# TELIT 3G MODULES PORTS ARRANGEMENTS

1VV0300971 Rev. 12 – 2017-02-14

TELIT
TECHNICAL
DOCUMENTATION

[01.2017]

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

## NOTICES LIST

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

## COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

## COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

# USAGE AND DISCLOSURE RESTRICTIONS

## I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

## II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation 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 prior written permission of Telit

## III. High Risk Materials

Components, units, or third-party products used in the product 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 and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

## IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

## V. 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.

# APPLICABILITY TABLE

## PRODUCTS

| | SW Versions | Modules |
|---|---|---|
| HE910 SERIES | 12.00.xxx | 3G |
| UE910 SERIES | 12.00.xxx | |
| UL865 SERIES | 12.00.xxx | |
| UE866 SERIES | 12.00.xxx | |

# CONTENTS

# FIGURES LIST

# TABLES LIST

# 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 (GPS, Python, 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, TT, PYSER, 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 Info and 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
- TS-SRD@telit.com (for Short Range Devices)

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 3G Modules AT Commands Reference Guide, 80378ST10091A
[3] Telit Easy Script Python, 80000ST10020a
[4] HE910 Hardware User Guide, 1vv0300925
[5] UE910 Hardware User Guide, 1vv0301012
[6] UL865 Hardware User Guide, 1vv0301050
[7] Telit's AppZone User Guide, 1vv0301082
[8] AppZone APIs User Guide, 1vv0301130
[9] UE866 Hardware User Guide, 1vv0301157
[10] UE910 3G Hardware User Guide, 1vv0301171
[11] UL865 3G Hardware User Guide, 1vv0301114

# 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 (VSD).

VSD is a software 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>USB (USB0÷USB6)[1]<br>SPI<br>HSIC | AT0, AT1, AT2, AT3<br>TT<br>DLink<br>VHWDTE0<br>VHWDTE1<br>PYSER, PYSER2, PYUSB0<br>Python Debugging<br>GPS | GPS<br>Python<br>AppZone | CMUX (VC1÷VC4)[2] |

*Tab.1: Items Connected by means of the VSD*

NOTICE:
in documents [4], [5], [6]. [9], [10], and [11] USIF0 and USIF1 are called respectively Modem serial Port1 and Modem serial Port 2.

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

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.

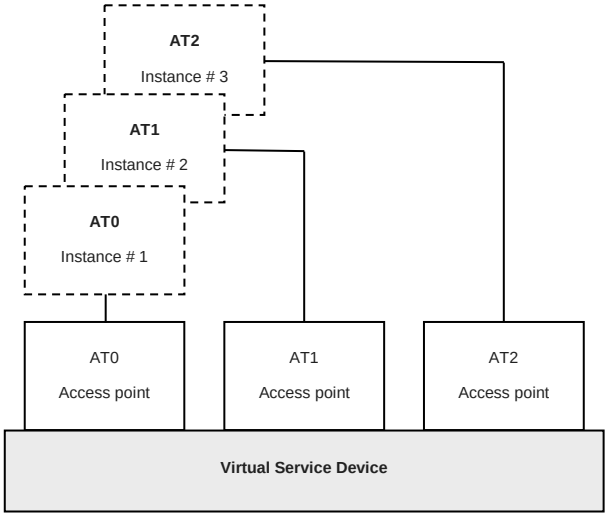



*Fig.1: AT Parser Instances*

# 3. PRELIMINARY INFORMATION

It is strongly recommended to use the AT#USBCFG=? and AT#PORTCFG=? 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 ports configuration, and the USB cable is plugged in. Fig.2 shows the USBx ←→ COMx ports mapping. The mapping depends on the Windows-PC configuration, in this case USB0 port is mapped into COM9 port.

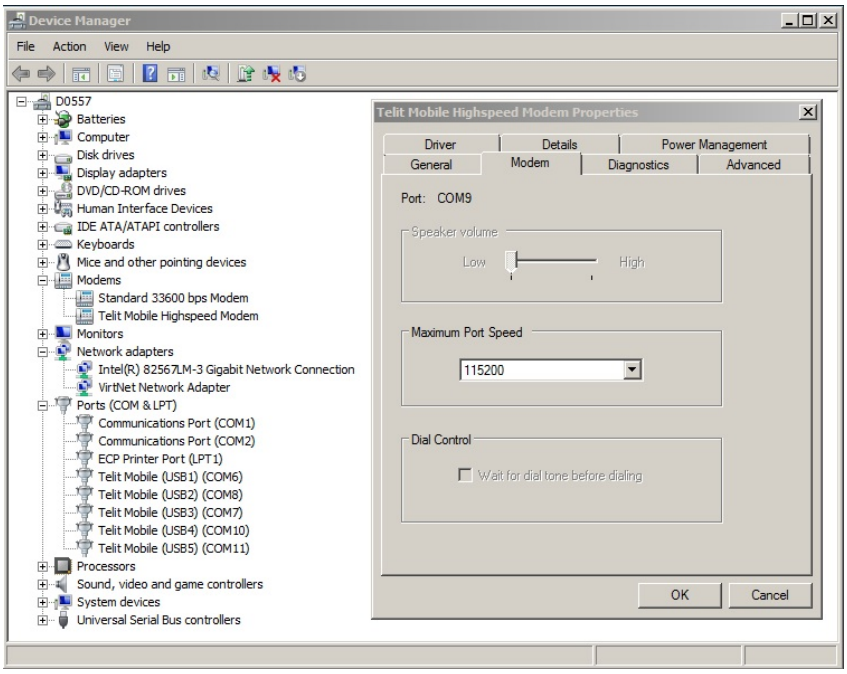



*Fig.2: USBx Mapped into Virtual COMx*

Tab.2 summarizes the mapping shown in Fig.2

| USBx Ports | COMx Ports |
|---|---|
| USB0 | COM9 |
| USB1 | COM6 |
| USB2 | COM8 |
| USB3 | COM7 |
| USB4 | COM10 |
| USB5 | COM11 |

*Tab.2: Mapping Table*

Enter the AT#USBCFG=? Test command to check the number of USB ports 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 | USB Ports Configurations | | | | Description | PID |
|---|---|---|---|---|---|---|
| | ACM | SS | ECM | DLINK | | |
| 0 (factory-setting) | ✓ | standard | | | All USBx ports support the ACM subclass, and Selective Suspend standard type. | 0X21 |
| 1 | Data only | standard | | ✓ | All USBx ports support the ACM subclass in Data only mode, Selective Suspend standard type, and DLink feature. | 0x26 |
| 2 | ✓ | standard | | ✓ | All USBx ports support the ACM subclass, Selective Suspend standard type, and DLink feature. | 0x21 |
| 3 | ✓ | standard | ✓ | | All USBx ports support the ACM subclass, Selective Suspend standard type, and ECM protocol. | 0x23 |
| 4 | ✓ | custom | | | All USBx ports support the ACM subclass, and Selective Suspend custom type. | 0x24 |
| 5 | ✓ | custom | ✓ | | All USBx ports implement the ACM subclass, Selective Suspend custom type, and ECM protocol. | 0x25 |

*Tab. 3: #USBCFG Modes*

Telit provides the USB driver to install on the PC-Windows (DTE). With Ubuntu operating system, the attached USB device works with in-box drivers. See chapter 10.3., and 10.3.1. Here is a brief description of ACM, SS, ECM, DLINK features provided by the different USB ports configuration modes.

- ACM (Abstract Control Model)

The USB device (module) implements the Abstract Control Model (ACM) subclass defined in the USB Communication Device Class (CDC) specification. The ACM subclass defines two interfaces, 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.4, and chapter 10.2.

| 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.4: Enpoints Configuration for ACM*

The USB host (PC) reads the current USB device configuration mode using the Device Control Interface EP0, see table below, 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.5: Enpoints Configuration for device Control Interface*

- ACM/Data only

The USB device (module) implements the simplified ACM/DATA only protocol (Data interface: bulk in, and bulk out endpoint); see table below, and 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".

- ECM (Ethernet Control Model)

ECM is an Ethernet emulation protocol defined by the USB Implementers Forum.

See the Ethernet Control Mode (ECM) AT commands, document [2].

- DLink

See Data Link AT command AT#DLINK, document [2]
See the following chapters 4.13., and 5.3.

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 “dmsg” 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 Variant value. The command returns a short description on the logical connection for each Variant value provided by the module. Here are all the possible configurations, regardless the module you are using.

```
AT#PORTCFG=?
#PORTCFG: Variant=0:   AT= USIF0  USB0  USB3;          MA(Trace)= USB1
#PORTCFG: Variant=1:   AT= USIF0  USB0  USB3;          MA(Trace)= USIF1
#PORTCFG: Variant=2:   AT= USIF0  SPI   USB0;          MA(Trace)= USB1
#PORTCFG: Variant=3:   AT= USIF0  USIF1 USB0;          MA(Trace)= USB1
#PORTCFG: Variant=4:   AT= USIF0  SPI   USB0  USB3;    MA(Trace)= USB1
#PORTCFG: Variant=5:   AT= SPI    USB0  USB3;          MA(Trace)= USB1
#PORTCFG: Variant=6:   AT= USIF0  SPI   USB0;          MA(Trace)= USB1
#PORTCFG: Variant=7:   AT= USIF0  USB0  USB3;          MA(Trace)= USB1;  3G-Trace= USB2
#PORTCFG: Variant=8:   AT= USB0   USB3  USB4;          MA(Trace)= USB1
#PORTCFG: Variant=9:   AT= USIF0  USB0  HSIC0;         MA(Trace)= USB1;  3G-Trace= USB2
#PORTCFG: Variant=10:  AT= USIF0  USB0  HSIC0  HSIC1;MA(Trace)= USB1;  3G-Trace= USB2
#PORTCFG: Variant=11:  AT= USIF0  USB3  USB0;          MA(Trace)= USB1;  ExtGNSS= USIF1
#PORTCFG: Variant=12:  AT= USIF0  USB0  USB3  USB4;    MA(Trace)= USB1;  DLINK= USB5
```

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=1:

- 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.
- USIF1 is connected to the TT Service Access Point.

Variant=2, 3...

- And so on for each Variant value.

The ports arrangement that you set, can be used with one of the six #USBCFG modes. The following table emphasizes that the #PORTCFG assigns a Service Access Point (SAP1, SAP...) to one USBx port, and #USBCFG assigns a "mode" to all USBx ports. Each mode defines a

product (USB device) by means of the PID, each product provides different features described in Tab. 3.

| USBx Ports | AT#PORTCFG=X | | | | | Modes (#USBCFG) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| | SAP1 | SAP2 | SAP3 | SAP... | SAP... | PIDs | | | | | |
| USB0 | | | | | | 0x21 | 0x26 | 0x21 | 0x23 | 0x24 | 0x25 |
| USB1 | | ✓ | | | ✓ | | | | | | |
| USB2 | | | | | | | | | | | |
| USB3 | | | | ✓ | | | | | | | |
| USB4 | | | | | | | | | | | |
| USB5 | | | | | | | | | | | |
| USB6 | NA | | | | | | | | | | |

*Tab.8: #PORTCFG & #USBCFG*

# 4. AT#PORTCFG COMMAND

The AT#PORTCFG command manages several internal ports arrangements by means of its parameter value called ‘Variant’, refer to document [2]. The tables and figures illustrated on the next pages show the various ports configurations obtained changing the "Variant" value of the command and plugging in/unplugging the USB cable.

Follow this sequence to make active the entered AT#PORTCFG command:

- Start from the configuration shown in Tab.11, it is the factory setting: #PORTCFG is 1;
- Enter, for example, the AT#PORTCFG=0 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=0 command is executed and the ports/services arrangement of Tab.9 is set. To power off/on the module, you can use the AT#REBOOT command.

The DTE shown in the next examples is a Windows-PC and does not provide the SPI and HSIC interfaces, the figures show SPI and HSIC ports in the same box, but they are two distinct entities.

Referring to the next tables, the use of the GPS access point is described in the chapters dedicated to the Build-in and External GPS receivers. See chapter 6.1.

In addition, the tables show the use of three types of trace tools in accordance with the selected AT#PORTCFG configuration:

- MA (tool for internal use only) or TTC (Telit Trace Client tool) must be connected to the same physical port as shown by the AT#PORTCFG=? Test command. Both tools collect the protocol binary file.
- 3G tool (for internal use only) must be connected to the physical serial ports as shown by the AT#PORTCFG=? Test command. It collects the binary file of the low-level protocol (layer 1 protocol).

MA and 3G tools are closed in round brackets. To have more information contact Telit Technical Support.

## 4.1 AT#PORTCFG=0

| AT#PORTCFG=0 | | | | | |
|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **TT** | **GPS** |
| **No USB cable** | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| **USBHSI0** | | | | | |
| **USBHSI1** | | | | | |
| **USBHSI2** | | | | | |
| **USBHSI3** | | | | | |
| **USIF0** | **X** | | | | |
| **USIF1** | | | | | |
| **SPI** | | | | | |

*Tab.9: #PORTCFG=0, no USB Cable*

| AT#PORTCFG=0 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.10: #PORTCFG=0, with USB Cable*

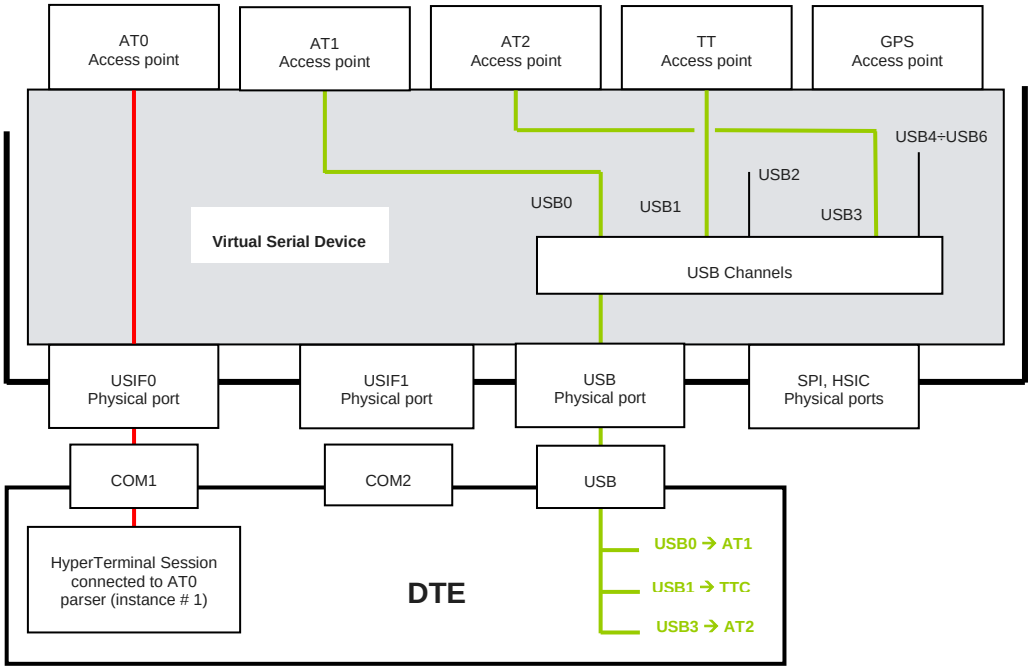



*Fig.3: #PORTCFG=0 + USB Cable*

## 4.2 AT#PORTCFG=1

| AT#PORTCFG=1 (factory-setting) | | | | | |
|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **TT** | **GPS** |
| **No USB cable** | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| **USBHSI0** | | | | | |
| **USBHSI1** | | | | | |
| **USBHSI2** | | | | | |
| **USBHSI3** | | | | | |
| **USIF0** | X | | | | |
| **USIF1** | | | | TTC (or MA) | |
| **SPI** | | | | | |

*Tab.11:#PORTCFG=1, no USB Cable*

| AT#PORTCFG=1 (factory-setting) | | | | | |
|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **TT** | **GPS** |
| **USB0** | | X | | | |
| **USB1** | | | | | |
| **USB2** | | | | | |
| **USB3** | | | X | | |
| **USB4** | | | | | |
| **USB5** | | | | | |
| **USB6** | NA | | | | |
| **USBHSI0** | | | | | |
| **USBHSI1** | | | | | |
| **USBHSI2** | | | | | |
| **USBHSI3** | | | | | |
| **USIF0** | X | | | | |
| **USIF1** | | | | TTC (or MA) | |
| **SPI** | | | | | |

*Tab.12: #PORTCFG=1, with USB Cable*

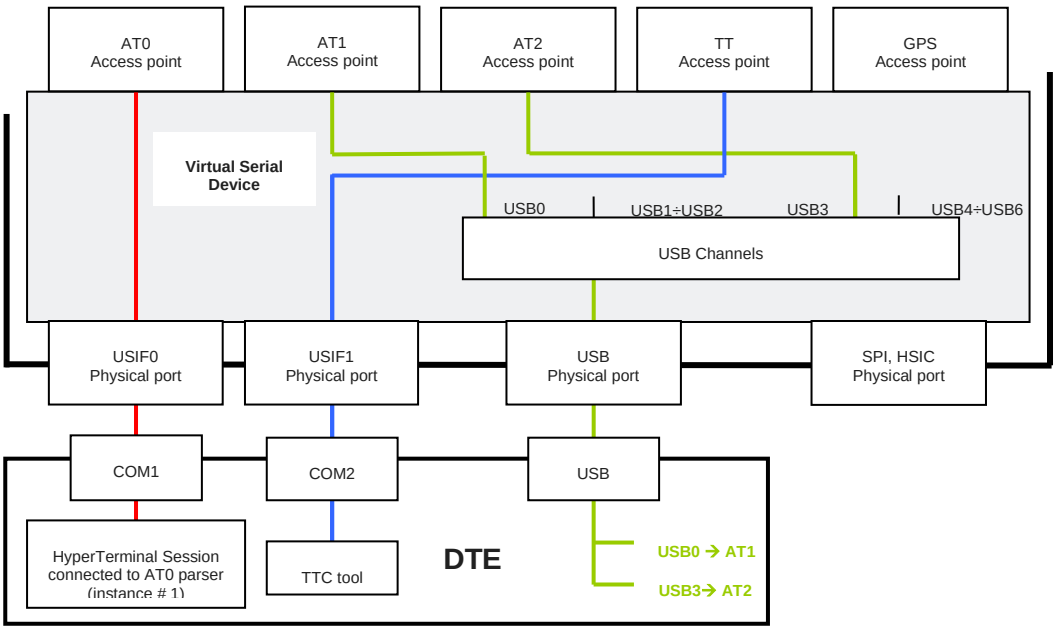



*Fig.4: #PORTCFG=1 + USB Cable*

## 4.3 AT#PORTCFG=2

| AT#PORTCFG=2 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | X | | |

*Tab.13: #PORTCFG=2, no USB Cable*

| AT#PORTCFG=2 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | X | | |

*Tab.14: #PORTCFG=2, with USB Cable*

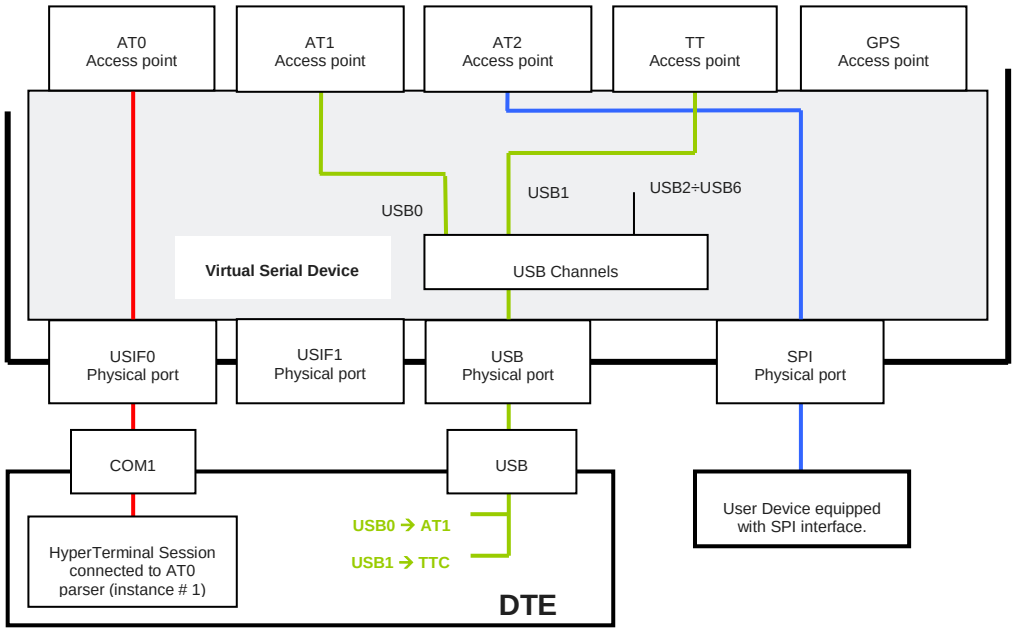



*Fig.5: #PORTCFG=2 + USB Cable*

## 4.4 AT#PORTCFG=3

| AT#PORTCFG=3 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | X | | |
| SPI | | | | | |

*Tab.15: #PORTCFG=3, no USB Cable*

| AT#PORTCFG=3 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | X | | |
| SPI | | | | | |

*Tab.16:#PORTCFG=3, with USB Cable*

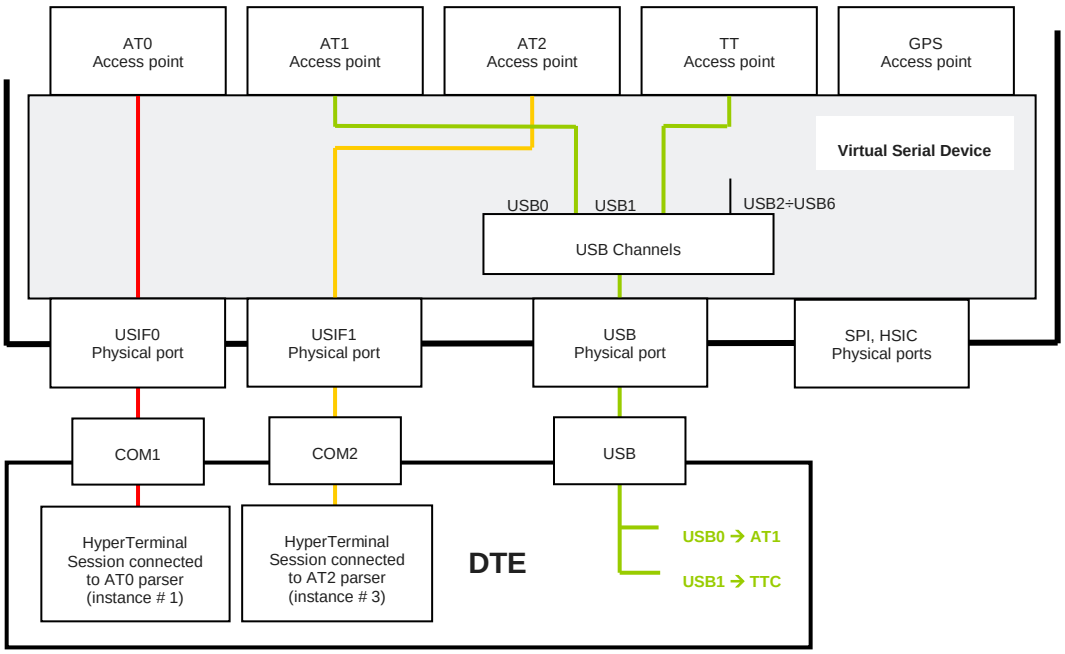



*Fig.6: #PORTCFG=3 + USB Cable*

## 4.5 AT#PORTCFG=4

| AT#PORTCFG=4 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | X | | | |
| USIF1 | | | | | |
| SPI | | | X | | |

*Tab.17: #PORTCFG=4, no USB Cable*

| AT#PORTCFG=4 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | X | | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | X | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.18: #PORTCFG=4, with USB Cable*

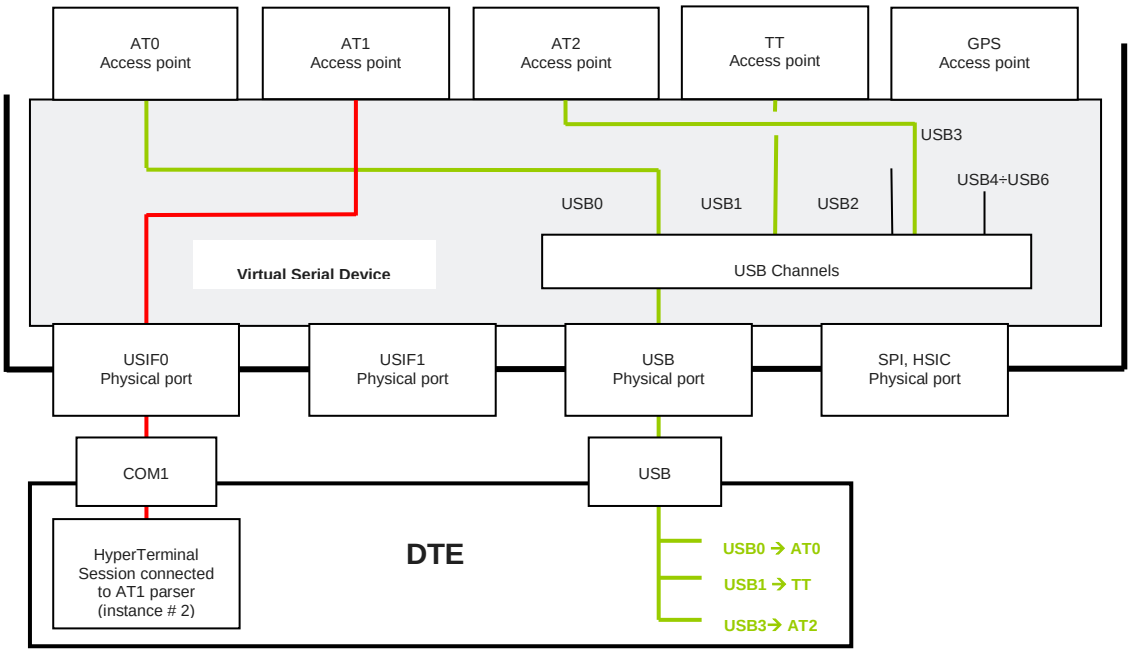



*Fig.7: #PORTCFG=4 + USB Cable*

## 4.6 AT#PORTCFG=5

| AT#PORTCFG=5 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | X | | |

*Tab.19: #PORTCFG=5, no USB Cable*

| AT#PORTCFG=5 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | X | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | X | | |

*Tab.20: #PORTCFG=5, with USB Cable*

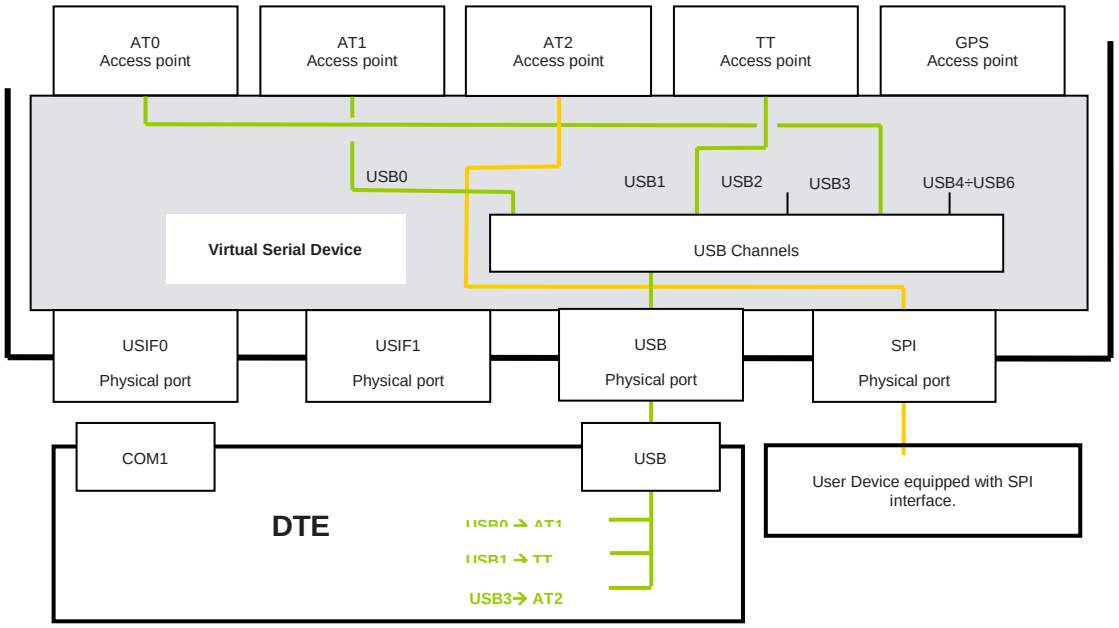



*Fig.8: #PORTCFG=5 + USB Cable*

## 4.7 AT#PORTCFG=6

| AT#PORTCFG=6 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | X | | |
| USIF1 | | | | | |
| SPI | X | | | | |

*Tab.21: #PORTCFG=6, no USB Cable*

| AT#PORTCFG=6 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | X | | |
| USIF1 | | | | | |
| SPI | X | | | | |

*Tab.22: #PORTCFG=6, with USB Cable*

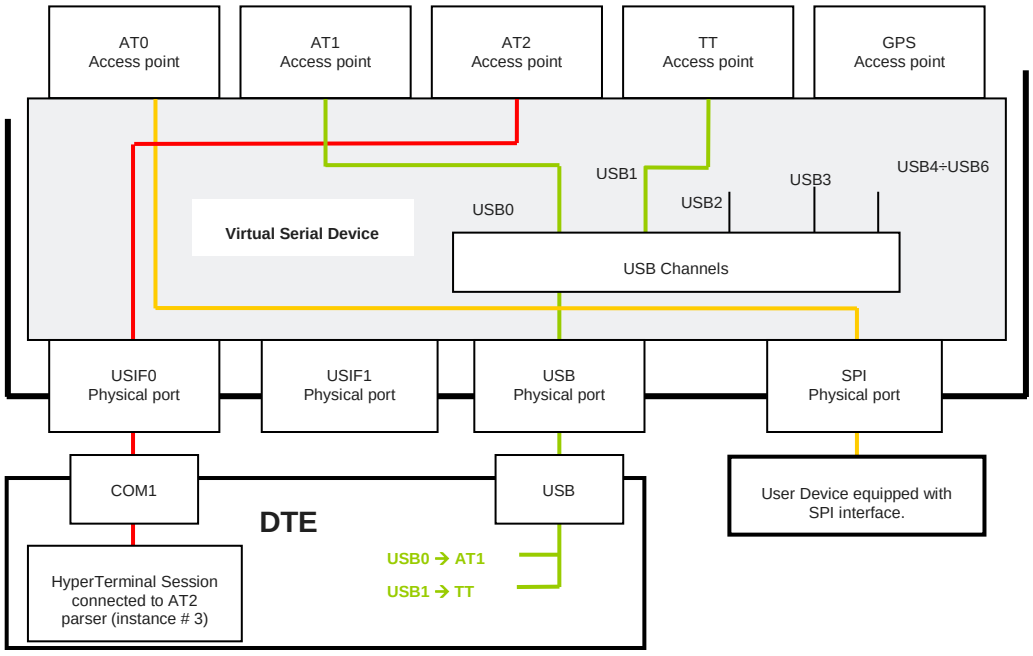



*Fig.9: #PORTCFG=6 + USB Cable*

## 4.8 AT#PORTCFG=7

| AT#PORTCFG=7 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.23: #PORTCFG=7, no USB Cable*

| AT#PORTCFG=7 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | ( 3G ) | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.24: #PORTCFG=7, with USB Cable*

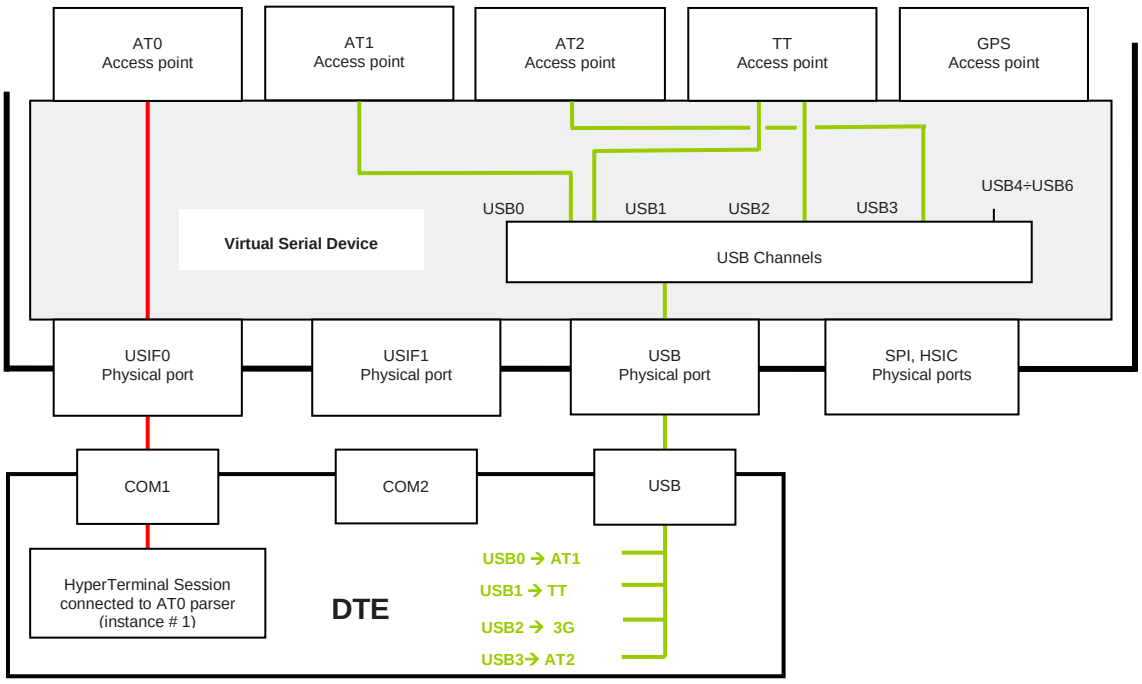



*Fig.10: #PORTCFG=7 + USB Cable*

## 4.9 AT#PORTCFG=8

| AT#PORTCFG=8 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| NO USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.25: #PORTCFG=8, no USB Cable*

| AT#PORTCFG=8 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | X | | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | X | | | |
| USB4 | | | X | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.26: #PORTCFG=8, with USB Cable*

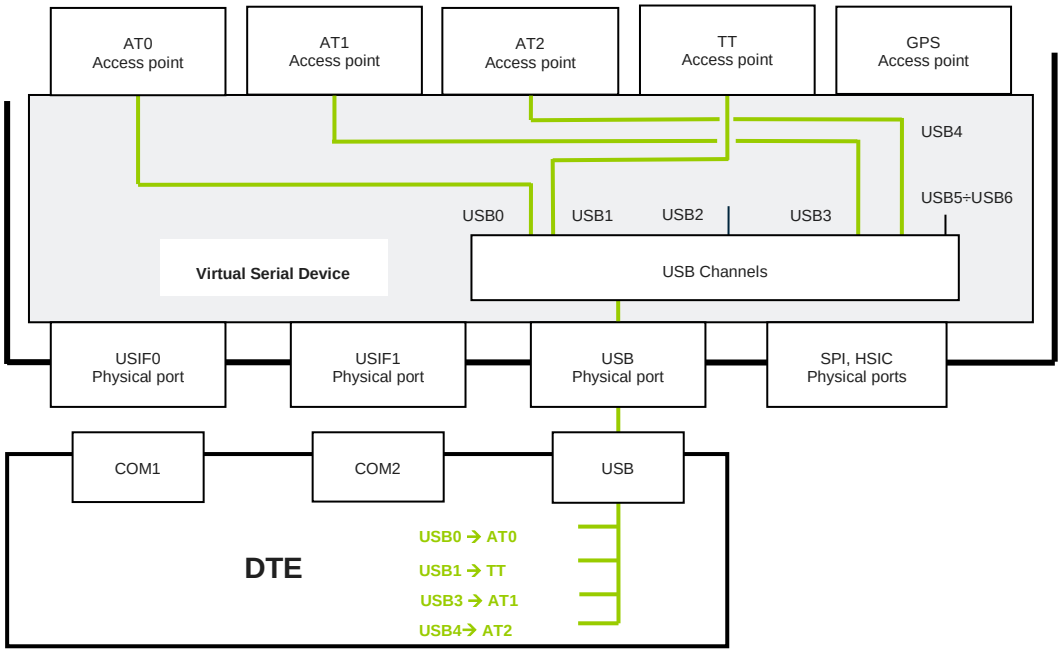



*Fig.11: #PORTCFG=8 USB Cable Only*

## 4.10 AT#PORTCFG=9

| AT#PORTCFG=9 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | X | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.27: #PORTCFG=9, no USB Cable*

| AT#PORTCFG=9 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | ( 3G ) | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | X | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.28: #PORTCFG=9, with USB Cable*

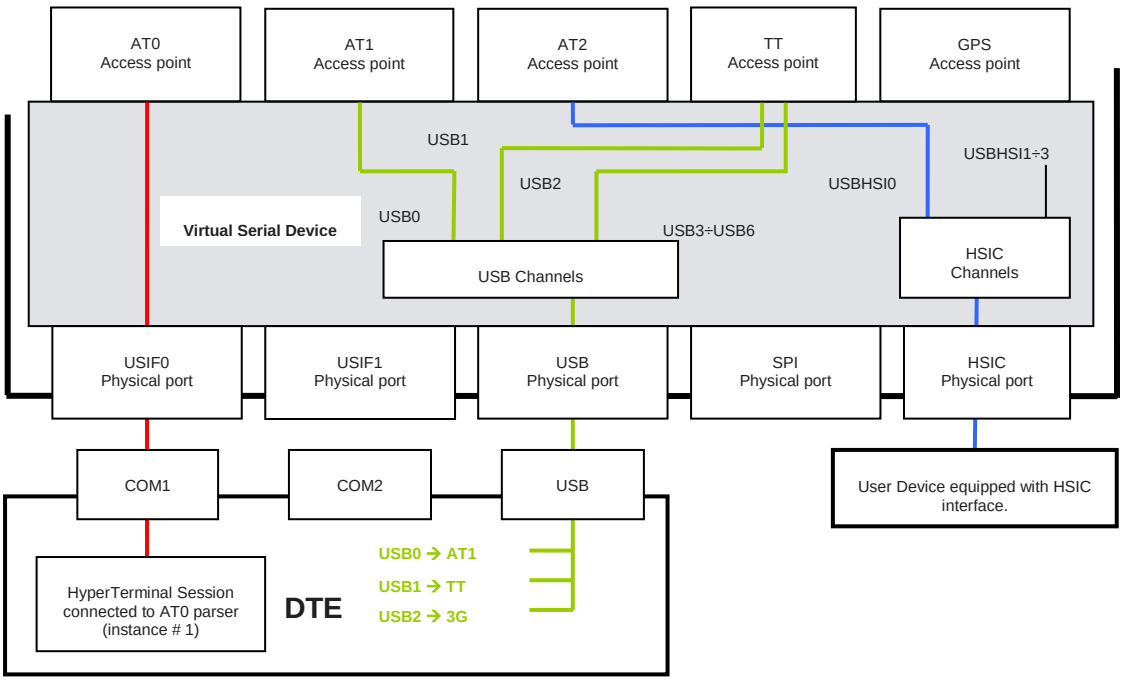



*Fig.12: #PORTCFG=9 + USB Cable*

## 4.11 AT#PORTCFG=10

| AT#PORTCFG=10 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | X | | | | |
| USBHSI1 | | X | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | X | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.29: #PORTCFG=10, no USB Cable*

| AT#PORTCFG=10 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | | X | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | ( 3G ) | |
| USB3 | | | | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | X | | | | |
| USBHSI1 | | X | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.30: #PORTCFG=10, with USB Cable*

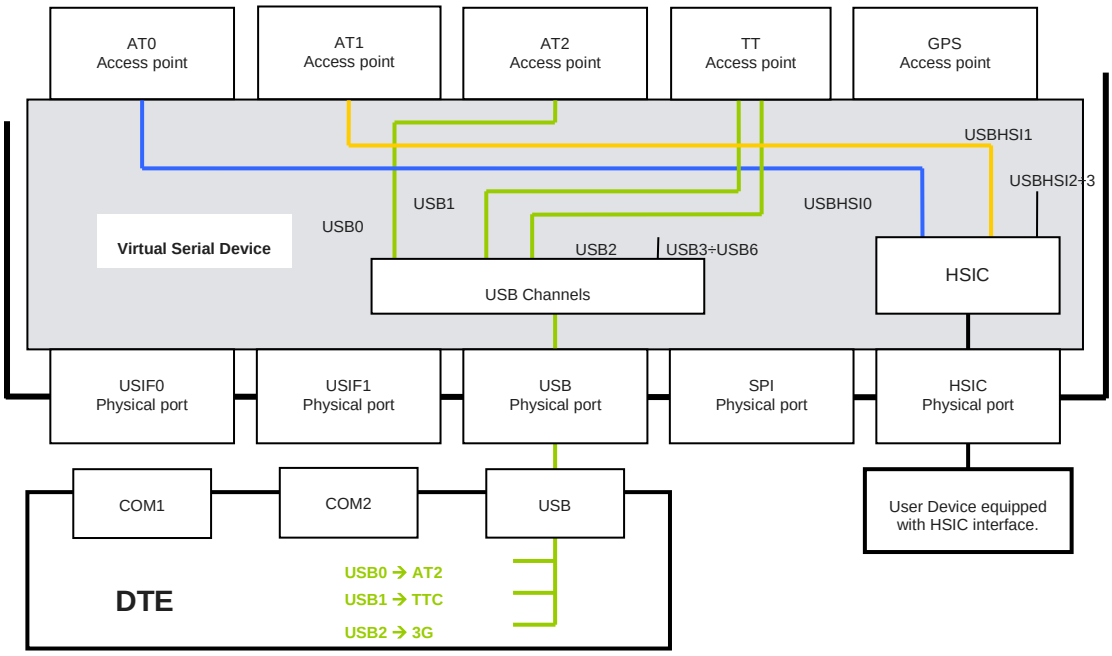



*Fig.13: #PORTCFG=10 + USB Cable*

## 4.12 AT#PORTCFGCFG=11

NOTICE:

the modules that are not equipped with an embedded GPS receiver, support Variant value equal to 11. It is used to connect the module to an external GPS receiver through USIF1 serial port, see chapter 6.1.2.

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.31: #PORTCFG=11, no USB Cable*

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | |
| USB1 | | | | TTC (or MA) | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | NA | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.32: #PORTCFG=11, with USB Cable*

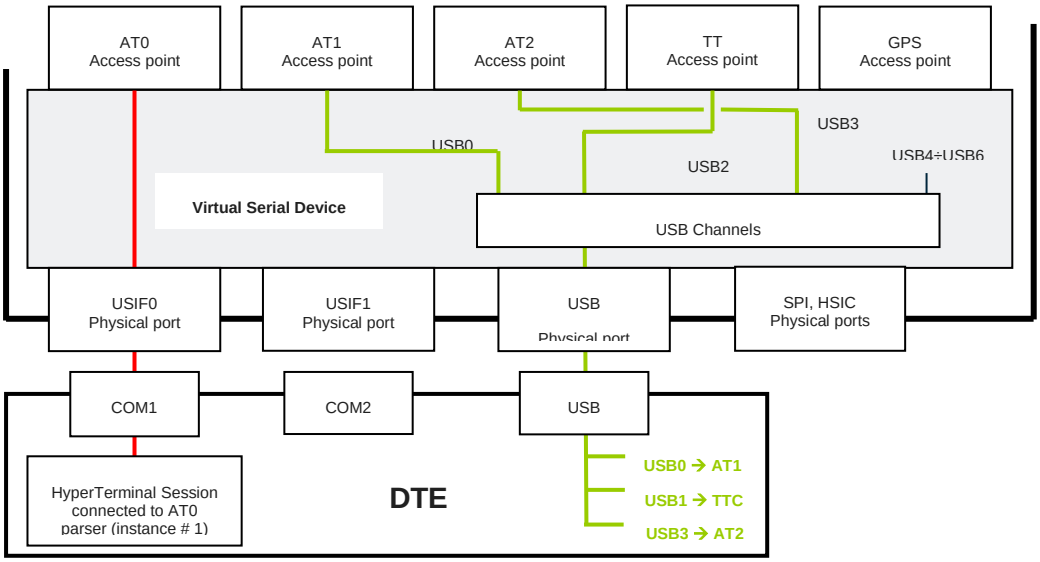



*Fig.14 : #PORTCFG=11 + USB Cable*

## 4.13 AT#PORTCFGCFG=12

AT#PORTCFG=12 provides two new access points: AT3, and DLink

| AT#PORTCFG=12 | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **AT3** | **TT** | **GPS** | **DLink** |
| **No USB cable** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| **USBHSI0** | | | | | | | |
| **USBHSI1** | | | | | | | |
| **USBHSI2** | | | | | | | |
| **USBHSI3** | | | | | | | |
| **USIF0** | X | | | | | | |
| **USIF1** | | | | | | | |
| **SPI** | | | | | | | |

*Tab.33: #PORTCFG=12, no USB Cable*

| AT#PORTCFG=12 | | | | | | | |
|---|---|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | AT3 | TT | GPS | DLink |
| USB0 | | X | | | | | |
| USB1 | | | | | TTC (or MA) | | |
| USB2 | | | | | | | |
| USB3 | | | X | | | | |
| USB4 | | | | X | | | |
| USB5 | | | | | | | X |
| USB6 | NA | | | | | | |
| USBHSI0 | | | | | | | |
| USBHSI1 | | | | | | | |
| USBHSI2 | | | | | | | |
| USBHSI3 | | | | | | | |
| USIF0 | X | | | | | | |
| USIF1 | | | | | | | |
| SPI | | | | | | | |

*Tab.34: #PORTCFG=12, with USB Cable*

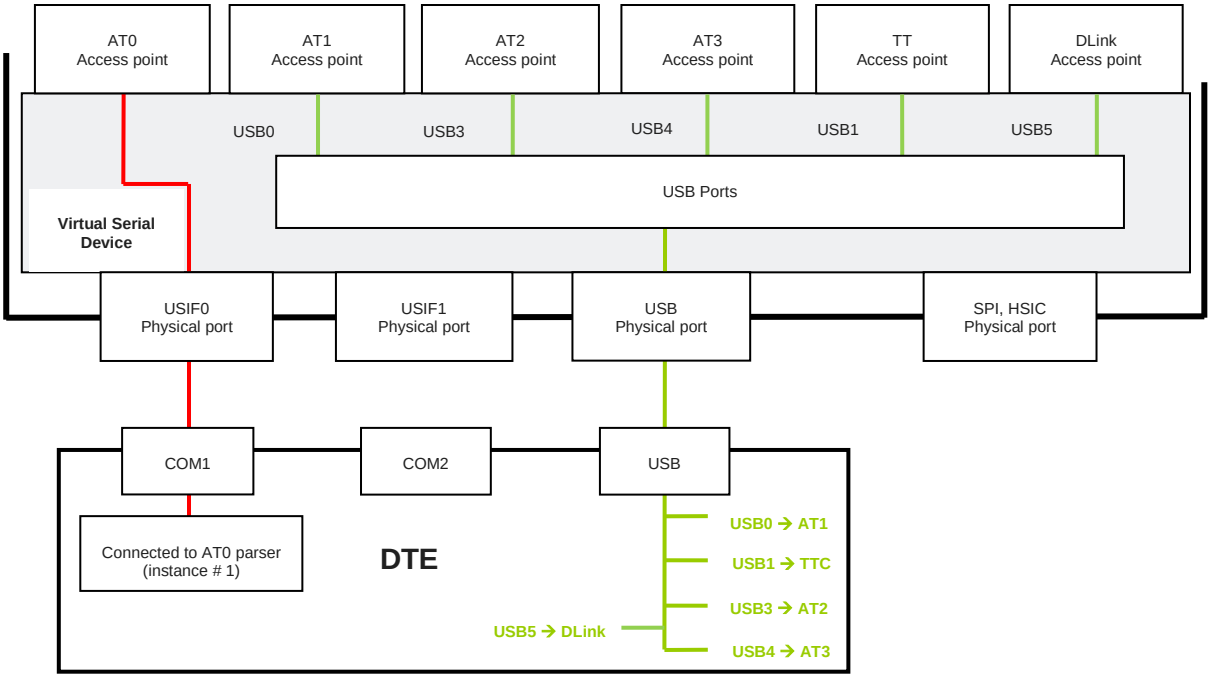



*Fig.15: #PORTCFG=12 + USB Cable*

# 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 based on CMUX protocol on USIF0 serial port.

Assume that the module is configured as indicated in Tab.11: #PORTCFG=1, and no USB cable plugged in. In addition, suppose that the used DTE is a Windows-PC, and Fig.16 shows its device configuration. Now, run on the DTE the Telit Serial Port MUX application configured as shown in Fig.17, and connect the MUX application to COM1 physical port, refer to Fig.18. When the user starts an application (e.g. 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.

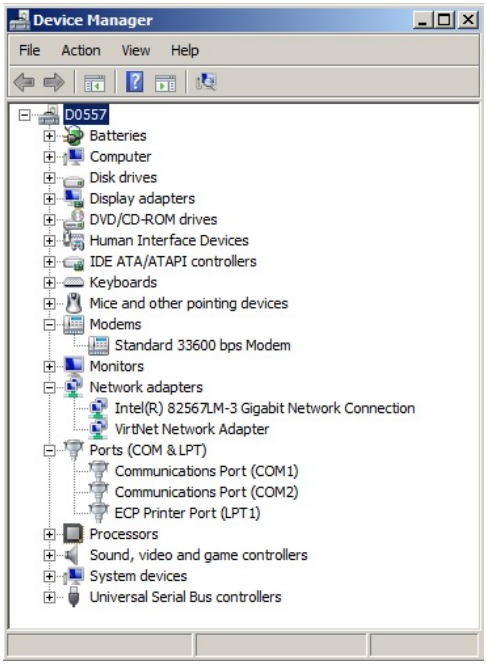



*Fig.16: Physical COMx Ports*

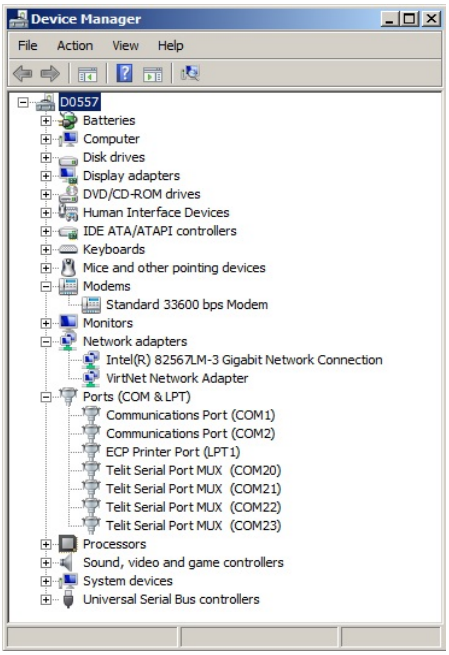



*Fig.17: Virtual Serial Ports of MUX*

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 | GPS |
|---|---|---|---|---|---|---|
| USB not used | | | | | | |
| HSIC not used | | | | | | |
| USIF0 ←→COM1 | COM20→VC1 | X | | | | |
| | COM21→VC2 | | X | | | |
| | COM22→VC3 | | | X | | |
| | COM23→VC4 | | | | | |
| USIF1 not used | | | | | | |
| SPI not used | | | | | | |

*Tab.35: Ports/Services Arrangement with CMUX Connected to USIF0*

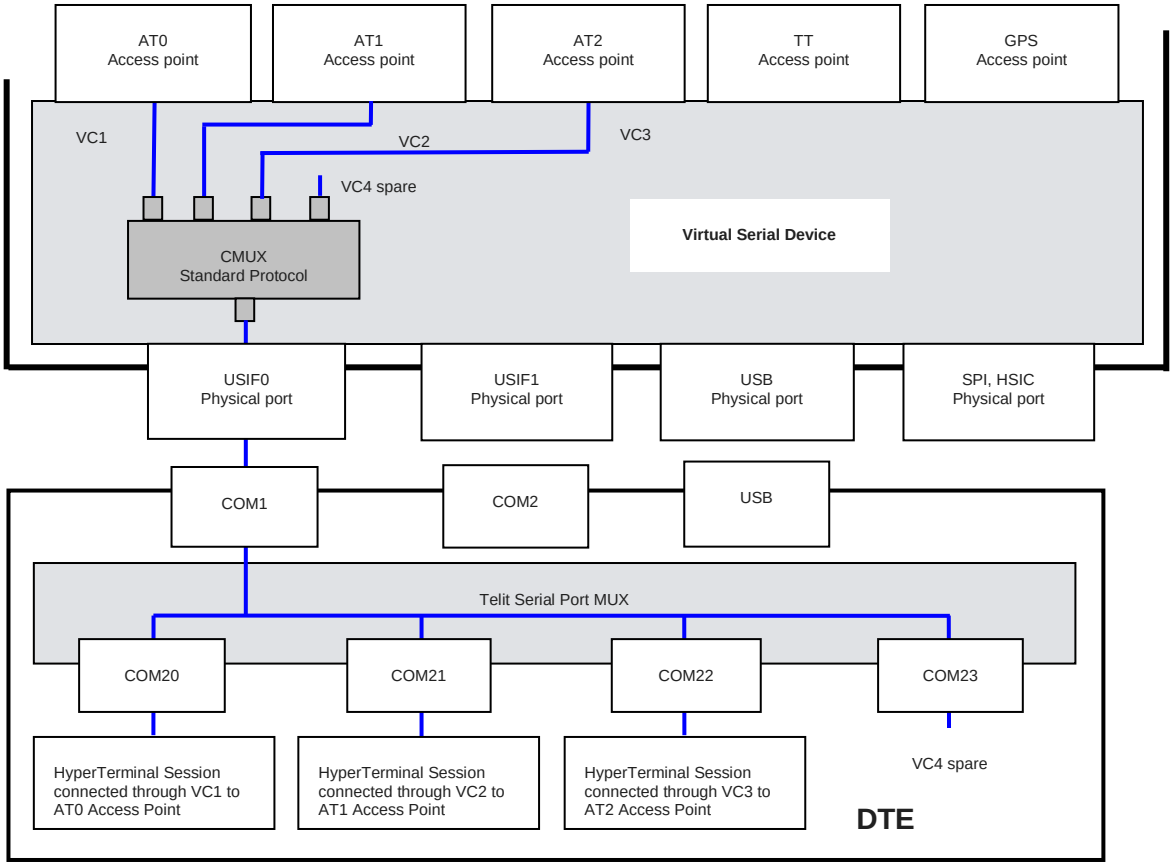



*Fig.18: CMUX Connected to USIF0*

### 5.1.1 Connection with TTC Tool

If TTC connection is needed, start from the configuration: #PORTCFG=1, no USB cable, see Tab.11. Follow the steps stated above, and refer to Fig.19. The table below summarizes the new configuration.

| Module ←→DTE connection | COMx → VCx | AT0 | AT1 | AT2 | TT | GPS |
|---|---|---|---|---|---|---|
| USB not used | | | | | | |
| HSIC not used | | | | | | |
| USIF0 ←→ COM1 | COM20→VC1 | X | | | | |
| | COM21→VC2 | | X | | | |
| | COM22→VC3 | | | X | | |
| | COM23→VC4 | | | | | |
| USIF1 ←→ COM2 | | | | | TTC | |
| SPI not used | | | | | | |

*Tab.36: Ports/Services Arrangement with CMUX + TTC*

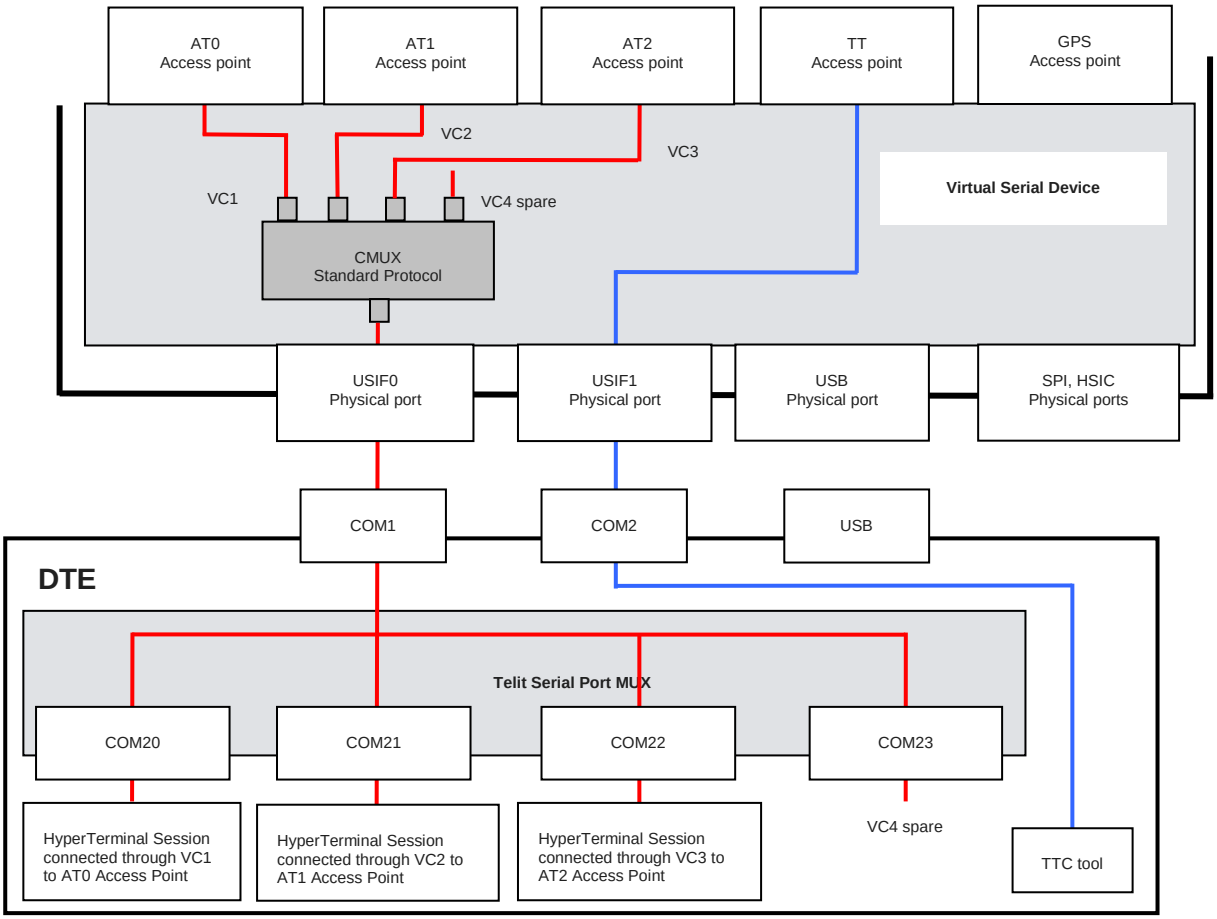



*Fig.19: CMUX Connected to USIF0 + TTC Connected to USIF1*

## 5.2 CMUX Protocol on USB3 Port

Here is an example of ports/services arrangement based on CMUX protocol on USB port.

Assume that the module is configured as indicated in Fig.3: #PORTCFG=0, 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.20, and connect the MUX application to USB3 port mapped into COM7 virtual port. When the user starts an application (e.g. 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, refer to Fig.21.

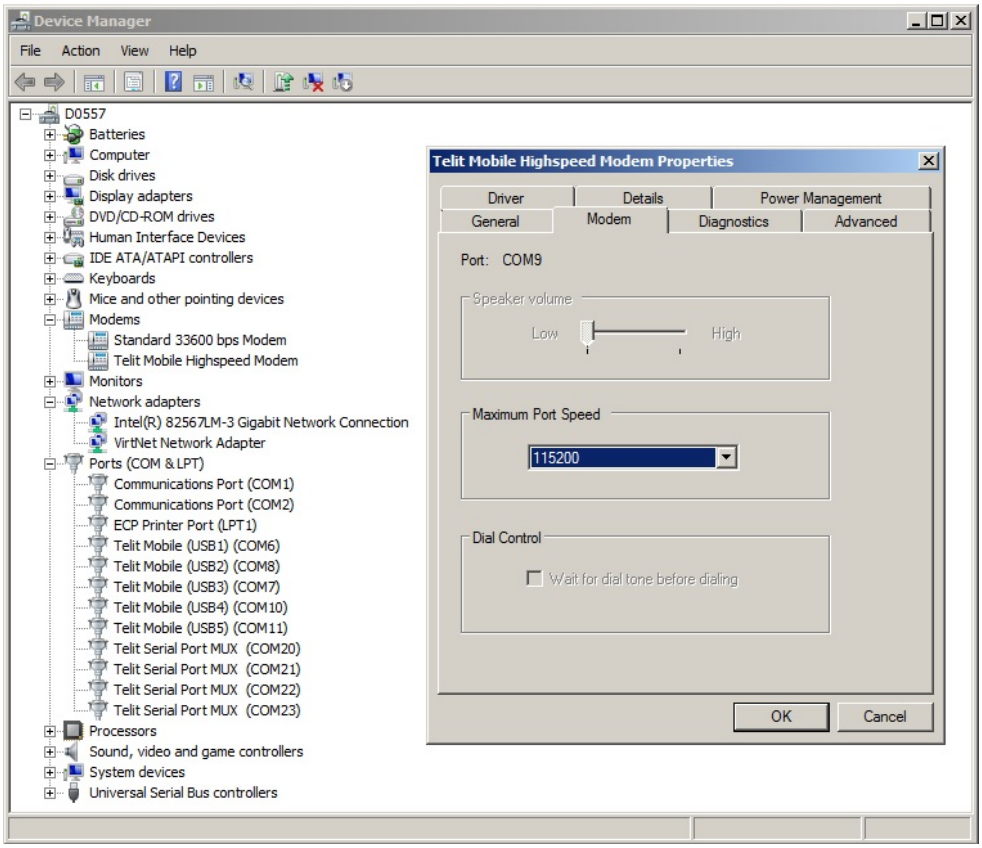



*Fig.20: Virtual Serial Ports of Telit Serial Port MUX*

The table below summarizes the new configuration.

| Module ←→DTE connection | Ports | USBx → COM | COMx → VCx | AT0 | AT1 | AT2 | TT | GPS |
|---|---|---|---|---|---|---|---|---|
| USB ←→ USB | USB0 | | | | | | | |
| | USB1 | | | | | | TTC | |
| | USB2 | | | | | | | |
| | USB3 | COM8 | COM20 → VC1 | X | | | | |
| | | | COM21 → VC2 | | X | | | |
| | | | COM22 → VC3 | | | X | | |
| | | | COM23 → VC4 | | | | | |
| | USB4 | | | | | | | |
| | USB5 | | | | | | | |
| | USB6 | | | | | | | |
| HSIC not used | | | | | | | | |
| USIF0 not used | | | | | | | | |
| USIF1 not used | | | | | | | | |
| SPI not used | | | | | | | | |

*Tab.37: Ports/Services Arrangement with CMUX Connected to USB3 Port*

NOTICE:

AT0 (instance # 1) is disconnected from USIF0 and connected to VC1/USB3/COM8/COM20, the TTC tool stays on USB1 port.

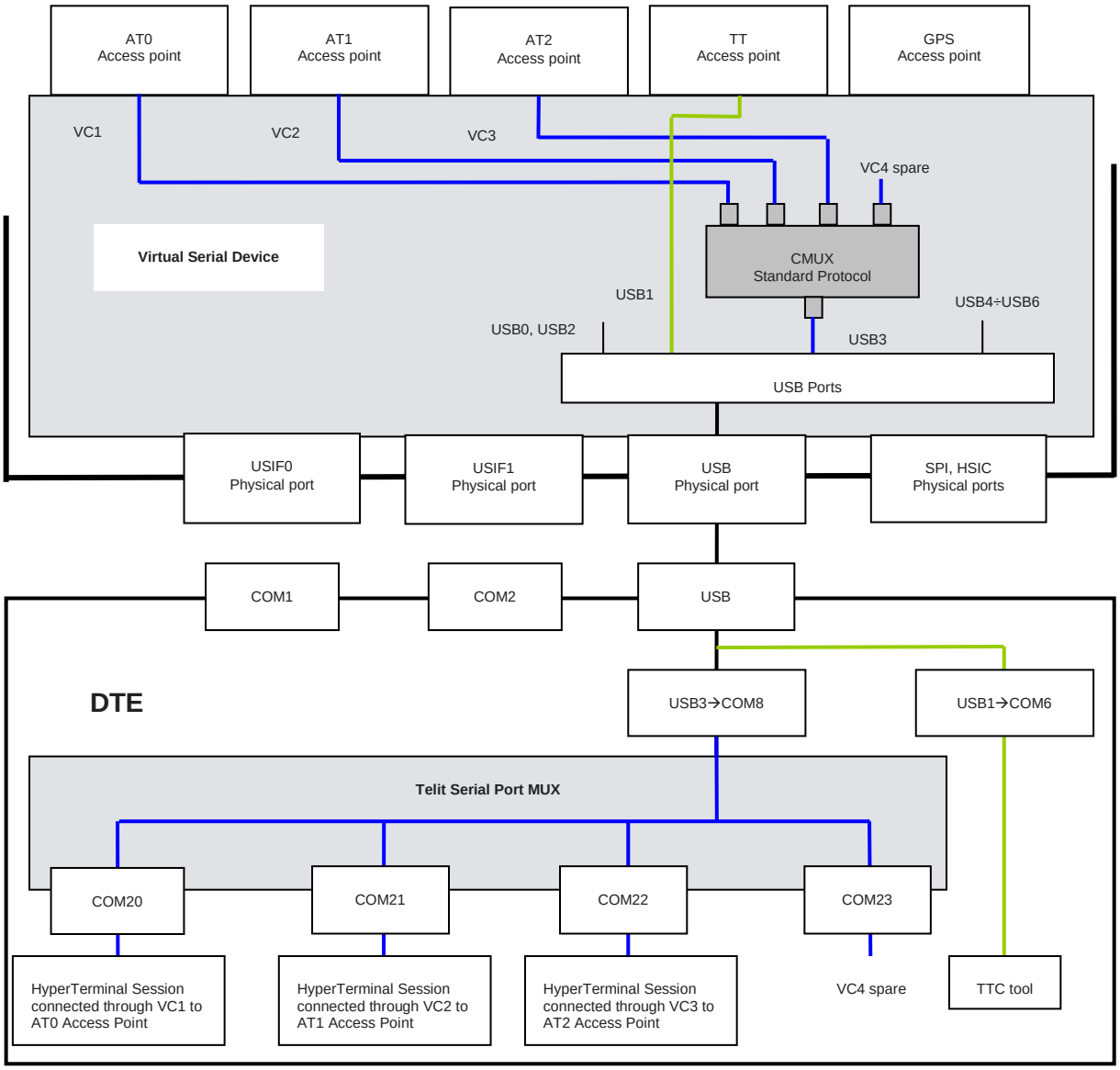



*Fig.21: CMUX Connected to USB3 Port*

## 5.3 CMUX Protocol and AT#PORTCFG=12

The DLink access point permits the logical connection between two user applications (named for example: "A" and "B") running on two different devices connected to the module by means of serial lines, refer to Fig.22.

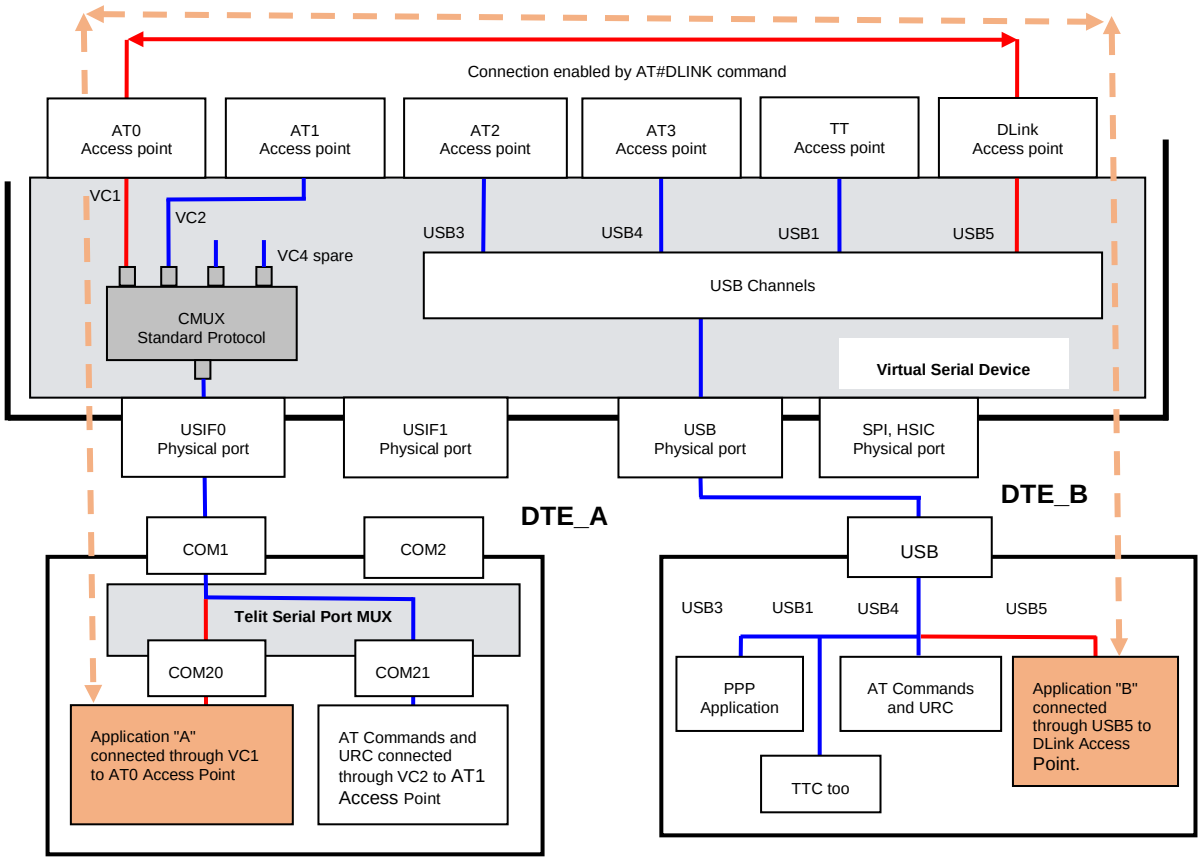



*Fig.22: CMUX & AT#PORTCFG=12*

Window-PC (DTE_A), equipped with the Telit Serial Port MUX tool, is running:

- User application "A"
- User application to send AT commands and receive URC (e.g. Hyper Terminal)

Window-PC (DTE_B) is running:

- User application "B"
- User PPP application
- User application to send AT commands and receive URC (e.g. Hyper Terminal)

The module is configured as indicated in Fig.15 #PORTCFG=12, and USB cable plugged in. The used DTE is a Windows-PC, and Fig.2 shows its device configuration. Run on the DTE the Telit Serial Port MUX application configured as shown in Fig.17, and connect the MUX application to COM1 physical port, refer to Fig.22.

When the user starts an application (e.g. Hyper Terminal) connected to one of the two Virtual Ports (COM20, COM21), Telit Serial Port MUX application sends automatically the AT+CMUX=0 command to the module and the CMUX protocol is activated.

Use the AT#DLINK command to enable the logical internal connection between AT0 and DLink access points. Now, user application "A" is connected to user application "B". Refer to document [2] to have information on the AT#DLINK syntax.

Before using the AT#DLINK command, you must issue the AT#PORTCFG=12 command and one of the two commands: AT#USBCFG=1 or AT#USBCFG=2 in accordance with your needs, see Tab. 3**.**

# 6. SERVICES

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

As stated in chapter 2 different Service Access Points connect the 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 VSD layer.

## 6.1 GPS

The GPS receiver can be “built-in” or external to the module.

### 6.1.1 Built-in GPS Receiver

The built-in GPS receiver can send NMEA sentences on different physical ports, in accordance with the current ports configuration. In general, NMEA sentences run on the physical port used by the operator to enter the AT$GPSP and AT$GPSNMUN commands, in this case AT commands and NMEA sentences share the same physical port at the same time. Refer to document [2] to have information on AT commands syntax.

Here are sub-chapters showing some examples of logical connections settings.

#### 6.1.1.1 AT#PORTCFG=0

Tab.9 shows the starting ports configuration of the module: #PORTCFG=0. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT commands entered through USIF0 port. AT0 parser executes the AT commands, and after that NMEA sentences and AT commands run on USIF0 port as summarized in Tab.38. See also Fig.23.

| AT#PORTCFG=0 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| No USB cable | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | X |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.38: USIF0 port supports NMEA sentences*

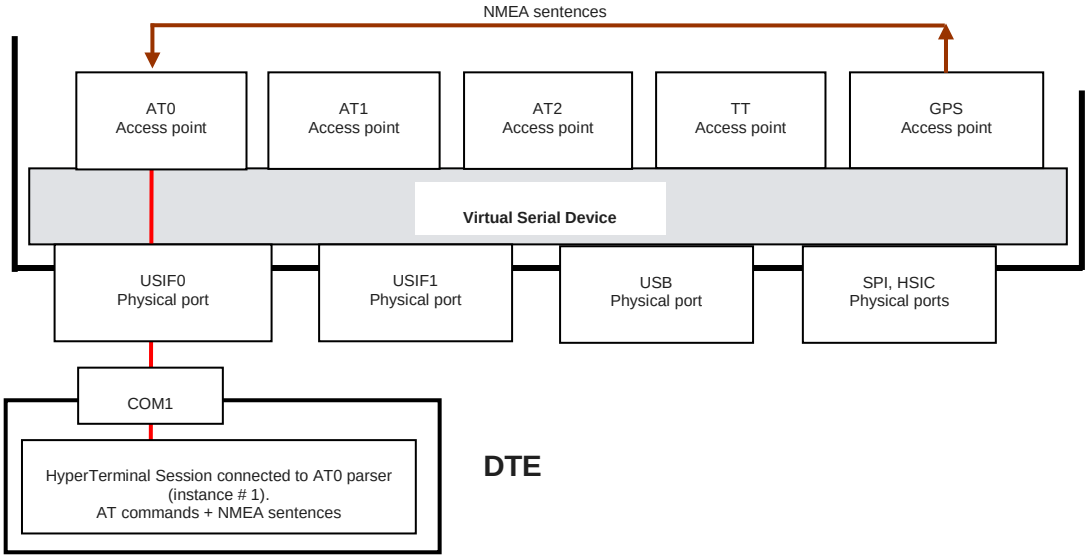



*Fig.23: USIF0 Port Supports AT Commands + NMEA Sentences*

### 6.1.1.2 AT#PORTCFG=0 + USB

Tab.10 shows the starting ports configuration of the module: #PORTCFG=0 + USB. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT 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 summarized and shown in table and figure below.

| AT#PORTCFG=0 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | | X | | | X |
| USB1 | | | | TTC | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab.39: USB0 Supports NMEA Sentences*

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 this example), or USB3 port .

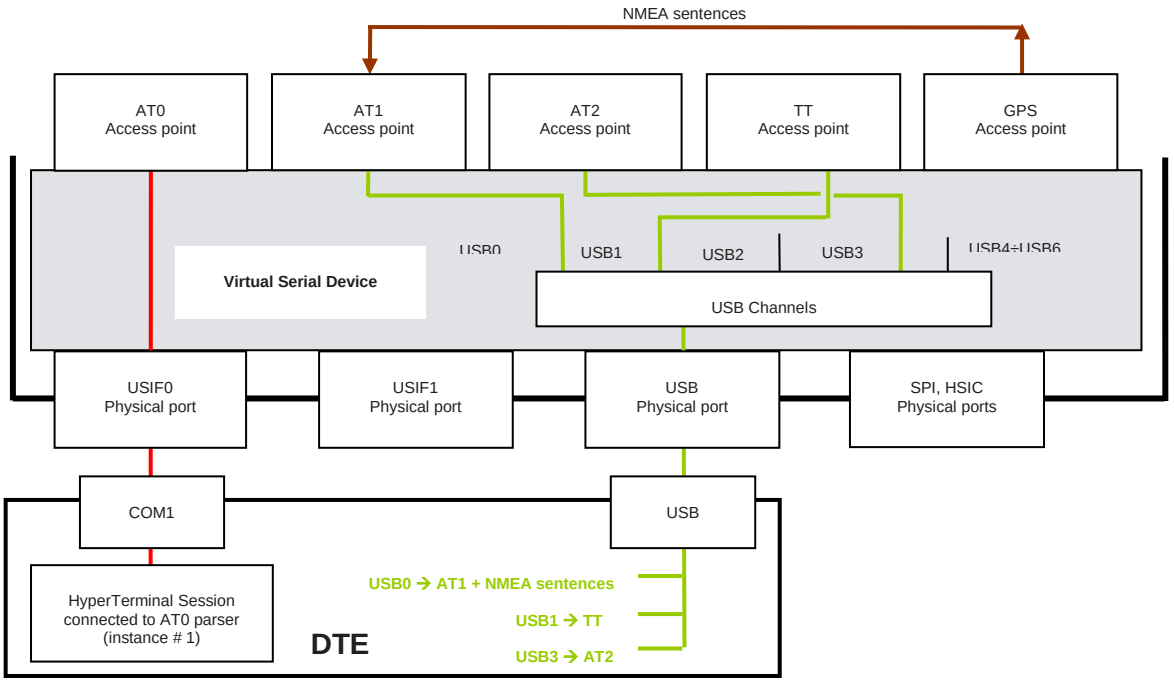



*Fig.24: USB0 Port Supports AT Commands + NMEA Sentences*

### 6.1.1.3 AT#PORTCFG=0 + USB3 + CMUX

Here is an example of ports/services arrangement running NMEA sentences on USB port using CMUX protocol.

Tab.37 shows the starting ports configuration of the module: #PORTCFG=0+USB+CMUX. Now, enable GPS/NMEA sentences via the AT$GPSP=1 and AT$GPSNMUN=1… AT commands entered, for example, through COM22/COM8/USB3/VC3 port. AT2 parser executes the AT commands, and after that NMEA sentences and AT commands run on COM22→VC3 port as summarized in the table below. See also Fig.25.

| Module←→DTE connection | Ports | USBx → COM | COMx → VCx | AT0 | AT1 | AT2 | TT | GPS |
|---|---|---|---|---|---|---|---|---|
| USB ←→ USB | USB0 | | | | | | | |
| | USB1 | | | | | | TTC | |
| | USB2 | | | | | | | |
| | USB3 | COM8 | COM20 → VC1 | X | | | | |
| | | | COM21 → VC2 | | X | | | |
| | | | COM22 → VC3 | | | X | | X |
| | | | COM23 → VC4 | | | | | |
| | USB4 | | | | | | | |
| | USB5 | | | | | | | |
| | USB6 | | | | | | | |
| HSIC not used | | | | | | | | |
| USIF0 not used | | | | | | | | |
| USIF1 not used | | | | | | | | |
| SPI not used | | | | | | | | |

*Tab.40: USB3-VC3 Port Supports AT Commands + NMEA Sentences*

NOTICE:

the user can issue the AT commands through COM20 → VC1/AT0, COM21 → VC2/AT1, or COM22 → VC3/AT2 port/parser. The NMEA sentences are routed respectively on VC1, VC2, or VC3 port. This example works with VC3 port.

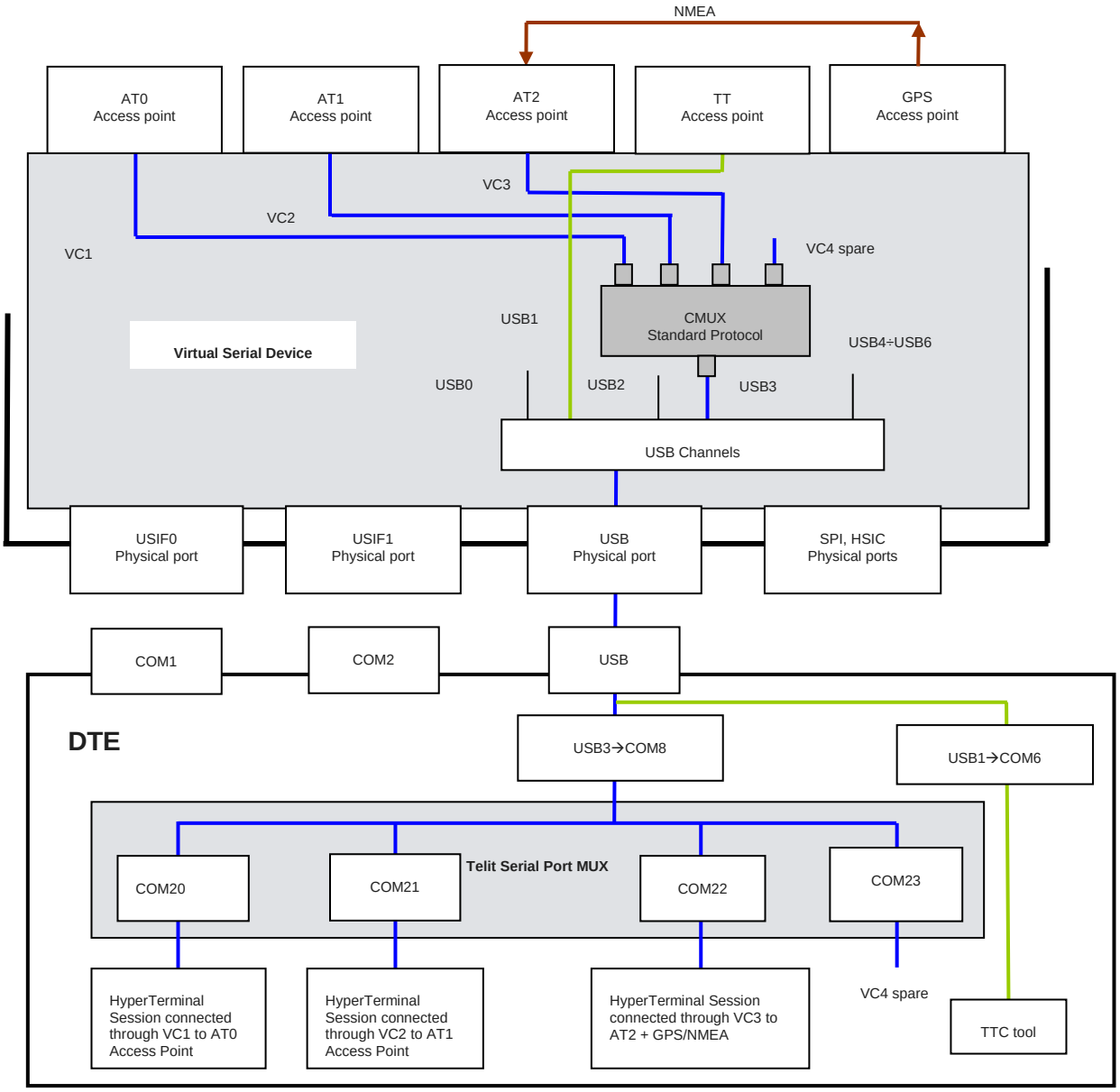



*Fig.25: USB3-VC3 Port Supports AT Commands + NMEA Sentences*

#### 6.1.1.4 AT#PORTCFG=4

Tab.17 shows the starting ports configuration of the module: #PORTCFG=4. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT commands entered, for example, through SPI port. AT2 parser executes the AT commands, and after that NMEA sentences and AT commands run on SPI port as summarized in the table below. See also the Fig.26.

| AT#PORTCFG=4 | | | | | |
|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **TT** | **GPS** |
| **No USB cable** | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| **USBHSI0** | | | | | |
| **USBHSI1** | | | | | |
| **USBHSI2** | | | | | |
| **USBHSI3** | | | | | |
| **USIF0** | | X | | | |
| **USIF1** | | | | | |
| **SPI** | | | X | | X |

*Tab. 41: SPI Port Supports NMEA Sentences*

NOTICE:

the user can issue the AT commands through USIF0/AT1, or SPI/AT2 port/parser. The NMEA sentences are routed respectively on USIF0 or SPI port. This example works with SPI port.

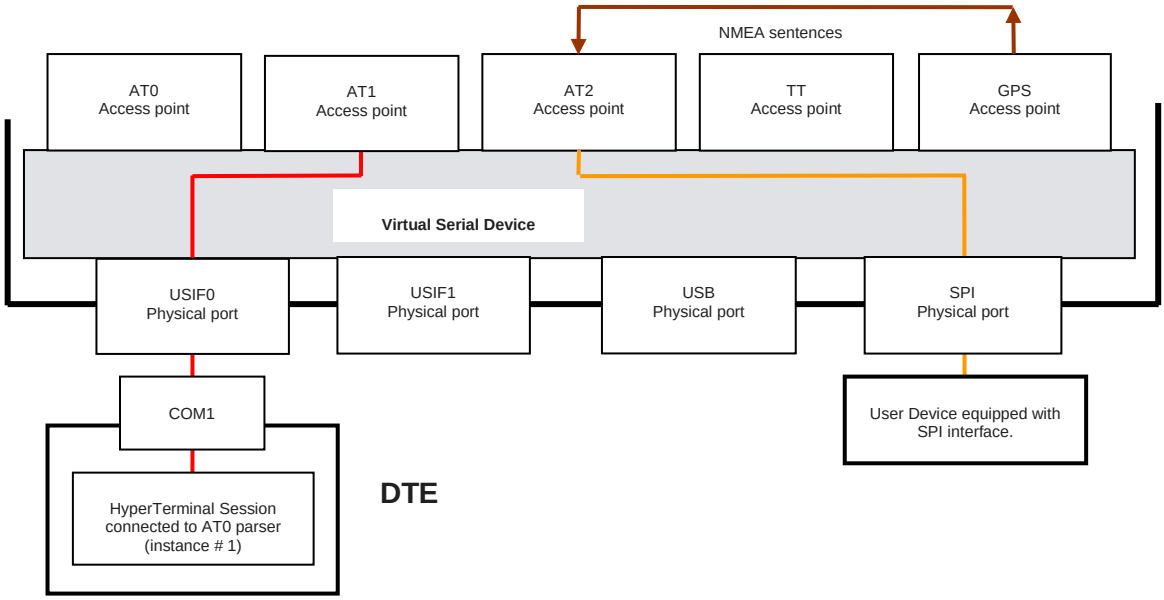



*Fig.26: SPI Port Supports AT Commands + NMEA Sentences*

### 6.1.1.5 AT#PORTCFG=8

Tab.22 shows the starting ports configuration of the module: #PORTCFG=8. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT commands entered, for example, through USB3 port. AT1 parser executes the AT commands, and after that NMEA sentences run on USB5 port as summarized in Tab.38. See also Fig.28

| AT#PORTCFG=8 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | GPS |
| USB0 | X | | | | |
| USB1 | | | | TTC | |
| USB2 | | | | | |
| USB3 | | X | | | |
| USB4 | | | X | | |
| USB5 | | | | | X |
| USB6 | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | | | | | |
| USIF1 | | | | | |
| SPI | | | | | |

*Tab. 1: USB Supports NMEA Sentences*

NOTICE:

the user can issue the AT commands through USB0/AT0, USB3/AT1, and USB4/AT2 ports/parsers. In any case, the GPS/NMEA sentences are routed on USB5 port.

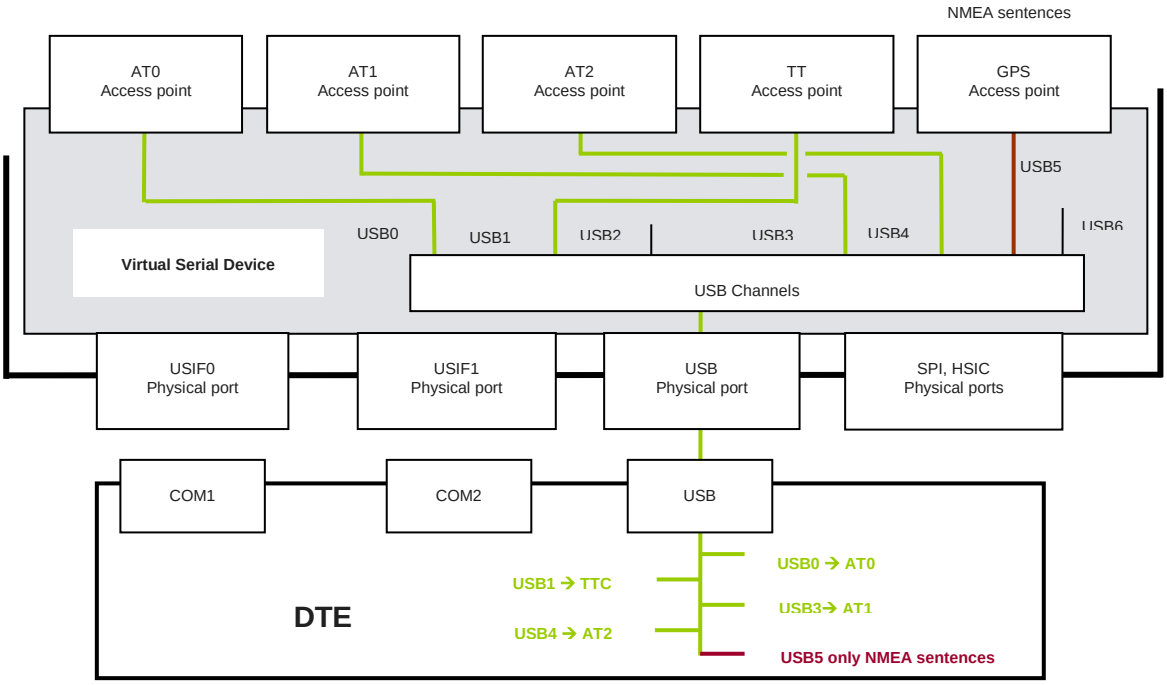



*Fig.27: USB5 Port Supports Only NMEA Sentences*

### 6.1.2. External GPS Receiver

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

#### 6.1.2.1 AT#PORTCFG=11

Tab.31 shows the starting ports configuration of the module: #PORTCFG=11. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT commands entered through USIF0 port. AT0 parser executes the AT commands, and after that NMEA sentences and AT commands run on USIF0 port as summarized in the table below. See also Fig.28.

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | **AT0** | **AT1** | **AT2** | **TT** | **/** |
| **No USB cable** | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| **USBHSI0** | | | | | |
| **USBHSI1** | | | | | |
| **USBHSI2** | | | | | |
| **USBHSI3** | | | | | |
| **USIF0** | **X** | | | | |
| **USIF1** | | | | | **External GPS** |
| **SPI** | | | | | |

*Tab.42: USIF1 Port Connected to External GPS*

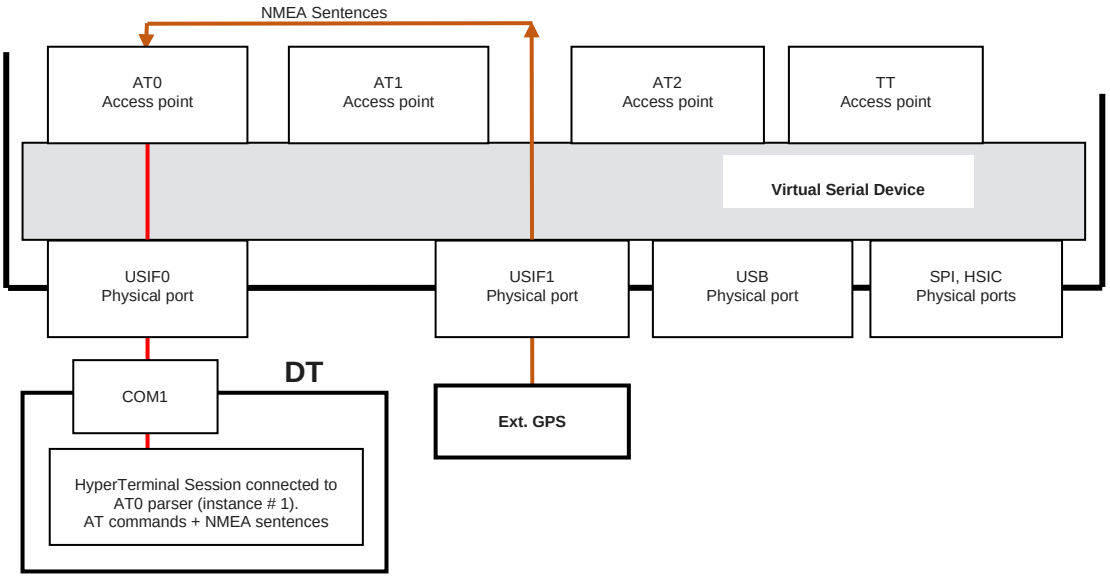



*Fig.28: USIF0 Port Support AT Commands + NMEA Sentences (External GPS)*

### 6.1.2.2 AT#PORTCFG=11 + USB

Tab.32 shows the starting ports configuration of the module: #PORTCFG=11+USB. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1… AT 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 summarized in the table below. See also Fig.29.

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 this example), or USB3 port.

| AT#PORTCFG=11 | | | | | |
|---|---|---|---|---|---|
| | AT0 | AT1 | AT2 | TT | / |
| USB0 | | X | | | |
| USB1 | | | | TTC | |
| USB2 | | | | | |
| USB3 | | | X | | |
| USB4 | | | | | |
| USB5 | | | | | |
| USB6 | | | | | |
| USBHSI0 | | | | | |
| USBHSI1 | | | | | |
| USBHSI2 | | | | | |
| USBHSI3 | | | | | |
| USIF0 | X | | | | |
| USIF1 | | | | | External GPS |
| SPI | | | | | |

*Tab.43: USIF1 Port Connected to External GPS + USB Cable*

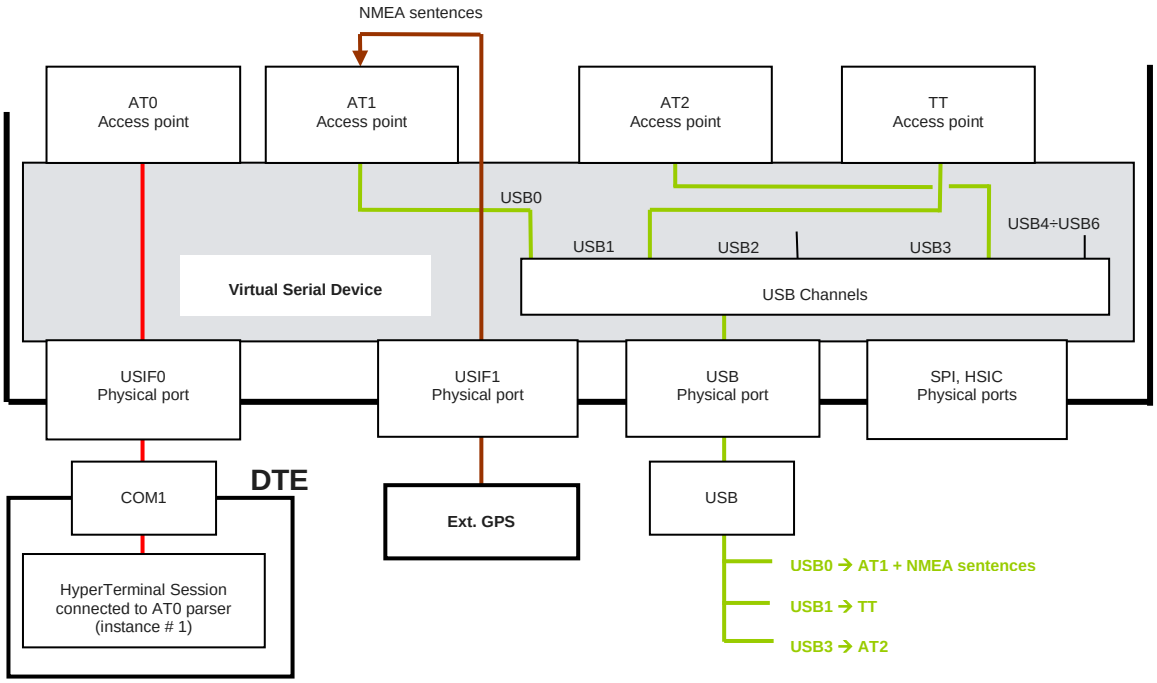



*Fig.29: USB0 Port Support AT Commands + NMEA Sentences (External GPS)*

## 6.2 Python

It is assumed that the reader is familiar with Python language. Refer to document [3] to have information on it. Telit's modules provide the Python programming language to develop user control scripts. As shown in Fig.30, the VSD provides two access points called VHWDTE0 and VHWDTE1. MDM and MDM2 Python modules are logically connected respectively to VHWDTE0 and VHWDTE1.

Assume that the module is using the ports configuration #PORTCFG=1, summarized on Tab.11 (factoring-setting), no USB cable is plugged in.

Referring to Fig.30: when the Python script runs the *import MDM instruction*, the VSD disconnects the USIF0/AT0 logical connection and sets up the logical connection VHWDTE0/AT0; now, the script can access AT0 parser. In the same way, *import MDM2 instruction* requires that the VSD sets up the logical connection VHWDTE1/AT1. The figure shows that USIF0 is disconnected and cannot be used by an external device.

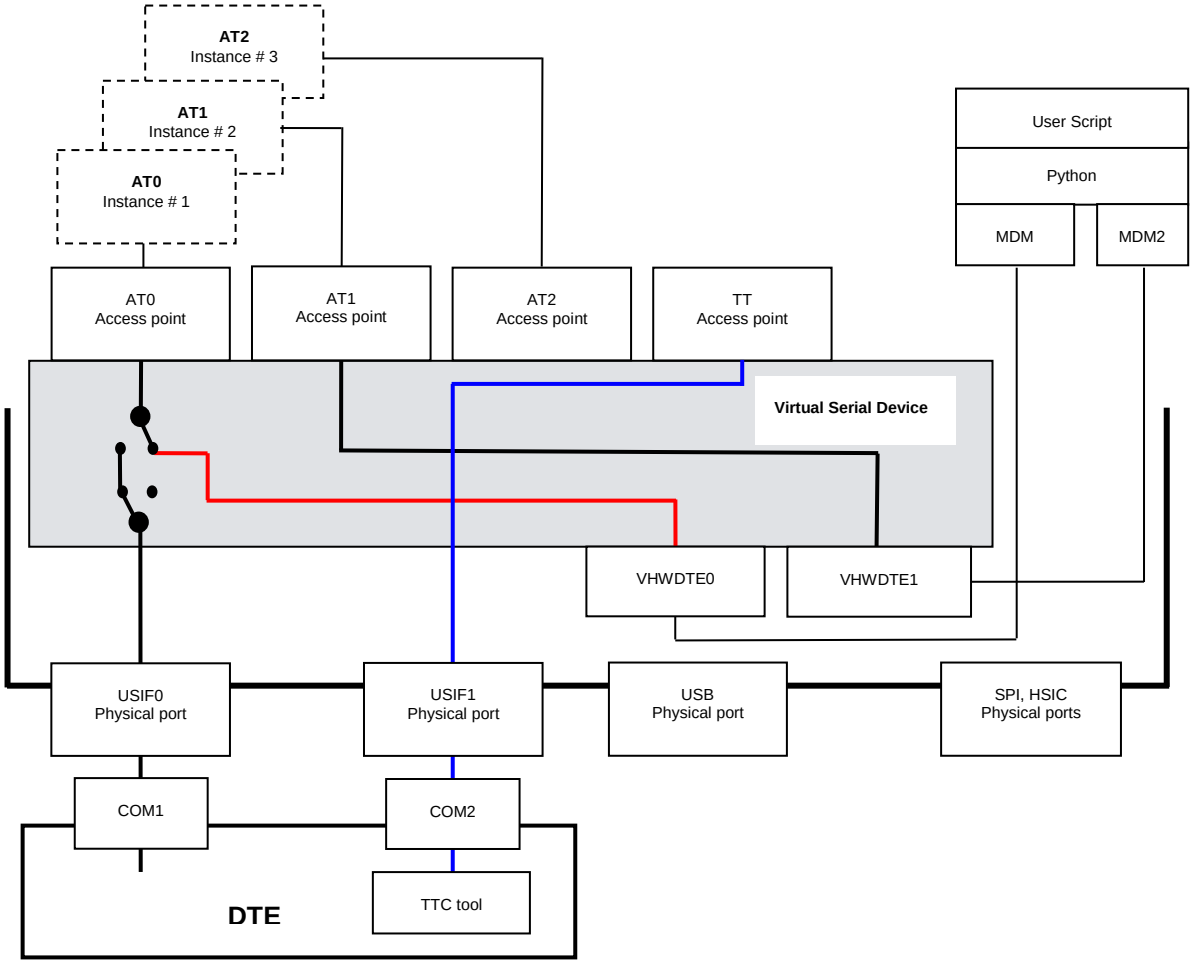



*Fig.30: Python & MDM, MDM2 Modules*

Python script can use the USIF0 port through the *import SER instruction*. See Fig.31

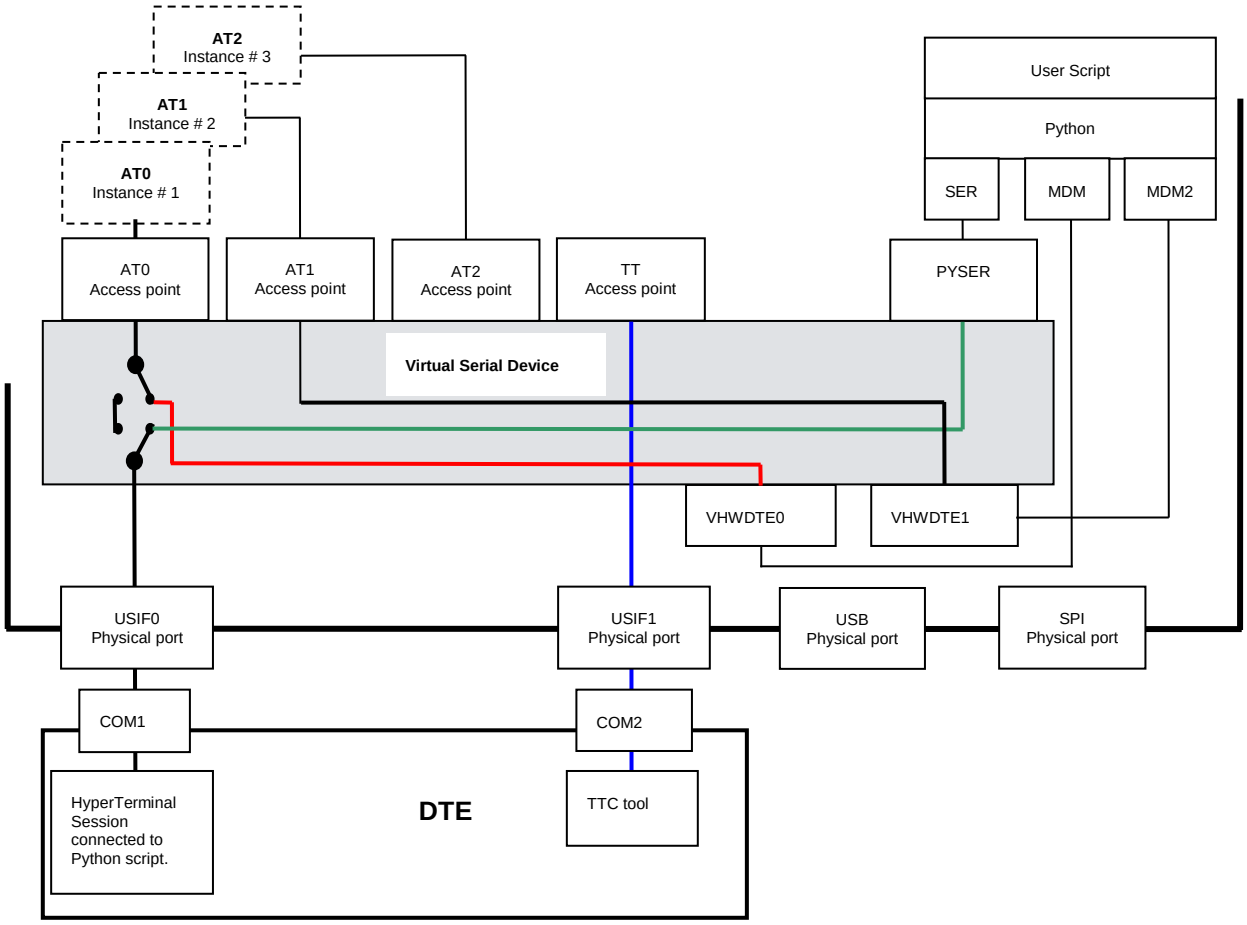



*Fig.31: Python & MDM, MDM2, SER Modules*

In accordance with the installed software version, Python script can run the *import USB0 instruction* to access the USB0 port by means of PYUSB0 access point. See Fig.32.

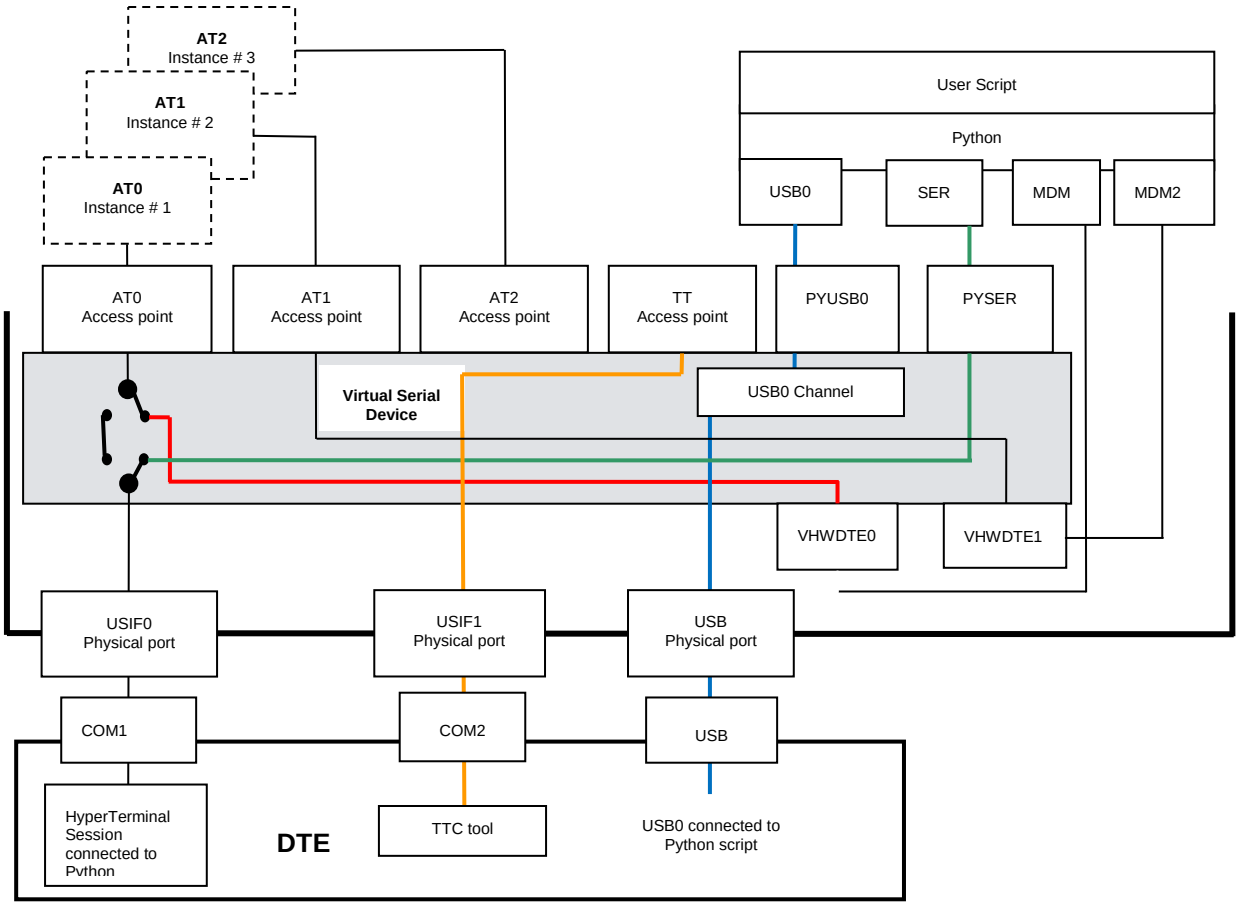



*Fig.32: Python & MDM, MDM2, SER, USB0 Modules*

The Python software modules MDM, MDM2, SER, and USB0 use four independent resources: AT0, AT1 Access Points, and USIF0 physical port, USB0 port. No resources contention can arise among them. MDM, MDM2, SER, and USB0 instructions steal the above-mentioned resources regardless their current owner.

As shown in the next sub-chapters there are other Python modules to create logical connection between a physical port and an Access point.

### 6.2.1 Python Script Debugging

Assume that the user needs to debug a new Python script. To perform the debugging session, the user forces the module into #PORTCFG=3 ports configuration, refer to Tab.15. Suppose that the Python script runs: *import MDM*, *import MDM2*, *import SER* and *print instructions*.

The next figure sketches the actions results of the first tree instructions. Moreover, the figure shows that the Python script switches the USIF1 from AT2 Access point to Python Debugging Access point; the print messages are available on USIF1 port.

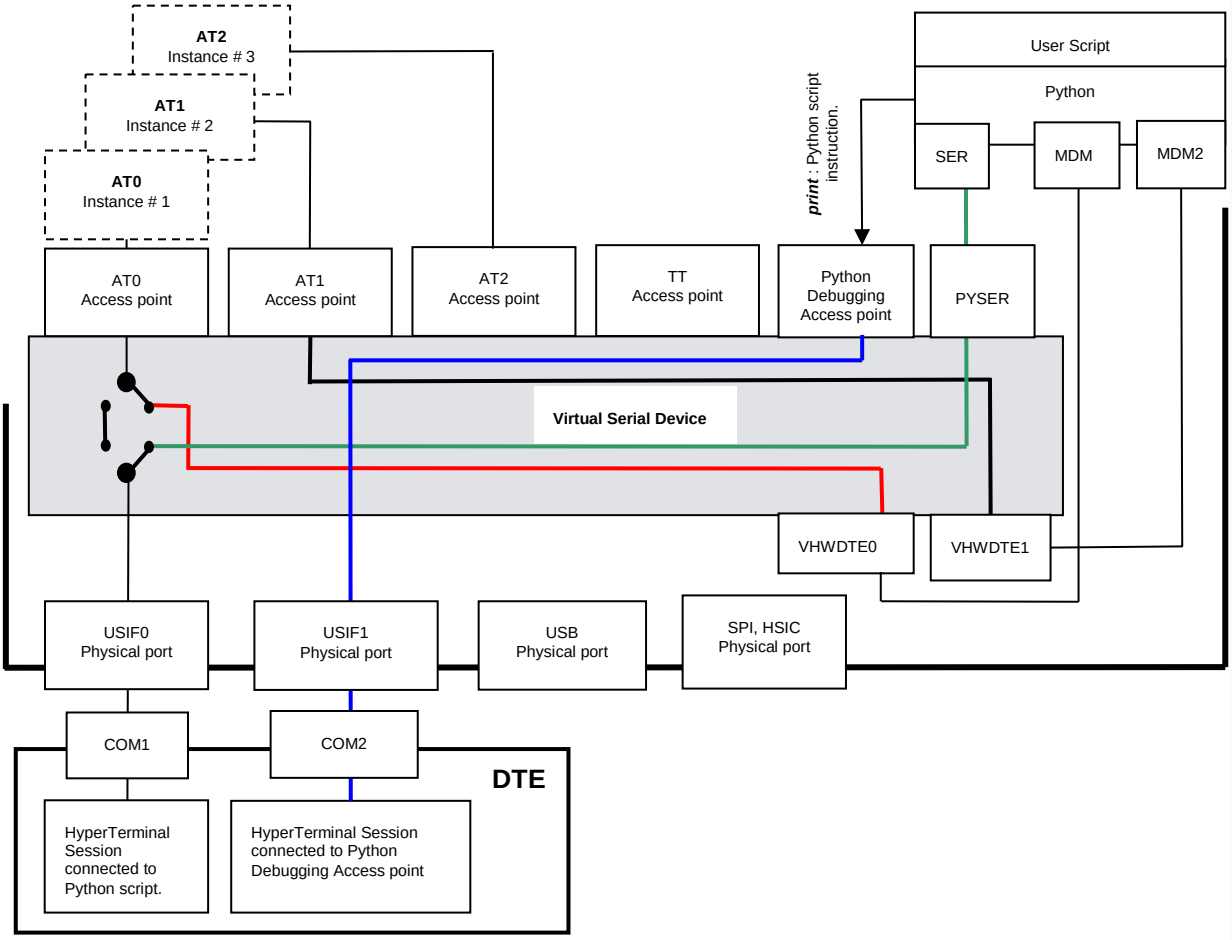



*Fig.33: Python & MDM, MDM2, SER and Print Modules*

### 6.2.2 SER2 Instruction

#### 6.2.2.1 AT#PORTCFG=0

Assume that the module is using the ports configuration #PORTCFG=0, no USB cable, refer to Tab.9. When the Python script runs the instruction *import MDM*, the VSD disconnects the USIF0/AT0 logical connection and establishes the logical connection VHWDTE0/AT0; now, the script can access AT0 parser. In the same way, *import MDM2* instruction requires that VSD sets up the logical connection VHWDTE1/AT1; refer to Fig.34.

In accordance with the installed software version, refer to document [3], Python script can run the *import SER2* instruction to use the USIF1 port through the access point PYSER2. The figure below shows the new connection

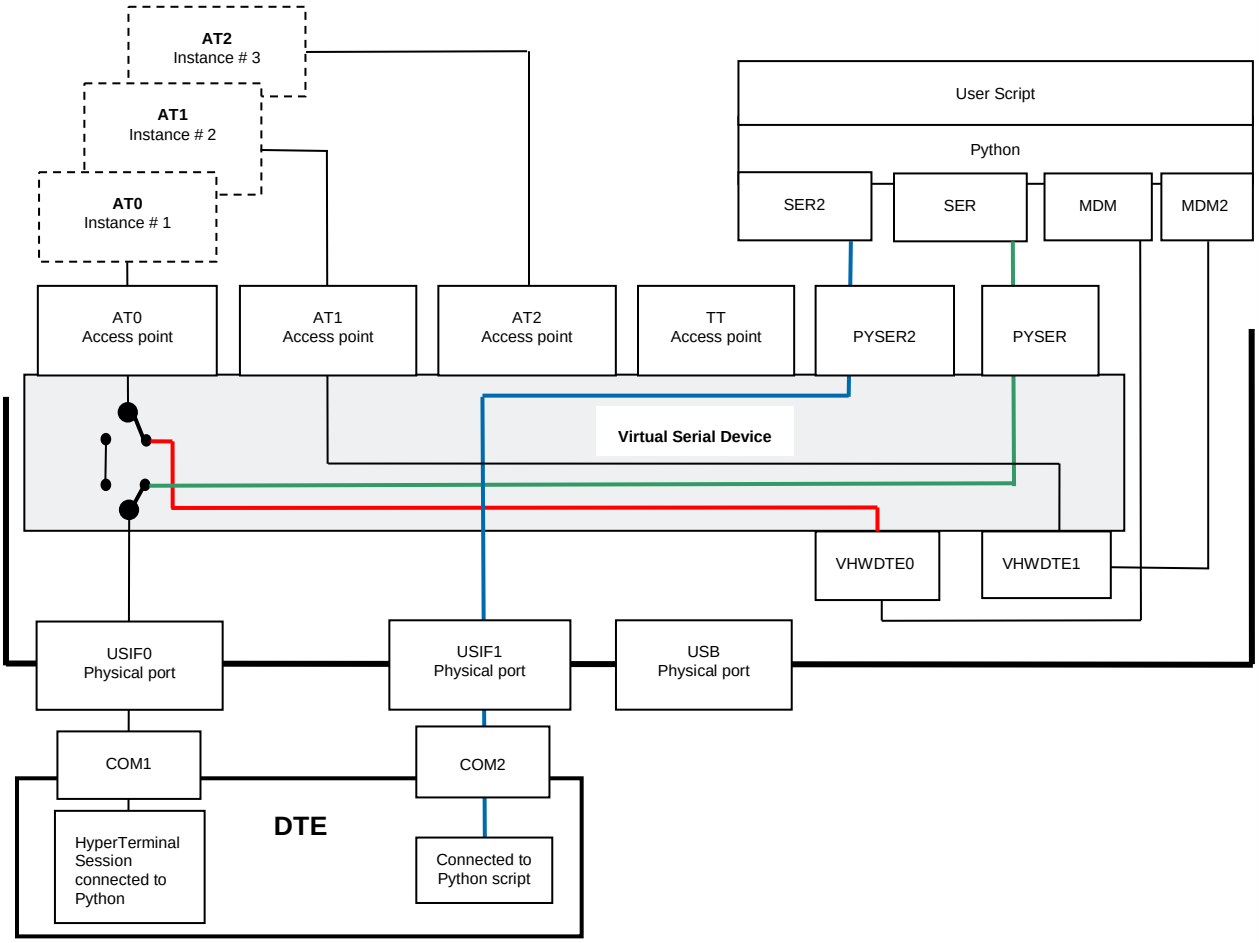



*Fig.34: Python & MDM, MDM2, SER, SER2 Modules*

### 6.2.2.2. AT#PORTCFG=3

Assume that the module is using the ports configuration #PORTCFG=3, no USB cable, refer to Tab.15. When the Python script runs the instruction *import MDM*, the VSD disconnects the USIF0/AT0 logical connection and establishes the logical connection VHWDTE0/AT0; now, the script can access AT0 parser. In the same way, *import MDM2 instruction* requires that VSD sets up the logical connection VHWDTE1/AT1; refer to Fig.34.

In accordance with the installed software version, refer to document [3], Python script can run the *import SER2 instruction*. In this case, the VSD disconnects the USIF1/AT2 logical connection and sets up the logical connection USIF1/PYSER2; now, the script can access USIF1.

The Python software modules MDM, MDM2, SER, and SER2 use four independent resources: AT0, AT1 Access Points, and USIF0, USIF1 physical ports. No resources contention can arise among them. As a rule, we can say that the MDM, MDM2, SER, and SER2 instructions steal the resources regardless their current owner.

NOTICE:

*print instruction* (see chapter 6.2.1) and SER2 instruction, both use the USIF1 hardware resource. In case of USF1 contention, SER2 instruction steals USIF1 to *print instruction*.

## 6.3 AppZone

NOTICE:

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

### 6.3.1 USIFx Ports

Example 1

Tab.11 shows the starting ports configuration of the module: #PORTCFG=1, no USB cable is connected. For example, run a user AppZone Application that does not use neither serial ports nor any ATx parsers; Fig.35 shows the resulting ports arrangement.

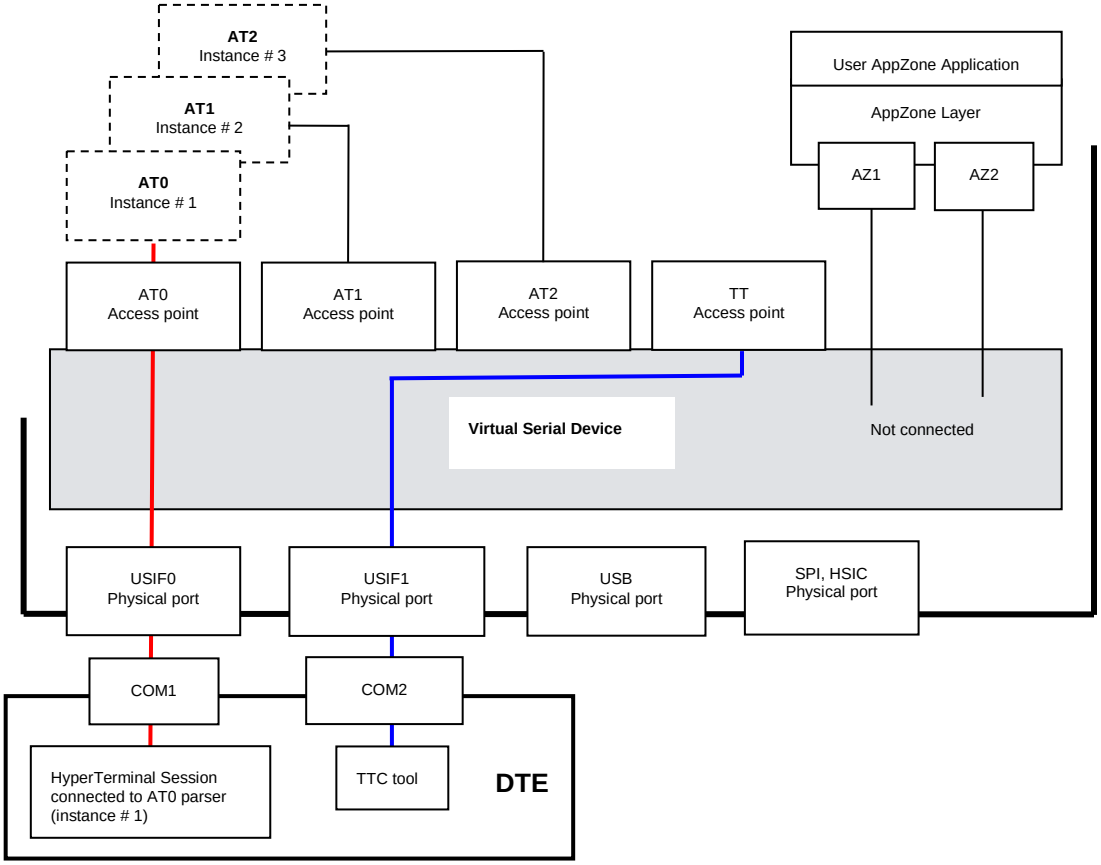



*Fig.35: AppZone Application without Connections*

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.36 shows the resulting ports configuration.

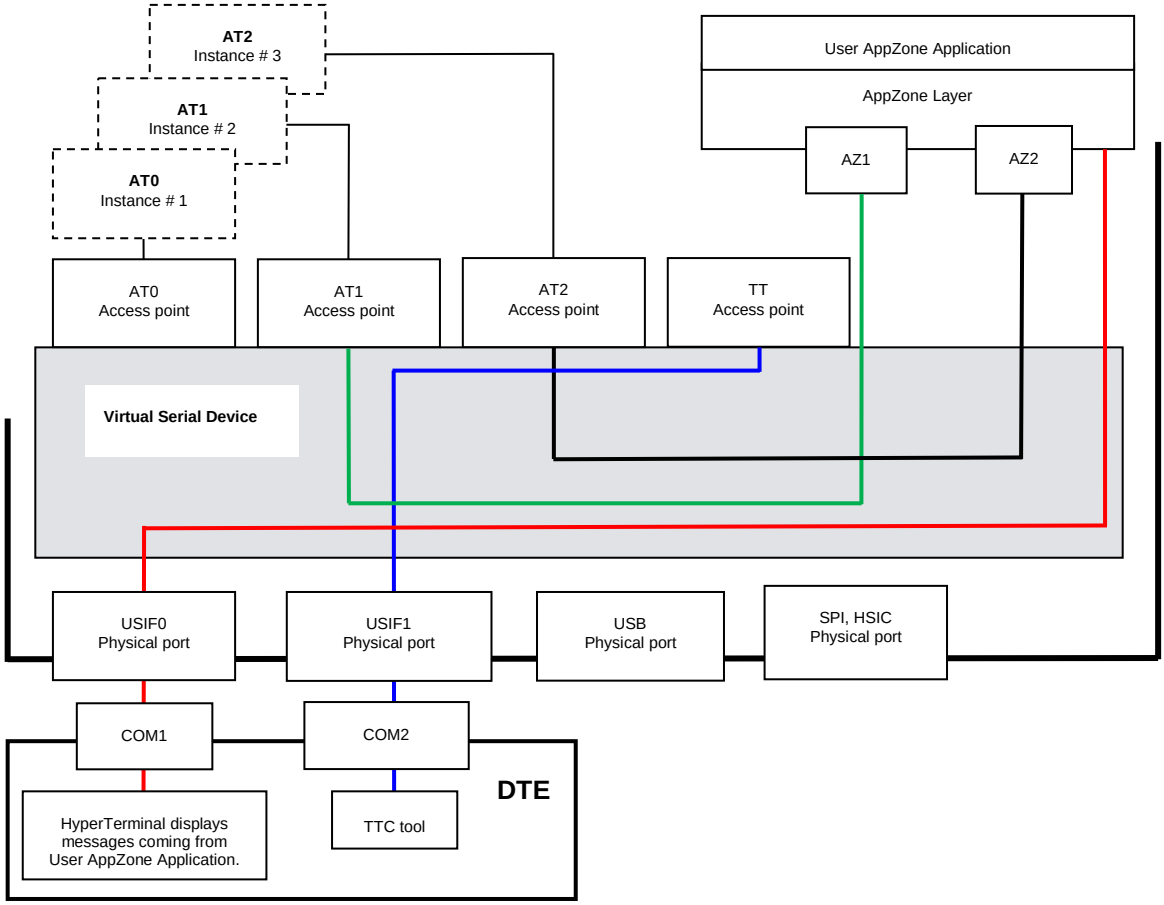



*Fig.36: AppZone Application Connected to AT1, AT2 Parsers, and USIF0 Serial Port*

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)**

API to route data, received from USIF0, to AT1 parser. Fig.37 shows the resulting ports configuration.

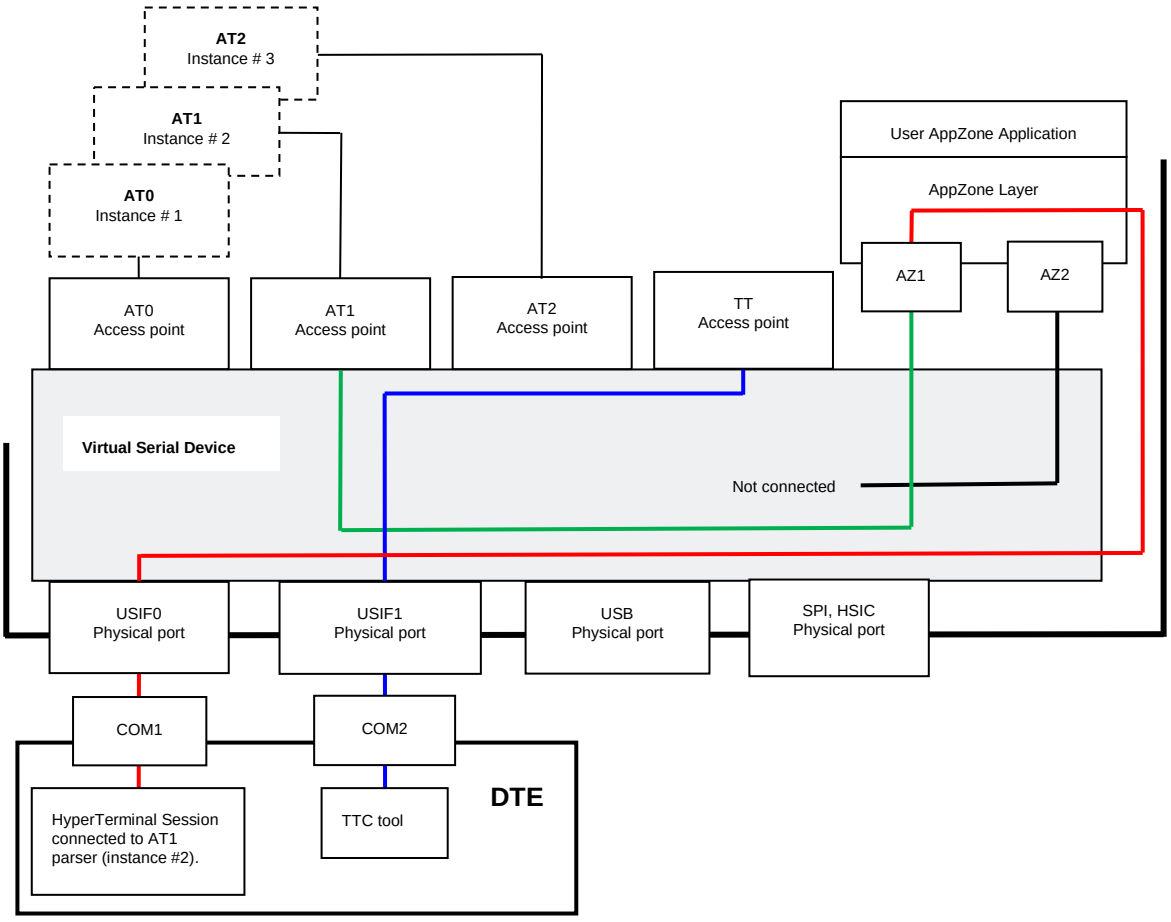



*Fig.37: USIF0 Connected to AT1 Parser through AppZone Layer*

### 6.3.2 USBx Ports

Example 1

Tab.12 shows the starting ports configuration of the module: #PORTCFG=1 (factory-setting) with USB cable. Out of the six USB ports, only two ports are available for the user AppZone applications: USB0, and USB3. Use:

**m2m_hw_usb_open(USB_CH0, handle0)** to disconnect USB0 port from AT1 parser, connect it to AppZone layer, and get its handle. When you use **USB_CH0**, the API try to open the USB0 port, and returns the related handle.

**m2m_hw_usb_open(USB_CH3, handle3)** to disconnect USB3 port from AT2 parser, connect it to AppZone layer, and get its handle. When you use **USB_CH3**, the API try to open the USB3 port, and returns the related handle.

Fig.38 shows the new configuration:

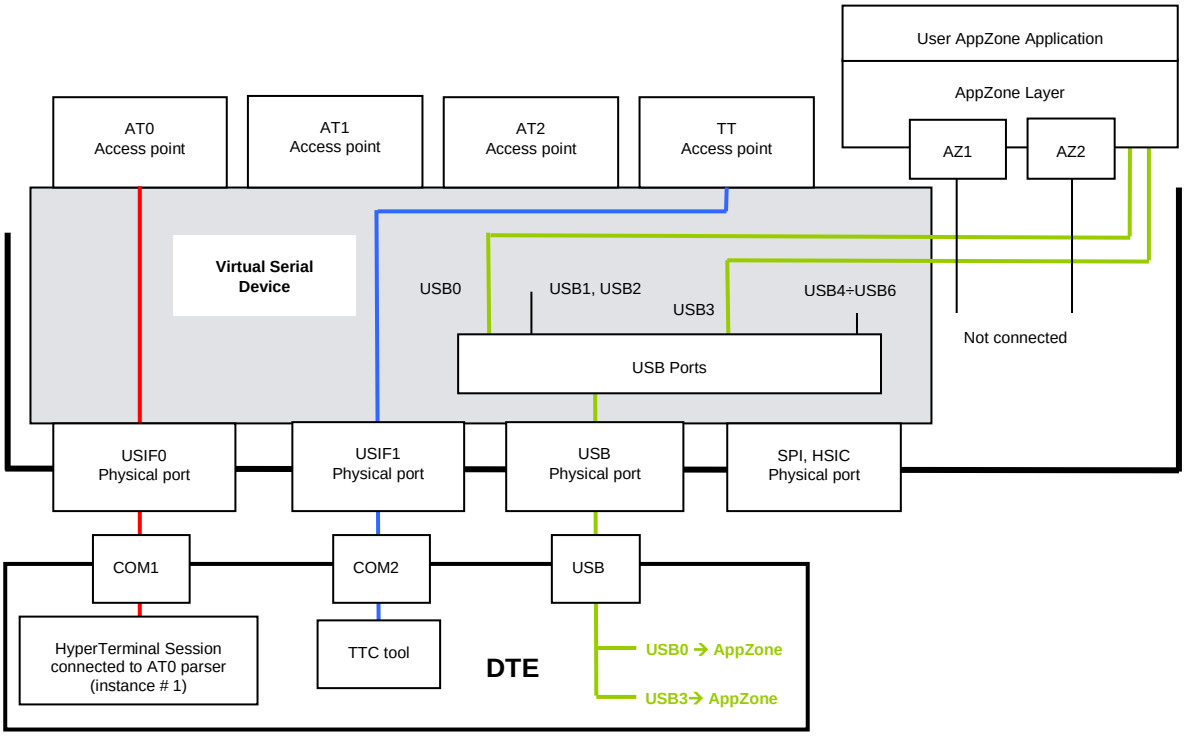



*Fig.38: USB0 and USB3 Ports Available for AppZone Application*

Example 2

Tab.26 shows the starting ports configuration of the module: #PORTCFG=8, with USB cable. Out of the six USB ports, only three ports are available for the user AppZone applications: USB0, USB3, and USB4, USB1 is reserved for TTC. Use:

**m2m_hw_usb_open(USB_CH0, handle0)** to disconnect USB0 port from AT0 parser, connect it to AppZone layer, and get its handle. When you use **USB_CH0**, the API try to open the USB0 port, and returns the related handle.

**m2m_hw_usb_open(USB_CH3, handle3)** to disconnect USB3 port from AT1 parser, connect it to AppZone layer, and get its handle. When you use **USB_CH3**, the API try to open the USB0 port, and returns the related handle.

**m2m_hw_usb_open(USB_CH4, handle4)** to disconnect USB4 port from AT2 parser, connect it to AppZone layer, and get its handle. When you use **USB_CH4**, the API try to open the USB0 port, and returns the related handle.

Fig.39 shows the new configuration:

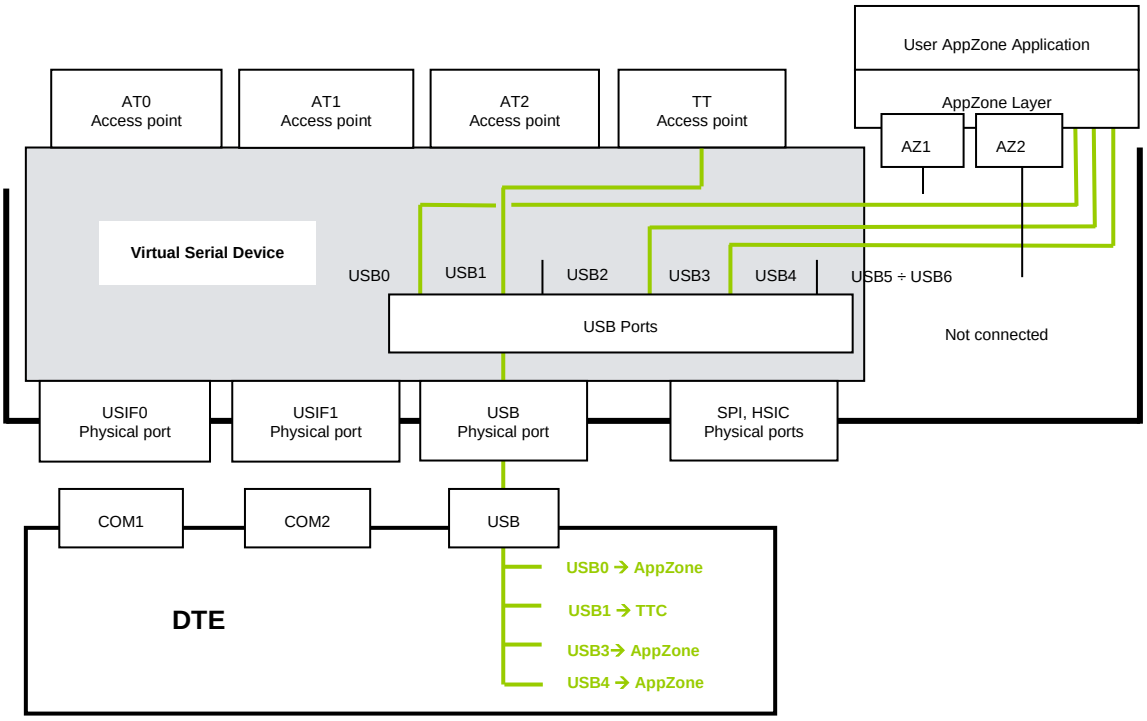



*Fig.39: USB0, USB3, and USB4 Ports Available for AppZone Application*

Example 3

Tab.26 shows the starting ports configuration of the module: #PORTCFG=8, with USB cable.

Use:

**m2m_hw_usb_ioctl(handle4, M2M_USB_AT_MODE_SET, M2M_HW_USB_IO_AT_MODE_ON)** API to route data received from USB4 port to AT1 parser.

Fig.40 shows the new configuration:

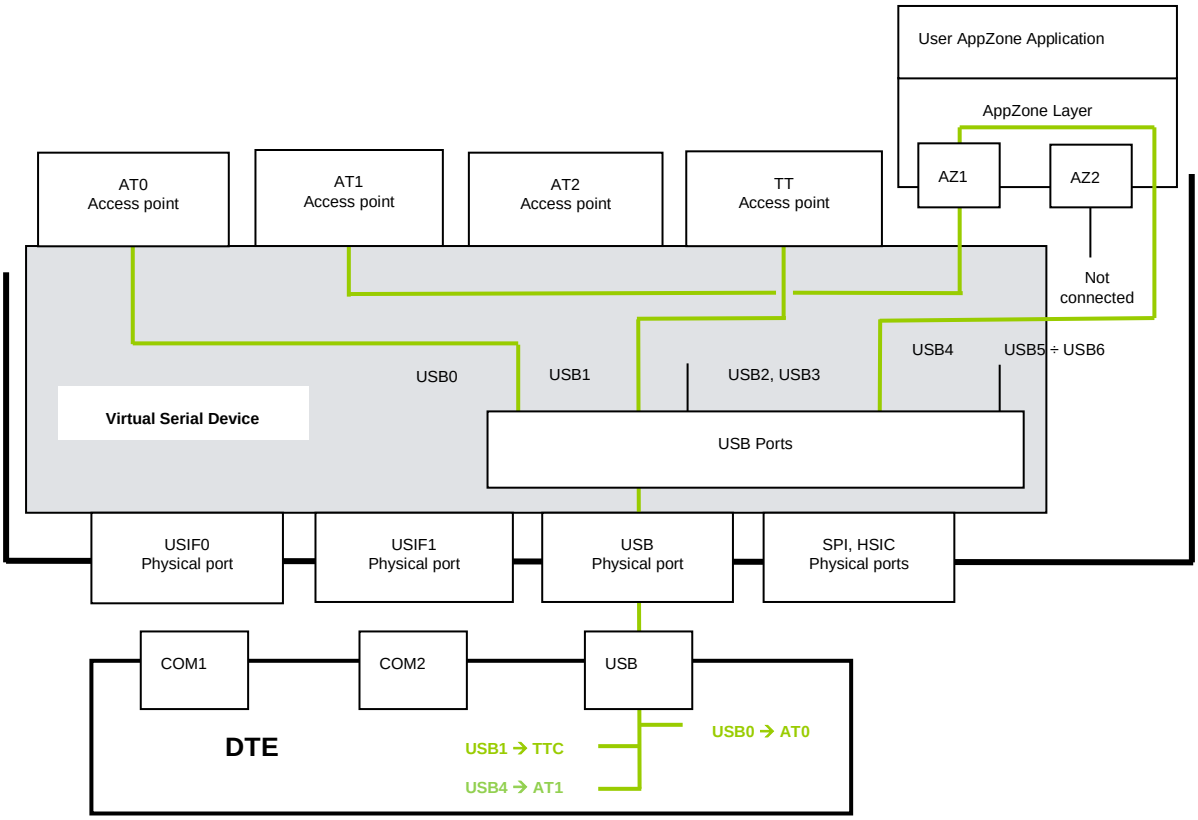



*Fig.40: USB4 Port connected to AT1*

Use:

**m2m_hw_usb_read(...)** or **m2m_hw_usb_write(...)** to read or write data from/to USBx ports.

NOTICE:

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

# 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.9: #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 (e.g. 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.18.

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.9: #PORTCFG=0, no USB Cable.

Example 2

Module: Tab.9: #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 COM20 ÷ COM23 to USB3→COM8 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.21: CMUX Connected to USB3 Port.

User action: disconnects USB cable.

Module: enters again the configuration shown on Tab.9: #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

| | |
|---|---|
| 3G Tool | Third Generation Trace Tool (for internal use only) |
| DTE | Data Terminal Equipment |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| HSIC | USB High Speed Inter-Chip Interface |
| MA | Mobile Analyzer (for internal use only) |
| NMEA | National Marine Electronics Association |
| OS | Operating System |
| PPP | Point to Point Protocol |
| SPI | Serial Peripheral Interface |
| TTC Tool | Telit Trace Client Tool |
| USIFx | Universal Serial Interface |
| VSD | Virtual Service Device |

# 9. MODULES & SW VER. TABLE

**SOFTWARE VER. TABLE**

| | SW Versions |
|---|---|
| HE910 Family | |
| HE910[3] | 12.00.xx6 |
| HE910-D | 12.00.xx6 |
| HE910-GL | 12.00.xx6 |
| HE910-EUR / HE910-EUD | 12.00.xx6 |
| HE910-EUG / HE910-NAG | 12.00.xx6 |
| HE910-NAR / HE910-NAD | 12.00.xx6 |
| UE/UL Family | |
| UE910-EUR / UE910-EUD | 12.00.xx6 |
| UE910-NAR / UE910-NAD | 12.00.xx6 |
| UL865-EUR / UL865-EUD | 12.00.xx6 |
| UL865-NAR / UL865-NAD | 12.00.xx6 |
| UL865-N3G | 12.00.xx6 |
| UE910-N3G | 12.00.xx6 |
| UL866-N3G | 12.00.xx6 |

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.

[3] HE910 is the “type name” of the products marketed as HE910-G & HE910-DG

## SERVICES COEXISTENCE TABLE

| | Services | | | |
|---|---|---|---|---|
| | Embedded GPS | External GPS | Python | AppZone |
| | | | Python and AppZone are mutually exclusive | |
| HE910 Family | | | | |
| HE910 | ✓ | | ✓ | ✓* |
| HE910-D | | ✓ | ✓ | |
| HE910-GL | | ✓ | ✓ | |
| HE910-EUR / HE910-EUD | | ✓ | ✓ | |
| HE910-EUG / HE910-NAG | ✓ | | ✓ | |
| HE910-NAR / HE910-NAD | | ✓ | ✓ | |
| UE/UL Family | | | | |
| UE910-EUR / UE910-EUD | | ✓ | ✓ | ✓* |
| UE910-NAR / UE910-NAD | | ✓ | ✓ | ✓* |
| UL865-EUR / UL865-EUD | | ✓ | ✓ | |
| UL865-NAR / UL865-NAD | | ✓ | ✓ | |
| UL865-N3G | | ✓ | ✓ | |
| UE910-N3G | | ✓ | ✓ | |
| UL866-N3G | | ✓ | ✓ | |

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

# 10. APPENDIXES

## 10.1 #USBCFG Modes

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

### 10.1.1 Mode #USBCFG=0

Set USB mode 0 (factoring setting).

**AT#USBCFG=0**

OK

Activate the just set mode

**AT#REBOOT**

OK

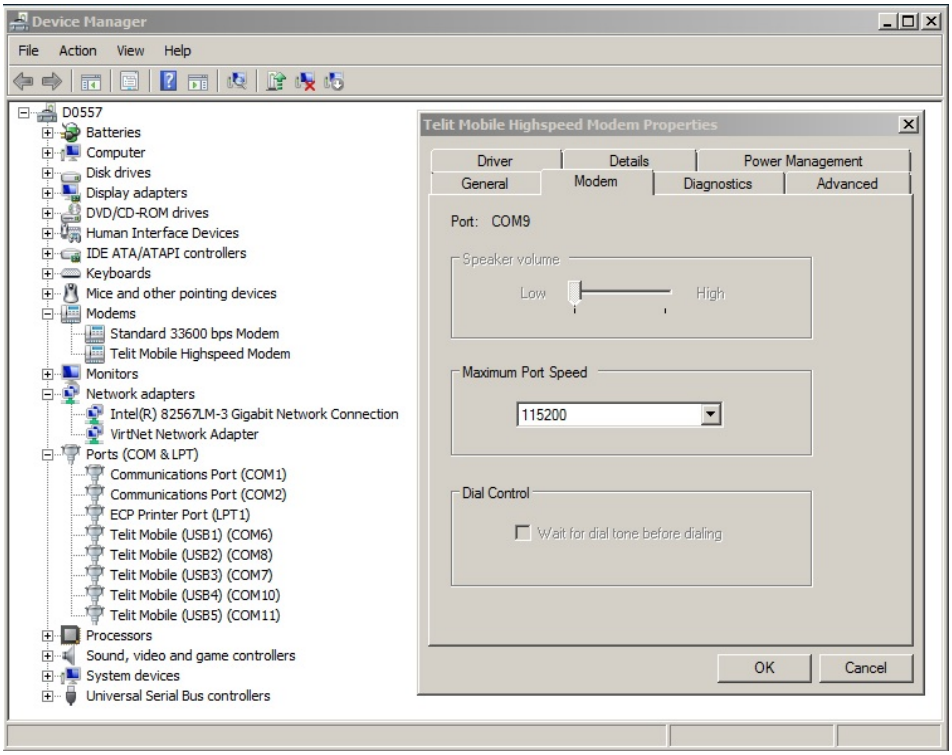
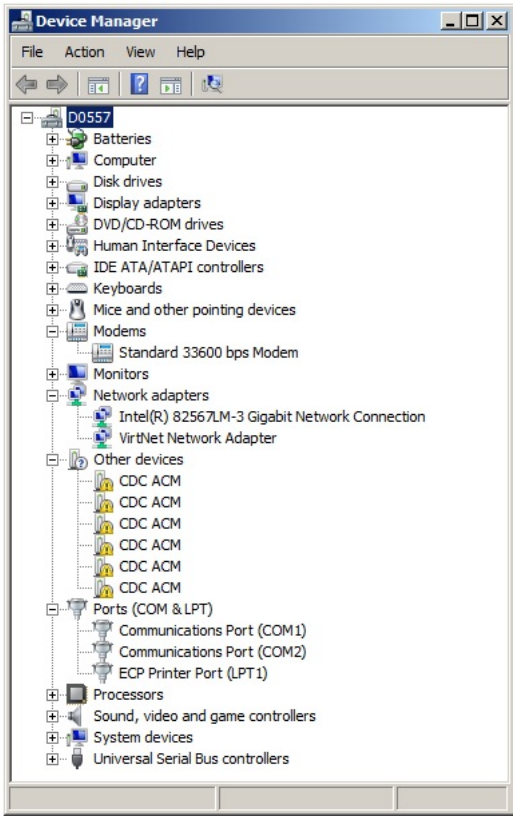



*Fig.41: #USBCFG=0*

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, and compare it with the Tab. 3.




### 10.1.1.1 Ubuntu OS

Fig.42 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=0021 (#USBCFG=0).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0021
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
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.42: #USBCFG=0 Mode, Ubuntu*

### 10.1.2 Mode #USBCFG=1

Mode #USBCFG=1 is not Windows.

Set USB mode 1

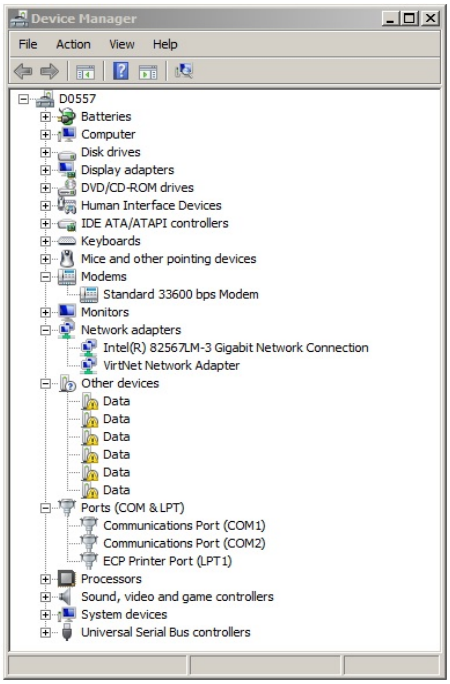
**AT#USBCFG=1**

OK

Activate the just set mode

**AT#REBOOT**

OK



#### 10.1.2.1 Ubuntu OS

Use the following commands to activate the right in-box driver.

user@d0508-U64:~ sudo su
root@d0508-U64:/home/user# modprobe option
root@d0508-U64:/home/user# echo 1bc7 0026 > /sys/bus/usb-serial/drivers/option1/new_id

Where: idVendor = 1bc7, idProduct = 0026

The following figure shows the message returned by the dmesg command when you plug in the USB cable, see Telit idVendor=1bc7, and idProduct=0026 (#USBCFG=1).

```
usbcore: registered new interface driver option
usbserial: USB Serial support registered for GSM modem (1-port)
usb 1-3: new high-speed USB device number 112 using ehci-pci
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0026
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM Data Only
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
option 1-3:1.0: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB0
option 1-3:1.1: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB1
option 1-3:1.2: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB2
option 1-3:1.3: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB3
option 1-3:1.4: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB4
option 1-3:1.5: GSM modem (1-port) converter detected
usb 1-3: GSM modem (1-port) converter now attached to ttyUSB5
```

*Fig.43: #USBCFG=1 Mode, Ubuntu*

### 10.1.3 Mode #USBCFG=2

Set USB mode 2.

**AT#USBCFG=2**

OK

Activate the just set mode

**AT#REBOOT**

OK

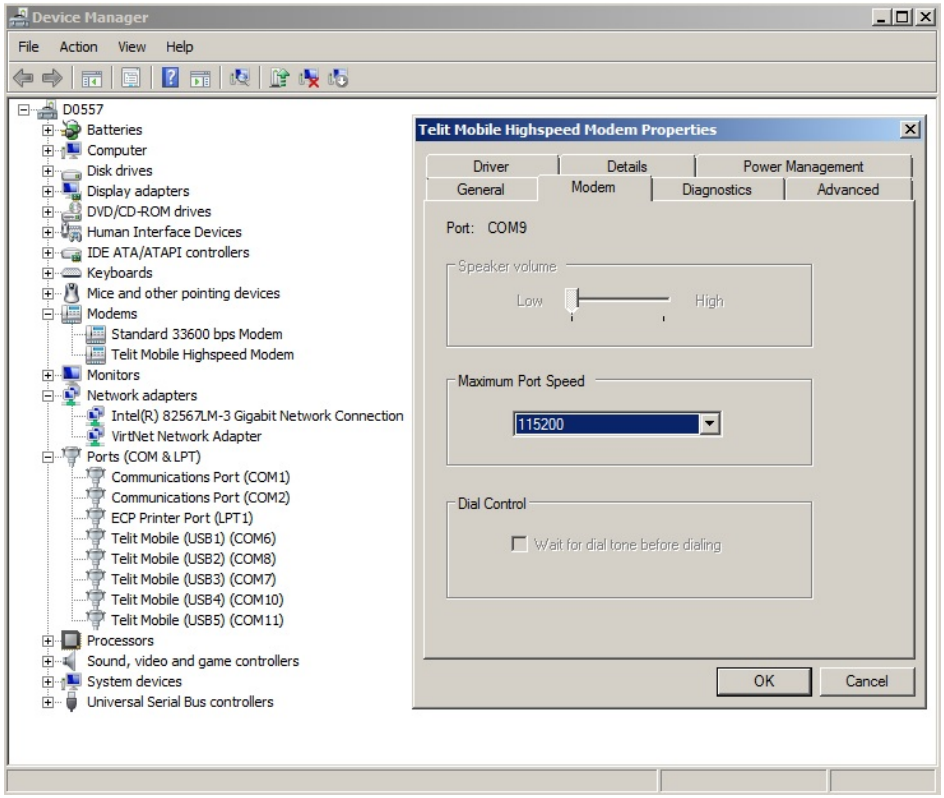
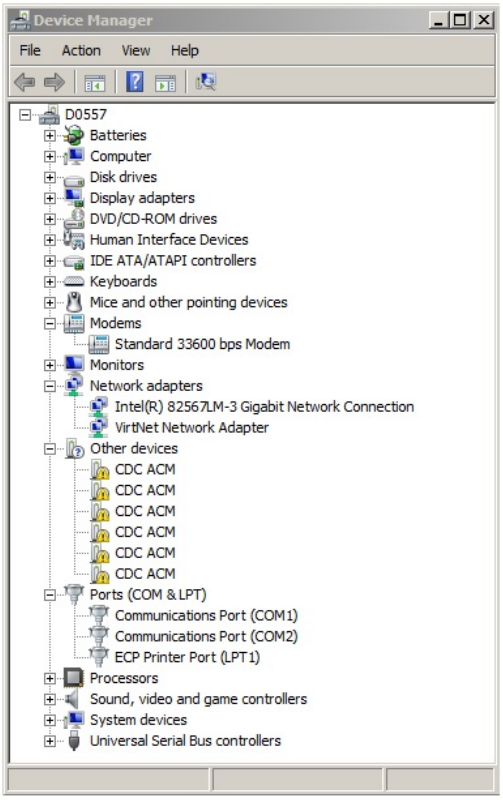



*Fig.44: #USBCFG=2*

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, and compare it with the Tab. 3.



#### 10.1.3.1 Ubuntu OS

Fig.45 shows the message returned by the dmesg command when you plug the USB cable in. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0021 (#USBCFG=2).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0021
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
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.45: #USBCFG=2 Mode, Ubuntu.*

### 10.1.4 Mode #USBCFG=3

Set USB mode 3.

**AT#USBCFG=3**

OK

Activate the just set mode

**AT#REBOOT**

OK

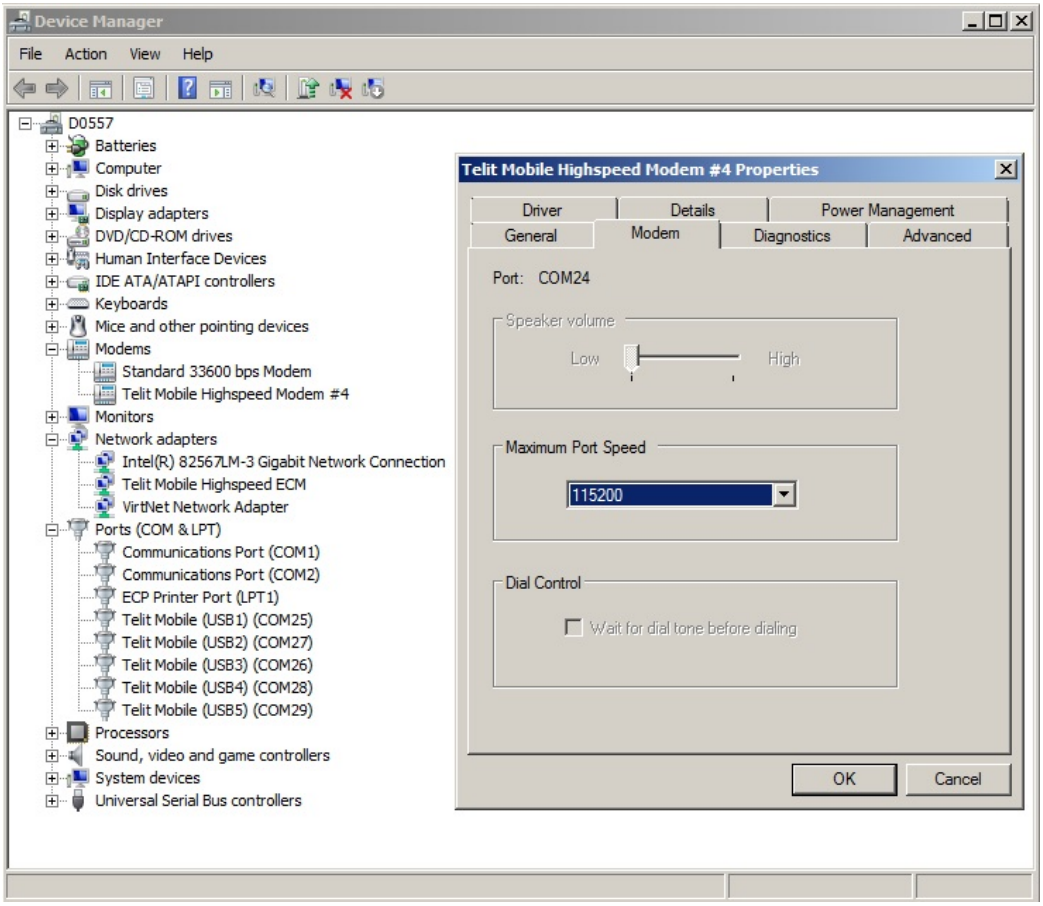
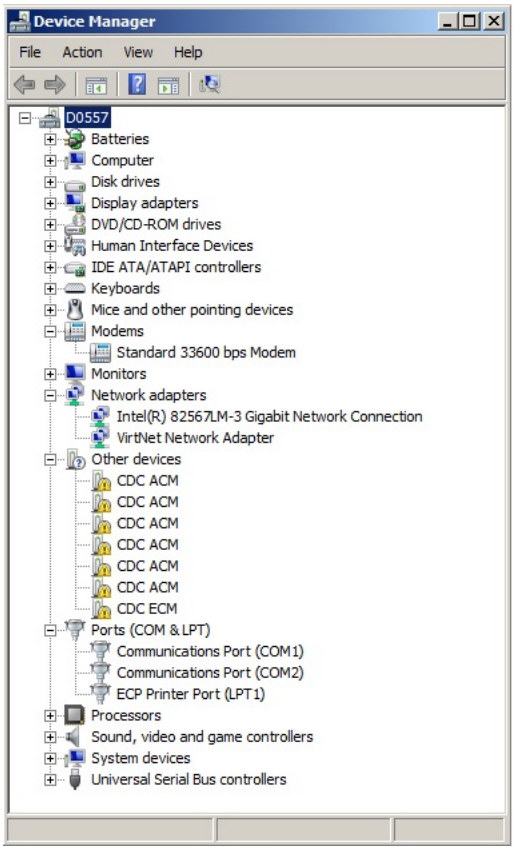



*Fig.46: #USBCFG=3*

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, and compare it with the Tab. 3.




#### 10.1.4.1 Ubuntu OS

Fig.47 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=0023 (#USBCFG=3).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0023
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM + 1 CDC-ECM
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
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_ether 1-3:1.12 wwan0: register 'cdc_ether' at usb-0000:00:1a.7-3, Mobile Broadband Network Device, 00:00:11:12:13:14
usbcore: registered new interface driver cdc_ether
```

*Fig.47: #USBCFG=3, Ubuntu*

### 10.1.5 Mode #USBCFG=4

Set USB mode 4.

**AT#USBCFG=4**

OK

Activate the just set mode

**AT#REBOOT**

OK

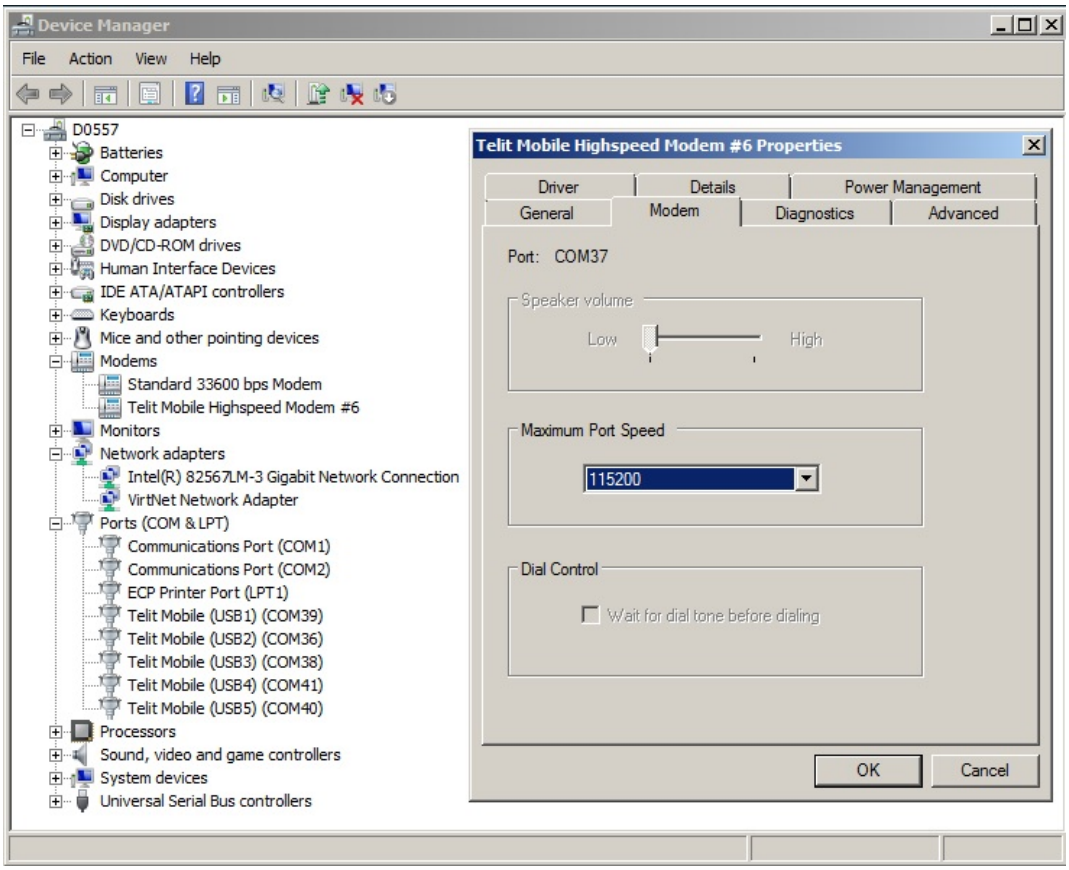
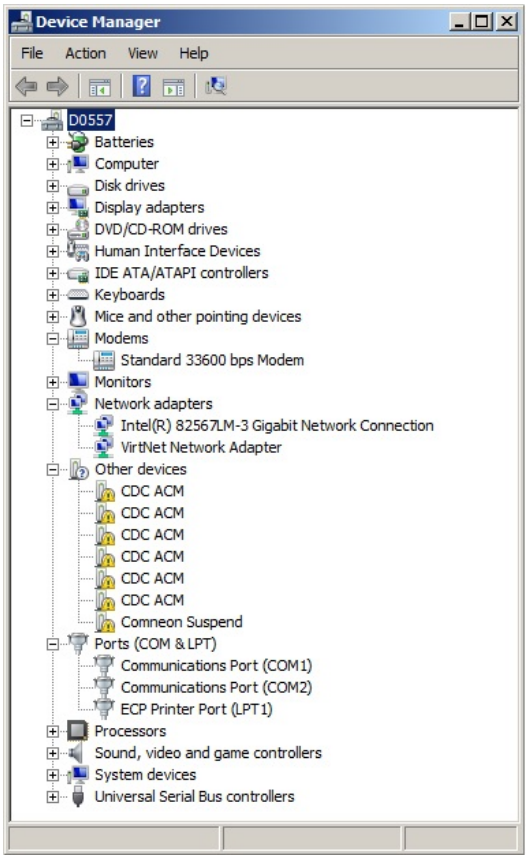



*Fig.48: #USBCFG=4*

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, and compare it with the Tab. 3.




#### 10.1.5.1 Ubuntu OS

Fig.49 shows the message returned by the dmesg command when you plug the USB cable in. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0024 (#USBCFG=4).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0024
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
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.49: #USBCFG=4, Ubuntu*

### 10.1.6 Mode #USBCFG=5

Set USB mode 5.

**AT#USBCFG=5**

OK

Activate the just set mode

**AT#REBOOT**

OK

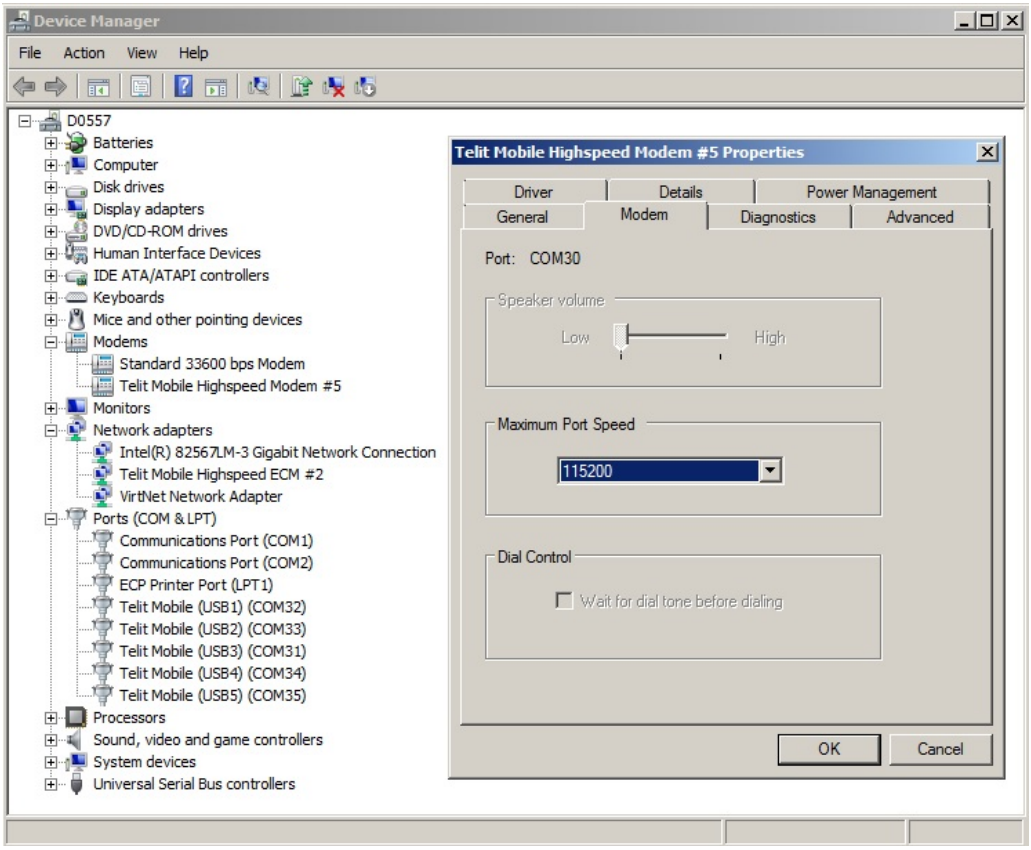
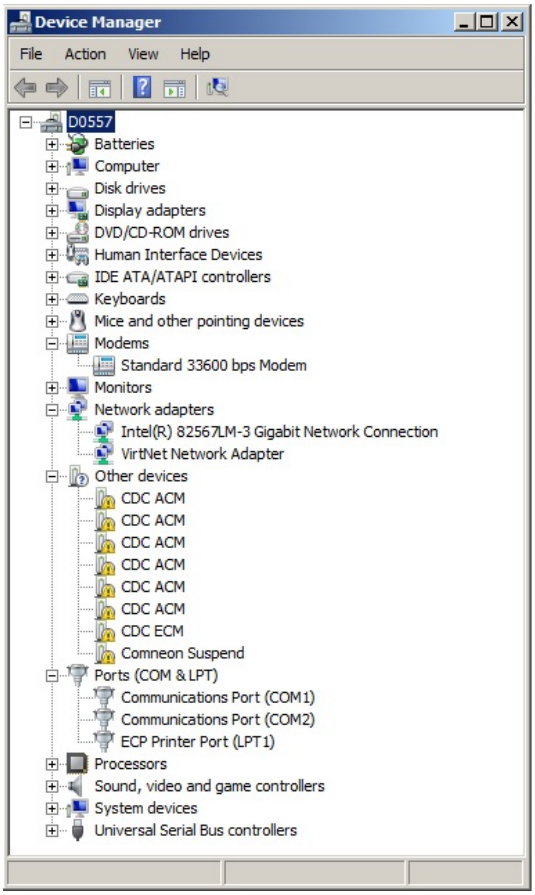



*Fig.50: #USBCFG=5*

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, and compare it with the Tab. 3.




### 10.1.6.1 Ubuntu OS

Fig.51 shows the message returned by the dmesg command when you plug the USB cable in. This OS uses the in-box driver, see Telit idVendor=1bc7, and idProduct=0025 (#USBCFG=5).

```
usb 1-3: New USB device found, idVendor=1bc7, idProduct=0025
usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3: Product: 6 CDC-ACM + 1 CDC-ECM
usb 1-3: Manufacturer: Telit
usb 1-3: SerialNumber: 357164040526850
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_ether 1-3:1.12 wwan0: register 'cdc_ether' at usb-0000:00:1a.7-3, Mobile Broadband Network Device, 00:00:11:12:13:14
```

*Fig.51: #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.52:

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 (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.

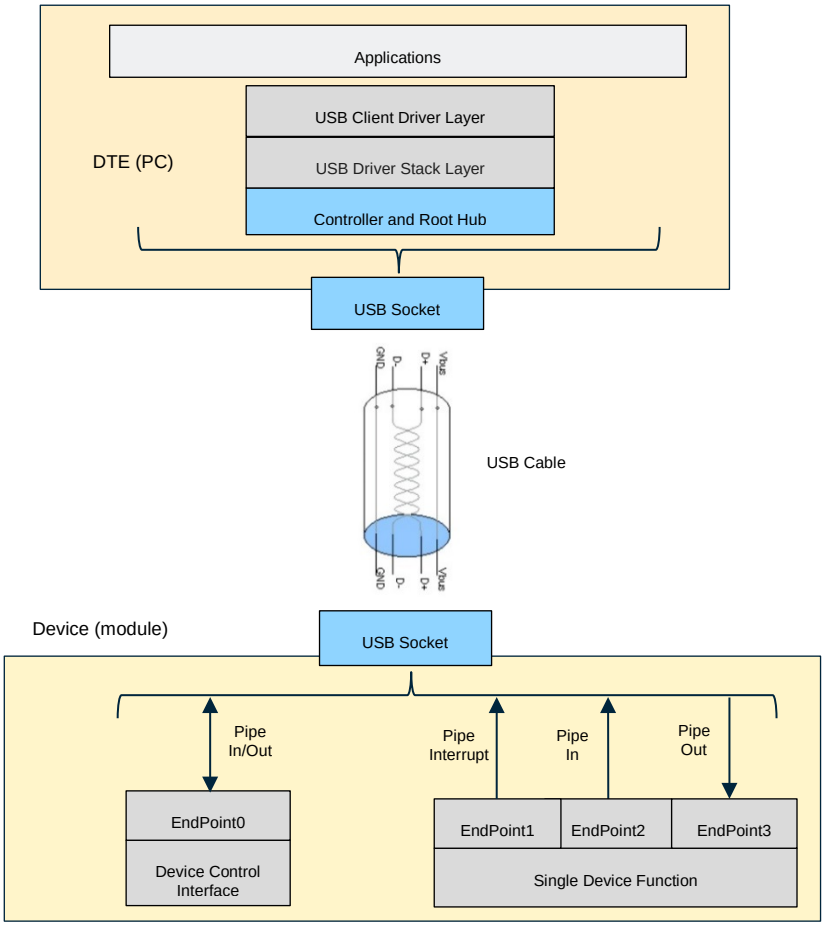



*Fig.52: A Short USB Overview*

## 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 Windows Device Manager, see the screenshot below.

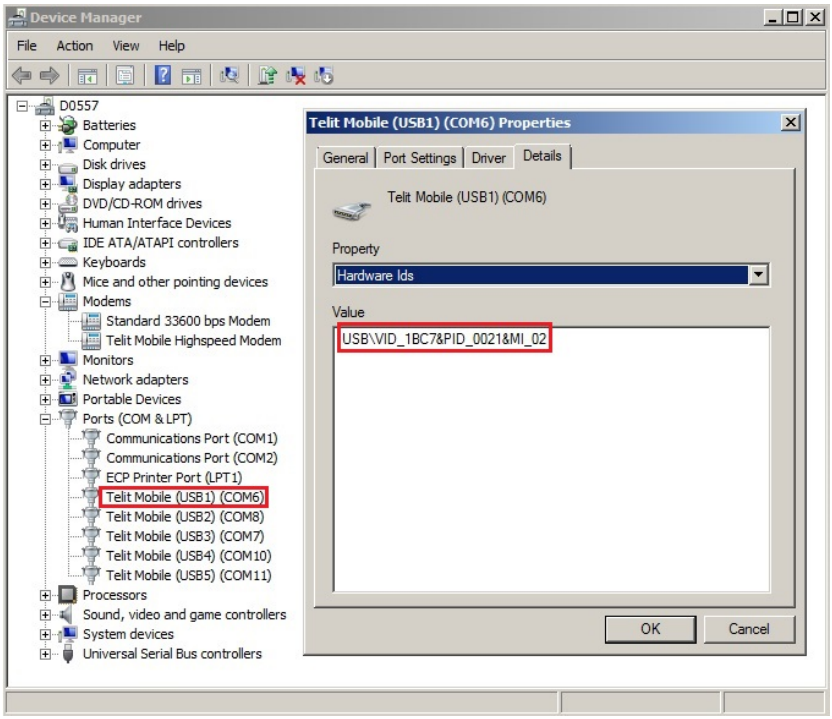



*Fig.53: Vendor Identifier & Product Identifier*

- 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) uses the 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.

# 11. DOCUMENT HISTORY

| Revision | Date | Changes |
|---|---|---|
| 0 | 2011-11-23 | First issue |
| 1 | 2011-11-23 | Mobile Analyzer changed in Trace Tool (Generic TT) |
| 2 | 2012-01-27 | Updated the Applicability Table<br>AT+CMUX=1 changed in AT+CMUX=0 |
| 3 | 2012-02-28 | Updated parameter range of AT#PORTCFG Command<br>Modified the SPI Physical port connection on all figures<br>Introduced TTC and 3G (see TT |
| 4 | 2012-07-03 | Added PORTCFG=7 and updated PORTCFG tables.<br>Modified document title “HE Family Ports Arrangements” in “HE910 Family Ports Arrangements”.<br>General review of the entire document. |
| 5 | 2012-09-17 | Added PORTCFG=8.<br>Updated Applicability Table |
| 6 | 2013-07-29 | The document title has been changed from “HE910 Family Ports Arrangements” to “HE910/UE910 Families Ports Arrangements”<br>Updated Applicability Table, and some figures.<br>Rearranged GPS chapters and modified some names of chapters.<br>Added the note about +CFUN command in chapter 2.<br><br>New features supported from SW version 12.00.004:<br>PORTCFG=9, PORTCFG=10, and HSIC physical port.<br>The factory setting has been changed from #PORTCFG=0 to #PORTCFG=1.<br><br>Added products:<br>UE910-EUR/EUD/12.00.xx4<br>UE910-NAR/NAD/12.00.xx4 |
| 7 | 2014-02-24 | Added some figures and related descriptions. |

| Revision | Date | Changes |
|---|---|---|
| 8 | 2014-02-28 | The document title has been changed from “HE910/UE910 Family Ports Arrangements” to “HE910/UE910/UL865 Families Ports Arrangements”.<br><br>Added products:<br>UL865-EUR/EUD 12.00.xx4<br>UL865-NAR/NAD 12.00.xx4<br>UL865-N3G 12.00.xx4 |
| 9 | 2015-02-16 | Chapters’ organization has been rearranged; some chapters have been reviewed or removed.<br><br>Added:<br>Services Coexistence Table, AppZone Service, AT#PORTCFG=11, and AT#PORTCFG=12<br><br>Updated Applicability Table<br>HE910 Family<br>HE910 12.00.xx6<br>HE910-D 12.00.xx6<br>HE910-EUR / HE910 EUD 12.00.xx6<br>HE910-EUG / HE910-NAG 12.00.xx6<br>HE910-NAR / HE910-NAD 12.00.xx6<br><br>UE/UL Family (Embedded)<br>UE910-EUR / UE910-EUD 12.00.xx6<br>UE910-NAR / UE910-NAD 12.00.xx6<br>UL865-EUR / UL865-EUD 12.00.xx6<br>UL865-NAR / UL865-NAD 12.00.xx6<br>UL865-N3G 12.00.xx6 |
| 10 | 2016-02-10 | This revision has adopted the new Telit format. According to the new format, chapters and figures has been rearranged.<br><br>The document title has been changed from “HE910/UE910/UL865 Family Ports Arrangements” to “Telit 3G Modules Ports Arrangements”<br><br>Has been added the chapters dedicated to the AT#USBCFG command.<br><br>Updated Applicability Table and Modules & SW ver. Table:<br><br>HE910 Series<br>HE910 12.00.xx6<br>HE910-D 12.00.xx6<br>HE910-GL 12.00.xx6<br>HE910-EUR / HE910 EUD 12.00.xx6 |

| Revision | Date | Changes |
|---|---|---|
| | | HE910-EUG / HE910-NAG 12.00.xx6<br>HE910-NAR / HE910-NAD 12.00.xx6<br><br>UE/UL Series (Embedded)<br>UE910-EUR / UE910-EUD 12.00.xx6<br>UE910-NAR / UE910-NAD 12.00.xx6<br>UL865-EUR / UL865-EUD 12.00.xx6<br>UL865-NAR / UL865-NAD 12.00.xx6<br>UL865-N3G 12.00.xx6<br>UE910-N3G 12.00.xx6<br>UE866-N3G 12.00.xx6 |
| 11 | 2016-07-06 | Changed AT#PORTSCFG=? into AT#PORTCFG=?, chapter 3. |
| 12 | 2017-02-14 | 2017 Template applied |

# SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

## www.telit.com


Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Co., Ltd.
8th Fl., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit Wireless Solutions
Technologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil



Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit www.telit.com

Copyright © 2016, Telit