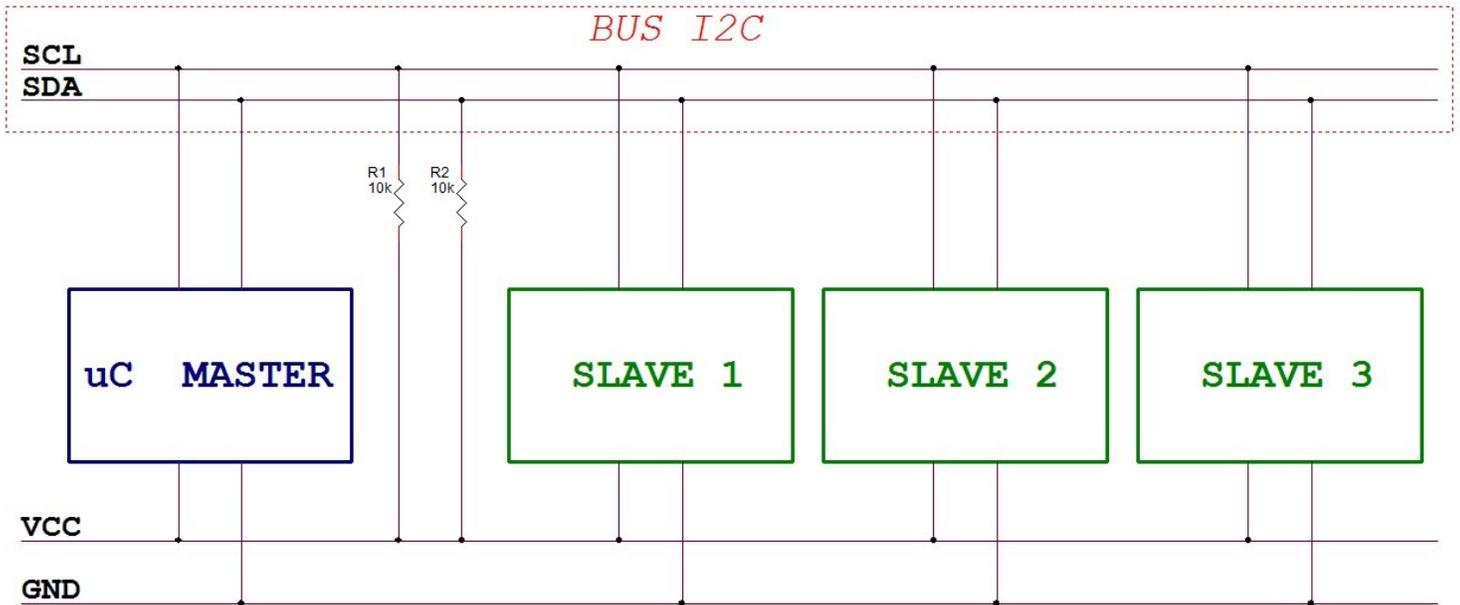


PROTOCOLLO I2C - GPIO expander PCF8574

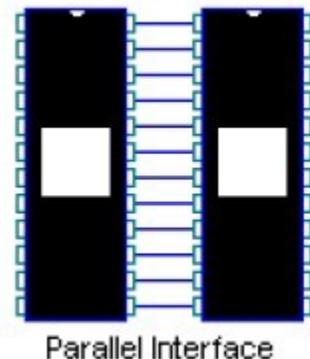
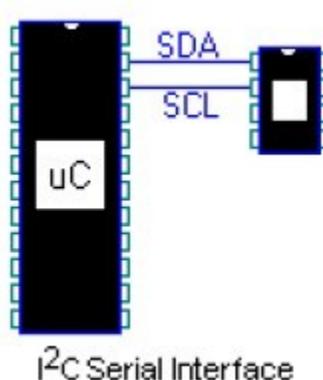
Il protocollo I2C è uno standard di comunicazione seriale sincrono half-duplex che utilizza due sole linee di collegamento SCL e SDA per consentire la comunicazione tra un dispositivo MASTER e più dispositivi SLAVE, come in figura.



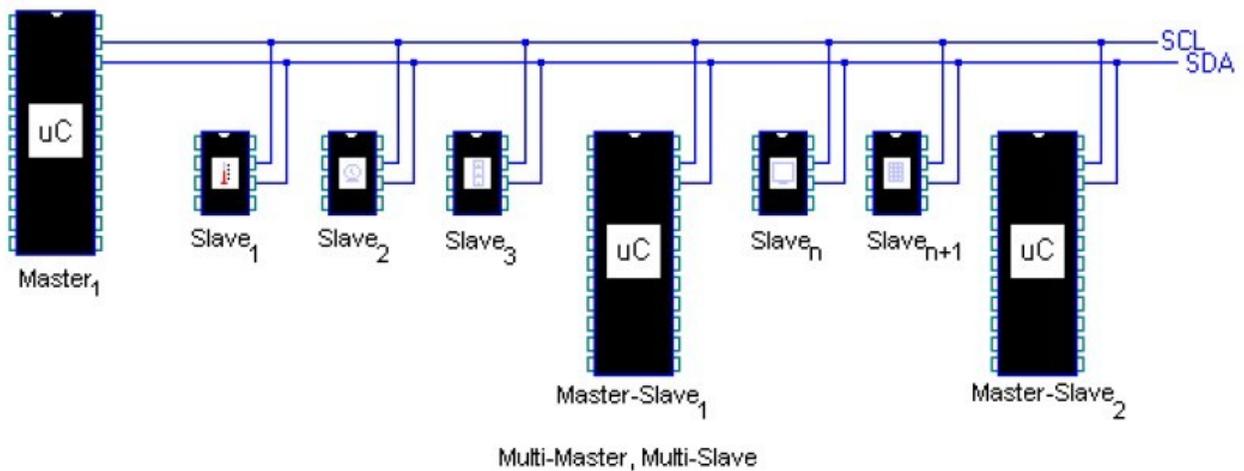
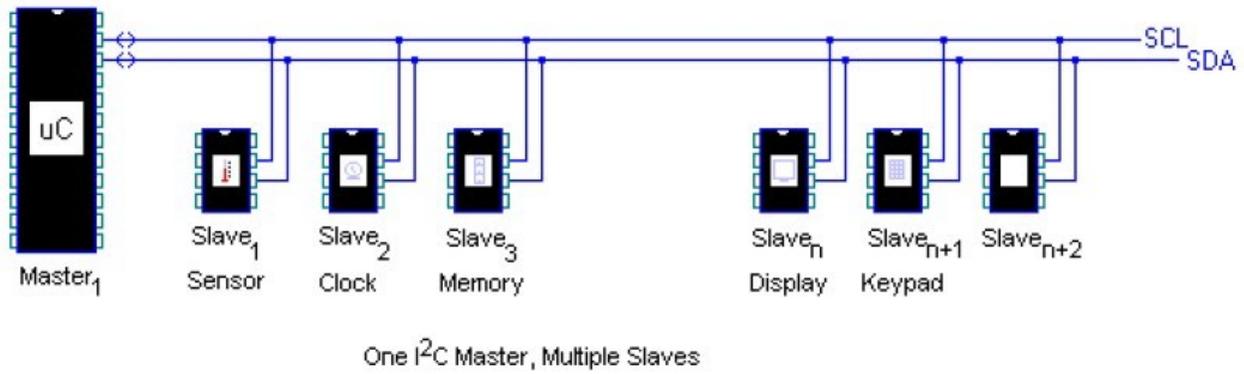
La comunicazione è sincrona, su una delle due linee (SCL) viene trasmesso il segnale di CLOCK che definisce la tempistica della trasmissione, e sull'altra linea (SDA) vengono inviati i singoli bit in maniera seriale. Essendo un'unica linea adibita alla trasmissione dei dati, la comunicazione potrà avvenire solamente in HALF-DUPLEX, e cioè in momenti differenti, o il master impegna la linea SDA per inviare dati allo slave, o viceversa.

Questo tipo di comunicazione viene spesso utilizzata per collegare svariati tipi di dispositivi (sensori, memorie, LCD) con distanze non elevate in quanto i segnali in gioco sono comunque segnali logici con bassi livelli di tensione e non di tipo differenziale, ma riferiti sempre alla GND che è comune tra i vari dispositivi.

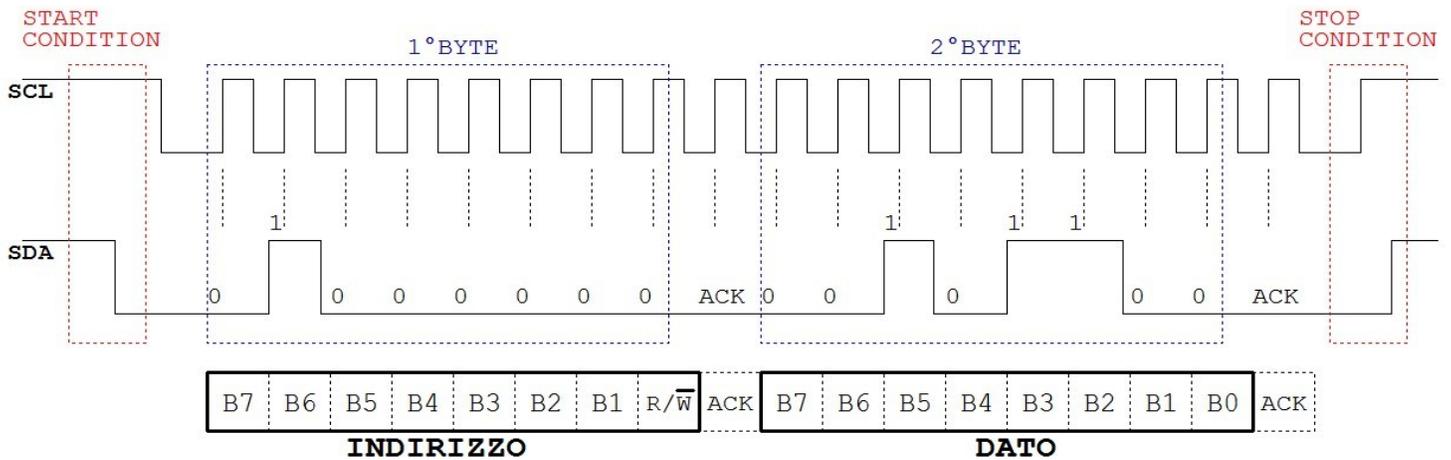
Rispetto ad una comunicazione parallela, la comunicazione seriale, offre sicuramente un minore impegno di linee di ingresso/uscita, pertanto nel settore dei microcontrollori, essa è sempre presente e spesso implementata al suo interno a livello hardware.



Con questo standard, è inoltre possibile organizzare una comunicazione MASTER-SLAVE o Multi-MASTER, come schematizzato nelle seguenti figure.



La trasmissione avviene secondo il seguente diagramma

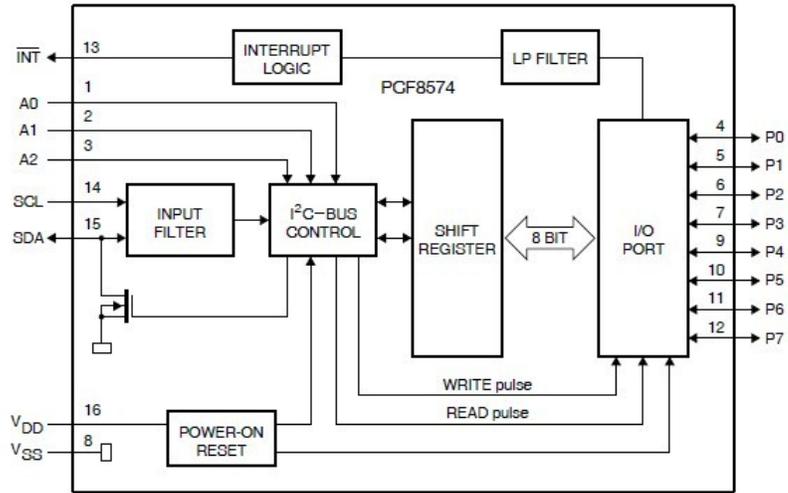
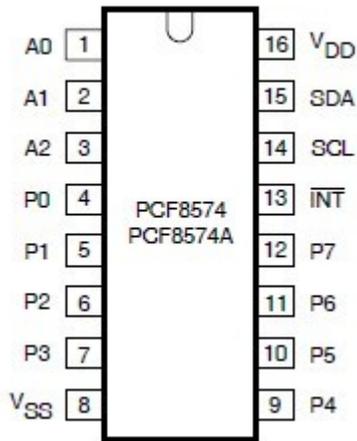


Le condizioni di START e STOP, sono identificate dal fronte di discesa di SDA con SCL=1 (Start) e dal fronte di salita di SDA con SCL=1 (Stop). Nel mezzo, vengono inviati i byte, il primo contiene l'indirizzo dello slave il secondo il dato da scrivere. Dopo ogni byte c'è un fronte di clock corrispondente al bit di acknowledgement inviato come risposta dallo SLAVE. La variazione dello stato della linea SDA durante l'invio dei bit, deve avvenire quando la linea SCL è a livello basso, per non essere identificata come condizione di START o STOP.

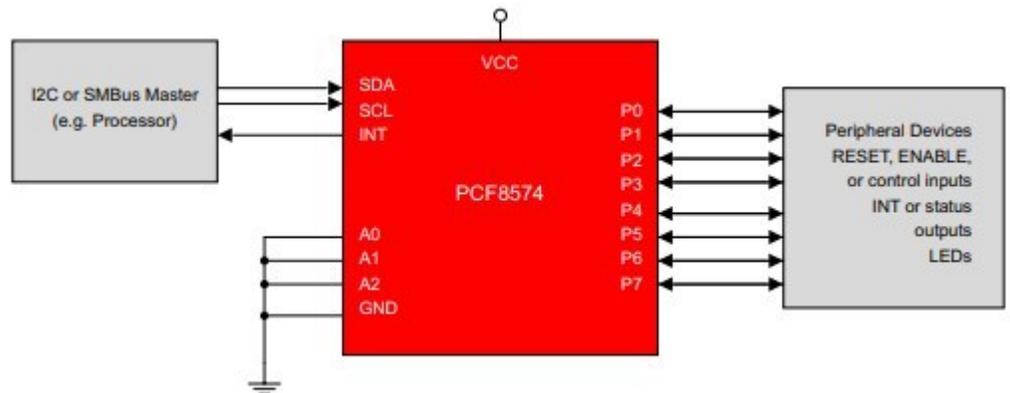
Il primo bit del byte dell'indirizzo, è dedicato alla selezione R/W (Read/Write) pertanto con i restanti 7 bit si potranno indirizzare 128 dispositivi.

Dopo questa rapida premessa andiamo a vedere il funzionamento di uno dei tanti dispositivi che funzionano sul bus I2C e cioè il GPIO expander (espansione di ingressi/uscite) PCF8574.

Questo integrato, si può trovare nel classico package di 16 pin come in figura.



Oltre alle 8 linee (GPIO) che possono essere utilizzate come ingresso o uscita, troviamo 3 linee A2,A1 e A0 con cui si assegna l'indirizzo del componente sul bus I2C.



C'è inoltre un segnale di ritorno dall'integrato al microcontrollore, che può essere utilizzato come interrupt.

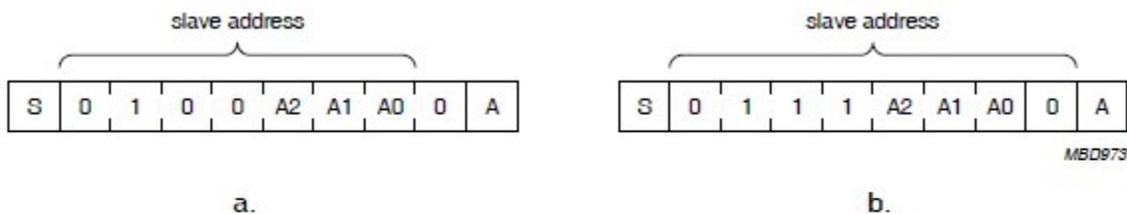
Se ad esempio dovessimo collegare 8 pulsanti o 8 sensori sulle 8 linee P0-P7, il microcontrollore dovrebbe testare ognuna di queste linee periodicamente per verificarne lo stato, questa modalità di test, viene definita "polling", ed impegna periodicamente il microcontrollore, non consentendo inoltre un controllo continuo di queste linee, in quanto il test avviene ad intervalli definiti.

Se ad esempio il microcontrollore leggesse le linee di ingresso una volta al secondo, potremmo non accorgerci di eventuali rapide variazioni di uno o più ingressi.

Per evitare questa problematica e per non impegnare il microcontrollore, si potrebbe pertanto utilizzare la linea di interrupt dell'integrato 8574 collegata al micro.

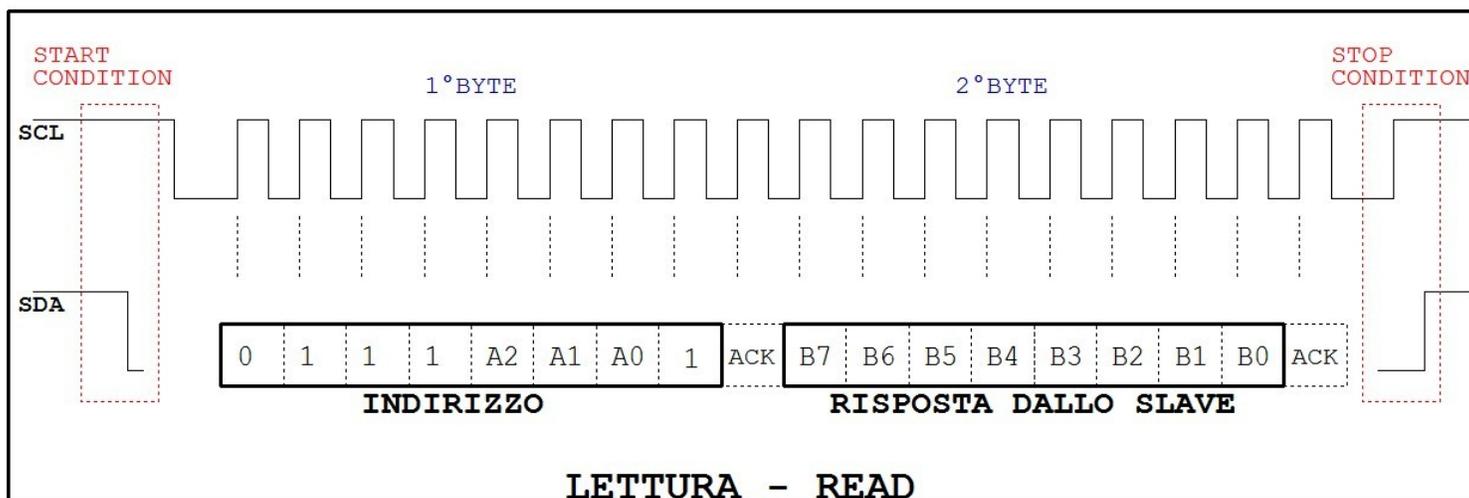
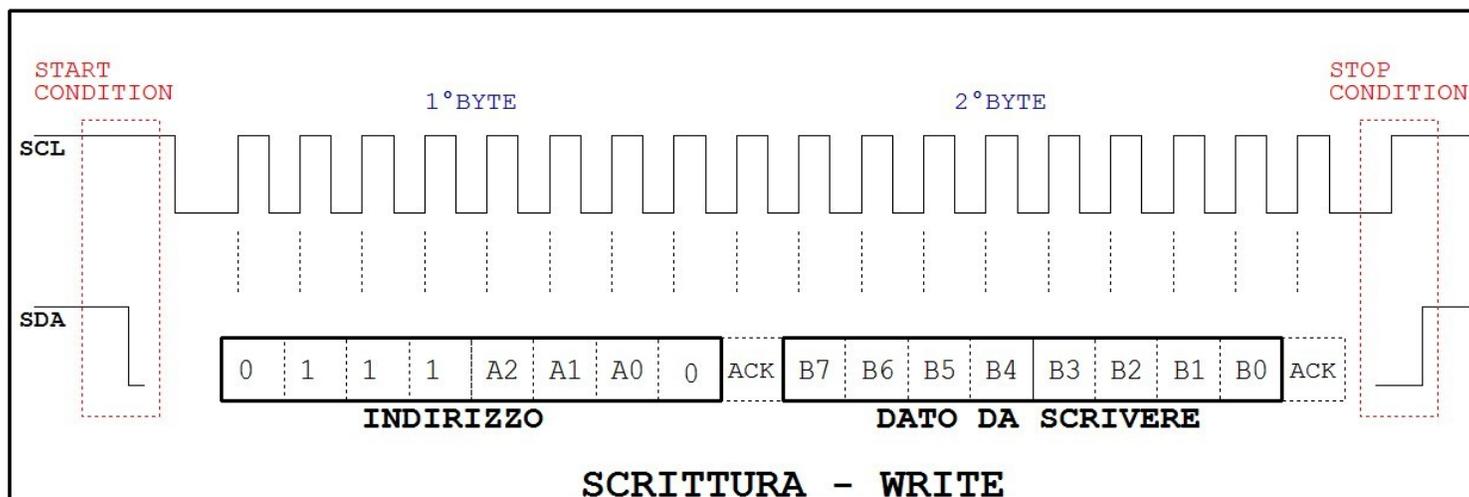
In questo caso questa linea verrebbe attivata solo in presenza di un segnale sulle linee di ingresso P0-P7, e collegando questa linea all'ingresso di interrupt del microcontrollore, una sua attivazione interromperebbe momentaneamente il programma per far eseguire al micro la relativa routine di interrupt, dove presumibilmente si andrà a leggere lo stato delle 8 linee.

L'indirizzamento di questo dispositivo avviene secondo la seguente struttura.



(a) PCF8574.
(b) PCF8574A.

C'è una prima parte di 4 bit fissa che vale 4 o 7 a seconda del modello 8574 oppure 8574A, ed una seconda parte dove inserire il valore delle 3 linee A2,A1 ed A0. L'ultimo bit a destra (bit0) viene messo a 0 se vogliamo scrivere o ad 1 se vogliamo leggere.

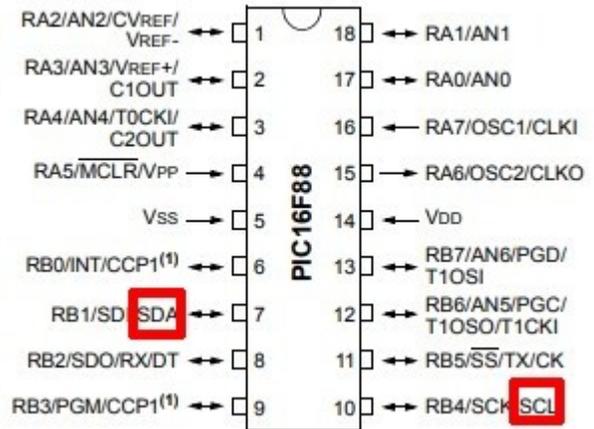


Nei micro viene implementato spesso un circuito hardware per gestire questo tipo di comunicazione, ad esempio nei microcontrollori PIC, ci sono già delle linee dedicate.

Nel caso del PIC16F88, però l'hardware interno, consente il funzionamento del microcontrollore solo come SLAVE e non come MASTER.

Nell'ambiente di sviluppo della mikroelettronika, MikroC, sono presenti delle librerie che consentono l'utilizzo della libreria hardware (in questo caso solo come SLAVE) o della libreria software che invece ci consentirà di utilizzare il micro anche come MASTER.

Andiamo ora a vedere il funzionamento di questa libreria.



The screenshot shows the MikroC IDE with a C program for I2C communication. The code is as follows:

```

//indirizzo dell'8574 0x70 per scrivere
//indirizzo dell'8574 0x71 per leggere

sbit Soft_I2C_Scl      at RB0_bit;
sbit Soft_I2C_Sda     at RB1_bit;
sbit Soft_I2C_Scl_Direction at TRISB0_bit;
sbit Soft_I2C_Sda_Direction at TRISB1_bit;

char input;

void main() {
    Soft_I2C_Init();           //inizializzazione libreria
    Soft_I2C_Start();         //inizio scrittura I2C
    Soft_I2C_Write(0x70);     //scrivo l'indirizzo dello slave in scrittura
    Soft_I2C_Write(0x00);    //scrivo il dato
    Soft_I2C_Stop();          //fine scrittura I2C

    Soft_I2C_Start();
    Soft_I2C_Write(0x71);     //scrivo l'indirizzo dello slave in lettura
    input=Soft_I2C_Read(0);   //leggo il dato no acknowledge
    Soft_I2C_Stop();          //fine lettura I2C
}
    
```

Annotations in the image include:

- Definizione dei bit SDA e SCL**: A red box highlights the bit definitions for SDA and SCL.
- scrittura**: A red label points to the I2C write sequence.
- lettura**: A red label points to the I2C read sequence.
- ATTIVARE LA LIBRERIA**: A red arrow points to the 'Software_I2C' library in the Library Manager.

Nell'immagine sopra sono riassunte le funzioni principali:

- Soft_I2C_Init()** - per inizializzare la libreria secondo la definizione dei bit fatta inizialmente
- Soft_I2C_Start()** - Genera la condizione di start
- Soft_I2C_Stop()** - Genera la condizione di stop
- Soft_I2C_Write(byte)** - Invia il byte sul bus
- Soft_I2C_Read(ack)** - Riceve e ritorna il valore letto sul bus, se ack=1 aspetta anche l'acknowledge

Nel programma di esempio sopra sono inoltre evidenziate le due sezioni per la scrittura e lettura, dove tra la condizione di start e quella di stop, si trovano i due byte e cioè l'indirizzo ed il dato.

A livello hardware leggendo il datasheet, possiamo inoltre notare che la corrente erogata sulle linee di uscita è molto bassa circa 1mA, mentre invece quella assorbita è più grande e cioè 25mA, pertanto è consigliabile collegare eventuali carichi in uscita verso la linea di alimentazione VCC, attivandoli così con un livello 0 logico.

Di seguito uno schema di collegamento di esempio, tra un PIC16F88 ed un integrato PCF8574, con un solo LED collegato sulla linea di uscita P4 (pin 9) da notare sulle due linee SDA e SCL, due resistenze di pull-up da 10k, in quanto le linee del bus I2C, per evitare conflitti hardware, sono open collector. In questo modo ogni dispositivo potrà al massimo collegare la linea verso il potenziale GND.

