

Author: Sierra Wireless		Date: January 15, 2015	
APN Content Level	BASIC <input checked="" type="checkbox"/>	INTERMEDIATE <input type="checkbox"/>	ADVANCED <input type="checkbox"/>
	Confidentiality	Public <input checked="" type="checkbox"/>	Private <input type="checkbox"/>
Hardware Compatibility	Product Line	AirPrime	Series HL854xx
Software Compatibility	ALL		



1 Version

Application Notes may be updated over their lifetime. To ensure you design with the correct version, please check the application notes page in <http://developer.sierrawireless.com/> for latest versions.

2 Introduction

This Application Note (APN) is provided to Sierra Wireless distributors and clients to aid more rapid development of embedded applications using the Sierra Wireless portfolio of cellular solutions. To request a new application note, contact your regional Sierra Wireless Product Marketing Manager.

3 Application Note Overview

This document describes the steps needed to get debug logs (using Mobile Analyser and 3G Firmware Trace Tool) for the HL854xx on Windows and on other OS, either through USB or UART. The following table lists debug log availability for different ports and software.

Table 1: Debug Log Availability

Port	Mobile Analyser (all other traces)	3G Firmware Trace Tool (3G DSP traces)	Supported OS
USB	Yes	Yes	Windows/Other
UART	Yes	No	Windows/Other
DEBUG	No	Yes	Windows

4 Mobile Analyser

4.1 Trace Formats

MA stands for Mobile Analyser which is an MS Windows tool used to capture a modem's real-time trace messages. These messages are structured in different formats, and are also displayed differently in the MA. The different message formats are listed below.

- ASCII Message (printf)
- SDL Message (stack)
- Monitoring Message
- LLT(Low Level Trace) Messages
- APOXI Message
- Bluetooth Message
- Gate Message
- 3G stack Message

4.2 Trace Types

Trace types are used for filtering traces in the Firmware. If a trace type is disabled, the Firmware does not send the traces of that type and the MA does **not** capture it either.

Defined trace types are:

- **pr**: printf – for ASCII Message
- **st**: stack – for SDL Message
- **lt**: LLT – for Low Level Trace Message
- **ap**: APOXI – for APOXI Message
- **bt**:Bluetooth – for Bluetooth Message
- **ga**: GATE – for Gate Message
- **ae**: AENEAS – for 3G stack Message
- **db**: debug
- **li**: LwIP (Lightweight-IP)

4.3 MA Repository

The MA repository is a database which contains methods for decoding the trace formats specified above.

There are two types of repositories:

- LAN Repository – this is provided on a LAN drive and ftp.
- Local Repository – this is a mirror of the LAN repository. A local repository should be prepared if a LAN repository is not set.

4.4 Trace Filtering

Both MA trace filtering and Firmware trace filtering should be configured prior to getting MA traces.

MA trace filtering is based on the following conditions via the MA filter menu:

- Trace types
- Inter process message ID (SDL type only)

Follow these steps to configure the MA trace filter.

1. Click *Filter > Specify Filter in Trace Tool Rules*.
2. From the Filter Messages in Trace Tool window, filters can be set either manually (by clicking Show All) or loaded from a file (by clicking Load From File), provided by Sierra Wireless.
3. Click OK.

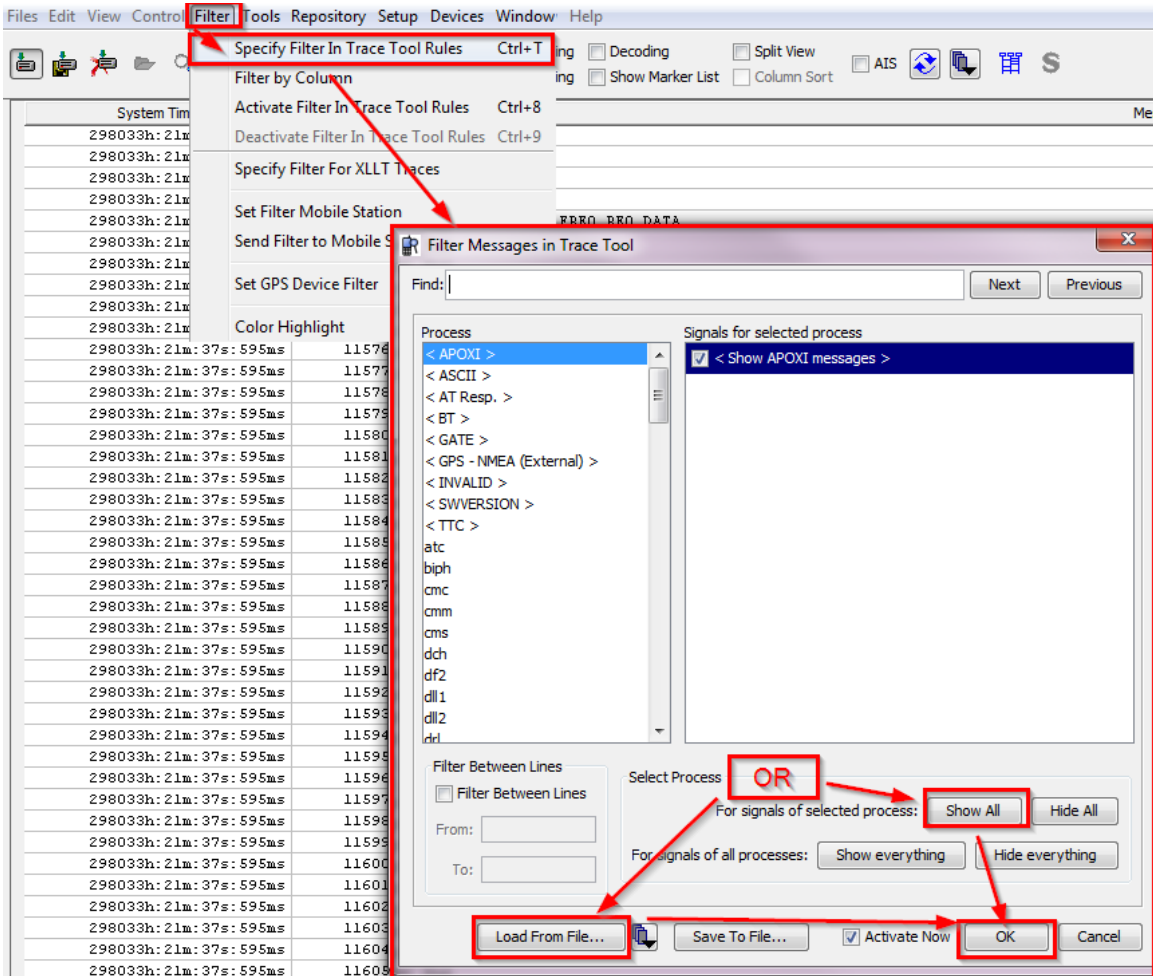


Figure 1. MA Trace Filter Configuration

The Firmware trace filtering must be configured with **AT+TRACE** (Intel traces) or **AT+SWITRC=1** (Sierra Wireless traces) before getting MA traces.

4.4.1 Using AT+TRACE

Command format:

AT+TRACE=[<mode>], [<speed>], ["<unit>=<umode>[,<unit>=<umode>[;...]]"], ["<method>"], [PowerSavingCountdown]

Example:

AT+TRACE=,,"ap=1,st=1,db=1,pr=1,bt=1,lt=1,li=1,ga=1,ae=0"

Configuration is saved persistently to non-volatile memory so it can be programmed once and reconfiguration is not necessary when the modem reboots.

4.4.2 Using AT+SWITRC=1

Sierra Wireless traces are traces that are used by the Sierra Wireless Firmware source code. Currently, they are formatted as MA trace format "printf". The filtering of Sierra Wireless traces is supported via the **AT+SWITRC=1** command instead of **AT+TRACE**.

Command format:

```
AT+SWITRC=1,<trace group>,<trace level>,<enable=1, disable=0>
```

or

```
AT+SWITRC=1,<trace group>,<32bits trace filter mask like FFFFFFFF>
```

or (reset to default)

```
AT+SWITRC=1,<trace group>,<
```

Sierra Wireless traces can be routed to any AT port or HW port (UART/I²C) via the **AT+SWITRC=30** command:

```
AT+SWITRC=30,<trace group>,<trace port>
```

Configuration is saved persistently to non-volatile memory so it can be programmed once and reconfiguration is not necessary when the modem reboots.

Note: **AT+SWITRC** is **not** available in Firmware 5.4.0.0 and earlier releases.

4.5 MA Pre-Configuration

4.5.1 Set Heap Size

1. Click *Setup > Options*.
2. From the Mobile Analyser Options window, select Java VM from the left pane and select 1024 MB from the drop-down list.

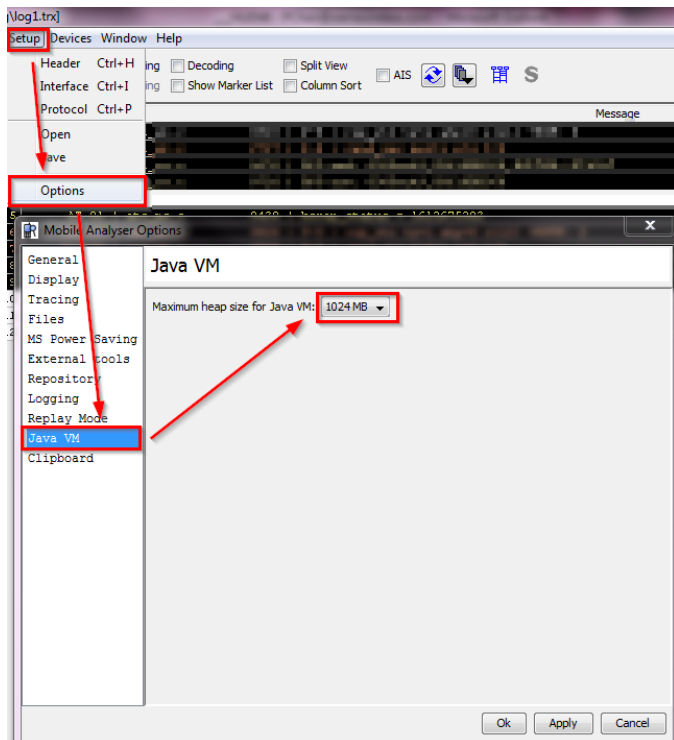


Figure 2. Setting Java VM Heap Size

4.5.2 Set and Apply MA Repository

The MA repository should be selected based on the Intel release name.

The correct Intel release name can be found in the latest integrated SBM listed in the HL854xx Firmware release change log file. To do this, search for “Integrate MOD_6260_VZZ.YYYY.YY_RXX” with the highest XX value.

For example, MOD_6260_V05.1412.00_R07 should be used for HL854xx Firmware RHL85xx.5.5.X.0.xxxxxxxxxxxx.x6250_1 (where X ≥ 1).

After retrieving the correct Intel release name, set the MA repository by following these steps:

1. Click *Setup > Options*.
2. From the Mobile Analyser Options window, select Repository from the left pane, and click Browse from the active repository location. The default path is “C:\Program Files\Intel\Mobile_Analyser\Repository” if the Mobile Analyser installation path is under “C:\Program Files\Intel”.

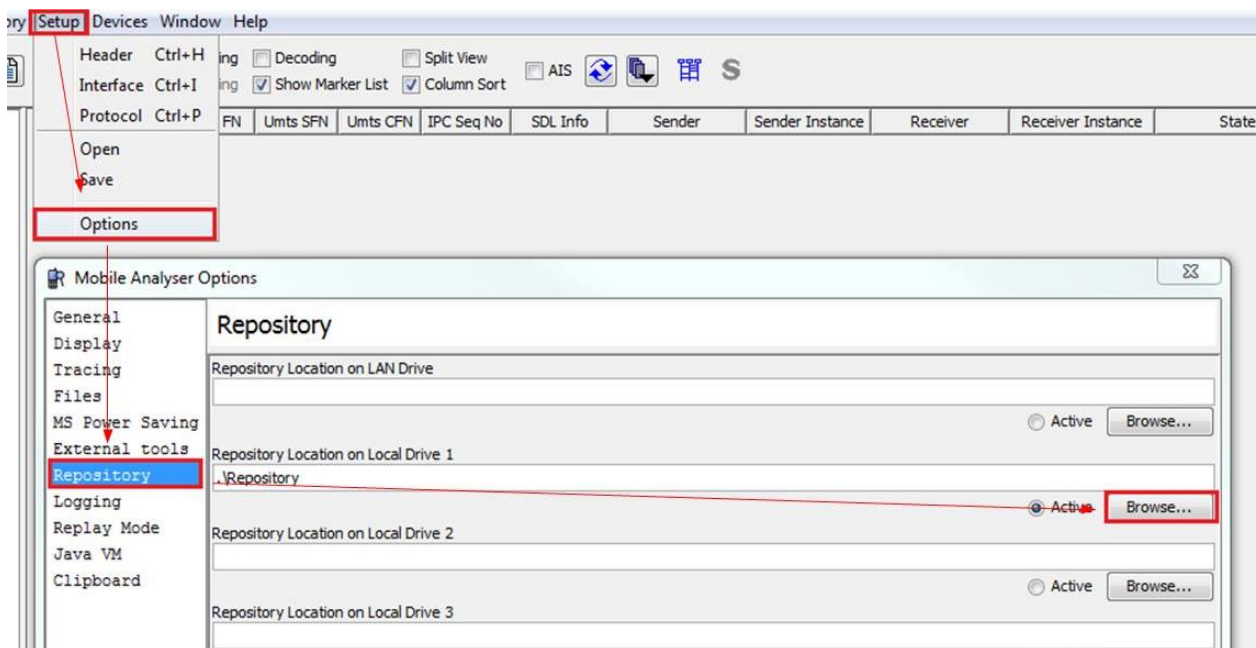
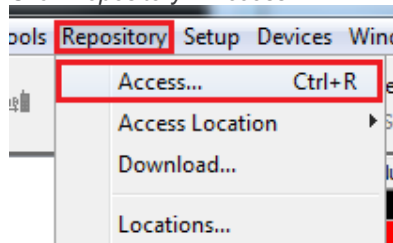


Figure 3. Setting MA Repository Path

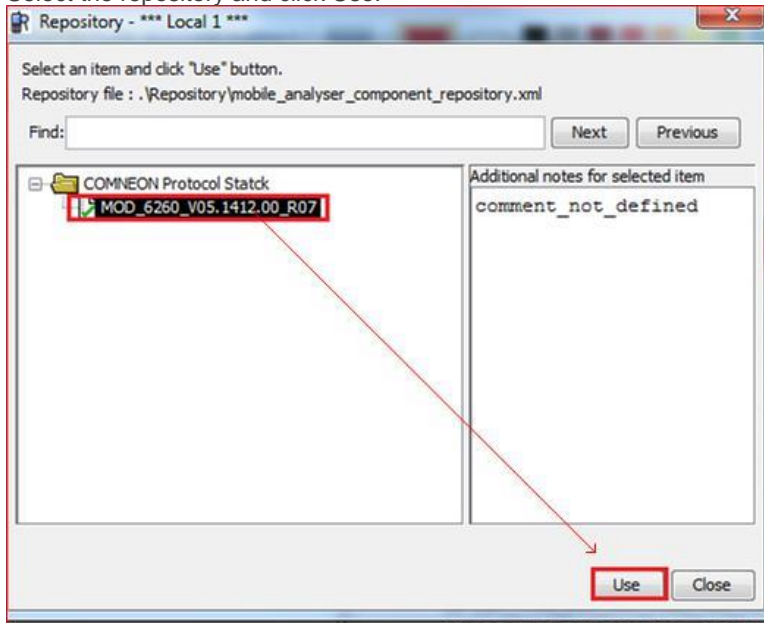
Note that all repository files should be copied into the repository location on the local drive path set above, i.e. C:\Program Files\Intel\Mobile_Analyser\Repository\.

After the MA repository has been set, apply the repository by following these steps:

1. Click *Repository > Access...*



2. Select the repository and click Use.



4.5.3 Set Up the Trace Filter

The Firmware trace filter should be configured (once per modem or issue) using **AT+TRACE** and/or **AT+SWITRC=1**.

For example:

```
AT+TRACE=,,"ap=1,st=1,db=1,pr=1,bt=1,lt=1,li=1,ga=1,ae=0"
```

```
AT+SWITRC=1,1,FFFFFFFF
```

For more information on command format, refer to section 4.4.1 Using AT+TRACE or 4.4.2 Using AT+SWITRC=1.

Usually, only "ASCII (printf)" traces are required by Sierra Wireless. A sample filter configuration is shown below.

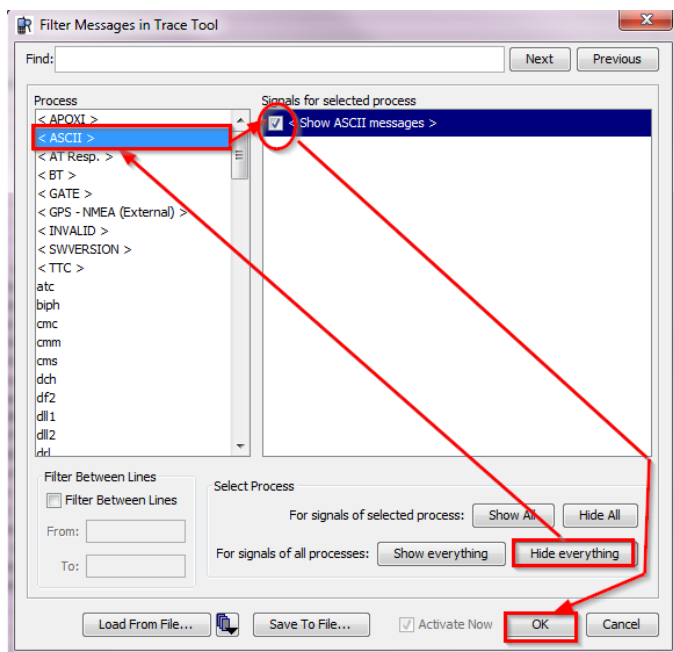


Figure 4. Sample Filter Configuration

4.5.4 Create a New Trace File

Create a new trace file before the trace session starts by following these steps:

1. Select *Files > New Trace File*.
2. Specify a name for the new trace file and click Save.

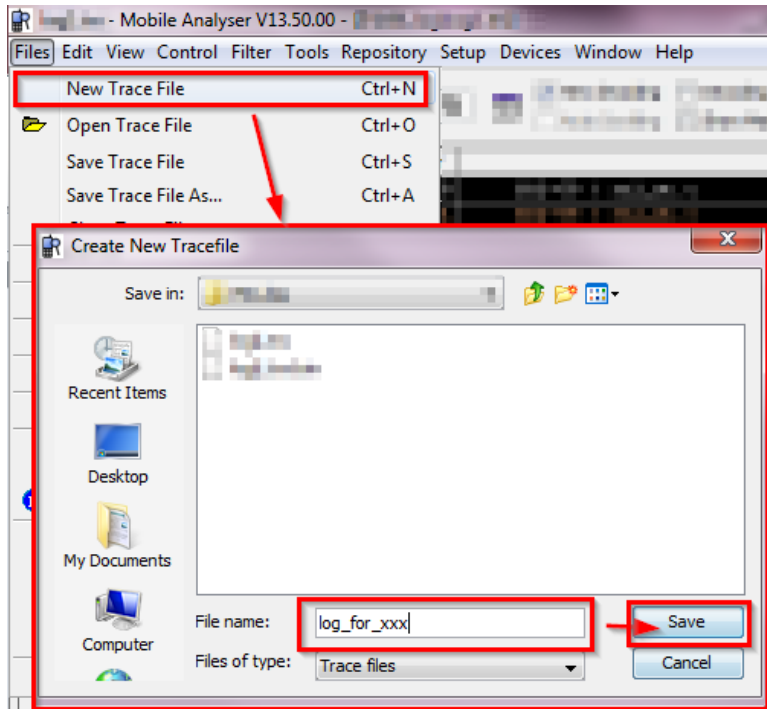


Figure 5. New Trace File Creation

5 3G Firmware Trace Tool

“3G FTT” stands for “3G Firmware Trace Tool”, which is a firmware trace tool used to retrieve 3G DSP traces.

5.1 SocMonDLL

SocMonDLL is a DLL file containing all firmware-dependent functions. SocMonDLL is not a part of the 3G FTT installation but is delivered with the Firmware and needs to be installed manually to the 3G FTT path. The current SocMonDLL for the HL854xx is “socmondll_203.325.003.dll”.

To set SocMonDLL, a folder “socmodnll” needs to be created in the 3G FTT path (“C:\Program Files\Intel\3G_FTT_V13.24.0000” by default). Then move the SocMonDLL files (e.g. socmondll_203.325.003.dll) into this folder (i.e. “C:\Program Files\Intel\3G_FTT_V13.24.0000\socmodnll”).

If no valid SocMonDLL is loaded when the 3G FTT software is launched, a pop-up window appears as show below.

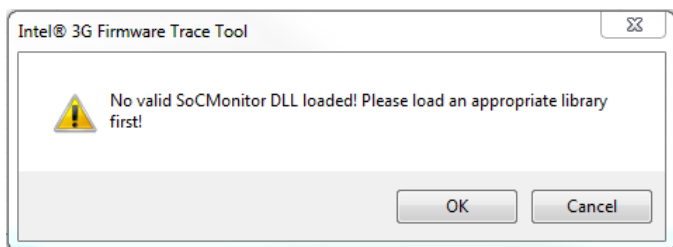


Figure 6. No Valid SocMonDLL is Loaded

To load SocMonDLL, click “OK” from the window above, and from the main window, select *Import > Load SoC-DLL...* as shown below and then select the SocMonDLL file.

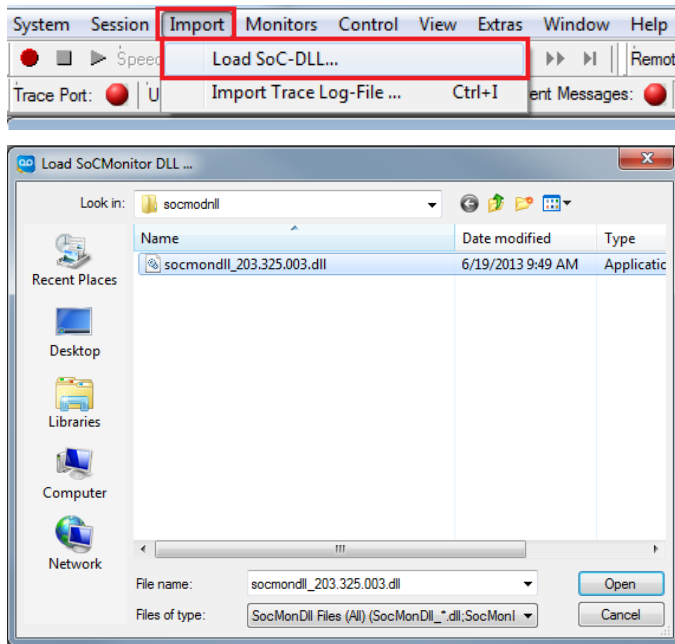
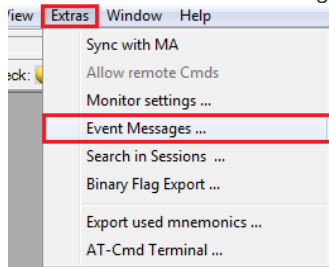


Figure 7. Load SocMonDLL

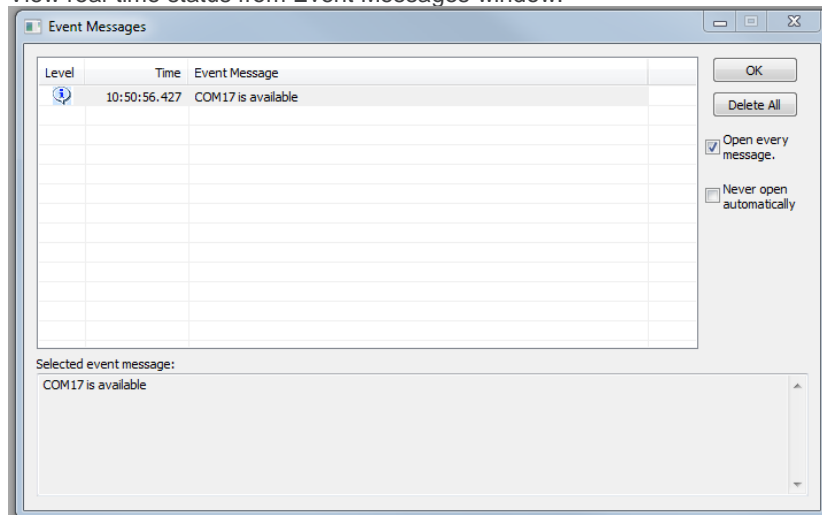
5.2 Event Messages

To view real-time modem status from Event Messages, follow these steps:

1. Select *Extras > Event Messages...*



2. View real-time status from Event Messages window.



5.3 UE Connection State Toolbar

The “UE Connection state” toolbar indicates the states of the Trace Port, the UE connection and the Trace Flag check.

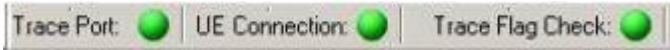


Figure 8. UE Connection State Toolbar

To ensure an effective tracing session, the “Trace Port” indicator must be green, indicating a successful connection with the trace port.

When the tracing starts, the “UE Connection” indicator turns green and yellow at intervals as the UE connection is polled periodically. If this indicator turns red or grey, check if there’s any error or failure on the modem.

If the trace-flags were read from the hardware, the received data will be compared with the target values.

6 Getting MA Traces on Windows through USB

6.1 COM Port Setup

The MA interface should be set up in accordance with the COM port used as in Windows Device Manager. In the screenshot below, “IMC Port (USB1)” corresponds to COM25.

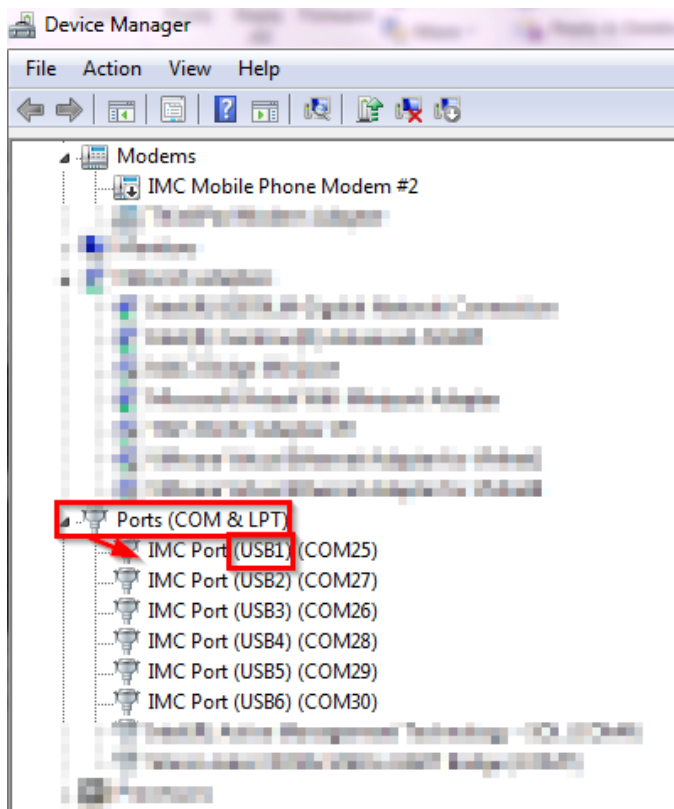


Figure 9. COM Port Setup

Configure the MA interface based on Figure 9 by selecting Setup > Interface. From the Trace Interface Settings window, set the correct configuration (as shown in the figure below) and click OK.

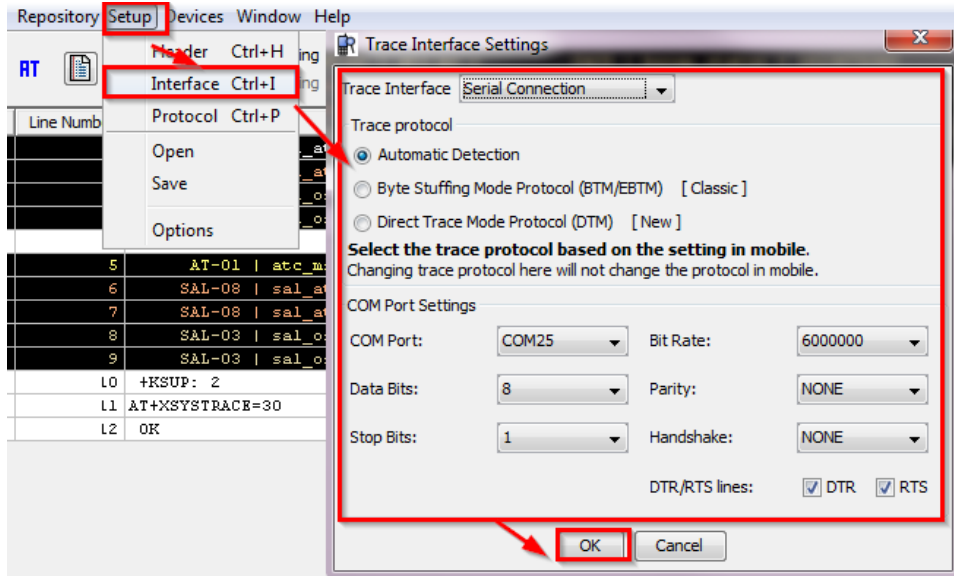


Figure 10. MA Interface Configuration

6.2 Start Tracing Using the Trace Button in MA

Before recording trace on MA, send **AT+TRACE=1** to enable tracing.

Tracing can be started by clicking the trace button on the menu bar.

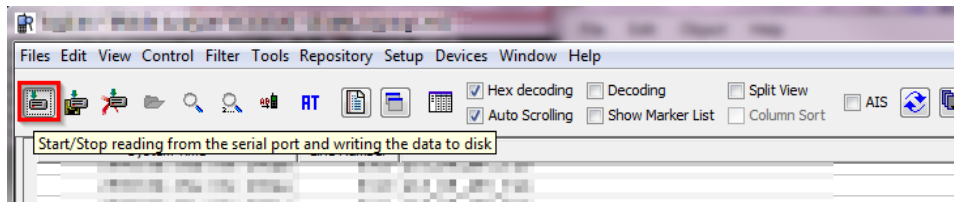


Figure 11. Trace Button

6.3 Stop Tracing and Save Trace File Set

To stop tracing, click the trace button from the menu bar.

To save the trace file, click *File > Save Trace File*.

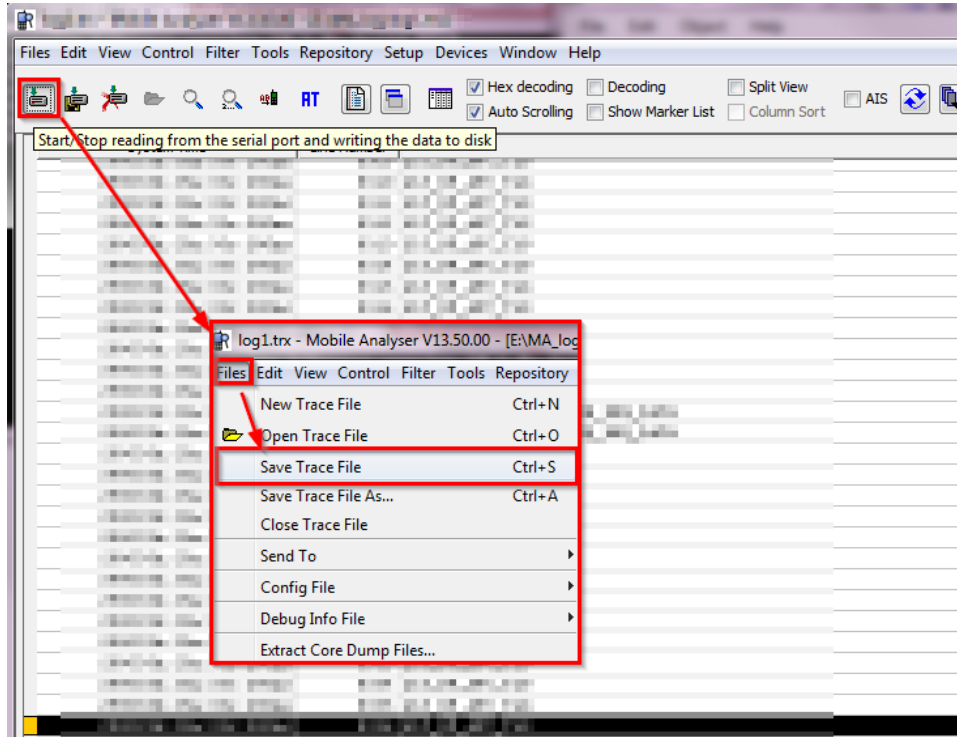


Figure 12. Stopping Trace and Saving Trace File

Output files for trace filename “log1” are shown in the figure below.

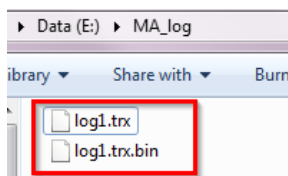


Figure 13. Output Files

7 Getting MA Traces on Windows through UART

7.1 Pre-Configuration

Taking trace on UART requires a specific configuration in advance in order to set the UART port as the trace port. To pre-configure the UART, follow these steps:

1. Send **AT+TRACE=1** to enable trace.
2. Send **AT+XSIO=11** to set UART as trace port.
3. Reboot the modem using **AT+CFUN=1, 1**.
4. Open MA and set the COM port according to the UART port in Windows Device Manager.

This setting will configure the UART as the trace port. After reboot, AT commands should not be sent with the AT terminal on the UART port as it no longer serves as AT port. Nevertheless, AT commands could be sent on the trace port with MA (see section 7.3 Send AT Commands on the Trace Port).

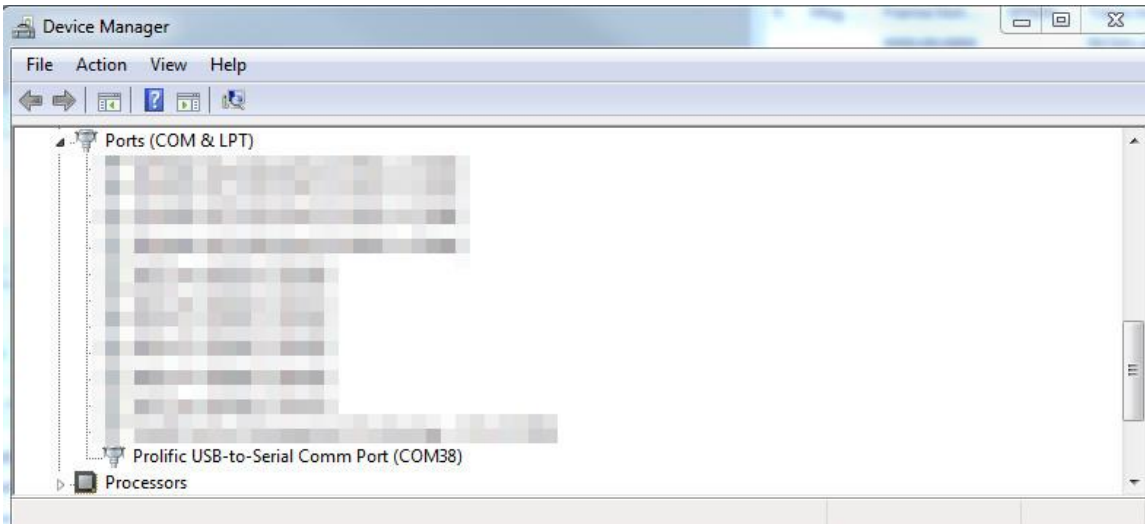


Figure 14. UART Pre-Configuration

If using a Prolific USB-to-Serial Comm Port, the COM port should be configured as COM38.

7.2 Start and Stop Trace

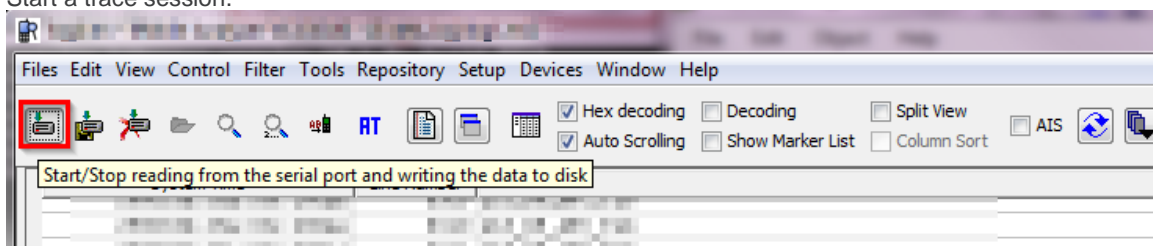
Trace could be taken using the same methods as describes in sections 6.2 Start Tracing Using the Trace Button in MA and 6.3 Stop Tracing and Save Trace File Set.

7.3 Send AT Commands on the Trace Port

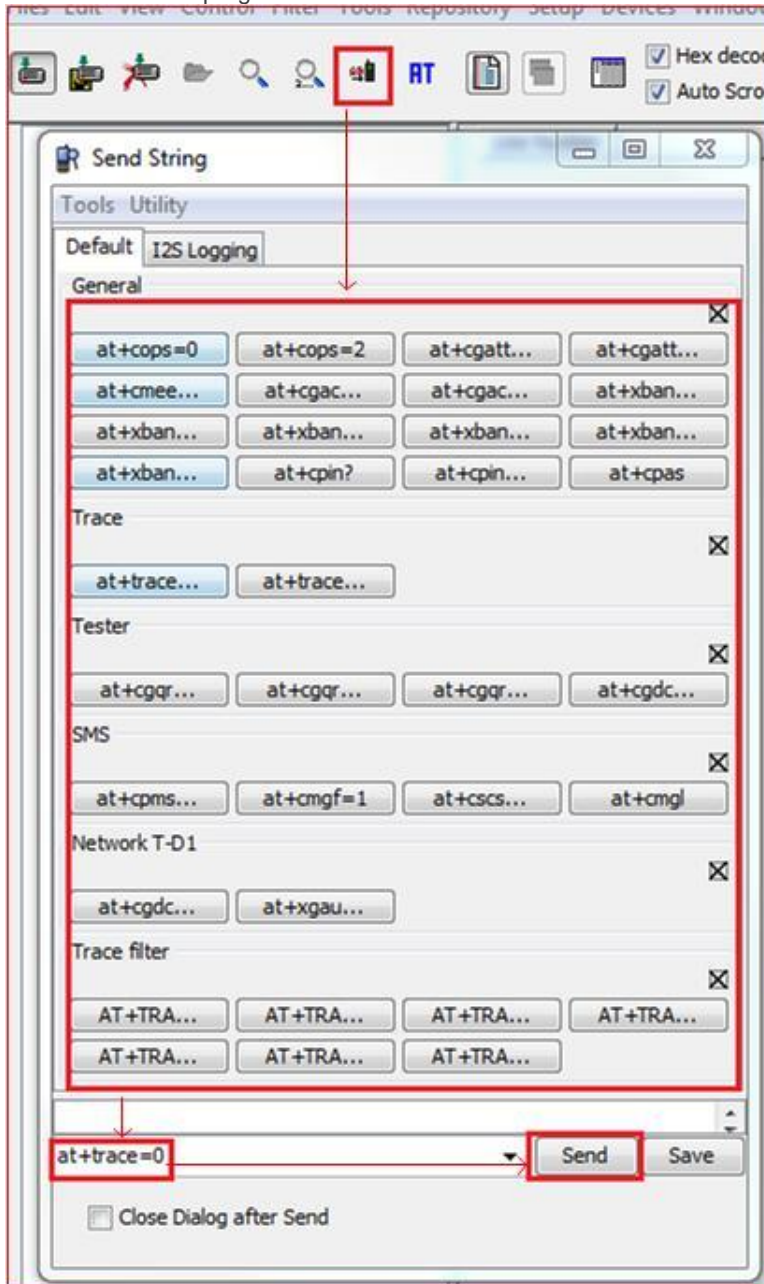
Instead of the AT port, the trace port could also convey AT strings to control the modem. AT commands can be sent on the trace port, relying on the built-in tool of MA.

Refer to the following steps for sending AT commands on the trace port.

1. Start a trace session.



- Open the AT panel using the button on the menu bar (as shown below) and choose the AT command to send; while a trace session is in progress.



8 Getting MA Traces on another OS through USB

As the MA tool only works on Windows, a simple program “tlog” (built with **tracelog.c**, see section 12) is applied to catch the logs (binary data) on USB, which are then decoded with MA on a Windows platform.

8.1 COM Port Configuration

An available port is a pre-requisite for taking traces. After connecting the modem via USB, check and match with the ports in */dev*.



```
$ ls -l /dev/ttyACM*
/dev/ttyACM0
/dev/ttyACM1
/dev/ttyACM2
/dev/ttyACM3
/dev/ttyACM4
/dev/ttyACM5
/dev/ttyACM6
```

Figure 15. USB Ports

Set the port in **tracelog.c**. Use “ttyACM1” for 2G trace and “ttyACM2” for 3G trace.

```
/* try ttyACM1 for 2G trace */
/* try ttyACM2 for 3G trace */

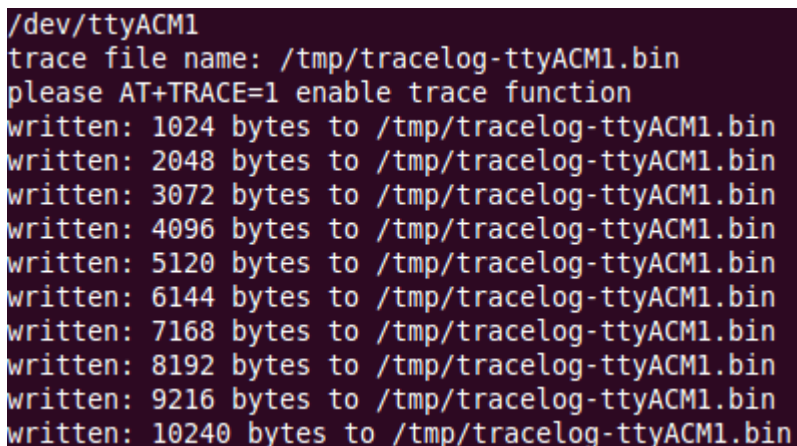
#define TRACE_PORT_NAME "ttyACM1"
#define BAUDRATE B115200
#define TRACE_FILE "/tmp/tracelog-%s.bin"
```

Figure 16. Set Trace Port

8.2 Start Tracing

With the ports configured as in section 7.1 Pre-Configuration, build the executable “tlog” with the OS compiler. If the port configuration and the connection are correct, a tracing session could be started by running “tlog”.

To start tracing data, send **AT+TRACE=1** via the AT port. Trace logs are then produced.



```
/dev/ttyACM1
trace file name: /tmp/tracelog-ttyACM1.bin
please AT+TRACE=1 enable trace function
written: 1024 bytes to /tmp/tracelog-ttyACM1.bin
written: 2048 bytes to /tmp/tracelog-ttyACM1.bin
written: 3072 bytes to /tmp/tracelog-ttyACM1.bin
written: 4096 bytes to /tmp/tracelog-ttyACM1.bin
written: 5120 bytes to /tmp/tracelog-ttyACM1.bin
written: 6144 bytes to /tmp/tracelog-ttyACM1.bin
written: 7168 bytes to /tmp/tracelog-ttyACM1.bin
written: 8192 bytes to /tmp/tracelog-ttyACM1.bin
written: 9216 bytes to /tmp/tracelog-ttyACM1.bin
written: 10240 bytes to /tmp/tracelog-ttyACM1.bin
```

Figure 17. Trace Log

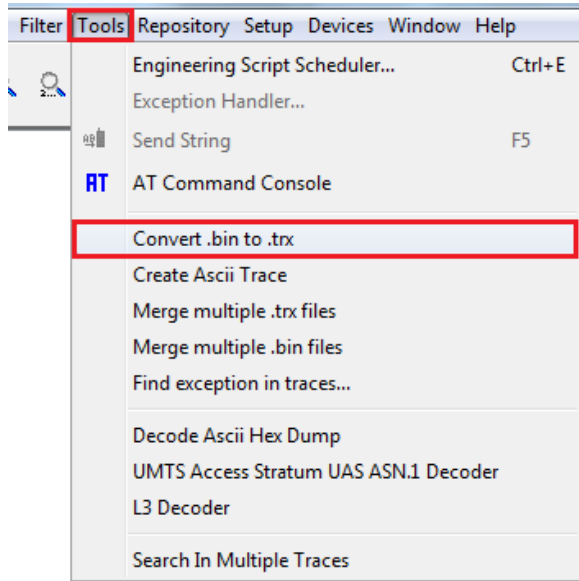
8.3 Stop Tracing and Decode Trace Log with MA

To stop the modem from sending trace data, send **AT+TRACE=0**.

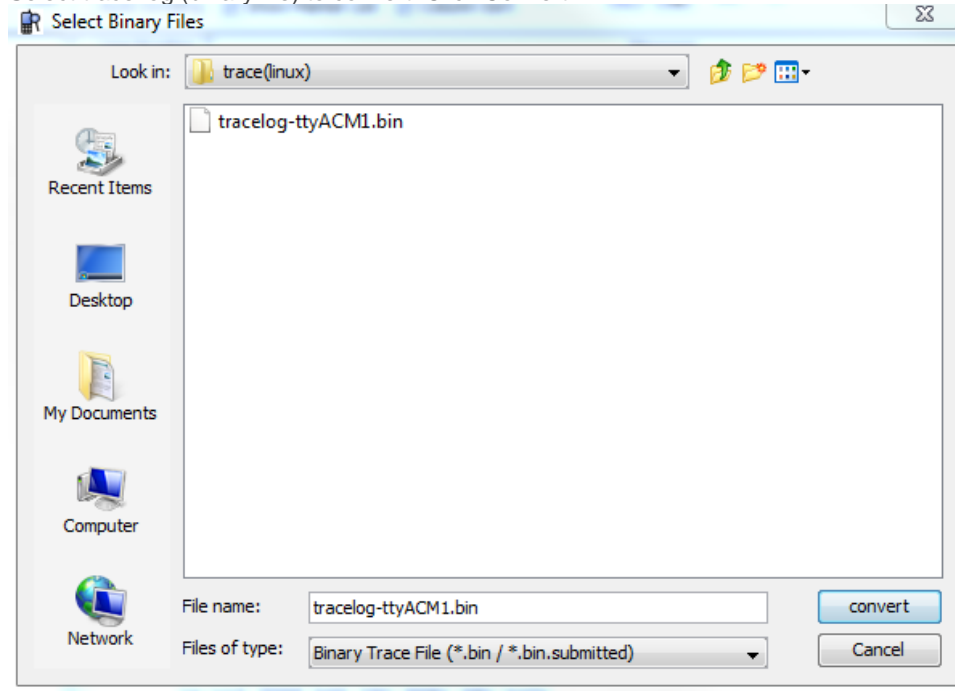
Stop the tracing session using an interrupt (e.g. Ctrl+C) and find the tracelog under the file path defined in variable "TRACE_FILE" in "tracelog.c" (i.e. "/tmp/tracelog-ttyACM1.bin").

A Windows platform PC is required to run MA in order to decode the binary trace log caught in the previous steps by converting the ".bin" format trace log to a corresponding ".trx" trace file. To convert file formats, follow these steps:

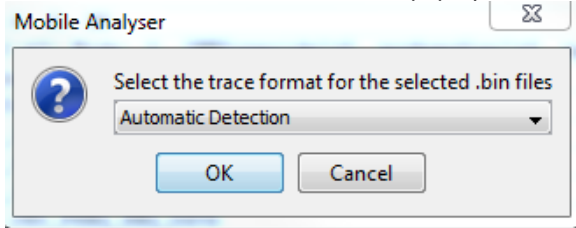
1. Click **Tools > Convert .bin to .trx**.



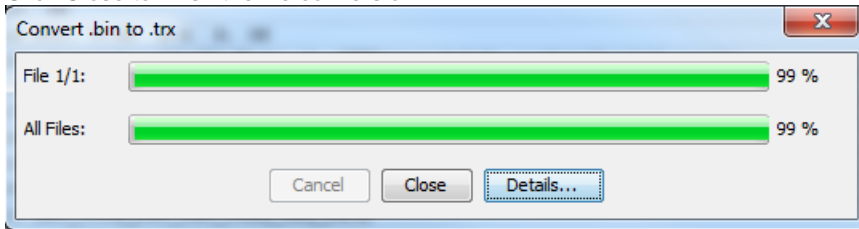
2. Select trace log (binary file) to convert. Click **Convert**.



3. Select "Automatic Detection" from the pop-up window. Click **OK**.

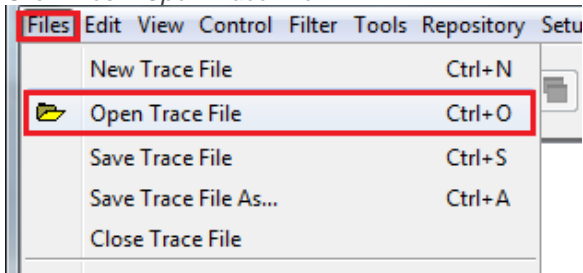


4. Click Close to finish the file conversion.

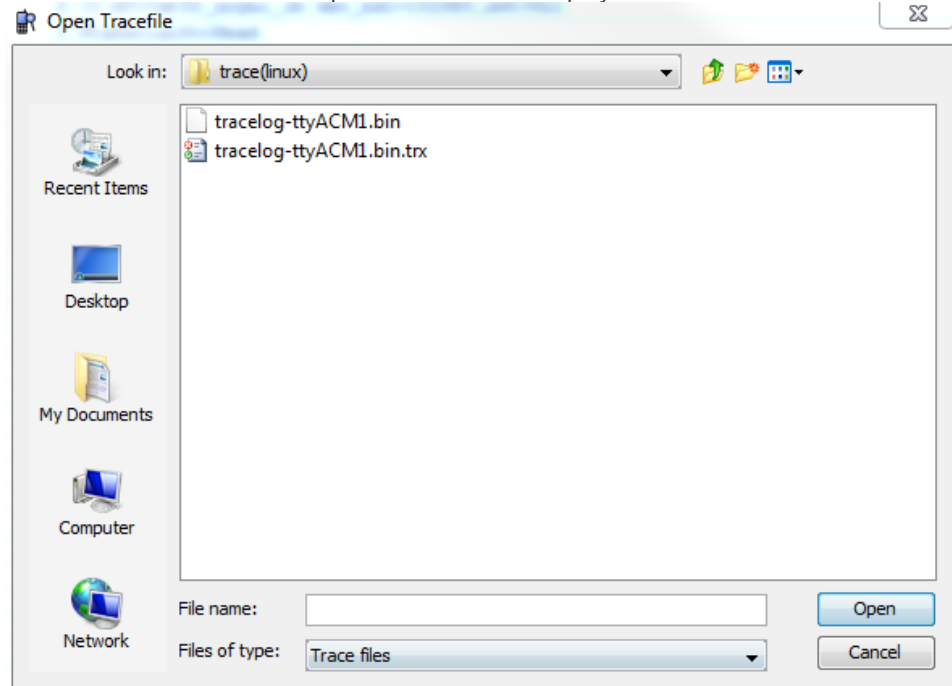


Note: Converted ".trx" files are created in the same directory as the ".bin" files. Load the ".trx" file with MA to display the trace content.

5. Click **Files > Open Trace File**.



6. Select the ".trx" file and click Open to decode and display the trace content.



9 Getting MA Traces on another OS through UART

9.1 COM Port Configuration

With the modem connected via UART, set TRACE_PORT_NAME in tracelog.c in accordance with that in /dev/.

1. Find the UART port in /dev.

```

/dev$ ls tty*
tty      tty14   tty20   tty27   tty33   tty4     tty46   tty52   tty59   tty8
tty0     tty15   tty21   tty28   tty34   tty40   tty47   tty53   tty6     tty9
tty1     tty16   tty22   tty29   tty35   tty41   tty48   tty54   tty60   ttyS0
tty10    tty17   tty23   tty3     tty36   tty42   tty49   tty55   tty61   ttyS1
tty11    tty18   tty24   tty30   tty37   tty43   tty5     tty56   tty62   ttyS2
tty12    tty19   tty25   tty31   tty38   tty44   tty50   tty57   tty63   ttyS3
tty13    tty2     tty26   tty32   tty39   tty45   tty51   tty58   tty7     ttyUSB0
  
```

2. Set the UART port in travelog.c.

```

16 #define TRACE_PORT_NAME "ttyUSB0"
17 #define BAUDRATE        B115200
18 #define TRACE_FILE      "/tmp/tracelog-%s.bin"
  
```

3. Build the executable file "tlog" with the OS compiler.

```

$ ls
Makefile README.txt tlog tmp tracelog.c
$ make
gcc -o tlog tracelog.c
  
```

9.2 Take Trace

Take trace by following these steps:

1. Send **AT+TRACE=1** to enable trace.
2. Send **AT+XSIO=11** to set UART as the trace port.
3. Reboot the modem using **AT+CFUN=1,1**.
4. Run "tlog" to catch the trace and follow the steps specified in section 8.3 Stop Tracing and Decode Trace Log with MA to decode the trace content.

This setting configures the UART as the trace port. After reboot, AT commands should not be sent with the AT terminal on the UART port.

10 Getting 3G Firmware Traces on Windows through USB

The 3G Firmware Trace Tool (3G FTT) is adopted in order to take 3G DSP traces.

10.1 COM Port Setup

The COM port should be set up in accordance with the COM port used as “IMC Port (USB2)” in Windows Device Manager. Taking section 6.1 COM Port Setup as an example, the corresponding port is COM27. COM port configuration should then be set as COM27.

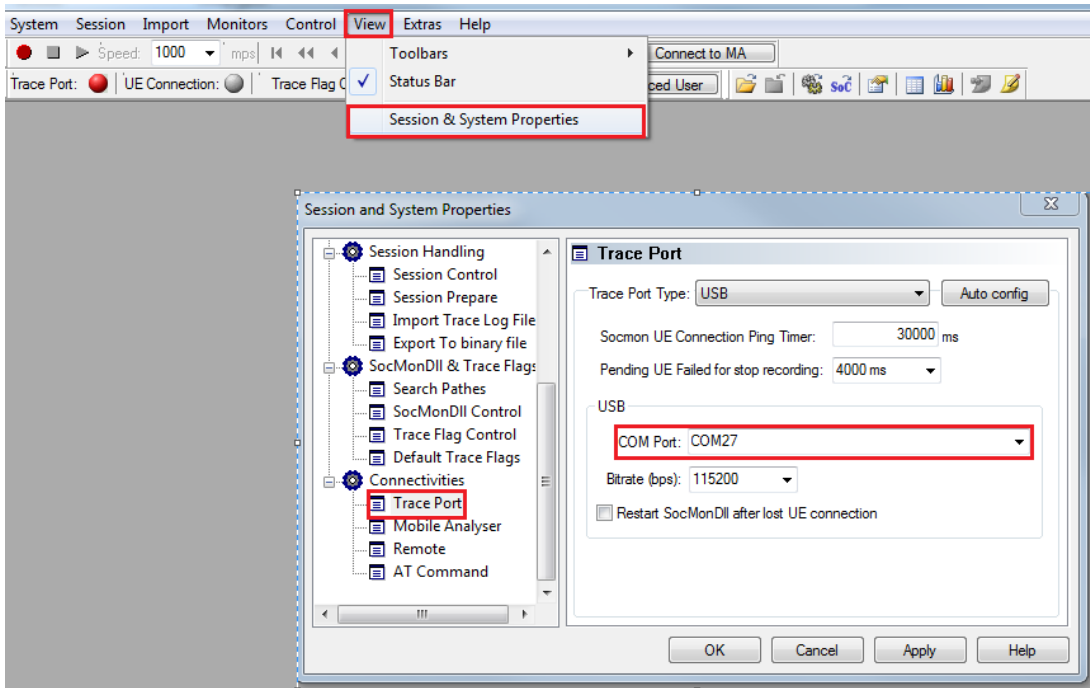


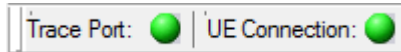
Figure 18. COM Port Setup

10.2 Start Tracing Using the “record” Button

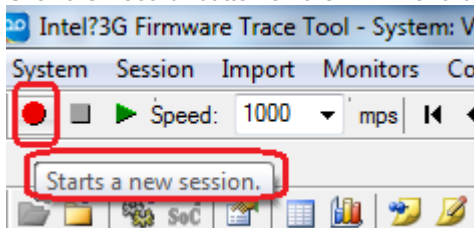
Follow these steps to start tracing.

1. Enabled tracing on the modem using the AT commands listed below.
`AT+XSYSTRACE=0,"3g_sw=1;bb_sw=1;digrf=1","digrf=0x84","oct=4"`
`AT+XL1SET="umts_fw_psvoff"`
`AT+TRACE=1`

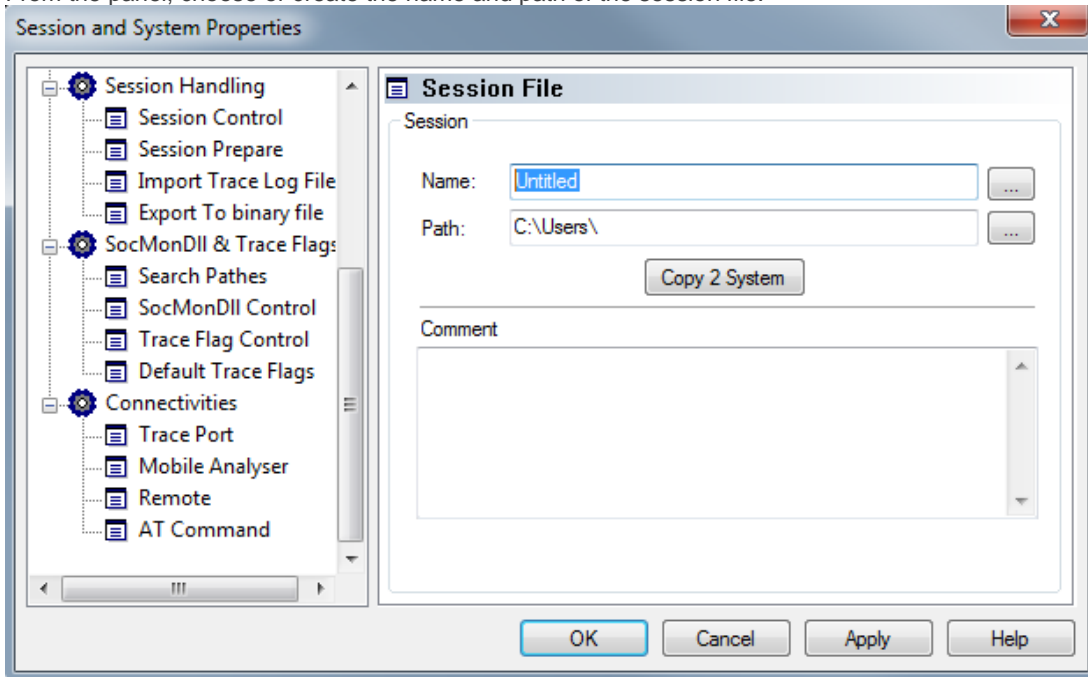
2. The UE connection should be available (displayed with a green icon) after step 1.



3. Click the “record” button on the FTT menu bar to start tracing.



- From the panel, choose or create the name and path of the session file.



10.3 Stop Tracing

Follow these steps to stop tracing.

- Send the following AT commands to stop the modem from sending trace data:

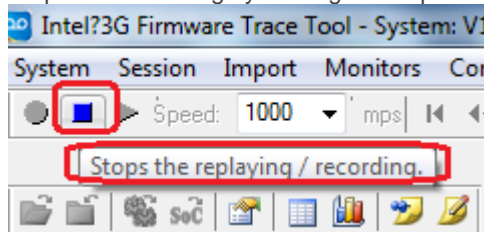
```
AT+XL1SET="umts_fw_psvon"
```

```
AT+TRACE=,,"ae=0"
```

```
AT+XSIO=0
```

```
AT+CFUN=1,1
```

- Stop trace recording by clicking the stop button from the menu bar.



- The trace file can be found in the path set in section 10.2 Start Tracing Using the "record" Button.

10.4 Monitors

Various monitors are available for displaying different tracing content as shown in the figures below.

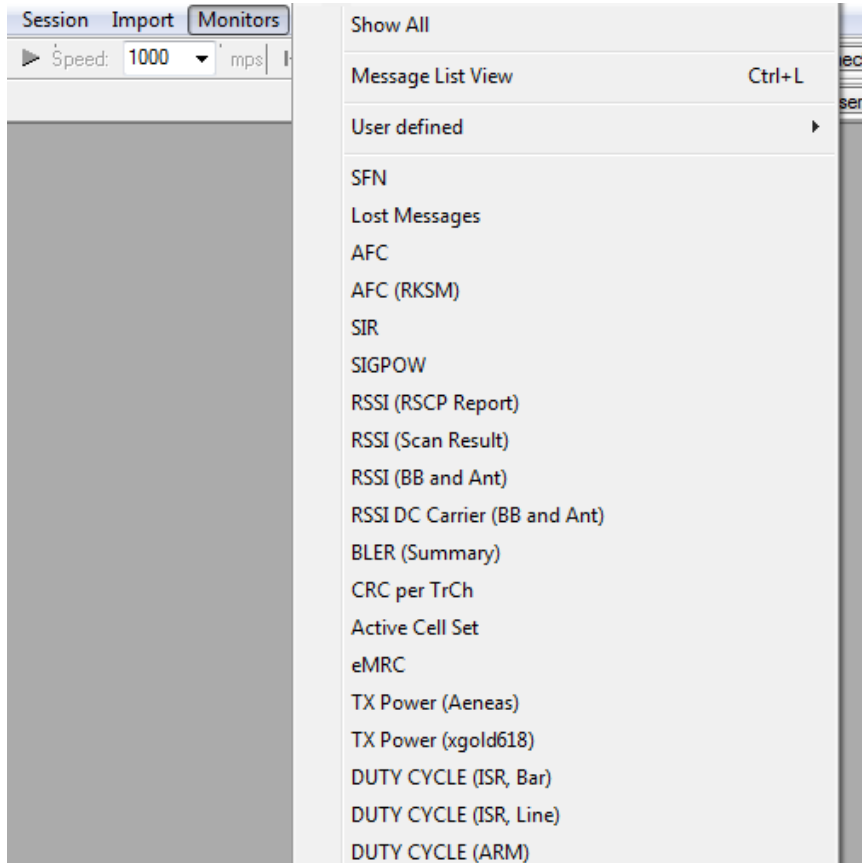


Figure 19. Available Monitors

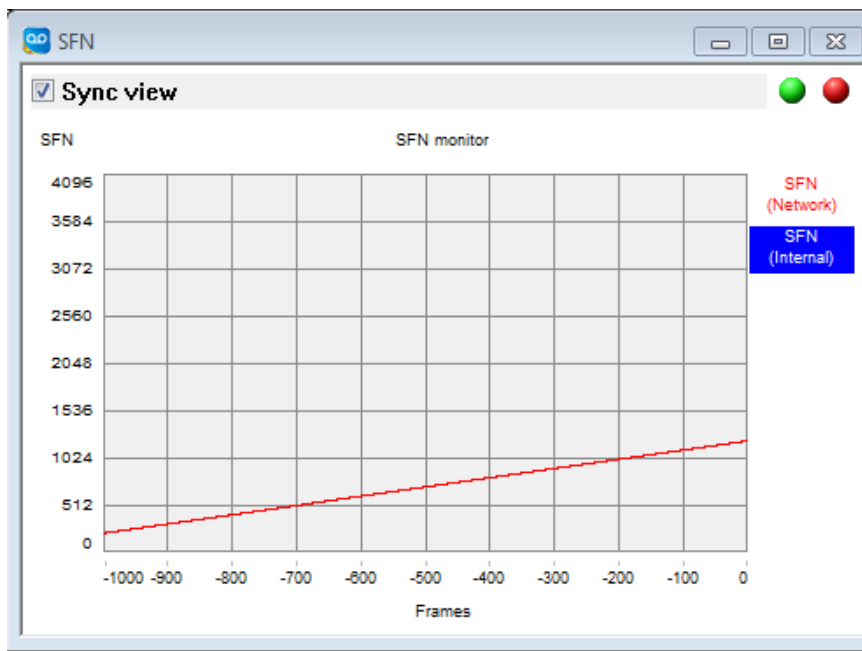


Figure 20. Trace Content Displayed using SFN Monitor

11 Getting 3G Firmware Traces on Windows through the Debug Port

11.1 COM Port Setup

Unlike using USB, the Debug port is a UART-like interface, hence it requires a different COM port configuration as shown in the figure below.

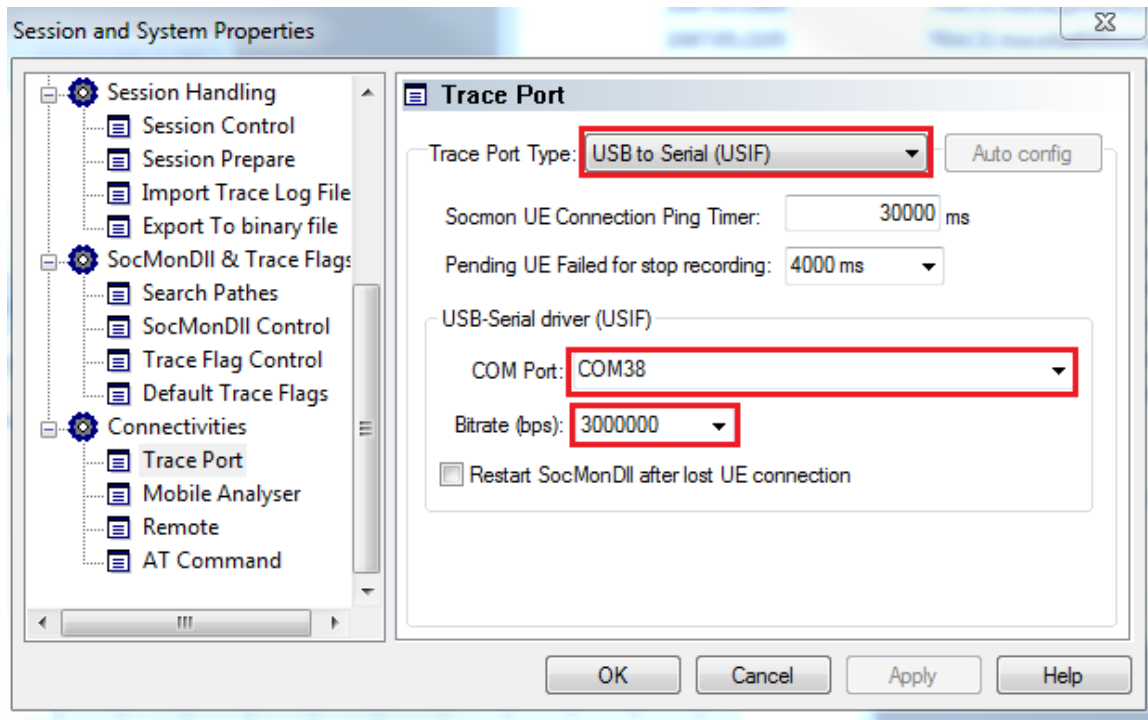


Figure 21. Debug Port Configuration

Based on Figure 14, the Debug port is COM38, so COM port is set to “COM38” in the figure above. The Trace Port Type should be USIF instead of USB; and the bitrate should be left at its default value of 3000000.

11.2 Start and Stop Trace

The following steps should be done prior to tracing:

1. Send `AT+XSIO=3` on UART to enable Debug for 3G debug trace.
2. Reboot the module by sending `AT+CFUN=1,1` on the UART.

To start tracing, send the following commands on the UART:

```
AT+XSYSTRACE=0,"3g_sw=1;bb_sw=1;digrf=1","digrf=0x84","oct=4"
AT+XL1SET="umts_fw_psvoff"
AT+TRACE=1
```

Once the steps above are completed, follow steps 2 to 4 of section 10.2 to take 3G DSP trace.

12 Tracelog.C

Tracelog.c is a sample C source code for capturing trace logon an OS other than Windows.



tracelog.c

13 Document History

Level	Date	History
1.0	January 15, 2015	Creation

14 Legal Notice

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where cellular modems are not advised without proper device certifications. These areas include environments where cellular radio can interfere such as explosive atmospheres, medical equipment, or any other equipment which may be susceptible to any form of radio interference. The Sierra Wireless modem can transmit signals that could interfere with this equipment. Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitations of Liability

This manual is provided "as is". Sierra Wireless makes no warranties of any kind, either expressed or implied, including any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The recipient of the manual shall endorse all risks arising from its use.

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Customer understands that Sierra Wireless is not providing cellular or GPS (including A-GPS) services. These services are provided by a third party and should be purchased directly by the Customer.

SPECIFIC DISCLAIMERS OF LIABILITY: CUSTOMER RECOGNIZES AND ACKNOWLEDGES SIERRA WIRELESS IS NOT RESPONSIBLE FOR AND SHALL NOT BE HELD LIABLE FOR ANY DEFECT OR DEFICIENCY OF ANY KIND OF CELLULAR OR GPS (INCLUDING A-GPS) SERVICES.

Patents

This product may contain technology developed by or for Sierra Wireless Inc.

This product includes technology licensed from QUALCOMM®.

This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group and MMP Portfolio Licensing.

Copyright

© 2015 Sierra Wireless. All rights reserved.

Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage®, WISMO® and the Sierra Wireless and Open AT logos are registered trademarks of Sierra Wireless, Inc. or one of its subsidiaries.

Watcher® is a registered trademark of NETGEAR, Inc., used under license.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh® and Mac OS X® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.



SIERRA
WIRELESS®