TRƯỜNG ĐẠI HỌC NHA TRANG **KHOA CÔNG NGHỆ THÔNG TIN**



**THIẾT KẾ ĐỒNG HỒ THỜI GIAN THỰC SỬ DỤNG MODULE DS1307**

**GVHD : Ths. ĐOÀN VŨ THỊNH**

**SVTH : Đặng Quang Nghĩa\_59131575**

**Nguyễn Đăng Trường Phát\_59131843**

**Lớp : 59CNTT-1**

Khánh Hòa, tháng 01 năm 2020

# MỤC LỤC

[MỤC LỤC 1](#_Toc30080674)

[DANH MỤC HÌNH 2](#_Toc30080675)

[TÓM TẮT 3](#_Toc30080676)

[1. GIỚI THIỆU 4](#_Toc30080677)

[1.1. Màn hình LCD 1602 4](#_Toc30080678)

[1.2. Module LCD I2C 5](#_Toc30080679)

[1.3. Board mạch Arduino Uno R3 6](#_Toc30080680)

[1.4. Module thời gian thực DS1307 9](#_Toc30080681)

[1.5. Một số linh kiện trang trí 10](#_Toc30080682)

[1.6. Phần mềm Fritzing 12](#_Toc30080683)

[1.7. Phần mềm Arduino IDE 12](#_Toc30080684)

[2. PHƯƠNG PHÁP NGHIÊN CỨU 14](#_Toc30080685)

[2.1. Mô phỏng nguyên lý hoạt động bằng phần mềm fritzing 14](#_Toc30080686)

[2.2. Lập trình 16](#_Toc30080687)

[2.2.1. Giao tiếp DS1307 (RTC) với Arduino 16](#_Toc30080688)

[2.2.2. Hiển thị thời gian trên LCD 18](#_Toc30080689)

[3. KẾT QUẢ 21](#_Toc30080690)

[THẢO LUẬN 23](#_Toc30080691)

[PHỤ LỤC 24](#_Toc30080692)

# DANH MỤC HÌNH

[Hình 1.1. Ảnh minh họa đồng hồ hiển thị trên LCD 4](#_Toc30080723)

[Hình 1.2. Màn hình LCD 1602 5](#_Toc30080724)

[Hình 1.3. Ảnh mặt trước và mặt sau của moudle LCD I2C 6](#_Toc30080725)

[Hình 1.4. Board mạch Arduino UNO R3 7](#_Toc30080726)

[Hình 1.5. Module thời gian thực DS1307 10](#_Toc30080727)

[Hình 1.6. Hộp đựng sản phẩm 11](#_Toc30080728)

[Hình1.7. Giao diện Fritzing 12](#_Toc30080729)

[Hình 1.8. Giao diện phần mềm Arduino IDE cài đặt thời gian cho DS1307 13](#_Toc30080730)

[Hình 2.1. Mô phỏng mạch đồng hồ thời gian thực hiển thị trên LCD 14](#_Toc30080731)

[Hình 2.2. Sơ đồ nguyên lý của mạch 15](#_Toc30080732)

[Hình 2.3.Sơ đồ khối chương trình đồng hồ thời gian thực 16](#_Toc30080733)

[Hình 2.4. Hai hàm chuyển đổi dữ liệu về int 17](#_Toc30080734)

[Hình 2.5. Cài đặt thời gian cho DS1307 17](#_Toc30080735)

[Hình 2.6. Trình tự tạo hàm để lấy thời gian hiển thị trên LCD 18](#_Toc30080736)

[Hình 2.7. Khai báo biến các thư viện 18](#_Toc30080737)

[Hình 2.8. Các hàm setup sử dụng trong chương trình 19](#_Toc30080738)

[Hình 2.9. Các hàm loop sử dụng trong chương trình 20](#_Toc30080739)

[Hình 3.1.Đồng hồ LCD, thành phần liên kết của sản phẩm 21](#_Toc30080740)

[Hình 3.2. Sản phẩm hoàn thiện với thời gian hiển thị trên màn hình LCD 22](#_Toc30080741)DANH MỤC HÌNH

# **TÓM TẮT**

Đồng hồ LCD hiện đang kinh doanh trên thị trường có thể theo dõi được ngày, tháng, năm dương lịch, âm lịch và giờ giấc giúp chúng ta sắp xếp được thời gian biểu của mình một cách hợp lý. Yêu cầu của bài toán là thiết kế, thi công, lắp đặt sản phẩm đồng hồ thời gian thực hiển thị các thông số bào gồm: Ngày, tháng, năm giờ, phút, giây. Thời gian tự động cập nhật ngay cả khi không có nguồn điện cung cấp. Để thực hiện các công việc này cần sử dụng đến các thành phần, linh kiện điện tử như: module lập trình Arduino UNO R3, module thời gian thực DS1307, màn hình LCD và các thành phần khác bổ trợ khác như LED đơn, điện trở, nguồn cung cấp.

Quá trình thiết kế mạch in bắt đầu với phần mềm Fritzing để xây dựng sơ đồ nguyên lý và tiến hành mô phỏng trạng thái hoạt động của hệ thống. Sau đó sẽ tiến hành lắp đặt linh kiện lên bản mạch. Thuật toán được lập trình trên phần mềm Arduino IDE ver1.8.10.

Kết quả thực hiện cho thấy sản phẩm có thể hiển thị thời gian với các thông số như yêu cầu đặt ra, thời gian được cập nhật sau mỗi giây.

# GIỚI THIỆU



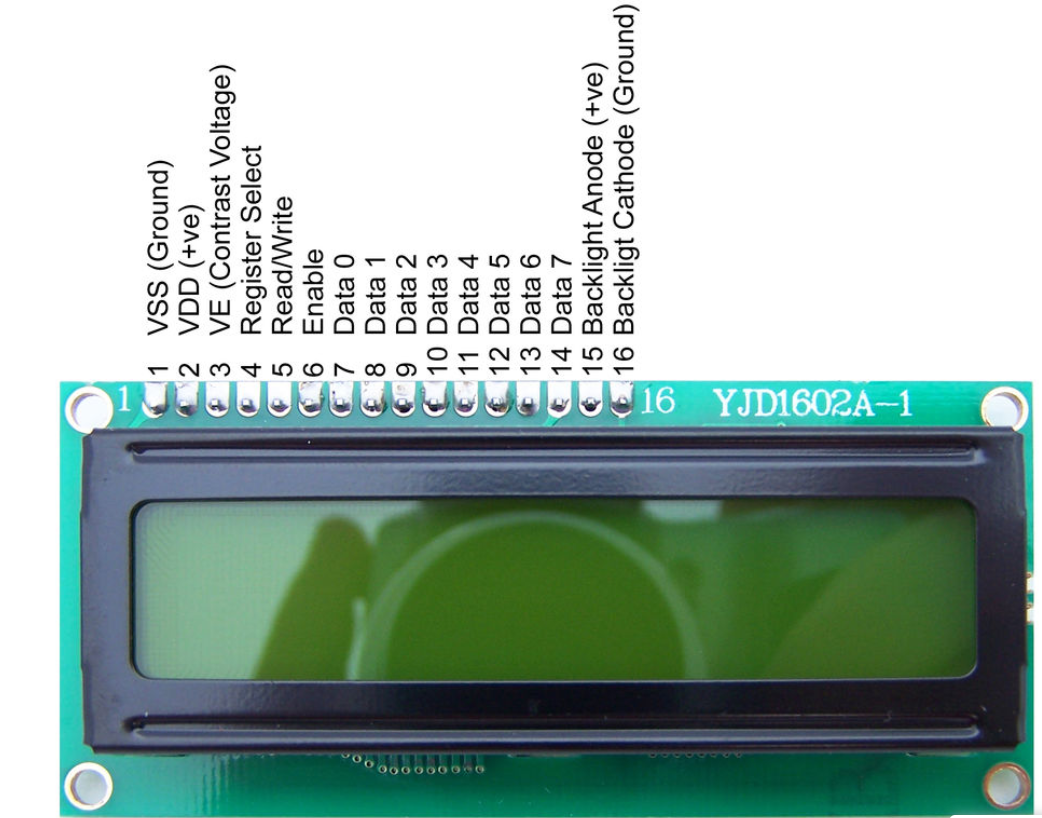
Hình 1.1. Ảnh minh họa đồng hồ hiển thị trên LCD

Hình 1. là một ví dụ về loại hình đồng hồ LCD . Thành phần cơ bản gồm màn hình LCD để cho biết thời gian như : ngày, tháng, năm, giờ, phút, giây và còn hiển thị thứ trong tuần. Để có thể tạo ra một đồng hồ LCD như trên cần vận dụng các kiến thức về lập trình thiết bị nhúng, cụ thể là kit Arduino, kiến thức về điện tử số và điện tử tương tự để tính toán các thành phần điện tử cơ bản và thực hiện các thao tác ghép nối các thành phần điện tử với nhau. Ngoài ra, một thành phần không thể thiếu chính là module thời gian thực DS1307 (https://www.electroschematics.com/ds1307-datasheet/) được dùng để đọc thời gian hiện hành và hiển thị lên màn hình LCD.

## Màn hình LCD 1602

Để hiển thị giờ, phút, ngày, tháng, năm cần có một màn hình LCD 1602.

LCD 1602 là một sản phẩm quen thuộc với những người mới học và muốn thực hiện các dự án về điện tử, lập trình. Với 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 dàng sử dụng hơn nếu đi kèm mạch chuyển tiếp I2C.



Hình 1.2. Màn hình LCD 1602

*Nguồn : (*[*https://sites.google.com/site/lcd24hgroup/lcd/tim-hieu-thong-so-ki-thuat-cua-lcd-1602*](https://sites.google.com/site/lcd24hgroup/lcd/tim-hieu-thong-so-ki-thuat-cua-lcd-1602)*)*

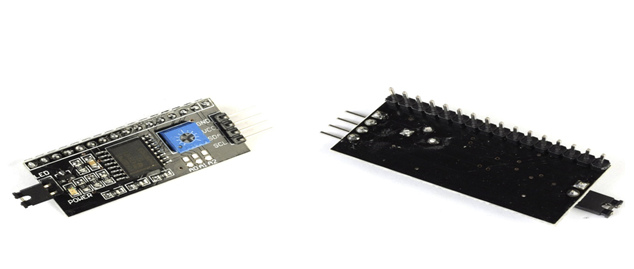
**Cấu tạo** :

* VSS: tương đương với GND – cực âm.
* VDD: tương đương với VCC – cực dương (5V).
* Constrast Voltage (Vo): điều khiển độ sáng màn hình.
* Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ.
* Read/Write (RW): Bạn sẽ đọc (read mode) hay ghi (write mode) dữ liệu? Nó sẽ phụ thuộc vào bạn gửi giá trị gì vào.
* Enable pin: Cho phép ghi vào LCD.
* D0 – D7: 8 chân dữ liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu bạn đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode).
* Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt/bật đèn màn hình LCD.

## Module LCD I2C

LCD có quá nhiều chân gây khó khăn trong quá trình kết nối và chiếm dụng nhiều chân của vi điều khiển?

Module chuyển đổi I2C cho LCD sẽ giải quyết vấn đề này, thay vì sử dụng 6 chân của vi điều khiển để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì module chuyển đổi chỉ cần sử dụng 2 chân (SCL, SDA) để kết nối. Module chuyển đổi I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 1602, LCD 2004, … ), kết nối với vi điều khiển thông qua giao tiếp I2C, tương thích với hầu hết các vi điều khiển hiện nay.



Hình 1.3. Ảnh mặt trước và mặt sau của moudle LCD I2C

*Nguồn : (*[*https://iotmaker.vn/mach-chuyen-doi-i2c-cho-lcd.html*](https://iotmaker.vn/mach-chuyen-doi-i2c-cho-lcd.html) *)*

## Board mạch Arduino Uno R3

Để để lấy được dữ liệu thời gian thực của RTC thì cần phải dùng đến 1 Mạch Arduino UNO R3.

Arduino giống như một máy tính nhỏ để người dùng có thể lập trình và thực hiện các dự án điện tử mà không cần phải có các công cụ chuyên biệt để phục vụ việc nạp code. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

**Cấu tạo :**

(1) USB: Cổng giao tiếp USB có 2 chức năng: cấp nguồn cho board mạch và truyền thông nối tiếp với máy tính trong việc nạp chương trình hay giao tiếp nối tiếp.

(2) Jack Power: Nguồn cấp cho board mạch Arduino. Có 2 loại nguồn có thể sử dụng được là nguồn xoay chiều (AC) tối đa 6V và nguồn một chiều (DC) tối đa 5V.

(3) RESET: Đặt lại trạng thái ngay khi nạp chương trình

(4) ISCP: Chân giao tiếp với USB, tín hiệu giao tiếp có thể giám sát từ đây.

(5) AREF và GND: GND hay chân Mass hay chân đất dùng để cấp nguồn 0V cho các modue khác có kết nối với Arduino. Trong khi chân AREF được dùng để phối hợp với các chân Analog Input (11) để điều chỉnh dải điện áp đầu vào cho ADC 10 bit.

(6) DIGITAL Input/Output: 12 chân tín hiệu số được dùng làm đầu vào hoặc đầu ra tại mỗi thời điểm.

(7) Serial Transmition: 2 chân RXD và TXD được dùng trong truyền thông nối tiếp. Khi 2 chân này cũng có thể được cấu hình làm đầu vào hoặc đầu ra tín hiệu số.

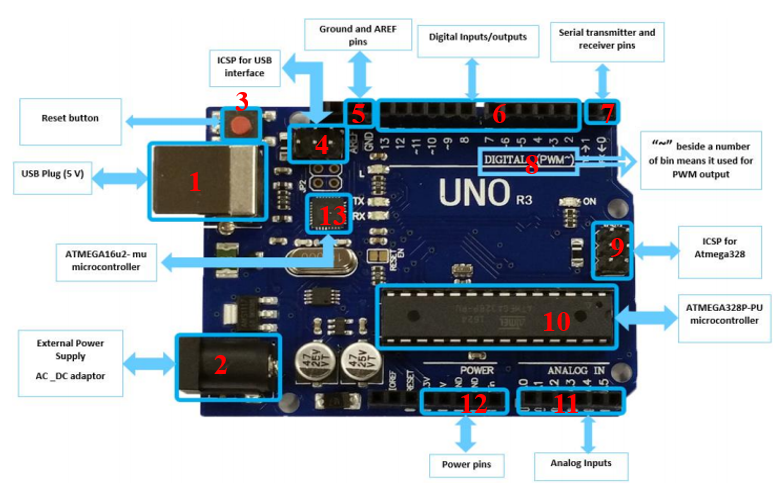
(8) PWM (Pulse Width Modulation): Chân băm xung có ký hiệu là ~ dùng để tạo ra chuỗi xung vuông trong điều khiển tốc độ động cơ hay hiển thị LED sáng dần hay tối dần.

(9) ICSP: Giao tiếp với Atmega328.

(10) IC lập trình ATMEGA16u2

(11) ANALOG Input: Tín hiệu chuyển đổi ADC được đưa vào các chân này. Trong trường hợp khác các chân này cũng có thể sử dụng để làm đầu vào/ra tín hiệu số.

(12) POWER: Cung cấp nguồn 5V, GND cho các thành phần mở rộng.



Hình 1.4. Board mạch Arduino UNO R3

*(Nguồn:* [*http://www.fecegypt.com/uploads/dataSheet/1522237550\_arduino%20uno%20r3.pdf*](http://www.fecegypt.com/uploads/dataSheet/1522237550_arduino%20uno%20r3.pdf)*)*

**Bảng 1.3. THÔNG SỐ CỦA ARDUINO UNO R3**

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dùng | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 32 KB, trong đó 0.5KB bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1KB (ATmega328) |

## Module thời gian thực DS1307

Đồng hồ hiển thị trên LCD cần module thời gian thực để cung cấp dữ liệu ngày, giờ cho Aruidno.

Đồng hồ thời gian thực DS1307 là chip thời gian thực hay RTC (Read time clock). Đây là một IC tích hợp cho thời gian bởi vì tính chính xác về thời gian tuyệt đối cho thời gian : Thứ, ngày, tháng, năm, giờ, phút, giây. Chip này có 7 thanh ghi 8 bit mỗi thanh ghi này chứa : Thứ , ngày, tháng, năm, giờ , phút, giây. Ngoài ra DS1307 còn chứa 1 thanh ghi điều khiển ngõ ra phụ và 56 thanh ghi trống các thanh ghi này có thể dùng như là RAM. DS1307 được đọc thông qua chuẩn truyền thông I2C nên do đó để đọc được và ghi từ DS1307 thông qua chuẩn truyền thông này. Do nó được giao tiếp chuẩn I2C nên cấu tạo bên ngoài nó rất đơn giản.

**Cấu tạo :**

X1 và X2 là đầu vào dao động cho DS1307. Cần dao động thạch anh 32.768Khz.[separator].

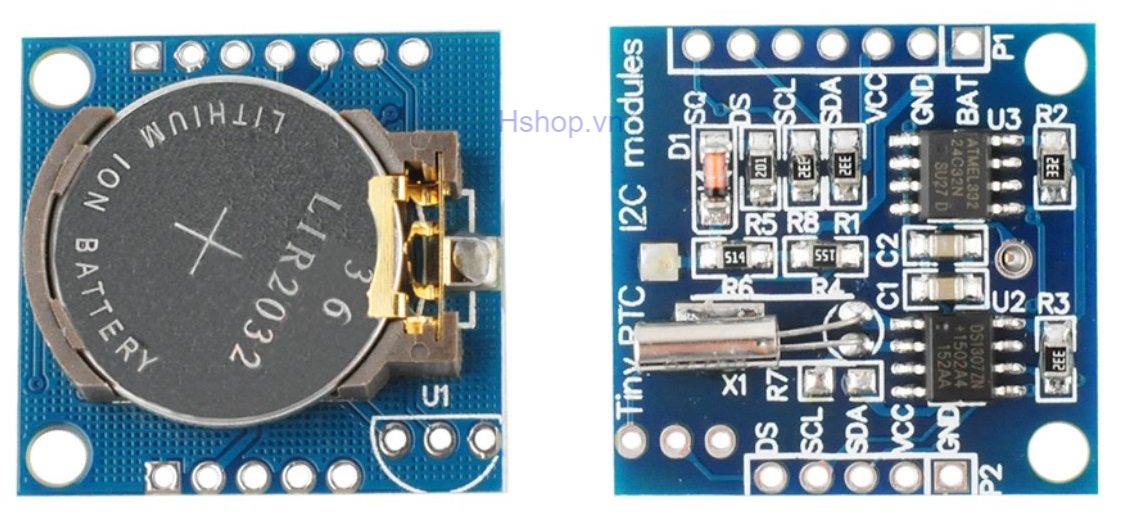
Vbat là nguồn nuôi cho chip. Nguồn này từ ( 2V- 3.5V) ta lấy pin có nguồn 3V. Đây là nguồn cho chip hoạt động liên tục khi không có nguồn Vcc mà DS1307 vẫn hoạt động theo thời gian.

VCC là nguồn cho giao tiếp I2C. Điện áp cung cấp là 5V chuẩn và được dùng chung với vi xử lý. Nếu mất nguồn Vcc mà Vbat vẫn có nguồn thì DS1307 vẫn hoạt động bình thường nhưng mà không ghi và đọc được dữ liệu.

GND là nguồn Mass chung cho cả Vcc và Vbat.

SQW/OUT là một ngõ ra phụ tạo xung dao động (xung vuông) ) được sử dụng cho các mục đích tạo xung, bộ đếm cho các thiết bị khác.

SCL (serial clcok) và SDA(Serial data) là hai bus dữ liệu của DS1307. Thông tin truyền và ghi đều được truyền qua 2 đường truyền này theo chuẩn I2C.



Hình 1.5. Module thời gian thực DS1307

*Nguồn : (http ://* [*https://hshop.vn/products/mach-thoi-gian-thuc-rtc-ds1307*](https://hshop.vn/products/mach-thoi-gian-thuc-rtc-ds1307) *)*

**Bảng 1.4. Bảng địa chỉ thanh ghi giá trị thời gian của DS1307**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Địa chỉ** | **BIT 7** | **BIT 6** | **BIT 5** | **BIT 4** | **BIT 3** | **BIT 2** | **BIT 1** | **BIT 0** | **Hàm** | **Phạm vi** |
| **00h** | CH | 10 Seconds | | | Seconds | | | | Seconds | 00-59 |
| **01h** | 0 | 10 Minutes | | | Minutes | | | | Minutes | 00-59 |
| **02h** | 0 | 12 | 10 Hour | 10  Hour | Hour | | | | Hour | 1-12 +AM/PM  00-23 |
| 24 | PM/AM |
| **03h** | 0 | 0 | 0 | 0 | 0 | DAY | | | Day | 01-07 |
| **04h** | 0 | 0 | 10 Date | | Date | | | |  | 01-31 |
| **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 |  |

## Một số linh kiện trang trí

Nhằm để trang trí cho đồng hồ không bị đơn điệu thì LED đơn cũng là một giải pháp độc đáo giành cho sản phẩm này. Đi kèm theo đó là những con điện trở giúp led không bị quá tải về điện áp để LED sáng bình thường.

Thiết kế hộp đựng sản phẩm :

**

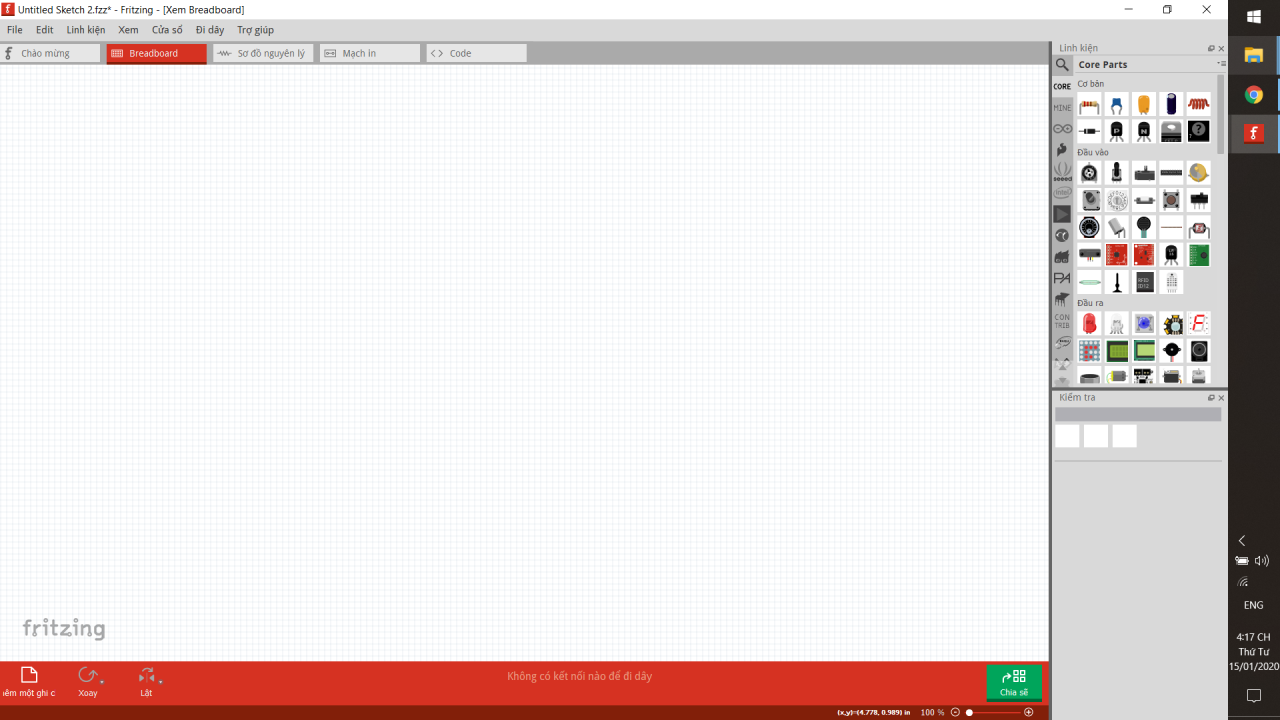
Hình 1.6. Hộp đựng sản phẩm

## Phần mềm Fritzing

Trước khi thi công mạch sản phẩm thực tế cần trải qua quá trình mô phỏng mạch để kiểm chứng thuật toán cũng như hoạt động logic của linh kiện điện tử. Fritzing là bộ công cụ chuyên về mô phỏng mạch điện tử chạy trên môi trường Windows.

Quan trọng nhất là thanh thư viện nơi chứa các linh kiện và vùng làm việc dùng để vẽ mạch nguyên lý (Hình 1.6.1). Ngoài ra, các nút mô phỏng được sử dụng cho việc mô phỏng nguyên lý hoạt động (Hình 1.6.1. bên dưới cùng).

Thanh tiêu đề



Thanh công cụ

Vùng thiết kế

Thư viện linh kiện cần lấy

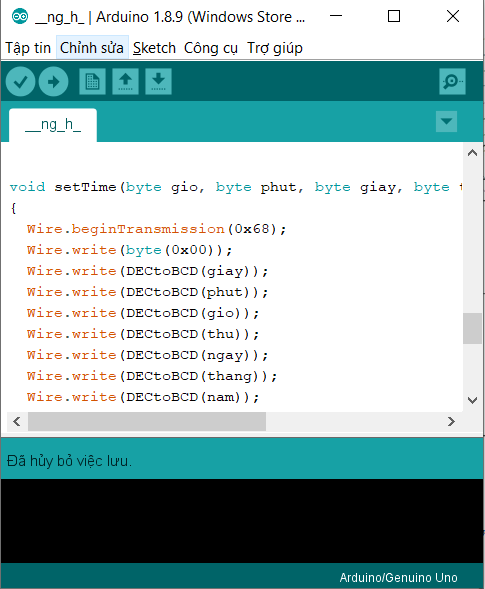
Hình1.7. Giao diện Fritzing

## Phần mềm Arduino IDE

Để giúp cho board mạch Arduino hoạt động thì cần dùng đến phần mềm và để hỗ trợ cho việc ấy thì sử dụng đến Arduino IDE.

Arduino IDE (Arduino Integrated Development Environment) là một chương trình phần mềm mã nguồn mở cho phép người dùng viết và tải lên mã lên vi điều khiển. Ứng dụng này hoạt động được cả 3 môi trường Windows, Linux và MacOS, mã nguồn của Arduino IDE được viết bởi C/C++ và tương thích hầu hết các board Arduino. Chương trình viết trên IDE có thể là C hoặc C++, sau khi biên dịch được nạp trực tiếp lên board mạch lập trình thông qua cổng USB.

Hình 1.7. là ví dụ về giao diện của Arduino IDE được sử dụng để cài đặt thời gian cho DS1307.



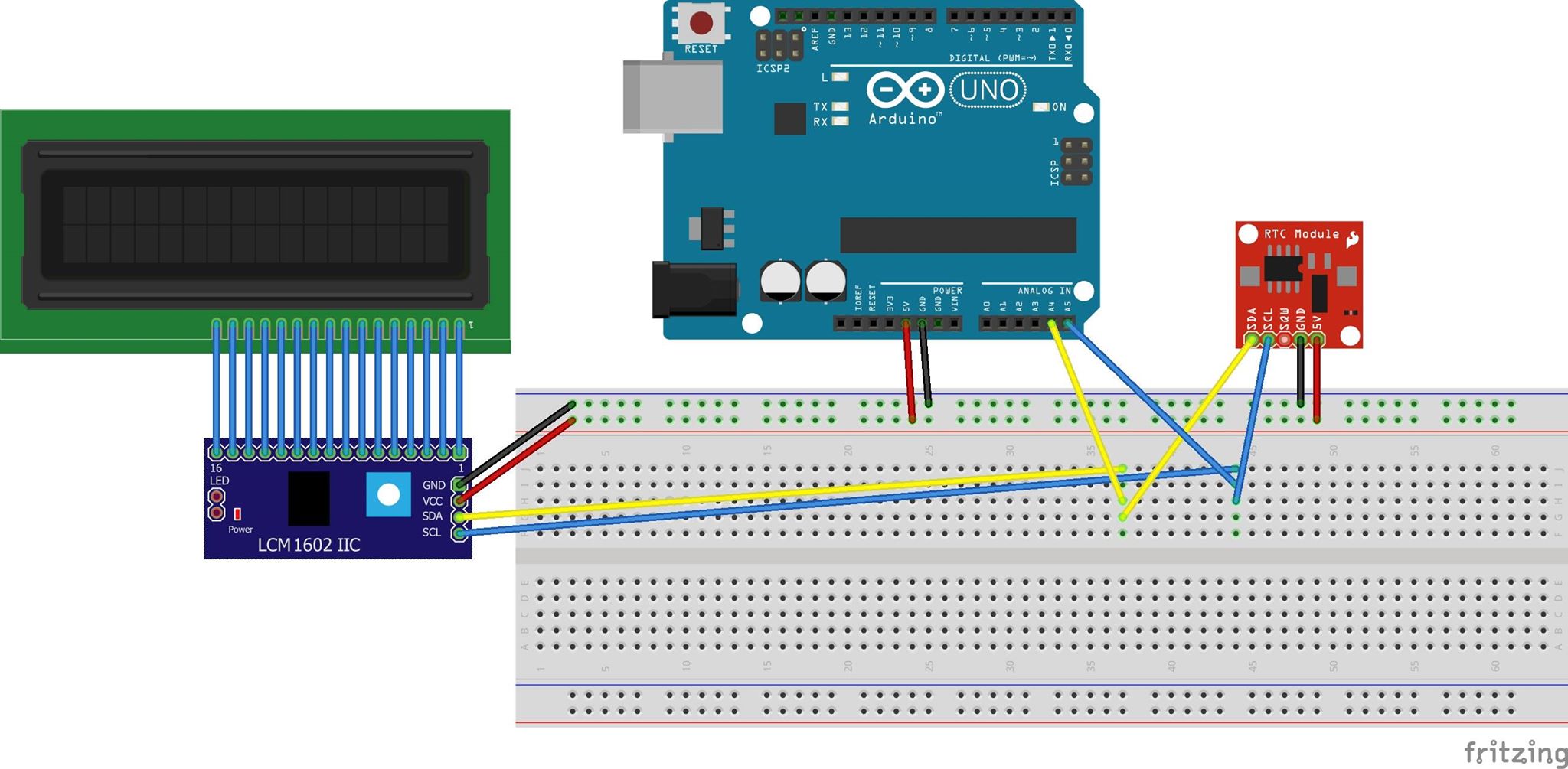
Hình 1.8. Giao diện phần mềm Arduino IDE cài đặt thời gian cho DS1307

# PHƯƠNG PHÁP NGHIÊN CỨU

Như đã đề cập, đồng hồ thời gian thực hiển thị trên LCD bắt đầu với những linh kiện hiển thị như LCD, kit Arduino, module cung cấp dữ liệu thời gian. Trong đó module thời gian thực DS1307 có nhiệm vụ cung cấp ngày giờ cho kit Arduino đọc dữ liệu, Arduino truyền dữ liệu hiển thị lên màn hình LCD. Các linh kiện và thiết bị cùng nhau phối hợp để tạo nên một sản phẩm hoàn chỉnh.

## Mô phỏng nguyên lý hoạt động bằng phần mềm fritzing

Để tiến hành làm sản phẩm trước hết cần phải sử dụng đến phầm mềm mô phỏng để giúp cho kiểm tra việc các chân nối để tránh các trường hợp những sai xót không đáng có. Đầu tiên ta sử dụng phần mềm fritzing để mô phỏng các linh kiện và cho mạch hoạt động.



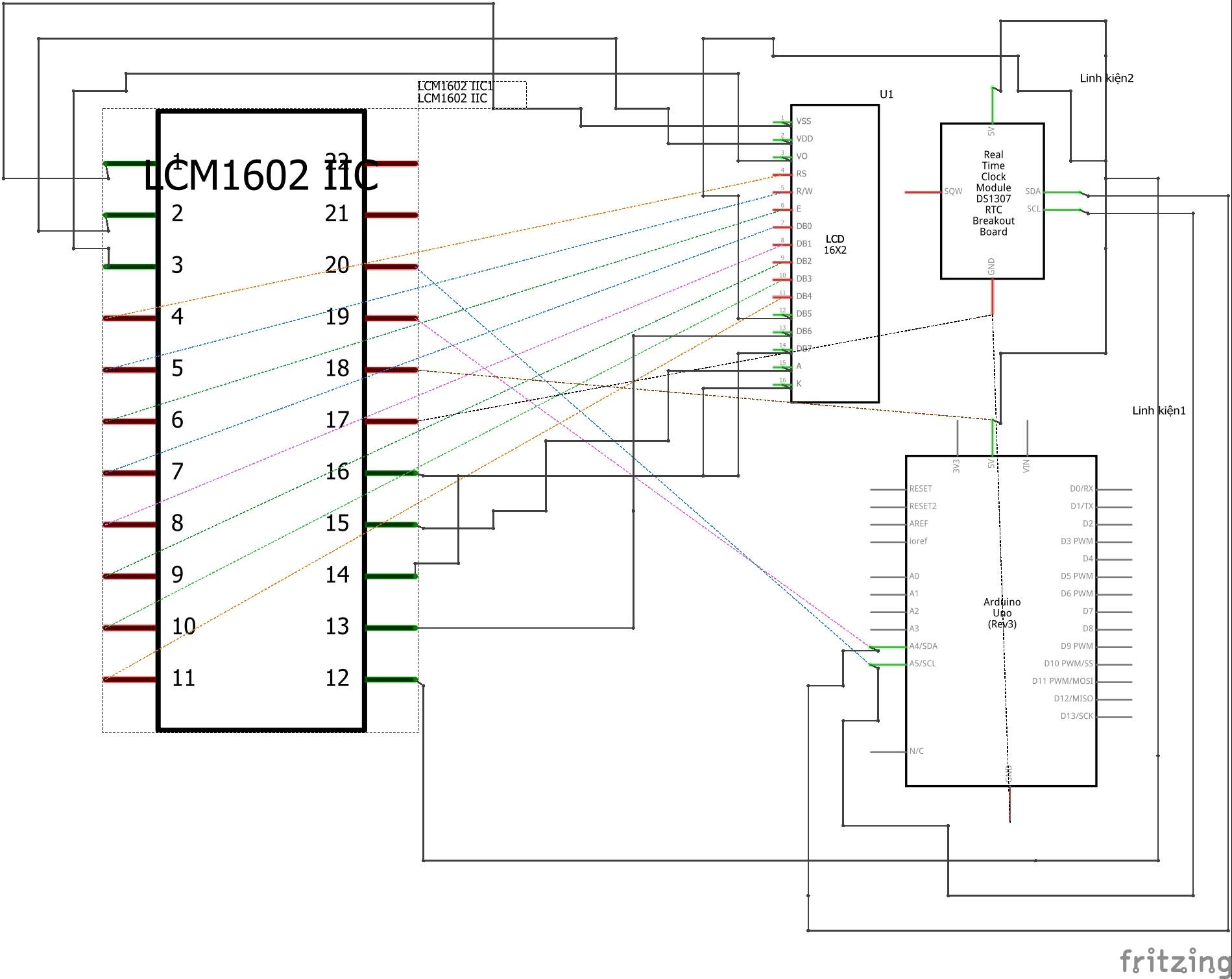
Arduino Uno R3

Module LCD I2C

Module DS1307

LCD 1602

Hình 2.1. Mô phỏng mạch đồng hồ thời gian thực hiển thị trên LCD



Hình 2.2. Sơ đồ nguyên lý của mạch

Trong hình 2.1.1. màn hình LCD dược dùng để hiển thị các thành phần của thời gian như : Giờ, Phút, Giây, Thứ, Ngày, Tháng, Năm. Mạch chuyển đổi I2C (kết nối trực tiếp với 16 chân của LCD) để tiết kiệm được các chân của mạch Arduino UNO R3. I2C có 2 chân SDA và SCL, trong đó SCL được sử dụng đồng bộ hóa việc truyền dữ liệu giữa Arduino UNO R3 với LCD 1302; chân SDA dùng cho việc truyền dữ liệu data. Module thời gian thực (khung màu đỏ) được dùng cho mô phỏng thời gian của đồng hồ. Board Arduino (màu xanh) là module dùng để lập trình mô phỏng.

Các bước tiến hành mô phỏng như sau:

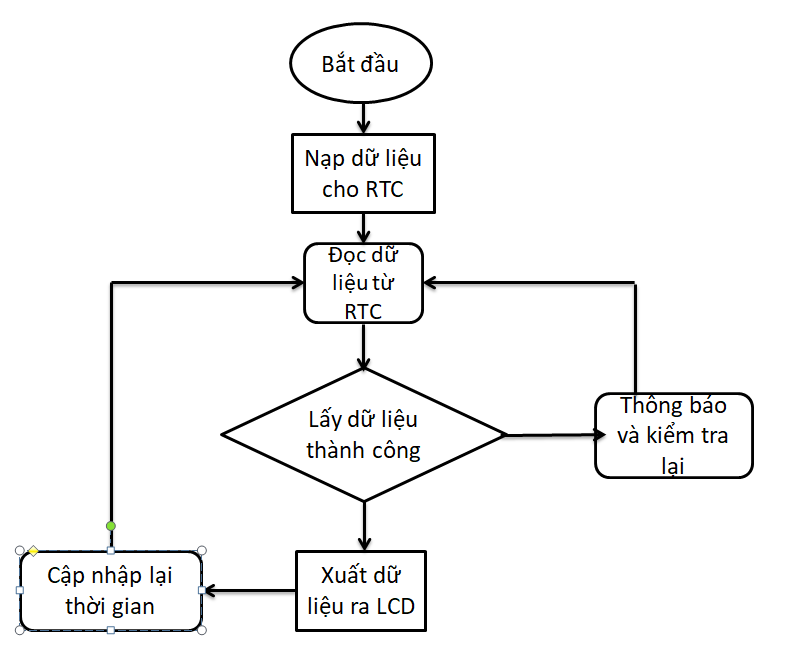
Bước 1 : Hoàn thiện việc lấy linh kiện và nối dây liên kết.

Bước 2 : Lập trình cho kit Arduino bằng phần mềm Arduino IDE.

Bước 3 : Nạp code cho Arduino bằng cách D\_Click lên biểu tượng  và nhấn  lựa chọn tập tin để thực thi.

Bước 4 : Nhấn  để thực hiện mô phỏng mạch.

## Lập trình

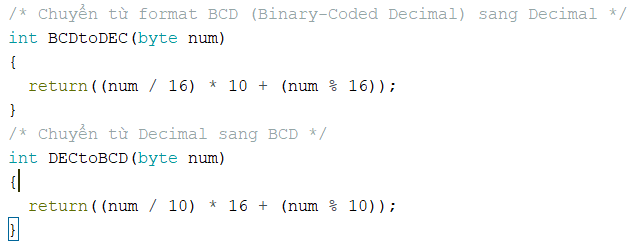


Hình 2.3.Sơ đồ khối chương trình đồng hồ thời gian thực

### Giao tiếp DS1307 (RTC) với Arduino

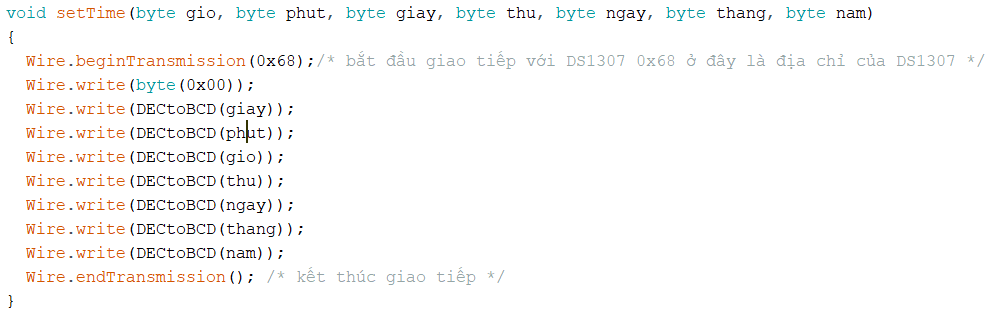
Đầu tiên muốn giao tiếp với DS1307 ở đây không dùng thư viện có sẵn vậy bên cần thư viện Wire của Arduino <Wire.h>

Để chuyển từ hệ số BCD (Binary-Coded Decimal) sang Decimal thì có hai hàm trả về dữ liệu INT như sau:



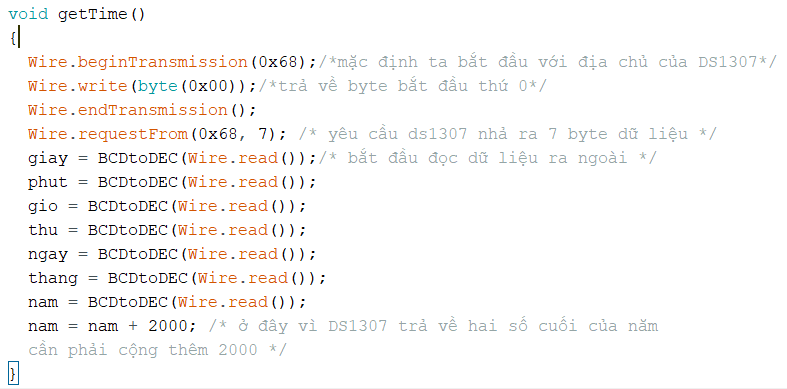
Hình 2.4. Hai hàm chuyển đổi dữ liệu về int

Sau khi được hai hàm như trên bắt đầu viết hàm cài đặt thời gian cho DS1307 trong hàm nay chuyển dữ liệu Binary Coded Decimal sang Decimal:



Hình 2.5. Cài đặt thời gian cho DS1307

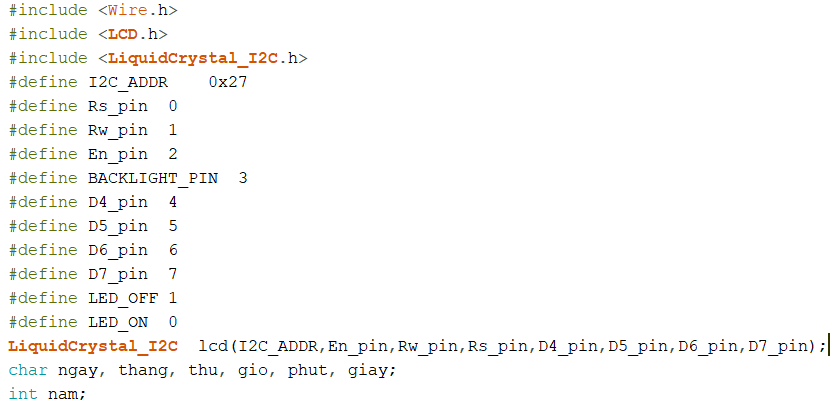
Khi cài đặt xong rồi cần tạo một hàm để lấy thời gian ra để hiện thị trên LCD như sau:



Hình 2.6. Trình tự tạo hàm để lấy thời gian hiển thị trên LCD

### Hiển thị thời gian trên LCD

1. ***Khái báo biến và các thư viện:***



Hình 2.7. Khai báo biến các thư viện

Khai báo thư viện Wire để thao tác với DS1307, thư viện LDC và LiquiCrytal\_I2C để thao tác trên màn hình, đặt mặc định các chân của màn hình LCD và địa chỉ cho LCD là 0x27 giống như địa chỉ của DS1307 là 0x68.

Khai báo các biến để cho các biến này chạy vào các hàm ở trên có lần lượt các biến char gồm ngày, tháng, thư, giờ, phút, giây. Riêng năm hiện tại đã là năm 2020 nên dùng biến int là hợp lí nhất.

1. ***Hàm Setup:***



Hình 2.8. Các hàm setup sử dụng trong chương trình

1. ***Hàm loop:***

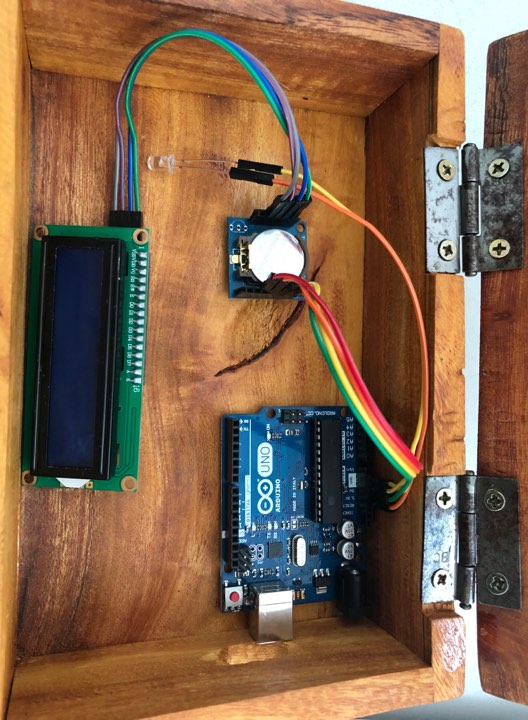


Hình 2.9. Các hàm loop sử dụng trong chương trình

Đầu tiên bật sáng màn hình lên sau khi hiển thị lời chào, tiếp theo dùng hàm getTime để lấy thời gian, bắt đầu sử dụng mảng buff in ra màn hình bằng hàm sprintf ở đây chung ta in ra “GIO: %02d:%02d:%02d” lần lượt là giờ phút giây, %02d ở đây là định dạng kiểu dữ liệu, tiếp theo set con trỏ về vị trí đầu tiên của dòng đầu tiên bằng hàm lcd.setCursor(1,1); , sao đó in ra màn hình bằng hàm lcd.print(buff), ở đây có chút vấn đề là DS1307 sẽ hiểu dữ liệu thứ theo thứ tự là 1 đến 7, nên ngày chủ nhật sẽ là 1 nếu in thứ 1 thì người dùng sẽ không hiểu nên cần có một hàm if else ở đây nếu dữ liệu là 1 thì in ra là CN còn lại là thứ như bình thường kết thúc hàm loop và kết thúc chương trình.

# KẾT QUẢ

Sau khi tiến hành nối dây cho các thành phần,linh kiện và phụ tùng cho vào hộp đựng sản phẩm kết quả được như hình 3.1.



Hình 3.1.Đồng hồ LCD, thành phần liên kết của sản phẩm

Chương trình nạp vào board mạch Arduino, thời gian thực được hiển thị lên màn hình LCD theo thứ tự từ trái sang phải từ trên xuống dưới ứng với giờ, phút, giây, thứ, ngày, tháng, năm. Hình 3.2. đồng hồ thời gian thực ứng với 15 giờ 51 phút 59 giây, thứ 4, ngày 15 tháng 01 năm 2020.



Hình 3.2. Sản phẩm hoàn thiện với thời gian hiển thị trên màn hình LCD

# **THẢO LUẬN**

Sau thời gian nổ lực học tập, tìm hiểu dưới sự hướng dẫn nhiệt tình của giáo viên hướng dẫn : Ths. Đoàn Vũ Thịnh, sản phẩm hoàn thiện với các chức năng đã được trình bày ở phần phương pháp. Toàn bộ quy trình thiết kế, thực hiện sản phẩm được hoàn thiện trong thời gian 5 tuần của đợt thực tập cơ sở. Sản phẩm đã hoàn thành đúng thời hạn quy định, về cơ bản đáp ứng được các yêu cầu đặt ra nhưng chương trình còn một số lỗi nhỏ do thiết bị không đáp ứng được yêu cầu đã đặt ra.

# PHỤ LỤC

/\* Khai báo các thư viện \*/

#include <Wire.h>

#include <LCD.h>

#include <LiquidCrystal\_I2C.h>

/\* Thiết lập địa chỉ mặc định của màn hình là 0x27 \*/

#define I2C\_ADDR 0x27

/\* Thiết lập các chân của màn hình \*/

#define Rs\_pin 0

#define Rw\_pin 1

#define En\_pin 2

#define BACKLIGHT\_PIN 3

#define D4\_pin 4

#define D5\_pin 5

#define D6\_pin 6

#define D7\_pin 7

#define LED\_OFF 1

#define LED\_ON 0

LiquidCrystal\_I2C lcd(I2C\_ADDR,En\_pin,Rw\_pin,Rs\_pin,D4\_pin,D5\_pin,D6\_pin,D7\_pin);

/\* tạo các biến char cho các thành phần ngày, tháng, thứ, giờ, phút, giây \*/

char ngay, thang, thu, gio, phut, giay;

/\* tạo biến năm kiểu int vì năm 2020 kiểu int là hợp lí \*/

int nam;

/\* hàm setup \*/

void setup()

{

lcd.begin(16,2); /\* khởi động mà hình LCD 16 2 ở đây là 16

và hai dòng \*/

/\* bật đèn nền cho màn hình LCD \*/

lcd.setBacklightPin(BACKLIGHT\_PIN,POSITIVE);

lcd.setBacklight(LED\_ON);

lcd.backlight();

/\* chỉnh vị trí con trỏ thụt vào 1 ô ở hàng thứ nhất \*/

lcd.setCursor(1,0);

/\*Hiển thị lời chào chúng ta có thể hiện thị gì ở đây

cũng đc

\*/

lcd.print("DONG HO DS1307");

delay(1500); /\* delay 1,5s để chúng ta có thể nhìn rõ lời chào \*/

lcd.clear();

/\* bắt đầu cài giờ vào ds1307 theo thời gian thực tế theo hàm setTime

ở trên ta nhập theo thứ tự giờ, phút, giây, thứ, ngày, tháng, năm

\*/

setTime(15,50,30,4,15,01,20);

}

/\* Chuyển từ format BCD (Binary-Coded Decimal) sang Decimal \*/

int BCDtoDEC(byte num)

{

return((num / 16) \* 10 + (num % 16));

}

/\* Chuyển từ Decimal sang BCD \*/

int DECtoBCD(byte num)

{

return((num / 10) \* 16 + (num % 10));

}

void setTime(byte gio, byte phut, byte giay, byte thu, byte ngay, byte thang, byte nam)

{

Wire.beginTransmission(0x68);/\* bắt đầu giao tiếp với DS1307 0x68 ở đây là địa chỉ của DS1307 \*/

Wire.write(byte(0x00));

Wire.write(DECtoBCD(giay));

Wire.write(DECtoBCD(phut));

Wire.write(DECtoBCD(gio));

Wire.write(DECtoBCD(thu));

Wire.write(DECtoBCD(ngay));

Wire.write(DECtoBCD(thang));

Wire.write(DECtoBCD(nam));

Wire.endTransmission(); /\* kết thúc giao tiếp \*/

}

void getTime()

{

Wire.beginTransmission(0x68);/\*mặc định ta bắt đầu với địa chủ của DS1307\*/

Wire.write(byte(0x00));/\*trả về byte bắt đầu thứ 0\*/

Wire.endTransmission();

Wire.requestFrom(0x68, 7); /\* yêu cầu ds1307 nhả ra 7 byte dữ liệu \*/

giay = BCDtoDEC(Wire.read());/\* bắt đầu đọc dữ liệu ra ngoài \*/

phut = BCDtoDEC(Wire.read());

gio = BCDtoDEC(Wire.read());

thu = BCDtoDEC(Wire.read());

ngay = BCDtoDEC(Wire.read());

thang = BCDtoDEC(Wire.read());

nam = BCDtoDEC(Wire.read());

nam = nam + 2000; /\* ở đây vì DS1307 trả về hai số cuối của năm

cần phải cộng thêm 2000 \*/

}

/\* ở đây ta tạo một mảng bất kì để in ra màn hình \*/

char buff [20];

void loop()

{

/\* khởi động đèn nền cho LCD \*/

lcd.backlight();

/\* sử dụng hàm getTime để lấy thời gian từ DS1307 \*/

getTime();

/\* In ra hàng đầu tiên của LCD là hàng Giờ \*/

sprintf(buff,"GIO: %02d:%02d:%02d",gio,phut,giay);

lcd.setCursor(0,0);

lcd.print(buff);

/\* Dùng If else để hiển thị CN thay vì là số 1 \*/

if(thu==1){

sprintf(buff,"CN %02d/%02d/%02d",ngay,thang,nam);

lcd.setCursor(1,1);

lcd.print(buff);

}

else

{

sprintf(buff,"THU%2d %02d/%02d/%02d",thu,ngay,thang,nam);

lcd.setCursor(0,1);

lcd.print(buff);

}

}

/\* Chuyển từ format BCD (Binary-Coded Decimal) sang Decimal \*/

int BCDtoDEC(byte num)

{

return((num / 16) \* 10 + (num % 16));

}

/\* kết thúc chương trình \*/