

La connessione SPI

SPI (*Serial Peripheral Interface*) è uno standard per lo scambio dati con una interconnessione seriale sincrona del genere Master-Slave.

SPI è inizialmente stato studiato da Motorola e diventa Microwire per National Semiconductors. Esistono poi una estensione QSPI (*Queued Serial Peripheral Interface*) e un MicrowirePLUS.

Il **Master** è il dispositivo che comanda il sistema; in generale si tratterà di un microcontroller.

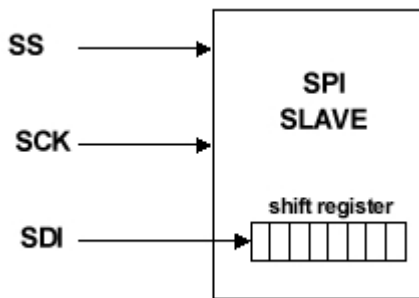
Esso ha la possibilità di inviare e ricevere dati e comandi e di iniziare la sessione di trasmissione. Fornisce anche il clock di sincronismo dello scambio di dati.

Lo **Slave** è un dispositivo periferico che può ricevere e inviare dati, ma non può inviare comandi, né iniziare una sessione di trasmissione. Il clock con cui trasmette o riceve dati è in ogni caso fornito dal Master e lo Slave non ha alcun controllo su questa linea.

E' possibile collegare più Slave ad uno stesso Master, ma nel sistema SPI **non sono previsti** più Master sullo stesso circuito dati.

Il protocollo SPI è molto semplice, dato che non prevede situazioni di arbitraggio multi-Master, né segnali di handshake. E, in effetti, lo standard SPI non ha una vera definizione formale, ma è piuttosto una serie di linee guida che i costruttori seguono. Questo rende SPI molto flessibile, ma anche molto variegato.

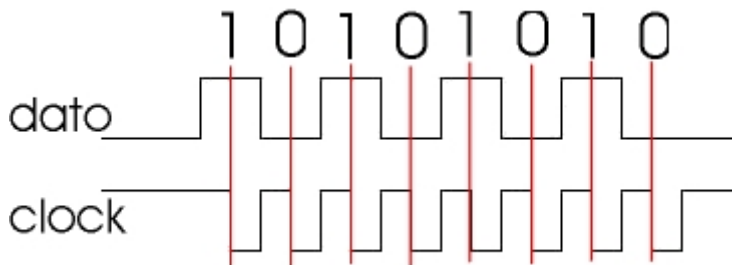
Si possono fare rientrare nella "categoria" SPI tutti quei dispositivi che sono costituiti essenzialmente da uno shift register. E, in pratica, proprio shift register del genere **LS/HC164**, **594/5** e **CD4094** sono un esempio di periferica SPI minimale.



L' hardware della comunicazione è costituito da una linea dati chiamata **SDI** (*Serial Data Input*) e da una di clock **SCK** (*Serial Clock*), per cui una periferica SPI può essere considerata analoga ad uno shift register del genere **SIPO** (*Serial In-Parallel Out*).

La periferica, o *Slave*, dispone solitamente di un sistema di chip select che il Master dovrà comandare per attivare la comunicazione (linea **SS** o **CS**).

Da parte sua il Master è essenzialmente uno shift register di tipo **PISO** (*Parallel In-Serial Out*).



Il dato viene posto sulla linea di trasmissione, un bit per volta, partendo da quello più significativo (MSb).

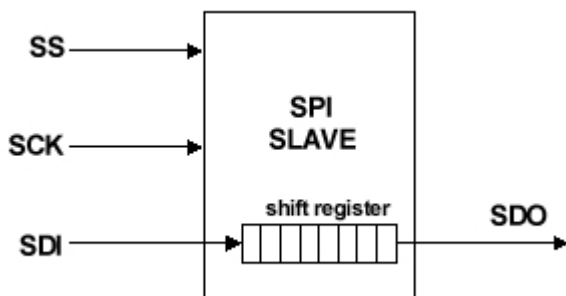
Per ogni bit, viene inviato un impulso di clock su cui sarà sincronizzata la ricezione.

Nel diagramma, come esempio la trasmissione del byte 10101010 con clock sul fronte di discesa.

Non è determinato il fronte di sincronismo del clock, che, a seconda della periferica, potrebbe essere anche quello di salita; e neppure lo stato della linea di clock quando è inattiva. Nel diagramma è a livello alto, ma potrebbe anche essere l' opposto. Queste opzioni sono descritte più avanti.

Non esiste alcun particolare handshake per la trasmissione che parte quando il Master seleziona lo Slave con cui vuole colloquiare, attraverso il suo chip select ed il successivo invio del clock secondo cui fare scorrere i dati tra gli shift register.

La più semplice simile configurazione necessita solamente di dati in partenza dal Master e in arrivo allo Slave, ma sono state realizzate un gran numero di altre funzioni, anche complesse (EEPROM, RAM, FLASH, AD, RTC, codec, ecc.), che devono inviare dati e rispondere a comandi.



Poichè la linea **SDI** è mono direzionale, lo standard aggiunge una linea **SDO** che la periferica usa per trasmettere verso il Master.

Se facciamo riferimento ad un generico shift register, la linea **SDO** sarebbe collegata al pin **Qn** di uscita del registro.

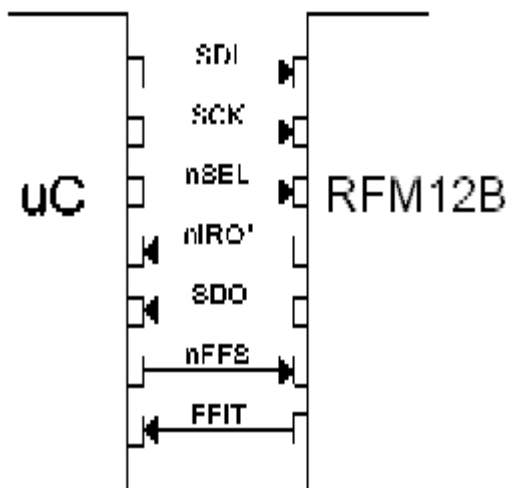
In pratica, quindi, una periferica SPI minimale trasmette il contenuto del registro shiftandolo sulla linea SDO mentre dall' altro estremo lo con i dati in arrivo.

Il fatto che le linee **SDI** e **SDO** siano mono direzionali permette di realizzare sistemi in cui entrambe siano attive contemporaneamente, realizzando una comunicazione **full duplex**: mentre arrivano dati dal Master sullo SDI, lo Slave invia altri dati sullo SDO.

Questo richiede una certa complessità di registri, buffer e sistemi di controllo all' interno dei dispositivi SPI.

SPI può operare a frequenze di clock molto elevate, anche oltre 10 MHz. La frequenza del clock dipende in parte dalle caratteristiche della periferica, ma molto dalla qualità del collegamento fisico tra questa e il Master.

L'ambito di SPI è, in realtà, assai vario, in quanto sono disponibili moltissimi tipi di periferiche, da quelle più semplici a quelle più complesse, come AD/DA multi canali, RTC, interfacce con altri bus (LAN, CAN, ecc), termometri, codec, ecc.



Alcune di esse rispondono a un set di comandi, inviando dati propri, oppure hanno pin aggiuntivi che permettono di implementare funzioni complesse nello scambio dei dati.

Ad esempio in figura l' ISM band transceiver module RFM12B, un modulo di comunicazione radio con interfaccia seriale sincrona SPI. Ai segnali standard sono aggiunti quelli relativi alle funzioni del modulo.

Con componenti come questo, la comunicazione sarà gestibile dai moduli sincroni integrati nei microcontroller, mentre occorrerà del software ad hoc per gestire le funzioni aggiuntive.

Esistono poi numerosi dispositivi simil-SPI, nel senso che scambiano dati secondo lo stesso principio di una linea dati sincrona con il clock su un'altra linea, ma che possono non corrispondere alle altre specifiche del bus definito da Motorola.

La connessione SPI

In ogni caso, la connessione **SPI** è definita per un minimo di 4 conduttori fisici:

- **SDO** - o **MOSI** oppure **SIMO**, linea di uscita dei dati dal Master e ingresso per lo Slave
- **SDI** - o **MISO** oppure **SOMI**, linea di ingresso dei dati nel Master e uscita per lo Slave
- **SCK** - clock di sincronismo della trasmissione generato dal Master, anche indicato come **SCLK** o **SCL** o **CK**
- **SS** - *Slave Select*, anche indicato come **CS** (*Chip Select*) che, a livello basso, abilita la periferica allo scambio dei dati.

I documenti di Motorola aggiungono anche:

- **MOMI** - Master Output, Master Input
- **SISO** - Slave Input, Slave Output

Questi sono pin bidirezionali che unificano MISO e MOSI in una unica linea.

Le denominazioni dei pin dipendono dal costruttore del dispositivo e possono essere abbastanza varie da dare origine a confusioni.

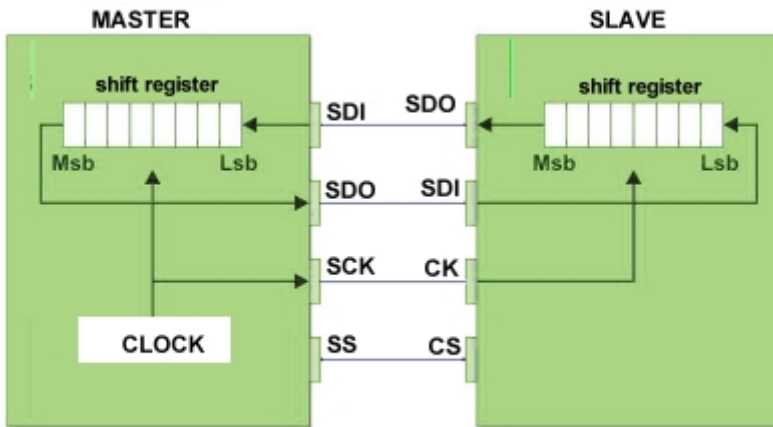
SCLK	SCK, CLK, CK
MOSI	SDI, DI, DIN, SI
MISO	SDO, DO, DOUT, SO
SS	nCS, CS, CSB, CSN, nSS, STE

NOTE

Note:

1. Esiste in pratica una notevole indeterminazione nei termini **SDI** e **SDO** tipici dei PIC in quanto hanno funzione opposta a seconda siano sul Master o sullo Slave. Meglio paiono gli acronimi **MISO** (*Master Input-Slave Output*) e **MOSI** (*Master Output-Slave Input*), che sono quelli usati dallo standard.
2. Da notare che esiste anche una soluzione **SPI 3-wires**, con una sola linea dati bi-direzionale, definita **SISO**, che trasporta i dati in entrata e in uscita su un unico conduttore. Analogo è **I²S**. Questa modalità normalmente non è supportata dai moduli di comunicazione sincrona dei microcontroller, ma può essere facilmente implementata via software.

Di base, il collegamento tipico tra Master e Slave SPI è questo:



semplificando, si tratta di uno scambio di dati tra shift register, il cui clock è sotto controllo esclusivo del Master.

Questo vuol dire che se lo Slave deve trasmettere dati, sarà compito del Master inviare il clock necessario.

Osserviamo che in un collegamento completo si realizza un "anello" in cui la linea **SDO** del Master invia dati all' ingresso **SDI** dello Slave e la linea **SDI** del Master riceve dati dallo **SDO** dello Slave.

In questo senso, un dispositivo SPI completo non è un solo ricevitore o solo trasmettitore, ma entrambi.

Inoltre il dato in ritorno consente al Master di verificare che lo Slave è attivo.

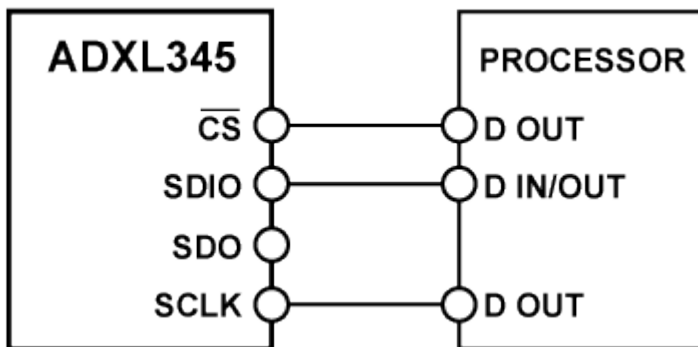
Se la linea **SDO** dello Slave è collegata allo **SDI** del Master, questo riceverà dati fino a che genererà impulsi di clock. Se i dati in arrivo non interessano, saranno scartati, ma, ripetiamo, se il Master vuole ricevere dati dallo Slave, deve inviare impulsi di clock, dato che lo Slave non dispone di clock proprio per la comunicazione; in questo caso il Master porrà sulla linea dati un valore "dummy", tipicamente 0 o 1, che lo Slave a sua volta scarnerà, essendo posto nello stato di trasmissione.

E' comunque necessario considerare le specifiche di ogni periferica, dato che alcune sono in grado di operare in **full duplex**, ovvero ricevere dati e nello stesso tempo trasmetterne altri, utilizzando sistemi interni di buffer.

Altri esempi di connessione SPI riguardano periferiche che hanno caratteristiche particolari.

A questo proposito si deve tener presente che una trasmissione sincrona composta da dato e clock è uno dei metodi più semplice ed efficaci per lo scambio di dati. Ne consegue che un elevato numero di periferiche utilizzano questo sistema, anche se non sempre sono considerabili come veri SPI, anche se, con opportuni adeguamenti software e/o hardware, è possibile inserirle in bus SPI.

Di questa categoria fanno parte molti dei dispositivi chiamati 2-wires o 3-wires, dove, ad esempio, la linea dati diventa bi direzionale o la lunghezza del dato è maggiore di 8 bit.

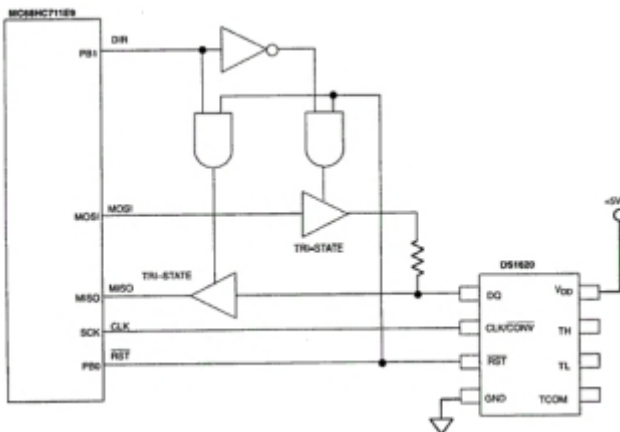


Ad esempio l' [accelerometro digitale ADXL345](#) di Analog Devices può operare sia con la connessione standard a 4 fili (SDI e SDO), ma anche a tre fili, dove il pin SDI diventa un **SDIO** bidirezionale.

Tra l' altro la stessa periferica può essere configurata per comunicare anche in I²C.

08654-003

Per contro, altre periferiche richiedono un adeguamento più massiccio per essere usate con interfacce SPI.



Ne è un esempio il sensore di temperatura DS1620, in cui:

- la linea dato DO è bi direzionale
- la lunghezza del dato di temperatura reso è a 9 bit invece che 8.
- il chip select RTS è valido a livello alto

Un adattamento allo standard Motorola SPI richiede un hardware aggiuntivo per adeguare i segnali.

L' [AN85 di Maxim](#) da cui è tratta l' immagine, dettaglia i particolari dell' intervento e della gestione software.

E, in effetti, è importante rilevare **la lunghezza del dato** richiesto o fornito dalla periferica, in quanto il "byte" di 8 bit può non essere l' unica dimensione possibile, tanto che alcune periferiche possono avere lunghezza dati variabile a seconda delle operazioni.

Dal punto di vista della trasmissione e ricezione, questo comporta una diversificazione degli algoritmi, con la fornitura del numero adeguato di impulsi di clock e la gestione dei buffer che, nei moduli integrati nei microcontroller, hanno solitamente la lunghezza standard di 1 byte= 8 bit.



Per quanto detto, e quanto indicato più avanti per le connessioni multi-Slave e le polarità del clock, **è indispensabile consultare con cura i fogli dati** delle periferiche utilizzate, onde risolvere eventuali incompatibilità dirette nella configurazione che si è prevista.

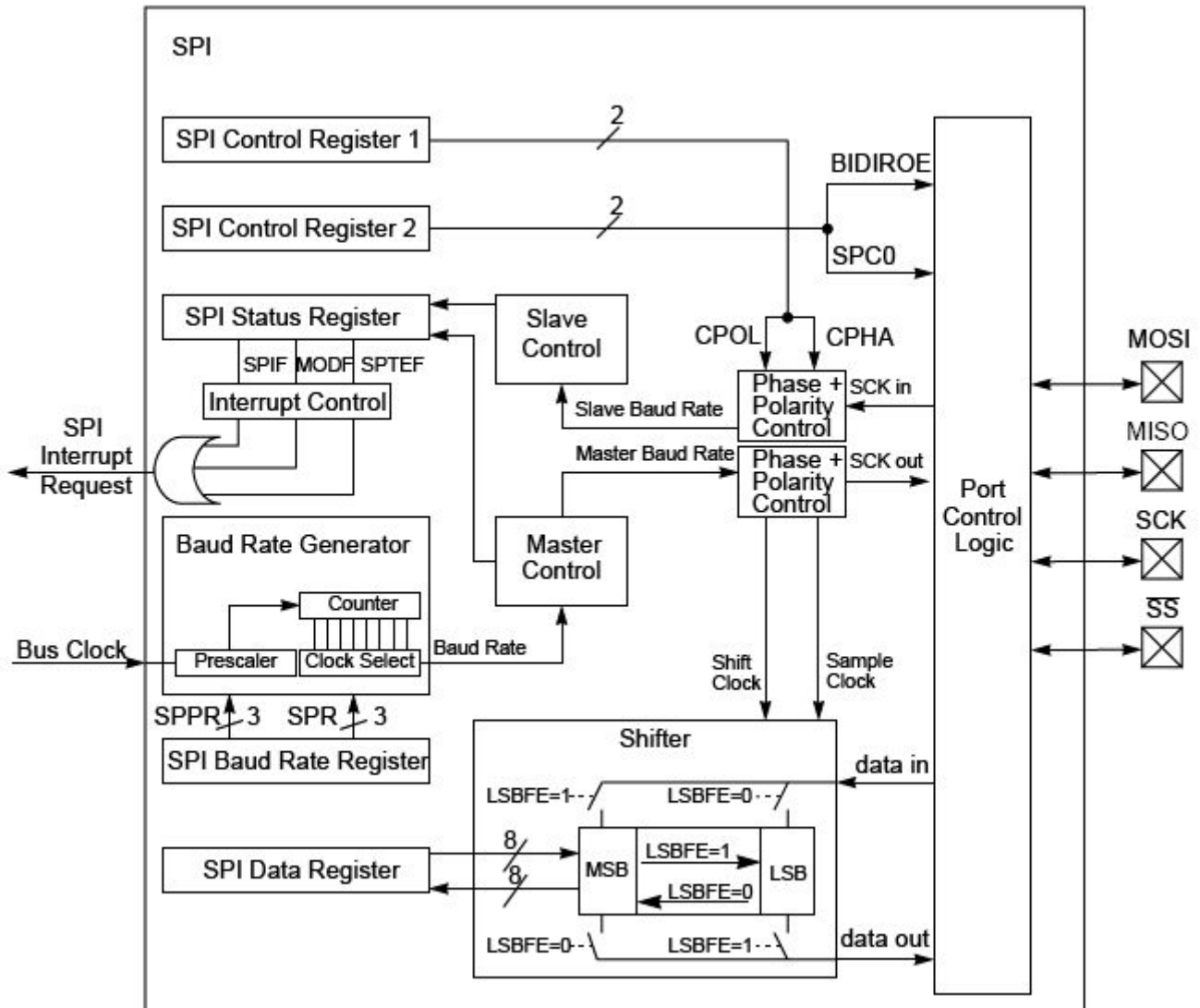
Ne risulta che in alcuni casi è più semplice emulare via software la connessione simil-SPI tra microcontroller e periferica piuttosto che utilizzare i moduli di comunicazione seriale sincrona integrati.

I moduli di comunicazione sincrona

Un drive per shift register può essere scritto con semplicità per qualsiasi microcontroller, ma clock elevati e trasmissioni full duplex costituiscono situazioni gravose da emulare in software.

Per evitare questo carico all' utente, nei principali microcontroller viene integrata una periferica specifica per la comunicazione SPI, dotata di meccanismi di buffer e di flag per la gestione sia in polling che in interrupt.

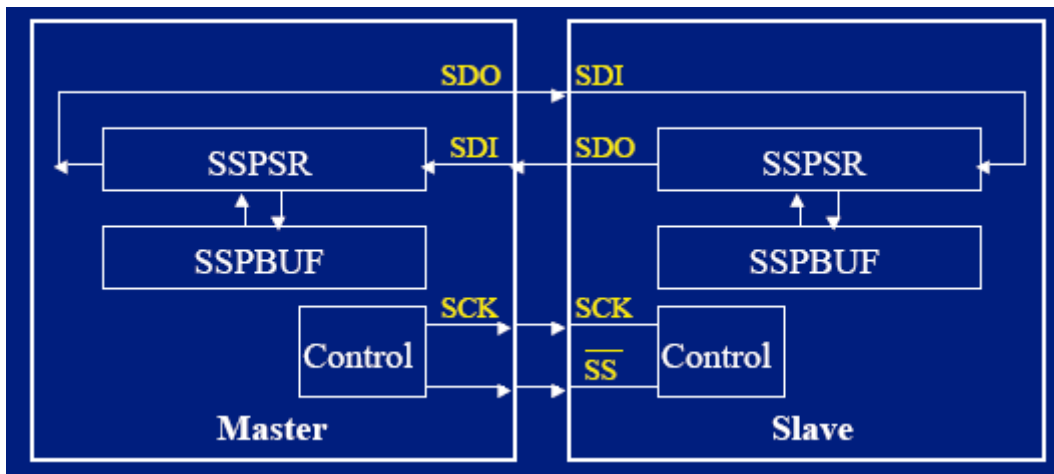
L' architettura di un modulo di gestione SPI è presentata nel documento Motorola **S12SPIV3/D**, SPI Block Guide V03.06, di cui si consiglia la consultazione per un maggiore approfondimento sulle caratteristiche del bus.



da cui possiamo notare la complessità che non si limita ad un solo registro di spostamento, comprende una complessa struttura di gestione dello shift register e delle polarità del clock per consentire una comunicazione interamente automatizzata anche nel caso di un full-duplex. I moduli integrati nei microcontroller ricalcano questa struttura, ma possono non essere esattamente identici, per cui occorrerà, prima di utilizzarli, documentarsi sui fogli informativi.

Nel modulo di gestione seriale sincrona tipico di un microcontroller, per i PIC chiamato **MSSP** o **SSP** (*Master Synchronous Serial Peripheral* o *Synchronous Serial Peripheral*), sono presenti, oltre agli shift register (**SSPSR** nell' immagine sotto), anche dei buffer (**SSPBUF** nell' immagine) per conservare il dato ricevuto.

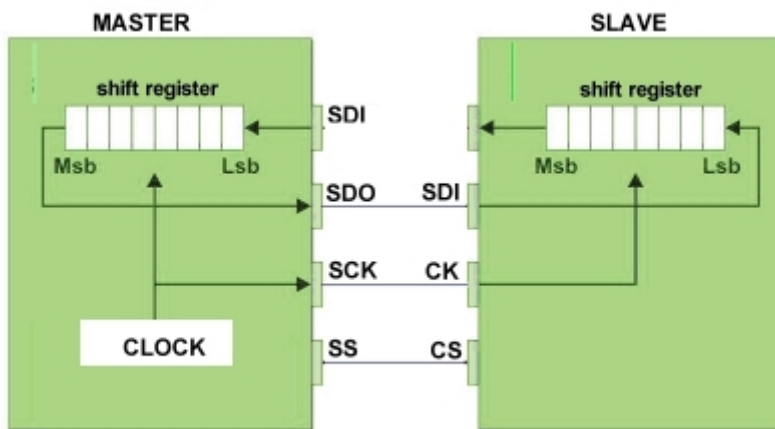
Una struttura bufferata è indispensabile alla gestione di trasmissioni che possono raggiungere velocità molto elevate, onde dare tempo al microcontroller di trattare i dati ricevuti e trasmessi.



Se quanto lo Slave trasmette non serve alla comunicazione, il Master dovrà comunque effettuare la lettura del buffer e scartare il risultato, oppure utilizzarlo anche solo come check dell' attività dello Slave.

Il modulo tipico, poi, contiene una logica in grado di controllare lo scambio di dati sincrono sulle linee SDO e SDI in modo del tutto autonomo, permettendone una gestione sia in polling che a interrupt, con vari registri preposti alle varie funzioni della comunicazione. Questo consente una grande flessibilità di impiego, potendo soddisfare anche le esigenze di scambio dati ad alta velocità che avvengono in concomitanza con altre operazioni del microcontroller.

E' ovvio che la connessione ad anello serve solamente se occorre ricevere dati dalla periferica. In un sistema minimale, si potrà risparmiare un conduttore (e un pin del microcontroller).



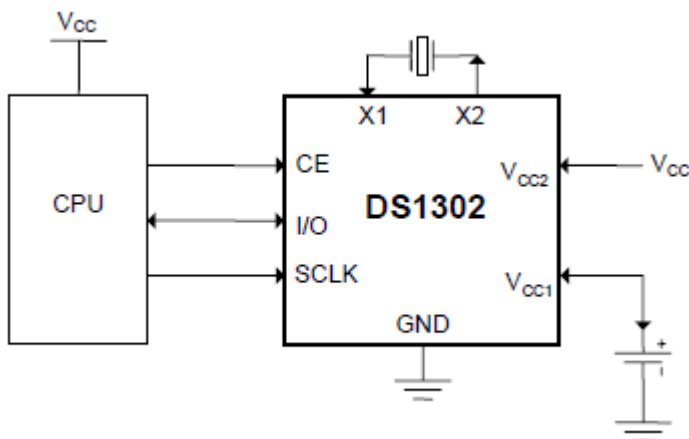
Il segnale di selezione da inviare allo Slave, se non gestito direttamente dal modulo, viene ottenuto da uno degli I/O disponibili sul microcontroller che opera come Master.

Microcontroller che dispongono di moduli dedicati a SPI potranno avere la possibilità di configurare, anche opzionalmente, una linea SS. Per alcuni moduli questa linea è valida solo in modo Slave, per altri è utilizzabile anche in modo Master per selezionare uno Slave.



Nel modulo MSSP dei PIC è previsto un pin **SS**, ma esso **funziona come chip select del modulo stesso esclusivamente quando questo è programmato per operare come Slave**. Nel caso in cui il microcontroller operi come Master, il chip select dello Slave dovrà essere ricavato da una delle linee di I/O disponibili

E, come accennato prima, è necessario osservare la polarità del chip select richiesto dalla periferica collegata, dato che generalmente si fa conto su CS/SS valido a livello basso, ma può capitare anche l'opposto.



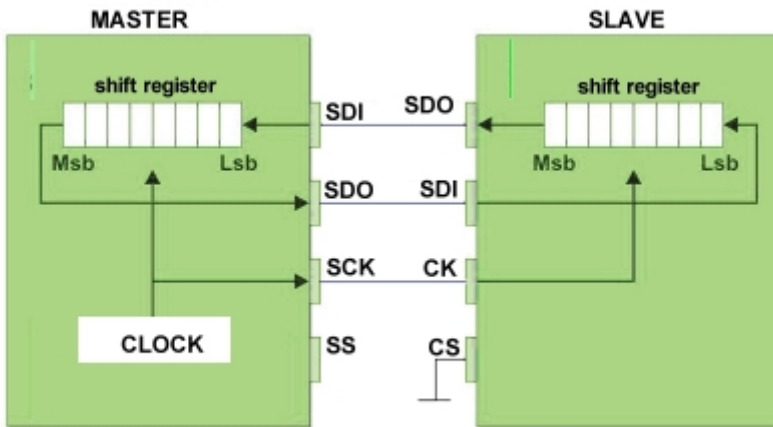
Ad esempio, l' RTC **DS1302 di Maxim** è un simil-SPI a tre fili con una linea bi-direzionale chiamata semplicemente I/O.

E' un esempio di periferica con il **chip select (CE) valido a livello alto**; tutte le linee di comunicazione sono **pull down** all' interno dell' integrato.

Inoltre, nelle operazioni sulla RAM integrata, può scambiare dati in burst.

Quindi, di nuovo, si richiama la necessità di **verificare con cura quali segnali sono necessari** per la periferica che si è scelta, dato che SPI è uno standard molto lasco e, al di fuori dell' ambiente Motorola, ha dato origine a una quantità notevole di dispositivi con caratteristiche molto varie.

Un ulteriore risparmio di conduttori è possibile se non si utilizza il segnale di selezione della periferica SS.



Il segnale di selezione dello Slave può essere omesso se il Master accede ad una sola periferica, che sarà sempre attiva ponendo il suo chip select al livello adeguato (solitamente basso, ma può anche essere l' opposto).

La selezione va utilizzata anche in questo caso se, in condizioni di non-selezione, la periferica va in basso consumo ed è necessario minimizzare l' energia impegnata.

La selezione della periferica diventa necessaria se il Master ne controlla diverse.



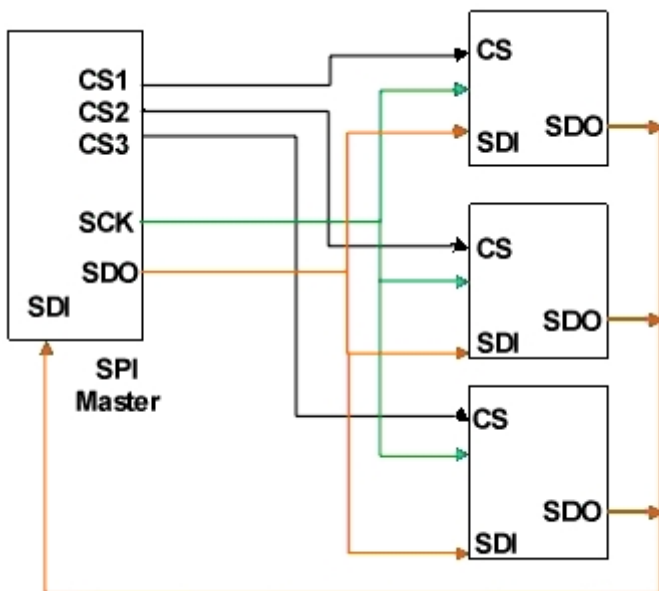
Come regola generale, il segnale di **CS** dovrebbe essere sempre usato in quanto **in varie periferiche ha funzione di ripristinare lo Slave in modo da essere pronto a ricevere il prossimo dato.**

Con questo si intende che il chip select non solo attiva la periferica, ma solitamente ha anche funzione di resettare i contatori degli shift register, riallineando i registri inteni per ricevere correttamente i prossimi dati.

Multi Slave

Quanto detto è valido nel caso di un semplice sistema Master-Slave, che, peraltro, è una delle situazioni più comuni. Ma SPI ammette anche un "bus" con più Slave.

In questo caso sono possibili due tipi di connessione:



Una prima possibilità, di uso generale, è quella di prevedere una **distribuzione in parallelo dei segnali SDI e SDO.**

Il Master comanda tante linee di chip select quante sono le periferiche. In sostanza, si tratta di utilizzare pin di I/O del microcontroller, comandandoli in modo da abilitare una sola periferica per volta.

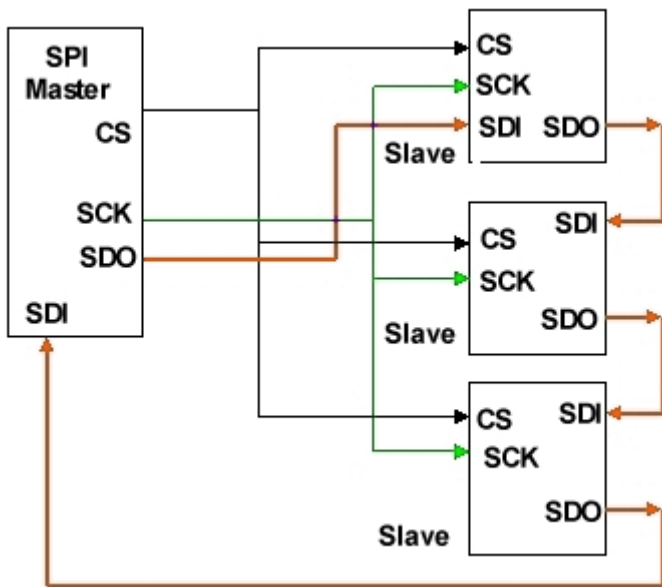
Questo sistema consente di collegare **tante periferiche quanti sono i chip select disponibili**, il che può limitare le possibilità di espansione in microcontroller con un basso numero di pin.

In genere, però, non si tratta di un grosso problema dato che è raro che in solo Master comandi un gran numero di Slave, vista anche la necessità di **limitare a lunghezze ragionevoli i conduttori del bus per consentire i transfer rate elevati che offre la connessione SPI.**

Il vantaggio principale di questa configurazione è che **ogni periferica, essendo abilitata separatamente, potrà disporre di una gestione propria.**

Sono così mescolabili periferiche con, ad esempio, chip set validi alti e bassi o con lunghezza di dati differente e modi di clock diversi: le periferiche non selezionate saranno disabilitate e indifferenti ai segnali sui pin, che si troveranno in 3-state, quindi con carichi ad alta impedenza che non influenzano il bus, permettendo al Master di configurarsi diversamente per ognuna di esse..

Problemi, però, possono insorgere in quanto **il protocollo originale SPI di Motorola si aspetta che tutti i dispositivi Slave utilizzino uscite MISO/DOUT 3-state**, ma alcune periferiche non hanno uscite 3-state, dato che prevedono una connessione in daisy chain (vedi sotto). Questa architettura, rispetto alla precedente, riduce il numero dei pin usati dal microcontrollore, perché solo una linea CS è necessaria. Poiché DOUT di ogni dispositivo è destinato a guidare il pin DIN di un altro, deve rimanere attivo. Se l'output DOUT è andato ad alta impedenza, l'input DIN di un dispositivo di daisy chain successivo non riceverebbe il dato circolante, bloccando il loop.



Il diagramma a lato presenta la soluzione con le periferiche in **daisy chain**.

I dati in uscita dal master sono inviati nella prima periferica, la cui uscita SDO è concatenata con l'ingresso della successiva e così via, fino all'ultima la cui uscita richiude l'anello sullo SDI del Master.

In questo caso, dato che tutte le periferiche devono essere attive per passarsi i dati nella catena, è necessario un solo segnale di chip select.

Questo collegamento non è possibile con tutti i tipi di periferiche, che devono avere **la stessa modalità di funzionamento e la stessa polarità di chip select**.

Questa soluzione è molto meno elegante di quella possibile con il bus I²C che seleziona le periferiche con indirizzi, ma, in alcuni casi, risolve con semplicità il problema di comandare più periferiche uguali, come ad esempio espansioni di I/O realizzate con catene di registri SIPO oppure di display controller.

SPI non descrive un modo specifico per implementare sistemi **multi-Master**, che, vista l'assenza di un protocollo di gestione della priorità sul bus, non è di facile soluzione. Però alcuni dispositivi SPI possono sostenere ulteriori segnali che rendono tali implementazioni possibili, così che, con un software opportuno, sarebbero realizzabili sistemi del genere. Tuttavia, c'è da dire che nella pratica situazioni multi-Master non sono comuni e, vista la loro non semplice realizzazione, nel caso in cui questo sia necessario, sarà più opportuno rivolgersi a standard che prevedono questa possibilità, come I2C.

Trasmissione dati in SPI

La trasmissione vera e propria avviene posizionando il bit da trasmettere sulla linea SDO e sincronizzandolo con un impulso di clock sulla linea SCK.

Una sessione di comunicazione si svolge in questo semplice modo:

1. il Master che seleziona lo Slave con cui vuole comunicare attraverso la linea SS. Solitamente i chip select sono a livello basso, ma è opportuno verificare questo, dato che alcune periferiche (ad esempio RTC) possono avere un chip select attivo a livello alto.
2. Dopo questo, solitamente si posiziona una breve attesa per la stabilizzazione o una attesa più lunga a seconda di quanto la periferica necessita per attivarsi.
3. Ora il Master genera il clock massimo che la periferica e la connessione fisica possono supportare e inizia l'invio dei dati, ponendo sul bus per primo il bit più significativo (MSB)
4. Durante l'emissione del clock, se esiste una connessione ad anello, il Master invia dati sulla linea MOSI e riceve quelli dalla periferica sulla linea MISO
5. Quando il clock viene sospeso, anche la trasmissione ha termine. Il Master rimuove il chip select della periferica, disabilitandola

Si tratta quindi, come abbiamo detto, di **un sistema seriale sincrono**, ma nel quale, a differenza di altri, **la precisione del clock non è fondamentale**, dato che **solamente un elemento** del bus, il Master, **lo genera** e tutti gli Slave si sincronizzano con questo.

Teoricamente, quindi, il clock può assumere qualsiasi valore o anche cambiare durante la trasmissione oppure essere sospeso, fermando la trasmissione stessa che riprende al momento del riavvio del clock. Tra l'altro, in un sistema multi-Slave non in daisy chain il Master può utilizzare clock diversi per ogni Slave, sfruttando quindi al massimo le loro possibilità.

La soluzione SPI è poco costosa in quanto, appunto, gli Slave non richiedono generatore di clock, ma neppure logiche complesse per la gestione del protocollo, che è semplice, perché basato sostanzialmente su un registro di spostamento (shift register) del genere **PIPO** (*Parallel In-Serial Out*) nell'unità che trasmette e un **SIPO** (*Serial In Parallel Out*) in quella che riceve.

Un sistema sincrono, in questo senso, è migliore di uno asincrono, come ad esempio RS-232, dove il clock determina la lunghezza degli impulsi dei dati e deve quindi essere molto preciso, richiedendo un oscillatore a quarzo ad entrambe le estremità della connessione. In SPI non serve questa precisione ed è adeguato anche un economico oscillatore RC. Così una comunicazione SPI emulata in software può essere interrotta da chiamate interrupt di altre periferiche senza pregiudicare i dati.

Abbiamo accennato che **SPI è uno standard poco costrittivo** e lascia molta autonomia ai progettisti di dispositivi, con il risultato che, in pratica, sono possibili varie combinazioni rispetto allo stato della linea di clock a riposo (*idle*) ed al fronte attivo su cui sincronizzare il dato. Un paio di parametri chiamati polarità clock (**CPOL** o **CKP**) e la fase del clock (**CPHA** o **CKE**) determinano i fronti del segnale di clock su cui sono sincronizzati e campionati i dati, oltre allo stato inattivo della linea del clock.

Ciascuno dei due parametri ha due stati possibili, il che dà origine a 4 combinazioni possibili, che sono tutte incompatibili tra di loro. Quindi una comunicazione SPI deve avere Master e Slave che impiegano lo stesso modo; se gli Slave hanno modi diversi, il Master dovrà modificare in diverse configurazioni per ognuno degli Slave.

I 4 modi sono identificati con un numero o una coppia di numeri.

Per CPOL = 0 il livello idle del clock è zero

- per CPHA = 0, vengono acquisiti i dati sul fronte di salita del clock (transizione low→high) e dati sono inviati sul fronte di discesa (transizione high→low).
Per CPHA = 1, vengono acquisiti i dati sul fronte di discesa e dati vengono emessi su quello di salita.

Per CPOL = 1 il livello idle del clock è uno (inverso del precedente CPOL = 0)

- per CPHA = 0, vengono acquisiti i dati sul fronte di discesa e dati vengono propagati fronte di salita.
Per CPHA = 1, vengono acquisiti i dati fronte di salita e dati vengono emessi sul fronte di discesa.

Cioè, CPHA = 0 significa campionamento (sample) sul fronte iniziale di commutazione del clock, mentre CPHA = 1 significa campionamento sul fronte finale del clock.

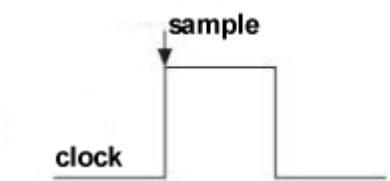
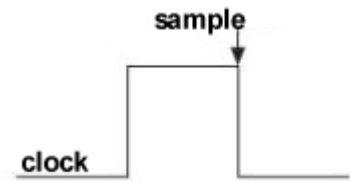
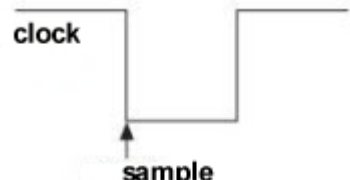
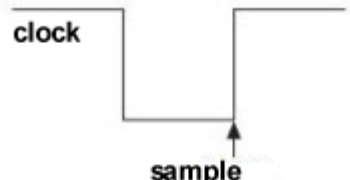
Da notare che, con CPHA = 0, i dati devono essere stabili per mezzo ciclo prima del primo ciclo di clock.

Per tutte le modalità di CPOL e CPHA, il valore iniziale del clock deve essere stabile prima che la linea del chip select sia attiva.

I segnali MOSI e MISO sono solitamente stabili (nei loro punti di ricezione) per metà ciclo fino al prossimo impulso di clock. Dispositivi di master e slave SPI possono effettuare campionamenti anche in altri punti: questo aggiunge più flessibilità per il canale di comunicazione tra master e slave.

Alcuni componenti utilizzano convenzioni diverse di denominazione: ad esempio, il TI MSP430 utilizza il nome UCCKPL invece di CPOL e la sua UCCKPH è l'inverso del CPHA.

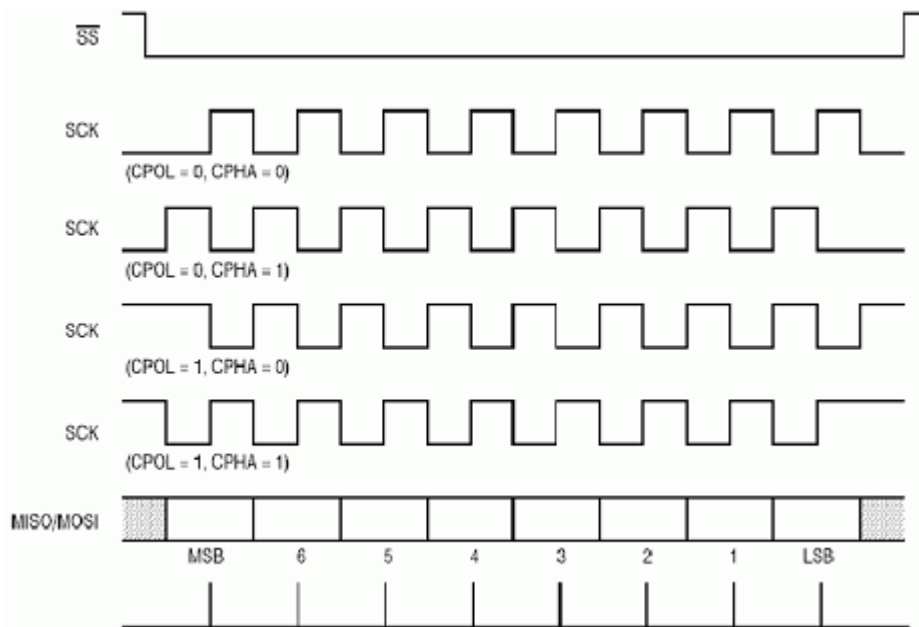
Per i PIC CPOL è CKP (Clock Polarity) e CPHA è CKE (Clock Edge).

SPI Mode	Clock idle	Clock edge	Note
0 - 0,0	0	0	 <p>Clock idle basso, sample sul fronte di salita</p>
1 - 0,1	0	1	 <p>Clock idle basso, sample sul fronte di discesa</p>
2 - 1,0	1	0	 <p>Clock idle basso, sample sul fronte di discesa</p>
3 - 1,1	1	1	 <p>Clock idle basso, sample sul fronte di salita</p>

I modi di uso più comune sono 0,0 e 1,1. Quando si collegano due chip insieme, occorre esaminare attentamente i valori di inizializzazione di fase del clock per essere sicuri di utilizzare le impostazioni corrette.

Questa situazione da origine all' impossibilità di collegare in daisy chain Slave che abbiamo modi diversi: nel concatenamento di Slave tutti dovranno poter operare nello stesso modo.

Nel caso invece di connessione parallela, il Master si potrà configurare in un modo per colloquiare con il primo Slave e in un altro modo per il secondo e così via; questo non comporta problemi agli Slave perchè solo per volta uno viene attivato dal chip select relativo.



Nel diagramma a lato le relazioni temporali tra dati e clock, intesi con il dispositivo selezionato con SS a livello basso.

Osserviamo che la trasmissione avviene **inviando per primo il bit più significativo** (MSB - bit 7 di un byte) e per ultimo quello meno significativo (LSB - bit 0).

SPI non ha un meccanismo di riconoscimento per confermare la ricezione di dati, dato che non ha un protocollo di comunicazione particolare; così il master SPI non è in grado di rilevare se lo Slave chiamato è efficiente, a meno di leggere un dato in risposta sulla linea SDO.

Inoltre SPI non offre alcun controllo di flusso, dato che non ha segnali, nè modi di handshake; dove questo sia necessario, occorre implementare qualcosa di esterno allo standard.

Gli Slave possono essere pensati come input / output del Master e SPI non specifica un particolare protocollo di livello superiore, come nel caso di presenza di più Master e della conseguente necessità di arbitraggio del bus. Il colloquio su SPI, anche se sono presenti più Slave, è sempre inteso per una singola coppia Master-Slave. E, in generale, un protocollo di livello superiore non è necessario, poichè che solo i dati grezzi vengono scambiati. In altre applicazioni dove sia richiesto un protocollo di livello superiore, sarà necessario implementarlo al di fuori dello standard.

Connessioni SPI

Come ultima considerazione, va detto che, se, ad esempio, RS-232 è stato pensato per collegare dispositivi esterni al sistema, **SPI è pensato essenzialmente per connessioni tra componenti sullo stesso circuito stampato o, al massimo, nella stessa apparecchiatura**, quindi su distanze limitate a centimetri o decine di centimetri: lo standard di SPI infatti non prevede particolari connettori o cavi o terminazioni.

E' ovvio che **dalla lunghezza e dalla qualità delle connessioni** tra Master e Slave dipenderà la massima frequenza di clock applicabile. Per cui, anche se alcune periferiche possono supportare clock di 50 MHz, il limite reale dipenderà dal collegamento.

Va tenuto anche presente che i segnali sul bus SPI sono a livello logico e bufferati solamente dalle possibilità di driving dei pin corrispondenti.

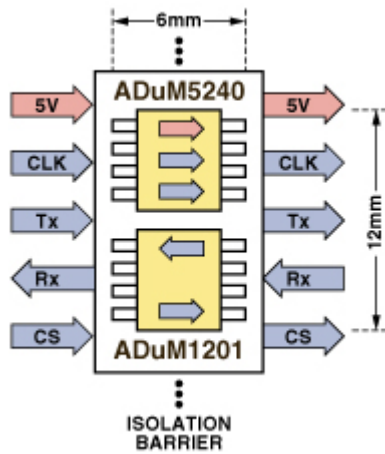
Potrà essere utile, per bus con molti componenti, utilizzare buffer esterni o anche linee open drain/open collector con pull-up, cosa facilmente implementabile dato che le linee di SPI sono tutte mono direzionali.

Va prestata però attenzione perchè **alcuni componenti hanno pull-down integrati** (come il già citato DS1302): in tal caso un pull-up esterno non ha senso.

La mono direzionalità dei segnali SPI consente, dove necessario, lunghe distanze di connessione (molte decine o centinaia di metri) utilizzando ad esempio **buffer differenziali**, realizzabili con comuni componenti per TIA/EIA-422 (RS-422), TIA/EIA-485 (RS-485), EIA/TIA-644 (LVDS) o circuiti di interfaccia compatibili con lo standard TIA/EIA-899 (M-LVDS).

La scelta di uno piuttosto che dell' altro standard di interfaccia differenziale dipende dall' applicazione poichè ci sono diversi vantaggi con ciascuno. In generale, le parti standard RS-485 o RS-422 hanno un trasferimento massimo di segnale di 30Mbps, uno swing di tensione più grande rispetto ai componenti LVDS e M-LVDS e possono fornire una più ampia gamma di tensione in modo comune. LVDS o M-LVDS producono meno EMI, consumano meno energia e operano con transfer rate maggiori. LVDS o M-LVDS si trovano più spesso nelle applicazioni in cui la velocità è un fattore critico, mentre RS-485 o RS-422 si trovano in applicazioni dove il rumore e la distanza sono maggiori. Con questi buffer, trasferimenti si possono raggiungere **distanze di molte centinaia di metri** tra Master e periferica.

Altrettanto semplice è l' **isolamento galvanico** con optoisolatori o simili.



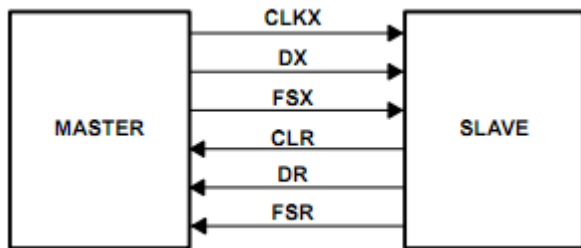
In applicazioni critiche è indispensabile aggiungere un sistema di isolamento completo tra unità a microcontroller e periferiche remote: questo evita i potenzialmente dannosi loop di terra, i rischi per l'operatore e i disturbi raccolti dalle linee.

Un esempio è rappresentato da un componente specifico di [Analog Devices](#).

Si tratta di un integrato della serie ISO-POWER in grado di isolare galvanicamente tutti i segnali del bus SPI, con un ingombro veramente minimo sul circuito stampato.

Altre varianti

Una variante di SPI è costituita dall' interfaccia **McBSP** (*Multichannel Buffered Serial Port*) di Texas Instruments, una comunicazione seriale sincrona analoga a I²S.



Il collegamento Master-Slave è illustrato nella figura a fianco.

L'interfaccia è composta da 6 linee: clock (CLKX), trasmissione dati (DX), trasmissione di sincronizzazione frame (FSX), clock reso (CLKR), dati resi (DR) e sincronizzazione frame resa (FSR).
Le funzioni di McBSP sono simili a SPI.

Entrambe le interfacce forniscono dati seriali (DX, DR) con un clock associato (CLKX, CLKR).

A differenza della SPI, che utilizza un singolo clock, il McBSP riceve i dati con un clock separato (CLKR) e seleziona la linea (FSR).

Le convenzioni di denominazione per McBSP sono diverse da quelle di SPI, ma si possono raffrontare.

Nella tabella , le corrispondenze tra SPI e McBSP.

SPI	McBSP
SCLK	CLKX
MOSI	DX
SS	FSX
-	CLKR
MISO	DR
-	FSR

I²S (*Integrated Interchip Sound*) è un'interfaccia elettrica seriale standard utilizzata per collegare dispositivi audio digitali. È comunemente usato per effettuare trasporto di dati da CD al DAC in un lettore CD o tra AD e DA, DSP, ecc. Il bus **I²S** separa i segnali di clock e dati, con conseguente connessione a jitter molto basso. Questo standard è stato introdotto nel 1986 e ha avuto una ultima revisione nel 1996. Il bus è composto da almeno tre linee, come uno SPI a tre fili .

Una documentazione introduttiva è [disponibile qui](#).

PRO SPI

- Elevata velocità di comunicazione
- Trasmissione full-duplex
- Protocollo con la massima flessibilità
- Trasferimento non limitato a 8 bit
- Struttura hardware semplice e robusta
- Bus a livello logico e basso consumo
- Nessun arbitraggio o speciali protocolli di indirizzamento e handshake
- Clock prodotto da una sola fonte e che non richiede alcuna precisione
- Gli Slave non necessitano di oscillatore
- I segnali sono unidirezionali e si prestano facilmente ad essere bufferati o isolati

CONTRO SPI

- Richiede più pin che non I²C, anche nella variante "3-Wire"
- Il sistema a chip select richiede un pin per ogni Slave
- SPI è multi Slave, ma non multi-Master
- Mancano handshake sul bus per controllare il flusso dei dati, lo stato delle periferiche e gli errori di trasmissione
- Supporta brevi distanze rispetto a RS-485, LIN o CAN-bus
- Ne esistono molte variazioni, rendendo a volte difficile il supporto a tali variazioni

SPI vs. I²C

Nell'ambito del trasferimento dati su corte distanze, **SPI** ha come diretto concorrente il bus **I²C**.

Sia SPI che I²C sono standard di trasferimento dati seriali sincroni. Entrambi sono pensati per comunicazioni su brevi distanze (stesso cs o apparecchiatura), tra master e slave a cui sono fatti accessi intermittenti, come ad esempio EEPROM e RTC.

SPI non è un protocollo strettamente definito e non integra meccanismi di handshake o priorità.

Da questo punto di vista I²C è molto più "protocollo", ovvero definisce meccanismi di handshake e di multi-Master.

Ne risulta che **SPI è decisamente più semplice**, tanto da poter essere emulato via software, cosa molto più complessa per I²C.

Se entrambi hanno la possibilità di gestire più periferiche, quelle di SPI sono solamente degli Slave e non hanno alcun potere sulla linea del clock, mentre quelle di I²C sono più complesse, essendo sia la linea dati che quella del clock bi-direzionali.

In questo senso SPI è decisamente migliore in fatto di semplicità di gestione e di velocità di trasferimento dati, principalmente in applicazioni "stream", come una comunicazione tra microprocessori o microcontroller e periferiche complesse, dato che la trasmissione e la ricezione, usando due linee separate, possono essere contemporanee (full duplex). Ad esempio tra codec e dsp, dove è richiesto un flusso simultaneo di dati bi-direzionali, sono dotati spesso di interfaccia SPI, mentre in I²C la bi-direzionalità della linea dati costringe ad un half-duplex.

Interfacce SPI compatibili possono operare con clock di 10 MHz (ed altre arrivano a 50 MHz), contro i circa 4 di I²C. Questo è ulteriormente a vantaggio del bus SPI. In questo senso SPI dovrebbe garantire un transfer rate almeno doppio di I²C.

Quando si comunica con più di una periferica, **I²C** ha, però, il vantaggio di selezionarla con l'invio di **un indirizzo sulla linea dati**, il che minimizza il numero degli I/O necessari e rende molto più elastica la gestione di un "bus"; per contro, la necessità di inviare l'indirizzo rallenta la comunicazione, anche se il sistema di handshake proprio di I²C consente al Master di rilevare se la periferica chiamata è operativa (cosa non possibile con SPI). In questo senso I²C è più indicato per comunicazioni in sistemi con varie periferiche e in cui il colloquio comporta flussi di dati più lunghi che non semplici blocchi o parole, ma quando questo è sufficiente in un half-duplex.

Inoltre **I²C dispone di un protocollo di handshake** per la correzione di errori di trasmissione, anche se SPI può modificare la frequenza di clock adattandola alle condizioni del mezzo di trasmissione e della periferica collegata.

SPI non è limitato a blocchi dati da 8 bit, così che è possibile inviare un messaggio con dimensioni arbitrarie, in sostanza, della grandezza dello shift register. Questo da una grande flessibilità con periferiche usano dati a più di 8 bit I²C codifica blocchi di 8 o 9 bit, il che è meno funzionale per periferiche che scambiano dati di 10, 12, 14 o più bit. Per contro, richiede modifiche agli algoritmi di controllo in sistemi in cui sono presenti Slave con parole di lunghezza diverse.

Inoltre le periferiche SPI possono utilizzare polarità e fronti del clock differenti, per cui è necessaria una accurata verifica delle caratteristiche dello Slave prima di inserirlo in un progetto.

L'interfaccia SPI, di base, non richiede resistori pull-up, che si traduce in minore consumo di energia. I²C necessita di pull-up, ma è più semplice, avendo un minor numero di linee, il che significa che un minor numero di pin sono impegnati dall'interfaccia. Per contro le linee sono bi-direzionali e diventa più complesso isolarle o bufferarle quando si voglia fare assumere loro lunghezze a cui i semplici segnali a livello logico riceverebbero una degradazione sensibile.

I dispositivi sul bus SPI, comunicando master-slave, in catene daisy chain di slave richiedono solitamente tante linee di chip select quante sono le periferiche. Per contro, i dispositivi di I²C operano con indirizzi. Ne risulta che un sistema SPI con molte periferiche richiederà più risorse hardware, mentre un sistema I²C richiederà una maggiore complessità software, dato che implementa un sistema di handshake.

SPI prevede un solo Master e, per le ragioni sopra elencate, è molto più efficiente di I²C per una situazione Master-Slave, ma lo diventa meno quanto più si aumentano gli Slave, anche se la gestione degli indirizzi penalizza il transfer rate.

Per riassumere:

	SPI	I ² C	Note
Topografia	master + slave	multi master + slave	Multi master possibile anche in SPI, ma fuori dallo standard e non semplice da implementare
Conduttori	3-4 (+n)	2	SPI richiede 1 conduttore in più per ogni chip select di periferica
Trasmissione	full duplex	half duplex	SPI usa linee separate per trasmissione e ricezione e consente clock di frequenza
Velocità di	alta	medio-bassa	

trasmissione			maggiore
Periferiche utili	limitate	molte	In SPI la quantità di Slave collegabili dipende dalla disponibilità di I/O per i chip select
Selezione delle periferiche	chip select	indirizzi	
Protocollo	semplice	complesso	I ² C utilizza pacchetti con priorità, sistemi di arbitraggio del bus, indirizzi di selezione delle periferiche, ecc.
Emulabile software	facilmente	non semplice	SPI non ha un protocollo vero e proprio e si presta bene al bit-banging di dati anche su microcontroller privi di moduli dedicati alla comunicazione sincrona
Gestione del clock	solo dal Master	da ogni dispositivo	La linea di clock di I ² C viene gestita dal dispositivo sul bus che sta trasmettendo. Questo richiede molta più precisione che non il sistema implementato da SPI.
Buffering delle linee	semplice	complesso	SPI ha linee mono direzionali. I ² C ha linee bi direzionali
Isolamento galvanico			

In sostanza, si potrebbe concludere che **SPI** sia decisamente migliore:

- nel caso in cui ci si trovi con **un solo Master e un numero limitato di Slave** (il che, in genere, è la situazione tipica di un sistema a microcontroller)
- quando si vuole **la massima velocità di trasmissione**
- quando non si dispone di modulo di comunicazione sincrona e **si emula la connessione via software**

I²C sarà necessario:

- in sistemi **multi-Master**
- in sistemi con **molte unità collegate** allo stesso bus
- e, in genere, dove esiste integrato nel microcontroller **un modulo di gestione della comunicazione seriale sincrona**, dato che una emulazione completa di I²C via software non è semplice.

Va osservato che un tipico **modulo di comunicazione seriale sincrona** integrato in un microcontroller non è una opzione trascurabile, ma un elemento di grande utilità nella gestione delle comunicazioni sincrone; seppure esse siano emulabili da software, il modulo dedicato consente operazioni più semplici, gestioni sia in polling che in interrupt (indispensabili queste per applicazioni ad alta velocità o con più task attive) oltre alla possibilità di modificare con semplici comandi il modo di funzionamento e ottenere risultati che sarebbero assai difficili da ottenere da una emulazione.

Questa utilità è più evidente per I²C, che presenta maggiori problematiche a causa del protocollo più complesso, ma anche in SPI offre soluzioni che una emulazione sarebbero quanto mai complicate, come il full duplex.

Si deve considerare però, che un modulo può gestire un solo modo di comunicazione per volta, dato che, in generale, fa capo agli stessi pin di I/O e che i due bus non sono compatibili tra di loro.

Quindi, dovendo disporre di una comunicazione I²C e SPI contemporaneamente, occorrerà scegliere un microcontroller con due o più moduli integrati.

Anche se potrà essere fattibile a volte utilizzare il modulo integrato per I²C ed emulare SPI via software.

In tutti i casi, la scelta di uno o dell' altro dipenderà in modo significativo dalle periferiche che si devono collegare, anche se attualmente il mercato offre una scelta molto ampia per cui si trovano sia in SPI che in I²C molte funzioni analoghe