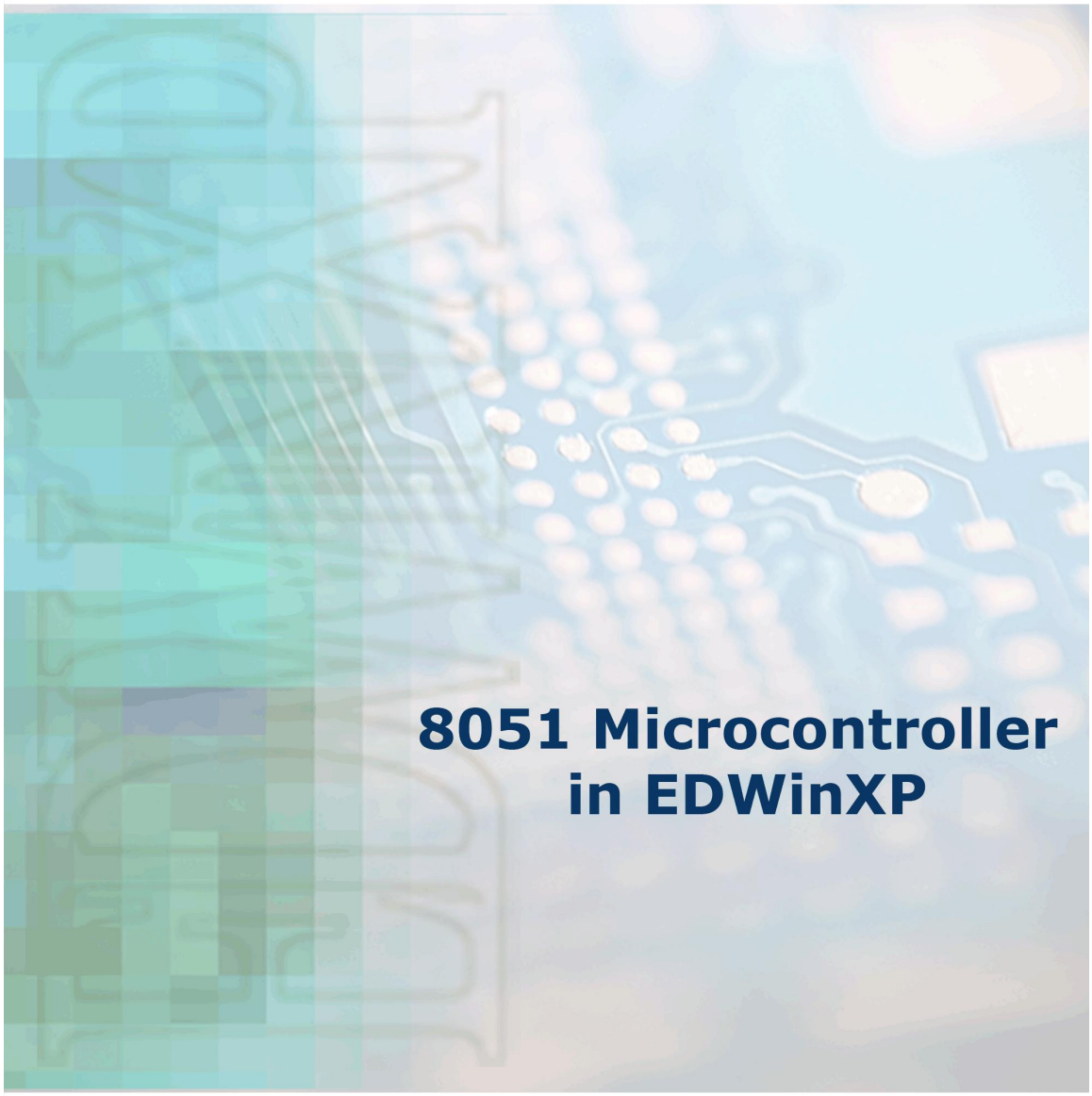
EDWinXP



8051 Microcontroller in EDWinXP

VISIONICS

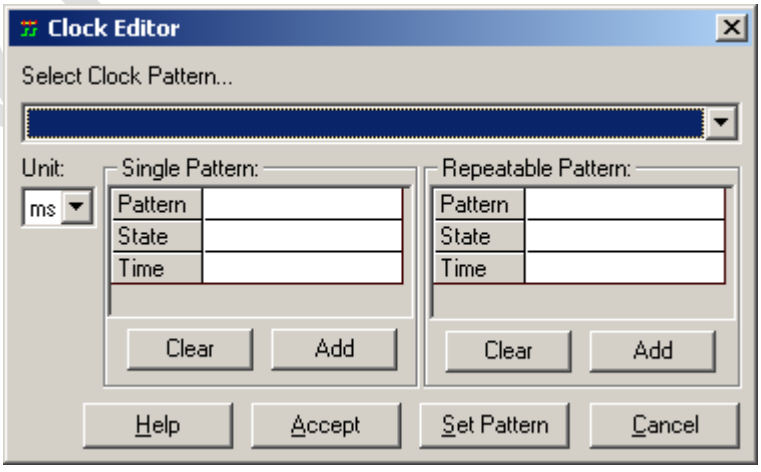
Contents

8051 Microcontroller in EDWinXP

The 8051 microcontrollers in EDWinXP help the users to build real-time circuits. The 8051 can be interfaced with external devices like seven segment displays, interrupt generator, serial/pattern generator etc.

Using the 8051 microcontroller in EDWinXP

1. Open EDWinXP Main from **Start→ Programs→ EDWinXP →Ver x.xx→ EDWinXP Main.**
2. Open the **Schematic Page (Mixed-Mode Simulator)** by double clicking on the **Page {MAIN PAGE}** under **Project→ Circuit [MAINHIER] →Diagram→ Page {MAIN PAGE}.**
3. Load 8051 microcontroller and Preprocess it.
4. The necessary timing for the controller is given by connecting a crystal across the pins XTAL1 and XTAL2. The timing can be provided manually by applying a clock of the desired frequency of the crystal to the pin XTAL1/XTAL2. This is done by selecting **Tools→ Instruments → Preset Logic → Clock Generator**
5. Connect nodes on **XTAL1/XTAL2**
6. Now by clicking on either of the two nodes **XTAL1/XTAL2** the following window appears



7. Provide the desired frequency,

Eg: if the crystal frequency is 6MHz, then the time period will be $(1)/(6x10^{6})$ i.e. $0.1667\ x10^{-6}$

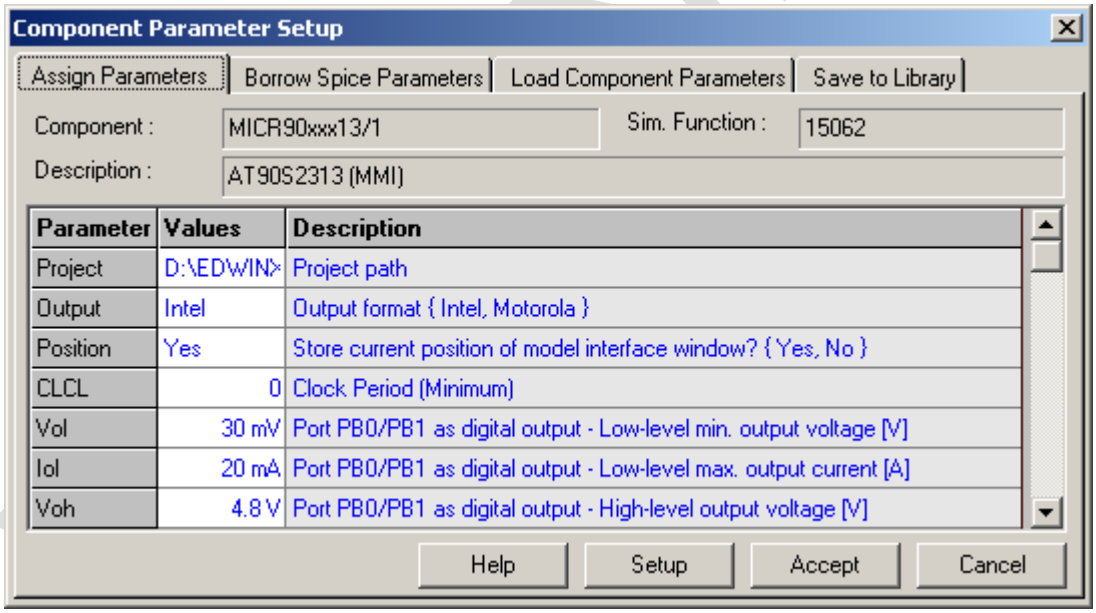
The clock fed to the XTAL pin is {L1us, H1us} as **Repeatable Pattern**

8. Click on Add → **Set Pattern** → **Accept**

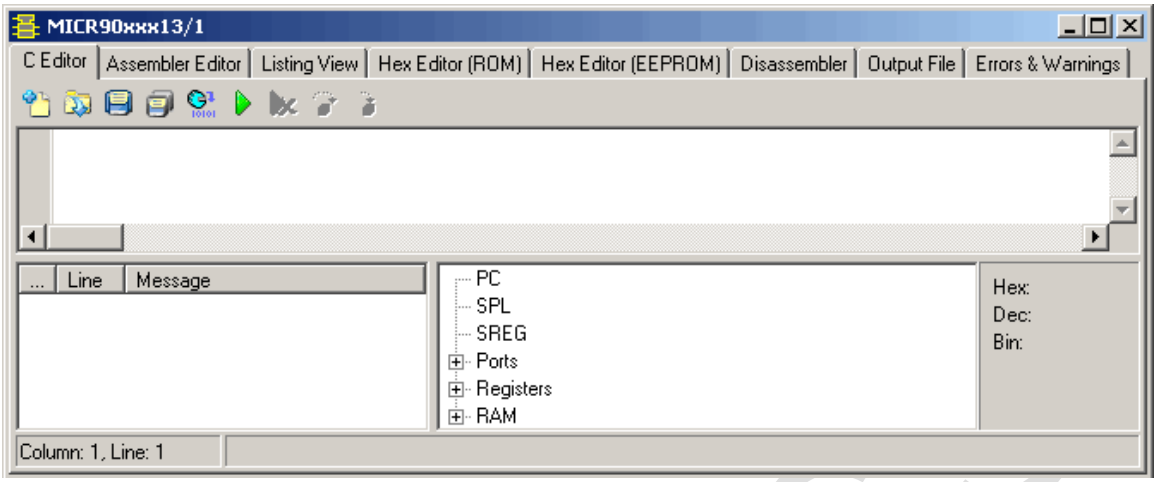
Writing code for 8051 microcontroller

1. Select **Components** from the **Tool** Menu.

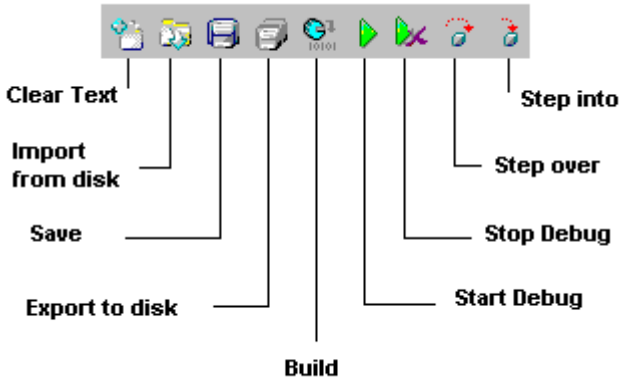
2. Choose **Component Properties** (second Function tool) → **Change Simulation parameters** (second option tool) and **Click** on the **8051** microcontroller. A new window pops up.



3**. Click Setup** tab, the code editor window pops up. Code can be written in this window. Code can be written either in C language or assembly language.



The tool bar in the code editor window is shown below



Clear text - Clears the source code in the code editor window.

Import from disk - Displays the 'Open' dialog box and allows loading a new text file

Save - Allows to save the source code.

Export to disk - Displays the 'Save' dialog box and allow to save the source code.

Build - Enables compilation of the source code.

Start Debug - Enters the Debug mode.

Stop Debug - Terminates debug mode.

Step over - Generates an event to notify model to step over the currently debugged part of the code.

Step into - Enables to view the line by line execution of the source code in debug mode.

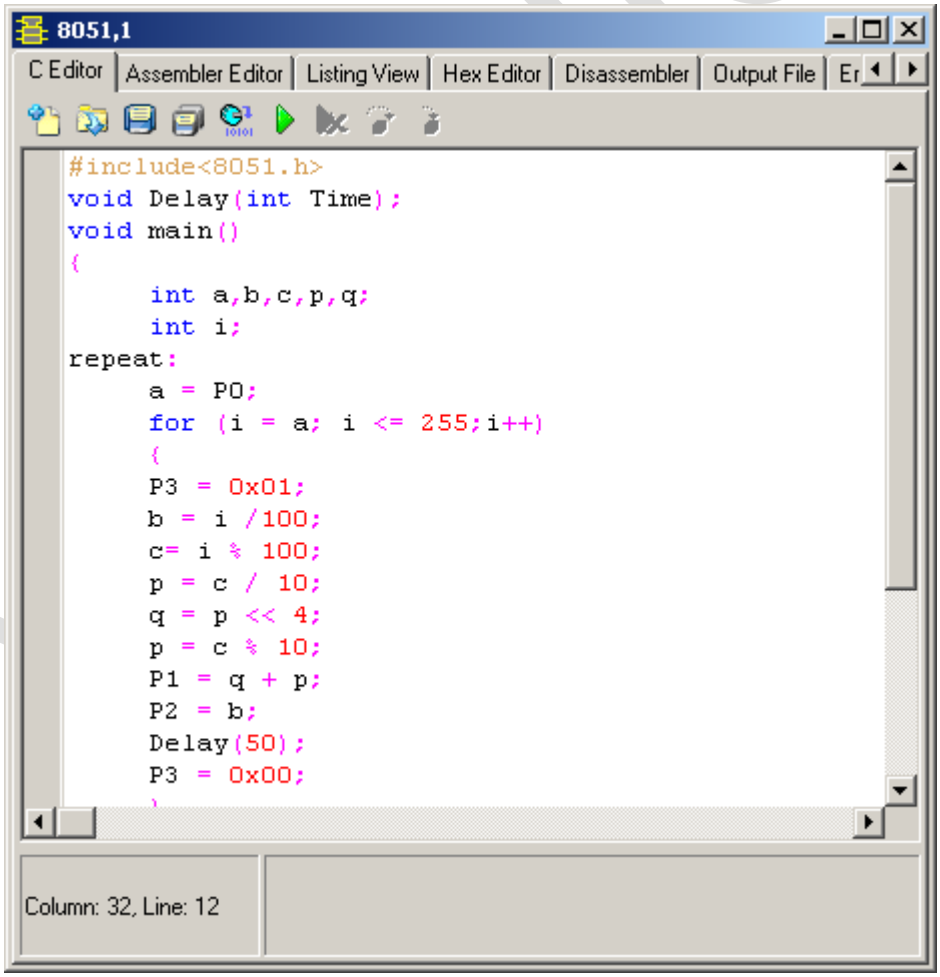
Writing source code in C language

Basic structure for writing code in **C** language is

```
#include<8051.h>
void main ()
{
}
```

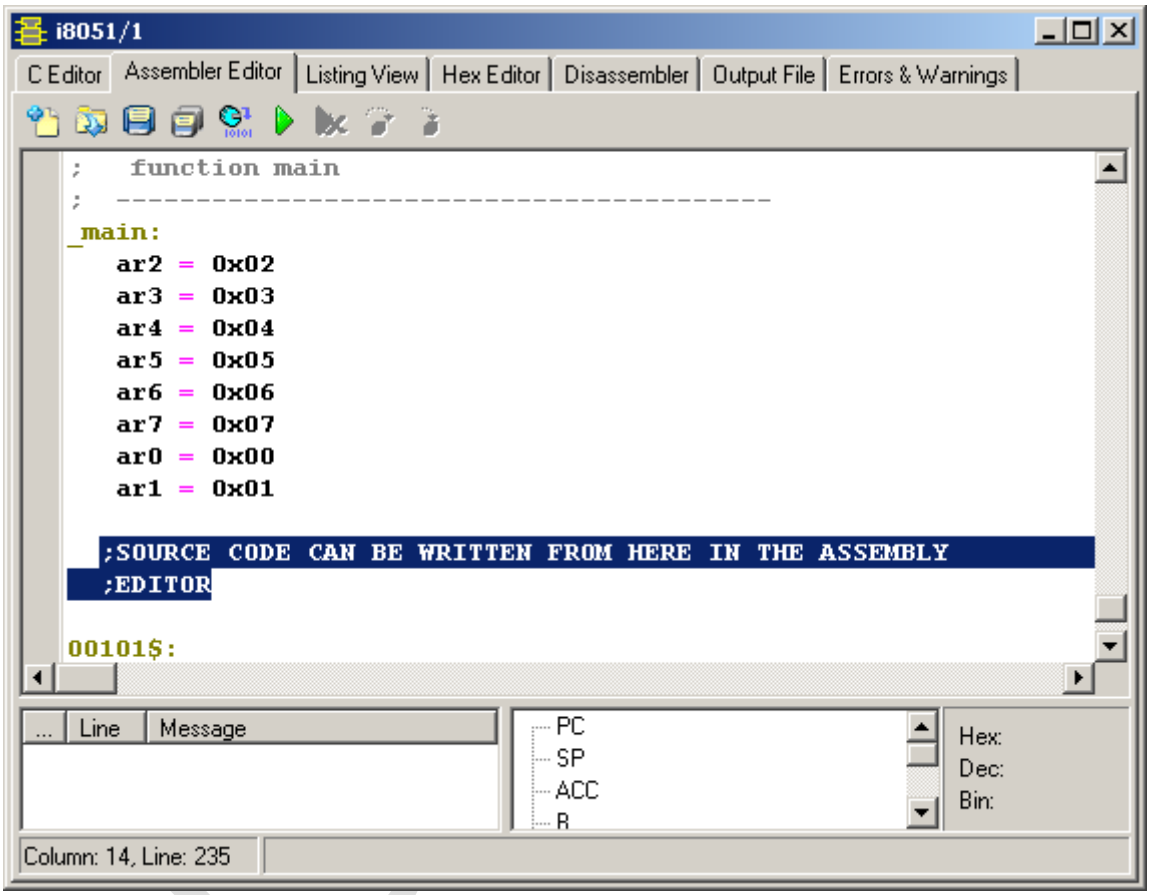
The code is written within the **main ()** function. For better understanding, the case of 8051**_Counter** is taken. Load **8051_Counter** and the steps mentioned till now in **Writing code for 8051 microcontroller** are executed.

The code for **8051_Counter** is as follows:



By compiling (using the **Build** button) the basic structure in the **C Editor,** the default structure in the **Assembly Editor** will be obtained.

The code is written within the **_main function**. The location to write the code is shown in the figure below.



The code in **Assembly Editor** is also compiled by using the **Build** button in the code editor Window.

Compilation messages will be shown under the **Line Message column**.

After successful compilation **click** the **Save** button and close the code editor window.

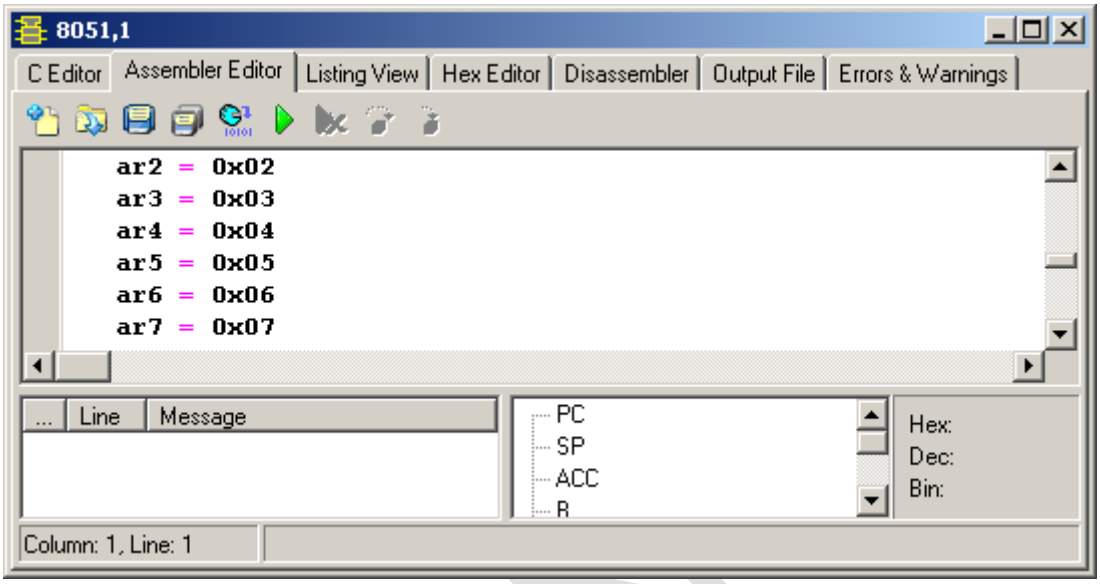
Click Accept in the **Component Parameter Setup window.**

After completing the schematic the circuit is **Preprocessed** and **Analysis** is **Run.**

Debugging the code

Place breakpoints to keep **track of program execution**.

Breakpoints can be placed by clicking on the gray space on left of the code statements.



Note: *To remove **Breakpoints** right click on them.*

Click on **Start Debug**  button in the code editor window. This will turn on the debug mode.

Close the **Code Editor** window and click **Accept** in **Component Parameter Setup.**

Go to **Schematic Editor** and **Run Analysis,** the code editor window pops up.

The program can be debugged using the **Step Over**  (switching between breakpoints) and **Step Into**  **(**step by step execution**)** buttons.

By clicking on the required ports and registers the changes in them can be viewed during the process.

After debugging the **Stop debug button**  is clicked to exit the Debug mode. The Analysis continues till simulation time ends.