

# Basic per PIC

```
File Edit Compile Options Help
[Icons]
1
2     DEVICE 16F84           * Microcontroller PIC16F84
3
4     INCLUDE I2CBUS        * Routine I2C
5
6     INIT I2CBUS           * Inizializzazione del bus
7
8
9     BSTART                * Condizione di inizio per la comunicazione
10    * I2C
11
12    BSTOP                 * Condizione di arresto
13
14
PIC-BASIC COMPILED OK 96 Words used.
```

L'ultima istruzione sarà quella della condizione di stop.

```
File Edit Compile Options Help
[Icons]
1
2     DEVICE 16F84           * Microcontroller PIC16F84
3
4     INCLUDE I2CBUS        * Routine I2C
5
6     INIT I2CBUS           * Inizializzazione del bus
7
8
9     BSTART                * Condizione di inizio per la comunicazione
10    * I2C
11
12    BUSOUT ($11010000)    * Indirizzo dello slave e ordine
13    * di scrittura
14    BUSOUT (0)            * Invio del valore 0
15
16    BSTOP                 * Condizione di arresto
17
18
PIC-BASIC COMPILED OK 102 Words used.
```

Il primo BSOUT serve per indirizzare lo "slave".

poco dopo quella di stop, quindi deve esistere un sistema per fare in modo che i "master" non cerchino di realizzare una comunicazione se nel frattempo il bus è già occupato da un altro.

## Configurazione del canale

Così come succedeva con le istruzioni per la comunicazione seriale con un computer,

quelle di comunicazione con il bus I2C sono progettate per il lavoro con il PIC16F84. Questo PIC non dispone di risorse hardware dedicate a questo scopo, di conseguenza la comunicazione viene fatta via software. I comandi INCLUDE e INIT serviranno per includere le routines di funzionamento, in questo caso del bus I2C, e per iniziarlo. Bisogna ricordare che le istruzioni I2C utilizzano piedini fissi, che sono il bit 0 della porta A come linea SDA e il bit 1 della porta A come linea SCL. Queste linee sono configurate automaticamente come uscite. Se la porta A si utilizza nello stesso programma per la comunicazione con il protocollo RS-232, il comando INIT I2CBUS deve essere successivo a INIT SERIAL.

## Istruzione di inizio

Dopo aver inizializzato il canale, per iniziare la comunicazione la prima istruzione sarà la condizione di inizio BSTART. Come si può vedere dalla figura, questa istruzione non ha parametri, quindi il suo utilizzo è molto semplice.

## Istruzione di stop

L'istruzione per la condizione di stop è simile alla precedente. È sufficiente porre BSTOP, senza parametri come mostrato nella figura. Verrà terminata in questo modo la comunicazione I2C. Per riattivarla bisogna nuovamente generare una condizione di inizio.

## Invio dei dati

L'istruzione per l'invio dei dati è BSOUT. L'unico parametro di cui ha bisogno, posto fra parentesi, può essere un numero, una variabile o un'espressione. Il primo byte che si invia corrisponde all'indirizzo dello "slave" con cui si vuole realizzare la comunicazione, e il bit meno significativo indica se si vuole scrivere (0), oppure leggere (1).

La quantità di byte inviati, il loro significato e il loro numero dipenderanno dal dispositivo con cui si sta realizzando la comunicazione.

```

1 LOOP:
2   Bstart           * Condizione di inizio
3   Busout ($11010000) * Slave da scrivere
4   Busout (1)       * Posizionamento sul registro 1
5   Busout (Minutes) * Scrittura di diversi registri
6   Busout (Hours)
7   Busout (Day)
8   Busout (Date)
9   Busout (Month)
10  Busout (Year)
11  Bstop
12  Return           * Condizione di arresto
13
14
15
16
17
18
    
```

Il significato dei byte dipenderà dal dispositivo.

## Ricezione dei dati

Con l'istruzione BUSIN si riceve un dato del bus. È necessario definire una variabile sulla quale scrivere questo dato per il suo successivo trattamento. Inoltre, in precedenza, è stato inviato un byte per indirizzare lo "slave" e informarlo che si vuole realizzare una lettura.

In uno stesso programma ci possono essere diverse istruzioni di inizio e di stop, di fatto sarà necessario chiudere una comunicazione e tornare ad aprirla quando si desidera passare da scrittura a lettura o viceversa.

```

1
2 DEVICE 16F84           * Microcontroller PIC16F84
3
4 INCLUDE I2CBUS        * Routine I2C
5
6 DIM bus              * Variabile per acquisire il dato
7
8 INIT I2CBUS          * Inizializzazione del bus
9
10
11 BSTART              * Condizione di inizio per la comunicazione
12 I2C
13
14 BUSOUT ($11010000)  * Indirizzo dello slave e ordine
15 di scrittura
16 BUSOUT (0)          * Invio del valore 0
17 BSTOP              * Condizione di arresto
18
19 BSTART              * Nuova condizione di inizio
20 BUSOUT ($11010001)  * Nuovo indirizzo e ordine di lettura
21 bus=BUSIN           * Si legge sulla variabile bus
22 BSTOP              * Condizione di fermata
    
```

PIC-BASIC COMPILED OK 110 Words used.

Esempio di ricezione dei dati.