

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



LOGIC DESIGN PROJECT

Đề tài: Thiết kế đồng hồ thời gian thực, có giao diện tương tác thông minh

GVHD: Phan Đình Thế Duy
Vũ Trọng Thiên
Lớp: L01
Nhóm thực hiện: Hồ Trọng Phúc - 2014159
Nguyễn Huy Hoàng - 2013231





Mục lục

| | |
|--|-----------|
| Chapter 1. Giới thiệu đề tài | 5 |
| 1 Giới thiệu chung về đồng hồ | 5 |
| 2 Sự xuất hiện và lịch sử phát triển của đồng hồ | 5 |
| 3 Các chức năng thường xuất hiện ở đồng hồ | 6 |
| Chapter 2. Ý tưởng đề tài | 7 |
| 1 Sự cần thiết của đồng hồ trong cuộc sống | 7 |
| 2 Tính tiện lợi và cần thiết của đề tài | 7 |
| 3 Các chức năng hiện thực trong đề tài | 8 |
| Chapter 3. Thiết bị sử dụng | 9 |
| 1 PIC18F4620 | 9 |
| 2 LCD 16x2 | 10 |
| 3 DS1307 | 10 |
| 4 Còi buzzer thụ động 5V | 11 |
| 5 Ma trận phím 4x4 | 11 |
| Chapter 4. Quá trình hiện thực | 13 |
| 1 Hiện thực chức năng | 13 |
| 1.1 Chức năng hiện thị thời gian và ngày, tháng, năm (<i>MODE 1</i>) . . . | 14 |
| 1.1.1 Giới thiệu sơ lược về DS1307 | 14 |
| 1.1.2 Quá trình hiện thực chức năng | 16 |
| 1.1.3 Hướng dẫn sử dụng chức năng | 19 |
| 1.2 Hiển thị menu (<i>MODE 2</i>) | 19 |
| 1.3 Chức năng hiển thị giờ thế giới (<i>MODE 3</i>) | 21 |

| | | |
|-------|--|----|
| 1.3.1 | Hướng dẫn sử dụng chức năng | 21 |
| 1.4 | Chức năng điều chỉnh thời gian và ngày, tháng, năm (<i>MODE 4</i>) . | 22 |
| 1.4.1 | Hướng dẫn sử dụng chức năng | 23 |
| 1.5 | Chức năng hiển thị số ngày và tuần hiện tại trong năm (<i>MODE 5</i>) | 24 |
| 1.5.1 | Hiển thị số ngày hiện tại trong năm | 24 |
| 1.5.2 | Hiển thị số tuần hiện tại trong năm | 25 |
| 1.5.3 | Hướng dẫn sử dụng chức năng | 25 |
| 1.6 | Chức năng bấm giờ (<i>MODE 6</i>) | 25 |
| 1.6.1 | Hướng dẫn sử dụng chức năng | 27 |
| 1.7 | Chức năng hẹn báo thức (<i>MODE 7</i>) | 27 |
| 1.7.1 | Hướng dẫn sử dụng chức năng | 30 |
| 1.8 | Chế độ hiển thị hoạt ảnh (<i>MODE 8</i>) | 31 |
| 1.8.1 | Cách để tạo ra một kí tự đặc biệt trên LCD | 31 |
| 1.8.2 | Hiển thị hoạt ảnh trên LCD | 33 |
| 1.8.3 | Hướng dẫn sử dụng chức năng | 34 |
| 1.9 | Chế độ tập trung (<i>MODE 9</i>) | 35 |
| 1.9.1 | Hướng dẫn sử dụng chức năng | 36 |
| 1.10 | Chức năng mở rộng ở <i>MODE 1</i> : Hiển thị thời gian theo giao diện khác | 36 |

Chapter 5. Kết quả hiện thực 39

Chapter 6. Kết luận 45

| | | |
|---|--|----|
| 1 | Đánh giá kết quả thu được | 45 |
| 2 | Những khó khăn gặp phải khi hiện thực đề tài | 45 |
| 3 | Hướng phát triển trong tương lai | 46 |

CHƯƠNG 1

Giới thiệu đề tài

1 Giới thiệu chung về đồng hồ

Đồng hồ là một dụng cụ thường dùng để đo khoảng thời gian dưới một ngày; khác với lịch, là một dụng cụ đo thời gian một ngày trở lên. Có những loại đồng hồ tân tiến và cấu trúc phức tạp đạt kỹ thuật đo thời gian rất chính xác. Ngoài những loại đồng hồ lớn đặt ở vị trí cố định, người ta cũng đã tạo ra loại đồng hồ nhỏ dễ dàng mang theo bên mình (gọi là đồng hồ đeo tay), ngoài chức năng cho biết giờ giấc còn là món hàng mỹ thuật có tính thời trang. Và từ lâu đồng hồ đã trở thành một dụng cụ không thể thiếu trong quá trình sinh hoạt hàng ngày của con người.

2 Sự xuất hiện và lịch sử phát triển của đồng hồ

Con người ghi nhận và đặt tên cho các khoảng thời gian là buổi, ngày, tháng, năm... nhưng những khoảng thời gian dưới một ngày thì trước khi có đồng hồ, việc đo đạc thiếu chính xác. Để nhất là ước tính vị trí của mặt trời trên bầu trời mà cho là sáng, trưa, chiều, tối. Ngắn hơn một buổi thì thuở trước người ta đốt nhang hay đèn cầy, mà tính lâu hay mau. Một loại bình đựng dùng cát hay nước cho chảy ra cũng là cách đo thời gian từ thời cổ đại văn minh Ai Cập và Trung Hoa.

Đồng hồ mặt trời Sundials và đồng hồ nước là những cỗ máy đo thời gian đầu tiên xuất hiện vào khoảng 3500 trước công nguyên. Cơ chế hoạt động dựa theo việc theo dõi mặt trăng, mặt trời, dòng thủy triều để ước lượng các mốc thời gian trong ngày. Đây là tiền đề cho sự phát triển của những chiếc đồng hồ về sau.

Chiếc đồng hồ được phát minh đầu tiên là đồng hồ bỏ túi, xuất hiện vào năm 1574 tại Thụy Sĩ. Được cấu tạo với một chiếc kim giờ, kim phút được thêm vào năm 1680

và khoảng một thập kỷ sau đó thì kim giây xuất hiện.

Loại đồng hồ như chúng ta biết đến ngày nay hình thành vào khoảng thế kỷ 17, xuất phát từ châu Âu. Đến thế kỷ 18 thì đồng hồ treo tường đã phổ biến là món hàng gia dụng của mọi giai cấp ở Âu châu. Kế tiếp là đồng hồ đeo tay và hiện tại đang phát triển và chiếm phần lớn trên thị trường là đồng hồ điện tử.

3 Các chức năng thường xuất hiện ở đồng hồ

Với các đồng hồ cơ truyền thống, các chức năng thường thấy đó là:

- Xem được thời gian: giờ, phút, giây
- Điều chỉnh thời gian
- Cài đặt báo thức

Kéo theo sự phát triển của công nghệ thì đồng hồ cũng có những cải tiến đáng kể, với sự ra đời của đồng hồ kỹ thuật số đã cho phép các chức năng của đồng hồ được mở rộng:

- Xem được cả thứ, ngày, tháng, năm
- Chức năng đếm thời gian
- Hiển thị thời gian với các giao diện bắt mắt, sáng tạo
- Chức năng giờ thế giới
- Hiển thị được nhiệt độ, độ ẩm

...

CHƯƠNG 2

Ý tưởng đề tài

1 Sự cần thiết của đồng hồ trong cuộc sống

Thế giới ngày càng phát triển thì việc kiểm soát và làm chủ thời gian là điều bắt buộc và đồng hồ chính là phương tiện hỗ trợ đắc lực. Việc biết được chính xác thời gian là việc vô cùng quan trọng trong mọi việc, tất cả thiết bị điện tử như GPS, vệ tinh, điện thoại,... đều có thể hoạt động được là nhờ có đồng hồ.

2 Tính tiện lợi và cần thiết của đề tài

Cuộc sống của con người ngày càng hiện đại và phát triển thì nhu cầu của con người ngày càng tăng lên. Đồng hồ lúc này không chỉ là dụng cụ dùng để xem thời gian trong ngày mà còn yêu cầu phải có các chức năng mở rộng đi kèm (xem được cả thứ, ngày, tháng, năm, xem được nhiệt độ, độ ẩm, các chức năng bấm giờ, xem giờ thế giới,...).

Việc tích hợp nhiều chức năng trong một chiếc đồng hồ giúp tăng tính tiện lợi, nâng cao trải nghiệm của người dùng, tiết kiệm được thời gian, thay vì phải sử dụng nhiều công cụ, thiết bị để thực hiện các chức năng khác nhau thì bây giờ chỉ cần một thiết bị thì người dùng có thể sử dụng được tất cả các chức năng đó.

Hiện thực các chức năng mới trên một chiếc đồng hồ giúp trải nghiệm sử dụng của người dùng trở nên hiện đại, thú vị hơn

3 Các chức năng hiện thực trong đề tài

Các chức năng của đồng hồ kỹ thuật số được thực hiện trong đề tài này:

- Chức năng điều chỉnh thời gian
- Hiển thị thời gian theo các giao diện khác nhau.
- Chức năng giờ thế giới
- Chức năng hiện số ngày và số tuần trong năm
- Chức năng báo thức
- Hoạt ảnh (Animation): Man Walking, Wave
- Chức năng đếm thời gian
- Chức năng tập trung

CHƯƠNG 3

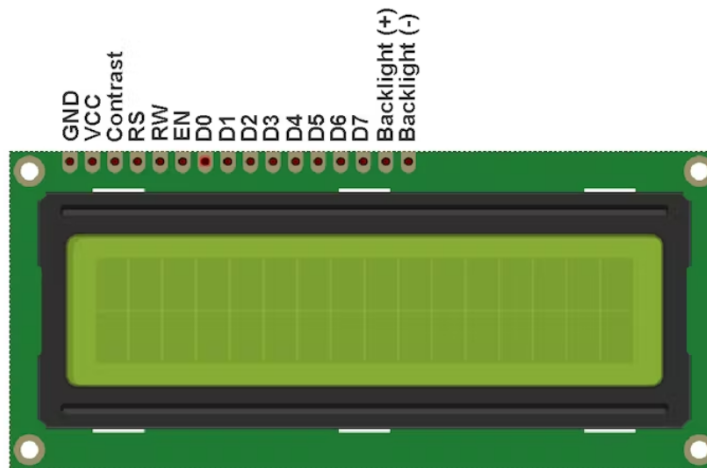
Thiết bị sử dụng

1 PIC18F4620



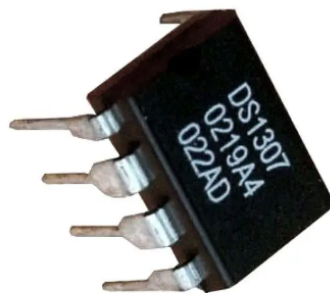
Hình 3.1: PIC18F4620

2 LCD 16x2



Hình 3.2: LCD 16x2

3 DS1307



Hình 3.3: DS1307

4 Còi buzzer thụ động 5V



Hình 3.4: Buzzer

5 Ma trận phím 4x4

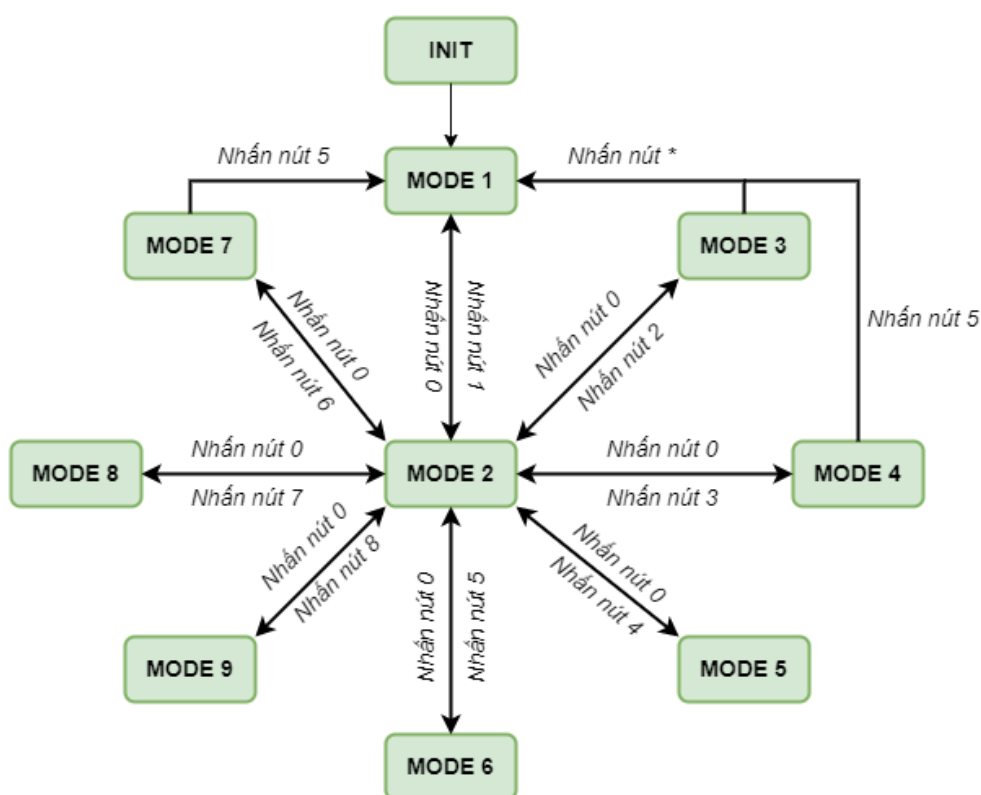


CHƯƠNG 4

Quá trình hiện thực

1 Hiện thực chức năng

Hiện thực các chức năng của đồng hồ theo một máy trạng thái như sau:



Hình 4.1: Máy trạng thái

Chi tiết về các trạng thái:

| Trạng thái | Mô tả trạng thái |
|------------|--|
| MODE 1 | Hiển thị thời gian và ngày, tháng, năm |
| MODE 2 | Hiển thị menu |
| MODE 3 | Chế độ giờ thế giới |
| MODE 4 | Chức năng điều chỉnh thời gian và ngày, tháng, năm |
| MODE 5 | Chức năng hiển thị số ngày và số tuần hiện tại trong năm |
| MODE 6 | Chức năng bấm giờ |
| MODE 7 | Chức năng hẹn giờ báo thức |
| MODE 8 | Chế độ hiển thị hoạt ảnh |
| MODE 9 | Chế độ tập trung |

1.1 Chức năng hiển thị thời gian và ngày, tháng, năm (MODE 1)

Ở chức năng này, ta sẽ tận dụng chức năng của IC DS1307 hay còn gọi là IC thời gian thực.

1.1.1 Giới thiệu sơ lược về DS1307

IC DS1307 có chức năng cung cấp thông tin thời gian hiện tại (thời gian thực): giờ, phút, giây, thứ, ngày tháng, năm một cách chính xác ngay cả khi thiết bị đã bị tắt (ngắt điện ngoài). Giao tiếp với vi điều khiển thông qua chuẩn I2C, và đóng vai trò là slave khi kết nối đến bus I2C này. Có thể đếm thời gian theo định dạng 24 giờ hoặc 12 giờ với chỉ thị AM/PM. Ngoài ra bên trong chip có bộ dò phát hiện mất nguồn và tự động chuyển sang sử dụng nguồn pin dự phòng. Một số tính năng nổi bật của IC RTC DS1307.

Các thanh ghi lưu trữ thời gian của DS1307:

- Định dạng theo kiểu BCD (Binary Coded Decimal value).
- Có tổng cộng 8 thanh ghi để đọc thời gian, ngày, tháng, năm và điều khiển.
- Sau khi đặt giá trị của các thanh ghi này, chúng sẽ tự cập nhật và ta có thể đọc các thanh ghi này để nhận các giá trị cập nhật.

| ADDRESS | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | FUNCTION | RANGE |
|---------|---------|------------|---------|----------|---------|------|------|------|------------|-------------------------|
| 00H | CH | 10 Seconds | | | Seconds | | | | Seconds | 00–59 |
| 01H | 0 | 10 Minutes | | | Minutes | | | | Minutes | 00–59 |
| 02H | 0 | 12 | 10 Hour | 10 Hour | Hours | | | | Hours | 1–12 +AM/PM 00–23 |
| | | 24 | PM/AM | | | | | | | |
| 03H | 0 | 0 | 0 | 0 | DAY | | | | Day | 01–07 |
| 04H | 0 | 0 | 10 Date | | Date | | | | | Date |
| 05H | 0 | 0 | 0 | 10 Month | Month | | | | Month | 01–12 |
| 06H | 10 Year | | | | Year | | | | Year | 00–99 |
| 07H | OUT | 0 | 0 | SQWE | 0 | 0 | RS1 | RS0 | Control | — |
| 08H-3FH | | | | | | | | | RAM 56 x 8 | 00H–FFH |

Địa chỉ 00H->02H: Thanh ghi lưu thông số về thời gian:

- Địa chỉ - 00H: Đọc/ghi giây:
 - Bit-7 là bit CH, là bit bật / tắt bộ dao động tinh thể, khi nó bằng 0, bộ dao động tinh thể được kích hoạt.
 - Các bit khác được sử dụng để đọc / ghi giây.
 - * Bit-4 đến 6 lưu trữ chữ số BCD phía trên của giây (giá trị từ 0 đến 5).
 - * Bit-0 đến 3 lưu trữ chữ số BCD thấp hơn của giây (giá trị từ 0 đến 9).
 - * Giá trị của giây bắt đầu từ 00 và kết thúc ở 59.
- Địa chỉ - 01H: Đọc/ghi phút:
 - Bit 4 đến 6 lưu hàng chục.
 - Bit 0 đến 3 lưu hàng đơn vị.
- Địa chỉ - 02H: Đọc/ghi giờ: Định dạng 24 giờ: Bit-6 = 0.
 - Bit-4 và Bit-5 được biểu thị là 10 Giờ, lưu trữ chữ số cao hơn của giá trị giờ (từ 0 đến 2).
 - Bit-0 đến Bit- 3 lưu trữ giá trị của chữ số thấp hơn của giờ (giá trị từ 0 đến 9).

Địa chỉ - 03H->06H: Thanh ghi lưu thông số ngày, tháng, năm:

- Địa chỉ - 03H: Bit 0 đến 2 để đọc ghi giá trị ngày (từ 01 đến 07).
- Địa chỉ - 04H:
 - Đọc giá trị ngày từ 01 – 31.
 - Bit 4, 5 để lưu hàng chục (0 đến 3), bit 0 đến 3 để lưu hàng đơn vị (0 đến 9).
- Địa chỉ - 05H:
 - Đọc giá trị tháng từ 01 – 12.

- Bit 4 lưu giá trị hàng chục (0 hoặc 1), bit 0 đến 3 lưu hàng đơn vị (0 đến 9).
- Địa chỉ - 06H:
 - Đọc ghi giá trị năm từ 00 – 99.
 - Bit 7 đến 4 lưu giá trị hàng chục (0 đến 9), bit 0 đến 3 lưu hàng đơn vị (0 đến 9).

1.1.2 Quá trình hiện thực chức năng

Ý tưởng cho việc thực hiện chức năng là chúng ta sẽ tiến hành lấy thông tin thời gian thực từ DS1307 thông qua giao tiếp theo chuẩn IC và tiến hành hiển thị các thông tin này lên LCD.

Hiện thực hàm lấy thông tin từ DS1307 theo chuẩn I2C

```
1 void Read_time(){
2     second = read_ds1307(ADDRESS_SECOND);
3     minute = read_ds1307(ADDRESS_MINUTE);
4     hour = read_ds1307(ADDRESS_HOUR);
5     day = read_ds1307(ADDRESS_DAY);
6     date = read_ds1307(ADDRESS_DATE);
7     month = read_ds1307(ADDRESS_MONTH);
8     if (year == 99 && read_ds1307(ADDRESS_YEAR) == 0){
9         sub_year++;
10    }
11    year = read_ds1307(ADDRESS_YEAR);
12 }
```

Program 4.1: Read_time function

Với các biến second, minute, hour, day, date, month, sub_year, year tương ứng cho giá trị lần lượt là giây, phút, giờ, thứ trong tuần, ngày, tháng, 2 số đầu của năm và 2 số cuối của năm. Do DS1307 chỉ trả về kết quả năm của 2 số cuối của năm nên ta cần thêm một biến sub_year để lưu trữ hai số đầu, biến sub_year sẽ tăng lên 1 đơn vị khi biến year đếm lên 99 và trở về 0.

Các giá trị ADDRESS_SECOND, ADDRESS_MINUTE, ADDRESS_HOUR, ADDRESS_DAY, ADDRESS_DATE, ADDRESS_MONTH, ADDRESS_YEAR tương ứng các địa chỉ các thanh ghi chứa các giá trị tương ứng của DS1307 thông qua giao tiếp I2C

```
1 #define ADDRESS_SECOND 0x00
2 #define ADDRESS_MINUTE 0x01
3 #define ADDRESS_HOUR 0x02
4 #define ADDRESS_DAY 0x03
5 #define ADDRESS_DATE 0x04
6 #define ADDRESS_MONTH 0x05
7 #define ADDRESS_YEAR 0x06
```


Hiện thực hàm cài đặt thông tin thời gian và truyền các giá trị này tới DS1307 theo chuẩn I2C

```
1 void Set_Time()  
2 {  
3     write_ds1307(ADDRESS_SECOND, second);  
4     write_ds1307(ADDRESS_MINUTE, minute);  
5     write_ds1307(ADDRESS_HOUR, hour);  
6     write_ds1307(ADDRESS_DAY, day);  
7     write_ds1307(ADDRESS_DATE, date);  
8     write_ds1307(ADDRESS_MONTH, month);  
9     write_ds1307(ADDRESS_YEAR, year);  
10 }
```

Program 4.2: Set_time function

Với hàm này, ta sẽ tạo các biến second, minute, hour, day, date, month, year với các giá trị mong muốn và đồng thời dùng giao tiếp I2C truyền các giá trị này tới DS1307 để DS1307 bắt đầu đếm thời gian từ thời gian đã thiết lập sẵn. Địa chỉ để truyền các giá trị này cũng sẽ tương tự như các địa chỉ đã đề cập ở trên

Hiện thực hàm hiển thị các thông tin thời gian lên LCD

- Hiển thị thời gian giờ, phút, giây:

```
1 void Display_time(){  
2     if (format_12h == 1 && format_24h == 1){  
3         LcdPrintStringS(0,8,"  ");  
4     }  
5     if (format_12h == 1 && format_24h == 0){  
6         if (hour > 12){  
7             hour = hour -12;  
8             isPM_or_AM = 1;  
9             LcdPrintStringS(0,8," PM");  
10        }  
11        else if (hour == 12){  
12            isPM_or_AM = 1;  
13            LcdPrintStringS(0,8," PM");  
14        }  
15        else{  
16            if (hour ==0 ) hour = 12;  
17            isPM_or_AM = 0;  
18            LcdPrintStringS(0,8," AM");  
19        }  
20    }  
21    LcdPrintNumS(0,0, hour/10);  
22    LcdPrintNumS(0,1, hour%10);  
23    LcdPrintStringS(0,2, ":");
```

```
24     LcdPrintNumS(0,3,minute/10);
25     LcdPrintNumS(0,4,minute%10);
26     LcdPrintStringS(0,5,":");
27     LcdPrintNumS(0,6,second/10);
28     LcdPrintNumS(0,7,second%10);
29
30     LcdPrintStringS(0,11,"UTC");
31     if(world_time[index_worldtime] < 0){
32         LcdPrintStringS(0,14,"-");
33         LcdPrintNumS(0,15,-world_time[index_worldtime
34     ]);
35     }
36     else{
37         LcdPrintStringS(0,14,"+");
38         LcdPrintNumS(0,15,world_time[index_worldtime
39     ]);
40     }
41 }
```

Đối với phần hiển thị này ta sẽ dùng các hàm đã hiển thị trong "lcd.c" là `LcdPrintNums` để gán một số nguyên vào buffer tại vị trí được khai báo ở 2 tham số trước trong hàm và `LcdPrintStringS` để lưu từng kí tự trong chuỗi string vào buffer. Để in giá trị giờ ta thực hiện lệnh `LcdPrintNumS(0,0,hour/10)` để in tại vị trí (0;0) chữ số đầu tiên trong số có 2 chữ số trả về từ việc đọc giá trị giờ từ DS1307 đồng thời lệnh `LcdPrintNumS(0,1,hour%10)` để in chữ số cuối của biến `hour`. Để tách biệt các thông số giờ, phút, giây với nhau ta thực hiện lệnh in một kí tự ":" ở giữa các thông số bằng câu lệnh `LcdPrintStringS(0,2,":")`. Và tương tự cho việc hiển thị thông số phút và giây.

Để hiển thị chức năng chuyển từ chế độ 24h sang chế độ hiển thị 12h (AM hoặc PM), ta sẽ dựa trên các nguyên tắc về chuyển từ 24h sang 12h kèm theo "AM", "PM" để hiển thị và các trường hợp đặc biệt 0h sẽ là 12 AM và 12h sẽ là 12 PM

- **Hiển thị ngày, tháng, năm:**

```
1 void Display_Date(){
2     LcdPrintStringS(1,0,day_week[day-1]);
3     LcdPrintStringS(1,3,", ");
4     LcdPrintNumS(1,4,date/10);
5     LcdPrintNumS(1,5,date%10);
6     LcdPrintStringS(1,6,"/");
7     LcdPrintNumS(1,7,month/10);
8     LcdPrintNumS(1,8,month%10);
9     LcdPrintStringS(1,9,"/");
10    LcdPrintNumS(1,10,sub_year/10);
11    LcdPrintNumS(1,11,sub_year%10);
```

```
12     LcdPrintNumS(1,12,year/10);  
13     LcdPrintNumS(1,13,year%10);  
14 }  
15
```

Việc hiện thực cũng sẽ tương tự với việc hiển thị giờ, phút, giây ở trên. Tuy nhiên có điểm khác là do giá trị thứ trong tuần trả về từ DS1307 chỉ là các con số từ 1-7 do đó để hiển thị thứ ta cần đưa tên của các thứ trong tuần vào một mảng

```
1 char*day_week[7] = {"Sun","Mon","Tue","Wed","Thus","Fri",  
    "Sat"};
```

Các giá trị 1-7 trừ đi 1 sẽ tương ứng với vị trí của thứ hiện tại trong mảng. Để tách biệt các thông số ta sẽ sử dụng ký tự "/"

1.1.3 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này:

| Nút nhấn | Chức năng |
|------------|--|
| Nút nhấn 1 | Chuyển sang chế độ hiển thị giờ theo format 12h |
| Nút nhấn 2 | Tắt format 12h và chuyển sang format 24h |
| Nút nhấn 3 | Chuyển sang chế độ chỉ hiển thị giờ và phút |
| Nút nhấn * | Tắt chế độ hiển thị giờ và phút, trở về chế độ hiển thị thông thường |
| Nút nhấn 0 | Bật menu |

1.2 Hiển thị menu (MODE 2)

Đối với một đồng hồ có nhiều chức năng, ta cần phải có một giao diện để hiển thị và hướng dẫn người dùng sử dụng các chức năng của đồng hồ. Ý tưởng để xây dựng việc hiển thị menu chính là hiển thị dòng chữ "Mời chọn chế độ" ở dòng thứ nhất của LCD và tiếp theo hiển thị một dòng chữ bao gồm: tên các chức năng và nhấn nút nào để chuyển từ menu sang chức năng đó. Tuy nhiên, do kích thước giới hạn của LCD ta không thể hiển thị dòng chữ như vậy cùng một lúc. Vì vậy, ta sẽ hiện thực một dòng chữ có khả năng chạy từ phải sang phải để giúp hiển thị hết các nội dung mong muốn.

```
1 void display_menu(){  
2     LcdPrintStringS(0,0,"Moi chon che do");  
3     if (flag_timer1 == 1){  
4         SetTimer1_ms(30);  
5         LcdPrintCharS(x,y,menu[k]);  
6         y++;  
7         k++;  
8         if (y >= 16){
```

```
9
10     count3++;
11     y = 0;
12     k = count3;
13     DisplayLcdScreen();
14     LcdClearS();
15 }
16 if (menu[k] == '\0'){
17     y = 0;
18     LcdClearS();
19     LcdPrintStringS(0,0,"Moi chon che do");
20     count3 = 0;
21     k = count3;
22 }
23 }
24 }
```

Program 4.3: Hàm Display_menu

Thực hiện lưu từng kí tự trong mảng menu[] vào buffer mỗi 12ms và khi số lượng các kí tự đạt tới 16 thì ta tiến hành in 16 kí tự này ra LCD bằng lệnh DisplayLcdScreen(). Biến count3 ban đầu được khởi tạo với giá trị bằng 0 và tăng lên một đơn vị khi số lượng kí tự đạt 16, mục đích là để tạo một hiệu ứng dịch chuyển từ phải sang trái (vị trí để lấy các kí tự trong mảng menu[] sẽ tăng 1 đơn vị). Khi đã dịch chuyển hết chuỗi lưu trong mảng menu thì ta tiến hành reset các giá trị phục vụ cho việc in các kí tự về giá trị ban đầu.

Hướng dẫn sử dụng:

Ở menu, ta có thể sử dụng các nút nhấn được quy định để mở các chức năng trên đồng hồ. Chi tiết như sau:

| Nút nhấn | Chức năng |
|------------|--|
| Nút nhấn 1 | Chế độ hiển thị thời gian, thứ ngày tháng năm |
| Nút nhấn 2 | Chế độ giờ thế giới |
| Nút nhấn 3 | Chế độ điều chỉnh thời gian, thứ ngày tháng năm |
| Nút nhấn 4 | Chế độ hiển thị số ngày và tuần hiện tại trong năm |
| Nút nhấn 5 | Chế độ bấm giờ |
| Nút nhấn 6 | Chế độ cài đặt báo thức |
| Nút nhấn 7 | Chế độ hiển thị hoạt ảnh |
| Nút nhấn 8 | Chế độ tập trung |

1.3 Chức năng hiển thị giờ thế giới (MODE 3)

Ý tưởng của chức năng là ta sẽ điều chỉnh thông số giờ của đồng hồ hiện tại có thể cộng thêm hoặc trừ đi phụ thuộc vào múi giờ mà người dùng chọn. Các thông số về múi giờ sẽ được lưu trữ trong một mảng

Các hàm để hiện thực chức năng này bao gồm: `display_worldtime` trong file `clock.c` và các mảng lưu trữ thông tin về múi giờ được lưu vào file `global.c`

Hướng dẫn sử dụng: Đầu tiên nhấn vào nút 0 để chuyển sang chế độ hiển thị menu trên LCD và khi đang ở menu nhấn nút 1 để chuyển sang chế độ giờ thế giới. Ở chế độ giờ thế giới thì sẽ hiển thị trên LCD danh sách múi giờ ở nơi trên nhiều quốc gia: Việt Nam, London, China, Moscow, Alaska,.. Nhấn vào 4 để con trỏ đi lên và 6 để con trỏ đi xuống, lúc này nếu muốn chọn múi giờ ở nơi nào thì nhấn nút 5 để chuyển và nhấn vào nút "." để thoát ra ngoài màn hình và xem kết quả.

```
1 void display_worldtime(){
2     int n;
3     lcd_write_cmd(64);
4     for (n = 0; n<=7; n++) lcd_write_data(arrow[n]);
5
6     if(index_worldtime == 0){
7         LcdPrintStringS(0,0,"World time");
8         LcdPrintStringS(1,0,countries[index_worldtime]);
9         LcdPrintCharS(1,15,0);
10
11     }
12     else if(index_worldtime+1 < num_countries){
13         LcdPrintStringS(0,0,countries[index_worldtime]);
14         LcdPrintStringS(1,0,countries[index_worldtime+1]);
15         LcdPrintCharS(0,15,0);
16     }
17     else{
18         LcdPrintStringS(0,0,countries[index_worldtime-1]);
19         LcdPrintStringS(1,0,countries[index_worldtime]);
20         LcdPrintCharS(1,15,0);
21     }
22 }
```

Program 4.4: Hàm `display_worldtime`

1.3.1 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|--------------|-------------------------------------|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 4,6 | Di chuyển giữa danh mục các múi giờ |
| Nút nhấn 5 | Chọn múi giờ |
| Nút nhấn * | Trở về chế độ hiển thị thông thường |

1.4 Chức năng điều chỉnh thời gian và ngày, tháng, năm (MODE 4)

Ý tưởng để hiện thực chức năng: ta tiến hành hiển thị các giá trị với bố cục giống như ở chế độ hiển thị thông thường. Ta cần phải sử dụng một nút nhấn để di chuyển và lựa chọn thông nào cần thiết để điều chỉnh, cần phải có một nút nhấn để tăng giá trị và một nút nhấn để giảm giá trị ở thông số đã chọn. Sau đó cần phải có nút nhấn để lưu giá trị hiện tại đã được điều chỉnh và chuyển về chế độ hiển thị thông thường và bắt đầu chạy bằng các thông số đã được điều chỉnh và lưu lại. Tuy nhiên để nhận biết được thông số nào đang được chọn ta cần phải tạo ra hiệu ứng chớp tắt tại vị trí thông số đó để giúp người dùng nhận biết được. Điều này sẽ được thực hiện thông qua cờ `blink_flag`, cờ này sẽ luôn phiên thay đổi giữa 0 và 1 trong thời gian 50ms để tạo hiệu ứng chớp tắt.

Các hàm để hiện thực chức năng này bao gồm: `increase`, `decrease`, `display_inc_and_dec_mode` được lưu trữ trong file `clock.c`

Hướng dẫn sử dụng: Ở chế độ này, Nhấn nút 1 để chọn vị trí mà người dùng muốn điều chỉnh thời gian. Khi đang chọn vị trí nào thì vị trí đó sẽ nháy. Nhấn nút 4 để giảm và 6 để tăng giá trị tại vị trí đang chọn.

```
1 void Display_inc_and_dec_mode(){
2     LcdPrintNumS(0,0, hour/10);
3     LcdPrintNumS(0,1, hour%10);
4     if(index_set == 1 && blink_flag == 1){
5         LcdPrintStringS(0,0, " ");
6         LcdPrintStringS(0,1, " ");
7         //blink_flag = 0;
8     }
9     LcdPrintStringS(0,2, ":");
10    LcdPrintNumS(0,3, minute/10);
11    LcdPrintNumS(0,4, minute%10);
12    if(index_set == 2 && blink_flag == 1){
13        LcdPrintStringS(0,3, " ");
14        LcdPrintStringS(0,4, " ");
15        // blink_flag = 0;
16    }
17    LcdPrintStringS(0,5, ":");
18    LcdPrintNumS(0,6, second/10);
19    LcdPrintNumS(0,7, second%10);
```

```
20     if(index_set == 3 && blink_flag == 1){
21         LcdPrintStringS(0,6," ");
22         LcdPrintStringS(0,7," ");
23         //blink_flag = 0;
24     }
25     LcdPrintStringS(1,0,day_week[day-1]);
26     if(index_set == 4 && blink_flag == 1){
27         LcdPrintStringS(1,0," ");
28         // blink_flag = 0;
29     }
30     LcdPrintStringS(1,3,", ");
31     LcdPrintNumS(1,5,date/10);
32     LcdPrintNumS(1,6,date%10);
33     if(index_set == 5 && blink_flag == 1){
34         LcdPrintStringS(1,5," ");
35         //blink_flag = 0;
36     }
37     LcdPrintStringS(1,7,"/");
38     LcdPrintNumS(1,8,month/10);
39     LcdPrintNumS(1,9,month%10);
40     if(index_set == 6 && blink_flag == 1){
41         LcdPrintStringS(1,8," ");
42         // blink_flag = 0;
43     }
44     LcdPrintStringS(1,10,"/");
45     LcdPrintNumS(1,11,sub_year/10);
46     LcdPrintNumS(1,12,sub_year%10);
47     LcdPrintNumS(1,13,year/10);
48     LcdPrintNumS(1,14,year%10);
49     if(index_set == 7 && blink_flag == 1){
50         LcdPrintStringS(1,11," ");
51         // blink_flag = 0;
52     }
53 }
```

Program 4.5: Hàm Display_inc_and_dec_mode

1.4.1 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|------------|----------------------------|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 4 | Nút nhấn để giảm giá trị |
| Nút nhấn 6 | Nút nhấn để tăng giá trị |
| Nút nhấn 5 | Lưu quá trình chỉnh sửa |
| Nút nhấn 1 | Chọn thông số để chỉnh sửa |

1.5 Chức năng hiển thị số ngày và tuần hiện tại trong năm (MODE 5)

Ở chế độ này sẽ hiện lên LCD số ngày và số tuần hiện tại trong năm.

1.5.1 Hiển thị số ngày hiện tại trong năm

```
1 void Day_Week_Number() {
2     int number_of_days;
3     ...
4     int days_of_month[12] = {0, 31, 59, 90, 120, 151, 181,
5     212, 243, 273, 304, 334};
6     int leap_day = 0;
7     if (((sub_year*100+year) %4 == 0) && ((sub_year*100+
8     year) % 100 != 0)) || ((sub_year*100+year) % 400 == 0))
9     ){
10        leap_day = 1;
11    }
12    if (leap_day == 1){
13        if (month <= 2){
14            number_of_days = days_of_month[(month-1)] + date;
15        }
16        else{
17            number_of_days = days_of_month[(month-1)] + date
18            + leap_day;
19        }
20    }
21    else{
22        number_of_days = days_of_month[(month-1)] + date;
23    }
24    ...
25    LcdClearS();
26    LcdPrintStringS(0,0,"Ngày:");
27    LcdPrintNumS(0,6,number_of_days);
28    ...
29    DisplayLcdScreen();
30 }
```


26 }

Ý tưởng của phần này là ta sẽ tạo một mảng có chứa số ngày bắt đầu của từng tháng theo thứ tự trong năm không nhuận. Nếu năm hiện tại không nhuận, ta sẽ hiển thị số ngày hiện tại trong năm sẽ bằng số ngày bắt đầu của tháng hiện tại cộng cho số ngày hiện tại trong tháng. Trong trường hợp năm hiện tại là năm nhuận, nếu tháng hiện tại nhỏ hơn hoặc bằng tháng 2, ta sẽ tính số ngày hiện tại trong năm như trong trường hợp trên, còn nếu tháng hiện tại lớn hơn tháng 2, ta sẽ tính số ngày hiện tại trong năm như các trường hợp ở trên đồng thời cộng thêm 1 (ngày nhuận ở tháng 2)

1.5.2 Hiển thị số tuần hiện tại trong năm

```
1 void Day_Week_Number() {  
2     ...  
3     int number_of_week;  
4     ...  
5     if (day == 1) {  
6         number_of_week = (number_of_days - 7 + 10) / 7;  
7     }  
8     else {  
9         number_of_week = (number_of_days - day + 10) / 7;  
10    }  
11    LcdClearS();  
12    ...  
13    LcdPrintStringS(1, 0, "Tuan : ");  
14    LcdPrintNumS(1, 6, number_of_week);  
15    DisplayLcdScreen();  
16 }
```

Ta sẽ sử dụng thông tin về số ngày hiện tại trong năm vừa tính được ở trên kết hợp với thông số về thứ của ngày hiện tại để tính ra số tuần hiện tại trong năm.

1.5.3 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|------------|-----------|
| Nút nhấn 0 | Bật menu |

1.6 Chức năng bấm giờ (MODE 6)

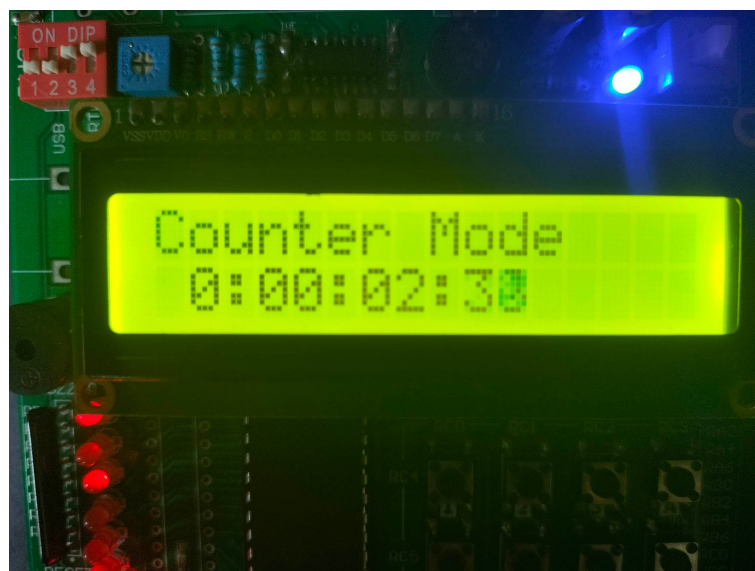
Ý tưởng hiện thực chức năng: Ta sẽ sử dụng một timer có chu kỳ là 10ms để đếm giá trị ms và khi giá trị ms này tăng lên đủ 1s thì giá trị giây sẽ tăng lên 1, cũng sẽ tương

tự với phút và giờ. Đồng thời hiển thị liên tục các giá trị này lên LCD.

Các hàm hiện thực chức năng này bao gồm: `display_counter`, `counter_run` trong file `clock.c` và hiện thực timer trong `while(1)` của hàm `main`

```
1 void display_counter(){
2     LcdPrintStringS(0,0,"Counter Mode");
3     if(counter[0] >= 10) LcdPrintNumS(1,0,(int)counter[0]/10)
4     ;
5     LcdPrintNumS(1,1,(int)counter[0]%10);
6     LcdPrintStringS(1,2,":");
7     LcdPrintNumS(1,3,(int)counter[1]/10);
8     LcdPrintNumS(1,4,(int)counter[1]%10);
9     LcdPrintStringS(1,5,":");
10    LcdPrintNumS(1,6,(int)counter[2]/10);
11    LcdPrintNumS(1,7,(int)counter[2]%10);
12    LcdPrintStringS(1,8,":");
13    LcdPrintNumS(1,9,(int)counter[3]/10);
14    LcdPrintNumS(1,10,(int)counter[3]%10);
15 }
```

Program 4.6: Hàm `display_counter`



Hình 4.2: Giao diện chức năng bấm giờ

Hướng dẫn sử dụng: Ở chế độ này, nhấn nút 5 để tăng thời gian và khi đang chạy nhấn nút 5 một lần nữa sẽ tạm dừng. Khi đang tạm dừng nhấn nút 6 để reset thời gian về lại 0.

1.6.1 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|------------|------------------------------------|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 5 | Chạy/tạm dừng đồng hồ bấm giờ |
| Nút nhấn 6 | Reset giá trị trên đồng hồ bấm giờ |

1.7 Chức năng hẹn báo thức (MODE 7)

Ý tưởng để hiện thực chức năng này là ta sẽ tạo ra một giao diện để người dùng có thể thao tác và cài đặt thời gian để báo thức (bao gồm giờ và phút). Khi cài đặt xong thì sẽ trở về giao diện bình thường và tới khi thời gian trên đồng hồ đúng bằng thời gian báo thức được thiết lập sẵn thì chuông sẽ kêu lên và màn hình sẽ hiện thị với nội dung là đã đến giờ báo thức. Lúc này người dùng có thể tắt báo thức bằng nút nhấn hoặc sau một khoảng thời gian thì báo thức cũng sẽ tự động tắt.

Đầu tiên để hiện thực một giao diện giúp người dùng cài đặt báo thức ta tiến hành như sau:

```
1 void Alarm(){
2     if (is_key_down(0)){
3         index_set = (index_set+1)%3;
4         if (index_set == 0) index_set = 1;
5     }
6
7     if (is_key_pressed(6)){
8         increase();
9     }
10    if (is_key_pressed(4)){
11        decrease();
12    }
13    if (is_key_pressed(5)){
14        hour_alarm = hour;
15        minute_alarm = minute;
16        status = MODE_1;
17        alarm_flag = 1;
18    }
19
20    LcdClearS();
21    LcdPrintStringS(0,0,"Moi dat bao thuc");
22    Display_alarm();
23    DisplayLcdScreen();
24 }
```

Việc hiện thực tăng giảm các thông số về giờ và phút cũng như chuyển đổi giữa các thông số sẽ tương tự như cách làm đã hiện thực ở chức năng điều chỉnh thời gian, thứ, ngày, tháng, năm. Ta sẽ hiển thị một dòng chữ "Mời đặt báo thức" ở dòng phía trên của LCD và dòng phía dưới sẽ hiển thị các thông số giờ và phút chớp tắt mỗi 50ms theo hàm Display_alarm() như sau:

```
1 void Display_alarm(){
2     LcdPrintNumS(1,5, hour/10);
3     LcdPrintNumS(1,6, hour%10);
4     if(index_set == 1 && blink_flag == 1){
5         LcdPrintStringS(1,5, "  ");
6
7     }
8     LcdPrintStringS(1,7, ":");
9     LcdPrintNumS(1,8, minute/10);
10    LcdPrintNumS(1,9, minute%10);
11    if(index_set == 2 && blink_flag == 1){
12        LcdPrintStringS(1,8, "  ");
13    }
14 }
```

Ta sẽ sử dụng 2 biến hour_alarm và minute_alarm để lưu trữ lại các giá trị giờ, phút báo thức mà người dùng muốn hẹn giờ báo thức, 2 biến này lưu giá trị khi nút nhấn 5 được nhấn. Sau khi nhấn nút số 5 thì đồng hồ sẽ quay trở về chế độ hiển thị giờ thông thường và cờ alarm_flag để đánh dấu người dùng đã đặt báo thức sẽ được tăng lên 1.

Khi ở chế độ hiển thị bình thường, nếu báo thức đã được cài đặt thì một chiếc chuông nhỏ sẽ được hiển thị lên ở góc dưới bên phải của LCD với hình dạng như sau:



Cách tạo ra chiếc chuông nhỏ này sẽ được trình bày ở phần sau. Khi thời gian của đồng hồ đúng bằng thời gian giờ, phút báo thức thì màn hình LCD sẽ hiển thị 2 dòng chữ "TIME OUT" và "Hãy tắt báo thức" chớp tắt liên tục mỗi 150ms cho tới khi hết thời gian 10s thì báo thức sẽ tự tắt và trở về hiển thị thời gian như bình thường. Trong thời gian 10s này cũng như trước khi tới giờ báo thức, người dùng có thể tắt thủ công bằng cách nhấn nút nhấn 5. Phần này được hiện thực như sau:

```
1 if (hour == hour_alarm && minute == minute_alarm &&
2     alarm_flag == 1 && second <= 10){
3     LcdClearS();
4     LcdPrintStringS(0,4, "TIME-OUT");
5     LcdPrintStringS(1,0, "HAY TAT BAO THUC");
6     PORTCbits.RC2 = 0;
7     if (flag_alarm_blink == 1){
```

```
7         LcdPrintStringS(0,3,"                ");
8         LcdPrintStringS(1,0,"                ");
9         flag_alarm_blink = 0;
10    }
11    if (buzzer_flag == 1){
12        PORTCbits.RC2 = 1;
13        buzzer_flag = 0;
14    }
15    if (is_key_down(5)){
16        alarm_flag = 0;
17    }
18    if (second == 10){
19        alarm_flag = 0;
20        PORTCbits.RC2 = 0;
21    }
22    DisplayLcdScreen();
23 }
```

Đồng thời trong quá trình báo thức thì tiếng chuông tạo ra từ buzzer sẽ kêu lên với chu kỳ bật tắt là 0.25s. Buzzer sẽ được bật bằng lệnh **PORTCbits.RC2 = 1** và tắt bằng lệnh **PORTCbits.RC2 = 0**. Tuy nhiên do buzzer và ma trận phím trong mạch hiện thực sử dụng chung một ngõ RC2 vì vậy để buzzer có thể hoạt động ta phải tiến hành điều chỉnh ở hàm `scan_key_matrix` và tiến hành tắt các nút nhấn ở cột nối với RC2 ở ma trận phím khi báo thức được cài đặt và sẽ trở về bình thường khi chưa cài báo thức hoặc báo thức được tắt. Cụ thể trong `while(1)` ở hàm `main` ta thực hiện như sau:

```
1 while (1)
2 {
3     if (flag_timer3 == 1){
4         SetTimer3_ms(50);
5         if (alarm_flag == 0){
6             scan_key_matrix();
7         }
8         else{
9             scan_key_matrix1();
10            PORTCbits.RC2 = 0;
11        }
12        ...
13    }
14    ...
15 }
```

Hàm `scan_key_matrix1` được hiện thực như sau:

```
1 void scan_key_matrix1()
2 {
3     int i,j;
4     for(i=0;i<MAX_ROW;i++)
```

```
5 {
6     PORT_BUTTON = ~arrayMaskOutputOfKey1[i];
7     for(j=0; j<MAX_COL; j++)
8     {
9         if (j == 2) continue;
10        if((PORT_BUTTON & arrayMaskInputOfKey[j]) == 0)
11            key_code[i*MAX_ROW+j] = key_code[i*MAX_ROW+j] + 1;
12        else
13            key_code[i*MAX_ROW+j] = 0;
14    }
15 }
16 }
```

Hiệu ứng chớp tắt của dòng chữ khi báo thức và bật tắt của buzzer được tạo ra thông qua các biến `flag_alarm_blink` và `buzzer_flag`. Các biến này tăng lên 1 hoặc giảm xuống 0 theo chu kì được quy định sẵn ở các biến `blink_count` (bằng 3 (150ms)) và `buzzer_count`(bằng 5 (250ms)). Các biến này sẽ bị trừ đi 1 đơn vị mỗi 50ms và khi xuống tới 0 thì sẽ tăng các biến `flag_alarm_blink`, `buzzer_flag` lên 1. Ta hiện thực trong vòng `while(1)` ở hàm `main` như sau:

```
1 while (1)
2 {
3     if (flag_timer3 == 1){
4         ...
5         blink_count--;
6         ...
7         buzzer_count--;
8     }
9     if (blink_count <= 0){
10        flag_alarm_blink = 1;
11        if (blink_count == -3){
12            blink_count = 3;
13        }
14    }
15    if (buzzer_count <= 0){
16        buzzer_flag = 1;
17        if (buzzer_count == -5){
18            buzzer_count = 5;
19        }
20    }
21    ...
22 }
```

1.7.1 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|------------|--|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 4 | Nút nhấn để giảm giá trị |
| Nút nhấn 6 | Nút nhấn để tăng giá trị |
| Nút nhấn 5 | Lưu giờ báo thức/ Tắt báo thức khi ở chế độ hiển thị bình thường |
| Nút nhấn 1 | Di chuyển giữa thông số giờ hoặc phút để cài đặt báo thức |

1.8 Chế độ hiển thị hoạt ảnh (MODE 8)

1.8.1 Cách để tạo ra một kí tự đặc biệt trên LCD

Một LCD sẽ gồm 3 vị trí lưu trữ chính bao gồm:

- **DDRAM (Data Display Random Access Memory):** Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình và khi ghi vào vùng RAM này một mã 8 bit, LCD sẽ hiển thị tại vị trí tương ứng trên màn hình một kí tự có mã 8 bit đã cung cấp
- **CGROM (Character Generation Read Only Memory):** Đây là vùng RAM dùng để chứa các kí tự mà LCD có thể hiển thị mà nó được thiết lập sẵn
- **CGRAM (Character Generation Random Access Memory):** Đây là vùng RAM dùng trống và cho phép người dùng có thể tạo ra các kí tự đồ họa của riêng mình. Tuy nhiên vùng này có dung lượng khá hạn chế (khoảng 8 kí tự loại 5x8 điểm ảnh)

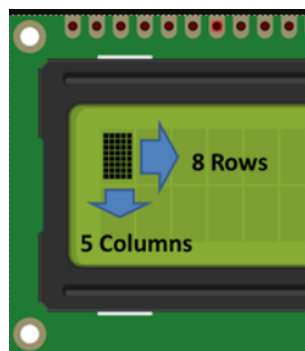
| Address | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|----------|------|------|------|---------|------|------|------|------|------|------|------|------|--------|------|------|------|
| xxxx0000 | (1) | | | 00P`P | | | | | | | | | -93αp | | | |
| xxxx0001 | (2) | | | !1AQaa | | | | | | | | | 74äq | | | |
| xxxx0010 | (3) | | | "2BRbr | | | | | | | | | 「イツxpe | | | |
| xxxx0011 | (4) | | | #3CScs | | | | | | | | | 」ウてεε | | | |
| xxxx0100 | (5) | | | \$4DTdt | | | | | | | | | 、エトμΩ | | | |
| xxxx0101 | (6) | | | %5EUeu | | | | | | | | | ・オナ1σÜ | | | |
| xxxx0110 | (7) | | | &6FVfv | | | | | | | | | ヲカニヨpΣ | | | |
| xxxx0111 | (8) | | | '7GWgw | | | | | | | | | アキヌgπ | | | |
| xxxx1000 | (1) | | | <8HXhx | | | | | | | | | イクネリJX | | | |
| xxxx1001 | (2) | | | >9IYiy | | | | | | | | | ウケル-y | | | |
| xxxx1010 | (3) | | | *:JZjz | | | | | | | | | エコんレjフ | | | |
| xxxx1011 | (4) | | | +;K[k< | | | | | | | | | オサヒロ*π | | | |
| xxxx1100 | (5) | | | ,<L¥1l | | | | | | | | | オシフワΦπ | | | |
| xxxx1101 | (6) | | | -=M]m> | | | | | | | | | ユズへンも÷ | | | |
| xxxx1110 | (7) | | | .>N^nn> | | | | | | | | | オセホゝñ | | | |
| xxxx1111 | (8) | | | /?O_o€ | | | | | | | | | ウツマ"ö | | | |

Như ta có thể thấy ở hình trên, các ô trống nằm ở bên trái ngoài cùng chính là CGRAM và ta có thể tận dụng vùng RAM này để tạo ra những kí tự đặc biệt để hiển thị nó trên LCD

Tạo kí tự đặc biệt trên CGRAM:

Để tạo kí tự đặc biệt ta cần phải ghi các giá trị vào vùng bộ nhớ của CGRAM, ô đầu tiên trong CGRAM sẽ bắt đầu ở vị trí 0x40 (mã HEX). Do CGRAM có kích thước là 64 byte do đó ta có thể tạo ra 8 kí tự đặc biệt (1 byte cho một hàng trong kí tự và 8 byte cho cả kí tự)

Mỗi kí tự trong LCD sẽ là tập hợp các pixel có kích thước 8x5:



Để tạo một kí tự ta cần truyền các bit để điều khiển bật tắt các pixel (bit 1 là bật còn bit 0 là tắt) theo từng dòng. Khi liên tục truyền vào 8 dòng ta sẽ thu được một kí tự đặc biệt được tạo ra theo mong muốn của người dùng

Các câu lệnh để tạo ra một kí tự đặc biệt được hiện thực như sau:

```
1  int n;  
2  lcd_write_cmd(64);  
3  for (n = 0; n<=7; n++) lcd_write_data(data0[n]);  
4  }
```

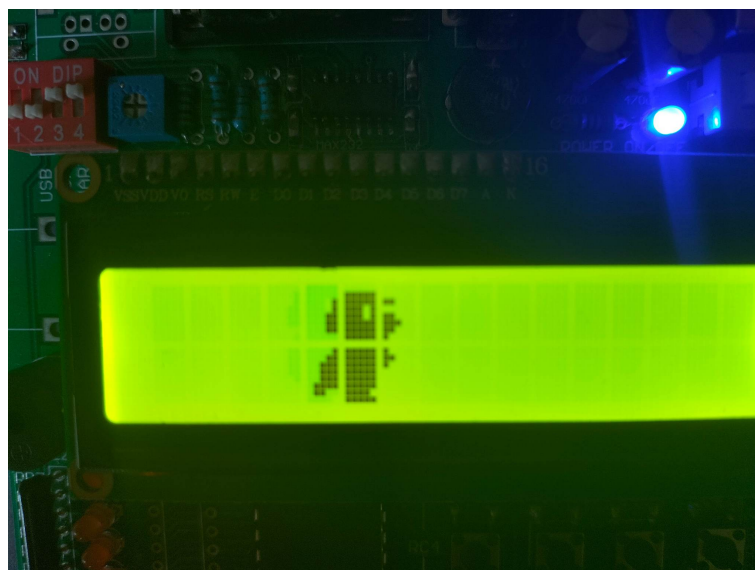
Ý tưởng của việc này là ta sẽ dùng lệnh `lcd_write_cmd` để gửi địa chỉ của ô trống trong CGRAM. Ở câu lệnh trên ta chọn ô trống đầu tiên (64 chuyển sang hex là 0x40) và sau đó ta thực hiện vòng lặp 8 lần để truyền từng dữ liệu vào từng dòng trong ô. Và nếu muốn thay đổi vị trí ô trống, ta chỉ cần cộng thêm hoặc trừ đi 8 địa chỉ của ô trống (ô trống đầu tiên là 64 còn ô trống cuối cùng là 120)

Để in kí tự đã tạo ra màn hình LCD ta chỉ cần gọi lệnh `LcdPrintCharS(x,y,k)` với x,y là vị trí muốn hiển thị trên LCD và k là số từ 0 -> 7 tương ứng với các ô nhớ trên CGRAM từ 1 -> 8

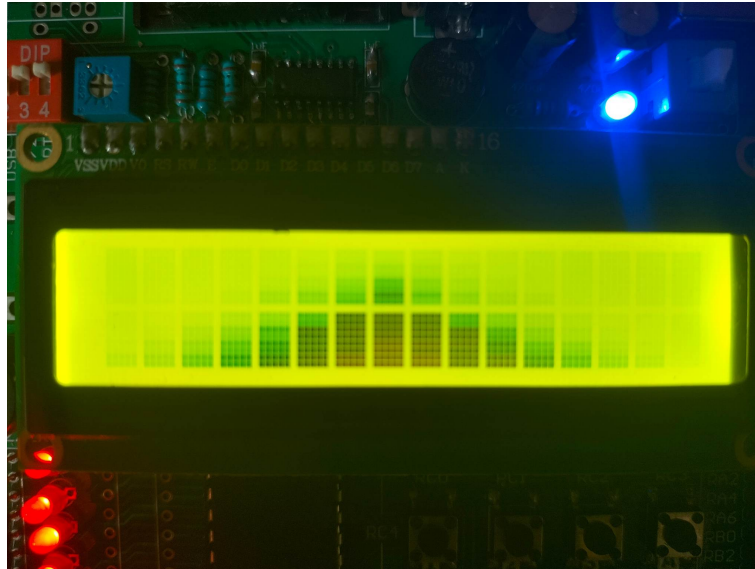
1.8.2 Hiện thị hoạt ảnh trên LCD

Ý tưởng của việc hiện thực các hoạt ảnh trên LCD là chúng ta sẽ tạo ra các kí tự đặc biệt trên LCD và sau đó hiển thị để ghép các kí tự này thành một hình ảnh nhất định và sẽ có nhiều hình ảnh khác nhau được tạo ra từ các kí tự đặc biệt do người dùng tạo ra. Và khi thực hiện hiển thị các hình ảnh này liên tiếp cách nhau một khoảng thời gian ngắn thì ta sẽ tạo ra hiệu ứng hình ảnh này đang chuyển động

Ở chế độ này sẽ có 2 lựa chọn ứng với 2 hoạt ảnh: Man Walking và Wave.



Hình 4.3: Hoạt ảnh Man Walking



Hình 4.4: Hoạt ảnh Wave

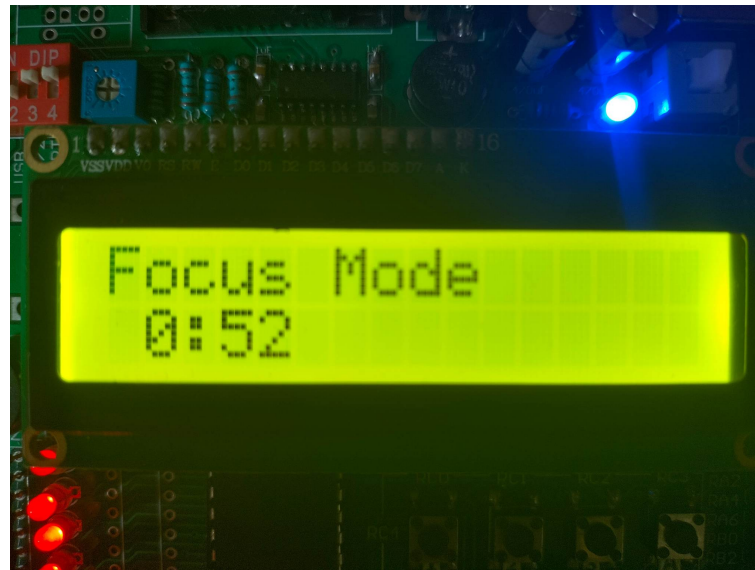
Các hàm đã hiện thực để 2 hoạt ảnh này sẽ nằm trong file `custom_character.c` và các data để tạo ra các kí tự đặc biệt được lưu trữ trong các file `"animation_data.c"`, `"animation_data1.c"`, `"animation_data2.c"`, `"animation_data3.c"`

1.8.3 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|--------------|---------------------------------------|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 4,6 | Di chuyển giữa danh mục các animation |
| Nút nhấn 5 | Chọn animation |

1.9 Chế độ tập trung (MODE 9)



Hình 4.5: Giao diện của chế độ tập trung

Ý tưởng hiện thực chức năng: Ta sẽ tạo ra một giao diện giúp người dùng cài đặt khoảng thời gian muốn đếm ngược và các giá trị về phút và giây sẽ được lưu lại (giới hạn thời gian đếm ngược là 1 tiếng). Khi người dùng nhấn nút bắt đầu đếm ngược thì cứ mỗi giây đếm lên của DS1307 thì biến giây đã được lưu lại và từ đó sẽ giảm thông số về phút. Khi thời gian đếm ngược chưa hết thì người dùng sẽ không thể thao tác bất kì thứ gì trên đồng hồ và chỉ khi thời gian đếm ngược về 0 thì người dùng mới có thể thao tác trên đồng hồ trở lại

Hướng dẫn sử dụng: Khi ở chế độ tập trung thì sẽ dùng nút 1 để chọn chỉnh giờ hoặc phút. Nhấn nút 4 để giảm và 6 để tăng thời gian, sau đó nhấn 5 để chạy chế độ tập trung. Khi chạy chế độ tập trung thì thời gian đã đặt sẽ giảm. Trong khi còn thời gian thì người dùng sẽ không thể nhấn bất kì nút nào và chỉ thao tác được sau khi thời gian về 0.

```
1 void focus_run(){
2     second = read_ds1307(ADDRESS_SECOND);
3     if(second_focus != second){
4         second_focus = second;
5         if(focus[0] > 0){
6             focus[1] = (focus[1] - 1 + 60) % 60;
7             if(focus[1] == 59){
8                 focus[0]--;
9             }
10        }
11        else{
```

```
12         if(focus[1] > 0) focus[1]--;
13     }
14 }
15 }
16
17 void Display_focus(){
18     LcdClearS();
19     LcdPrintStringS(0,0,"Focus Mode");
20     if(focus[0] >= 10) LcdPrintNumS(1,0,(int)focus[0]/10);
21     LcdPrintNumS(1,1,(int)focus[0]%10);
22     LcdPrintStringS(1,2,":");
23     LcdPrintNumS(1,3,(int)focus[1]/10);
24     LcdPrintNumS(1,4,(int)focus[1]%10);
25     if(index_focus == 1 && blink_flag == 1){
26         LcdPrintStringS(1,0," ");
27         LcdPrintStringS(1,1," ");
28         //blink_flag = 0;
29     }
30     if(index_focus == 2 && blink_flag == 1){
31         LcdPrintStringS(1,3," ");
32         LcdPrintStringS(1,4," ");
33         //blink_flag = 0;
34     }
35     DisplayLcdScreen();
36 }
```

Program 4.7: Hàm focus_run và Display_focus

1.9.1 Hướng dẫn sử dụng chức năng

Các nút nhấn được sử dụng trong chức năng này;

| Nút nhấn | Chức năng |
|------------|-----------------------------|
| Nút nhấn 0 | Bật menu |
| Nút nhấn 4 | Nút nhấn để giảm giá trị |
| Nút nhấn 6 | Nút nhấn để tăng giá trị |
| Nút nhấn 5 | Bắt đầu đếm ngược thời gian |
| Nút nhấn 1 | Chọn thông số để cài đặt |

1.10 Chức năng mở rộng ở MODE 1: Hiển thị thời gian theo giao diện khác

Để tạo ra sự phong phú trong cách hiển thị thời gian cho người dùng, nhóm đã tạo ra một giao diện khác để hiển thị thời gian nhằm nâng cao trải nghiệm của người dùng

khi sử dụng đồng hồ:



Hình 4.6: Giao diện hiển thị bình thường



Hình 4.7: Giao diện khác để hiển thị thời gian

Để hiện thực giao diện này, nhóm đã ứng dụng cách tạo ra các kí tự đặc biệt đã được trình bày ở chức năng hoạt ảnh. Và thực hiện ghép các kí tự này để tạo thành các con số từ 0->9 giúp hiển thị được thời gian một cách bắt mắt. Cách cập nhật giá trị thời gian sẽ hoàn toàn tương tự như ở MODE1 tuy nhiên thay vì hiển thị các thông số như bình thường ta sẽ hiển thị bằng các cách ghép các kí tự đặc biệt thành các con số. Do



giới hạn về mặt kích thước của LCD nên ta chỉ có thể hiển thị cùng lúc giá trị giờ và phút của đồng hồ. Đồng thời ở giao diện này, ta hoàn toàn có thể thực hiện các chức năng chuyển đổi AM/PM giống như ở giao diện hiển thị bình thường

Các hàm để hiện thực chức năng này bao gồm: `Display_time_big_num` ở file `clock.c` và các hàm trong file `custom_character.c`

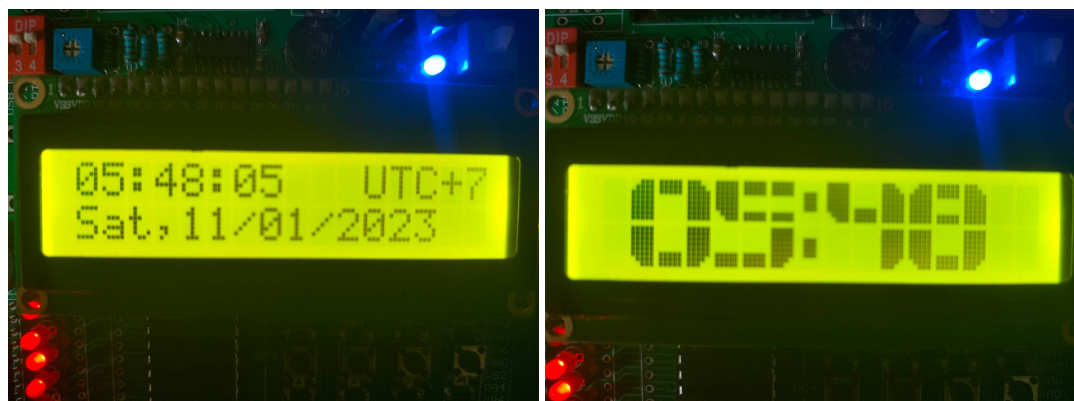
CHƯƠNG 5

Kết quả hiện thực

Sau khi tiến hành thực hiện các bước hiện thực đã nêu ở trên, nhóm đã thu được kết quả là một đồng hồ thời gian thực có giao diện tương tác thông minh có thể hoạt động tương đối ổn định. Đồng hồ bao gồm các chức năng cơ bản cần có ở một chiếc đồng hồ thông thường (hiển thị thời gian, ngày tháng, điều chỉnh thời gian, hẹn giờ báo thức,...) bên cạnh đó nhóm đã hiện thực được trên đồng hồ một số chức năng mở rộng (chế độ chọn giờ thế giới, chế độ tập trung, hiển thị hoạt ảnh,...) giúp đa dạng hóa hoạt động sử dụng của người dùng, tạo sự mới mẻ, thú vị khi sử dụng.

Hình ảnh các chức năng khi hiện thực trong thực tế:

- Chức năng hiển thị thời gian, ngày tháng năm:



Hình 5.1: Hiển thị thời gian, ngày, tháng, năm

- Chức năng giờ thế giới:



Hình 5.2: Chức năng giờ thế giới

- Chức năng điều chỉnh thời gian, ngày tháng năm:



Hình 5.3: Chức năng điều chỉnh thời gian

- Chức năng hiển thị số ngày, số tuần trong năm:



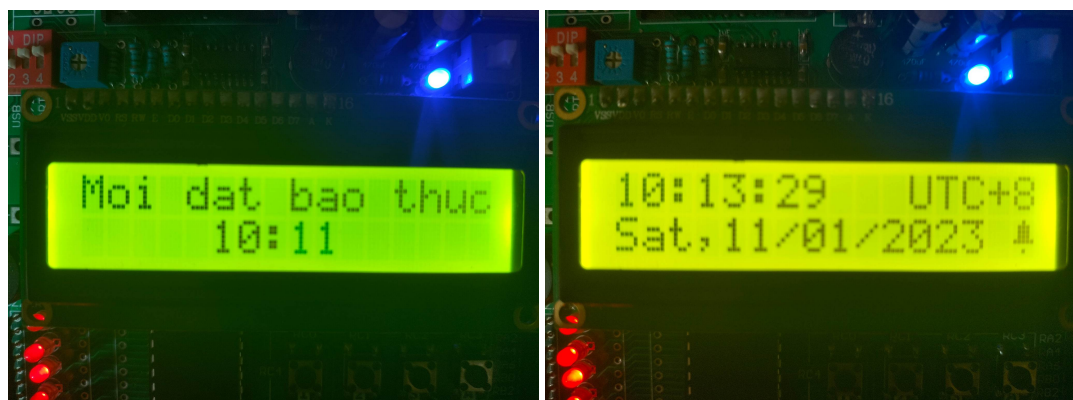
Hình 5.4: Hiển thị số ngày, tuần trong năm

- Chức năng bấm giờ:



Hình 5.5: Chức năng bấm giờ

- Chức năng hẹn giờ báo thức:



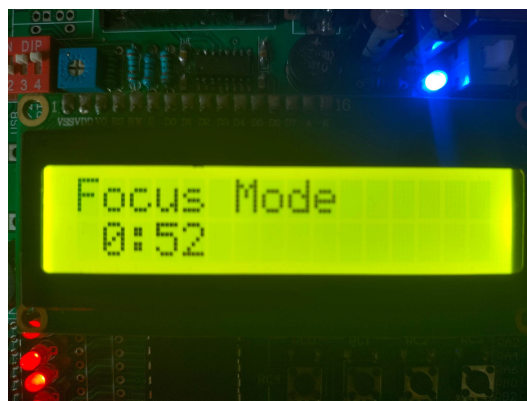
Hình 5.6: Chức năng báo thức

- Chức năng hiển thị hoạt ảnh:



Hình 5.7: Hiển thị hoạt ảnh

- Chế độ tập trung:



Hình 5.8: Chế độ tập trung



Video demo: https://drive.google.com/file/d/1vRwqpyQ15Xmt4KqTejxi9WmSP/view?usp=share_link



CHƯƠNG 6

Kết luận

1 Đánh giá kết quả thu được

Nhóm đã tìm hiểu và thực hiện thành công một đồng hồ với các giao diện tương tác thông minh. Qua quá trình hiện thực phần mềm cũng như phần cứng, nhóm đã có cái nhìn khách quan, cơ bản về cách nhận biết các linh kiện điện tử, nắm được cách hàn các linh kiện vào mạch, nhận biết cũng như sửa các lỗi gặp phải trong quá trình hiện thực phần cứng. Trong quá trình hiện thực phần mềm, nhóm đã có cơ hội tiếp cận những kiến thức mới, học hỏi các kiến thức về hiển thị LCD, timer, DS1307, sử dụng giao tiếp theo chuẩn I2C, các kiến thức về hiện thực máy trạng thái, cách làm việc với ngôn ngữ C cũng như phần mềm MPLAB,...

Thành viên của nhóm đã có sự phối hợp khá ăn ý với nhau trong việc tìm hiểu, hiện thực đề tài cũng như trong việc hoàn thiện bài báo cáo, giúp đỡ, hỗ trợ lẫn nhau vượt quá những khó khăn để đưa ra sản phẩm cuối cùng hoàn chỉnh.

2 Những khó khăn gặp phải khi hiện thực đề tài

Trong quá trình hiện thực đề tài, nhóm đã gặp phải những khó khăn như sau:

- Khó khăn trong quá trình hàn các linh kiện vào mạch: mỗi hàn bị lỗi, gặp khó khăn khi hàn các linh kiện có kích thước nhỏ, tốn khá nhiều thời gian trong việc nhận biết các linh kiện,...
- Khó khăn trong quá trình kiểm tra mạch và sửa lỗi
- Khó khăn khi hiện thực và sửa lỗi trong quá trình hiện thực phần mềm

3 Hướng phát triển trong tương lai

Trong tương lai khi có cơ hội được phát triển, hoàn thiện hệ thống, nhóm sẽ có một số kế hoạch:

- Hiện thực thêm chức năng mới cho đồng hồ: có chức năng kết nối với internet, cập nhật và hiển thị các giá trị về thời tiết, nhiệt độ, độ ẩm, hiện thực một số chức năng hỗ trợ cho người dùng giải trí,...
- Nâng cấp phần cứng cho đồng hồ: nâng cấp LCD có kích thước lớn hơn, có khả năng hiển thị màu,...
- Cải thiện hiệu suất và nâng cao tính ổn định của hệ thống, cải thiện giải thuật của các chức năng đã hiện thực
- Nâng cấp điều khiển đồng hồ thay vì chỉ thông qua các nút nhấn, ta sẽ có thể điều khiển thông qua một trang web, một ứng dụng hay các thiết bị khác,...

...