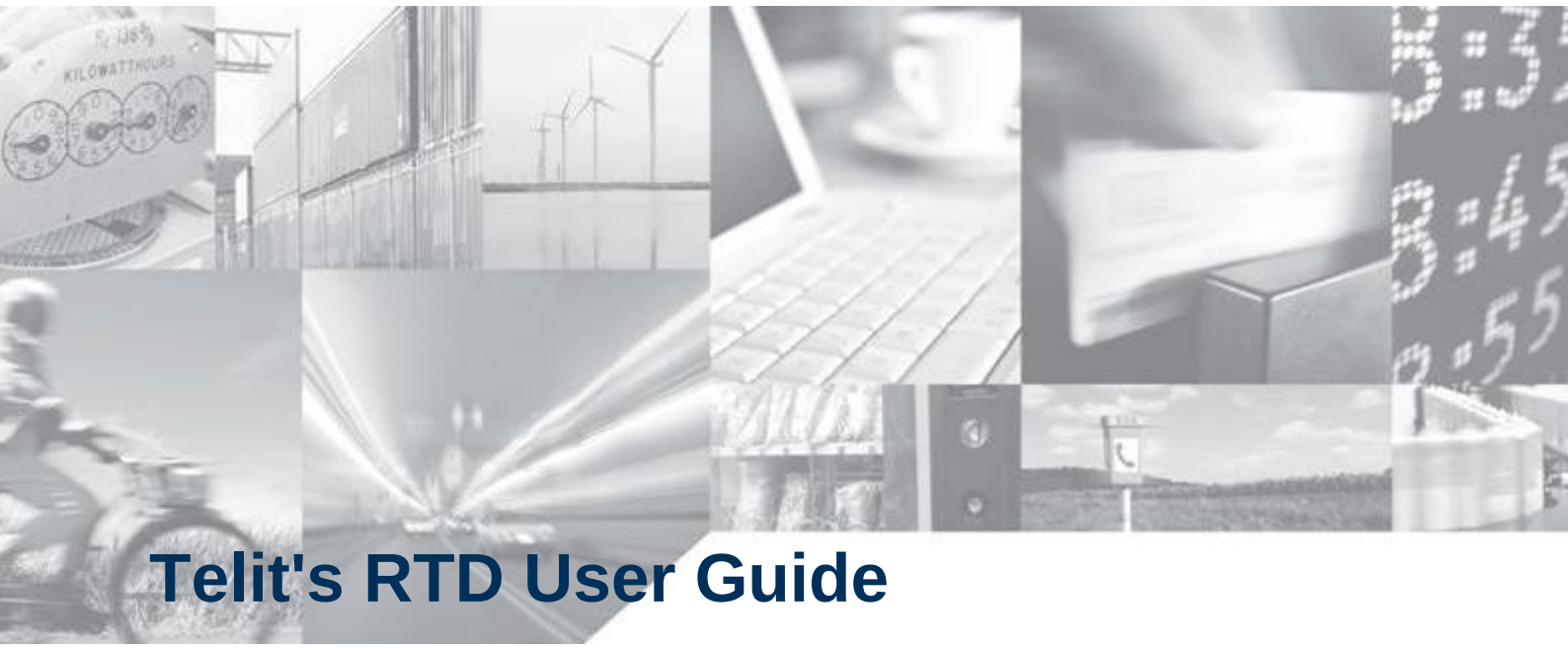


Telit's RTD User Guide

80000NT10021a Rev. 3 - 19/06/2013

Making machines talk.

GM Family (Modem)
GM862-GPS
GC Family (Compact)
GC864-QUAD
GC864-QUAD-V2
GC864-DUAL-V2
GE/GL Family (Embedded)
GE863-GPS
GE863-PRO3
GE864-QUAD
GE864-QUAD-V2
GE864-QUAD-Automotive–V2
GE864-QUAD-ATEX
GE864-DUAL-V2
GE864-GPS
GE865-QUAD
GL865-DUAL
GL865-DUAL V3
GL865-QUAD
GL868-DUAL
GE910-QUAD
GE910-QUAD V3
GE910-GNSS
GT Family (Terminal)
GT863-PY
GT864-QUAD
GT864-PY

APPLICABILITY TABLE

NOTE: If it is not explicitly indicated, all the features described on the present document are applicable on all Telit Modules indicated in the tables above.

Contents

DISCLAIMER

The information contained in this document is the proprietary information of Telit Communications S.p.A. and its affiliates (“TELIT”). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Telit, is strictly prohibited.

Telit makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Telit does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information.

Telit disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules.

Telit reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will, nevertheless be incorporated into new editions of this document.

1. Introduction

1.1 Scope

The purpose of this document is to describe the setup of the Telit Trace Tool and how to start and configure a Trace Session as it is supported by **Telit modules GSM/GPRS family**.

Real Time Debugger (**RTD**) is a tool dedicated to tracing the behaviour of the Telit module firmware (FW). To do his RTD has to be connected to the module by a serial link, and this can be done in several ways, depending on the availability of a serial channel in the environment where the module is used.

Once linked together, module can communicate with RTD and a set of strings with module information will be reported... For most scenarios and analysis we will not need to gather all the module traces, but rather only the traces related to the functionality of the investigation. This leads to the definition of "**class *of trace"*** i.e. a subset of all the possible information that can be received from the module. This is important because some of the information will not be useful in the investigation so it's important to gather the right amount of information by selecting the proper trace classes.

This document will be divided into two main sections to explain how to connect RTD to the module and how to select the right traces for different situations.

1.2 Document Organization

This User Guide contains the following chapters:

- “Chapter 1 Introduction”: Provides a scope for this User Guide, text conventions, and related documents.
- “Chapter 2 Serial Communication”: A tutorial on how to connect the **Telit module** to the RTD tool running on PC, with respect of the different type of connections it can exist.
- “Chapter 3 TelitSerialPortMux communication”: Brief introduction to the TelitSerialPortMux, a tool to communicate via virtual port between module and PC.
- “Chapter 4 RTD tool“: Is a brief, but thorough, description of the tool and the way to set up it.
- “Chapter 5 Class of trace”: Provides a definition for what a class of trace and how to set them for the need of trace
- “Chapter 6 Running the RTD”: The RTD tool at work: how to use it.
- “Chapter 7 Appendix”: Useful information to know related to RTD tool usage.

1.3 Document History

Revision	Date	Changes
0	2008-21-04	First release.
1	2009-01-19	Updated document and Errata Corrige
2	2011-05-02	Updated HW flow control ch 3 and fig. 5. Updated fig 3
3	2013-06-19	Added V3 and GE910 products. Updated the Serial Port Mux chapter

2 Serial Communication

As stated in the preliminary chapter, we can link RTD tool to module in several ways, this will depend on the availability of the serial port of the module and in some case on the interaction with the Python tool, if it is present and Python scripts are running.

2.1 The MODULE Serial PORT

Telit Modules can have one or two ports, it depends on the product. From now we will use the following naming convention for that:

- **ASC0** Program/Data PORT: used for AT command communications and updating FW.
- **ASC1** Auxiliary PORT: if present can be used for RTS tool.

For convenience the following table shows the pinout reference for the two serial ports.

Port pin out for the different products [1]

Table 1 Serial port pins

Product	ASC0			ASC1		
GM862-QUAD GM862-QUAD-PY	TX	→	Pin 37	TRACE TX	→	Pin 35
	RX	←	Pin 20	TRACE RX	←	Pin 41
GM862-GPS	TX	→	Pin 37	TRACE TX	→	Not Available
	RX	←	Pin 20	TRACE RX	←	Not Available
GE863-QUAD GE863-QUAD-PY	TX	→	Pin 38	TRACE TX	→	Pin 25
	RX	←	Pin 37	TRACE RX	←	Pin 26
GE863-GPS	TX	→	Pin 38	TRACE TX	→	Not Available
	RX	←	Pin 37	TRACE RX	←	Not Available
GE863-PRO[3]	TX	→	Ball E10	TRACE TX	→	Ball A10
	RX	←	Ball E9	TRACE RX	←	Ball A11
GC864-QUAD V2 GC864-QUAD-PY	TX	→	Pin 26	TRACE TX	→	Pin 24
	RX	←	Pin 25	TRACE RX	←	Pin 23
GE864-QUAD V2 GE864-QUAD-PY GE864-AUTO GE864-ATEX	TX	→	Ball E7	TRACE TX	→	Ball D11
	RX	←	Ball H8	TRACE RX	←	Ball F10
GL865-DUAL GL865-QUAD GL865-DUAL V3 GL868-DUAL	TX	→	Pin 8	TRACE TX	→	Pin 45
	RX	←	Pin 7	TRACE RX	←	Pin 44
GE865-QUAD	TX	→	Ball A4	TRACE TX	→	Ball D1
	RX	←	Ball A3	TRACE RX	←	Ball E1
GE910-QUAD GE910-GNSS GE910-QUAD V3	TX	→	Ball M15	TRACE TX	→	Ball D15
	RX	←	Ball N15	TRACE RX	←	Ball E15

[1] Pay attention at the different naming convention used in the Hardware user guide

2.2 Module mounted on EVK2. ASC1 available

This section is applicable to modules family without the GPS features.

Applicable list
GM Family (Modem)
GM862-QUAD
GM862-QUAD-PY
GC Family (Compact)
GC864-QUAD V2
GC864-QUAD/PY
GE/GL Family (Embedded)
GE863-QUAD
GE863-PY
GE863- PRO[3]
GE864-QUAD V2
GE864-QUAD/PY
GE864-AUTO V2
GE864-ATEX
GE865-QUAD
GE910-QUAD/GNSS & V3
GL865-DUAL/QUAD & V3

When mounted on the EVK2 both serial ports of the module are available from the connectors on the motherboard. It is possible to use the two RS232 stacked 9-pins connectors or a standard USB 1.1; the selection of RS-232 versus USB is controlled by the series of 10 jumpers on headers PL301 PL302 PL303 (see Figure 1).

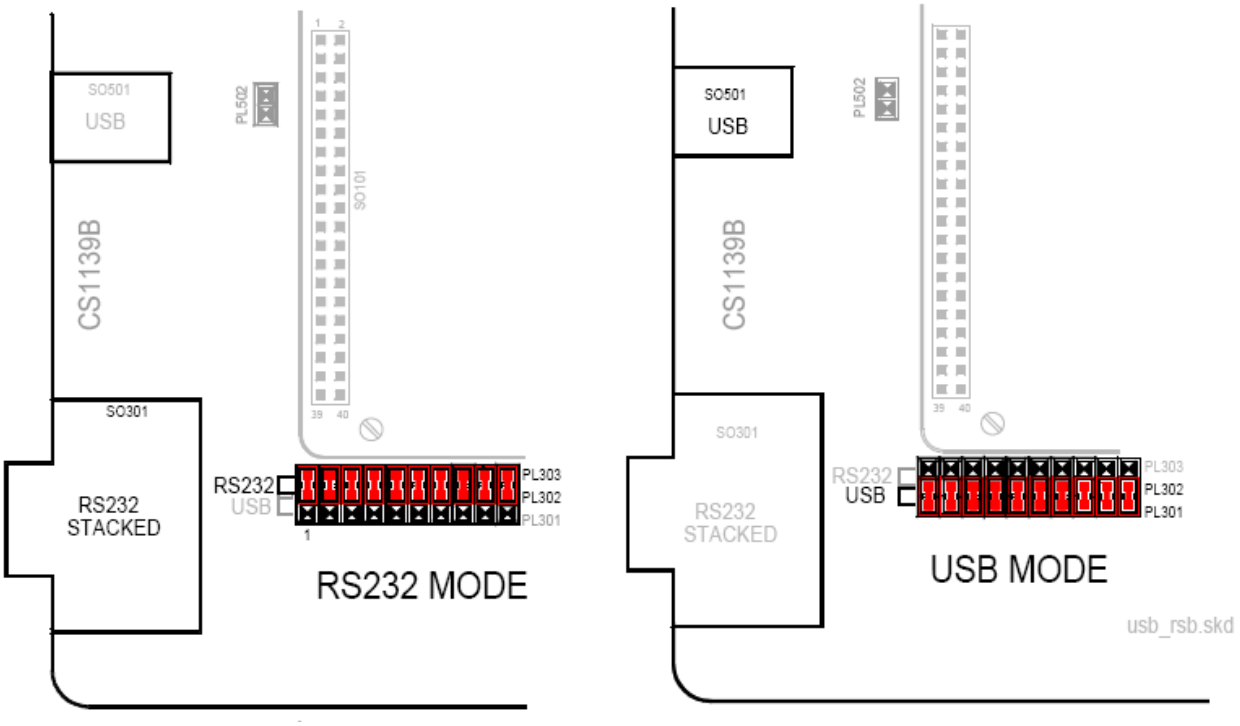


Figure 1 - Serial connectors selections jumper.

If standard RS232 is selected then the Double 9way D-Socket Connector must be used: the **upper** one for data communications (**AT commands or data send/receive**), while the **lower** one is used for connecting to **RTD** (Figure 2).

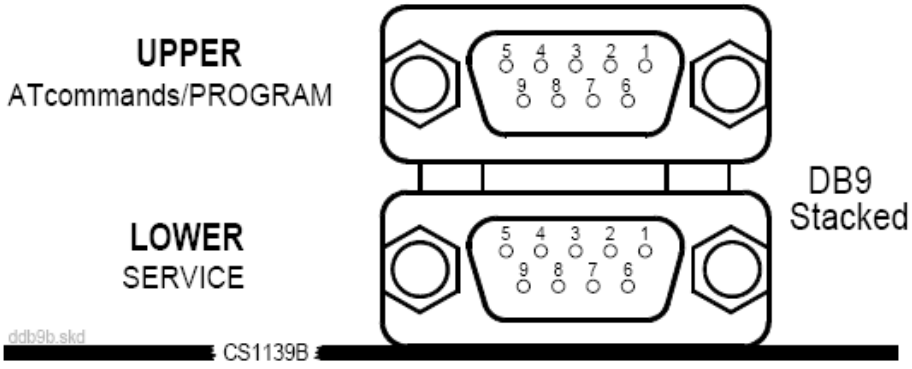


Figure 2 - RS232 connectors

If **USB** is selected the communication goes via USB cable (window driver is required). Once connected it, two virtual serial ports appear on your system. The low-numbered one is dedicated for trace, while the other for data and AT command (see Figure 3).

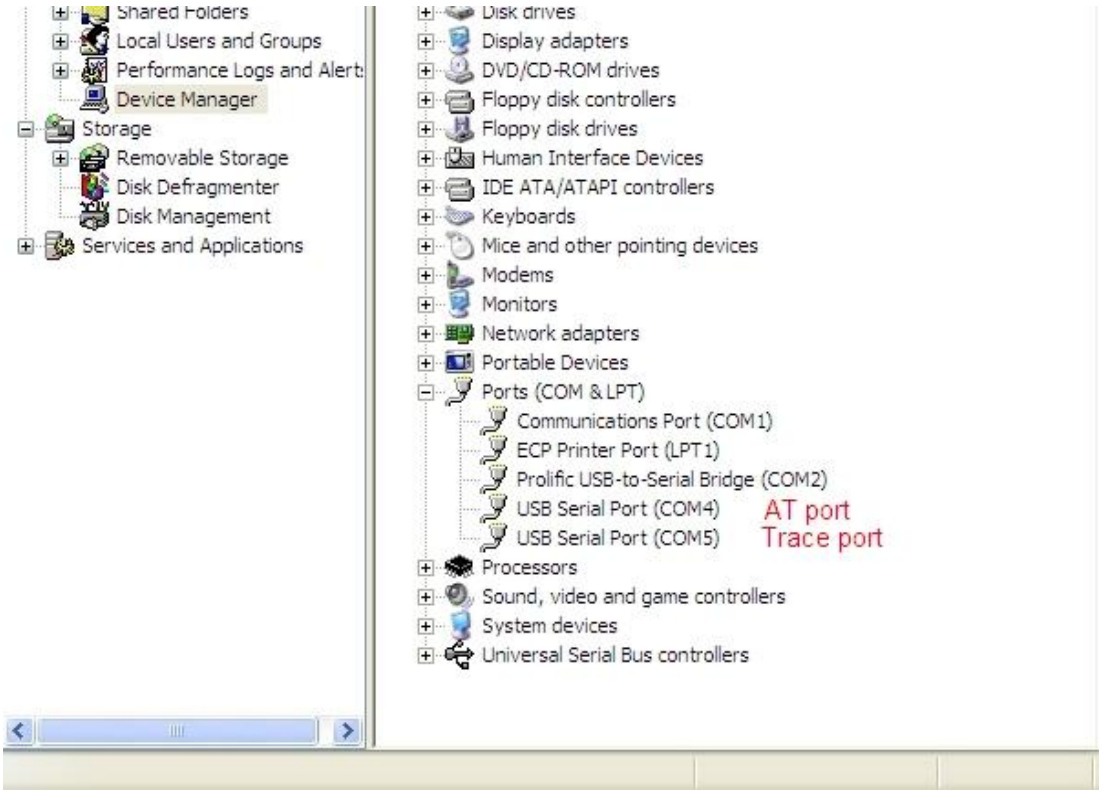


Figure 3 - Port Selection

Once linked the module, trace procedure can be start (cf. 4_RTD Tool).

2.3 Module mounted on EVK2. ASC1 not available

This section is applicable to modules family with no access to the serial port 1 (Trace Port) that is all the modules with GPS features, and the GT863-PY GT864-PY and GT864-QUAD

Applicable list
GM Family (Modem)
GM862-GPS
GE/GL Family (Embedded)
GE863-GPS
GE864-GPS
GT Family
GT864-QUAD
GT864-PY
GT863-PY

The **GPS** family modules **don't have** the ASC1 port available so trace can be done via ASC0 (prog port). In this case communications from module and RTD go via ASC0 and TelitSerialPortMux Tool, so refer to section 3 TelitSerialPortMux.
After connecting the module via TelitSerialPortMux trace procedure can be start (cf. sec 4 RTD Tool).

2.4 Module standalone. ASC1 available.

This section is applicable to modules family without the GPS features.

Applicable list
GM Family (Modem)
GM862-QUAD
GM862-QUAD-PY
GC Family (Compact)
GC864-QUAD V2
GC864-QUAD/PY
GE/GL Family (Embedded)
GE863-QUAD
GE863-PY
GE863- PRO[3]
GE864-QUAD V2
GE864-QUAD/PY
GE864-AUTO V2
GE864-ATEX
GE865-QUAD
GE910-QUAD/GNSS & V3
GL865-DUAL/QUAD & V3

You can link the **RTD** to the trace port of the module (**ASC1**). In this case you must be aware of the fact that the voltage level on the PC RS232 is different from that on the module pins, so you have to interface to the device through a level translator (cf. App 7.1: Level translation).
Once linked the module, trace procedure can be start (cf. sec 4 RTD Tool).

IMPORTANT!

Remember that a level translation **MUST** be done from ASC1 and PC serial connector as recommended by Hardware User Guide[2] .

[2] **GM862-GPS Hardware User Guide 1vv0300794**

2.5 Module standalone. ASC1 not available

This section is applicable to modules family with no access to the serial port 1 (Trace Port) that is all the modules with GPS features, and the GT863-PY GT864-PY and GT864-QUAD

Applicable list
GM Family (Modem)
GM862-GPS
GE/GL Family (Embedded)
GE863-GPS
GE864-GPS
GT Family
GT864-QUAD
GT864-PY
GT863-PY

The **GPS** family modules **don't have** the ASC1 port available so trace must be done via ASC0 (prog port). In this case communications from module and RTD go via ASC0 and TelitSerialPortMux Tool, so refer to section 3 TelitSerialPortMux.
In this case you must be aware of the fact that the voltage level on the PC RS232 is different from that on the module pins, so you have to interface to the device through a level translator ((cf. App 7.1: Level translation).
Once linked the module, trace procedure can be start (cf. sec 4 RTD Tool).

3 TelitSerialPortMux communication

TelitSerialPortMux is a tool that allows the communication from the module to the external PC trough 4 virtually serial links, all multiplexed in one physical port (blue area below).
The virtual port #3 (red area below) is dedicated to traces, so this can be the solution for connecting RTD when module is not mounted on the EVK2 or for all the products that doesn't have the trace port available.

First of all you have to install the TelitSerialPortMux application on a PC, and when running it a monitor window will appear like that shown in .
The tool can work if the FW of the module implements the Cmux protocol. This is available from software version 7.02.04.

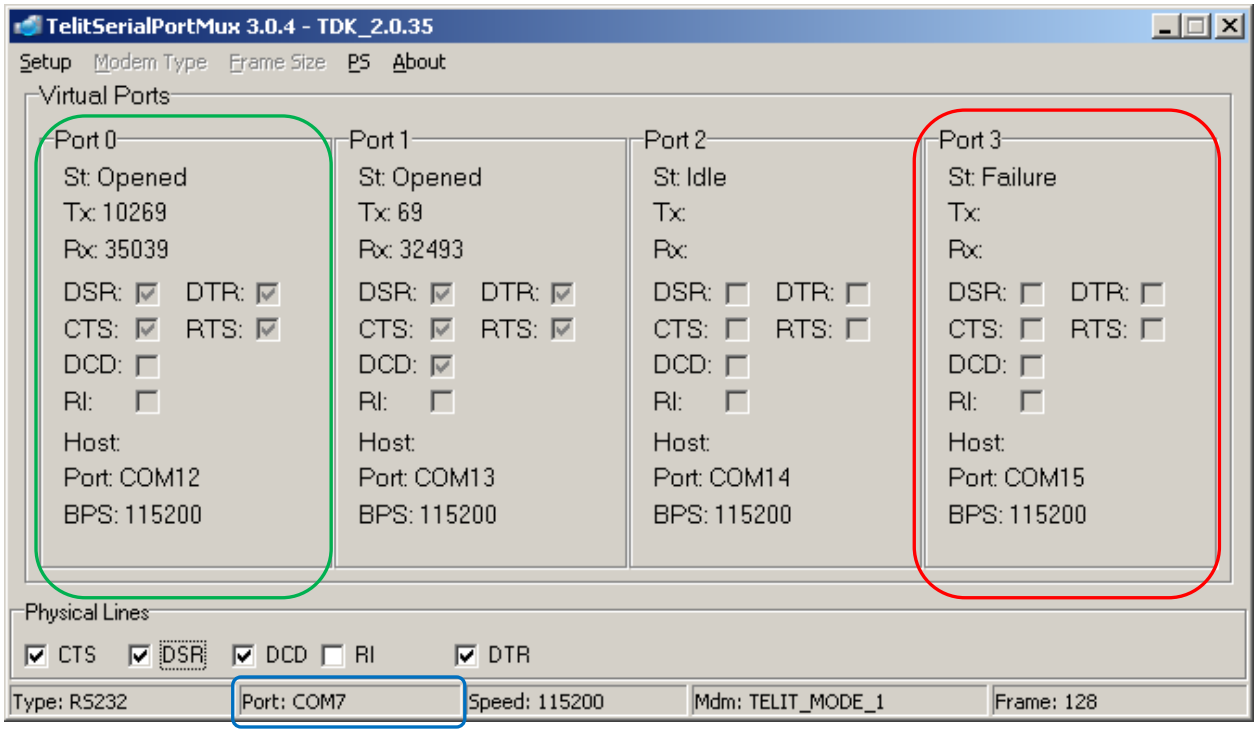


Figure 4 - TelitSerialPortMux Monitor Window

Connect the RS232 serial cable between the serial port for AT commands (ASC0) and a PC RS232 COM port.
The CMUX needs all the UART connections, except RING and DCD, to implement the flow control.
A workaround does exist to avoid the usage of the full HW flow control; it consists of connecting TX and RX only and emulates the HW flow control. From PC side loop together DTR with DSR line and CTS with RTS (Figure 5).

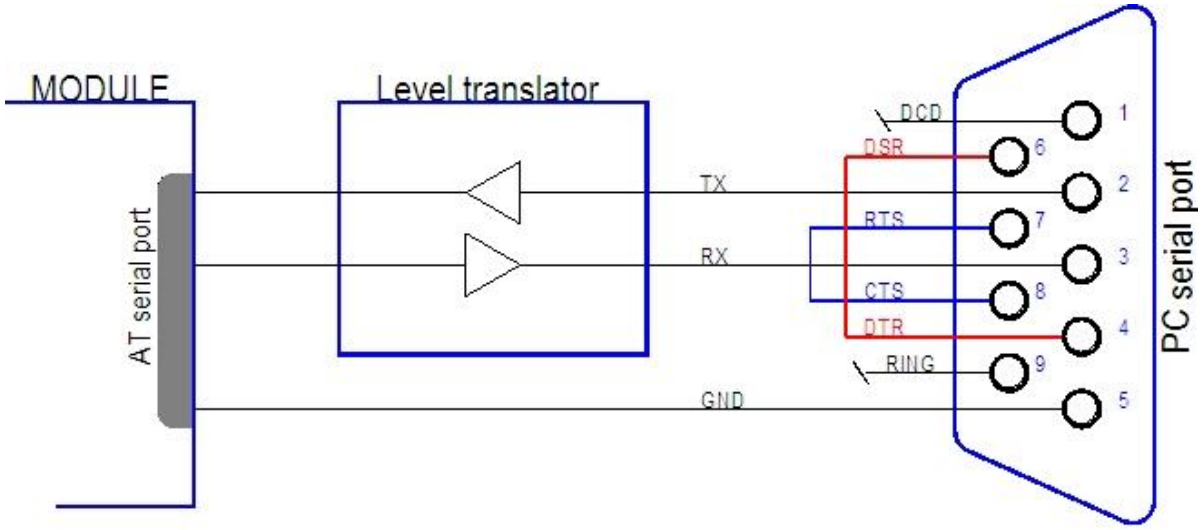


Figure 5 - UART Workaround

When running the program you can use the first 3 virtual ports for **data communication** and/or **AT command**, while the **Virtual Port #4** can be used for **connecting to RTD.**

You can change the name of the virtual port, and the real port used on the PC from the Setup window (see Figure 6).

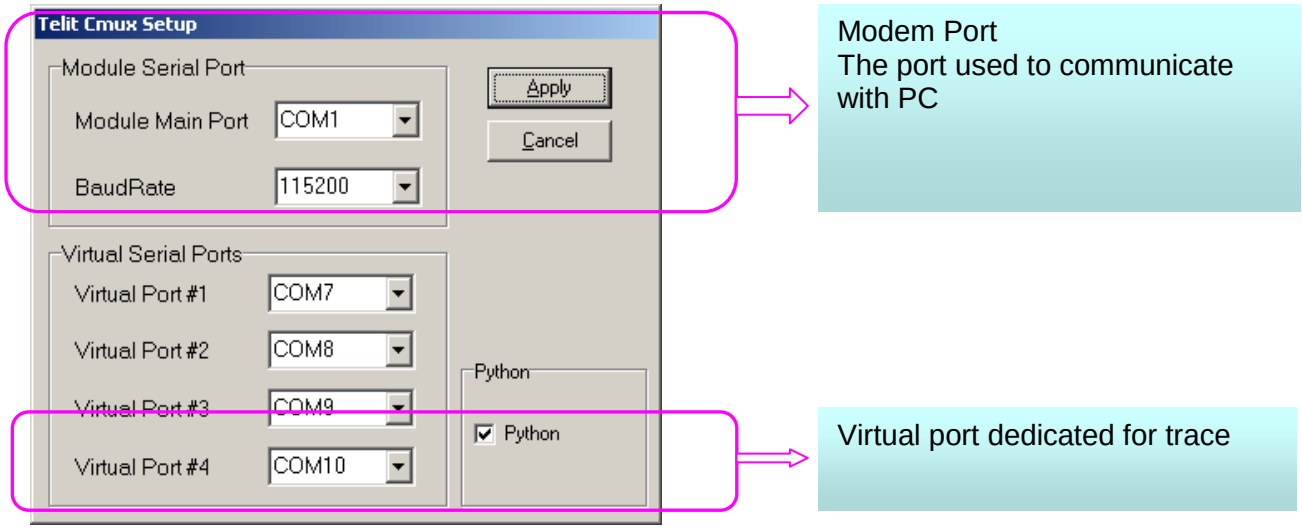


Figure 6 Setup Window

From the Set Up you have to select which port to use on your PC to communicate with the module, and the name of the four virtual ports installed by the tool on your PC.
Remember that the **Virtual Port#4** is the port dedicated to the trace.

IMPORTANT!

1. Remember that a level translation **MUST** be done from ASC1 and PC serial connector as recommended by Hardware User Guide[3] .

2. Pay attention to **not conflict names of virtual ports** with others port already presents on your PC.

---

[3] **GM862-GPS Hardware User Guide 1vv0300794**

4 RTD tool

RTD is a PC Software tool that allows users to debug FW running inside the module by tracing the states of tasks that are running. Traces are presented in the form of strings composed inside module FW and carrying information to help understand the FW behaviour. Figure 7 shows an example of how RTS should look while running.

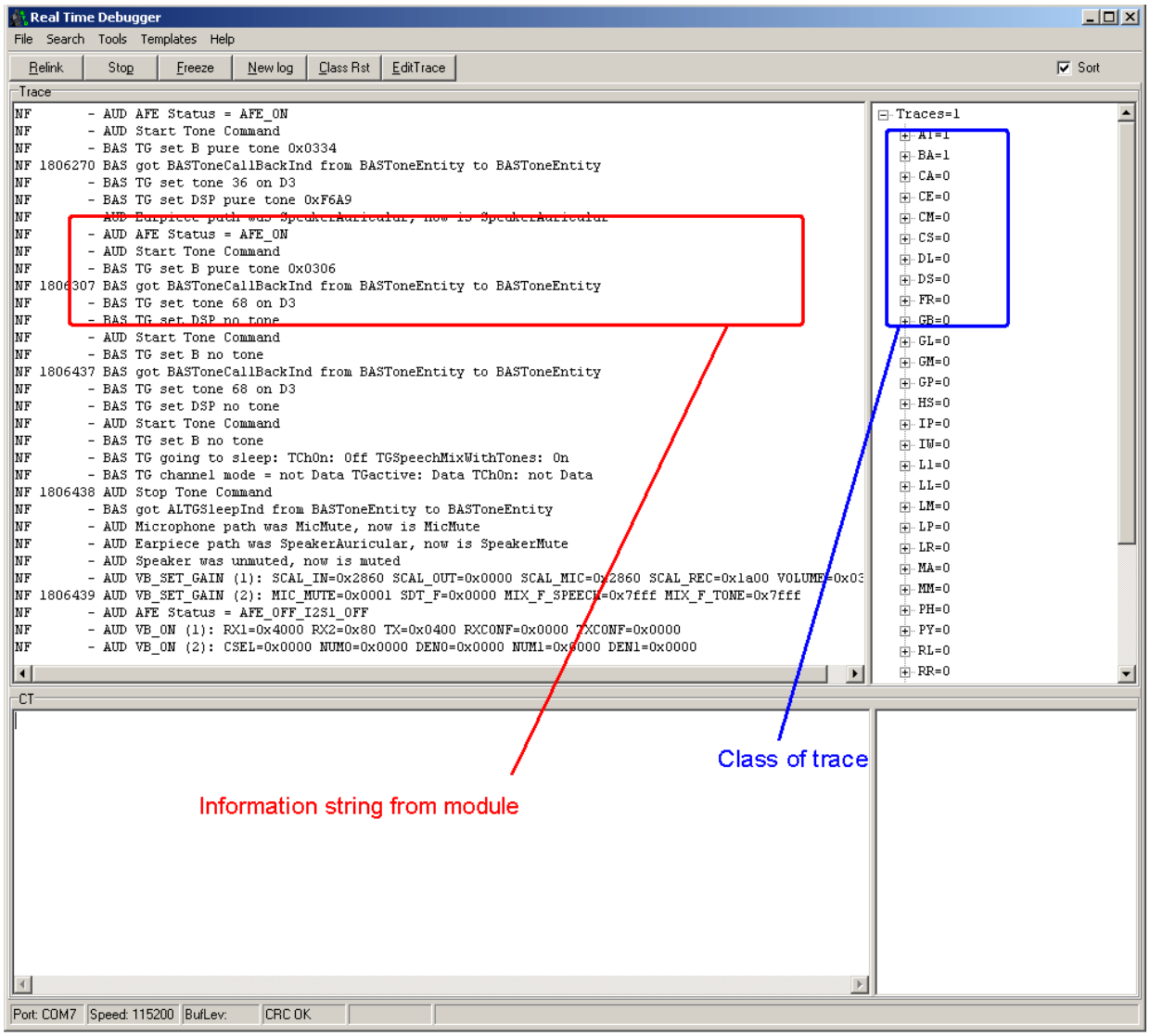
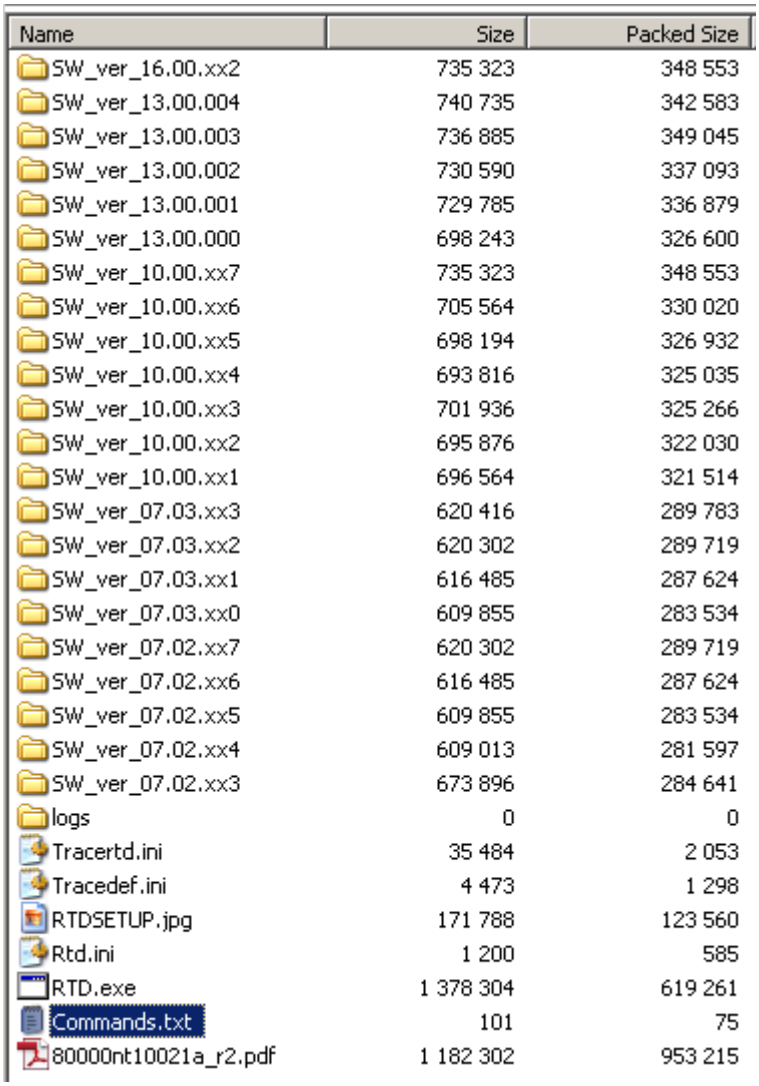


Figure 7 - RTD window

RTD comes in a zipped file; after unzipped you'll have a folder called RTD organized like in Figure 8.



Name	Size	Packed Size
SW_ver_16.00.xx2	735 323	348 553
SW_ver_13.00.004	740 735	342 583
SW_ver_13.00.003	736 885	349 045
SW_ver_13.00.002	730 590	337 093
SW_ver_13.00.001	729 785	336 879
SW_ver_13.00.000	698 243	326 600
SW_ver_10.00.xx7	735 323	348 553
SW_ver_10.00.xx6	705 564	330 020
SW_ver_10.00.xx5	698 194	326 932
SW_ver_10.00.xx4	693 816	325 035
SW_ver_10.00.xx3	701 936	325 266
SW_ver_10.00.xx2	695 876	322 030
SW_ver_10.00.xx1	696 564	321 514
SW_ver_07.03.xx3	620 416	289 783
SW_ver_07.03.xx2	620 302	289 719
SW_ver_07.03.xx1	616 485	287 624
SW_ver_07.03.xx0	609 855	283 534
SW_ver_07.02.xx7	620 302	289 719
SW_ver_07.02.xx6	616 485	287 624
SW_ver_07.02.xx5	609 855	283 534
SW_ver_07.02.xx4	609 013	281 597
SW_ver_07.02.xx3	673 896	284 641
logs	0	0
Tracertd.ini	35 484	2 053
Tracedef.ini	4 473	1 298
RTDSETUP.jpg	171 788	123 560
Rtd.ini	1 200	585
RTD.exe	1 378 304	619 261
Commands.txt	101	75
80000nt10021a_r2.pdf	1 182 302	953 215

Figure 8 - RTD installation folder

Running the **RTD.exe** will execute the tool and some additional settings are needed, how explained in the next chapter.

4.1 Trace Setup

From menu → File → Setup a window like this will appear (Figure 9)

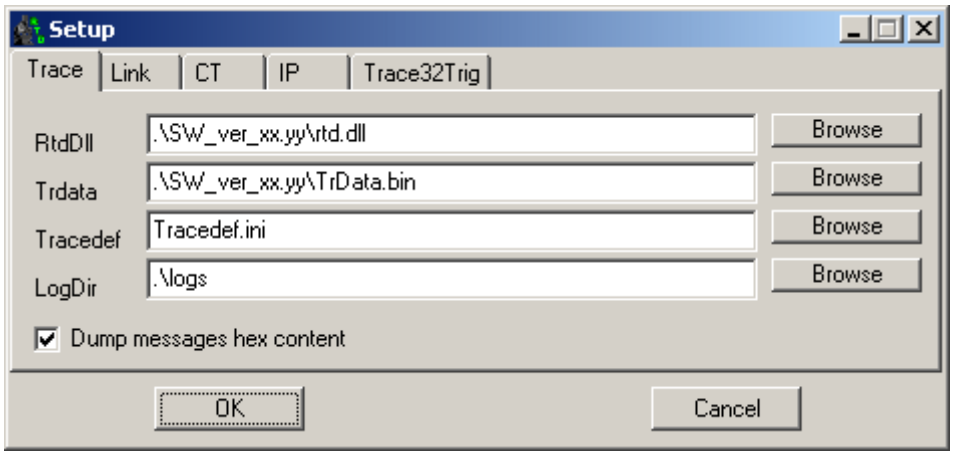


Figure 9 - Trace Setup Window

Trace tab allow general settings of the trace.

➔ Every FW version of the module has its own trace definitions that allow the synchronization between module and tool. These FW-dependant files are the **rtd.dll** and **TrData.bin**. They are located into the **SW_ver_xx_yy** folder. You can have multiple folder like this, one for every FW version you plain to test, and so just change the name of the folder into the relative path.

➔ Tracedef is the file where different class of trace (see following chapter) can be set. Generally you don't need to change anything here.

➔ LogDir is the folder where RTD saves the file reporting all the trace. Logs are simple text files with the extension .rtd and are auto numbered.

Current logging file is trac.rtd; if you stop and then restart the tool this file assumes the name trac000x.rtd, where x is a progressive number, and a new trac.rtd is generated.

In case you need you can create others folder to organize your logs.
If you have selected a wrong path for the Sw version (dll and TrData.bin not aligned with the SW version on the module), after clicking on the OK you'll have an error message like BAD CRC and the toolbar at the bottom of the trace tool has a red background colour (See Figure 10).

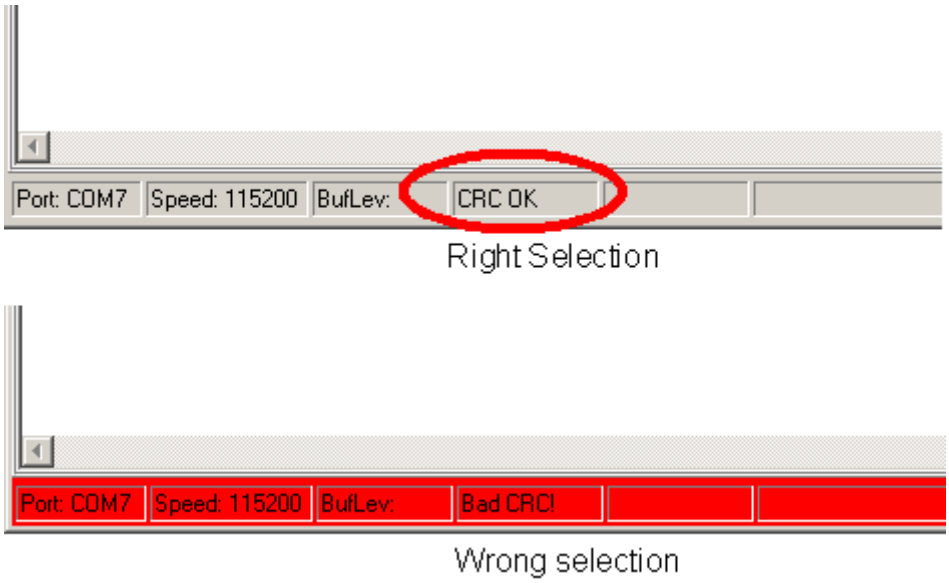


Figure 10 - FW Selection

4.2 Link Setup

Setup the link is straightforward, you only need to select port name and baud rate.
The port selection combo box is automatically filled with the port available on your PC, included the virtually installed from **CMUX** if used.

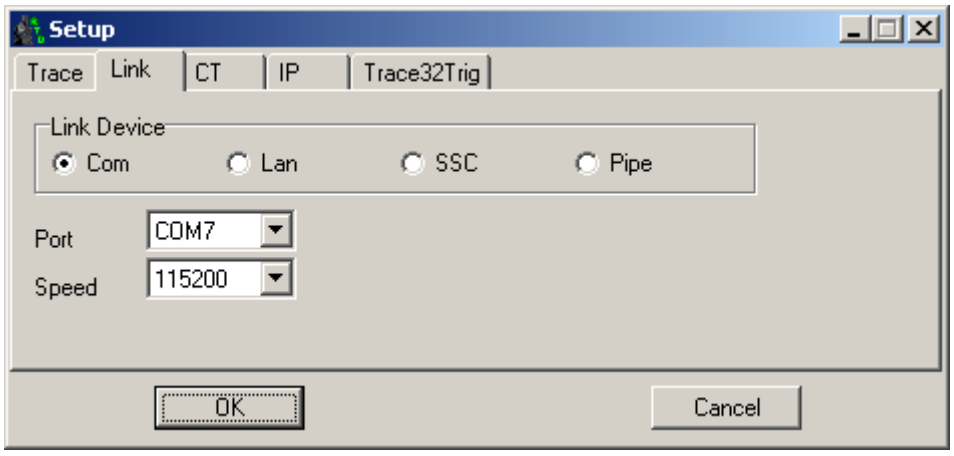


Figure 11 - Link Setup Window

5 Class of trace

While running the FW on module it's possible to trace a lot of information strings reporting messages regarding the current status of the FW. These strings are grouped into different sets of classes, and each of these classes is divided into a subset of strings reporting information on a single behaviour. It is possible, and advisable, to select only the required set of this classes to report trace regarding your specific need. This can be done on the **Class Selection Window**.

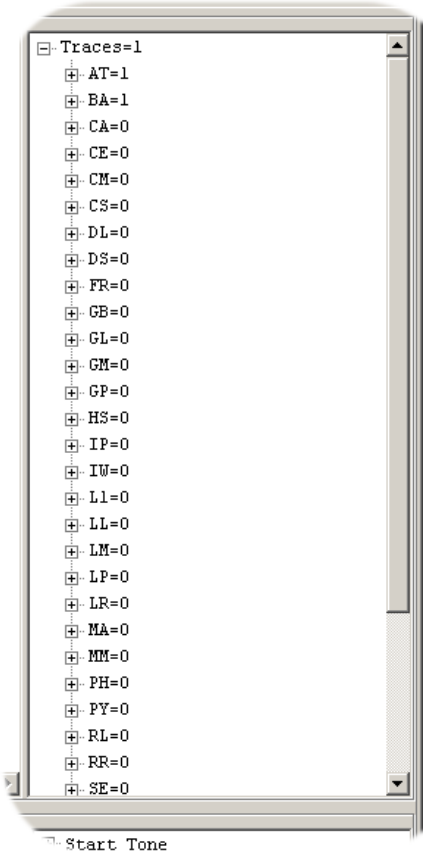


Figure 12 - Class Selection Window

Only the selected classes (that with value 1) will be reported. To select/unselect a class right click on it.

Telit has created some preconfigured sets of trace classes (templates) to help gather traces files for a few common scenarios.

From menu → Templates; then a dropdown list of configuration appear (Figure 13)

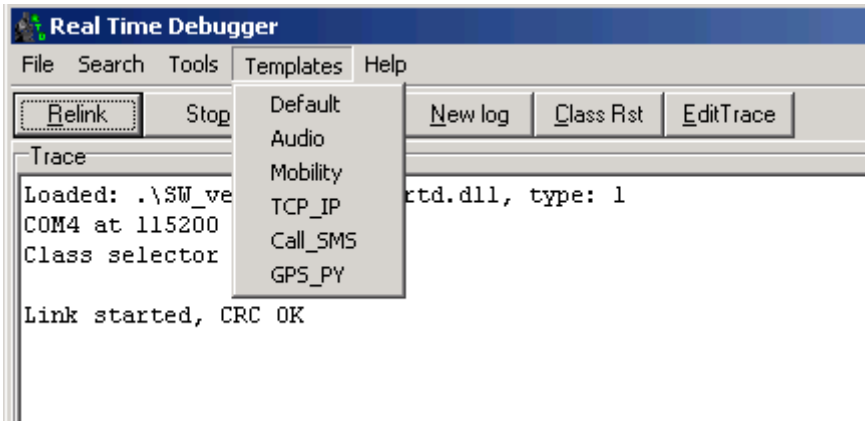


Figure 13 - Templates Menu

This enables all the trace classes needed for tracing **Audio, Mobility, …** behaviour together with the AT command executions.
Figure 13 is only an example. Others templates are available and the profiles can be easily upgraded in future.

WARNING

Select **ALL** the class available leads to a great amount of traffic on serial link and can arm the PC and/or the normal flow of SW into the module.

6 Running the RTD

Once executed all the setup as described in previous sections and the tool is running, it must be explained how to start and stop the trace, and how get information from it.
The **RTD** menu appears as in Figure 14

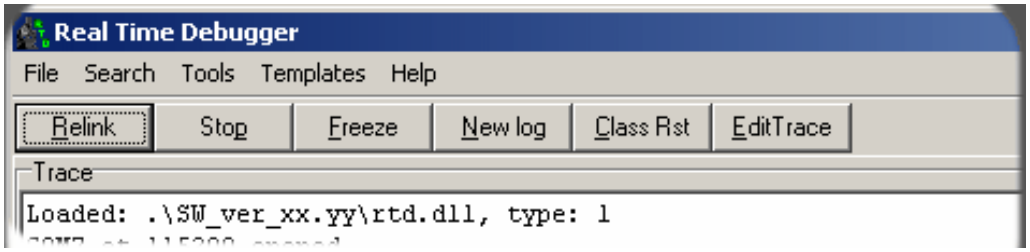


Figure 14 RTD Running menu

The menu functions are:

- "**Relink**" forces the synchronization when the module is ON.
- "**Stop**" disables all traces without changing the user class configuration.
- "**Freeze**" pause the current trace. The serial port is closed so it can be used for other application.
- "**New log**" backups log files and create a new one.
  - After a New Log execution, the current trace is saved in the folder specified as trac00xx.rtd where xx is a progressive number.
  - A new file named Trac.rtd is generated.
- "**Class Rst**" restores the default class configuration. **Don't use it** 
- "**Edit Trace**" calls editor associated to log file.
  - If no editor is associated to the extension .rtd nothing happen, so you have to select some program for this (the common Windows™ notepad will work).

The procedure is: run RTD.exe, so when ready to do the test, press "**relink**" and go on until the behaviour you want to investigate occurs. Now you can pause the tracing with "**freeze**" function; the file Trac.rtd in the folder you have selected in the setup procedure is the useful trace you can send us.

7 Appendix

7.1 Level Translation

This section provides more explanations and hints about the direct connection from module to PC. As stated in the previous sections, depending on the module you have (with or without the GPS feature) you can have or not the availability of the trace port (ASC1).
In both cases you can communicate with the RTD tool, directly from ASC1 when you have it, through TelitSerialPortMux otherwise.

Anyway a level translation **must** be used in between to adapt different level of the RS232 lines (cf. Figure 15).

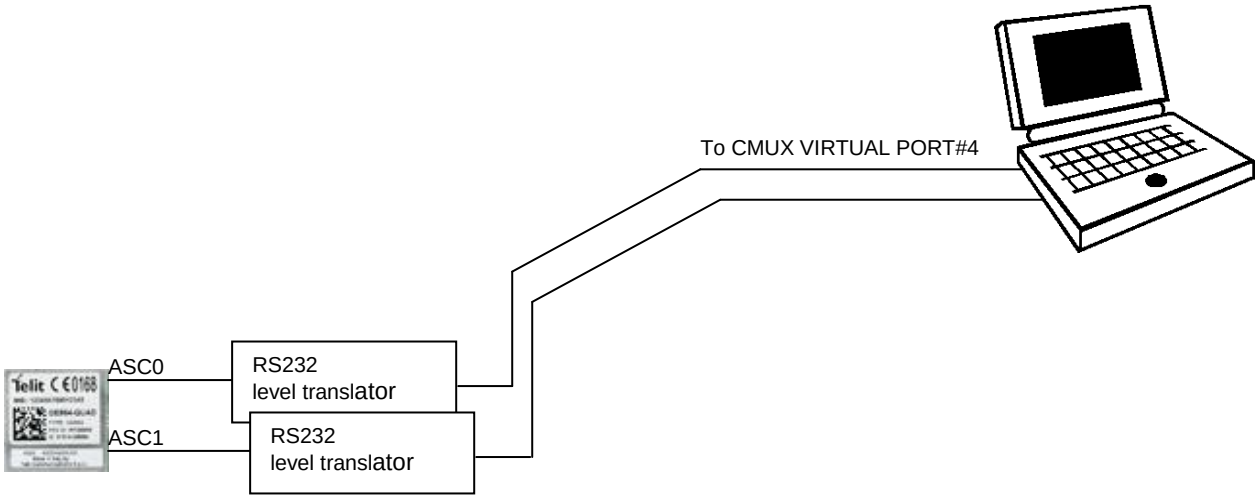


Figure 15 - Module connection

If you have an EVK2 available you can use the translator circuit mounted on it (cf. Figure 16).

Please notice that the EVK2 must be powered on with a module inserted on the interface connectors in order to power supply the translator.

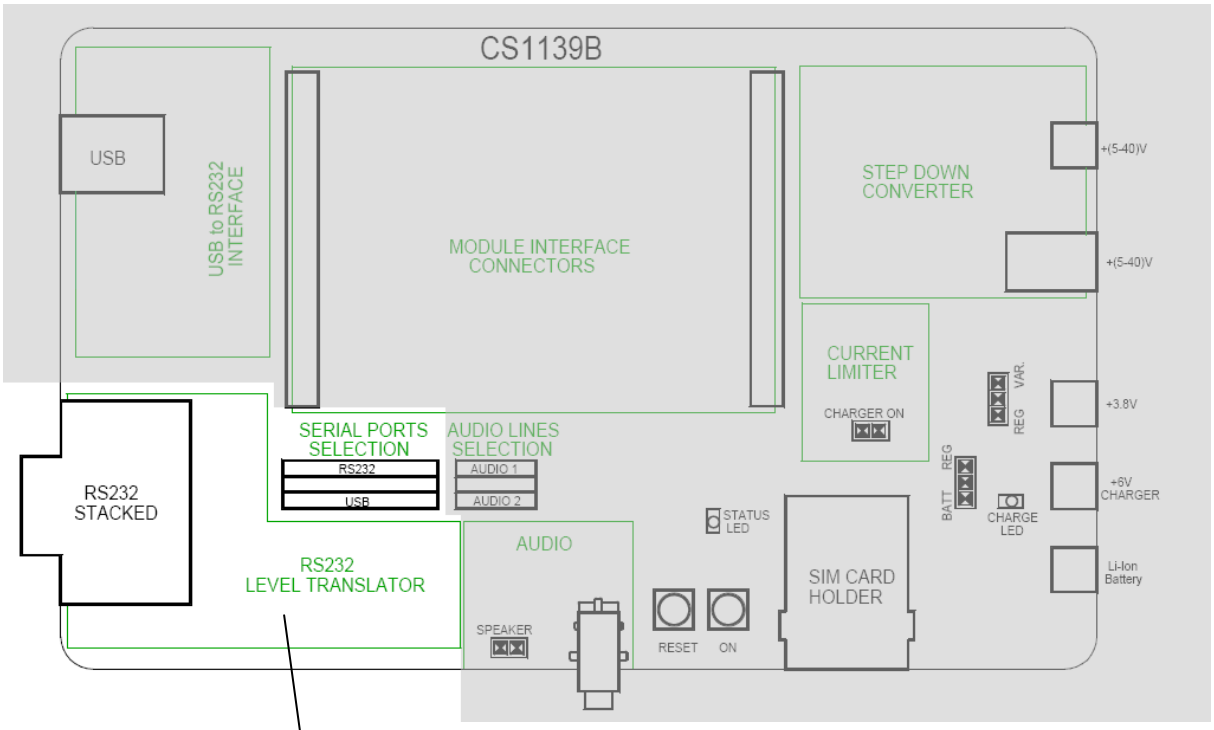
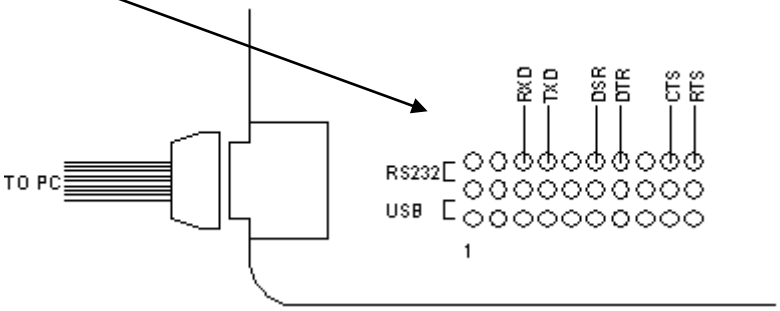


Figure 16 - EVK2: The level translator area



If you don't have an EVK2 available you must implement your own serial adapter. There is a large choice of chip available on market to do the job, following you will find an example using a common chip.

Ex: common level translation chip.

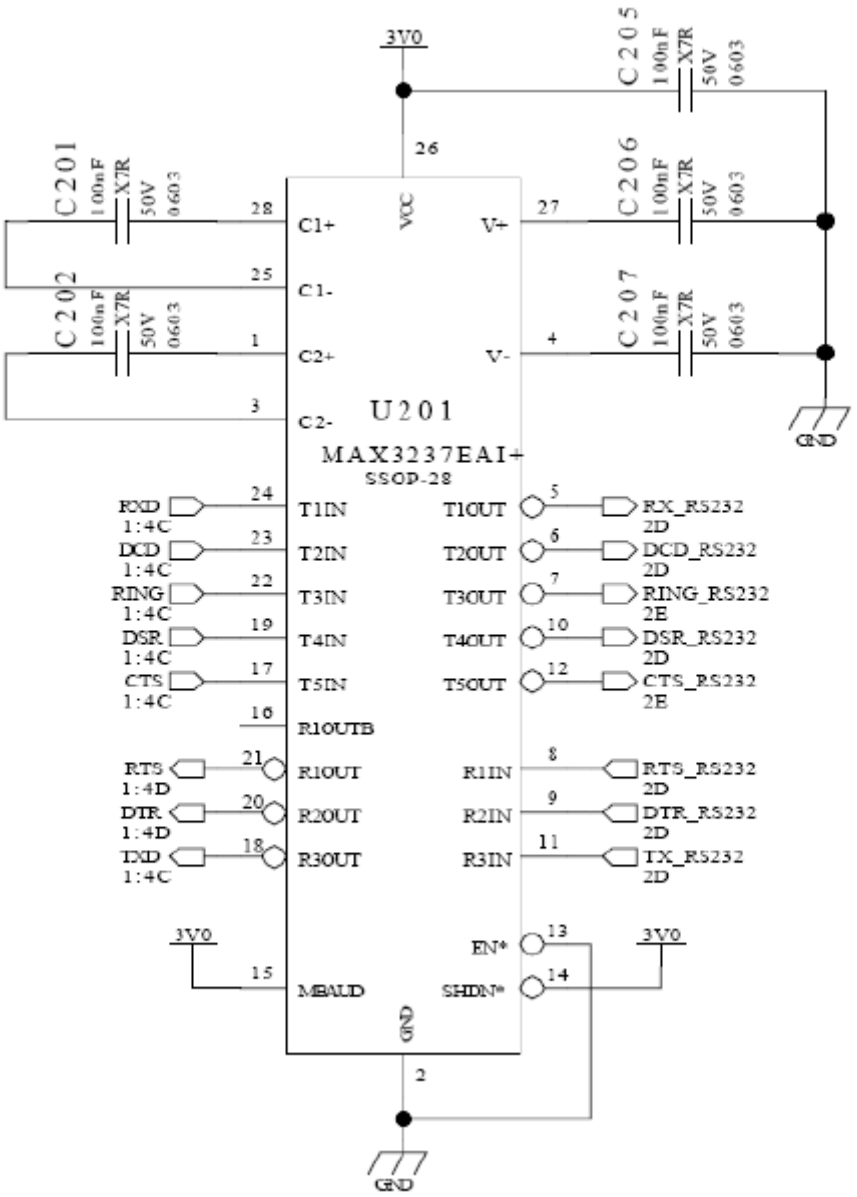


Figure 17 Level Translator chip

8 Acronyms and Abbreviations

RTD	Real Time Debugger
CTS	Clear To Send
DCE	Data Circuit-Terminating Equipment
DTE	Data Terminal Equipment
DTMF	Dual Tone Multiple Frequency
DTR	Data Terminal Ready
ME	Mobile Equipment
TE	Terminal Equipment (DTE or User Terminal)
UART	Universal Asynchronous Receiver Transmitter
FW	Telit module embedded firmware