**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**&**

**TRUYỀN THÔNG VIỆT HÀN**



**ĐỒ ÁN CHUYÊN NGÀNH 2**

**XÂY DỰNG HỆ THỐNG IOT TRONG NUÔI THỦY SẢN.**

|  |  |
| --- | --- |
| Ssssssssd Sinh viên thực hiện: | Nguyễn Thành Quốc. |
| MSV: | 21CE109. |
| GVHD: | TS.Nguyễn Tuấn Anh. |

Đà Nẵng, năm 2024

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**&**

**TRUYỀN THÔNG VIỆT HÀN**



**ĐỒ ÁN CHUYÊN NGÀNH 2**

**XÂY DỰNG HỆ THỐNG IOT TRONG NUÔI THỦY SẢN.**

|  |  |
| --- | --- |
| Ssssssssd Sinh viên thực hiện: | Nguyễn Thành Quốc. |
| MSV: | 21CE109. |
| GVHD: | TS.Nguyễn Tuấn Anh. |

Đà Nẵng, năm 2024

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

**LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN.**

|  |  |  |
| --- | --- | --- |
| **Stt** | **Thời gian** | **Nội dung làm việc** |
| 1 | Từ tuần 1.  (1-7/10/2024). | * Lập kế hoạch chi tiết. * Thu thập tài liệu. * Xây dựng mô hình kiến trúc hệ thống. |
| 2 | Từ tuần 2 đến tuần 8.  (11/10-24/11/2024). | * Lựa chọn phần cứng phù hợp với hệ thống. * Thiết kế mạch trên altium. * Thiết kế phần mềm. * Xây dựng app di động. * Cài đặt hệ thống. |
| 3 | Tuần 9  (25/11-30/11). | * Viết báo cáo. |

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành đến quý thầy cô và những người đã chân tình giúp đỡ em trong quá trình thực hiện đồ án của mình.

Đồ án chuyên ngành đã là một trong những trải nghiệm tuyệt vời trong hành trình học tập của em. Qua đó, em học được rất nhiều kiến thức và kỹ năng quan trọng trong chuyên ngành của mình.

Em muốn bày tỏ lòng biết ơn đến giảng viên hướng dẫn của mình đã dành thời gian, công sức của mình để hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện đồ án chuyên ngành. Những lời khuyên, hướng dẫn và phản hồi của Thầy Tuấn Anh đã giúp em hiểu sâu hơn về chuyên ngành cũng như giúp em hoàn thiện đồ án của mình một cách tốt nhất.

Một lần nữa, xin chân thành cảm ơn quý thầy cô và những người bạn, người anh đã giúp đỡ tôi trong quá trình thực hiện đồ án chuyên ngành của mình.

*Em xin chân thành cảm ơn*!

**MỤC LỤC**

[*LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN. ii*](#_Toc183849677)

[*LỜI CẢM ƠN iii*](#_Toc183849678)

[*MỤC LỤC iv*](#_Toc183849679)

[*DANH MỤC HÌNH ẢNH. vi*](#_Toc183849680)

[*DANH MỤC BẢNG BIỂU viii*](#_Toc183849681)

[*MỞ ĐẦU 1*](#_Toc183849682)

[*CHƯƠNG 1: TỔNG QUAN. 2*](#_Toc183849683)

[*1.1. ĐẶT VẤN ĐỀ. 2*](#_Toc183849684)

[*1.2. MỤC TIÊU VÀ NHIỆM VỤ. 2*](#_Toc183849685)

[*1.2.1. Mục tiêu và nhiệm vụ. 2*](#_Toc183849686)

[*1.2.2. Nhiệm vụ nghiên cứu. 3*](#_Toc183849687)

[*1.3. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU. 4*](#_Toc183849688)

[*1.3.1. Đối tượng nghiên cứu. 4*](#_Toc183849689)

[*1.3.2. Phạm vi nghiên cứu. 4*](#_Toc183849690)

[*1.4. PHƯƠNG PHÁP NGHIÊN CỨU. 4*](#_Toc183849691)

[*1.4.1. Khảo sát. 4*](#_Toc183849692)

[*1.4.2. Thực nghiệm. 4*](#_Toc183849693)

[*1.5. DỰ KIẾN KẾT QUẢ. 4*](#_Toc183849694)

[*1.5.1. Hệ thống Iot hoàn chỉnh. 4*](#_Toc183849695)

[*1.5.2. Phần cứng. 5*](#_Toc183849696)

[*1.5.3. App di động. 5*](#_Toc183849697)

[*1.6. Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN. 5*](#_Toc183849698)

[*1.6.1. Ý nghĩa khoa học. 5*](#_Toc183849699)

[*1.6.2. Ý nghĩa thực tiễn. 6*](#_Toc183849700)

[*1.7. BỐ CỤC. 6*](#_Toc183849701)

[*CHƯƠNG 2: CƠ SỞ LÝ THUYẾT. 7*](#_Toc183849702)

[*2.1. TỔNG QUAN VỀ IOT. 7*](#_Toc183849703)

[*2.1.1. Khái niệm. 7*](#_Toc183849704)

[*2.1.2. Kiến trúc Iot. 7*](#_Toc183849705)

[*2.2. CHUẨN GIAO TIẾP ONE-WIRE. 10*](#_Toc183849706)

[*2.3. CHUẨN GIAO TIẾP SPI. 12*](#_Toc183849707)

[*2.4. CHUẨN GIAO TIẾP UART. 13*](#_Toc183849708)

[*2.5. TỔNG QUAN VỀ FIREBASE. 14*](#_Toc183849709)

[*2.6. ARDUINO NANO. 14*](#_Toc183849710)

[*2.7. ESP32 WROM. 16*](#_Toc183849711)

[*2.8. TFT SPI MSP2402. 18*](#_Toc183849712)

[*2.9. GỚI THIỆU RELAY. 19*](#_Toc183849713)

[*2.10. PC817. 20*](#_Toc183849714)

[*2.11. HCSR-04. 21*](#_Toc183849715)

[*2.12. DS18B20. 23*](#_Toc183849716)

[*2.13. CẢM BIẾN ĐỘ ĐỤC. 24*](#_Toc183849717)

[*2.14. CẢM BIẾN PH4502C. 24*](#_Toc183849718)

[*CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ 26*](#_Toc183849719)

[*3.1. GIỚI THIỆU. 26*](#_Toc183849720)

[*3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG. 26*](#_Toc183849721)

[*3.2.1. Thiết kế sơ đồ khối hệ thống. 26*](#_Toc183849722)

[*3.2.2. Tính toán và thiết kế mạch. 27*](#_Toc183849723)

[*3.2.3. Biểu đồ usecase. 35*](#_Toc183849724)

[*3.2.4. Mối quan hệ giữa độ hòa tan oxy(DO) và nhiệt độ nước. 36*](#_Toc183849725)

[*3.2.5. Sơ đồ nguyên lý toàn mạch. 39*](#_Toc183849726)

[*CHƯƠNG 4: THI CÔNG HỆ THỐNG. 40*](#_Toc183849727)

[*4.1. GIỚI THIỆU 40*](#_Toc183849728)

[*4.2. THI CÔNG BO MẠCH 40*](#_Toc183849729)

[*4.3. LẬP TRÌNH HỆ THỐNG. 42*](#_Toc183849730)

[*4.3.2. App di động. 46*](#_Toc183849731)

[*CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 53*](#_Toc183849732)

[*5.1. KẾT LUẬN 53*](#_Toc183849733)

[*5.2. HƯỚNG PHÁT TRIỂN 53*](#_Toc183849734)

[*TÀI LIỆU THAM KHẢO 54*](#_Toc183849735)

**DANH MỤC HÌNH ẢNH.**

[Hình 2.1. Kiến trúc và nguyên tắc thiết kế hệ thống iot. 8](#_Toc183849171)

[Hình 2.2. Giao tiếp one-wire 11](#_Toc183849172)

[Hình 2.3. Giao tiếp one-wire 12](#_Toc183849173)

[Hình 2.4. Chuẩn giao tiếp SPI 13](#_Toc183849174)

[Hình 2.5. Giao tiếp nhiều thiết bị 13](#_Toc183849175)

[Hình 2.6. Chuẩn giao tiếp UART 14](#_Toc183849176)

[Hình 2.7. Firebase cloud 14](#_Toc183849177)

[Hình 2.8. ESP32-WROOM 17](#_Toc183849178)

[Hình 2.9. Relay 21](#_Toc183849179)

[Hình 2.10. PC817 22](#_Toc183849180)

[Hình 2.11. HC\_SR04 23](#_Toc183849181)

[Hình 2.12. DS18b20 24](#_Toc183849182)

[Hình 2.13. Cảm biến độ đục nước 25](#_Toc183849183)

[Hình 2.14. PH4502C 26](#_Toc183849184)

[Hình 3.1. Sơ đồ khối hệ thống 27](#_Toc183849185)

[Hình 3.2. Mô hình hoạt động hệ thống. 28](#_Toc183849186)

[Hình 3.4. Khối nguồn hạ áp 5V 31](#_Toc183849187)

[Hình 3.5. Khối nguồn hạ áp 3V. 31](#_Toc183849188)

[Hình 3.5. Khối nút nhấn 32](#_Toc183849189)

[Hình 3.6. Khối buzzer 33](#_Toc183849190)

[Hình 3.7. Khối Sensor 34](#_Toc183849191)

[Hình 3.8. Khối hiển thị 34](#_Toc183849192)

[Hình 3.9. Khối esp32 35](#_Toc183849193)

[Hình 3.10. Khối điều khiển trung tâm 35](#_Toc183849194)

[Hình 3.11. Use case 36](#_Toc183849195)

[Hình 3.12. Sơ đồ nguyên lý hệ thống. 40](#_Toc183849196)

[Hình 4.1. Sơ đồ mạch in 41](#_Toc183849197)

[Hình 4.2. Mạch in 3D 42](#_Toc183849198)

[Hình 4.3. Lưu đồ giải thuật của arduino , cảm biến và thiết bị ngoại vi 44](#_Toc183849199)

[Hình 4.4. Lưu đồ giải thuật esp32 và arduino 45](#_Toc183849200)

[Hình 4.5. Lưu đồ giải thuật điều khiển thiết bị bằng app thông qua gateway 46](#_Toc183849201)

[Hình 4.6. Home page 47](#_Toc183849202)

[Hình 4.7. Giao diện thay đổi ngưỡng an toàn 48](#_Toc183849203)

[Hình 4.8. Giao diện trạng thái thiết bị. 49](#_Toc183849204)

[Hình 4.9. Giao diện realtime chart 50](#_Toc183849205)

[Hình 4.10. Biểu đồ tháng 51](#_Toc183849206)

[Hình 4.11.Giao diện xem thông báo vượt ngưỡng 52](#_Toc183849207)

[Hình 4.12. Thông báo vượt ngưỡng về mail người dùng. 53](#_Toc183849208)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.2. Bảng cấu hình chân arduino 16](#_Toc183849463)

[Bảng 2.3. Thông số kỹ thuật WROOM 18](#_Toc183849464)

[Bảng 2.4. Bảng thông số kỹ thuật màn hình TFT 19](#_Toc183849465)

[Bảng 2.5. Cấu hình chân TFT SPI2402 20](#_Toc183849466)

[Bảng 3.1. Bảng tham chiếu thể hiện mối quan hệ giữa nhiệt độ và độ hòa tan oxy. 39](#_Toc183849467)

[Bảng 4.1.Bảng linh kiện sử dụng trong hệ thống 43](#_Toc183849468)

**MỞ ĐẦU**

**Nông nghiệp đóng vai trò quan trọng trong đời sống con người, cung cấp lương thực, thực phẩm và tạo ra nguồn thu nhập ổn định cho nông dâng.** Tuy nhiên, ngành nông nghiệp hiện nay đang phải đối mặt với nhiều thách thức như biến đổi khí hậu, sâu bệnh hại, thiếu hụt lao động, v.v. Do đó, việc ứng dụng khoa học kỹ thuật vào sản xuất nông nghiệp là một giải pháp thiết yếu để nâng cao hiệu quả và năng suất nuôi trồng.

**Công nghệ IoT (Internet vạn vật) là một trong những công nghệ tiên tiến** đang được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả lĩnh vực nông nghiệp. Hệ thống IoT có thể giúp giám sát các yếu tố môi trường như nhiệt độ, độ ẩm, ánh sáng, độ pH,.... một cách tự động và liên tục. Nhờ vậy, người nông dân có thể dễ dàng nắm bắt tình trạng môi trường nuôi trồng và đưa ra các biện pháp chăm sóc phù hợp, kịp thời.

**Với nhận thức về tầm quan trọng của ứng dụng IoT trong nông nghiệp,** em đã thực hiện đề tài "Xây dựng hệ thống IoT trong nuôi thủy sản". Đề tài này nhằm mục đích nghiên cứu, thiết kế và xây dựng một hệ thống IoT có thể giúp giám sát và chăm sóc thủy sản, đồng thời kiểm soát các yếu tố môi trường tác động đến khả năng sinh trưởng và phát triển của thủy sản.

# CHƯƠNG 1: TỔNG QUAN.

* 1. **ĐẶT VẤN ĐỀ.**

Xuất thân trong một gia đình thuần nông, em thấu hiểu sự vất vả của người nông dân. Nhớ lại những năm tháng trước đây, khi máy móc chưa phổ biến, công việc đồng áng chủ yếu dựa vào sức người, hiệu quả lao động thấp. Ví dụ như việc thu hoạch lúa, nông dân phải trải qua nhiều công đoạn tốn nhiều thời gian và sức lực. Tuy nhiên, sự ra đời của máy gặt đập đã cách mạng hóa ngành nông nghiệp, giúp giảm thiểu đáng kể sức lao động của người nông dân.

Điều này cho thấy, việc ứng dụng khoa học kỹ thuật vào sản xuất nông nghiệp là vô cùng quan trọng. Và nuôi trồng thủy sản cũng không phải là ngoại lệ. Đây là một ngành sản xuất đầy rủi ro, chịu ảnh hưởng lớn bởi các yếu tố môi trường và dịch bệnh.

Các yếu tố gây ra dịch bệnh trong nuôi trồng thủy sản rất đa dạng, có thể kể đến như: chất lượng nước kém do ô nhiễm, sự thay đổi đột ngột của các yếu tố môi trường, mật độ nuôi quá cao, nguồn giống bệnh và quản lý vệ sinh kém. Những yếu tố này không chỉ gây thiệt hại về kinh tế mà còn ảnh hưởng đến an toàn thực phẩm và môi trường.

**Vậy làm thế nào để giảm thiểu rủi ro dịch bệnh và nâng cao hiệu quả sản xuất trong nuôi trồng thủy sản?** Đây chính là vấn đề mà em muốn tìm hiểu và giải quyết thông qua đề tài này.

* 1. **MỤC TIÊU VÀ NHIỆM VỤ.**

**1.2.1. Mục tiêu và nhiệm vụ.**

**1.2.1.1. Mục tiêu chung.**

* **Xây dựng hệ thống Iot hiệu quả nhắm giám sát và điểu khiển môi trường nuôi thủy sản, nhắm nâng cao năng suất và chất lượng thủy sản.**
* **Giảm thiểu ruổi ro dịch bệnh, giảm thiểu chi phí sản xuất và tăng lợi nhuận cho người nuôi thủy sản.**

**1.2.1.2. Mục tiêu cụ thể.**

* **Về phần cứng: thiết mạch điện tử bao gồm vi điều khiển trung tâm, các thiết bị cảm biến, và các linh kiện điện tử khác.**
* **Về phần mềm: App di động dùng để đo lường và kiểm soát các thông số môi trường trong ao nuôi.**

**1.2.2. Nhiệm vụ nghiên cứu.**

**1.2.2.1. Nghiên cứu tiền đề.**

* **Thu thập thông tin về các hệ thống Iot hiện có trong nuôi trồng thủy sản.**
* **Xác định các loại cảm biến phù hợp để đo các thông số môi trường.**

**1.2.2.2. Thiết kế hệ thống.**

* **Thiết kế kiến trúc hệ thống Iot.**
* **Thiết kế mạch điện tử trên altium.**
* **Phát triển app di động người dùng.**

**1.2.2.3. Triển khai hệ thống.**

* **Lắp đặt và cài đặt hệ thống.**
* **Thu thập và xử lý dữ liệu.**
  1. **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU.**

**1.3.1. Đối tượng nghiên cứu.**

* **Hệ thống Iot: Các thiết bị cảm biến, vi điều khiển, app di động, firebase.**
* **Muôi trường nuôi trồng thủy sản: Ao nuôi, các thông số môi trường(nhiệt độ, độ pH, độ đục,...).**
* **Dữ liệu: Dữ liệu thu thập từ cảm biến,...**
* **Người nuôi thủy sản: Kiến thức, kinh nghiệm và ý kiến của người nuôi trồng về hệ thống Iot.**

**1.3.2. Phạm vi nghiên cứu.**

* **Thời gian: 2 tháng (1/10/2024 – 1/12/2024).**
* **Không gian: Ao nuôi thủy sản.**
* **Đối tượng: Thủy sản được nuôi trong ao, lồng.**
  1. **PHƯƠNG PHÁP NGHIÊN CỨU.**

**1.4.1. Khảo sát.**

* **Khảo sát người nuôi: Thu thập thông tin về kiến thức, kinh nghiệm và đánh giá của người nuôi thủy sản về hệ thống Iot.**

**1.4.2. Thực nghiệm.**

* **Xây dựng và triển khai hệ thống: Thiết kế, lắp đặt và vận hành hệ thống Iot tại một ao nuôi.**
* **Thu thập dữ liệu: dữ liệu thu thập từ các cảm biến.**
  1. **DỰ KIẾN KẾT QUẢ.**

**1.5.1. Hệ thống Iot hoàn chỉnh.**

* **Một hệ thống Iot đước thiết kế và triển khai thành công.**
* **Hệ thống có khả năng thu thập dữ liệu, tự động hóa một số quá trình nuôi trồng (như sục khí oxi,...).**

**1.5.2. Phần cứng.**

* **Thu thập dữ liệu từ cảm biến**
* **Điều khiển tự động và thủ công các thiết bị (máy sục oxi, máy sưởi, máy lọc li tâm, ...).**
* **Còi báo động và led nhấp nháy khi các thông số (pH, độ đục, độ oxi, nhiệt độ,..) không nằm trong phạm vi cho phép.**
* **Hiển thị dữ liệu thu được từ cảm biến lên LCD.**

**1.5.3. App di động.**

* **Hiển thị biểu đồ thống kê theo tháng.**
* **Tin nhắn cảnh báo khi các thông số như (pH, độ đục, nhiệt độ, độ oxi, ...) không nằm trong phạm vi cho phép.**
* **Hiển thị trạng thái hoạt động của từng thiết bị.**
* **Điều khiển tự động và thủ công các thiết bị (máy sục oxi, máy sưởi, máy lọc li tâm...).**
  1. **Ý NGHĨA KHOA HỌC VÀ THỰC TIỄN.**

**1.6.1. Ý nghĩa khoa học.**

* **Các dữ liệu được thu thập được từ hệ thống Iot có thể sử dụng để phân tích về mối quan hệ giữa các yếu tố môi trường và sinh trưởng của thủy sản.**

**1.6.2. Ý nghĩa thực tiễn.**

* **Hệ thống Iot giúp theo dõi liên tục các thông số môi trường, phát hiện sớm các yếu tố bất thường và kiệp thời điều chỉnh.**
* **Tự động hóa quá trình chăm sóc thủy sản, giúp tiết kiệm thời gian và công sức.**
* **Duy trì chất lượng nước ổn định, giúp thủy sản có môi trường sống tốt, từ đó nâng cao chất lượng sản phẩm.**
* **Người nuôi sử dụng hiệu quả các nguồn lực như điện, nước.**
* **Giảm thiểu rủi ro dịch bệnh.**
* **Tự động hóa giúp giảm thiểu lượng chất thải và hóa chất sử dụng trong quá trình nuôi trồng.**
  1. **BỐ CỤC.**

Chương 1: Tổng quan.

Chương 2: Cơ sở lý thuyết.

Chương 3: Tính toán và thiết kế.

Chương 4: Thi công hệ thống.

Chương 5: Kết luận và hướng phát triển.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.

**2.1. TỔNG QUAN VỀ IOT.**

**2.1.1. Khái niệm.**

Internet of thing là khái niệm cho phép giao tiếp giữa các thiết bị và application thông qua mạng internet, giúp các thiết bị phần cứng có thể truyền thông tin, theo dõi, kiểm soát và giám sát từ xa. Ban đầu, Iot dựa trên các thiết bị nhận diện như RFID. Khái niệm Iot cho phép theo dõi vị trí GPS, giao tiếp máy-máy(M2M), xe kết nối, thiết bị đeo thông minh, và ứng dụng trong công nghiệp 4.0. Iot đã biến các thành phố thông minh thành hiện thực và hứa hẹn phát triển xe tự lái trong tương lai.

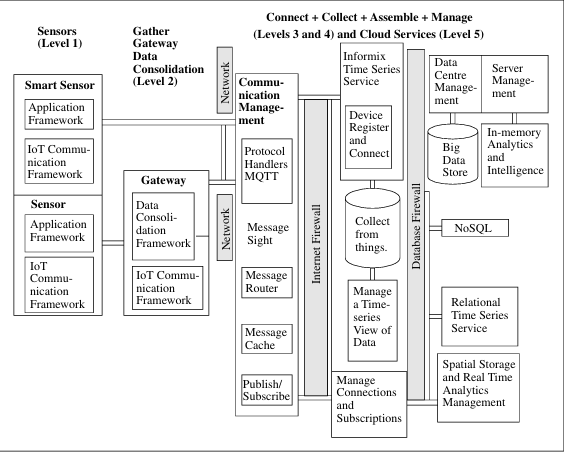
**2.1.2. Kiến trúc Iot.**

**2.1.2.1.Khái niệm khung Iot(Iot conceptual framework).**

* Là khái niệm mô tả cách mà các đối tượng vật lý tương tác và giao tiếp với nhau thông qua internet.
* Một khung Iot cơ bản bao gồm:

Physical objects + Controller, Sensor and Actuators + internet = internet of thing.

**2.1.2.2. Kiến trúc và nguyên tắc thiết kế.**

****

Hình 2.1. Kiến trúc và nguyên tắc thiết kế hệ thống iot.

**a) Sensors(level 1)**

* **Smart Sensor:** các cảm biến thông minh có khả năng thu thập dữ liệu từ môi trường xung quanh, ví dụ như nhiệt độ, độ ẩm, ánh sáng, chuyển động. Có khả năng xử lý dữ liệu ngay tại chỗ, lọc nhiễu, so sánh các giá trị ngưỡng,.. Có thể truyền dữ liệu đã được xử lý đến các thiết bị khác.
* **Sensor:** chỉ đơn thuần thu thập dữ liệu về một hoặc một số ít thông số vật lý như bhieetj độ, độ ẩm, áp suất, ... bằng cách chuyển đổi tín hiệu cảm biến thành tín hiệu điện.

**b) Gather Data Consolidation(level2)**

* **Gateway:** là cầu nối giữa các thiết bị Iot và hệ thống đám mây. Gateway có nhiệm vụ thu thập dữ liệu từ cảm biến, xử lý sơ bộ và chuyển dữ liệu lên đám mây.

**c) Connect + Collect + Assemble + Manage(level 3 và level4)**

* **Network communication management:** quản lý giao tiếp mạng giữa các thiết bị Iot và hệ thống trung tâm.
* **Protocol Handlers:** xử lý các giao thức truyền thông như MQTT, HTTPs,... để đảm bảo việc trao đổi dữ liệu diễn ra một cách hiệu quả và an toàn.
* **Device Register and connect:** quản lý việc đăng ký và kết nối các thiết bị Iot vào hệ thống.
* **Collect from things:** thu thập dữ liệu từ các thiết bị Iot.

**d) Cloud Services(level5)**

* **Time Series Service**: Lưu trữ và quản lý dữ liệu theo chuỗi thời gian, giúp theo dõi sự thay đổi của các biến số theo thời gian.
* **Data Centre Management:** Quản lý trung tâm dữ liệu, bao gồm việc lưu trữ, bảo mật và truy xuất dữ liệu.
* **Server Management:** Quản lý các máy chủ trong hệ thống, đảm bảo hoạt động ổn định và hiệu quả.
* **Big Data Store:** Lưu trữ một lượng lớn dữ liệu từ các thiết bị IoT.
* **In-memory Analytics and Intelligence:** Xử lý dữ liệu một cách nhanh chóng và đưa ra các phân tích, dự đoán.
* **NoSQL:** Cơ sở dữ liệu không quan hệ, linh hoạt để lưu trữ dữ liệu có cấu trúc không cố định.
* **Relational Time Series Service:** Cơ sở dữ liệu quan hệ chuyên dụng cho dữ liệu chuỗi thời gian.
* **Spatial Storage and Real Time Analytics Management:** Lưu trữ và phân tích dữ liệu không gian, giúp theo dõi vị trí và các sự kiện xảy ra trong không gian.
* **Manage Connections and Subscriptions:** Quản lý các kết nối và đăng ký giữa các thiết bị và dịch vụ.

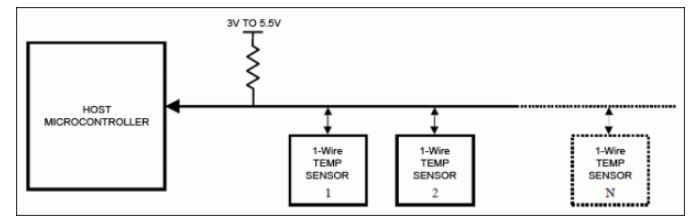
**2.2. CHUẨN GIAO TIẾP ONE-WIRE.**

Chuẩn giao tiếp one-wire được thiết kế và phát triển bởi Dallas Semicondutor. One-wire dùng một dây để truyền nhận dữ liệu ở tốc độ thấp.

Chuẩn giao tiếp one-wire là chuẩn giao tiếp không đồng bộ và bán song công(half-duplex). One-wire chỉ sử dụng một dây nối nguồn và truyền dữ liệu, nên khi không có dữ liệu trên đường truyền thì nó phải ở mưc logic cao, do đó cần kết nối dây này với nguồn thông qua một điện trở kéo lên.

One-wire sử dụng các cổng logic CMOS/TTL, tương ứng với mức logic 0 thì điện áp đỉnh ở mức 0.8V và điện áp tối thiểu cho mức logic 1 là 2.2V. Các thiết bị one-wire có nguồn cấp trong khoảng 2.8V đến 6V.

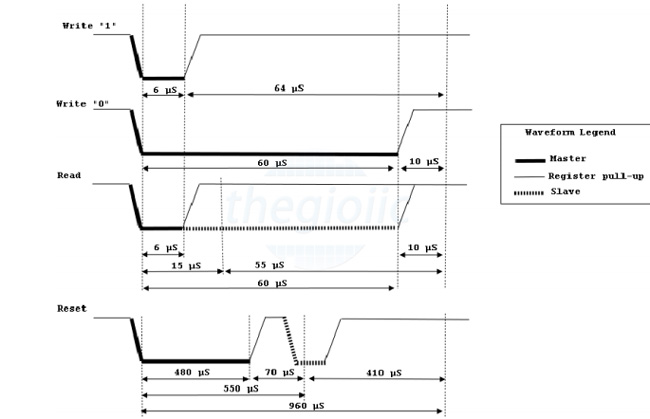
One-wire tuân thủ theo mô hình master-slave. Trên một đường truyền chỉ có một master duy nhất, và có thể có nhiều slave.



Hình 2.2. Giao tiếp one-wire

One – wire có 3 trạng thái hoạt dộng là cơ bản là Read, Write, Reset/Present.

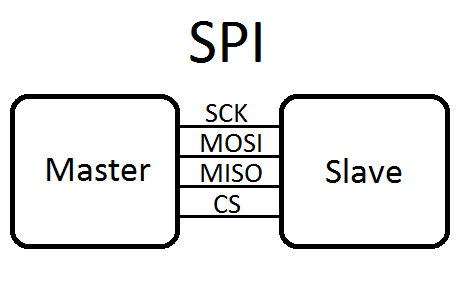
* Ở trạng thái read master sẽ kéo tín hiệu xuống mức logic thấp trong khoảng 0-15µs. Nếu slave muốn gửi bit 1 sẽ giải phóng đường truyền về mức cao, gửi bit 0 slave sẽ giữ đường truyền ở mức thấp trong khoảng thời gian 15µs đến 60µs.
* Ở trạng thái write, với bit 1 master sẽ kéo trạng thái đường truyền xuống thấp trong khoảng 1 đến 15µs, sau đó sẽ giải phóng đường truyền về mức cao. Với bit 0 master sẽ kéo trạng thái đường truyền xuống thấp trong khoảng 60µs đến 120µs, sau đó sẽ giải phóng đường truyền về mức cao. Giữa các lần gửi bit sẽ có khoảng thời gian nghỉ tối thiểu là 1µs.
* Ở trạng thái Reset/Present, master kéo tín hiệu xuống mức thấp trong khoảng thơi gian 480µs đến 640µs. Khoảng thời gian này được gọi là khoảng reset. Sau khoảng thời gian này, nếu có slave được kết nối vào mạng thì slave sẽ trả về tín hiệu present, khi này slave kéo tín hiệu xuống mức logic thấp trong khoảng thời gian 60µs đến 240µs.



Hình 2.3. Giao tiếp one-wire.

**2.3. CHUẨN GIAO TIẾP SPI.**

SPI(Serial Peripheral Interface) là một chuẩn truyền thông đồng bộ, được dùng để giao tiếp giữa các thiết bị trong hệ thống điện tử. Nó cho phép truyền dữ liệu giữa một thiết bị master và nhiều thiết bị slave thông qua các đường tín hiệu chung.

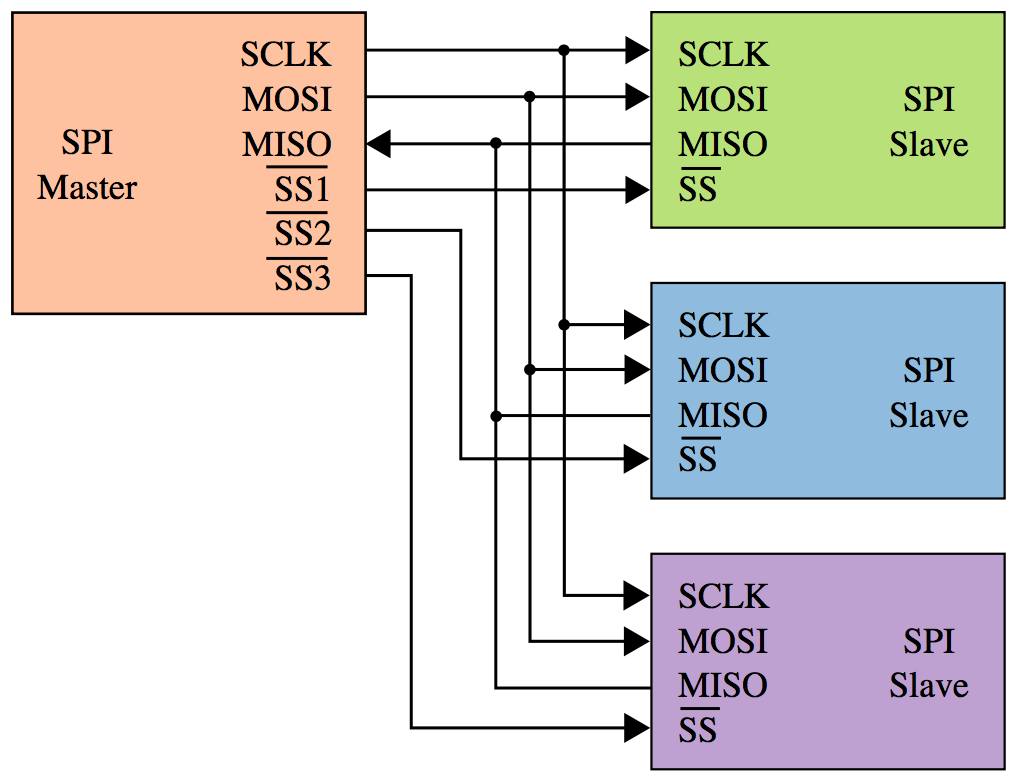


Hình 2.4. Chuẩn giao tiếp SPI

**Các thành phần giao tiếp SPI**

* **SCLK(Serial Clock):** Tín hiệu clock điều khiển đồng bộ.
* **MOSI(Master out slave In):** Tín hiệu dữ liệu được truyền từ thiết bị master tới một thiết bị slave.
* **MISO(Master in slave out):** Tín hiệu dữ liệu được truyền từ thiết bị slave về thiết bị master.
* **SS/CS(slave select/chip select):** Tín hiệu được sử dụng để chọn thiết bị slave nào tham gia truyền thông.

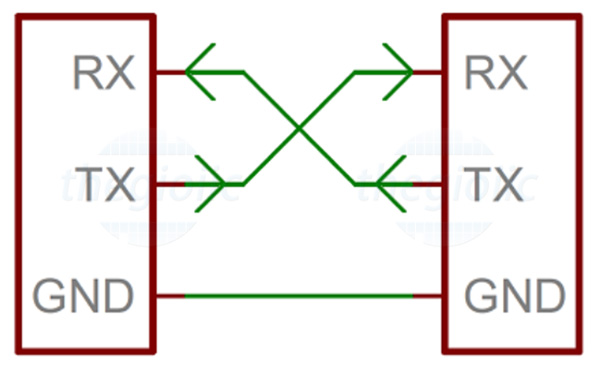
Chuẩn giao tiếp SPI hoạt động dựa trên việc truyền dữ liệu tuần tự giữa một thiết bị master và một hoặc thiết bị slave. Giao tiếp SPI sử dụng một tín hiệu clock chung và các tín hiệu truyền dữ liệu đông thời.



Hình 2.5. Giao tiếp nhiều thiết bị

**2.4. CHUẨN GIAO TIẾP UART.**

UART(universal asynchronous receive-transmitter) là bộ truyền nhận dữ liệu bất đồng bộ có thể cấu hình được tốc độ.



Hình 2.6. Chuẩn giao tiếp UART

**Các chế độ truyền trong uart**

* **Simplex:** Chỉ tiến hành truyền nối tiếp một chiều.
* **Half duplex:** Dữ liệu sẽ đi theo một hướng tại một thời điểm.
* **Full duplex:** Thực hiện giao tiếp đồng thời từ mỗi master và slave.

**2.5. TỔNG QUAN VỀ FIREBASE.**

Firebase là một dịch vụ cơ sở dữ liệu được hoạt động trên nền tảng cloud và được sử dụng trên hệ thống máy chủ Google, là nền tảng để phát triển ứng dụng di động và trang web nhanh chón, dễ dàng.



Hình 2.7. Firebase cloud

**2.6. ARDUINO NANO.**

Arduino Nano là một bo mạch vi điều khiển nhỏ gọn và mạnh mẽ, phù hợp cho các dự án điện tử, robot và IoT. Nó được thiết kế với kích thước nhỏ, tối ưu cho các ứng dụng yêu cầu tiết kiệm không gian. Bo mạch sử dụng chip vi điều khiển ATmega328P (phiên bản phổ biến) hoặc ATmega168 trong các dòng cũ hơn.

**Thông số kỹ thuật**

|  |  |
| --- | --- |
| Vi điều khiển | Atmega328P-8bit |
| Điện áp hoạt động | 5V |
| Điện áp đầu vào khuyến nghị | 7V-12V |
| Analog pins | 8(A0-A5) |
| Digital I/O pins | 14 |
| Dòng điện trên các chân I/O | 40mA |
| Dòng điện trên chân 3V3 | 50mA |
| Bộ nhớ Flash | 32KB |
| SRAM | 2KB |
| EEPROM | 1KB |
| Tần số | 16MHz |
| Các chuẩn kết nối | IIC, SPI, UART. |

Bảng 2.1. Thông số kỹ thuật arduino

**Cấu hình sơ đồ chân.**

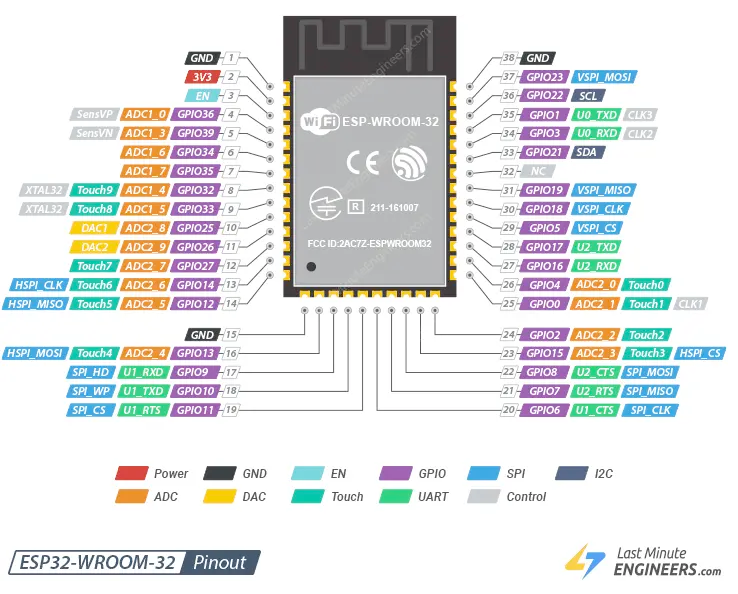
|  |  |  |
| --- | --- | --- |
| **Pin** | **Pin name** | **Mô tả** |
| Power | Vin,3.3V,5V,GND | * Vin: điện áp đầu vào cho arduino sử dụng nguồn ngoài(6-12V). * 5V: nguồn điện quy định dùng để cấp nguồn cho các thành phần khác. * 3.3V được tạo ra bởi bộ chỉnh áp trên arduino. * GND: nối đất. |
| Reset | Reset | Reset vi điều khiển. |
| Analog pins | A0-A7 | Dùng để đo điện áp analog trong khoảng 0 đến 5V. |
| I/O pins | D0-D13 | Dùng làm chân input hoặc output. |
| Serial | Rx,Tx | Dùng để gửi và nhận dữ liệu nối tiếp(UART). |
| Ngắc ngoài. | 2,3 | Kích hoạt ngắc. |
| PWM | 3, 5, 6, 9, 11 | Đầu ra PWM 8bit. |
| SPI | 10(SS),11(MOSI), 12(MISO),13 (SCK) | Dùng để giao tiếp SPI. |
| Led | 13 |  |
| IIC | A4(SDA),A5(SCA) |  |
| AREF | AREF | Tham chiếu điện áp đầu vào. |

Bảng 2.2. Bảng cấu hình chân arduino

**2.7. ESP32 WROM.**

ESP32-WROOM là một module vi điều khiển đa năng và mạnh mẽ, được phát triển bởi Espressif Systems. Đây là một trong những module phổ biến nhất trong lĩnh vực Internet of Things (IoT) và các ứng dụng thiết bị nhúng, nhờ khả năng tích hợp Wi-Fi, Bluetooth và các tính năng xử lý tiên tiến trong một thiết kế nhỏ gọn.

Module ESP32-WROOM được thiết kế để đáp ứng các nhu cầu đa dạng, từ việc điều khiển thiết bị thông minh trong nhà đến các hệ thống công nghiệp phức tạp. Với khả năng xử lý mạnh mẽ, giao tiếp linh hoạt và hiệu suất tiêu thụ năng lượng tối ưu, ESP32-WROOM đã trở thành lựa chọn hàng đầu cho các nhà phát triển trên toàn cầu.



Hình 2.8. ESP32-WROOM

**Thông số kỹ thuật**

|  |  |
| --- | --- |
| **Vi điều khiển** | **Wifi BLE SoC ESP32 ESP-WROOM-32** |
| Điện áp sử dụng | 2.2V~3.6VDC |
| Dòng điện sử dụng | ~90mA |
| Nhân xử lý trung tâm | ESP32-D0WDQ6 Dual-core low power Xtensa® 32-bit LX6 microprocessors. |
| ROM | 448KBytes |
| SRAM | 520 KBytes |
| WiFi | 802.11 b/g/n/d/e/i/k/r (802.11n up to 150 Mbps) |
| Bluetooth | Bluetooth v4.2 BR/EDR and BLE specification |
| Wi-Fi mode | Station/softAP/SoftAP+station/P2P |
| Bảo mật | WPA/WPA2/WPA2-Enterprise/WPS |
| Encryption | AES/RSA/ECC/SHA |
| Giao thức mạng | IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT |
| Interfaces | SD-card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S , IR, GPIO, capacitive touch sensor, ADC, DAC, Hall sensor, temperature sensor |

Bảng 2.3. Thông số kỹ thuật WROOM

**2.8. TFT SPI MSP2402.**

TFT SPI MSP2402 là một loại màn hình màu LCD nhỏ gọn, sử dụng công nghệ TFT (Thin Film Transistor) kết hợp với giao tiếp SPI (Serial Peripheral Interface), giúp hiển thị đồ họa và văn bản rõ nét trên không gian nhỏ. Với kích thước phổ biến là 2.4 inch, màn hình này thường được sử dụng trong các dự án nhúng, như hệ thống hiển thị dữ liệu, điều khiển robot, hoặc các ứng dụng IoT.

TFT SPI sử dụng driver **ILI9341** hoặc tương đương, hỗ trợ nhiều thư viện mã nguồn mở trên Arduino IDE, STM32CubeIDE, và các nền tảng phát triển khác.

Chuẩn giao tiếp SPI tốc độ cao được sử dụng trong TFT giúp truyền dữ liệu nhanh, phù hợp với các vi điều khiển phổ biến như Arduino, STM32, ESP32.

**Thông số kỹ thuật**

|  |  |
| --- | --- |
| **Tên** | **Mô tả** |
| Display color | RGB 65K màu |
| SKU | MSP2402 |
| Kích thước | 2.4 inch |
| Kiểu | TFT |
| Driver IC | ILI9341 |
| Resolution | 320\*240 pixel |
| Chuẩn giao tiếp | 4-wire SPI |
| Active Area | 36.72x48.96(mm) |
| PCB size | 77.18x42.72(mm) |
| Nhiệt độ hoạt động | -20 đến 70 độ C |
| Nhiệt độ bảo quảng | -30 đến 80 độ C |
| Điện áp hoạt động | 3.3V/5V |
| Dòng | 90mA |
| Trọng lượng | 25g |

Bảng 2.4. Bảng thông số kỹ thuật màn hình TFT

**Cấu hình sơ đồ chân**

|  |  |
| --- | --- |
| **Pins** | **Mô tả** |
| VCC | Điện áp hoạt động LCD 3.3 đến 5V |
| GND | Nối đất |
| CS | Tín hiệu điều khiển lựa chọn slave |
| Reset | Tín hiệu reset LCD |
| DC/RS | Thanh ghi LCD/tín hiệu điều khiển lựa chọn dữ liệu |
| SDI(MOSI) | Tín hiệu ghi dữ liệu bus SPI |
| SCK | Tín hiệu clock điều khiển đồng bộ. |
| LED | Đèn nền LCD |
| SDO(MISO) | Bus LCD SPI đọc tín hiệu. |
| T\_CLK | Tín hiệu clock điều khiển đồng bộ cảm ứng |
| T\_CS | Chân kích hoạt cảm ứng |
| T\_DIN | Chân dữ liệu vào của bảng cảm ứng, nhận lệnh và thông tin từ vi điều khiển. |
| T\_DO | Chân dữ liệu ra của bảng cảm ứng |
| T\_IRQ | Chân ngắt thông báo sự kiện chạm. |

Bảng 2.5. Cấu hình chân TFT SPI2402

**2.9. GỚI THIỆU RELAY.**

Relay (Rơ-le) là một thiết bị chuyển mạch điện tử hoặc cơ học được sử dụng để điều khiển một mạch điện khác bằng cách sử dụng tín hiệu điều khiển nhỏ (thường từ một vi điều khiển hoặc cảm biến). Relay thường được ứng dụng trong các hệ thống tự động hóa, bảo vệ mạch, và điều khiển thiết bị điện công suất cao.

Relay đóng vai trò như một công tắc điện tự động, cho phép cách ly giữa mạch điều khiển (thường là điện áp thấp) và mạch tải (điện áp cao).



Hình 2.9. Relay

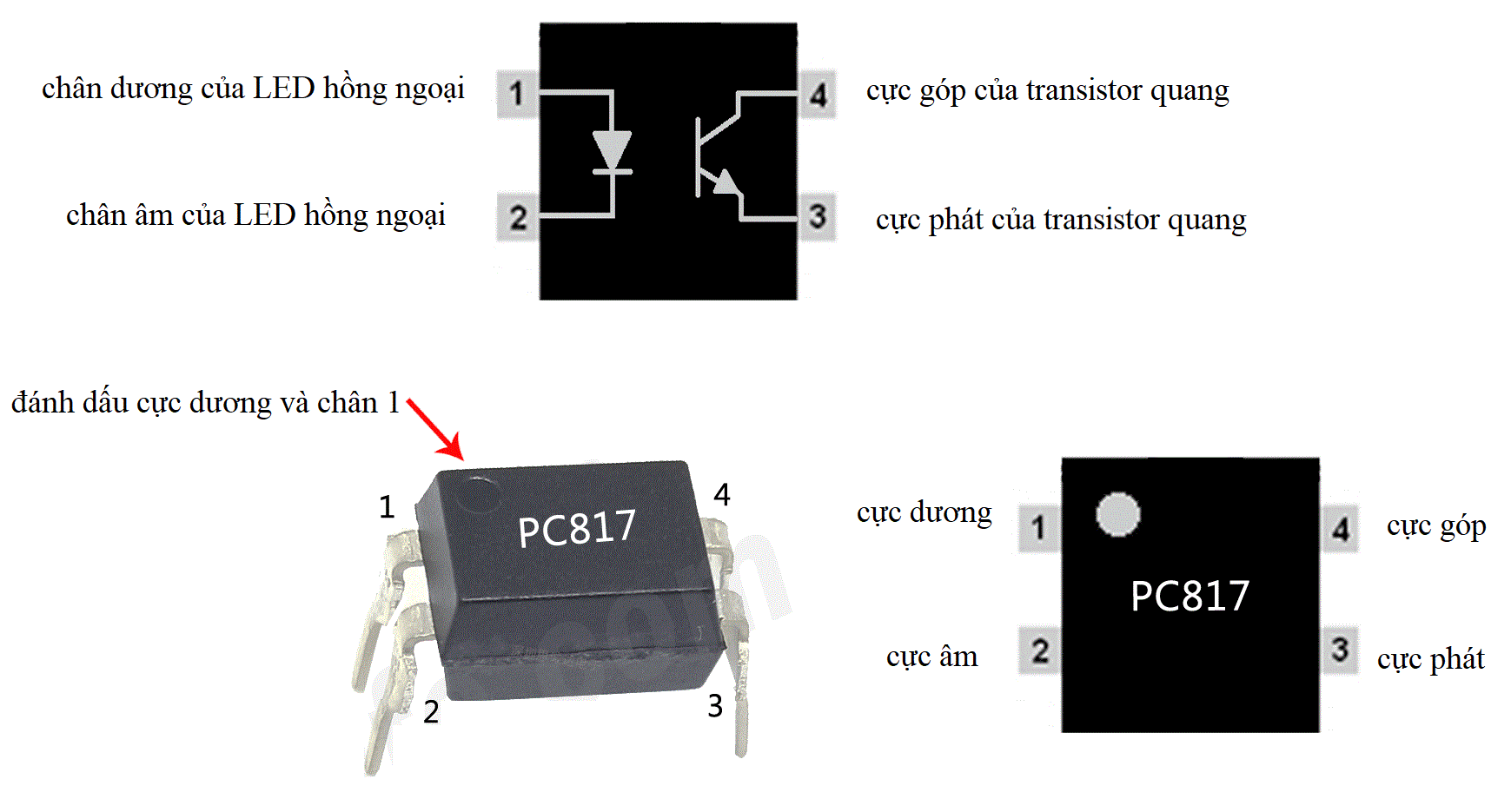
**Nguyên lý hoạt động**

* **Dòng điện qua cuộn dây:** lúc này nó tạo ra từ trường xung quanh lõi từ, từ trường này hút phần tiếp điểm động(armature), thay đổi trạng thái của các tiếp điểm(tiếp điểm NO đóng, tiếp điểm NC mở).
* **Dòng điện bị ngắt:** lúc này từ trường biến mất, là xo kéo tiếp điểm động về vị trí ban đầu.
* **Chuyển mạch:** Sự thay đổi trạng thái của các tiếp điểm cho phép điều khiển mạch tải lớn bằng tín hiệu nhỏ từ mạch điều khiển.

**2.10. PC817.**

PC817 là một loại optocoupler (còn gọi là photo-coupler hoặc optical isolator) được sử dụng phổ biến trong các mạch điện tử để cách ly tín hiệu giữa hai mạch điện. Nó sử dụng ánh sáng để truyền tín hiệu từ phía phát sang phía thu, giúp cách ly hoàn toàn về mặt điện giữa mạch điều khiển và mạch tải.

Optocoupler này được sản xuất bởi nhiều nhà sản xuất linh kiện điện tử, với đặc điểm nổi bật là kích thước nhỏ gọn, dễ sử dụng, và giá thành thấp, phù hợp cho các ứng dụng trong điều khiển công suất, mạch bảo vệ, và giao tiếp tín hiệu.



Hình 2.10. PC817

**Nguyên lý hoạt động**

* **Phía đầu vào:** Khi có dòng điện chạy qua led hồng ngoại ở phía phát, LED phát ra ánh sáng hồng ngoại.
* **Phía đầu vào:** Transistor quang ở phía thu nhận ánh sáng này, dẫn đến việc tạo dòng điện ở mạch đầu ra.
* **Cách ly điện:** Do tínhiệu được truyền qua ánh sáng, nên tín hiệu đầu vào và đầu ra được cách ly hoàn toàn.

**2.11. HCSR-04.**

HC-SR04 là một loại cảm biến siêu âm được sử dụng để đo khoảng cách với độ chính xác cao bằng cách sử dụng nguyên lý phản xạ âm thanh. Loại cảm biến này phổ biến trong các ứng dụng như robot tự hành, hệ thống đo lường tự động, và các thiết bị IoT nhờ chi phí thấp, thiết kế đơn giản, và hiệu quả ổn định.



Hình 2.11. HC\_SR04

**Nguyên lý hoạt động.**

* Chân Trig nhận tín hiệu HIGH trong khoảng 10µs, cảm biến sẽ phát sóng siêu âm với tần số 40KHz qua đầu phát.
* Chân Echo sẽ nhận tín hiệu HIGH từ đầu thu tín hiệu.
* Tính khoảng cách:

D: khoảng cách(cm).

T: thời gian(µs).

v: Vận tốc âm thanh(343m/s trong không khí).

**2.12. DS18B20.**

DS18B20 là một cảm biến nhiệt độ kỹ thuật số chính xác cao, giao tiếp qua giao thức 1-Wire. Đây là loại cảm biến phổ biến trong các ứng dụng đo lường nhiệt độ nhờ thiết kế đơn giản, khả năng đo chính xác và tích hợp trực tiếp với các vi điều khiển như Arduino, Raspberry Pi.

Cảm biến DS18B20 có thể hoạt động trong môi trường nhiệt độ từ -55°C đến +125°C và cung cấp dữ liệu nhiệt độ với độ phân giải từ 9 đến 12 bit.



Hình 2.12. DS18b20

**Nguyên lý hoạt động**

* DS18B20 giao tiếp qua giao thức 1-Wire, chỉ sử dụng một dây (chân DQ) để truyền dữ liệu. Tất cả giao tiếp được đồng bộ hóa bằng một vi điều khiển hoặc máy tính.
* Cảm biến sử dụng cảm biến bán dẫn bên trong để đo nhiệt độ môi trường, dữ liệu đo được chuyển đổi thành giá trị kỹ thuật số và gửi qua chân DQ.
* Người dùng có thể cấu hình độ phân giải từ 9 bit (ít chính xác nhưng nhanh) đến 12 bit (chính xác cao nhưng chậm hơn).
* Mỗi DS18B20 có một mã 64-bit duy nhất, cho phép sử dụng nhiều cảm biến trên cùng một đường dây dữ liệu mà không bị xung đột.

**2.13. CẢM BIẾN ĐỘ ĐỤC.**

Cảm biến độ đục nước là một loại cảm biến được sử dụng để đo mức độ đục hoặc sạch của nước. Nó hoạt động dựa trên nguyên lý tán xạ ánh sáng hoặc độ hấp thụ ánh sáng khi đi qua môi trường nước. Cảm biến này thường được sử dụng trong các ứng dụng như:

* + - Giám sát chất lượng nước trong ao nuôi, hồ cá, hoặc hệ thống xử lý nước.
    - Hệ thống quan trắc môi trường, đo độ đục trong sông, suối, và nước thải.
    - Ứng dụng công nghiệp để kiểm tra chất lượng dung dịch.

Hình 2.13. Cảm biến độ đục nước

**2.14. CẢM BIẾN PH4502C.**

PH4502C là một cảm biến pH analog, được thiết kế để đo độ pH trong môi trường nước. Cảm biến này có thể được sử dụng trong các ứng dụng như kiểm tra chất lượng nước, hồ nuôi cá, xử lý nước thải, hoặc các nghiên cứu hóa học. Cảm biến PH4502C cung cấp tín hiệu điện áp analog, tỉ lệ thuận với độ pH của mẫu nước, giúp dễ dàng tích hợp vào các hệ thống điều khiển tự động, bao gồm vi điều khiển như Arduino hoặc ESP32.



Hình 2.14. PH4502C

**Nguyên lý hoạt động**

* Cảm biến PH4502C hoạt động dựa trên nguyên lý đo điện thế giữa hai điện cực khi tiếp xúc với dung dịch có ion H+ (hydrogen ion). Mỗi ion H+ có khả năng thay đổi điện thế của điện cực, và từ đó, tín hiệu điện được sinh ra tương ứng với độ pH của dung dịch.

**CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ**

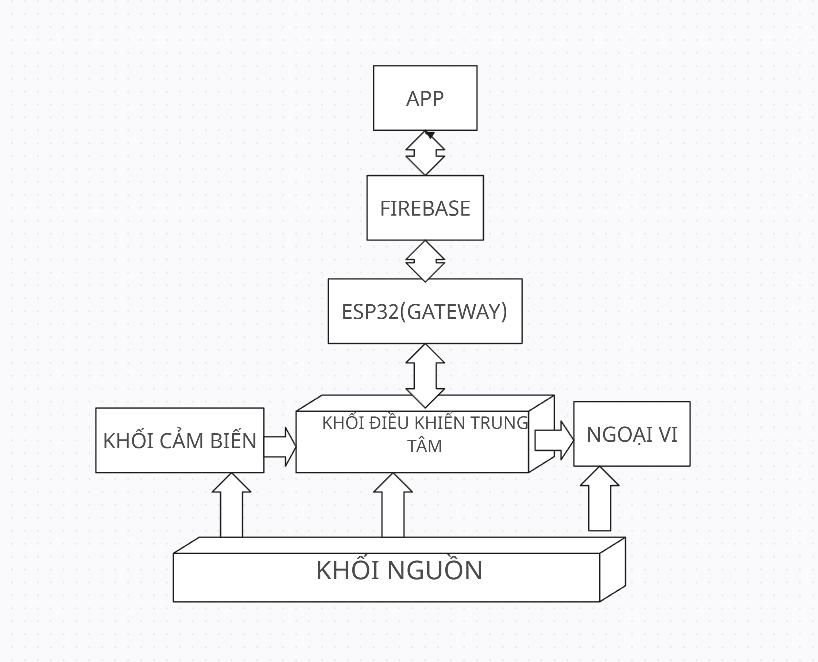
**3.1. GIỚI THIỆU.**

Đề tài “Xây dựng hệ thống iot trong nuôi thủy sản” gồm các khối MCU(arduino nano) và cảm biến để thu thập, xử lý các thông số như nhiệt độ, độ pH, mực nước, độ đục nước. Tại đây dữ liệu sẽ được khối MCU xử lý và gửi lên gateway(esp32) thông qua giao thức uart.

Tại gateway(esp32), dữ liệu sẽ được lưu trữ vào firebase và app di động sẽ lấy dữ liệu từ firebase hiển thị cho người dùng. Với việc hiển thị dữ liệu lên app, người dùng có thể giám sát tình trạng môi trường nuôi. Đồng thời người dùng có thể tự thiết lập các thông số ngưỡng, cũng như điều khiển các thiết bị.

**3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG.**

**3.2.1. Thiết kế sơ đồ khối hệ thống.**

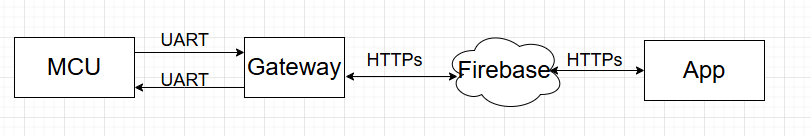
****

Hình 3.1. Sơ đồ khối hệ thống

**Chức năng từng khối**

* **Khối xử lý trung tâm:** là khối điều khiển chỉnh của hệ thống, khối này thu thập dữ và xử lý dữ liệu từ cảm biến, sau đó điều khiển các thiết bị ngoại vi. Các dữ liệu sau khi được xử lý sẽ gửi qua esp32(gateway) bằng chuẩn giao tiếp UART. Ngoài ra, khối xử lý trong tâm còn nhận dữ liệu từ gateway.
* **Khối cảm biến:**  gồm các cảm biến thu thập thông số môi trường như pH, nhiệt độ, độ đục, mực nước.
* **Khối thiết bị ngoại vi:** gồm các thiết bị điện được điều khiển bởi khối xử lý trung tâm, thông qua relay.
* **Khối Gateway(esp32):** khối này là cổng truyền nhận dữ liệu giữa khối xử lý trung tâm và khối firebase.
* **Khối firebase:** là cloud để lưu trữ dữ liệu.
* **Khối App:** là khối hiển thị thông số môi trường, trạng thái thiết bị, thiết lập và điều khiển các thiết bị.
* **Khối nguồn:** Cung cấp nguồn cho toàn bộ hoạt động của hệ thống.

**Mô hình hoạt động**

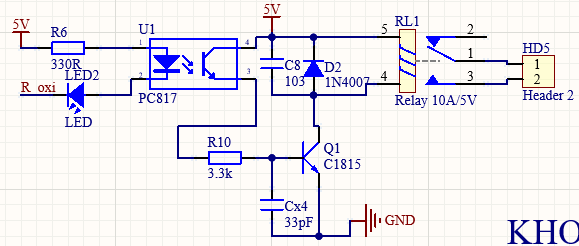
****

Hình 3.2. Mô hình hoạt động hệ thống.

**3.2.2. Tính toán và thiết kế mạch.**

**3.2.2.1. Khối relay điều khiển.**

Để điều khiển các thiết bị 220AC bằng điện áp thấp 5V, chúng ta dùng relay, đồng thời phải đảm bảo cách ly điện và bảo vệ giữa mạch điều khiển và thiết bị tải.



Hình 3.3. Khối relay điều khiển

Sử dụng pc817 để cách ly vi điều khiển với tải, đảm bảo an toàn. Pc817 hoạt động dựa trên led hồng ngoại, khi led hồng ngoại sáng làm cho phototransistor bên trong pc817 dẫn điện. Điều này cho phép dòng điện chạy qua pc817.

Dòng khuyến nghị của nhà sản xuất cho led hồng ngoại bên trong pc817 là 5mA đến 20mA.

Điện áp khuyến nghị của nhà sản xuất là từ 1.2V đến 1.4V. Khi đó điện trở cản dòng của pc817 sẽ là:

* Điện trở giá trị gần nhất với 350R là 330R, chọn = 330R.
* Vcc: nguồn cấp cho relay
* khoảng 1.2 đến 1.4 volt.
* điện áp bão hòa transistor khoảng 0.2V đến 0.3V.
* 5mA đến 20mA.

Theo nhà sản xuất, khi relay được cấp nguồn 5V thì điện áp qua cuộn dây là 71.4mA. Hệ số khuếch đại của transistor c1815 =100.

Để relay hoạt động, thì transistor ở trạng thái bão hòa(trạng thái bật hoàn toàn). Điều kiện bão hòa transistor:

() = = 0.714mA(1).

(2).

Từ (2) ta nhận thấy càng lớn thì càng nhỏ. Kết hợp với điều kiện để dòng điện bão hòa ở (1) thì chọn .

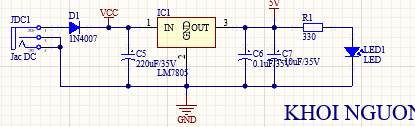
Hai tụ gốm C8 và Cx4 dùng để giảm nhiễu và ổn định tín hiệu.Vì khi relay hoạt động thì, thì cuộn dây có thể sinh ra các xung điện áp(spike) do hiện tượng cảm khán. Những xung này có khả năng gây nhiễu.

**3.2.2.2. Khối nguồn hạ áp.**

Trong hệ thống “Xây dựng hệ thống iot trong nuôi thủy sản”, chúng ta dùng nguồn 9V cung cấp cho arduino uno, nguồn 3.3V cung cấp cho màn hình TFT SPI, nguồn 5V cung cấp cho các linh kiện còn lại.

Để đảm bảo cung cấp nguồn hợp lý cho hệ thống, chúng ta cần thiết kế khối nguồn có đầu vào 9V và sử dụng hai IC tuyến tính là LM7805 và LM1117T để hạ áp lần lượt xuống 5V và 3.3V.

**Khối hạ áp 5V**



Hình 3.4. Khối nguồn hạ áp 5V

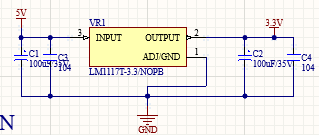
Trong mạch chúng ta sử dụng led để nhận biết trạng thái của nguồn. Led đỏ dùng dòng 10mA đến 20mA và điện áp rơi khoảng 1.8V đến 2.2V. Từ đây, ta có:

* Điện trở gần nhất với 300 là 330Ω, chọn .

1N4007 dùng dể chặn dòng ngược, bảo vệ các linh kiện không bị hỏng bởi dòng ngược, đồng thời giúp giảm nhiễu và cải thiện chất lượng nguồn DC.

Các tụ điện có trong mạch, được sử dụng để giảm nhiễu và cải thiện độ ổn định của điện áp.

**Khối hạ áp 3V**

****

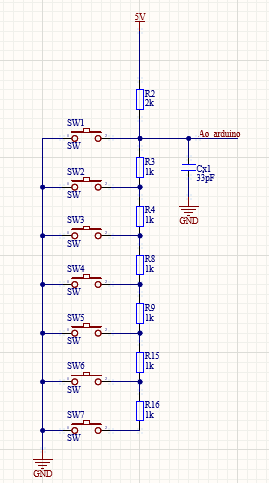
Hình 3.5. Khối nguồn hạ áp 3V.

Trong mạch trên chúng ta dùng LM1117, nhận điện áp 5V từ khối hạ áp 5V, để tiếp tục hạ áp xuống còn 3.3V.

Các tụ trong mạch dùng để lọc nhiễu và ổn định điện áp.

**3.2.2.3. Khối nút nhấn.**

Trong hệ thống này chúng ta sử dụng arduino nano để làm khối xử lý trung tâm. Chính vị vậy các chân I/O không nhiều để chúng ta sử dụng cho khối nút bấm. Vì thế , chúng ta chọn giải pháp dùng một chân arnalog để điều khiển nhiều nút nhận. Phương pháp này có đỗ trễ lớn hơn so với phương pháp sử dụng các chân I/O. Tuy nhiên, trong trường hợp này là phương án hợp lý khi các chân I/O của arduino nano không đủ.



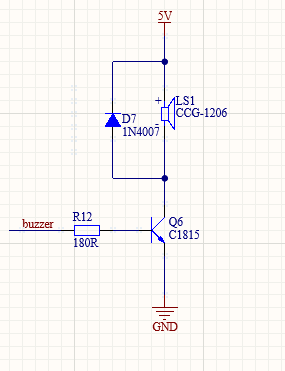
Hình 3.5. Khối nút nhấn

Trong mạch trên chúng ta sử dụng kỹ thuật chia áp đơn giản, kỹ thuật này giúp phân mức điện áp tại các nút nhấn( tại mỗi nút nhấn sẽ có mức điện áp khác nhau).

Dùng thiết bị voltage để đo điện áp từng nút khi nhấn, rồi quy đổi ra tín hiệu analog(0-1023). Lúc này ta sẽ có được từng mức logic tương ứng cho từng nút.

**3.2.2.4. Khối buzzer cảnh báo.**

Khi có các giá trị vượt ngưỡng, chúng ta sẽ dùng còi để cảnh báo ngưỡng nguy hiểm của các thông số môi trường.

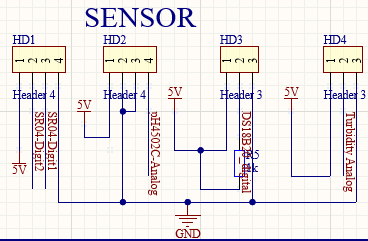


Hình 3.6. Khối buzzer

Dòng điện lớn nhất cho phép của buzzer theo nhà sản xuất là 50mA. Ta có hệ số khuếch đại của c1815 hoạt động của mức điện áp 5V là 71.4. Khi đó ta dễ dàng xác định được khoảng giá trị của .(Tính tương tự như mục 3.2.2.1).

**3.2.2.5. Khối cảm biến.**

Chúng ta dùng các cảm biến để đọc các thông số môi trường, giúp dễ dàng quản lý chất lượng môi trường nuôi thủy sản.



Hình 3.7. Khối Sensor

Cảm biến ds18b20 cần nối vơi điện trở kéo lên (pull-up) để duy trì tín hiệu ở mức logic HIGH. Để tránh tình trạng tín hiệu bị nhiễu, thường chúng ta nên chọn điện trở có giá trị từ 4.7kΩ đến 10 kΩ.

**3.2.2.6. Khối LCD hiển thị.**

Chúng ta sử dụng LCD tft SPI để hiển thị các thông số môi trường ảnh hưởng đến chất lượng nuôi thủy sản.

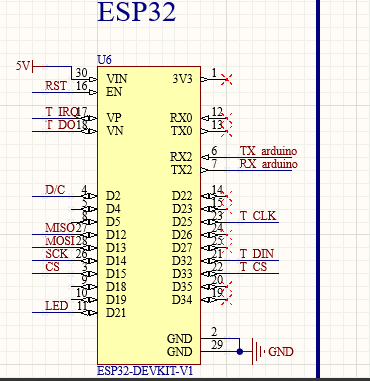


Hình 3.8. Khối hiển thị

TFT sẽ có điện áp hoặc động từ 3.3V đến 5V, và sử dụng chuẩn giao tiếp SPI.

**3.2.2.7. Khối gateway(esp32).**

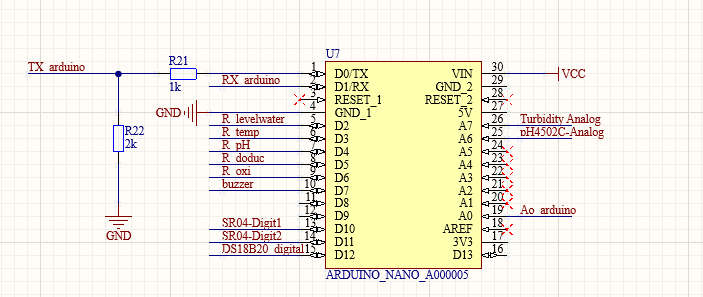
Khối này đóng vai trò là cổng kết nối giao tiếp giữa khối điều khiển trung tâm với cloud.



Hình 3.9. Khối esp32

**3.2.2.8. Khối điều khiển trung tâm.**

Đây là khối xử lý trung tâm của hệ thống, có nhiệm vụ đọc các thông số môi trường từ cảm biến, và điều khiển các thiết bị điện.

****

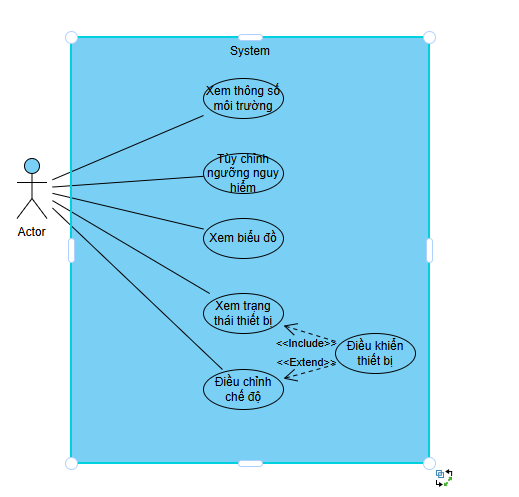
Hình 3.10. Khối điều khiển trung tâm

Đối với arduino nano, các chân hoạt động ở mức điện áp logic là 5V, tuy nhiên các chân của esp32 lại hoạt động ở mức logic là 3.3V. Chính về thế, khi arduino gửi tín hiệu 5V từ chân TX của mình đến chân RX esp32, mức tín hiệu này là quá cao so với mức 3.3V của esp32,để tránh làm hỏng chân RX của esp32 chúng ta cần chia áp xuống 3.3V.

* , *suy ra*: (1).

Chọn điện trở và điện trở sao cho thỏa điều kiện (1).

**3.2.3. Biểu đồ usecase.**

****

Hình 3.11. Use case

Ứng dụng di động này cho phép theo dõi và điều khiển các thiết bị trong nuôi thủy sản, đồng thời cho phép người dùng giám sát các thông số môi trường. Người dùng có thể thực hiện các chức năng như: Xem thông số môi trường, tùy chỉnh ngưỡng ngủy hiểm, xem biểu đồ, xem trạng thái hoạt động của thiết bị, điều khiển thiết bị, xem nhật ký cảnh báo.

**3.2.4. Mối quan hệ giữa độ hòa tan oxy(DO) và nhiệt độ nước.**

Sử dụng phương pháp hồi quy tuyến tính để ước lượng độ hòa tan Oxy(mối quan hệ trong môi trường nước ngọt).

**Trong đó:**

* X: giá trị của biến dự đoán(nhiệt độ nước ngọt).
* kết quả dự đoán(độ hòa tan oxy).
* hệ số góc.
* hệ số chặn.

Bảng liên hệ giữ nhiệt độ và độ hòa tan oxy trong môi trường nước ngọt.

|  |  |
| --- | --- |
| **Nhiệt độ()** | **DO(mg/l)** |
| 0 | 14.6 |
| 1.1 | 14.1 |
| 1.2 | 13.7 |
| 3.3 | 13.3 |
| 4.4 | 12.9 |
| 5.6 | 12.2 |
| 6.7 | 11.9 |
| 7.8 | 11.6 |
| 8.9 | 11.3 |
| 10 | 11 |
| 11.1 | 10.7 |
| 12.2 | 10.4 |
| 13.3 | 10.2 |
| 14.4 | 9.9 |
| 15.6 | 9.7 |
| 16.7 | 9.5 |
| 17.8 | 9.3 |
| 18.9 | 9.1 |
| 20 | 8.9 |
| 21.1 | 8.7 |
| 22.2 | 8.6 |
| 23.3 | 8.5 |
| 23.4 | 8.3 |
| 25.6 | 8.2 |
| 26.7 | 8 |
| 27.8 | 7.8 |
| 28.9 | 7.7 |
| 30 | 7.5 |
| 31.1 | 7.4 |
| 32.2 | 7.3 |
| 33.3 | 7.1 |
| 34.4 | 7 |
| 35.6 | 6.9 |
| 36.7 | 6.8 |
| 37.8 | 6.6 |
| 38.9 | 6.5 |
| 40 | 6.4 |
| 41.1 | 6.3 |
| 42.2 | 6.2 |
| 43.3 | 6.1 |
| 44.4 | 6 |
| 45.6 | 5.9 |

Bảng 3.1. Bảng tham chiếu thể hiện mối quan hệ giữa nhiệt độ và độ hòa tan oxy.

**Trung bình cộng của X(nhiệt độ nước):**

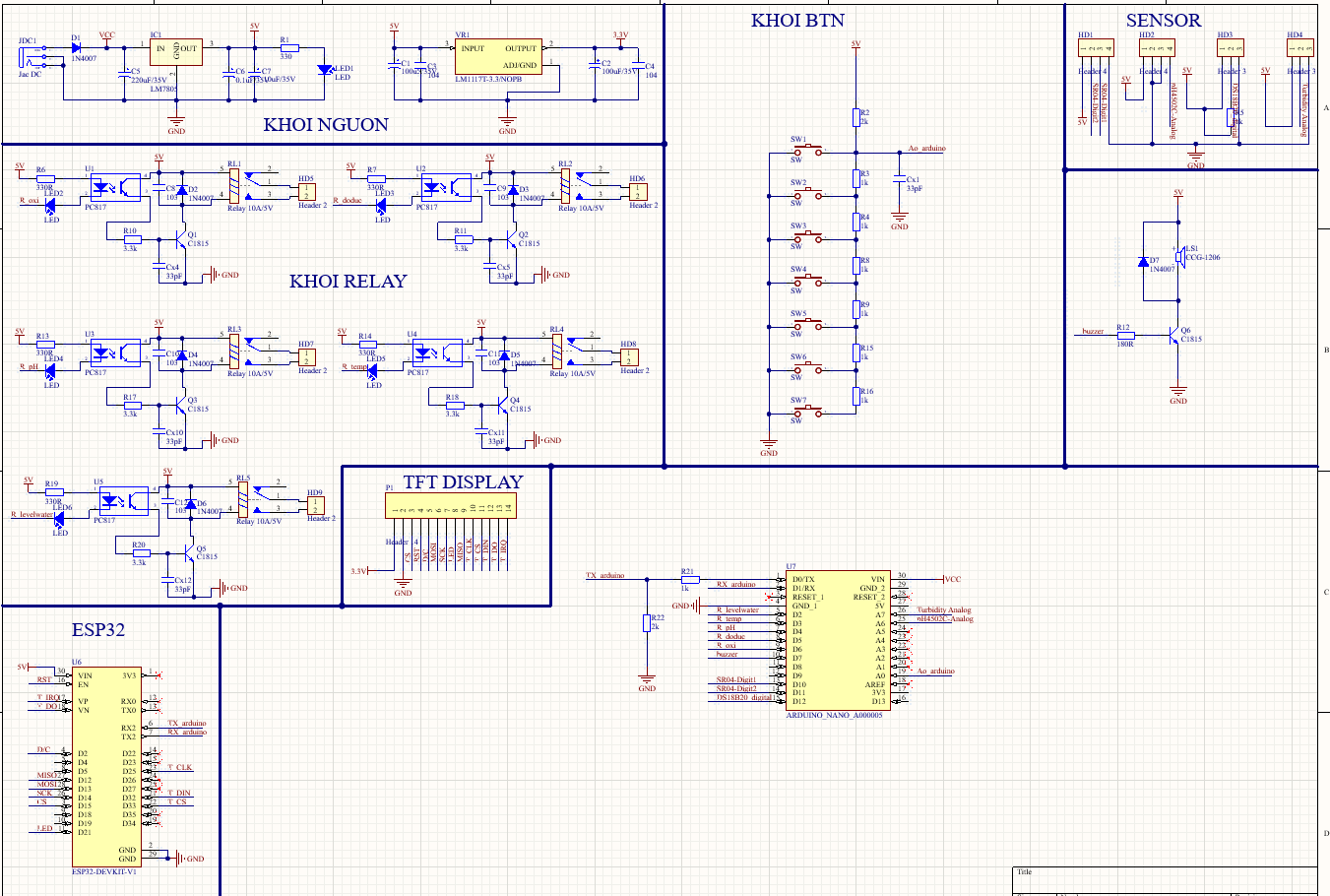
**Trung bình cộng của Y(độ hòa tan oxy trong nước).**



*Suy ra,* công thức thể hiện mối quan hệ giữa nhiệt độ và độ hòa tan oxy sẽ là: (3).

Công thức (3) được sử dụng để tính độ hòa tan oxy trong hệ thống.

**3.2.5. Sơ đồ nguyên lý toàn mạch.**

****

Hình 3.12. Sơ đồ nguyên lý hệ thống.

Sơ đồ nguyên lý trên mô tả các thành phần điện tử và cảm biến được kết nối để thu thập dữ liệu, xử lý dữ liệu và truyền dữ liệu qua mạng.

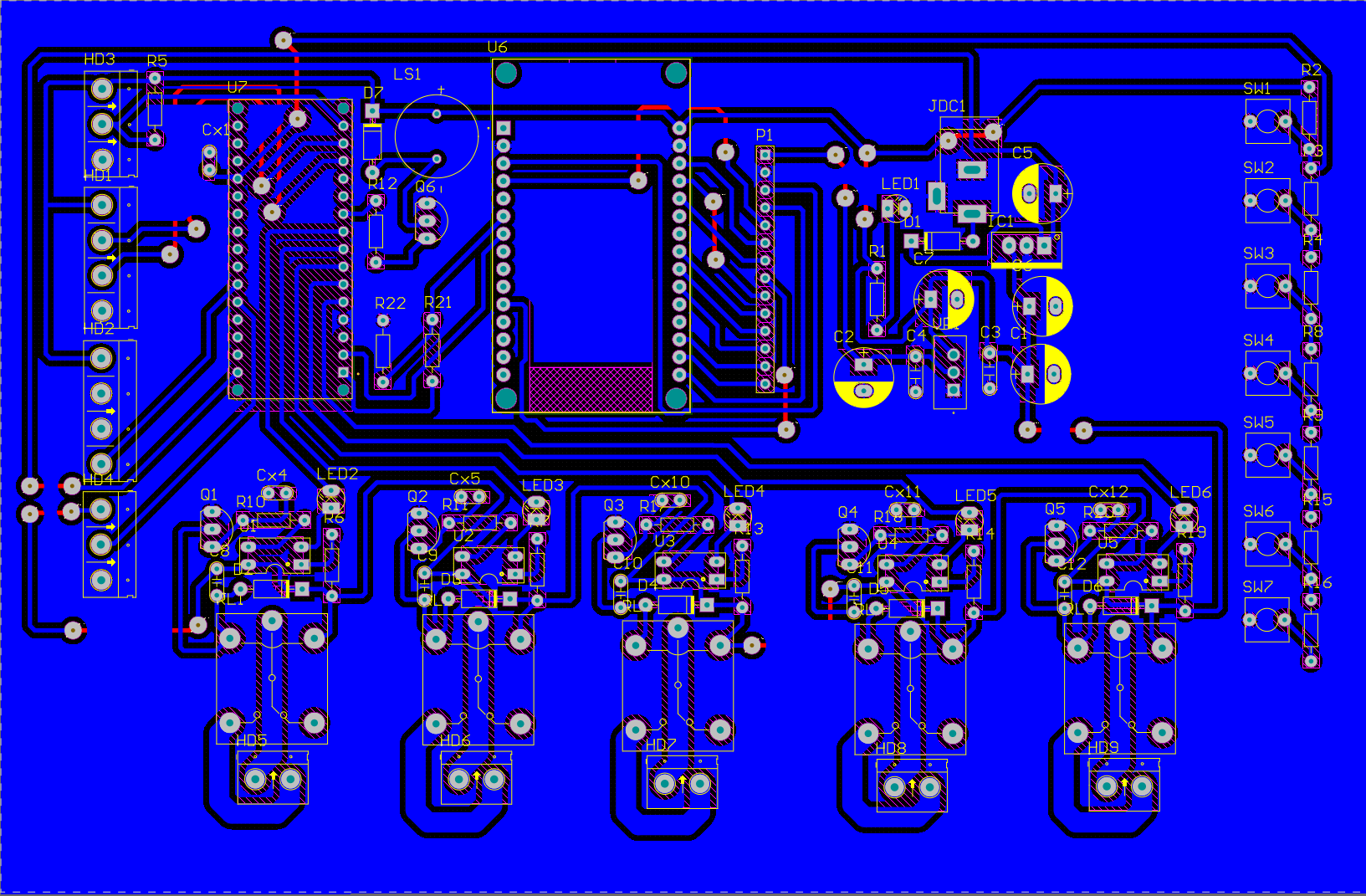
**CHƯƠNG 4: THI CÔNG HỆ THỐNG.**

**4.1. GIỚI THIỆU**

Sau quá trình tính toán, lựa chọn linh kiện phù hợ, chúng ta sẽ bắt đầu thi công và lắp ráp hệ thống. Phần này trình bày việc thiết kế bo mạch, và app di động sử dụng cho hệ thống.

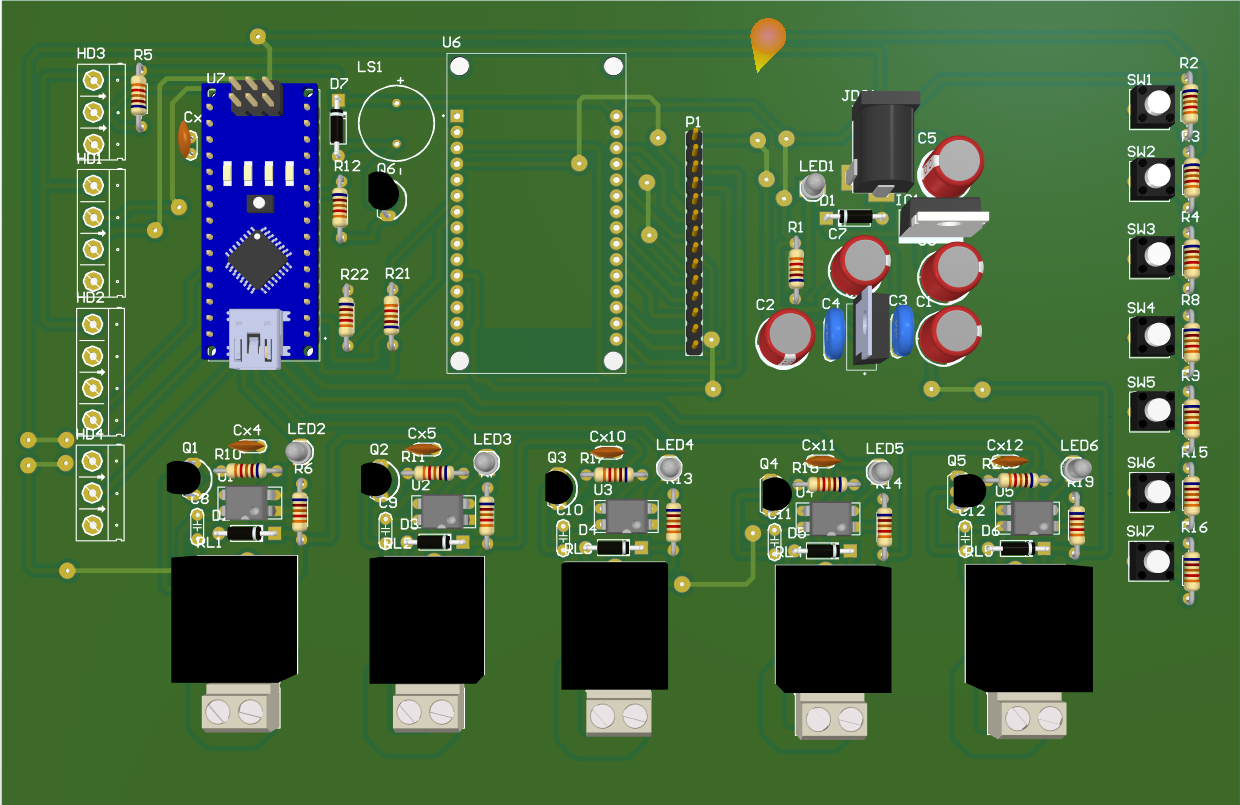
**4.2. THI CÔNG BO MẠCH**

Sau khi thiết kế xong sơ đồ nguyên lý hệ thống, chúng ta sẽ tiến hành vẽ PCB cho bo mạch.



Hình 4.1. Sơ đồ mạch in

Bên dưới là sơ đồ mạch in ở dạng 3D, với sơ đồ mạch in chúng ta có thể quan sát được mặt trước, mặt sau, vị trí các linh kiện trên bo mạch. Điều này giúp chúng ta điều chỉnh bố cục hợp lý hơn.



Hình 4.2. Mạch in 3D

**Linh kiện được sử dụng trong hệ thống**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên linh kiện** | **Số lượng** | **Điện áp** |
| 1 | Relay | 5 | 5V |
| 2 | Pc817 | 5 | 5V |
| 3 | Transistor c1815 | 6 |  |
| 4 | Led | 6 | 5V |
| 5 | Điện trở | 16 |  |
| 6 | 1N4007 | 7 |  |
| 7 | Tụ gốm | 13 |  |
| 8 | Tụ hóa | 5 |  |
| 9 | Header | 9 |  |
| 10 | Buzzer | 1 | 5V |
| 11 | Arduino Uno | 1 | 9V |
| 12 | Esp32 | 1 | 5V |
| 13 | Hàng rào đơn cái | 5 |  |
| 14 | Jack DC | 1 |  |
| 15 | LCD | 1 | 3.3V |
| 16 | Cảm biến pH4502C | 1 | 5V |
| 17 | Cảm biến độ đục nước | 1 | 5V |
| 18 | Cảm biến nhiệt độ ds18b20 | 1 | 5V |
| 19 | Cảm biến siêu âm hcsr04 | 1 | 5V |
| 20 | LM7805 | 1 | 9V |
| 21 | LM1117T | 1 | 5V |

Bảng 4.1.Bảng linh kiện sử dụng trong hệ thống

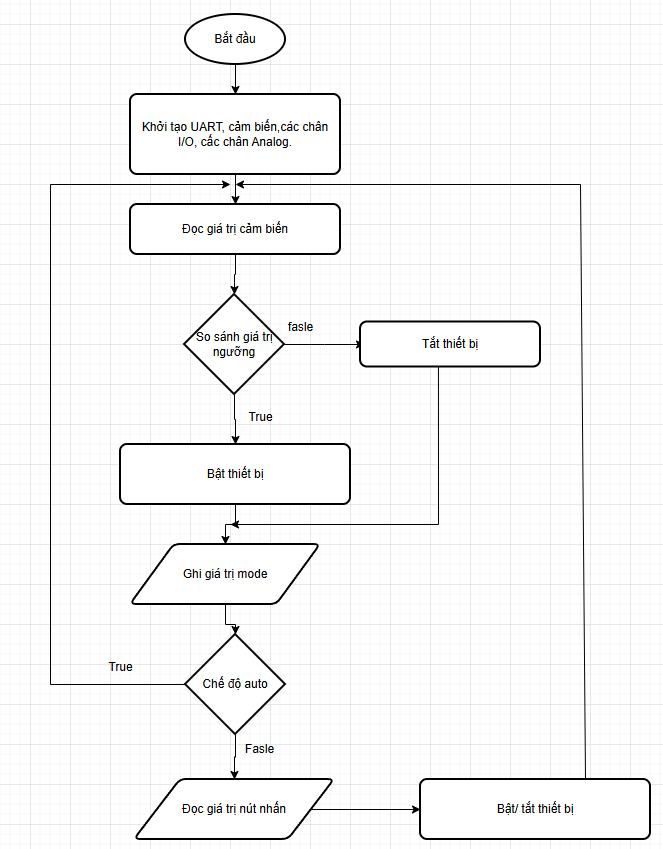
**4.3. LẬP TRÌNH HỆ THỐNG.**

**4.3.1. Lưu đồ giải thuật.**

**Các chức năng của hệ thống bao gồm:**

* Hệ thống cho phép người dùng theo dõi các thông số môi trường như nhiệt độ, độ đục nước, độ pH, mực nước, độ hòa tan oxy.
* Hệ thống cho phép quan sát biểu đồ theo tháng và biểu đồ thời gian thực.
* Hệ thống thực hiện cảnh báo về mail, và lưu nhật ký cảnh báo để người dùng dễ theo dõi.
* Hệ thống cho phép xem trạng thái hoạt động của thiết bị.
* Ở chế độ tự động, hệ thống sẽ điều khiển các thiết bị theo các giá trị ngưỡng mà người dùng cài đặt.
* Ở chế độ thủ công, người dùng có thể bật/tắt các thiết bị điện thông qua nút nhấn vật lý hoặc điều khiển thông qua app di động.
* Hệ thống cho phép người dùng điều chỉnh ngưỡng an toàn của cảm biến.

**Lưu đồ gải thuật của arduino, cảm biến và khối ngoại vi:**



Hình 4.3. Lưu đồ giải thuật của arduino , cảm biến và thiết bị ngoại vi

Chương trình bắt đầu sẽ khởi tạo các biến, khởi tạo các chân I/O, analog. Mặc định ban đầu sẽ là trạng auto. Ban đầu giá trị của các cảm biến sẽ được đọc, sau đó sẽ so sánh với giá trị ngưỡng, nếu nằm trong ngưỡng cho phép thì tắt thiết bị và tiếp tục chương trình, nếu sai thì sẽ bật thiết bị. Tiếp tục ghi giá trị mode. Nếu chế độ mode là auto thì quay lại vòng lặp, nếu chế độ mode là manual thì đọc các giá trị nút nhận và bật/tắt các thiết bị, tiếp tục quay trở lại vòng lặp chính.

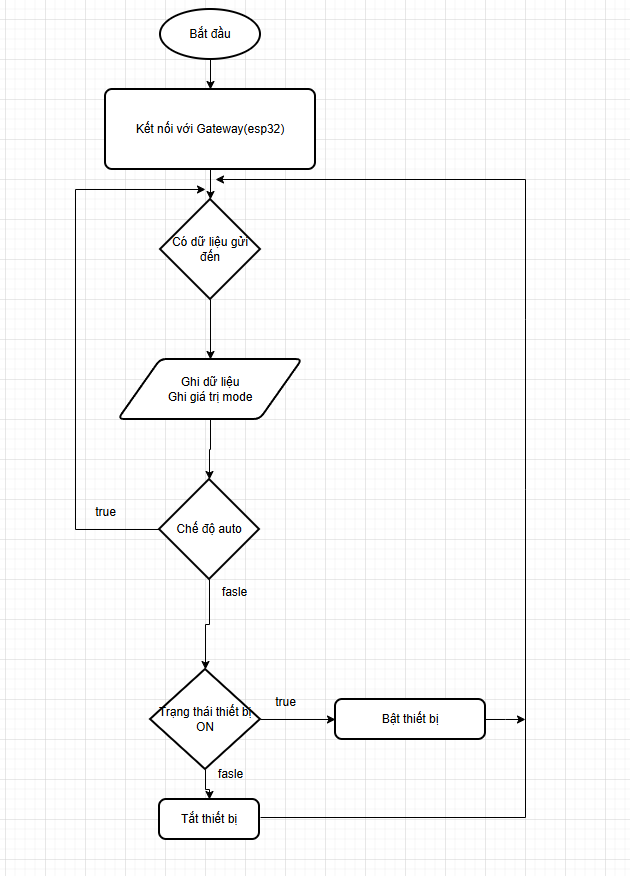
**Lưu đồ giải thuật gateway(esp32) với arduino uno.**

****

Hình 4.4. Lưu đồ giải thuật esp32 và arduino

Chương trình bắt đầu khởi tạo SPI, giao thức UART, khi nhận được dữ liệu từ arduin, thì hệ thống sẽ ghi dữ liệu và bắt đầu hiển thị dữ liệu lên LCD, đồng thời gửi lên firebase.

**Lưu đồ giải thuật điều khiển thiết bị bằng app thông qua gateway**

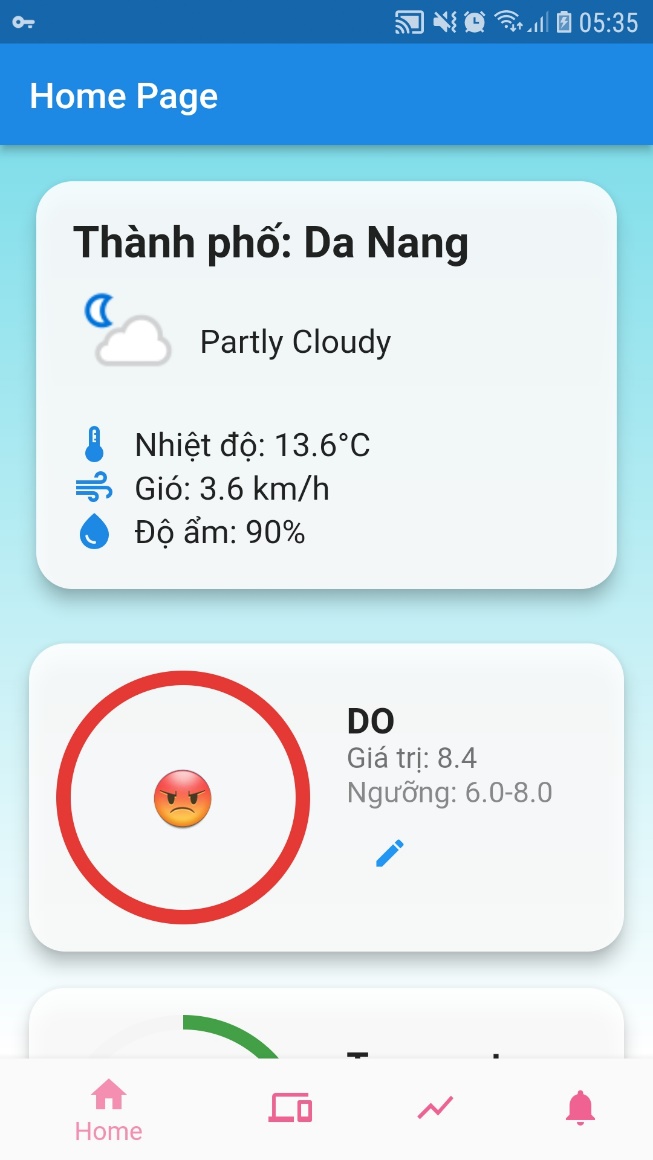
****

Hình 4.5. Lưu đồ giải thuật điều khiển thiết bị bằng app thông qua gateway

Bắt đầu kết nối với gateway, khi có dữ liệu gửi đến từ gateway(esp32), arduino sẽ ghi dữ liệu, sau đó kiểm tra chế độ mode có là auto hay không, nếu có sẽ quay lại tiếp tục chương trình, nếu không sẽ kiểm tra trạng thái thiết bị, nếu true bật thiết bị và tiếp tục vòng lặp, nếu fasle sẽ tắt thiết bị và quay trở lại vòng lặp.

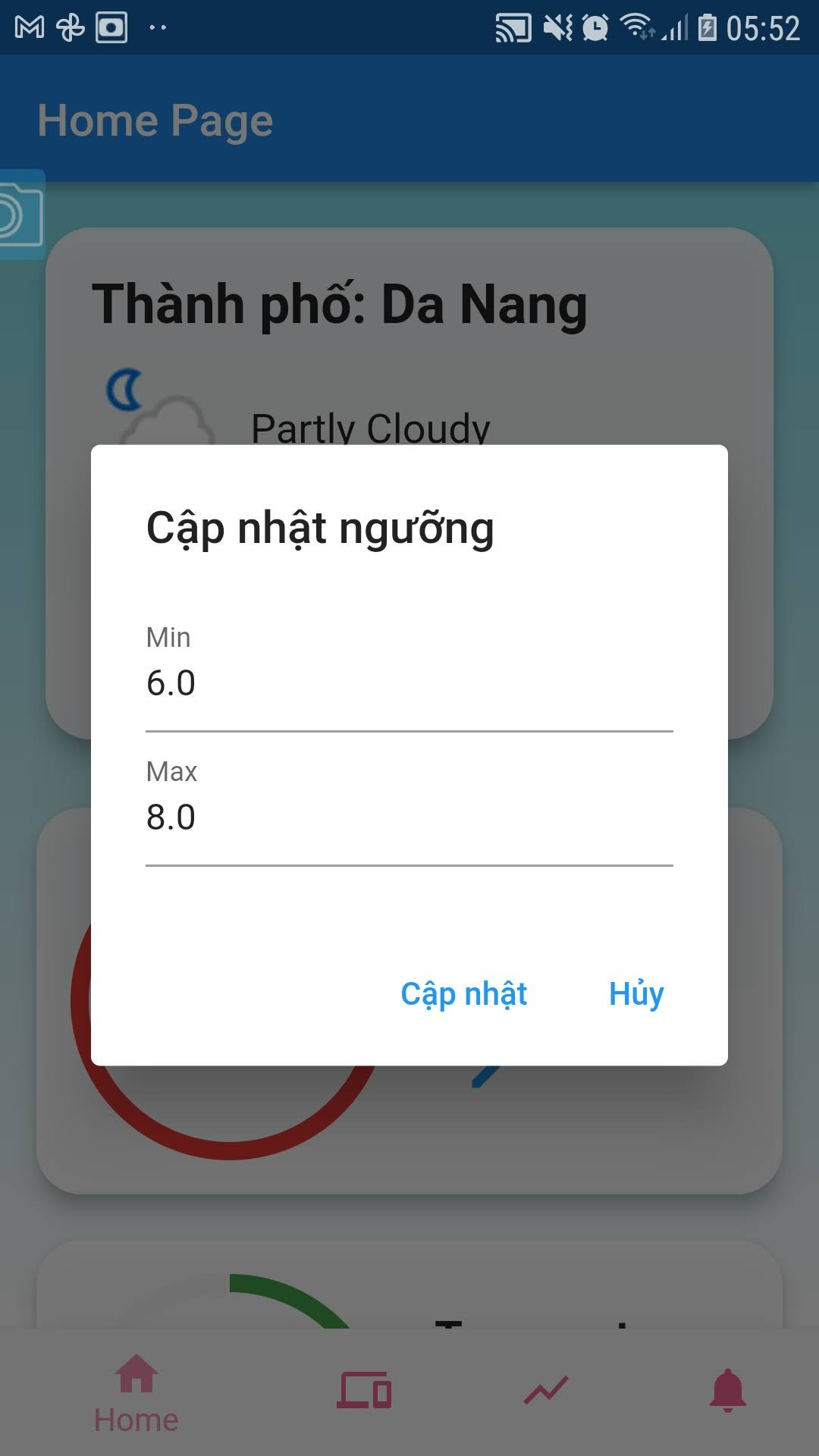
**4.3.2. App di động.**

**Giao diện trang home**

****

Hình 4.6. Home page

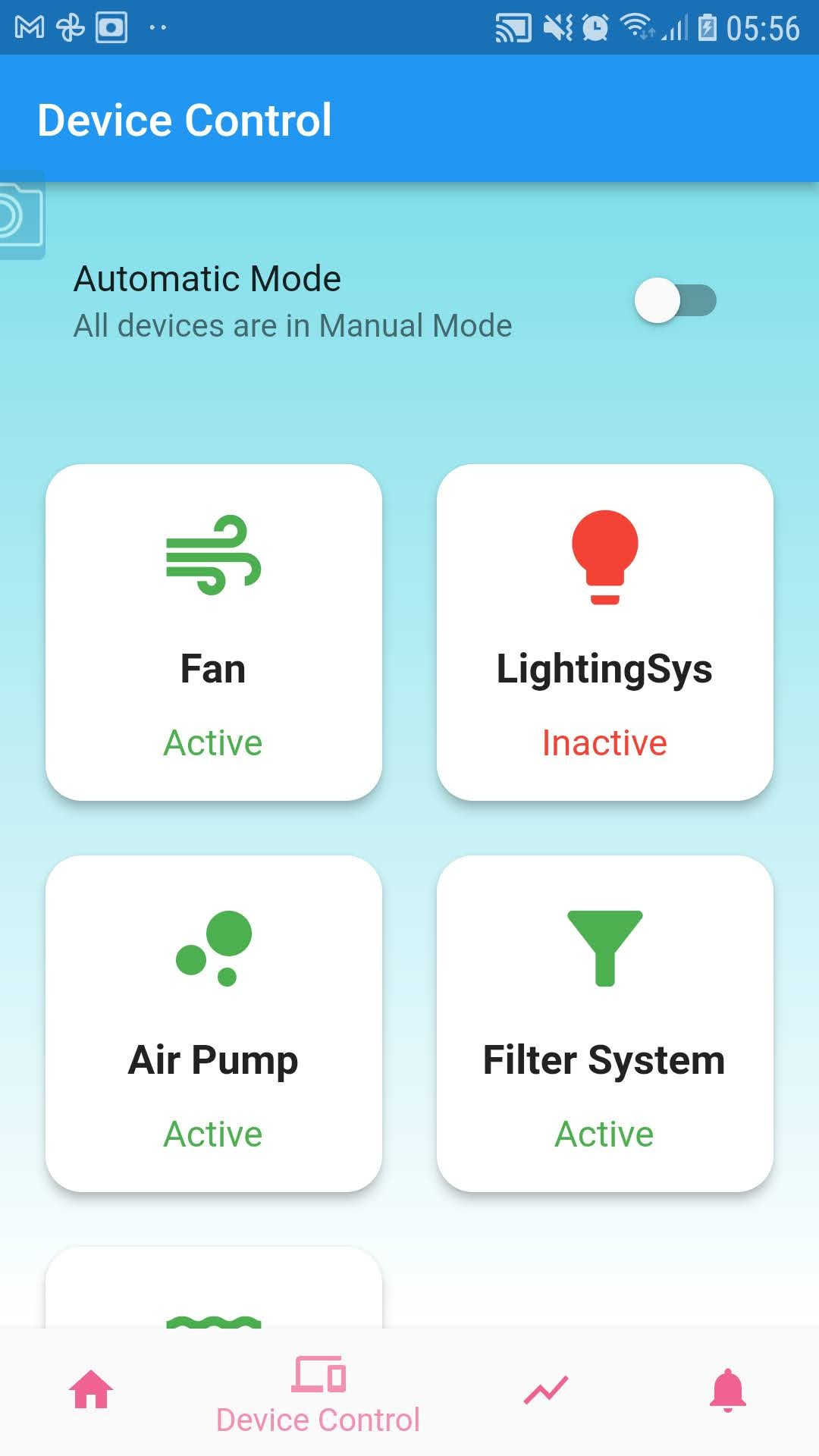
Giao diện home, cho phép người dùng theo dõi thông số môi trường nuôi thủy sản,và cho phép người dùng điều chỉnh ngưỡng thông số môi trường theo nhu cầu của mình.



Hình 4.7. Giao diện thay đổi ngưỡng an toàn

Cho phép người dùng cập nhật ngưỡng max, min của thông số môi trường.

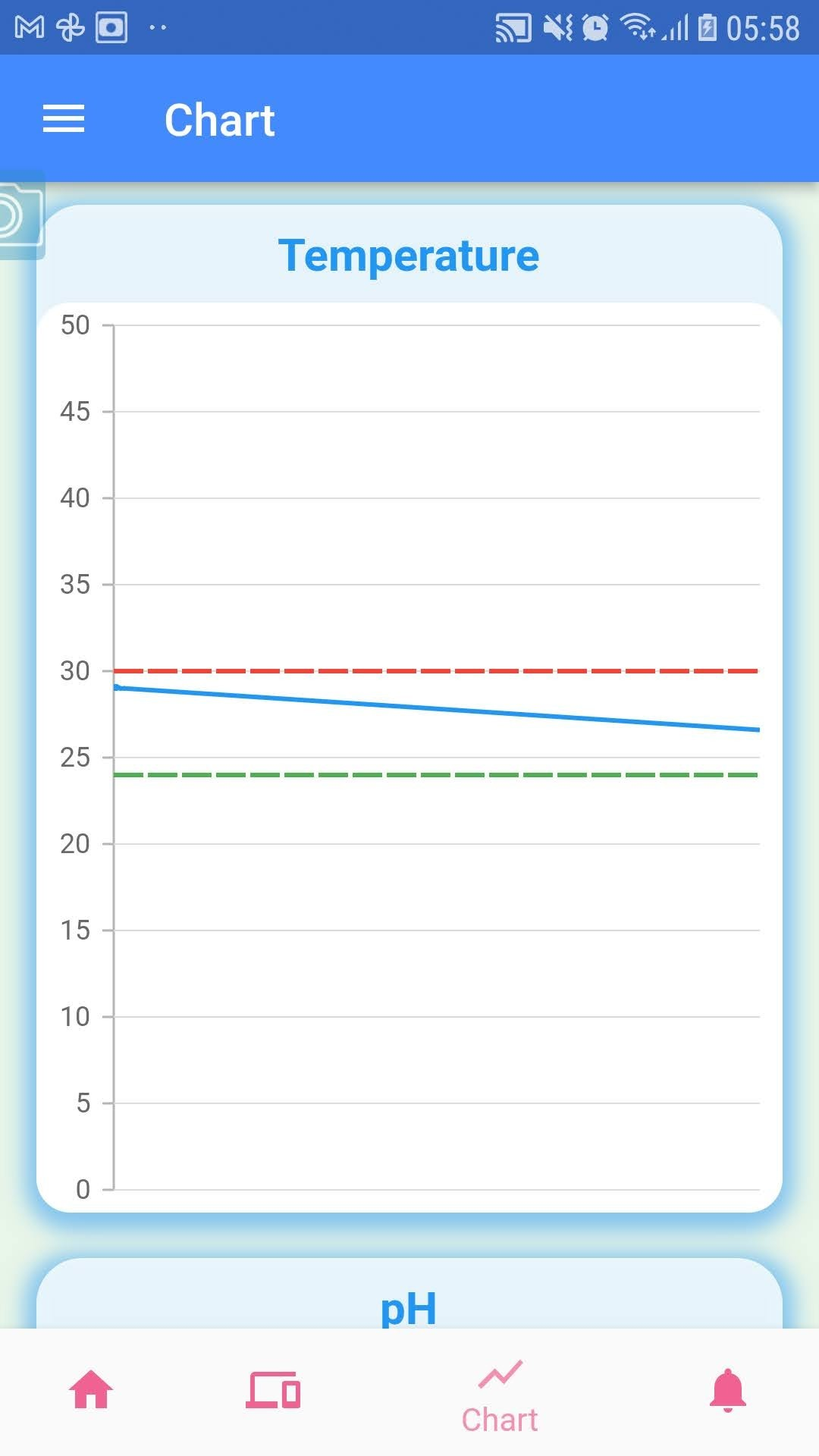
**Giao diện trang Device control**

****

Hình 4.8. Giao diện trạng thái thiết bị.

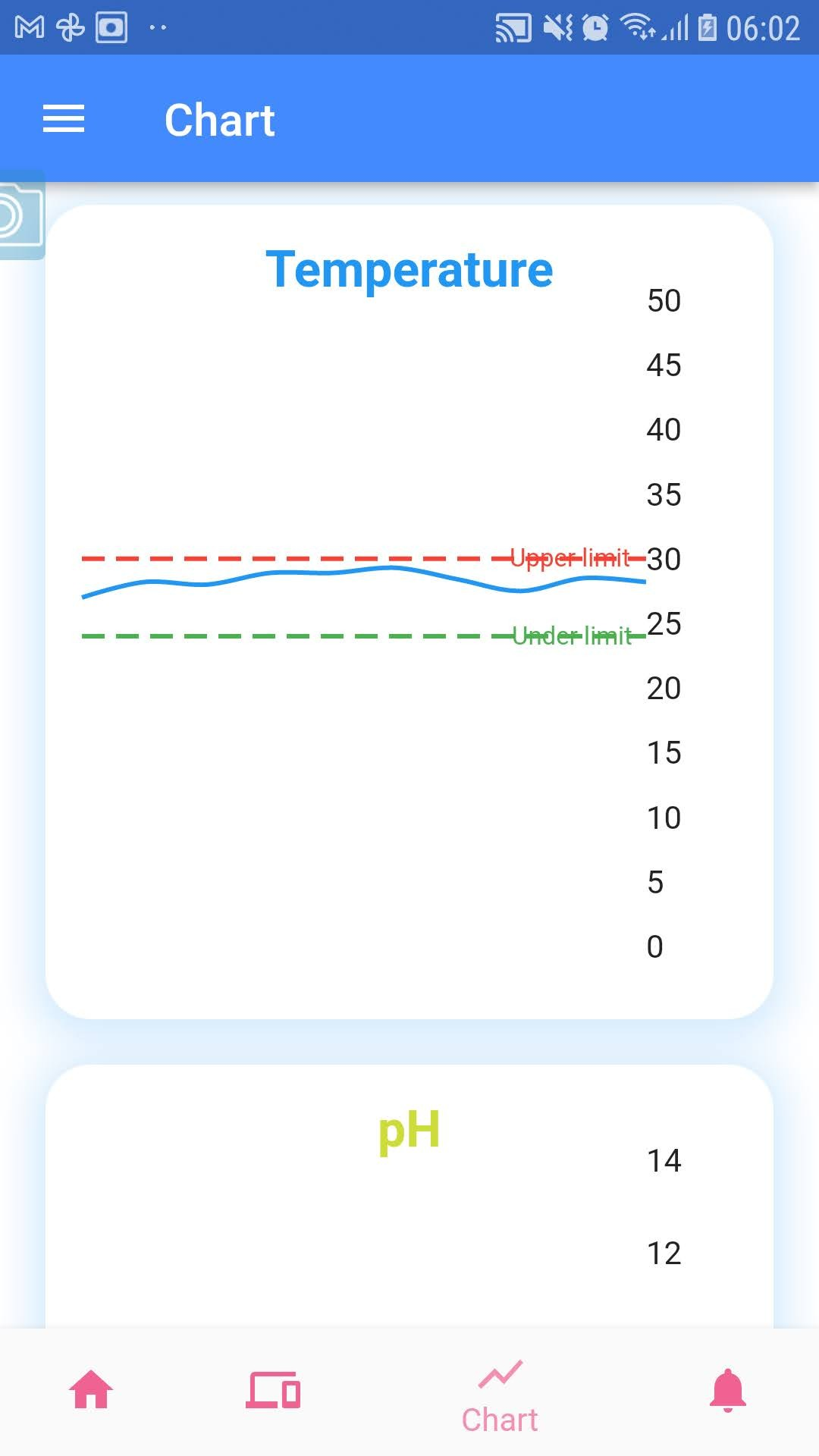
Trang này cho phép người dùng xem trạng thái thiết bị, và điều khiển thiết bị từ xa.

**Giao diện Chart**



Hình 4.9. Giao diện realtime chart

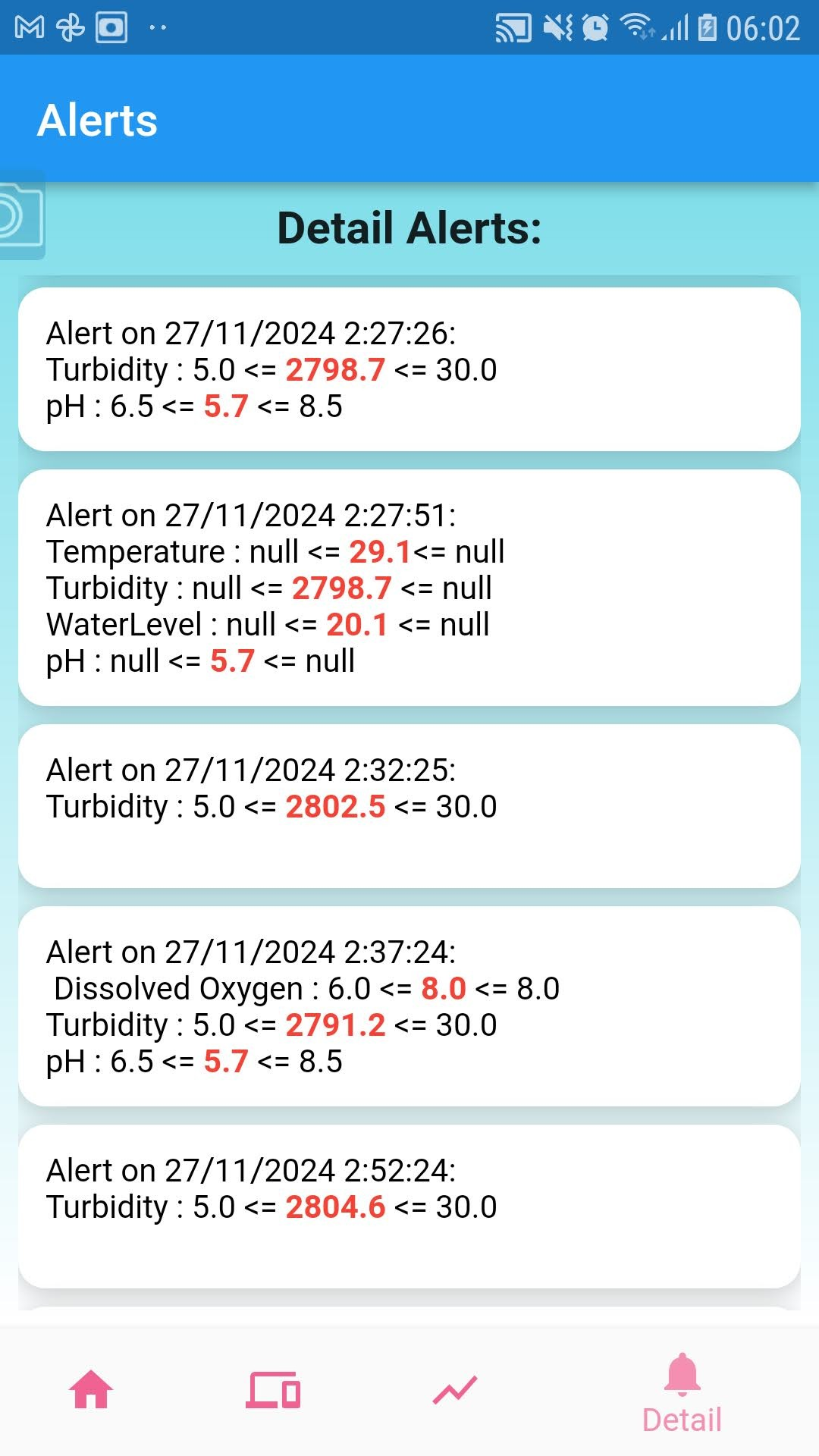
Biểu đồ realtime cho phép người dùng theo dõi thông số cảm biến theo thời gian thực.



Hình 4.10. Biểu đồ tháng

Biểu đồ tháng cho phép người dùng quan sát thông số một trường theo tháng, các thông số môi trường được tính trung bình từng ngày.

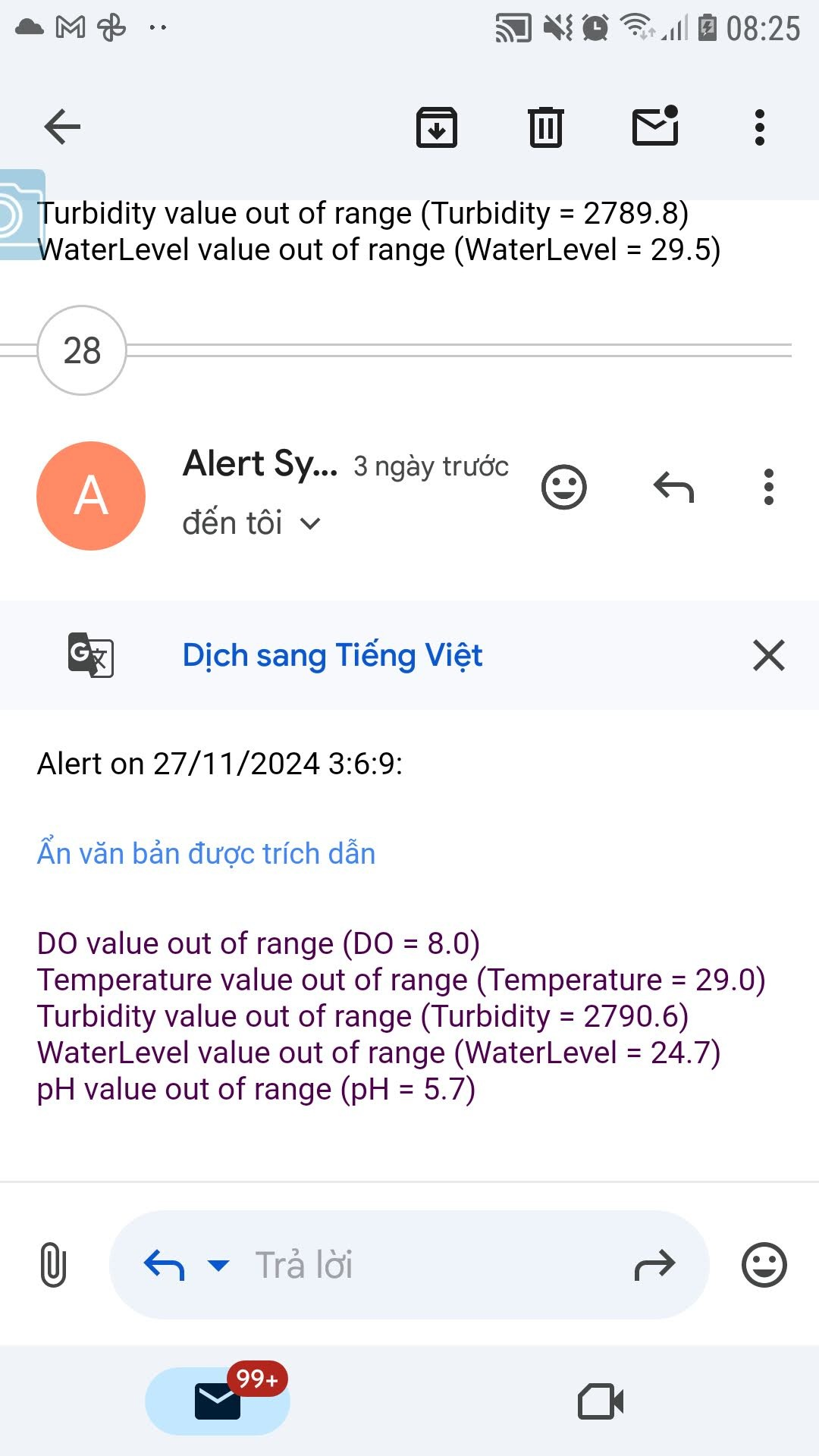
**Giao diện trang Detail**



Hình 4.11.Giao diện xem thông báo vượt ngưỡng

Trang này hiển thị thông báo gồm thời gian mà các thông số môi trường vượt ngưỡng.

Giao diện thông báo mail người dùng.



Hình 4.12. Thông báo vượt ngưỡng về mail người dùng.

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1. KẾT LUẬN**

Đề tài "Xây dựng hệ thống IoT trong nuôi thủy sản" đã hoàn thành mục tiêu thiết kế và triển khai một hệ thống giám sát và điều khiển tự động. Hệ thống giúp người nuôi dễ dàng theo dõi các thông số môi trường quan trọng như nhiệt độ, độ pH, nồng độ oxy hòa tan... theo thời gian thực thông qua ứng dụng di động. Đồng thời, chức năng điều khiển tự động các thiết bị như máy sục khí, máy bơm và hệ thống chiếu sáng giúp người dùng thuận lợi hơn trong việc quản lý ao nuôi từ xa.

**5.2. HƯỚNG PHÁT TRIỂN**

Đề tài cơ bản đã đáp ứng được cac yêu cầu dặt ra, tuy nhiên để hoàn thiện đề tài hơn nữa thì cần phải cải tiến thêm.

* Bổ sung các loại cảm biến mới như cảm biến amoniac, cảm biến độ mặn, cảm biến CO₂ để đáp ứng nhu cầu giám sát chuyên sâu cho các loại thủy sản khác nhau.
* Áp dụng trí tuệ nhân tạo (AI) và máy học để dự đoán các rủi ro như sự giảm đột ngột của nồng độ oxy hoặc sự thay đổi nhiệt độ bất thường, từ đó hỗ trợ người nuôi đưa ra quyết định kịp thời.
* Xây dựng hệ thống điều khiển dựa trên thuật toán tự động như PID hoặc logic mờ để điều chỉnh các thiết bị hoạt động theo thời gian thực.
* Kết hợp với hệ thống năng lượng mặt trời hoặc gió để giảm chi phí vận hành, thân thiện với môi trường và nâng cao tính bền vững.

# TÀI LIỆU THAM KHẢO

1. [Warren Gay],[2022], [Beginning STM32: Developing with FreeRTOS, libopencm3 and GCC]
2. [Phạm Quốc Phương ],[2015], [*Vi điều khiển]*,[ Đại học Công nghệ TP. Hồ Chí Minh]
3. [Rajkumar Buyya, Amir Vahid Dastjerdi], [2016], [Internet of Things: Principles and Paradigms].
4. [Raj Kamal],[],[internet of things architecture and design printciples]
5. [Định nghĩa và phương pháp đo oxy hòa tan (DO) trong nước](https://epcb.vn/blogs/news/dinh-nghia-va-phuong-phap-do-oxy-hoa-tan-do-trong-nuoc)
6. [GitHub - jeffmer/micropython-ili9341: Micropython Driver for ILI9341 display](https://github.com/jeffmer/micropython-ili9341)