TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT HÀN

**KHOA KĨ THUẬT MÁY TÍNH & ĐIỆN TỬ**



BÁO CÁO HỆ ĐIỀU HÀNH NHÚNG

**ĐỀ TÀI: HỆ THỐNG GIÁM SÁT THỜI TIẾT**

Sinh viên thực hiện: : **Đặng Trần Quang Hậu – 21CE015**

Lớp: : **21CE1**

Giảng viên hướng dẫn: : **TS. Vương Công Đạt**

***Đà nẵng, ngày 25 tháng 05 năm 2024***

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT HÀN

**KHOA KĨ THUẬT MÁY TÍNH & ĐIỆN TỬ**



BÁO CÁO HỆ ĐIỀU HÀNH NHÚNG

**ĐỀ TÀI: HỆ THỐNG GIÁM SÁT THỜI TIẾT**

Sinh viên thực hiện: **Đặng Trần Quang Hậu – 21CE015**

Lớp: **21CE1**

Giảng viên hướng dẫn: **TS. Vương Công Đạt**

***Đà Nẵng, ngày 25 tháng 5 năm 2024***

**NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

….………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Đà nẵng, ngày 25 tháng 05 năm 2024**

**Giảng viên hướng dẫn**

**Ts.Vương Công Đạt**

# **LỜI CẢM ƠN**

Sau thời gian thực hiện báo cáo môn học Hệ điều hành nhúng, đến nay công việc liên quan đến báo cáo đã hoàn tất. Ở phần đầu của báo cáo, cho phép chúng em có đôi điều gửi đến những người chúng em vô cùng biết ơn.

Để hoàn thành tốt bài báo cáo đồ án môn học lần này chúng em đã nhận được sự hỗ trợ, giúp đỡ của thầy cô. Với tình cảm sâu sắc, chân thành, cho phép chúng em được bày tỏ lòng biết ơn sâu sắc đến tất cả quý thầy cô đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu.

Trước hết chúng em xin gửi tới các thầy cô trường Đại học Công nghệ thông tin và truyền thông Việt – Hàn lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của các giảng viên đã truyền tải các kĩ năng cần thiết, đến nay chúng em đã có thể hoàn thành báo cáo đồ án môn học Hệ điều hành nhúng.

Đặc biệt chúng em xin gửi lời cảm ơn chân thành nhất tới thầy giáo – **TS.Vương Công Đạt** đã quan tâm, giúp đỡ tận tình chúng em hoàn thành tốt báo cáo này trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Chúng em xin chân thành cảm ơn!

**Đà nẵng, ngày 25 tháng 5 năm 2024**

**Sinh viên thực hiện**

**Đặng Trần Quang Hậu**

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 5](#_Toc167492532)

[1.1. Phân tích yêu cầu đề tài 5](#_Toc167492533)

[1.1.1. Tên đề tài: 5](#_Toc167492534)

[1.1.2. Chức năng 5](#_Toc167492535)

[1.2. Các ứng dụng của đề tài 5](#_Toc167492536)

[1.3. Công nghệ sử dụng 6](#_Toc167492537)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10](#_Toc167492538)

[2.1. Tổng quan về môn học hệ điều hành nhúng 10](#_Toc167492539)

[2.2. Tổng quan về FreeRTOS 11](#_Toc167492540)

**[2.2.1. Nguyên lý hoạt động của RTOS](#_Toc167492541)** [11](#_Toc167492541)

**[2.2.2. Phân loại RTOS](#_Toc167492542)** [12](#_Toc167492542)

**[2.2.3. Các chức năng của RTOS](#_Toc167492543)** [13](#_Toc167492543)

[2.3. Giới thiệu phần cứng 14](#_Toc167492544)

**[2.3.1. Module Esp8266](#_Toc167492545)** [14](#_Toc167492545)

**[2.3.2. Cảm biến nhiệt độ, độ ẩm](#_Toc167492546)** [15](#_Toc167492546)

**[2.3.3. Màn hình Oled 0.96inch](#_Toc167492547)** [17](#_Toc167492547)

**[2.3.4. Phần mềm thực hiện Arduino IDE](#_Toc167492548)** [18](#_Toc167492548)

**[2.3.5. Ngôn ngữ lập trình C++](#_Toc167492549)** [18](#_Toc167492549)

[2.7 Kết chương 2 20](#_Toc167492550)

[CHƯƠNG 3: TRIỂN KHAI XÂY DỰNG 21](#_Toc167492551)

[3.1. Danh sách thông số các linh kiện có trong mạch 21](#_Toc167492552)

**[3.2.1. Cách kết nối chân với nhau](#_Toc167492569)** [21](#_Toc167492569)

**[3.2.3. Sơ đồ mạch trên Protues](#_Toc167492570)** [22](#_Toc167492570)

[3.3. Nguyên lý hoạt động của mạch 22](#_Toc167492571)

[3.5. Mạch PCB 23](#_Toc167492572)

[3.6. Kết chương 3 24](#_Toc167492573)

[KẾT LUẬN 25](#_Toc167492574)

[TÀI LIỆU THAM KHẢO 27](#_Toc167492582)

**DANH MỤC HÌNH VẼ**

[Hình 1. 1 PROTEUS 7](#_Toc167509820)

[Hình 1. 2 Phần mềm Altium 8](#_Toc167509821)

[Hình 2. 1 Hệ Thống Nhúng 10](#_Toc167509830)

[Hình 2. 2 Cơ chế hướng sự kiện 11](#_Toc167509831)

[Hình 2. 3Cơ chế chia sẻ thời gian 11](#_Toc167509832)

[Hình 2. 4 Hard real time OS 12](#_Toc167509833)

[Hình 2. 5 Soft Real-Time Operating System: 12](#_Toc167509834)

[Hình 2. 6 Firm Real-Time Operating System 13](#_Toc167509835)

[Hình 2. 7 Chức năng cơ bản của RTOS 14](#_Toc167509836)

[Hình 2. 8 Sơ đồ chân kết nối ESP8266 15](#_Toc167509837)

[Hình 2. 9 Sơ đồ chân DHT11 16](#_Toc167509838)

[Hình 2. 10 Màn hình Oled 0.96inch 17](#_Toc167509839)

[Hình 2. 11 Giao diện phần mềm Arduino IDE 18](#_Toc167509840)

[Hình 2. 12 Ưu điểm của ngôn ngữ C++ 19](#_Toc167509841)

[Hình 3. 1 Sơ đồ mạch 21](#_Toc167509842)

[Hình 3. 2 Sơ đồ trên Protues 22](#_Toc167509843)

[Hình 3. 3 Mạch PCB 23](#_Toc167509844)

**DANH MỤC BẢNG**

[Bảng 1. 1 Kế hoạch thực hiện 4](#_Toc167510078)

[Bảng 2. 2 Kết nối chân DHT11 16](#_Toc167510274)

[Bảng 3. 1 Linh Kiện 21](#_Toc167510316)

**MỞ ĐẦU**

1. **Giới thiệu**

Trong thế giới công nghệ ngày nay, hệ điều hành nhúng đóng vai trò cực kỳ quan trọng trong việc điều khiển và quản lý các thiết bị nhúng. Từ điện thoại thông minh, đồng hồ thông minh, thiết bị điều khiển trong ô tô cho đến các thiết bị y tế và các hệ thống công nghiệp, hệ điều hành nhúng đã trở thành "nền tảng vô hình" đằng sau sự hoạt động của hàng tỷ thiết bị trên khắp thế giới.

Hệ điều hành nhúng là một phần mềm hoặc một hệ thống phần cứng và phần mềm kết hợp, được tối ưu hóa để chạy trên các thiết bị nhúng có tài nguyên hạn chế, như bộ vi xử lý nhỏ, bộ nhớ hạn chế và điện năng thấp. Nó được thiết kế để điều khiển và quản lý các chức năng cơ bản của thiết bị, như điều khiển đèn, giao tiếp mạng, cảm biến và định vị.

Một trong những ứng dụng quan trọng của hệ điều hành nhúng là trong lĩnh vực giám sát thời tiết. Việc giám sát và dự báo thời tiết đã trở thành một yếu tố quan trọng trong đời sống hàng ngày của chúng ta. Hệ thống giám sát thời tiết đóng vai trò quan trọng trong việc thu thập dữ liệu, phân tích và cung cấp thông tin thời tiết chính xác và kịp thời cho người dùng.

Hệ điều hành nhúng chịu trách nhiệm điều khiển và quản lý các tác vụ trong hệ thống giám sát thời tiết. Nó cung cấp giao diện cho việc thu thập dữ liệu từ cảm biến, xử lý và phân tích dữ liệu, lưu trữ và truyền tải thông tin thời tiết cho người dùng. Hệ điều hành nhúng cũng có nhiệm vụ quản lý tài nguyên, quản lý nguồn năng lượng và đảm bảo tính ổn định và đáng tin cậy của hệ thống.

Trong môn học hệ điều hành nhúng, chúng ta sẽ có cơ hội áp dụng các kiến thức về hệ điều hành nhúng và lập trình nhúng để xây dựng một hệ thống giám sát thời tiết nhúng thực tế. Việc thực hành này sẽ giúp chúng ta rèn luyện kỹ năng lập trình, hiểu rõ về cách hoạt động của các thiết bị nhúng và cách tương tác với chúng.

* ESP8266 là một module Wi-Fi có giá thành thấp, nhỏ gọn và dễ sử dụng. Với khả năng kết nối mạng không dây, ESP8266 cho phép chúng ta thu thập dữ liệu từ các cảm biến và truyền nó đến một trung tâm giám sát. Sự phổ biến của ESP8266 giúp đơn giản hóa việc triển khai hệ thống giám sát thời tiết.
* DHT11 là một cảm biến giá thành thấp, dễ sử dụng và phổ biến trong việc đo nhiệt độ và độ ẩm trong môi trường. Sử dụng DHT11, chúng ta có thể thu thập dữ liệu về nhiệt độ và độ ẩm của môi trường xung quanh một cách đáng tin cậy và chính xác.
* Màn hình OLED (Organic Light-Emitting Diode) cung cấp một giao diện hiển thị tốt, tiêu thụ năng lượng thấp và kích thước nhỏ gọn. Sử dụng một màn hình OLED, chúng ta có thể hiển thị thông tin về thời tiết như nhiệt độ, độ ẩm và biểu đồ dự báo trực quan và dễ đọc.

Thiết kế hệ thống trên một PCB riêng biệt giúp tối ưu hóa việc kết nối và tích hợp các linh kiện. Việc sử dụng PCB giúp giảm không gian và đơn giản hóa việc kết nối giữa các thành phần, tạo ra một hệ thống nhỏ gọn và dễ dàng lắp đặt.Các thành phần được gắn chặt và đúng vị trí trên PCB, tạo ra một hệ thống ổn định với hiệu suất cao và giảm thiểu rủi ro sự cố. Đó là lý do nhóm chúng em chọn đề tài "Hệ thống giám sát thời tiết".

1. **Mục tiêu của đề tài**

Mục tiêu của đề tài “Hệ thống giám sát thời tiết ” là thiết kế và xây dựng một hệ thống giám sát thời tiết sử dụng ESP8266, cảm biến nhiệt độ và độ ẩm DHT11, cùng với màn hình OLED, và thiết kế mạch PCB để tích hợp các thành phần này vào một thiết bị duy nhất. Ngoài ra, nhóm sử dụng hệ điều hành FreeRTOS để quản lý và điều khiển các tác vụ trong hệ thống.

Mục tiêu của đề tài bao gồm:

* Tìm hiểu về ESP8266: Đặc điểm, khả năng kết nối Internet và giao tiếp với các thiết bị khác. Tìm hiểu cách sử dụng ESP8266, lập trình và kết nối với các thành phần khác.
* Tìm hiểu về cảm biến DHT11: Cảm biến đo nhiệt độ và độ ẩm DHT11. Đánh giá đặc điểm kỹ thuật, cách kết nối và cách đọc dữ liệu từ cảm biến.
* Tìm hiểu về màn hình OLED: Màn hình OLED làm việc dựa trên công nghệ OLED để hiển thị thông tin. Nghiên cứu cách kết nối màn hình OLED với ESP8266 và cách điều khiển hiển thị trên màn hình.
* Thiết kế mạch PCB: Xây dựng mạch PCB để tích hợp ESP8266, cảm biến DHT11 và màn hình OLED vào một thiết bị duy nhất. Sử dụng phần mềm thiết kế mạch như Eagle hoặc KiCad để đảm bảo các chân kết nối và liên kết chính xác trên mạch PCB.
* Sử dụng hệ điều hành FreeRTOS: Nghiên cứu về hệ điều hành FreeRTOS và khả năng của nó trong việc quản lý và điều khiển các tác vụ trong hệ thống nhúng. Áp dụng FreeRTOS để tạo ra các tác vụ độc lập cho việc đọc dữ liệu từ cảm biến, hiển thị trên màn hình và gửi thông tin qua mạng.
* Lập trình và phát triển phần mềm: Lập trình ESP8266 và hệ điều hành FreeRTOS để quản lý các tác vụ và tương tác với các thành phần khác. Đảm bảo tính ổn định của hệ thống và hiệu suất trong việc thu thập, xử lý và hiển thị dữ liệu thời tiết.
* Kiểm thử và đánh giá: Kiểm tra hệ thống để đảm bảo hoạt động chính xác. Đánh giá hiệu suất và độ ổn định của hệ thống giám sát thời tiết trong các điều kiện khác nhau.

1. **Nội dung và kế hoach thực hiện**
2. **Nội dung thực hiện**

Các nội dung cần thực hiện trong đề tài

* Tìm hiểu về nguyên lý hoạt động: Nắm vững nguyên lý hoạt động của mạch giám sát thời tiết, bao gồm esp8266, cảm biến nhiệt độ, độ ẩm DHT11, màn hình oled 0.96inch. Tìm hiểu về các thành phần điện tử được sử dụng trong mạch và cách chúng tương tác với nhau.
* Thiết kế mạch điều khiển: Xác định và thiết kế mạch giám sát thời tiết dựa trên yêu cầu cụ thể của đề tài. Bao gồm cả việc lựa chọn cảm biến nhiệt độ độ ẩm phù hợp, lựa chọn linh kiện và mạch tích hợp, thiết kế mạch in và bố trí linh kiện.
* Hiệu chỉnh và thử nghiệm: Tiến hành hiệu chỉnh và thử nghiệm mạch điều khiển để đảm bảo hoạt động chính xác và ổn định. Điều chỉnh ngưỡng nhiệt độ và các thiết lập khác để đáp ứng yêu cầu cụ thể của không gian sử dụng. Kiểm tra tính tin cậy và hiệu suất của mạch trong các điều kiện khác nhau.
* Tối ưu hóa và cải tiến: Đánh giá và tối ưu hóa hiệu suất tiêu thụ năng lượng của mạch để đảm bảo tính bền vững và tiết kiệm năng lượng. Xem xét các yếu tố khác như tích hợp hẹn giờ, chế độ hoạt động đa dạng, tích hợp vào hệ thống tổng thể quản lý thông minh, nếu cần.
* Tài liệu hóa và báo cáo: Tạo tài liệu hóa đầy đủ về mạch điều khiển, bao gồm sơ đồ mạch, bảng mạch, mã lập trình và hướng dẫn sử dụng. Viết báo cáo tổng kết về quá trình thực hiện đề tài, kết quả thử nghiệm và đánh giá hiệu suất của mạch.

**Làm slide thuyết trình về dự án**

* Công cụ thực hiện: PowerPoint

**Làm báo cáo dự án**

* Công cụ thực hiện: MS Word

1. **Kế hoạch thực hiện**

|  |  |
| --- | --- |
| **Thời gian** | **Nội dung thực hiện** |
| Tuần 1 đến tuần 3 | Tìm hiểu về đề tài và hướng phát triển đề tài |
| Tuần 3 đến tuần 5 | Tiến hành vẽ mạch nguyên lý và mạch in trên protues |
| Tuần 6 đến tuần 8 | Tiến hành mua các linh kiện, in mạch in và hàn mạch |
| Tuần 9 đến tuần 11 | Kiểm tra lại sản phẩm và hoàn thiện báo cáo slide |

Bảng 1. 1 Kế hoạch thực hiện

1. **Bố cục báo cáo**

Sau phần *Mở đầu*, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1. *Tổng quan về đề tài.* Trong chương này, báo cáo trình bày chung về đề tài các khái niệm căn bản của mạch, tầm quan trong của mạch

Chương 2. *Cơ sở lý thuyết*. Nội dung chương bao gồm các nội dung :

Chương 3.Triển khai xây dựng*.* Chương này bao gồm các nội dung :

* Trình bày sơ đồ nguyên lý của mạch điều khiển
* Giải thích vai trò và hoạt động của từng thành phần trong mạch

Chương 4. *Xây dựng mạch thực tế.* Chương này sẽ giới thiệu về quy trình làm mạch và phần demo sản phẩm

Chương 5. *Kết quả đạt được và cải tiến* . Chương này sẽ tổng hợp lại kết quả đạt được và đề xuất những cải tiến mới cho mạch

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* 1. **Phân tích yêu cầu đề tài** 
     1. **Tên đề tài:**

Xây dựng mạch “***HỆ THỐNG GIÁM SÁT THỜI TIẾT***”

* + 1. **Chức năng**

Đo nhiệt độ và độ ẩm: Hệ thống sử dụng cảm biến DHT11 để đo nhiệt độ và độ ẩm trong môi trường. Cảm biến này sẽ thu thập thông tin và truyền cho ESP8266 để xử lý.

Kết nối mạng Wi-Fi: ESP8266 là một vi mạch Wi-Fi tích hợp, cho phép hệ thống kết nối với mạng Wi-Fi và truyền dữ liệu thời tiết tới một máy chủ hoặc ứng dụng di động. Việc kết nối mạng Wi-Fi giúp hệ thống cập nhật thông tin thời tiết và truyền tải dữ liệu một cách liên tục.

Hiển thị trên màn hình OLED: Hệ thống được trang bị một màn hình OLED để hiển thị thông tin thời tiết. Màn hình sẽ hiển thị nhiệt độ, độ ẩm và các biểu đồ thể hiện thay đổi trong thời gian thực. Điều này giúp người dùng dễ dàng quan sát và đánh giá tình trạng thời tiết.

Thiết kế mạch PCB: Hệ thống được thiết kế và tích hợp trên một mạch PCB nhỏ gọn. Thiết kế mạch PCB giúp tăng tính ổn định và đảm bảo kết nối chính xác giữa các thành phần. Ngoài ra, nó cũng giúp tối ưu hóa không gian và dễ dàng triển khai hệ thống.

Sử dụng hệ điều hành FreeRTOS: Hệ thống áp dụng hệ điều hành FreeRTOS để quản lý và điều khiển các tác vụ trong hệ thống. FreeRTOS cung cấp khả năng lập lịch và ưu tiên các tác vụ, đồng thời đảm bảo tính tin cậy và ổn định của hệ thống. Việc sử dụng FreeRTOS giúp quản lý các tác vụ đọc dữ liệu từ cảm biến, hiển thị trên màn hình và gửi thông tin qua mạng một cách hiệu quả và đồng bộ.

Mở rộng cảm biến: Ngoài cảm biến nhiệt độ và độ ẩm, mạch cũng có thể được mở rộng để kết nối và thu thập dữ liệu từ các cảm biến khác như cảm biến ánh sáng, cảm biến áp suất không khí, hoặc cảm biến khí độc.

Giao diện người dùng đa dạng: Mạch có thể được tích hợp với các giao diện người dùng khác nhau như giao diện web hoặc ứng dụng di động, giúp người dùng dễ dàng theo dõi và quản lý thông tin thời tiết từ xa.

* 1. **Các ứng dụng của đề tài**

Giám sát thời tiết chính xác: Hệ thống cho phép đo và giám sát nhiệt độ và độ ẩm trong môi trường một cách chính xác. Thông qua cảm biến DHT11, dữ liệu được thu thập và hiển thị trên màn hình OLED, cung cấp người dùng thông tin chi tiết về tình trạng thời tiết hiện tại.

Hiển thị trực quan: Màn hình OLED trong hệ thống cho phép hiển thị thông tin thời tiết một cách trực quan và dễ đọc. Người dùng có thể dễ dàng nhìn thấy nhiệt độ, độ ẩm và các biểu đồ thể hiện thay đổi trong thời gian thực. Điều này giúp người dùng nắm bắt thông tin thời tiết một cách nhanh chóng và thuận tiện.

Kết nối mạng và truyền dữ liệu: Với sự hỗ trợ của ESP8266, hệ thống có khả năng kết nối với mạng Wi-Fi và truyền dữ liệu thời tiết tới một máy chủ hoặc ứng dụng di động. Điều này cho phép người dùng cập nhật thông tin thời tiết từ xa và theo dõi các thay đổi trong thời gian thực.

Thiết kế mạch PCB nhỏ gọn: Thiết kế mạch PCB giúp tối ưu hóa không gian và đảm bảo tính ổn định của hệ thống. Hệ thống có thể được tích hợp vào các thiết bị nhỏ gọn và di động như đồng hồ thông minh, thiết bị đeo tay hoặc các trạm giám sát thời tiết tự động.

Sử dụng hệ điều hành FreeRTOS: Ứng dụng hệ điều hành FreeRTOS giúp quản lý và điều khiển các tác vụ trong hệ thống một cách hiệu quả. Hệ điều hành này cung cấp khả năng lập lịch và ưu tiên các tác vụ, đồng thời đảm bảo tính tin cậy và ổn định của hệ thống. Điều này cho phép hệ thống thực hiện đồng thời các tác vụ đo nhiệt độ, độ ẩm, hiển thị và truyền tải dữ liệu một cách đồng bộ.

Mở rộng và tích hợp: Hệ thống có thể được mở rộng và tích hợp vào các ứng dụng khác như dự báo thời tiết, cảnh báo và liên kết với các thiết bị thông minh khác trong mạng. Điều này tạo ra tiềm năng cho việc phát triển hệ thống giám sát thời tiết với các tính năng và ứng dụng đa dạng hơn.

* 1. **Công nghệ sử dụng**
     1. **Phần mềm Protues**

Phần mềm Proteus là một môi trường mô phỏng phần cứng (hardware) và phần mềm (software) được sử dụng rộng rãi trong lĩnh vực thiết kế và kiểm tra vi mạch điện tử. Nó cung cấp các công cụ mô phỏng và xem trước cho vi mạch điện tử, cho phép người dùng thiết kế, mô phỏng, và kiểm tra các mạch điện tử trước khi chúng được triển khai trong thực tế.

Proteus bao gồm hai thành phần chính: ISIS và ARES. ISIS cho phép người dùng tạo ra các mạch điện tử bằng cách kết nối các thành phần điện tử trong thư viện của nó và mô phỏng hoạt động của chúng. Nó cung cấp một giao diện trực quan cho việc thiết kế mạch và cho phép người dùng kiểm tra và sửa lỗi mạch trước khi chuyển sang giai đoạn thực tế.

ARES là một công cụ thiết kế mạch in (PCB) tích hợp trong Proteus. Người dùng có thể chuyển các mạch đã thiết kế từ ISIS sang ARES để tạo ra một mạch in PCB hoàn chỉnh. ARES cung cấp các tính năng như định vị linh hoạt, định tuyến tự động và kiểm tra luật mạch in để giúp người dùng tạo ra mạch in PCB chất lượng cao.

Proteus cũng hỗ trợ việc mô phỏng và kiểm tra các vi mạch điều khiển (microcontroller) thông qua việc tích hợp sắp xếp mô phỏng cho các vi điều khiển phổ biến như 8051, PIC, Arduino và nhiều loại vi điều khiển khác.



Hình 1. 1 PROTEUS

* + 1. **Phần mềm Altium**

Phần mềm Altium Designer là một công cụ thiết kế mạch in (PCB) và quản lý dự án điện tử được sử dụng rộng rãi trong lĩnh vực thiết kế và phát triển vi mạch điện tử. Nó cung cấp một môi trường tích hợp cho việc thiết kế, mô phỏng, mô hình hóa và sản xuất các mạch điện tử. Altium Designer là một phần mềm thiết kế mạch điện tử (EDA - Electronic Design Automation) phổ biến được sử dụng để thiết kế các mạch in và hệ thống điện tử. Dưới đây là một số tính năng chính của Altium Designer:

* Altium Designer cung cấp nhiều tính năng quan trọng cho quá trình thiết kế PCB. Các tính năng chính bao gồm:
* Schematic Editor: Cho phép người dùng tạo ra sơ đồ mạch điện tử thông qua kết nối các thành phần điện tử và đường dẫn tín hiệu.
* PCB Editor: Cho phép người dùng thiết kế mạch in PCB bằng cách định vị các thành phần, định tuyến đường dẫn tín hiệu, và tạo layout mạch in chính xác.
* Simulation and Analysis: Altium Designer cung cấp các công cụ mô phỏng và phân tích mạch, cho phép người dùng kiểm tra và đánh giá hiệu suất và tương tác của mạch điện tử trước khi chuyển sang giai đoạn sản xuất.
* Library Management: Nó cung cấp quản lý thư viện linh hoạt, cho phép người dùng lưu trữ và quản lý các thành phần điện tử, mô hình và thông tin linh kiện.
* Manufacturing Outputs: Altium Designer hỗ trợ tạo các đầu ra sản xuất như file Gerber, BOM (Bill of Materials), và thông tin thiết kế nhằm hỗ trợ quá trình sản xuất và lắp ráp mạch điện tử.
* Thiết kế đa lớp và đa bản in: Altium Designer hỗ trợ thiết kế mạch in với nhiều lớp, bản in và kích thước khác nhau, từ các mạch đơn giản đến các hệ thống phức tạp.



Hình 1. 2 Phần mềm Altium

* + 1. **Phần mềm Arduino IDE**

Phần mềm Arduino IDE (Integrated Development Environment) là một môi trường phát triển tích hợp được sử dụng cho vi điều khiển Arduino. Nó là một phần mềm mã nguồn mở và miễn phí, được phát triển bởi Arduino.cc.

Arduino IDE cung cấp một giao diện đồ họa dễ sử dụng cho việc lập trình vi điều khiển Arduino. Với Arduino IDE, người dùng có thể viết code, biên dịch và nạp chương trình vào vi điều khiển Arduino để điều khiển các thiết bị và thực hiện các tác vụ như đọc cảm biến, điều khiển động cơ, giao tiếp với các module ngoại vi, và nhiều ứng dụng khác.

Các tính năng chính của Arduino IDE bao gồm:

* Code Editor: Arduino IDE cung cấp một trình soạn thảo mã để viết code Arduino. Nó hỗ trợ các ngôn ngữ lập trình dựa trên Wiring và C++.
* Compilation và Upload: Arduino IDE cho phép biên dịch mã nguồn thành mã máy thích hợp cho vi điều khiển Arduino và tải chương trình lên vi điều khiển thông qua cổng kết nối.
* Libraries: Arduino IDE đi kèm với một số thư viện tiêu chuẩn cho vi điều khiển Arduino, giúp người dùng dễ dàng sử dụng các chức năng và module phổ biến.
* Serial Monitor: IDE cung cấp một Serial Monitor cho phép người dùng giao tiếp với vi điều khiển Arduino thông qua cổng giao tiếp nối tiếp (Serial) và theo dõi dữ liệu gửi và nhận.
* Debugging: Mặc dù Arduino IDE không cung cấp các công cụ debugging đầy đủ, nhưng người dùng có thể sử dụng các phương pháp debug thông qua việc sử dụng câu lệnh in ra (print) và theo dõi các giá trị biến trong quá trình chạy chương trình.



Hình 1.3: Arduino IDE

* 1. **Kết chương 1**

*Thông qua tìm hiểu chung về đề tài và tìm hiều được các phần mềm để vẽ mach, các ứng dụng của mạch , từ đó làm cơ sở để trình bày các cơ sở lý thuyết ở chương tiếp theo*

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1. Tổng quan về môn học hệ điều hành nhúng**

Môn học "Hệ điều hành nhúng" (Embedded Operating Systems) là một môn học trong lĩnh vực kỹ thuật điện tử và viễn thông, tập trung vào việc nghiên cứu và áp dụng hệ điều hành trong các hệ thống nhúng.

Hệ điều hành nhúng là một phần mềm chạy trên vi điều khiển nhúng, làm nhiệm vụ quản lý tài nguyên phần cứng và cung cấp các dịch vụ để phát triển và thực thi các ứng dụng nhúng. Một hệ điều hành nhúng phải đảm bảo hoạt động ổn định và hiệu quả trên các hệ thống có tài nguyên hạn chế về bộ nhớ, xử lý và năng lượng.

Một số nội dung chính trong môn học "Hệ điều hành nhúng" bao gồm:

* Kiến thức về vi điều khiển nhúng: Nắm vững kiến thức về vi điều khiển nhúng, kiến trúc phần cứng và nguyên lý hoạt động của vi điều khiển.
* Kiến thức về hệ điều hành: Tìm hiểu về cấu trúc và nguyên lý hoạt động của hệ điều hành nhúng, bao gồm quản lý tiến trình, lập lịch, quản lý bộ nhớ, truy cập vào thiết bị ngoại vi, và giao tiếp với ứng dụng.
* Phát triển ứng dụng nhúng: Học cách phát triển phần mềm nhúng sử dụng các công cụ và ngôn ngữ lập trình phổ biến như C/C++. Tìm hiểu về việc lập trình các chức năng hệ điều hành, truy cập tài nguyên phần cứng, và xử lý sự kiện.
* Giao tiếp và giao thức nhúng: Tìm hiểu về các giao thức giao tiếp nhúng như UART, SPI, I2C và Ethernet. Nắm vững cách sử dụng các giao thức này để giao tiếp với các thiết bị ngoại vi và các hệ thống khác.
* Kiểm thử và gỡ lỗi: Học cách kiểm thử và gỡ lỗi các ứng dụng nhúng, sử dụng các công cụ và kỹ thuật phù hợp để đảm bảo tính ổn định và đáng tin cậy của hệ thống.



Hình 2. 1 Hệ Thống Nhúng

## 2.2. Tổng quan về FreeRTOS

RTOS (Real-Time Operating System) một hệ điều hành thời gian thực, được dùng trong lĩnh vực thiết kế các hệ thống nhúng. RTOS được thiết kế để giải quyết các yêu cầu về phản hồi thời gian thực và hiệu suất cao cho các ứng dụng cần xử lý số lượng lớn dữ liệu và thực hiện nhiều tác vụ đồng thời.

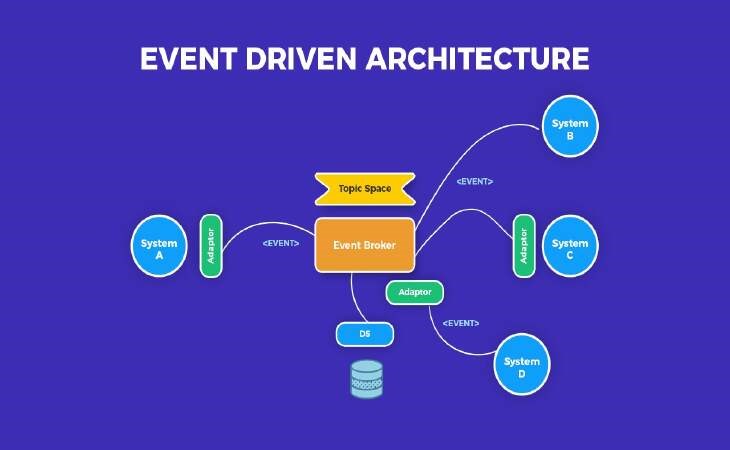
RTOS có khả năng quản lý và phân chia thời gian CPU cho các tiến trình và luồng dữ liệu một cách linh hoạt, đồng thời cung cấp các tính năng liên quan đến bộ định thời, bộ lập lịch, bộ định thời thực, và các giải pháp kiểm soát lỗi.

Với các tính năng này, RTOS cho phép các ứng dụng nhúng thực hiện các tác vụ phân tán, đáp ứng nhanh, và đáp ứng các yêu cầu hàng đầu của các ứng dụng như: IoT, xe tự lái, điện toán đám mây,...

### **2.2.1. Nguyên lý hoạt động của RTOS**

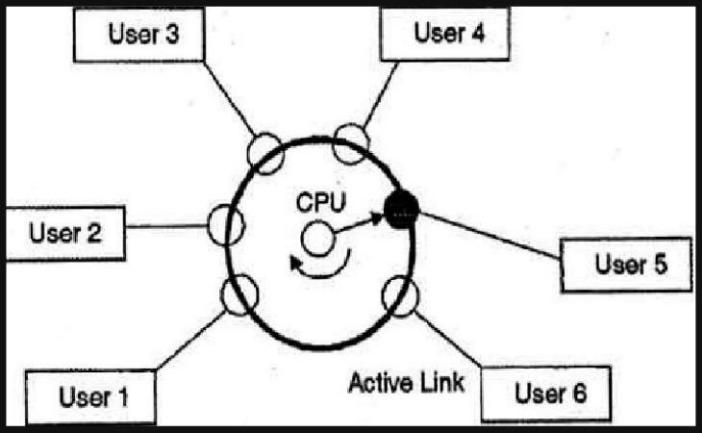
RTOS hoạt động dựa trên hai cơ chế là hướng sự kiện (event-driven) hoặc chia sẻ thời gian (time-sharing):

* Cơ chế hướng sự kiện sẽ giải quyết và điều phối các tác vụ (task) thông qua mức độ ưu tiên của chúng.



Hình 2. 2 Cơ chế hướng sự kiện

* Cơ chế chia sẻ thời gian sẽ chuyển đối các tác vụ dựa trên phản ứng ngắt của xung nhịp.

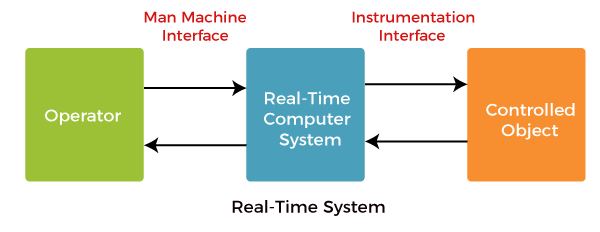


Hình 2. 3Cơ chế chia sẻ thời gian

**2.2.2. Phân loại RTOS**

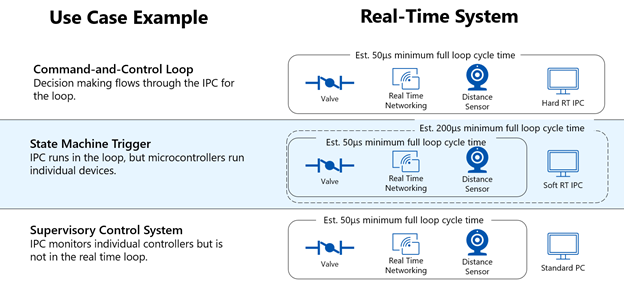
RTOS thường được chia làm 3 loại chính:

* Hard Real-Time Operating System: Loại hệ điều hành này có khả năng đáp ứng các yêu cầu về thời gian thực trong mọi trường hợp, không bao giờ bị lỡ hẹn. Các ứng dụng sử dụng hard real-time operating system thường là các thiết bị yêu cầu xử lý ngay lập tức như xe tự lái, máy bay không người lái, v.v.



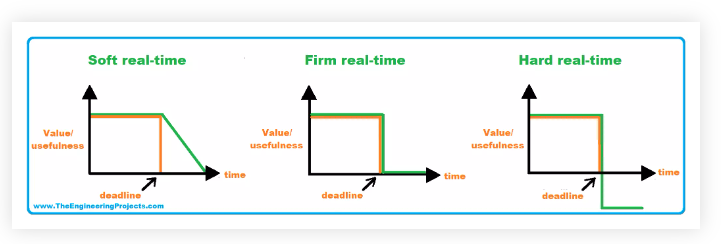
Hình 2. 4 Hard real time OS

* Soft Real-Time Operating System: Loại hệ điều hành này đảm bảo đáp ứng các yêu cầu về thời gian thực nhưng có thể bị lỡ hẹn trong một số tình huống nhất định. Các ứng dụng sử dụng soft real-time operating system thường là các ứng dụng IoT, điện toán đám mây, v.v.



Hình 2. 5 Soft Real-Time Operating System:

* Firm Real-Time Operating System: Loại hệ điều hành này nằm giữa Hard và Soft Real-Time Operating System. Nó đảm bảo thực hiện các tác vụ theo thời gian quy định, tuy nhiên có thể có một số trường hợp bị lỡ hẹn. Các ứng dụng sử dụng firm real-time operating system thường là các ứng dụng xử lý dữ liệu và đáp ứng nhanh như các hệ thống điều khiển tự động.



Hình 2. 6 Firm Real-Time Operating System

### **2.2.3. Các chức năng của RTOS**

Bộ lập lịch (Scheduler) có ba trạng thái:

* Ready to run: Là trạng thái mà task đã có đủ các tài nguyên để khởi chạy nhưng chưa chạy. Đây là trạng thái chuẩn bị của task.
* Running: Là trạng thái mà task đang được thực thi.
* Blocked: Khi task không có đủ các tài nguyên cần thiết để chạy thì nó sẽ được đưa về trạng thái blocked

Dịch vụ thời gian thực (Real Time Services) các dịch vụ của RTOS bao gồm:

* Xử lý ngắt (Interrupt handling services).
* Dịch vụ quản lý thời gian (Time services).
* Dịch vụ quản lý thiết bị (Device management services).
* Dịch vụ quản lý bộ nhớ (Memory management services).
* Dịch vụ quản lý các kết nối Vào - Ra (IO services).

Đồng bộ và thông điệp (Synchronization and Messaging) các thông điệp sử dụng để trao đổi thông tin giữa các hệ thống khác nhau hoặc giữa các task. Các dịch vụ quản lý thông điệp bao gồm:

* Semaphores: Dùng để đồng bộ hóa quyền truy cập vào các tài nguyên dùng chung.
* Event Flags: Dùng để đồng bộ hóa các hoạt động cần có sự phối hợp của nhiều task.
* Mailboxes, Pipes, Message queues: Dùng để quản lý các thông điệp gửi đi đến giữa các task.



Hình 2. 7 Chức năng cơ bản của RTOS

**2.3. Giới thiệu phần cứng**

### **2.3.1. Module Esp8266**

ESP8266 là một mô-đun Wi-Fi nhỏ gọn và mạnh mẽ, được phát triển bởi Espressif Systems. Nó kết hợp Wi-Fi điều khiển và chức năng vào một chip duy nhất, cho phép các thiết bị điện tử thông tin kết nối với Wi-Fi mạng và truyền dữ liệu qua Internet.

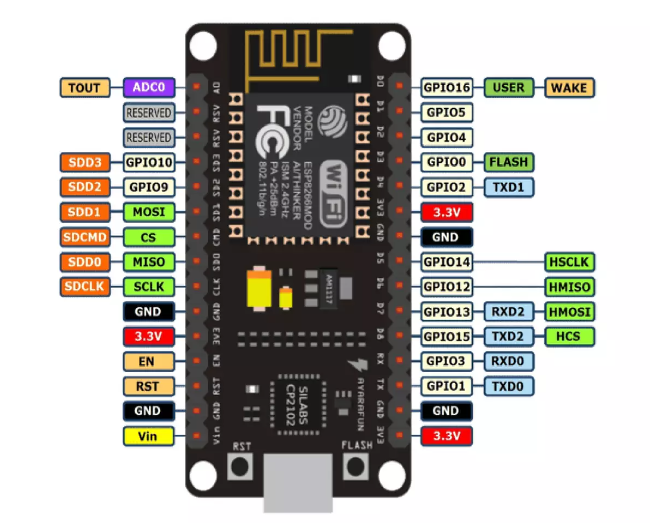
ESP8266 được tích hợp sẵn bộ vi xử lý Tensilica L106 32-bit, đi kèm với bộ nhớ flash hợp lý để lưu trữ chương trình và dữ liệu. Nó hỗ trợ giao thức TCP/IP và có khả năng kết nối với Wi-Fi mạng, giúp nó trở thành một phổ biến lựa chọn cho các ứng dụng IoT (Internet of Things) và các dự án điện tử thông minh.

ESP8266 có nhiều phiên bản và mô-đun khác nhau như ESP-01, ESP-12E, ESP-12F và ESP-12S. Mỗi bản có số lượng chân và tính năng kết nối khác nhau, nhưng chúng đều cung cấp khả năng kết nối Wi-Fi đáng tin cậy ở phiên bản và khả năng cài đặt hoạt động.

Cấu tạo của NodeMCU ESP8266

Module ESP8266 có các chân dùng để cấp nguồn và thực hiện kết nối.Chức năng của các chân như sau:

* Bộ vi điều khiển: CPU RISC 32-bit Tensilica Xtensa LX106
* Điện áp hoạt động: 3.3V
* Điện áp đầu vào: 7-12V
* Chân I / O kỹ thuật số (DIO): 16
* Chân đầu vào tương tự (ADC): 1
* UARTs: 1
* SPI: 1
* I2Cs: 1
* Bộ nhớ Flash: 4 MB
* SRAM: 64 KB
* Tốc độ đồng hồ: 80 MHz
* USB-TTL dựa trên CP2102 được bao gồm trên bo mạch, cho phép Plugn Play
* Ăng-ten PCB
* Mô-đun có kích thước nhỏ để phù hợp thông minh bên trong các dự án IoT của bạn



Hình 2. 8 Sơ đồ chân kết nối ESP8266

Module ESP-12E kết hợp với firmware ESP8266 trên Arduino và thiết kế phần cứng giao tiếp tiêu chuẩn đã tạo nên NodeMCU, loại Kit phát triển ESP8266 phổ biến nhất trong thời điểm hiện tại. Với cách sử dụng, kết nối dễ dàng, có thể lập trình, nạp chương trình trực tiếp trên phần mềm Arduino, đồng thời tương tích với các bộ thư viện Arduino sẵn có.

Thông số kỹ thuật

* IC chính: ESP8266 Wifi SoC.
* Phiên bản firmware: NodeMCU Lua
* Chíp nạp và giao tiếp UART: CP2102.
* GPIO tương thích hoàn toàn với firmware Node MCU.
* Cấp nguồn: 5VDC MicroUSB hoặc Vin.
* GIPO giao tiếp mức 3.3VDC
* Tích hợp Led báo trạng thái, nút Reset, Flash.
* Tương thích hoàn toàn với trình biên dịch Arduino.
* Kích thước: 25 x 50 mm

### **2.3.2. Cảm biến nhiệt độ, độ ẩm**

DHT11 là một cảm biến kỹ thuật số để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kỳ bộ vi điều khiển vi nào như Arduino, Raspberry Pi, ... để đo độ ẩm và nhiệt độ ngay lập tức.

DHT11 là một cảm biến độ ẩm tương đối. Để đo không khí xung quanh, cảm biến này sử dụng một điện trở nhiệt và một cảm biến độ ẩm điện dung.

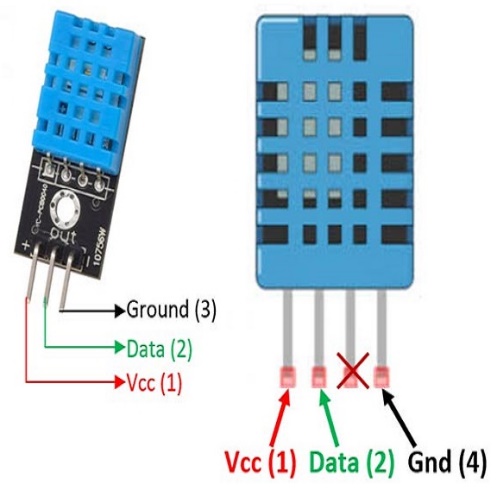
Cấu tạo cảm biến nhiệt độ độ ẩm DHT11

Cảm biến DHT11 bao gồm một phần tử cảm biến độ ẩm điện dung và một điện trở nhiệt để cảm nhận nhiệt độ. Tụ điện cảm biến độ ẩm có hai điện cực với chất nền giữ ẩm làm chất điện môi giữa chúng. Thay đổi giá trị điện dung xảy ra với sự thay đổi của các mức độ ẩm. IC đo, xử lý các giá trị điện trở đã thay đổi này và chuyển chúng thành dạng kỹ thuật số.

Để đo nhiệt độ, cảm biến này sử dụng một nhiệt điện trở có hệ số nhiệt độ âm, làm giảm giá trị điện trở của nó khi nhiệt độ tăng. Để có được giá trị điện trở lớn hơn ngay cả đối với sự thay đổi nhỏ nhất của nhiệt độ, cảm biến này thường được làm bằng gốm bán dẫn hoặc polymer.

|  |  |  |
| --- | --- | --- |
| Số chân | Tên chân | Mô tả |
| 1 | VCC | Nguồn 3.5V đến 5.5V |
| 2 | Data | Đầu ra cả nhiệt độ, độ ẩm thông qua dữ liệu nối tiếp |
| 3 | NC | Không có kết nối và do đó không có sử dụng |
| 4 | Ground | Nối đất |

Bảng 2. 2 Kết nối chân DHT11



Hình 2. 9 Sơ đồ chân DHT11

Cảm Biến Nhiệt Độ Và Độ Ẩm DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.

Thông số kỹ thuật DHT11

* Điện áp hoạt động: 3V - 5V DC
* Dòng điện tiêu thụ: 2.5mA
* Phạm vi cảm biến độ ẩm: 20% - 90% RH, sai số ±5%RH
* Phạm vi cảm biến nhiệt độ: 0°C ~ 50°C, sai số ±2°C
* Tần số lấy mẫu tối đa: 1Hz (1 giây 1 lần)
* Kích thước: 23 \* 12 \* 5 mm

### **2.3.3. Màn hình Oled 0.96inch**

OLED (viết tắt bởi Organic Light Emitting Diode: Diode phát sáng hữu cơ) đang trở thành đối thủ cạnh tranh cũng như ứng cử viên sáng giá thay thể màn hình LCD.

Màn hình OLED gồm những lớp như tấm nền, Anode, lớp hữu cơ, cathode. Và phát ra ánh sáng theo cách tương tự như đèn LED. Quá trình trên được gọi là phát lân quang điện tử.

Những ưu điểm có thể kể đến trên màn hình OLED là những lớp hữu cơ nhựa mỏng, nhẹ mềm dẻo hơn những lớp tinh thể trên LED hay LCD nhờ vậy mà có thể ứng dụng OLED để chế tạo màn hình gập cuộn được. Độ sáng của OLED cũng tốt hơn LED và không cần đèn nền như trên LCD nên sử dụng pin ít hơn. Góc nhìn cũng cài thiện hơn những công nghệ tiền nhiệm, khoảng 170 độ.



Hình 2. 10 Màn hình Oled 0.96inch

Thông số kĩ thuật màn hình OLED 0.96inch I2C

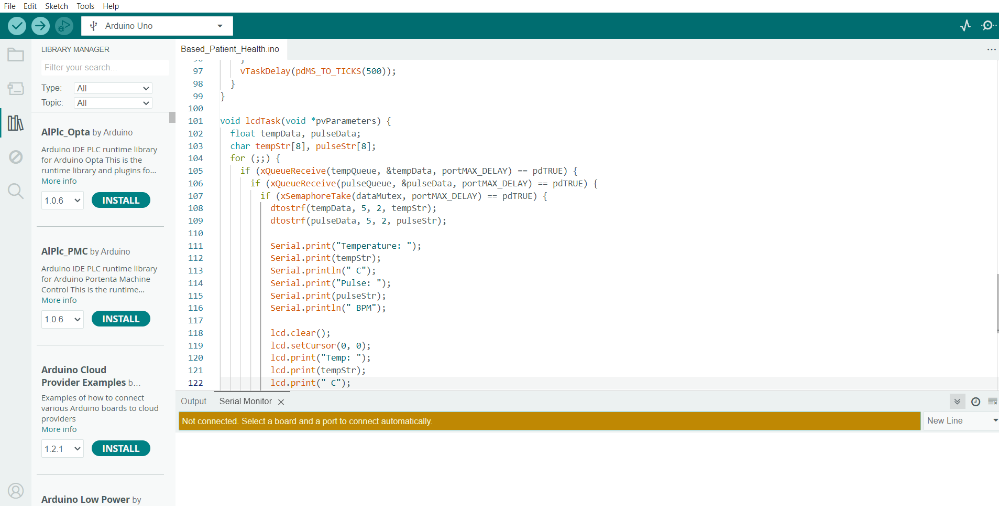
* Điện ấp sử dụng: 3V3 đến 5V (DC)
* Công suất tiêu thụ: 0.04W
* Góc hiển thị: Lớn hơn 160 độ (Em chưa hiểu chỗ này, bác nào giải thích hộ)
* Độ phân giải: 128X64 pixel (Điểm ảnh)
* Độ rộng màn hình: 0.96inch
* Giao tiếp: I2C
* Màu: Trắng và Đen
* Driver: SSD1306

### **2.3.4. Phần mềm thực hiện Arduino IDE**

Arduino IDE (Môi trường phát triển tích hợp) là một phần mềm được phát triển để thiết lập chương trình và tải chương trình cho bo mạch Arduino. Nó cung cấp một giao diện đơn giản và dễ sử dụng để viết, biên dịch và tải lên nguồn mã hóa cho Arduino dự án

Đặc điểm và tính năng:

* Giao diện đơn giản: Arduino IDE có giao diện người dùng thân thiện và dễ sử dụng. Nó cung cấp các công cụ cần thiết để viết nguồn mã hóa, xem kết quả biên dịch và tải lên chương trình trên bo mạch Arduino.
* Trình biên dịch và tải lên: Arduino IDE đi kèm với AVR-GCC biên dịch, cho phép biên dịch mã nguồn Arduino thành máy mã thích hợp để điều khiển AVR trên bo mạch Arduino. Nó cũng hỗ trợ chức năng tải chương trình lên bo mạch Arduino thông qua cổng USB hoặc các phương pháp khác.
* Mở rộng thư viện: Arduino IDE có phong phú mở rộng thư viện, cho phép người dùng truy cập và sử dụng các thư viện có sẵn để mở rộng chức năng của bo mạch Arduino. Thư viện bao gồm các tiện ích chức năng và các biến phổ chức năng như điều khiển đèn LED, giao tiếp tiếp theo cảm biến, điều khiển cơ sở và nhiều thứ khác nữa.
* Hỗ trợ trình cài đặt ngôn ngữ: Arduino IDE hỗ trợ trình cài đặt ngôn ngữ dựa trên Wiring, một ngôn ngữ dễ hiểu và dễ học được phát triển đặc biệt cho việc lập trình các dự án Arduino. Người dùng có thể viết mã nguồn Arduino bằng ngôn ngữ Wiring hoặc C/C++ thông thường.
* Môi trường thử nghiệm: Arduino IDE cung cấp một môi trường thử nghiệm hợp nhất, cho phép người dùng kiểm tra và gỡ lỗi chương trình trên bo mạch Arduino. Người dùng có thể sử dụng các công cụ như bộ theo dõi Serial Monitor để kiểm tra và ghi lại dữ liệu từ bo mạch Arduino.



Hình 2. 11 Giao diện phần mềm Arduino IDE

### **2.3.5. Ngôn ngữ lập trình C++**

C++ là một ngôn ngữ lập trình được phát triển vào năm 1979 tại phòng thí nghiệm Bell bởi Bjarne Stroustrup. Đây được gọi là ngôn ngữ bậc trung (middle-level) và là một phần mở rộng của ngôn ngữ lập trình C hoặc “C với các lớp Class” vì C++ là sự kết hợp các tính năng của ngôn ngữ cấp cao và cấp thấp.

Ngôn ngữ lập trình này có thể chạy trên nhiều nền tảng khác nhau như: Windows, macOS và Unix. Nó có nhiều tính năng như: lập trình tổng quát, lập trình thủ tục, lập trình hướng đối tượng,...

C++ có những điểm mạnh nổi bật như:

Tính phổ biến: Lập trình viên C++ là một trong những công việc được các nhà tuyển dụng trên toàn cầu tìm kiếm nhiều. Điều này có thể cho thấy C++ là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Dễ hiểu và dễ sử dụng: C++ được biết đến là ngôn ngữ lập trình đơn giản nhất. Lập trình này hoạt động theo nguyên tắc phân nhỏ các nhiệm vụ lớn thành các nhiệm vụ nhỏ cho bất kỳ dự án nào. Cấu trúc câu lệnh dễ hiểu, ít từ khóa nên khá dễ học và dễ sử dụng.

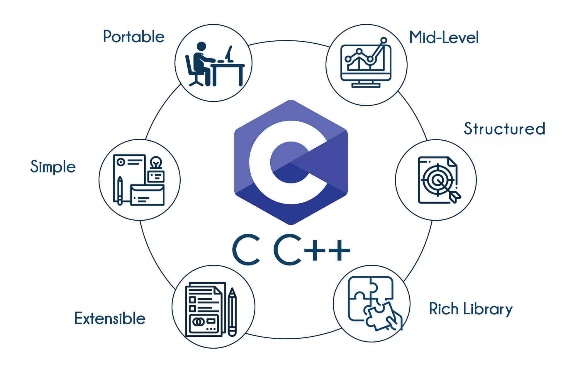
Thư viện đa dạng, phong phú: có nhiều tài nguyên hỗ trợ cho ngôn ngữ này như đồ họa API, 2D, 3D, vật lý các thiết bị âm… giúp cho các lập trình viên dễ thực thi.

Đa mô hình: C++ cũng cho phép bạn lập trình theo cấu trúc tuyến tính, hướng chức năng, hướng đối tượng đa dạng tùy theo yêu cầu của người lập trình.

Tính di động cao: một đoạn mã code C++ có thể chuyển đổi dễ dàng khi chạy trong macOS, Window hay Linux. Nhờ đó mà lập trình viên không phải tốn thời gian viết lại chương trình khi sử dụng nó để chạy trên nhiều platform khác nhau.

Có nhiều công cụ, phần mềm và IDE hỗ trợ giúp bạn đơn giản hóa công việc.

Chính vì sở hữu nhiều ưu điểm mà C++ đáng để học trong những ngôn ngữ lập trình.



Hình 2. 12 Ưu điểm của ngôn ngữ C++

2.7 Kết chương 2

*Trong chương này ta đã tìm hiểu về môn học hệ điều hành nhúng là gì, tìm hiểu về tổng quan FreeRTOS, cách thức hoạt động và các phần cứng cần có để thực hiện mạch. Phần mềm thực hiện và ngôn ngữ lập trình. Tiếp theo ta đến chương 3 để tìm hiểu sâu về quá trình thiết kế và nguyên lí hoạt động của mạch. Thực hiện kết nối thực tế.*

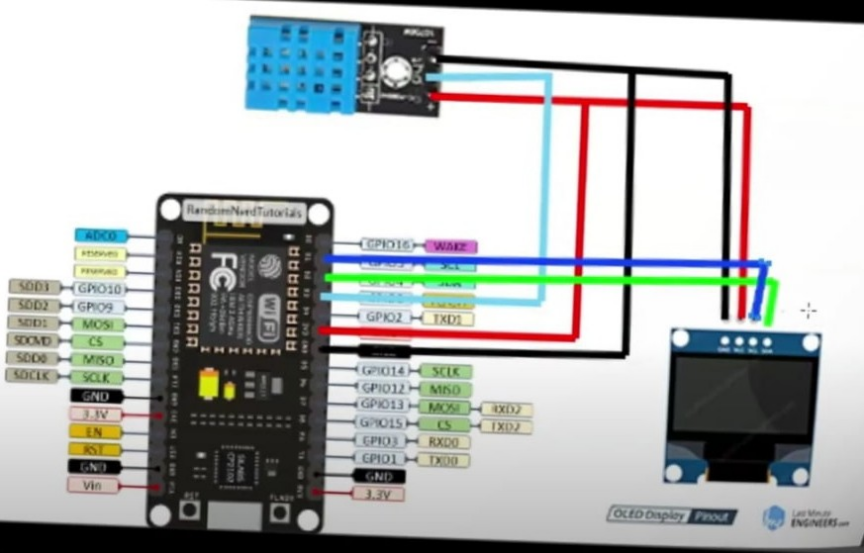
**CHƯƠNG 3: TRIỂN KHAI XÂY DỰNG**

**3.1. Danh sách thông số các linh kiện có trong mạch**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên linh kiện** | **Thông số kỹ thuật** |
| **1** | **Node MCU Esp8266** | **Chip Wi-Fi: ESP8266EX. Nguồn cấp: 3.0V ~ 3.6V DC** |
| **2** | **Cảm biến độ ẩm DHT11** | **Đầu ra cả nhiệt độ và độ ẩm thông qua dữ liệu nối tiếp. Nguồn 3.5V đến 5.5V** |
| **3** | **Màn hình oled 0.96inch** | **3V3 đến 5V** |

Bảng 3. 1 Linh Kiện

**3.2. Sơ đồ mạch**

****

Hình 3. 1 Sơ đồ mạch

### **3.2.1. Cách kết nối chân với nhau**

Kết nối ESP8266 và màn hình OLED:

* Chân SDA\_PIN (D2) của ESP8266 kết nối với chân SDA của màn hình OLED.
* Chân SDC\_PIN (D1) của ESP8266 kết nối với chân SDC của màn hình OLED.

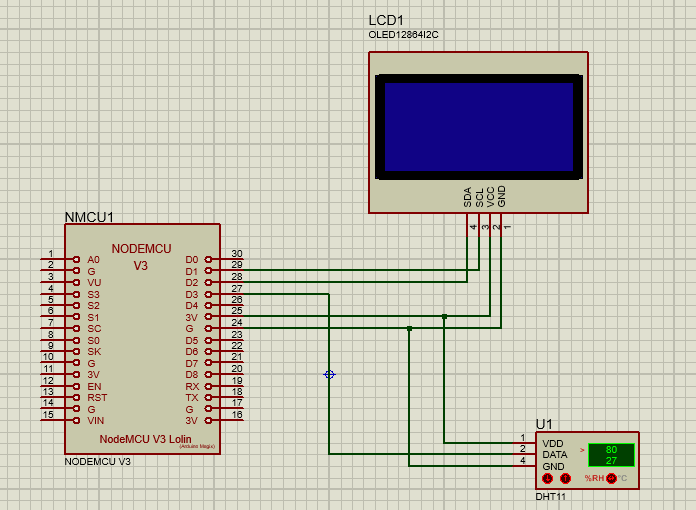
Kết nối ESP8266 và DHT11:

* Chân SDA\_PIN (D2) của ESP8266 kết nối với chân dữ liệu (Data Pin) của DHT11.
* Chân nguồn (Power Pin) của DHT11 kết nối với nguồn điện 3.3V trên ESP8266.

Kết nối màn hình OLED và DHT11:

* Chân SDA của màn hình OLED kết nối với chân dữ liệu (Data Pin) của DHT11.
* Chân SDC của màn hình OLED kết nối với chân đồng hồ (Clock Pin) của DHT11.

### **3.2.2. Sơ đồ mạch trên Protues**



Hình 3. 2 Sơ đồ trên Protues

## 3.3. Nguyên lý hoạt động của mạch

ESP8266: Đây là vi điều khiển WiFi có khả năng kết nối mạng và điều khiển các thiết bị khác. ESP8266 được sử dụng để giao tiếp với cảm biến DHT11 và điều khiển hiển thị trên màn hình OLED.

DHT11: Đây là cảm biến nhiệt độ và độ ẩm. Nó sử dụng một cảm biến nhiệt độ và một cảm biến độ ẩm để đo các thông số này trong môi trường xung quanh. DHT11 giao tiếp thông qua giao thức 1-Wire, trong đó ESP8266 đọc dữ liệu từ cảm biến.

Màn hình OLED: Đây là một loại màn hình hiển thị gồm các điểm ảnh nhỏ (pixel) sáng tỏ. Nó được sử dụng để hiển thị các thông tin như nhiệt độ, độ ẩm và các dữ liệu khác từ DHT11. Màn hình OLED sử dụng giao thức I2C để giao tiếp với ESP8266.

Cách hoạt động của mạch như sau:

ESP8266 kết nối với DHT11 thông qua chân dữ liệu (SDA\_PIN). ESP8266 gửi tín hiệu để yêu cầu dữ liệu từ DHT11.

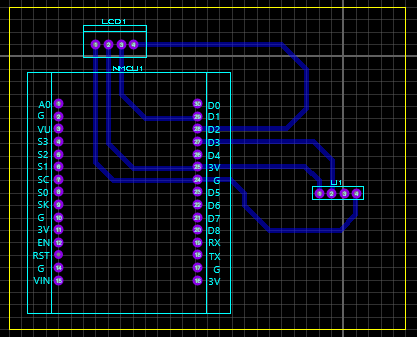
DHT11 đo nhiệt độ và độ ẩm trong môi trường xung quanh. Sau đó, nó truyền dữ liệu đo được về ESP8266 thông qua chân dữ liệu.

ESP8266 nhận dữ liệu từ DHT11 và sử dụng nó để xử lý hoặc hiển thị thông tin.

ESP8266 kết nối với màn hình OLED thông qua chân SDA\_PIN và SDC\_PIN. Nó gửi dữ liệu hiển thị (như nhiệt độ và độ ẩm đo được từ DHT11) đến màn hình OLED để được hiển thị.

Màn hình OLED nhận dữ liệu từ ESP8266 và hiển thị các thông tin tương ứng lên màn hình.

## 3.5. Mạch PCB

******

Hình 3. 3 Mạch PCB

## 3.6. Kết chương 3

*Qua quá trình xây dựng và thiết kế hệ thống, ta có thể hiểu sâu hơn về mô hình hệ thống, các thành phần trong ứng dụng, cách các thành phần kết nối với nhau, hiểu sâu hơn về cách chúng hoạt động.*

# **KẾT LUẬN**

1. **Kết quả đạt được**

Trong môn học "Hệ điều hành nhúng", nhóm đã tiến hành nghiên cứu và thực hiện đề tài "Hệ thống giám sát thời tiết sử dụng ESP8266, DHT11, màn hình OLED với thiết kế mạch PCB và ứng dụng hệ điều hành FreeRTOS." Dưới đây là những kết quả đạt được từ đề tài này:

Thiết kế mạch PCB: nhóm đã thành công trong việc thiết kế mạch PCB cho hệ thống giám sát thời tiết. Mạch PCB được thiết kế sao cho nhỏ gọn, tiết kiệm không gian và tối ưu hóa hiệu suất hoạt động. Các linh kiện như ESP8266, DHT11 và màn hình OLED được tích hợp trên mạch PCB một cách chính xác và hiệu quả.

Giao tiếp với ESP8266 và DHT11:nhóm đã thành công trong việc thiết lập giao tiếp giữa mạch PCB và các linh kiện chính như ESP8266 và cảm biến nhiệt độ/độ ẩm DHT11. ESP8266 được sử dụng để kết nối với mạng Wi-Fi và truyền dữ liệu thời tiết, trong khi DHT11 thu thập dữ liệu nhiệt độ và độ ẩm từ môi trường xung quanh.

Ứng dụng hệ điều hành FreeRTOS:nhóm đã triển khai hệ điều hành nhúng FreeRTOS trên hệ thống giám sát thời tiết. FreeRTOS cho phép quản lý nhiều tác vụ đồng thời và đáp ứng các yêu cầu thời gian thực. Việc sử dụng FreeRTOS giúp tăng tính ổn định và đáng tin cậy của hệ thống, đồng thời tối ưu hóa việc sử dụng tài nguyên.

Hiển thị dữ liệu trên màn hình OLED: nhóm đã thành công trong việc hiển thị dữ liệu thời tiết lên màn hình OLED. Màn hình OLED cung cấp một giao diện trực quan và dễ đọc cho người dùng, hiển thị thông tin như nhiệt độ, độ ẩm và các thông báo thời tiết.

Đáng tin cậy và hiệu suất cao: Kết quả đạt được từ đề tài này là hệ thống giám sát thời tiết đáng tin cậy và hiệu suất cao. Nhờ sử dụng các công nghệ như ESP8266, DHT11, màn hình OLED và hệ điều hành FreeRTOS, hệ thống có khả năng thu thập và hiển thị dữ liệu thời tiết một cách chính xác và liên tục.  
2. Hướng nghiên cứu

Tích hợp thêm các cảm biến và chức năng: Để mở rộng khả năng giám sát và phân tích dữ liệu thời tiết, có thể xem xét việc tích hợp thêm các cảm biến khác như cảm biến ánh sáng, cảm biến độ ồn, hoặc cảm biến khí quyển. Điều này sẽ cung cấp thông tin phong phú hơn về môi trường xung quanh và giúp phân tích các yếu tố khác nhau ảnh hưởng đến thời tiết.

Phát triển giao diện người dùng: Hiện tại, hệ thống giám sát thời tiết của chúng tôi đã có màn hình OLED để hiển thị dữ liệu. Tuy nhiên, có thể xem xét phát triển giao diện người dùng tương亲nắm với các tính năng bổ sung như xem dự báo thời tiết, lưu trữ dữ liệu lịch sử hoặc tùy chỉnh cài đặt. Điều này giúp cải thiện trải nghiệm người dùng và tăng tính tương tác của hệ thống.

Kết nối đám mây và Internet: Một hướng phát triển quan trọng là kết nối hệ thống giám sát thời tiết với đám mây và Internet. Việc này giúp chia sẻ dữ liệu thời tiết với người dùng từ xa, cập nhật dữ liệu liên tục, hoặc phân tích dữ liệu trên nền tảng đám mây. Sử dụng giao thức MQTT hoặc RESTful API có thể giúp thiết lập kết nối đám mây và truyền dữ liệu thời tiết một cách dễ dàng.

Tối ưu hóa tiêu thụ năng lượng: Trong các ứng dụng nhúng, tiêu thụ năng lượng là một yếu tố quan trọng. Hướng phát triển tiếp theo có thể tập trung vào tối ưu hóa tiêu thụ năng lượng của hệ thống giám sát thời tiết. Điều này có thể đạt được bằng cách sử dụng các kỹ thuật tiết kiệm năng lượng như chế độ ngủ cho các linh kiện không hoạt động hoặc sử dụng các linh kiện thân thiện với năng lượng như cảm biến nhiệt độ và độ ẩm tiết kiệm năng lượng.

Mở rộng ứng dụng và tích hợp hệ thống: Đề tài hiện tại tập trung vào việc giám sát thời tiết, tuy nhiên, có thể xem xét mở rộng ứng dụng và tích hợp hệ thống với các ứng dụng và thiết bị khác. Ví dụ, có thể tích hợp hệ thống giám sát thời tiết với các hệ thống tự động hoá nhà thông minh, hệ thống tưới cây tự động, hoặc ứng dụng trong lĩnh vực nông nghiệp. Điều này sẽ mang đến sự linh hoạt và ứng dụng rộng rãi cho hệ thống.

Nghiên cứu và đánh giá hiệu suất hệ thống: Để đảm bảo hiệu suất tốt của hệ thống giám sát thời tiết, có thể nghiên cứu và đánh giá hiệu suất của nó. Điều này bao gồm việc đo lường độ chính xác của các cảm biến, thời gian phản hồi của hệ thống và khả năng chịu tải. Nghiên cứu hiệu suất sẽ giúp cải thiện và tối ưu hóa hệ thống để đáp ứng tốt hơn với yêu cầu thực tế.

**TÀI LIỆU THAM KHẢO**

1. **Syhaunguyen.** *GiaotrinhAltium.* HaNoi : s.n., 2022.

2. **LDNam.** LDNam.Net. *LDNam.Net.* [Online] 12 24, 2020.

https://so.ldnam.net/index.php/tai-file-mach-bao-dong-da-nang-ldnam-195.

3. **GROUP, 3M.** chotroihn. *Huongdanlammach.* [Online] 11 21, 2017. https://chotroihn.vn/huong-dan-mach-bat-tat-thiet-bi-tu-dong-theo-nhiet-do.

4. **elec291.** *Introduction to PCB layout.* Chicago : s.n., 2020.

5. Simon Monk**.** Programming ESP8266 with Arduino IDE

https://help.ubidots.com/en/articles/928408-program-the-esp8266-with-the-arduino-ide-in-3-simple-steps

6. **Muhammad.** [Online] . IoT Patient Health Monitoring with ESP8266 & Arduino. *How To Electronics.* [Online] https://how2electronics.com/iot-patient-health-monitoring-with-esp8266-arduino/.

**7. Jonathan W. Valvano.** Embedded Systems: Real-Time Operating Systems forArm Cortex-M Microcontrollers

https://faculty.uobasrah.edu.iq/uploads/1631127575.pdf