



PERFECT WIRELESS EXPERIENCE

FIBOCOM MA510-GL-00-90 Series

FibocomOpen Application Guide

Version: V1.0.1

Date: 2021-02-19



Applicability Type

No.	Product Model	Description
1	MA510-GL-00-90	Cat M1/Cat NB2/EGPRS

FIBOCOM
Confidential



Copyright

Copyright © 2020 Fibocom Wireless Inc. All rights reserved.

Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or transmit the document in any form.

Notice

The document is subject to update from time to time owing to the product version upgrade or other reasons. Unless otherwise specified, the document only serves as the user guide. All the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Trademark



The trademark is registered and owned by Fibocom Wireless Inc.

FIBOCOM
Confidential

Change History

Version	Author	Reviewer	Approver	Date	Description
V1.0.1	Liu Ben	Zhu Gang	Zhang Jun	2021-02-19	Corrected errors
V1.0.0	Liu Ben	Zhu Gang	Zhang Jun	2020-12-17	Initial version

FIBOCOM
Confidential

Contents

1	Introduction	6
2	Fibocom Solution Overview	7
2.1	General Overview	7
2.2	Fibocom Architecture	7
3	Build Fibocom Applications	9
3.1	FibocomOpen SDK Package	9
3.2	Compile FibocomOpen Applications	10
3.3	Run FibocomOpen Applications.....	10
3.3.1	<i>Load Binary Images from Alternate File Systems</i>	<i>10</i>
3.3.2	<i>Use QFLOG to Load Binary Images to the Device</i>	<i>12</i>
3.3.3	<i>Use QFLOG to Debug</i>	<i>13</i>
4	GPIO Pin Mapping	14
4.1	GPIO Pin Mapping of MA510-GL-00-90 FibocomOpen	14
4.2	GPIOs	15
4.3	UART Interfaces	16
4.4	I2C Interface.....	17
5	Available Memory Mapping	19
6	Download Image	20
7	Appendixes	21
7.1	Appendix A Terms and Abbreviations.....	21
7.2	Appendix B Compiling Environment Setup	22
7.2.1	<i>LLVM Installation</i>	<i>22</i>
7.3	Appendix C Download and Install Python	22
7.3.1	<i>Download Python.....</i>	<i>22</i>
7.3.2	<i>Install Python</i>	<i>23</i>

1 Introduction

This document mainly introduces FibocomOpen solution, how to set the compilation environment in Windows and Linux OS, how to compile user applications in FibocomOpen SDK and how to run user applications

FIBOCOM
Confidential

2 Fibocom Solution Overview

2.1 General Overview

Fibocom MA510-GL-00-90 modules can communicate with external MCUs through AT commands. Based on rich on-chip resources and FibocomOpen solution, the modules can provide corresponding hardware resources, including RAM and NAND flash, and also some peripheral devices, including UART and IIC. Additionally, the modules are available with corresponding Software Development Kit (SDK) to help customers simplify application development.

FibocomOpen provides an infrastructure for applications to dynamically load modules that are built from the resident component of the application. Each module is built independently with a common preamble structure attached in the binary. The preamble contains various details about the modules, including:

- A single thread entry point
- Stack size priority
- Module ID
- Callback thread stack size/priority and so on.



2.2 Fibocom Architecture

The following diagram shows the architecture of FibocomOpen solution.

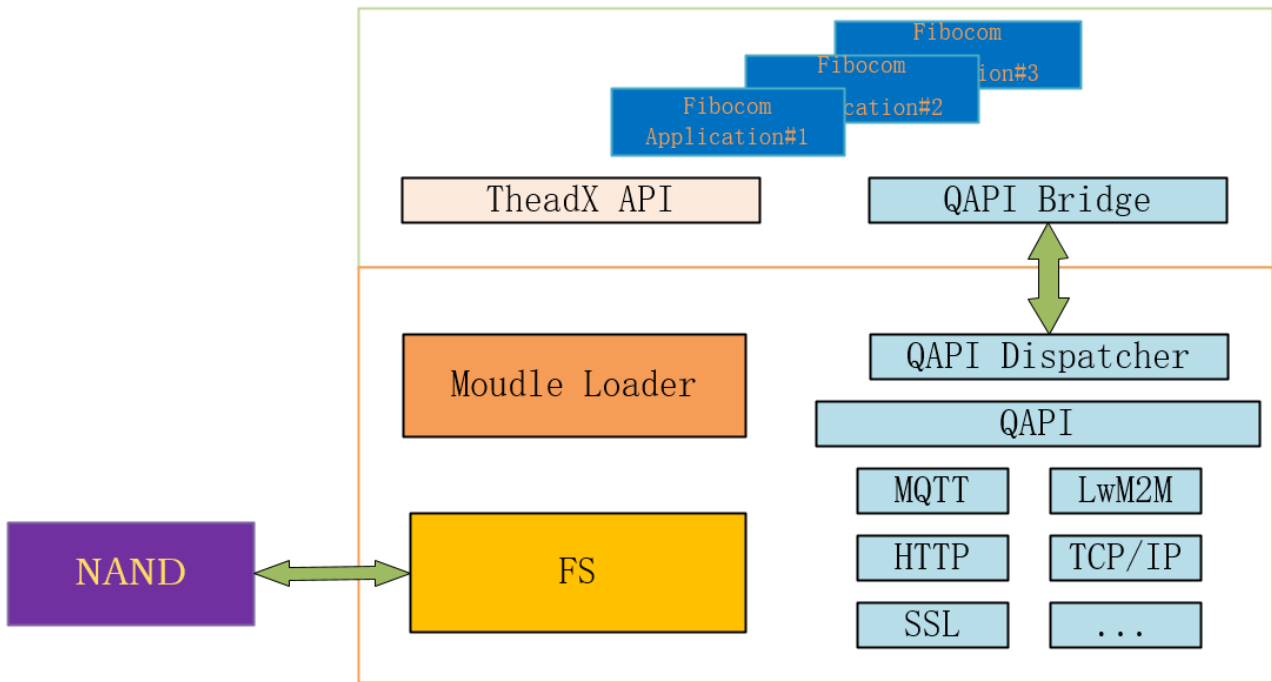


Figure 2-1 Architecture of Fibocom solution



Note:

1. It is recommended to run only a single FibocomOpen application for the purpose of secure storage of application images. In such a case, it is easy to maintain the images and therefore a NAND flash is available to store them. If multiple FibocomOpen applications are run, then the application images must be stored and loaded from the file system.
2. Storing customer application binary images in NAND flash is still under development.

3 Build Fibocom Applications

3.1 FibocomOpen SDK Package

The following shows the folder structure of Fibocom_MA510-GL-00-90_FibocomOpen_SDK_Package.

 bin	2021-01-22 15:52
 common	2020-12-26 17:05
 fibocom	2021-01-22 15:52
 llvmtools	2020-12-26 17:05
 Tracker_ref	2020-12-26 17:05
 build_all.sh	2021-01-22 15:51
 example.txt	2020-12-26 17:05

Figure 3-1 Folder structure of MA510-GL-00-90 FibocomOpen SDK package

Table 3-1 Description of MA510-GL-00-90-Fibocom SDK package directories

Directory	Description
bin	Contains demo Application programs generated after building
common	Contains all common header files and source files
fibocom	All demo source code
llvmtools	llvm tools
example.txt	A compile config file
build_all.sh	Shell script for building Fibocom examples



Note:

- If you want to modify the demo program, you can modify it by entering the corresponding demo folder in fibocom. Finally, refer to sections 3.2 and 3.3 for compiling and run Fibocom applications.
- If you want to add a new folder, you can create a new folder under fibocom/example directory (You can refer to other files in this directory) and modify example.txt. Finally, refer to sections 3.2 and 3.3 for compiling and run Fibocom applications.

3.2 Compile FibocomOpen Applications

Run build_all.sh help to show compile help information;

```
haopeng@ubuntu-236:~/9205/mdm-tx-1.0/fibo_open_sdk_v2$ ./build_all.sh help
help menu:
[ cmd - build_demo.sh llvm build_id ]
```

Figure 3-2 Run build_all.sh help

Run build_all.sh llvm to show currently supported examples;

```
liuben@ubuntu14:~/qcom/mdm-tx-1.0/fibo_open_sdk$ ./build_all.sh llvm
Supported example :
at          [ cmd - ./build_all.sh llvm at          ]
non_ip      [ cmd - ./build_all.sh llvm non_ip      ]
dns         [ cmd - ./build_all.sh llvm dns         ]
location    [ cmd - ./build_all.sh llvm location    ]
lwm2m       [ cmd - ./build_all.sh llvm lwm2m       ]
ping        [ cmd - ./build_all.sh llvm ping        ]
ssl         [ cmd - ./build_all.sh llvm ssl         ]
socket      [ cmd - ./build_all.sh llvm socket      ]
coap        [ cmd - ./build_all.sh llvm coap        ]
mqtt        [ cmd - ./build_all.sh llvm mqtt        ]
hello_world [ cmd - ./build_all.sh llvm hello_world   ]
psm         [ cmd - ./build_all.sh llvm psm         ]
Tracker     [ cmd - ./build_all.sh llvm Tracker     ]
uart        [ cmd - ./build_all.sh llvm uart        ]
```

Figure 3-3 Run build_all.sh llvm

For example, compile at demo, run ./build_all.sh llvm at, it will generate bin file and ini file in bin folder.

```
bin/
├── at_dam_demo.bin
├── at_dam_demo.elf
├── at_dam_demo.map
└── oem_app_path.ini
```

Figure 3-4 Run build_all.sh llvm at

3.3 Run FibocomOpen Applications

There are two methods to run Fibocom applications: one is to load binary images from the alternating file systems (**option 1**), and the other is to push binary images use QFLOG (**option 2**). The option 2 is only used to debug applications

3.3.1 Load Binary Images from Alternate File Systems

To run the Fibocom application binary images in this option, please upload the application binary images

and oem_app_path.ini into the alternate file systems of the modules with QEFS Explorer.

oem_app_path.ini file contains the names of application binary images. This file must be stored in the /datatx/ directory. And only binary files under /datatx/ directory can be loaded. Take at application as an example, we need to put oem_app_path.ini and at_dam_demo.bin in the /datatx directory.

After uploading these two files into alternate file systems, reboot the module and then the application binary images will be loaded into RAM and started by the Module Loader.

Please refer to the following steps:

- Connect the module to a PC via USB, boot the module and make sure that Fibocom USB diag Port, Open QPST tool. At the same time, click Start clients and select EFS Explorer

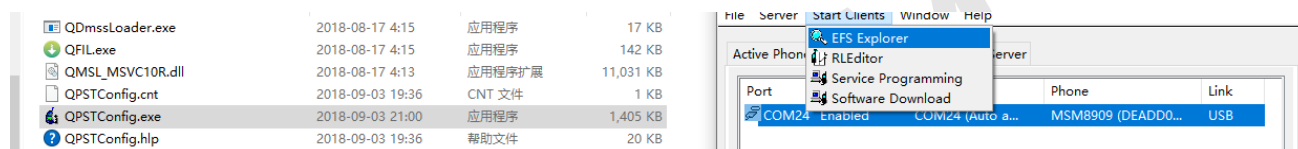


Figure 3-5 Open QPST tool and select EFS Explorer

- After opening, select datatx and open it

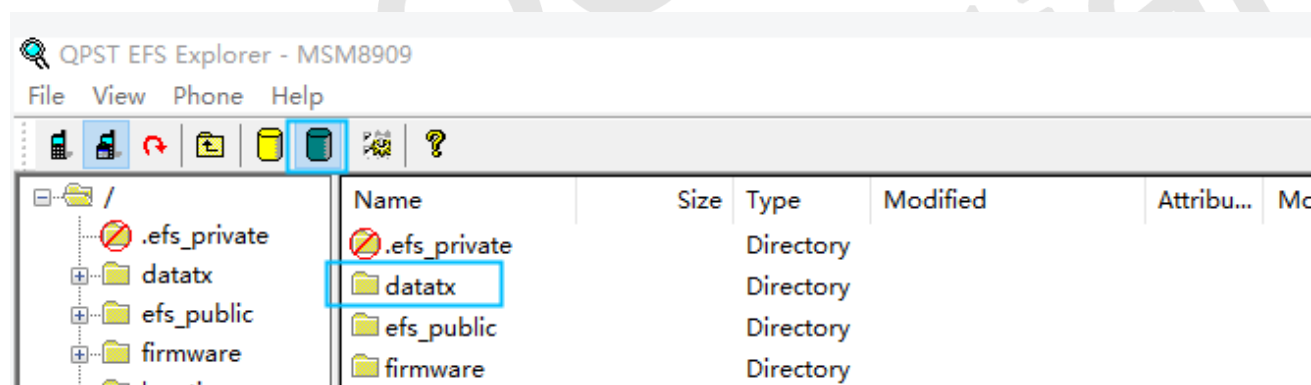


Figure 3-6 Select datatx

- Put the oem_app_path.ini and at_dam_demo.bin files generated in Figure 3-4 into the /datatx directory and restart.

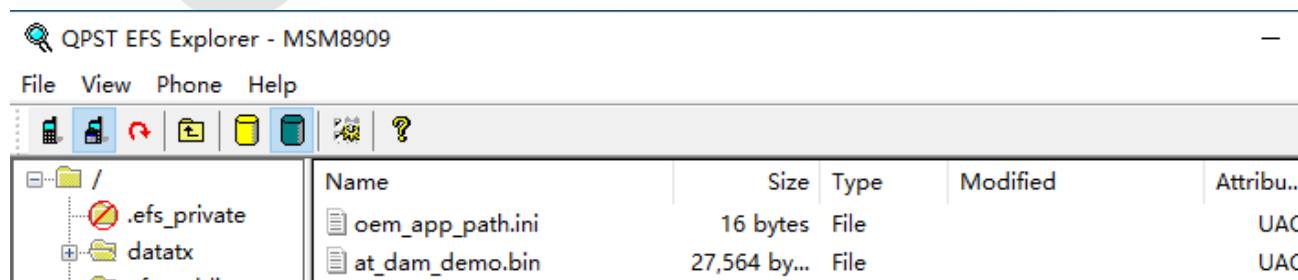


Figure 3-7 Put the oem_app_path.ini and at_dam_demo.bin files into /datatx

3.3.2 Use QFLOG to Load Binary Images to the Device

Use QTI flashing and logging (QFLOG) scripts to load user applications to the device, as well as to log debug messages. Use QFLOG by performing the following steps:

1. Setup python2.7 environment;
2. Install pyserial with command **pip install pyserial**;
3. Run Windows Command Prompt as administrator;
4. Run QFLOG\Setup.bat to set up the QFLOG directory path;
5. Set NV 65768 to 82 with QXDM and reboot the device to use DIAG COM port;
6. To find the SER5 COM port number, navigate to Device Manager from your Windows machine.

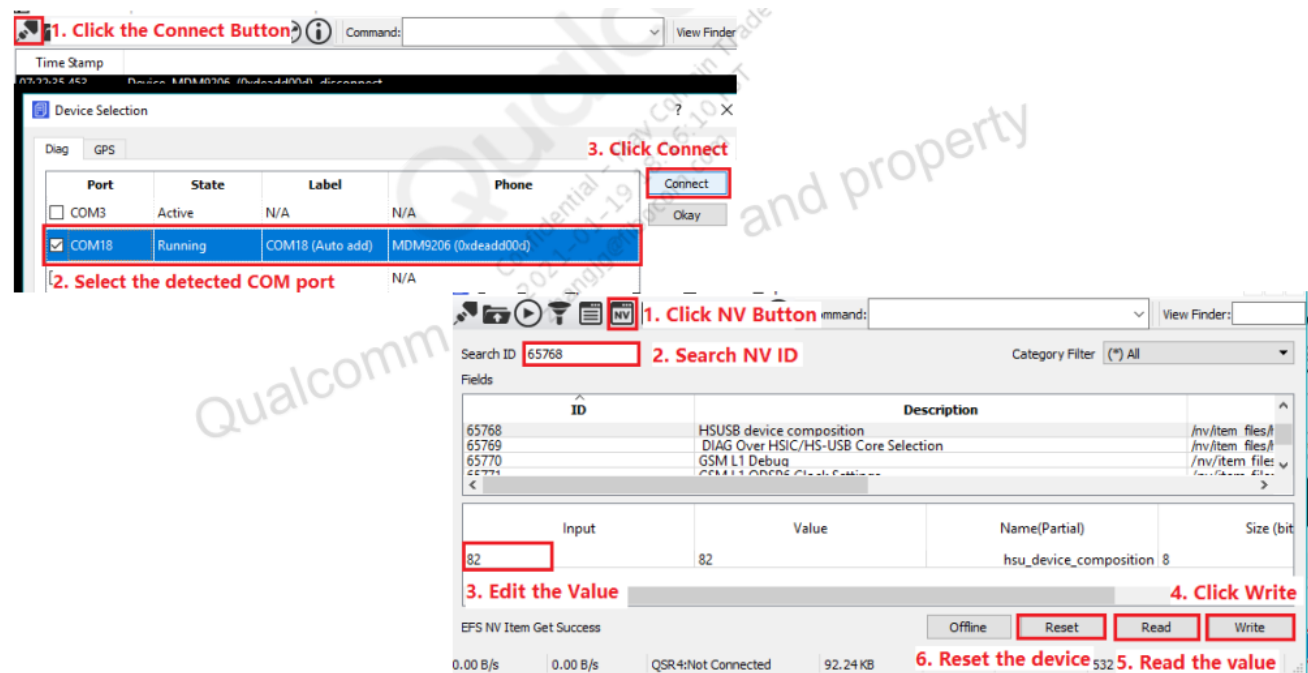


Figure 3-8 Steps to use QFLOG

7. To find the SER5 COM port number, navigate to Device Manager from your Windows machine.
8. Push the following file to /datatx in Alternate file system via EFS explorer and reset the device. This text file contains numerical value 1 content.



9. Run this command **python C:\QFLOG\src\QFLOGPackage\QFLOG.py -p COM85 HELLO** to send a **HELLO** command to check connectivity, The device responds with received ACK.

```
C:\Users\Administrator\Desktop\MDM9205\QFLOG>python src\QFLOGPackage\QFLOG.py -p COM80 HELLO
2021-01-27 16:26:57,558 QFLOG 24972 24956 INFO : Sending HELLO to device
serconn write
2021-01-27 16:26:57,592 QFLOG 24972 27248 INFO : Received ACK
```

Figure 3-9 python C:\QFLOG\src\QFLOGPackage\QFLOG.py -p COM85 HELLO

Reproduction forbidden without Fibocom Wireless Inc. written authorization - All rights reserved.

10. To flash the test application to the device, run the following script with the PUSH command, as shown in the following figure:

QFLOG.py -p <COMPORT> PUSH -f <absolute bin path>

```
C:\Users\Administrator\Desktop\MDM9205\QFLOG>python src\QFLOGPackage\QFLOG.py -p COM80 PUSH -f C:\Users\Administrator\Desktop\hello.bin
2021-01-27 16:28:35,569 QFLOG 26788 26840 INFO : Resetting the context by sending a HELLO packet
2021-01-27 16:28:35,661 QFLOG 26788 27196 INFO : Sending HELLO to device
serconn write
2021-01-27 16:28:35,713 QFLOG 26788 27320 INFO : Received ACK
start QFLOGPush tx threadstart QFLOGPush rx thread
2021-01-27 16:28:35,805 QFLOG 26788 27464 INFO : packetTX: {'_QFLOGPacket__totSize': 3L, '_QFLOGPacket__payload': 'hello.bin', '_QFLOGPacket__ctxid': 0, '_QFLOGPacket__cmd': 2, '_QFLOGPacket__pad': 0, '_QFLOGPacket__pktid': 1,
_QFLOGPacket__payloadLen': 9}
2021-01-27 16:28:35,846 QFLOG 26788 27464 INFO : bytearray: '\x00\x02\x01\x00\x00\x00\x00\x00\x00\x00\x00\x08\x05\x06\x0f\x02\x09\x06'
serconn write
2021-01-27 16:28:36,084 QFLOG 26788 27464 INFO : Pushing hello.bin to device
serconn write
2021-01-27 16:28:36,255 QFLOG 26788 26536 INFO : Received ACK
2021-01-27 16:28:36,629 QFLOG 26788 27464 INFO : Push complete
```

Figure 3-10 QFLOG.py -p <COMPORT> PUSH -f <absolute bin path>

3.3.3 Use QFLOG to Debug

Make the following code changes to test applications, and to enable logging through QFLOG scripts:

1. Include the qflog_utils.h file in your code.
2. Use the following code in user applications, to log a message:

```
QFLOG_MSG(MSG_SSID_DFLT,MSG_MASK_2, "Presence sensor application successfully
registered ");

QFLOG_MSG(MSG_SSID_DFLT,MSG_MASK_2, "Sensor time value is %d
",sensor_time_val);
```

To view the logs from the device, run the following Python script, as shown in the screenshot below:

QFLOG.py -p <COMPORT> VIEW_LOGS

```
C:\Users\Administrator\Desktop\MDM9205\QFLOG>python src\QFLOGPackage\QFLOG.py -p COM80 VIEW_LOGS
2021-01-27 17:31:55,128 QFLOG 13124 4980 INFO : Sending HELLO to device
serconn write
2021-01-27 17:31:55,131 QFLOG 13124 1176 INFO : Received ACK
2021-01-27 17:31:55,134 QFLOG 13124 7964 INFO : Enabling logging
serconn write
2021-01-27 17:32:08,941 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:test_dam_app_start entry
2021-01-27 17:32:08,943 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:tx_thread_create with ret[0]
2021-01-27 17:32:08,944 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:basic_dam_app_start exit with ret[0]
2021-01-27 17:32:08,944 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_task_entry
2021-01-27 17:32:08,944 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_task_init
2021-01-27 17:32:08,944 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_cmdq_init
2021-01-27 17:32:08,944 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_mutex_init
2021-01-27 17:32:08,946 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_signal_init
2021-01-27 17:32:08,946 QFLOG 13124 14440 INFO : [LOG]: HELLO WORLD TEST...
2021-01-27 17:32:08,946 QFLOG 13124 14440 INFO : [LOG]: Timer set successfully.
2021-01-27 17:32:08,947 QFLOG 13124 14440 INFO : [LOG]: HELLO WORLD TEST END...
2021-01-27 17:32:08,947 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_signal_wait
2021-01-27 17:32:13,944 QFLOG 13124 14440 INFO : [LOG]: Period Timer(5seconds) reach, send signal to task
2021-01-27 17:32:13,947 QFLOG 13124 14440 INFO : [LOG]: HELLO WORLD TIMER...
2021-01-27 17:32:13,950 QFLOG 13124 14440 INFO : [LOG]: DAM_APP:dam_obj_set_signal
```

Figure 3-11 QFLOG.py -p <COMPORT> VIEW_LOGS

4 GPIO Pin Mapping

This chapter mainly introduces the GPIO pin mapping of the FibocomOpen modules, including the mapping of related peripherals, such as UART, IIC and so on.

Table 4-1 Definition of I/O parameters

Type	Description
AI	Analog Input
AO	Analog Output
BCMOS	Bidirectional digital with CMOS input
DI	Digital Input
IO	Bidirectional
OD	Open Drain
PI	Power Input
PO	Power Output
PU	Pull-up
PD	Pull-down

4.1 GPIO Pin Mapping of MA510-GL-00-90 FibocomOpen

The following tables show the GPIO pin mapping of MA510-GL-00-90 FibocomOpen module.

Table 4-2 GPIO pin mapping of MA510-GL-00-90 FibocomOpen

Pin Name	Pin No.	Function 1	Function 2	Function 3	Function 4	Reset ¹⁾	Interrupt
UART2_TXD	40	GPIO0	UART2_TXD	-	-	PD	Support
UART2_RXD	39	GPIO1	UART2_RXD	-	-	PD	Support
UART2_CTS	75	GPIO2	UART2_CTS	-	-	PD	Support
UART2_RTS	76	GPIO3	UART2_RTS	-	-	PD	-
UART3_TXD	78	GPIO4	UART3_TXD	-	-	PD	Support

Reproduction forbidden without Fibocom Wireless Inc. written authorization - All rights reserved.

UART3_RXD	77	GPIO5	UART3_RXD	-	-	PD	Support
I2C_SDA	52	GPIO6	I2C_SDA	-	-	PD	Support
I2C_SCL	56	GPIO7	I2C_SCL	-	-	PD	-
UART1_DTR	15	GPIO8	UART1_DTR	-	-	PD	Support
UART1_DCD	17	GPIO9	UART1_DCD	-	-	PD	Support
UART1_RING	22	GPIO10	UART1_RING	-	-	PD	-
UART1_DSR	16	GPIO11	UART1_DSR	-	-	PD	-
UART1_TXD	19	GPIO12	UART1_TXD	-	-	PD	-
UART1_RXD	18	GPIO13	UART1_RXD	-	-	PD	Support
UART1_RTS	20	GPIO15	UART1_RTS	-	-	PD	-
UART1_CTS	21	GPIO14	UART1_CTS	-	-	PD	Support
GPIO1	53	GPIO31	-	-	-	-	-
GPIO2	54	GPIO26	-	-	-	-	-
GPIO3	58	GPIO33	-	-	-	-	-
GPIO4	62	GPIO34	-	-	-	-	-



Note:

1. The pin functions 1, 2, 3 and 4 take effect only after software configuration.
2. Please refer to Table 4-1 for more details about the symbol description.
3. “*” means under development.
4. “/” means not supported.

4.2 GPIOs

MA510-GL-00-90 FibocomOpen supports 20 GPIOs. Each GPIO can be configured by QAPI in FibocomOpen applications. When a GPIO is configured into an output, its drive strength can be configured.

4.3 UART Interfaces

MA510-GL-00-90 FibocomOpen provides three UART interfaces: UART1, UART2 and UART3.

- UART1 interface is used for at command communication.
- UART2 interface is used for debug serial port.
- UART3 interface is used for peripherals and can be reused for other functions.

The following table shows the pin definitions for the three UART interfaces.

Table 4-3 Pin definition of UART1 interface

Pin Name	Pin No	I/O	Description	DC Characteristic	Comment
UART1_DTR	15	DI	data terminal ready	$V_{ILmin}=-0.3V$ $V_{ILmax}=0.6V$ $V_{IHmin}=1.2V$ $V_{IHmax}=2.0V$	1.8V power domain. If unused, keep this pin open.
UART1_DSR	16	I/O	Data equipment preparation	$V_{ILmin}=-0.3V$ $V_{ILmax}=0.6V$ $V_{IHmin}=1.2V$ $V_{IHmax}=2.0V$	1.8V power domain UART1 reserved function
UART1_DCD	17	DO	Data carrier detection	$V_{OLmax}=0.45V$ $V_{OHmin}=1.35V$	1.8V power domain. If unused, keep this pin open.
UART1_RXD	18	DI	Module receives data	$V_{ILmin}=-0.3V$ $V_{ILmax}=0.6V$ $V_{IHmin}=1.2V$ $V_{IHmax}=2.0V$	1.8V power domain. If unused, keep this pin open.
UART1_TXD	19	DO	Module sends data	$V_{OLmax}=0.45V$ $V_{OHmin}=1.35V$	1.8V power domain. If unused, keep this pin open.
UART1_RTS	20	DO	Request to send	$V_{ILmin}=-0.3V$ $V_{ILmax}=0.6V$ $V_{IHmin}=1.2V$ $V_{IHmax}=2.0V$	1.8V power domain. If unused, keep this pin open.
UART1_CTS	21	DI	Clear send	$V_{OLmax}=0.45V$ $V_{OHmin}=1.35V$	1.8V power domain. If unused, keep this pin open.
UART1_RING	22	DO	Ring indication	$V_{OLmax}=0.45V$ $V_{OHmin}=1.35V$	1.8V power domain. If unused, keep this pin open.



Note:

The UART1 interface is only used for AT Command communication. It cannot be configured or used in customer's own applications.

Table 4-4 Pin definition of UART2 interface

Pin Name	Pin No	Function 1	Function 2	Function 3	Function 4
UART2_TXD	40	GPIO0	UART2_TXD	-	-
UART2_RXD	39	GPIO1	UART2_RXD	-	-



Note:

The UART2 interface is only used for debug UART. Cannot be configured or used in the customer's own application.

Table 4-5 Pin definition of UART3 interface

Pin Name	Pin No	Function 1	Function 2	Function 3	Function 4
UART3_TXD	78	GPIO4	UART3_TXD	-	-
UART3_RXD	77	GPIO5	UART3_RXD	-	-



Note:

- In the fibocom application, use BLSP2_ UART_ RX / TX to configure uart3
- Uart3 interface does not support flow control



4.4 I2C Interface

MA510-GL-00-90 FibocomOpen provides one Inter-Integrated Circuit (I2C) interface for data communication. The interface supports fast-mode plus and master mode only.

The I2C interface is multiplexed from GPIOs. The I2C interface pins are open drain that must be pulled up to 1.8 V, and the pull-up resistors should be provided externally.

The following table shows the pin definition.

Table 4-6 Pin definition of I2C interface

Pin Name	Pin No	Function 1	Function 2	Function 3	Function 4
I2C_SDA	52	GPIO6	I2C_SDA	-	-
I2C_SCL	56	GPIO7	I2C_SCL	-	-



Note:

In the fibocom application, use BLSP2_I2C_SDA/SCL to configure I2C.

FIBOCOM
Confidential

5 Available Memory Mapping

Fibocom MA510-GL-00-90 FibocomOpen modules have fine-tuned the memory space division, including the NAND flash and RAM spaces, to meet the diverse needs of customers.

Fibocom provides four spaces for customers to use in FibocomOpen solutions:

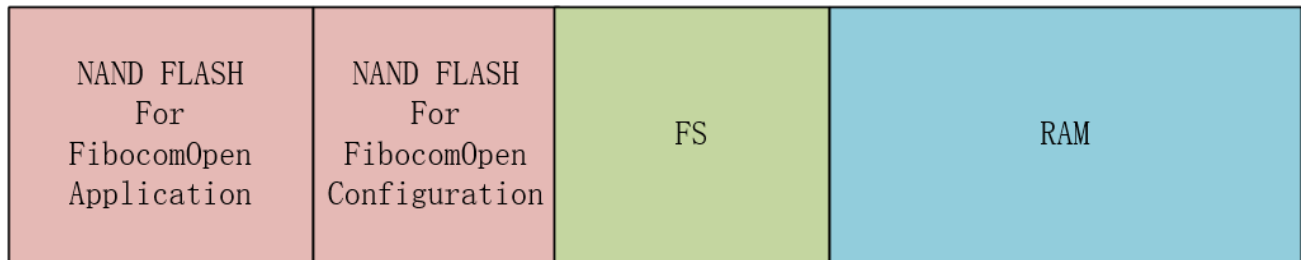


Figure 5-1 Memory space for FibocomOpen app

- NAND flash for FibocomOpen application: Used to store FibocomOpen application images.
- NAND flash for FibocomOpen configuration: Used to store the key configuration data of FibocomOpen applications.
- FS: Used to store customers' temporary configurations, files and application logs.
- RAM: RAM space.

Table 5-1 Available memory mapping for FibocomOpen applications

Model	NAND Flash for FibocomOpen Application (MB)	NAND Flash for FibocomOpen Configuration (KB)	FS(MB)	RAM(MB)
MA510-GL-00-90	1.5	128	2.5	3



Note:

The above is the memory space information provided without VoLTE/Audio function.

6 Download Image

- Connect the module to a PC via USB, boot the module and make sure that Fibocom USB diag Port, Open Fibocom_MDM_MultiUpdater tool, as step 1 in the following figure.
- Click **Browse** (step 2 in the following figure) to select image file prog_firehose_nand.elf, loading ramprogram_nand_p2K_b128K.xml and patch_p2k_b128K.xml at the same time.
- Check port (step 3 in the figure below) and click **Station Config** (step 4 in the figure below)
- Click **Download** (step 5 in the figure below)
- Press and hold reset. Restart the module and upgrade successfully.

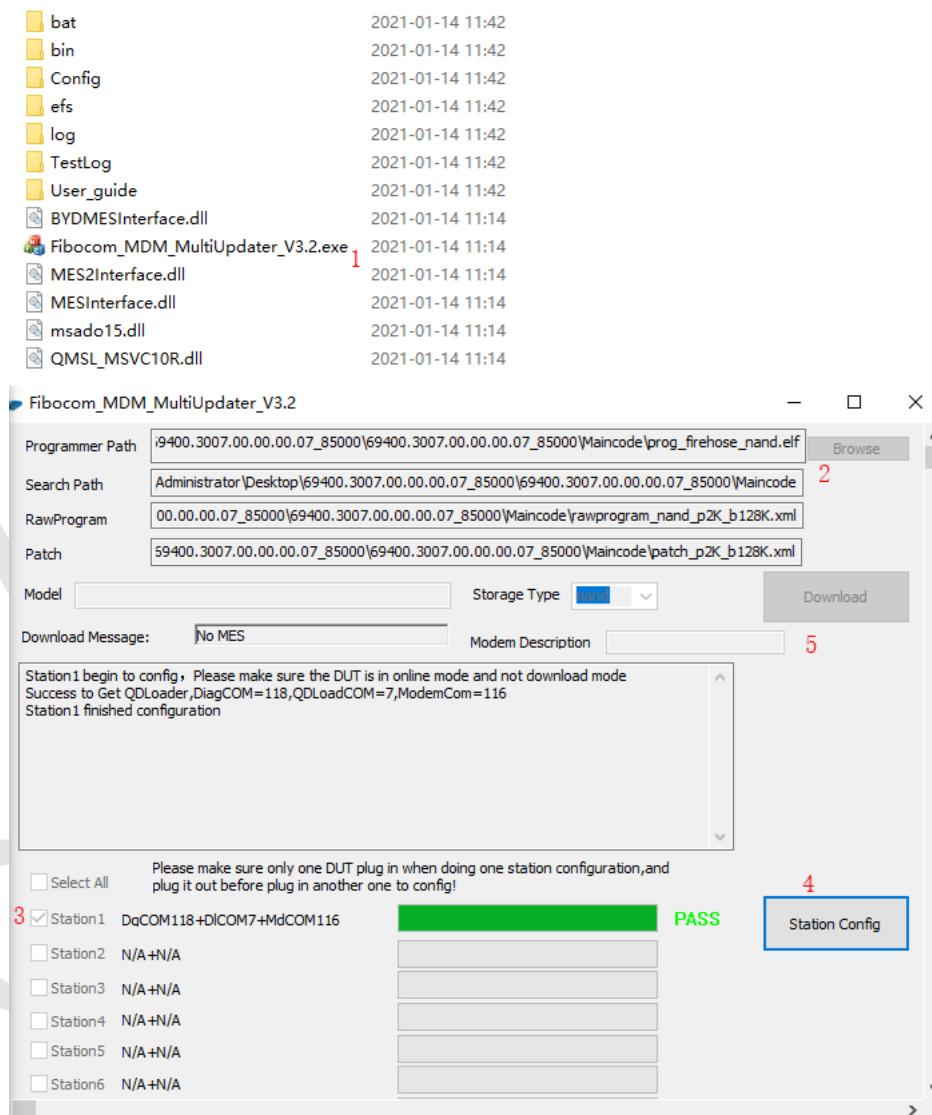


Figure 6-1 Download image

7 Appendixes

7.1 Appendix A Terms and Abbreviations

Table 7-1 Terms and abbreviations

Abbreviation	Description
API	Application Programming Interface
GPIO	General-Purpose Input/Output
HTTP	Hyper Text Transfer Protocol
I2C	Inter-Integrated Circuit
MCU	Microcontroller Unit
OS	Operating System
QAPI	Qualcomm™ Application Programming Interface
RAM	Random Access Memory
ROM	Read Only Memory
SDK	Software Development Kit

7.2 Appendix B Compiling Environment Setup

FibocomOpen solution supports the LLVM compiler provided by Qualcomm only.

Table 7-2 Compiling environment requirement

Component	Source or Binary Only	Toolchain Required for Building Source	Supported Build Host
FibocomOpen SDK	Source	LLVM 4.0.3	Windows 7/ Windows 10/Linux

7.2.1 LLVM Installation

Before install the LLVM, please request the compiler tool package from Fibocom.

7.3 Appendix C Download and Install Python

7.3.1 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/>.

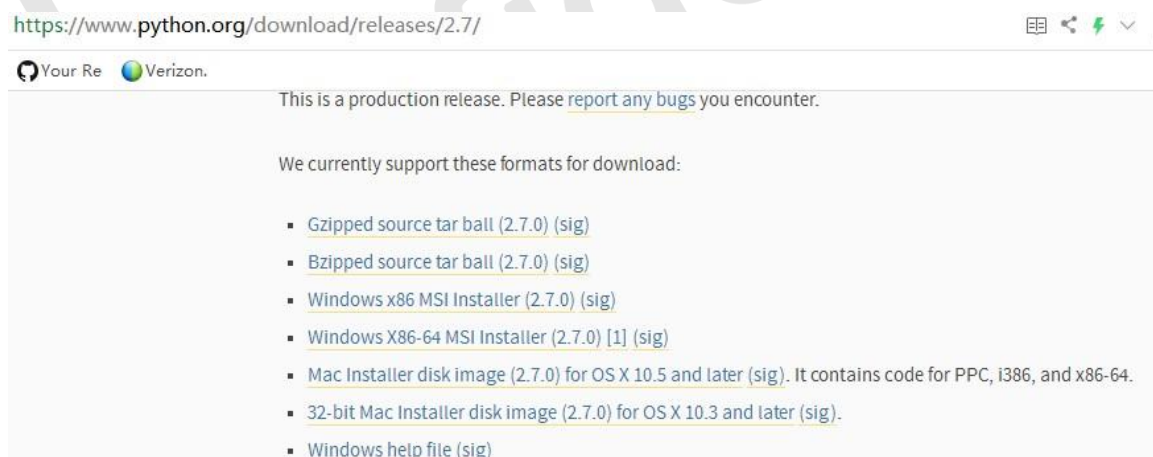


Figure 7-1 Python download page screenshot

Download x86/x86-64 versions as needed.

7.3.2 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 7-2 Python setup

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

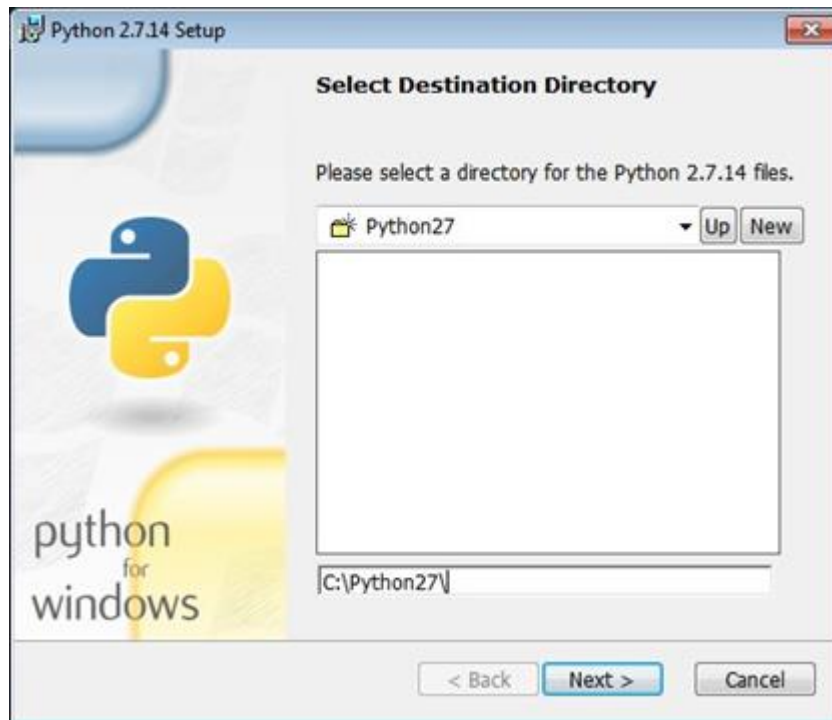


Figure 7-3 Select installation directory

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



Figure 7-4 Options for customization

Step 4: Please wait during installation process.

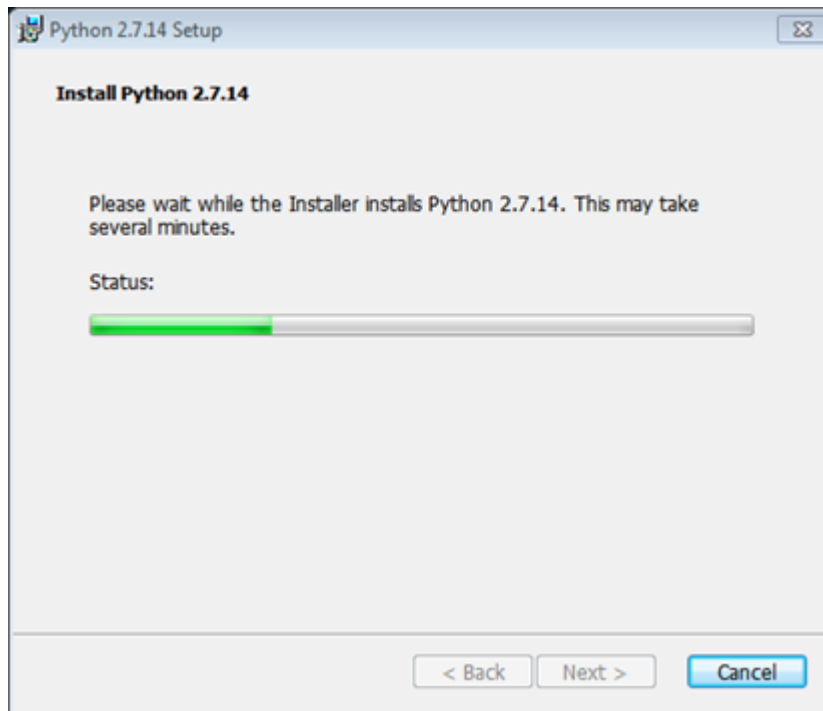


Figure 7-5 Installation

Step 5: Complete installation.



Figure 7-6 Installation completed