Aggiornamento
MPLAB X_parte 1
Riferimento versione 3.20
Apertura di un progetto e impiego del simulatore

L'ambiente di sviluppo MPLAB viene gradualmente rinnovato e sostituito da una nuova versione denominata MPLAB X, utilizzabile con i sistemi operativi Windows, Linux e Mac.
La novità più evidente di MPLAB X è un diverso aspetto dell'IDE.
Il progetto che si realizza non è legato soltanto al compilatore utilizzato ma anche al programmer/debugger.
Nel passaggio da una versione all'altra Microchip ha reso compatibili i progetti realizzati con la vecchia versione, progetti che si possono importare e convertire per il nuovo IDE, mantenendo intatte però le prerogative di ciascun ambiente.
E' possibile quindi lavorare su uno stesso progetto con entrambe le versioni; l'aggiornamento dei sorgenti non preclude la visualizzazione degli stessi nei differenti IDE.
Chi, per diversi motivi, decide di continuare a lavorare con la vecchia versione Microchip mantiene ancora, allo stato attuale, la possibilità di scaricare una versione in lento sviluppo.
In figura 1 vengono riportate le icone di due versioni.

Figura 1

Pagina iniziale

La pagina iniziale (*Start Page*) di MPLAB X viene riprodotta in figura 2.

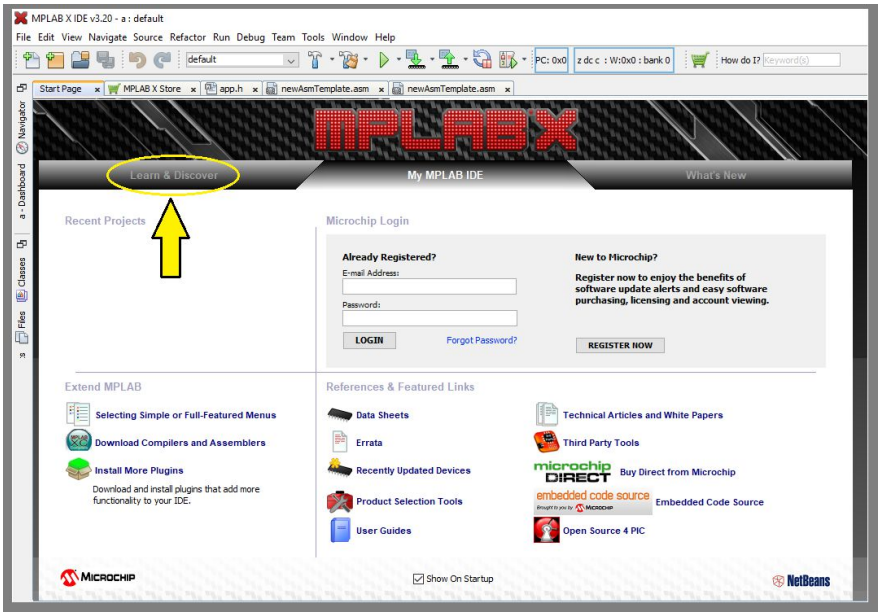

Figura 2

Selezionando *Learn & Discover* e di seguito *Quick Start* o *Get Started* (figura 3) si ha accesso a due guide semplificate (in lingua inglese) che consentono un primo approccio all'ambiente di sviluppo.

Figura 3

Gli argomenti contenuti nella guida *Quick Start* vengono sintetizzati in figura 4.

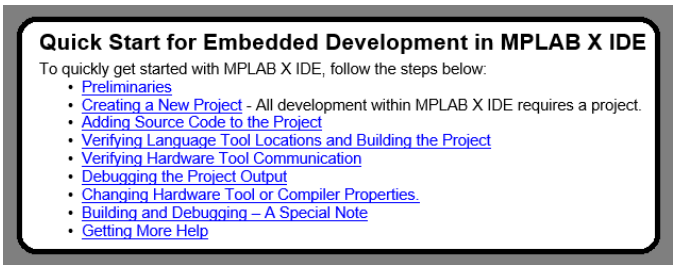

Figura 4

Apertura di un progetto

Per aprire un nuovo progetto dalla *Start Page* si può procedere in uno dei due modi seguenti (figura 5):

- selezionando **File** e di seguito **New Project** dal menù a tendina;
- facendo clik sull'icona corrispondente situata nella barra sottostante .

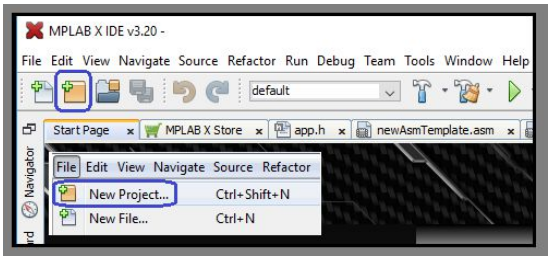

Figura 5

Con l'apertura di un progetto si ha accesso alla finestra **New Project** riportata parzialmente in figura 6.

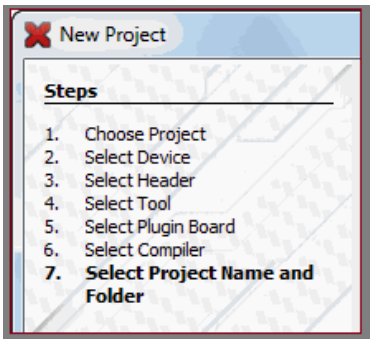

Figura 6

La definizione di un progetto prevede sette step successivi (non tutti necessari per la definizione del medesimo) e l'apertura delle finestre corrispondenti.
Di seguito vengono presi in considerazione soltanto gli step necessari per realizzare un progetto con il microcontrollore 16F84A.
E' il programma stesso che durante la fase di definizione di un progetto salta gli step che non sono necessari, in questo caso il 3 e il 5.

Step 1_*Choose Project* (figura 7)
Consente la selezione della categoria (in questo caso *Microchip Embedded*) e del tipo di progetto (in questo caso *Standalone Projects*).

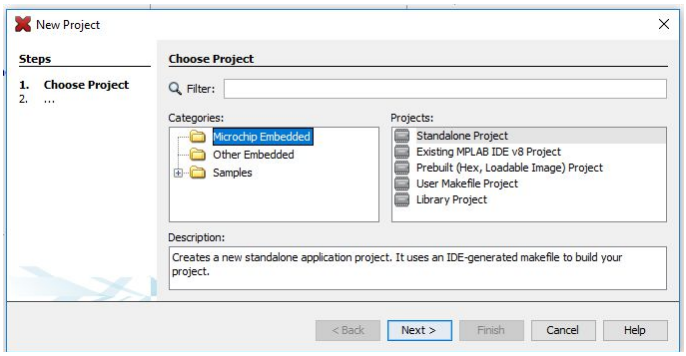

Figura 7

Step 2_*Select Device* (figura 8)
Consente la selezione del microcontrollore che si intende utilizzare (famiglia e sigla del dispositivo, in questo caso 16F84A).

Figura 8

Step 4_*Select Tool* (figura 9)
Viene scelto lo strumento (tool) con il quale programmare, in questo caso il simulatore (*Simulator*) messo a disposizione dal programma per verificare la correttezza del software.

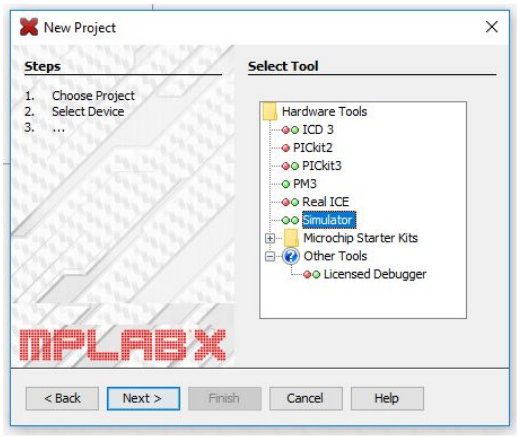

Figura 9

Step 6_*Select Compiler* (figura 10)
Viene scelto il compilatore, in questo caso *mpasm*.

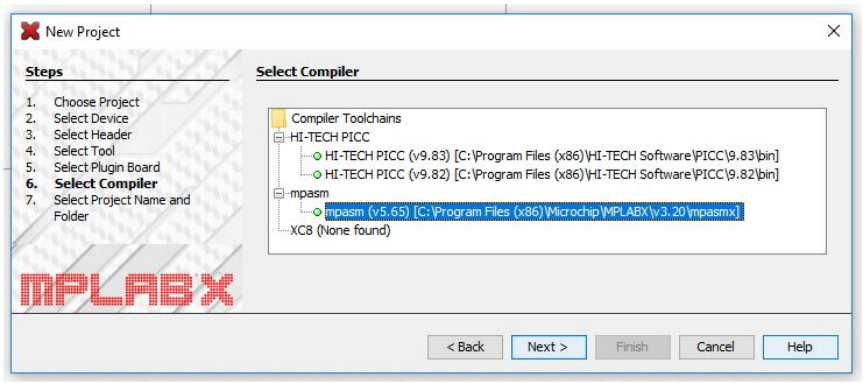

Figura 10

Step 7_*Select Project Name and Folder* (figura 11)
Viene scelto il nome del progetto e la posizione della cartella che contiene tutti i file di progetto.

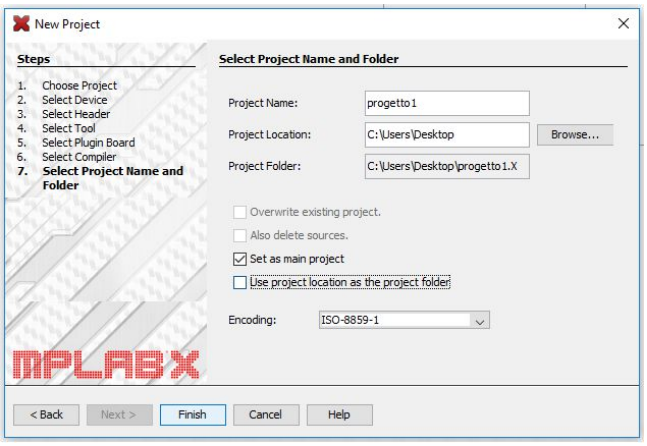

Figura 11

Una volta definito il progetto compare sulla sinistra la finestra **Projects** (figura 12) che contiene le cartelle e i file necessari per sviluppare il medesimo.

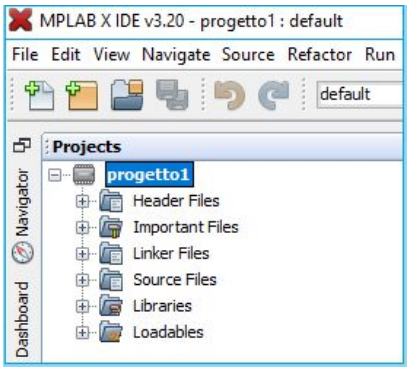

Figura 12

Creazione di un file sorgente

A questo punto è possibile aggiungere o creare i file sorgenti (*Source File*) selezionando di seguito **File** e **New File** dal menu principale (figura 13).

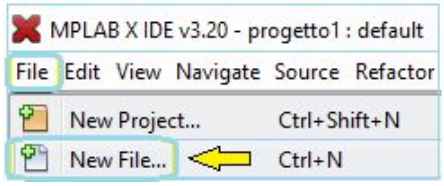

Figura 13

La creazione di un file sorgente prevede due step successivi e due conseguenti finestre.
Selezionando New File si apre la prima finestra che consente la scelta del tipo di file (*Choose File Type*).
In questo caso si è scelto di programmare in Assembler (da elenco *Categories*); il file sorgente ha estensione *.asm* (da elenco *File Types*).

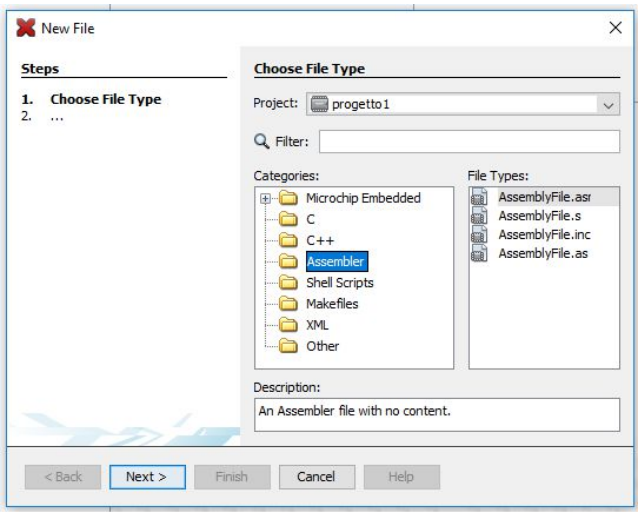

Figura 14

La seconda finestra definisce il nome del file e la sua posizione all'interno del progetto (*Name and Location*).

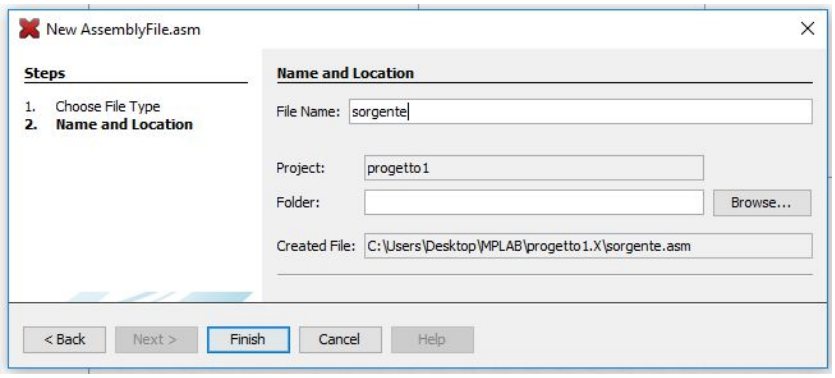

Figura 15

Il file sorgente che si trova ora nella cartella di progetto scelta in precedenza deve essere successivamente trasferito in *Source Files* della finestra *Projects*.
Si deve evidenziare la cartella e, utilizzando il tasto destro del mouse, selezionare il comando *Add Existing Item* (figura16).

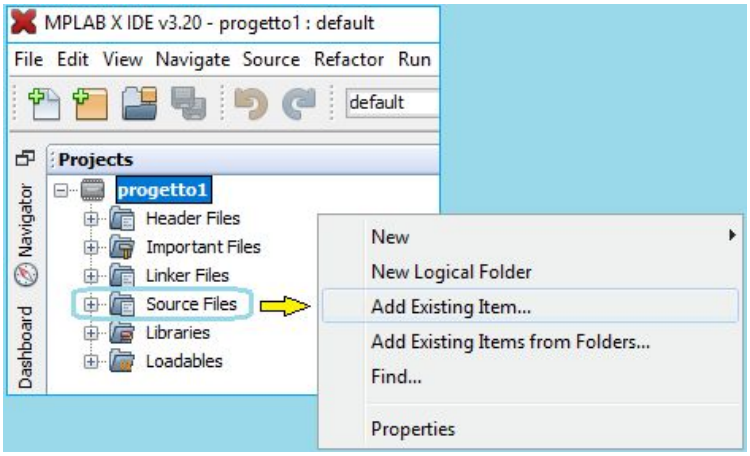

Figura 16

Il trasferimento del file avviene selezionandolo dalla finestra *Select Item* che compare di seguito (figura 17).

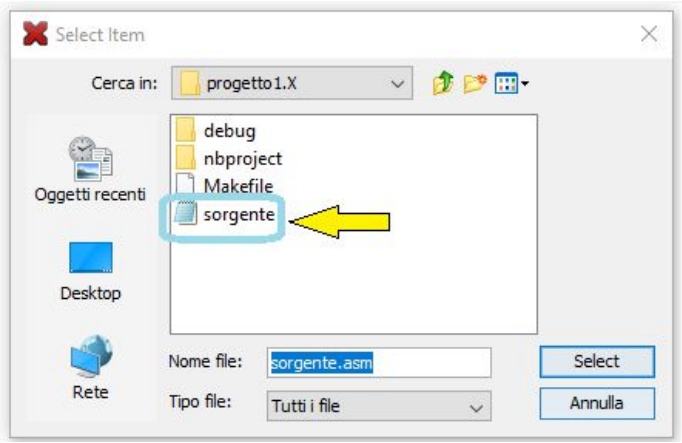

Figura 17

Il file sorgente si trova a questo punto all'interno della cartella Source Files (figura 18).

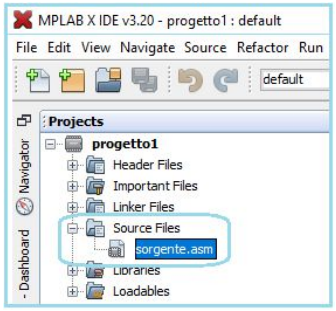

Figura 18

Un esempio di file sorgente in linguaggio Assembler viene di seguito riportato (figura 19).

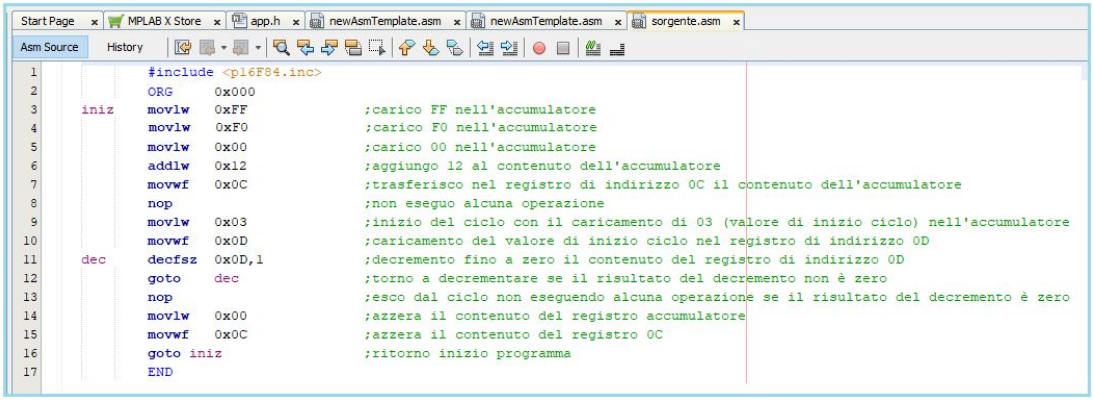

Figura 19

Impiego del simulatore

Con la creazione del progetto (step 4) è stato scelto come strumento hardware (*Hardware Tool*) il simulatore (*Simulator*).
Per procedere con la simulazione si deve selezionare l'icona *Debug Project* (figura 20).
Questa prima operazione serve per verificare la presenza di errori di sintassi.
Al termine di questa operazione nella parte in basso a sinistra del video (finestra *Output*) compare la dicitura *Build successful* nel caso di assenza di errori, la dicitura *Build failed* se invece sono contenuti degli errori dei quali viene specificata la natura.
Se non sono presenti errori si può procedere con la simulazione (nelle modalità *Step Into* e *Continue*) utilizzando le icone che compaiono di lato a *Debug Project* (figura 20).
Una *sessione di debug* ha termine selezionando *Finish Debugger Session*.

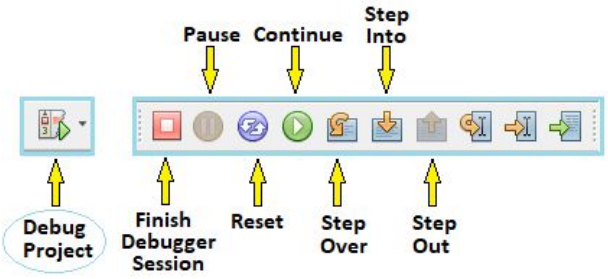

Figura 20

Modalità di simulazione
La simulazione può essere effettuata nelle modalità *Step Into* e *Continue*.

*Modalità Step Into_*
L'esecuzione del programma si interrompe ad ogni riga con la possibilità di controllare lo stato di registri e variabili.
L'avanzamento riga per riga avviene facendo click su *Step Into*.
Selezionando *Reset* il cursore si riposiziona sulla prima istruzione.

*Modalità Continue_*
L'esecuzione del programma avviene in modo continuo.
Può essere interrotta selezionando *Pause* e riavviata dal punto in cui era stata interrotta facendo click nuovamente su *Continue*.
Dopo aver selezionato *Pause* si può scegliere di far ripartire la simulazione dalla prima istruzione selezionando *Reset*.

Visualizzazione dei risultati della simulazione
I risultati della simulazione possono essere visualizzati:

- in formato numerico;
- graficamente.

Si ha accesso ai risultati selezionando *Window>Simulator o Window>PIC Memory Views* dal menu principale (figura 21).

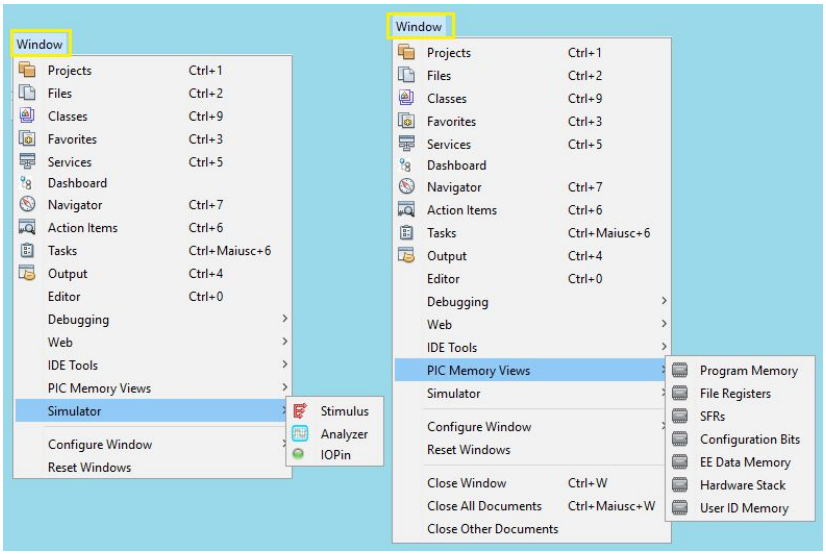

Figura 21

I risultati della simulazione sono visibili nella corrispondente finestra di lavoro che si apre dopo la selezione.

*Visualizzazione in formato numerico_*
Si procede da *Window>PIC Memory Views*.
Si può ad esempio visualizzare (figura 22) il contenuto dei registri speciali (SFRs) come il registro accumulatore, il program counter, i registri TRISA e TRISB.

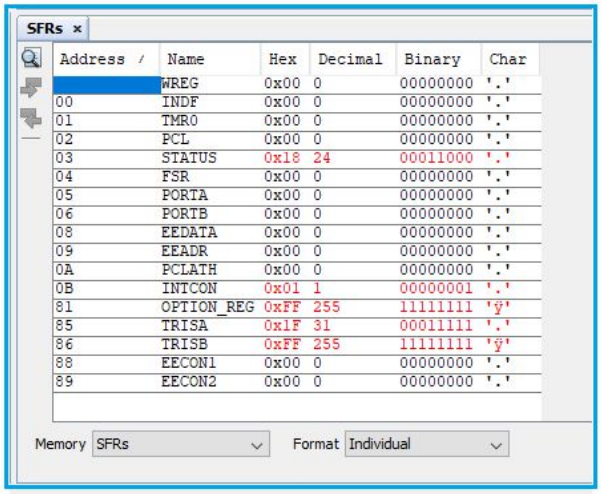

Figura 22

Selezionando *Window>Simulator>IOPin* (figura 23) si può visualizzare lo stato dei segnali presenti sui pin di ingresso e di uscita.

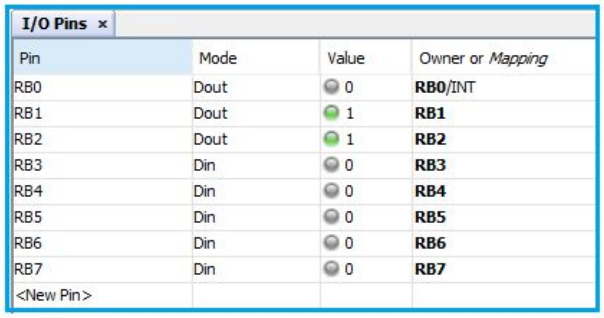

Figura 23

*Visualizzazione grafica_*
L'andamento nel tempo delle degli ingressi e delle uscite può essere visualizzato graficamente selezionando di seguito *Window>Simulator>Analyzer*.
Un esempio viene riportato in figura 24.

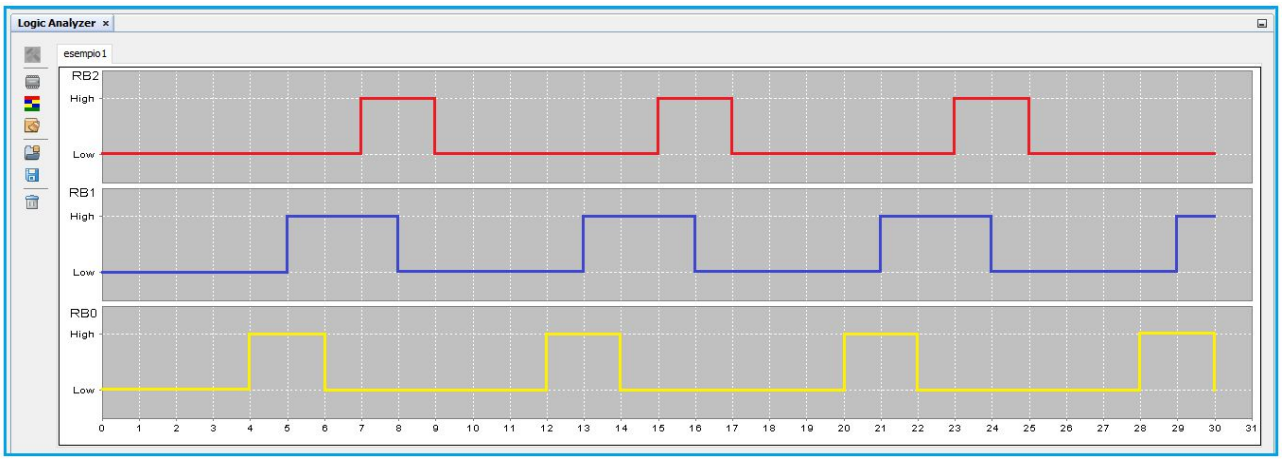

Figura 24

Impiego dei punti di interruzione (*Breakpoints*)_

La simulazione nel modo *Continue* può essere interrotta automaticamente in corrispondenza di punti particolari denominati *punti di interruzione*.

I punti di interruzione vengono definiti facendo click con il cursore sul numero che contraddistingue la riga (esempio in figura 25).

In questo caso ne sono stati definiti quattro che consentono in particolare la visualizzazione del contenuto del registro TRISB nelle quattro situazioni previste dal programma.

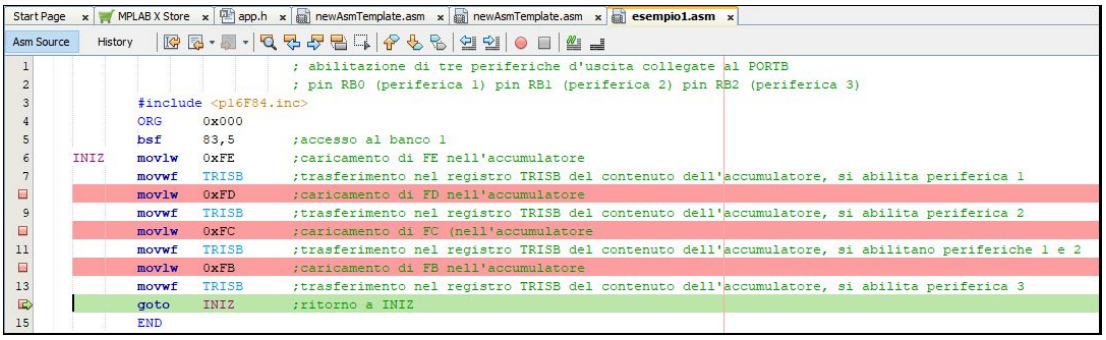

Figura 25

I numeri di riga corrispondenti ai punti di interruzione si riconoscono in quanto, con il click, vengono sostituiti da piccoli quadrati.

I punti di interruzione così definiti vengono salvati automaticamente diventando parte integrante del programma.

I punti di interruzione si eliminano facendo click con il cursore sui quadratini.

L'esecuzione del programma riprende dal punto in cui è stata interrotta quando si seleziona nuovamente *Continue*.

Selezionando di seguito *Window>Debugging>Breakpoints* (figura 26) si accede alla finestra riportata in figura 27 che mostra l'elenco dei punti di interruzione in precedenza definiti.

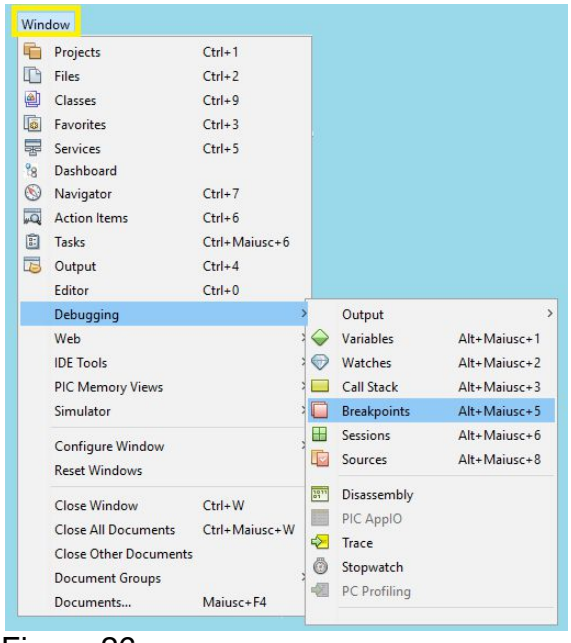

Figura 26

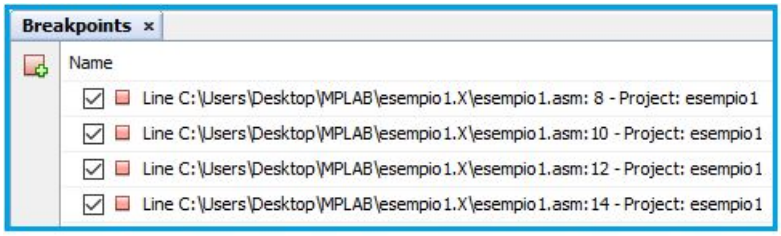

Figura 27

Impiego degli stimoli (*Stimulus*)_

Gli stimoli sono degli strumenti messi a disposizione per simulare il comportamento del programma in relazione a specifici ingressi quali possono essere quelli dovuti a pulsanti, interruttori, segnali di clock.

Si consideri come esempio il programma riportato in figura 28.

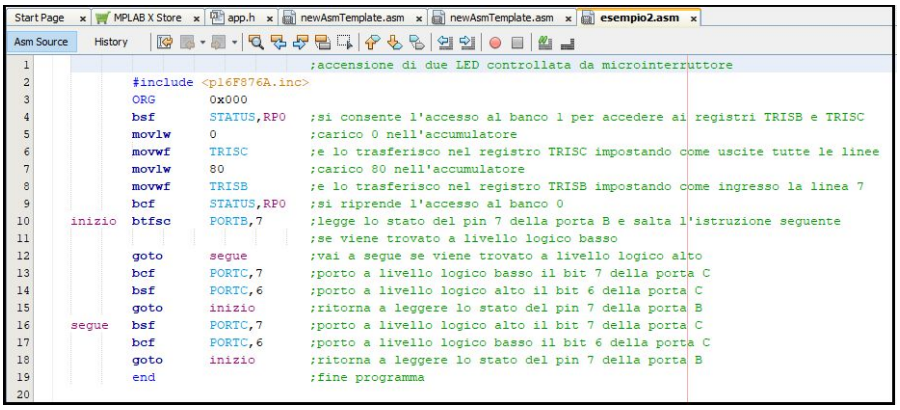

Figura 28

Alla porta B_pin 7 viene collegato un microinterruttore.

In relazione allo stato del microinterruttore si possono accendere/spegnere due LED collegati alla porta C_pin 6 e 7.

Se lo stato del pin 7 della porta B è a livello logico alto (interruttore chiuso) si accende il LED collegato al pin 7 della porta C; viceversa (interruttore aperto), se è a livello logico basso, si accende il LED collegato al pin 6 della porta C.

Gli stimoli vengono definiti utilizzando la finestra *Stimulus* che si apre seguendo il percorso *Window>Simulator>Stimulus* (figura 29).

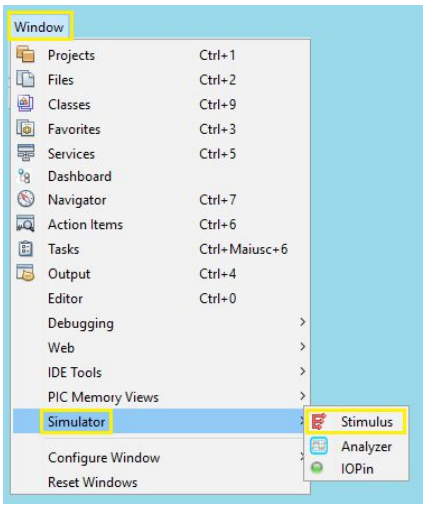

Figura 29

Nel caso dell'esempio (figura 30) viene definito uno stimolo di tipo asincrono (*Asynchronous*) sul pin RB7.

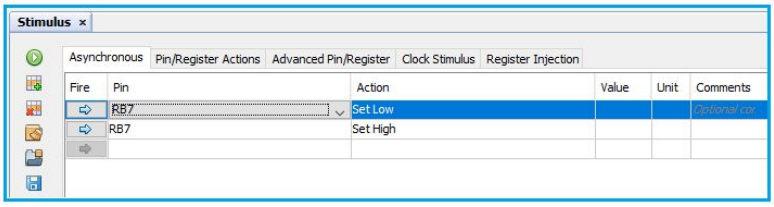

Figura 30

Lo stimolo viene reso attivo facendo click con il mouse sulle frecce (*Fire*) nel corso della simulazione; il pin può essere portato (*Action*) a livello basso (*Set Low*) o alto (*Set High*). Come conseguenza della scelta operata viene modificato il contenuto del registro PORTC (figura 31) che vale 64 nel caso di interruttore aperto, 128 nel caso di interruttore chiuso.

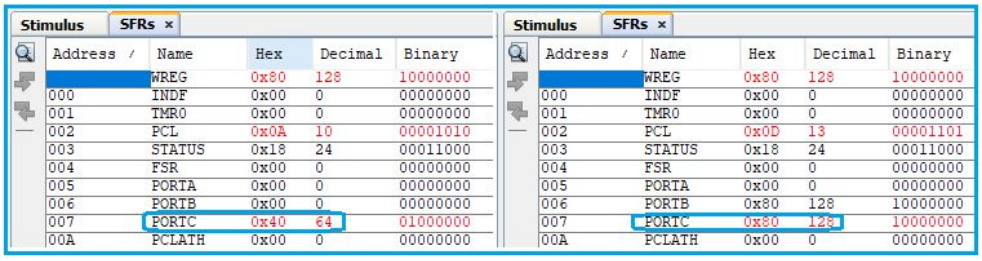

Figura 31