

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA**



HỆ THỐNG NHÚNG

BÀI TIỂU LUẬN:

SMART LAMP

Nhóm thực hiện:	Vũ Thị Linh	21010474	K15-DTVT1
	Lê Thị Hoàng Bắc	21010469	K15-DTVT1
	Khổng Thị Dung	21012374	K15 – DTVT1

Giảng viên hướng dẫn: TS. Nguyễn Ngọc Thanh
Khoa: Điện – Điện tử

Hà Nội, tháng 07 năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA



HỆ THỐNG NHÚNG

BÀI TIỂU LUẬN:

SMART LAMP

Nhóm thực hiện:	Vũ Thị Linh	21010474	K15-DTVT1
	Lê Thị Hoàng Bắc	21010469	K15-DTVT1
	Khổng Thị Dung	21012374	K15 – DTVT1

Giảng viên hướng dẫn: TS. Nguyễn Ngọc Thanh
Khoa: Điện – Điện tử

Hà Nội, tháng 07 năm 2024

TIẾN ĐỘ THỰC HIỆN

I) Thành viên nhóm

STT	Hình ảnh	Thông tin cá nhân
1		<ul style="list-style-type: none"> - Họ và tên: Lê Thị Hoàng Bắc - Mã SV: 21010469 - Lớp: K15-DTVT1 - Trường Đại học Phenikaa. - Nơi ở: Hà Nội
2		<ul style="list-style-type: none"> - Họ và tên: Vũ Thị Linh - Mã SV: 21010469 - Lớp: K15-DTVT1 - Trường Đại học Phenikaa. - Nơi ở: Hà Nội
3		<ul style="list-style-type: none"> - Họ và tên: Vũ Thị Linh - Mã SV: 21012374 - Lớp: K15-DTVT1 - Trường Đại học Phenikaa. - Nơi ở: Hà Nội

II) Phân công nhiệm vụ

Thành viên	Nhiệm vụ được giao	Nhóm đánh giá
Vũ Thị Linh	Led (nút nhấn)	
Lê Thị Hoàng Bắc	Software (Web, SQL) Firmware (DHT22, Led(Cầu vòng), LCD Hardware (Mô hình) Báo cáo, Slide	Đạt
Khổng Thị Dung	Led (nút nhấn)	

Mục lục

LỜI NÓI ĐẦU	7
Lý do chọn đề tài.....	7
Mục tiêu	8
CHƯƠNG I: CƠ SỞ LÝ THUYẾT	9
1.1. Tổng quan về hệ thống nhúng	9
1.1.2 Raspberry Pi 4 mode B	10
1.1.3. Cảm biến đo nhiệt độ độ ẩm DHT22	19
1.1.4. LCD I2C	21
1.1.5. Module LED RGB 5050 SMD KY-009	22
1.2. Webserver (Apache).....	24
1.3. Cơ sở dữ liệu MySQL (MariaDB).....	25
1.6. Kết luận chương.....	27
CHƯƠNG II: THIẾT KẾ HỆ THỐNG	28
2.1. Mô tả hệ thống	28
2.1.1. Giới thiệu	28
2.1.2. Tính năng	28
2.1.3. Phạm vi áp dụng	28
2.2. Sơ đồ khối hệ thống	29
2.3. Sơ đồ mạch điện và giải thuật	30
2.3.1. Kết nối Raspberry Pi 4 với DHT22	30
2.3.2. Kết nối Raspberry Pi 4 với LCD I2C	33
2.3.3. Kết nối Raspberry Pi 4 với LED RGB	37
2.3. Thiết kế cơ sở dữ liệu	39
2.4. Lưu đồ thuật toán.....	40
2.5. Lựa chọn ngôn ngữ lập trình và thư viện.....	41
2.6. Kết luận chương.....	42
CHƯƠNG III: KẾT QUẢ	44
3.1. Sản phẩm đạt được.....	44

3.1.1. Phần mềm	44
3.1.2. Phần cứng	46
3.2. Nhận xét.....	46
3.2.1. Phần mềm	46
3.2.2. Phần cứng	46
3.2.3. Cải tiến.....	46
TÀI LIỆU THAM KHẢO.....	47

DANH MỤC HÌNH ẢNH

<i>Hình 1. 1 Hệ thống nhúng</i>	<i>9</i>
<i>Hình 1. 2 Raspberry Pi 4 Model B</i>	<i>10</i>
<i>Hình 1. 3 Cấu hình Raspberry Pi 4 mode B</i>	<i>12</i>
<i>Hình 1. 4 Sơ đồ chân Raspberry Pi 4 mode B.....</i>	<i>13</i>
<i>Hình 1. 5 Cảm Biến DHT22</i>	<i>19</i>
<i>Hình 1. 6 Module LCD I2C</i>	<i>21</i>
<i>Hình 1. 7 Module LED RGB 5050 SMD KY-009</i>	<i>22</i>
<i>Hình 1. 8 Giao thức UDP.....</i>	<i>24</i>
<i>Hình 1. 11 SQL Server.....</i>	<i>25</i>
<i>Hình 2. 1 Sơ đồ khối hệ thống</i>	<i>29</i>
<i>Hình 2. 2 Sơ đồ tổng quát.....</i>	<i>30</i>
<i>Hình 2. 3 Raspberry và DHT22.....</i>	<i>30</i>
<i>Hình 2. 4 Single-Wire Two-Way (SWTW).....</i>	<i>31</i>
<i>Hình 2. 5 Nguyên lý SWTW</i>	<i>32</i>
<i>Hình 2. 6 Raspberry và LCD.....</i>	<i>33</i>
<i>Hình 2. 7 Inter – Integrated Circuit (I2C).....</i>	<i>34</i>
<i>Hình 2. 8 Step conect I2C with Raspberry</i>	<i>34</i>
<i>Hình 2. 9 Raspberry và LED</i>	<i>37</i>
<i>Hình 2. 10 Chế độ xung điều khiển</i>	<i>38</i>
<i>Hình 2. 11 Lưu đồ thuật toán</i>	<i>40</i>
<i>Hình 3. 1 Giao diện phần mềm.....</i>	<i>44</i>
<i>Hình 3. 2 Cơ sở dữ liệu Led</i>	<i>44</i>
<i>Hình 3. 3 Cơ sở dữ liệu DHT22</i>	<i>45</i>
<i>Hình 3. 4 Phần cứng.....</i>	<i>46</i>

LỜI NÓI ĐẦU

Lý do chọn đề tài

Trong bối cảnh cuộc sống hiện đại ngày càng phát triển, nhu cầu về không gian sống thông minh, tiện nghi và tiết kiệm năng lượng ngày càng gia tăng. Đèn thông minh (Smart Lamp) nổi lên như một giải pháp hữu hiệu đáp ứng những yêu cầu này. Không chỉ đơn thuần là thiết bị chiếu sáng, đèn thông minh còn tích hợp nhiều tính năng đa dạng như điều khiển từ xa, tùy chỉnh màu sắc, cường độ sáng, hẹn giờ, cảm biến môi trường... mang lại trải nghiệm sống hiện đại và tiện ích cho người dùng. Chính vì tính thực tiễn cao, phù hợp với xu hướng công nghệ, khả năng mở rộng và phát triển, cũng như mang lại nhiều thách thức và cơ hội học hỏi, đề tài "Đèn thông minh" đã được lựa chọn để nghiên cứu và xây dựng trong khuôn khổ môn học Hệ thống nhúng.

Đèn thông minh là một sản phẩm công nghệ có tính ứng dụng rộng rãi trong đời sống hàng ngày, từ gia đình, văn phòng đến các không gian công cộng. Việc nghiên cứu và xây dựng một hệ thống đèn thông minh không chỉ mang lại giá trị học thuật mà còn có thể áp dụng vào thực tế, góp phần nâng cao chất lượng cuộc sống. Hơn nữa, đèn thông minh là một ví dụ điển hình của thiết bị IoT, cho phép kết nối và điều khiển từ xa thông qua mạng Internet, phù hợp với xu hướng công nghệ đang phát triển mạnh mẽ hiện nay. Việc nghiên cứu về đèn thông minh sẽ giúp người học tiếp cận và làm chủ các công nghệ IoT tiên tiến.

Đèn thông minh là một hệ thống mở, có thể dễ dàng tích hợp thêm các tính năng mới như điều khiển bằng giọng nói, học hỏi thói quen người dùng, kết nối với các thiết bị thông minh khác trong nhà... Điều này tạo ra nhiều cơ hội để nghiên cứu, phát triển và sáng tạo thêm các ứng dụng mới dựa trên nền tảng đèn thông minh. Đồng thời, việc xây dựng một hệ thống đèn thông minh đòi hỏi kiến thức và kỹ năng về nhiều lĩnh vực khác nhau như điện tử, lập trình, mạng máy tính, cơ sở dữ liệu... Đây là cơ hội để người học trau dồi kiến thức chuyên môn, rèn luyện kỹ năng thực hành và phát triển tư duy sáng tạo.

Với những lý do trên, đề tài "Đèn thông minh" được đánh giá là một đề tài hấp dẫn, thiết thực và mang tính ứng dụng cao, phù hợp với mục tiêu của môn học Hệ thống nhúng.

Mục tiêu

Mục tiêu của đề tài này là thiết kế và xây dựng một hệ thống đèn thông minh tích hợp nhiều tính năng ưu việt, mang lại trải nghiệm chiếu sáng tiện nghi và hiện đại cho người dùng. Cụ thể, đề tài hướng đến các mục tiêu sau:

- Xây dựng hệ thống điều khiển đèn thông minh qua giao diện web: Phát triển một giao diện web trực quan, thân thiện với người dùng, cho phép điều khiển từ xa các tính năng của đèn như bật/tắt, điều chỉnh độ sáng, thay đổi màu sắc RGB.
- Tích hợp cảm biến DHT22 để đo lường và hiển thị thông tin môi trường: Sử dụng cảm biến DHT22 để đo nhiệt độ và độ ẩm trong môi trường xung quanh, hiển thị thông tin này trên màn hình LCD I2C, đồng thời lưu trữ dữ liệu vào cơ sở dữ liệu SQL để theo dõi và phân tích.
- Lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu SQL: Thiết kế cơ sở dữ liệu để lưu trữ thông tin về màu sắc đèn, nhiệt độ, độ ẩm. Xây dựng các chức năng truy xuất dữ liệu từ cơ sở dữ liệu để hiển thị trên màn hình LCD và phục vụ cho việc phân tích, báo cáo.
- Ứng dụng các công nghệ tiên tiến: Vận dụng kiến thức và kỹ năng về Raspberry Pi, DHT22, LCD I2C, LED RGB, các ngôn ngữ lập trình và cơ sở dữ liệu để xây dựng một hệ thống đèn thông minh hoàn chỉnh.
- Đánh giá hiệu năng và tối ưu hóa hệ thống: Tiến hành các bài kiểm tra, đánh giá độ chính xác của cảm biến, độ ổn định của hệ thống, tốc độ phản hồi khi điều khiển qua giao diện web. Từ đó, đề xuất các giải pháp tối ưu hóa để nâng cao hiệu suất hoạt động của hệ thống.

CHƯƠNG I: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về hệ thống nhúng

Hệ thống nhúng là một hệ thống máy tính chuyên dụng được thiết kế để thực hiện một nhiệm vụ cụ thể hoặc một tập hợp các nhiệm vụ trong một môi trường xác định. Chúng thường được tích hợp vào các thiết bị điện tử hoặc cơ khí lớn hơn, đóng vai trò điều khiển, giám sát hoặc xử lý dữ liệu. Điểm đặc biệt của hệ thống nhúng là chúng hoạt động độc lập, không cần sự can thiệp của người dùng và thường có yêu cầu cao về độ tin cậy, hiệu suất và thời gian phản hồi.



Hình 1. 1 Hệ thống nhúng

Các thành phần chính của một hệ thống nhúng bao gồm:

- **Phần cứng:** Thường bao gồm một vi xử lý hoặc vi điều khiển, bộ nhớ (RAM, ROM, Flash), các thiết bị ngoại vi (cảm biến, bộ truyền động, giao diện người dùng) và các mạch điện tử hỗ trợ.
- **Phần mềm:** Là các chương trình được nhúng vào bộ nhớ của hệ thống, điều khiển hoạt động của phần cứng và thực hiện các chức năng cụ thể của hệ thống.
- **Hệ điều hành thời gian thực (RTOS):** Trong nhiều trường hợp, hệ thống nhúng sử dụng RTOS để quản lý tài nguyên, lập lịch tác vụ và đảm bảo tính thời gian thực của hệ thống.

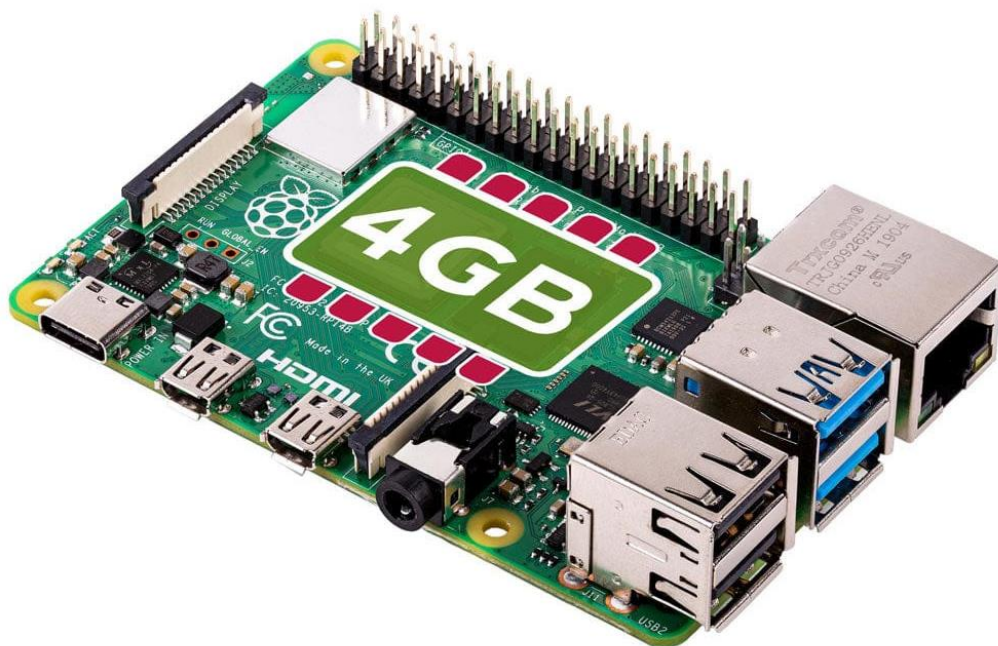
Các ứng dụng của hệ thống nhúng rất đa dạng và phong phú, có thể kể đến như:

- Điện tử tiêu dùng: Điện thoại di động, máy ảnh, tivi thông minh, thiết bị gia dụng thông minh...
- Công nghiệp: Hệ thống điều khiển tự động, robot công nghiệp, thiết bị đo lường, giám sát...
- Ô tô: Hệ thống điều khiển động cơ, hệ thống phanh ABS, hệ thống giải trí đa phương tiện...
- Y tế: Máy theo dõi bệnh nhân, máy trợ tim, thiết bị chẩn đoán hình ảnh...
- Hàng không vũ trụ: Hệ thống điều khiển bay, hệ thống định vị, hệ thống thông tin liên lạc...

Đề tài "Đèn thông minh" là một ví dụ cụ thể của hệ thống nhúng, trong đó Raspberry Pi đóng vai trò là vi xử lý trung tâm, điều khiển các thành phần khác như cảm biến DHT22, màn hình LCD I2C và đèn LED RGB. Phần mềm được lập trình trên Raspberry Pi sẽ xử lý dữ liệu từ cảm biến, điều khiển ánh sáng và màu sắc của đèn LED, giao tiếp với người dùng qua giao diện web và lưu trữ dữ liệu vào cơ sở dữ liệu.

1.1.2 Raspberry Pi 4 mode B

Raspberry Pi 4 Model B: Trái tim của hệ thống đèn thông minh



Hình 1. 2 Raspberry Pi 4 Model B

Raspberry Pi 4 Model B là một máy tính đơn board mạnh mẽ sử dụng hệ điều hành Linux, linh hoạt và giá cả phải chăng, được lựa chọn là bộ não của hệ thống đèn thông minh trong đề tài này. Với khả năng xử lý vượt trội, kết nối đa dạng và cộng đồng hỗ trợ rộng lớn, Raspberry Pi 4 Model B cung cấp một nền tảng lý tưởng để xây dựng và phát triển các ứng dụng nhúng.

Các thông số kỹ thuật nổi bật:

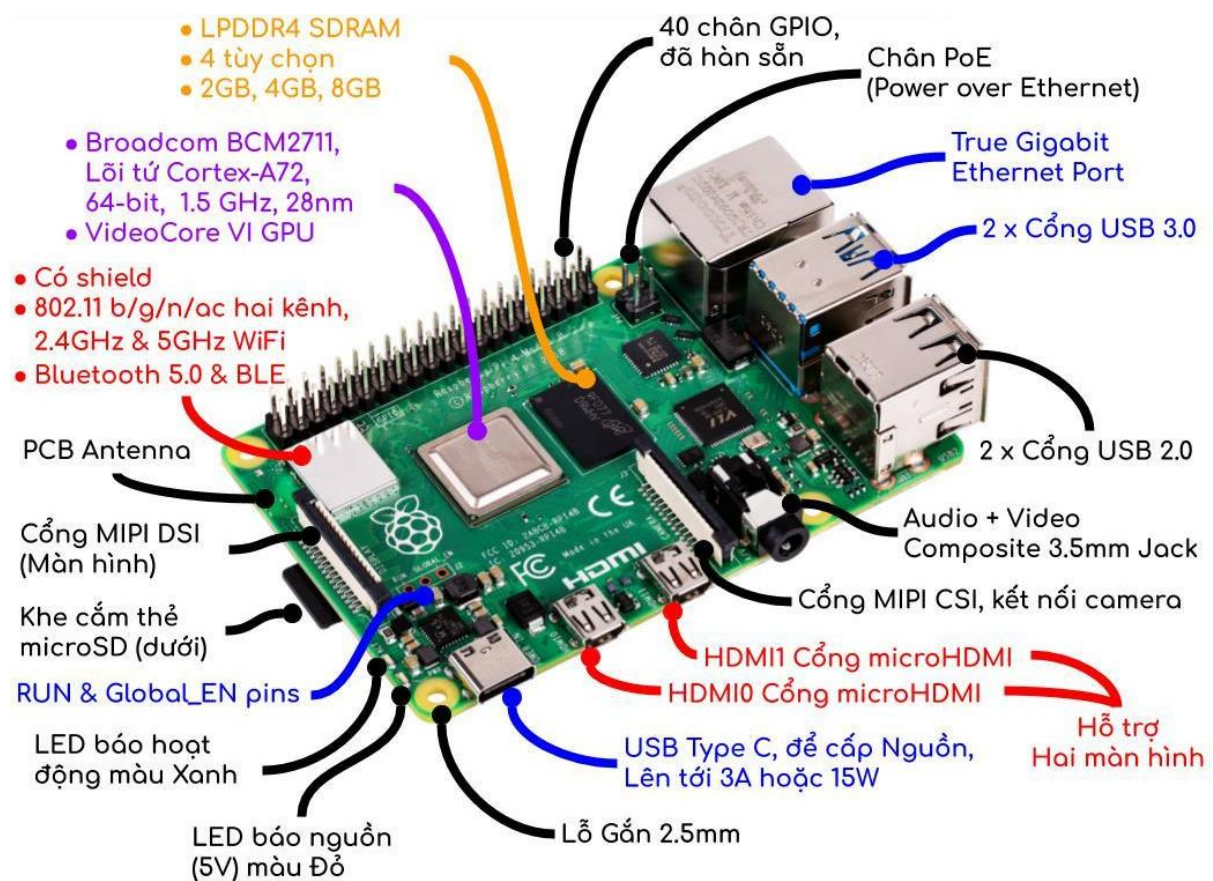
- Vi xử lý: Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, mang lại hiệu năng xử lý mạnh mẽ cho các tác vụ phức tạp.
- Bộ nhớ: Tùy chọn RAM 2GB, 4GB LPDDR4-2400 SDRAM, đảm bảo khả năng chạy đa nhiệm mượt mà và xử lý dữ liệu lớn.
- Kết nối không dây:
 - Wi-Fi: Hỗ trợ chuẩn 802.11ac dual-band (2.4GHz và 5.0GHz), cho phép kết nối mạng nhanh chóng và ổn định.
 - Bluetooth 5.0: Hỗ trợ kết nối với các thiết bị ngoại vi không dây như chuột, bàn phím, loa...
- Kết nối có dây:
 - Gigabit Ethernet: Cung cấp kết nối mạng có dây tốc độ cao, lý tưởng cho việc truyền tải dữ liệu lớn.
 - USB: 2 cổng USB 3.0 và 2 cổng USB 2.0, cho phép kết nối với nhiều thiết bị ngoại vi khác nhau.
- Đồ họa:
 - 2 cổng micro HDMI: Hỗ trợ xuất hình ảnh ra hai màn hình 4K đồng thời.
 - OpenGL ES 3.0: Hỗ trợ đồ họa 3D, giúp hiển thị giao diện người dùng đẹp mắt và mượt mà.
- GPIO: 40 chân GPIO (General Purpose Input/Output), cho phép kết nối và điều khiển các thiết bị ngoại vi như cảm biến, đèn LED, động cơ...
- Hệ điều hành: Hỗ trợ nhiều hệ điều hành khác nhau, trong đó Raspberry Pi OS (dựa trên Debian) là lựa chọn phổ biến nhất.

Lý do lựa chọn Raspberry Pi 4 Model B:

- Hiệu năng mạnh mẽ: Đáp ứng tốt các yêu cầu xử lý dữ liệu, điều khiển thiết bị và chạy các ứng dụng phức tạp.
- Kết nối đa dạng: Hỗ trợ đầy đủ các kết nối cần thiết cho hệ thống đèn thông minh, bao gồm Wi-Fi, Bluetooth, Ethernet, USB và GPIO.
- Cộng đồng hỗ trợ lớn: Có rất nhiều tài liệu, hướng dẫn và dự án mẫu liên quan đến Raspberry Pi, giúp người dùng dễ dàng tìm hiểu và phát triển ứng dụng.
- Giá cả phải chăng: So với các máy tính đơn board khác có cùng hiệu năng, Raspberry Pi 4 Model B có giá thành rất cạnh tranh.

Với những ưu điểm trên, Raspberry Pi 4 Model B là một lựa chọn hoàn hảo để làm nền tảng cho hệ thống đèn thông minh trong đề tài này.

Cấu hình Raspberry Pi 4 mode B



Hình 1. 3 Cấu hình Raspberry Pi 4 mode B

Sơ đồ chân Raspberry Pi 4 mode B

FUNCTION		PIN	PIN	FUNCTION	
3V3	3V3	1	2	5V	5V
GPI02	SPI3 MOSI/SDA3	3	4	5V	5V
GPI03	SPI3 SCLK/SCL3	5	6	GND	GND
GPI04	SPI4 CE0 N/SDA 3	7	8	TXD1/SPI5 MOSI	TXD1/SPI5 MOSI
GND	GND	9	10	RXD1/SPI5 SCLK	RXD1/SPI5 SCLK
GPI017		11	12	SPI6 CEO N	SPI6 CEO N
GPI027	SPI6 CE1 N	13	14	GND	GND
GPI022	SDA6	15	16	SCL6	SCL6
3V3	3V3	17	18	SPI3 CE1 N	SPI3 CE1 N
GPI010	SDA5	19	20	GND	GND
GPI09	RXD4/SCL4	21	22	SPI4 CE1 N	SPI4 CE1 N
GPI011	SCL5	23	24	SDA4/TXD4	SDA4/TXD4
GND	GND	25	26	SCL4/SPI4 SCLK	SCL4/SPI4 SCLK
GPI00	SPI3 CEO N/TXD2/SDA6	27	28	SPI3 MISO/SCL6/RXD2	SPI3 MISO/SCL6/RXD2
GPI05	SPI4 MISO/RXD3/SCL3	29	30	GND	GND
GPI06	SPI4 MOSI/SDA4	31	32	SDA5/SPI5 CEO N/TXD5	SDA5/SPI5 CEO N/TXD5
GPI013	SPI5 MISO/RXD5/SCL5	33	34	GND	GND
GPI019	SPI6 MISO	35	36	SPI1 CE2 N	SPI1 CE2 N
GPI026	SPI5 CE1 N	37	38	SPI6 MOSI	SPI6 MOSI
GND	GND	39	40	SPI6 SCLK	SPI6 SCLK
	I2C			Ground	Ground
	UART			5V Power	5V Power
	SPI			3V3 Power	3V3 Power

Hình 1. 4 Sơ đồ chân Raspberry Pi 4 mode B

Raspberry Pi 4 có thể sử dụng trong hệ thống nhúng bên ngoài để giao tiếp tín hiệu. Có tổng cộng 40 chân, trong đó 28 chân là chân GPIO và các chân còn lại là chân nguồn. Các chân GPIO không chỉ thực hiện các chức năng I / O đơn giản mà còn hỗ trợ giao thức UART, SPI và I2C. Các giao thức này dành riêng cho mọi chân và tất cả chức năng của chúng được thảo luận dưới đây:

Chân cấp nguồn

Power In: Trong Raspberry pi, có hai cách cấp nguồn, một là từ cổng nguồn USB-C và thứ hai là từ các chân 5V. Chân 5V được kết nối trực tiếp với cổng adapter USB-C.

Đầu vào ở chân 5V phải ổn định và theo đúng thông số kỹ thuật. Trong trường hợp có điện áp cao hơn, thiết bị có thể bị cháy.

Các chân đầu vào 5V sẽ không có cầu chì và bộ điều chỉnh điện áp khi được sử dụng làm đầu vào cấp nguồn, do đó nguồn điện 5V phải theo đúng thông số kỹ thuật để tránh hư hại. Chân đầu vào cấp nguồn cho Raspberry Pi 4 được cung cấp bên dưới:

- Từ chân 2 đến 6 -> + 5V
- Chân 6 —> GND

Power Out: Có hai loại chân nguồn ra trong Raspberry pi 4 là 3V3 và 5V. 5V được kết nối trực tiếp với cổng USB nhưng 3V3 được kết nối thông qua bộ điều chỉnh điện áp cho ra đầu ra 3V ổn định. Tất cả các chân nguồn ra được cung cấp bên dưới:

- 3V3 - Chân 1, chân 17
- 5V - Chân 2, chân 6

Ground: Raspberry Pi 4 có nhiều chân ground được kết nối bên trong và các chân này này có thể làm điểm nối đất chung cho nguồn điện hoặc thiết bị bên ngoài. Danh sách các chân ground được đưa ra dưới đây:

- Chân 6
- Chân 9
- Chân 14
- Chân 20
- Chân 25
- Chân 30
- Chân 34
- Chân 39

Chân I/O digital

Hầu hết mọi thiết bị đều cần các chân đầu vào và đầu ra để giao tiếp. Trong thiết bị này có 28 chân GPIO được sử dụng làm đầu vào và đầu ra digital. Các chân GPIO trong bộ điều khiển có một số giá trị mặc định.

Các chân GPIO từ 0-9 sẽ ở trạng thái logic cao và từ 10 trở lên các chân sẽ ở trạng thái logic thấp. Tất cả các chân đó trong Raspberry Pi 4 đều được cung cấp bên dưới:

- GPIO0 - Chân 27
- GPIO1 - Chân 28

- GPIO2 - Chân 3
- GPIO3 - Chân 5
- GPIO4 - Chân 7
- GPIO5 - Chân 29
- GPIO6 - Chân 31
- GPIO7 - Chân 26
- GPIO8 - Chân 24
- GPIO9 - Chân 21
- GPIO10 - Chân 19
- GPIO11 - Chân 23
- GPIO12 - Chân 32
- GPIO13 - Chân 33
- GPIO14 - Chân 8
- GPIO15 - Chân 10
- GPIO16 - Chân 36
- GPIO17 - Chân 11
- GPIO18 - Chân 12
- GPIO19 - Chân 35
- GPIO20 - Chân 38
- GPIO21 - Chân 40
- GPIO22 - Chân 15
- GPIO23 - Chân 16
- GPIO24 - Chân 18
- GPIO25 - Chân 22
- GPIO26 - Chân 37
- GPIO27 - Chân 13

Tất cả các chân GPIO trong Raspberry Pi 4 không có chức năng đầu vào-đầu ra. Mỗi chân GPIO có thể thực hiện các chức năng khác thông qua lập trình.

Module giao tiếp dữ liệu nối tiếp Raspberry Pi

Các chân UART trong Raspberry Pi

Có nhiều giao thức nối tiếp và UART là một trong số đó. Nó khá phổ biến vì có hệ thống giao tiếp đơn giản và phụ thuộc vào hầu hết các phần mềm. Có nhiều chân giao tiếp UART trong Raspberry pi 4 được đưa ra bên dưới:

- TXD1 - GPIO14 - Chân 8
- RXD1 - GPIO15 - Chân 10
- TXD2 - GPIO0 - Chân 27
- RXD2 - GPIO1 - Chân 28
- TXD3 - GPIO5 - Chân 29
- RXD3 - GPIO4 - Chân 7
- TXD4 - GPIO8 - Chân 24
- RXD4 - GPIO9 - Chân 21
- TXD5 - GPIO12 - Chân 32
- RXD5 - GPIO13 - Chân 33

Chân giao tiếp SPI

Một số thiết bị sử dụng giao thức SPI và nó giúp điều khiển nhiều thiết bị bằng cách sử dụng 1 đường truyền dữ liệu duy nhất. Trong Raspberry pi 4 có nhiều chân SPI được sử dụng cho giao tiếp SPI. Chân SPI của Raspberry Pi 4 được cung cấp bên dưới:

- SPI3 CEO N - GPIO0 - Chân 27
- SPI3 MISO - GPIO1 - Chân 28
- SPI3 MOSI - GPIO2 - Chân 3
- SPI3 SCLK - GPIO3 - Chân 5
- SPI4 CEO N - GPIO4 - Chân 7
- SPI4 MISO - GPIO5 - Chân 29
- SPI4 MOSI - GPIO6 - Chân 31
- SPI4 SCLK - GPIO7 - Chân 26
- SPI0 CE1 N - GPIO8 - Chân 24
- SPI0 CE0 N - GPIO9 - Chân 21
- SPI0 MISO - GPIO10 - Chân 19

- SPI0 MOSI - GPIO11 - Chân 23
- SPI5 CEO N / SPI0 SCLK - GPIO12 - Chân 32
- SPI5 MISO - GPIO13 - Chân 33
- SPI5 MOSI - GPIO14 - Chân 8
- SPI5 SCLK - GPIO15 - Chân 10
- CTS0 - GPIO16 - Chân 36
- RTS0 - GPIO17 - Chân 11
- SPI6 CEO N - GPIO18 - Chân 12
- SPI6 MISO - GPIO19 - Chân 35
- SPI6 MOSI - GPIO20 - Chân 38
- SPI6 SCLK - GPIO21 - Chân 40

Chân giao tiếp I2C

Raspberry Pi 4 có hỗ trợ giao thức I2C. Là giao thức được sử dụng ở một số cảm biến và động cơ. Tất cả các chân này được đưa ra bên dưới:

- SDA0 / SDA6 - GPIO0 - Chân 27
- SCL0 / SCL6 - GPIO1 - Chân 28
- SDA1 / SDA3 - GPIO2 - Chân 3
- SCL1 / SCL3 - GPIO3 - Chân 5
- SDA3 - GPIO4 - Chân 7
- SCL3 - GPIO5 - Chân 29
- SDA4 - GPIO6 - Chân 31
- SCL4 - GPIO7 - Chân 26
- SDA4 - GPIO8 - Chân 24
- SCL4 - GPIO9 - Chân 21
- SDA5 - GPIO10 - Chân 19
- SCL5 - GPIO11 - Chân 23
- SDA5 - GPIO12 - Chân 32
- SCL5 - GPIO13 - Chân 33
- SDA6 - GPIO22 - Chân 15

- SCL6 - GPIO23 - Chân 16

Các chân GPIO PWM

Để tạo ra tín hiệu đầu ra xung mong muốn Raspberry Pi 4 có một số chân PWM. Các chân này cấp dữ liệu trực tiếp cho các thiết bị ngoại vi điện áp thấp. Để tạo ra tín hiệu đầu tiên các chân phải được lập trình trước. Tất cả các chân PWM được đưa ra bên dưới:

- PWM0 - GPIO12 - Chân 32
- PWM1 - GPIO13 - Chân 33
- PWM0 - GPIO18 - Chân 12
- PWM1 - GPIO19 - Chân 35

Các chân SDIO Raspberry Pi

Trong Raspberry Pi 4 có một khe cắm thẻ SD nhưng các cgaab GPIO cũng hỗ trợ khả năng tương thích với thẻ SD. Chân SDIO trên thiết bị có thể được sử dụng cho thẻ SD khi có ứng dụng yêu cầu:

- SD0CLK / SD1 CLK - GPIO22 - Chân 15
- SD0 CMD / SD1 CMD - GPIO23 - Chân 16
- SD0 DATA0 / SD1 DAT0 - GPIO24 - Chân 18
- SD0 DAT1 / SD1 DAT1 - GPIO25 - Chân 22
- SD1 DAT2 / SD1 DAT2 - GPIO26 - Chân 37
- SD0 DAT3 / SD1 DAT3 - GPIO27 - Chân 13

1.1.3. Cảm biến đo nhiệt độ độ ẩm DHT22



Hình 1. 5 Cảm Biến DHT22

DHT22 là một cảm biến nhiệt độ và độ ẩm kỹ thuật số có độ chính xác cao, được tích hợp trong hệ thống đèn thông minh để cung cấp thông tin về môi trường xung quanh. Với thiết kế nhỏ gọn, dễ sử dụng và giá thành hợp lý, DHT22 là lựa chọn phổ biến cho các ứng dụng IoT và tự động hóa.

Thông số kỹ thuật:

- Điện áp hoạt động: 3.3V - 5V
- Dòng tiêu thụ: 2.5mA (tối đa khi truyền dữ liệu)
- Phạm vi đo nhiệt độ: -40°C đến 80°C
- Độ chính xác nhiệt độ: $\pm 0.5^{\circ}\text{C}$
- Phạm vi đo độ ẩm: 0-100% RH

- Độ chính xác độ ẩm: 2-5% RH
- Tần số lấy mẫu: 0.5Hz (2 giây/lần)
- Giao tiếp: Giao tiếp 1 dây (single-wire)

Nguyên lý hoạt động: DHT22 sử dụng một cảm biến độ ẩm điện dung và một thermistor (cảm biến nhiệt độ) để đo lường các thông số môi trường. Dữ liệu được chuyển đổi thành tín hiệu kỹ thuật số và truyền qua giao tiếp 1 dây tới Raspberry Pi.

Ưu điểm:

- Độ chính xác cao: Cung cấp dữ liệu nhiệt độ và độ ẩm với độ chính xác tốt, đáp ứng yêu cầu của nhiều ứng dụng.
- Dễ sử dụng: Giao tiếp 1 dây đơn giản, dễ dàng kết nối và tích hợp với Raspberry Pi.
- Giá thành hợp lý: DHT22 có giá thành phải chăng, phù hợp với các dự án tự làm và ứng dụng thương mại.

Ứng dụng trong hệ thống đèn thông minh:

- Hiển thị thông tin môi trường: DHT22 cung cấp dữ liệu nhiệt độ và độ ẩm để hiển thị trên màn hình LCD I2C, giúp người dùng theo dõi môi trường xung quanh.
- Lưu trữ dữ liệu: Dữ liệu từ DHT22 được lưu vào cơ sở dữ liệu SQL, phục vụ cho việc phân tích, báo cáo và đưa ra các quyết định điều khiển tự động.
- Điều khiển đèn tự động: Dựa vào dữ liệu nhiệt độ và độ ẩm, hệ thống có thể tự động điều chỉnh độ sáng và màu sắc của đèn LED RGB để tạo ra môi trường chiếu sáng thoải mái và tiết kiệm năng lượng. Ví dụ, đèn có thể tự động điều chỉnh ánh sáng ấm hơn khi nhiệt độ giảm hoặc điều chỉnh độ sáng giảm khi độ ẩm tăng.
- Cảnh báo môi trường: Hệ thống có thể gửi cảnh báo đến người dùng khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng cho phép.

Cảm biến DHT22 đóng vai trò quan trọng trong việc thu thập dữ liệu môi trường, giúp hệ thống đèn thông minh trở nên thông minh và tiện ích hơn. Với khả năng đo lường chính xác, dễ sử dụng và giá thành hợp lý, DHT22 là một lựa chọn tối ưu cho các ứng

dụng IoT và tự động hóa.

1.1.4. LCD I2C

Module LCD I2C: Cửa sổ giao tiếp của hệ thống đèn thông minh



Hình 1. 6 Module LCD I2C

Module LCD I2C là một màn hình tinh thể lỏng (LCD) được tích hợp thêm chip điều khiển giao tiếp I2C, giúp đơn giản hóa việc kết nối và điều khiển so với các loại LCD truyền thống. Trong hệ thống đèn thông minh, module LCD I2C đóng vai trò như một "cửa sổ" giao tiếp, hiển thị thông tin về trạng thái hoạt động của đèn, nhiệt độ, độ ẩm và các thông số khác.

Đặc điểm nổi bật:

- Giao tiếp I2C: Sử dụng giao thức I2C (Inter-Integrated Circuit) để giao tiếp với Raspberry Pi, chỉ cần 2 dây SDA (Serial Data) và SCL (Serial Clock) để truyền dữ liệu, giúp tiết kiệm chân GPIO và đơn giản hóa kết nối.

- Kích thước đa dạng: Có nhiều loại module LCD I2C với kích thước khác nhau, phổ biến là 16x2 (16 cột và 2 dòng) và 20x4 (20 cột và 4 dòng), tùy thuộc vào nhu cầu hiển thị thông tin của hệ thống.
- Hiển thị ký tự và đồ họa: Hỗ trợ hiển thị ký tự chữ và số, cũng như các ký tự đặc biệt và đồ họa đơn giản.
- Điều khiển độ sáng: Có thể điều chỉnh độ sáng của màn hình bằng cách thay đổi điện áp cấp cho đèn nền.

Ứng dụng trong hệ thống đèn thông minh:

- Hiển thị thông số môi trường: Hiển thị nhiệt độ và độ ẩm đo được từ cảm biến DHT22, giúp người dùng theo dõi môi trường xung quanh.
- Tạo giao diện người dùng: Module LCD I2C có thể được sử dụng để tạo một giao diện người dùng đơn giản, cho phép người dùng tương tác với hệ thống thông qua các nút bấm hoặc bộ mã hóa quay.

Module LCD I2C là một thành phần quan trọng trong hệ thống đèn thông minh, giúp hiển thị thông tin và tạo giao diện người dùng một cách đơn giản và hiệu quả. Với khả năng giao tiếp I2C, kích thước đa dạng và thư viện hỗ trợ sẵn có, module LCD I2C là một lựa chọn tối ưu cho các ứng dụng nhúng.

1.1.5. Module LED RGB 5050 SMD KY-009

Module LED RGB 5050 SMD KY-009: Bảng màu đa sắc cho hệ thống đèn thông minh



Hình 1. 7 Module LED RGB 5050 SMD KY-009

Module LED RGB 5050 SMD KY-009 là một module điều khiển đèn LED RGB (Đỏ - Xanh lá - Xanh dương) được tích hợp sẵn trên một mạch in nhỏ gọn. Module này sử dụng một đèn LED SMD 5050, loại đèn có kích thước 5.0mm x 5.0mm, cho khả năng phát ra ánh sáng với nhiều màu sắc khác nhau bằng cách kết hợp ba màu cơ bản.

Đặc điểm nổi bật:

- LED SMD 5050: Đèn LED SMD 5050 có độ sáng cao và góc chiếu rộng, tạo ra ánh sáng đều và mạnh mẽ.
- Điều khiển PWM: Module sử dụng kỹ thuật điều chế độ rộng xung (PWM - Pulse Width Modulation) để điều khiển độ sáng của từng màu LED, từ đó tạo ra hàng triệu màu sắc khác nhau.
- Dễ dàng kết nối: Module có 4 chân kết nối: VCC, GND và 3 chân điều khiển màu sắc (R, G, B), tương thích với các chân GPIO của Raspberry Pi.
- Thư viện hỗ trợ: Có sẵn các thư viện lập trình cho Raspberry Pi (như RPi.GPIO) giúp dễ dàng điều khiển module LED RGB.

Ứng dụng trong hệ thống đèn thông minh:

- Tạo ánh sáng màu sắc đa dạng: Module LED RGB cho phép tạo ra hàng triệu màu sắc khác nhau, mang đến trải nghiệm chiếu sáng đa dạng và phong phú cho người dùng.
- Điều khiển màu sắc từ xa: Kết hợp với giao diện web, người dùng có thể dễ dàng thay đổi màu sắc của đèn theo ý muốn.
- Tạo hiệu ứng ánh sáng: Module có thể được lập trình để tạo ra các hiệu ứng ánh sáng động như nhấp nháy, chuyển màu, thay đổi độ sáng...
- Tạo môi trường ánh sáng phù hợp: Với khả năng điều chỉnh màu sắc linh hoạt, đèn thông minh có thể tạo ra môi trường ánh sáng phù hợp với từng hoạt động và tâm trạng của người dùng. Ví dụ, ánh sáng ấm áp cho buổi tối thư giãn, ánh sáng trắng sáng cho công việc tập trung...

Module LED RGB 5050 SMD KY-009 là một thành phần quan trọng trong hệ thống đèn thông minh, mang đến khả năng tạo ra ánh sáng màu sắc đa dạng và điều khiển từ

xa. Với thiết kế nhỏ gọn, dễ sử dụng và thư viện hỗ trợ sẵn có, module này là một lựa chọn lý tưởng cho các ứng dụng chiếu sáng thông minh và sáng tạo.

1.2. Webserver (Apache)

Apache: Cầu nối giữa người dùng và hệ thống đèn thông minh



Hình 1. 8 Giao thức UDP

Apache là một trong những web server phổ biến và mạnh mẽ nhất thế giới, được sử dụng rộng rãi để phục vụ các trang web và ứng dụng web. Trong hệ thống đèn thông minh, Apache đóng vai trò là cầu nối giữa người dùng và Raspberry Pi, cho phép người dùng tương tác và điều khiển đèn thông qua giao diện web.

Đặc điểm nổi bật:

- Mã nguồn mở: Apache là phần mềm mã nguồn mở, miễn phí và được phát triển bởi cộng đồng Apache Software Foundation. Điều này đảm bảo tính minh bạch, an toàn và khả năng tùy biến cao.
- Đa nền tảng: Apache có thể chạy trên nhiều hệ điều hành khác nhau như Linux, Windows và macOS, mang lại sự linh hoạt trong việc triển khai.
- Hiệu năng cao: Apache được tối ưu hóa để xử lý hàng nghìn yêu cầu đồng thời, đảm bảo khả năng phục vụ mượt mà cho các ứng dụng web có lưu lượng truy cập lớn.
- Module mở rộng: Apache hỗ trợ một hệ thống module phong phú, cho phép mở rộng chức năng của web server để đáp ứng các yêu cầu cụ thể của ứng dụng.

- Cộng đồng hỗ trợ lớn: Apache có một cộng đồng người dùng và nhà phát triển đông đảo, sẵn sàng hỗ trợ và chia sẻ kiến thức.

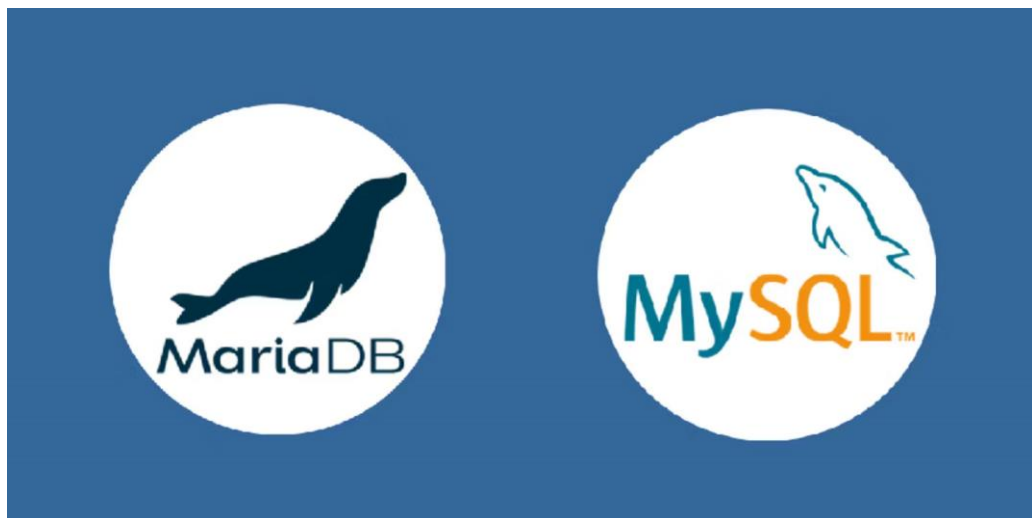
Ứng dụng trong hệ thống đèn thông minh:

- Phục vụ giao diện web: Apache chịu trách nhiệm nhận các yêu cầu từ trình duyệt web của người dùng và gửi lại các trang web điều khiển đèn.
- Xử lý dữ liệu: Apache có thể xử lý các dữ liệu được gửi từ giao diện web, ví dụ như lệnh bật/tắt đèn, thay đổi màu sắc, điều chỉnh độ sáng...
- Tương tác với Raspberry Pi: Apache có thể giao tiếp với Raspberry Pi thông qua các cơ chế như CGI (Common Gateway Interface) hoặc WSGI (Web Server Gateway Interface) để gửi các lệnh điều khiển và nhận dữ liệu từ Raspberry Pi.
- Bảo mật: Apache cung cấp các tính năng bảo mật như xác thực người dùng, mã hóa dữ liệu và kiểm soát truy cập, giúp bảo vệ hệ thống đèn thông minh khỏi các truy cập trái phép.

Apache là một web server mạnh mẽ và linh hoạt, đóng vai trò quan trọng trong việc kết nối người dùng với hệ thống đèn thông minh thông qua giao diện web. Với sự hỗ trợ của Apache, việc điều khiển và quản lý đèn trở nên dễ dàng và thuận tiện hơn bao giờ hết.

1.3. Cơ sở dữ liệu MySQL (MariaDB)

MySQL: Kho lưu trữ thông tin của hệ thống đèn thông minh



Hình 1. 9 SQL Server

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến, được sử dụng rộng rãi trong các ứng dụng web và nhúng. Trong hệ thống đèn thông minh, MySQL đóng vai trò là kho lưu trữ thông tin, cho phép lưu trữ, truy xuất và quản lý dữ liệu liên quan đến hoạt động của đèn.

Vai trò:

- Lưu trữ dữ liệu: MySQL lưu trữ các thông tin quan trọng của hệ thống như:
- Trạng thái đèn: Bật/tắt, độ sáng, màu sắc RGB hiện tại.
- Dữ liệu cảm biến: Nhiệt độ và độ ẩm đo được từ cảm biến DHT22 theo thời gian.
- Cài đặt người dùng: Các tùy chọn cấu hình của người dùng, ví dụ như lịch trình bật/tắt đèn, màu sắc yêu thích...
- Truy xuất dữ liệu: Cho phép các thành phần khác của hệ thống (như Raspberry Pi, giao diện web) truy xuất dữ liệu để hiển thị, phân tích hoặc điều khiển đèn.
- Quản lý dữ liệu: Cung cấp các công cụ để quản lý dữ liệu như thêm, sửa, xóa, tìm kiếm và sắp xếp.

Cấu trúc cơ sở dữ liệu:

Trong hệ thống đèn thông minh, chúng ta có thể thiết kế một cơ sở dữ liệu đơn giản với các bảng sau:

Bảng `led_rgb`: Lưu trữ trạng thái hiện tại của đèn.

- `id` (INT, PRIMARY KEY): ID của bản ghi.
- `red` (INT): Giá trị màu đỏ (0-255).
- `green` (INT): Giá trị màu xanh lá (0-255).
- `blue` (INT): Giá trị màu xanh dương (0-255).
- `timecreate` (DATETIME): Thời gian cập nhật trạng thái.

Bảng `dht22_readings`: Lưu trữ dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22.

- `id` (INT, PRIMARY KEY): ID của bản ghi.
- `temperature` (FLOAT): Nhiệt độ đo được (°C).
- `humidity` (FLOAT): Độ ẩm đo được (%).
- `timecreate` (DATETIME): Thời gian đo.

Tương tác với cơ sở dữ liệu:

PHP sẽ được sử dụng để kết nối và tương tác với cơ sở dữ liệu MySQL. Các thao tác cơ bản bao gồm:

- Kết nối đến cơ sở dữ liệu.
- Thực hiện các truy vấn SQL (SELECT, INSERT) để đọc, ghi, cập nhật và xóa dữ liệu.
- Đóng kết nối.

1.6. Kết luận chương

Chương 1 đã cung cấp một cái nhìn tổng quan về các thành phần và công nghệ cốt lõi sẽ được sử dụng để xây dựng hệ thống đèn thông minh. Chúng ta đã tìm hiểu về khái niệm hệ thống nhúng, một hệ thống máy tính chuyên dụng được thiết kế để thực hiện các nhiệm vụ cụ thể. Raspberry Pi 4 Model B, với khả năng xử lý mạnh mẽ và tính linh hoạt, được chọn làm trái tim của hệ thống. Cảm biến DHT22 sẽ đóng vai trò như "đôi mắt" của hệ thống, cung cấp thông tin về nhiệt độ và độ ẩm môi trường xung quanh. Module LCD I2C sẽ hiển thị các thông số này, cũng như trạng thái hoạt động của đèn, giúp người dùng dễ dàng theo dõi. Module LED RGB 5050 SMD KY-009 sẽ mang đến khả năng tùy biến màu sắc ánh sáng, tạo nên không gian sống đa dạng và phù hợp với nhu cầu của người dùng. Webserver Apache sẽ là cầu nối giữa người dùng và hệ thống, cho phép điều khiển đèn từ xa thông qua giao diện web. Cuối cùng, cơ sở dữ liệu MySQL sẽ lưu trữ mọi thông tin cần thiết, từ trạng thái đèn, dữ liệu cảm biến đến cài đặt người dùng. Với những kiến thức nền tảng này, chúng ta đã sẵn sàng để bước vào giai đoạn thiết kế và triển khai hệ thống đèn thông minh.

CHƯƠNG II: THIẾT KẾ HỆ THỐNG

2.1. Mô tả hệ thống

2.1.1. Giới thiệu

Hệ thống đèn thông minh được thiết kế nhằm mục đích cung cấp một giải pháp chiếu sáng hiện đại, tiện nghi và tiết kiệm năng lượng. Hệ thống này bao gồm các thành phần phần cứng và phần mềm được tích hợp chặt chẽ, cho phép người dùng điều khiển đèn từ xa thông qua giao diện web, tùy chỉnh màu sắc và độ sáng theo ý muốn, đồng thời theo dõi các thông số môi trường như nhiệt độ và độ ẩm.

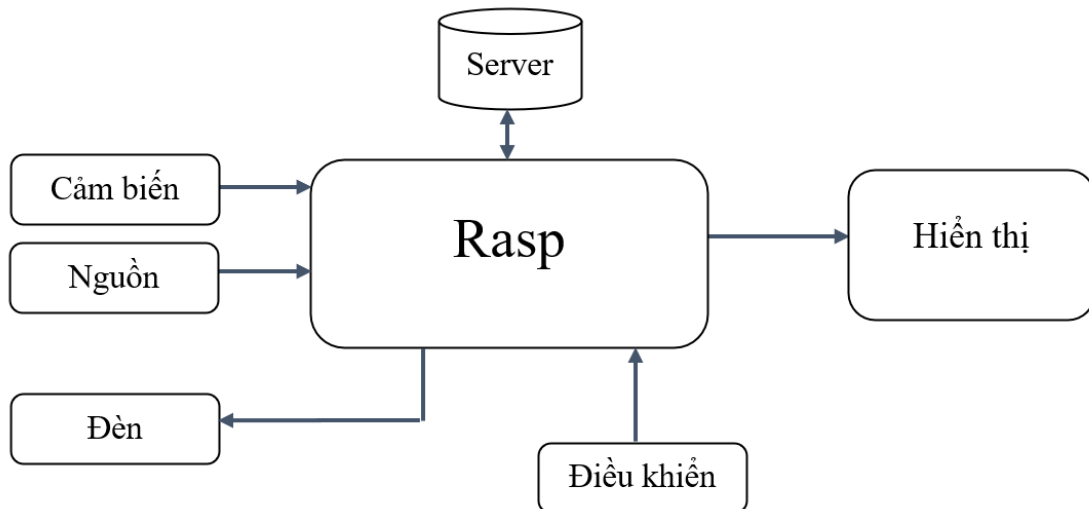
2.1.2. Tính năng

- Điều khiển từ xa: Người dùng có thể bật/tắt đèn, điều chỉnh độ sáng và màu sắc từ bất kỳ đâu thông qua giao diện web trên máy tính, điện thoại hoặc máy tính bảng.
- Hiển thị thông tin môi trường: Hệ thống hiển thị nhiệt độ và độ ẩm môi trường trên màn hình LCD, giúp người dùng theo dõi và điều chỉnh môi trường sống.
- Lưu trữ dữ liệu: Dữ liệu về trạng thái đèn, nhiệt độ và độ ẩm được lưu trữ trong cơ sở dữ liệu, cho phép người dùng xem lại lịch sử và phân tích xu hướng.
- Tùy chỉnh linh hoạt: Người dùng có thể tùy chỉnh màu sắc ánh sáng theo ý muốn, tạo ra không gian sống phù hợp với tâm trạng và hoạt động.

2.1.3. Phạm vi áp dụng

Phạm vi của hệ đèn thông minh của nhóm chúng có thể áp dụng ở trong phòng ngủ với mức ánh sáng vừa đủ hoặc một không gian có diện tích gần như phòng ngủ để có được hiệu quả chiếu sáng tốt nhất.

2.2. Sơ đồ khối hệ thống



Hình 2. 1 Sơ đồ khối hệ thống

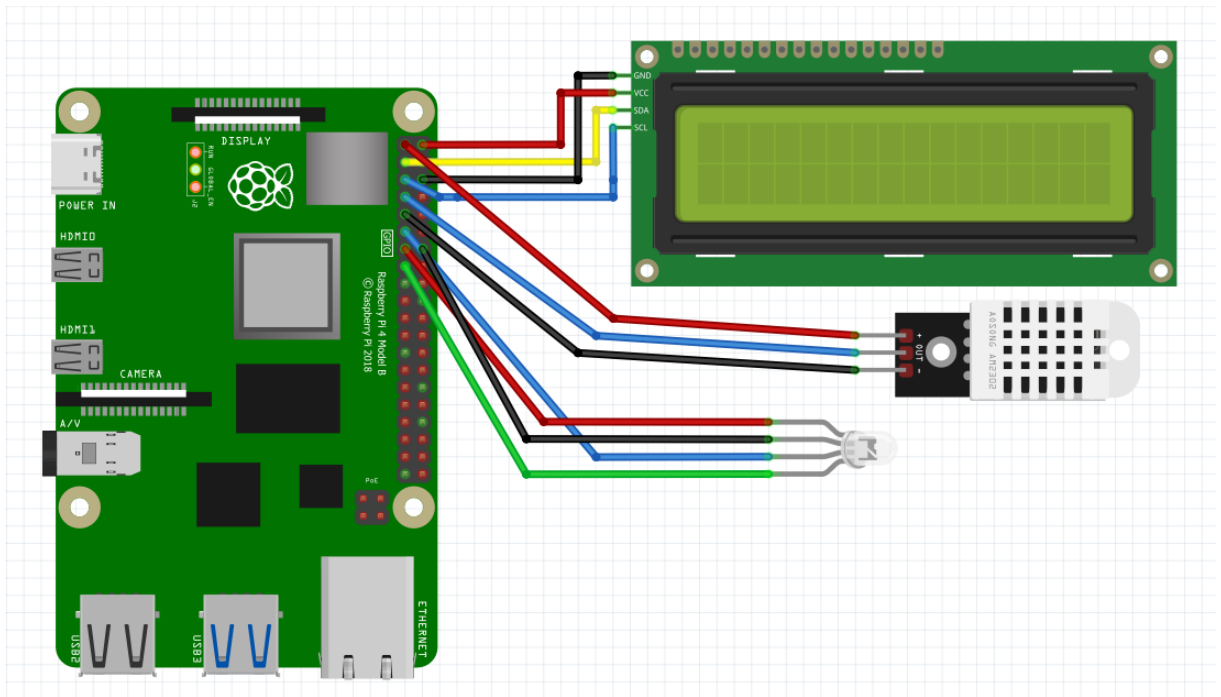
Hệ thống đèn thông minh được xây dựng dựa trên nền tảng Raspberry Pi 4 Model B, kết hợp với các thành phần phần cứng như cảm biến DHT22, màn hình LCD I2C và module LED RGB. Phần mềm điều khiển được viết bằng Python và C, giao tiếp với các thành phần phần cứng thông qua các thư viện và giao thức tương ứng. Giao diện web được xây dựng bằng HTML, CSS, JavaScript và PHP, cho phép người dùng tương tác với hệ thống từ xa.

Các thành phần chính:

- Raspberry Pi 4 Model B: Bộ não của hệ thống, xử lý dữ liệu, điều khiển các thành phần khác và giao tiếp với giao diện web.
- Cảm biến DHT22: Đo nhiệt độ và độ ẩm môi trường.
- LCD I2C: Hiển thị thông tin trạng thái môi trường.
- Module LED RGB: Cung cấp ánh sáng với màu sắc tùy chỉnh.
- Webserver Apache: Phục vụ giao diện web hiển thị nhiệt độ - độ ẩm và điều khiển đèn.
- Cơ sở dữ liệu MySQL: Lưu trữ dữ liệu về trạng thái đèn và nhiệt độ, độ ẩm

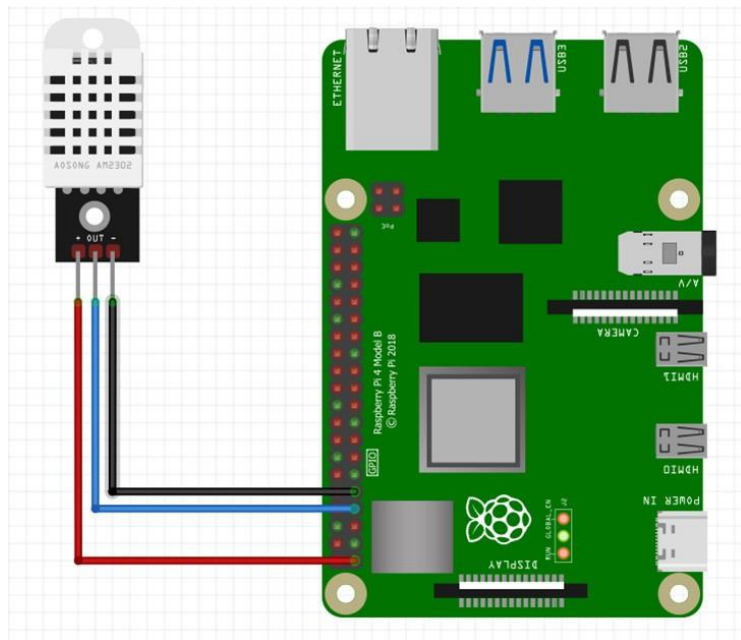
2.3. Sơ đồ mạch điện và giải thuật

Sơ đồ mạch điện tổng quát:



Hình 2. 2 Sơ đồ tổng quát

2.3.1. Kết nối Raspberry Pi 4 với DHT22

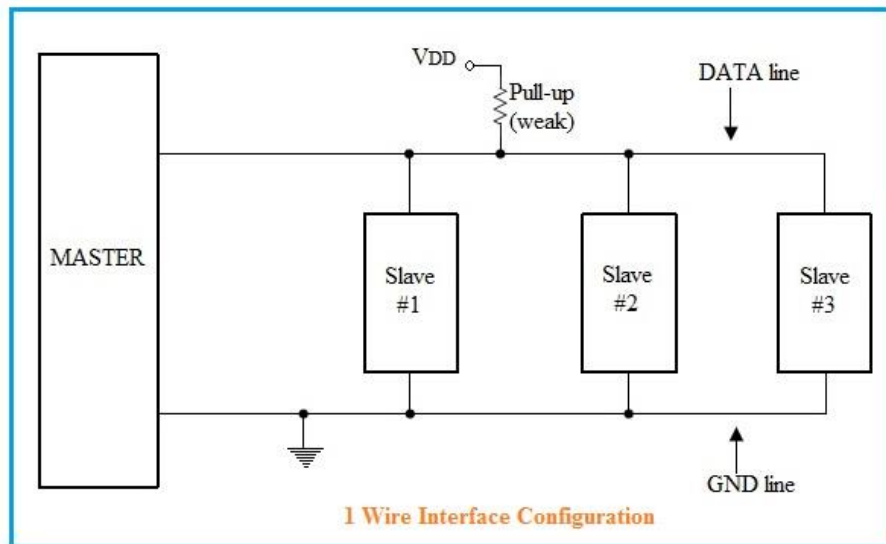


Hình 2. 3 Raspberry và DHT22

Sơ đồ chân kết nối:

- (1) – VDD – dây đỏ
- GPIO 4 (7) – DATA – dây xanh
- (9) – GND – dây đen

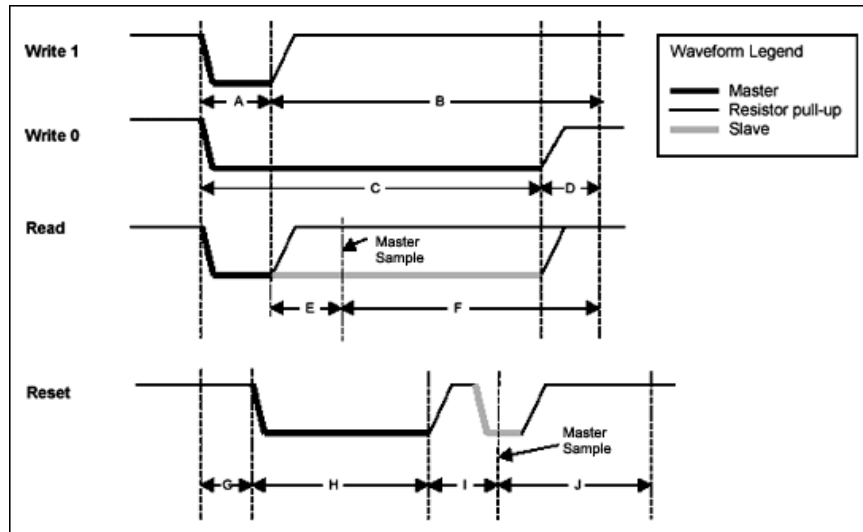
Giao thức truyền dữ liệu từ DHT22 tới Raspberry Pi 4 là giao thức Single-Wire Two-Way (SWTW)



Hình 2. 4 Single-Wire Two-Way (SWTW)

Nguyên lý giao thức SWTW:

- Các tín hiệu sử dụng là Reset, 0 write, 1 write, Read.
- Write 1: Truyền đi bit 1: Master kéo xuống mức 0 khoảng A(us) rồi về mức 1 khoảng B lần
- Write 0: truyền đi bit 0: Master kéo xuống 0 khoảng C rồi trả về 1 khoảng D
- Read: Đọc một Bit: Master kéo xuống 0 khoảng A rồi trả về 1. delay khoảng E rồi đọc giá trị slave gửi về. delay F.
- Restart: Chuẩn bị giao tiếp. Master kéo xuống 0 một khoảng H rồi nhả lên mức 1 sau đó cấu hình Master là chân In delay 1(us) rồi đọc giá trị slave trả về. Nếu = 0 thì cho phép giao tiếp =1 đường truyền lỗi hoặc slave đang bận.



Hình 2. 5 Nguyên lý SWTW

Giải thuật đọc và gửi dữ liệu cảm biến DHT22 lên SQL

Giải thuật này tập trung vào việc thu thập dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22, sau đó lưu trữ chúng vào cơ sở dữ liệu MySQL để phục vụ cho việc phân tích, hiển thị và điều khiển hệ thống đèn thông minh.

Chuẩn bị:

- Chương trình bắt đầu bằng việc nhập các thư viện cần thiết để làm việc với cảm biến DHT22, giao tiếp với cơ sở dữ liệu MySQL và điều khiển thời gian.
- Cảm biến DHT22 được kết nối với Raspberry Pi qua chân GPIO số 4.

Đọc dữ liệu từ cảm biến:

- Chương trình liên tục thực hiện vòng lặp vô hạn để đọc dữ liệu nhiệt độ và độ ẩm từ cảm biến DHT22.
- Nếu đọc thành công, các giá trị nhiệt độ và độ ẩm sẽ được in ra màn hình console.
- Nếu có lỗi xảy ra trong quá trình đọc, chương trình sẽ in ra thông báo lỗi và đợi một khoảng thời gian ngắn trước khi thử lại.

Kết nối đến cơ sở dữ liệu MySQL: Chương trình thiết lập kết nối đến cơ sở dữ liệu MySQL trên máy chủ localhost với tên người dùng msi, mật khẩu 123456 và tên cơ sở dữ liệu sensor.

Lưu dữ liệu vào cơ sở dữ liệu:

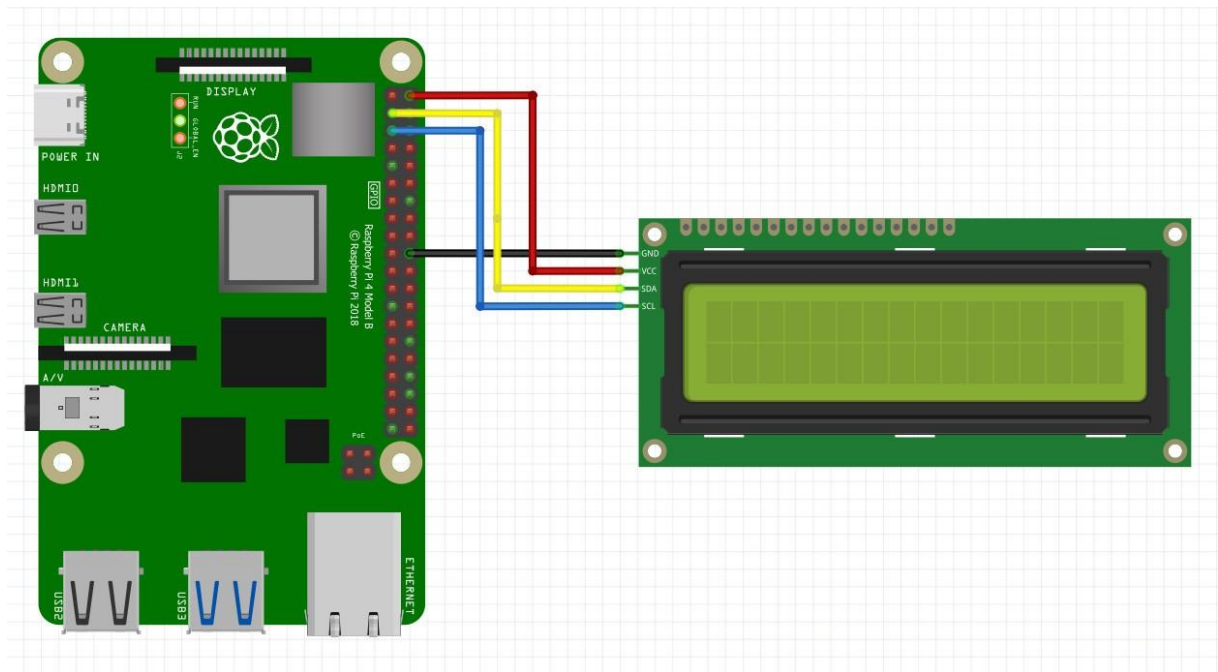
- Chương trình tạo một con trỏ để thực hiện các truy vấn SQL.

- Truy vấn SQL INSERT INTO dht22_readings (temperature, humidity) VALUES (%s, %s) được sử dụng để chèn các giá trị nhiệt độ và độ ẩm vừa đọc được vào bảng dht22_readings trong cơ sở dữ liệu.
- Sau khi thực hiện truy vấn, chương trình xác nhận giao dịch để đảm bảo dữ liệu được lưu trữ thành công.

Đóng kết nối: Cuối cùng, chương trình đóng kết nối đến cơ sở dữ liệu MySQL.

Lặp lại: Chương trình tạm dừng trong 10 giây, sau đó quay lại bước 2 để tiếp tục đọc và lưu dữ liệu mới từ cảm biến.

2.3.2. Kết nối Raspberry Pi 4 với LCD I2C

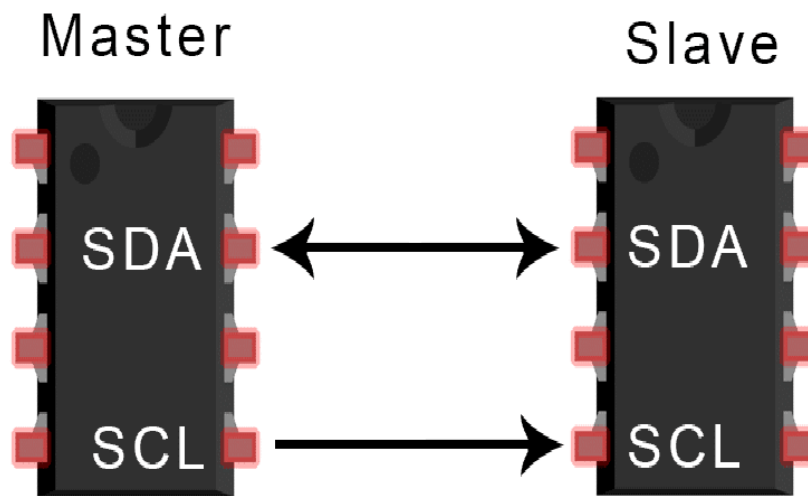


Hình 2. 6 Raspberry và LCD

Sơ đồ chân:

- (2) – VDD – dây đỏ
- GPIO 2 (3) – SDA – dây vàng
- GPIO 3 (5) – SCL – dây xanh
- (20) – GND – dây đen

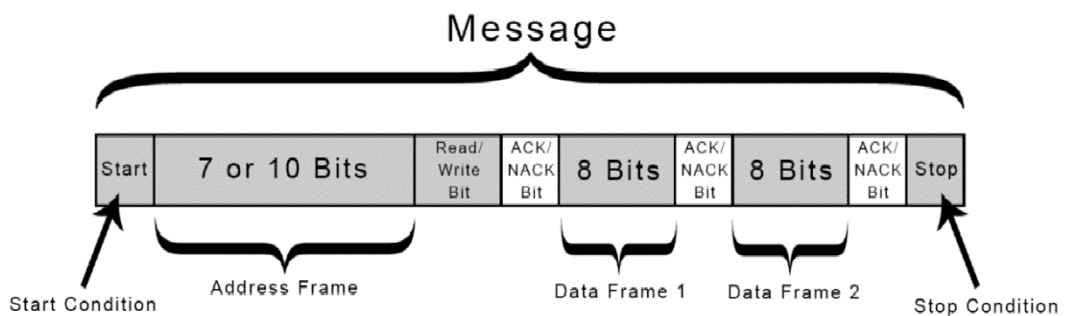
Giao thức truyền dữ liệu từ Raspberry Pi 4 tới màn hình LCD là giao thức Inter – Integrated Circuit (I2C)



Hình 2. 7 Inter – Integrated Circuit (I2C)

Để có thể sử dụng giao thức I2C cho Raspberry và LCD thì cần phải thực hiện các bước kết nối sau đây:

- Kích hoạt I2C bằng lệnh `sudo raspi-config`
- Kiểm tra module đã kết nối hay chưa bằng lệnh `lsmod | grep i2c`
- Cài các package cần thiết `sudo apt-get install i2c-tools` và `sudo apt-get install libi2c-dev`
- Tìm địa chỉ LCD bằng lệnh `i2cdetect -y 1`



Hình 2. 8 Step connect I2C with Raspberry

Nguyên lý giao thức I2C:

- Master gửi điều kiện khởi động đến slave được kết nối bằng cách chuyển đường SDA từ mức điện áp cao sang mức điện áp thấp trước khi chuyển đường SCL từ mức cao xuống mức thấp.

- Master gửi cho slave địa chỉ 7 hoặc 10 bit của slave mà nó muốn giao tiếp, cùng với bit đọc / ghi.
- Slave so sánh địa chỉ được gửi với địa chỉ của chính nó. Nếu địa chỉ trùng khớp, slave sẽ trả về một bit ACK bằng cách kéo dòng SDA xuống thấp cho một bit. Nếu địa chỉ từ master không khớp với địa chỉ của slave, slave rời khỏi đường SDA cao.
- Master gửi hoặc nhận khung dữ liệu. Sau khi mỗi khung dữ liệu được chuyển, thiết bị nhận trả về một bit ACK khác cho thiết bị gửi để xác nhận đã nhận thành công khung.
- Để dừng truyền dữ liệu, master gửi điều kiện dừng đến slave bằng cách chuyển đổi mức cao SCL trước khi chuyển mức cao SDA.

Giải thuật lấy và hiển thị dữ liệu từ SQL xuống LCD

Giải thuật này sẽ truy vấn cơ sở dữ liệu MySQL để lấy dữ liệu nhiệt độ và độ ẩm mới nhất từ bảng dht22_readings, sau đó hiển thị các giá trị này lên màn hình LCD I2C.

Khởi tạo kết nối MySQL:

- Sử dụng hàm `mysql_init()` để khởi tạo một kết nối đến MySQL.
- Cung cấp thông tin máy chủ (localhost), tên người dùng (msi), mật khẩu (123456) và tên cơ sở dữ liệu (sensor).
- Sử dụng hàm `mysql_real_connect()` để thiết lập kết nối thực tế đến MySQL. Nếu không thể kết nối, chương trình sẽ in ra thông báo lỗi và thoát.

Khởi tạo WiringPi và LCD:

- `wiringPiSetup()`: Khởi tạo thư viện WiringPi để làm việc với GPIO của Raspberry Pi.
- `wiringPiI2CSetup(I2C_ADDR)`: Thiết lập giao tiếp I2C với module LCD tại địa chỉ 0x27.
- `lcd_init()`: Khởi tạo LCD (xóa màn hình, đặt con trỏ về vị trí ban đầu, thiết lập chế độ hiển thị, v.v.).

Vòng lặp vô hạn: Chương trình liên tục thực hiện các bước sau trong vòng lặp `while(1)`:

Truy vấn dữ liệu từ MySQL:

- `mysql_query(conn, "SELECT temperature, humidity FROM dht22_readings`

ORDER BY timestamp DESC LIMIT 1"): Thực hiện truy vấn SQL để lấy giá trị nhiệt độ và độ ẩm mới nhất từ bảng dht22_readings. Kết quả truy vấn được sắp xếp theo thời gian giảm dần (ORDER BY timestamp DESC) và chỉ lấy một bản ghi (LIMIT 1).

Xử lý kết quả truy vấn:

- `mysql_store_result(conn)`: Lưu trữ kết quả truy vấn vào biến `res`.
- `mysql_fetch_row(res)`: Lấy một hàng dữ liệu từ kết quả truy vấn và lưu vào biến `row`.

Chuyển đổi dữ liệu và hiển thị lên LCD:

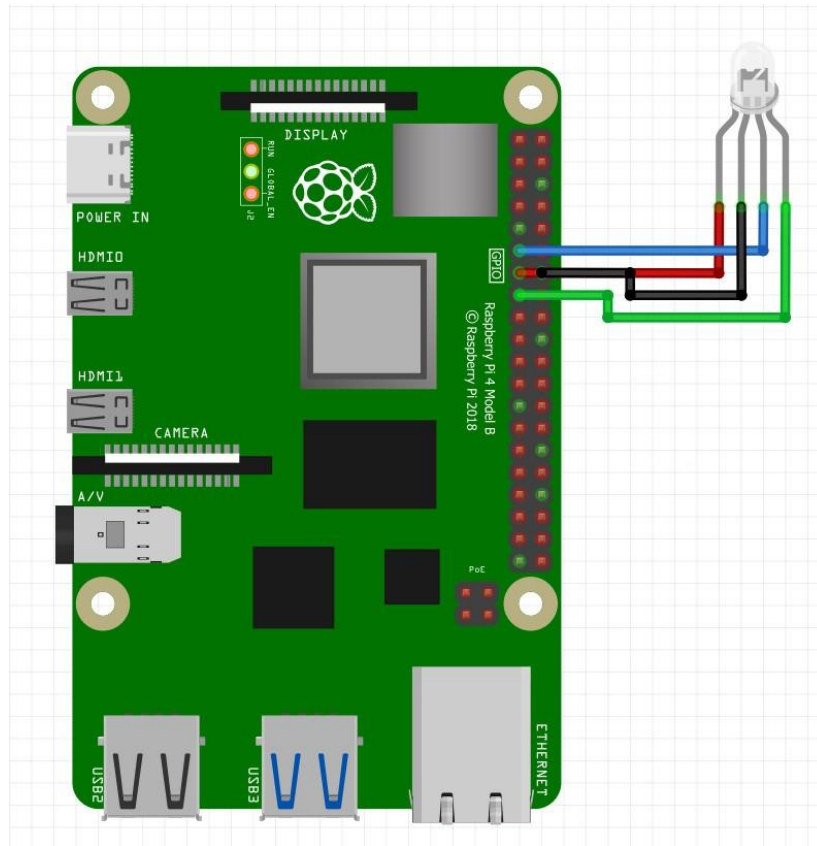
- `atof(row[0]), atof(row[1])`: Chuyển đổi các giá trị nhiệt độ và độ ẩm (kiểu chuỗi) từ `row` sang kiểu số thực (float).
- `lcdLoc(LINE1), lcdLoc(LINE2)`: Đặt con trỏ LCD ở dòng 1 và dòng 2.
- `println("Temp: "), println("Humi: ")`: Ghi các chuỗi "Temp: " và "Humi: " lên LCD.
- `typeFloat(temp), typeFloat(humi)`: Ghi các giá trị nhiệt độ và độ ẩm lên LCD.
- `println("*C"), println("%")`: Ghi các ký tự "*C" và "%" lên LCD.
- `delay(2000)`: Tạm dừng 2 giây để người dùng có thể xem dữ liệu trên LCD.

Xóa màn hình LCD: `ClrLcd()`: Xóa màn hình LCD và đưa con trỏ về vị trí ban đầu.

Giải phóng bộ nhớ và lặp lại:

- `mysql_free_result(res)`: Giải phóng bộ nhớ đã sử dụng để lưu trữ kết quả truy vấn.
- Vòng lặp quay lại bước 4 để thực hiện truy vấn mới và cập nhật dữ liệu trên LCD.

2.3.3. Kết nối Raspberry Pi 4 với LED RGB



Hình 2. 9 Raspberry và LED

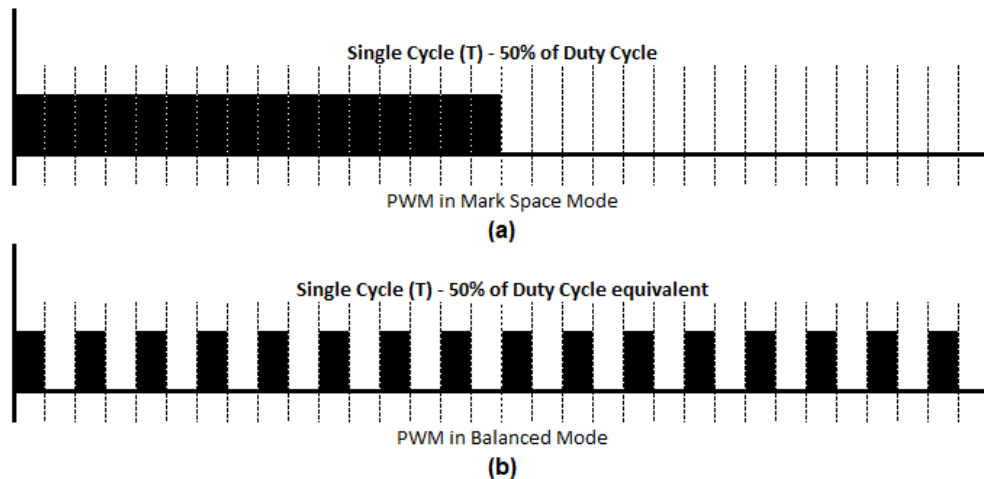
Sơ đồ chân

- GPIO 27 (13) – R – dây đỏ
- GPIO 22 (15) – G – dây xanh lá cây
- GPIO 17 (11) – B – dây xanh dương
- (14) – GND – dây đen

Đèn LED RGB hoạt động dựa trên sự phối hợp về cường độ ánh sáng của 3 nguồn màu: đỏ, xanh lá cây và xanh dương. Cường độ ánh sáng của 3 nguồn màu lại được điều khiển bằng cách thay đổi độ rộng xung của chúng.

Có 2 chế độ điều khiển độ rộng xung (PWM) trong Raspberry Pi 4 là:

- Chế độ Balance: điều khiển chu kì tắt
- Chế độ Mark: space



Hình 2. 10 Chế độ xung điều khiển

Nhóm sử dụng chế độ Balance áp dụng cho sản phẩm này.

Giải thuật lấy dữ liệu từ SQL và điều khiển LED

Giải thuật này sẽ truy vấn cơ sở dữ liệu MySQL để lấy các giá trị màu sắc (đỏ, xanh lá, xanh dương) từ bảng led_rgb. Sau đó, nó sẽ điều khiển độ sáng của từng kênh màu trên module LED RGB để tạo ra màu sắc mong muốn.

Khởi tạo kết nối MySQL: Tương tự như giải thuật trước, sử dụng mysql_init() và mysql_real_connect() để thiết lập kết nối đến cơ sở dữ liệu MySQL.

Khởi tạo WiringPi và chân GPIO:

- wiringPiSetup(): Khởi tạo thư viện WiringPi.
- pinMode(0, OUTPUT), pinMode(2, OUTPUT), pinMode(3, OUTPUT): Thiết lập các chân GPIO 0, 2, 3 làm ngõ ra để điều khiển các kênh màu xanh dương, đỏ và xanh lá của LED RGB.
- softPwmCreate(0, 0, 100), softPwmCreate(2, 0, 100), softPwmCreate(3, 0, 100): Khởi tạo PWM trên các chân GPIO này với dải giá trị từ 0 đến 100.

Vòng lặp vô hạn: Chương trình liên tục thực hiện các bước sau trong vòng lặp while(1): Truy vấn dữ liệu từ MySQL: mysql_query(conn, "SELECT red, green, blue FROM led_rgb ORDER BY timestamp DESC LIMIT 1"): Thực hiện truy vấn SQL để lấy các giá trị màu sắc (đỏ, xanh lá, xanh dương) mới nhất từ bảng led_rgb. Dữ liệu được sắp xếp theo thời gian giảm dần và chỉ lấy một bản ghi.

Xử lý kết quả truy vấn: Tương tự như giải thuật trước, sử dụng mysql_store_result() và

`mysql_fetch_row()` để lấy dữ liệu từ kết quả truy vấn.

Chuyển đổi và điều khiển LED RGB:

- `atof(row[0])`, `atof(row[1])`, `atof(row[2])`: Chuyển đổi các giá trị màu (kiểu chuỗi) sang kiểu số thực (float).
- Các giá trị màu được chia tỷ lệ từ 0-255 xuống 0-100 để phù hợp với dải giá trị của PWM.
- `softPwmWrite(0, blue)`, `softPwmWrite(2, green)`, `softPwmWrite(3, red)`: Điều khiển độ sáng của các kênh màu xanh dương, xanh lá và đỏ của LED RGB bằng cách thay đổi chu kỳ xung PWM.

Giải phóng bộ nhớ và lặp lại:

- `mysql_free_result(res)`: Giải phóng bộ nhớ đã sử dụng để lưu trữ kết quả truy vấn.
- Vòng lặp quay lại bước 4 để thực hiện truy vấn mới và cập nhật màu sắc của LED RGB.

2.3. Thiết kế cơ sở dữ liệu

Để lưu trữ và quản lý dữ liệu của hệ thống đèn thông minh, chúng ta sẽ sử dụng cơ sở dữ liệu MySQL. Dưới đây là thiết kế chi tiết cho hai bảng dữ liệu chính:

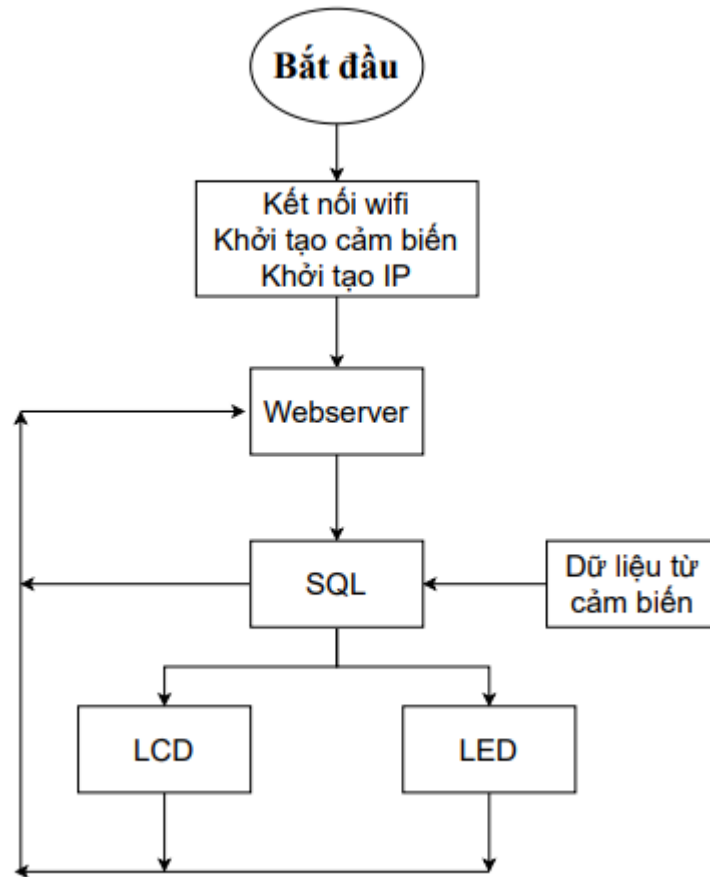
Thiết kế cơ sở dữ liệu cho DHT22

Bảng ***dht22_readings*** sẽ lưu trữ các giá trị nhiệt độ và độ ẩm được đo từ cảm biến DHT22 theo thời gian.

Thiết kế cơ sở dữ liệu cho LED RGB

Bảng ***led_rgb*** sẽ lưu trữ các giá trị màu sắc (đỏ, xanh lá, xanh dương) của đèn LED RGB được điều khiển bởi người dùng.

2.4. Lưu đồ thuật toán



Hình 2. 11 Lưu đồ thuật toán

- **Bắt đầu:** Khởi động hệ thống.
- **Kết nối WiFi:** Thiết bị thiết lập kết nối mạng không dây.
- **Khởi tạo cảm biến:** Cảm biến được chuẩn bị để thu thập dữ liệu.
- **Khởi tạo IP:** Thiết bị thiết lập địa chỉ IP để giao tiếp mạng.
- **Webserver:**
 - Một máy chủ web (có thể nhúng trong thiết bị) chạy để cung cấp giao diện người dùng (UI) và API (Application Programming Interface) cho phép tương tác với hệ thống từ xa.
 - Nhận dữ liệu từ các cảm biến thông qua SQL.

- **SQL:** Cơ sở dữ liệu lưu trữ dữ liệu cảm biến để xử lý và phân tích sau này.
- **Dữ liệu từ cảm biến:** Dữ liệu được thu thập từ cảm biến DHT22 và truyền đến SQL để lưu trữ.
- **LCD:** Màn hình LCD hiển thị thông tin cho người dùng giá trị nhiệt độ và độ ẩm
- **LED:** Hiện thị màu sắc ánh sáng RGB

2.5. Lựa chọn ngôn ngữ lập trình và thư viện

Để xây dựng hệ thống đèn thông minh, chúng ta sẽ sử dụng các ngôn ngữ lập trình và thư viện sau:

Ngôn ngữ lập trình Python:

- Lý do lựa chọn: Python là một ngôn ngữ lập trình bậc cao, dễ đọc, dễ học và có cú pháp đơn giản. Python có một cộng đồng người dùng lớn và nhiều thư viện hỗ trợ cho việc phát triển ứng dụng nhúng, đặc biệt là trên Raspberry Pi.
- Ứng dụng: Python sẽ được sử dụng để viết chương trình điều khiển phần cứng (cảm biến DHT22), giao tiếp với cơ sở dữ liệu MySQL và xử lý logic nghiệp vụ của hệ thống.
- Thư viện hỗ trợ cho Python:
 - Adafruit DHT: Thư viện hỗ trợ giao tiếp với cảm biến DHT22 để đọc dữ liệu nhiệt độ và độ ẩm.
 - RPi.GPIO: Thư viện điều khiển các chân GPIO của Raspberry Pi, cho phép tương tác với các thiết bị ngoại vi như LCD I2C và LED RGB.
 - PyMySQL: Thư viện kết nối và thao tác với cơ sở dữ liệu MySQL, giúp thực hiện các truy vấn và cập nhật dữ liệu.

Ngôn ngữ lập trình C:

- Lý do lựa chọn: C là một ngôn ngữ lập trình hệ thống, cho phép truy cập trực tiếp vào phần cứng và tài nguyên hệ thống. C có hiệu suất cao và thường được sử dụng để lập trình các thiết bị nhúng.
- Ứng dụng: C sẽ được sử dụng để viết các chương trình điều khiển trực tiếp LCD I2C và LED RGB, tận dụng tối đa hiệu suất và khả năng kiểm soát phần cứng của Raspberry Pi.

- Thư viện hỗ trợ cho C:
 - WiringPi: Thư viện cung cấp các hàm để làm việc với GPIO của Raspberry Pi, giúp đơn giản hóa việc điều khiển các thiết bị ngoại vi như LCD I2C và LED RGB.
 - mysqlclient: Thư viện C cho phép kết nối và tương tác với cơ sở dữ liệu MySQL.

Ngôn ngữ lập trình web (HTML, CSS, JavaScript):

- Lý do lựa chọn: Đây là bộ ba công nghệ cơ bản để xây dựng giao diện web. HTML định nghĩa cấu trúc của trang web, CSS tạo kiểu dáng và JavaScript xử lý các tương tác phía người dùng.
- Ứng dụng: HTML, CSS và JavaScript sẽ được sử dụng để xây dựng giao diện web điều khiển đèn thông minh, cho phép người dùng bật/tắt đèn, điều chỉnh độ sáng, màu sắc và xem thông tin môi trường.

PHP:

- Lý do lựa chọn: PHP là một ngôn ngữ lập trình phía máy chủ phổ biến, thường được sử dụng để xử lý các yêu cầu từ giao diện web và tương tác với cơ sở dữ liệu.
- Ứng dụng: PHP sẽ được sử dụng để xử lý các yêu cầu từ giao diện web, gửi lệnh điều khiển đến Raspberry Pi, đọc dữ liệu từ cảm biến và cơ sở dữ liệu, sau đó trả về kết quả cho giao diện web.

Thư viện hỗ trợ cho PHP:

- MySQLi: Thư viện hỗ trợ kết nối và thao tác với cơ sở dữ liệu MySQL từ PHP.

2.6. Kết luận chương

Chương 2 đã trình bày chi tiết về thiết kế hệ thống đèn thông minh, bao gồm các khía cạnh quan trọng như mô tả hệ thống, sơ đồ khối, sơ đồ mạch điện, thiết kế cơ sở dữ liệu, luồng dữ liệu, giải thuật xử lý dữ liệu cảm biến, giải thuật điều khiển LED và lựa chọn ngôn ngữ lập trình cùng các thư viện hỗ trợ.

Hệ thống được thiết kế với mục tiêu cung cấp giải pháp chiếu sáng thông minh, tiện lợi và tiết kiệm năng lượng, tích hợp khả năng điều khiển từ xa, hiển thị thông tin môi trường và tùy chỉnh màu sắc ánh sáng. Sơ đồ khối và sơ đồ mạch điện giúp minh họa rõ

ràng cấu trúc và kết nối giữa các thành phần phần cứng, tạo nền tảng vững chắc cho việc triển khai thực tế.

Thiết kế cơ sở dữ liệu chi tiết cho cảm biến DHT22 và LED RGB đảm bảo việc lưu trữ và quản lý dữ liệu một cách khoa học, tạo tiền đề cho việc phân tích và cải tiến hệ thống trong tương lai. Luồng dữ liệu được thiết lập rõ ràng, thể hiện sự tương tác chặt chẽ giữa các thành phần và đảm bảo thông tin được truyền tải chính xác và hiệu quả.

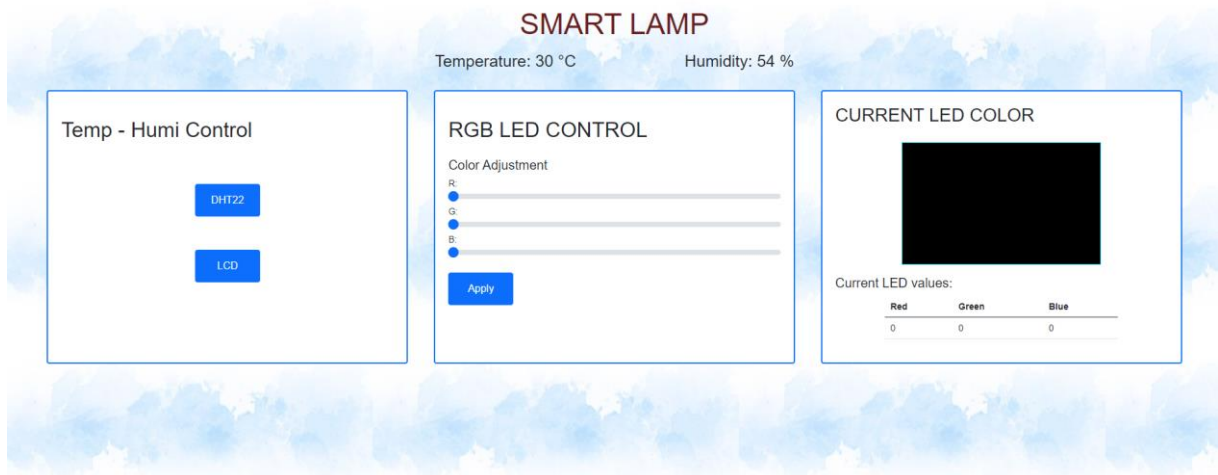
Các giải thuật xử lý dữ liệu cảm biến và điều khiển LED RGB đã được trình bày chi tiết, giúp hiểu rõ cách thức hệ thống hoạt động và cách thức các thành phần tương tác với nhau. Việc lựa chọn ngôn ngữ lập trình Python và C, cùng với các thư viện hỗ trợ phù hợp, đã tạo điều kiện thuận lợi cho việc phát triển phần mềm điều khiển và giao tiếp giữa các thành phần.

CHƯƠNG III: KẾT QUẢ

3.1. Sản phẩm đạt được

3.1.1. Phần mềm

Giao diện Web



Hình 3. 1 Giao diện phần mềm

Cơ sở dữ liệu

```
msi@raspberrypi: ~
```

File	Edit	Tabs	Help
------	------	------	------

```
MariaDB [sensor]> select * from led_rgb;
```

id	red	green	blue	timestamp
1	255	255	0	2024-07-09 05:39:53
2	0	0	0	2024-07-09 05:40:52
3	0	0	0	2024-07-09 05:40:52
4	0	0	0	2024-07-09 05:40:52
5	0	0	0	2024-07-09 05:40:54
6	144	78	107	2024-07-09 16:14:35
7	255	0	0	2024-07-09 16:16:30
8	255	255	255	2024-07-09 16:16:36
9	0	255	255	2024-07-09 16:16:39
10	255	0	255	2024-07-09 16:16:53
11	255	0	255	2024-07-09 16:31:21
12	255	0	255	2024-07-09 16:33:38
13	255	0	255	2024-07-09 16:33:39
14	255	0	255	2024-07-09 16:33:39
15	255	255	0	2024-07-09 16:33:46
16	255	0	255	2024-07-09 16:33:56
17	255	255	0	2024-07-09 16:34:30
18	255	255	255	2024-07-09 16:39:29
19	0	255	255	2024-07-09 16:39:32

Hình 3. 2 Cơ sở dữ liệu Led

```
msi@raspberrypi: ~  
File Edit Tabs Help  
MariaDB [sensor]> select * from dht22_readings;  
+-----+-----+-----+-----+  
| id | temperature | humidity | timestamp |  
+-----+-----+-----+-----+  
| 1 | 28.6 | 71.1 | 2024-06-12 00:55:23 |  
| 2 | 28.7 | 72 | 2024-06-12 00:55:34 |  
| 3 | 28.6 | 72.3 | 2024-06-12 00:55:44 |  
| 4 | 28.7 | 72.5 | 2024-06-12 00:55:54 |  
| 5 | 26.2 | 64.2 | 2024-06-12 11:03:20 |  
| 6 | 26.2 | 64.2 | 2024-06-12 11:03:31 |  
| 7 | 27.2 | 54.4 | 2024-06-12 17:36:04 |  
| 8 | 27.2 | 54.3 | 2024-06-12 17:36:15 |  
| 9 | 27.3 | 54.3 | 2024-06-12 17:36:25 |  
| 10 | 27.3 | 54.4 | 2024-06-12 17:36:35 |  
| 11 | 27.4 | 54.5 | 2024-06-12 17:36:45 |  
| 12 | 28.5 | 53.8 | 2024-06-13 19:22:08 |  
| 13 | 27.5 | 49.7 | 2024-06-13 19:22:18 |  
| 14 | 28.4 | 58.9 | 2024-06-14 11:18:55 |  
| 15 | 28.1 | 57.4 | 2024-06-14 11:19:15 |  
| 16 | 27.9 | 57.3 | 2024-06-14 11:19:26 |  
| 17 | 27.8 | 57.7 | 2024-06-14 11:19:36 |  
| 18 | 27.7 | 58.1 | 2024-06-14 11:19:46 |  
| 19 | 27.6 | 58 | 2024-06-14 11:19:56 |  
| 20 | 27.8 | 58.8 | 2024-06-14 11:20:17 |
```

Hình 3. 3 Cơ sở dữ liệu DHT22

3.1.2. Phần cứng



Hình 3. 4 Phần cứng

3.2. Nhận xét

3.2.1. Phần mềm

Hệ thống hoạt động ổn định. Dữ liệu nhiệt độ - độ ẩm được cảm biến đọc tương đối chính xác (sai số vẫn có nhưng không quá nhiều). LCD hoạt động bình thường. Các dữ liệu nhiệt độ - độ ẩm, giá trị RGB đã gửi lên web, điều khiển được led và LCD để hiển thị. Web hoạt động tốt có thể hiển thị được giá trị cần thiết.

3.2.2. Phần cứng

Phần cứng còn hơi thô sơ.

3.2.3. Cải tiến

- Phần cứng đẹp mắt hơn
- Hệ thống xử lý được nhiều bài toán hơn
- Tối ưu được việc điều khiển từ trên web xuống

TÀI LIỆU THAM KHẢO

[1] Bài giảng môn hệ thống nhúng của thầy Nguyễn Ngọc Thanh

[2] Bài giảng môn hệ thống nhúng SPKT:

https://www.youtube.com/playlist?list=PLgyH_I-Ia6LNYo3HSof9pooBToSy4Uoql

[3] Hướng dẫn build file C và python để chạy nền web:

[https://www.quora.com/How-can-one-run-a-C-program-in-](https://www.quora.com/How-can-one-run-a-C-program-in-PHP#:~:text=You%20may%20use%20the%20exec,run%20your%20C%20program%20executables.&text=You%20can%20use%20built-in,inside%20PHP%2C%20thats%20called%20FFI.&text=By%20using%20FFI%20your%20PHP,capable%20of%20producing%20an%20ABI)

[PHP#:~:text=You%20may%20use%20the%20exec,run%20your%20C%20program%20executables.&text=You%20can%20use%20built-](https://www.quora.com/How-can-one-run-a-C-program-in-PHP#:~:text=You%20may%20use%20the%20exec,run%20your%20C%20program%20executables.&text=You%20can%20use%20built-in,inside%20PHP%2C%20thats%20called%20FFI.&text=By%20using%20FFI%20your%20PHP,capable%20of%20producing%20an%20ABI)

[in,inside%20PHP%2C%20thats%20called%20FFI.&text=By%20using%20FFI%20your%20PHP,capable%20of%20producing%20an%20ABI](https://www.quora.com/How-can-one-run-a-C-program-in-PHP#:~:text=You%20may%20use%20the%20exec,run%20your%20C%20program%20executables.&text=You%20can%20use%20built-in,inside%20PHP%2C%20thats%20called%20FFI.&text=By%20using%20FFI%20your%20PHP,capable%20of%20producing%20an%20ABI)

[https://www.tutorialspoint.com/run-a-python-program-from-](https://www.tutorialspoint.com/run-a-python-program-from-php#:~:text=In%20PHP%2C%20the%20'shell_exec',returns%20no%20output%20at%20all)

[php#:~:text=In%20PHP%2C%20the%20'shell_exec',returns%20no%20output%20at%20all](https://www.tutorialspoint.com/run-a-python-program-from-php#:~:text=In%20PHP%2C%20the%20'shell_exec',returns%20no%20output%20at%20all)

[4] Các lý thuyết về các giao thức sử dụng:

<https://www.thegioic.com/tin-tuc/gioi-thieu-ve-chuan-giao-tiep-one-wire>

<https://dientutuonglai.com/chuan-giao-tiep-i2c-la-gi.html>

[5] DHT22 datasheet:

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

[6] LCD I2C datasheet:

https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf