

TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
HCMC University of Technology and Education



MÔN HỌC: ĐỒ ÁN 1
HỆ THỐNG KIỂM SOÁT HỒ CÁ

SVTH1: VÕ MINH THUẬN - 21161366

SVTH2: LÊ QUANG THƯỜNG - 21161367

KHÓA: 2021 - 2025

NGÀNH: CNKT ĐIỆN TỬ - VIỄN THÔNG

GVHD: TH.S NGUYỄN ĐÌNH PHÚ

TP.HCM, Ngày 17 Tháng 06 Năm 2024



NHIỆM VỤ ĐỒ ÁN 1

Họ và tên sinh viên: Võ Minh Thuận MSSV: 21161366

Họ và tên sinh viên: Lê Quang Thương MSSV: 21161367

Ngành: CNKT Điện Tử - Viễn Thông

Giảng viên hướng dẫn: ThS. Nguyễn Đình Phú

Ngày nhận đề tài: 01/03/2024 Ngày nộp đề tài: 17/06/2024

1. Tên đề tài: Hệ thống kiểm soát hồ cá
2. Các số liệu, tài liệu ban đầu: Giáo trình Điện tử cơ bản; Giáo trình vi xử lý: vi điều khiển PIC 16F877A; Giáo trình thực tập vi xử lý.
3. Nội dung thực hiện đề tài: Thiết kế Hệ thống kiểm soát hồ cá dùng Pic 16F877A. Hiển thị trên Lcd những thông tin về nhiệt độ hồ cá, nhiệt độ giới hạn cho hồ cá, thời gian thực, thời gian cho cá ăn. Điều khiển nhiệt độ về trạng thái nhiệt độ giới hạn để nhiệt độ hồ cá đạt nhiệt độ cài đặt, giúp cá phát triển tốt nhất. Điều khiển cho cá ăn theo thời gian được thiết lập sẵn.
4. Sản phẩm: Mạch bao gồm các khối, khối xử lý trung tâm, khối hiển thị, khối thời gian thực, khối nút nhấn, khối nhiệt độ, khối role, khối servo, khối hiển thị.

GIẢNG VIÊN HƯỚNG DẪN



TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN - ĐIỆN TỬ

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING



PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT

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

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

3. Khuyết điểm:

.....

.....

.....

4. Điểm sinh viên thực hiện 1:.....

5. Điểm sinh viên thực hiện 2:.....

Tp. Hồ Chí Minh, ngày 17 tháng 06 năm 2024

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)



LỜI CẢM ƠN

Để đề tài được hoàn thành theo đúng tiến độ và mục tiêu đặt ra cũng như đạt được kết quả như trên thì ngoài sự nỗ lực của bản thân, em còn nhận được sự chỉ bảo và giúp đỡ của giáo viên hướng dẫn và bạn bè xung quanh.

Em xin chân thành cảm ơn:

- Sự hướng dẫn và góp ý tận tình của thầy Nguyễn Đình Phú. Cảm ơn thầy đã cho em nhiều thông tin chuyên ngành, hỗ trợ và chỉnh sửa cho em những vấn đề chưa hoàn thiện trong lúc hoàn thành đồ án 1 của mình.

- Các bạn sinh viên trong và ngoài lớp đã giúp đỡ về mặt phương tiện, sách vở, ý kiến đóng góp, sửa đổi . . .

Trong quá trình thực hiện đề tài, dù cố gắng hoàn thành tốt nhất có thể nhưng em cũng sẽ khó tránh khỏi nhiều thiếu sót. Rất mong nhận được sự góp ý, phê bình, chỉ dẫn của thầy, các bạn sinh viên và bạn đọc.

Em xin chân thành cảm ơn !

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	5
DANH MỤC BẢNG BIỂU.....	8
CHƯƠNG 1: TỔNG QUAN	9
1.1. GIỚI THIỆU TÌNH HÌNH NGHIÊN CỨU HIỆN NAY	9
1.2. MỤC TIÊU NGHIÊN CỨU	9
1.3. NHIỆM VỤ NGHIÊN CỨU	9
1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	10
1.4.1. Đối tượng nghiên cứu	10
1.4.2. Phạm vi nghiên cứu	10
1.5. PHƯƠNG PHÁP NGHIÊN CỨU	10
1.6. BỐ CỤC CỦA ĐỒ ÁN	11
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	12
2.1. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN PIC 16F877A.....	12
2.1.1. Vi điều khiển PIC 16F877A	12
2.1.2. Kiến trúc của PIC 16F877A	13
2.1.3. Sơ đồ nối chân và chức năng của từng chân	14
2.2. CẢM BIẾN NHIỆT ĐỘ IC DS18B20	18
2.2.1. Giới thiệu chung về DS18B20.....	18
2.2.2. Sơ đồ chân của DS18B20	18
2.2.3. Một số chú ý khi sử dụng DS18B20	20
2.3. IC THỜI GIAN THỰC DS1307	20
2.3.1. Giới thiệu về DS1307	20

ĐỒ ÁN 1

2.3.2. Sơ đồ chân của DS1307.....	20
2.3.3. Tổ chức thanh ghi trong DS1307	21
2.3.4. Cách hoạt động và một số chú ý khi sử dụng DS1307.....	24
2.4. HIỂN THỊ - LCD 16x02.....	25
2.5. IC 7805	27
2.6. Tụ điện.....	28
2.7. Điện trở	30
2.8. MỘT SỐ LÝ THUYẾT CƠ BẢN ĐƯỢC SỬ DỤNG	32
2.8.1. I2C	32
2.8.2 UART	36
2.8.3. 1-WIRE.....	37
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	40
3.1. GIỚI THIỆU	40
3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	40
3.2.1. Thiết kế sơ đồ khối hệ thống	40
3.2.2. Tính toán và thiết kế mạch	41
3.2.2.1. Khối nguồn 1:	42
3.2.2.2. Khối nguồn 2	43
3.2.2.3. Khối PIC 16F877A.....	44
3.2.2.4. Khối 4 nút nhấn	45
3.2.2.5. Khối ds1307.....	46
3.2.2.6. Khối ds18b20.....	46
3.2.2.7. Khối lcd 1602	47
3.2.2.8. Khối 2 relay	48

ĐỒ ÁN 1

3.2.2.9. Khối esp-8266	48
3.2.3. Sơ đồ nguyên lý toàn mạch	49
CHƯƠNG 4: THI CÔNG HỆ THỐNG	50
4.1. GIỚI THIỆU	50
4.2. THI CÔNG HỆ THỐNG	50
4.2.1. Thi công bo mạch	50
4.2.2. lắp ráp và kiểm tra	51
4.3. LẬP TRÌNH HỆ THỐNG	52
4.3.1. Lưu đồ giải thuật.....	52
4.3.1.1. Lưu đồ giải thuật cho pic16f877a.....	52
Chương trình code pic:	66
4.3.1.2. Lưu đồ giải thuật cho esp8266	67
Chương trình code esp:.....	68
4.3.1.3. Lưu đồ giải thuật cho ứng dụng di động android	69
Chương trình code app android:	70
4.3.2. Phần mềm lập trình cho vi điều khiển	70
4.3.2.1. Phần mềm pic-c	70
4.3.2.2. Phần mềm Arduino.....	74
4.3.3. Phần mềm lập trình cho ứng dụng di động.....	77
4.4. HƯỚNG DẪN SỬ DỤNG, THAO TÁC.....	80
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ	89
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	92
6.1. KẾT LUẬN	92
6.2. HƯỚNG PHÁT TRIỂN	92

ĐỒ ÁN 1

TÀI LIỆU THAM KHẢO 93

DANH MỤC HÌNH ẢNH

Hình 2. 1 Vi điều khiển PIC 16F877A - I/P thực tế	12
Hình 2. 2 Cấu trúc cơ bản của vi điều khiển PIC 16F877A	14
Hình 2. 3 Sơ đồ chân vi điều khiển PIC 16F877A	14
Hình 2. 4 Mạch ứng dụng PIC 16F877A	18
Hình 2. 5 Sơ đồ chân DS18B20	19
Hình 2. 6 Sơ đồ kết nối nguồn chung với vi xử lý của DS18B20	19
Hình 2. 7 Sơ đồ chân của DS1307	21
Hình 2. 8 Tổ chức thanh ghi bên trong của DS1307	22
Hình 2. 9 Tổ chức thanh ghi được mã hoá theo bit bên trong của DS1307	22
Hình 2. 10 Màn hình LCD 16x2	25
Hình 2. 11 Hình 7805 dạng transistor	28
Hình 2. 12 Hình các loại tụ điện	30
Hình 2. 13 Hình vạch đọc điện trở	31
Hình 2. 14 Hình ví dụ tính điện trở	32
Hình 2. 15 Hình i2c protocols	33
Hình 2. 16 Hình khung dữ liệu i2c	34
Hình 3. 1 Sơ đồ khối hệ thống	40
Hình 3. 2 Khối nguồn 1	42
Hình 3. 3 Khối nguồn 2	43
Hình 3. 4 Khối PIC 16F877A	44
Hình 3. 5 Bảng đặc tính i2c	45
Hình 3. 6 Khối 4 nút nhấn	45
Hình 3. 7 Khối ds1307	46
Hình 3. 8 Khối ds18b20	46
Hình 3. 9 Khối lcd 1602	47
Hình 3. 10 Khối 2 relay	48

ĐỒ ÁN 1

Hình 3. 11 Khởi esp-8266.....	48
Hình 3. 12 Sơ đồ nguyên lý toàn mạch	49
Hình 4. 1 Hệ thống	50
Hình 4. 2 Lưu đồ chương trình chính cho pic16f877a	52
Hình 4. 3 Lưu đồ setup_initialize	53
Hình 4. 4 Lưu đồ ds1307_setup.....	54
Hình 4. 5 Lưu đồ servo_degree_0	55
Hình 4. 6 Lưu đồ servo_degree_90	56
Hình 4. 7 Lưu đồ menu.....	57
Hình 4. 8 Lưu đồ handel_1	58
Hình 4. 9 Lưu đồ handle_2.....	59
Hình 4. 10 Lưu đồ handle_3.....	60
Hình 4. 11 Lưu đồ handle_4.....	61
Hình 4. 12 Lưu đồ handle_5.....	62
Hình 4. 13 Lưu đồ handle_6.....	62
Hình 4. 14 Lưu đồ control_temperature	63
Hình 4. 15 Lưu đồ control_servo.....	64
Hình 4. 16 Lưu đồ handle_uart.....	65
Hình 4. 17 Lưu đồ send_oke	66
Hình 4. 18 Lưu đồ chương trình cho esp.....	67
Hình 4. 19 Lưu đồ app ứng dụng.....	69
Hình 4. 20 Cài ideutilsupd.....	70
Hình 4. 21 Quá trình cài 1 pic	71
Hình 4. 22 Quá trình cài 2 pic	71
Hình 4. 23 Quá trình cài 3 pic	72
Hình 4. 24 Cài pic-c.....	72
Hình 4. 25 Thiết lập file cài đặt.....	72
Hình 4. 26 Chọn Configure	73

ĐỒ ÁN 1

Hình 4. 27 Lập trình pic-c	73
Hình 4. 28 Setup 1 arduino.....	74
Hình 4. 29 setup 2 arduino.....	75
Hình 4. 30 setup 3 arduino.....	75
Hình 4. 31 Setup 4 arduino.....	76
Hình 4. 32 Finnal setup Arduino	76
Hình 4. 33 Hình biên dịch App B5.1	79
Hình 4. 34 Hình giao diện app.....	80
Hình 4. 35 Hình công tắc, đèn báo, nguồn cấp.....	81
Hình 4. 36 Hình hiển thị lcd – thông tin ban đầu	82
Hình 4. 37 HỈnh hiển thị lcd với nhiệt độ và thời gian.....	82
Hình 4. 38 Hình vị trí nút nhấn.....	83
Hình 4. 39 Hình hiển thị ở handle_2	83
Hình 4. 40 Hình hiển thị ở handle_3	84
Hình 4. 41 Hình hiển thị ở handle_3 chỉnh gpg	85
Hình 4. 42 Hình hiển thị ở handle_3 chỉnh ntn	85
Hình 4. 43 Hình hiển thị ở handle_3 chỉnh thu	85
Hình 4. 44 Hình hiển thị ở handle_4	86
Hình 4. 45 Hình hiển thị ở handle_5	86
Hình 4. 46 Hình hiển thị ở handle_6	87
Hình 4. 47 Hình 2 realy	87
Hình 4. 48 Hình chân cắm kết nối pic-kit2	88
Hình 5. 1 Hình kết quả 1.....	89
Hình 5. 2 Hình kết quả 2.....	90
Hình 5. 3 Hình kết quả 3.....	90
Hình 5. 4 Hình kết quả 4.....	91

DANH MỤC BẢNG BIỂU

Bảng 2. 1 Bảng chức năng của các chân tín hiệu PIC 16F877A.....	14
Bảng 2. 2 Chức năng của các chân DS18B20	18
Bảng 2. 3 Chức năng của các chân DS1307.....	20
Bảng 2. 4 Sơ đồ chân LCD 16x2.....	26
Bảng 2. 5 Địa chỉ của LCD 16x2	26
Bảng 3. 1 Dòng tiêu thụ cho khối nguồn 1.....	42
Bảng 3. 2 Dòng tiêu thụ cho khối nguồn 2.....	43
Bảng 5. 1 Bảng đo nhiệt độ cảm biến ds18b20	89

CHƯƠNG 1: TỔNG QUAN

1.1. GIỚI THIỆU TÌNH HÌNH NGHIÊN CỨU HIỆN NAY

Kiểm soát hồ cá là một chủ đề nóng hổi trong cộng đồng người nuôi cá cảnh hiện nay. Nhờ sự phát triển của khoa học kỹ thuật, việc quản lý môi trường nước, thức ăn, ánh sáng và các yếu tố khác trong hồ cá đã trở nên dễ dàng và hiệu quả hơn bao giờ hết. Kiểm soát hồ cá hiệu quả giúp đảm bảo môi trường sống lý tưởng cho cá, nâng cao sức khỏe và tuổi thọ của chúng. Việc ứng dụng công nghệ và kiến thức sinh học một cách thông minh sẽ giúp người nuôi tạo dựng một hồ cá đẹp mắt, cân bằng và bền vững. Công nghệ thông minh cũng góp phần nâng cao khả năng kiểm soát hồ cá. Các ứng dụng di động cho phép người chơi có thể dễ dàng theo dõi và điều chỉnh các thông số nước, kiểm tra lượng thức ăn, giám sát sức khỏe,... của cá từ xa. Người chơi cá có thể sử dụng các thiết bị tự động để kiểm soát các yếu tố trong hồ, giúp tiết kiệm thời gian và công sức. Với mục đích là tìm hiểu và vận dụng được các kiến thức đã học được chúng em đã lên ý tưởng để thực hiện đề tài **“Thiết Kế Hệ Thống Kiểm Soát Hồ Cá”** trong khuôn khổ của đồ án 1.

1.2. MỤC TIÊU NGHIÊN CỨU

Áp dụng những kiến thức đã học từ các môn học như mạch điện, điện tử cơ bản, kỹ thuật vi xử lý, thực tập vi xử lý.

Khi nghiên cứu đề tài này, chúng em muốn sử dụng vi điều khiển kết hợp với các module đã học tạo ra 1 sản phẩm có ích cho xã hội cũng như vận dụng được các kiến thức về thiết kế hệ thống. Đồng thời, đây còn là cơ hội để em tự kiểm tra lại kiến thức, khả năng tiếp thu và ứng dụng vào đời sống thực tế, từ đó tích lũy kinh nghiệm, làm tiền đề cho các dự án và các dự án sau này.

1.3. NHIỆM VỤ NGHIÊN CỨU

Đề tài thiết kế hệ thống kiểm soát hồ cá này có thể giúp mọi người nuôi có thể kiểm soát được nhiệt độ của hồ cá có phù hợp với loại cá mình đang nuôi hay không, nhiệt độ sẽ được hiển thị trên LCD. Từ đó hướng đến việc điều chỉnh nhiệt độ hồ cá trở về lại mức lý tưởng để cá có thể phát triển tốt. Ngoài ra hệ thống có thêm chức năng cho cá ăn theo thời gian mà người nuôi thiết lập sẵn.

1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

1.4.1. Đối tượng nghiên cứu

- Phần cứng:

- + Vi điều khiển pic 16f877a.
- + Module thời gian thực DS1307.
- + Module nhiệt độ DS18B20.
- + Nút nhấn.
- + Module Lcd 16x2.
- + Tụ, trở,...

- Phần mềm:

- + Ngôn ngữ lập trình cho vi xử lý (C) trên nền tảng PICCS IDE.
- + Ngôn ngữ lập trình cho esp (C) trên nền tảng ARDUINO IDE.
- + Ngôn ngữ lập trình ứng dụng di động (JavaScripts) trên nền tảng REACT-NATIVE.
- + Mô phỏng phần cứng trên phần mềm Proteus.
- + Thiết kế pcb phần cứng trên phần mềm Kicad.

1.4.2. Phạm vi nghiên cứu

Hệ thống có chức năng chỉnh sửa thời gian thực như giờ phút giây, ngày tháng năm, thứ, chỉnh sửa nhiệt độ giới hạn từ 01 đến 99 độ, chỉnh sửa thời gian cho cá ăn, hiển thị những thông tin về thời gian thực, nhiệt độ và chỉnh sửa thời gian, nhiệt độ giới hạn. Cho phép điều khiển tắt bật 2 role để cho phép thiết bị làm lạnh và thiết bị làm nóng hoạt động bằng cách so sánh nhiệt độ được với nhiệt độ giới hạn. Cho phép điều khiển servo trong 10s để cho cá ăn khi thời gian cài bằng thời gian thực từ cảm biến DS1307. Cho phép gửi dữ liệu về nhiệt độ hồ cá, nhiệt độ giới hạn, thời gian cho cá ăn lên firebase để hiển thị lên ứng dụng điện thoại được thiết kế để giám sát và điều khiển servo cho cá ăn và cài đặt nhiệt độ giới hạn và thời gian cho cá ăn.

1.5. PHƯƠNG PHÁP NGHIÊN CỨU

Để thực hiện được đề tài này, em đã tìm hiểu các cách thức hoạt động của vi điều khiển pic 16F877A và cách thức giao tiếp với nút nhấn và Lcd 16x2, các hoạt động truyền, nhận

ĐỒ ÁN 1

và xuất dữ liệu trên vi điều khiển,... Sử dụng ngôn ngữ lập trình C với các phần mềm dùng để xây dựng chương trình điều khiển và mô phỏng phần cứng.

1.6. BỐ CỤC CỦA ĐỒ ÁN

Chương 1: Tổng quan: Chương này em trình bày về tình hình nghiên cứu hiện nay, tính cấp thiết của đề tài, mục tiêu nghiên cứu, nhiệm vụ cần nghiên cứu, đối tượng và phạm vi nghiên cứu và phương pháp nghiên cứu.

Chương 2: Cơ sở lý thuyết: chương này em trình bày về sơ lược về vi điều khiển PIC 16F877A, LCD 16x2, IC DS1307, các linh kiện khác.

Chương 3: Tính toán và thiết kế hệ thống: chương này em trình bày các yêu cầu khi thiết kế hệ thống, cách thiết kế phần cứng và phần mềm, lưu đồ giải thuật của mạch.

Chương 4: Thi công hệ thống: chương này em trình bày các kết quả đã đạt được, hình ảnh khi hoàn thành mạch trên phần mềm và mạch in đã hoàn thiện, số liệu thống kê các linh kiện và video giới thiệu mạch đã hoàn thành.

Chương 5: Kết quả, nhận xét và đánh giá: chương này chúng em sẽ đánh giá hệ thống qua các thí nghiệm để kiểm tra xem hệ thống có chạy đúng các chức năng mà mục tiêu ban đầu đưa ra hay không,...

Chương 6: Kết luận và hướng phát triển: chương này em trình bày kết luận và các hạn chế và phát triển của đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN PIC 16F877A

Trên thị trường hiện nay có rất nhiều dòng vi điều khiển mà ta có thể lựa chọn để hỗ trợ cho chúng ta trong việc lập trình và nghiên cứu như PIC 16F8XX, Arduino, Raspberry Pi,... Để hoàn thành được mục tiêu đã đề ra trong đồ án 1 này em đã vận dụng được các kiến thức học được từ môn Vi xử lý và Thực tập vi xử lý, em đã quyết định chọn PIC 16F877A.

2.1.1. Vi điều khiển PIC 16F877A

Bộ vi điều khiển PIC16F877A là một trong những bộ vi điều khiển nổi tiếng nhất trong giới kỹ thuật điện tử. Bộ vi điều khiển này có nhiều ưu điểm thuận tiện cho việc sử dụng, lập trình chương trình điều khiển cũng rất dễ dàng.

Một trong các ưu điểm nổi bật là có thể ghi xóa nhiều lần, vì có công nghệ bộ nhớ Flash. Nó có tất cả 40 chân trong đó có 33 chân là I/O. PIC 16F877A được sử dụng nhiều trong các dự án có dùng vi điều khiển PIC.

Bạn có thể tìm thấy PIC 16F877A trên nhiều thiết bị điện tử. Được sử dụng cho các thiết bị điều khiển từ xa, thiết bị bảo mật và an ninh, các thiết bị tự động trong gia đình và một số các thiết bị trong công nghiệp.

Một EEPROM cũng được trang bị cho PIC 16F877A giúp lưu trữ một số thông tin vĩnh viễn như mã bộ truyền, tần số bộ nhận và các dữ liệu liên quan khác. Chi phí cho bộ vi điều khiển này thấp.

PIC 16F877A hoạt động linh hoạt được sử dụng trong nhiều lĩnh vực mà các vi điều khiển trước đây chưa được sử dụng, ví dụ trong các ứng dụng vi xử lý, chức năng hẹn giờ...

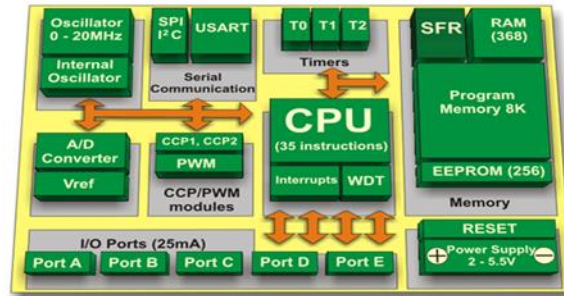


Hình 2. 1 Vi điều khiển PIC 16F877A - I/P thực tế

2.1.2. Kiến trúc của PIC 16F877A

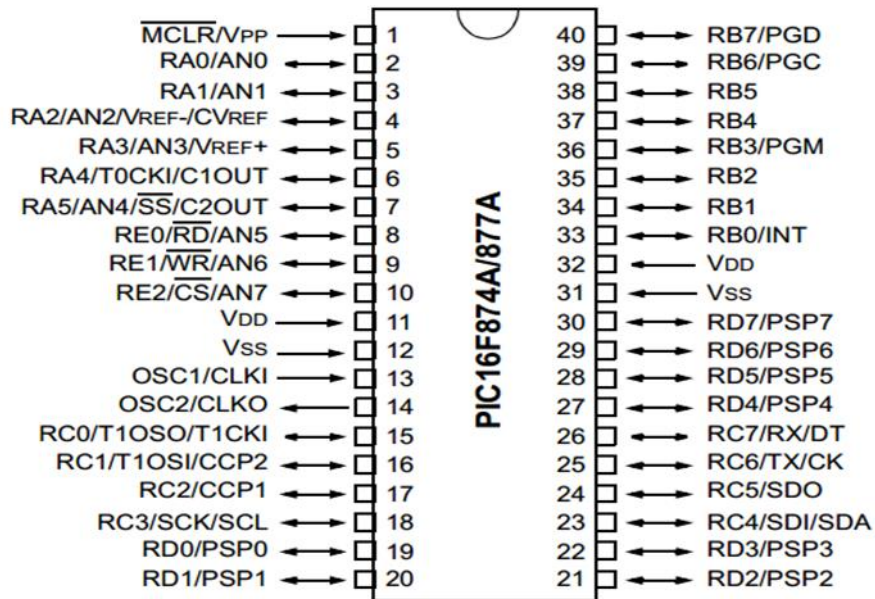
- Cấu trúc cơ bản của PIC 16F877A

- + 8K x 14 bits/word Flash ROM
- + 368 x 8 Bytes RAM.
- + 256 x 8 Bytes EEPROM.
- + 5 Port xuất/nhập (A, B, C, D, E) tương ứng 35 chân ra.
- + 2 Bộ định thời 8-bit Timer 0 và Timer 2.
- + 1 Bộ định thời 16-bit Timer 1, có thể hoạt động ở chế độ tiết kiệm năng lượng (SLEEP MODE) với nguồn xung clock ngoài.
- + 2 Bộ Capture/ Compare/ PWM (Bắt Giữ/ So Sánh/ Điều Biên Xung)
- + 1 Bộ biến đổi Analog to Digital 10 bit, 14 ngõ vào.
- + 2 Bộ so sánh tương tự (Comparator).
- + 1 Bộ định thời giám sát (Watch Dog Timer).
- + 1 Cổng giao tiếp song song 8 bit.
- + 1 Port nối tiếp.
- + 15 Nguồn ngắt (Interrupt).
- + Chế độ tiết kiệm năng lượng (Sleep Mode).
- + Tập lệnh gồm 35 lệnh có độ dài 14 bit.
- + Tần số hoạt động tối đa 20 MHz.
- + PIC16F877A có các chân với sự phân chia cấu trúc như sau:
 1. Có 5 port xuất/nhập.
 2. Có 8 kênh chuyển đổi A/D 10-bit.
 3. Có 2 bộ PWM.
 4. Có 3 bộ định thời: timer0, timer1 và timer2.
 5. Có giao tiếp truyền nối tiếp: chuẩn RS 232, I2C...
 6. Có giao tiếp LCD.



Hình 2. 2 Cấu trúc cơ bản của vi điều khiển PIC 16F877A

2.1.3. Sơ đồ nối chân và chức năng của từng chân



Hình 2. 3 Sơ đồ chân vi điều khiển PIC 16F877A

Hình trên cho thấy vi điều khiển có 36 chân tín hiệu điều khiển và 4 chân nguồn cấp cho vi điều khiển với mức điện áp là 5V. Tín hiệu điều khiển là 5V.

Bảng 2. 1 Bảng chức năng của các chân tín hiệu PIC 16F877A

Chân điều khiển	Chức năng của các chân
Chân MCLR/Vpp	MCLR: là ngõ vào reset tích cực mức thấp. Vpp: ngõ vào nhận điện áp khi ghi dữ liệu vào bộ nhớ nội flash
Chân RA0/AN0	RA0: xuất / nhập số AN0: ngõ vào tương tự của kênh 0

ĐỒ ÁN 1

Chân RA1/NA	RA1: xuất / nhập số AN1: ngõ vào tương tự kênh 1
Chân RA2/AN2/VREF- /CVREF	RA2: xuất / nhập số AN2: ngõ vào tương tự của kênh thứ 2 VREF-: ngõ vào điện áp chuẩn (thấp) của bộ ADC. CVREF: điện áp tham chiếu VREF ngõ ra bộ so sánh
Chân RA3/AN3/VREF+	RA3: xuất / nhập số AN3: ngõ vào tương tự kênh 3 VREF+: ngõ vào điện áp chuẩn (cao) của bộ A/D
Chân RA4/TOCKI/C1OUT	RA4: xuất / nhập số - mở khi được cấu tạo là ngõ ra TOCKI: ngõ vào xung clock từ bên ngoài cho Timer0 C1OUT: ngõ ra bộ so sánh 1
Chân RA5/AN4/ SS /C2OUT	RA5: xuất / nhập số AN4: ngõ vào tương tự kênh 4 SS: ngõ vào chọn lựa SPI phụ C2OUT: ngõ ra bộ so sánh 2
Chân RE0/ RD /AN5	RE0: xuất / nhập số RD : điều khiển đọc port song song AN5: ngõ vào tương tự kênh 5
Chân RE1/ WR /AN6	RE1: xuất / nhập số WR : điều khiển ghi port song song AN6: ngõ vào tương tự kênh 6 Chân RE2/ CS /AN7:
RE2: xuất / nhập số	RE2: xuất / nhập số CS : Chip chọn lựa điều khiển port song song AN7: ngõ vào tương tự kênh 7

ĐỒ ÁN 1

Chân OSC1/CLKI	OSC1: ngõ vào dao động thạch anh hoặc ngõ vào nguồn xung ở bên ngoài. Ngõ vào có mạch Schmitt Trigger nếu sử dụng dao động RC CLKI: ngõ vào nguồn xung bên ngoài
Chân OSC2/CLKO:	OSC2: ngõ ra dao động thạch anh. Kết nối đến thạch anh hoặc bộ cộng hưởng CLKO: ở chế độ RC, ngõ ra của OSC2, bằng $\frac{1}{4}$ tần số của OSC1 và chính là tốc độ của chu kì lệnh
Chân RC0/T1OSO/T1CKI	RC0: xuất / nhập số T1OSO: ngõ ra của bộ dao động Timer1 T1CKI: ngõ vào xung Clock từ bên ngoài Timer1
Chân RC1/T1OSI/CCP2	RC1: xuất / nhập số T1OSI: ngõ vào của bộ dao động Timer1 CCP2: ngõ vào Capture2, ngõ ra Compare2, ngõ ra PWM2
Chân RC2/CCP1	RC2: xuất / nhập số CCP1: ngõ vào Capture1, ngõ ra Compare1, ngõ ra PWM1
Chân RC3/SCK/SCL	SCK: ngõ vào xung clock nối tiếp đồng bộ/ngõ ra của chế độ SPI SCL: ngõ vào xung clock nối tiếp đồng bộ/ngõ ra của chế độ I2C
Chân RD0/PSP0	RD0: xuất / nhập số PSP0 : dữ liệu port song song
Các chân RD1/PSP1, RD2/PSP2, RD3/PSP3, RD4/PSP4, RD5/PSP5, RD6/PSP6, RD7/PSP7	Chức năng tương tự chân RD0/PSP0
Chân RB0/INT	RB0: xuất / nhập số INT: ngõ vào nhận tín hiệu ngắt ngoài

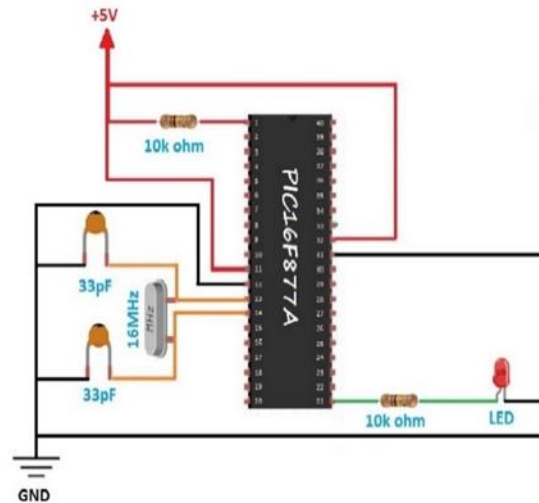
ĐỒ ÁN 1

Chân RB1, RB2, RB4, RB5	Xuất / nhập số
Chân RB3/PGM	RB3: xuất / nhập số PGM: chân cho phép lập trình điện áp thấp ICSP
Chân RB6/PGC, RB7/PGD	RB6: xuất / nhập số PGC: mạch gỡ rối và xung clock cho lập trình ICSP RB7: xuất / nhập số PGD: mạch gỡ rối và dữ liệu cho lập trình ICSP
Chân RC4/SDI/SDA	RC4: xuất / nhập số SDI: dữ liệu vào SPI SDA: xuất nhập dữ liệu I2C
Chân RC5/SDO	SDO: dữ liệu ra SPI
Chân RC6/TX/CK	TX: truyền bất đồng bộ USART CK: xung đồng bộ USART
Chân RC7/RX/DT	RC7: xuất / nhập số RX: nhận dữ liệu bất đồng bộ USART DT: dữ liệu đồng bộ USART
Chân Vdd và chân Vss	Là các chân nguồn của PIC

- Thông số kỹ thuật PIC16F877A

- + Điện áp hoạt động 4V - 5.5V.
- + Nhiệt độ hoạt động: -40 độ C - 85 độ C.
- + Tần số: 20MHz.
- + Loại bộ nhớ chương trình: FLASH.
- + Đầu I / O: 36.
- + Dung lượng bộ nhớ chương trình 14KB.

- Mạch ứng dụng của PIC 16F877A



Hình 2. 4 Mạch ứng dụng PIC 16F877A

Mạch khi ứng dụng thực tế thì cần phải có tụ thạch anh để chạy chương trình trong PIC 16F877A. Hình 2.4 trên là mạch điều khiển 1 đèn led và có 1 tụ thạch anh 16MHZ. Tụ thạch anh 16MHZ là tự tạo chu kì thực hiện 1 dòng lệnh như là 16MHZ thì 1 dòng lệnh thực hiện mất thời gian là $1/16\text{MHZ} = 0.0000000625\text{s}$.

2.2. CẢM BIẾN NHIỆT ĐỘ IC DS18B20

2.2.1. Giới thiệu chung về DS18B20

DS18B20 là một cảm biến nhiệt độ kỹ thuật số được sản xuất bởi Maxim Integrated. Nó sử dụng giao tiếp 1-Wire đơn giản và có nhiều ưu điểm như độ chính xác cao, độ phân giải cao, giá rẻ và dễ sử dụng.

2.2.2. Sơ đồ chân của DS18B20

Bảng 2. 2 Chức năng của các chân DS18B20

Số chân	Tên chân	Mô tả
1	Ground	Chân nối đất
2	Vcc	Chân cấp nguồn cho cảm biến , có thể là 3.3V hoặc 5V
3	Data	Chân xuất đầu ra giá trị nhiệt độ có thể đọc được bằng giao thức giao tiếp 1-wire



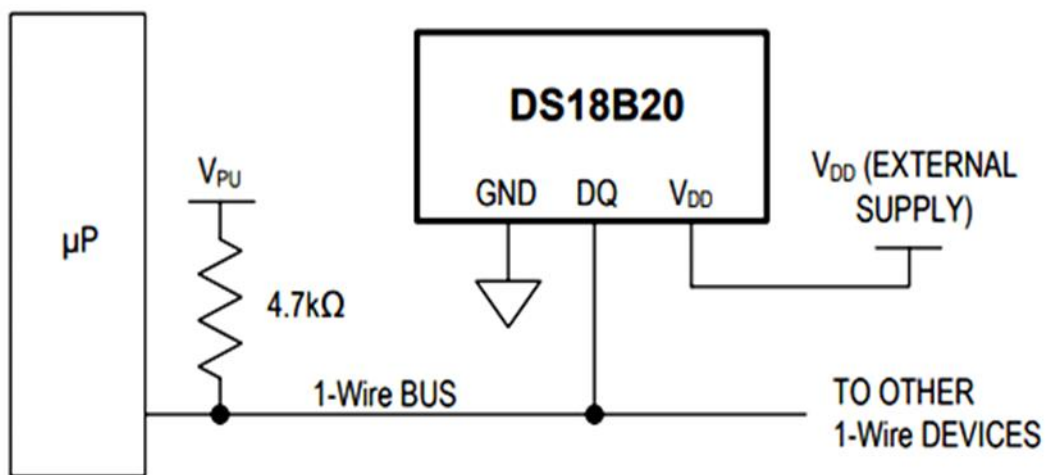
Hình 2. 5 Sơ đồ chân DS18B20

- Thông số kỹ thuật:

- + Dải đo nhiệt độ: -55°C đến $+125^{\circ}\text{C}$
- + Độ chính xác: $\pm 0,5^{\circ}\text{C}$ (trong phạm vi -10°C đến $+85^{\circ}\text{C}$)
- + Độ phân giải: 12bit
- + Giao thức: 1-Wire
- + Nguồn điện áp: 3V đến 5,5V

- Cách sử dụng cảm biến DS18B20:

Cảm biến hoạt động với giao thức 1-wire. Nó yêu cầu chân data được kết nối với bộ vi điều khiển bằng một điện trở pull-up và hai chân còn lại được sử dụng để cấp nguồn như hình dưới đây.



Hình 2. 6 Sơ đồ kết nối nguồn chung với vi xử lý của DS18B20

ĐỒ ÁN 1

Điện trở pull-up được sử dụng để giữ đường truyền dữ liệu ở trạng thái logic cao khi không sử dụng. Giá trị nhiệt độ do cảm biến đo được sẽ được lưu trong thanh ghi 2byte bên trong cảm biến.

Dữ liệu này có thể được đọc bằng cách sử dụng giao thức 1-wire bằng cách gửi theo một chuỗi dữ liệu. Có hai lệnh được gửi để đọc các giá trị, một là lệnh ROM và lệnh kia là lệnh function. Giá trị địa chỉ của mỗi bộ nhớ ROM theo trình tự như trong datasheet. Phải đọc datasheet để hiểu cách giao tiếp với cảm biến.

2.2.3. Một số chú ý khi sử dụng DS18B20

- DS18B20 có thể bị nhiễu bởi các nguồn điện áp cao
- DS18B20 không có khả năng chống nước
- Nên sử dụng điện trở kéo 4.4kΩ trên chân dữ liệu để đảm bảo tín hiệu ổn định
- Nên sử dụng tụ điện 100nF để lọc nhiễu nguồn
- Nên sử dụng dây dẫn ngắn để kết nối DS18B20 kết nối với vi điều khiển
- Nên đặt DS18B20 trong môi trường phù hợp với dải đo nhiệt độ của nó

2.3. IC THỜI GIAN THỰC DS1307

2.3.1. Giới thiệu về DS1307

DS1307 là một vi mạch đồng hồ thời gian thực (RTC) được sử dụng rộng rãi trong các ứng dụng điện tử. Vi mạch này có thể đếm thời gian chính xác với độ sai lệch thấp, giúp cho nó trở nên lý tưởng cho các thiết bị cần theo dõi thời gian thực

2.3.2. Sơ đồ chân của DS1307

Bảng 2. 3 Chức năng của các chân DS1307

Chân	Tên	Chức năng
1	Vcc	Chân nguồn dương
2	Vss	Chân nguồn âm
3	SQW/OUT	Chân đầu ra xung vuông
4	SCL	Chân đồng hồ I2C

ĐỒ ÁN 1

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

Hình 2. 8 Tổ chức thanh ghi bên trong của DS1307

	BIT7							BIT0	
00H	CH	10 SECONDS			SECONDS				00-59
	0	10 MINUTES			MINUTES				00-59
	0	12 24	10 HR A/P	10 HR	HOURS				01-12 00-23
	0	0	0	0	0	DAY			1-7
	0	0	10 DATE		DATE				01-28/29 01-30 01-31
	0	0	0	10 MONTH	MONTH				01-12
	10 YEAR				YEAR				00-99
	OUT	0	0	SQWE	0	0	RS1	RS0	
07H									

Hình 2. 9 Tổ chức thanh ghi được mã hoá theo bit bên trong của DS1307

- Thanh ghi giây (0x00): Đây là thanh ghi giây của DS1307. Nhìn trên bảng trên ta thấy được từ bit 0 đến bit 3 là dùng để mã hóa số BCD hàng đơn vị của giây. Tiếp theo từ bit 4 đến bit 6 dùng để mã hóa BCD hàng chục của giây. Tại sao nó chỉ sử dụng có 3bit này là do giây của chúng ta lớn nhất chỉ đến 59 nên hàng chục lớn nhất là 5 nên chỉ cần 3 thanh ghi này là cũng đủ mã hóa rồi! Còn bit thứ 7 có tên là “CH” theo tôi nó có nghĩa là “Clock Halt – Treo đồng hồ” Do đó nếu mà bit 7 này mà được đưa lên 1 tức là khóa đồng hồ nên do đó nó vô hiệu hóa chip và chip không hoạt động. Nên do vậy lúc nào cũng phải cho bit 7 này luôn xuống 0 từ lúc đầu (cái này sử dụng lệnh end với 0x7F)

ĐỒ ÁN 1

- Thanh ghi phút (0x01): Đây là thanh ghi phút của DS1307. Cũng nhìn trên bảng thanh ghi này được tổ chức như thanh ghi giây. Cũng là 3bit thấp dùng để mã hóa BCD chữ số hàng đơn vị và số hàng chục chỉ lớn nhất là 5 nên do đó chỉ cần dùng từ bit 4 đến bit 6 để mã hóa BCD tiếp chữ số hàng chục. Nhưng thanh ghi này có sự khác biệt với thanh ghi giây là bit 7 nó đã mặc định bằng 0 rồi nên do đó chúng ta không phải làm gì với bit 7 mà kệ nó!
- Thanh ghi giờ (0x02): Đây là thanh ghi giờ của DS1307 và tôi thấy thanh ghi này được coi là phức tạp nhất vì nó lằng nhằng nhưng mà nhìn bảng thì thấy các tổ chức của nó cũng hợp lý. Trước tiên chúng ta thấy được rằng từ bit 0 đến bit 3 nó dùng để mã hóa BCD của chữ số hàng đơn vị của giờ. Nhưng mà giờ nó còn có chế độ 24h và 12h nên do đó nó phức tạp ở các bit cao (bit 4 đến bit 7) và sự chọn chế độ 12h và 24h nó lại nằm ở bit 6. Nếu bit 6=0 thì ở chế độ 24h thì do chữ số hàng chục lớn nhất là 2 nên do đó nó chỉ dùng 2bit (bit 4 và bit 5) để mã hóa BCD chữ số hàng chục của giờ. Nếu bit 6 =1 thì chế độ 12h được chọn nhưng do chữ số của hàng chục của giờ trong chế độ này chỉ lớn nhất là 1 nên do đó bit thứ 4 là đủ để mã hóa BCD chữ số hàng chục của giờ rồi nhưng mà bit thứ 5 nó lại dùng để chỉ buổi sáng hay chiều, nếu mà bit 5 = 0 là AM và bit 5 =1 là PM. Trong cả 2 chế độ 12h và 24h thì bit 7 =0 nên ta ko cần chú ý đến thanh ghi này.
- Thanh ghi thứ (0x03): Đây là thanh ghi thứ trong tuần của DS1307 và thanh ghi này khá là đơn giản trong DS1307. Nó dùng số để chỉ thứ trong tuần nên do đó nó chỉ lấy từ 1 đến 7 tương đương từ thứ hai đến chủ nhật. Nên do đó nó dùng 3bit thấp (bit 0 đến bit 2) để mã hóa BCD ra thứ trong ngày. Còn các bit từ 3 đến 7 thì nó mặc định bằng 0 và ta không làm gì với các bit này!
- Thanh ghi ngày (0x04): Đây là thanh ghi ngày trong tháng của DS1307. Do trong các tháng có số ngày khác nhau nhưng mà nằm trong khoảng từ 1 đến 31 ngày. Do đó thanh ghi này các bit được tổ chức khá là đơn giản. Nó dùng 4bit thấp (bit0 đến bit 3) dùng để mã hóa BCD ra chữ số hàng đơn vị của ngày trong tháng. Nhưng do chữ số hàng chục của ngày trong tháng chỉ lớn nhất là 3 nên chỉ dùng bit 4 và bit 5 là đủ mã hóa BCD rồi. Còn bit 6 và bit 7 chúng ta không làm gì và nó mặc định bằng 0.

ĐỒ ÁN 1

- Thanh ghi tháng (0x05): Đây là thanh ghi tháng trong năm của DS1307. Tháng trong năm chỉ có từ 1 đến 12 tháng nên việc tổ chức trong bit cũng tương tự như ngày trong tháng nên do cũng 4bit thấp (từ bit 0 đến bit 3) mã hóa BCD hàng đơn vị của tháng. Nhưng do hàng chục chỉ lớn nhất là 1 nên chỉ dùng 1bit thứ 4 để mã hóa BCD ra chữ số hàng chục và các bit còn lại từ bit 5 đến bit 7 thì bỏ trống và nó mặc định cho xuống mức 0.
- Thanh ghi năm (0x06): Đây là thanh ghi năm trong DS1307. DS1307 chỉ có 100 năm thôi tương đương với 00 đến 99 nên nó dùng tất cả các bit thấp và bit cao để mã hóa BCD ra năm!
- Thanh ghi điều khiển (0x07): Đây là thanh ghi điều khiển quá trình ghi của DS1307 và Quá trình ghi phải được kết thúc bằng địa chỉ 0x93.

2.3.4. Cách hoạt động và một số chú ý khi sử dụng DS1307

- Cách sử dụng:

- + Cung cấp điện áp: DS1307 cần được cung cấp điện áp từ 2.4V đến 5.5V. Chân Vcc được sử dụng để kết nối với nguồn điện dương, và chân Vss được sử dụng để kết nối với đất.

- + Giao tiếp I2C: DS1307 sử dụng giao thức I2C để giao tiếp với vi điều khiển. Vi điều khiển sẽ đóng vai trò là thiết bị chủ, và DS1307 sẽ đóng vai trò là thiết bị phụ. Hai chân SCL và SDA được sử dụng để truyền dữ liệu giữa vi điều khiển và DS1307.

- + Thiết lập thời gian: Vi điều khiển cần gửi dữ liệu thời gian đến DS1307 để thiết lập thời gian ban đầu. Dữ liệu thời gian được lưu trữ trong các thanh ghi của DS1307.

- + Đọc thời gian: Vi điều khiển có thể đọc dữ liệu thời gian từ DS1307 bất cứ lúc nào. Dữ liệu thời gian được đọc từ các thanh ghi của DS1307.

- + Báo thức: DS1307 có thể được cấu hình để tạo ra một báo thức khi thời gian đạt đến một giá trị cụ thể. Vi điều khiển có thể thiết lập thời gian báo thức và cấu hình các tùy chọn báo thức.

- + Pin dự phòng: DS1307 có thể được sử dụng với pin Lithium để duy trì thời gian khi mất nguồn chính. Pin Lithium được kết nối với chân Vbat của DS1307.

- Chú ý:

- + DS1307 là một vi mạch CMOS, cần được bảo vệ khỏi tĩnh điện.

ĐỒ ÁN 1

- + Việc hàn DS1307 cần được thực hiện cẩn thận để tránh làm hỏng vi mạch.

2.4. HIỂN THỊ - LCD 16x02

Có nhiều cách để hiển thị thời gian như màn hình OLED, Led 7 đoạn, Ma trận Led đơn, LCD,... Các linh kiện đều có những ưu điểm và nhược điểm khác nhau về cách hiển thị và thi công mạch in. Vì vậy, để đáp ứng được đề tài mà em đang làm thì em đã sử dụng LCD 16x2.

Màn hình text LCD1602 xanh lá sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ sử dụng thích hợp cho những người mới học và làm dự án.



Hình 2. 10 Màn hình LCD 16x2

- Thông số kỹ thuật:

- + Điện áp hoạt động là 5 V.
- + Kích thước: 80 x 36 x 12.5 mm.
- + Chữ đen, nền xanh lá.
- + Khoảng cách giữa hai chân kết nối là 0.1-inch tiện dụng khi kết nối với Breadboard.
- + Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- + Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.

- + Có thể được điều khiển với 6 dây tín hiệu.

- Sơ đồ chân và chức năng của các chân:

Bảng 2. 4 Sơ đồ chân LCD 16x2

Chân	Tên	Chức năng
1	VSS	Nối đất
2	VDD	Cung cấp điện áp (5V)
3	Vo	Điều chỉnh độ tương phản
4	RS	Chọn chế độ dữ liệu/lệnh
5	R/W	Chọn chế độ đọc/ghi
6	E	Kích hoạt lệnh
7-14	D0-D7	Dữ liệu (8 bit)
15	LED+	Anode của đèn nền
16	LED-	Cathode của đèn nền

- Trong 16 chân của LCD được chia ra làm 3 dạng tín hiệu như sau:

+ Các chân cấp nguồn: Chân số 1 là chân nối mass (0V), chân thứ 2 là Vdd nối với nguồn +5V. Chân thứ 3 dùng để chỉnh contrast thường nối với biến trở.

+ Các chân điều khiển: Chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.

+ Các chân dữ liệu D7÷D0: Chân số 7 đến chân số 14 là 8 chân dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.

- Địa chỉ cho LCD

Bảng 2. 5 Địa chỉ của LCD 16x2

ĐC Hàng 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
ĐC Hàng 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3
ĐC Hàng 3	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7
ĐC Hàng 4	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7

2.5. IC 7805

LM7805 hay còn được gọi là IC 7805, đây là loại IC điều chỉnh nguồn điện áp đầu ra +5V. IC 7805 thuộc dòng IC ổn áp dương thuộc họ LM78xx, sản xuất trong gói TO-220. Đây là dòng IC được dùng rộng rãi hiện nay trong các mạch điện tử và những thiết bị điện tử thương mại.

IC 7805 có nhiều tính năng khác nhau được tích hợp trong những ứng dụng điện tử như là dòng điện đầu ra 1,5A, bảo vệ quá nhiệt, bảo vệ quá tải, dòng điện tĩnh thấp, giữ cho các transistor công suất có thể hoạt động an toàn và ổn định,...

Thông số kỹ thuật Transistor 7805:

- Datasheet: 7805
- Số chân: 3
- Điện áp ở ngõ ra: 5v
- Điện áp ở ngõ vào: 7V – 18V DC
- Dòng điện ngõ ra: 1A
- Mức nhiệt độ hoạt động: 0 độ C – 125 độ C
- Công suất làm việc cực đại: 5W

Sơ đồ IC ổn áp 7805

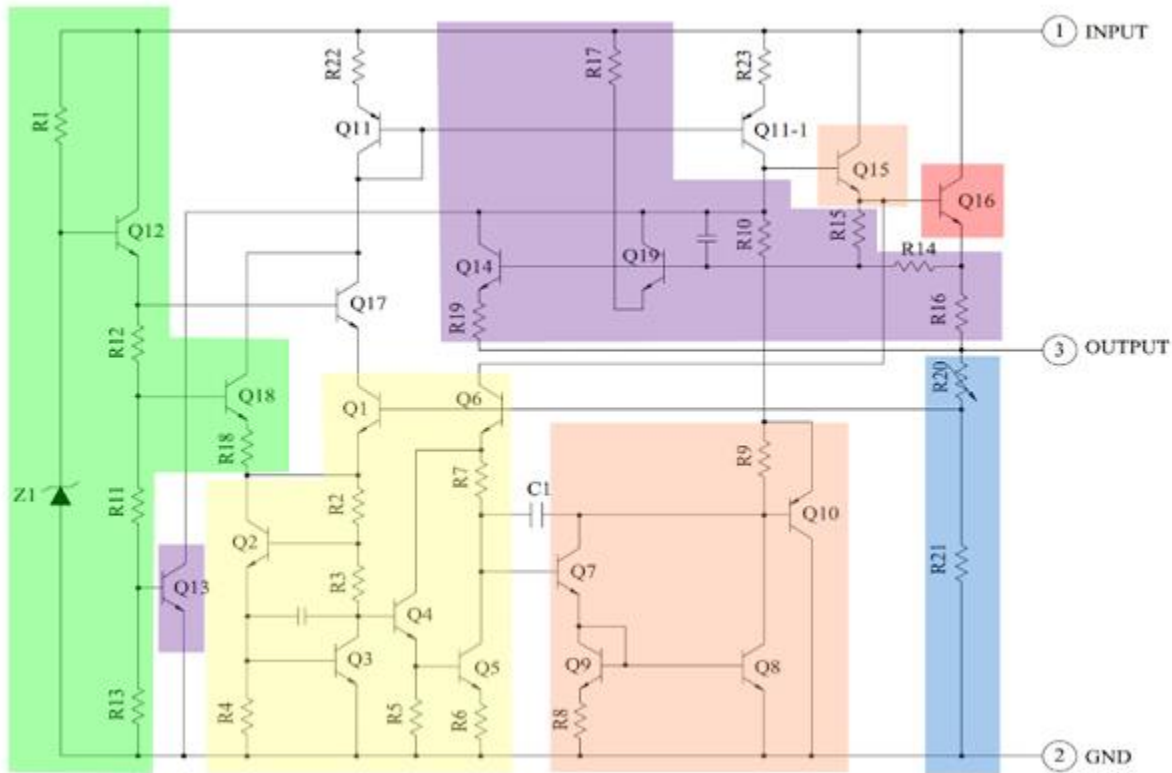
Trung tâm của IC 7805 là một bóng bán dẫn (Q16) có thể điều khiển dòng điện giữa đầu vào và đầu ra. Chúng còn có thể điều khiển được điện áp đầu ra. Tham chiếu bandgap (màu vàng) giữ giúp cho điện áp luôn được ổn định. Chúng lấy điện áp đầu ra đã được chia tỷ lệ làm đầu vào (Q1 và Q6) và cũng cung cấp những tín hiệu lỗi (đến Q7) để chỉ báo nếu phát hiện mức điện áp quá cao hoặc quá thấp. Nhiệm vụ chính của bandgap đó chính là cung cấp tham chiếu ổn định và chính xác, ngay cả trong trường hợp nhiệt độ của con chip thay đổi.

Tín hiệu lỗi từ tham chiếu bandgap sẽ được khuếch đại bởi bộ khuếch đại lỗi (màu cam). Tín hiệu khuếch đại này sẽ điều khiển bóng bán dẫn đầu ra thông qua Q15. Chúng sẽ tiến hành đóng vòng phản hồi tiêu cực điều khiển điện áp đầu ra.

Mạch khởi động (màu xanh lá cây) giúp cung cấp dòng điện ban đầu cho mạch bandgap. Mạch màu tím mang đến khả năng bảo vệ chống quá nhiệt (Q13), điện áp đầu vào quá mức

ĐỒ ÁN 1

(Q19), dòng điện đầu ra quá mức (Q14). Các mạch này sẽ làm giảm dòng điện đầu ra hoặc tắt bộ điều chỉnh, bảo vệ cho chúng khỏi bị hư hại trong trường hợp xảy ra lỗi. Bộ chia điện áp (màu xanh lam) giảm điện áp trên chân đầu ra để sử dụng nhờ tham chiếu bandgap.



Hình 2. 11 Hình 7805 dạng transistor

2.6. Tụ điện

Tụ điện có hai chân có thể có phân cực hoặc không phân cực, nếu tụ có phân cực cần cấp đúng điện áp để tụ có thể hoạt động (cực dương có hiệu điện thế cao hơn cực âm)

Về cơ bản tụ là một linh kiện điện tử dùng để chứa điện tích. Nó bao gồm hai bản dẫn điện ngăn cách bởi một lớp cách điện (điện môi). Khi hai bản được tích điện trái dấu, tụ sẽ tạo ra một điện trường. Do đó, giữ hai đầu tụ sẽ tồn tại một điện áp. Điện áp giữa hai đầu tụ được tính bằng

$$C = \frac{Q}{U}$$

U: là điện áp giữa hai đầu tụ (V)

Q: là điện tích mà tụ tích được (điện tích của bản dương - đo bằng Coulomb)

C: điện dung của tụ (biểu thị độ chứa điện - đo bằng Farad)

Phân loại tụ điện theo đặc điểm cấu tạo

- **Tụ điện gốm (tụ đất):** loại tụ này được làm từ ceramic, vỏ ngoài của tụ thương được bọc keo hoặc nhuộm màu. Các loại gốm thường được sử dụng cho loại tụ này là COG, X7R, Z5U v.v...
- **Tụ gốm đa lớp:** là loại tụ gốm có nhiều lớp bản cực cách điện bằng gốm. Tụ này đáp ứng cao tần và điện áp cao hơn loại tụ gốm "thường" khoảng 4-->5 lần
- **Tụ giấy:** Là tụ điện có bản cực là các lá nhôm hoặc thiếc cách nhau bằng lớp giấy tẩm dầu cách điện làm dung môi.
- **Tụ mica màng mỏng:** cấu tạo với các lớp điện môi là mica nhân tạo hay nhựa có cấu tạo màng mỏng (thin film) như Mylar, Polycarbonate, Polyester, Polystyrene (ổn định nhiệt 150 ppm / C)
- **Tụ bạc - mica:** là loại tụ điện mica có bản cực bằng bạc, khá nặng. Điện dung từ vài pF đến vài nF, độ ổn nhiệt rất bé. Loại tụ này chuyên dùng cho các mạch cao tần
- **Tụ hóa học:** Là tụ giấy có dung môi hóa học đặc hiệu --> tạo điện dung cao và rất cao cho tụ điện. Nếu bên ngoài có vỏ nhôm bọc nhựa thì còn gọi là tụ nhôm
- **Tụ siêu hóa (Super Chemical Capacitance):** dùng dung môi đất hiếm, tụ này năng hơn tụ nhôm hóa học và có trị số cực lớn, có thể đến hàng Farad. Tụ có thể dùng như một nguồn pin cao cấp cho vi xử lý hay các mạch đồng hồ (Clock) cần cấp điện liên tục
- **Tụ hóa sinh** hay còn gọi là **siêu tụ điện** thay thế cho pin trong việc lưu trữ điện năng trong các thiết bị điện tử di động, dùng Iginate trong tạo biến nầu làm nền cho dung môi --> lượng điện tích trữ siêu lớn và giả chỉ 15% sau mỗi chu kỳ 10.000 lần sạc
- **Tụ tantalium:** tụ này có bản cực nhôm và dùng gel tantal làm dung môi, có trị số rất lớn với có thể tích nhỏ



Hình 2. 12 Hình các loại tụ điện

2.7. Điện trở

Khái niệm: Điện trở là một linh kiện điện tử thụ động trong mạch điện có biểu tượng R. Nó là đại lượng vật lý đặc trưng cho tính chất cản trở dòng điện của vật liệu. Điện trở kháng được định nghĩa là tỉ số của hiệu điện thế giữa hai đầu vật thể đó với cường độ dòng điện đi qua nó

Công thức tính: $R=U/I$. Trong đó :

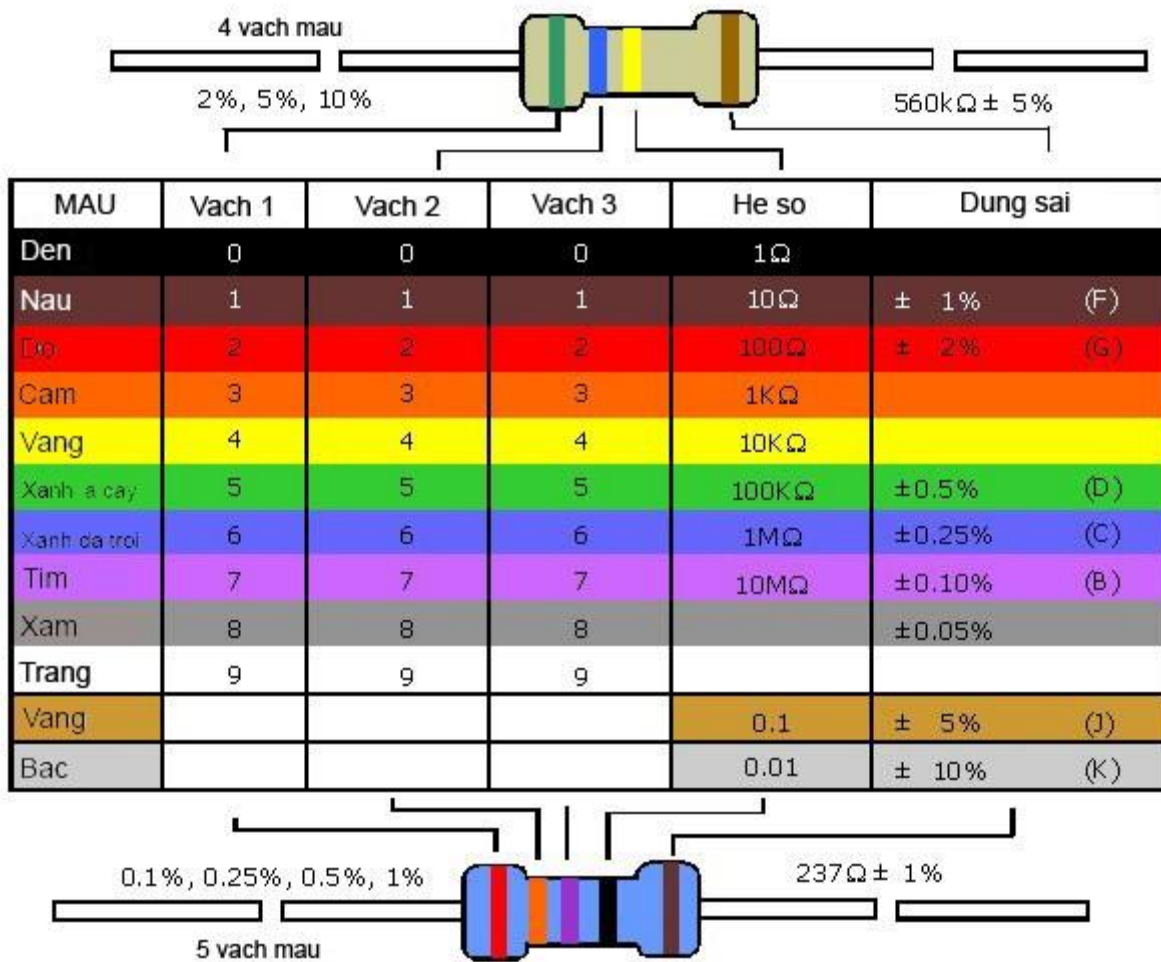
U: là hiệu điện thế giữa hai đầu vật dẫn điện, đo bằng vôn (V).

I: là cường độ dòng điện đi qua vật dẫn điện, đo bằng ampe (A).

R: là điện trở của vật dẫn điện, đo bằng Ohm (Ω).

Do đó, bản chất nó là 1 sợi dây dẫn có điện trở rất lớn (thực ra lớn bé còn tùy thuộc vào giá trị của nó), điện trở không phân cực, tức là không phân biệt âm dương.

ĐỒ ÁN 1



Hình 2. 13 Hình vạch đọc điện trở

Mỗi điện trở có 1 giá trị nhất định, vòng màu in trên điện trở thể hiện giá trị của nó.

{Thông thường, điện trở có 4 vòng màu.}

- 2 vòng màu đầu là 2 chữ số đầu của giá trị.
- Vòng thứ 3 thể hiện số chữ số “0” đứng sau.
- Vòng thứ 4 thể hiện sai số.

Có tất cả 12 màu, mỗi màu có 1 giá trị khác nhau.

● Resistor color code



Example 1

(Brown=1),(Black=0),(Orange=3)

$$10 \times 10^3 = 10k \text{ ohm}$$

Tolerance(Gold) = $\pm 5\%$



Example 2

(Yellow=4),(Violet=7),(Black=0),
(Red=2)

$$470 \times 10^2 = 47k \text{ ohm}$$

Tolerance(Brown) = $\pm 1\%$

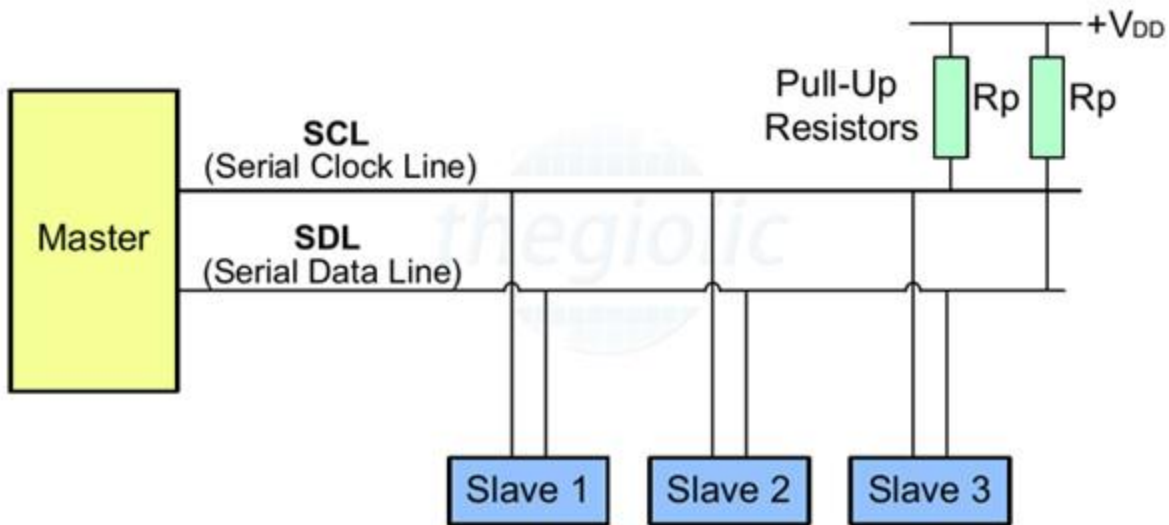
Color	Value	Multiplier	Tolerance (%)
Black	0	0	-
Brown	1	1	± 1
Red	2	2	± 2
Orange	3	3	± 0.05
Yellow	4	4	-
Green	5	5	± 0.5
Blue	6	6	± 0.25
Violet	7	7	± 0.1
Gray	8	8	-
White	9	9	-
Gold	-	-1	± 5
Silver	-	-2	± 10
None	-	-	± 20

Hình 2. 14 Hình ví dụ tính điện trở

2.8. MỘT SỐ LÝ THUYẾT CƠ BẢN ĐƯỢC SỬ DỤNG

2.8.1. I2C

I2C là một giao thức truyền thông nối tiếp được sử dụng để kết nối các thiết bị điện tử với nhau. Nó được phát triển bởi Philips Semiconductors (nay là NXP Semiconductors) vào đầu những năm 1980 và đã trở thành một tiêu chuẩn công nghiệp phổ biến.



Hình 2. 15 Hình i2c protocols

Giống như giao tiếp UART, I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị:

- SDA (Serial Data) - đường truyền cho master và slave để gửi và nhận dữ liệu.
- SCL (Serial Clock) - đường mang tín hiệu xung nhịp.

Các bit dữ liệu sẽ được truyền từng bit một dọc theo một đường duy nhất (SDA) theo các khoảng thời gian đều đặn được thiết lập bởi 1 tín hiệu đồng hồ (SCL).

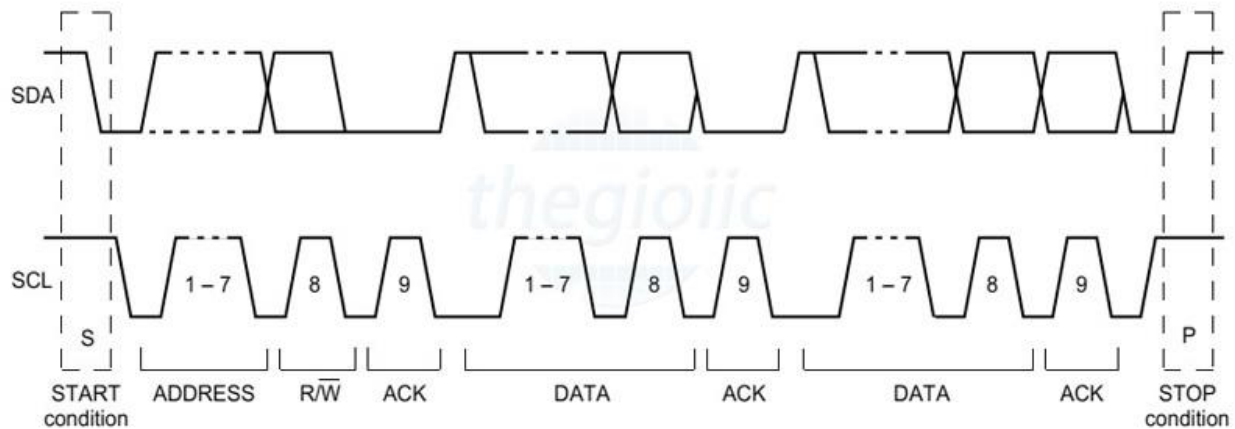
Cách hoạt động của I2C

Giao tiếp I2C bao gồm quá trình truyền nhận dữ liệu giữa các thiết bị chủ tớ, hay Master - Slave.

Thiết bị Master là 1 vi điều khiển, nó có nhiệm vụ điều khiển đường tín hiệu SCL và gửi nhận dữ liệu hay lệnh thông qua đường SDA đến các thiết bị khác.

Các thiết bị nhận các dữ liệu lệnh và tín hiệu từ thiết bị Master được gọi là các thiết bị Slave. Các thiết bị Slave thường là các IC, hoặc thậm chí là vi điều khiển.

Master và Slave được kết nối với nhau bằng hai đường bus SCL và SDA đều hoạt động ở chế độ Open Drain, nghĩa là bất cứ thiết bị nào kết nối với mạng I2C này cũng chỉ có thể kéo 2 đường bus này xuống mức thấp (LOW), nhưng lại không thể kéo được lên mức cao. Vì để tránh trường hợp bus vừa bị 1 thiết bị kéo lên mức cao vừa bị 1 thiết bị khác kéo xuống mức thấp gây hiện tượng ngắn mạch. Do đó cần có 1 điện trở (từ 1 – 4,7 kΩ) để giữ mạch định ở mức cao.



Hình 2. 16 Hình khung dữ liệu i2c

Với I2C, dữ liệu được truyền trong các tin nhắn. Tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của địa chỉ slave và một hoặc nhiều khung dữ liệu chứa dữ liệu đang được truyền. Thông điệp cũng bao gồm điều kiện khởi động và điều kiện dừng, các bit đọc / ghi và các bit ACK / NACK giữa mỗi khung dữ liệu:

- Điều kiện khởi động: Đường SDA chuyển từ mức điện áp cao xuống mức điện áp thấp trước khi đường SCL chuyển từ mức cao xuống mức thấp.
- Điều kiện dừng: Đường SDA chuyển từ mức điện áp thấp sang mức điện áp cao sau khi đường SCL chuyển từ mức thấp lên mức cao.
- Bit địa chỉ: Thông thường quá trình truyền nhận sẽ diễn ra với rất nhiều thiết bị, IC với nhau. Do đó để phân biệt các thiết bị này, chúng sẽ được gán 1 địa chỉ vật lý 7 bit cố định.
- Bit đọc / ghi: Bit này dùng để xác định quá trình là truyền hay nhận dữ liệu từ thiết bị Master. Nếu Master gửi dữ liệu đi thì ứng với bit này bằng '0', và ngược lại, nhận dữ liệu khi bit này bằng '1'.
- Bit ACK / NACK: Viết tắt của Acknowledged / Not Acknowledged. Dùng để so sánh bit địa chỉ vật lý của thiết bị so với địa chỉ được gửi tới. Nếu trùng thì Slave sẽ được đặt bằng '0' và ngược lại, nếu không thì mặc định bằng '1'.
- Bit dữ liệu: Gồm 8 bit và được thiết lập bởi thiết bị gửi truyền đến thiết bị nhận. Sau khi các bit này được gửi đi, lập tức 1 bit ACK/NACK được gửi ngay theo sau để

xác nhận rằng thiết bị nhận đã nhận được dữ liệu thành công hay chưa. Nếu nhận thành công thì bit ACK/NACK được set bằng '0' và ngược lại.

Quá trình truyền nhận

- Khi bắt đầu Master sẽ gửi đi 1 xung Start bằng cách kéo lần lượt các đường SDA, SCL từ mức 1 xuống 0.
- Tiếp theo đó, Master gửi đi 7 bit địa chỉ tới các Slave cùng với bit Read/Write.
- Slave sẽ so sánh địa chỉ vừa được gửi tới. Nếu trùng khớp, Slave sẽ xác nhận bằng cách kéo đường SDA xuống 0 và set bit ACK/NACK bằng '0'. Nếu không trùng khớp thì SDA và bit ACK/NACK đều mặc định bằng '1'.
- Thiết bị Master sẽ gửi hoặc nhận khung bit dữ liệu. Nếu Master gửi đến Slave thì bit Read/Write ở mức 0. Ngược lại nếu nhận thì bit này ở mức 1.
- Nếu như khung dữ liệu đã được truyền đi thành công, bit ACK/NACK được set thành mức 0 để báo hiệu cho Master tiếp tục.
- Sau khi tất cả dữ liệu đã được gửi đến Slave thành công, Master sẽ phát 1 tín hiệu Stop để báo cho các Slave biết quá trình truyền đã kết thúc bằng cách chuyển lần lượt SCL, SDA từ mức 0 lên mức 1.

Các chế độ hoạt động của I2C:

- Chế độ chuẩn (standard mode) với tốc độ 100 kBit/s.
- Chế độ tốc độ thấp (low speed mode) với tốc độ 10 kBit/s.

Khác với giao tiếp SPI chỉ có thể có 1 Master, giao tiếp I2C cho phép chế độ truyền nhận dữ liệu giữa nhiều thiết bị Master khác nhau với thiết bị Slave. Tuy nhiên quá trình này có hơi phức tạp vì thiết bị Slave có thể nhận 1 lúc nhiều khung dữ liệu từ các thiết bị Master khác nhau, điều đó đôi khi dẫn đến xung đột hoặc sai sót dữ liệu nhận được.

Để tránh điều đó, khi làm việc ở chế độ này, mỗi thiết bị Master cần phát hiện xem đường SDA đang ở trạng thái nào. Nếu SDA ở mức 0, nghĩa là đang có 1 thiết bị Master khác đang có quyền điều khiển và phải chờ đến khi truyền xong. Ngược lại nếu SDA ở mức 1, nghĩa là đường truyền SDA đã an toàn và có sử dụng.

2.8.2 UART

UART là viết tắt của Universal Asynchronous Receiver/Transmitter, là một giao thức truyền thông nối tiếp được sử dụng rộng rãi để truyền dữ liệu giữa các thiết bị điện tử. Giao thức này đơn giản và dễ sử dụng, chỉ cần hai dây tín hiệu để truyền và nhận dữ liệu.

UART là giao thức truyền thông không đồng bộ, nghĩa là không có tín hiệu xung nhịp chung giữa các thiết bị truyền và nhận. Dữ liệu được truyền theo từng bit, mỗi bit được biểu thị bằng một mức điện áp khác nhau. Tốc độ truyền dữ liệu được xác định bởi baud rate, được đo bằng bit trên giây (bps). UART có hai chế độ hoạt động: Simplex, chỉ truyền dữ liệu theo một chiều, và Duplex, cho phép truyền dữ liệu theo cả hai chiều.

- Khung dữ liệu:

- + Mỗi byte dữ liệu được truyền trong một khung dữ liệu.
- + Khung dữ liệu bao gồm các thành phần sau:
 - + Start bit: Bit bắt đầu, là một bit có mức điện áp cao, báo hiệu cho thiết bị nhận rằng dữ liệu sắp được truyền.
 - + Data bits: Các bit dữ liệu, bao gồm 8bit dữ liệu thực tế.
 - + Parity bit: Bit chẵn lẻ, được sử dụng để kiểm tra lỗi truyền dữ liệu.
 - + Stop bit: Bit dừng, là một bit có mức điện áp cao, báo hiệu cho thiết bị nhận rằng dữ liệu đã được truyền xong.

- Các loại UART:

Có hai loại UART chính:

- + UART phần cứng: Là một phần cứng được tích hợp sẵn trong vi điều khiển hoặc vi xử lý.
- + UART phần mềm: Là một phần mềm được viết trên vi điều khiển hoặc vi xử lý để mô phỏng chức năng của UART phần cứng.

- Ứng dụng:

UART được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

- + Kết nối máy tính với các thiết bị ngoại vi như máy in, chuột, bàn phím.
- + Kết nối các thiết bị nhúng với nhau.
- + Điều khiển các thiết bị điện tử từ xa.

ĐỒ ÁN 1

- Ưu điểm:

- + UART là giao thức đơn giản và dễ sử dụng.
- + UART chỉ cần hai dây tín hiệu để truyền và nhận dữ liệu.
- + UART có thể truyền dữ liệu với tốc độ cao.

- Nhược điểm:

- + UART không có khả năng chống nhiễu tốt.
- + UART không có khả năng kiểm tra lỗi dữ liệu cao.

=> Tóm lại, UART là một giao thức truyền thông nối tiếp đơn giản và dễ sử dụng, được sử dụng rộng rãi trong nhiều ứng dụng khác nhau.

2.8.3. 1-WIRE

Chuẩn giao tiếp One Wire là chuẩn giao tiếp không đồng bộ bán song công (half-duplex: tại một thời điểm, tín hiệu chỉ có thể chạy theo một hướng).

- Khởi tạo: Vi điều khiển gửi tín hiệu reset kéo dài ít nhất 480 micro giây, sau đó vi điều khiển chờ khoảng 60-240 micro giây để nhận phản hồi từ DS18B20. DS18B20 phản hồi bằng cách kéo dây dữ liệu xuống mức thấp trong 60-240 micro giây

- Viết dữ liệu (Write Data): Để gửi một bit dữ liệu:

- Bit 0: Dây dữ liệu kéo xuống mức thấp trong ít nhất 60 micro giây

- Bit 1: Dây dữ liệu kéo xuống mức thấp trong khoảng 1-15 micro giây, sau đó lên mức cao.

- Đọc dữ liệu (Read Data):

Vi điều khiển kéo dây dữ liệu xuống mức thấp trong ít nhất 1 micro giây rồi thả lên mức cao.. DS18B20 điều khiển dây dữ liệu trong khoảng 15 micro giây tiếp theo:

- Bit 0: Dây dữ liệu ở mức thấp.

- Bit 1: Dây dữ liệu ở mức cao.

Khung dữ liệu bao gồm :

- READ ROM (33h)

Cho phép đọc ra 8 byte mã đã khắc bằng laser trên ROM, bao gồm: 8 bit mã định tên linh kiện (10h), 48 bit số xuất xưởng, 8 bit kiểm tra CRC. Lệnh này chỉ dùng khi trên bus

ĐỒ ÁN 1

có 1 cảm biến DS1820, nếu không sẽ xảy ra xung đột trên bus do tất cả các thiết bị tó cùng đáp ứng.

- MATCH ROM (55h)

Lệnh này được gửi đi cùng với 64 bit ROM tiếp theo, cho phép bộ điều khiển bus chọn ra chỉ một cảm biến DS1820 cụ thể khi trên bus có nhiều cảm biến DS1820 cùng nối vào. Chỉ có DS1820 nào có 64 bit trên ROM trùng khớp với chuỗi 64 bit vừa được gửi tới mới đáp ứng lại các lệnh về bộ nhớ tiếp theo. Còn các cảm biến DS1820 có 64 bit ROM không trùng khớp sẽ tiếp tục chờ một xung reset. Lệnh này được sử dụng cả trong trường hợp có một cảm biến một dây, cả trong trường hợp có nhiều cảm biến một dây.

- SKIP ROM (CCh)

Lệnh này cho phép thiết bị điều khiển truy nhập thẳng đến các lệnh bộ nhớ của DS1820 mà không cần gửi chuỗi mã 64 bit ROM. Như vậy sẽ tiết kiệm được thời gian chờ đợi nhưng chỉ mang hiệu quả khi trên bus chỉ có một cảm biến.

- SEARCH ROM (F0h)

Lệnh này cho phép bộ điều khiển bus có thể dò tìm được số lượng thành viên tó đang được đấu vào bus và các giá trị cụ thể trong 64 bit ROM của chúng bằng một chu trình dò tìm.

- ALARM SEARCH (ECh)

Tiến trình của lệnh này giống hệt như lệnh Search ROM, nhưng cảm biến DS1820 chỉ đáp ứng lệnh này khi xuất hiện điều kiện cảnh báo trong phép đo nhiệt độ cuối cùng. Điều kiện cảnh báo ở đây được định nghĩa là giá trị nhiệt độ đo được lớn hơn giá trị TH và nhỏ hơn giá trị TL là hai giá trị nhiệt độ cao nhất và nhiệt độ thấp nhất đã được đặt trên thanh ghi trong bộ nhớ của cảm biến.

Sau khi thiết bị chủ (thường là một vi điều khiển) sử dụng các lệnh ROM để định địa chỉ cho các cảm biến một dây đang được đấu vào bus, thiết bị chủ sẽ đưa ra các lệnh chức năng DS1820. Bằng các lệnh chức năng thiết bị chủ có thể đọc ra và ghi vào bộ nhớ nháp (scratchpad) của cảm biến DS1820. khởi tạo quá trình chuyển đổi giá trị nhiệt độ đo được và xác định chế độ cung cấp điện áp nguồn. Các lệnh chức năng có thể được mô tả ngắn gọn như sau:

ĐỒ ÁN 1

- WRITE SCRATCHPAD (4Eh)

Lệnh này cho phép ghi 2 byte dữ liệu vào bộ nhớ nháp của DS1820. Byte đầu tiên được ghi vào thanh ghi TH (byte 2 của bộ nhớ nháp) còn byte thứ hai được ghi vào thanh ghi TL (byte 3 của bộ nhớ nháp). Dữ liệu truyền theo trình tự đầu tiên là bit có ý nghĩa nhất và kế tiếp là những bit có ý nghĩa giảm dần. Cả hai byte này phải được ghi trước khi thiết bị chủ xuất ra một xung reset hoặc khi có dữ liệu khác xuất hiện.

- READ SCRATCHPAD (BEh)

Lệnh này cho phép thiết bị chủ đọc nội dung bộ nhớ nháp. Quá trình đọc bắt đầu từ bit có ý nghĩa nhất của byte 0 và tiếp tục cho đến byte thứ 9 (byte 8 – CRC). Thiết bị chủ có thể xuất ra một xung reset để làm dừng quá trình đọc bất kỳ lúc nào nếu như chỉ có một phần của dữ liệu trên bộ nhớ nháp cần được đọc.

- COPYSCRATCHPAD (48h)

Lệnh này copy nội dung của hai thanh ghi TH và TL (byte 2 và byte 3) vào bộ nhớ EEPROM. Nếu cảm biến được sử dụng trong chế độ cấp nguồn 1 bắt đầu việc đo.

- CONVERT T (44h)

Lệnh này khởi động một quá trình đo và chuyển đổi giá trị nhiệt độ thành số (nhị phân). Sau khi chuyển đổi giá trị kết quả đo nhiệt độ được lưu trữ trên thanh ghi nhiệt độ 2 byte trong bộ nhớ nháp Thời gian chuyển đổi không quá 200 ms, trong thời gian đang chuyển đổi nếu thực hiện lệnh đọc thì các giá trị đọc ra đều bằng 0.

- READ POWER SUPPLY (B4h)

Một lệnh đọc tiếp sau lệnh này sẽ cho biết DS1820 đang sử dụng chế độ cấp nguồn như thế nào, giá trị đọc được bằng 0 nếu cấp nguồn bằng chính đường dẫn dữ liệu và bằng 1 nếu cấp nguồn qua một đường dẫn riêng.

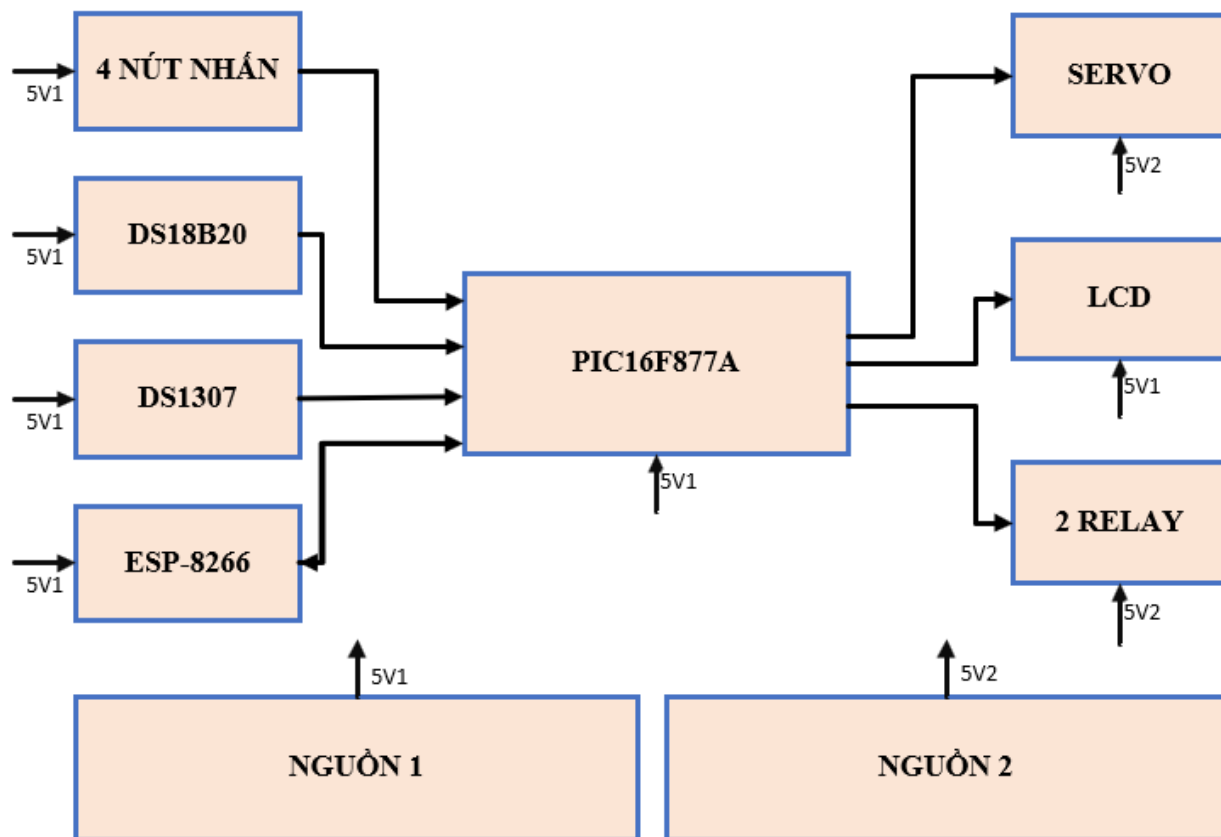
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.1. GIỚI THIỆU

Chương này là những tính toán về hoạt động, nguồn và phạm vi hoạt động của hệ thống, cách thiết kết, tính toán các linh kiện cần thiết cho hệ thống.

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

Do đề tài em chọn là hệ thống kiểm soát hồ cá theo nhiệt độ và có thể giám sát trên ứng dụng di động nên hệ thống có khối truyền thông, khối điều khiển trung tâm, khối đầu vào, khối ngõ ra.

Theo hình 3.1 ta thấy hệ thống gồm có 10 khối: khối 4 nút nhấn, khối ds18b20, khối ds1307, khối esp-8266, khối pic16f877a, khối servo, khối lcd, khối 2 relay, khối nguồn 1 và khối nguồn 2.

ĐỒ ÁN 1

- Chức năng của các khối:

+ Khối 4 nút nhấn: Chức năng cài đặt thời gian thực, thời gian cho cá ăn, nhiệt độ giới hạn, chọn menu cài đặt cho hệ thống.

+ Khối DS18B20: Chức năng đọc nhiệt độ từ cảm biến ds18b20 qua chuẩn 1 dây.

+ Khối DS1307: Chức năng đọc thời gian thực từ cảm biến ds1307 qua chuẩn i2c.

+ Khối ESP-8266: Chức năng giao tiếp với PIC-16f877A để lưu dữ liệu về thời gian cho cá ăn, nhiệt độ giới hạn, điều khiển servo cho cá ăn lên firebase và gửi dữ liệu từ ESP-8266 để cài đặt thời gian cho cá ăn, cài đặt giới hạn nhiệt độ, tín hiệu điều khiển servo cho cá ăn xuống PIC-16f877A để thực thi lệnh.

+ Khối PIC-16f877A: Chức năng đọc dữ liệu từ ds1307 để đọc thời gian thực, đọc dữ liệu từ ds18b20 về nhiệt độ hồ cá, đọc và gửi dữ liệu với esp-8266 để cài đặt thời gian cho cá ăn và nhiệt độ giới hạn, pic điều khiển servo để cho cá ăn và điều khiển relay cho 2 thiết bị làm lạnh và làm nóng để nhiệt độ hồ cá theo nhiệt độ cài đặt giới hạn nhiệt độ.

+ Khối 2 relay: Chức năng bật tắt 2 thiết bị làm nóng, làm lạnh nước hồ cá theo nhiệt độ cài đặt đã thiết lập.

+ Khối servo: Chức năng điều khiển máng cho cá ăn theo thời gian đã thiết lập cho cá ăn là thời gian nào.

+ Khối lcd: Chức năng hiển thị, cài đặt những thông số như thời gian cho cá ăn, nhiệt độ giới hạn nước hồ cá.

+ Khối nguồn 1: Chức năng cung cấp nguồn cho các linh kiện chính như cảm biến ds18b20, ds1307, pic16f877a, esp-8266, lcd để tránh những tính hiệu nhiễu từ các thiết bị điều khiển công suất là relay.

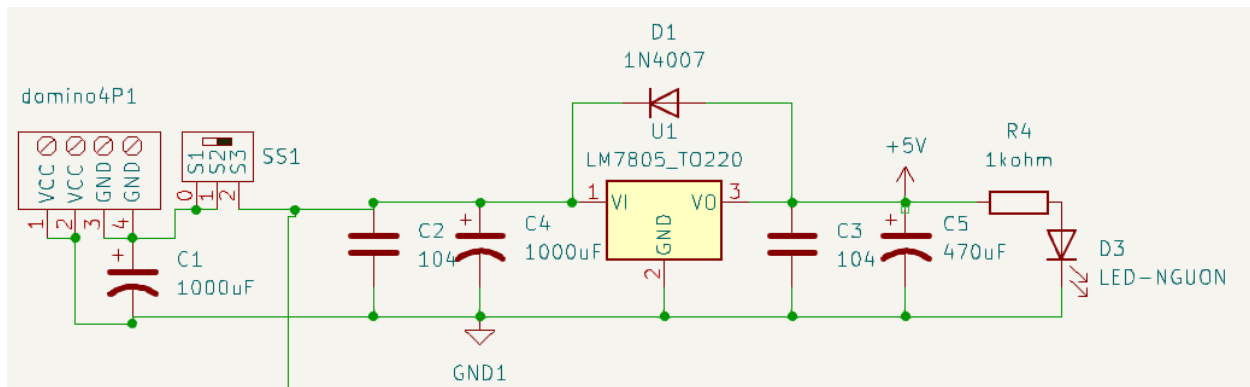
+ Khối nguồn 2: Chức năng cung cấp nguồn cho relay và servo để tránh nhiễu xung từ relay ảnh hưởng đến tín hiệu điều khiển từ bộ trung tâm pic16f877a.

3.2.2. Tính toán và thiết kế mạch

Từ sơ đồ khối hệ thống thì chúng em sẽ theo đó và thiết mạch tương ứng cho ứng dụng đó như sau:

ĐỒ ÁN 1

3.2.2.1. Khối nguồn 1:



Hình 3. 2 Khối nguồn 1

Bảng 3. 1 Dòng tiêu thụ cho khối nguồn 1

STT	TÊN	DÒNG VÀO TỐI ĐA	LOẠI NGUỒN
1	Pic16f877a	400mA	5V
2	Esp-8266	320mA	5V
3	Lcd 16x02	20mA	5V
4	Ds18b20	60uA	5V
5	Ds1307	1.5mA	5V
Dòng tổng:			742mA

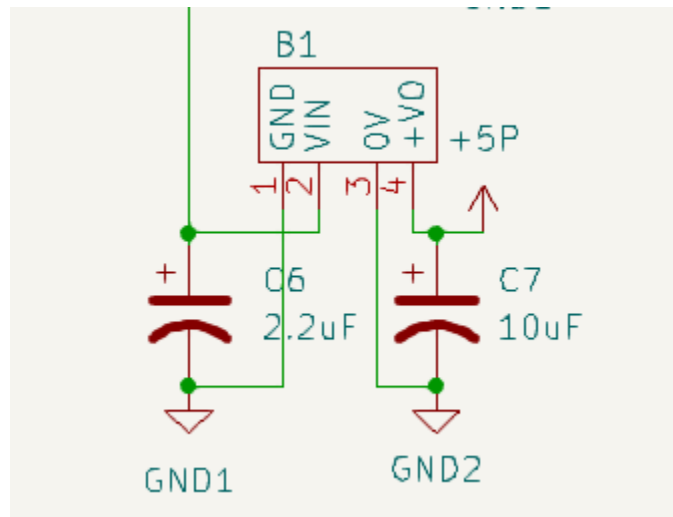
- Tính dòng pic16f877a dòng tối đa cho mỗi chân gpio là 25mA, hệ thống sử dụng 16 chân gpio nên dòng tổng cho pic16f877a là $25\text{mA} * 16 = 400\text{mA}$.

- Tính dòng esp-8266 dòng tiêu thụ tối đa khi ở chức năng wifi cường độ cao là 320mA, nên dòng tổng của esp-8266 là 320mA.

- Tính lcd1602 dòng hoạt động bình thường là 20-40mA, nên chọn dòng tổng tối đa của lcd1602 là 40mA.

- Dòng tổng của các linh kiện là 742mA nên chọn ic7805 (ic ổn áp 5V), ic 7805 có dòng tải tối đa lên tới 1.5A và dòng hoạt động ổn định là 800mA nên hệ thống hoạt động ổn định.

3.2.2.2. Khối nguồn 2



Hình 3. 3 Khối nguồn 2

Bảng 3. 2 Dòng tiêu thụ cho khối nguồn 2

STT	TÊN	DÒNG VÀO TỐI ĐA	LOẠI NGUỒN
1	2 Relay	126mA	5V
2	Servo SG92R	524mA	5V
Dòng tổng:			650mA

- Tính dòng Servo:

Ta biết tốc độ góc không tải của Servo SG92R là 0.1 giây / 60 ° hoặc 10.47 rad/s.

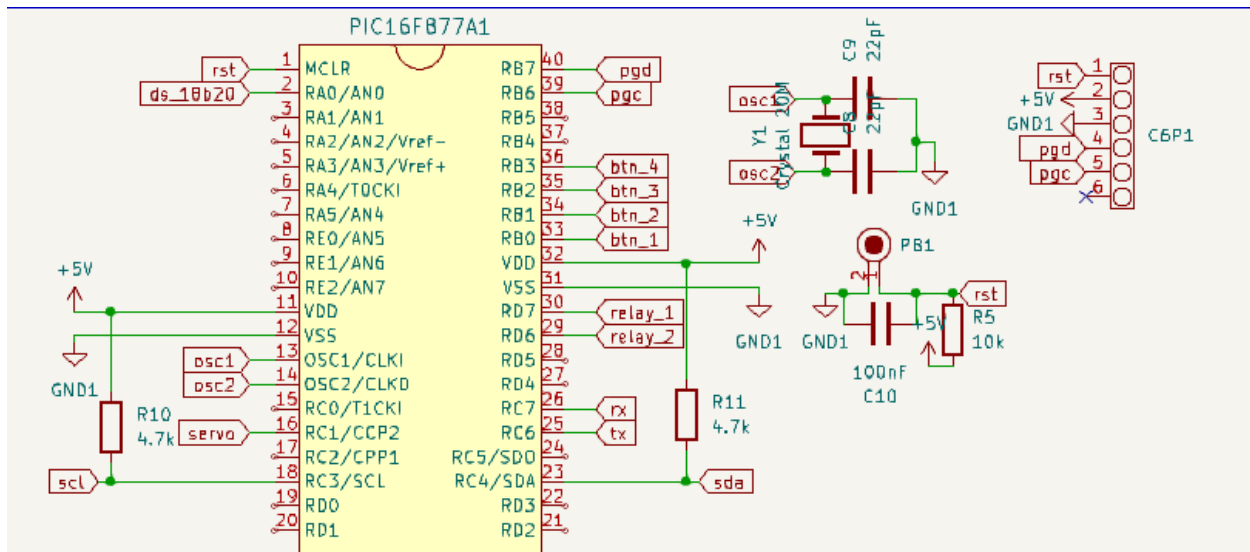
Tính cơ năng của Servo SG92R là $P = Tw = (0.25 \text{ Nm})(10.47 \text{ rad/s}) = 2.62 \text{ watts}$.

Tính tổng dòng là $I = P/V = (2.62 \text{ watt})/(5V) = 0.524A = 524 \text{ mA}$.

- Tính dòng 2 relay do thực nghiệm đo dòng hoạt động của 1 relay là khoảng 63mA nên dòng tổng 2 relay là $63(\text{mA}) * 2 = 126(\text{mA})$.

- Dòng tổng các linh kiện là 650mA và nguồn phải cách ly với nguồn chính nên sử dụng B1205 là linh kiện ổn áp 5v, dòng đáp ứng của B1205 là 2A nên module đáp ứng được các linh kiện của hệ thống.

3.2.2.3. Khởi PIC 16F877A



Hình 3. 4 Khởi PIC 16F877A

Khởi xử lí trung tâm Pic16f877a có thiết kế với thạch anh sử dụng là 20Mhz và có sử dụng chuẩn truyền i2c ở 2 chân RC3-SCL, RC4-SDA, 2 chân được kéo trở 4.7kohm lên VCC.

- Tính toán chọn trở kéo lên cho chân RC3 và RC4:

Ta có giá trị điện trở cần đảm bảo 2 yếu tố là thỏa mãn phù hợp mức logic => Rmin, thỏa mãn rise time của tín hiệu => Rmax

Công thức Rmin được tính

$$R_P(\min) = \frac{(V_{CC} - V_{OL}(\max))}{I_{OL}}$$

Công thức Rmax được tính :

$$V(t) = V_{CC} \times \left(1 - e^{-\frac{t}{RC}}\right)$$

Thường VIH và VIL bằng 0.7*VCC và 0.3*VCC nên :

For $V_{IH} = 0.7 \times V_{CC}$:

$$V_{IH} = 0.7 \times V_{CC} = V_{CC} \times \left(1 - e^{\frac{-t_1}{R_p \times C_b}} \right)$$

For $V_{IL} = 0.3 \times V_{CC}$:

$$V_{IL} = 0.3 \times V_{CC} = V_{CC} \times \left(1 - e^{\frac{-t_2}{R_p \times C_b}} \right)$$

The rise time for the I2C bus can be written as:

$$t_r = t_2 - t_1 = 0.8473 \times R_p \times C_b$$

Từ đó tính Rmax là $R_p(\max)$ là:

$$R_p(\max) = \frac{t_r}{(0.8473 \times C_b)}$$

Table 1. Parametrics from I2C specifications

	Parameter	Standard Mode (Max)	Fast Mode (Max)	Fast Mode Plus (Max)	Unit
t_r	Rise time of both SDA and SCL signals	1000	300	120	ns
C_b	Capacitive load for each bus line	400	400	550	pF
V_{OL}	Low-level output voltage (at 3 mA current sink, $V_{CC} > 2$ V)	0.4	0.4	0.4	V
	Low-level output voltage (at 2 mA current sink, $V_{CC} \leq 2$ V)	–	$0.2 \times V_{CC}$	$0.2 \times V_{CC}$	V

Hình 3. 5 Bảng đặc tính i2c

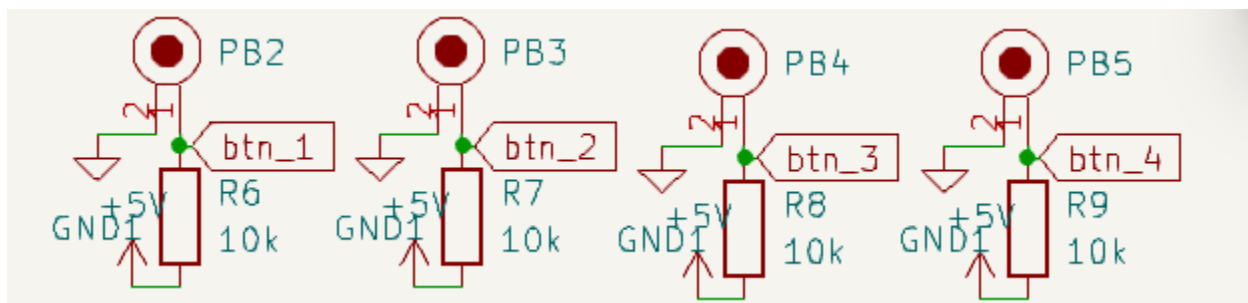
Ta chọn chế độ standard thì có rise time là 1000ns, C_b là 400pF, V_{CC} là 5v.

Suy ra $R_p(\max) = (1000 \times 10^{-9}) / (0.8473 \times 400 \times 10^{-12}) = 5.9 \text{ kohm}$.

Tính $R_p(\min) = (5 - 0.4) / (3 \times 10^{-3}) = 1.5 \text{ kohm}$.

Suy ra chọn giá trị điện trở pull up là 4.7kohm là phù hợp với tốc độ truyền standard (100kbit) cũng như nhu cầu của hệ thống.

3.2.2.4. Khối 4 nút nhấn

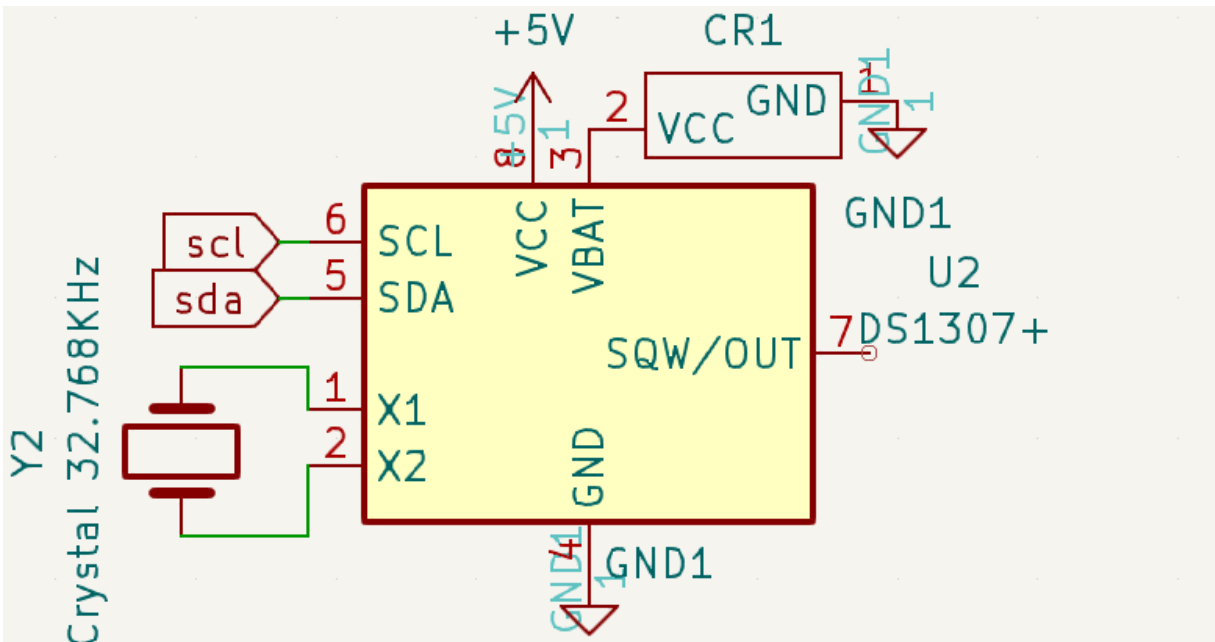


Hình 3. 6 Khối 4 nút nhấn

ĐỒ ÁN 1

Tất cả các nút nhấn đều có trở 10kohm kéo lên nguồn VCC để tạo tín hiệu mức 1 và mức 0 cho pic16f877a (dạng tích cực mức thấp).

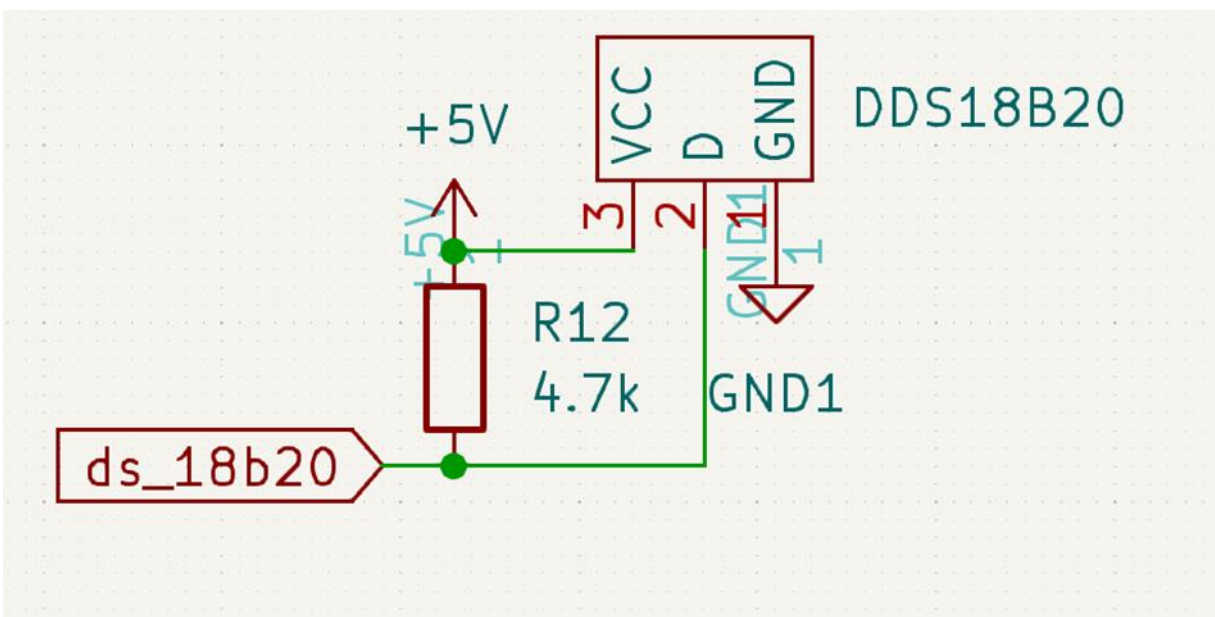
3.2.2.5. Khối ds1307



Hình 3. 7 Khối ds1307

- Khối ds1307 sử dụng thạch anh 32.768Khz dựa vào datasheet của nhà cung cấp.

3.2.2.6. Khối ds18b20

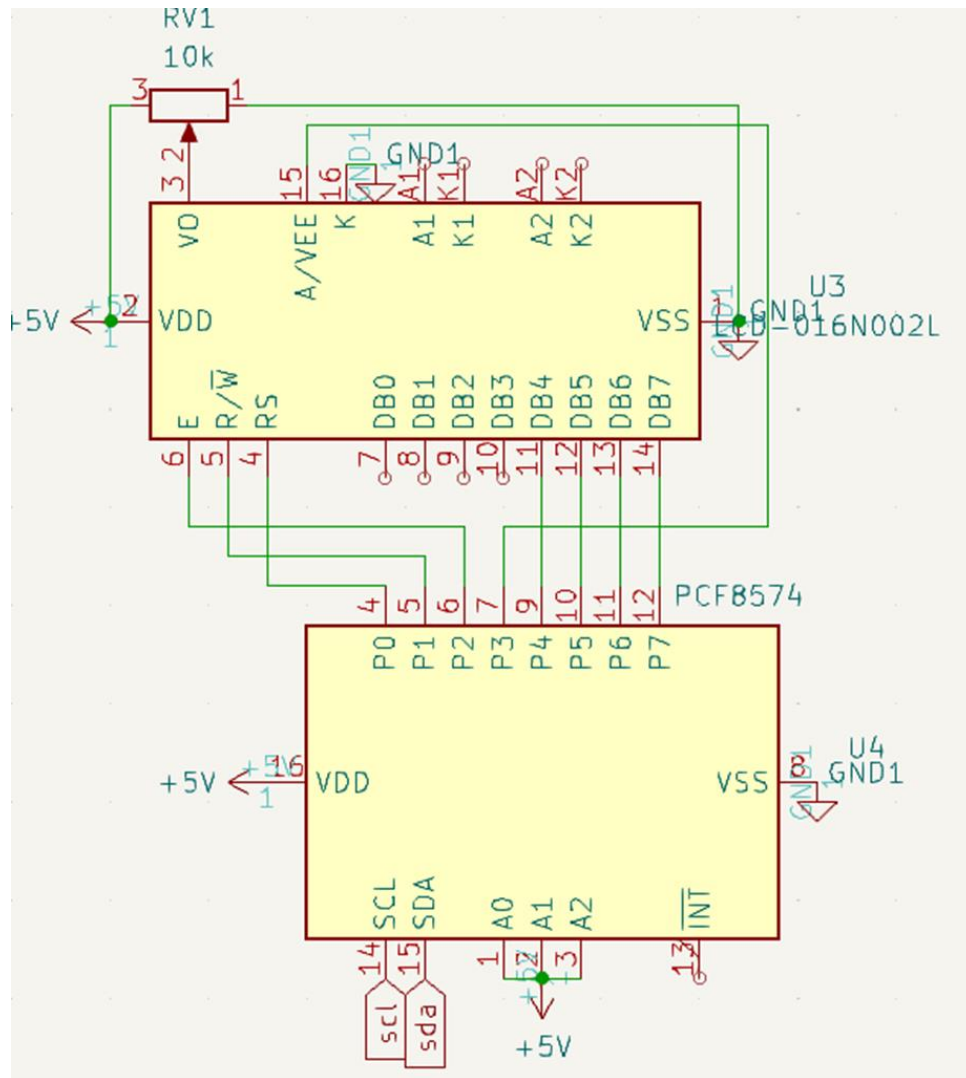


Hình 3. 8 Khối ds18b20

ĐỒ ÁN 1

- Khối ds18b20 có sử dụng trở 4.7kohm dựa vào datasheet của nhà cung cấp.

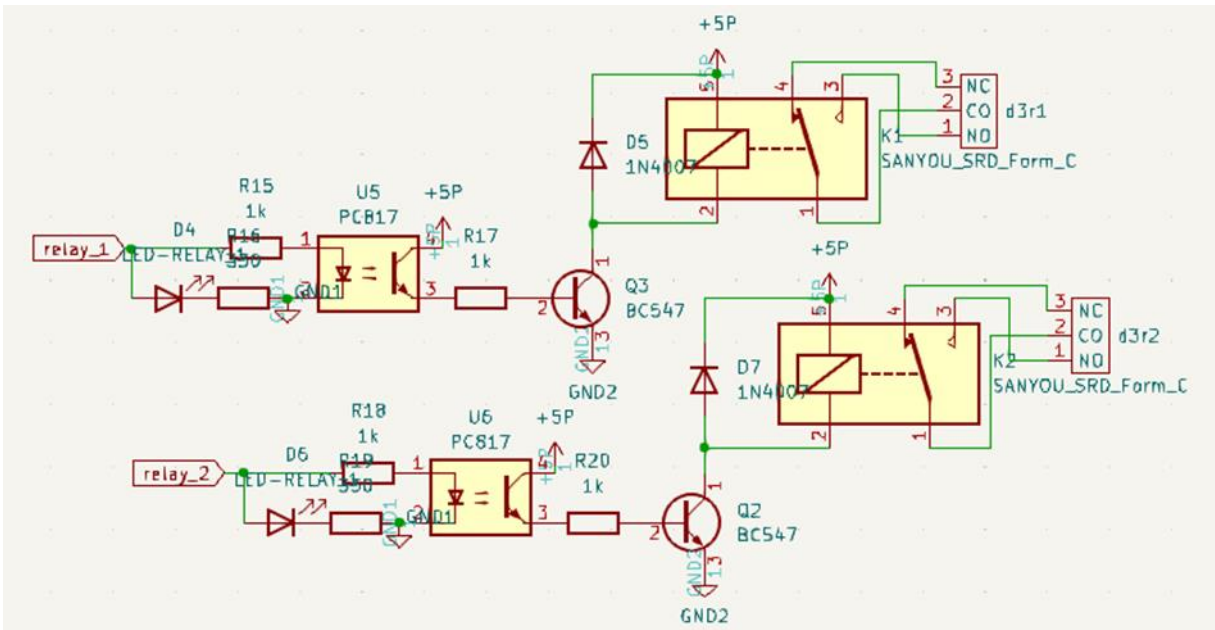
3.2.2.7. Khối lcd 1602



Hình 3. 9 Khối lcd 1602

- khối lcd 16x02 thiết kế dạng sử dụng i2c để giao tiếp hiển thị nên lcd sử dụng chế độ truyền 4 dây, chân vd có biến trở 10kohm để chỉnh độ sáng.

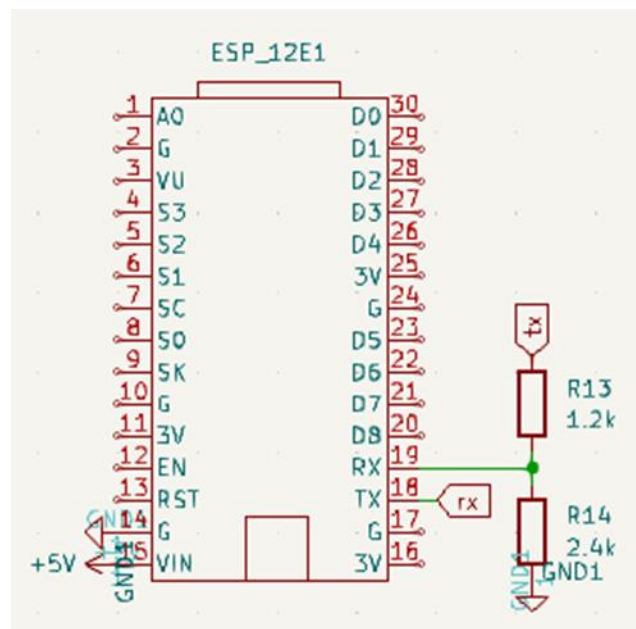
3.2.2.8. Khối 2 relay



Hình 3. 10 Khối 2 relay

- Khối 2 relay: chọn relay 5vdc vì hệ thống sử dụng nguồn hệ thống là 5vdc, tải của relay là 10A/250vac, 10A/30vdc vì hệ thống chỉ quản lí tải nhẹ như thiết bị gia nhiệt 12v/15w, thiết bị làm lạnh 12v/10w.

3.2.2.9. Khối esp-8266



Hình 3. 11 Khối esp-8266

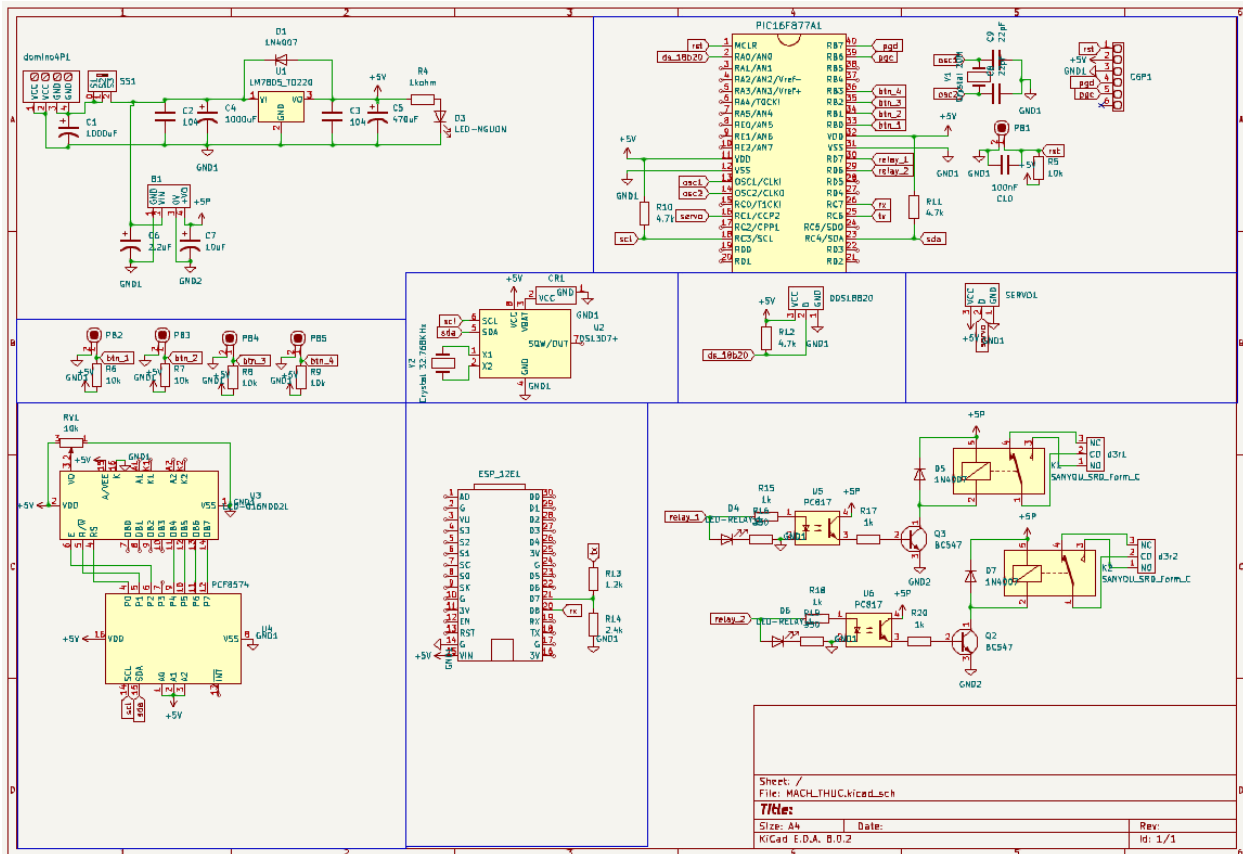
ĐỒ ÁN 1

- Khối esp-8266: chân rx nối với cầu phân áp với giá trị điện trở tự chọn là 1.2kohm và 2.4kohm thì điện áp ngõ ra của pic16f877a là 5vdc. Ta có công thức tính như sau:

$$V_o = 5(vdc) * (2.4(kohm)) / (2.4(kohm) + 1.2(kohm)) = 3.33(vdc)$$

Do điện áp gpio của esp-8266 là 3.3vdc nên chọn các giá trị điện trở đã chọn.

3.2.3. Sơ đồ nguyên lý toàn mạch



Hình 3. 12 Sơ đồ nguyên lý toàn mạch

Đây là hình 3.12 là sơ đồ kết nối toàn mạch từ các linh kiện riêng lẻ kết nối thành 1 hệ thống hoàn chỉnh.

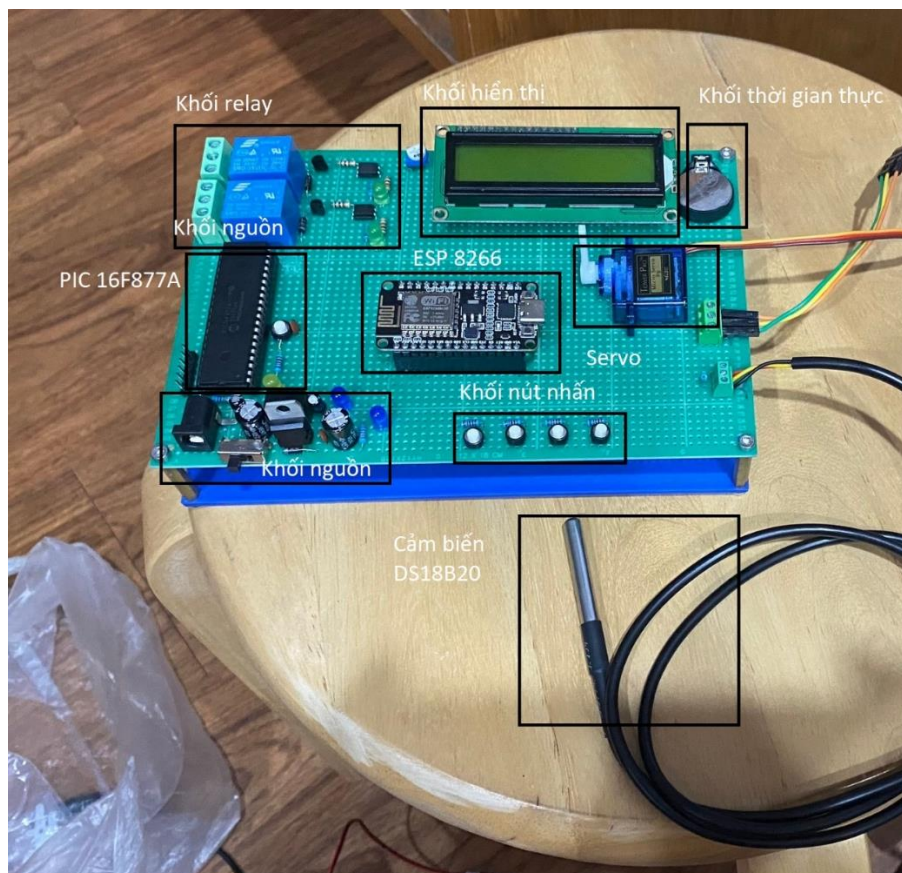
CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.1. GIỚI THIỆU

Chương 4 sẽ gồm thi công phần cứng và xây dựng phần mềm cho hệ thống kiểm soát hồ cá.

4.2. THI CÔNG HỆ THỐNG

4.2.1. Thi công bo mạch



Hình 4. 1 Hệ thống

Hệ thống gồm các khối sau:

- Khối nguồn: Cung cấp nguồn hoạt động cho hệ thống
- Khối PIC 16F877A: Khối xử lý trung tâm
- Khối hiển thị: hiển thị các thông tin của hệ thống , nhiệt độ đo được, ngày tháng năm, giờ phút giây
- Khối nút nhấn: thao tác phần cứng

ĐỒ ÁN 1

- Khối thời gian thực: cung cấp thời gian chính xác và liên tục, ngay cả khi hệ thống chính bị tắt
- Khối ES_p 8266: sẽ giúp cho hệ thống giao tiếp và thực hiện các thao tác trên phần mềm điện thoại
- Khối servo: gạt máng ăn cho bể cá
- Khối cảm biến DS18B20: đọc nhiệt độ và gửi về cho PIC
- Khối relay: bật tắt các thiết bị nóng lạnh để làm cân bằng nhiệt độ của nước

4.2.2. lắp ráp và kiểm tra

Lắp ráp theo qui trình sau đây:

- Hàn jack cắm nguồn, hàn khối nguồn chuyển đổi 12V sang 5V, có gắn diot bảo vệ đầu vào của khối nguồn 12V sang 5V. Kiểm tra nguồn ở đầu ra là 5V hay không, nếu đúng hàn tiếp. Ở ngõ ra 5V nối với trở 470 ôm, có công tắc gạt để bật tắt hệ thống và led báo có nguồn.

- Hàn đế chân của PIC 16F877A, sau đó hàn các chân dung để nạp chương trình ra 1 hàng rào chân để tiện cho quá trình kiểm tra và sửa chữa chương trình cho PIC. Đồng thời là hàn nút nhất reset có gắn trở 470 ôm kéo lên và được kết nối đến chân 1 của PIC16F877A.

- Hàn 4 nút nhấn để thao tác các chức năng của hệ thống. Mỗi nút nhấn đều có điện trở 10K kéo lên, và kết nối đến các chân 33 (nút nhấn 1), chân 34 (nút nhấn 2) , chân 35 (nút nhấn 3) và chân 36 (nút nhấn 4) của PIC 16F887A.

- Hàn chân cắm cho esp sau đó cho kết nối chân 20 (D8) của ESP826 với chân 26 (RX), kết nối chân 21(D7) đến chân 25 (TX) và tín hiệu này được kéo xuống bởi 2 điện trở 1.2K và 2.4K.

- Chân data của servo kết nối với chân 16 của PIC.
- Chân data của ds18b20 kết nối với chân 2 của PIC.
- Chân 30 và chân 29 của PIC sẽ lần lượt là 2 chân tín hiệu điều khiển 2 relay 1 và 2.

ĐỒ ÁN 1

- Cho LCD 16x2 giao tiếp với IC PCF8574 để rút gọn chân kết nối với PIC. Lúc này ta chỉ cần kết nối chân SCL và SDA của PCF8574 lần lượt đến chân 18 và 13 của PIC. Cả 2 tín hiệu đều được kéo lên bởi điện trở 4.7K.

- Ds1307 sẽ hoạt động với thạch anh tần số 32,768kHz và cũng được kết nối đến chân 23 (SDA) và chân 18 (SCL) của PIC. Ngoài ra ds1307 còn được cung cấp nguồn bên ngoài bởi pin CMOS để ds1307 vẫn luôn hoạt động dù hệ thống chưa được cấp điện.

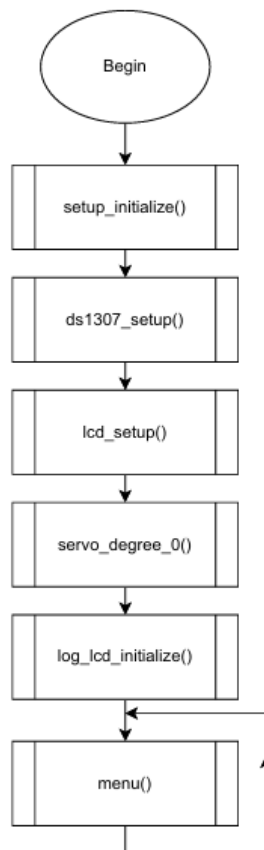
- Ta dùng thạch anh với tần số là 20MHz để cung cấp cho PIC ở 2 chân OSC1 và OSC2.

- Sau đó tiến hành kiểm tra lại nguồn cung cấp 1 lần nữa, nạp chương trình và chạy thử hệ thống.

4.3. LẬP TRÌNH HỆ THỐNG

4.3.1. Lưu đồ giải thuật

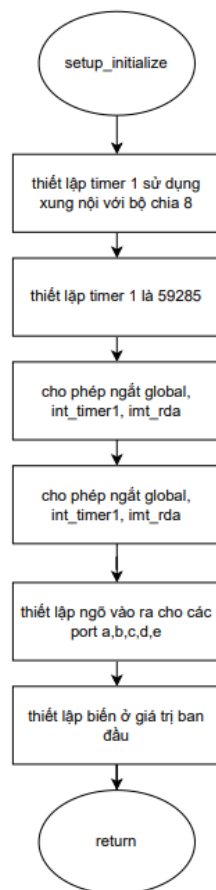
4.3.1.1. Lưu đồ giải thuật cho pic16f877a



Hình 4. 2 Lưu đồ chương trình chính cho pic16f877a

ĐỒ ÁN 1

Theo hình 4.1 ta thấy, ban đầu chương trình cài đặt các trạng thái ban đầu như cấu hình ngõ vào ngõ ra cho các chân tín hiệu, thiết lập ngắt cho hệ thống là sử dụng ngắt nhận dữ liệu uart, ngắt timer 1, thiết lập các trạng thái ban đầu của các biến sử dụng. Tiếp theo là cài đặt ds1307 kiểm tra đã cài đặt thời gian cho cảm biến chưa, nếu chưa cài đặt thì cài đặt thời gian mặc định cho ds130, nếu đã cài đặt thì bỏ qua cài đặt thời gian mặc định cho ds1307. Khi cài đặt ds1307 xong thì thiết lập hiển thị lcd kiểu 4 chân dữ liệu và các lệnh cần thiết cho lcd. Sau đó là điều khiển servo về vị trí ban đầu là 0 độ. Khi thực hiện xong thì hiển thị ra Lcd các thông tin cơ bản về tên đề tài và năm thực hiện đề tài. Sau khi cài đặt hết tất cả điều cần thiết cho hệ thống thì vào vòng lặp vô tận while true với hàm con là menu để thực hiện các việc về đọc nhiệt độ, thời gian thực, cài đặt nhiệt độ giới hạn, nhiệt độ giới hạn, giao tiếp với esp để gửi và nhận dữ liệu điều khiển cho hệ thống.



Hình 4. 3 Lưu đồ setup_initialize

ĐỒ ÁN 1

Ban đầu thiết lập bộ đếm timer 1 dùng xung nội và bộ chia trước là 8 để tạo bộ đếm 10ms theo công thức tính như sau:

Tần số sử dụng cho hệ thống là 20MHz.

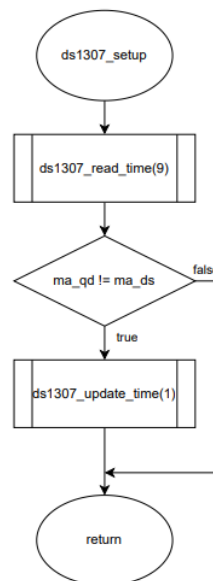
=> $F(t1) = (20\text{MHz}/4)/8 = 625000 \text{ Hz} \Rightarrow 1\text{s}$ thì cần phải có 625000 xung.

=> 10ms thì cần $(10\text{ms} * 625000)/1\text{s} = 6250$ xung.

Ta có bộ timer 1 là 16bit = $2^{16}-1 = 65535$

=> Điểm thiết lập timer 1 giá trị ban đầu là $65535 - 6250 = 59285$ xung.

Sau khi thiết lập timer1 xong thì thiết lập cho phép ngắt global thì cho phép pic16f877a được phép sử dụng ngắt. Bật ngắt int_timer1 và int_rda để thực hiện các công việc trong bộ đếm 10ms và nhận dữ liệu khi được esp8266 gửi dữ liệu. Tiếp theo thiết lập ngõ vào ra cho các chân tín hiệu điều khiển và giao tiếp như i2c, uart cũng như là chân điều khiển servo, relay, chân tín hiệu là nút nhấn, ds18b20, ds1307. Cuối cùng là đặt trạng thái ban đầu cho các biến mô trường xong trở về chương trình chính.

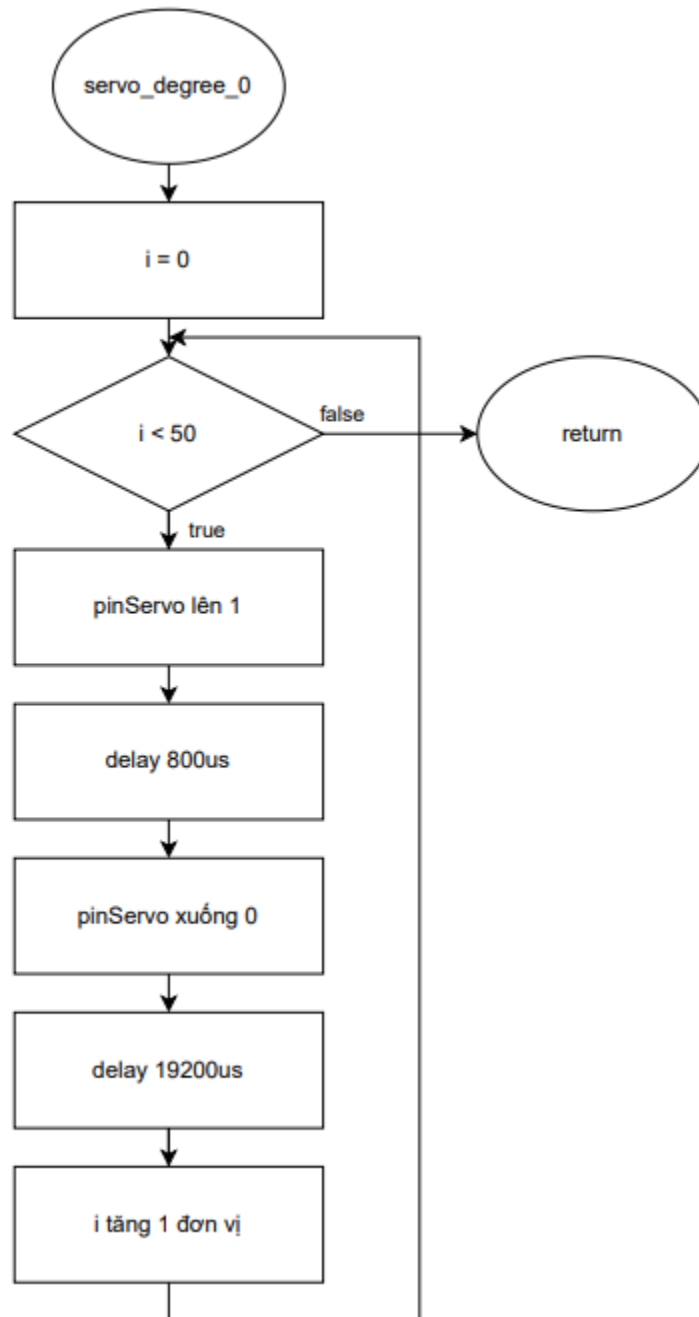


Hình 4. 4 Lưu đồ ds1307_setup

Bước đầu tiên là đọc tất cả dữ liệu từ ds1307 về cho pic16f877a để kiểm tra mã qd. Mã qd là mã do người dung định nghĩa để biết ds1307 đã được thiết lập thời gian ban đầu cho ds1307 chưa. Nếu ma_qd khác ma_ds thì được hiểu là chưa thiết lập thời gian ban đầu từ khi mua linh kiện về (true) thì ds1307 sẽ update thời gian ban đầu cho ds1307, nếu ma_qd

ĐỒ ÁN 1

bằng ma_ds thì được hiểu là ds1307 đã được thiết lập thời gian ban đầu rồi nên không cần thiết lập thời gian ban đầu cho ds1307 (false) trở về chương trình chính.

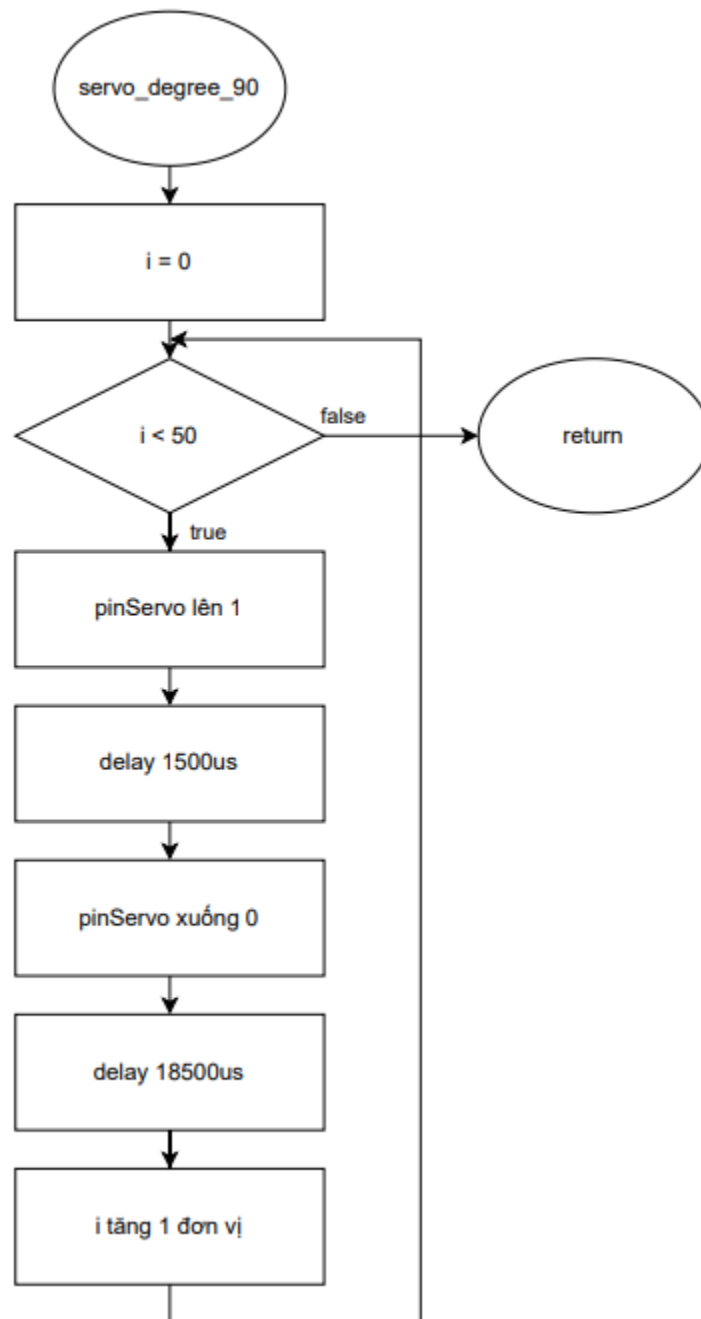


Hình 4. 5 Lưu đồ servo_degree_0

Tạo hàm for chạy từ 0 đến 49 để được tần số 50hz. Để servo ở góc 0 độ thì ta cần gửi 50 xung liên tiếp là với giá trị mức 1 là 800us và khoảng mức 0 là thì servo sẽ quay ở góc 0 độ. Ban đầu cho `i = 0` sau đó kiểm tra `I < 50` nếu đúng thì cho `pinServo` lên 1, chờ trong

ĐỒ ÁN 1

800us thì cho pinServo xuống 0 và chờ trung 19200us rồi tăng i lên 1, tiếp tục kiểm tra $i < 50$ thì thực hiện tiếp tục, nếu sai thì trở về chương trình chính.



Hình 4. 6 Lưu đồ servo_degree_90

Tạo hàm for chạy từ 0 đến 49 để được tần số 50hz. Để servo ở góc 0 độ thì ta cần gửi 50 xung liên tiếp là với giá trị mức 1 là 1500us và khoảng mức 0 là 18500us thì servo sẽ quay ở góc 0 độ. Ban đầu cho $i = 0$ sau đó kiểm tra $i < 50$ nếu đúng thì cho pinServo lên 1,

ĐỒ ÁN 1

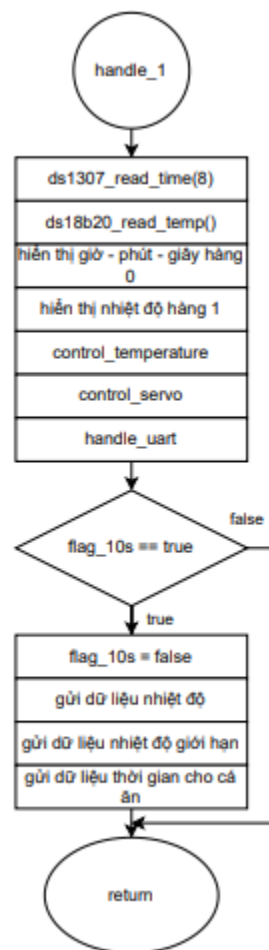
chờ trong 1500us thì cho pinServo xuống 0 và chờ trung 18500us rồi tăng i lên 1, tiếp tục kiểm tra $i < 50$ thì thực hiện tiếp tục, nếu sai thì trở về chương trình chính.



Hình 4. 7 Lưu đồ menu

ĐỒ ÁN 1

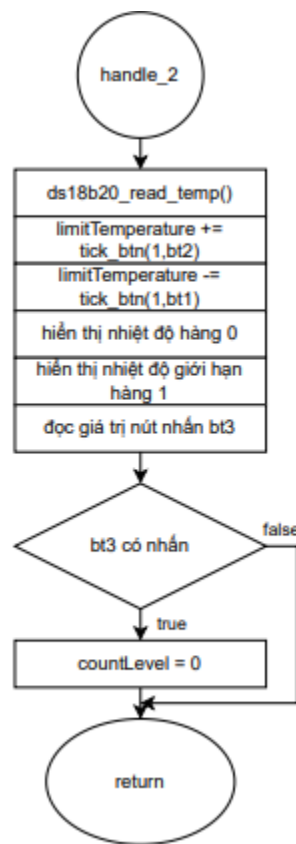
Ban đầu kiểm tra nút nhấn bt0, nếu có nhấn (true) thì cho countLevel tăng 1 sai thì không tăng countLevel khi xong thì kiểm tra tiếp countLevel \geq maxLevel nếu đúng thì đặt lại countLevel thành 0 còn sai thì bỏ qua thiết lập lại countLevel tiếp tục kiểm tra countLevel khác countLevelRev nếu đúng thì countLevel thay đổi thì đặt lại countlevelRev bằng countLevel là xoá dữ liệu hiển thị trên lcd còn sai thì bỏ qua đặt lại countLevelRev với xoá dữ liệu lcd. Tiếp theo kiểm tra countLevel bằng 0 đúng thì thực hiện lệnh trong hàm con handel_1 sai thì kiểm tra tiếp bằng 1 đúng thì thực hiện lệnh trong hàm con handel_2 sai thì kiểm tra tiếp bằng 2 đúng thì thực hiện lệnh trong hàm con handel_3 sai thì kiểm tra tiếp tới bằng 3 đúng thì thực hiện lệnh trong hàm con handle_4 sai thì kiểm tra tới bằng 4 đúng thì thực hiện lệnh trong hàm con handel_5 sai thì thực hiện lệnh trong handel_6. Sau đó trở về chương trình chính.



Hình 4. 8 Lưu đồ handel_1

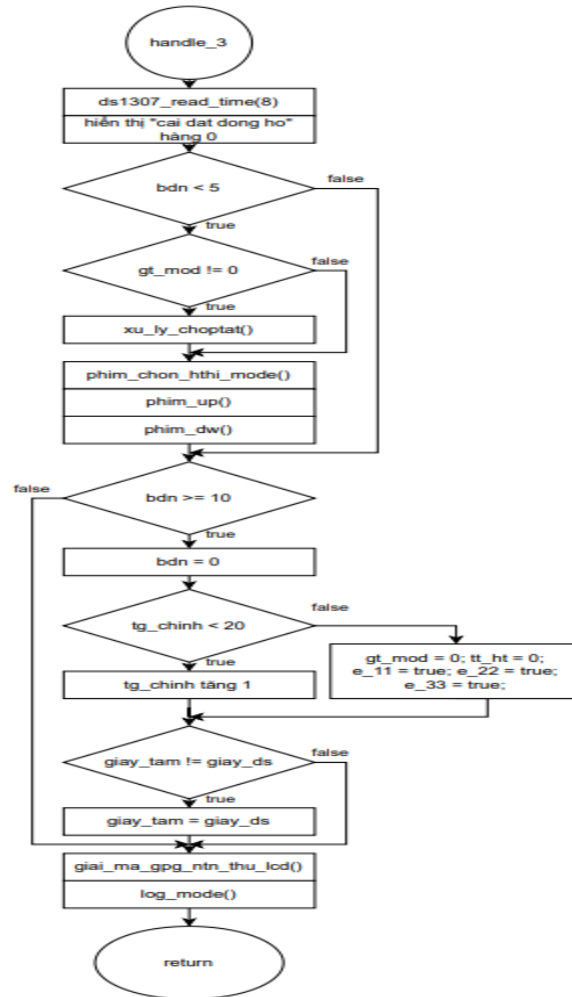
ĐỒ ÁN 1

Ban đầu đọc dữ liệu thời gian từ ds1307, sau đó đọc dữ liệu nhiệt độ từ ds18b20. Tiếp theo hiển thị giờ - phút – giây đọc được từ ds1307 ở hàng 0 và dữ liệu nhiệt độ đọc được ở hàng 1. Sau đó điều khiển relay dựa theo nhiệt độ giới hạn và nhiệt độ đọc được từ ds18b20. Sau đó điều khiển servo khi thời gian đọc được từ ds1307 bằng với thời gian cài đặt cho cá ăn. Kiểm tra flag_10s (cờ 10s được thiết lập trong timer1), nếu cờ 10s đúng thì đặt cờ 10s lại thành false, gửi dữ liệu về nhiệt độ, nhiệt độ giới hạn, thời gian cho cá ăn cho esp. Sau đó trở về chương trình chính.



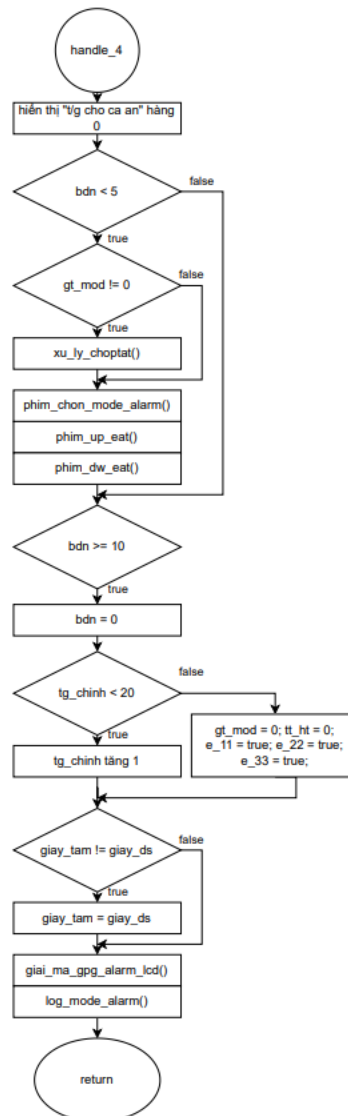
Hình 4. 9 Lưu đồ handle_2

Đầu tiên là đọc dữ liệu nhiệt độ từ cảm biến ds18b20. Đọc giá trị nút nhấn bt1, khi nhấn đề thì giá trị limitTemperature giảm. Đọc giá trị nút nhấn bt2, khi nhấn đề thì giá trị limitTemperature tăng. Hiển thị nhiệt độ đọc được từ cảm biến ds18b20 ở hàng 0, hiển thị nhiệt độ giới hạn ở hàng 1. Đọc giá trị nút nhấn bt3, kiểm tra khi bt3 có nhấn đúng thì thiết lập lại countLevel = 0 sai thì bỏ qua thiết lập countLevel = 0 và trở về chương trình chính.



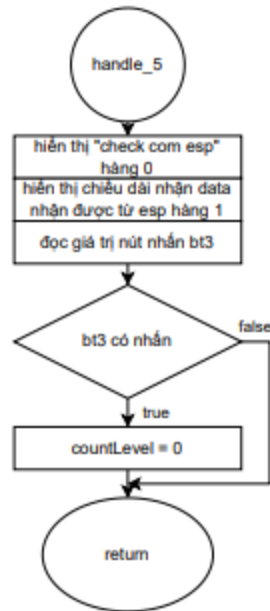
Hình 4. 10 Lưu đồ handle_3

Đầu tiên đọc dữ liệu thời gian từ cảm biến ds1307, hiển thị “cai dat dong ho” ở hàng 0. Sau đó kiểm tra $bdn < 5$ đúng thì kiểm tra tiếp $gt_mod \neq 0$ đúng thì xử lý chớp tắt, sai thì bỏ qua xử lý chớp tắt và thực hiện chọn mode từ $phim_chin_hthi_mode()$, $phim_up()$, $phim_dw()$, nếu $bdn < 5$ sai thì bỏ qua bước kiểm tra gt_mod và tiếp tục kiểm tra $bdn \geq 10$, đúng thì cho $bdn = 0$, kiểm tra $tg_chinh < 20$ đúng thì tg_chinh tăng 1 sai thì $gt_mod = 0$, $tt_ht = 0$, $e_11 = true$, $e_22 = true$, $e_33 = true$, sau đó kiểm tra $giay_tam \neq giay_ds$ đúng thì $giay_tam = giay_ds$ sai thì bỏ qua set $giay_tam = giay_ds$. Nếu $bdn \geq 10$ sai thì bỏ qua các lệnh $bdn = 0$ và tiếp tục thực hiện lệnh $giai_ma_gpg_ntn_thu_lcd()$ và $log_mode()$ để hiển thị trên lcd ở hàng 1. Sau đó trở về chương trình chính.



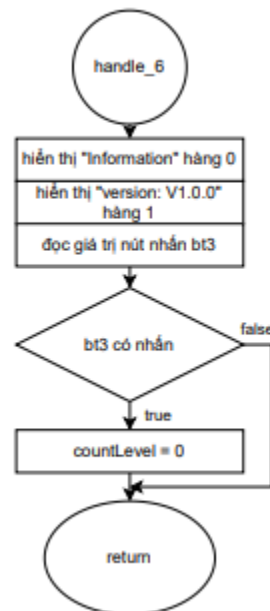
Hình 4. 11 Lưu đồ handle_4

Đầu tiên hiển thị “t/g cho ca an” ở hàng 0. Sau đó kiểm tra $bdn < 5$ đúng thì kiểm tra tiếp gt_mod khác 0 đúng thì xử lí chop tắt, sai thì bỏ qua xử lí chop tắt và thực hiện chọn mode từ `phim_chon_mode_alarm()`, `phim_up_eat()`, `phim_dw_eat()`, nếu $bdn < 5$ sai thì bỏ qua bước kiểm tra gt_mod và tiếp tục kiểm tra $bdn \geq 10$, đúng thì cho $bdn = 0$, kiểm tra $tg_chinh < 20$ đúng thì tg_chinh tăng 1 sai thì $gt_mod = 0$, $tt_ht = 0$, $e_11 = true$, $e_22 = true$, $e_33 = true$, sau đó kiểm tra $giay_tam \neq giay_ds$ đúng thì $giay_tam = giay_ds$ sai thì bỏ qua set $giay_tam = giay_ds$. Nếu $bdn \geq 10$ sai thì bỏ qua các lệnh $bdn = 0$ và tiếp tục thực hiện lệnh `giai_ma_gpg_alarm_lcd()` và `log_mode_alarm()` để hiển thị trên lcd ở hàng 1. Sau đó trở về chương trình chính.



Hình 4. 12 Lưu đồ handle_5

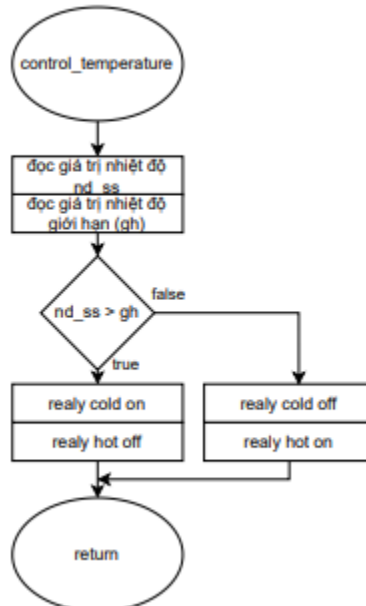
Hiển thị “check com esp” ở hàng 0. Hiển thị chiều dài chuỗi dữ liệu nhận được từ esp ở hàng 1. Đọc giá trị nút nhấn bt3 kiểm tra nếu có nhấn bt3 thì countLevel = 0 không nhấn thì bỏ qua countLevel = 0 và cả hai đều trở về chương trình chính.



Hình 4. 13 Lưu đồ handle_6

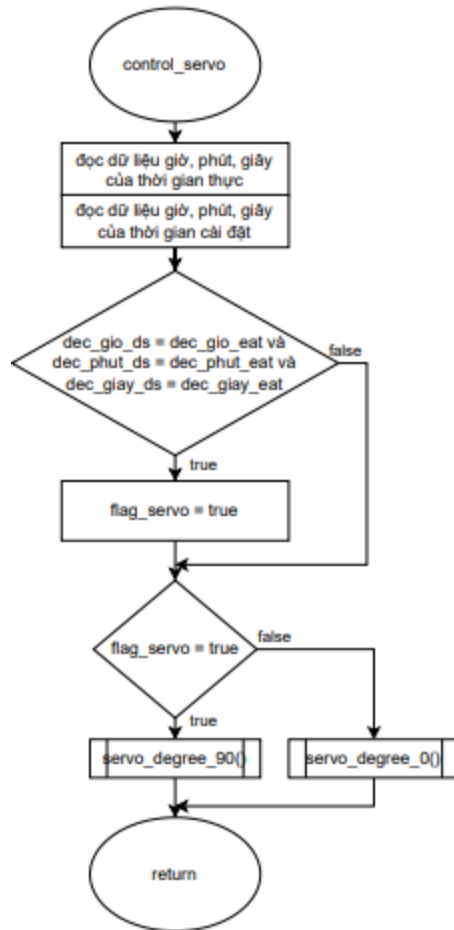
ĐỒ ÁN 1

Hiển thị “information” ở hàng 0. Hiển thị “version: V1.0.0” ở hàng 1. Đọc giá trị nút nhấn bt3 kiểm tra nếu có nhấn bt3 thì countLevel = 0 không nhấn thì bỏ qua countLevel = 0 và cả hai đều trở về chương trình chính.



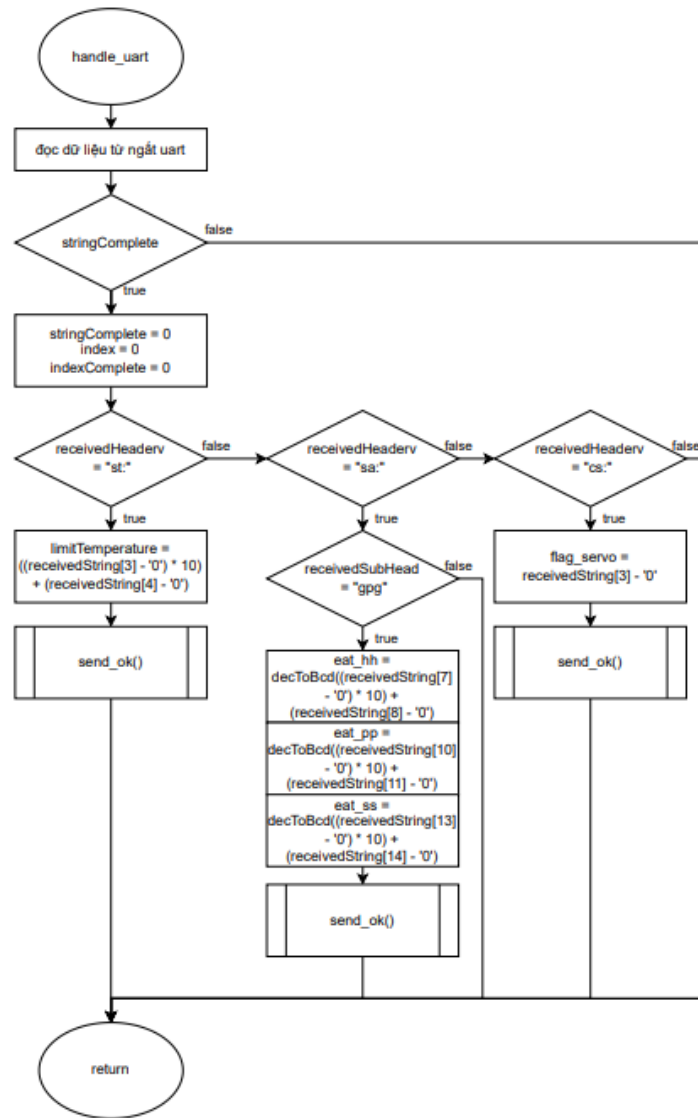
Hình 4. 14 Lưu đồ `control_temperature`

Ban đầu đọc giá trị nhiệt độ (`nd_ss`) và giá trị giới hạn (`gh`) rồi vào hàm if để so sánh nếu `nd_ss > gh` đúng là hồ nước đang có nhiệt độ lớn hơn nhiệt độ giới hạn cài đặt, bật relay cho dàn lạnh hoạt động và tắt hoạt động dàn nóng. Ngược lại thì cho dàn nóng hoạt động và tắt dàn lạnh bằng relay. Sau đó trở về chương trình chính.



Hình 4. 15 Lưu đồ control_servo

Ban đầu đọc dữ liệu về giờ, phút, giây của ds1307 và dữ liệu giờ, phút, giây cài đặt rồi so sánh nếu giờ, phút, giây của ds1307 bằng với giờ, phút, giây cài đặt thì flag_servo = true. Nếu flag_servo bằng true thì servo_degree_90 là cho servo ở vị trí 90 độ ngược lại là false thì servo ở vị trí 0 độ. Sau đó trở về chương trình chính.

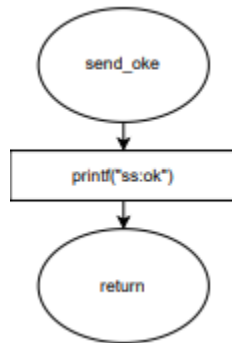


Hình 4. 16 Lưu đồ handle_uart

Ban đầu đọc dữ liệu từ uart. Kiểm tra stringComplete, nếu đúng thì gán stringComplete = 0, index = 0, indexComplete = 0, sau đó kiểm tra header nếu là “st:” là set temperature, “sa:” là set alarm, “cs:” là control servo. Khi là set temperature thì chuyển đổi dữ liệu nhận được từ dạng ký tự thành số nguyên bằng cách lấy ký tự đó trừ cho ký tự ‘0’ thì ta được số nguyên mong muốn nhân 10 là được vị trí hàng chục và sau đó gửi send_ok lại cho esp để xác nhận đã nhận dữ liệu thành công. Khi là set alarm thì kiểm tra subheader là “gpg” là chỉnh giờ phút giây đúng thì ta chuyển đổi từ ký tự thành số nguyên giờ phút giây sai thì

ĐỒ ÁN 1

trở về chương trình chính. Khi là control servo thì cho flag_servo lên 1 và gửi send_ok sai thì trở về chương trình chính.



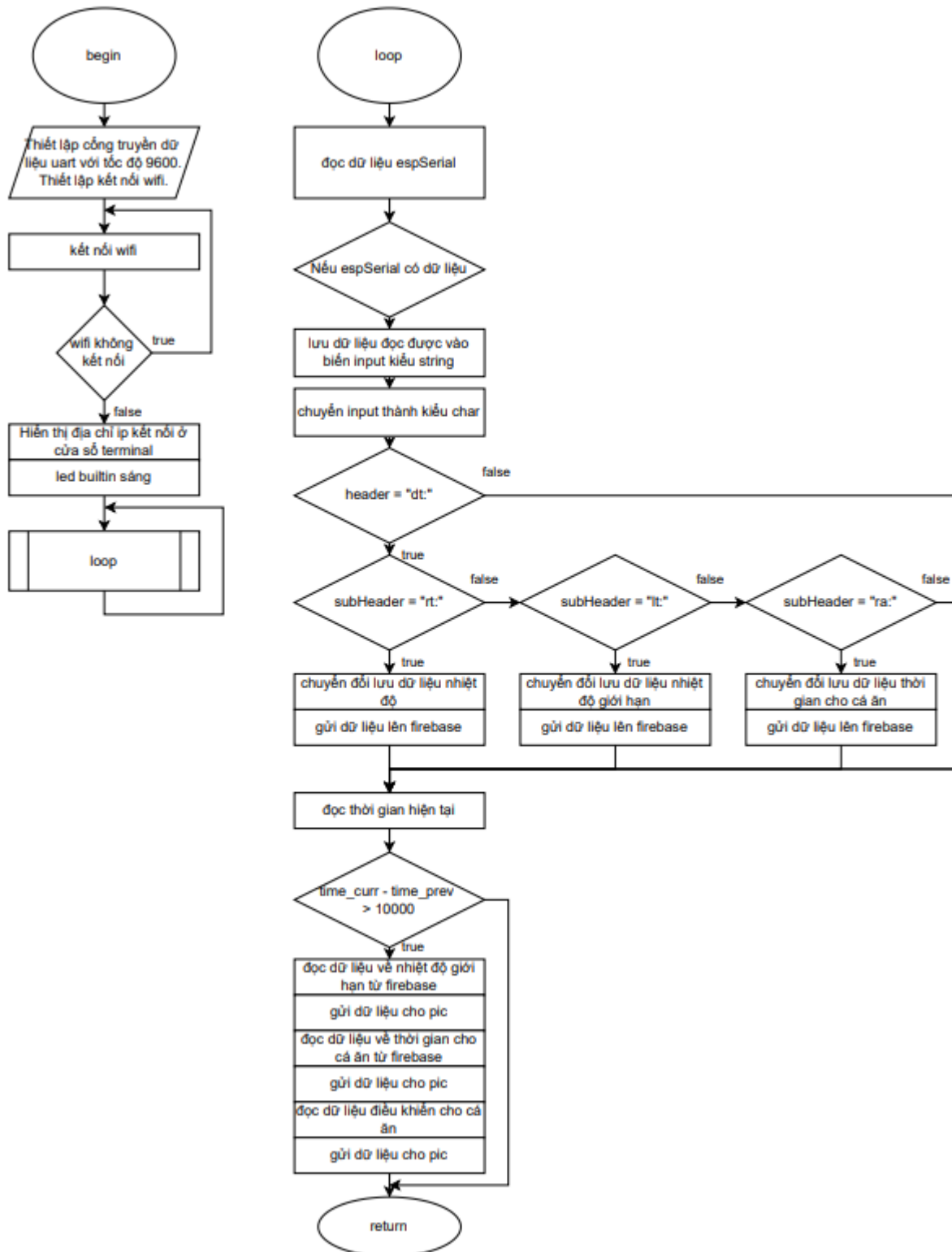
Hình 4. 17 Lưu đồ send_oke

Khi send_ok được gọi thì gửi 1 lệnh printf("ss:ok") cho esp để xác nhận đã nhận dữ liệu và thực hiện thành công.

Chương trình code pic:

https://github.com/VM-Thuan-2003/do_an_1/tree/main/PROJECT/CODE_FINNAL

4.3.1.2. Lưu đồ giải thuật cho esp8266



Hình 4. 18 Lưu đồ chương trình cho esp

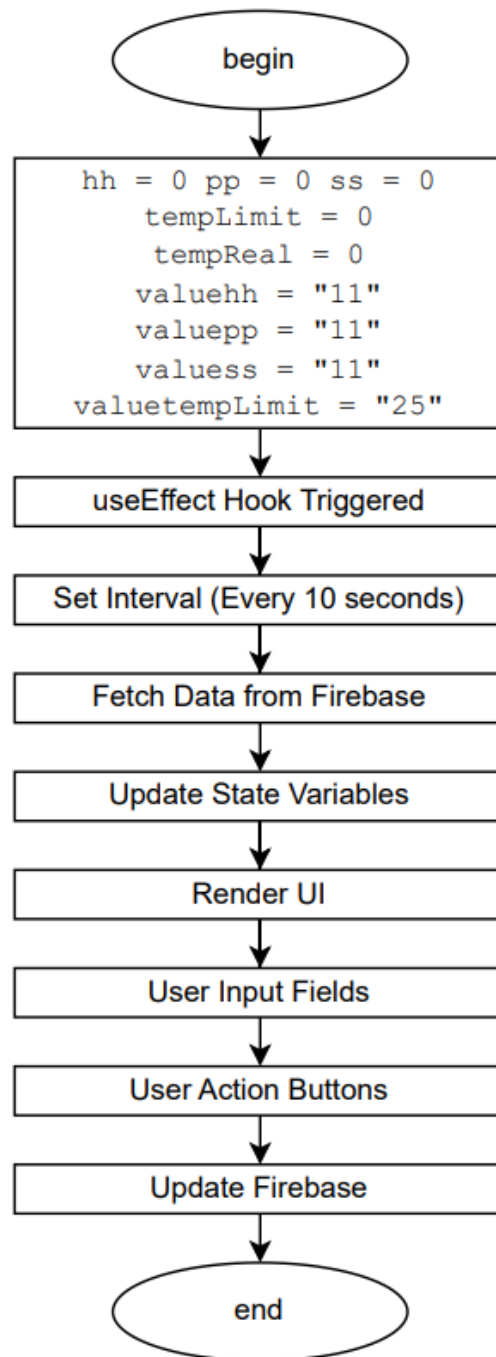
ĐỒ ÁN 1

Ban đầu esp thiết lập tốc độ cho cổng uart là cổng Serial của esp và espSerial là cổng giao tiếp với pic để đọc và truyền dữ liệu hiển thị, điều khiển cho pic. Sau khi thiết lập tốc độ xong thì thiết lập chế độ kết nối ở chế độ STATION MODE để có thể gửi dữ liệu lên firebase. Wifi sẽ kết nối với mạng cá nhân của người dung như mạng di động hoặc mạng wifi gia đình đều có thể được. Khi trong quá trình kết nối là không kết nối được thì quay lại cố gắng kết nối đến khi nào thành công. Khi kết nối thành công thì hiển thị kết nối thành công và địa chỉ ip mà thiết bị đã kết nối và đèn bultin trong esp sẽ sáng. Khi xong là việc chuẩn bị đã hoàn tất. Sau đó esp đọc dữ liệu cổng giao tiếp uart Esp nếu có dữ liệu thì đọc dữ liệu được gửi từ pic cho esp và kiểm tra nếu header có dữ liệu là “dt:” là data đúng thì tiếp tục kiểm tra subHeader nếu subHeader là “rt:” là real temperature là dữ liệu về nhiệt độ hồ cá đọc được từ cảm biến ds18b20 và chuyển đổi dữ liệu từ string thành int để gửi dữ liệu lên firebase dạng số nguyên theo đường dẫn là “data/temperature/realtime” với giá trị là giá trị được chuyển đổi, nếu sai thì kiểm tra bằng “lt:” là limit temperature là dữ liệu nhiệt độ giới hạn từ pic gửi cho esp và chuyển đổi từ string thành int và gửi dữ liệu lên firebase theo đường dẫn là “data/temperature/limit” với dữ liệu số nguyên đã được chuyển đổi, nếu sai thì kiểm tra bằng giá trị “ra:” là rtc alarm là thời gian đặt cho cá ăn được gửi từ pic và chuyển đổi thành số nguyên giờ, phút, giây và gửi dữ liệu lên firebase theo đường dẫn là “data/alarm/eat/hh-pp-ss” với giá trị được chuyển đổi, nếu sai thì thoát điều kiện, nếu không có dữ liệu của cổng espSerial thì không thực hiện so sánh header, subHeader. Tiếp theo là đọc thời gian hiện tại và kiểm tra nếu thời gian hiện tại trừ cho thời gian trước đó lớn hơn 10000(10s) sai thì thoát điều kiện và trở về chương trình chính, đúng thì đọc dữ liệu nhiệt độ giới hạn theo đường dẫn “set/temperature/limit” và gửi dữ liệu cho pic để cập nhật, sau đó đọc dữ liệu về thời gian cho cá ăn theo đường dẫn là “set/alarm/hh-pp-ss” và gửi dữ liệu cho pic để cập nhật, sau đó đọc dữ liệu và điều khiển servo cho cá ăn qua ứng dụng theo đường dẫn “control/servo” và gửi dữ liệu điều khiển cho pic để thực hiện, sau đó trở về chương trình chính để thực hiện lại từ đầu.

Chương trình code esp:

https://github.com/VM-Thuan-2003/do_an_1/tree/main/PROJECT/CODE_ESP_FINAL

4.3.1.3. Lưu đồ giải thuật cho ứng dụng di động android



Hình 4. 19 Lưu đồ app ứng dụng

Ban đầu app thiết lập các biến môi trường như giờ, phút, giây, giá trị nhiệt độ giới hạn, giờ phút giây cho cá ăn. Sau đó thiết lập biến dung useEffect Hook Triggered để dung cho việc cập nhật giá trị các biến theo framework của reactnative lập trình ứng dụng. Thiết lập

ĐỒ ÁN 1

vòng lặp nhắc lại 10s 1 lần để thực hiện việc đọc dữ liệu là fetch data Firebase để cập nhật dữ liệu cho các biến hiển thị trên màn hình ứng dụng. Sau đó cập nhật trạng thái ứng dụng với các biến mới, tiếp theo là render UI để hiển thị. Tiếp theo là đọc các trường dữ liệu trên màn hình để thu thập dữ liệu người dùng muốn thiết lập như thời gian cho cá ăn và nhiệt độ giới hạn cho cá phát triển. Nút nhấn trên màn hình thực hiện việc gửi dữ liệu lên firebase để điều khiển thiết bị pic ở bên dưới. cuối cùng là kết thúc.

Chương trình code app android:

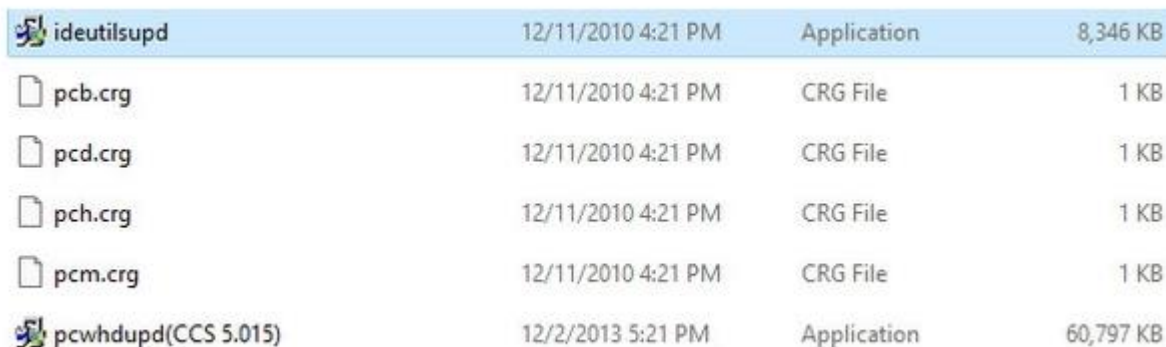
https://github.com/VM-Thuan-2003/do_an_1/tree/main/PROJECT/app-da1/appDA1







4.3.2. Phần mềm lập trình cho vi điều khiển

4.3.2.1. Phần mềm pic-c

B1. Cài phần mềm theo link: <https://linkxink.net/4eu9x>

B2. Cài file ideutilsupd

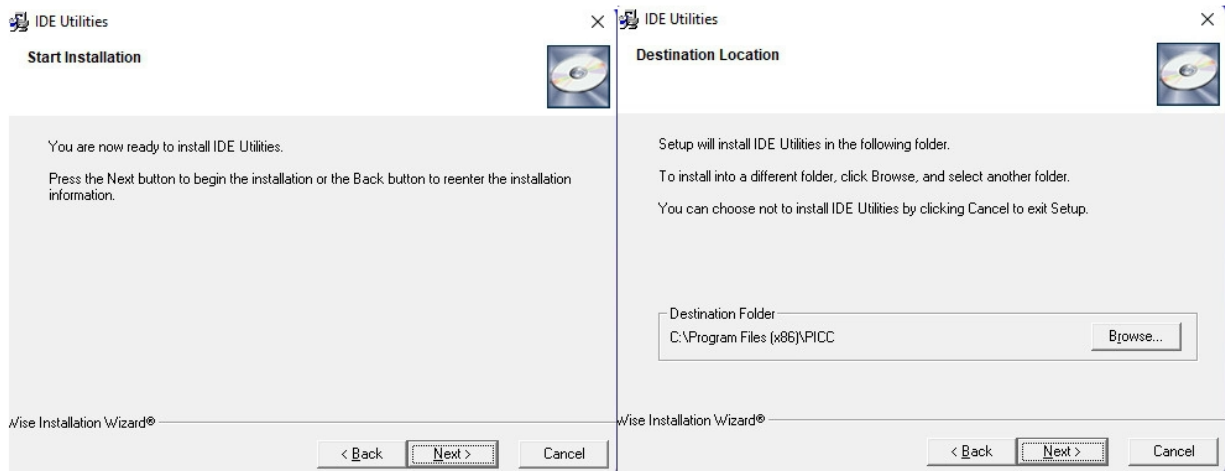


 ideutilsupd	12/11/2010 4:21 PM	Application	8,346 KB
 pcb.crg	12/11/2010 4:21 PM	CRG File	1 KB
 pcd.crg	12/11/2010 4:21 PM	CRG File	1 KB
 pch.crg	12/11/2010 4:21 PM	CRG File	1 KB
 pcm.crg	12/11/2010 4:21 PM	CRG File	1 KB
 pcwhdupd(CCS 5.015)	12/2/2013 5:21 PM	Application	60,797 KB

Hình 4. 20 Cài ideutilsupd

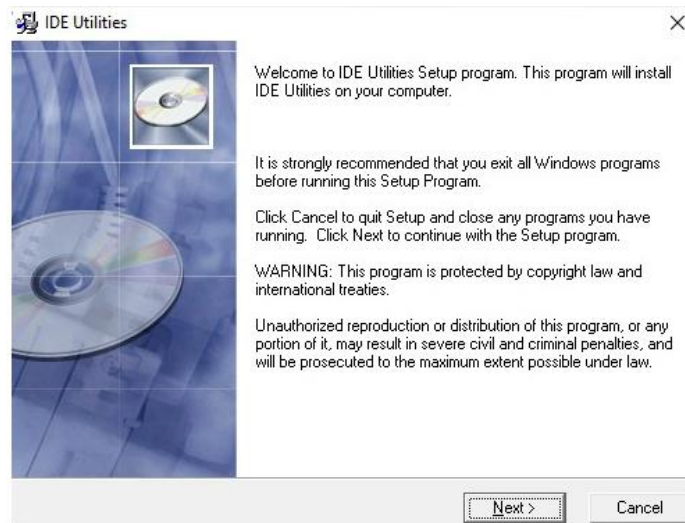
Nhấn 2 lần vào ideutilsupd để thực hiện cài đặt. Các bạn chọn NEXT và cuối cùng là Finish để hoàn thành file cài đặt.

ĐỒ ÁN 1

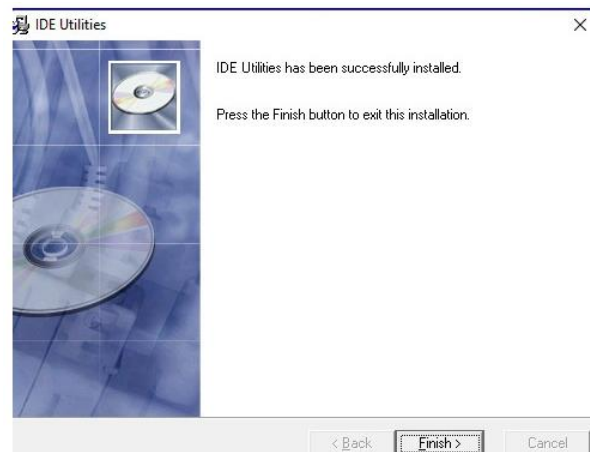


Hình 4. 21 Quá trình cài 1 pic

Chọn Browse để chọn nơi lưu trữ cho ứng dụng của bạn.



Hình 4. 22 Quá trình cài 2 pic

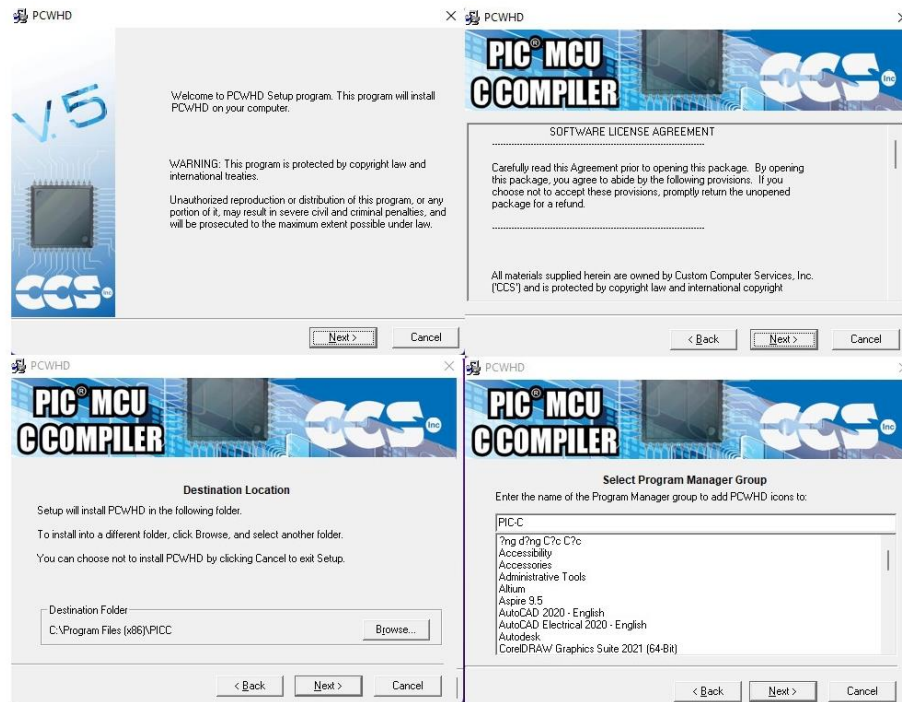


Hình 4. 23 Quá trình cài 3 pic

Nhấn Finish để hoàn thành việc cài đặt.

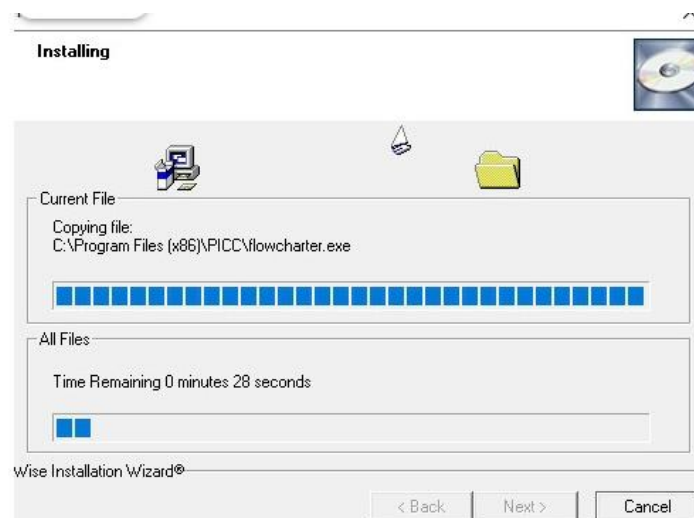
B3. Các bạn cài đặt file pcwhdupd(CCS 5.015).

Các bạn tiếp tục chọn NEXT cho đến khi xuất hiện bảng cài đặt.



Hình 4. 24 Cài pic-c

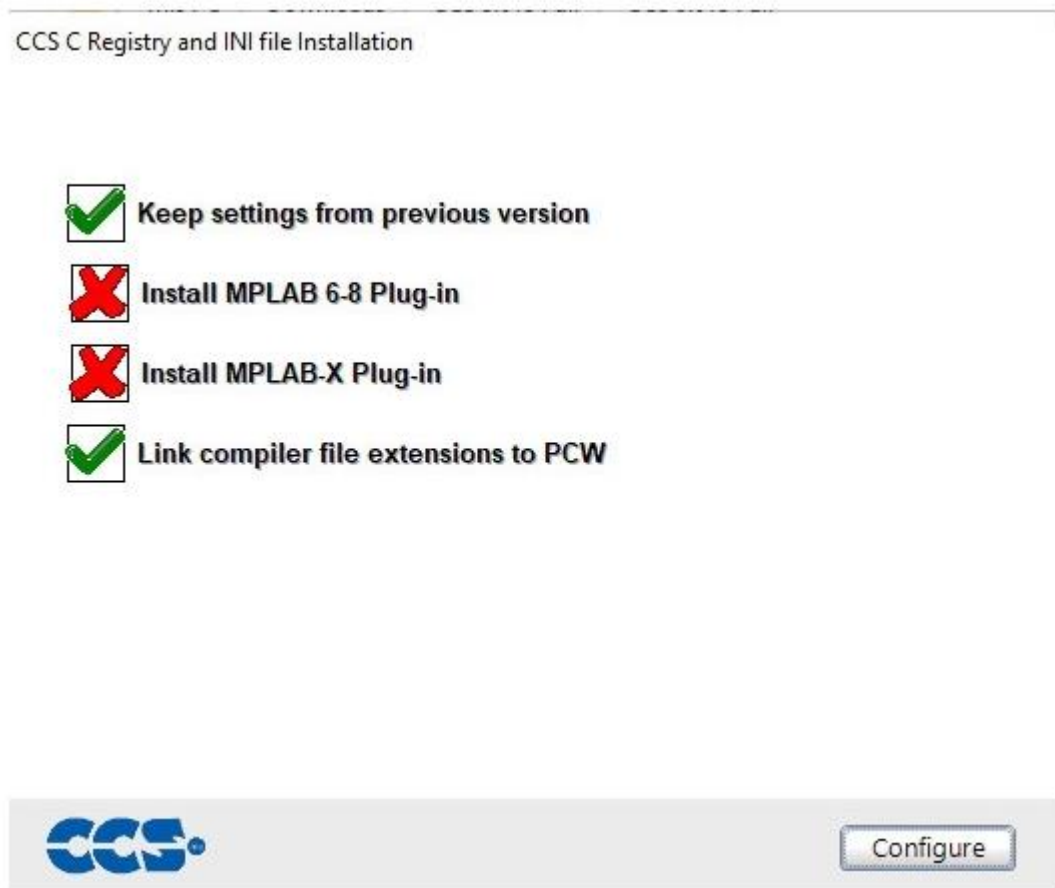
Sau đó sẽ xuất hiện File cài đặt các bạn chờ trong ít phút rồi chọn Finish.



Hình 4. 25 Thiết lập file cài đặt

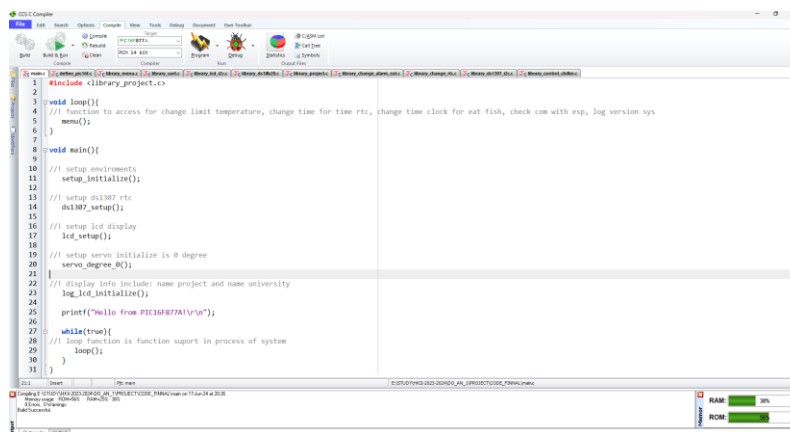
ĐỒ ÁN 1

Quá trình cài đặt hoàn tất.



Hình 4. 26 Chọn Configure

B4. Thực hiện lập trình code với pic-c



Hình 4. 27 Lập trình pic-c

Lập trình code của bạn với màn hình lập trình theo đề bài của bạn.

ĐỒ ÁN 1

B5. Upload code xuống pic để chạy chương trình, bạn nhấn nút compile để xử lý code tạo file Hex. Khi file Hex được tạo thì bạn có thể dung file Hex để mô phỏng hoặc sử dụng trên kit thực.

4.3.2.2. Phần mềm Arduino

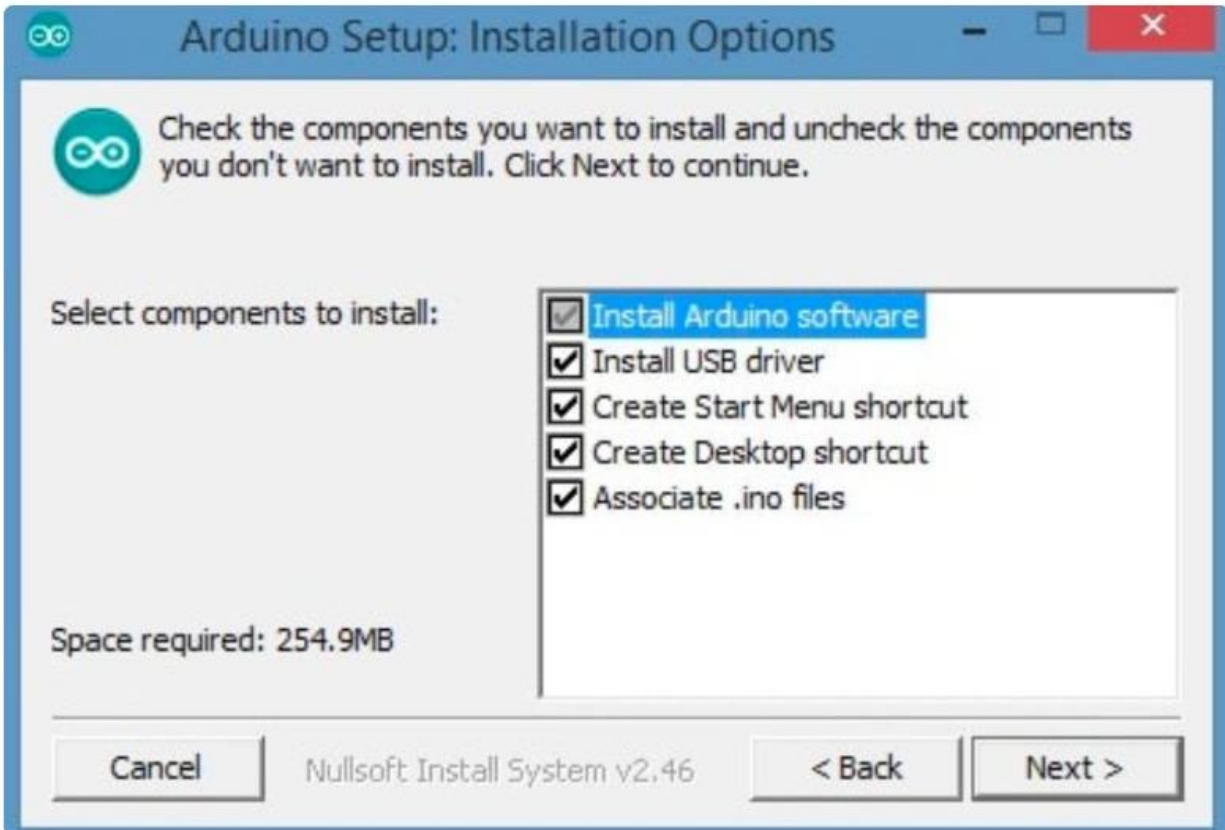
B1. Tải phần mềm Arduino mới nhất tại đây: <https://www.arduino.cc/en/software>

B2. Chọn file .exe để cài đặt. Chọn đồng ý để chấp thuận cấp phép:



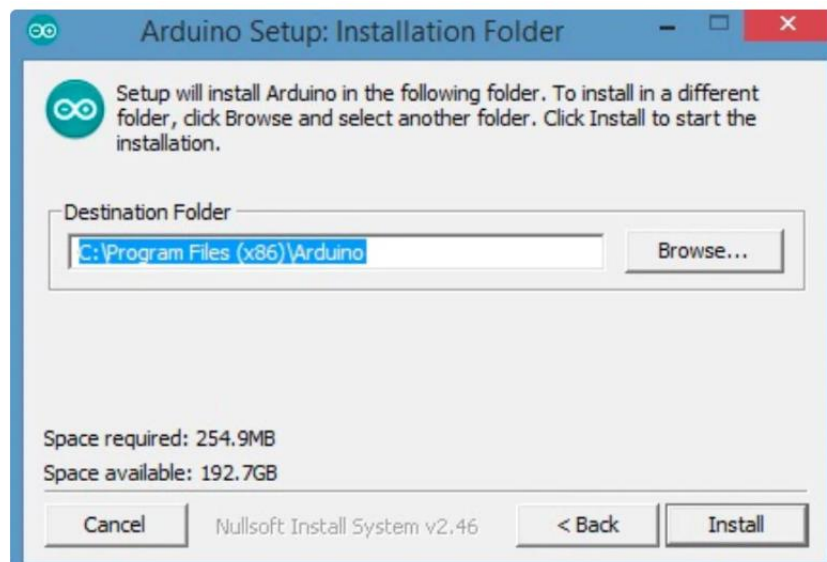
Hình 4. 28 Setup 1 arduino

B3. Quyết định thành phần nào sẽ cài đặt, sau đó nhấp vào tiếp theo (Next):



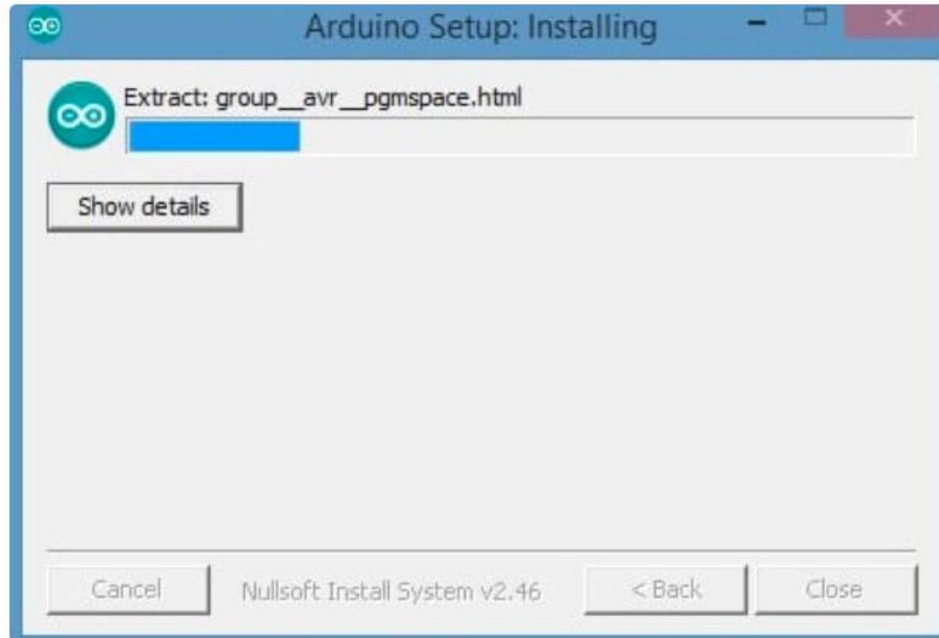
Hình 4. 29 setup 2 arduino

B4. Chọn thư mục để cài đặt chương trình, sau đó nhấp vào cài đặt (Install):



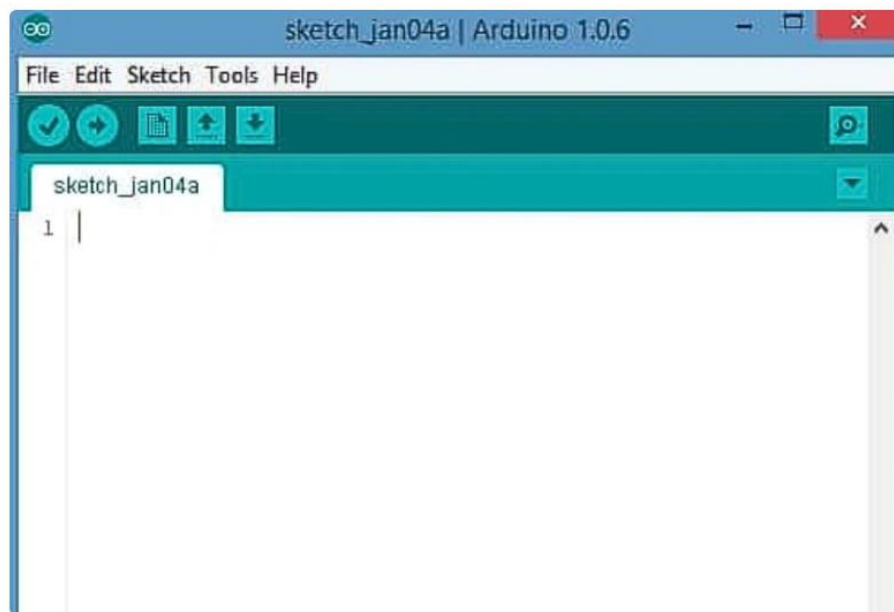
Hình 4. 30 setup 3 arduino

B5. Đóng chương trình cài đặt xong rồi nhấn “Close”:



Hình 4. 31 Setup 4 arduino

B6. Sau khi hoàn tất, bạn có thể tìm thấy lỗi tắt Arduino trên Màn hình của bạn và nhấp vào nó. IDE sẽ mở ra và bạn sẽ thấy trình soạn thảo:



Hình 4. 32 Finnal setup Arduino

B7. Thực hiện lập trình code với Arduino ide.

B8. Nhấn nút mũi tên sang phải để thực hiện việc compile để upload code xuống esp để chạy chương trình.

4.3.3. Phần mềm lập trình cho ứng dụng di động

B1. Cài nvm cho window để quản lí các phiên bản Node.js

- B11. Truy cập vào trang web sau: <https://github.com/coreybutler/nvm-windows>
- B12. Kéo xuống đọc phần README và chọn phần Download Now! Và vào phần Assets chọn download nvm-setup.exe để tải nvm về cho Window.

• B13. Khi đã tải file nvm-setup.exe về máy tính thì nhấn đúp vào file đã tải để cài đặt phần mềm

• B14. Khi đã nhấn đúp chuột để cài phần mềm thì chọn I accept the agreement để tiếp tục, tiếp theo chọn nơi để lưu cho file của bạn (khuyến khích lưu vào ổ C của máy để ít xảy ra lỗi), sau đó chọn nơi lưu Symlink thì để mặc định và nhấn Next cuối cùng chọn Install để thực hiện cài đặt.

B2. Cài phiên bản Node.js để thực hiện viết chương trình

- B21. Vào terminal của máy tính
- B22. Nhập lệnh `nvm install 20.11.0` là lệnh để tải môi trường Node.js phiên bản 20.11.0.

• B23. Khi lệnh chạy xong thì ta nhập lệnh `nvm use 20.11.0` để chọn phiên bản hiện tại ta sẽ sử dụng là 20.11.0

• B24. Để kiểm tra lại ta đã được chọn phiên bản hiện tại là 20.11.0 chưa thì ta nhập lệnh `nvm list` để kiểm tra lại phiên bản hiện tại của mình đã được trở vào 20.11.0 hay chưa, thành công thì con trỏ sẽ trở vào 20.11.0.

B3. Tạo dự án để viết app android và chạy chương trình trên môi trường Node.js

- B31. Ta vào Terminal để thực hiện tạo dự án
- B32. Ta dùng lệnh `cd path` với path là đường dẫn trở vào nơi bạn muốn lưu dự án
- B33. Sau đó ta nhập lệnh `npx create-expo-app android-app` để thực hiện tạo dự án sử dụng react-native thuộc môi trường Node.js

• B34. Khi tạo dự án thành công thì ta nhập lệnh `cd android-app` để vào dự án, nếu bạn code trên visual studio code thì ta thêm lệnh code. Để mở folder dự án đang làm việc lên visual studio code để làm việc.

ĐỒ ÁN 1

- B35. Để chạy chương trình dự án để mô phỏng giao diện giúp việc tạo app được thích nghi hơn thì ta nhập lệnh `npm start` để chạy chương trình.

- B36. Sau khi chạy xong có cho ta 1 đường link với 1 mã QR để kết nối. nếu ta lấy điện thoại thực để mô phỏng giao diện thì trên thiết bị thực ta cài phần expo go trên android và ios đều có.

- B37. Sau khi cài phần mềm xong:

- + Đối với điện thoại android thì ta vào điện thoại chọn quét QR hoặc nhập link thủ công đều được

- + Đối với điện thoại ios thì ta vào máy ảnh của máy quét mã QR thì sẽ được tự động chuyển qua phần mềm expo go với dự án hiện tại được gửi lên.

B4. Viết chương trình cá nhân hóa vào dự án android của chúng ta

<https://github.com/VM-Thuan->

[2003/TT_HTN_SPK/tree/main/PROJECT_CK/CODE/APP/app-android](https://github.com/VM-Thuan-2003/TT_HTN_SPK/tree/main/PROJECT_CK/CODE/APP/app-android)

B5. Biên dịch chương trình tạo file android-app.apk

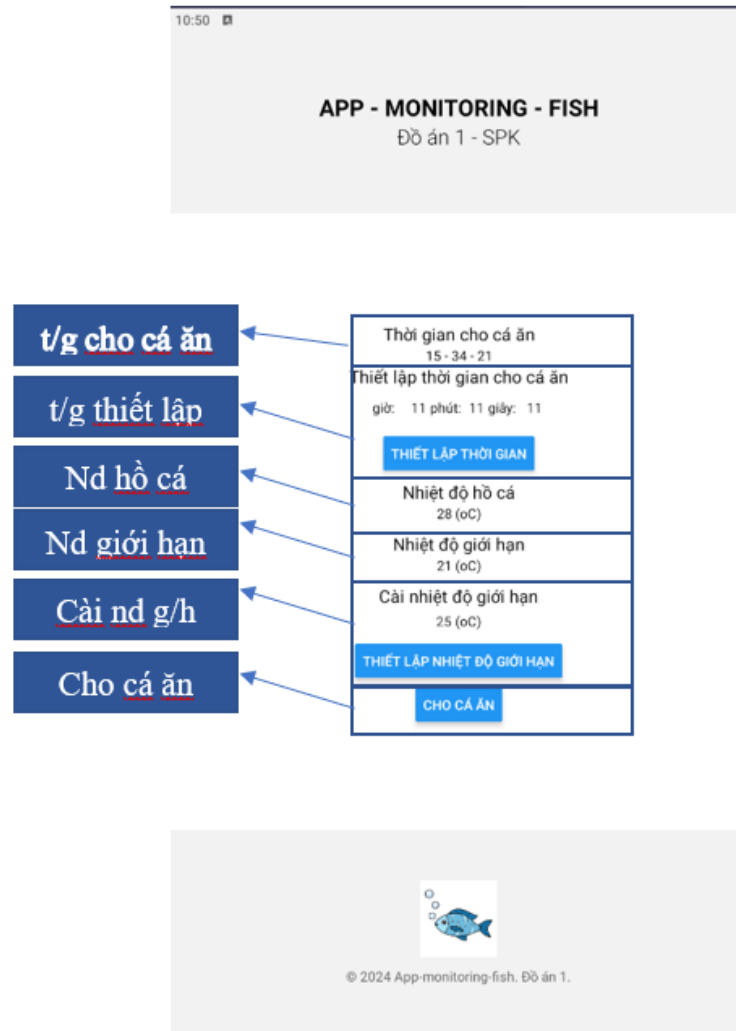
- B51. Tạo file eas.json và nội dung trong file eas.json



```
1  {
2    "build": {
3      "preview": {
4        "android": {
5          "buildType": "apk"
6        }
7      },
8      "preview2": {
9        "android": {
10         "gradleCommand": ":app:assembleRelease"
11       }
12     },
13     "preview3": {
14       "developmentClient": true
15     },
16     "production": {}
17   }
18 }
```

Hình 4. 33 Hình biên dịch App B5.1

- B52. Mở terminal trong dự án hiện tại và nhập lệnh `npm install -g eas-cli` để sử dụng dịch vụ EAS cho terminal của bạn.
 - B53. Sau đó nhập lệnh `eas login` để đăng nhập vào eas (đăng ký tài khoản trên expo.dev)
 - B54. Sau đó nhập lệnh `eas build -p android` để tạo 1 profile lên Android Play Store build của expo
- B55. Sau đó nhập lệnh `eas build -p android --profile preview` để tạo file apk.
- * Giao diện App



Hình 4. 34 Hình giao diện app

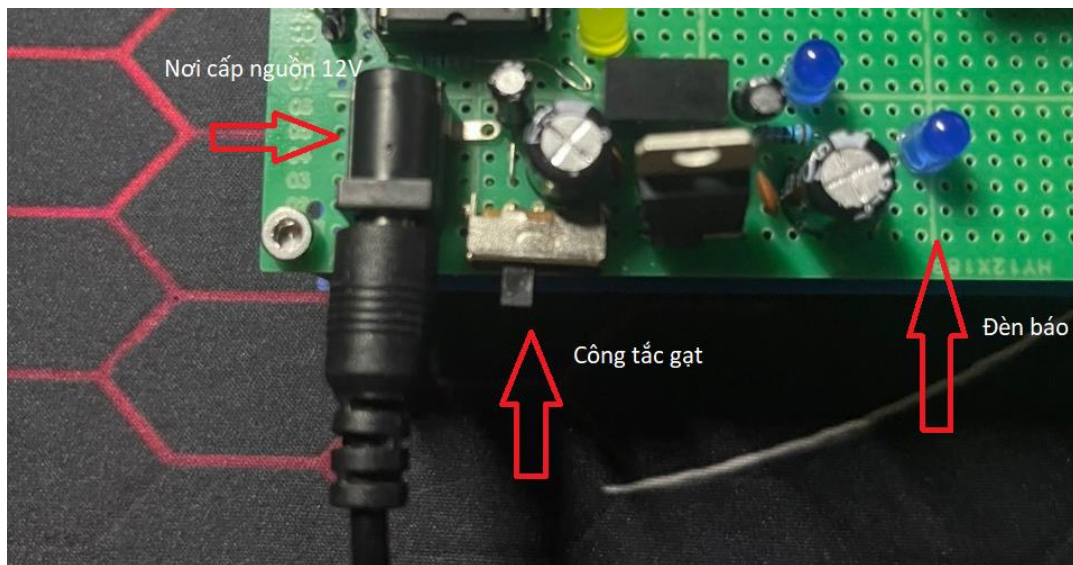
Hình giao diện app gồm hiển thị thời gian cho cá ăn, thời gian cho cá ăn có được cài trên ứng dụng và nhấn thiết lập thời gian để gửi dữ liệu cho pic để thiết lập thời gian cho cá ăn. Và hiển thị nhiệt độ hồ cá và nhiệt độ giới hạn, hiển thị cài nhiệt độ giới hạn và có nút nhấn để gửi dữ liệu cho pic để thiết lập nhiệt độ giới hạn. nút nhấn cho cá ăn để cho cá ăn mà không theo thời gian thiết lập sẵn.

4.4. HƯỚNG DẪN SỬ DỤNG, THAO TÁC

Sau đây là một số hướng dẫn sử dụng hệ thống dành cho người dùng thao thác một cách dễ dàng

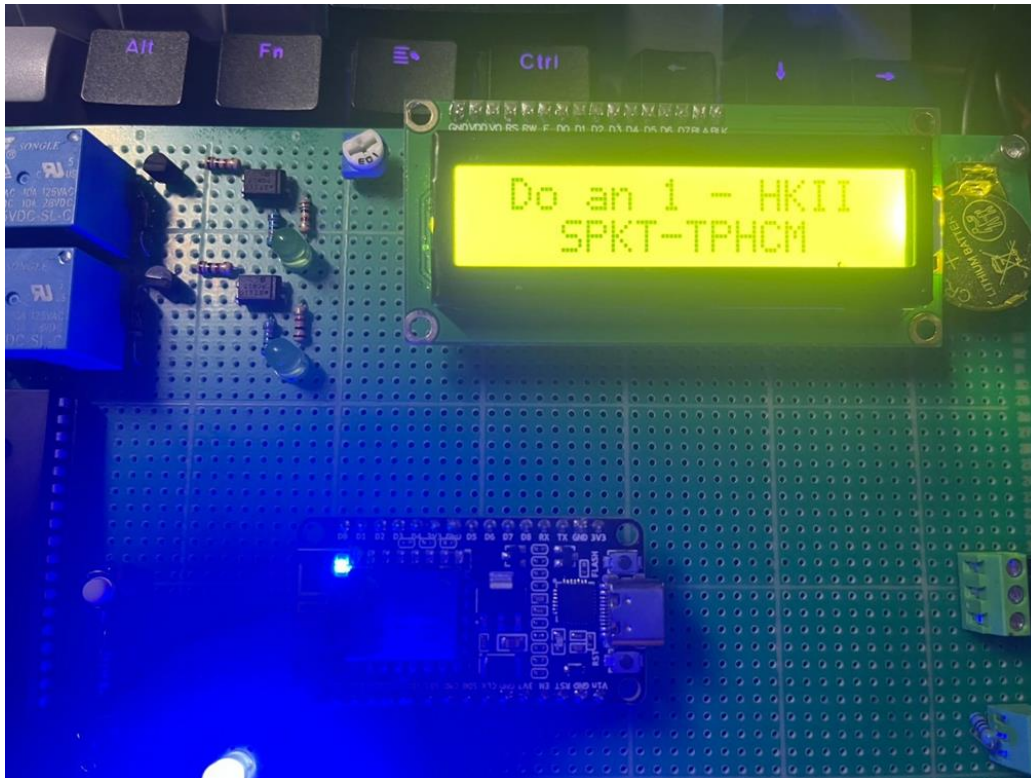
ĐỒ ÁN 1

Bước 1 : Cấp nguồn 12V cho hệ thống, ở đây sử dụng adapter 12V. Sau khi cấp nguồn thì gạt công tắc mở nguồn, đèn báo hiệu có điện sáng lên

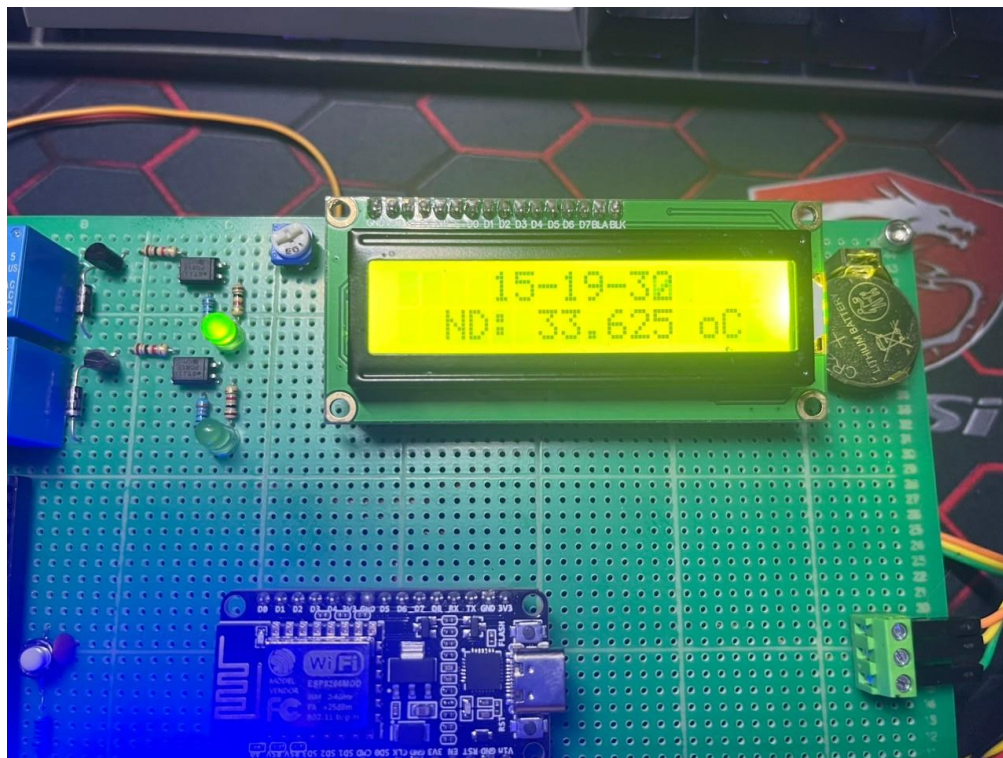


Hình 4. 35 Hình công tắc, đèn báo, nguồn cấp

Bước 2: Lúc này, đèn báo nguồn đã sáng và LCD sẽ hiện đoạn kí tự giới thiệu sản phẩm. Sau một khoảng thời gian khoảng 3s thì màn hình LCD sẽ hiển thị chế độ 1 - ngày tháng năm ở hàng 0 và nhiệt độ của bể cá đo được từ cảm biến sẽ hiển thị ở hàng 1.



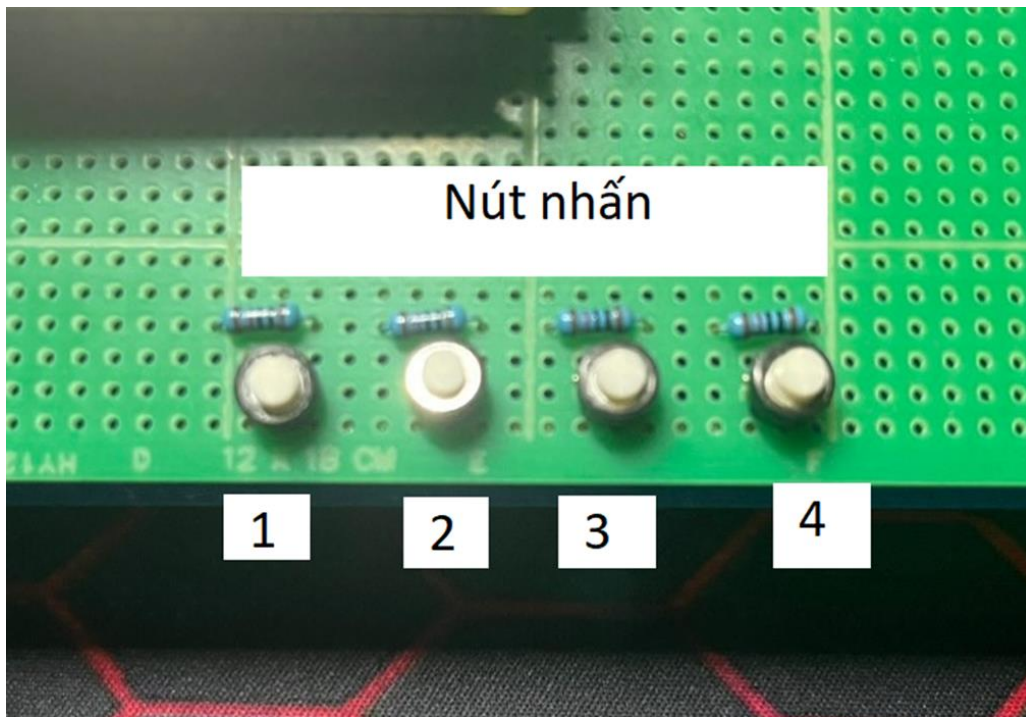
Hình 4. 36 Hình hiển thị lcd – thông tin ban đầu



Hình 4. 37 HÌnh hiển thị lcd với nhiệt độ và thời gian

ĐỒ ÁN 1

Ta có thể chọn các chế độ vận hành và cài đặt thông số của hệ thống bằng các nút nhấn.



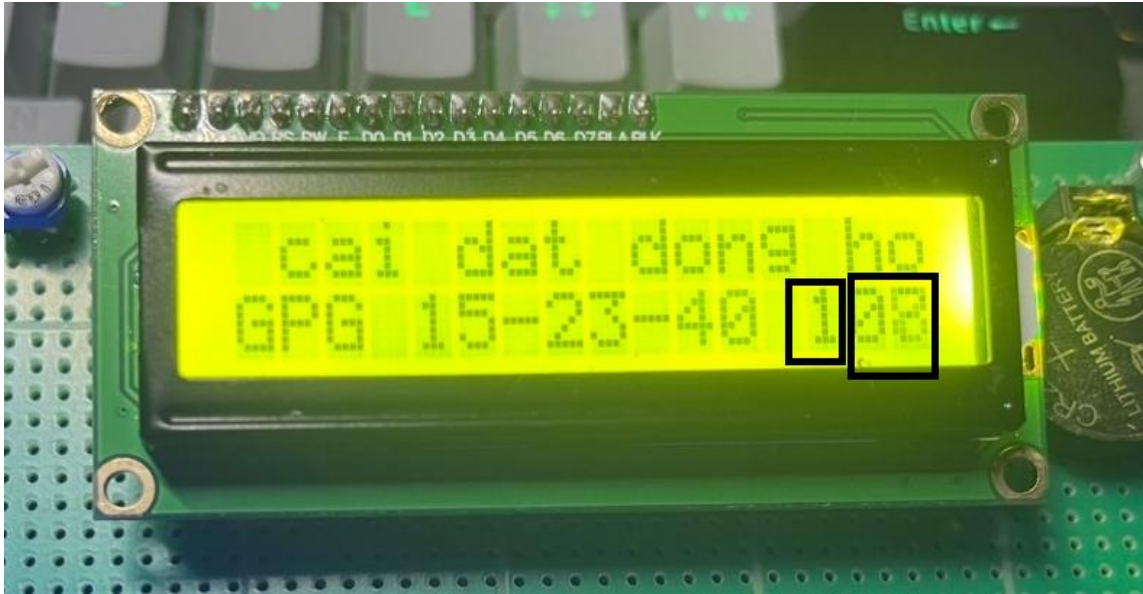
Hình 4. 38 Hình vị trí nút nhấn

Bước 3: Nhấn nút nhất số 4 lần 1. Lúc này hệ thống sẽ chuyển sang chế độ 2 – Hiển thị nhiệt độ hiện tại và nhiệt độ cài giới hạn. Ta có thể thao tác tăng nhiệt độ cài bằng nút nhấn số 3 và giảm nhiệt độ cài bằng nút nhấn số 2.



Hình 4. 39 Hình hiển thị ở handle_2

Bước 4: Nhấn nút nhất số 4 lần 2. Lúc này hệ thống sẽ chuyển sang chế độ 3 – Cài đặt đồng hồ.



Hình 4. 40 Hình hiển thị ở handle_3

Như ảnh trên ô 13 của hàng 1 sẽ hiển thị đối tượng cần chỉnh .

Số	1	2	3	4	5	6	7
Đối tượng	Giây	Phút	Giờ	Năm	Tháng	Ngày	Thứ

Đối tượng nào được chỉnh thì sẽ được nhấp nhấp liên tục trong 20s. Sau 20s đối tượng đó sẽ tự động được lưu. Ô số 14 và 15 của hàng số 1 sẽ hiển thị đếm 20s.

Đầu tiên sẽ là giao diện chỉnh giờ phút giây.

Nhấn nút nhấn số 1 lần 1 sẽ chỉnh giây.

Nhấn nút nhấn số 1 lần 2 sẽ chỉnh phút.

Nhấn nút nhấn số 1 lần 3 sẽ chỉnh giờ.



ĐỒ ÁN 1

Hình 4. 41 Hình hiển thị ở handle_3 chỉnh gpg

Tiếp theo là giao diện chỉnh ngày tháng năm:

Nhấn nút nhấn số 1 lần 4 sẽ chỉnh năm.

Nhấn nút nhấn số 1 lần 5 sẽ chỉnh tháng.

Nhấn nút nhấn số 1 lần 6 sẽ chỉnh ngày.



Hình 4. 42 Hình hiển thị ở handle_3 chỉnh ntn

Và cuối cùng là giao diện chỉnh thứ:

Nhấn nút nhấn số 1 lần 7 sẽ chỉnh thứ.



Hình 4. 43 Hình hiển thị ở handle_3 chỉnh thu

Bước 5: Nhấn nút nhất số 4 lần 3. Lúc này hệ thống sẽ chuyển sang chế độ 4 – Cài đặt thời gian cho cá ăn.

ĐỒ ÁN 1

Lưu ý: Đây chỉ là cài đặt một mốc thời gian trong 1 ngày và sẽ được lặp lại vào 24h sau. Các thao tác chỉnh giống như chỉnh giờ phút giây ở chế độ 3. Khi đã đến giờ cài đặt thì PIC sẽ điều khiển servo được đặt ở bể, giúp gạt thức ăn xuống bể.



Hình 4. 44 Hình hiển thị ở handle_4

Bước 6: Nhấn nút nhất số 4 lần 4. Lúc này hệ thống sẽ chuyển sang chế độ 5 – Giao diện kiểm tra kết nối giữa ESP8266 và PIC 16F877A.



Hình 4. 45 Hình hiển thị ở handle_5

Bước 7: Nhấn nút nhất số 4 lần 5. Lúc này hệ thống sẽ chuyển sang chế độ 6 – Giao diện version của hệ thống.

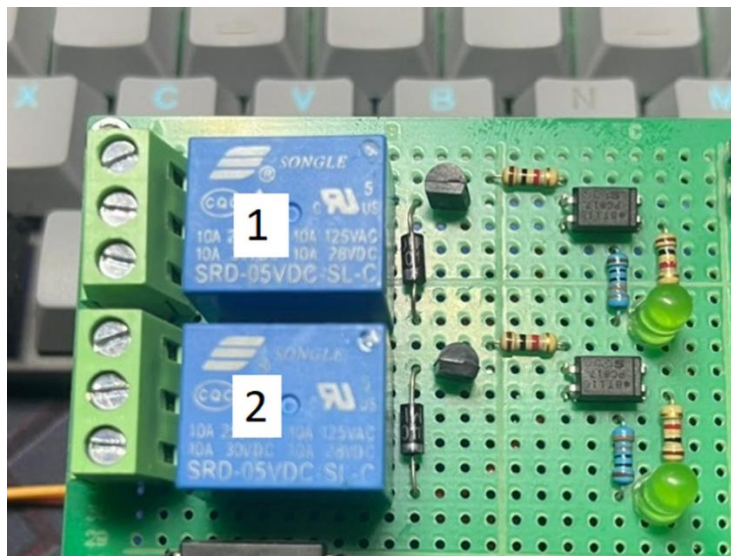


Hình 4. 46 Hình hiển thị ở handle_6

Bước 8: Kết nối các thiết bị làm nóng và làm lạnh nước để luôn đảm bảo nhiệt độ nước trong bể cá luôn ổn định ở mức nhiệt độ cài:

- Relay số 1 sẽ kết nối với thiết bị làm lạnh.
- Relay số 2 sẽ kết nối với thiết bị làm nóng.

Khi relay nào được bật thì sẽ có đèn báo để cho người dung biết được relay có đang hoạt động hay không.



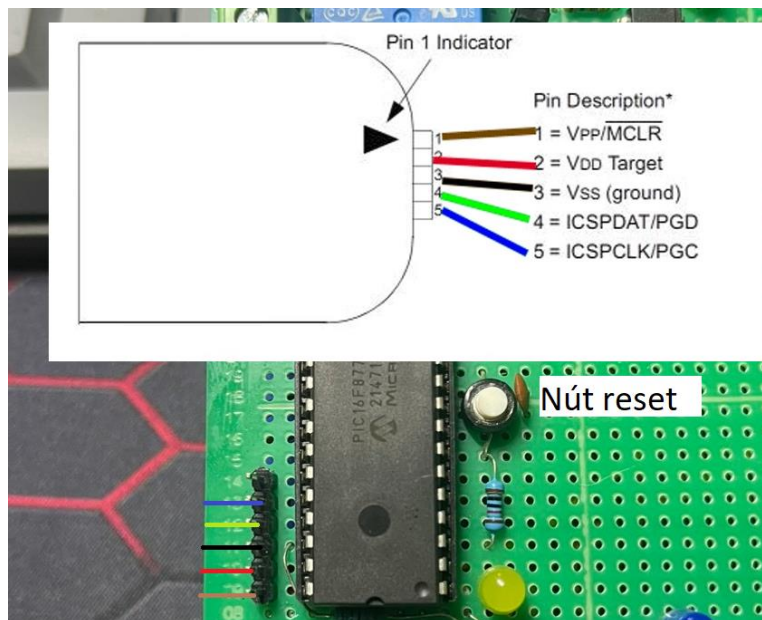
Hình 4. 47 Hình 2 relay

Bước 9: Khởi động lại, khắc phục lỗi hoặc muốn thay đổi chương trình của vi điều khiển:

- Nút reset giúp khởi động lại hệ thống.

ĐỒ ÁN 1

- Hàng rào chân được hàn sẵn ra bên ngoài giúp cho việc nạp chương trình dễ dàng hơn cho người dung. Kết nối kit nạp với các chân được hàn sẵn của PIC như hình.



Hình 4. 48 Hình chân cắm kết nối pic-kit2

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

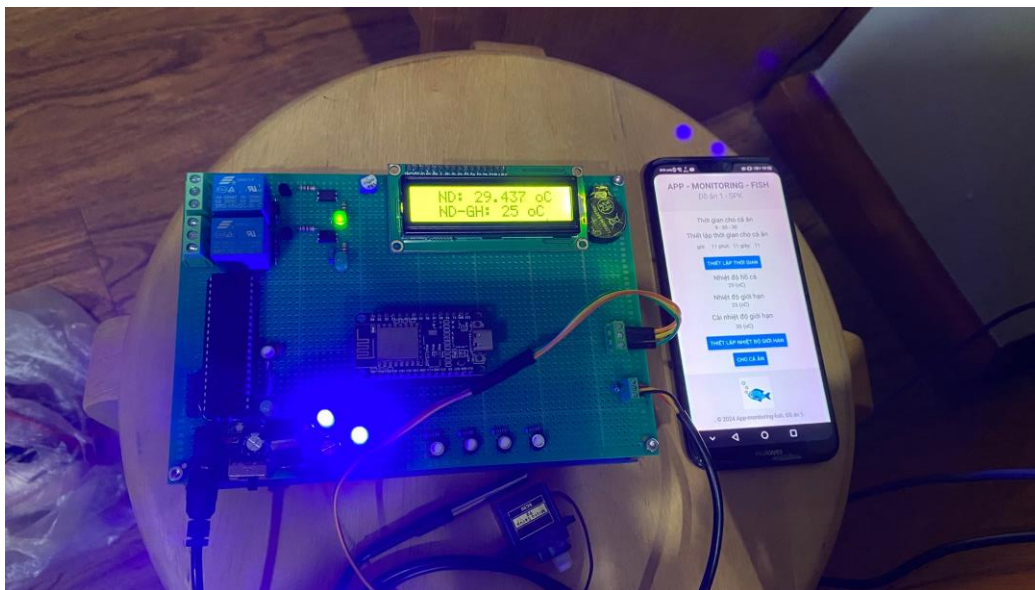
- Kết quả:

Tiến hành chạy thử hệ thống, hệ thống đo nhiệt độ của nước khá chính xác qua nhiều lần đo với các nhiệt độ nước khác nhau.

Lần đo	Nước đá	Nước bình thường	Nước ấm
1	16.5 độ C	27.3 độ C	35.5 độ C
2	16.5 độ C	26.9 độ C	35.6 độ C
3	16.7 độ C	27.1 độ C	35.5 độ C

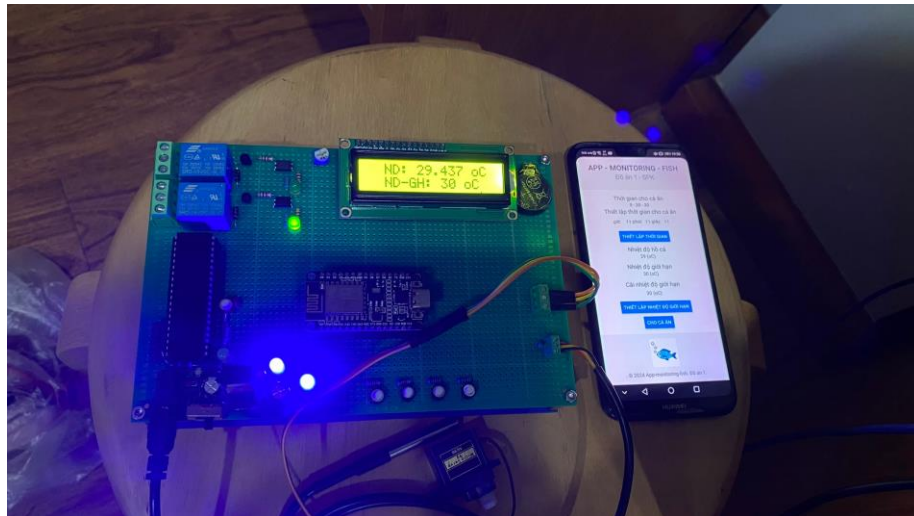
Bảng 5. 1 Bảng đo nhiệt độ cảm biến ds18b20

Khi nhiệt độ của nước ở trên ngưỡng nhiệt độ cài đặt thì relay điều khiển thiết bị làm mát được bật. Như trong ảnh đã hiển thị rõ nhiệt độ của hồ hiện tại là 29 độ so với nhiệt độ ngưỡng là 25 độ thì lúc này relay của thiết bị làm mát được bật lên



Hình 5. 1 Hình kết quả 1

Khi nhiệt độ của nước ở dưới ngưỡng nhiệt độ cài đặt thì relay điều khiển thiết bị làm nóng được bật. Như trong ảnh đã hiển thị rõ nhiệt độ của hồ hiện tại là 29 độ so với nhiệt độ ngưỡng là 30 độ thì lúc này relay của thiết bị làm nóng được bật lên



Hình 5. 2 Hình kết quả 2

Khi chưa đến thời gian cho cá ăn



Hình 5. 3 Hình kết quả 3

Khi chưa nhấn trên ứng dụng thì hướng servo ban đầu là hướng lên.

ĐỒ ÁN 1

Và sau khi đến thời gian cho cá ăn, servo sẽ gạt qua 1 góc 90 độ



Hình 5. 4 Hình kết quả 4

- Nhận xét và đánh giá

Tất cả các thao tác phần cứng như cài nhiệt độ giới hạn, điều chỉnh giờ, phút, giây, ngày, tháng, năm, thứ đều thực hiện đúng như ý muốn, những hệ thống gửi dữ liệu vẫn còn bị mất dữ liệu nên việc gửi dữ liệu lên firebase bị ngắt quãng.

Phần mềm trên điện thoại cũng hoạt động tốt chức năng theo dõi nhiệt độ, cài nhiệt độ giới hạn gửi dữ liệu lên firebase tốt.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. KẾT LUẬN

Hệ thống đã hoàn thành hơn 95% so với mục tiêu đề ra ban đầu. Điều thực hiện đầy đủ các chức năng như là đọc nhiệt độ, cài nhiệt độ giới hạn, đọc thời gian thực sau đó hiển thị và cho phép điều chỉnh, có phần mềm kiểm soát và có thể thao tác trên phần mềm. Tóm lại, hệ thống có sai số đo lường thấp, cực kì dễ sử dụng và thân thiện với người dùng. Tuy nhiên ở đây hệ thống ở đây chưa có tính thẩm mỹ cao vì được thi công trên mạch hàn chứ không phải mạch in và thời gian đáp ứng hơi chậm, khoảng 20s-1p tùy theo kết nối mạng

6.2. HƯỚNG PHÁT TRIỂN

Hệ thống giám sát hồ cá là một đề tài tuy không quá khó như việc áp dụng vi điều khiển PIC16F877A trong giám sát nhiệt độ hồ cá và cho cá ăn tự động mang lại nhiều lợi ích rõ rệt. Chúng em sẽ hướng đến việc cải tiến hệ thống sử dụng năng lượng tái tạo (năng lượng mặt trời), tích hợp cảm biến để phân tích dữ liệu chất lượng của nước.... Hệ thống này nếu được ứng dụng trong lĩnh vực chăn nuôi thủy hải sản thì không chỉ cải thiện điều kiện nuôi trồng mà còn tối ưu hóa quy trình, tiết kiệm nguồn lực và nâng cao hiệu quả kinh tế. Với sự tiến bộ của công nghệ, việc ứng dụng các hệ thống tự động hóa như thế này sẽ ngày càng trở nên phổ biến, góp phần vào sự phát triển bền vững của ngành nuôi trồng thủy sản.

TÀI LIỆU THAM KHẢO

- [1]Trang hỗ trợ tìm kiếm thông tin - <https://vi.wikipedia.org>
- [2]Nguyễn Đình Phú, “Thực hành vi điều khiển - PIC”, Đại học Sư Phạm Kỹ Thuật Tp.HCM 2016.
- [3]Trang chủ cung cấp thông số linh kiện của nhà sản xuất - www.alldatasheet.com
- [4]<https://www.thegioiic.com/tin-tuc/gioi-thieu-ve-chuan-giao-tiep-uart>
- [5]<https://www.thegioiic.com/tin-tuc/tim-hieu-ve-chuan-giao-tiep-i2c>
- [6]<https://lidinco.com/tu-dien-cuon-cam-dien-tro>