

Python Debug Tools for GPS Modules

05/09/07



1.1 Debug Python script

The debug of the active Python script can be done both on the emulated environment of the [Telit Python Package](#) (refer to its documentation) or directly on the target with the second serial port pin EMMI TX (actually a not translated RS232 serial port as the RXD pin).

Connect to the module serial port EMMI TX at 115200 8-N-1 with hardware flow control active. Now you can see all Python outputs to stdout and stderr:

- Python information messages (for example the version);
- Python error information;
- Results of all Python “print” statements.

The [Telit GM862-GPS](#) and [GE863-GPS](#) have the second serial port pin EMMI TX used for continuous direct output of GPS NMEA sentences that’s why there is another procedure to follow for debugging of the Telit GPS modules. There are two ways to perform direct debugging: activate SSC port or use CMUX.

1.1.1 Debug Python script on GPS modules using SSC bus

SSC (Serial Synchronous Controller) port can be configured to be compatible to the SPI Interface, available via 4 GPIO pins. In this case the Python debug data will be read from the USB port placed on the EVK2.

Note: for the direct debug of GPS modules a software version starting from 7.02.001 is needed

1.1.1.1 Installation of the drivers

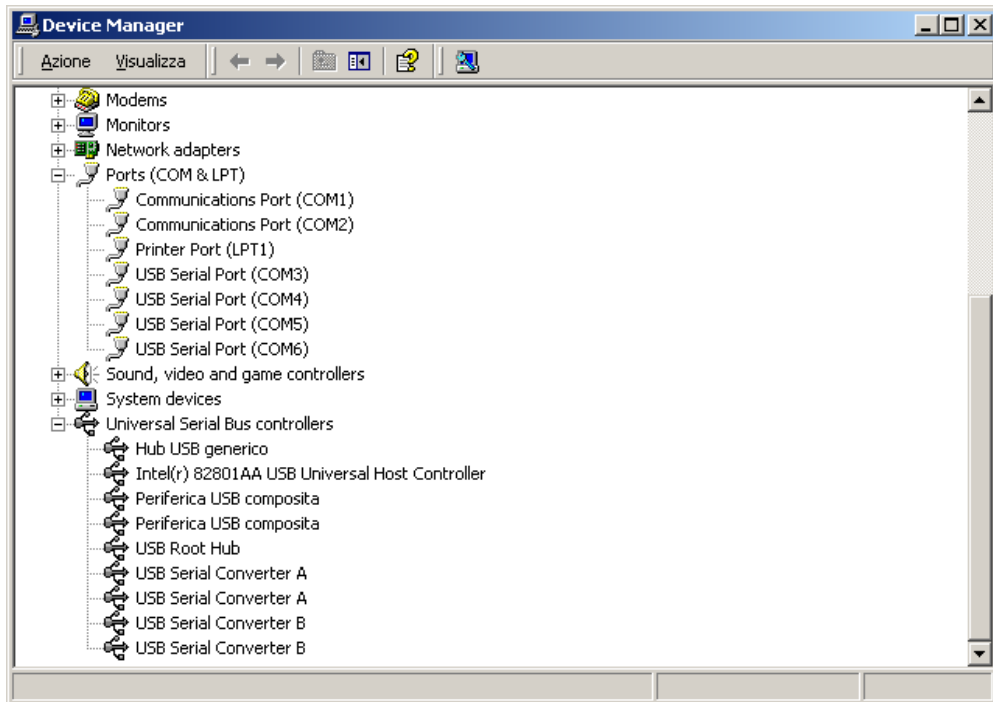
Before starting the process of debug the drivers should be installed in the following way:

- Download the FTDI drivers and the installation guide in order to use the USB port placed on the EVK2 (<http://www.ftdichip.com/Drivers/D2XX.htm>)
- Save the drivers (unzipped) on the PC
- After connecting USB cable with PC and USB port placed on the EVK2 (that has been powered on): the installation procedure should start, according to the installation guide instructions
- When the installation is concluded you will have four new COM ports (see Control Panel – System – Hardware – Device Manager) and one not visible SSC port



Python Debug Tools for GPS Modules

05/09/07



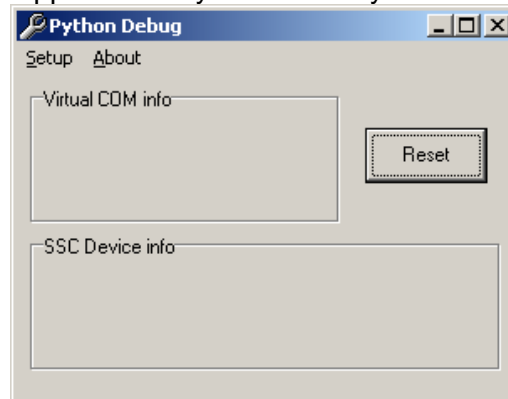
- close any application controlling the serial ports and install the “Python Debug” application (<http://www.telit.com> download zone)

NOTE: if an error messages appears during the installation, it will be necessary to close any application controlling the serial ports

- the tool is identified by the following Icon on your desktop



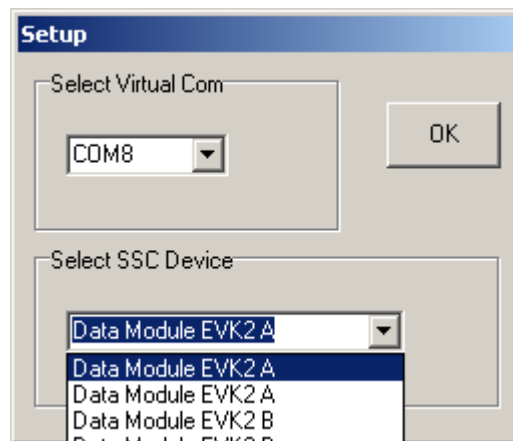
- the following box should appear when you run the Pythondebug.exe for the first time:



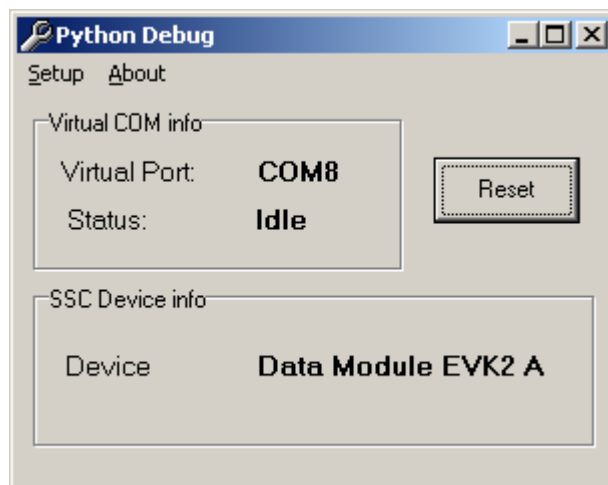
Python Debug Tools for GPS Modules

05/09/07

- Select the Setup option
- Then select a Virtual COM, different from the other COM ports preferably (“COM8” in the figure), and associate to it the first SSC device that appearing in the list (“Data Module EVK2A” in the figure),



- the following figure should appear:



NOTE: If the PC uses the EVK2 RS232 upper port (ASC0) to send AT commands, remember to put all jumpers to set RS232 mode. This will not affect reading of Python debug data from the USB port



1.1.1.2 Debugging process

After the successful installation of the drivers process of direct debugging of the Telit GPS modules can start. The steps are the following:

- Switch on the module and activate the SSC output with the following AT command: AT#SSCTRACE=1
- Download and enable your Python script, then power OFF the module.
- to be sure that DTR input to the module is HIGH disconnect the RS232 cable from the module side (i.e. RS232 DTR on the modem serial port is LOW);
- check if you have the USB cable connected between the USB port of the PC and the USB port placed on the EVK2
- every time you power ON the module you have to click on the *Reset* button in the *Python Debug* application (necessary to reactivate the association between the virtual Com port and the SSC device)
- Run a terminal emulator application (e.g. Hyper Terminal) to trace the activity of the Python script, with the following setting:

connected COM	virtual Port set in Python Debug (COM8 in the example)
Bit rate	115 200
Data bits	8
Parity	No parity
Stop bit	1
Flow control	Hardware

- Power ON the module and you should see the script starting and the debug info appearing on the terminal emulator window.
- If the debug strings do not appear on the screen: power OFF the module again, reset the *Python Debug* application or exit from the Python debug application, check again if USB cable is correctly connected.



1.1.2 Debug Python script on GPS modules using CMUX

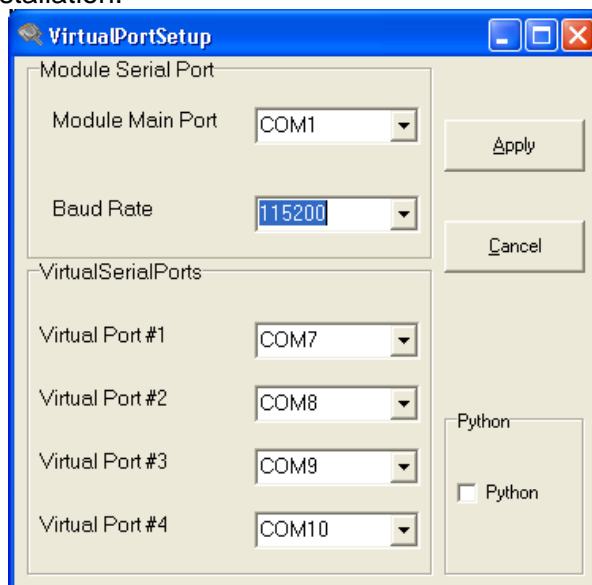
CMUX (Converter-Multiplexer) is a multiplexing protocol implemented in the Telit module that can be used to send data, SMS, fax, TCP data. The Multiplexer mode enables one serial interface to transmit data to four different customer applications. This is achieved by providing four virtual channels using a Multiplexer (Mux).

With activating of the CMUX feature debugging data can be received on the serial ASC0 port mounted on EVK2.

NOTE: for the direct debug of GPS modules a software version starting from 7.02.X01 is needed.

1.1.2.1 Installation

- Install the *Telit Serial Port Mux* ver 1.09 application on your PC. A box similar to this will appear at the end of installation:



- Select the baud rate and then click on the Apply button

1.1.2.2 Debugging process

NOTE: If the PC uses the EVK2 RS232 upper port (ASC0) to send AT commands, remember to put all jumpers to set RS232 mode.

- Switch ON the module and with a terminal emulator (e.g. *Hyper Terminal*) and send the following commands to the module:



Python Debug Tools for GPS Modules

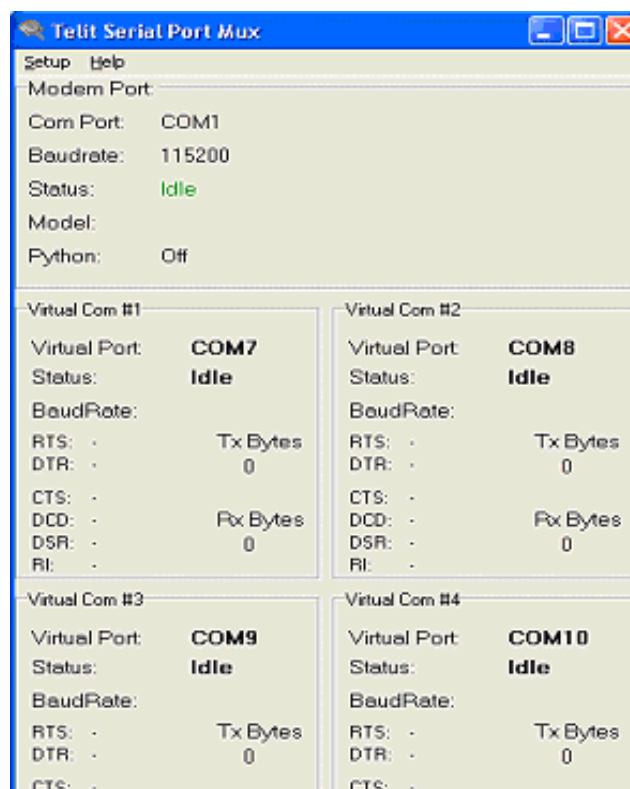
05/09/07

AT#SSCTRACE=0 disabled SSC output

AT#CMUXSCR=1,<bitrate> activated the CMUX feature on the module; put the desired bit rate (e.g. 115200)

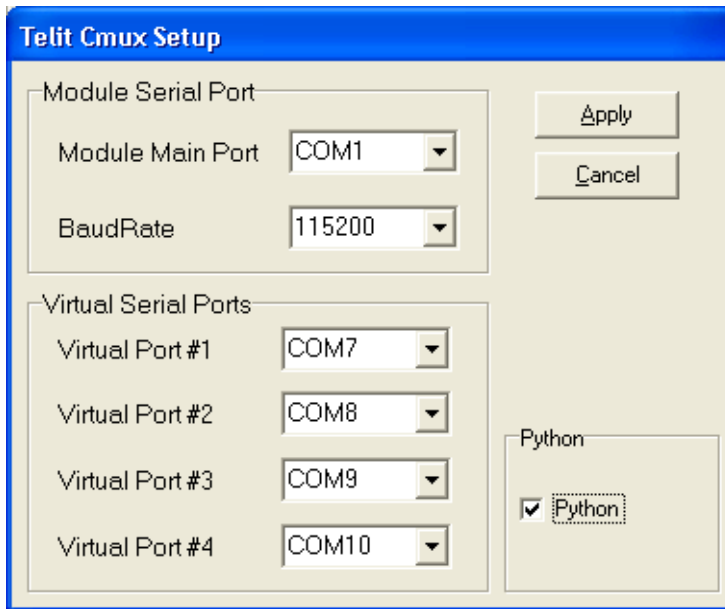
AT#STARTMODESCR=1,10 module waits for minimum 10 seconds (recommended value; can be changed) and if there is no AT commands sent in this period (except AT<Enter>) start the enabled Python script, regardless of the DTR status (low or high).

- Download and enable your Python script, then power OFF the module.
- Close any application controlling the serial ports (e.g. *Hyper Terminal*)
- Run the *Telit Serial Port Mux* ; a figure similar to the one below will appear:



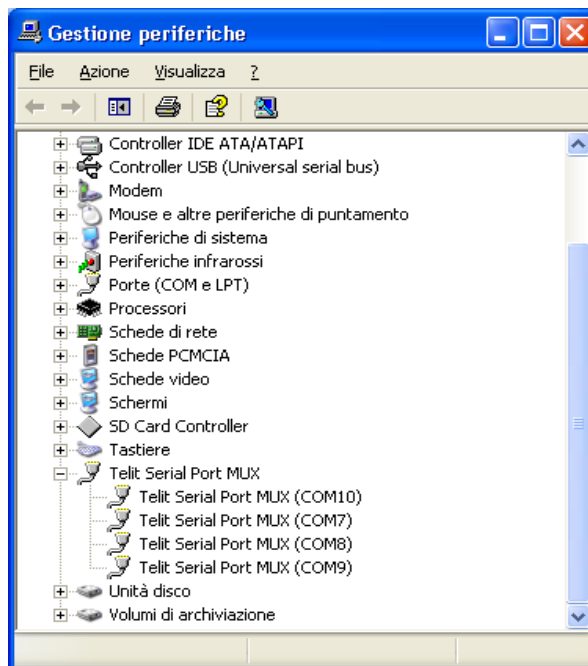
Python Debug Tools for GPS Modules
05/09/07

Select the following Setup option:



Set the *Module Main Port* as the real COM port you have available (e.g. COM1 in the figure), check the Python box and then select the Apply button.

- After this step, you will have 4 new *Telit Serial Port Mux* ports (see *Control Panel – System – Hardware – Device Manager*) as in the figure below:



Python Debug Tools for GPS Modules

05/09/07

- Run a terminal emulator application (e.g. *Hyper Terminal*) to trace the activity of the Python script, with the following setting:

connected COM	virtual port #4 set in Telit CMUX window (COM10 in the figure)
Bit rate	115 200
Data bits	8
Parity	No parity
Stop bit	1
Flow control	Hardware

In the *Telit Serial Port Mux* window, “Status:” of the Virtual Port#4, after establishing connection in *Hyper Terminal*, will change from *Idle* to *Opened*

- Power on the module and wait for at least 10 seconds without to send any AT command (except AT<Enter>);
In the *Telit Serial Port Mux* window, “Status:” of the *Modem Port:* will change in the following way (before 10 seconds expired):
Idle → cycle between Connecting and Error → Connected

After 10 seconds you should see the script starting and the debug info appearing on the terminal emulator window.

- If an ERROR messages appears in the Virtual Port #1,2,3,4 boxes, close any application controlling the serial ports and then restart the Telit CMUX application. If this procedure is not sufficient to avoid ERROR message, reset the PC, run again *Telit Serial Port Mux* with the same settings and repeat the procedure as described above.
- If you need to debug the same Python application again, then:
 - Disconnect the terminal emulator application (eg. *Hyper Terminal*) from the Virtual Port#4 (in this case COM10)
 - “Status:” of the Virtual Port#4 in the *Telit Serial Port Mux* window, should change from Opened to Idle
 - Switch off the module
 - Connect the terminal emulator application to Virtual Port#4 (in this case COM10)
 - “Status:” of the Virtual Port#4 in the *Telit Serial Port Mux* window¹, should change from Idle to Opened
 - Switch on the module and wait for the “Status:” of the *Modem Port* in the *Telit Serial Port Mux* window to go connected

¹ If the *Telit Serial Port Mux* application seems to be frozen, please consider that it becomes active after the module is switched on.

