智云服
ZHIYUNFU

OpenCPU GCC
Installation Guide

GSM/GPRS Module Series

Rev. OpenCPU_GCC_Installation_Guide_V1.1

Date: 2017-07-20

About the Document

History

Revision	Date	Author	Description
1.0	2017-07-01	Chunmao Li	Initial
1.1	2017-07-10	Chunmao Li	Updated the description of how to get the compiler installation package

Contents

Figure Index

1 Introduction

This document mainly introduces how to set up GCC compiler environment for Windows, and how to compile App in OpenCPU SDK using GCC.

2 Installation

OpenCPU uses “Sourcery CodeBench Lite” as GCC compiler. Developers need to get the setup package from ZF Technical support .

2.1. System Requirements

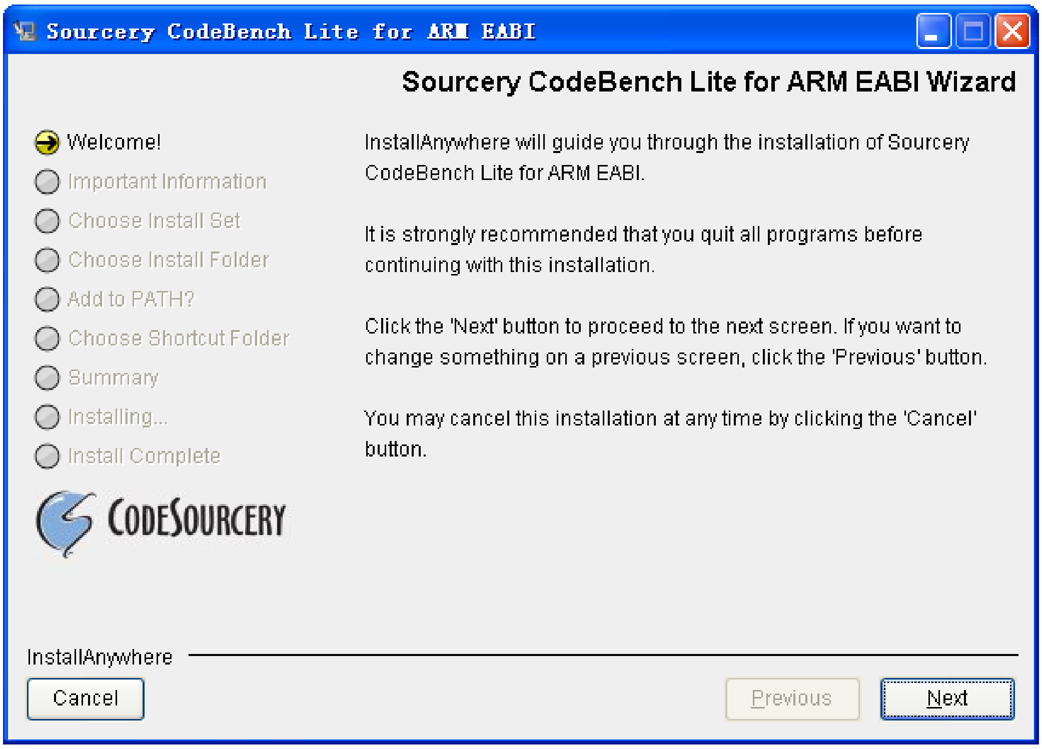
This version of Sourcery CodeBench supports the following host operating systems and architectures:

- Microsoft Windows XP (SP1 or later)
- Windows Vista
- Windows 7 systems using IA32, AMD64, and Intel 64 processors

In order to install and use Sourcery CodeBench Lite, you must have at least 512MB of available memory.

2.2. Install GCC

Running the Installer - Double click the GCC installer to start to install GCC. After the installer starts, follow the on-screen dialogs to install Sourcery CodeBench Lite. The installer is intended to be self-explanatory and on most pages the defaults are appropriate.



Sourcery CodeBench Lite for ARM EABI

Sourcery CodeBench Lite for ARM EABI Wizard

- Welcome!
- Important Information
- Choose Install Set
- Choose Install Folder
- Add to PATH?
- Choose Shortcut Folder
- Summary
- Installing...
- Install Complete

InstallAnywhere will guide you through the installation of Sourcery CodeBench Lite for ARM EABI.

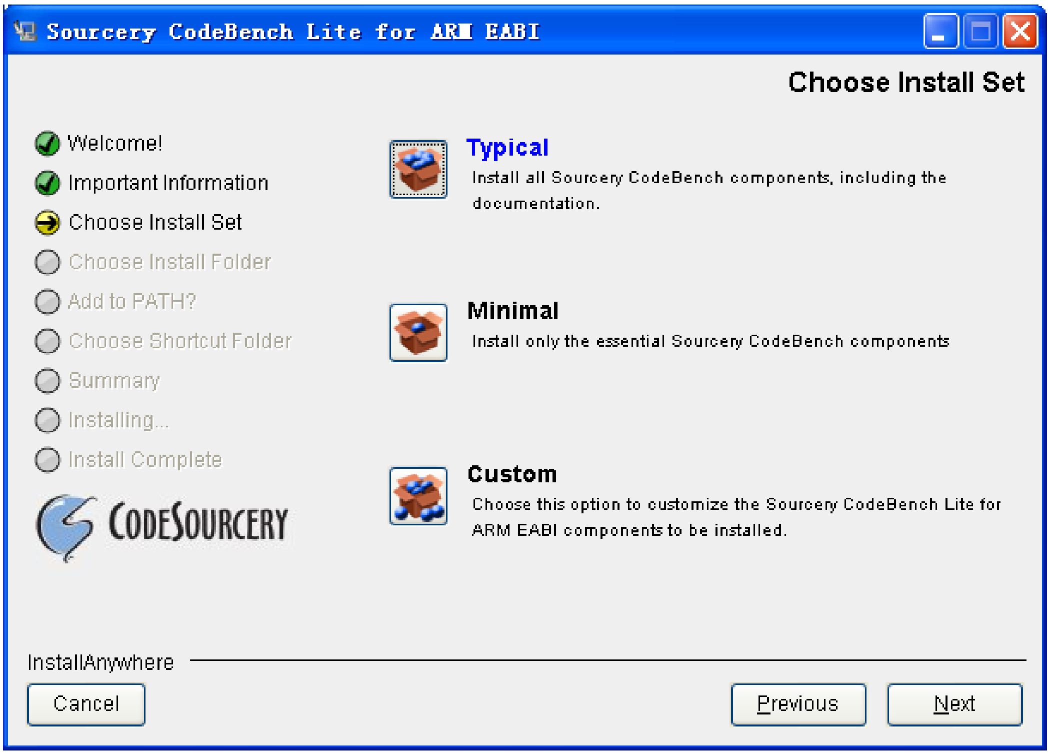
It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

Figure 1: Running the Installer

Choose Install Set - Select the typical install set.



Sourcery CodeBench Lite for ARM EABI

Choose Install Set

- Welcome!
- Important Information
- Choose Install Set
- Choose Install Folder
- Add to PATH?
- Choose Shortcut Folder
- Summary
- Installing...
- Install Complete

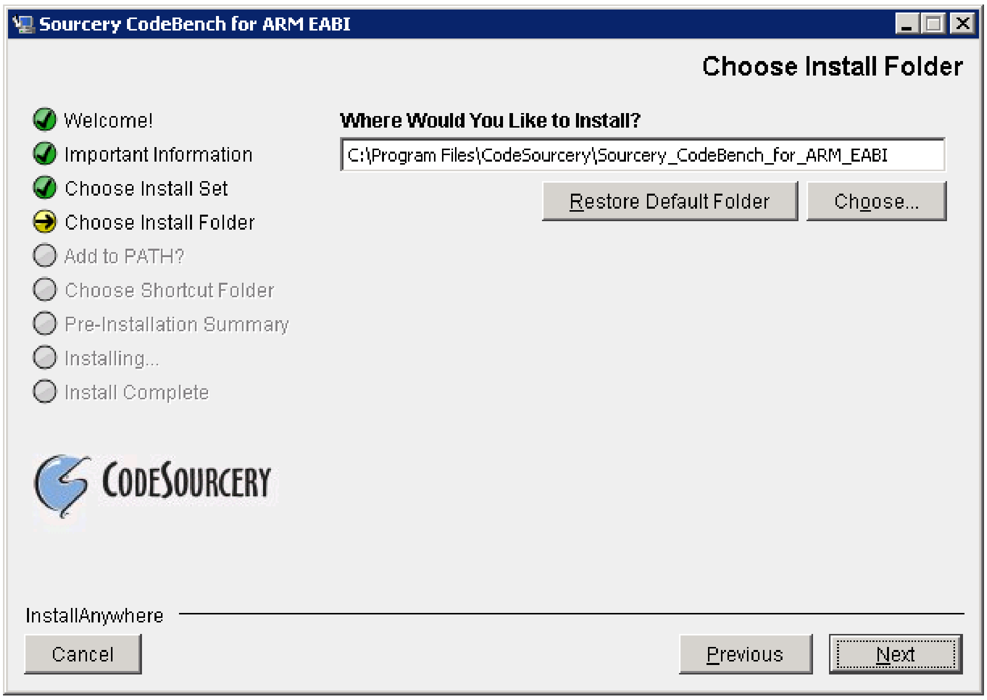
Typical
Install all Sourcery CodeBench components, including the documentation.

Minimal
Install only the essential Sourcery CodeBench components

Custom
Choose this option to customize the Sourcery CodeBench Lite for ARM EABI components to be installed.

Figure 2: Choose Install Set

Choose Install Folder - You may want to change the install directory pathname and customize the shortcut installation.



Sourcery CodeBench for ARM EABI

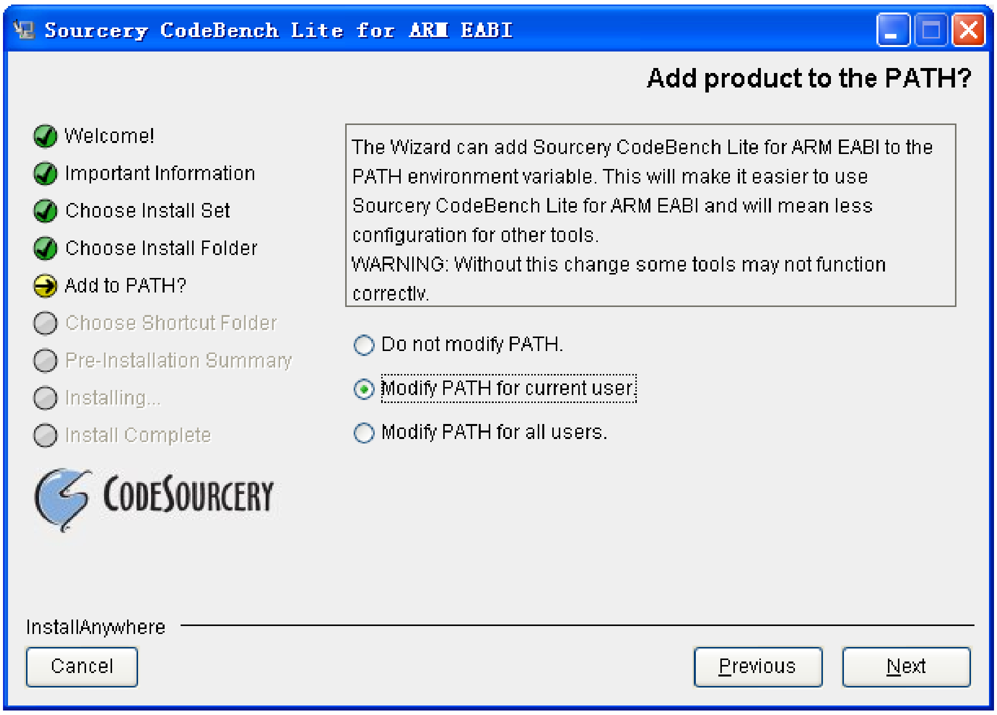
Choose Install Folder

Where Would You Like to Install?

C:\Program Files\CodeSourcery\Sourcery_CodeBench_for_ARM_EABI

Figure 3: Choose Install Folder

Add Product to the PATH - Keep the default choice to allow the installer to set environment variable.



Sourcery CodeBench Lite for ARM EABI

Add product to the PATH?

The Wizard can add Sourcery CodeBench Lite for ARM EABI to the PATH environment variable. This will make it easier to use Sourcery CodeBench Lite for ARM EABI and will mean less configuration for other tools.
WARNING: Without this change some tools may not function correctly.

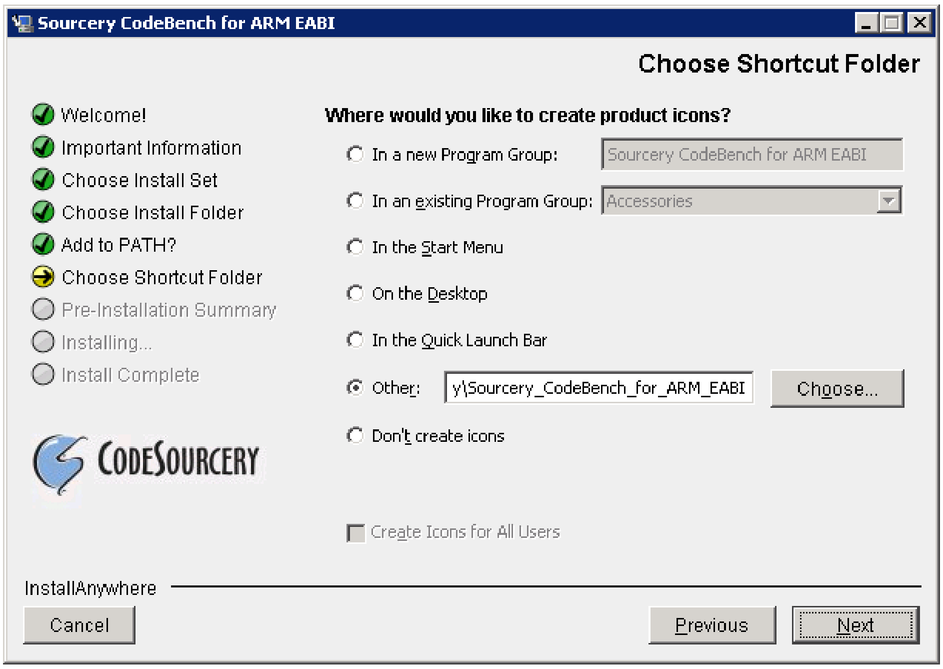
○ Do not modify PATH.

● Modify PATH for current user.

○ Modify PATH for all users.

Figure 4: Add Product to the PATH

Choose Shortcut Folder - You can customize where the installer creates shortcuts for quick access to Sourcery CodeBench Lite. When the installer has finished, it asks if you want to launch a viewer for the Getting Started guide. Finally, the installer displays a summary screen to confirm a successful install before it exits.



Sourcery CodeBench for ARM EABI

Choose Shortcut Folder

- ✓ Welcome!
- ✓ Important Information
- ✓ Choose Install Set
- ✓ Choose Install Folder
- ✓ Add to PATH?
- ➔ Choose Shortcut Folder
- ○ Pre-Installation Summary
- ○ Installing...
- ○ Install Complete

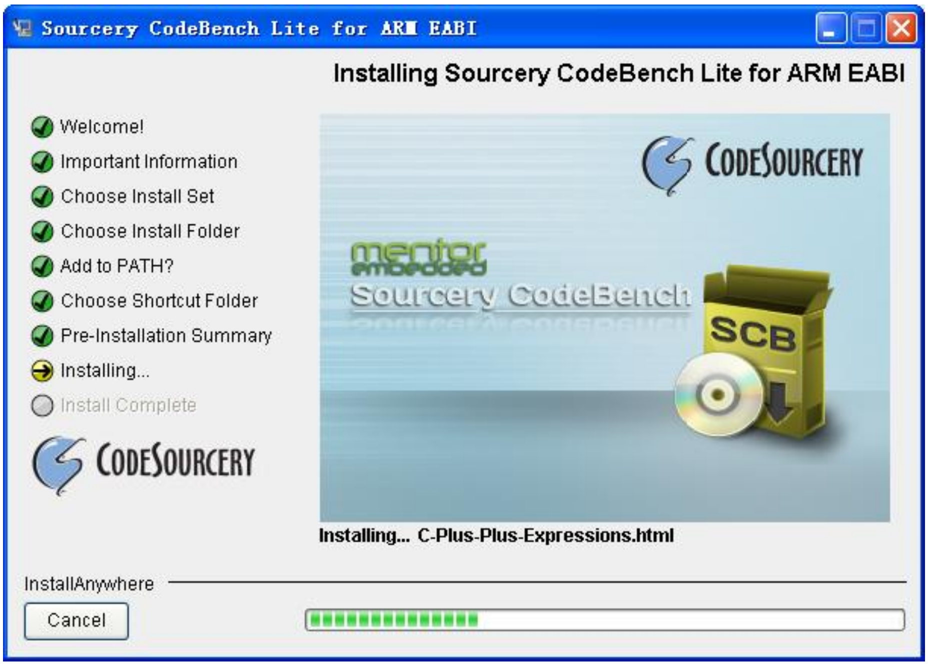
Where would you like to create product icons?

- ○ In a new Program Group: Sourcery CodeBench for ARM EABI
- ○ In an existing Program Group: Accessories
- ○ In the Start Menu
- ○ On the Desktop
- ○ In the Quick Launch Bar
- ● Other: y\Sourcery_CodeBench_for_ARM_EABI
- ○ Don't create icons

□ Create Icons for All Users

Figure 5: Choose Shortcut Folder

Click “Next” to Complete Installation.



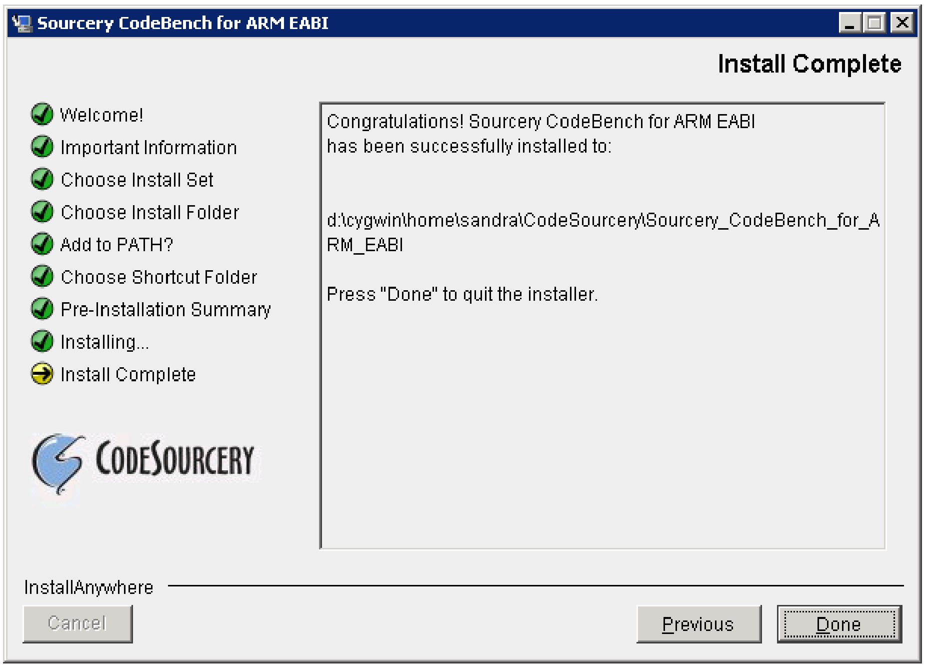
Sourcery CodeBench Lite for ARM EABI

Installing Sourcery CodeBench Lite for ARM EABI

- ✓ Welcome!
- ✓ Important Information
- ✓ Choose Install Set
- ✓ Choose Install Folder
- ✓ Add to PATH?
- ✓ Choose Shortcut Folder
- ✓ Pre-Installation Summary
- ➔ Installing...
- ○ Install Complete

Installing... C-Plus-Plus-Expressions.html

Figure 6: Installing



Congratulations! Sourcery CodeBench for ARM EABI has been successfully installed to:

d:\cygwin\home\sandra\CodeSourcery\Sourcery_CodeBench_for_ARM_EABI

Press "Done" to quit the installer.

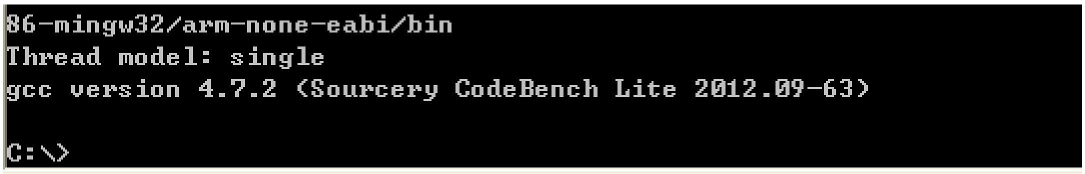
Figure 7: Install Complete

2.3. Verify Installation

By default, the GCC installer sets up the environment variable. You can verify whether your PATH is set up correctly by starting a new cmd.exe shell and running:

```
arm-none-eabi-gcc –v
```

Verify whether the last line of the output contains “**Sourcery CodeBench Lite 2012.09-63**”.



```
86-mingw32/arm-none-eabi/bin
Thread model: single
gcc version 4.7.2 (Sourcery CodeBench Lite 2012.09-63)

C:\>
```

If not, you can manually set up the environment using the following command in a **cmd.exe** shell:

```
SET PATH=%PATH%;”installdir\bin"
```

For example:

```
SET PATH=%PATH%;”D:\Program Files\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_EABI\bin”
```

And then verify again:

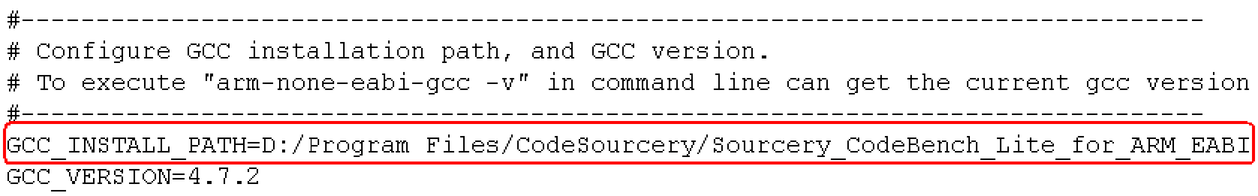
```
arm-none-eabi-gcc -v
```

3 Configuration

Before compiling App with Sourcery CodeBench Lite, you have to configure the installation path and the GCC environment library path in OpenCPU SDK.

3.1. Configure the Installation Path

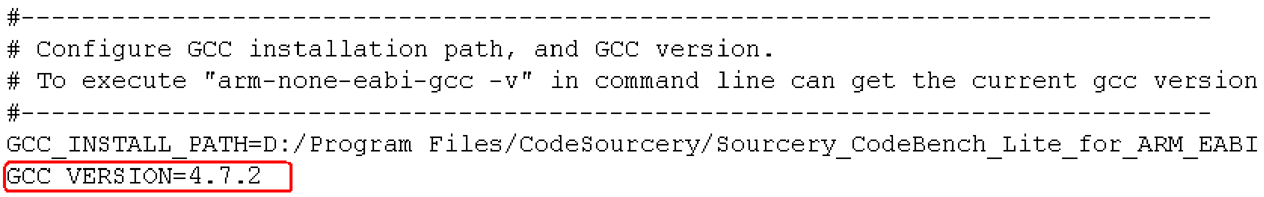
Open \SDK\make\gcc\gcc_makefile, and change the value of GCC_INSTALL_PATH accordingly.



```
#-------------------------------------------------------------------------------
# Configure GCC installation path, and GCC version.
# To execute "arm-none-eabi-gcc -v" in command line can get the current gcc version
#-------------------------------------------------------------------------------
GCC_INSTALL_PATH=D:/Program Files/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_EABI
GCC_VERSION=4.7.2
```

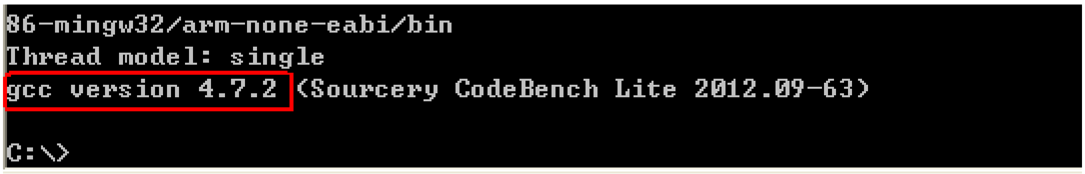
3.2. Configure GCC Version

If you do not install the version “arm-2012.09-63-arm-none-eabi” which is built in GCC version 4.7.2 (use “arm-none-eabi-gcc –v” to check), you need to change the value of GCC_VERSION accordingly in the file \SDK\make\gcc\gcc_makefile.



```
#-------------------------------------------------------------------------------
# Configure GCC installation path, and GCC version.
# To execute "arm-none-eabi-gcc -v" in command line can get the current gcc version
#-------------------------------------------------------------------------------
GCC_INSTALL_PATH=D:/Program Files/CodeSourcery/Sourcery_CodeBench_Lite_for_ARM_EABI
GCC_VERSION=4.7.2
```

Here, “4.7.2” indicates the GCC version, which should be corresponding to the version of the currently installed GCC (you can check the current GCC version using “arm-none-eabi-gcc –v” in command line). Then the compiler can search the correct path during compiling.



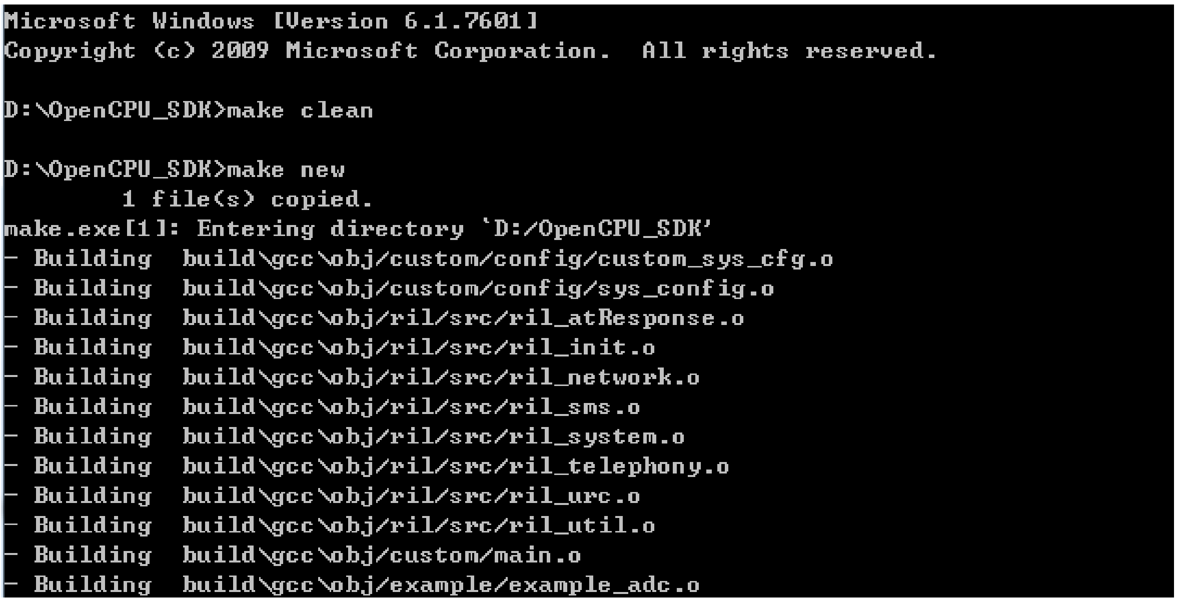
```
86-mingw32/arm-none-eabi/bin
Thread model: single
gcc version 4.7.2 (Sourcery CodeBench Lite 2012.09-63)

C:\>
```

4 Compile

Now, the GCC compiling environment is set up successfully. In OpenCPU, compiling commands are executed in command line. The compiling and clean commands are defined as below.

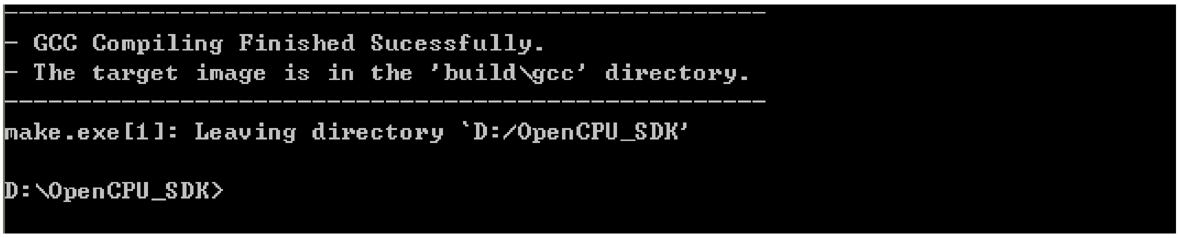
```
make clean
make new
```

The compiling and clean commands need to be executed in the root directory of SDK.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

D:\OpenCPU_SDK>make clean

D:\OpenCPU_SDK>make new
        1 file(s) copied.
make.exe[1]: Entering directory `D:/OpenCPU_SDK'
- Building  build\gcc\obj/custom/config/custom_sys_cfg.o
- Building  build\gcc\obj/custom/config/sys_config.o
- Building  build\gcc\obj/ril/src/ril_atResponse.o
- Building  build\gcc\obj/ril/src/ril_init.o
- Building  build\gcc\obj/ril/src/ril_network.o
- Building  build\gcc\obj/ril/src/ril_sms.o
- Building  build\gcc\obj/ril/src/ril_system.o
- Building  build\gcc\obj/ril/src/ril_telephony.o
- Building  build\gcc\obj/ril/src/ril_urc.o
- Building  build\gcc\obj/ril/src/ril_util.o
- Building  build\gcc\obj/custom/main.o
- Building  build\gcc\obj/example/example_adc.o
```

After successfully compiling, you will see the output shown as below.



```
----------------------------------------------------
- GCC Compiling Finished Sucessfully.
- The target image is in the 'build\gcc' directory.
----------------------------------------------------
make.exe[1]: Leaving directory `D:/OpenCPU_SDK'

D:\OpenCPU_SDK>
```

Please refer to “*OpenCPU User Guide*” document for more information about development environment.