BG96 QuecOpen Application Note

LPWA Module Series

Rev. BG96_QuecOpen_Application_Note_V1.0

Date: 2020-06-10

Status: Released

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

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

For technical support, or to report documentation errors, please visit:
http://www.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

Revision History

Version	Date	Author	Description
1.0	2020-06-10	Egbert XU/ Justice HAN/ Cristiano RAO	Initial

Contents

Table Index

Figure Index

1 Introduction

This document mainly introduces how to set up the compiling environment for Quectel BG96 QuecOpen® module in Windows and Linux Operating Systems, how to compile user applications in BG96 QuecOpen® SDK, and how to run and update user applications in QuecOpen® solution.

2 QuecOpen® Solution Overview

2.1. General Overview

Quectel BG96 QuecOpen module 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. QuecOpen Architecture

The following diagram shows the architecture of QuecOpen solution.

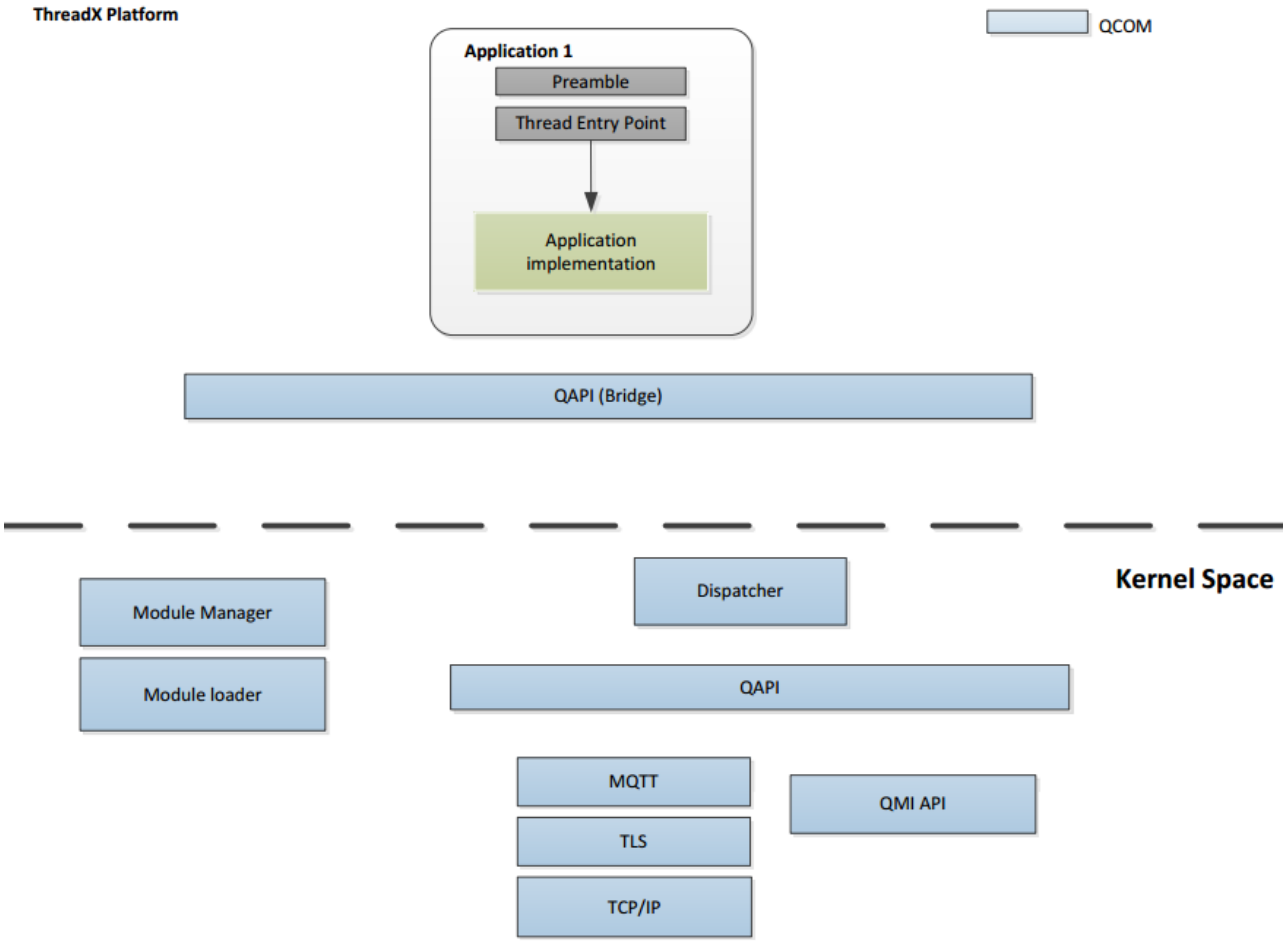

Figure 1: Architecture of QuecOpen® Solution

3 Build QuecOpen® Applications

3.1. QuecOpen SDK Package based on TX2.0

3.1.1. SDK Package Structure

The following shows the folder structure of *Quectel_BG96_QuecOpen_SDK_Package* (based on TX2.0).

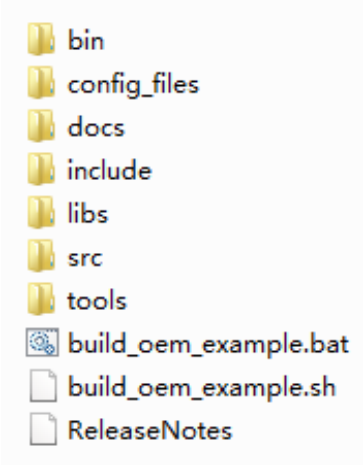

Figure 2: Folder Structure of BG96 QuecOpen SDK Package (TX2.0)

Table 1: 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

3.1.2. Build QuecOpen User Application

Before application building, a correct path must be set 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, obtain the license first. Fortunately, a 30-day evaluation license is available on the ARM official website for new users.

To build the example codes in the *Quectel_BG96_QuecOpen_SDK_Package*, 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 are 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       ]
```

Figure 3: Help Tips (TX2.0)

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 (e.g., *example_uart.bin*) is created under the path */bin*.

3.2. QuecOpen SDK Package based on TX3.0.1

3.2.1. SDK Package Structure

The following shows the folder structure of *Quectel_BG96_QuecOpen_SDK_Package* (based on TX3.0.1).

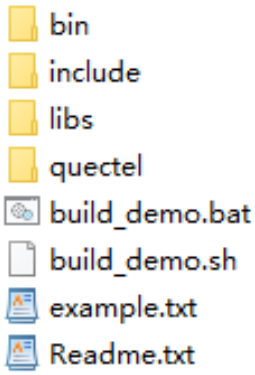

Figure 4: Folder Structure of BG96 QuecOpen® SDK Package (TX3.0.1)

Table 2: 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
build_demo.bat	Batch script for building Quectel examples
build_demo.sh	Shell script for building Qualcomm examples
example.txt	Configuration files for compilation options of examples
Readme.txt	Key information for QuecOpen SDK package

3.2.2. Build QuecOpen User Applications

Before application building, set up the compiling environment and set a correct path for the compiler tool in the build script. For more details of compiling environment setup, see ***Appendix B***.

- In *build_demo.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\includ
e
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_demo.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\includ
e
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*, run the help command first to check the supported example codes, and then run the build commands for applications compilation from the command line in Linux or Windows OS.

- **Run help command to check supported example codes**

help build in Linux:

```
./build_demo.sh llvm help
```

Or

help build in Windows:

```
build_demo.bat llvm help
```

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

```
adc          [ cmd - build_demo.bat llvm adc          ]
analog_spi   [ cmd - build_demo.bat llvm analog_spi   ]
atc_pipe     [ cmd - build_demo.bat llvm atc_pipe     ]
atc_sms      [ cmd - build_demo.bat llvm atc_sms      ]
atfwd        [ cmd - build_demo.bat llvm atfwd        ]
dam_fota     [ cmd - build_demo.bat llvm dam_fota     ]
device_info  [ cmd - build_demo.bat llvm device_info  ]
dns_client   [ cmd - build_demo.bat llvm dns_client   ]
file         [ cmd - build_demo.bat llvm file         ]
fota         [ cmd - build_demo.bat llvm fota         ]
ftp_client   [ cmd - build_demo.bat llvm ftp_client   ]
gpio         [ cmd - build_demo.bat llvm gpio         ]
gpio_int     [ cmd - build_demo.bat llvm gpio_int     ]
gps          [ cmd - build_demo.bat llvm gps          ]
http         [ cmd - build_demo.bat llvm http         ]
i2c          [ cmd - build_demo.bat llvm i2c          ]
lwm2m        [ cmd - build_demo.bat llvm lwm2m        ]
mqtt         [ cmd - build_demo.bat llvm mqtt         ]
ping         [ cmd - build_demo.bat llvm ping         ]
psm          [ cmd - build_demo.bat llvm psm          ]
qt_adc       [ cmd - build_demo.bat llvm qt_adc       ]
qt_gps       [ cmd - build_demo.bat llvm qt_gps       ]
rtc          [ cmd - build_demo.bat llvm rtc          ]
spi          [ cmd - build_demo.bat llvm spi          ]
ssl          [ cmd - build_demo.bat llvm ssl          ]
task_create  [ cmd - build_demo.bat llvm task_create  ]
```

Figure 5: Help Tips (TX3.0)

NOTE

The above example code lists are provided for your reference only. See the SDK for the specific contents.

- **Run build commands for applications compilation**

Take UART compilation as an example:

- In Linux:

New build:
./build_demo.sh llvm uart

Clean build:
./build_demo.sh llvm -c

- In Windows:

New build:
build_demo.bat llvm uart

Clean build:

```
build_demo.bat llvm -c
```

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

4 Run QuecOpen® Applications

To run the QuecOpen application binary file, upload the application binary image and *oem_app_path.ini* into the alternate file systems of the module 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, reboot the module and then the application binary image is loaded into RAM and started by the Module Loader.

NOTE

For detailed usage of QEFS Explorer, see *Quectel_BG96_QEFS_Explorer_User_Guide*.

5 Update QuecOpen® Applications

5.1. Download the New Image

Users can download their new application image from their own Revision Control Server. For example, HTTP(S) QAPIs can be used to access the HTTP(S) server and download the new image to the file system and store it in a specified path.

In the file system, there are two separate places to store the current and new application images.

Taking UART application as an example, the path for the current image is:

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

The new image can be downloaded into:

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

5.2. Update the Content of *oem_app_path.ini* File

Accordingly, the content of *oem_app_path.ini* file need to be modified 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 current boot-up image
- */datatx/quectel_uart_demo_upgrade.bin* indicates the full path of the new image

5.3. Reset the Module

Reboot the module after downloading is completed. While booting, the Module Loader checks whether the **path #2** is present and will load it if it presents.

- If loading fails or the file is not present, the Module Loader reverts to **path #1** and deletes **path #2**.

- If the Module Loader loads the new application image 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 current image from that of new image.
2. The name and storage path of the current and new images are customer specific. They can be modified according to specific demands. But the delimiter ":" must not be modified or replaced with other symbols.

6 Pin Mapping

This chapter mainly introduces the pin mapping of BG96 QuecOpen module, including the mapping of related peripherals, such as UART, IIC and SPI.

6.1. Pin Mapping

The following table shows the pin mapping of BG96 QuecOpen module.

Table 3: Multiplexing Pins of BG96 QuecOpen Module

Pin Name	Pin No.	Mode 1	Mode 2	Mode 3	Mode 4	Reset [1)]	Interrupt*	Remark
GPIO01	4	GPIO_23	GPIO_23	SPI1_CLK	/	B-PU	/	BOOT_ CONFIG_4
GPIO02	5	GPIO_20	UART1_TX	SPI1_MOSI	/	B-PD	●	
GPIO03	6	GPIO_21	UART1_RX	SPI1_MISO	/	B-PD	●	
GPIO04	7	GPIO_22	GPIO_22	SPI1_CS_N	/	B-PD	●	
GPIO05	18	GPIO_11	/	SPI2_CLK	I2C2_SCL	B-PU	●	
GPIO06	19	GPIO_10	/	SPI2_CS_N	I2C2_SDA	B-PD	/	
GPIO07	22	GPIO_09	UART2_RX	SPI2_MISO	/	B-PD	●	
GPIO08	23	GPIO_08	UART2_TX	SPI2_MOSI	/	B-PD	●	
GPIO09	26	GPIO_15	GPIO_15	/	/	B-PD	/	
GPIO10	27	GPIO_12	UART3_TX	/	/	B-PD	●	
GPIO11	28	GPIO_13	UART3_RX	/	/	B-PD	●	
GPIO19	40	GPIO_19	/	/	I2C1_SCL	B-PD	/	
GPIO20	41	GPIO_18	/	/	I2C2_SDA	B-PD	/	

GPIO21	64	GPIO_07	/	/	/	B-PD	/

NOTES

1. Mode 1 to Mode 4 are the pin functions available through software configuration and the pin functions in Model 1/2/3/4 take effect only after software configuration.
2. The BOOT_CONFIG_4 pin (GPIO01) cannot be pulled up before startup.
3. "*" means under development.
4. "●" means supported.
5. "/" means not supported.
6. The "Pin Name" refers to the pin names in QuecOpen solution, which may be different from the corresponding pin name in BG96 module of standard version.

6.1.1. GPIOs

BG96 QuecOpen module supports maximum up to 14 GPIOs based on application designs. Each GPIO can be configured through QAPIs in QuecOpen applications. When a GPIO is configured into an output, its drive strength can be configured, maximally up to 16 mA.

Some GPIOs support interrupt function, and specified GPIOs can be configured as an interrupt.

6.1.2. UART Interfaces

The module provides four UART interfaces: main UART, UART1, UART2 and UART3.

- The main UART interface can only be used for AT command communication.
- UART1, UART2 and UART3 interfaces are used for communication and data transmission with peripherals, and can also be multiplexed into other functions.

The following tables show the pin definition of the four UART interfaces.

Table 4: Pin Definition of Main UART Interface

Pin Name	Pin No.	I/O	Description	Comment
DTR	30	DI	Data terminal ready	1.8 V power domain
RXD	34	DI	Receive data	1.8 V power domain
TXD	35	DO	Transmit data	1.8 V power domain
CTS	36	DO	Clear to send	1.8 V power domain

RTS	37	DI	Request to send	1.8 V power domain
DCD	38	DO	Data carrier detection	1.8 V power domain
RI	39	DO	Ring indication	1.8 V power domain

NOTE

The main UART interface is designed only for AT command communication. It cannot be configured for other application purposes.

Table 5: Pin Definition of UART1 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO02	5	IO	GPIO_20	**UART1_TX**	SPI1_MOSI	/	1.8 V power domain.
GPIO03	6	IO	GPIO_21	**UART1_RX**	SPI1_MISO	/	1.8 V power domain.

NOTES

1. In QuecOpen application, QAPI_UART_PORT_001_E can be used to select and configure UART1.
2. UART1 port does not support flow control.

Table 6: Pin Definition of UART2 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO07	22	IO	GPIO_09	**UART2_RX**	SPI2_MISO	/	1.8 V power domain.
GPIO08	23	IO	GPIO_08	**UART2_TX**	SPI2_MOSI	/	1.8 V power domain.

NOTES

1. In QuecOpen application, QAPI_UART_PORT_002_E can be used to select and configure UART2.
2. UART2 port does not support flow control.

Table 7: Pin Definition of UART3 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO10	27	IO	GPIO_12	**UART3_TX**	/	/	1.8 V power domain.
GPIO11	28	IO	GPIO_13	**UART3_RX**	/	/	1.8 V power domain.

NOTE

In QuecOpen application, QAPI_UART_PORT_003_E can be used to select and configure UART3.

6.1.3. I2C Interfaces

BG96 QuecOpen module provides two Inter-Integrated Circuit (I2C) interfaces for data communication. The interfaces support high-speed mode, and do not support multi-master.

I2C interface pins of the module are open-drain outputs multiplexed from GPIOs, and pull-up resistors should be provided externally.

The following table shows the pin definition of I2C interfaces.

Table 8: Pin Definition of the I2C1 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO19	40	IO	GPIO_19	/	/	**I2C1_SCL**	1.8 V power domain.
GPIO20	41	IO	GPIO_18	/	/	**I2C2_SDA**	1.8 V power domain.

NOTE

In QuecOpen application, QAPI_I2CM_INSTANCE_004_E can be used to select and configure I2C1.

Table 9: Pin Definition of the I2C2 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO05	18	IO	GPIO_11	/	SPI2_CLK	**I2C2_SCL**	1.8 V power domain.

GPIO06	19	IO	GPIO_10	/	SPI2_CS_N	**I2C2_SDA**	1.8 V power domain.

NOTE

In QuecOpen application, QAPI_I2CM_INSTANCE_005_E can be used to select and configure I2C2.

6.1.4. SPI Interfaces

BG96 QuecOpen module provides two SPI interfaces which support only master mode with a maximum clock frequency up to 50 MHz.

The following table shows the pin definition of SPI interfaces.

Table 10: Pin Definition of the SPI1 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO01	4	IO	GPIO_23	GPIO_23	**SPI1_CLK**	/	1.8 V power domain. Cannot be pulled up before startup.
GPIO02	5	IO	GPIO_20	UART1_TX	**SPI1_MOSI**	/	1.8 V power domain.
GPIO03	6	IO	GPIO_21	UART1_RX	**SPI1_MISO**	/	1.8 V power domain.
GPIO04	7	IO	GPIO_22	GPIO_22	**SPI1_CS_N**	/	1.8 V power domain.

NOTE

In QuecOpen application, QAPI_SPIM_INSTANCE_6_E can be used to select and configure SPI1.

Table 11: Pin Definition of the SPI2 Interface

Pin Name	Pin No.	I/O	Mode 1	Mode 2	Mode 3	Mode 4	Remark
GPIO05	18	IO	GPIO_11	/	**SPI2_CLK**	I2C2_SCL	1.8 V power domain.
GPIO06	19	IO	GPIO_10	/	**SPI2_CS_N**	I2C2_SDA	1.8 V power domain.
GPIO07	22	IO	GPIO_09	UART2_RX	**SPI2_MISO**	/	1.8 V power domain.

GPIO08	23	IO	GPIO_08	UART2_TX	**SPI2_MOSI**	/	1.8 V power domain.

NOTE

In QuecOpen application, QAPI_SPIM_INSTANCE_5_E can be used to select and configure SPI2.

7 Appendix A References

Table 12: Related Documents

SN	Document Name	Remark
[1]	Quectel_BG96_QEFS_Explorer_User_Guide	User guide of BG96 QEFS Explorer tool
[2]	Quectel_BG96_QuecOpen_Hardware_Design	Hardware design manual of BG96 module in QuecOpen solution
[3]	Quectel_BG96_DFOTA_User_Guide	BG96 DFOTA user guide
[4]	Quectel_BG96_GNSS_Application_Note	BG96 GNSS application note
[5]	Quectel_BG96_AT_Commands_Manual	BG96 AT commands manual

Table 13: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
DOS	Disk Operating System
GPIO	General-Purpose Input/Output
HTTP(S)	Hyper Text Transfer Protocol (Secure)
MCU	Microcontroller Unit
OS	Operating System
QAPI	Qualcomm™ Application Programming Interface
RAM	Random Access Memory
ROM	Read Only Memory
SDK	Software Development Kit
SPI	Serial Peripheral Interface

8 Appendix B Compiling Environment Setup

Quectel BG96 QuecOpen module 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 14: 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 15: 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. The tool can also be requested from Quectel.

8.1. Setup Compiling Environment for TX2.0

8.1.1. Download and Install ARM Compiler Tool

Before installing the ARM compiler tool, contact an ARM agent/distributor to obtain the installation software and license.

8.1.2. Download and Install Cygwin

8.1.2.1. Download Cygwin

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

8.1.2.2. Install Cygwin

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

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

Figure 6: Cygwin Setup Program

Step 2: Choose the installation type.

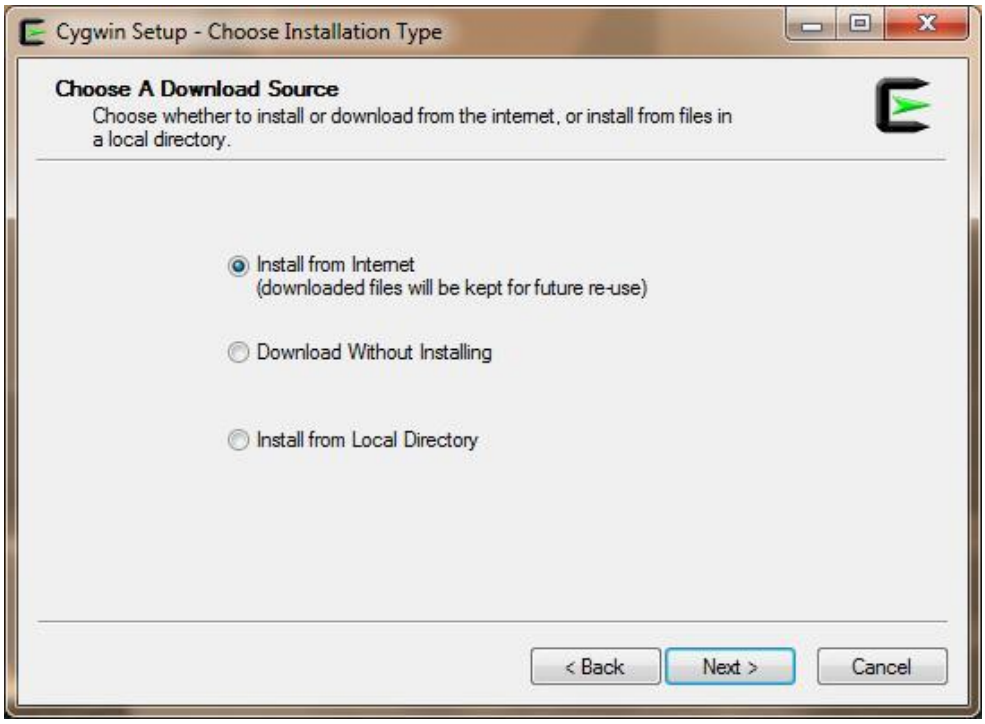

Figure 7: Choose Installation Type

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

Figure 8: Choose Installation Directory and Parameters

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

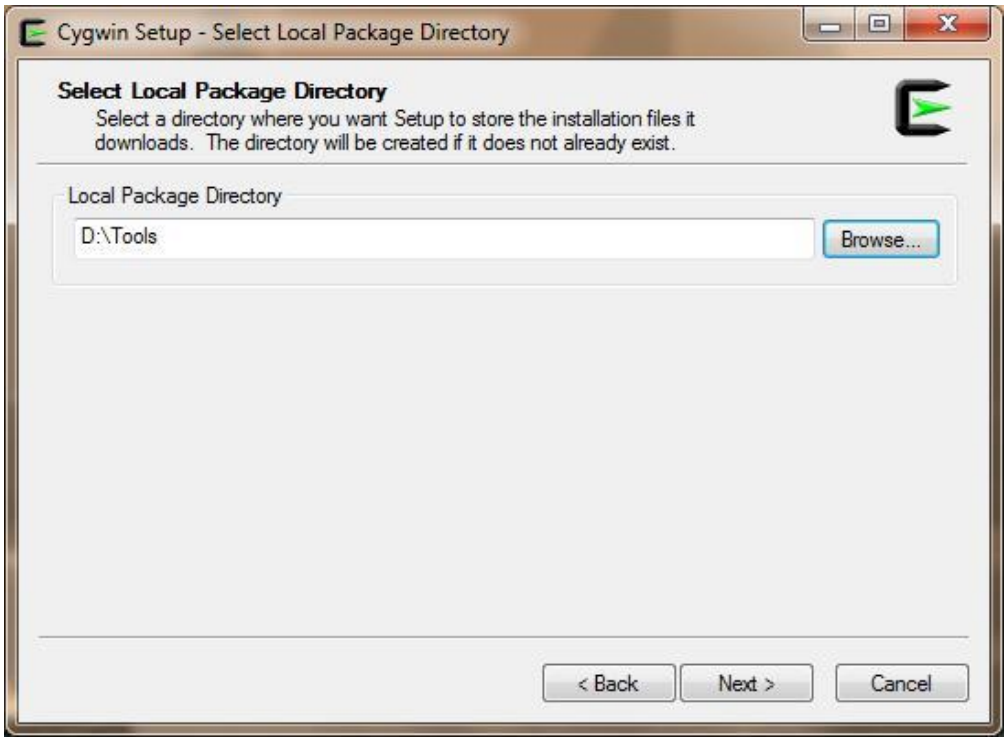

Figure 9: Select Local Package Directory

Step 5: Select the type of internet connection.

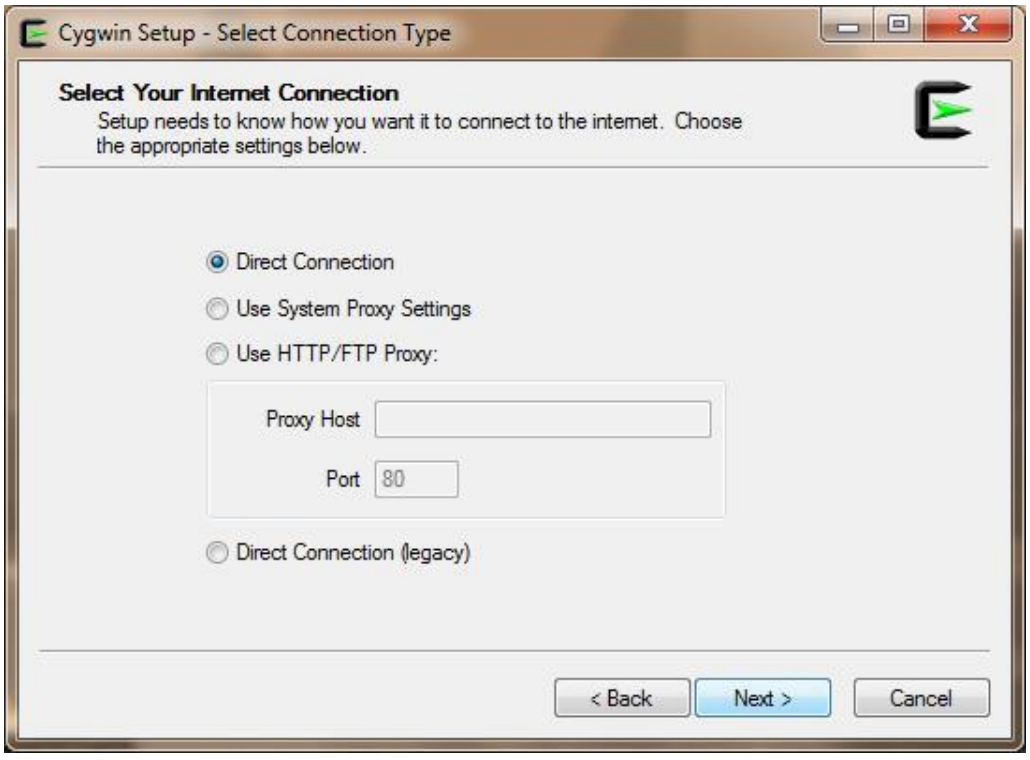

Figure 10: Select Internet Connection Type

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

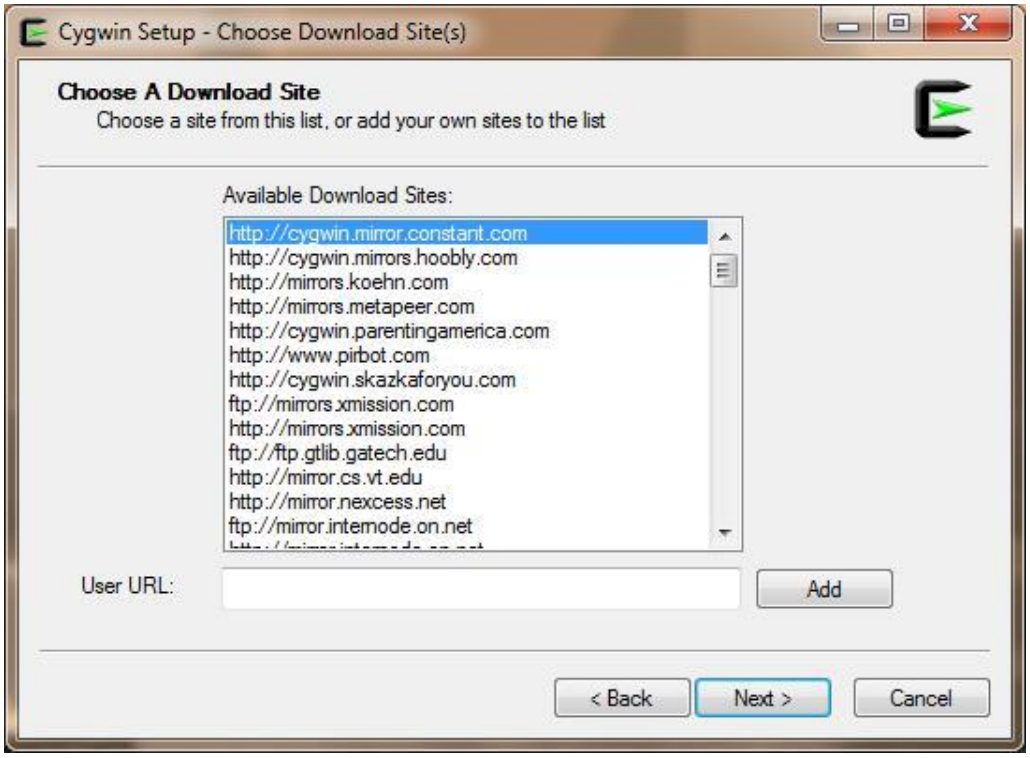

Figure 11: Choose a Download Site

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

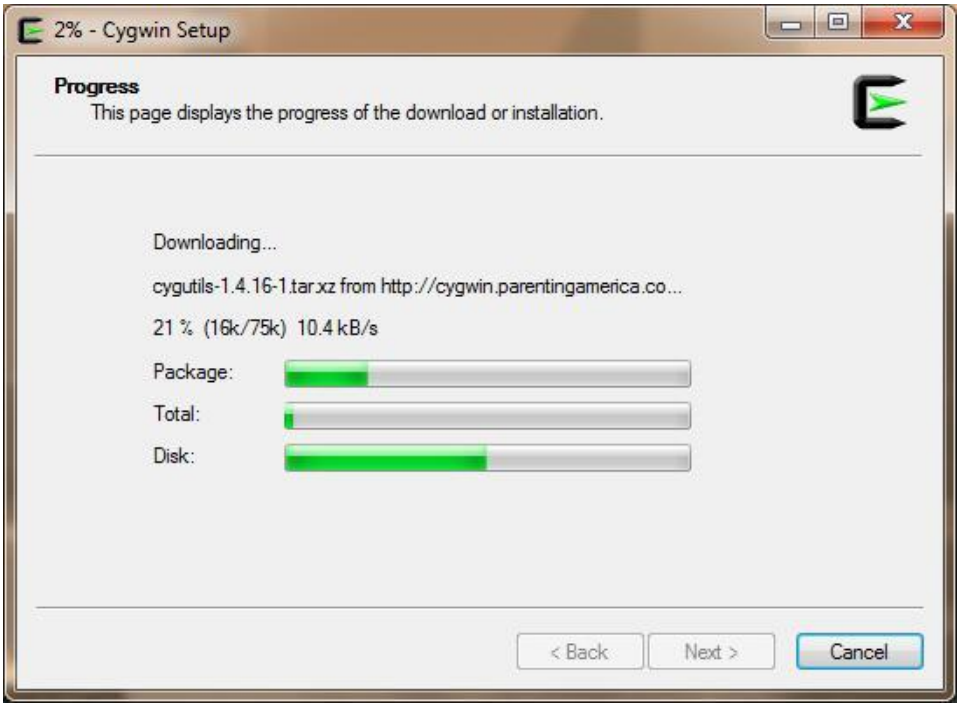

Figure 12: Process of Download or Installation

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

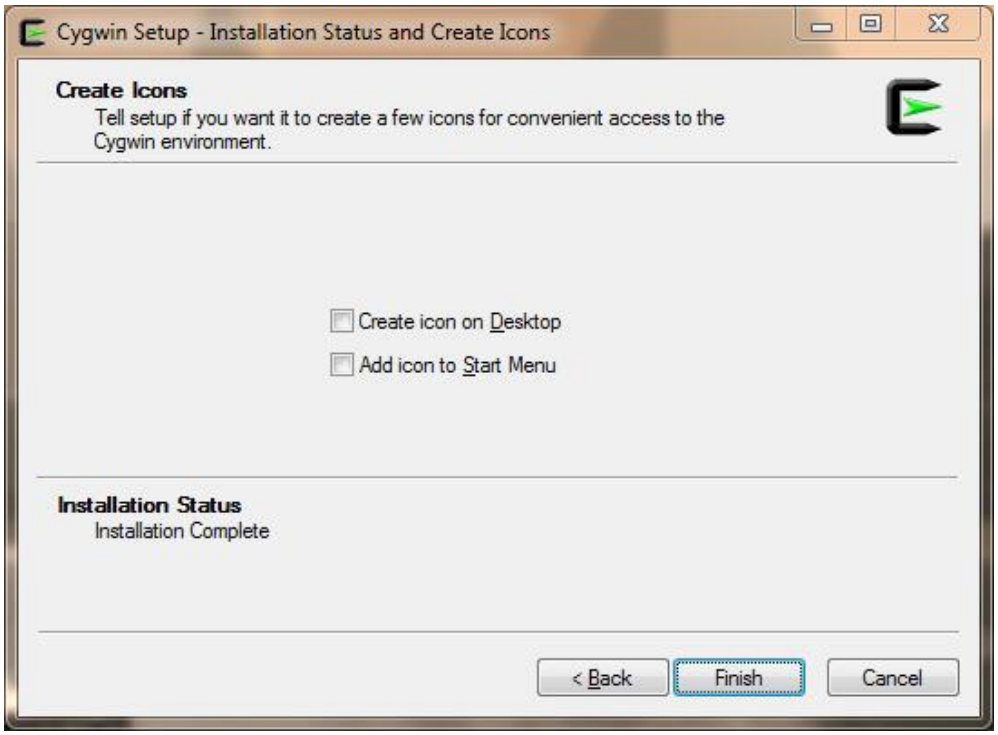

Figure 13: Create Icon and Complete Installation

After successful installation of both ARM compiler tool and Cygwin, compiling of QuecOpen SDK can be started. See ***Chapter 3*** and ***Chapter 4*** for details about compilation and running of QuecOpen SDK.

8.2. Setup Compiling Environment for TX3.0.1

8.2.1. Download LLVM

Licensed users can download LLVM compiler through the Qualcomm ChipCode™ portal. If no license is available, contact Quectel Technical Supports.

8.2.1.1. For Windows 7 Build Hosts

Figure 14: LLVM Download for Windows 7 Build Hosts

8.2.1.2. For Linux Build Hosts

Figure 15: LLVM Download for Linux Build Hosts

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

8.2.2. Download and Install Cygwin

See ***Chapter 8.1.2*** for details.

8.2.3. Download and Install Python

8.2.3.1. 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/.

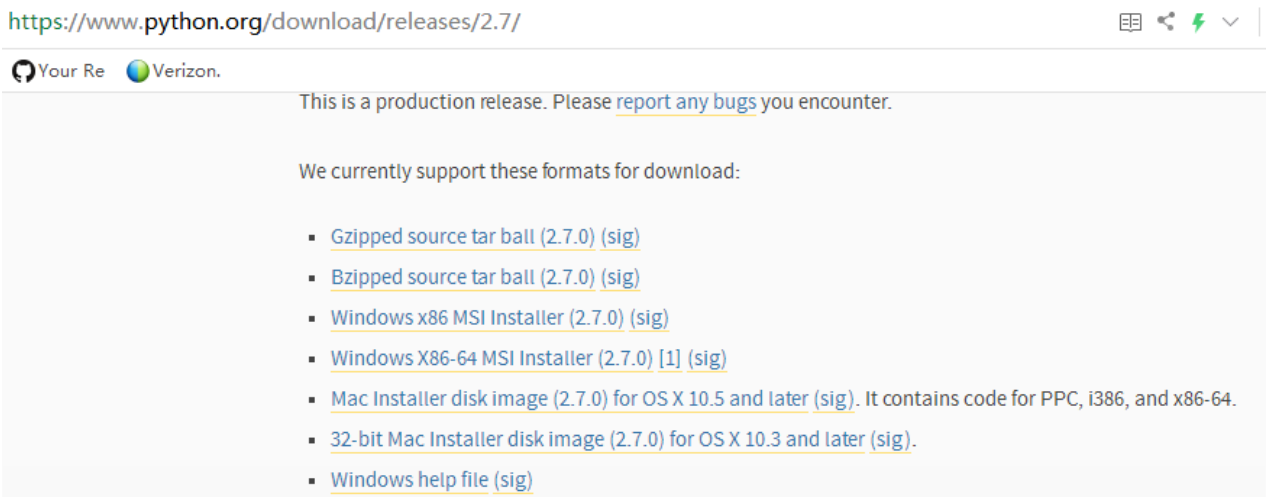

Figure 16: Python Download Page Screenshot

Download x86/x86-64 versions as needed.

8.2.3.2. Install Python

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

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

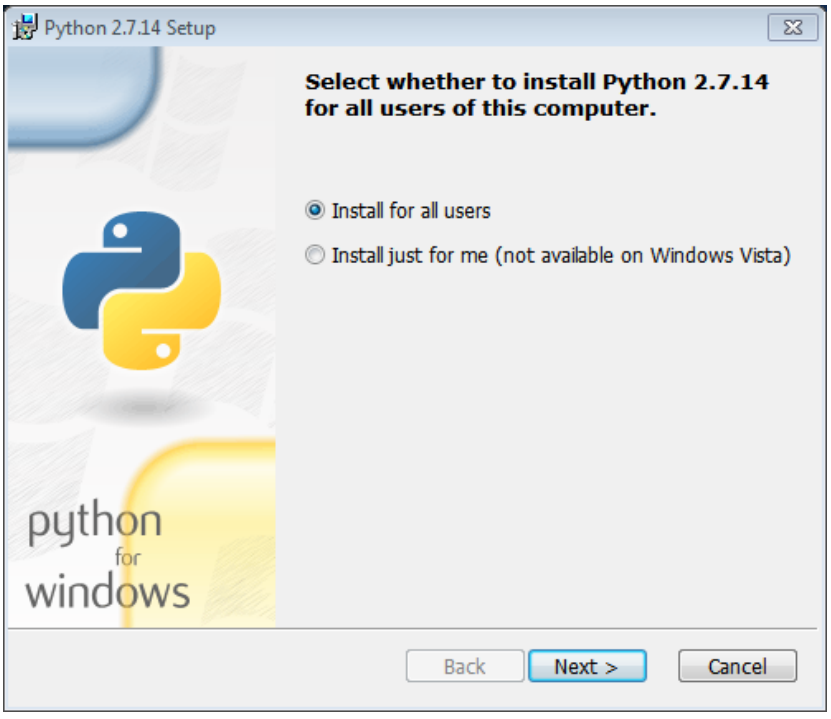

Figure 17: Python Setup

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

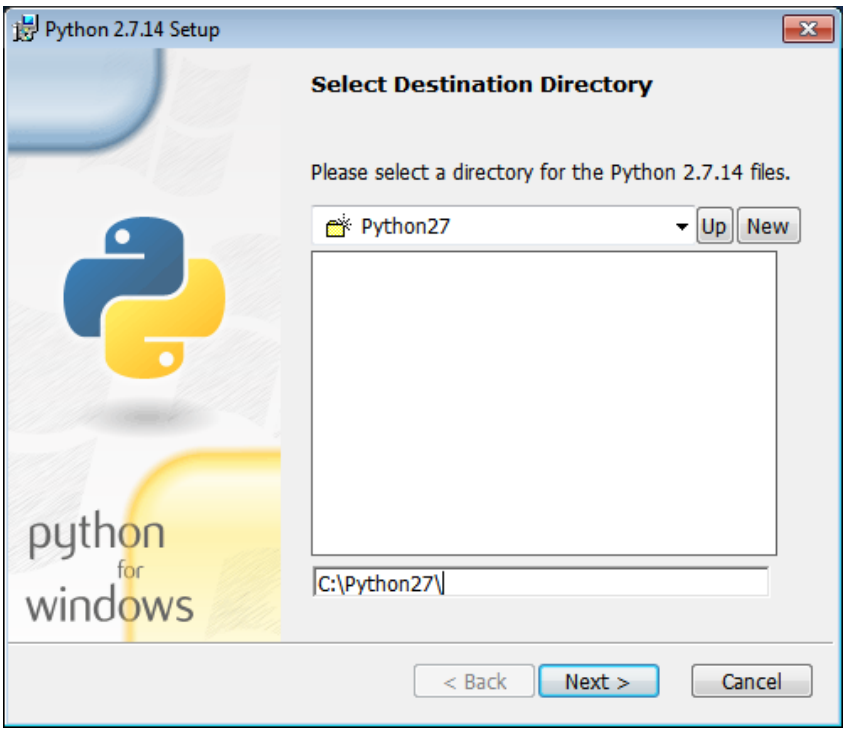

Figure 18: Select Installation Directory

Step 3: Options for customization. Keep the default options.

Figure 19: Options for Customization

Step 4: Wait during installation process.

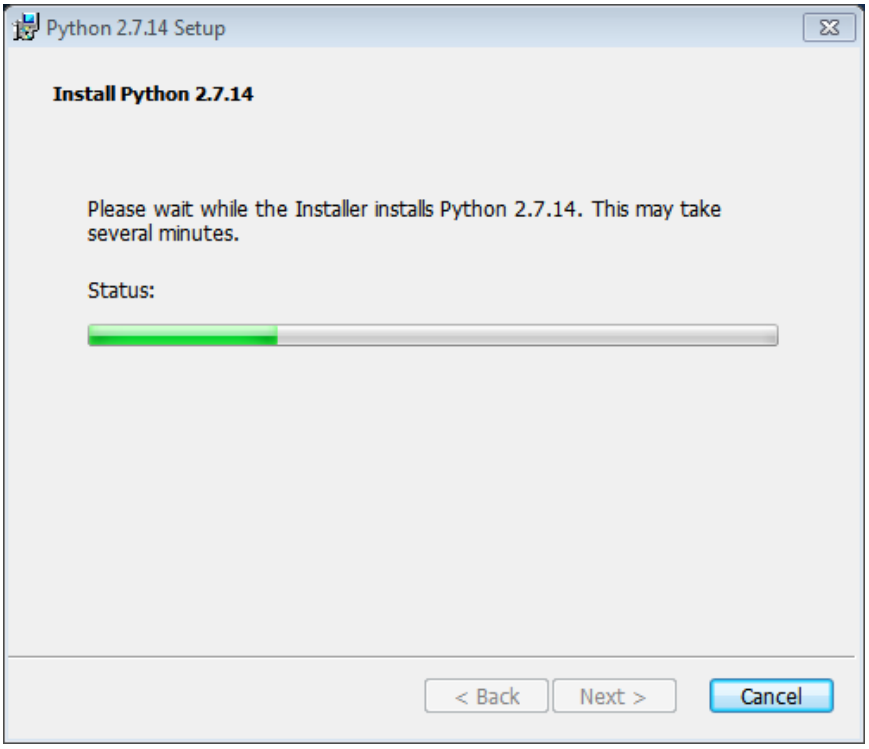

Figure 20: Installing

Step 5: Complete installation.

Figure 21: Installation Completed