**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**

-----------------⸙∆⸙-----------------



ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ THI CÔNG TRỤ SẠC XE MÁY ĐIỆN KẾT NỐI QUA HỆ THỐNG IOT ĐỂ ĐIỀU KHIỂN VÀ GIÁM SÁT TRÊN ĐIỆN THOẠI

**GVHD: TS. Nguyễn Văn Thái**

**SVTH: MSSV:**

**Ngô Xuân Thức 21151360**

**Tp. Hồ Chí Minh tháng 6 năm 2025**

**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**

-----------------⸙∆⸙-----------------



ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ THI CÔNG TRỤ SẠC XE MÁY ĐIỆN KẾT NỐI QUA HỆ THỐNG IOT ĐỂ ĐIỀU KHIỂN VÀ GIÁM SÁT TRÊN ĐIỆN THOẠI

**GVHD: TS. Nguyễn Văn Thái**

**SVTH: MSSV:**

**Ngô Xuân Thức 21151360**

**Tp. Hồ Chí Minh tháng 6 năm 2025**

LỜI CẢM ƠN

Nhóm xin chân thành gửi lời cảm ơn đến Thầy – Tiến sĩ Nguyễn Văn Thái, giảng viên hướng dẫn trực tiếp đã tạo điều kiện cho nhóm được thực hiện đề tài, tận tình giúp đỡ, đề xuất nội dung thực hiện, đưa ra các ý tưởng và giải pháp giúp nhóm vượt qua những khó khăn trong quá trình nhóm thực hiện đề tài.

Nhóm xin chân thành gửi lời cảm ơn đến tập thể các thành viên thuộc phòng nghiên cứu 3DvisionLab đã nhiệt tình hỗ trợ nhóm hoàn thành đồ án tốt nghiệp.

Nhóm chân thành cảm ơn những người bạn người anh cùng khoa Điện – Điện tử đã hỗ trợ nhóm rất nhiều trong quá trình thực hiện đồ án tốt nghiệp.

Nhóm xin cảm ơn các thầy các cô thuộc Khoa Điện – Điện tử trường Đại học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh. Đặc biệt là các thầy cô thuộc bộ môn Điều khiển Tự động đã chỉ bảo giảng dạy cho chúng em các kiến thức cơ sở ngành và chuyên nghành, là nền tảng kiến thức để chúng em áp dụng hoàn thành đồ án

Chúng con xin cảm ơn bố mẹ và gia đình đã động viên con về mặt tinh thần cũng như là tài chính trong những lúc khó khăn, tạo điều kiện thuận lợi nhất cho chúng con hoàn thành đồ án tốt nghiệp.

LỜI CAM ĐOAN

Nhóm xin cam đoan đề tài “***Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại thông minh***” là nhóm tự   
thực hiện, tham khảo các tài liệu trước đó, cùng với sự hỗ trợ giúp đỡ của   
Thầy - TS Nguyễn Văn Thái và hoàn toàn không sao chép từ bất cứ tài liệu hay   
công trình nghiên cứu khác.

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2025

Sinh viên thực hiện 1

(Ký và ghi rõ họ tên)

Ngô Xuân Thức

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  TRƯỜNG ĐẠI HỌC SPKT TP.HCM  KHOA ĐIỆN – ĐIỆN TỬ  BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN | CỘNG HÓA XÃ HỘI CHỦ NGHĨA VIỆT NAM  ĐỘC LẬP – TỰ DO – HẠNH PHÚC  ----o0o---- |

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**Họ và tên sinh viên:** Ngô Xuân Thức **MSSV:** 21151360

**Ngành:** Công nghệ kỹ thuật Điều khiển và Tự động hóa. **Lớp:** 211511C

**Hệ đào tạo:** Đại học chính quy **Khóa:** 2021

**Giảng viên hướng dẫn:** TS. Nguyễn Văn Thái.

**Ngày nhận đề tài: 1/2/2025** **Ngày nộp đề tài: 4/7/2025**

1. Tên đề tài: ***Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại .***

**Nội dung thực hiện đề tài:**

* Thiết kế được board mạch IoT điều khiển và giám sát sạc. Có khả năng thu thập dữ liệu từ đồng hồ điện với các thông số: Dòng điện, điện áp, công suất.
* Hiển thị dữ liệu thông tin sạc xe lên App.
* Lập trình nhúng IoT cho board mạch giao tiếp vời màn hình. Thực hiện tính toán điện năng tiêu thụ tải.

**GIẢNG VIÊN HƯỚNG DẪN BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  TRƯỜNG ĐẠI HỌC SPKT TP.HCM  KHOA ĐIỆN – ĐIỆN TỬ  BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN | CỘNG HÓA XÃ HỘI CHỦ NGHĨA VIỆT NAM  ĐỘC LẬP – TỰ DO – HẠNH PHÚC  ----o0o---- |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Họ và tên sinh viên:** Ngô Xuân Thức **MSSV:** 21151360

**Ngành:** Công nghệ kỹ thuật Điều khiển và Tự động hóa. **Lớp:** 211511C

**Giáo viên hướng dẫn:** TS. Nguyễn Văn Thái

Tên đề tài: ***Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại .***

**1. Nội dung đề tài và khối lượng công việc thực hiện**: *Hoàn thành các nội dung đặt ra của đề tài*. *Cụ thể là:*

* Thiết kế được board mạch IoT điều khiển và giám sát sạc. Có khả năng thu thập dữ liệu từ đồng hồ điện với các thông số: Dòng điện, điện áp, công suất.
* Hiển thị dữ liệu thông tin sạc xe lên App.
* Lập trình nhúng IoT cho board mạch giao tiếp với màn hình. Thực hiện tính toán điện năng tiêu thụ tải.
* Thiết kế trụ sạc xe điện với 2 đầu cắm. Có thể sạc cùng lúc 2 phương tiện.

**2. Hạn chế:**

3. Đề nghị bảo vệ hay không:

4. Đánh giá loại:

5. Điểm: Bằng chữ:

Tp. Hồ Chí Minh, ngày 30 tháng 6 năm 2023

Giảng viên hướng dẫn

(Ký và ghi rõ họ và tên)

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  TRƯỜNG ĐẠI HỌC SPKT TP.HCM  KHOA ĐIỆN – ĐIỆN TỬ  BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN | CỘNG HÓA XÃ HỘI CHỦ NGHĨA VIỆT NAM  ĐỘC LẬP – TỰ DO – HẠNH PHÚC  ----o0o---- |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

**Họ và tên sinh viên:** Ngô Xuân Thức **MSSV:** 21151360

**Ngành:** Công nghệ kỹ thuật Điều khiển và Tự động hóa. **Lớp:** 211511C

**Giáo viên phản biện:**

**Tên đề tài:** *Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại .*

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

**1. Về nội dung đề tài và khối lượng thực hiện:**

**2. Ưu điểm:**

**3. Khuyết điểm:**

**4: Đề nghị cho bảo vệ hay không?**

**5. Câu hỏi phản biện:**

**6. Điểm:**  **Bằng chữ:**

Tp. Hồ Chí Minh, ngày.... tháng.... năm 2025

**GIÁO VIÊN PHẢN BIỆN**

|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  TRƯỜNG ĐẠI HỌC SPKT TP.HCM  KHOA ĐIỆN – ĐIỆN TỬ  BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN | CỘNG HÓA XÃ HỘI CHỦ NGHĨA VIỆT NAM  ĐỘC LẬP – TỰ DO – HẠNH PHÚC  ----o0o---- |

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: Ngô Xuân Thức MSSV: 21151360

Tên đề tài: ***Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại .***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | NGÀY/TUẦN | NỘI DUNG SV THỰC HIỆN | XÁC NHẬN GV |
| 1 | Tuần 1 24/2 – 3/3 | Gặp GVHD được phổ biến quy định, lựa chọn và xác nhận về đề tài. Lựa chọn ý tưởng thiết kế, thi công. |  |
| 2 | Tuần 2 3/3 – 10/3 | Kết nối phần cứng, lập trình giao tiếp mạch ESP32 với đồng hồ đo điện DDSU666. |  |
| 3 | Tuần 3, 4 10/3 – 24/3 | Nghiên cứu hệ thống tổng quát. Nghiên cứu nguyên lý cơ bản hoạt động RTOS, đọc tài liệu về màn hình DWIN, cảm biến SHT31, thẻ nhớ SD nguyên lý hoạt động, tìm hiểu cách giao tiếp với chúng. |  |
| 4 | Tuần 5 24/3 – 31/3 | Thiết kế mạch nguyên lý, layout và đặt PCB. |  |
| 5 | Tuần 6 31/3 – 7/4 | Thiết kế giao diện trên màn hình DWIN HMI, giao tiếp màn hình với vi điều khiển ESP32. |  |
| 6 | Tuần 7, 8 7/4 – 21/4 | Viết thư viện đồng hồ đo điện DDSU666, SHT31 và SD Card. Viết API để sử dụng cảm biến. |  |
| 7 | Tuần 9, 10 21/5 – 5/5 | Nhận PCB, hàn mạch, kiểm tra và sửa lỗi mạch. Viết thư viện cho màn hình DWIN HMI. |  |
| 8 | Tuần 11 5/5 – 12/5 | Hiển thị các dữ liệu đồng hồ đo điện, các cảm biến lên màn hình DWIN, tiến hành Post dữ liệu lên App. |  |
| 9 | Tuần 12 12/5 – 19/5 | Thiết kế phần trụ sạc xe điện. Sửa lỗi phần giao tiếp giữa đồng hồ DDSU666 và ESP32. Viết hàm tính toán điện năng tiêu thụ. |  |
| 10 | Tuần 13 19/5 – 26/5 | Lắp ráp trụ sạc, sửa lỗi phần mềm. Lập trình phần sạc ONLINE quét mã QR |  |
| 11 | Tuần 14 26/5 – 2/6 | Thực nghiệm sạc xe. Điều khiển trạm sạc tính toán điện năng tiêu thụ và post dữ liệu sạc lên app |  |
| 12 | Tuần 15 2/6 – 9/6 | Viết báo cáo. |  |

Giảng viên hướng dẫn

(Ký và ghi rõ họ và tên)

MỤC LỤC

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc139312094)

[1.1. Đặt vấn đề 1](#_Toc139312095)

[1.2. Mục tiêu nghiên cứu 2](#_Toc139312096)

[1.3. Phương pháp nghiên cứu 2](#_Toc139312097)

[1.4. Giới hạn đề tài 2](#_Toc139312098)

[1.5. Nội dung nghiên cứu 2](#_Toc139312099)

[1.6. Bố cục báo cáo 3](#_Toc139312100)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 5](#_Toc139312101)

[2.1. Tổng quan về IoT (Internet of things) 5](#_Toc139312102)

[2.1.1. Định nghĩa 5](#_Toc139312103)

[2.1.2. Ứng dụng của IoT 6](#_Toc139312108)

[2.2. Công nghệ Wi-Fi (Wireless Fidelity) 8](#_Toc139312109)

[2.2.1. Giới thiệu 8](#_Toc139312110)

[2.2.2. Hoạt động của Wi-Fi: 9](#_Toc139312111)

[2.2.3. Ưu, nhược điểm của Wi-Fi: 10](#_Toc139312112)

[2.3. Công nghệ truyền thông LoRa 10](#_Toc139312115)

[2.3.1. Nguyên lý hoạt động 11](#_Toc139312116)

[2.3.2. Cấu trúc gói tin 11](#_Toc139312117)

[2.3.3. Các thông số 12](#_Toc139312118)

[2.4. Chuẩn giao tiếp SPI, I2C, UART 13](#_Toc139312122)

[2.4.1. Chuẩn giao tiếp SPI 13](#_Toc139312123)

[2.4.2. Chuẩn giao tiếp I2C 15](#_Toc139312127)

[2.4.3. Chuẩn giao tiếp UART 18](#_Toc139312131)

[2.5. Các vi điều khiển 19](#_Toc139312135)

[2.5.1. ESP32 – D0WD – V3 19](#_Toc139312136)

[2.5.2. STM32F103C8T6 22](#_Toc139312137)

[2.5.3. ESP8266 24](#_Toc139312138)

[2.6. Giao thức HTTP 26](#_Toc139312141)

[2.6.1. Tổng quan 26](#_Toc139312142)

[2.6.2. Các mã lỗi của HTTP: 26](#_Toc139312143)

[2.6.3. Các phương thức yêu cầu của HTTP: 27](#_Toc139312144)

[2.7. Cảm biến 27](#_Toc139312147)

[2.7.1. Cảm biến PMS7003 27](#_Toc139312148)

[2.7.2. Cảm biến SHT31 29](#_Toc139312151)

[2.7.3. Cảm biến MQ7 32](#_Toc139312154)

[2.8. Các phần mềm sử dụng trong đề tài 34](#_Toc139312157)

[2.8.1. Kicad 34](#_Toc139312158)

[2.8.2. Visual Studio Code 35](#_Toc139312159)

[2.8.3. ESP-IDF 35](#_Toc139312160)

[2.8.4. STM32CubeIDE 36](#_Toc139312161)

[CHƯƠNG 3. THIẾT KẾ HỆ THỐNG 37](#_Toc139312162)

[3.1. Yêu cầu hệ thống 37](#_Toc139312163)

[3.2. Thiết kế phần cứng 40](#_Toc139312166)

[3.2.1. Sơ đồ nguyên lý 40](#_Toc139312167)

[3.2.2. Layout PCB 46](#_Toc139312170)

[3.3. Thiết kế phần mềm 51](#_Toc139312173)

[3.3.1. Node 51](#_Toc139312174)

[3.3.2. Gateway 56](#_Toc139312178)

[CHƯƠNG 4. KẾT QUẢ 63](#_Toc139312184)

[4.1. Kết quả phần cứng 63](#_Toc139312185)

[4.1.1. Gateway 63](#_Toc139312186)

[4.1.2. Node 63](#_Toc139312187)

[4.2. Kết quả phần mềm 64](#_Toc139312188)

[4.3. Thực nghiệm độ chính xác của thiết bị với máy đo trên thị trường 68](#_Toc139312192)

[4.4. Thực nghiệm độ ổn định của hệ thống với khoảng cách trên 1000m 70](#_Toc139312193)

[4.5. Thực nghiệm tính linh hoạt của hệ thống 74](#_Toc139312194)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 78](#_Toc139312195)

[5.1. Kết quả đạt được 78](#_Toc139312196)

[5.2. Hướng phát triển 78](#_Toc139312199)

DANH MỤC HÌNH ẢNH

[Hình 2. 1 Tổng quan về “IoT – Internet of things” 4](#_Toc199859076)

[Hình 2. 2 Mô hình ứng dụng của IoT trong y tế 6](#_Toc199859077)

[Hình 2. 3 Trụ sạc xe máy điện do VinFast phát triển 8](#_Toc199859078)

[Hình 2. 4 Mô hình giao tiếp SPI giữa Master và các Slave 10](#_Toc199859079)

[Hình 2. 5 Giản đồ xung clock SPI trường hợp CPHA = 0 và CPHA =1 11](#_Toc199859080)

[Hình 2. 6 Khung truyền của giao thức I2C 13](#_Toc199859081)

[Hình 2. 7 Truyền dữ liệu từ DATA BUS đến UART1 15](#_Toc199859082)

[Hình 2. 8 Quá trình truyền dữ liệu UART 15](#_Toc199859083)

[Hình 2. 9 Khung truyền của giao tiếp UART 16](#_Toc199859084)

[Hình 2. 10 Sơ đồ khối chức năng của ESP32-WROOM-32E-N4 22](#_Toc199859085)

[Hình 2. 11 Sơ đồ chân ESP32-WROOM-32E-N4 23](#_Toc199859086)

[Hình 2. 12 IC MAX3485 24](#_Toc199859087)

[Hình 2. 13 IC DS3231 25](#_Toc199859088)

[Hình 2. 14 Cảm biến SHT31 27](#_Toc199859089)

[Hình 2. 15 Sơ đồ khối cảm biến SHT31 28](#_Toc199859090)

[Hình 2. 16 Hình ảnh mặt dưới của SHT31 28](#_Toc199859091)

[Hình 2. 17 Địa chỉ cảm biến SHT31 29](#_Toc199859092)

[Hình 2. 18 Thanh ghi lựa chọn chế độ lặp lại và đồng hồ kéo dài 29](#_Toc199859093)

[Hình 2. 19 Giao thức truyền dữ liệu của SHT31 30](#_Toc199859094)

[Hình 2. 20 Đồng hồ đo điện năng DDSU666 31](#_Toc199859095)

[Hình 2. 21 Sơ đồ khối tổng quát của DDSU666 31](#_Toc199859096)

[Hình 2. 22 Phần mềm EAGLE 32](#_Toc199859097)

[Hình 2. 23 Visual Studio Code 33](#_Toc199859098)

[Hình 2. 24 Arduino IDE 33](#_Toc199859099)

[Hình 2. 25 DGUS\_V7.647 34](#_Toc199859100)

[Hình 2. 26 SolidWorks 34](#_Toc199859101)

[Hình 3. 1 Sơ đồ khối phần cứng của trụ sạc xe điện 36](#_Toc199859145)

[Hình 3. 2 Sơ đồ nguyên lý mạch chuyển đổi nguồn 220VAC sang 5VDC và 3.3VDC 37](#_Toc199859146)

[Hình 3. 3 Sơ đồ nguyên lý mạch ESP32-WROOM-32E-N4 38](#_Toc199859147)

[Hình 3. 4 Sơ đồ nguyên lý mạch đọc dữ liệu RS485 38](#_Toc199859148)

[Hình 3. 5 Sơ đồ nguyên lý DS3231 39](#_Toc199859149)

[Hình 3. 6 Sơ đồ nguyên lý cụm cảm biến 40](#_Toc199859150)

[Hình 3. 7 Sơ đồ nguyên lý thẻ nhớ SD Card 40](#_Toc199859151)

[Hình 3. 8 Sơ đồ nguyên lý khối điều khiển Relay 41](#_Toc199859152)

[Hình 3. 9 Sơ đồ nguyên lý khối chống kích relay khi reset 42](#_Toc199859153)

[Hình 3. 10 Mặt trên PCB board điều khiển trụ sạc xe máy điện 43](#_Toc199859154)

[Hình 3. 11 Mặt dưới PCB board điều khiển trụ sạc xe máy điện 43](#_Toc199859155)

[Hình 3. 12 Giao diện HMI ở chế độ sạc ONLINE 45](#_Toc199859156)

[Hình 3. 13 Giao diện HMI ở chế độ sạc OFFLINE 45](#_Toc199859157)

[Hình 3. 14 Giao diện HMI lựa chọn phương thức sạc khi ở chế độ OFFLINE 46](#_Toc199859158)

[Hình 3. 15 Giao diện HMI khi đang sạc OFFLINE cổng 1 47](#_Toc199859159)

[Hình 3. 16 Giao diện HMI khi đang sạc ONLINE cổng 2 47](#_Toc199859160)

[Hình 3. 17 Giao diện HMI khi kết thúc sạc ở cổng 1 48](#_Toc199859161)

[Hình 3. 18 Giao diện HMI hiển thị và cài đặt mật khẩu cho trụ sạc 48](#_Toc199859162)

[Hình 3. 19 Giao diện HMI hiển thị và thay đổi giá điện 49](#_Toc199859163)

[Hình 3. 20 Giao diện HMI hiển thị và thay đổi Wi Fi 49](#_Toc199859164)

[Hình 3. 21 Giao diện HMI thông báo lỗi nhập sai mật khẩu 50](#_Toc199859165)

[Hình 3. 22 Giao diện HMI thông báo lỗi nhập sai định dạng 50](#_Toc199859166)

[Hình 3. 23 Giao diện HMI thông báo lỗi nhập số tiền không hợp lệ 51](#_Toc199859167)

[Hình 4. 1 Mạch GateWay 63](#_Toc139312644)

[Hình 4. 2 Mặt trước của mạch Node 63](#_Toc139312645)

[Hình 4. 3 Mặt sau của mạch Node 64](#_Toc139312646)

[Hình 4. 4 Các Node được kết nối thành công vào mạng LoRa 64](#_Toc139312647)

[Hình 4. 5 Gateway nhận gói tin cấu hình từ Node 65](#_Toc139312648)

[Hình 4. 6 Gateway nhận các giá trị cảm biến trả về từ các Node 65](#_Toc139312649)

[Hình 4. 7 POST dữ liệu cảm biến lên Server 65](#_Toc139312650)

[Hình 4. 8 Giao diện LCD mạch Node 66](#_Toc139312651)

[Hình 4. 9 Giao diện hiển thị giá trị nhiệt độ, độ ẩm của mạch Node 66](#_Toc139312652)

[Hình 4. 10 Giao diện hiển thị bụi mịn PM2.5, nồng độ khí CO của mạch Node 67](#_Toc139312653)

[Hình 4. 11 Giao diện app IoTVision của Node 480F và C477 67](#_Toc139312654)

[Hình 4. 12 Máy đo Fujie 68](#_Toc139312655)

[Hình 4. 13 Phạm vi thử nghiệm đạt 2,69 km2 đo theo Google Map 71](#_Toc139312656)

[Hình 4. 14 Phạm vi truyền nhận đo theo bản đồ vệ tinh 71](#_Toc139312657)

[Hình 4. 15 Khoảng cách từ Gateway đến các Node đo theo Google Map 72](#_Toc139312658)

[Hình 4. 16 Giá trị cảm biến của các Node tại các vị trí được đánh dấu trên bản đồ 72](#_Toc139312659)

[Hình 4. 17 Giá trị cảm biến của các Node được hiển thị trên app 73](#_Toc139312660)

[Hình 4. 18 Bảng liệt kê số lần truyền nhận thành công 74](#_Toc139312661)

[Hình 4. 19 Những vị trí không thể truyền nhận do mật độ vật cản cao 74](#_Toc139312662)

[Hình 4. 20 Kết nối Access Point của mạch Node 75](#_Toc139312663)

[Hình 4. 21 Truy cập vào Web Server của mạch Node 76](#_Toc139312664)

[Hình 4. 22 Nhập chuỗi CONFIG LORA 77](#_Toc139312665)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Bảng mô tả các chân cảm biến PMS7003 28](#_Toc139317758)

[Bảng 2. 2 Bảng mô tả các chân cảm biến SHT31 31](#_Toc139317759)

[Bảng 4. 1 Mẫu giá trị cảm biến của máy đo Fujie 68](#_Toc139317764)

[Bảng 4. 2 Mẫu giá trị cảm biến của Node 1 69](#_Toc139317765)

[Bảng 4. 3 Mẫu giá trị cảm biến của Node 2 69](#_Toc139317766)

[Bảng 4. 4 Mẫu giá trị cảm biến của Node 3 69](#_Toc139317767)

[Bảng 4. 5 Mẫu giá trị cảm biến của Node 4 70](#_Toc139317768)

DANH MỤC CÁC TỪ VIẾT TẮT

IoT: Internet of Things.

LoRa: Long Range.

LPWAN: Low Power Wide Area Network

SPI: Serial Peripheral Interface.

I2C: Inter – Integrated Circuit.

UART: Universal Asynchronous Receiver-Transmitter.

IC: Integrated Circuit.

IDE: Integrated Development Environment.

MCU: Micro Controller Unit.

SoC: System on Chip.

Wi-Fi: Wireless Fidelity

CRC: Cyclic Redundacy Check.

SF: Spreading Factor.

BW: Bandwidth.

CR: Coding Rate.

SCL: Serial Clock Line

SDA: Serial Data Line.

CO: Carbon Monoxide.

# TỔNG QUAN

## Đặt vấn đề

Trong bối cảnh biến đổi khí hậu ngày càng nghiêm trọng và nguồn nhiên liệu hóa thạch đang dần cạn kiệt, việc chuyển đổi sang các phương tiện giao thông sử dụng năng lượng sạch như xe máy điện đang trở thành xu hướng tất yếu trên toàn cầu, trong đó có Việt Nam. Xe máy điện không chỉ giúp giảm thiểu khí thải độc hại ra môi trường mà còn góp phần tiết kiệm chi phí vận hành và bảo trì cho người sử dụng. Tuy nhiên, một trong những thách thức lớn nhất hiện nay trong quá trình phổ cập xe máy điện là hạ tầng trạm sạc còn hạn chế, thiếu sự linh hoạt và khó quản lý hiệu quả.

Để giải quyết vấn đề này, nhóm tập trung thiết kế một hệ thống IoT (Internet of Things) trạm sạc xe máy điện ứng dụng việc tự động hóa, điều khiển từ xa và giám sát hệ thống theo thời gian thực. Việc ứng dụng IoT vào các trụ sạc xe máy điện sẽ góp phần nâng cao tính tiện ích, khả năng quản lý, và tối ưu hiệu suất vận hành hệ thống sạc, đồng thời mang lại trải nghiệm hiện đại, dễ sử dụng cho người dùng thông qua các thiết bị di động.

Xuất phát từ thực tiễn đó, đề tài “Thiết kế và thi công trụ sạc xe máy điện kết nối qua hệ thống IoT để điều khiển và giám sát trên điện thoại” được thực hiện với mục tiêu xây dựng một hệ thống trụ sạc thông minh, có khả năng giám sát điện áp, dòng điện, thời gian sạc, và trạng thái hoạt động từ xa thông qua ứng dụng trên điện thoại thông minh. Cùng với thao tác trên màn hình để người dùng có thể trực tiếp thao tác điều khiển việc sạc pin cho xe điện. Nâng cao trải nghiệm người dùng. Hệ thống sẽ giúp người dùng chủ động hơn trong việc kiểm soát quá trình sạc, đồng thời hỗ trợ nhà quản lý theo dõi, thống kê và bảo trì hệ thống một cách hiệu quả hơn.

## Mục tiêu nghiên cứu

Nghiên cứu, thiết kế và thi công thành công trụ sạc xe máy điện gồm 2 cổng sạc. Có thể lựa chọn sạc ở 2 chế độ ONLINE và OFFLINE. Ở chế độ sạc OFFLINE người dùng thao tác trên trực tiếp màn hình để lựa chọn sạc theo thực tế sạc hay sạc theo số tiền mặc định. Ở chế độ sạc ONLINE người dùng quét mã QR được in trên trụ sạc để sạc tự động theo đúng số tiền đã thanh toán.

## Phương pháp nghiên cứu

Đề tài sử dụng tổng hợp các phương pháp nghiên cứu sau:

* Phương pháp nghiên cứu lý thuyết.
* Phương pháp thực nghiệm.

## Giới hạn đề tài

Do thời gian thực hiện có hạn nên đề tài của nhóm còn có những giới hạn như sau:

* Chưa phát triển giao thức giao tiếp giữa các trụ sạc với nhau.
* Hệ thống đo đạc tính toán điện năng tiêu thụ chưa có sự kiểm định với các thiết bị chuyên dụng.
* Chưa có cơ chế tự động trả lại tiền thừa nếu sạc chưa hết số tiền nhập vào mà dung lượng pin đã đầy hoặc là mất kết nối điện đột ngột.

## Nội dung nghiên cứu

Nội dung nghiên cứu bao gồm:

* Nghiên cứu các mẫu trụ sạc xe máy điện trên thị trường hiện nay.
* Tham khảo các bài báo khoa học, tài liệu, công trình nghiên cứu có liên quan đến thiết kế và thi công trụ sạc xe máy điện. Nghiên cứu cách giao tiếp đọc thông tin từ đồng hồ đo điện năng tiêu thụ của tải.
* Nghiên cứu các tài liệu có liên quan đến hệ điều hành RTOS, thực hiện thử nghiệm thực tế hoạt động của hệ điều hành với vi điều khiển ESP32.
* Nghiên cứu và thiết kế board mạch IoT điều khiển, giám sát việc sạc xe và đưa thông tin sạc xe lên server.
* Tham khảo các tài liệu về cách thức hoạt động của các giao thức I2C, UART, RS485, SPI, POST, GET .
* Giao tiếp RS485 giữa MCU và đồng hồ đo điện năng DDSU666, viết thư viện giao tiếp với đồng hồ đo điện, cài đặt cấu hình các thanh ghi để đọc được giá trị đồng hồ trả về.
* Lập trình luồng hoạt động của giao tiếp người dùng với màn hình HMI.
* Hiển thị các thông số khi sạc để cho người dùng có thể theo dõi, như nhiệt độ, độ ẩm của trụ sạc, thời gian sạc, số KWh, công suất, số tiền sạc khách sẽ phải trả tại mỗi cổng
* Hiệu chỉnh, sửa lỗi, tối ưu phần cứng và phần mềm.

## Bố cục báo cáo

Chương 1: Tổng quan.

Chương này giới thiệu về đề tài sẽ nghiên cứu, bao gồm lý do chọn đề tài, phương pháp nghiên cứu, mục tiêu nghiên cứu, nội dung nghiên cứu và những giới hạn trong phạm vi của nghiên cứu của đề tài.

Chương 2: Cơ sở lý thuyết.

Phần cơ sở lý thuyết đề cập đến các nội dung về kiến thức, lý thuyết, chắc năng của các bộ sạc xe máy điện hiện nay. Giới thiệu tổng quan, trình bàu các đặc điểm của vi điều khiển ESP32, cảm biến SHT31. Đồng hồ đo điện DDSU666. Ngoài ra còn gốm các lý thuyết và thành phần chức năng của các chuẩn giao tiếp I2C, SPI, UART, RS485. Giới thiệu các phần mềm được sử dụng trong đề tài này.

Chương 3: Thiết kế hệ thống

Trình bày chi tiết về yêu cầu của hệ thống. Thiết kế phần cứng, sơ đồ nguyên lý mạch phần cứng, thiết kế phần cơ khí trụ sạc, sơ đồ trình tự thiết kế phần mềm trình tự thiết kế trụ sạc. Ngoài ra, nhóm thể hiện và giải thích lưu đồ giải thuật của phần mềm.

*Chương 4: Kết quả*

Chương này cho thấy các kết quả đạt được của đề tài. Thực nghiệm sạc xe ở chế độ ONLINE, chế độ OFFLINE. Sạc cùng lúc 2 thiết bị và post dữ liệu thông tin sạc lên trên App

Chương 5: Kết luận và hướng phát triển.

Trong chương này, nhóm rút ra kết luận từ những kết quả đạt được và đưa ra các hướng phát triển tiếp theo trong tương lai.

# 

# CƠ SỞ LÝ THUYẾT

## Tổng quan về IoT (Internet of things)

### Định nghĩa

IoT (Internet of Things hay Internet vạn vật) là khái niệm mô tả mạng lưới các thiết bị vật lý, phương tiện, thiết bị gia dụng và các mục khác được kết nối với Internet và có khả năng thu thập, chia sẻ dữ liệu với nhau. Các thiết bị này thường được trang bị cảm biến, phần mềm, và các công nghệ khác để giao tiếp và trao đổi dữ liệu một cách tự động. Sự kết hợp của các thiết bị thông minh này cho phép chúng tương tác với nhau và với con người thông qua mạng internet.

Điểm nổi bật của IoT là tính kết nối và khả năng tương tác giữa các thiết bị, cho phép chúng tự động hoạt động, đưa ra quyết định hoặc kích hoạt hành động dựa trên dữ liệu thu thập được. IoT có ứng dụng rộng rãi trong nhiều lĩnh vực như nhà thông minh, y tế, nông nghiệp, sản xuất, quản lý năng lượng và nhiều ứng dụng khác.



***Hình 2. 1*** *Tổng quan về “IoT – Internet of things”*

#### Thiết bị: Các thiết bị IoT bao gồm cảm biến, bộ điều khiển và bộ xử lý. Chúng có khả năng thu thập dữ liệu và gửi chúng thông qua mạng Internet.

* Cảm biến: Các cảm biến giúp thu thập dữ liệu từ môi trường xung quanh, như ánh sáng, nhiệt độ, độ ẩm, v.v..
* Bộ xử lý: Xử lý dữ liệu và thực hiện các tác vụ tính toán, cho phép đưa ra quyết định và hành động.
* Bộ điều khiển: Bộ điều khiển điều phối hoạt động của các thiết bị và quản lý truyền thông giữa chúng.

#### Kết nối mạng: Để đảm bảo khả năng truyền và nhận dữ liệu giữa các thiết bị, IoT dựa vào các giao thức mạng như Wi-Fi, Bluetooth, Zigbee, LoRaWAN và 5G. Việc kết nối mạng đóng vai trò then chốt trong việc giúp các thiết bị IoT tương tác và hoạt động đồng bộ với nhau. Nhờ vào sự phát triển nhanh chóng của công nghệ truyền thông, IoT ngày càng được ứng dụng phổ biến rộng rãi trong nhiều lĩnh vực như nhà thông minh, đô thị thông minh, nông nghiệp, y tế và các ngành công nghiệp khác.

#### Nền tảng phần mềm: Các nền tảng phần mềm hỗ trợ việc quản lý và phân tích dữ liệu từ các thiết bị IoT, cho phép ứng dụng và dịch vụ phát triển trên nền tảng này.

#### Dữ liệu: Dữ liệu thu thập từ các thiết bị IoT cung cấp thông tin để phân tích, khai thác và hỗ trợ quá trình đưa ra quyết định. Được ứng dụng trong giám sát và chuẩn đoán AI

### Ứng dụng của IoT

**Nhà thông minh:** IoT cho phép điều khiển và giám sát các thiết bị trong gia đình như đèn, điều hòa, camera, hệ thống an ninh thông qua điện thoại hoặc giọng nói, giúp nâng cao chất lượng sống, tiện nghi khi sử dụng và đảm bảo an toàn cho cuộc sống.

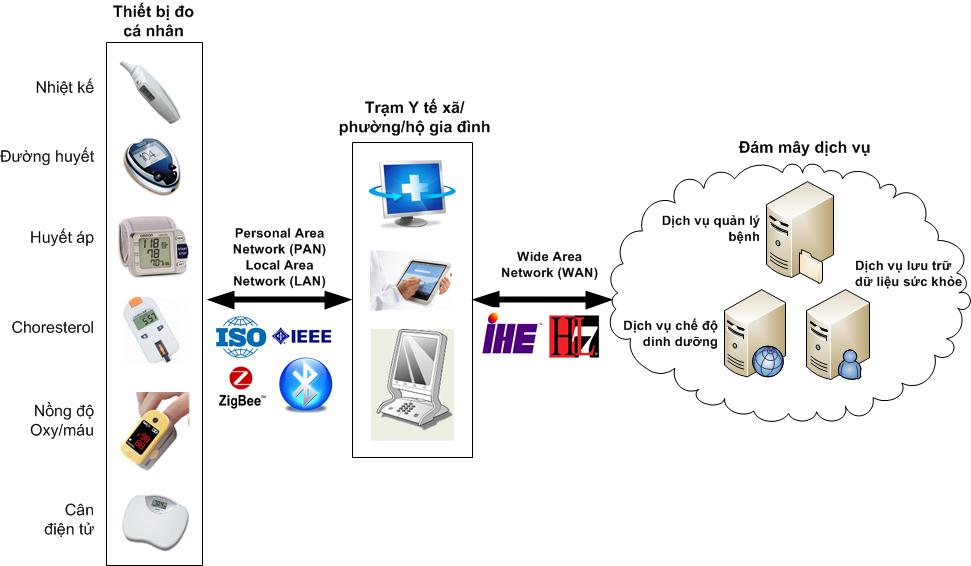
**Thành phố thông minh:** Hệ thống IoT được triển khai để quản lý giao thông, chiếu sáng công cộng, thu gom rác thải, và giám sát chất lượng không khí, góp phần xây dựng môi trường sống hiện đại, phát triển giảm thiểu các yếu tố ô nhiễm môi trường cũng như là lãng phí tài nguyên.

**Nông nghiệp thông minh:** IoT hỗ trợ giám sát điều kiện đất, nước, nhiệt độ, và tự động hóa quy trình tưới tiêu hoặc bón phân, giúp gia tăng năng suất và tiết kiệm tài nguyên. Bảo vệ con người khỏi tiếp xúc với các hoá chất độc hại.

**Y tế thông minh:** Thiết bị IoT theo dõi tình trạng sức khỏe bệnh nhân theo thời gian thực, hỗ trợ chẩn đoán sớm phát hiện và điều trị bênh. Nâng cao chất lượng chăm sóc y tế.

**Công nghiệp 4.0:** IoT được tích hợp trong các dây chuyền sản xuất để giám sát, thu thập dữ liệu và phân tích hiệu suất hoạt động, giúp tự động hóa và nâng cao chất lượng và hiệu quả sản xuất công nghiệp

**Giao thông thông minh:** IoT giúp theo dõi phương tiện, tối ưu hóa lộ trình, kiểm soát đèn tín hiệu giao thông và giảm thiểu ùn tắc, phát hiện sai phạm giao thông đảm bảo an toàn cho người tham gia giao thông.



***Hình 2. 2*** *Mô hình ứng dụng của IoT trong y tế*

Tuy nhiên, song song với sự phát triển mạnh mẽ và tiềm năng to lớn của IoT, vẫn tồn tại nhiều thách thức và vấn đề cần được chú trọng quan tâm, bao gồm:

**Bảo mật và quyền riêng tư:** Việc thu thập và truyền tải dữ liệu cá nhân qua mạng khiến các thiết bị IoT trở thành mục tiêu tiềm năng cho các cuộc tấn công mạng. Đảm bảo an toàn thông tin và bảo mật dữ liệu người dùng là một trong những thách thức lớn trong lĩnh vực IoT.

**Tương thích và chuẩn hóa:** Các thiết bị IoT đến từ nhiều nhà sản xuất khác nhau, sử dụng các chuẩn và giao thức không đồng nhất, gây khó khăn trong việc tích hợp và tương tác giữa các hệ thống đã xây dựng độc lập.

**Quản lý dữ liệu lớn:** IoT tạo ra một lượng dữ liệu khổng lồ, đòi hỏi phải có hệ thống lưu trữ, xử lý và phân tích hiệu quả để khai thác thông tin triệt để thu thập những thông tin có giá trị và bỏ đi các thông tin không cần thiết.

**Tiêu thụ năng lượng:** Nhiều thiết bị IoT hoạt động liên tục và yêu cầu tiêu thụ điện năng thấp, vì vậy việc thiết kế giải pháp tiết kiệm năng lượng là điều cần thiết.

**Chi phí triển khai và bảo trì:** Việc triển khai một hệ thống IoT hoàn chỉnh có thể tốn kém, đặc biệt là trong các ngành công nghiệp lớn, kèm theo đó là chi phí bảo trì và nâng cấp thường xuyên.

## Hệ thống trụ sạc cho xe máy điện

### Giới thiệu

Trụ sạc xe máy điện là thiết bị cung cấp điện xoay chiều (AC) từ lưới điện đến xe máy điện thông qua cổng sạc tiêu chuẩn. Khác với hệ thống sạc DC (nơi bộ sạc nằm trong trụ sạc), trụ sạc AC không thực hiện chuyển đổi dòng điện mà chỉ đóng vai trò là điểm cấp điện trung gian, cho phép bộ sạc bên trong xe thực hiện quá trình chuyển đổi AC sang DC để sạc pin.

Trong hệ thống này, bộ sạc chính nằm trong xe, còn trụ sạc chủ yếu chịu trách nhiệm về kết nối, cấp điện an toàn và hỗ trợ giám sát/điều khiển.



***Hình 2. 3*** *Trụ sạc xe máy điện do VinFast phát triển*

### Cấu trúc cơ bản của trụ sạc xe máy điện

Trụ sạc AC cho xe máy điện thường bao gồm các thành phần sau:

* **Nguồn cấp điện lưới**: Thường là 220V AC, 50Hz.
* **Ổ cắm/cổng sạc**: Tương thích với loại đầu sạc của xe (thường là loại 2 chấu hoặc loại chuyên dụng do nhà sản xuất xe cung cấp).
* **Công tắc/contactor điều khiển nguồn**: Dùng để bật/tắt nguồn điện cấp cho xe, có thể điều khiển trực tiếp bằng tay hoặc thông qua hệ thống IoT.
* **Bộ đo lường điện năng (energy meter)**: Đo công suất, dòng điện, điện áp, điện năng tiêu thụ để giám sát quá trình sạc xe và tính phí cho mỗi lần sạc.
* **Vi điều khiển hoặc bộ điều khiển trung tâm**: Xử lý tín hiệu từ cảm biến, điều khiển contactor, giao tiếp với mạng IoT.
* **Màn hình hoặc đèn báo**: hiển thị trạng thái hoạt động, sạc, lỗi,...
* **Giao tiếp IoT (Wi-Fi, Bluetooth, LTE, Zigbee, LoRaWAN** , **v.v.)**: truyền dữ liệu đến điện thoại thông minh hoặc hệ thống server giám sát trung tâm.

### Hoạt động của trụ sạc xe máy điện

Quá trình vận hành của trụ sạc xe máy điện diễn ra theo các bước như sau:

Người dùng tiến hành kết nối đầu sạc của bộ sạc xe máy điện với cổng sạc ở trên trụ sạc. Kiểm tra xem tình trạng kết nối vật lý (có chắc chắn hay không) bộ sạc có bị lỏng dây hay không… Khi các điều kiện trên được đảm bảo, hệ thống sẽ kích hoạt contactor hoặc rơ-le để cấp nguồn AC từ lưới điện đến xe máy, cho phép bộ sạc bên trong xe thực hiện quá trình sạc pin. Hệ thống kích hoạt thông qua thao tác của người dùng (thao tác trên App của trạm sạc hoặc là quét mã qr chuyển khoản). Trong suốt thời gian sạc, hệ thống điều khiển sẽ thực hiện các chức năng như

* Theo dõi liên tục giá trị dòng điện và điện áp được cấp cho xe.
* Tính toán và ghi nhận lượng điện năng tiêu thụ khi sạc pin (kWh).
* Truyền dữ liệu thời gian thực đến hệ thống giám sát trung tâm hoặc ứng dụng điện thoại thông minh thông qua kết nối Internet IoT (Wi-Fi, LTE, Bluetooth, LTE, Zigbee, LoRaWAN,...).

Khi pin xe đã đầy hoặc người dùng chủ động dừng quá trình sạc qua ứng dụng hoặc các nút dừng trong hệ thống sạc, hệ thống sẽ ngắt nguồn điện cấp cho xe, ghi lại các thông tin liên quan như thời gian sạc, năng lượng tiêu thụ, trạng thái hoạt động. Hiển thị hoặc gửi thông báo đến người dùng nếu cần thiết.

### Ưu, nhược điểm của trụ sạc xe máy điện:

#### Ưu điểm:

* Chi phí thấp: Trụ sạc AC không cần tích hợp bộ chuyển đổi nguồn công suất lớn (như trụ DC), do đó giá thành sản xuất và bảo trì rẻ hơn nhiều.
* Thiết kế đơn giản, dễ triển khai: Sơ đồ mạch và cấu trúc vật lý ít phức tạp, thuận tiện cho việc lắp đặt ở nhà dân, bãi xe, khu dân cư, công viên hoặc điểm sạc công cộng.
* Tính tương thích: Phần lớn xe máy điện hiện nay đều sử dụng bộ sạc tích hợp trong xe, nên trụ sạc AC dễ dàng đáp ứng với yêu cầu sạc.
* Tính mở rộng: Dễ dàng nâng cấp phần mềm điều khiển, tích hợp thêm chức năng thanh toán, đặt lịch sạc, cảnh báo từ xa,...
* Tích hợp hệ thống điều khiển từ xa (IoT): Có thể kết nối qua Wi-Fi, Bluetooth, LTE, Zigbee, LoRaWAN để giám sát, điều khiển từ điện thoại thông minh hoặc hệ thống SCADA.

#### Nhược điểm:

* Phụ thuộc vào bộ sạc tích hợp trong xe: Nếu bộ sạc trong xe hỏng hoặc không tương thích, trụ sạc xe máy điện không thể sử dụng hiệu quả.
* Không phù hợp với sạc nhanh: Trụ sạc xe điện sử dụng điện áp AC n không cung cấp dòng điện một chiều trực tiếp cho pin nên không thể hỗ trợ công nghệ sạc nhanh như trụ DC.

## Chuẩn giao tiếp SPI, I2C, UART, RS485

### Chuẩn giao tiếp SPI

#### Giới thiệu chung:

Giao thức giao tiếp SPI được hãng Motorola phát hành vào giữa những năm 1980 nhằm đáp ứng khả năng truyền nhận, giao tiếp nhanh giữa các thiết bị với tốc độ từ vài MHz đến hàng chục MHz.

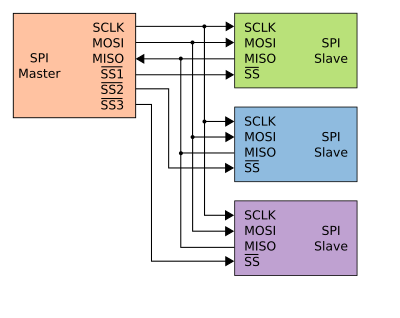
SPI (Serial Peripheral Interface), hay còn được gọi là Four wire, là một chuẩn giao tiếp 4 dây. Nó sử dụng kênh truyền nối tiếp song công (full duplex) với hai đường truyền dữ liệu riêng biệt, cho phép truyền dữ liệu và nhận dữ liệu đồng thời.

SPI hoạt động dựa trên mô hình Master-Slave, trong đó Master là thiết bị điều khiển chính và Slave thường là các ngoại vi.

Hệ thống SPI thường có một thiết bị Master, nhưng có khả năng có nhiều thiết bị Slave cùng được kết nối trên cùng 1 bus. [7].

Thiết bị Master sẽ kết nối với Slave qua 4 chân tín hiệu:

* MOSI (Master Output/Slave Input) – Đường gửi dữ liệu từ Master đến Slave.
* MISO (Master Input/Slave Output) – Đường gửi dữ liệu từ Slave cho Master.
* SCLK (Serial Clock) – xung nhịp Clock đồng bộ cho các thiết bị.
* SS/CS (Slave Select/Chip Select) - Được sử dụng để Master cho phép Slave để thiết lập giao tiếp. Trong một số trường hợp hệ thống bao gồm một Master giao tiếp với nhiều Slave, Master điều khiển mức logic trên chân SS/CS của mỗi Slave xuống mức thấp (mức 0) để biết Master đang giao tiếp với Slave tương ứng nào.



***Hình 2. 4*** *Mô hình giao tiếp SPI giữa Master và các Slave*

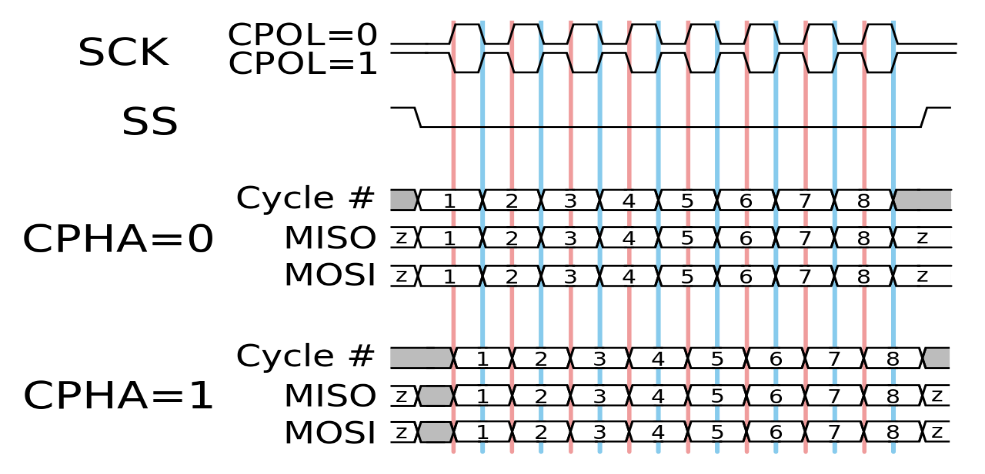
#### Hoạt động:

Cả Master và Slave đều được trang bị một thanh ghi dịch 8 bit (Shift Register) và một bộ tạo dao động (Clock Generator).

Khi Master bắt đầu truyền dữ liệu, nó sẽ ghi 8 bit dữ liệu vào thanh ghi dịch của mình và sau đó truyền dữ liệu qua đường MOSI (Master Output Slave Input) đến Slave. Tương tự, khi Slave truyền dữ liệu, các bit trên thanh ghi dịch của Slave được truyền tới Master qua đường MISO (Master Input Slave Output).

Nhờ việc trao đổi dữ liệu giữa hai thanh ghi, các thông tin được truyền tải giữa Master và Slave. Quá trình ghi và đọc dữ liệu vào Slave diễn ra đồng thời, tạo nên tốc độ truyền dữ liệu nhanh chóng. Vì vậy, giao thức SPI được coi là một giao thức rất hiệu quả. Giao thức SPI còn có các cấu hình quy định thời điểm nhận dữ liệu giữa Master và Slave dựa trên hai bit là CPOL (Clock Polarity - Độ lệch xung nhịp) và CPHA (Clock Phase - Pha xung nhịp):

* CPOL = 0, khi xung SCK ở mức thấp thì không truyền dữ liệu, CPOL = 1 thì xung SCK ở mức cao khi nghỉ.
* CPHA = 0, lấy dữ liệu khi xung SCK thay đổi trạng thái ở cạnh đầu tiên và ở cạnh thứ 2 trong khi CPHA = 1.



***Hình 2. 5*** *Giản đồ xung clock SPI trường hợp CPHA = 0 và CPHA =1*

#### Ưu nhược điểm:

* Ưu điểm:
  + Dữ liệu truyền nhận liên tục và không gián đoạn trong lúc truyền, vì không cần tốn thời gian giải mã bit Start và bit Stop như giao thức UART.
  + Hệ thống chỉ định lựa chọn Slave giao tiếp đơn giản, không phức tạp.
  + Hai đường dây MOSI và MISO riêng biệt nên cùng 1 thời điểm đồng thời nhận và truyền dữ liệu.
  + Tốc độ truyền dữ liệu nhanh.
* Nhược điểm:
  + Sử dụng nhiều dây để giao tiếp truyền nhận.
  + Không thể nhận biết được việc truyền dữ liệu thành công hay thất bại.
  + Khoảng cách truyền nhận rất ngắn.
  + Không có cơ chế truyền đồng thời dữ liệu cho nhiều Slave

### Chuẩn giao tiếp I2C

#### Giới thiệu chung:

I2C (Inter-Integrated Circuit) là một chuẩn truyền thông nối tiếp (serial communication protocol)  giao tiếp này được sử dụng rộng rãi trong các ứng dụng điện tử để kết nối các hệ thống nhúng và các thiết bị điện tử khác nhau. Giao thức này được giới thiệu vào năm 1982 bởi công ty Philips Semiconductors (nay là NXP Semiconductors), ngày nay I2C đã trở thành một giao thức phổ biến vì tính đơn giản, hiệu quả và linh hoạt của nó.

Giao thức I2C cần hai dây là SDA (Serial Data Line) và SCL (Serial Clock Line) để giao tiếp giữa các thiết bị với nhau. Dây SDA được sử dụng để truyền dữ liệu giữa các thiết bị, trong khi dây SCL được sử dụng để cung cấp xung nhịp Clock để đồng bộ hóa tín hiệu trên dây SDA.

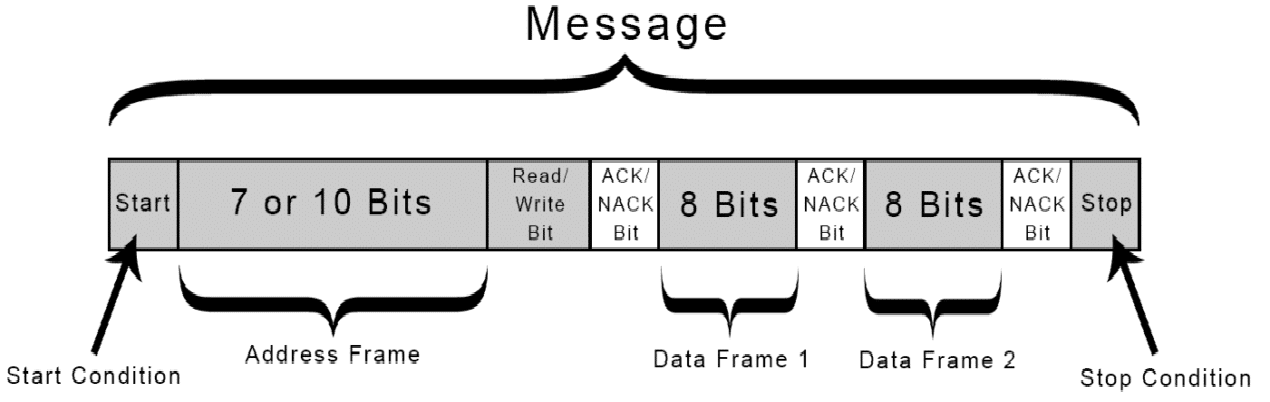
Ưu điểm lớn nhất của I2C đó là kết nối nhiều thiết bị trên cùng một đường dây truyền thông. Mỗi thiết bị được gán với một địa chỉ nhất định cho phép chúng giao tiếp với nhau thông qua đường dây tín hiệu chung.

I2C cung cấp nhiều tốc độ truyền dữ liệu khác nhau, như là tốc độ tiêu chuẩn 100 kbps (kilobit trên giây) hoặc 400 kbps. Ngoài ra, chuẩn I2C còn hỗ trợ các tốc độ nhanh hơn như 1 Mbps và 3.4 Mbps, được gọi là tốc độ nhanh (fast mode) và tốc độ siêu nhanh (high-speed mode).

Với tính đơn giản và hiệu suất tốt, chuẩn giao tiếp I2C đã trở thành một phương pháp phổ biến trong việc kết nối và giao tiếp giữa các hệ thống nhúng và thiết bị điện tử trong các ứng dụng như vi điều khiển, cảm biến, IC thời gian thực, EEPROM (Electrically Erasable Programmable Read-Only Memory) và nhiều thiết bị khác [8].

#### Hoạt động:

* Giao thức I2C sử dụng hai dây truyền tín hiệu:
  + SCL - Serial Clock Line: Đây là đường truyền tín hiệu xung đồng hồ (Clock), được thiết bị chủ (Master) tạo ra để đồng bộ quá trình truyền dữ liệu với thiết bị tớ Slave.
  + SDA - Serial Data Line: Đây là đường truyền dữ liệu, được sử dụng để truyền và nhận dữ liệu.
* Giao tiếp I2C diễn ra giữa thiết bị chủ (Master) và các thiết bị tớ (Slave).
  + Thiết bị chủ thường là một vi điều khiển hoặc máy tính nhúng chịu trách nhiệm tạo xung đồng hồ trên đường truyền SCL và điều khiển quá trình truyền/nhận dữ liệu thông qua đường SDA đến các thiết bị khác.
  + Các thiết bị tớ thường là những thiết bị nhận dữ liệu và phản hồi lại thông tin theo lệnh nhận được từ thiết bị chủ. Đây thường là các vi mạch tích hợp (IC) hoặc cũng có thể là vi điều khiển và máy tính nhúng khác.
* Khung truyền dữ liệu I2C:



**Hình 2. 6** *Khung truyền của giao thức I2C*

* + Address Frame:

Trong giao thức truyền nhận dữ liệu I2C thường sẽ có nhiều thiết bị trên cùng 1 bus I2C các thiết bị này được phân biệt với nhau thông qua các địa chỉ riêng biệt. Mỗi thiết bị sẽ được gán với 1 địa chỉ vật lý cố định với độ dài 7 bit hoặc là 10 bit.

* + Read/Write Bit:

Bit này đóng vai trò xác định xem đó là lệnh truyền tín hiệu để nhận dữ liệu từ thiết bị Slave hay là đọc dữ liệu từ Slave từ thiết bị Master. Nếu Master muốn truyền dữ liệu đi, bit này sẽ có giá trị là '0'. Ngược lại, nếu Master đang nhận dữ liệu lại từ Slave, bit này sẽ có giá trị là '1'.

* + Bit ACK/NACK:

ACK là viết tắt của Acknowledged / Not Acknowledged. Nếu một khung địa chỉ hoặc khung dữ liệu được nhận thành công. Nếu thành công, thiết bị Slave sẽ trả lời bằng bit '0' tức ACK; ngược lại, nếu không thành công, thiết bị Slave sẽ trả lời mặc định bằng bit '1' NACK.

* + Data Frame:

Dữ liệu truyền gồm 8 bit và được thiết lập bởi thiết bị gửi để truyền đến thiết bị nhận. Ngay sau khi 8 bit này được gửi đi, một bit ACK/NACK được gửi ngay sau đó để xác nhận liệu thiết bị Slave nhận đã nhận dữ liệu thành công hay chưa. Nếu việc nhận thành công, bit ACK/NACK sẽ được đặt là '0' ACK, và ngược lại, nếu không thành công, bit này sẽ có giá trị mặc định là bit '1' NACK.

#### Ưu nhược điểm:

* Ưu điểm:
  + Chỉ cần dùng 2 dây để giao tiếp.
  + Hỗ trợ đồng thời nhiều thiết bị giao tiếp trên 1 bus.
  + Cải thiện cơ chế phát hiện và sửa lỗi, giao thức I2C dựa vào tính năng ACK/NACK, đây là tính năng sửa lỗi mạnh mẽ so với các giao thức khác.
* Nhược điểm:
  + Tốc độ khá chậm so với giao thức UART hoặc SPI.
  + Kích thước khung truyền bị giới hạn 8 bit.
  + Dễ bị nhiều trong môi trường công nghiệp, nhạy cảm với nhiễu điện từ đặc biệt ở khoảng cách truyền dài và ở môi trường có nhiều thiết bị công suất cao.
  + Không truyền nhận đồng thời cùng một thời điểm như giao thức UART và SPI.

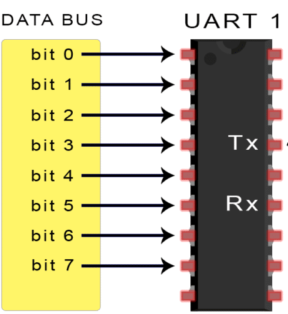
### Chuẩn giao tiếp UART

#### Giới thiệu chung:

UART (Universal Asynchronous Receiver - Transmitter) là một giao thức truyền thông phần cứng dùng giao tiếp nối tiếp không đồng bộ và có thể cấu hình được tốc độ truyền. Mục đích chính của UART là truyền và nhận dữ liệu nối tiếp.

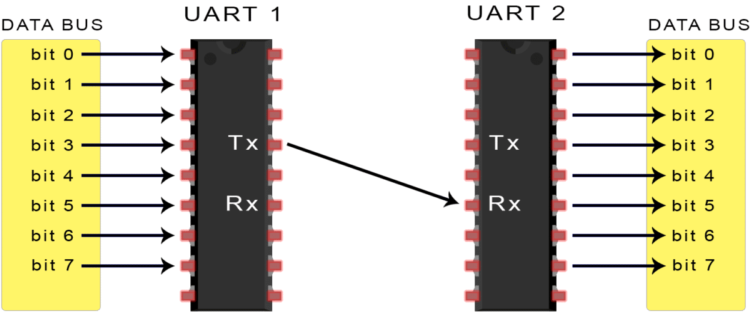
Giao thức UART đơn giản và được áp dụng rộng rãi trong các giao thức truyền tín hiệu, bao gồm hai đường truyền dữ liệu độc lập là: TX (truyền) và RX (nhận). Dữ liệu được truyền và nhận dưới dạng các khung dữ liệu có cấu trúc chuẩn.

#### Hoạt động:



**Hình 2. 7** *Truyền dữ liệu từ DATA BUS đến UART1*

Dữ liệu sẽ được chuyển từ Bus dữ liệu đến UART1 truyền ở dạng song song. Sau khi nhận được dữ liệu, lúc này nó sẽ thêm một bit start, một bit chẵn lẻ và một bit stop để tạo thành gói dữ liệu (packet data).



**Hình 2. 8** *Quá trình truyền dữ liệu UART*

Gói dữ liệu được truyền ra từng bit nối tiếp tại chân Tx của UART 1 và UART 2 sẽ nhận gói dữ liệu từng bit tại chân Rx của nó. Sau khi nhận xong dữ liệu, UART 2 chuyển đổi dữ liệu trở lại dạng song song và loại bỏ bit start, bit chẵn lẻ và bit stop.

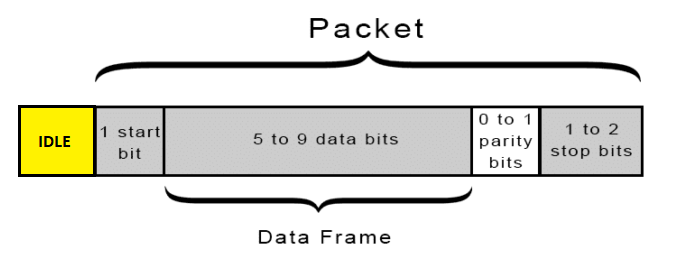
UART (Universal Asynchronous Receiver-Transmitter) là giao thức truyền dữ liệu nối tiếp theo một trong ba chế độ hoạt động như sau:

* Chế độ Simplex: Chỉ tiến hành giao tiếp một chiều.
* Chế độ Half duplex: Dữ liệu sẽ đi theo một hướng tại một thời điểm.
* Chế độ Full duplex: Thực hiện giao tiếp đồng thời cùng thời điểm đến và đi từ mỗi Master và Slave.

Giao thức UART (Universal Asynchronous Receiver-Transmitter) cung cấp kết nối truyền thông giữa các chip thông qua việc kết nối chân TX (truyền) của một chip với chân RX (nhận) của một chip khác và ngược lại. Trong quá trình truyền dữ liệu, thông thường sử dụng mức điện áp 3.3V hoặc 5V.

* Khung truyền dữ liệu UART

Dữ liệu được truyền trong giao tiếp UART được tổ chức thành các gói (Packets). Mỗi Packets chứa 1 bit Start, 5 đến 9 bit dữ liệu (tùy thuộc vào UART), 1 bit Parity và 1 hoặc 2 bit Stop.



**Hình 2. 9** *Khung truyền của giao tiếp UART*

* IDLE (Trạng thái bus rảnh)
* Là không có sự truyền/nhận dữ liệu trên bus.
* Giá trị của IDLE sẽ luôn luôn ở mức cao để phát hiện điều kiện bắt đầu của Start bit
* Start bit (Bit khởi đầu):
* Đường truyền dữ liệu trong giao tiếp UART thường được giữ ở mức điện áp cao(IDLE) khi nó không truyền dữ liệu.
* Để bắt đầu truyền dữ liệu, UART truyền sẽ kéo đường truyền từ mức cao xuống mức thấp trong một chu kỳ đồng hồ.
* Khi UART 2 phát hiện sự chuyển đổi điện áp cao xuống thấp, nó bắt đầu đọc các bit trong khung dữ liệu ở tần số của tốc độ truyền (Baud rate).
* Data bits (Bit dữ liệu) [0-8]:
* Khung dữ liệu chứa dữ liệu thực tế đang được truyền. Nó có thể dài từ 5 bit đến 8 bit nếu sử dụng bit Parity (bit chẵn lẻ).
* Nếu không sử dụng bit Parity, khung dữ liệu có thể dài 9 bit. Trong hầu hết các trường hợp, dữ liệu được gửi với bit LSB (bit có trọng số thấp nhất) trước tiên.
* Nếu muốn gửi dữ liệu thì phải gửi 1 chuỗi bit. Tại 1 thời điểm chỉ gửi được 1 bit data. Nếu gửi 1 byte thì gửi 8 lần.
* Parity bit (Bit chẵn lẻ)
* Dùng để kiểm tra dữ liệu truyền có đúng không (một cách tương đối). Có 2 loại parity là parity chẵn (even parity) và parity lẻ (odd parity).
* Nếu bit Parity là 0 (chẵn), thì tổng các bit 1 trong khung dữ liệu phải là một số chẵn.
* Nếu bit Parity là 1 (lẻ), thì tổng các bit 1 trong khung dữ liệu sẽ là một số lẻ.
* Nếu bit Parity là 1 (lẻ), thì tổng các bit 1 trong khung dữ liệu sẽ là một số lẻ.

#### Ưu nhược điểm:

* Ưu điểm:
  + Chỉ dùng 2 dây và nối chung GND để truyền dữ liệu.
  + Cấu trúc khung đơn giản và bản chất không đồng bộ của giao thức giúp đơn giản hóa việc tích hợp vào nhiều ứng dụng khác nhau.
  + UART cho phép thiết lập tốc độ baud rate tùy ý (miễn là hai bên cùng cài đặt giống nhau), linh hoạt cho nhiều nhu cầu khác nhau khi truyền nhận dữ liệu.
* Nhược điểm:
  + Kích thước của khung dữ liệu tối đa là 9 bit. Không hiệu quả khi truyền dữ liệu lớn.
  + Không hỗ trợ truyền dữ liệu đến nhiều thiết bị trên cùng một bus như I2C hoặc RS-485. Nếu muốn giao tiếp với nhiều thiết bị, phải sử dụng nhiều cổng UART hoặc chuyển sang giao thức khác.
  + Không có chuẩn vật lý cụ thể nên khi giao tiếp với các thiết bị khác nhau, cần mạch chuyển mức logic phù hợp.

### Chuẩn giao tiếp RS485

#### Giới thiệu chung:

RS-485 (Viết tắt của Recommended Standard 485) là một chuẩn giao tiếp truyền thông nối tiếp được sử dụng nhiều trong truyền nhận dữ liệu giữa các thiết bị điện tử. RS-485 được phát triển vào năm 1983 nhằm cải tiến chuẩn RS422. RS-485 được thiết kế hỗ trợ cả kết nối điểm-điểm và đa điểm (tối đa 32 thiết bị trên cùng một đường truyền tín hiệu).

Chuẩn giao tiếp truyền thông này đáp ứng nhu cầu truyền dữ liệu ở khoảng cách xa với độ ổn định và tin cậy cao, đặc biệt là ở trong môi trường nhiễu. Khả năng kết nối nhiều thiết bị một đường truyền làm tăng tính linh hoạt và giảm chi phí và tăng khả năng mở rộng kết nối hệ thống.

#### Hoạt động:

RS-485 sử dụng tín hiệu vi sai để truyền dữ liệu, có nghĩa là nó dựa vào sự khác biệt về điện áp giữa hai dây (A và B) để xác định trạng thái của bit dữ liệu. Tín hiệu truyền đi là hiệu số điện áp giữa hai dây dẫn vi sai A và B, mức logic trên mỗi chân khi so với đất phải ngược với chân còn lại.

Trạng thái logic cao được xác định khi điện áp của dây A lớn hơn dây B tối thiểu là 200mV, và -200mV xác định trạng thái logic thấp. Nếu điện áp VAB nằm trong khoảng -200mV đến 200mV thì tín hiệu được coi là không xác định. Điện áp của mỗi dây tín hiệu so với đất nằm trong khoảng từ -7V đến +12V.

Tín hiệu vi sai của RS-485 được truyền qua cặp dây xoắn, là hai dây có chiều dài bằng nhau và được xoắn lại với nhau. Việc sử dụng cặp dây xoắn giúp giảm thiểu nhiễu do tác động từ bên ngoài lên cả hai dây tín hiệu, vì chúng có cùng môi trường ảnh hưởng. Bằng cách lấy hiệu điện áp trên hai dây A và B, nhiễu chung có thể loại bỏ, đặc biệt là khi truyền ở khoảng cách xa và với tốc độ cao. Điều này cải thiện khả năng chống nhiễu, tăng tốc độ truyền và cho phép truyền dữ liệu ở khoảng cách xa hơn so với các chuẩn truyền thông sử dụng tín hiệu đơn. Để ngăn chặn tín hiệu phản xạ và giao thoa, cần có điện trở đầu cuối (Terminating Resistor) trên mỗi hai đầu cặp dây vi sai của chuẩn truyền thông RS-485. Giá trị lí tưởng của điện trở đầu cuối là trở kháng đặc tính của đường dây vi sai, thường là từ 100 đến 120Ω.

#### Ưu nhược điểm:

* Ưu điểm
* Khả năng chống nhiễu: RS485 được thiết kế với cấu trúc cáp đôi xoắn và cơ chế truyền tín hiệu vi sai, việc này giúp tăng cường khả năng chống nhiễu đáng kể. Ở trong công nghiệp có nhiều nhiễu điện từ, RS485 vẫn duy trì được hiệu quả truyền dữ liệu cao, làm giảm nguy cơ mất mát hoặc sai lệch thông tin.
* Khoảng cách truyền: RS485 hỗ trợ truyền dữ liệu qua khoảng cách lên đến 1.2 km, một khoảng cách đáng kể so với nhiều chuẩn giao tiếp khác. Điều này làm tăng khả năng linh hoạt trong việc thiết kế hệ thống và mở rộng phạm vi ứng dụng, đặc biệt trong các cơ sở công nghiệp rộng lớn và hệ thống quản lý tòa nhà.
* Tốc độ truyền dữ liệu cao: cổng RS485 cung cấp tốc độ truyền tối đa lên đến 10 Mbps, tuỳ vào khoảng cách và cấu hình của hệ thống. Tốc độ này đủ cao để xử lý hầu hết các yêu cầu truyền dữ liệu trong môi trường công nghiệp và tự động hóa.
* Nhược điểm
* Giới hạn về số lượng thiết bị: Mặc dù cáp truyền thông RS485 hỗ trợ giao tiếp đa điểm, có thể kết nối nhiều thiết bị trên cùng một mạng, nhưng nó vẫn có giới hạn về số lượng thiết bị có thể kết nối. Điều này đôi khi làm hạn chế khả năng mở rộng của hệ thống trong các ứng dụng lớn.
* Độ phức tạp trong cài đặt: Việc thiết lập và cấu hình một hệ thống cáp truyền thông RS485 có thể phức tạp và yêu cầu kiến thức kỹ thuật chi tiết. So với các chuẩn giao tiếp đơn giản hơn, RS485 đòi hỏi sự hiểu biết về cấu trúc mạng, thiết lập địa chỉ thiết bị và cấu hình các thông số truyền dữ liệu, điều này có thể là thách thức cho những người mới làm quen với hệ thống.

## Vi điều khiển ESP32-WROOM-32E-N4

### Tổng quan

ESP32-WROOM-32E-N4 là một module Wi-Fi & Bluetooth tích hợp, dựa trên vi điều khiển SoC ESP32 do Espressif Systems phát triển. Đây là phiên bản nâng cấp của dòng ESP8266, với nhiều cải tiến vượt trội về hiệu năng xử lý, khả năng kết nối và tính năng bảo mật.

Module sử dụng vi xử lý Xtensa LX6 32-bit của Tensilica, hỗ trợ hai cấu hình: lõi đơn hoặc lõi kép (dual-core), cho phép thực hiện song song nhiều tác vụ, rất phù hợp với các ứng dụng IoT, nhúng thời gian thực, và thiết bị thông minh.

Một số điểm nổi bật của ESP32-WROOM-32E-N4:

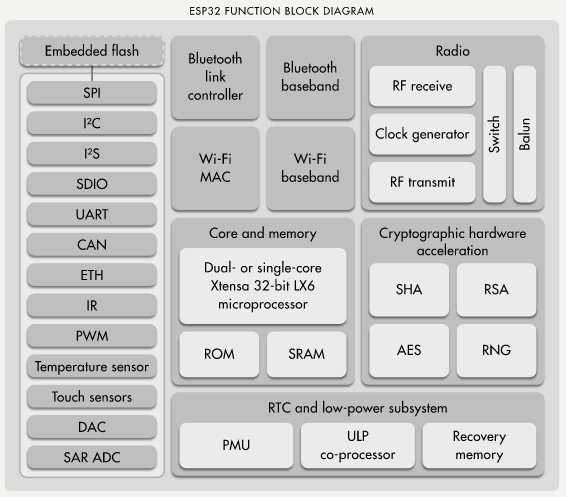
* Tích hợp Wi-Fi 802.11 b/g/n (hỗ trợ cả chế độ Station, SoftAP và Station + AP), cho phép giao tiếp mạng không dây linh hoạt.
* Tích hợp Bluetooth v4.2 BR/EDR và BLE, giúp kết nối với các thiết bị di động hoặc cảm biến BLE.
* Bộ nhớ flash 4MB tích hợp sẵn, phù hợp để lưu trữ chương trình và dữ liệu cho các ứng dụng vừa và nhỏ.
* Hỗ trợ nhiều giao thức ngoại vi: SPI, I2C, UART, PWM, ADC, DAC, CAN, SDIO… giúp dễ dàng kết nối với các cảm biến, thiết bị ngoại vi và mô-đun mở rộng.
* Tích hợp bộ bảo mật phần cứng như AES, SHA-2, RSA, ECC, Random Number Generator, hỗ trợ mã hóa dữ liệu và xác thực an toàn.
* Tiêu thụ điện năng thấp, hỗ trợ nhiều chế độ ngủ (sleep modes), rất phù hợp cho các ứng dụng dùng pin hoặc yêu cầu tiết kiệm năng lượng.

Với hiệu năng mạnh mẽ, khả năng kết nối đa dạng và kích thước nhỏ gọn, ESP32-WROOM-32E-N4 được sử dụng rộng rãi trong các ứng dụng như: nhà thông minh, điều khiển thiết bị qua Wi-Fi, giám sát môi trường, thiết bị đeo thông minh, Robotics, và các hệ thống IoT trong công nghiệp.

### Thông số kỹ thuật

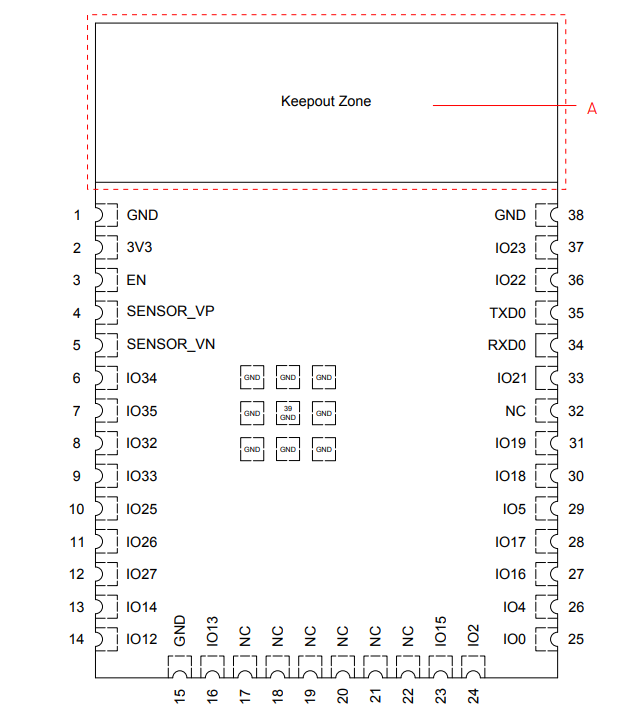
* Model: Wifi BLE SoC ESP32 ESP-WROOM-32E chính hãng Espressif (tương thích hoàn toàn với phiên bản cũ ESP-WROOM-32 hiện đã ngưng sản xuất).
* ESP32-D0WD-V3 embedded, Xtensa® dual-core 32-bit LX6 microprocessor, up to 240 MHz.
* Điện áp sử dụng: từ 3~3.6VDC.
* Dòng điện sử dụng: ~90mA.
* 448 KB ROM for booting and core functions.
* 520 KB SRAM for data and instructions.
* 16 KB SRAM in RTC.
* 4MB (XXN4) / 16MB (XX0H28) SPI Flash.
* Kiểu Anten: PCB.
* Wifi
* 802.11b/g/n.
* Bit rate: 802.11n up to 150 Mbps.
* A-MPDU and A-MSDU aggregation.
* 0.4 µs guard interval support.
* Center frequency range of operating channel: 2412 ~ 2484 MHz .
* Bluetooth
* Bluetooth V4.2 BR/EDR and Bluetooth LE specification.
* Class-1, class-2 and class-3 transmitter.
* AFH.
* CVSD and SBC.
* Hardware
* Interfaces: SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®, compatible with ISO11898-1).
* Bộ dao động tạo xung 40 MHz.
* Kích thước: 18 x 25.5 x 3.1mm.

ESP32-WROOM là một module vi điều khiển mạnh mẽ cùng với khả năng tích hợp nhiều chức năng của phần cứng ngay trên cùng một board mạch, giúp giảm thiểu số linh kiện ngoại vi và đơn giản hóa quá trình thiết kế hệ thống nhúng. Các chức năng được mô tả ở hình 2. 10



**Hình 2. 10** *Sơ đồ khối chức năng của ESP32-WROOM-32E-N4*

ESP32-WROOM-32E-N4 là một module dạng SMD có tổng cộng 38 chân (pins), trong đó bao gồm các chân cấp nguồn, chân I/O tích hợp đa chức năng và có các chân chuyên dụng cho các chuẩn giao tiếp không dây. Các chân này được người sử dụng cấu hình cho nhiều mục đích sử dụng khác nhau tùy theo yêu cầu của ứng dụng và dự án. Hình 2.11 thể hiện sơ đồ chân của vi điều khiển ESP32-WROOM-32E-N4.

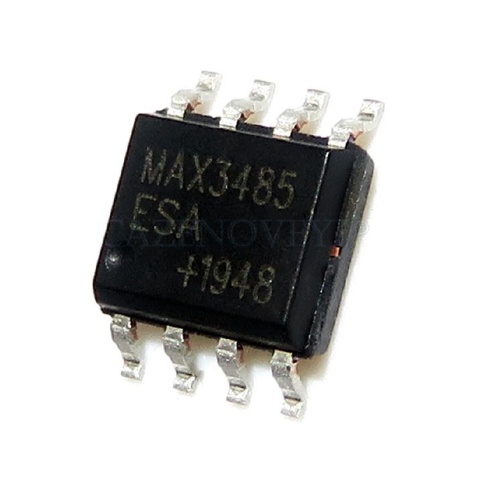


**Hình 2. 11** *Sơ đồ chân ESP32-WROOM-32E-N4*

## IC Truyền nhận RS-485 MAX3485

### Tổng quan

MAX3485 được thiết kế như là các bộ thu phát truyền nhận công suất thấp dành cho chuẩn giao tiếp RS485. Được sản xuất bởi hãng Maxim Integrated (nay thuộc Analog Devices). Mỗi IC MAX3485 chứa một bộ điều khiển và một bộ thu tín hiệu. Hỗ trợ chuẩn giao tiếp RS-485 và RS-422 giao thức truyền thông vi sai giúp tăng khả năng chống nhiễu và truyền xa. MAX3485 cho phép truyền với tốc độ lên tới 10Mbps và có chế độ tiết kiệm năng lượng. Nó có chức năng bảo vệ ngắn mạch và chống tiêu hao năng lượng quá mức, đầu ra của trình điều khiển được đặt vào trạng thái trở kháng cao. MAX3485 có tính năng bảo vệ an toàn đảm bảo đầu ra trạng thái cao nếu cả hai đầu vào đều được để mở mạch. MAX3485 được thiết kế để giao tiếp RS-485 theo kiểu bán song công (quá trình truyền hoặc nhận dữ liệu từ master và slave tại 1 thời điểm).



**Hình 2. 12** *IC MAX3485*

### Thông số kỹ thuật

Bảng 2. Thông số kỹ thuật IC MAX3485

|  |  |
| --- | --- |
| **Thuộc tính** | **Giá trị** |
| Loại | Transceiver |
| Giao thức | RS422, RS485 |
| Số lượng Drivers/Receiver | 1/1 |
| Kiểu truyền | Half Duplex |
| Độ trễ Receiver | 20 mV |
| Tốc độ dữ liệu | 10Mbps |
| Điện áp cấp | 3.135V ~ 3.465V |
| Nhiệt độ hoạt động | -40°C ~ 85°C |
| Kiểu chân | Dán bề mặt |
| Số chân | 8 |

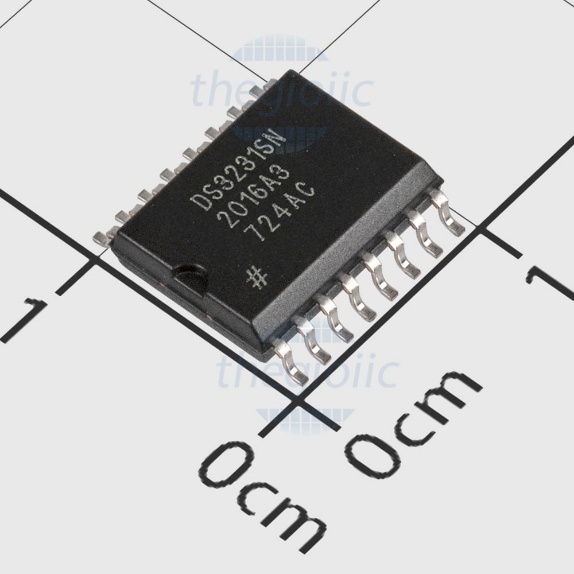
## IC Thời gian thực DS3231

### Tổng quan

DS3231 là một IC thời gian thực (RTC – Real Time Clock) với độ chính xác cao do Maxim Integrated sản xuất. Nó được thiết kế ra để cung cấp thông tin về thời gian và ngày tháng năm một cách liên tục và chính xác. Kể cả khi mất nguồn bởi nó có tích hợp nguồn pin dự phòng khi mất điện (pin coin 3V – CR2032).

Tích hợp bộ dao động bên trong giúp giảm nhiễu và sai số về thời gian. Khác với các IC thời gian thực sử dụng thạch anh ngoài, DS3231 sử dụng thạch anh nội và được bù nhiệt đảm bảo độ chính xác cao hơn.

Sử dụng giao tiếp I2C dễ dàng giao tiếp và tương thích với rất nhiều loại vi điều khiển, máy tính nhúng … Sai số cực thấp chỉ khoảng 2ppm (khoảng 1 phút/năm).



**Hình 2. 13** *IC DS3231*

### Thông số kỹ thuật

Điện áp hoạt động: từ 3.3~5VDC.

Giao tiếp: I2C

Lưu trữ và cung cấp các thông tin về thời gian thực: ngày, tháng, năm, giờ, phút, giây,...

Có pin backup duy trì thời gian thực trong trường hợp IC không được cấp nguồn.

Kích thước: 38 x 22 x 14mm

## Giao thức HTTP

### Tổng quan

HTTP (HyperText Transfer Protocol) là giao thức sử dụng để truyền nhận dữ liệu. Dữ liệu được truyền qua HTTP dưới dạng plain text. HTTP định nghĩa cách máy khách (client) và máy chủ (server) giao tiếp với nhau để truyền tải các tài nguyên như văn bản, hình ảnh, video, hoặc dữ liệu. Các tài nguyên không được mã hóa hoặc bảo mật, điều này làm cho dữ liệu dễ bị nghe lén và đánh cắp.

Giao thức này thường được sử dụng trong World Wide Web (WWW) để truyền tải dữ liệu giữa máy chủ và trình duyệt Web. Giao thức này sử dụng cổng 80 (port 80) làm cổng chính để giao tiếp. HTTP là một giao thức stateless, nghĩa là mỗi yêu cầu từ máy khách (client) đến máy chủ (server) đều được xử lý độc lập, không giữ lại trạng thái của các yêu cầu trước đó.

### Các mã lỗi của HTTP:

Khi người dùng truy cập vào một trang web thông qua các trình duyệt, thi thoảng sẽ gặp phải các thông báo lỗi HTTP được phản hồi từ máy chủ. Một số lỗi phổ biến như:

* Lỗi 404 (HTTP 404 – Not Found): Thông báo lỗi này sẽ xuất hiện khi địa chỉ (URL) mà người dùng yêu cầu không có tồn tại trên máy chủ. Nguyên nhân có thể do đường dẫn bị sai hoặc trang đã bị xoá hoặc di chuyển đến một nơi khác.
* Lỗi 401 (HTTP 401 - Unauthorized): Xảy ra khi người dùng cố gắng truy cập vào một tài nguyên yêu cầu xác thực nhưng chưa đăng nhập hoặc thông tin đăng nhập không hợp lệ
* Lỗi 500 (HTTP 500 - Internal Server Error): Đây là lỗi nội bộ của máy chủ, được xảy ra khi máy chủ gặp sự cố trong quá trình xử lý yêu cầu. Nguyên nhân có thể là do lỗi phần mềm, do cấu hình sai hoặc do sự cố tạm thời từ hệ thống backend.

### Các phương thức yêu cầu của HTTP:

Giao thức HTTP hỗ trợ nhiều phương thức khác nhau để máy khách (client) gửi yêu cầu tới máy chủ (sever). Trong đó hai phương thức được phổ biến và sử dụng nhiều nhất đó là GET và POST

#### GET:

Phương thức GET được sử dụng để yêu cầu truy xuất dữ liệu từ một nguồn được chỉ định (thường là một trang web hoặc API). Khi sử dụng phương thức GET:

* Dữ liệu sẽ được gửi kèm theo yêu cầu và sẽ được đính kèm trực tiếp trên URL dưới dạng là chuỗi truy vấn (query string).
* Dữ liệu này là công khai và có thể nhìn thấy bởi bất kỳ ai nếu như URL bị lộ.
* GET có giới hạn về độ dài URL, thường giới hạn trong khoảng 2000 ký tự, tuỳ thuộc vào trình duyệt sử và máy chủ sử dụng.
* Ưu điểm nổi bật của GET là cho phép đánh dấu trang (bookmark) hoặc chia sẻ liên kết dễ dàng, đồng thời có lợi cho SEO(Search Engine Optimization).

#### POST:

Phương thức POST được sử dụng để gửi dữ liệu đến máy chủ để xử lý (ví dụ: gửi form, thông tin hệ thống, tải tệp …) Khi sử dụng POST:

* Dữ liệu sẽ được gửi ở trong phần thân (body) của yêu cầu, không hiển thị ở trên URL
* Không giới hạn kích thước dữ liệu, phù hợp với các nhu cầu gửi dữ liệu lớn như upload file, hình ảnh, video,...
* Phương thức POST an toàn hơn GET về mặt bảo mật thông tin người dùng (mặc dù POST vẫn cần mã hóa HTTPS để đảm bảo an toàn một cách thực sự).

## Màn hình DWIN HMI

Màn hình DWIN HMI là dòng sản phẩm màn hình cảm ứng công nghiệp do hãng DWIN Technology phát triển, chuyên dùng để hiển thị và tương tác giữa con người với hệ thống điều khiển nhúng hoặc tự động hóa. Dòng màn hình này nổi bật nhờ khả năng hoạt động ổn định, giao diện thân thiện, hỗ trợ lập trình dễ dàng và tích hợp nhiều chuẩn giao tiếp phổ biến như UART, RS232, RS485 và CAN.

### Tổng quan

DWIN cung cấp nhiều loại màn hình với các kích thước đa dạng như 2.4 inch, 2.8 inch, 3.5 inch, 4.3 inch, 5.0 inch, 7.0 inch, 10.1 inch và lớn hơn, đáp ứng nhu cầu khác nhau trong các hệ thống từ đơn giản đến phức tạp. Mỗi dòng màn hình đều được tích hợp bộ xử lý đồ họa T5L, hỗ trợ hệ điều hành DGUS II hoặc TA để xây dựng giao diện người dùng.

Các tính năng chung của màn hình DWIN bao gồm:

* Kích thước đa dạng: Từ 2.4 inch đến hơn 10 inch, phù hợp với nhiều loại ứng dụng.
* Độ phân giải: Tùy từng model, phổ biến như 320x240, 480x272, 800x480, 1024x600 pixel,...
* Bộ xử lý đồ họa: Tích hợp vi điều khiển T5L do DWIN phát triển, tối ưu cho hiển thị giao diện và truyền thông.
* Giao tiếp: Hỗ trợ UART TTL (3.3V/5V), RS232, RS485, một số model còn hỗ trợ CAN hoặc USB.
* Phần mềm thiết kế: DWIN cung cấp phần mềm DGUS (DWIN Graphic User System), cho phép người dùng kéo-thả để thiết kế giao diện một cách trực quan, dễ dàng tải chương trình qua thẻ SD hoặc cổng UART.
* Hệ điều hành tích hợp: Có thể sử dụng hệ điều hành DGUS II (dễ dùng, không cần lập trình) hoặc hệ TA (mức thấp hơn, lập trình linh hoạt hơn).
* Cảm ứng: Hỗ trợ cả cảm ứng điện trở và điện dung tùy theo phiên bản màn hình.

### Thông số kỹ thuật

## Cảm biến

### Cảm biến SHT31

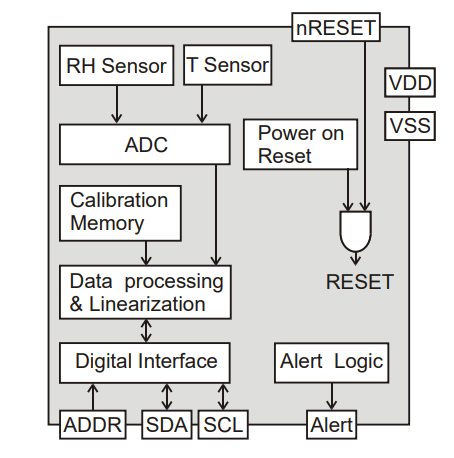
Cảm biến SHT31 là một loại cảm biến đo lường nhiệt độ độ ẩm kỹ thuật số chất lượng cao được sử dụng để đo các thông số độ ẩm và nhiệt độ không khí trong nhiều ứng dụng. Là sản phẩm xuất xứ từ Sensirion, một công ty chuyên về phát triển và sản xuất các cảm biến đo lường chất lượng không khí. SHT31 hỗ trợ chuẩn giao tiếp I2C cùng với kích thước nhỏ gọn và độ chính xác cao (sai lệch 0.3 °C, 2 %RH), cảm biến SHT31 được áp dụng đa dạng trong các ứng dụng về môi trường.



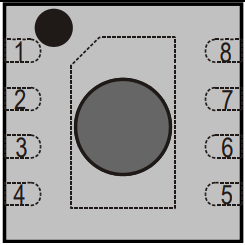
**Hình 2. 14** *Cảm biến SHT31*

Một số ưu điểm của cảm biến SHT31:

* Độ chính cao và ổn định lâu dài.
* Khả năng xử lý cao.
* Tỷ lệ tín hiệu so với nhiễu cao.
* Kích thước nhỏ.
* Hỗ trợ giao tiếp I2C lên tới 1MHz và 2 địa chỉ tùy chọn.



***Hình 2. 15*** *Sơ đồ khối cảm biến SHT31*



***Hình 2. 16*** *Hình ảnh mặt dưới của SHT31*

|  |  |  |
| --- | --- | --- |
| Pin 1 | SDA | Dữ liệu vào/ra nối tiếp |
| Pin 2 | ADDR | Địa chỉ I2C |
| Pin 3 | ALERT | Chân cảnh báo |
| Pin 4 | SCL | Xung đồng hồ nối tiếp |
| Pin 5 | VDD | Chân cấp nguồn |
| Pin 6 | nRESET | Chân reset mức thấp |
| Pin 7 | R |  |
| Pin 8 | VSS | Ground |

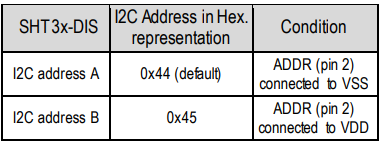
***Bảng 2. 2 Bảng mô tả các chân cảm biến SHT31***

#### Nguyên lý hoạt động

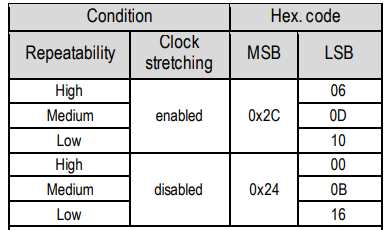
Cảm biến SHT31 sử dụng nguyên lý đo điện dung để đo độ ẩm tương đối. Thành phần cảm biến bao gồm một tụ điện có điện môi là một loại polyme. Điện môi hấp thụ hoặc giải phóng nước phục thuộc vào độ ẩm xung quanh tương ứng. Sự thay đổi này trong hằng số điện môi và khiến cho điện dung của tụ điện bị thay đổi, và được đo bằng một mạch điện tử. Để đo nhiệt độ, SHT31 sử dụng các bóng bán dẫn lưỡng cực [11].

Cảm biến hỗ trợ chuẩn giao tiếp I2C nên có thể dùng các vi điều khiển để giao tiếp thông qua các chân SCL và SDA. Cảm biến SHT31 có hỗ trợ đo ở các chế độ Single Shot và Period. Ngoài ra có thể cấu hình thêm chế độ Clock Stretching để đo ở chế độ đồng hồ kéo dài.

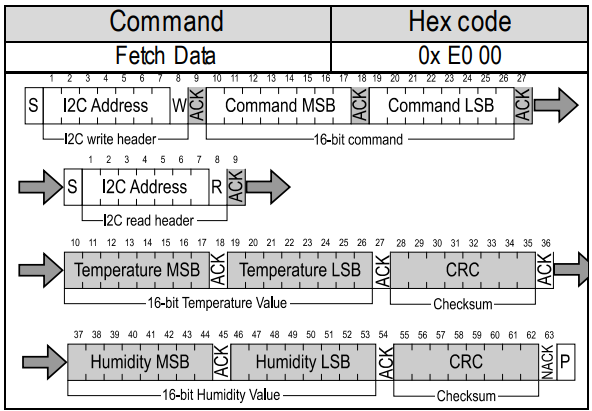
Dưới đây là địa chỉ cũng như một số thanh ghi quan trọng của cảm biến SHT31.



***Hình 2. 17*** *Địa chỉ cảm biến SHT31*



***Hình 2. 18*** *Thanh ghi lựa chọn chế độ lặp lại và đồng hồ kéo dài*



***Hình 2. 19*** *Giao thức truyền dữ liệu của SHT31*

#### Thông số kỹ thuật

* Điện áp hoạt động: 2.4 – 5.5V
* Thang đo độ ẩm: 0 – 100% RH.
* Thang đo nhiệt độ: -40 – 125 °C.
* Sai lệch đo: ±2 %RH và ±0.3 °C.
* Giao tiếp: I2C.

### Đồng hồ đo điện năng DDSU666

DDSU666 là công tơ điện tử đo năng lượng một pha, được thiết kế để đáp ứng nhu cầu giám sát điện năng và đo lường thông số điện trong các hệ thống điện công nghiệp và dân dụng. Thiết bị tích hợp cả chức năng đo lường và giao tiếp truyền thông, đóng vai trò là một bộ công tơ thông minh thế hệ mới.

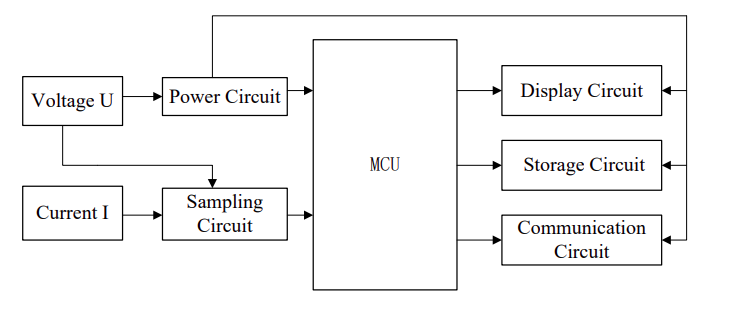
Đặc điểm nổi bật của sản phẩm:

* Thiết bị có khả năng đo năng lượng điện chiều thuận (tích cực) và năng lượng điện chiều ngược (tiêu cực). Trong đó, phần năng lượng tiêu cực sẽ được cộng dồn vào năng lượng dương để tham khảo (không sử dụng làm cơ sở tính toán hóa đơn).
* Sử dụng màn hình LCD chất lượng cao có khả năng hoạt động ổn định trong dải nhiệt độ rộng, đảm bảo hiển thị rõ ràng trong cả môi trường khắc nghiệt như nhà máy, ngoài trời hoặc trong tủ điện công nghiệp.
* Thiết bị được thiết kế theo chuẩn gắn ray DIN35mm, nhỏ gọn, dễ lắp ráp trong các tủ điện tiêu chuẩn.Thiết kế modular giúp việc mở rộng, bảo trì và kết nối mạng (qua RS485) trở nên nhanh chóng và tiện lợi, phù hợp với hệ thống đo lường và giám sát điện hiện đại.



**Hình 2. 20** *Đồng hồ đo điện năng DDSU666*

Hình 2.21 mô tả sơ đồ khối tổng quát của thiết bị



**Hình 2. 21** *Sơ đồ khối tổng quát của DDSU666*

#### Nguyên lý hoạt động

#### Thông số kỹ thuật

## Các phần mềm sử dụng trong đề tài

### EAGLE

Eagle là một phần mềm thiết kế mạch điện tử (EDA - Electronic Design Automation) phổ biến, được sử dụng để vẽ sơ đồ nguyên lý (schematic) và thiết kế bo mạch in (PCB). Eagle cung cấp giao diện trực quan, thư viện linh kiện phong phú và hỗ trợ nhiều lớp mạch. Ngoài ra, phần mềm còn cho phép kiểm tra lỗi thiết kế (DRC/ERC) và xuất file Gerber để sản xuất PCB. Eagle được sử dụng rộng rãi trong cả học tập và công nghiệp nhờ tính linh hoạt và khả năng tùy chỉnh cao.



***Hình 2. 22*** *Phần mềm EAGLE*

### Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mạnh mẽ, miễn phí và đa nền tảng do Microsoft phát triển. VS Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, Python, JavaScript, v.v., với hệ sinh thái mở rộng phong phú thông qua các tiện ích mở rộng (extensions). Giao diện hiện đại, khả năng gợi ý thông minh (IntelliSense), debug tích hợp, và khả năng tùy biến cao giúp VS Code trở thành công cụ lý tưởng cho cả sinh viên và lập trình viên chuyên nghiệp.



Hình 2. 23 Visual Studio Code

### Arduino IDE

Arduino IDE là phần mềm phát triển chính thức dành cho các vi điều khiển Arduino, giúp người dùng dễ dàng viết mã, biên dịch và nạp chương trình vào bo mạch. Giao diện đơn giản, thân thiện với người mới bắt đầu, cùng với thư viện phong phú hỗ trợ nhiều loại cảm biến và module. Arduino IDE hỗ trợ ngôn ngữ lập trình C/C++ và có khả năng giám sát dữ liệu thời gian thực qua Serial Monitor, rất phù hợp cho các dự án IoT và điều khiển nhúng.



***Hình 2. 24*** *Arduino IDE*

### DGUS\_V7.647

DGUS\_V7.647 là phần mềm chuyên dụng do DWIN phát triển, dùng để thiết kế và cấu hình giao diện người dùng (HMI) cho các màn hình cảm ứng của hãng. Phần mềm cho phép người dùng kéo thả giao diện, thêm nút bấm, hiển thị dữ liệu, ảnh động, và cấu hình các vùng bộ nhớ tương tác với vi điều khiển thông qua UART hoặc RS232/RS485. DGUS\_V7.647 hỗ trợ mô phỏng trước khi nạp và xuất file cấu hình để nạp vào màn hình thông qua thẻ SD. Đây là công cụ quan trọng trong việc xây dựng giao diện trực quan, chuyên nghiệp cho các hệ thống nhúng.



***Hình 2. 25*** *DGUS\_V7.647*

### SolidWorks

SolidWorks là phần mềm thiết kế cơ khí 3D chuyên nghiệp, được sử dụng rộng rãi trong các lĩnh vực kỹ thuật và công nghiệp. Phần mềm cho phép người dùng tạo mô hình 3D chi tiết, lắp ráp các bộ phận, và mô phỏng hoạt động cơ cấu. SolidWorks hỗ trợ cả bản vẽ kỹ thuật 2D và xuất file cho in 3D hoặc gia công CNC, giúp tăng độ chính xác và hiệu quả trong quá trình thiết kế – chế tạo sản phẩm.



***Hình 2. 26*** *SolidWorks*

# THIẾT KẾ HỆ THỐNG

## Yêu cầu hệ thống

Hệ thống được xây dựng phải đảm bảo một số yêu cầu sau đây:

* Hệ thống có 2 chế độ sạc linh hoạt tuỳ vào mục đích sử dụng, tương tác thân thiện với người dùng chế độ ONLINE (Quét mã QR) và chế độ OFFLINE (Thao tác trên màn hình HMI).
* Hệ thống đưa được các thông tin sạc trong quá trình sạc lên App để tiện lợi cho mục đích theo dõi và kiểm soát phiên sạc của người chủ trạm sạc cũng như là người dùng.
* Hệ thống cung cấp 2 cổng sạc trên trụ có thể sạc cùng lúc tại cùng một thời điểm.
* Hệ thống cho phép người chủ trạm sạc cài đặt và điều chỉnh các thông số (giá điện, mật khẩu trụ sạc).
* Hệ thống phải có tính chính xác tương đối cao.

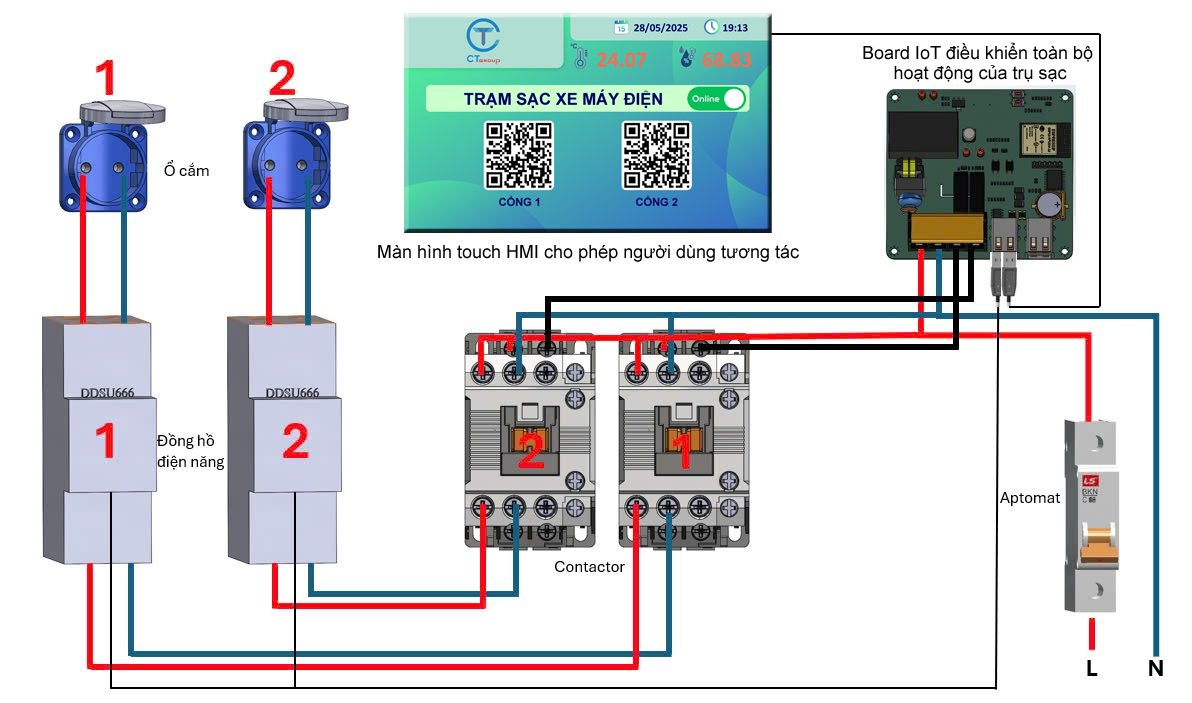
### Về phần cứng

Từ những yêu cầu trên kết hợp với sự hướng dẫn của GVHD và khả năng của bản thân, nhóm đã quyết định sử dụng vi điều khiển ESP32 là bộ xử lý chính điều khiển toàn bộ hoạt động của trụ sạc, cũng như kết nối truyền dữ liệu về App. Kết hợp cùng với màn hình touch HMI cho phép người dùng tương tác với trụ sạc, như chọn cổng sạc, chọn số tiền dự tính trả cho lần sạc (sạc ở chế độ OFFLINE), hiển thị các thông số khi sạc để cho người dùng có thể theo dõi, như nhiệt độ, độ ẩm của trụ sạc, thời gian sạc, số KWh, công suất, số tiền sạc khách sẽ phải trả tại mỗi cổng. Thiết kế board mạch IoT chuyển đổi nguồn 220V AC sang 5V và 3.3V để cấp cho vi điều khiển ESP32 cùng 1 số ngoại vi. Kết hợp với đọc giá trị trả về từ 2 đồng hồ đo điện năng DDSU666 theo chuẩn RS485.

Cùng với mạch đóng ngắt Relay để điều khiển ON/OFF cổng sạc. Vi điều khiển ESP32 ra chân để kết nối thêm với các cảm biến như SHT31, DS3231. Board mạch IoT tích hợp được nhiều chức năng nổi bật như:

* Cho phép người chủ trạm sạc và người sử dụng có thể giám sát việc sạc xe trên App hoặc trực tiếp trên màn hình HMI ở trụ sạc.
* Người chủ trạm sạc có thể thay đổi được giá điện và mật khẩu của trụ sạc.
* Tích hợp màn hình HMI thuận tiện cho việc thao tác sạc xe, giao diện thân thiện với người dùng.
* Có bảo vệ chống quá tải, ngắn mạch, chống rò điện, khi sạc đầy tự động ngắt

Từ những yêu cầu và tính năng của trụ sạc nhóm xây dựng sơ đồ khối tổng quát cho hệ thống như hình 3.1



***Hình 3. 1*** *Sơ đồ khối phần cứng của trụ sạc xe điện*

Về mặt cơ khí, nhóm đã thiết kế mô hình trụ sạc với cấu trúc vững chắc, đảm bảo tính an toàn, thẩm mỹ khi thi công lắp đặt ngoài trời. Khung trụ được chế tạo từ vật liệu kim loại sơn tĩnh điện để chống gỉ sét, chịu được điều kiện thời tiết khắc nghiệt. Bên trong trụ được bố trí hợp lý để lắp đặt các linh kiện điện tử như board mạch điều khiển, đồng hồ đo điện, contacter và các cảm biến mà vẫn đảm bảo không gian thông thoáng giúp tản nhiệt hiệu quả. Mặt sau trụ có cửa mở với khóa bảo vệ để thuận tiện cho việc bảo trì và kiểm tra thiết bị. Mặt trước thiết kế để màn hình HMI cho người dùng thuận tiện thao tác. Nhóm lựa chọn phần cổng sạc thụt vào trong và có bảo vệ bằng nhựa và nắp che để hạn chế việc mất an toàn điện khi sạc. Phần trên trụ sạc nhóm có thiết kế hình mái để nước mưa không bị tồn đọng. Ngoài ra, nhóm cũng bố trí các lỗ thoát khí và khe hở ở 2 bên trụ sạc nhằm chống tụ ẩm bên trong. Đỉnh trụ có thể được trang bị tấm năng lượng mặt trời hoặc nắp che mưa tùy theo điều kiện triển khai. Thiết kế mô hình đảm bảo dễ dàng mở rộng và lắp đặt thực tế trong các bãi đỗ xe, khu dân cư, các quán cafe hoặc các địa điểm công cộng.

### Về phần mềm

Để đáp ứng các yêu cầu đã được đặt ra, khi mới cấp nguồn thì hệ thống sẽ đọc các thông tin của trụ sạc được lưu trong EEPROM và thông tin của các phiên sạc trước đó được lưu trong thẻ nhớ SD Card. Trái tim của trụ sạc sẽ là board IoT điều khiển toàn bộ hoạt động của trụ sạc, với các chức năng chính như sau:

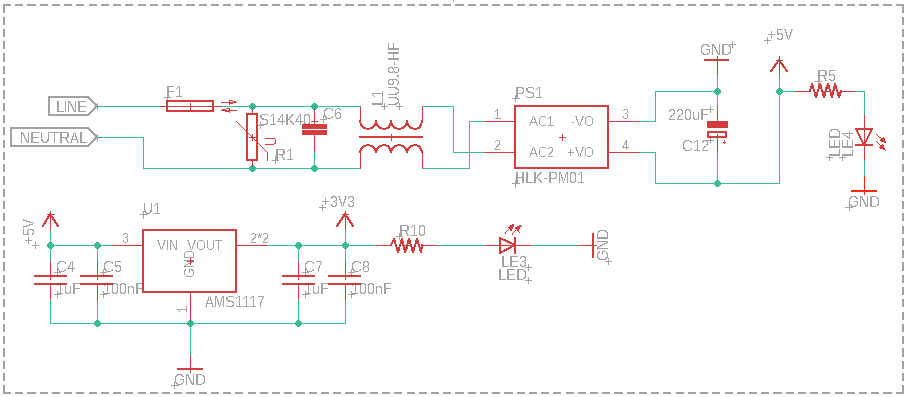
* Nhận lệnh chọn cổng sạc của người dùng thông qua màn hình HMI
* Nhận lệnh sạc cho xe tương ứng số tiền người dùng trả, OFFLINE thì thông qua nhập số tiền trên màn hình HMI; ONLINE thì thông qua chuyển khoản bằng quét QR Code trên màn hình HMI
* Điều khiển cấp nguồn điện để sạc cho xe máy điện thông qua 2 khởi động từ cho 2 cổng sạc
* Đọc giá trị đo năng lượng điện đã sạc cho xe điện từ 2 đồng hồ đo điện năng thông qua cổng giao tiếp RS-485, từ đó tính toán thời gian sạc cho xe tương ứng với số tiền mà người dùng chi trả.
* Xác nhận mật khẩu đăng nhập để cho phép nhân viên trực có thể đăng nhập vào trụ sạc để thiết lập các thông số của trụ sạc, cũng như cho phép nhân viên trực nhập số tiền sạc xe cho người dùng.
* Đọc các giá trị nhiệt độ và độ ẩm từ cảm biến SHT31 thông qua cổng giao tiếp I2C để hiển thị trên màn hình HMI, cũng là để theo dõi nếu có sự cố quá nhiệt trong trụ sạc do chập điện, quá tải sạc,... có thể gây ra tình trạng cháy trụ sạc.
* Board IoT có tích hợp mô-đun thời gian thực (RTC) để hiển thị thời gian thực trên màn hình HMI, cũng như dùng để tính toán chính xác thời gian sạc cho xe của người dùng tương ứng số tiền mà người dùng chi trả.
* Board IoT có tính năng tự động cập nhập đơn giá điện theo đơn giá thực tế từng ngày để kịp thời tính toán tiền sạc của người dùng một cách chính xác nhất
* Dữ liệu của trụ sạc sẽ được hiển thị real-time trên Mobile App để người dùng có thể theo dõi.

## Thiết kế phần cơ khí

## Thiết kế phần cứng

### Sơ đồ nguyên lý

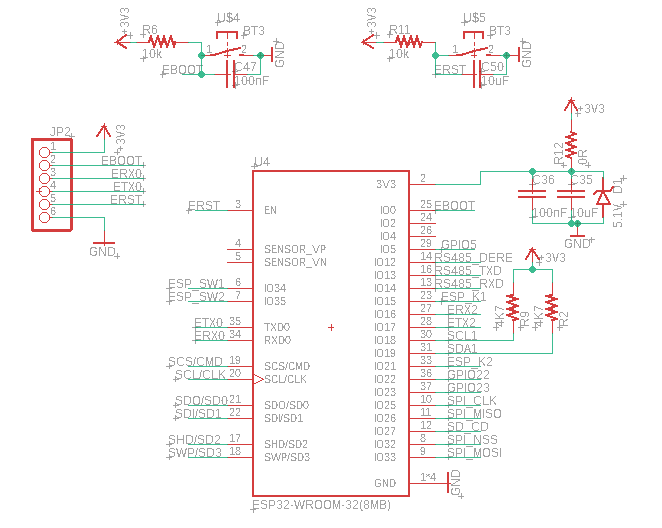
* Khối nguồn:



***Hình 3. 2*** *Sơ đồ nguyên lý mạch chuyển đổi nguồn 220VAC sang 5VDC và 3.3VDC*

Khối nguồn có tính năng bảo vệ chống sét đánh vào mạch và bảo vệ quá tải cho board mạch trụ sạc xe điện. Trong khối nguốn này cùng con HLK-PM01 để chuyển đổi nguồn điện 220VAC sang 5VDC. Thêm vào đó là khối mạch nhỏ sử dụng IC ổn áp AMS1117-3,3V để chuyển đổi 5VDC xuống 3.3VDC cấp cho vi điều khiển ESP32. Khối nguồn này có sử dụng các tụ điện C6 và cuộn cảm L1 (40mH UU9.8-403) để lọc nhiễu cho nguồn AC.

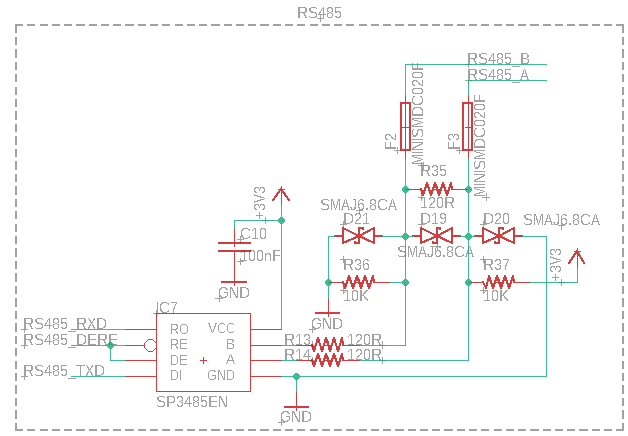
* Khối ESP32:



***Hình 3. 3*** *Sơ đồ nguyên lý mạch ESP32-WROOM-32E-N4*

ESP32-WROOM-32E-N4 sử dụng 2 nút nhất BOOT và RESET để nạp chương trình firmware. Nhóm cũng thiết kế 2 điện trở kéo lên 4k7 cho đường bus I2C và ra chân để nạp chương trình firmware cho chip.

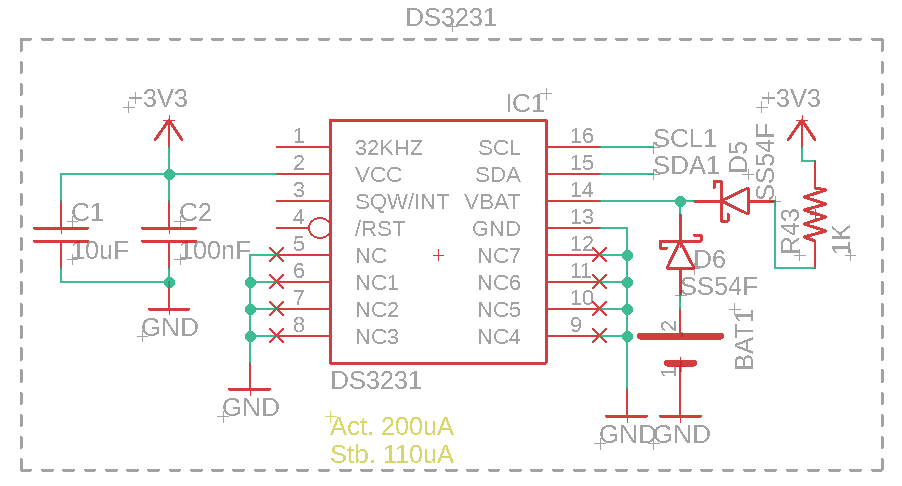
* Khối đọc RS485:



***Hình 3. 4*** *Sơ đồ nguyên lý mạch đọc dữ liệu RS485*

Khối đọc RS485 sử dụng IC SP3485EN để thực hiện giao tiếp truyền nhận dữ liệu theo chuẩn RS485. Hai điện trở 120Ω (R13, R14) được mắc đối xứng để làm điện trở kết thúc (termination resistor), giúp giảm nhiễu và đảm bảo tín hiệu ổn định trên đường truyền dài. Ngoài ra, các diode TVS (SMAJ6.8CA) và cầu chì tự phục hồi (F2) được sử dụng để bảo vệ mạch khỏi xung điện và quá dòng, nâng cao độ an toàn và độ bền của hệ thống. Tín hiệu điều khiển chân DE/RE giúp lựa chọn chế độ truyền hoặc nhận trong giao tiếp bán song công.

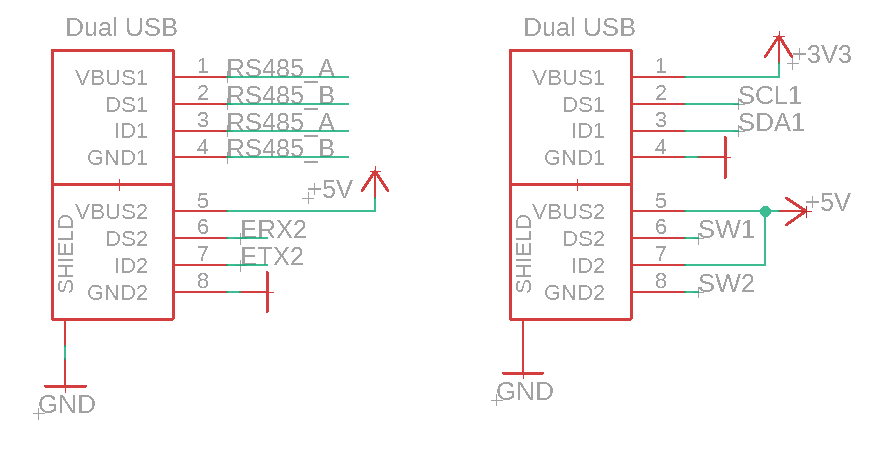
* Khối DS3231:



***Hình 3. 5*** *Sơ đồ nguyên lý DS3231*

Mạch trạm sạc xe máy điện cần lưu trữ thời gian ngay cả khi mạch mất nguồn. Vì vậy, nhóm sử dụng IC DS3231 và 1 pin CMOS 3.3V để đếm thời gian và lưu trữ thời gian khi mạch bị mất điện đột ngột.

* Khối Sensor:



***Hình 3. 6*** *Sơ đồ nguyên lý cụm cảm biến*

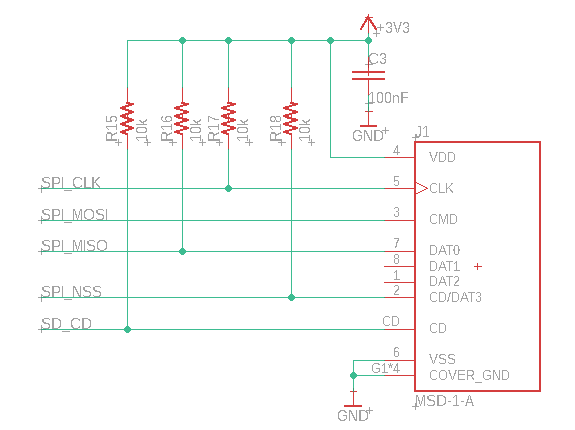
Cảm biến SHT31 sử dụng chuẩn giao tiếp I2C, hoạt động ở mức điện áp 3.3V.

Đồng hồ đo điện DDSU666 sử dụng chuẩn giao tiếp RS485.

Màn hình DWIN HMI sừ dụng chuẩn giao tiếp UART.

Nhóm thiết kế thành 2 cổng dual USB để tiện lợi cho việc lắp đặt và di chuyển.

* Khối SD Card:

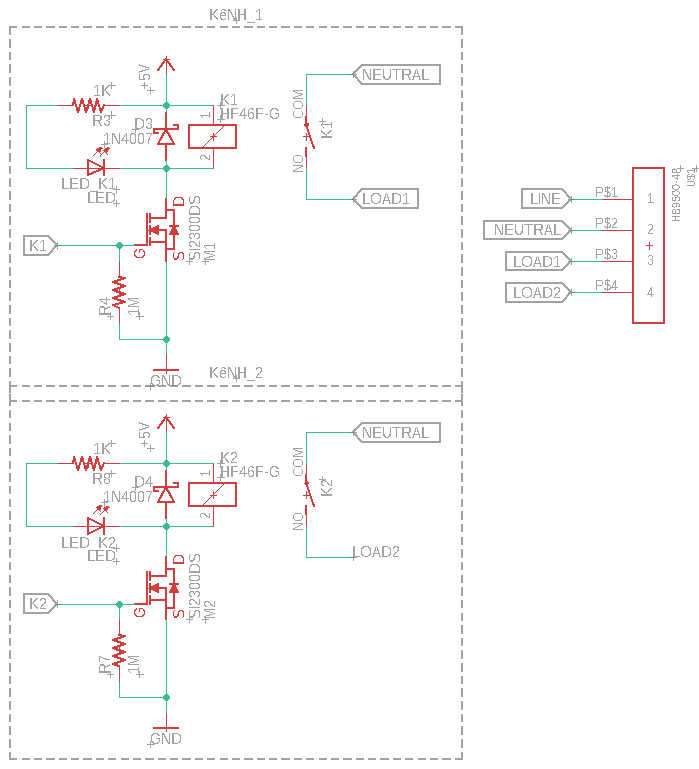


***Hình 3. 7*** *Sơ đồ nguyên lý thẻ nhớ SD Card*

Khối SD Card sử dụng chuẩn giao tiếp SPI để kết nối với vi điều khiển. Tụ lọc C3 (100nF) giúp khử nhiễu cho nguồn VDD (3.3V) của thẻ nhớ. Các điện trở kéo lên để tránh các chân tín hiệu ở trạng thái treo.

SD Card giao tiếp qua 4 đường tín hiệu SPI:

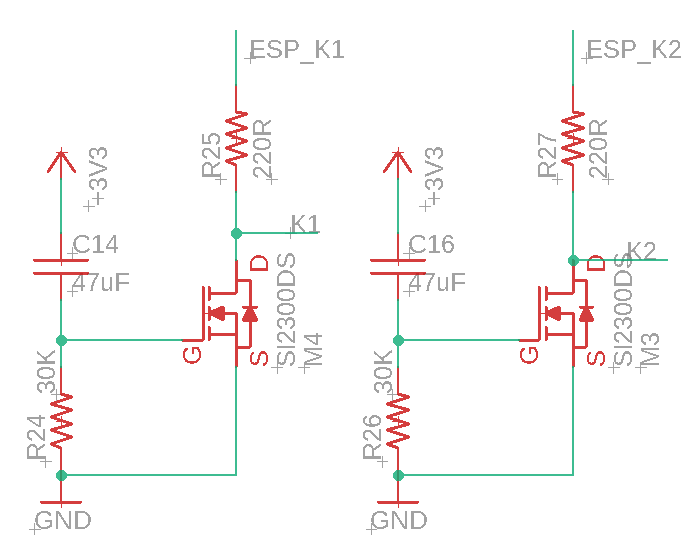
* + SPI\_CLK: Chân xung clock.
  + SPI\_MISO: Chân truyền dữ liệu từ Slave đến Master (Master In Slave Out).
  + SPI\_MOSI: Chân truyền dữ liệu từ Master đến Slave (Master Out Slave In).
  + SPI\_NSS: Chân lựa chọn thiết bị.
* Khối điều khiển Relay



***Hình 3. 8*** *Sơ đồ nguyên lý khối điều khiển Relay*

Khối điều khiển Relay được thiết kế để điều khiển đóng/ngắt hai tải độc lập (LOAD1 và LOAD2), sử dụng relay loại HF46F-G. Mỗi kênh điều khiển gồm một transistor công suất S2300DS đóng vai trò như một công tắc điều khiển dòng kích relay. Tín hiệu kích được đưa vào chân G của transistor thông qua điện trở phân áp và được bảo vệ bằng diode 1N4007 để chống dòng điện ngược khi relay ngắt. Mỗi kênh còn có thêm LED hiển thị trạng thái hoạt động và được cách ly bằng diode. Khi transistor dẫn, relay đóng tiếp điểm NO, cấp điện cho tải tương ứng. Thiết kế này giúp điều khiển tải xoay chiều một cách an toàn, hiệu quả và cách ly tốt với mạch điều khiển.

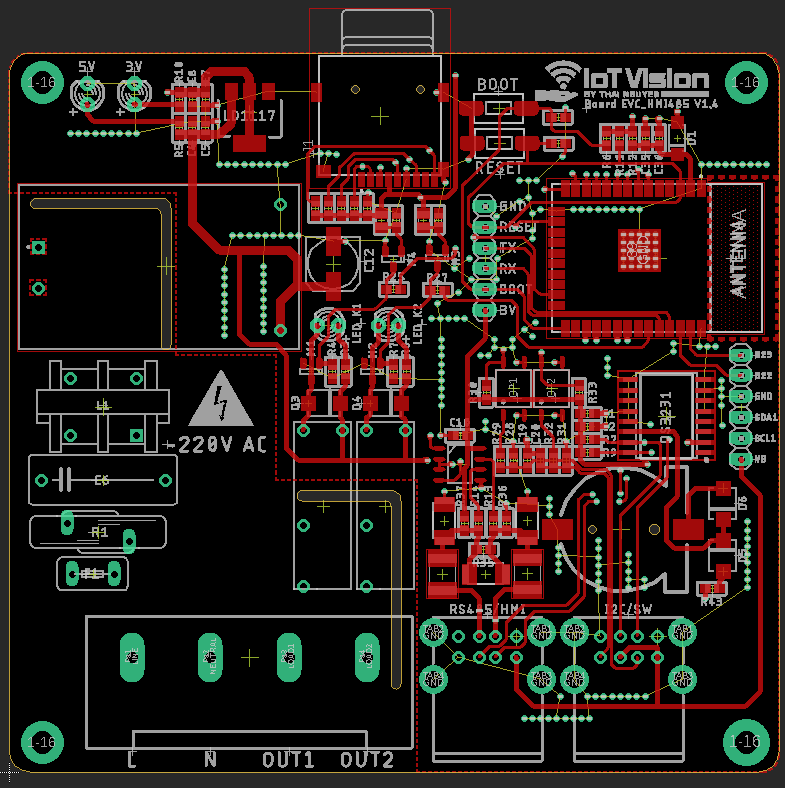
* Khối bảo vệ chân kích Relay chống kích khi Reset



***Hình 3. 9*** *Sơ đồ nguyên lý khối chống kích relay khi reset*

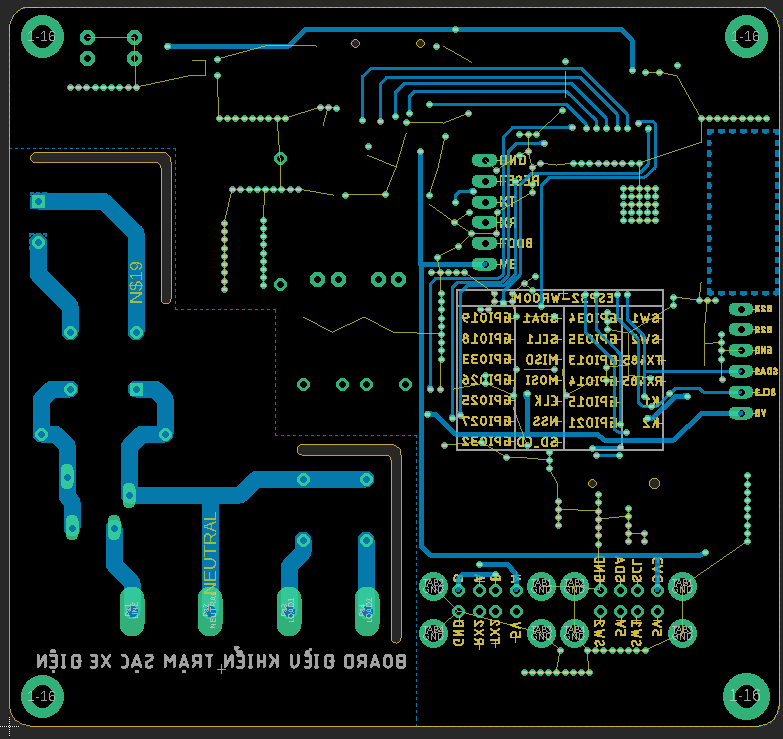
### Layout PCB

#### Mặt trên PCB



***Hình 3. 10*** *Mặt trên PCB board điều khiển trụ sạc xe máy điện*

#### Mặt dưới PCB



**Hình 3. 11** *Mặt dưới PCB board điều khiển trụ sạc xe máy điện*

## Thiết kế phần mềm

### Thiết kế chương trình điều khiển và giám sát toàn bộ hoạt động của trụ sạc

#### Tổng quan ý tưởng

Board IoT sẽ nhận lệnh chọn cổng sạc bắt đầu sạc từ thao tác người dùng thông qua màn hình HMI hoặc là nhận lệnh chọn cổng sạc bắt đầu sạc thông qua chuyển khoản bằng quét QR Code trên màn hình HMI.

Khi nhận được tín hiệu bắt đầu sạc thì Board sẽ điều khiển cấp nguồn điện để sạc cho xe máy điện thông qua khởi động từ. Đọc giá trị đo năng lượng điện đã sạc cho xe điện từ 2 đồng hồ đo điện năng thông qua cổng giao tiếp RS-485. Để việc đọc và tính toán chính xác cho lượng điện năng tiêu thụ thì cứ sau 1s thì board sẽ ngắt và đọc giá trị công suất tải tức thời và cộng dồn lại. Đồng thời gửi dữ liệu của phiên sạc lên màn hình HMI và App. Khi nhận lệnh kết thúc sạc từ màn hình HMI hoặc khi đã sạc hết số tiền chuyển khoản bằng quét mã QR thì board điều khiển ngừng cấp nguồn điện dừng sạc cho thiết bị. Chuyển trang giao diện HMI sang trang hiển thị thông tin khi kết thúc sạc. Bao gồm các thông tin về khoảng thời gian sạc, lượng điện năng tiêu thụ và số tiền khách hàng phải trả. Đồng thời lưu lại lịch sử phiên sạc vào thẻ nhớ SD Card bằng giao thức SPI. Và hiển thị các thông tin đó trong mục “Lịch Sử Sạc”. Khi sạc ở chế độ ONLINE thì khi đầy pin thì hệ thống sẽ tự động ngắt kết nối với nguồn điện. Khi ở chế độ không sạc Đọc các giá trị nhiệt độ và độ ẩm từ cảm biến SHT31 thông qua cổng giao tiếp I2C để hiển thị trên màn hình HMI và trên App cũng là để theo dõi nếu có sự cố quá nhiệt trong trụ sạc do chập điện, quá tải sạc,... có thể gây ra tình trạng cháy trụ sạc.

Có trang cài đặt để người chủ trạm sạc đăng nhập vào để thay đổi linh hoạt các thông số của trụ sạc như là mật khẩu trụ sạc, đơn giá điện, thông tin về Wifi.

#### Chương trình của MCU ESP32

#### Lưu đồ giải thuật

### Thiết kế giao diện HMI

#### Yêu cầu thiết kế

Giao diện HMI được xây dựng với mục tiêu trực quan, dễ sử dụng và đáp ứng đầy đủ các chức năng cần thiết trong quá trình sạc xe điện. Người dùng có thể chọn một trong hai chế độ sạc: ONLINE (quét mã QR) hoặc OFFLINE (người dùng hoặc chủ trạm sạc thao tác trực tiếp trên màn hình HMI). Các yêu cầu cụ thể cho giao diện gồm:

* **Hiển thị trạng thái sạc:** Giao diện cần thông báo rõ ràng tình trạng hiện tại của trụ sạc, như: “Đang sạc”, “Kết thúc sạc”, “Hoàn tất”, v.v.
* **Lượng điện năng đã sạc:** Hiển thị số KWh đã sạc theo định dạng: xx,xx kWh, giúp người dùng theo dõi điện năng đã sạc cho pin.
* **Chế độ sạc:** Ở chế độ sạc OFFLINE có thể chọn theo sạc theo số tiền cố định hay là sạc theo thực tế sạc
* **Cài đặt đơn giá điện:** Cho phép người quản trị trụ sạc thay đổi đơn giá điện theo thời điểm, ví dụ: 10.500 VNĐ/kWh, với khả năng thiết lập linh hoạt từ giao diện HMI.
* **Đặt lại mật khẩu quản trị trụ sạc:** Giao diện cho phép đặt lại mật khẩu cho từng trụ sạc một cách bảo mật, giúp người chủ trạm sạc kiểm soát truy cập thiết bị một cách chặt chẽ.

#### Thiết kế giao diện chọn chế độ sạc và cổng sạc

Giao diện hiển thị mặc định là ở chế độ ONLINE với 2 mã QR để chuyển khoản tương ứng với 2 cổng sạc. Có nút chuyển đổi sang màn hình chế độ OFFLINE một cách linh hoạt. Được thể hiện ở hình 3. 12 và hình 3.13



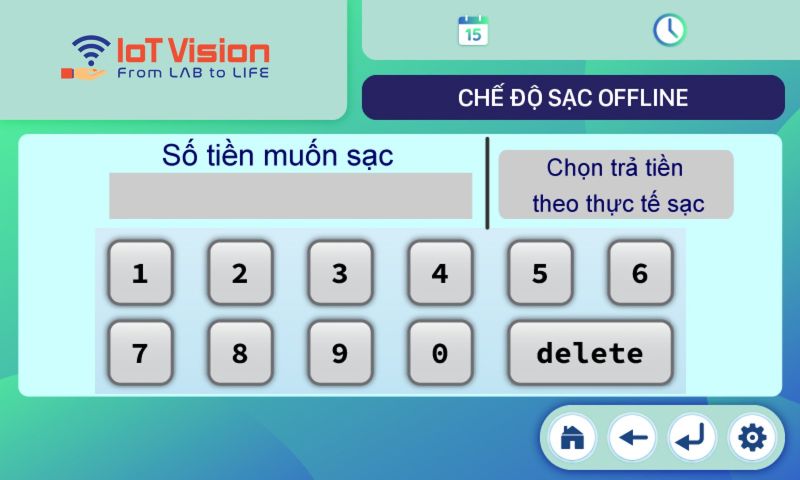
***Hình 3. 12*** *Giao diện HMI ở chế độ sạc ONLINE*



***Hình 3. 13*** *Giao diện HMI ở chế độ sạc OFFLINE*

#### Thiết kế giao diện HMI chọn chế độ sạc theo số tiền hay thực tế sạc khi sạc OFFLINE

Khi trạm sạc hoạt động ở chế độ OFFLINE (không kết nối mạng, không đồng bộ với server), hệ thống vẫn phải cho phép người dùng chọn và điều khiển quá trình sạc một cách độc lập, an toàn và dễ sử dụng. Giao diện HMI được thiết kế với 2 tùy chọn chính cho người dùng lựa chọn phương thức sạc, bao gồm nhập số tiền muốn sạc và chọn trả tiền theo thực tế sạc.



Hình 3. Giao diện HMI lựa chọn phương thức sạc khi ở chế độ OFFLINE

#### Thiết kế giao diện HMI hiển thị thông tin đang sạc và khi kết thúc sạc

Giao diện HMI trong quá trình sạc được thiết kế nhằm cung cấp đầy đủ thông tin cho người dùng một cách rõ ràng, trực quan, đồng thời hỗ trợ thao tác nhanh chóng và chính xác. Bao gồm các thông tin về trạng thái, thời gian sạc, điện năng, công suất và số tiền tạm tính. Ở chế độ OFFLINE thì màn hình có các nút để người dùng có thể chuyển sang trang khác, quay lại, xem thông tin lịch sử sạc và có thêm nút “Kết thúc sạc” để người sử dụng có thể ngắt sạc bất cứ lúc nào. Nhưng khi ở chế độ ONLINE thì mục tiêu là thao tác quét mã và sạc đơn giản nên màn hình chỉ có nút chuyển trang qua lại giữa trang thông tin 2 cổng sạc. Giao diện thông tin sạc OFFLINE được thể hiện ở hình 3.14. Còn giao diện thông tin sạc ONLINE được thể hiện ở hình 3.15.

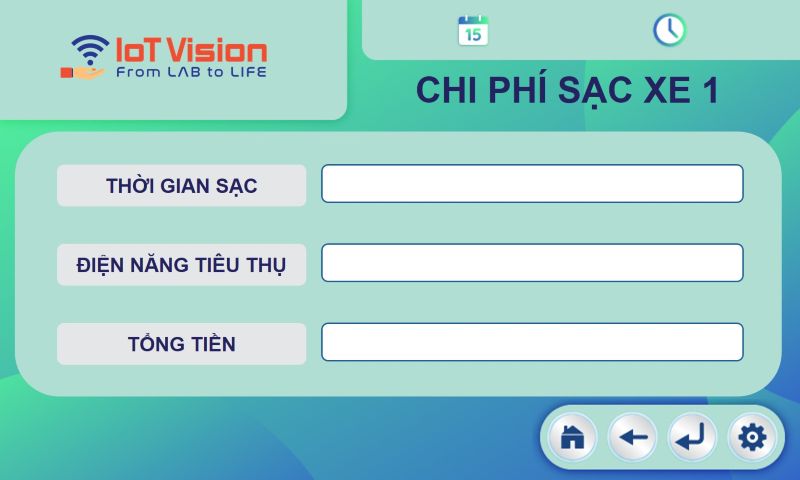


***Hình 3. 15*** *Giao diện HMI khi đang sạc OFFLINE cổng 1*



***Hình 3. 16*** *Giao diện HMI khi đang sạc ONLINE cổng 2*

Khi kết thúc sạc màn hình sẽ hiển thị khoảng thời gian sạc, lượng điện năng đã sạc và tổng số tiền phải trả.



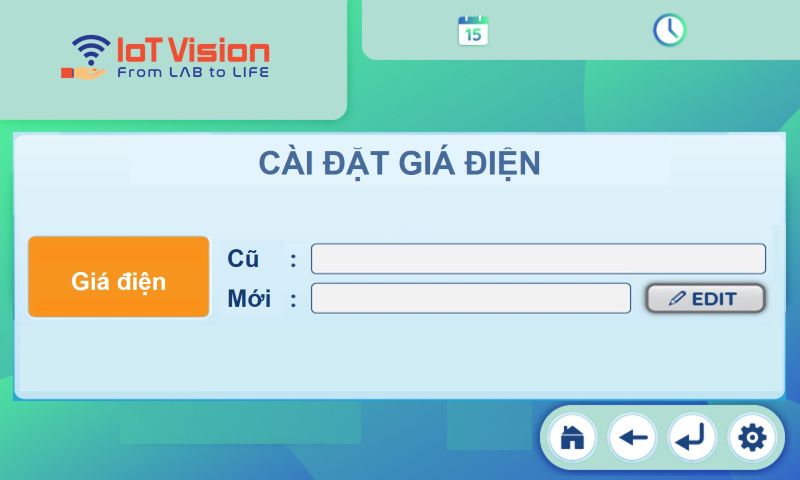
***Hình 3. 17*** *Giao diện HMI khi kết thúc sạc ở cổng 1*

#### Thiết kế giao diện HMI hiển thị và cài đặt chỉnh sửa các thông số sạc

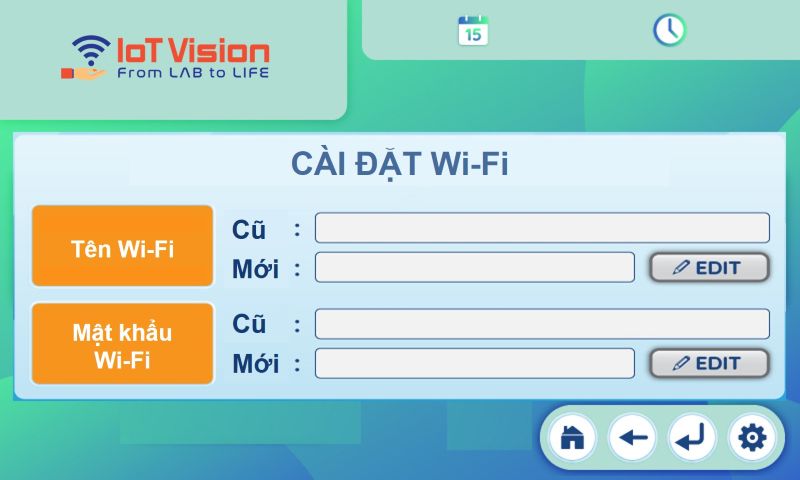
Giao diện HMI cho phép người dùng truy cập vào mục cài đặt để chỉnh sửa các thông số quan trọng như mật khẩu trụ sạc, đơn giá điện (VNĐ/kWh), và thông tin kết nối WiFi (SSID, mật khẩu). Việc thay đổi các thông số này được bảo vệ bằng mật khẩu quản trị nhằm đảm bảo tính bảo mật. Các thông tin sau khi cập nhật sẽ được lưu vào bộ nhớ để duy trì ngay cả khi mất điện.



***Hình 3. 18*** *Giao diện HMI hiển thị và cài đặt mật khẩu cho trụ sạc*



***Hình 3. 19*** *Giao diện HMI hiển thị và thay đổi giá điện*



***Hình 3. 20*** *Giao diện HMI hiển thị và thay đổi Wi Fi*

#### Thiết kế màn hình HMI hiển thị các chế độ báo lỗi

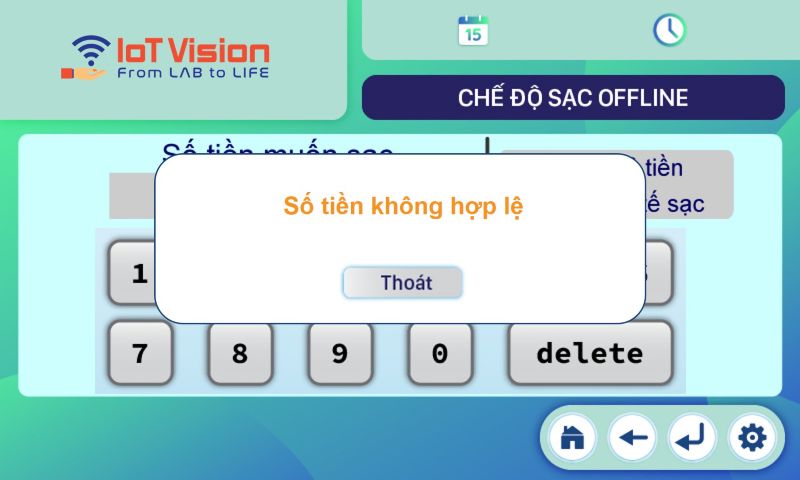
Ngoài ra giao diện còn được thiết kế với các màn hình báo lỗi khi người dùng nhập sai thông tin về số tiền, mật khẩu, chỉnh sửa nhưng chưa nhập …



***Hình 3. 21*** *Giao diện HMI thông báo lỗi nhập sai mật khẩu*



***Hình 3. 22*** *Giao diện HMI thông báo lỗi nhập sai định dạng*



***Hình 3. 23*** *Giao diện HMI thông báo lỗi nhập số tiền không hợp lệ*

# KẾT QUẢ

Nội dung chương này sẽ trình bày kết quả sau khi phần cứng của nhóm đã thi công gồm các mạch Node và Gateway cũng như phần mềm bên trong mỗi thiết bị. Ngoài ra, nhóm cũng sẽ tiến hành thực nghiệm hệ thống về độ chính xác của các cảm biến, độ ổn định ở khoảng cách xa cũng như tính linh hoạt trong việc triển khai các Node gia nhập vào hệ thống.

## Kết quả phần cứng

## Kết quả phần mềm

## Thực nghiệm độ chính xác của thiết bị với máy đo trên thị trường

## Thực nghiệm độ ổn định của hệ thống với khoảng cách trên 1000m

## Thực nghiệm tính linh hoạt của hệ thống

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Từ những kết quả nghiên cứu ở trên, nhóm rút ra được kết luận như sau:

#### Kết quả

Nhóm đã thiết kế và triển khai thành công hệ thống IoT quan trắc môi trường không khí có khả năng hoạt động theo như mong đợi của nhóm. Hệ thống triển khai được trong phạm vi rộng (2,69 km2), khoảng cách triển khai theo mô hình giữa các điểm Node đến Gateway lớn hơn 1000m (1200 – 1400m), tỷ lệ truyền nhận thành công cao (hơn 85%). Giá trị cảm biến thu thập được so với thiết bị mẫu có sai số thấp.

#### Hạn chế

Tuy nhiên, còn một số hạn chế trong quá trình triển khai hệ thống như không tạo được môi trường khí CO lý tưởng để thu thập và đánh giá mức độ chính xác của dữ liệu   
cảm biến khí CO, hệ thống không truyền nhận thành công hoặc tỉ lệ mất dữ liệu cao   
(hơn 50%) khi triển khai thiết bị trong các khu vực có nhiều vật cản giữa Node và Gateway (khu dân cư, tòa nhà cao tầng, khu công nghệ cao…). Chưa thử nghiệm   
hệ thống trong các điều kiện thời tiết xấu và không có tính năng lưu trữ dữ liệu cảm biến thu thập được.

## Hướng phát triển

Các nội dung cần cải thiện và phát triển:

* Kiểm định các dữ liệu đo được với thiết bị đo chuẩn chuyên dụng
* Xử lý dữ liệu cảm biến thu thập được theo các tiêu chuẩn về chất lượng không khí.
* Cải thiện tính năng thay đổi tần số lấy mẫu của giao thức.
* Thêm một số tính năng cho hệ thống như xoá thiết bị, xuất file dữ liệu có dạng txt, csv...
* Tích hợp module GPS + 3G, 4G LTE cho hệ thống.
* Tích hợp màn hình HMI cho phép Gateway thao tác quản lý các Node.
* Phát triển chương trình của thiết bị Node cho phép hoạt động độc lập, truyền dữ liệu trực tiếp qua Wi-Fi.
* Tích hợp truyền dữ liệu với chuẩn Ethernet và RS485 cho Gateway.
* Cải thiện và bổ sung các tính năng cho app Android.
* Tìm kiếm giải pháp sử dụng pin thay thế cho nguồn Adapter để tăng sự linh hoạt hơn cho hệ thống.
* Tăng bán kính truyền nhận lên 3000m nhưng vẫn không thay đổi thời gian lấy mẫu.
* Tích hợp thẻ nhớ microSD lưu dữ liệu đo được.
* Tích hợp các chuẩn bảo mật cho hệ thống.
* Xây dựng mô hình Machine Learning dự đoán chất lượng không khí của   
  môi trường.

TÀI LIỆU THAM KHẢO

1. Karen Rose, Scott Eldridge, Lyman Chapin, “The Internet of Things: An Overview”, 2015 The Internet Society (ISOC).
2. Adel Ismail Al-Alawi, “WiFi Technology: Future Market Challenges and Opportunities”, Journal of Computer Science 2 (1): 13-18, 2006.
3. Thạc sĩ Nguyễn Hữu Trung, “NGHIÊN CỨU, THIẾT KẾ CÔNG TẮC IOT VÀ PHÁT TRIỂN HỆ THỐNG QUẢN LÝ THIẾT BỊ IOT”.
4. Semtech Corporation, “What are LoRa® and LoRaWAN®?”, <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>
5. Cục tần số vô tuyến điện, “Mạng LPWAN cho các ứng dụng IoT”, <https://rfd.gov.vn/tin-tuc/pages/the-gioi-vo-tuyen.aspx?ItemID=2590>
6. Semtech, “SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver”, <https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf>
7. Deviot, “Giao tiếp SPI”, <https://deviot.vn/blog/giao-tiep-spi.74706311>.
8. Deviot, “Giao tiếp I2C”, <https://deviot.vn/blog/giao-tiep-i2c.05019305>.
9. BKAII, “Tìm hiểu về truyền thông UART”, <https://bkaii.com.vn/tin-tuc/1099-tim-hieu-ve-truyen-thong-uart-khai-niem-nguyen-ly-va-ung-dung>.
10. PLANTOWER, “Digital universal particle concentration sensor”, <https://download.kamami.pl/p564008-PMS7003%20series%20data%20manua_English_V2.5.pdf>
11. Sensirions, “Technology to the respective measuring principles”, <https://sensirion.com/products/technology>
12. CIRCUITS DIY, “MQ7 Carbon Monoxide (CO) Sensor Module”, [https://www.circuits-diy.com/mq7-carbon-monoxide-co-gas-sensor-module/#:~:text=Working%20Principle,surface%20of%20the%20tin%20dioxide.](https://www.circuits-diy.com/mq7-carbon-monoxide-co-gas-sensor-module/%23:~:text=Working%20Principle,surface%20of%20the%20tin%20dioxide.)
13. PCB Layout Việt Nam, “Giới thiệu phần mềm KiCad”, xem ngày 30/6/2023
14. STMicroelectronics, “Product overview”, xem ngày 30/6/2023