

BG96 ThreadX DAM Application Note

LTE Module Series

Rev. BG96_ThreadX_DAM_Application_Note_V1.0

Date: 2017-08-03



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

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2017. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2017-08-03	Hyman DING	Initial

Quectel
Preliminary

Contents

About the Document.....	2
Contents.....	3
Table Index.....	4
Figure Index.....	5
1 Introduction	6
2 Overview of ThreadX with DAMs.....	7
2.1. General Overview	7
2.2. Architecture of ThreadX with DAMs.....	7
3 Setup Compiling Environment	9
3.1. Download and Install ARM Compiler Tool.....	9
3.1.1. Download ARM Compiler Tool	9
3.1.2. Install ARM Compiler Tool	12
3.2. Download and Install Cygwin.....	15
3.2.1. Download Cygwin	15
3.2.2. Install Cygwin	15
4 How to Build ThreadX DAM SDK.....	21
4.1. Quectel ThreadX DAM SDK Package	21
4.2. Build ThreadX DAM Application.....	22
5 How to Run ThreadX DAM	23
6 Appendix A References.....	24
7 Appendix B List of Header Files and Libraries.....	25

Table Index

TABLE 1: REQUIREMENT FOR COMPILING ENVIRONMENT	9
TABLE 2: DESCRIPTION OF THREADX DAM SDK PACKAGE DIRECTORIES	21
TABLE 3: RELATED DOCUMENTS	24
TABLE 4: TERMS AND ABBREVIATIONS	24
TABLE 5: LIST OF HEADER FILES	25
TABLE 6: LIST OF LIBRARIES	26

Quectel
Preliminary

Figure Index

FIGURE 1: ARCHITECTURE OF THREADX WITH DAMS	8
FIGURE 2: “DOWNLOADS” AND “DEVELOPMENT TOOLS” PAGES	10
FIGURE 3: CLICK “DS-5 DEVELOPMENT STUDIO”	10
FIGURE 4: DOWNLOAD THE CORRESPONDING TOOL.....	11
FIGURE 5: CONFIRMATION OF DETAILS.....	11
FIGURE 6: ARM COMPILER 5 SETUP	12
FIGURE 7: END-USER LICENSE AGREEMENT	13
FIGURE 8: CUSTOM SETUP.....	13
FIGURE 9: “SYSTEM PENDING REBOOT” WARNING.....	14
FIGURE 10: READY TO INSTALL ARM COMPILER 5.....	14
FIGURE 11: FINISH INSTALLATION OF ARM COMPILER TOOL.....	15
FIGURE 12: CYGWIN SETUP PROGRAM.....	16
FIGURE 13: CHOOSE INSTALLATION TYPE	16
FIGURE 14: CHOOSE INSTALLATION DIRECTORY AND PARAMETERS	17
FIGURE 15: SELECT LOCAL PACKAGE DIRECTORY	17
FIGURE 16: SELECT INTERNET CONNECTION TYPE	18
FIGURE 17: CHOOSE A DOWNLOAD SITE	18
FIGURE 18: PROCESS OF DOWNLOAD OR INSTALLATION	19
FIGURE 19: CREATE ICON AND COMPLETE INSTALLATION	19
FIGURE 20: FOLDER STRUCTURE OF QUECTEL THREADX DAM SDK PACKAGE	21

1 Introduction

Quectel BG96 module supports ThreadX with Downloadable Application Modules (ThreadX with DAMs) which is realized through building and running ThreadX DAM SDK package in ThreadX OS as DAM.

This document describes the structure of *Quectel_ThreadX_DAM_SDK_Package*, and also introduces how to build and run the package in ThreadX OS as DAM.

Quectel
Preliminary

2 Overview of ThreadX with DAMs

2.1. General Overview

ThreadX with DAMs 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. Architecture of ThreadX with DAMs

The following diagram shows the architecture of ThreadX with DAMs.

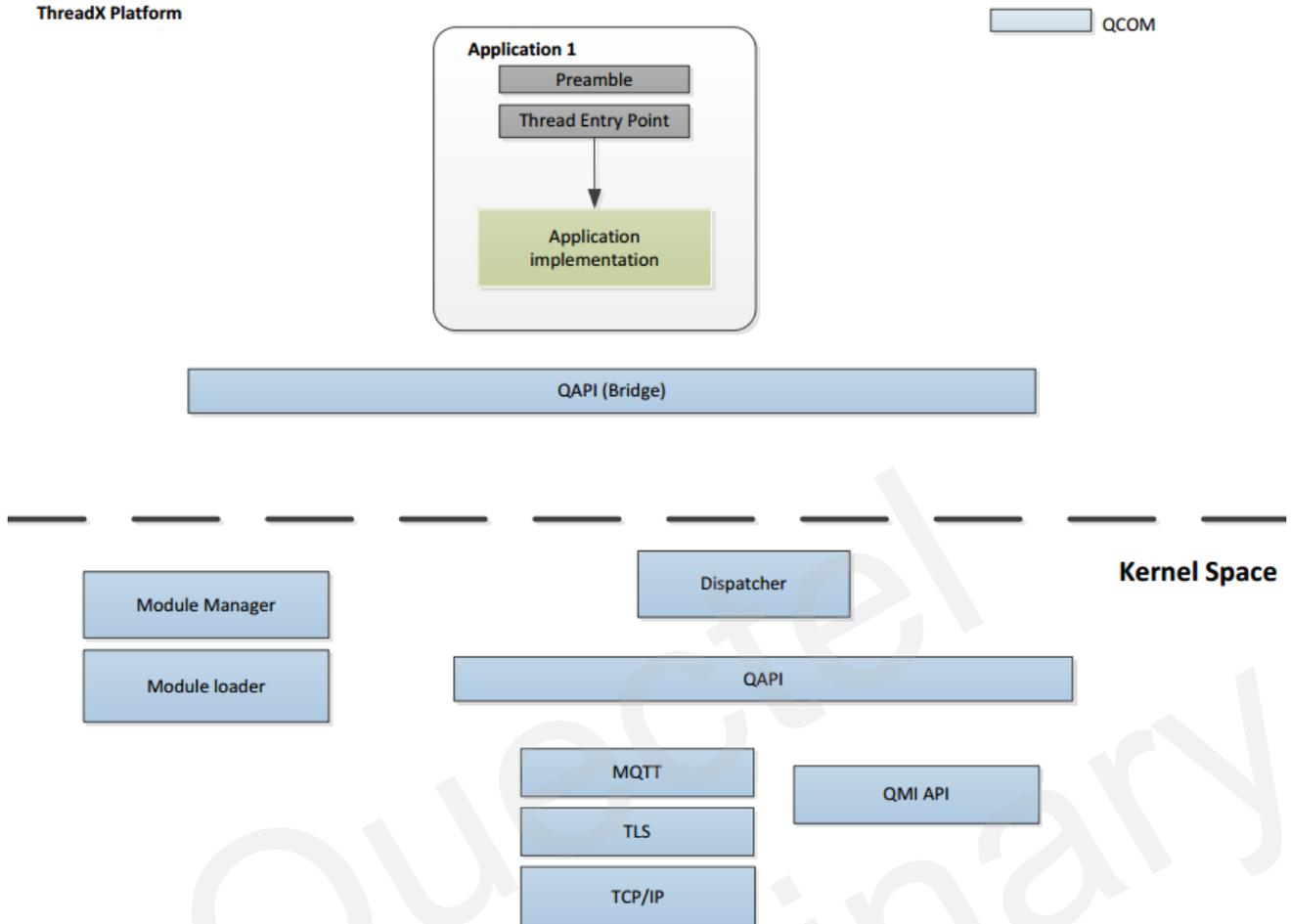


Figure 1: Architecture of ThreadX with DAMs

3 Setup Compiling Environment

While compiling ThreadX DAM application, the host's operating system and compilation tools should meet the requirements shown below.

Table 1: Requirement for Compiling Environment

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

3.1. 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. 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:

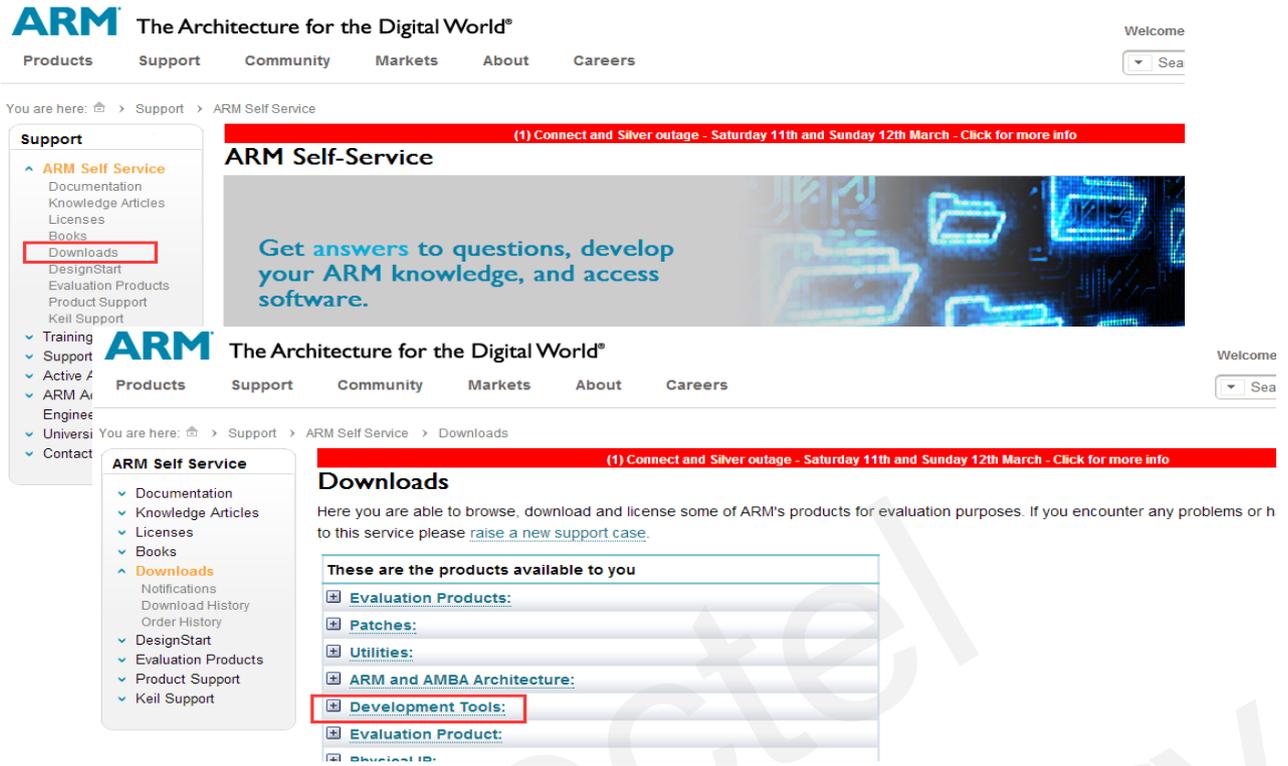


Figure 2: “Downloads” and “Development Tools” Pages

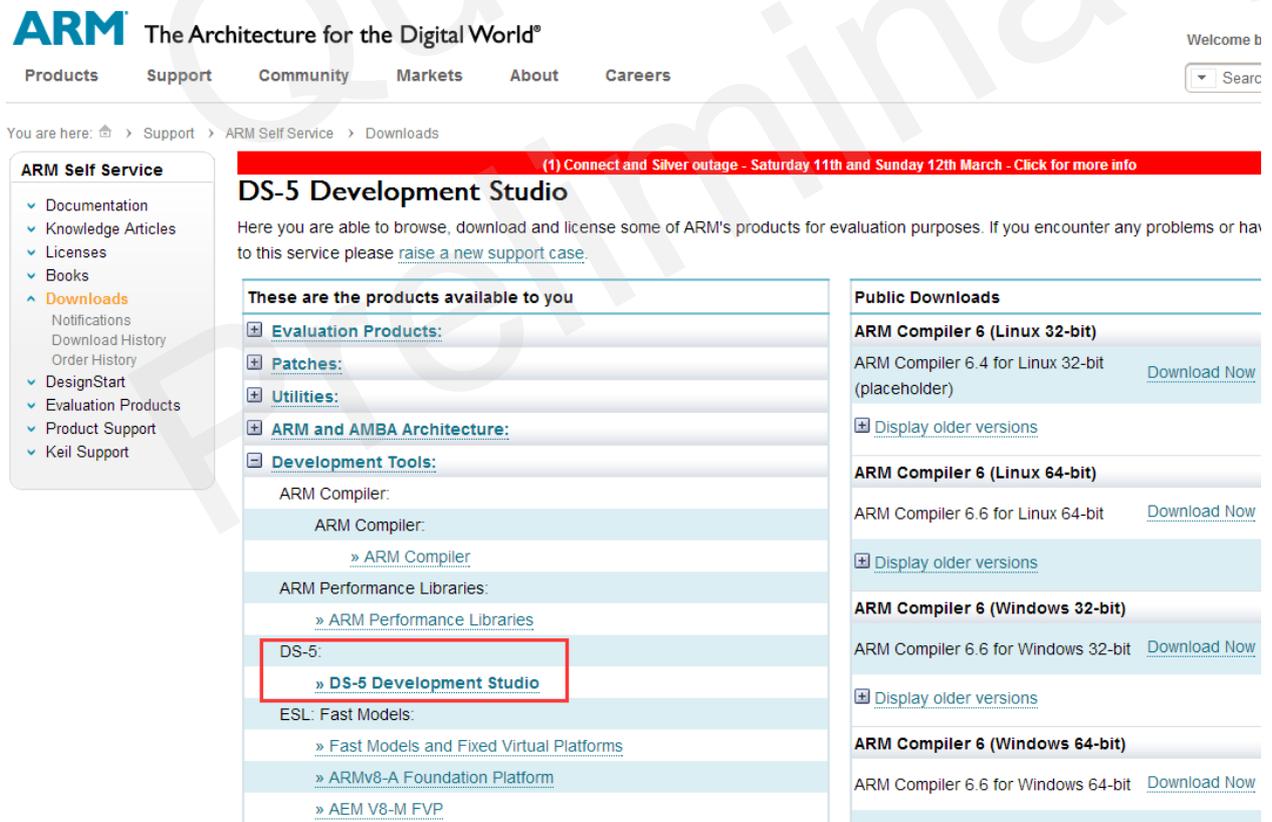


Figure 3: Click “DS-5 Development Studio”

(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.

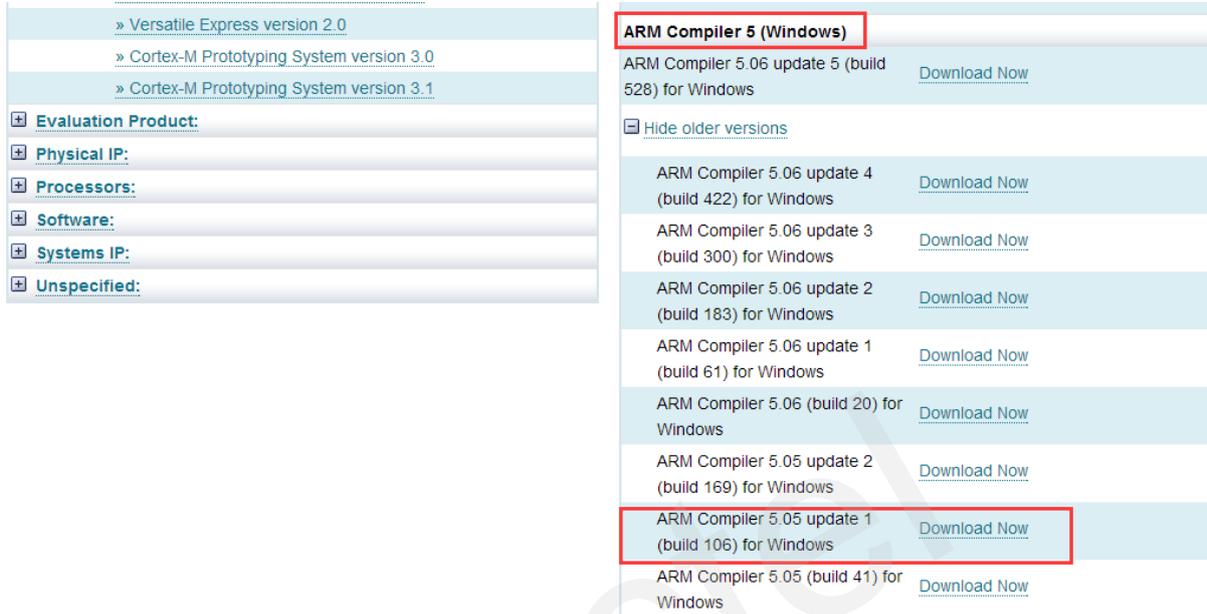


Figure 4: Download the Corresponding Tool

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

I Agree *

Please confirm the details listed below

Address:

Town/City:

State/Country:

Zip/Postal Code:

Country:

Email: ql_arm_01@126.com

Telephone Number:

Fax:

Job Title:

Reason:

I Agree for ARM to contact me with related information for these products *

Note: * indicates a mandatory field

Figure 5: Confirmation of Details

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

3.1.2. 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**”.

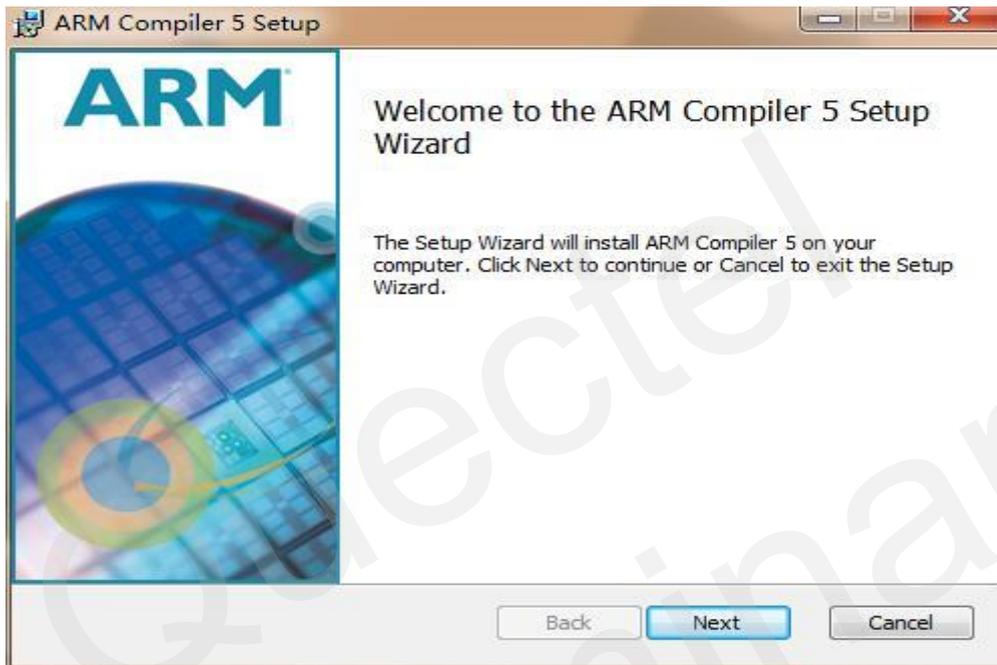


Figure 6: ARM Compiler 5 Setup

Step 2: Accept the terms in the license agreement and then click “Next”.

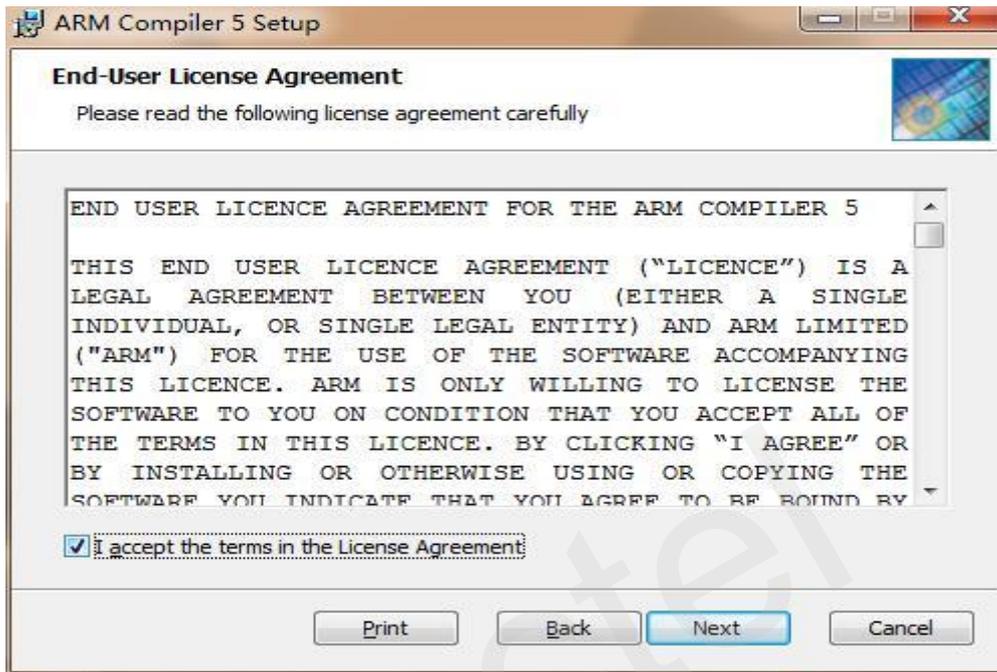


Figure 7: End-User License Agreement

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

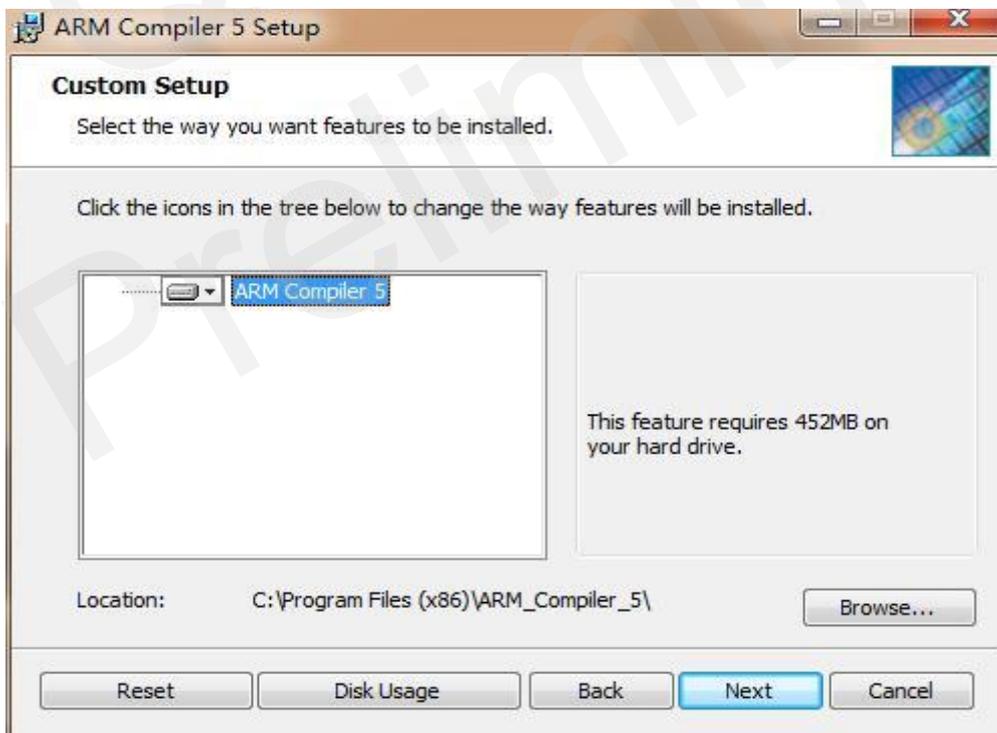


Figure 8: Custom Setup

Step 4: Ignore “System Pending Reboot” warning, and click “Next”.

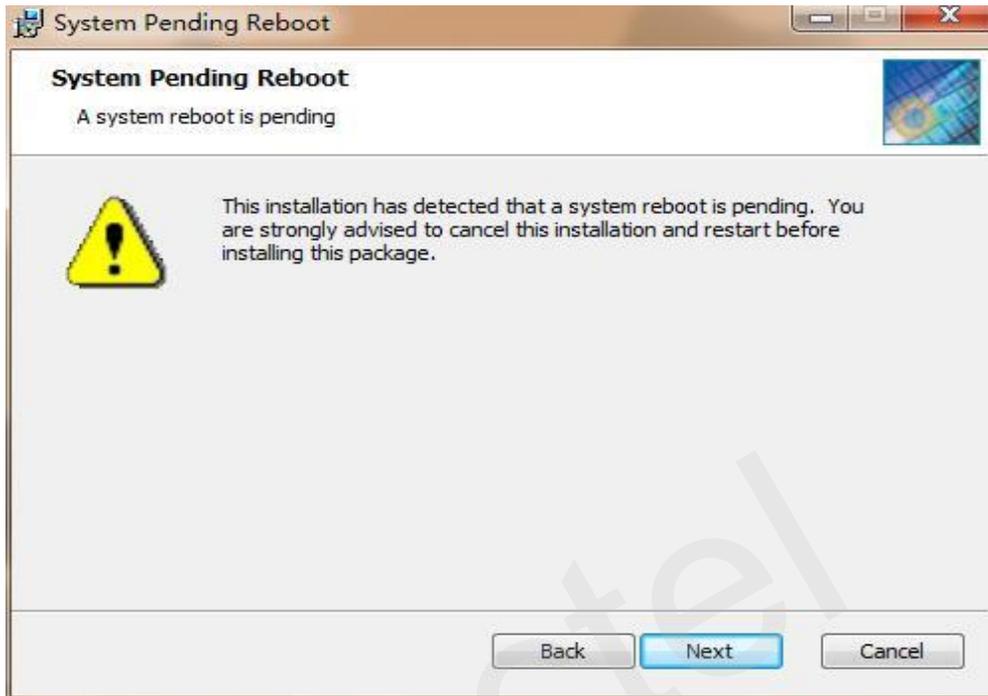


Figure 9: “System Pending Reboot” Warning

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

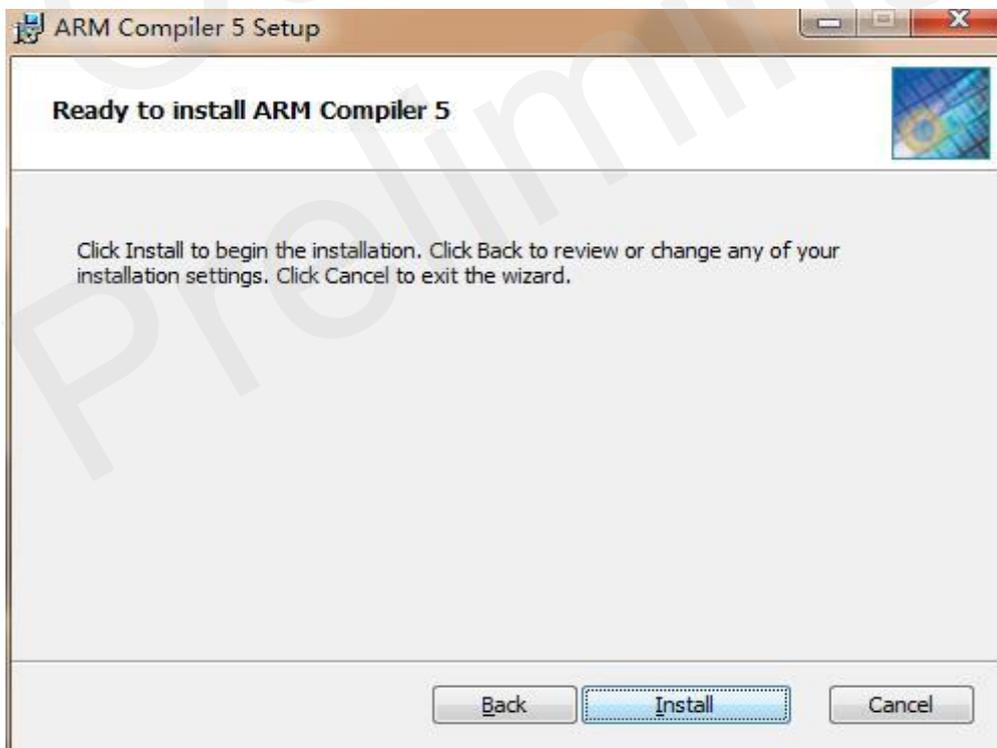


Figure 10: Ready to install ARM Compiler 5

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

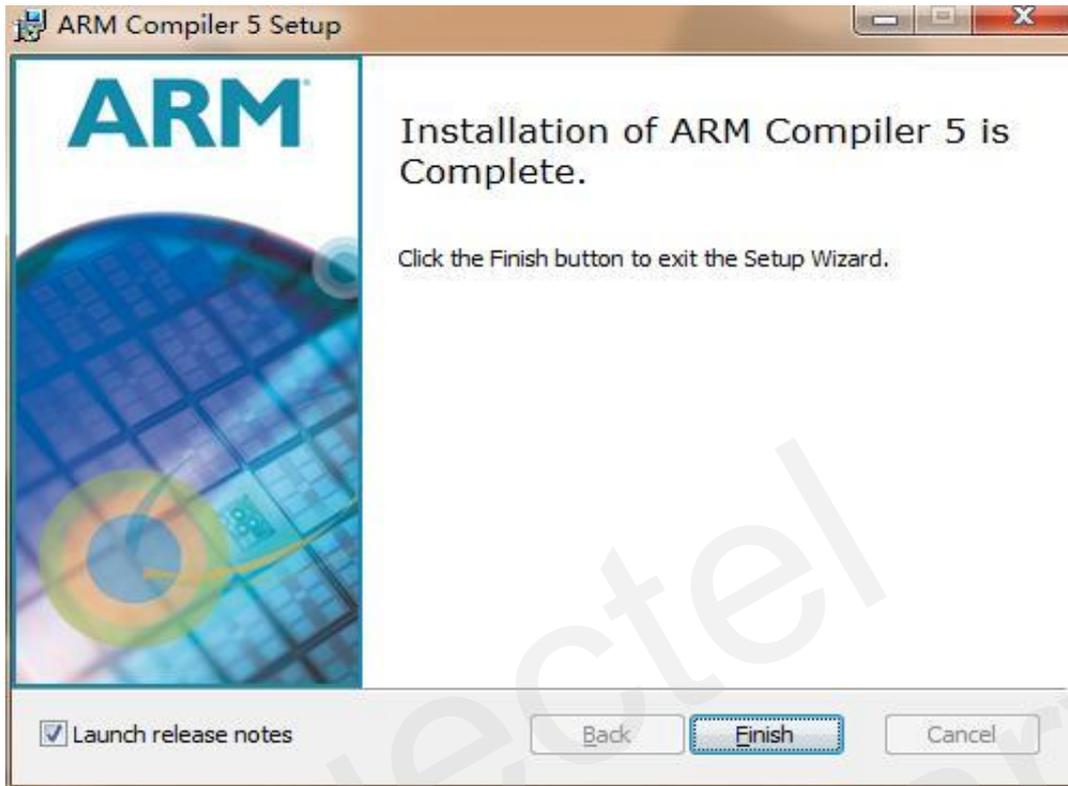


Figure 11: Finish Installation of ARM Compiler Tool

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

3.2. Download and Install Cygwin

3.2.1. Download Cygwin

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

3.2.2. Install Cygwin

To install the environment where you can compile the ThreadX DAM 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.

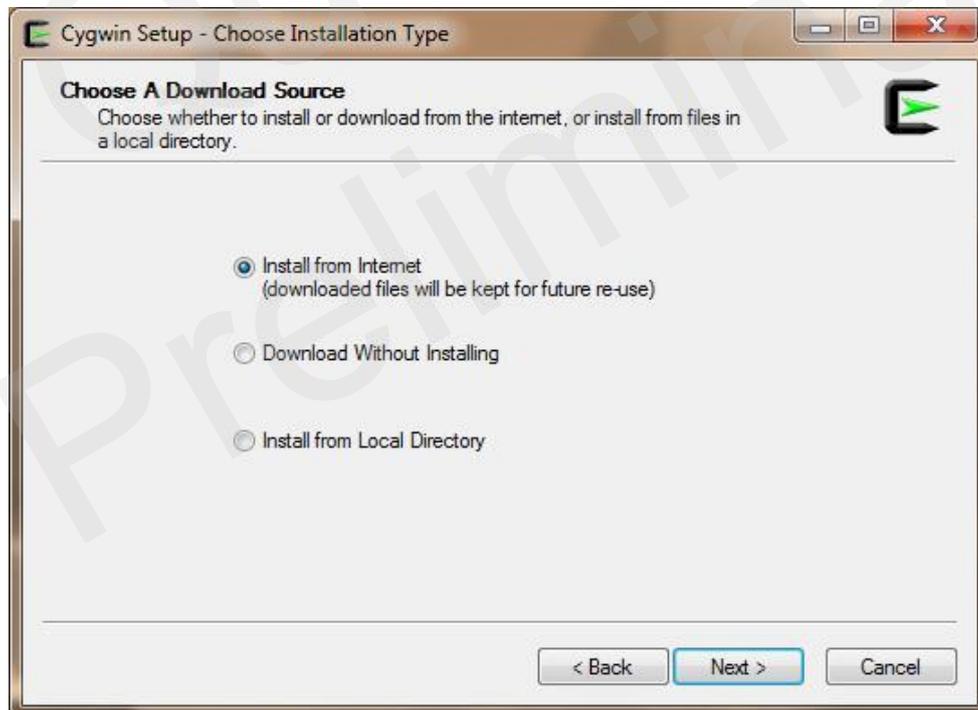


Figure 13: Choose Installation Type

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.

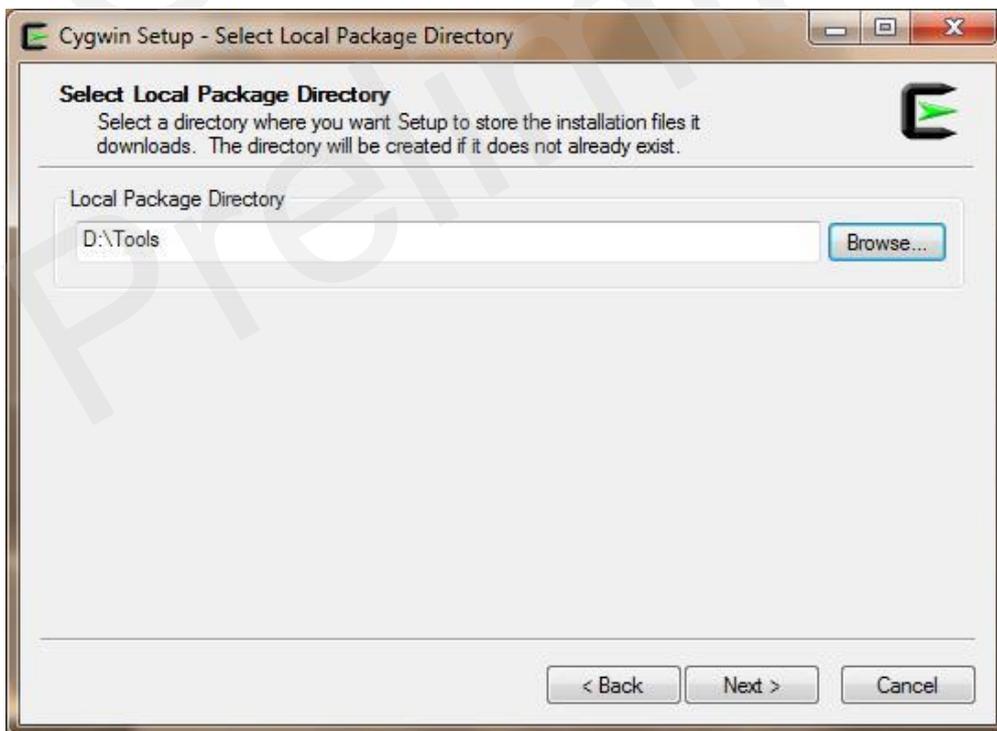


Figure 15: Select Local Package Directory

Step 5: Select the type of internet connection.

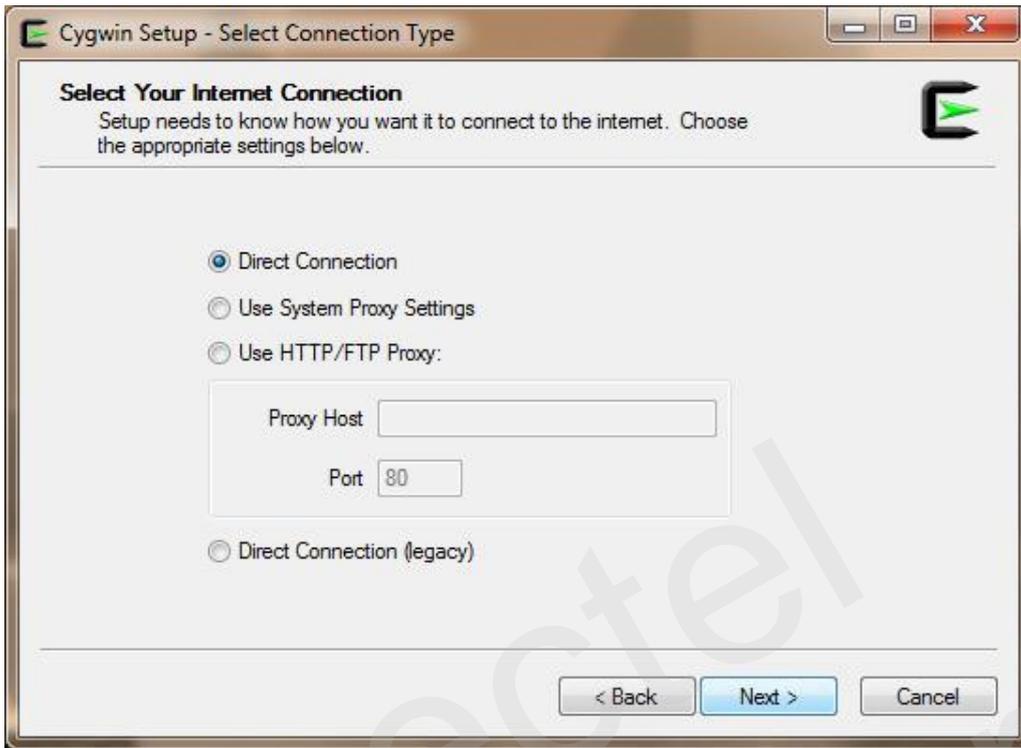


Figure 16: Select Internet Connection Type

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

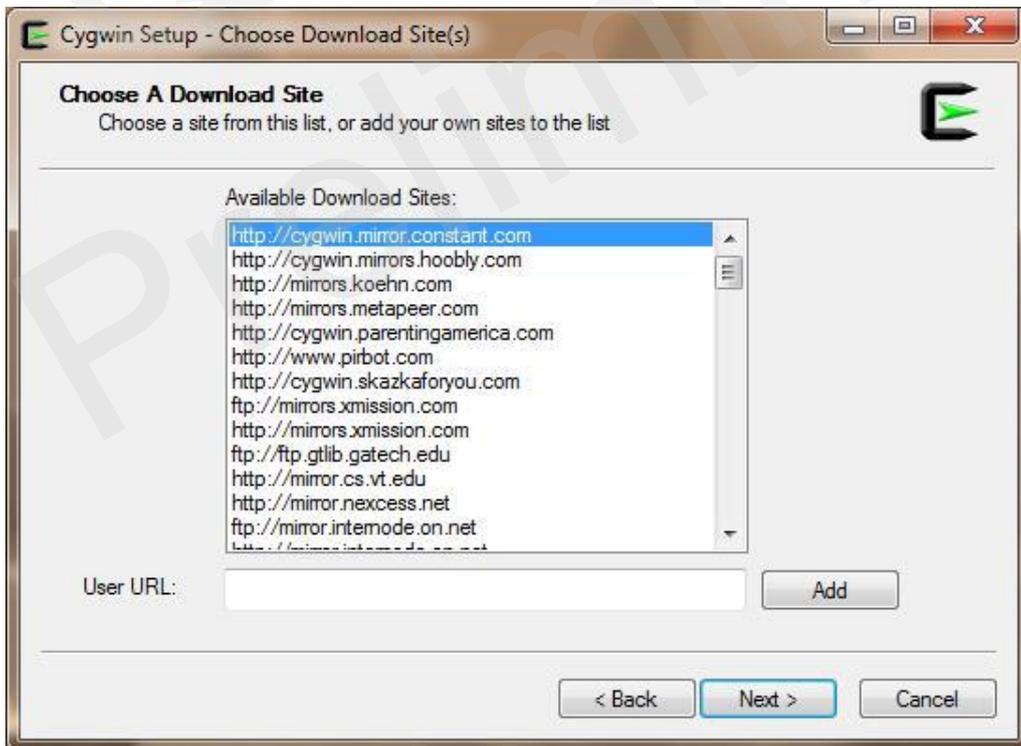


Figure 17: Choose a Download Site

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

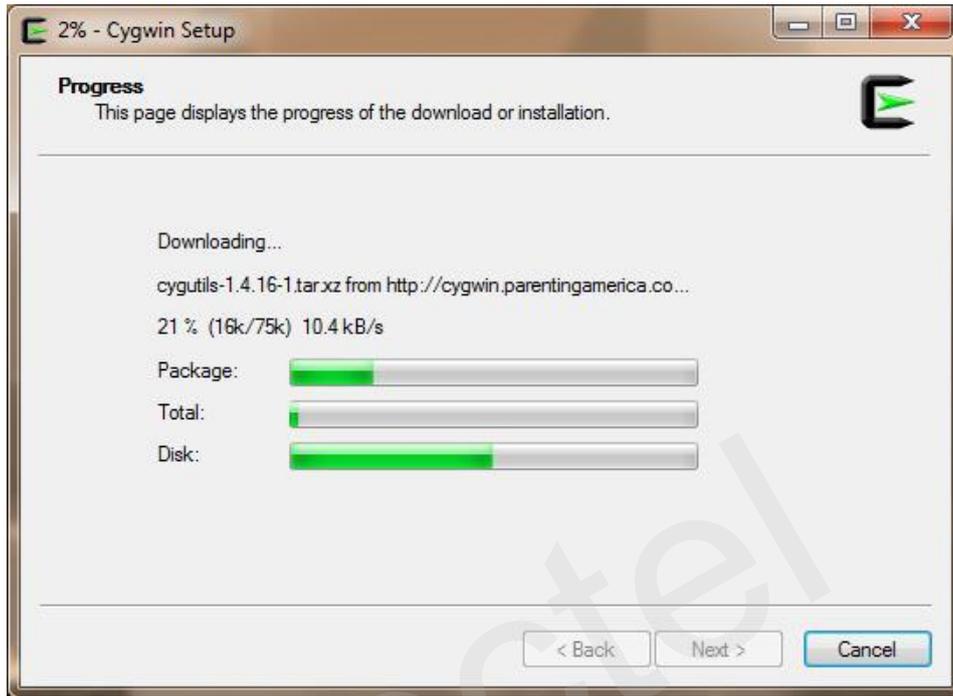


Figure 18: Process of Download or Installation

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

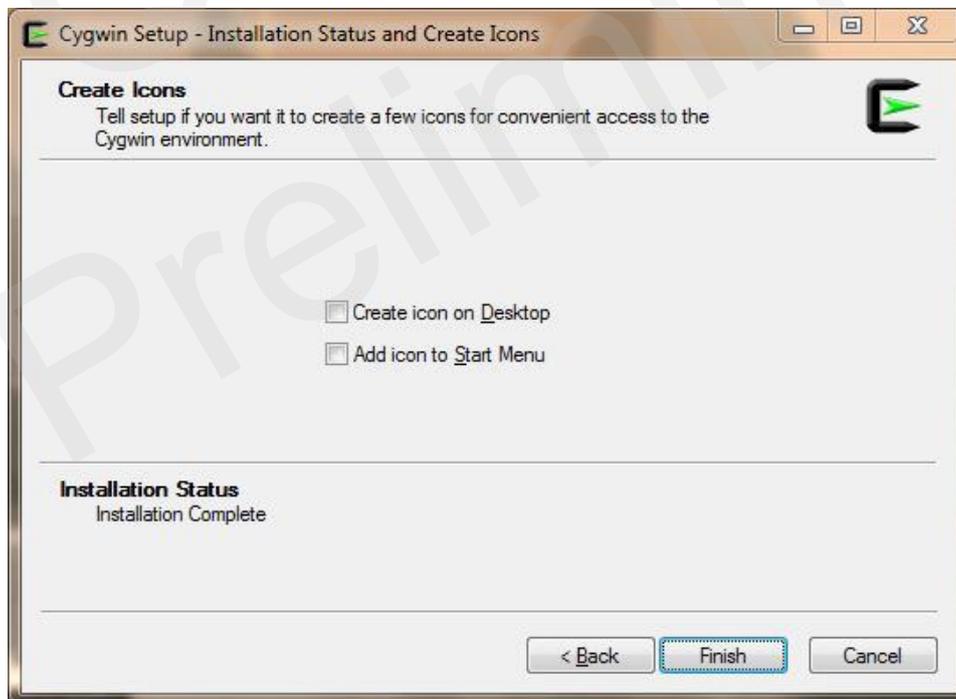


Figure 19: Create Icon and Complete Installation

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

Quectel
Preliminary

4 How to Build ThreadX DAM SDK

4.1. Quectel ThreadX DAM SDK Package

The following shows the folder structure of *Quectel_ThreadX_DAM_SDK_Package* which is created for non-licensed customers.

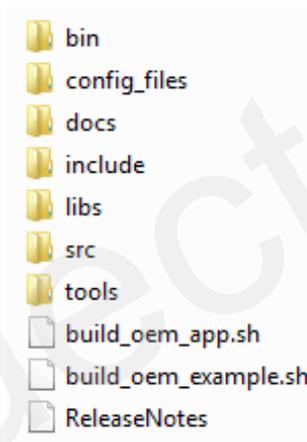


Figure 20: Folder Structure of Quectel ThreadX DAM SDK Package

Table 2: Description of ThreadX DAM SDK Package Directories

Directories	Description/Function	Remark
<i>bin</i>	Application gets created in this folder after successful compilation.	
<i>config_files</i>	Contains application related configuration file: <i>oem_app_path.ini</i>	
<i>docs</i>	Guide documents	
<i>Include</i>	Header files needed for compilation provided by Quectel.	Please refer to Appendix B for the complete list of header files and libraries.
<i>libs</i>	Required libraries should be copied here.	
<i>src</i>	Application source code.	
<i>tools</i>	Tools for development.	

4.2. Build ThreadX DAM Application

To build the *Quectel_ThreadX_DAM_SDK_Package* as DAM, customers just need to run the following command from command line in Cygwin:

New build:

```
./build_oem_app.sh
```

Clean build:

```
./build_oem_app.sh -c
```

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

NOTE

ThreadX DAM application build process is recommended for the off-target ThreadX DAM application development, as in this mode the application can be deployed and updated separately from the image.

Quectel
Preliminary

5 How to Run ThreadX DAM

To run the ThreadX DAM application binary file (*oem_app.bin*), customers only need upload *oem_app.bin* 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 *oem_app.bin*.

After uploading these two files into alternate file systems, restart BG96 and the DAM image will be uploaded and started by the Module Loader.

NOTE

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

6 Appendix A References

Table 3: Related Documents

SN	Document Name	Remark
[1]	Quectel_ThreadX_and_QAPI_Application_User_Guide	Introduce how to set up ThreadX compiler environment for Windows, and how to use QAPI to realize customer requirements with ThreadX OS.
[2]	Quectel_BG96_QEFS_Explorer_User_Guide	Introduce how to use the QEFS Explorer tool to access alternate file systems.

Table 4: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
DAM	Downloadable Application Module
MQTT	Message Queuing Telemetry Transport
OS	Operating System
QMI	Qualcomm MSM Interface
SDK	Software Development Kit
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security

7 Appendix B List of Header Files and Libraries

Table 5: List of Header Files

SN	Header Files	SN	Header Files
[1]	msgcfg.h	[36]	qapi_psm_types.h
[2]	msgtgt.h	[37]	qapi_ril_base.h
[3]	msg_mask.h	[38]	qapi_socket.h
[4]	qapi.h	[39]	qapi_spi_master.h
[5]	qapi_adc.h	[40]	qapi_ssl.h
[6]	qapi_adc_types.h	[41]	qapi_status.hqapi_timer.h
[7]	qapi_dam_buses.h	[42]	qapi_sensors.h
[8]	qapi_data_txm_base.h	[43]	qapi_timer_id.h
[9]	qapi_device_info.h	[44]	qapi_tlmm.h
[10]	qapi_diag.h	[45]	qapi_tsens.h
[11]	qapi_diag_codes.h	[46]	qapi_tsens_types.h
[12]	qapi_diag_types.h	[47]	qapi_txm_base.h
[13]	qapi_dnsc.h	[48]	qapi_types.h
[14]	qapi_dss.h	[49]	qapi_uart.h
[15]	qapi_fs.h	[50]	qapi_usb.h
[16]	qapi_fs_types.h	[51]	qapi_usb_types.h
[17]	qapi_ftl.h	[52]	fx_api.h

[18]	qapi_ftl_types.h	[53]	txm_module.h
[19]	qapi_gpoint.h	[54]	txm_module_port.h
[20]	qapi_httpc.h	[55]	tx_api.h
[21]	qapi_i2c_master.h	[56]	tx_block_pool.h
[22]	qapi_location.h	[57]	tx_byte_pool.h
[23]	qapi_location_txm.h	[58]	tx_event_flags.h
[24]	qapi_lwm2m.h	[59]	tx_initialize.h
[25]	qapi_mqtt.h	[60]	tx_low_power.h
[26]	qapi_netbuf.h	[61]	tx_mutex.h
[27]	qapi_netprofile.h	[62]	tx_port.h
[28]	qapi_netservices.h	[63]	tx_queue.h
[29]	qapi_net_status.h	[64]	tx_semaphore.h
[30]	qapi_ns_gen_v4.h	[65]	tx_thread.h
[31]	qapi_ns_gen_v6.h	[66]	tx_timer.h
[32]	qapi_ns_utils.h	[67]	tx_trace.h
[33]	qapi_pmapp_rtc.h	[68]	tx_user.h
[34]	qapi_psm.h	[69]	stringl.h
[35]	qapi_psm_status.h		

Table 6: List of Libraries

SN	Libraries
[1]	txm_lib.lib
[2]	timer_dam_lib.lib