

# MPLAB - IDE

## Introduzione

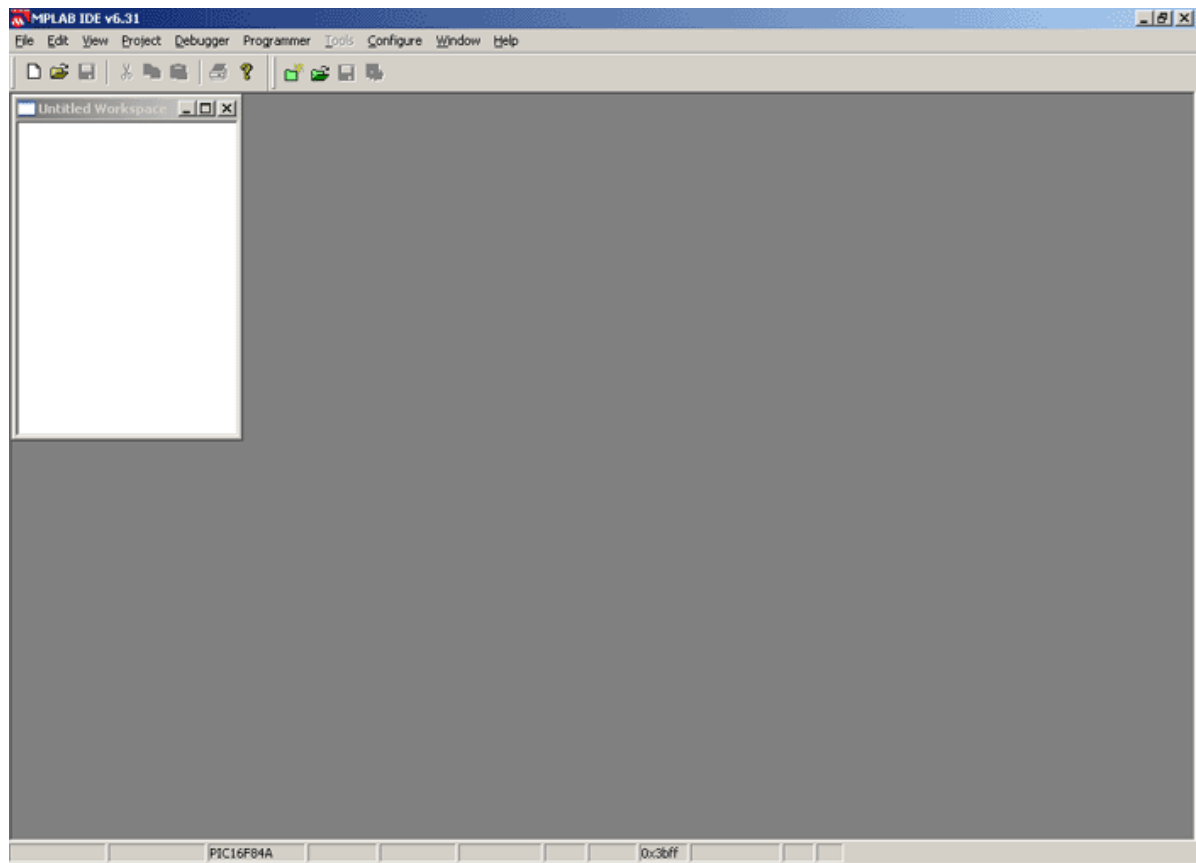
1. Benvenuto in MPLAB
2. Creare un progetto
3. Creare un nuovo file assembler
4. Scrivere un programma
5. Toolbar icons
6. Simulatore MPSIM

## Introduzione

MPLAB è un programma che permette di creare e sviluppare un programma con estrema facilità. Lo si può descrivere come uno ambiente di sviluppo (IDE) per un determinato linguaggio standard necessario per programmare un PC. Non occorre eseguire funzioni mediante riga di comando ma tale ambiente di sviluppo è in grado di supportare l'operatore in tutte le fasi, dal scrittura del codice fino alla simulazione e anche fino alla fase di emulazione.

## 1. Benvenuti in MPLAB

Lanciamo il programma MPLAB. MPLAB è una windows like application.

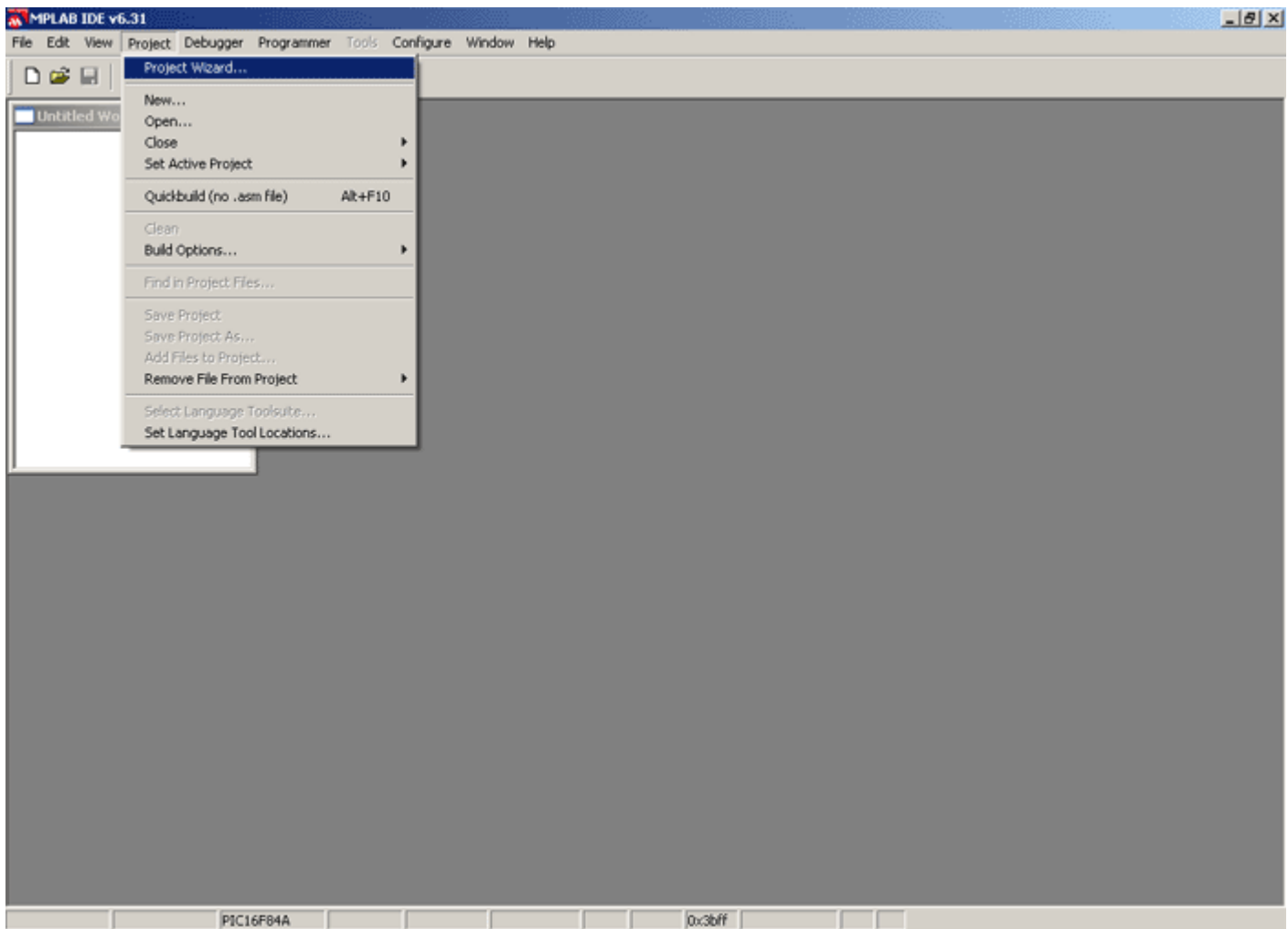


**La prima schermata dopo l'apertura di MPLAB**

## **2. Creazione di un progetto**

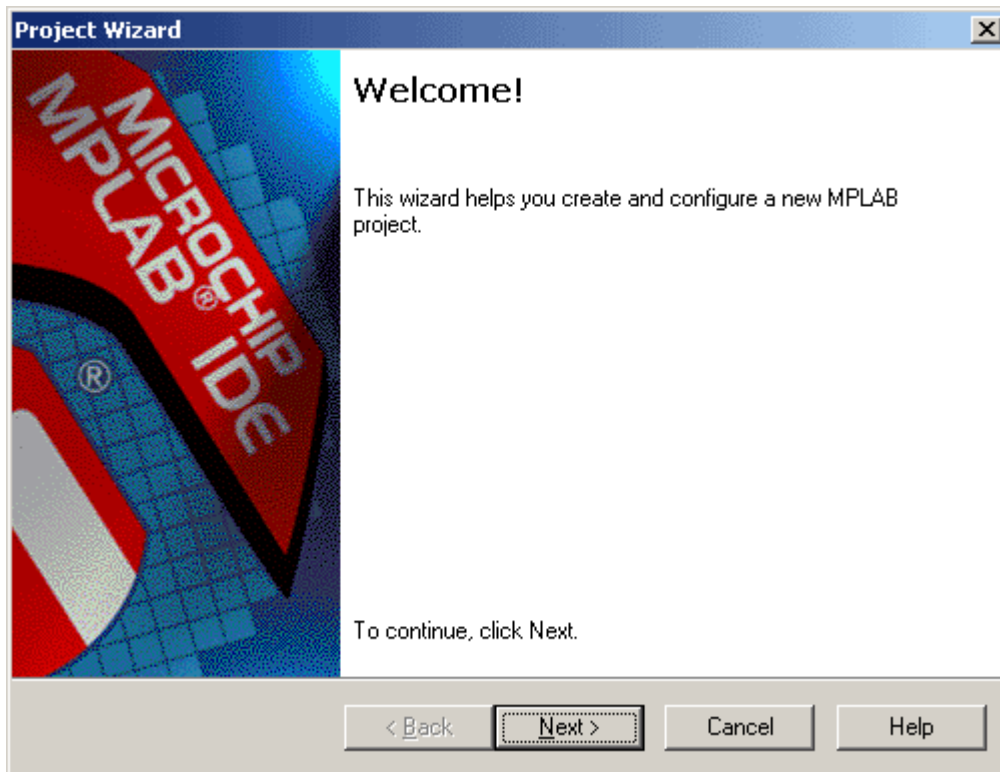
La preparazione del programma da caricare nel microcontrollore richiede i seguenti passi:

1. Creare un progetto
2. Scrivere il programma
3. Convertirlo in binario (compilare).



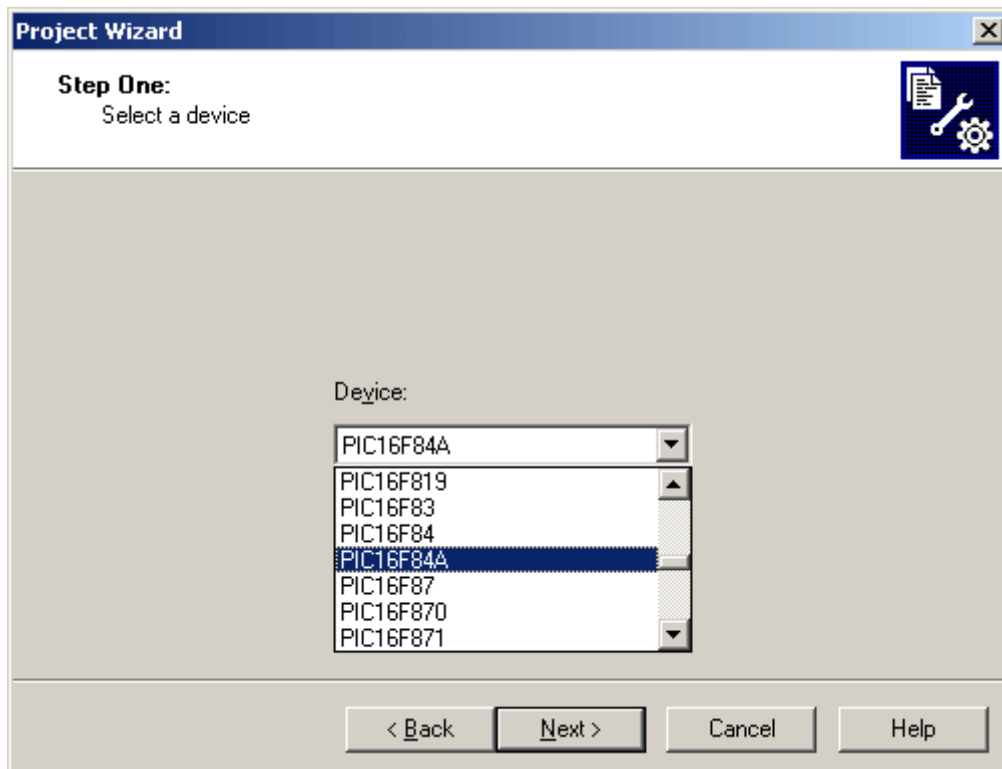
### **Aprire un nuovo progetto**

Cliccare su PROJECT e successivamente su PROJECT WIZARD, sarà aperta la seguente finestra.



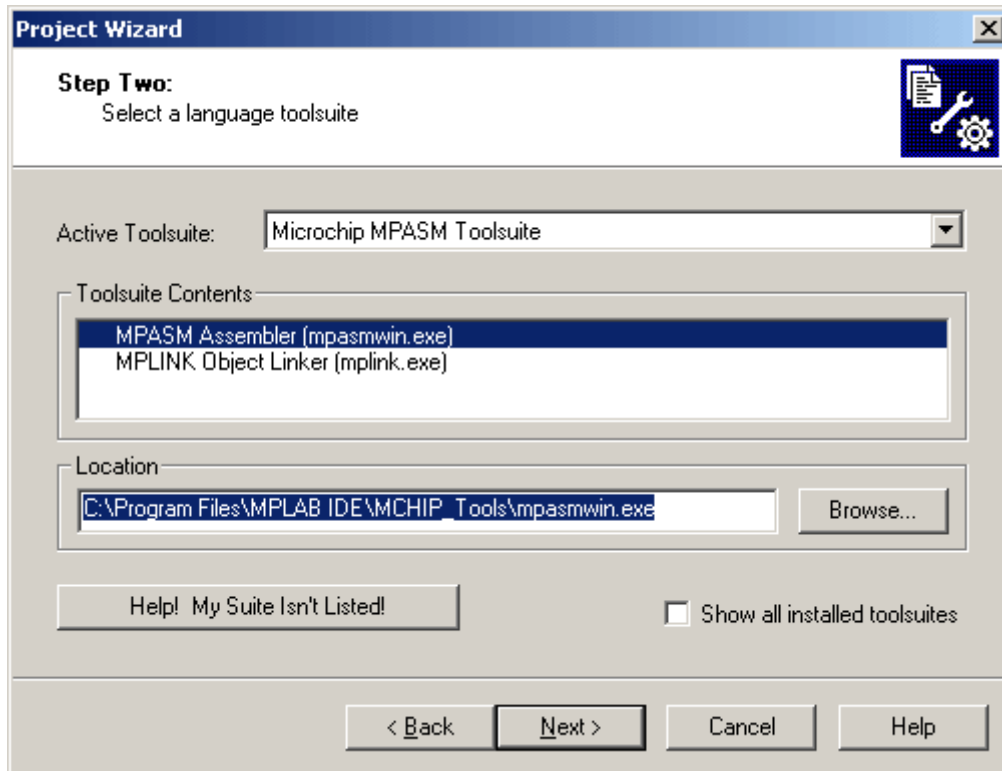
### Creare un nuovo progetto

Cliccare su NEXT per continuare. Occorre adesso scegliere l'appropriato microcontrollore della famiglia PIC. Noi sceglieremo il PIC16F84A.



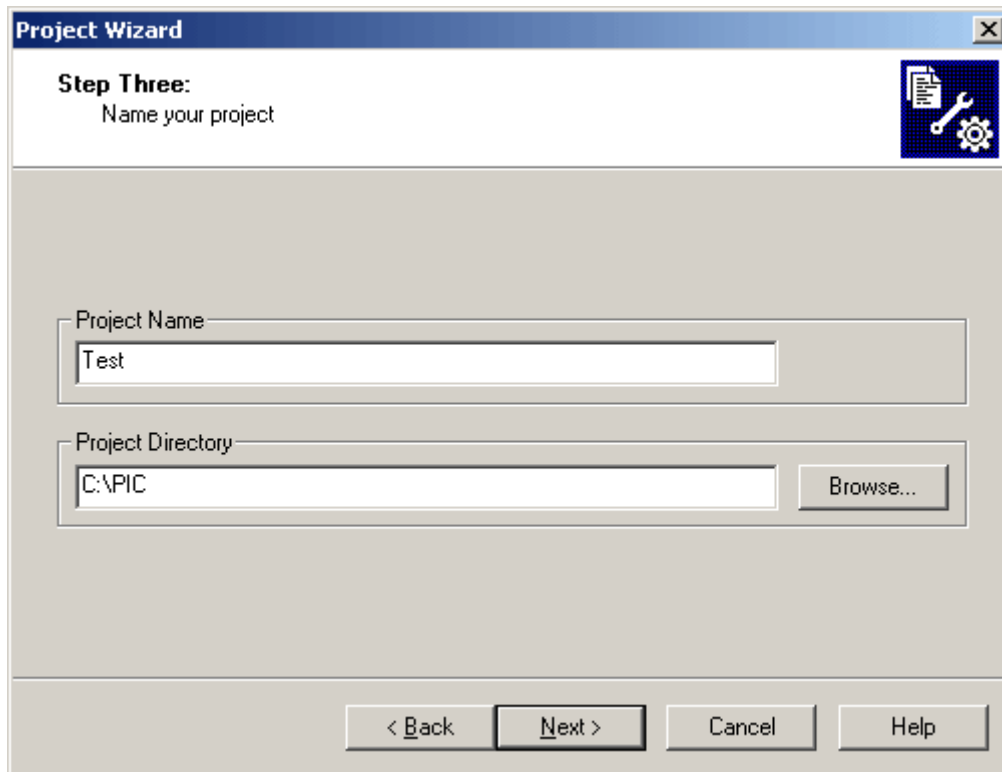
### Scegliere il microcontrollore appropriato

Il passo successivo consiste nel definire il linguaggio che si intende utilizzare per la programmazione. Useremo direttamente il linguaggio assembler, pertanto occorre selezionare l'opzione MPASM Assembler come indicato in figura sotto.



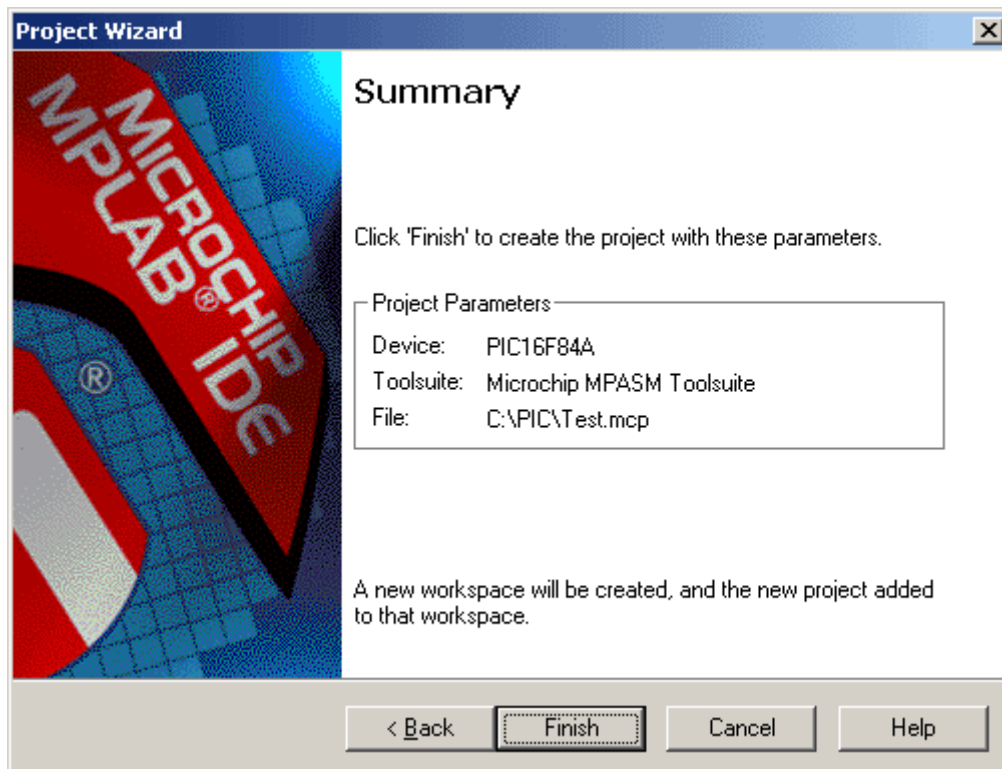
### **Selezionare il corretto "language toolsuite"**

Indichiamo il nome del progetto e il percorso. Il nome dovrebbe riflettere lo scopo e il contenuto del programma. Raggruppiamo il progetto in una cartella in grado di ricordarci il contenuto; nel nostro caso si utilizzerà il nome PIC come qui di seguito riportato in figura.



### Nome del progetto

Cliccare NEXT --> si aprirà la finestra "summary".

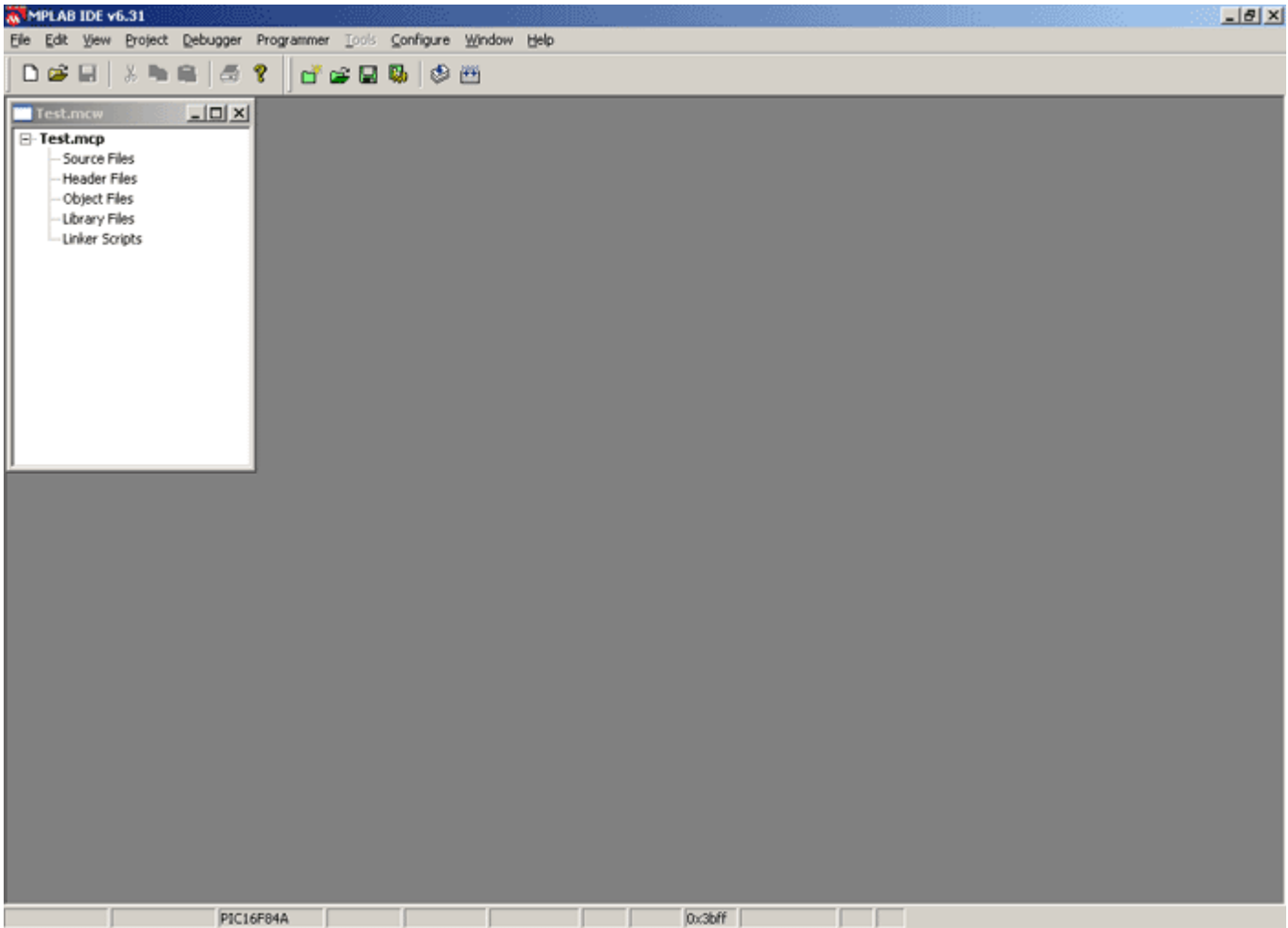


### Summary contenente i parametri precedentemente definiti

Cliccare FINISH per creare il progetto.

### 3. Creare un nuovo file assembler

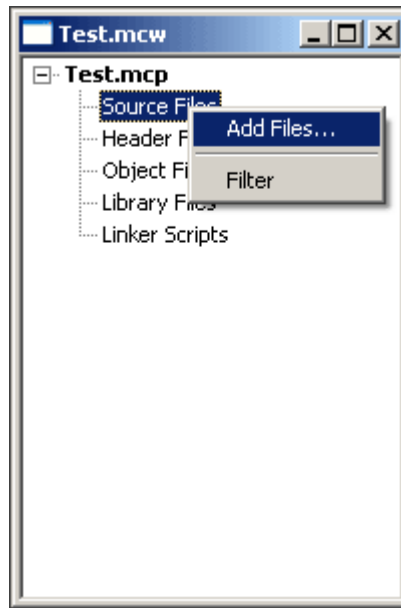
Dopo aver creato il progetto mediante il wizard dovrebbe apparire la seguente schermata.



#### Apertura di un nuovo progetto

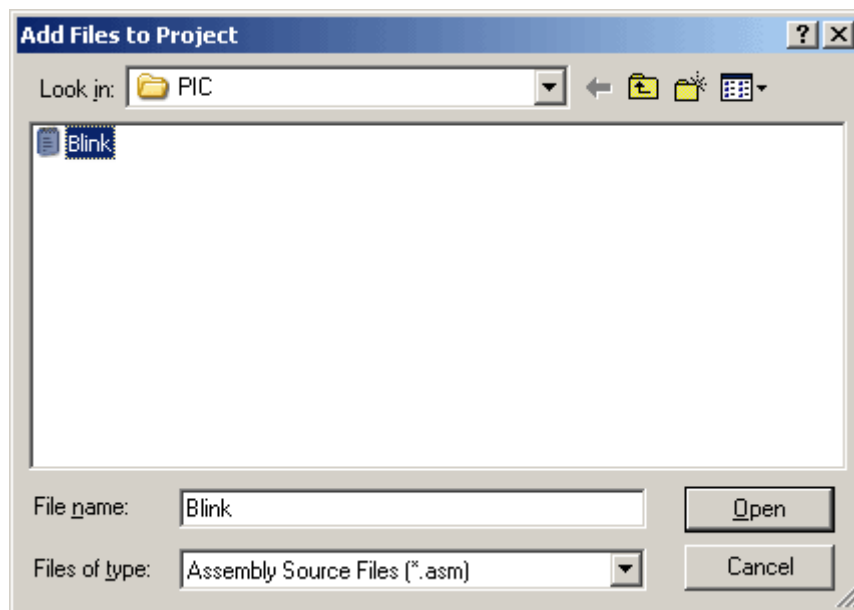
Adesso dobbiamo scrivere il programma, questa operazione richiede ulteriori file da dover aprire. Cliccare su FILE > NEW, si aprirà una nuova finestra all'interno della working area di MPLAB (La nuova finestra conterrà il programma che si dovrà scrivere). Dopo aver aperto il nuovo file dobbiamo salvarlo (è buona regola), useremo il solito percorso C:\PIC ed il nome del file sarà "Blink.asm", questo è un nome che ci permetterà di ricordare la natura del programma (ad esempio questo farà lampeggiare (blink) uno o più diodi led sulla porta B di un microcontrollore).

Il nuovo file, "Blink.asm" dovrebbe così essere incluso al progetto. Premere il tasto destro del mouse sul "source file" nella finestra "blink.mcp". Questo aprirà una piccola finestra con due opzioni – scegliamo la prima, "Add Files".



**Inserimento di un nuovo file Assembler nel progetto  
(immagine con nome progetto errata)**

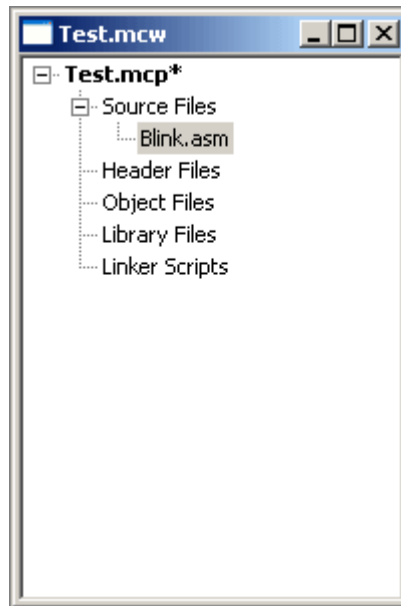
Si apre una nuova 'browse window' cerchiamo la nostra cartella PIC e selezioniamo il file "Blink.asm", come nella immagine sotto.



**Selezionare il file**

L'aspetto della finestra del progetto dopo aver aggiunto il file blink.asm sarà il seguente.





## 4. Scriviamo un programma

Solo dopo aver terminato le precedenti operazioni saremo in grado di iniziare a scrivere un programma. Il programma che noi creeremo è estremamente semplice ed è stato realizzato solo allo scopo di illustrare la creazione di un progetto e la sua simulazione.

```
;Programma per settare la porta B al valore logico VERO
;MCU utilizzato: PIC16F84 Data: 2/03/2005

;***** Dichiarazioni e configurazioni per il microcontrollore *****
PROCESSOR 16f84
#include "p16f84.inc"

__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

;***** Dichiarazione delle variabili *****
Cblock 0x0C ; Beginning of RAM
endc ; No variables

;***** Struttura della memoria del programma *****

ORG 0x00 ; Reset vector
goto Main ; After reset jump to location

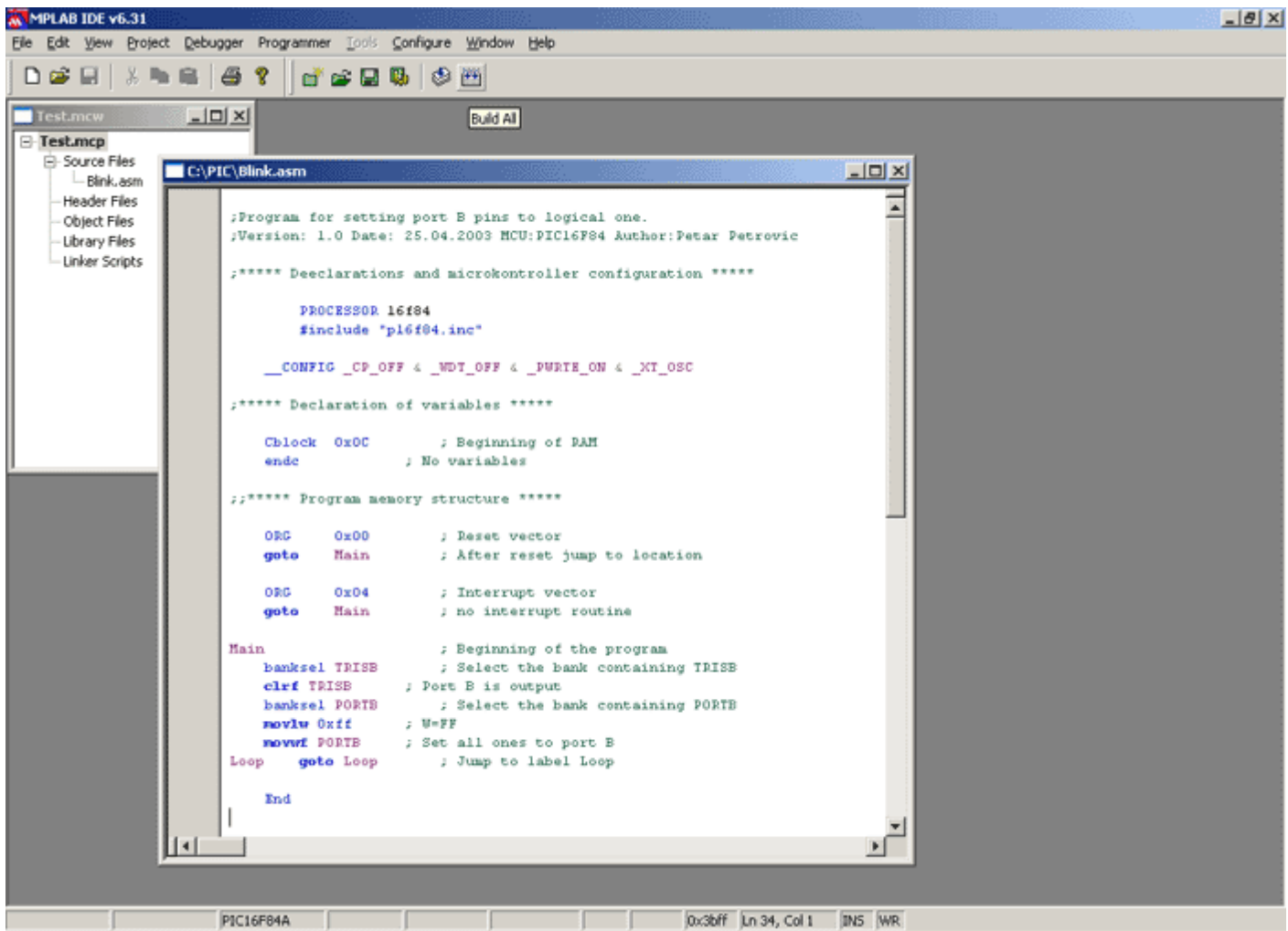
ORG 0x04 ; Interrupt vector
goto Main ; No interrupt routine

Main ; Beginning of the program
banksel TRISB ; Select the bank containing TRISB
clrf TRISB ; Port B is output
banksel PORTB ; Select the bank containing PORTB
movlw 0xff ; W=FF
movwf PORTB ; Set all ones to port B

Loop ; Label Loop
goto Loop ; Jump to label Loop

End
```

Dovete scrivere il programma nella nuova finestra appena aperta (copy/paste).




### Finestra principale con il programma blink.asm














Al termine della scrittura del programma nella finestra "Blink.asm", siamo pronti a compilare il progetto, per farlo andiamo su PROJECT -> BUILD ALL; questo comando traduce il file assembler in un file eseguibile in formato HEX (esadecimale). Apparirà una finestra di log nella quale verrà indicato se la traduzione è andata a buon fine. "BUILD SUCCEEDED" è un messaggio che indica la corretta traduzione (nessun errore).

In caso di errori è possibile fare doppio click sul messaggio d'errore nella finestra 'Output' ed automaticamente verrà mostrato il codice assembler esattamente alla linea dove si è incontrato l'errore.

## 5. Toolbar icons

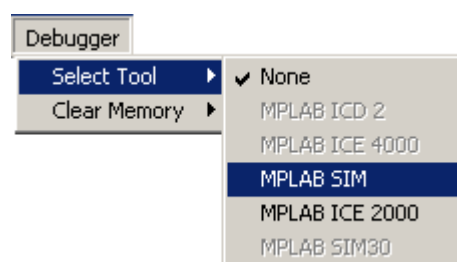
La seguente tabella contiene la descrizione di ciascuna icona sulla toolbar.

	Nuovo file assembler. Viene utilizzato per includere un file assembler in un progetto.
---	--

	Apri un file assembler esistente.
	Salva il programma (Assembler file)
	Cut text.
	Copy text.
	Paste text.
	Stampa il file aperto.
	Help sul progetto aperto e dell'MPLAB.
	Crea un nuovo progetto.
	Apri un progetto preesistente.
	Save Workspace. Workspace salvata in questo modo mantiene tutti i parametri e settaggi delle finestre. Conviene salvare il progetto frequentemente specialmente se si lavora con finestre multiple nel simulatore disposte a nostro piacere.
	Build options.
	Make A seguito di eventuali errori riscontrati durante la simulazione il programma deve essere riparato. Poichè il simulatore utilizza solo file di tipo HEX come input, il programma deve essere tradotto nuovamente in modo tale da creare il codice relativo alle modifiche apportate. Questa icona (Make) traduce il progetto dall'inizio e crea una versione del file HEX gestibile dal simulatore.
	Simile alla precedente icona, in questo caso viene tradotto l'intero progetto e non solo il file assembler ove il codice è stato modificato.






## 6. Il simulatore MPLAB SIM

Il simulatore è una parte dell'ambiente MPLAB il quale permette di avere una visione globale sul comportamento del nostro programma per un determinato microcontrollore permettendoci di verificare che il comportamento dello stesso sia corretto. Permette inoltre di comprendere meglio il comportamento del microcontrollore. Tramite il simulatore possiamo monitorare il valore corrente delle variabili utilizzate, dei registri e lo stato delle porte a livello di pin. Il valore aggiunto dal simulatore non è sempre costante infatti dipende dal programma che stiamo simulando. Se abbiamo un programma semplice (come questo da noi creato come esempio), la simulazione non è così importante poiché portare il valore sul pin della porta B non è poi così difficile da realizzare. Diversamente si rivela uno strumento indispensabile nel caso di programmi più complessi che magari contengano timers, oppure operazioni matematiche. La simulazione, così come indicato dal nome, "simula il lavoro di un microcontrollore". Il simulatore esegue il programma linea per linea ed evidenzia i valori dei registri e variabili del microcontrollore. Dopo aver scritto il programma è buona norma simulare le reazioni del sistema utilizzando il programma, e successivamente far girare l'applicazione in un sistema reale. Generalmente il passo della simulazione viene sempre considerato superfluo e viene sistematicamente scartato.

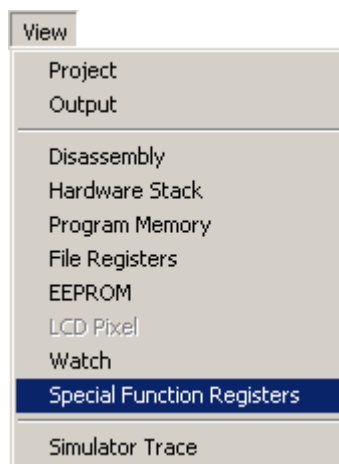


**Apertura del programma di simulazione**

Per aprire il simulatore cliccare su: DEBUGGER > SELECT TOOL > MPLAB SIM, come mostrato nell'immagine sopra. Appariranno quattro nuove icone sulla destra. Solo relative alla simulazione con il seguente significato:

	Esegue il programma alla velocità normale. Il simulatore esegue il programma fino alla "pausa" mediante l'apposito bottone.
	Termina l'esecuzione del programma. Il programma può proseguire l'esecuzione step-by-step o alla velocità normale.
	Esecuzione del programma Step-by-step. Cliccando viene eseguita la linea successiva del programma. Permette anche di entrare nelle macro e nelle subroutine.
	Come la precedente ma non entra in macro e subroutine.
	Resetta il microcontrollore. Il Program counter viene posizionato all'inizio del programma e la simulazione può ricominciare.

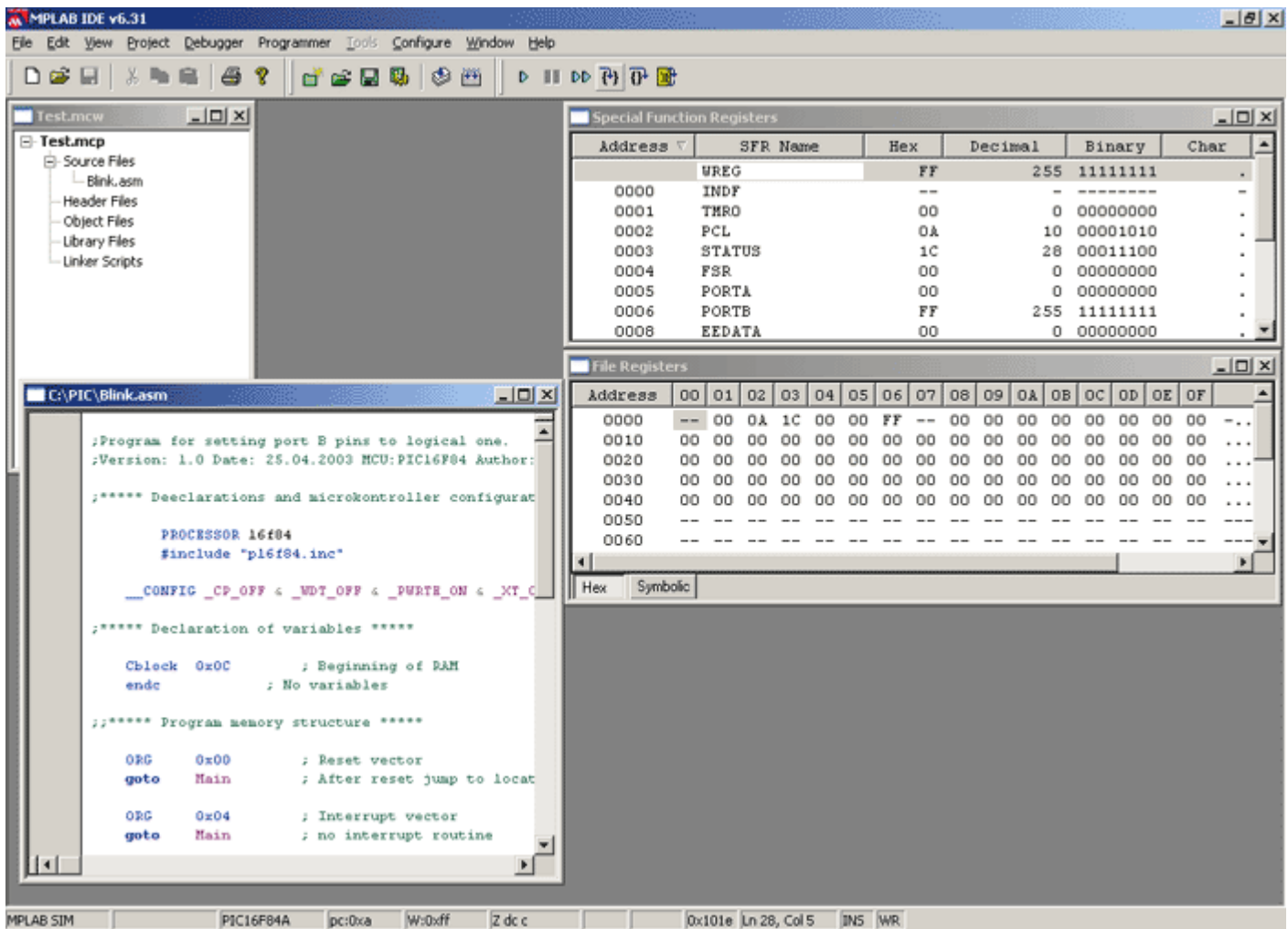
La prima cosa da fare è il reset del microcontrollore mediante il comando DEBUGGER > RESET oppure cliccando sulla icona di reset. Verrà evidenziata in verde la prima linea del programma, e il program counter PCL verrà resettato a zero, questo può essere visto nella finestra dei registri (*Special Functions Registers window*).



Una delle principali caratteristiche del simulatore è l'abilità di visualizzare lo stato dei registri all'interno del microcontrollore. Questi vengono anche chiamati "special function registers", ovvero SFR. Possiamo visualizzarli cliccando su VIEW > SPECIAL FUNCTION REGISTERS.

Oltre agli SFR, è utile dare uno sguardo nei file registers. La rispettiva finestra può essere aperta cliccando su VIEW > FILE REGISTERS.

Inoltre se il programma contiene delle variabili può essere interessante monitorarle. A ciascuna variabile viene assegnata una finestra (*Watch Windows*) cliccando su VIEW > WATCH.



### Simulatore con le finestre SFR e File registers aperte

Quando tutte le variabili e registri di nostro interesse sono visibili nella workarea, possiamo iniziare la simulazione. Possiamo impartire il comando *Step Into* oppure *Step Over*, a seconda se si voglia entrare o meno nelle subroutine. I medesimi comandi possono essere impartiti da tastiera mediante i tasti F7 o F8.

Nella finestra SFR, Si può osservare come il registro W riceva il valore 0xFF e lo trasferisca alla porta B.

Cliccando nuovamente F7, non avremo altri risultati e non andremo da altre parti poichè il programma è entrato in un loop infinito. Questo rappresenta un loop dal quale un microcontrollore non può uscire fintanto non sopraggiunge un interruzione (se queste sono utilizzate nel programma), oppure fino a quando il microcontrollore non viene resettato.