

Il primo programma: ingressi/uscite

Iniziamo con questo programma una nuova sezione. Dopo aver visto i concetti generali della programmazione e delle istruzioni fornite dal LetPicBasic, arriva il momento di mettere in pratica tutto questo mediante l'applicazione su programmi completi.

Inoltre quando avremo imparato le istruzioni per fare le stesse cose in Assembler potremo comparare entrambi i linguaggi.

Fasi di un progetto

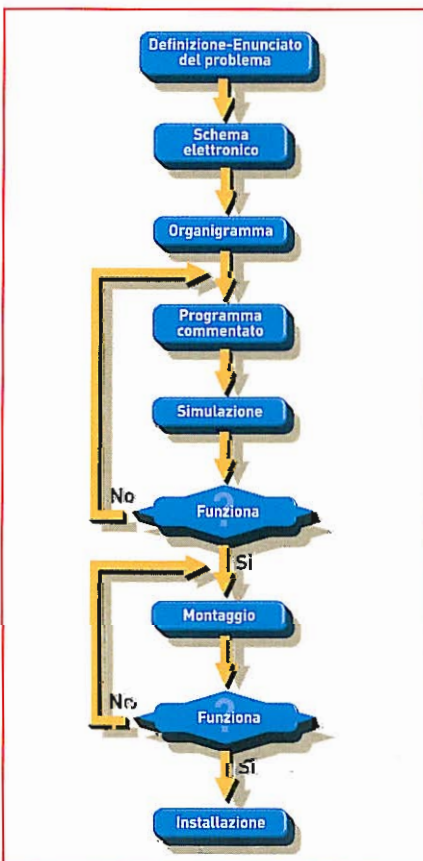
Tratteremo i nostri programmi come progetti completi, in modo da realizzare tutti i passi che li compongono, dalla definizione del problema, sino alla sua simulazione e messa in marcia. Il lavoro sarà indipendente dal linguaggio di programmazione, però quando arriveremo al momento di dover scrivere un programma ne dovremo scegliere uno specifico.

Dato che dobbiamo imparare a programmare in entrambi i linguaggi, elaboreremo diverse soluzioni. Dopo la scrittura del programma normalmente si esegue la simulazione, anche se questo passaggio non sempre è possibile, dato che, ad esempio,

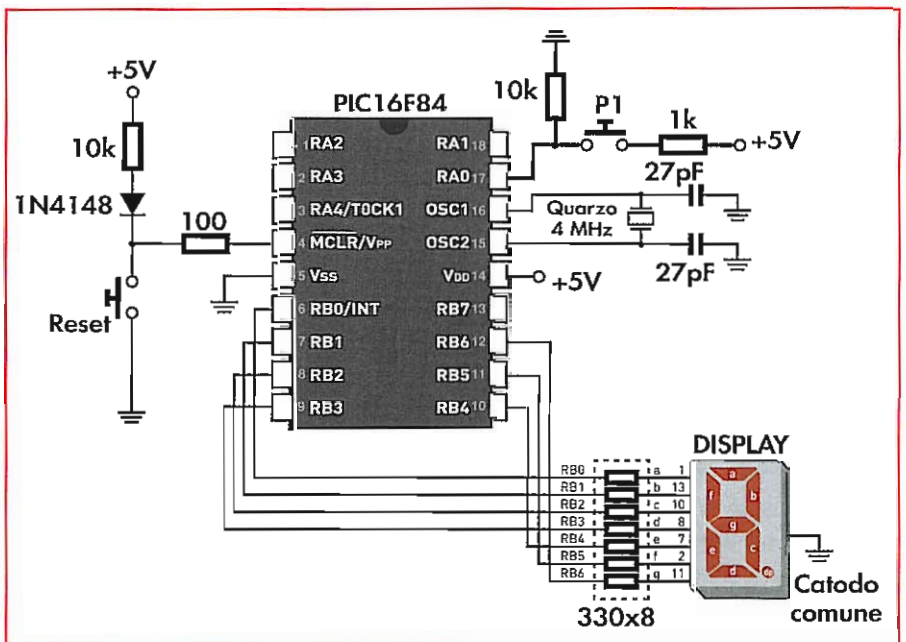
il LetPicBasic non ha questa parte di simulazione. Il compilatore, invece, è dotato di questo simulatore che impareremo a utilizzare. Quando il programma non funziona durante la simulazione, di solito è colpa di un'errata trasposizione dell'organigramma nel programma, bisogna quindi correggerlo, anche se è possibile che l'errore si trovi nella realizzazione dello schema elettronico, oppure nella definizione del problema, nel cui caso bisogna tornare ancora più indietro.

Definizione del problema

Una caratteristica fondamentale dei microcontroller è la loro



Passaggi tipici di un progetto.



Schema elettronico dell'esercizio proposto.



capacità di comunicare con l'esterno mediante le porte di ingresso e uscita. Il nostro primo programma farà uso di queste per dimostrare l'enorme potenzialità che forniscono.

Supponiamo di dover implementare un sistema per mostrare i turni di richiesta di un ristorante. Con un pulsante si incrementerà il turno ogni volta che viene premuto, questo turno sarà visualizzato da un display a 7 segmenti. Arrivando al numero 9 il turno inizierà di nuovo.

```

File Edit Compile Options Help
[Icons]
1 DEVICE 16F84          * PIC16F84
2 DIM Contatore        * Per supportare il N° di impulsi
3
4 * Numeri in codice a 7 segmenti
5 DATA 63,6,91,79,102,109,124,7,127,103
6
7 DEFINE PortA=*00000001 * RA0 ingresso
8 DEFINE PortB=*00000000 * Uscite
9
10 Symbol Pul=A.0      * Nome del pin di ingresso
11
12 * Inizio del programma principale
13 Contatore=0
14
15 INC:  BUTTON Pul    * Primo cambio da 0 a 1
16      BUTTON Pul    * Cambio da 1 a 0
17
18      Contatore=Contatore+1
19      RESTORE Contatore * Si posiziona all'indirizzo
20      READ  PortB     * Visualizza sul display
21
22      IF Contatore=9 THEN Contatore=0
23      GOTO INC
24
25 END
PIC-BASIC COMPILED OK 57 Words used.
    
```

Schema elettronico

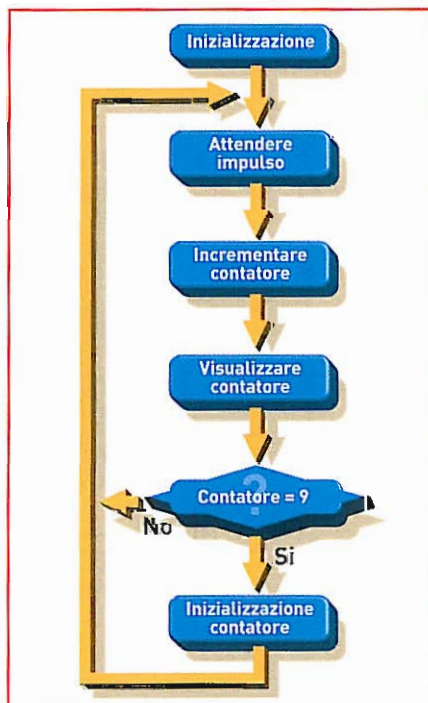
Tutti i nostri schemi elettronici avranno il PIC come parte fondamentale, con diversi piedini utilizzati per le funzioni di base, e altri per i dispositivi di ingresso/uscita con la relativa circuiteria. La scelta del microcontroller è molto importante perché da esso dipende, fra le altre cose, il numero di piedini e delle risorse disponibili. Per il momento lavoreremo con il 16F84, per provare i programmi compilati con il LetPicBasicLite. Il quarzo utilizzato insieme ai condensatori determinerà la velocità di esecuzione delle istruzioni, e con esse il funzionamento dei temporizzatori. Il fatto che i dispositivi siano collegati a livello alto o a livello basso, anche se non varia la complessità, influisce sulla programmazione. Nel nostro caso il pulsante sarà collegato per livello basso, quindi inserirà un "1" ogni volta che viene premuto.

Il display a 7 segmenti è a catodo comune, per cui gli stessi segmenti si accenderanno inviando un valore "1".

Programma dell'esercizio proposto.

Organigramma

L'idea originale è stata plasmata nell'organigramma della figura: dopo ogni attivazione del



Organigramma dell'esercizio proposto.

pulsante si incrementerà un contatore che sarà quello che viene visualizzato sul display. Arrivati al numero 9 si ricomincia il conteggio.

Programma commentato

Nella figura è riportato il programma risultante. Come si vede mette in pratica molti dei concetti che abbiamo già studiato, come ad esempio l'utilizzo delle tabelle, l'acquisizione degli impulsi e le strutture di controllo. È possibile utilizzare il programma usando altre opzioni.

Messa a punto

Per il momento non sarà possibile la simulazione del programma, però non preoccupatevi, tutto arriverà.

Nei prossimi programmi impareremo a simularli e li monteremo per verificare il loro reale funzionamento.