BỘ CÔNG THƯƠNG **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC Độc lập – Tự do – Hạnh phúc**



**ĐỀ CƯƠNG ĐỀ TÀI**

**ĐỒ ÁN MÔN PHÁT TRIỂN ỨNG DỤNG IOT**

**ĐỀ TÀI : Hệ thống giám sát nhiệt độ - độ ẩm môi trường qua LORA**

**Giảng viên hướng dẫn :**  Mai Xuân Hòa

**Nhóm sinh viên thực hiện:**  Nhóm 6

**Lớp học phần: D16DTRB**

Vũ Trần Nhật Minh  **Mã sinh viên:** 21810510049

Vũ Hải Nguyên  **Mã sinh viên:** 21810550375

Bì Trọng Bình  **Mã sinh viên:** 21810550513

Phạm Bình Minh **Mã sinh viên:** 21810550434

Mạc Xuân Bình **Mã sinh viên:** 21810510042

Phan Tiến Vượng **Mã sinh viên:** 21810550388

Hà Nội, tháng 4 năm 2025

**LỜI MỞ ĐẦU**

Trong thời đại công nghiệp 4.0 hiện nay, việc áp dụng các giải pháp công nghệ vào quản lý và giám sát môi trường ngày càng trở nên phổ biến và cần thiết. Các thông số như nhiệt độ và độ ẩm đóng vai trò quan trọng trong nhiều lĩnh vực khác nhau, từ nông nghiệp, sản xuất công nghiệp, đến các môi trường lưu trữ như kho bãi, phòng thí nghiệm, hay thậm chí là không gian sống. Việc giám sát và kiểm soát chính xác các thông số này không chỉ giúp bảo vệ chất lượng sản phẩm, tối ưu hóa quy trình sản xuất mà còn đảm bảo môi trường sống và làm việc an toàn, ổn định.

Hệ thống giám sát nhiệt độ, độ ẩm qua LoRa (Long Range) là một giải pháp hiệu quả và hiện đại, đáp ứng nhu cầu quản lý từ xa các thông số môi trường. Công nghệ LoRa sử dụng sóng vô tuyến tần số thấp để truyền dữ liệu với phạm vi rộng và tiêu thụ năng lượng thấp, giúp hệ thống có khả năng hoạt động liên tục trong thời gian dài mà không cần nhiều năng lượng. Điều này đặc biệt hữu ích cho việc triển khai tại các khu vực rộng lớn, khó tiếp cận, hoặc nơi cần theo dõi nhiều điểm từ xa.

Đề tài "Hệ thống giám sát nhiệt độ, độ ẩm qua LoRa" nhằm nghiên cứu và phát triển một hệ thống giám sát môi trường tự động, sử dụng công nghệ LoRa để thu thập và truyền tải dữ liệu một cách chính xác, hiệu quả. Hệ thống không chỉ giúp theo dõi liên tục các thông số nhiệt độ và độ ẩm mà còn cho phép cảnh báo kịp thời khi các giá trị vượt ngưỡng an toàn, giúp người quản lý có thể phản ứng và điều chỉnh môi trường nhanh chóng.

Bài nghiên cứu sẽ trình bày về nguyên lý hoạt động của công nghệ LoRa, thiết kế và triển khai hệ thống giám sát, cũng như phân tích hiệu quả và ứng dụng thực tiễn của hệ thống này trong các lĩnh vực khác nhau. Đây là một hướng đi mới, mang tính ứng dụng cao, góp phần cải thiện và tối ưu hóa các quy trình giám sát môi trường trong tương lai.

**Mục lục**

[**LỜI MỞ ĐẦU** 1](#_Toc180429045)

[**NỘI DUNG** 5](#_Toc180429046)

[**CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN** 5](#_Toc180429047)

[**1.1. Tìm hiểu chung về hệ thống giám sát nhiệt độ, độ ẩm qua LoRa** 5](#_Toc180429048)

[**1.2. Một số nghiên cứu về hệ thống giám sát nhiệt độ, độ ẩm qua LoRa** 6](#_Toc180429049)

[**1.2.1. Nghiên cứu ở nước ngoài** 6](#_Toc180429050)

[**1.2.2. Nghiên cứu ở trong nước** 8](#_Toc180429051)

[**1.2.3. Đánh giá các nghiên cứu đã công bố** 11](#_Toc180429052)

[**1.3. Xác định bài toán cần giải quyết của đề tài** 12](#_Toc180429053)

[**1.3.1. Mục tiêu phát triển cho sản phẩm** 12](#_Toc180429054)

[**1.3.2. Thông số kỹ thuật đề ra của hệ thống giám sát nhiệt độ, độ ẩm qua LoRa** 12](#_Toc180429055)

[**1.4. Kết thúc chương 1** 13](#_Toc180429056)

[**CHƯƠNG II: PHÂN TÍCH HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ, ĐỘ ẨM QUA LORA** 14](#_Toc180429057)

[**2.1** **Phân tích, thiết kế tổng quan hệ thống** 14](#_Toc180429058)

[**2.2** **Phân tích, thiết kế mô hình hệ thống giám sát môi trường** 15](#_Toc180429059)

[**2.2.1** **Phân tích thiết kế sơ đồ khối tổng quan của hệ thống giám sát môi trường** 15](#_Toc180429060)

[**2.2.2** **Phân tích, lựa chọn linh kiện hệ thống giám sát môi trường** 17](#_Toc180429061)

[**2.2.3** **Thiết kế mạch in** 37](#_Toc180429062)

[**2.2.4** **Thiết kế sơ đồ thuật toán** 48](#_Toc180429063)

[**CHƯƠNG III: CHẾ TẠO VÀ THỬ NGHIỆM HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ, ĐỘ ẨM QUA LORA** 53](#_Toc180429064)

[**3.1. Chế tạo hệ thống giám sát nhiệt độ, độ ẩm qua lora** 53](#_Toc180429065)

[**3.2. Kiểm tra và đánh giá** 55](#_Toc180429066)

[3.2.1. Thử nghiệm độ chính xác của các cảm biến nhiệt độ DHT11 55](#_Toc180429067)

[3.2.2. Thử nghiệm độ chính xác của module LORA 56](#_Toc180429068)

[3.2.3. Đánh giá chung các thử nghiệm 56](#_Toc180429069)

[**3.2.4 Mã code chạy chương trình** 57](#_Toc180429070)

[**3.3. Kết luận chương 3** 64](#_Toc180429071)

[**KẾT LUẬN CHUNG** 65](#_Toc180429072)

[**TÀI LIỆU THAM KHẢO** 66](#_Toc180429073)

**DANH MỤC HÌNH VẼ**

[Hình 1. 1 Hoạt động về việc đo nhiệt độ, độ ẩm thông qua LoRa 5](#_Toc180425881)

[Hình 1. 2 Hoạt động về việc đo nhiệt độ, độ ẩm thông qua LoRa 7](#_Toc180425882)

[Hình 1. 3 Sơ đồ khối hệ thống thu nhập và điều khiển các thông số môi trường 9](#_Toc180425883)

[Hình 2. 1 Sơ đồ khối tổng quan của hệ thống 14](file:///D:\Đồ%20án%20ĐTVT\Báo%20cáo%20đồ%20án%20bản%20đầy%20đủ.docx#_Toc180425886)

[Hình 2. 2 Sơ đồ khối một hệ thống giám sát môi trường 16](#_Toc180425887)

[Hình 2. 3 Cảm biến nhiệt độ - độ ẩm DHT11 18](#_Toc180425888)

[Hình 2. 4 Sơ đồ dao động của thạch anh 19](file:///D:\Đồ%20án%20ĐTVT\Báo%20cáo%20đồ%20án%20bản%20đầy%20đủ.docx#_Toc180425889)

[Hình 2. 5 Module thời gian thực RTC DS3231 21](#_Toc180425890)

[Hình 2. 6 Màn hình TFT LCD 1.44 inch giao tiếp SPI 23](#_Toc180425891)

[Hình 2. 7 Module LORA 28](#_Toc180425892)

[Hình 2. 8 ESP32 DEVKIT V1 29](#_Toc180425893)

[Hình 2. 9 Chip ESP32 30](#_Toc180425894)

[Hình 2. 10 Sơ đồ chân ESP32 DEVKIT V1 30](#_Toc180425895)

[Hình 2. 11 Pin 18650 35](#_Toc180425896)

[Hình 2. 12 Mạch sạc pin 1cell 18650 có bảo vệ TP4056 1A với cổng nối micro USB 37](#_Toc180425897)

[Hình 2. 13 Sơ đồ nguyên lý khối nguồn 38](#_Toc180425898)

[Hình 2. 14 Sơ đồ nguyên lý cảm biến DHT11 38](#_Toc180425899)

[Hình 2. 15 Sơ đồ nguyên lý khối điều khiển 39](#_Toc180425900)

[Hình 2. 16 Sơ đồ nguyên lý khối thông báo 39](#_Toc180425901)

[Hình 2. 17 Sơ đồ nguyên lý khối thời gian thực 40](#_Toc180425902)

[Hình 2. 18 Sơ đồ nguyên lý khối hiển thị 40](#_Toc180425903)

[Hình 2. 19 Sơ đồ nguyên lý khối truyền thông 41](#_Toc180425904)

[Hình 2. 20 Sơ đồ khối xử lý trạm từ xa 41](#_Toc180425905)

[Hình 2. 21 Sơ đồ nguyên lý khối xử lý trung tâm 42](#_Toc180425906)

[Hình 2. 22 Sơ đồ nguyên lý mạch phát 43](#_Toc180425907)

[Hình 2. 23 Sơ đồ nguyên lý mạch thu 44](#_Toc180425908)

[Hình 2. 24 Mạch in 2D mạch truyền 46](#_Toc180425909)

[Hình 2. 25 Mạch in 2D mạch thu 47](#_Toc180425910)

[Hình 3. 1 Cấu hình thông số mũi dao CNC 53](#_Toc180425940)

[Hình 3. 2 Mô phỏng đường đi của dao khi khắc mạch 54](#_Toc180425941)

[Hình 3. 3 Hàn và lắp linh kiện lên mạch điều khiển 2 mặt trước và sau 55](#_Toc180425942)

**DANH MỤC BẢNG**

[Bảng 1: Chức năng các chân ESP32 31](#_Toc180426194)

[Bảng 2: Các mức điện áp sử dụng mạch 35](#_Toc180426195)

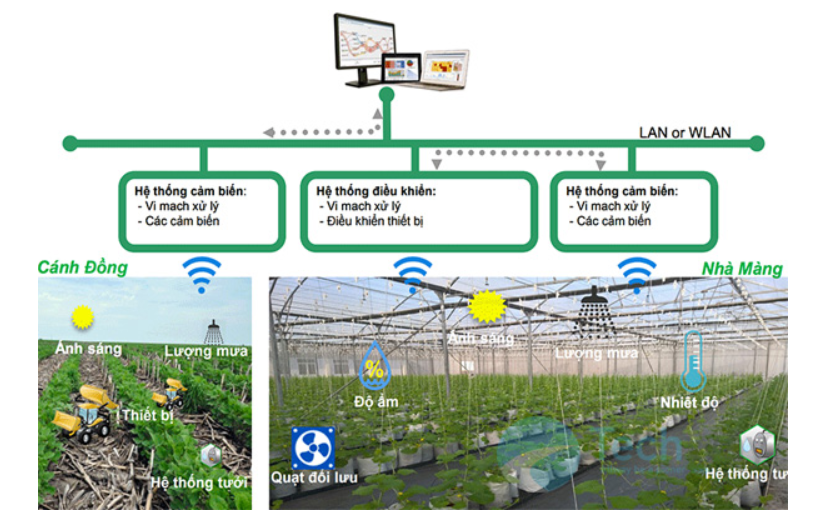
[Bảng 3: Bảng so sánh kết quả đo 55](#_Toc180426196)

[Bảng 4: Kết quả so sánh độ chính của module LORA 56](#_Toc180426197)

**NỘI DUNG**

**CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN**

* 1. **. Tìm hiểu chung về hệ thống giám sát nhiệt độ, độ ẩm qua LoRa**

Trong bối cảnh biến đổi khí hậu và sự gia tăng nhu cầu theo dõi môi trường, hệ thống giám sát nhiệt độ môi trường qua công nghệ LoRa (Long Range) đã trở thành một giải pháp hiệu quả và thiết thực. Công nghệ LoRa cho phép truyền tải dữ liệu với khoảng cách xa, đồng thời tiêu thụ năng lượng thấp, điều này đặc biệt quan trọng trong các ứng dụng giám sát từ xa. Hệ thống giám sát nhiệt độ môi trường qua LoRa gồm các thành phần chính bao gồm cảm biến nhiệt độ, module LoRa và nền tảng xử lý dữ liệu. Cảm biến nhiệt độ thu thập thông tin về nhiệt độ không khí và môi trường xung quanh. Dữ liệu này được truyền qua module LoRa tới trung tâm xử lý, nơi thông tin được phân tích và hiển thị trên các nền tảng quản lý, nhằm hỗ trợ người dùng theo dõi và ra quyết định kịp thời. Một trong những lợi ích lớn nhất của hệ thống này là khả năng hoạt động trong những điều kiện khó khăn, nơi mà mạng di động hoặc Internet có thể không ổn định. Nhờ vào hệ thống LoRa, người dùng có thể nhận được dữ liệu liên tục và chính xác, từ đó nâng cao khả năng dự báo và ứng phó với các biến động môi trường.

Hình 1. 1 Hoạt động về việc đo nhiệt độ, độ ẩm thông qua LoRa

Những tính năng cơ bản của việc giám sát nhiệt độ, độ ẩm qua LoRa:

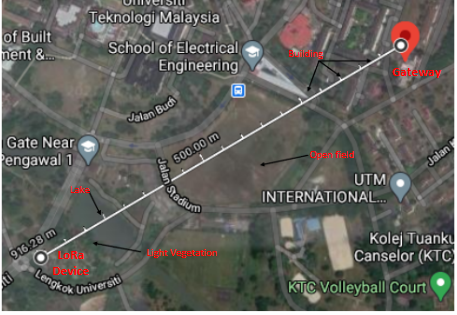
* Đo và ghi dữ liệu nhiệt độ
* Giám sát nhiệt độ, độ ẩm từ xa
* Cảnh báo khi nhiệt độ vượt ngưỡng
* Lưu trữ và phân tích dữ liệu

**1.2. Một số nghiên cứu về hệ thống giám sát nhiệt độ, độ ẩm qua LoRa**

**1.2.1. Nghiên cứu ở nước ngoài**

Một số nghiên cứu ở nước ngoài:

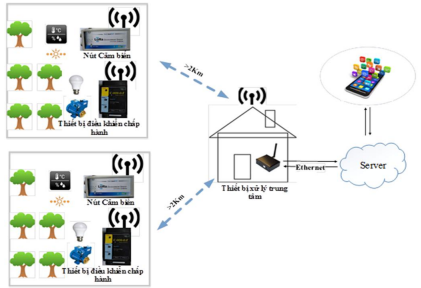
Sự biến đổi theo chu kỳ ngày và các tác động của điều kiện thời tiết dựa trên liên kết LoRa trong một thiết lập LoRaWAN đã được trình bày. Phân tích tác động của nhiệt độ trên tàu, nhiệt độ khí quyển, độ ẩm tương đối và bức xạ mặt trời đối với liên kết truyền thông LoRa đã được cung cấp. Tác động của điều kiện mưa đến liên kết LoRa cũng đã được báo cáo. Phép đo RSSI trong ba ngày liên tiếp cho thấy các mẫu tương tự với tín hiệu mạnh hơn trong suốt ban ngày khi bức xạ mặt trời cao. Một mối tương quan dương giữa bức xạ mặt trời và RSSI đã được quan sát. Điều này có thể do ảnh hưởng của các hoạt động trong tầng đối lưu như chỉ số khúc xạ, tầng bay hơi và các điều kiện khí tượng khác mà đã tồn tại. Cần nghiên cứu thêm để điều tra tác động của các điều kiện khí tượng như nhiệt độ biển dương, sự tồn tại của các trung tâm áp suất cao, độ khúc xạ của tầng đối lưu thấp. Sự gia tăng bức xạ mặt trời cũng làm tăng nhiệt độ trên thiết bị và trong khí quyển, điều này có thể được gây ra bởi nhiệt độ tiếp giáp của bán dẫn trong thiết bị LoRa. Lượng mưa gần như không tác động đến RSSI trong hệ thống LoRa, ngay cả khi có tỷ lệ mưa cao 180 mm/h từ nghiên cứu này ở khoảng cách truyền 916,25 m và mức RSSI được ghi nhận. Đặc điểm được quan sát này cho thấy cần thực hiện thêm các nghiên cứu về tác động của điều kiện thời tiết lên LoRa. Các phát hiện từ nghiên cứu này sẽ hữu ích trong việc mô hình hóa sự lan truyền của LoRa (bao gồm cả các bộ thu phát IoT liên quan khác) và các nghiên cứu về độ tin cậy của ứng dụng LoRa trong các trường hợp sử dụng IoT. Các công việc trong tương lai sẽ xem xét tác động của các điều kiện khí tượng hiện hành lên RSSI cho nhiều liên kết (nhiều nút hơn), các tần số phát khác nhau.

Hình 1. 2 Hoạt động về việc đo nhiệt độ, độ ẩm thông qua LoRa

Trong khu vực thông minh của Skellefteå, miền bắc Thụy Điển, Đại học Công nghệ Luleå đã triển khai LoRaWAN cho mục đích nghiên cứu. Trong khu vực này, các loại dịch vụ IoT khác nhau đang được phát triển để cải thiện chất lượng cuộc sống của công dân. Vì các cảm biến nói trên được lan truyền rộng rãi trên khu vực thông minh này, LoRaWAN là sự lựa chọn tốt nhất cho cơ sở hạ tầng kết nối tổng thể. Mục tiêu của họ là quan sát ảnh hưởng của những thay đổi cực đoan về thời tiết dọc theo năm ở thành phố Skellefteå đến tín hiệu LoRa. Việc đánh giá tác động của nhiệt độ trong LoRa/LoRaWAN còn hạn chế trong tài liệu nhưng đã có những đóng góp. Trong tài liệu “ Đánh giá thử nghiệm về độ tin cậy của truyền thông không dây công suất thấp tầm xa LoRa.”, các tác giả trình bày một nghiên cứu thử nghiệm về độ tin cậy của LoRa trong các tình huống ngoài trời và trong nhà, cho thấy độ nhạy của LoRa với nhiệt độ và độ ẩm. Họ sử dụng môi trường phòng thí nghiệm được kiểm soát, trong đó các nút LoRa được làm nóng bằng bóng đèn, ở nhiệt độ thay đổi từ 0 đến 60°C. Trong khi đánh giá được thực hiện trong tài liệu “ Đánh giá thử nghiệm về độ tin cậy của truyền thông không dây công suất thấp tầm xa LoRa.”, tập trung vào nhiệt độ của cảm biến LoRa, chúng tôi tập trung vào nhiệt độ môi trường, bao gồm toàn bộ hệ thống (cảm biến, cổng và kênh liên lạc). Đặc tính kênh LoRa ở Nam Cực được trình bày trong tài liệu “ Cảm biến LoRa Đặc điểm kênh trạm di động đến trạm cơ sở ở Nam Cực.” Các tác giả điều tra cả sóng mang 434 MHz và 868 MHz về phạm vi phủ sóng và RSSI. Có thể liên lạc LoRa ở Nam Cực ở khoảng cách 20 km, miễn là nó ở điều kiện LoS. Giao tiếp NLoS cũng có thể thực hiện được nhưng bị mất gói đáng kể.  Khác với công việc của họ, cuộc thử nghiệm kéo dài một ngày và họ sử dụng ăng-ten định hướng ở cả máy thu và máy phát. Các tác giả của tài liệu “Tác động của sự thay đổi nhiệt độ đến độ tin cậy của LoRa—Đánh giá thử nghiệm.”, “Khám phá các yếu tố phần cứng và môi trường.” thực hiện các nghiên cứu thực nghiệm về tác động của nhiệt độ đến hiệu suất LoRa. Họ xác nhận sự gián đoạn của liên kết liên lạc khi nhiệt độ ngày càng tăng, khiến việc trao đổi các gói không thể thực hiện được. Họ cũng nghiên cứu xem các cài đặt PHY khác nhau có thể bị ảnh hưởng như thế nào do nhiệt độ dao động. Tuy nhiên, việc lựa chọn chính xác các tham số LoRa PHY có thể chống lại đáng kể các tác động tiêu cực của nhiệt độ cao. Kết quả của họ cũng cho thấy tác động tiêu cực đến hiệu suất LoRa khi có sự kết hợp giữa quãng đường dài và tuyết.  Vì Skellefteå là vùng có khí hậu cận Bắc Cực nên nó được bao phủ bởi tuyết ít nhất 5 tháng mỗi năm. Điều này phải được xem xét khi lập kế hoạch định vị cả cảm biến và cổng, nhằm giảm thiểu tác động tiêu cực của tuyết trong quá trình truyền tín hiệu. Khi nghiên cứu sâu hơn, họ dự định đánh giá tuyết ảnh hưởng như thế nào đến việc truyền tín hiệu khi xem xét các khoảng cách khác nhau giữa máy phát và máy thu. Nói chung, nghiên cứu của chúng tôi chỉ ra rằng người ta phải tính đến nhiệt độ môi trường trong kế hoạch triển khai LoRaWAN, chủ yếu là do ảnh hưởng của nó đến khả năng của pin, điều kiện lan truyền và hành vi tiếng ồn. Do nhiệt độ tác động lên các yếu tố đó nên nó cũng có mối quan hệ chặt chẽ với phạm vi phủ sóng và tốc độ truyền tải dự kiến ​​của LoRaWAN, trở nên quan trọng đối với các khu vực có biên độ nhiệt cao.

**1.2.2. Nghiên cứu ở trong nước**

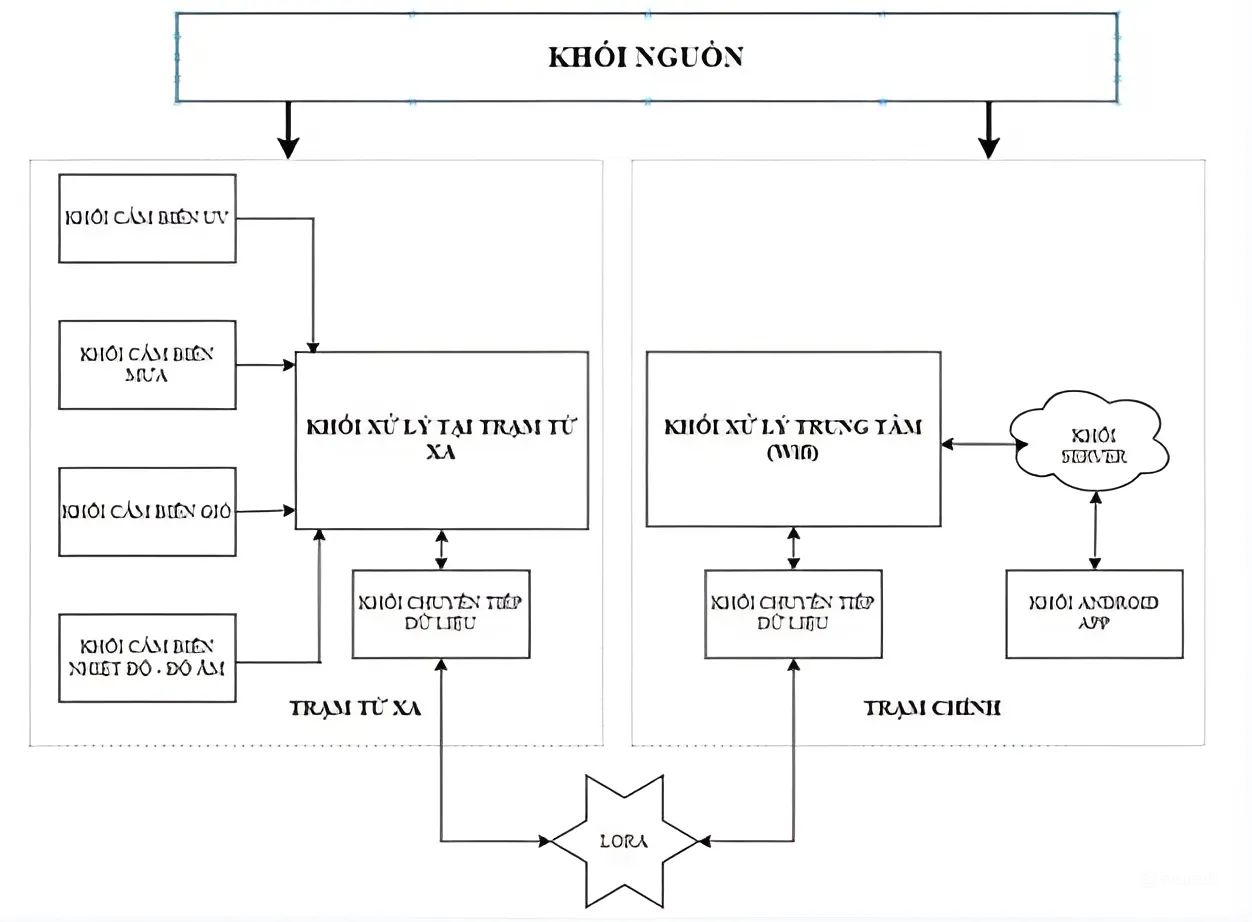
Hiện nay, các hướng nghiên cứu về ứng dụng công nghệ cao trong giám sát từ xa các thông số môi trường trong nông nghiệp thường sử dụng các công nghệ truyền thông truyền thống như: Zigbee, Wifi, GSM/GPRS, Bluetooth… Trong những công nghệ truyền thông kể trên, công nghệ Wifi, Zigbee, Bluetooth có phạm vi hoạt động không cao từ 10 đến 100m. Với phạm vi hoạt động như vậy, các chuẩn truyền thông trên chỉ phù hợp cho việc giám sát trong một khu vực nhỏ. Khi khu vực giám sát lớn hoặc có nhiều khu vực, thì việc xây dựng mạng cảm biến trở lên phức tạp và tốn kém chi phí hơn. Nhằm mở rộng nhu cầu kết nối số lượng lớn cảm biến trên phạm vi rộng và tiêu thụ ít năng lượng phục vụ cho nhu cầu vận hành bảo trì, chuẩn truyền thông LoRa đã ra đời với nhiều ưu điểm vượt trội so với các công nghệ sẵn có. Khoảng cách truyền thông có thể lên tới 2 - 20km và có thể hoạt động trên băng tần không phải cấp phép với tốc độ thấp từ 0,3 kbps đến 50kbps. Thời gian có thể duy trì kết nối và chia sẻ dữ liệu lên đến 10 năm với năng lượng pin. Trong nghiên cứu này, nhóm nghiên cứu đã chế tạo thành công hệ thống mạng cảm biến và điều khiển không dây sử dụng chuẩn truyền thông LORA. Module LoRa RA02 (433MHz) được lựa chọn sử dụng. Kết quả thử nghiệm cho thấy, tốc độ truyền lớn nhất đạt được khi truyền 10 byte dữ liệu là 10,37s đạt được khi cấu hình hệ số trải phổ SF = 7 và độ rộng băng thông BW = 500kHz. Khoảng cách truyền lớn nhất đạt được là 3,5km trong môi trường đô thị với thông số cấu hình SF = 12, BW = 62,5KHz. Thử nghiệm hoạt động của hệ thống tại vườm ươm học viện nông nghiệp cho thấy hệ thống hoạt động ổn định.



Hình 1. 3 Sơ đồ khối hệ thống thu nhập và điều khiển các thông số môi trường

Hình 2.3 thể hiện sơ đồ khối hệ thống thu thập và điều khiển các thông số môi trường được nhóm nghiên cứu xây dựng. Dữ liệu môi trường bao gồm nhiệt độ, độ ẩm không khí, độ ẩm đất, cường độ ánh sáng được thu thập qua nút cảm biến và truyền về thiết bị xử lý trung tâm. Sau đó dữ liệu được truyền lên server qua kết nối Ethernet sử dụng phương thức giao tiếp MQTT. Người dùng có thể theo dõi các thông số môi trường qua ứng dụng được cài đặt trên điện thoại android và ra lệnh điều khiển bật tắt các thiết bị, cơ cấu chấp hành như bơm, đèn, quạt thông gió… Bài báo đã trình bày được thiết kế chế tạo, thử nghiệm thành công hệ thống mạng cảm biến và điều khiển không dây LoRa sử dụng cho bài toán nông nghiệp. Kết quả thử nghiệm cho thấy, trong điều kiện đô thị, khoảng cách đạt được từ 500m - 3,9km tùy theo các thông số cài đặt và khu vực lắp đặt có nhiều hay ít nhà cao tầng che chắn. Kết quả bài báo cho thấy, công nghệ LoRa rất phù hợp cho những bài toán liên quan đến IoT như đô thị thông minh, nông nghiệp thông minh.

Hiện nay biến đổi khí hậu đang là vấn đề nóng trên toàn thế giới. Biểu hiện chính của biến đổi khí hậu là sự nóng lên toàn cầu, cũng như sự thay đổi của các vấn đề thời tiết. Biến đổi khí hậu không chỉ ảnh hưởng đến đời sống, sức khỏe của con người mà còn ảnh hưởng đến hoạt động sản xuất kinh doanh, đặc biệt trong nông nghiệp, du lịch. Do đó, việc giám sát thời tiết là quan trọng. Hệ thống trạm giám sát thời tiết sử dụng vi điều khiển trung tâm và vi điều khiển phụ là 2 board ESP32, module cảm biến mưa, module cảm biến gió, module cảm biến UV ML8511, module cảm biến nhiệt độ – độ ẩm SHT30, module LoRa SX1278 ra đời với mục đích đó. Hiện họ nghiên cứu đã thi công hệ thống trạm giám sát thời tiết đáp ứng được các mục tiêu đưa ra ban đầu của đề tài. Hệ thống hoạt động tương đối ổn định, dễ dàng giám sát qua web, có đầy đủ các chức năng cơ bản của một hệ thống trạm giám sát thời tiết như: giám sát các chỉ số thời tiết tại trạm phụ, truyền không dây trong môi trường qua LoRa với tầm hoạt động tương đối tốt. Giao diện giám sát qua web trực quan, dễ sử dụng. Theo yêu cầu của đề tài đã đưa ra, họ đã thiết kế sơ đồ khối của hệ thống bao gồm các khối: khối xử lý trung tâm, khối xử lý tại trạm từ xa, khối cảm biến tia UV, khối cảm biến mưa, khối cảm biến gió, khối cảm biến nhiệt độ – độ ẩm, khối chuyển tiếp dữ liệu, khối nguồn, khối server, khối android app. Sơ đồ khối của hệ thống được mô tả như hình dưới đây.



**1.2.3. Đánh giá các nghiên cứu đã công bố**

Những nghiên cứu đã công bố có các ưu điểm như sau:

* Phân tích tác động thời tiết đến LoRa, xác nhận ảnh hưởng bức xạ mặt trời và nhiệt độ.
* Đánh giá ảnh hưởng của nhiệt độ và tuyết, đề xuất cách tối ưu hóa LoRa ở khu vực có điều kiện khắc nghiệt.
* Ứng dụng LoRa trong nông nghiệp, thành công trong giám sát môi trường với phạm vi truyền ổn định.
* Hệ thống giám sát thời tiết ổn định, dễ sử dụng, theo dõi các yếu tố thời tiết qua LoRa.

Bên cạnh các ưu điểm đã đạt được, các nghiên cứu vẫn còn tồn tại một số nhược điểm như:

* Cần nghiên cứu thêm về mưa và nhiều liên kết.
* Còn một số hạn chế như chưa thử nghiệm được nhiều đối với môi trường ngoài đô thị có ít nhà cao tầng; chưa đo và khảo sát được công suất phát thực tế của thiết bị.

**1.3. Xác định bài toán cần giải quyết của đề tài**

**1.3.1. Mục tiêu phát triển cho sản phẩm**

- Mở rộng phạm vi giám sát diện rộng.

* Tối ưu hóa tiêu thụ năng lượng.
* Đảm bảo hoạt động ổn định trong thời tiết khắc nghiệt.
* Nâng cao độ chính xác và tin cậy của dữ liệu.
* Hỗ trợ nhiều loại cảm biến môi trường.

**1.3.2. Thông số kỹ thuật đề ra của hệ thống giám sát nhiệt độ, độ ẩm qua LoRa**

Từ mục tiêu nghiên cứu và các ưu nhược điểm của các nghiên cứu đã có, thông số kỹ thuật của thiết bị như bảng dưới đây:

Bảng thông số kỹ thuật của thiết bị:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nội dung | Số lượng | Thông số |
| I | Các thông số vật lý | | |
| 1 | KIT ESP-WROM-32 | 2 |  |
| 2 | Mạch thu phát RF UART LoRa SX1278 | 2 | 433Mhz 3000m |
| 3 | Màn hình LCD TFT | 1 | 1.44 inch giao tiếp SPI |
| 4 | USB TO COM/TTL Chip PL2303HX V3 | 1 |  |
| 5 | Anten RF433 3Dbi SMA Đực | 2 | 5cm loại thẳng |
| 6 | Mạch sạc pin TP4056 có IC bảo vệ | 1 | 1A 5V Li-on Battery |
| 7 | Led báo | 1 | 5mm |
| 8 | Nút nhấn 4 chân | 3 | 6x6x5mm |
| 9 | Còi xung 5V | 1 | 12x9.97mm |
| 10 | DHT11 | 1 |  |
| 11 | IC ổn áp LM 7805 SMD | 1 |  |
| 12 | Adapter 12V 2A | 1 |  |
| 13 | Module thời gian thực RTC DS3231 | 1 | 33x22x14mm |
| II | Chức năng | | |
| 1 | Đo nhiệt độ, độ ẩm |  | Có |
| 2 | Cảnh báo nhiệt độ |  | Có |
| 3 | Điều chỉnh ngưỡng cảnh báo |  | Có |

**1.4. Kết thúc chương 1**

Chương 1 đã trình bày những nghiên cứu chung về hệ thống giám sát nhiệt độ môi trường qua LoRa. Tìm hiểu và đánh giá những nghiên cứu trong và ngoài nước về đối tượng của đề tài. Sau khi có được những kết quả nghiên cứu tổng quan, chương I đã đưa ra mục tiêu phát triển sản phẩm của đề tài và các thông số kỹ thuật, chức năng cụ thể của hệ thống giám sát nhiệt độ môi trường qua LoRa.

Đây là những cơ sở, mục tiêu để nghiên cứu, thiết kế sản phẩm trong chương 2.

**CHƯƠNG II: PHÂN TÍCH HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ, ĐỘ ẨM QUA LORA**

1. **Phân tích, thiết kế tổng quan hệ thống**

Hệ thống giám sát nhiệt độ môi trường cho phép người sử dụng giám sát từ xa nhiệt độ - độ ẩm môi trường cần thiết với nhu cầu được sử dụng, hệ thống giám sát đó có mô hình được miêu tả như hình 2.1

A diagram of a company

Description automatically generated

Hình 2. 1 Sơ đồ khối tổng quan của hệ thống

Hệ thống giám sát môi trường sẽ bao gồm:

* Cấp hiện trường là các bộ : Bộ cảm biến và các bộ thu phát tín hiệu, thu thập dữ liệu và gửi về máy chủ.
* Cấp trung tâm là máy chủ (sever): Nhận dữ liệu từ các bộ ở hiện trường và truyền dữ liệu tới các khối cần thiết để phân tích dữ liệu.

1. **Phân tích, thiết kế mô hình hệ thống giám sát môi trường**
2. **Phân tích thiết kế sơ đồ khối tổng quan của hệ thống giám sát môi trường**

Từ những mục tiêu và thông số kỹ thuật của hệ thống giám sát môi trường đề ra trong mục 1.3, đề tài đề xuất các giải pháp thực hiện các tính năng như sau:

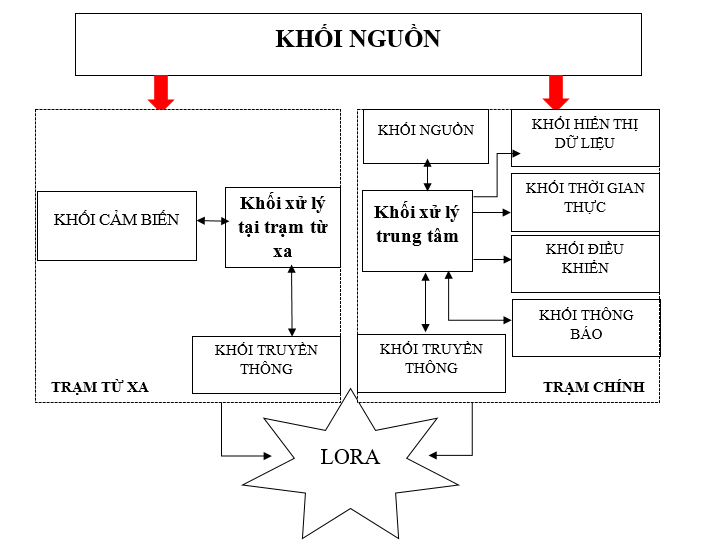
* Chức năng hiển thị thời gian thực: Hệ thống có trang bị module thời gian thực giống như một chiếc đồng hồ liên tục cập nhật ngày giờ cụ thể.
* Chức năng hiển thị dữ liệu: Hệ thống giám sát môi trường có trang bị một màn hình TFT LCD giúp hiển thị các thông số dữ liệu của môi trường như nhiệt độ-độ ẩm và ngày giờ được cung cấp theo thời gian thực.
* Chức năng thông báo khi vượt ngưỡng: Hệ thống sẽ được cài đặt theo yêu cầu của người dùng một mức điều kiện môi trường về nhiệt độ độ ẩm cho phép được gọi là ngưỡng. Hệ thống sẽ báo hiệu về khối xử lý trung tâm để truyền tới thiết bị kết nối và sẽ cho báo hiệu tại nơi giám sát với tiếng còi kêu và đèn LED nháy báo hiệu vượt ngưỡng cho phép hoặc thấp hơn ngưỡng cho phép về điều kiện môi trường.

Từ những giải pháp trên, sơ đồ khối tổng quan của hệ thống được chỉ ra trong hình 2.2.

* **Khối cảm biến:**

**Cảm biến nhiệt độ - độ ẩm:** Sử dụng để đo các thông số về điều kiện môi trường như nhiệt độ - độ ẩm, gửi tín hiệu về khối xử lý trung tâm.

* **Khối thời gian thực:** Sử dụng module thời gian thực DS3231 để cập nhật liên tục dữ liệu về thời gian thực.
* **Khối xử lý tại trạm từ xa:**
* Giao tiếp với khối cảm biến và khối thời gian để lấy dữ liệu .
* Thực hiện nhiệm vụ lọc nhiễu, tính toán và chuẩn bị dữ liệu cho khối chuyển tiếp.
* **Khối chuyển tiếp dữ liệu:** Chuyển tiếp dữ liệu từ khối xử lý tại trạm từ xa đến trạm chính thông qua hệ thống truyền thông LoRa (một giao thức truyền thông không dây tầm xa).
* **Khối hiển thị dữ liệu:** Sử dụng màn hình TFT LCD để hiển thị các dữ liệu thu được.
* **Khối thông báo:** Thông báo qua còi và LED tại trạm từ xa
* **Khối xử lý trung tâm:** Dữ liệu từ khối chuyển tiếp được phân tích và xử lý:
* Dữ liệu thu được từ khối cảm biến và khối thời gian thực sẽ được xử lý và hiển thị trên **Khối hiển thị dữ liệu.**
* Phát tín hiệu thông báo qua còi và LED để người dùng nắm bắt thông tin và có thể đưa ra những biện pháp xử lý phù hợp.
* **LORA:** Là giao thức truyền thông tầm xa dùng để truyền dữ liệu từ trạm từ xa về trạm chính.
* **Khối nguồn:** Có chức năng cung cấp nguồn năng lượng cho hệ thống



Hình 2. 2 Sơ đồ khối một hệ thống giám sát môi trường

1. **Phân tích, lựa chọn linh kiện hệ thống giám sát môi trường**
2. **Khối cảm biến**

* **Cảm biến nhiệt độ độ ẩm DHT11**

Cảm biến nhiệt độ và độ ẩm DHT11 là một trong những loại cảm biến phổ biến và dễ sử dụng trong các dự án đo đạc và theo dõi môi trường. Nó có khả năng đo nhiệt độ trong khoảng từ 0°C đến 50°C với sai số ±2°C và độ ẩm từ 20% đến 90% với sai số ±5%. DHT11 sử dụng giao thức truyền dữ liệu đơn giản với tốc độ chậm, phù hợp cho các ứng dụng nhỏ như hệ thống IoT, nhà thông minh và các dự án DIY. Điểm mạnh của DHT11 là giá thành rẻ, kích thước nhỏ gọn, và khả năng kết hợp dễ dàng với vi điều khiển như Arduino, ESP32 hay Raspberry Pi. Tuy nhiên, do dải đo hạn chế và độ chính xác không cao so với các cảm biến cao cấp khác, DHT11 phù hợp hơn cho những ứng dụng không đòi hỏi độ chính xác tuyệt đối.

**Nguyên lý hoạt động** của cảm biến nhiệt độ độ ẩm DHT11 dựa trên hai thành phần chính: *cảm biến nhiệt độ* và *cảm biến độ ẩm điện dung*. Cụ thể:

**Cảm biến nhiệt độ:**

- DHT11 sử dụng một cảm biến nhiệt điện trở (thermistor) để đo nhiệt độ. Thành phần này thay đổi điện trở theo nhiệt độ môi trường. Từ đó, cảm biến chuyển sự thay đổi điện trở này thành tín hiệu điện tương ứng với nhiệt độ đo được.

**Cảm biến độ ẩm:**

- Độ ẩm được đo bằng cảm biến độ ẩm điện dung. Cảm biến này bao gồm hai tấm điện cực kẹp giữa là một lớp vật liệu hấp thụ hơi nước (polymer). Khi môi trường có sự thay đổi về độ ẩm, lượng nước hấp thụ trong lớp polymer thay đổi, dẫn đến sự thay đổi điện dung giữa hai điện cực. Sự thay đổi điện dung này được chuyển đổi thành tín hiệu điện để tính toán độ ẩm.

**Quá trình xử lý tín hiệu:**

- DHT11 có một mạch xử lý bên trong, giúp chuyển đổi tín hiệu từ cảm biến nhiệt độ và độ ẩm thành tín hiệu số. Sau khi đo đạc, dữ liệu sẽ được gửi đến bộ vi điều khiển thông qua một giao thức giao tiếp 1 dây(single-wire communication). Tín hiệu này bao gồm một chuỗi các bit đại diện cho nhiệt độ và độ ẩm.

**Chu trình đo:**

- Cảm biến DHT11 đo nhiệt độ và độ ẩm trong chu kỳ khoảng 2 giây một lần, và dữ liệu được truyền dưới dạng gói dữ liệu 40 bit: bao gồm 16 bit cho độ ẩm, 16 bit cho nhiệt độ và 8 bit checksum để kiểm tra lỗi.

-Với nguyên lý hoạt động này, DHT11 cung cấp dữ liệu môi trường ở mức cơ bản, đủ cho nhiều ứng dụng phổ thông nhưng có giới hạn về độ chính xác và dải đo.

**THÔNG SỐ KỸ THUẬT**

* Điện áp hoạt động: 5VDC
* Chuẩn giao tiếp: TTL, 1 wire.
* Khoảng đo độ ẩm: 20%-80%RH sai số ± 5%RH
* Khoảng đo nhiệt độ: 0-50°C sai số ± 2°C
* Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
* Kích thước: 28mm x 12mm x10m

Hình 2. 3 Cảm biến nhiệt độ - độ ẩm DHT11

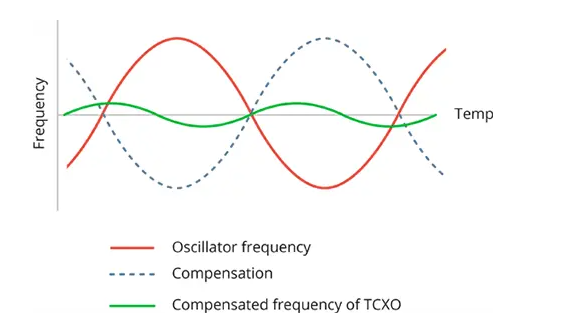
1. **Khối thời gian thực**

* **Module thời gian thực DS3231**

DS3231 là một module thời gian thực (RTC) có độ chính xác cao, sử dụng giao thức giao tiếp I2C. Được tích hợp với thạch anh dao động và cảm biến nhiệt độ, DS3231 duy trì thời gian chính xác ngay cả khi mất nguồn điện nhờ pin dự phòng. Nó có khả năng hiển thị giờ, phút, giây, cùng với ngày, tháng, năm và tự động điều chỉnh năm nhuận. DS3231 thường được sử dụng trong các dự án yêu cầu theo dõi thời gian dài hạn như đồng hồ, thiết bị đo lường và hệ thống nhúng.

**Nguyên lý hoạt động** của module thời gian thực DS3231 dựa trên việc duy trì thời gian chính xác bằng cách sử dụng một thạch anh dao động tích hợp và cảm biến nhiệt độ bên trong để tự động bù trừ nhiệt độ, giúp giảm sai số về thời gian.

* **Thạch anh dao động tích hợp (TCXO)**

DS3231 có thạch anh dao động được tích hợp sẵn trong IC, giúp tránh sai số do các yếu tố bên ngoài như môi trường hoặc tác động vật lý. Thạch anh này duy trì xung nhịp chính xác để tính toán thời gian.

Hình 2. 4 Sơ đồ dao động của thạch anh

Chú thích:

-Đường màu đỏ: Dao động của tần số của thạch anh (Oscillator frequency)

-Đường màu xanh đứt nét: Biểu đồ bù nhiệt (Compenstation) điều chỉnh sai lệch tần số do nhiệt độ

-Đường màu xanh lá cây:Tần số đã được bù nhiệt của dao động (Compensated frequency of TCXO)

Sơ đồ này mô tả quá trình bù nhiệt cho dao động thạch anh nhằm duy trì độ chính xác cao hơn trong điều kiện nhiệt độ thay đổi. TCXO (Temperature Compensated Crystal Oscillator) là một dạng dao động thạch anh có khả năng tự điều chỉnh để bù sai lệch do nhiệt độ, giúp duy trì tần số ổn định trong nhiều điều kiện môi trường khác nhau.

* **Cảm biến nhiệt độ bù nhiệt**

DS3231 đi kèm với một cảm biến nhiệt độ để đo và bù trừ những thay đổi nhiệt độ, vì nhiệt độ có thể ảnh hưởng đến tần số dao động của thạch anh. Nhờ vào bù nhiệt này, DS3231 duy trì thời gian với độ chính xác rất cao (sai lệch chỉ khoảng ±2 ppm, tương đương với khoảng 1 phút sai lệch sau 1 năm).

* **Giao tiếp I2C**

DS3231 sử dụng giao thức I2C để giao tiếp với vi điều khiển. Thông qua giao thức này, vi điều khiển có thể đọc và ghi thời gian (giờ, phút, giây, ngày, tháng, năm) vào DS3231. Vi điều khiển cũng có thể cấu hình báo thức hoặc các sự kiện thời gian cụ thể.

* **Pin dự phòng:**

DS3231 có chân cắm cho pin dự phòng (thường là pin lithium CR2032). Khi hệ thống chính mất điện, pin này cung cấp nguồn cho DS3231 để tiếp tục duy trì thời gian mà không bị gián đoạn. Khi có nguồn trở lại, nó sẽ đồng bộ hóa thời gian với hệ thống mà không cần cài đặt lại.

**THÔNG SỐ KỸ THUẬT MODULE THỜI GIAN THỰC RTC DS3231**

Size: dài 38mm, rộng 22mm, cao 14mm

Khối lượng 8g

Điện thế hoạt động 3.3 – 5.5V

Clock: high-precision clock on chip DS3231

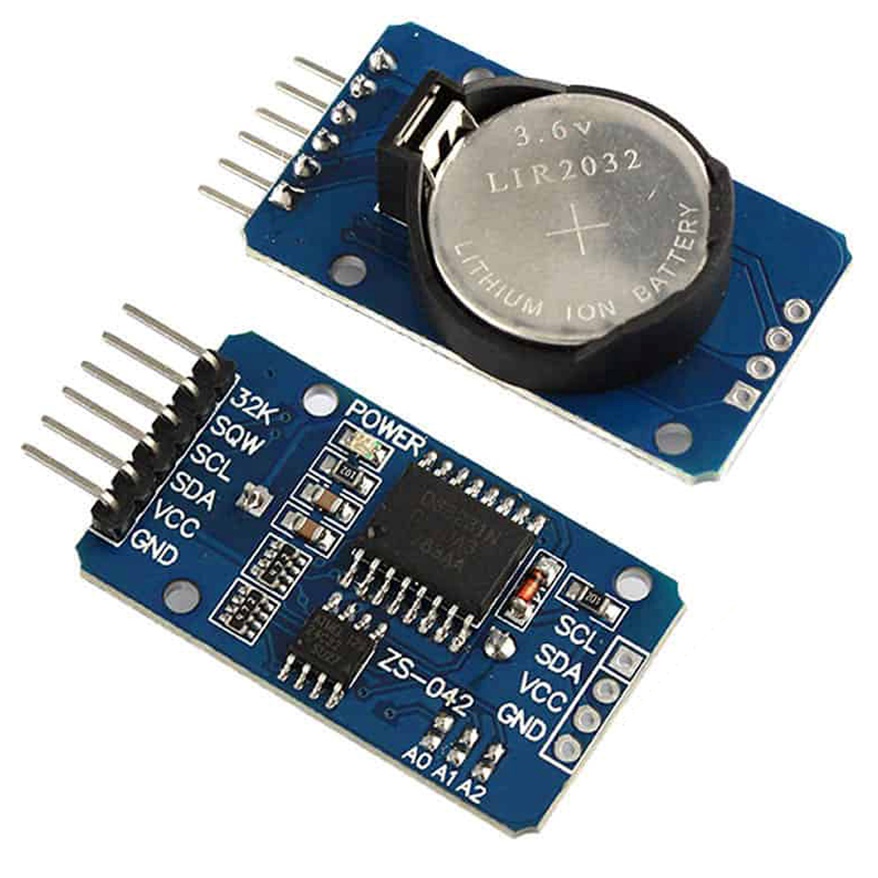
Clock Accuracy: 040 ℃ range, the accuracy 2ppm, the error was about 1 minute

Thông tin Thời gian: giờ, phút, giây, ngày, thứ, tháng, năm, đến 2100.

Cảm biến nhiệt trên IC có độ chính xác ± 3 ℃

I2C bus có tốc độ tối đa 400Khz

Kèm thêm pin sạc được CR2032

Kèm thêm memory IC AT24C32 (32k bits)

Hình 2. 5 Module thời gian thực RTC DS3231

1. **Khối hiển thị dữ liệu**

* **Màn hình TFT LCD 1.44 Inch giao tiếp SPI**

TFT LC1.4 inch giao tiếp SPI là màn hình nhỏ gọn, sử dụng công nghệ cho độ tương phản cao và màu sắc sống động. Giao tiếp SPI giúp kết nối dễ dàng với vi điều khiển như Arduino, ESP32 hoặc Raspberry Pi. Với kích thước 1.4 inch, màn hình này lý tưởng cho các ứng dụng nhỏ như đồng hồ thông minh, thiết bị IoT, và các dự án nhúng yêu cầu hiển thị sắc nét, tiết kiệm điện năng.

**Nguyên lý hoạt động** chi tiết của màn hình TFT LCD 1.4 inch giao tiếp SPI có thể được chia làm ba phần chính: hoạt động của LCD, cơ chế điều khiển TFT, và giao thức truyền thông SPI.

* + **TFT LCD (Thin Film Transistor Liquid Crystal Display)**

**Cấu tạo:**

* Bao gồm nhiều lớp: lớp tinh thể lỏng nằm giữa hai lớp kính mỏng và các lớp lọc màu (RGB) để tái tạo màu sắc.
* Phía sau lớp tinh thể lỏng là đèn nền (thường là LED) để cung cấp ánh sáng cho toàn bộ màn hình.
* Mỗi điểm ảnh (pixel) có một bóng bán dẫn màng mỏng (TFT) giúp điều khiển lượng ánh sáng đi qua.
* **TFT (Thin Film Transistor) - Bộ điều khiển ma trận**
* **Cấu tạo ma trận:** TFT là một công nghệ sử dụng các transistor phim mỏng để điều khiển từng pixel riêng biệt trên màn hình. Đây là một hệ thống điều khiển ma trận, trong đó mỗi pixel LCD được liên kết với một transistor điều khiển.
* **Hoạt động của ma trận TFT:** TFT hoạt động như một công tắc cho từng pixel. Mỗi pixel có một transistor riêng, và khi điện áp được cung cấp cho transistor đó, nó mở ra và cho phép dòng điện chạy qua pixel, khiến nó phát sáng. Các transistor điều khiển lượng dòng điện đến từng pixel, từ đó quyết định độ sáng và màu sắc của pixel.
* **Làm tươi màn hình:** Do các điểm ảnh được điều khiển độc lập, màn hình TFT LCD có thể làm tươi (refresh) nội dung rất nhanh, thích hợp cho hiển thị các nội dung động và video mà không có hiện tượng bóng mờ.

**→ Tóm tắt:** TFT đảm bảo rằng mỗi pixel được kích hoạt một cách chính xác và độc lập, từ đó tối ưu hiệu suất hiển thị của màn hình.

* **Giao thức SPI (Serial Peripheral Interface)**
* **SPI là giao thức truyền thông nối tiếp:** SPI hoạt động bằng cách truyền dữ liệu từng bit một, đồng bộ theo xung nhịp (clock). Nó sử dụng 4 chân cơ bản: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Clock) và CS (Chip Select). Trong trường hợp này, màn hình TFT LCD là slave và vi điều khiển (như Arduino hoặc Raspberry Pi) là master.
* **Truyền dữ liệu:** Vi điều khiển gửi các lệnh và dữ liệu đến màn hình thông qua chân MOSI và SCK. Dữ liệu này bao gồm các thông tin về hình ảnh hoặc các tín hiệu điều khiển pixel (chẳng hạn màu sắc, độ sáng).
* **MOSI:** Truyền dữ liệu từ vi điều khiển (Master) sang màn hình (Slave).
* **SCK (Clock):** Điều khiển tốc độ truyền dữ liệu.
* **CS (Chip Select):** Kích hoạt màn hình (Slave) khi cần giao tiếp, giúp hệ thống có thể kết nối nhiều thiết bị qua giao thức SPI.
* **Dữ liệu và hình ảnh:** Sau khi nhận dữ liệu, màn hình TFT LCD sử dụng mạch điều khiển để giải mã các lệnh và hiển thị hình ảnh tương ứng trên từng pixel.

**→ Tóm tắt:** SPI là giao thức truyền thông giúp vi điều khiển gửi dữ liệu và lệnh điều khiển đến màn hình TFT LCD, từ đó màn hình sẽ hiển thị nội dung theo yêu cầu.

**THÔNG SỐ KỸ THUẬT:**

* Kích thước: 1.44 inch
* Độ phân giải: 128x128
* Đèn nền: màu trắng - 2 đèn led
* Giao thức: SPI
* Màu sắc: 65K
* Điện áp hoạt động của màn hình TFT: 3.3 VDC

Hình 2. 6 Màn hình TFT LCD 1.44 inch giao tiếp SPI

**Khối điều khiển**

Khối điều khiển cho phép người dùng tại trạm thu có thể điều chỉnh các giá trị ngưỡng của môi trường và đưa ra tín hiệu cảnh báo khi các giá trị của môi trường vượt qua giá trị ngưỡng đã cài đặt.

* **Vị trí và nhiệm vụ khối điều khiển**

**Vị trí:** Khối điều khiển nằm trong trạm chính, giữa khối xử lý trung tâm và khối thông báo. Khối điều khiển chuyển dữ liệu đã cài đặt cho khối trung tâm, nhận dữ liệu của khối cảm biến và gửi tín hiệu về lại khối trung tâm.

**Nhiệm vụ:**

* Nút "Tăng" (S1): Khi nhấn nút này, dòng điện từ nguồn +3.3V sẽ đi qua điện trở R1 (220Ω) và chuyển trạng thái mạch từ mức cao xuống mức thấp (nối với GND). Nút này thường dùng để tăng giá trị hoặc di chuyển lên trong các thao tác điều khiển.
* Nút "Giảm" (S2): Tương tự, khi nhấn nút này, dòng điện đi qua điện trở R2 (220Ω) và cũng chuyển trạng thái mạch từ mức cao xuống mức thấp. Nút này có nhiệm vụ giảm giá trị hoặc di chuyển xuống trong menu hoặc tùy chọn điều khiển.
* Nút "Enter" (S3): Khi nhấn nút này, dòng điện đi qua điện trở R3 (220Ω) và trạng thái mạch thay đổi giống như hai nút trên. Nút này thường được sử dụng để xác nhận hoặc chọn một giá trị nào đó.

**Phương thức hoạt động:**

Khối điều khiển gồm 3 nút nhấn: Tăng, Giảm và Enter được nối với các điện trở R1, R2, R3 tương ứng và tiêu thụ điện áp 3,3V của ESP32 tại khối xử lý trung tâm. Khối điều khiển sẽ là công cụ giúp người dùng đặt giá trị ngưỡng cho hệ thống và gửi tín hiệu khi giá trị môi trường vượt qua giá trị ngưỡng đã cài đặt trước đó.

Khi không có nút nào được nhấn, các điểm kết nối giữa các nút nhấn và điện trở đều ở mức điện áp cao (+3.3V). Khi một trong các nút được nhấn, mạch sẽ kết nối điểm đó trực tiếp với GND (mức điện áp thấp), làm thay đổi trạng thái từ cao xuống thấp. Đây là cách hệ thống nhận biết được khi nào nút nhấn được kích hoạt. Khi nhấn nút Tăng (S1), dòng điện từ +3.3V đi qua điện trở R1 và kết nối trực tiếp với GND, tạo ra tín hiệu mức thấp (0V). Tín hiệu này được vi điều khiển hoặc mạch chính nhận diện để thực hiện chức năng tăng giá trị hoặc di chuyển lên trong menu. Tương tự, khi nút Giảm (S2) được nhấn, dòng điện đi qua điện trở R2 và tiếp đất, tạo ra tín hiệu mức thấp, giúp giảm giá trị hoặc di chuyển xuống. Khi nhấn nút Enter (S3), dòng điện đi qua điện trở R3 và nối xuống GND, tạo ra tín hiệu mức thấp, cho phép xác nhận hoặc chọn một giá trị. Các điện trở 220Ω được sử dụng nhằm hạn chế dòng điện khi nút được nhấn, bảo vệ các thành phần trong mạch khỏi dòng điện quá cao và đảm bảo rằng tín hiệu được duy trì ổn định, không bị nhiễu. Tóm lại, nguyên lý hoạt động của khối điều khiển là dựa trên sự thay đổi mức điện áp khi nhấn các nút để gửi tín hiệu tới mạch điều khiển chính, cho phép thực hiện các thao tác điều hướng và lựa chọn.

1. **Khối thông báo**

Khối thông báo chịu trách nhiệm nhận dữ liệu từ khối xử lý trung tâm để phát hiện và gửi cảnh báo khi có điều kiện bất thường xảy ra

* **Vị trí và nhiệm vụ của khối thông báo**

**Vị trí:** Khối thông báo nằm trong trạm chính, giữa khối xử lý trung tâm và khối hiển thị dữ liệu. Nó nhận dữ liệu từ khối xử lý trung tâm, sau đó thực hiện việc kiểm tra, phân tích các điều kiện và đưa ra cảnh báo.

**Nhiệm vụ:**

* Nhận tín hiệu dữ liệu từ khối xử lý trung tâm, bao gồm các giá trị từ các cảm biến (nhiệt độ, độ ẩm, thời gian thực, dữ liệu thời tiết).
* Kiểm tra ngưỡng: Khối thông báo sẽ so sánh dữ liệu nhận được với các ngưỡng an toàn đã được cài đặt sẵn (ví dụ: nhiệt độ vượt quá mức 40°C).
* Tạo tín hiệu cảnh báo: Khi phát hiện điều kiện vượt ngưỡng, khối này sẽ tạo cảnh báo (âm thanh, đèn nháy).

**Phương thức hoạt động**

Khối thông báo đóng vai trò quan trọng trong hệ thống, giúp phát hiện kịp thời các điều kiện bất thường (như nhiệt độ, độ ẩm hoặc dữ liệu thời tiết vượt ngưỡng), đồng thời gửi cảnh báo về trạm chính để người dùng có thể can thiệp hoặc xử lý kịp thời.

Khối thông báo nhận dữ liệu từ khối xử lý trung tâm trong trạm chính. Dữ liệu này bao gồm các dữ liệu từ khối cảm biến và khối hiển thị truyền tới khối xử lý tại trạm từ xa, thông qua khối chuyển tiếp dữ liệu sẽ được phân tích, tính toán và xử lý để đưa tới khối trung tâm. Từ đó các dữ liệu đã được phân tích và xử lý sẽ được gửi tới khối thông báo. Khối thông báo lúc này sẽ có nhiệm vụ kiểm tra các giá trị nhận được so với các giá trị ngưỡng được quy định sẵn. Khi một điều kiện bất thường được phát hiện sẽ là một chuỗi các hành động của khối thông báo

**Kích hoạt cảnh báo:** Kích hoạt còi báo động hoặc đèn LED để cảnh báo ngay tại chỗ.

**Tích hợp và phản hồi**

* **Tích hợp với các khối khác:** Khối thông báo làm việc đồng bộ với các khối khác trong hệ thống (cảm biến, khối xử lý trung tâm, và khối hiển thị dữ liệu) để tạo ra một hệ thống giám sát và cảnh báo an toàn hiệu quả.
* **Phản hồi từ người dùng:** Sau khi nhận cảnh báo, người dùng có thể phản hồi qua ứng dụng, chẳng hạn như xác nhận đã nhận thông báo hoặc thực hiện hành động thích hợp.

1. **Khối chuyển tiếp dữ liệu (LORA)**

LORA (Long Range Radio) là một công nghệ truyền thông không dây, sử dụng tần số vô tuyến để truyền tải dữ liệu với khoảng cách xa (từ 2 km đến 15 km) và tiêu thụ năng lượng rất thấp. LoRa thường được sử dụng trong các ứng dụng IoT (Internet of Things), nơi cần thu thập và truyền tải dữ liệu từ các cảm biến đến trung tâm xử lý. Thông thường sẽ sử dụng 2 module LORA để truyền và nhận dữ liệu.

**Nguyên lý hoạt động** Kỹ thuật điều chế Chirp Spread Spectrum là nền tảng để phát triển công nghệ LoRa. Đầu tiên, tín hiệu dữ liệu gốc sẽ được khuếch đại tần số cao hơn, sau đó nó được mã hóa thành chuỗi tín hiệu chirp, cuối cùng thì sẽ được gửi đi từ anten. Nhờ có nguyên lý hoạt động như trên, tín hiệu LoRa có thể được truyền đi xa với lượng công suất thấp và cả tín hiệu không mạnh bằng tín hiệu nhiễu bên ngoài môi trường. Chirp Spread Spectrum (CSS) là một kỹ thuật sử dụng các xung chirp (chuỗi tín hiệu về tần số) để mã hóa các thông tin. Cách thức hoạt động của LORA trong khối chuyển tiếp dữ liệu

* **Thu thập dữ liệu từ cảm biến**
* **Ghi nhận thông tin:** Các cảm biến sẽ đo đạc các thông số môi trường như nhiệt độ và độ ẩm và chuyển đổi thành tín hiệu điện.
* **Xử lý dữ liệu:** Vi điều khiển sẽ đọc dữ liệu từ cảm biến, có thể thực hiện các phép tính như trung bình hoặc giá trị tối đa để giảm thiểu độ nhiễu.
* **Chuyển đổi và truyền dữ liệu**
* **Mã hóa dữ liệu:** Dữ liệu thu được sẽ được mã hóa trước khi gửi đi để bảo mật.
* **Truyền qua module LoRa**: Vi điều khiển gửi dữ liệu đã mã hóa đến module LoRa thông qua giao tiếp SPI (Serial Peripheral Interface).
* **Gửi đi:** Module LoRa sẽ phát sóng dữ liệu qua tần số đã chọn (ví dụ 868MHz hoặc 915MHz) đến thiết bị nhận.
* **Nhận dữ liệu tại khối xử lý trung tâm**
* **Thiết bị nhận:** Một module LoRa tương ứng tại trạm trung tâm sẽ nhận dữ liệu từ thiết bị gửi.
* **Giải mã:** Dữ liệu nhận được sẽ được giải mã để phục vụ cho việc xử lý tiếp theo.
* **Xử lý và lưu trữ dữ liệu**
* **Xử lý dữ liệu:** Dữ liệu giải mã sẽ được phân tích tại trung tâm. Nếu có điều kiện bất thường, hệ thống sẽ đưa ra các cảnh báo hoặc thông báo.
* **Lưu trữ:** Dữ liệu có thể được lưu trữ trên một máy chủ hoặc cơ sở dữ liệu (có thể sử dụng đám mây hoặc local server) để phục vụ cho việc phân tích sau này.
* **Chức năng của khối chuyển tiếp dữ liệu bằng LORA**
* **Truyền tải dữ liệu**

Chế độ truyền: Khối xử lý dữ liệu có thể cấu hình để truyền dữ liệu theo chế độ định kỳ hoặc theo yêu cầu (on-demand).

Khoảng cách truyền: LoRa có khả năng truyền dữ liệu trong khoảng cách lên đến 15 km trong điều kiện lý tưởng, lý tưởng cho các ứng dụng nông nghiệp, môi trường, và đô thị.

* **Bảo mật và đáng tin cậy**

Mã hóa dữ liệu: Dữ liệu truyền qua LoRa thường được mã hóa để bảo vệ thông tin, sử dụng các thuật toán như AES (Advanced Encryption Standard).

Xác thực thiết bị: Mỗi thiết bị đều có token hoặc khóa riêng để xác thực, đảm bảo rằng chỉ những thiết bị được phép mới có thể gửi hoặc nhận dữ liệu.

**THÔNG SỐ KỸ THUẬT**

* Hãng sản xuất : EBYTE
* IC chính: SX1278 của SEMTECH.
* Điện áp hoạt đông: 2.3 – 5.5 VDC
* Tốc độ truyền: 0.3 – 19.2 Kbps ( mặc định 2.4 Kbps)
* Điện áp giao tiếp: TTL-3.3V
* Giao tiếp UART Data bits 8, Stop bits 1, Parity none, tốc độ từ 1200 – 115200
* Tần số: 410 – 441Mhz
* Dung lượng bộ nhớ đệm: 512bytes
* Hỗ trợ 65536 địa chỉ cấu hình
* Nhiệt độ hoạt động: -40 ~ 85° C
* Độ ẩm làm việc: 10 ~ 90%
* Model: E32-433T20D Lora SX1278 433Mhz (TTL-100)

Công suất: 20dbm (100mW)

Khoảng cách truyền tối đa trong điều kiện lý tưởng: 3000m

Độ nhạy: -130dBm

Dòng điện làm việc: 110mA

Kích thước: 21 x 36mm

Trọng lượng: 8g

* Model: E32 433T30D Lora SX1278 433Mhz

Công suất: 30dbm (1W)

Khoảng cách truyền tối đa trong điều kiện lý tưởng: 8000m

Độ nhạy: -147dBm

Dòng điện làm việc: 610mA

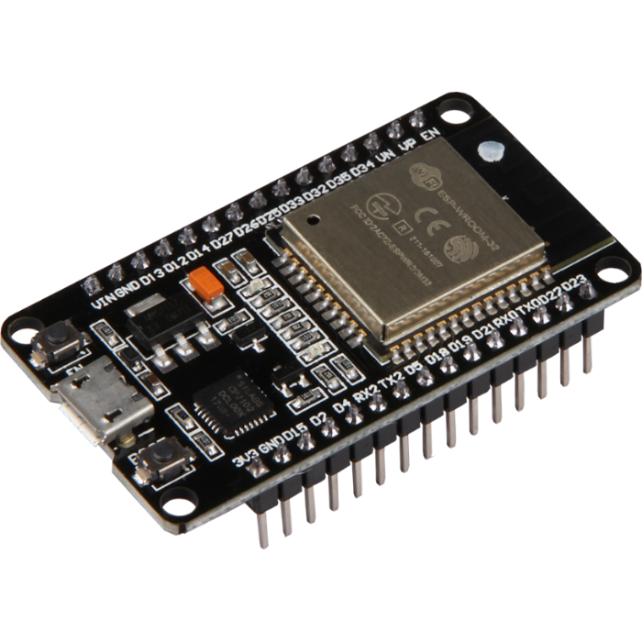
Kích thước: 24 x 43mm

Trọng lượng: 10g

Hình 2. 7 Module LORA

1. **Khối xử lý trung tâm và khối xử lý trạm từ xa**

Với nhu cầu điều khiển hệ thống thực hiện các chức năng đã đề ra trong mục 1.3.2, đề tài lựa chọn ESP32 DEVKIT V1 làm khối xử lý trung tâm. ESP32 DEVKIT V1 tích hợp sẵn wifi; Hiệu năng cao, chip xử lý mạnh; Dễ dàng lập trình

ESP32 DEVKIT V1 được phát triển trên nền module trung tâm là ESP32 với công nghệ Wifi, BLE, nhân ARM SoC tích hợp mới nhất hiện nay và có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code.

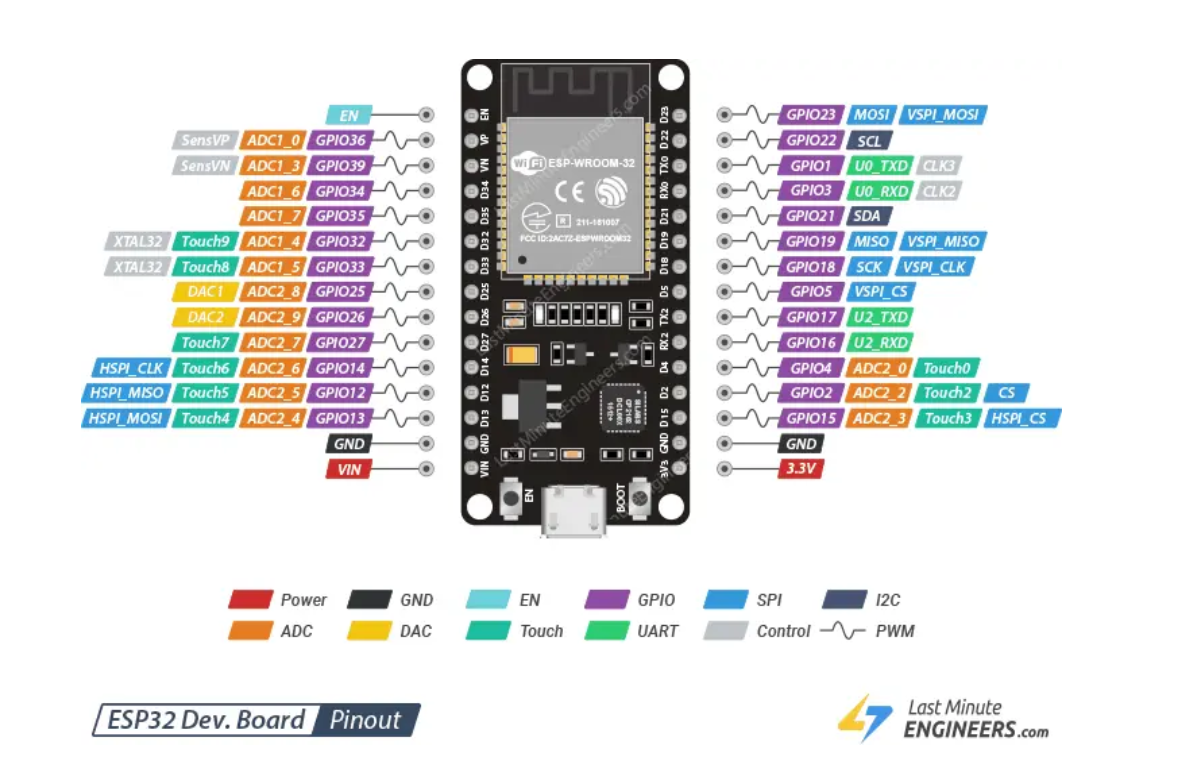
Hình 2. 8 ESP32 DEVKIT V1

ESP32 DEVKIT V1 bao gồm chip ESP32 là bộ xử lý trung tâm, tích hợp bộ thu phát Wifi, có bộ nhớ Flash 4MB. Có tổng cộng 13 chân GPIO (D0-D8, RX, TX, SD2, SD3). Với chức năng là các ngõ vào và ngõ ra sử dụng các hàm pinMode(), degitalWrite() và degitalRead() để điều khiển. Các chân (trừ D0) để thực hiện các chuẩn giao tiếp I2C, PWM. ESP32 DEVKIT V1 sử dụng cáp USB để giao tiếp với máy tính. Thông qua cổng này người lập trình có thể nạp chương trình cho ESP32 DEVKIT V1, ngoài ra cổng USB là nguồn cung cấp điện áp 5V cho ESP32 DEVKIT V1 hoạt động.

ESP32 là một mạch vi điều khiển giúp chúng ta có thể điều khiển các thiết bị ngoại vi khác. Bên cạnh đó ESP32 là sự kết hợp giữa module Wifi tích hợp sẵn bên trong vi điều khiển chính sử dụng chip ESP32 SoC (System on Chip) được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển thông qua Internet.



Hình 2. 9 Chip ESP32

**ESP32 DEVKIT V1 có thể quét và kết nối đến một mạng Wifi bất kỳ (Wifi Client) để thực hiện các tác vụ như lưu trữ, truy cập dữ liệu từ Server, tạo điểm truy cập Wifi (Wifi Access point) cho phép các thiết bị khác kết nối, giao tiếp và điều khiển đồng thời nó cũng là một Server để xử lý dữ liệu từ các thiết bị sử dụng Internet khác. Với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CP2102.

Hình 2. 10 Sơ đồ chân ESP32 DEVKIT V1

Bảng 1: Chức năng các chân ESP32

| **Thứ tự** | **Tên tín hiệu** | **I/O** | **Mô tả** |
| --- | --- | --- | --- |
| 1 | VDDA | Nguồn | Điện áp tương tự từ 2.5 V– 3.6V |
| 2 | LNA | I/O | Cổng giao tiếp RF. |
| 3 | VDD3P3 | Nguồn | Điện áp khuếch đại 2.5 V– 3.6V |
| 4 | VDD3P3 | Nguồn | Điện áp khuếch đại 2.5 V– 3.6V |
| 5 | VDD\_RTC | Nguồn | NC (1.1V) |
| 6 | TOUT | I | Chân ADC. Chân này được sử dụng để kiểm tra điện áp cung cấp điện của VDD3P3 và điện áp đầu vào của TOUT. |
| 7 | CHIP\_EN | I | Chân cho phép chip  Mức cao: bật, chip hoạt động bình thường.  Mức thấp: tắt, dòng điện nhỏ tiêu thụ |
| 8 | XPD\_DCDC | XPD\_DCDC | GPIO16 |
| 9 | MTDI | I/O | GPIO16; HSPI\_CLK |
| 10 | VDDPST |  | GPIO12; HSPI\_MISO |
| 11 | VDDPST | Nguồn | Điện áp cũng cấp IO và điện áp số (1.8V–3.6V) |
| 12 | MTCK | I/O | GPIO13; HSPI\_MOSI; UART0\_CTS |
| 13 | MTDO | I/O | GPIO15; HSPI\_CS; UART0\_RTS |
| 14 | GPIO2 | I/O | UART Tx trong quá trình lập trình Flash; GPIO2 |
| 15 | GPIO0 | I/O | GPIO0; SPI\_CS2 |
| 16 | GPIO4 | I/O | GPIO4 |
| 17 | VDDPST | Nguồn | Điện áp cũng cấp IO và điện áp số (1.8V–3.6V) |
| 18 | SDIO\_DAT  A\_2 | I/O | Kết nối với chân SD\_D2; SPIHD; HSPIHD;  GPIO9 |
| 19 | SDIO\_DAT  A\_3 | I/O | Kết nối với chân SD\_D3; HSPIWP; GPIO10 |
| 20 | SDIO\_CMD | I/O | Kết nối với chân SD\_CMD; SPI\_CS0; GPIO11 |
| 21 | SDIO\_CLK | I/O | Kết nối với chân SD\_CLK; SPI\_CLK; GPIO6 |
| 22 | SDIO\_DAT  A\_0 | I/O | Kết nối với chân SD\_D0; SPI\_MISO; GPIO7 |
| 23 | SDIO\_DAT  A\_1 | I/O | Kết nối với chân SD\_D1; SPI\_MOSI; GPIO8 |
| 24 | GPIO5 | I/O | GPIO5 |
| 25 | U0RXD | I/O | UART Rx trong quá trình Flash; GPIO3 |
| 26 | U0TXD | I/O | UART Tx trong quá trình Flash; GPIO1;  SPI\_CS1 |
| 27 | XTAL\_OUT | I/O | Kết nối với đầu ra thạch anh, có thể sử dụng để cung cấp BT đầu vào xung clock. |
| 28 | XTAL\_IN | I/O | Kết nối với ngõ vào thạch anh |
| 29 | VDDD | Nguồn | Điện áp tương tự 2.5V – 3.6V |
| 30 | VDDA | Nguồn | Điện áp tương tự 2.5V – 3.6V |
| 31 | RES12K | I | Kết nối với điện trở 12 KΩ và kết nối với đất |
| 32 | EXT\_RSTB | I | Tín hiệu đặt lại bên ngoài (hoạt động ở mức thấp) |

**THÔNG SỐ KĨ THUẬT**

* Kích thước: 18 mm x 20 mm x 3 mm
* CPU: Xtensa Dual-Core 32-bit LX6 với tần số hoạt động lên đến 240 MHz
* Bộ nhớ trong:
* 448 KBytes ROM cho booting và các tính năng của lõi chip.
* 520 KBytes SRAM trên chip dùng cho dữ liệu và các lệnh instruction.
* 8 KBytes SRAM trong RTC (gọi là RTC SLOW Memory) để truy xuất bởi các bộ co-processor
* 8 KBytes SRAM trong RTC (gọi là RTC FAST Memory) dùng cho lữu dữ liệu, truy xuất bởi CPU khi RTC đang boot từ chế độ Deep-sleep.
* 1 Kbit EFUSE, với 256 bit cho hệ thống (địa chỉ MAC và cấu hình chip), 768 còn lại cho ứng dụng người dùng, gồm cả mã hóa bộ nhớ Flash và định ID cho chip.
* Kết nối WiFi:
* Wi-Fi: 802.11 b/g/n/e/i
* Bluetooth: BR/EDR phiên bản v4.2 và BLE
* Ethernet MAC hỗ trợ chuẩn: DMA và IEEE 1588
* Bus hỗ trợ mang CAN 2.0
* Giao tiếp ngoại vi:
* Bộ chuyển đổi ADC 12 bit, 16 kênh
* Bộ chuyển đổi 8-bits DAC: 2 kênh
* 10 chân để giao tiếp với cảm biến chạm (touch sensor)
* IR (TX/RX)
* Ngõ ra PWM cho điều khiển Motor
* LED PWM: 16 kênh
* Cảm biến Hall
* Cảm biến nhiệt độ
* 4 X SPI
* 2 X I²S
* 2 X I²C
* 3 X UART
* Nhiệt độ hoat động ổn định: -40 o*C đến 85* oC
* Điện áp hoạt động: 2.2 - 3.6V
* Dòng tiêu thụ ổn định: 80mA

1. **Khối nguồn**

Khối nguồn này sử dụng nguồn điện ngoài thông qua một adapter để chuyển đổi từ nguồn xoay chiều (AC) thành nguồn một chiều (DC) 12V. Adapter giúp giảm và chỉnh lưu điện áp từ lưới điện (thường là 220V AC) xuống 12V DC, sau đó cung cấp cho các khối khác trong mạch. Các mức điện áp sử dụng trong mạch biểu diễn trong bảng:

Bảng 2: Các mức điện áp sử dụng mạch

|  |  |  |
| --- | --- | --- |
| **STT** | **Khối** | **Mức điện áp** |
| 1 | Khối nguồn | Đầu vào adapter: 12VDC  Đầu ra mạch: 5VDC |
| 2 | Khối hiển thị OLED TFT | 5VDC |
| 3 | Khối cảnh báo còi + LED | 3.3VDC |
| 4 | Khối thời gian thực DS3231 | 5VDC |
| 5 | Khối cảm biến DHT11 | 5VDC |
| 6 | Khối truyền thông LORA |  |

Pin 18650 về bản chất là loại pin Lithium-ion (Li-Ion) , chúng ta có thể bắt gặp pin Lithium-ion trong nhiều hình dạng kích thước khác nhau, chúng là bước đột phá cho các sản phẩm cầm tay như máy ảnh, máy cắt, máy khoan cầm tay, pin điện thoại, đèn pin, quạt cầm tay,…

Hình 2. 11 Pin 18650

**THÔNG SỐ KỸ THUẬT**

• Dung lượng danh nghĩa: 1500 mAh.

• Kích thước sản phẩm: 18 x 65 mm.

• Điện áp định mức: 3.7V.

• Điện áp sạc: 4.2V.

Ngoài việc sử dụng pin 18650, khối nguồn còn chứa một mạch sạc pin 1sell 18650 có bảo vệ TP4056 1A với cổng nối micro USB. Mạch sạc pin TP4056 là một module sạc được thiết kế để sạc các loại pin lithium-ion, đặc biệt là pin 18650, với điện áp danh định là 3.7V. Mạch này thường có cổng micro USB để kết nối với nguồn sạc (5V) và có thể cung cấp dòng sạc lên tới 1A. Ngoài ra, mạch TP4056 còn tích hợp các chức năng bảo vệ quan trọng như bảo vệ quá dòng, quá áp và quá xả, giúp đảm bảo an toàn cho cả pin và thiết bị.

**Nguyên Lý Hoạt Động Của Mạch TP4056**

**Kết Nối Nguồn Sạc:** Khi cắm nguồn sạc 5V vào cổng micro USB (hoặc thông qua các chân IN+ và IN-), IC TP4056 bắt đầu hoạt động và cung cấp dòng điện để sạc pin.

**Điều Chỉnh Dòng Sạc:** IC TP4056 tự động điều chỉnh dòng sạc dựa trên điện trở thiết lập trên mạch (thường là 1.2kΩ để cung cấp dòng sạc tối đa 1A). Dòng sạc sẽ duy trì ổn định ở mức này cho đến khi điện áp pin tiến gần đến 4.2V.

**Kiểm Soát Điện Áp:**

* Khi pin đạt đến 4.2V, TP4056 chuyển từ chế độ sạc nhanh sang chế độ sạc duy trì (trickle charge), giảm dòng sạc để tránh làm hỏng pin.
* Nếu pin đầy, TP4056 sẽ tự động ngắt sạc để tránh tình trạng quá sạc, bảo vệ tuổi thọ của pin.

**Chức Năng Bảo Vệ:**

* Bảo vệ quá áp: Nếu điện áp pin vượt quá 4.2V, mạch sẽ ngắt sạc.
* Bảo vệ quá xả: Khi điện áp pin giảm xuống dưới mức cho phép (khoảng 2.4-2.5V), IC DW01 ngắt kết nối để ngăn chặn hư hại cho pin.
* Bảo vệ quá dòng: Nếu dòng xả vượt mức an toàn, mạch sẽ ngắt để bảo vệ pin và thiết bị.

**Quá Trình Xả:**

* Khi pin được sạc đầy và ngắt khỏi nguồn sạc, pin sẽ cung cấp điện qua các cổng OUT+ và OUT-.
* IC bảo vệ và MOSFET sẽ tiếp tục giám sát dòng xả để đảm bảo an toàn trong quá trình hoạt động.

Hình 2. 12 Mạch sạc pin 1cell 18650 có bảo vệ TP4056 1A với cổng nối micro USB

**THÔNG SỐ KỸ THUẬT**

* Điện áp vào: 5VDC
* Cổng Sạc MicroUSB ( Cổng sạc điện thoại thông dụng hiện nay )
* Điện Áp Sạc: 4.2V
* Dòng sạc tối đa: 1A
* Bảo vệ xả đến điện áp: 2.5V ( Tránh làm cạn Pin)
* Bảo Vệ Xả Quá Dòng: 3A
* Kích thước: 26x17MM

1. **Thiết kế mạch in**
2. **Thiết kế phần cứng**

Từ những lựa chọn linh kiện các khối cho hệ thống giám sát môi trường, thiết kế mạch của phần cứng cho đề tài như sau:

1. **Thiết kế mạch nguyên lý**
2. **Khối nguồn**

**A diagram of a circuit

Description automatically generated**

Hình 2. 13 Sơ đồ nguyên lý khối nguồn

Khối nguồn sử dụng nguồn điện 12V qua IC ổn áp 7805 cho dòng đầu ra 5V để cung cấp nguồn điện cho các khối khác trong mạch. Mạch sẽ lấy nguồn ngoài 220V rồi qua một adapter biến dòng xoay chiều 220V thành dòng 1 chiều 12V và dòng sẽ đi vào jack cắm cấp nguồn cho mạch. Điện áp 12V lúc này sẽ được biến đổi qua ic ổn áp 7805 để hạ áp xuống 5V và 2 tụ gốm sẽ giúp mạch lọc nguồn.

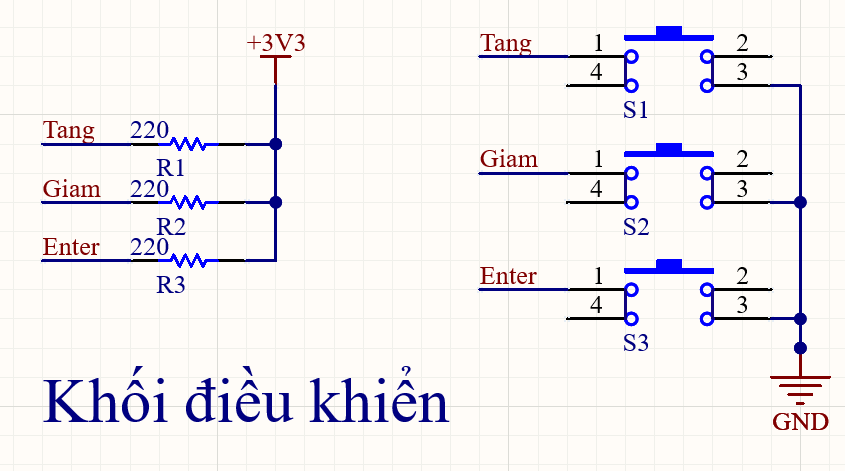
1. **Khối cảm biến**

A diagram of a circuit

Description automatically generatedMạch sử dụng cảm biến nhiệt độ độ ẩm DHT11 để đo thông số môi trường

Hình 2. 14 Sơ đồ nguyên lý cảm biến DHT11

1. **Khối điều khiển**

****Khối điều khiển sử dụng 3 công tắc nút nhấn lấy nguồn 3.3V từ khối xử lý trung tâm cho phép người dùng có thể cài đặt các giá trị thông số ngưỡng của môi trường để có thể đưa ra những biện pháp an toàn khi tình trạng môi trường chuyển biến xấu.

Hình 2. 15 Sơ đồ nguyên lý khối điều khiển

1. **Khối thông báo (cảnh báo)**

A diagram of a circuit

Description automatically generatedKhối cảnh báo sử dụng còi và đèn LED sẽ báo hiệu tại trạm khi thông số môi trường nhiệt độ độ ẩm vượt qua ngưỡng an toàn được người dùng thiết lập từ trước qua **Khối điều khiển** để người dùng nắm được thông tin và có những biện pháp phù hợp.

Hình 2. 16 Sơ đồ nguyên lý khối thông báo

1. **Khối thời gian thực**

A diagram of a computer chip

Description automatically generatedKhối thời gian thực sử dụng nguồn 5V từ ESP32 cho module thời gian thực RTC DS3231 giúp ta có thể theo dõi thời gian ngày giờ cập nhật liên tục theo thời gian thực hiện tại

Hình 2. 17 Sơ đồ nguyên lý khối thời gian thực

1. **Khối hiển thị dữ liệu**

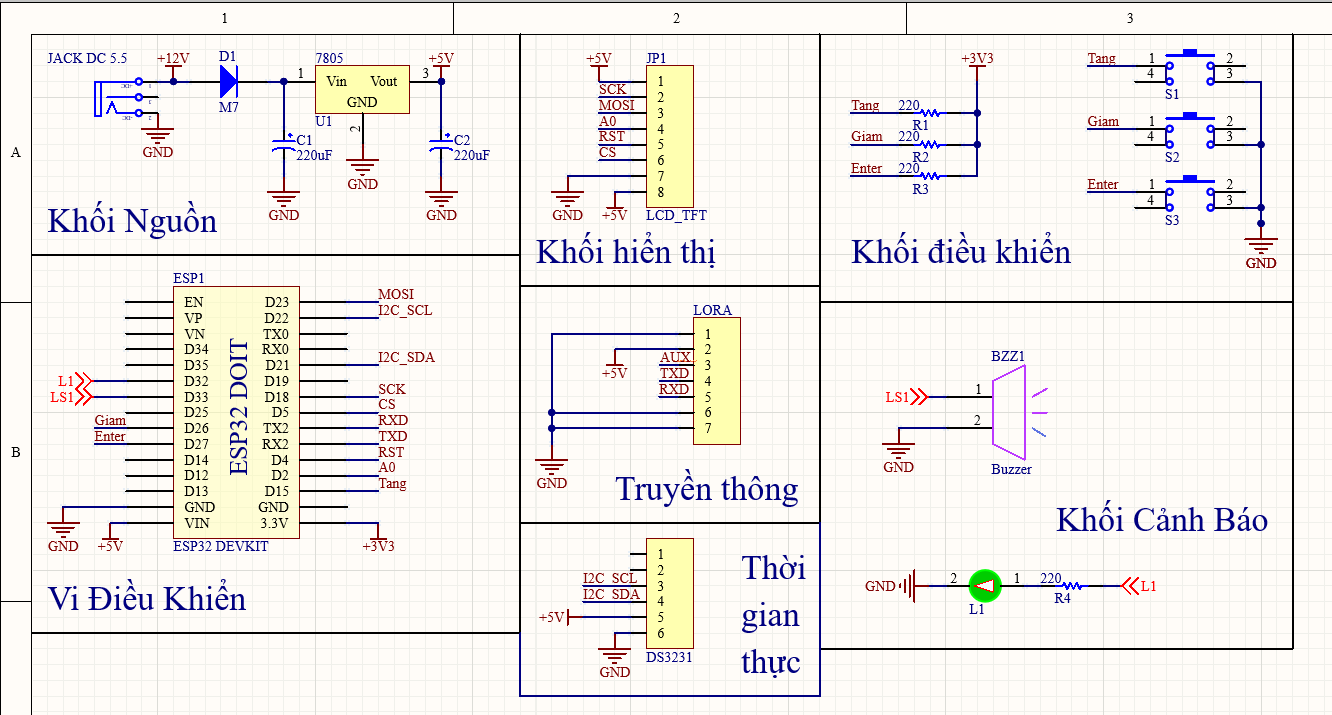
A diagram of a circuit

Description automatically generatedKhối hiển thị dữ liệu sử dụng nguồn 5V từ khối xử lý trung tâm cho màn hình TFT LCD hiển thị dữ liệu thu được tại trạm giúp có thể theo dõi trực tiếp tại trạm. Các thông số được hiển thị trên màn hình TFT LCD sẽ là ngày giờ thời gian thực của khối thời gian thực và các chỉ số nhiệt độ độ ẩm môi trường của khối cảm biên DHT11

Hình 2. 18 Sơ đồ nguyên lý khối hiển thị

**Khối truyền thông**

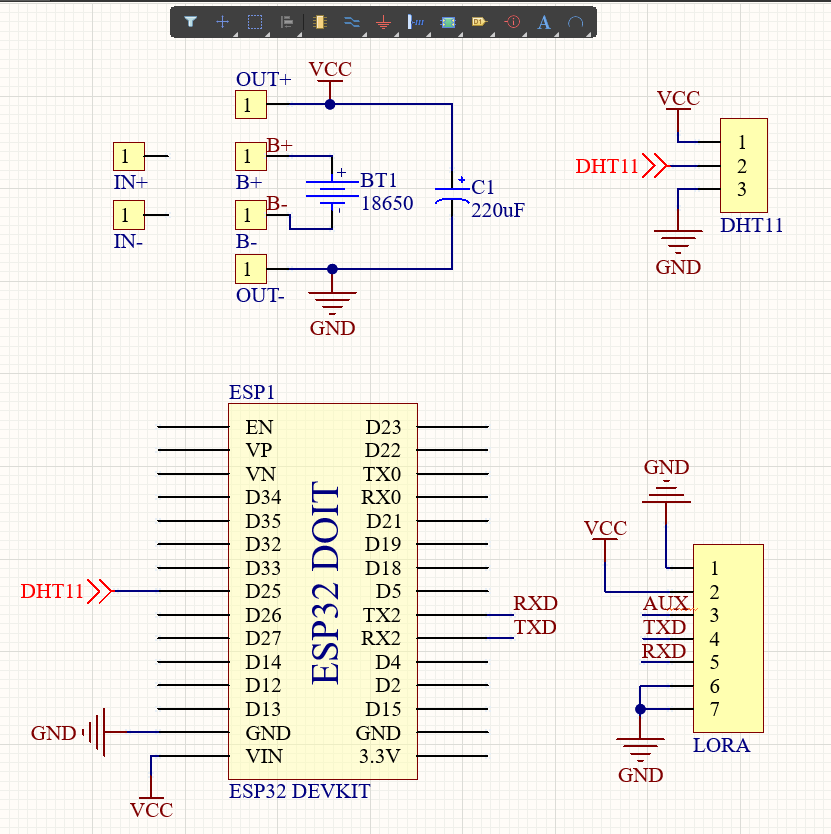
Khối truyền thông sử dụng nguồn 5V từ đầu ra VCC của ESP32 tại khối xử lý trung tâm cho 2 LORA truyền tín hiệu từ trạm phát đến nơi nhận. Tín hiệu có thể truyền đi xa nhất 3km



Hình 2. 19 Sơ đồ nguyên lý khối truyền thông

1. **Khối xử lý trạm từ xa**

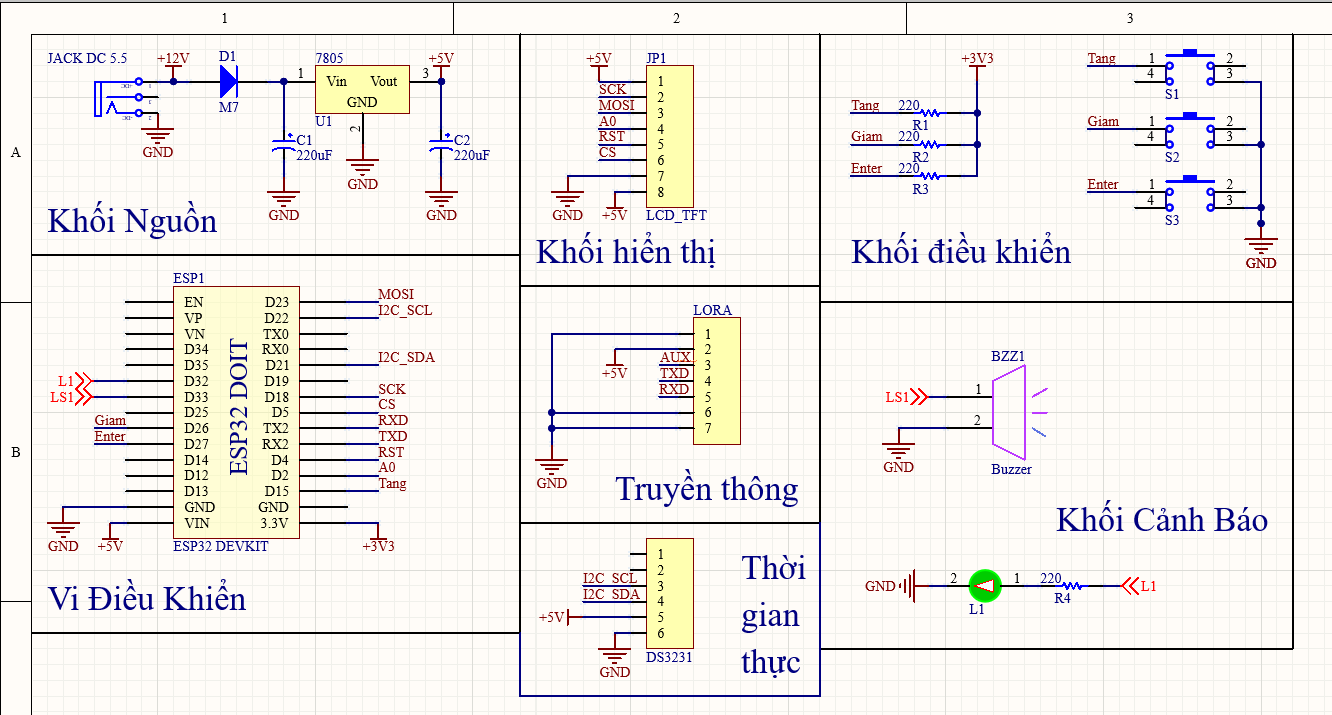
Khối xử lý trạm từ xa sử dụng nguồn 4,2V từ mạch sạc TP4056 để giao tiếp với các khối



Hình 2. 20 Sơ đồ khối xử lý trạm từ xa

**Khối xử lý trung tâm**

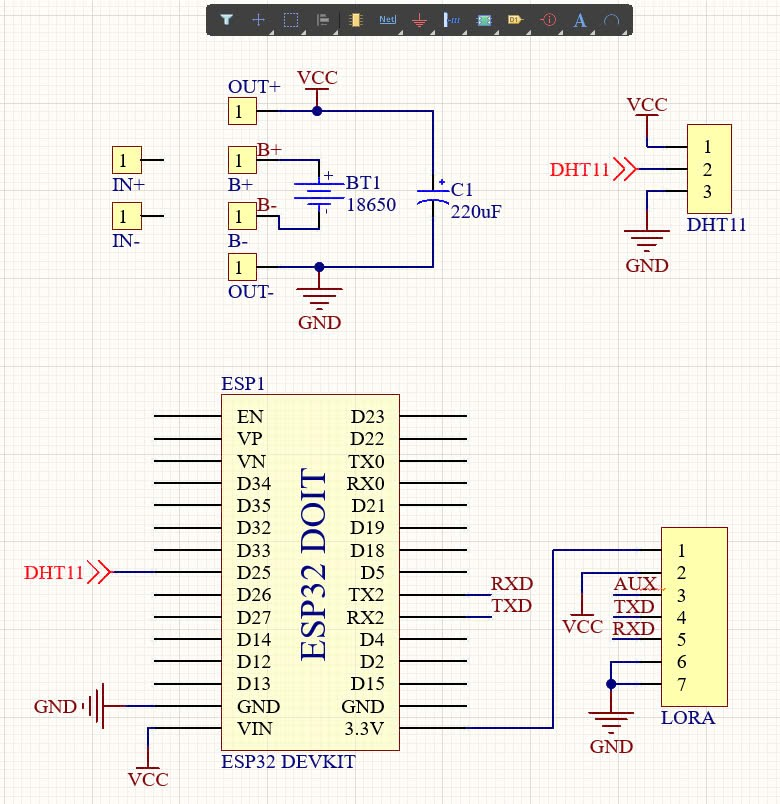
Khối xử lý trung tâm sử dụng nguồn 5V từ đầu ra Vout của IC ổn áp LM7805, giao tiếp với các module ngoại vi.



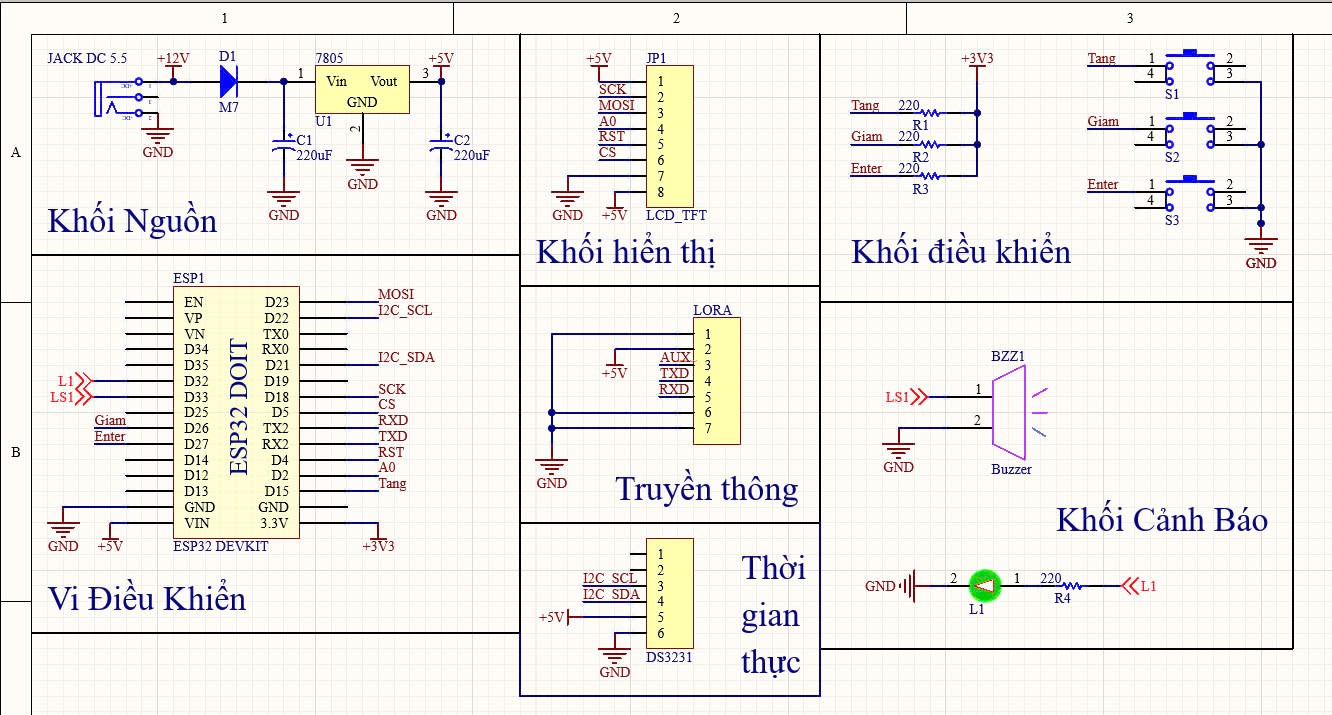
Hình 2. 21 Sơ đồ nguyên lý khối xử lý trung tâm

Các tín hiệu được kết nối tới khối xử lý trung tâm theo đúng các tín hiệu ở các khối ngoại vi

1. **Sơ đồ nguyên lý tổng quát**

****Mô hình mạch của hệ thống giám sát môi trường qua LORA được chia làm 2 phần chính mạch truyền hay gửi tín hiệu và mạch thu hay nhận tín hiệu truyền tới ứng với 2 LORA được sử dụng trong hệ thống với chức năng truyền phát tín hiệu

Hình 2. 22 Sơ đồ nguyên lý mạch phát

**

Hình 2. 23 Sơ đồ nguyên lý mạch thu

**NGUYÊN LÝ HOẠT ĐỘNG:**

* **Mạch truyền (phát):** Mạch gồm một cảm biến nhiệt độ độ ẩm môi trường DHT11, tụ hóa, một nguồn cấp (đế pin 18650), mạch sạc pin có bảo vệ nguồn TP4056 1A Micro USB, Module LORA và ESP32

Mạch sử dụng vi điều khiển ESP32 DEV KIT làm trung tâm điều khiển, nhận nguồn từ pin 18650 Khi nhận nguồn 5V từ pin, nguồn điện sẽ chạy qua và cung cấp năng lượng cho toàn bộ hệ thống. Tụ điện C1 trong mạch được sử dụng để làm ổn định điện áp của pin, giảm nhiễu, mạch bảo vệ pin giúp pin tránh việc sạc hoặc xả quá mức, đảm bảo hoạt động an toàn và bền bỉ, đặc biệt quan trọng khi các thiết bị trên mạch (ESP32, cảm biến DHT11, LORA) tiêu thụ dòng điện không đều, giúp giảm việc dao động điện áp.

Cảm biến DHT11 được kết nối với vi điều khiển trung tâm ESP32 để đo thông số nhiệt độ, độ ẩm môi trường. Dữ liệu từ cảm biến DHT11 được truyền đến ESP32 thông qua chân GPIO. ESP32 sẽ xử lý dữ liệu này và giao tiếp với module LORA. LORA bên mạch này sẽ là mạch truyền, truyền dữ liệu tới thiết bị LORA bên mạch thu (nhận) trong khoảng cách xa mà kEVKIThông cần xử dụng WIFI hay BLUETOOTH.

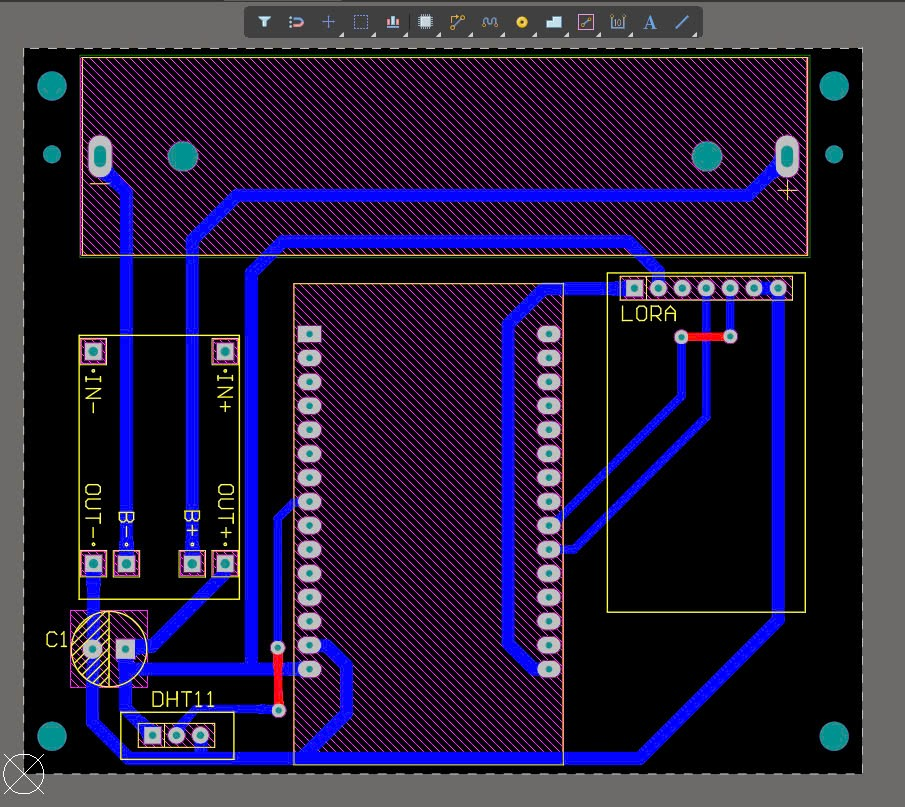
* **Mạch thu (nhận):** Mạch được chia làm nhiều khối ngoại vi với các chức năng khác nhau và quan trọng là một vi điều khiển trung tâm ESP32 đóng vai trò là khối xử lý trung tâm của mạch

Mạch sử dụng vi điều khiển ESP32 DEVKIT làm trung tâm điều khiển, lấy năng lượng từ nguồn điện ngoài (nguồn xoay chiều 220V), qua một adapter giúp biến dòng xoay chiều 220V thành dòng 1 chiều 12V. Nguồn 1 chiều 12V sẽ được chuyển vào khối nguồn, qua JACK DC cấp nguồn đii qua IC ổn áp xuống thành nguồn 1 chiều 5V qua các tụ C1, C2 để lọc nhiễu và cấp điện áp cho toàn mạch.

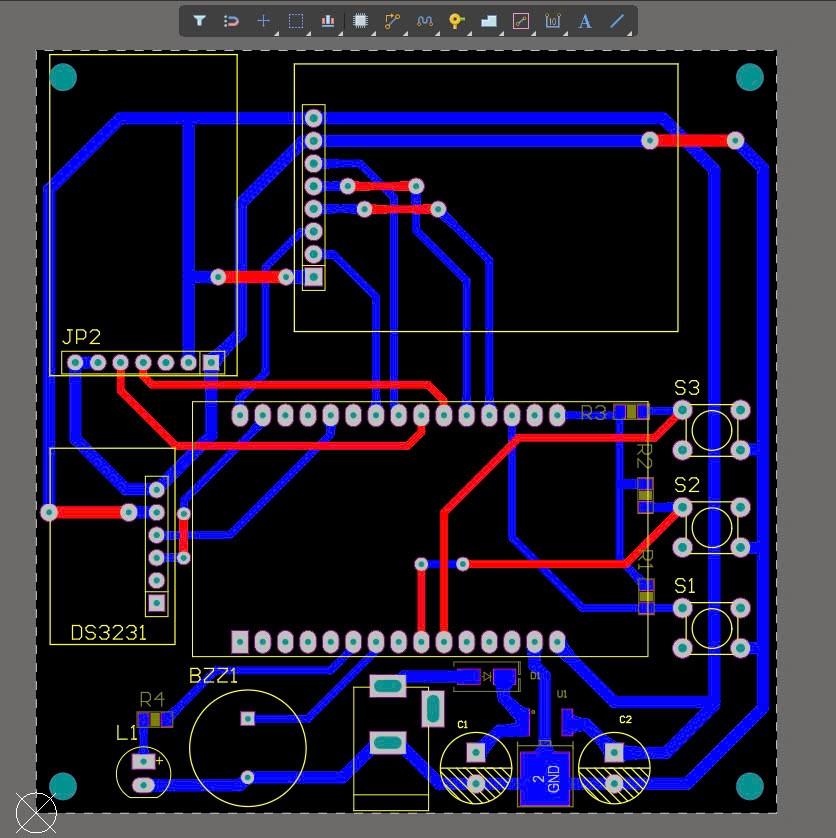
Mạch sử dụng vi điều khiển ESP32 DEVKIT làm trung tâm điều khiển, nhận nguồn 5V và giao tiếp với các khối ngoại vi. Module thời gian thực RTC DS3231 sẽ cung cấp dữ liệu cụ thể về thời gian để hỗ trợ các tính năng liên quan đến lịch trình, hay chỉ đơn giản là theo dõi giờ giấc. Trung tâm điều khiển nhận tín hiệu được truyền đến qua LORA của cảm biến DHT11 và dữ liệu giờ giấc từ RTC DS3231, tổng hợp dữ liệu, tính toán và gửi đến màn hình LCD TFT. Màn hình nhận và sẽ hiển thị dữ liệu ngày giờ và các thông số của môi trường nơi trạm truyền gửi tới. Ba nút bấm tại khối điều khiển giúp người dùng có thể cài đặt trực tiếp thông số ngưỡng của nhiệt độ và độ ẩm môi trường. Khi thông số ấy vượt quá thông số ngưỡn được cài đặt, tín hiệu sẽ được gửi tới bộ xử lý trung tâm, bộ xử lý trung tâm sẽ tổng hợp và xử lý dữ liệu, gửi dữ liệu đến khối thông báo (cảnh báo). Lúc này khi nhận được tín hiệu, LED sẽ nhấp nháy và còi sẽ kêu, báo hiệu cho người dùng biết và để người dùng đưa ra được những giải pháp phù hợp.

1. **Thiết kế mạch in**

* **Mạch truyền (phát)**



Hình 2. 24 Mạch in 2D mạch truyền

* **Mạch thu (nhận)**

Hình 2. 25 Mạch in 2D mạch thu

1. **Thiết kế sơ đồ thuật toán**
2. **Mạch truyền (Phát)**

Tạo chuỗi giá trị đọc “Nhiệt độ” và “Độ ẩm” từ cảm biến

Đọc nhiệt độ độ ẩm

In ra Serial: “Bắt đầu gửi dữ liệu từ cảm biến”

Khởi tạo Serial (115200) cho máy tính.

Khời tạo Serial (9600) cho Uart

Khởi tạo DHT11

Đọc cảm biến

**Sai**

In ra màn hình: “Không thể đọc cảm biến DHT!”

Chờ 2 giây

In giá trị lên Serial

**Đúng**

Gửi tín hiệu qua LORA

1. **Mạch thu (Nhận)**

* **Thuật toán cập nhật thời gian**

**Sai**

**Đúng**

Hiển thị giá trị giờ-phút-giây,  
ngày-tháng-năm

Định dạng thêm số 0 đằng trước

Định dạng như bình thường

Lấy chỉ số ngày-tháng-năm,  
giờ-phút-giây từ mạch RTC

Giá trị giờ-phút-giây<10

* **Thuật toán cập nhật nhiệt độ độ ẩm**

**Đúng**

**Sai**

Kiểm tra xem có dữ liệu từ LORA không

Chuyển **chuỗi** dữ liệu thành kiểu **số thực**

Hiển thị giá trị nhiệt độ độ ẩm lên màn hình TFT

**Sai**

**Đúng**

Kiểm tra nhiệt độ độ ẩm có vượt ngưỡng cảnh báo

Tắt đèn LED

Tắt còi báo động

Bật đèn LED

Bật còi báo động

* **Thuật toán kiểm tra nút nhấn**

**Sai**

**Sai**

**Sai**

Nút giảm được ấn

**Sai**

**Sai**

Tăng giá trị cảnh báo ngưỡng nhiệt độ 1 đơn vị

Nút tăng được ấn

Lưu giá trị độ ẩm vào EEPROOM

**Đúng**

**Đúng**

**Đúng**

**Đúng**

Hiển thị giá trị vừa tăng

Tăng giá trị ngưỡng cảnh báo độ ẩm 1 đơn vị

Nút tăng được ấn

Giảm giá trị cảnh báo ngưỡng nhiệt độ 1 đơn vị

Hiển thị giá trị vừa giảm

**Đúng**

**Đúng**

**Đúng**

**Đúng**

Xóa màn hình

Xóa màn hình

Tăng giá trị số đếm ENTER lên 1 đơn vị

Nút ENTER được nhấn

**Đúng**

Quay lại màn hình chính

Hiển thị giá trị vừa giảm

Hiển thị giá trị vừa tăng

Giảm giá trị cảnh báo ngưỡng độ ẩm 1 đơn vị

Nút giảm được ấn

Hiển thị giao diện chính độ ẩm

Hiển thị giao diện chính nhiệt độ

Biến đếm số lần nút ENTER = 0

Gán giá gtri biến đếm nhấn ENTER = 0

Nút ENTER được nhấn 1 lần

Nút ENTER được nhấn >2 lần

Nút ENTER được nhấn 2 lần

* **Sơ đồ khối tổng quát**

Kiểm tra nút nhấn

Cập nhật thời gian trên màn hình TFT

Cập nhật nhiệt độ độ ẩm trên màn hình TFT

Khối lọc giao tiếp Serial  
Cấu hình các chân GPIO  
Khối tạo EEPROOM  
Khối tạo RTC  
Khối tạo màn hình TFT

**CHƯƠNG III: CHẾ TẠO VÀ THỬ NGHIỆM HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ, ĐỘ ẨM QUA LORA**

**3.1. Chế tạo hệ thống giám sát nhiệt độ, độ ẩm qua lora**

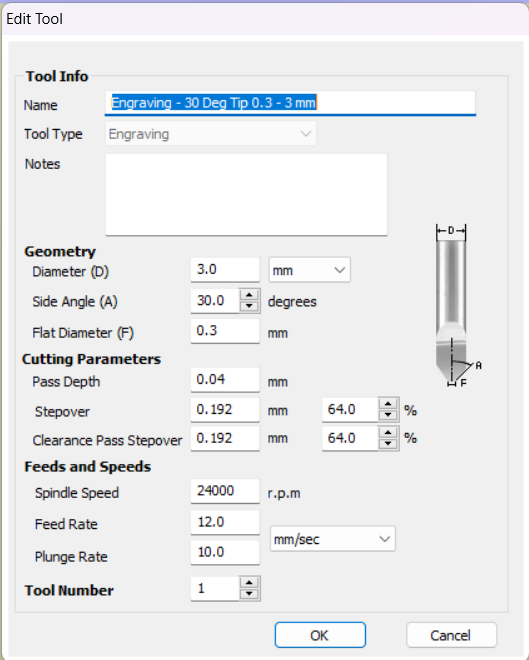
**Bước 1:** Dựa vào sơ đồ nguyên lý, chuẩn bị đầy các linh kiện điện tử cần thiết (IC, điện trở, tụ điện, ESP32, LCD TFT diode, v.v.)

**Bước 2:** Xuất file để CNC mạch

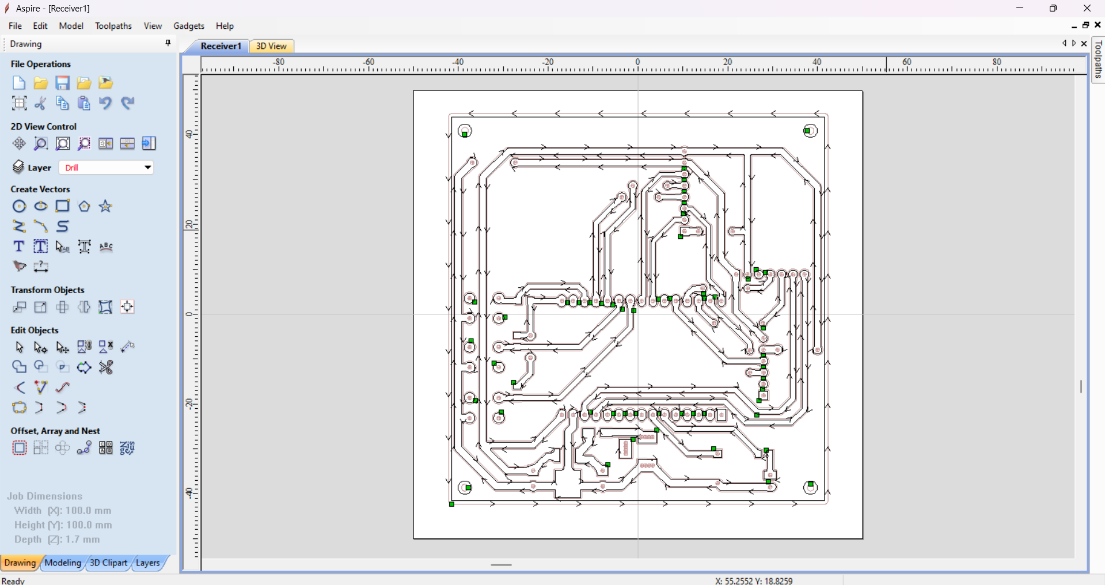
* Sau khi thiết kế hoàn tất, xuất file Gerber từ phần mềm Altium

**Bước 3:** Cài đặt phần mềm CNC

* Nạp file thiết kế vào phần mềm điều khiển CNC: Dùng Vectric Aspire để nạp file Gerber/Excellon vào máy CNC.
* Cấu hình thông số CNC: Cài đặt các thông số cần thiết như:
* Tốc độ quay của mũi phay (Spindle speed): Tùy thuộc vào chất liệu và độ dày của tấm phím đồng.
* Tốc độ cắt (Feed rate): Điều chỉnh tốc độ di chuyển của mũi phay để đảm bảo độ chính xác và tránh làm hỏng tấm đồng.
* Độ sâu khắc (Cut depth): Đặt độ sâu đủ để loại bỏ lớp đồng mà không làm hỏng lớp vật liệu nền (thường khoảng 0.1mm - 0.2mm).



Hình 3. 1 Cấu hình thông số mũi dao CNC

**

Hình 3. 2 Mô phỏng đường đi của dao khi khắc mạch

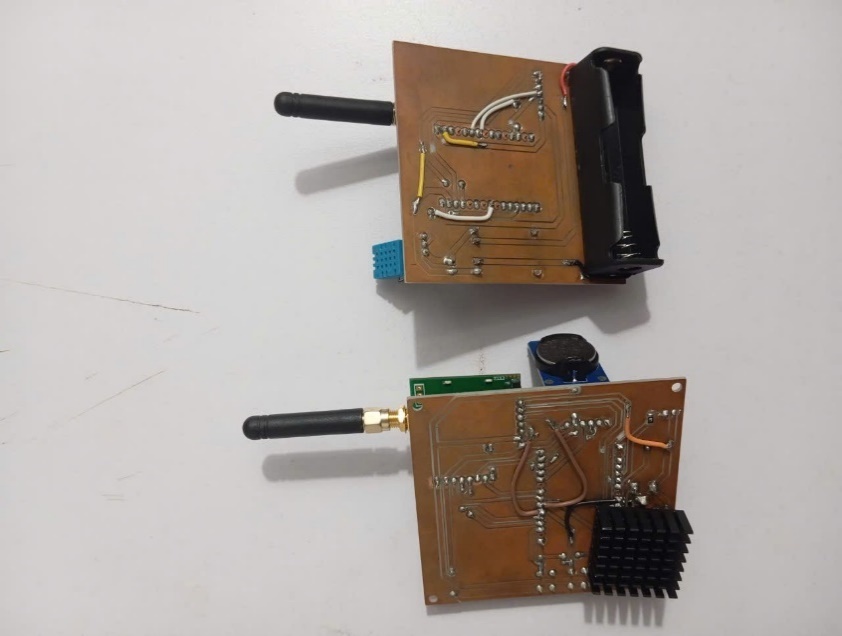
**Bước 4:** Thi công khắc mạch PCB

* Khắc mạch: Máy CNC sẽ phay loại bỏ lớp đồng không cần thiết, chỉ giữ lại các đường dẫn mạch theo thiết kế.
* Khoan lỗ linh kiện: Sau khi khắc xong, máy CNC sẽ thực hiện khoan lỗ tại các vị trí đã được xác định cho chân linh kiện (thường là các lỗ nhỏ từ 0.6mm - 1mm tùy linh kiện).

**Bước 5: Kiểm tra mạch in**

Sử dụng đồng hồ vạn năng để kiểm tra các đường mạch in, đảm bảo không có chỗ nào bị đứt hoặc chạm chập.

**Bước 6:** Thi công hàn linh kiện và lắp linh kiện lên mạch.



Hình 3. 3 Hàn và lắp linh kiện lên mạch điều khiển 2 mặt trước và sau

**Bước 7:** Nạp code và kiểm thử các tính năng cơ bản

## **3.2. Kiểm tra và đánh giá**

### 3.2.1. Thử nghiệm độ chính xác của các cảm biến nhiệt độ DHT11

Quy trình thực nghiệm:

* Chuẩn bị cảm biến nhiệt độ cầm tay (Điện thoại)
* Đo nhiệt độ bằng cảm biến trong hệ thống
* Ghi lại kết quả và so sánh

Bảng 3: Bảng so sánh kết quả đo

|  |  |  |
| --- | --- | --- |
| **Lần đo thứ** | **Máy cầm tay** | **Cảm biến DHT11** |
| 1 | 30.5 | 30 |
| 2 | 31.1 | 30.7 |
| 3 | 31.8 | 30.9 |
| 4 | 31 | 30.2 |
| 5 | 30.8 | 30.1 |

Dựa vào bảng kết quả đo cho thấy cảm biến đo nhiệt trong hệ thống hoạt động chính xác. Độ sai lệch lớn nhất là 0.9 độ

### 3.2.2. Thử nghiệm độ chính xác của module LORA

Để thực hiện thử nghiệm này, cần đặt khoảng cách của 2 mạch truyền, nhận ở một khoảng cách nhất định, đồng thời xem độ chính xác việc truyền nhận của hệ thống, có truyền nhận được không.

Bảng 4: Kết quả so sánh độ chính của module LORA

|  |  |  |
| --- | --- | --- |
| **Lần thử nghiệm thứ** | **Khoảng Cách** | **Kết quả truyền nhận** |
| 1 | 50m | Nhận được dữ liệu |
| 2 | 100m | Nhận được dữ liệu |
| 3 | 250m | Nhận được dữ liệu |
| 4 | 350m | Nhận được dữ liệu |
| 5 | 500m | Nhận được dữ liệu |
| 6 | 600m | Nhận được dữ liệu |
| 7 | 650m | Không nhận được dữ liệu |
| 8 | 1km | Không nhận được dữ liệu |

Sau nhiều lần thử nghiệm thì trong môi trường đô thị phức tạp như đường phố Hà Nội, với nhiều tòa nhà cao tầng, xe cộ và các nguồn nhiễu khác, khoảng cách truyền mô-đun Lora AS32 sẽ bị giảm so với điều kiện thuận lợi.

### 3.2.3. Đánh giá chung các thử nghiệm

Dựa trên các thử nghiệm đã thực hiện, hệ thống giám sát nhiệt độ, độ ẩm qua LoRa hoạt động ổn định với mức độ chính xác cao. Đối với cảm biến nhiệt độ DHT11, sai số giữa các lần đo so với thiết bị đo cầm tay không đáng kể, cho thấy cảm biến hoạt động tốt trong phạm vi yêu cầu. Độ sai lệch nhỏ nhất là 0,4 độ và lớn nhất là 0,9 độ, trong phạm vi chấp nhận được đối với loại cảm biến này.

Đối với module LoRa, quá trình truyền nhận dữ liệu cũng đạt kết quả khả quan trong khoảng cách lên đến 600m trong môi trường đô thị phức tạp, nhưng bắt đầu gặp vấn đề khi khoảng cách vượt quá 650m. Điều này phù hợp với thực tế, khi các yếu tố như nhiễu sóng, vật cản và mật độ các tòa nhà cao tầng ảnh hưởng lớn đến khả năng truyền dữ liệu của LoRa. Với điều kiện thử nghiệm ở các khu vực ít chướng ngại vật hơn, khoảng cách truyền có thể sẽ cải thiện đáng kể.

**3.2.4 Mã code chạy chương trình**

**a. Code truyền**

#include <DHT.h>

#include <HardwareSerial.h>

#define DHTPIN 25       // Chân kết nối với DHT

#define DHTTYPE DHT11  // Hoặc DHT22 nếu bạn dùng cảm biến này

DHT dht(DHTPIN, DHTTYPE);

HardwareSerial LoRaSerial(2);  // Sử dụng UART2

void setup() {

  Serial.begin(115200);

  LoRaSerial.begin(9600, SERIAL\_8N1, 16, 17); // Baudrate của LoRa AS32 là 9600

  dht.begin();  // Khởi động cảm biến DHT

  delay(2000);

  Serial.println("Bắt đầu gửi dữ liệu từ cảm biến...");

}

void loop() {

  // Đọc nhiệt độ và độ ẩm

  float humidity = dht.readHumidity();

  float temperature = dht.readTemperature();

  // Kiểm tra nếu có lỗi khi đọc

  if (isnan(humidity) || isnan(temperature)) {

    Serial.println("Không thể đọc cảm biến DHT!");

    return;

  }

  // Chuẩn bị dữ liệu gửi đi, thêm ký tự phân cách giữa 2 giá trị

  String message = String(temperature,1) + "," + String(humidity,1);

  // Gửi dữ liệu qua LoRa

  LoRaSerial.println(message);

  Serial.println("Gửi tin nhắn: " + message);

  delay(2000);  // Gửi mỗi 2 giây

}

**b. Code nhận**

#include <HardwareSerial.h>

HardwareSerial LoRaSerial(2);  // Sử dụng UART2

#include <Adafruit\_GFX.h>    // Core graphics library

#include <Adafruit\_ST7735.h> // Hardware-specific library for ST7735

#include <SPI.h>

#include <Wire.h>

#include <RTClib.h>

#include <EEPROM.h>  // Thư viện EEPROM để lưu giá trị vào bộ nhớ

RTC\_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thur", "Fri", "Sat"};

#define TFT\_CS         5

#define TFT\_RST        4

#define TFT\_DC         2

#define TFT\_SCLK       18

#define TFT\_MOSI       23

Adafruit\_ST7735 tft = Adafruit\_ST7735(TFT\_CS,  TFT\_DC, TFT\_RST);

const int button1Pin = 15;  // Nút nhấn tăng giá trị

const int button2Pin = 26;  // Nút nhấn giảm giá trị

const int button3Pin = 27;  // Nút nhấn chuyển đổi màn hình và lưu giá trị

const int ledPin = 32;      // Đèn LED

const int buzzerPin = 33;   // Còi báo

float temperatureWarning = 0;  // Ngưỡng nhiệt độ cảnh báo

float humidityWarning = 0;     // Ngưỡng độ ẩm cảnh báo

float step = 0.5;              // Bước tăng/giảm giá trị

const int EEPROM\_TEMP\_ADDR = 0;  // Địa chỉ lưu nhiệt độ trong EEPROM

const int EEPROM\_HUMI\_ADDR = 10; // Địa chỉ lưu độ ẩm trong EEPROM

unsigned long prevTime = millis();

unsigned long lastButtonPress = 0;

unsigned long debounceDelay = 200;

int screenState = 0;  // 0: Màn hình chính, 1: Điều chỉnh nhiệt độ, 2: Điều chỉnh độ ẩm

static const uint8\_t bitmap\_Temp [] = {

  0x00, 0x06, 0x0f, 0x00, 0x00, 0x06, 0x1f, 0x80, 0x00, 0x06, 0x30, 0xc0, 0x02, 0x00, 0x20, 0x40,

  0x07, 0x00, 0x20, 0x40, 0x03, 0x0f, 0x26, 0x40, 0x00, 0x3f, 0xe6, 0x40, 0x00, 0x70, 0x66, 0x40,

  0x00, 0xc0, 0x26, 0x40, 0x00, 0xc0, 0x26, 0x40, 0x01, 0x80, 0x26, 0x40, 0x39, 0x80, 0x26, 0x40,

  0x39, 0x80, 0x26, 0x40, 0x01, 0x80, 0x26, 0x40, 0x00, 0xc0, 0x26, 0x40, 0x00, 0xc0, 0x26, 0x40,

  0x00, 0x70, 0x66, 0x40, 0x00, 0x3f, 0xe6, 0x60, 0x01, 0x1f, 0xe6, 0x60, 0x07, 0x00, 0xcf, 0x30,

  0x06, 0x00, 0xcf, 0x30, 0x00, 0x06, 0xd9, 0xb0, 0x00, 0x06, 0xcf, 0xb0, 0x00, 0x06, 0xcf, 0x30,

  0x00, 0x00, 0x60, 0x60, 0x00, 0x00, 0x70, 0xe0, 0x00, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x0f, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

static const uint8\_t bitmap\_Humi [] = {

  0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x0a, 0x00, 0x00, 0x00, 0x11, 0x00,

  0x00, 0x10, 0x11, 0x80, 0x00, 0x30, 0x20, 0x80, 0x00, 0x28, 0x20, 0x80, 0x00, 0x4c, 0x40, 0x40,

  0x00, 0x44, 0x42, 0x40, 0x00, 0x86, 0x42, 0x40, 0x01, 0x82, 0x60, 0x40, 0x01, 0x01, 0x20, 0x80,

  0x02, 0x01, 0x1f, 0x00, 0x02, 0x00, 0x80, 0x00, 0x06, 0x00, 0x80, 0x00, 0x04, 0x00, 0x40, 0x00,

  0x0c, 0x00, 0x40, 0x00, 0x08, 0x00, 0x60, 0x00, 0x08, 0x00, 0x20, 0x00, 0x08, 0x00, 0x20, 0x00,

  0x08, 0x02, 0x20, 0x00, 0x08, 0x06, 0x20, 0x00, 0x08, 0x0c, 0x60, 0x00, 0x0c, 0x08, 0x40, 0x00,

  0x06, 0x00, 0xc0, 0x00, 0x03, 0x01, 0x80, 0x00, 0x01, 0xee, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

void setup() {

  Serial.begin(115200);

  LoRaSerial.begin(9600, SERIAL\_8N1, 16, 17); // Baudrate của LoRa AS32 là 9600

  Serial.println("Chờ nhận dữ liệu LoRa...");

  pinMode(button1Pin, INPUT\_PULLUP);

  pinMode(button2Pin, INPUT\_PULLUP);

  pinMode(button3Pin, INPUT\_PULLUP);

  pinMode(ledPin, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  digitalWrite(ledPin, LOW);

  digitalWrite(buzzerPin, LOW);

  // Đọc giá trị ngưỡng cảnh báo từ EEPROM

  EEPROM.begin(512);

  EEPROM.get(EEPROM\_TEMP\_ADDR, temperatureWarning);

  EEPROM.get(EEPROM\_HUMI\_ADDR, humidityWarning);

  EEPROM.commit();

  Wire.begin(21, 22); // SDA = GPIO 21, SCL = GPIO 22

  if (!rtc.begin()) {

    Serial.println("Couldn't find RTC");

    while (1);

  }

  // rtc.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_));

  tft.initR(INITR\_BLACKTAB);

  tft.fillScreen(ST7735\_BLACK);

  tft.setRotation(3);

}

void loop() {

  unsigned long currentTime = millis();

  if (currentTime - prevTime > 1000) {

    if (screenState == 0) {

      updateTime();

      updateWeather();

    }

    prevTime = currentTime;

  }

  checkButtons();  // Kiểm tra các nút nhấn

}

void checkButtons() {

  if (digitalRead(button3Pin) == LOW && millis() - lastButtonPress > debounceDelay) {

    lastButtonPress = millis();

    screenState++;

    if (screenState > 2) {

      screenState = 0;  // Quay lại màn hình chính

    }

    if (screenState == 1) {

      // Xóa màn hình và chuyển sang màn hình điều chỉnh nhiệt độ

      tft.fillScreen(ST7735\_BLACK);

      tft.setCursor(20, 30);

      tft.setTextColor(ST77XX\_GREEN, ST77XX\_BLACK);

      tft.setTextSize(2);

      tft.println("Set Temp:");

      displayTemperatureWarning();

    } else if (screenState == 2) {

      // Xóa màn hình và chuyển sang màn hình điều chỉnh độ ẩm

      tft.fillScreen(ST7735\_BLACK);

      tft.setCursor(20, 30);

      tft.setTextColor(ST77XX\_GREEN, ST77XX\_BLACK);

      tft.setTextSize(2);

      tft.println("Set Humi:");

      displayHumidityWarning();

    } else if (screenState == 0) {

      // Lưu ngưỡng nhiệt độ và độ ẩm vào EEPROM và quay lại màn hình chính

      EEPROM.put(EEPROM\_TEMP\_ADDR, temperatureWarning);

      EEPROM.put(EEPROM\_HUMI\_ADDR, humidityWarning);

      EEPROM.commit();

      Serial.print("Temperature and humidity warning saved: ");

      Serial.print(temperatureWarning);

      Serial.print(" C, ");

      Serial.println(humidityWarning);

      tft.fillScreen(ST7735\_BLACK);

      Serial.println("Back to main screen");

    }

  }

  if (screenState == 1 && millis() - lastButtonPress > debounceDelay) {

    // Điều chỉnh ngưỡng nhiệt độ

    if (digitalRead(button1Pin) == LOW) {

      temperatureWarning += step;

      displayTemperatureWarning();

      lastButtonPress = millis();

    } else if (digitalRead(button2Pin) == LOW) {

      temperatureWarning -= step;

      displayTemperatureWarning();

      lastButtonPress = millis();

    }

  }

  if (screenState == 2 && millis() - lastButtonPress > debounceDelay) {

    // Điều chỉnh ngưỡng độ ẩm

    if (digitalRead(button1Pin) == LOW) {

      humidityWarning += step;

      displayHumidityWarning();

      lastButtonPress = millis();

    } else if (digitalRead(button2Pin) == LOW) {

      humidityWarning -= step;

      displayHumidityWarning();

      lastButtonPress = millis();

    }

  }

}

void updateTime() {

  DateTime now = rtc.now();

  tft.setCursor(13, 10);

  tft.setTextColor(ST77XX\_GREEN, ST77XX\_BLACK);

  tft.setTextSize(1);

  String thu\_ = daysOfTheWeek[now.dayOfTheWeek()];

  tft.println(thu\_);

  tft.setCursor(58, 10);

  tft.setTextColor(ST77XX\_GREEN, ST77XX\_BLACK);

  tft.setTextSize(1);

  String currentDate = String(now.day()) + "-" + String(now.month()) + "-" + String(now.year());

  tft.println(currentDate);

  tft.drawRect(12, 24, 106, 28, ST77XX\_MAGENTA); // x, y, width, height

  tft.setCursor(18, 31);

  tft.setTextColor(ST77XX\_WHITE, ST77XX\_BLACK);

  tft.setTextSize(2);

  String formattedTime =

    (now.hour() < 10 ? "0" + String(now.hour()) : String(now.hour())) + ":" +

    (now.minute() < 10 ? "0" + String(now.minute()) : String(now.minute())) + ":" +

    (now.second() < 10 ? "0" + String(now.second()) : String(now.second()));

  tft.print(formattedTime);

}

void updateWeather() {

  if (LoRaSerial.available()) {

    String receivedMessage = LoRaSerial.readString();

    int commaIndex = receivedMessage.indexOf(',');

    if (commaIndex != -1) {

      String receivedTempStr = receivedMessage.substring(0, commaIndex);

      String receivedHumStr = receivedMessage.substring(commaIndex + 1);

      // Chuyển đổi từ String sang float

      float receivedTemp = receivedTempStr.toFloat();

      float receivedHum = receivedHumStr.toFloat();

      tft.drawBitmap(12, 60, bitmap\_Temp, 30, 30, ST7735\_BLUE);

      tft.setCursor(46, 67);

      tft.setTextColor(ST77XX\_CYAN, ST77XX\_BLACK);

      tft.setTextSize(2);

      tft.print(receivedTemp);

      printText("o", 105, 65, 1, ST77XX\_CYAN);

      printText("C", 110, 67, 2, ST77XX\_CYAN);

      tft.drawBitmap(12, 96, bitmap\_Humi, 30, 30, ST7735\_YELLOW);

      tft.setCursor(46, 103);

      tft.setTextColor(ST77XX\_CYAN, ST77XX\_BLACK);

      tft.setTextSize(2);

      tft.print(receivedHum);

      printText("%", 94, 103, 2, ST77XX\_CYAN);

      // Kiểm tra nếu vượt ngưỡng cảnh báo, đồng thời tránh trường hợp nhận giá trị rỗng hoặc không hợp lệ

      if (receivedTemp > temperatureWarning || receivedHum > humidityWarning) {

        digitalWrite(ledPin, HIGH);

        tone(buzzerPin, 3000);  // Kích hoạt còi báo với tần số 3000Hz

      } else {

        digitalWrite(ledPin, LOW);

        noTone(buzzerPin);  // Tắt còi báo

      }

    }

  }

}

void displayTemperatureWarning() {

  tft.setCursor(20, 50);

  tft.setTextColor(ST77XX\_CYAN, ST77XX\_BLACK);

  tft.setTextSize(2);

  tft.print(temperatureWarning);

  tft.println(" C");

}

void displayHumidityWarning() {

  tft.setCursor(20, 50);

  tft.setTextColor(ST77XX\_CYAN, ST77XX\_BLACK);

  tft.setTextSize(2);

  tft.print(humidityWarning);

  tft.println(" %");

}

void printText(String text, int x, int y, int text\_size, uint16\_t color) {

  tft.setCursor(x, y);

  tft.setTextColor(color, ST77XX\_BLACK);

  tft.setTextSize(text\_size);

  tft.println(text);

}

## **3.3. Kết luận chương 3**

Trong chương này, chúng tôi đã chế tạo và thử nghiệm thành công hệ thống giám sát nhiệt độ và độ ẩm sử dụng công nghệ LoRa. Hệ thống bao gồm các cảm biến DHT11 và module LoRa, cho phép giám sát từ xa trong môi trường đô thị. Các thử nghiệm cho thấy cảm biến DHT11 có độ chính xác cao, sai số không lớn, trong khi module LoRa hoạt động hiệu quả trong khoảng cách lên đến 600m trong môi trường có nhiều nhiễu.

Tuy nhiên, để cải thiện khả năng truyền dữ liệu trong các khu vực có nhiều vật cản, cần xem xét các biện pháp như nâng cấp ăng-ten, sử dụng module LoRa có công suất cao hơn hoặc tối ưu hóa vị trí đặt thiết bị truyền và nhận. Hệ thống có thể mở rộng và ứng dụng trong nhiều lĩnh vực khác nhau, từ giám sát nông nghiệp đến công nghiệp thông minh, khi các điều kiện thử nghiệm và ứng dụng được tối ưu hóa.

# **KẾT LUẬN CHUNG**

Trong đề tài này, chúng em đã nghiên cứu và phát triển hệ thống giám sát nhiệt độ môi trường sử dụng công nghệ LoRa. Hệ thống bao gồm các cảm biến nhiệt độ, module LoRa để truyền dữ liệu và một máy chủ trung tâm để thu thập và phân tích thông tin. Qua quá trình thử nghiệm, hệ thống đã chứng minh được khả năng giám sát nhiệt độ hiệu quả, trong khi module LoRa hoạt động hiệu quả trong khoảng cách lên đến 600m trong môi trường có nhiều nhiễu và tiêu thụ năng lượng thấp.

**Kết quả đạt được:**

* Thiết kế và triển khai thành công hệ thống giám sát nhiệt độ. Hệ thống hoạt động ổn định, cung cấp dữ liệu nhiệt độ chính xác và liên tục.
* Khả năng mở rộng và ứng dụng thực tế: Hệ thống có thể dễ dàng mở rộng để giám sát nhiều loại cảm biến khác nhau và ứng dụng trong nhiều lĩnh vực như nông nghiệp, công nghiệp và quản lý môi trường.
* Tiết kiệm năng lượng: Sử dụng công nghệ LoRa giúp giảm thiểu tiêu thụ năng lượng, kéo dài tuổi thọ của các thiết bị cảm biến.

**Hướng nghiên cứu tiếp theo:**

* Tích hợp thêm các loại cảm biến khác: Như cảm biến độ ẩm, chất lượng không khí để cung cấp thông tin môi trường toàn diện hơn.
* Phát triển ứng dụng di động: Giúp người dùng dễ dàng theo dõi và quản lý dữ liệu từ xa.
* Nâng cao bảo mật dữ liệu: Đảm bảo an toàn thông tin trong quá trình truyền và lưu trữ dữ liệu.
* Cải thiện khả năng truyền dữ liệu trong các khu vực có nhiều vật cản, sử dụng module LoRa có công suất cao hơn hoặc tối ưu hóa vị trí đặt thiết bị truyền và nhận.

**TÀI LIỆU THAM KHẢO**

1. [LM340, LM340A and LM7805 Family Wide VIN 1.5-A Fixed Voltage Regulators](https://www.ti.com/lit/ds/symlink/lm340.pdf)
2. [Sóng LORA cho ESP32](https://www.electroniclinic.com/esp32-lora-home-automation-long-rage-lora-sender-and-receiver-lora-sx1278/)
3. [Learning Electronics](https://lastminuteengineers.com/?__im-dzruiPrM=4340584258504190263)
4. [Arduino Tutorial: Using the 1.44" Color TFT display (ILI9163C) with Arduino](https://youtu.be/DSYB6sWGemU?si=YMYjdRuhz4-A4bKP)
5. [Real-time-clock module (RTC) explained](https://youtu.be/dEWzm3AM_Tw?si=l3vPHOLfikT5de8H)
6. [Cảm biến nhiệt độ-độ ẩm](https://mecsu.vn/ho-tro-ky-thuat/dht11-cam-bien-nhiet-do-va-do-am.0j8)
7. [Mạch sạc pin 1sell 18650 có bảo vệ TP4056](https://youtu.be/UDu_Q-xXEJE?si=sB-9SaQ0HPEEvYeD)
8. [Công nghệ LoRa và ứng dụng trong nông nghiệp công nghệ cao](https://tapchikhcn.haui.edu.vn/media/29/uffile-upload-no-title29282.pdf)
9. [Effect of Weather Condition on LoRa IoTCommunication Technology in a Tropical Region: Malaysia](https://www.researchgate.net/publication/351596875_Effect_of_Weather_Condition_on_LoRa_IoT_Communication_Technology_in_a_Tropical_Region_Malaysia)
10. [Tác động nhiệt độ trong LoRaWAN — Một nghiên cứu điển hình ở miền bắc Thụy Điển](https://www.mdpi.com/1424-8220/19/20/4414#B25-sensors-19-04414)
11. [Nghiên cứu và thiết kế hệ thống trạm giám sát thời tiết từ xa](https://tudonghoangaynay.vn/nghien-cuu-va-thiet-ke-he-thong-tram-giam-sat-thoi-tiet-tu-xa-7204.html)