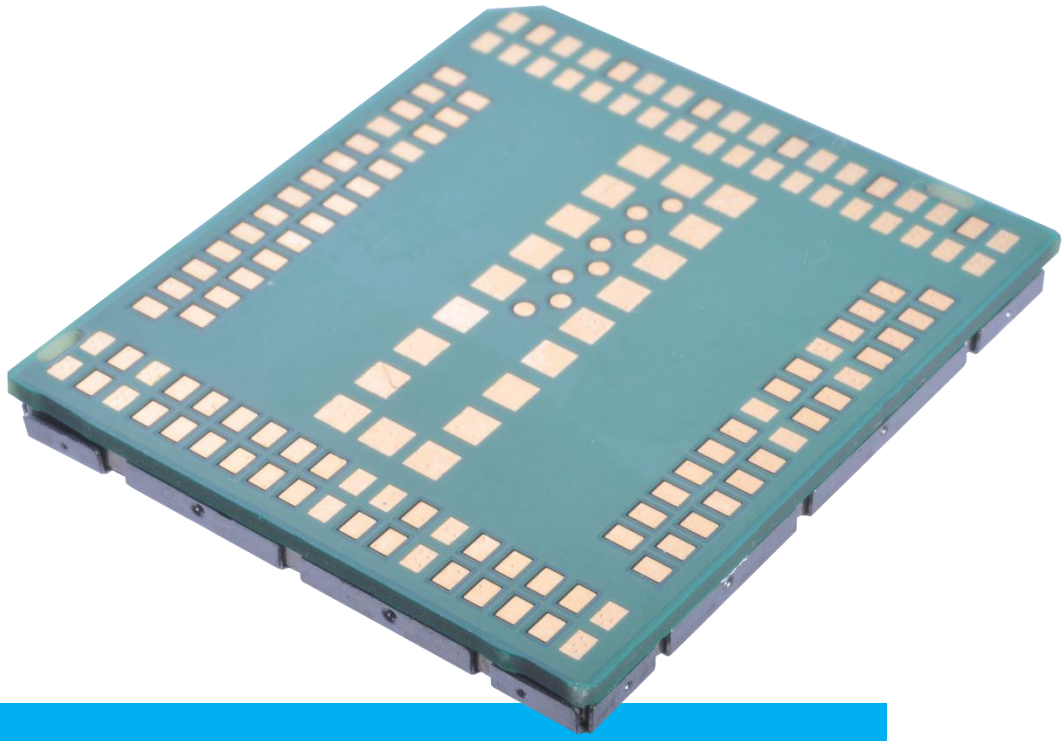


Software Developer Guide

Project Name: M14A2A/M18Q2(M14Q2)-Series

Author: Wistron NeWeb Corporation

Revision: 1.3.4

Revision Date: 2016/08/16

Contact Information

Technical Support Website	https://SupportIoT.wnc.com.tw
Company Website	www.wnc.com.tw

Revision History

Rev. #	Author	Summary of Changes	Date
0.8	WNC MBU	Initial release for M18Q2 software developer guide	2016/1/26
1.1	WNC MBU	Refine software developer guide for merging M18Q2 and M14A2	2016/3/18
1.2	WNC MBU	Change Format	2016/4/28
1.3	WNC MBU	Refine some descriptions and correct part of FOTA	2016/06/08
1.3.1	WNC MBU	Remove AT@INTERNET command and the related parts in the document.	2016/07/22
1.3.2	WNC MBU	Modify for GBD's comments.	2016/08/08
1.3.3	WNC MBU	Add abbreviation table in 6.1.	2016/08/09
1.3.4	WNC MBU	Modify for GBD's comments	2016/08/16

Contents

1. Introduction

This document introduces the WNC data module and all the other modules using it in the WNC product line.

This document also describes the required background knowledge about the WNC data module and the reference guides for software developers who will integrate their external host processors with the WNC data module.

The WNC data module can be configured to several types of configurations for different external host processors that communicate and send/receive data to/from the Internet. The basic concept is that the WNC data module provides proper interfaces for its control and for data communication traffic, which supports as many external host processors with different network capabilities. *Figure 1* depicts the basic concept's architecture.

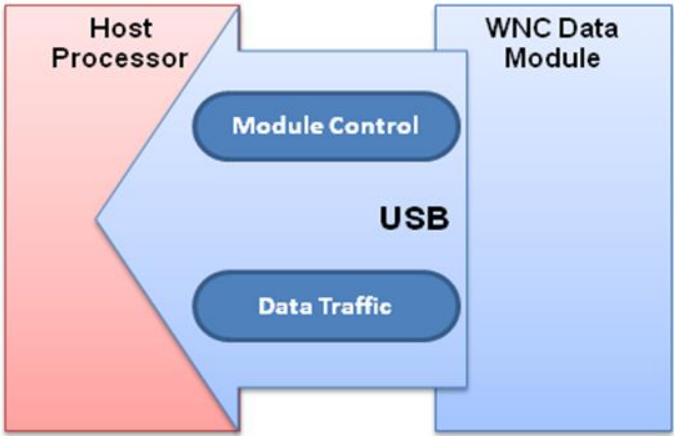


Figure 1.

The architectures of individual interfaces for the external host processors are described in the following sections.

1.1. Three Types of Interface Configurations

The WNC data module supports three types of interface configurations for external host processors with different capabilities of network connection:

- QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.
- AT Commands to control a USB CDC-ACM device and data traffic over a USB

CDC-ECM interface.

- AT commands for control and data traffic transmitted through a USB CDC-ACM device or UART.

Individual interface types are defined in section 1.3. Type Definitions.

1.2. System Requirements

The requirements of the external host processor connecting to the WNC data module must be installed or implemented during system initialization. If these requirements are not properly installed or implemented by the external host processor, the WNC data module may not be detected and will not work as described in the documents. Please ensure that the requirements of the following sections are understood and the prerequisite requirements are fulfilled in the external host processor. The possible integration architectures of the external host system and the WNC module system are depicted in *Figure 1.2*.

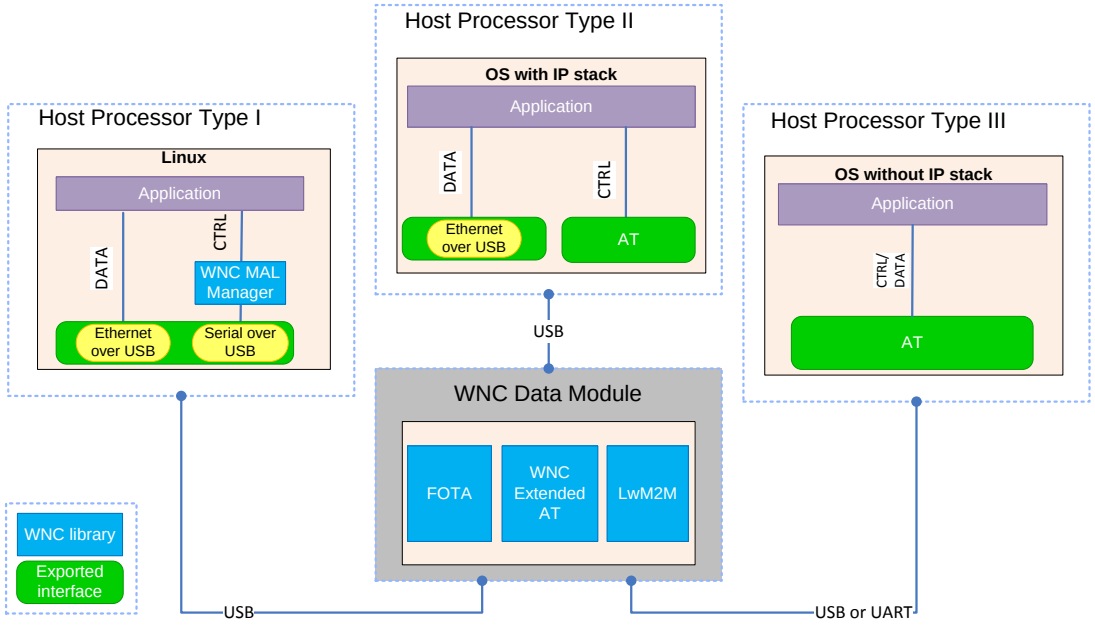


Figure 1.2

The external host processor must act as a USB host and have the proper interface drivers for individual types of interfaces what the WNC data module is used. The USB of the external host processor must support a USB host mode compatible with USB 2.0 and can recognize WNC's USB VID/PID.(Please check the "Contact Information" in this document for getting

more details about this topic.)

Individual interface types are configured by the specific initialization sequences. The initialization sequences are based on the proper USB interface drivers installed. After the proper USB interface drivers are installed, the external host processor begins detecting the enumeration of the USB device and recognizes the WNC data module. (if a correct power sequence is provided from the external host processor. Please check the "Contact Information" in this document for getting more details about this topic.) For different types of interface configurations, the external host processor must complete the dedicated initialization sequence which initializes the WNC data module into a proper state. The initialization sequences are different for individual interface types and will be described in section "2.1. Prerequisite Requirements".

1.3. Type Definitions

The three types of interfaces for external host processors are defined as follows:

Type I: QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.

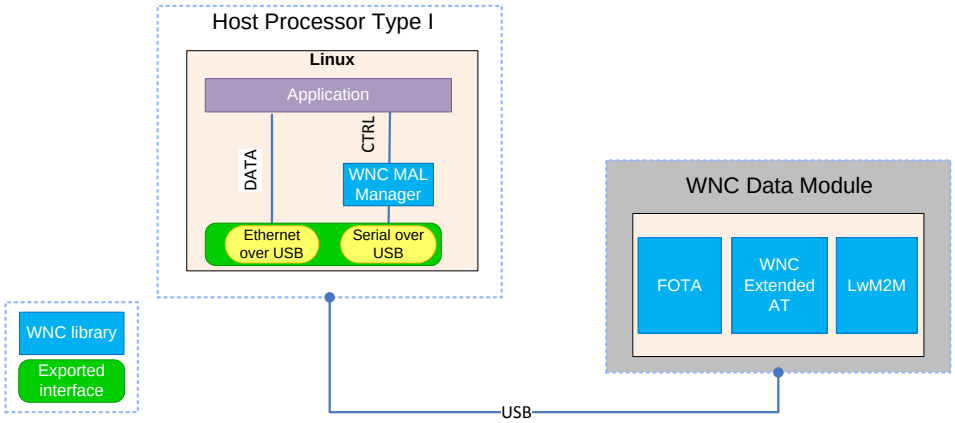


Figure 1.3.1

Under this type of interface, QMI and RmNet are proprietary stacks for it. WNC software encapsulates the QMI and RmNet proprietary stack as a MAL Manager SDK for easy integration with the application layer. Currently, this encapsulation software stack only supports external host processors using the Linux operating system.

The MAL Manager SDK is based on the Qualcomm QMI messages and RmNet interface design but with an application-friendly interface named as the "MAL

Manager API" in JSON format. The WNC MAL Manager SDK provides a JSON format generator for the application layer to generate the MAL Manager API. It is easily ported for web applications such as a HTTP servers (usually used with web-based configurations in wireless gateways or sensor devices) or LWM2M for device management. Details of the MAL Manager API are provided in the MAL Manager SDK document "**WNC MAL Manager JSON API**".

The WNC MAL Manager SDK needs the relative RmNet driver which will be also integrated into the software stack of the external host processor. The RmNet driver emulates a network card device in the Linux network subsystem and exports itself as a regular Ethernet network card interface. It serves as an IP-based interface for a DHCP client and is compatible with socket-layer operations in Linux. WNC will assist customers who prefer this type of interface to integrate the WNC MAL Manager SDK into the software stack in the external host processor according to the MAL Manager SDK document "**WNC MAL Manager Developer Guide**" by means of WNC's porting experience on the Linux platform.

If the external host processor is a wireless gateway/router, an LTE access point, or a network product of this type (especially if the external host processor is with the Linux operating system), this interface type of the WNC data module is the best alternative to configure with the external host processor.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

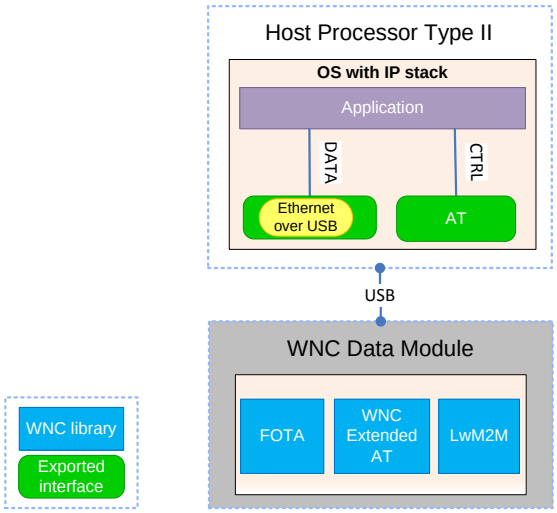


Figure 1.3.2

For the external host processors with TCP/IP protocol stack, this type of interface

provides AT commands via a USB CDC-ACM device interface, as with the traditional RS232 circuitry, to establish data communication over the mobile network. When a data call to a communication service is made successfully, the CDC-ECM interface becomes available for sending and receiving data-packets. The external host processor must request an IP by using a DHCP client via this CDC-ECM interface. After obtaining an IP address, the data packets can be transmitted and received through this CDC-ECM interface to the Internet.

The WNC data module supports the AT command set defined by the 3GPP standard for the control of the data module and data communication over the mobile network. Refer to the document "**WNC AT Command Guide**" for details.

If the external host processor can communicate with certain 3G modules and the developers have experience with the AT command set defined by the 3GPP standard, then this is the best type of configuration.

Type III: AT commands for control and data traffic transmitted through a USB CDC-ACM device or UART.

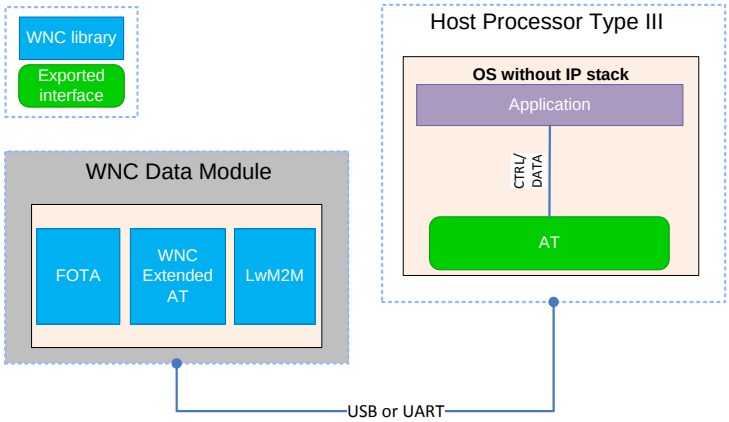


Figure 1.3.3

This type of interface emulates the traditional RS232 circuitry over USB but supports an IP-based proprietary AT command set for data communication of the external host processor without a TCP/IP software stack. The IP-based proprietary AT commands provide the socket-like operations via a TCP/IP software stack embedded in the WNC data module for the Internet connection(Refer to 7. Reference R1). The IP-based proprietary AT commands also provide several popular high-level protocols and network tools, such as FTP, PING, and DNS resolving - all over TCP/IP.

Detailed specifications of WNC's proprietary IP-based AT command set is described in the document "**WNC AT Command Guide**".

If the external host processor is a simple device that does not have the functionality for data communication and will integrate the module to connect to the Internet directly, implementing the TCP/IP steps above becomes unnecessary.

2. Configuration for The Module Interface

2.1. Prerequisite Requirements

The external host processors work with these interfaces which rely on the functionalities provided by USB/UART drivers on the external host processors themselves. These drivers are parts of the software on the external host processors and must be prepared and configured properly before powering up the WNC data module.

Type I: QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.

- The software in the external host processor must be compatible with the Linux operating system.
- The USB in the external host processor must be USB host mode compatible with USB 2.0 and recognize WNC's USB VID/PID.(Please check the "Contact Information" in this document for getting more details about this topic.)
- The external host processor must have a TCP/IP protocol stack and DHCP client functionality.
- The WNC MAL Manager SDK requires the compiler tools of the external host processor for building the binary executable components. Refer to the document "**WNC MAL Manager Developer Guide**" for additional details when working with WNC software.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

- The USB in the external host processor must be USB host mode compatible with USB 2.0 and recognize WNC's USB VID/PID.(Please check the "Contact Information" in this document for getting more details about this topic.)
- The external host processor must have a TCP/IP protocol stack and DHCP client functionality.

Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

- The USB in the external host processor must be USB host mode compatible with USB 2.0 and recognize WNC's USB VID/PID.(Please check the "Contact Information" in this document for getting more details about this topic.)
- If the external host processor connects with the WNC data module via UART, please verify the UART driver is ready and has been configured correctly under the settings. The default settings are 115,200bps for default baud rate, 8 bits for data width, parity check disabled, one bit for stop bit width, and no flow control enabled.

2.2. Driver

Type I: QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.

Package Release (**Note:** * indicates M14A2 only; ** indicates M18Q2 only)

WNC software will provide two USB drivers for this type of interface for each model:

- Ethernet device driver: rmnet_usb[**]
- Serial device driver: qcserial[**]

  Or
- Ethernet device driver: cdc_ether[*]
- Serial device driver: cdc-acm[*]

Each driver can be built as a Linux kernel module or directly embedded in the Linux kernel. If the kernel module is preferred, the driver will use a .ko file format. Please

insert these kernel modules into the external host system at the proper time. If drivers are embedded in the Linux kernel, all items are embedded in a binary software image of the external host processor.

Note: Due to the possibility of different versions of Linux in the external host system, the USB subsystem of Linux could have changes in its architecture. A questionnaire providing feedback information regarding the external host platform will be necessary. As a result of the completion of the questionnaire, WNC will release patches based on the Linux version in the external host processor for integration. (Please check the "Contact Information" in this document for getting the patches from the WNC software supporting.)

There are several .patch files included in the release package. These files are patches for patching into your Linux source. They are included in formal patch-file formats by the diff tool usually used in the Linux system. Please execute the command below under the root folder of the Linux source:

- patch -p1 <filename.patch>
  <filename.patch> is the name of the patch file.

Every patch file contains a sequential number in front of its file name as a prefix. **Note:** Please follow the sequence to patch these files one-by-one.

Linux Configuration (**Note:** * indicates M14A2 only; ** indicates M18Q2 only)

The Linux system build provides many configurations for different features and platforms. The module drivers are components of the Linux drivers. Correct configurations should be set up when compiling.

- Ethernet device driver: CONFIG_MSM_RMNET_USB[**]
- Serial device driver: CONFIG_USB_SERIAL_QUALCOMM[**]
  or
- Ethernet device driver: CONFIG_USB_NET_CDCETHER[*]
- Serial device driver: CONFIG_USB_ACM[*]

Please enable the above configurations to be "y" for embedding in Linux or "m" for kernel modules. You can easily execute "make menuconfig" in the command line under the root folder of the Linux kernel source. The configurations are depicted in the figure below.

For qcserial drivers[**]:

```
.config - Linux/mips 3.10.12 Kernel Configuration
> Device Drivers > USB support > USB Serial Converter support
                                                    USB Qualcomm Serial modem
  CONFIG_USB_SERIAL_QUALCOMM:

  Say Y here if you have a Qualcomm USB modem device.  These are
  usually wireless cellular modems.

  To compile this driver as a module, choose M here: the
  module will be called qcserial.

  Symbol: USB_SERIAL_QUALCOMM [=m]
  Type  : tristate
  Prompt: USB Qualcomm Serial modem
    Location:
      -> Device Drivers
        -> USB support (USB_SUPPORT [=y])
          -> USB Serial Converter support (USB_SERIAL [=m])
    Defined at drivers/usb/serial/Kconfig:509
    Depends on: USB_SUPPORT [=y] && USB [=y] && USB_SERIAL [=m]
    Selects: USB_SERIAL_WWAN [=m]
```

The configuration settings can be accessed using the following pathway: Device Drivers→USB support→USB Serial Converter support→USB Qualcomm Serial modem

Note: The Linux driver system has a hierarchical structure with certain types of dependencies between drivers. You should ensure that the below drivers are also inserted into the external host platform if they are kernel modules. Drivers are accessed using the following pathway: Device Drivers→USB support→USB host driver

For rmnet_usb drivers[**]:

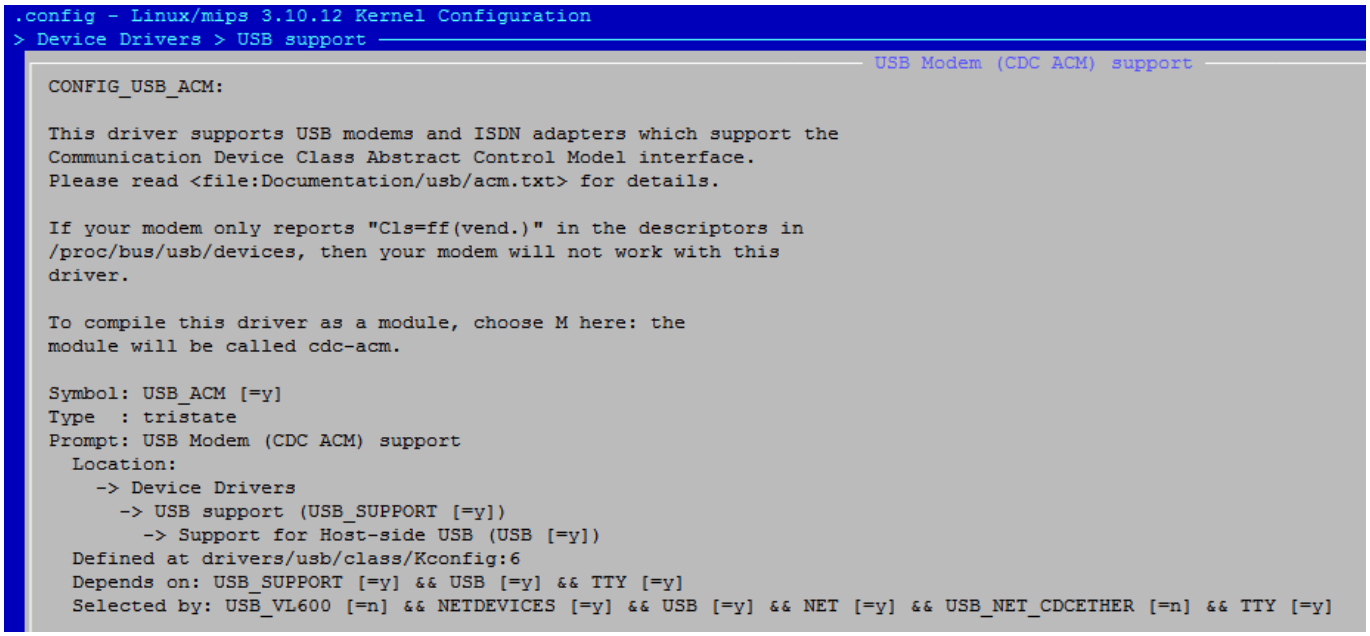
```
.config - Linux/mips 3.10.12 Kernel Configuration
> Device Drivers > Network device support > USB Network Adapters
                                                    RMNET USB Driver
  CONFIG_MSM_RMNET_USB:

  Select this if you have a Qualcomm modem device connected via USB
  supporting RMNET network interface.

  To compile this driver as a module, choose M here: the module
  will be called rmnet_usb. If unsure, choose N.

  Symbol: MSM_RMNET_USB [=m]
  Type  : tristate
  Prompt: RMNET USB Driver
    Location:
      -> Device Drivers
        -> Network device support (NETDEVICES [=y])
          -> USB Network Adapters
    Defined at drivers/net/usb/Kconfig:536
    Depends on: NETDEVICES [=y] && USB [=y] && NET [=y] && USB_USBNET [=m]
```

The above configurations can be accessed using the following pathway: Device

Drivers→Network device support→USB Network Adapters→RMNET USB Driver

Note: The Linux driver system has a hierarchical structure with certain types of dependencies between drivers. You should ensure that the below drivers are also inserted in the external host platform if they are kernel modules. Drivers are accessed using the following pathway: Device Drivers→USB support→USB Wireless Device Management support

For cdc-acm drivers[*]:



The configuration settings can be accessed using the following pathway: Device Drivers→USB support→USB Modem (CDC ACM) support

Note: The Linux driver system has a hierarchical structure with certain types of dependencies between drivers. You should ensure that the below drivers are also inserted into the external host platform if they are kernel modules. Drivers are accessed using the following pathway: Device Drivers→USB support→USB host driver

For cdc_ecm drivers[*]:

```
.config - Linux/mips 3.10.12 Kernel Configuration
> Device Drivers > Network device support > USB Network Adapters
                                    CDC Ethernet support (smart devices such as cable modems)
  CONFIG_USB_NET_CDCETHER:

  This option supports devices conforming to the Communication Device
  Class (CDC) Ethernet Control Model, a specification that's easy to
  implement in device firmware.  The CDC specifications are available
  from <http://www.usb.org/>.

  CDC Ethernet is an implementation option for DOCSIS cable modems
  that support USB connectivity, used for non-Microsoft USB hosts.
  The Linux-USB CDC Ethernet Gadget driver is an open implementation.
  This driver should work with at least the following devices:

    * Dell Wireless 5530 HSPA
    * Ericsson PipeRider (all variants)
    * Ericsson Mobile Broadband Module (all variants)
    * Motorola (DM100 and SB4100)
    * Broadcom Cable Modem (reference design)
    * Toshiba (PCX1100U and F3507g/F3607gw)
    * ...

  This driver creates an interface named "ethX", where X depends on
  what other networking devices you have in use.  However, if the
  IEEE 802 "local assignment" bit is set in the address, a "usbX"
  name is used instead.

  Symbol: USB_NET_CDCETHER [=n]
  Type  : tristate
  Prompt: CDC Ethernet support (smart devices such as cable modems)
    Location:
      -> Device Drivers
        -> Network device support (NETDEVICES [=y])
          -> USB Network Adapters
            -> Multi-purpose USB Networking Framework (USB_USBNET [=m])
    Defined at drivers/net/usb/Kconfig:188
```

The configuration settings can be accessed using the following pathway: Device Drivers→Network device support→USB Network Adapters→Multi-purpose USB Networking Framework→CDC Ethernet support

Note: The Linux driver system has a hierarchical structure with certain types of dependencies between drivers. You should ensure that the below drivers are also inserted into the external host platform if they are kernel modules. Drivers are accessed using the following pathway: Device Drivers→Network device support→ USB Network Adapters→Multi-purpose USB Networking Framework

When inserting the kernel module into the external host system (**Note:** * indicates M14A2 only; ** indicates M18Q2 only)

After producing the kernel modules, please insert them into the external host platform during runtime.

- insmod rmnet_usb.ko/modprobe rmnet_usb[**]
- insmod qcserial.ko/modprobe qcserial[**]

  or

- insmod cdc_ether.ko/modprobe cdc_ether[*]
- insmod cdc-acm.ko/modprobe cdc-acm[*]

You can ensure that they are inserted into the external host system using the following command:

lsmod

The two drivers above must appear in the list of the lsmod's response.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

Drivers prepared on the external host processor should meet the following type of interfaces:

This type of interface requires two USB drivers:

- USB standard CDC ECM interface
- USB standard CDC ACM interface

If the external host processor is not compatible with the Linux operating system, please ensure that the USB software stack supports these two USB interface drivers.

If the external host processor is compatible with the Linux operating system, you can enable the following two configurations in the Linux system build.

- CONFIG_USB_NET_CDCETHER
- CONFIG_USB_ACM

Start the drivers during runtime.

Please ensure these two drivers are ready when the WNC data module is powered on.

If the external host processor is compatible with the Linux operating system and these two drivers are built to be kernel modules, you can insert them using the following two commands:

- insmod cdc_ether.ko/modprobe cdc_ether
- insmod cdc-acm.ko/modprobe cdc-acm

You can ensure they are inserted into the external host system using the following command:

- lsmod

The two drivers above must appear in the list of the lsmod's response.

Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

Drivers prepared on the external host processor

This type of interface requires the following USB driver:

- USB standard CDC ACM interface

If the external host processor is not compatible with the Linux operating system, please ensure the USB software stack supports the USB interface driver.

If the external host processor is compatible with the Linux operating system, you can enable the following configuration in the Linux system build:

- CONFIG_USB_ACM

Start the driver during runtime.

Please ensure the driver is ready when the WNC data module is powered on.

If the external host processor is compatible with the Linux operating system and the driver is built to be kernel modules, you can insert it using the following command:

- insmod cdc-acm.ko/modprobe cdc-acm

You can ensure it is inserted into the external host system using the following command:

- lsmod

The driver above must appear in the list of the lsmod's response.

2.3. Initialization

Type I: QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.

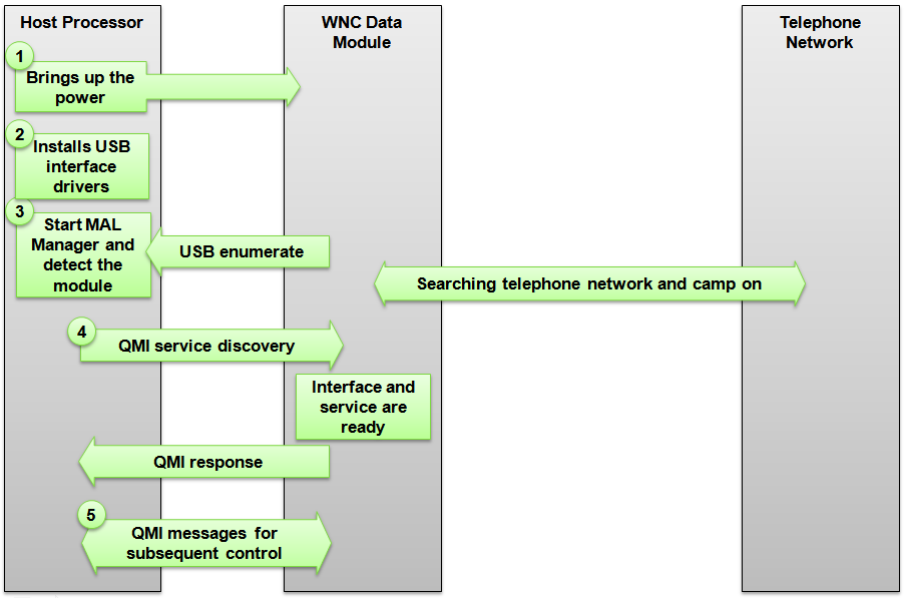
1. The external host processor powers on the WNC data module. Please refer to the WNC software and hardware for details.

2. The external host processor correctly installs the two USB interface drivers for the M18Q2 or M14A2 models, respectively, when the WNC data module undergoes a steady power-on.

3. The external host processor executes the WNC MAL Manager, and the WNC MAL Manager detects the enumeration of the WNC data module automatically.

4. The WNC MAL Manager running on the external host processor issues the first QMI message (normally a QMI message for service-compatibility discovery) to the WNC data module. The WNC data module receives the QMI message and verifies it as the instruction for the configuration of a type I interface. The WNC module proceeds with the necessary initializations and configurations and then responds to the WNC MAL Manager. The WNC MAL Manager confirms that the type I of the interface is activated.

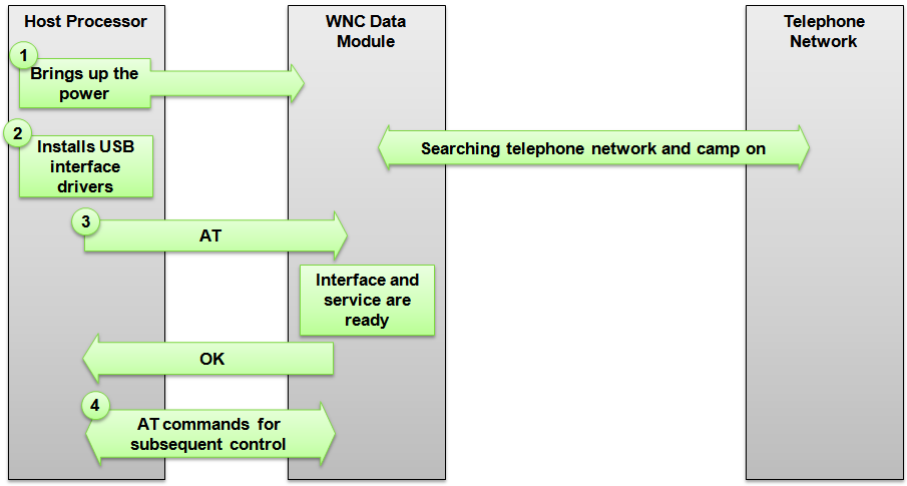
5. The WNC MAL Manager performs the necessary initializations and configurations to set up several services for the application layer. It also notifies the application layer of information regarding the WNC data module's status for subsequent controls.



Note: Detailed initialization of the MAL Manager is found in the document "**WNC MAL Manager Developer Guide**". Refer to the document for updated information.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

1. The external host processor powers on the WNC data module. Please check with WNC software and hardware for details.
2. The external host processor correctly installs the two USB interface drivers when the WNC data module undergoes a steady power-on.
3. The external host processor must configure USB CDC-ACM interface as a serial port with baud rate 115,200bps, 8 bits for data length, none parity check, 1 bit for stop bit, and no kind of flow control.
4. The external host processor should continuously send “AT” messages and waits for an “OK” response message responded via a USB standard CDC ACM interface to make sure the WNC data module is service-ready.
5. The external host processor can now send AT commands to perform the necessary initializations and configurations for subsequent controls.



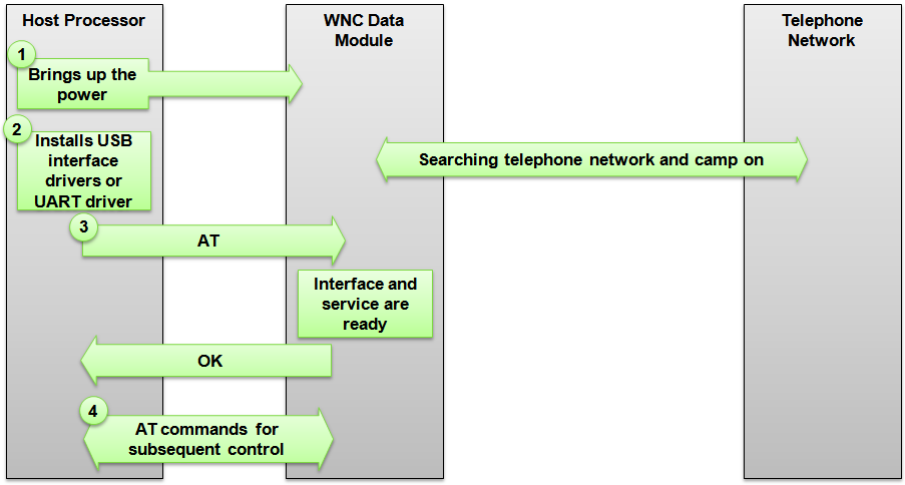
Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

1. The external host processor powers on the WNC data module. Please refer to the WNC software and hardware for details.
2. The external host processor correctly installs the two USB interface drivers when the WNC data module undergoes a steady power-on.

3. The external host processor must configure USB CDC-ACM interface as a serial port with baud rate 115,200 bps, 8 bits for data length, none parity check, 1 bit for stop bit, and no kind of flow control.

4. The external host processor should continuously send “AT” messages and waits for an “OK” response message responded via a USB standard CDC ACM interface or UART to make sure the WNC data module is service-ready. The interface activated depends on which hardware interface, USB CDC-ACM or UART, the first “AT” is sent to. Once one of USB CDC-ACM and UART is activated, the other one will be inactivated automatically.

5. The external host processor can now send AT commands to perform the necessary initializations and configurations for subsequent controls.



3. MAL Manager

3.1. Introduction

The MAL Manager is a software package running on the external host processor provided by the WNC MAL Manager SDK. It assists the external host developers to control the WNC data module and to easily establish data communication. It is not necessary for the external host software to process the detailed QMI message or event flows between the external

host processor and the WNC data module. The detailed parameters of the QMI message that the external host developers require and the background knowledge of mobile network technologies which is necessary to understand how to configure them is encapsulated by MAL Manager API or handled by MAL Manager. The MAL Manager and the MAL Manager API replace the complicated QMI message and event handling with the configuration style of the MAL Manager API, which is easier to be used by applications and is more human-readable.

The MAL Manager greatly simplifies the difficulty in controlling the data module, establishment of data communication, and monitoring the status of the mobile network. This may save substantial development time when implementing software for controlling the module and establishing data communication. External host developers can then focus on the applications they want to implement.

The system architecture is depicted in *Figure 3.1* below.

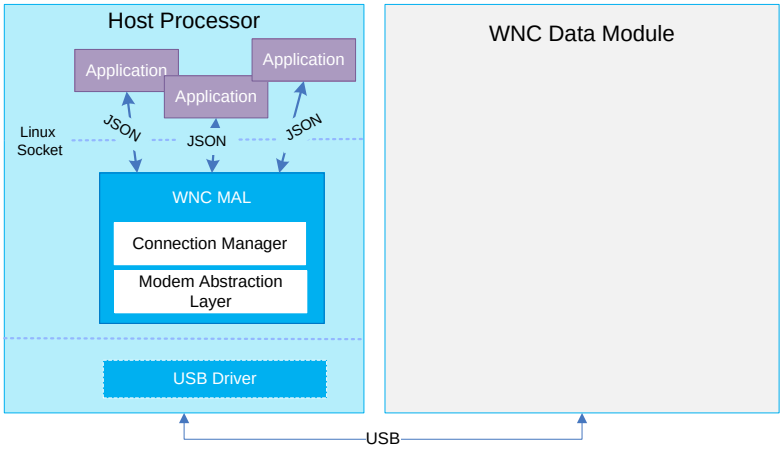


Figure 3.1

Figure 3.2 is an example which illustrates how the MAL Manager simplifies the substantial implementation efforts of the external host developers who try to establish data communication via the WNC data module.

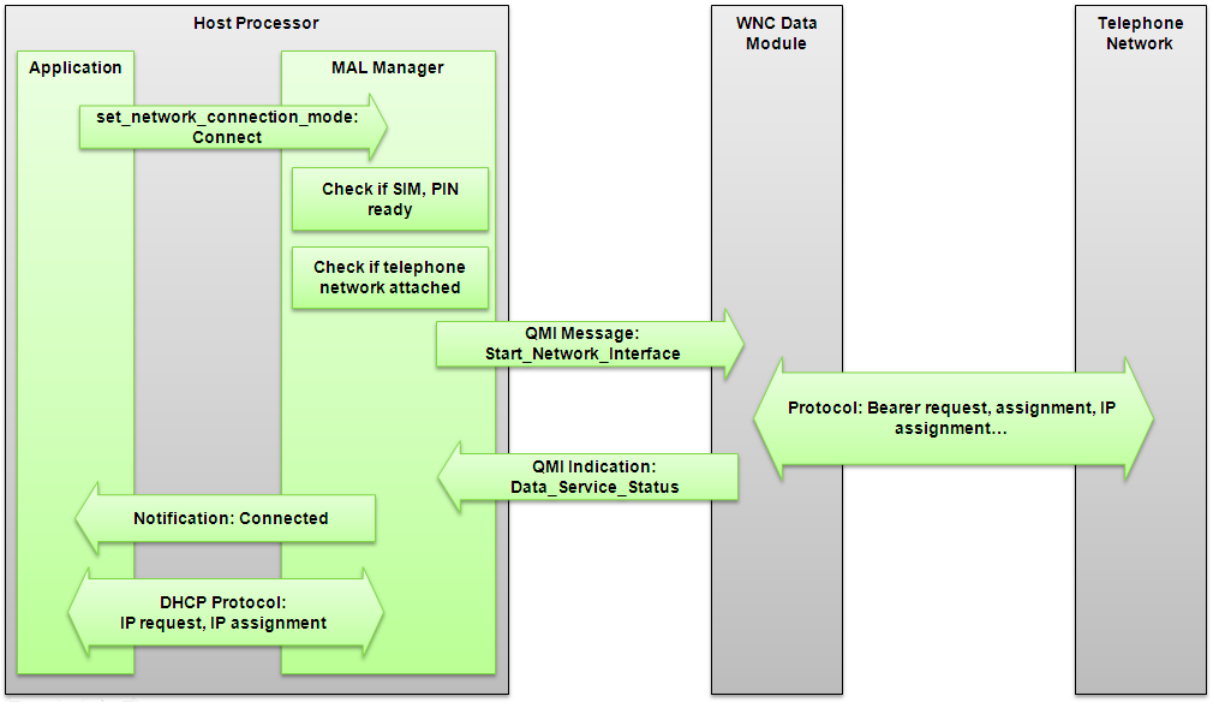


Figure 3.2

3.2. Features

- Only supports the Linux operating system.
- API with JSON format for easy using from the application layer.
- Processes the details of the mobile network to set up a data communication.
- The SDK provides a sample web server for the reference of a router/gateway product.

3.3. References

- Refer to the document "**WNC MAL Manager Developer Guide**" for details.

4. Firmware Upgrade Over The Air (FOTA)

4.1. Introduction

The firmware upgrade of a traditional premise device is performed by the manufacturing vendor or the customer service direct center. The devices with data communication with the Internet can download new firmware from the server and upgrade it. This is called “Firmware upgrade Over The Air”, or “FOTA”.

The WNC data module contains a firmware upgrade over the air mechanism for upgrading the WNC data module itself or even the external host processor. An embedded Light Weight M2M(LWM2M) client is tasked with communicating with the server maintained by the manufacturing vendor or the mobile network operator. The firmware upgrade is triggered automatically on demand from the server. It is not necessary for the external host processor to process the details of the firmware download and integrity check, but it will process the notification from the WNC data module. This is an easy way for the external host processor performing the firmware upgrade and allows the remote server to trigger the firmware upgrade process.

4.2. Features

- Embedded Light-Weight M2M client for device remote control.
- Firmware upgrade automatically performed on-demand from the server.
- Integrity check for ensuring the security of device firmware.
- Embedded storage to save the external host processor’s storage costs.

4.3. Light Weight M2M

4.3.1. System Architecture

The Firmware Upgrade Over The Air is based on the Light Weight M2M architecture. The LWM2M server is the remote server which monitors the status of devices and pushes messages to instruct devices for management. The Firmware Upgrade Over The Air is one

feature of the Light Weight M2M architecture. The architecture of the Light Weight M2M is depicted in *Figure 4.3*.

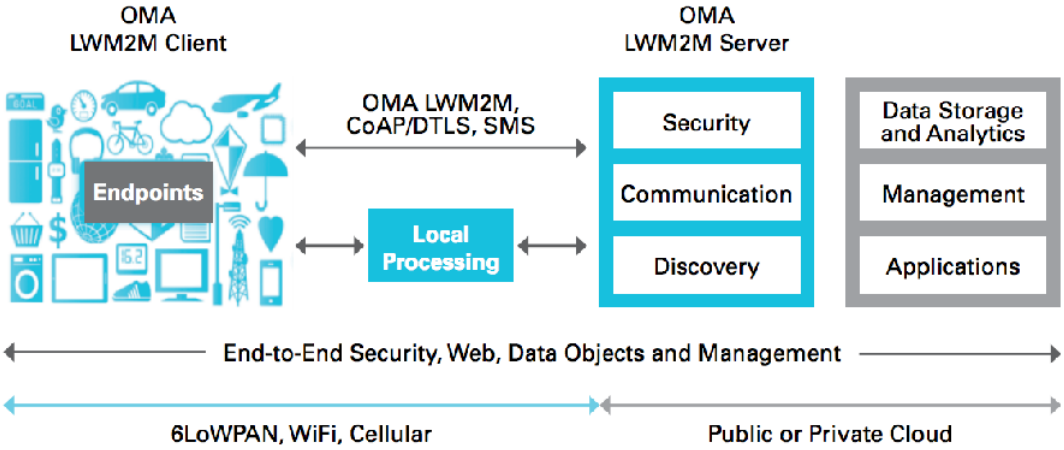


Figure 4.3

4.3.2. Supporting Features

- Data object[#1] for monitoring device status
- Data object and management for firmware upgrade

Note #1: An data object is a data unit for exchanging information, events, requests responses between the LwM2M client and server.

4.3.3. Module Firmware Upgrade Workflow

Type I: QMI messaging that controls over a USB serial device and monitors data traffic over a USB Ethernet device.

Refer to the document "**WNC MAL Manager Developer Guide**".

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

The flow is the same as the Type III interface. Refer to *Figure 4.3.3.1, 4.3.3.2, and 4.3.3.3.*

Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

- FOTA for the module:

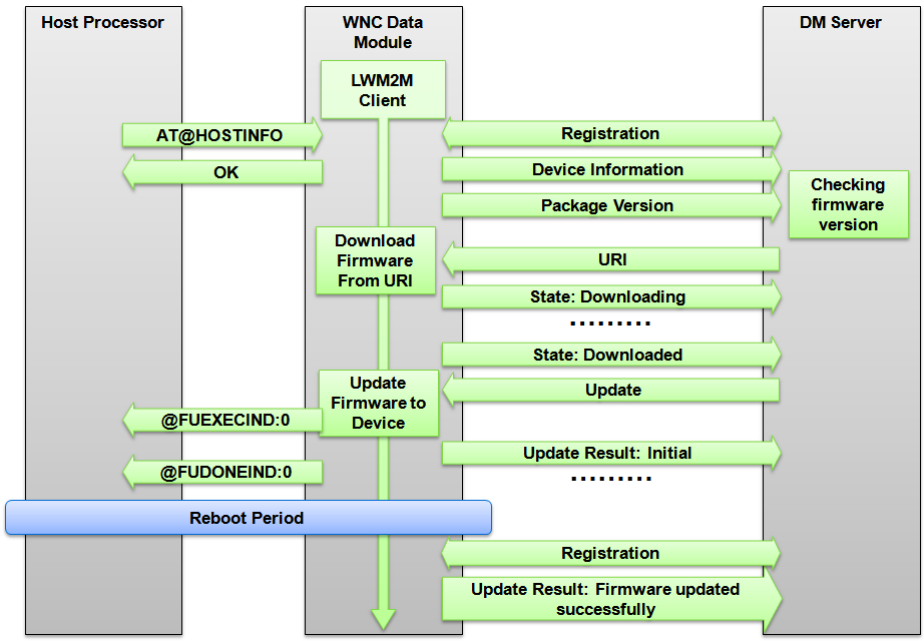


Figure 4.3.3.1

- FOTA for the external host processor:
  - When downloading firmware:

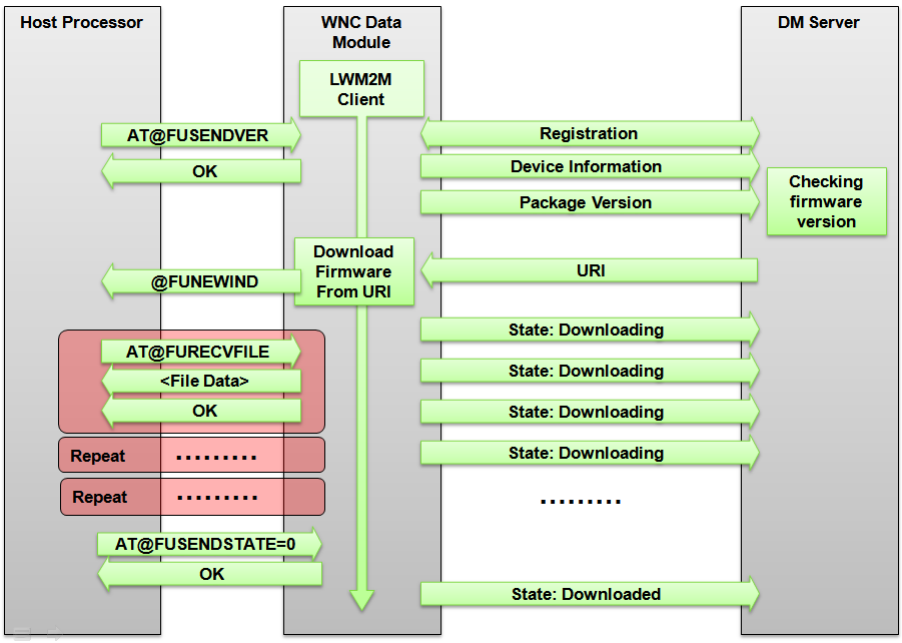


Figure 4.3.3.2

- When updating firmware:

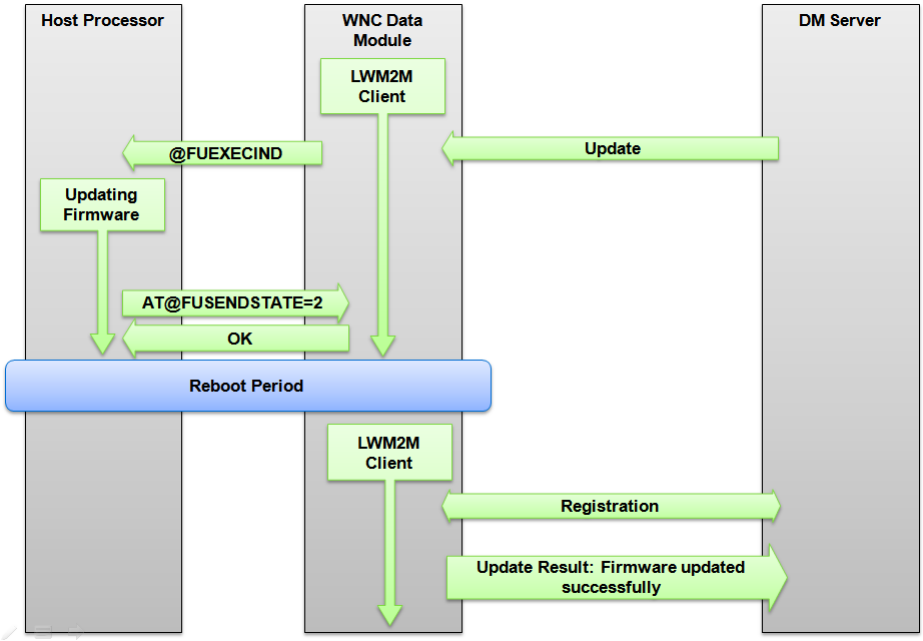


Figure 4.3.3.3

5. Setup Sequence

For the QMI message for control over a USB serial device and data traffic over a USB Ethernet device within the type I interface, please refer to the document "**WNC MAL Manager Developer Guide**" for details. The following sections just describe the initial sequence for Type II and Type III interface.

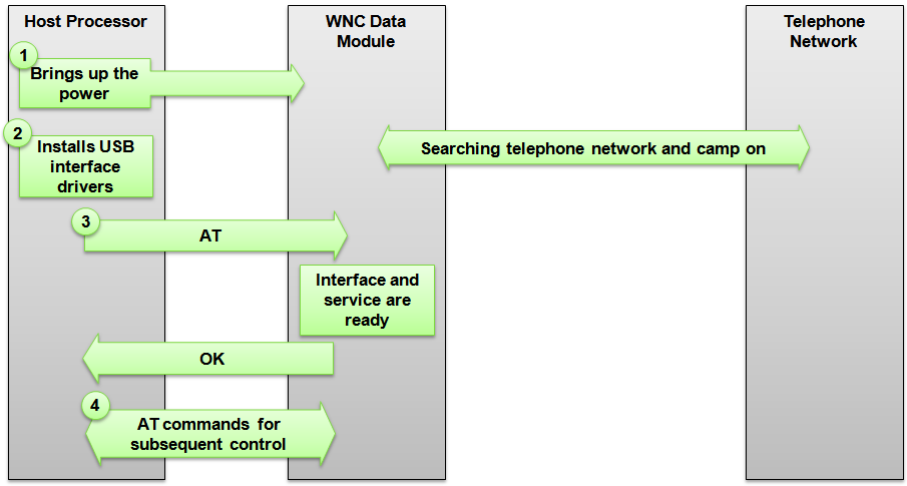
5.1. Initialization

After bringing up the power of the WNC data module, the external host processor must initialize the specific USB interface drivers and issue proper AT commands to detect if the specific interface is ready for serving.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

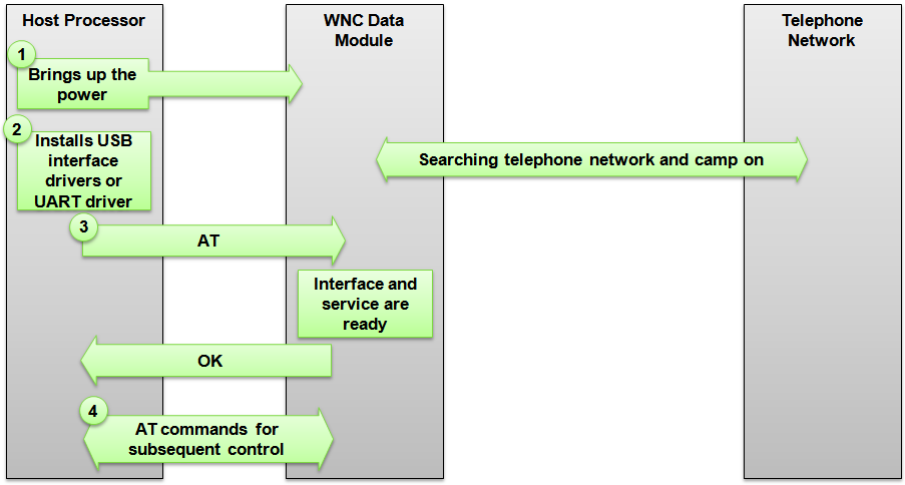
The external host processor must initialize a USB standard CDC ACM and CDC ECM

interface driver. After detecting the USB interface enumeration of the WNC data module, the external host processor must continuously send “AT” messages via the USB CDC ACM interface and waits for any “OK” message from the WNC data module. If the “OK” is correctly received, this is confirmation that the WNC data module is ready in its interface and services.



Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

The external host processor must only initialize a USB standard CDC ACM interface driver or its proprietary UART driver. After detecting the USB interface enumeration of the WNC data module, the external host processor must continuously send “AT” messages via the USB CDC ACM interface or UART and waits for any “OK” message from the WNC data module. If the “OK” is correctly received, this is confirmation that the WNC data module is ready in its interface and services.



5.2. PIN And Personalization

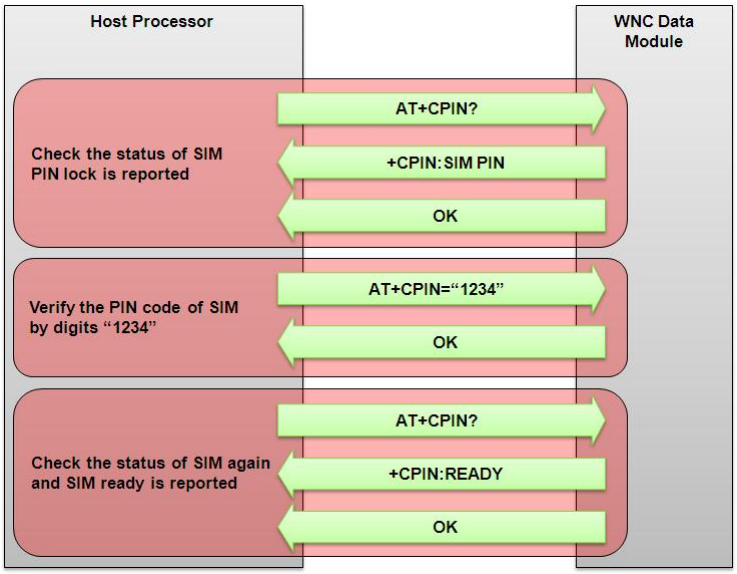
After switching to the specific interface, the SIM card may or may not be inserted, or the PIN lock of the inserted SIM has been enabled. The software of the external host processor must detect these situations and begin the procedure to verify the PIN for PIN lock. Nevertheless, some types of SIM cards or device personalization features will be implemented in the WNC data module that have been requested from the mobile network operator for business strategies. The external host processor should employ the following method to verify a PIN lock. This procedure is identical for the Type II and Type III interface configurations:

1. Send “AT+CPIN?” to query the SIM PIN status.
2. If “READY” is issued, this indicates that no PIN lock has been enabled. It is not necessary to send “AT+CPIN=<your_pin>” to verify a PIN lock; do not proceed with the following steps.
3. If “SIM PIN” is issued, send “AT+CPIN=<your_pin>”, where “<your_pin>” is the PIN used for the PIN lock.

4. If “SIM PUK” is issued, send “AT+CPIN=<your_pin>,<new_pin>”, where “<new_pin>” is a new PIN to replace the old PIN. This will verify the PUK lock and setup of a new PIN.

Note: “<your_pin>” and “<new_pin>” should each be included within double quotation marks when sending the related commands.

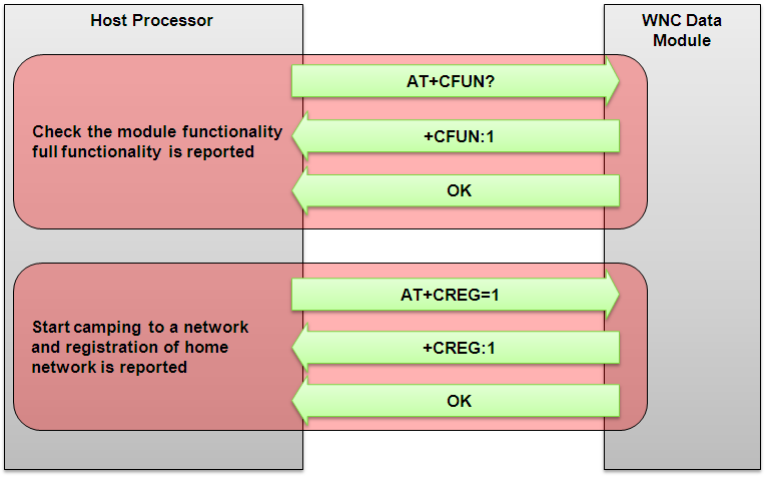
Example:



5.3. Establish and Disconnect Data Service

Before issuing the AT command to begin data service, “AT+CFUN” and “AT+CREG” should be used to check the device functionality and to instruct the WNC module to remain on the mobile network. After remaining on the mobile network, some parameters should be configured properly for setting the profile of the PDP context. It is generally not necessary to configure these parameters because they are provided by default by the SIM card present from the mobile network operator. If this is the case, the external host processor may skip this setup sequence to send AT commands to establish data service.

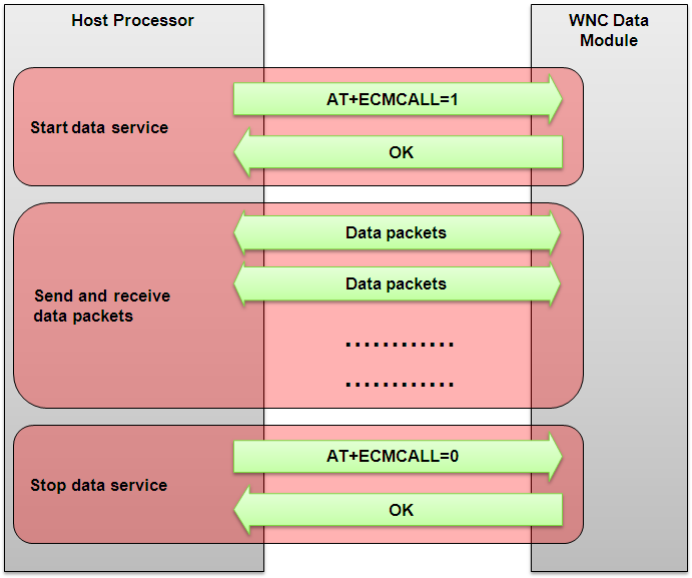
Example:



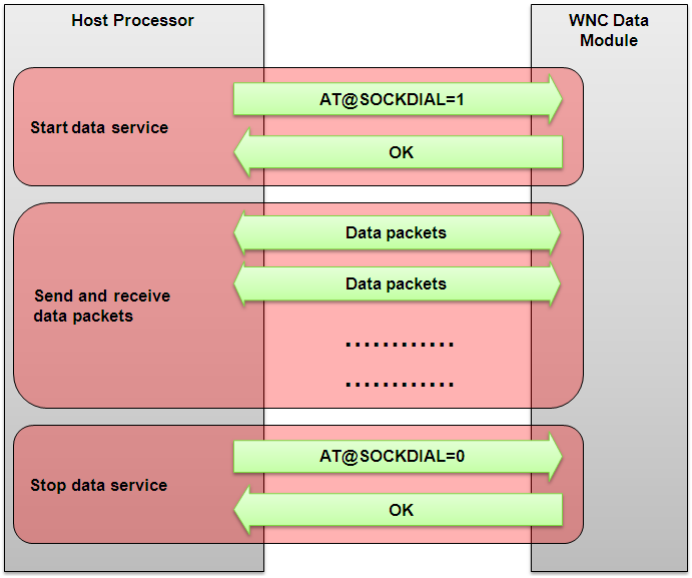
If connected on the mobile network, the external host processor can begin to set up the WNC data module for establishing data service. The Type II and Type III interface configurations use different AT commands to establish and disconnect the data service.

The following flowcharts depict the Type II and Type III interface configurations.

Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.



Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.



5.4. Transferring and Receiving Data Packets

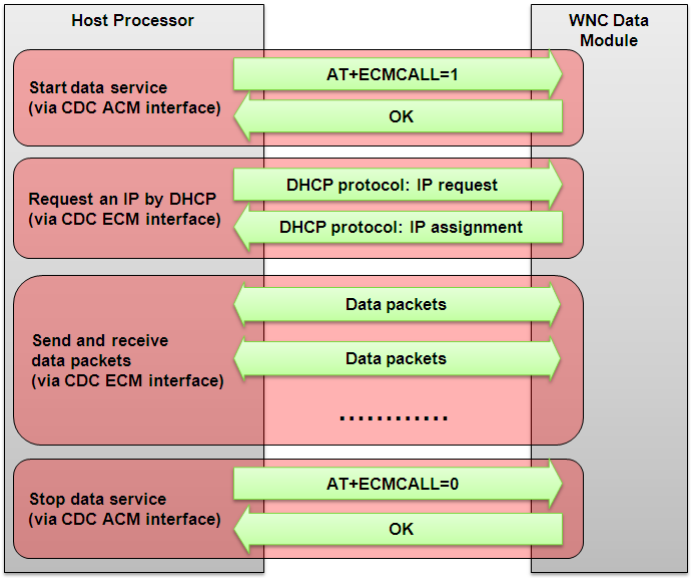
Data packets which exchange data end-to-end are described in this section. For the Type II and Type III interface configurations, their packets utilize a different level in protocols. Type III interface configuration is dedicated for an external host processor, which does not contain a TCP/IP protocol in its software stack and provides socket-like AT commands for the developers with experience programming or widely used sockets. This reduces developers' time spent on the integration of a TCP/IP protocol stack and the cost of software, especially for devices in the IoT that previously did not require an Internet connection, such as a TV, a refrigerator, or even a light bulb.

Type II interface configuration operates in the same manner as traditional data communication within 2G/3G mobile networks. A user places a call for the data service provided from the mobile network, however, there is no need to process PPP protocol via a USB CDC ACM interface to reach a preset gateway in the mobile network for connecting to the Internet. A PPP connection is not necessary in the modern mobile network with all-IP-base core network. On the other hand, this Type II interface provides a USB Ethernet interface as the traditional Ethernet card device connecting with the external host processor. The external host processor still must execute a DHCP client to request an IP from the module via the USB Ethernet interface. After obtaining an IP, the external host processor can begin sending and receiving data packets.

The following examples illustrate the Type II and Type III interface configuration.

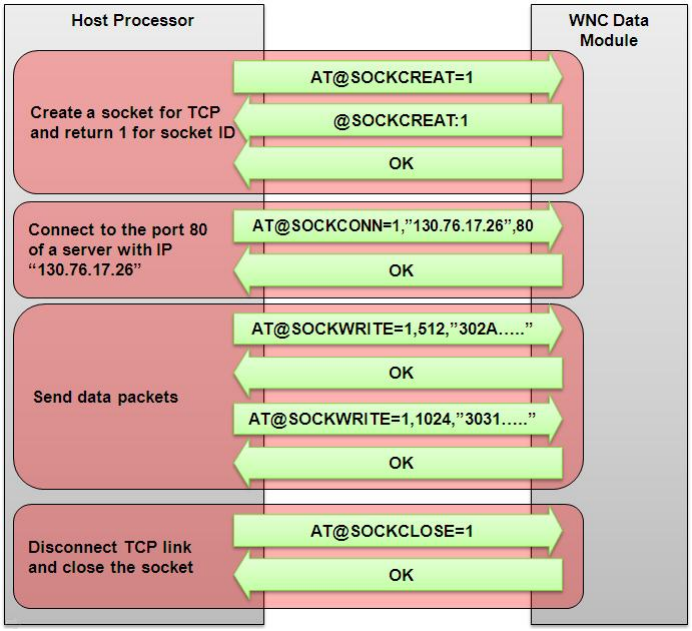
Type II: AT Commands to control a USB CDC-ACM device and data traffic over a USB CDC-ECM interface.

Example:

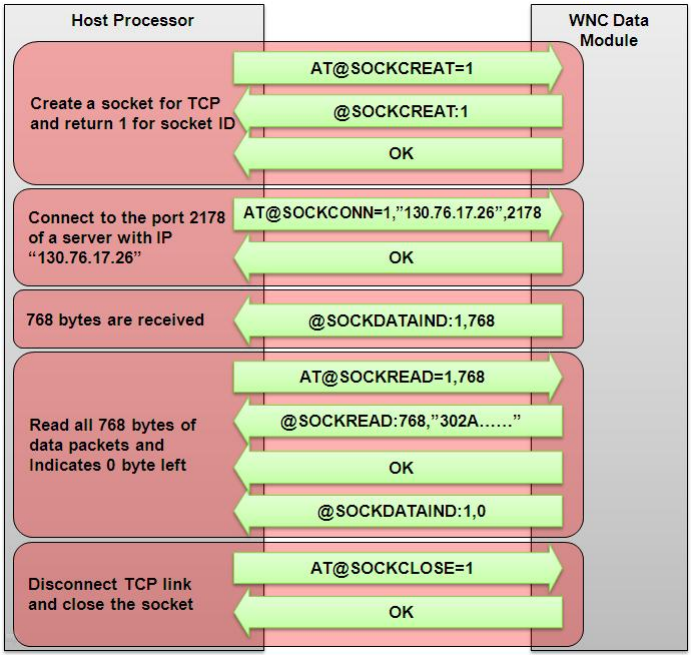


Type III: AT commands for control and data traffic over a USB CDC-ACM device or UART.

Example: Sending TCP packets



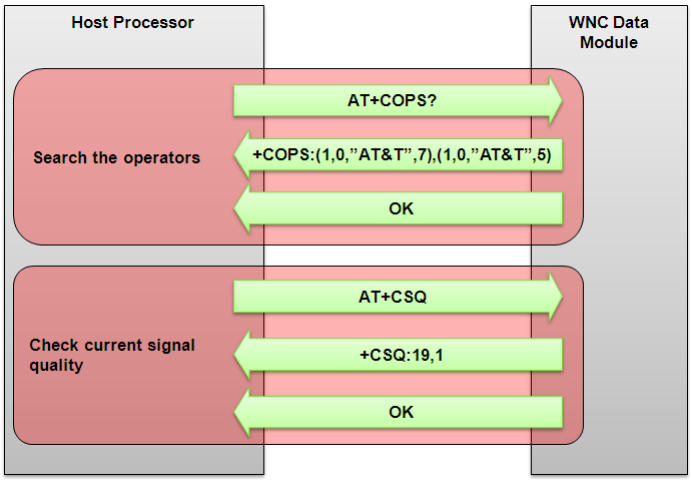
Example: Receiving TCP packets



5.5. Network Status Monitoring

A wireless mobile network contains more interferences than a wired network. The external host processor may need to obtain the current status of the mobile network and address some unexpected occurrence. The WNC data module provides several commands for monitoring the status of the mobile network (such as network signal quality) which operators can view. These commands are the same for the Type II and Type III interface configuration.

Example:



6. FAQ

6.1. List of Abbreviation

Table 1. List of Abbreviation

Abbreviation	Definition
3GPP	3rd Generation Partnership Project
APN	Access Point Name
API	Application program interface
AT	Attention command
CDC-ACM	Communications Device Class-Abstract Control Model
CDC-ECM	Communications Device Class-Ethernet Control Model
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FAQ	Frequently Asked Questions
FOTA	Firmware upgrade Over The Air
FTP	File Transfer Protocol
HTPP	Hypertext Transfer Protocol

ID	Identification
IP	Internet Protocol
JSON	JavaScript Object Notation
LTE	Long Term Evolution
LWM2M	Light-Weight Machine 2 Machine
MAL	Modem Abstraction Layer
M2M	Machine 2 Machine
MSM	Mobile Station Mobile
N/A	Not/Applicable
OS	Operating System
PIN	Personal Identification Number
PUK	Pin Unblocking Code
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
QMI	Qualcomm MSM Interface
RmNet	Rm Network interface
SDK	Software Development Kit
SIM	Subscriber Identity Module
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UIM	User Identity Module
USB	Universal Serial Bus
VID/PID	Vendor ID/ Product ID
WNC	Wistron NeWeb Corporation
WMCCM	WNC Connection Manager

7. References

R1. Please refer to the section “10. Internet Service Commands” in the document “**[WNC M18Q2 M14A2A]AT Command Guide**”.