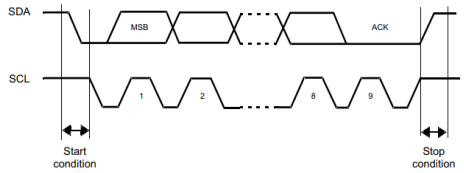
BASIC UNDERSTANDING OF 12C PERIPHERAL OF STM32

Configuration as master (7bit addressing)

- Standard mode (up to 100 kHz),Fast mode (up to 400 kHz)



- I2C protocol start and stop condition
- SDA transitions when SCL is kept at same level only for start and stop condition

MASTER MODE

Master mode is selected as soon as the Start condition is generated on the bus with a START bit.

The following is the required sequence in master mode.

- Program the peripheral input clock in I2C_CR2 register in order to generate correct timings
- Configure the clock control registers
- Configure the rise time register
- Program the I2C_CR1 register to enable the peripheral
- Set the START bit in the I2C_CR1 register to generate a Start condition

The peripheral input clock frequency must be at least:

- 2 MHz in Sm mode
- 4 MHz in Fm mode

Registers for I2C peripheral

0x4000 5800 - 0x4000 5BFF	I2C2
0x4000 5400 - 0x4000 57FF	I2C1

Table 189. I²C Interrupt requests

Interrupt event	Event flag	Enable control bit
Start bit sent (Master)	SB	ITEVFEN
Address sent (Master) or Address matched (Slave)	ADDR	
10-bit header sent (Master)	ADD10	
Stop received (Slave)	STOPF	
Data byte transfer finished	BTF	
Receive buffer not empty	RxNE	ITEVFEN and ITBUFEN
Transmit buffer empty	TxE	

Table 189. I²C Interrupt requests (continued)

Interrupt event	Event flag	Enable control bit
Bus error	BERR	ITERREN
Arbitration loss (Master)	ARLO	
Acknowledge failure	AF	
Overrun/Underrun	OVR	
PEC error	PECERR	
Timeout/Tlow error	TIMEOUT	
SMBus Alert	SMBALERT	

SCL master clock generation

- The CCR bits are used to generate the high and low level of the SCL clock
- If the SCL line is low,(slave is stretching the bus) the *high level counter* stops until the SCL line is detected high.

HIGH LEVEL COUNTER

- Controls the duration of high phase of SCL
- Starts when SCL rises ->reaches max prgrammed value -> SCL becomes logic low

START condition

- Setting the START bit causes the interface to generate a Start condition

Once the Start condition is sent:

- The SB bit is set by hardware and an interrupt is generated if the ITEVFEN bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address

Slave addressing

- In 7-bit addressing mode, one address byte is sent.

  As soon as the address byte is sent,
  - The ADDR bit is set by hardware and an interrupt is generated if the ITEVFEN bit is set.

  Then the master waits for a read of the SR1 register followed by a read of the SR2 register

The master can decide to enter Transmitter or Receiver mode depending on the LSB of the slave address sent.

- In 7-bit addressing mode,
  - To enter Transmitter mode, a master sends the slave address with LSB reset.
  - To enter Receiver mode, a master sends the slave address with LSB set.

- When the acknowledge pulse is received, the TxE bit is set

Receiving the data

Master Receiver mode.
After each byte the interface generates in sequence:
1. An acknowledge pulse if the ACK bit is set
2. The RxNE bit is set

Stopping communication

The master sends a NACK for the last byte received from the slave.
1. To generate the nonacknowledge pulse after the last received data byte, the ACK bit must be cleared just after reading the second last data byte (after second last RxNE event).
2. To generate the Stop/Restart condition, software must set the STOP/START bit just after reading the second last data byte (after the second last RxNE event).

BASIC UNDERSTANDING OF 12C PERIPHERAL OF STM32

Register Summary

The I2C Control Register 1 (I2C_CR1)
PE (Bit 0): Peripheral Enable. Set to 1 to enable the I2C peripheral.

ACK (Bit 10): Acknowledge Enable. Used by the master to enable an ACK response from the slave for data reception.

STOP (Bit 9): Stop Generation. A software-set bit that triggers the hardware to generate a STOP condition. Once the STOP condition is sent, the peripheral automatically clears the MSL bit in I2C_SR2, transitioning to slave mode.

START (Bit 8): Start Generation. A software-set bit that triggers the hardware to generate a START condition. The peripheral sets the MSL bit in I2C_SR2, entering master mode. If the peripheral is already in master mode, setting this bit generates a Repeated-Start condition.

NOSTRETCH (Bit 7): No Clock Stretching Enable. Set to 1 to disable the clock stretching feature in slave mode, preventing a slave from holding the SCL line low.

The I2C Control Register 2 (I2C_CR2)
This register primarily handles peripheral clock frequency and interrupt configuration :

FREQ[5:0] (Bits 5-0): Peripheral Clock Frequency. This field must be set to the value of the APB peripheral clock in MHz.

ITERREN (Bit 8): Error Interrupt Enable. Enables an interrupt for error conditions such as Bus Error (BERR), Arbitration Lost (ARLO), and Acknowledge Failure (AF).

ITEVTEN (Bit 9): Event Interrupt Enable. Enables an interrupt for key communication events like START bit sent, address sent, and byte transfer finished.

ITBUFEN (Bit 10): Buffer Interrupt Enable. Enables an interrupt when the data register is empty (TxE) or not empty (RxNE).

The I2C Clock Control Register (I2C_CCR)
The I2C_CCR register is essential for configuring the SCL clock speed in master mode :

F/S (Bit 15): Master Mode Selection. 0 for Standard mode, 1 for Fast mode.

DUTY (Bit 14): Fast Mode Duty Cycle. Used to select the duty cycle of the SCL clock in Fast mode

CCR[11:0] (Bits 11-0): Clock Control. This field contains the clock value that determines the SCL frequency. The value is calculated based on the peripheral clock frequency and the desired baud rate.

The I2C Data Register (I2C_DR)

The I2C_DR is a bidirectional 8-bit register for data transfer.
In master transmitter mode, the software writes the data to be transmitted into this register.In master receiver mode, the software reads this register to retrieve the data received from the slave.This register is also used during the address phase of a transaction to transmit the slave's 7-bit address and the Read/Write bit.

I2C Status Registers

These two registers provide real-time status information about the I2C peripheral and bus. The software's interaction with the hardware state machine is entirely dependent on polling and responding to these flags.

I2C_SR1 Flags :

SB: Start Bit sent. Set after the hardware generates a START condition.

ADDR: Address sent/received. Set after the slave's address has been transmitted and an ACK has been received.

TxE: Transmit buffer empty. Indicates that the I2C_DR is ready to accept the next byte.

RxNE: Receive buffer not empty. Indicates that a new byte has been received and is available in I2C_DR for reading.

BTF: Byte Transfer Finished. Set when a byte transfer has been completed and the hardware is waiting for the next action.

AF: Acknowledge Failure. An ACK was expected but a NACK was received.

ARLO: Arbitration Lost. Indicates that another master has taken control of the bus.BERR: Bus Error. A misplaced START or STOP condition was detected on the bus..

I2C_SR2 Flags :

MSL: Master/Slave mode. A 1 indicates the peripheral is in master mode.
BUSY: Bus busy. A 1 indicates that a START condition has been generated and the bus is occupied

Checklist for Master Configuration

REGISTER->FIELD/BIT->VALUE/FORMULA->PURPOSE

RCC_APB1ENR -> I2C1EN (Bit 21) -> 1 -> Enable the I2C1 peripheral clock.

I2C_CR1 -> PE (Bit 0) -> 0 -> Disable the peripheral before configuration.

I2C_CR2 -> FREQ[5:0] ->PCLK1 in MHz -> Set the peripheral clock frequency for timing calculations.

I2C_CCR -> F/S (Bit 15)-> 0 for 100 kHz; 1 for 400 kHz ->Select Standard or Fast mode.

I2C_CCR -> CCR[11:0] -> PCLK1 / (2 * 100 kHz) for 100 kHz; PCLK1 / (3 * 400 kHz) for 400 kHz -> Calculate the clock control value.

I2C_TRISE -> TRISE[5:0] -> (PCLK1 / 1 MHz) + 1 -> Configure bus rise time.

I2C_CR1 -> PE (Bit 0)-> 1->Enable the peripheral after all configuration is complete.