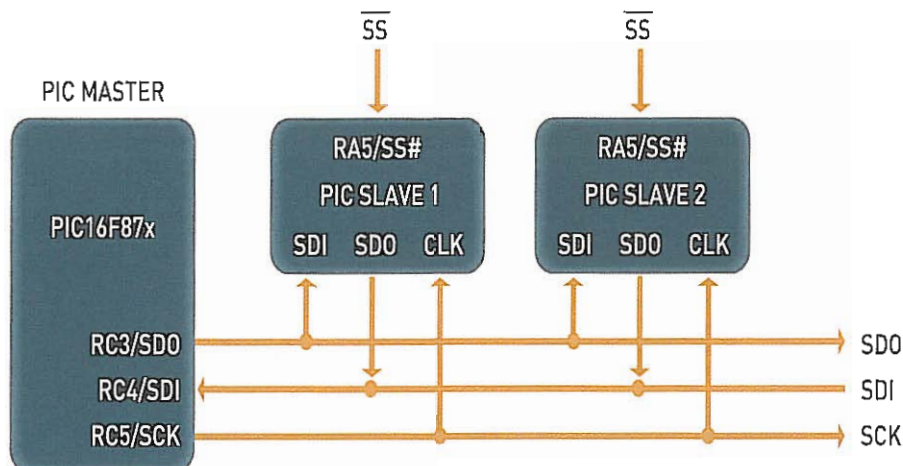


Comunicazione SPI (I)

Anche se i microcontroller nacquero con la filosofia di costruire un sistema "chiuso", cioè con delle risorse fisse autosufficienti senza la necessità di far uso di elementi esterni, con il tempo si è visto che, pur mantenendo valida la linea generale, a volte l'utilizzo di questi elementi incrementava moltissimo la potenza delle applicazioni. Questo è il caso della comunicazione seriale per il trasferimento dei dati digitali fra sistemi.



La comunicazione seriale con diverse periferiche amplia enormemente la potenza delle applicazioni.

La comunicazione seriale nei PIC

La famiglia dei PIC16F87X è dotata del modulo MSSP (Master Synchronous Serial Port) come dispositivo. Con esso si

facilita la comunicazione seriale fra microcontroller, oppure fra microcontroller e diversi dispositivi come memorie, convertitori A/D, ecc., tutto questo senza la necessità di implementare il protocollo

di comunicazione. L'unica restrizione sta nell'utilizzare i piedini della Porta C e i registri predisposti a questo scopo. La comunicazione che si ottiene con l'MSSP è seriale sincrona, e il protocollo può essere SPI (Serial Peripheral Interface) oppure I2C (Inter-Integrated Circuit).

In questo caso prenderemo in considerazione il primo tipo, dove i dispositivi fondamentali utilizzati sono le memorie (RAM ed EEPROM), che permettono di disporre di un immagazzinamento extra di dati nelle applicazioni che lo richiedono. Sono necessarie tre linee per poterla implementare.

REGISTRO SSPSTAT

SMP	CKE	D/A#	P	S	R/W#	UA	BF
7							0
SMP							
<ul style="list-style-type: none"> Il bit 7 del registro SSPSTAT determina il momento del campionamento del bit di ingresso. In modo Master SPI, se SMP = 1 il dato di ingresso si campiona alla fine dell'impulso di clock e se vale 0 alla metà del medesimo. In modo Slave il bit ASP non ha significato e deve valere 0. 							
CKE							
<ul style="list-style-type: none"> Con questo bit si seleziona il fronte di clock attivo. Se il bit CKP = 0 sul registro SSPCON significa che lo stato di inattività sul segnale del clock è quello basso, e invece se CKP = 1 il dato si trasmette sul fronte di salita di SCK e se CKP = 0 sul fronte di discesa. Se CKP = 1 lo stato di inattività sul clock sarà quello alto, quando CKP = 1 il dato si trasmette sul fronte di discesa di SCK e se CKP = 0 su quello di salita. 							
BF							
<ul style="list-style-type: none"> Quando vale 1 indica che il registro SSPBUF è pieno. 							

Distribuzione e utilizzo dei bit del registro SSPSTAT nella comunicazione in modo SPI.

Passi nella comunicazione

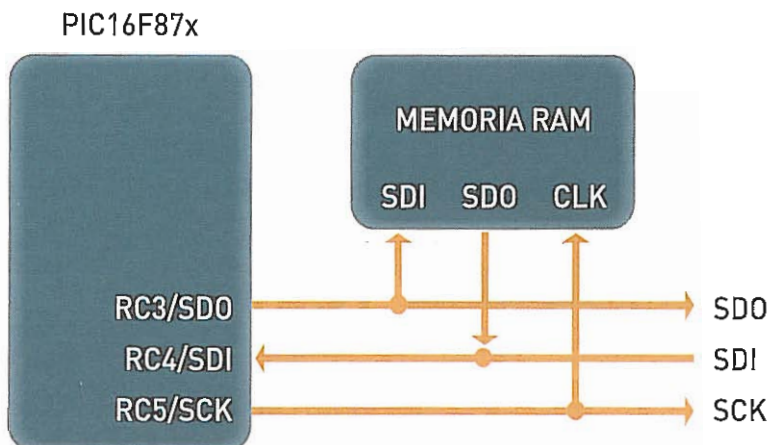
Grazie al fatto che il protocollo è già implementato all'interno del



REGISTRO SSPCON

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0																								
7							0																								
SSPM3-SSPM0																															
<ul style="list-style-type: none"> Con i 4 bit meno significativi del registro SSPCON si stabilisce il modo lavoro del modulo MSSP. 																															
SSPM3-0																															
<table border="0"> <tr> <td>0000</td> <td>Modo Master SPI, con clock a $F_{osc} / 4$</td> </tr> <tr> <td>0001</td> <td>Modo Master SPI, con clock a $F_{osc} / 16$</td> </tr> <tr> <td>0010</td> <td>Modo Master SPI, con clock a $F_{osc} / 64$</td> </tr> <tr> <td>0011</td> <td>Modo Master SPI, con clock uguale all'uscita del TMR2/2</td> </tr> <tr> <td>0100</td> <td>Modo Slave SPI, con clock uguale al pin SCK e SS# a livello basso</td> </tr> <tr> <td>0101</td> <td>Modo Slave SPI, con clock uguale al pin SCK, SS# non attivo e libero di essere utilizzato come I/O</td> </tr> <tr> <td>0110</td> <td>Modo Slave I2C, con indirizzo da 7 bit</td> </tr> <tr> <td>0111</td> <td>Modo Slave I2C, con indirizzo da 10 bit</td> </tr> <tr> <td>1000</td> <td>Modo I2C Master con clock = $F_{osc} / [4 * (SSPAD + 1)]$</td> </tr> <tr> <td>1011</td> <td>Modo I2C Master controllato da firmware</td> </tr> <tr> <td>1110</td> <td>Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 7 bit</td> </tr> <tr> <td>1111</td> <td>Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 10 bit</td> </tr> </table>								0000	Modo Master SPI, con clock a $F_{osc} / 4$	0001	Modo Master SPI, con clock a $F_{osc} / 16$	0010	Modo Master SPI, con clock a $F_{osc} / 64$	0011	Modo Master SPI, con clock uguale all'uscita del TMR2/2	0100	Modo Slave SPI, con clock uguale al pin SCK e SS# a livello basso	0101	Modo Slave SPI, con clock uguale al pin SCK, SS# non attivo e libero di essere utilizzato come I/O	0110	Modo Slave I2C, con indirizzo da 7 bit	0111	Modo Slave I2C, con indirizzo da 10 bit	1000	Modo I2C Master con clock = $F_{osc} / [4 * (SSPAD + 1)]$	1011	Modo I2C Master controllato da firmware	1110	Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 7 bit	1111	Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 10 bit
0000	Modo Master SPI, con clock a $F_{osc} / 4$																														
0001	Modo Master SPI, con clock a $F_{osc} / 16$																														
0010	Modo Master SPI, con clock a $F_{osc} / 64$																														
0011	Modo Master SPI, con clock uguale all'uscita del TMR2/2																														
0100	Modo Slave SPI, con clock uguale al pin SCK e SS# a livello basso																														
0101	Modo Slave SPI, con clock uguale al pin SCK, SS# non attivo e libero di essere utilizzato come I/O																														
0110	Modo Slave I2C, con indirizzo da 7 bit																														
0111	Modo Slave I2C, con indirizzo da 10 bit																														
1000	Modo I2C Master con clock = $F_{osc} / [4 * (SSPAD + 1)]$																														
1011	Modo I2C Master controllato da firmware																														
1110	Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 7 bit																														
1111	Modo I2C Master controllato per firmware, con bit di inizio e stop, attivato interrupt, indirizzo da 10 bit																														
CKP																															
<ul style="list-style-type: none"> Determina le polarità o lo stato di inattività del clock. Nel modo SPI se CKP = 1 questo stato di inattività è alto e se CKP = 0 è basso. 																															
SSPEN																															
<ul style="list-style-type: none"> Bit di attivazione del modulo MSSP. Se SSPEN = 1 si attiva la porta seriale in modo SPI e i piedini SCK, SDO, SDI e SS# svolgono il lavoro corrispondente. Se SSPEN = 0 non funziona in modo SPI e i piedini precedenti sono utilizzati come I/O digitali. 																															
SSPOV																															
<ul style="list-style-type: none"> Impostato a 1 indica "l'overflow" il che significa che l'informazione ricevuta su SSPSR e che è stata passata su SSPBUF, quando arriva un altro byte su SSPSR, non è ancora stata letta e verrà mascherata l'informazione precedentemente esistente su SSPBUF. 																															
WCOL																															
<ul style="list-style-type: none"> Quando questo bit passa a 1 indica che si è verificata una "collisione". Il dato da trasmettere, che è stato scritto su SSPBUF, è spostato su SSPSR per spedirlo via seriale. Quando il registro SSPBUF carica su SSPSR un byte senza aver dato il tempo di uscire a quello precedente si produce una collisione. 																															

Distribuzione e utilizzo dei bit del registro SSPCON nella comunicazione in modo SPI.



PIC, il modo di mandare i dati via seriale è molto semplice.

È sufficiente impostare sul registro SSPBUF il dato da trasmettere e il PIC si incaricherà di generare gli impulsi di clock e spedire i bit uno ad uno. Su SSPBUF troveremo anche l'informazione ricevuta, trasformata nuovamente da seriale in parallelo. Il bit BF indicherà quando è arrivato un nuovo dato. SDO è l'uscita dei dati, SDI l'ingresso degli stessi, e SCK il clock di sincronizzazione.

Inoltre, se la comunicazione si esegue fra PIC uno di questi sarà il master e gli altri gli slave, e questi ultimi dovranno utilizzare un quarto piedino (SS#) collegato a massa. SCK deve essere sempre generato da un master e le linee di ingresso/uscita devono essere incrociate con le periferiche.

Registri utilizzati

Come qualsiasi dispositivo di un PIC, prima di poterlo utilizzare è necessario configurarlo, cosa che si realizza tramite i registri SSPCON e i tre bit del registro SSPSTAT. Nelle figure (a lato e alla pagina precedente) sono riportate le funzioni di questi bit, in un prossimo capitolo, dedicato a questo tipo di comunicazione vedremo diversi esempi di collegamenti in questo modo.

La comunicazione seriale fra PIC permette la distribuzione del lavoro in una applicazione.

