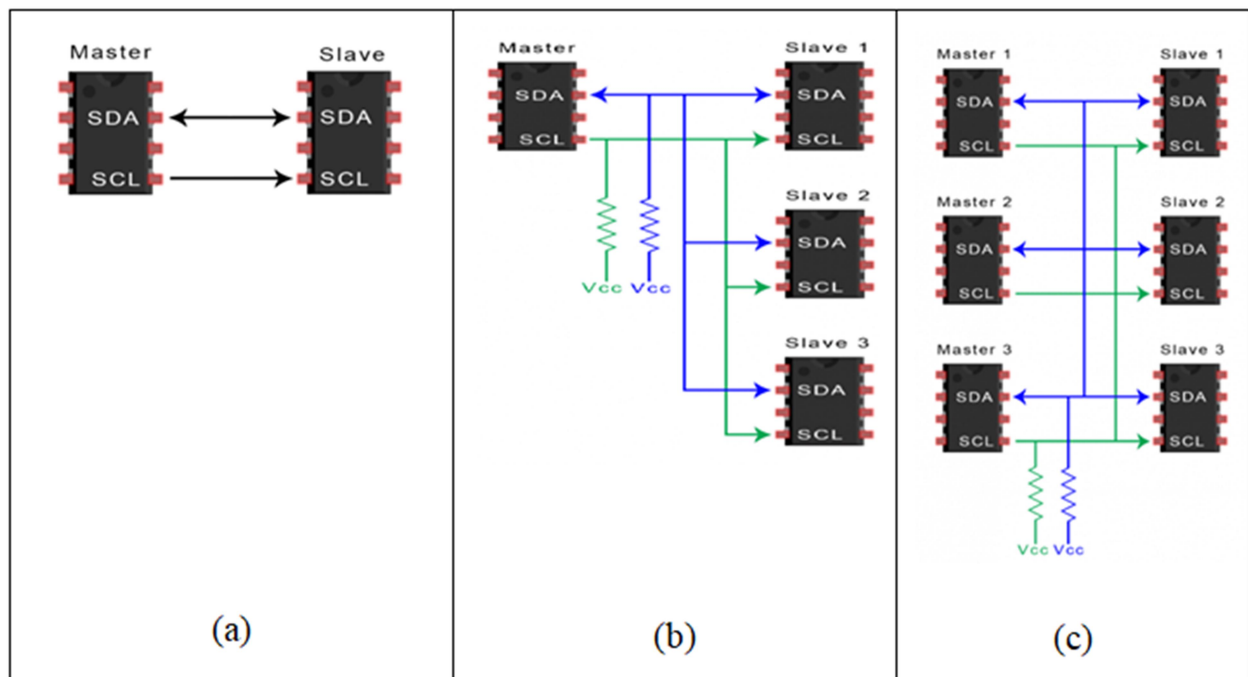


## GIAO THỨC TRUYỀN THÔNG I2C

### I. TRUYỀN THÔNG I2C

#### 1. Giới thiệu truyền thông I2C

I2C là viết tắt của cụm từ Inter-Integrated Circuit, 1 chuẩn truyền thông để giao tiếp giữa các thiết bị được Philips Semiconductor sáng lập và xây dựng thành chuẩn vào năm 1990.



Hình 1. Mô hình truyền thông I<sup>2</sup>C.

#### 2. Giao thức truyền thông I<sup>2</sup>C

##### 2.1. Chế độ tiêu chuẩn, chế độ nhanh và chế độ nhanh tăng cường (Standard-mode, Fast-mode, Fast-mode plus)

Với 2 dây dữ liệu tuần tự (DATA) và xung nhịp (SCL) truyền thông tin giữa các thiết bị kết nối với bus. Mỗi thiết bị được nhận dạng bởi một địa chỉ duy nhất (ngoại trừ thiết bị là VĐK, LCD, bộ nhớ hay bàn phím) và có thể hoạt động cả thu và truyền dữ liệu tùy thuộc vào chức năng của thiết bị (LCD có thể chỉ nhận dữ liệu trong khi bộ nhớ có thể truyền và nhận dữ liệu). Một VĐK khởi tạo việc truyền nhận



## 2.2. Khái niệm chung sử dụng trong I2C

| Thuật ngữ tiếng Anh | Ghi chú  |
|---------------------|--|
| Transmitter         | Thiết bị truyền dữ liệu lên đường truyền   |
| Receiver            | Thiết bị nhận dữ liệu từ đường truyền  |
| Controller          | Thiết bị khởi tạo và kết thúc truyền nhận, phát xung nhịp  |
| Target              | Thiết bị được xác định địa chỉ bởi Controller  |
| Multi-controller    | Hơn một thiết bị Controller điều khiển đường truyền mà không làm mất dữ liệu   |
| Arbitration         | Thủ tục nhằm đảm bảo nếu có nhiều hơn một controller đồng thời điều khiển đường truyền thì chỉ một controller được phép và dữ liệu truyền nhận của controller đó không bị gián đoạn. |
| Synchronization     | Thủ tục đồng bộ tín hiệu xung nhịp của hai hoặc nhiều thiết bị   |

## 2.3. Tín hiệu SDA và SCL

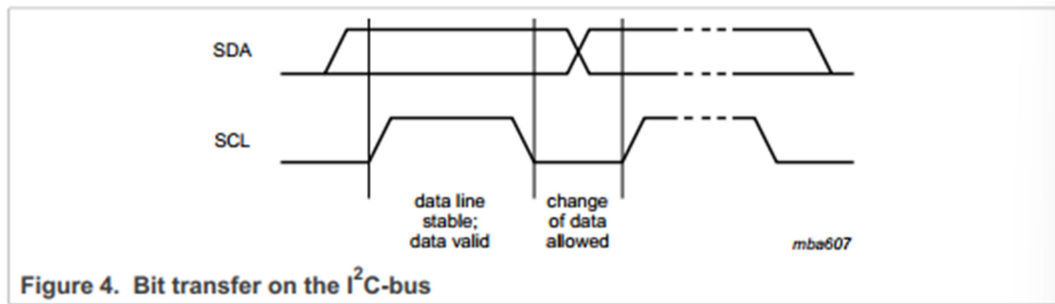
SDA và SCL là các đường truyền hai hướng nối với nguồn cấp dương qua điện trở kéo lên. Khi đường truyền rảnh, cả hai đường này đều được kéo lên mức HIGH. Dữ liệu trên đường truyền I2C có thể đạt tốc độ 100kbit/s ở chế độ Standard mode, 400kbit/s ở chế độ Fast-mode, 1Mbit/s ở chế độ Fast-mode Plus và 3,4Mbit/s ở chế độ High-speed

### 2.3.1. Mức logic tín hiệu SDA và SCL

Tùy thuộc vào công nghệ chế tạo (CMOS, NMOS, bipolar) của thiết bị trên đường truyền I2C, mức logic LOW và HIGH không cố định và phụ thuộc vào mức điện áp nguồn VDD. Mức logic tham chiếu được đặt ở mức 30% và 70% của VDD. Mức điện áp logic LOW = 0,3 VDD và HIGH = 0,7 VDD. Một vài thiết bị có mức logic cứng LOW = 1,5V và HIGH = 3V.

### 2.3.2. Tính hợp lệ của dữ liệu

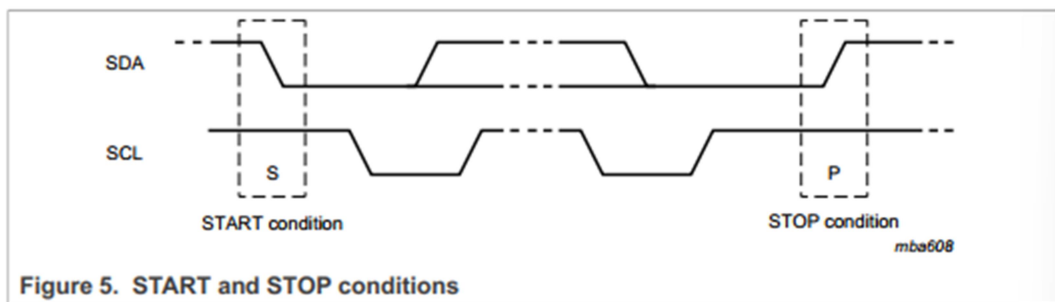
Dữ liệu trên đường SDA phải ổn định trong suốt chu kỳ HIGH của tín hiệu clock. Trạng thái HIGH hay LOW của đường dữ liệu chỉ có thể thay đổi khi mức logic trên đường SCL ở mức LOW. Mỗi một xung clock sẽ truyền một bit dữ liệu trên đường dây.



Hình 2. Dữ liệu hợp lệ truyền trên I<sup>2</sup>C

### 2.3.3. Điều kiện START và STOP

Mọi truyền nhận đều bắt đầu bằng START(S) và kết thúc bằng STOP(P). Tín hiệu START xác định khi SDA chuyển từ mức HIGH xuống LOW trong khi SCL ở mức HIGH. Tín hiệu STOP khi SDA chuyển từ mức LOW sang mức HIGH khi SCL ở mức HIGH.

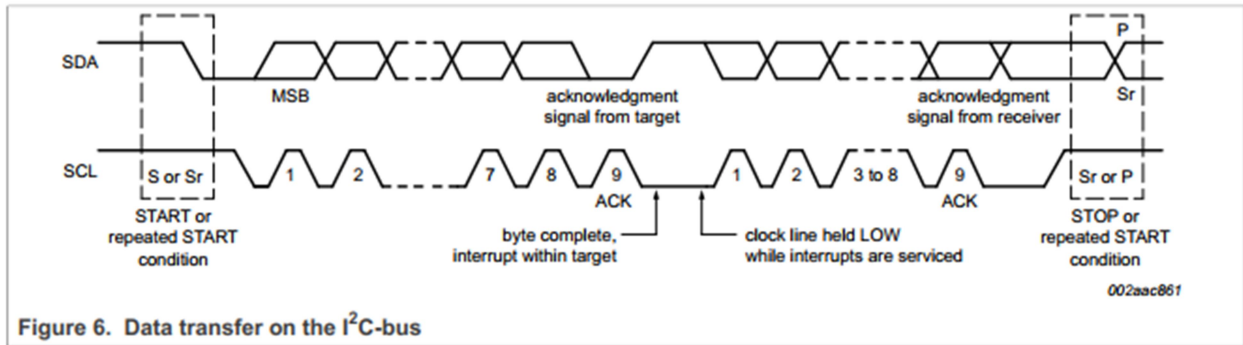


Hình 3. Điều kiện Start, Stop trong truyền thông I<sup>2</sup>C.

START và STOP được điều khiển bởi VĐK. Đường truyền ở trạng thái bận sau khi có lệnh START, rảnh sau khi có tín hiệu STOP. Đường truyền duy trì trạng thái bận khi có một tín hiệu START được gửi ngay sau khi kết thúc một khung truyền (Sr),

### 2.3.4. Định dạng byte truyền thông I<sup>2</sup>C

Mọi byte đưa lên đường SDA đều có độ dài 8 bit. Số lượng byte mỗi lần truyền là không giới hạn. Mỗi byte phải được kèm theo một bit Acknowledge. Với mỗi byte, Most Significant Bit (MSB) được truyền đầu tiên.



Hình 4. Định dạng khung truyền dữ liệu trong I<sup>2</sup>C.

### 2.3.5. Acknowledge (ACK) và Not Acknowledge (NACK)

Bit ACK được gửi sau mỗi byte, bit ACK cho phép thiết bị nhận xác nhận với thiết bị truyền là 1 byte đã được nhận thành công và có thể gửi byte kế tiếp. Chân SCL phát xung nhịp thứ 9 sau mỗi 8 xung cho 1 byte dữ liệu.

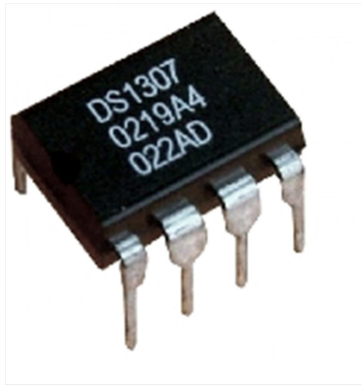
Tín hiệu ACK được định nghĩa như sau: transmitter giải phóng đường SDA khi có xung SCL thứ 9, Receiver kéo đường SDA xuống mức 0 trong suốt thời gian ở mức cao của tín hiệu SCL. Tín hiệu SDA duy trì ở mức cao trong suốt thời gian mức cao của tín hiệu SCL báo hiệu một tín hiệu NACK. Nếu có tín hiệu NACK controller có thể kết thúc truyền thông hoặc khởi tạo điều kiện START để bắt đầu khung truyền mới. Có 5 khả năng dẫn đến việc xuất hiện tín hiệu NACK:

- Không có Receiver trên đường truyền hoặc địa chỉ của Receiver không đúng.
- Receiver đang thực hiện các tác vụ riêng nên không sẵn sàng trả lời Controller.
- Trong quá trình truyền nhận, Receiver nhận được dữ liệu hoặc lệnh không phù hợp.
- Receiver không nhận được dữ liệu trong quá trình truyền, nhận.
- Một Controller-receiver gửi tín hiệu kết thúc lên đường truyền.

## II. ĐỒNG HỒ THỜI GIAN THỰC DS1307

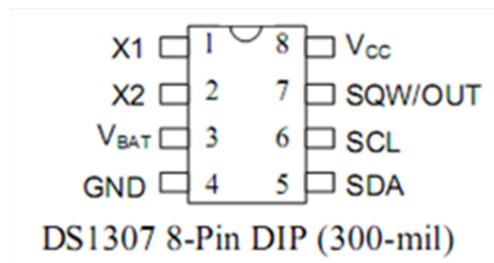
### 1. Giới thiệu DS1307

Đồng hồ thời gian thực là sản phẩm của hãng Dallas Semiconductor. Dữ liệu trên DS1307 được đọc và ghi thông qua truyền thông I<sup>2</sup>C.



Hình 5. Chip DS1307

Cấu tạo bên ngoài gồm 8 chân.



Hình 6. Sơ đồ chân và chức năng của chip DS1307

Các chân DS1307 có chức năng như sau:

- X1 và X2: là 2 ngõ kết nối với 1 thạch anh 32,768 KHz làm nguồn dao động cho chip.
- V<sub>BAT</sub>: cực dương của 1 nguồn pin 3v nuôi chip.
- GND: chân mass chung cho cả Pin 3v và V<sub>CC</sub>.
- V<sub>CC</sub>: nguồn cho giao diện I2C, thường là 5v dùng chung với vi điều khiển.

**Chú ý:** nếu không cấp nguồn V<sub>CC</sub> nhưng V<sub>BAT</sub> vẫn được cấp thì chip DS1307 vẫn hoạt động.

- SQW/OUT: ngõ phụ tạo xung vuông.
- SCL, SDA: là 2 đường xung nhịp và dữ liệu của giao diện I2C

## 2. Thanh ghi trên DS1307

DS1307 có tất cả **7 thanh ghi 8 bit** chứa thời gian là giây, phút, giờ, thứ, ngày, tháng, năm. Địa chỉ các thanh ghi này từ 0x00 đến 0x07

Ngoài ra DS1307 còn có 1 thanh ghi điều khiển ngõ ra phụ và 56 thanh ghi trống có thể dùng như RAM.

Địa chỉ của DS1307 luôn là 0x68. Do đó chúng ta không thể dùng đồng thời 02 DS1307 trên một đường truyền I<sup>2</sup>C (điều này cũng không cần thiết).

| ADDRESS | BIT 7   | BIT 6      | BIT 5   | BIT 4    | BIT 3   | BIT 2 | BIT 1 | BIT 0 | 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<br>+AM/PM<br>00–23 |
|         |         | 24         | PM/AM   |          |         |       |       |       |               |                         |
| 03h     | 0       | 0          | 0       | 0        | 0       | DAY   |       |       | Day           | 01–07                   |
| 04h     | 0       | 0          | 10 Date |          | 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<br>56 x 8 | 00h–FFh                 |

Hình 7. Thanh ghi trong DS1307

### 2.1. Thanh ghi giây (Seconds)

- Thanh ghi đầu tiên trong bộ nhớ DS1307, địa chỉ của nó là 0x00.
- 4 bit thấp thanh ghi chứa mã BCD của 4 bit chữ số hàng đơn vị của giá trị giây.
- 3 bit cao tiếp theo chứa mã BCD của 3 bit chữ số hàng chục giá trị giây.
- Bit 7(bit cao nhất): luôn luôn phải đặt về mức 0 để chip hoạt động.

### 2.2. Thanh ghi phút(Minutes)

- Có địa chỉ 0x01, chứa giá trị phút của đồng hồ.
- Giống thanh ghi giây. Bit 7 luôn được đặt mức 0.

### 2.3. Thanh ghi giờ (Hours)

- Thanh ghi có địa chỉ 0x20.
- 4 bit thấp (bit 0 đến bit 3) dùng để chỉ chữ số hàng đơn vị của giờ.
- Bit 6 dùng để đặt chế độ giờ. Bit 6 = 0 chế độ 24 giờ. Bit 6 = 1 chế độ 12 giờ.
- Ở chế độ 24 giờ:
  - o Bit 4 và bit 5 để chỉ chữ số hàng chục của giờ.
- Ở chế độ 12 giờ:
  - o Bit 4 dùng để chỉ chữ số hàng chục của giờ.
  - o Bit 5 dùng để chỉ buổi trong ngày. Bit 5 = 0: Am. Bit 5 = 1: Pm.

2.4. *Thanh ghi thứ:*

- Nằm ở địa chỉ 0x30.
- 3 bit thấp để chỉ mã PCD của thứ từ 2 đến 7.

2.5. *Thanh ghi ngày, tháng, năm*

Tương tự như thanh ghi thứ.



### III. SỬ DỤNG TRUYỀN THÔNG I<sup>2</sup>C TRÊN RASPBERRY PI

#### 1. Cấu hình truyền thông I<sup>2</sup>C trên Raspberry

Raspbian mặc định ngắt truyền thông I2C, chúng ta cần phải thiết lập mở I2C trước.

- Bước 1:  
Gọi Raspi-config bằng dòng lệnh sau trong Terminal  
**sudo raspi-config**
- Bước 2:  
Lựa chọn Advanced options → I2C → Yes. → Finish
- Bước 3:  
Gõ dòng **lệnh sudo nano/etc/modules** để sửa file modules
- Bước 4:  
Thêm 2 dòng này ở cuối file:  
**I2c-bcm2708**  
**I2c-dev**
- Bước 5:  
Cài đặt thư viện I<sup>2</sup>C cho python bằng dòng lệnh  
Sudo apt-get install python3-smbus
- Bước 6:  
Khởi động lại Raspberry Pi
- Kiểm tra  
Sau khi khởi động xong có thể kiểm tra các cổng I<sup>2</sup>C có hoạt động không bằng lệnh:  
Lsmod | grep i2c  
Câu lệnh trên liệt kê các cổng I<sup>2</sup>C có trên Raspberry Pi. Nếu có dòng chữ 'i2c\_bcm....' Thì việc mở cổng I<sup>2</sup>C đã thành công.

## 2. Chương trình đọc thời gian từ DS1307 viết bằng Python cho Raspberry PI3

```
import RPi.GPIO as GPIO
import smbus
import time
DS1307 = 0x68 # DIA CHI CUA DS1307
bus = smbus.SMBus(1)
#các hàm sử dụng trong truyền thông I2C
#write_byte(addr,val) gửi 1 byte dữ liệu
#read_byte(addr) đọc 1 byte dữ liệu
#write_i2c_block_data(addr,cmd,vals) gửi mảng dữ liệu
#read_i2c_block_data(addr,cmd)

def BCD2DEC(bcd):
    # dec = (((bcd>>4)&0x07)*10 + bcd&0x0F)
    dec = ((bcd//16)*10 + bcd%16)
    return dec

def DEC2BCD(dec):
    bcd = (dec//10)*16 + dec%10

def readDS1307():
    bus.write_byte(DS1307,0x00)
    data = bus.read_i2c_block_data(DS1307,0x00,0x07)
    second = BCD2DEC(data[0])
    minute = BCD2DEC(data[1])
    hour = BCD2DEC(data[2])
    wday = BCD2DEC(data[3])
    day = BCD2DEC(data[4])
    month = BCD2DEC(data[5])
    year = BCD2DEC(data[6])
    year += 2000
    print("Gio: %d Phut: %d Giay %d"%(hour,minute,second))
while(1):
    readDS1307()
    time.sleep(1)
```

#### IV. BÀI TẬP

1. Viết chương trình trên Python đọc thời gian từ DS1307 sử dụng thư viện **smbus** hiển thị lên LCD16x2 với:
  - Hàng 1 hiển thị GIỜ: PHÚT: GIÂY
  - Hàng 2 hiển thị NGÀY – THÁNG – NĂM.
2. Không sử dụng thư viện smbus, tự viết truyền thông I<sup>2</sup>C giữa DS1307 và Raspberry Pi.