Telit wireless solutions

HE910 Family Ports Arrangements

(Virtual Service Device)

User Guide

1vv0300971 Rev.4 – 2012-07 -03

Making machines talk.

APPLICABILITY TABLE

NOTICE: *the information provided by the present document is concerning the products having a software version equal or less than the versions showed on the following table. The* ***software version*** *is indicated by the digits in bold style; the platform version is indicated by the first two digits..*

If some features are only provided by a particular software version, suitable information will be pointed out in the guide.

Table Legend: ● command is supported; – command is not supported

HE910 Family	Platform Version	AT Ref. Guide	Technology Family	#SELINT=0	#SELINT=1	#SELINT=2
HE910 [1]	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●
HE910-GA	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●
HE910-D	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●
HE910-EUR/ HE910-EUD	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●
HE910-EUG/ HE910-NAR	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●
HE910-NAD/ HE910-NAG	12.**00**.xx**3**	[2]	HSPA-GSM/GPRS	–	–	●

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

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

Notice

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 entirely 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

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.

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

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.

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.

Contents

Figures

Tables

1. Introduction

The purpose of the present document is to provide a guideline to assign the physical interfaces of the module to the services implemented by the module itself, e.g.: AT Parsers, Python, TT, etc. The HE910 Family modules need to be configured in suitable way in order to avoid hardware/software resources conflicts.

1.1. Scope

Scope of this guide is to cover the major ports/services arrangements offered to the user by the modules belonging to the HE910 Family. With ports/services arrangements is intended the allocation of an available serial port to an available service (e.g. AT Parser, Python, TT, etc.).

1.2. Audience

This document is intended for User Application designers who want to exploit at best the communication resources offered by the HE910 Family modules without run up against resources contention among services.

1.3. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit Technical Support Center (TTSC) at:

TS-EMEA@telit.com
TS-NORTHAMERICA@telit.com
TS-LATINAMERICA@telit.com
TS-APAC@telit.com

Alternatively, use:

http://www.telit.com/en/products/technical-support-center/contact.php

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).

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. Related Documents

[1] Telit' CMUX Implementation User Guide, Telit document: 1vv0300994
[2] HE910 AT Commands Reference Guide, Telit document: 80378ST10091A
[3] /
[4] Telit Easy Script Python, Telit document: 80000ST10020a
[5] HE910 Hardware User Guide, Telit document: 1vv0300925

1.5. Document History

Revision	Date	Changes
0	2011-11-23	First issue
1	2011-11-28	Mobile Analyzer changed in Trace Tool (Generic TT)
2	2012-01-27	Updated the Applicability Table AT+CMUX=1 changed in AT+CMUX=0
3	2012-02-28	Updated parameter range of AT#PORTCFG Command Modified the SPI Physical port connection on all figures Introduced TTC and 3G (see TT)
4	2012-07-03	Added PORTCFG=7 and updated PORTCFG tables. Modified document title "HE Family Ports Arrangements" in "HE910 Family Ports Arrangements". General review of the entire document.

1.6. Abbreviation and acronyms

3G	Third Generation Trace Tool (for internal use only)
DTE	Data Terminal Equipment
GPS	Global Positioning System
TTC	Telit Trace Client Tool
NMEA	National Marine Electronics Association
SPI	Serial Peripheral Interface
USIFx	Universal Serial Interface
VSD	Virtual Service Device

2. HE910 Family Ports Arrangements and VSD

Before describing the several ports arrangements supported by the HE910 Family, refer to [5], it is useful to introduce the Virtual Serial Device.

Virtual Serial Device, hereafter called VSD, is a piece of software designed to run on HE910 Family. It basically manages virtual connections among the physical serial ports, accessible to the user, and the services provided by the module. To accomplish this activity, VSD supports several Access Points used as anchorage points for the logical connections. The following table shows the items involved in the connections management: Physical Serial Ports, Logical Access Points, AT Parser and TT Utilities, Services and Protocols. The VSD supports several configurations of these items that will be explained in this document.

Physical Serial Ports	Logical Access Points	AT Parsers and TT Utilities	Services	Protocols
USIF0[2] USIF1 USB (USB0÷USB6)[4] SPI	AT0 AT1 AT2 TT VHWDTE0 VHWDTE1 PYSER Python Debugging GPS	Instance #1 Instance #2 Instance #3 TTC, 3G	Python GPS	CMUX (VC1÷VC4)[3]

Tab. 1: Services & other Items

It is advisable to remind the concept of instances and their relationships with the Access Points. HE910 Family provides three AT Commands Parser Instances which are logically independent. Each one is managed by the same control software block and is connected to an Access Point as sketched on the figure below.

[2] In document [5] USIF0 and USIF1 are called respectively Modem Serial Port 1 and Modem Serial Port 2.
[3] 4 CMUX channels: VC1÷VC4.
[4] 7 USB channels: USB0÷USB6.

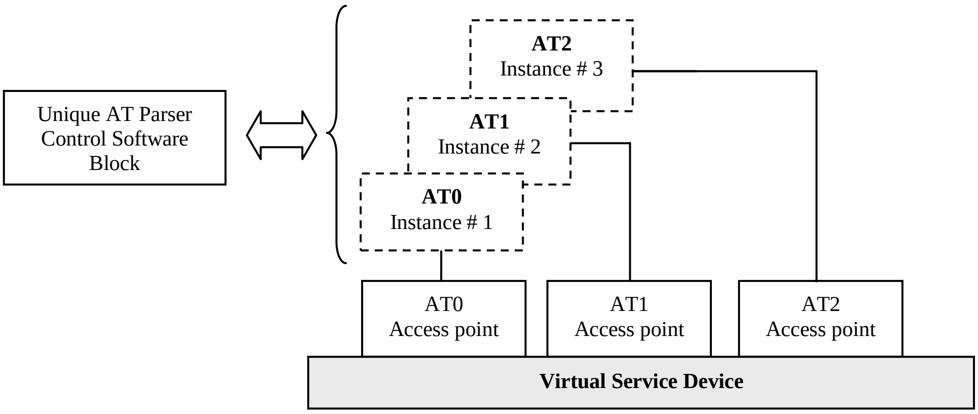


fig. 1: AT Parser Instance concept

2.1. The Device connected to the Module

The several examples of the module ports arrangements reported in this document refer to the use of a generic device connected to the module, and based on Windows PC. The physical connection between the module and the generic device can be accomplished via different interfaces provided by the module itself; a basic sketch is showed on fig. 2. It is worth noting that the generic device can be developed by the user in accordance with its needs. In general, the device can be a micro-controller entirely developed by the user and equipped with an operating system accomplishing the requirements of the user application. Only for practical reasons we use the acronym DTE to identified the device throughout the document.

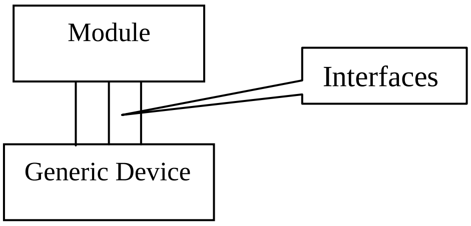


fig. 2: Module and Generic Device

In some examples where the module ports arrangement involves also the SPI interface are indicated two devices connected to the module: DTE and User Device with SPI interface. This distinction is used only to point out that the SPI interface must be provided by the user and the standard Windows PC (DTE) used by the examples does not provide it.

2.2. Factory Ports Arrangement with no USB cable

Let’s assume that the factory setting[5] of the module is not changed and the USB cable is not plugged in. Now, power on the module: the factory arrangement of the internal connections among physical ports and “Access points” is depicted on fig. 3. Tab. 2 summarizes the factory arrangement, USB0÷USB6 are the channels provided by the USB port, and they are not used in this example.

	AT0	AT1	AT2	TT	GPS/NMEA
USB0					
USB1					
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI					

Tab. 2: Factory Ports Arrangement

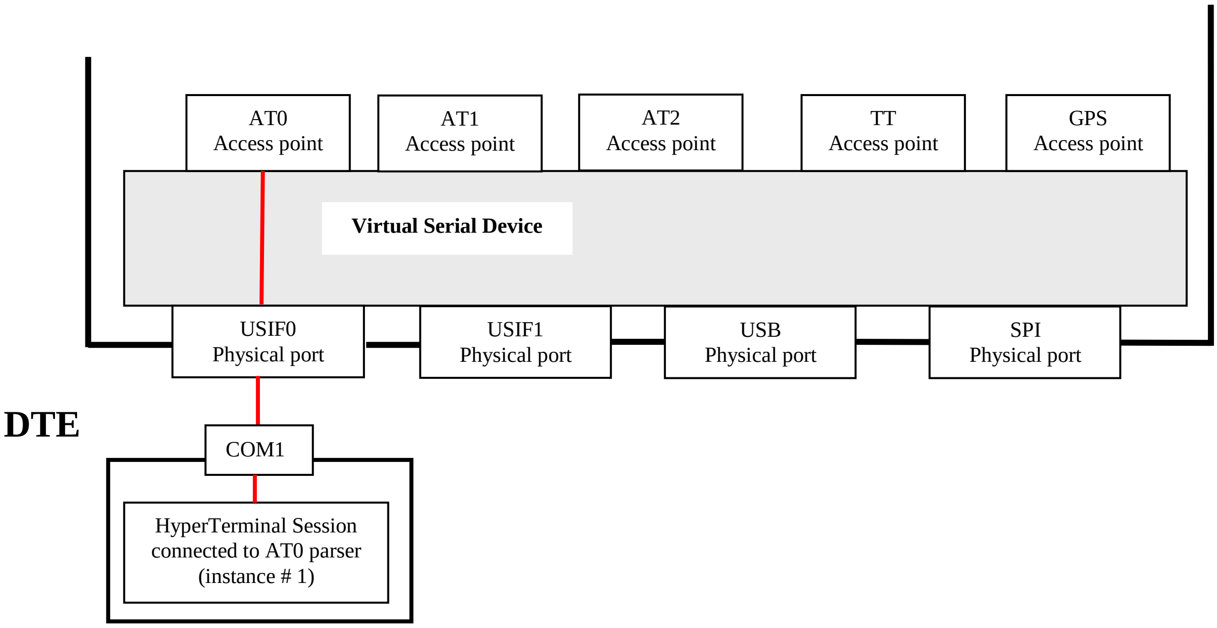


fig. 3: Factory Ports Arrangement

[5] AT#PORTCFG=0, refer to Chapter 2.4 and document [2].

2.2.1. GPS/NMEA sentences on USIF0

Assume that the module is configured as showed on fig. 3. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1,… commands, see [2], entered through USIF0 port. The entered commands are elaborated by AT0 parser and after that, the module enters the configuration showed on fig. 4. Tab. 3 summarizes the new internal ports configuration.

	AT0	AT1	AT2	TT	GPS/NMEA
USB0					
USB1					
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0	X				X
USIF1					
SPI					

Tab. 3: USIF0 port supports NMEA sentences

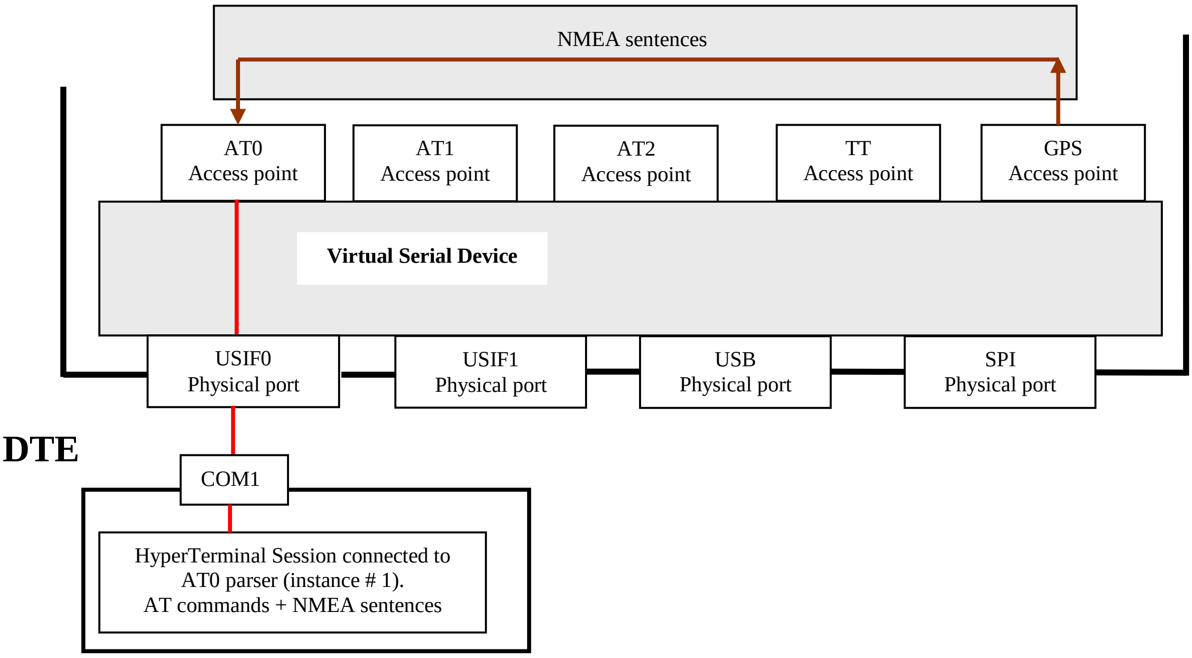


fig. 4: USIF0 port supports AT commands + NMEA sentences

NOTICE: NMEA sentences are sent on the physical port used by the operator to enter the AT$GPSP and AT$GPSNMUN commands. On the same physical port is still possible enter AT commands.

2.3. Factory Ports Arrangement with USB cable

Assume that the module is powered on and its configuration[6] is sketched by fig. 3. Now, connect the USB cable to the module. The module recognizes the “plug in” event and assumes the factory arrangement depicted on fig. 5. Tab. 4 summarizes the new factory configuration. USB0÷USB6 are the channels provided by the USB port. The suitable USB drivers running on Windows PC are provided by Telit. The mapping rule between USBx channel and virtual COMx port is showed on chapter 2.4.

	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3			X		
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI					

Tab. 4: Factory Ports Arrangement when USB cable is connected

NOTICE: figures reported on the following pages and illustrating different ports arrangements show the use of two types of trace applications in accordance with the selected configuration via AT#PORTCFG command, refer to chapter 2.4:

- TTC (Telit Trace Client tool);

- 3G tool (for internal use only).

[6] AT#PORTCFG=0, refer to Chapter 2.4.

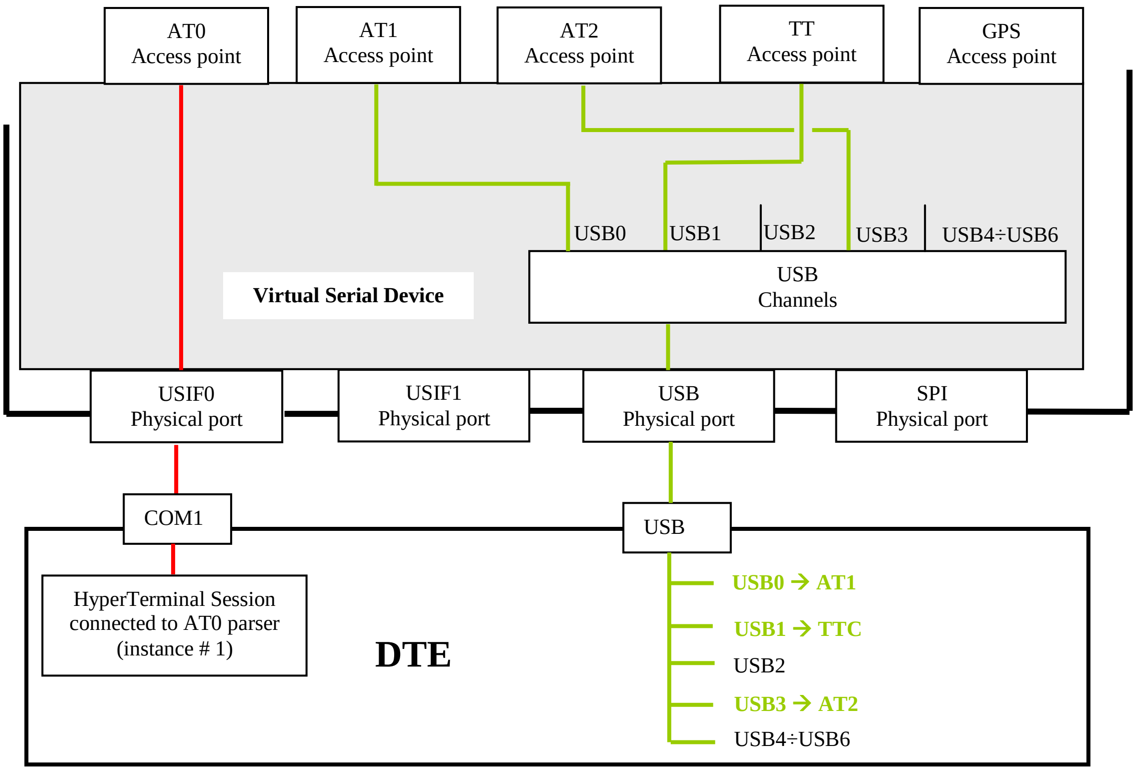


fig. 5: Factory Ports Arrangement when USB cable is connected

2.3.1. GPS/NMEA sentences on USB0

Assume that the module is configured as showed on fig. 5. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1,… commands entered through USB port, channel USB0. The commands are elaborated by AT1 parser and after that, the module enters the configuration showed on fig. 6. Tab. 5 summarizes the new internal ports configuration.

	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			X
USB1				TTC	
USB2					
USB3			X		
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI					

Tab. 5: USB0 channel supports NMEA sentences

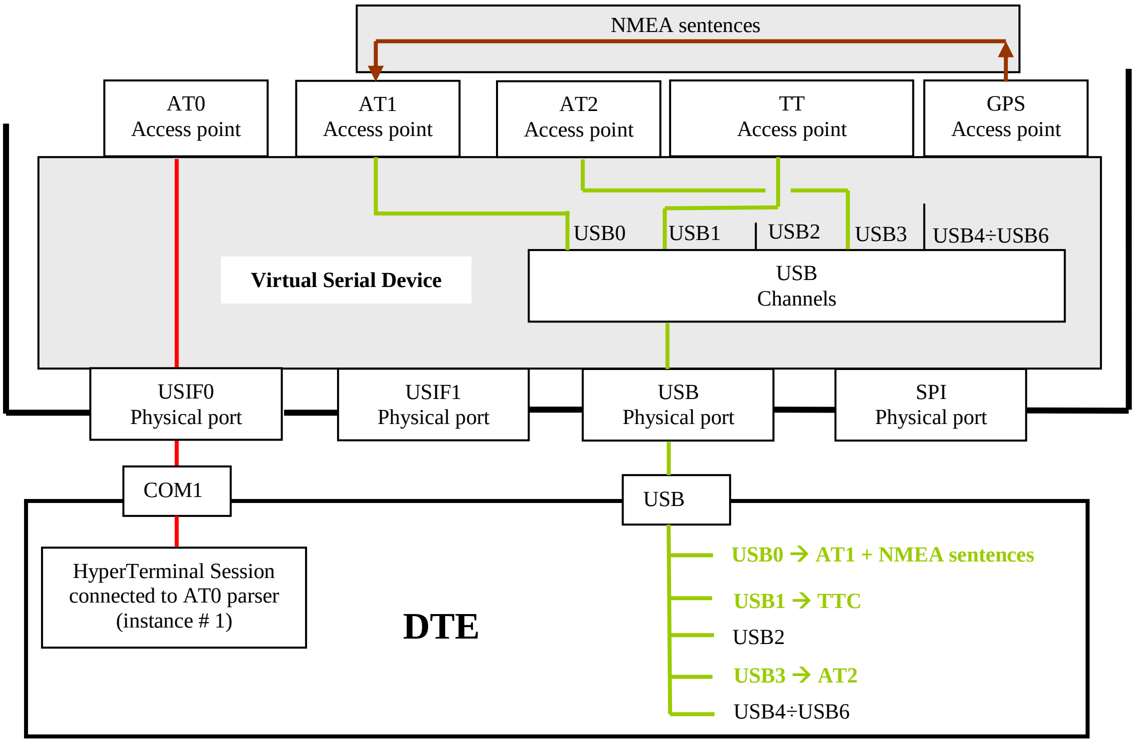


fig. 6: USB0 channel supports AT commands + NMEA sentences

NOTICE: NMEA sentences are sent on the USB channel used by the operator to enter the AT$GPSP and AT$GPSNMUN commands. On the same USB channel is still possible enter AT commands.

2.4. AT#PORTCFG Command

The AT#PORTCFG command manages several internal ports arrangements by means of its parameter value, refer to [2]. The tables and figures reported on the next pages show the various ports configurations that can be obtained changing the parameter value of the command and connecting the USB cable to the module. Use the following sequence to make active the entered AT#PORTCFG:

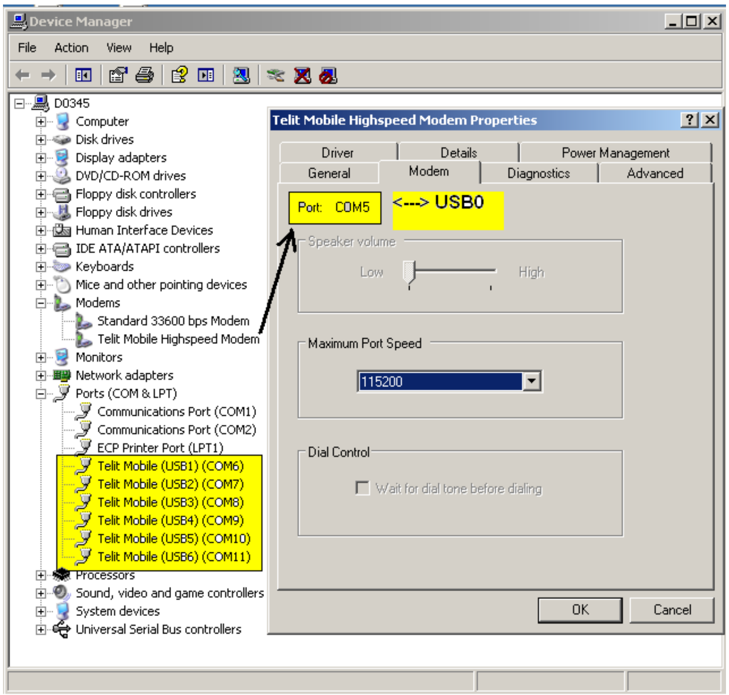
- Assume to start from the configuration showed on fig. 3, it is the factory setting: #PORTCFG is 0**Error! Reference source not found.**;

- Enter, for example, the AT#PORTCFG=1 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=1 command is executed and the ports arrangement of Tab. 9 is implemented.

Before dealing with the several AT#PORTCFG configurations it is advisable to see how the USBx channels are mapped into virtual COMx ports on the DTE side. The figure below shows an example of mapping. In general, it depends from the Windows PC configuration.


Device Manager
File
Action
View
Help
D0345
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Floppy disk controllers
Floppy disk drives
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Modems
Standard 33600 bps Modem
Telit Mobile Highspeed Modem
Monitors
Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
ECP Printer Port (LPT1)
Telit Mobile (USB1) (COM6)
Telit Mobile (USB2) (COM7)
Telit Mobile (USB3) (COM8)
Telit Mobile (USB4) (COM9)
Telit Mobile (USB5) (COM10)
Telit Mobile (USB6) (COM11)
Processors
Sound, video and game controllers
System devices
Universal Serial Bus controllers
Telit Mobile Highspeed Modem Properties
Driver
Details
Power Management
General
Modem
Diagnostics
Advanced
Speaker volume
Low
High
Dial Control
OK
Cancel

Port: COM5 <---> USB0

Maximum Port Speed: 115200

□ Wait for dial tone before dialing

fig. 7: USBx channels mapped into virtual COMx ports

From the fig. 7 is derived the mapping table showed below:

USB CHANNELS	VIRTUAL PORTS
USB0	VCOM5
USB1	VCOM6
USB2	VCOM7
USB3	VCOM8
USB4	VCOM9
USB5	VCOM10
USB6	VCOM11

Tab. 6: Mapping Table

Now, you are able to assign a VCOMx port to each USBx channels indicated on the following #PORTCFG tables.

AT#PORTCFG=0					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0	X				
USIF1					
SPI					

Tab. 7: #PORTCFG=0, no USB cable

AT#PORTCFG=0					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3			X		
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI					

Tab. 8: #PORTCFG=0, with USB cable

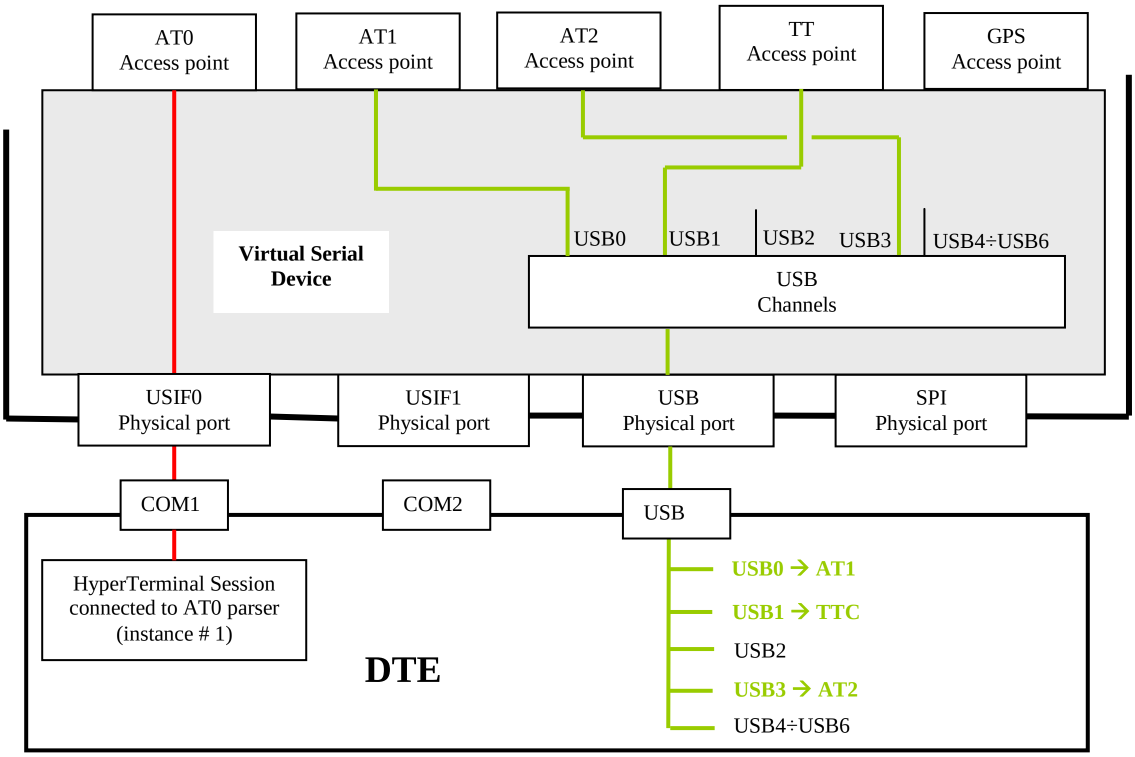


fig. 8: #PORTCFG=0 + USB cable connected

AT#PORTCFG=1					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0	X				
USIF1				TTC	
SPI					

Tab. 9: #PORTCFG=1, no USB cable

AT#PORTCFG=1					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1					
USB2					
USB3			X		
USB4					
USB5					
USB6					
USIF0	X				
USIF1				TTC	
SPI					

Tab. 10: #PORTCFG=1, with USB cable

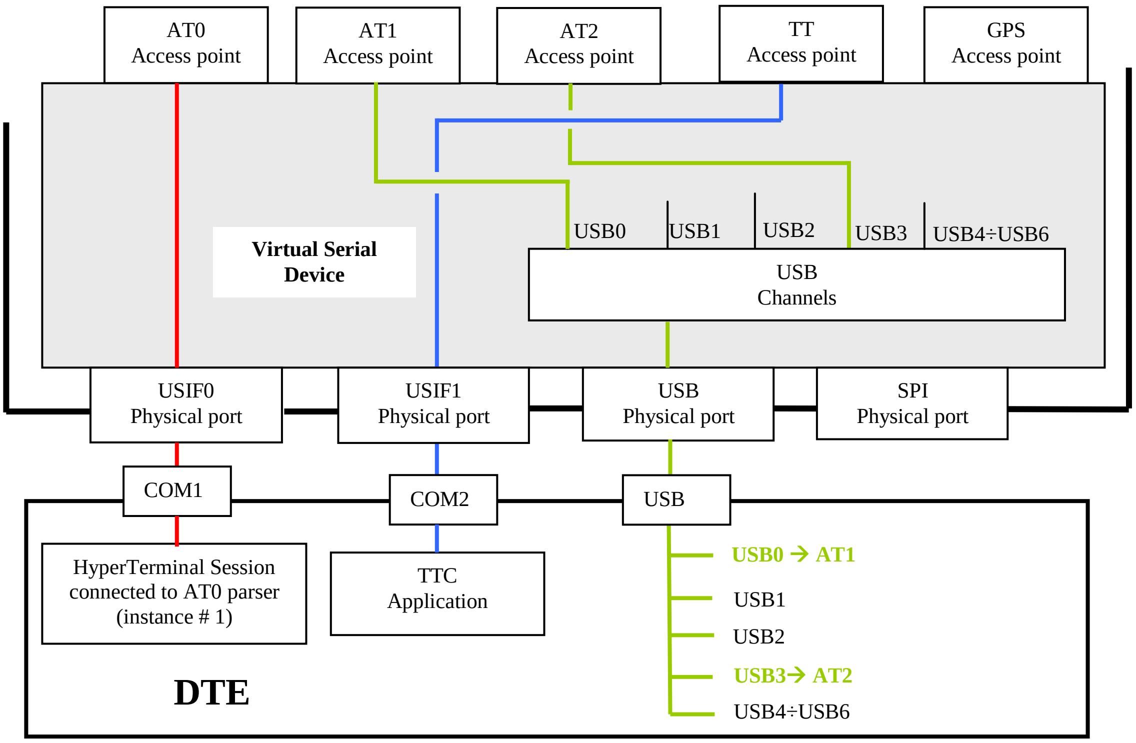


fig. 9: #PORTCFG=1 + USB cable connected

AT#PORTCFG=2					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0	X				
USIF1					
SPI			X		

Tab. 11: #PORTCFG=2, no USB cable

AT#PORTCFG=2					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI			X		

Tab. 12: #PORTCFG=2, with USB cable

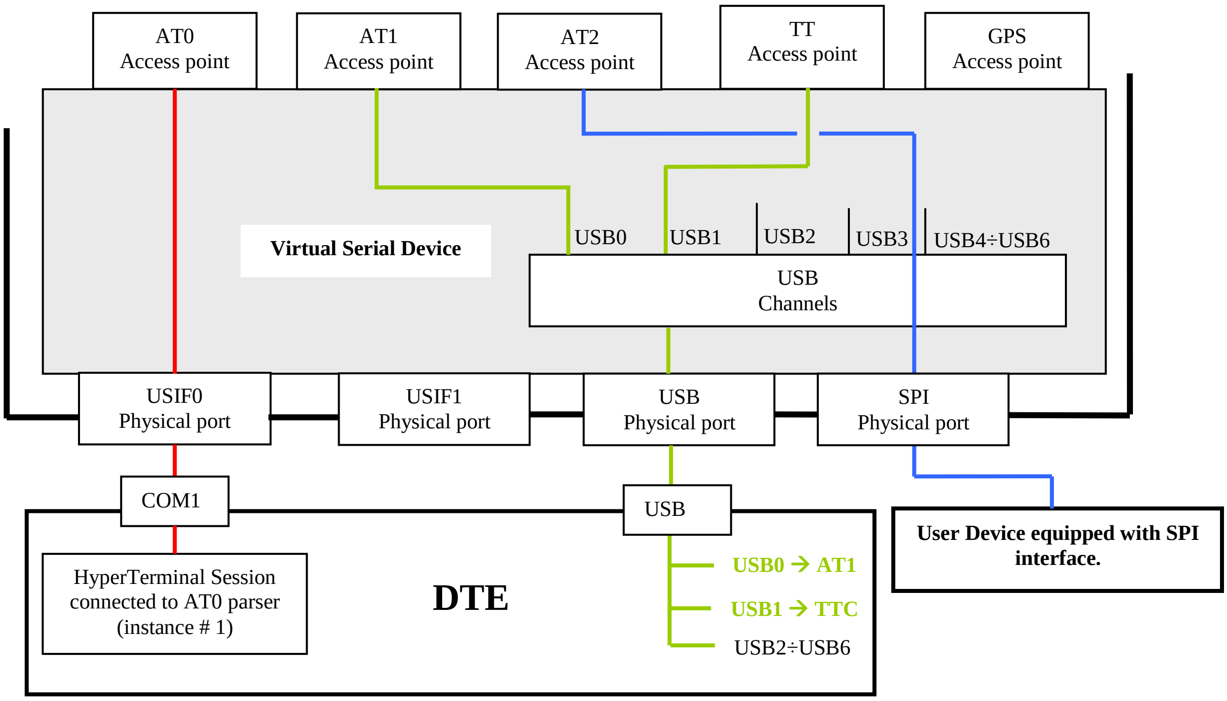


fig. 10: #PORTCFG=2 + USB cable connected

AT#PORTCFG=3					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0	🟥 X				
USIF1			X		
SPI					

Tab. 13: #PORTCFG=3, no USB cable

AT#PORTCFG=3					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0	X				
USIF1			X		
SPI					

Tab. 14: #PORTCFG=3, with USB cable

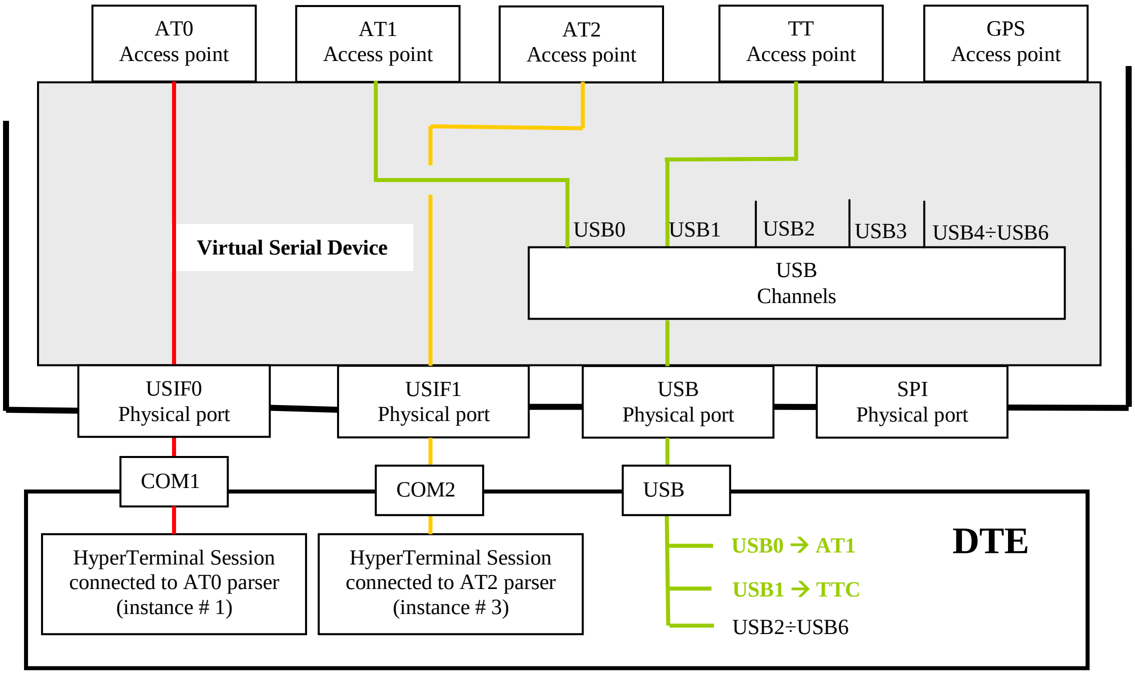


fig. 11: #PORTCFG=3 + USB cable connected

AT#PORTCFG=4					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0		X			
USIF1					
SPI			X		

Tab. 15: #PORTCFG=4, no USB cable

AT#PORTCFG=4					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0	X				
USB1				TTC	
USB2					
USB3			X		
USB4					
USB5					
USB6					
USIF0		X			
USIF1					
SPI					

Tab. 16: #PORTCFG=4, with USB cable

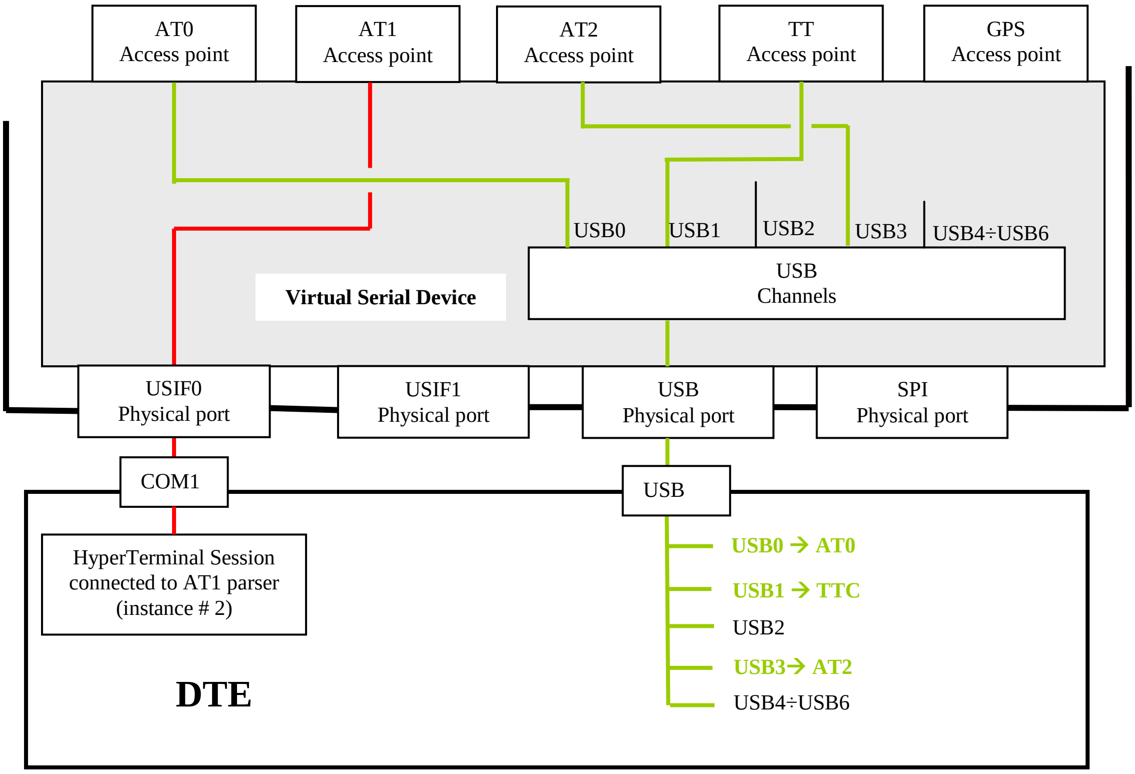


fig. 12: #PORTCFG=4 + USB cable connected

AT#PORTCFG=5					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0					
USIF1					
SPI			X		

Tab. 17: #PORTCFG=5, no USB cable

AT#PORTCFG=5					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3	X				
USB4					
USB5					
USB6					
USIF0					
USIF1					
SPI			X		

Tab. 18: #PORTCFG=5, with USB cable

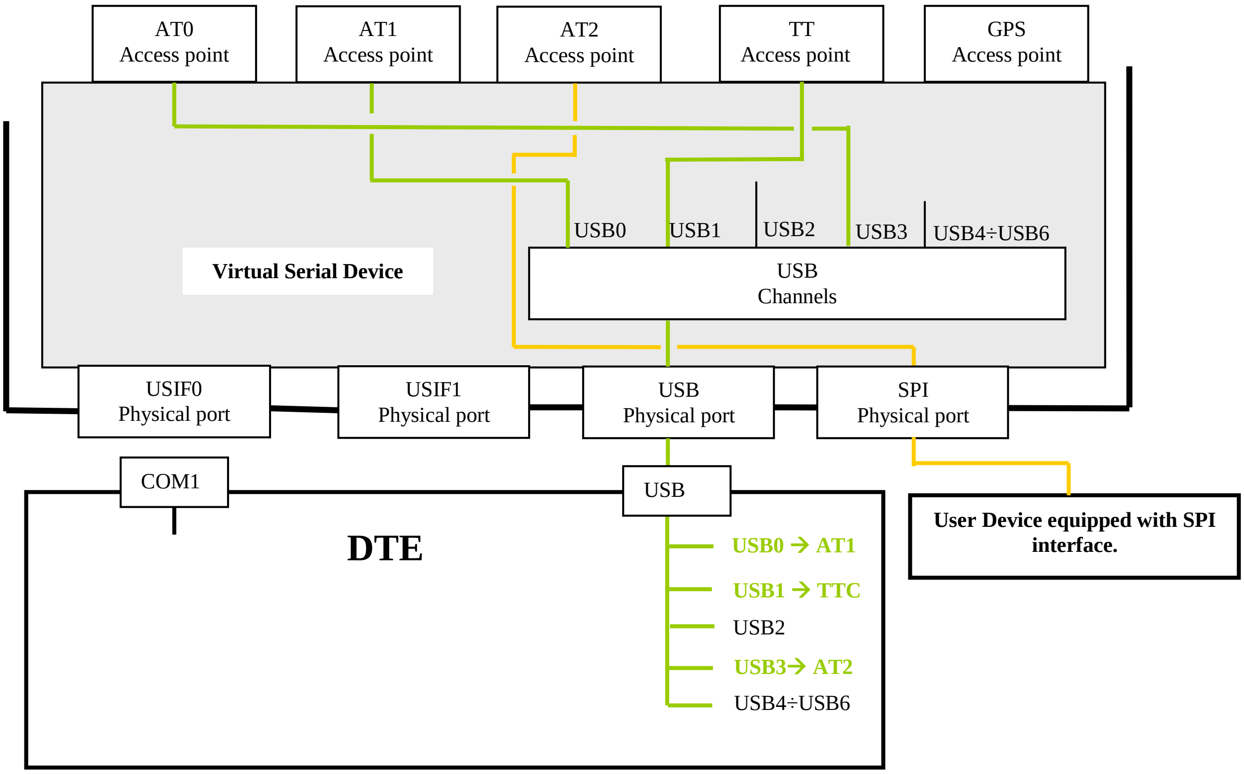


fig. 13: #PORTCFG=5 + USB cable connected

AT#PORTCFG=6					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0			X		
USIF1					
SPI	X				

Tab. 19: #PORTCFG=6, no USB cable

AT#PORTCFG=6					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0			X		
USIF1					
SPI	X				

Tab. 20: #PORTCFG=6, with USB cable

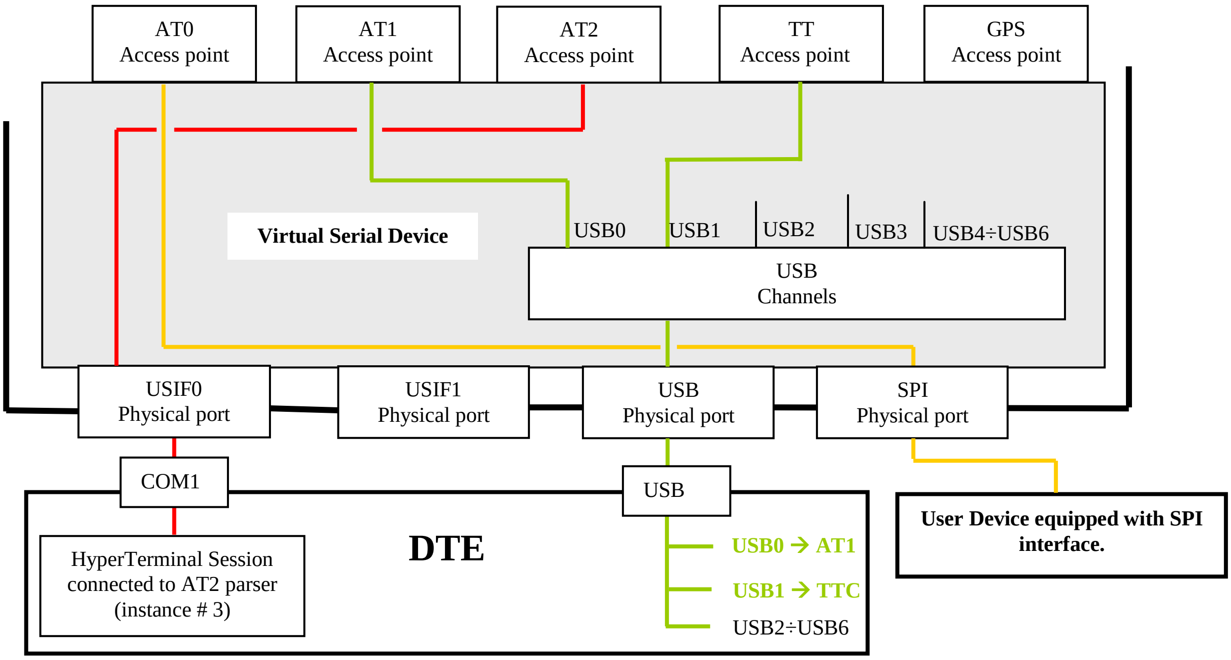


fig. 14: #PORTCFG=6 + USB cable connected

AT#PORTCFG=7					
	AT0	AT1	AT2	TT	GPS/NMEA
USB cable no connected					
USIF0	X				
USIF1					
SPI					

Tab. 21: #PORTCFG=7, no USB cable

AT#PORTCFG=7					
	AT0	AT1	AT2	TT	GPS/NMEA
USB0		X			
USB1				TTC	
USB2				3G	
USB3			X		
USB4					
USB5					
USB6					
USIF0	X				
USIF1					
SPI					

Tab. 22: #PORTCFG=7, with USB cable

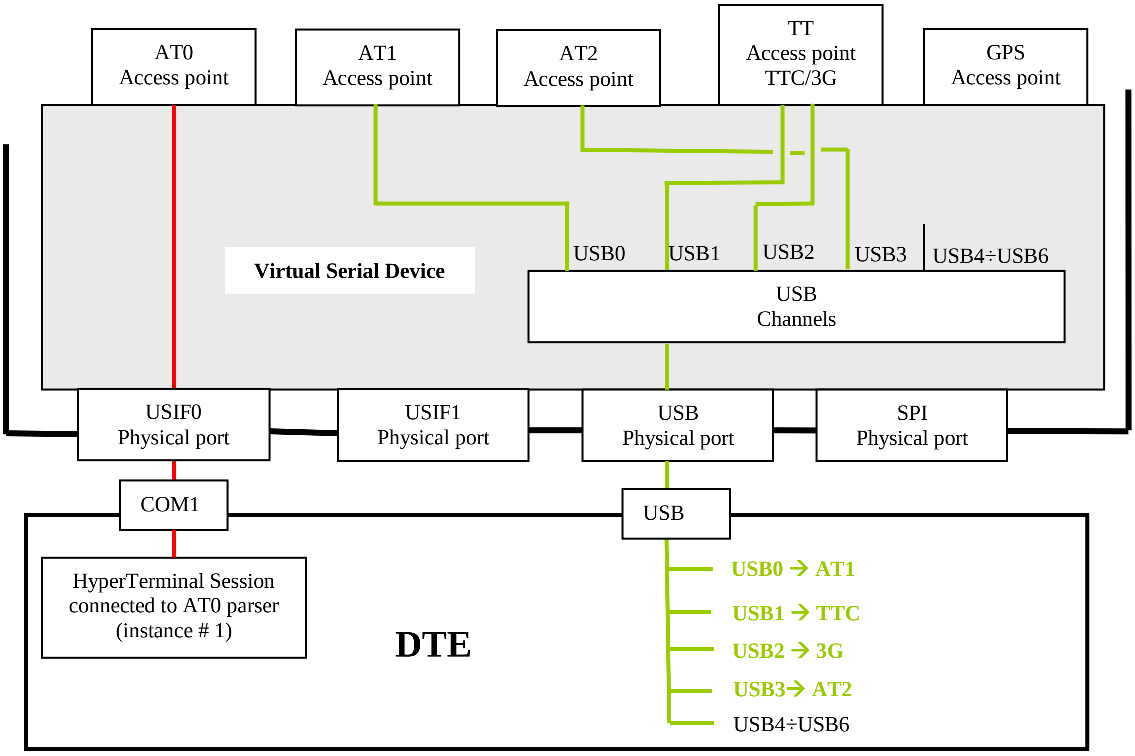


fig. 15: #PORTCFG=7 + USB cable connected

2.4.1. GPS/NMEA sentences on SPI

Assume that the module is configured as showed on Tab. 15. Now, enable GPS/NMEA sentences via AT$GPSP=1 and AT$GPSNMUN=1,… commands entered through SPI port. The commands are elaborated by AT2 parser and the module enters the configuration showed on fig. 16**Error! Reference source not found.**. Tab. 23**Error! Reference source not found.** summarizes the new internal ports configuration.

	AT0	AT1	AT2	TT	GPS/NMEA
USB0					
USB1					
USB2					
USB3					
USB4					
USB5					
USB6					
USIF0		X			
USIF1					
SPI			X		X

Tab. 23: SPI port supports NMEA sentences

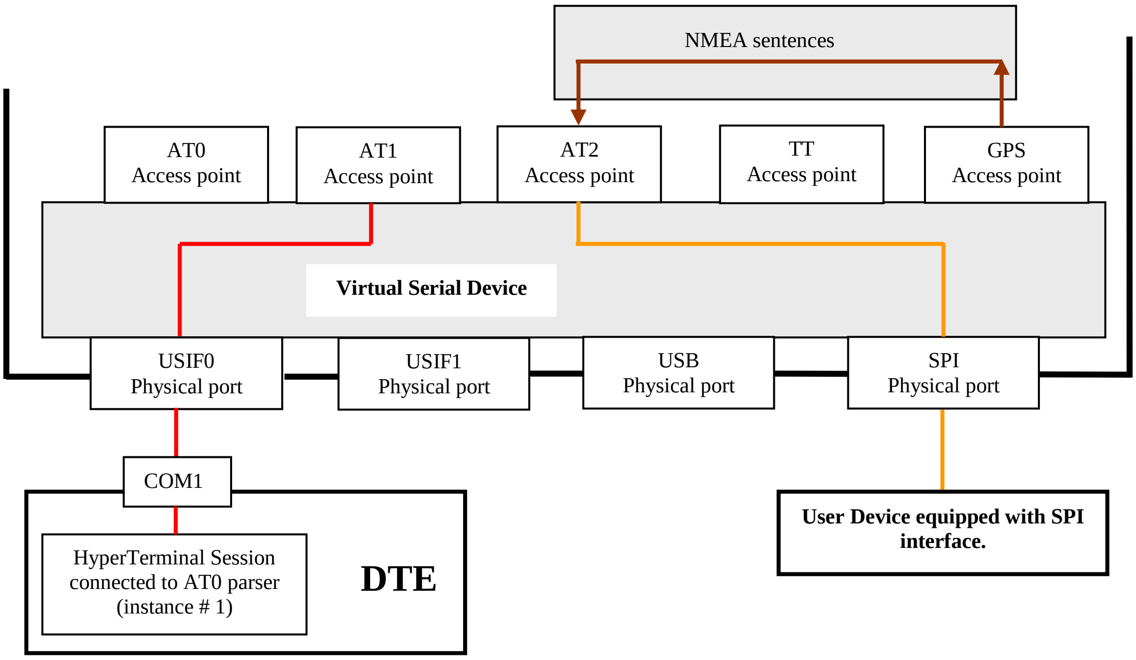


fig. 16: SPI port supports AT commands + NMEA sentences

NOTICE: NMEA sentences are sent on the physical port used by the operator to enter the AT$GPSP and AT$GPSNMUN commands. On the same physical port is still possible enter AT commands.

2.5. USIF0 & AT+CMUX Command

This chapter shows an example of ports arrangement supporting CMUX protocol.

Assume that the module is configured as indicated on fig. 3**Error! Reference source not found.**: #PORTCFG=0 no USB cable plugged in. In addition, suppose that the used DTE is a Windows PC and its device configuration is showed by fig. 17. Now, run on the DTE the TELIT Serial Port MUX application configured as showed on fig. 18, and connect it to COM1 physical port, refer to fig. 19. When the user starts an application (e.g. Hyper Terminal) connected to one of the four Virtual Ports, TELIT Serial Port MUX application sends automatically the AT+CMUX=0 command to the module and the CMUX protocol is activated.

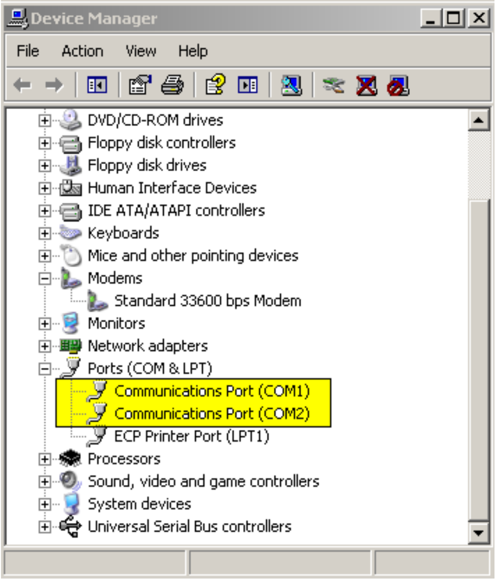


fig. 17: Physical COMx Ports

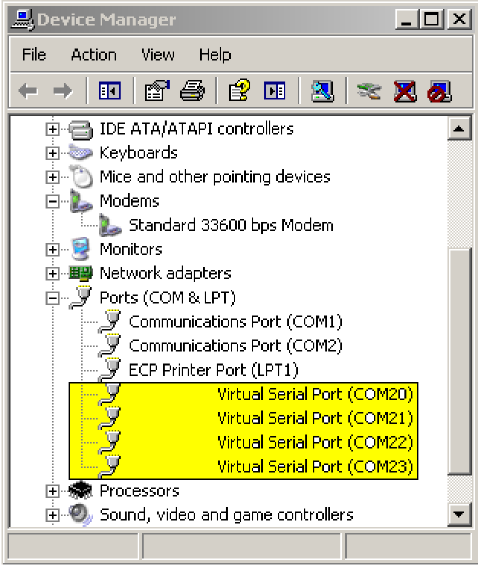

Device Manager
File
Action
View
Help
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Modems
Standard 33600 bps Modem
Monitors
Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
ECP Printer Port (LPT1)
Virtual Serial Port (COM20)
Virtual Serial Port (COM21)
Virtual Serial Port (COM22)
Virtual Serial Port (COM23)
Processors
Sound, video and game controllers

fig. 18: Virtual Serial Ports of MUX

TELIT Serial Port MUX application must be configured so that its virtual serial ports do not create conflict with the physical or virtual serial ports already present on the Windows PC (DTE). An example is showed by fig. 18. Tab. 24 summarizes the new configuration.

Module/DTE connection	VCOMx → VCx	AT0	AT1	AT2	TT	GPS/NMEA
USB / --						
USIF0 / COM1	VCOM20→VC1	X				
	VCOM21→VC2		X			
	VCOM22→VC3			X		
	VCOM23→VC4					
USIF1 / --						
SPI / --						

Tab. 24: Ports Arrangement with CMUX

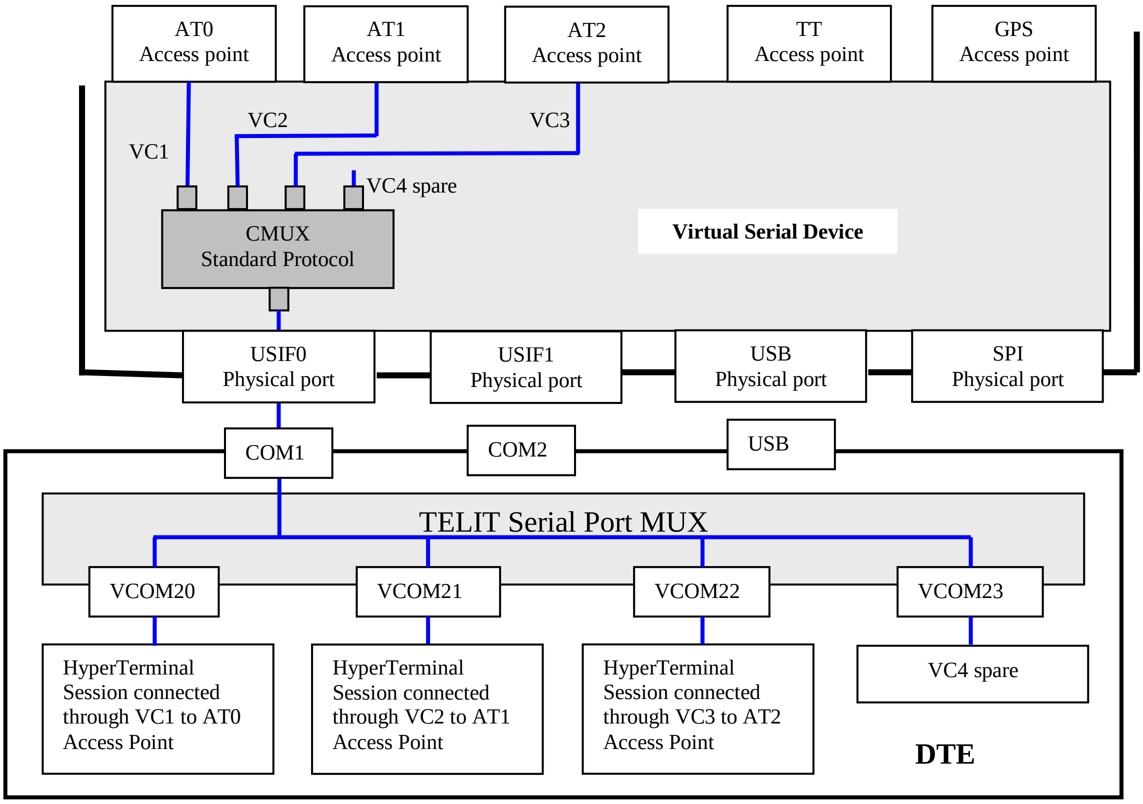


fig. 19: Ports Arrangement with CMUX

If TTC service is needed start from the configuration: #PORTCFG=1, no USB cable, see Tab. 9. Follow the steps stated above, refer to fig. 20. Tab. 25 summarizes the new configuration.

Module/DTE connection	VCOMx → VCx	AT0	AT1	AT2	TT	GPS/NMEA
USB / --						
USIF0 / COM1	VCOM20→VC1	X				
	VCOM21→VC2		X			
	VCOM22→VC3			X		
	VCOM23→VC4					
USIF1 / COM2					TTC	
SPI / --						

Tab. 25: Ports Arrangement with CMUX + TTC

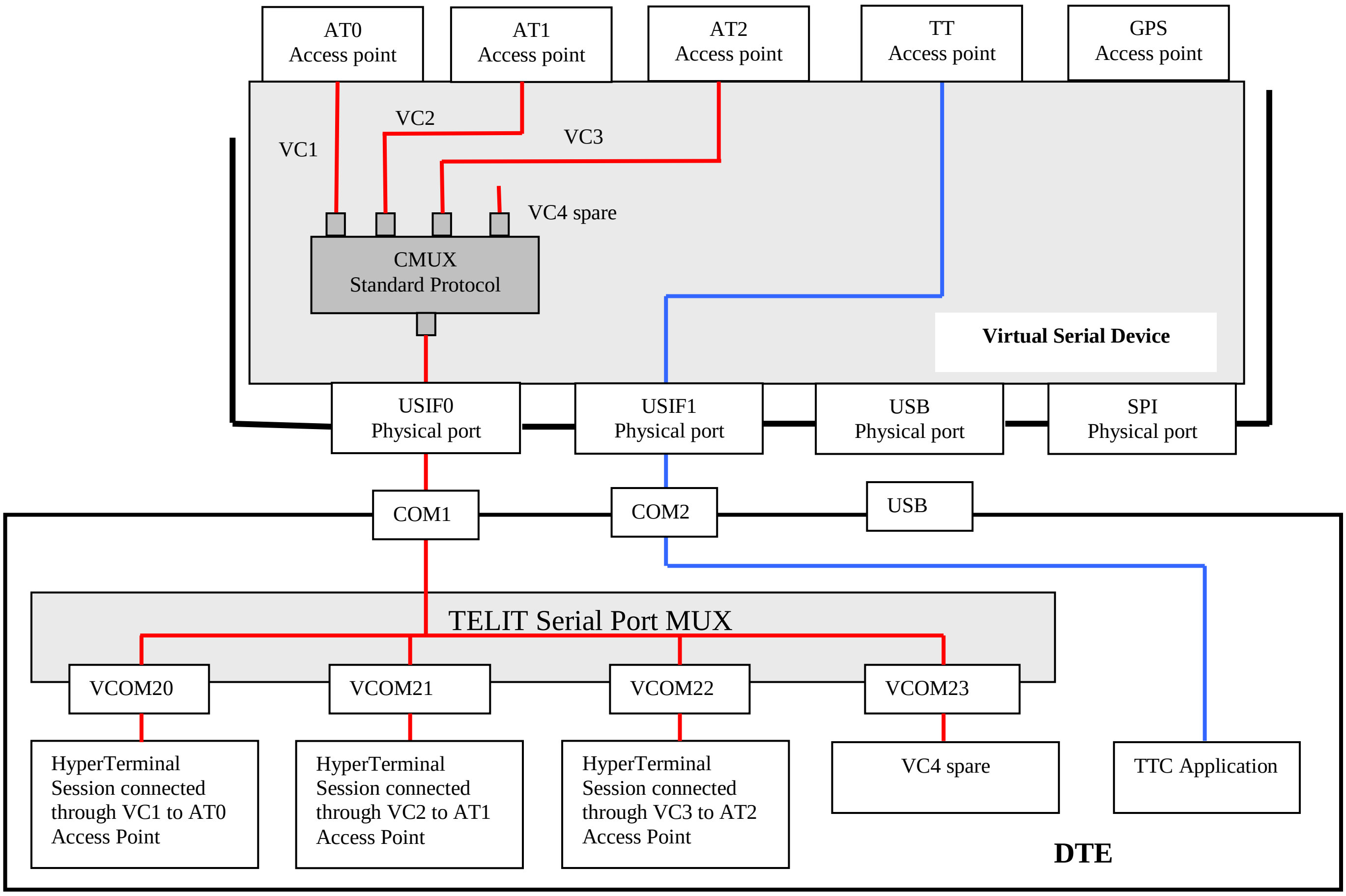


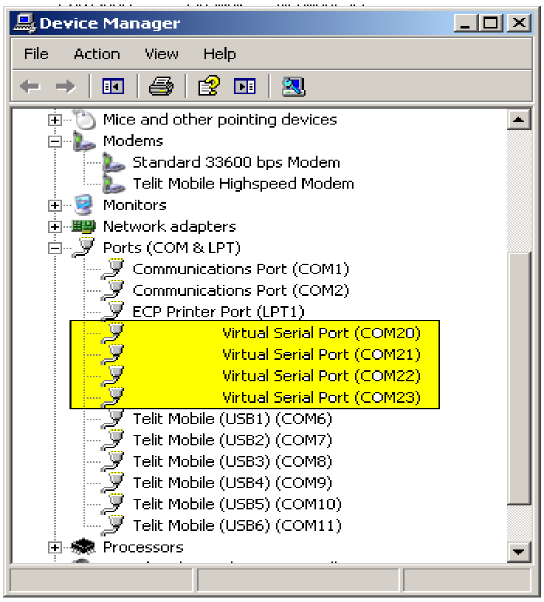
fig. 20: Ports Arrangement with CMUX + TTC

NOTICE: if you need to develop a Multiplexing Protocol running on a desired environment, e.g. a user micro-controller, refer to [1] to get detailed information about the protocol.

2.6. USB & AT+CMUX Command

This chapter shows an example of ports arrangement supporting USB channels, and CMUX protocol.

Assume that the module is configured as indicated on fig. 8**Error! Reference source not found.**: #PORTCFG=0 + USB cable plugged in. In addition, suppose that the used DTE is a Windows PC and its device configuration is showed by fig. 7. Now, run on the DTE the TELIT Serial Port MUX application configured as showed on fig. 21, and connect it to USB3 channel mapped into VCOM8 virtual port as showed on Tab. 6, refer to fig. 22. When the user starts an application (e.g. Hyper Terminal) connected to one of the four Virtual Ports, TELIT Serial Port MUX application sends automatically the AT+CMUX=0 command to the module and the CMUX protocol is activated.



Mice and other pointing devices
Modems
Standard 33600 bps Modem
Telit Mobile Highspeed Modem
Monitors
Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Communications Port (COM2)
ECP Printer Port (LPT1)
Virtual Serial Port (COM20)
Virtual Serial Port (COM21)
Virtual Serial Port (COM22)
Virtual Serial Port (COM23)
Telit Mobile (USB1) (COM6)
Telit Mobile (USB2) (COM7)
Telit Mobile (USB3) (COM8)
Telit Mobile (USB4) (COM9)
Telit Mobile (USB5) (COM10)
Telit Mobile (USB6) (COM11)
Processors

fig. 21: Virtual Serial ports of TELIT Serial Port MUX

TELIT Serial Port MUX application must be configured so that its virtual serial ports do not create conflict with the physical or virtual serial ports already present on the Windows PC (DTE). An example is showed by fig. 21.

It is worth noting that the AT0 (instance # 1) is disconnected from USIF0 and connected to USB3/VCOM8/VCOM20→VC1, the TTC stays on USB1. The table below summarizes the new ports arrangement.

Module/DTE connection	Channels	USBx → VCOM	VCOMx → VCx	AT0	AT1	AT2	TT	GPS/NMEA
USB / USB	USB0							
	USB1						TTC	
	USB2							
	USB3	VCOM8	VCOM20 → VC1	X				
			VCOM21 → VC2		X			
			VCOM22 → VC3			X		
			VCOM23 → VC4					
	USB4							
	USB5							
	USB6							
USIF0 / --								
USIF1 / --								
SPI / --								

Tab. 26: Ports Arrangement when CMUX is connected to USB3 channel

NOTICE: if you need to develop a Multiplexing Protocol running on a desired environment, e.g. a user micro-controller, refer to [1] to get detailed information about the protocol.

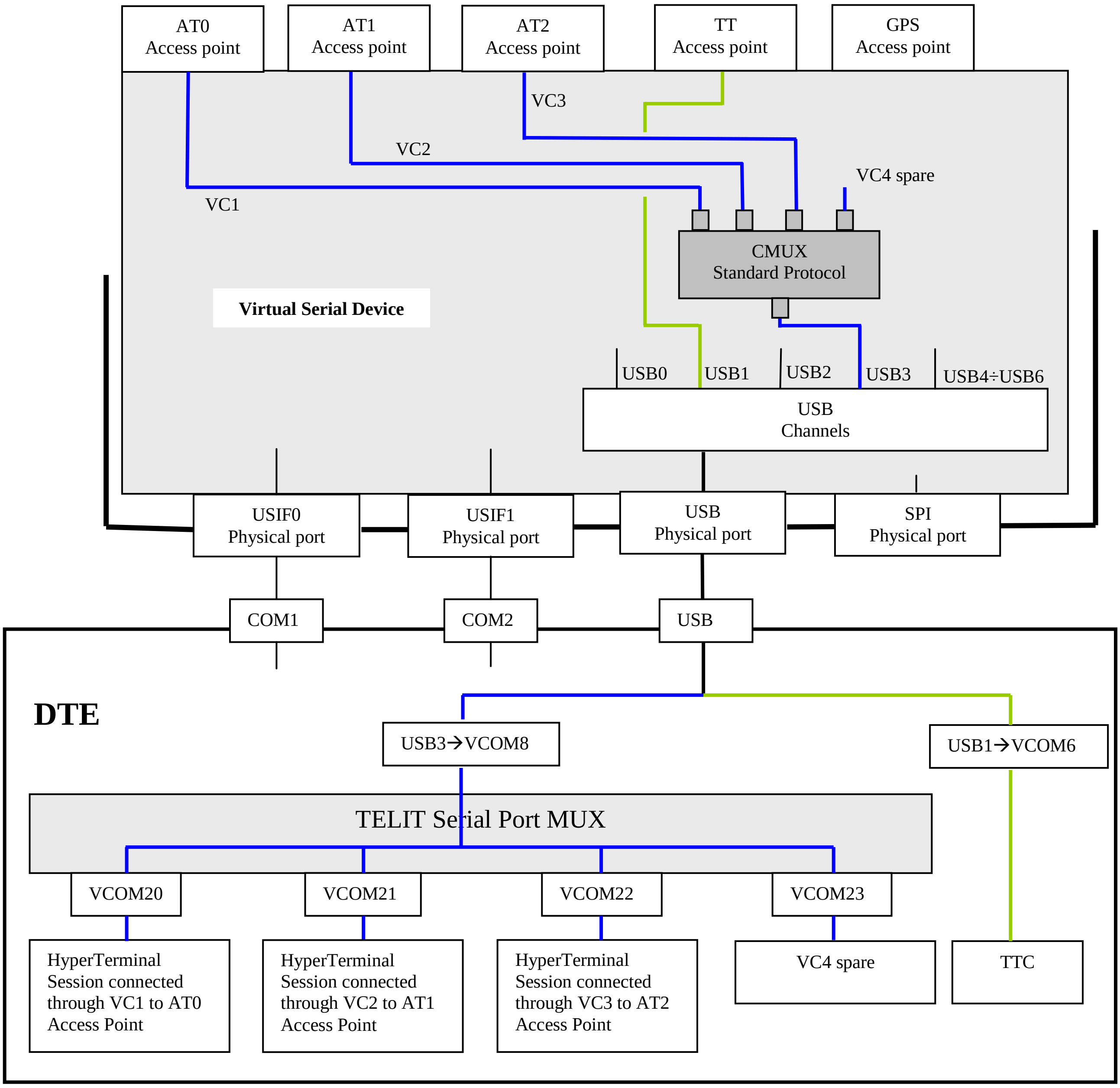


fig. 22: Ports Arrangement when CMUX is connected to USB3 channel

2.6.1. GPS/NMEA sentences on USB3-VC3

This chapter shows an example of ports arrangement supporting USB channels, CMUX protocol and NMEA sentences (GPS Service).

Assume that the module is configured as showed on fig. 22. Now, enable GPS/NMEA sentences via the AT$GPSP=1 and AT$GPSNMUN=1,… commands entered through USB3-VC3 port. The entered commands are elaborated by AT2 parser and after that, the module enters the configuration showed on fig. 23. Tab. 27 summarizes the new internal ports configuration.

Module/DTE connection	Channels	USBx → VCOM	VCOMx → VCx	AT0	AT1	AT2	TT	GPS/NMEA
USB / USB	USB0							
	USB1						TTC	
	USB2							
	USB3	VCOM8	VCOM20 → VC1	X				
			VCOM21 → VC2		X			
			VCOM22 → VC3			X		X
			VCOM23 → VC4					
	USB4							
	USB5							
	USB6							
USIF0 / --								
USIF1 / --								
SPI / --								

Tab. 27: USB3-VC3 port supports AT commands + NMEA sentences

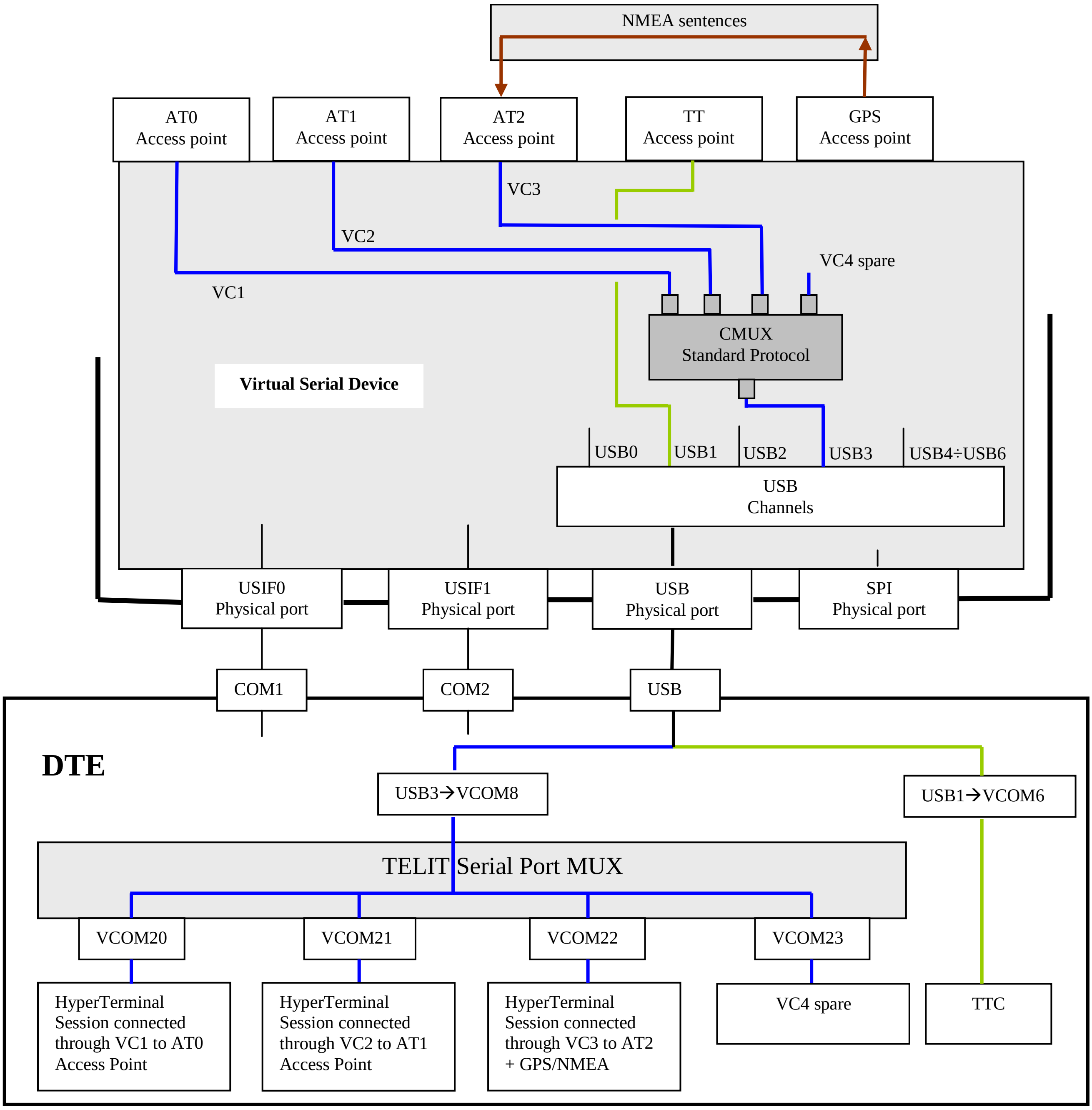


fig. 23: USB3-VC3 port supports AT commands + NMEA sentences

NOTICE: NMEA sentences are sent on the USB channel/VCx used by the operator to enter the AT$GPSP and AT$GPSNMUN commands. On the same channel is still possible enter AT commands.

3. The Winning Ports Configuration

Having at disposal several ports configurations, the following question arises when the user sets different configurations in sequence: is the last configuration affected by the previous ones?

There are two methods to change module ports arrangement without turn off/on the module:

- Plug in/out the USB cable;
- Enter the AT+CMUX=0[7] command.

NOTICE: to execute the entered AT#PORTCFG command and get the desired ports arrangement it is needed to turn off/on the module.

The modules provide the following policy: regardless of the actions or commands sequence that user has used to set the module into the desired ports configuration the module must always enter the last requested configuration. Follow two examples.

Example 1

Module: let’s suppose that its ports configuration is showed on fig. 3.

User action: the user runs on the Windows PC the TELIT Serial Port MUX application so configured: Module Main Port: COM1; Virtual Ports: COM20÷COM23.

PC: it 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 showed on fig. 19.

User action: now, the user connects USB cable.

Module: it enters the configuration showed on fig. 5.

PC: it provides seven new virtual “COM” logically connected to the seven USB channels. The CMUX protocol is disabled and the TELIT Serial Port MUX

[7] TELIT Serial Port MUX application automatically sends the AT+CMUX=0 command to the module, see chapters 2.5, 2.6.

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: now, the user disconnects USB cable.

Module: it enters again the configuration showed on fig. 3.

Example 2

Module: let’s suppose that its configuration is showed on fig. 3.

User action: the user connects USB cable.

Module: in accordance with the user action, the module enters the configuration showed on fig. 5.

PC: it provides seven virtual “COM” required by USB drivers to logically connect the seven USBX channels.

User action: the user runs on the Windows PC the TELIT Serial Port MUX application so configured: Module Main Port: USB3→VCOM8; Virtual Ports: VCOM20 ÷ VCOM23.

PC: it 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 showed on fig. 22.

User action: now, the user disconnects USB cable.

Module: it enters the configuration showed on fig. 3.

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

From the above mentioned examples it may be inferred that the last required port configuration overrides the previous one.

4. Services

4.1. Python

The HE910 Family modules provide the Python programming language to offer to the user a tool to develop control scripts in accordance with its communication and hardware needs, see [4]. As showed on fig. 24 the VSD provides two access points called VHWDTE0 and VHW DTE1. MDM and MDM2 Python modules are logically connected respectively to VHW DTE0 and VHWDTE1 access points.

Let’s assume that the factory setting[8] of the module is not changed and the USB cable is not plugged in. Now, power on the module: the factory arrangement of the internal connections among physical ports and “access points” is depicted on fig. 3. Tab. 2 summarizes the factory arrangement. When the Python script runs the Python instruction ***import MDM***[9], the VSD disconnects the USIF0/AT0 logical connection and establishes the logical connection VHWDTE0/AT0; consequently the Python script can access AT0 parser. In the same way, ***import MDM2*** instruction forces the VSD to establish the logical connection VHWDTE1/AT1. From fig. 24 it is possible infer that USIF0 is disconnected and unutilized from external module side.

Python script can run another Python software module to use the USIF0 port using the instruction ***import SER***. The fig. 25 shows the new connection: through the physical port USIF0 it is possible to be connected with the Python script.

The three Python software modules (MDM, MDM2 and SER) use three independent resources: USIF0 physical port, AT0, and AT1 Access Points. No resources contention can arise among them. As a rule, we can say that the MDM, MDM2 and SER instructions steal the above mentioned resources regardless their current owner.

[8] AT#PORTCFG=0, refer to Chapter 2.4.
[9] It is assumed that the reader is familiar with Python language.

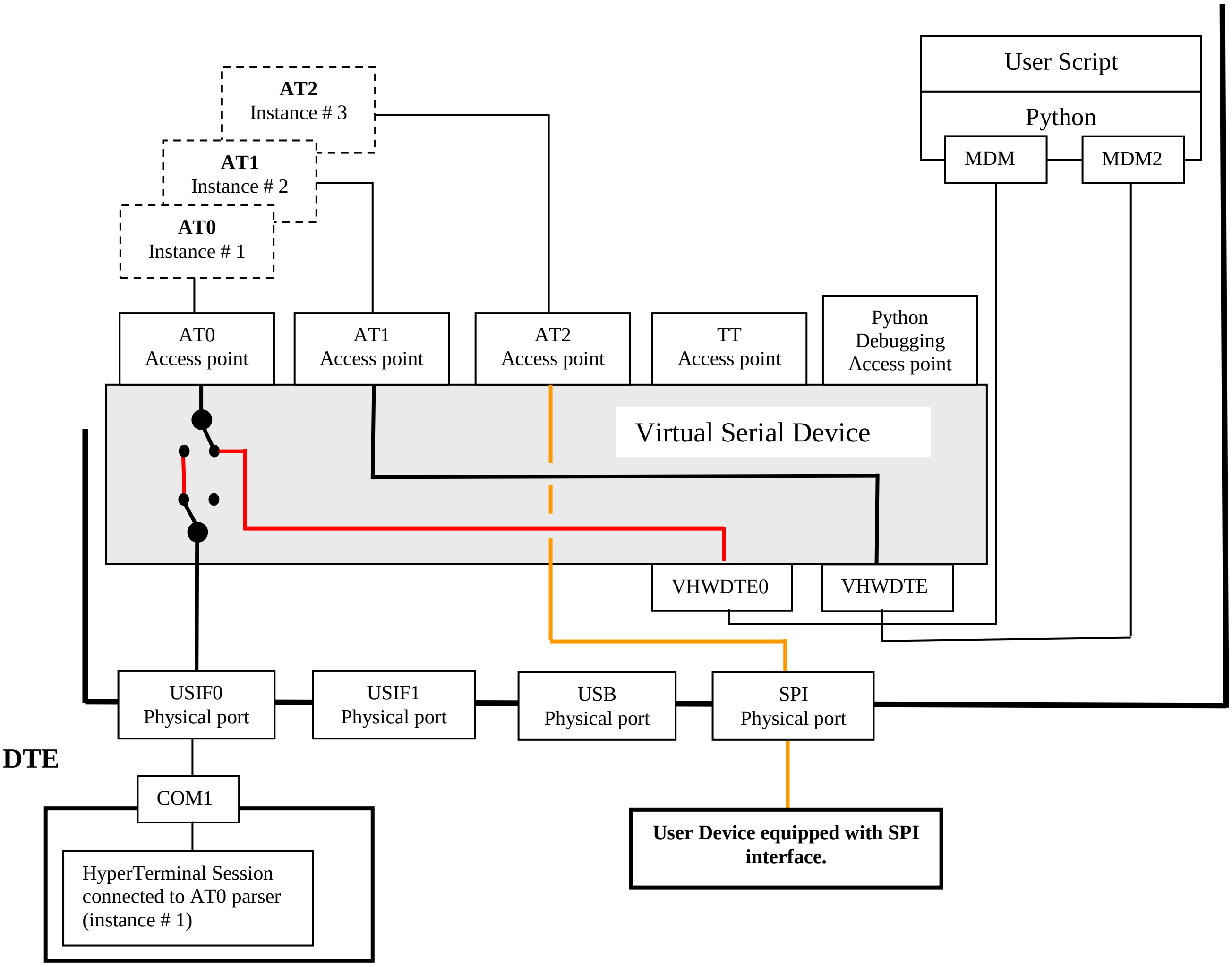


fig. 24: Python & MDM, MDM2 modules

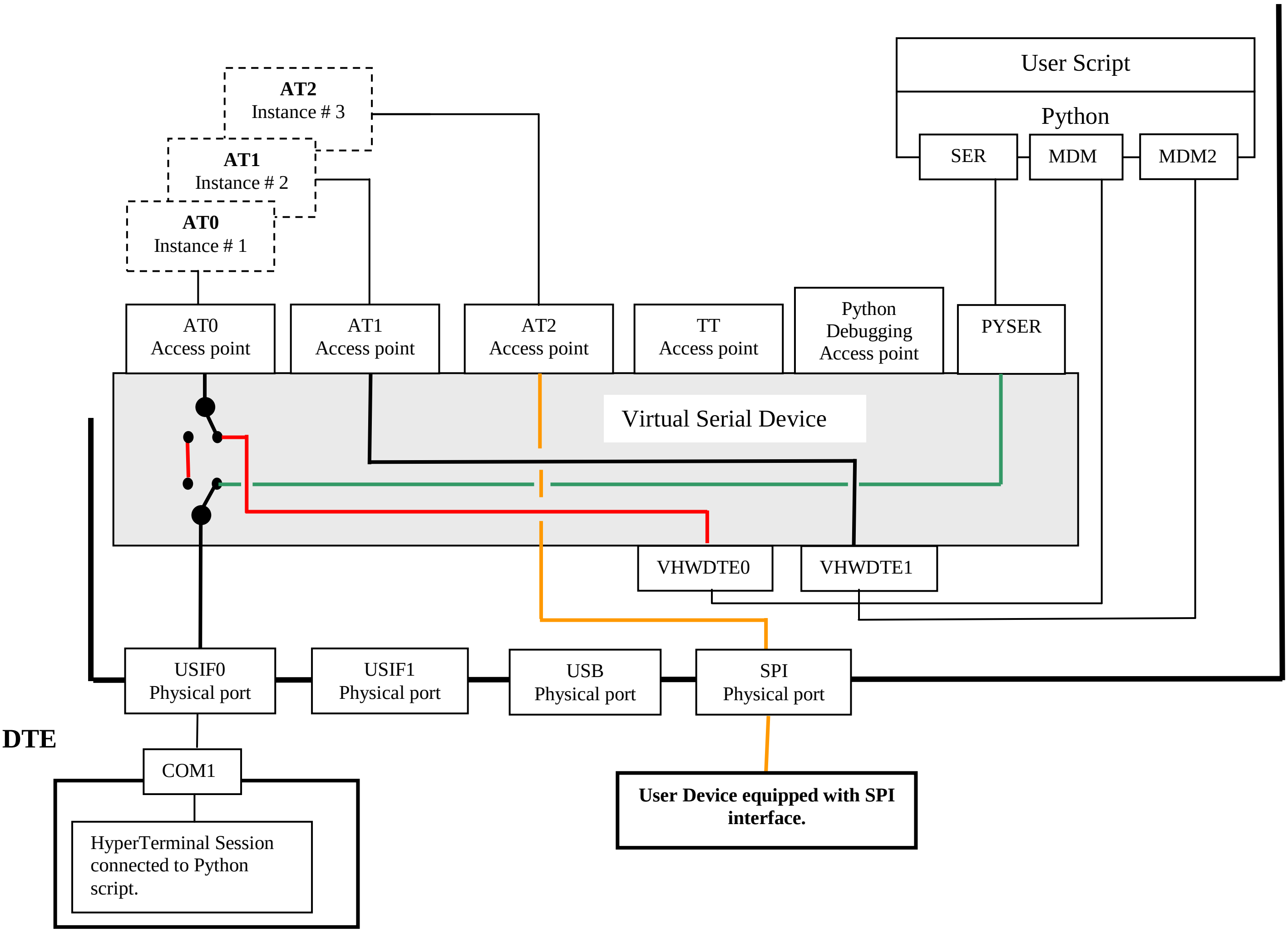


fig. 25: Python & MDM, MDM2, SER modules

4.1.1. Python script debugging

Assume that the user needs to debug a new Python script. To perform the debug session, the user forces the module into #PORTCFG=3 configuration, refer to Tab. 13. Now, suppose that the Python script runs: ***import MDM, import MDM2, import SER*** and ***print*** instructions. The figure below sketches the actions results of the first tree instructions, plus the action result of the last one that makes available print messages on the generic hyper terminal connected to USIF1 port.

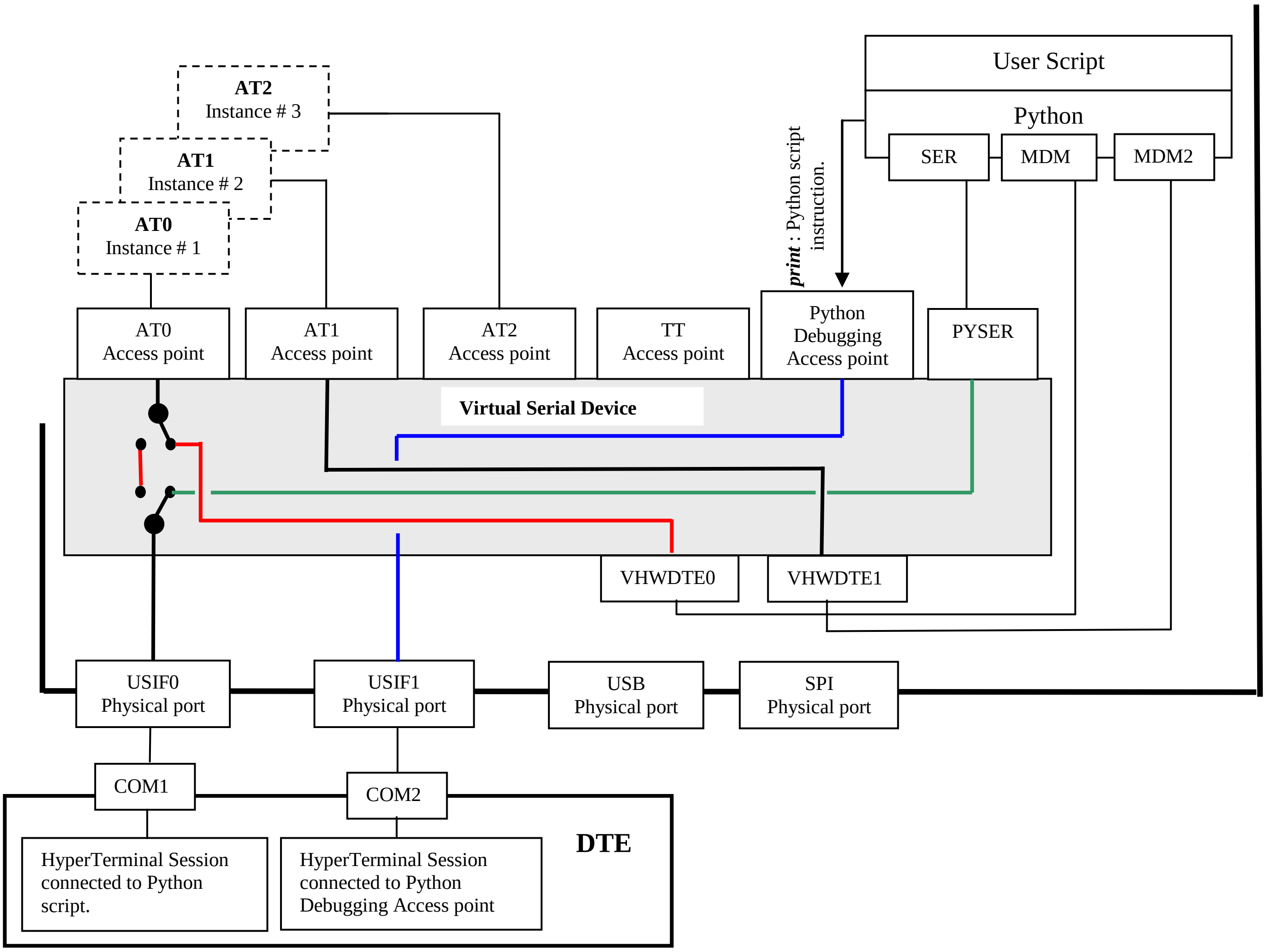


fig. 26: Python & MDM, MDM2, SER and print modules