

ingresso, su che variabile verrà memorizzato il totale, e il periodo di tempo da tenere in considerazione. Questo periodo si misura in millisecondi. L'istruzione COUNTER testerà il piedino di ingresso durante questi millisecondi a una frequenza che dipenderà dall'oscillatore collegato al PIC.

In questo modo se si sta lavorando con un quarzo da 4 MHz, il pin verrà testato ogni 20 μ s, mentre se si lavorerà con un quarzo da 20 MHz il controllo verrà eseguito ogni 4 μ s. Un'altra istruzione che può avere lo stesso aspetto della precedente, ma che serve per un'altra cosa, è PULSIN. In questo caso l'istruzione attende un impulso di ingresso, che può essere alto (passaggio da 0 a 1) o basso (passaggio da 1 a 0), e da quel momento inizia a contare il tempo. Quando il piedino torna al livello iniziale (termina l'impulso) l'istruzione carica sulla variabile il valore contato. PULSIN attende un massimo di 0,65535 secondi l'arrivo dell'impulso, se in questo tempo non arriva alcun impulso sul pin, restituisce il valore 0, e il programma continua con il resto delle istruzioni. Se si utilizza una variabile di tipo BYTE per memorizzare il tempo, invece di una di tipo word, PULSIN restituirà solamente gli 8 bit meno significativi del valore del tempo misurato, questo può generare un errore quando i valori superano gli 8 bit. Per contro esistono anche le istruzioni di generazione di impulsi. La PULSOUT funziona in modo opposto alla PULSIN, producendo un impulso di una

```
File Edit Compile Options Help
[Icons] [Run] [Stop] [Refresh] [Print] [Save] [Open] [Close]

DEVICE 16F877           'Si definisce il PIC da utilizzare
DIM A as WORD          'Variabile per contare il tempo
TRISB = %11111111     'Si definisce la PortaB come ingresso
SYMBOL PUL = PORTB.0  'Pulsante di ingresso degli impulsi
A = PULSIN PUL, 1     'Su A si acquisisce la durata dell'impulso
END

PICBASIC PLUS COMPILED OK. 56 Words used
28 Variables used in the 16F877 from a possible 368
```

Esempio di utilizzo dell'istruzione PULSIN.

```
File Edit Compile Options Help
[Icons] [Run] [Stop] [Refresh] [Print] [Save] [Open] [Close]

DEVICE 16F877           'Si definisce il PIC da utilizzare
TRISB = %00000000     'Si definisce la PortaB come uscita
SYMBOL PUL = PORTB.0  'Piedino per la generazione degli impulsi
PULSOUT PUL, 100, HIGH 'Si genera un impulso alto su PORTB.0
END

PICBASIC PLUS COMPILED OK. 54 Words used
26 Variables used in the 16F877 from a possible 368
```

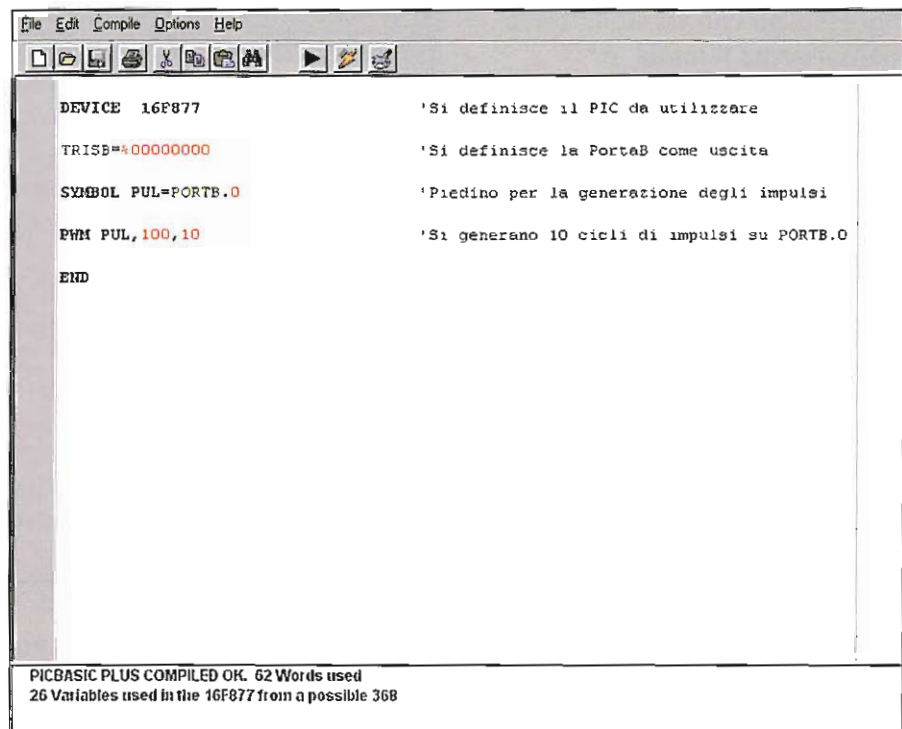
Esempio di utilizzo dell'istruzione PULSOUT.

durata specificata su di un piedino. Se non viene specificato il terzo parametro dell'istruzione, il valore iniziale dell'impulso sarà l'inverso di quello del piedino al momento dell'esecuzione dell'istruzione. Altrimenti è possibile specificare

il tipo di impulso, sia con i valori (0 -1) che con il nome (LOW-HIGH). Ancora una volta l'oscillatore collegato determinerà il numero da impostare per determinare la durata dell'impulso. Quindi se si utilizza un quarzo da 4 MHz,

il valore che si imposterà nell'istruzione (in microsecondi) dovrà essere moltiplicato per 10 per calcolare il tempo totale; questo nell'esempio della figura darebbe una durata di 1 ms. Se il quarzo fosse da 20 MHz, il valore si moltiplicherebbe per 2, operazione che in questo esempio porterebbe a un risultato di 200 μ s. Se invece di un solo impulso si desidera generare diversi impulsi consecutivi, cioè un treno di impulsi, nel LetPicBasicPlus ci sono due istruzioni. Entrambe seguono lo stesso concetto, però di solito vengono utilizzate con dispositivi differenti. PWM genera un treno di X cicli di impulsi, tanti come sono specificati nel terzo parametro dell'istruzione, con un'ampiezza dell'impulso positivo specificata dal secondo parametro. Sia l'ampiezza che il numero di cicli può variare fra 0 e 255. Se il valore dell'ampiezza è zero, sul piedino selezionato avremo come uscita un livello basso continuo, mentre se è 255, il piedino rimarrà a 1.

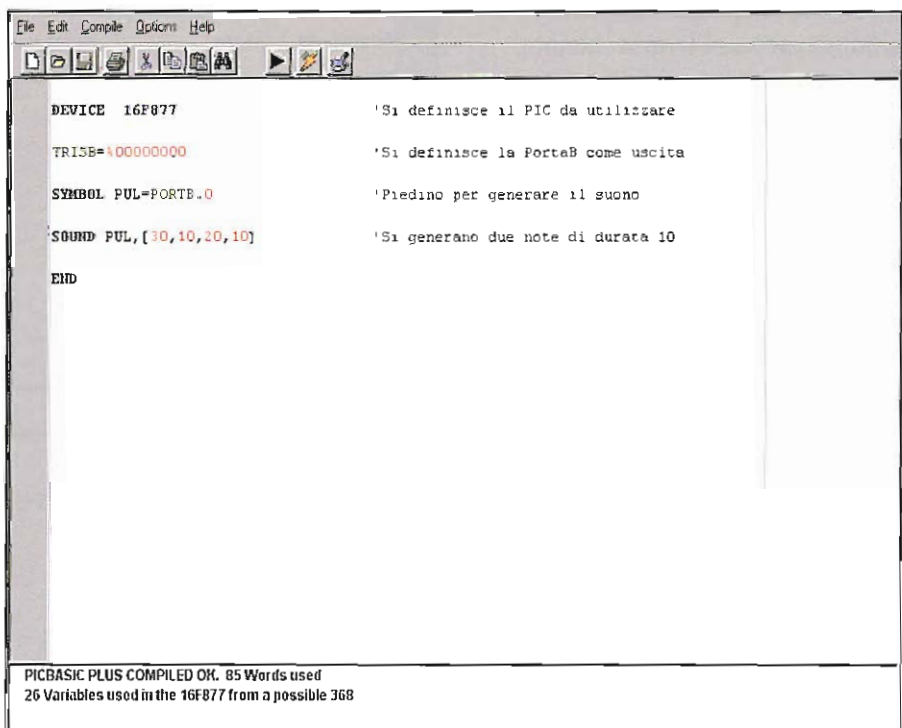
Normalmente si utilizza nel controllo dei motori, per variare la loro velocità cambiando l'ampiezza dell'impulso. In ultimo abbiamo l'istruzione SOUND, che ha lo stesso nome di quella esistente nel LetPicBasicLite, però è utilizzata con altri parametri e con altri scopi. Il suo compito è fornire suoni formati dalla coppia frequenza (o tono) - durata. Il primo parametro sarà il piedino sul quale si otterrà il suono, e il resto la coppia di dati di cui abbiamo detto in precedenza. Collegando il piedino in questione a un altoparlante, potremo ascoltare le melodie.



```
File Edit Compile Options Help
[Icons]
DEVICE 16F877           'Si definisce il PIC da utilizzare
TRISB=%00000000       'Si definisce la PortaB come uscita
SYMBOL PUL=PORTB.0    'Piedino per la generazione degli impulsi
PWM PUL,100,10         'Si generano 10 cicli di impulsi su PORTB.0
END

PICBASIC PLUS COMPILED OK. 62 Words used
26 Variables used in the 16F877 from a possible 368
```

L'istruzione PWM di solito si utilizza per il controllo dei motori.



```
File Edit Compile Options Help
[Icons]
DEVICE 16F877           'Si definisce il PIC da utilizzare
TRISB=%00000000       'Si definisce la PortaB come uscita
SYMBOL PUL=PORTB.0    'Piedino per generare il suono
SOUND PUL,[30,10,20,10] 'Si generano due note di durata 10
END

PICBASIC PLUS COMPILED OK. 85 Words used
26 Variables used in the 16F877 from a possible 368
```

Con SOUND si possono comporre melodie e ascoltarle tramite l'altoparlante.