

Comunicazione mediante il bus I2C

Nei capitoli precedenti abbiamo parlato delle caratteristiche della comunicazione I2C sotto diversi punti di vista. Il LetPicBasicPlus così come il LetPicBasicLite, fornisce una serie di istruzioni specifiche per il lavoro con i dispositivi I2C. Vediamo ora di conoscere queste istruzioni con un esempio concreto di comunicazione.

Passi della comunicazione

La comunicazione I2C è caratterizzata dal fatto di seguire delle fasi molto definite, quali la generazione della "condizione di inizio", l'indirizzamento dello slave, l'invio o la ricezione di dati e il termine della comunicazione. Nel LetPicBasicLite avevamo istruzioni specifiche per ogni cosa, nel LetPicBasicPlus invece le istruzioni si riducono a due, senza che per questo il concetto cambi.

Trasmissione o invio di dati

La trasmissione da un PIC a un dispositivo I2C si risolve con l'istruzione BUSOUT. Il primo parametro è un valore di controllo. Poi ci può essere in modo opzionale un indirizzo (posizione all'interno del dispositivo), e in ultimo una o più

```

FILE  EDIT  COMPILER  OPTIONS  HELP
[Icons]
[Icons]

DEVICE 16F877

DIM control, dir, var1 as BYTE

'BUSOUT Control, (Indirizzo), [Variabile (,Variabile...)]

BUSOUT control, dir, [var1]

PICBASIC PLUS COMPILED OK. 98 Words used
29 Variables used in the 16F877 from a possible 368
    
```

Esempio dell'utilizzo dell'istruzione BUSOUT.

```

FILE  EDIT  COMPILER  OPTIONS  HELP
[Icons]
[Icons]

DEVICE 16F877

DIM var1 as BYTE
DIM dir1, var2 as WORD
SYMBOL control %10100000

dir1=00
var1=200

BUSOUT control, dir1, [var1]

var2=990

BUSOUT control, dir1, [var1, var2]

BUSOUT control, dir1, ["Ciao mondo", var1, var2]

PICBASIC PLUS COMPILED OK. 166 Words used
30 Variables used in the 16F877 from a possible 368
    
```

Invio di diversi dati.

variabili di dato. Il parametro di controllo può essere una costante o un'espressione, però con la dimensione di un byte. I sette bit più significativi fanno riferimento all'indirizzo dello slave con cui si vuole comunicare, e il bit 0 indica se ciò che si vuole fare su questo dispositivo è una lettura o una scrittura. Nella trasmissione, il valore di questo primo bit dovrà essere 0, per realizzare una scrittura. Nell'esempio alla pagina precedente, si riporta prima la sintassi dell'istruzione in modo generale, e poi si sostituiscono i parametri con variabili specifiche precedentemente definite, anche se non si specifica che valori hanno queste variabili.

Questo valore potrebbe essere fisso, derivare da qualche calcolo, oppure essere inserito da un utente esterno che potrebbe realizzare diverse operazioni con differenti dispositivi. Come si può osservare nel secondo esempio di invio di dati, l'indirizzo interno può contenere fino a 16 bit nel caso fosse necessario, questo dipenderà dal dispositivo slave con cui stiamo comunicando. Le variabili che si inviano, a loro volta, possono avere dimensione byte o word, inoltre è possibile inviare catene di caratteri direttamente inserite fra parentesi quadre. Se i dati inviati sono diretti tutti allo stesso slave è sufficiente una sola istruzione BUSOUT, e sarà il compilatore che la adatterà al tipo di comunicazione reale.

Ricezione dei dati

Per la ricezione dei dati si utilizza l'istruzione BUSIN. Essa ha due modalità, riportate

```
File Edit Compile Options Help
[Icons]
DEVICE 16F877
DIR control, dir, var1 as BYTE
'Variabile= BUSIN Control, (Indirizzo)
var1= BUSIN control, dir
PICBASIC PLUS COMPILED OK. 123 Words used
29 Variables used in the 16F877 from a possible 368
```

Utilizzo dell'istruzione BUSIN per la ricezione di un dato singolo.

```
File Edit Compile Options Help
[Icons]
DEVICE 16F877
DIR control, dir, var1, var2 as BYTE
'BUSIN Control, (Indirizzo), [Variabile {, variabile...}]
BUSIN control, dir, [var1, var2]
PICBASIC PLUS COMPILED OK. 126 Words used
30 Variables used in the 16F877 from a possible 368
```

Utilizzo dell'istruzione BUSIN per la ricezione di diversi dati.