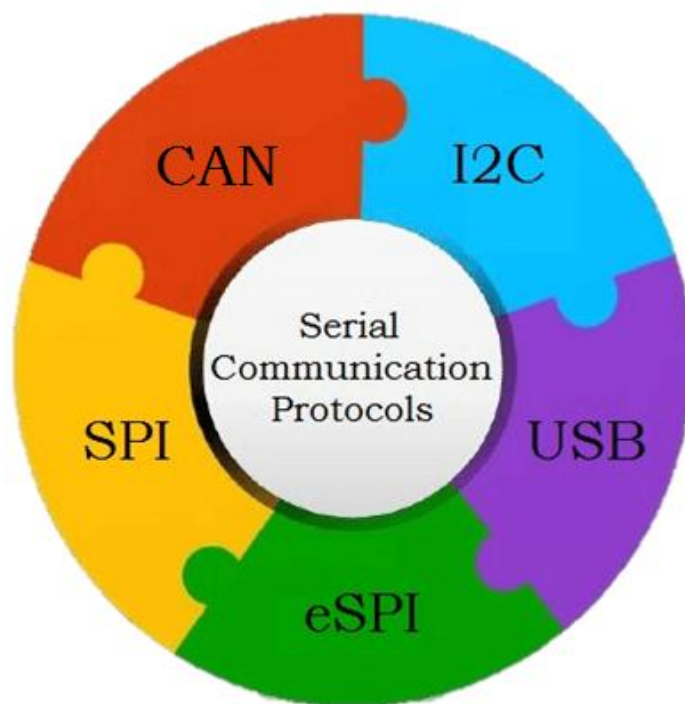


# Guerino Mangiamele

## I protocolli di comunicazione seriale nei sistemi a Microcontrollore



### Sommario

|                                                               |    |
|---------------------------------------------------------------|----|
| Introduzione .....                                            | 2  |
| I principali protocolli di comunicazione seriale .....        | 3  |
| Interfaccia seriale sincrona.....                             | 4  |
| Interfaccia seriale asincrona.....                            | 5  |
| I Principali Protocolli seriali sincroni .....                | 7  |
| Il Protocollo SPI.....                                        | 8  |
| La comunicazione seriale del Protocollo I <sup>2</sup> C..... | 11 |
| Il protocollo USB .....                                       | 12 |
| Il protocollo CAN.....                                        | 13 |
| I Protocolli seriali asincroni .....                          | 14 |
| Il protocollo asincrono RS232 .....                           | 15 |
| Il protocollo asincrono RS485 .....                           | 15 |
| Il protocollo 1-Wire.....                                     | 16 |
| Tabella Comparativa tra tipologie di protocolli seriali.....  | 17 |

## Introduzione

I sistemi a microprocessore o a microcontrollore comunicano con altre periferiche per mezzo di protocolli e interfacce parallele o seriali.

### Definizioni

#### COMUNICAZIONE

*Per Comunicazione si intende lo scambio di dati e informazioni tra due o più mezzi. Nei sistemi digitali, lo scambio di dati avviene sotto forma di bit*

#### PROTOCOLLO DI COMUNICAZIONE

*Nell'ambito della comunicazione dati di tipo digitale tra dispositivi elettronici, si definisce un PROTOCOLLO un insieme di regole prestabilite per lo scambio dei dati (velocità di trasmissione, quantità di bit scambiati, presenza di bit o byte di inizio e fine trasmissione, livelli di tensione del segnale, sincronismo, ecc.....)*

#### INTERFACCIA DI COMUNICAZIONE

*E' intesa come un collegamento che si realizza tra due sistemi mediante l'uso di dispositivi, circuiti o architetture dedicate, per garantire un corretto scambio dei dati. Sono quindi dispositivi elettronici che provvedono ad adattare i segnali provenienti da diverse periferiche garantendone la compatibilità delle caratteristiche elettriche, logiche e funzionali di ciascuno degli elementi collegati. (tensioni, correnti, informazioni, connettori e modalità operative ammissibili e congrue)*

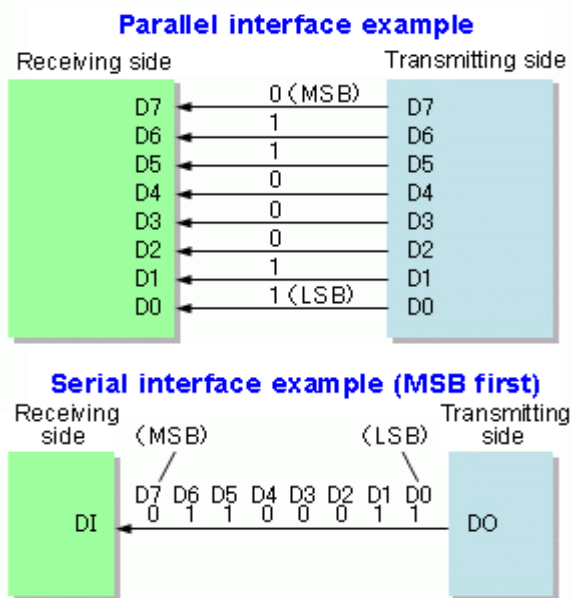
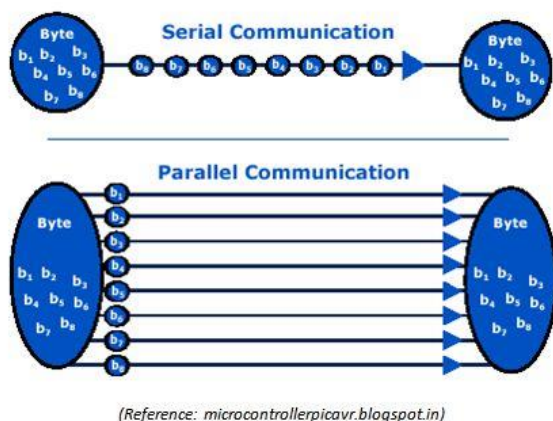
In un sistema di trasmissione dati, ci sarà sempre un dispositivo che trasmette (sender) e uno o più dispositivi che ricevono (Receiver).

Ora, i dati digitali disponibili possono essere inviati o contemporaneamente, secondo una comunicazione di tipo "parallelo" o in modo sequenziale, cioè un bit dopo l'altro. In questo caso il protocollo di comunicazione è noto come protocollo di comunicazione seriale.

La comunicazione seriale è la tecnologia basilare adottata nel trasferimento dei dati digitali. Innumerevoli sono le tipologie di protocolli seriali impiegati :

E' opportuno solo ricordare che tutti i sistemi wireless, dalla trasmissione televisiva digitale al più classico WIFI domestico utilizza la modalità seriale per lo scambio di dati digitali, così come, naturalmente le connessioni internet e quindi qualsiasi forma di streaming video o audio.

Esempi di protocolli di comunicazione parallela sono ISA, ATA, SCSI, PCI e IEEE-488. Allo stesso modo ci sono diversi esempi di protocolli di comunicazione seriale come USB, CAN, ETHERNET, I2C, SPI, RS232, 1-Wire, SATA ecc.



da wikipedia

I sistemi di comunicazione tra periferiche di tipo parallelo sono ormai stati abbandonati da diversi anni, dopo l'introduzione di interfacce di tipo seriale ad alta velocità.

## I principali protocolli di comunicazione seriale

La comunicazione seriale è l'approccio più utilizzato per trasferire informazioni tra periferiche di elaborazione dati. Ogni dispositivo elettronico, sia che si tratti di Personal Computer (PC) o Mobile, funziona su comunicazione seriale. Il protocollo è la forma di comunicazione sicura e affidabile con un insieme di regole indirizzate dall'host di origine (mittente) e dall'host di destinazione (destinatario) simili alla comunicazione parallela.

### Modalità di trasmissione nella comunicazione seriale

Come sappiamo, nella comunicazione seriale i dati vengono inviati sotto forma di bit cioè impulsi binari dove "uno binario" rappresenta la logica ALTA e lo zero rappresenta la logica BASSA. Esistono diversi tipi di comunicazione seriale a seconda del tipo di modalità di trasmissione e trasferimento dati. Le modalità di trasmissione sono classificate come

- Simplex
- Half Duplex
- Full Duplex.

### Trasmissione Simplex o Unidirezionale

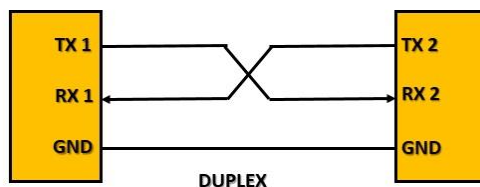
Nel metodo se il mittente sta trasmettendo i dati, il destinatario può solo accettare e viceversa. Quindi il metodo simplex è una tecnica di comunicazione unidirezionale. Noti esempi sono la televisione e la radio.

### Trasmissione Half Duplex:

Nel metodo half duplex sia mittente che destinatario possono essere attivi ma non contemporaneamente. Quindi, se il mittente sta trasmettendo, il destinatario può accettare ma non può inviare e allo stesso modo viceversa. L'esempio ben noto di half duplex è Internet in cui l'utente invia una richiesta di dati e la ottiene dal server.

### Trasmissione Full Duplex:

Nel metodo full duplex, sia il ricevitore che il trasmettitore possono scambiarsi dati contemporaneamente. L'esempio noto è il telefono cellulare.

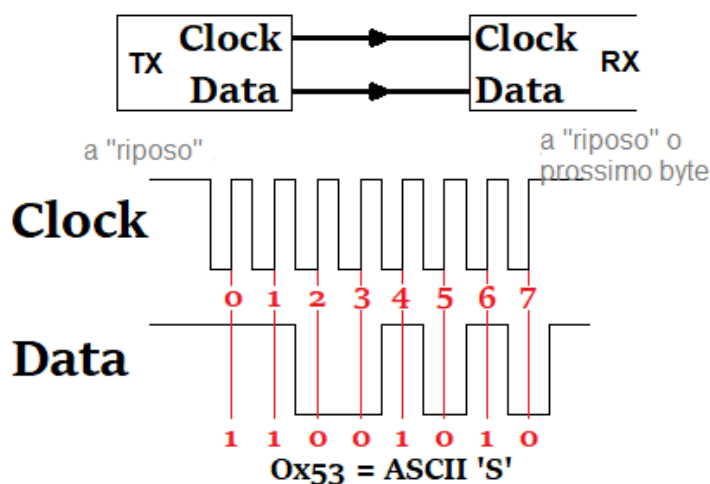


In questo caso, per un'adeguata trasmissione dei dati, il sincronismo fornito dal Clock gioca un ruolo importante. Per evitare anomalie di trasmissione e perdita di dati, i clock presenti sui dispositivi in collegamento devono essere perfettamente sincronizzati.

## Interfaccia seriale sincrona

È una connessione punto a punto da un master a uno slave. In questo tipo di interfaccia, tutti i dispositivi utilizzano un unico bus per condividere sia i dati che il clock.

Vengono condivise due linee : la linea dati seriale e la linea di clock



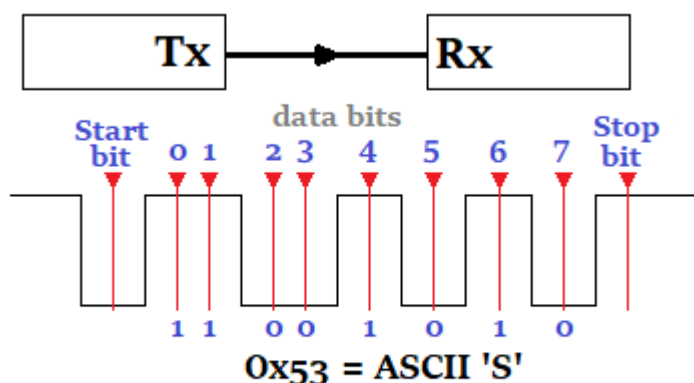
Comunicazione Seriale Sincrona

Il trasmettitore invia i dati, fornendo anche il segnale di clock .

Sul lato ricevitore, i dati vengono estratti utilizzando il clock fornito dal trasmettitore e converte i dati seriali nella forma parallela. Gli esempi ben noti sono I2C e SPI.

*In elettronica, per "Clock" si intende un segnale ad onda quadra (un susseguirsi di uno logico e zero logico) ad una determinata frequenza*

## Interfaccia seriale asincrona



Comunicazione Seriale Asincrona

Nell'interfaccia seriale asincrona, il segnale di clock esterno è assente. Le interfacce seriali asincrone possono essere viste principalmente in applicazioni a lunga distanza e sono perfette per la comunicazione stabile. Nell'interfaccia seriale asincrona, i problemi di derivanti dalla perdita di dati dovuti all'assenza della sorgente di clock esterna viene compensata da diversi parametri come controllo del flusso di dati, controllo degli errori, controllo della velocità di trasmissione, controllo della trasmissione e controllo della ricezione. La velocità di trasmissione dei dati è generalmente inferiore alla trasmissione sincrona.

Sul lato trasmettitore, c'è uno spostamento di dati paralleli sulla linea seriale utilizzando il segnale di clock interno. Viene aggiunto i bit di inizio, fine e controllo di parità. Sul lato ricevitore, il ricevitore estrae i dati utilizzando il proprio clock e riconverte i dati seriali nella forma parallela dopo aver rimosso i bit di inizio, arresto e parità. Gli esempi ben noti sono RS-232, RS-422 e RS-485.

Non è necessaria alcuna linea di clock separata nella comunicazione asincrona

Quando si ha a che fare con la trasmissione seriale si incontrano spesso questi termini.

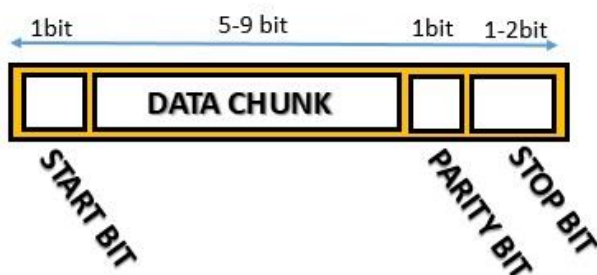
**Baud rate:** la velocità in baud è la velocità alla quale i dati vengono trasferiti tra il trasmettitore e il ricevitore sotto forma di **bit al secondo** (bps). La velocità di trasmissione più comunemente usata è 9600bps. Ma ci sono altre selezioni di velocità di trasmissione come 1200, 2400, 4800, 57600, 115200. Per potersi comprendere correttamente il baud rate deve essere lo stesso tra tutti i dispositivi connessi. Altri termini relativi alla comunicazione seriale

**Framing:** il framing si riferisce al numero di bit di dati da inviare dal trasmettitore al ricevitore. La maggior parte dell'applicazione utilizza 8 bit come bit di dati standard, ma può essere selezionata anche come 5, 6 o 7 bit.

**Sincronizzazione:** i bit di sincronizzazione sono importanti per selezionare un blocco di dati. Indicano l'inizio e la fine dei bit di dati. Il trasmettitore imposterà i bit di inizio e fine al frame di dati e il ricevitore lo identificherà eseguendone l'elaborazione.

### Controllo degli errori nella comunicazione seriale asincrona

Il controllo degli errori gioca un ruolo importante durante la comunicazione seriale in quanto vi sono molti fattori che influenzano e aggiungono il rumore nella comunicazione seriale. Per eliminare questo errore vengono, per esempio utilizzati i bit di parità per la verifica della correttezza dei dati inviati.



Il controllo di parità consiste nell'aggiungere agli n bit di ogni carattere un ulteriore bit (P) detto di parità, in modo da portare il numero di bit di valore 1 ad un valore dispari (parità dispari) o pari

(parità pari). Per esempio, impostando un controllo a “parità pari”, se il numero di bit ad 1 di un byte di dati da inviare è dispari ( per esempio 3 ), il trasmettitore imposterà il registro “bit di parità” a 1 , in modo da inviare un numero di bit ad 1 pari. Se al ricevitore giungono un numero di bit dispari, vorrà dire che è presente un errore nella trasmissione e quindi richiede al trasmettitore il reinvio dei dati .

Questa tecnica non corregge gli errori e non consente di riconoscere se ci sono nello stesso carattere due bit errati (o meglio un numero pari di bit errati).

7

*Esempio :*

*Supponiamo di inviare il byte 01101011 utilizzando il controllo errori a “parità pari”, in questo caso bit inviati sono in numero dispari, quindi il trasmettitore setta ad “1” il bit di parità .*

*Supponiamo che durante la trasmissione avvenga un errore sul quarto bit (lo zero diventa 1 a causa di rumori o interferenze esterne) . Il ricevitore “conta “ il numero di 1 e verifica che sono in numero dispari e non pari come ci si aspettava. Quindi scarta il dato e richiede una nuova trasmissione del byte*

Principali differenze tra trasferimento dati asincrono e sincrono.

Quando si utilizza un protocollo seriale asincrono, il mittente deve fornire un segnale di sincronizzazione prima di trasferire qualsiasi dato. Questo deve essere fatto prima di ogni messaggio trasmesso.

Non è richiesto alcun segnale di clock tra il mittente e il destinatario del messaggio.

È supportata la trasmissione di dati a lunga distanza.

L'affidabilità aumenta con la trasmissione dei dati asincrona.

Quando viene utilizzato il protocollo seriale sincrono, sia il mittente che il destinatario utilizzano lo stesso segnale di clock durante il trasferimento dei dati. Alcune altre caratteristiche della trasmissione sincrona sono:

I protocolli sincroni supportano velocità di trasferimento dati più elevate.

Il segnale dell'orologio deve essere comunicato tra il mittente e il destinatario.

È necessaria una configurazione master / slave per implementare il trasferimento dati sincrono.

## I Principali Protocolli seriali sincroni

Sono sincroni i protocolli seriali quali SPI, I2C, CAN ampiamente utilizzati nella comunicazione tra microcontrollori e tra questi e sensori o schede Front end per sensori ed attuatori.

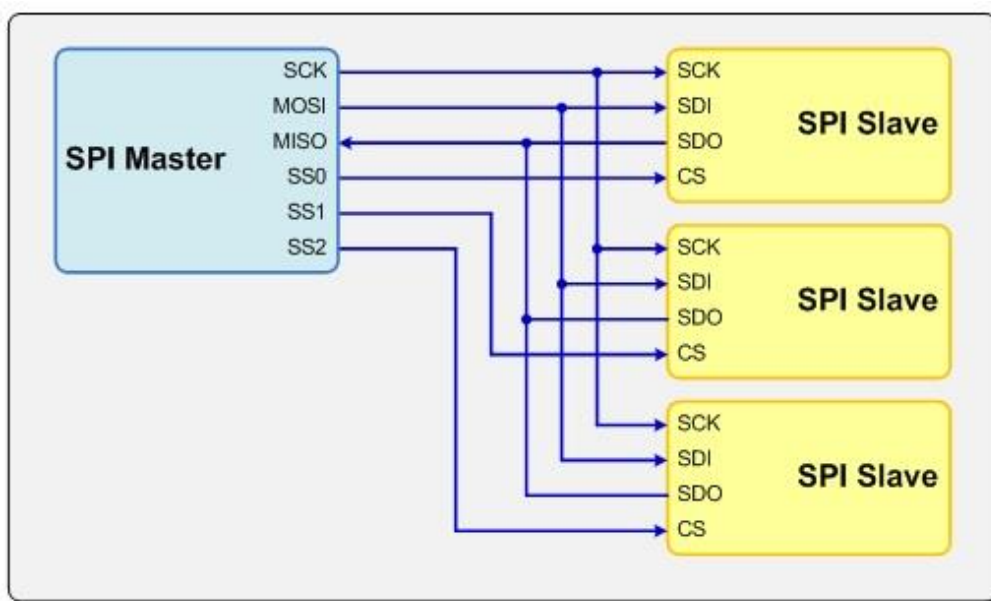
Sono largamente diffusi, anche in microcontrollori di fascia bassa e a basso costo, i protocolli UART, SPI e I2c .

## Il Protocollo SPI

La Serial Peripheral Interface (SPI) è un'interfaccia sincrona che consente l'interconnessione di più microcontrollori SPI.

Ormai, la quasi totalità dei microcontrollori, anche quelli a bassissimo costo possono gestire protocolli SPI.

Nella comunicazione in modalità SPI, sono necessari cavi separati per dati e linea di clock.



Il BUS SPI Architettura Single MASTER Multi SLAVE

La configurazione è di tipo MASTER/SLAVE con un dispositivo a più alta priorità (MASTER) che gestisce lo scambio dati con gli SLAVE.

Sono quattro le linee che gestiscono i segnali SPI di base (MISO, MOSI, SCK e SS), Vcc e Ground fanno parte della comunicazione dei dati.

Quindi necessitano 5 fili ( 6 con Vcc) per inviare e ricevere dati da slave o master.

Teoricamente, l'SPI può avere un numero illimitato di slave. La comunicazione dei dati è configurata nei registri SPI. L'SPI può fornire fino a 60 Mbps di velocità ed è ideale per la comunicazione dati ad alta velocità a corto raggio.

Valori tipici di tensione sono: 1.8V, 3.3V, 3.6V e 5V

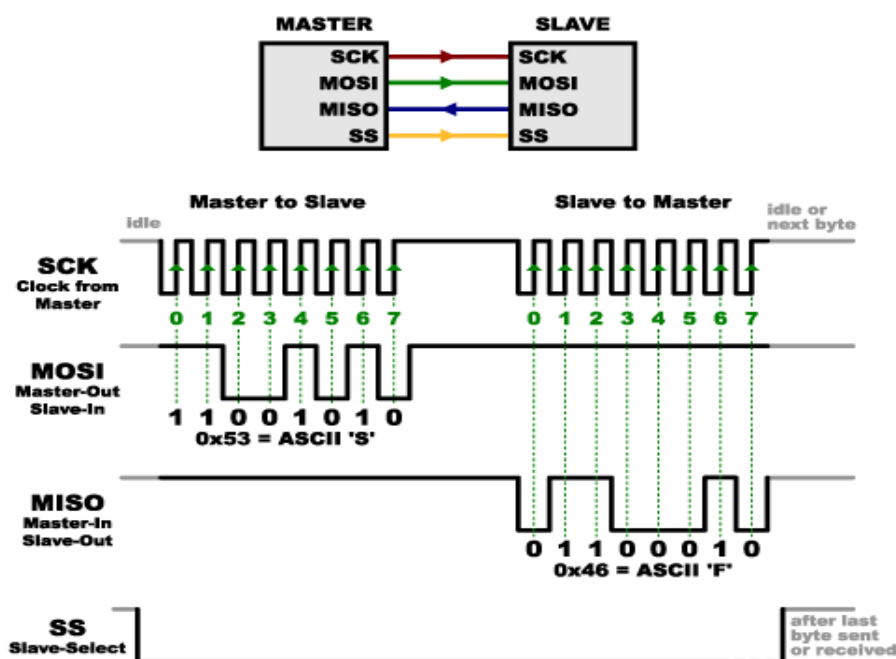


## L'interfaccia SPI

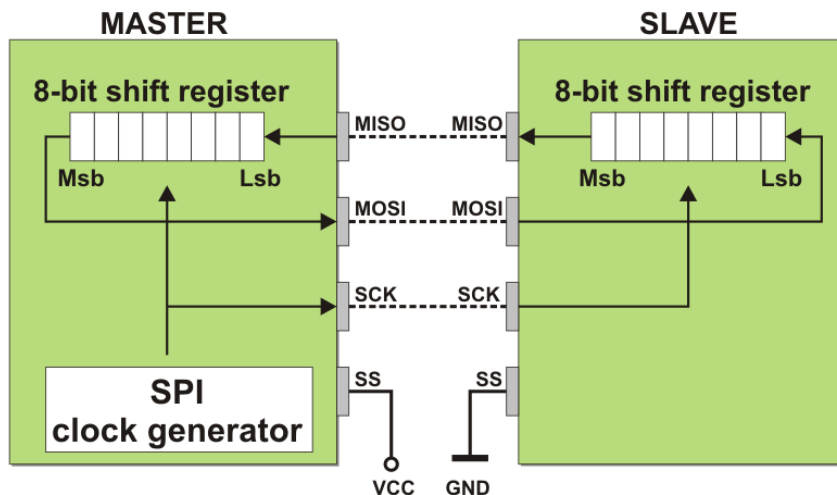
Il BUS è a 4 fili più il collegamento Vref

- **SCLK (SCK)** è il canale del clock, scandisce i tempi in cui master e slave possono trasmettere e ricevere i bit sul bus. Il canale è gestito dal master, che richiede di volta in volta, una "parola a 8 bit"; La comunicazione avviene a 1 o 2 byte
- **MOSI (SIMO)** è il canale di trasmissione del dato dal master allo slave ( Master Output Slave Input )
- **MISO (SOMI)** è il canale di trasmissione del dato dallo Slave al Master (Master Input Slave Output )
- **CS (SS) CHIP SELECT** è il canale attraverso cui il master sceglie con quale dispositivo slave comunicare. ATTIVO BASSO .

Tutte le periferiche Slave sono collegate agli stessi piedini SCK, MOSI e MISO , invece il segnale di CS per ciascun slave viene fornito da una porta settata come OUTPUT del microcontrollore , nella configurazione DAISY CHAIN è sufficiente una sola porta .



Il BUS SPI Diagramma temporale di scambio dati



#### COMUNICAZIONE FULL DUPLEX TRA MASTER E SLAVE

La struttura ad «anello» permette una comunicazione Full Duplex

Generalmente viene inviato per primo il MSB che occupa la posizione LSB dello SR del corrispondente. E' comunque possibile configurare la direzione dei dati.

- A) Il dispositivo master manda «basso» CS per selezionare il corrispondente slave
- B) Invia il segnale di clock su SCLK;
- C) Ad ogni ciclo di clock avviene la comunicazione di un BIT sul canale MOSI ;
- D) Il dispositivo slave riceverà il segnale di clock e il primo bit sul canale MOSI, interpretandolo come l'inizio della comunicazione. In contemporanea invierà un altro dato sul canale MISO

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

Se lo Slave deve trasmettere dati, sarà compito del Master inviare il clock necessario.

Sono state sviluppate altre varianti dell'Interfaccia SPI, che mirano al miglioramento delle caratteristiche di trasmissione dati.

Il protocollo eSPI di Intel permette un collegamento più veloce ed un utilizzo di un minor numero di pin.

([https://www.intel.com/content/dam/support/us/en/documents/software/chipset-software/327432-004\\_espi\\_base\\_specification\\_rev1.0\\_cb.pdf](https://www.intel.com/content/dam/support/us/en/documents/software/chipset-software/327432-004_espi_base_specification_rev1.0_cb.pdf) )

Una ulteriore variante dell'Interfaccia SPI è quella definita dal protocollo MICROWIRE (seriale a 3 fili full-duplex) con velocità di trasmissione fino a 3Mbps, a volte usata per scambio dati con memorie EEPROM e altri chip periferici.

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.

Vantaggi dell'interfaccia SPI

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

11

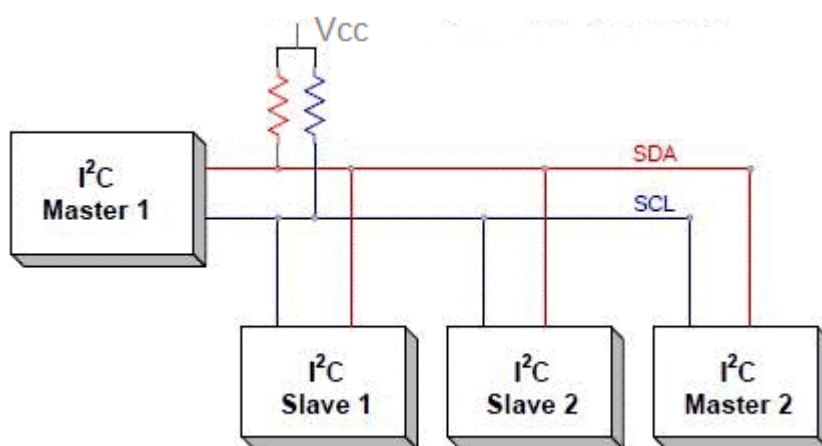
Svantaggi dell'interfaccia SPI

- Utilizzo di un grande numero di linee.
- Mancanza di un protocollo e conferma di trasmissione
- Mancanza d'indirizzamento delle periferiche (escluso segnale SS)

Per approfondire ...

- <https://www.makeritalia.org/tutorial/2018/05/08/arduino-spi/>
- <https://www.laurtec.it/tutorial/elettronica-digitale/129-interfaccia-spi>

## La comunicazione seriale del Protocollo I<sup>2</sup>C



Comunicazione Seriale I<sup>2</sup>C

Il protocollo di comunicazione IIC ( Inter Integrated Circuit), noto come I<sup>2</sup>C, è un protocollo di comunicazione **seriale sincrono** .

Creato da Philips, permette ai dispositivi dotati di tale interfaccia di scambiarsi dati fra loro utilizzando solo due fili (oltre a Vcc e GND), uno per i dati ed uno per sincronizzare la comunicazione.

**SCL Serial Clock Line**      Clock per la sincronizzazione della comunicazione  
**SDA Serial Data Line**      Linea su cui transitano i dati

Realizzare un bus I<sup>2</sup>C è molto semplice, sono sufficienti due resistenze di pull-up collegate a Vref, una per ciascuno dei due segnali SDA ed SCL.

Lo standard per la velocità di clock del bus è di 10 KHz e di 100 KHz, ma si arriva anche a 3.4 MHz .

Possono essere collegati fino a 128 o 1024 devices, con indirizzamento a 7 bit o a 10 bit con tecnica Multi master / Multi Slave.

Contrariamente al SPI, richiede un indirizzo

Permette comunicazioni a distanze limitate ed è ideale per la comunicazione tra periferiche a bordo della stessa scheda. Come l'SPI, i microcontrollori dispongono generalmente di pin I2C per collegare qualsiasi dispositivo I2C:

*Per approfondire...*

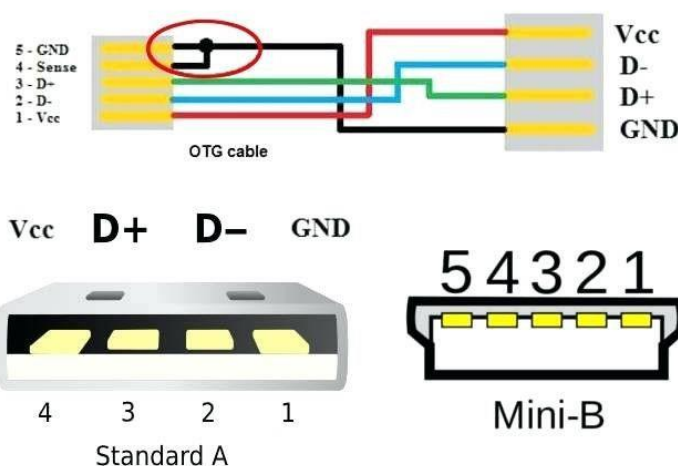
- [Come utilizzare I2C in Arduino: comunicazione tra due schede Arduino](#)
- <https://resources.altium.com/it/p/spi-versus-i2c-how-choose-best-protocol-your-memory-chips>

## Il protocollo USB

USB (Universal Serial Bus) è ampiamente il protocollo più utilizzato con diverse versioni e velocità. È possibile collegare un massimo di 127 periferiche a un singolo controller host USB. USB funge da dispositivo "plug and play".

L'interfaccia USB è ormai universalmente utilizzata in dispositivi come tastiere, stampanti, dispositivi multimediali, fotocamere, scanner e mouse.

È progettato per una facile installazione, una classificazione dei dati più rapida, meno cavi e supporta la funzione "hot swap" per la connessione e disconnessione rapida. Ha sostituito le porte seriali e parallele più voluminose e lente. Per la comunicazione dati utilizza segnali elettrici differenziali per ridurre le interferenze e consentire la trasmissione ad alta velocità su lunghe distanze.



Un bus differenziale è costruito con due fili, uno rappresenta i dati trasmessi e l'altro il suo complemento. L'idea è che la tensione "media" sui fili non trasporta alcuna informazione, con conseguente minore interferenza. Sul connettore USB sono anche presenti la Vcc e GND per l'alimentazione delle periferiche. Le versioni USB supportano velocità diverse come 1,5 Mbps (USB v1.0), 480 Mbps (USB2.0), 5 Gbps (USB v3.0).

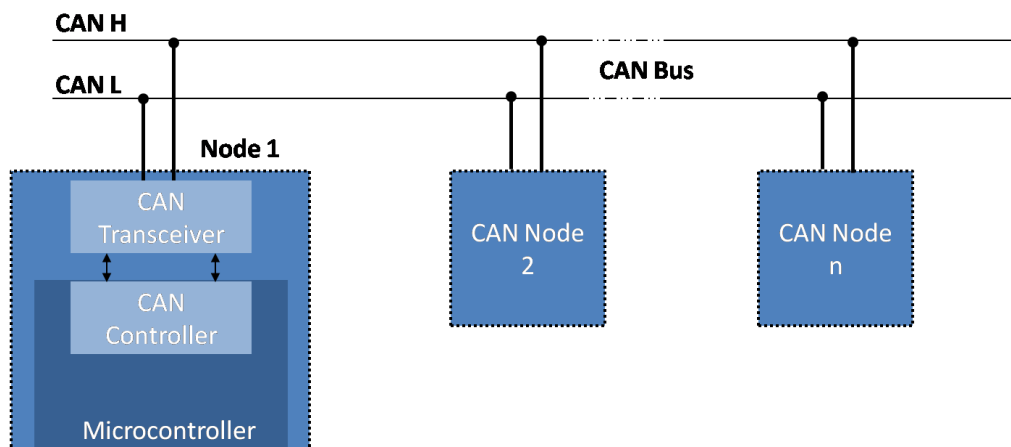
La versione USB 4.0 permette velocità sino a **40Gbps**, con velocità in condizioni standard di almeno 10 Gbps. La lunghezza del cavo USB può raggiungere i 5 metri senza hub e 40 metri con hub.

Per approfondire...

- [Comunicazione seriale con Arduino](#)
- [USB Wikipedia](#)

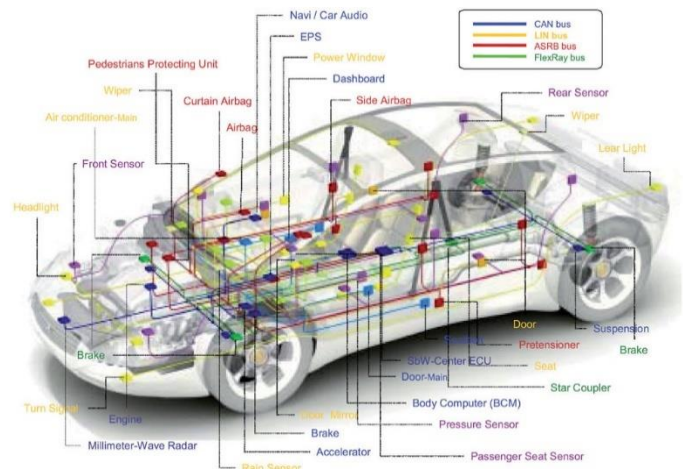
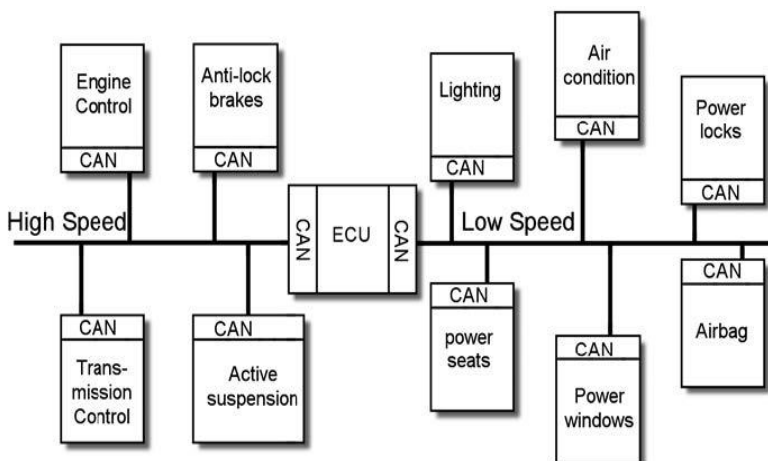
## Il protocollo CAN

La Controller Area Network (CAN) viene utilizzata in ambito Industriale ed è ormai lo standard di riferimento in applicazioni automotive per lo scambio dati tra macchine utensili o tra i tanti dispositivi elettronici ECU ed i sensori presenti nelle moderne automobili.



Il protocollo CAN (detto anche CANBus), essenzialmente costituito da due fili a cui vengono connessi i nodi CAN è robusto, a basso costo e basato su messaggi e specifici codici ID di indirizzamento e identificazione. Il sistema CAN bus consente la diagnosi e la configurazione centralizzata degli errori su tutte le ECU.

I messaggi CAN sono prioritari tramite ID in modo che gli ID con la priorità più alta non vengano interrotti. Ogni ECU contiene un chip per ricevere tutti i messaggi trasmessi, decidere la rilevanza e agire di conseguenza - ciò consente una facile modifica e l'inclusione di nodi aggiuntivi (ad esempio, data logger CAN bus). I sistemi bus CAN possono fornire velocità fino a **1 Mbps**.



Per approfondire...

- [Quaderni di Automazione il CANbus in automobile](#)
- [Esempi di Applicazioni Open Source in ambito automotive](#)

## I Protocolli seriali asincroni

Sono molto impiegati quando si tratta di realizzare un trasferimento dati affidabile su lunghe distanze. La comunicazione asincrona non richiede un segnale di clock sincronizzato su entrambi i dispositivi. Ogni dispositivo ascolta e invia

indipendentemente impulsi digitali che rappresentano bit di dati a una velocità concordata.

La comunicazione seriale asincrona viene talvolta definita seriale Transistor-Transistor Logic (TTL), dove il livello alto di tensione è il logico 1 e il livello basso equivale a 0 logico.

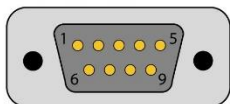
Quasi ogni microcontrollore presente sul mercato ha almeno un Ricevitore Trasmettitore Asincrono Universale (**UART**) per comunicazione seriale. Esempi tipici sono i protocolli RS232, RS422, RS485 ecc.

15

## Il protocollo asincrono RS232

L'RS232 (standard consigliato 232) è un protocollo molto comune utilizzato per collegare diverse periferiche come monitor, strumenti di misura, CNC, ecc. L'RS232 è disponibile in connettori maschio e femmina.

DB9M Connector



RS232 Pin Out

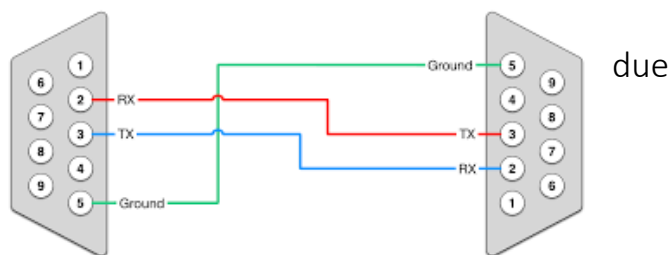
| Pin # | Signal |
|-------|--------|
| 1     | DCD    |
| 2     | RX     |
| 3     | TX     |
| 4     | DTR    |
| 5     | GND    |
| 6     | DSR    |
| 7     | RTS    |
| 8     | CTS    |
| 9     | RI     |

L'RS232 è una topologia punto-punto (peer-to-peer) con massimo un dispositivo collegato e copre distanze fino a 15 metri a 9600 bps; quindi, contrariamente ai bus, sulla linea dati possono essere collegati solo 2 dispositivi, con funzione di trasmettitore o ricevitore. Le informazioni sull'interfaccia RS-232 vengono trasmesse digitalmente da 0 e 1 logici.

Il logico "1" (MARK) corrisponde a una tensione nell'intervallo da -3 a -15 V. Lo "0" logico (SPAZIO) corrisponde a una tensione nell'intervallo da +3 a +15 V.

Lo standard prevede l'utilizzo di connettori di tipo DB9 a 9 pin (Tx, Rx, RTS, CTS, DTR, DSR, DCD, GND).

Nei collegamenti di periferiche con microcontrollori, vengono solo utilizzati fili TX e RX oltre al riferimento di GND.



## Il protocollo asincrono RS485

Poiché RS485 utilizza la topologia multipunto, è più utilizzato nei settori industriali e laddove è necessario uno scambio dati tra periferiche e macchine utensili poste a lunga distanza. Come il protocollo RS422 può collegare 32 driver di linea e 32 ricevitori in configurazioni differenziali ma con l'aiuto di ripetitori aggiuntivi e amplificatori di segnale può permettere la connessione contemporanea fino a 256 dispositivi. L'RS-485 non ha uno specifico connettore standard, ma spesso le schede RS485 vengono

commercializzate con una semplice morsettiera o un connettore DB9. La velocità di funzionamento dipende anche dalla lunghezza della linea e può raggiungere i 10 Mbit /s a 10 metri. La tensione sulle linee è nell'intervallo da -7 V a +12 V. Esistono due tipi di RS-485 come la modalità half duplex RS-485 con 2 contatti e la modalità full duplex RS-485 con 4 contatti.

Per approfondire...

- <https://it.emcelettronica.com/la-comunicazione-dati-in-rs-485>
- [Comunicazione dati MODBUS Seriale RS-485 con Arduino UNO come Slave](#)
- [Esempio di Comunicazione RS-485 tra Raspberry Pi e Arduino Uno](#)

## Il protocollo 1-Wire

Il protocollo 1-Wire introdotto dalla Dallas Semiconductors è di tipo master/slave asincrono. Come il bus I2C è di tipo half-duplex con comunicazione su di un solo filo con data transfer di tipo bidirezionale. Dallo stesso filo è possibile alimentare il dispositivo slave, pur con assorbimento ridotto. La velocità massima è di **16 Kbps**, ma possono essere collegate periferiche anche a qualche migliaio di metri con l'uso di appropriate resistenze di pull-up. Viene utilizzato per il collegamento a sensori e dispositivi elettronici dislocati all'aperto e spesso in siti non presidiati.

Per approfondire...

[1-Wire Design Guide](#)



## Tabella Comparativa tra tipologie di protocolli seriali

| <b>Table 1 Protocol comparison</b> |                    |              |               |                    |                         |                            |                     |
|------------------------------------|--------------------|--------------|---------------|--------------------|-------------------------|----------------------------|---------------------|
| <b>Name</b>                        | <b>Sync /Async</b> | <b>Type</b>  | <b>Duplex</b> | <b>Max devices</b> | <b>Max speed (Kbps)</b> | <b>Max distance (Kbps)</b> | <b>Pin count(1)</b> |
| <b>RS-232</b>                      | async              | peer         | full          | 2                  | 20(2)                   | 30(3)                      | 2(4)                |
| <b>RS-422</b>                      | async              | multi-drop   | half          | 10(5)              | 10,000                  | 4,000                      | 1(6)                |
| <b>RS-485</b>                      | async              | multi-point  | half          | 32(5)              | 10,000                  | 4,000                      | 2                   |
| <b>I<sup>2</sup> C</b>             | sync               | multi-master | half          | -7                 | 3,400                   | <>                         | 2                   |
| <b>SPI</b>                         | sync               | multi-master | full          | -7                 | >1,000                  | <>                         | 3+1(8)              |
| <b>Microwire</b>                   | sync               | master/slave | full          | -7                 | >625                    | <>                         | 3+1(8)              |
| <b>1-Wire</b>                      | async              | master/slave | half          | -7                 | 16                      | 1,000                      | 1s                  |