

## Controllo dei tempi

**A**nche se abbiamo cambiato il compilatore, il controllo dei tempi continua a essere necessario per la realizzazione dei nostri programmi, dato che la destinazione finale continua a essere la stessa: un microcontroller. Tuttavia, nel LetPicBasicPlus il modo di controllare i tempi è cambiato rispetto al LetPicBasicLite, conservando alcune delle istruzioni, facendone sparire altre e creandone alcune nuove. Vediamo alcuni esempi di tutto questo.

### Generazione di "delays"

Nel LetPicBasicPlus, per la generazione dei "delays" o "perdite di tempo", si lavora con due delle tre istruzioni di cui disponeva già il LetPicBasicLite: DELAYUS e DELAYMS.

In entrambe, il parametro da indicare è un numero il cui valore massimo è 65535, che in un caso significherà microsecondi e in un altro millisecondi. Questo numero si può fornire come una costante, una variabile o un'espressione. Tuttavia, ci sono diverse variazioni rispetto alle istruzioni già viste. La prima cosa diversa è il limite, che è sostanzialmente maggiore, anche se come nel LetPicBasicLite bisogna controllarlo, soprattutto se arriva dall'esterno, dato che l'errore si produrrà al momento dell'esecuzione. Inoltre non è necessario introdurre

```

File Edit Compile Options Help
[Icons]
DECLARE XTAL 20 'Quarzo da 20 MHz
DIM VAR as BYTE
VAR=30
DELAYUS 8 'Tempo di 8 microsecondi
DELAYUS VAR 'Tempo di 30 microsecondi
DELAYUS VAR+8 'Tempo di 38 microsecondi
DELAYMS 8 'Tempo di 8 millisecondi
DELAYMS VAR 'Tempo di 30 millisecondi
DELAYMS VAR+8 'Tempo di 38 millisecondi

PICBASIC PLUS COMPILED OK. 62 Words used
27 Variables used in the DEFAULT 16F84 from a possible 68
    
```

### Esempio dell'utilizzo delle istruzioni delayus e delayms.

```

File Edit Compile Options Help
[Icons]
DECLARE XTAL 4 'Quarzo da 4 MHz
DIM VAR as BYTE
VAR=2
DELAYUS 2 'Tempo di 2 microsecondi
'Parametro corretto
DELAYUS VAR 'Tempo di 2 microsecondi
'Non è possibile con un quarzo
'da 4 MHz

PICBASIC PLUS COMPILED OK. 30 Words used
27 Variables used in the DEFAULT 16F84 from a possible 68
    
```

### Restrizioni nell'utilizzo dell'istruzione delays.

il parametro fra parentesi, e l'espressione si può inserire direttamente nell'istruzione senza la necessità di farlo a parte. Un cambiamento importante è che la frequenza di funzionamento del microcontroller si può definire con la direttiva DECLARE. In questo modo, dopo aver dichiarato la velocità, non dovremo più realizzare i calcoli per sapere a quanto corrisponderà il valore passato all'istruzione come parametro. Nell'esempio il quarzo è da 20 MHz e si realizza un delay da, per esempio, 30 ms. Se invece di questo quarzo se ne dichiara un altro da 4 MHz, le istruzioni continueranno a essere le stesse, e sarà lo stesso compilatore che si incaricherà di generare il codice macchina adeguato. Esiste solamente una restrizione nell'uso dell'istruzione DELAYUM, ed è relativa al parametro nel caso in cui venga fornito sotto forma di variabile. Quando il tempo si specifica direttamente, il suo valore minimo potrà essere di 1 microsecondo, però se viene utilizzata una variabile, bisognerà rispettare un valore minimo che dipenderà in ogni caso dall'oscillatore usato, così come riportato nella tabella allegata. Tuttavia, se non si rispettano questi valori non si produrrà un errore, anche se le temporizzazioni ottenute non saranno corrette.

FREQUENZA DEL QUARZO (in MHz)	MINIMO DELAY (in microsecondi)
4	24
8	12
10	8
16	5
20	2

Valori minimi di delay a seconda del quarzo utilizzato se il parametro è una variabile.

```

File Edit Compile Options Help
[Icons]
DDI  VAR aa BYTE
VAR=60
SYMBOL LED=PORTB.0 'Dichiarazione di un LED sul bit 0 della PORTB
LOOP: HIGH LED 'Si accende il LED
      SLEEP 60 'Rimane in riposo per 1 minuto
      LOW LED 'Si spegne il LED
      SLEEP VAR 'Un altro minuto di riposo
      GOTO LOOP 'Inizia nuovamente
PICBASIC PLUS COMPILED OK. 66 Words used
27 Variables used in the DEFAULT 16184 from a possible 68
    
```

Utilizzo dell'istruzione "sleep" in un programma.

```

File Edit Compile Options Help
[Icons]
DDI  VAR aa BYTE
VAR=6
SYMBOL LED=PORTB.0 'Dichiarazione di un LED sul bit 0 della PORTB
LOOP: HIGH LED 'Si accende il LED
      SNOOZE 6 'Rimane in riposo 1152 ms
      LOW LED 'Si spegne il LED
      SNOOZE VAR 'Un altro intervallo uguale di riposo
      GOTO LOOP 'Inizia nuovamente
PICBASIC PLUS COMPILED OK. 31 Words used
27 Variables used in the DEFAULT 16184 from a possible 68
    
```

Utilizzo dell'istruzione "snooze" in un programma.

## Istruzioni di "riposo"

Anche qui disponiamo di istruzioni per portare il microcontroller in uno stato chiamato "di riposo", con il conseguente risparmio di

energia. La prima di queste è l'istruzione SLEEP, che abbiamo già visto nel LetPicBasicLite.

Il parametro sarà una costante o una variabile fra 1 e 65535, che si suppone il tempo in secondi in cui

il microcontroller rimarrà "addormentato". Nell'esempio possiamo vedere l'intermittenza di un diodo LED collegato al bit 0 della porta B, fra l'accensione e lo spegnimento sono state