Immaginate la situazione in cui mandiamo a un LED un 1 (lo accendiamo), uno 0 (lo spegniamo), un 1 (lo accendiamo), uno 0 (lo spegniamo) ecc. Ovviamente, in questo modo facciamo lampeggiare il LED. La velocità con cui lampeggia dipende dal secondo parametro che si può specificare. Anche in questo caso a seconda del linguaggio che si sta utilizzando sarà possibile controllare la lunghezza del periodo a "uno" (tempo durante il quale la linea è a 1) rispetto a una lunghezza fissa di "zero", la lunghezza del tempo a "uno" rispetto a una lunghezza fissa del periodo (somma del tempo a "uno" e a "zero") o qualsiasi combinazione che faccia variare la relazione fra il tempo a "uno" rispetto a quello a "zero"; il risultato sarà sempre lo stesso.

Notate che negli esempi delle figure è stato utilizzato un metodo "manuale". Se confrontiamo i due cicli, vedremo che, mantenendo la stessa proporzione fra la durata del livello "uno" e del livello "zero", varierà il periodo con il risultato che il lampeggio del LED sarà più lento. Questo si può fare introducendo l'istruzione "delayms" fra l'accensione e lo spegnimento del LED, in quanto l'unica cosa che fa è "perdere tempo". È anche possibile ridurre il valore che si pone come parametro, aumentando in questo modo la velocità del lampeggio.

Come caso estremo potremmo eliminare questa istruzione: in questo caso anche se il programma continua a far cambiare lo stato del LED il nostro occhio non percepirà più la variazione, e lo vedremmo sempre acceso. Il secondo esempio può risultare più difficile da capire se non si esegue una verifica pratica; ciò che viene fatto è variare la lunghezza dello stato a 1 mantenendo il periodo costante. In questo modo se tutto il periodo sarà occupato dal valore 1 torneremo nello stesso caso estremo dell'esempio precedente: sempre acceso. Se invece diminuiamo la lunghezza dello stato 1 a favore dello stato 0 la velocità si riduce. Questa tecnica è conosciuta come PWM (modulazione di ampiezza di impulsi) e si utilizza spesso per pilotare a differenti velocità i motori a corrente continua. Sino ad ora abbiamo utilizzato istruzioni note per controllare la velocità; utilizziamo adesso un'istruzione del LetPicBasic che rende più semplice questo controllo, molto utilizzata in numerose periferiche di uscita: "sound". Questa nuova istruzione manda un treno di impulsi sul pin

```
File  Edit  Compile  Options  Help

1
2         Device  16F84                  ' Si definisce il PIC16F84
3
4         Dim I
5         Define  PORTB=$00000000        ' Si dichiara la PortaB come uscita
6         Symbol  LED=B.0                ' Si assegna un nome al bit 0 della portaB
7
8         clear   LED                    ' Si inizializza il LED allo stato spento
9   FOR I=1 TO 20
10        set LED                        ' Si accende
11        delayms (30)                   ' Si attende un determinato tempo
12        clear LED                      ' Si spegne
13        delayms (30)                   ' Si attende un determinato tempo
14        NEXT I                         ' Si ripete il ciclo 20 volte
15
16  FOR I=1 TO 20
17        set LED                        ' Si accende
18        delayms (40)                   ' Si attende un determinato tempo
19        clear LED                      ' Si spegne
20        delayms (20)                   ' Si attende un determinato tempo
21        NEXT I                         ' Si ripete il ciclo 20 volte
22
23  END
PIC-BASIC COMPILED OK  64 Words used.
```

Variazione della lunghezza del parametro 1.

```
File  Edit  Compile  Options  Help

1
2         Device  16F84                  ' Si definisce il PIC16F84
3
4         Define  PORTB=$00000000        ' Si dichiara la PortaB come uscita
5         Symbol  LED=B.0                ' Si assegna un nome al bit 0 della portaB
6
7
8         Sound (60, 20, LED)
9         Sound (80, 20, B.0)
10
11  END
12
13
14
15
16
17
18
PIC-BASIC COMPILED OK  35 Words used.
```

Istruzione per il controllo della velocità.

specificato come terzo parametro. La velocità di variazione 1-0-1 la impone il primo parametro, con un valore che va da 0 a 255, il tempo durante il quale si produce questo treno di impulsi viene determinato dal secondo parametro, anch'esso con un range da 0 a 255. Come prima si può utilizzare un dispositivo come un LED o un motore a corrente continua, benché esista anche un'altra possibilità, indicata dal nome dell'istruzione: la generazione di suono. Le diverse velocità e durate possono produrre in un altoparlante differenti suoni. Se vi intendete un po' di musica non vi risulterà difficile fare delle prove con qualche valore, e cercare di comporre una canzone.

Scambio di valori fra registri

Come potrete vedere l'istruzione che ci interessa in questo caso potrà essere utilizzata in molte occasioni, si tratta di casi generali, ma che facilitano lo sviluppo di operazioni molto comuni nei microcontroller; l'utilizzo di questa terza istruzione tuttavia è meno frequente delle altre due. Non dovete preoccuparvi se non riuscite a vederne un'applicazione immediata, a volte non si comprende la reale potenza di un'istruzione sino a che non ci torna utile per risolvere un problema specifico. Stiamo parlando dell'istruzione "swap". Questa istruzione scambia il valore delle due variabili che sono fornite come parametri. Come si può vedere dall'esempio, se la variabile A all'inizio vale 5 e la variabile B vale 8 dopo aver eseguito

```
File  Edit  Compile  Options  Help

1
2         Device  16F84                    ' Si definisce il PIC16F84
3
4         Dim A, B
5
6         A=5
7         B=8
8
9         Swap A,B
10
11  END
12
13
14
15
16
17
18

PIC-BASIC COMPILED OK  15 Words used.
```

Scambio di valori fra due variabili con l'istruzione swap.

```
File  Edit  Compile  Options  Help

1
2         Device  16F84                    ' Si definisce il PIC16F84
3
4         Dim A, B, Aux
5
6         A=5
7         B=8
8
9         Aux=A
10        A=B
11        B=Aux
12  END
13
14
15
16
17
18
19

PIC-BASIC COMPILED OK  13 Words used.
```

Scambio del valore fra due variabili utilizzando una terza variabile di appoggio.

l'istruzione A varrà 8 e B varrà 5. Lo stesso programma potrebbe essere realizzato senza utilizzare questa istruzione, con l'aiuto di una variabile ausiliaria intermedia, ed è proprio ciò che si fa negli altri linguaggi quando non esiste questa istruzione. Un'applicazione di questo scambio di variabili potrebbe essere un sistema che sta acquisendo, ad esempio, valori di temperatura, dove si desideri anche ottenere, nell'arco della giornata, il valore massimo, quello minimo, la media, ecc.

Ogni nuovo valore che arriva viene comparato con le variabili che sino a quel momento contengono i predetti limiti e nel caso debbano essere aggiornati, si applica lo scambio di variabili. Notate che è necessario conservare il vecchio valore per ottenere una media, altrimenti sarebbe sufficiente eseguire un'assegnazione del nuovo valore.