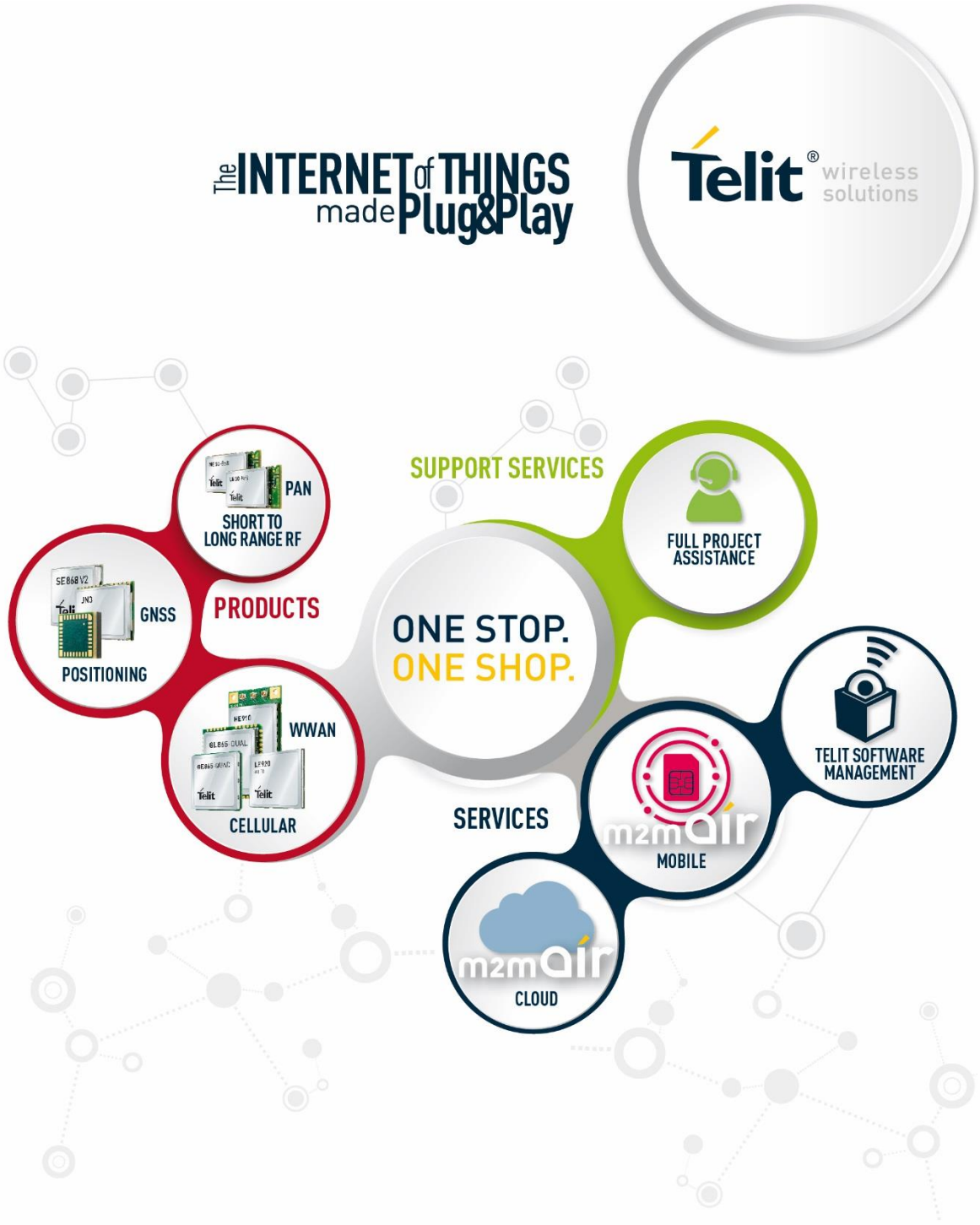



# LE910 V2 SERIES PORTS ARRANGEMENTS

# APPLICABILITY TABLE

## PRODUCTS

- LE910 V2 SERIES

# SPECIFICATION SUBJECT TO CHANGE WITHOUT NOTICE

## LEGAL NOTICE

These Specifications are general guidelines pertaining to product selection and application and may not be appropriate for your particular project. Telit (which hereinafter shall include, its agents, licensors and affiliated companies) makes no representation as to the particular products identified in this document and makes no endorsement of any product. Telit disclaims any warranties, expressed or implied, relating to these specifications, including without limitation, warranties or merchantability, fitness for a particular purpose or satisfactory quality. Without limitation, Telit reserves the right to make changes to any products described herein and to remove any product, without notice.
It is possible that this document may contain references to, or information about Telit products, services and programs, that are not available in your region. Such references or information must not be construed to mean that Telit intends to make available such products, services and programs in your area.

## USE AND INTELLECTUAL PROPERTY RIGHTS

These Specifications (and the products and services contained herein) are proprietary to Telit and its licensors and constitute the intellectual property of Telit (and its licensors). All title and intellectual property rights in and to the Specifications (and the products and services contained herein) is owned exclusively by Telit and its licensors. Other than as expressly set forth herein, no license or other rights in or to the Specifications and intellectual property rights related thereto are granted to you. Nothing in these Specifications shall, or shall be deemed to, convey license or any other right under Telit's patents, copyright, mask work or other intellectual property rights or the rights of others.

You may not, without the express written permission of Telit: (i) copy, reproduce, create derivative works of, reverse engineer, disassemble, decompile, distribute, merge or modify in any manner these Specifications or the products and components described herein; (ii) separate any component part of the products described herein, or separately use any component part thereof on any equipment, machinery, hardware or system; (iii) remove or destroy any proprietary marking or legends placed upon or contained within the products or their components or these Specifications; (iv) develop methods to enable unauthorized parties to use the products or their components; and (v) attempt to reconstruct or discover any source code, underlying ideas, algorithms, file formats or programming or interoperability interfaces of the products or their components by any means whatsoever. No part of these Specifications or any products or components described herein may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without the prior express written permission of Telit.

## HIGH RISK MATERIALS

Components, units, or third-party products contained or used with the products described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (“High Risk Activities"). Telit, its licensors and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

## TRADEMARKS

You may not and may not allow others to use Telit or its third party licensors’ trademarks. To the extent that any portion of the products, components and any accompanying documents contain proprietary and confidential notices or legends, you will not remove such notices or legends.

## THIRD PARTY RIGHTS

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

# CONTENTS

# FIGURES

# TABLES

# 1 INTRODUCTION

## 1.1 Scope

The present document provides a guideline to connect logically the physical serial ports of the module to the services supported by the module itself (GNSS, AppZone, etc.). It is up to the user to configure the module in suitable way to avoid hardware/software resources conflicts. With the generic "ports/services arrangement" expression is intended each possible set of logical connections regarding physical ports and Service Access Points supported by the used module (e.g. AT0, AT1, AT2, etc.).

## 1.2 Audience

This guide is addressed to the application developers that need to exploit at best the logical connections regarding the physical serial ports and services provided by the used module, without run up against resources contentions among the involved services.

## 1.3 Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:
TS-EMEA@telit.com
TS-AMERICAS@telit.com
TS-APAC@telit.com

Alternatively, use:
http://www.telit.com/support

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:
http://www.telit.com

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.
Telit appreciates feedback from the users of our information.

## 1.4 Text Conventions


Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

## 1.5 Related Documents

[1] Telit's CMUX Implementation User Guide, 1vv0300994
[2] Telit 4G Modules AT Commands Reference Guide...
[3] LE910 V2 Hardware User Guide, 1vv030...
[4] NCM Protocol User Guide, 1vv0301246
[5]
[6]
[7] Telit's AppZone User Guide, 1vv0301082
[8] AppZone APIs User Guide, 1vv0301130
[9]
[10]
[11]

# 2 VIRTUAL SERVICE DEVICE

Before describing the several ports/services arrangements set up by means of the AT#PORTCFG command, it is useful introduce the Virtual Serial Device software layer.

Virtual Serial Device, called VSD, is a switching layer designed to run on Telit's modules. It manages virtual connections between the physical serial ports, accessible to the user, and the services provided by the module. VSD supports several Service Access Points used as anchorage points for the logical connections. The table below shows the physical and logical objects involved in the connections arrangement: Physical Serial Ports, Services Access Points, Services, and CMUX channels.

| Physical Serial Ports | Service Access Points | Services | CMUX Channels |
|---|---|---|---|
| USIF0<br>USIF1<br>USBx (USB0÷USB6)[1] | AT0, AT1, AT2<br>TT | External GNSS Receiver<br>AppZone | CMUX (VC1÷VC4)[2] |

Tab. 1: Items Connected by means of the VSD

**NOTICE**: in document [3], USIF0 and USIF1 are called respectively Modem Serial Port1 and Modem Serial Port 2.

Referring to Fig. 1:
the modules provide three AT Commands Parser Instances, which are logically independent and connected to three different Service Access Points; each parser recognizes and executes the AT commands received on its Service Access Point.

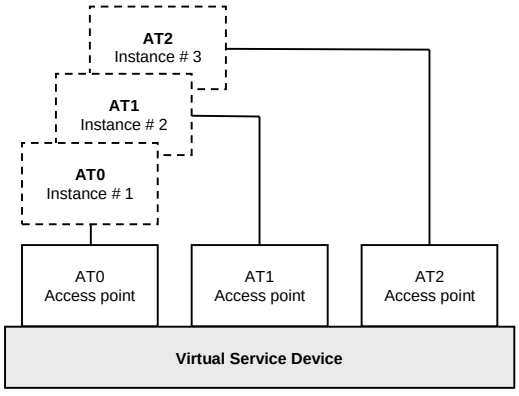



Fig. 1: AT Parser Instances

[1] Seven USB ports (or channels): USB0÷USB6.
[2] Four CMUX channels: VC1÷VC4, see document [1].

# 3 PRELIMINARY INFORMATION

It is strongly recommended to use the AT#USBCFG=? and AT#PORTSCFG=? Test commands to have information respectively on the USBx ports configuration modes and ports arrangements provided by the module that you are using. Refer to document [2] to have information on AT commands syntax.

## 3.1 USB Ports Configuration Modes

Assume that the module is using the factory-setting configuration, and the USB cable is connected to the PC. Fig. 2 shows the USBx ports←→virtual COMx ports mapping; the mapping depends on the Windows-PC configuration. In the current Windows-PC configuration, USB0 port is mapped into COM17 port as shown in the figure below. Telit provides the USB driver to install on the PC (DTE), see chapter 10.3.

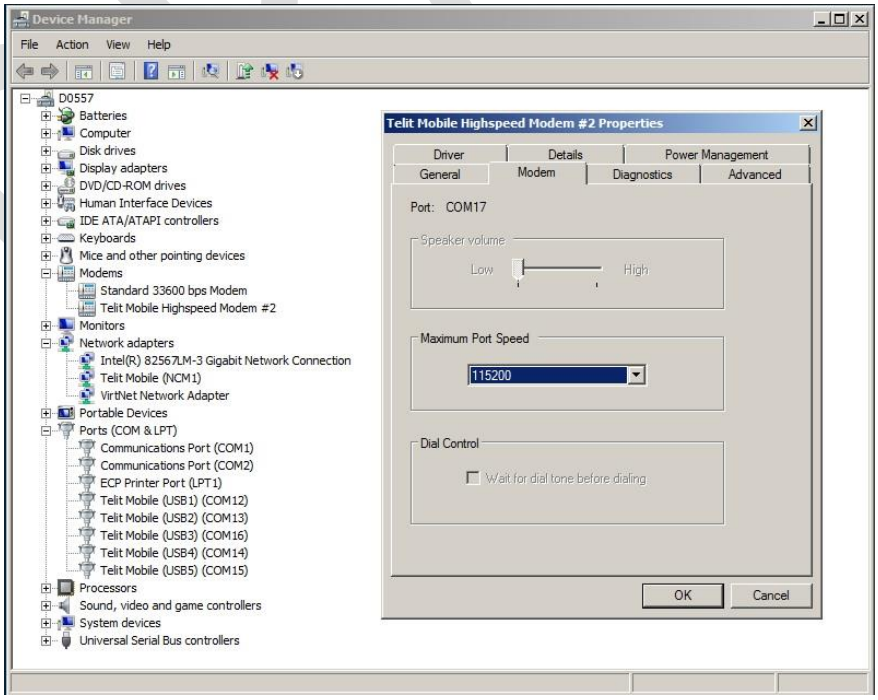



Fig. 2: USBx Ports Mapped into Virtual COMx Ports

Tab. 2 summarizes the mapping shown in Fig. 2.

| USBx Ports | COMx Ports |
|---|---|
| USB0 | COM17 |
| USB1 | COM12 |
| USB2 | COM13 |
| USB3 | COM16 |
| USB4 | COM14 |
| USB5 | COM15 |

Tab. 2: Mapping Table

Enter the AT#USBCFG=? Test command to check the number of USB ports configuration modes provided by the module.

AT#USBCFG=?
#USBCFG: (0-5) ← the module provides six modes
OK

Check the current mode
AT#USBCFG?
#USBCFG: 0 ← 0 is the factory-setting mode
OK

Tab. 3 describes the USB port configuration for each mode; see also chapter 10.1.

| Mode | ACM | SS | NCM | MBIM | DLINK | Description | PID |
|---|---|---|---|---|---|---|---|
| 0 (default) | ✓ | standard | ✓ | | | All available USBx ports implement the ACM subclass, Selective Suspend standard type is supported, NCM protocol is enabled. | 0X36 |
| 1 | Data only | standard | | | In progress | All available USBx ports implement the ACM subclass in Data only mode, Selective Suspend standard type is supported, DLink feature is in progress. | 0x34 |
| 2 | ✓ | standard | | | In progress | All available USBx ports implement the ACM subclass, Selective Suspend standard type is supported, DLink feature is in progress. | 0x35 |
| 3 | ✓ | standard | (✓) | ✓ | | All available USBx ports implement the ACM subclass, Selective Suspend standard type is supported. The USB device starts with the MBIM protocol, the host may force the device to use the NCM protocol. | 0x32 |
| 4 | ✓ | custom | ✓ | | | All available USBx ports implement the ACM subclass, Selective Suspend custom type is supported, NCM protocol is enabled. | 0x37 |
| 5 | ✓ | custom | (✓) | ✓ | | All available USBx ports implement the ACM subclass, Selective Suspend custom type is supported. The USB device starts with the MBIM protocol, the host may force the device to use the NCM protocol. | 0x33 |

Tab. 3: #USBCFG Mode

Here is a brief description of the features (ACM, SS, NCM, MBIM, DLINK) provided by the different USB ports configuration modes.

The USB host (PC) reads the current USB device configuration mode using the Device Control Interface (EP0, see Tab. 4, and chapter 10.2).

| Endpoint | Direction | Type | Description | USB Interfaces |
|---|---|---|---|---|
| EP0 | IN/OUT | Control | Standard request, class request | Interface used for controlling the device. |

Tab. 4: Endpoints Configuration for Device Control Interface

- ACM (Abstract Control Model)

The USB device (module) implements the Abstract Control Model (ACM) subclass defined in the USB Communication Device Class (CDC) specification. This protocol requires two USB interfaces (see chapter 10.2), a data interface that consists of two endpoints (bulk in, and bulk out) and a control interface that consists of one interrupt in endpoint; see Tab. 5.

With some operating systems (for example Linux), the device can work with in-box drivers. It means that no Telit's USB driver is needed when Linux is used.

| Endpoint | Direction | Type | Description | USB Interfaces |
|---|---|---|---|---|
| EP1 | IN | Interrupt | State notification from device to host | Control interface |
| EP2 | IN | Bulk | Data transfer from device to host | Data Interface |
| EP3 | OUT | Bulk | Data transfer from host to device | |

Tab. 5: Endpoints Configuration for ACM

- ACM/Data only

In addition to the standard CDC/ACM compliant protocol, the USB device (module) implements the simplified ACM/DATA only protocol (Data interface: bulk in, and bulk out endpoint); see chapter 10.2.

| Endpoint | Direction | Type | Description | USB Interfaces |
|---|---|---|---|---|
| EP2 | IN | Bulk | Data transfer from device to host | Data Interface |
| EP3 | OUT | Bulk | Data transfer from host to device | |

Tab. 6: Endpoints Configuration for ACM/Data only

- SS

Two Selective Suspend types are implemented:

- Custom: the user application sends messages to power off/on the device. The mechanism to use depends on the operating system and the type of device: composite or non-composite.

- Standard: if the host does not send anything to the device for a certain time interval, the device is automatically powered down. The host can resumes the device, or the device can signal a "remote wake up".

- NCM (Network Control Model)

NCM is a protocol, which targets Ethernet capable devices in the USB layer. USB host and device (module) can efficiently exchange Ethernet frames by this protocol. Regarding packet format, NCM will use 802.3 (i.e. IP packets with Ethernet headers).
Refer to document [4] to get information on how activate the NCM protocol.

- MBIM (Mobile Broadband Interface Model)

The MBIM is a protocol specifically designed for high-speed mobile broadband modem devices. It provides multiple IP connections over a single USB interface, without the need of 802.3 frames (as was the case with ECM and NCM). MBIM uses raw IP packets (i.e. just IP packets without Ethernet headers). In addition, it provides a control protocol to talk to modem devices

- DLink

It is in progress.

Tab. 7 shows the #USBCFG modes supported by Windows and Ubuntu Operating Systems, see chapter 10.1.

| | Windows | Ubuntu |
|---|---|---|
| Mode | Telit Driver | in-box Driver |
| 0 | ✓ | ✓ |
| 1 | | |
| 2 | ✓ | ✓ |
| 3 | ✓ | ✓ |
| 4 | ✓ | ✓ |
| 5 | ✓ | ✓ |

Tab. 7: #USBCFG Modes & OS

Now we go back to AT commands description.

Enter the following AT commands to change the USB configuration mode.

Set the desired USB mode, for example 5.
AT#USBCFG=5
OK

Check the #USBCFG value.
AT#USBCFG?
#USBCFG: 5
OK

After entering the USB mode, power off/on the module (USB device) to make active the new mode, or enter the AT#REBOOT command.

Chapter 10.1 shows for each USB configuration mode:

- the "Device Manager" screenshot in Windows,
- the information returned by the "dmesg" command in Ubuntu.

## 3.2 Serial Ports & Service Access Points

Enter the AT#PORTCFG=? Test command to know at which Service Access Point a physical serial port is connected in accordance with the used Variant value. It returns, for each Variant value, a short description on the logical connection.

```
AT#PORTCFG=?
#PORTCFG: Variant=0:   AT= USIF0 USB0 USB3; STT (Trace) = USB1
#PORTCFG: Variant=1:   not supported by this product
#PORTCFG: Variant=2:   not supported by this product
#PORTCFG: Variant=3:   AT= USIF0 USIF1 USB0; STT(Trace) = USB1
#PORTCFG: Variant=4:   not supported by this product
#PORTCFG: Variant=5:   not supported by this product
#PORTCFG: Variant=6:   not supported by this product
#PORTCFG: Variant=7:   not supported by this product
#PORTCFG: Variant=8:   AT= USB0 USB3 USB4; STT(Trace) = USB1
#PORTCFG: Variant=9:   not supported by this product
#PORTCFG: Variant=10: not supported by this product
#PORTCFG: Variant=11: AT= USIF0 USB3 USB0; STT(Trace)= USB1;  ExtGNSS= USIF1
OK
```

Message meaning for each Variant value:

Variant=0:

- USIF0, USB0, and USB3 are connected to AT parsers that are not specified in the message. The following sub-chapters specify the AT parsers, and the other Service Access Points for each Variant value.
- USB1 is connected to the TT Service Access Point.

Variant=8:

- USB0, USB3, USB4 are connected to AT parsers that are not specified in the message The following sub-chapters specify the AT parsers, and the other Service Access Points for each Variant value.
- USB1 is connected to the TT Service Access Point.

Variant=...

- And so on for each supported Variant value.

---

The ports arrangement that you have set, can be used with one of the six #USBCFG modes described in Tab. 3.

---

# 4 AT#PORTCFG COMMAND

AT#PORTCFG command manages several internal ports arrangements by means of its parameter value called ‘Variant’, refer to document [2]. The tables and figures reported on the next pages show the various ports configurations obtained changing the "Variant" value of the command and/or connecting the USB cable to the module.
Here is the sequence to make active the entered AT#PORTCFG command:

- Start from the configuration shown in Tab. 8, it is the factory setting: #PORTCFG is 0;
- Enter, for example, the AT#PORTCFG=3 command through USIF0 port, AT0 parser elaborates the just entered command, but no actions are taken;
- Power down the module;
- Power on the module. The AT#PORTCFG=3 command is executed and the ports/services arrangement of Tab. 10 is set. To power off/on the module, you can use the AT#REBOOT command.

The tables on the following pages show the use of the STT trace tools. To have more information contact Telit Technical Support.

## 4.1 AT#PORTCFG=0

| AT#PORTCFG=0 (Factory Setting) | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |

Tab. 8: #PORTCFG=0, no USB Cable

| AT#PORTCFG=0 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| USB0 | | X | | | |
| USB1 | | | | STT (Trace) | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |

Tab. 9: #PORTCFG=0, with USB Cable

The USBx port configuration mode depends on the mode set using the AT#USBCFG command, see Tab. 3: #USBCFG Mode.

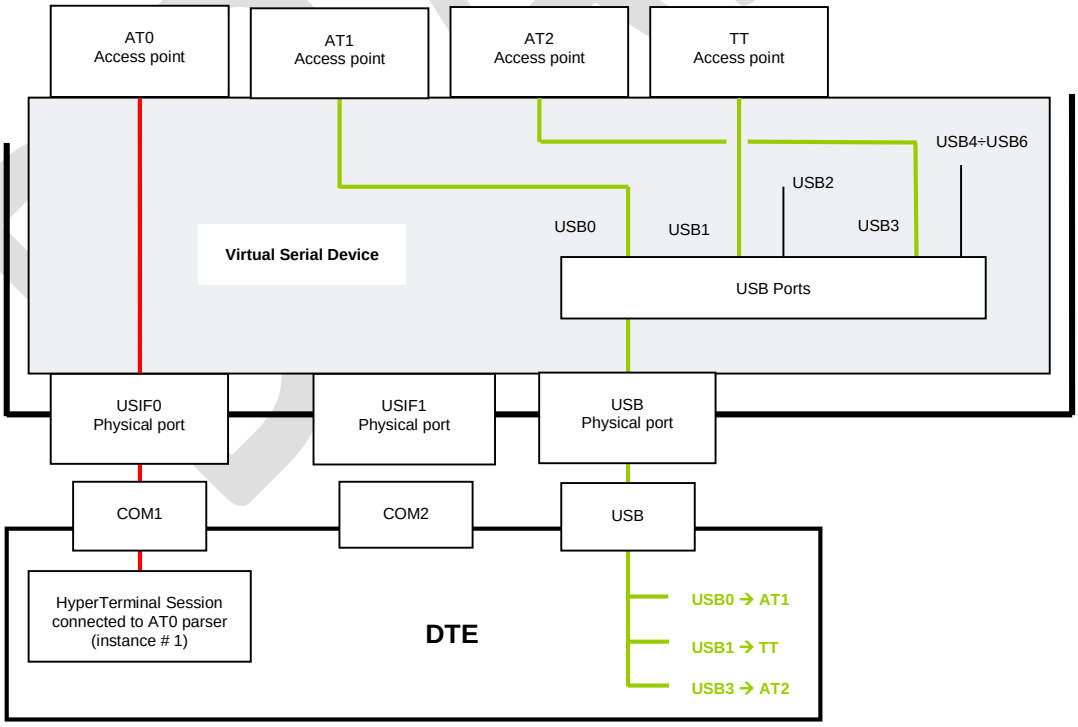



Fig. 3: #PORTCFG=0 + USB Cable

## 4.2 AT#PORTCFG=3

| AT#PORTCFG=3 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | X | | |

Tab. 10: #PORTCFG=3, no USB Cable

| AT#PORTCFG=3 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| USB0 | | X | | | |
| USB1 | | | | STT (Trace) | |
| USB2 | | | | | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | X | | |

Tab. 11: #PORTCFG=3, with USB Cable

The USBx port configuration mode depends on the mode set using the AT#USBCFG command, see Tab. 3: #USBCFG Mode.

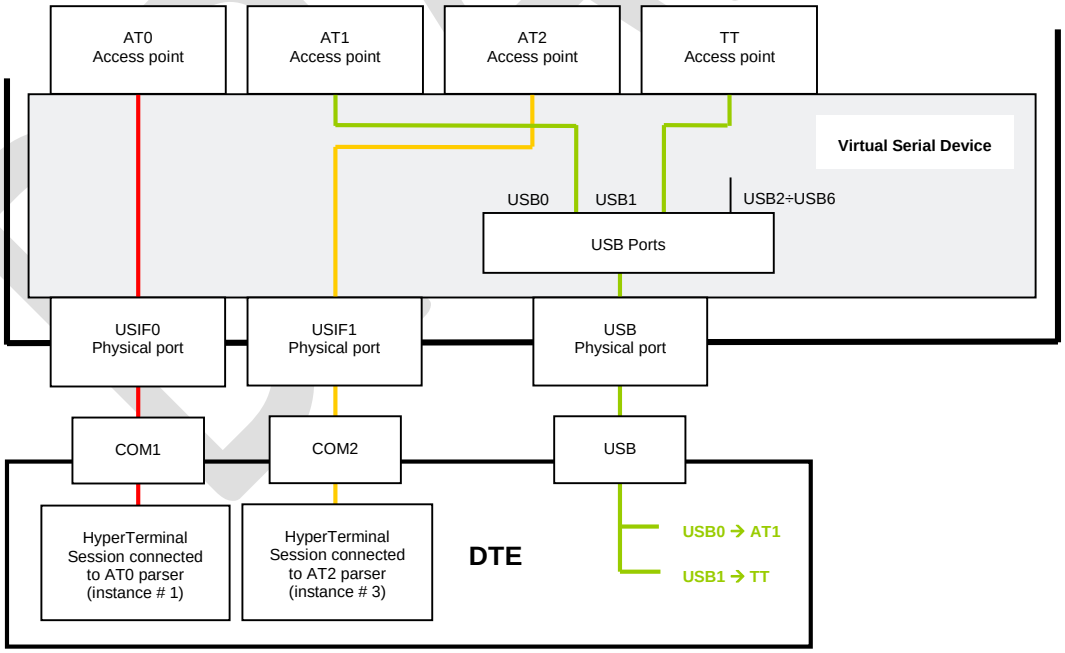



Fig. 4: #PORTCFG=3 + USB Cable

# 4.3 AT#PORTCFG=8

| AT#PORTCFG=8 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| NO USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |

Tab. 12: #PORTCFG=8, no USB Cable

| AT#PORTCFG=8 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| USB0 | X | | | | |
| USB1 | | | | STT (Trace) | |
| USB2 | | | | | |
| USB3 | | X | | | |
| USB4 | | | X | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |

Tab. 13: #PORTCFG=8, with USB Cable

The USBx port configuration mode depends on the mode set using the AT#USBCFG command, see Tab. 3: #USBCFG Mode.

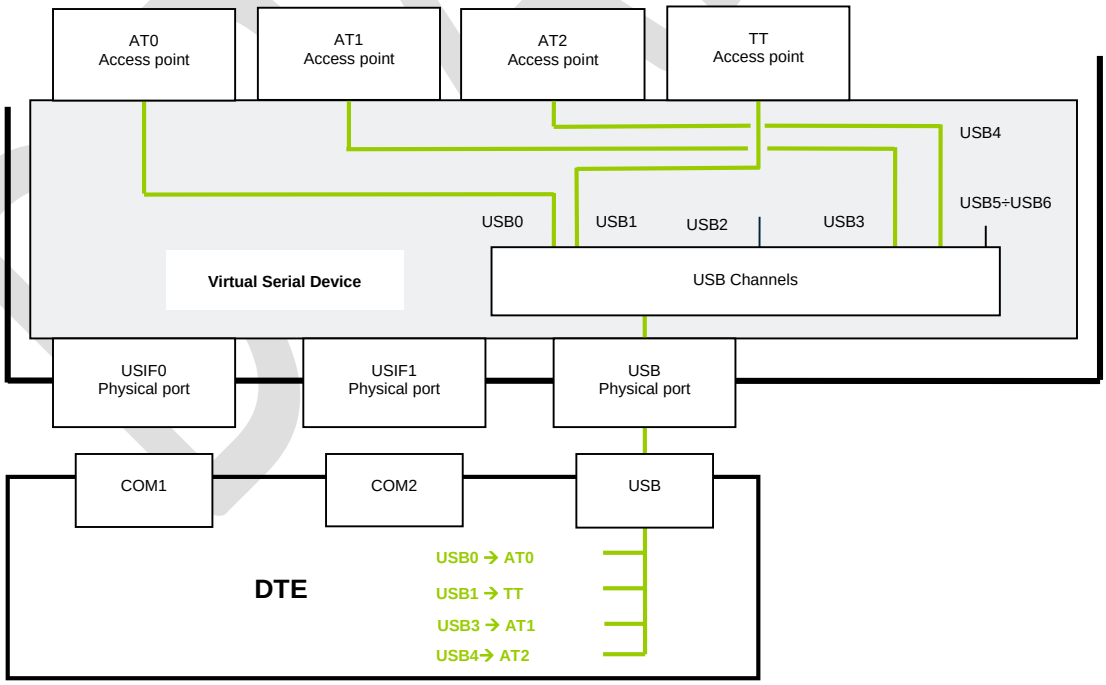



Fig. 5: #PORTCFG=8 USB Cable Only

# 4.4 AT#PORTCFG=11

AT#PORTCFG=11 is used to connect the module to an external GNSS receiver through USIF1 serial port, see chapter 6.1

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | Ext. GNSS |

Tab. 14: #PORTCFG=11, no USB Cable

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| USB0 | | X | | | |
| USB1 | | | | STT (Trace) | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | Ext. GNSS |

Tab. 15: #PORTCFG=11, with USB Cable

The USBx port configuration mode depends on the mode set using the AT#USBCFG command, see Tab. 3: #USBCFG Mode.

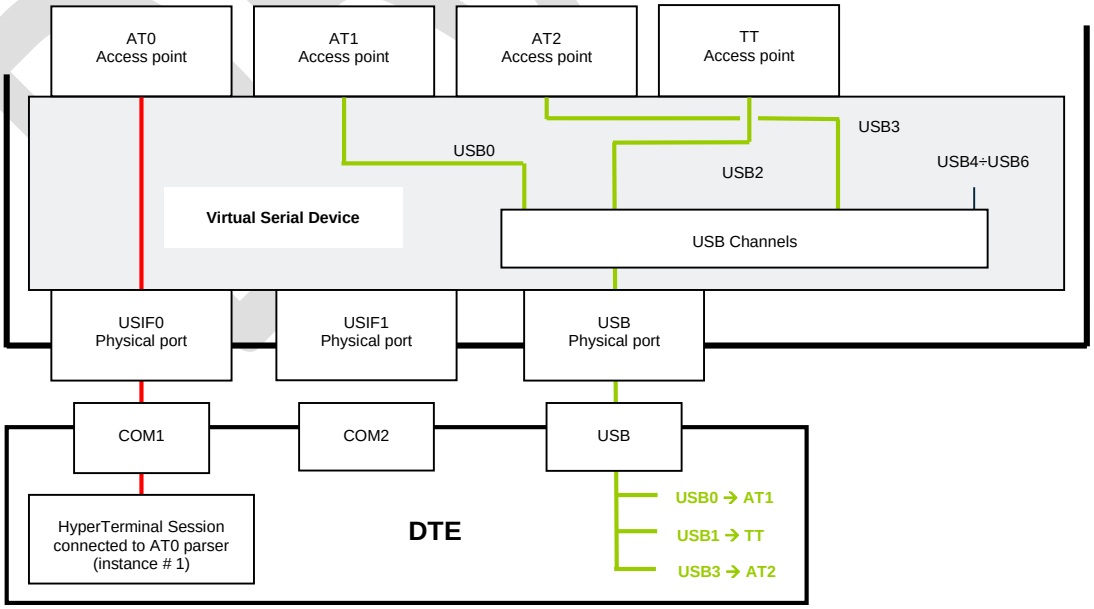



Fig. 6: #PORTCFG=11 + USB Cable

# 5 CMUX PROTOCOL

This section shows examples of ports/services arrangement using CMUX protocol. If you need to develop a Multiplexing Protocol running on your application processor (for example a user micro-controller), refer to document [1] to get detailed information.

## 5.1 CMUX Protocol on USIF0 Port

Here is an example of ports/services arrangement supporting CMUX protocol on USIF0 serial port. Assume that the module is configured as indicated in Tab. 8: #PORTCFG=0 (default), and no USB cable plugged in. In addition, suppose that the used DTE is a Windows-PC, and Fig. 7 shows its device configuration. Now, run on the DTE the TELIT Serial Port MUX application configured as shown in Fig. 8, and connect the MUX application to COM1 physical port, refer to Fig. 9. When the user starts an application (for example Hyper Terminal) connected to one of the three Virtual Ports (COM20 ÷ COM22), TELIT Serial Port MUX application sends automatically the AT+CMUX=0 command to the module and the CMUX protocol is activated.

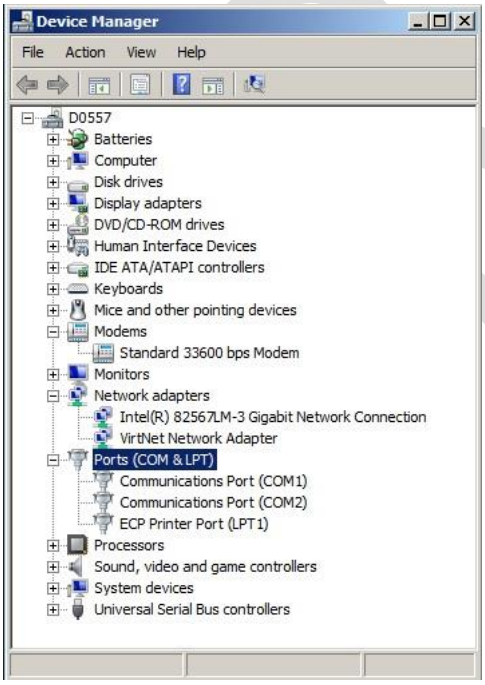



Fig. 7: Physical COMx Ports

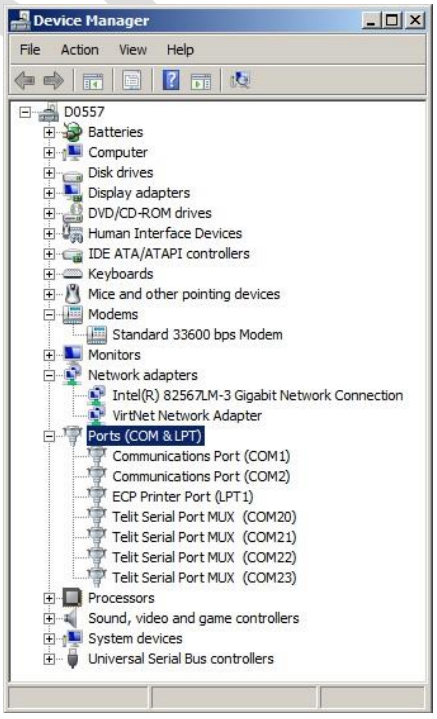



Fig. 8: Virtual Serial Ports of MUX

The configuration of the TELIT Serial Port MUX application must avoid virtual serial ports conflict with the physical or virtual serial ports already present on the Windows-PC. The table below summarizes the new configuration.

| Module ←→DTE connection | COMx → VCx | AT0 | AT1 | AT2 | TT |
|---|---|---|---|---|---|
| USB not used | | | | | |
| USIF0 ←→COM1 | COM20→VC1 | X | | | |
| | COM21→VC2 | | X | | |
| | COM22→VC3 | | | X | |
| | COM23→VC4 | | | | |
| USIF1 not used | | | | | |

Tab. 16: Ports/Services Arrangement with CMUX Connected to USIF0

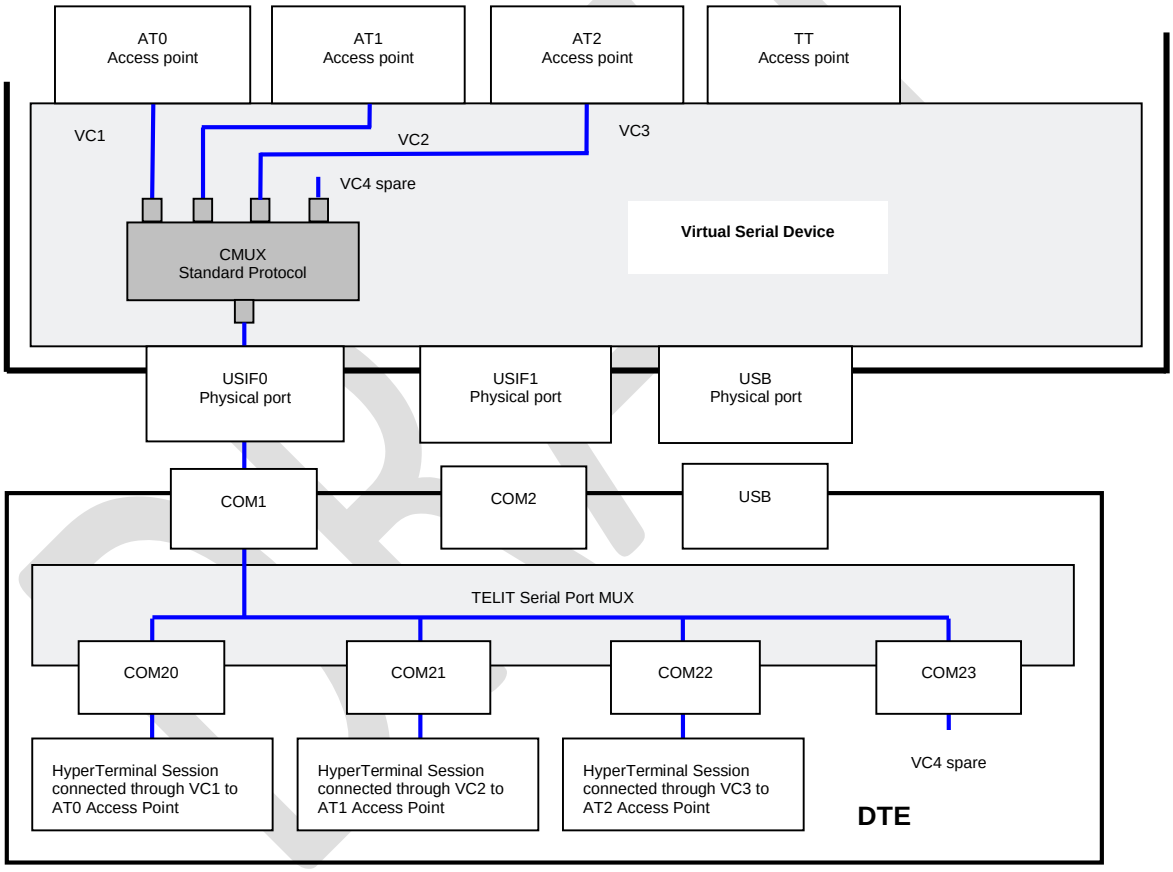



Fig. 9: CMUX Connected to USIF0

## 5.2 CMUX Protocol on USB3 Port

Here is an example of ports/services arrangement supporting CMUX protocol on USB3 port.

Assume that the module is configured as indicated in Fig. 3: #PORTCFG=0 (factory setting), and USB cable plugged in. In addition, suppose that the used DTE is a Windows PC, and Fig. 2 shows its device configuration. Now, run on the DTE the TELIT Serial Port MUX application configured as shown in Fig. 10, and connect the MUX application to USB3 port mapped into COM16 virtual port, refer to Fig. 11. When the user starts an application (for example Hyper Terminal) connected to one of the three Virtual Ports (COM20 ÷ COM22), TELIT Serial Port MUX application sends automatically the AT+CMUX=0 command to the module and the CMUX protocol is activated.

---

In the current Windows-PC configuration the USB0 port is mapped into COM17 port, refer to Fig. 10.

---

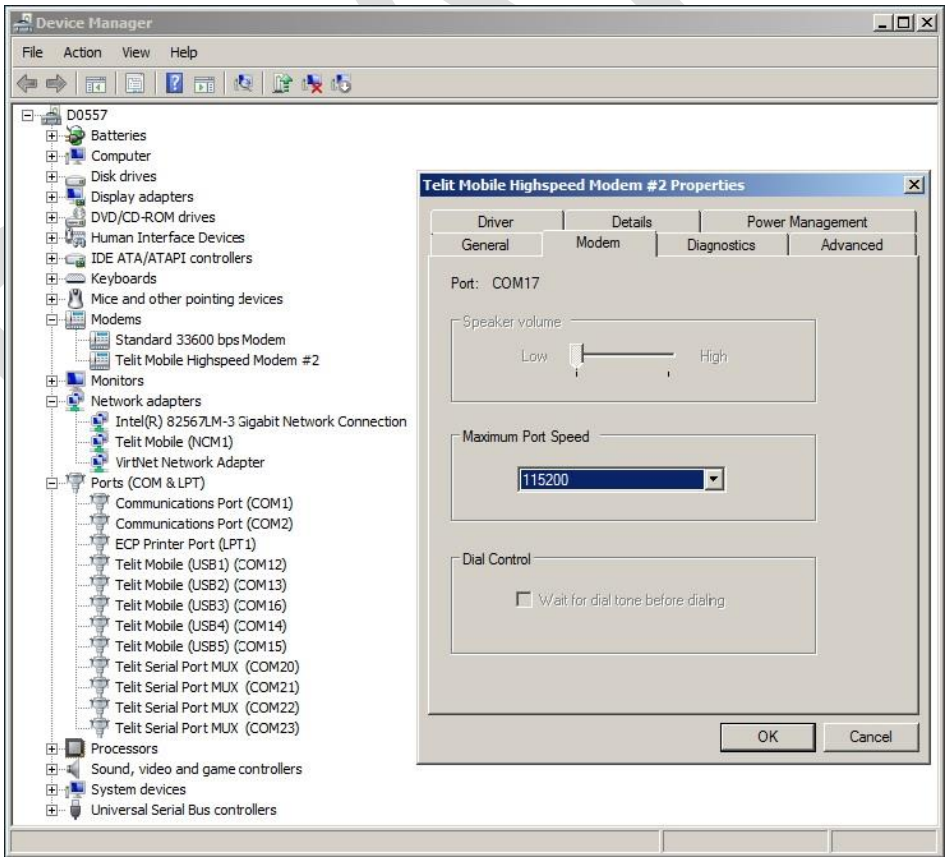



Fig. 10: Virtual Serial Ports of Telit Serial Port MUX

The table below summarizes the new configuration.

| Module ←→DTE connection | USB Ports | USBx → COMx | COMx → VCx | AT0 | AT1 | AT2 | TT | / |
|---|---|---|---|---|---|---|---|---|
| USB ←→ USB | USB0 | | | | | | | |
| | USB1 | | | | | | STT | |
| | USB2 | | | | | | | |
| | USB3 | COM16 | COM20 → VC1 | X | | | | |
| | | | COM21 → VC2 | | X | | | |
| | | | COM22 → VC3 | | | X | | |
| | | | COM23 → VC4 | | | | | |
| | USB4 | | | | | | | |
| | USB5 | | | | | | | |
| | USB6 | | | | | | | |
| USIF0 not used | | | | | | | | |
| USIF1 not used | | | | | | | | |

Tab. 17: Ports/Services Arrangement with CMUX Connected to USB3 Port

**NOTICE**: AT0 (instance # 1) is disconnected from USIF0 and connected to VC1/USB3/COM16/COM20. The STT tool stays on USB1 port.

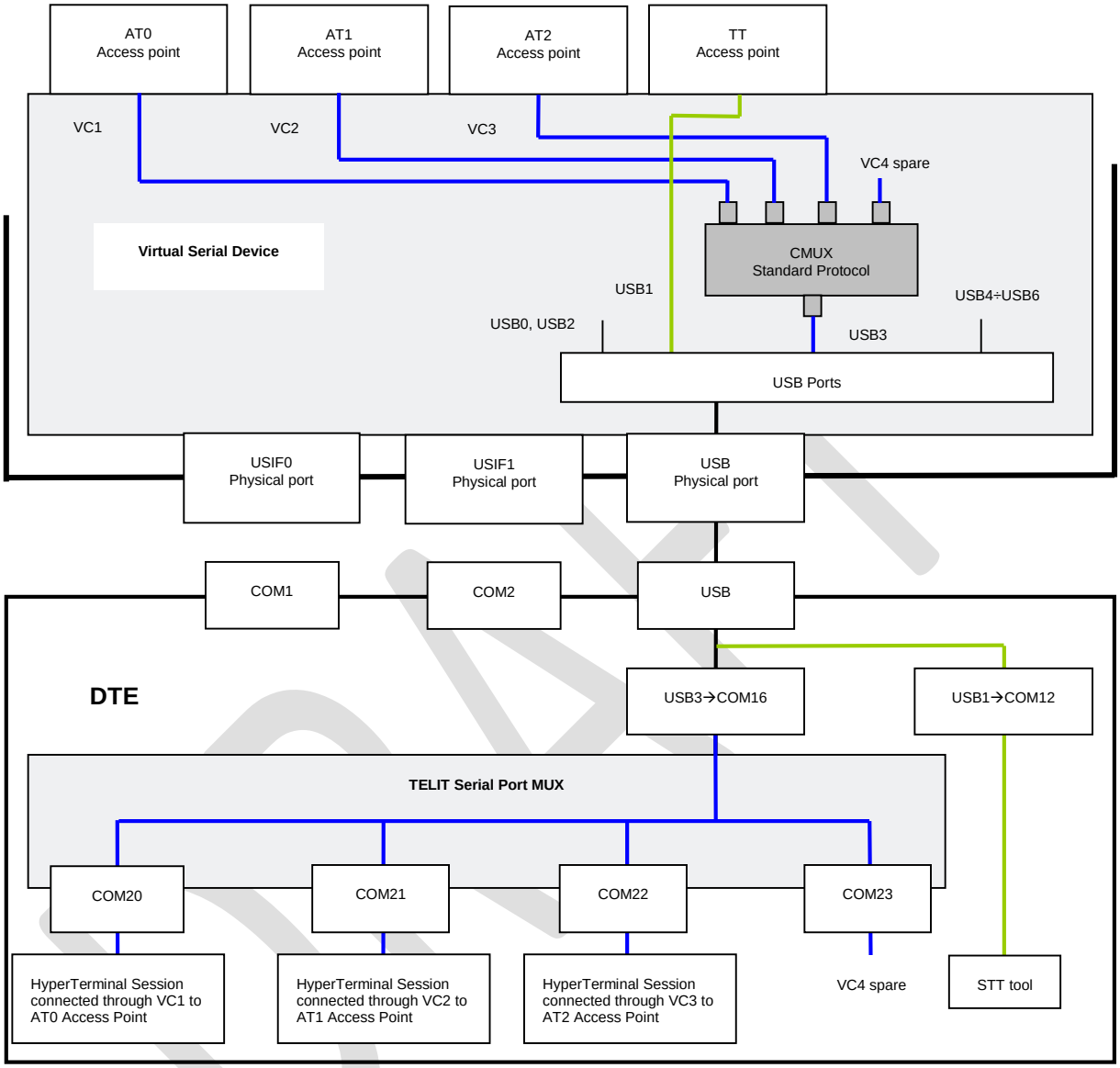



Fig. 11: CMUX Connected to USB3 Port

# 6 SERVICES

The modules series covered by the present document provide the services as indicated in the **SERVICES COEXISTENCE TABLE**, see chapter 9.

Different Access Points connect these services to the Virtual Serial Device software layer. This section describes how the user can access the supported service by means of the external physical serial ports, which in their turn are connected to the Virtual Serial Device.

## 6.1 External GNSS Receiver

The external GNSS receiver is connected to the module through USIF1 serial port.

### 6.1.1 AT#PORTCFG=11

Tab. 14 summarizes the starting configuration of the module (#PORTCFG=11). Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… commands entered through USIF0 port. AT0 parser executes the AT commands, and after that NMEA sentences and AT commands run on USIF0 port as shown in Fig. 12.

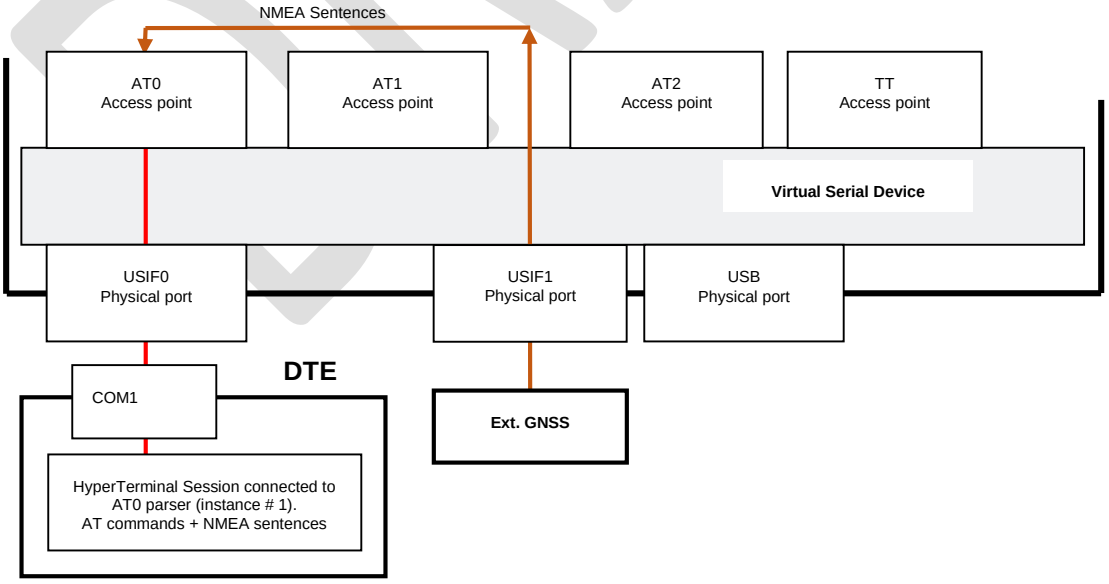



Fig. 12: USIF0 Port Support AT Commands + NMEA Sentences (External GNSS)

### 6.1.2 AT#PORTCFG=11 + USB

Tab. 15 summarizes the starting configuration of the module (#PORTCFG=11+USB). Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… commands entered, for example, through USB0 port. AT1 parser executes the AT commands, and after that NMEA sentences and AT commands run on USB0 port as shown in Fig. 13.

**NOTICE:** the user can issue the AT commands through USIF0/AT0, USB0/AT1, or USB3/AT2 port/parser. The NMEA sentences are routed respectively on USIF0, USB0 (as shown by the example), or USB3 port

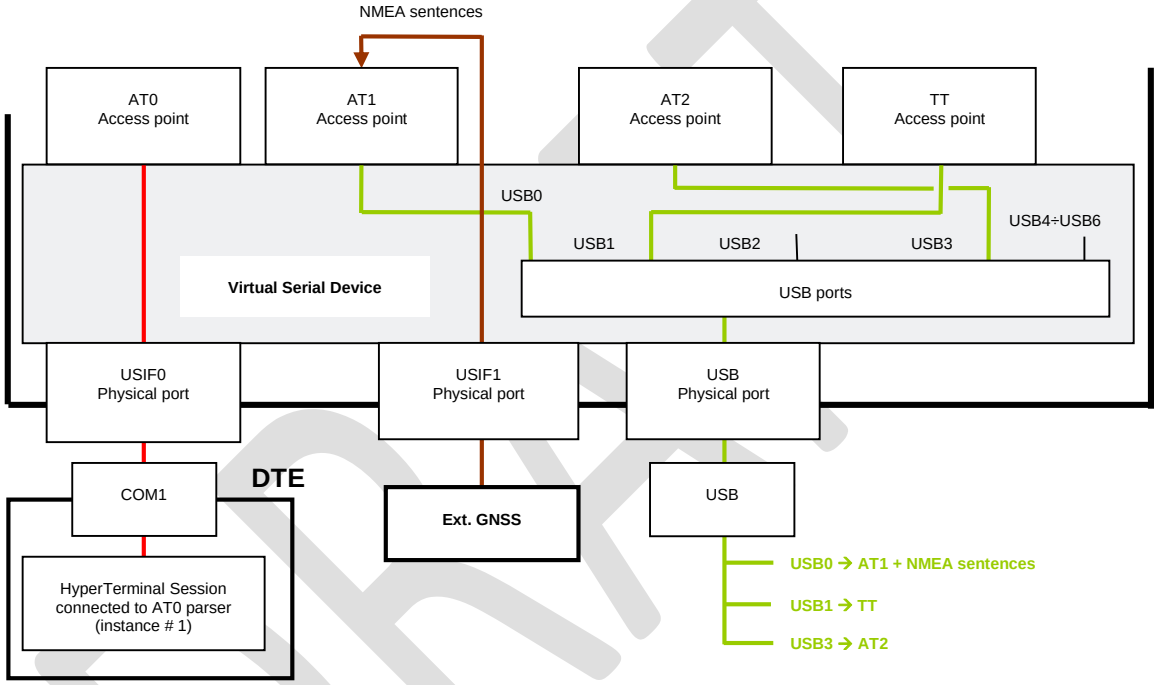



Fig. 13: USB0 Port Support AT Commands + NMEA Sentences (External GNSS)

# 6.2 AppZone

Refer to documents [7], and [8] to have information on the AppZone layer and its functions (APIs).

## 6.2.1 USIFx Ports

Example 1

Tab. 8 shows the starting ports configuration of the module: #PORTCFG=0 (factory setting), no USB cable is connected. For example, run a user AppZone Application that does not use neither serial ports nor any ATx parsers; Fig. 14 shows the resulting ports arrangement.

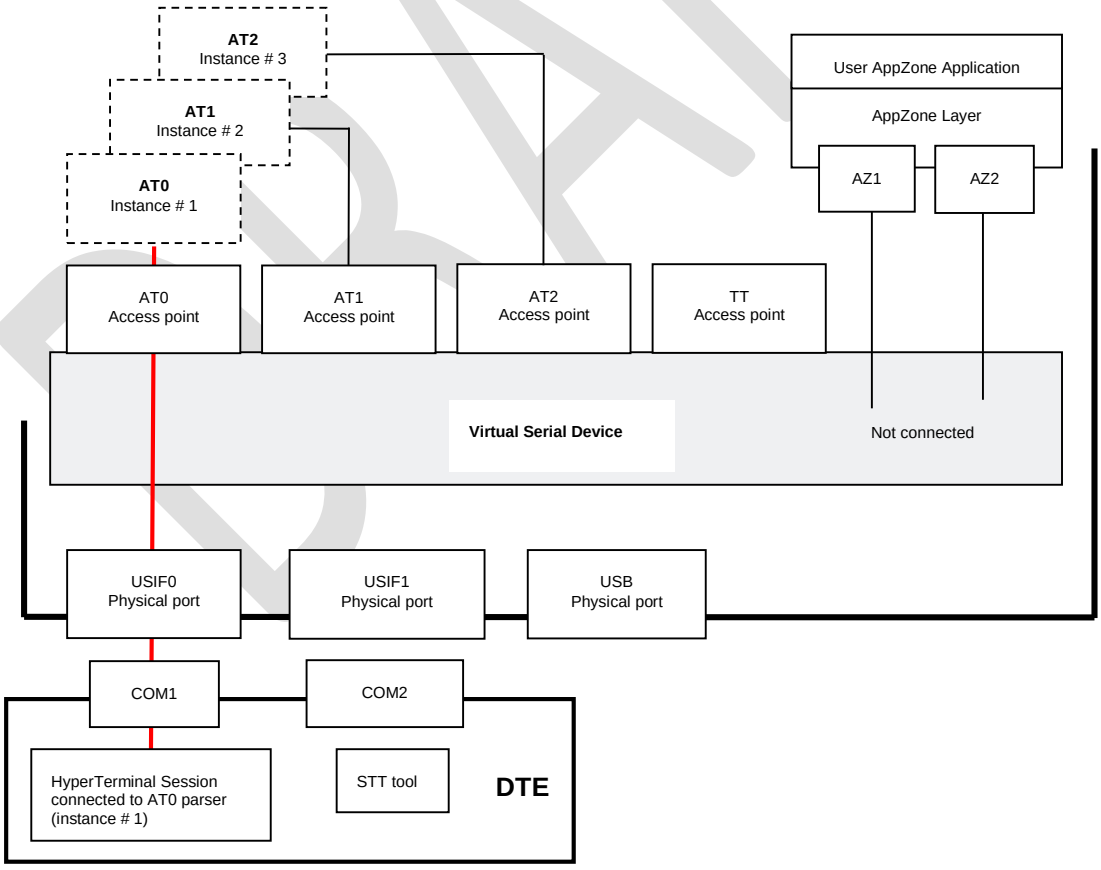



Fig. 14: AppZone Application without Connections

Example 2

Starting from the configuration of the Example 1, use **m2m_os_iat_set_at_command_instance(...)** function to connect logically the AZ1 and AZ2 Service Access Points respectively to AT1 and AT2 parsers, in addition use **PrintToUart(...)** function to use USIF0 port. Fig. 15 shows the resulting ports configuration.

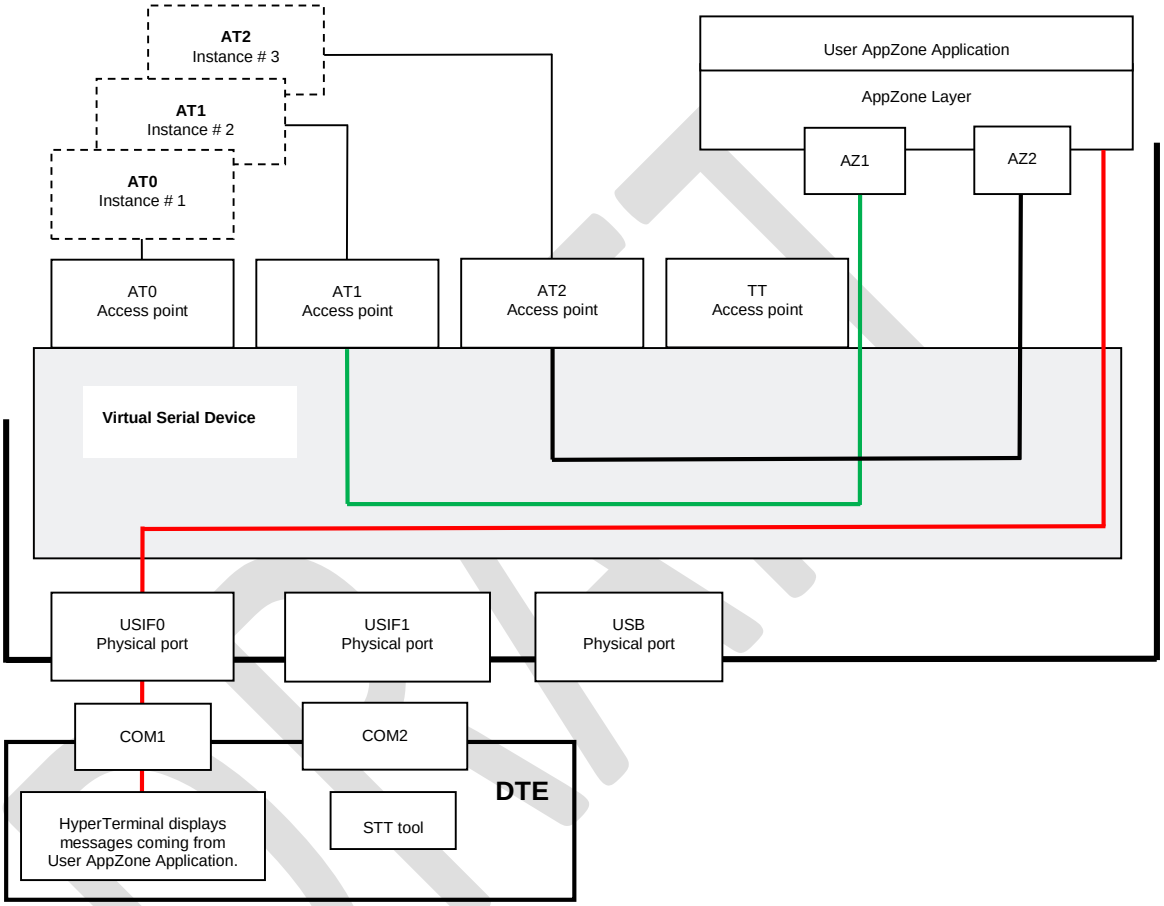



Fig. 15: AppZone Application Connected to AT1, AT2 Parsers, and USIF0 Serial Port

Example 3

Starting from the configuration of the Example 1, use

**m2m_hw_uart_ioctl (uart_fd, M2M_HW_UART_IO_AT_MODE_SET, M2M_HW_UART_IO_AT_MODE_ON)**

function to route data, received from USIF0, to AT1 parser. Fig. 16 shows the resulting ports configuration.

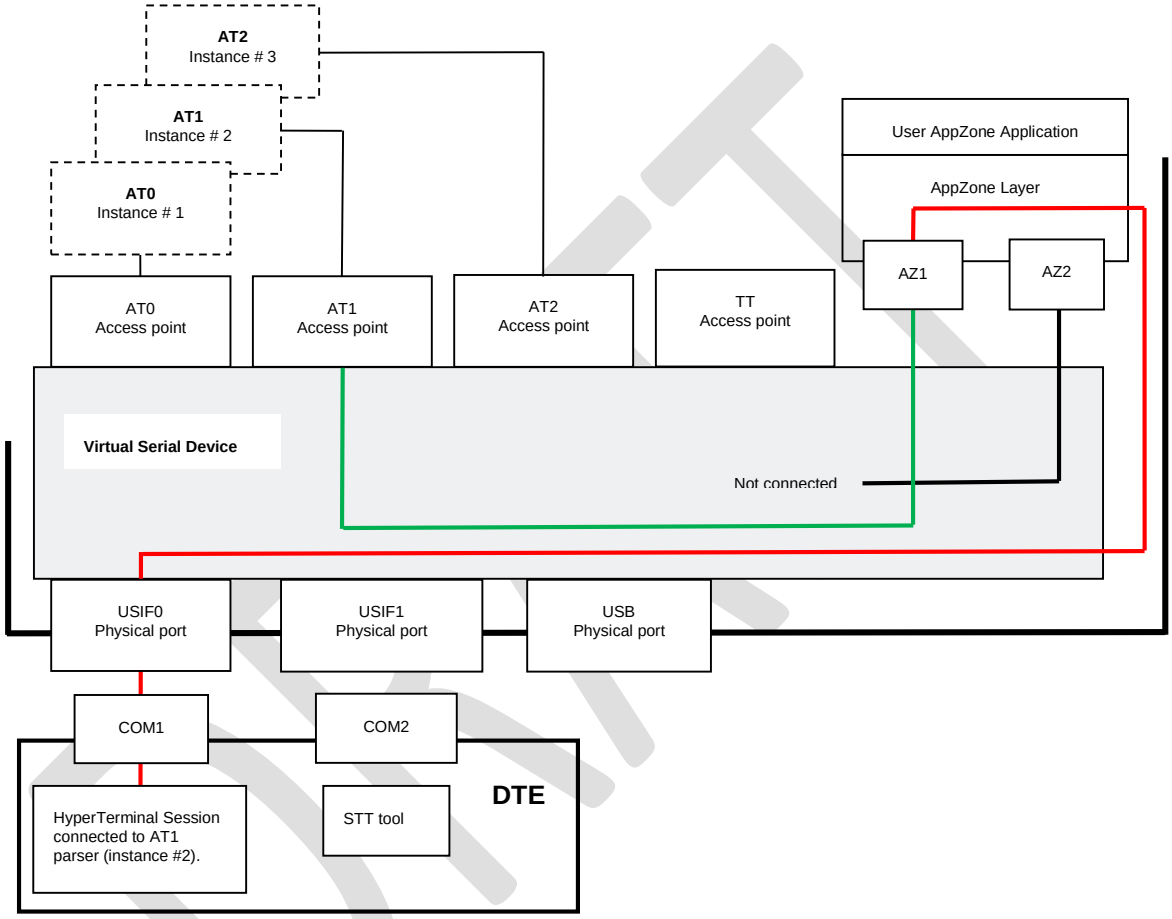



Fig. 16: USIF0 Connected to AT1 Parser through AppZone Layer

### 6.2.2 USB Ports and Instances

Example 1

Tab. 8 shows the starting configuration of the module: #PORTCFG=0 (factory setting), no USB cable is connected. Now, connect the USB cable to the module. The module recognizes the “plug in” event and assumes the factory ports/services arrangement depicted in Fig. 3.

**NOTICE**: Out of the seven USB ports, only two ports are available for the user AppZone applications.

Referring to Fig. 17, use:
**m2m_hw_usb_open(USB_CH0, handle0)** to disconnect USB0 port from AT1 parser, connect it to AppZone layer, and get its handle.

**m2m_hw_usb_open(USB_CH3, handle3)** to disconnect USB3 port from AT2 parser, connect it to AppZone layer, and get its handle.

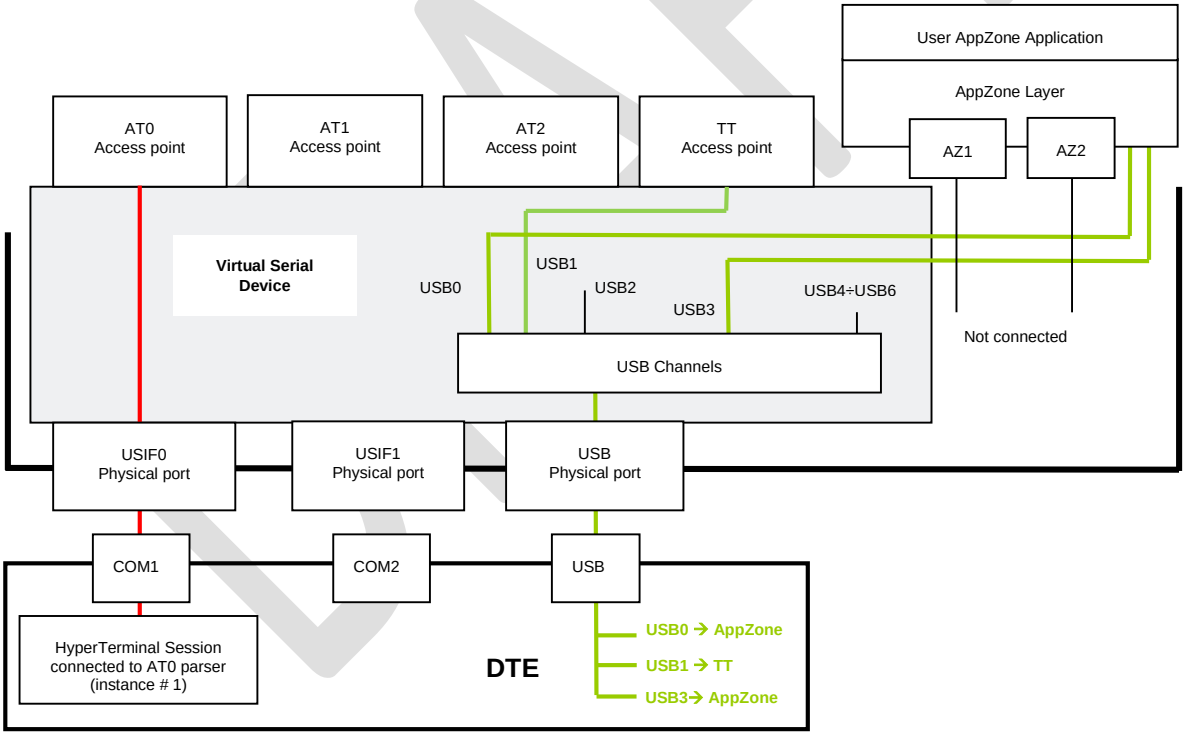



Fig. 17: USB0 and USB3 Ports Available to AppZone Application

**NOTICE**: **m2m_hw_usb_open(...)** returns the control to the calling task only if the USB cable is connected.

Use:
**m2m_hw_usb_read(...)** or **m2m_hw_usb_write(...)** to read or write data from/to USB0 and USB3 ports.

The table below summarizes the relationship among the following items:

USB instance: used to manage, for example, lock and unlock among USB ports.
handle: used to manage the USB port
channel: used to identified the name of the USB port

| USB instance, refer to: m2m_hw_usb_get_instance | Handle, refer to: m2m_hw_usb_open | USB channel name, refer to: m2m_hw_usb_getch_from_handle |
|---|---|---|
| USER_USB_INSTANCE_T enum | handle | M2M_USB_CH enum |
| USER_USB_INSTANCE_T enum | handle | M2M_USB_CH enum |
| USER_USB_INSTANCE_T enum | handle | M2M_USB_CH enum |

Tab. 18: USB Instances, Handles, and Channels (or Ports)

Example 2

Tab. 13 summarizes the starting ports configuration of the module: #PORTCFG=8, with USB cable.

**NOTICE**: starting the user AppZone Application from the configuration shown in the Tab. 13, three USB ports are available for the user application: USB0, USB3, and USB4. USB1 is reserved for STT trace.

Referring to Fig. 18, use:
**m2m_hw_usb_open(USB_CH0, handle0)** to disconnect USB0 port from AT0 parser, connect it to AppZone layer, and get its handle.

**m2m_hw_usb_open(USB_CH3, handle3)** to disconnect USB3 port from AT1 parser, connect it to AppZone layer, and get its handle.

**m2m_hw_usb_open(USB_CH4, handle4)** to disconnect USB4 port from AT2 parser, connect it to AppZone layer, and get its handle.

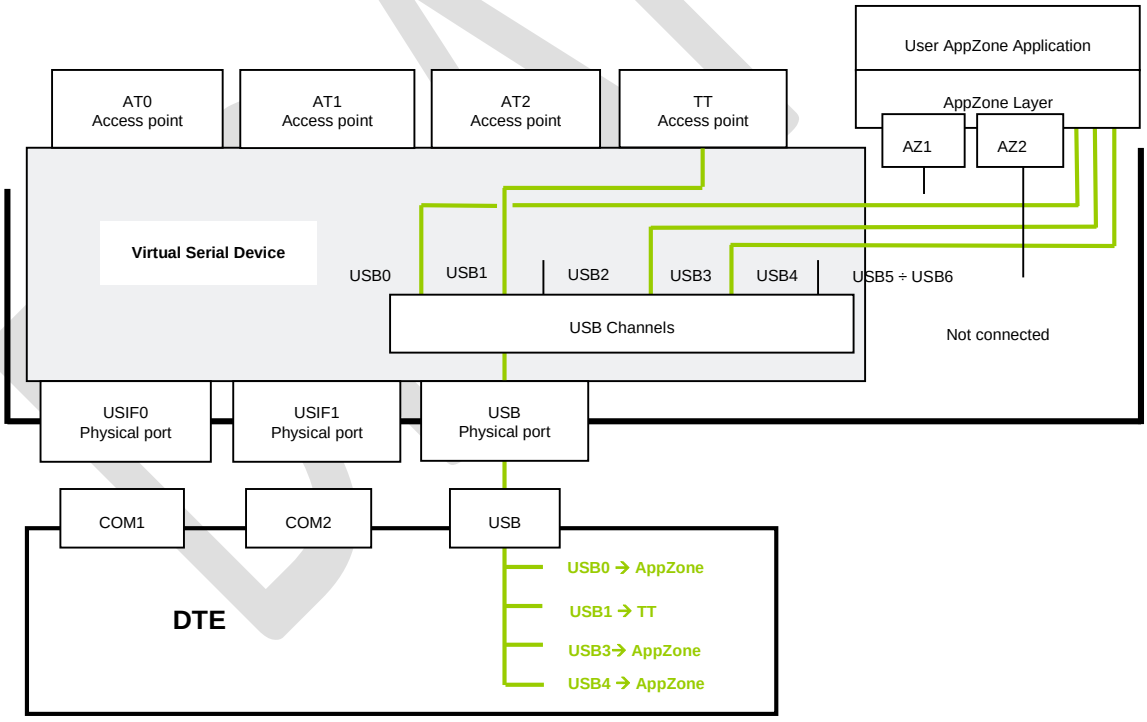



Fig. 18: USB0, USB3, and USB4 Ports Available to AppZone Application

Use:
**m2m_hw_usb_read(...)** or **m2m_hw_usb_write(...)** to read or write data from/to USB0, USB3, and USB4 ports.

Example 3

Tab. 13 summarizes the starting ports configuration of the module: #PORTCFG=8, with USB cable.

Referring to Fig. 19, use:
**m2m_hw_usb_ioctl(handle4, M2M_USB_AT_MODE_SET, M2M_HW_USB_IO_AT_MODE_ON)**
function to route data received from USB4 port to AT1 parser.

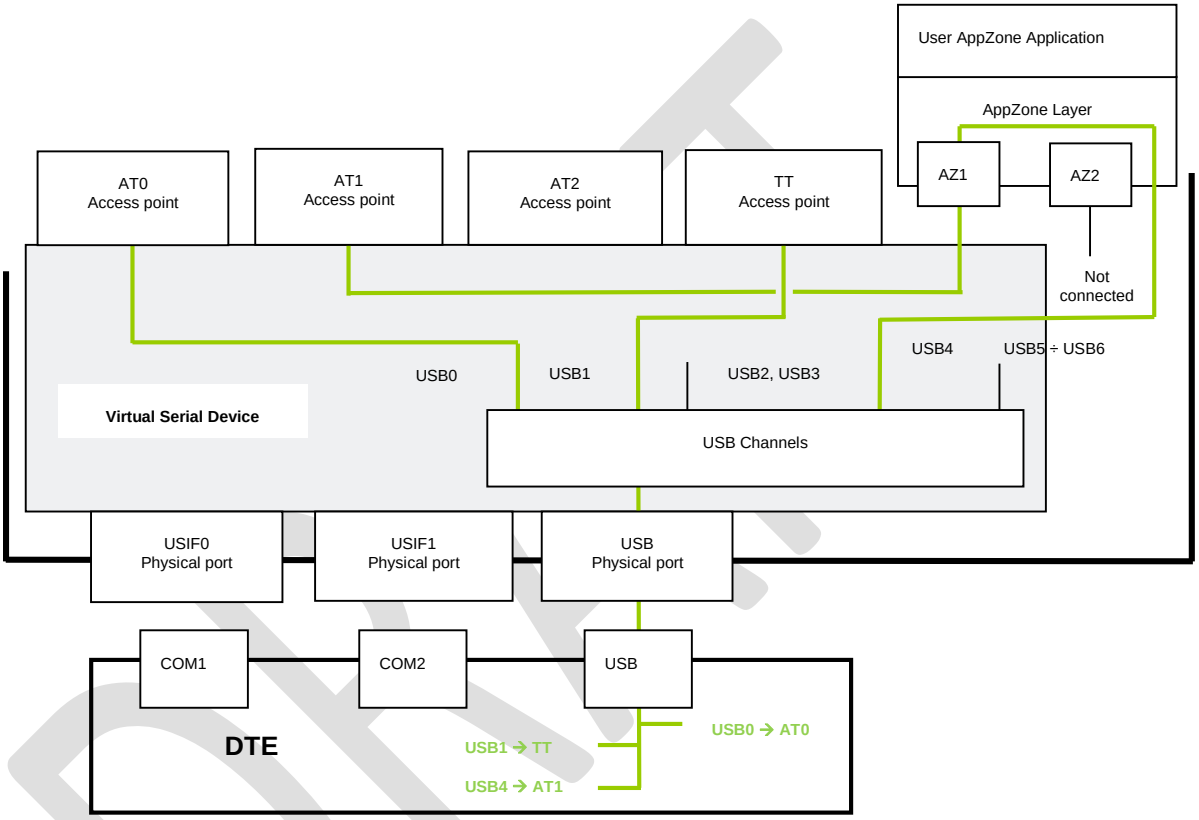



Fig. 19: USB4 Port connected to AT1

# 7 THE WINNING CONFIGURATION

Here are two examples showing that the last port configuration set by the user overrides the previous one.

There are two ways to change module ports/services arrangement in addition to use AT#PORTCFG command:

- Plug in/out the USB cable;
- Enter the AT+CMUX=0 command.

**NOTICE**: TELIT Serial Port MUX application sends automatically the AT+CMUX=0 command to the module, see chapter 5.

Example 1

Module: Tab. 8: #PORTCFG=0, no USB Cable summarizes module ports configuration.

User action: runs the TELIT Serial Port MUX application on Windows-PC; the application connects logically Virtual Ports COM20÷COM23 to COM1.

PC: provides the required Virtual Ports. When the user starts an application (for example Hyper Terminal) connected to one of the three Virtual Ports (the fourth one is spare), TELIT Serial Port MUX application sends the AT+CMUX=0 command to the module.

Module: in accordance with the received command, the involved AT Parser starts the CMUX protocol. The module enters the configuration shown on Fig. 9.

User action: now, the user connects USB cable.

Module: enters the configuration shown on Fig. 3: #PORTCFG=0 + USB Cable.

PC: provides seven new virtual “COM” connected logically to the seven USB ports. The CMUX protocol is disabled, and the TELIT Serial Port MUX application running on Windows PC is no more connected to the module, it should be closed. COM1 is ready for new applications (e.g. Hyper Terminal).

User action: disconnects USB cable.

Module: enters again the configuration shown in Tab. 8: #PORTCFG=0, no USB Cable.

Example 2

Module: Tab. 8: #PORTCFG=0, no USB Cable summarizes module ports configuration.

User action: connects USB cable.

Module: in accordance with the user action, the module enters the configuration shown in Fig. 3: #PORTCFG=0 + USB Cable

PC: provides seven virtual “COM” required by USB drivers to connect logically the seven USBx ports.

User action: runs the TELIT Serial Port MUX application on the Windows PC; the application connects logically Virtual Ports VCOM20 ÷ VCOM23 to USB3→COM16 port.

PC: provides the required Virtual Ports. When the user starts an application (e.g. Hyper Terminal) on a Virtual Ports, TELIT Serial Port MUX sends the AT+CMUX=0 command to the module.

Module: in accordance with the received command, the involved AT Parser starts the CMUX protocol. The module enters the configuration shown in Fig. 11

User action: disconnects USB cable.

Module: enters again the configuration shown on Tab. 8: #PORTCFG=0, no USB Cable.

PC: discards the seven virtual “COM” connected logically to the seven USBx ports. The CMUX protocol is disabled, TELIT Serial Port MUX application running on Windows PC is no more connected to the module, and it should be closed.

The two examples show that the last required port configuration overrides the previous one.

# 8 ACRONYMS & ABBREVIATIONS

| | |
|---|---|
| ACM | Abstract Control Model |
| DTE | Data Terminal Equipment |
| GNSS | Global Navigation Satellite System |
| MBIM | Mobile Broadband Interface Model |
| NCM | Network Control Model |
| NMEA | National Marine Electronics Association |
| PID | Product IDentifier |
| USIFx | Universal Serial Interface |
| VSD | Virtual Service Device |

# 9 MODULES & SW VER. TABLE

**SOFTWARE VER. TABLE**

| | SW Versions |
|---|---|
| **LE910 V2 Series** | |
| LE910-SV V2 | 20.00.000-B0xx |
| LE910-AU V2 | 20.00.100-B0xx |
| LE910-EU V2 | 20.00.400-B0xx |
| LE910-NA V2 | 20.00.500-B0xx |

The table below summarizes the Services provided by the modules, and shows their coexistence. The available Service depends on the software version installed on the modules.

**SERVICES COEXISTENCE TABLE**

| | Services | |
|---|---|---|
| | **External GNSS** | **AppZone** |
| **LE910 V2 Series** | | |
| LE910-SV V2 | ✓ | ✓* |
| LE910-AU V2 | ✓ | ✓* |
| LE910-EU V2 | ✓ | ✓* |
| LE910-NA V2 | ✓ | ✓* |

(*): AppZone available on demand on specific part numbers.

# 10 APPENDIXES

## 10.1 #USBCFG Modes

If you are using a Windows-PC, before plugging in the USB cable, and issuing the AT commands described in the following chapters, install the USB driver provided by Telit, see chapter 10.3.

### 10.1.1 #USBCFG=0

Set USB mode 0 (factory setting).
AT#USBCFG=0
OK

Activate the just set mode
AT#REBOOT
OK

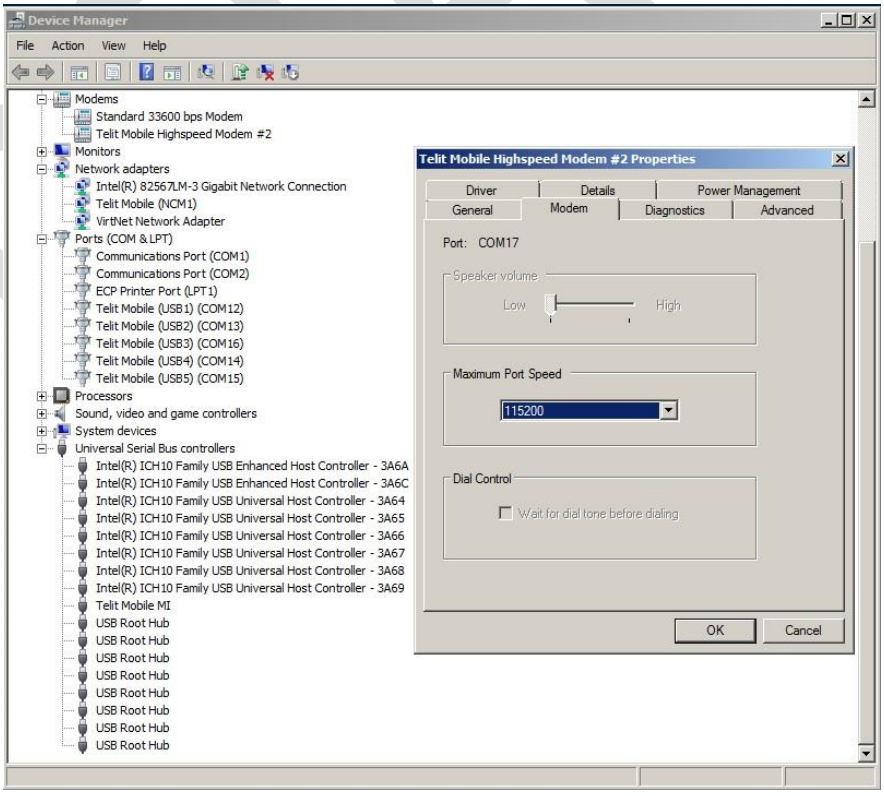
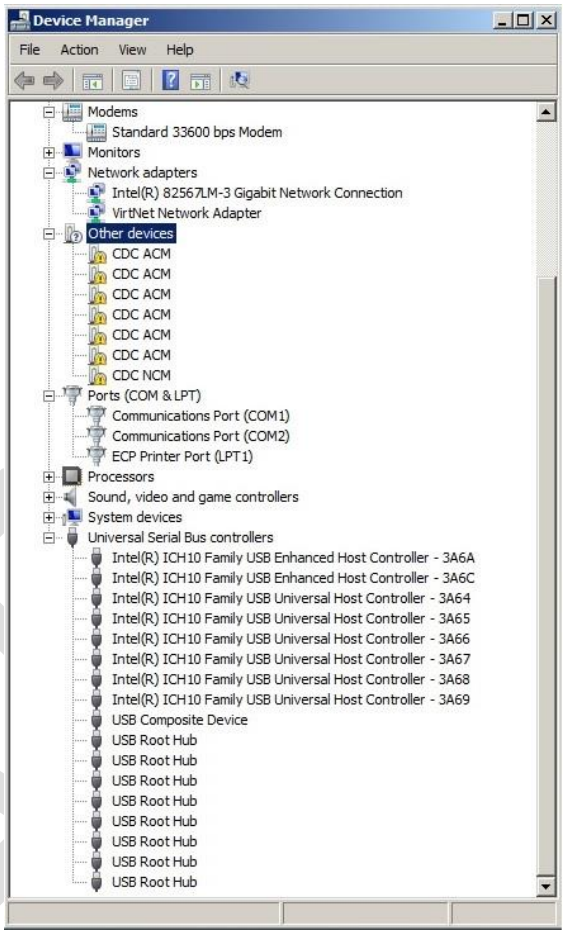



Fig. 20: #USBCFG=0

If the device driver is not installed, and you plug in the USB cable, the "Device Manager" displays the folder "Other devices". See the figure on the right side, and compare it with the Tab. 3, mode=0.




### *10.1.1.1 Ubuntu*

Fig. 21 shows the message returned by the dmesg command when you plug in the USB cable. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0036 (#USBCFG=0).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0036
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: FIH7160
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 351622079900102
cdc_acm 1-3:1.0: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.0: ttyACM0: USB ACM device
cdc_acm 1-3:1.2: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.2: ttyACM1: USB ACM device
cdc_acm 1-3:1.4: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.4: ttyACM2: USB ACM device
cdc_acm 1-3:1.6: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.6: ttyACM3: USB ACM device
cdc_acm 1-3:1.8: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.8: ttyACM4: USB ACM device
cdc_acm 1-3:1.10: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.10: ttyACM5: USB ACM device
cdc_ncm 1-3:1.12: MAC-Address: 00:00:11:12:13:14
cdc_ncm 1-3:1.12: setting rx_max = 16384
cdc_ncm 1-3:1.12 usb0: register 'cdc_ncm' at usb-0000:00:1a.7-3, CDC NCM, 00:00:11:12:13:14
IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready
```

Fig. 21: #USBCFG=0, Ubuntu

### 10.1.2 #USBCFG=1

#USBCFG=1 mode in progress.

Set USB mode 1.
AT#USBCFG=1
OK

Activate the just set mode
AT#REBOOT
OK

### 10.1.3 #USBCFG=2

Set USB mode 2.
AT#USBCFG=2
OK

Activate the just set mode
AT#REBOOT
OK

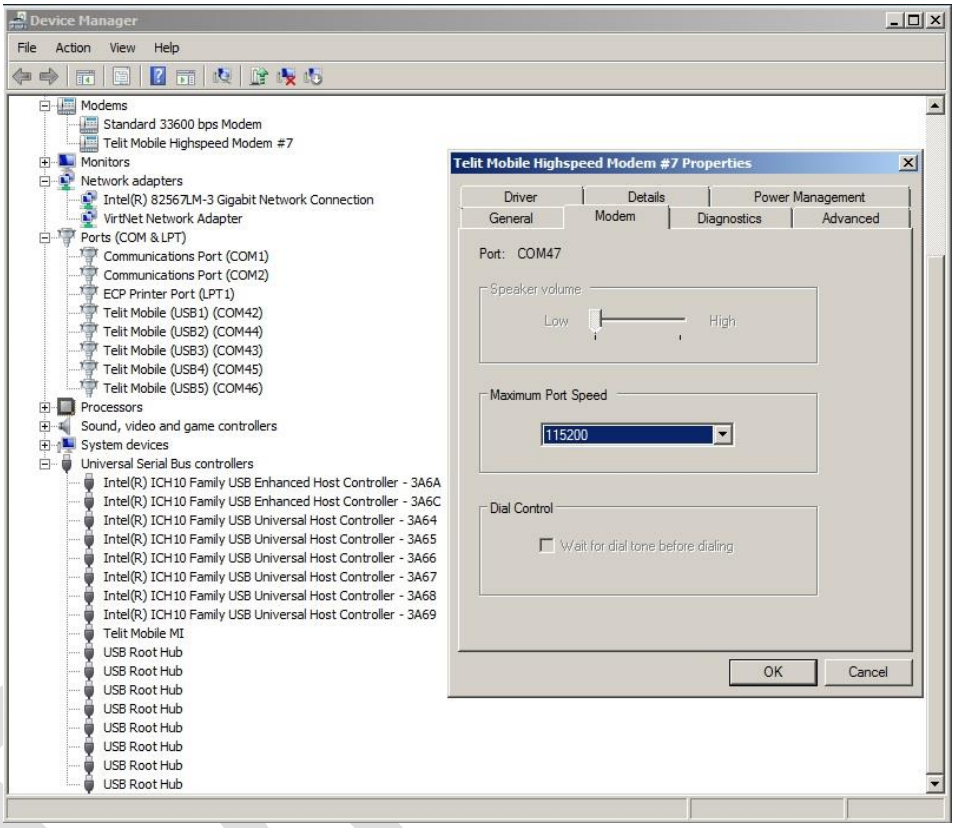
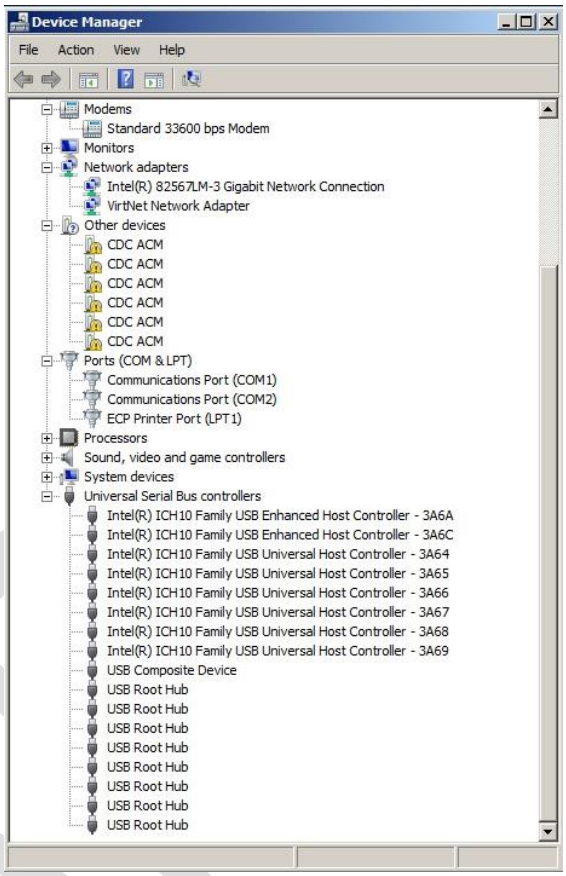



Fig. 22: #USBCFG=2

If the device driver is not installed, and you plug in the USB cable, the "Device Manager" displays the folder "Other devices". See the figure on the right side, and compare it with the Tab. 3, mode=2.




### *10.1.3.1 Ubuntu*

Fig. 23 shows the message returned by the dmesg command when you plug in the USB cable. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0035 (#USBCFG=2).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0035
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: FIH7160
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 351622079900102
cdc_acm 1-3:1.0: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.0: ttyACM0: USB ACM device
cdc_acm 1-3:1.2: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.2: ttyACM1: USB ACM device
cdc_acm 1-3:1.4: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.4: ttyACM2: USB ACM device
cdc_acm 1-3:1.6: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.6: ttyACM3: USB ACM device
cdc_acm 1-3:1.8: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.8: ttyACM4: USB ACM device
cdc_acm 1-3:1.10: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.10: ttyACM5: USB ACM device
```

Fig. 23: #USBCFG=2, Ubuntu

### 10.1.4 #USBCFG=3

Set USB mode 3.
AT#USBCFG=3
OK

Activate the just set mode
AT#REBOOT
OK

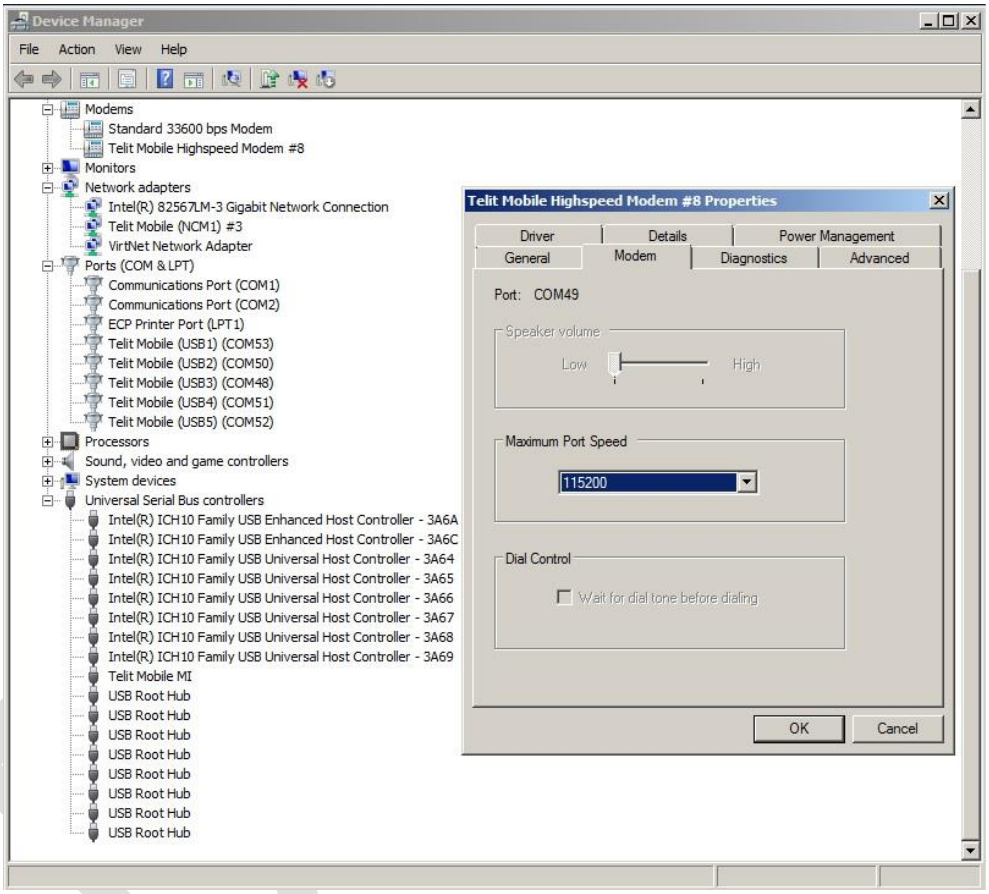
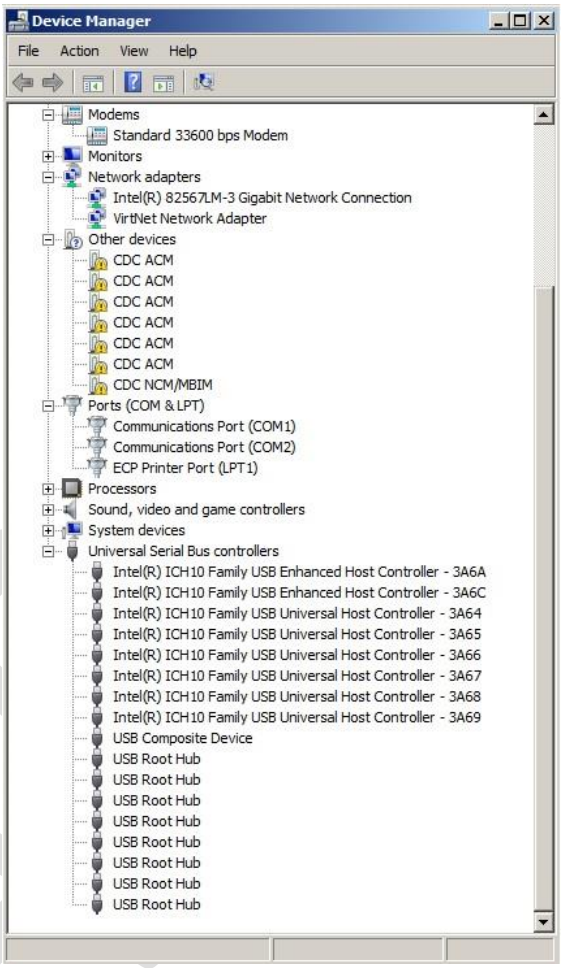



Fig. 24: #USBCFG=3

If the device driver is not installed, and you plug in the USB cable, the "Device Manager" displays the folder "Other devices". See the figure on the right side, and compare it with the Tab. 3, mode=3.




### *10.1.4.1 Ubuntu*

Fig. 25 shows the message returned by the dmesg command when you plug in the USB cable. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0032 (#USBCFG=3).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0032
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: FIH7160
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 351622079900102
cdc_acm 1-3:1.0: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.0: ttyACM0: USB ACM device
cdc_acm 1-3:1.2: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.2: ttyACM1: USB ACM device
cdc_acm 1-3:1.4: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.4: ttyACM2: USB ACM device
cdc_acm 1-3:1.6: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.6: ttyACM3: USB ACM device
cdc_acm 1-3:1.8: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.8: ttyACM4: USB ACM device
cdc_acm 1-3:1.10: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.10: ttyACM5: USB ACM device
cdc_mbim 1-3:1.12: setting rx_max = 16384
cdc_mbim 1-3:1.12: cdc-wdm0: USB WDM device
cdc_mbim 1-3:1.12 wwan0: register 'cdc_mbim' at usb-0000:00:1a.7-3, CDC MBIM, ee:a7:2d:7f:ce:7d
```

Fig. 25: #USBCFG=3, Ubuntu

### 10.1.5 #USBCFG=4

Set USB mode 4.
AT#USBCFG=4
OK

Activate the just set mode
AT#REBOOT
OK

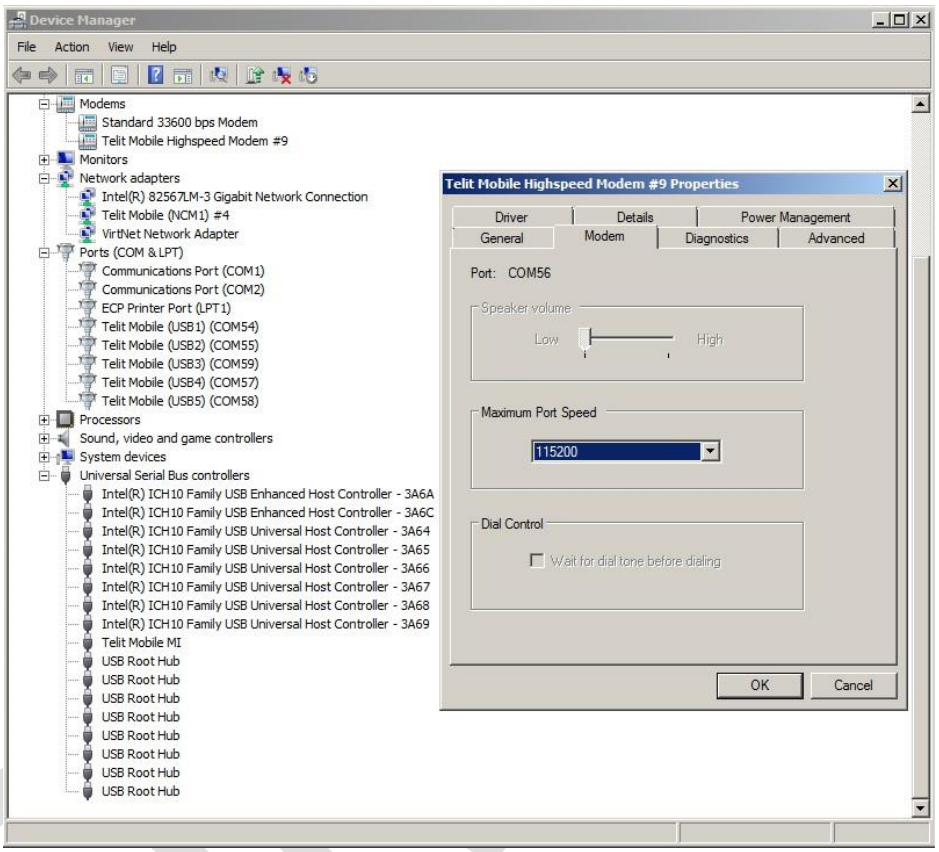
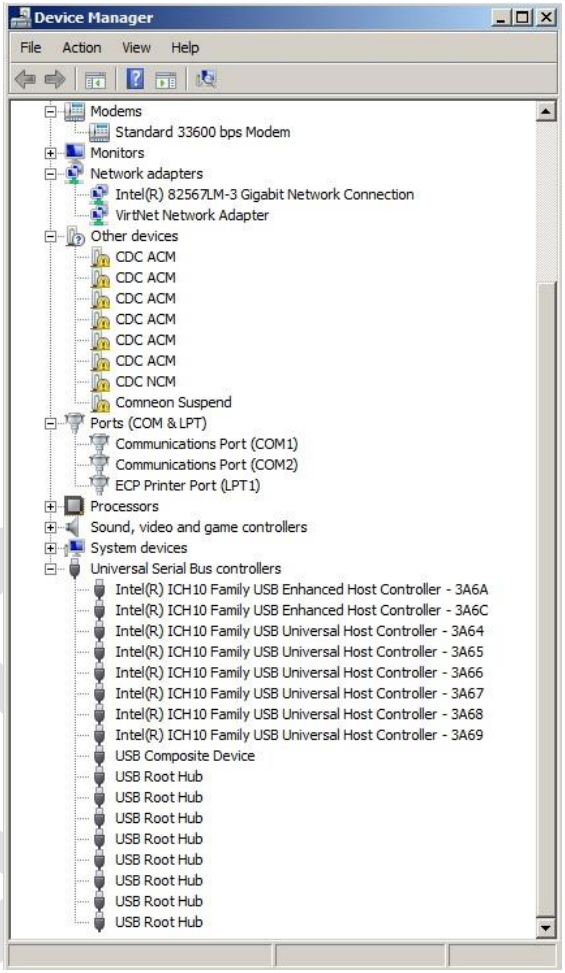



Fig. 26: #USBCFG=4

If the device driver is not installed, and you plug in the USB cable, the "Device Manager" displays the folder "Other devices". See the figure on the right side, and compare it with the Tab. 3, mode=4.




### *10.1.5.1 Ubuntu*

Fig. 27 shows the message returned by the dmesg command when you plug in the USB cable. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0037 (#USBCFG=4).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0037
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: FIH7160
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 351622079900102
cdc_acm 1-3:1.0: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.0: ttyACM0: USB ACM device
cdc_acm 1-3:1.2: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.2: ttyACM1: USB ACM device
cdc_acm 1-3:1.4: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.4: ttyACM2: USB ACM device
cdc_acm 1-3:1.6: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.6: ttyACM3: USB ACM device
cdc_acm 1-3:1.8: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.8: ttyACM4: USB ACM device
cdc_acm 1-3:1.10: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.10: ttyACM5: USB ACM device
cdc_ncm 1-3:1.13: MAC-Address: 00:00:11:12:13:14
cdc_ncm 1-3:1.13: setting rx_max = 16384
cdc_ncm 1-3:1.13 usb0: register 'cdc_ncm' at usb-0000:00:1a.7-3, CDC NCM, 00:00:11:12:13:14
IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready
```

Fig. 27: #USBCFG=4, Ubuntu

### 10.1.6 #USBCFG=5

Set USB mode 5.
AT#USBCFG=5
OK

Activate the just set mode
AT#REBOOT
OK

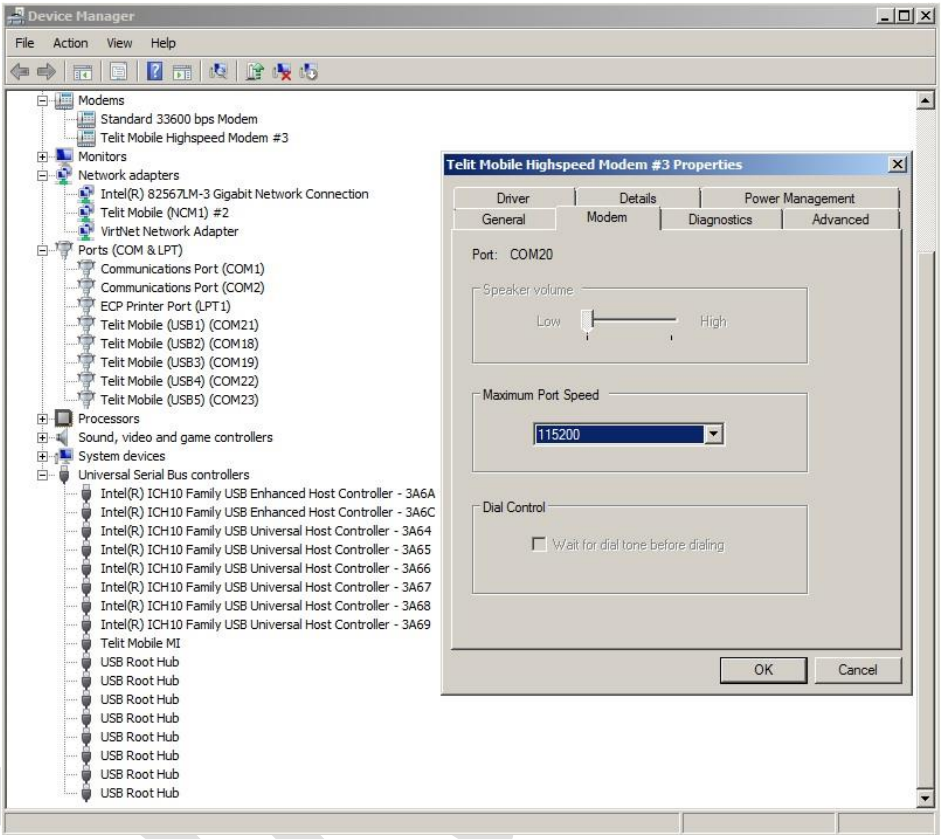
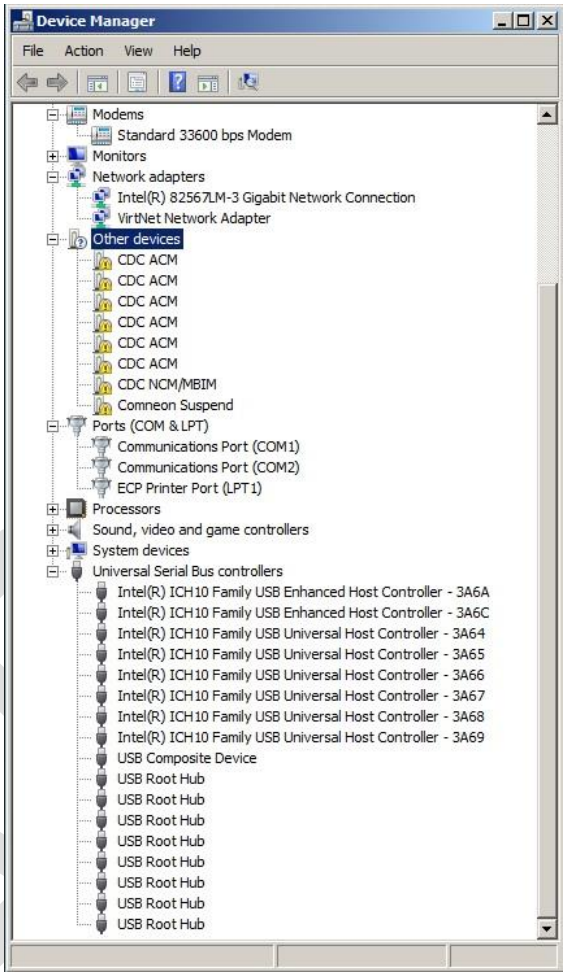



Fig. 28: #USBCFG=5

If the device driver is not installed, and you plug in the USB cable, the "Device Manager" displays the folder "Other devices". See the figure on the right side, and compare it with the Tab. 3, mode=5.




### *10.1.6.1 Ubuntu*

Fig. 29 shows the message returned by the dmesg command when you plug in the USB cable. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0033 (#USBCFG=5).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0033
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: FIH7160
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 351622079900102
cdc_acm 1-3:1.0: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.0: ttyACM0: USB ACM device
cdc_acm 1-3:1.2: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.2: ttyACM1: USB ACM device
cdc_acm 1-3:1.4: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.4: ttyACM2: USB ACM device
cdc_acm 1-3:1.6: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.6: ttyACM3: USB ACM device
cdc_acm 1-3:1.8: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.8: ttyACM4: USB ACM device
cdc_acm 1-3:1.10: This device cannot do calls on its own. It is not a modem.
cdc_acm 1-3:1.10: ttyACM5: USB ACM device
cdc_mbim 1-3:1.13: setting rx_max = 16384
cdc_mbim 1-3:1.13: cdc-wdm0: USB WDM device
cdc_mbim 1-3:1.13 wwan0: register 'cdc_mbim' at usb-0000:00:1a.7-3, CDC MBIM, ee:a7:2d:7f:ce:7d
```

Fig. 29: #USBCFG=5, Ubuntu

## 10.2 USB Interfaces & Endpoints

The target of this chapter is to provide a short overview of USB interfaces and endpoints.

Referring to Fig. 30:

Endpoints

Each USB device has a number of endpoints. Each endpoint is a source or sink of data. In general, a device can have up to 16 out and 16 in endpoints. Out always means from host (for example, Windows-PC) to device (module). In always means from device to host.
Endpoint 0 is a special case, which is a combination of endpoint 0 out and endpoint 0 in (EP0), and is used for controlling the device.

Pipe

A logical data connection between the host and a particular endpoint.

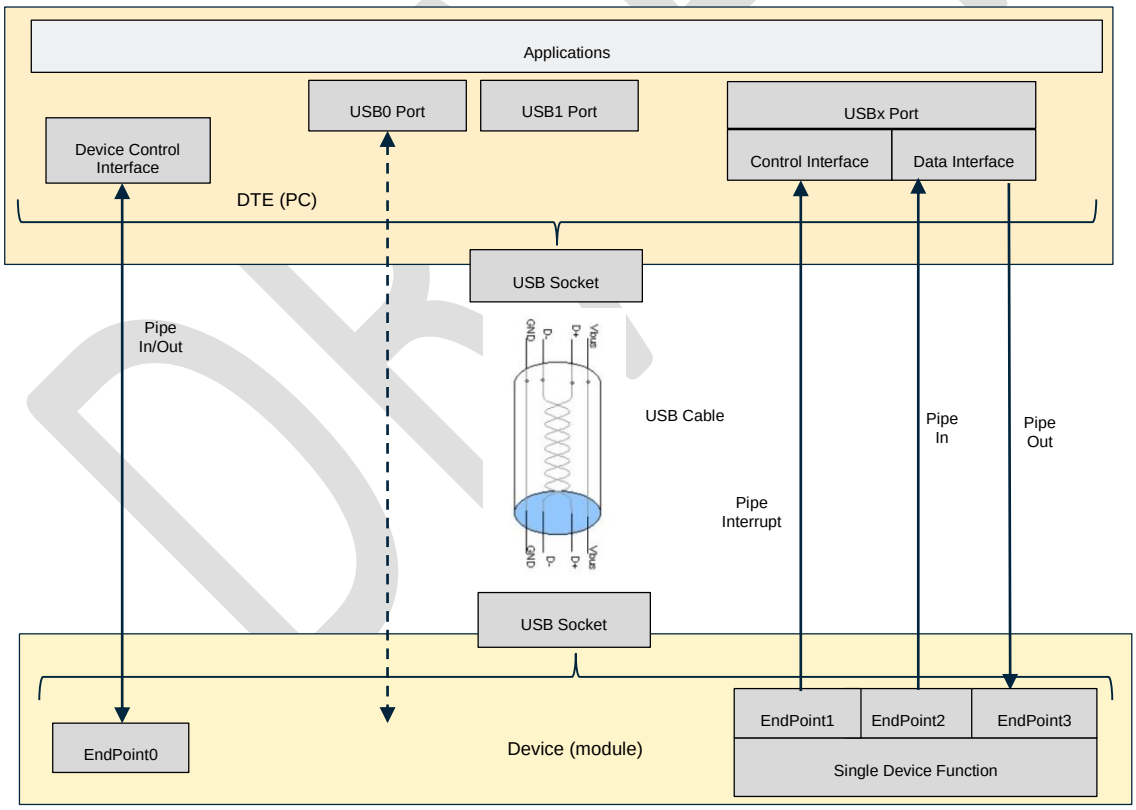



Fig. 30: USB Interfaces & Endpoints

## 10.3 USB Driver

Telit provides the USB driver to install on Windows-PC. Before installing the driver, it is suggested to remove the old one, if present. To verify if the right USB driver has been installed check its Vendor Identifier and the Product Identifier by means of the GUI provided by Windows, see the screenshot below.

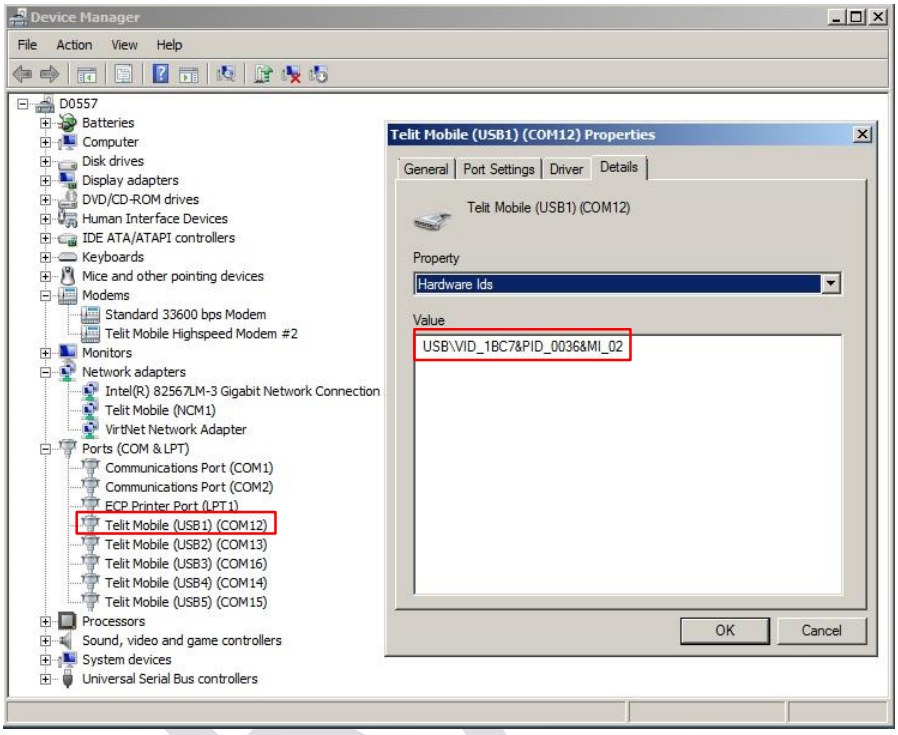



Fig. 31: Vendor Identifier & Product Identifier

- Telit Vendor Identifier is 1BC7.
- Product Identifier depends on the #USBCGF mode, refer to Tab. 3

### 10.3.1 Ubuntu

With Ubuntu, the USB device (module) can work with in-box drivers. It means that no Telit's USB driver is needed. To get Telit Vendor Identifier and Product Identifier in Ubuntu environment use the command dmesg as shown in sub-chapters of the chapter 10.1. To have more information refer to document [4].

# 11 DOCUMENT HISTORY

| Revision | Date | Changes |
|---|---|---|
| 0 | 2016-01-21 | First issue |


Design Review
120°
180°
Process
Validation
240°
RF/EMC Pre-Certification
Module Selection
60°
300°
Approvals
Support
360°
Value Added Services & Connectivity
Enterprise Integration &
Application Enablement
m2mair