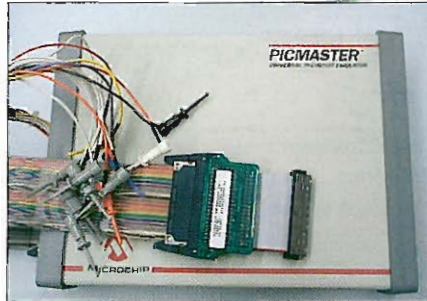


## MPLAB (III)

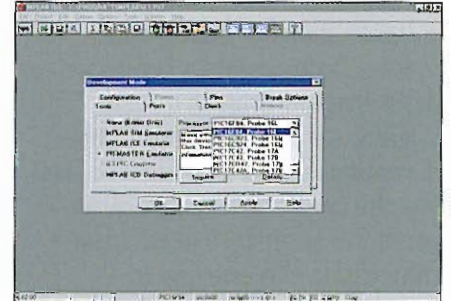
**D**opo aver compilato il programma sorgente libero da errori, arriva il momento di verificare che esegua effettivamente quello che desideriamo, cioè oltre a non avere errori di sintassi, non ne abbia nemmeno di tipo semantico. Potremmo scriverlo direttamente sul microcontroller e, se il programma è semplice, è molto probabile che i controlli eseguiti siano sufficienti e funzioni al primo colpo. Normalmente però, per programmi di complessità un po' più elevata, non tutto funziona come vorremmo, ed è difficile capire dove si trovi l'errore. È in questi casi che diventa necessario l'utilizzo di strumenti più avanzati.

### Gli emulatori in circuito

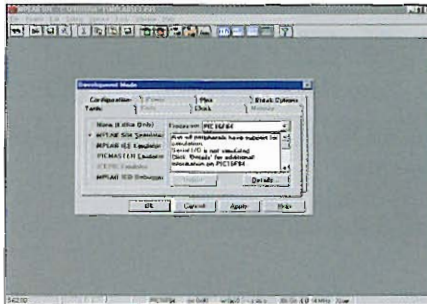
Una delle possibilità è utilizzare un emulatore in circuito; si tratta di uno strumento hardware che si collega al computer con le funzioni del microcontroller, permettendoci di eseguire il programma linea per linea in tempo reale, però con il vantaggio di poter seguire sul monitor del computer lo stato dei registri interni, la linea di esecuzione, i salti, ecc. vedendo così dove si trova l'errore nel programma. MPLAB dispone di strumenti software per lavorare, ad esempio, con l'emulatore PICMASTER. Dobbiamo solo collegare "la testa di prova" con l'hardware dell'applicazione da testare, come se si trattasse di un PIC normale, e iniziare



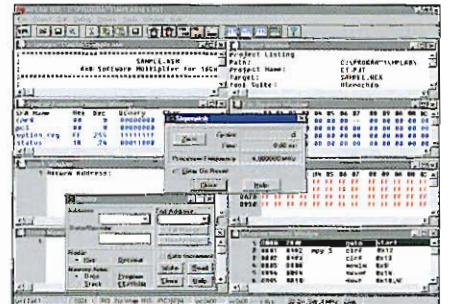
Gli emulatori in circuito sostituiscono il microcontroller nelle applicazioni sotto prova.



Per lavorare con l'emulatore bisogna cambiare il modo di sviluppo.



Anche la scelta del simulatore si fa tramite il modo di sviluppo.



Finestre utili nella simulazione.

l'emulazione. Per fare questo bisogna cambiare il modo di sviluppo in MPLAB passando in "modo emulazione" e bisogna scegliere il microcontroller da emulare.

### Il simulatore

Anche se l'utilizzo dell'emulatore può risultare molto pratico, si tratta di uno strumento piuttosto costoso, e per la maggioranza delle prove è sufficiente un simulatore software. MPLAB comprende in modo gratuito un simulatore completo per l'assembler del PIC. Anche se per

ora non abbiamo ancora lavorato in assembler, vediamo le caratteristiche più importanti che incontreremo man mano nei differenti programmi. Per lavorare con il simulatore bisognerà nuovamente cambiare il modo lavoro. Una finestra ci indicherà le limitazioni che troveremo nella simulazione.

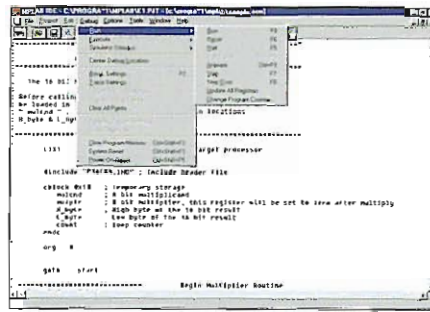
### Finestre utili per la simulazione

Selezionando il simulatore come strumento di lavoro, possiamo verificare come il numero di opzioni disponibili sia aumentato.

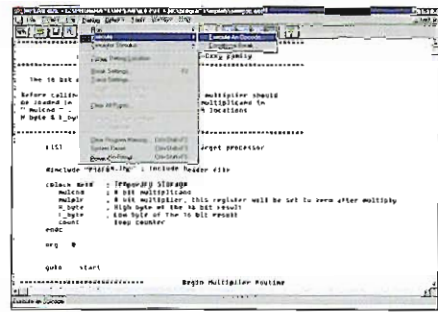


Aprire il progetto e il programma che abbiamo commentato vedendo la procedura di compilazione, andate al menù delle finestre (Window) e selezionate tutte quelle che potete. Ordinatele. La conoscenza della maggioranza di queste finestre sarà utile al momento di realizzare una simulazione. Nel file che contiene il programma verranno segnalate le istruzioni man mano che entrano in esecuzione, in modo che sia possibile capire dove si trova il punto di lavoro in ogni momento.

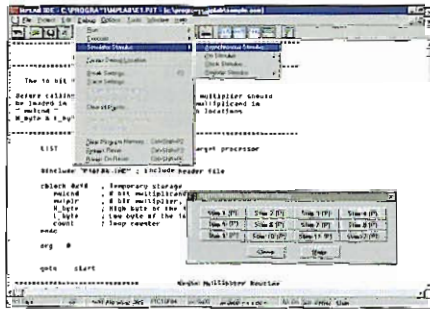
La finestra di progetto (Project Window) mostra le caratteristiche generali dello stesso. Nella finestra di clock (Stopwatch) abbiamo il riferimento del tempo di esecuzione del programma, molto utile dato che una simulazione non è realizzata in tempo reale. La finestra dei registri (File Register Window) mostra il valore di questi, mentre la finestra dei registri speciali (Special Function Register Window) mostra i valori dei registri specifici. Sia gli uni che gli altri variano con l'esecuzione del programma. La modifica dei dati delle differenti memorie (RAM, programma, EEPROM e stack) per simulare le differenti condizioni di lavoro che si possono verificare si fa tramite la finestra Modify. Anche la memoria di programma e la EEPROM sono accessibili tramite le loro finestre corrispondenti. Nella finestra di stack (Stack Window) vedremo come vengono scritti e letti gli indirizzi delle chiamate e dei ritorni dalle subrutines. Nella finestra che ci interessa, Trace Memory Window, vedremo riportata l'informazione al passaggio da una determinata



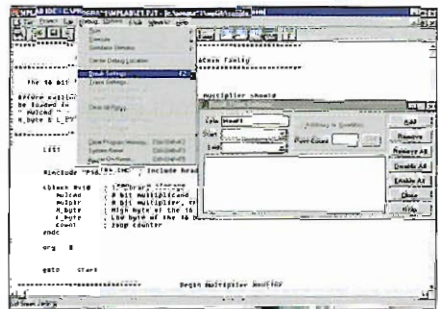
Opzioni del submenù Run.



Opzioni per simulare parti del programma.



Opzioni per introdurre dati al programma.



Opzioni per fermare il programma in un determinato punto.

parte del programma, per la sua analisi successiva.

## Opzioni della simulazione

Le informazioni della simulazione appaiono nelle finestre viste in precedenza, ma come si realizza una simulazione? Le opzioni sono molte, dipende tutto da ciò che si vuole ottenere. Tramite il menù Debug si può accedere a tutte le possibilità. È possibile eseguire il programma una sola volta, vedendo solo lo stato finale (Run), oppure passo a passo (Step e Step Over), da un punto determinato (Change Program Counter) è possibile anche far girare il programma vedendo come cambiano dinamicamente i

registri (Animate); è anche possibile che si vogliono eseguire le istruzioni di salto per verificare che questi ultimi vengano fatti con i dati esistenti (Execute An Opcode) o simulare il programma sino a che si compia una condizione (Conditional Break). Un'opzione molto interessante è quella che ci permette d'introdurre differenti stimoli al sistema (Simulator Stimulus), in altre parole, fornire dei segnali che simulino le informazioni captate dai sensori. Infine, un'opzione immancabile in tutti i simulatori, e che anche MPLAB possiede, è quella di lavorare con i punti di rottura (Break Points), in modo che sia possibile vedere quando il programma passa per determinati punti, al fine di localizzare più facilmente le istruzioni in conflitto.

