AN10864

Importing FreeRTOS to LPC17xx CMSIS project

Rev. 01 — 28 August 2009 Application note

Document information

Info	Content
Keywords	FreeRTOS, LPC1700
Abstract	Import FreeRTOS to LPC1700CMSIS project

Revision history

Rev	Date	Description
01	20090828	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

1. Introduction

FreeRTOSTM is a portable, open source, royalty free, mini Real Time Kernel - a free to download and free to deploy RTOS that can be used in commercial applications.

This document describes how to import open FreeRTOS source code into an LPC1700CMSIS project package to be able to compile properly.

2. Requirement

2.1 FreeRTOS source

The FreeRTOS version used to demonstrate is Version 5.4.0; it can be downloaded from this site below:

http://www.nxp.com/redirect/freertos.org/

2.2 Tool suite

Current compiler tool used is CodeSourcery based on GNU-GCC version 4.3.3 with Eclipse IDE.

Please refer to “LPC17xx_SoftwareDevelopmentToolchain” document for information regarding:

- Compiler toolchain installation
- Java + Eclipse IDE installation
- Flash Magic installation
- J-link Segger installation (option – used to debug program in case of using with debugger toolset)
- System Environment setting
- How to import source code in a release package to project in Eclipse.

2.3 Hardware

2.3.1 Basic

MCB1700 version 1.0 from Keil comes with NXP LPC1768.

See http://www.nxp.com/redirect/keil.com for more details.

Notes:

FreeRTOS had a demo on LPC1766. This demo was developed on an LPC1766 that had been mounted on a MCB2300 development board. But this demo is now deprecated.

The new one is NXP LPC1768 Cortex-M3 Red Suite demo that runs on RDB1768 from CodeRed.

This document shows how to configure an environment to build with CodeSourcery toolchain on Eclipse IDE and modify existing source code to let the demo run on MCB1700.

MCB1700 has some jumpers that are related to functionality configuration and need to be set accordingly to requirement of this demo. The default jumper setting state can be found in “LPC17xx_ExampleDescription” document, chapter 2 or this link below:

http://www.nxp.com/redirect/keil.com/mcb1700_su_jumpers

2.3.2 Addition

Make sure all jumpers below are in the correct state during normal operation on this demo:

- VBUS: shorted.
- VDDIO: shorted.
- VDDREG: shorted.
- LED: shorted.
- E/U: shorted between pin 2 and pin 3 (further side from USB port).
- E/C: shorted between pin 2 and pin 3 (further side from Ethernet port).
- RST and ISP: shorted during Flash burning and need to be removed during normal operation.

3. Procedure

3.1 Import, modify source file content and create “make” target

- Extract FreeRTOS source file.
- Go to the “.\FreeRTOS\Demo\CORTEX_LPC1768_GCC_RedSuite” folder, then run “CreateProjectDirectoryStructure.bat” file.

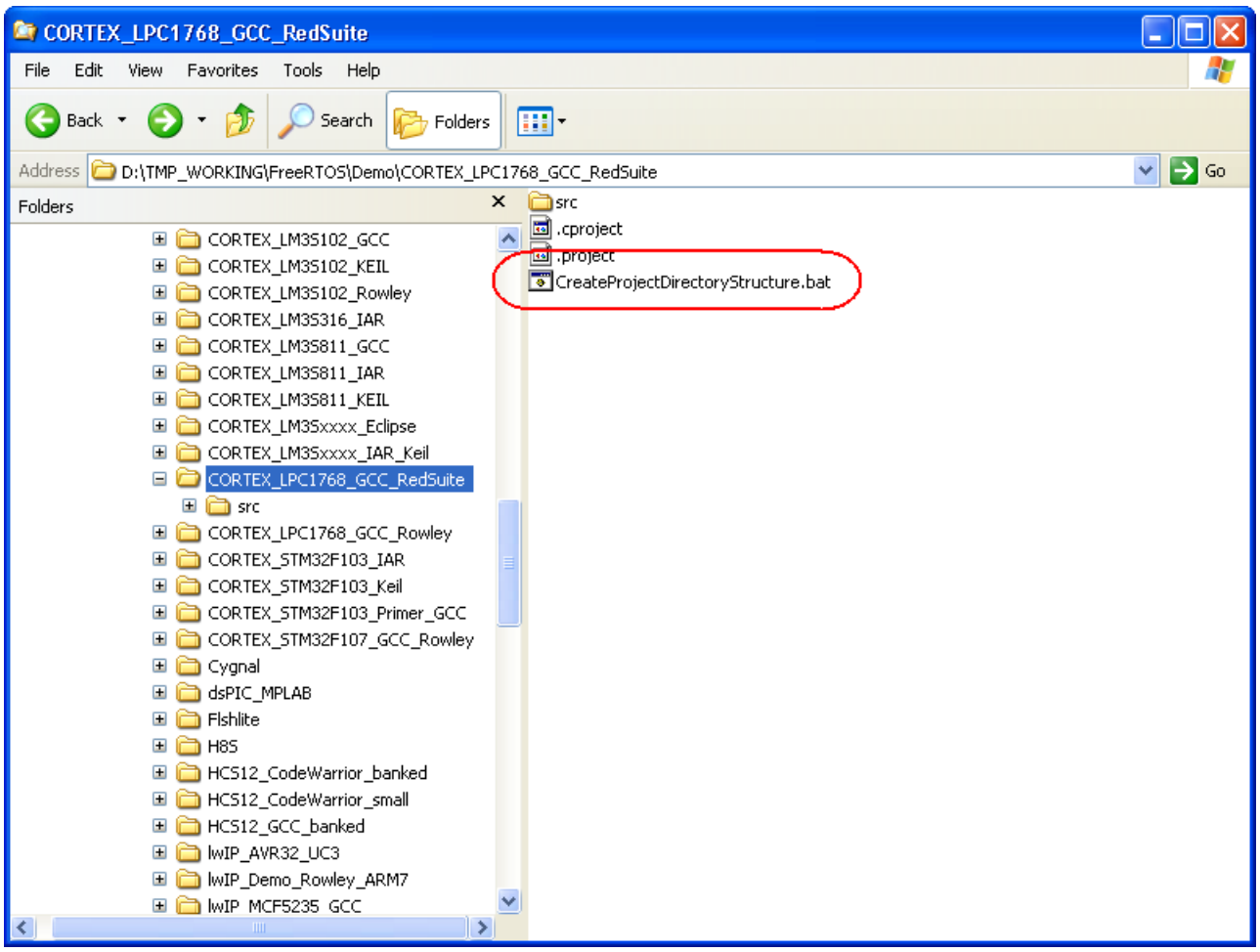


Fig 1. Run .bat file

Result after running the .bat file should be like this:

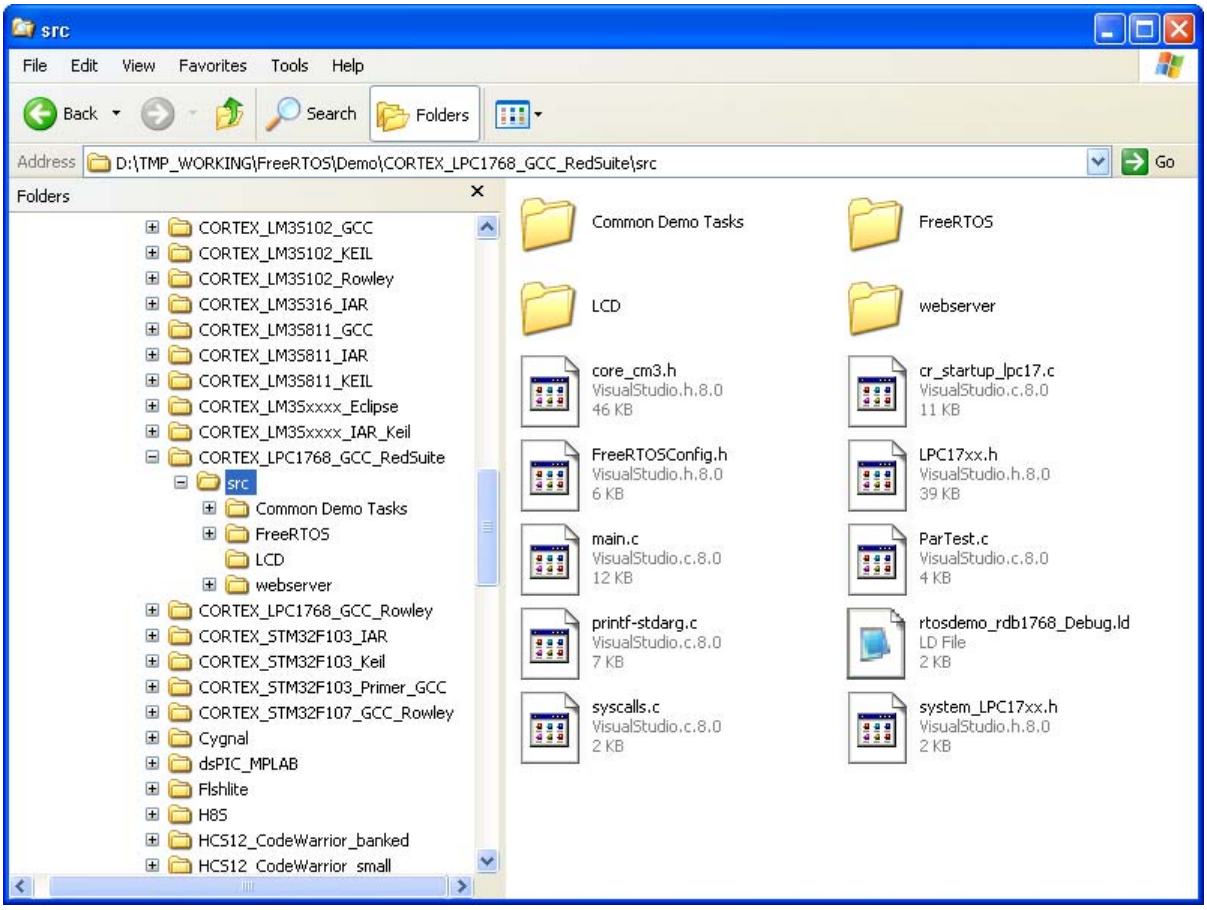
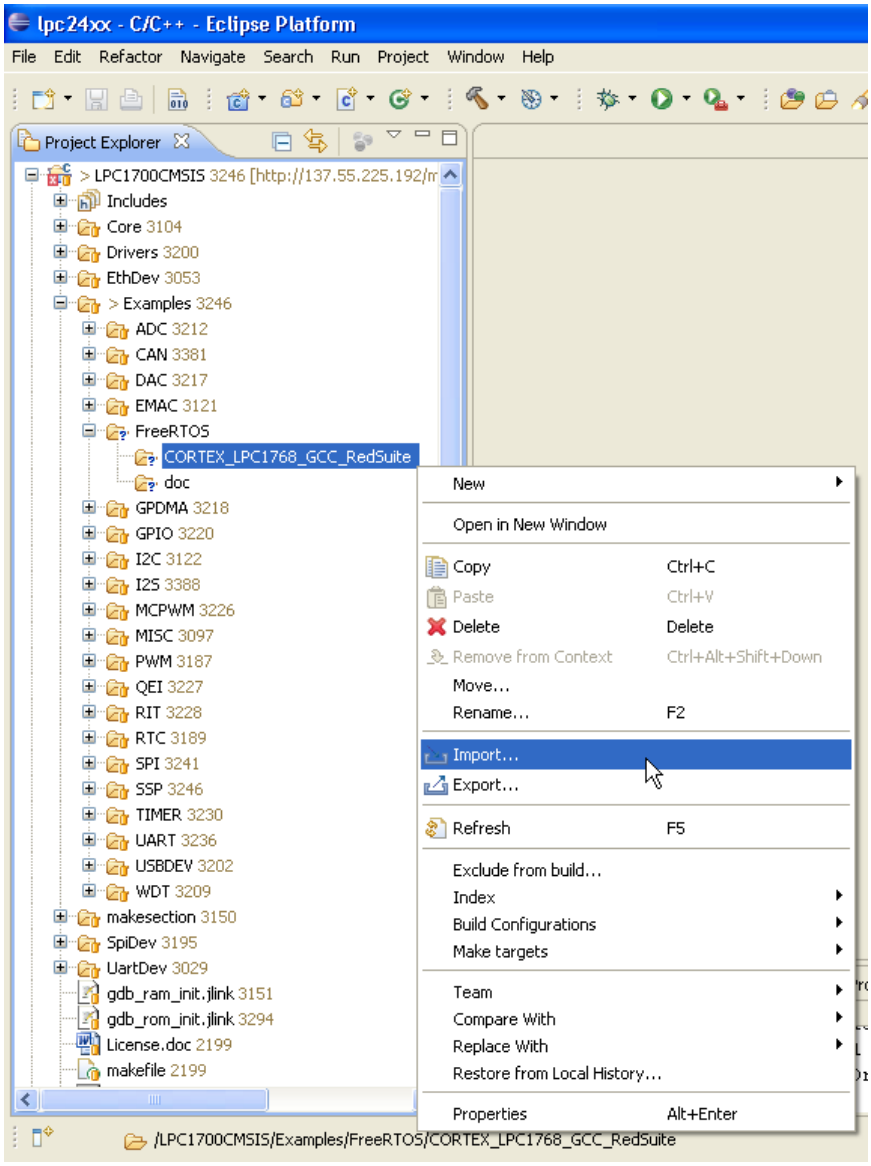
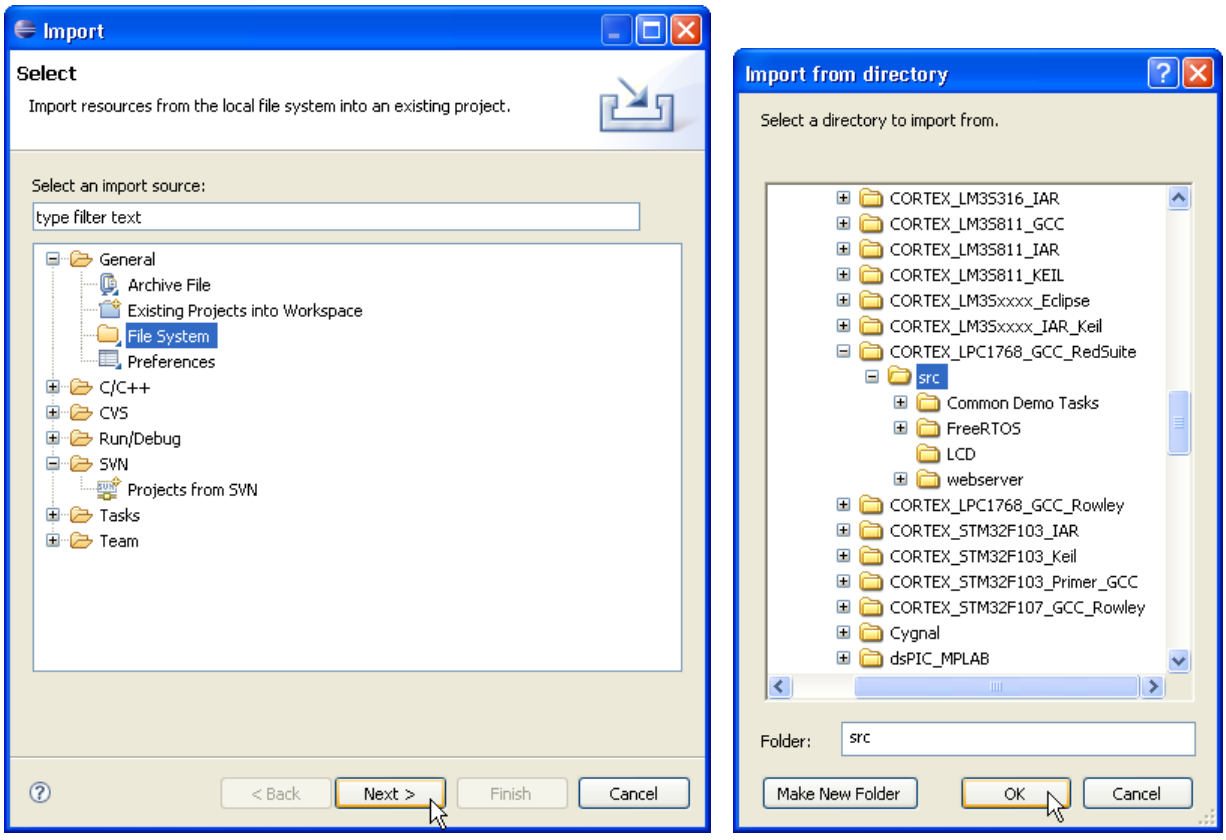


Fig 2. Result after run .bat file

- Create new folder and sub-folder under ".\Examples" at root of project in Eclipse, in this case, it would be ".\Example\FreeRTOS\CORTEX_LPC1768_GCC_RedSuite".
- Import all files and folders inside ".\FreeRTOS\Demo\CORTEX_LPC1768_GCC_RedSuite" (FreeRTOS source file after extracted) into ".\Example\FreeRTOS\CORTEX_LPC1768_GCC_RedSuite" folder of project in Eclipse.





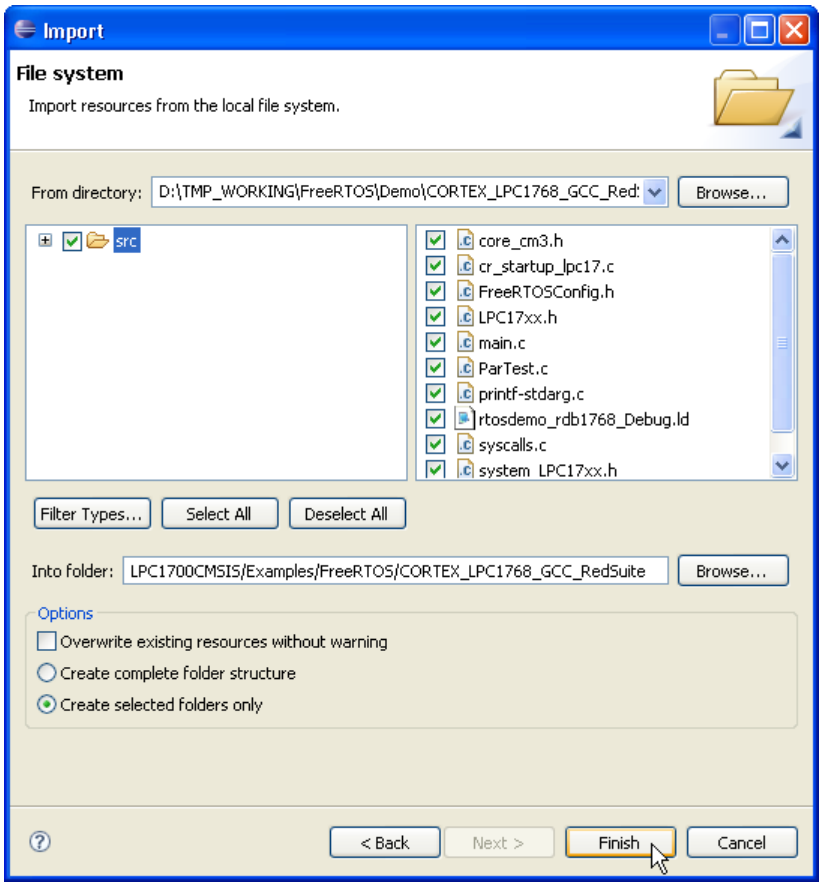


Fig 3. Import source file

The result should be like this:

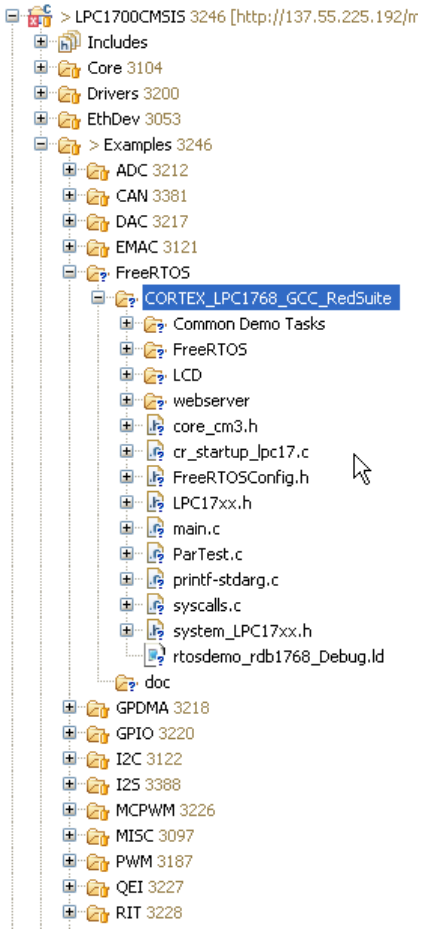


Fig 4. Result in project window after importing

- From this time, all files and folders mentioned are all files and folders inside ".\Example\FreeRTOS\CORTEX_LPC1768_GCC_RedSuite" folder – root of demo example folder.
- Re-name "Common Demo Tasks" folder to "CommonDemoTasks" – remove 'space'.
- Copy "makefile" (comes together with this package) to root of demo and modify this file.
- Create new folder named "bin" to the root of demo.
- Delete "LCD" folder (will be replaced by a new one).
- Make a target.

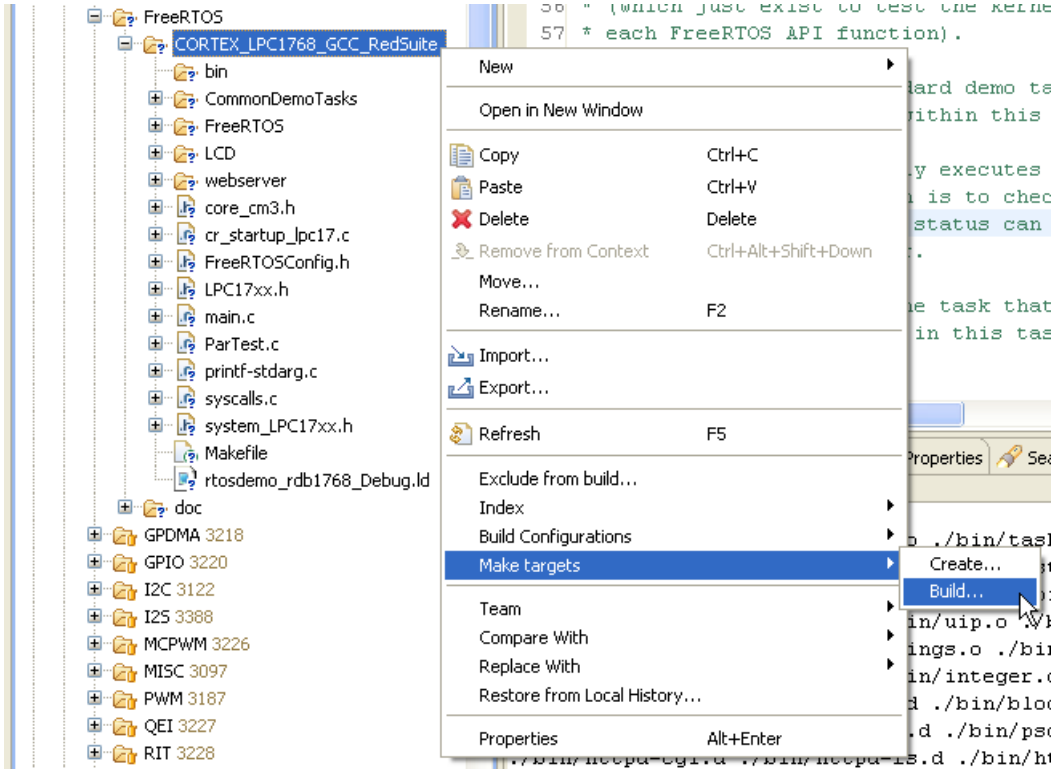


Fig 5. Make a target

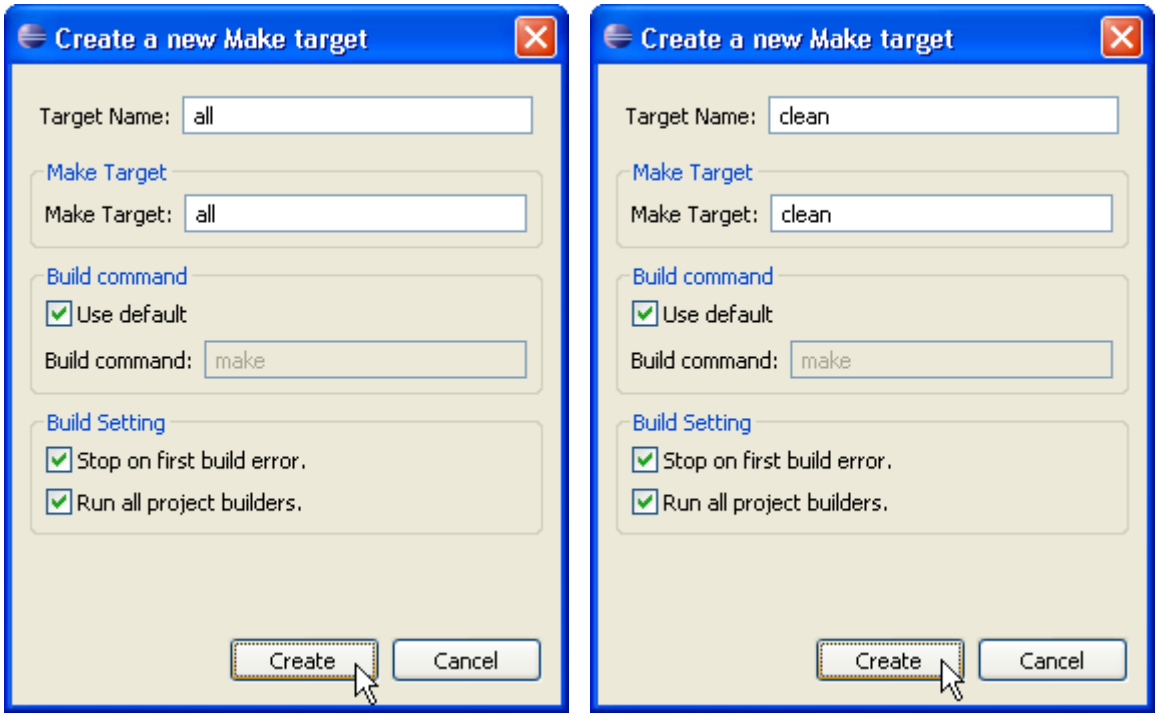


Fig 6. Make a target (clean and all)

3.2 Source file modification

3.2.1 GLCD

In order to re-target the LCD control (designed to work with LCD on RDB1768 from CodeRed) to match the GLCD on MCB1700 board, a new GLCD driver must be used to replace the old one.

This GLCD driver comes together with Keil SD File Demo on LPC1768 and can be downloaded from Keil. Only two files: “GLCD.h”, “GLCD.c” will be copied in to GLCD folder above.

Change ASCII_Table[] variable to const type to save memory space in RAM (in “.\GLCD\font.h” file).

3.2.2 LEDs

The GPIO used to drive the LEDs needs to be changed in “ParTest.c” file (Re-target to P2.2 → P2.6)

3.2.3 Using LPC1700CMSIS CM3 Core

The essential files in CM3 core reside at “.\Core\CM3”:

Table 1. CM3 core files

CM3 Core File	Function
.\Core\CM3\cm3_core.h	CMSIS Cortex-M3 Core Peripheral Access Layer Header File
.\Core\CM3\cm3_core.c	CMSIS Cortex-M3 Core Peripheral Access Layer Source File
.\Core\CM3\LPC17xx.h	CMSIS Cortex-M3 Core Peripheral Access Layer Header File for NXP LPC17xx Device Series
.\Core\CM3\system_LPC17xx.h	CMSIS Cortex-M3 Device Peripheral Access Layer Header File for the NXP LPC17xx Device Series
.\Core\CM3\system_LPC17xx.c	CMSIS Cortex-M3 Device Peripheral Access Layer Source File for the NXP LPC17xx Device Series

Delete these files inside root of demo since they are in CM3 core folder already: “core_cm3.h”, “LPC17xx.h” and “system_LPC17xx.h”.

Re-name “LPC17xx.h” at root of demo to “LPC17xx_FreeRTOS_def.h” to avoid any conflict with “LPC17xx.h” in CMSIS CM3 core. “LPC17xx_FreeRTOS_def.h” expands some old macros that are used in some source files of this demo (such as “.\webserver\emac.c”, .

Modify “Makefile” at root of demo to include these object files in “.\Core\CM3” folder that will be built together.

Modify “Makefile” at root of demo to set path of included files folder as “.\Core\CM3”.

Note: Startup file in CM3 core is ignored since FreeRTOS uses file “cr_startup_lpc17.c” as its own startup file.

3.2.4 Using LPC1700CMSIS Firmware Library

Here's LPC1700CMSIS firmware library section

Table 2. LPC1700CMSIS firmware library

LPC1700CMSIS fwlib part	Function
.\Drivers\include*.h	Driver Included files of LPC1700CMSIS fwlib.
.\Drivers\source*.c	Driver Source files folder of LPC1700CMSIS fwlib.
.\Drivers\library\DriversLPC17xxgnu.a [1]	Driver library file of LPC1700CMSIS fwlib.

[1] Archived after compiling all LPC1700CMSIS project.

Modify "Makefile" at root of demo to include all header files of fwllib in ".\Drivers\include" folder.

Modify "Makefile" at root of demo to include library file ".\Drivers\library\DriversLPC17xxgnu.a" that is required during linking object files.

3.2.5 Main

Comment-out this line following in "main.c" file (at the root of demo)

Fig 7. Comment-out this line.

- Modify source code in "main.c" file to include required header files.
- In this demo, UART driver of LPC1700CMSIS fwlib is used in lpc1700cmsis_fwlib_uart_demo() function.

Note:

In case of using driver of LPC1700CMSIS fwlib, SystemInit() function should be called instead of using prvSetupHardware() function of FreeRTOS. Since FreeRTOS initializes system clock in its own way that is not compliant with CMSIS convention, while all driver in LPC1700CMSIS fwlib are compliant with that rule.

3.2.6 Build and execute program

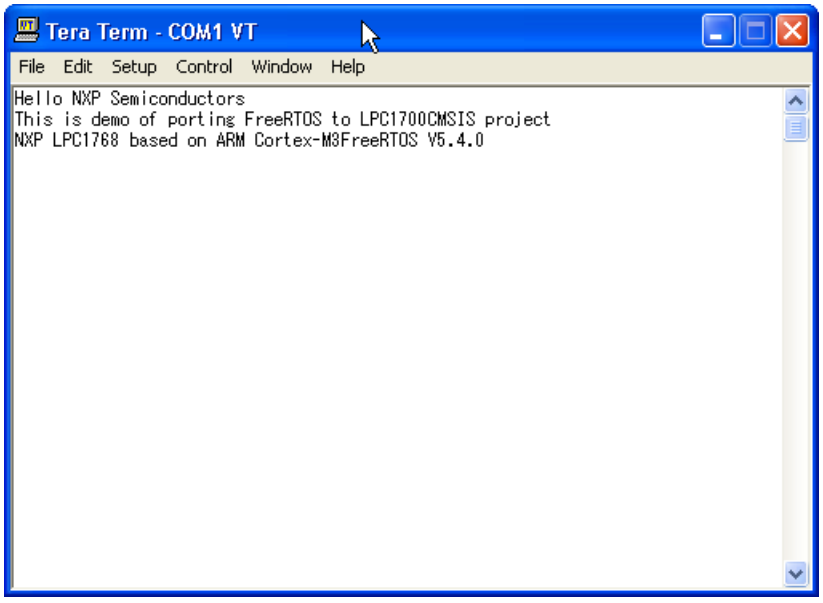
When everything is completed, all source files will be built into “Burn.hex” file when executing “make all” command; result should be like this after compiling:

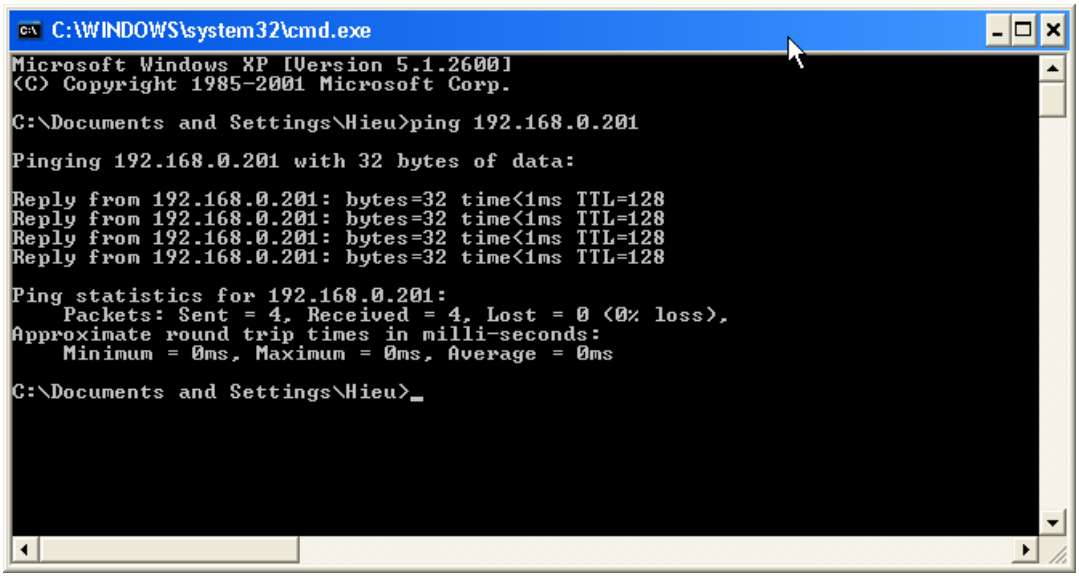
```
-MMD -MP -MF"bin/http-strings.d" -MT"bin/http-strings.d"    -o"bin/http-strings.o" "./webserver/http-strings
arm-none-eabi-gcc -nostdlib -mthumb -nostartfiles --no-gc-sections -Xlinker -Map=./bin/output.map -mcpu= -T
rtosdemo_rdb1768_Debug.ld -o"./bin/RTOSDemo.axf"  ./bin/port.o ./bin/list.o ./bin/tasks.o ./bin/queue.o
./bin/heap_2.o ./bin/main.o ./bin/BlockQ.o ./bin/integer.o ./bin/PollQ.o ./bin/semtest.o ./bin/GenQTest.o
./bin/QPeek.o ./bin/recmutex.o ./bin/GLCD.o ./bin/ParTest.o ./bin/flash.o ./bin/blocktim.o ./bin/printf-stda
./bin/cr_startup_lpc17.o ./bin/system_LPC17xx.o ./bin/syscalls.o  ./bin/uip_arp.o ./bin/psock.o ./bin/timer.
./bin/uip.o ./bin/uIP_Task.o ./bin/emac.o ./bin/httpd.o ./bin/httpd-cgi.o ./bin/httpd-fs.o ./bin/http-string
arm-none-eabi-objcopy -O ihex ./bin/RTOSDemo.axf ./bin/Burn.hex
```

Fig 8. Result after compiling

Using this “Burn.hex” file to burn chip.

There’s a pre-compiled “Burn.hex” file in “.\bin” folder as default release state of this package. The default IP address in this demo set to 192.168.0.201. The result is shown as follows:





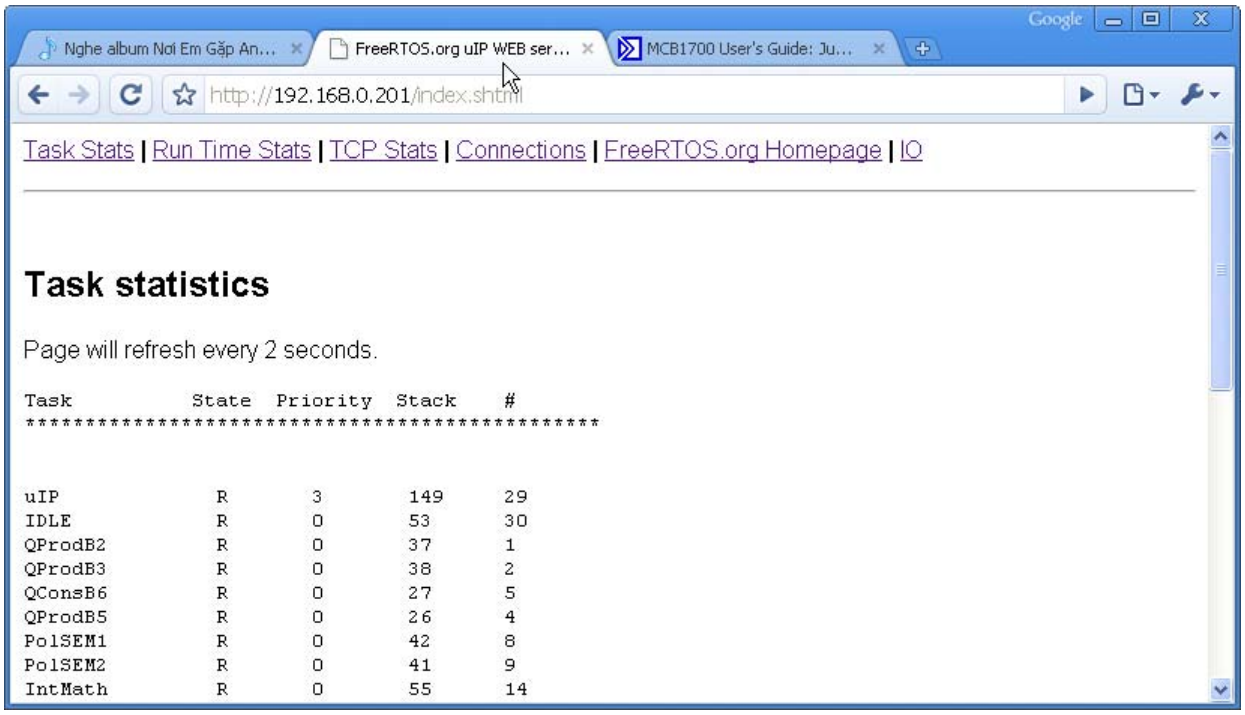


Fig 9. "Ping" command and working with web browser

4. Legal information

4.1 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.

4.2 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.

4.3 Trademarks

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

5. Contents

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

Date of release: 28 August 2009
Document identifier: AN10864_1