Istruzioni generali

Abbiamo visto sinora diverse istruzioni raggruppate per la funzione che devono svolgere. Questa volta studieremo tre istruzioni che non hanno una relazione diretta con nessun'altra, ma che possono essere utili in diversi programmi. Le prime due introducono ed estraggono dati dal sistema, anche se in un modo molto particolare, la terza lavora con i dati che già si trovano nei registri.

Acquisizione di impulsi esterni

Supponiamo di voler controllare l'accesso a un'area avente una capienza limitata, in modo che arrivati al numero massimo si accenda un LED per segnalare questo stato, e impedire l'accesso a ulteriori persone. Potremmo realizzare il programma della prima figura, dove ogni volta che si rileva un 1 proveniente da un sensore significa che è passata una persona e che dobbiamo incrementare il contatore, accendendo il LED nel caso in cui questo incremento raggiunga il valore massimo permesso. Questo modo di incrementare "manualmente" un contatore non è ottimale per almeno due motivi: fa perdere tempo al processore, che mentre si occupa del contatore non può fare altre cose e, in linea teorica, potrebbe addirittura passare qualche persona senza essere rilevata; in effetti questa possibilità è piuttosto remota, dato che ci sono poche istruzioni, però mentre il processore sta eseguendo le altre istruzioni non controlla il sensore.

Per risolvere questo problema procediamo per gradi. Proviamo ora a utilizzare un'altra istruzione che voi già conoscete: l'istruzione "button".

Come ricorderete, con questa

```
File  Edit  Compile  Options  Help

1
2           Device   16F84                ' Si definisce il PIC16F84
3           Dim      TOT                 ' Su questa variabile si memorizzerà il
4                                        ' numero di persone che entrano
5           Dim      MAX                 ' Su questa variabile si memorizzerà il
6                                        ' numero massimo di persone che
7                                        ' può contenere l'area
8           Define   PORTB=%11111111     ' Si dichiara la PortaB come ingresso
9           Define   PORTA=%00000000     ' Si dichiara la PortaA come uscita
10          Symbol   LED=A.0             ' Si assegna un nome al bit 0 della porta A
11
12          TOT=0 : MAX=20
13          clear    LED                 ' Si Inizializza il LED allo stato spento
14 LOOP:    If PORTB&1=1 then TOT=TOT+1  ' Se il bit 0 della portaB vale 1
15                                       ' significa che sta entrando un'altra
16                                       ' persona e bisogna sommare 1 al contatore
17          If TOT=MAX then goto LUM     ' Se il massimo è stato raggiunto
18                                       ' bisogna accendere il LED
19          goto LOOP                    ' altrimenti si prosegue nel conteggio
20 LUM:     set LED
21
22 END

PIC-BASIC COMPILED OK  34 Words used.
```

Prima stesura del programma per controllare l'accesso a un'area.

```
File  Edit  Compile  Options  Help

1           Device   16F84                ' Si definisce il PIC16F84
2           Dim      TOT                 ' Su questa variabile si memorizzerà il
3                                        ' numero di persone che entrano
4           Dim      MAX                 ' Su questa variabile si memorizzerà il
5                                        ' numero massimo di persone che
6                                        ' può contenere l'area
7           Define   PORTB=%11111111     ' Si dichiara la PortaB come ingresso
8           Define   PORTA=%00000000     ' Si dichiara la PortaA come uscita
9           Symbol   LED=A.0             ' Si assegna un nome ai bit 0 delle
10          Symbol   INT=B.0             ' porte A e B
11
12          TOT=0 : MAX=20
13          clear    LED                 ' Si Inizializza il LED allo stato spento
14 LOOP:    Button   INT                 ' Il programma non continua sino a che il
15                                       ' bit 0 della portaB non passa a 1
16          Button   INT                 ' Per ogni persona ci sono due cambi di
17                                       ' livello
18          TOT=TOT+1                    ' Si incrementa il contatore
19          if TOT=MAX then goto LUM     ' Se il massimo è stato raggiunto
20                                       ' bisogna accendere il LED
21          goto LOOP                    ' altrimenti si prosegue nel conteggio
22 LUM:     set LED
23
24 END

PIC-BASIC COMPILED OK  35 Words used.
```

Con il secondo programma è meno probabile che ci "scappi" qualcuno.

istruzione il programma si ferma sino a quando il pin specificato cambia di valore. In questo modo ci sono meno possibilità che ci "scappi" qualcuno, dato che il programma rimane fermo sino a quando non arriva qualcuno, e in quel momento esegue le operazioni di somma e di verifica. Dato che si attende un cambio di livello, e il sensore collegato al sistema fornisce due variazioni consecutive per persona, bisognerà utilizzare due istruzioni di questo tipo per fare in modo che il contatore si incrementi di una sola unità per volta. In questo modo però, non abbiamo risolto il problema rappresentato dal fatto che il processore non può dedicarsi ad altre cose, al contrario lo abbiamo aggravato, dato che passa la maggior parte del tempo ad attendere che entri qualcuno. L'utilizzo dell'istruzione "counter", così chiamata perché si avvale di un contatore "automatico", risolve entrambi i problemi contemporaneamente. La prima cosa che dobbiamo fare è abilitare il contatore con il parametro "on", con il quale inizializziamo anche il contatore a 0, poi decidiamo se si deve incrementare con il passaggio da 0 a 1 (utilizzando come secondo parametro "low") oppure da 1 a 0 (utilizzando come secondo parametro "high"). Per fermare il contatore il parametro sarà "off".

Quello che non possiamo scegliere, con questo contatore, è il piedino di ingresso a cui collegare gli impulsi che faranno incrementare il contatore stesso, dato che è obbligatoriamente il pin RA4. Dovremo quindi collegare il sensore di passaggio a questo pin. Notate che se fosse necessario potremmo passare il tempo

```
File  Edit  Compile  Options  Help

1
2           Device   16F84                  ' Si definisce il PIC16F84
3           Dim      TOT                    ' Su questa variabile si memorizzerà il
4                                           ' numero di persone che entrano
5           Dim      MAX                    ' Su questa variabile si memorizzerà il
6                                           ' numero massimo di persone che
7                                           ' può contenere l'area
8           Define   PORTA=%11111111        ' Si dichiara la PortaA come ingresso
9           Define   PORTB=%00000000        ' Si dichiara la PortaB come uscita
10          Symbol   LED=B.0                ' Si assegna un nome ai bit 0 della
11                                          ' porta B
12          TOT=0 : MAX=20
13          clear    LED                    ' Si Inizializza il LED allo stato spento
14          Counter ON,High                 ' Si inizializza il contatore
15  LOOP:   TOT=Counter                     ' Si prende il valore del contatore
16          if TOT=MAX then goto LUM        ' Se il massimo è stato raggiunto
17                                          ' bisogna accendere il LED
18          goto LOOP                       ' altrimenti si prosegue nel conteggio
19  LUM:    set LED
20
21  END
22
23
PIC-BASIC COMPILED OK   20 Words used.
```

Programma risolto con l'istruzione "counter".

```
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 (40)                    ' 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 velocità di lampeggio di un LED.

sorvegliando il contatore, ma non saremmo comunque noi a farlo incrementare, il processo è automatico. Non sarà nemmeno necessario utilizzare due volte la stessa istruzione, come nel caso precedente, visto che abbiamo anche specificato il tipo di variazione di livello a cui il contatore risponde. Per verificare il valore del contatore lo dobbiamo assegnare a una variabile, e poi lavorare con quest'ultima.

Generazione di treni di impulsi

Un treno di impulsi non è altro che una sequenza di uno e zero, e a seconda del linguaggio di programmazione e dell'istruzione che si sta utilizzando, è possibile specificare il numero di volte che si ripete l'insieme 1 – 0 o il tempo durante il quale si manda questo treno di impulsi. In entrambi i casi specifichiamo una durata.