**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**BÀI TẬP LỚN**

**THIẾT KẾ HỆ THỐNG NHÚNG**

**ĐỀ TÀI**

**ĐỒNG HỒ KỸ THUẬT SỐ HIỂN THỊ LÊN LCD DÙNG DS1307**

**GVHD: Nguyễn Phan Hải Phú**

**Nhóm: 6 – Lớp: L03**

|  |  |
| --- | --- |
| **SINH VIÊN THỰC HIỆN** | **MSSV** |
| HOÀNG VŨ | 2213992 |
| LỀNH TÚ LỘC | 2211921 |
| LƯƠNG TRẦN ĐÔNG KHANG | 2211435 |

Thành phố Hồ Chí Minh, ngày 1 tháng 12 năm 2024

**MỤC LỤC**

[**1. GIỚI THIỆU** 1](#_Toc183972364)

[1.1. Tổng quan 1](#_Toc183972365)

[1.2. Nhiệm vụ đề tài 1](#_Toc183972366)

[1.3. Phân chia công việc 1](#_Toc183972367)

[**2. TỔNG QUAN LÝ THUYẾT VÀ LINH KIỆN** 2](#_Toc183972368)

[2.1. MODULE HW-111 2](#_Toc183972369)

[2.2. DS1307 3](#_Toc183972370)

[2.3. AT24C32 4](#_Toc183972371)

[2.4. LCD1602 4](#_Toc183972372)

[2.5. BUTTON 5](#_Toc183972373)

[2.6. SWITCH 6](#_Toc183972374)

[2.7. BUZZER 7](#_Toc183972375)

[2.8. ATMEGA32 8](#_Toc183972376)

[**3. YÊU CẦU HỆ THỐNG** 8](#_Toc183972377)

[**4. NHỮNG VẤN ĐỀ THIẾT KẾ** 11](#_Toc183972378)

[**5. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG** 11](#_Toc183972379)

[5.1. Nguyên lý hoạt động 11](#_Toc183972380)

[5.2. Thành phần phần cứng 12](#_Toc183972381)

[5.3. Sơ đồ khối phần cứng 13](#_Toc183972382)

[5.4. Giải thích các khối 13](#_Toc183972383)

[5.5. Sơ đồ mạch thiết kế 14](#_Toc183972384)

[**6. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM** 18](#_Toc183972385)

[6.1. Lưu đồ giải thuật 18](#_Toc183972386)

[6.2. Chương trình phần mềm 19](#_Toc183972387)

[**7. KẾT QUẢ THỰC HIỆN** 29](#_Toc183972388)

[7.1. Mô phỏng mạch 29](#_Toc183972389)

[7.2. Sản phẩm thực tế 30](#_Toc183972390)

[7.3. Hạn chế 30](#_Toc183972391)

[**8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 31](#_Toc183972392)

[8.1. Kết luận 31](#_Toc183972393)

[8.2. Hướng phát triển 32](#_Toc183972394)

[**9. TÀI LIỆU THAM KHẢO** 32](#_Toc183972395)

# **1. GIỚI THIỆU**

1.1. Tổng quan

Đề tài đồng hồ báo thức sử dụng ATmega32 yêu cầu sinh viên thiết kế và tạo ra một đồng hồ có thể hiển thị thời gian và báo thức. Hệ thống sẽ đọc giá trị thời gian thông qua DS1307 và hiển thị lên LCD đồng thời so sánh với thời gian cài đặt trước để nhằm báo thức. Đồng thời, có tích hợp khả năng tuỳ chỉnh thời gian để linh hoạt sử dụng. Để hoàn thành đề tài, sinh viên cần có kiến thức về vi điều khiển, lập trình nhúng, mạch điện và các module. Qua đó, sinh viên sẽ rèn luyện kỹ năng thiết kế hệ thống nhúng và có sản phẩm thực tế để ứng dụng.

1.2. Nhiệm vụ đề tài

Nhiệm vụ chính của hệ thống là nhằm đảm bảo sự nắm bắt về thời gian và báo thức nhằm nhắc nhở mốc thời gian quan trọng tránh để lãng quên hoặc trễ giờ. Đảm bảo được sự chủ động của việc sử dụng lượng thời gian trong ngày. Do đó cần đảm bảo độ trễ là tối thiểu giữa hệ thống và thời gian thực tế. Thậm trí là một vật dụng tốt để trang trí trên bàn học hoặc đầu giường.

1.3. Phân chia công việc

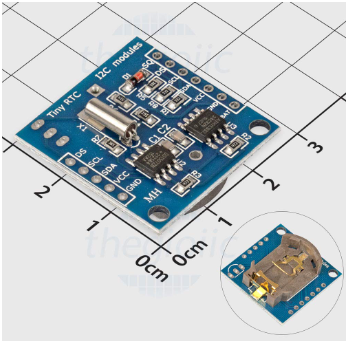
|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | Công việc | Đánh giá |
| Hoàng Vũ | Thiết kế phần mềm, Thiết kế phần cứng | 40% |
| Lềnh Tú Lộc | Hỗ trợ phần mềm, vẽ PCB, viết báo cáo | 30% |
| Lương Trần Đông Khang | Hỗ trợ phần cứng, vẽ PCB, làm slide | 30% |

Trong quá trình thực hiện đề tài, các thành viên cùng nhau thực hiện và hỗ trợ lẫn nhau, tất cả đều hoàn thành công việc đúng thời gian để chuẩn bị báo cáo.

Thời gian thực hiện đề tài: 2 tháng

# **2. TỔNG QUAN LÝ THUYẾT VÀ LINH KIỆN**

## 2.1. MODULE HW-111

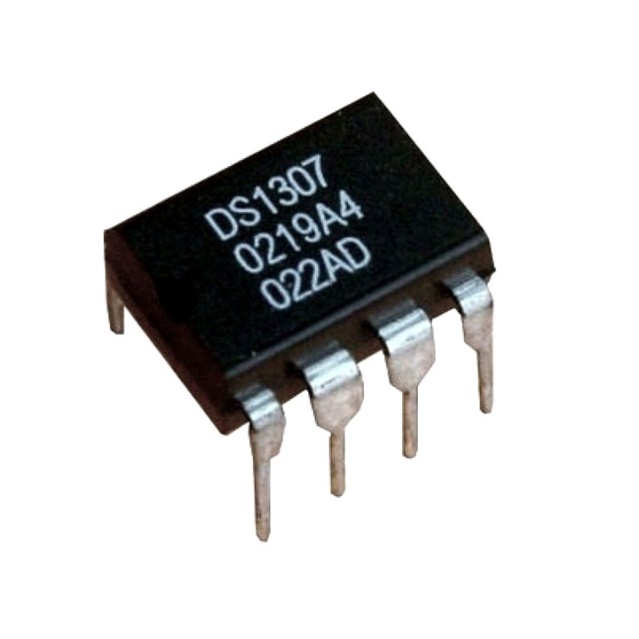


Là một mạch thời gian thực sử dụng DS1307 được sử dụng để cung cấp thông tin thời gian: ngày, tháng, năm, giờ, phút, giây,...cho vi điều khiển qua giao tiếp I2C. Ngoài ra mạch còn được tích hợp thêm IC EEPROM AT24C32 để lưu trữ thông tin khi cần, thích hợp cho các ứng dụng điều khiển hoặc đồng bộ dữ liệu thời gian thực RTC. Có một nguồn dự phòng từ pin CR2032 nhằm giữ cho DS1307 hoạt động dù có bị mất điện. Nếu chỉ sử dụng pin có thể sử dụng lên tới 1 năm. Nên có thể đảm bảo DS1307 hoạt động tới khi hết life circle của sản phẩm.

**Thông số kỹ thuật:**

* IC DS1307 giao tiếp I2C real-time clock chip (RTC)
* Bộ nhớ 24C32 EEPROM I2C 32K
* Sử dụng pin 2032 Lithium
* IC DS1307 không thể đọc, viết và lưu data khi không có nguồn.
* Kích thước: 27x28x8.4mm
* Điện áp làm việc: 3.3V đến 5V

## 2.2. DS1307



Chức năng thời gian thực: Đếm giây, phút, giờ, ngày, tháng và năm với khả năng bù năm nhuận đến năm 2100.

Bộ nhớ RAM không mất dữ liệu: 56 byte RAM không mất dữ liệu dùng để lưu trữ dữ liệu.

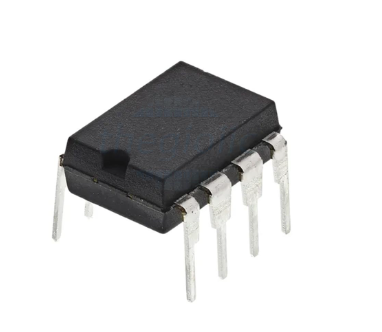
Giao thức 2 dây: Giao tiếp thông qua giao thức I2C (2 dây SDA và SCL).

Tiêu thụ năng lượng thấp, tiêu thụ ít hơn 0,5mA ở chế độ pin dự phòng khi bộ dao động đang chạy.

Dải nhiệt độ hoạt động trong khoảng từ -40oC đến +85oC.

Gói linh kiện có sẵn trong các gói 8 chân DIP hoặc SOIC.

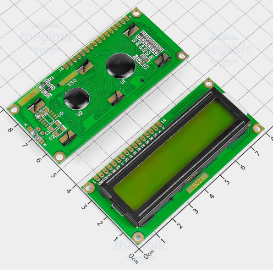
## 2.3. AT24C32



Thông số kỹ thuật:

* Định dạng bộ nhớ EEPROM
* Dung lượng 32Kbit.
* Giao tiếp I2C.
* Nhiệt độ hoạt động 0°C đến 70°C

## 2.4. LCD1602

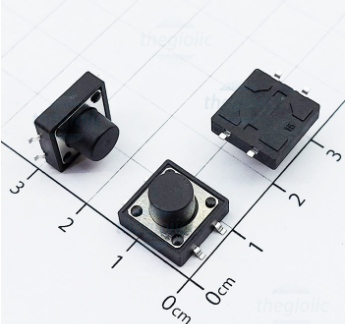


LCD ký tự 16x2 chữ đen nền vàng xanh có IC điều khiển HD44780 Super Twisted Nematic (STN) LCD. Giao tiếp đơn giản với 4 bit hoặc 8 bit. Giao tiếp dễ dàng với các loại vi điều khiển và board Arduino, ATMEL AVR hoặc PIC.

Thông số kĩ thuật:

* LCD STN độ tương phản cao 16x2
* Chữ đen nền vàng xanh
* Điện áp hoạt động: +5.0VDC
* Đèn LED nền vàng xanh
* Ký tự 5x8 dot
* IC điều khiển HD44780 hoặc tương đương
* Giao tiếp 4 or 8 bit
* Module: 80.0 x 36.0 x 13.5mm
* Khung chữ: 66.0 x 16.0mm

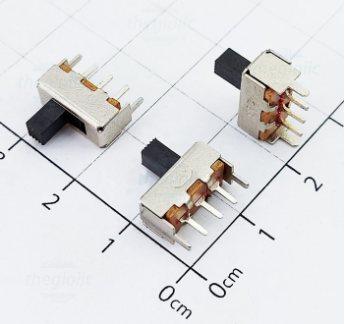
## 2.5. BUTTON



Thông số kỹ thuật:

* Hoạt động: nhấn nhả
* Kích thước: 12x12x8mm
* Dòng định mức tiếp điểm 0.1A @ 12 VDC
* Nhiệt độ hoạt động: -35 đến +85°C
* Tuổi thọ: 100000 lần

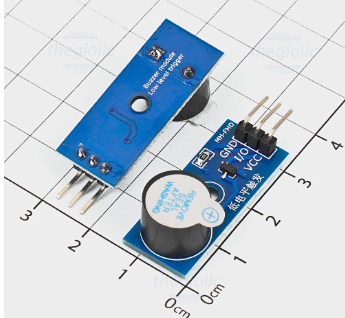
## 2.6. SWITCH



Thông số kỹ thuật:

* Chế độ hoạt động: ON-OFF
* Dòng điện định mức: 0.5A
* Điện trở tiếp điểm: 20m
* Điện áp định mức: 50V DC
* Nhiệt độ hoạt động: -30 → +70°C
* Kích thước: 21.1 x 5.5 x 6mm
* Tuổi thọ: 100000 lần

## 2.7. BUZZER

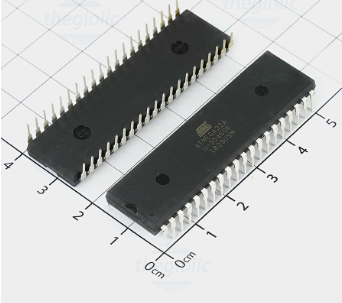


Module còi 5V (Buzzer) được sử dụng để phát ra âm thanh khi kích tín hiệu, được ứng dụng trong các hệ thống cảnh báo, báo trộm,... Thông số kỹ thuật  Điện áp sử dụng: 3.3~5VDC Tín hiệu kích: TTL mức thấp Low 0VDC.

Thông số kỹ thuật:

* Điện áp sử dụng: 3.3~5VDC
* Tín hiệu kích: TTL mức thấp Low 0VDC
* Kích thước: 38 x 13 mm

## 2.8. ATMEGA32



Thông số kỹ thuật:

* Họ IC: AVR.
* Độ rộng data bus: 8-bit
* Tốc độ: 8MHz.
* Giao tiếp: I2C, SPI, UART/USART
* Số cổng I/O: 32.
* Kích thước bộ nhớ: 32KB(16Kx16)
* Loại bộ nhớ: FLASH
* EEPROM, RAM: 1K x 8, 2K x 8.
* Điện áp cấp: 2.7 V ~ 5.5 V.
* Nhiệt độ hoạt động: -40°C ~ 85°C

# **3. YÊU CẦU HỆ THỐNG**

* Tên sản phầm

Đồng hồ kỹ thuật số hiển thị lên LCD dùng DS1307.

* Mục đích

Liên tục cập nhật thời gian từ DS1307 và hiển thị lên LCD nhằm giúp người dùng nắm bắt được thời gian và báo thức khi đến mốc thời gian đã cài đặt sẵn.

* Input và output

- Input:

+ 5 nút nhấn: 1 nút nhấn reset, 1 nút nhấn tăng đơn vị thời gian, 1 nút nhấn giảm đơn vị thời gian, 1 nút chỉnh thời gian, 1 nút tắt báo thức.

+ 2 switch: 1 switch cho phép cài đặt thời gian báo thức, 1 switch cho phép bật/tắt chế độ báo thức.

+ 1 biến trở: điều chỉnh độ phân giải của màn hình LCD.

+ Module RTC DS1307: liên tục gửi giá trị giờ, phút, giây cho MCU

- Output:

+ LCD 16x02: Hiển thị thời gian đọc từ DS1307 và chế độ báo thức ON/OFF.

+ BUZZER: Báo thức khi thời gian chạy tới đúng thời gian đặt đúng báo thức.

* Trường hợp sử dụng

Khi bật nguồn, LCD sẽ hiển thị thời gian đọc từ module DS1307. Nếu thời gian chạy chưa đúng thì nhấn nút Setting Time để chỉnh thời gian cho đúng thời gian thực. Nếu cài báo thức thì bật switch Alarm ON/OFF và cài đặt thời gian theo ý người dùng muốn báo thức bằng switch Setting Alarm. Khi thời gian chạy đúng thời gian cài đặt báo thức thì Buzzer sẽ reo lên. Người dùng có thể dùng nút nhấn Turn Off Alarm để dừng báo thức.

* Chức năng

Hệ thống đọc thời gian từ module RTC DS1307 và hiển thị lên LCD giờ, phút, giây, ngày, tháng, năm. Khi bật chế độ báo thức lên thì nếu thời gian chạy đúng với thời gian báo thức thì Buzzer sẽ reo lên.

* Hiệu suất

- Sai lệch thời gian là 1 ngày lệch 10s.

* Chi phí sản xuất

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **TÊN LINH KIỆN** | **SỐ LƯỢNG** | **THÀNH TIỀN** |
| 1 | NPN - 2N2222 | 1 Cái | 500đ |
| 2 | Công tắc gạt SS-12F44 | 2 Cái | 1.000đ |
| 3 | Nút nhấn 12x12x7mm | 4 Cái | 2.400đ |
| 4 | Điện trở 10K 1/4W 1% | 1 Gói | 3.000đ |
| 5 | ATMEGA32A-PU | 1 Cái | 82.000đ |
| 6 | Tụ gốm 22pF – 50V | 1 Gói | 1.000đ |
| 7 | Thạch anh 32.768KHz | 1 Cái | 1.400đ |
| 8 | Biến trở đứng 1K | 1 Cái | 2.000đ |
| 9 | LCD nền xanh chữ vàng | 1 Cái | 25.000đ |
| 10 | DS1307 | 1 Cái | 5.000đ |
| 11 | Buzzer 60dB | 1 Cái | 3.000đ |
| 12 | Hộp pin 1x9V | 1 Cái | 8.500đ |
| 13 | L7805ADCD2T – CN | 1 Cái | 5.000đ |
| 14 | Hàng rào cái | 2 Cái | 4.000đ |
| Tổng: | | | 143.800đ |

* Công suất, nguồn điện

- Nguồn cung cấp: Sử dụng adapter chuyển từ 220VAC thành 5VDC

- Công suất tiêu thụ: ≤10W

* Kích thước vật lý, khối lượng

- Kích thước 150 cm2

- Khối lượng: < 500g

* Lắp đặt: được đặt trên bàn làm việc

# **4. NHỮNG VẤN ĐỀ THIẾT KẾ**

* Vấn đề ràng buộc:
* Giá thành < 200 nghìn VNĐ
* Thời gian sử dụng, tuổi thọ của sản phẩm: < 5 năm
* Hệ thống tiêu thụ năng lượng thấp: 5V
* Kích thước nhỏ gọn: < 150 cm2
* Khối lượng: < 500g
* Vấn đề chức năng:
* Điện áp 5V khi cấp cho các linh kiện được mắc song song để ổn áp
* Vật liệu chịu được nhiệt độ tốt không bị chảy vào ngày nắng nóng và phát thải khí có hại.
* Vấn đề thời gian thực:
* LCD là bộ hiển thị thời gian thực mềm với độ trễ < 0.25s
* Buzzer có độ trễ < 0.1s
* Vấn đề đồng thời
* Vừa đọc thời gian từ Module DS1307 và vừa so sánh với thời gian báo thức được cài đặt, nếu trùng thì bật báo thức.
* Vấn đề phản hồi
* Tương tác liên tục: Hệ thống liên tục đọc thời gian từ DS1307 và hiển thị lên LCD.
* Phản ứng các sự kiện không có chu kì: Thời gian cài đặt báo thức là theo ý muốn người dùng.

# **5. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG**

## 5.1. Nguyên lý hoạt động

- Sau khi khởi động, màn hình LCD sẽ hiển thị giao diện giới thiệu thiết bị “ALARMCLOCK by N6” trong 2 giây.

- Sau đó hệ thống sẽ hiện thời gian ngày tháng năm. Khi mở lần đầu người dùng sẽ phải thiết lập thời gian ngày tháng năm cho hệ thống.

- Thiết bị gồm 4 nút ấn và 2 nút gạt:

* + - SWITCH chọn mode điều chỉnh: khi SWITCH 1 ON thì hệ thống sẽ nhận biết người dung muốn điều chỉnh thời gian ngày tháng năm thứ. Nếu SWITCH 1 OFF thì hệ thống sẽ biết người dùng muốn điều chỉnh thời gian báo thức.
    - SWITCH bật tắt báo thức: khi SWITCH 2 ON thì báo thức sẽ bật và OFF thì báo thức sẽ tắt. Trạng thái của báo thức sẽ hiển thị trên LCD là ON và OFF tương ứng.
    - SETTING MODE BUTTON: khi ấn nút này hệ thống sẽ vào màn hình điều chỉnh của từng thông số và sẽ hiện trên màn hình để người dùng biết. Cứ mỗi lần ấn nút này sẽ tự động chuyến đến giá trị tiếp theo và khi đến giá trị điều chỉnh cuối ấn một lần nữa sẽ trở về màn hình chính.
      * Nếu SWITCH 1 ON: thì các giá trị điều chỉnh lần lượt là hour -> minute -> second -> year -> month -> day -> date.
      * Nếu SWITCH 1 OFF: thì các giá trị lần lượt là ALARMHOUR -> ALARMMINUTE.
* INCREASE BUTTON: Dùng để tăng các giá trị trong mode điều chỉnh khi đạt max sẽ tự động trở về giá trị 0.
* DECREASE BUTTON: Dùng để giảm các giá trị trong mode điều chỉnh khi đạt giá trị 0 sẽ tự động chuyển về giá trị max được thiết lập.
* TURN OFF ALARM BUTTON: Dùng để dừng tiếng báo thức nhằm cho hệ thống không kêu nữa, khi đã dừng báo thức thì phải thiết lập các giá trị thời gian báo thức trở lại.

## 5.2. Thành phần phần cứng

5.2.1. MCU

8-bit Microcontroller: ATmega32A thu thập các dữ liệu từ DS1307 đồng thời nhận tín hiệu từ các nút và nút gạt để hiện thông tin lên ngõ ra.

5.2.2. Ngoại vi (Peripherals)

* Ngõ vào:
* Nút nhấn
* Nút gạt switch
* Module RTC
* Ngõ ra:
  + LCD 1602
  + Buzzer

5.2.3. NGUỒN (Power Supply)

Nguồn 5V từ adapter dô kit ATmega và mắc song song ra các linh kiện.

## 5.3. Sơ đồ khối phần cứng

A diagram of a computer

Description automatically generated

## 5.4. Giải thích các khối

a. Khối xử lý trung tâm ATmega32

Đây là bộ xử lý trung tâm của hệ thống, điều khiển các thành phần khác. Nó thực hiện nhiệm vụ đọc dữ liệu từ DS1307 và các chân GPIO để điều chỉnh thời gian, bật tắt báo thức, giao tiếp với LCD và RTC.

b. Khối thời gian thực RTC: DS1307

Giao tiếp I2C: cung cấp thời gian thực.

Các giá trị điện trở, thạch anh, pin có sẵn trên module theo DATASHEET.

c. Các khối giao tiếp nút

- Nút ấn: Vào mode điều chỉnh, tang giảm giá trị, tắt báo thức.

- Nút gạt: Chọn mode điều chỉnh, bật tắt báo thức.

Đặc điểm:

- Điện trở 10Kohm kéo xuống đất, là nút thường tích cực thấp.

- Tụ gốm 22pF để chống rung phần cứng.

d. Buzzer

- Báo thức, hoạt động trong vùng 3.3 ~ 5VDC.

- Dùng một trở 220 Ohm và BJT NPN 2N2222 để đệm dòng áp cho buzzer.

## 5.5. Sơ đồ mạch thiết kế

a. Tổng quan sơ đồ

A diagram of a computer

Description automatically generated

b. Sơ đồ mạch nguồn

LCD1602

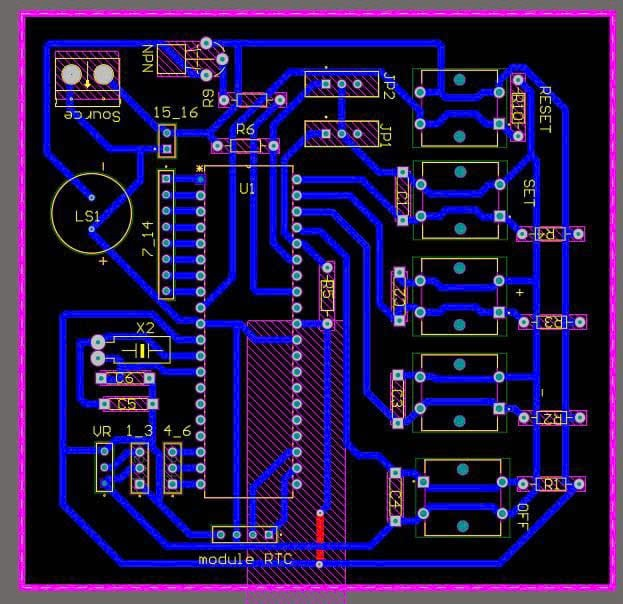
5VDC

DS1307

ATmega32

buzzer

c. Layout PCB trong Altium

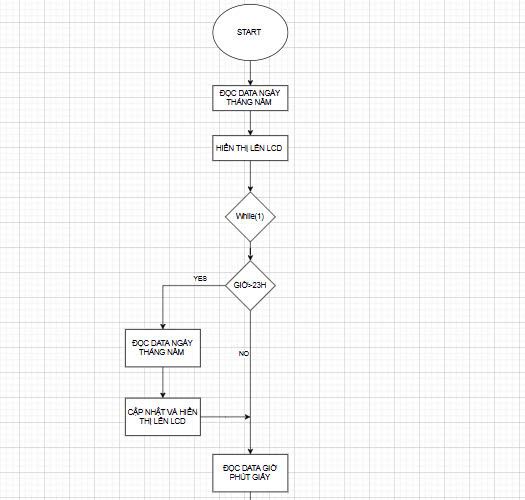


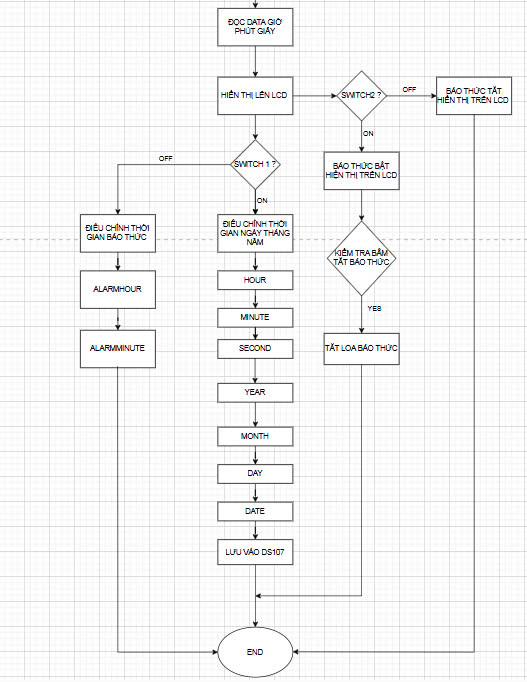
A green circuit board with many different components

Description automatically generated

# **6. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM**

## 6.1. Lưu đồ giải thuật





## 6.2. Chương trình phần mềm

|  |
| --- |
| #include <avr/io.h>  #define F\_CPU 11059200  #include <util/delay.h>  #include <stdlib.h>  #include <avr/interrupt.h>  #define F\_CPU 8000000UL  #include <stdio.h>  #include <string.h>  #include <stdbool.h>  #include <math.h>  #include <LCD.h>  #include <I2C.h>  int second,minute,hour, day, date, month, year;  int seconds,minutes,hours,dates,dayss,months,years;  int ALhour,ALMIN,ALH,ALM;  char tmp[2]; |

- #include <avr/io.h>: Bao gồm các định nghĩa cho việc truy cập vào các thanh ghi và cổng của vi điều khiển AVR.

- #define F\_CPU 11059200 và #define F\_CPU 8000000UL: Định nghĩa tần số của vi điều khiển, giúp các hàm delay và các thao tác thời gian khác hoạt động chính xác.

- #include <util/delay.h>: Sử dụng các hàm tạo độ trễ.

- #include <stdlib.h>, #include <stdio.h>, #include <string.h>, #include <stdbool.h>, #include <math.h>: Bao gồm các thư viện chuẩn của C cho việc xử lý chuỗi, toán học và các thao tác logic khác.

- #include <LCD.h>: Thư viện để điều khiển màn hình LCD.

- #include <I2C.h>: Thư viện để giao tiếp I2C.

- second, minute, hour, day, date, month, year: Các biến lưu trữ thời gian thực (giây, phút, giờ, ngày, ngày tháng, tháng, năm).

- seconds, minutes, hours, dates, dayss, months, years: Các biến lưu trữ thời gian khác, có thể được sử dụng cho mục đích so sánh hoặc thay đổi tạm thời.

- ALhour, ALMIN, ALH, ALM: Các biến lưu trữ giờ và phút cho báo thức (alarm).

- char tmp[2]: Một mảng ký tự tạm thời, có thể được sử dụng để lưu trữ các giá trị chuỗi nhỏ hoặc các ký tự riêng lẻ.

|  |
| --- |
| *uint8\_t* BCDToDecimal(*uint8\_t* bcd) {  return ((bcd >> 4) \* 10) + (bcd & 0x0F);  }  // Convert decimal to BCD  *uint8\_t* decimalToBCD(*uint8\_t* decimal) {  return ((decimal / 10) << 4) | (decimal % 10);  } |

Hai lệnh BCDToDecimal và decimalToBCD là hai lệnh dùng để chuyển đổi qua lại giữa số thập phân và số BCD. Vì RTC thường lưu trữ thời gian dưới dạng BCD và cần chuyển đổi về dạng thập phân để hiển thị giờ, phút, giây lên màn hình LCD. Khi cần lưu trữ hoặc gửi dữ liệu dưới dạng BCD, ta cần chuyển đổi từ số thập phân sang BCD.

|  |
| --- |
| void RTC\_WRITE\_CALENDER (char WRITE\_CLOCK\_CALENDER, char years, char months, char dates,char dayss)  {  years = decimalToBCD(years);  months = decimalToBCD(months);  dayss = decimalToBCD(dayss);  dates = decimalToBCD(dates);    I2C\_Start(Device\_write\_address);  I2C\_Write(WRITE\_CLOCK\_CALENDER);    I2C\_Write(dayss);  I2C\_Write(dates);  I2C\_Write(months);  I2C\_Write(years);  I2C\_Stop();    }  void RTC\_WRITE\_CLOCK (char WRITE\_CLOCK\_ADDRESS, char hours, char minutes, char seconds)  {  hours = decimalToBCD(hours);  minutes = decimalToBCD(minutes);  seconds = decimalToBCD(seconds);  I2C\_Start(Device\_write\_address);  I2C\_Write(WRITE\_CLOCK\_ADDRESS);    I2C\_Write(seconds);  I2C\_Write(minutes);  I2C\_Write(hours);    I2C\_Stop();  }  void RTC\_READ\_CLOCK (char READ\_CLOCK\_ADDRESS)  {  I2C\_Start(Device\_write\_address);  I2C\_Write(READ\_CLOCK\_ADDRESS);  I2C\_Repeated\_Start(Device\_Read\_address);    second = I2C\_Read\_Ack();  minute = I2C\_Read\_Ack();  hour = I2C\_Read\_NACK();  I2C\_Stop();  }  void RTC\_READ\_CALENDER( char READ\_CALENDER\_ADDRESS)  {  I2C\_Start(Device\_write\_address);  I2C\_Write(READ\_CALENDER\_ADDRESS);  I2C\_Repeated\_Start(Device\_Read\_address);  day = I2C\_Read\_Ack();  date = I2C\_Read\_Ack();  month = I2C\_Read\_Ack();  year = I2C\_Read\_NACK();  I2C\_Stop();  } |

- **Ghi thông tin vào RTC**: Các hàm RTC\_WRITE\_CALENDER và RTC\_WRITE\_CLOCK sử dụng giao tiếp I2C để ghi thông tin ngày tháng và giờ vào RTC, sau khi chuyển đổi các giá trị từ thập phân sang BCD.

- **Đọc thông tin từ RTC**: Các hàm RTC\_READ\_CLOCK và RTC\_READ\_CALENDER sử dụng giao tiếp I2C để đọc thông tin giờ và ngày tháng từ RTC, và lưu trữ vào các biến tương ứng.

|  |
| --- |
| void adjust\_value(char \*label, int \*value, int max\_value, unsigned char position) {  char temp[10];  send\_a\_command(0x01);  *\_delay\_ms*(1);  send\_a\_command(0x80);  send\_a\_string(label);  *\_delay\_ms*(1);  while (1) {  send\_a\_command(position);  *\_delay\_ms*(1);  *sprintf*(temp, "%02d", \*value);  send\_a\_string(temp);  *\_delay\_ms*(1);  if (bit\_is\_set(PINA, 1)) {  if (\*value < max\_value) {  (\*value)++;  } else {  \*value = 0;  }  *\_delay\_ms*(10);  }  if (bit\_is\_set(PINA, 2)) {  if (\*value > 0) {  (\*value)--;  } else {  \*value = max\_value;  }  *\_delay\_ms*(6);  }  if (bit\_is\_set(PINA, 0)) break;  }  }  int NHUANKH(int year) {  return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);  }  int MAXDATE(int month, int year) {  if (month == 2)  { // Tháng 2  if (*is\_leap\_year*(year)) return 29;    else return 28;  }  else if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) return 31;  else return 30;  } |

- Hàm adjust\_value: cho phép điều chỉnh giá trị hiển thị trên màn hình LCD thông qua các nút bấm.

- Hàm NHUANKH: kiểm tra xem năm có phải đang là năm nhuận không.

- Hàm MAXDATE: trả về số ngày tối đa trong một tháng của một năm nhất định.

|  |
| --- |
| int main(void)  {  char temp [10];  char buffer[20];  char\* days[7] = {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"};  I2C\_Init();  DDRA = 0xFF;  LCD\_Data\_Dir = 0xFF;  LCD\_Command\_Dir |= (1 << RS)| (1 << RW) | (1 << EN); // setting compare value equal to counter clock frequency to get an interrupt every second  send\_a\_command(0x01);  *\_delay\_ms*(1);  send\_a\_command(0x38);  *\_delay\_ms*(1);  send\_a\_command(0x0C);  *\_delay\_ms*(1);    send\_a\_string("BTL TKHTN NHOM 6");  *\_delay\_ms*(250);  send\_a\_command(0x01); |

- Khởi tạo các biến và mảng ngày trong tuần.

- Khởi tạo giao tiếp I2C và cấu hình các cổng vào/ra.

- Thiết lập màn hình LCD và hiển thị dòng chữ “BTL TKHTN NHOM 6”.

|  |
| --- |
| START:  send\_a\_command(0x80);  //format 24h  RTC\_READ\_CALENDER(3);  *\_delay\_ms*(1);  *sprintf*(buffer,"%02x/%02x/%02x %3s ", date, month, year, days[day-1] );  send\_a\_string(buffer);  *\_delay\_ms*(1); |

- Đọc và hiển thị lịch từ RTC.

|  |
| --- |
| while (1)  {  if(hour>23)  {  send\_a\_command(0x80);  //format 24h  RTC\_READ\_CALENDER(3);  *\_delay\_ms*(1);  *sprintf*(buffer,"%02x/%02x/%02x %3s ", date, month, year, days[day-1] );  send\_a\_string(buffer);  *\_delay\_ms*(1);  }  RTC\_READ\_CLOCK(0);  *\_delay\_ms*(1);  *sprintf*(buffer, "%02x:%02x:%02x ",(hour & 0b00111111), minute, second);  send\_a\_command(0xC0); // dua con tro toi vi tri thu 4  send\_a\_string(buffer); // gui chuoi ki tu ra LCD  *\_delay\_ms*(1);  send\_a\_command(0xC0 + 0x09); |

- Đọc giờ từ RTC và hiển thị lên màn hình LCD.

|  |
| --- |
| if (bit\_is\_set(PINA,5))  {  send\_a\_string(" ON ");  if(bit\_is\_set(PINA,3)&&(bit\_is\_clear(PINA,7)))  {  ALMIN = 0;  ALhour = 0;  }  if ((decimalToBCD(ALhour) == hour) && (decimalToBCD(ALMIN) == minute)) PORTA &= (0 << PINA7);  else PORTA |= (1 << PINA7);  }  else  {  send\_a\_string("OFF");  PORTA |= (1 << PINA7);  } |

- Kiểm tra trạng thái báo thức và điều khiển loa dựa trên trạng thái này.

|  |
| --- |
| if((bit\_is\_set(PINA,4))&&(bit\_is\_set(PINA,0)))  {  seconds = BCDToDecimal(second);  minutes = BCDToDecimal(minute);  hours = BCDToDecimal(hour);  dayss = BCDToDecimal(day);  dates = BCDToDecimal(date);  months = BCDToDecimal(month);  years = BCDToDecimal(year);  adjust\_value("hour: ", &hours, 23,0x85);  adjust\_value("minute: ", &minutes, 59,0x87);  adjust\_value("second: ", &seconds, 59,0x87);  adjust\_value("year: ", &years, 100,0x85);  adjust\_value("month: ", &months, 12,0x86);  adjust\_value("date: ", &dates, MAXDATE(years,months),0x85);  send\_a\_command(0x01);  send\_a\_command(0x80);  send\_a\_string("thu: ");  while(1)  {  send\_a\_command(0x84);  *\_delay\_ms*(1);  *sprintf*(temp,"%02s", days[dayss-1]);  send\_a\_string(temp);  if (bit\_is\_set(PINA, 1)) // Nút t?ng  {  if (dayss < 7) dayss++; // T?ng gi  else dayss = 0; // ??t l?i v? 0 n?u gi? ??t 24  // Th?i gian tr? ?? tránh nh?n nhi?u l?n  }  if (bit\_is\_set(PINA,2)) // Nút gi?m  {  if (dayss > 0) dayss--; // Gi?m gi?  else dayss = 6; // ??t l?i v? 23 n?u gi? ??t 0  // Th?i gian tr? ?? tránh nh?n nhi?u l?n  }  if(bit\_is\_set(PINA,0)) break;  }  send\_a\_command(0x01);  RTC\_WRITE\_CLOCK(0,hours,minutes,seconds);  RTC\_WRITE\_CALENDER(3,years,months,dates,dayss);  goto START; |

- Cho phép người dùng điều chỉnh thời gian (giờ, phút, giây, ngày, tháng, năm) thông qua các nút nhấn.

|  |
| --- |
| if((bit\_is\_clear(PINA,4))&&(bit\_is\_set(PINA,0)))  {  ALH = BCDToDecimal(hour);  adjust\_value("ALHOUR: ", &ALH, 23,0x87);  ALhour = ALH;    ALM = BCDToDecimal(minute);  adjust\_value("ALminute: ",&ALM,59,0x89);  ALMIN = ALM;  send\_a\_command(0x01);  goto START; |

- Cho phép người dùng thiết lập báo thức (giờ, phút) thông qua các nút nhấn.

# **7. KẾT QUẢ THỰC HIỆN**

## 7.1. Mô phỏng mạch

A computer screen shot of a circuit board

Description automatically generated

## 7.2. Sản phẩm thực tế

A close-up of a circuit board

Description automatically generated

## 7.3. Hạn chế

Sau khi mô phỏng cũng như là thực hiện mạch xong thì nhóm nhận thấy hệ thống có một số hạn chế nhất định sau:

- Độ chính xác thời gian: Với độ lệch là cứ 1 ngày 10s thì nhóm nhận thấy hệ thống không phù hợp với các ứng dụng yêu cầu độ chính xác cao, nếu chỉ dùng để báo thức hoặc nhắc nhở về thời gian thì sẽ không gây hậu quả đáng kể cho người dùng. Bên cạnh đó nếu nhận thấy thời gian hiển thị trên hệ thống sai lệch so với thời gian thực thì người dùng có thể chỉnh giờ thông qua các nút nhấn.

- Cấu hình thời gian: Do DS1307 không có cơ chế tự động cập nhật thời gian từ mạch đồng hồ chính quốc tế (NTP) hoặc GPS, do đó bạn phải cập nhật thời gian bằng tay nếu thiết bị bị mất điện. DS1307 chỉ có thể lưu trữ thời gian và ngày trong định dạng BCD (Binary-Coded Decimal), do đó bạn cần phải chuyển đổi giữa BCD và số nguyên.

- Hệ thống sử dụng giao tiếp 8 bit LCD nên sẽ tốn nhiều chân PORT của vi xử lý hơn so với các giao tiếp khác như 4 bit, I2C,... Vì vậy, nếu thiết lập thêm chức năng thì phải lựa chọn lại chân vi xử lý cũng như là thiết lập lại chân PORT.

- Nguồn điện: Hệ thống sử dụng adapter để chuyển từ 220VAC thành 5VDC, chính vì vậy mà đồng hồ chỉ sử dụng trong phạm vi ngắn, không linh động trong việc di chuyển vd: để trên bàn học, gần nơi có nguồn điện 220VAC. Khi cúp điện lưới thì hệ thống sẽ không hoạt động.

- Đồng hồ chỉ cài đặt được một khung giờ báo thức, không cài đặt được nhiều khung giờ để báo thức. Loa cũng phát được một loại âm thanh “bíp”, không phát nhạc được. Nếu muốn phát nhạc thì phải sử dụng các loại module khác có bộ nhớ lưu trữ tập tin âm thanh.

# **8. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## 8.1. Kết luận

Bài tập lớn Thiết kế hệ thống nhúng với chủ đề Đồng hồ kỹ thuật số hiển thị LCD dùng DS1307 đã cung cấp một cái nhìn sâu rộng về ứng dụng vi điều khiển trong việc quản lý thời gian thực. Qua quá trình thiết kế và triển khai, nhóm đã thực hiện thành công một hệ thống nhúng có khả năng hiển thị chính xác thời gian và ngày tháng, đồng thời cung cấp chức năng báo thức.

Việc sử dụng DS1307 – một module đồng hồ thời gian thực phổ biến, kết hợp với màn hình LCD đã chứng minh được tính hiệu quả và khả năng ứng dụng cao của hệ thống. Bên cạnh đó, nhóm cũng học được thêm các kỹ thuật lập trình giao tiếp I2C, xử lý ngắt ngoại vi, hiển thị dữ liệu lên LCD và đã được áp dụng các kĩ năng trên một cách hiệu quả, giúp hệ thống hoạt động ổn định và chính xác.

Về đề tài Đồng hồ kỹ thuật số hiển thị lên LCD dùng DS1307, nhóm đã thiết kế và hoàn thành hệ thống trên. Sản phẩm của nhóm đáp ứng được đầy đủ các vấn đề ràng buộc: giá thành sản phẩm 143.800VNĐ (<200.000VNĐ), công suất ≤ 10W, kích thước sản phẩm 10x10(cm) (<150cm2), tuổi thọ của sản phẩm (dự đoán) < 5 năm, khối lượng 300g (<500g).

Tuy nhiên, hệ thống cũng tồn tại một số hạn chế như độ chính xác thời gian chưa cao và nguồn điện để duy trì hoạt động của đồng hồ chưa thật sự hiệu quả và vẫn còn nhiều thiếu sót, chính vì vậy, nhóm rất mong nhận được sự góp ý từ thầy.

## 8.2. Hướng phát triển

Từ các hạn chế và nhược điểm của hệ thống mà nhóm đã đề cập ở phần trên. Sau đây là một số hướng phát triển mà nhóm đề xuất để sản phẩm có thể hoàn thiện hơn:

- Cải thiện độ chính xác của thời gian bằng việc thay thế DS1307 bằng DS3231, DS3231 là một IC đồng hồ thời gian thực có độ chính xác cao hơn, giảm mức sai số thời gian xuống gần như không đáng kể và ít chịu ảnh hưởng của nhiệt độ môi trường.

- Thay thế hiển thị LCD bằng các loại màn hình khác có độ phân giải cao hơn như OLED hoặc TFT để có thể hiển thị đa dạng hơn.

- Tích hợp thêm các tính năng thông minh hơn như sử dụng các giao tiếp không dây như Wi – Fi, Bluetooth cho phép điều khiển từ xa bằng các ứng dụng di động, web hoặc tích hợp các cảm biến môi trường như ánh sáng, nhiệt độ, độ ẩm,... để hiển thị thêm thông tin và tự động điều chỉnh báo thức theo điều kiện cụ thể.

- Tích hợp các hệ thống điều khiển trong gia đình như: điều khiển quạt, đèn hoặc các hệ thống tưới cây tự động dựa trên thời gian và lịch trình đã được cài đặt.

- Nâng cao trải nghiệm người dùng bằng việc tích hợp thêm các âm báo thức khác nhau hoặc là nhạc để người dùng có thể chọn âm báo theo ý thích.

# **9. TÀI LIỆU THAM KHẢO**

(1) Điện tử tương lai. Truy cập từ: <https://dientutuonglai.com/ve-mach-nguyen-ly-bang-altium.html>

(2) Nguyễn Lý Thiên Trường. *Slide bài giảng: Chương 6: Giao tiếp ngoại vi*

(3) Nguyễn Lý Thiên Trường. *Slide bài giảng: Chương 8: Cổng nối tiếp*

(4) Nguyễn Lý Thiên Trường. *Slide bài giảng: Chương 10: Ngắt*

(5) Sepehr Naimi (12/12/2017). *C programming in Atmel Studio 7*

(6) Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi (2014). *The AVR Micontroller anh Embedded Systems: Using Assembly and C*