BG96-QuecOpen
Application Note

LTE Module Series

Rev. BG96-QuecOpen_Application_Note_V1.0

Date: 2018-05-02

Status: Preliminary

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

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE 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	2018-05-02	Hyman DING	Initial

Contents

About the Document **2**
Contents **3**
Table Index **4**
Figure Index **5**

1 Introduction **6**

2 BG96-QuecOpen Solution Overview **7**
2.1. General Overview 7
2.2. BG96-QuecOpen Architecture 7

3 Setup Compiling Environment **9**
3.1. Setup Compiling Environment for TX2.0 9
3.1.1. Download and Install ARM Compiler Tool 9
3.1.2. Download and Install Cygwin 16
3.2. Setup Compiling Environment for TX3.0.1 20
3.2.1. Download LLVM 20
3.2.2. Download and Install Cygwin 22
3.2.3. Download and Install Python 22

4 Build QuecOpen Application **26**
4.1. QuecOpen SDK Package based on TX2.0 26
4.1.1. SDK Package Structure 26
4.1.2. Build QuecOpen User Application 27
4.2. QuecOpen SDK Package based on TX3.0.1 29
4.2.1. SDK Package Structure 29
4.2.2. Build QuecOpen User Application 29

5 Run QuecOpen Application **32**

6 Update QuecOpen Application **33**

7 Appendix A References **34**

Table Index

TABLE 1: COMPILING ENVIRONMENT REQUIREMENT FOR TX2.0 9
TABLE 2: COMPILING ENVIRONMENT REQUIREMENT FOR TX3.0.1 9
TABLE 3: DESCRIPTION OF BG96-QUECOPEN SDK PACKAGE DIRECTORIES (TX2.0) 26
TABLE 4: DESCRIPTION OF BG96-QUECOPEN SDK PACKAGE DIRECTORIES (TX3.0.1) 29
TABLE 5: RELATED DOCUMENTS 34
TABLE 6: TERMS AND ABBREVIATIONS 34

Figure Index

FIGURE 1: ARCHITECTURE OF BG96-QUECOPEN 8
FIGURE 2: “DOWNLOADS” AND “DEVELOPMENT TOOLS” PAGES 10
FIGURE 3: CLICK “DS-5 DEVELOPMENT STUDIO” 11
FIGURE 4: DOWNLOAD THE CORRESPONDING TOOL 11
FIGURE 5: CONFIRMATION OF DETAILS 12
FIGURE 6: ARM COMPILER 5 SETUP 13
FIGURE 7: END-USER LICENSE AGREEMENT 13
FIGURE 8: CUSTOM SETUP 14
FIGURE 9: “SYSTEM PENDING REBOOT” WARNING 14
FIGURE 10: READY TO INSTALL ARM COMPILER 5 15
FIGURE 11: FINISH INSTALLATION OF ARM COMPILER TOOL 15
FIGURE 12: CYGWIN SETUP PROGRAM 16
FIGURE 13: CHOOSE INSTALLATION TYPE 17
FIGURE 14: CHOOSE INSTALLATION DIRECTORY AND PARAMETERS 17
FIGURE 15: SELECT LOCAL PACKAGE DIRECTORY 18
FIGURE 16: SELECT INTERNET CONNECTION TYPE 18
FIGURE 17: CHOOSE A DOWNLOAD SITE 19
FIGURE 18: PROCESS OF DOWNLOAD OR INSTALLATION 19
FIGURE 19: CREATE ICON AND COMPLETE INSTALLATION 20
FIGURE 20: LLVM DOWNLOAD FOR WINDOWS 7 BUILD HOSTS 21
FIGURE 21: LLVM DOWNLOAD FOR LINUX BUILD HOSTS 22
FIGURE 22: PYTHON DOWNLOAD PAGE SCREENSHOT 23
FIGURE 23: PYTHON SETUP 23
FIGURE 24: SELECT INSTALLATION DIRECTORY 24
FIGURE 25: OPTIONS FOR CUSTOMIZATION 24
FIGURE 26: INSTALLING 25
FIGURE 27: INSTALLATION COMPLETED 25
FIGURE 28: FOLDER STRUCTURE OF BG96-QUECOPEN SDK PACKAGE (TX2.0) 26
FIGURE 29: FOLDER STRUCTURE OF BG96-QUECOPEN SDK PACKAGE (TX3.0.1) 29

1 Introduction

This document mainly introduces how to establish BG96-QuecOpen compiler environment in Windows and Linux Operating Systems, how to compile user application in BG96-QuecOpen SDK and how to run and update user application based on BG96-QuecOpen solution.

2 BG96-QuecOpen Solution Overview

2.1. General Overview

Quectel BG96-QuecOpen provides an infrastructure for applications to dynamically load modules that are built from the resident component of the application. The module is useful for the following scenarios:

- Total application code size exceeds the available memory
- New application modules need to be added after the core image is deployed
- Partial firmware updates are required

Each module is built independently with a common preamble structure attached in the binary. The preamble contains various details about the module, including:

- a single thread entry point
- stack size priority
- module ID
- callback thread stack size/priority, and so on.

2.2. BG96-QuecOpen Architecture

The following diagram shows the architecture of BG96-QuecOpen.

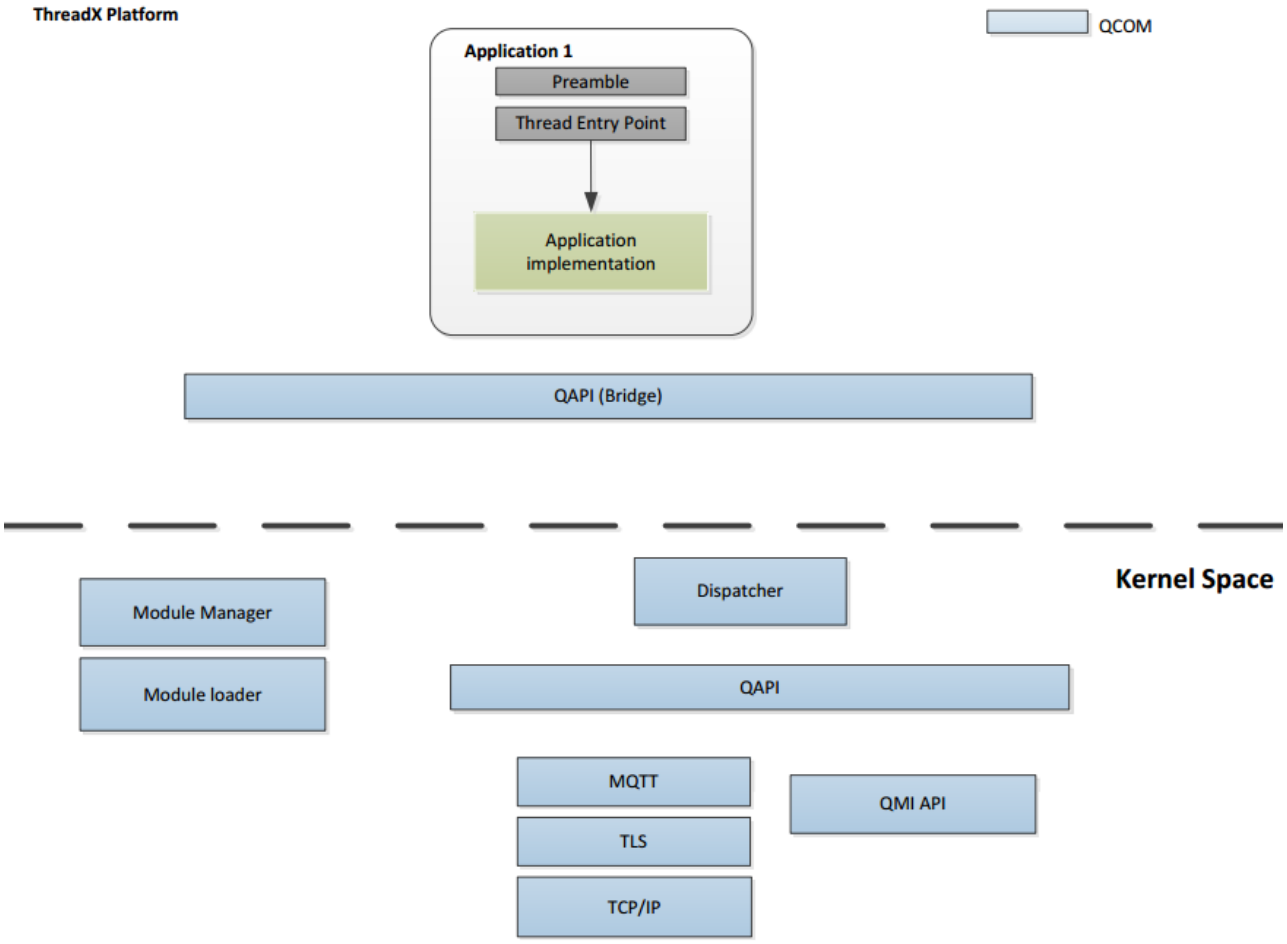

Figure 1: Architecture of BG96-QuecOpen

3 Setup Compiling Environment

Quectel BG96-QuecOpen includes two revisions based on different baseline of MDM9206. One is based on TX2.0 and the other is based on TX3.0.1. Compilation tools are different for different baselines.

While compiling BG96-QuecOpen user application, the host's operating system and compilation tools should meet the requirements shown below.

Table 1: Compiling Environment Requirement for TX2.0

Component	Source or Binary Only	Toolchain Required for Building Source	Cygwin	Supported Build Hosts
QuecOpen SDK	Source	ARM complier tools 5.05 (build 106)	Cygwin 2.8.0	Windows 7

Table 2: Compiling Environment Requirement for TX3.0.1

Component	Source or Binary Only	Toolchain Required for Building Source	Cygwin	Python	Supported Build Hosts
QuecOpen SDK	Source	LLVM 4.0.3	Cygwin 2.8.0	Python 2.7	Windows 7/ Linux

NOTE

Licensed users can download LLVM compiler through the Qualcomm ChipCode™ portal.

3.1. Setup Compiling Environment for TX2.0

3.1.1. Download and Install ARM Compiler Tool

The following mainly introduces how to download and install ARM complier tool in Windows build environment.

3.1.1.1. Download ARM Compiler Tool

Step 1: Create an account in the following page: https://silver.arm.com.

Step 2: Open the ARM complier tool download page: https://silver.arm.com/browse.

(1) Click "**Downloads**" → "**Development Tools**" → "**DS-5 Development Studio**", as illustrated below:

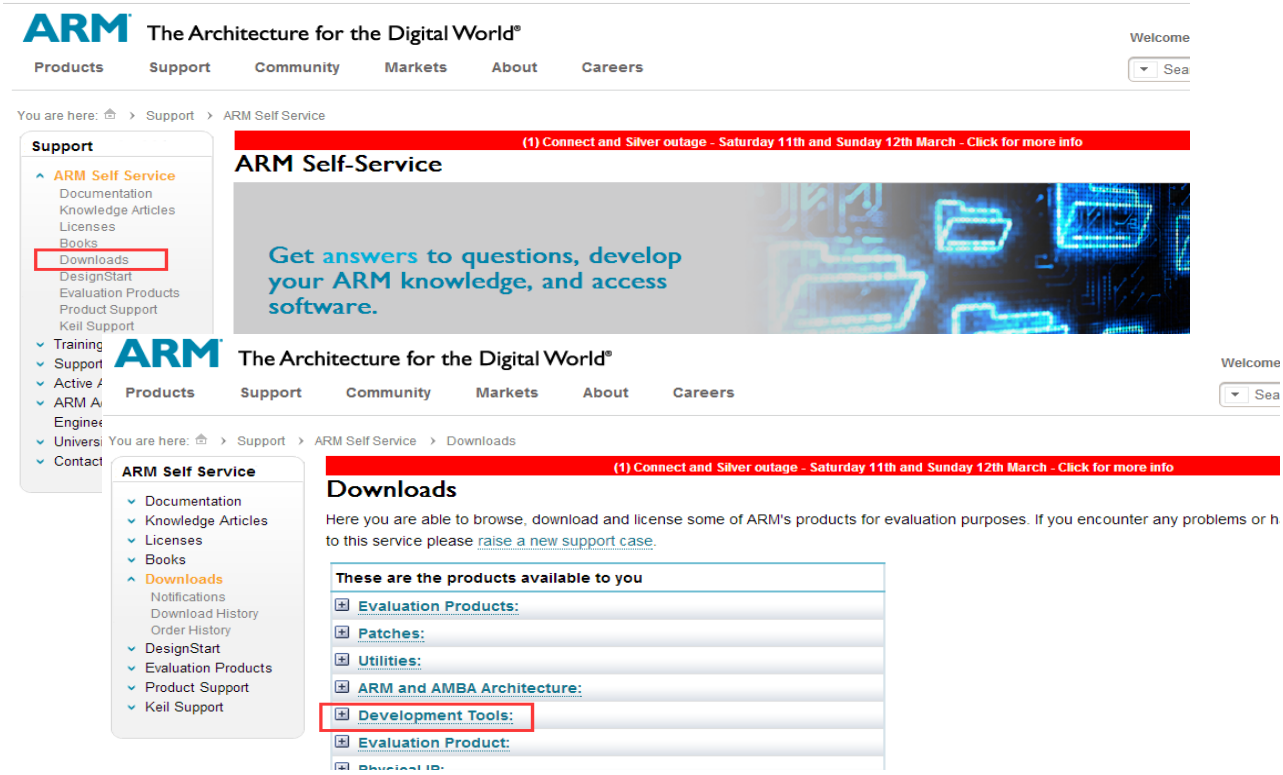

Figure 2: "Downloads" and "Development Tools" Pages

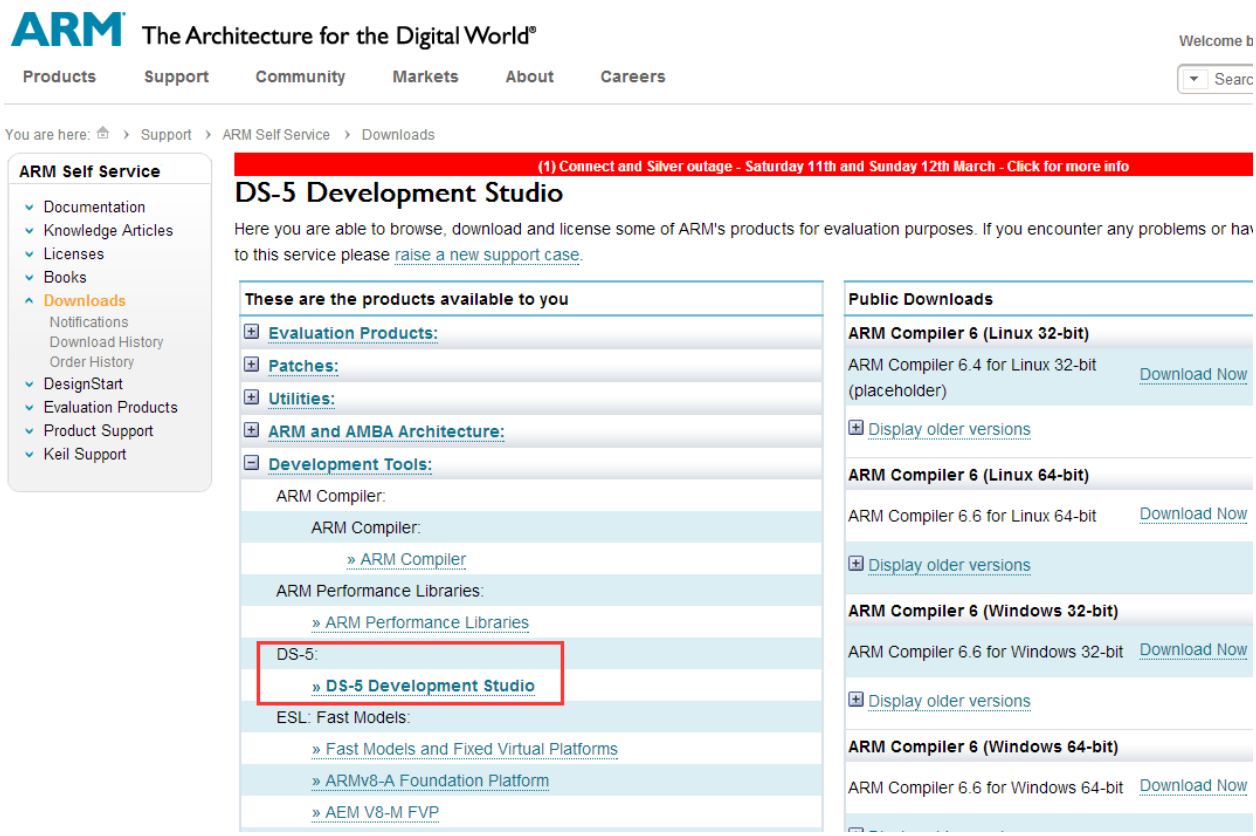

Figure 3: Click "DS-5 Development Studio"

(2) Under "**ARM Compiler 5 (Windows)**", click the "**Download Now**" button after "**ARM Compiler 5.05 update 1 (build 106) for Windows**" to download the corresponding ARM compiler tool for Windows.

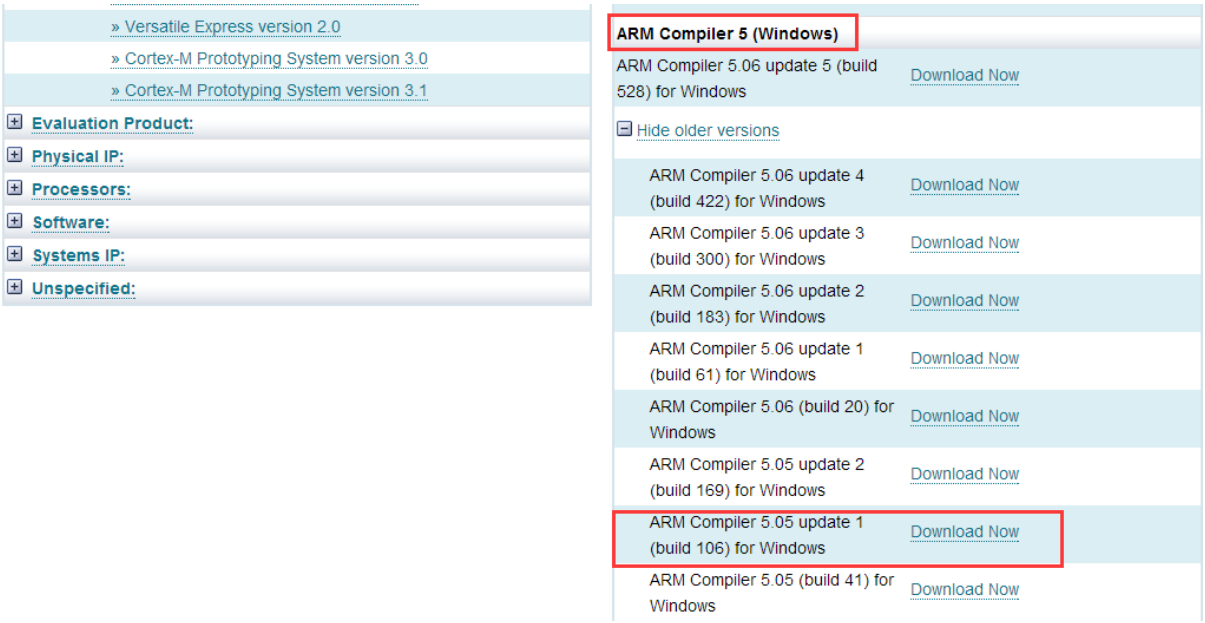

Figure 4: Download the Corresponding Tool

(3) After clicking "**Download Now**", there is a need to confirm the details shown as below:

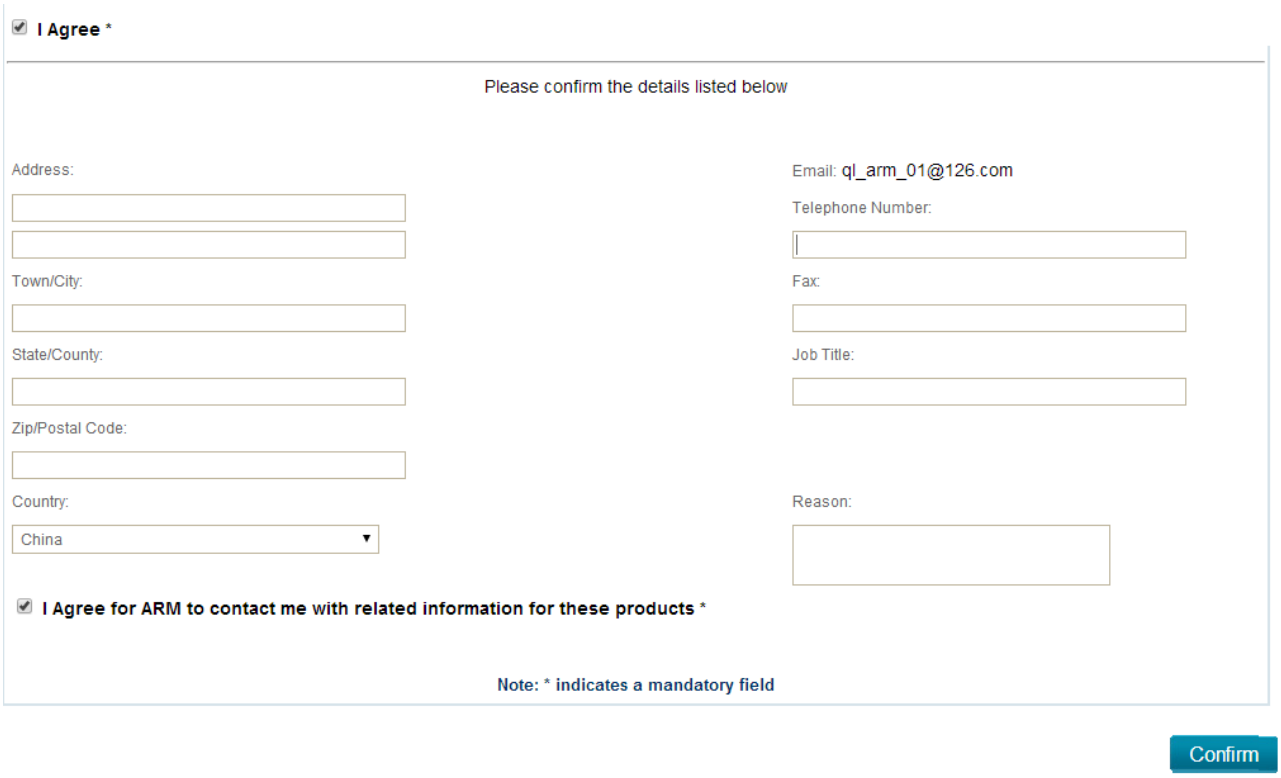

Figure 5: Confirmation of Details

(4) Finally click "**Confirm**" button and then the tool packet will be downloaded.

3.1.1.2. Install ARM Compiler Tool

After downloading ARM compiler tools, you can follow the steps illustrated below to finish installation of ARM compiler tool.

Step 1: Run "ARM Compiler 5 Setup" program and then click "**Next**".

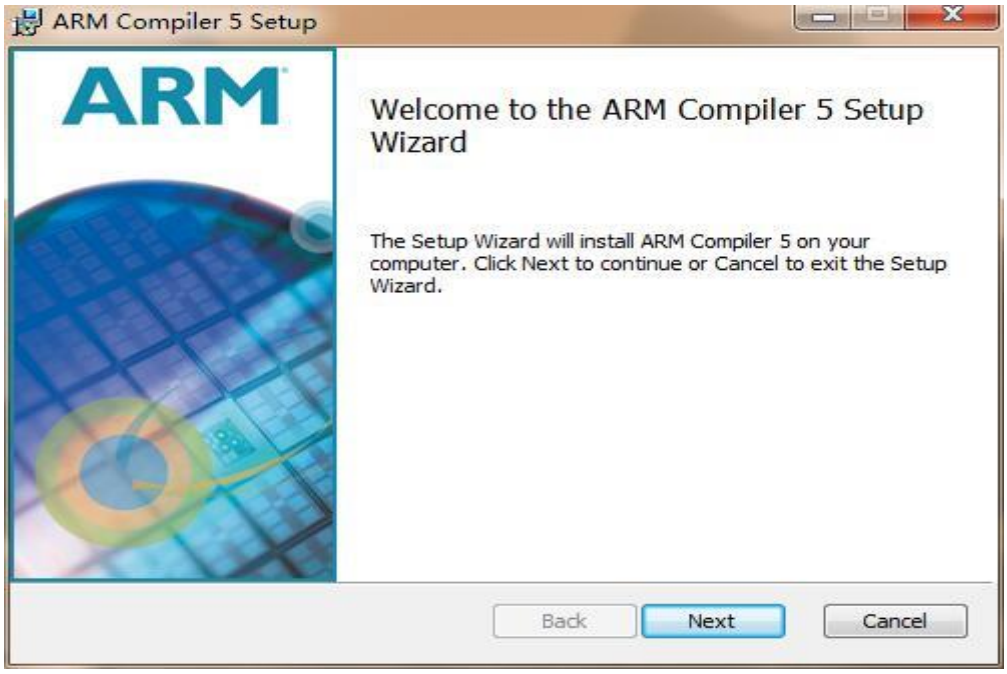

Figure 6: ARM Compiler 5 Setup

Step 2: Accept the terms in the license agreement and then click "**Next**".

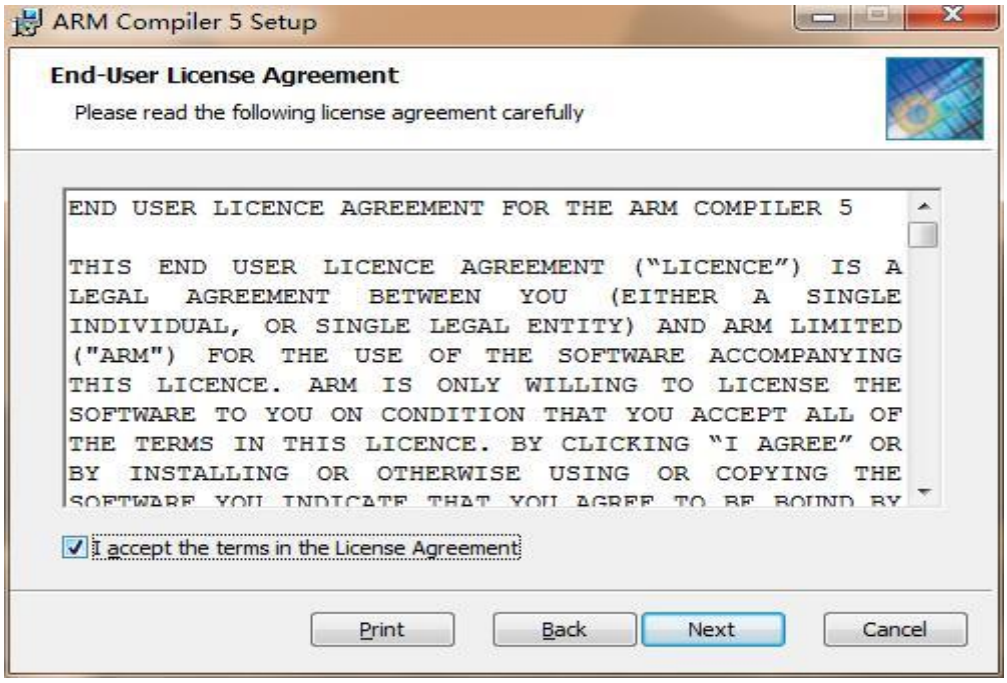

Figure 7: End-User License Agreement

Step 3: Select the way you want features to be installed.

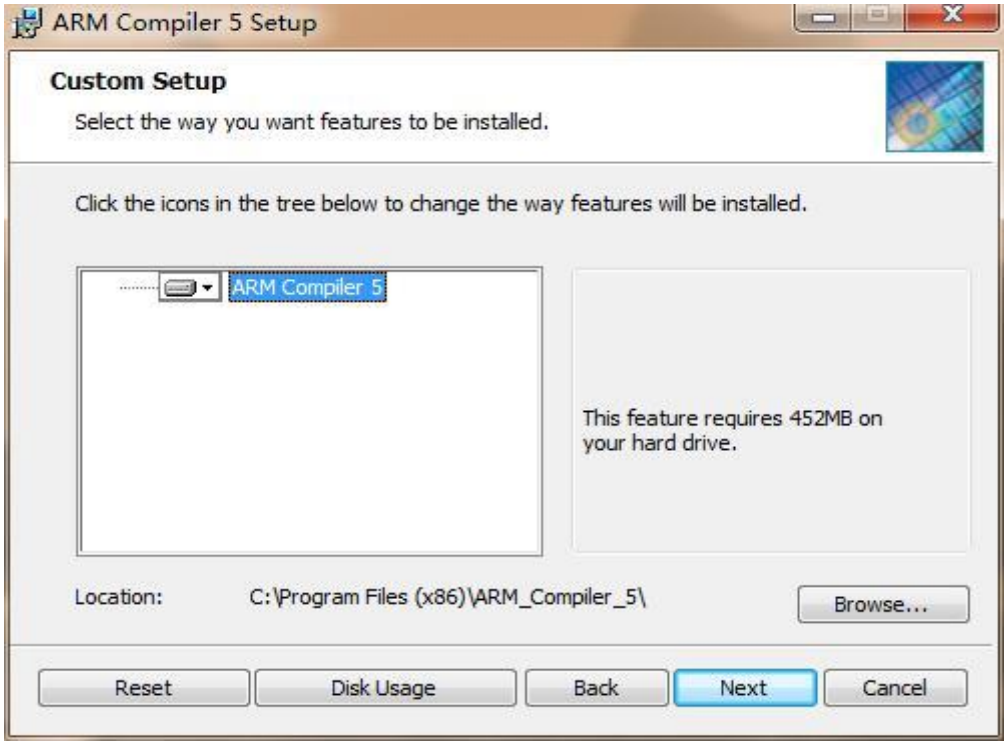

Figure 8: Custom Setup

Step 4: Ignore "**System Pending Reboot**" warning, and click "**Next**".

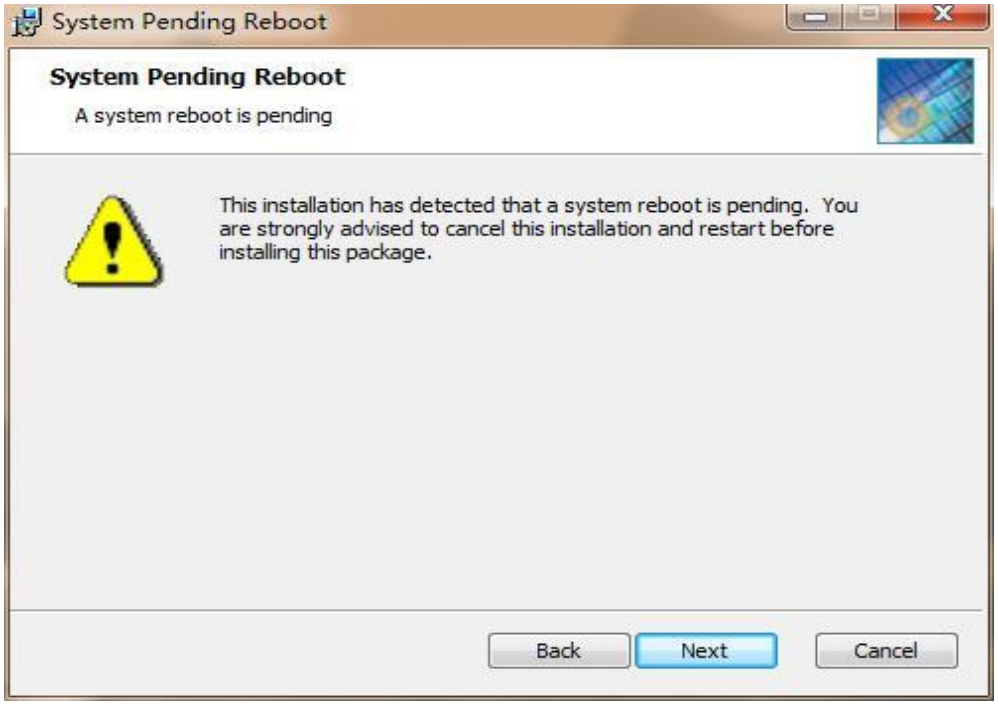

Figure 9: "System Pending Reboot" Warning

Step 5: Click "**Install**" to begin the installation, then wait while the setup wizard installs ARM compiler 5.

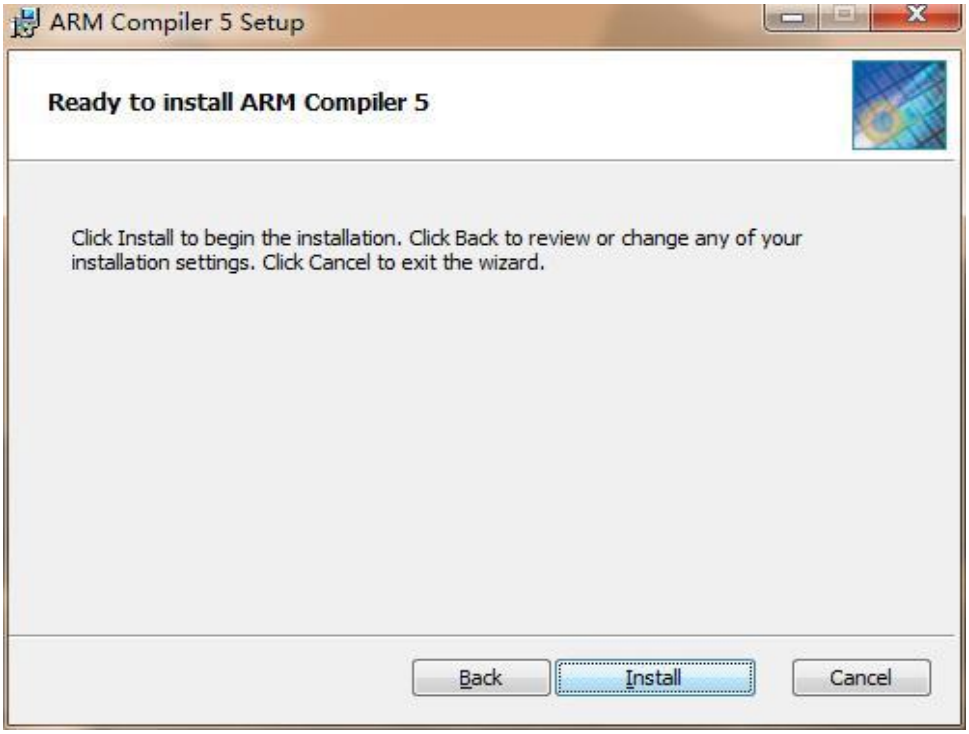

Figure 10: Ready to install ARM Compiler 5

Step 6: Click the "**Finish**" button to exit the setup wizard and complete the compiler tool installation.

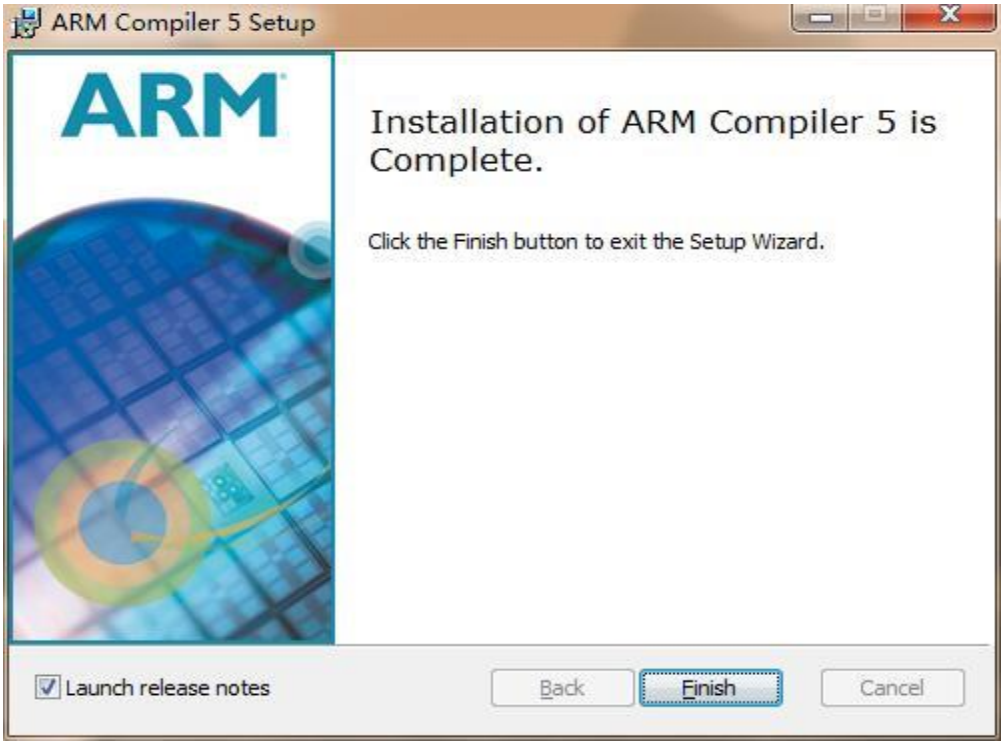

Figure 11: Finish Installation of ARM Compiler Tool

After successful installation of ARM compiler 5, there is a need to restart the computer to make the compilation tool take effect.

3.1.2. Download and Install Cygwin

3.1.2.1. Download Cygwin

Open the Cygwin download page shown as below to download the corresponding revision of Cygwin for Windows: https://cygwin.com/install.html.

3.1.2.2. Install Cygwin

To install the environment where you can compile the BG96 QuecOpen application, please follow the steps below:

Step 1: Run "**Cygwin Setup**" program and then click "**Next**".

Figure 12: Cygwin Setup Program

Step 2: Choose the installation type.

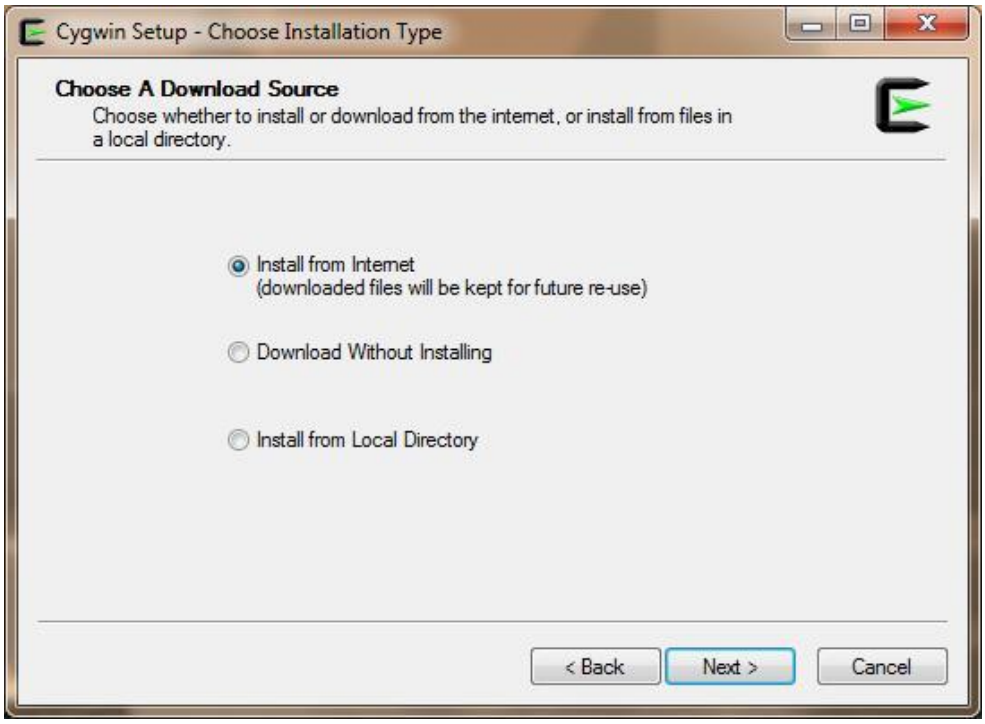

Figure 13: Choose Installation Type

Step 3: Select the directory where you want to install Cygwin, and also please choose a few installation parameters.

Figure 14: Choose Installation Directory and Parameters

Step 4: Select a directory where you want the setup to store the downloaded installation files.

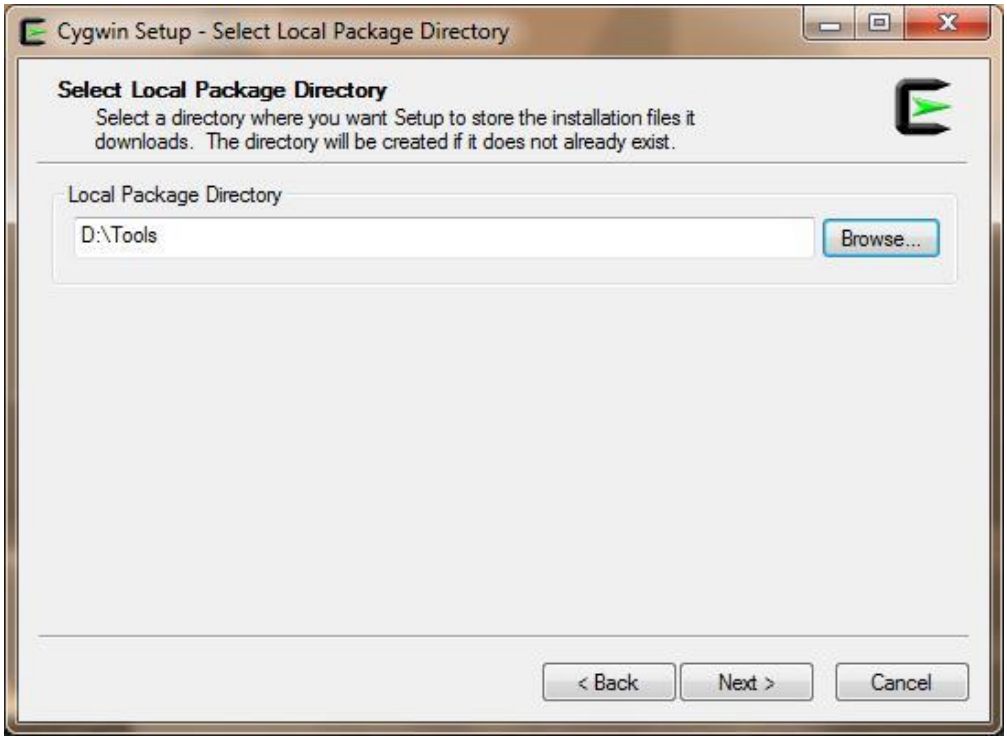

Figure 15: Select Local Package Directory

Step 5: Select the type of internet connection.

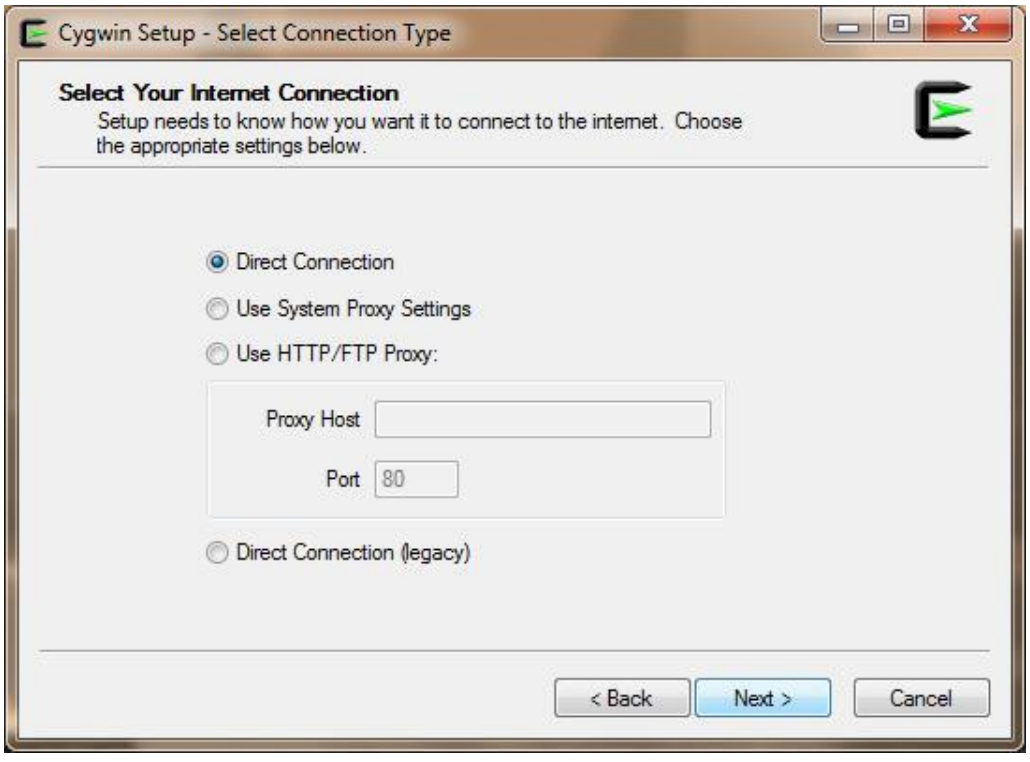

Figure 16: Select Internet Connection Type

Step 6: Then you will see the installation progress. After it completes, please choose a site from the list.

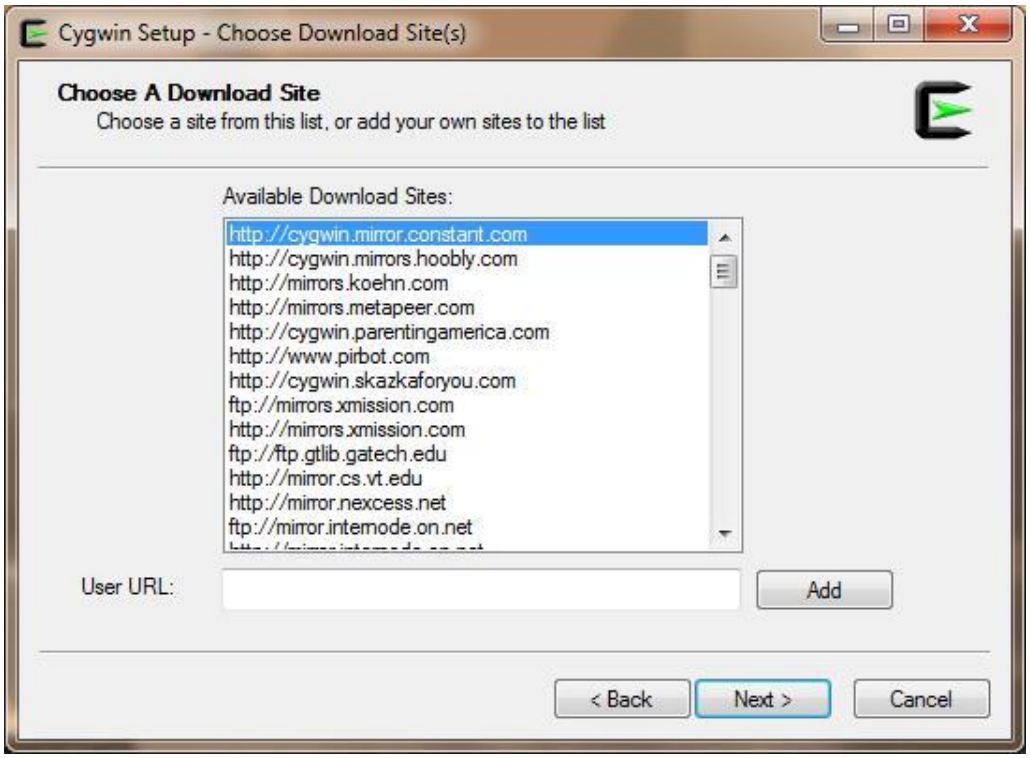

Figure 17: Choose a Download Site

Step 7: Please wait patiently during download or installation progress. When it completes, click "**Next**".

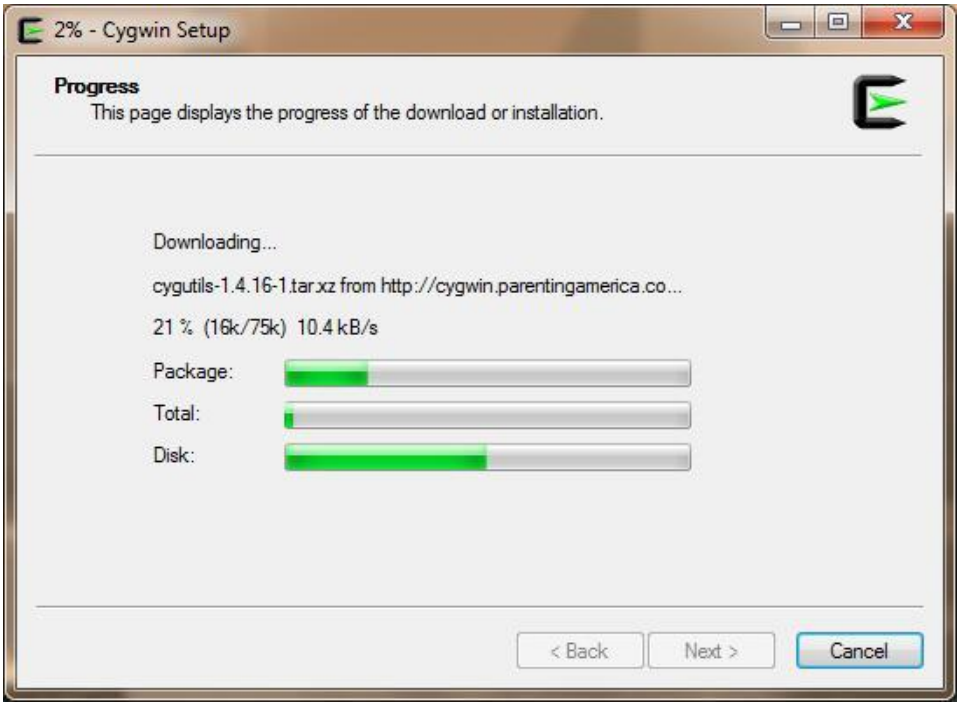

Figure 18: Process of Download or Installation

Step 8: Create an icon for Cygwin, and then click "**Finish**" to complete installation.

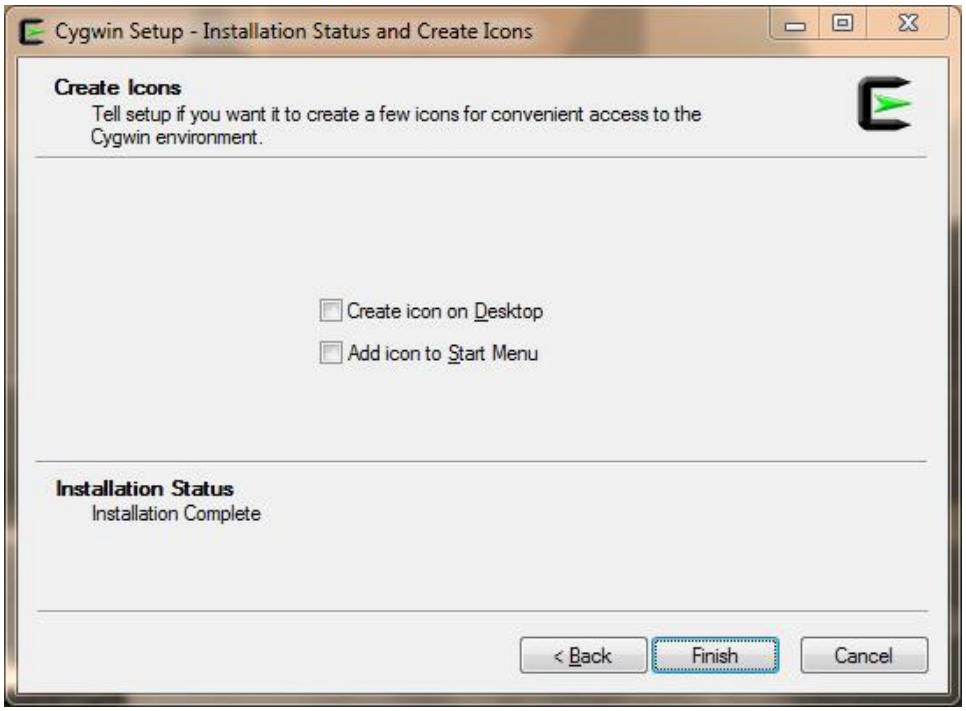

Figure 19: Create Icon and Complete Installation

After successful installation of both ARM compiler tool and Cygwin, customers can start compiling QuecOpen SDK. For details about compilation and running of QuecOpen SDK, please refer to ***Chapter 4*** and ***Chapter 5*** for details.

3.2. Setup Compiling Environment for TX3.0.1

3.2.1. Download LLVM

Licensed users can download LLVM compiler through the Qualcomm ChipCode™ portal. If Quectel customers have no license, please contact with Quectel.

3.2.1.1. For Windows 7 Build Hosts

Figure 20: LLVM Download for Windows 7 Build Hosts

3.2.1.2. For Linux Build Hosts

Figure 21: LLVM Download for Linux Build Hosts

After download, the tool can be used without installation procedure.

3.2.2. Download and Install Cygwin

Please refer to ***Chapter 3.1.2*** for details.

3.2.3. Download and Install Python

3.2.3.1. Download Python

Open the Python download page shown as below to download the corresponding revision of Python for Windows/Linux: https://www.python.org/download/releases/2.7/.

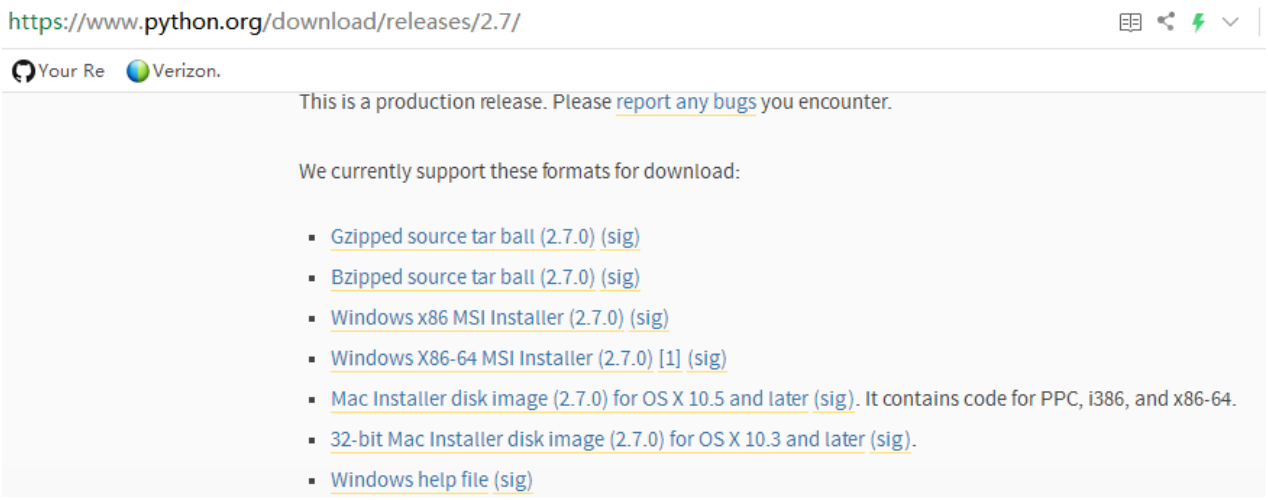

Figure 22: Python Download Page Screenshot

Download x86/x86-64 versions as needed.

3.2.3.2. Install Python

After download is completed, please follow the steps illustrated below to finish installation.

Step 1: Run "**Python-2.7.0.msi**" program and also please choose a few installation parameters, then click "**Next**".

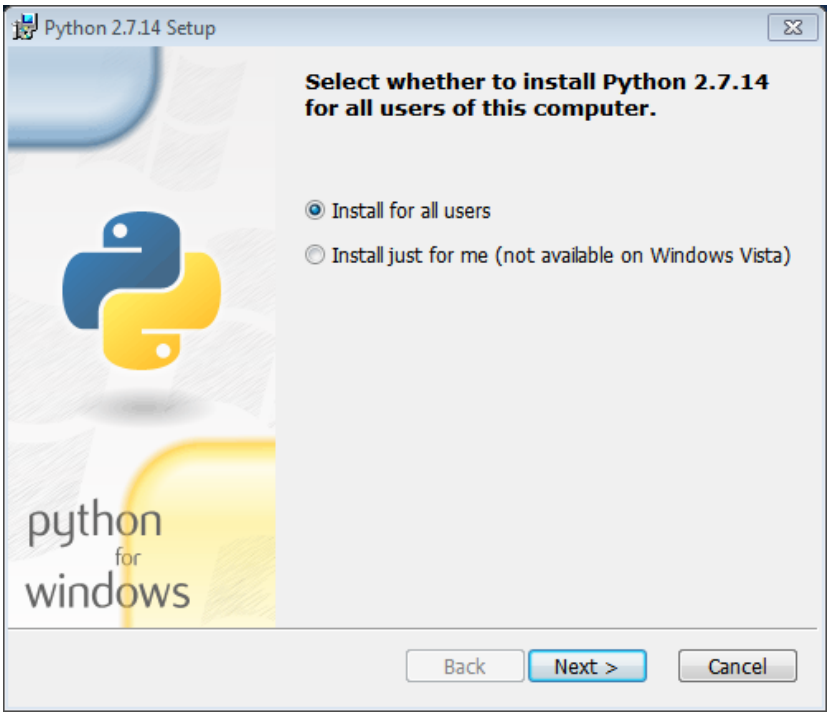

Figure 23: Python Setup

Step 2: Select the directory where Python is to be installed.

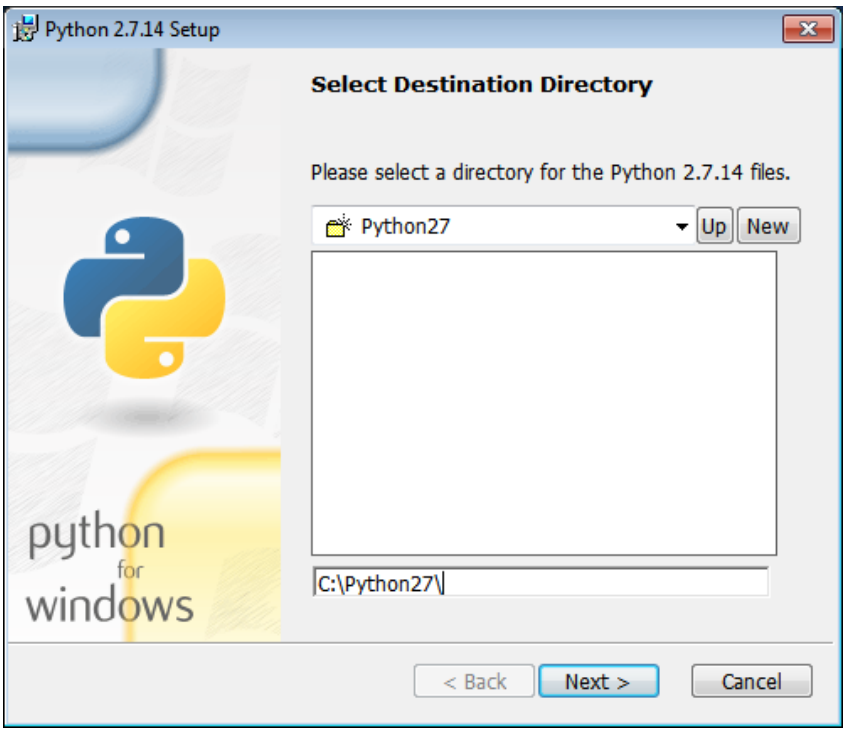

Figure 24: Select Installation Directory

Step 3: Options for customization. Please keep the default options.

Figure 25: Options for Customization

Step 4: Please wait during installation process.

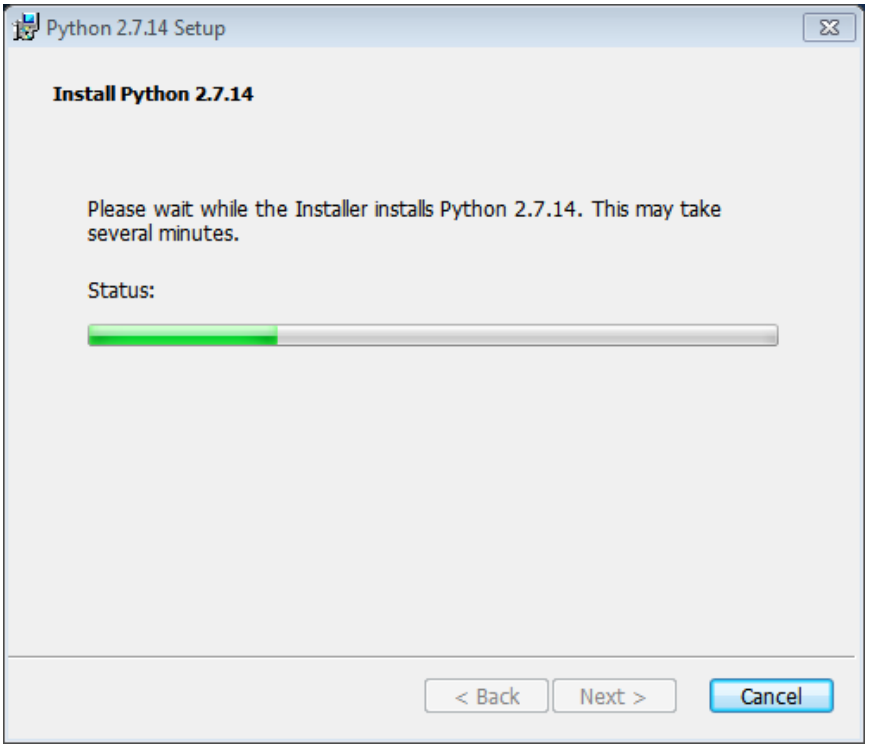

Figure 26: Installing

Step 5: Complete installation.

Figure 27: Installation Completed

4 Build QuecOpen Application

4.1. QuecOpen SDK Package based on TX2.0

4.1.1. SDK Package Structure

The following shows the folder structure of *Quectel_BG96_QuecOpen_SDK_Package* (based on TX2.0) which is created for non-licensed customers.

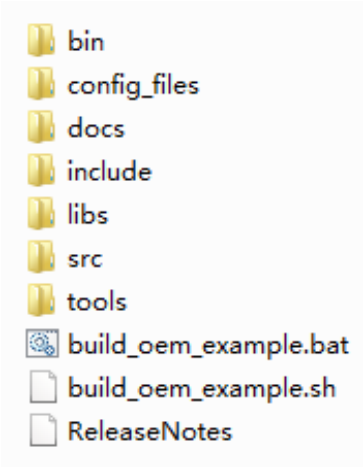

Figure 28: Folder Structure of BG96-QuecOpen SDK Package (TX2.0)

Table 3: Description of BG96-QuecOpen SDK Package Directories (TX2.0)

Directories	Description/Function
bin	Application gets created in this folder after successful compilation
config_files	Contains application related configuration file: *oem_app_path.ini*
docs	Guide documents
Include	Header files needed for compilation provided by Quectel
libs	Required libraries should be copied here

src	Application source code
tools	Tools for development
build_oem_example.bat	Batch script for Windows build hosts
build_oem_example.sh	Shell script for Windows build hosts
ReleaseNotes	Firmware release notes

4.1.2. Build QuecOpen User Application

Before application building, customers must set a correct path for the compiler tools in the build script.

In *build_oem_example.bat*:

```
set TOOL_PATH_ROOT=C:\compile_tools
set TOOLCHAIN_PATH=%TOOL_PATH_ROOT%\ARM_Compiler_5\bin
set LM_LICENSE_FILE=%TOOL_PATH_ROOT%\license.dat
```

In *build_oem_example.sh*:

```
TOOLCHAIN_PATH="C:/compile_tools/ARM_Compiler_5/bin"
export LM_LICENSE_FILE="C:/compile_tools/license.dat"
```

NOTE

The ARM compiler is not provided for free. If it is intended to be used, customers have to obtain the license first. Fortunately, a 30-day evaluation license is available for new users.

To build the example codes in the *Quectel_BG96_QuecOpen_SDK_Package*, customers just need to run the following command from command line in Cygwin or DOS window:

The following help commands can be used to know which examples are supported in this SDK package:

```
help build in Linux:
./build_oem_example.sh help
```

Or

```
help build in Windows:
build_oem_example.bat help
```

After input the help command, tips shown as below will be available:

```
Supported example :
   device_info [ cmd - build_oem_example.bat device_info ]
   gpio        [ cmd - build_oem_example.bat gpio        ]
   gpio_int    [ cmd - build_oem_example.bat gpio_int    ]
   gps         [ cmd - build_oem_example.bat gps         ]
   qt_gps      [ cmd - build_oem_example.bat qt_gps      ]
   http        [ cmd - build_oem_example.bat http        ]
   psm         [ cmd - build_oem_example.bat psm         ]
   rtc         [ cmd - build_oem_example.bat rtc         ]
   task_create [ cmd - build_oem_example.bat task_create ]
   tcp_client  [ cmd - build_oem_example.bat tcp_client  ]
   time        [ cmd - build_oem_example.bat time        ]
   timer       [ cmd - build_oem_example.bat timer       ]
   uart        [ cmd - build_oem_example.bat uart        ]
   atc_pipe    [ cmd - build_oem_example.bat atc_pipe    ]
   atc_sms     [ cmd - build_oem_example.bat atc_sms     ]
   i2c         [ cmd - build_oem_example.bat i2c         ]
   mqtt        [ cmd - build_oem_example.bat mqtt        ]
   spi         [ cmd - build_oem_example.bat spi         ]
   dns_client  [ cmd - build_oem_example.bat dns_client  ]
   adc         [ cmd - build_oem_example.bat adc         ]
   qt_adc      [ cmd - build_oem_example.bat qt_adc      ]
   nonip       [ cmd - build_oem_example.bat nonip       ]
   fota        [ cmd - build_oem_example.bat fota        ]
   lwm2m       [ cmd - build_oem_example.bat lwm2m       ]
```

Take UART compilation as an example:

- In Cygwin:

New build:
./build_oem_example.sh uart

Clean build:
./ build_oem_example.sh -c

- In DOS window:

New build:
build_oem_example.bat uart

Clean build:
build_oem_example.bat -c

Once the build process is completed, the application binary (*example_uart.bin*) will be created under the path */bin*.

4.2. QuecOpen SDK Package based on TX3.0.1

4.2.1. SDK Package Structure

The following shows the folder structure of *Quectel_BG96_QuecOpen_SDK_Package* (based on TX3.0.1) which is created for non-licensed customers.

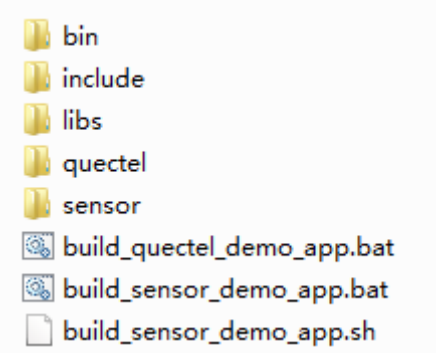

Figure 29: Folder Structure of BG96-QuecOpen SDK Package (TX3.0.1)

Table 4: Description of BG96-QuecOpen SDK Package Directories (TX3.0.1)

Directories	Description/Function
bin	Application gets created in this folder after successful compilation
Include	Header files needed for compilation provided by Quectel
libs	Required libraries should be copied here
quectel	Quectel example source codes
sensor	Qualcomm example source codes
build_quectel_demo_app.bat	Batch script for building Quectel example
Build_sensor_demo_app.bat	Batch script for building Qualcomm example
Build_sensor_demo_app.sh	Shell script for building Qualcomm example

4.2.2. Build QuecOpen User Application

Before application building, customers must set a correct path for the compiler tools in the build script.

- In *build_quectel_demo_app.bat*:

```
set TOOL_PATH_ROOT=C:\compile_tools
set TOOLCHAIN_PATH=%TOOL_PATH_ROOT%\LLVM\4.0.3\bin
set
TOOLCHAIN_PATH_STANDARdS=%TOOL_PATH_ROOT%\LLVM\4.0.3\armv7m-none-eabi\libc\include
set LLVMLIB=%TOOL_PATH_ROOT%\LLVM\4.0.3\lib\clang\4.0.3\lib
set LLVMLINK_PATH=%TOOL_PATH_ROOT%\LLVM\4.0.3\tools\bin
set PYTHON_PATH=%TOOL_PATH_ROOT%\Python27\python.exe
```

- In *build_quectel_demo_app.sh*:

```
set TOOL_PATH_ROOT=C:\compile_tools
set TOOLCHAIN_PATH=%TOOL_PATH_ROOT%\LLVM\4.0.3\bin
set
TOOLCHAIN_PATH_STANDARdS=%TOOL_PATH_ROOT%\LLVM\4.0.3\armv7m-none-eabi\libc\include
set LLVMLIB=%TOOL_PATH_ROOT%\LLVM\4.0.3\lib\clang\4.0.3\lib
set LLVMLINK_PATH=%TOOL_PATH_ROOT%\LLVM\4.0.3\tools\bin
set PYTHON_PATH=%TOOL_PATH_ROOT%\Python27\python.exe
```

To build the example codes in the *Quectel_BG96_QuecOpen_SDK_Package*, customers just need to run the following command from command line in Linux or Windows OS:

The following help commands can be used to know which examples are supported in this SDK package:

```
help build in Linux:
./build_quectel_demo_app.sh help
```

Or

```
help build in Windows:
build_quectel_demo_app.bat help
```

After input the help command, tips shown as below will be available:

```
Supported example :
   adc          [ cmd - build_quectel_demo_app.bat adc          ]
   device_info  [ cmd - build_quectel_demo_app.bat device_info  ]
   dns_client   [ cmd - build_quectel_demo_app.bat dns_client   ]
   gpio         [ cmd - build_quectel_demo_app.bat gpio         ]
   gpio_int     [ cmd - build_quectel_demo_app.bat gpio_int     ]
   gps          [ cmd - build_quectel_demo_app.bat gps          ]
   http         [ cmd - build_quectel_demo_app.bat http         ]
   i2c          [ cmd - build_quectel_demo_app.bat i2c          ]
   mqtt         [ cmd - build_quectel_demo_app.bat mqtt         ]
   psm          [ cmd - build_quectel_demo_app.bat psm          ]
   rtc          [ cmd - build_quectel_demo_app.bat rtc          ]
   spi          [ cmd - build_quectel_demo_app.bat spi          ]
   task_create  [ cmd - build_quectel_demo_app.bat task_create  ]
   tcp_client   [ cmd - build_quectel_demo_app.bat tcp_client   ]
   time         [ cmd - build_quectel_demo_app.bat time         ]
   timer        [ cmd - build_quectel_demo_app.bat timer        ]
   uart         [ cmd - build_quectel_demo_app.bat uart         ]
```

Take UART compilation as an example:

- In Linux:

New build:
./build_oem_example.sh uart

Clean build:
./ build_oem_example.sh -c

- In Windows:

New build:
build_oem_example.bat uart

Clean build:
build_oem_example.bat -c

Once the build process is completed, the application binary (*quectel_demo_uart.bin*) will be created under the path */bin*.

5 Run QuecOpen Application

To run the QuecOpen application binary file, customers only need to upload the application binary image and *oem_app_path.ini* into the alternate file systems of BG96 by QEFS Explorer.

The *oem_app_path.ini* file includes the full path of the location of application binary image. This file must be stored in the */datatx/* directory.

After uploading these two files into alternate file systems, restart BG96 and the application binary image will be loaded into RAM and started by the Module Loader.

NOTE

For detailed usage of QEFS Explorer, please refer to *Quectel_BG96_QEFS_Explorer_User_Guide*.

6 Update QuecOpen Application

Customers can download their new application image form their own Revision Control Server. For example, they can use HTTP(s) to access the HTTP(s) server and download the upgrade image to the file system and store it in the specified path.

In the file system, there are two places which are used to store the application images. For instance, the path to existing UART application could be:

- */datatx/quectel_uart_demo.bin* (path #1)

An upgraded image can be downloaded into:

- */datatx/upgrade/quectel_uart_demo_upgrade.bin* (path #2)

Also, customers need modify the *oem_app_path.ini* file contents according to the format shown as below:

/datatx/quectel_uart_demo.bin:/datatx/ upgrade/quectel_uart_demo_upgrade.bin

- "/datatx/quectel_uart_demo.bin" indicates the full path of the boot-up image
- "/datatx/quectel_uart_demo_upgrade.bin" indicates the full path of the upgraded image

After download is completed, customers only need to restart the module.

While booting, the Module Loader can check:

- Whether the path #2 is present and will load it if it presents.
- If loading fails or the file is not present, it reverts to path #1 and will delete path #2.
- If it loads the new app successfully, it will copy path #2 to path #1, and delete path #2 after copy completed.

NOTES

1. ":" is the delimiter, and is used to distinguish the path of original version from that of upgraded version.
2. The firmware name and storage path of the old firmware and upgraded firmware are customer specific. Customers could modify them according to their individual demands. But the delimiter ":" must not be modified or replaced with other symbols.

7 Appendix A References

Table 5: Related Documents

SN	Document Name	Remark
[1]	Quectel_BG96_QEFS_Explorer_User_Guide	QEFS Explorer tool user guide

Table 6: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
OS	Operating System
HTTP	Hyper Text Transfer Protocol
SDK	Software Development Kit
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
SSL	Security Socket Layer