

Bus I²C Specifiche

Da: The I²C BUS SPECIFICATION

Ver: 2.1

Data: Gennaio 2000

Ogg.: Traduzione per promemoria

1 Premessa

Questo documento non vuole essere una vera e propria traduzione del documento originale, ma bensì un promemoria per l'utilizzo futuro del bus I²C nei sistemi a microcontrollore.

Il bus I²C nasce dalle esigenze interne della PHILIPS Semiconductors, che dapprima lo utilizza nei suoi apparati poi, vista l'efficienza e la versatilità decide di renderlo di pubblico dominio, rimanendo Philips stessa gestore dello standard, di fatti per l'assegnazione dell'indirizzo ad un dispositivo bisogna far richiesta a Philips stessa.

2 Benefits

Nell'elettronica di consumo, nelle telecomunicazioni e nell'elettronica industriale, ci sono spesso molte similitudini tra i vari progetti. Per esempio quasi tutti i sistemi hanno:

- Intelligenza di controllo, generalmente un microcontrollore
- Circuiti quali LCD driver, porte di I/O remote, RAM, EEPROM, o convertitori

La similitudine tra questi settori ha indotto Philips a sviluppare un semplice bus bi direzionale a 2 fili, inter IC bus o I²C bus.

Tutti i dispositivi I²C compatibili incorporano un'interfaccia interna che permette loro di comunicare con ognuno attraverso il bus I²C. Questa filosofia di progetto ha risolto molti problemi d'interfacciamento incontrati durante lo sviluppo di sistemi di controllo digitali.

Di seguito alcune delle caratteristiche salienti del bus I²C:

- Presenza di solo due linee, linea dati seriale (SDA) e linea clock seriale (SCL)
- Ogni dispositivo connesso al bus è indirizzabile via software attraverso un unico indirizzo persiste una semplice relazione master/slave; il master può operare come master trasmettitore o come master ricevitore
- E' realmente un bus multi master incluso il rilevamento di collisione e l'arbitraggio per prevenire la corruzione dei dati qualora due o più master simultaneamente iniziano un trasferimento di dati
- Seriale, 8 bit, bi direzionale, supporta tre diverse tipologie 100Kbit/s nella modalità Standard , 400Kbit/s nella modalità Fast e 3.4Mbit/s in modo HIGH
- Ogni dispositivo possiede un filtro contro gli spikes sul bus al fine di conservare l'integrità dei dati
- Il numero dei dispositivi collegabili al bus è limitato alla sola capacità totale, che non deve superare i 400pF

2.1 Benefit in fase di progetto

Con l'utilizzo di dispositivi compatibili I²C si passa dalla fase di funzionalità a blocchi al prototipo. Inoltre non avendo i dispositivi necessità di circuiti d'interfacciamento con il bus questo consente l'implementazione del sistema in modo rapido ed efficiente.

Di seguito alcune delle caratteristiche salienti dei dispositivi e del bus I²C in fase progettuale:

- Riduzione del tempo dalla fase di funzionalità a blocchi alla fase di stesura dello schema, poiché le funzioni dei dispositivi corrispondono nelle due fasi.
- Nessun tipo di interfacciamento con il bus essendo i dispositivi stessi già provvisti d'interfaccia.
- L'indirizzo predefinito e il protocollo di trasferimento dati consentono al sistema di essere completamente definibile via software.
- Lo stesso dispositivo può essere usato in svariate applicazioni.
- Il tempo di progetto viene ridotto una volta acquisita familiarità con il bus e i vari dispositivi.
- Tutti i dispositivi possono essere aggiunti o tolti dal bus senza alterare le funzionalità del bus stesso.
- Diagnosi e debug semplici, malfunzionamenti facilmente tracciabili.
- Sviluppo software ridotto assemblando i moduli in librerie riutilizzabili.

In aggiunta a queste caratteristiche questi dispositivi offrono anche alcuni vantaggi per lo sviluppo di applicazioni portatili e a batteria, i dispositivi sono tutti:

- Estremamente a basso consumo
- Alta immunità ai disturbi
- Ampio spettro di funzionamento per l'alimentazione e il range di temperatura.

2.2 Benefit in fase di realizzazione

Il bus I²C non offre solo vantaggi in fase di sviluppo ma anche in fase di realizzazione, poiché:

- La presenza di soli due fili minimizza le connessioni e le piste sul PCB, realizzando dei PCB economici.
- La completa integrazione del bus elimina la necessità di dispositivi di decodifica degli indirizzi.
- La capacità di funzionamento in modalità multi master consente un rapido testing e un allineamento con i dispositivi finali.
- La disponibilità di dispositivi in diversi contenitori consentono, là dove richiesto di minimizzare l'ingombro del sistema.

Questi sono solo alcuni dei vantaggi del bus. In aggiunta si può considerare che un dispositivo compatibile I²C consente l'incremento del sistema e la sua espandibilità in modo flessibile e rapido. In questo modo intere famiglie di dispositivi possono essere sviluppati intorno ad un modulo base.

3 Introduzione alle specifiche del bus I²C

Per applicazioni che richiedono un controllo a microcontrollore, si possono definire alcuni criteri di progetto:

- Un sistema completo necessita di un microcontrollore, dei dispositivi di periferica, come memorie e espansioni di I/O.
- Il costo delle connessioni tra i vari dispositivi deve essere minimizzato.
- Un sistema con performance di controllo non richiede un alta velocità di trasferimento dati.
- Soprattutto l'efficienza del sistema dipende dai dispositivi selezionati e dalla natura della struttura del bus utilizzato.

Per procedere alla soddisfazione di questi criteri di progetto, abbiamo bisogno di un bus seriale. Sebbene un bus seriale non ha la stessa capacità di un bus parallelo, noi abbiamo necessità di ridurre il numero di connessioni (fili e pin IC). Comunque un bus non è inteso solo come interconnessioni di fili, comprende anche il formato e la procedura di comunicazione con il sistema.

La comunicazione tra dispositivi su bus seriale deve avere una forma di protocollo che eviti la possibilità di confusione, la perdita di dati e il blocco delle informazioni. Dispositivi veloci devono poter colloquiare con dispositivi più lenti. Il sistema non deve dipendere dai dispositivi ad esso connessi, altrimenti modifiche ed implementazioni si renderebbero impossibili. Deve esistere la possibilità di decidere con quale dispositivo di controllo colloquiare e quando. E, se sono presenti dispositivi con clock diverso connessi al bus, la sorgente master del clock bus deve essere definibile.

Tutti questi criteri hanno coinvolto lo sviluppo delle specifiche I²C.

4 L'idea I²C

Il bus I²C supporta qualsiasi tipo di IC (e.g. NMOS, CMOS, Bipolar). Due fili, serial data (SDA) e serial clock (SCL), per trasportare le informazioni tra i dispositivi connessi al bus. Ogni dispositivo è riconosciuto da un indirizzo univoco e può operare sia come trasmettitore che come ricevitore, dipende dalle funzionalità del dispositivo stesso. Ovviamente un LCD driver è solo un ricevitore, mentre una memoria può sia ricevere che trasmettere dati. In aggiunta al trasmettitore e al ricevitore, un dispositivo può essere considerato master o slave (vedi Tabella 1).

Si dice dispositivo master il dispositivo che inizia il trasferimento di dati sul bus generando lui il clock e consentendo di conseguenza il trasferimento, al tempo stesso gli altri dispositivi connessi si considerano slave.

TERMINE	DESCRIZIONE
Trasmettitore	Dispositivo in grado di inviare dati sul bus
Ricevitore	Dispositivo in grado di ricevere dati
Master	Dispositivo che inizia il trasferimento, genera il clock e termina il trasferimento
Slave	Dispositivo indirizzabile dal master
Multi Master	Più di un dispositivo master può controllare il bus senza causare corruzione dati
Arbitraggio	Procedura che assicura, se più di un master assume il controllo del bus a uno solo e consentito trasmettere senza corruzione dati
Sincronizzazione	Procedura che sincronizza il clock tra due o più dispositivi

Il bus I²C è un bus multi master, questo significa che più di un dispositivo ha la capacità di controllare il bus al quale è collegato.

Da evidenziare che la relazione trasmettitore ricevitore e master slave che troviamo sul bus I²C non è permanente ma dipende dalla direzione dei dati in quel periodo.

La possibilità di avere più master ha fatto sì che venisse sviluppata una procedura di arbitraggio.

La generazione del segnale di clock è sempre affidata al dispositivo master, ogni master genera il suo segnale di clock quando inizia un trasferimento dati.

5 Caratteristiche generali

Sia SDA che SCL sono linee bi direzionali connesse a tensione positiva per mezzo di due resistenze di pull-up. Quando il bus non è impegnato ambedue le linee sono alte, di conseguenza lo stadio d'uscita di un qualunque dispositivo I²C compatibile deve essere open drain o open collector. Il trasferimento dati sul bus può avvenire alla velocità massima di:

- Standard Mode 100Kbit/sec Max
- Fast Mode 400Kbit/sec Max
- High speed Mode 3.4Mbit/sec Max

Il numero massimo di dispositivi connessi al bus dipende solo dalla massima capacità ammissibile totale, che non deve superare i 400pF.

6 Trasferimento di bit

A causa della diversa natura dei dispositivi connessi al bus (NMOS, CMOS, etc.), il livello logico '0' (LOW) e '1' (HIGH) non sono fissi ma dipendono esclusivamente dal livello di tensione positiva ad esso applicata. Deve essere generato un impulso di clock per ogni bit trasferito.

6.1 Validità del dato

Il dato sulla linea SDA deve essere stabile durante tutto il periodo alto dell'impulso di clock. Lo stato di alto o basso della linea SDA può cambiare solo durante lo stato basso della linea SCL. (vedi fig. 1)

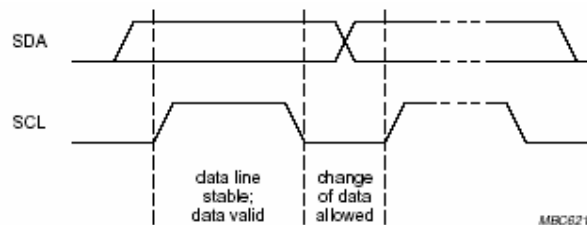


fig. 1 Trasferimento bit sul Bus I²C

6.2 Condizioni di START (S) e STOP (P)

Nella procedura di comunicazione del bus I²C, l'unica situazione definita è la situazione di START (S) e di STOP (P), mostrate in figura 2.

Di seguito definiamo che il dato è SDA e il clock è SCL

Un cambiamento di SDA, da alto a basso mentre SCL è alto risulta essere una condizione di START (S).

Mentre un cambiamento di SDA da basso a alto con SCL alto definisce una condizione di STOP (P).

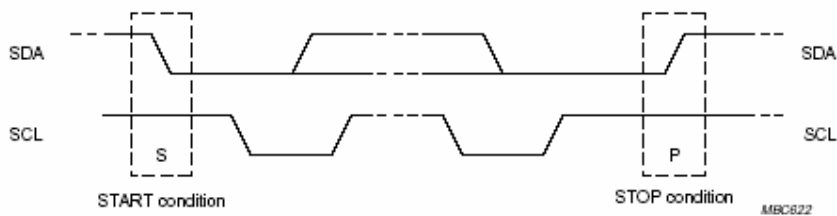


fig. 2 START e STOP

Le condizioni di START e STOP sono sempre generate dal dispositivo master. Il bus viene considerato impegnato dopo una condizione di START. Il bus viene considerato libero dopo un determinato tempo dalla condizione di STOP. Questo condizione viene descritta più avanti (forse sez. 10??)

Il bus resta impegnato (busy) se invece di un STOP viene generato uno START ripetuto (Sr), la condizione di START o START ripetuto ha la stessa identica funzionalità.

Il rilevamento di una delle due condizioni, START o STOP, da parte di un dispositivo slave è abbastanza semplice se questi incorpora l'interfaccia hardware necessaria.

Comunque, con dispositivi come alcuni microcontrollori che non possiedono l'interfaccia per il bus, questi devono poter campionare la linea dati SDA almeno due volte per ogni periodo di clock al fine di poter determinare la transazione.

7 Trasferimento Dati

7.1 Formato Byte

Ogni byte trasmesso sulla linea SDA deve essere lungo 8 bit, il numero di byte trasmessi è illimitato. Ogni byte deve essere seguito da un bit di acknowledge (ACK). Il dato deve essere trasferito con il bit più significativo (MSB) per primo (fig. 3). Se uno slave non può ricevere o terminare un altro byte di dati finché non è stata completata un'altra funzione, per esempio esecuzione di una routine interna di interrupt, questi può tenere la linea del clock SCL bassa in modo da forzare il dispositivo Master in uno stato d'attesa. Il trasferimento dei dati riprenderà non appena il dispositivo slave sarà in grado di ricevere o trasmettere un altro byte di dati valido rilasciando la linea SCL.

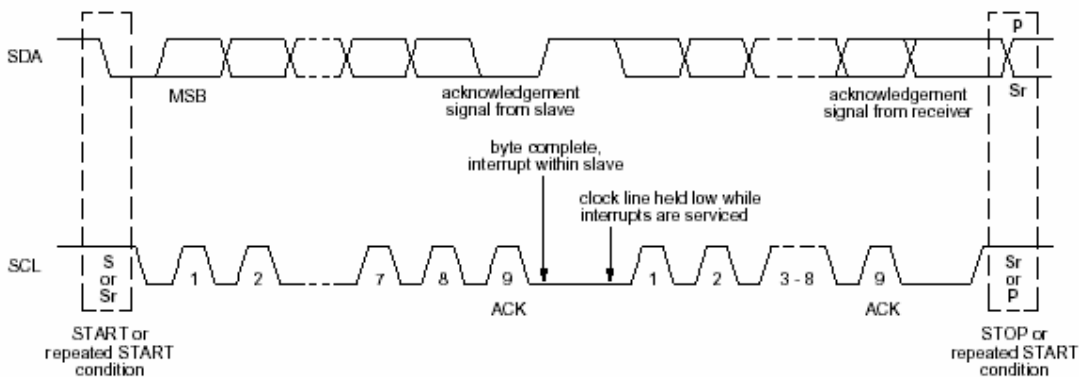


fig. 3 Trasferimento dati bus I²C

7.2 Acknowledge

Il trasferimento dati seguito dall'acknowledge, in seguito ACK, è sempre obbligatorio. L'impulso di clock relativo al ACK è sempre generato dal master, il trasmettitore rilascia la linea SDA (che va alta per effetto del pull up) durante l'impulso di clock per ACK.

Il ricevitore deve abbassare la linea SDA durante l'impulso di clock per ACK e trattenerla bassa durante tutto il periodo alto di clock (fig. 4).

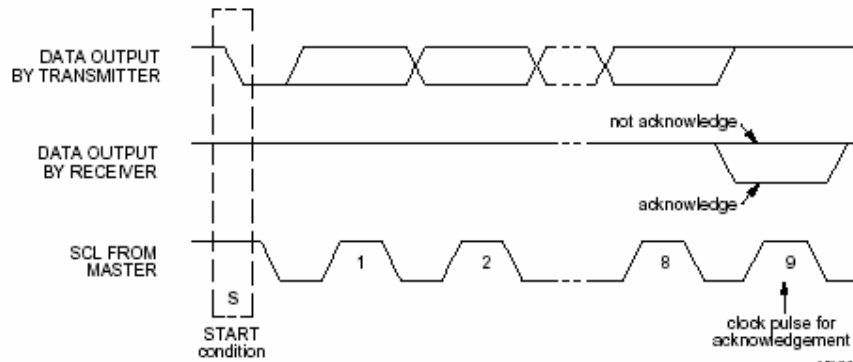


fig. 4 Acknowledge bus I²C

Normalmente, un ricevitore che ha ricevuto un indirizzamento, in seguito ADD, è obbligato a generare un ACK dopo ogni byte ricevuto.

Quando uno slave non genera ACK dopo aver ricevuto ADD (per esempio non può funzionare da trasmettitore o ricevitore poiché non è stato progettato per quello scopo), la linea SDA viene lasciata alta dallo slave. Il master può allora generare uno STOP o uno START ripetuto.

Se un master ricevitore è coinvolto in un trasferimento, l'ACK sull'ultimo byte trasmesso non viene generato e sull'impulso di clock per ACK il master ricevitore genera uno STOP o uno START ripetuto.

8 Arbitraggio e generazione del clock

8.1 Sincronizzazione

Tutti i dispositivi master generano il proprio clock sulla linea SCL durante il trasferimento di un messaggio sul bus. Il dato viene ritenuto valido solo durante il periodo in cui il clock è alto, in questo periodo il dato deve essere stabile. Un segnale di clock definito è comunque necessario per stabilire una procedura di arbitraggio bit a bit.

Continua sul documento originale (The I²C Bus Specification Ver 2.1 Jan 2000 page 11)

9 Indirizzi a 7 BIT

Il dato viene trasferito come mostrato in fig. 5. Dopo una condizione di START (S), viene trasmesso l'indirizzo dello slave. Questo indirizzo è formato da 7 bit seguiti da un ottavo bit che indica la direzione del dato (R/W), ottavo bit con valore zero indica una trasmissione (WRITE) mentre con valore uno indica una richiesta di dato (READ). Un trasferimento di dati termina sempre con una condizione di STOP (P) generata dal Master. Comunque, nel caso in cui il dispositivo master vuole continuare la comunicazione sul bus, può anche essere generata una condizione di START ripetuto (Sr) seguita da un indirizzo di un altro slave senza generare la prima condizione di STOP. Diverse combinazioni di read/write sono possibili con questo tipo di trasferimento.

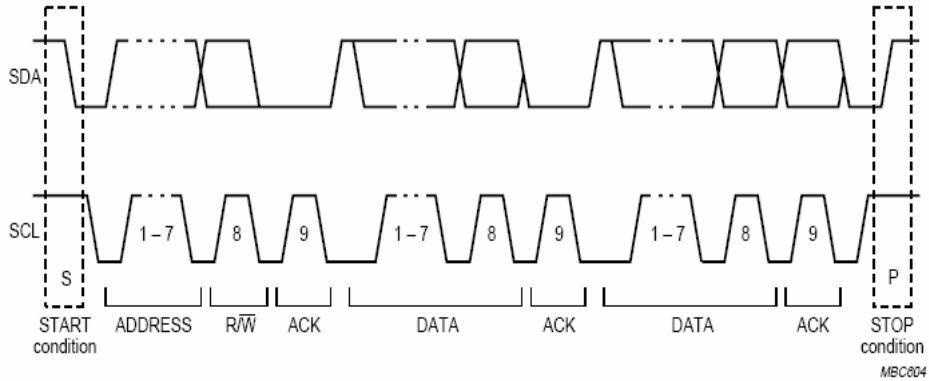


fig. 5 Trasferimento completo di dati bus I²C

Possibili trasferimenti di dati:

- Master Trasmettitore trasmette ad uno slave ricevitore. La direzione del trasferimento non cambia (fig. 6).

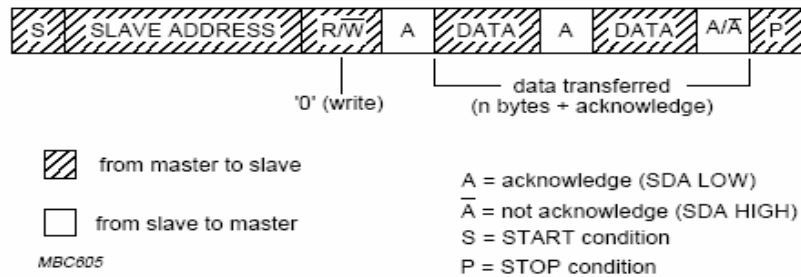


fig. 6

- Master ricevitore immediatamente dopo aver trasmesso il primo byte (address) allo slave (fig. 7). Nel momento del ricevimento del primo ACK il master trasmettitore inizia a diventare master ricevitore e lo slave ricevitore diventa slave trasmettitore. Il primo ACK è generato dallo slave. La condizione di STOP (P) è generata dal master che ha precedentemente inviato un NACK allo slave.

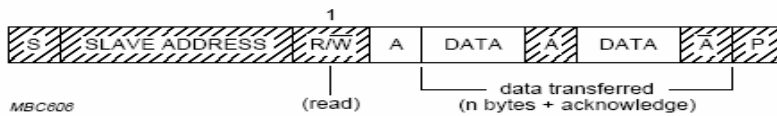


fig. 7

- Formato Combinato (fig. 8). Durante un cambiamento di direzione con un trasferimento in atto, lo START e l'indirizzamento dello slave vengono ripetute, ma con il bit di direzione R/W invertito. Se un master ricevitore genera una condizione di START ripetuto, ha precedentemente inviato un NACK.

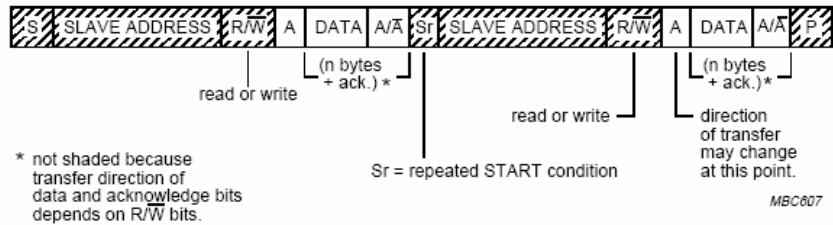


Fig. 8

10 Indirizzamento a 7 BIT

La procedura di indirizzamento sul bus I2C è tale che il primo byte dopo lo START generalmente determina il dispositivo slave che il master seleziona.

L'eccezione "general call address" indirizza tutti i dispositivi, quando viene usata questo indirizzo tutti i dispositivi dovrebbero, in teoria, rispondere con un ACK, comunque dispositivi non progettati per riconoscere questa funzione ignorano questo indirizzo. Il secondo byte della "general call address" indica il tipo di azione da compiere, più avanti approfondiremo maggiormente questo tipo di procedura.

10.1 Definizione dei Bits del primo Byte

I primi sette bit del primo byte esprimono l'indirizzo del dispositivo slave (fig. 9).

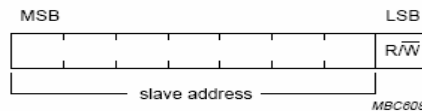


fig. 9

L'ottavo bit LSB (bit meno significativo) determina la direzione del messaggio.

Uno zero nel LSB significa che il master vuole scrivere informazioni sul dispositivo slave selezionato, mentre un uno nel LSB significa che il master vuole leggere informazioni dal dispositivo slave selezionato.

Quando viene inviato un indirizzo, ogni dispositivo connesso al bus confronta i primi 7 bit inviati dopo lo START, con il proprio indirizzo, se questo coincide il dispositivo slave si considera automaticamente indirizzato dal master come slave ricevitore o slave trasmettitore a seconda del valore del LSB.

Un indirizzo slave è composto da una parte fissa e da una programmabile. La parte programmabile consente di inserire in un sistema più dispositivi uguali (e.g. memorie, I/O) il numero massimo di dispositivi uguali collegabili allo stesso bus dipende dal numero di pin disponibili.

Il comitato per il bus I2C coordina l'assegnazione degli indirizzi, maggiori informazioni in merito possono essere reperite sul sito della Philips Semiconductors.

Due gruppi di 8 indirizzi (0000XXXX 1111XXXX) sono stati riservati per gli scopi mostrati nella tabella seguente. La combinazione 11110XX è stata riservata per l'indirizzamento a 10 bit (non trattata in questa prima stesura).

Slave ADD	R/W BIT	DESCRIZIONE
0000 000	0	General call Address
0000 000	1	START Byte (1)
0000 001	X	CBUS address (2)
0000 010	X	Riservato per bus diversi (3)
0000 011	X	Riservato per future implementazioni
0000 1xx	X	Codice per master High speed
1111 1xx	X	Riservato per future implementazioni
1111 0xx	X	Indirizzamento a 10 bit

Note:

Nessun dispositivo slave deve inviare ACK al ricevimento dello SART Byte.

Questo indirizzo è riservato per consentire l'utilizzo misto di CBUS e I2C.

Indirizzo riservato per consentire l'utilizzo con altri tipi di bus compatibili con I2C.

10.1.1 General call Address

La procedura "general call address" è utilizzata per indirizzare tutti i dispositivi connessi al bus. Comunque, se un dispositivo non necessita il supporto di questa struttura, può ignorare questo indirizzo ma non deve inviare alcun ACK, al ricevimento di questo byte. Se un dispositivo richiede dei dati dalla procedura general call, deve inviare l'ACK e comportarsi come slave ricevitore. Il secondo e i successivi byte devono essere seguiti da ACK emesso da tutti i dispositivi in grado gestire il dato. Un dispositivo slave che non è in grado di gestire il dato deve ignorarlo e non inviare NACK. Il significato della procedura general call è espresso nel secondo byte (fig. a).

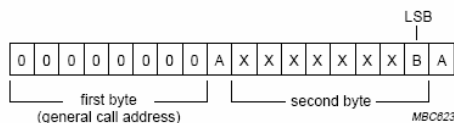


fig. a

Ci sono due casi da considerare:

Quando LSB B vale 'zero'.

Quando LSB B vale 'uno'.

Quando LSB B vale 'zero', il secondo byte ha il seguente significato:

0000 0110 (H 06). Reset e scrittura della parte programmabile hardware del address del dispositivo slave. Al ricevimento di questi due byte durante la procedura di general call, tutti i dispositivi disegnati per rispondere, subiranno un reset della parte programmabile hardware del loro address. Assicurarsi che nessun dispositivo al termine di questo comando non tenga basse le linee SDA e SCL, causando un blocco del bus.

0000 0100 (H 04). Scrittura della parte programmabile hardware del address del dispositivo slave. Tutti i dispositivi che hanno una parte programmabile hardware predisposti per il funzionamento della general call cattureranno, questa parte programmabile al ricevimento del secondo byte senza subire un reset.

0000 0000 (H 0). Questo codice non permette l'invio di un secondo byte.

La procedura per la sequenza di programmazione di questa parte è descritta nei singoli datasheet dei dispositivi.

Il codice non rimane fissato nel dispositivo che a suo volta la ignorerà.

Quando LSB B vale 'uno', la sequenza dei due byte è una 'hardware general call'. Questo significa che la sequenza è stata trasmessa da un dispositivo master settato via hardware, tipo una tastiera che non può essere programmata per trasmettere uno specifico indirizzo (address) slave. Poiché un dispositivo hardware master non è in grado di conoscere in anticipo quale dispositivo slave dovrà ricevere il messaggio, questi può solo generare una hardware general call con il suo address, identificandosi da solo nel sistema (fig. b).

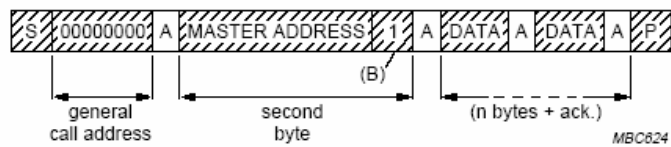


fig. b

I sette bit contenuti nel secondo byte contengono l'address del dispositivo master hardware. Questo indirizzo viene catturato da un dispositivo intelligente (e.g. microcontrollore) connesso sul bus che interpreta le informazioni inviate dal master hardware. Se il dispositivo master hardware può agire anche come slave il suo indirizzo slave sarà lo stesso del master hardware.

In alcuni sistemi un'alternativa può essere quella di impostare il dispositivo hardware master come slave ricevitore subito dopo aver eseguito un reset del sistema. In questo modo il dispositivo master del sistema chiama il master hardware trasmettitore (che ora è settato come slave ricevitore) al quale devono essere trasmessi i dati (fig. c). Dopo questa procedura di programmazione il dispositivo master hardware ritorna ad essere master hardware trasmettitore.

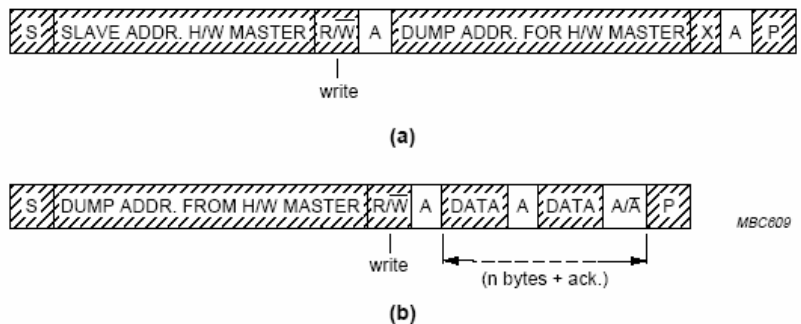


Fig. c

10.1.2 START byte

I microcontrollore possono essere collegati al bus I2C in due modi. Un microcontrollore con interfaccia hardware per il bus I2C che può essere programmato per agire solo con interventi via interruptus. Quando un dispositivo non è provvisto di un'interfaccia bisogna costantemente

monitorare il bus attraverso il software. Ovviamente in questo caso il microcontrollore spende molto tempo a monitorare piuttosto che a svolgere le funzioni primarie per il quale viene utilizzato.

C'è perciò una differenza di velocità tra un dispositivo hardware veloce ed un microcontrollore relativamente lento che deve far conto sul polling software.

In questo caso, un trasferimento di dati può essere preceduto da una procedura di start che risulta essere molto più lunga del normale (fig. d).

La procedura di START consiste nell'inviare:

Una condizione di START (S)

Uno byte di START (0000 0001)

Un impulso di clock per ACK

Una condizione di START ripetuto (Sr)

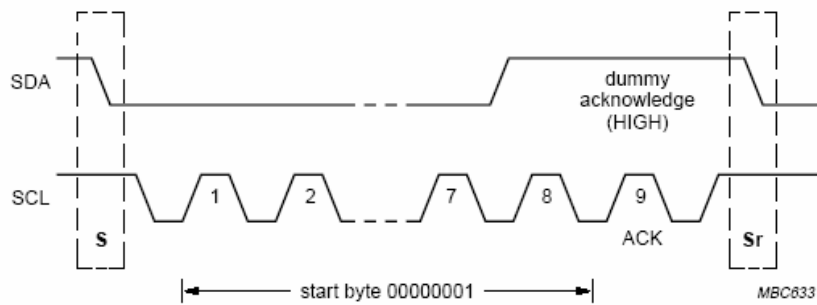


fig. d