sát, điều khiển, phân tích và tối ưu hóa thế giới của chúng ta. Nó đang là một xu hướng ngày càng nóng trên thế giới bởi tính ứng dụng vào đời sống của nó là vô cùng tiềm năng.



Hình 1.2 Hình ảnh về Một số ứng dụng IOT hàng đầu hiện nay

Một số ứng dụng được xem là hàng đầu trong nghiên cứu về lĩnh vực IOT như là Nhà thông minh, Thiết bị có thể đeo được, Giao thông thông minh, Y tế thông minh, Hạ tầng thông minh, Thành phố thông minh,...Trong đó, Chăn nuôi thông minh thường bị bỏ qua trong khi nước ta lại là nước có thể mạnh về trồng trọt chăn nuôi.

Chính vì lẽ này, nhóm đã chọn đề tài Hệ thống theo dõi thông tin môi trường để giúp mọi người có cái nhìn khác hơn, quan tâm hơn về vấn đề phân tích dữ liệu môi trường trong hoạt động canh tác chăn nuôi ở vùng quê nông thôn. Tuy chỉ với quy mô nhỏ nhằm mục đích học tập nhưng thông qua thiết bị nhỏ bé mà nhóm dự định làm sẽ được đặt ở các cửa ra vào, trong các sân vườn, gian nhà,...để đo đạc và thống kê lại các thông số của môi trường khu vực đó. Từ đó, ta biết cách để kiểm soát nhiệt độ, độ ẩm cho phù hợp như tưới phun sương giúp giảm nhiệt độ khi trời quá nóng chẳng hạn. Hay thậm chí là phát hiện các chuyển động để biết được có kẻ gian vào vườn trộm cắp. Đặc biệt là vùng thôn quê đa số các nhà đều nuôi gà và luôn xảy ra tình trạng trộm cắp.

Việc ứng dụng IOT vào đời sống sẽ giúp quản lý và tổ chức công việc đạt hiệu quả cao. Góp phần làm giảm đi những công việc tay chân cực nhọc, giảm đi sự can thiệp của con người vào các quy trình trong mọi lĩnh vực của đời sống xã hội. Thúc đẩy sự nghiên cứu và phát triển trí tuệ của nhân loại.

## CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

### 2.1 Phần cứng

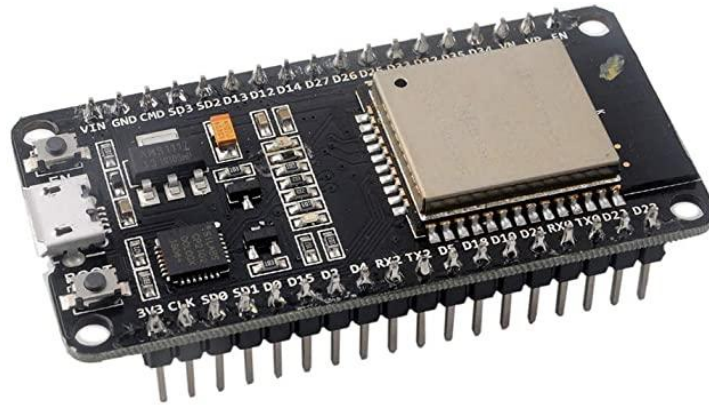
STT	Tên linh kiện	Số Lượng	Thông số kỹ thuật
1	Vi điều khiển ESP32 DevKitC	1	Điện áp: 5V DC CPU: Wifi BLE Soc ESP32 Dual core (Tốc độ xử lý 160-240MHz) RAM: 320KB SRAM Tích hợp mạch nạp và giao tiếp UART CP2102. Tích hợp Led Status, nút BOOT và UNABLE. Kích thước: 28.33 x 51.45mm.
2	Cảm biến nhiệt độ, độ ẩm DHT-11	1	Điện áp: 3-5V DC Dòng điện: 2.5mA max Đo tốt ở độ ẩm 20-70%RH với sai số 5%. Đo tốt ở nhiệt độ 0-50°C với sai số $\pm 2^{\circ}\text{C}$ . Tần số lấy mẫu: 1Hz max (1s/lần) Kích thước: 15 x 12 x 5.5mm.
3	Cảm biến chuyển động PIR HC-SR501	1	Điện áp: 4.5-20V DC Có 2 chế độ hoạt động: (L) không lặp lại kích hoạt và (H) lặp lại kích hoạt. Độ trễ: 0.5-200s Góc quét: <100 độ Sử dụng cảm biến: 500 BP Trọng lượng: 6g Khoảng cách phát hiện: 2-4.5m Kích thước: 32 x 24mm

4	Màn hình LCD 1602 Tích hợp chuyển đổi I2C	1	Điện áp: 2.5-6V DC Driver sử dụng: HD44780 Khả năng hiển thị: 2 dòng (16 ký tự/dòng) Có giao tiếp I2C Trọng lượng: 5g Kích thước: 41.5 x 19 x 15.3mm
5	Đèn LED	2	Dòng điện bình thường: 20mA Điện áp: 2,5V max Nhiệt độ hoạt động: -40-85°C
6	Điện trở cảm	4	Mô hình: 100/1K/10K $\Omega$ - 1/4W (Sai số: 5%) Nhiệt độ hoạt động: -55-155°C Loại: Điện trở cố định Linh kiện xuyên lỗ: 0.5mm
7	Còi Buzzer 5V	1	Điện áp: 1.5-5V DC Tần số hoạt động: 2-5Khz Trọng lượng: 1g Kích thước: 12 x 8.5mm.
8	Bóng bán dẫn NPN BC337	1	Loại bóng bán dẫn NPN Dòng điện: 800mA max Điện áp: 45V max Công suất: 625mW max Tần số: 210MHz

Bảng 2.1 Bảng các thông tin chi tiết phần cứng của hệ thống

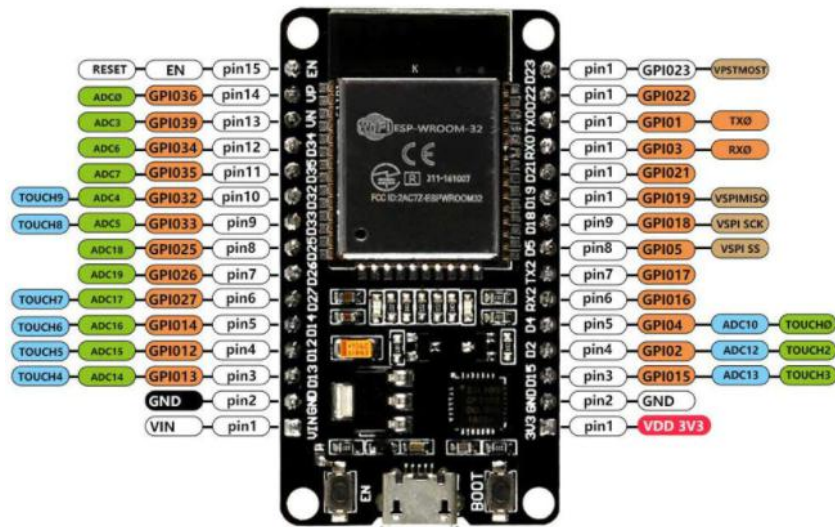


### 2.1.1 Vi điều khiển ESP32 DevKitC



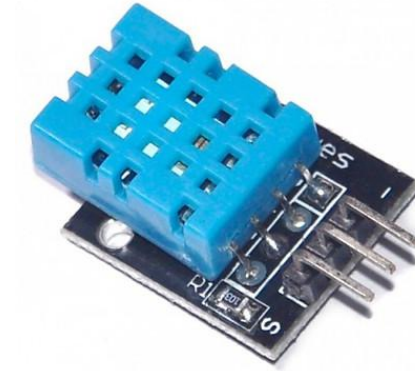
Hình 2.1 Hình ảnh về Vi điều khiển ESP32 DevKitC

Là một vi điều khiển giá rẻ, năng lượng thấp có tích hợp WiFi và Bluetooth. Nó được sử dụng rất nhiều lĩnh vực IOT hiện nay. Sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.



Hình 2.2 Hình ảnh về Thông tin các cổng của Vi điều khiển ESP32 DevKitC

### 2.1.2 Cảm biến nhiệt độ, độ ẩm DHT-11



Hình 2.3 Hình ảnh về Cảm biến nhiệt độ, độ ẩm DHT-11

Là cảm biến độ ẩm và nhiệt độ giá rẻ, khá thông dụng cho các thiết bị ứng dụng IOT hiện nay với điểm mạnh là rất dễ lấy dữ liệu thông qua giao tiếp 1-wire (giao tiếp digital một dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu được tích hợp trong cảm biến giúp cho việc dữ liệu nhận về được chính xác mà không cần phải trải qua bất kỳ công đoạn tính toán.

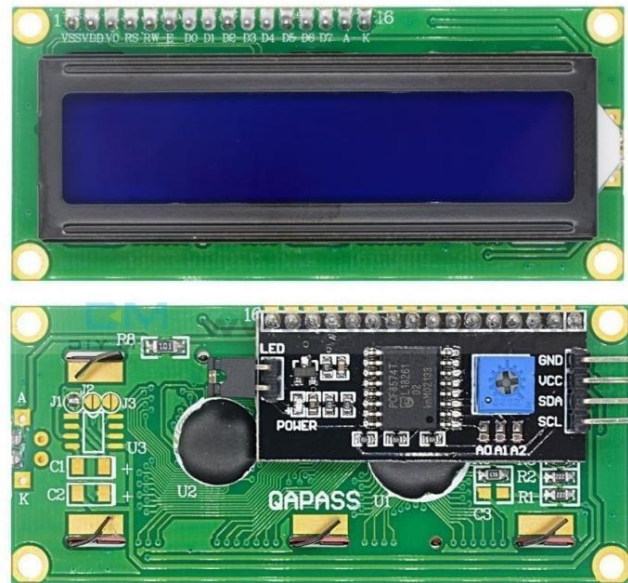
### 2.1.3 Cảm biến chuyển động PIR HC-SR501



Hình 2.4 Hình ảnh về Cảm biến chuyển động PIR HC-SR501

Là cảm biến thụ động dùng nguồn kích thích là tia hồng ngoại (IR) - các tia nhiệt phát ra từ các vật thể nóng như thân thể con người (hay nguồn nhiệt bất kì). Cấu tạo của nó được hiểu đơn giản là một lăng kính (thường làm bằng plastic và chế tạo theo kiểu lăng kính fresnel) chắn trước mắt các mắt cảm biến. Nó có tác dụng chặn lại và phân thành nhiều vùng cho phép tia hồng ngoại đi vào mắt cảm biến, hai đơn vị của mắt cảm biến có tác dụng phân thành hai điện cực âm-dương. Khi hai điện cực này được tuần tự kích hoạt thì sẽ sinh ra một xung điện kích hoạt cảm biến. Cảm biến này gọi là thụ động vì nó không dùng nguồn nhiệt tự phát mà chỉ phụ thuộc vào nguồn nhiệt bên ngoài.

#### 2.1.4 Màn hình LCD 1602 Tích hợp chuyển đổi I2C



Hình 2.5 Hình ảnh về Màn hình LCD 1602 Tích hợp chuyển đổi I2C

Là một chiếc màn hình mini phổ biến, giá thành lại rẻ cùng với khả năng hiển thị 2 dòng, 16 ký tự mỗi dòng. Nó được tích hợp mô-đun giao tiếp I2C giúp giao tiếp dễ dàng, nhanh chóng mà không phải hàn I2C. Thông thường, nó được ứng dụng trong việc hiển thị các đoạn chữ để kiểm tra kết quả lập trình, hiển thị thông tin về nhiệt độ, độ ẩm, chuyển động,... cũng được đưa vào chiếc màn hình nhỏ bé này.

### 2.1.5 Đèn LED



Hình 2.6 Hình ảnh về Một số màu đèn LED

Là các đi-ốt có khả năng phát ra ánh sáng hay tia hồng ngoại, tử ngoại. Nó được cấu tạo từ một khối bán dẫn loại p ghép với một khối bán dẫn loại n. Khối bán dẫn loại p chứa nhiều lỗ trống tự do mang điện tích dương nên khi ghép với khối bán dẫn n chứa các điện tử tự do thì các lỗ trống này có xu hướng chuyển động khuếch tán sang khối n. Cùng lúc khối p lại nhận thêm các điện tích âm từ khối n chuyển sang. Kết quả là khối p tích điện âm (thiếu hụt lỗ trống và dư thừa điện tử) trong khi khối n tích điện dương (thiếu hụt điện tử và dư thừa lỗ trống).

### 2.1.6 Điện trở cảm



Hình 2.7 Hình ảnh về Điện trở cảm

Là linh kiện điện tử thụ động gồm 2 tiếp điểm kết nối, thường được dùng để hạn chế cường độ dòng điện chảy trong mạch, điều chỉnh mức độ tín hiệu, chia điện áp, kích hoạt các linh kiện điện tử chủ động như bóng bán dẫn, tiếp điểm cuối trong đường truyền điện,...

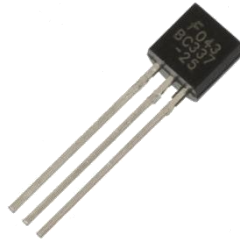
### 2.1.7 Còi Buzzer 5V



Hình 2.8 Hình ảnh về Còi Buzzer 5V

Là còi chip hoặc còi xung, phát ra âm thanh bíp bíp. Với thiết kế nhỏ gọn, tuổi thọ cao, nó thường được dùng trong các mạch điện tử nhằm mục đích báo động.

### 2.1.8 Bóng bán dẫn NPN BC337



Hình 2.9 Hình ảnh về Bóng bán dẫn NPN BC337

Là một linh kiện bán dẫn cần thiết cho việc xử lý và hoạt động của nhiều sản phẩm điện tử khác nhau, chức năng chủ yếu được sử dụng để khuếch đại các luồng tín hiệu.

## 2.2 Các tiện ích bổ sung

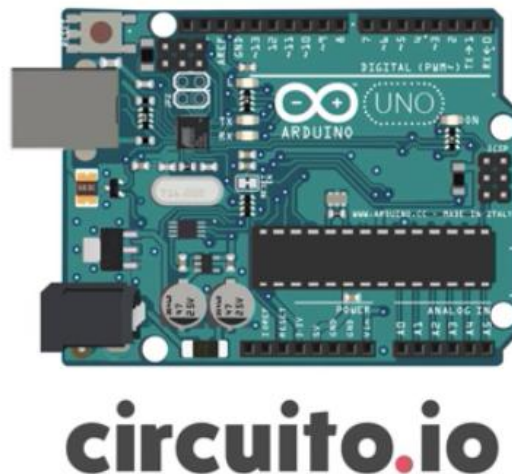
### 2.2.1 Công cụ mô phỏng trực quan đa nền tảng - Cisco Packet Tracer



Hình 2.10 Hình ảnh về Công cụ Cisco Packet Tracer

Là một công cụ mô phỏng trực quan đa nền tảng cho phép người dùng tạo cấu trúc liên kết mạng và bắt chước các mạng máy tính hiện đại, mô phỏng cấu hình của bộ định tuyến và chuyển mạch bằng giao diện dòng lệnh mô phỏng. Nó sử dụng kéo và thả giao diện người dùng, cho phép người dùng thêm và xóa các thiết bị mạng mô phỏng khi họ thấy phù hợp. Ngoài ra, nó hỗ trợ cả lập trình khối giúp cho việc lập trình trở nên nhẹ nhàng hơn.

### 2.2.2 Công cụ thiết kế mạch điện tử trực tuyến - Circuito.io



Hình 2.11 Hình ảnh về Công cụ Circuito.io

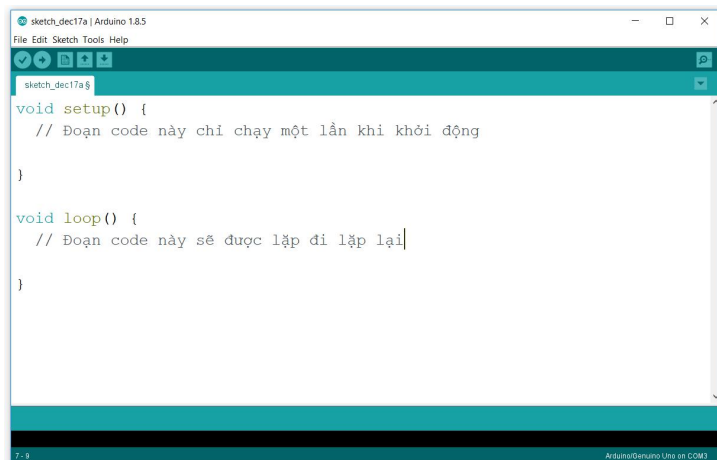
Là một công cụ trực tuyến để thiết kế các mạch điện tử hoàn chỉnh. Ưu điểm của nó là có thể tạo ra nhanh việc nối dây giữa các thiết bị chỉ bằng cách kéo thả thiết bị vào và đồng thời hỗ trợ bổ sung các bóng bán dẫn hoặc điện trở khi cần thiết, giúp người dùng có thể code ngay lập tức và chính xác cho mạch điện tử. Nhược điểm của nó không thể chạy mô phỏng trực tuyến cho người dùng chuyên nghiệp như Tinkercad.

### 2.2.3 Môi trường phát triển tích hợp - Arduino IDE



Hình 2.12 Hình ảnh về Arduino

Là một môi trường phát triển tích hợp để viết và nạp các đoạn mã chương trình vào các thiết bị điện tử như ESP32, Arduino,... Nó hỗ trợ ngôn ngữ C, C++. Cung cấp một thư viện phần mềm từ dự án Wiring và nhiều thủ tục đầu vào và đầu ra phổ biến. Hai hàm cơ bản là `setup()` để khởi động bản phác thảo và `loop()` để tạo vòng lặp chương trình chính. Nó được biên dịch và liên kết với một chương trình gốc `main()` thành một chương trình điều hành tuần hoàn.

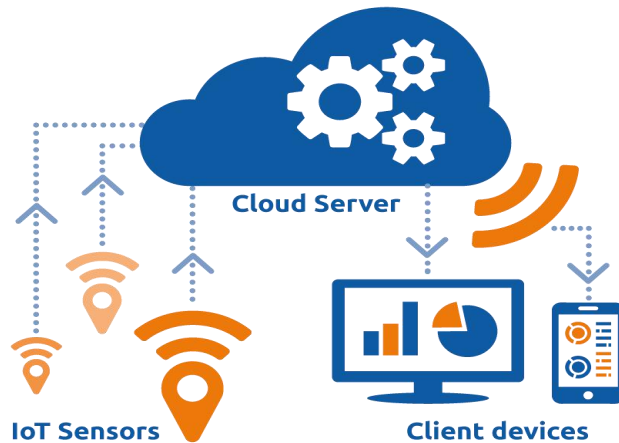


Hình 2.13 Hình ảnh về Giao diện ArduinoIDE



Với sự phổ biến ngày càng tăng của Arduino như một nền tảng phần mềm, các nhà cung cấp khác bắt đầu triển khai các trình biên dịch và công cụ mã nguồn mở tùy chỉnh có thể xây dựng và tải lên các bản phác thảo cho các bộ vi điều khiển khác không được dòng vi điều khiển chính thức của Arduino hỗ trợ.

#### 2.2.4 Dịch vụ nền tảng phân tích - ThingSpeak



Hình 2.14 Hình ảnh về Nguyên lý hoạt động của ThingSpeak

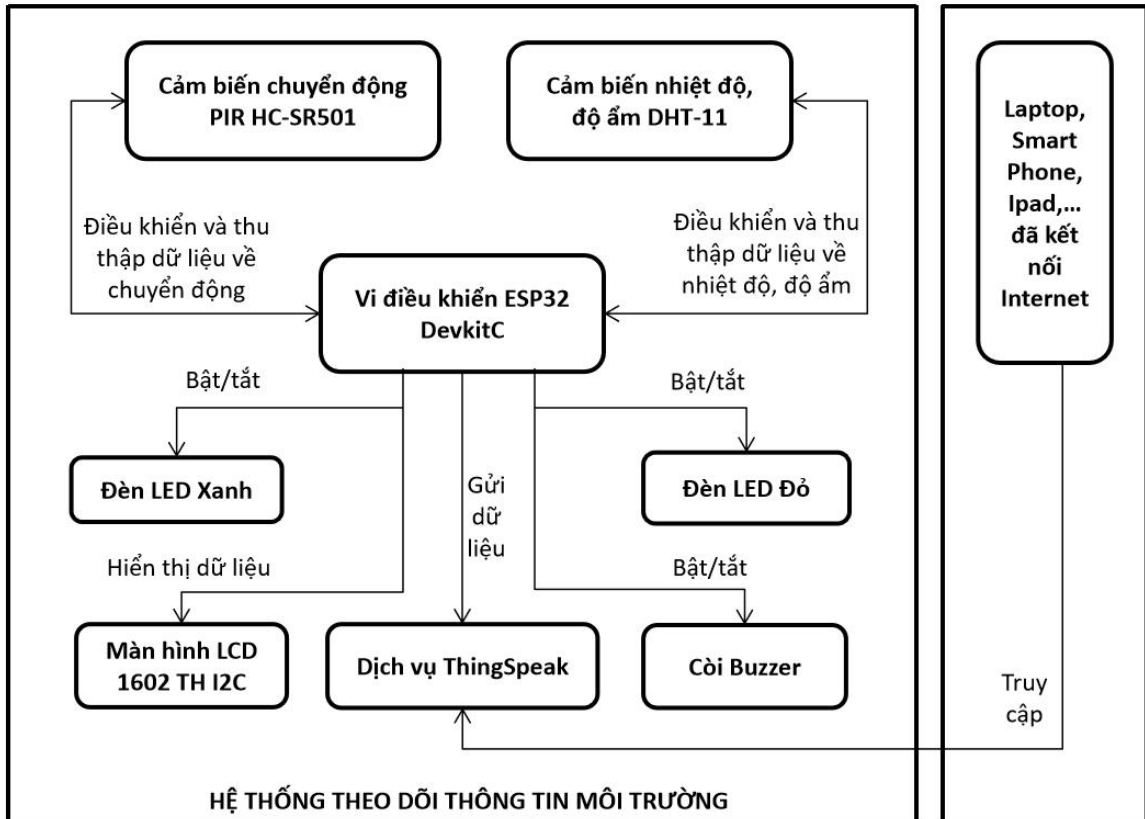
Là một dịch vụ nền tảng phân tích IOT cho phép tổng hợp, trực quan hóa và phân tích các luồng dữ liệu thành biểu đồ trực tiếp theo thời gian thực trên đám mây. Nó giúp cho việc xây dựng hệ thống IOT mà không cần thiết lập máy chủ hoặc phát triển phần mềm web. Chẳng hạn như khi muốn theo dõi nhiệt độ, độ ẩm của vườn nuôi gia súc chỉ cần đặt thiết bị khu vực đó và kết nối với tài khoản ThingSpeak. Thông qua đó, ta sẽ xem được các thông tin được thu thập trực tiếp liên tục tại vườn nuôi đó.



## CHƯƠNG 3 - PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1 Mô hình hóa

#### 3.1.1 Sơ đồ Tổng quát hệ thống

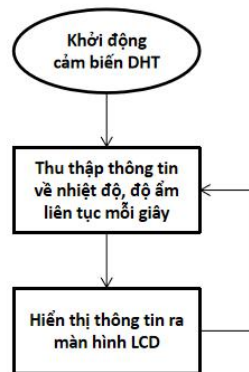


Hình 3.1 Hình ảnh về Sơ đồ Mô hình hóa tổng quát hệ thống

#### 3.1.2 Sơ đồ Hoạt động chính

##### 3.1.2.1 Thu thập thông tin về Nhiệt độ và Độ ẩm

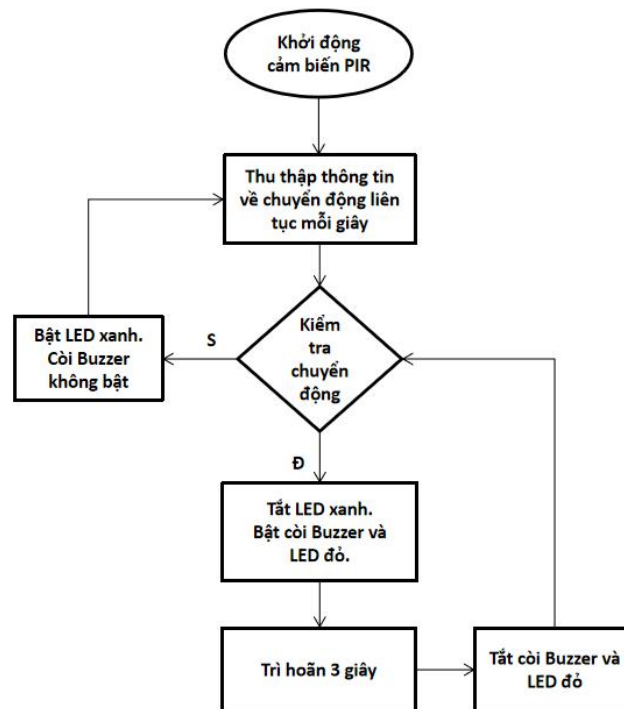
Cảm biến nhiệt độ, độ ẩm DHT-11 sẽ đo nhiệt độ, độ ẩm của khu vực gần nó liên tục với tần số 1Hz (1 lần/s) và cập nhật liên tục lên màn hình LCD 1602. Và vi điều khiển ESP32 DevKitC sẽ gửi những dữ liệu thu thập được lên ThingSpeak.



Hình 3.2 Hình ảnh về Sơ đồ hoạt động Thu thập thông tin nhiệt độ, độ ẩm

### 3.1.2.2 Thu thập thông tin về Chuyển động

Cảm biến chuyển động PIR HC-SR501 quét xanh khu vực có bán kính 5-7m để kiểm tra sự chuyển động. Ở trạng thái không có chuyển động (bình thường) thì còi Buzzer sẽ tắt và đèn LED xanh bật. Ở trạng thái có phát hiện chuyển động thì còi Buzzer sẽ được bật lên cùng với đèn LED đỏ và đồng thời đèn LED xanh sẽ được tắt đi.



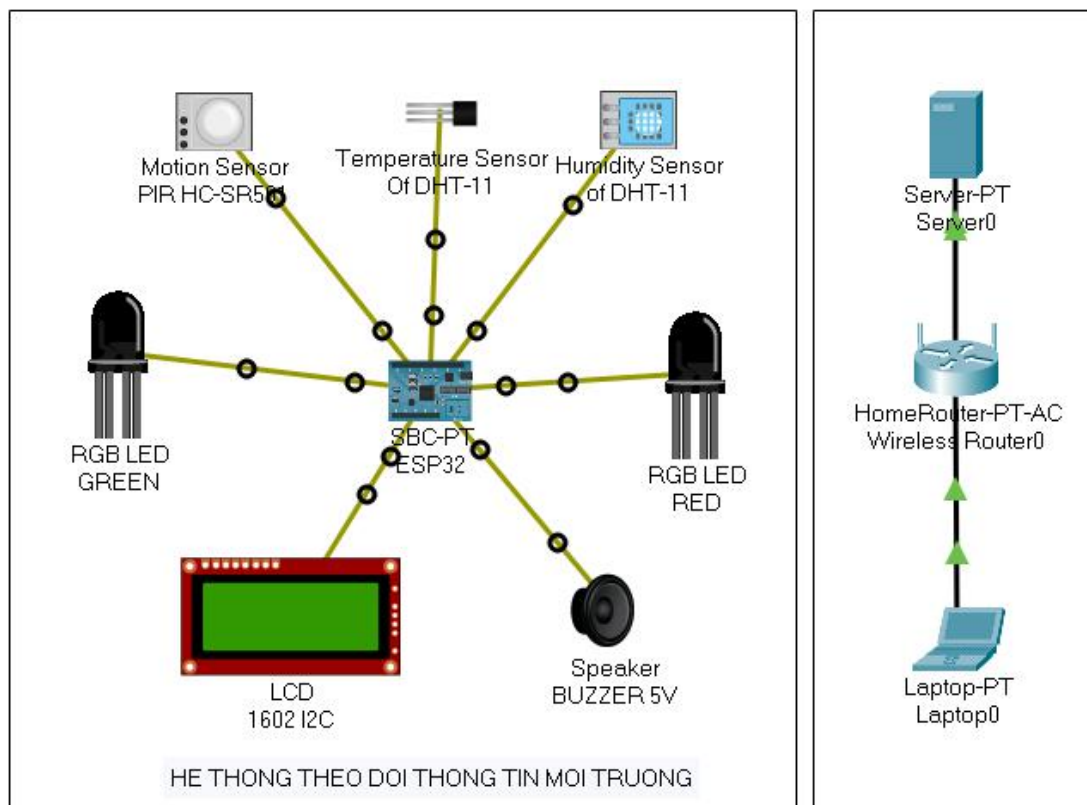
Hình 3.3 Hình ảnh về Sơ đồ hoạt động Thu thập thông tin về chuyển động

### 3.2 Mô phỏng

Trước khi triển khai mạch thực tế cần mô phỏng để đảm bảo về mặt logic hoạt động khớp với ý tưởng đề ra. Do đó, nhóm sử dụng công cụ mô phỏng trực quan đa nền tảng mang tên Cisco Packet Tracer để thực hiện việc này thông qua 3 giai đoạn.

#### 3.2.1 Tạo các kết nối cho mô phỏng

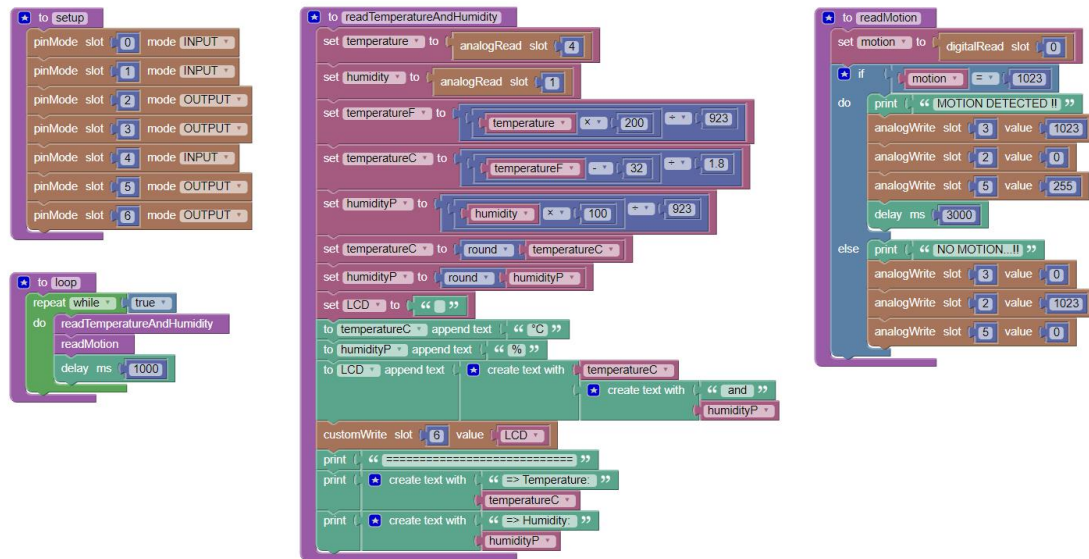
Dựa vào sơ đồ mô hình hóa tổng quát hệ thống (Hình 3.1), thực hiện các kết nối dây tương ứng để chuẩn bị cho việc lập trình khối tiếp theo.



Hình 3.4 Hình ảnh về Tạo các kết nối cho việc mô phỏng

#### 3.2.2 Lập trình khối cho mạch mô phỏng

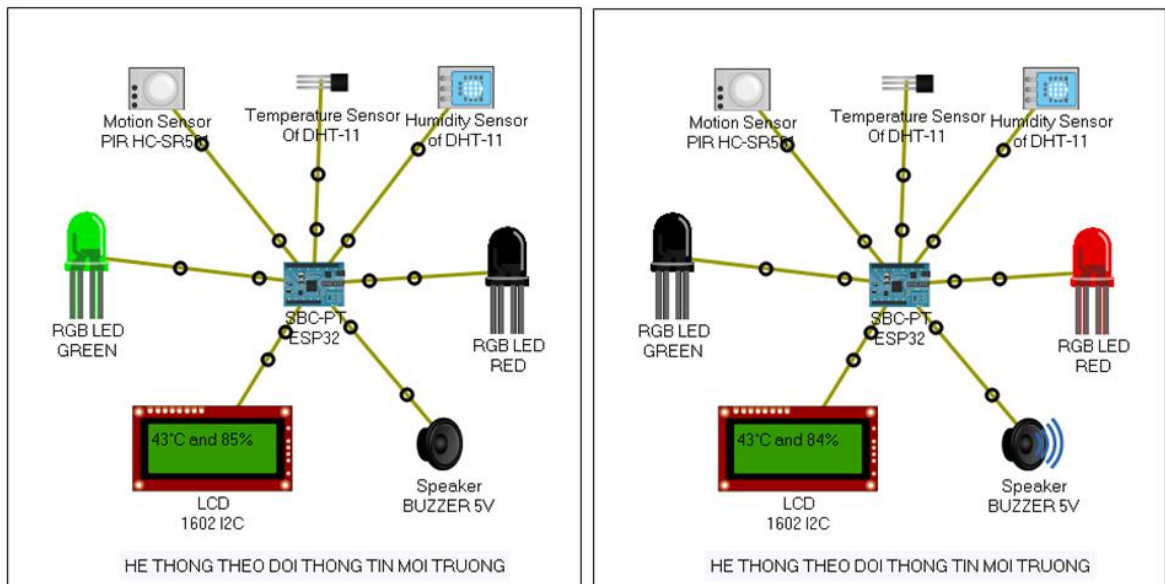
Thực hiện lập trình khối cho vi điều khiển như sau:



Hình 3.5 Hình ảnh về Lập trình khối cho mạch mô phỏng

Hàm setup() sẽ đảm nhận việc khởi động pin các linh kiện. Hàm loop() sẽ thực hiện vòng lặp giúp lấy thông tin liên tục từ việc đọc các cảm biến nhiệt độ độ ẩm trong hàm readTemperatureAndHumidity() và phát hiện chuyển động trong hàm readMotion().

### 3.2.3 Chạy mô phỏng

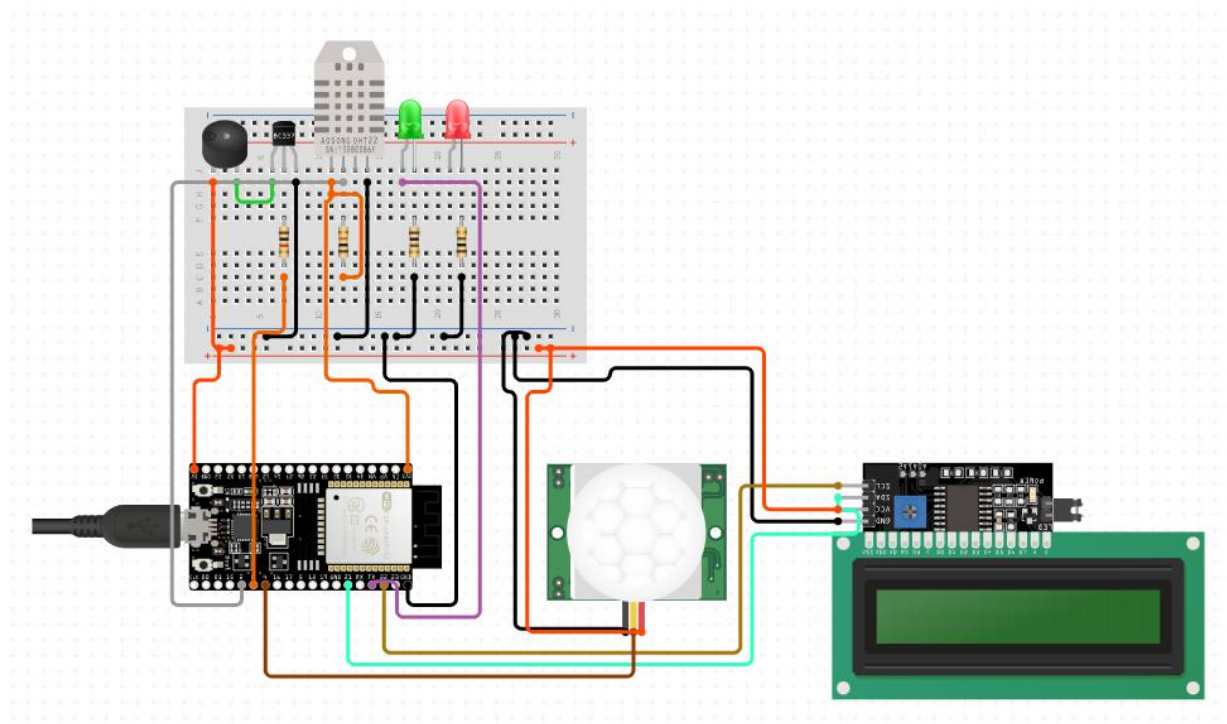


Hình 3.6 Hình ảnh về Kết quả khi chạy mô phỏng hệ thống

Tại Hình 3.6, mạch được nạp điện và đang hoạt động thu thập thông tin về nhiệt độ, độ ẩm, và kiểm tra chuyển động. Kết quả khi chạy mô phỏng cho thấy hình phía bên trái mang ý nghĩa đang ở trạng thái bình thường (không phát hiện chuyển động nào), hình bên phải mang ý nghĩa trạng thái báo động (có phát hiện chuyển động thông qua con chuột man hình vừa lướt qua cảm biến chuyển động). Với việc lập trình khối đã giúp cho việc chạy mô phỏng diễn ra đúng với ý tưởng, logic hoạt động của hệ thống theo dõi thông tin môi trường.

### 3.3 Mô hình mạch khi triển khai thực tế

Mô hình mạch khi triển khai thực tế sẽ được kết nối bởi các dây nối với chân linh kiện như sau:



Hình 3.7 Hình ảnh về Các kết nối chân linh kiện

## CHƯƠNG 4 - TRIỂN KHAI MẠCH THỰC TẾ

### 4.1 Lập trình lệnh cho mạch thực tế

#### 4.1.1 Khai báo các thư viện sử dụng

```
#include <DHT.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <LiquidCrystal_I2C.h>
```

Đầu tiên, khai báo một số thư viện phụ trợ giúp hỗ trợ cho việc lập trình mạch thực tế.

#### 4.1.2 Khai báo các biến

```
int PIRPIN = 14;
int LEDPIN_RED = 2;
int LEDPIN_GREEN = 4;
int BUZZERPIN = 27;
int DHTPIN = 13;
int DHTTYPE = DHT11;
char wifiName[] = "Asus";
char wifiPass[] = "87654321";
char server[] = "api.thingspeak.com";
String THINGSPEAK_APIKEY = "18425AZU8BBP3R98";
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2);
WiFiClient client;
```

Khai báo các chân PIN của cảm biến PIR là 14, đèn LED đỏ là 2, đèn LED xanh là 4, còi BUZZER là 27, cảm biến DHT là 13.

Biến wifiName và wifiPass dùng cho truy cập wifi. Server mà phía client sẽ kết nối đến là “api.thingspeak.com” bởi biến server. Biến THINGSPEAK\_APIKEY dùng để xác định chính xác kênh nào được tạo trên dịch vụ ThingSpeak.

Khai báo DHT dht(DHTPIN, DHTTYPE) được hiểu là khai báo chân kết nối DHTPIN và loại DHTTYPE của cảm biến DHT-11.

Khai báo LiquidCrystal\_I2C lcd(0x27,16,2) được hiểu là khai báo địa chỉ 0x27 cho I2C và màn hình LCD có khả năng hiển thị 16x2.

Khai báo WiFiClient client được hiểu là phía client sẽ có kết nối đến server đã khai báo ở trên.

### 4.1.3 Khởi tạo các hàm con

#### 4.1.3.1 Kết nối với internet

```
void connectToWifi(){
    Serial.print("Connecting");
    WiFi.begin(wifiName, wifiPass);
    while(WiFi.status() != WL_CONNECTED){
        Serial.print(".");
        delay(500);
    }
    Serial.println("\r\nWiFi connected");
}
```

Vi điều khiển ESP32 DevKitC sẽ kết nối với Wifi thông qua hàm connectToWifi() với wifiName là tên wifi và wifiPass là mật khẩu wifi. Nếu trạng thái của wifi vẫn chưa kết nối được thì hiển ra màn hình Serial dấu chấm mang ý nghĩa là thiết bị đang cố kết nối cho người dùng biết đồng thời chờ 0.5s để kiểm tra lại kết nối. Sau khi kết nối thành công thì sẽ hiển thị ra màn hình Serial nội dung đã kết nối thành công.

#### 4.1.3.2 **Bật tắt còi và đèn khi phát hiện chuyển động**

```
void playBuzzer(int statusPIR){
    if(statusPIR == 1){
        digitalWrite(LEDPIN_GREEN, LOW);
        digitalWrite(LEDPIN_RED, HIGH);
        digitalWrite(BUZZERPIN, HIGH);
        delay(3000);
    }
    else{
        digitalWrite(LEDPIN_GREEN, HIGH);
        digitalWrite(LEDPIN_RED, LOW);
        digitalWrite(BUZZERPIN, LOW);
    }
}
```

Tham số nhận vào sẽ là 1 và 0, mang ý nghĩa là trạng thái của cảm biến PIR lúc bấy giờ. Nếu phát hiện chuyển động nghĩa là tham số nhận vào bằng 1 thì tắt đèn LED xanh, bật đèn LED đỏ, bật còi BUZZER và trì hoãn 3s. Ngược lại, nếu không phát hiện chuyển động nào nữa thì trả về trạng thái bình thường như ban đầu bằng cách bật đèn LED xanh trở lại, tắt đèn LED đỏ, tắt còi BUZZER.

#### 4.1.3.3 **Hiển thị nhiệt độ và độ ẩm ra màn hình LCD**

```
void printToLCD(float t, float h){
    lcd.setCursor(0,0);
    lcd.print("Nhiệt độ: ");
    lcd.setCursor(0,1);
    lcd.print("Độ ẩm: ");
```



```

        lcd.setCursor(10,0);
        lcd.print(String(t, 2).c_str());
        lcd.print(" ");
        lcd.setCursor(10,1);
        lcd.print(String(h, 2).c_str());
    }

```

Nhiệt độ, độ ẩm sẽ được hiển thị ra màn hình Serial. Đồng thời, ở 2 hàng của màn LCD 1602 cũng được hiển thị thông tin tương ứng bởi hàm `setCursor()` sẽ chỉ định vị trí hàng hiển thị lên màn hình LCD.

#### 4.1.3.4 Đưa dữ liệu lên dịch vụ *ThingSpeak*

```

void printToThingSpeak(float t,float h){
    String body = "field1=" + String(t, 1) + "&field2=" + String(h, 1);
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY:"+THINGSPEAK_APIKEY+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(body.length());
    client.print("\n\n");
    client.print(body);
    client.print("\n\n");
}

```

Tại dịch vụ ThingSpeak ta có 2 trường dùng để trực quan hóa dữ liệu nhiệt độ và độ ẩm lần lượt là field1 và field2. Phía client sử dụng method POST để gửi dữ liệu thông qua giao thức HTTP với server là api.thingspeak.com. Định danh kênh ThingSpeak thông qua biến khởi tạo ban đầu THINGSPEAK\_APIKEY. Đối với application/x-www-form-urlencoded là kiểu mã hóa mặc định nếu thuộc tính enctype không có giá trị.

#### 4.1.4 Khởi chạy hệ thống

```
void setup(){
    Serial.begin(9600);
    dht.begin();
    lcd.init();
    lcd.backlight();
    pinMode(LEDPIN_RED, OUTPUT);
    pinMode(LEDPIN_GREEN, OUTPUT);
    pinMode(PIRPIN, INPUT);
    pinMode(BUZZERPIN, OUTPUT);
    connectToWifi();
}
```

Khởi chạy hay khởi động các thành phần là những gì ta cần đặt trong hàm setup(). Thiết lập Serial với tốc độ truyền dữ liệu là 9600 bit/s.

Đối với đèn LED và còi BUZZER ta sẽ khai báo pinMode cho chúng là OUTPUT.

Đối với cảm biến chuyển động PIR ta sẽ khai báo pinMode là INPUT. Hàm init() và backlight() sẽ lần lượt là khởi động màn hình và bật sáng đèn nền của LCD 1602.

Kết nối tới wifi bởi hàm connectToWifi().

#### 4.1.5 Vòng lặp chương trình

```

void loop(){
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    int statusPIR = digitalRead(PIRPIN);
    if(isnan(t) || isnan(h)) {}
    else{printToLCD(t,h);}
    if(statusPIR == HIGH){
        playBuzzer(1);
        Serial.println("Phat hien chuyen dong!");
    }
    else{
        playBuzzer(0);
        Serial.println("Khong phat hien chuyen dong!");
    }
    delay(1000);
    if (client.connect(server, 80)) {
        printToThingSpeak(t,h);
        Serial.printf("Nhiet do %s - Do am %s\r\n", String(t, 2).c_str(),
            String(h, 2).c_str());
    }
    client.stop();
}

```

Tạo 2 biến *t* và *h* để gán giá trị đọc được từ cảm biến DHT-11. Với *t* là giá trị nhiệt độ, *h* là giá trị độ ẩm. Biến *statusPIR* để gán trạng thái chuyển động.

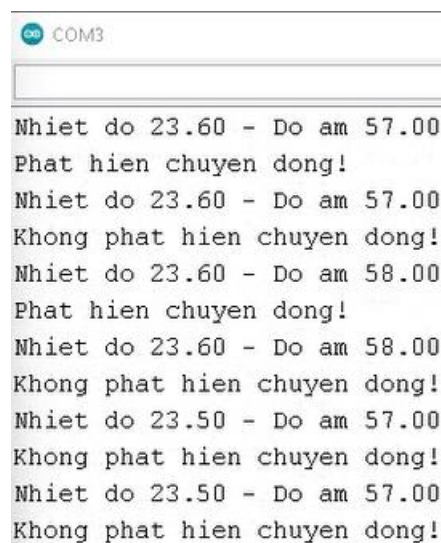
Cảm biến DHT-11 sẽ đôi khi không thu được thông số nên cần phải có hàm kiểm tra xem có lấy thông số được hay không bằng hàm `isnan()`. Nếu không phải số thì không làm gì cả, ngược lại thì hiển thị ra màn hình LCD bằng hàm `con printToLCD()` đã tạo trước đó.

Cảm biến PIR SR-501 quét quanh khu vực để xác định các chuyển động. Do đó, một hàm kiểm tra tiếp theo dùng để đọc từ cảm biến chuyển động PIR, nếu đọc được giá trị HIGH nghĩa là có chuyển động, thì `playBuzzer()` với tham số là 1, ngược lại thì tham số là 0.

Hàm kiểm tra cuối cùng chính là để xác định rằng client đã kết nối đến server qua cổng 80 thành công thì sẽ đưa dữ liệu lên ThingSpeak bằng hàm `con printToThingSpeak()` và in ra màn hình Serial thông số nhiệt độ, độ ẩm đã làm tròn đến 2 chữ số thập phân.

## 4.2 Kết quả đạt được

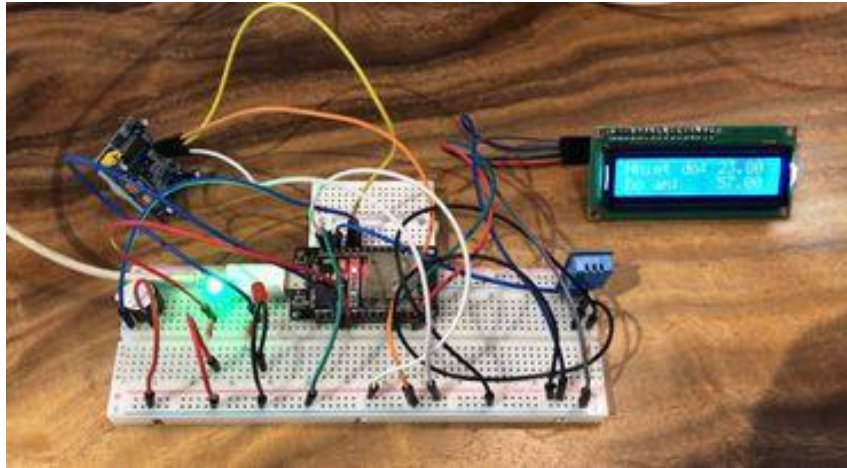
### 4.2.1 Kết quả trên Màn hình Serial



```
COM3
Nhiet do 23.60 - Do am 57.00
Phat hien chuyen dong!
Nhiet do 23.60 - Do am 57.00
Khong phat hien chuyen dong!
Nhiet do 23.60 - Do am 58.00
Phat hien chuyen dong!
Nhiet do 23.60 - Do am 58.00
Khong phat hien chuyen dong!
Nhiet do 23.50 - Do am 57.00
Khong phat hien chuyen dong!
Nhiet do 23.50 - Do am 57.00
Khong phat hien chuyen dong!
```

Hình 4.1 Hình ảnh về Các thông tin thu thập được trên màn hình Serial

#### 4.2.2 Kết quả trên Mạch thực tế



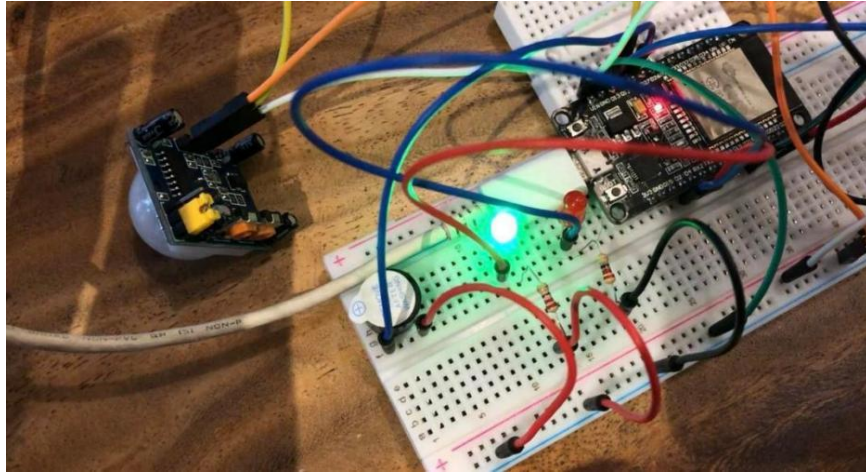
Hình 4.2 Hình ảnh về Tổng quan sản phẩm mạch thực tế



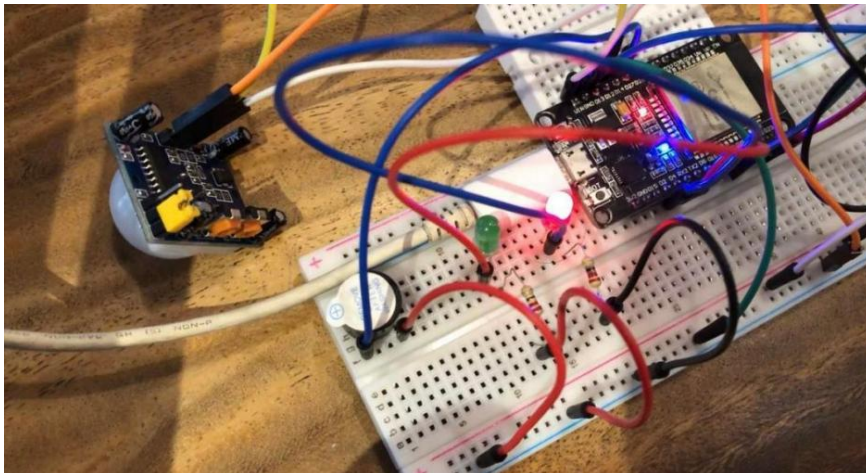
Hình 4.3 Hình ảnh về Số liệu nhiệt độ và độ ẩm hiển thị trên màn hình LCD



Hình 4.4 Hình ảnh về Sự thay đổi số liệu khi có tác động bởi môi trường bên ngoài



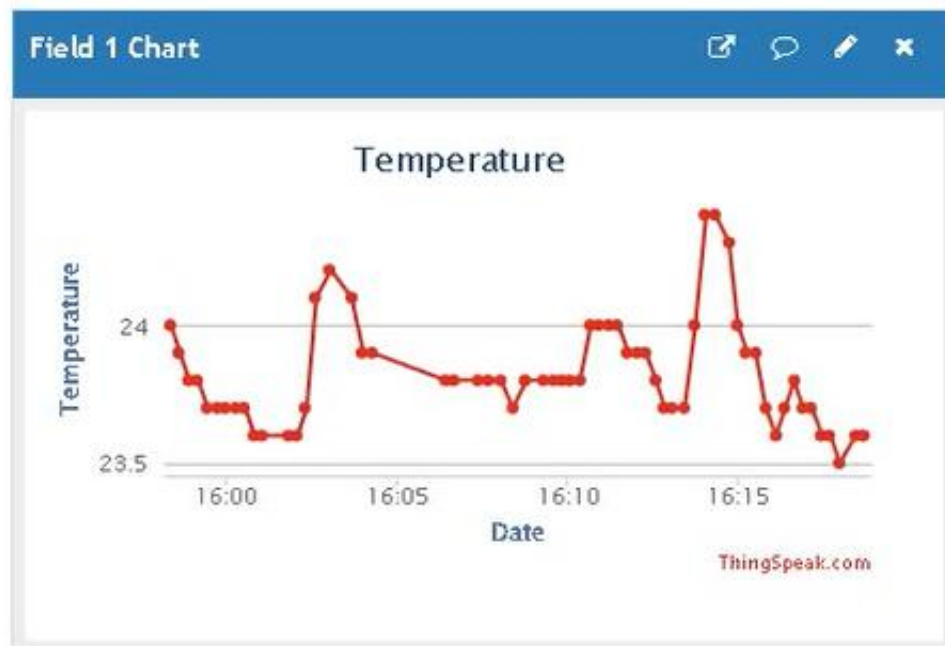
Hình 4.5 Hình ảnh về Trạng thái khi không phát hiện chuyển động



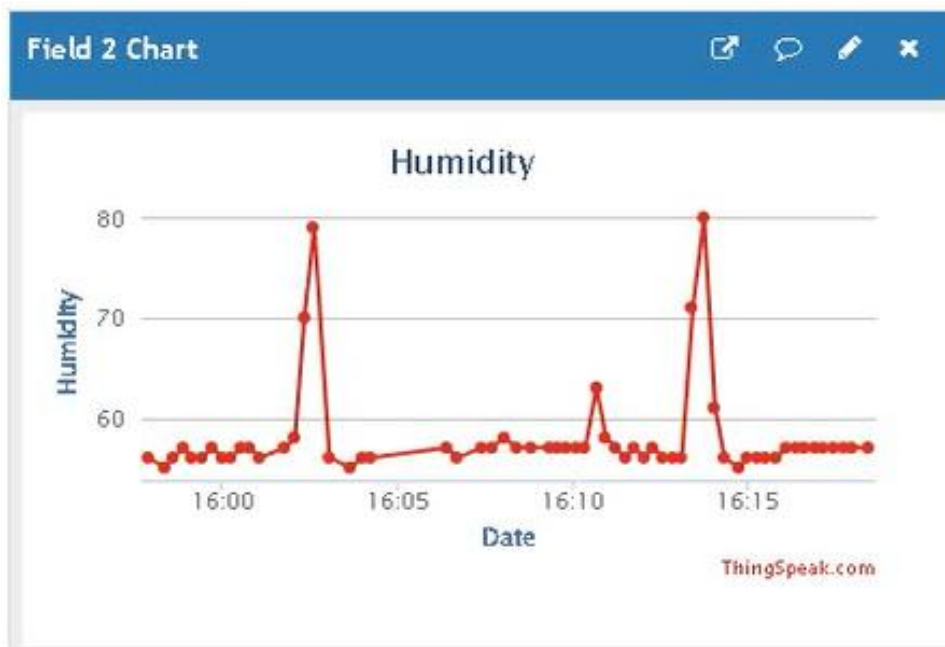
Hình 4.6 Hình ảnh về Trạng thái khi có phát hiện chuyển động



### 4.2.3 Kết quả trên ThingSpeak



Hình 4.7 Hình ảnh về Trực quan hóa dữ liệu Nhiệt độ trên ThingSpeak



Hình 4.8 Hình ảnh về Trực quan hóa dữ liệu Độ ẩm trên ThingSpeak

## CHƯƠNG 5 – TỔNG KẾT

### 5.1 Nhận xét kết quả

Đo được nhiệt độ, độ ẩm và hiển thị nhiệt độ, độ ẩm ra màn hình.

Phát hiện được các chuyển động và báo động bằng còi và đèn led tương ứng.

Kết nối Internet, đưa dữ liệu thu thập được lên dịch vụ ThingSpeak để xem dữ liệu trực quan hóa hơn.

### 5.2 Thuận lợi và khó khăn

Về Thuận lợi:

- + Phần cứng: Khi khởi động, các thiết bị hoạt động tốt không bị hư hỏng.
- + Phần mềm: Các logic hoạt động đúng với ý tưởng đã đề ra.

Về Khó khăn:

- + Gặp một ít khó khăn khi triển khai mạch thực tế do nhóm chưa có nhiều kiến thức về lập trình mạch nhưng qua tìm hiểu từ nhiều nguồn cuối cùng cũng đã hoàn thiện cơ bản.
- + Cảm biến chuyển động có độ nhạy chưa tốt khi xác định chuyển động, cụ thể là đôi khi phát hiện có chuyển động chậm hơn thực tế nếu khoảng cách từ 2m trở lên.
- + Dịch vụ ThingSpeak nhận dữ liệu chậm và đôi khi không nhận được dữ liệu mà hệ thống đưa lên.

### 5.3 Định hướng phát triển

Các linh kiện thô sơ mà nhóm lựa chọn sẽ là tiền đề cho các linh kiện cao cấp hơn khi sử dụng cho các thiết bị hệ thống cao cấp, giúp cho hệ thống chạy ổn định hơn. Có thể thêm một số thứ khác bổ sung như giúp điều khiển tắt mở hệ thống theo dõi thông tin môi trường này từ xa hoặc thêm một hộp bao bọc lại để bảo vệ, chống nắng mưa,...



## PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ

Họ và tên	MSSV	Các công việc cần thực hiện	Đánh giá
Tô Vĩnh Khang (Nhóm trưởng)	51800408	<ul style="list-style-type: none"> <li>-Thảo luận nhóm, lên ý tưởng đề tài.</li> <li>-Quản lý dự án và Phân công công việc cho từng thành viên nhóm.</li> <li>-Tìm hiểu thông tin về vi điều khiển ESP32 DevKitC, cảm biến Chuyển động PIR HC-SR501 và cảm biến Nhiệt độ, độ ẩm DHT-11.</li> <li>-Mô phỏng các hoạt động chính của hệ thống trên công cụ Cisco Packet Tracer.</li> <li>-Thiết kế Chi tiết các kết nối của linh kiện trên công cụ Circuito.io và Lập trình mạch thực tế.</li> <li>-Trình bày nội dung Báo cáo Chương 1, Chương 2 (2.1.1-2.1.4), Chương 3, Chương 4 (4.1.4, 4.1.5), Chương 5.</li> <li>-Đánh số mục lục file báo cáo, nộp bài báo cáo cuối kỳ theo từng đợt đúng hạn (Trước các ngày 26/10, 9/11, 23/11).</li> </ul>	100%

Đỗ Vũ Quốc Tùng	51800323	<ul style="list-style-type: none"> <li>-Thảo luận nhóm, lên ý tưởng đề tài.</li> <li>-Tìm hiểu thông tin về Màn hình LCD 1602 Tích hợp chuyển đổi I2C và Đèn LED.</li> <li>-Tìm hiểu thông tin về dịch vụ ThingSpeak.</li> <li>-Tìm mua các thiết bị điện tử, lắp đặt dựa trên mạch thiết kế đã làm trên công cụ Circuito.io và nạp code bằng ArduinoIDE.</li> <li>-Quay video demo chạy thiết bị.</li> <li>-Trình bày nội dung Báo cáo Chương 2 (2.1), Chương 4 (4.1.1 - 4.1.3).</li> <li>-Nộp các phần được giao trước 1 ngày đến hạn của từng đợt báo cáo (Trước các ngày 25/10, 8/11, 22/11) cho nhóm trưởng.</li> </ul>	100%
Lâm Văn Bảo	51702063	<ul style="list-style-type: none"> <li>-Thảo luận nhóm, lên ý tưởng đề tài.</li> <li>-Tìm hiểu thông tin về Điện trở cảm, còi Buzzer 5V và Bóng bán dẫn NPN BC337.</li> <li>-Tạo tài khoản và đưa dữ liệu lên dịch vụ ThingSpeak.</li> <li>-Edit video demo.</li> <li>-Trình bày nội dung Báo cáo Chương 2 (2.1.5 - 2.1.8), Chương 4 (4.2).</li> <li>-Nộp các phần được giao trước 1 ngày đến hạn của từng đợt báo cáo (Trước các ngày 25/10, 8/11, 22/11) cho nhóm trưởng.</li> </ul>	100%

Bảng phân công công việc và đánh giá

## **TÀI LIỆU THAM KHẢO**

### **Tài liệu Tiếng Anh**

- [1] Cisco, Supplemental Material of IOT Fundamentals, Ton Duc Thang University.
- [2] Electronic Component, [https://en.wikipedia.org/wiki/Electronic\\_component](https://en.wikipedia.org/wiki/Electronic_component)
- [3] ESP32, <https://esp32.com/>
- [4] Packet Tracer, <https://www.netacad.com/courses/packet-tracer/>
- [5] Circuito, <https://www.circuito.io/>
- [6] Arduino IDE, <https://www.arduino.cc/en/Guide/Environment>
- [7] ThingSpeak, <https://thingspeak.com/>
- [8] Espressif, <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>

### **Tài liệu Tiếng Việt**

- [9] Khue Nguyen, Lập trình ESP32 từ A tới Z, <https://khuenguyencreator.com/lap-trinh-esp32-tu-a-toi-z/>
- [10] Dan Nguyen, Cách dùng cảm biến chuyển động, <http://arduino.vn/bai-viet/522-pir-sensor-alarm>
- [11] Share4Happy, Đọc cảm biến nhiệt độ, độ ẩm DHT11 với ESP32 sử dụng Arduino IDE, <https://www.youtube.com/watch?v=0iQOJoB432o>