RS485 / SPI / I²C

Jun Okamoto Jr.

RS485

Características Gerais

- Usado em sistemas de controle industrial
  - ex: Modbus, Profibus
- Somente padrão de interface elétrica (≠ RS232 que padroniza tudo)
  - transmissão diferencial com 2 fios
- Longas distâncias (até 1.200 m ≅ 4.000 ft.)
- Master / slave (half-duplex)
  - Full-duplex possível com 4 fios
- 32 transmissores / 32 receptores (256 possível)

Padrão RS485

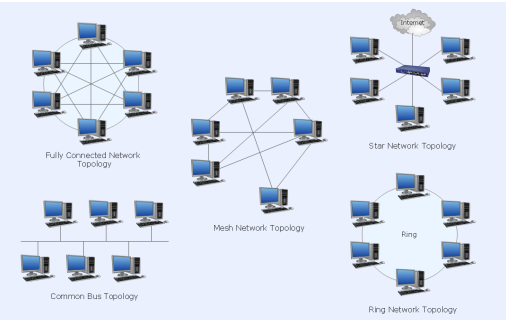
- Lógico (não padrão)
  - Assíncrono, n-bits (8 ou 9), taxa de baud (= RS232)
- Elétrico (padrão)
  - Driver diferencial, sinais A e B (≠ RS232)
  - H: $V_{OA}–V_{OB} < –200$ mV; L: $V_{OA}–V_{OB} > +200$ mV
- Mecânico (não padrão)
  - normalmente, bone 2 pinos (≠ RS232)

RS232 x RS485

Characteristics of RS232 and RS485		
Parameter	**RS232**	**RS485**
Cabling	Single-edned	Differential
Numbers of devices	1 transmitter 1 receiver	32 transmitters 32 receivers
Mode of operation	Simplex or full duplex	Simplex or half duplex
Maximum cable length	50 feet	4000 feet
Maximum data rate	20 kbits/s	10 Mbits/s
Signaling	unbalanced	balanced
Typical logic levels	±5 ~ ±15V	±1.5 ~ ±6V
Minimun receiver input impedance	3~7kΩ	12kΩ
Receiver sensitivity	±3V	±200mA

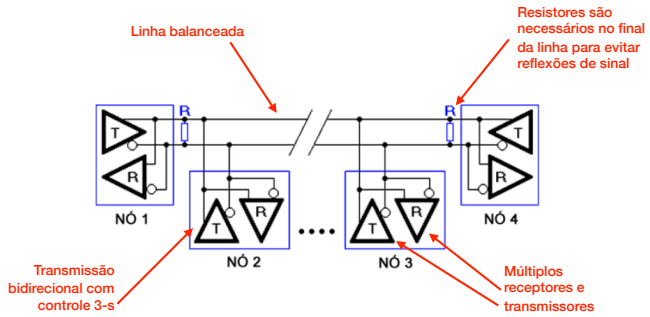
http://www.sopto.com/protocol_converter_tdmoip/article-3611.shtml

Topologias de rede



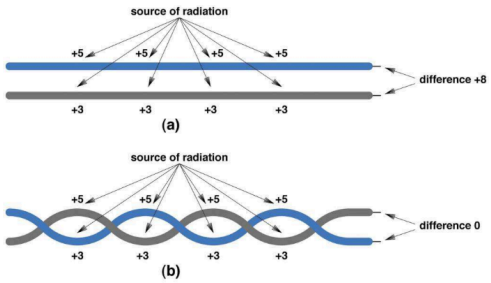
http://www.conceptdraw.com/How-To-Guide/picture/Computer-and-networks-Common-network-topologies.png

Topologia da rede para RS485: Barramento ou bus



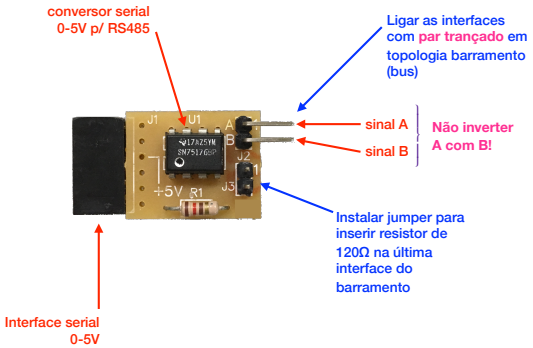
https://www.citisystems.com.br/rs485/

Par trançado: imunidade ao ruído



https://www.eeweb.com/quizzes/twisted-pairs

RS485 no PI-7

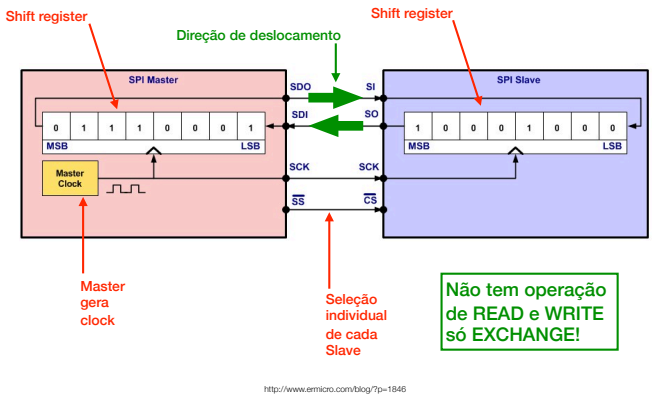


SPI

Características Gerais

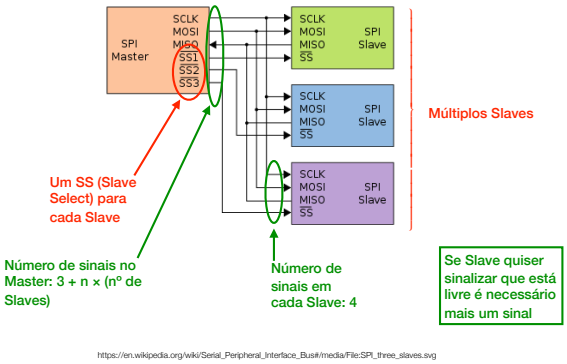
- Master / Slave (~full-duplex)
- Síncrono, n-bits de dados, bidirecional
- Sinais: MOSI, MISO, SCLK, SS
- Taxa de transferência de MHz
- Ajuste de polaridade e fase do clock do Master de acordo com o suportado pelo Slave
- Implementação de software simples

Funcionamento básico



http://www.ermicro.com/blog/?p=1846

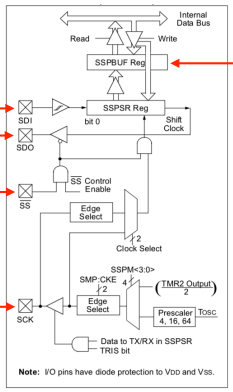
Topologia



https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus#/media/File:SPI_three_slaves.svg

Módulo SPI no PIC 16F886 (Master ou Slave)

Figura 13-1: MSSP Block Diagram (SPI mode) do *data sheet*, p. 187



Buffer de transmissão e recepção

Pino RC4

Pino RC5

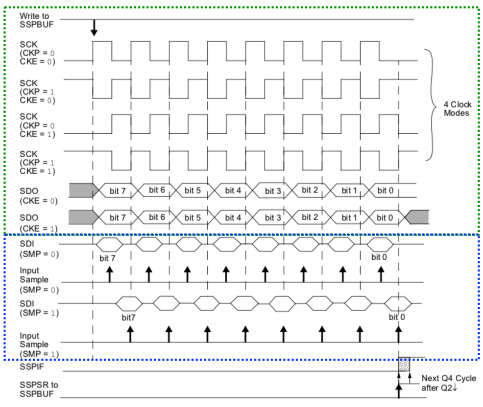
Pino RA5
entrada só em modo SLAVE

Pino RC3
entrada em modo SLAVE
saída em modo MASTER

```
// Para usar o módulo SPI do PIC
// deve-se programar os TRISxx

// O TRISC4 para o SDI é
// controlado automaticamente
// pelo módulo SPI do PIC

TIRSC5 = 0; // SDO é saída

TRISC3 = 0; // SCK é saída
            // se for modo
            // MASTER ou
TRISC3 = 1; // SCK é entrada
            // se for modo SLAVE

TRISA5 = 1; // SS é entrada e é
            // somente usado em
            // modo SLAVE
```

Modo Master: Clock do SPI e Amostragem do SDI

Figura 13-2: SPI Mode waveform (Master mode) do *data sheet*, p. 189



Programação no modo Master de acordo com o que o Slave consegue entender

Frequência de SCK:
- Fosc/4
- Fosc/16
- Fosc/64
- saída do Timer 2 / 2

Máximo: 10 Mbps

Amostragem do SDI

Programação do SPI no PIC

REGISTER 13-1: SSPSTAT: SSP STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

SMP: 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

BF: Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

CKE: SPI Clock Edge Select bit
CKP = 0:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK
CKP = 1:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK

Programação do SPI no PIC

REGISTER 13-2: SSPCON: SSP CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

SSPOV: Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, 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 SSPBUF register (must be cleared in software).
0 = No overflow

SSPEN: 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 SCK, SDO, SDI and $\overline{SS}$ as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

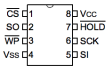
CKP: Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

SSPM<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 = SCK pin, $\overline{SS}$ pin control enabled
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin

Exercício

- Programar módulo SPI do PIC 16F886, com clock de 20MHz, em modo MASTER para acessar a memória EEPROM 25LCxxxx

PDIP/SOIC
(P, SN, SM)



Pin Function Table

Name	Function
$\overline{CS}$	Chip Select
SO	Serial Data Output
$\overline{WP}$	Write-Protect
Vss	Ground
SI	Serial Data Input
SCK	Serial Clock Input
$\overline{HOLD}$	Hold Input
Vcc	Supply Voltage

FIGURE 1-2: SERIAL INPUT TIMING

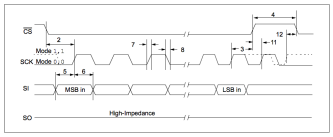
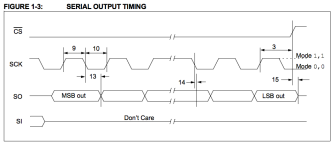


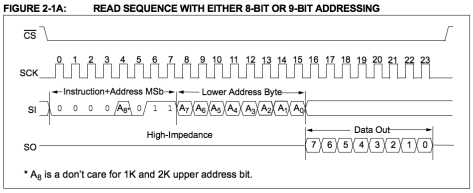
FIGURE 1-3: SERIAL OUTPUT TIMING



http://ww1.microchip.com/downloads/en/DeviceDoc/22040A.pdf

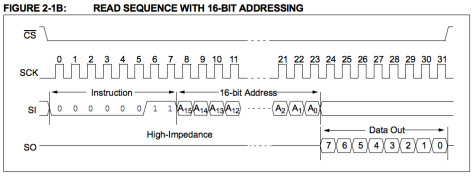
Leitura de dados na EEPROM 25LCxxxx

FIGURE 2-1A: READ SEQUENCE WITH EITHER 8-BIT OR 9-BIT ADDRESSING



* A_8 is a don't care for 1K and 2K upper address bit.

FIGURE 2-1B: READ SEQUENCE WITH 16-BIT ADDRESSING



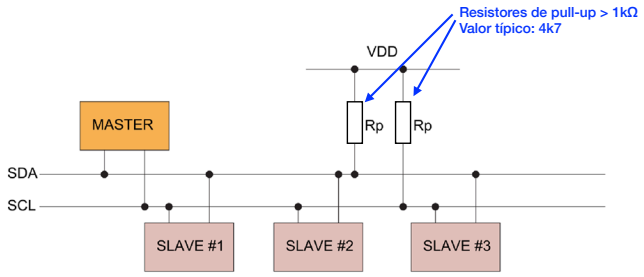
http://ww1.microchip.com/downloads/en/DeviceDoc/22040A.pdf

I²C

Características Gerais

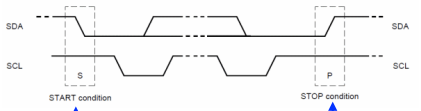
- I²C = Inter-Integrated Circuit
- Síncrono, multi-master / multi-slave, 8-bits de dados, 7 ou 10-bits de endereços de slaves
- Sinais: SDA, SCL (bidirecionais, **open-collector** ou **open-drain**)
- Velocidades: até 100kbps (modo padrão), 400kbps (modo fast)
- Arbitração em modo multi-master
- Limite do número de slaves:
  - Espaço de endereçamento (127 p/ end. 7-bits ou 1023 p/ end. 10-bits)
  - Capacitância da linha deve ser < 400 pF
- Implementação de software mais complexa do que SPI

Topologia (single MASTER)



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

Condição de Start e Stop

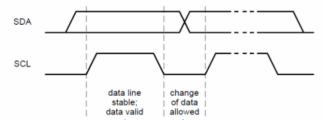


START — MASTER coloca SDA em LOW enquanto mantém SCL em HIGH

STOP — MASTER libera SDA para HIGH enquanto mantém SCL em HIGH

http://i2c.info/i2c-bus-specification

Transferência de dados

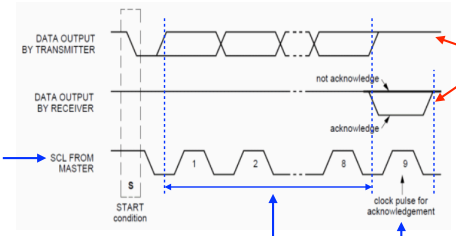


Dados permanecem válidos enquanto SCL em HIGH

Dados podem mudar quando SCL em LOW

http://i2c.info/i2c-bus-specification

Acknowledge



É só uma linha: SDA

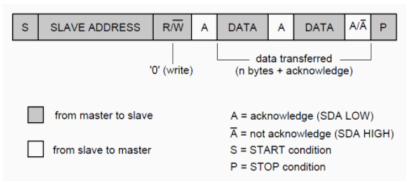
Sempre são 9 pulsos de clock

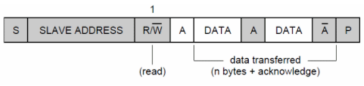
Sempre são transferidos 8-bits no 9º bit deve vir um ACK senão o MASTER interrompe a transferência e sinaliza a condição STOP

ACK: no 9º bit de clock o SLAVE puxa o SDA para LOW

http://i2c.info/i2c-bus-specification

Transferências de dados entre Master e Slave

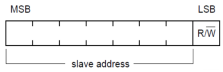




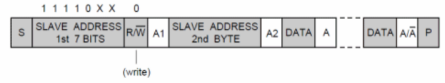
http://i2c.info/i2c-bus-specification

Endereçamento de 7 e 10-bits

- Endereçamento de 7-bits

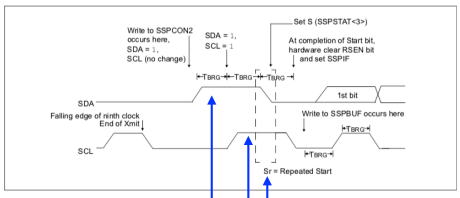


- Endereçamento de 10-bits



http://i2c.info/i2c-bus-specification

Modo MASTER: Condição Repeat Start



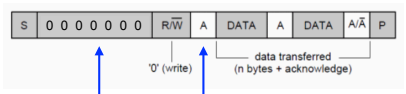
MASTER faz SDA HIGH enquanto mantém SCL em LOW por 1 período de clock

MASTER faz SCL HIGH por 1 período de clock

MASTER faz SDA LOW enquanto mantém SCL em HIGH por 1 período de clock

Figura 13-14: Repeat Start Condition Waveform do *data sheet* do PIC 16F886, p. 201

Modo MASTER: General Call



1. MASTER envia endereço com 7-bits ZERO em operação de WRITE

2. Neste caso todos os SLAVEs recebem a mensagem

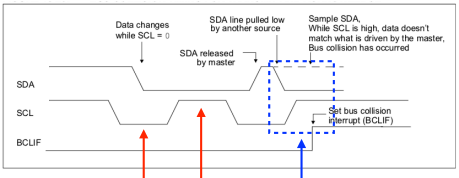
3. O SLAVE que for tratar a mensagem responde com ACK. Pode ser mais de um.

4. O MASTER não consegue distinguir quais SLAVEs responderam com ACK, ma continua a transmissão de dados

adaptado de http://i2c.info/i2c-bus-specification

Modo multi-MASTER: Colisão e arbitragem

- Exemplo de colisão em SDA



Lembrando que SDA pode mudar quando SCL é LOW

e SDA deve ficar estável quando SCL é HIGH

1. MASTER coloca SCL em HIGH

2. Em todo SDA HIGH, o MASTER confere o SDA. Se estiver diferente do que deveria, uma colisão é detectada

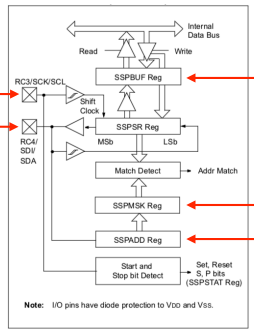
Figura 13-20: Bus collision timing for transmit and acknowledge do *data sheet* do PIC 16F886, p. 207

Módulo I²C no PIC 16F886

- Funciona em modo Master ou modo Slave
- Só funciona por interrupção
- Tratamento automático para:
  - Repeated start
  - General Call
  - Collision

Módulo I²C no PIC 16F886 (Slave)

Figura 13-6: MSSP Block Diagram (I²C mode) do *data sheet*, p. 193



Pino RC3

Pino RC4

Buffer de transmissão e recepção

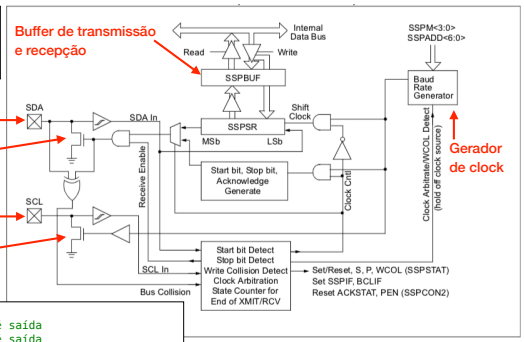
Máscara de endereço do SLAVE

Registrador de endereço do SLAVE

```
// Modo SLAVE
TRISC3 = 1; // SCL é
            // entrada
TRISC4 = 1; // SDA é
            // entrada

// o módulo I2C
// reconfigura o RC4
// para saída no caso
// de transmissão de
// dados do SLAVE
```

Módulo I²C no PIC 16F886 (Master)

Figura 13-10: MSSP Block Diagram (I²C master mode) do *data sheet*, p. 197



Pino RC4

Saída open-drain

Pino RC3

Saída open-drain

Gerador de clock

```
// Modo MASTER
TRISC3 = 0; // SCL é saída
TRISC4 = 0; // SDA é saída

// o módulo I2C reconfigura o RC4 para entrada
no caso de transmissão de dados do SLAVE
```

Programação do SPI no PIC

REGISTER 13-1: SSPSTAT: SSP STATUS REGISTER

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

SMP: Sample bit
In I²C 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)

D/A: Data/Address bit (I²C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

P: Stop bit
(I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN 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

S: Start bit
(I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN 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

R/W: Read/Write bit information (I²C mode only)
In I²C Slave mode:
1 = Read
0 = Write
In I²C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
1 = Transmit is in progress
0 = Transmit is not in progress

UA: Update Address bit (10-bit I²C mode only)
1 = Indicates that the user needs to update the address in the
0 = Address does not need to be updated SSPADD register

BF: Buffer Full Status bit
Receive (SPI and I²C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit (I²C mode only):
1 = Data transmit in progress, SSPBUF is full
0 = Data transmit complete, SSPBUF is empty

Programação do SPI no PIC

REGISTER 13-2: SSPCON: SSP CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

WCOL: Write Collision Detect bit
Master mode:
1 = A write to the SSPBUF register was attempted while the I²C conditions were not valid for a transmission to be started
0 = No collision
Slave mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

SSPOV: Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, 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 SSPBUF register (must be cleared in software).
0 = No overflow

SSPEN: Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output
In I²C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

CKP: Clock Polarity Select bit
In I²C Slave mode:
SCK release control
1 = Release clock
0 = Holds clock low (clock stretch).
In I²C Master mode:
Unused in this mode

SSPM<3:0>: Synchronous Serial Port Mode Select bits
0110 = I²C Slave mode, 7-bit address
0111 = I²C Slave mode, 10-bit address
1000 = I²C Master mode, clock = FOSC / (4 * (SSPADD+1))
1001 = Load Mask function
1010 = Reserved
1011 = I²C firmware controlled Master mode (Slave idle)
1100 = Reserved
1101 = Reserved
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

Exercício para casa

- Configurar o módulo I²C do PIC 16F886 em modo SLAVE, com endereço de 7-bits.
  - Programar a máscara 0x7f e o endereço do SLAVE como sendo 0x55.
- Escrever a rotina de tratamento de interrupção para receber dados e colocá-los num buffer que possa ser acessado do programa principal
- Escrever a rotina de tratamento de interrupção para enviar dados de um buffer. A quantidade de bytes a serem transmitidos é determinada por uma variável global.