Master Synchronous Serial Port

Periférico responsável pela configuração do SPI e do I2C do microcontrolador.

Registradores

- SSPxSTAT → configurações gerais do MSSP
- SSPCON 1 e 2 → configurações gerais do MSSP
- SSPxADD → Este registrador é de endereçamento (modo slave) ou define o baudrate da comunicação (modo master)
- SSPBUF → Registrador de envio ou recebimento de dado

O modo SPI pode trabalhar com 3 pinos do microcontrolador no modo MASTER

- Serial Data Out (SDO) -RC5/SDO
- Serial Data In (SDI) - RC4/SDI/SDA
- Serial Clock (SCK) - RC3/SCK/SCL

Ou com 4 pinos no modo SLAVE, quando é acrescentado à função Slave Select (SS) - RA5.

As diferenças entre os protocolos de comunicação são:

Tecnologia	Barramento de comunicação	Taxa máxima	Fluxo de dados
UART (RS232)	2	115,2 kbps	Half ou Full Duplex
SPI	3 + número de Slaves	2 Mbps	Full duplex
I2C	2 (até 127 dispositivos)	400 kbps	Half duplex

Protocolo I2C

Teoria

O I2C é apropriado para periféricos onde simplicidade e baixo custo são mais importantes que velocidade. Algumas aplicações são:

- Descrever dispositivos conectados via pequenas tabelas de configuração ROM para habilitar operações plug and play, como detecção de presença serial de EEPROMs em módulos de memória dual in-line, e Dados de identificação de display extendido para monitores VGA, DVI e HDMI
- Monitoramento de sistema para PC via SMBus. SMBus é um barramento simples de dois fios de terminação única para fins de comunicação leve. É comumente encontrado em placas-mãe de computador para comunicação com a fonte de alimentação para instruções ON/OFF.
- Acesso à clocks de tempo de real e chips de NVRAM que armazenam configurações
- Acessar DACs e ADC de baixa velocidade
- Configurações de monitores: luz de fundo; contraste, balanço de cor, tonalidade
- Volume em alto-falantes inteligentes
- Controlar pequenos displays LCD e OLED
- Sensores
- On/Off de fontes de componentes do sistema

A quantidade de nodes em que podem compartilhar um canal de comunicação I2C é limitado ao número de endereçamento compatível e a capacitância máxima de 400pF, o que restringe a comunicação para poucos metros.

Introdução

O I2C é um protocolo de comunicação muito utilizado para unir periféricos de baixa velocidade à microcontroladores em curtas distâncias, geralmente na mesma placa. A comunicação utiliza somente duas vias: uma de Clock (SCL) e uma de Data (SDA).

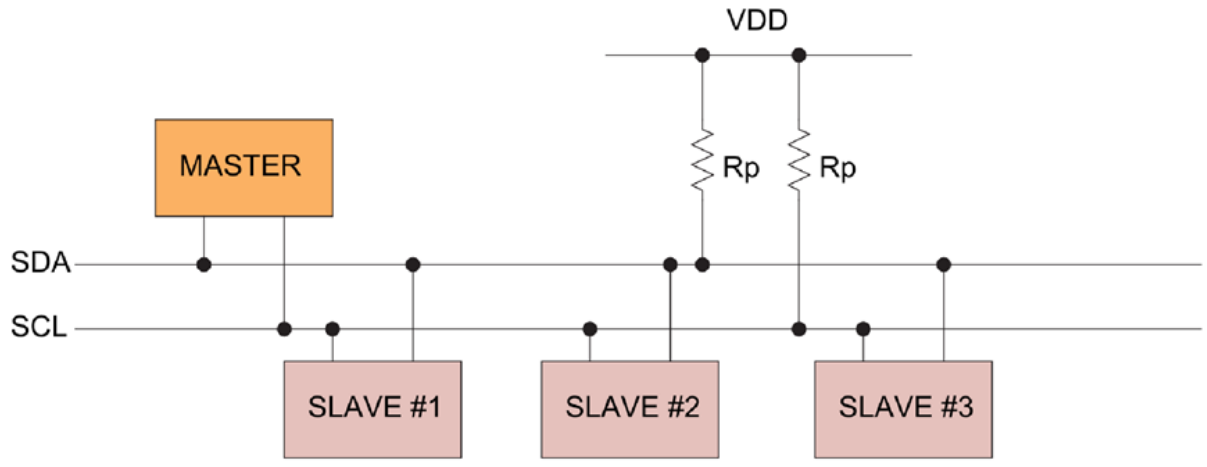
- Apenas dois canais de comunicação são necessários
- Usa o paradigma de Master/Slave.
- Até 127 dispositivos no modo endereçamento de 7-bit
- Barramento deve ter capacitância máxima de 400pF
- Leitura de dados somente com o clock em nível alto

O protocolo é perfeito para aplicações onde o baixo custo e simplicidade são mais importantes que a velocidade. A capacidade de resumir vários barramentos de 8 bits em apenas 2 é sua principal característica. Alguns exemplos de utilização:

- Acelerômetros
- Displays LCD com módulo serial
- Memórias
- Outros microcontroladores

Como funciona o protocolo

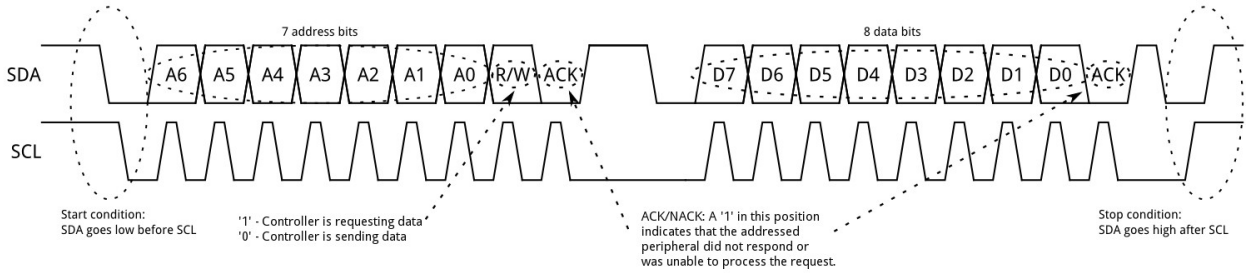
Na comunicação I2C, o byte é enviado em somente 1 canal. Este canal é chamado de **SDA** e está ligado em paralelo com todos os dispositivos da rede, ou seja, todos os dispositivos receberão o mesmo byte. Ainda, os dois canais devem possuir resistores de Pull-Up conectados, geralmente de 2.2K, (os valores são padronizados) para elevar o estado lógico do pino do master quando ele estiver configurado em alta impedância (entrada).



Fonte: https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.html

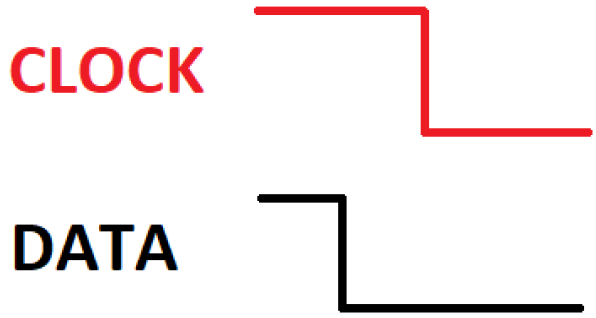
Para que o slave seja capaz de ler os bits corretamente, o clock é utilizado para sincronizar esta leitura. Sempre quando o clock estiver em alta, uma leitura é realizada. Se o clock subir com SDA em alta é lido 1, se o clock subir com SDA em baixa, é lido 0.

Além dos bytes, existem alguns bits que possuem funções específicas no protocolo como: início da comunicação; fim da comunicação; reconhecimento da comunicação. Estes bits serão detalhados melhor a frente. Veja abaixo o processo de comunicação completo:

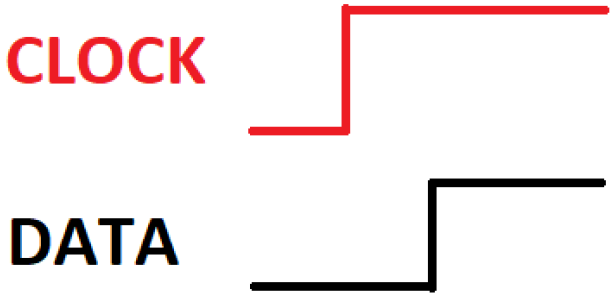


Bits de sinalização

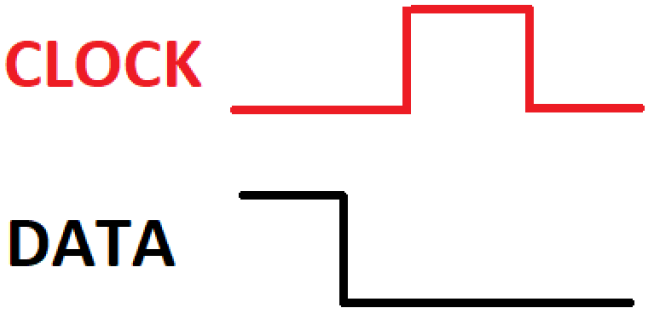
- START bit→ Indica o início da comunicação para o Slave. O Slave reconhece o bit de start quando recebe um sinal de Clock (SCL) em alta e um sinal de data (SDA) caindo. A partir daí, ele estará pronto para receber o primeiro byte com o endereçamento e o bit R/W.



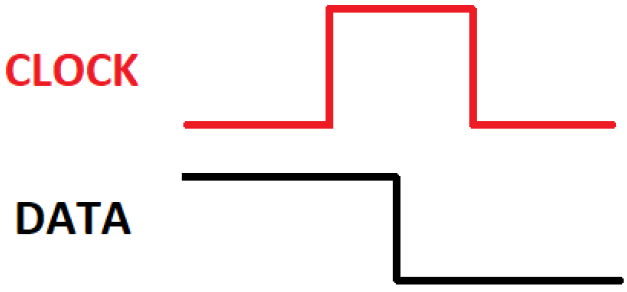
- STOP bit→ Indica o fim da comunicação para o Slave. Clock (SCL) estando em alta e data (SDA) subir



- ACK bit→ bit de confirmação. É o bit que indica ao Master que o Slave recebeu o dado e está pronto para receber mais um. Este bit está no 9º pulso de clock (SCL) da comunicação. Enquanto data (SDA) está em baixa, o clock (SCL) realiza um pulso completo.

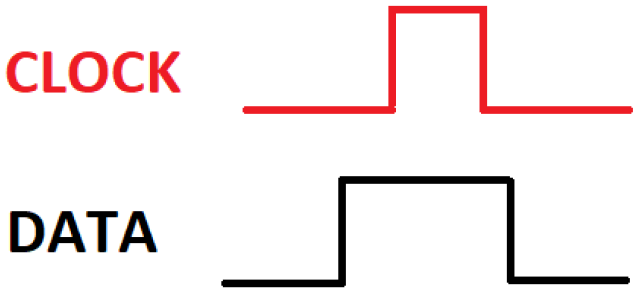


- Repeated start bit → Uma condição de stop seguida de outra de start. É utilizado no modo combinado no Repeated Start para leitura de um endereço específico de memória.



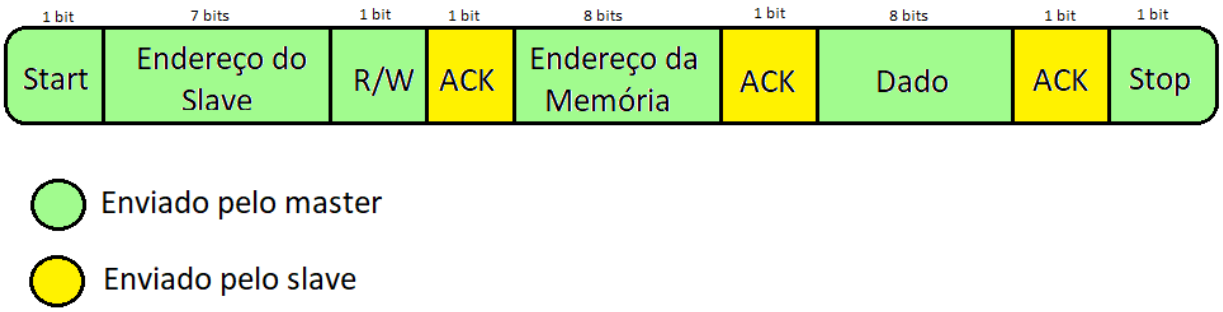
- NACK bit → Negação, finalização. Este bit indica o fim do processo de comunicação, quando o slave recebe este bit ele para de incrementar a sua

memória interina. Um pulso no SCL quando SDA estiver alto



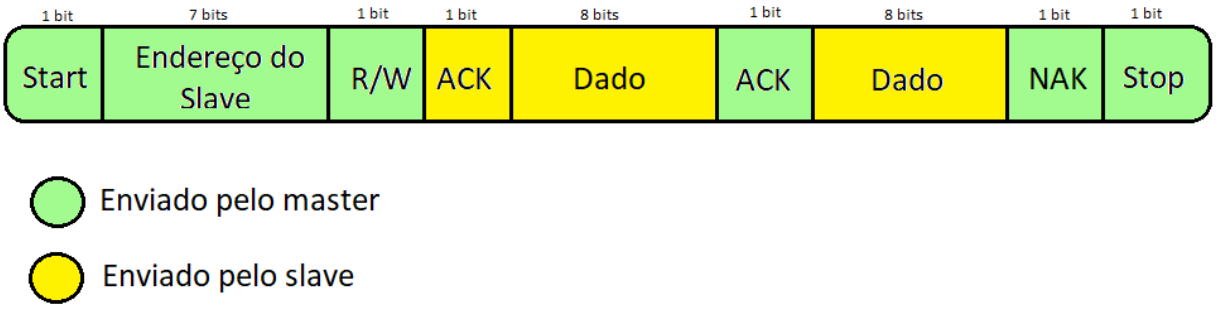
Exemplo - Escrita em uma memória EEPROM

O processo começa com um bit de start enviado pelo Master seguido pelo endereço do Slave + bit de Write. O Slave retorna ACK para dizer ao Master que entendeu e espera um endereço de memória. O Master envia este endereço de memória, o Slave retorna outro ACK, e o Master passa a enviar os dados. A comunicação continua até que o Master envie o bit de Stop.



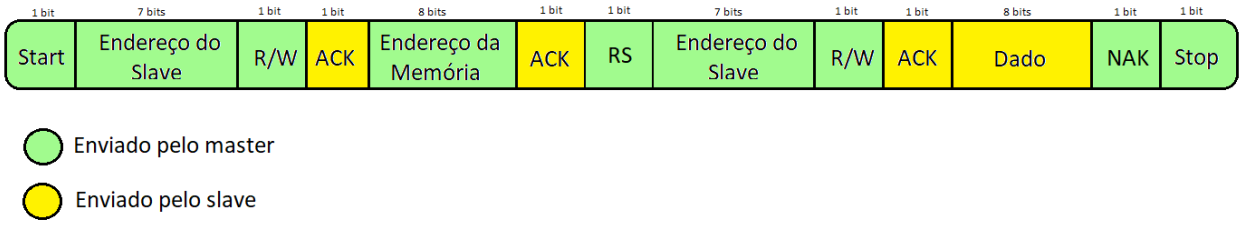
Exemplo - Leitura de uma memória EEPROM

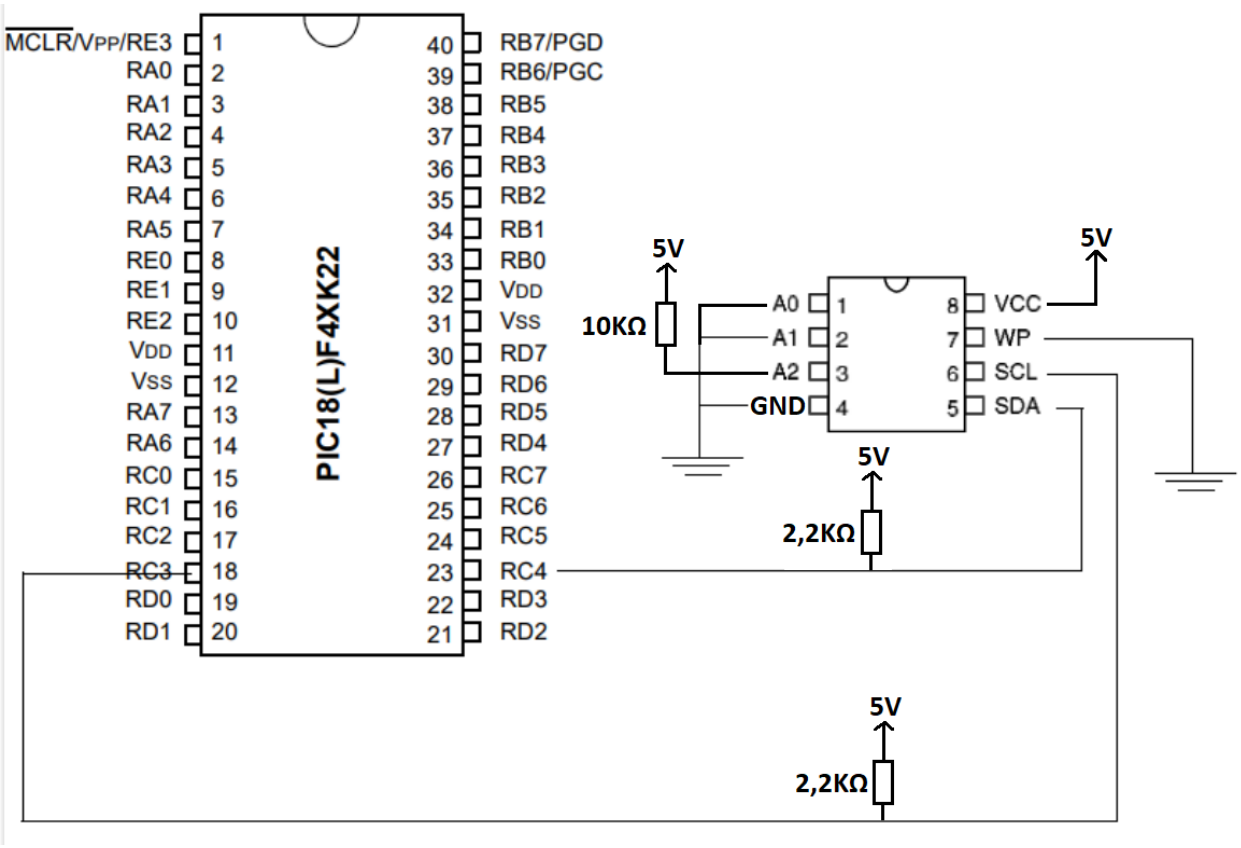
O modo de leitura padrão ocorre de forma semelhante ao de escrita. A diferença é que agora, após o primeiro ACK, é o Slave que passa a enviar dados e o Master responde com ACK. A cada ACK enviado pelo Master, o Slave incrementa seu endereço de memória e envia o byte correspondente. Este processo se repete até que o Master envie o NAK, informando que não quer mais receber dados.



O problema do modo acima é que o escravo não espera receber um endereço de memória, como ocorre no modo escrita, ele retorna os dados a partir do primeiro endereço. Para contornar esta situação, que em muitos casos pode ser um problema, existe o modo combinado.

O modo combinado é chamado assim pois combina os processos de escrita e leitura. A comunicação inicia como um processo de escrita padrão, mas sofre um **Restart** logo após o envio do endereço de memória. O endereço fica gravado no Slave, mas ele não sabe mais se será escrito ou lido. A partir deste ponto, o processo de leitura pode ocorrer normalmente.





Com esta memória, o próprio projetista pode definir qual será o endereço dela (pelo menos em parte). No caso de comunicação entre dois microcontroladores, você pode fazer este endereçamento via software, mas como não é possível gravar software dentro de uma EEPROM, criamos este endereçamento com os pinos A0, A1 e A2. Dos 8 bits de endereçamento, os 4 bits mais significativos vem pré-definidos de fábrica como 1010, o pino menos significativo é quem indica modo Read(1) ou Write(0) R/W. Os últimos três são os pinos físicos A0, A1 e A2. O formato do endereçamento fica assim:

Mas para o modelo FT24C08A específico, somente o A2 funciona como endereçamento, enquanto os pinos restantes não são utilizados. No software, você deve utilizar:

- Escrita → 10101000;

- Leitura → 10101001;

Registrador SSPxSTAT

REGISTER 15-2: SSPxSTAT: SSPx STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **SMP:** SPI Data Input Sample bit

SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time

SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode

In I^2C Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

bit 6 **CKE:** SPI Clock Edge Select bit (SPI mode only)

In SPI Master or Slave mode:
1 = Transmit occurs on transition from active to Idle clock state
0 = Transmit occurs on transition from Idle to active clock state

In I^2C mode only:
1 = Enable input logic so that thresholds are compliant with SMbus specification
0 = Disable SMbus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
(I^2C mode only. This bit is cleared when the MSSPx module is disabled, SSPxEN is cleared.)
1 = Indicates that a Stop bit has been detected last (this bit is '0' on Reset)
0 = Stop bit was not detected last

bit 3 **S:** Start bit
(I^2C mode only. This bit is cleared when the MSSPx module is disabled, SSPxEN is cleared.)
1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)
0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ bit information (I^2C mode only)
This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit, or not $\overline{ACK}$ bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
OR-ing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPxADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPxBUF is full
0 = Receive not complete, SSPxBUF is empty
Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPxBUF is empty

- SMP → Bit de controle do Slew Rate, velocidade da comunicação
- CKE → Borda do sinal de clock, somente para o modo SPI.
- D/A → Flag que indica se o ultimo byte recebido foi dado ou endereço
- P → Flag de detecção do bit de stop
- S → Flag de detecção do bit de start
- R/W → Bit de leitura ou escrita
- UA → Indica se o usuário precisa atualizar o endereço do registrador SSPxADD
- BF → O registrador SSPBUF está cheio, tanto na escrita quanto no recebimento

SSPxCON1

REGISTER 15-3: SSPxCON1: SSPx CONTROL REGISTER 1

R/C/HS-0	R/C/HS-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPxOV	SSPxEN	CKP	SSPxM<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets	
'1' = Bit is set	'0' = Bit is cleared	HS = Bit is set by hardware	C = User cleared

bit 7 **WCOL:** Write Collision Detect bit
Master mode:
1 = A write to the SSPxBUF register was attempted while the I^2C conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPxOV:** Receive Overflow Indicator bit(1)
In SPI mode:
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register (must be cleared in software).
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte. SSPxOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow

bit 5 **SSPxEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as the source of the serial port pins(2)
0 = Disables serial port and configures these pins as I/O port pins
In I^2C mode:
1 = Enables the serial port and configures the SDAx and SCLx pins as the source of the serial port pins(3)
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C Slave mode:
SCLx release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I^2C Master mode:
Unused in this mode

bit 3-0 **SSPxM<3:0>:** Synchronous Serial Port Mode Select bits

0000 = SPI Master mode, clock = FOSC/4
0001 = SPI Master mode, clock = FOSC/16
0010 = SPI Master mode, clock = FOSC/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control enabled
0101 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control disabled, $\overline{SSx}$ can be used as I/O pin
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1000 = I^2C Master mode, clock = FOSC / (4 * (SSPxADD+1))(4)
1001 = Reserved
1010 = SPI Master mode, clock = FOSC/(4 * (SSPxADD+1))
1011 = I^2C firmware controlled Master mode (slave idle)
1100 = Reserved
1101 = Reserved
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

- WCOL → Colisão de escrita. Escrever no registrador SSPxBUF quando esta cheio
- SSPOV → Colisão de recebimento. Receber outro dado ser ler SSPxBUF antes
- SSPEN → Habilita o módulo MSSP
- CKP → O slave pausa o clock da rede para que possa processar o dado recebido
- SSPxM → Seleciona o modo de operação

SSPxCON2

REGISTER 15-4: SSPxCON2: SSPx CONTROL REGISTER 2

R/W-0	R-0	R/W-0	R/S/HC-0	R/S/HC-0	R/S/HC-0	R/S/HC-0	R/W/HC-0
GCEN	ACKSTAT	ACKDT	ACKEN(1)	RCEN(1)	PEN(1)	RSEN(1)	SEN(1)
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Cleared by hardware S = User set

bit 7 **GCEN:** General Call Enable bit (in I^2C Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPxSR
0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I^2C mode only)
1 = Acknowledge was not received
0 = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in I^2C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN(1):** Acknowledge Sequence Enable bit (in I^2C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDAx and SCLx pins, and transmit ACKDT data bit. Automatically cleared by hardware.
0 = Acknowledge sequence idle

bit 3 **RCEN(1):** Receive Enable bit (in I^2C Master mode only)
1 = Enables Receive mode for I^2C
0 = Receive idle

bit 2 **PEN(1):** Stop Condition Enable bit (in I^2C Master mode only)
SCKx Release Control:
1 = Initiate Stop condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Stop condition Idle

bit 1 **RSEN(1):** Repeated Start Condition Enabled bit (in I^2C Master mode only)
1 = Initiate Repeated Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle

bit 0 **SEN(1):** Start Condition Enabled bit (in I^2C Master mode only)
In Master mode:
1 = Initiate Start condition on SDAx and SCLx pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

- GCEN → Habilita interrupção por comunicação I2C no modo slave
- ACKSTAT → Estado do bit ACK recebido
- ACKDT → Estado do bit ACK enviado

- ACKEN → Transmite o bit ACK na rede
- RCEN → Habilita o modo de recepção do I2C
- PEN → Envia o bit de stop
- RSEN → Envia o bit de Repeated Start
- SEN → Envia o bit de start (master) - Habilita o modo Clock streching (slave)

Registrador SSPADD

No modo **Master**, este registrador armazena o BaudRate da comunicação:

```
SSP1ADD = (_XTAL_FREQ/1000000/4) - 1; // baudrate para 100KHz
```

No modo **slave**, este registrador indica o endereço do microcontrolador em uma rede I2C.

Interrupções do MSSP

Os bit de interrupção são os mesmos para I2C e para SPI. No entanto, a ocorrência deles se da de forma diferente. No modo I2C, a sinalização da interrupção (SSPIF) se dá pelos seguintes motivos:

- Condição de Início
- Condição de Parada
- Transmissão ou recepção completada
- Transmissão do Acknowledge
- Condição de Repetead Start