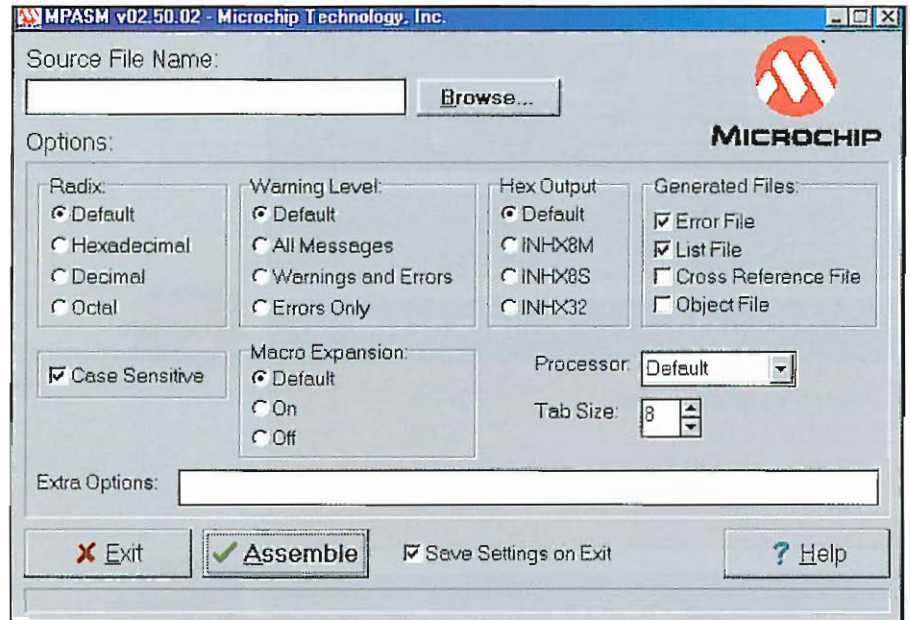


IL MPASM

Sino ad ora abbiamo parlato di due programmi per la programmazione ad alto livello di microcontroller. L'MPASM, invece, è un compilatore che traduce dal linguaggio assembler al codice macchina comprensibile per il microcontroller. Il compilatore contenuto nel LetPicBasic e nel LetPicBasicPlus traduce dal linguaggio all'assembler, e poi chiama l'MPASM per tradurlo in codice macchina. Anche se normalmente la chiamata a questo programma non si fa direttamente, perché viene utilizzato dagli altri linguaggi di programmazione, in questo capitolo vedremo le opzioni che presenta, nel caso in cui ci possa interessare configurarlo in altro modo.

Installazione

Questo programma non necessita di un'installazione

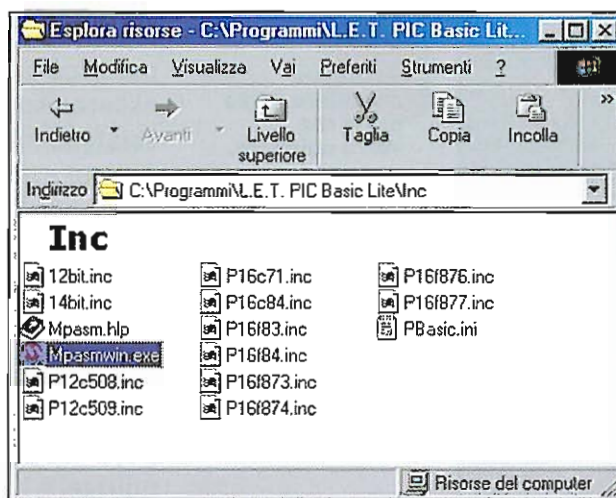


Videata che appare aprendo l'MPASMWIN.

particolare, dato che essendo utilizzato dal LetPicBasic e dal LetPicBasicPlus è già stato incluso nelle loro directory. Se andiamo a vedere nella directory "Inc" di uno di essi, potremmo trovare il file

"Mpsasmwin", che è la versione dell'MPASM per l'ambiente Windows.

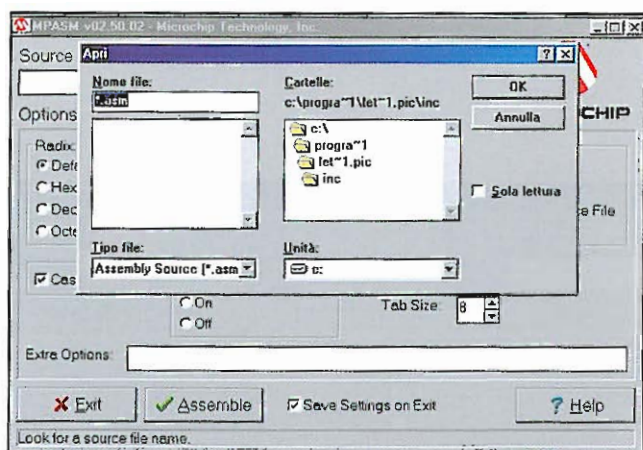
Esegendolo appare un'unica videata da cui si realizza la configurazione del compilatore e l'assemblaggio dei programmi.



Directory dove si trova l'MPASM.

Ambiente dell'applicazione

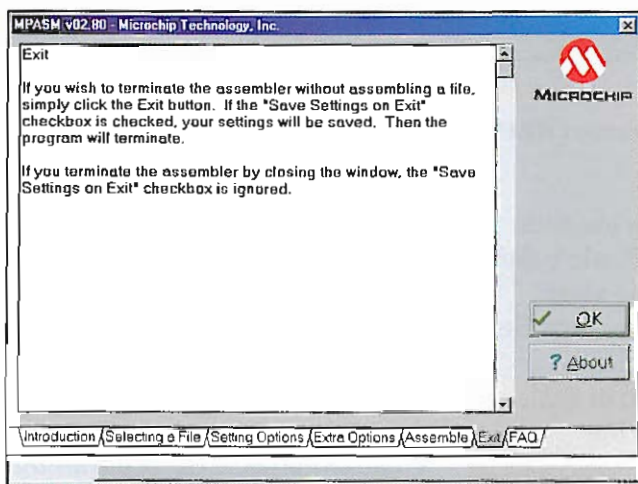
Da questa videata è possibile scegliere l'indirizzo dove si trova il programma da compilare. Se non viene specificato si prende per default la stessa directory dove si trova MPASM. L'opzione "Radix" permette di determinare il tipo di dato numerico per default nei



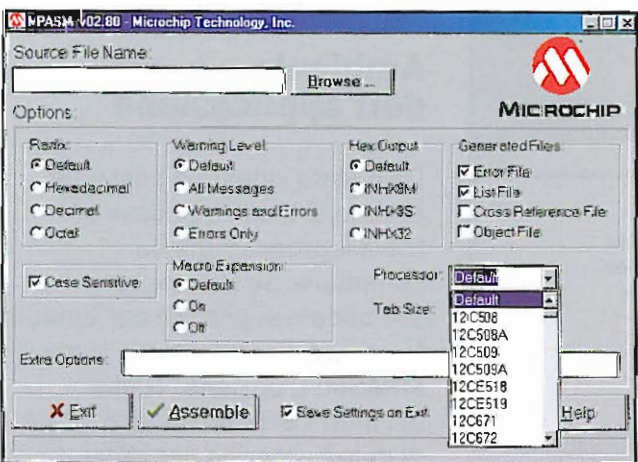
Si può scegliere la directory dove si trova il file da compilare.

TIPO DI DATO	MODULO DI DEFINIRLO
Valore esadecimale	h'0A'
Valore ottale	o'45'
Valore binario	b'10101100'
Valore decimale	d'34'

Formato per determinare il tipo di dato.



Cliccando sul pulsante di aiuto si possono ottenere descrizioni dettagliate sulle opzioni.



Dobbiamo segnalare il tipo di Pic che verrà utilizzato.

programmi. Se si sceglie ad esempio "Hexadecimal" questo sarà il tipo di dato numerico, se non viene specificato nulla davanti ai dati. In ogni caso anche se si sceglie un tipo di valore di default, si potrà modificare nello stesso programma il tipo di dato che si desidera, seguendo il modo illustrato nella figura. Inoltre si possono specificare di quale tipo di errori e messaggi vogliamo essere avvisati. Questo si fa nel paragrafo dei livelli di avviso (Warning Level). Il formato del codice macchina si sceglie nel sottomenù "Ex output". Questo è importante, perché a seconda del tipo di microcontroller su cui verrà diretto il codice, sarà necessario un formato oppure un altro. I file generati con "Generated File" possono essere di diverso tipo, conviene come minimo selezionare il file d'errore. È necessario scegliere i PIC che si utilizzano, dato che il compilatore ha bisogno di conoscerne il modello per verificare che il codice sia giusto o fornire messaggi d'errore. Se selezioniamo l'opzione "Case Sensitive" il compilatore interpreterà in modo differente le lettere minuscole e quelle maiuscole. Le macro possono essere espresse oppure no, esistono inoltre una serie di opzioni extra che possono essere definite. Quando sono state decise tutte le opzioni si può assemblare il programma selezionando il pulsante "Assemble". Se sono necessarie più informazioni su alcune di queste opzioni abbiamo la possibilità di utilizzare l'aiuto "Help".

