ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ

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

BÁO CÁO ĐỒ ÁN MÔN HỌC

MÔ HÌNH NHÀ KÍNH THÔNG MINH

GVHD: Thầy Bùi Quốc Bảo

SVTH:

Nguyễn Thanh Tâm – 1613058

Nguyễn Quốc Bảo – 1610186

Đỗ Thanh Tân – 1613077

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2019

LỜI CẢM ƠN

Để thực hiện đồ án này, chúng em xin chân thành cảm ơn các thầy cô giáo trong Trường ĐHBK Tp.HCM nói chung và các thầy cô trong khoa Điện – Điện tử nói riêng đã truyền đạt cho chúng em các kiến thức đại cương và chuyên ngành, giúp chúng em có được kiến thức vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Đặc biệt chúng em xin gửi lời cảm ơn chân thành nhất tới thầy Bùi Quốc Bảo đã quan tâm giúp đỡ và tận tình hướng dẫn chúng em hoàn thành đồ án này trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án này không thể tránh khỏi những thiếu sót. Nhóm em rất mong nhận được sự chỉ bảo và đóng góp ý kiến của thầy cô và các bạn giúp nhóm em bổ sung kiến thức để phục vụ tốt hơn công tác thực tế sau này.

Chúng em chân thành cảm ơn.

Tp. Hồ Chí Minh, ngày 3 tháng 6 năm 2019

TÓM TẮT NỘI DUNG BÁO CÁO

Phần 1. Giới thiệu đề tài

Trong phần này, nhóm em trình bày tổng quan về đề tài như:

- Giới thiệu đề tài.
- Mục tiêu đề tài.
- Nhiệm vụ đặt ra.
- Phân chia công việc nhóm.

Phần 2. Lý thuyết

Trình bày kiến thức tổng quan về NodeMCU ESP8266, các thiết bị ngoại vi sử dụng như LCD, cảm biến nhiệt độ độ ẩm, cảm biến ánh sáng, cảm biến độ ẩm đất, nguồn cung cấp.

Phần 3. Thiết kế và thực hiện phần cứng

Nội dung phần này bao gồm:

- Yêu cầu hệ thống.
- Đặc tả hệ thống.
- Thực hiện và thiết kế phần cứng.
- Sơ đồ Schematics từng khối.

Phần 4. Thiết kế phần mềm

- Lưu đồ giải thuật.
- Phần mềm sử dụng

Phần 5. Thiết kế web server

Phần 6. Sản phẩm hoàn thiện

Phần 7. Kết luận và hướng phát triển

- Kết luận.
- Hướng phát triển.

MỤC LỤC

DANH SÁCH HÌNH MINH HỌA

DANH SÁCH BẢNG SỐ LIỆU

1. GIỚI THIỆU

1.1 Tổng quan

Ngày nay, các hệ thống IoT được sử dụng ngày càng nhiều trong các lĩnh vực đời sống. Một trong các lĩnh vực đi tiên phong ứng dụng công nghệ này là nông nghiệp, với rất nhiều nhà kính thông minh được xây dựng, từ quy mô trang trại tới hộ gia đình, góp phần tiết kiệm tối đa các tài nguyên như ánh sáng, nước, phân bón, điện năng, công sức mà vẫn cho năng suất và hiệu quả cao nhất.

Đề tài Đồ án môn học: " Thiết kế mô hình nhà kính thông minh " của nhóm em, được thực hiện với mong muốn nâng cao kiến thức của bản thân cũng như mong muốn nâng cao giá trị của các sản phẩm công nghệ và cuộc sống con người.

Trong đề tài này chúng em sử dụng các cảm biến ánh sáng, nhiệt độ, độ ẩm đất giao tiếp với ESP8266 NodeMCU. Tự động hóa quản lí các thông số môi trường sinh trưởng của cây trong 1 nhà kính, hiển thị lên 1 màn hình LCD, đồng thời gửi các thông số dữ liệu lên web server để có thể phân tích và điều khiển từ xa.

1.2 Nhiệm vụ đề tài

Nội dung 1: Tìm hiểu về vi điều khiển NoteMCU8266.

Nội dung 2: Tìm hiểu về các module: cảm biến nhiệt độ độ ẩm không khí DHT22, cảm biến ánh sáng quang trở, cảm biến độ ẩm đất.

Nội dung 3: Xây dựng giải thuật để lập trình cho vi điều khiển đọc dữ liệu từ các cảm biến và hiện thị lên LCD, đồng thời gửi dữ liệu lên web server.

Nội dung 4: Xây dựng web server có chức năng nhận và phân tích dữ liệu gửi từ vi điều khiển.

1.3 Phân chia công việc trong nhóm

TEAM CONTRACT	
Thành viên	
Nguyễn Thanh Tâm	
Đỗ Thanh Tân	
Nguyễn Quốc Bảo	
Công việc	Thành viên phụ trách
a. Develop system design	Nguyễn Thanh Tâm
b. Design hardware	Đỗ Thanh Tân
c. Design software	Nguyễn Quốc Bảo Đỗ Thanh Tân
d. Design websever	Nguyễn Thanh Tâm Nguyễn Quốc Bảo
e. Contruction and test	All
Họp nhóm	Cuối tuần (thứ 7 hoặc chủ nhật)
Quy định	- Hoàn thành công việc được giao - Tôn trọng ý kiến cá nhân trong nhóm

Bảng 1: Bảng phân chia công việc nhóm.

	Week					
	3-4	5-8	9-10	11-13	14-15	16-19
1.System requirement	x					
2.System specification	x					
3.Design hardware part		x				
3.1 Schemactic		x				
3.2 PCB			x			
3.3 Làm mô hình nhà kính				x	x	
4.Develop software part						
4.1.Viết giải thuật			x			
4.2. Viết code			x	x		
5.Develop websever					x	x
6. Hoàn thiện và test						x

Bảng 2: Bảng theo dõi tiến trình thực hiện sản phẩm.

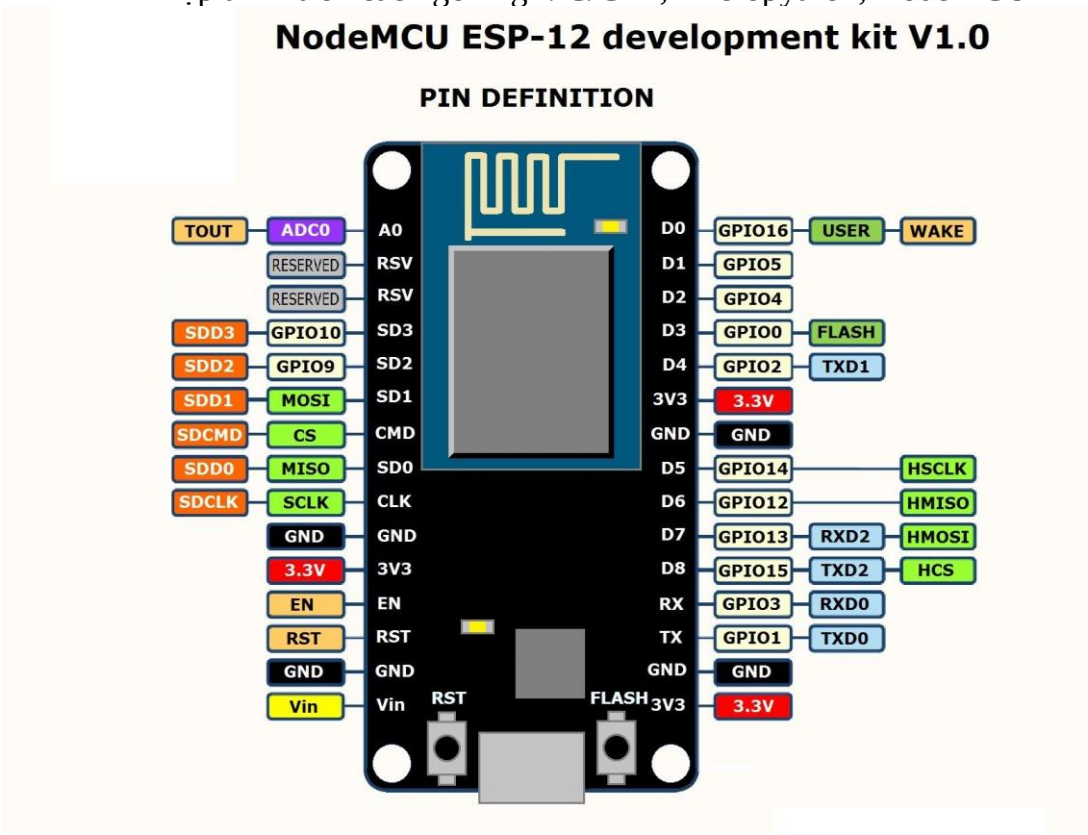
2. LÝ THUYẾT

2.1 MCU: ESP8266 NodeMCU

NodeMCU được phát triển dựa trên Chip WiFi ESP8266EX bên trong Module ESP-12E dễ dàng kết nối WiFi với một vài thao tác. Board còn tích hợp IC CP2102, giúp dễ dàng giao tiếp với máy tính thông qua Micro USB để thao tác với board. Và có sẵn nút nhấn, led để tiện qua quá trình học, nghiên cứu.

Thông số kỹ thuật:

Chip: ESP8266EX
WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
Điện áp hoạt động: 3.3V
Số chân I/O: 11
Số chân Analog Input: 1
Bộ nhớ Flash: 4MB
Giao tiếp: Cable Micro USB
Hỗ trợ bảo mật: WPA/WPA2
Tích hợp giao thức TCP/IP
Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU

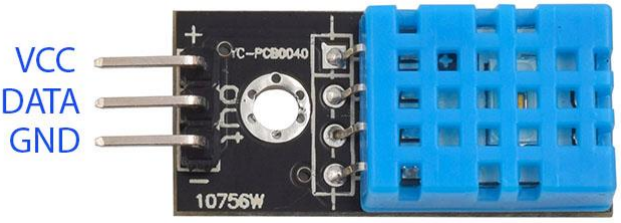


Hình 2.1: Sơ đồ ra chân module ESP8266 NodeMCU

ESP8266 là một vi điều khiển có các module ngoại vi thông dụng như GPIO, SPI, I2C, onewire,… Điểm đặc biệt là ESP8266 có tích hợp thêm module wifi giúp nó có thể dễ dàng kết nối internet. Thế nên ESP8266 rất thích hợp cho các dự án nhúng và IOT cũng như trong đề tài này. Ngoài ra, do tính phổ biến và giá rẻ, ESP8266 có được đông đảo cộng đồng hỗ trợ.

2.2 Cảm biến

a. DHT22:



Hình 2.2a: Cảm biến DHT11

- Cảm biến số nhiệt độ và độ ẩm DHT22 với độ ổn định cao, có khả năng hoạt động liên tục trong thời gian dài.
- Thông số kỹ thuật:
    + Điện áp hoạt động: 5VDC
    + Dãi độ ẩm hoạt động: 0% - 100% RH, sai số ±2%RH
    + Dãi nhiệt độ hoạt động: -40°C ~ 80°C, sai số ±0.5°C
    + Giao tiếp: onewire.

b. Cảm biến độ ẩm đất:

Hình 2.2b: Cảm biến độ ẩm đất

- Cảm biến độ ẩm đất V1.2 có thiết kế đơn giản, dễ sử dụng, ngõ ra Analog, thích hợp với các dự án tự động, vườn thông minh.
- Thông số kỹ thuật:
    + Điện áp hoạt động: 3.3~5.5VDC.
    + Điện cực phủ sơn chống ăn mòn cho độ bền và độ ổn định cao.
    + Điện áp xuất ra chân Analog: 0~3VDC.

c. Cảm biến ánh sáng quang trở:

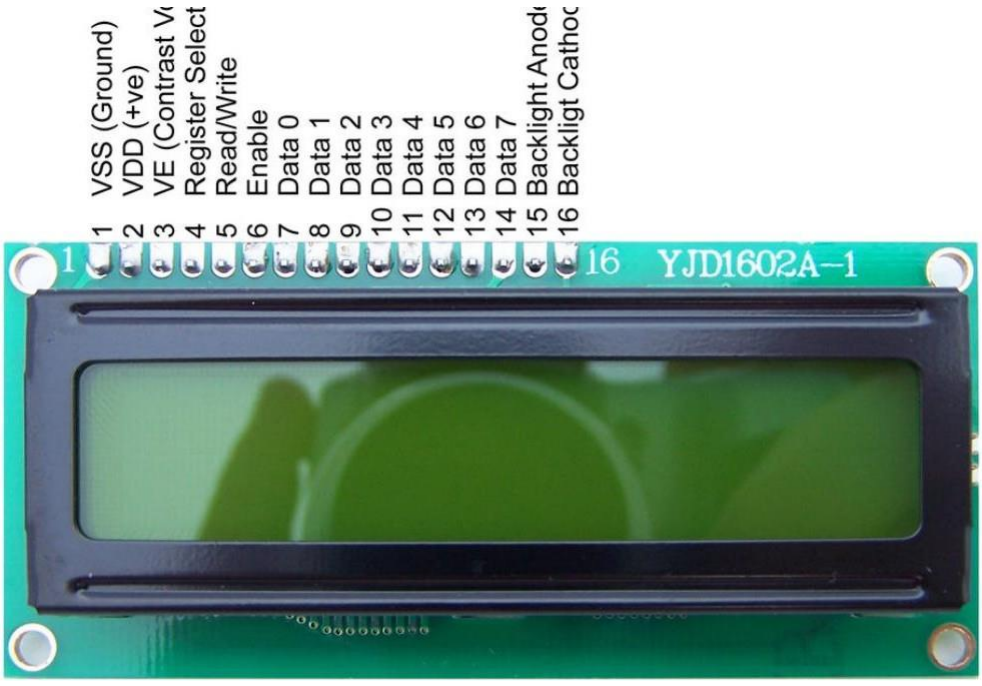
Hình 2.2c: Cảm biến ánh sáng

- Cảm biến ánh sáng quang trở phát hiện cường độ ánh sáng.
- Thông số kỹ thuật:
  + Điện áp làm việc: 3.3 ~ 5VDC.
  + Output: Digital.
  + Có thể điều chỉnh cường độ ánh sáng phát hiện bằng biến trở gắn trên cảm biến.

2. Modul Relay:

- Module 2 Relay Kích H/L (12VDC).
- Dòng tiêu thụ: khoảng 200mA/1Relay.

3. Nguồn:
   a. Nguồn tổ ong 12VDC – 5A.
   b. Mạch nguồn Switching LM2596 – 5V.
   c. Mạch nguồn tuyến tính LDO AMS1117 3V3 – 1A.
4. LCD 16x2



Hình 2.2d: Màn hình LCD 16x2

3. YÊU CẦU HỆ THỐNG

Name: Mô hình nhà kính thông minh.

Purpose: Hệ thống nhà kính thông minh kiểm soát độ ẩm, nhiệt độ, ánh sáng thông qua web server theo thời gian thực.

Input:

- Cảm biến ánh sáng, nhiệt độ, độ ẩm không khí, độ ẩm đất.
- Switch 3 tiếp điểm: dùng để xác lập chế độ điều khiển tự động hay điều khiển bằng tay.
- Nút nhấn : điều khiển trực tiếp bơm và đèn.

Output:

- Relay điều khiển máy bơm, bóng đèn, đèn cấp nhiệt.
- Thông tin dữ liệu cần được hiển thị lên sever:
  + Các thông số cảm biến.
  + Bơm được bật.
  + Đèn được bật.

Use case:

Chế độ tự động:

- ***Brief description***:

Tự động cập nhật các thông số về nhiệt độ, ánh sáng, độ ẩm, lên sever. Tự động điều khiển đèn, đèn sưởi, máy bơm dựa trên các thông số của cảm biến.

- ***Basic flow***
  + Switch-mode ở chế độ tự động.
  + Cập nhật dữ liệu nhiệt độ, độ ẩm, ánh sáng mỗi 5-10 phút một lần. Xử lý dữ liệu và gửi lên sever.
  + Cảm biến độ ẩm đất :

    Ướt: trên 88% là mức cao, không cần bơm thêm nước.

    Bình thường: từ 66 – 88% là mức hoạt động bình thường, ổn định.

    Khô : Dưới 66%, tự động điều khiển relay bơm nước vào đất.
  + Nhiệt độ môi trường:

    Lạnh: Dưới 15 độ C, cần bật đèn sưởi.

    Bình thường: Từ 16 – 24 độ C.

Nóng: Từ 25 độ C trở lên, không được bật lò sưởi.

+ Ánh sáng:

Bình thường: >= 300Lux, không cần bật đèn.

Tối: Dưới 300Lux, cần bật đèn chiếu sáng.

Chế độ điều khiển bằng tay:

- ***Brief description:***

Ở chế độ này, vô hiệu hóa chức năng điều khiển tự động, ta tùy ý điều khiển trực tiếp máy bơm, đèn chiếu sáng, đèn sưởi qua các nút nhấn.

Vẫn tự động cập nhật dữ liệu lên sever.

- ***Basic flow:***
  + Chuyển switch-mode sang chế độ đk bằng tay.
  + Bộ xử lý vẫn đọc về dữ liệu và gửi lên sever.
  + Điều khiển trực tiếp máy bơm, đèn, đèn sưởi qua các nút nhấn.

Function:

- Mode: 2 chế độ làm việc: bằng tay hoặc tự động.
- Cập nhật và xử lý dữ liệu:
  + Đọc về dữ liệu từ các cảm biến 5 – 10 phút 1 lần, xử lý dữ liệu và gửi lên sever.
- Điều khiển các thiết bị:
  + Tự động: Điều khiển thiết bị dựa trên các dữ liệu đọc về.
  + Bằng tay: Điều khiển bằng nút nhấn trực tiếp.

Performance:

- Tự động reset khi hệ thống có vấn đề.
- Bộ điều khiển sử dụng nguồn hạ xuống từ điện 220VAC.
- Có pin dự phòng.
- Các thiết bị máy bơm, đèn chiếu sáng, đèn sưởi sử dụng điện lưới.
- Kết nối internet và server hoạt động ổn định.

Power:

- Sử dụng nguồn biến đổi từ điện lưới AC 220V.
- Có pin dự phòng trường hợp mất điện.

4. ĐẶC TẢ HỆ THỐNG

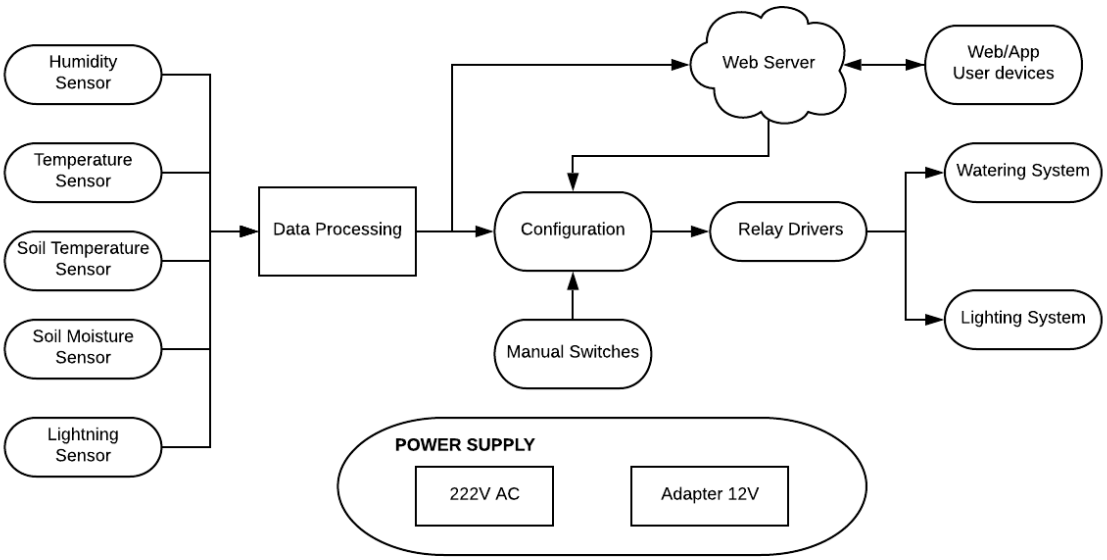
External Environment:

- Hệ thống nhà kính có thể triển khai ở các khu vực có khí hậu tương đối phù hợp cho sản phẩm nông nghiệp canh tác.
- Mô hình ngoài trời, xây dựng diện tích rộng.

System Connectivy:

- Hệ thống chạy bằng lưới điện 220VAC và Acqui dự phòng.
- Hệ thống kết nối với Internet thông qua mạng lưới wifi có sẵn.

System Block Diagram:



Hình 4.1: Sơ đồ khối hệ thống

- Sensor:
  + Purpose: Thu thập thông số môi trường theo thời gian thực.
  + Requirement: Sensor thu được nhiệt độ, độ ẩm, ánh sáng của môi trường và đất. Đảm bảo hoạt động ổn định trong thời gian dài.
- Data Processing:
  + Purpose: Xử lý thông số từ cảm biến và qui đổi theo hệ đơn vị chuẩn.
  + Reqirement: Độ chính xác tương đối, có lọc nhiễu nếu tín hiệu không ổn định.
- Web Server:
  + Purpose: Lấy dữ liệu từ nhà kính lên server và gởi tín hiệu điều khiển của con người xuống nhà kính.
  + Requirement: Ổn định, giá rẻ, bảo mật.
- Configuration:

    + Purpose: Tự động thực hiện các điều khiển từ các dữ liệu nhận được, bao gồm cảm biến và tín hiệu điều khiển từ con người.
    + Requirement: Có giải thuật điều khiển các thông số môi trường hợp lý.
- Relay for watering and lightning system:
    + Purpose: Thực hiện các hành động phần cứng để thay đổi thông số môi trường.
    + Requirement: Công suất hợp lý và ổn định.
- Power Supply:
    + Purpose: Cung cấp năng lượng cho toàn hệ thống
    + Requirement: Đủ công suất, hiệu suất chuyển đổi cao.

5. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

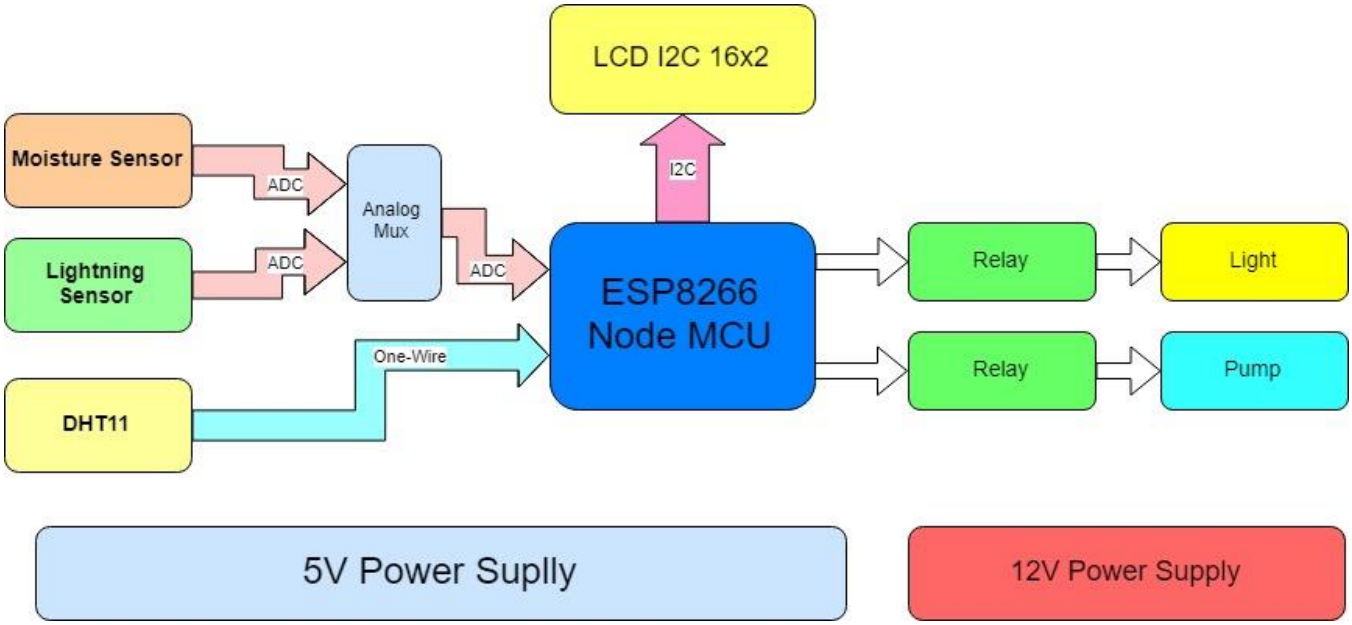
5.1 MÔ HÌNH NHÀ KÍNH

Nhóm thực hiện mô hình nhà kính đơn giản bằng cách ghép các tấm mica và cố định bằng silicon.

Hình 5.1: Mô hình sản phẩm

5.2 BOARD MẠCH

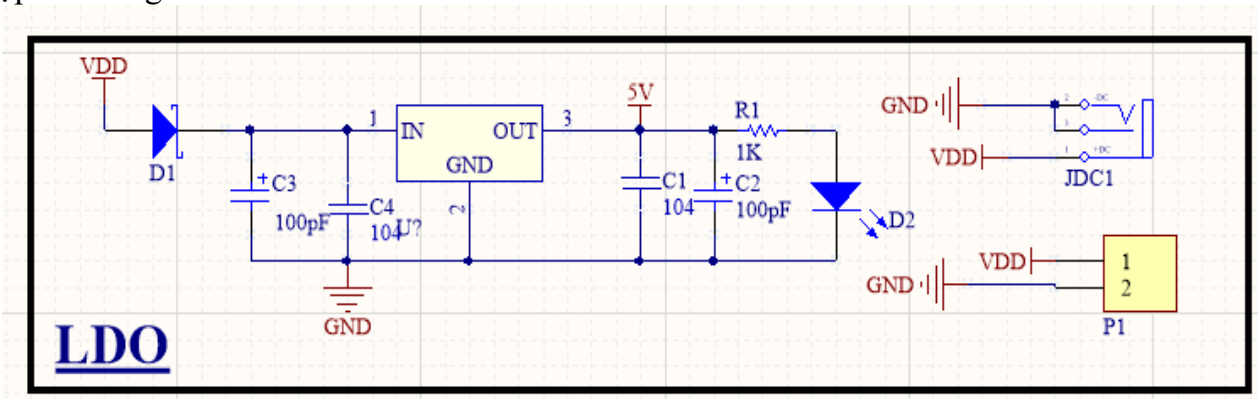
Sơ đồ khối của phần cứng:



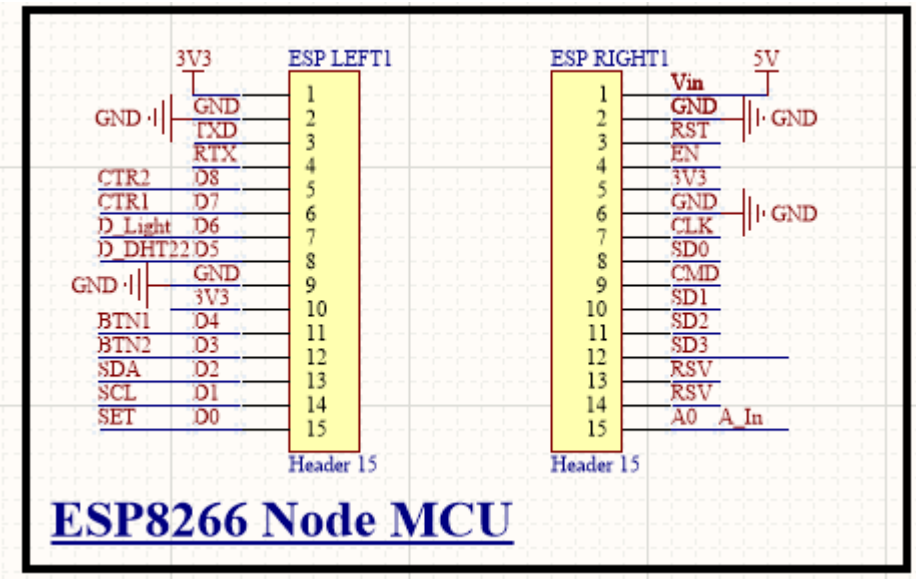
Hình 5.2a: Sơ đồ khối phần cứng

Các thành phần của board bao gồm khối nguồn cung cấp, module vi điều khiển và các cảm biến, LCD và Relay.

Phần nguồn được cung cấp từ Adapter (12V) sau đó được hiệu chỉnh thành các điện áp phù hợp cho từng module.

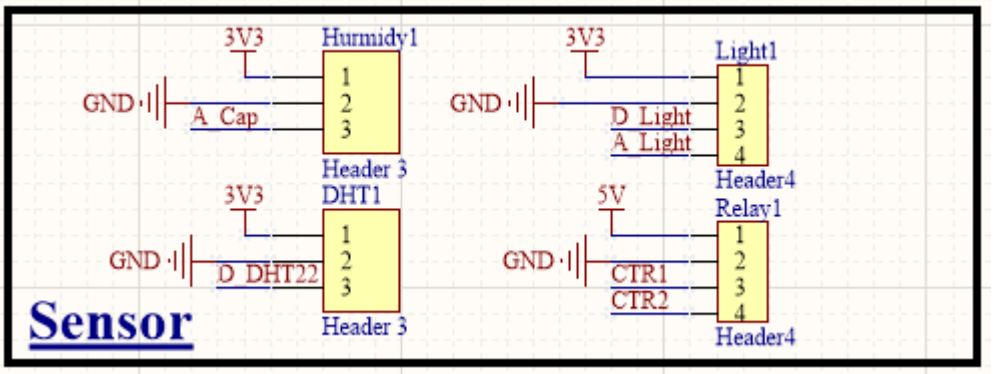


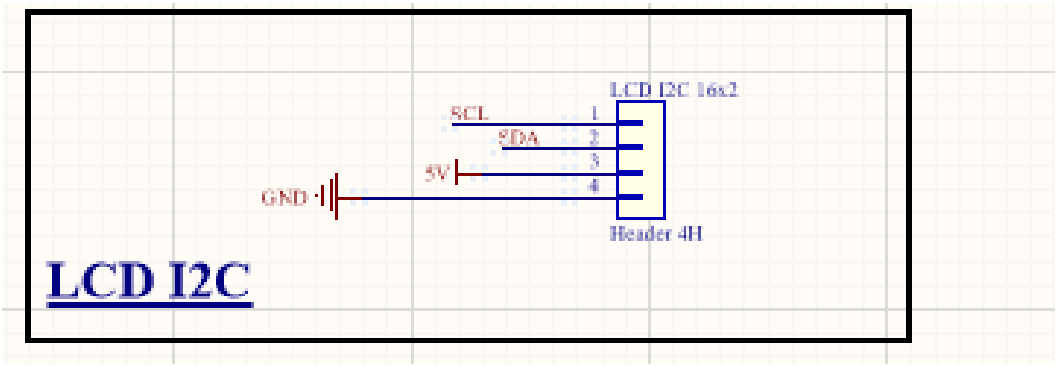
Hình 5.2b: Khối nguồn phần cứng



Hình 5.2c: Header ra chân cho ESP8266

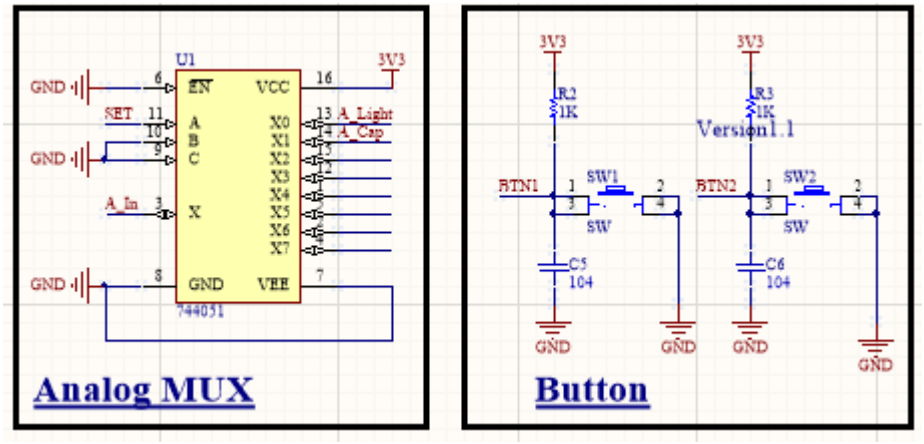
Do chủ yếu sử dụng module nên trên board có các header ra chân phù hợp với từng loại





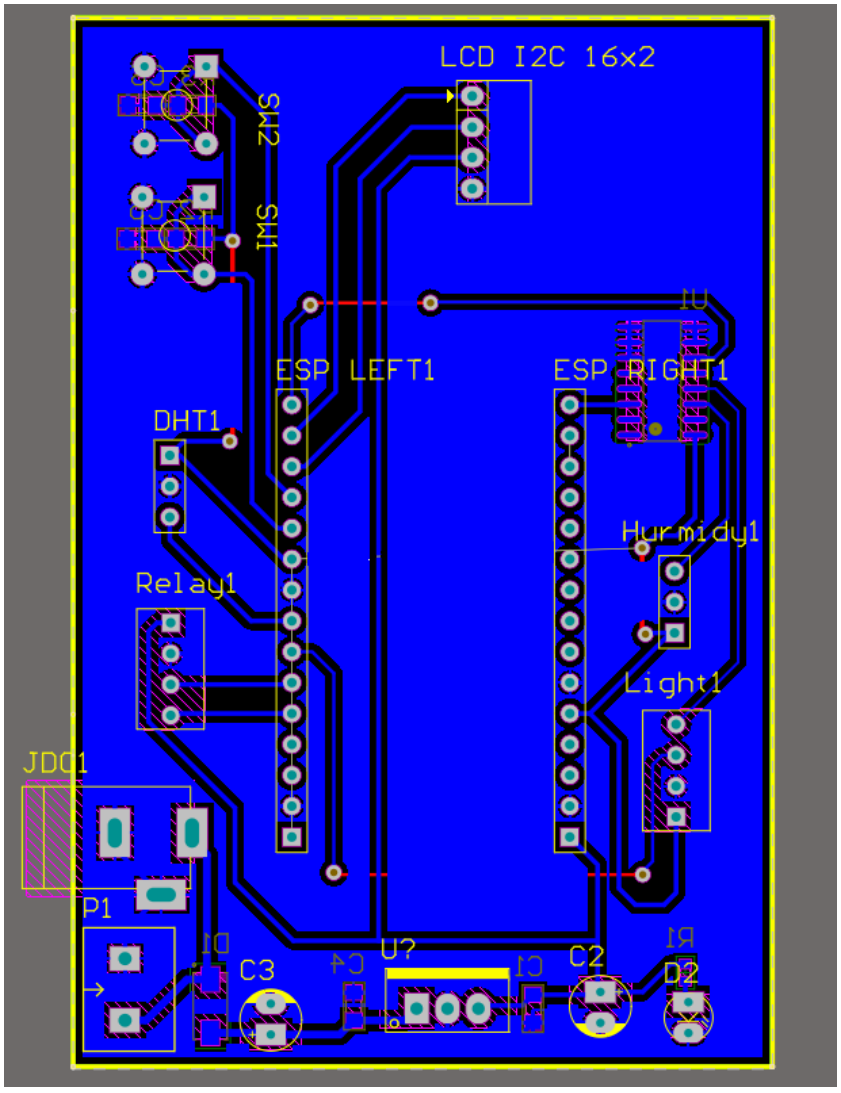
Hình 5.2d: Header ra chân cho LCD và các cảm biến

Do ESP8266 chỉ có 1 chân ADC in nên phải dùng IC mux tính hiệu analog (74HC4051) để có thể lái nhiều kênh ADC đọc vào 1 cách độc lập. Ngoài ra mạch còn có nút nhấn để hiệu chỉnh trạng thái của relay.

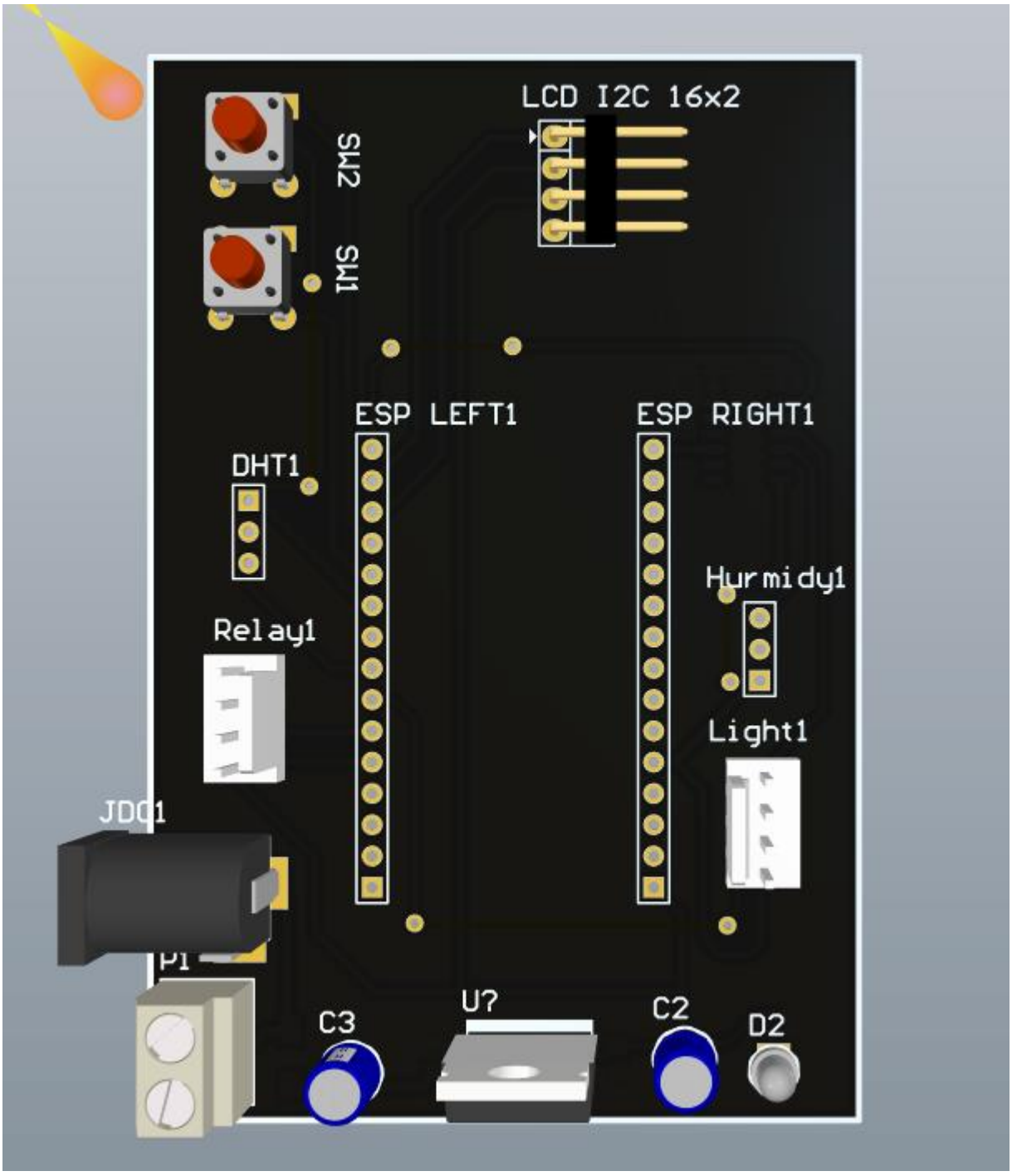


Hình 5.2e : Sơ đồ bộ dồn kênh Analog và nút nhấn

Hình ảnh sau khí thiết kế board mạch:



Hình 5.2f: Mạch in sau khi thiết kế

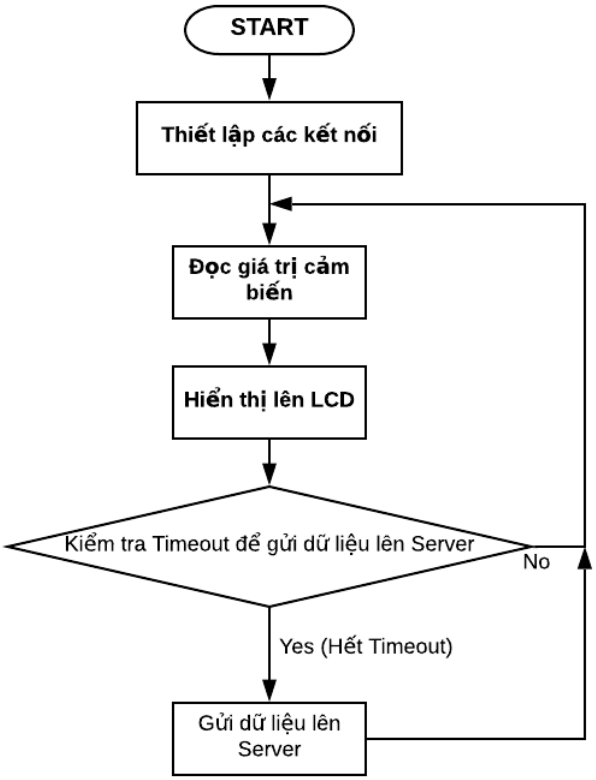


Hình 5.2g: Sơ đồ 3D PCB sau thiết kế

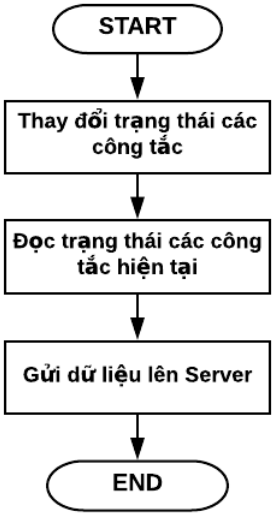
6. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

6.1 THIẾT KẾ CHƯƠNG TRÌNH NHÚNG CHO NODEMCU

Lưu đồ giải thuật chính:



Lưu đồ giải thuật chương trình ngắt khi người dùng sử dụng công tắc:



Lưu đồ giải thuật chương trình ngắt khi có tín hiệu điều khiển gửi đến:

6.2 THIẾT KẾ MQTT SERVER VÀ DATABASE

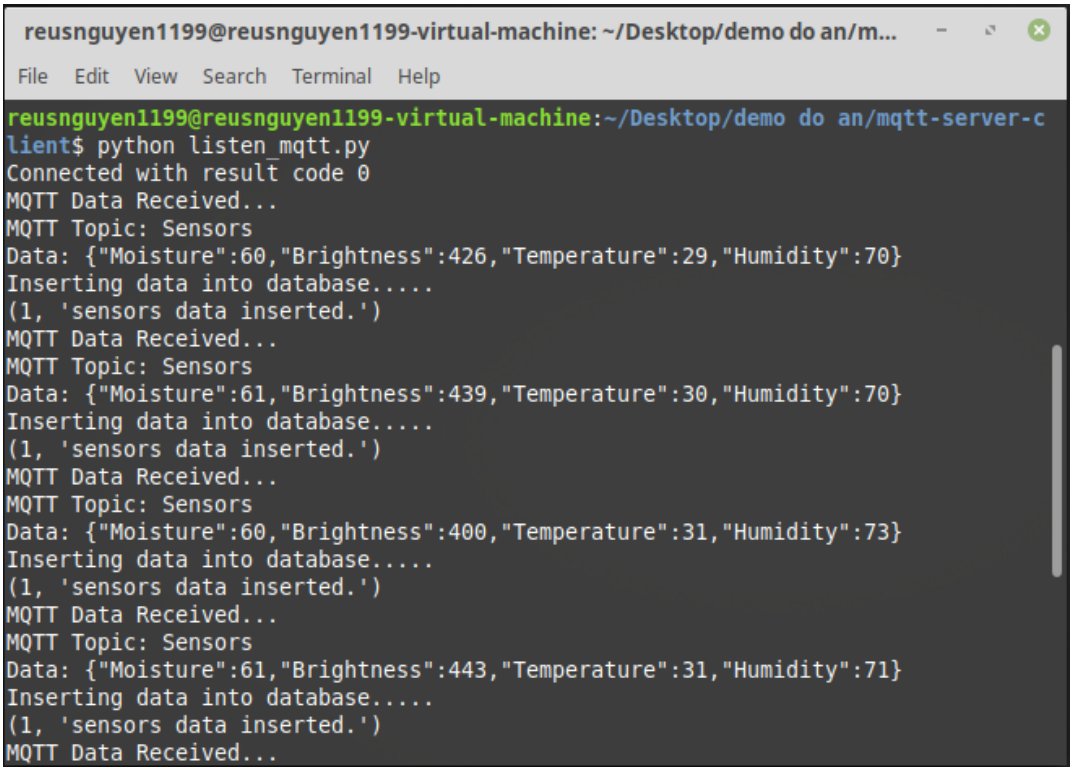
MQTT Server (Broker):

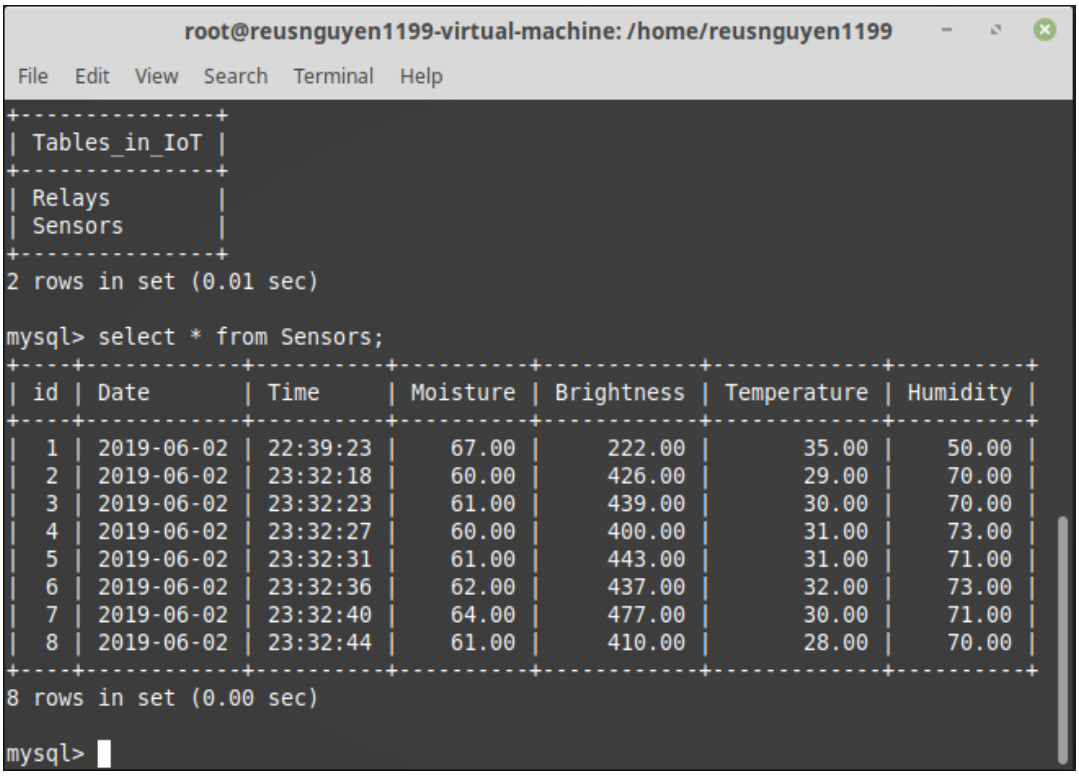
Nhóm sử dụng MQTT Broker là Mosquitto, cài đặt trên hệ điều hành Ubuntu.

Database:

Nhóm sử dụng Hệ quản trị cơ sở dữ liệu là MySQL, tạo một database để lưu dữ liệu với tên "IoT", bao gồm 2 tables là "Sensors" và "Relays"

Việc nhận dữ liệu và ghi vào database sẽ do một đoạn Script Python thực hiện (hình minh họa bên dưới)



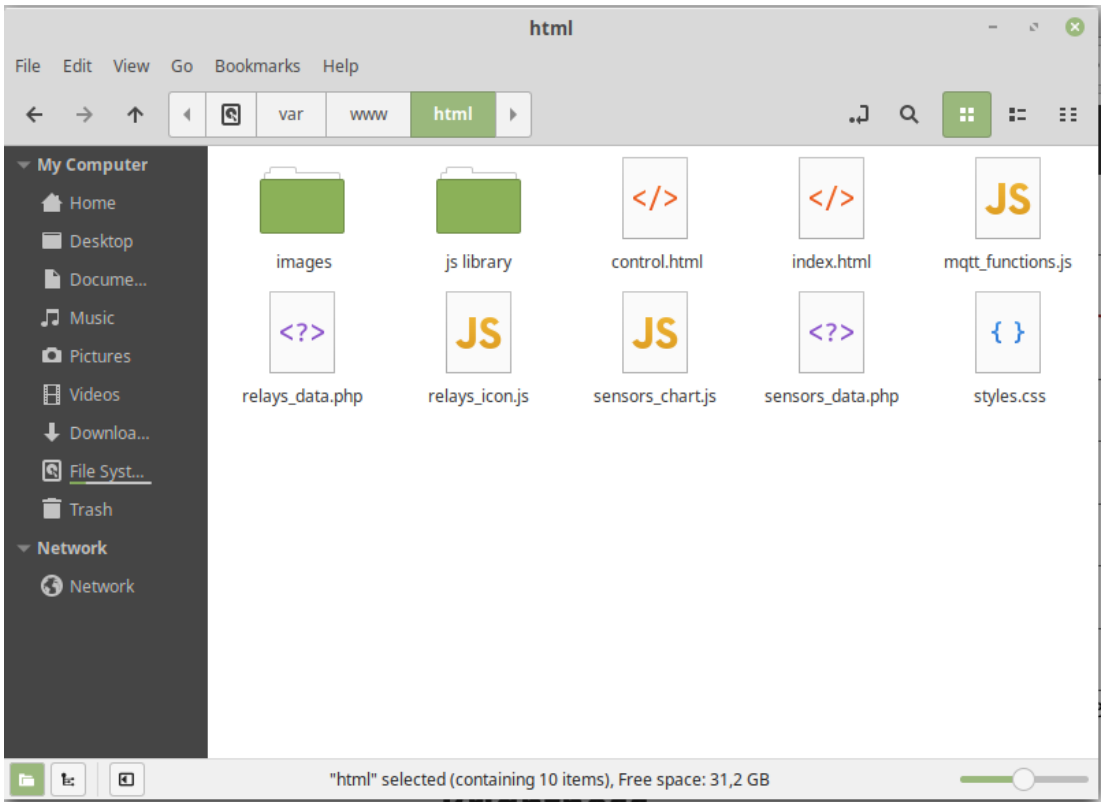


Hình 6.2: Dữ liệu nhận được tại MQTT Broker và dữ liệu trong database

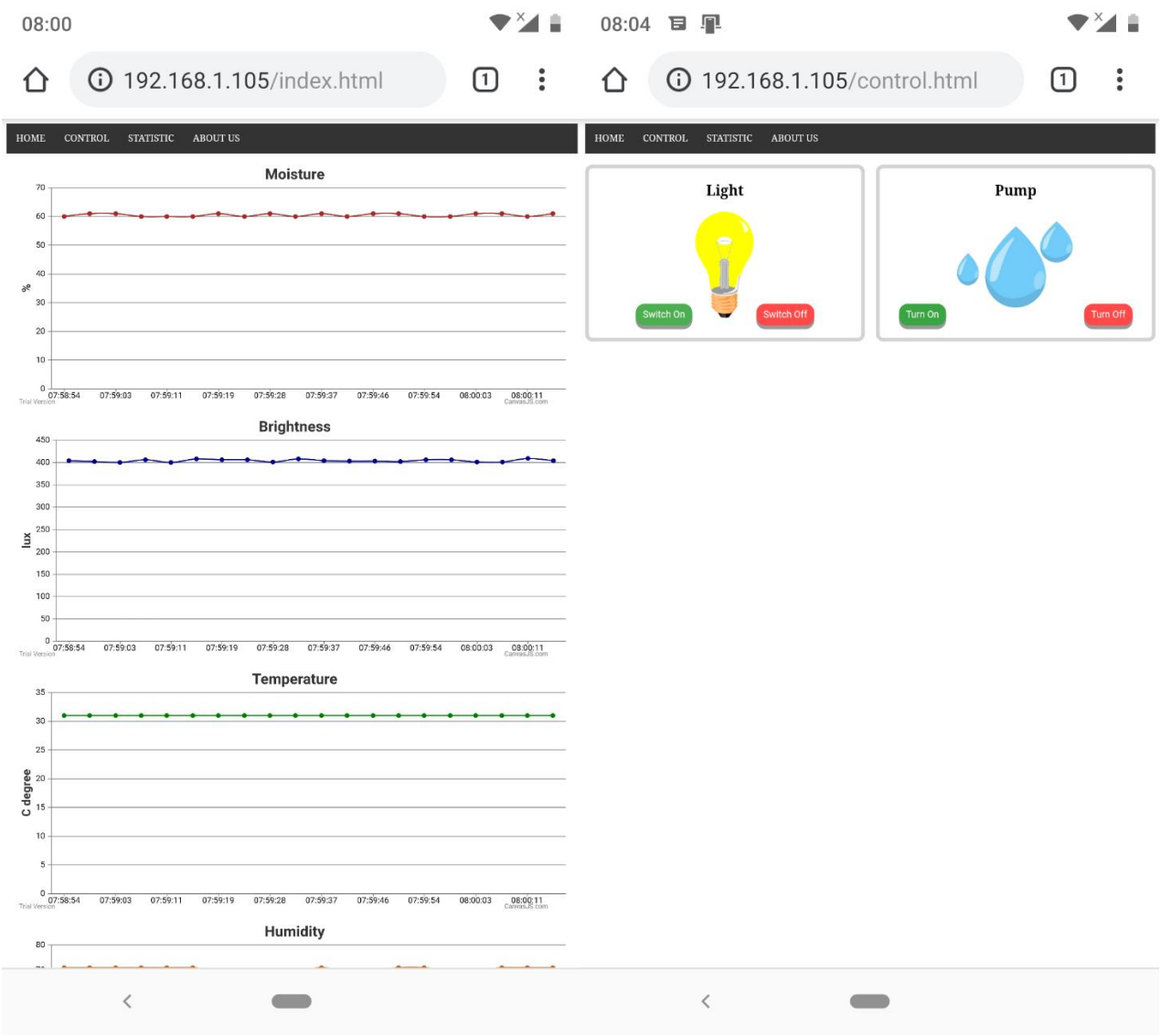
6.3 THIẾT KẾ WEB SERVER

Web Server: Nhóm sử dụng web server là Apache Webserver , chạy trên hệ điều hành Ubuntu

Thư viện để render các biểu đồ: Canvas Chart JS



Hình 6.3a: Các file cần thiết cho Web server



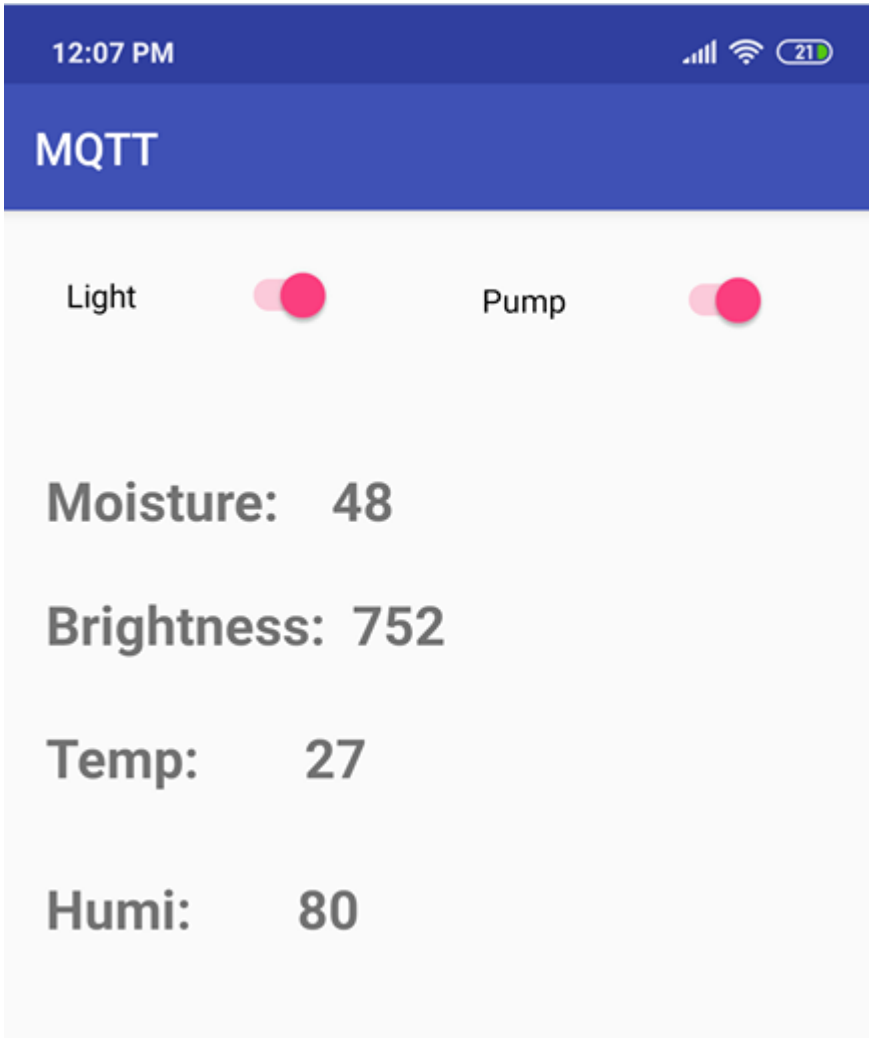
Hình 6.3b: Giao diện web trên smartphone

6.4 THIẾT KẾ ANDROID APP

Để hệ thống hoàn thiện và dễ truy cập hơn, nhóm thử tạo một app android đơn giản để quản lý mô hình.

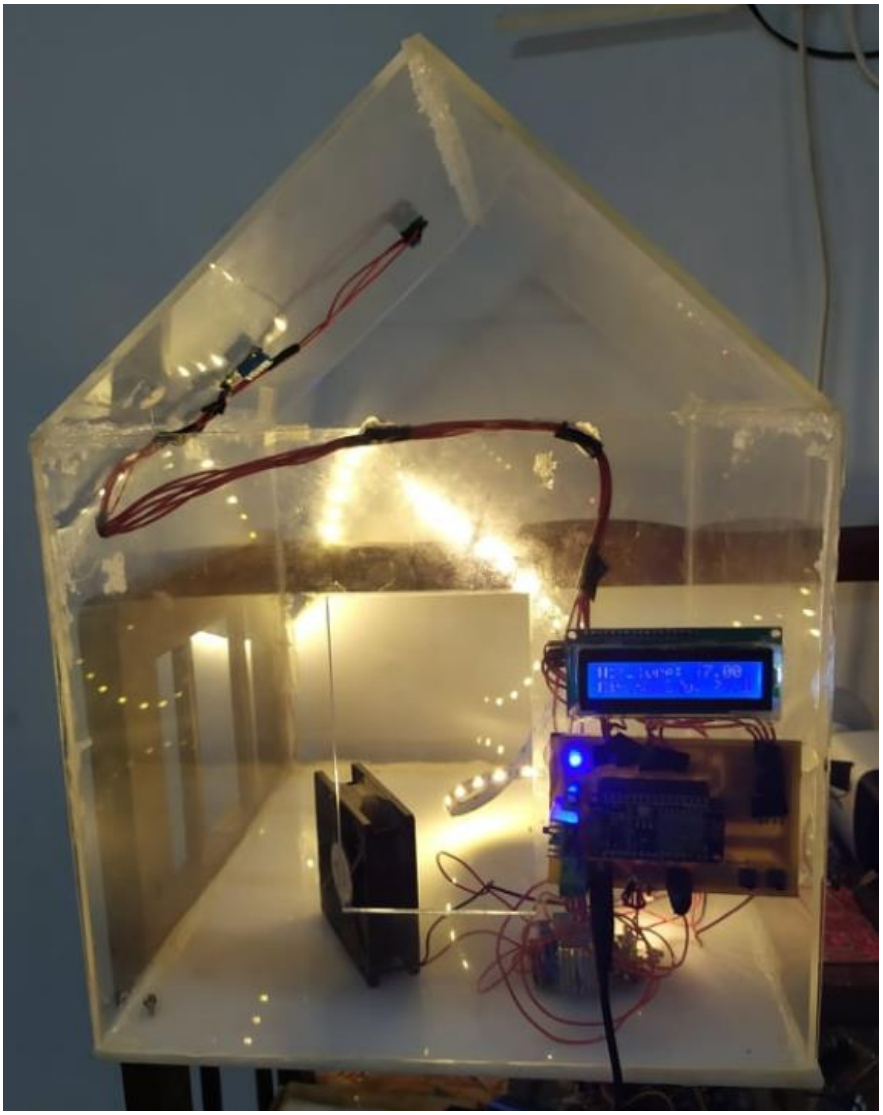
Trong đề tài, nhóm sử dụng phần mềm Android Studio để viết app bằng ngôn ngữ Java. Sử dụng thư viện MQTT có sẵn để đồng bộ dữ liệu giữa Client và MQTT Broker.

Chương trình sẽ nhận gói data Json từ MQTT Broker và hiển thị ra bên ngoài giao diện người dùng. Đồng thời sẽ gởi các tín hiệu điều khiển đèn và máy bơm từ yêu cầu của người dùng lên Broker.



Hình 6.4: Giao diện app Android

7. HOÀN THIỆN VÀ CHẠY THỬ



Hình 7.1: Mô hình nhà kính hoàn thiện.

8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sản phẩm hoạt động đúng yêu cầu đề ra, tuy nhiên điều khiển từ web đến hệ thống vẫn chưa đủ nhanh (độ trễ khoảng 1-2s)

Hướng phát triển:

- Phát triển các app trên nhiều nền tảng để phân phối tài nguyên xử lý
- Ứng dụng các framework vào phát triển nền tảng web

9. TÀI LIỆU THAM KHẢO

https://dev.mysql.com/doc/refman/8.0/en/

https://www.oasis-open.org/committees/download.php/49205/MQTT-OASIS-Webinar.pdf

https://httpd.apache.org/docs/2.4/

10. PHỤ LỤC

Python script để nhận dữ liệu từ vi điều khiển:

```
import paho.mqtt.client as mqtt
from store_data import data_handler

# MQTT Settings
MQTT_Broker = "192.168.1.105"
MQTT_Port = 1883
Keep_Alive_Interval = 45
MQTT_Topic1 = "Sensors"
MQTT_Topic2 = "Relays"

def on_connect(client, userdata, flags, rc):
  print("Connected with result code "+str(rc))
  client.subscribe(MQTT_Topic1)
  client.subscribe(MQTT_Topic2)

def on_message(client, userdata, msg):
    print("MQTT Data Received...")
    print("MQTT Topic: " + msg.topic)
    print("Data: " + str(msg.payload))
    data_handler(msg.topic, msg.payload)


client = mqtt.Client()
client.connect(MQTT_Broker,MQTT_Port,Keep_Alive_Interval)
```

```
client.on_connect = on_connect
client.on_message = on_message

client.loop_forever()
```

Các source code khác:

https://www.dropbox.com/sh/gx87rb16w8k5s0q/AAAys4W5ayJnKwDgSLAZYwg4a?dl=0