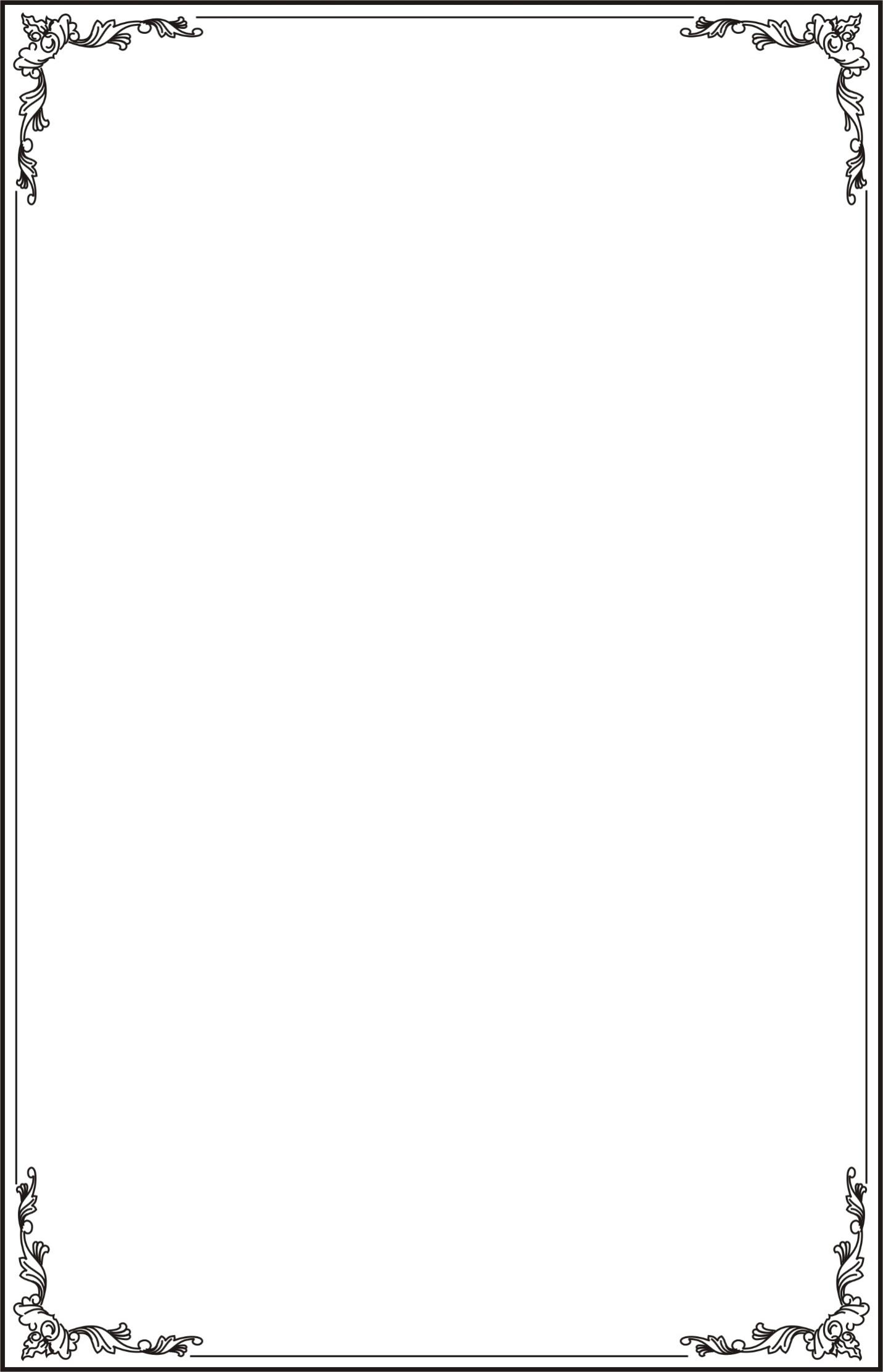
****TRƯỜNG ĐẠI HỌC**

**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 THƯƠNG-21161367**

**Khoá**: **2021-2025**

**Ngành**: **CNKT ĐIỆN TỬ - VIỄN THÔNG**

**GVHD**: **TH.S NGUYỄN ĐÌNH PHÚ**

TP-HCM, Ngày Tháng 04 Năm 2024

|  |
| --- |
| Khoa Điện-Điện Tử, Đại học Sư Phạm Kỹ Thuật TP.HCM |

**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:

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 dựa trên lập trình vi điều khiển từ khối xử lý trung tâm giao tiếp với các khối ngoại vi để xuất kết quả hiển thị được lên LCD 16x2, sử dụng () để cài đặt thời gian cho cá ăn, sử dụng () để cân bằng nhiệt độ trong hồ cá.
4. Sản phẩm: Mạch bao gồm các khối như khối xử lý trung tâm, khối hiển thị, khối thời gian thực, khối nút nhấn,......

GIẢNG VIÊN HƯỚNG DẪN

|  |  |
| --- | --- |
| Khoa Điện-Điện Tử, Đại học Sư Phạm Kỹ Thuật TP.HCM |  |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

**NHẬN XÉT**

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

.........................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Ưu điểm:

.........................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Khuyết điểm:

.........................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Điểm sinh viên thực hiện 1 :.......................................................................................................
2. Điểm sinh viên thực hiện 2 :.......................................................................................................

Tp*. Hồ Chí Minh, ngày tháng năm 2024*

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

|  |  |
| --- | --- |
|  | Khoa Điện-Điện Tử, Đại học Sư Phạm Kỹ Thuật TP.HCM |

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 nổ 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 xót. 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** 7](#_Toc163614690)

[**DANH MỤC BẢNG BIỂU** 8](#_Toc163614691)

[**CHƯƠNG 1: TỔNG QUAN** 9](#_Toc163614692)

[1.1. GIỚI THIỆU TÌNH HÌNH NGHIÊN CỨU HIỆN NAY 9](#_Toc163614693)

[1.2. MỤC TIÊU NGHIÊN CỨU 9](#_Toc163614694)

[1.3. NHIỆM VỤ NGHIÊN CỨU 9](#_Toc163614695)

[1.4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 9](#_Toc163614696)

[**1.4.1. Đối tượng nghiên cứu** 9](#_Toc163614697)

[**1.4.2.Phạm vi nghiên cứu** 10](#_Toc163614698)

[1.5. PHƯƠNG PHÁP NGHIÊN CỨU 10](#_Toc163614699)

[1.6. BỐ CỤC CỦA ĐỒ ÁN 10](#_Toc163614700)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 11](#_Toc163614701)

[2.1. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN PIC 16F877A 11](#_Toc163614702)

[**2.1.1.** **Vi điều khiển PIC 16F877A** 11](#_Toc163614703)

[**2.1.2.** **Kiến trúc của PIC 16F877A** 12](#_Toc163614704)

[**2.1.3.** **Sơ đồ nối chân và chức năng của từng chân** 13](#_Toc163614705)

[2.2. CẢM BIẾN NHIỆT ĐỘ IC DS18B20 16](#_Toc163614706)

[**2.2.1.** **Giới thiệu chung về DS18B20** 16](#_Toc163614707)

[**2.2.2.** **Sơ đồ chân của DS18B20** 16](#_Toc163614708)

[2.3. IC THỜI GIAN THỰC DS1307 18](#_Toc163614709)

[**2.3.1 Giới thiệu về DS1307** 18](#_Toc163614710)

[**2.3.2. Sơ đồ chân của DS1307** 18](#_Toc163614711)

[**2.3.3. Tổ chức thanh ghi trong DS1307** 19](#_Toc163614712)

[**2.3.4.Cách hoạt động và một số chú ý khi sử dụng DS1307** 21](#_Toc163614713)

[2.4. HIỂN THỊ - LCD 16x02 21](#_Toc163614714)

[2.5. MỘT SỐ LÝ THUYẾT CƠ BẢN ĐƯỢC SỬ DỤNG 23](#_Toc163614715)

[**2.5.1. I2C** 23](#_Toc163614716)

[**2.5.2. UART** 25](#_Toc163614717)

[**CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG** 27](#_Toc163614718)

[3.1. GIỚI THIỆU 27](#_Toc163614719)

[3.2. TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG 27](#_Toc163614720)

[**3.2.1** **Thiết kế sơ đồ khối hệ thống** 27](#_Toc163614721)

[**3.2.2** **Tính toán và thiết kế mạch** 27](#_Toc163614722)

[**3.2.3** **Sơ đồ nguyên lý của toàn mạch** 27](#_Toc163614723)

[**CHƯƠNG 4:** **THI CÔNG HỆ THỐNG** 27](#_Toc163614724)

[4.1. GIỚI THIỆU 27](#_Toc163614725)

[4.2. THI CÔNG HỆ THỐNG 27](#_Toc163614726)

[**4.2.1** **Thi công bo mạch** 27](#_Toc163614727)

[**4.2.2** **Lắp ráp và kiểm tra** 27](#_Toc163614728)

[4.3. ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH 27](#_Toc163614729)

[**4.3.1** **Đóng gói bộ điều khiển** 27](#_Toc163614730)

[**4.3.2** **Thi công mô hình** 27](#_Toc163614731)

[4.4. LẬP TRÌNH HỆ THỐNG 27](#_Toc163614732)

[**4.4.1** **Lưu đồ giải thuật** 27](#_Toc163614733)

[**4.4.2** **Phần mềm lập trình cho vi điều khiển** 27](#_Toc163614734)

[**4.4.3** **Phần mềm lập trình cho điện thoại, máy tính, …** 27](#_Toc163614735)

[4.5. LẬP TRÌNH MÔ PHỎNG 27](#_Toc163614736)

[**4.5.1** **Lưu đồ** 27](#_Toc163614737)

[**4.5.2** **Xử lý tín hiệu hay hình ảnh** 27](#_Toc163614738)

[4.6. VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC 27](#_Toc163614739)

[**4.6.1** **Viết tài liệu hướng dẫn sử dụng** 28](#_Toc163614740)

[**4.6.2** **Quy trình thao tác** 28](#_Toc163614741)

[**CHƯƠNG 5: KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ** 28](#_Toc163614742)

[5.1. CẢM BIẾN 28](#_Toc163614743)

[5.2. KẾT QUẢ MÔ PHỎNG 28](#_Toc163614744)

[**5.2.1** **Hình ảnh tiền xử lý** 28](#_Toc163614745)

[**5.4.2** **Kết quả thống kê** 28](#_Toc163614746)

[**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 28](#_Toc163614747)

[6.1. KẾT LUẬN 28](#_Toc163614748)

[6.2. HƯỚNG PHÁT TRIỂN 28](#_Toc163614749)

[**TÀI LIỆU THAM KHẢO** 28](#_Toc163614750)

[**PHỤ LỤC** 28](#_Toc163614751)

.

# **DANH MỤC HÌNH ẢNH**

# **DANH MỤC BẢNG BIỂU**

# **CHƯƠNG 1: TỔNG QUAN**

* 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. **MỤC TIÊU NGHIÊN CỨU**

Áp dụng các kiến thức đã học từ các môn học như Mạch điện, Điện tử cơ bản, Thực tập Điện tử cơ bản, Kỹ thuật Vi Xử Lý, Ngôn ngữ lập trình C.

()

Khi nghiên cứu đề tài này, chúng em muốn sử dụng vi điều khiển kết hợp với các module để tạo ra một 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 hay các project sau này

* 1. **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 setup sẵn.

* 1. **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

**1.4.1. Đối tượng nghiên cứu**

Về phần cứng:

* Vi điều khiển PIC 16F877A
* Module thời gian thực DS1307.
* Nút nhấn.
* Module LCD 16x2.
* Buzzer.

Về phần mềm:

* Ngôn ngữ lập trình vi xử lý cho CCS.
* Viết code trên phần mền CCS.
* Mô phỏng phần cứng trên “Proteus 8 Professional”.

**1.4.2.Phạm vi nghiên cứu**

Trong thực tế, mô hình của hệ thống được lắp đặt ở cá bể cá tại nhà nên kích thước của hệ thống sẽ ở mức vừa để phù hợp với nhu cầu của người dùng. Trên thị trường có rất nhiều sản phẩm về kiểm soát nhiệt độ hồ cá, do vấn đề thời gian và kinh tế nên em đã chọn thiết kế hệ thống đơn giản hơn nhưng vẫn đảm bảo được các chức năng chính có kết hợp hiển thị LCD và nút nhấn điều chỉnh thời gian cho cá ăn.

* 1. **PHƯƠNG PHÁP NGHIÊN CỨU**

Để thực hiện được đề tài này, emđã 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 và xuất dữ liệu trên vi diều 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. **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 kê 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 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**

* 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 đươc từ môn Vi xử lý và Thực tập vi xử lý, em đã quyết định chọn PIC 16F877A.

1. **Vi điều khiển PIC 16F877A**

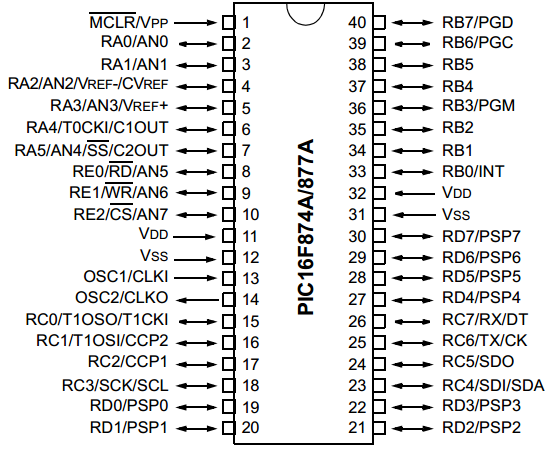
|  |  |
| --- | --- |
| Bộ vi điều khiển PIC16F877A là một trong những bộ vi điều khiển nổi tiếng nhất trong giới kỹ thuật điện tử. Bộ vi điều khiển này có nhiều ưu điểm thuận tiện cho việc sử dụng, lập trình chương trình điều khiển cũng rất dễ dàng.  Một trong các ưu điểm nổi bật là có thể ghi xóa nhiều 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… | Kết quả hình ảnh cho pic 16f877a  **Hình 2.1.** Vi điều khiểnPIC 16F877A – I/P thực tế |

1. **Kiến trúc của PIC 16F877A**

* **Cấu trúc cơ bảng 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 Biển 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: * Có 5 port xuất/nhập. * Có 8 kênh chuyển đổi A/D 10-bit. * Có 2 bộ PWM. * Có 3 bộ định thời: timer0, timer1 và timer2. * Có giao tiếp truyền nối tiếp: chuẩn RS 232, I2C… * Có giao tiếp LCD. | Ảnh có chứa văn bản, ngoài trời, xanh lục  Mô tả được tạo tự động  **Hình 2.2.** Cấu trúc cơ bản của vi điều khiển PIC 16F877A |

1. **Sơ đồ nối chân và chức năng của từng chân**

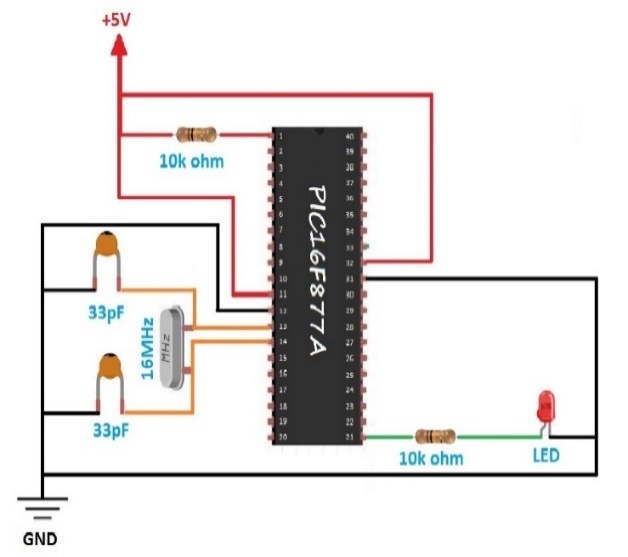


**Hình 2.3.** Sơ đồ chân vi điều khiển PIC 16F877A

**Bảng 2.1:** Chức năng sơ đồ chân của 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 |
| 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  VREE+: 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 |
| 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 ¼ 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 tin hiệu ngắt ngoài |
| 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 gở rối và xung clock cho lập trình ICSP * RB7: xuất / nhập số * PGD: mạch gở 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: 33.
* Dung lượng bộ nhớ chương trình 14KB.
* **Mạch ứng dụng của PIC 16F877A**



**Hình 2.4**. Mạch ứng dụng PIC 16F877A

* 1. **CẢM BIẾN NHIỆT ĐỘ IC DS18B20**
  2. **Giới thiệu chung về DS18B20**

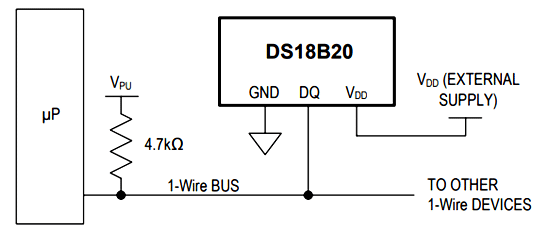
DS18B20 là một cảm biến nhiệt độ kỹ thuật số được sản xuất bởi Maxim Integrated. Nó sử dụng giao tiếp 1-Wire đơn giản và có nhiều ưu điểm như độ chính xác cao, độ phân giải cao, giá rẻ và dễ sử dụng.

* 1. **Sơ đồ chân của 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 dầu ra giá trị nhiệt dộ có thể đọc được bằng giao thức giao tiếp 1-wire |   **Bảng 2.2:** Chức năng của các chân DS18B20 | Cấu hình chân  **Hình 2.5.** Sơ đồ chân DS18B20 |

* **Thông số kỹ thuật:**
* Dải đo nhiệt độ: -55°C đến +125°C
* Độ chính xác: ±0,5°C (trong phạm vi -10°C đến +85°C)
* Độ phân giải: 12 bit
* 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 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

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.

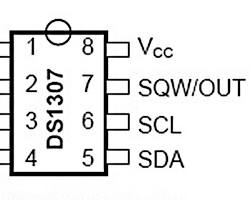
* 1. **Một số chú ý khi sử dụng DS18B20**
* DS18B20 có thể bị nhiễu bởi các nguồn điện áp cao
* DS18B20 không có khả năng chống nước
* Nên sử dụng điện trở kéo 4.4kΩ trên chân dữ liệu để đảm bảo tín hiệu ổn định
* Nên sử dụng tụ điện 100nF để lọc nhiệu nguồn
* Nên sử dụng dây dẫn ngắn để kết nối DS18B20 kết nối với vi điều khiển
* Nên đặt DS18B20 trong môi trường phù hợp với dải đo nhiệt độ của nó
  1. **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**

|  |  |  |
| --- | --- | --- |
| **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 |
| 5 | SDA | Chân dữ liệu I2C |
| 6 | RS | Chân chọn chế độ |
| 7 | WR | Chân ghi dữ liệu |
| 8 | CE | Chân chip enable |

****

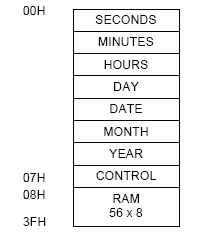
**Hình 2.7.** Sơ đồ chân của DS1307

**Bảng 2.3:** Chức năng của các chân DS1307

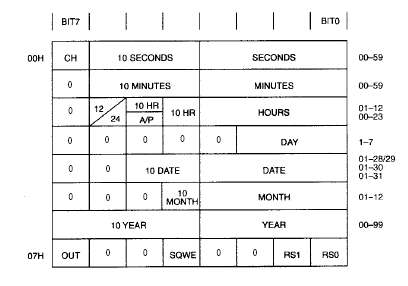
* **Thông số kỹ thuật:**
* Nguồn cung cấp: 5VDC.
* Khả năng lưu trữ 32K bit với EEPROM AT24C32.
* Sử dụng giao thức 2 dây I2C.
* Lưu trữ thông tin giờ phút giây AM/PM.
* Lịch lưu trữ chính xác lên đến năm 2100.
* Có pin đồng hồ lưu trữ thông tin.
* Có ngõ ra tần số 1Hz.
* Kích thước: 16 x 22 x 23mm.

### **2.3.3. Tổ chức thanh ghi trong DS1307**

Trong bộ nhớ của DS1307 có tất cả 64 thanh ghi địa chỉ từ 0 đến 63 và được bắt đầu từ 0x00 đến 0x3F nhưng trong đó chỉ có 8 thanh ghi đầu là thanh ghi thời gian thực nên tìm hiểu và đi sâu vào 8 thanh ghi ( chức năng và địa chỉ thanh ghi thời gian thực này). Nhìn vào bảng thanh ghi trong datasheet ta sẽ thấy như sau :



**Hình 2.8.** Tổ chức thanh ghi bên trong của DS1307



**Hình 2.9.** Tổ chức thanh ghi được mã hóa 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 3 bit 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)

+ **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 3 bit thấp dùng để mã hóa BCD chữ số hàng đơn vị và số hàng trục 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 nhà 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 trục lớn nhất là 2 nên do đó nó chỉ dùng 2 bit ( bit 4 và bit 5 ) để mã hóa BCD chữ số hàng trục của giờ. Nếu bit 6 =1 thì chế độ 12h được chọn nhưng do chữ số của hàng trục 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 trục 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 3 bit 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 1đến 31 ngày. Do đó thanh ghi này các bit được tổ chức khá là đơn giản. Nó dùng 4 bit 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 trục 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.

+ **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 4 bit 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 1 bit thứ 4 để mã hóa BCD ra chữ số hàng trục 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.
* Việc hàn DS1307 cần được thực hiện cẩn thận để tránh làm hỏng vi mạch.
  1. **HIỂN THỊ - LCD 16x02**

Có nhiều cách để hiện thị thời gian như màn hinh OLED, Led 7 đoạn, Ma trận Led đơn, LCD,…Các linh kiện điều 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.

**Ảnh có chứa văn bản, thiết bị điện tử, máy tính

Mô tả được tạo tự động**

**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 hổ trợ việc kết nối, đi dây điện.
* Có đèn led nền, có thể dùng biến trở hoặc PWM điều chình 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**

|  |  |  |
| --- | --- | --- |
| **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 |

**Bảng 2.4:** Sơ đồ chân LCD 16x2

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. ChânR/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

Ảnh có chứa văn bản, bàn

Mô tả được tạo tự động

* 1. **MỘT SỐ LÝ THUYẾT CƠ BẢN ĐƯỢC SỬ DỤNG**

**2.5.1. 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.**

* **Nguyên tắc hoạt động:**

I2C sử dụng hai dây tín hiệu để truyền dữ liệu:

* SDA (Serial Data Line): Dây này được sử dụng để truyền dữ liệu giữa các thiết bị.
* SCL (Serial Clock Line): Dây này được sử dụng để đồng bộ hóa việc truyền dữ liệu.
* **Đặc điểm:**
* Giao thức hai dây: I2C chỉ sử dụng hai dây tín hiệu để truyền dữ liệu, giúp giảm chi phí và độ phức tạp của hệ thống.
* Giao thức đồng bộ: I2C sử dụng tín hiệu xung nhịp để đồng bộ hóa việc truyền dữ liệu, giúp đảm bảo độ chính xác và tin cậy cao.
* Giao thức đa chủ: I2C cho phép nhiều thiết bị chủ (master) điều khiển một hoặc nhiều thiết bị phụ (slave).
* Có thể định địa chỉ: Mỗi thiết bị trên bus I2C được gán một địa chỉ duy nhất, cho phép thiết bị chủ truy cập từng thiết bị riêng lẻ.
* Có tính năng kiểm tra lỗi: I2C sử dụng mã CRC để kiểm tra lỗi dữ liệu, giúp đảm bảo tính toàn vẹn của dữ liệu được truyền.
* **Tốc độ truyền dữ liệu:**

I2C có thể hoạt động ở nhiều tốc độ truyền dữ liệu khác nhau, từ 100kbps đến 3.4Mbps. Tốc độ truyền dữ liệu được chọn dựa trên nhu cầu của ứng dụng cụ thể.

* **Ứng dụng:**
* I2C được sử dụng rộng rãi trong nhiều ứng dụng khác nhau, bao gồm:
* Điều khiển thiết bị: I2C được sử dụng để điều khiển các thiết bị điện tử như LED, LCD, ADC, DAC, v.v.
* Thu thập dữ liệu: I2C được sử dụng để thu thập dữ liệu từ các cảm biến, bộ nhớ, v.v.
* Mạng cảm biến: I2C được sử dụng để kết nối các cảm biến trong mạng cảm biến không dây.
* **Ưu điểm:**
* Dễ sử dụng: I2C là một giao thức đơn giản và dễ sử dụng.
* Chi phí thấp: I2C chỉ sử dụng hai dây tín hiệu, giúp giảm chi phí hệ thống.
* Độ tin cậy cao: I2C sử dụng mã CRC để kiểm tra lỗi dữ liệu, giúp đảm bảo tính toàn vẹn của dữ liệu được truyền.
* Tính linh hoạt: I2C có thể được sử dụng trong nhiều ứng dụng khác nhau.
* **Nhược điểm:**
* Tốc độ truyền dữ liệu thấp: So với các giao thức truyền thông khác như SPI, I2C có tốc độ truyền dữ liệu thấp hơn.
* Độ dài cáp ngắn: I2C không phù hợp cho việc truyền dữ liệu over long distances.

Tóm lại, I2C là một giao thức truyền thông nối tiếp đơn giản, dễ sử dụng và có chi phí thấp. Nó được sử dụng rộng rãi trong nhiều ứng dụng khác nhau.

**2.5.2. 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 8 bit 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.

* **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.

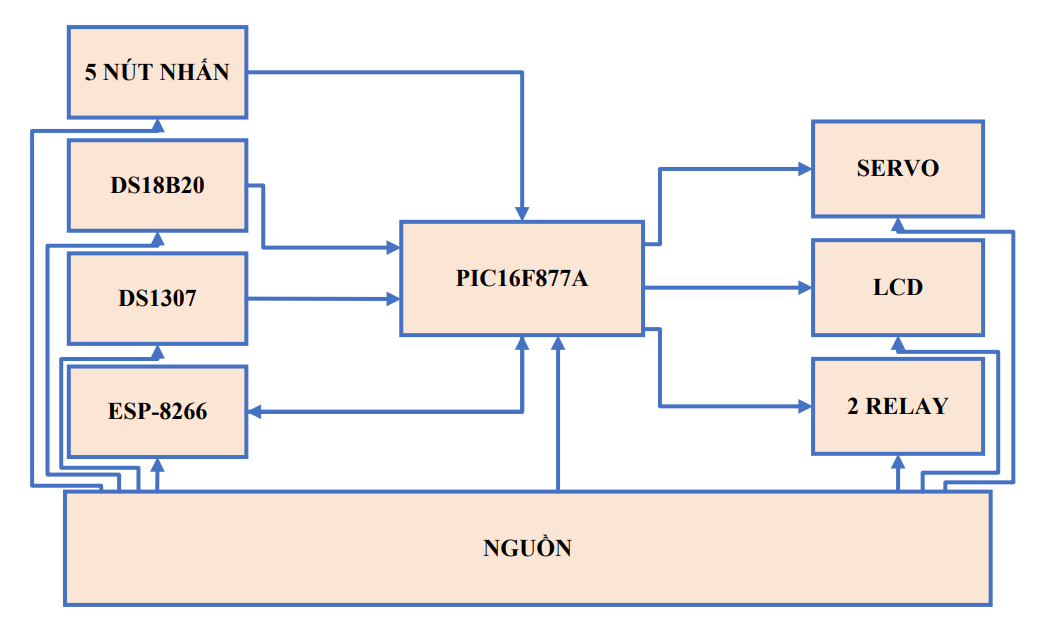
Tóm lại, UART là một giao thức truyền thông nối tiếp đơn giản và dễ sử dụng, được sử dụng rộng rãi trong nhiều ứng dụng khác nhau.

# **CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG**

**3.1. GIỚI THIỆU**

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

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

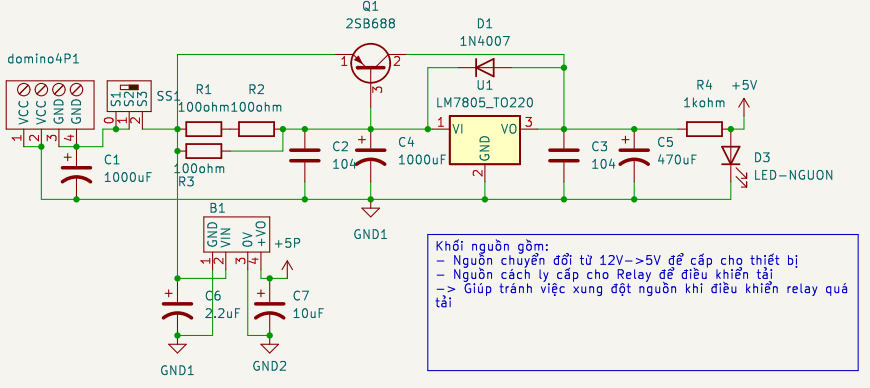


**Hình 3.1.** Sơ đồ khối hệ thống

Theo hình 3.1 ta thấy hệ thống gồm 9 khối. Trong đó được chia thành 4 khối chính là khối đọc dữ liệu đầu vào là 3 khối: khối 5 nút nhấn đọc giá trị để chỉnh nhiệt độ giới hạn và chỉnh thời gian cho cá ăn, khối ds18b20 là đọc giá trị nhiệt độ hồ cá, khối ds1307 đọc giá trị thời gian thực để xác định thời gian cho cá ăn; khối điều khiển hệ thống gồm 3 khối: khối servo để điều khiển thả thực ăn cho cá ăn, khối lcd để hiển thị thông tin về nhiệt độ, thời gian thưc, thông số cài giới hạn và thời gian cho cá ăn, khối 2 relay để điều khiển thiết bị làm lạnh và làm nóng hồ cá giúp hồ cá giữ ổn định nhiệt độ; khối xử lí trung tâm là khối pic16f877a để nhận các giá trị nhiệt độ và thời gian bên ngoài, dữ liệu từ khối esp-8266 và xử lí điều khiển ngõ ra là khối điều khiển hồ cá; khối giao tiếp mạng là khối esp-8266 để thực hiện việc lưu trự dữ liệu lên firebase để giám sát nhiệt độ hồ cá; khối nguồn là khối cung cấp nguồn ngõ vào cho tất cả các linh kiện của hệ thống.

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

#### 3.2.2.1 Khối nguồn



**Hình 3.1.** Khối nguồn

Table

|  |  |  |  |
| --- | --- | --- | --- |
| STT | TÊN | DÒNG VÀO TỐI ĐA | LOẠI NGUỒN |
| 1 | Pic16f877a | 160mA | Nguồn chung |
| 2 | Esp-8266 | 100mA | Nguồn chung |
| 3 | Lcd 16x02 | 1mA | Nguồn chung |
| 4 | 2 Relay | 126mA | Nguồn cách ly |
| 5 | Servo SG92R | 524mA | Nguồn chung |
| Dòng tổng lại nguồn chung: | | | 785(mA) |
| Dòng tổng lại nguồn cách ly: | | | 126(mA) |

- 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 Nm)(10.47 rad/s) = 2.62 watts.

Tính tổng dòng là I= P/V = (2.62watt)/(5V) = 0.524A = 524 mA.

- Tính dòng Pic16f877a dòng tối đa cho mỗi gpio là 10mA mà hệ thống sử dụng 16 gpio nên dòng tổng là 16\*10(mA) = 160mA.

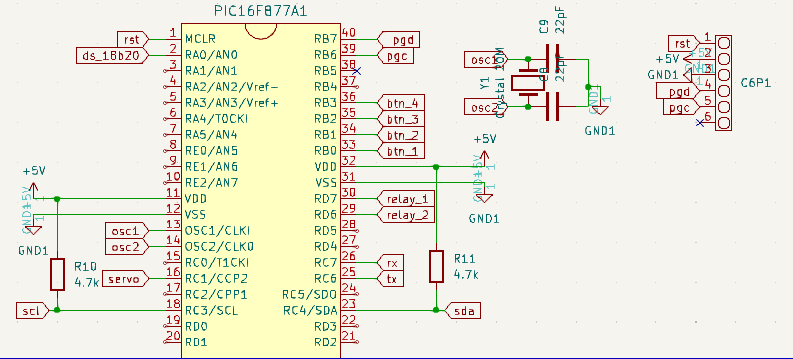
- Tính dòng Esp-8266 do sử dụng loại lua esp có cổng giao tiêp ch340 nên dòng nuôi cho Esp-8266 là khoảng 83mA nên làm tròn 100mA.

- Tính dòng Lcd 160x2 do lcd tiêu thụ năng lượng nên dòng tiêu thụ là khoảng 1mA

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

=> Do trong hệ thống có dùng relay để điều khiển thiết bị nên em thiết kế có nguồn cách ly để cấp cho khối 2 relay để tránh xung điện từ relay trả về cho pic16f877a gây hư cho con pic.

#### 3.2.2.2 Khối xử lí pic16f877a



**Hình 3.2.** Khối xử lí trung tâm

Khối 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 thõa mãn phù hợp mức logic => Rmin, thõa mãn rise time của tín hiệu => Rmax

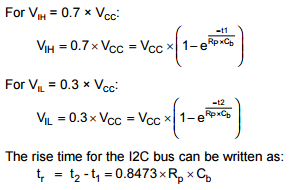
Công thức Rmin được tính

13

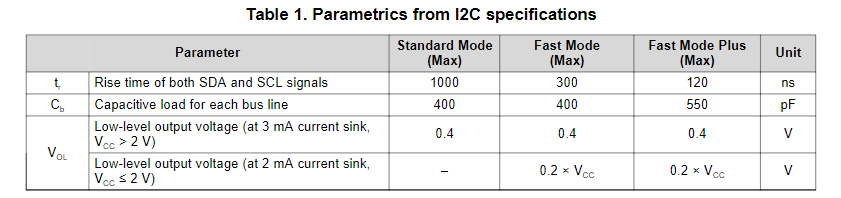
Công thức Rmax được tính

14

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



Từ đó tính Rmax là Rp(max) là

16 

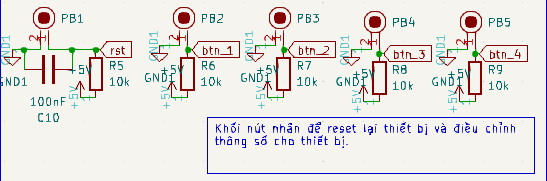
Ta chọn chế độ standard thì có rise time là 1000ns, cb là 400pF, vcc là 5v.

Suy ra Rp(max) = (1000\*10^-9)/(0.8473\*400\*10^-12) = 5.9kohm.

Tính Rp(min) = (5 – 0.4)/(3\*10^-3) = 1.5kohm.

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.

#### 3.2.2.3 Khối đọc dữ liệu đầu vào

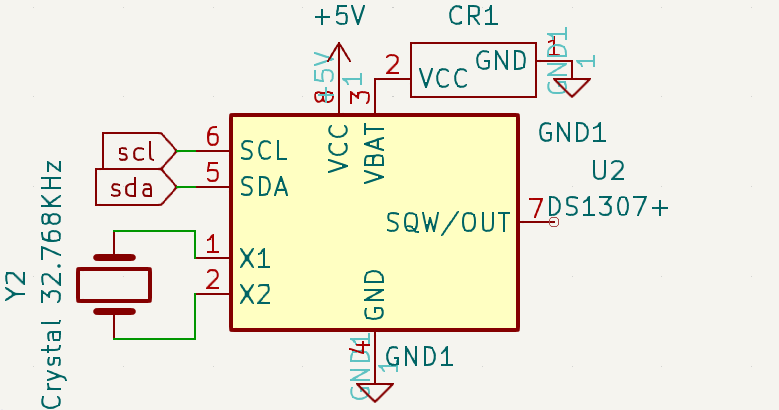


**Hình 3.3.** Khối 5 nút nhấn

- Khối 5 nút nhấn:

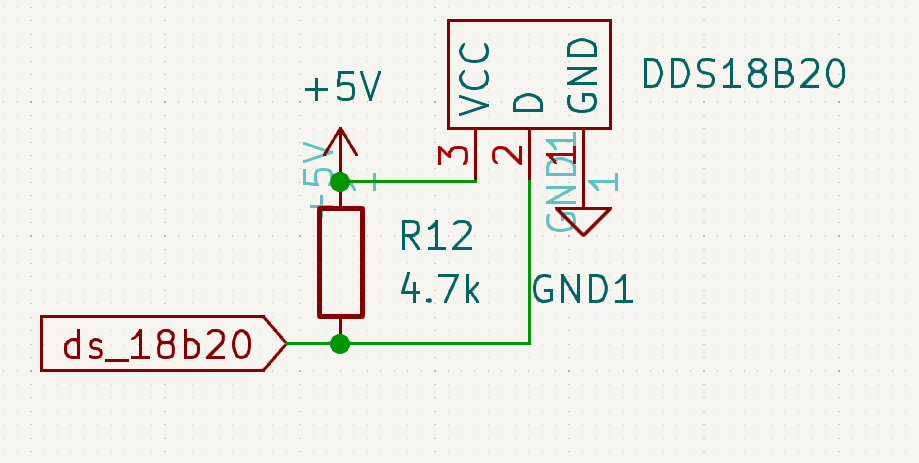
+ Nút reset có tụ 100nF gắn vào 2 đầu của nút nhấn giúp tự động reset khi cấp nguồn.

+ 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).



**Hình 3.4.** Khối ds1307

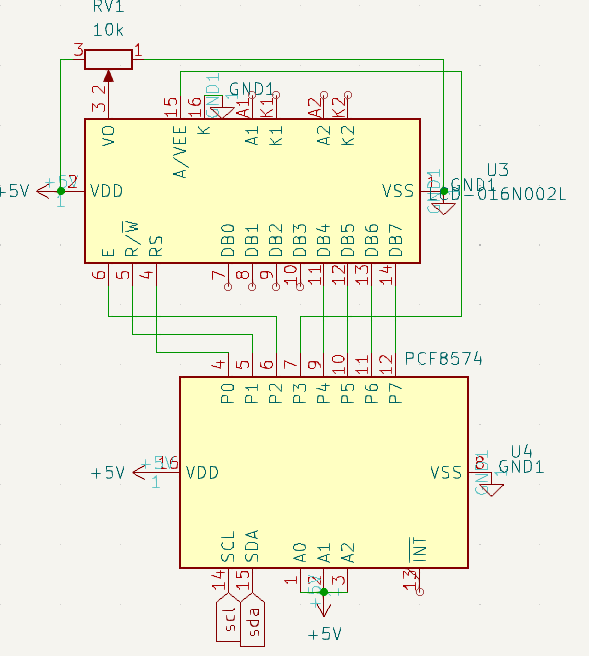
- Khối ds1307 sử dụng thạch anh 32.768Khz dựa vào datasheet của nhà cung cấp.



**Hình 3.4.** Khối ds18b20

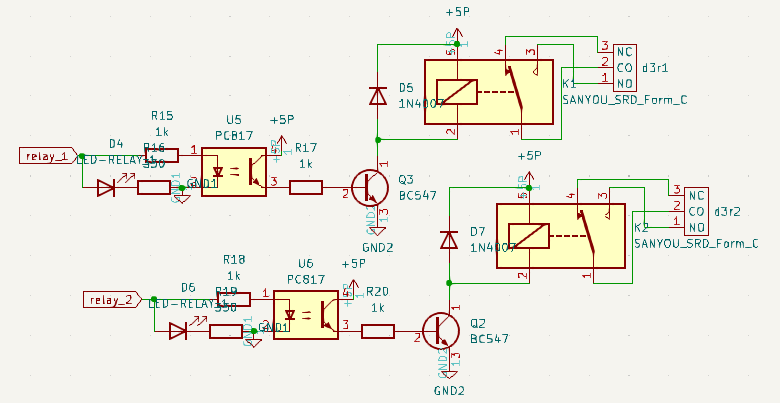
- Khối ds18b20 có sử dụng trở 4.7kohm dựa vào datasheet của nhà cung cấp.

#### 3.2.2.3 Khối điều khiển hệ thống



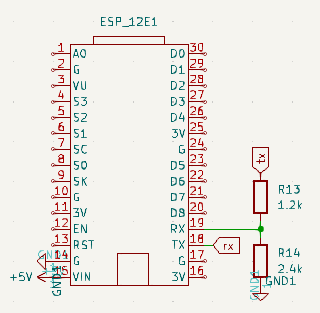
**Hình 3.5.** Khối lcd 16x02

- 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.

 **Hình 3.6.** 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.

#### 3.2.2.3 Khối esp-8266



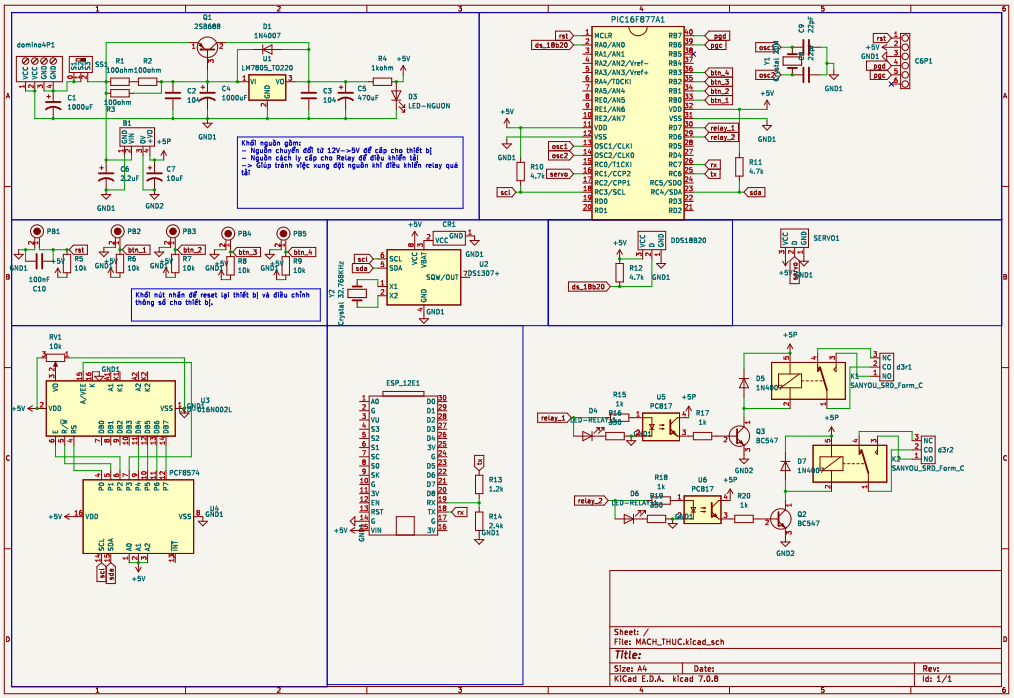
**Hình 3.7.** Khối esp-8266

- 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 sao:

Vo = 5(vdc)\*(2.4(kohm))/(2.4(kohm)+1.2(kohm)) = 3.33(vdc)

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

### **3.2.3 Sơ đồ nguyên lý của toàn mạch**



**Hình 3.8.** Sơ đồ nguyên lý toàn mạch

# **CHƯƠNG 4:** **THI CÔNG HỆ THỐNG**

**4.1. GIỚI THIỆU**

**4.2. THI CÔNG HỆ THỐNG**

### **4.2.1 Thi công bo mạch**

### **4.2.2 Lắp ráp và kiểm tra**

## 4.3. ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

### **4.3.1 Đóng gói bộ điều khiển**

### **4.3.2 Thi công mô hình**

## 4.4. LẬP TRÌNH HỆ THỐNG

### **4.4.1 Lưu đồ giải thuật**

### **4.4.2 Phần mềm lập trình cho vi điều khiển**

### **4.4.3 Phần mềm lập trình cho điện thoại, máy tính, …**

## 4.5. LẬP TRÌNH MÔ PHỎNG

### **4.5.1 Lưu đồ**

### **4.5.2 Xử lý tín hiệu hay hình ảnh**

## 4.6. VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

### **4.6.1 Viết tài liệu hướng dẫn sử dụng**

### **4.6.2 Quy trình thao tác**

# **CHƯƠNG 5: KẾT QUẢ, NHẬN XÉT VÀ ĐÁNH GIÁ**

## 5.1. CẢM BIẾN

## 5.2. KẾT QUẢ MÔ PHỎNG

### **5.2.1 Hình ảnh tiền xử lý**

### **5.4.2 Kết quả thống kê**

# **CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## 6.1. KẾT LUẬN

## 6.2. HƯỚNG PHÁT TRIỂN

# **TÀI LIỆU THAM KHẢO**

# **PHỤ LỤC**