I2C stm32f103

1.1 I2c master mode

Ở chế độ, , giao diện I2c bắt đầu dữ liệu truyền và khởi tạo tín hiệu xung. 1 dãy dữ liệu truyền luôn luôn bắt đầu bằng START CONDITION và kết thúc bằng STOP CONDITION

Chế độ master được lựa chọn ngay khi tín hiệu START\_CONDITION được khởi tạo trên bus cùng với start bit

Dưới đây là một dãy các yêu cầu trong chế độ master

Lập trình ngoại vi xung đầu vào tring thanh ghi I2C\_CR2 để mà khởi tạo chính xác thời gian

Cấu hình những thanh ghi điều kiển xung

Lậo trình thanh ghi I2C\_CR1 để cho phép ngoại vi

Đặt START bit trong thanh ghi I2C\_CR1 để khởi tạo điều kiển bắt đầu

Tần số xung ngoại vi đầu vào cần phải tối thiểu : 2MHz ở chế độ Sm và 4MHz ở chế độ Fm

1.3 Điều kiển bắt đầu, start condition

Việc mà đặt tín hiểu bắt đầu là để giao diện tạo 1 Start condition và kích hoạt chế độ master (MSL bit được bật) khi BUSY bit bị xoá

Ghi chú: ở chế độ master, việc đặt START CONDITION cũng khởi tạo ReSTART CONDITION ở cuối của chuyển đổi dữ liệu hiện thời

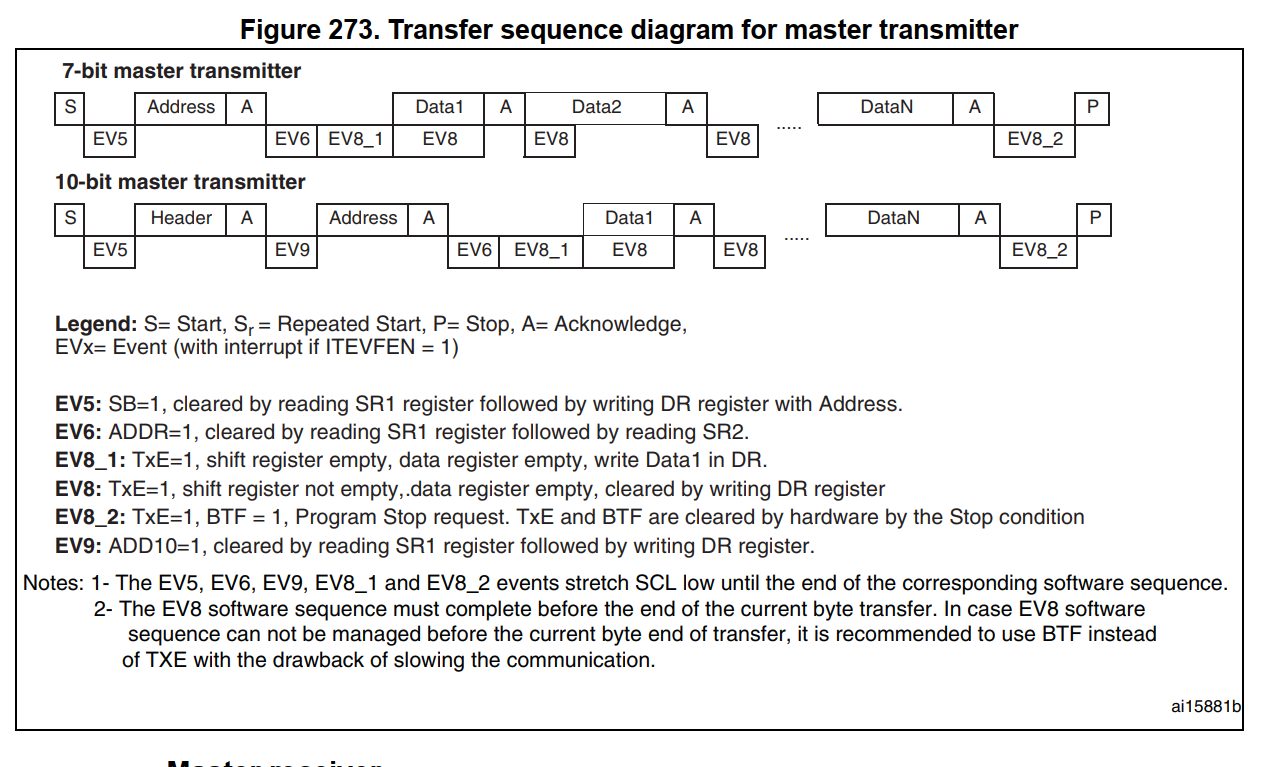
1… Master gữi dữ liệu

Sau khi truyền địa chỉ và sau khi xoá ADDR, thiết bị chủ gữi những byte dự liệu từ DR đến dây SDA thông qua thanh ghi dịch

Master đợi đến khi dữ liệu đầu tiên được ghi và I2C\_DR

Khi nào xung acknowledge được nhận, bit TxE được bật bởi phần cứng và 1 ngắt được khởi tạo nếu ITEVFEN và ITBUFEN được bật

Nếu TxE được bật và 1 dữ liệu đã không được ghi và DR trước khi kết thúc 1 truyền dữ liệu cuối cùng , BTF được bât và giao diện chờ cho đến khi BTF được xoá bở việc đọc SR1 theo sau đó là ghi DR,



Các bước để ghi 1 byte dữ liệu, 7 bit address

Gữi bit Start

EV5 : SB = 1, xoá bởi đọc SR1 được theo sau bởi việc đọc DR cùng với địa chỉ

EV6 : ADDR = 1, xoá bởi việc đọc SR1 sau khi đọc SR2

EV8\_1 : TxE = 1, thanh ghi dịch không trống, thanh ghi dữ liệu trống, ghi data 1 DR

EV8 : TxE = 1, thanh ghi dịch không trống, thanh ghi dữ liệu trống, xoá bởi việc ghi DR

EV8\_2 : TxE = 1, BTF = 1, yêu cầu lập trình stop condition, TXE và BTF bị xoá bởi phần cứng bởi STOP CONDITION

|  |
| --- |
| // create start condition  I2C1->CR1 |= (1<<8);  // EV5 : check Start condition generated  while(! (I2C1->SR1 & (1<<0)));  status = I2C1->SR1;  I2C1->DR = address;    // EV6: Received address matched  while(! (I2C1->SR1 & (1<<1)));  status = I2C1->SR1;  status = I2C1->SR2;    // EV8\_1: write data DR  while(! (I2C1->SR1 & (1<<7)));  I2C1->DR = byte;    while(! (I2C1->SR1 & (1<<7)));  while(! (I2C1->SR1 & (1<<2)));    // EV8\_2: create stop condition  I2C1->CR1 |= (1<<9);  status = I2C1->SR1;  status = I2C1->SR2; |

MASTER RECEIVER

Sau khi chuyển đổi địa chỉ, và sau khi xoá ADDR, giao diện I2C đi vào Chế độ nhận master. Trong chế độ nàym giao diện nhận những byte từ đường SDA trong thanh ghi DR thông qua thanh ghi dịch. Sau mỗi một byte, giao diện khởi tạo theo chuỗi sau :

1. Một cái xung ACK nếu bit ACK được bật
2. Bit RxNE được set và ngắt được khảoi tạo nến ITVEFEN và ITBUFEN được đặt

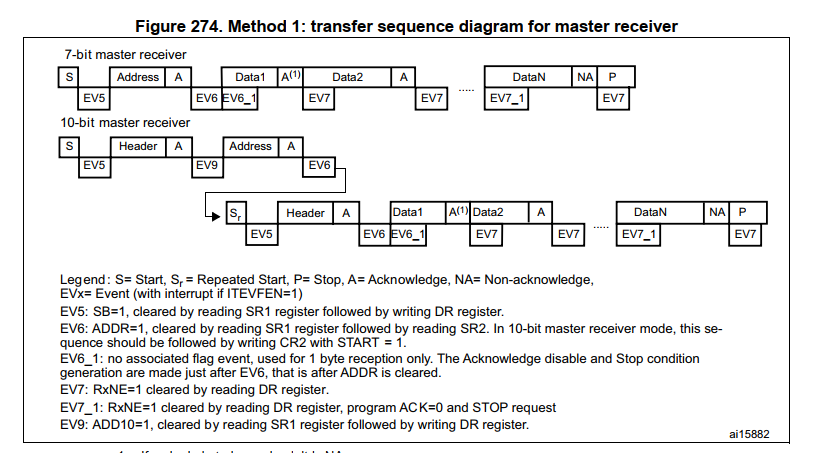
Nếu bit RxNE được đặt và dữ liệu DR không được đọc trước sự kết thúc của ngoại lệ dữ liệu cuối cùng, bit BUF được bật bởi phần cứng và giao diện vẫn gữi BTF cho đến khi xoá bằng cách đọc SR1 theo sau bưởi việ đọc DR, một khảo delay SCL thấp

Đóng kết nối :

Phương pháp 1: phương pháp là 1 trường hợp khi I2C được sử dụng cùng với ngắt cái mà có mức ưa tiên cao nhất trong ứng dụng

Thiết bị chủ gữi một NACK cho byte cuối cùng nhận được từ slave. Sau khi việc nhận cái NACK, thiế bị tớ giải phóng sự điều kiển của đường SCL và SDA. Sau đó thiết bị master có thể gữi stop hoăc restart condition

1. Để khởi tạo xung NACK sau việc byte dữ liệu đã nhận cuối cùng, bit ADC cần được xoá sau khi đọc the cuối cùng thức 2 (sau khi sự kiên RxNE cuối thứ 2)
2. Để khởi tạo Stop/Restart condition, phần mền cần phải set STOP và Start bit sau khi đọc byte cuối cùng thứ 2
3. Trong trường hợp 1 byte được nhận, ACK không cho phép và khởi tạo điện kiện STOP được làm chỉ sau EV6 (in EV6\_1, chỉ sau xoá ADDR )



EV5: SB=1, được xoá khi đọc tg SR1, sau đó là ghi tg DR

EV6 : ADDR = 1, xoá bởi việc đọc thanh ghi SR1, sau đó đọc SR2.