



Telit Modules Linux USB Drivers

Software User Guide

1VV0301371 Rev.18

2024-09-09

Released

Confidential



Contents

1	Applicability Table.....	3
2	Introduction.....	4
2.1	Scope	4
2.2	Audience	4
2.3	Contact Information, Support	4
2.4	Conventions	4
2.5	Terms and Conditions	5
2.6	Disclaimer	5
3	Operating System Setup.....	6
3.1	USB Compositions	6
3.1.1	PIDs and Related Compositions	6
3.2	Multi-configuration Compositions	9
3.2.1	Kernel Module Option	9
3.2.2	Kernel Module qmi_wwan	10
3.2.2.1	qmi_wwan and Internal QMAP	11
3.2.2.2	qmi_wwan and rmnet	11
4	Using the Modem.....	12
4.1	Using the Serial Ports	12
4.1.1	Data Connection through Serial Ports	12
4.2	Using the Network Adapter	12
4.2.1	Data Connection through the Network Interface	13
4.3	Using the Modem with Modem Manager and Network Manager	13
5	Flashing Devices.....	14
5.1	Flashing Device 0x18d1:0xd00d	14
5.2	Flashing Device 0x058b:0x0041	14
5.3	Flashing Device 0x8087:0x0716	15
5.4	Flashing Device 0x05c6:0x9008	15
6	Telit Kernel Commits.....	16
7	Additional Kernel Commits.....	21
7.1	Raw-IP Support and Important Fixes for qmi_wwan	21
7.2	QMAP Support in qmi_wwan	21
8	Linux Version Finder	22
9	Acronyms and Abbreviations.....	23
10	Related Documents.....	24
11	Document History.....	25



1 Applicability Table

Warning: The kernel version information is related to the default composition of the modem: for detailed information on the kernel version in which a composition is supported please refer to chapter Telit Kernel Commits.

The Products list indicates the generic modem families and does not include variants: please refer to the list of compositions in section PIDs and Related Compositions to check support for a modem.

Table 1: Applicability Table

Products	AVAILABLE SINCE THE KERNEL VERSION
DE910 SERIES	3.4
ELS63 SERIES	5.4
FD980 SERIES	5.14
FE912C04 SERIES	6.10
FN920C04 SERIES	6.9
FE990 SERIES	6.3
FN980 SERIES	5.5
FN990 SERIES	5.16
GE910 SERIES	4.4
HE910 SERIES	4.4
LE866 SERIES	2.6.39
LE910 SERIES	3.18
LE910Cx Linux-based SERIES	4.11
LE910Cx Thread-x SERIES	5.8
LE910D1 SERIES	2.6.39
LE910Q1 SERIES	5.4
LE910R1 SERIES	5.17
LE910S1 SERIES	5.13
LE910 V2 SERIES	3.12
LM940 SERIES	4.10
LM960 SERIES	4.10
LN920 SERIES	5.15
LN940 SERIES	4.20
ME910C1 SERIES	4.15
MEx10G1 SERIES	5.5
ML865C1 SERIES	4.15
ML865G1 SERIES	5.5
UE866 SERIES	4.4
UE910 SERIES	4.4
UL865 SERIES	4.4



2 Introduction

2.1 Scope

This document describes which Linux kernel drivers should be used for the Telit modules listed in the applicability table and how Linux devices can be used for typical use cases.

2.2 Audience

This document is intended for Telit customers, especially system integrators, who are about to implement the Telit modules listed in the applicability table in a Linux environment.

2.3 Contact Information, Support

For technical support and general questions, e-mail:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com
- TS-ONEEDGE@telit.com

Alternatively, use: <https://www.telit.com/contact-us/>

For Product information and technical documents, visit: <https://www.telit.com>

2.4 Conventions

Note: Provide advice and suggestions that may be useful when integrating the module.

Danger: This information MUST be followed, or catastrophic equipment failure or personal injury may occur.

Warning: Alerts the user on important steps about the module integration.

All dates are in ISO 8601 format, that is YYYY-MM-DD.

2.5 Terms and Conditions

Refer to <https://www.telit.com/hardware-terms-conditions/>.

2.6 Disclaimer

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY. TELIT CINTERION RESERVES THE RIGHT TO MAKE CHANGES TO THE PRODUCTS DESCRIBED HEREIN. THE SPECIFICATIONS IN THIS DOCUMENT ARE SUBJECT TO CHANGE AT THE DISCRETION OF TELIT CINTERION WITHOUT PRIOR NOTICE. THIS DOCUMENT IS PROVIDED ON "AS IS" BASIS ONLY AND MAY CONTAIN DEFICIENCIES OR INADEQUACIES. TELIT CINTERION DOES NOT ASSUME ANY LIABILITY FOR INFORMATION PROVIDED IN THE DOCUMENT OR ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT DESCRIBED HEREIN.

TELIT CINTERION GRANTS A NON-EXCLUSIVE RIGHT TO USE THE DOCUMENT. THE RECIPIENT SHALL NOT COPY, MODIFY, DISCLOSE, OR REPRODUCE THE DOCUMENT EXCEPT AS SPECIFICALLY AUTHORIZED BY TELIT CINTERION.

TELIT CINTERION AND THE TELIT CINTERION LOGO, ARE TRADEMARKS OF TELIT CINTERION AND ARE REGISTERED IN CERTAIN COUNTRIES. ALL OTHER REGISTERED TRADEMARKS OR TRADEMARKS MENTIONED IN THIS DOCUMENT ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS AND ARE EXPRESSLY RESERVED BY TELIT CINTERION (AND ITS LICENSORS).



3 Operating System Setup

The Telit modules listed in the Applicability Table expose different kinds of devices according to the Product ID (PID) in use.

The table below lists the association between the device type and the kernel driver used:

Table 2: Device Types and Related Kernel Modules

Device Type	Kernel Module
Serial port following the CDC-ACM standard	cdc_acm (CONFIG_USB_ACM)
Serial port (reduced ACM)	option (CONFIG_USB_SERIAL_OPTION)
Network adapter following the CDC-ECM standard	cdc_ether (CONFIG_USB_NET_CDCETHER)
Network adapter following the CDC-NCM standard	cdc_ncm (CONFIG_USB_NET_CDC_NCM)
Network adapter following Microsoft RNDIS specification	rndis_host (CONFIG_USB_NET_RNDIS_HOST)
Mobile broadband adapter following CDC-MBIM standard	cdc_mbim (CONFIG_USB_NET_CDC_MBIM)
Rmnet mobile broadband adapter	qmi_wwan (CONFIG_USB_NET_QMI_WWAN)
Android Debug Bridge (ADB)	N/A (managed at the userspace level)
Audio device	snd-usb-audio (CONFIG_SND_USB_AUDIO)

To use a specific device type, the related module should be included in the kernel build.

Warning: Certain kernel modules may only be accessible from a particular kernel version onwards for example, `cdc_mbim` is available from 3.8. If the kernel version in use does not support the required driver, consider upgrading the kernel or backporting the needed patches. A simple way to understand if a specific commit (or a set of commit) is available in stable kernels and in which version has been introduced is described in chapter Linux Version Finder

3.1 USB Compositions

3.1.1 PIDs and Related Compositions

The following table lists the currently supported USB compositions in Linux according to the PID:

Note: a composition could have hidden devices. The description in the following table reports all the devices available in the composition, also the hidden ones.

Table 3: PIDs and Related Compositions

PID	Composition
0x0021	6 CDC-ACM devices
0x0022	3 CDC-ACM devices
0x0023	6 CDC-ACM devices + 1 CDC-ECM network adapter
0x0032	6 CDC-ACM devices + 1 MBIM adapter
0x0035	6 CDC-ACM devices
0x0036	6 CDC-ACM devices + 1 CDC-NCM network adapter
0x0100	4 CDC-ACM devices + 1 CDC-NCM network adapter
0x1003	3 reduced ACM devices
0x1004	4 reduced ACM devices
0x1005	4 reduced ACM devices
0x1006	3 reduced ACM devices
0x1010	4 reduced ACM devices
0x1012	3 reduced ACM devices
0x1031	3 reduced ACM devices + 1 rmnet adapter
0x1033	3 reduced ACM devices + 1 ECM network adapter
0x1035	2 reduced ACM devices + 1 ECM network adapter
0x1040	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1041	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1042	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1043	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1045	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB + 1 audio device
0x1050	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1051	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1052	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1053	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1055	5 reduced ACM devices + 1 ADB
0x1056	Configuration #1: Mass storage device Configuration #2: 5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1057	6 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1058	6 reduced ACM devices + 1 ADB
0x1060	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1061	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1062	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1063	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1070	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1071	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1072	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1073	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1075	1 reduced ACM device + 1 ADB
0x1080	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1081	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1082	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1083	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1085	1 reduced ACM device + 1 ADB
0x10a0	3 reduced ACM devices + 1 rmnet adapter
0x10a4	3 reduced ACM devices + 1 rmnet adapter
0x10a9	2 reduced ACM devices + 1 rmnet adapter + 1 ADPL + 1 ADB

0x110b	3 reduced ACM devices + 1 ECM network adapter
0x1200	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1201	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1203	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB
0x1204	5 reduced ACM devices + 1 MBIM adapter + 1 ADB
0x1206	5 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1207	2 reduced ACM devices
0x1208	3 reduced ACM devices + 1 ADB
0x1211	1 reduced ACM device + 1 ECM network adapter + 1 ADB
0x1212	1 reduced ACM device + 1 ADB
0x1213	1 reduced ACM device + 1 ECM network adapter
0x1214	2 reduced ACM devices + 1 ECM network adapter + 1 ADB
0x1230	5 reduced ACM devices + 1 rmnet adapter + 1 ADB + 1 audio device
0x1231	5 reduced ACM devices + 1 RNDIS network adapter + 1 ADB + 1 audio device
0x1250	4 reduced ACM devices + 1 rmnet adapter
0x1260	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1261	5 reduced ACM devices + 1 rmnet adapter + 1 ADB
0x1900	4 reduced ACM devices + 1 rmnet adapter
0x1901	4 reduced ACM devices + 1 MBIM adapter
0x2300	Config. 1: 3 CDC-ACM devices + 1 RNDIS network adapter Config. 2: 3 CDC-ACM devices + 1 ECM network adapter
0x3000	3 reduced ACM devices + 1 rmnet adapter
0x3001	2 reduced ACM devices + 1 rmnet adapter + 1 ADPL + 1 ADB
0x7010	3 reduced ACM devices + 1 RNDIS network adapter
0x7011	3 reduced ACM devices + 1 ECM network adapter
0x701a	3 reduced ACM devices + 1 RNDIS network adapter
0x701b	3 reduced ACM devices + 1 ECM network adapter
0x7020	3 reduced ACM devices + 1 RNDIS network adapter
0x7021	3 reduced ACM devices + 1 ECM network adapter

For additional details on the composition, please refer to the software user guide of the module in use.

The command:

```
$ lsusb
```

can be used to display a list of USB devices that are currently connected to the host.

Warning: Refer to the modem user guide to understand how to change the USB composition and to identify the scope of the exposed devices.



3.2 Multi-configuration Compositions

Certain compositions show multiple configurations (for example, 0x1056). By default, the system will employ the first configuration.

To change the configuration, the desired value must be written to the file:

```
/sys/bus/usb/devices/<device path>/bConfigurationValue
```

For Example,

```
# echo <configuration value> > /sys/bus/usb/devices/<device path>/bConfigurationValue
```

Tool `usb_modeswitch` can also be used to change the configuration, for example:

```
# usb_modeswitch -v 0x1bc7 -p <pid> -u <configuration value>
```

3.2.1 Kernel Module Option

When using a supported composition requiring the option kernel module and the serial ports are not available in `/dev` directory, it is possible that the particular composition has been integrated into a newer kernel version than the one that is currently in use.

The solution is to upgrade the kernel version or backport the needed patches among the ones listed in Telit Kernel Commits.

It is possible to add runtime support for the current composition by issuing the following commands with root privileges:

```
# modprobe option
# echo 1bc7 <PID> > /sys/bus/usb-serial/drivers/option1/new_id
```

where `<PID>` is the PID of the composition to be supported.

Warning: If a network adapter is also available in the composition, make sure it is properly recognized by the kernel before adding runtime support for serial ports. If the ADB device is available in the composition, adding runtime support for serial ports prevents the ADB device from working properly since it is bound to a serial port.

For the most up-to-date list of Telit PIDs supported as an option, refer to the source code in mainline, looking for all the device entries with `TELIT_VENDOR_ID` Vendor ID (VID).

Note: If support for a composition missing in section PIDs and Related Compositions is needed, contact customer support with the request.

3.2.2 Kernel Module qmi_wwan

When using a supported composition requiring the qmi_wwan kernel module and no modem-related network interface is available in the list provided by the command:

```
$ ip link show
```

it is possible that support for that composition has been added in a newer kernel version than the one in use.

The solution is to upgrade the kernel version or backport the needed patches among the ones listed in chapter Telit Kernel Commits

Warning: If the kernel in use has commit cdc-wdm: fix "out-of-sync" due to missing notifications, it should be reverted as done in commit USB: Revert "cdc-wdm: fix "out-of-sync" due to missing notifications".

For the most up-to-date list of Telit PIDs supported as an option, refer to the source code in mainline, looking for all the entries of the device with Vendor ID (VID) 0x1bc7.

Note: If support for a composition missing in section PIDs and Related Compositions is needed, contact customer support with the request.

Warning: When QMAP is not set, the rx URB size in qmi_wwan should be greater than 2048 bytes. This setting can be configured at runtime, changing the MTU of the network interface to an allowed value > 2048 (should not be a multiple of the endpoint max packet size) before setting up the data connection, for example, ip link set <network interface name> mtu 2500. For a permanent setting, the following line: dev->rx_urb_size = 2048; should be added to qmi_wwan.c function qmi_wwan_bind before returning in the successful case.

3.2.2.1 qmi_wwan and Internal QMAP

Since kernel version 4.12, qmi_wwan supports Qualcomm Multiplexing and Aggregation Protocol (QMAP).

QMAP is needed for multiple concurrent PDNs management and to get the most from high-cat modems in terms of throughput.

Kernel side QMAP management is done through qmi_wwan sysfs files add_mux/del_mux: check kernel documentation for further details.

QMAP enablement requires also a special modem configuration executed at the userspace level: the procedure to follow depends on the used tools.

Please check relevant qmi_wwan QMAP-related fixes in section QMAP Support in qmi_wwan to get a stable version.

The master netdevice should have its MTU at least as the maximum downlink aggregated size.

For further details on qmi_wwan internal QMA, refer to “Enabling QMAP in Linux Application Note”.

3.2.2.2 qmi_wwan and rmnet

Starting with kernel version 5.12, the qmi_wwan supports kernel module rmnet for using QMAP (see commit net: qmi_wwan: Add pass-through mode). Kernel side QMAP management is done through qmi_wwan sysfs files pass_through.

For more information, refer to kernel documentations.

QMAP enablement requires also a special modem configuration executed at the userspace level: the procedure to follow depends on the used tools.

The qmi_wwan netdevice should have its MTU at least as the maximum downlink aggregated size.

For more information on rmnet QMAP, refer to “Enabling QMAP in Linux Application Note”.

Warning: the kernel module rmnet should be used instead of the qmi_wwan inbox QMAP.

4 Using the Modem

4.1 Using the Serial Ports

According to the driver in use, the following devices are created for serial ports:

Table 4: Device Names and Related Kernel Modules

Device Type	Kernel Module
/dev/ttyACMx	cdc_acm
/dev/ttyUSBx	option

These are Linux character devices and support most of the features implemented by the tty layer. For example, a terminal emulator like Minicom can be used to send AT commands.

When writing code for using these devices, please refer to the programming language API related to character devices. For example, C-based applications can use the exported functions in the system header files *fcntl.h* and *unistd.h*.

For more information, refer to the related man page.

Warning: When sending AT commands, it is mandatory to have the DTR asserted to get the response.

4.1.1 Data Connection through Serial Ports

To create dial-up connections through serial ports the software `pppd` can be used. For more information and updated source code, refer to the [pppd official website](#).

4.2 Using the Network Adapter

If a network adapter or mobile broadband device is available and the related kernel module is loaded, a network interface is created by the operating system.

Standard Linux commands (for example `ip`, `ifconfig`) can be used to manage the network interface.

For more information, refer to the man page of the command.



4.2.1 Data Connection through the Network Interface

To establish a data connection via the network interface, please follow the instructions listed in the table below, which are specific to the kernel module being used:

Table 5: Kernel Module In Use for the Network Device and Related Data Connection Procedure

Kernel Module	Procedure
qmi_wwan	<p>The libqmi project can be used.</p> <p>For more information, refer to the project documentation.</p> <p>Telit provides also a proprietary QMI SDK, for more information refer to QMI SDK and TQCM User Guide.</p>
cdc_mbim	<p>The libqmi project can be used.</p> <p>For more information, refer to the project documentation.</p>
cdc_ether	<p>AT commands should be used.</p> <p>For more information, refer to the modem documentation.</p>
cdc_ncm	<p>AT commands should be used.</p> <p>For more information, refer to the modem documentation.</p>
rndis_host	<p>AT commands should be used.</p> <p>For more information, refer to the modem documentation.</p>

4.3 Using the Modem with Modem Manager and Network Manager

[ModemManager](#) is a DBus-activated daemon that controls mobile broadband (2G/3G/4G) devices and connections.

ModemManager provides a unified high-level API for communicating with mobile broadband modems, regardless of the protocol used to communicate with the actual device (AT commands, MBIM, QMI).

For managing non-AT-based modems, ModemManager uses external libraries such as freedesktop.org libqmi for QMI-based modems, and libmbim for MBIM-based modems.

ModemManager can be used with *NetworkManager* for easier network connection management.

[NetworkManager](#) is the standard Linux network configuration tool suite. It supports a wide range of networking setups, from desktop to server and mobile, integrating well with popular desktop environments and server configuration management tools.

NetworkManager provides a complete D-Bus API used to access the NetworkManager daemon. This interface can be used to query network status and the details of network interfaces such as current IP addresses or DHCP options. The API can also be used for managing the connections (including creation, activation, and deactivation).

NetworkManager uses *ModemManager* to support mobile broadband devices.



5 Flashing Devices

The modems listed in the following table support firmware updates through special flashing devices that may require binding to a kernel module:

Table 6: Flashing Devices

Product	VID:PID	Kernel Module	Device Name
FD980, FE990, FN980, FN990, LE910C1-EUX, LN920	0x1bc7:0x9010	option	/dev/ttyUSBx
GE/HE/UE910, UE866, UL865	0x058b:0x0041	usb-serial-simple	/dev/ttyUSBx
LE910Cx, LM940, LM960	0x18d1:0xd00d	Managed at the userspace level	n/a
LE910 V2	0x8087:0x0716	usb-serial-simple	/dev/ttyUSBx
LE866, LE910D1	0x216F:0x0051	cdc_acm	/dev/ttyACMx
LE910Q1, ELS63	0x1bc7:0x920A	cdc_acm	/dev/ttyACMx
LE910S1	0x1bc7:0x9200	option	/dev/ttyUSBx
LE910R1	0x1bc7:0x9201	option	/dev/ttyUSBx
FE912C04, FN920C04, ME910G1	0x05c6:0x9008	qcserial	/dev/ttyUSBx

The Flashing devices available in GE/HE/UE910, UE866, UL865, LE910 V2, LE940B6, and LE866 appear for a few seconds when the modem is turned on. However, if the flashing application is not running, the flashing device disconnects, and the modem continues in normal operative mode.

For flashing device support in kernel versions, refer to Flashing Devices.

5.1 Flashing Device 0x18d1:0xd00d

The Flashing device 0x18d1:0xd00d is managed at the userspace level by the Telit firmware update application known as “uxfp”. For more information, refer to uxfp User Guide.

Legacy Telit firmware update application lxfp requires binding the device to the option driver.

This can be permanently done by adding the following line:

```
{ USB_DEVICE(0x18d1, 0xd00d) }
```

to the struct usb_device_id option_ids in the kernel source drivers/usb/serial/option.c

For testing purposes the procedure described in section [Multi-configuration Compositions](#) can be used:

```
$ modprobe option
$ echo 18d1 d00d > /sys/bus/usb-serial/drivers/option1/new_id
```

5.2 Flashing Device 0x058b:0x0041

Even though the flashing device 0x058b:0x0041 presents itself as an ACM device, it should be driven by the kernel driver usb-serial-simple. Support for this device is available since kernel version 4.4.

Previous kernel versions require commits [USB: cdc_acm Ignore Infineon Flash Loader utility](#) and [USB: serial: Another Infineon Flash loader USB ID](#).

5.3 Flashing Device 0x8087:0x0716

Support for flashing device 0x8087:0x0716 is available since kernel version 3.12 with driver usb-serial-simple.

Previous kernel versions require commit USB: serial: move the "simple" drivers into usb-serial-simple.c.

5.4 Flashing Device 0x05c6:0x9008

Support for flashing device 0x05c6:0x9008 is available since kernel version 2.6.30 with driver qcserial.

Previous kernel versions require commit USB: qcserial: Add extra device IDs.

6 Telit Kernel Commits

Below is a list of kernel commits related to the compositions available for the modems listed in the applicability table. Consider backporting if the required PID is unavailable in the used kernel version.

Table 7: Kernel Commits Related to Telit Modules

Summary	VID:PID	Commit	Availability
USB: option driver: adding support for Telit CC864-SINGLE, CC864-DUAL and DE910-DUAL modems	0x1bc7:0x1005	7204cf584836c24b4b06e4ad4a8e6bb8ea84908e	v3.4-rc1
	0x1bc7:0x1006		
	0x1bc7:0x1010		
usb: option driver, add support for Telit UE910v2	0x1bc7:0x1012	d6de486bc22255779bd54b0fceb4c240962bf146	v3.15-rc2
USB: option: add support for Telit LE920	0x1bc7:0x1200	03eb466f276ceef9dcf023dc5474db02af68aad9	v3.8-rc7
NET: qmi_wwan: add Telit LE920 support	0x1bc7:0x1200	3d6d7ab5881b1d4431529410b949ba2e946f3b0f	v3.8-rc7
net: qmi_wwan: add Telit LE920 newer firmware support	0x1bc7:0x1201	905468fa4d54c3e572ed3045cd47cce37780716e	v3.13-rc1
usb: option: add support for Telit LE910	0x1bc7:0x1201	2d0eb862dd477c3c4f32b201254ca0b40e6f465c	v3.18-rc3
USB: cdc_acm: Ignore Infineon Flash Loader utility	0x058b:0x0041	f33a7f72e5fc033daccbb8d4753d7c5c41a4d67b	v4.4-rc5
USB: serial: Another Infineon flash loader USB ID	0x058b:0x0041	a0e80fbd56b4573de997c9a088a33abbc1121400	v4.4-rc5
USB: serial: option: Adding support for Telit LE922	0x1bc7:0x1042	ff4e2494dc17b173468e1713fdf6237fd8578bc7	v4.5-rc2
	0x1bc7:0x1043		
USB: serial: option: add support for Telit LE922 PID 0x1045	0x1bc7:0x1045	5deef5551c77e488922cc4bf4bc76df63be650d0	v4.5-rc7
net: usb: cdc_ncm: adding Telit LE910 V2 mobile broadband card	0x1bc7:0x0036	79f4223257bfe52b0a26d0d7ad4019e764be6ce	v4.6-rc2
USB: serial: option: add support for Telit LE910 PID 0x1206	0x1bc7:0x1206	3c0415fa08548e3bc63ef741762664497ab187ed	v4.8-rc1
USB: serial: option: add support for Telit LE920A4	0x1bc7:0x1207	01d7956b58e644ea0d2e8d9340c5727a8fc39d70	v4.8-rc3
	0x1bc7:0x1208		
	0x1bc7:0x1211		
	0x1bc7:0x1212		

	0x1bc7:0x1213		
	0x1bc7:0x1214		
NET: usb: qmi_wwan: add support for Telit LE922A PID 0x1040	0x1bc7:0x1040	9bd813da24cd49d749911d7fdc0e9ae9a673d746	v4.9-rc8
NET: usb: cdc_mbim: add quirk for supporting Telit LE922A	0x1bc7:0x1041	7b8076ce8a00d553ae9d3b7eb5f0cc3e63cb16f1	v4.9
USB: serial: option: add support for Telit LE922A PIDs 0x1040, 0x1041	0x1bc7:0x1040 0x1bc7:0x1041	5b09eff0c379002527ad72ea5ea38f25da8a8650	v4.10-rc1
drivers: net: usb: qmi_wwan: add QMI QUIRK_SET_DTR for Telit PID 0x1201	0x1bc7:0x1201	14cf4a771b3098e431d2677e3533bdd962e478d8	v4.11-rc7
net: usb: qmi_wwan: add Telit ME910 support	0x1bc7:0x1100	4c54dc0277d0d55a9248c43aebd31858f926a056	v4.12-rc1
usb: serial: option: add Telit ME910 support	0x1bc7:0x1100	40dd46048c155b8f0683f468c950a1c107f77a7c	v4.12-rc1
net: usb: qmi_wwan: add Telit ME910 PID 0x1101 support	0x1bc7:0x1101	c647c0d62c82eb3ddf78a0d8b3d58819d9f552aa	v4.15-rc4
USB: serial: option: add support for Telit ME910 PID 0x1101	0x1bc7:0x1101	08933099e6404f588f81c2050bfec7313e06eeaf	v4.15-rc6
net: usb: cdc_mbim: add flag FLAG_SEND_ZLP	0x1bc7:0x1041	9f7c728332e8966084242fcd951aa46583bc308c	v4.17
USB: serial: option: add Telit LN940 series	0x1bc7:0x1900 0x1bc7:0x1901	28a86092b1753b802ef7e3de8a4c4a69a9c1bb03	v4.20
qmi_wwan: Added support for Telit LN940 series	0x1bc7:0x1900	1986af16e8ed355822600c24b3d2f0be46b573df	v4.20
usb: cdc-acm: send ZLP for Telit 3G Intel based modems	0x1bc7:0x0021 0x1bc7:0x0023	34aabf918717dd14e05051896aaecd3b16b53d95	v5.0-rc2
USB: serial: option: add Telit ME910 ECM composition	0x1bc7:0x1102	6431866b6707d27151be381252d6eef13025cfce	v5.1-rc1
net: usb: qmi_wwan: add Telit 0x1260 and 0x1261 compositions	0x1bc7:0x1260 0x1bc7:0x1261	b4e467c82f8c12af78b6f6fa5730cb7dea7af1b4	v5.2-rc2
USB: serial: option: add Telit 0x1260 and 0x1261 compositions	0x1bc7:0x1260 0x1bc7:0x1261	f3dfd4072c3ee6e287f501a18b5718b185d6a940	v5.2-rc5
USB: serial: option: add Telit FN980 compositions	0x1bc7:0x1050 0x1bc7:0x1051 0x1bc7:0x1052	5eb3f4b87a0e7e949c976f32f296176a06d1a93b	v5.4-rc3

	0x1bc7:0x1053		
net: usb: qmi_wwan: add Telit 0x1050 composition	0x1bc7:0x1050	e0ae2c578d3909e60e9448207f5d83f785f129f	v5.4-rc4
USB: serial: option: add Telit ME910G1 0x110a composition	0x1bc7:0x110a	0d3010fa442429f8780976758719af05592ff19f	v5.5-rc6
USB: serial: option: add ZLP support for 0x1bc7/0x9010	0x1bc7:0x9010	2438c3a19dec5e98905fd3ffcc2f24716aceda6b	v5.5-rc6
USB: serial: option: add ME910G1 ECM composition 0x110b	0x1bc7:0x110b	8e852a7953be2a6ee371449f7257fe15ace6a1fc	v5.6-rc7
net: usb: qmi_wwan: add Telit LE910C1-EUX composition	0x1bc7:0x1031	591612aa578cd7148b7b9d74869ef40118978389	v5.7
USB: serial: option: add Telit LE910C1-EUX compositions	0x1bc7:0x1031 0x1bc7:0x1033	399ad9477c523f721f8e51d4f824bdf7267f120c	v5.8-rc1
USB: serial: option: add LE910Cx compositions 0x1203, 0x1230, 0x1231	0x1bc7:0x1203 0x1bc7:0x1230 0x1bc7:0x1231	489979b4aab490b6b917c11dc02d81b4b742784a	v5.10-rc3
net: usb: qmi_wwan: add Telit LE910Cx 0x1230 composition	0x1bc7:0x1230	5fd8477ed8ca77e64b93d44a6dae4aa70c191396	v5.10-rc3
USB: serial: option: add Telit FN980 composition 0x1055	0x1bc7:0x1055	db0362eeb22992502764e825c79b922d7467e0eb	v5.10-rc3
usb: serial: option: add Telit LE910-S1 compositions 0x7010, 0x7011	0x1bc7:0x7010 0x1bc7:0x7011	e467714f822b5d167a7fb03d34af91b5b6af1827	v5.13-rc4
usb: serial: option: add Telit FD980 composition 0x1056	0x1bc7:0x1056	5648c073c33d33a0a19d0cb1194a4eb88efe2b71	v5.14-rc5
net: usb: cdc_mbim: avoid altsetting toggling for Telit LN920	0x1bc7:0x1061	aabbdc67f3485b5db27ab4eba01e5fbf1ffea62c	v5.15-rc1
net: usb: qmi_wwan: add Telit 0x1060 composition	0x1bc7:0x1060	8d17a33b076d24aa4861f336a125c888fb918605	v5.15-rc1
USB: serial: option: add Telit LN920 compositions	0x1bc7:0x1060 0x1bc7:0x1061 0x1bc7:0x1062 0x1bc7:0x1063	7bb057134d609b9c038a00b6876cf0d37d0118ce	v5.15-rc3
USB: serial: option: add Telit LE910Cx composition 0x1204	0x1bc7:0x1204	f5a8a07edafed8bede17a95ef8940fe3a57a77d5	v5.15-rc6
USB: serial: option: add Telit LE910S1 0x9200 composition	0x1bc7:0x9200	e353f3e88720300c3d72f49a4bea54f42db1fa5e	v5.16-rc3

USB: serial: option: add Telit FN990 compositions	0x1bc7:0x1070 0x1bc7:0x1071 0x1bc7:0x1072 0x1bc7:0x1073	2b503c8598d1b232e7fc7526bce9326d92331541	v5.16-rc6
net: usb: qmi_wwan: add Telit 0x1070 composition	0x1bc7:0x1070	94f2a444f28a649926c410eb9a38afb13a83ebe0	v5.16-rc6
net: usb: cdc_mbim: avoid altsetting toggling for Telit FN990	0x1bc7:0x1071	21e8a96377e6b6debae42164605bf9dcbe5720c5	v5.17-rc5
USB: serial: option: add Telit LE910R1 compositions	0x1bc7:0x701a 0x1bc7:0x701b 0x1bc7:0x9201	cfc4442c642d568014474b6718ccf65dc7ca6099	v5.17-rc6
USB: serial: option: add Telit 0x1057, 0x1058, 0x1075 compositions	0x1bc7:0x1057 0x1bc7:0x1058 0x1bc7:0x1075	f32c5a0423400e01f4d7c607949fa3a1f006e8fa	v5.18-rc5
net: usb: qmi_wwan: add Telit 0x1057 composition	0x1bc7:0x1057	f01598090048f5f732ea7aa64b2f194131ce60d2	v5.19-rc1
net: usb: qmi_wwan: add Telit 0x1250 composition	0x1bc7:0x1250	2c262b21de6dc93ac4d8c7a4cea0da4226b451fb	v5.19-rc1
USB: serial: option: add Telit LE910Cx 0x1250 composition	0x1bc7:0x1250	342fc0c3b345525da21112bd0478a0dc741598ea	v5.19-rc4
USB: serial: option: add Telit FE990 compositions	0x1bc7:0x1080 0x1bc7:0x1081 0x1bc7:0x1082 0x1bc7:0x1083	773e8e7d07b753474b2ccd605ff092faaa9e65b9	V6.3-rc6
net: usb: qmi_wwan: add Telit 0x1080 composition	0x1bc7:0x1081	382e363d5bed0cec5807b35761d14e55955eee63	V6.3-rc2
net: usb: cdc_mbim: avoid altsetting toggling for Telit FE990	0x1bc7:0x1081	418383e6ed6b4624a54ec05c535f13d184fbf33b	V6.3-rc2
USB: serial: option: add Telit LE910C4-WWX 0x1035 composition	0x1bc7:0x1035	6a7be48e9bd18d309ba25c223a27790ad1bf0fa3	v6.6-rc7
USB: serial: option: add Telit FN920C04 rmnet compositions	0x1bc7:0x10a0 0x1bc7:0x10a4 0x1bc7:0x10a9	582ee2f9d268d302595db3e36b985e5cbb93284d	V6.9-rc5

net: usb: qmi_wwan: add Telit FN920C04 compositions	0x1bc7:0x10a0 0x1bc7:0x10a4 0x1bc7:0x10a9	0b8fe5bd73249dc20be2e88a12041f8920797b59	v6.9-rc6
net: usb: qmi_wwan: add Telit FN912 compositions	0x1bc7:0x3000 0x1bc7:0x3001	77453e2b015b5ced5b3f45364dd5a72dfc3bdecb	v6.10-rc6
USB: serial: option: add Telit FN912 rmnet compositions	0x1bc7:0x3000 0x1bc7:0x3001	9a590ff283421b71560deded2110dbdcbe1f7d1d	v6.10
USB: serial: option: add Telit generic core-dump composition	0x1bc7:0x9000	4298e400dbdbf259549d69c349e060652ad53611	v6.10



7 Additional Kernel Commits

7.1 Raw-IP Support and Important Fixes for qmi_wwan

Below is a list of commits for adding Raw-Ip support to qmi_wwan and addressing important issues.

Table 8: qmi_wwan Relevant Kernel Commits

Summary	Commit	Availability
net: qmi_wwan: MDM9x30 specific power management	93725149794d3d418cf1eddcac60c7b536c5faa1	v4.5-rc1
usbnet: allow mini-drivers to consume L2 headers	81e0ce79f2919dbd5f025894d29aa806af8695c7	v4.5-rc1
net: qmi_wwan: support "raw IP" mode	32f7adf633b9f99ad5089901bc7ebff57704aaa9	v4.5-rc1
net: qmi_wwan: should hold RTNL while changing netdev type	6c730080e663b1d629f8aa89348291fbc4c46cd9	v4.5-rc1
net: qmi_wwan: ignore bogus CDC Union descriptors	34a55d5e858e81a20d33fd9490149d6a1058be0c	v4.5-rc1
qmi_wwan: Add missing skb_reset_mac_header-call	0de0add10e587effa880c741c9413c874f16be91	v4.14
usbnet: fix alignment for frames with no ethernet header	a4abd7a80addb4a9547f7dfc7812566b60ec505c	v4.15-rc3
qmi_wwan: set FLAG_SEND_ZLP to avoid network initiated disconnect	245d21190aec547c0de64f70c0e6de871c185a24	v4.16-rc1
qmi_wwan: Fix out-of-bounds read	904d88d743b0c94092c5117955eab695df8109e8	v5.2-rc7

7.2 QMAP Support in qmi_wwan

Below is a list of commits for adding QMAP support to qmi_wwan.

Table 9: qmi_wwan QMAP Relevant Kernel Commits

Summary	Commit	Availability
net: usb: qmi_wwan: add qmap mux protocol support	c6adf77953bcec0ad63d7782479452464e50f7a3	v4.12-rc1
qmi_wwan: fix NULL deref on disconnect	bbae08e592706dc32e5c7c97827b13c1c178668b	v4.13-rc5
qmi_wwan: Fix qmap header retrieval in qmimux_rx_fixup	d667044f49513d55fcfefe4fa8f8d96091782901	v4.20
qmi_wwan: add MTU default to qmap network interface	f87118d5760f00af7228033f8e783c7f380d2866	v5.0-rc3
qmi_wwan: add support for QMAP padding in the RX path	61356088ace1866a847a727d4d40da7bf00b67fc	v5.2-rc6
qmi_wwan: add network device usage statistics for qmimux devices	44f82312fe9113bab6642f4d0eab6b1b7902b6e1	v5.2-rc6
qmi_wwan: avoid RCU stalls on device disconnect when in QMAP mode	a8fdde1cb830e560208af42b6c10750137f53eb3	v5.2-rc6
qmi_wwan: extend permitted QMAP mux_id value range	36815b416fa48766ac5a98e4b2dc3ebc5887222e	v5.2-rc6
qmi_wwan: Increase headroom for QMAP SKBs	2e4233870557ac12387f885756b70fc181cb3806	v5.12
net: usb: qmi_wwan: add qmap id sysfs file for qmimux interfaces	e594ad980ec26fb7351d02c84abaa77ecdb4e522	v5.12-rc1-dontuse
net: usb: qmi_wwan: allow qmimux add/del with master up	6c59cff38e66584ae3ac6c2f0cbd8d039c710ba7	v5.12-rc3

8 Linux Version Finder

Linux Version Finder is a Python tool for finding Linux kernel versions in which a commit (or a set of commits) is available, searching the Linux kernel stable tree. The tool can be used in conjunction with the commits listed in [Flashing Devices](#) to identify the stable versions in which a composition is supported.

Linux Version Finder is available at Telit Github Organization: check the [README.md](#) file for instructions about usage.

9 Acronyms and Abbreviations

Table 10: Acronyms and Abbreviations

Acronym	Definition
ACM	Abstract Control Model
ADB	Android Debug Bridge
CDC	Communications Class Device
ECM	Ethernet Control Model
MBIM	Mobile Broadband Interface Model
NCM	Network Control Model
PPP	Point-to-Point Protocol
QMAP	Qualcomm Multiplexing and Aggregation Protocol
USB	Universal Serial Bus



10 Related Documents

Refer to <https://dz.telit.com/> for current documentation and downloads.

Table 11: Acronyms and Abbreviations

Doc Code	Document Title
1W0301643	Telit QMI SDK and TQCM User Guide
1W0301613	uxfp Software User Guide
80455NT12007A	Enabling QMAP in Linux Application Note
N/A	AT Commands Reference Guide of Telit modules is listed in the applicability table.

11 Document History

Table 12: Document History

Revision	Date	Changes
18	2024-07-15	<p>Added FE912C04 to the applicability table.</p> <p>Added FE912C04 compositions 0x3000, 0x3001.</p> <p>Added FN920C04 to the applicability table.</p> <p>Added FN920C04 compositions 0x10a0, 0x10a4, 0x10a9.</p> <p>Added LE910Cx Thread-x-based composition 0x1035.</p> <p>Added LE910Q1 and ELS63 to the applicability table.</p> <p>Added LE910Q1/ELS63 compositions 0x7020, 0x7021.</p> <p>Added LE910Q1/ELS63 PID 0x920A to firmware update compositions.</p> <p>Added generic core-dump patch to patches list.</p>
17	2023-05-15	<p>Added FE990 compositions 0x1080, 0x1081, 0x1082, 0x1083</p> <p>Added config symbols in section 2.1</p> <p>Added reference to QMAP document 80455NT12007A</p> <p>Update template</p>
16	2022-09-12	<p>Added LE910Cx Linux-based composition 0x1250</p> <p>Added Linux Version Finder chapter</p> <p>Updated Table 6 (flashing devices) with ME910G1 device and added related paragraph</p>
15	2022-05-27	<p>Added FN980 compositions 0x1057, 0x1058</p> <p>Added FN990 composition 0x1075</p> <p>Added rmnet kernel module paragraph</p> <p>Modified applicability table for LE910Cx families</p> <p>Added applicability table disclaimer</p>
14	2022-03-02	<p>Added LE910R1 in applicability table</p> <p>Added LE910R1 compositions 0x701A, 0x701B</p> <p>Added LE910R1 composition 0x9201 and related entry in Table 6</p>
13	2021-12-13	<p>Added LE910C1 composition 0x1204</p> <p>Added LE910S1 composition 0x9200 and related entry in Table 6</p>

		Added FN990 compositions 0x1070, 0x1071, 0x1072, 0x1073
12	2021-09-24	Added LN920 in applicability table and related kernel commits
11	2021-08-09	Added FD980 composition 0x1056 Added FD980 in applicability table Added paragraph "Multi-configuration compositions"
10	2021-06-14	Changed document template and reworded some paragraphs Fixed FN980 kernel version availability in the applicability table Added LE910S1 to applicability table Added LE910S1 0x7010 and 0x7011 composition description and related kernel commits Added QMAP paragraph and related kernel commits Modified flashing device information for 0x18d1:0xd00d and 0x8087:0x0801 (removed) Changed kernel commit references from GitHub to git.kernel.org
9	2020-11-09	Added LE910Cx compositions 0x1203, 0x1230, 0x1231 and FN980 composition 0x1055 Removed references to deprecated Modem Manager and Network Manager documents Removed references to deprecated lxfp
8	2020-09-01	Added LE910C1-EUX support and updated kernel patches list Added LM960 0x1040 qmi_wwan rx urb size note
7	2020-03-27	Changed ME910G1 to MEx10G1 in applicability table Added ML865C1 and ML865G1 to applicability table Updated kernel patches list for composition 0x110b
6	2020-01-13	Added ME910G1 0x110a composition Added FN980 0x9010 flashing device composition Updated kernel patches list Updated applicability table
5	2019-10-21	Added FN980 in applicability table and related kernel commits
4	2019-05-24	Removed automotive modules from applicability table Added LN940 and UE866 in applicability table Added LM940 kernel commit for fixing big data packets issue Added ME910 composition 0x1102, LECx910 compositions 0x1260 and 0x1261

		Updated kernel patches list
3	2018-05-07	Added LE866 flashing device details Added kernel commit for PID 0x0036 Added LE910D1 in applicability table
2	2018-02-13	Added LM960 in applicability table Added ME910 composition 0x1101 Added "Additional Kernel Commits" chapter Added "Minimum Kernel Version" in applicability table
1	2017-11-24	Added LE920A4 and LE910C1 composition 0x1201 Added LM940 in applicability table Added reference to commit cdc-wdm: fix "out-of-sync" due to missing notifications
0	2017-04-28	First issue

From Mod.0817 Rev.10



© Telit Cinterion. All rights reserved.

