# OpenCPU GCC Installation Guide

**GSM/GPRS Module Series**

Rev. OpenCPU_GCC_Installation_Guide_V1.1

Date: 2015-06-04

**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Mail: info@quectel.com

**Or our local office, for more information, please visit:**
http://www.quectel.com/support/salesupport.aspx

**For technical support, to report documentation errors, please visit:**
http://www.quectel.com/support/techsupport.aspx
Or Email: Support@quectel.com

## GENERAL NOTES

QUECTEL OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

## COPYRIGHT

# About the Document

## History

| Revision | Date | Author | Description |
|---|---|---|---|
| 1.0 | 2013-09-23 | Stanley YONG | Initial |
| 1.1 | 2015-06-04 | Stanley YONG | 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 Quectel Technical support (support@quectel.com).

## 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.

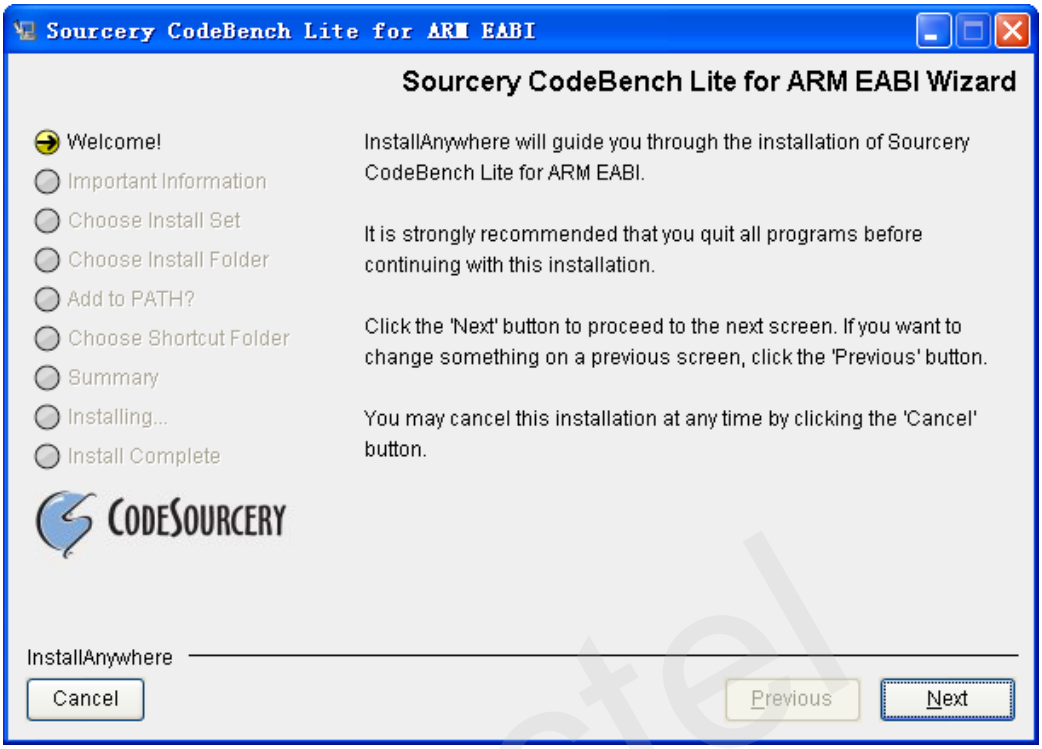


**Figure 1: Running the Installer**

**Choose Install Set** - Select the typical install set.

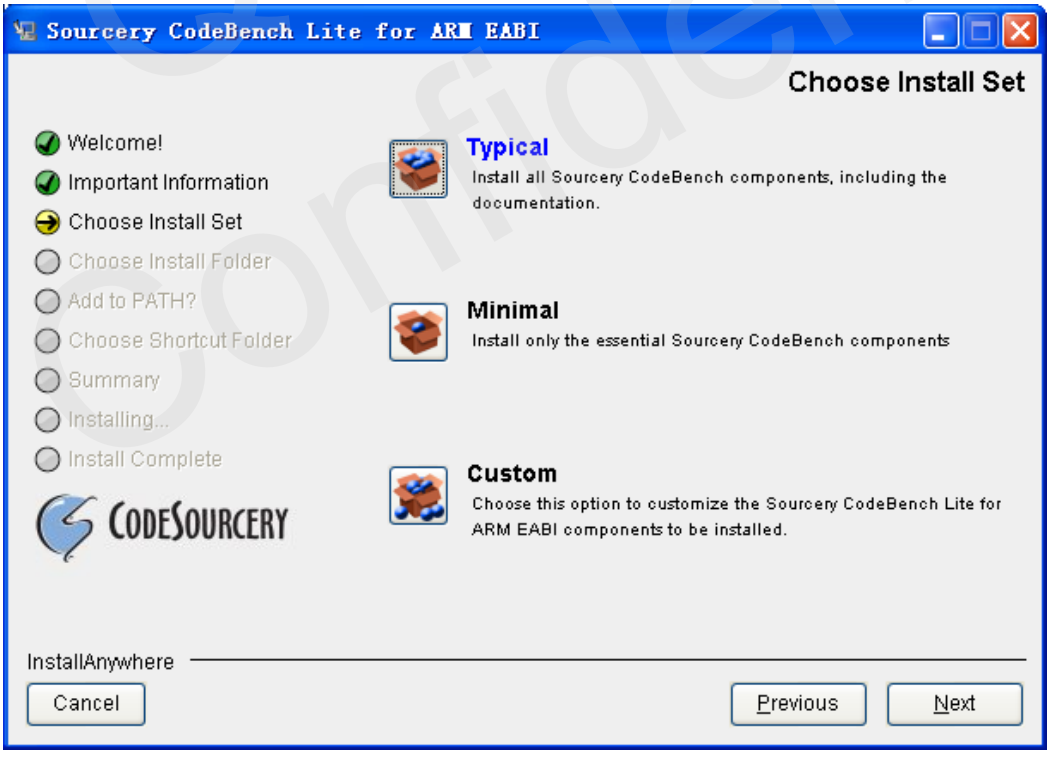


**Figure 2: Choose Install Set**

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

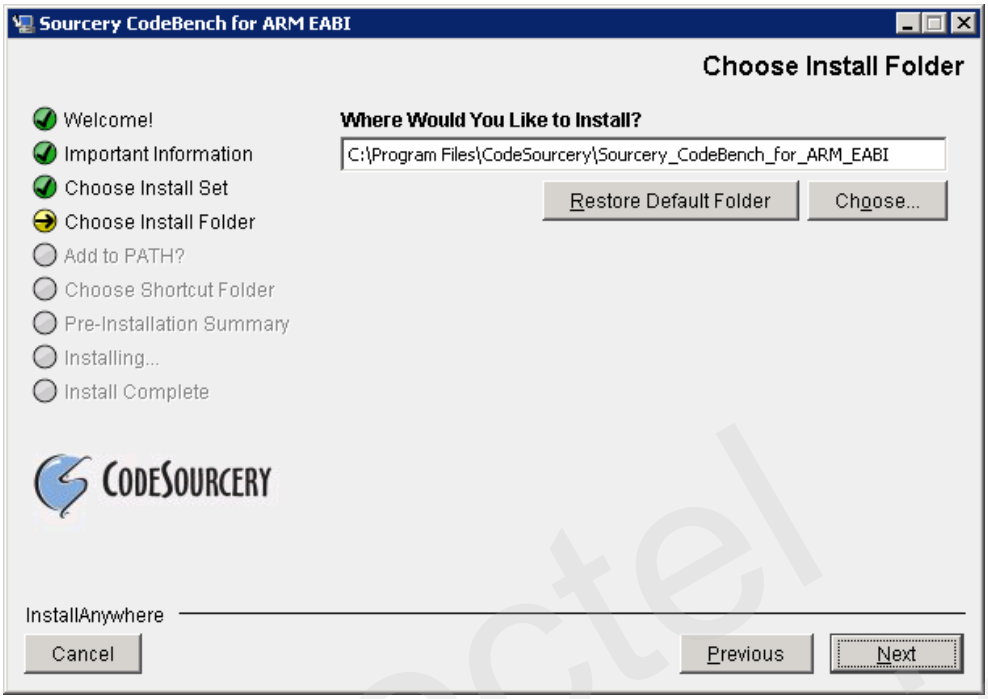


**Figure 3: Choose Install Folder**

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

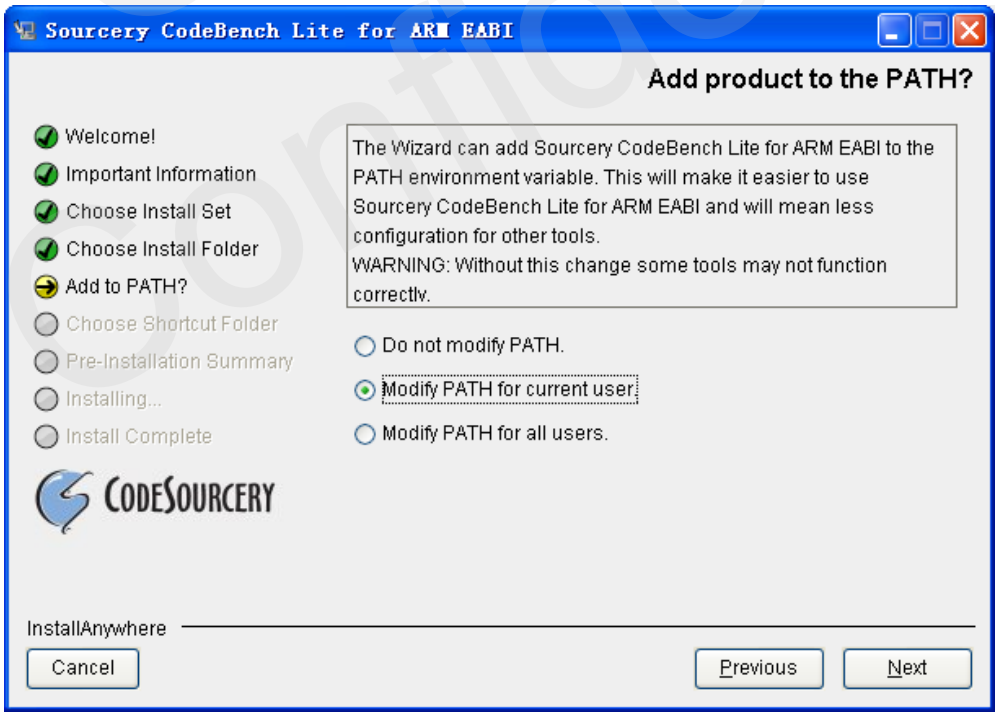


**Figure 4: Add Product to the PATH**

**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.

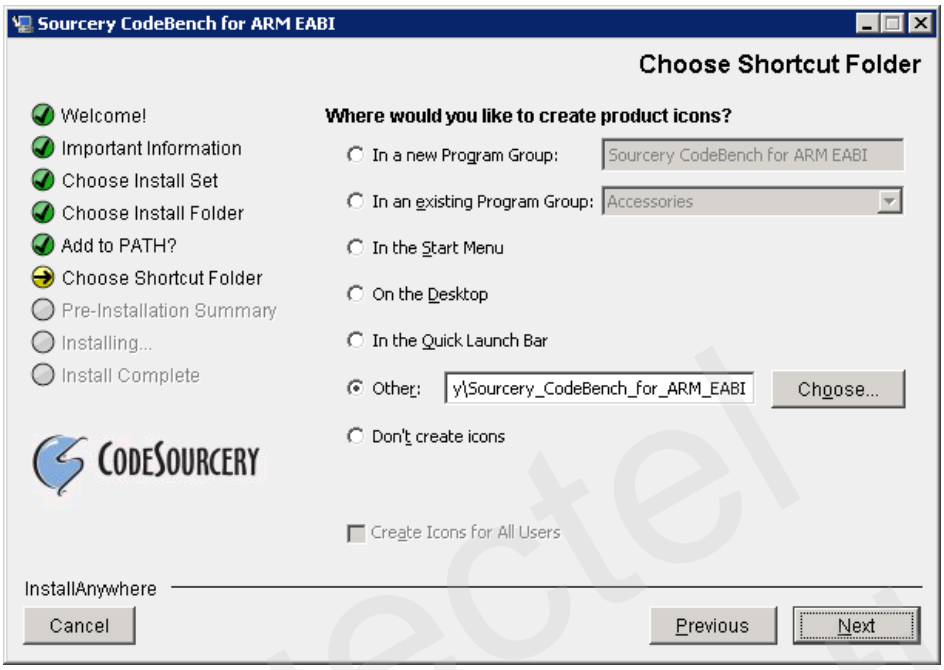


**Figure 5: Choose Shortcut Folder**

Click “Next” to Complete Installation.

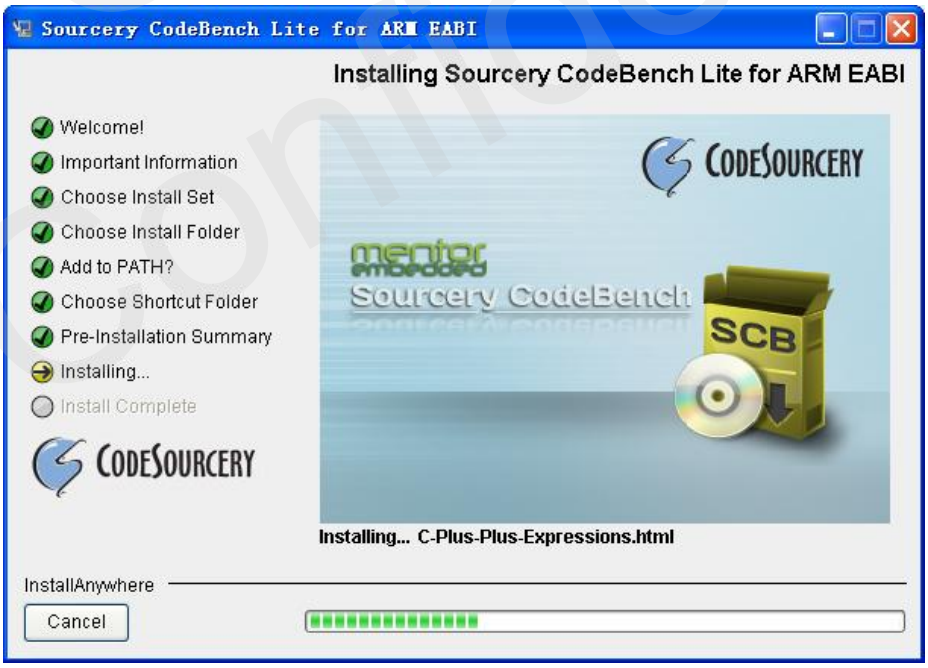


**Figure 6: Installing**

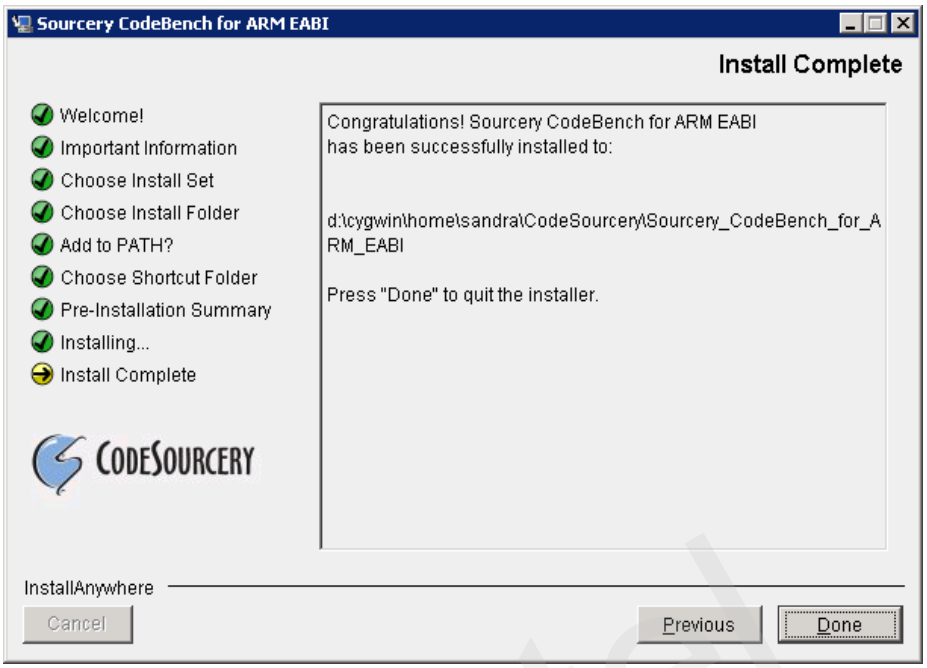


**Figure 7: Install Complete**

## 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.