TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**



**HIỂN THỊ THỜI GIAN THỰC LÊN MÀN HÌNH LCD BẰNG MODULE THỜI GIAN THỰC RTC DS1307**

**GVHD : Ths. ĐOÀN VŨ THỊNH**

**SVTH : Phạm Ngọc Phú – MSSV: 59131878**

**Nguyễn Trung Khiêm – MSSV: 59131062**

**Lớp : 59CNTT-1**

Khánh Hòa, 17 tháng 01 năm 2020

# MỤC LỤC

[MỤC LỤC 1](#_Toc30146424)

[DANH MỤC HÌNH 2](#_Toc30146425)

[DANH MỤC BẢNG 3](#_Toc30146426)

[TÓM TẮT 4](#_Toc30146427)

[1.GIỚI THIỆU 5](#_Toc30146428)

[1.1. Màn hình LCD 1602 5](#_Toc30146429)

[1.2. Moudle LCD I2C 7](#_Toc30146430)

[1.3. Moudle thời gian thực DS1307 8](#_Toc30146431)

[1.4. Arduino UNO R3 12](#_Toc30146432)

[1.5. Phần mềm Arduino IDE 13](#_Toc30146433)

[1.7. Phần mềm Fritzing 14](#_Toc30146434)

[2.PHƯƠNG PHÁP NGHIÊN CỨU 15](#_Toc30146435)

[2.1. Lập trình 15](#_Toc30146436)

[2.1.1. Giao tiếp Arduino với RTC DS1307 15](#_Toc30146437)

[2.1.2. Hiển thị thời gian trên LCD 16](#_Toc30146438)

[2.2. Mô phỏng mạch và sơ đồ nguyên lý trên phần mềm Fritzin 17](#_Toc30146439)

[3. KẾT QUẢ 19](#_Toc30146440)

[4. THẢO LUẬN 21](#_Toc30146441)

[PHỤ LỤC 22](#_Toc30146442)

# DANH MỤC HÌNH

[Hình 1.1. Hình minh họa 5](#_Toc30146699)

[Hình 1.2. Màn hình LCD 1602 6](#_Toc30146700)

[Hình 1.3 : Mạch chuyển đổi I2C cho LCD 8](#_Toc30146701)

[Hình 1.4. LCD 1602 có tích hợp Module chuyển đổi I2C 8](#_Toc30146702)

[Hình 1.5. Module thời gian thực DS1307 9](#_Toc30146703)

[Hình 1.6. Module Arduino UNO R3 12](#_Toc30146704)

[Hình 1.7. giao diện phần mềm Arduino IDE 14](#_Toc30146705)

[Hình 1.8. Phầm mềm Fritzing 14](#_Toc30146706)

[Hình 2.1. Sơ đồ khối chương trình đồng hồ thời gian thực hiển thị trên LCD 15](#_Toc30146707)

[Hình 2.2. Trình tự các bước khởi tạo RTC 16](#_Toc30146708)

[Hình 2.3. Cài đặt thời gian cho DS1307 16](#_Toc30146709)

[Hình 2.4. Hiển thị thời gian cho LCD 17](#_Toc30146710)

[Hình 2.5. Mạch mô phỏng 17](#_Toc30146711)

[Hình 2.6. Sơ đồ nguyên lí 18](#_Toc30146712)

[Hình 3.1. Màn hình LCD, và các thành phần liên kết của sản phẩm 19](#_Toc30146713)

[Hình 3.2. Sản phẩm hoàn thiện với thời gian thực được hiển thị trên LCD 20](#_Toc30146714)

# DANH MỤC BẢNG

[Bảng 1.1. Cấu tạo màn hình LCD 1602 6](#_Toc30146915)

[Bảng 1.2. Thông số kĩ thuật 8](#_Toc30146916)

[Bảng 1.3. Bảng địa chỉ thanh ghi giá trị thời gian của DS1307 10](#_Toc30146917)

[Bảng 1.4. Bảng tầng số trạng thái chân RS1 và RS0 12](#_Toc30146918)

[Bảng 1.5. Thông số của Arduino UNO R3 13](#_Toc30146919)

[Bảng 2.1. Sơ đồ nối dây giữa Module RTC DS1307 với Arduino Uno R3 15](#_Toc30146920)

[Bảng 3.1. Cách nối dây giữa màn hình LCD với Arduino 19](#_Toc30146921)

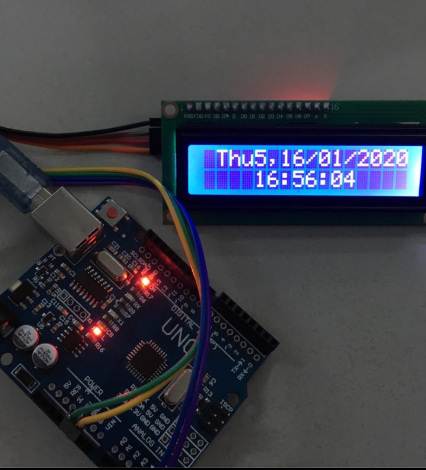
[Bảng 3.2. Cách nối dây giữa Module RTC DS 1307 với Arduino Uno R3 19](#_Toc30146922)

# 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.

# 1.GIỚI THIỆU

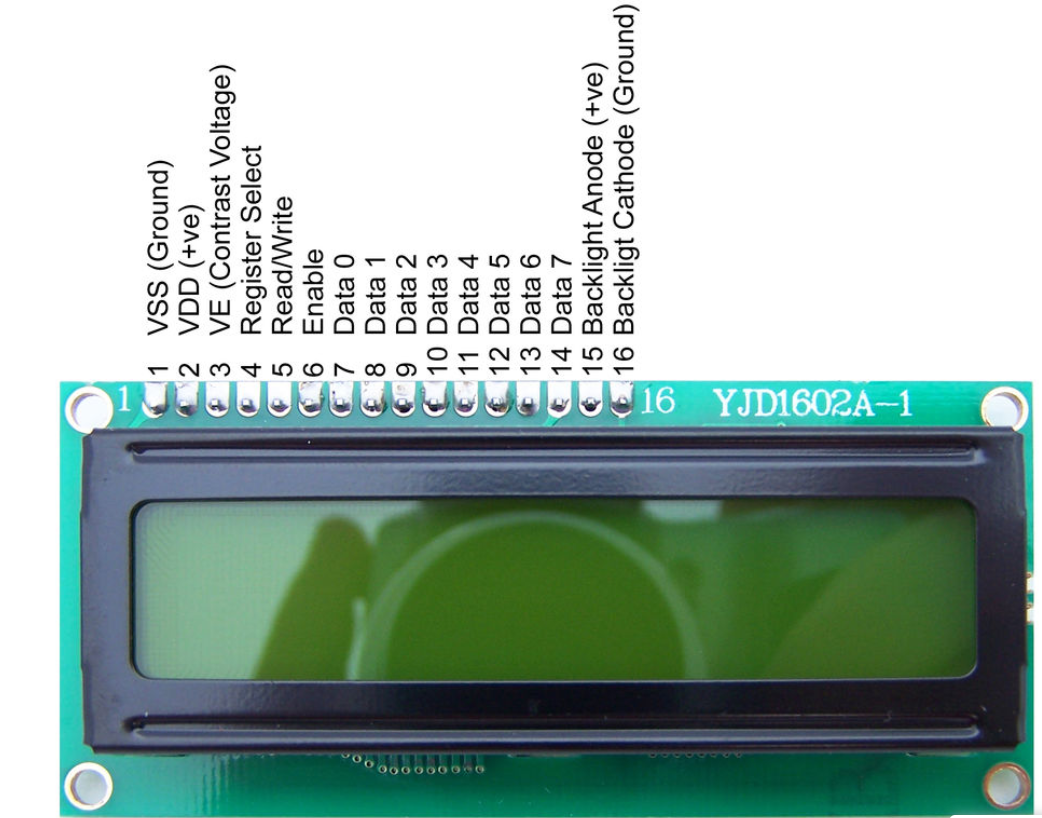


Hình 1.1. Hình minh họa

Hình 1.1 Là một ví dụ về loại hình đồng hồ số . 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 số 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 các thành phần LED.

## 1.1. Màn hình LCD 1602

LCD 1602 một sản phẩm quen thuộc với những người mới học và muốn thực hiện các dự án về điện tử, lập trình. Với khả hiển thị kí tự với 2 dòng trong đó mỗi dòng gồm 16 ký tự, với sự hiển thị như vậy thì có thể áp dụng được trong rất nhiều những ứng dụng như hiển thị thông tin, hiển thị trạng thái, hiển thị thông số… đặc biệt đối với đề tài này LCD còn được dùng để hiển thị thời gian.



Hình 1.2. Màn hình LCD 1602

**Cấu tạo:**

Bảng 1.1. Cấu tạo màn hình LCD 1602

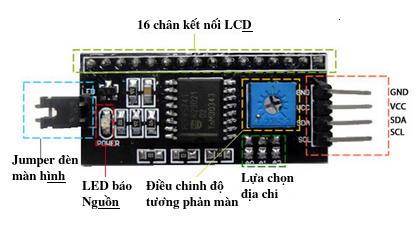
|  |
| --- |
|  |
| **Chân** | **Ký hiệu** | **Mô tả** |
| 1 | Vss | Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển |
| 2 | VDD | Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển |
| 3 | VEE | Điều chỉnh độ tương phản của LCD. |
| 4 | RS | Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi.  + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read)  + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD. |
| 5 | R/W | Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc. |
| 6 | E | Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.  + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.  + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp. |
| 7 - 14 | DB0 - DB7 | Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :  + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.  + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7 |
| 15 | - | Nguồn dương cho đèn nền |
| 16 | - | GND cho đèn nền |

## 1.2. Moudle LCD I2C

Module chuyển đổi I2C cho LCD sẽ giải quyết vấn đề thay vì sử dụng tối thiểu 6 chân của vi điều khiển để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì với module chuyển đổi bạn 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.

Bảng 1.2. Thông số kĩ thuật

|  |  |
| --- | --- |
| Điện áp hoạt động | 2.5-6V DC. |
| Hỗ trợ màn hình | LCD1602,1604,2004 (driver HD44780). |
| Giao tiếp | I2C |
| Địa chỉ mặc định | 0X27 |
| Kích thước | 41.5mm(L)x19mm(W)x15.3mm(H) |
| Trọng lượng | 5g |



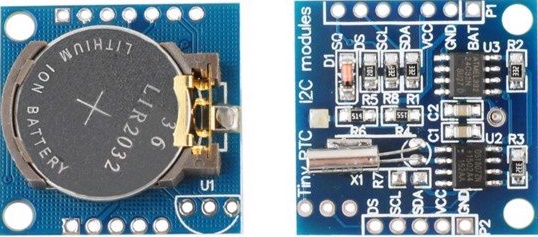
Hình 1.3 : Mạch chuyển đổi I2C cho LCD



Hình 1.4. LCD 1602 có tích hợp Module chuyển đổi I2C

## 1.3. Moudle thời gian thực DS1307

Những linh kiện đã đề cập ở trên vẫn chưa thể tạo thành thời gian cho đồng hồ vạn niên được, việc cho các LED sáng theo giờ cần phải cập nhật thời gian cho vi điều khiển. Module 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: Thứ, ngày,tháng, năm, giờ, phút, giây. DS1307 được sản xuất bởi hãng Dallas, 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 bộ nhớ RAM. DS1307 sử dụng chuẩn truyền thông I2C nên cấu tạo bên ngoài nó rất đơn giản.



Hình 1.5. Module thời gian thực DS1307

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

**Cấu tạo:**

X1 và X2 là đầu vào dao động cho DS1307 sử dụng thạch anh có tần số dao động là 32.768Khz. Thạch anh là linh kiện điện tử tạo giao động ổn định cho bộ điều khiển, thạch anh được sử dụng trong đa phần thiết bị điện tử vì nó rất ít bị ảnh hưởng bởi nhiệt độ hơn các mạch dao động RC hay RLC.

Vbat là nguồn nuôi cho DS1307, nguồn này có điện áp từ 2V- 3.5V. Nguồn nuôi (Pin CMOS) được dùng để cấp nguồn cho RTC hoạt động trong trường hợp module thời gian thực không được cấp nguồn VCC. Nguồn nuôi này có thể hoạt động đến 10 năm khi module đã tắt nguồn.

VCC là nguồn cung cấp cho giao tiếp I2C, điện áp cung cấp là 5V. Nếu có 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 đất (mass) chung cho cả Vcc và Vbat.

SQW/OUT là 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 và SDA, trong đó SCL (serial clock) 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.

I2C (Inter-Integrated Circuit) là một loại bus nối tiếp được phát triển bởi hãng sản xuất linh kiện điện tử Philips. I2C sử dụng hai đường truyền tín hiệu: một đường xung nhịp đồng hồ (SCL) và một đường dữ liệu (SDA). SCL và SDA luôn được kéo lên nguồn bằng một điện trở treo có giá trị 4,7 KΩ. Các chế độ hoạt động của I2C bao gồm: (1) Chế độ chuẩn (standard mode) hoạt động ở tốc độ 100 Kbit/s; (2) Chế độ tốc độ thấp (low-speed mode) hoạt động ở tốc độ 10 Kbit/s.

I2C sử dụng 7 bit để định địa chỉ, do đó trên một bus có thể đánh 128 địa chỉ (112 nút (node) và 16 địa chỉ dành cho mục đích riêng). Điểm mạnh của I2C chính là hiệu suất và sự đơn giản của nó: một khối điều khiển trung tâm có thể điều khiển cả một mạng thiết bị mà chỉ cần hai đường điều khiển. Ban đầu, loại bus này chỉ được dùng trong các linh kiện điện tử của Philips. Sau đó, do tính ưu việt và đơn giản của nó, I2C đã được chuẩn hóa và được dùng rộng rãi trong các module truyền thông nối tiếp của vi mạch tích hợp ngày nay. Vì Arduino đã có hỗ trợ sẵn điện trở treo ở đầu vào SDA và SCL nên khi kết nối với DS1307 không cần sử dụng đến trở treo bên ngoài nữa. Để có thể sử dụng giao tiếp I2C, thư viện Wire.h cần được thiết lập trong Arduino IDE.

Bảng 1.3. 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 |  | |

Trong bộ nhớ của DS1307 có tất cả 64 thanh ghi địa chỉ từ 0 đến 63 và được bắt đầu từ 0x00 đến 0x3F nhưng trong đó chỉ có 8 thanh ghi đầu là thanh ghi thời gian thực Bảng 1.5 là địa chỉ cụ thể cho các thành phần thời gian như sau: Giây, phút, giờ, thứ, ngày, tháng, năm bắt đầu từ địa chỉ 00h và kết thúc là địa chỉ 06h.

Thanh ghi Control dùng để điều khiển ngõ ra của chân SQW/OUT trong trường hợp RTC đóng vai trò là bộ cấp nguồn dao động cho các thiết bị khác.

(1) Thanh ghi giây (0x00): Bit 0 đến bit 3 (Bảng 1.5) dùng để mã hóa bộ đếm BCD hàng đơn vị của giây. Tiếp theo từ bit 4 đến bit 6 dùng để mã hóa BCD hàng chục của giây. Lý do số giây hàng đơn vị cần sử dụng đến 4 bit và số giây hàng chục chỉ cần 3 bit mã hóa là vì hàng chục chỉ hiển thị tối đa là số 5 (101) và số giây hàng đơn vị hiển thị tối đa số 9 (1001). Từ đó bit 7 của thanh ghi này luôn luôn được gán giá trị là 0 (mã lệnh là AND bit với giá trị 0x7F ngay khi khởi động chương trình).

(2) Thanh ghi phút (0x01): 4 bit thấp dùng để mã hóa BCD chữ số hàng đơn đơn vị và 3 bit cao để mã hóa số phút hàng chục (Bảng 1.5). Nhưng thanh ghi này có sự khác biệt với thanh ghi giây là bit 7 mặc định là 0 rồi nên không cần phải thực hiện thao tác như thanh ghi giây đối với bit 7 của thanh ghi này.

(3) Thanh ghi giờ (0x02): Bit 0 đến bit 3 để mã hóa BCD của chữ số hàng đơn vị của giờ và bit 4 đến bit 7 để mã hóa số giờ hàng chục (Bảng 1.5). Tuy nhiên có 2 chế độ để hiển thị số giờ: chế độ 24h và 12h nên được thiết lập bởi bit 6. Nếu bit 6 có giá trị là 0 (chế độ 24h) nên chỉ sử dụng bit 4 và bit 5 để mã hóa BCD chữ số hàng chục của giờ (do giá trị lớn nhất hiển thị số giờ là 23). Nếu bit 6 có giá trị bằng 1 (chế độ 12h) nên chỉ cần dùng duy nhất bit 4 để mã hóa BCD chữ số hàng trục của giờ (do giá trị lớn nhất của số hàng chục của giờ có thể hiển thị là 12). Ngoài ra, ở chế độ 12h, bit 5 được dùng để chỉ buổi sáng/AM (bit 5 có giá trị là 0) hay chiều/PM (bit 5 có giá trị là 1). Trong cả 2 chế độ 12h và 24h bit 7 luôn luôn có giá trị là 0.

(4) Thanh ghi thứ trong tuần (0x03): 3 bit thấp (bit 0 đến bit 2) để mã hóa BCD cho các thứ trong tuần (1 = Chủ nhật và 7 = thứ 7). Do đó, các bit cao từ 3 đến 7 mặc định là 0 (Bảng 1.5).

(5) Thanh ghi ngày (0x04): 4 bit thấp (bit 0 đến bit 3) dùng để mã hóa BCD ra chữ số hàng đơn vị của ngày trong tháng (từ 1 đến 9) và bit 4 và bit 5 mã hóa BCD cho số hàng chục của ngày trong tháng (từ 0 đến 3). Bit 6 và bit 7 mặc định bằng 0 (Bảng 1.5).

(6) Thanh ghi tháng (0x05): 4 bit thấp (từ bit 0 đến bit 3) mã hóa BCD hàng đơn vị của tháng (từ 1 đến 9). Bit thứ 4 để mã hóa BCD số hàng chục của tháng (hiển thị số 0 và 1). Các bit còn lại (bit 5 đến bit 7) mặc định là 0 (Bảng 1.5).

(7) Thanh ghi năm (0x06): Sử dụng 8 bit ( bit 0 đến bit 7) để mã hóa BCD của số năm. Do RTC chỉ hiển thị 2 số cuối cùng của năm (từ 00 đến 99). (Bảng 1.5)

(8) Thanh ghi điều khiển (0x07): Đây là thanh ghi điều khiển quá trình ghi đọc của DS1307 với giá trị 0x93.

(8) OUT: Chọn mức logic (1 hoặc 0) xuất ra tại chân SQW/OUT khi chức năng SQW không được kích hoạt (SQW có giá trị là 0).

(9) SQW: Tạo xung vuông trên chân SQW/OUT, tần số xung vuông được thiết lập bởi RS0, RS1.

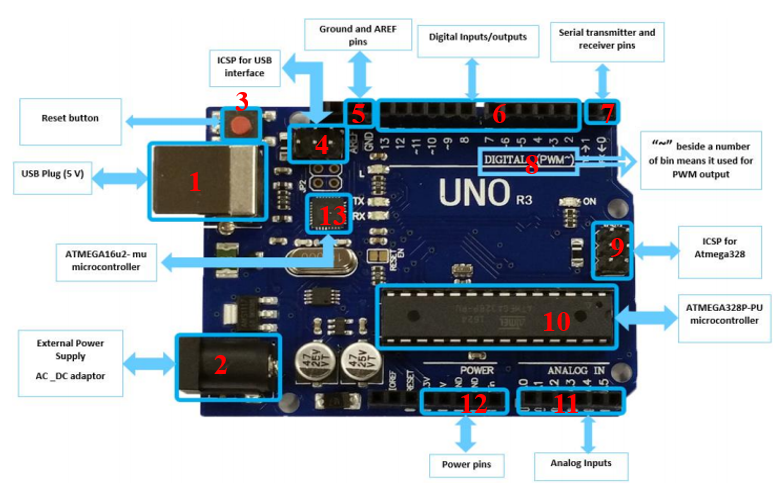
Bảng 1.4. Bảng tầng số trạng thái chân RS1 và RS0

|  |  |  |
| --- | --- | --- |
| RS1 | RS0 | Tần Số |
| 0 | 0 | 1Hz |
| 0 | 1 | 4.096KHz |
| 1 | 0 | 8.192KHz |
| 1 | 1 | 32,768KHz |

## 1.4. Arduino UNO R3

Để đọc dữ liệu từ module thời gian thực DS1307 và hiển thị lên LCD cần sử dụng Arduino UNO R3 ([**http://electronoobs.com/eng\_arduino\_tut31\_sch3.php**](http://electronoobs.com/eng_arduino_tut31_sch3.php)).

Arduino - 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.



Hình 1.6. Module 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)***)***

**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: 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.

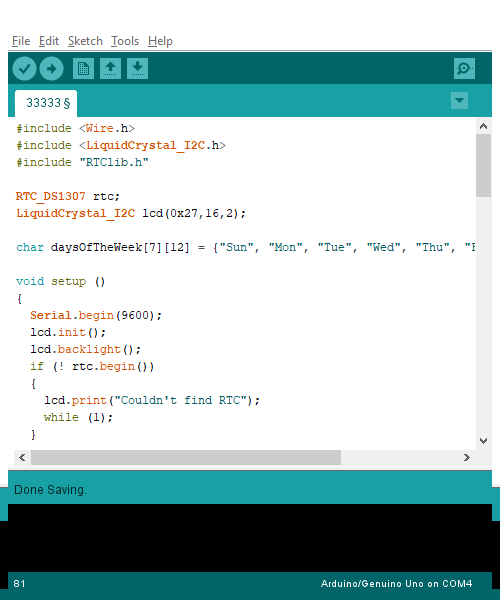
(12) POWER: Cung cấp nguồn 5V, GND cho các thành phần mở rộng.

Bảng 1.5. 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) |

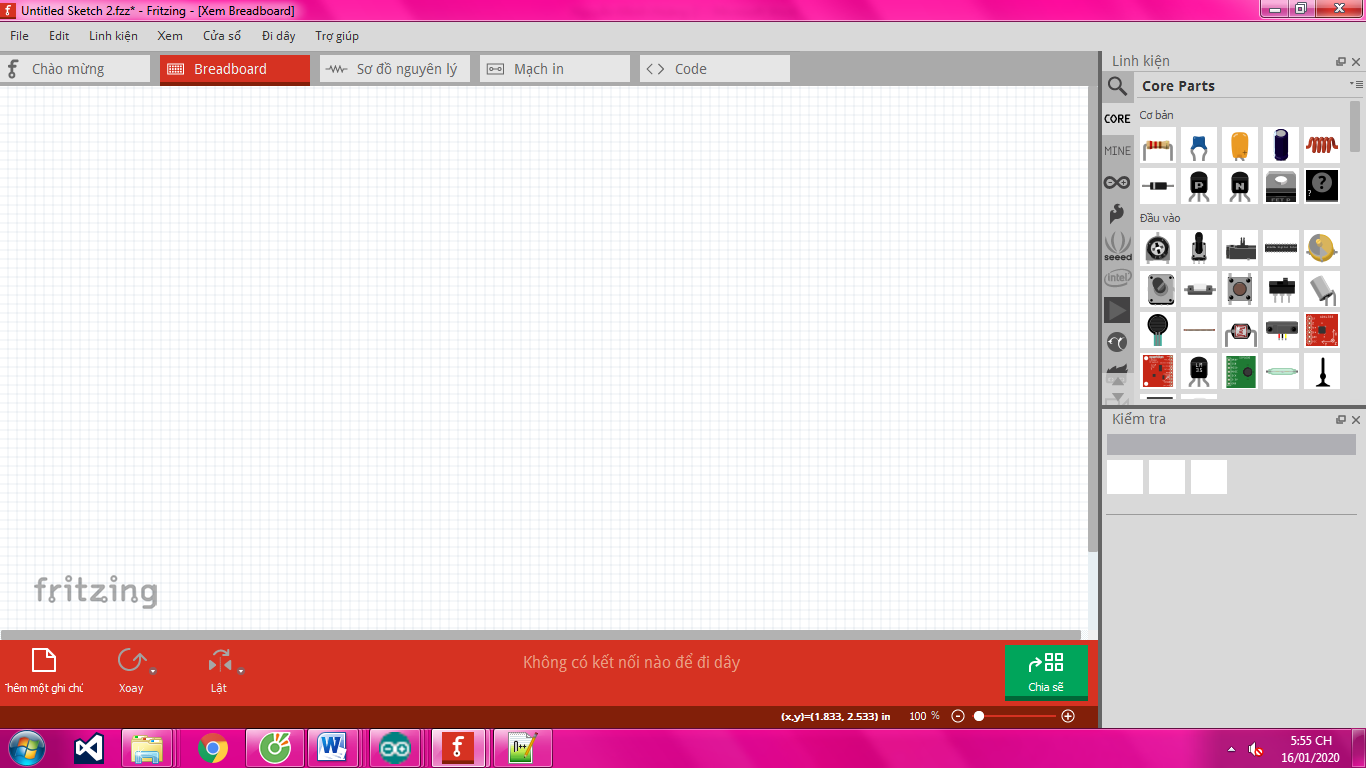
## 1.5. Phần mềm Arduino IDE

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. giao diện phần mềm Arduino IDE

## 1.7. Phần mềm Fritzing



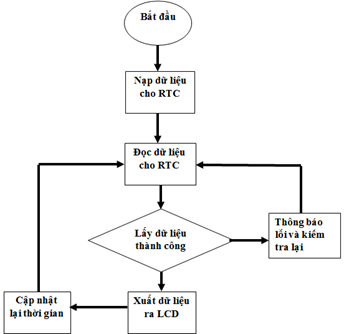
Hình 1.8. Phầm mềm Fritzing

Để thiết kế một sơ đồ mạch điện ứng dụng hoàn chỉnh, phục vụ cho quá trình học tập và nghiên cứu thì bạn cần có nhiều kinh nghiệm và linh kiện điện tử để thử nghiệm. Một giải pháp hiệu quả cho tình huống này là sử dụng phần mềm Fritzing, bạn sẽ nhanh chóng tạo ra những sơ đồ mạch nguyên lý, sơ đồ lắp đặt và từ đó lựa chọn loại PCB sao cho phù hợp với nhu cầu thiết kế.

Quan trọng nhất thanh thư viện là nơi chứa các linh kiện và các vùng làm việc dùng để vẽ mạch nguyên lý.

# 2.PHƯƠNG PHÁP NGHIÊN CỨU

## 2.1. Lập trình



Hình 2.1. Sơ đồ khối chương trình đồng hồ thời gian thực hiển thị trên LCD

### 2.1.1. Giao tiếp Arduino với RTC DS1307

Bảng 2.1. Sơ đồ nối dây giữa Module RTC DS1307 với Arduino Uno R3

|  |  |
| --- | --- |
| **Module RTC DS1307** | **Arduino UNO R3** |
| GND | GND |
| VCC | 5V |
| SDA | A4/SDA |
| SCL | A5/SCL |

RTC DS1307 hoạt động với vai trò slave trên đường bus nối tiếp. Việc truy cập được thi hành với chỉ thị START và một mã thiết bị nhất định được cung cấp bởi địa chỉ các thanh ghi. Tiếp theo đó các thanh ghi sẽ được truy cập liên tục đến khi chỉ thị STOP được thực thi.

**Mô tả hoạt động của các chân:**

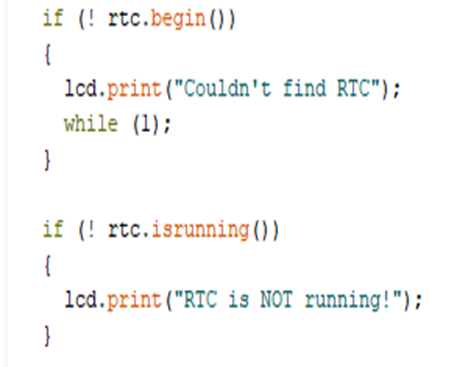
VCC, GND: nguồn một chiều được cung cấp tới các chân này. VCC là đầu vào 5V. Khi 5V được cung cấp thì thiết bị có thể truy cập hoàn chỉnh và dữ liệu có thể đọc và viết.

SCL(serial clock input): SCL được sử dụng để đồng bộ sự chuyển dữ liệu trên đường dây nối tiếp.

SDA( serial data inout/output) : là chân cho hai đường dây nối tiếp

Để giao tiếp giữa DS1307 với Arduino cần dùng đến thư viện Wire.h

Để cho RTC giao tiếp được với Arduino thì cần phải sử dụng đến thư viện “LiquidCrystal\_I2C.h” và thư viện “RTClib.h” công dụng của thư viện này giúp việc lấy dữ liệu từ bộ nhớ của RTC ra xử lý và nạp vào vi điểu khiển.



Hình 2.2. Trình tự các bước khởi tạo RTC

**Bước 1:** Khởi tạo RTC: rtc.begin();

**Bước 2:** Kiểm tra trạng thái hoạt động của RTC: !rtc.isrunning(), kết quả trả về giá trị TRUE khi RTC chưa hoạt động, ngược lại là FALSE.

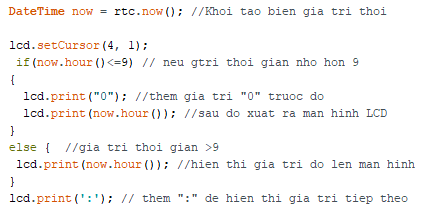
**Bước 3:** Cập nhập thời gian cho RTC

` 

Hình 2.3. Cài đặt thời gian cho DS1307

### 2.1.2. Hiển thị thời gian trên LCD

Để hiển thị thời gian thực trên LCD thì cần dùng đến thư viện LiquidCrystal\_I2C.h

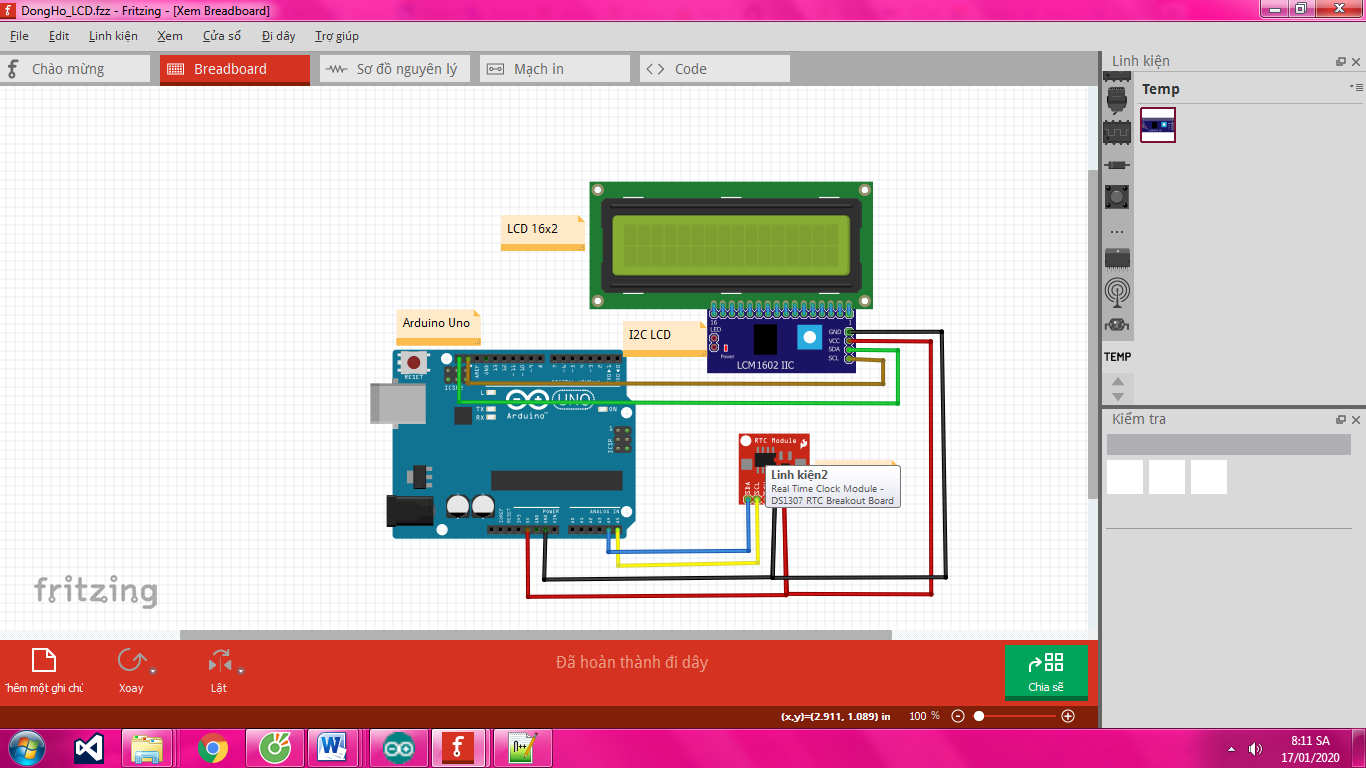


Hình 2.4. Hiển thị thời gian cho LCD

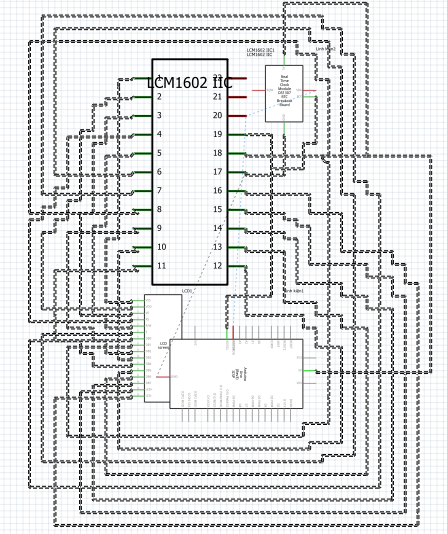
Hình 2.4 là các bước hiển thị số giờ lên màn hình LCD, tương tự như các hàm thời gian phút, giây, thứ, ngày, tháng, năm ( xem mã nguồn ở phần phụ lục ).

## 2.2. Mô phỏng mạch và sơ đồ nguyên lý trên phần mềm Fritzin

Mô phỏng mạch là việc làm có ý nghĩa quan trọng, thông qua việc mô phỏng các giải thuật được kiểm tra một cách chi tiết nhằm tránh những sai sót không đáng có.



Hình 2.5. Mạch mô phỏng



Hình 2.6. Sơ đồ nguyên lí

Trong hình 2.1 LCD được dùng để hiển thị các thành phần như: Thứ, Ngày, Tháng, Năm, Giờ, Phút, Giây.Mạch chuyển đổi I2C (kết nối trực tiếp với 16 chân của LCD) để tiết kiệm được các chân của Arduino UNO R3. I2C có 2 chân SDA VÀ SCL, trong đó SCL là chân CLOCK, được sử dụng để đồng bộ hóa viêc truyền dữ liệu giữa Arduino UNO R3 với LCD 1602. Chân SDA sẽ 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 như sau:

**Bước 1:** Hoàn thiện việc lấy linh kiện và nói dây liên kết

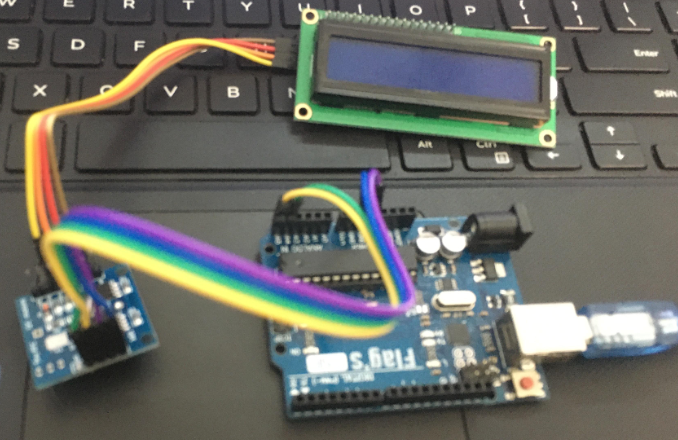
**Bước 2:** Lập trình cho kit Arduino bằng phần mềm Arduino IDE

**Bước 3:** Nạp code cho Arduino bằng cách D\_Click lên biểu tượng board lập trình và lựa chọn tập tin có phần mở rộng .INO để thực thi.

**Bước 4:** Nhấn nút https://scontent.fsgn2-4.fna.fbcdn.net/v/t1.15752-9/82413863_860658057682727_3114504743104282624_n.png?_nc_cat=111&_nc_ohc=BXOfRf9jIPsAX-BGGdm&_nc_ht=scontent.fsgn2-4.fna&oh=c95bcc2273ae8b759d48a425ed6d9856&oe=5EDBAA19 để bắt dầu mô phỏng, nhấn núthttps://scontent.fsgn2-4.fna.fbcdn.net/v/t1.15752-9/82403127_597337601055042_2959385226045816832_n.png?_nc_cat=101&_nc_ohc=LSHQXSgeLRoAX9RXkFW&_nc_ht=scontent.fsgn2-4.fna&oh=e76a73b24fe3699ad534b694cdefebe2&oe=5E8F1E33 để tạm dừng.

# 3. KẾT QUẢ

Sau khi tiến hành nối dây cho các thành phần, kết quả như hình 3.1



Hình 3.1. Màn hình LCD, và các thành phần liên kết của sản phẩm

Cách nối dây:

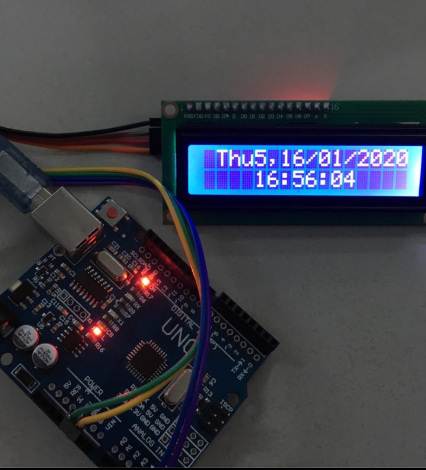
Bảng 3.1. Cách nối dây giữa màn hình LCD với Arduino

|  |  |
| --- | --- |
| **Màn hình LCD 16x2** | **Arduino Uno R3** |
| GND | GND |
| VCC | 5V |
| SDA | A4/SDA |
| SCL | A5/SCL |

Bảng 3.2. Cách nối dây giữa Module RTC DS 1307 với Arduino Uno R3

|  |  |
| --- | --- |
| **Module RTC DS1307** | **Arduino Uno R3** |
| GND | GND |
| VCC | 5V |
| SDA | A4/SDA |
| SCL | A5/SCL |

Chương trình được nạp vào board mạch Arduino, thời gian được hiển thị lên LED 7 đoạn ở theo từ trái sáng phải từ trên xuống dưới tương ứng với ngày, tháng năm, giờ, phút, giây. Hình 3.3 đồng hồ thời gian thực tương ứng với ngày 16 tháng 01 năm 2020, 15 giờ 45 phút 37 giây.



Hình 3.2. Sản phẩm hoàn thiện với thời gian thực được hiển thị trên LCD

# 4. 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.

Sản phẩm hoàn thiện với các chức năng đã được trình bày ở phần phương pháp. Toàn bộ quy trình thiết kế, thi công sản phẩm được hoàn thiện trong thời gian 5 tuần của đợt thực tập cơ sở. Sản phẩm về cơ bản đáp ứng được các yêu cầu đặt ra trong yêu cầu của Giáo viên hướng dẫn.

# PHỤ LỤC

**Code hiển thị thời gian thực trên LCD**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

LiquidCrystal\_I2C lcd(0x27,16,2);

char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};

void setup () {

Serial.begin(9600); // Khởi động một cổng Serial với một baudrate cho trước là 9600, hay còn gọi là tốc độ truyền thông là 9600

lcd.init();

lcd.backlight();

if (! rtc.begin()) {

lcd.print("Couldn't find RTC");

while (1);

}

if (! rtc.isrunning()) {

lcd.print("RTC is NOT running!");

}

rtc.adjust(DateTime(F(\_DATE\_), F(\_TIME\_)));// tu dong cap nhat thoi gian tren may tinh

}

void loop () {

DateTime now = rtc.now(); //Khoi tao bien gia tri thoi

lcd.setCursor(4, 1);

if(now.hour()<=9) // neu gtri thoi gian nho hon 9

{

lcd.print("0"); //them gia tri "0" truoc do

lcd.print(now.hour()); //sau do xuat ra man hinh LCD

}

else { //gia tri thoi gian >9

lcd.print(now.hour()); //hien thi gia tri do len man hinh

}

lcd.print(':'); // them ":" de hien thi gia tri tiep theo

if(now.minute()<=9){

lcd.print("0");

lcd.print(now.minute());

}

else {

lcd.print(now.minute());

}

lcd.print(':');

if(now.second()<=9) {

lcd.print("0");

lcd.print(now.second());

}

else {

lcd.print(now.second());

}

lcd.print(" ");

lcd.setCursor(1, 0);

lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);

lcd.print(",");

if(now.day()<=9){

lcd.print("0");

lcd.print(now.day());

}

else {

lcd.print(now.day());

}

lcd.print('/');

if(now.month()<=9){

lcd.print("0");

lcd.print(now.month());

}

else {

lcd.print(now.month());

}

lcd.print('/');

if(now.year()<=9)

{

lcd.print("0");

lcd.print(now.year());

}

else {

lcd.print(now.year());

}

}