AN10862

LPC1000 software development toolchain

Rev. 03 — 25 January 2010 Application note

Document information

Info	Content
Keywords	LPC1000, GNU-GCC
Abstract	How to use the GNU toolchain with the Cortex LPC1000 series

NXP founded by Philips

Revision history

Rev	Date	Description
03	20100125	Include “SWD” in debug config and IAR test examples
02	20091214	Include Keil test examples
01	20090827	Initial version

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

© NXP B.V. 2010. All rights reserved.

1. Introduction

LPC1000 is the new chip series that is based on the ARM Cortex processor family with many advanced features.

This document will show how to use the toolchain in LPC1000 series such as: LPC17xx (Cortex-M3), LPC13xx (Cortex-M3) and LPC11xx (Cortex-M0) families.

In this document, we will illustrate the setup for microcontroller LPC1768 on Keil MCB1700 evaluation board.

2. Software update requirement

Here is a list of setup files should be installed:

- Java Runtime Environment (required by IDE – Eclipse)
- CodeSourcery Toolchain
- C/C++ Eclipse IDE
- Zylin CDT plugin for Eclipse.
- Segger J-link GDB server and driver.
- Flash magic.

2.1 Java Runtime Enviroment

First, we must check for Java Runtime Environment on system since Eclipse IDE is written partially in JAVA.

- Go to the command prompt, type command “Java –version” and look at the result.

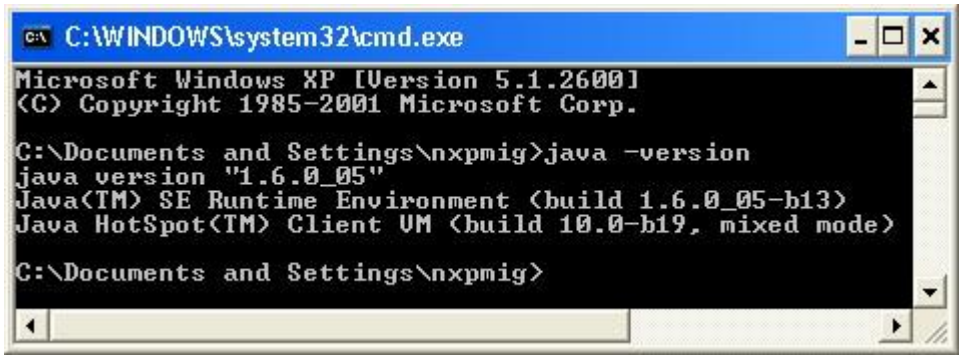


Fig 1. JAVA version check

- If windows can not recognize this command or the version is not equal to or higher than 1.6.0_01, it means that you need to also download and install JRE in the next step.

Downloading Java Runtime Environment

- Go to the website: http://www.nxp.com/redirect/java.sun.com
- Look for “Download” item, follow these links and find the setup file “Java Run Time Environment” (JRE) (newest version) like this:

© NXP B.V. 2010. All rights reserved.

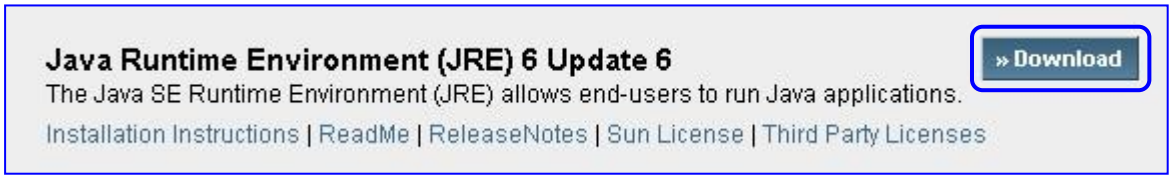


Fig 2. Download JAVA Runtime Environment

The version found on the website may be slightly different than the example given in this application note.

2.2 CodeSourcery toolchain

Go to http://www.nxp.com/redirect/codesourcery.com/release830 and download the latest package release for Windows platform, and then install this package in your system after removing the previous version of CodeSourcery Lite.

Please note that the GNU version in “makeconfig” file must be changed to correspond to this release version.

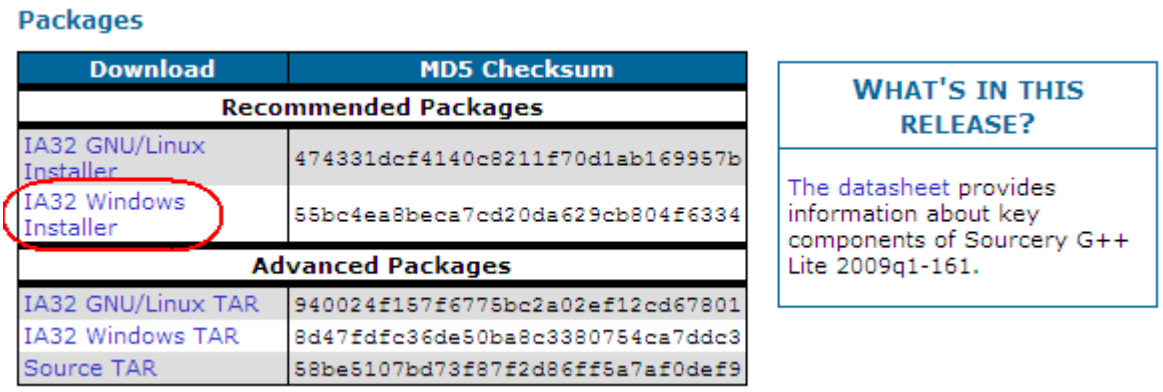


Fig 3. Update CodeSourcery toolchain

2.3 Downloading Eclipse IDE and plug-in

2.3.1 Download Eclipse IDE for C/C++

Go to this website to download the Integrated Development Environment – IDE of Eclipse:

http://www.nxp.com/redirect/eclipse.org/downloads

Fig 4. Choose “Eclipse IDE for C/C++ Developers” item to download

Install Eclipse IDE after downloading and set all options to default during installation.

© NXP B.V. 2010. All rights reserved.

2.3.2 Install Zylin CDT plug-in

Go through Help → Software update…

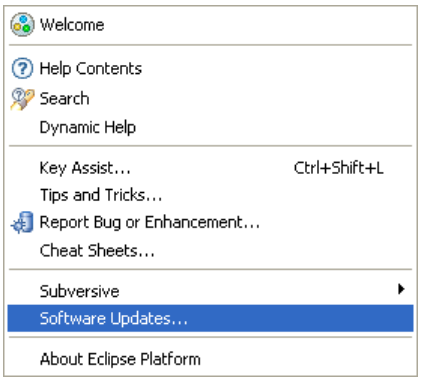


Fig 5. Choose Software Update in Eclipse

If the site http://download.eclipse.org/tools/cdt/releases/ganymede exists in Available Software, choose all items inside there, and then click Install… Follow remaining steps to complete the update.

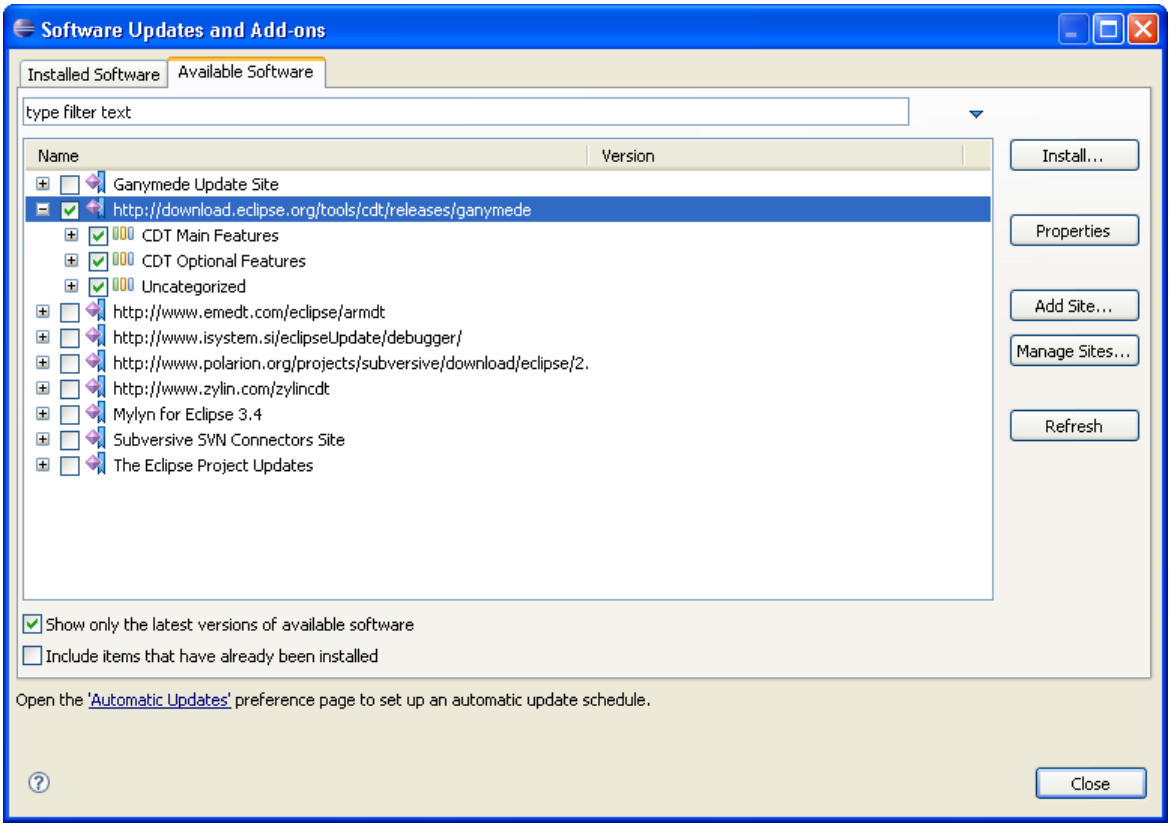


Fig 6. Check all items inside http://download.eclipse.org/tools/cdt/releases/ganymede sub-item

© NXP B.V. 2010. All rights reserved.

If this site address does not exist, click Add Site and fill this address in to dialog http://www.nxp.com/redirect/download.eclipse.org/ganymede

For the Zylin Embedded CDT plug-in, because its repository has been removed and changed to http://opensource.zylin.com/zylincdt, we need to re-target this link in Eclipse software update.

Click the Add Site button as shown in Fig 6, fill in the location field with the new site address, and follow the remaining steps to complete the update.

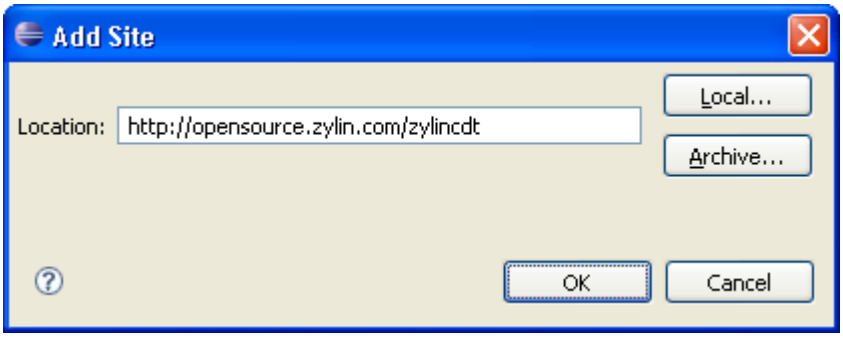


Fig 7. Update Zylincdt with new repository

2.4 Segger J-link

Go through http://www.nxp.com/redirect/segger.com/download_jlink.html to download the latest J-Link software.

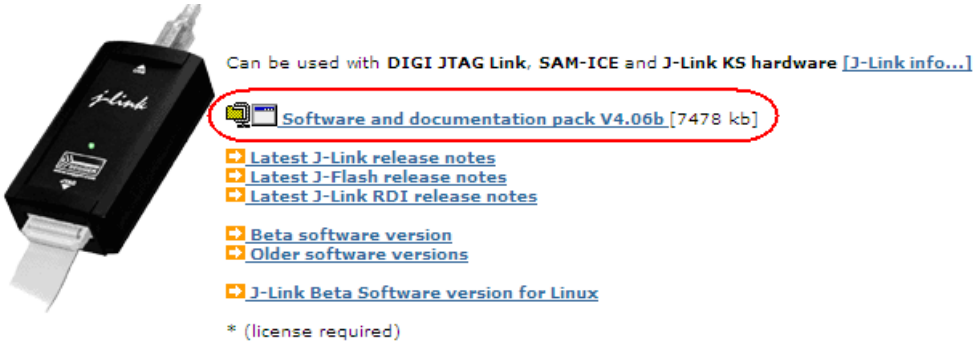


Fig 8. Update J-link software from SEGGER

2.5 Flash Magic

Flash Magic has supported flash downloading for LPC17xx series.

http://www.nxp.com/redirect/flashmagictool.com/

3. Running Eclipse for the first time

3.1 Eclipse workspace setup

- Initialize Eclipse on the desktop:

© NXP B.V. 2010. All rights reserved.

Fig 9. Initialize eclipse icon on the desktop

- Set workplace for the first time here. In this case, a new workspace named “nxpdrv” (also the name of the new folder) will be created in the root of the C: drive; all the projects will be located in “C:\nxpdrv\”.

© NXP B.V. 2010. All rights reserved.

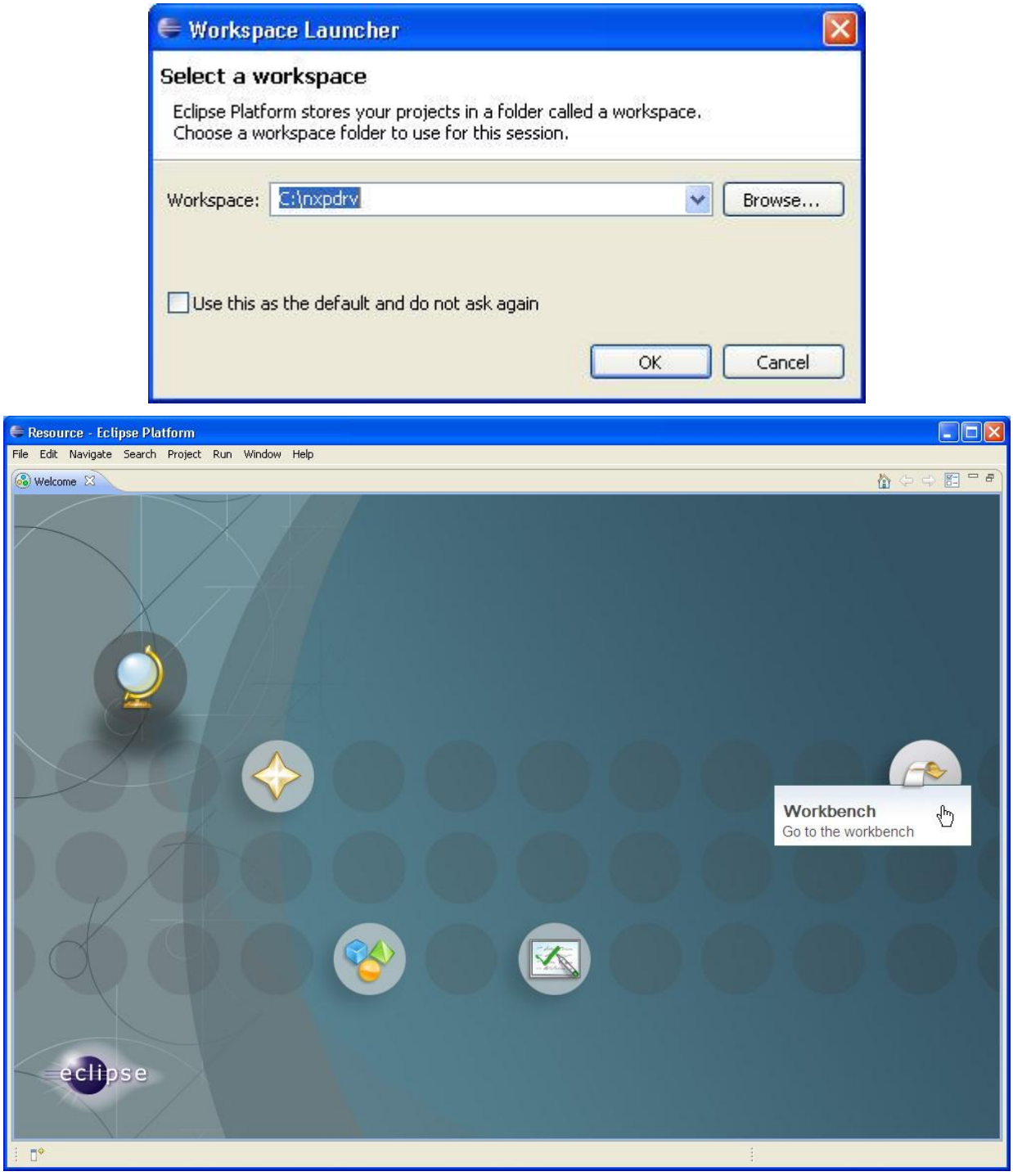


Fig 10. Set workplace and Eclipse for the first time

- Open perspective: Window → Open Perspective → Other ...

© NXP B.V. 2010. All rights reserved.

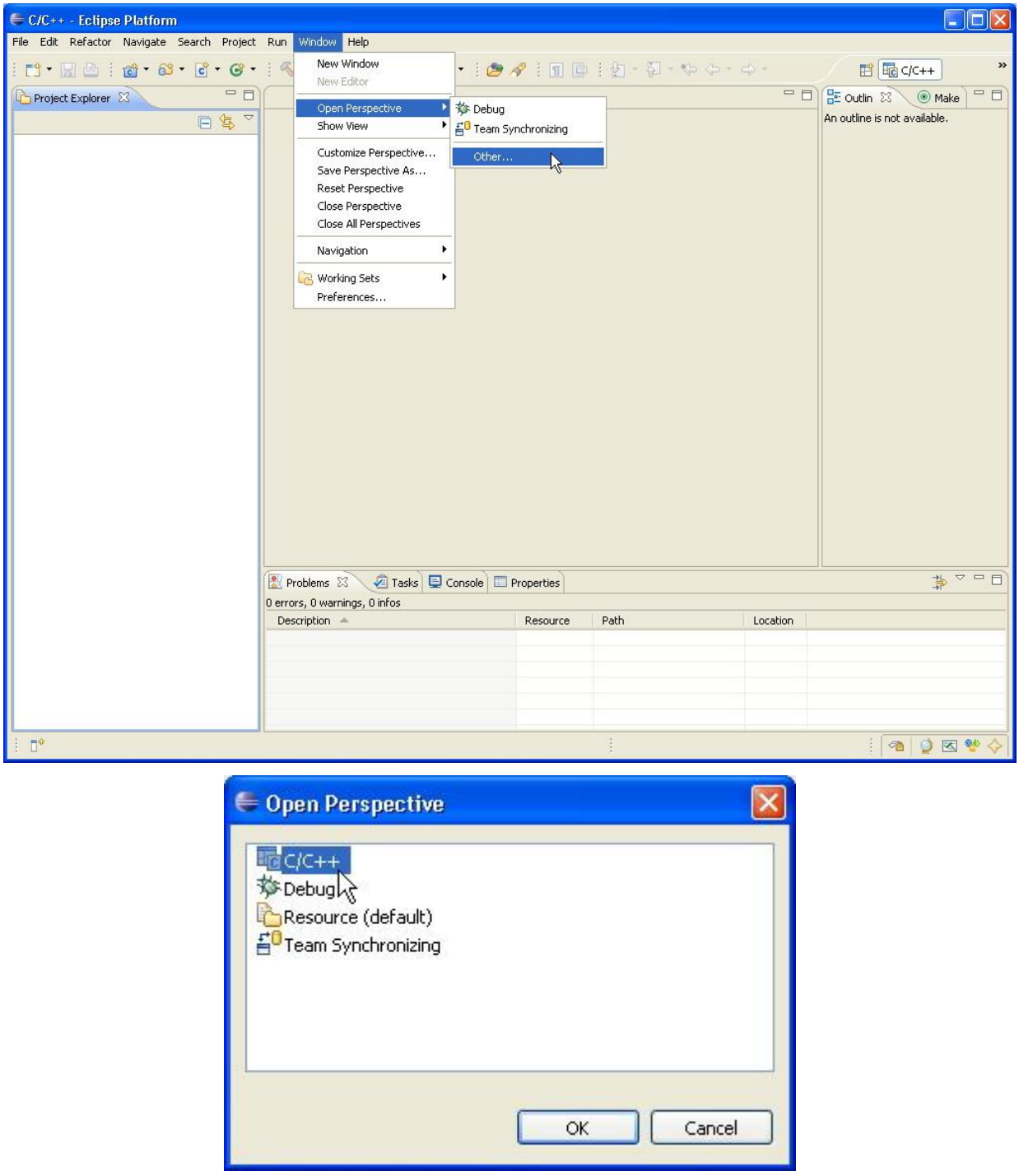


Fig 11. Open perspective for C/C++

3.2 Setting up External tools for Eclipse

We could add some tools to Eclipse for convenience. All tools we can add here are:

- Flash magic.

© NXP B.V. 2010. All rights reserved.

- GDB Server
- Serial terminal (existing on Windows)

Flash magic tool

Run → External Tools → Open External Tools Dialog…

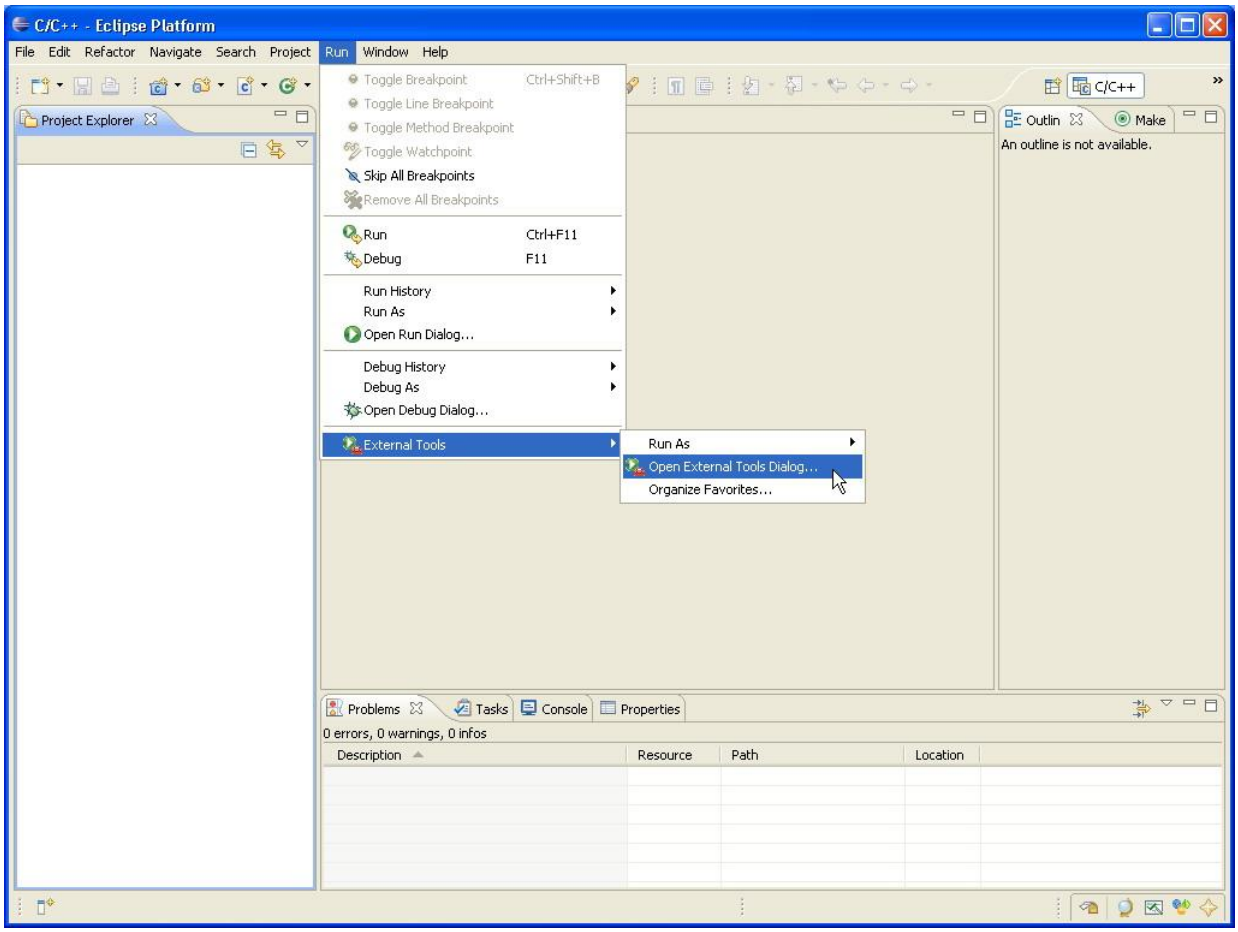


Fig 12. Steps to setting Flash magic as external tool (Step 1)

© NXP B.V. 2010. All rights reserved.

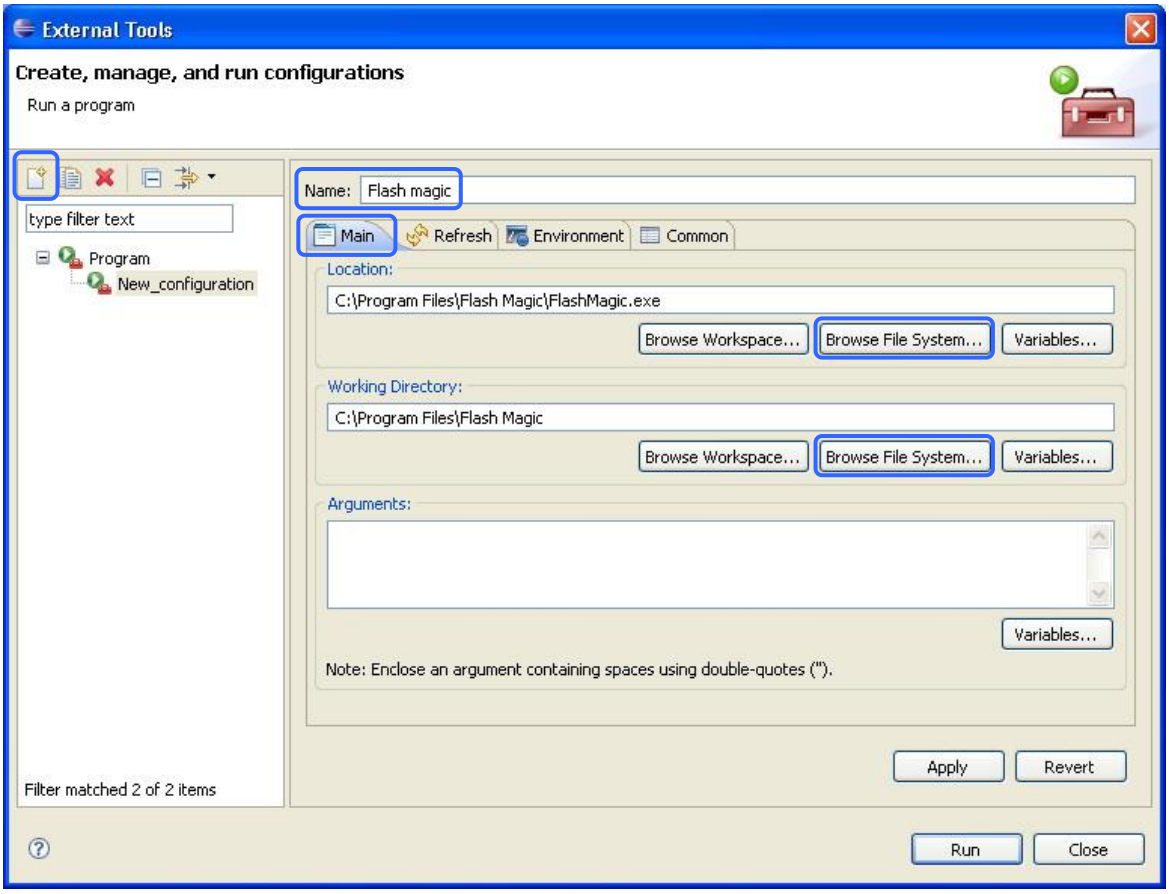


Fig 13. Steps to setting Flash magic as external tool (Step 2)

© NXP B.V. 2010. All rights reserved.

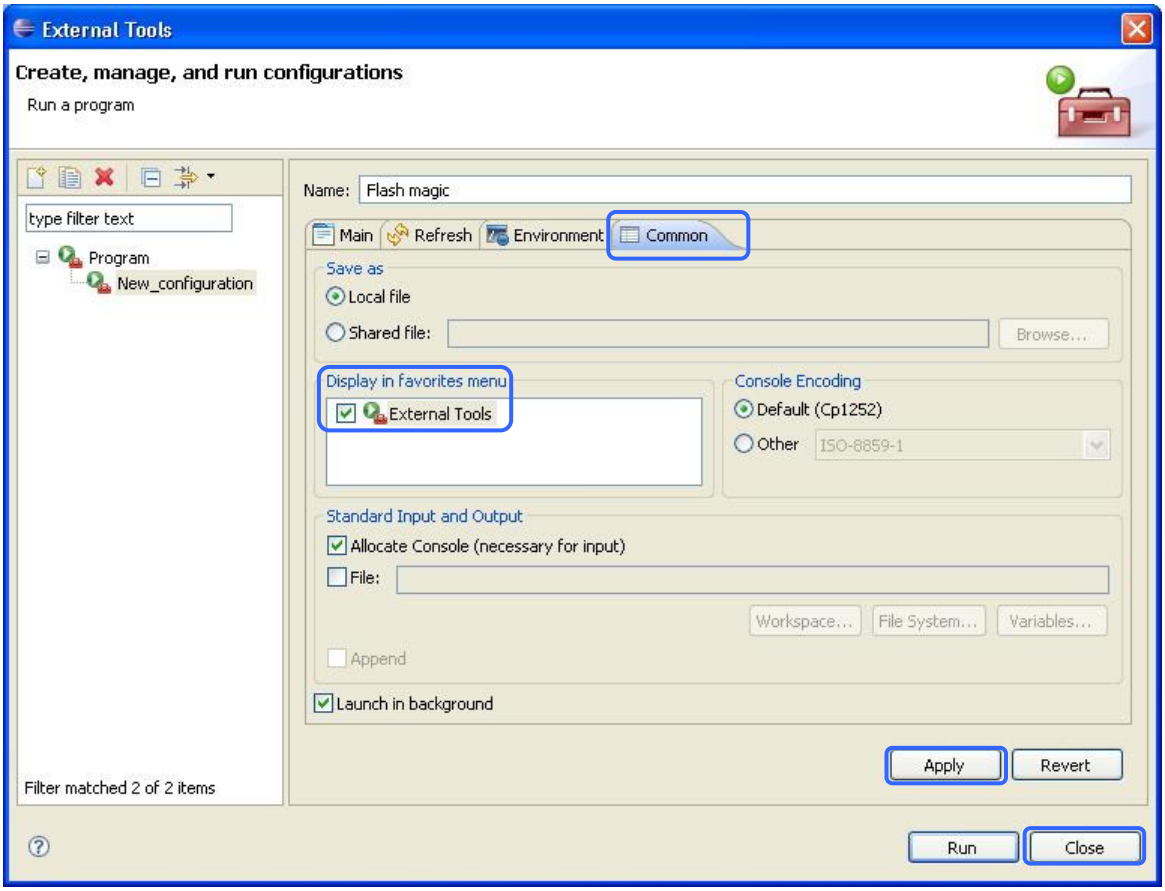


Fig 14. Steps to setting Flash magic as external tool (Step 3)

© NXP B.V. 2010. All rights reserved.

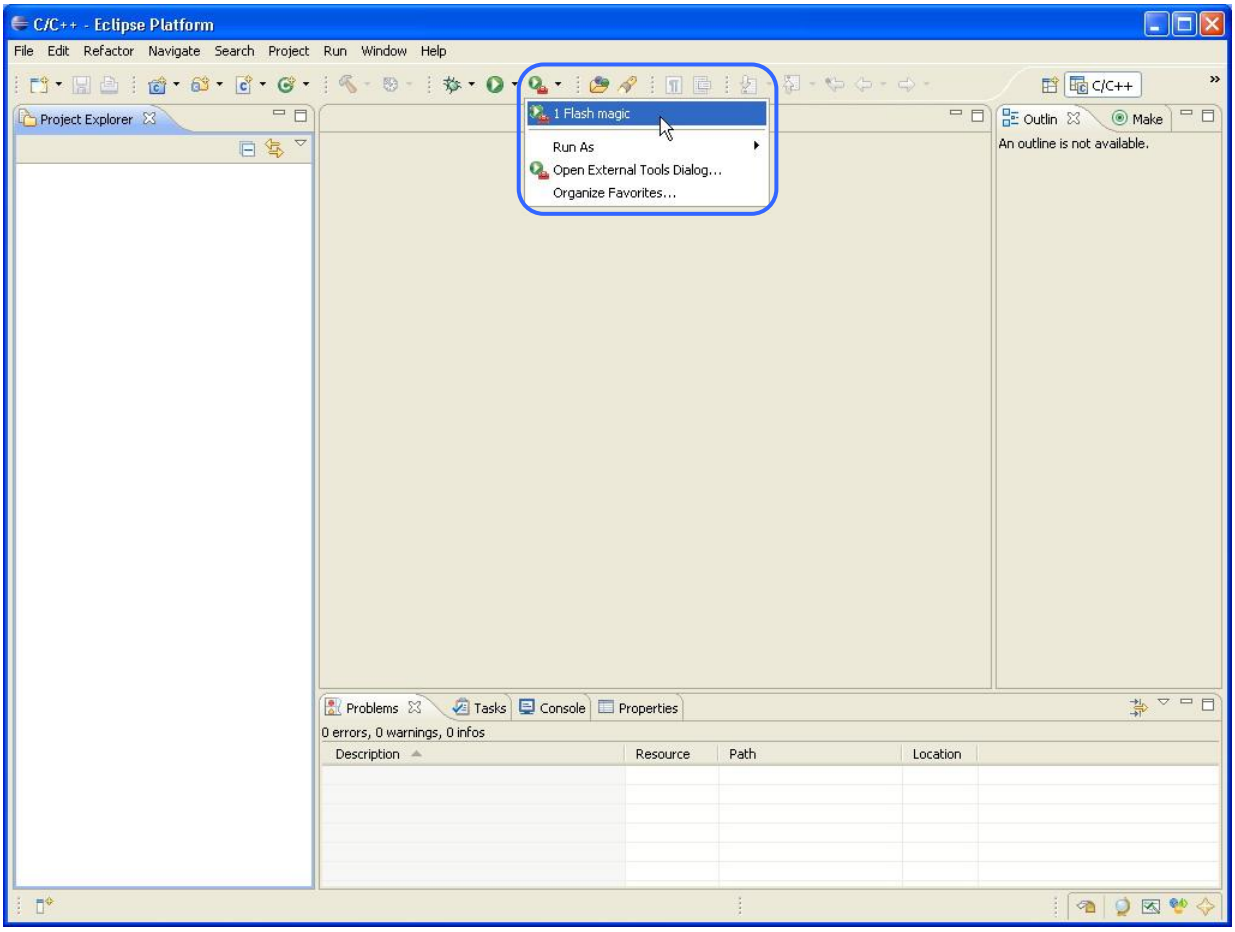


Fig 15. The result after setting

Follow these steps above to add other tools

- GDB server is located at directory that Jlink was installed, such as: “C:\Program Files\SEGGER\JLinkARM_Vxxx\JLinkGDBServer.exe”

  where Vxxx is the Jlink version that you’re using.

  Ex: JLinkARM_V408, JLinkARM_V409j…

© NXP B.V. 2010. All rights reserved.

Note: If you want to use SWD debug mode (for LPC13xx and LPC11xx devices only support SWD), include "-if SWD" in Argument tab in External Tools Configurations of gdb program.

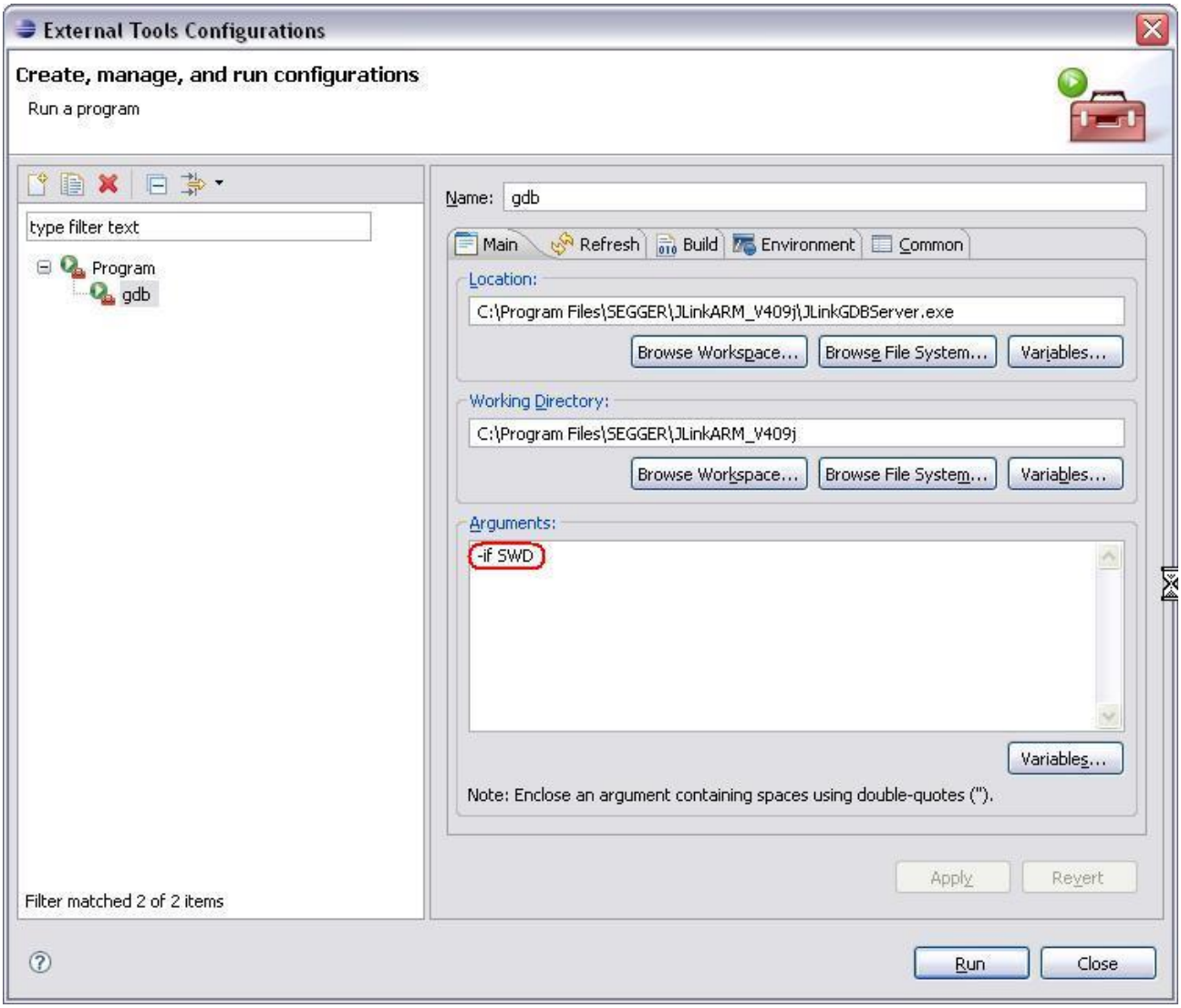


Fig 16. Configuration for cater SWD option

© NXP B.V. 2010. All rights reserved.

- Serial terminal is located at C:\Program Files\Windows NT".

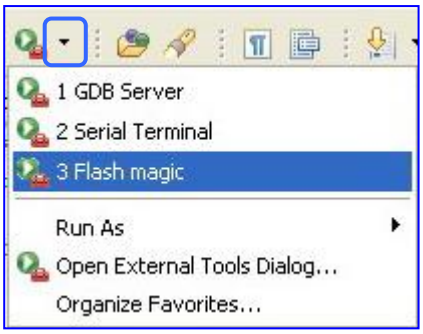


Fig 17. The result after all external tools setting

4. Creating and working with LPC1000CMSIS project

These following steps demonstrate how to build examples in project with CodeSourcery Lite/GNU toolchain in Eclipse IDE:

The project we used to represent here is LPC1700CMSIS. Other LPC1000CMSIS project are correlative.

4.1 Create new workspace

In this case, a "nxpdrv" workspace is created on C:\ drive.

4.2 Create new project

In this case, a new project is created with the name "LPC1700CMSIS".

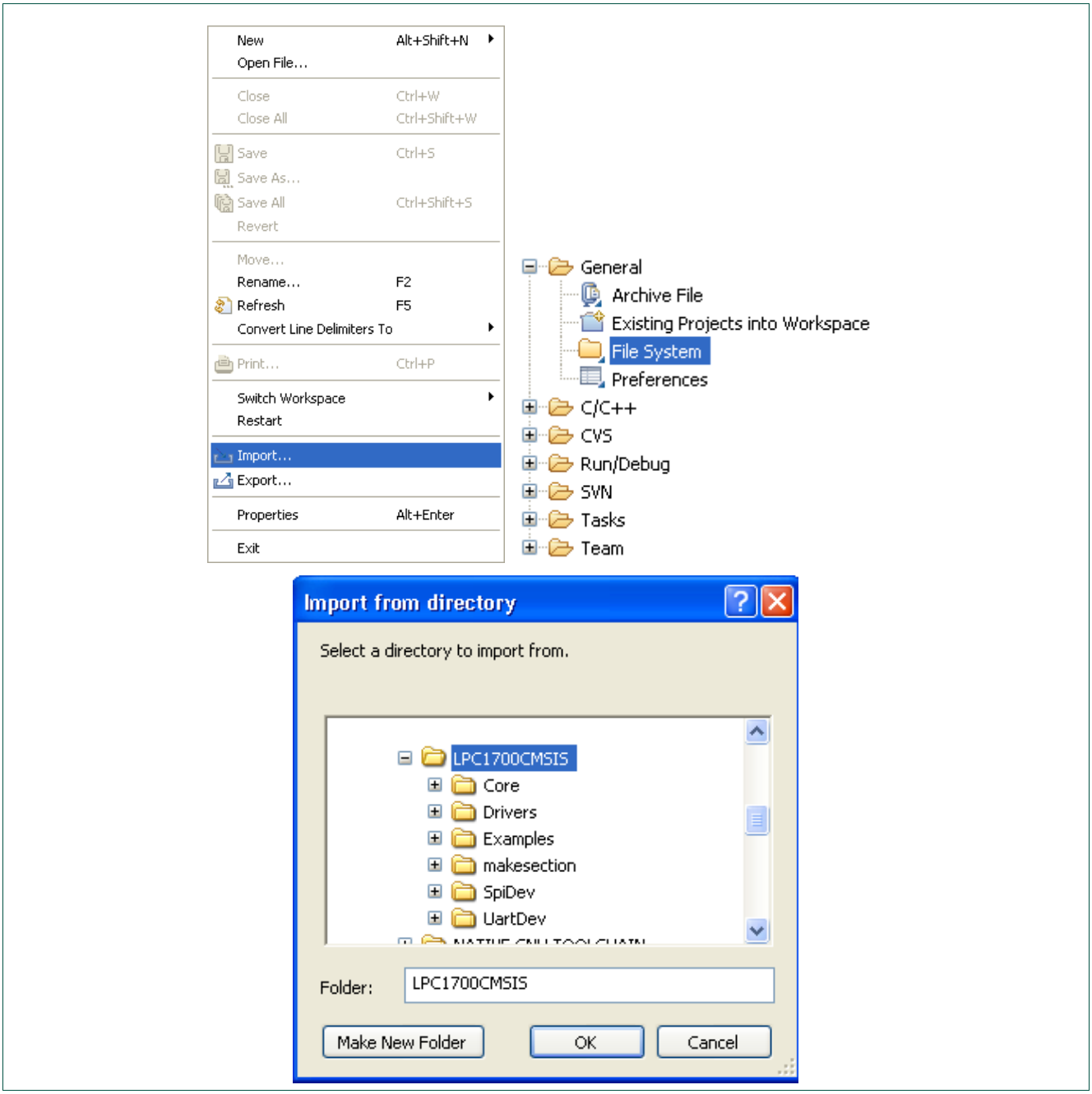
4.3 Extract the package

In this case, the "zip" package is extracted on D:\ driver, a folder named "LPC1700CMSIS" will be generated on D:\ drive.

4.4 Import package resource file into project

Import all the source files in project package (File → Import… → Choose File System → Browse to D:\LPC1700CMSIS)

© NXP B.V. 2010. All rights reserved.



© NXP B.V. 2010. All rights reserved.

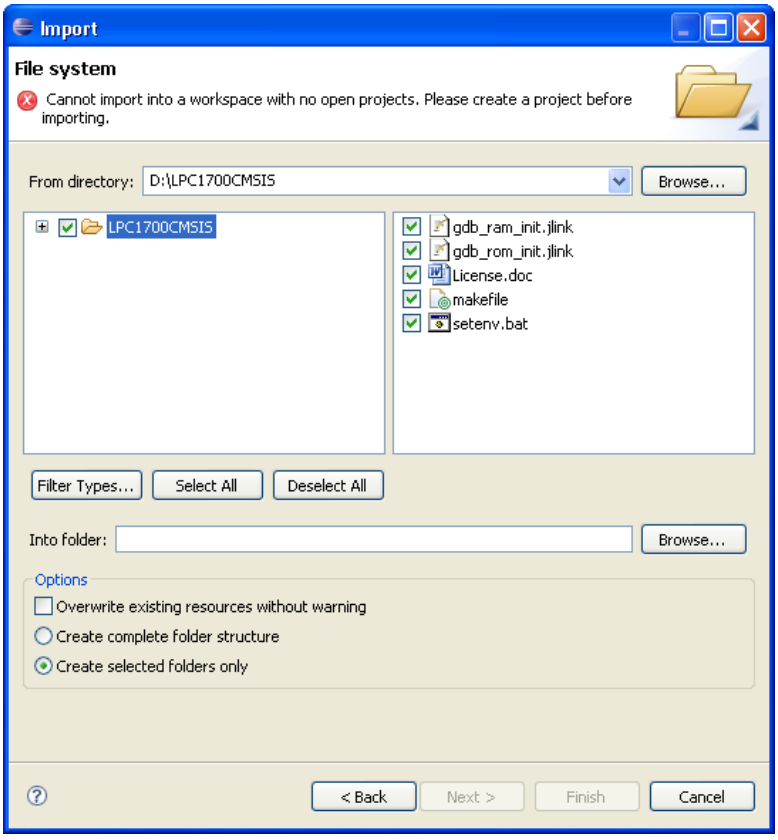


Fig 18. Import File system in resource package

The result should be like this on the left window (project window):

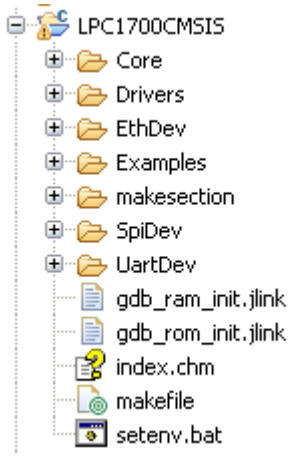


Fig 19. Result at the left window (Project window) after importing

Configure “make” environment: Open “makeconfig” file and edit the “PROJ_ROOT” symbol as shown in Fig 20:

© NXP B.V. 2010. All rights reserved.

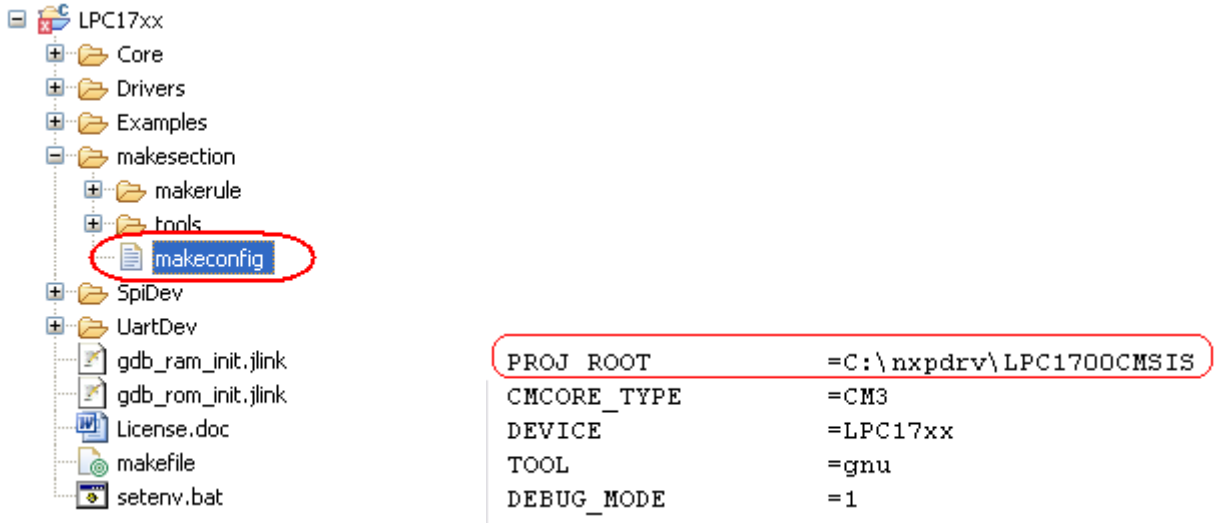


Fig 20. Configure "make" environment in "makeconfig" file

(In this case, "nxpdrv" means the eclipse workspace that was created first, and "LPC1700CMSIS" is the name of this project; note the forward slash style is used).

Close the Eclipse IDE.

4.5 Configure environment of "make utility"

In this case, all "make tool" resides in C:\nxpdrv\LPC1700CMSIS\makesection\tools

In order to make the system recognize the command located in private tool that its path is not included in the existing PATH variable of the system and user environment, the user must set the path of private tool to the "path" variable of the system variable environment.

In each project after importing the file system, all of the required tools are in the in "tools" folder; the user must set the path of the "tools" at the head of the "path" in system variables.

Note: Make sure all old paths of previous similar utility tools are removed in both user and system variables.

- On desktop, go to Start → Control Panel → System…

© NXP B.V. 2010. All rights reserved.

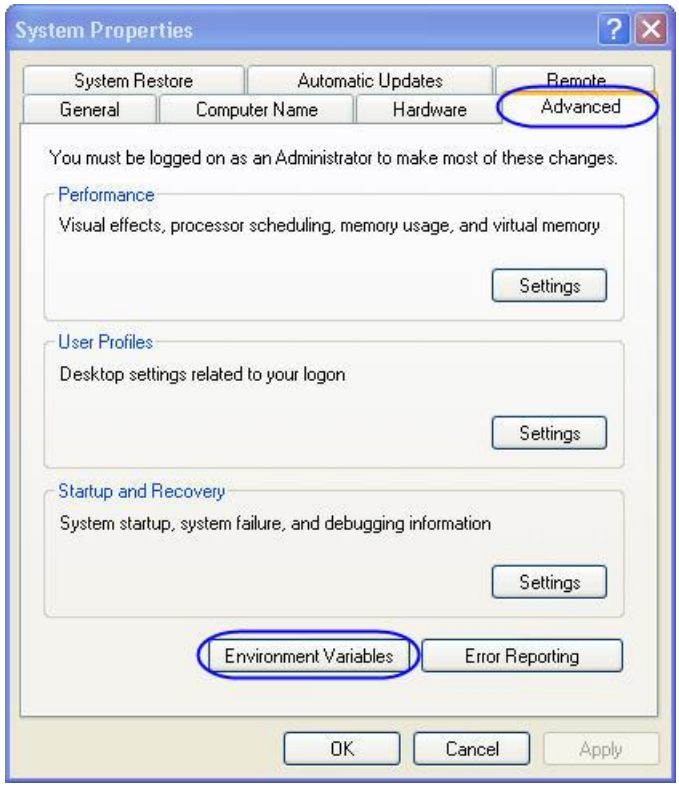


Fig 21. Environment configuration (Step 1)

© NXP B.V. 2010. All rights reserved.

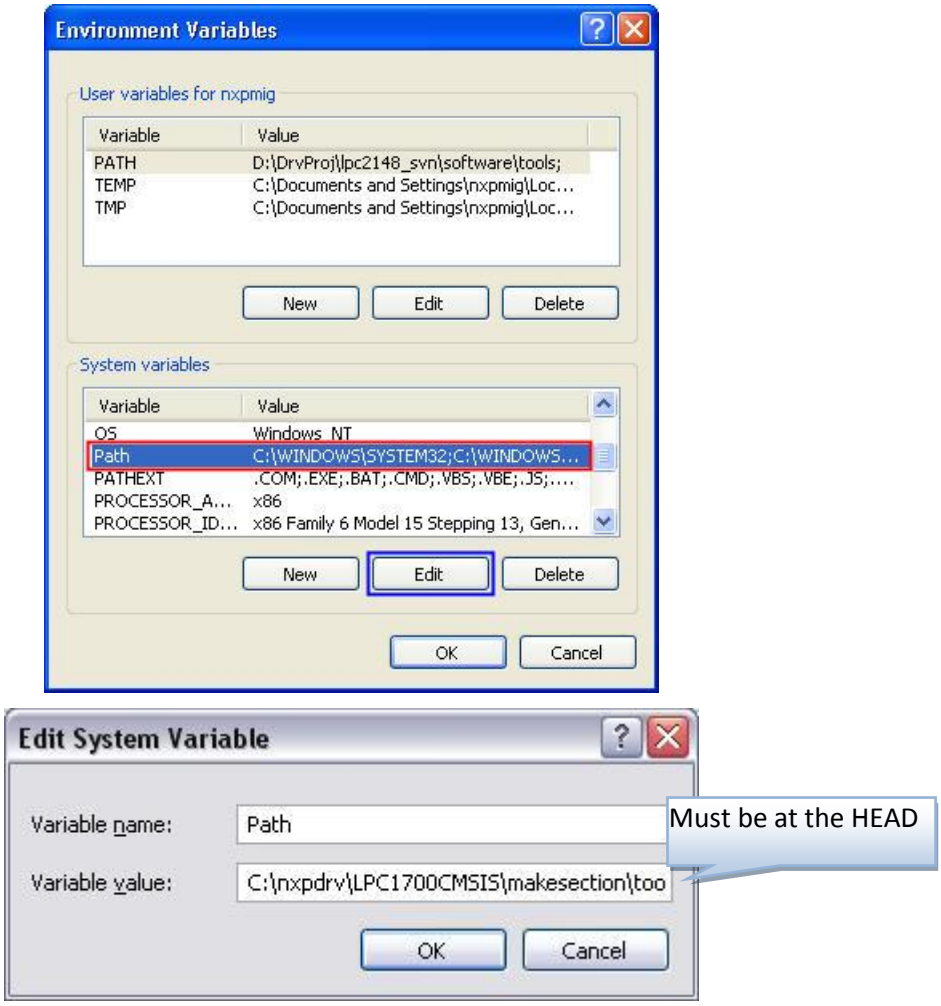


Fig 22. Environment configuration (Step 2)

4.6 Compile project

Re-open Eclipse, and then go through Project →Property… to configure the properties for the “LPC1700CMSIS” project as shown in Fig 23:

© NXP B.V. 2010. All rights reserved.

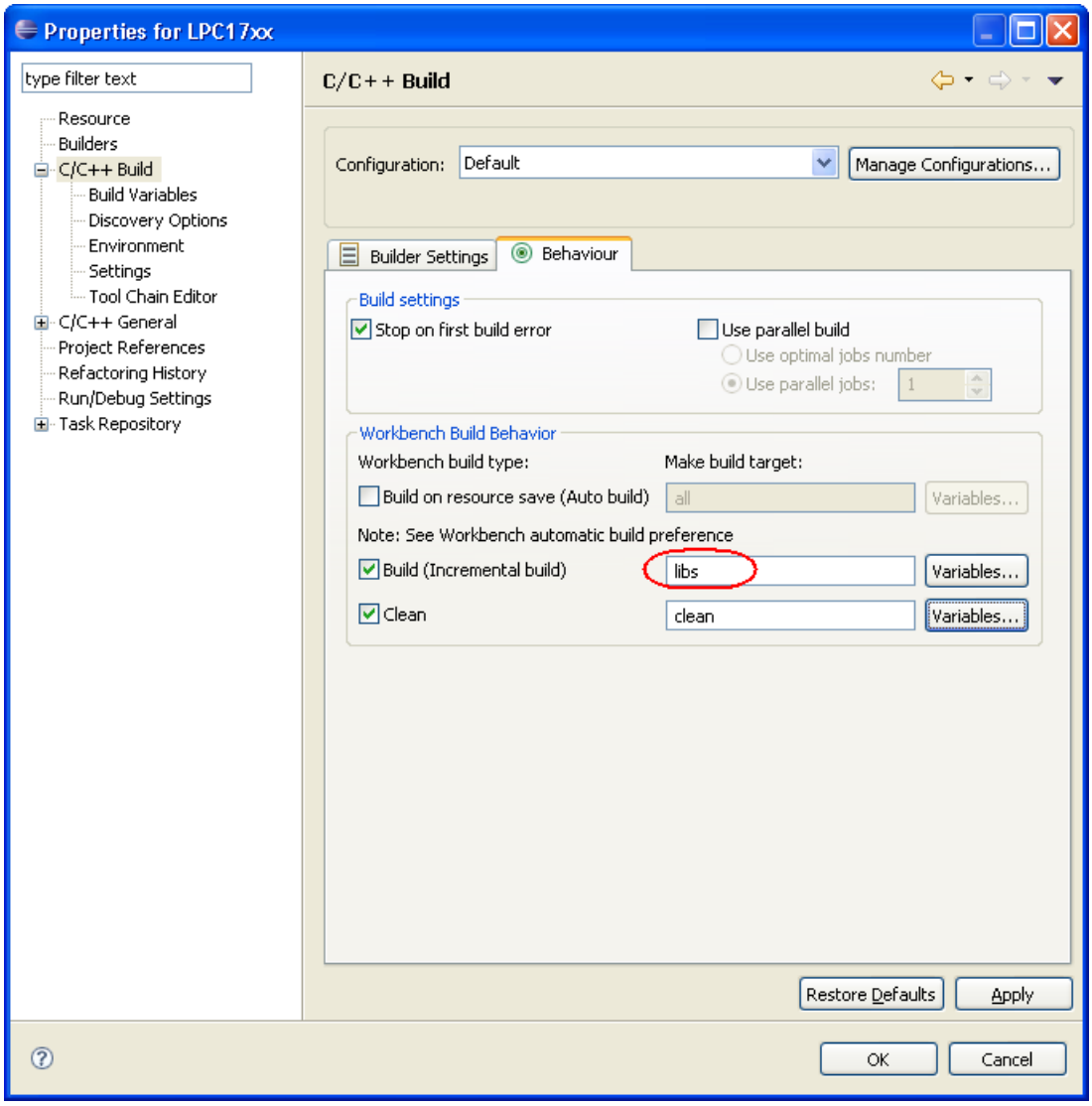


Fig 23. Configure “libs” for compiling entire project mode

© NXP B.V. 2010. All rights reserved.

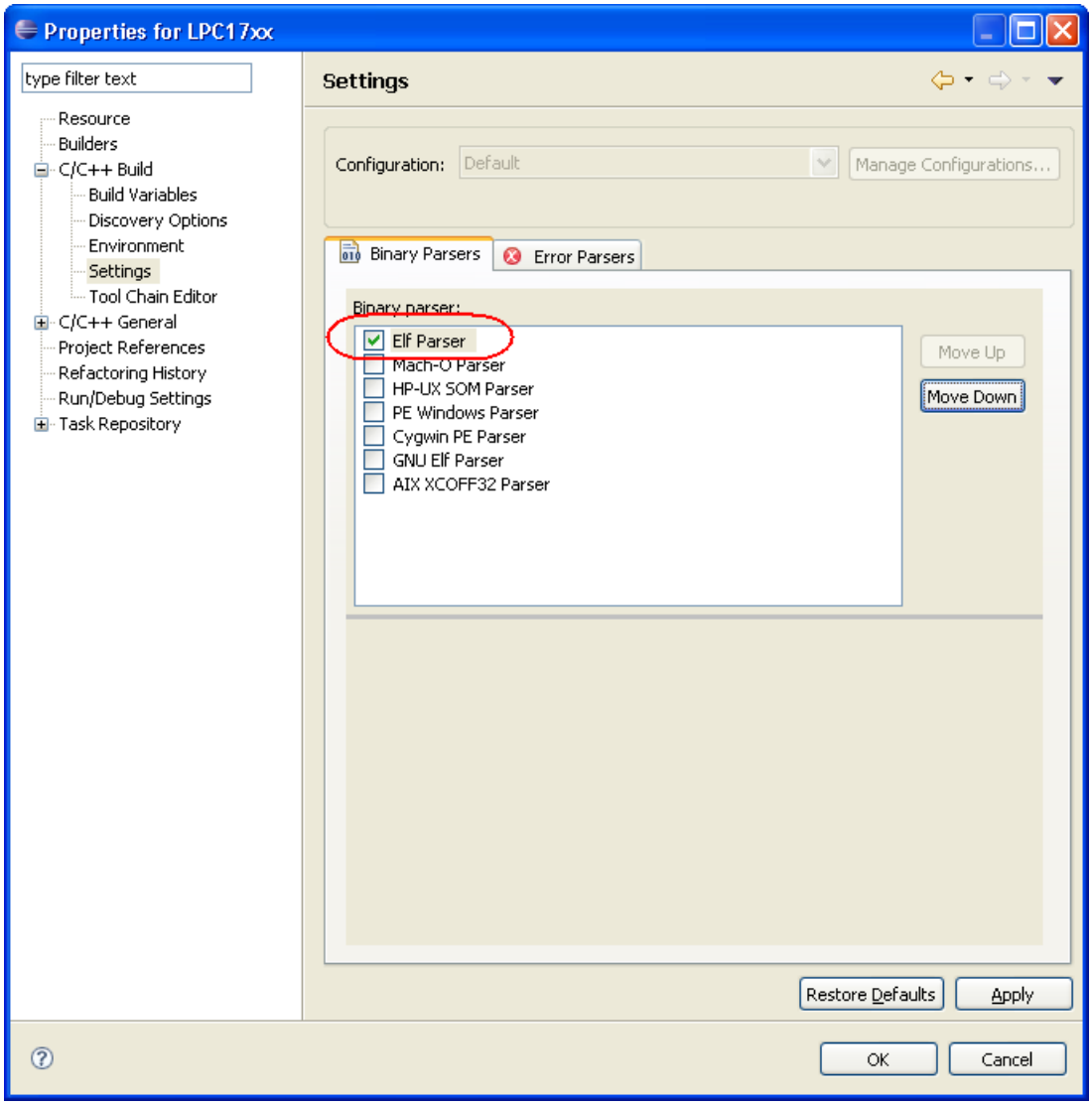


Fig 24. Check this option to allow .elf file to be recognized

Now, build entire the project…

© NXP B.V. 2010. All rights reserved.

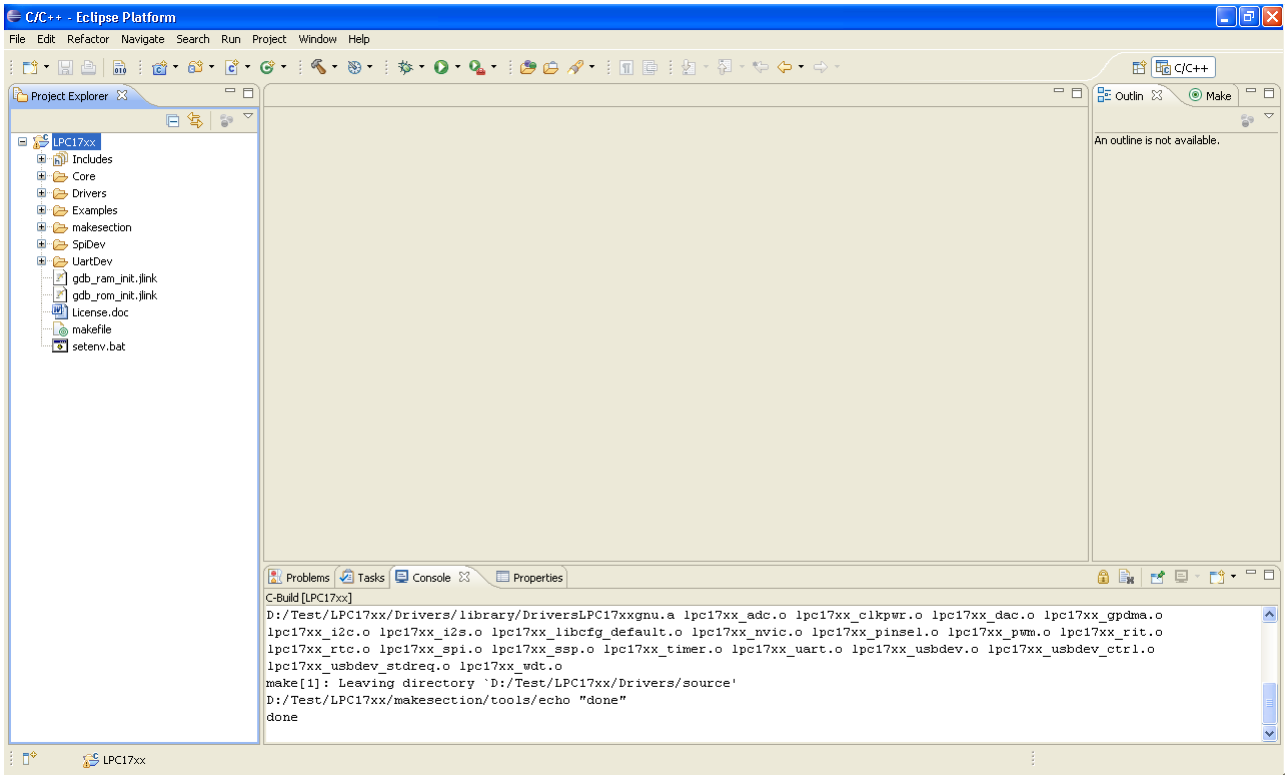


Fig 25. Result in Console window after building project

To build the example, right click on the root of that example → Make target → Build…

(Note: In this case, the example is placed at C:\nxpdrv\LPC1700CMSIS\Examples\, for each peripheral. There are some sub-folders that contain specific examples for various purposes to test this peripheral)

© NXP B.V. 2010. All rights reserved.

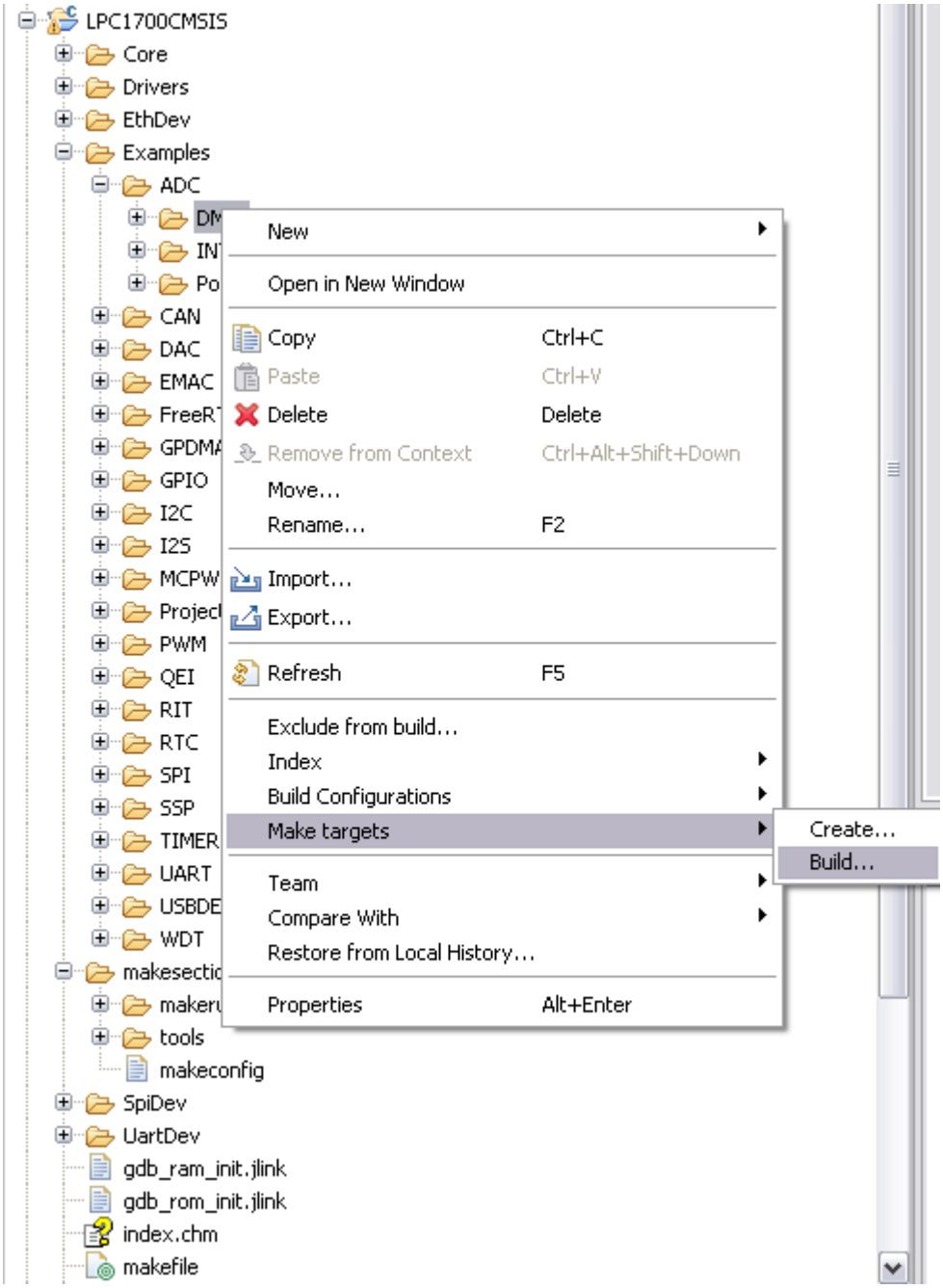


Fig 26. Create “make target” for each example (1)

Create these “make target”: “clean”, “ram” and “rom”…

- RAM mode, target name is 'ram'
- ROM mode, target name is 'rom'
- Clean: should be named as 'cleanall' in each example, this target should be executed before any 'make' session!!!

© NXP B.V. 2010. All rights reserved.

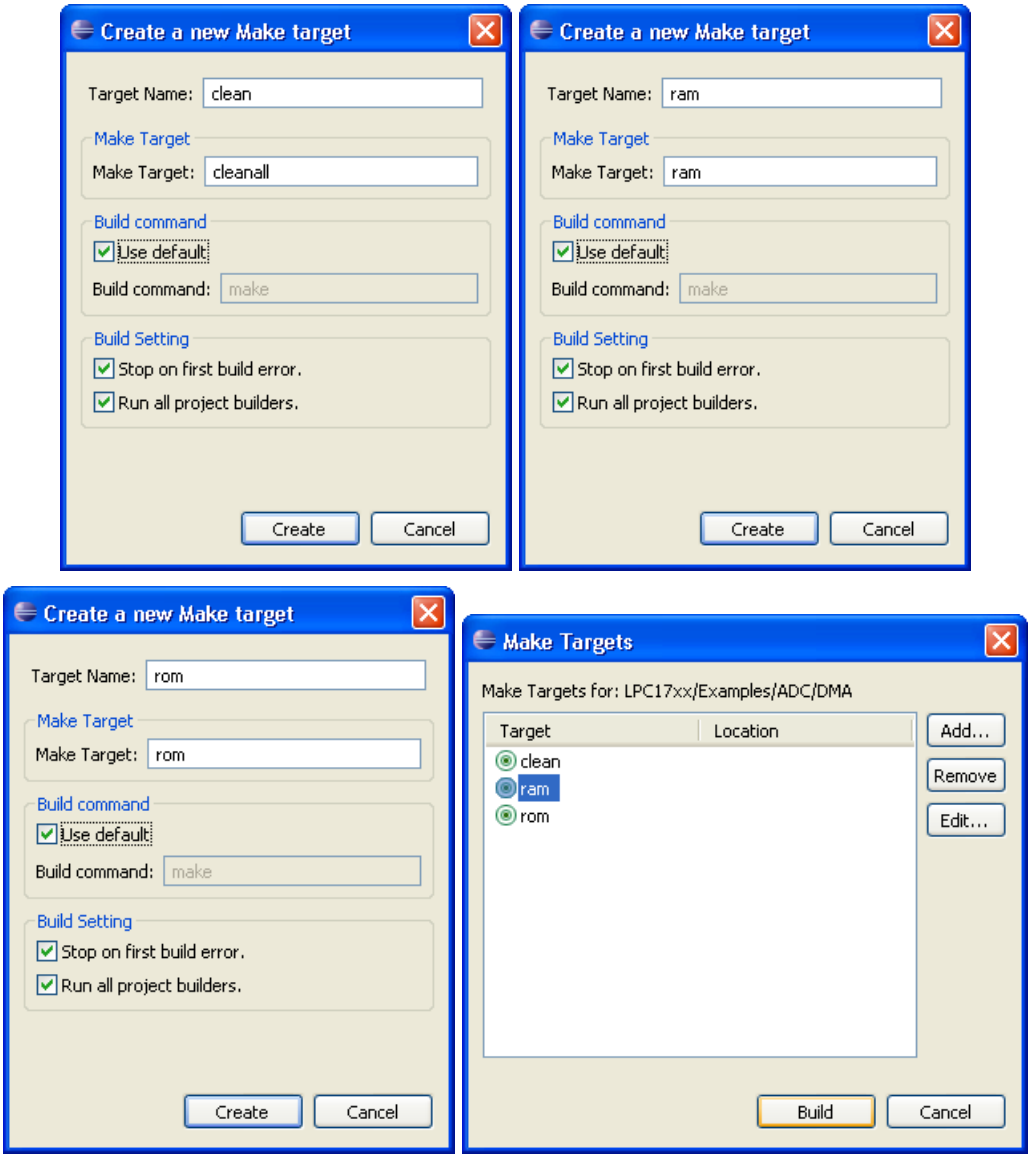


Fig 27. Create “make target” for each example (2)

Now, build the example in ram mode…

The result should like the example shown in Fig 28 after building successfully at the left window (Project window).

© NXP B.V. 2010. All rights reserved.

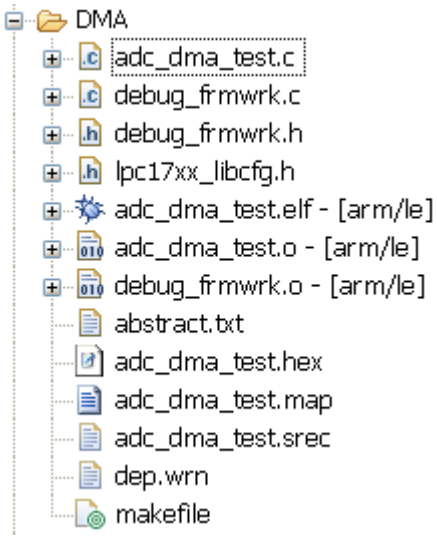


Fig 28. Result after building the “DMA” example

For more details, please refer to the “Using CodeSourcery toolchain and Eclipse IDE” document.

4.7 Execute example in ROM mode

Use FlashMagic to burn an image file (.hex) file (that can be observed only after building in ROM mode).

4.8 Debug the project

Basically, each example can be built in RAM mode or ROM (FLASH) mode.

Debugging in RAM mode requires fewer steps than ROM mode because the image file after building in RAM mode is loaded directly into the RAM section without any external tools.

4.8.1 Debug in RAM mode

Note: “make ram” must be completed before debugging in RAM mode

In this case, the “uart_interrupt” example is used to debug in RAM mode.

- Setting up Debug session: Go to menu Run → Open Debug Dialog…

© NXP B.V. 2010. All rights reserved.

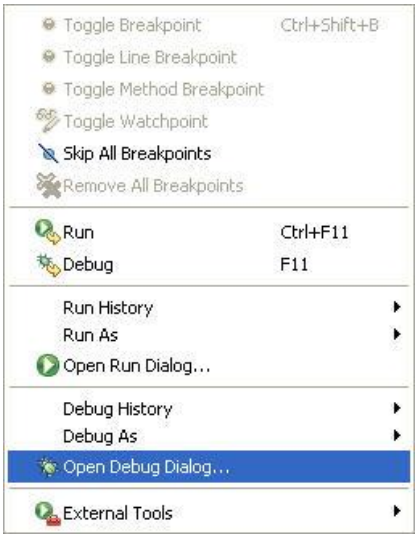


Fig 29. Open Debug dialog

© NXP B.V. 2010. All rights reserved.

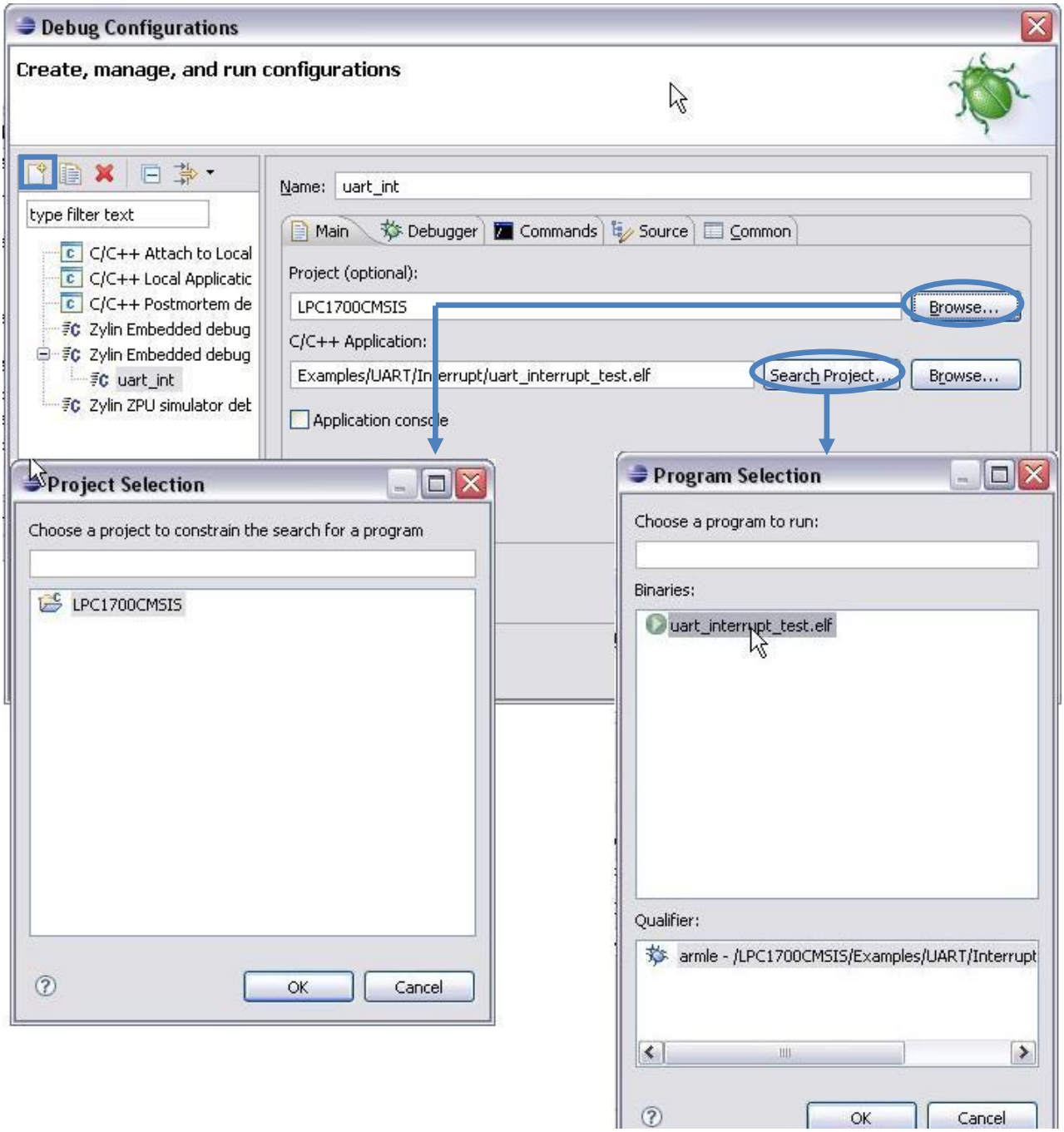


Fig 30. Create debug session (Step 1)

© NXP B.V. 2010. All rights reserved.

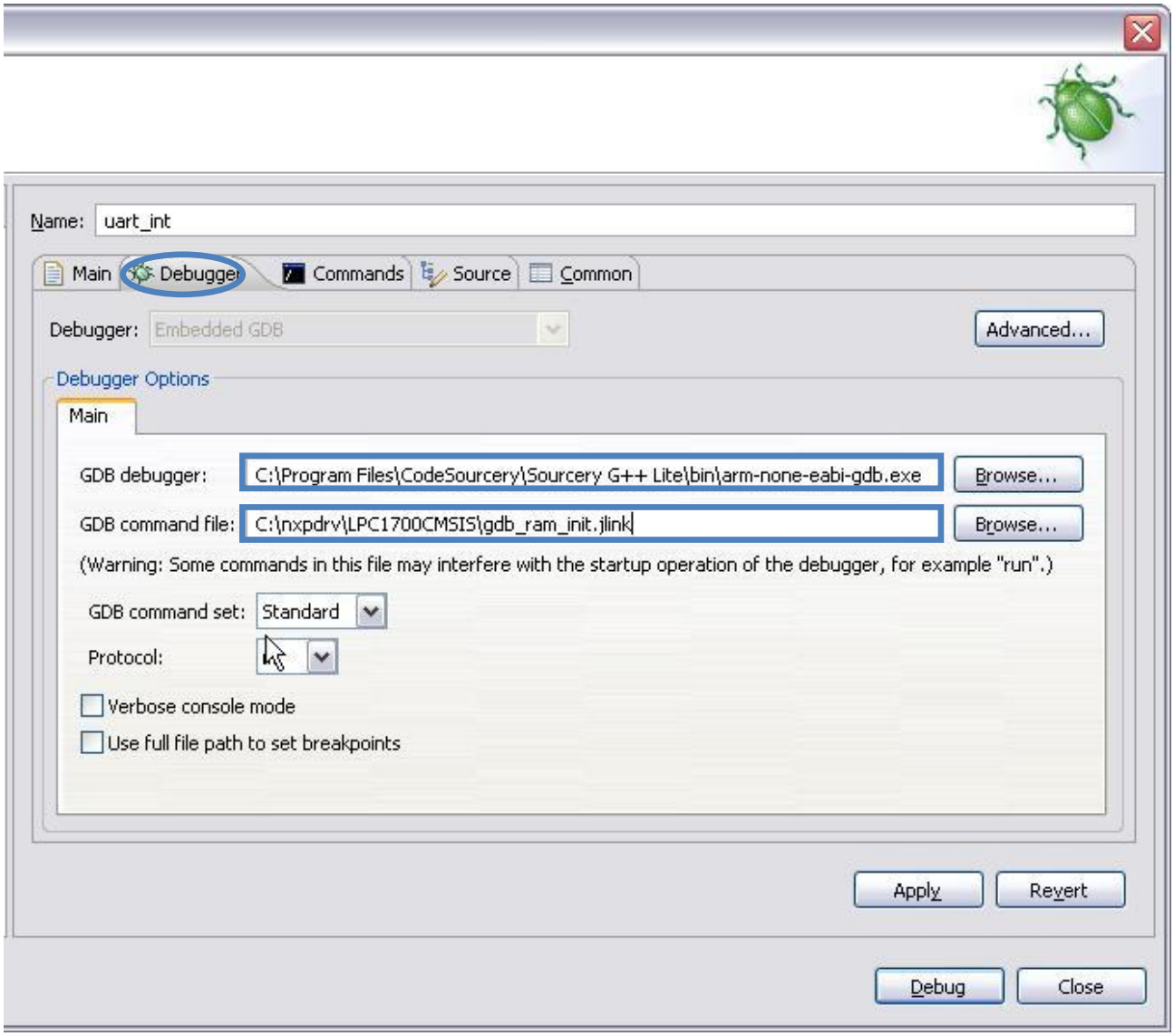


Fig 31. Create debug session (Step 2)

© NXP B.V. 2010. All rights reserved.

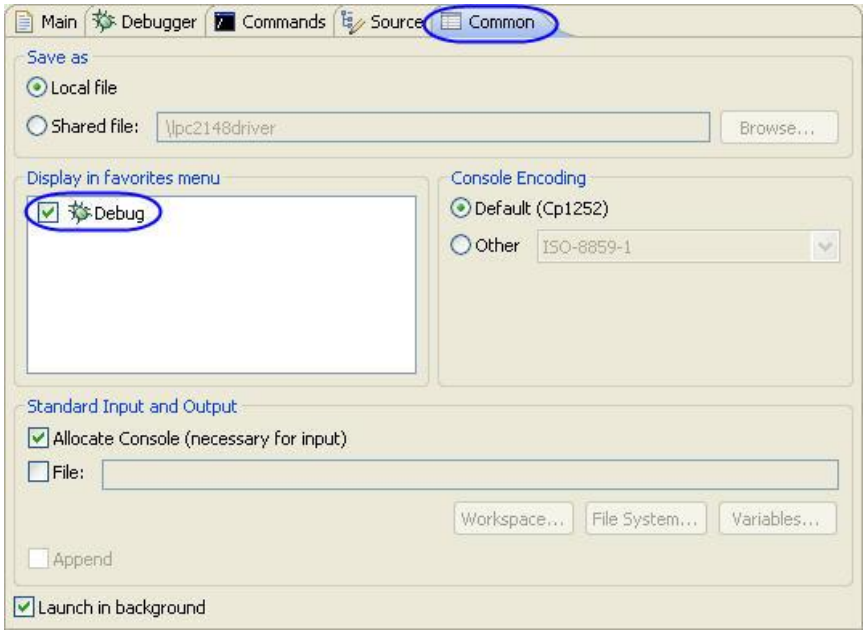


Fig 32. Create debug session (Step 3)

- Debugging in RAM mode (Note: Open “uart_interrupt_test.c” file to set breakpoint in next step).

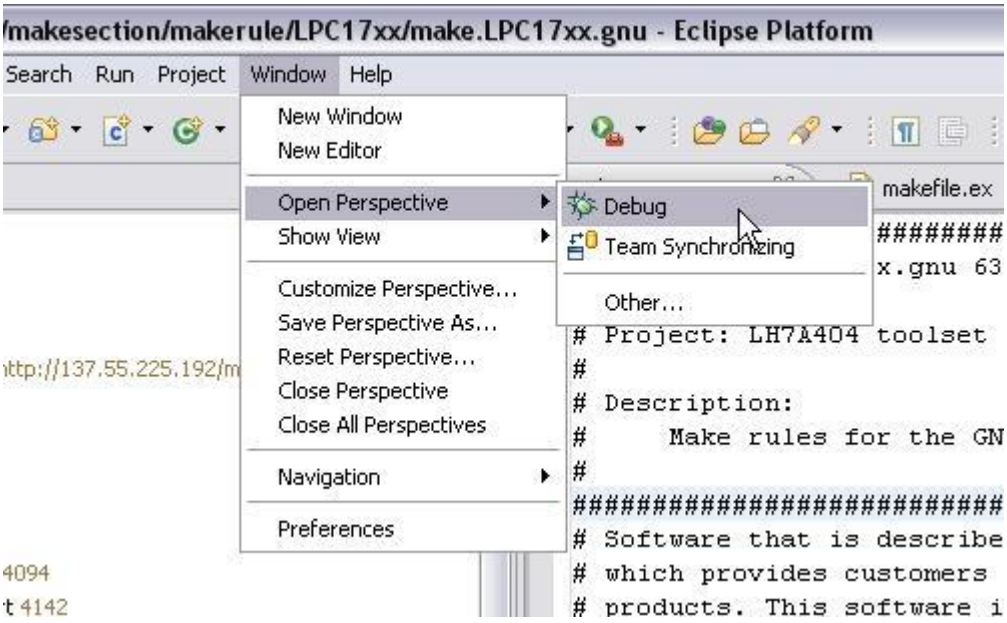


Fig 33. Open Perspective to debug window

© NXP B.V. 2010. All rights reserved.

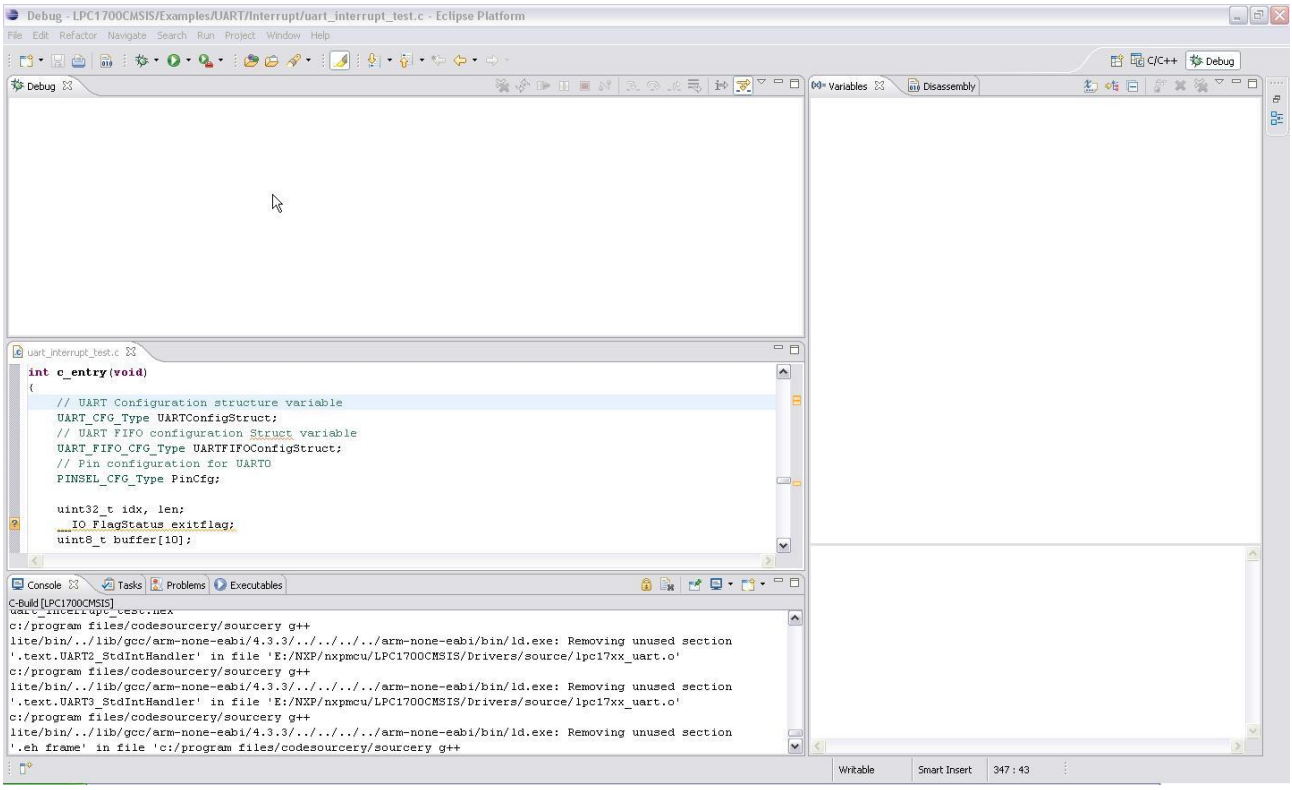


Fig 34. Overview of debug window

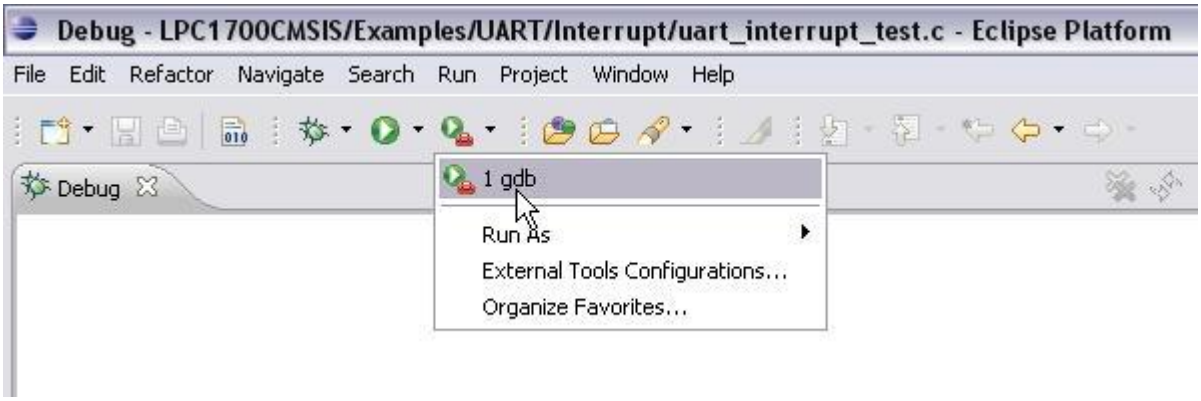


Fig 35. Open GDB server in external tool list

© NXP B.V. 2010. All rights reserved.

Fig 36. Start in free mode

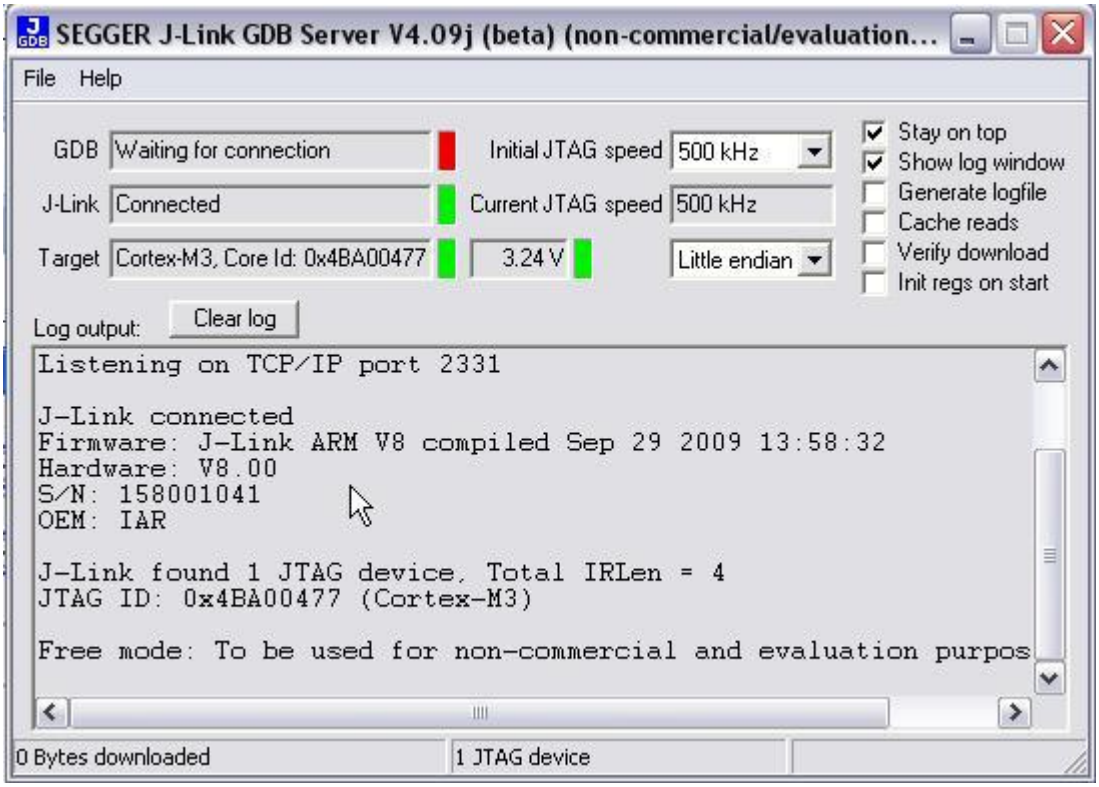


Fig 37. Configure parameter on J-link GDB server

© NXP B.V. 2010. All rights reserved.

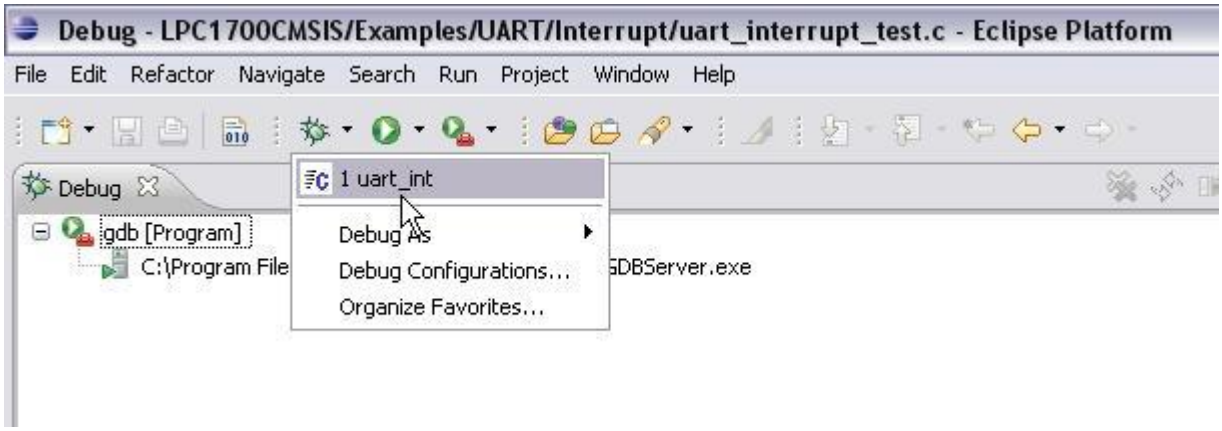


Fig 38. Open uart_interrupt_test debug session

Fig 39. All function buttons

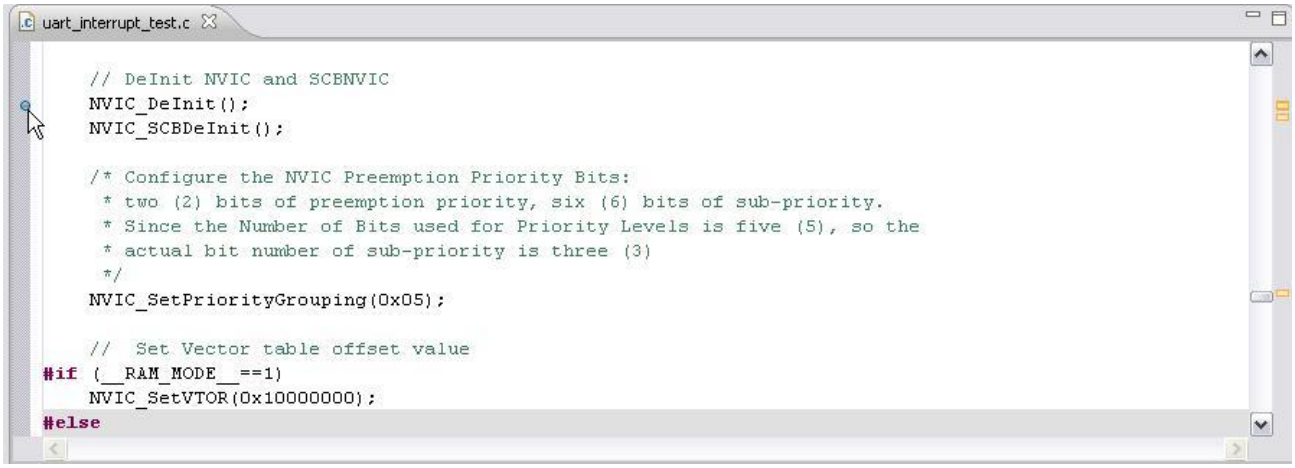


Fig 40. Set a break-point on uart_int.c file

© NXP B.V. 2010. All rights reserved.

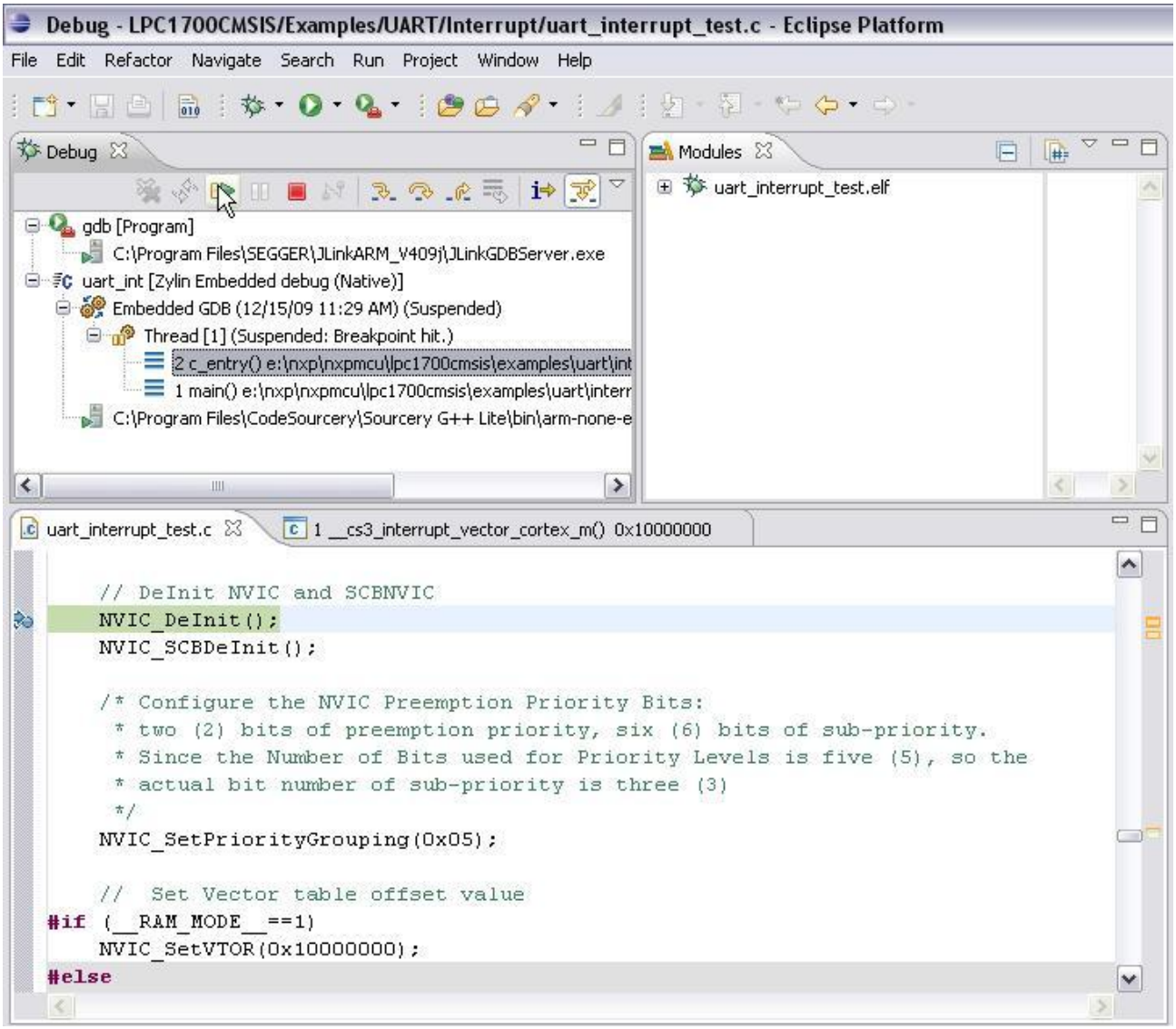


Fig 41. Run to the first break point

4.8.2 Debug in ROM (FLASH) mode

- Build the example in ROM mode first.
- Use Flash Magic to burn an image file (.hex) into the chip
- Start a debug session in ROM mode.

Steps to configure debugging session in ROM mode are the same with RAM mode.

In fact, Flash downloading using J-link is possible but this requires some license from SEGGER.

© NXP B.V. 2010. All rights reserved.

5. Compiling in command line

The package project is supported to build without Eclipse IDE.

5.1 Extract the package

In this case, the “zip” package is extracted on C:\nxpdrv, and a folder named “LPC1700CMSIS” will be generated on C:\nxpdrv.

Modify the value of symbol “PROJ_ROOT” in the “makeconfig” file corresponding to the current location of project root (note that a forward slash is used in this case):

PROJ_ROOT = C:\nxpdrv\LPC1700CMSIS

Set environment: Refer to chapter 4.5.

Here is the way to build the UART1 Full Modem example in RAM mode

Fig 42. Build example in RAM mode (Command prompt window)

Fig 43. Build example in ROM mode (Command prompt window)

Fig 44. Clean the example (Command prompt window)

© NXP B.V. 2010. All rights reserved.

6. Using LPC1000CMSIS driver package with extension tools

The project we used to represent here is LPC1700CMSIS. Other LPC1000CMSIS project are correlative.

6.1 Keil RealView - MDK

There's a Keil Project template inside .\Examples\Project_Template\Keil_RealView_MDK directory.

This project is configured to add all required startup file (asm file used for RealView compiler), ARM Cortex-M3 core files and driver source files.

Following target properties have been configured:

- Target device (LPC1768)
- Memory layout with linker.
- Output and list directory.
- C/C++ definitions and Common C/C++ Include Paths (.\Core\CM3\CoreSupport; .\Core\CM3\DeviceSupport\NXP\LPC17xx; .\Drivers\include)
- Flash programming and debug tool.

The main program is template.c for demonstration. However, this demo program can be easily removed to add other main program source files for user's purposes.

Here're all steps to import demo (main program) source file into exist Keil RealView-MDK project sample:

6.1.1 Open Keil RealView project

Follow .\Examples\Project_Template\Keil_RealView_MDK to open LPC1700CMSIS.uvproj (or .uv2 with older Keil project version) project file.

© NXP B.V. 2010. All rights reserved.

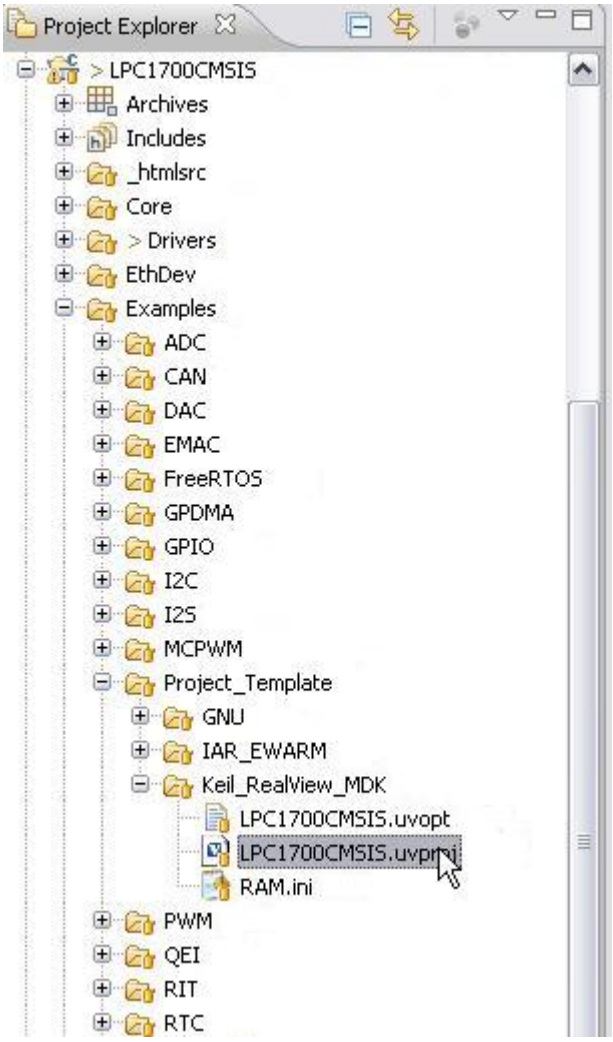
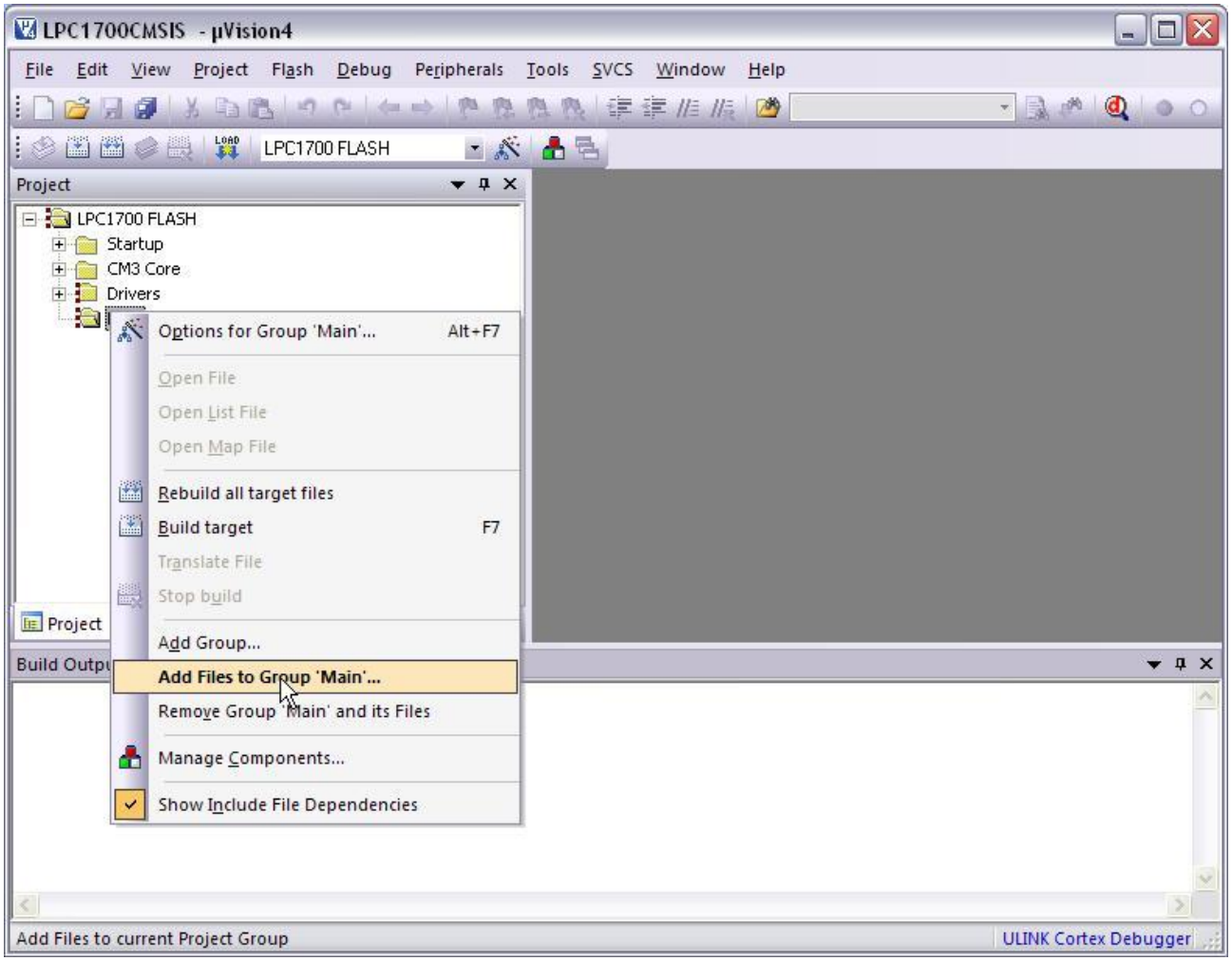


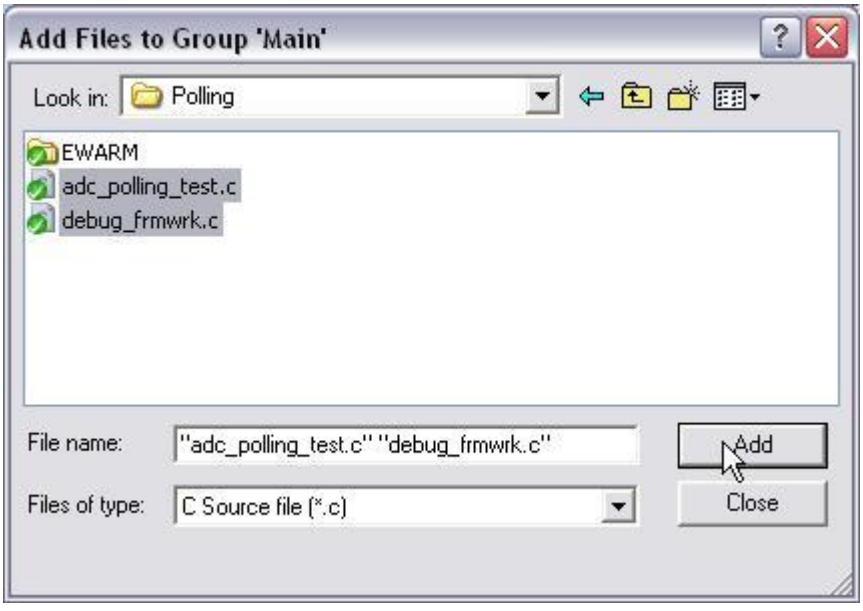
Fig 45. Open Keil RealView-MDK project sample

6.1.2 Add main program source files

ADC Polling example is used in this case. Add all required source files (*.c) in .\Examples\ADC\Polling directory:

© NXP B.V. 2010. All rights reserved.





© NXP B.V. 2010. All rights reserved.

Fig 46. Add main program source file to project

6.1.3 Add Included Path

Add .\Examples\ADC\Polling into C/C++ Include Path of project.

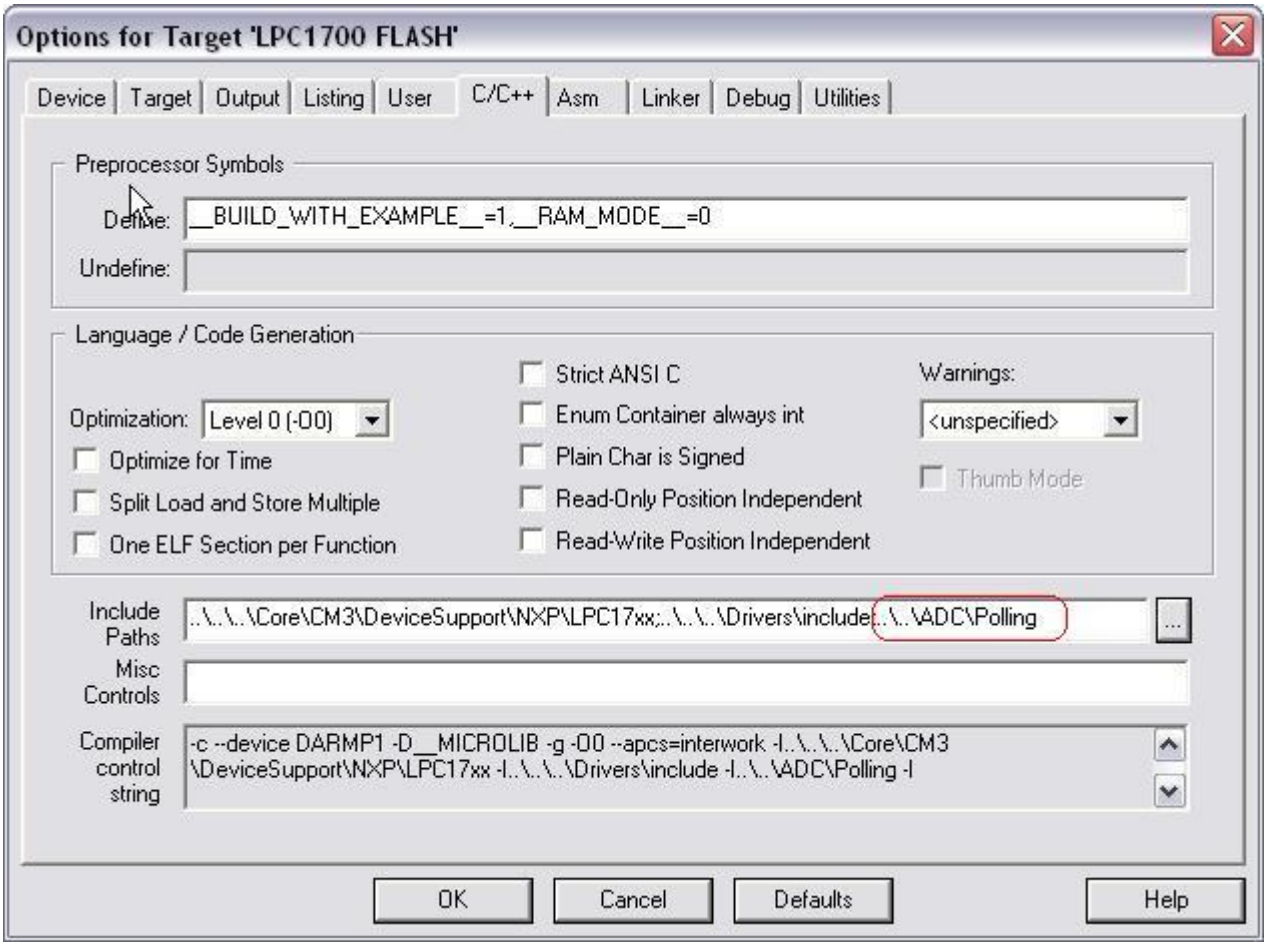


Fig 47. Add include path

Note:

There's a file named "lpc17xx_libcfg.h" in each example directory. This file used to enable/disable individual peripheral driver in driver source code for compiler to build.

Specified symbol should be defined in order to enable corresponding peripheral driver used in main program.

For example:

If ADC, UART0 and TIMER are used in this demo program, corresponded symbol for these two peripheral drivers should be defined:

#define _ADC 1

#define _UART 1

#define _UART0 1

© NXP B.V. 2010. All rights reserved.

#define _TIM 1

See more in .\Drivers\Include\lpc17xx_libcfg_default.h to know more detail about these symbols.

6.1.4 Build project and get target

The rest is very common. Execute build then burn image file into target and let it run!

6.2 IAR Embedded Workbench IDE

In each example has attached IAR project in EWARM folder. It allows you run this example in IAR Embedded Workbench IDE. This project is configured to add all required startup file (asm file used for IAR compiler), ARM Cortex-M3 core files and driver source files.

Here're all steps to run exist IAR examples or import demo source file into exist IAR project sample.

6.2.1 Use exist IAR example

Run .eww file in EWARM folder. Such as: follow .\Example\ADC\Polling\EWARM to open adc_polling.eww file.

Click "Dowload and Debug" icon to start building and let it run.

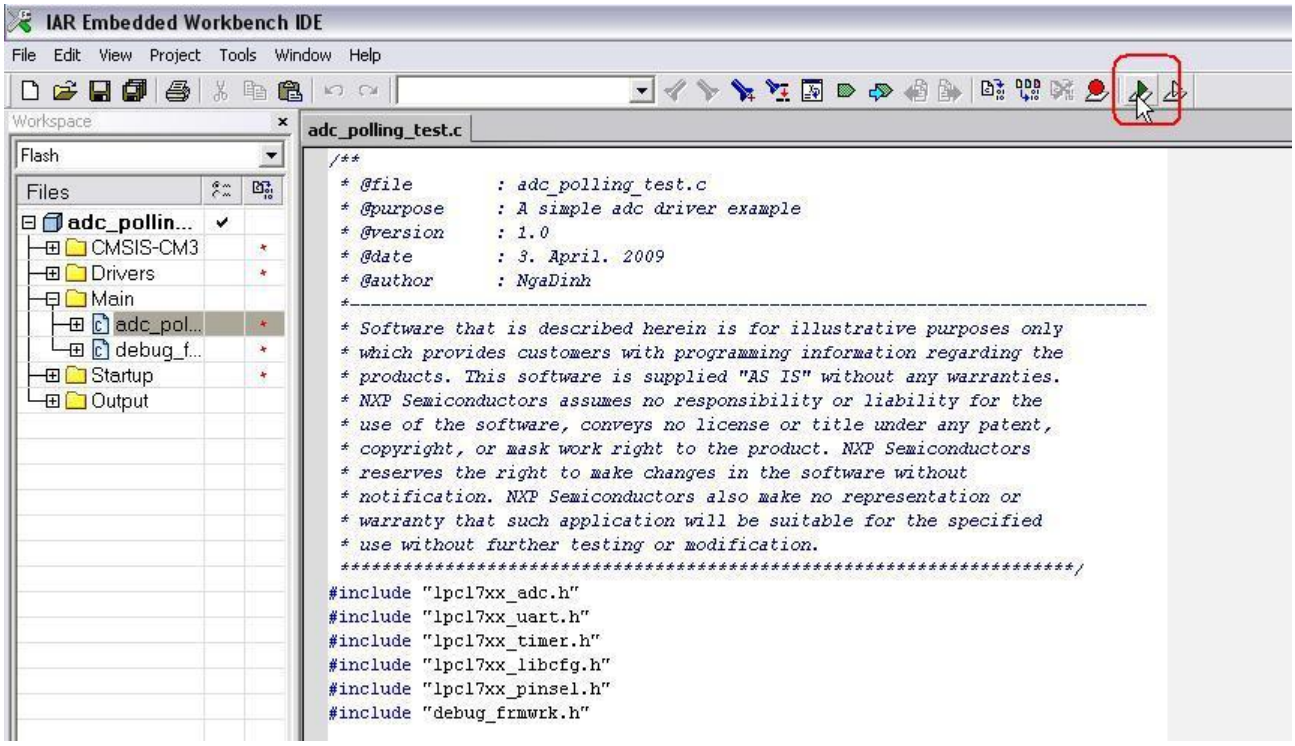


Fig 48. Run IAR example

© NXP B.V. 2010. All rights reserved.

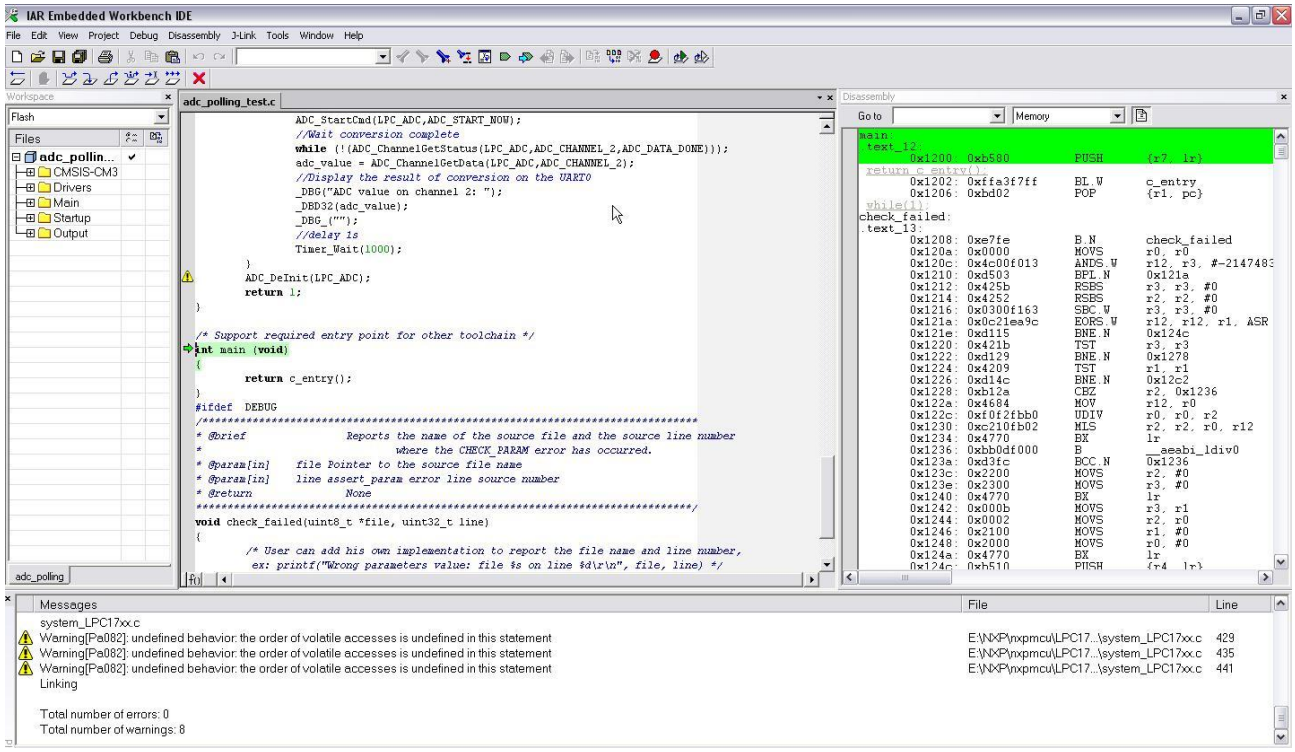


Fig 49. Debug IAR window

Switch to RAM workspace if want to run in RAM mode.

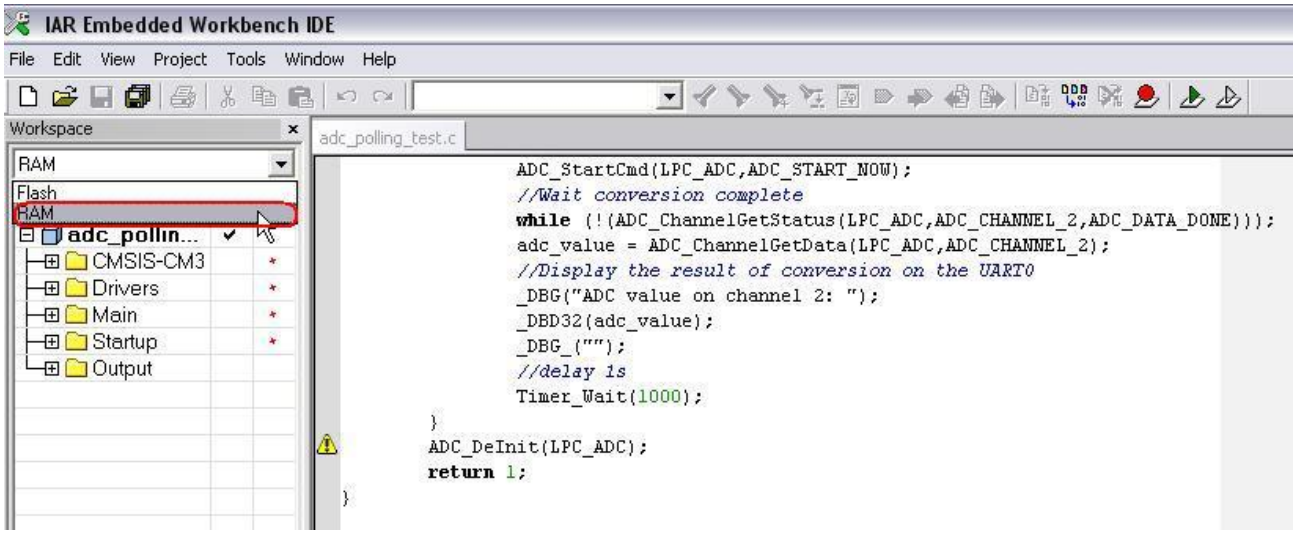


Fig 50. Switch to RAM mode

© NXP B.V. 2010. All rights reserved.

6.2.2 Import demo source code in exist IAR project sample

There's a IAR Project template inside .\Examples\Project_Template\IAR_EWARM directory. The main program is template.c for demonstration. However, this demo program can be easily removed to add other main program source files for user's purposes.

6.2.2.1 Open IAR project

Follow .\Example\Project_Template\IAR_EWARM directory to open LPC1700CMSIS.eww file.

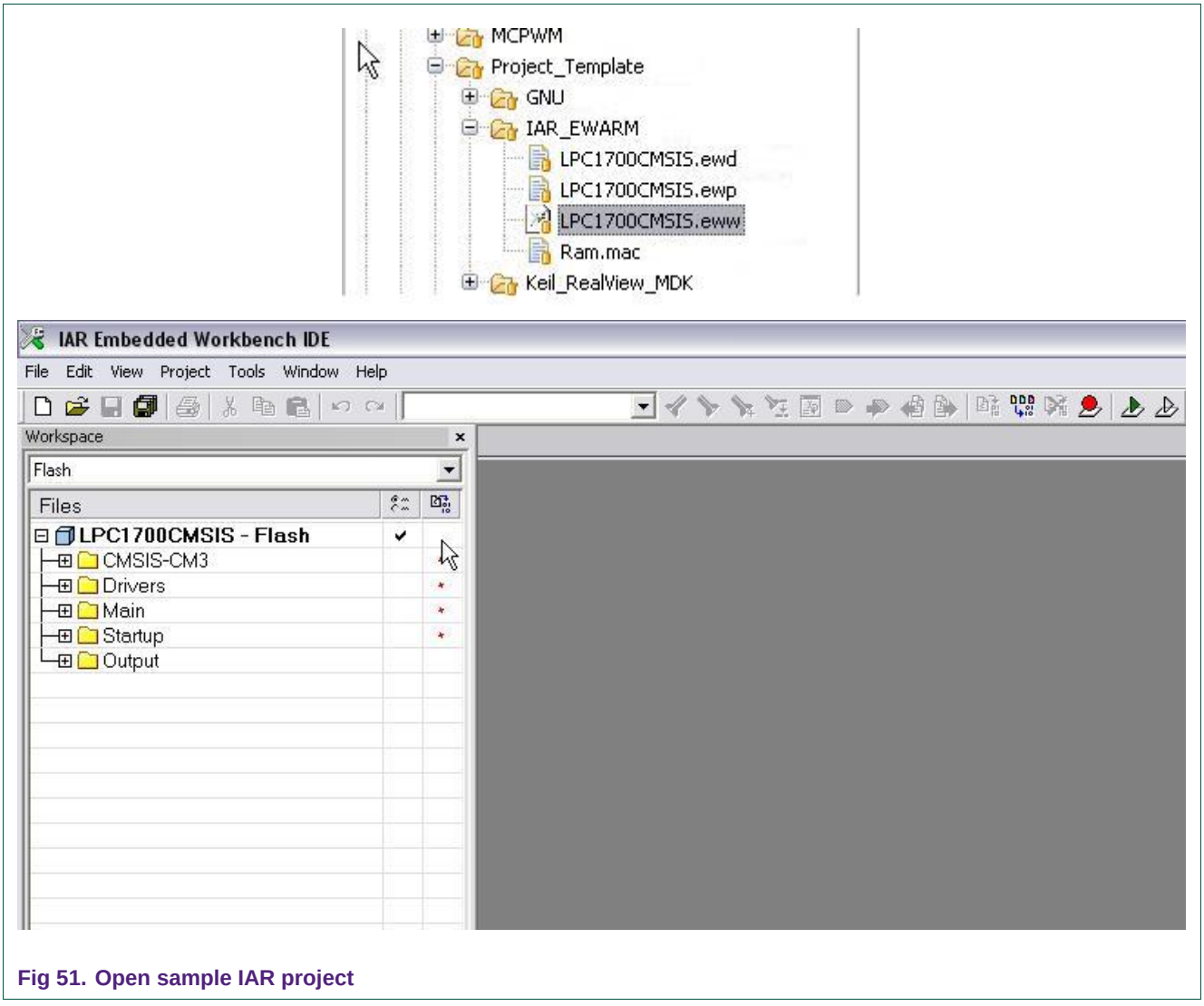
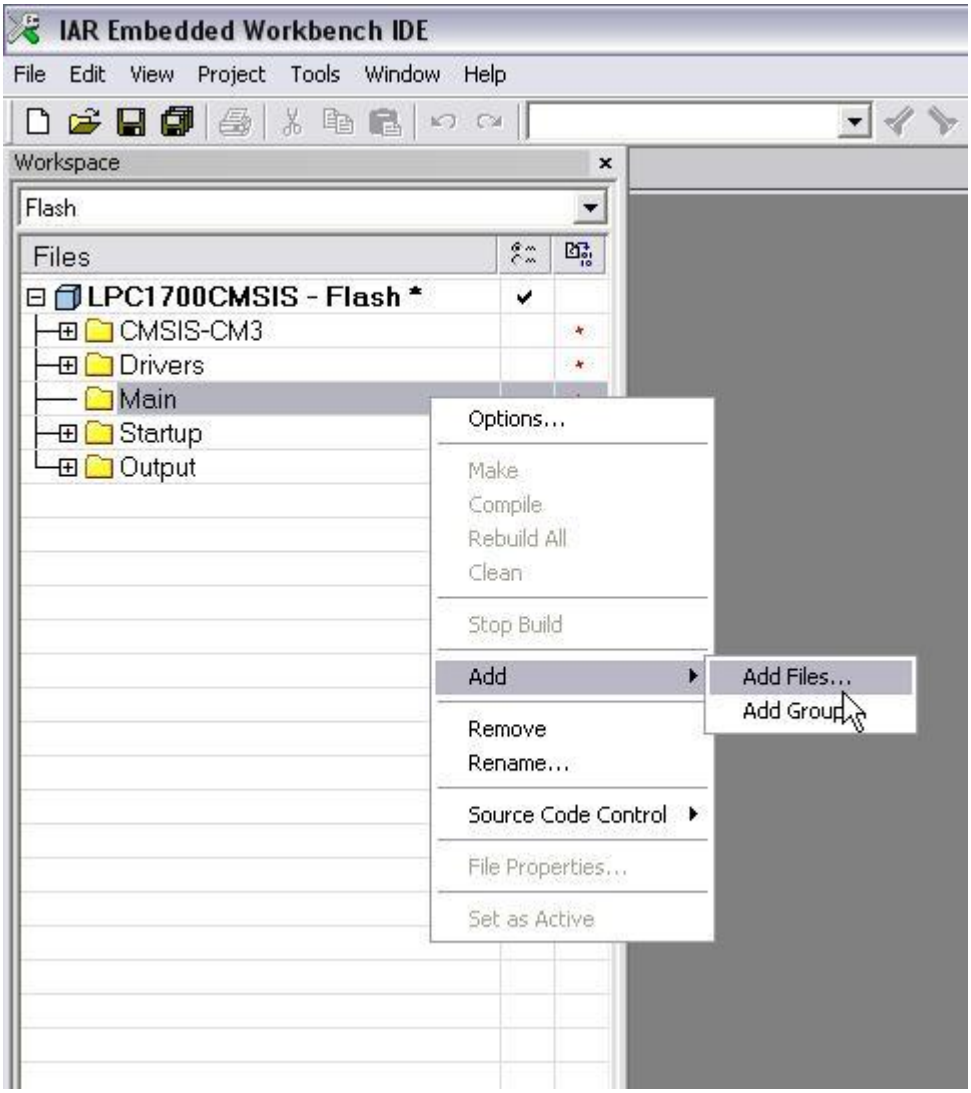


Fig 51. Open sample IAR project

6.2.2.2 Add main source code

ADC Polling example is used in this case. Add all required source files (*.c) in .\Examples\ADC\Polling directory:

© NXP B.V. 2010. All rights reserved.



© NXP B.V. 2010. All rights reserved.

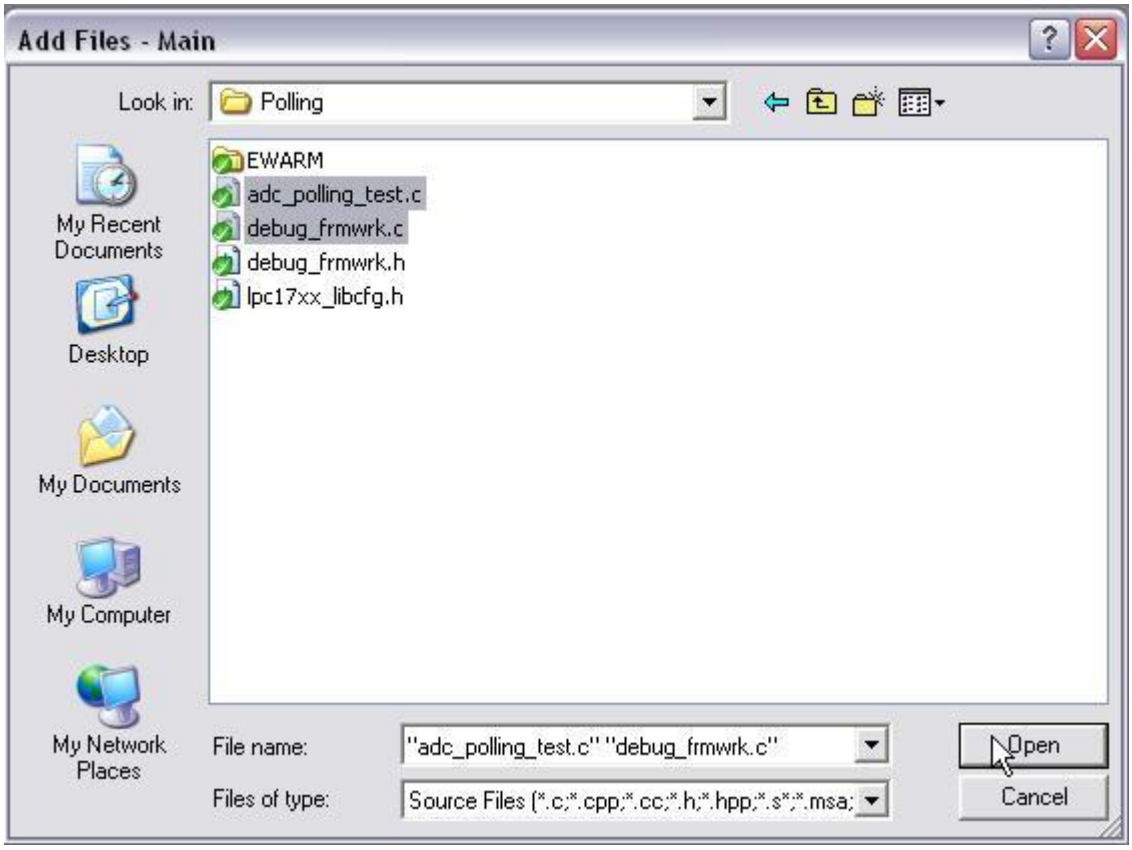


Fig 52. Add source code in IAR project

6.2.2.3 Add include path

Open Option window by right click into LPC1700CMSIS-Flash workspace.

Add .\Examples\ADC\Polling into C/C++ Compiler/ Preprocessor tab as follows:

© NXP B.V. 2010. All rights reserved.

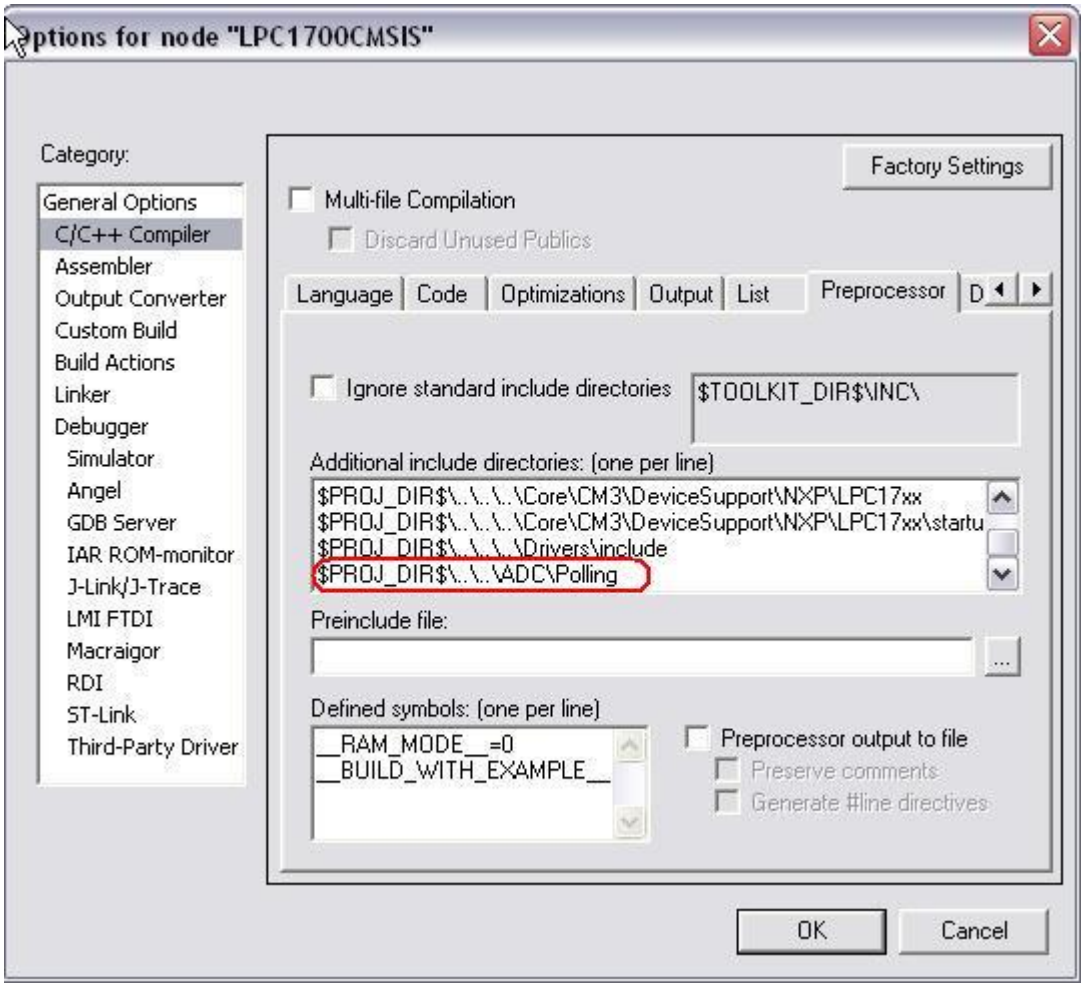


Fig 53. Add include path

Now, follow the same steps in “Use exist IAR example” to run it.

© NXP B.V. 2010. All rights reserved.

7. Appendix

Project Files Structure Diagram

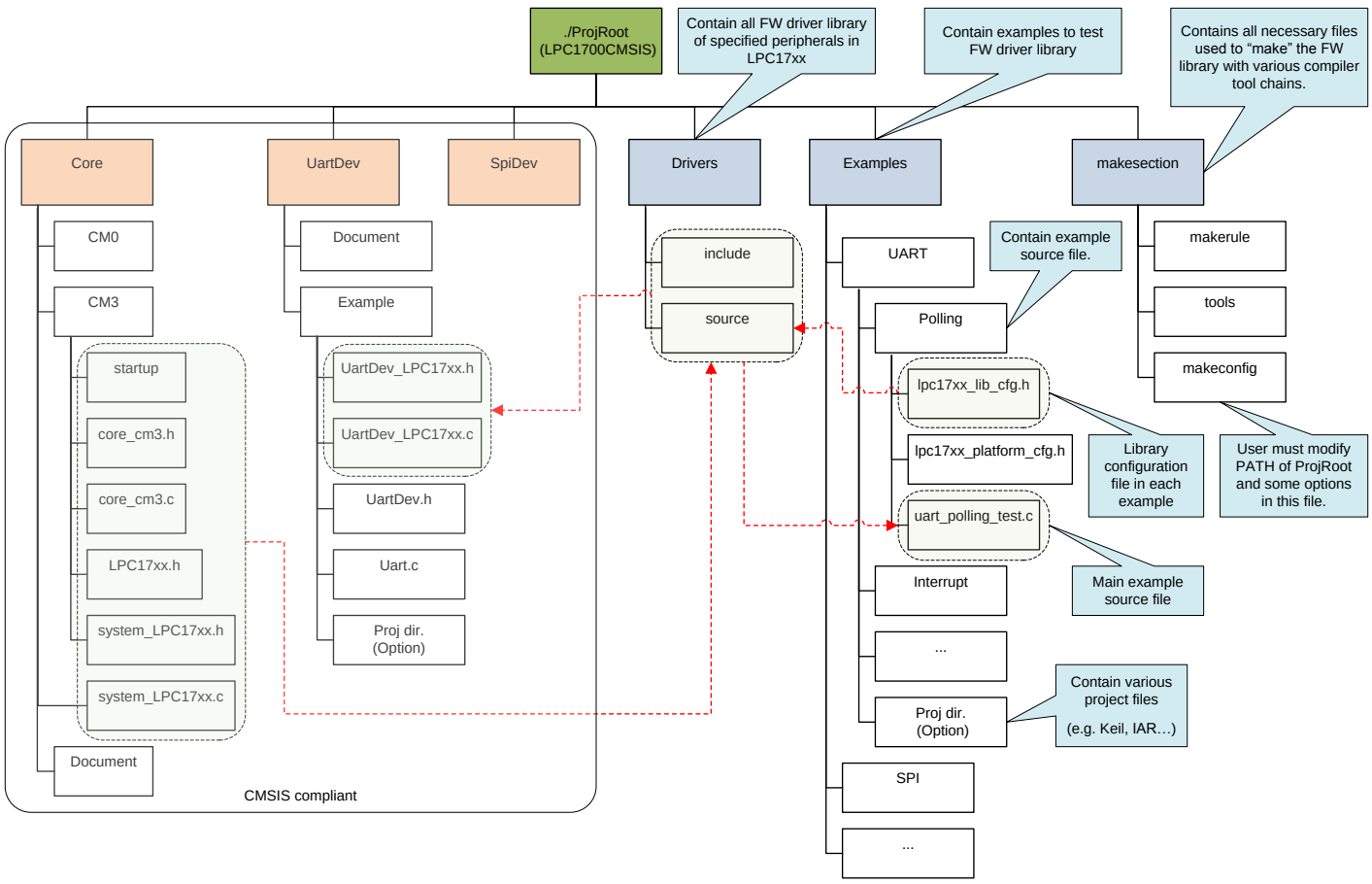


Fig 54. Project Files Structure Diagram

© NXP B.V. 2010. All rights reserved.

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

8.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

8.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

© NXP B.V. 2010. All rights reserved.

9. Contents

1. Introduction 3
2. Software update requirement 3
2.1 Java Runtime Enviroment 3
2.2 CodeSourcery toolchain 4
2.3 Downloading Eclipse IDE and plug-in 4
2.3.1 Download Eclipse IDE for C/C++ 4
2.3.2 Install Zylin CDT plug-in 5
2.4 Segger J-link 6
2.5 Flash Magic 6
3. Running Eclipse for the first time 6
3.1 Eclipse workspace setup 6
3.2 Setting up External tools for Eclipse 9
4. Creating and working with LPC1000CMSIS project 15
4.1 Create new workspace 15
4.2 Create new project 15
4.3 Extract the package 15
4.4 Import package resource file into project 15
4.5 Configure environment of “make utility” 18
4.6 Compile project 20
4.7 Execute example in ROM mode 26
4.8 Debug the project 26
4.8.1 Debug in RAM mode 26
4.8.2 Debug in ROM (FLASH) mode 34
5. Compiling in command line 35
5.1 Extract the package 35
6. Using LPC1000CMSIS driver package with extension tools 36
6.1 Keil RealView - MDK 36
6.1.1 Open Keil RealView project 36
6.1.2 Add main program source files 37
6.1.3 Add Included Path 39
6.1.4 Build project and get target 40
6.2 IAR Embedded Workbench IDE 40
6.2.1 Use exist IAR example 40
6.2.2 Import demo source code in exist IAR project sample 42
6.2.2.1 Open IAR project 42
6.2.2.2 Add main source code 42
6.2.2.3 Add include path 44
7. Appendix 46
8. Legal information 47
8.1 Definitions 47
8.2 Disclaimers 47
8.3 Trademarks 47
9. Contents 48

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

© NXP B.V. 2010. All rights reserved.

For more information, please visit: http://www.nxp.com
For sales office addresses, email to: salesaddresses@nxp.com

Date of release: 25 January 2010
Document identifier: AN10862_3

founded by
PHILIPS