

I2C sembra facile ma ...

In rete si trovano Librerie I2C e qualche esempio quasi SEMPRE centrato sulla comunicazione tra MASTER e SLAVE ma con uno slave costituito da una Memoria Eprom o da un Orologio.

Io devo far comunicare due PIC uno Master PIC18F4550 (con funzione MSPP) e uno Slave PIC16F88 (Con funzione SSP).

Purtroppo questa comunicazione non è identica a quella che avviene con EEPROM o OROLOGI nel senso che il PIC SLAVE svolge SUE operazioni e deve comunicare con il MASTER attraverso INTERRUPT.

La chiave sembra stare TUTTA nella routine di **interrupt** dello SLAVE che ho scaricato dal sito <http://www.vincenzov.net/tutorial/elettronica-di-base/> e che ripropongo spiegando di seguito quali sono le anomalie.

Il codice Master esegue (si tratta delle routine i2c.h ed i2c.c prelevate dal C HITECH):

```
i2c_InitMaster();
RIS1=i2c_WriteTo( Indir_Slave );
RIS2=i2c_PutByte( Dato);
i2c_Stop();
```

Posso garantire che il bus funziona e con oscilloscopio ho testato l'output delle LINEE SCL SDA e anche se controllato rapidamente solo sui due Byte funziona correttamente.

Il Codice dello SLAVE e la routine di interrupt sono le seguenti:

```
// Usare il PIC16 come periferica I2C - Versione 0.1 - Gennaio 2010
// Copyright © 2009, VincenzoV.net (http://www.vincenzov.net)
// Creative Commons | Attribuzione-Condividi allo stesso modo 3.0 Unported.
// Creative Commons | Attribution-Share Alike 3.0 Unported
// http://www.vincenzov.net/tutorial/elettronica-di-base/

#include <htc.h>
#define _XTAL_FREQ 8000000
#define I2C_ADDR 0x51

__CONFIG(opportuno)

unsigned char DATO;

void main(void)
{
// inizializzazioni PIC16F88

//Configure as I2C slave
//xxxx 0110 = I2C Slave mode, 7 bit address, no interrupts on start and stop
//xxx1 xxxx = CKP = 1, Enable clock
//xx1x xxxx = SSPEN, Enable SSP
SSPCON = 0b00110110;
SSPADD = I2C_ADDR << 1; // Set I2C address
SSPSTAT = 0x00; // All bits must be cleared
SSPIE = 1; // Enable the SSP module and the SSP interrupt
SSPIF = 0;
PEIE = 1;
GIE = 1;
for(;;);
}

void interrupt isr(void) // Interrupt Service Routine
{if (SSPIF) // I2C interrupt
    {if(!DA && !RW && BF && START)
```

```

    { // ADDR da master: write to slave (1)
      SSPBUF = SSPBUF;
      if(SSPOV) // Check overflow
      { //..... // Overflow handling, if required
        SSPOV = 0;
      }
    }
  else if(DA && !RW && BF)
    { // DATO da master: write to slave (2)
      DATO = SSPBUF;
      if(SSPOV) // Check overflow
      { //..... // Overflow handling, if required
        SSPOV = 0;
      }
    }
  else if(!DA && RW && !BF)
    { // ADDR da master: read from slave (3)
      SSPBUF = DATO; // Data will be write to master
      CKP = 1; // Release clock (data ready to be send to master)
    }
  else if(DA && RW && !BF && START)
    { // Read un DATO dallo slave (next byte) (4)
      SSPBUF = DATO; // Data will be write to master
      CKP = 1; // Release clock (data ready to be send to
master)
    }
  else if(DA && !RW && !BF)
    { // NAK Da master (complete transmission has
occurred) (5)
    }
  }
  SSPIF = 0; // Re-enable interrupt
}

```

#### ANOMALIE:

1) Se Resetto entrambi i PIC ottengo (SEMPRE) nel main del PIC18F4550:  
 RIS1 = 0 // ACK che significa che L'indirizzo dello SLAVE e stato riconosciuto  
 RIS2 = 0 // ancora ACK Ma non doveva esser un NOT ACK ? Per il DATO

Soprattutto la Linea RB1 SDA (colpa del PIC88) RIMANE BASSA!!!

2) Se ora resetto solo Il PIC18F4550 ottengo  
 RIS1 = 0 // ACK che significa che L'indirizzo dello SLAVE e stato riconosciuto  
 RIS2 = 1 // Per il DATO

la Linea RB1 SDA del PIC88 e normalmente ALTA

3) Se ora resetto ANCORA solo Il PIC18F4550 ottengo  
 RIS1 = 1 // ACK che significa che L'indirizzo dello SLAVE NON è stato riconosciuto  
 RIS2 = 1 // Per il DATO

la Linea RB1 SDA del PIC88 e normalmente ALTA

4) Se ora resetto ANCORA riprende da 1).

Se è di qualche utilità ho testato la routine di interrupt e Ogni volta (nei 4 CASI) passa per rami DIVERSI DELLA IF.

Vi ringrazio per l'attenzione e ancora di più se qualcuno di voi ha INDICAZIONI o banalmente LIBRERIE SLAVE (con INTERRUPT) FUNZIONANTI .

Giovanni