Libreria

Normalmente quando si iniziano a sviluppare programmi come quelli di ingresso/uscita — visti in un fascicolo precedente — ci si basa sulle istruzioni che ci fornisce il linguaggio di programmazione.

Quando i programmi crescono in complessità, o le periferiche utilizzate si fanno più sofisticate, si rende necessario l'utilizzo delle librerie. In seguito, migliorando la qualità dei programmi e la capacità di utilizzo del linguaggio, si arriva alla modifica delle librerie predefinite per adattarle a casi specifici.

Concetto di libreria

Una libreria non è altro che un insieme di procedimenti, funzioni e definizioni che semplificano la gestione di elementi complessi. Per il loro corretto utilizzo è necessario conoscere il nome di questi procedimenti e dei parametri da ricevere e restituire. Possiamo paragonare l'utilizzo delle librerie per il programmatore all'utilizzo di un telecomando della televisione per la programmazione del videoregistratore, dei canali tv ecc.

Con la pratica le librerie tenderanno a diventare sempre più "piccole" e sarà normale modificarle per adattarle alle caratteristiche di ogni applicazione. Potranno essere necessarie funzioni per la realizzazione dei calcoli, per la conversione, la validazione, la gestione delle periferiche, la comunicazione, ecc.

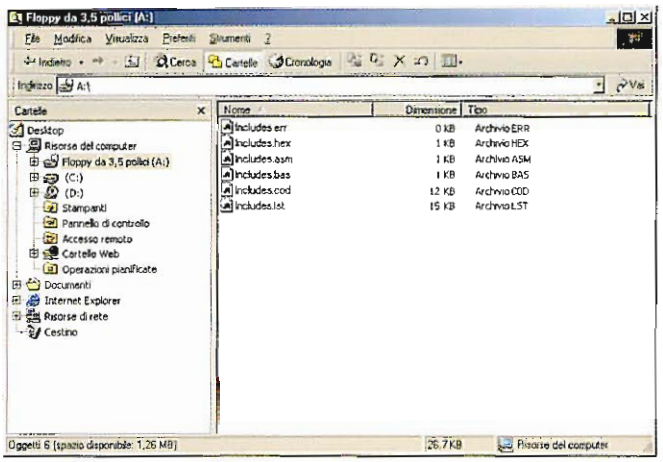
Una volta verificata una libreria possiamo dimenticarci dei dettagli di implementazione e utilizzarla come una libreria standard. In più, il concetto di libreria è molto generale e non è legato a nessun linguaggio di programmazione specifico, quindi una libreria utilizzata in un linguaggio potrebbe essere stata scritta in un altro linguaggio.

È anche molto comune l'acquisto di pacchetti di librerie. Estensioni tipiche secondo i linguaggi di programmazione sono: .LIB, .OBJ, .BPI, .DLL, .INC

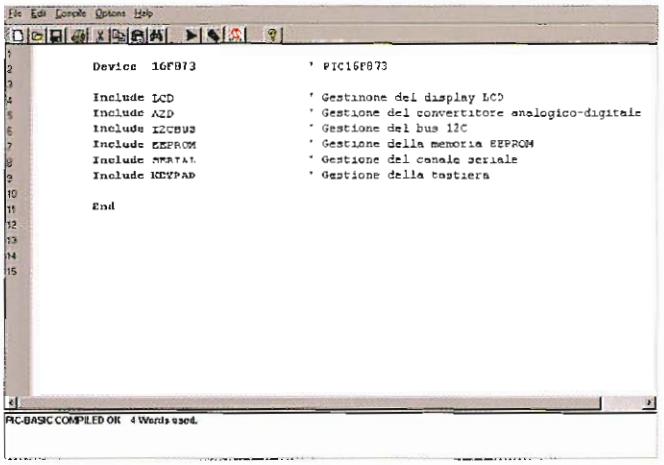
Le librerie del LetPicBasic

Fino a questo momento abbiamo utilizzato le librerie anche se non conoscevamo il loro termine esatto di definizione. Ogni volta che facciamo un INCLUDE "includiamo" ciò che nel LetPicBasic è chiamato "pacchetto": insieme di procedimenti per la gestione di dispositivi. Abbiamo a disposizione le sei librerie

Librerie fornite dal LetPicBasic.



File generati a partire dalla compilazione.



riportate nella figura. Il programma è realizzato e compilato con il LetPicBasicPro, cioè la versione professionale, dato che il LetPicBasicLite accetta solo il PIC16F84, che non ha convertitore analogico digitale.

Modifiche delle librerie nel LetPicBasic

Normalmente le librerie sono file esterni a cui si può accedere per vedere il loro contenuto e quindi per modificarlo. Tuttavia nel LetPicBasic questo non è permesso. Non esiste nessun file esterno dato che è il compilatore stesso che si incarica di inserire le istruzioni necessarie quando si utilizza un "include" e la successiva inizializzazione. Per un programmatore agli inizi questo non costituisce un grosso problema, però quando si vuole approfondire un discorso sulle routines, ad esempio cambiare le linee associate al display LCD, ci si scontra con questo limite. In ogni caso anche se non possiamo cambiare direttamente le librerie nel linguaggio Basic per il PIC come sarebbe auspicabile, impareremo a utilizzare i file che ci mette a disposizione il compilatore.

Fate attenzione a dove vengono scritti i file generati a partire dalla compilazione. Tutti possono essere aperti per la loro visualizzazione con un editor di testo. Se li ordiniamo per dimensione, il primo di tutti è il file di errore. Si suppone che appaia vuoto dato che non si è verificato alcun errore. "includi.hex" è il file che sarà utilizzato dall'hardware che scrive il microcontroller. Il suo formato è esadecimale ed è quello più vicino al linguaggio macchina. "includi.asm" è la traduzione che il compilatore fa in linguaggio assembler. Questo sarà il prossimo linguaggio che vedremo.

Nella figura possiamo osservare che ci sono due comandi "include" con un nome dopo di essi, gli "include" di questi livelli sono riferiti a file separati, ed è proprio qui che volevamo arrivare, perché è possibile entrare in questi file per poterli modificare. A seconda di cosa avremo bisogno vedremo le librerie a cui potremo accedere grazie a questo assemblatore. "includi.bas" è il file che abbiamo scritto in linguaggio Basic per PIC.

"includi.cod" e "includi.lst" non sono altro che modifiche intermedie del file "includi.asm" con comandi di controllo.

Includi.hex - Blocco note

File Modifica Cerca ?

```
:0800000064006300A1100286E
:02400E00323D41
:00000001FF
```

Il file esadecimale è incomprensibile per noi.

includi.asm - Blocco note

File Modifica Cerca ?

```
;--------------------------------
;-- L.E.T PIC BASIC PRO v7.00 --
;--------------------------------
        LIST  p=16f873 , w=2, x=on, r=DEC

        #include "P16F873.inc"
        #include "14bit.inc"

@Fin    Clrwdt
        Sleep
        F@Goto @Fin
        END
```

Programma tradotto in linguaggio assembler; il nostro prossimo obbiettivo.

Includi.lst - Blocco note

File Modifica Cerca ?

```
MPASM 02.50.02 Intermediate INCLUDES.ASM   23-07-2003  10:45:23
 PAGE  1

LOC OBJECT CODE     LINE SOURCE TEXT
  VALUE

                    00001 ;--------------------------------
                    00002 ;-- L.E.T PIC BASIC PRO v7.00 --
                    00003 ;--------------------------------
                    00004         LIST P=16f873 , w=2, x=ON, R=DEC
                    00005
                    00006         #INCLUDE "P16F873.inc"
                    00001
                    00334         LIST
                    00007         #INCLUDE "14bit.inc"
                    00001
                    00541         LIST
                    00542
0000                00543 start@
                    00544
                    00545
```

"includi.lst" contiene alcuni dati aggiuntivi a "includi.asm".