RG520N&RG525F&RG5x0F&RM5x0N Series

Data Call Application Note

5G Module Series

Version: 1.1

Date: 2025-06-11

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our 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 offices. 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 us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

About the Document

Revision History

Version	Date	Author	Description
-	2022-05-13	Ozzy ANG	Creation of the document
1.0	2022-10-08	Ozzy ANG	First official release
1.1	2025-06-11	Demon YANG	1. Added RG525F-NA and RG530F series modules. 2. Updated RM520N-GL to RM520N series. 3. Deleted the ECM data call on Windows system. 4. Added the description of the ECM data call on Linux system (Chapter 3.9).

Contents

About the Document **3**
Contents **4**
Table Index **5**
Figure Index **6**
1 Introduction **7**
1.1. Applicable Modules 7
2 Data Call Methods **8**
2.1. Data Call Scenarios 8
2.2. Data Call Methods 9
3 Data Call Steps **10**
3.1. NDIS Data Call (USB + TE + Windows + QMI) 10
3.2. QMI_WWAN Data Call (USB + TE + Linux + QMI) 11
3.3. GobiNet Data Call (USB + TE + Linux + QMI) 12
3.4. PCIe_MHI Data Call (PCIe + TE + Linux + QMI) 13
3.5. MBIM(USB) Data Call (USB + TE + Linux + MBIM) 14
3.6. MBIM(USB) Data Call (USB + TE + Windows + MBIM) 15
3.7. PCIe_MHI_MBIM Data Call (PCIe + TE + Windows + MBIM) 16
3.8. PCIe_MHI_MBIM Data Call (PCIe + TE + Linux + MBIM) 16
3.9. ECM Data Call (USB + AP + Linux) 17
3.10. QMAP Data Call (RGMII/PCIe RC + AP) 18
4 Appendix References **20**

Table Index

Table 1: Applicable Modules 7
Table 2: Data Call Scenarios and Corresponding Data Call Methods 9
Table 3: Related Documents 20
Table 4: Terms and Abbreviations 20

Figure Index

Figure 1: Confirm Driver Installation 10
Figure 2: NDIS Data Call (USB + TE + Windows + QMI) 10
Figure 3: MBIM Data Call (USB + TE + Windows + MBIM) 15
Figure 4: MBIM Data Call (PCIe + TE + Windows + MBIM) 16

1 Introduction

Quectel 5G RG520N series, RG525F-NA, RG5x0F family and RM5x0N family modules support data call. This document introduces the data call methods as well as the corresponding steps and development considerations of these modules.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Family	Module
-	RG520N Series
-	RG525F-NA
RG5x0F	RG520F Series
	RG530F Series
RM5x0N	RM520N Series
	RM530N-GL

2 Data Call Methods

2.1. Data Call Scenarios

Data call scenarios can be classified by the:

1. **Communication method** between the host and the module in the data call:
   - QMI (application layer communication protocol)
   - MBIM (application layer communication protocol)
   - None (standard interface bottom layer communication protocol, such as Ethernet and Wi-Fi)

2. **Interface type** used for the data call:
   - USB
   - PCIe Interface
     - EP mode (module serves as a device)
     - RC mode (module serves as a master, such as Ethernet, WLAN Wi-Fi device)
   - RGMII

3. **Initiator** of the data call:
   - TE: Host initiates the data call.
   - AP: Module initiates the data call.

4. **Host operating system (OS)** for the data call:
   - Windows
   - Linux (Android)

2.2. Data Call Methods

According to the data call scenarios listed in ***Chapter 2.1***, the corresponding data call methods supported by the module are listed in the table below.

Table 2: Data Call Scenarios and Corresponding Data Call Methods

Interface Types	Initiators	Host OS	Communication Methods	Data Call Methods
USB	TE	Windows	QMI	NDIS
USB	TE	Linux	QMI	QMI_WWAN/GobiNet
USB	TE	Windows/Linux	MBIM	MBIM(USB)
PCIe (EP Mode)	TE	Linux	QMI	PCIe_MHI_Rmnet
PCIe (EP Mode)	TE	Windows/Linux	MBIM	PCIe_MHI_MBIM
USB	AP	Linux	-	ECM
RGMII/PCIe RC	AP	-	-	QMAP

NOTE

1. Except for QMAP, the other data call methods are named after the driver name.
2. For detailed steps of each data call method, see ***Chapter 3***. For details about AT commands, see ***document [1][1]*** or contact Quectel Technical Support.

3 Data Call Steps

3.1. NDIS Data Call (USB + TE + Windows + QMI)

1. Install NDIS driver Quectel_Windows_USB_Driver(Q)_For_NDIS, which is provided by Quectel on the host.
2. Connect the module to the host via USB cable.
3. Check if the driver “Quectel Wireless Ethernet Adapter” is loaded under “Network adapters” on the host as shown in the figure below:

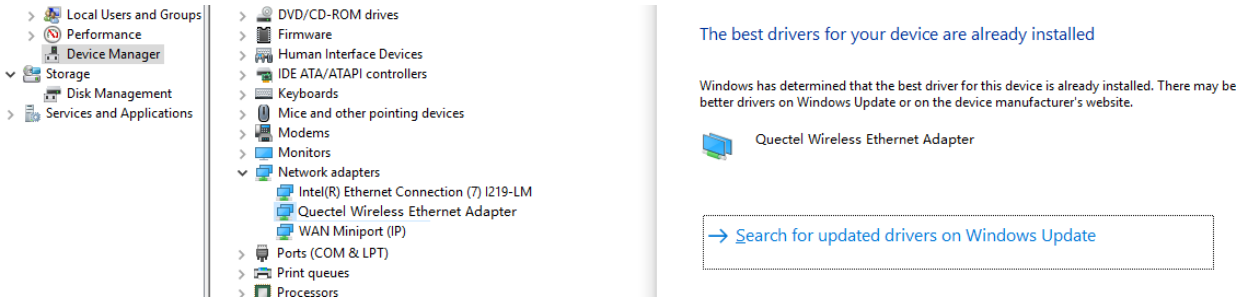

Figure 1: Confirm Driver Installation

4. Click the network icon in the lower right corner of the host taskbar to initiate a data call.

Figure 2: NDIS Data Call (USB + TE + Windows + QMI)

3.2. QMI_WWAN Data Call (USB + TE + Linux + QMI)

1. Contact Quectel Technical Support for installation packages of the following drivers and tool:
   - Quectel_Linux_USB_Serial_Option_Driver (Driver for TTY port, such as AT and DM.)
   - Quectel_Linux&Android_QMI_WWAN_Driver (Driver for usbnet data port)
   - QConnectManager_Linux (Data call tool)

2. Integrate and compile these drivers and tool on the host by executing **make install** in the driver file.
   - If multiple data call with driver is required, add parameter *qmap_mode=<number of data call channels>* in the command line when executing the command for loading the driver. Currently, the maximum number of data call channels is 4. For example:

     ```
     insmod qmi_wwan_q.ko qmap_mode=4
     ```

     qmap_mode=0 indicates a single data call without UL AGG.
     qmap_mode=1 indicates a single data with UL AGG.
     qmap_mode=2/3/4 indicates multiple data call with UL AGG.

   - If bridge mode with driver is required (i.e. to bridge the data call to other devices on the host to provide network access to these devices). You can refer to the following examples in */log* directory of quectel-CM source code package.

log/how_to_use_bridge.txt	Bridge mode for a single data call
/loghow_to_use_bridge_and_QMAP.txt	Bridge mode for multiple data call

3. Connect the module to the host via USB cable, and execute **AT+QCFG="usbnet"** to query the NIC data call method. If it is not RmNet, configure it by executing **AT+QCFG="usbnet",0**. For details about the AT commands, see ***document [1]***.

4. Perform a data call with quectel-CM tool.
   1) Execute **quectel-CM -h** to view the usage of the tool parameters.
   2) You can refer to the following examples in */log* directory of the tool source code package.

qmi_wwan_q.txt	Local single data call without using QMAP
qmi_wwan_q_bridge.txt	Single data call in bridge mode without using QMAP
qmi_wwan_q_qmap=1.txt	Local single data call with QMAP
qmi_wwan_q_qmap=1_bridge.txt	Single data call in bridge mode with QMAP
qmi_wwan_q_qmap=4.txt	Local multiple data call with QMAP
qmi_wwan_q_qmap=4_bridge.txt	Multiple data call in bridge mode with QMAP

NOTE

1. In the context of QMI_WWAN data calls, QMAP is the multiple data call function of the QMI_WWAN driver, which is different from the QMAP data call introduced in ***Chapter 3.10***
2. QMAP feature of the module is enabled by default, i.e., even if *qmap_mode* is not configured when loading the driver, *qmap_mode=1* will be set automatically when running the driver. Therefore, *qmi_wwan_q.txt* and *qmi_wwan_q_bridge.txt* examples are equivalent to *qmi_wwan_q_qmap=1.txt* and *qmi_wwan_q_qmap=1_bridge.txt* in practice.
3. Do **NOT** use Linux original drivers (qmi_wwan.ko and qcserial.ko) on the host when using Quectel modules. You can check if the Linux original drivers are installed by running **lsmod**. If the drivers are installed, you should run **rmmod** to uninstall them. Before connecting a Quectel module to the host, make sure that drivers qmi_wwan_q.ko and option.ko provided by Quectel are installed.

3.3. GobiNet Data Call (USB + TE + Linux + QMI)

1. Contact Quectel Technical Support for installation package of the following drivers and tool:
   - Quectel_Linux_USB_Serial_Option_Driver (Driver for TTY port, such as AT and DM.)
   - Quectel_Linux&Android_GobiNet_Driver (Driver for usbnet data port)
   - QConnectManager_Linux (Data call tool)

2. Integrate and compile these drivers and tool on the host by executing make install in the driver file.
   - If multiple data call with driver is required, add parameter *qmap_mode=<number of data call channels>* in the command line when executing the command for loading the driver. Currently, the maximum number of data call channels is 4. For example:

     ```
     insmod GobiNet.ko qmap_mode=4
     ```

     qmap_mode=0 indicates a single data call without UL AGG.
     qmap_mode=1 indicates a single data with UL AGG.
     qmap_mode=2/3/4 indicates multiple data call with UL AGG.

After the driver is installed successfully, it will overwrite Linux original driver GobiNet.

3. Connect the module to the host via USB cable, and execute **AT+QCFG="usbnet"** to query the NIC data call method. If it is not RmNet, configure it by executing **AT+QCFG="usbnet",0**. For details about the AT commands, see ***document [1]***.

4. Perform a data call with quectel-CM tool.
   1) Execute **quectel-CM -h** to view the usage of the tool parameters.
   2) You can refer to the following examples in */log* directory of quectel-CM source code package.

gobinet.txt	Local single data call without using QMAP
gobinet_bridge.txt	Single data call in bridge mode without using QMAP
gobinet_qmap=1.txt	Local single data call with QMAP
gobinet_qmap=1_bridge.txt	Single data call in bridge mode with QMAP
gobinet_qmap=4.txt	Local multiple data call with QMAP
gobinet_qmap=4_bridge.txt	Multiple data call in bridge mode with QMAP

NOTE

1. In the context of GobiNet data calls, QMAP is the multiple data call function of the GobiNet driver, which is different from the QMAP data call introduced in ***Chapter 3.10***.
2. QMAP feature of the module is enabled by default, i.e., even if *qmap_mode* is not configured when loading the driver, *qmap_mode=1* will be set automatically when running the driver. Therefore, *gobinet.txt* and *gobinet_bridge.txt* examples are equivalent to *gobinet_qmap=1.txt* and *gobinet_qmap=1_bridge.txt* in practice.
3. Do **NOT** use Linux original driver qcserial.ko on the host when using Quectel modules. You can check if the Linux original drivers are installed by running **lsmod**. If the drivers are installed, you should run **rmmod** to uninstall them. Before connecting a Quectel module to the host, make sure that drivers GobiNet.ko and option.ko provided by Quectel are installed.

3.4. PCIe_MHI Data Call (PCIe + TE + Linux + QMI)

PCIe_MHI uses PCIe interface to initiate a data call. The host is a PCIe RC and the module is a PCIe EP. The data call steps are listed below:

1. Contact Quectel Technical Support for installation package of the following driver and tool:
   - Quectel_Linux_PCIE_MHI_Driver (Driver for MHI data port)
   - QConnectManager_Linux (Data call tool)

2. Integrate and compile these drivers and tool on the host by executing **make install** in the driver file.
   - If multiple data call with driver is required, add parameter *qmap_mode=<number of data call channels>* in the command line when executing the command for loading the driver. Currently, the maximum number of data call channels is 4. For example:

     ```
     insmod pcie_mhi.ko qmap_mode=4
     ```

     qmap_mode=0 indicates a single data call without UL AGG.
     qmap_mode=1 indicates a single data with UL AGG.
     qmap_mode=2/3/4 indicates multiple data call with UL AGG.

     View the device port loaded by the host, where /dev/mhi_DUN is the AT port.

3. Connect the module to the host via PCIe.
4. Perform a data call with quectel-CM tool.
   1) Execute **quectel-CM -h** to view the usage of the tool parameters.
   2) You can refer to the following examples in */log* directory of quectel-CM source code package.

pcie_mhi_qmap=1.txt	Local single data call with QMAP
pcie_mhi_qmap=1_bridge.txt	Single data call in bridge mode with QMAP
pcie_mhi_qmap=4.txt	Local multiple data call with QMAP
pcie_mhi_qmap=4_bridge.txt	Multiple data call in bridge mode with QMAP

NOTE

1. In the context of PCIe_MHI_Rmnet data calls, QMAP is the multiple data call function of the PCIe_MHI driver, which is different from the QMAP data call introduced in ***Chapter 3.10***.
2. QMAP feature of the module is enabled by default, i.e., even if *qmap_mode* is not configured when loading the driver, *qmap_mode=1* will be set automatically when running the driver.
3. Before connecting the module to the host, make sure that pcie_mhi.ko driver provided by Quectel is installed.

3.5. MBIM(USB) Data Call (USB + TE + Linux + MBIM)

1. Contact Quectel Technical Support for installation package of the following drivers and tool:
   - Quectel_Linux_USB_Serial_Option_Driver (Driver for TTY port, such as AT and DM.)
   - cdc_mbim_drv_Linux (Driver for usbnet data port)
   - QConnectManager_Linux (Data call tool)

2. Integrate and compile these drivers and tool on the host by executing **make install** in the driver file.

   To perform multiple data call with the driver, add *mbim_qmap_mode=<number of data call channels>* parameter in the command line when you execute the command for loading the driver. The maximum number of data call channels is 16. Example:

   ```
   insmod quec_cdc_mbim.ko mbim_qmap_mode=4
   ```

3. Connect the module to the host via USB cable.

4. Set the module network data call method to MBIM with **AT+QCFG="usbnet",2**. For details of the AT command, see ***document [1]***.

5. Start a data call with quectel-CM.

1) Execute **quectel-CM -h** to check help information of quectel-CM.
2) Refer to the following examples in */log* directory of quectel-CM source code package.

cdc_mbim.txt	Local single data call with QMAP
cdc_mbim_vlan.txt	Local multiple data call with QMAP

NOTE

1. In the context of MBIM data calls, QMAP is the multiple data call function of the MBIM driver, which is different from the QMAP data call introduced in ***Chapter 3.10***.
2. QMAP feature of the module is enabled by default, i.e., even if *qmap_mode* is not configured when loading the driver, *qmap_mode=1* will be set automatically when running the driver.
3. Do **NOT** use Linux original driver qcserial.ko on the host when using Quectel modules. You can check if the Linux original drivers are installed by running **lsmod**. If the drivers are installed, you should run **rmmod** to uninstall them. Before connecting a Quectel module to the host, make sure that driver quec_cdc_mbim.ko provided by Quectel is installed.

3.6. MBIM(USB) Data Call (USB + TE + Windows + MBIM)

1. Connect the module to the host via USB cable.

2. Set the module network card data call method to MBIM with **AT+QCFG="usbnet",2**. For details of the AT command, see ***document [1]***.

3. Make sure that the driver loaded for the device in Windows system is the default MBIM driver.

4. Click the network icon in the lower right corner of the host task bar to initiate a data call.

Figure 3: MBIM Data Call (USB + TE + Windows + MBIM)

3.7. PCIe_MHI_MBIM Data Call (PCIe + TE + Windows + MBIM)

1. Contact Quectel Technical Support for installation package of the Windows PCIe driver and install it.
2. Connect the module to the host via PCIe.
3. Set the module network card data call method to MBIM with **AT+QCFG="pcie_mbim",1**. For details of the AT command, see ***document [1]***.
4. Confirm that the driver loaded by the host Windows system for the device is the driver provided by Quectel. If not, please contact Quectel Technical Support.
5. Click the network icon in the lower right corner of the host taskbar to initiate a data call.

Figure 4: MBIM Data Call (PCIe + TE + Windows + MBIM)

3.8. PCIe_MHI_MBIM Data Call (PCIe + TE + Linux + MBIM)

This data call uses PCIe interface to initiate a data call. The host serves as a PCIe RC and the module serves as a PCIe EP. The data call steps are listed below:

1. Contact Quectel Technical Support for installation package of the following driver and tool:
   - Quectel_Linux_PCIE_MHI_Driver (Driver for MHI data port)
   - QConnectManager_Linux (Data call tool)
2. Integrate and compile these drivers and tool on the host by executing **make install** in the driver file.
   - To enable the driver in MBIM mode, add *mhi_mbim_enabled=1* in the command line when executing the command for loading the driver.
   - If multiple data call with driver is required, add parameter *qmap_mode=<number of data call*

channels> in the command line when executing the command for loading the driver. Currently, the maximum number of data call channels is 4. For example:

```
insmod pcie_mhi.ko mhi_mbim_enabled=1 qmap_mode=4
```

qmap_mode=0 indicates a single data call without UL AGG.
qmap_mode=1 indicates a single data with UL AGG.
qmap_mode=2/3/4 indicates multiple data call with UL AGG.

View device port loaded by the host, where /dev/mhi_DUN is the AT port.

3. Connect the module to the host via PCIe.

4. Set the module NIC data call method to MBIM with **AT+QCFG="pcie_mbim",1**. For details of the AT command, see ***document [1]***.

5. Start a data call with quectel-CM.

   1) Execute **quectel-CM -h** to check help information of quectel-CM.
   2) Refer to the following examples in /log directory of quectel-CM source code package.

pcie_mhi_mbim.txt	Local single data call with QMAP
pcie_mhi_mbim_qmap=4.txt	Local multiple data call with QMAP

NOTE

1. In the context of PCIe_MHI_MBIM data calls, QMAP is the multiple data call function of the PCIe_MHI_MBIM driver, which is different from the QMAP data call introduced in ***Chapter 3.10***.
2. QMAP feature of the module is enabled by default, i.e., even if *qmap_mode* is not configured when loading the driver, *qmap_mode=1* will be set automatically when running the driver.
3. Before connecting the module to the host, make sure that pcie_mhi.ko driver provided by Quectel is installed.

3.9. ECM Data Call (USB + AP + Linux)

1. Contact Quectel Technical Support for the installation package of the following driver: Quectel_Linux_USB_Serial_Option_Driver (Driver for TTY port, such as AT and DM.)

2. Integrate and compile the above driver on the host by executing **make install** in the driver file.

3. Connect the module to the host via USB cable.

4. Set the module data call method to ECM with **AT+QCFG="usbnet",1** via AT port /dev/ttyUSB2 (the

third TTY port of the module) when booting the module for the first time. For details of the AT command, see ***document [1]***.

5. Reboot the module.

6. After the module is rebooted, the host matches the Linux original ECM driver automatically and generates a virtual Ethernet interface usb0 based on USB. The module performs a data call internally and provides network access to the host via NAT. The host only needs to send DHCP to ECM network card to request the IP address.

7. To enable multiple data call in Linux based on ECM, see ***document [1]***.

NOTE

1. Please contact Quectel Technical Support to confirm whether the ECM data call on Linux system can be used for commercial purposes.
2. Do **NOT** use Linux original driver qcserial.ko on the host when using Quectel modules. You can check if the Linux original drivers are installed by running **lsmod**. If the drivers are installed, you should run **rmmod** to uninstall them. Before connecting a Quectel module to the host, make sure that driver option.ko provided by Quectel is installed.

3.10. QMAP Data Call (RGMII/PCIe RC + AP)

1. Contact Quectel Technical Support for the installation package of the following driver:
   Quectel_Linux_USB_Serial_Option_Driver (Driver for TTY port, such as AT and DM.)

2. Integrate/compile the above driver on the host by executing **make install** in the driver file.

3. Reboot the module and configure the target interface.

   - If RGMII is used:
     1) Enable RGMII network card with **AT+QETH="rgmii"**. For details of the AT command, see ***document [1]***.
     2) Enable the data call with **AT+QMAP**. You can also enable IPPassthrough as needed with **AT+QMAP**. For details of the AT command, see ***document [1]***.

   - If PCIe_RC_ETH such as PHY chip RTL8125PHY or AQC107 is used:
     1) Enable the corresponding PHY driver with **AT+QETH="eth_driver"**. For details of the AT command, see ***document [1]***.
     2) Enable the data call with **AT+QMAP**. You can also enable IPPassthrough as needed with **AT+QMAP**. For details of the AT command, see ***document [1]***.

NOTE

Do **NOT** use Linux original driver qcserial.ko on the host when using Quectel modules. You can check if the Linux original drivers are installed by running **lsmod**. If the drivers are installed, you should run **rmmod** to uninstall them. Before connecting a Quectel module to the host, make sure that driver option.ko provided by Quectel is installed.

4 Appendix References

Table 3: Related Documents

Document Name
[1] Quectel_RG520N&RG525F&RG5x0F&RM5x0N_Series_AT_Commands_Manual

Table 4: Terms and Abbreviations

Abbreviation	Description
AP	Application Processor
DHCP	Dynamic Host Configuration Protocol
ECM	Ethernet Control Model
ETH	Ethernet
IP	Internet Protocol
MBIM	Mobile Broadband Interface Model
MHI	Modem Host Interface
NAT	Network Address Translation
NDIS	Network Driver Interface Specification
PCIe EP	PCI Express Endpoint Device
PCIe RC	PCI Express Root Complex
QMI	Qualcomm Message Interface
RGMII	Reduced Gigabit Media Independent Interface
TE	Terminal Equipment

UL AGG	Uplink Aggregation
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WWAN	Wireless Wide Area Network