



DOMAIN TECHNOLOGIES

8051 Quick Start Guide 1.1

BoxView IDE

Integrated Development Environment

Getting Started with BoxView IDE

This quick start guide to BoxView IDE will provide users with steps to using their development environment: create a simple application, define a debug session, start the simulator, and view project debug windows. Throughout the following pages, each user step is denoted with a button:

- Sample User Step

A Tutorial and Users Guide are also available within BoxView IDE; access is provided using BoxView IDE menu option Help->Help Contents.

Development Environment Overview

BoxView Integrated Development Environment (IDE) features the high level language, embedded processor target debugger as part of the Eclipse software development environment. The Eclipse platform is an open, industry supported, extensible, software development platform.

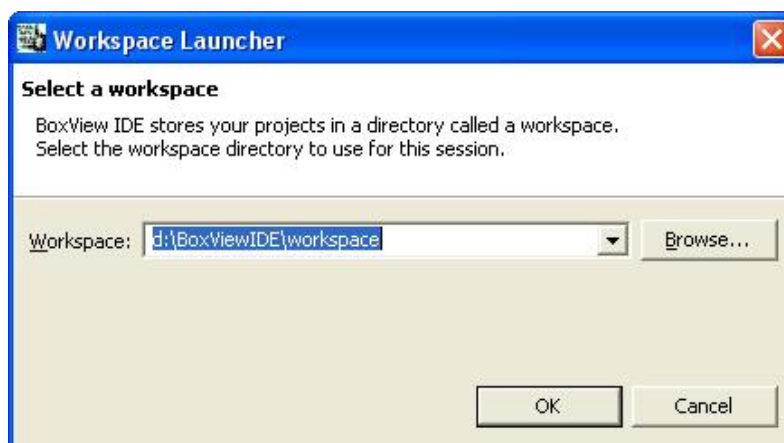
- After installing and registering the product, load BoxView IDE from the desktop.

Several windows are displayed, including a navigator window which shows folders containing sample projects. There are three basic types of resources that exist in the Development Environment; a resource is a collective term for the projects, folders, and files viewed in the IDE:

Files	Files in the file system
Folders	Directories on a file system. Folders are contained in projects or other folders. Folders can contain files and other folders.
Projects	Contain folders and files. Projects are used for builds, version management, sharing, and resource organization. Like folders, projects map to directories in the file system. (When you create a project, you specify a location for it in the file system.)

The directory location on your computer where the resources will be stored is called a workspace. The sample projects are stored in the default installed location. The Development Environment supports multiple workspaces.

- Create a new workspace by selecting *File->Switch Workspace...* from the main IDE menu.
- Use the *Browse* button to define the directory that will hold your project files and folders:



Windows in BoxView IDE

The BoxView IDE contains many windows or views. A set of views is called a perspective. The IDE has 2 predefined perspectives: the *Project Perspective* and the *Debug Perspective*. The *Project Perspective* displays a set of windows valuable when manipulating project files. The *Debug Perspective* displays windows valuable when debugging a project. Users can change perspectives by choosing menu option *Window->Open Perspective* or by using the associated button on the IDE's button bar:



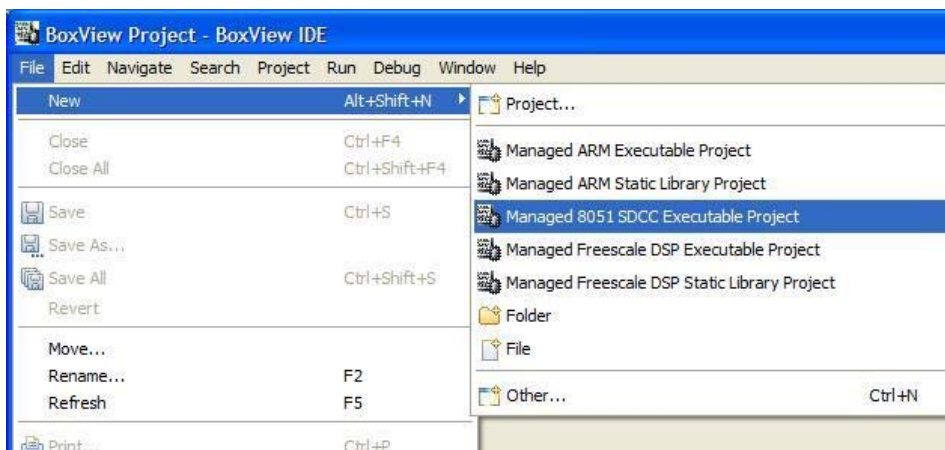
Users can add windows to any perspective by using the IDE's *Window* pull-down menu. Users can close windows in a perspective by selecting the window's close icon:



Create a Project

The Navigator window provides a hierarchical view of the resources in your workspace; users can open files for editing by double clicking on the filename in the navigator window.

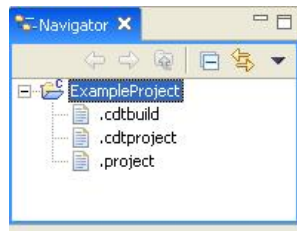
- Right-click with your mouse in the Navigator to open a pop-up menu that allows operations such as create new, copy, paste, import, refresh, etc.
- Choose *New->Managed 8051 SDCC Executable Project*:



The New Executable project dialog will be displayed.

- Define the new project name in the entryfield; for this example type *ExampleProject*
- Choose the *Finish* button.

The Navigator will now contain your new project. Double click on the *ExampleProject* in the Navigator window. Three files are created automatically by the IDE and are contained in the project:



Create a Source File

Create a source file in your new project that will contain a simple program.

- Select *ExampleProject* and choose *File->New->File* from the IDE main menu.

The New File dialog will be displayed.

- Type *simpleExample.c* in the file name entry field.
- Choose the *Finish* button.

An empty source file named *simpleExample.c* has been added to the *ExampleProject* project. The empty source file is opened in the IDE. Write the content of the new application.

- Fill the *simpleExample.c* editor window with the following instructions:

```
int val;
int val2;

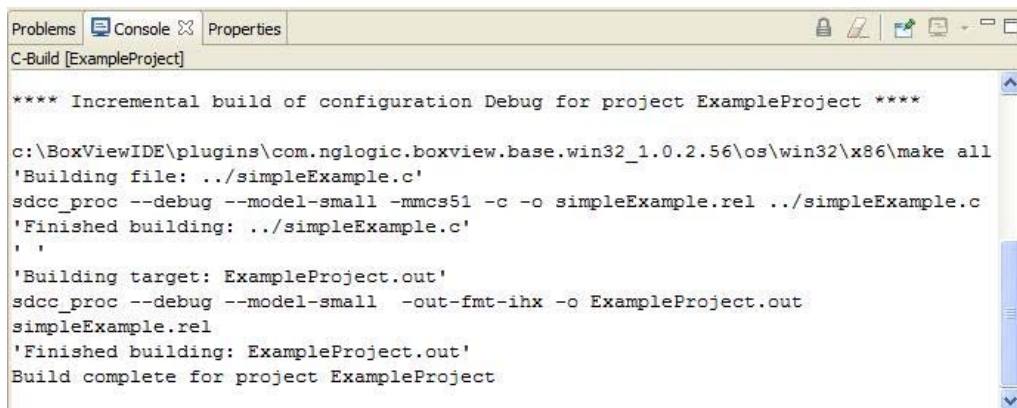
int test (int v)
{
    return v * 2;
}

int main(void)
{
    val++;
    while (val < 100)
    {
        val2 = val / 2;
        val = test(val2);
    }
    return 0;
}
```

- Save the *simpleExample.c* program by choosing *File->Save* or using the floppy disk icon.

Anytime a source file is saved, the automatic build operation will be triggered.

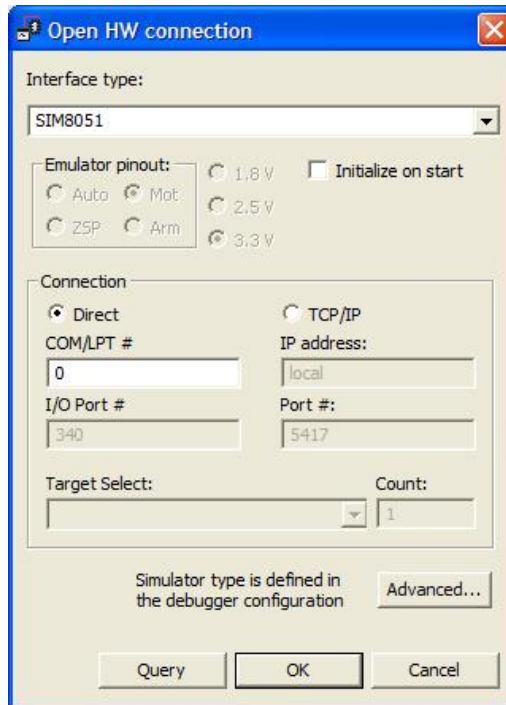
- Select the *Console* window for detail build information results:



Configure the Debug Session

Establish a connection to the target hardware and configure the debug session.

- Open the connection to the target, by choosing menu option *Debug->Open Connection* or use the *Open Connection* button on the IDE's button bar. Choose an embedded processor simulator for the interface type and press *OK* to connect:

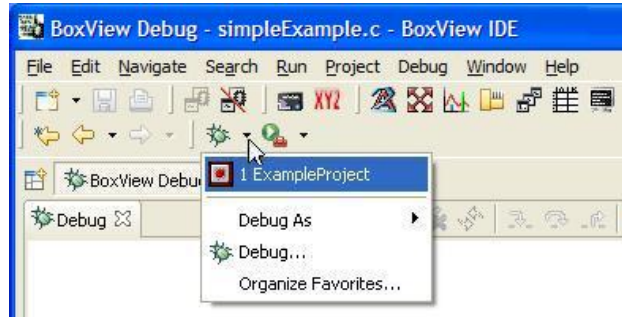


- Change perspectives, to obtain the best views for debugging, by choosing menu option *Window->Open Perspective->BoxView Debug* or use the *BoxView Debug* button on the IDE's button bar
- Select from the main menu *Run-->Debug*
- In the *Debug* session configuration dialog, select *DSP Debugger* in the *Configurations* listbox and choose the *New* button. The *Debug* fields on the right pane will be replaced with fields defining a new session configuration.
- From the *Main* tab of the *Debug* configuration dialog, Name the debug session by typing *ExampleProject* in the *Name* entry field.
- Fill in the *Project* entry field by using the *Browse* button to select *ExampleProject*
- The *Application* entry field will be automatically filled in with the output contained in the chosen project.
- Select the *Debugger* tab of the *Debug* configuration dialog
- In the *Debugger target* drop box, define the connection type for the debug session to be *Emulator*
- Select the *Stop at main() on startup* checkbox.
- Save the session configuration by selecting the *Apply* button

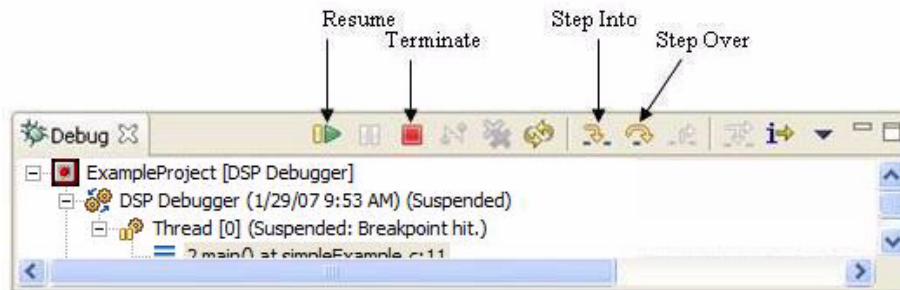
Debugging Your Application

Once the debug session has been configured, the debugger can be started.

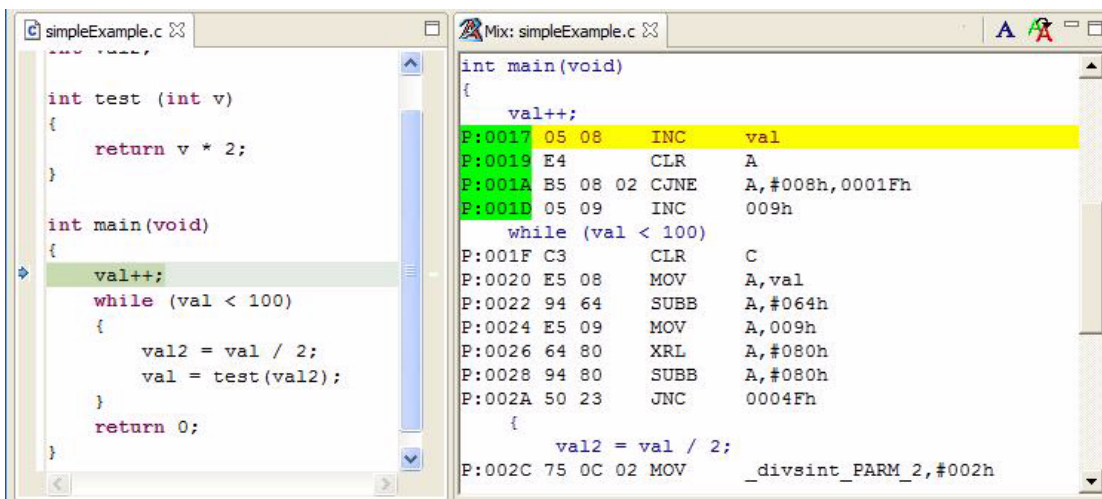
- Start your debug session for this named session by selecting the *Debug* button. Future debug sessions for this project can be started through the debug icon:



Many windows are populated in the Debug perspective when the application is started. In the debug window, note the application has stopped (as defined in our session configuration):

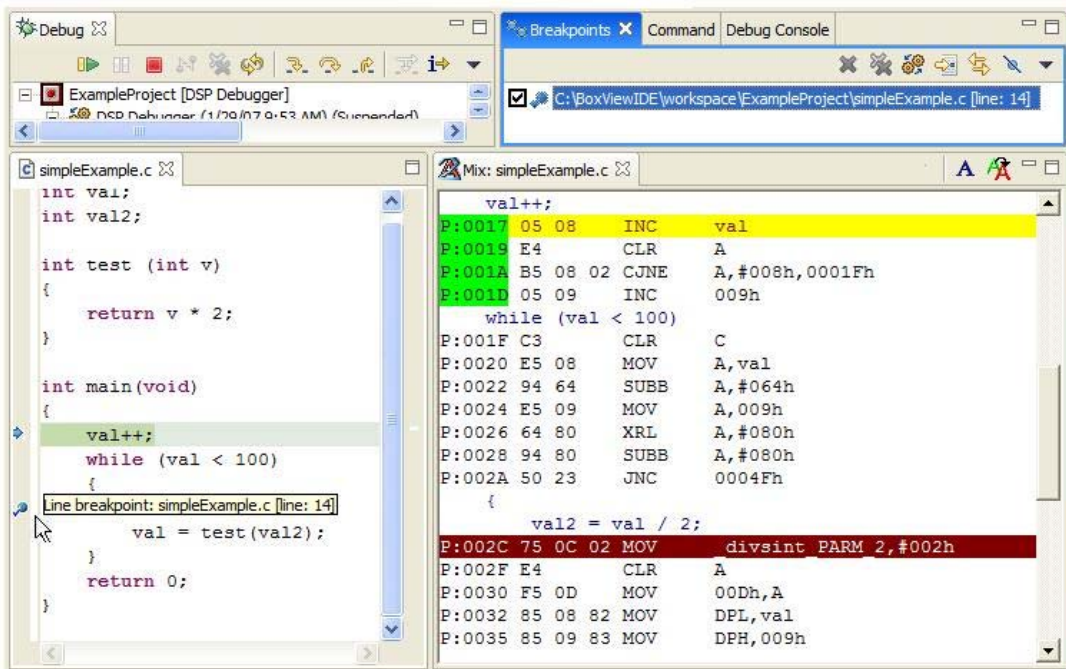


Below, the Source window shows an arrow in the margin for the *Current Program Counter* at the first instruction in the program. The Code Window shows the *Current Program Counter* by highlighting in yellow; the assembler lines representing the *Selected Source Line* are highlighted in green.

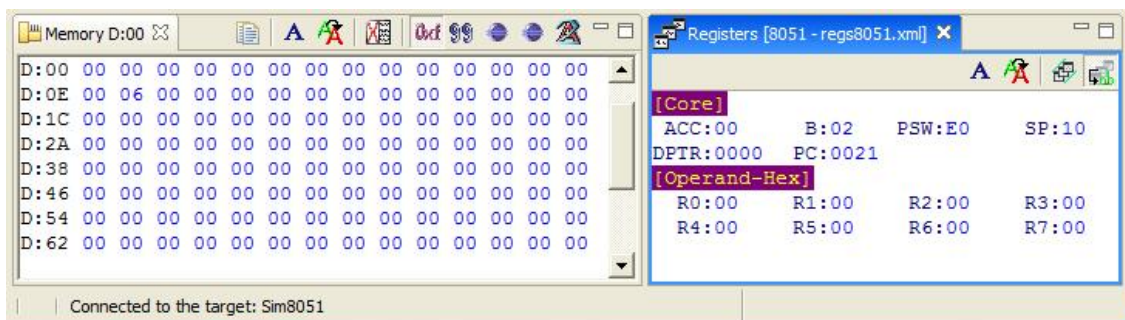


- Set a breakpoint by double clicking in the margin of the source window on the *val2=val/2* instruction.

Below the source window shows the added breakpoint on the selected instruction, the code window shows the breakpoint in burgundy, and the breakpoint window displays the line number of the source file associated with the breakpoint:



- If your debug perspective does not contain a Memory window, add one by selecting *Debug -> Memory* from the main BoxView IDE Window.
- If your debug perspective does not contain a Registers window, add one by selecting *Debug -> Registers* from the main BoxView IDE Window. Perform a right mouse click in the *Registers* window to display the window menu; select the option *load registers->8051*. Again perform a right click in the *Registers* window; select the option *Registers Section-> Core* to display the core registers. Once again performing a right click in the *Registers* window and select the option *load registers->Operand-Hex*.



- Use the *Step Into* button of the Debug window (or IDE menu option *Run->Step Into*) to walk through the project .
- Use the *Resume* button (or IDE menu option *Run->Resume*) to hit the breakpoint
- In the Debug window, use the *Terminate* button (or IDE menu option *Run->Terminate*) to halt the execution.
- Choose *Debug->Close Connection* to close the simulator target hardware connection

For more information on debug windows, review the *Debug Menu Options* Section of the Users Guide.



© Domain Technologies Inc.
811 East Plano Pkwy • Suite 115
Phone 972.578.1121 • Fax 972.578.1086
<http://www.domaintec.com>
email: support@domaintec.com