

IL PROTOCOLLO I2C

Introduzione

Scopo di questo Tutorial è quello di introdurre le caratteristiche del protocollo I²C al fine di permetterne un corretto utilizzo.

Verranno illustrate le potenzialità di questo standard che sta divenendo ormai una linea guida per molte tipologie di progetti. La sua semplicità d'uso e le innumerevoli applicazioni in cui si presta, lo rendono degno di nota e di apprezzamento.

Uno sguardo d'insieme

Il protocollo I²C è uno standard ideato dalla Philips (ora NXP) che ne possiede per altro la proprietà.

Venne ideato nel 1980 per superare le difficoltà inerenti all'utilizzo di bus paralleli per la comunicazione tra un'unità di controllo e le varie periferiche.

Le soluzioni che vengono adottate per permettere la comunicazione tra un microcontrollore e le periferiche esterne fanno generalmente uso di una comunicazione parallela. Per tale ragione il bus¹ su cui deve viaggiare l'informazione è costituito da molti fili. Fin quando bisogna collegare una sola periferica al microcontrollore i problemi legati alla presenza di molte linee possono essere tenuti sotto controllo. Qualora le periferiche dovessero essere più di una, far giungere il bus ad ogni periferica può diventare un problema.

Un semplice bus ad otto linee comporta comunque la presenza ad ogni integrato di almeno altrettanti pin necessari per la comunicazione. Questo significa che le dimensioni dell'integrato vengono ad essere dipendenti dalla dimensione del bus stesso. Ciò comporta che lo stesso PCB (Print Circuit Board) sul quale andrà montato l'integrato sarà più grande e quindi più costoso.

Questi problemi vengono interamente superati dal bus I²C, che permette una comunicazione tra periferiche con due sole linee²; questa è la ragione per cui non è raro avere integrati con bus I²C a soli otto pins.

I²C bus è uno standard seriale che a

differenza del protocollo RS232³, che permette un collegamento punto punto tra due sole periferiche, permette di collegare sullo stesso bus un numero elevato di periferiche, ognuna individuata da un proprio indirizzo.

La possibilità di poter collegare più periferiche sullo stesso bus è permesso anche dal bus CAN. Quest'ultimo protocollo è stato ideato per operare in ambienti particolarmente rumorosi e in cui si debba raggiungere un grado di sicurezza nella trasmissione dati particolarmente elevata. Per tali ragioni il bus CAN è ormai accettato come standard in ambito automobilistico, per mettere in comunicazione i vari dispositivi elettronici che sempre più frequentemente vengono installati a bordo.

Un notevole vantaggio dei dispositivi che fanno uso del bus I²C è quello della loro semplicità d'uso. Infatti tutte le regole del protocollo che bisogna rispettare per una corretta comunicazione vengono gestite a livello hardware, dunque il progettista non si deve preoccupare di nulla.

Da quando è nato, tale protocollo è stato aggiornato al fine di adeguarlo alle diverse esigenze che il mondo dell'elettronica andava mostrando.

Tutte le modifiche apportate sono sempre state compatibili dall'alto verso il basso, ovvero gli integrati che soddisfano gli ultimi standard possono comunicare sempre con gli integrati della generazione precedente.

La prima versione del bus I²C permette di trasmettere fino a 100Kbit/s (modalità standard⁴). Questa velocità è stata portata a 400Kbit/s nelle modifiche apportate nel 1992 (modalità veloce⁵). Nel 1998 la velocità è stata portata fino a 3.4Mbit/s (modalità ad alta velocità⁶).

Non necessariamente gli integrati di ultima generazione devono rispettare la modalità ad alta velocità.

Le periferiche che fanno uso del bus I²C per comunicare con un'unità di controllo sono : memorie, real time clock calendar, LCD, potenziometri digitali, convertitori A/D,

¹ Per bus si intende semplicemente un insieme di linee sui cui viaggia un segnale elettrico.

² Alle due linee bisogna comunque aggiungere la linea di massa comune.

³ Si veda la dispensa "Il protocollo RS232" per maggior chiarimenti sull'argomento.

⁴ Standard mode.

⁵ Fast speed mode.

⁶ High speed mode.

sintonizzatori radio, controller per toni DTMF, periferiche generiche per estendere il numero degli ingressi o delle uscite (PCF8574), sensori per la temperatura, controllori audio e molto altro. Si capisce dunque che le possibilità di scelta da parte del progettista sono notevoli.

Un altro vantaggio che permette di raggiungere il bus I²C è quello di poter aggiungere o togliere delle periferiche dal bus senza influenzare il resto del circuito. Questo si traduce in una facile scalabilità verso l'alto del sistema, ovvero si può migliorare un sistema aggiungendo nuove caratteristiche senza dover toccare l'hardware⁷.

Specifiche elettriche del bus I²C

Il bus I²C è un bus seriale che necessita di sole due linee nominate SDA (Serial Data) e SCL (Serial Clock) più la linea di massa. Ambedue le linee sono bidirezionali⁸. La prima è utilizzata per il transito dei dati che sono in formato ad 8 bit, mentre la seconda è utilizzata per trasmettere il segnale di clock necessario per la sincronizzazione della trasmissione. Il bus I²C permette la connessione di più periferiche su uno stesso bus ma permette la comunicazione tra due soli dispositivi alla volta.

Chi trasmette le informazioni è chiamato trasmettitore mentre chi le riceve è chiamato ricevitore. L'essere il trasmettitore o il ricevitore non è una posizione fissa, ovvero, un trasmettitore può anche divenire ricevitore in una differente fase della trasmissione dati.

In ogni comunicazione è invece fissa la posizione del cosiddetto Master (Padrone) e del cosiddetto Slave (Schiavo). Il Master è il dispositivo che inizia la comunicazione ed è lui a terminarla, lo slave può solo ricevere o trasmettere informazioni su richiesta del Master.

Non tutti i dispositivi possono essere dei Master del bus I²C. Per esempio una memoria

per il mantenimento dei dati non sarà un Master del bus, mentre è ragionevole supporre che un microcontrollore lo possa essere⁹.

Su uno stesso bus è inoltre possibile la presenza di più Master, ma solo uno alla volta ricoprirà questo ruolo. Se per esempio due microcontrollori iniziano una comunicazione, anche se potenzialmente potrebbero essere ambedue dei Master, solo uno lo sarà, in particolare il Master sarà quello che ha iniziato la comunicazione, mentre l'altro sarà uno slave.

Ogni periferica inserita nel bus I²C possiede un indirizzo che sul bus la individua in modo univoco. Questo indirizzo può essere fissato dal produttore in sede di fabbricazione o parzialmente fissato dal progettista. L'indirizzo è costituito da 7 bit nelle versioni standard o da 10 bit nelle versioni estese.

Nel caso di indirizzamento a 7 bit si avrebbe potenzialmente la possibilità di indirizzare 128 periferiche mentre nel caso di 10 bit si avrebbe la possibilità di indirizzare fino a 1024 periferiche.

Il numero di periferiche ora citate non sono comunque raggiungibili dal momento che alcuni indirizzi, come si vedrà, sono riservati a funzioni speciali.

Nel caso in cui l'indirizzo che l'integrato ha all'interno del bus I²C venga fissato dall'industria, conduce al fatto che su un bus non potranno essere presenti due integrati dello stesso tipo.

Questa soluzione viene generalmente scelta per per i real time clock calendar, ovvero per gli orologi. E' ragionevole infatti presumere che in un circuito, e in particolare sullo stesso bus, sia presente un solo orologio che mantenga ora e data.

Tale scelta porta il vantaggio che l'integrato potrà avere meno pin. Se proprio si dovesse avere la necessità di inserire due orologi¹⁰, o comunque due periferiche con stesso indirizzo è necessario dividere il bus in due, questo può per esempio essere ottenuto con l'integrato PCA9544 della Philips.

⁹ Molti microcontrollori integrano al loro interno l'hardware necessario per la gestione del bus I²C sia in modalità Master che Slave.

¹⁰ Il PCF8583 possiede un pin, nominato A0, per impostare il bit meno significativo dell'indirizzo. Dunque sarà possibile inserire due clock calendar di questo tipo sullo stesso bus.

⁷ Naturalmente il software dell'unità di controllo dovrà essere cambiato affinché possa riconoscere la nuova periferica.

Quanto detto non vale se il software è predisposto per accettare la nuova periferica, la quale può esser dunque inserita senza alcuna modifica né hardware né software.

⁸ Dal momento che la linea dati è bidirezionale si ha che il sistema è half-duplex (si veda "Il protocollo RS232" per maggior chiarimenti)

Qualora si debba far uso di memorie, un eventuale limite come per l'orologio potrebbe essere notevole. Per tale ragione in questo caso si ha la possibilità di impostare l'indirizzo dell'integrato intervenendo su alcuni bit. Ad esempio la memoria 24LC256¹¹ della Microchip possiede tre pin in uscita nominati A0, A1, A2, per mezzo dei quali è possibile impostare i tre bit meno significativi dell'indirizzo che caratterizza la memoria. Questo significa che sarà possibile mettere fino ad otto memorie dello stesso tipo sullo stesso bus (si veda Figura 1). Il numero di pin disponibili al progettista, per modificare l'indirizzo, varia a seconda dell'applicazione dell'integrato.

Un limite sul numero massimo di periferiche che è possibile connettere sul bus è dunque imposto dall'indirizzo. Un vincolo molto più stringente è imposto dalla capacità totale della linea che deve essere limitata a non più di 400 pF. Il valore di questa capacità viene a dipendere dal numero di dispositivi e dalla lunghezza del bus stesso. Potenzialmente, dal momento che una linea tipicamente introduce una capacità parassita di circa 80 pF/m, potrà essere lunga fino a 5m.

Il limite imposto dalla capacità viene a dipendere dalla velocità con cui devono avvenire le transizioni dei bit dal livello basso al livello alto (rise time).

Qualora si debbano raggiungere distanze maggiori, o il numero dei dispositivi è così elevato da superare tale capacità, è possibile spezzare il bus, in due o più parti, per mezzo di ripetitori (Repeater) quali il PCA9515, PCA9516 e il PCA9518 della Philips. Per mezzo di questi integrati sarà possibile avere 400 pF per ogni semi parte del bus.

Per mezzo del P82B96, si ha invece la possibilità di raggiungere distanze di 100m alla frequenza di 71Kbit/s e di ben 1Km alla frequenza di 31Kbit/s.

Da quanto appena esposto si comprende che un bus che possa funzionare a 100Kbit/s non necessariamente deve lavorare a tale frequenza. Sarà il Master, opportunamente impostato, a scandire il sincronismo e dunque la velocità di trasmissione.

La necessità di dividere il bus in due la si può

avere anche nel caso in cui siano presenti dispositivi con diversi standard di velocità.

La divisione del bus è necessaria solo nel caso in cui si voglia far comunicare le periferiche a più alta velocità al loro massimo. Nel caso ci si adegua alla periferica più lenta non è necessario dividere il bus. Gli integrati che sfruttano I²C possono lavorare a diverse tensioni, dai tipici 5V a 3.3V ed anche 2.5V¹² permettendo consumi così ridotti da poter essere montati anche sui dispositivi portatili alimentati a batteria. Nel caso si debba far comunicare periferiche, che certamente staranno in due distinti bus, che lavorano a tensioni differenti, è necessario porre un ponte tra i due bus al fine di permettere una corretta comunicazione.

Ultima nota, prima di vedere come avviene una comunicazione sul bus I²C, riguarda le linee SDA e SCL. Queste linee devono essere implementate per mezzo di uscite open drain o open collector. Questa nota è particolarmente importante qualora si voglia implementare il protocollo I²C interamente per via software.

Tale caratteristica rende necessaria una resistenza di pull up per ogni linea, ovvero di una resistenza collegata tra la linea e Vcc, come riportato in Figura 1. Questo significa che quando le linee SDA e SCL non sono utilizzate, sono a livello alto.

Valori tipici per le resistenze di pull up sono compresi tra 2K Ω e 10K Ω . Il primo valore è utilizzato per un bus fino a 400Kbit/s mentre il secondo per velocità fino a 100Kbit.

Il valore corretto viene comunque a dipendere, oltre che dalla frequenza di trasmissione, anche dalla capacità totale di linea.

Quanto segue, fino a Figura 2, è un appunto generalmente non utile per applicazioni comuni.

Nel caso di trasmissioni a 100Kbit/s la resistenza pull up è sufficiente fino a capacità totali di linea pari a 400 pF. Nel caso di velocità a 400 Kbit/s la resistenza di pull up è sufficiente fino a capacità totali di linea non superiori a 200 pF, mentre per velocità a 3.4 Mbits/s la resistenza è idonea fino a capacità non superiori a 100 pF.

Cosa significa questo limite? Il problema della capacità di linea è legato al tempo di salita con cui si riesce ad ottenere una variazione dal

¹¹ La Microchip produce una vasta gamma di memorie con interfacciamento I²C.

¹² Il livello di tensione di lavoro tende ormai a spostarsi a 1.8V.

livello logico basso a livello logico alto. Nel caso in cui si faccia uso di una resistenza, i fronti di salita hanno un andamento esponenziale tipico, della carica e scarica di un condensatore.

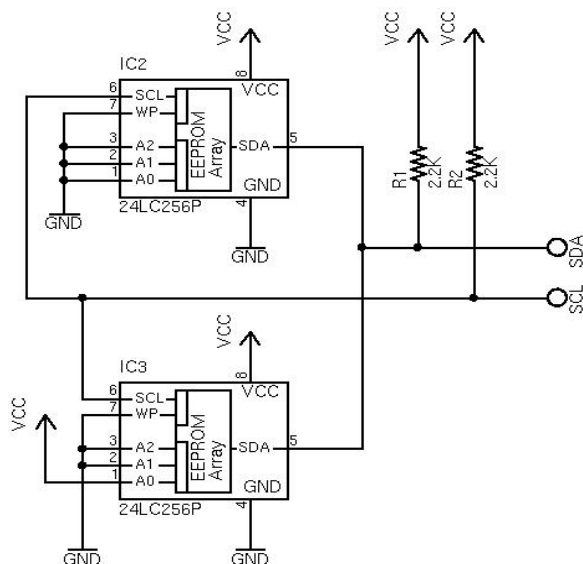


Figura 1: Collegamento dei resistori di pull up e dettaglio dell'indirizzamento di dispositivi dello stesso tipo

All'aumentare della capacità, il tempo di salita tende a degradarsi a tal punto da non rispettare più le specifiche elettriche I²C. Per porre rimedio a questo limite bisogna porre, in sostituzione della resistenza di pull up, un generatore di corrente costante.

Tale generatore, avendo una corrente indipendente dal livello di carica della capacità di linea, provvederà a far ottenere dei fronti di salita la cui pendenza rimane costante¹³. La differenza dei fronti di salita ottenuti nel caso di resistenza di pull up e di generatore costante sono riportati in Figura 2.

Comunicazioni sul bus I²C

Vediamo ora come avviene una comunicazione su di un bus I²C. Come detto solo le periferiche che possono essere dei

Master possono avviare una comunicazione.

Supponiamo che un microcontrollore voglia leggere da una memoria esterna collegata al bus I²C dei dati precedentemente memorizzati.

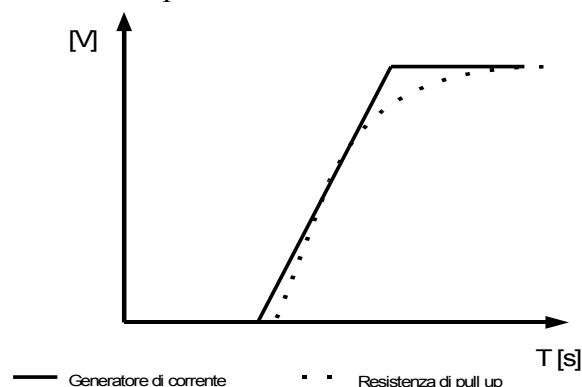


Figura 2: Differenza tra i fronti d'onda nel caso di una resistenza di pull up e di un generatore di corrente

Le fasi che devono essere seguite sono le seguenti:

1. Il Master controlla se le linee SDA e SCL non siano attive, ovvero siano poste ambedue a livello alto¹⁴.
2. Se il bus è libero invia il messaggio che fa capire alle altre periferiche che il bus è ora occupato. Le altre periferiche si metteranno allora in ascolto per comprendere con chi il Master ha intenzione di comunicare.
3. Il Master provvede all'invio del segnale di sincronizzazione sulla linea SCL, che sarà rappresentato da un onda quadra, non necessariamente periodica.
4. Il Master invia l'indirizzo della periferica con la quale vuole parlare.
5. Il Master segnala poi se la comunicazione che vuole intraprendere verso la periferica è di lettura o scrittura.
6. Il Master attende la risposta da parte della periferica che nella chiamata ha riconosciuto il suo indirizzo. Se nessuna

¹³ Questo può essere un modo per ottenere un segnale a forma triangolare. Normalmente si preferisce però generare un segnale ad onda quadra e poi provvedere all'integrazione di quest'ultima.

¹⁴ Il livello alto, in termini di tensione, viene a dipendere dagli integrati che si sta utilizzando. Se $V_{cc}=3.3V$, i resistori di pull up porteranno le linee SDA e SCL al livello pari a 3.3. Diversamente se l'alimentazione è ottenuta con 5V il livello alto sarà di 5V.

periferica risponde il Master libera il bus.

7. Dopo l'avvenuto riconoscimento della periferica, il Master inizia lo scambio dei dati. Lo scambio avviene inviando pacchetti di 8 bit. Ad ogni pacchetto si deve attendere il segnale che avvisa dell'avvenuta ricezione.
8. Quando la trasmissione è terminata il Master libera il bus inviando un segnale di stop.

Vediamo in dettaglio come vengono realmente ottenute le varie fasi.

fase 1-2

L'hardware del microcontrollore, dedicato alla gestione dell'interfaccia I²C, controlla le linee SDA e SCL per un tempo superiore al massimo periodo di trasmissione consentito dall'hardware.

Se il bus risulta libero, il Master invia la sequenza di Start, che consiste nel portare la linea SDA a livello basso quando SCL è a livello alto. L'andamento delle due linee SDA e SCL, per segnalare lo Start, è riportata in Figura 3.

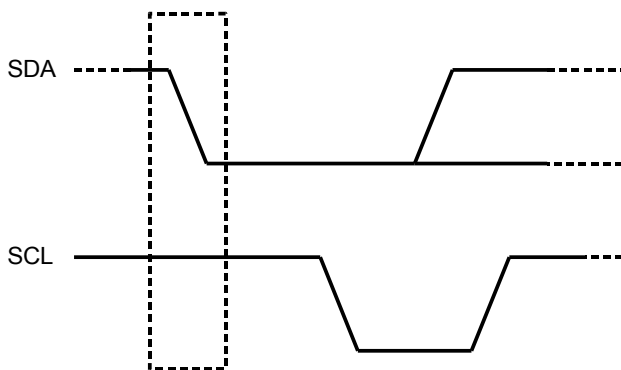


Figura 3: Sequenza di Start

Dopo l'invio della sequenza di Start, il bus è considerato occupato.

fase 3

Dopo la transizione della linea SDA da alto a basso, il Master invia il segnale di sincronizzazione per le altre periferiche. A differenza della sequenza di Start e di Stop la linea SDA assume un valore valido solo se la

linea SCL è a livello basso.

Questo vuol dire che non sono ammesse transizioni di livello della linea SDA durante il livello alto della linea SCL, se non da parte del Master per inviare un nuovo Start o uno Stop.

fase 4-5

Come detto il formato dell'indirizzo può essere sia a 7 bit che a 10 bit, per semplicità si considera prima il formato a 7 bit.

I 7 bit dell'indirizzo vengono inviati dal bit più significativo al bit meno significativo. In coda a questo indirizzo viene aggiunto un bit per segnalare se il Master vuole intraprendere, con la periferica individuata da tale indirizzo, una comunicazione di scrittura o di lettura. In particolare se tale bit è 0 vuol dire che il Master vuole scrivere sulla periferica, se il bit è 1 vuol dire che il Master vuole leggere della periferica.

Il formato del byte (8 bit) che vengono così inviati è riportato in Figura 4 a.

Nel caso in cui l'indirizzo della periferica è individuato da 10 bit, si capisce che sarà necessario inviare un secondo byte.

Per mantenere la compatibilità con il formato a 7 bit è però necessario qualcosa che permetta alle periferiche in ricezione di capire che il primo byte non rappresenta in realtà il formato d'indirizzo a 7 bit. Questo lo si è ottenuto proibendo alcuni indirizzi nel formato a 7 bit.

Byte 1/1

A6	A5	A4	A3	A2	A1	A0	R/W
----	----	----	----	----	----	----	-----

a)

Byte 1/2

1	1	1	1	0	A9	A8	R/W
---	---	---	---	---	----	----	-----

Byte 2/2

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

b)

Figura 4: Formato dell'indirizzo a 7 bit (a) e a 10 bit (b)

Se gli indirizzi proibiti¹⁵ dovessero essere utilizzati vorrebbe dire che il primo byte serve per indirizzare una periferica con indirizzo a 10 bit.

Gli indirizzi proibiti che individuano il formato d'indirizzamento a 10 bit sono rappresentati dalle combinazioni 1111xxx, dove x può valere sia 0 che 1.

In particolare la prima x vicino all'1 viene posta a 0 così con il primo byte si inviano solo i due bit più significativi dell'indirizzo a 10 bit.

L'ottavo bit rappresenta come nel caso ad indirizzamento a 7 bit la condizione di lettura o scrittura. Gli altri 8 bit, dell'indirizzo a 10 bit, vengono inviati con un secondo byte. Il formato dell'indirizzamento a 10 bit è riportato in Figura 4 b.

fase 6

L'invio dell'indirizzo a 7 bit e della modalità del colloquio (lettura/scrittura), avviene grazie ad otto transizioni, da livello alto basso, della linea SCL. Al nono impulso della linea SCL il Master si aspetta una risposta di un bit da parte della periferica che ha chiamato¹⁶. La risposta della periferica chiamata consiste nel mantenere a livello basso la linea SDA, per la durata di un ciclo SCL. In gergo si dice che il Master attende l'Acknowledge da parte della periferica chiamata. Una sola periferica risponderà alla chiamata del Master.

Nel caso di indirizzamento a 10 bit si può anche avere che più periferiche rispondano al primo byte, ma solo una risponderà anche al secondo.

Qualora la periferica non sia presente il Master libera il bus permettendo ad eventuali altri Master di prenderne il controllo.

fase 7

Dopo l'avvenuto riconoscimento, avviene lo scambio dei dati verso la periferica, nel caso di scrittura, o dalla periferica al Master, in caso di lettura. In una comunicazione si possono avere sia fasi di scrittura che di lettura. Si prenda ad

esempio una fase di lettura da memoria. In primo luogo bisognerà scrivere dei byte per impostare l'indirizzo che si vuole leggere, poi si potrà effettivamente leggere il byte¹⁷.

Ad ogni invio di un byte sarà necessario l'Acknowledge del byte inviato o ricevuto. In particolare se il Master invia un byte allo Slave si aspetta, dopo l'ottavo bit, un bit basso sulla linea SDA. Se lo Slave sta inviando un byte al Master si aspetta che quest'ultimo invii un bit alto dopo, aver ricevuto il byte. La mancanza dell'Acknowledge determina un errore di comunicazione.

fase 8

Quando la comunicazione è terminata, il Master libera il bus, inviando la sequenza di Stop.

Questa consiste nella transizione dal livello basso ad alto della linea SDA, quando la linea SCL è alta.

In Figura 5 è riportata la sequenza di Stop generata dal Master.

Se il Master deve effettuare un'altra comunicazione con un altro Slave, piuttosto che liberare il bus e rischiare di perdere il diritto del controllo, può inviare una nuova sequenza di Start.

Il vecchio slave comprenderà che la comunicazione con lui è terminata.

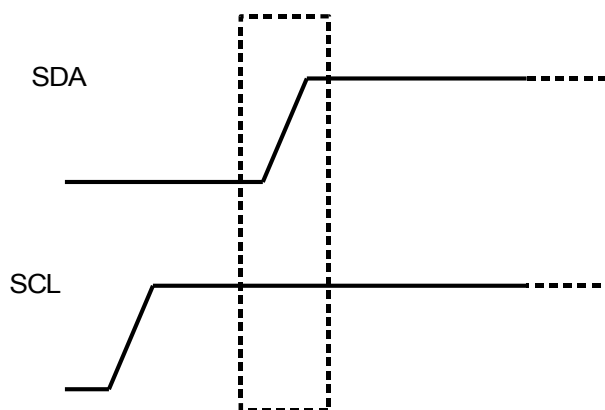


Figura 5: Sequenza di Stop

Le potenzialità dell'interfaccia I²C consistono nel fatto di poter disporre di una larga varietà di

¹⁵ In realtà sono presenti altri gruppi d'indirizzi proibiti che individuano altre funzioni speciali o sono stati lasciati per future estensioni dello standard I²C.

¹⁶ Si capisce che colui che fa il software deve conoscere a priori gli indirizzi delle periferiche montate o che possono essere montate.

¹⁷ Le memorie permettono istruzioni speciali per saltare la fase d'indirizzamento qualora si debbano leggere più indirizzi consecutivi. Comunque l'indirizzo d'inizio dovrà essere inviato.

dispositivi che possiedono al loro interno l'hardware necessario per la gestione automatica del protocollo.

Unica complicazione potrebbe essere la gestione dell'hardware da parte del microcontrollore. Infatti, per poter utilizzare l'hardware I²C interno al microcontrollore, è comunque necessaria una fase di inizializzazione. Le procedure che si realizzano, possono poi essere riutilizzate in altre applicazioni, passando per questo problema una sola volta.

Se si fa uso dei microcontrollori della Microchip, è presente il tool gratuito Maestro, che mette a disposizione un gran numero di funzioni, belle e pronte, da utilizzare per il I²C e molto altro.

Nel caso si programmi i microcontrollori con linguaggi ad alto livello, si hanno generalmente già disponibili le funzioni per la gestione del bus I²C.

Bibliografia

www.philips.com : da questo sito è possibile scaricare le specifiche tecniche ufficiali sul bus I²C (AN10216-01 I²C Manual; The I²C-Bus Specification, Version 2.1).

www.microchip.com : da questo sito è possibile scaricare il tool Maestro.