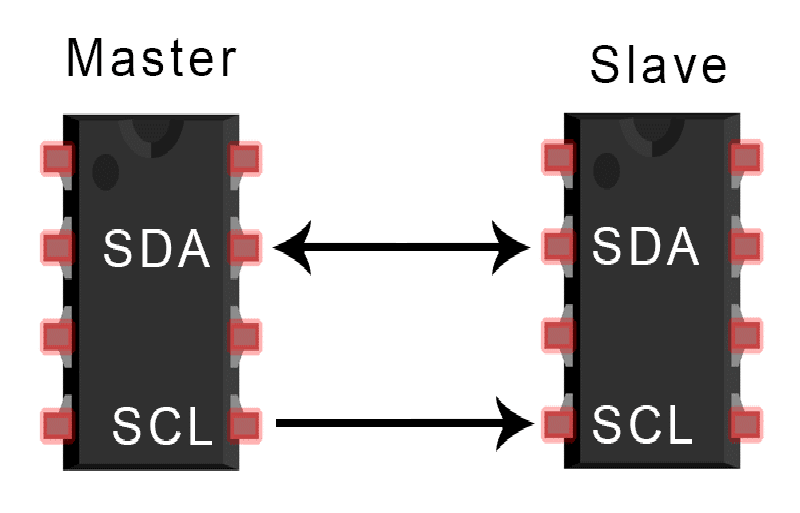
Giao tiếp ngoại vi I2C.

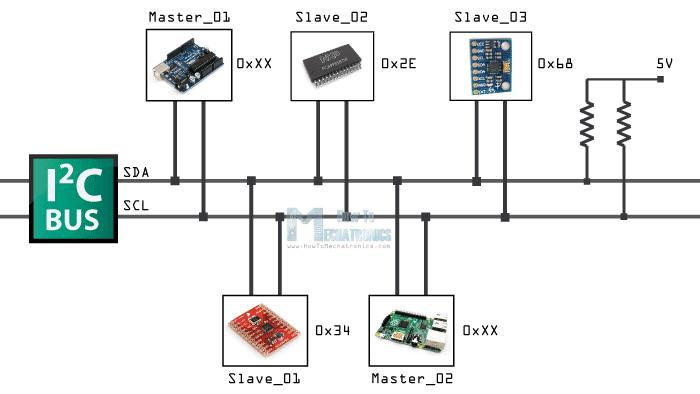
1. I2C (Inter- Integrated Circuit)

* Là 1 chuẩn đồng bộ khi truyền dữ liệu giữa bộ xử lý trung tâm với các IC trên 1 board mạch.
* Dạng Master và Slave: Nhiều Master và nhiều Slave.
* I2C sử dụng 2 đường truyền tín hiệu là SDL và SDA
  + SDL (Serial Clock Line): Xung nhịp đồng bộ do Master phát.
  + SDA (Serial Data Line): Đường truyền dữ liệu.

Hình ảnh:



* Các chân SDA và SDL của các Master và Slave được nối với nhau.



1. Ưu điểm của giao thức I2C.

Ưu điểm:

* Chỉ cần 2 đường bus chung để truyền nhận dữ liệu.
* Dữ liệu được trích mẫu thông qua xung nên không cần có sự quy ước từ phía Slave. → **Master có thể thay đổi tốc độ đọc, ghi bất kì lúc nào.**
* Khả năng mở rộng mạng tốt.
* Cơ chế truyền đơn giản.

Hạn chế:

* Tốc độ truyền dữ liệu thấp hơn những giao thức như SPI (10 – 20 MB/s),…

1. Nguyên lý.

* Sử dụng hệ thống 7 bit để xác định 1 thiết bị.**(Address)**
* Chế độ truyền: Bán song công.**(Half duplex)**
* Các chế độ hoạt động:
  + Standard mode: Tốc độ 100 KB/s
  + Fast mode: Tốc độ 400 KB/s
  + Fast mode+ : Tốc độ 1 MB/s
  + High speed mode: Tốc độ 3.4 MB/s

A diagram of a machine

Description automatically generated

* Khung truyền I2C.
* Khi bắt đầu, chân SCL đang ở mức cao, chân SDA chuyển từ mức cao xuống mức thấp. Thể hiện việc bắt đầu truyền.(Bit Start)
* Master gửi địa chỉ của Slave mà nó muốn truyền nhận dữ liệu qua chân SDA. Sau đó các Slave sẽ so sánh địa chỉ.
  + Nếu trùng: Slave sẽ gửi 1 bit (ACK = 0) trở lại master.
  + Nếu không: Chân SDA vẫn sẽ ở mức cao. Không xảy ra truyền nhận.
* Sau khi xác nhận có Slave, Master sẽ gửi 1 bit cho biết nó muốn truyền hay nhận dữ liệu. Cụ thể:
  + Bit 0: Master muốn gửi dữ liệu đến Slave
  + Bit 1: Master muốn nhận dữ liệu từ Slave.
* Khi truyền dữ liệu:

A screenshot of a computer code

Description automatically generated

* Sau khi gửi bit ghi dữ liệu, Master sẽ gửi địa chỉ mà nó muốn ghi lên Slave.
* Ví dụ: Trên RTC DS1307:

A white and black document with numbers and letters

Description automatically generated with medium confidence

* Address 00h sẽ ứng với số giây, 01h sẽ ứng với số phút,…
* Sau khi nhận được và xác nhận địa chỉ, Slave sẽ tiếp tục gửi bit ACK (0) hoặc NACK (1) để xác nhận có hay không có địa chỉ.
* Nếu nhận được bit ACK, Master sẽ tiếp tục ghi data trên thanh ghi. Slave sẽ truyền lại bit ACK/NACK để báo cho Master đã nhận được data hay chưa.
* Nếu tiếp tục truyền data, những data này sẽ được ghi trên thanh ghi tiếp theo. VD: sau khi ghi trên 00h, nếu tiếp tục truyền thì data sẽ đẩy sang 01h.
* Khi đọc dữ liệu

A screenshot of a computer code

Description automatically generated

* Tương tự khi ghi dữ liệu, Master cũng sẽ gửi bit ghi (0) qua SDA sau đó gửi địa chỉ thanh ghi.
* Sau khi nhận phản hồi, Master sẽ gửi bit Sr báo hiệu đọc liên tục.
* Tiếp theo, Master tiếp tục gửi địa chỉ Slave và bit đọc. Sau đó sẽ tương tự với bước ghi, Master sẽ đọc dữ liệu từ các thanh ghi liên tiếp.

1. Config và code. (Module DS1307). (STM32F103C8T6)
2. Config trên CubeMx.

* Tại System Core :
* Tab Sys
  + Debug : Serial Wire
  + Timebase Source : SysTick

A screenshot of a computer

Description automatically generated

* Tab RCC
  + High Speed Clock (HSE) :

Chọn Crystal/Ceramic Resonator nếu sử dụng bộ dao động ngoại.(Thạch anh,…)

Chọn BYPASS Clock Source nếu muốn dùng bộ dao động nội.(Max speed 64 MHz)

* Tại Connectivity.

Chọn 1 trong 2 tab I2C1 hoặc I2C2 (Có thể chọn cả 2). Chọn mode I2C.

Đối với STM32F103C8T6. Nếu chọn I2C1, sẽ sử dụng 2 chân:

* + SCL: PB6
  + SDA: PB7

Nếu chọn I2C2:

* + SCL: PB10
  + SDA: PB11

A screenshot of a computer

Description automatically generated

Tại tab Configuration, đối với STM32F103C8T6, chỉ có 2 chế độ I2C Speed Mode là Standart Mode và Fast Mode.

Sau khi config xong, ta generate code.

1. Code

Nếu muốn config lại, có thể mở lại file CubeMx hoặc config trực tiếp trong file main.c (Không khuyến khích)

A screenshot of a computer code

Description automatically generated

* Khi generate code, CubeMx sẽ tạo ra cho ta 1 biến con trỏ hi2c1 :

A computer code with black text

Description automatically generated

Chuột phải, sau đó chọn Go to define vào struct để có thể đến file thư viện HAL cho I2C.

Nếu khéo xuống dưới sẽ thấy các hàm.

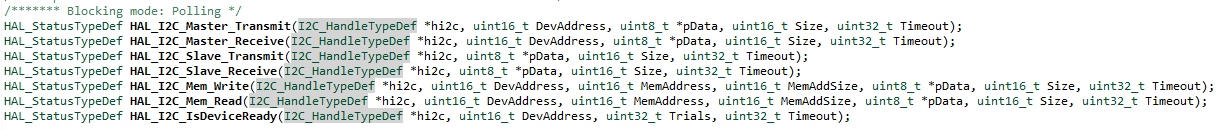


Figure 1. Các hàm truyền nhận thông thường

A close-up of a text

Description automatically generated

Figure 2. Các hàm truyền nhận ngắt.(Phải cấu hình ngắt trong NVIC)

A screen shot of a computer

Description automatically generated

Figure 3. Các hàm Callback.

* Code truyền nhận dữ liệu cho RTC DS1307.

Đầu tiên, ta phải biết các thông số của DS1307.

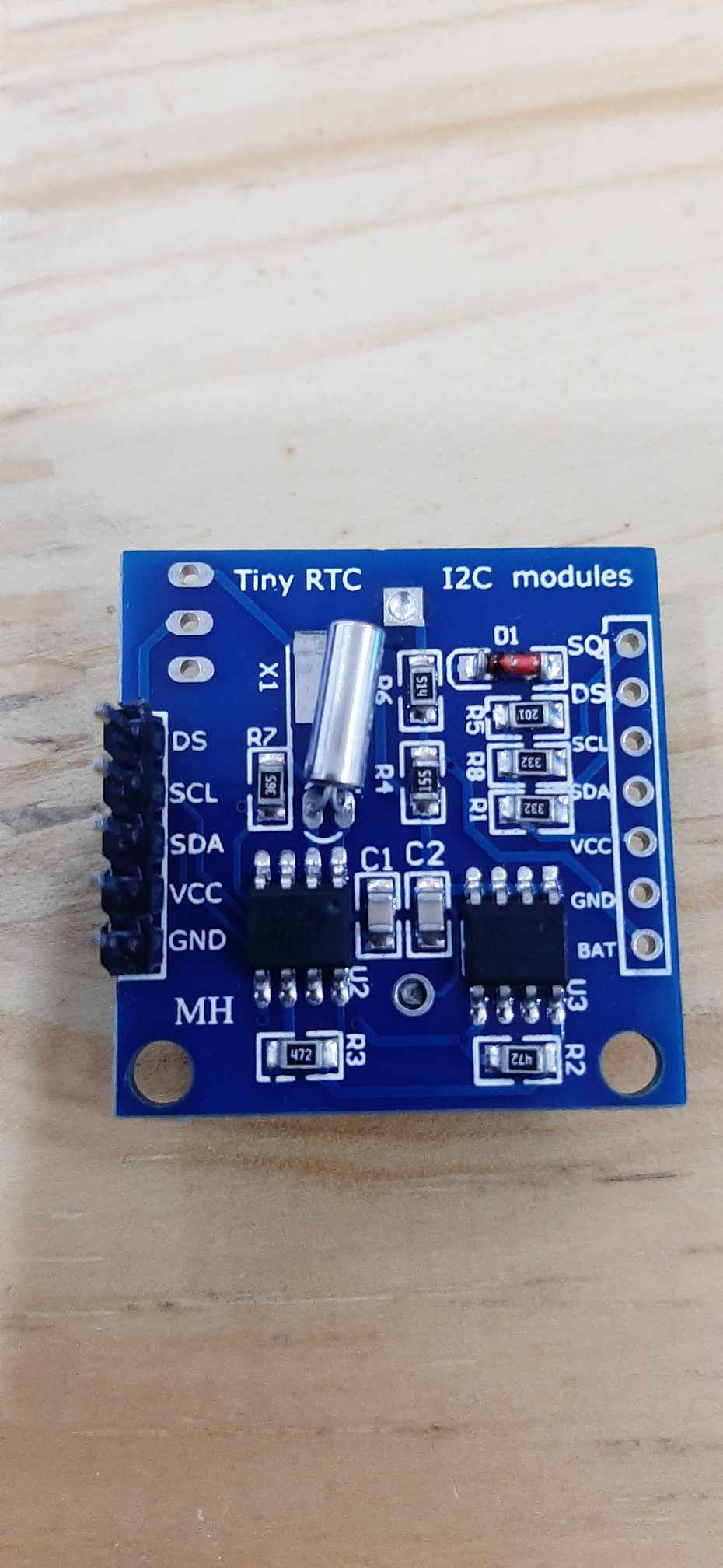


Figure 4. Mặt sau của DS1307.(Sử dụng 4 chân SCL, SDA, VCC và GND trong bài này)

  
Cắm các chân SCL vào PB6, SDA vào PB7, VCC vào nguồn 3V3, GND vào GND.

Figure 5. Mặt trước của DS1307.

(Lưu ý, pin Lithium sử dụng dể nuôi clock bên trong Module, còn chân VCC và GND sử dụng để cấp nguồn khi truyền nhận dữ liệu.)

Các thông số:

* Địa chỉ của DS1307: 1101000 (0x68<<1) (0x68 dịch đi 1 bit.)
* Địa chỉ các thanh ghi:

A white and black document with numbers and text

Description automatically generated with medium confidence

* Define constant:



* Bước đầu, ta khởi tạo struct để lưu trữ các giá trị:

A screen shot of a computer code

Description automatically generated

* DS1307 lưu trữ thông tin dưới kiểu BCD.(Có thể tìm hiểu trên GG) để có thể dễ đếm. Vậy ta cần tạo hàm để chuyển đổi giữa 2 kiểu dữ liệu:

A computer code with black and white text

Description automatically generated

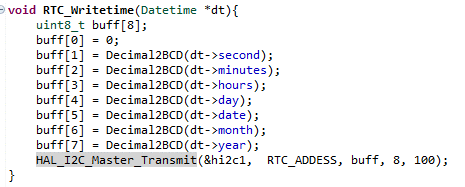
* Sau đó, ta tạo hàm để viết giá trị vào trong DS1307:

Hàm sử dụng:

**HAL\_I2C\_Master\_Transmit**(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddress, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout)

Giải thích:

* \*hi2c: Biến con trỏ được khởi tạo ban đầu.(hi2c1)
* DevAddress: Địa chỉ Module DS1307.(0x68<<1)
* \*pData: Dữ liệu muốn truyền.
* Size: Độ dài dữ liệu
* Timeout: Thời gian Timeout (ms).(Tránh cho việc truyền bị lỗi)



Khi khởi tạo lần đầu, DS1307 sẽ không có giá trị. Ta cần nạp các giá trị ban đầu cho nó, sau đó có thể bỏ hàm này**.(buff[0] là địa chỉ thanh ghi đầu tiên ta muốn ghi)**

A screenshot of a computer program

Description automatically generated

(Có thể comment lại)

Sau khi chạy code, thông tin sẽ được nạp vào Module DS1307.

* Hàm để đọc giá trị từ RTC:

**HAL\_I2C\_Master\_Receive**(I2C\_HandleTypeDef \*hi2c, uint16\_t DevAddress, uint8\_t \*pData, uint16\_t Size, uint32\_t Timeout)

(Tương tự hàm Transmit)

A screen shot of a computer program

Description automatically generated

* Như đã trình bày ở trên, ta cần phải truyền cho nó 1 dữ liệu ban đầu (Transmit) để bắt đầu quá tình Receive.
* Sau đó sử dụng hàm Recive để lấy giá trị rồi ghi lại vào con trỏ \*dt.
* Sau khi code được hàm Readtime. Ta phải đưa nó vào vòng While (1) để tiến hành đọc thời gian liên tục.

A screenshot of a computer code

Description automatically generated

* Mỗi giây ta sẽ đọc 1 lần. Có thể sử dụng HAL\_GetTick() để không gây gián đoạn chương trình.
* Có thể tiến hành Debug để kiểm tra lại:
* Build:

A screenshot of a computer screen

Description automatically generated

* Debug.

Trong Live expressive:

A screenshot of a computer

Description automatically generated

Thông tin vẫn tiếp tục chạy:

11 giờ, 12 phút, 44 giây, Thứ 6 ngày 10/11/2023.