Adapting Code::Blocks IDE for ARM Development

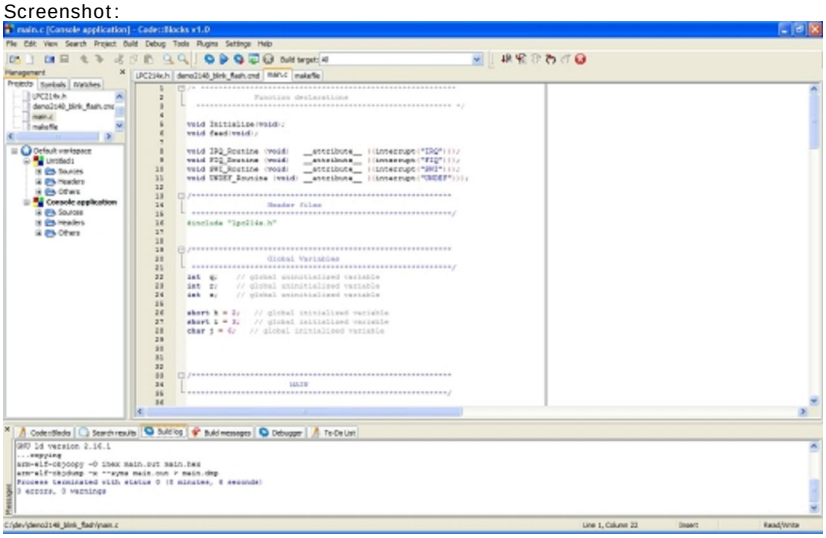
Posted By vsergeev On January 8, 2006 @ 1:57 am In Uncategorized | No Comments

This brief guide can be followed after the ARM-GCC toolchain is setup. For instructions to setup the ARM-GCC toolchain, refer to the ¡Setting up the ARM-GCC Toolchain on Windows¡ page of this site.

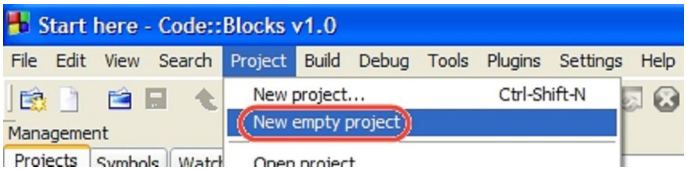
I recently found perhaps the only other suitable IDE for Windows named Code::Blocks [1]. Unlike Eclipse, it¡s slim, fast and native. Like Eclipse, it¡s highly configurable and can be used with GCC. In addition, Code::Blocks can be easily used with GDB to debug your code (major factor in choosing another IDE). This is good to use later with OCDRemote/gcc to debug your ARM code (although I haven¡t configured it, yet.)

So here are the instructions for setting up the Code::Blocks to work with the ARM GCC toolchain. To do this, all that needs to be done is configuring Code::Blocks to use an external Makefile, but finding the location of these settings took a rather long time.

Screenshot:

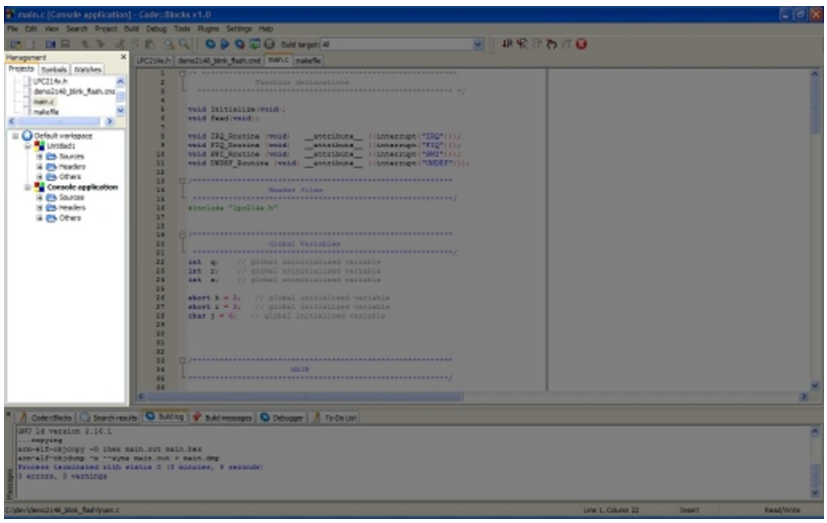
 [2]

Step One ¨ Create a New Project
Since none of the included project templates really fit for ARM GCC development, it¡s easiest to start out by making a new blank project. So do this by clicking on the ¡Project¡ menu, then ¡New empty project¡. Save the project in the folder of your project, where you plan to hold the code.



You will see several window panes in the IDE. I will briefly describe them here:

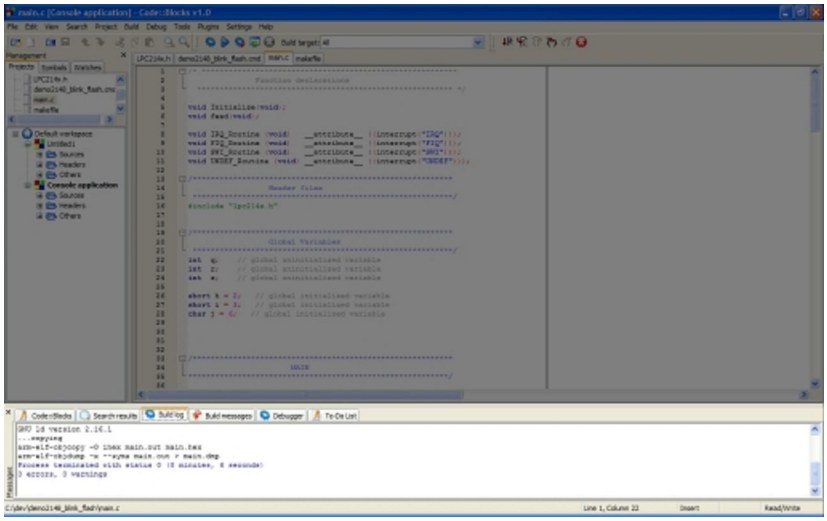
- Management Pane

 [3]

The Management Pane contains three tabs that are all somehow related to the code itself.
First is the ¡Projects¡ tab, which lists the opened projects in the workspace, and the files that are associated with the project. The files are automatically separated into three sections: Sources, Headers, and Others. Double-clicking on a file will bring it up for editing in the editor.
Second is the ¡Symbols¡ tab, which outlines all of the functions, variables, enumerations, preprocessor directives, and classes existing in the code of the project. Double-clicking on any one of these elements will bring it up highlighted in the editor.
The third and final tab is the ¡Watches¡ tab, which is used to add debugging watches to variables.

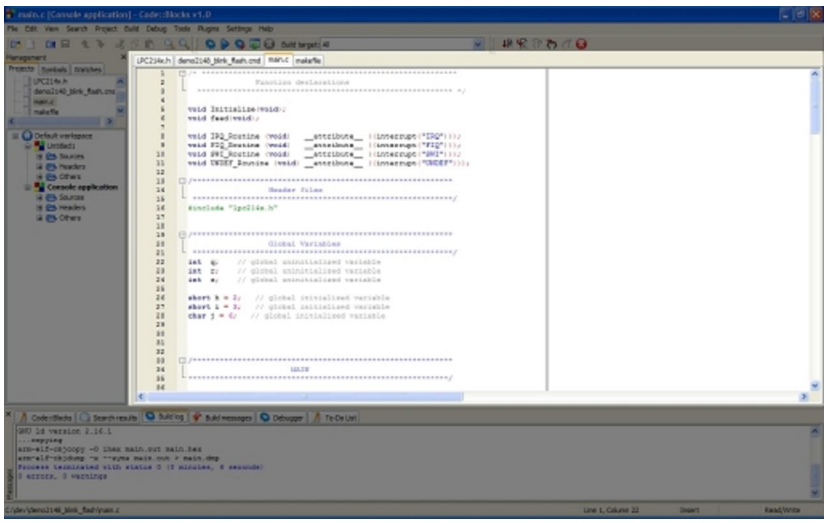
- Messages Pane

- Messages Pane



[4]
The Messages Pane stores the general output from several sources. It has tabs to easily switch between these message categories.
The ¡Code::Blocks¡ messages tab obviously holds the message output from Code::Blocks itself, which is usually irrelevant to development. If a problem occurs with the IDE, this would be a good place to check for troubleshooting. The next tab is the ¡Search Results¡ tab, which is pretty self-explanatory. Clicking on a search result will open the file and highlight the line containing the search term(s) in the editor. The next tab is the ¡Build log¡, and holds the commands made to the compiler, linker and other tools. Next is the ¡Build messages¡ tab, which lists the errors or warnings present in the code that was caught by the compiler, including the file and line number at which the error/warning is located. The next tab is the ¡Debugger¡ tab, which will contain the output of the chosen debugger. The final tab is the ¡To-Do List¡ tab, which lists To-Do items that are present in the code. To-Do items are defined by Code::Blocks as a comment in code, and can be added by right-clicking on a line in the editor and clicking ¡Add To-Do Item¡¡ .

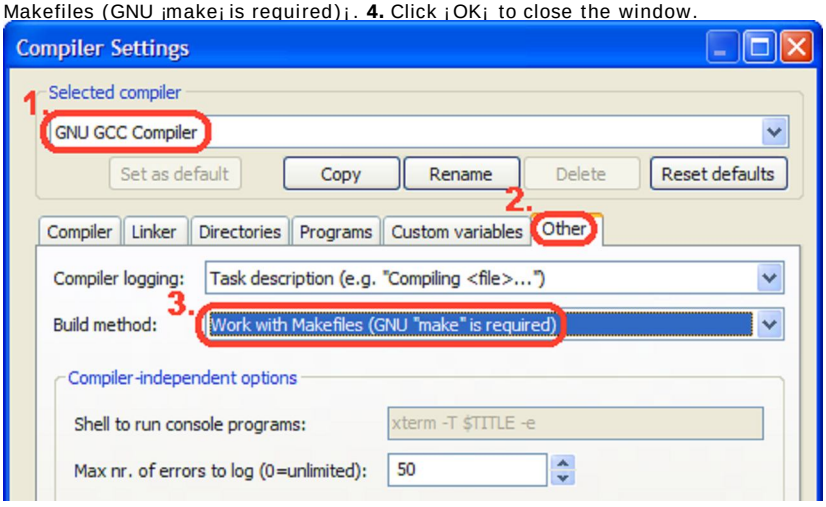
- Editor Window



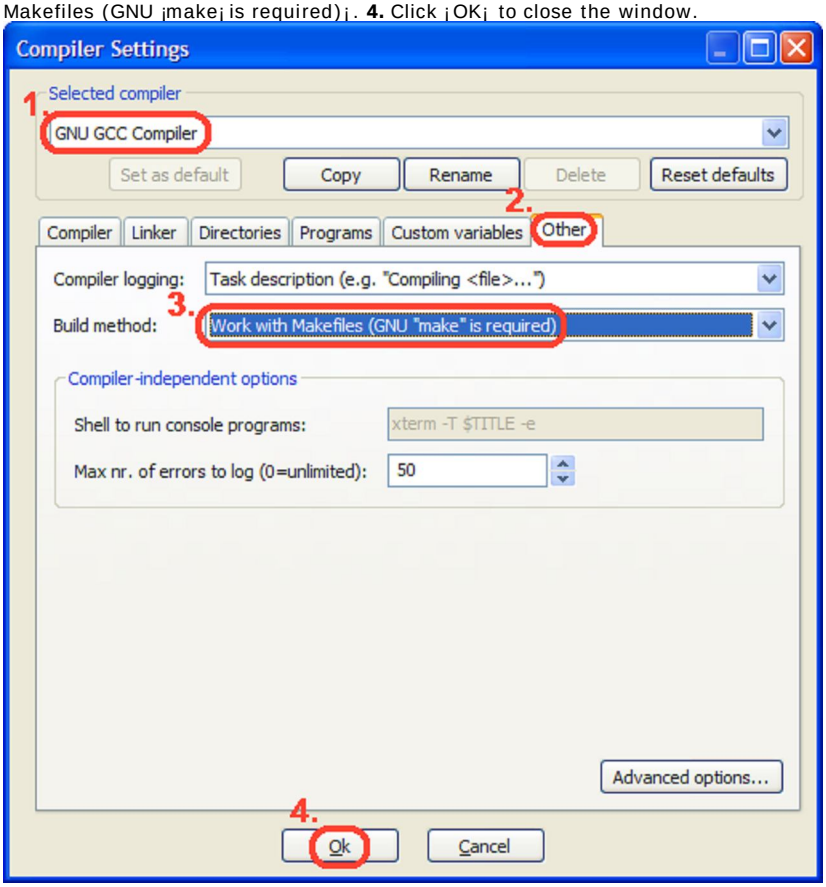
[5]
Probably the most important part of the IDE, the text editor itself. It supports syntax highlighting, several forms of autocompletion, collapsing code segments, and much more. It can be highly customized under the ¡Settings¡ menu, in the ¡Editor¡ settings. Notaby, the font, indentations, colors, and auto-completion can be configured. In addition, ¡Default Code¡ can be defined to be written whenever a new source/header file is created.

Step Two ¨ Configure Build Settings
The next step is to setup Code::Blocks to use an external Makefile instead of automatically and directly calling GCC.

Click on the ¡Settings¡ menu, then ¡Compiler¡.
1. Keep your Selected Compiler as the ¡GNU GCC Compiler¡. **2.** Click on the ¡Other¡ tab. **3.** Change ¡Build Method¡ to ¡Work with Makefiles (GNU ¡make¡ is required)¡. **4.** Click ¡OK¡ to close the window.



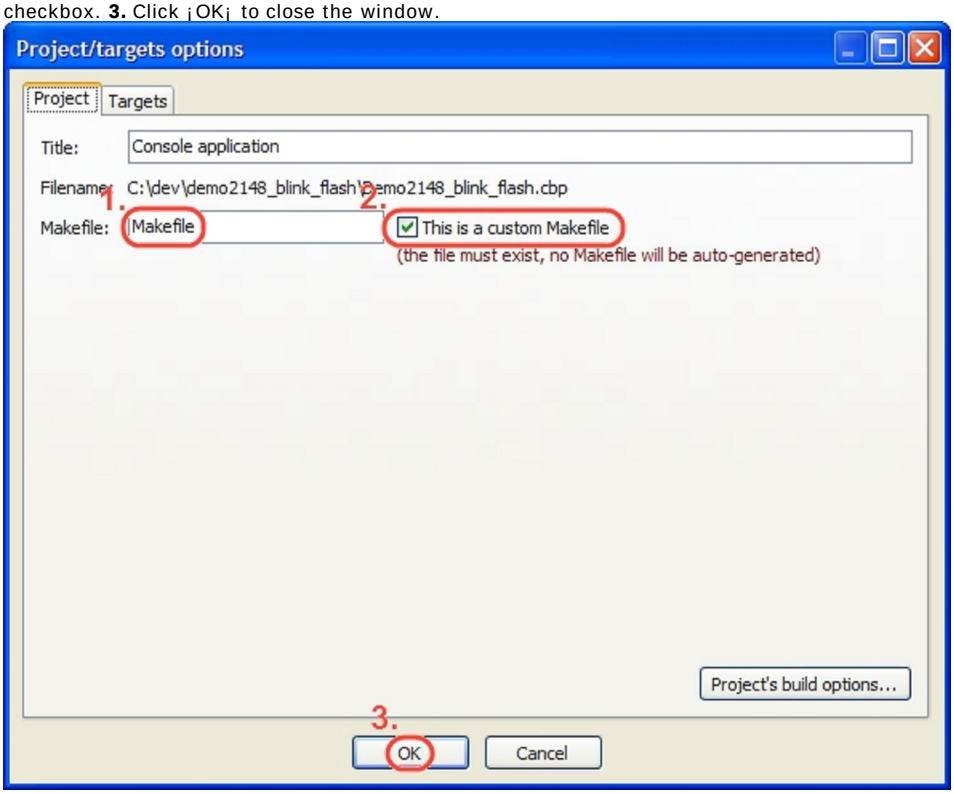
Makefiles (GNU ¡make¡ is required)¡. **4.** Click ¡OK¡ to close the window.



[6]

Next the path of the Makefile must be defined. Click on the ¡Project¡ menu, then ¡Properties¡.

1. Type the name of the external Makefile (given it is in the same directory as where the project is saved, if not, modify the path revelant to the project file or provide an absolute path) in the ¡Makefile:¡ textbox. **2.** Check the ¡This is a Custom Makefile¡ checkbox. **3.** Click ¡OK¡ to close the window.



[7]

Here is a list of several features Code::Blocks features which you can further configure if you please:

- Intellisense-like Completion
- Keyword Auto-Completion
- User-Defined External Tools
- Project To-Do list
- Built-In Compiler Flag Configuration (although obviously irrelevant if using your own Makefile)
- Very Adaptable Text Editor
- Importing of Dev-C++, MS Visual Studio, or MS Visual C++ projects/workspaces/solutions.
- Additional Code::Blocks Plugins

Code::Blocks Messages Pane Thumb

Code::Blocks Messages Pane Thumb

URLs in this post:

[1] Code::Blocks: **http://www.codeblocks.org/**
[2] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-PlainWindow.jpg**
[3] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-ManagementPane.jpg**
[4] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-MessagesPane.jpg**
[5] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-EditorWindow.jpg**
[6] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-CompilerSettings.jpg**
[7] Image: **http://www.frozeneskimo.com/electronics/wp-content/uploads/2006/01/CodeBlocks-ProjectOptions.jpg**