

IL PROTOCOLLO SPI

Il protocollo SPI o meglio l'interfaccia SPI, venne originariamente ideata dalla Motorola (ora Freescale) a supporto dei propri microprocessori e microcontrollori. A differenza dello standard I²C ideato dalla Philips, l'interfaccia SPI non è stata mai standardizzata; ciononostante è divenuta di fatto uno standard. Il non avere delle regole ufficiali ha portato all'aggiunta di molte caratteristiche ed opzioni che vanno opportunamente selezionate ed impostate al fine di permettere una corretta comunicazione tra le varie periferiche interconnesse.

L'interfaccia SPI descrive una comunicazione singolo Master singolo Slave, ed è di tipo sincrono e full-duplex. Il clock viene trasmesso con una linea dedicata (non necessariamente una trasmissione sincrona ha una linea dedicata al clock) ed è possibile sia trasmettere che ricevere dati in contemporanea. In Figura 1 è riportato uno schema base di collegamento tra due periferiche che fanno uso dell'interfaccia SPI.

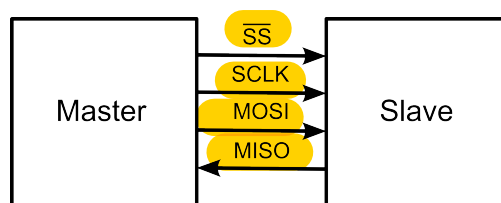


Figura 1: Schema base di collegamento tra periferiche facenti uso dell'interfaccia SPI.

Dalla Figura 1 è possibile subito notare quanto appena detto, ovvero che la comunicazione avviene generalmente tra un Master ed uno Slave. L'interfaccia presenta inoltre 4 linee di collegamento (esclusa la massa comunque necessaria), per cui lo standard SPI è anche noto come 4 *Wire Interface*. È compito del Master avviare la comunicazione e fornire il clock allo Slave. La nomenclatura delle varie linee presenti nell'interfaccia SPI è normalmente la seguente:

- **MISO** : Master Input Slave Output
- **MOSI** : Master Output Slave Input

- **SCLK** : Serial Clock (generato dal Master)
- **SS** : Slave Select (Selezione dello Slave)

Oltre a questa nomenclatura, più o meno standard, sono presenti altre sigle. Per esempio i microcontrollori quali i PIC18 della Microchip o gli MSP430 della Texas Instruments, fanno uso di una nomenclatura leggermente differente ma comunque intuitiva.

Per esempio la linea MOSI viene anche chiamata:

- **SDO** (Serial Data Out), **DO** (Data Out), **DOUT** e **SO** (Serial Out)

La linea MISO viene anche chiamata:

- **SDI** (Serial Data In), **DI** (Data In), **DIN** e **SI** (Serial In)

La linea di Clock viene anche chiamata:

- **CLK**, **SCK** (Serial Clock), **SCK**.

La linea di Enable viene anche chiamata:

- **CS** (Chip Select), **CE** (Chip Enable)

Già da questa prima spiegazione si capisce che non si ha un vero standard. Tornando alla Figura 1 è possibile notare anche la direzione di ogni linea dati. Sebbene sia il Master a dover avviare la trasmissione dati, uno Slave potrebbe richiedere l'avvio di una comunicazione per mezzo di una linea d'interruzione (non prevista dall'interfaccia SPI). Il Master, alla richiesta di una interruzione andrebbe a leggere la periferica.

Si noti che la comunicazione SPI prevede la presenza di un **SS** (*Slave Select*), per cui anche se la comunicazione avviene tra un solo Master e un solo Slave, il Master può selezionare lo Slave con cui effettuare la comunicazione, sia per scrivere che leggere dati. Qualora siano presenti più Slave, il loro collegamento è generalmente effettuato come in Figura 2.

Come visibile si ha ancora un unico Master, che ha questa volta il compito di selezionare lo Slave con il quale avviare la comunicazione; infatti un solo Slave alla volta deve essere attivo. Questo collegamento è possibile solo se la periferica Slave supporta l'opzione di avere la linea MISO di tipo *three states* o *floating* (alta impedenza). In particolare lo stato di alta impedenza deve essere impostato ogni qual volta lo Slave risulti disattivo. Questo è richiesto poiché tutti i MISO sono collegati in parallelo, e non è cosa buona avere uscite in parallelo che abbiano livelli logici differenti (si potrebbero avere dei cortocircuiti che porterebbero alla rottura dei dispositivi stessi).

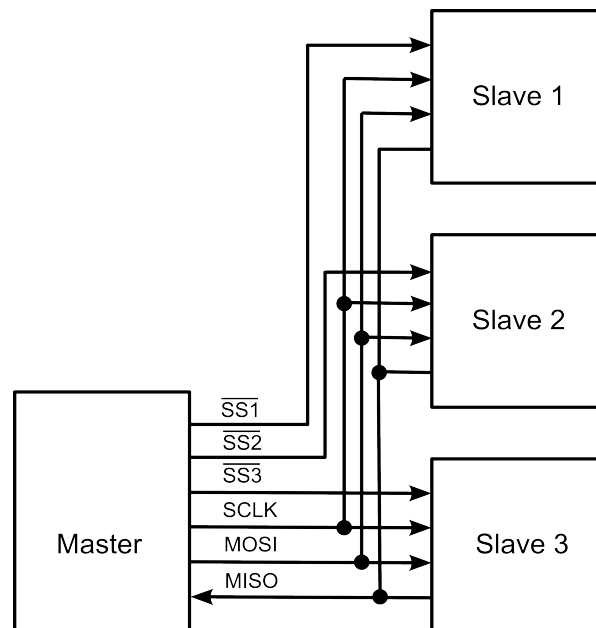


Figura 2: Collegamento di più periferiche facendo uso dell'interfaccia SPI.

Un'altra osservazione da tenere a mente sia sulla Figura 1 che Figura 2 è che la linea SS è attiva bassa. Seppur questo non è uno standard, i microcontrollori della Motorola vennero pensati con questa opzione.

Il segnale *Slave Select* (SS) non è sempre presente nei microcontrollori che possiedono l'interfaccia SPI e per la sua implementazione è necessario distinguere due casi:

Il microcontrollore è impostato per lavorare come Master

Lo *Slave Select* deve essere opportunamente abilitato prima dell'avvio della comunicazione con lo Slave. Il suo corretto utilizzo può cambiare a seconda della modalità di trasmissione utilizzata (si veda di seguito). Frequentemente i microcontrollori in modalità Master fanno uso di semplici I/O generici che vengono assegnati uno per periferica.

Il microcontrollore è impostato per lavorare come Slave

Lo *Slave Select* ha lo scopo di abilitare il microcontrollore al fine di permettergli una corretta comunicazione. Quando il segnale SS non è presente si fa spesso uso di un ingresso con interruzione, in maniera da poter prontamente individuare la richiesta del Master di avviare la comunicazione. Si ricorda che il segnale SS, qualora più Slave dovessero condividere la linea MISO, quando disabilitato, dovrebbe portare l'uscita MISO dello Slave in uno stato di alta impedenza.

L'interfaccia SPI e la sua implementazione

Un modulo di un microcontrollore che implementi l'interfaccia SPI si presenta come un registro a scorrimento (*Shift Register*) sia in una periferica Master che Slave. La connessione tra Master e Slave, mettendo in evidenza i registri a scorrimento, si presenta come in Figura 3.

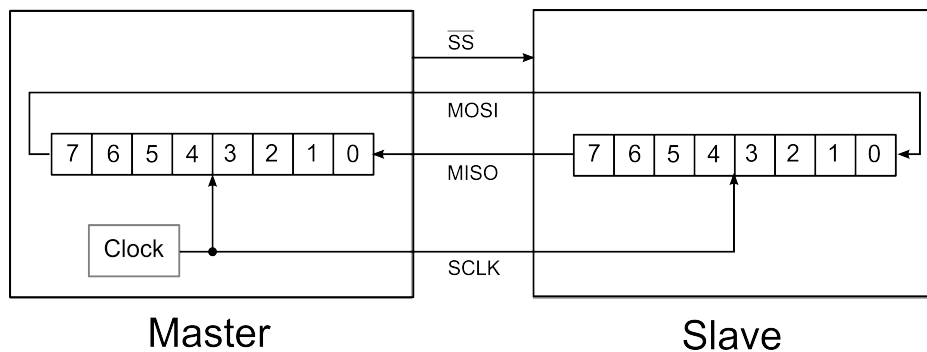


Figura 3: Connessione tra Master e Slave mettendo in evidenza i registri a scorrimento interni.

La rappresentazione per mezzo di un solo registro a scorrimento, seppur semplice, rappresenta in maniera piuttosto realistica l'interfaccia SPI. La dimensione del registro a scorrimento non è specificata, ma spesso la sua dimensione è di un solo byte. In microcontrollori a 16 e 32 bit, per la dimensione stessa dei registri interni al microcontrollore, si può spesso avere l'opzione di selezionare la dimensione del registro, usando le dimensioni comuni 8, 16, 32 (alcune volte si trova anche il formato a 7 bit ma questa è un'opzione raramente usata).

Vediamo ora di analizzare come avviene la trasmissione di un byte. Prima di avviare una comunicazione, il Master attiva la linea SS relativa allo Slave con cui vuole effettuare la comunicazione e successivamente fornisce il clock alla frequenza con cui avverrà la trasmissione. Dopo l'attivazione dello Slave, i bit interni al registro a scorrimento del Master vengono traslati all'esterno (linea MOSI) a partire dal bit più significativo MSB (*Main Significant Bit*). Il bit traslato entra nel registro dello Slave, il quale a sua volta inizia a svuotare il proprio registro inviando il bit più significativo attraverso la linea MISO. La comunicazione termina quando l'ottavo bit viene trasmesso.

Da quanto detto si capisce che se il Master volesse leggere dallo Slave deve comunque inviare un dato fittizio allo Slave. Allo stesso modo, se il Master volesse solo impostare lo Slave o inviare solo Dati, riceverà comunque dei dati fittizi dallo Slave, a meno di ignorare e non collegare la linea MISO.

Rispetto a quanto appena detto ci sono naturalmente delle eccezioni, in particolare la linea SS potrebbe non essere usata qualora nel sistema si abbia un solo Slave. In alcune periferiche, quali per esempio gli ADC (*Analog to Digital Converter*), la linea SS è quasi sempre obbligatoria, visto che oltre a rappresentare la linea *Slave Select*, svolge spesso la mansione di avvio conversione (*Start Conversion*). In situazioni come queste potrebbe essere necessario ritardare il Clock prima di poter effettivamente leggere il risultato della conversione.

L'eventuale mancanza della linea SS, nel caso dei microcontrollori MSP430, viene chiamata *3 Wire communication*, ovvero comunicazione a 3 fili, che si contrappone al fatto che l'interfaccia SPI è nota come *4 Wire communication*. Questa nomenclatura è facile però confonderla con la più nota *3 Wire communication* in cui la linea SS è presente ma le linee MOSI e MISO sono unite in una unica linea per formare una comunicazione *half-duplex*. Questo causa una diminuzione delle informazioni che possono viaggiare nel sistema, vista la natura *half-duplex* del bus. Solo a scopo di completezza si ricorda che esiste anche la comunicazione *1 Wire*, ideata dalla Dallas-Maxim, ma che non ha quasi nulla in comune con l'interfaccia SPI, se non per il fatto di essere un protocollo seriale.

Per complicare un attimo la situazione è bene tenere a mente che molti

microcontrollori hanno l'opzione di trasmettere il byte in uscita non a partire dal bit più significativo ma da quello meno significativo. Come ogni opzione utilizzata si deve avere la premura di controllare che lo Slave sia compatibile con tale opzione. La pratica di trasmettere il bit dal meno significativo a quello più significativo viene normalmente utilizzata nella comunicazione 3 Wire (*half-duplex*).

Come detto la rappresentazione con un solo registro a scorrimento non è una semplificazione a solo scopo didattico, infatti viene spesso utilizzata. Ciononostante molti microcontrollori offrono anche un registro a scorrimento per direzione di trasmissione, come anche un buffer dati al fine da poter caricare il dato ricevuto e permettere una nuova ricezione limitando eventuali perdite d'informazione. Il modulo SPI è spesso supportato da interruzioni in maniera da poter armonizzare l'utilizzo dello stesso con il resto delle attività della CPU.

Si noti che quanto descritto non fa riferimento ad alcun protocollo o formattazione dei dati. Il byte trasmesso è un semplice byte e il suo significato dipenderà dall'applicazione. Questo in contrapposizione con standard come l'I²C e CAN, in cui i vari byte possiedono un determinato significato. La mancanza di un protocollo fa dell'interfaccia SPI una soluzione semplice da implementare anche senza l'ausilio di un modulo dedicato. D'altro lato la mancanza di un protocollo di "supervisione", impedisce all'interfaccia SPI di sapere se il Master è effettivamente collegato ad uno Slave; questo nello Standard I²C è implementato per mezzo del meccanismo di *Acknowledgment*. Per mezzo dell'interfaccia SPI è comunque possibile controllare la presenza di uno Slave, ma è necessario implementarlo, per esempio creando un byte domanda del tipo...ci sei?

Una volta compreso come è realizzato un modulo con interfaccia SPI, è possibile comprendere il terzo modo con cui si possono collegare tra loro più Slave. In Figura 4 è riportato il cosiddetto collegamento in *daisy chain* degli Slave.

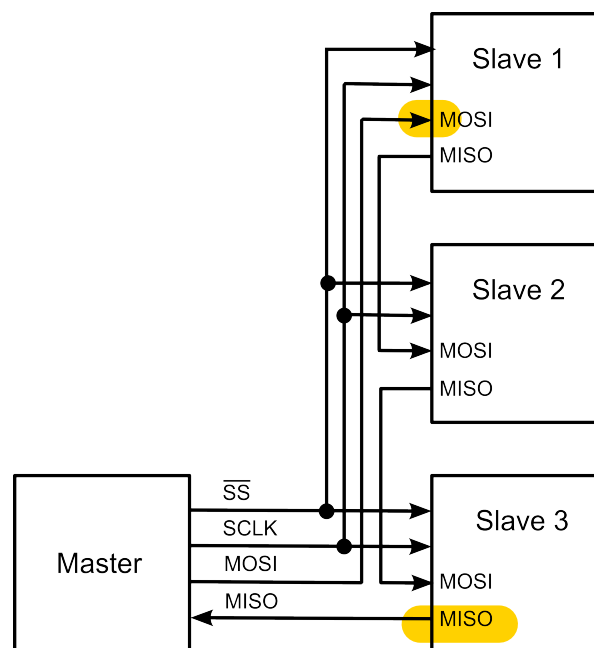


Figura 4: Collegamento in daisy chain di tre Slave.

Dalla Figura 4 si capisce che la catena di Slave è ottenuta collegando l'uscita MISO del primo Slave all'ingresso MOSI del Secondo Slave e così via. L'ultimo Slave avrà il MISO collegato effettivamente al Master. La catena di Slave viene attivata da una sola

linea SS, quindi si ha la possibilità di risparmiare diversi pin altrimenti necessari per ogni singolo Slave. Per comunicare con un dispositivo è necessario questa volta inviare sempre tre byte (uno per periferica) anche se il dispositivo con cui si vuole colloquiare è il primo (in lettura è obbligatorio, mentre in sola scrittura si potrebbe evitare). Questa configurazione, seppur attraente non permette un numero indefinito di periferiche. A seconda della frequenza del segnale di clock e dalla tipologia delle periferiche utilizzate potrebbe essere necessario dover rispettare dei tempi minimi e massimi di lettura o scrittura oltre i quali non è possibile andare.

La tecnica *daisy chain* è la stessa utilizzata dall'interfaccia JTAG (*Joint Test Action Group*) molto simile all'interfaccia SPI, ovvero basata su registri a scorrimento. L'interfaccia JTAG è divenuta uno standard per simulare circuiti integrati montati su sistemi complessi, ma viene sempre più frequentemente utilizzata anche per programmare i dispositivi quali microcontrollori, FPGA e CPLD.

Modalità di trasmissione SPI

Da quanto fin ora letto, ci si sarà resi conto che l'interfaccia SPI non è molto complicata. Proprio per tale ragione riscuote un certo successo. Unica sua difficoltà sta nel fatto che non è presente uno standard, per cui bisogna fare attenzione alle impostazioni del Master e dello Slave. In aggiunta a quanto detto sopra, l'interfaccia SPI può essere impostata per trasmettere o ricevere in quattro modalità differenti. La modalità selezionata deve essere la stessa sia per il Master che per lo Slave. Dal momento che lo Slave spesso non ha modo di cambiare modalità, è il Master a doversi adeguare.

Le quattro modalità appartengono in un certo qual modo allo "standard" SPI, visto che sono descritte nei Datasheet della Motorola (Freescale).

Le quattro modalità vengono normalmente impostate per mezzo di due parametri (spesso implementati con due bit), nominati CPOL (*Clock Polarity*) e CPHA (*Clock Phase*). La polarità consiste semplicemente nell'avere o meno una porta NOT sulla linea di Clock, ovvero quando CPOL = 0 si ha il clock "normale", mentre quando CPOL=1 si ha il clock invertito. Questo comporta che tutte le operazioni che avvengono sul fronte di salita, qualora CPOL sia impostato su 1, avverranno sul fronte di discesa. Allo stesso modo, le operazioni che avvengono normalmente su fronte di discesa, impostando CPOL=1, avverranno sul fronte di salita.

Il parametro CPHA, permette d'impostare la fase del campionamento, ovvero quando i dati devono essere letti (campionati). In Tabella 1 è riportata la corrispondenza tra i valori dei parametri CPOL, CPHA e la relativa modalità.

Modalità	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

Tabella 1 : Corrispondenza tra modalità e parametri CPOL, CPHA.

Si fa presente che alcuni microcontrollori, come per esempio i PIC18 e MSP430 hanno il significato di CPHA invertito. Quando uno Slave richiede quindi la modalità 0 secondo lo standard Freescale CPOL=0 e CPHA=0, per alcuni microcontrollori potrebbe essere

CPOL=0 e CPHA=1. Frequentemente il nome dei bit non è lo stesso per cui bisogna prestare ulteriore attenzione. Vediamo in maggior dettaglio le varie modalità. In Figura 141 è riportato il diagramma temporale nel caso in cui CPHA=1.

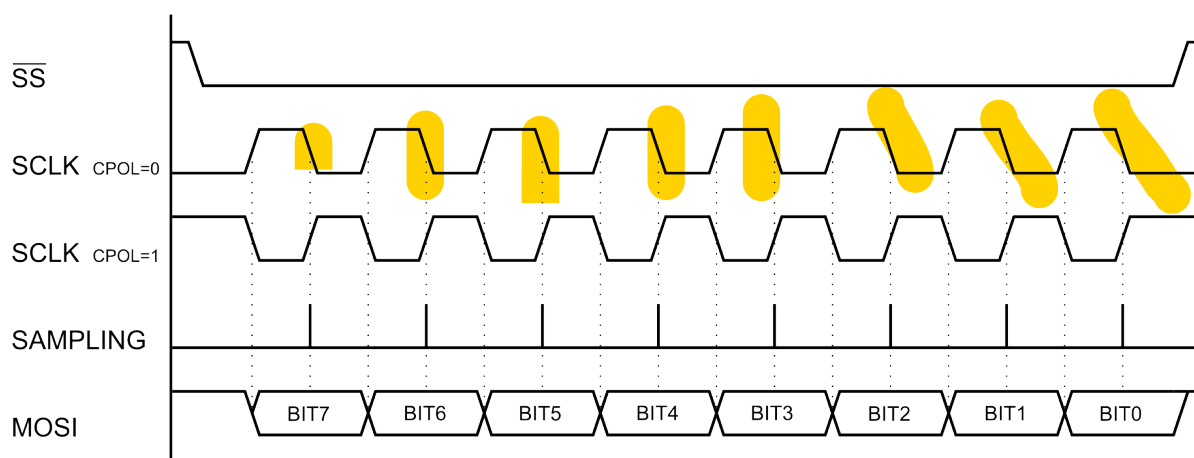


Figura 141: Diagramma temporale con il parametro CPHA=1.

Si noti che il caso CPOL=0 e CPOL=1 sono inclusi nello stesso diagramma e comporta, come detto, solo l'inversione dei fronti sul quale avviene una determinata operazione. Nella trattazione che segue si considera il caso CPOL=0 (modalità 1). La trasmissione dei dati avviene secondo la seguente sequenza:

- Il Master attiva lo *Slave Select*.
- Il Master attiva il Clock.
- Sul fronte di salita del clock il bit MSB dei registri a scorrimento interni, viene posto sulla linea MOSI (dal Master) e sulla linea MISO (dallo Slave).
- Sul fronte di discesa il Master e lo Slave campionano (*sampling*) il bit e lo traslano nel registro interno.
- Sul nuovo fronte di salita il nuovo bit viene posto sulle linee MOSI e MISO.
- Il discorso continua fino all'ultimo bit.

In questa modalità e nella 3 (CPOL=1, CPHA=1) la linea *Slave Select* rimane attiva per tutta la trasmissione. Se ulteriori byte devono essere inviati, la linea SS può rimanere ininterrottamente attiva, cosa non vera se CPHA = 0. Questo significa che moduli con interfaccia SPI con registri a 8 bit, possono leggere e scrivere periferiche che hanno registri a scorrimento 16 bit (come anche 3 o più byte), ma solo se si fa uso della modalità 1 o 3. In Figura 5 è riportato il caso in cui CPHA sia uguale a 0. Come nell'analisi precedente, il caso CPOL=0 e CPOL=1 sono inclusi nello stesso diagramma temporale.

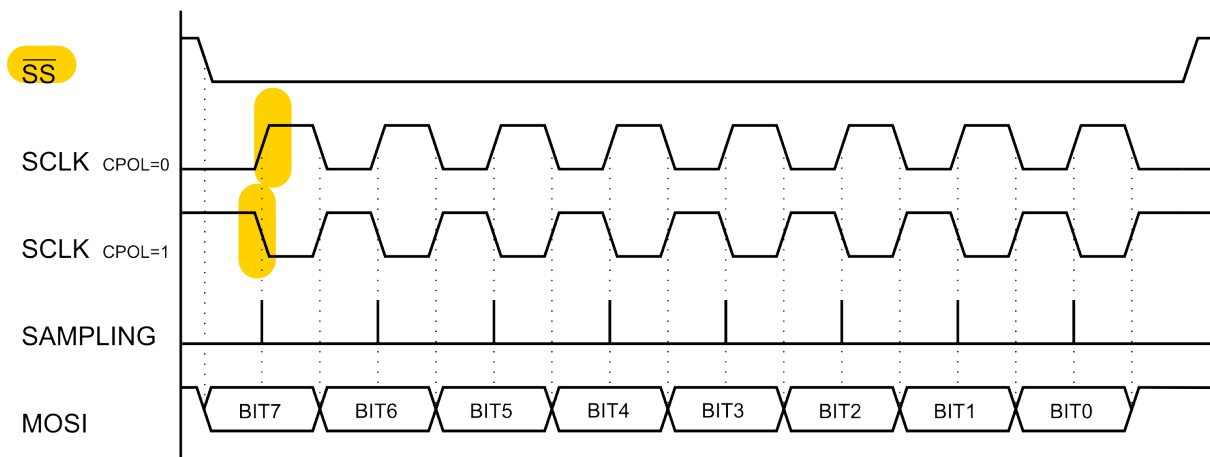


Figura 5: Diagramma temporale con il parametro CPHA=0.

Come per la Figura 4, anche in questo caso si analizza il caso CPOL=0 ovvero la modalità 0.

- Il Master attiva lo *Slave Select*.
- Sul fronte di discesa della linea SS il bit MSB dei registri a scorrimento interni, viene rispettivamente messo sulla linea MOSI (dal Master) e MISO (dallo Slave).
- Il master attiva il Clock.
- Sul fronte di salita il Master e lo Slave leggono (campionano) il bit e lo traslano nel registro interno.
- Sul nuovo fronte di discesa il nuovo bit viene posto sulle linee MOSI e MISO.
- Il discorso continua fino all'ultimo bit.

Per la modalità 0 e 2 è necessario che ad ogni invio di byte la linea SS venga disattivata e riattivata. Infatti il bit MSB viene posto rispettivamente sulle linee MOSI e MISO sul fronte di discesa della linea stessa. In queste modalità si capisce che non si può fare a meno della linea SS e che Master con registri interni a 8 bit possono colloquiare solo con Slave che hanno il registro interno di ugual dimensione.

Frequenza di trasmissione e livelli elettrici dell'interfaccia SPI

Arrivati a questo punto, dopo la carrellata di opzioni e accortezze vi sarete certamente resi conto che in tutta questa descrizione, pur avendo detto che il Master deve fornire il Clock, non ho fatto alcun cenno alla frequenza di trasmissione. Effettivamente, diversamente da uno standard asincrono come il protocollo RS232, l'interfaccia SPI non ha una frequenza operativa prefissata o standard, ed in particolare lavorando sui fronti del clock non è nemmeno necessario che abbia valori particolari (per esempio non c'è un valore minimo formale sotto il quale non si può andare). Questa peculiarità discende dal fatto che il clock è fornito dal Master e lo Slave si adegua a quest'ultimo. In uno Standard asincrono come il protocollo RS232, dal momento che il clock non viene fornito, è necessario che Master e Slave lavorino ad una determinata frequenza nota a priori (escluso il caso di utilizzo di algoritmi di auto baud rate).

Applicazioni con interfaccia SPI fanno uso di comunicazioni a frequenze che partono da poche decine di KHz fino ad arrivare a decine di MHz (80MHz-100MHz). Il limite

massimo viene a dipendere dalle periferiche di cui si sta facendo uso. Per esempio molti microcontrollori supportano frequenze massime fino a 10MHz. Per frequenze più alte si fa spesso uso di macchine a stati implementate dentro CPLD o FPGA, che permettono di raggiungere le frequenze massime sopra citate, che per altro potrebbero anche essere superate. Anche DSP di fascia alta permettono di superare il limite dei 10MHz che caratterizza molti microcontrollori. Il limite massimo della frequenza è in generale legato alla frequenza di clock e dalla tecnologia con cui vengono realizzati gli integrati.

L'interfaccia SPI, oltre a non specificare la frequenza operativa non impone alcun vincolo sulle tensioni che caratterizzano i livelli logici. Ciononostante si capisce che due dispositivi al fine di comunicare tra loro debbano lavorare con gli stessi livelli logici, altrimenti non si potrebbero capire. Qualora due dispositivi lavorino a tensioni diverse, per esempio uno a 3.6V e uno a 5V, è necessario far uso di traslatori di livello. Valori tipici di tensione sono: 1.8V, 3.3V, 3.6V e 5V.

Applicazioni dell'interfaccia SPI

Sebbene l'interfaccia SPI sembri un po' "disordinata" a causa della mancanza di uno standard, il suo utilizzo, una volta letto il Datasheet dello Slave al quale bisogna adeguarsi, è piuttosto facile, per cui è utilizzata in molte applicazioni.

L'applicazione madre è stata quella di estendere gli ingressi e le uscite di un microprocessore e microcontrollore, ma il suo utilizzo si è subito esteso ad altre applicazioni. Sistemi tipici che fanno uso dell'interfaccia SPI sono: *Real Time Clock Calendar*, memorie RAM, memorie EEPROM, amplificatori a guadagno variabile, controllori video, convertitori analogico digitali (ADC), convertitori Digitali Analogico (DAC), display LCD Alfanumerici e grafici.

Nello sviluppo delle applicazioni, sebbene sia il Master a comandare la comunicazione, come detto, è in realtà lo Slave ad imporre le regole. Lo Slave, di qualunque natura esso sia, potrà infatti comunicare con un Master in una determinata modalità che non può essere in generale cambiata. Per tale ragione, durante la scrittura del software si deve avere la premura di impostare il Master (frequentemente un microcontrollore) al fine di poter correttamente comunicare con lo Slave. Infatti il Master, diversamente dallo Slave ha la possibilità di essere programmato al fine di potersi adattare allo Slave. Da questo discende che in uno stesso Bus si potrebbero avere Slave che fanno uso di opzioni e modalità di comunicazione differenti. Naturalmente nel caso di uno Slave realizzato con un microcontrollore, si ha più flessibilità.

Il protocollo I²C e SPI a confronto

Le applicazioni dell'interfaccia SPI sembrerebbero essere le stesse del protocollo I²C, ed in effetti molte applicazioni potrebbero essere sviluppate usando l'uno o l'altro "standard". Ciononostante ci sono dei pregi e difetti che possono rendere una soluzione più attraente dell'altra:

Vantaggi dell'interfaccia SPI

- L'interfaccia SPI è full-duplex.
- Si possono raggiungere frequenze di trasmissione elevate.
- Interfaccia ed utilizzo semplici (mancanza di un protocollo).
- Facilità di isolare galvanicamente le linee unidirezionali.
- Lunghezza arbitraria dei dati da trasmettere.

Svantaggi dell'interfaccia SPI

- Utilizzo di un numero maggiore di linee.
- Mancanza di un protocollo e conferma di trasmissione (alcune volte è un vantaggio).
- Mancanza d'indirizzamento delle periferiche (escluso segnale SS).

Alcune volte si legge che tra gli svantaggi dell'interfaccia SPI rispetto ad altri protocolli (non solo il protocollo I²C), ci sia anche il fatto che non possa raggiungere elevate distanze, questo non è corretto. Sebbene l'interfaccia SPI sia stata pensata per il collegamento tra un Master e uno Slave a brevi distanze (stessa scheda), facendo uso di driver è possibile estendere la distanza alla pari di altri protocolli come per esempio il protocollo RS232, RS485. Facendo uso dei driver RS232 e RS485, l'interfaccia SPI¹ raggiunge le stesse distanze dei suddetti protocolli. In particolare il protocollo RS485 descrive il *physical layer* di una comunicazione, per cui questo può essere adottato anche per l'interfaccia SPI, visto che non fa uso di uno specifico *physical layer*.

¹ Alcune periferiche, come per esempio MCP23S17 (I/O Expander) supportano, in base al modello, sia il protocollo I²C che SPI. In particolare i modelli con interfaccia SPI ereditano la caratteristica dell'indirizzamento tipico del protocollo I²C, oltre alla linea SS.

Descrizione del modulo SPI

Il PIC18F4550 come molti altri microcontrollori PIC18 possiede il modulo interno MSSP, ovvero il *Master Synchronous Serial Port*. Tale modulo, come visto, oltre a poter essere configurato come modulo I²C, può essere configurato anche come modulo SPI. Le due modalità sono mutuamente esclusive, ovvero o si imposta il modulo come I²C o come SPI. Qualora si voglia supportare entrambi è necessario scegliere un microcontrollore con due moduli MSSP o implementare un protocollo via software². In Figura 6 è riportato il modulo MSSP in configurazione SPI.

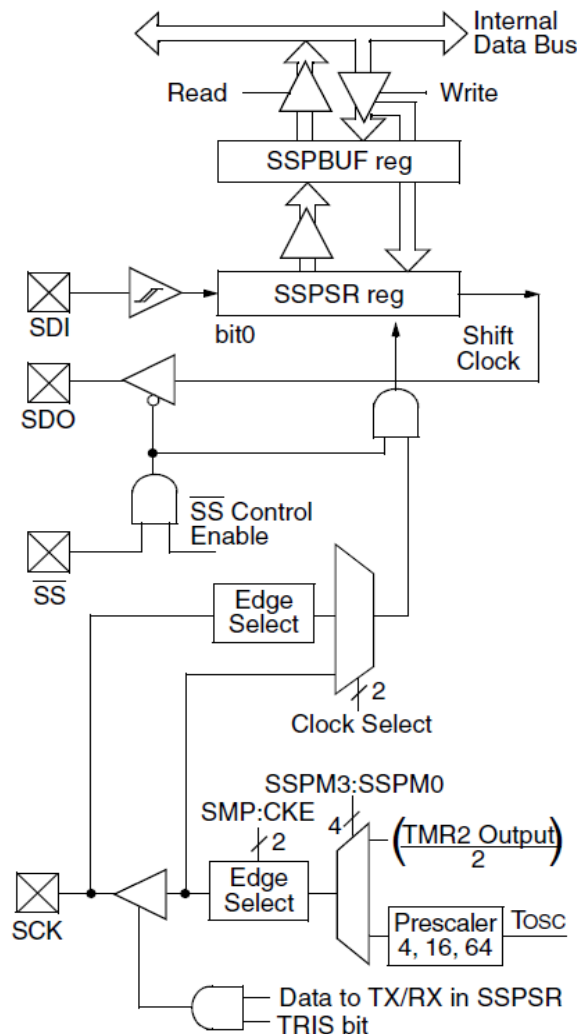


Figura 6: Modulo MSSP in configurazione SPI.

Si può notare la presenza del registro a scorrimento SSPSR e la presenza del buffer SSPBUF. Il modulo MSSP supporta l'interfaccia SPI in tutte e quattro le modalità descritte ed in particolare supporta sia la modalità a 3 che 4 linee. La linea SS può essere disattivata qualora non sia utilizzata. La linea SDI per l'ingresso dei dati, possiede un buffer con *Trigger di Schmitt* al fine di rimuovere disturbi sulla linea e rendere dunque la

² Qualora ci si trovi ad implementare un protocollo via software si preferisce in generale implementare l'interfaccia SPI piuttosto che il protocollo I²C, che in generale richiede maggior attenzioni.

trasmissione dei dati più affidabile.

Il modulo SPI può essere configurato sia in modalità Master che Slave per cui la linea SCK deve essere impostata come Output o Input a seconda dei casi. In particolare nel caso del Master la linea SCK deve essere impostata come Output, visto che il clock viene fornito dal Master. Nel caso dello Slave la linea SCK deve essere invece impostata come Input. Per questa semplice differenza, il Master e lo Slave hanno un comportamento diverso in caso il microcontrollore sia in stato di SLEEP. Nel caso del Master, entrando in SLEEP viene interrotta la comunicazione, che riprende dal punto interrotto nel momento in cui il microcontrollore si risveglia. Nel caso dello Slave il modulo, anche se il microcontrollore è in stato di SLEEP e i clock interni sono disattivi, può ricevere comunque dati e generare un Interrupt all'arrivo di un byte. Questo discende dal fatto che nel caso della modalità Slave, il clock viene fornito dal Master.

Alla ricezione di ogni byte, ovvero quando il byte ricevuto viene trasferito dal registro SSPSR al registro SSPBUF, viene settato il Flag SSPIF del registro PIR1. Nel caso in cui le interruzioni siano attivate, viene generata un'interruzione con la relativa priorità impostata.

Registri di configurazione del modulo SPI

I registri interni al PIC18F4550 per la configurazione del modulo MSSP in modalità SPI sono:

- SSPCON1
- SSPSTAT
- SSPBUF
- SSPSR (non accessibile direttamente)

Nome	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	SPPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	SSPPIE	ADIE	RCIF	TXIF	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	SPPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
TRISC	TRISC7	TRISC6	-	-	-	TRISC2	TRISC1	TRISC0
SSPBUF	Buffer di ricezione e trasmissione del modulo MSSP							
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
SSPSTAT	SMP	CKE	D/A'	P	S	R/W'	UA	BF

Il contenuto dei registri presentati nel Capitolo sono forniti per semplificare la comprensione delle librerie e del programma ma non rappresentano una fonte alternativa al Datasheet. A seconda del modello del PIC utilizzato i registri descritti potrebbero essere differenti o avere bit con significato differente.

Configurazione dei pin

Al fine di utilizzare il modulo SPI è necessario impostare come ingressi o uscite i pin associati alle linee SDI, SDO, SCK, SS

- La linea RB0 rappresenta la linea SDI del modulo SPI.
- La linea RB1 rappresenta la linea SCK del modulo SPI.
- La linea RC7 rappresenta la linea SDO del modulo SPI.
- La linea RA5 rappresenta la linea SS del modulo SPI.

Prima di abilitare il modulo e poter avviare la comunicazione bisogna propriamente impostare le linee di comunicazione. Bisogna in particolare distinguere due casi, ovvero il caso Master e Slave:

Master

- Impostare la linea RB0 come ingresso (disabilitare l'ingresso analogico).
- Impostare la linea RB1 come uscita (disabilitare l'ingresso analogico).
- Impostare la linea RC7 come uscita.
- Impostare la linea RA5 come uscita (disabilitare l'ingresso analogico).

Slave

- Impostare la linea RB0 come ingresso (disabilitare l'ingresso analogico).
- Impostare la linea RB1 come ingresso (disabilitare l'ingresso analogico).
- Impostare la linea RC7 come uscita.
- Impostare la linea RA5 come ingresso (disabilitare l'ingresso analogico).

In particolare tra i due diversi casi, le linee SCK e SS assumono un valore diverso, vista la differente funzione delle linee.

La configurazione dei pin riportata è valida solo per il PIC18F4550 e per la maggior parte dei PIC18 con modulo USB integrato. La posizione delle linee potrebbe cambiare a seconda del PIC utilizzato.

Registro SSPSTAT: MSSP 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 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Leggenda

R = Readable bit
-n = Value at POR

W = Writable bit
1 = Bit is set

U = Unimplemented bit read as 0
0 = Bit is cleared

S : Settable bit
x = Bit is unknown

Bit 7 **SMP** : Slew rate Control bit

Modalità SPI Master

1 : Data In campionato alla fine di output time
0 : Data In campionato al centro di output time

Modalità SPI Slave

Il bit deve essere posto a 0 quando si è in modalità Slave.

Bit 6 **CKE** : Selezione del Clock del modulo SPI

1 : La trasmissione avviene alla transizione del clock da Active ad Idle
0 : La trasmissione avviene alla transizione del clock da Idle ad Active

Bit 5 **D/ \overline{A}** : Data/ $\overline{\text{Address}}$ bit

Utilizzato solo in modalità I²C.

Bit 4 **P** : Stop bit

Utilizzato solo in modalità I²C.

Bit 3 **S** : Start bit

Utilizzato solo in modalità I²C.

Bit 2 **R/ \overline{W}** : Read/ $\overline{\text{Write}}$ Information bit

Utilizzato solo in modalità I²C.

Bit 1 **UA** : Update Address (10Bit mode only)

Utilizzato solo in modalità I²C.

Bit 0 **BF** : Buffer Full Status Bit

In Modalità TX

1 : SSPBUF è pieno, ricezione completata.
0 : SSPBUF è vuoto, ricezione non completa.

Registro SSPCON1: MSSP Control Register

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 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Leggenda

R = Readable bit
-n = Value at POR

W = Writable bit
1 = Bit is set

U = Unimplemented bit read as 0
0 = Bit is cleared

S : Settable bit
x = Bit is unknown

Bit 7

WCOL : Write Collision Detect Bit (solo in trasmissione)

1 : SSBUFF è scritto prima che la trasmissione del byte precedente è terminata. Deve essere resettato via Software,
0 : Nessuna Collisione.

Bit 6

SSPOV : Receive Overflow Indicator bit

Modalità SPI Slave

1 : Un byte è stato ricevuto mentre SSBUFF conteneva ancora il vecchio byte. Il Byte in ingresso deve essere sempre letto anche se il Master deve solo inviare dati.
0 : Nessun Overflow.

Bit 5

SSPEN : Master Synchronous Port Enable bit

1: Abilita la porta seriale, le linee SCK, SDO, SDI, SS devono essere opportunamente impostate.
0: Disabilita la porta seriale.

Bit 4

CKP : Selezione della polarità del Clock

1: Lo stato Idle equivale al livello alto del clock.
0: Lo stato Idle equivale al livello basso del clock.

Bit 3-0

SSMP3:SSPM0 : Master Synchronous Serial Port Mode Select bits

0101: Modalità SPI Slave, SS pin disabilitato.
0100: Modalità SPI Slave, SS pin abilitato.
0011: Modalità SPI Master, clock = TMR2 out/2.
0010: Modalità SPI Master, clock = Fosc/64.
0001: Modalità SPI Master, clock = Fosc/16.
0000: Modalità SPI Master, clock = Fosc/4.