AVR483: DB101 Firmware - Getting Started

8-bit AVR® Microcontrollers

Application Note

Features

- **Detailed walkthrough**
- **Uses IAR Embedded Workbench**
- **From new project to running code**
- **Uses modules from Application Note AVR482**

1 Introduction

Having all the software modules for DB101 provided by the Application Note AVR®482 is great, but where do you start? What do you need to make your own small application? Surely, you don't want to start out with the demo application that comes shipped with DB101 and break it down.

This application explains, step by step, how to create a new firmware project, add the bare essentials for a basic graphics application, build it and run it on the DB101.

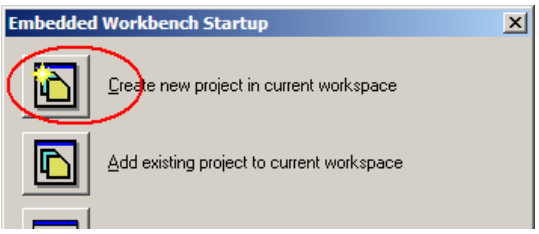
Figure 1-1. The DB101 board.

Rev. 8101A-AVR-09/07

2 Your First DB101 Application

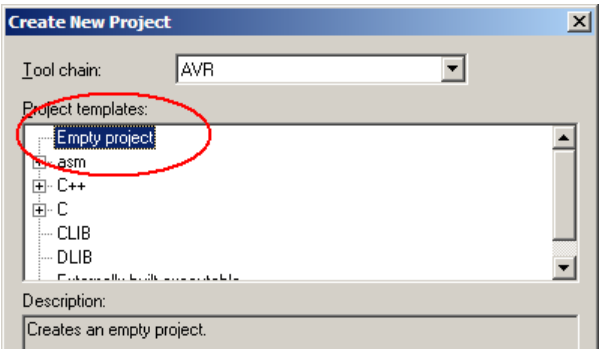
1. First, open IAR Embedded Workbench®. This walkthrough is based on IAR® EW version 4.30A. The 4K code-size limited Kickstart version will do just fine as well. Select “Create new project in current workspace” as shown in Figure 2-1.

Figure 2-1. “Startup” Dialog Box.



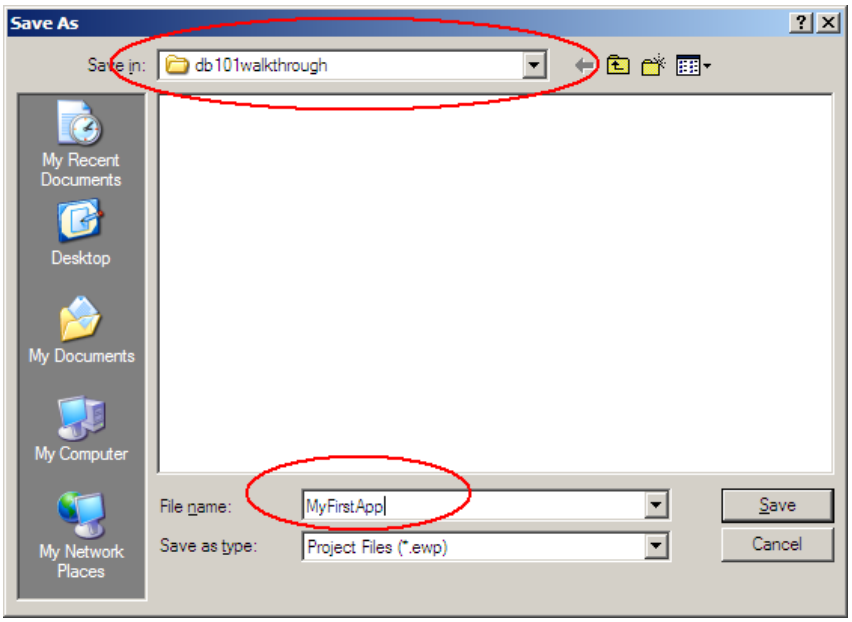
2. The “Create New Project” dialog box will open. Select “Empty project”, as shown in Figure 2-2, and click “OK”.

Figure 2-2. “New Project” Dialog Box.



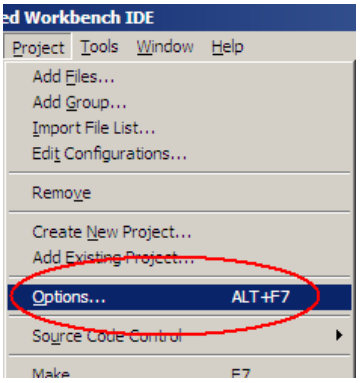
3. IAR will then ask you to select a location and filename for your new project, as shown in Figure 2-3. Create a new folder somewhere and select a name for your project. We've chosen to call it `MyFirstApp`. Click "Save" to move on.

Figure 2-3. "Save Project" Dialog Box.



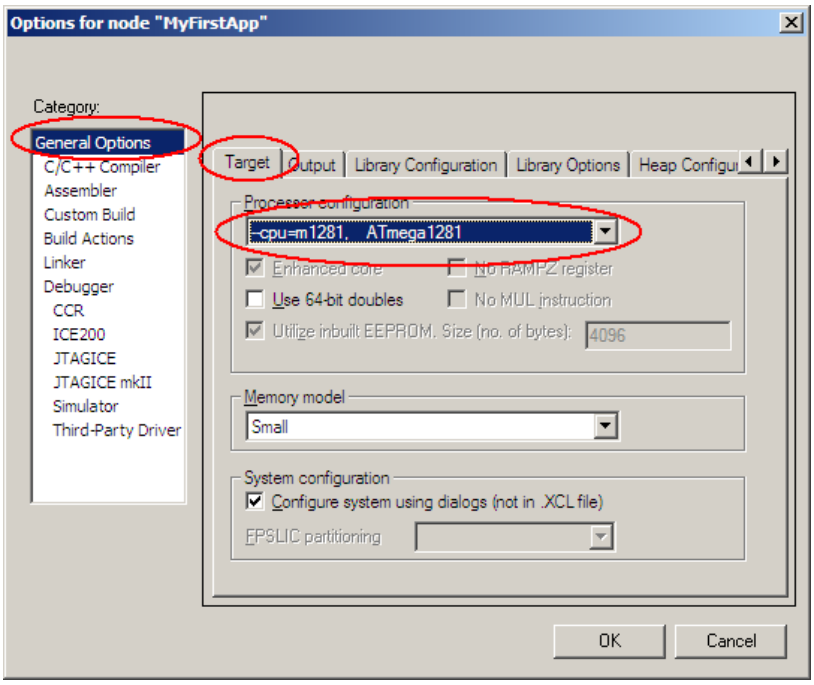
4. You now have a new project in your editor. Before we add files to the project, we need to configure the project options according to the AVR part we are using. Select the "Options" item in the "Project" menu, as shown in Figure 2-4, to open the Project Options dialog box.

Figure 2-4. The "Project" Menu.



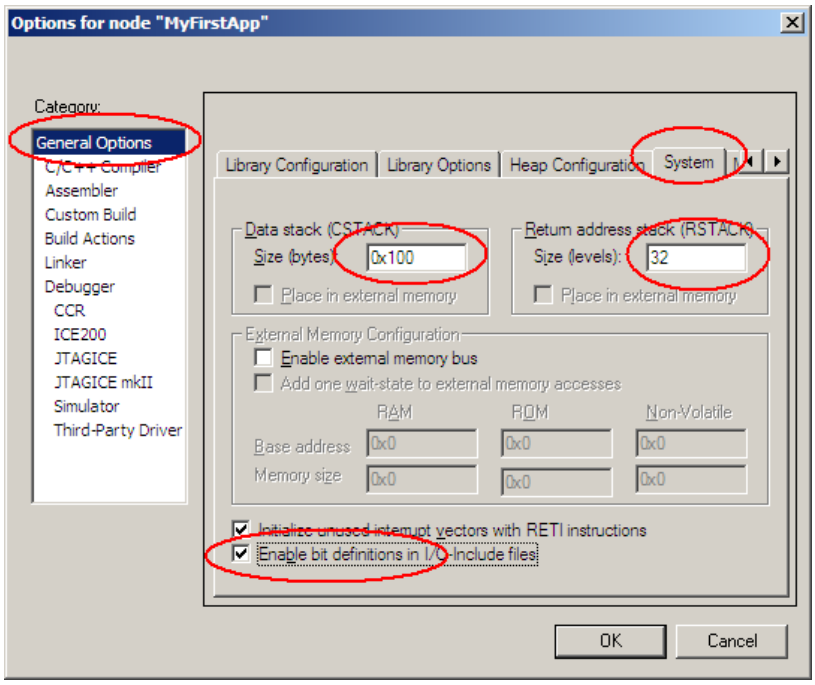
5. Select the “General Options” category, then the “Target” tab. Select the `ATmega1281` device in the drop down box shown in Figure 2-5.

Figure 2-5. “Target” Tab Options.



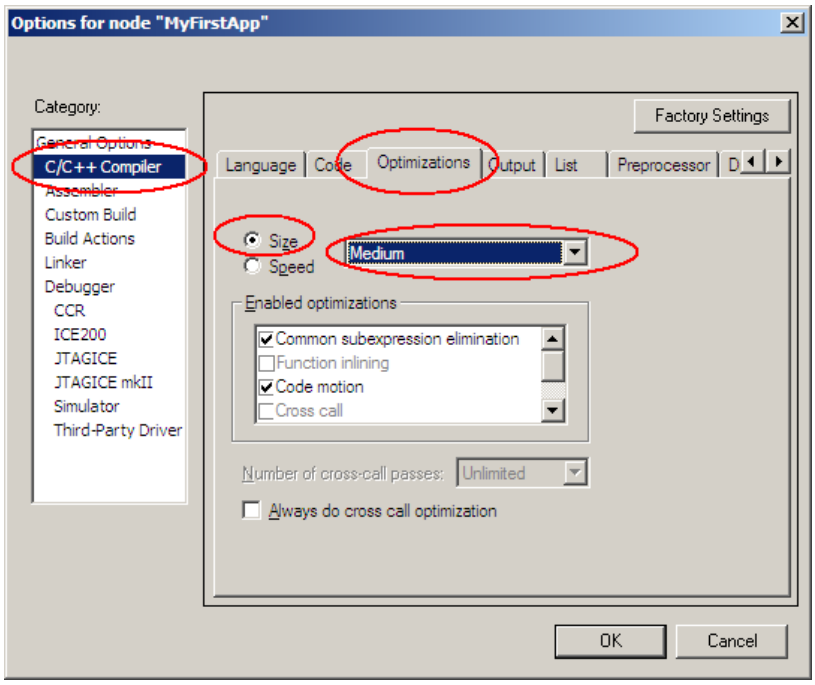
6. Now select the “System” tab in the same category, and make sure the “Enable bit definitions in I/O-Include files” box is checked. Also make sure the “Data stack” and “Return address stack” values are set as shown in Figure 2-6.

Figure 2-6. “System” Tab Options.



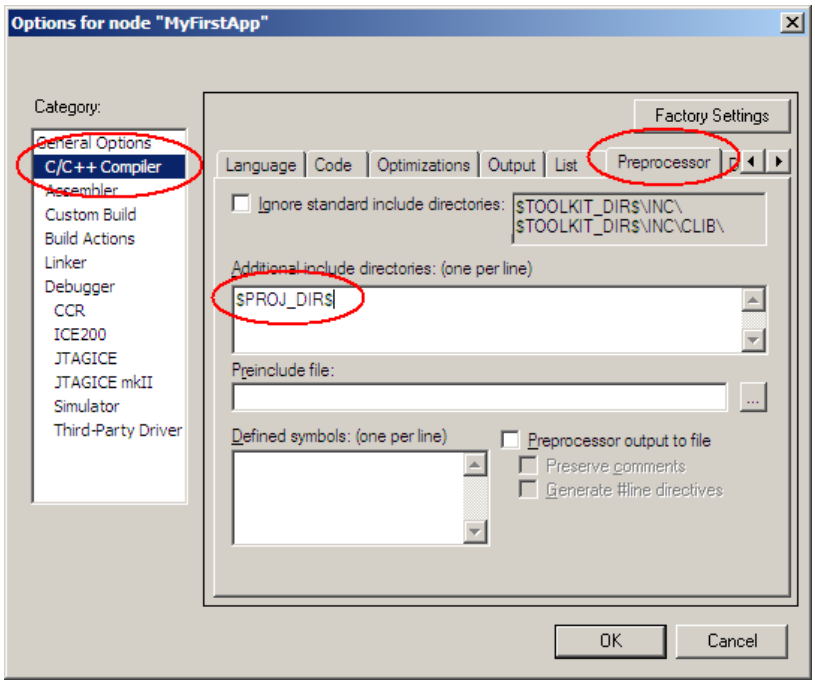
7. Select the “C/C++ Compiler” category, then the “Optimizations” tag. Make sure medium size optimization is selected. The tab is shown in Figure 2-7.

Figure 2-7. “Optimizations” Tab Options.



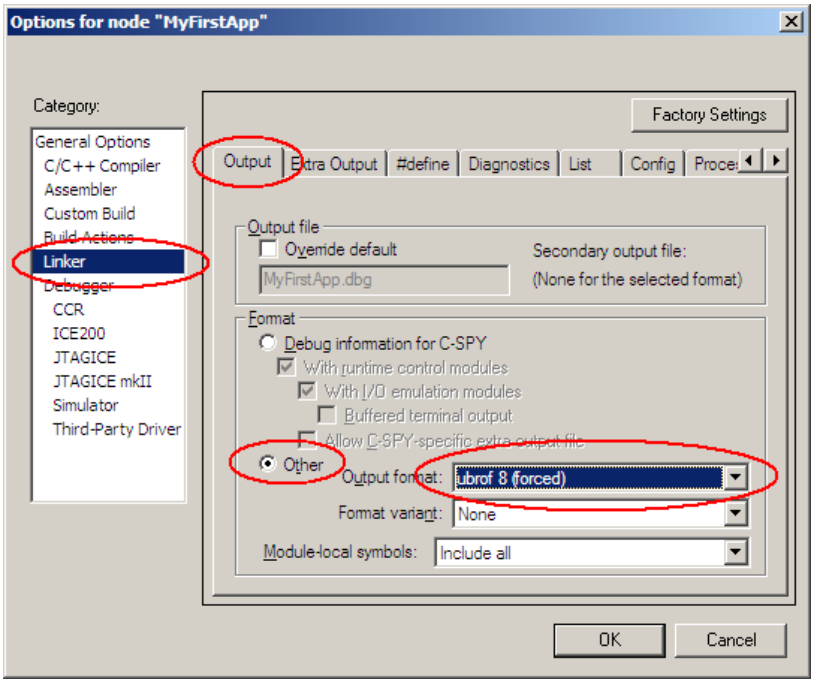
8. Now select the “Preprocessor” tab in the same category, and type “$PROJ_DIR$” in the include directory path box as shown in Figure 2-8. This tells the compiler to look inside the project directory for all the library and driver module files.

Figure 2-8. “Preprocessor” Tab Options.



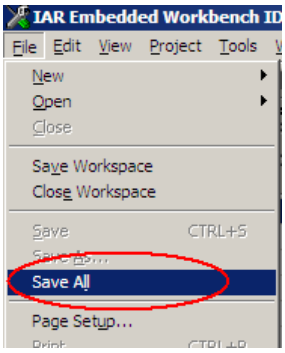
9. Select the “Linker” category, then select the “Output” tab. Make sure the “ubrof 8 (forced)” format is selected, as shown in Figure 2-9. After that, click “OK” to apply your changes.

Figure 2-9. “Output” Tab Options.



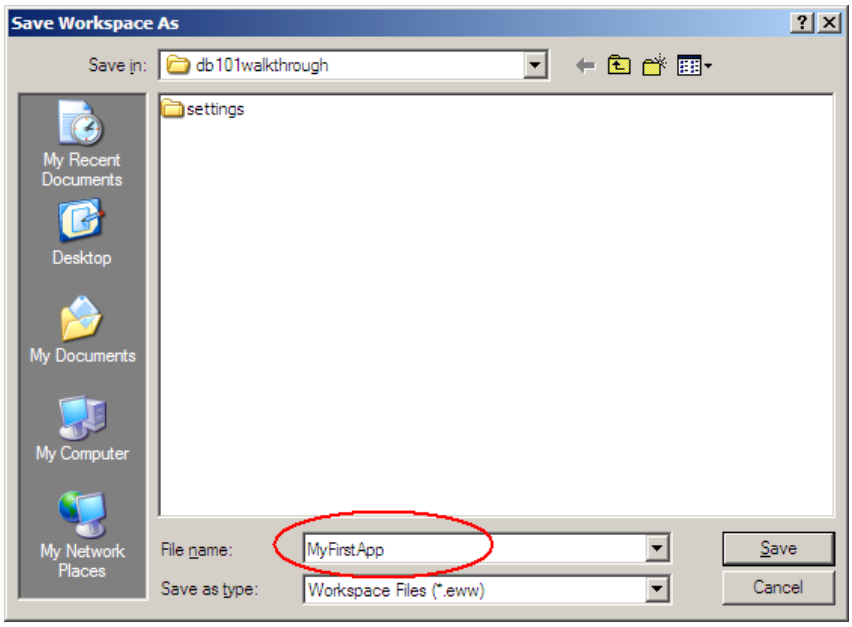
10. This is a good time to save your entire workspace, with project settings and all. Select the “Save All” item in the “File” menu, as shown in Figure 2-10. This will save your project settings and then take you to the “Save Workspace” dialog box.

Figure 2-10. The “File” Menu.



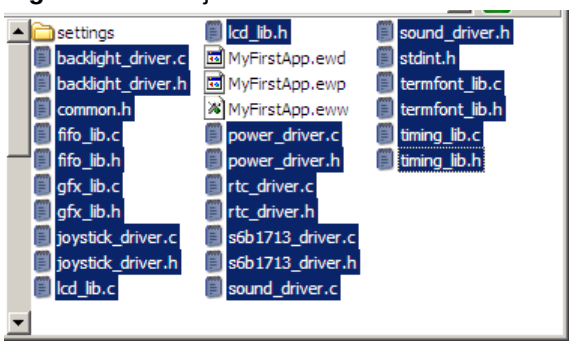
11. IAR will then ask you to select a location and filename for your workspace, as shown in Figure 2-11. We recommend that you locate the workspace file in the same folder as the rest of your project files.

Figure 2-11. "Save Workspace" Dialog Box.



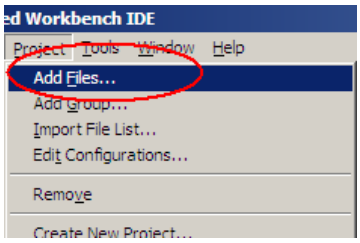
12. Now, we need to copy a few files from the DB101 firmware package to our new project folder. Figure 2-12 shows the folder contents after copying the necessary files. Not all files will be used directly in this demo, but there are dependencies between some modules, so all files shown need to be there.

Figure 2-12. Project Folder Contents



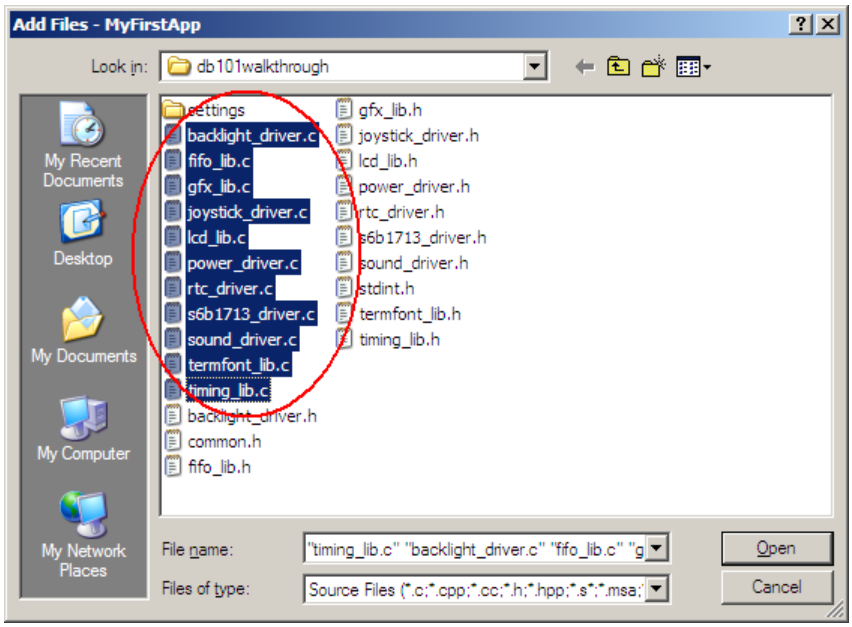
13. Then we need to add the source files to our project. Select the “Add Files” item in the “Project” menu, as shown in Figure 2-13.

Figure 2-13. The “Project” Menu.



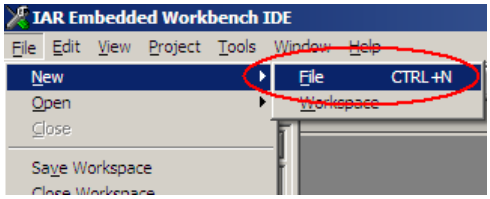
14. In the “Add Files” dialog box, select all source files, that is every file with the “.c” extension, as shown in Figure 2-14. Then click “Open” to add the files to your project.

Figure 2-14. “Add Files” Dialog Box.



15. With all library and driver modules added to our project, we need to create a main source file for our demo application. Select the “New -> File” item in the “File” menu as shown in Figure 2-15. IAR will create a new file named “`Untitled1`” or something similar, and open the file in the editor.

Figure 2-15. The “File” Menu.



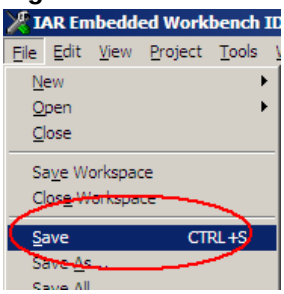
16. Find the “`main.c`” file in the code package that comes with this document and copy the contents into the new file in the editor. The editor should new look as shown in Figure 2-16.

Figure 2-16. File Contents

```
Untitled1 *
1  /*! \file  main.c
2   *
3   * My first application for DB101.
4   */
5  #include "lcd_lib.h"
6  #include "gfx_lib.h"
7  #include "joystick_driver.h"
8  #include "timing_lib.h"
9  #include "rtc_driver.h"
10 #include "backlight_driver.h"
11 #include "common.h"
12
13 #include <ioavr.h>
14 #include <inavr.h>
15
16 //! Timing event used by joystick driver.
17 TIMING_event_t joystickCallbackEvent;
18
19 //! Prototype for our first demo function.
20 void moving_lines_demo( void );
21
22 //! Application starts here.
23 void main( void )
24 {
```

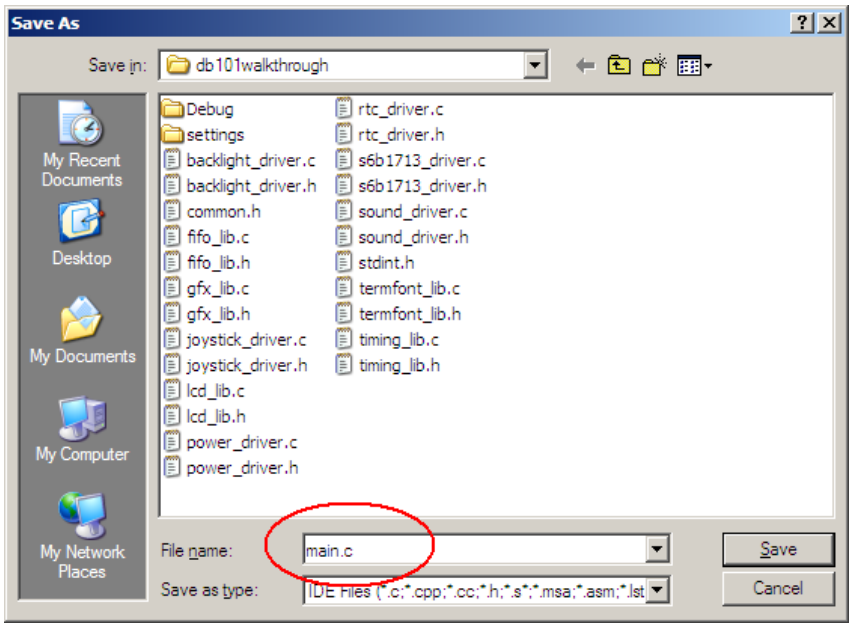
17. Now you should save the new file. Select the “Save” item in the “File” menu as shown in Figure 2-17.

Figure 2-17. The “File” Menu.



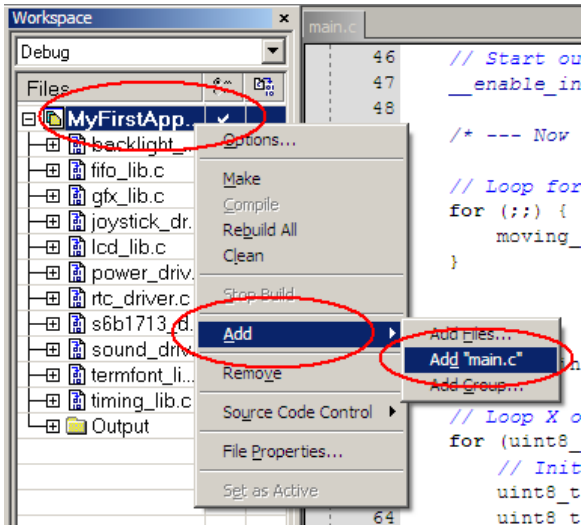
18. Since this is the first time you save the new file, IAR will ask you for a file name. We chose to call it "`main.c`" as shown in Figure 2-18.

Figure 2-18. "Save File" Dialog Box.



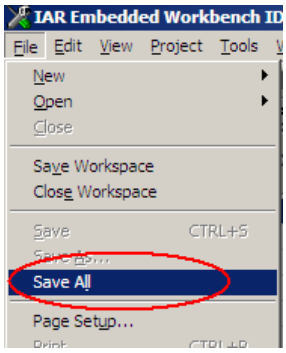
19. Our new source file needs to be added to our project. An easy way to do this, when there's only one new file to add, is to right-click on the project name in the "Workspace" window, select "Add", and then "Add "main.c"" as shown in Figure 2-19.

Figure 2-19. The Right-click Project Menu



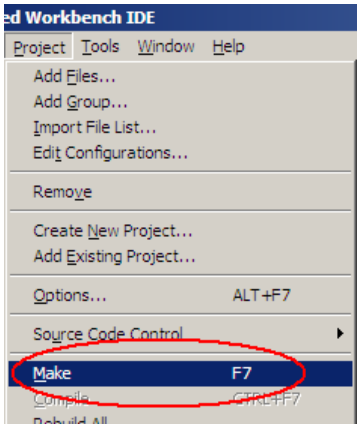
20. Now is a good time to save project and workspace again, as shown in Figure 2-20. Since you've already given a name to project, workspace and source files, IAR will not ask you for a name again.

Figure 2-20. The "File" Menu.



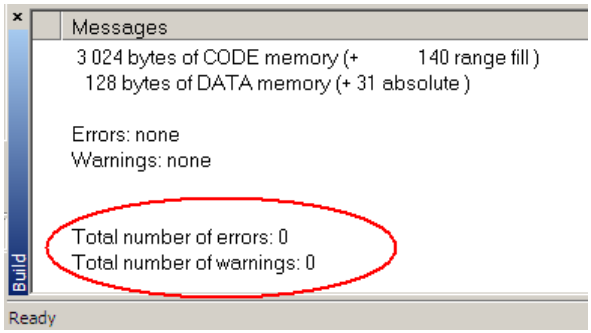
21. With all files ready, it's time to compile and build our application. Select the "Make" item in the "Project" menu as shown in Figure 2-21.

Figure 2-21. The "Project" Menu.



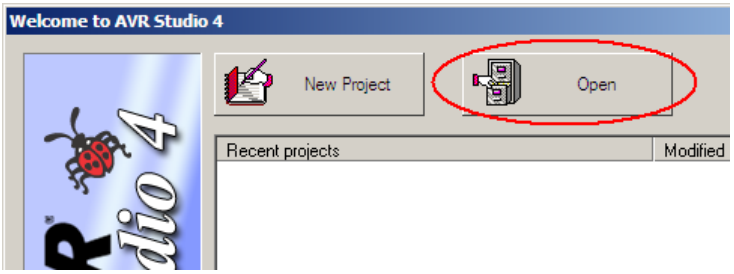
22. During the build process the Message window will show the progress. When the build is complete, the Message window should report no errors and no warnings, as shown in Figure 2-22. It not, locate the errors or warnings and resolve.

Figure 2-22. The "Message" Window.



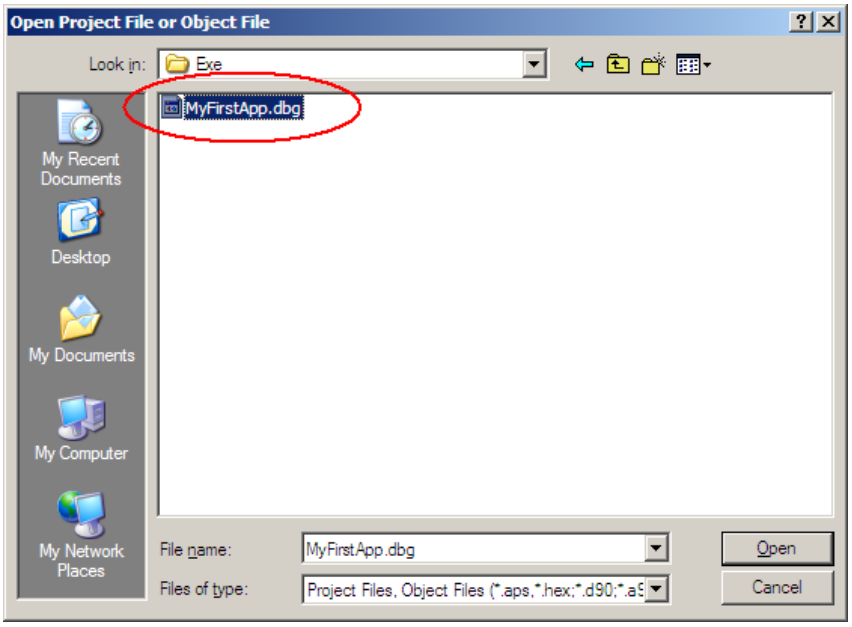
23. The demo application is not built en ready for running on the DB101. Open AVR Studio® and click “Open” in the “Welcome” dialog box, as shown in Figure 2-23.

Figure 2-23. “Welcome” Dialog Box.



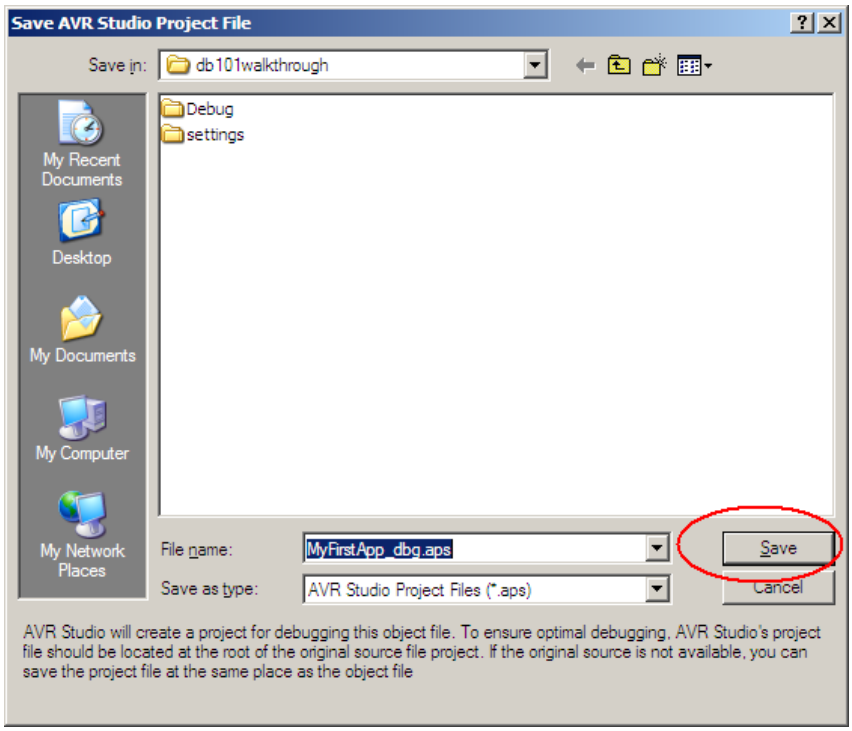
24. In the “Open Project File or Object File” dialog box, locate and select the debug file for your demo application. It will be located under the “Debug\Exe” folder in your project folder. The filename will be the same as your project file, as shown in Figure 2-24. Now click “Open”.

Figure 2-24. “Open Project File or Object File” Dialog Box.



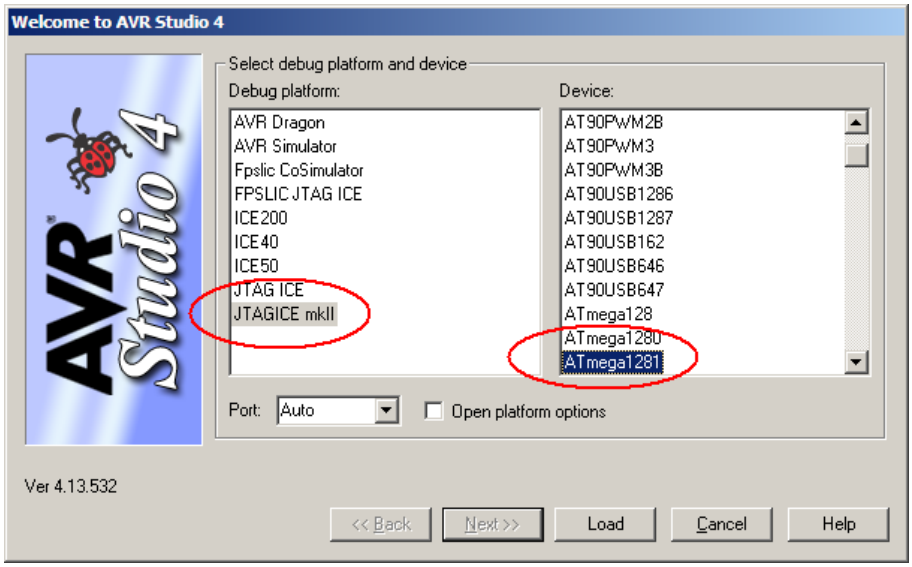
25. AVR Studio will ask you for a filename for the AVR Studio project. We recommend that you just accept the suggested file and location, as shown in Figure 2-25.

Figure 2-25. "Save AVR Studio Project File" Dialog Box.



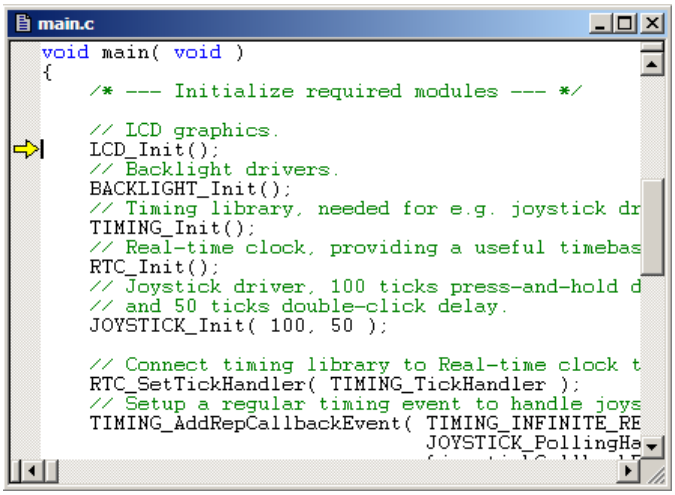
26. Now, you need to select which debug platform and device you are using. Select "JTAGICE mkII" and "ATmega1281" as shown in Figure 2-26. Make sure your JTAGICE mkII is connected to your computer and to the DB101 board, and the both are powered, then click "Load".

Figure 2-26. "Debug and Platform Selection" Dialog Box.



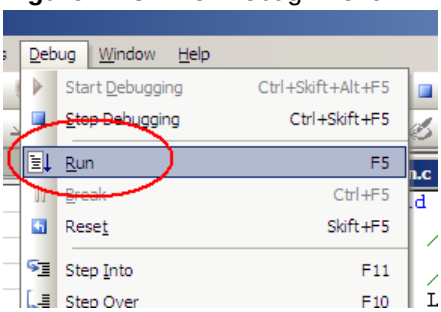
27. AVR Studio will now connect to your JTAGICE mkII and program your application into the ATmega1281 on the DB101 board. After a short while, the debug session starts, and AVR Studio indicates that code execution is stopped at the first line of the `main()` function, as shown in Figure 2-27.

Figure 2-27. Debug Session Waiting to Run



28. Now, everything you have to do is select the “Run” item in the “Debug” menu, and then use the joystick to change the line on the LCD.

Figure 2-28. The “Debug” Menu.



29. If you want to, you can break code execution, go back to IAR, change the code, go back to AVR Studio, accept reload, and then run the code again. Enjoy!

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

8101A-AVR-09/07