

# Il modulo MSSP

Nei PIC **MSSP** (*Master Synchronous Serial Port* - port seriale sincrono con funzione di master) è il modulo periferico dedicato alle comunicazioni seriali di tipo sincrono, ovvero costituite da una linea dati e da una linea di clock. Quelle codificate sono:

- **SPI** (*Serial Peripheral Interface*), in modo master e slave (a 7 o 10 bit)
- **I<sup>2</sup>C** (*Inter Integrated Circuit*), in modo master, slave e multi-master
- **SMBus**, una variazione di I<sup>2</sup>C

Si tratta di sistemi molto diffusi in quanto esiste una gamma assai ampia di periferiche che possono colloquiare con il microcontroller attraverso pochi collegamenti (tipicamente da 2 a 4); ne fanno parte EEPROM, driver per display, AD e DA converter, RTC, ecc.

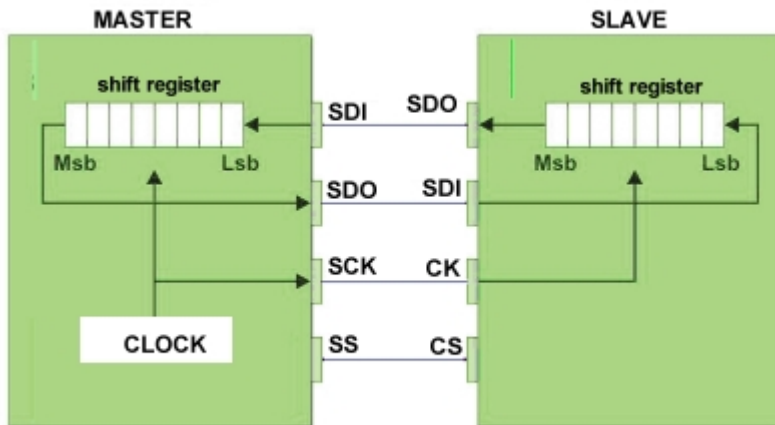
Il modulo MSSP dispone di alcuni registri di controllo con cui l'utente programma le varie funzioni e gestisce la comunicazione. I moduli MSSP delle varie famiglie di PIC MSSP sono analoghi a quello qui descritto.

Da notare che alcuni PIC dispongono di modulo **SSP** (*Synchronous Serial Port*) che si differenzia da MSSP per quanto riguarda la I<sup>2</sup>C, mentre le funzioni in SPI sono le stesse: qui il modulo SSP o MSSP consente sia il modo Master che quello Slave.

## SPI

è un sistema di trasmissione dati tra un dispositivo master e uno o più dispositivi slave.

L'interfaccia fisica è composta essenzialmente da una linea seriale su cui sono inviati blocchi dati sincronizzati da un segnale di clock.



La linea definita come SDO (Serial Data Out) dal master trasmette i bit contenuti in uno shift register, a partire dall' MSB.

Sullo slave, un ingresso SDI (Serial Data Input) convoglia i dati in arrivo in un analogo shift register che li converte da seriale a parallelo.

Il clock per l'avanzamento degli shift register, sincronizzato con i dati, è fornito dal master su una linea SCK.

Una linea **SDO** chiude l'anello di circolazione dei dati collegando l'uscita dello shift register dello slave con l'ingresso di quello del master. Una ulteriore linea **SS** può essere attivata per selezionare il PIC se configurato come Slave (chip select a livello basso).

Il modulo MSSP consente la trasmissione e la ricezione contemporanee di blocchi dati a 8 bit (full duplex). Per questo il modulo ha associati vari registri. In particolare i registri dati sono composti da una coppia di registri : **SSPREG**, lo shift register vero e proprio e **SSPBUF**, un buffer intermedio di salvataggio dei dati.

Ci sono poi tre registri (**SSPSTAT**, **SSPCON1**, **SSPCON2**) per il controllo delle funzioni del modulo. Attraverso questi registri si potrà definire la configurazione da attribuire alle funzioni del modulo stesso.



Il modulo può operare come interfaccia SPI o I<sup>2</sup>C e i pin dedicati alla comunicazione sono gli stessi; **poichè i due bus non sono compatibili, un singolo modulo SSP potrà operare solo o come SPI o come I2C.**

Dovendo utilizzare contemporaneamente entrambe le interfacce, occorrerà scegliere un PIC dotato di due o più moduli MSSP, oppure utilizzare il modulo per I<sup>2</sup>C ed emulare SPI da software.

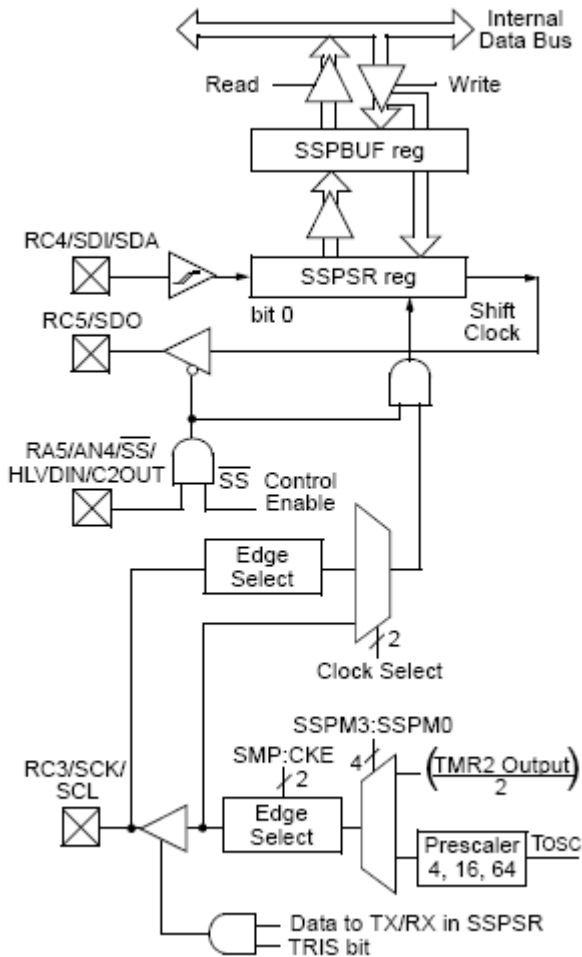
Va da se che le possibilità funzionali del modulo sono maggiori di quello che sarebbe ragionevolmente possibile con una emulazione software.

I registri base sono tutti a 8 bit, quindi il modulo SPI potrà comunicare in full duplex su dati di questa dimensione. E' possibile il funzionamento in tutti e quattro i modi di clock.

La corrispondenza tipica tra linee SPI e pin è la seguente:

Funzione	Pin	Note
<b>SDO</b>	<b>RC5</b>	Serial Data Out
<b>SDI</b>	<b>RC4</b>	Serial Data In
<b>SCK</b>	<b>RC3</b>	Serial Clock
<b>/SS</b>	<b>RA5</b>	Slave Select (negato) - opzionale

Il diagramma logico del modulo MSSP per la funzione SPI è il seguente:



Le operazioni di scrittura e lettura non sono effettuate sullo shift register SSPREG, ma sul buffer intermedio SSPBUF, collegandolo con il bus dati interno.

I dati in arrivo dalla linea SDI sono filtrati da un trigger di Scmitt e inviati al registro di scorrimento, la cui uscita è collegata con la linea SDO.

Nel PIC, la linea SS è funzionale solamente quando l' MSSP è configurato come slave. Essa controlla il gate della linea SDO, inibendo se a livello alto, l' uscita, oltre al clock allo shift register .

Il clock dipende dal Timer 2 o dal TOSC con un prescaler. Alcuni bit dei registri di controllo permettono di selezionare la sorgente (SSPM3:0) e i fronti attivi (SMP:CKE).

Il clock è inviato al pin SCK attraverso un gate che è abilitato solamente se un dato è presente per la trasmissione in SSPSR o ne deve essere ricevuto uno.

Da osservare che il pin SCK dipende anche dal TRIS relativo (TRISC,3), ovvero sarà necessario programmare RC3 come uscita per permettere al clock di raggiungere il pin.

Ovviamente i pin, se programmati per SPI, non possono essere usati per altre funzioni.



E' opportuno ribadire che il pin SS è utilizzato esclusivamente nel modo Slave. In questo modo sarà ancora possibile abilitarne la funzione oppure no; in questo secondo caso il pin corrispondente sarà libero per essere usato in altre applicazioni.

Se in Slave l' opzione SS è abilitata, è indispensabile configurare il TRIS del pin come ingresso, dato che questo non viene effettuato automaticamente dall' abilitazione del modulo MSSP.

In modo Master SS non alcuna funzione il pin corrispondente è libero.

E' ragionevole questa gestione di SS in quanto lascia l' utente libero di utilizzarne l' automatismo, ma anche di scrivere una gestione software diversa da quella a chip select, ad esempio emulando indirizzi nel microcontroller.

Si deve anche considerare che lo standard SPI Motorola prevede periferiche con l' uscita che va in 3-state quando la periferica è disabilitata. Questo consente di collegare gli Slave su linee comuni SDI e SDO. E' necessario, in questo caso, che una sola periferica possa comandare la linea di uscita, altrimenti si verificherebbe un conflitto sul bus. Questo è ottenibile facilmente, e in modo trasparente per l' utente, attraverso l' uso di SS.

Alternativamente sarà possibile disabilitare SDO configurando il pin come ingresso: questo disabilita esclusivamente questa uscita, che potrà essere ripristinata ri configurando il TRIS di SDO come output.

## I Registri SPI

MSSP ha quattro registri per controllare il modo SPI.

- **SSPSR** - è lo shift register vero e proprio e non è direttamente accessibile
- **SSPBUF** - è il buffer dei dati in arrivo e partenza ed è leggibile e scrivibile
- **SSPSTAT** - è il registro di stato della comunicazione
- **SSPCON1** - è il registro di controllo del modulo SPI

In ricezione, la combinazione **SSPSR** e **SSPBUF** crea un sistema a doppio buffer. Quando il dato è completato in **SSPSR** esso viene copiato in **SSPBUF**, così da poterne ricevere uno successivo, mentre l' unità centrale elabora quello corrente. Un flag **SSPIF** viene settato per comunicare l' avvenuta ricezione.

Durante la trasmissione invece si tratta di un buffer semplice: scrivendo **SSPBUF** si scriveranno sia questo che **SSPSR**. Sarà quindi necessario attendere la completa trasmissione del byte caricato nel registro di scorrimento prima di caricare il successivo.

## SSPSTAT

**SSPTAT** (*SSPSTATUS*) è il registro di status della comunicazione, ma contiene anche elementi relativi al funzionamento come I<sup>2</sup>C.

bit	SMP	CKE	D/A	P	S	R/W	UA	BF
<b>SSPSTAT</b> funzione	R/W	R/W	R	R	R	R	R	R
default	0	0	0	0	0	0	0	0

- bit 7 **SMP** : (*SaMPle Timing*) bit di selezione del campionamento  
in modo Master SPI  
 1 = il dato in ingresso è campionato al termine del tempo dati  
 0 = il dato in ingresso è campionato alla metà del tempo dati  
in modo Slave SPI  
 deve essere posto a 0 se l' MSSP lavora in questo modo
- bit 6 **CKE** : (*Clock Edge select*) selezione del fronte del clock per la trasmissione  
 1 = transizione tra il modo attivo e quello idle del clock  
 0 = transizione tra il modo idle e quello attivo del clock  
 La polarità del clock in idle è selezionata da SSPCON1<4>
- bit 5 **D/A** : dato/indirizzo  
 bit usato nel modo I<sup>2</sup>C
- bit 4 **P** : stop bit  
 bit usato nel modo I<sup>2</sup>C  
 Questo bit è a 0 se il modulo MSSP è disabilitato
- bit 3 **S** : start bit  
 bit usato nel modo I<sup>2</sup>C
- bit 2 **R/W** : read/write  
 bit usato nel modo I<sup>2</sup>C
- bit 1 **D/A** : dato/indirizzo  
 bit usato nel modo I<sup>2</sup>C
- bit 0 **BF** : Buffer Full  
 Solo per la ricezione  
 1 = SSPBUF è pieno  
 0 = SSPBUF vuoto o non completo

**SMP** controlla la temporizzazione del campionamento dei dati in arrivo quando il modulo funziona come Master; se il modulo è configurato come Slave SMP va messo a livello basso.

Attraverso questo bit è possibile adattare il Master per essere compatibile con qualunque Slave.

Il tempo di campionamento del dato in ingresso viene sincronizzato a metà o alla fine del tempo di emissione del dato in uscita.

**CKE** controlla quando il dato è trasmesso in relazione ai fronti di commutazione del clock. CKP e CKE insieme controllano il **modo SPI che** si sta usando. Dovendo colloquiare con periferiche aventi modo diverso, i due bit saranno manipolati da programma per ottenere il corretto modo per l' accesso a ogni Slave. I 4 modi possibili sono descritti più avanti.

**BF** indica che il buffer di ricezione è completo e il dato può essere letto (questo sia in condizione Master che in Slave). Il flag si cancella automaticamente leggendo SSPBUF

## SSPCON1

È il registro di controllo della comunicazione.

bit	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
<b>SSPCON1</b> funzione	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	0	0	0	0	0	0	0	0

- bit 7 **WCOL** : Write Collision Detect  
Valido solo per la trasmissione  
 1 = SSPBUF è stato sovrascritto mentre stava completando una trasmissione.  
 Il flag va cancellato da software  
 0 = nessuna collisione
- bit 6 **SSPOV** : (*Synchronous Serial Port Overflow*) Overflow in ricezione  
in modo SPI Slave  
 1 = è stato ricevuto un byte mentre non era ancora stato scaricato il precedente  
 0 = no overflow  
 \*\* vedere nota successiva
- bit 5 **SSPEN** : *Synchronous Serial Port Enable*  
 1 = abilita la funzione e configura i pin relativi

0 = disabilita la funzione e rilascia i pin per altri usi

\*\*\* vedere nota successiva

- bit 4 **CKP** : Clock polarity (nello stato idle)
  - 1 = idle a livello alto
  - 0 = idle a livello basso
- bit 3:0 **SSPM3:0** : Mode select
  - 0101 = SPI Slave, clock = SCK, SS disabilitata e utilizzabile come GPIO
  - 0100 = SPI Slave, clock = SCK, SS abilitato
  - 0011 = SPI Master, clock = TMR2/2
  - 0010 = SPI Master, clock = Fosc/64
  - 0001 = SPI Master, clock = Fosc/16
  - 0000 = SPI Master, clock = Fosc/4

#### Note:

**SSPOV**: l' overflow di ricezione può accadere solamente in modo Slave se l' unità centrale non riesce ad elaborare i dati ricevuti con sufficiente velocità. L' utente deve leggere SSPBUF anche se sta solo trasmettendo dati per evitare che il flag di overflow sia settato. Questo **flag va cancellato da programma**. In modo master l' overflow non può essere settato fino a che una nuova ricezione (e trasmissione) è inizializzata scrivendo il registro SSPBUF.

**SSPEN**: l' abilitazione del modulo MSSP con **SSPEN** dirige il pin relativi a questo modulo, ma NON ne stabilisce la **direzione**.



La direzione corretta dei pin usati per la comunicazione SPI, compreso SS, **dovrà essere gestita dall' utente con il registro TRIS relativo**.

Se i pin usati da SPI non sono configurati correttamente, la comunicazione non potrà funzionare, a anche abilitando il modulo MSSP.

**CKP**, assieme a **CKE**, determina il modo SPI corrente.

**SSPM3:0** permettono di settare la modalità Master o Slave.

Nel modo Slave SS può venire attribuito al pin RC5 oppure questo può essere rilasciato come I/O per altri usi.

Nel modo Master è programmabile la sorgente per il clock della trasmissione (nel modo Slave il clock sarà fornito dalla linea SCK proveniente dal Master esterno). In questo caso occorrerà utilizzare generici pin di I/O per la selezione delle periferiche collegate.

## Programmazione del modulo MSSP

Nell' inizializzazione del modulo MSSP occorre tenere presenti vari parametri.

In primo luogo, la modalità operativa, ovvero se Master o Slave. Questo si effettua con i bit SSPM3:0 di SSPCON1.

La scelta comprende anche la sorgente del clock (in modo Master) e l' uso del pin SS in modo Slave.

In modo Slave, se SS viene utilizzato, il pin corrispondente dovrà essere definito come ingresso con il relativo TRIS. Se non viene usato, è libero per altre applicazioni.

Sia in Slave che in Master occorre ricordare che i pin dedicati al bus SPI vanno settati correttamente nei relativi TRIS, per cui, in modo Master SDO è una uscita, come SCK, mentre SDI è un ingresso.

**Questo è obbligatorio.**

Va da se che in modo Slave SCK sarà invece l' input del clock fornito dal Master esterno.

In particolare, per il modulo MSSP dei PIC enhanced:

- **SDI deve avere TRISC, 4= 1 come ingresso**
- **SDO deve avere TRISC, 5= 0 come uscita**
- **SCK (in modo Master) deve avere TRISC, 3 = 0 (uscita)**  
**SCK (modalità Slave) deve avere TRISC, 3 = 1 (ingresso)**
- **SS (solo in modo Slave), se usato, deve avere TRISA,5 = 1 (ingresso)**

Al POR, SDI è per default una uscita, ma è opportuno prevedere comunque per sicurezza, nel setup degli I/O per il modulo MSSP, la sua configurazione come uscita, setup che è possibile non avvenga subito a l POR, ma in fasi successive dove c'è stata una qualche manipolazione degli I/O.

Qualsiasi funzione della porta seriale che non è desiderato può essere sovrascritto da programma nel corrispondente registro dati direzione (TRIS) sul valore opposto.

Ovvero:

- non utilizzando la linea SDI, RC4 può essere programmato come uscita ed utilizzato come GPIO generico
- avendo la necessità di escludere l' uscita SDO, mandandola in una situazione che elimina conflitto su un bus multi periferiche, si potrà programmare RC5 come ingresso

- programmando RC3 come ingresso in modo Master non si otterrà alcuna uscita del clock. Ovviamente RC3 non dovrà essere programmato come uscita nel caso in cui il PIC sia uno Slave, onde evitare conflitti sul bus del clock con il Master e possibili guasti ai componenti.

In modo Master il clock della trasmissione può essere una frazione del clock principale (/64, /16, /4), il che permette un clock massimo di 10 MHz con clock principale a 40 MHz. Ovviamente il clock dovrà essere quello che la periferica e la connessione relativa possono supportare.

Nel caso in cui nessuna delle tre opzioni dipendenti dal clock principale è utilizzabile, sarà possibile impiegare la metà del clock fornito da **Timer2**. L'uso di **Timer2** in questo contesto è relativo esclusivamente alla frequenza in uscita, che sarà necessario determinare con l'opportuna programmazione dei registri del Timer stesso. La cosa non influisce sull'uso del timer in altre parti del programma, ma, ovviamente, un cambio della frequenza generata si rifletterà sulle operazioni di MSSP.

La seconda importante azione è quella di determinare il modo di funzionamento della comunicazione SPI con i bit CKE e CKP.

La definizione di uno dei 4 possibili modi dipende dalla periferica con cui si vuole colloquiare. In un sistema multi-Slave con abilitazione degli Slave attraverso chip select, potranno essere usate periferiche con modi diversi, in quanto i due bit si possono scrivere in ogni momento prima della ricezione/trasmissione.

Per abilitare la porta seriale si utilizza il bit SSPEN (SSPCON<1:5>).

Per reimpostare o riconfigurare una diversa modalità SPI occorre disabilitare SSPEN, reinizializzare SSPCON come richiesto e quindi riabilitare SSPEN.

Vediamo un breve esempio di setup del modulo MSSP come Master:

```
#define CS PORTA,5 ;utilizza RA5 come I/O di chip select della periferica
.....

; configurazione MSSP
bsf LATA, 5 ; preselect CS a livello alto ; CS I/O come uscita
bcf TRISA,5 ; pin CS come uscita
bcf TRISC,3 ; SCK come uscita (Master)
bcf TRISC,5 ; e SDO come uscita (Master)

movlw 0x40 ; Modo 1,1, sincro a metà
movwf SSPSTAT
movlw 0x31 ; Modo 1,1 SPI Master, 1/16 Tosc, SPI abilitato
movwf SSPCON1
```

Abbiamo visto come MSSP sia costituito da un registro di scorrimento per la trasmissione/ricezione (SSPSR) e da un registro di buffer (SSPBUF).

Attraverso SSPSR i dati sono movimentati dentro e fuori il PIC, **emettendo per primo l' MSb** (e, in ricezione, ricevendolo per primo).

In ricezione, una volta ricevuti gli 8 bit del dato, il byte viene spostato al registro SSPBUF e la situazione è comunicata settando il flag BF (SSPSTAT <0>) e l'interrupt flag bit SSPIF.

Il doppio buffering dei dati ricevuti permette al prossimo byte di arrivare in SSPSR mentre l'unità centrale sta leggendo il precedente contenuto in SSPBUF.

Qualsiasi scrittura del registro SSPBUF durante la trasmissione/ricezione dei dati verrà ignorata e la collisione di scrittura sarà rilevata dal flag WCOL (SSPCON1 7), che sarà necessario cancellare da software in modo da poter individuare questa situazione anomala. Quando l'applicazione si aspetta di ricevere dati validi, è consigliabile leggere il SSPBUF prima che il prossimo byte di dati da trasferire sia passato in SSPBUF: il bit di buffer pieno, BF (SSPSTAT <0>), indica quando SSPBUF è stato caricato con i dati ricevuti, così da evitarne la sovrascrittura. Quando viene letto SSPBUF, il bit BF è riportato a 0 automaticamente.

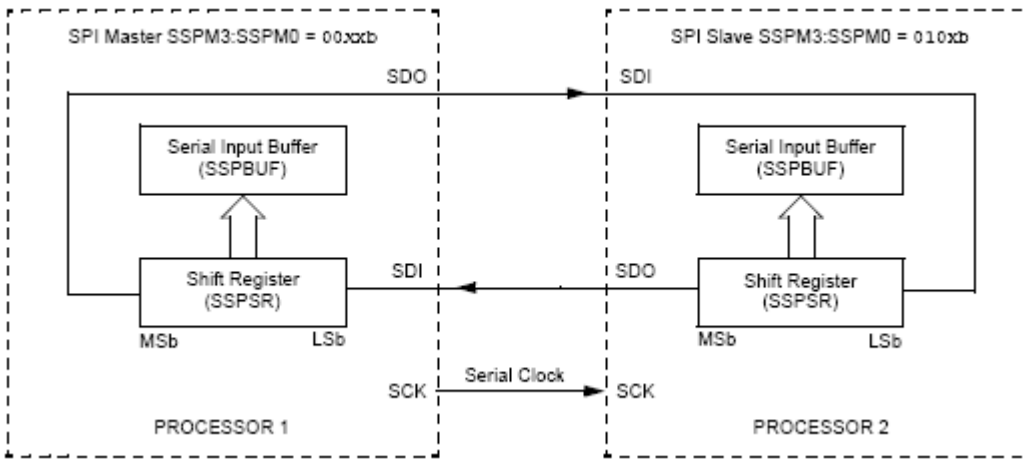
I dati ricevuti possono essere irrilevanti se il PIC è solo un trasmettitore e, in tal caso, saranno scartati.

In generale, l'interrupt di MSSP viene utilizzata per determinare quando è stata completata la trasmissione/ricezione e l'SSPBUF deve essere letto e/o scritto. Comunque è possibile utilizzare un polling software per evitare una collisione di scrittura o lo stato di buffer pieno. Il registro SSPSTAT indicherà le varie condizioni di stato.

Nell'esempio viene illustrato il caricamento di SSPBUF (SSPSR) per la trasmissione dati. SSPSR non è direttamente accessibile e viene raggiunto solo attraverso SSPBUF.

```
LOOP BTFSB SSPSTAT, BF ;dato ricevuto o trasmissione completata?
BRA LOOP ;no - loop
MOVF SSPBUF, W ;copia SSPBUF in WREG
MOVWF RXDATA ;salva dato in RAM
MOVF TXDATA, W ;muove dato da trasmettere in W
MOVWF SSPBUF ;trasmette dato
```

## Collegamento tipico



La figura mostra una tipica connessione tra due unità SPI, in questo caso due microcontroller, uno operante come Master e uno come Slave.

Il controller master (processore 1) avvia il trasferimento dei dati inviando il segnale SCK.

I dati sono spostati dai registri a scorrimento sul fronte del clock programmato (entrambi i processori devono essere programmati per lo stesso modo di funzionamento con CKP).

Il valore attribuito ai dati ( significativi o dummy) dipenderà dal software applicativo. Questo porta a tre scenari per la trasmissione dati:

1. Invio dati dal Master allo Slave: il Master invia dati significativi - lo Slave invia dati fittizi (scarico del suo shift register)
2. Full duplex: il Master invia dati significativi - lo Slave invia dati significativi
3. Lettura dati da parte del Master: il Master invia dati fittizi (generando il clock, scarica il suo shift register- lo Slave invia i dati significativi)

Il Master può avviare il trasferimento dei dati in qualsiasi momento, perché esso controlla la SCK. Nel modo Master i dati sono trasmessi non appena viene scritto il registro SSPBUF.

Se l'operazione di SPI è solo ricevere, l'output SDO potrebbe essere disattivato, programmando il pin come input.

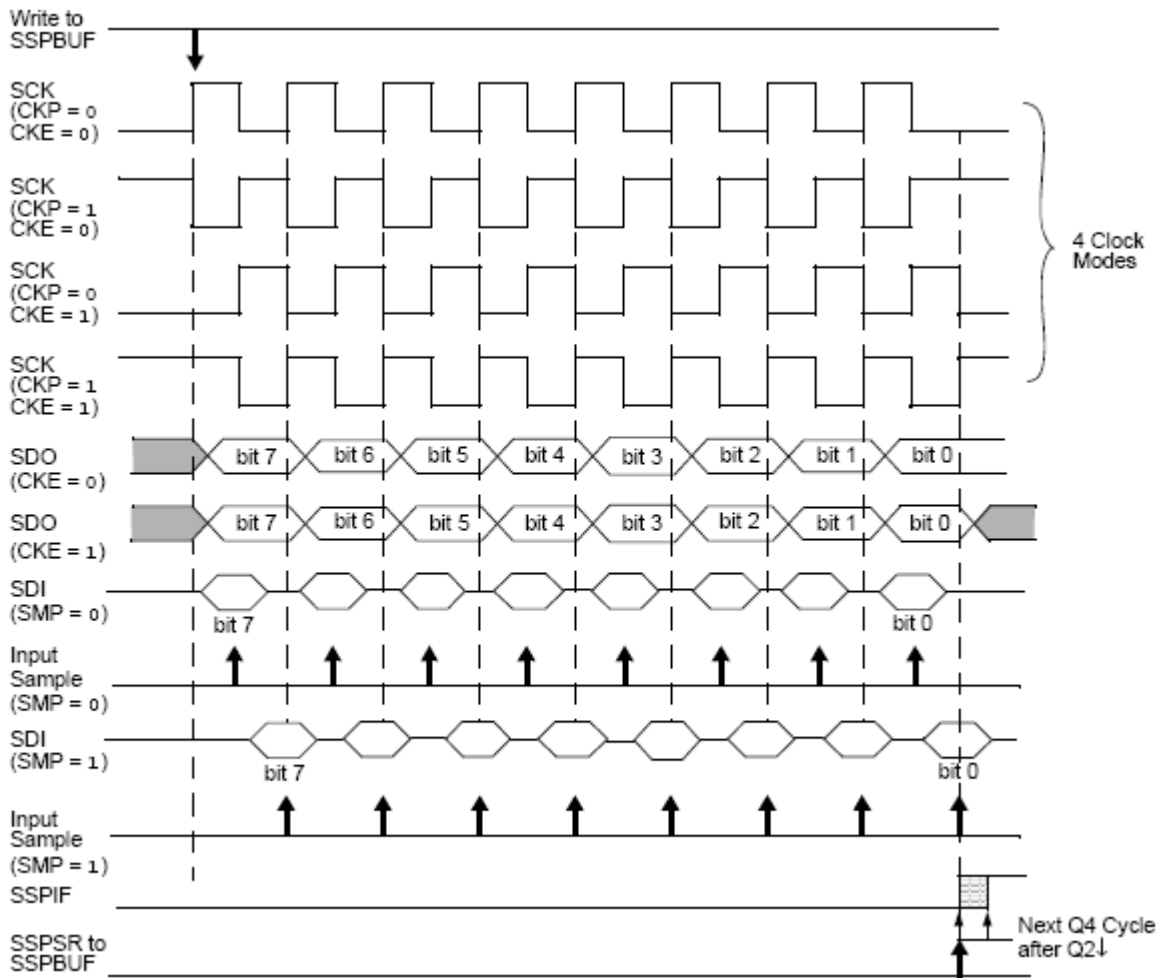
Il registro SSPSR continuerà a spostare i bit presenti sul pin SDI secondo il clock. Appena un byte è ricevuto viene caricato nel registro SSPBUF e vengono impostati il flag di interrupt e il bit di stato. La condizione potrebbe essere utile anche nelle applicazioni di ricevitore come "Line Activity Monitor".

La polarità del clock è selezionata programmando opportunamente il bit CKP (SSPCON<1:4>). Le possibili configurazioni danno origine a diverse forme d'onda come visibile nei diagrammi seguenti.

Nel modo Master, la frequenza di clock (bit rate) è programmata dall'utente solo sul Master, dato che gli Slave dipendono da questo solo clock. Siccome è l'utente a dover scegliere quale polarità e fronte del clock va impiegata, occorre prestare attenzione alle esigenze della periferica collegata.

Una prima variabile è il livello a cui sta la linea di clock quando non ci sono trasmissioni. Questa condizione di riposo si definisce *idle*. Si può quindi avere uno stato *idle* alto o basso programmando il bit **CKP** (*Clock Polarity*) in **SSPCON1**.

A seconda delle richieste della periferica, il sincronismo tra il dato e il clock può essere effettuato sul fronte di salita o su quello di discesa del clock stesso.. Questo si ottiene con la programmazione del bit **CKE** (*Clock Edge*). Le 4 possibili variazioni sono schematizzate nel diagramma seguente.



Quanto sopra è valido sia nell' impiego come Master, sia in quello come Slave. Va aggiunto che in modo Slave, i dati sono trasmessi e ricevuti sugli impulsi di clock esterno al pin SCK e quindi la programmazione di CKE e CKP dovrà essere adeguata. Inoltre il PIC in modo Slave, se è selezionata l' opzione che attiva SS, potrà comunicare solamente se questo pin è a livello basso.

Se si seleziona l' uso di SS (abilitato SSPCON1 <3:0> = 04h) si permette una modalità Slave automatica: quando il pin SS è basso, trasmissione e ricezione sono attivate e il pin SDO è comandato dall' uscita dello shift register. Quando il pin SS va a livello alto, il pin SDO non è più comandato e la trasmissione è sospesa, anche se nel mezzo di un byte trasmesso; il pin passa in stato floating. In queste condizioni resistori di pull-up o pull-down esterni possono essere desiderabili, a seconda dell' applicazione, onde non lasciare floating la linea.

Quando l'ultimo bit è agganciato, viene impostato il flag di interrupt SSPIF.

Prima di attivare il modulo in modalità Slave SPI, la linea di clock deve corrispondere al proprio stato inattivo. Questo può essere osservato leggendo il pin SCK. Lo stato di idle è determinato da CKP (SSPCON1 4).

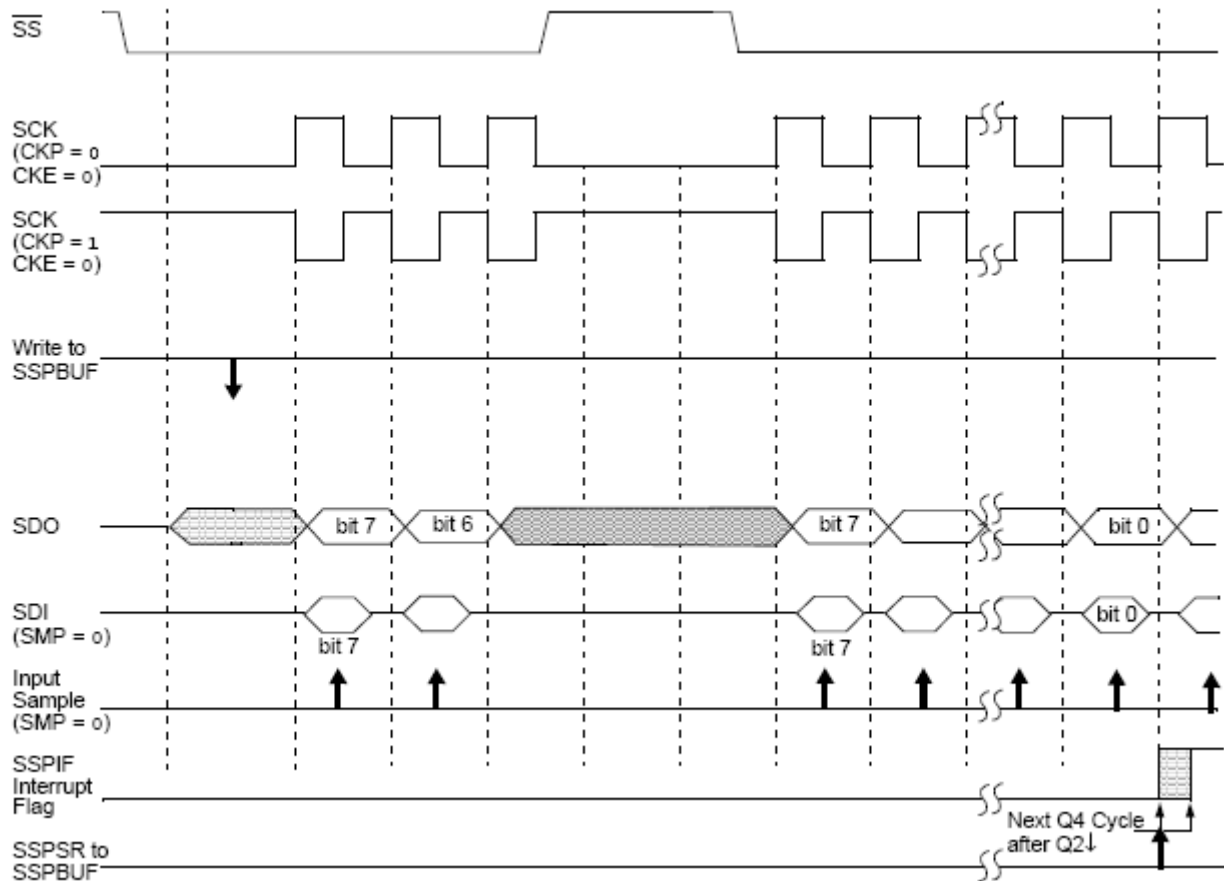
Se lo slave è in stato di Sleep, la ricezione dei dati è una fonte per il wake-up.

Nota :

1. Quando l'interfaccia SPI è in modalità Slave con SS abilitato (SSPCON1 <3:0> = 0100), il modulo SPI resetterà se il pin SS va a livello alto.
2. Se l'interfaccia SPI viene utilizzata in modalità Slave con CKE=1, allora deve essere attivato il controllo del pin di SS.

Quando viene reimpostato il modulo SPI, il contatore di bit è resettato. Questo può essere fatto sia portando il pin SS ad un livello, sia de selezionando il bit SSPEN.

Il diagramma seguente riporta le forme d'onda di sincronismo dello Slave:

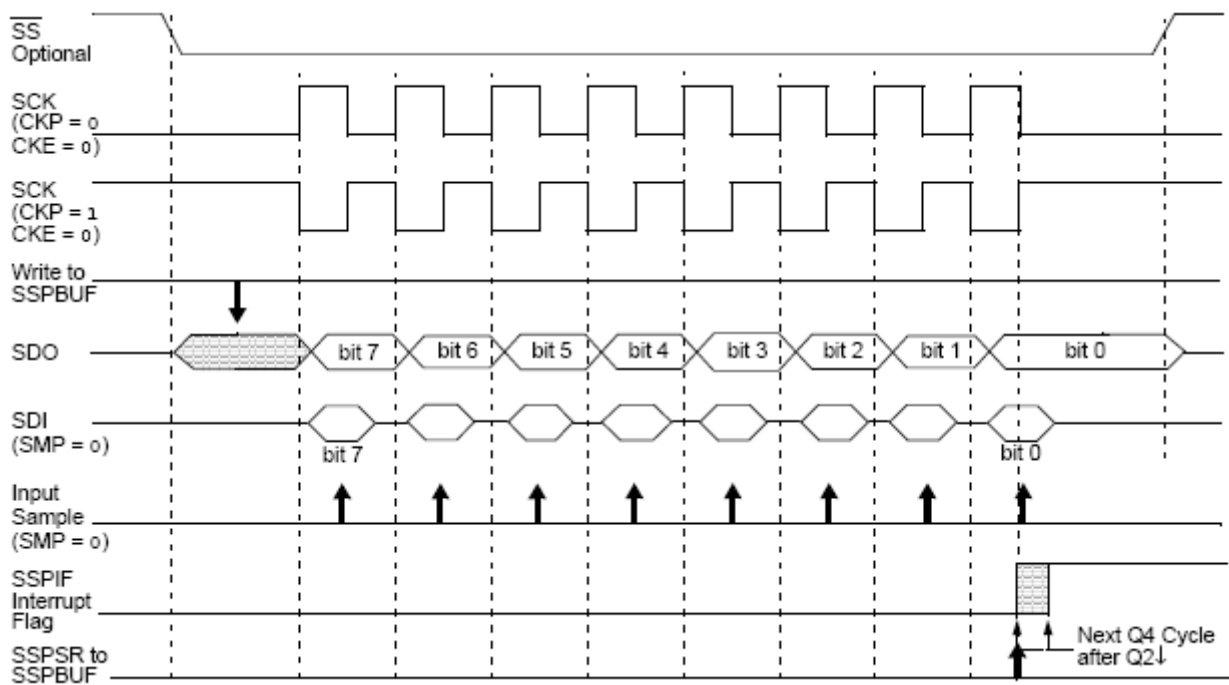


Il modo 0,0 gestisce la linea di clock praticamente all' opposto del modo 1,0.

Osserviamo anche che con SMP=0 il campionamento del dato è fatto alla metà del tempo di durata del bit sulla linea SDI.

Il flag **SSPIF** e il trasferimento del dato da SSPSR a SSPBUF avvengono dopo la ricezione dell' ultimo bit. Il livello basso su SS abilita la ricezione.

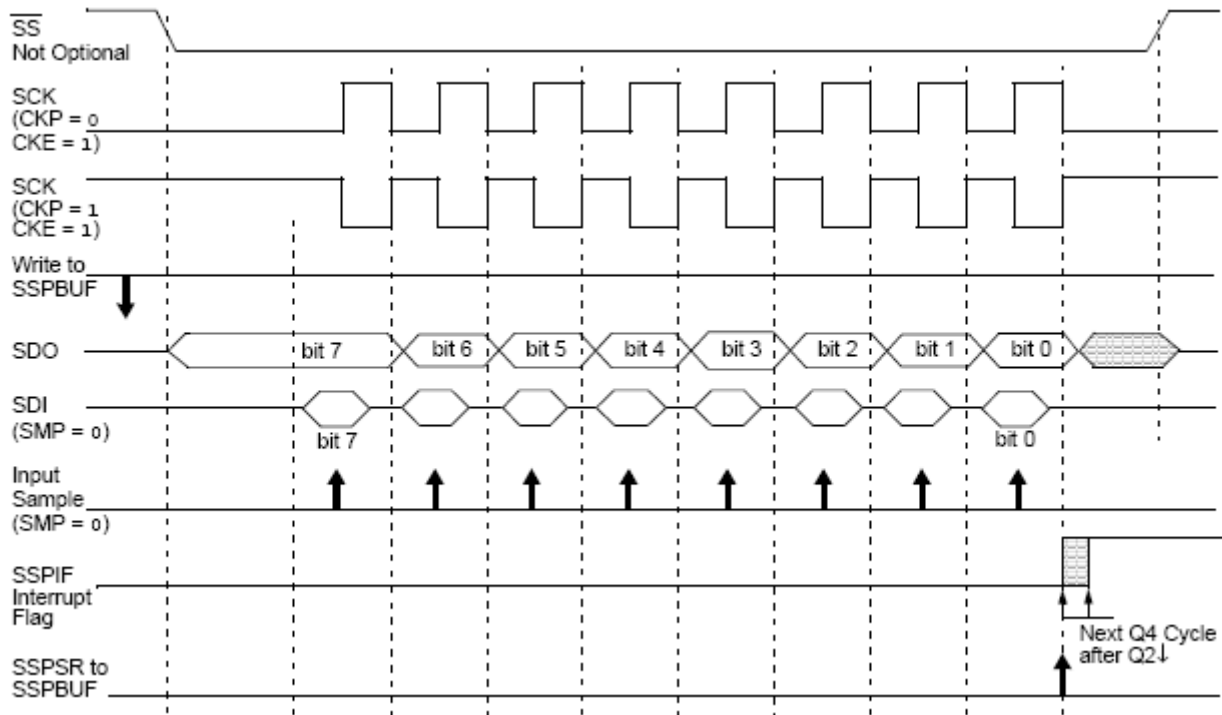
In particolare, se il bit CKE viene impostato a 0



Osserviamo che lo stato di SDO permane sul livello dell' ultimo bit trasmesso fino a che non viene rimossa l' abilitazione di SS.

Per CKE = 1. In questo caso si deve osservare che la funzione SS non è opzionale e va abilitata ed utilizzata.





## SETUP di MSSP

Un breve esempio di setup dell' MSSP come Slave:

```

; configurazione MSSP
bsf LATA, 5 ; preselect CS a livello alto
bcf TRISA,5 ; pin CS come uscita
bsf TRISC,4 ; pin SDI come input
bsf TRISC,3 ; SCK come ingresso (Slave)
bcf TRISC,5 ; e SDO come uscita (Slave)

movlw 0x40 ; Modo 1,1, sincro a metà
movwf SSPSTAT
movlw 0x34 ; Modo 1,1 SPI Slave, SS abilitato, SPI abilitato
movwf SSPCON1

```

## SPI a due fili e altre configurazioni

Alcune periferiche utilizzano una sola linea dati bi direzionale, per cui sono necessari solamente questa linea e il clock, con l' eventuale SS. Si tratta quindi di una comunicazione half duplex, dove di volta in volta una o l' altro dei dispositivi trasmette o riceve su questa linea.

Il modulo MSSP non ha una struttura per trattare questa configurazione in modo diretto, ma si può operare semplicemente **collegando il pin SDO al pin SDI**.

Si deve considerare che, quando una unità trasmette, la linea bi direzionale viene comandata dal suo pin SDO; per questa ragione lo standard Motorola prevede che le linee di trasmissione vadano in 3-state se la periferica non è selezionata. Questo serve ad **evitare un conflitto sul bus** nella sfortunata circostanza in cui due elementi tentassero contemporaneamente di trasmettere.

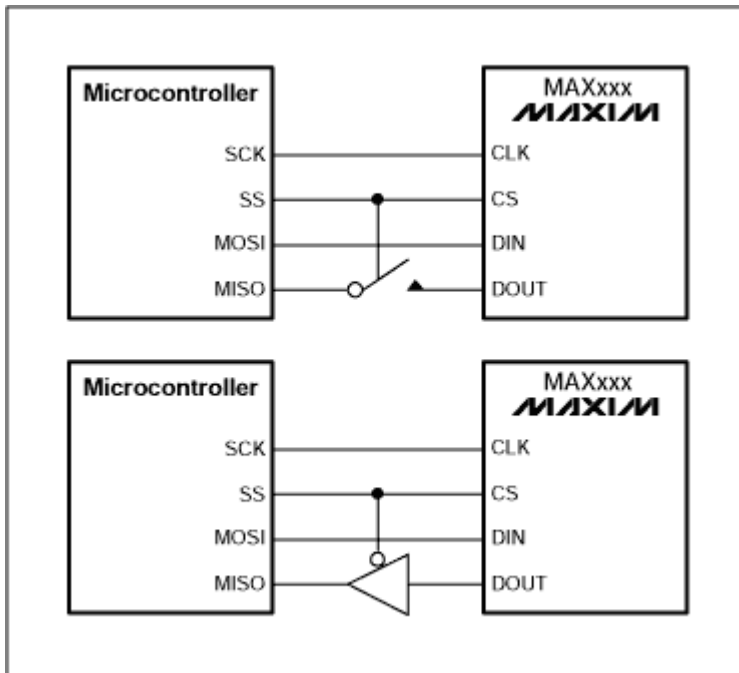
Utilizzando il PIC come ricevitore, il pin **SDO** non deve trasmettere; ci sono due soluzioni:

- Nel caso di impiego del PIC come Slave, l' uso di SS implementa automaticamente la funzione di 3-state di SDO quando il microcontroller Slave non è selezionato (per altro obbligatorio se si è programmato CKE=1).
- però è possibile che per CKE=0 in Slave SS non venga utilizzato. Quindi, in questo caso, e per la modalità Master, la soluzione è quella di configurare il pin SDO come input. Questo disabilita la trasmissione e il pin SDO come input è un carico trascurabile per la linea. Si opererà viceversa quando cambia la direzione della comunicazione.

Essendo sempre un ingresso, SDI può non crea conflitto sul bus.

Analogo problema quando su un bus sono mescolati dispositivi con linee in tree-state a dispositivi periferici senza questa funzione. Il 3-state delle uscite è un punto fondamentale nel collegamento delle periferiche su linee comuni SDI e SDO.

Se nel PIC, con i metodi sopra indicati, è sempre possibile disabilitare SDO, esistono periferiche che non dispongono di uscite 3-state; in questo caso l' [AN 1879 di MAXIM](#) offre una possibile soluzione.



Se l'applicazione richiede che un mix di 3-state e non sullo stesso bus SPI, i dispositivi non 3-state possono essere convertiti aggiungendo un commutatore analogico o un gate tree-state tra il pin di emissione dei dati (SDO o DOUT) e il pin di MISO del Master.

Lo switch è comandato dal CS che seleziona il dispositivo e quindi attiva la DOUT allo stesso tempo. Sono adatti switch analogici come MAX4595 e MAX4502 (package SC-70 e SOT-23). In alternativa, potrebbe essere utilizzato un terzo di uno switch SPDT triplo MAX4053 o 74HC4053, oppure buffer tree-state Fairchild TinyLogic NC7SZ125 e TI SN74AHC1G125 o anche il classico 74HC125 (che contiene 4 buffer in un unico package).

Per contro, in una configurazione daisy chain in cui si utilizzi il PIC come Slave, occorrerà una gestione differente della comunicazione, dato che deve essere assicurata la circolazione dei dati sull'anello. In questo caso non sarà possibile configurare CKE=1 poichè questo, in modo Slave, obbliga all'uso di SS.

## Operatività in power management

Il clock principale, in modalità a bassa potenza, potrebbe essere di valore inferiore a quello previsto per la comunicazione SPI e va considerato che, in caso di modalità di sospensione, tutti i clock sono interrotti.

In modalità Idle, un clock viene fornito per le periferiche: quel clock dovrebbe essere la fonte primaria, il clock secondario (oscillatore a 32,768 kHz) o la fonte INTOSC. Nella maggior parte dei casi, la velocità che il Master usa per il clock non è importante (se non ai fini della velocità di trasmissione) in quanto gli Slave si sincronizzano su questo clock. Tuttavia, l'impatto reale di questa condizione dovrebbe essere valutato caso per caso.

Se sono attivate le funzioni di interrupt per MSSP, esse sono in grado di svegliare il controller dalla modalità di sospensione o da una delle modalità inattive, quando il master ha completato l'invio del dato in corso. Se non si desidera un'uscita dalla modalità di sospensione o Idle, le funzioni di interrupt devono essere disattivate.

Se è selezionata la modalità di sospensione, tutti gli clock del modulo sono interrotti e la trasmissione/ricezione rimarrà sospesa fino al wake up.

In modalità Slave, il registro a scorrimento di trasmissione/ricezione opera in modo asincrono rispetto al microcontroller. Questo permette al dispositivo di essere inserito in qualsiasi modalità di gestione del consumo energetico durante la ricezione di dati. Quando sono stati ricevuti tutti gli 8 bit, l' interrupt del modulo MSSP verrà impostato e se abilitato, provvederà al wake up del microcontroller.

## SPI e interrupt

Un solo flag identifica la condizione di interrupt del moduli MSSP

- **SSPIF** PIR1<3> questo bit va a 1 se la trasmissione o la ricezione sono completate. Va cancellato da programma.

Prima di abilitare l' interrupt di MSSP è opportuno cancellare questo flag per evitare false chiamate.

Per il resto, la gestione dell' interrupt è analoga a quella degli altri moduli; sono presenti:

- **SSPIE** PIE1<3> a livello 1 abilita l' interrupt di MSSP
- **SSPIP** IPIR1<3> a livello 1 attribuisce la priorità alta all' interrupt di MSSP, se IPEN RCON<7> è abilitato

Essendo un **interrupt periferico**, quello di MSSP richiede l' abilitazione sia di **GIE/GIEH** che di **PEIE/GIEL**.

## I registri interessati alla comunicazione SPI

Il foglio dati riporta come al solito un elenco dei registri che sono interessati dalle operazioni del modulo MSSP per le operazioni SPI.

**TABLE 17-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TRISA	TRISA7 <sup>(2)</sup>	TRISA6 <sup>(2)</sup>	PORTA Data Direction Control Register						52
TRISC	PORTC Data Direction Control Register								52
SSPBUF	MSSP Receive Buffer/Transmit Register								50
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	50
SSPSTAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	50

Il fondo bianco delle celle identifica quali registri e bit vanno considerati quando si affronta la programmazione (mentre le celle con fondo grigio indicano bit o registri non interessati).

La gestione dei bit dei registri **PIR1/IPR1/PIE1** sarà considerata dove necessaria, sempre ricordando che i flag dei registri **PIR** sono testabili anche in polling e vanno azzerati da programma quando necessario.

La collocazione dei flag e dei bit nei registri suddetti indica che gli interrupt fanno parte del gruppo dei periferici e la loro chiave di abilitazione, nel caso di interrupt senza priorità, è il bit **PEIE** di **INTCON**.

## Silicon Errata

Il modulo MSSP si trova ad avere possibili problemi dovuti alla realizzazione. Questi problemi sono riportati in documenti chiamati **Errata** da Microchip; essi individuano i problemi e suggeriscono possibili soluzioni.

Anche nel caso della famiglia 18F43/2321 il documento **DS80285b** riporta un problema rilevato proprio sul modulo MSSP.

Nel modo Slave con SS abilitato (SSMP3:0=0100) e CKE=0, il tempo minimo tra il fronte di discesa di SS e il primo fronte di SCK deve essere maggiore di quello specificato nel parametro 70 (table 26-16 del foglio dati), ovvero al minimo 3 Tcy.

Questo tempo minimo è lo stesso necessario tra SS a livello basso e la scrittura di SSPBUF.

In questo caso il modulo MSSP funziona correttamente. Se, invece, il tempo tra la selezione dello Slave con SS e il primo impulso del clock è minore ci potranno essere problemi nell'invio del dato. Problemi che non si verificano se SSPBUF viene caricato prima che lo Slave sia selezionato.

Da osservare che gli *Errata* sono relativi solamente ai dispositivi con ID e revisione indicate dallo stesso documento.

E' pratica indispensabile, scelto un dispositivo, non solo controllarne il foglio dati, ma anche verificare l'esistenza di *Errata*. Si deve dire che i problemi descritti da questi documenti sono assai spesso marginali ed è possibile che l'utente nella sua applicazione non li incroci mai. Ma, in ogni caso, dato che sono tali da creare situazioni in cui il debug diventa drammatico, è opportuno tenerne conto.

## T&T : SPI per i PIC

Questi sono alcuni punti salienti di cui occorre tener conto usando il modulo MSSP in SPI.

- SSP e MSSP per le applicazioni SPI sono analoghi**
- Il modulo MSSP controlla sia la comunicazione SPI che la I<sup>2</sup>C, ma può essere configurato **solamente per una alla volta**. Nel caso occorra avere sullo stesso PIC entrambi i bus si dovrà scegliere un modello dotato di almeno 2 moduli MSSP.
- Nei PIC, **la linea SS è funzionale solamente quando l' MSSP è configurato come slave**. Essa controlla il gate della linea SDO, inibendo se a livello alto, l' uscita, oltre al clock allo shift register .
- Sia in Slave che in Master occorre ricordare che i pin dedicati al bus SPI vanno settati correttamente **nei relativi TRIS**, per cui SDO è una uscita, mentre SDI è un ingresso. SCK è una uscita in modo Master, ma un ingresso in modo Slave
- Per **CKE = 1** in modo Slave **la funzione SS non è opzionale** e va abilitata ed utilizzata.
- Il foglio dati consiglia, prima di attivare il modulo in modalità Slave, di **verificare che la linea di clock deve corrispondere al proprio stato inattivo** (idle). Questo può essere osservato leggendo il pin SCK. Lo stato di idle è determinato da CKP (SSPCON1 4).
- Verificare le caratteristiche delle periferiche collegate per determinare con esattezza il **modo di funzionamento** rispetto al clock e la **polarità** del chip select.
- resettare SSPIF** prima di abilitare l' interrupt

9. Per quanto riguarda i flag di controllo:
    - **BF** viene resettato automaticamente alla lettura di SSPBUF
    - **WCOL** e **SSPOV** vanno cancellati da programma
  10. Dovendo cambiare parametri al modulo:
    - **disabilitarlo con SSPEN=0**
    - effettuare **le modifiche** volute
    - **riabilitare** il modulo
- 

## **SPI + C**

I linguaggi C per i PIC forniscono librerie di funzioni per il setup e l'uso del modulo MSSP