

TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
“HỌC PHẦN “INTERNET OF THINGS”

Đề tài:

ĐIỀU KHIỂN HỆ THỐNG TRONG GIA ĐÌNH

GVHD: Ths. Phạm Trung Minh

Sinh viên thực hiện:

Phạm Bá Huy – 87726

Vũ Trung Kiên – 86313

Phạm Quang Long – 86847

Hải Phòng, tháng 8 năm 2023

TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM

-----***-----

BÀI TẬP LỚN

HỌC PHẦN: HỌC PHẦN “INTERNET OF THINGS”

Chủ đề 01:

1. Tên đề tài

Điều khiển hệ thống trong gia đình

2. Mục đích

Tìm hiểu nguyên lý, công dụng, cách thức hoạt động và ứng dụng vào thực tế của việc điều khiển hệ thống trong gia đình có sử dụng kết nối mạng.

3. Công việc cần thực hiện

- Làm báo cáo bài tập lớn.
- Cài đặt, thực thi hệ thống điều khiển thiết bị trong gia đình.
- Bảo vệ bài tập lớn.

4. Yêu cầu

- Kết quả làm bài tập lớn: Báo cáo bài tập lớn
- Báo cáo bài tập lớn phải được trình bày theo mẫu quy định (kèm theo), báo cáo có thể kết xuất thành tệp định dạng PDF và nộp qua email hoặc có thể in ấn.
- Hạn nộp báo cáo bài tập lớn: 9/2023

5. Tài liệu tham khảo

- <http://arduino.vn/>
- <http://iot47.com/>

Hải Phòng, tháng 8 năm 2023

MỤC LỤC

MỤC LỤC	3
1. Giới thiệu bài toán	4
2. Mục tiêu đạt được	5
3. Ứng dụng thực tiễn	7
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	9
1.1. ỨNG DỤNG CỦA IOT	9
1.1.1. Cấu trúc cơ bản của một hệ thống ứng dụng công nghệ IOT	9
1.2. GIỚI THIỆU VỀ ỨNG DỤNG WEBSOCKET	10
1.3. GIỚI THIỆU VỀ GIAO THỨC I2C:	12
1.4. CẤU TRÚC PHẦN CỨNG	13
1.4.1. Kit ESP8266	13
1.4.2. Cảm biến nhiệt độ độ ẩm DHT11	14
1.4.3. Arduino Uno R3	15
CHƯƠNG 2. THIẾT KẾ VÀ THI CÔNG	18
THIẾT KẾ	18
2.2. NGUYÊN TẮC HOẠT ĐỘNG	18
2.3. THI CÔNG HỆ THỐNG	22
2.4. CODE CHƯƠNG TRÌNH:	24
2.4.1 Server.js	24
2.4.2 ESP8266.uno	29
2.4.3 Arduino.uno	33

LỜI MỞ ĐẦU

1. ĐẶT VẤN ĐỀ

Ngày nay với sự phát triển không ngừng của khoa học và công nghệ với những ứng dụng của khoa học kỹ thuật tiên tiến, thế giới chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Một trong những ứng dụng quan trọng trong công nghệ điện tử là Internet of Things hay IOT. IOT- Internet Vạn Vật là thuật ngữ dùng chung để đề cập đến hàng tỷ thiết bị vật lý trên khắp thế giới hiện đang được kết nối với Internet. Chúng thu thập và chia sẻ dữ liệu với nhau. Nhờ sự xuất hiện của chip máy tính siêu rẻ và sự phổ biến của mạng không dây, người ta có thể biến bất cứ thứ gì từ thứ nhỏ như viên thuốc đến lớn như máy bay trở thành một phần của IOT.

1. Giới thiệu bài toán

Trong thời đại hiện đại, sự phát triển nhanh chóng của công nghệ đã mở ra những cánh cửa mới cho cuộc sống hàng ngày của chúng ta. Trong đó, Internet of Things (IoT) đã trở thành một xu hướng quan trọng, mang lại sự kết nối giữa các thiết bị và hệ thống thông qua internet. Một trong những lĩnh vực ứng dụng tiềm năng của IoT là điều khiển hệ thống trong gia đình.

Bài toán "Điều Khiển Hệ Thống Trong Gia Đình thông qua IoT" đang ngày càng thu hút sự chú ý của cả người tiêu dùng và các nhà nghiên cứu trong lĩnh vực công nghệ thông tin. Trong một ngôi nhà thông minh, các thiết bị như đèn chiếu sáng, quạt điều hòa, thiết bị âm thanh, máy giặt, và thậm chí là hệ thống an ninh có thể được kết nối và điều khiển từ xa thông qua mạng internet.

Điều này mang lại nhiều lợi ích cho người sử dụng, bao gồm tiện lợi, tiết kiệm năng lượng, và tăng cường tính năng an ninh.

Bài toán này đặt ra một loạt các thách thức thú vị, từ việc thiết kế giao thức kết nối an toàn để đảm bảo thông tin không bị xâm phạm, đến việc phát triển ứng dụng điều khiển đơn giản và thân thiện với người dùng. Ngoài ra, việc tích hợp trí tuệ nhân tạo và học máy cũng có thể giúp hệ thống tự động học và thích ứng với thói quen của người dùng.

Qua bài toán này, chúng ta có thể nhìn thấy tiềm năng của IoT trong việc cải thiện cuộc sống hàng ngày. Việc tạo ra một môi trường sống thông minh, tiện nghi và hiệu quả về năng lượng đang là mục tiêu quan trọng của nhiều người và tổ chức. Với sự phát triển liên tục của công nghệ, chúng ta có thể kỳ vọng thấy nhiều giải pháp mới và tiến bộ hơn trong lĩnh vực này trong tương lai.

Nhìn chung, bài toán "Điều Khiển Hệ Thống Trong Gia Đình thông qua IoT" không chỉ mang lại sự tiện lợi mà còn mở ra cơ hội cho sự đổi mới và sáng tạo trong việc quản lý và tương tác với môi trường sống của chúng ta.

2. Mục tiêu đạt được

Mục tiêu của bài toán "Điều Khiển Hệ Thống Trong Gia Đình thông qua IoT" là tạo ra một môi trường sống thông minh và tiện nghi. Điều này bao gồm việc điều khiển từ xa các thiết bị như đèn, quạt, máy giặt thông qua điện thoại hoặc máy tính, giúp tiết kiệm năng lượng và tối ưu hóa trải nghiệm người dùng. Ở đây, nhóm chúng em sẽ điều khiển đèn và xem thông tin nhiệt độ, độ ẩm phòng.

Ý tưởng

1. Chọn Cảm Biến Nhiệt Độ và Độ Ẩm:

Sử dụng cảm biến nhiệt độ và độ ẩm như DHT11 hoặc DHT22 để đo nhiệt độ và độ ẩm trong phòng.

2. Lập Trình Một Vi Điều Khiển:

Sử dụng một vi điều khiển như Arduino hoặc Raspberry Pi để làm trung tâm điều khiển cho hệ thống của bạn.

Sử dụng ESP8266 Node MCU để giao tiếp giữa phần cứng và sever trên một hệ thống.

3. Kết Nối Cảm Biến:

Kết nối cảm biến nhiệt độ và độ ẩm với vi điều khiển thông qua các chân GPIO hoặc các giao tiếp khác (I2C, SPI, UART).

4. Lập Trình Đọc Dữ Liệu Cảm Biến:

Sử dụng mã lập trình để đọc dữ liệu từ cảm biến, có thể tìm thấy thư viện và ví dụ mã nguồn mở trực tuyến cho các cảm biến này.

5. Điều Khiển LED:

Kết nối LED với vi điều khiển và lập trình để điều khiển chúng

6. Hiển Thị Dữ Liệu:

Có thể hiển thị dữ liệu nhiệt độ và độ ẩm trên một màn hình LCD, hoặc gửi dữ liệu này đến một ứng dụng điện thoại thông qua kết nối Wi-Fi hoặc Bluetooth nếu bạn sử dụng Raspberry Pi hoặc một phiên bản có tích hợp sẵn kết nối không dây.

7. Tự Động Hóa:

Sử dụng các điều kiện và quy tắc để tự động điều khiển các thiết bị khác trong nhà, chẳng hạn như quạt hoặc máy điều hòa dựa trên dữ liệu nhiệt độ và độ ẩm.

8. Giao Diện Người Dùng (Tuỳ Chọn):

Tạo một giao diện người dùng trực quan để hiển thị dữ liệu và cho phép người dùng tương tác với hệ thống của bạn.

Lợi ích:

Tiết kiệm năng lượng: Bằng cách điều khiển đèn từ xa, bạn có thể giảm thiểu sự lãng phí năng lượng do quên khi ra ngoài.

Giám sát từ xa: Nếu bạn tích hợp kết nối mạng không dây vào hệ thống, bạn có thể giám sát nhiệt độ và độ ẩm từ xa thông qua ứng dụng di động hoặc trang web, giúp bạn biết được tình trạng của gia đình khi bạn không ở nhà.

Bảo vệ tài sản: Điều khiển nhiệt độ và độ ẩm có thể giúp bảo vệ tài sản như tranh, sách, đồ gỗ, và thiết bị điện tử khỏi hỏng hóc hoặc hủy hoại do điều kiện môi trường không thích hợp.

Tự động hóa thông minh: Hệ thống này có thể được tích hợp vào hệ thống tự động hóa thông minh lớn hơn, cho phép bạn tạo ra các quy tắc và kịch bản phức tạp hơn dựa trên dữ liệu từ cảm biến nhiệt độ và độ ẩm.

Tiện lợi và thoải mái: Hệ thống này giúp đơn giản hóa việc điều khiển đèn và quản lý môi trường trong gia đình của bạn, tạo ra sự tiện lợi và thoải mái.

Tiết kiệm thời gian: Bạn không cần phải thủ công điều khiển đèn hoặc kiểm tra nhiệt độ và độ ẩm thường xuyên, giúp bạn tiết kiệm thời gian và công sức.

3. Ứng dụng thực tiễn

Đề tài có ứng dụng rất lớn trong lĩnh vực điều khiển thiết bị trong gia đình, đặc biệt trong xã hội hiện đại ngày càng phát triển khi mà sự tiện lợi và nhanh chóng luôn luôn được đặt lên hàng đầu. Bài tập này có thể áp dụng cho từng gia

đình với nhiều mục đích khác nhau. Tuy nhiên trong bài nay nhóm chúng em chỉ xin giới thiệu mô hình thu nhỏ có khả năng mở rộng cao.

Áp dụng điều khiển hệ thống trong gia đình chúng ta có thể làm:

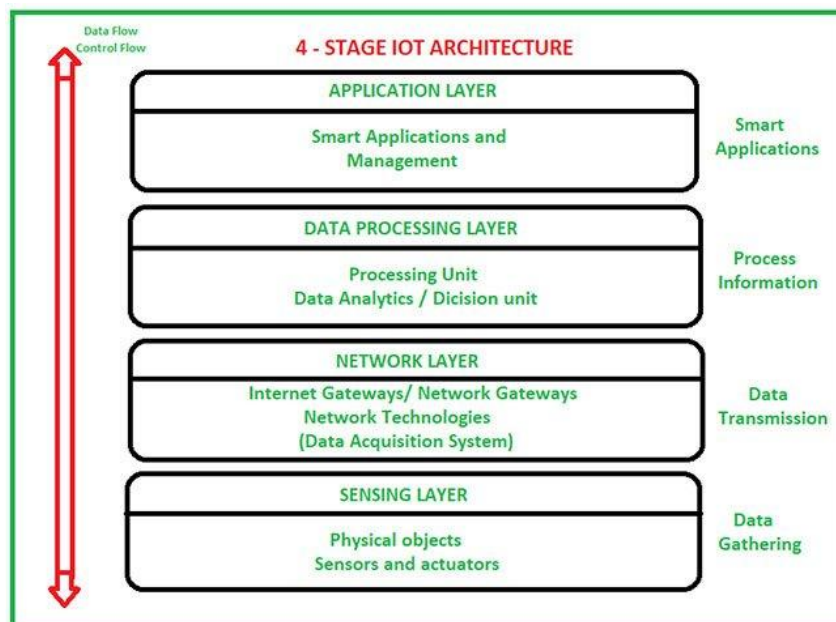
- Điều Khiển Ánh Sáng.
- Quản Lý Nhiệt Độ, Độ Ẩm.
- Điều Khiển Rèm Cửa Tự Động.
- Điều Khiển Thiết Bị Âm Thanh.

Và còn nhiều những ứng dụng khác đang và đã được áp dụng làm thuận tiện cho cuộc sống của chúng ta.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. ỨNG DỤNG CỦA IOT

1.1.1. Cấu trúc cơ bản của một hệ thống ứng dụng công nghệ IOT



- Lớp cảm biến (SENSING LAYER)

Cảm biến, thiết bị truyền động, các thiết bị có trong lớp cảm biến này. Các bộ phận cảm biến hoặc bộ truyền động này nhận dữ liệu (Thông số vật lý/ môi trường), xử lý dữ liệu và phát dữ liệu qua mạng.

- Lớp mạng (NETWORK LAYER)

Các cổng Internet (mạng), hệ thống thu nhập dữ liệu (Data Acquisition System -DAS) xuất hiện trong lớp này. DAS thực hiện chức năng tổng hợp và chuyển đổi dữ liệu (Thu thập dữ liệu và tổng hợp dữ liệu sau đó chuyển đổi dữ liệu analog của cảm biến sang dữ liệu digital,...) Các cổng nâng cao chủ yếu mở ra kết nối giữa mạng cảm biến và Internet cũng thực hiện nhiều chức năng cơ bản, như bảo vệ chống phần mềm độc hại và lọc một số lần ra quyết định dựa trên dữ liệu đã nhập và các dịch vụ quản lý dữ liệu.

- Lớp xử lý dữ liệu (DATA PROCESSING LAYER)

Đây là đơn vị xử lý của hệ sinh thái IoT. Tại đây, dữ liệu được phân tích và xử lý trước khi gửi đến trung tâm dữ liệu, nơi dữ liệu được truy cập bởi các ứng dụng phần mềm thường được gọi là ứng dụng kinh doanh. Đây là nơi dữ liệu được theo dõi và quản lý. Các hành động khác cũng được chuẩn bị tại lớp này.

- Lớp ứng dụng (APPLICATION LAYER)

Đây là lớp cuối cùng trong 4 giai đoạn của kiến trúc IoT. Trung tâm dữ liệu hoặc đám mây (Cloud) là giai đoạn quản lý, nơi dữ liệu được quản lý và sử dụng bởi các ứng dụng người dùng cuối như nông nghiệp, chăm sóc sức khỏe, hàng không vũ trụ, nông nghiệp, quốc phòng và những ứng dụng khác.

- Đặc tính cơ bản của IoT

Tính kết nối liên thông (Interconnectivity)

Với IoT thì bất cứ điều gì cũng có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.

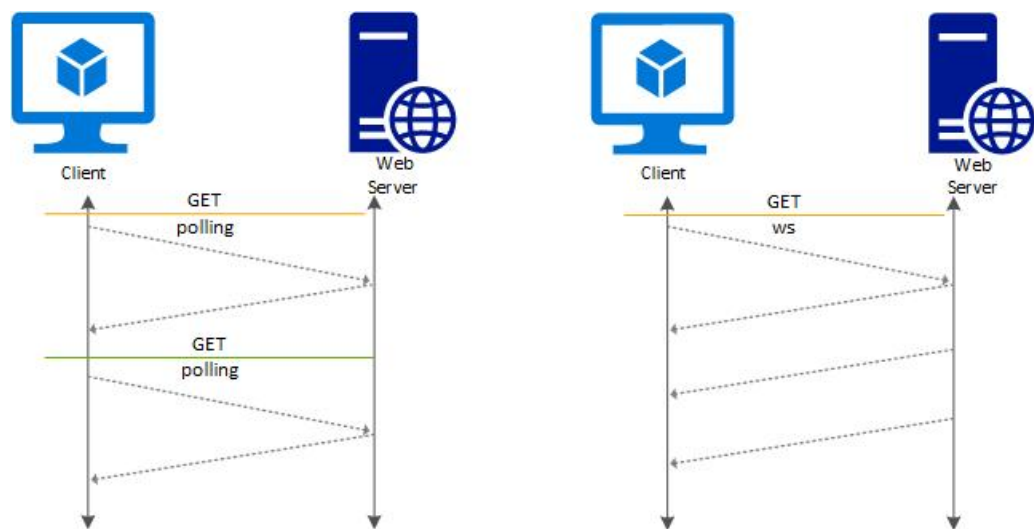
1.2.GIỚI THIỆU VỀ ỨNG DỤNG WEBSOCKET.

Websocket là công nghệ hỗ trợ giao tiếp hai chiều giữa máy tính khách hàng và máy chủ bằng cách sử dụng các socket TCP để tạo ra các kết nối hiệu quả và rẻ tiền. Chúng được thiết kế đặc biệt để sử dụng trong các ứng dụng web, nhưng các lập trình viên có thể tích hợp chúng vào bất kỳ loại ứng dụng nào.

WebSocket mới trong HTML5 là một kỹ thuật Ajax ngược. Sockets cho phép các kênh giao tiếp song song hai chiều và hiện được hỗ trợ bởi nhiều trình duyệt (Firefox, Google Chrome và Safari). Các kết nối được mở thông qua các yêu cầu HTTP (HTTP request) được gọi là các ràng buộc Sockets với các tiêu đề đặc biệt.

Kết nối được duy trì để bạn có thể đọc và ghi dữ liệu bằng JavaScript như thể bạn đang sử dụng các socket TCP thuần túy. Dữ liệu được gửi qua

giao thức HTTP (thường được sử dụng trong các kỹ thuật Ajax) chứa rất nhiều dữ liệu không cần thiết trong tiêu đề.



Cách sử dụng Websocket

Chúng ta hãy xem xét chức năng của Websocket và áp dụng chúng vào quy trình làm việc của bạn. Có thể chạy cùng lúc nhiều socket giúp tăng tốc công việc và nâng cao hiệu quả công việc. Sockets được hỗ trợ bởi hầu hết các hệ điều hành như Windows và Linux. Và nó được sử dụng trong nhiều ngôn ngữ lập trình khác nhau.

Sockets giúp lập trình viên kết nối các ứng dụng trong môi trường kết nối Internet để gửi và nhận dữ liệu bằng phương thức TCP/IP và UDP.

Hai ứng dụng phải biết thông tin IP và cổng của nhau nếu chúng cần trao đổi dữ liệu. Có nhiều loại socket khác nhau, tùy thuộc vào cách dữ liệu được truyền (giao thức). Phổ biến nhất là TCP và UDP.

Ưu – nhược điểm của Websocket

Về ưu điểm:

WebSockets cung cấp giao tiếp hai chiều mạnh mẽ với độ trễ thấp và tỷ lệ lỗi thấp. Nó không yêu cầu nhiều kết nối như phương pháp bỏ phiếu dài Comet và không có nhược điểm của phát trực tuyến Comet.

API cũng là một thư viện tuyệt vời để xử lý các kết nối lại, hết thời gian chờ, yêu cầu ajax (yêu cầu Ajax), xác nhận và các phương tiện truyền tải tùy chọn khác nhau (Ajax long polling và JSONP polling) trực tiếp mà không cần các lớp bổ sung. Điều này rất đơn giản so với Comet thường yêu cầu.

Về nhược điểm:

Đây là một thông số kỹ thuật mới cho HTML5, vì vậy không phải tất cả các trình duyệt đều hỗ trợ nó. Không có khu vực bắt buộc. Sockets là công TCP, không phải là yêu cầu HTTP, vì vậy việc sử dụng các dịch vụ nhận biết yêu cầu như SessionInViewFilter của Hibernate không phải là chuyện nhỏ.

1.3. GIỚI THIỆU VỀ GIAO THỨC I2C:

I2C (Inter-Integrated Circuit) là một giao thức truyền thông kỹ thuật số dùng để kết nối giữa các thiết bị điện tử trong một mạch điện. Giao thức này cho phép các thiết bị trao đổi dữ liệu và điều khiển lẫn nhau qua một dây dẫn chung. I2C ban đầu được phát triển bởi Philips Semiconductor (hiện là NXP Semiconductors) vào những năm 1980. Giao thức này đã nhanh chóng trở thành một tiêu chuẩn công nghiệp phổ biến, được sử dụng rộng rãi trong nhiều ứng dụng khác nhau như điện tử tiêu dùng, ô tô, thiết bị y tế, viễn thông, và nhiều lĩnh vực khác.

I2C sử dụng hai dây dẫn:

SCL (Serial Clock Line): Dây đồng hồ nối các thiết bị để đồng bộ hóa truyền thông.

SDA (Serial Data Line): Dây dữ liệu nối các thiết bị để truyền và nhận dữ liệu.

I2C có hai vai trò chính:

Master: Thiết bị chủ quản lý việc khởi tạo và kiểm soát truyền thông. Master có thể gửi yêu cầu đọc và ghi tới các thiết bị khác.

Slave: Thiết bị tùy thuộc vào master và thực hiện các yêu cầu ghi và đọc từ master.

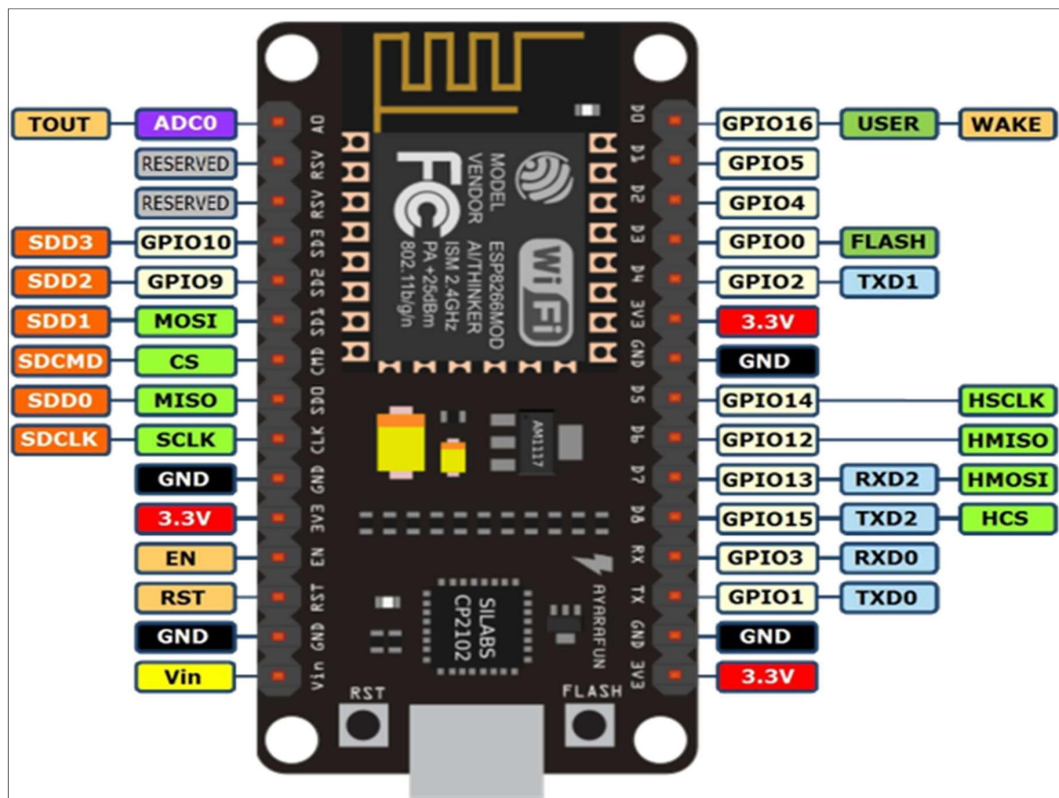
Giao thức I2C sử dụng địa chỉ của các thiết bị để xác định người nhận hoặc người gửi dữ liệu

1.4. CẤU TRÚC PHẦN CỨNG

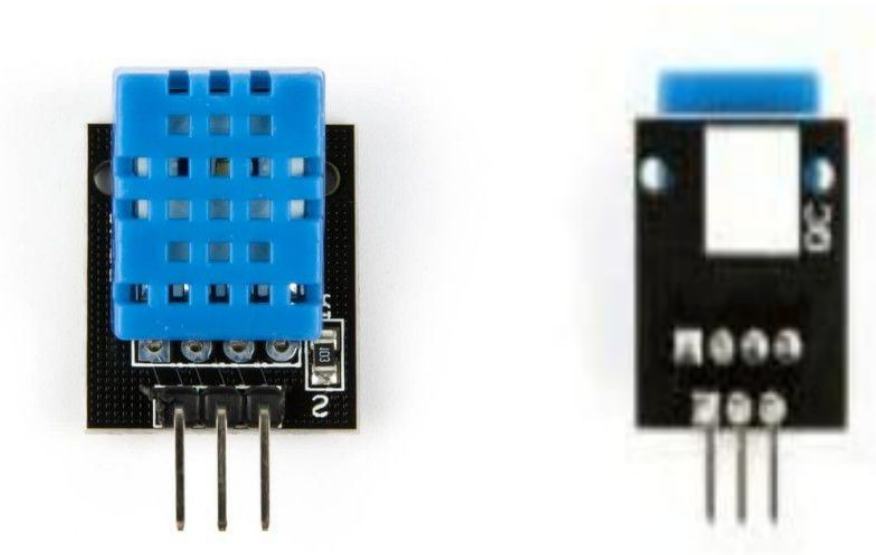
1.4.1. Kit ESP8266

Kit ESP8266 là kit phát triển dựa trên nền chip Wifi SoC ESP8266 với thiết kế dễ dàng sử dụng vì tích hợp sẵn mạch nạp sử dụng chip CP2102. Bên trong ESP8266 có sẵn một lõi vi xử lý vì thế bạn có thể trực tiếp lập trình cho ESP8266 mà không cần thêm bất kì con vi xử lý nào nữa. Hiện tại có hai ngôn ngữ có thể lập trình cho ESP8266, sử dụng trực tiếp phần mềm IDE của Arduino để lập trình với bộ thư viện riêng hoặc sử dụng phần mềm node MCU

Kit RF thu phát Wifi ESP8266 NodeMCU được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT



1.4.2. Cảm biến nhiệt độ độ ẩm DHT11



Cảm biến **DHT11** là một loại cảm biến đo độ ẩm và nhiệt độ trong môi trường. Nguyên tắc hoạt động của cảm biến DHT11 dựa trên việc sử dụng một cảm biến đo sự thay đổi của điện trở trong môi trường để ước tính độ ẩm và sử dụng một cảm biến đo nhiệt độ.

DHT11 bao gồm một cảm biến đo độ ẩm và một cảm biến đo nhiệt độ, cùng với một vi mạch điều khiển và giao tiếp. Dưới đây là cách cảm biến DHT11 hoạt động:

Đo độ ẩm: Cảm biến đo độ ẩm của DHT11 dựa trên sự thay đổi của điện trở giữa hai điện cực dẫn điện trong phản ứng với sự thay đổi độ ẩm trong môi trường. Cảm biến có một lớp chất nhạy ẩm được đặt giữa hai điện cực. Khi độ ẩm tăng, chất nhạy ẩm sẽ hấp thụ nước và dẫn đến sự thay đổi điện trở giữa hai điện cực. Điện trở này được đo và chuyển đổi thành giá trị độ ẩm.

Đo nhiệt độ: Cảm biến đo nhiệt độ sử dụng một cảm biến nhiệt độ tích hợp để đo nhiệt độ môi trường. Thông thường, cảm biến nhiệt độ này sẽ thay đổi điện trở hoặc điện áp theo nhiệt độ.

Vi mạch điều khiển và giao tiếp: Dữ liệu từ cảm biến độ ẩm và nhiệt độ được xử lý bởi một vi mạch điều khiển tích hợp trên cảm biến DHT11. Vi mạch này chịu trách nhiệm đọc giá trị điện trở hoặc điện áp từ cảm biến và tính toán giá trị độ ẩm và nhiệt độ tương ứng.

Cảm biến DHT11 thường sử dụng giao tiếp số để truyền dữ liệu về độ ẩm và nhiệt độ. Thông qua một chân dữ liệu (data pin), cảm biến sẽ truyền dữ liệu theo một giao thức đơn giản với vi điều khiển ngoại vi (ví dụ: Arduino). Cách giao tiếp chính thường là sử dụng giao thức dạng "single-wire" (giao tiếp qua một dây duy nhất) để đồng bộ hóa việc đọc dữ liệu.

Sau khi cảm biến DHT11 đã đọc và tính toán được độ ẩm và nhiệt độ, dữ liệu này có thể được truyền tới vi điều khiển để sử dụng và hiển thị, chẳng hạn như hiển thị trên màn hình hoặc lưu trữ vào bộ nhớ.

Lưu ý rằng cảm biến DHT11 có độ chính xác tương đối thấp so với các cảm biến cao cấp khác, như DHT22 hay các cảm biến chuyên nghiệp hơn.

1.4.3. Arduino Uno R3

Nguyên lý hoạt động:

Arduino Uno R3 được sử dụng bằng cách gắn vào máy tính thông qua một cáp USB. Sau khi đã lắp đặt xong, chúng ta sẽ sử dụng pin hoặc bộ chuyển đổi AC-DC để cung cấp điện cho mạch kit. Khi đầu nối thành công, mạch sẽ kích hoạt và bắt đầu.



Vai trò của mạch kit Arduino UNO R3

UNO được thiết kế để hỗ trợ sự phát triển của phần mềm Arduino IDE 1.0. Lý do mạch kit này có tên Arduino UNO R3 là vì chúng là phiên bản sửa đổi mới nhất, thứ 3 của Arduino Uno. Có một số thay đổi:

Chip điều khiển USB được thay đổi từ ATmega8U2 (flash 8K) thành ATmega16U2 (flash 16K). Điều này không làm tăng flash hoặc RAM có sẵn cho các bản phác thảo.

Trang bị thêm ba chân mới. Trong đó, các chân I2C (A4, A5) được đưa ra bên cạnh bảng gần AREF. Một chân IOREF bên cạnh chân đặt lại, là một bản sao của chân 5V.

Nút đặt lại hiện nằm bên cạnh đầu nối USB, giúp dễ tiếp cận hơn khi sử dụng tấm chắn.

Ngoài ra, mạch kit này cũng đóng vai trò quan trọng và chính trong bảng bảng USB-Arduino

Đặc điểm

Một trong những ưu điểm nổi bật của mạch kit arduino uno r3 là người sử dụng có thể thay đổi bộ vi điều khiển trên bảng trong trường hợp họ gặp phải sự cố hay mắc lỗi.

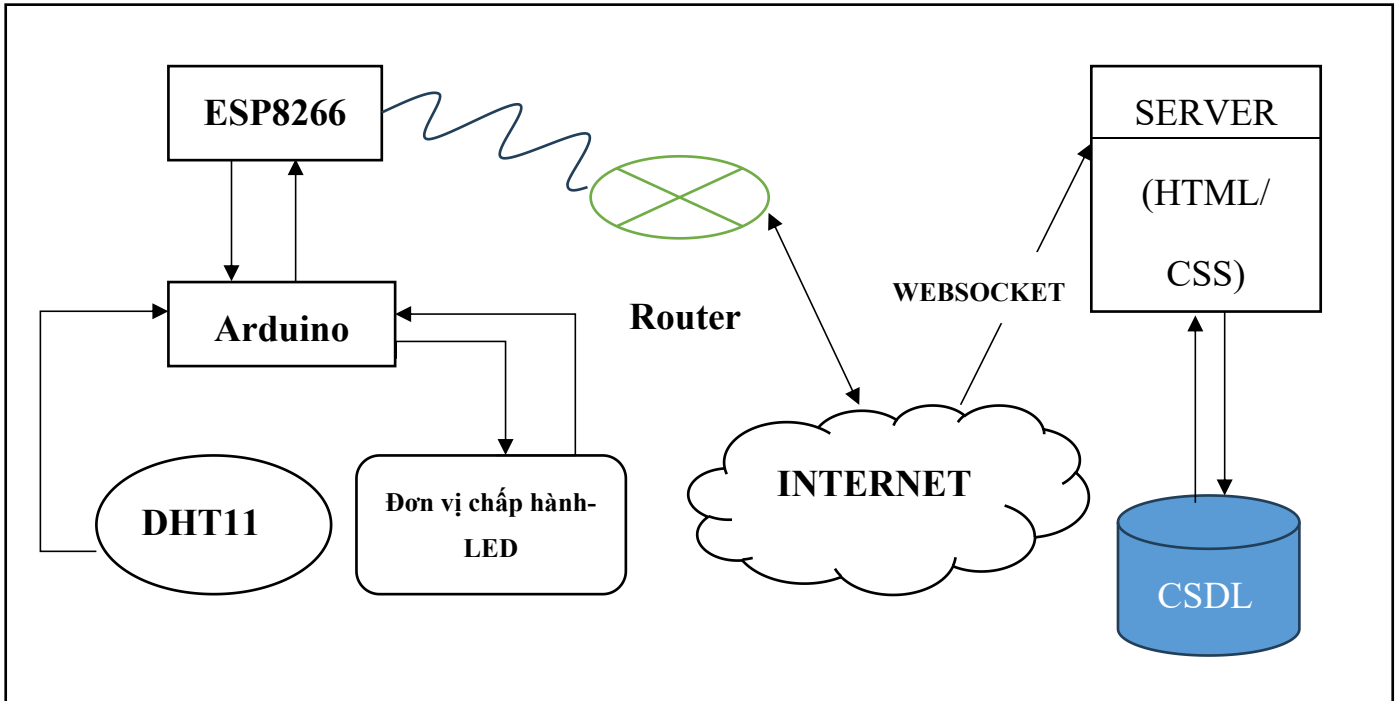
Ngoài ra, bộ kit này còn mang đến cho người sử dụng nhiều tính năng tuyệt vời như:

Khả năng tháo rời.

- Tích hợp sẵn trong DIP (gói nội tuyến kép).
- Khả năng điều khiển ATmega328.
- Dễ dàng tải lập trình.
- Ưu điểm cuối cùng nhưng không kém phần quan trọng là: Arduino có một cộng đồng hỗ trợ lớn và một bộ thư viện hỗ trợ phong phú. Cùng với “lá chắn” phần cứng bổ sung phía sau. Điều này sẽ là một sự lựa chọn tuyệt vời cho những người mới bắt đầu làm việc trong lĩnh vực thiết bị điện tử nhúng.

CHƯƠNG 2. THIẾT KẾ VÀ THI CÔNG

2.1. THIẾT KẾ



Sơ đồ khối hệ thống

2.2. NGUYÊN TẮC HOẠT ĐỘNG

Hệ thống gồm 2 phần:

- **Phần hệ thống nhúng:** Ghép nối các thiết bị: ESP8266 với cảm biến chuyển động, ESP8266 và bóng đèn.

Nguyên tắc hoạt động:

+ Khi có yêu cầu bật/tắt đèn từ web client, một request được gửi về server của WebSocket và xử lý để đưa vào ESP8266, sau khi ESP8266 nhận được tín hiệu bật đèn từ sever thì bắt đầu gửi sang Arduino nhờ I2C để bật đèn

+ Dữ liệu từ cảm biến độ ẩm và nhiệt độ được xử lý bởi một vi mạch điều khiển tích hợp trên cảm biến DHT11. Vi mạch này chịu trách nhiệm đọc giá trị điện trở hoặc điện áp từ cảm biến và tính toán giá trị độ ẩm và nhiệt độ tương ứng.

Cảm biến DHT11 thường sử dụng giao tiếp số để truyền dữ liệu về độ ẩm và nhiệt độ. Thông qua một chân dữ liệu (data pin), cảm biến sẽ truyền dữ liệu theo một giao thức đơn giản với vi điều khiển ngoại vi (ví dụ: Arduino). Cách giao tiếp chính thường là sử dụng giao thức dạng "single-wire" (giao tiếp qua một dây duy nhất) để đồng bộ hóa việc đọc dữ liệu.

Sau khi cảm biến DHT11 đã đọc và tính toán được độ ẩm và nhiệt độ, dữ liệu này có thể được truyền tới vi điều khiển để sử dụng và hiển thị, chẳng hạn như hiển thị trên màn hình hoặc lưu trữ vào bộ nhớ.

- **Phần kết nối hệ thống IOT:** Kết nối ESP8266 với Server WebSocket

+ **Tạo địa chỉ trên Ngrok:** Ngrok sẽ tạo ra một địa chỉ public, giống như <https://randomstring.ngrok.io>. Bất kỳ ai truy cập địa chỉ này sẽ được định tuyến đến localhost của bạn trên cổng bạn đã chỉ định (trong ví dụ trên, cổng 80). Ngrok cho phép bạn public địa chỉ local của thiết bị IoT và tạo ra một địa chỉ public truy cập từ bất kỳ đâu trên Internet.

+ **Gửi/ Nhận yêu cầu bật đèn LED:** Arduino sẽ nghe tín hiệu "t" từ cổng Serial. Khi nhận được tín hiệu, Arduino sẽ đảo trạng thái đèn LED và gửi thông điệp trạng thái về cho server.

Khi nhấn nút trên giao diện client web, một tín hiệu "toggle_light" sẽ được gửi từ client đến server thông qua kết nối WebSocket. Server sau đó xử lý tín hiệu và gửi lệnh tương ứng tới ESP8266 thông qua hàm `websocket.onEvent(websocketEvent)` nhận được thông qua `Chedo = (char*)payload`.

Hàm `loop()` chạy và `Guidulieu_Arduino()` nhận được và tiếp tục truyền tín hiệu dạng kí tự thông qua giao thức I2C

Arduino nhận được thông qua `onReceive()`, kiểm tra nếu `Wire.available() > 0` tức nhận được ký tự thì bật đèn LED

+ Gửi yêu cầu cảm biến:

Arduino đọc dữ liệu từ cảm biến DHT11 (độ ẩm và nhiệt độ) sử dụng thư viện DHT. Arduino sử dụng thư viện Wire để gửi dữ liệu độ ẩm và nhiệt độ tới ESP8266 thông qua giao thức I2C.

Dữ liệu được gửi từ Arduino tới ESP8266 qua I2C là hai giá trị số thực: độ ẩm và nhiệt độ.

ESP8266 sử dụng thư viện Wire để nhận dữ liệu từ Arduino thông qua giao thức I2C. ESP8266 sử dụng thư viện `WebSocketClient` để kết nối tới server WebSocket và gửi dữ liệu độ ẩm và nhiệt độ. Khi Arduino gửi dữ liệu qua I2C, ESP8266 nhận dữ liệu này trong hàm `requestEvent()`.

Server WebSocket tiếp tục nhận tín hiệu từ client và quản lý các kết nối WebSocket.

Khi ESP8266 gửi dữ liệu độ ẩm và nhiệt độ qua kết nối WebSocket, server nhận dữ liệu này và có thể giải mã chuỗi JSON để trích xuất các giá trị độ ẩm và nhiệt độ.

Dữ liệu này có thể được sử dụng để cập nhật trực tiếp trên giao diện người dùng của bạn hoặc thực hiện các hành động khác tương ứng với dữ liệu độ ẩm và nhiệt độ

+ Cập nhật dữ liệu JSON

Trong hàm `loop()`, hàm `sendReadings()` sẽ đọc các thông số LED và cảm biến DHT11 sau đó gán vào `DynamiJsonDocument` và truyền qua websocket nhờ `websocket.sendTxt(JSON)`;

Ở server, hàm `onmessage()` của `WebSocket` liên tục lắng nghe từ các phản hồi client về và sau đó cập nhật các giá trị cảm biến lên web client

+ **Cập nhật Cơ sở dữ liệu:**

Khi bạn chạy ứng dụng Node.js thông qua lệnh `node app.js`, nó bắt đầu lắng nghe trên một cổng cụ thể (trong ví dụ này, cổng 3000)

Ứng dụng kết nối đến cơ sở dữ liệu MySQL bằng cách sử dụng thông tin đăng nhập như tên người dùng và mật khẩu để xác thực vào cơ sở dữ liệu.

Ứng dụng Express định nghĩa hai route: `/fetch` và `/insert`.

Khi bạn gửi một yêu cầu GET đến `/fetch`, ứng dụng thực hiện truy vấn `SELECT` trên cơ sở dữ liệu MySQL để lấy dữ liệu từ bảng và trả về kết quả dưới dạng JSON.

Khi bạn gửi một yêu cầu POST đến `/insert`, ứng dụng nhận dữ liệu gửi lên, thực hiện truy vấn `INSERT` để chèn dữ liệu vào cơ sở dữ liệu, và sau đó trả về thông báo xác nhận.

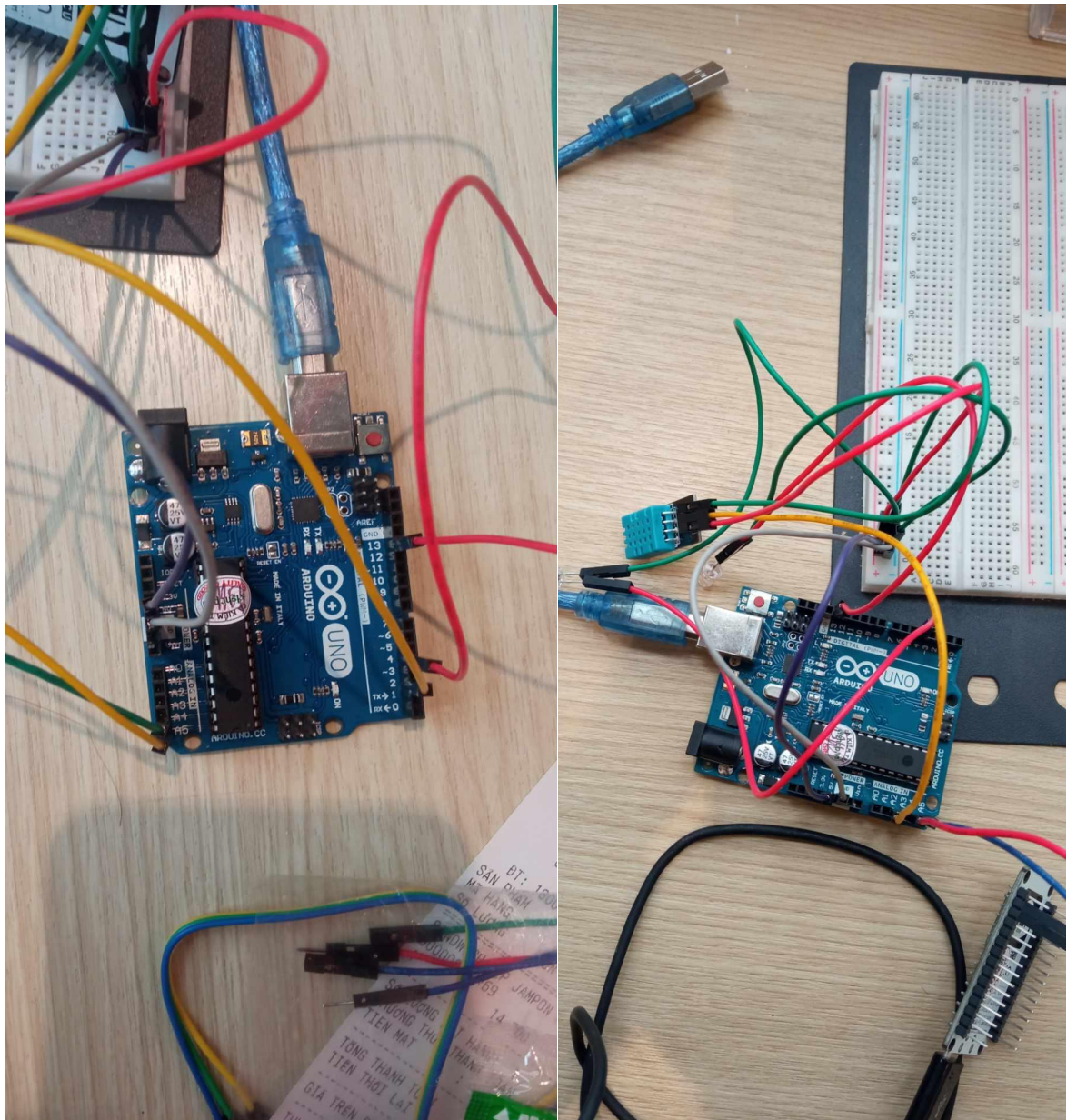
Trang web HTML chứa nút "Chèn".

Khi bạn nhấp vào "Chèn", JavaScript sử dụng `fetch()` để gửi yêu cầu GET đến `/fetch`. Khi nhận được dữ liệu JSON từ server, nó có thể xử lý hoặc hiển thị dữ liệu lên trang web.

Khi bạn nhấp vào "Insert Data", JavaScript sử dụng `fetch()` để gửi yêu cầu POST đến `/insert` với dữ liệu cần chèn. Sau khi server thực hiện truy vấn `INSERT`, nó trả về thông báo xác nhận và JavaScript có thể xử lý kết quả hoặc hiển thị nó trên trang web.

Khi bạn truy cập ứng dụng từ trình duyệt và nhấp vào nút "Chèn" ứng dụng sẽ tương tác với cơ sở dữ liệu MySQL thông qua các địa chỉ đã định nghĩa và cung cấp dữ liệu hoặc kết quả trả về cho người dùng.

2.3. THI CÔNG HỆ THỐNG



Hình ảnh lắp đặt

Nguồn cấp cho ESP8266 là nguồn 5V thông qua cáp kết nối Micro USB.
Nguồn cấp cho cảm biến DHT11 là 5V lấy từ VCC của Arduino

- Chân tín hiệu cảm biến nối với chân D2 của Arduino Uno R3
- Công giao tiếp I2C giữa Arduino UnoR3 và ESP8266 lần lượt D2-A5 và D1-A4
- Ý nghĩa các nút:
 - Nút bật LED gửi tín hiệu từ client web về server WebSocket và sau đó mới gửi tiếp tới Arduino để bật đèn
 - Humidity nhận dữ liệu cập nhật số độ ẩm theo % dựa trên cảm biến DHT11
 - Temperature nhận dữ liệu cập nhật số độ C ($^{\circ}C$) dựa trên cảm biến DHT11



2.4. CODE CHƯƠNG TRÌNH:

2.4.1 Server.js

```
const WebSocket = require('ws');

const http = require('http');

const fs = require('fs');

const path = require('path');

const express = require('express'); // Thêm Express.js

const bodyParser = require('body-parser');

const mysql = require('mysql');

const app = express(); // Khởi tạo Express.js


// Phục vụ các tệp tĩnh từ thư mục public

app.use(express.static(path.join(__dirname, 'public')));

////////////////////////////////////

// parse application/x-www-form-urlencoded

app.use(bodyParser.urlencoded({ extended: false }));


// parse application/json

app.use(bodyParser.json())
```

```

// MySQL

const pool = mysql.createPool({
  connectionLimit : 10,
  host            : 'localhost',
  user            : 'root',
  password        : '',
  database        : 'n10i'
})

app.get('/get', (req, res) => {
  pool.getConnection((err, connection) => {
    if(err) throw err

    console.log('connected as id ' + connection.threadId)

    connection.query('SELECT date, led1, led2, temperature, humidity
FROM dbiot', (err, rows) => {
      connection.release() // return the connection to pool

      if (!err) {
        res.send(rows)
      } else {
        console.log(err)
      }
    })

    // if(err) throw err
  })
})

```

```

        console.log('The data from database table are: \n', rows)
    })
})
});

app.post('/insert', (req, res) => {

    pool.getConnection((err, connection) => {

        if(err) throw err

        const params = req.body;

        connection.query('INSERT INTO dbiot SET ?', params, (err, rows) =>
{

        connection.release();

        if (!err) {

            // res.json(params);

            res.send('oke');

        } else {

            console.log(err);

        }

        //console.log('The data from dbiot table are:11 \n', rows);

    })

})

```

```

});

app.get('/:date', (req, res) => {
  pool.getConnection((err, connection) => {
    const date = req.params.date;
    if(err) throw err
    connection.query('SELECT * FROM dbiot WHERE date
LIKE ?', [date + '%'], (err, rows) => {
      connection.release() // return the connection to pool
      if (!err) {
        res.send(rows)
      } else {
        console.log(err)
      }
      //console.log('aaaa: \n', rows)
    })
  })
});

app.get('/index', function(req, res) {
  res.render('index.html');
});

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```
// Tạo máy chủ HTTP để phục vụ trang web và kết nối WebSocket
```

```
const server = http.createServer(app);
```

```
// Tạo máy chủ WebSocket
```

```
const wss = new WebSocket.Server({ server });
```

```
var str,tm,hm;
```

```
wss.on('connection', (ws) => {
```

```
    console.log('Client connected');
```

```
    ws.on('message', (message) => {
```

```
        // console.log(`Received: ${message}`);
```

```
        // Trả lại tin nhắn cho tất cả các kết nối khác
```

```
        wss.clients.forEach((client) => {
```

```
            if (client !== ws && client.readyState === WebSocket.OPEN) {
```

```
                client.send(message+ "");
```

```
            }
```

```
        });
```

```
    });
```

```
    ws.on('close', () => {
```

```
        console.log('Client disconnected');
```

```
    });
```

```
});
```

```
server.listen(3000, () => {
    console.log('Server is running on http://localhost:3000');
});
```

2.4.2 ESP8266.uno

```
#include <Arduino.h>

#include <ArduinoJson.h>

#include <Wire.h>

#include <ESP8266WiFi.h>

#include <WebSocketsClient.h>

unsigned long times;

WebSocketsClient webSocket;

const char* ssid = "Phạm Bá Huy";

const char* password = "huyent61dh";

const char* ip_host = "192.168.243.55";

const uint16_t port = 3000;

float receivedTemperature = 0.0;

float receivedHumidity = 0.0;

float t,h;

String Chedo;

String Led;

void websocketEvent(WStype_t type, uint8_t* payload, size_t length) {
```

```

Chedo = (char*)payload;

switch (type) {

  case WStype_DISCONNECTED:

    Serial.printf("[WSc] Disconnected!\n");

    break;

  case WStype_CONNECTED:

    {

      Serial.printf("[WSc] Connected to url: %s\n", payload);

    }

    break;

  case WStype_BIN:

    Serial.println(Chedo);

    break;

}

}

void setup() {

  Serial.begin(9600);

  Wire.begin(5, 4);

  times = millis();

  Serial.println("ESP8266 Websocket Client");

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {

```

```

    delay(500);

    Serial.print(".");
}

Serial.println(WiFi.localIP());

WebSocket.begin(ip_host, port);

WebSocket.onEvent(WebSocketEvent);

}

void Guidulieu_Arduino() {

    Wire.beginTransmission(8); /* begin with device address 8 */

    Wire.write(Chedo.c_str()); /* sends hello string */

    Wire.endTransmission(); /* stop transmitting */

}

void loop() {

    readArduino();

    sendReadings();

    Guidulieu_Arduino();

    WebSocket.loop();

}

const size_t bufferSize = JSON_OBJECT_SIZE(4);

void sendReadings() {

```



```

// Tạo một đối tượng JSON
DynamicJsonDocument jsonBuffer(bufferSize);

Led = Chedo.c_str();

if(Led == "2"){
    jsonBuffer["led1"] = 2;
}else if(Led == "3"){
    jsonBuffer["led1"] = 3;
}else{
    jsonBuffer["led1"] = 3;
}

if(Led == "4"){
    jsonBuffer["led2"] = 4;
}else if(Led == "5"){
    jsonBuffer["led2"] = 5;
}else{
    jsonBuffer["led2"] = 5;
}

jsonBuffer["temperature"] = t;
jsonBuffer["humidity"] = h;

String jsonString;

serializeJson(jsonBuffer, jsonString);

WebSocket.sendTXT(jsonString);

```

```

}

void readArduino(){

  String string, string1, string2;

  Wire.requestFrom(8, 10);

  do {

    char c = Wire.read();

    string = string + c;

    string1 = string.substring(0, 5);

    string2 = string.substring(5);

  } while (Wire.available());

  h = atof(string1.c_str());

  t = atof(string2.c_str());

  Serial.print(string1);

  Serial.println();

  Serial.print(string2);

  Serial.println();

}

```

2.4.3 Arduino.uno

```

#include <Wire.h>

#include <DHT.h>

#define led1 4

#define led2 13

#define dhtPin 2

```

```

DHT dht(dhtPin, DHT11);

char vy[7];

char ix[8];

void onRequest(int count) {
    float humidity, temperature;

    humidity = float(dht.readHumidity());
    temperature = float(dht.readTemperature());

    dtostrf(humidity, 5, 2, ix);
    dtostrf(temperature, 5, 2, vy);

    Wire.write(ix);

    Wire.write(vy);
}

void setup() {
    Wire.begin(8);

    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);

    Wire.onReceive(onReceive); /* register receive event */

    Serial.begin(9600); /* start serial for debug */

    dht.begin();

    Wire.onRequest(onRequest);
}

void loop() {

```

```

    readDHTData();
}

void onReceive(int count) {
    if (Wire.available() > 0) {
        char c = Wire.read();
        Serial.print(c);
        if (c == '2') {
            digitalWrite(13, HIGH);
        } else if (c == '3') {
            digitalWrite(13, LOW);
        } else if (c == '4') {
            digitalWrite(4, HIGH);
        } else if (c == '5') {
            digitalWrite(4, LOW);
        }
    }
}

void readDHTData() {
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();
    //Serial.print("Nhiệt độ (°C): ");
    Serial.println(temperature);
    //Serial.print("Độ ẩm (%): ");

```

```
Serial.println(humidity);  
  
Wire.beginTransmission(8);  
  
Wire.write((uint8_t*)&humidity, sizeof(float));  
  
Wire.write((uint8_t*)&temperature, sizeof(float));  
  
Wire.endTransmission();  
  
}
```

KẾT LUẬN

Bài tập lớn này đã mang lại cho chúng em kiến thức trong việc thực hiện một hệ thống đơn giản liên quan đến Internet of Things (IoT). Chúng em đã thành công trong việc tạo ra một ứng dụng đơn giản cho gia đình, cho phép chúng em điều khiển một đèn LED và hiển thị thông tin về độ ẩm và nhiệt độ trong môi trường. Dưới đây là một số điểm chúng em đã học được từ dự án này:

Kết nối và Điều Khiển Thiết Bị: Chúng em đã học cách kết nối và điều khiển một thiết bị vật lý (đèn LED) bằng cách sử dụng một vi điều khiển nhúng (trong trường hợp này, Arduino). Điều này đã giúp chúng em hiểu cách tương tác với thiết bị thông qua mã lập trình.

Sử Dụng Cảm Biến: Chúng em đã sử dụng cảm biến nhiệt độ và độ ẩm để thu thập dữ liệu từ môi trường. Điều này đã giúp chúng em hiểu cách sử dụng các cảm biến để đọc thông tin môi trường và sử dụng dữ liệu này trong ứng dụng của mình.

Lập Trình Ứng Dụng Web: Chúng em đã phát triển một ứng dụng web đơn giản để điều khiển đèn LED và hiển thị dữ liệu độ ẩm và nhiệt độ. Điều này đã giúp chúng em làm quen với lập trình web và cách giao tiếp giữa trình duyệt và máy chủ.

Tự Động Hóa: Bài tập lớn này đã cho thấy sức mạnh của tự động hóa thông qua việc kết hợp dữ liệu từ cảm biến để điều khiển thiết bị (đèn LED). Điều này có tiềm năng giúp tiết kiệm năng lượng và tạo ra môi trường thoải mái trong gia đình.

Trong tổng kết, dự án "Điều Khiển Đèn LED và Hiển Thị Thông Tin Độ Ẩm và Nhiệt Độ" không chỉ là một bài tập lớn thú vị mà còn là một cơ hội để phát triển kỹ năng và hiểu biết về lĩnh vực IoT và tự động hóa môi trường.

Chúng em cảm ơn thầy (cô) đã giúp đỡ và xem xét do thời gian có hạn nên đề tài không tránh khỏi những sai sót, mong thầy góp ý, đánh giá giúp chúng em hoàn thiện.