AVR483: DB101 Firmware - Getting Started

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.





8-bit **AVR**[®] Microcontrollers

Application Note

Rev. 8101A-AVR-09/07





2 Your First DB101 Application

 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.

Create New Proje	ct		×
<u>I</u> ool chain: Project templates:	AVR	•	
Empty project			
CLIB			•
Description:			
Creates an empty (project.		

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.

Save As		<u>?</u> ×
Sale in:	🔁 db 101walkthrough	
My Recent		
Documents Desktop		
My Documents		
My Computer		
San Naturada	File name: MyFirstApp Save as type: Project Files (* ewn) Can	/e
Places		

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.

Options for node "MyF	irstApp"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger CCR ICE200 JTAGICE JTAGICE JTAGICE mkII Simulator Third-Party Driver	Target Output Library Configuration Library Options Heap Configuration Processes configuration
	OK Cancel

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.

Options for node "MyFin Categoru: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger CCR ICE200 JTAGICE JTAGICE JTAGICE mkII Simulator Third-Party Driver	Ibrary Configuration Library Options Heap Configuration System Data stack (CSTACK) Betum address steck (RSTACK) Size (bytes 0x100 Size (levels): 32 Place in external memory Place in external memory Place in external memory External Memory Configuration Flace in external memory bus Add one wait-state to external memory accesses Base address 0x0 0x0 0x0 Memory size 0x0 0x0 0x0 Initiatec unused integrut vectors with RETI instructions Initiatec unused integrut vectors with RETI instructions	
ľ	Initialize unused interrupt vectors with RETI instructions Enable bit definitions in 1/0 include files OK Cancel	





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.

Options for node "MyFi	rstApp"
Category: General Options G/C++ Compiler Assembler Custom Build Build Actions Linker Debugger CCR ICCE200 JTAGICE JTAGICE mkII Simulator Third-Party Driver	Factory Settings Language Cole Optimizations Output List Preprocessor D Size Size Medium Common subexpression elimination Function inlining Code motion Cross cal Number of cross-call passes: Unlimited Aways do cross call optimization
	OK Cancel

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.

Options for node "MyFi	rstApp"
Options for node "MyFi Category: Penferal Options C/C++ Compiler Accembler Custom Build Build Actions Linker Debugger CCR ICE200 JTAGICE JTAGICE mkII Simulator Third-Party Driver	rstApp" Factory Settings Language Code Optimizations Output Ignore standard include directories: STOOLKIT_DIRS\INC\STOOLKIT_DIRS\INC\CLIB\ Additional include directories: (one per line) SPROJ_DIRS SPROJ_DIRS Image: Code of the symbols: (one per line) Preinclude file: Image: Code of the symbols: (one per line) Preinclude file: Image: Code of the symbols: (one per line) Image: Code of the symbols: (one per line) Image: Code of the symbols: (one per line) Image: Code of the symbols: (one per line) Image: Code of the symbols: (one per line)
	OK Cancel





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.

Category: General Options C/C++ Compiler Assembler	Output Etra Output #define Diagnostics List Config Proce:
Custom Build Build Actions Linker Debogger CCR ICE200 JTAGICE JTAGICE JTAGICE mKII Simulator Third-Party Driver	Output file Oyemide default MyFirstApp.dbg (None for the selected format) Eomat Debug information for C-SPY With runtime control modules With runtime control modules With runtime control modules With runtime control modules Buffered terminal output Fullow Q-SPY-specific extra output file Other Output for fat: Ubrof 8 (forced) Format variant: None
	Module-local symbols: Include all

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.



AVR483

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.

Save Workspace	e As						? ×
Save <u>i</u> n:	b 101wal	kthrough		•	🗕 🖻 🚔	· 🏢 -	
My Recent Documents	🛅 settings						
My Documents							
My Computer							
My Network Places	File <u>n</u> ame: Save as <u>type</u> :	MyFirstApp Workspace	Files (*.eww)		ľ		<u>S</u> ave Cancel

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.

U ,
d Workbench IDE
Project Tools Window Help
Add Files
Add Group
Import File List
Edi <u>t</u> Configurations
Remo <u>v</u> e
Create New Project

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.



Add Files - MyFirst	tApp	<u>?</u> ×
Look <u>i</u> n:	🔁 db 101 walkthrough 📃 🖛 🗈 💣 🎟 🗸	
My Recent Documents Desktop My Documents My Computer	interfings gfx_lib.h baddight_driver.ct joystick_driver.h ffo_lib.c lcd_lib.h gfx_lib.c power_driver.h joystick_driver.ct intc_driver.h lcd_lib.c intc_driver.h lcd_lib.c intc_driver.h power_driver.ct is6b1713_driver.h rtc_driver.ct istdint.h s6b1713_driver.ct itming_lib.h termfont_lib.ct itming_lib.h termfont_lib.ct fming_lib.h fming_lib.ct itming_lib.h ffo_lib.h ffo_lib.h	
My Network Places	File name: "timing_lib.c" "backlight_driver.c" "fifo_lib.c" "g Ope Files of type: Source Files (".c;".cpp;".cc;".h;".hpp;".s";".msa; Cancel	en

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.



¹⁰ AVR483

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.

Save As					<u>?</u> ×
Save <u>i</u> n:	🗀 db 101 walkthrough		•	+ 🗈 💣 🎟+	
My Recent Documents Desktop My Documents My Computer My Network	Debug settings backlight_driver.c backlight_driver.h common.h fifo_lib.c fifo_lib.h gfx_lib.c gfx_lib.h joystick_driver.c lcd_lib.c lcd_lib.h power_driver.c power_driver.h File name: main	 rtc_driver.c rtc_driver.h s6b1713_driver.c s6b1713_driver.h sound_driver.c sound_driver.h stdint.h termfont_lib.c timing_lib.c timing_lib.h 			Save
	Save as type: IDE	Files (".c;".cpp;".cc;".h;".	s*;*.msa	a;*.asm;*.lst 💌	Cancel

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.





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.

Жг	AR Em	ibedd	ed Work	bench I	ID
Eile	<u>E</u> dit	<u>V</u> iew	Project	<u>T</u> ools	V
N	ew			•	
<u>0</u>	pen			•	
9	ose				ł
Si	a <u>v</u> e Wo	orkspac	te		ł
C	los <u>e</u> W	orkspa	ce		ł
S	ave		CT	RL+S	
5	ere <u>e</u> s				
Si	ave Aļ)	
P	ane Se	tup			
D	din E		ст		

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.

ed Workbench IDE						
Project	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp			
Add E	iles					
Add G	<u>i</u> roup					
<u>I</u> mpor	rt File Lis	st				
Edi <u>t</u> C	Edi <u>t</u> Configurations					
Remo <u>v</u> e						
Create <u>N</u> ew Project						
Add Existing Project						
Options ALT+F7						
So <u>u</u> rce Code Control						
Make			F7			
Comp			CTRL+F7			

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.

Welcome to AVR Studio	4		
R dio 4	New Project Recent projects	Open	Modified

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.

Save AVR Studio	Project File				<u>? ×</u>
Save in:	b 101walkthr	ough	•	- 🗈 💣 🎟	-
My Recent Documents Desktop My Documents	Debug				
My Computer My Network Places	File <u>n</u> ame: Save as <u>type</u> :	MyFirstApp_dbg.aps AVR Studio Project Files (*.a	ps) ure optimal		Save Cancel
file should be loca save the project fi	ted at the root of the e at the same place	original source file project. If the as the object file	ne original	source is not ava	ilable, you can

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 mkll 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

🖹 main	
voi	d main(void)
1	/* Initialize required modules */
4	<pre>// LCD graphics. LCD_Init(); // Backlight drivers. BACKLIGHT_Init(); // Timing library, needed for e.g. joystick dr TIMING_Init(); // Real-time clock, providing a useful timebas RTC_Init(); // Joystick driver, 100 ticks press-and-hold d // and 50 ticks double-click delay. JOYSTICK_Init(100, 50);</pre>
	<pre>// Connect timing library to Real-time clock t RTC_SetTickHandler(TIMING_TickHandler); // Setup a regular timing event to handle joys TIMING_AddRepCallbackEvent(TIMING_INFINITE_RE JOYSTICK_PollingHa </pre>

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	Figure	2-28.	The	"Debug"	Menu.
-----------------------------	--------	-------	-----	---------	-------

<u>D</u> ebug <u>W</u> indow <u>H</u> elp	
Start Debugging	Ctrl+Skift+Alt+F5
Stop Debugging	Ctrl+Skift+F5
	F5
17 Break	Ctrl+F5
🔄 Rese <u>t</u>	Skift+F5
Step Into	F11
Step Over	F10 I

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

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2007 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR® AVR Studio® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.