SIERRA WIRELESS

WCPU Debugging Tools

4111816-1.0

Author:	Sierra Wireless	Date:	February 22, 2012

APN Content Level	BASIC		INTERMEDIATE	✓	ADVANCED		Confidentiality	Public	✓	Private	

Hardware Compatibility	Product Line	AirPrime	Series	Q26xx	SL60xx
				WMPxxx	

Software Compatibility	Series	Q26xx : >7.x	
		OTHERS : 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 www.sierrawireless.com for latest versions.

2 Overview

There are currently multiple tools available to log the AirPrime Q268x, WMP and SL6xxx families to both debug Open AT applications and provide information to Sierra Wireleess technical support for indepth analyss of problems. This document outlines the tools available, how they should be used to provide information to Sierra and developers and also suggests filters to be using in common scenario's.

3 Tools outline

This section outlines the tools that are available and provides a high level description for each.

3.1 TMT

TMT is the original tool that is capable of logging all operations within the WCPU. This is no longer available on the Sierra Wireless website but if required can be obtained from your Sierra Wireless FAE.

3.2 Developer Studio

Developer Studio is the generic development environment that is used to develop an Open AT Framework-based application. The Target Management Perspective within the Studio enables the developer to select, view and record all traces coming from the unit. At the time of this document's publication, Developer Studio is not a full replacement for TMT but it is sufficient for debugging the majority of the application.

Additionally, Developer Studio enables the developer to insert break points into their application and single step through it viewing all variables in real time.

4 TMT

TMT is the original tool with which realtime logging of a WCPU is undertaken. It is Windows-based and is able to log everything in the system from SIM messaging to cell selection to Open AT application traces. Logging can be directed over either of the UART's or the USB. Logging data is started automatically when communication is started through the serial link manager and the PC and unit are synchronized (discussed later).

4.1 Workspaces

For each firmware that Sierra Wireless generates, a workspace is also produced. The workspace gives TMT details about the specific build of firmware i.e. EEPROM address, registers, etc.

There are a number of files that are contained in a workspace and are described below.

- DiagnoseTips.ini - Tooltips displayed for each trace level in Diagnose Property Page.
- MokaWatches.ini - Tooltips displayed in Watches panels.
- RemoteDefs.ini - Definitions of software elements, mailboxes and tasks
- Remtrace.csv - Remote traces (if the target supports them)
- Ram Symbols .s or s64k - RAM symbols
- E2p.h - EEPROM symbols
- E2p_csn.cso - Necessary to get the CSN1-encoded details of an EEPROM variable
- Register.ini - Registers
- Sys_ftr.ho - Features

- Backtraces.axf – Required to decode ADS Backtraces

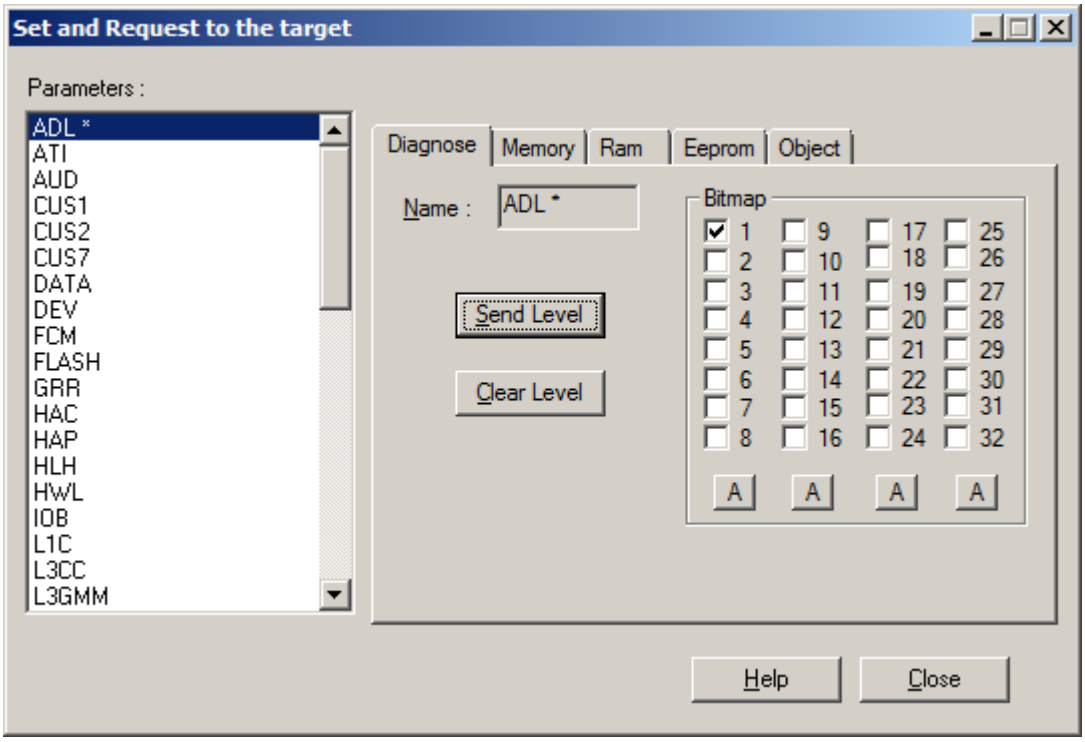
All of the above are generally delivered in two files and when loading a workspace loaded into TMT automatically.

- .wks – Contains path details for the above and basic TMT setup details.
- .zip – Contains all of the above files which the wks file points to.

If you need to do anything more than loading a standard workspace you will be advised of what this is by your Sierra Wireless technical contact, as it is beyond the scope of this document.

4.2 Filters

The tool can log anything within the system but a filter needs to be activated to define what it is actually going to be requested from the unit. See section 0 describing how to start logging and the relevance of this and the following windows.



4.2.1 Getting information about the target

Prior to setting the levels on the unit, the button "Get information about target" needs to be clicked to perform the following operations:

- Currently set levels.
- Levels that are contained within the software but not in the workspace i.e. ATMGR within AirPrime Q26EX.

4.3 Logging

When taking logs for analysis by Sierra Wireless the following points need to be taken into account.

- Logging must be performed from power up of the unit being logged. A lot of information is output at this time which is not available at any other time and which provides an initial basis and context for the log file.
- Timing must be enabled.
- The path for all logs files are stored by default in the Traces directory of the installation i.e. C:\Program Files\Wavecom\Totem\Traces
- All trace files are created with a date stamp i.e. Trace0_08Mar2011.txt, there is no other identification therefore careful identification of files names need to be carried out.

4.3.1 Using TMT without flow control lines

It is highly recommended to use flow control However it is possible to use TMT without flow control, depending on the system. If it does not work then the serial port settings need to be changed for this.

The settings are contained in the RS_comm.ini file in the installation directory, the default contents are shown below:

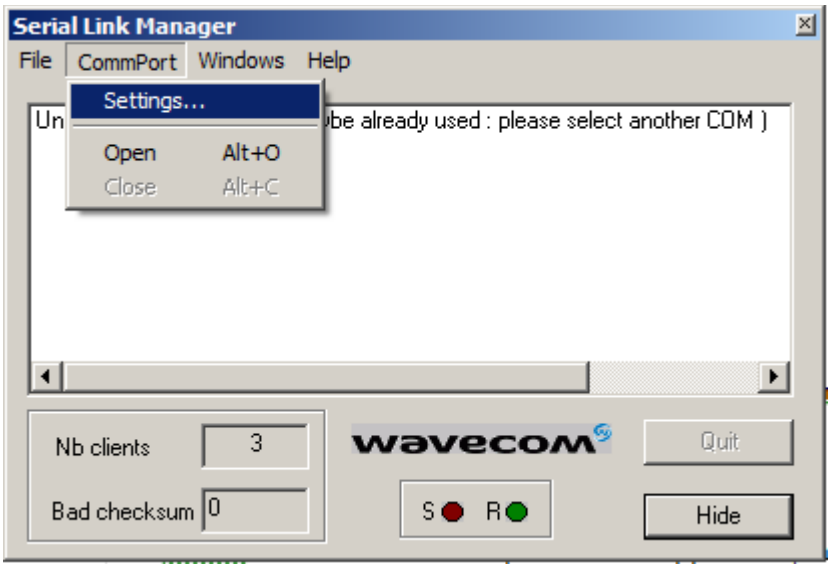
```
[RS_Settings]
Port=COM1
BaudRate=115200
DataBits=8
Parity=None
StopBit=1
Handshaking=Hardware
Power Control Enabled=0
```

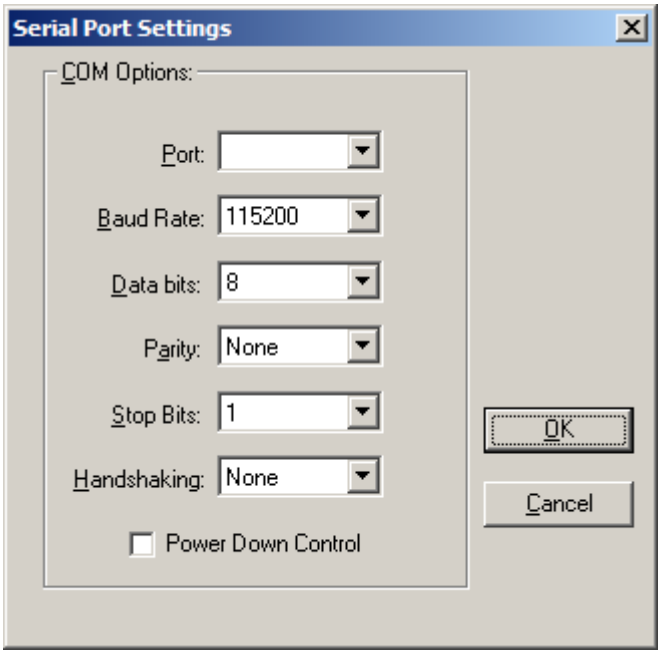
Note: The handshaking parameter needs to be changed to None.

4.3.2 Logging with 32KHz mode

32KHz mode is used to save power in the application, in this mode all interfaces (with the exception of select IO pins) are turned off, logging through TMT is not possible in this state. To allow logging while the unit is not in this state, the power down control needs to be turned on.

To do this open the serial link manager via its icon in the task bar (bottom right hand corner) and the CommPort – Settings window.





Select power down control. Tracing will then continue when the unit exits Low Power mode.

4.4 Surviving power cycles

TMT as previously described must have the interface over which it is communicating put into HAPC mode and synchronized to enable the debug information to flow correctly. When a unit is power cycled, the UART status is lost and, as a result, TMT needs to be re-synchronised. The UART can be forced into this mode by default.

4.4.1 AT command

Open AT Framework 2.34 and later contains a hidden AT command that has been implemented to allow the WCPU to automatically put either UART's 1 or 2 into HAPC mode allowing it to automatically synchronise with TMT (assuming it is connected to the specified interface):

```
AT+WDM=<port>
```

<port>	UART to use
1	Use UART1
2	Use UART2

The USB cannot be used for this operation due to the hardware drivers being reset during a power cycle, which would require the port to be redetected by the PC and the connection having to be re initialized at the PC end.

4.4.2 Downloadable dwl files

The same can be achieved using X modem downloadable e2p files, which are available from your Sierra Wireless technical contact. Below are the files that will do this for UART2.

- Tmt_synk_uart2 – Once downloaded and the unit is power cycled, UART2 will permanently power into HAPC mode.
- Tmt_synk.dwl – This file will return UART2 back to normal operation.

These two files are also available in the Attachments tab of this PDF file.

4.5 Basic procedure for starting logging

Below is the basic procedure to start the logging, referring to all of the previous sections.

1. Connect all hardware.
2. Confirm hardware interfaces are enabled (through AT+WMFM)
3. Select the correct PC serial port within TMT (F10 for settings – Com tab).

4. Load the correct workspace for the firmware being used on the unit.
5. Press ‘Synchronise’. 
6. Press ‘Get information about target’. 
7. Press ‘Get versions’. 
8. Press ‘Set and request’. 
9. Select the correct levels to use.
   After doing so this data should be visible coming out of the unit (assuming air interface (i.e. layer 3 radio resource which are regularily received) levels have been selected, some levels such as SIM operation will not be seen regularly).
10. Ensure the timing and saving tick boxes are enabled on the right hand side of the screen.
11. Power cycle the unit
12. Press ‘Synchronise’ again.
13. Confirm in the traces directory that the log file is being saved correctly i.e. file has been created and is growing as the traces are output on the screen.

Tip: *TMT can stop scrolling if there is a large amount of data output. Scrolling will never stop by disabling auto stop through Preferences – Settings – Misc (tab), ‘Disable traces scrolling autostop’. This is useful to ensure that logging has not stopped due to a power cycle.*

Tip: *As a default on installation, TMT will automatically force itself to check for its latest version on the internal Sierra Wireless drive. Uncheck this in Preferences – Settings – Startup (tab), then check the ‘Do not check for latest version on server’.*

5 Open AT Framework Libraries

Multiple parts of functionality released by Sierra Wireless firmware is available primarily as APIs within Open AT Framework. For customers who do not wish to work with Open AT Framework and purely want to use traditional AT commands access to this is given through a Library (aka “plug-in”) which is an Open AT Framework application. The Libraries currently available are listed below.

- JammingSoft (Jamming detection)
- Security
- WMMP
- SMS PDU
- Open SIM access (OSA)
- Internet (IP AT commands)
- C-GPS (Opus I, Opus II, Location Library)
- GR Library
- Extended AT commands

5.1 Internet Application and Internet Library activity and IP logging

Debugging information can be output by Internet Application through unsolicited responses. These are turned on through the AT+WIPCFG command and can be output on any interface regardless of which one the Internet Application AT commands are being received on.

5.1.1 Raw IP from TMT log files and conversion to Wireshark

Detailed logging of individual IP packets are enabled within the TMT data flows for both but these must be turned on through the APIs (for Internet Library) and AT commands (for Internet Library). To do so, see the following sections. Once the flows have been turned on it is possible to record a TMT log file containing the raw IP information. This can then be converted into a format which is readable by Wireshark.

To perform the conversion follow the procedure below.

1. Install an older version of SDK which contains the Open AT IDE (e.g. Open AT Framework 2.20).

2. Copy the file (tdump.awk, included in the Attachments tab of this PDF) to C:\OpenAT\IDE\IDE\1.06.04 (this version may change depending on the SDK version) \bin\
3. Open Open AT IDE by selecting : Start -> All Programs ->Wavecom ->Open AT IDE
4. Change the prompt path to C:\OpenAT\IDE\IDE\1.06.04\bin .
5. Copy the TMT trace.txt log file (containing the NET 10 levels) to the above location.
6. Execute: tdump.awk trace.txt trace.cap from the command line.
   The trace.cap file will be generated in the same location where the trace file is located.
7. Open the trace.cap with WireShark to analyse the traces.

At the time of this document's publication, the feature has been captured by Sierra Wireless product marketing and is intended for a future release of tools which will be made available to customers.

Note: In TMT to see the NET category when setting levels you have to hit the 'Get information about target' button when you have connected to the unit (and use the correct workspace for the firmware).

Note: IP communications will only be recorded for the sockets being opened, it does not include the initial negotiation of the IP address from the APN.

Wireshark is a commonly used protocol analysis tool which provides a user interface enabling engineers to look at problems more easily. Wireshark can be downloaded from the following link:

http://www.wireshark.org/

If running Wireshark, WinPCap will also probably be required.

http://www.winpcap.org/

5.1.1.1 Internet Library

Net level 10 is the filter that must be set to monitor the IP packets in the logging. This is enabled through the API as detailed below; further details can be found in the Internet Library manual.

In **wip_netInitOpts**, use the options:

- WIP_NET_OPT_DEBUG_PORT, WIP_NET_DEBUG_PORT_TRACE, WIP_NET_OPT_DEBUG_PKT_MAXLEN, 1500,

and in **wip_bearerSetOpts** API, use the option:

- WIP_BOPT_DEBUG_PKT, TRUE,

5.1.1.2 Internet Application

The below sequence shows the command required to enable the logging as described above and the net level 10 traces in TMT.

At+wipcfg=2,5,1	//WIP debug traces sent towards UART 1 as described above
At+wipcfg=2,15,1500	//Set WIP_NET_OPT_DEBUG_PKT_MAXLEN, 1500
At+wipbr=2,6,22,1	//Enable trace NET 10, this is unable to be sent prior to starting the bearer, if you do it will result in an error

The example below shows how to turn the responses on and what can be seen from a resulting successful connection.

```
at+wipcfg=2,5,1    //Turn debug information on using UART1
OK
at+creg?
+CREG: 0,1
OK
at+wipbr=1,6
OK
[GPRS]: open: -> DISCONNECTED
at+wipbr=2,6,11,"orangeinternet"
OK
at+wipbr=4,6,0
```

```
OK
[GPRS]: start: -> CONNECTING
[GPRS]: GPRS CTX (cid=1) found
[GPRS]: GPRS EVENT SETUP OK (cid=1): GPRS activate
[GPRS]: GPRS CTX (cid=1) found
[GPRS]: GPRS EVENT: 27 (cid=1)
[GPRS]: GPRS CTX (cid=1) found
[GPRS]: GPRS EVENT ACTIVATE OK (cid=1)
[GPRS]: GPRS: -> CONNECTED
at+cipcreate=2,1,"198.168.0.1",80
ERROR
at+wipcreate=2,1,"www.google.com",80
OK
[WIP] new TCPCLIENT 0x180c5e3c

+WIPREADY: 2,1
[ATWIP]: subevh_sock_write... 0
```

5.2 Other Libraries

Most other Open AT Framework Libraries do not have this level of debug capability as they have specific functions.

6 Developer Studio

Developer Studio is the fully integrated development environment for Open AT Framework with several perspectives implemented to aid the developer.

- **Open AT Framework**– All development, building and viewing of code is performed here.
- **Debug** – Used in JTAG and non GDB debugging (see following section).
- **Target Management** – Used for live tracing of applications and viewing of current unit status.
- **Packages Manager** – Used to control and manage the software packages currently installed within Developer Studio.

The following subsections will only describe the use of perspectives related to debugging of applications.

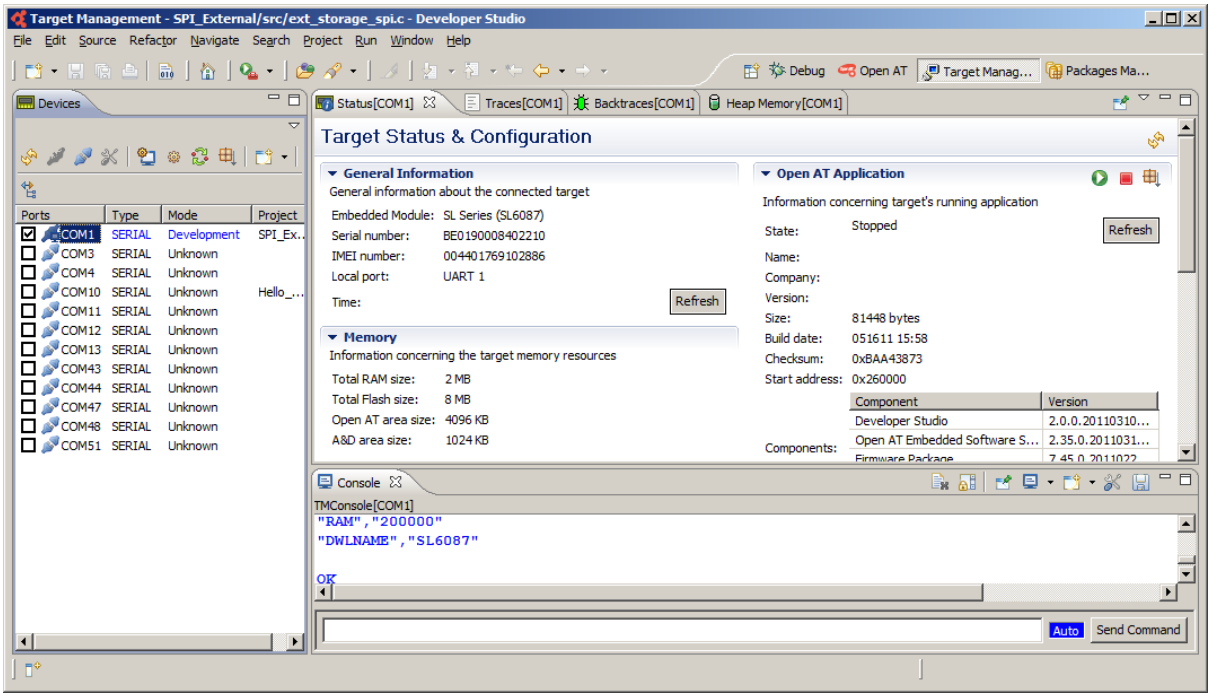
6.1 Target management perspective

The target management perspective is intended to allow customers to do the following:

- Take traces
- Download applications and firmware.
- Control the applications i.e., start, stop, delete, etc.

Within the perspective there are multiple tabs as follows:

- Status – Defines the status of the unit on the currently connected ports
- Traces – Shows traces from the currently running application. Typically these would be output from the application itself through the TRACE(()) API while shown through the ADL level.
- Backtraces - This view allows the developer to retrieve and decode backtraces from the connected target.



6.1.1 Notes

- This is not currently a replacement for TMT. Developer Studio does not have access to the debug files as previously described, and as such is unable to decode all traces from the unit in the same way that TMT is able to.
- Users are able to monitor and log multiple units simultaneously through multiple ports.

6.2 Using GDB

From firmware version R7.45/Developer Studio 2.0.0 and later, the firmware of the unit contains a GDB stub which enables the implementation of breakpoints in the code (normal signaling software continues to run in the background). Through this mechanism users are able to set break points either prior to running the application on the target or during runtime. The developer is then able to monitor variables, manually change them to test for specific scenarios, etc.

6.2.1 GDB limitations

The following are limitations to the GDB.

- There are two (2) breakpoints. The user interface implies there are more but trying to set more will result in unspecified behavior.
- When suspending an application, AT commands will stop responding, due to the fact that the application has stopped along with entire AT mechanism. It will return to operation once execution is resumed.
- When suspending a running application, it is impossible to see the call stack because it is still running in the background. (This is only visible for breakpoints.)
- Local variables' values cannot be displayed.
- No breakpoint can be set in the entry point of an application task.
- After terminating the GDB debugging, it is recommended to reboot the target.

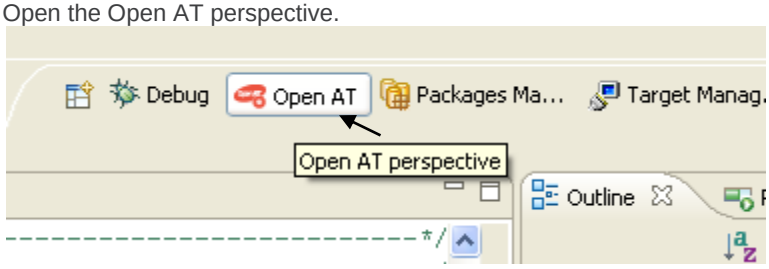
6.3 Setup for Developer Studio

6.3.1 Configuring the connection

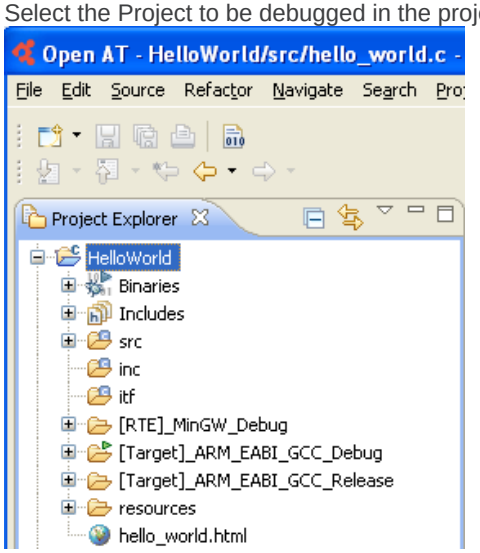
The following section describes how to create the connection inside Developer Studio.

This should only be required once per application.

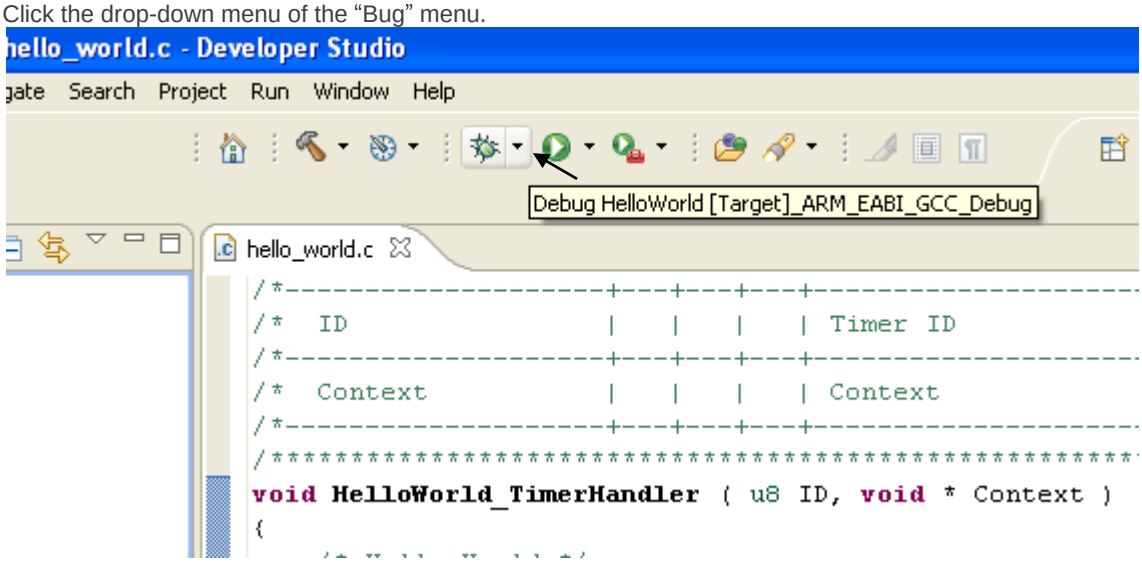
1. Open the Open AT perspective.



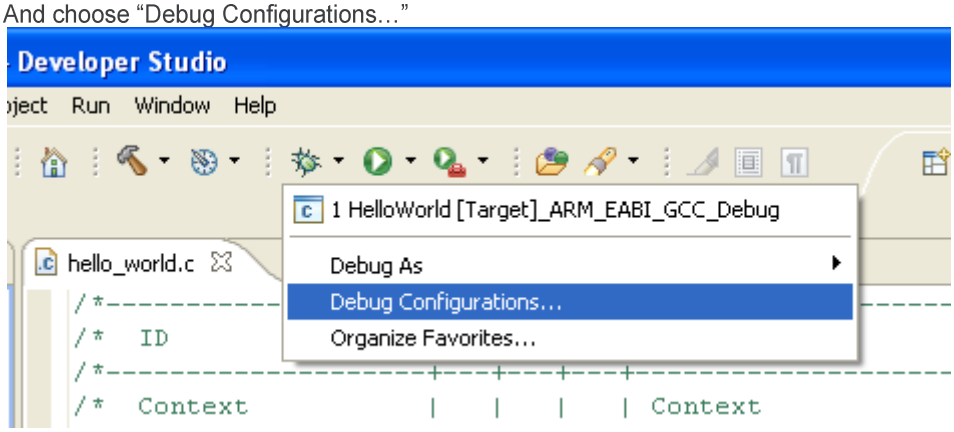
2. Select the Project to be debugged in the project explorer.



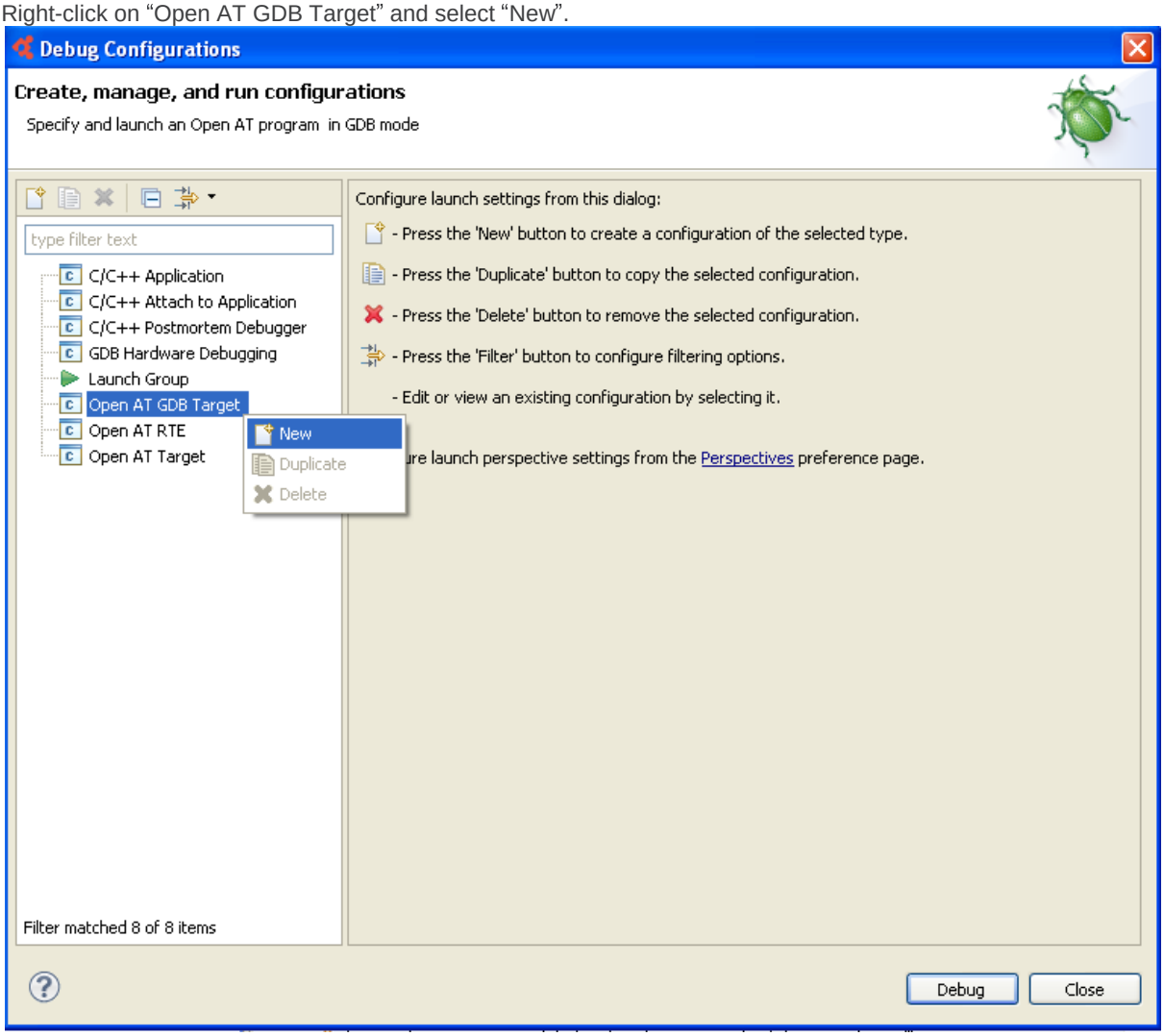
3. Click the drop-down menu of the “Bug” menu.



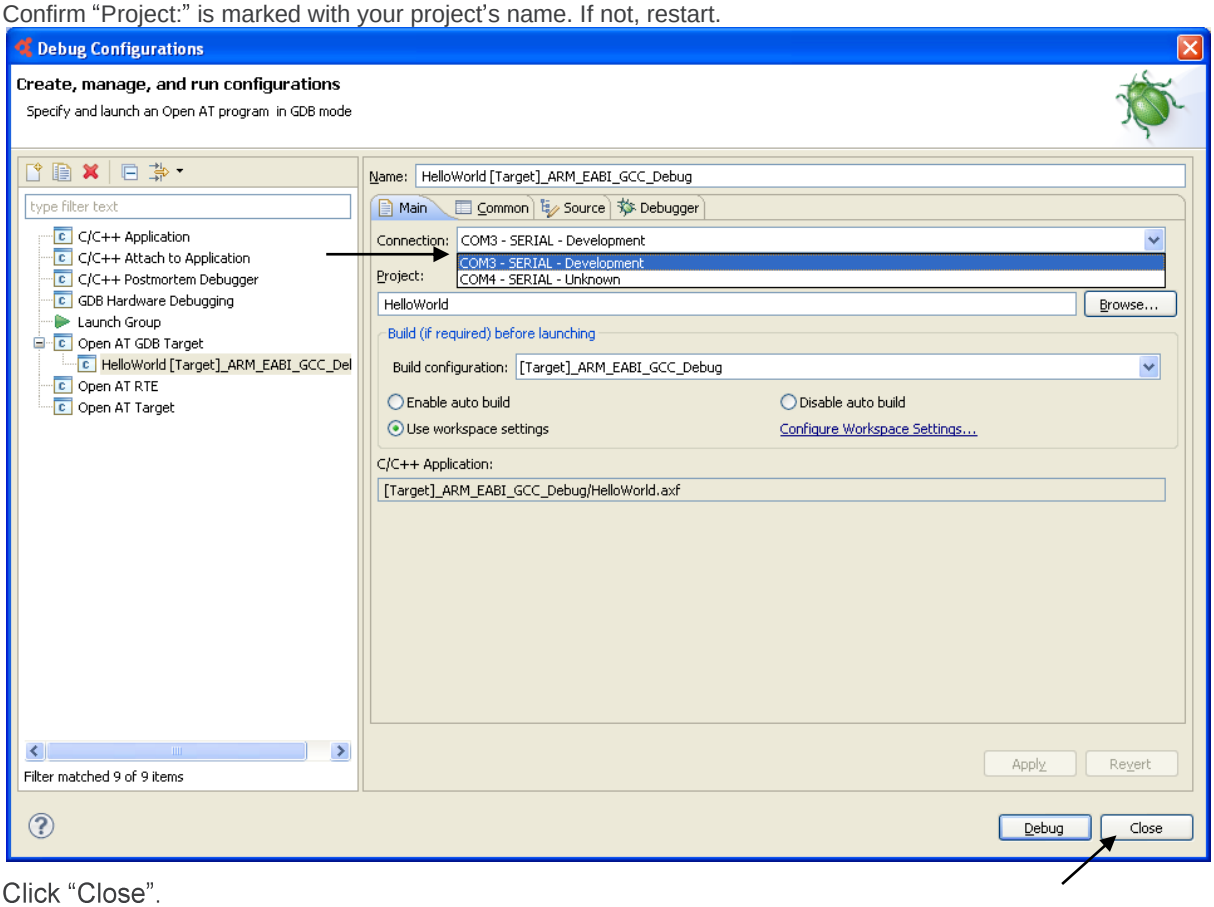
4. And choose “Debug Configurations…”



5. Right-click on “Open AT GDB Target” and select “New”.



6. Confirm that the port being used for AT commands and Traces is present.
   In this example it is COM3.
7. Confirm “Project:” is marked with your project’s name. If not, restart.



8. Click “Close”.

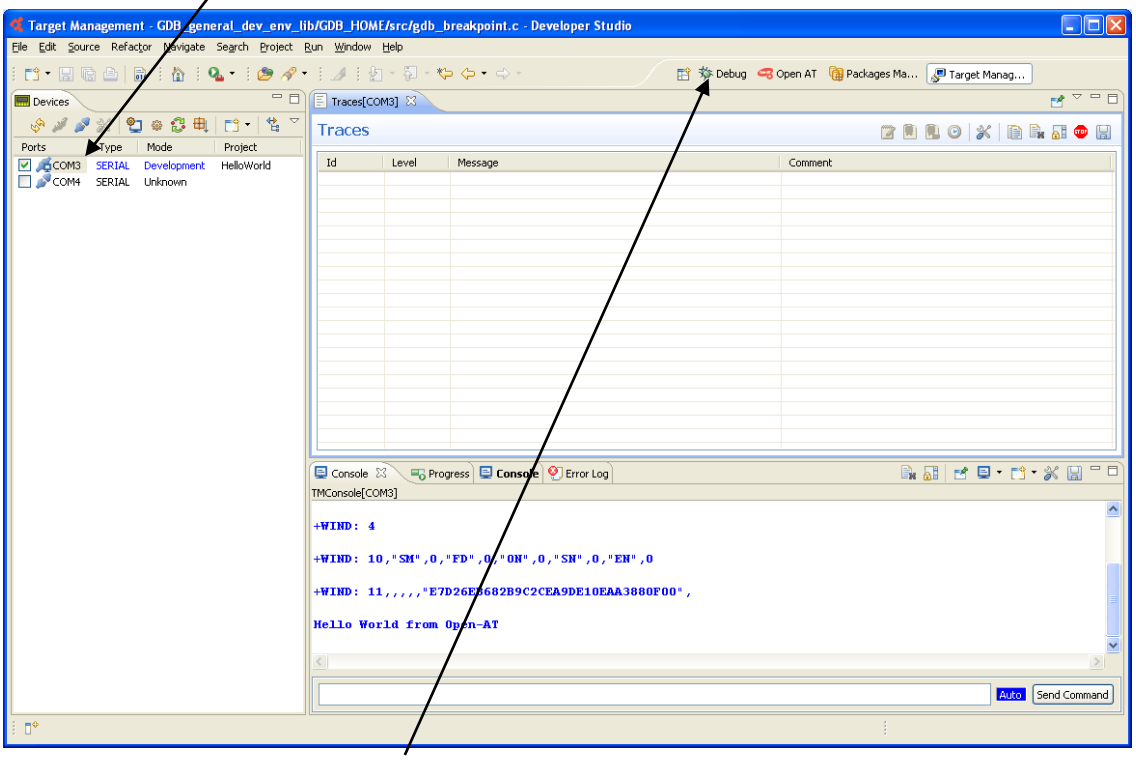
You have now configured the connection and it is saved to be used later.

6.3.2 Debugging with GDB

6.3.2.1 Connecting to target

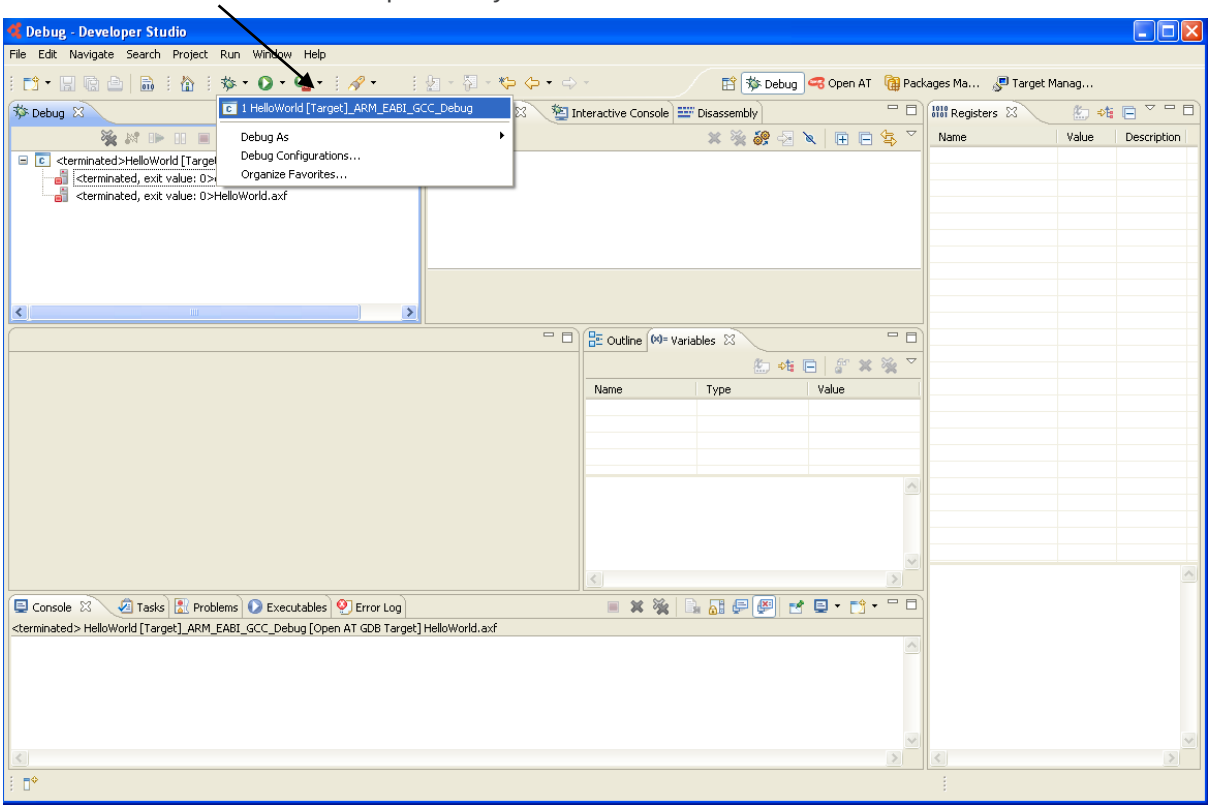
This section describes how to connect Developer Studio and GDB using the connection configured in the previous section.

9. Ensure AT commands and Traces within Developer Studio are up and running on the same port as specified for your connection (in this example, COM3)

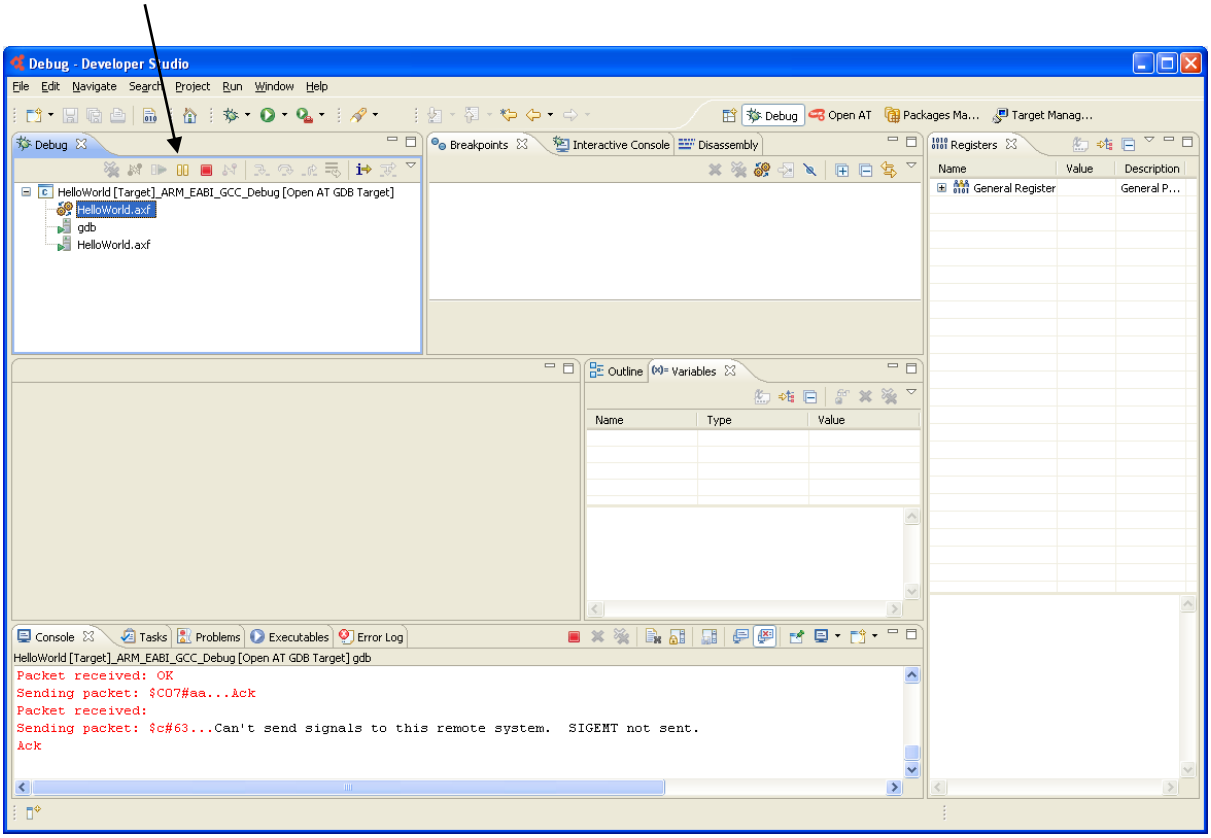


10. Click the “Debug perspective”.

11. Choose the connection that was previously created.



12. The unit should now be connected to the still running target and it should look like the figure below. Press “Suspend” to suspend the execution of the application.

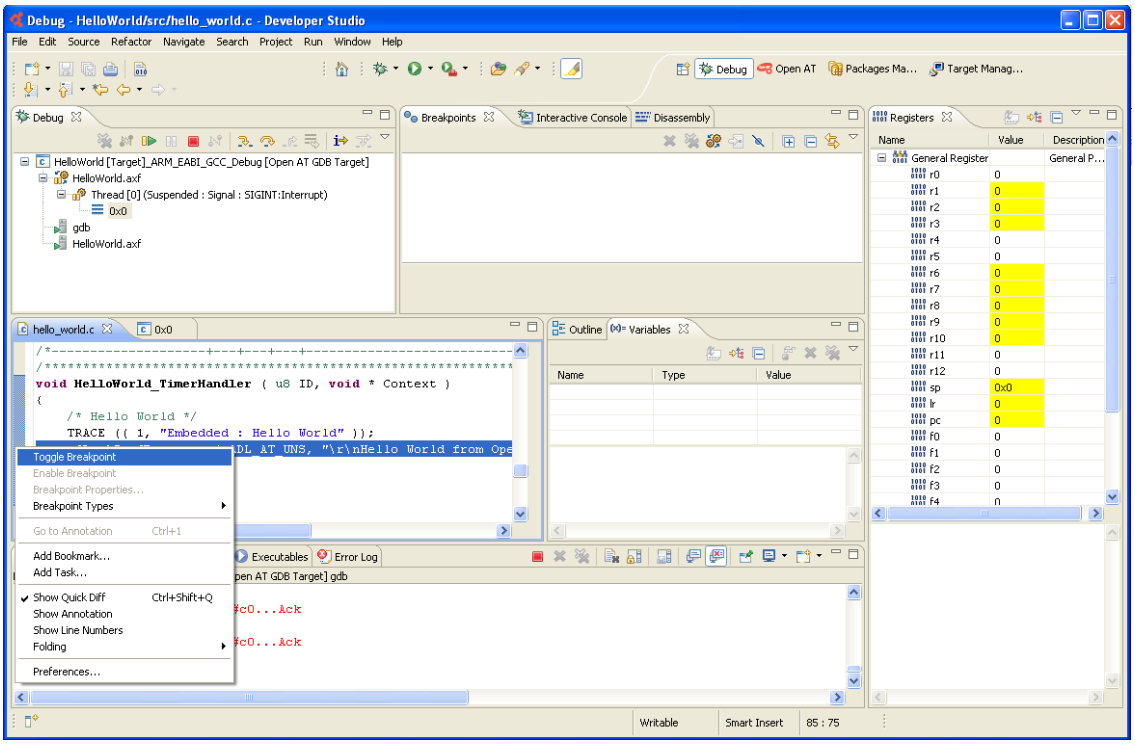


The application is now suspended.

It is not possible to see the call stack when just suspending the target.

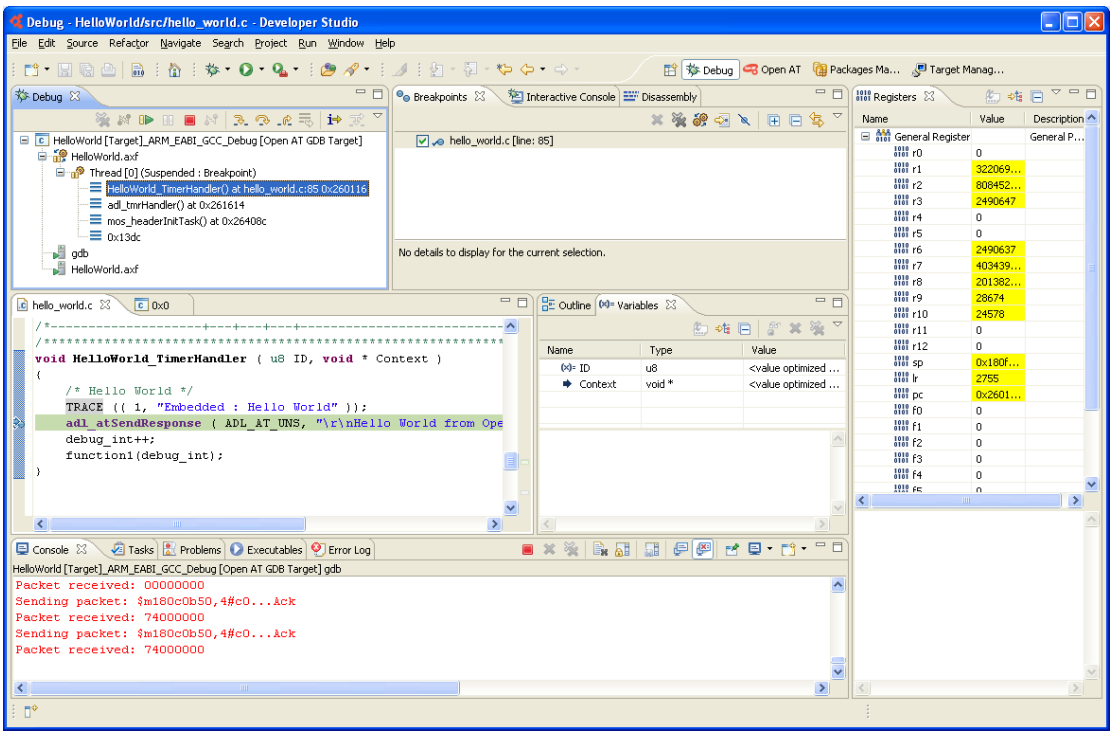
It is now time to set a breakpoint (it is also possible to set a break point in the Open AT perspective and then just run the target as descried in the previous steps).

13. Right-click (or double-click) on the left part of the code in which you want to put a breakpoint.



14. Set a breakpoint in your code and “Resume” the execution of the application.
    For the initial setup, choose code that you know will execute to be sure it will hit a breakpoint.

15. When you hit the breakpoint, you will see the call stack and more.



For a small overview of basic operations of the Developer Studio debugger interface review the next chapter.

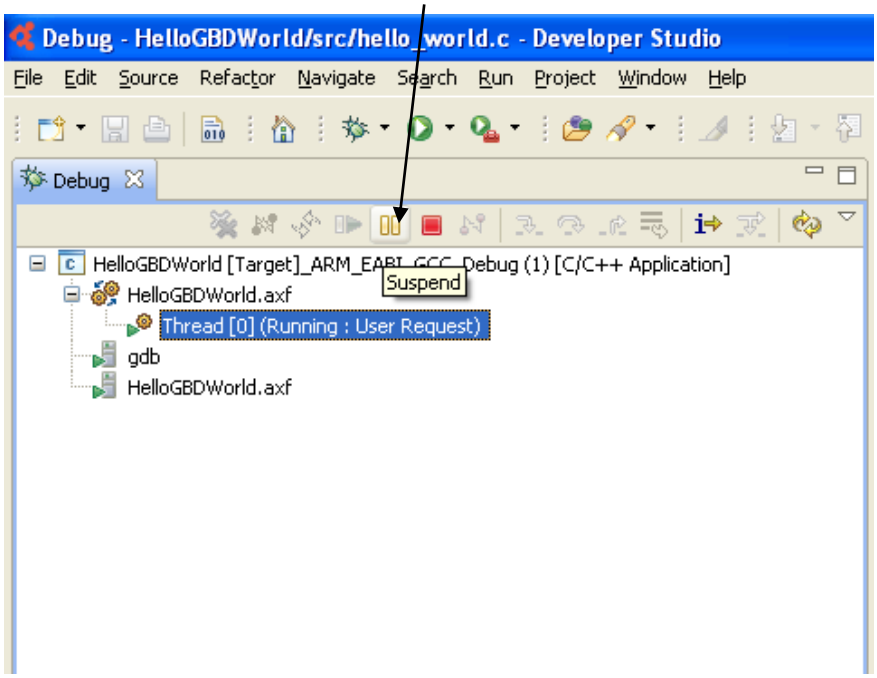
6.3.3 Overview of basic operations

6.3.3.1 Introduction

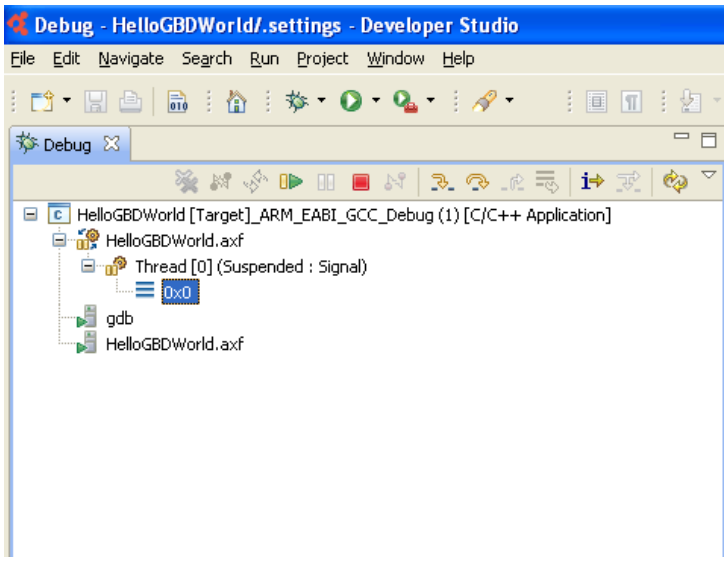
The following section is intended as a brief introduction to the basic operations of the GDB debugger.

6.3.3.2 Suspend a running target

If there are no previously set breakpoints in the Open AT perspective, the target will not stop (unless hitting a Data Abort or similar operation). In this case, press the “Suspend” button.



The following will then be displayed:

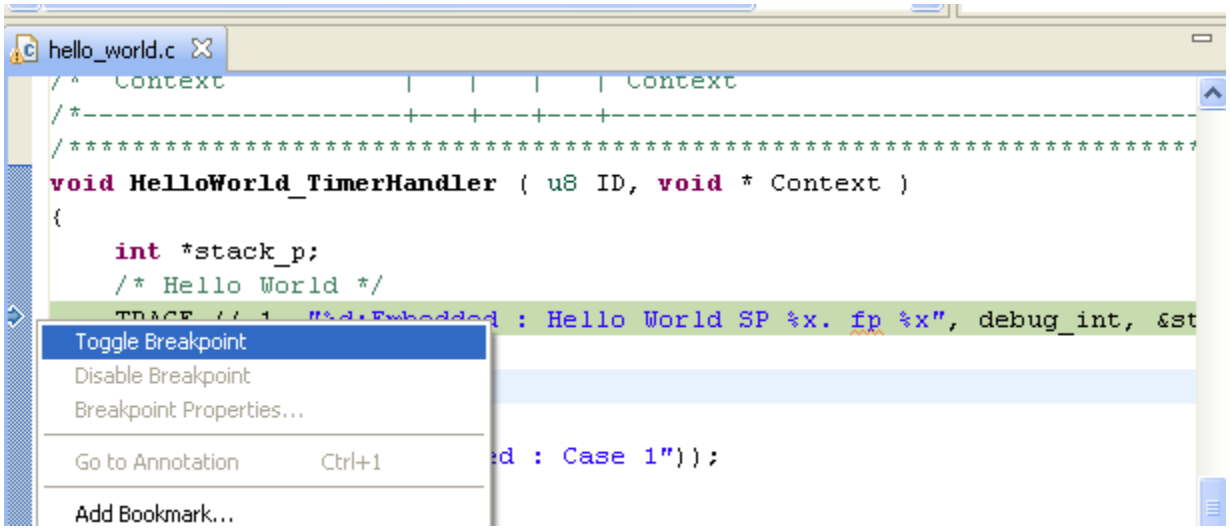


It is not possible to see where the application has stopped when doing “Suspend”. However, the developer is now able to set up to two (2) breakpoints.

6.3.3.3 Setting breakpoints

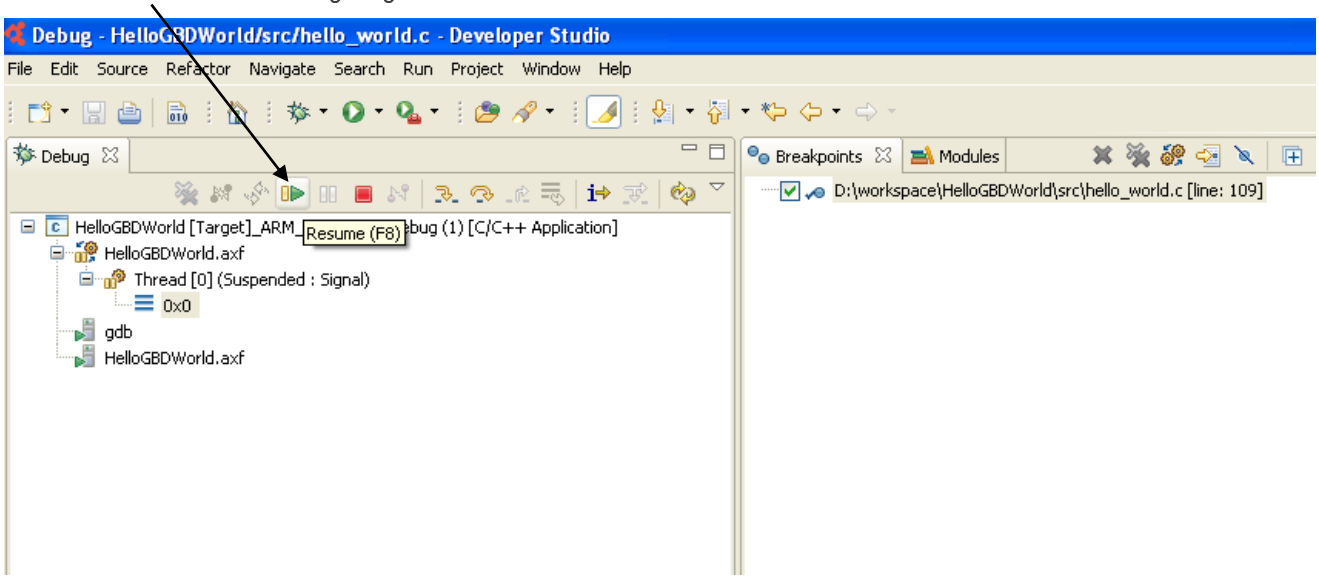
Right-click in your code and put the breakpoint where desired (or double-click the left border of the code window).

Note that Developer Studio remembers where breakpoints are inserted for the next run. Remember the limit of two (2) breakpoints.



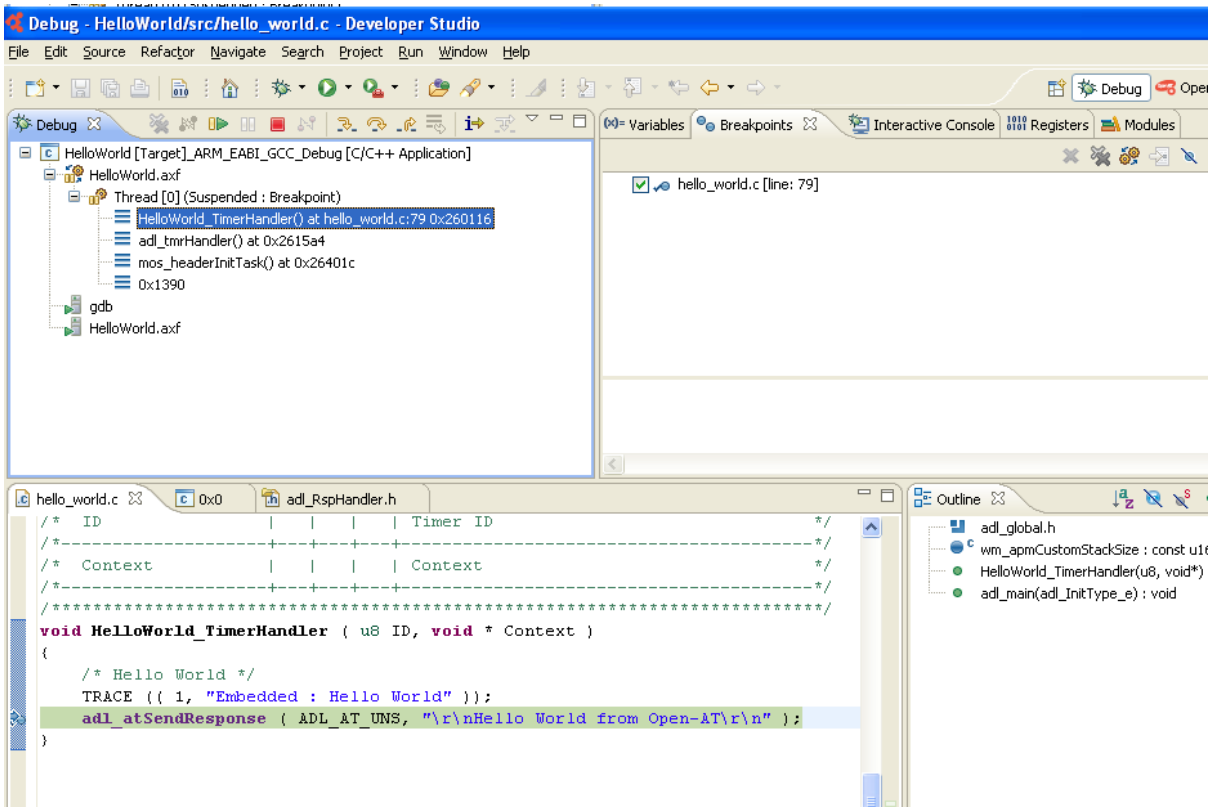
6.3.3.4 Resume

Press “Resume” to start the target again.



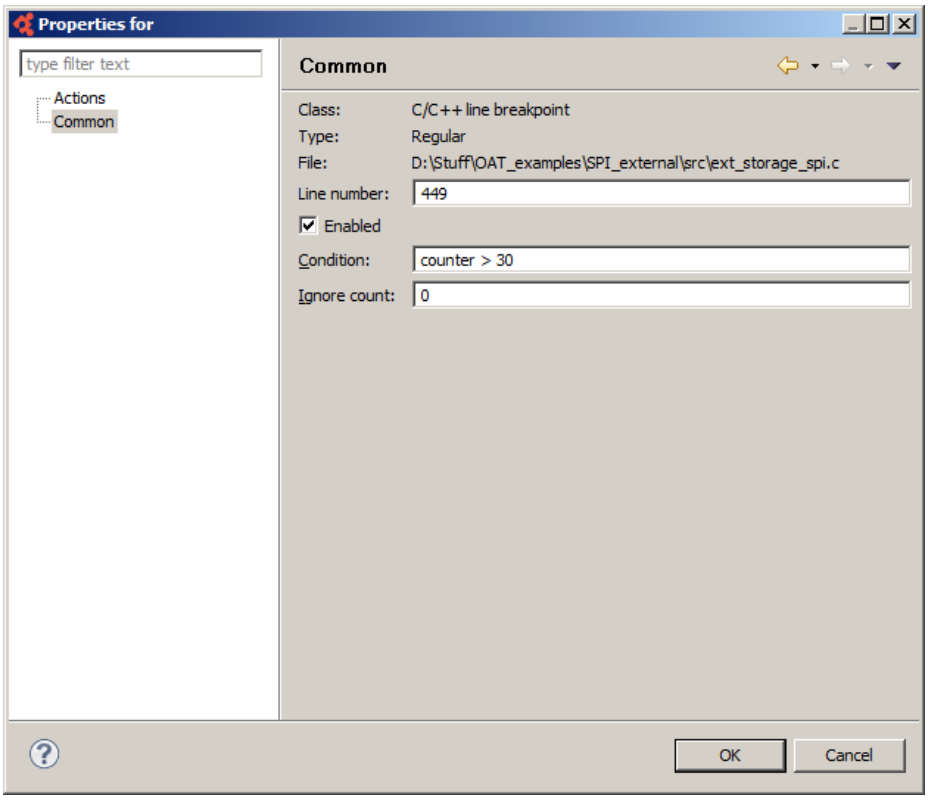
6.3.3.5 Hitting the breakpoint

After a while Developer Studio will return if the breakpoint is hit. The call stack will now be displayed.



6.3.3.6 Conditional break points

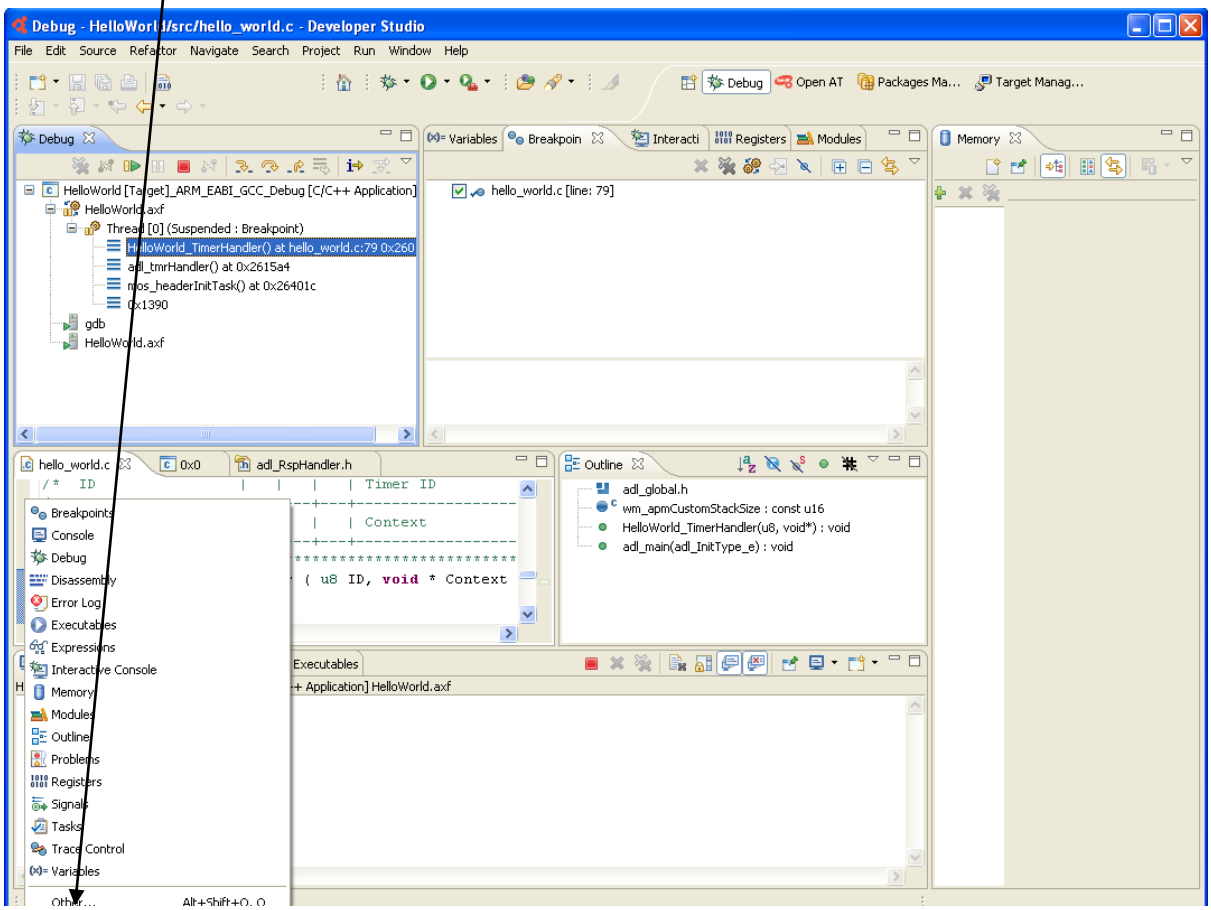
It is possible to set conditional break points i.e. stop when counter x reaches 30, through the properties window of the breakpoint.



6.3.3.7 Other views and accessible information

This document cannot cover all perspectives and views available. Click on the menu on the left corner to access available information found in normal debuggers.

If further help is required please contact your normal support channels.



7 Appendixes

7.1 Suggested filters

Below are suggested filters for commonly reported issues and scenarios.

Scenario/Issue	Filter
Basic	ATI [1] L3RR[3;4] L3MM [3;4] L3GMM [3;4] L3CC [3;4] L3SMG [3;4] MMT [3;4]
Mobile unable to perform network GSM and/or GPRS attach	ATI[1] L3SMG [3;4] L3GMM[3;4] L3MM[3;4] L3RR[3;4;14]
Mobile unable to activate a GPRS session	ATI[1;5] MMT [3;4;16] L3SMG [3;4] L3GMM[3;4] LLC[3;4] L3RR[3;4] PPS [1;3;4;6;14]
Mobility management	L3GMM[3;4] L3MM[3;4] L3RR[3;4;14]
PLMN selection	L3RR[3;4;5;14] GRR[9;10] L3MM [3;4] L3GMM [3;4]
Cell selection	L3RR[3;4;14;17]
Neighbor cell list is not the one expected as in SI2/2bis/2ter or SI5/5bis/5ter or PSI3/PSI3bis and in PacketMeasurementOrder/PacketCellChangeOrder	L3RR[3;4;17;18]
C1, C2, C31 or C32 criteria	L3RR[3;4;14;15]
Wrong PSIlistening	L3RR[3;4;16]
Neighboring cell management	L3RR[1;3;4;17;18]
Data transfer	L3RR[3;4] RLU[1] RLD[8] DATA[26] GRR[7;8;19;20]
Bad message reception on CCCH or PCCCH	L3RR[3;4] GRR[3;4;13]
Transfer problems on PACCH	L3RR[3;4] GRR[19]
Bad transfer rate	L1C[9;10] DATA[26]

Scenario/Issue	Filter
Bad RLC/MAC control blocks reassembly	L3RR[2;3;4]
Missing messages	L1C[5]
Hopping problems	L3RR[3;4;9;19]
SMS	ATI[1 ;7] MMT [3;4;7] L3SM [3;4] L3GMM[3;4] L3MM[3;4] L3RR[3;4] LLC [3;4]
Voice Call	L3RR[3;4] MMT [3;4] L3CC [3;4] L3MM[3;4] ATI [1]
CSD call (Circuit switched data)	L3CC [3;4] DATA [1;2;6;7] L3RR [3;4] ATI [1;2;4;5] MMT [3;4;5]
Internet Application and Internet Library (see previous section)	Net [10]
SIM interface	SIM [1;2;3;4;5;6;7;8;13;14}
Q26EX (Confirm 'Get information about target' is selected before sending and receiving levels)	ATI [1] AUD [2;5;8;19] L3RR [3;4] ATMGR [13] MMT [3;4] L3CC [3;4]
PPP	ATI [26] MMT [3;4;27] SNDCP [3;4;5;6] L3RR [3;4] L3SMG [3;4] GRR [16;19]
CHAP	ADL All ATI [1;2;4;5] NET All DATA All L3GMM [3;4] L3MM [3;4] L3SMG [3;4]

7.2 RTK exception cause values

The table below lists the RTK exception cause values that would be seen from the stored EEPROM backtraces.

RTK Exception	Value	Cause
RTK_EXCPT_EVENT_OUT_OF_MEMORY	20	Event creation error : Out of memory
RTK_EXCPT_BAD_HANDLE	21	Bad event handle
RTK_EXCPT_NOT_ALLOWED	22	Call of function @ ref RTKEventWait not allowed in idle task
RTK_EXCPT_NOT_FOUND	23	Event not found in event list
RTK_EXCPT_BAD_WAIT_MODE	24	Bad wait mode
RTK_EXCPT_NON_ALLOWED_REGION	25	Not allowed region for wait event
RTK_EXCPT_TIMER_USED_ERROR	26	Timer identifier already used
RTK_EXCPT_EVENT_TIMER_ERROR	27	Task not found in event task list
RTK_EXCPT_SLEEP_BAD_PERIOD	80	Not allowed to put to sleepa tast for a null period
RTK_EXCPT_SLEEP_BAD_TASK_ID	81	Not allowed to put to sleep an idle task
RTK_EXCPT_SLEEP_TIMER_USED_ERROR	82	Timer identifier already used
RTK_EXCPT_BAD_TIMER_TYPE	89	Bad timer type (internal error)
RTK_EXCPT_NO_TIMER_AVAILABLE	90	No more available resources to start timer
RTK_EXCPT_BAD_SEM_TIMER_PERIOD	91	Not allowed call RTKSendConsumeDelay with null delay
RTK_EXCPT_BAD_TIMER_PERIOD	92	Not allowed to start a timer with null period
RTK_EXCPT_CLEAR_TIMER	94	Internal error : unable to clear an expired timer
RTK_EXCPT_SEM_TIMER_ERROR2	95	Any task is waiting for this semaphore
RTK_EXCPT_BAD_TIMER_MAILBOX_NUMBER	99	Bad mailbox identifier value (RTK10_Timer extract)
RTK_EXCPT_BAD_TIMER_MAILBOX_NUMBER2	100	Bad mailbox identifier (RTKStart timerFacto)
RTK_EXCPT_BAD_TASK_STATE	110	Bad task state (init)
RTK_EXCPT_BAD_TASK_NUMBER	111	Bad task ID (RTKGetTaskState function)
RTK_EXCPT_TASK_NOT_INITIALISED	113	Bad task state (init)
RTK_EXCPT_TASK_ALREADY_STARTED	114	Bad task state (init)
RTK_EXCPT_NON_ALLOWED_OPBGT	116	Idle task is not allowedto wait for a message on a mailbox
RTK_EXCPT_TO_MANY_TASK	117	Too many tasks
RTK_EXCPT_NO_STARTABLE_TASK	118	No startable task
RTK_EXCPT_COUNTER_OVERFLOW	120	Region counter overflow
RTK_EXCPT_COUNTER_UNDERFLOW	121	Region counter underflow
RTK_EXCPT_BAD_SEM_NUMBER	130	Bad semaphore value (RTKInitSem)
RTK_EXCPT_SEM_PRODUCE_OVERFLOW	133	Semaphore produce overflow
RTK_EXCPT_SEM_UNDERFLOW	134	Semaphore produce underflow
RTK_EXCPT_SEM_TIMER_ERROR	137	No more task should be writing for this semaphore
RTK_EXCPT_SEM_TIMER_USED_ERROR	138	Timer identifier already used

RTK Exception	Value	Cause
RTK_EXCPT_SEM_TIMER_STOP_ERROR	139	Not allowed to stop a timer linked to a semaphore
RTK_EXCPT_BAD_MAILBOX_NUMBER	140	Bad mailbox number (RTKInitMailBox)
RTK_EXCPT_NULL_PTR_SEND_PRIO	141	Null message pointer
RTK_EXCPT_BAD_SEND_MAILBOX_NUMBER	142	Bad mailbox number (RTKSend)
RTK_EXCPT_RECEIVE_MAILBOX_NUMBER	143	Bad mailbox number(RTKSendWithPrio)
RTK_EXCPT_OBJECT_OVERFLOW	144	No more free envelope
RTK_EXCPT_BAD_RECEIVE_MAILBOX_NUMBER	145	Bad mailbox number (RTKReceive)
RTK_EXCPT_BAD_TEST_RECEIVE_MAILBOX_NUMBER	146	Bad mailbox number (RTKInitiMailBox)
RTK_EXCPT_NO_MEMORY_AVAILABLE	153	No memory availablefor memory allocation request
RTK_EXCPT_RELEASE_OUT_OF_HEAP	155	Bloc address not in the given heap
RTK_EXCPT_DESTROYED_HEAP	159	Heap destroyed
RTK_EXCPT_BAD_HEAP_RELEASE	161	Bad bloc headeror footer. Header or footer may have been destroyed.
RTK_EXCPT_REL_MEMGLB_FAIL	163	Bad global release. Bloc address not in the given heap
RTK_EXCPT_BLOC_ALLOC_OVERFLOW	166	Heap destroyed
RTK_EXCPT_DESTROYED_HEAP2	172	Heap destroyed
RTK_EXCPT_BAD_EXTRACTION_TIMER	173	Bad timer extraction
RTK_EXCPT_NULL_MESSAGE_POINTER	180	Null message pointer
RTK_EXCPT_LINK_HEAP_NOT_HEAP	181	Bloc address not in the given heap
RTK_EXCPT_LINK_HEAP_NULL	182	Link counter null
RTK_EXCPT_LINK_HEAP_MAX_VALUE	183	Link counter max value
RTK_EXCPT_STACK_OVERFLOW	190	Task stack overflow
RTK_EXCPT_STACK_ILLEGAL_ACCESS	191	Illegal access to protected call stack (in write throw mode)
RTK_EXCPT_IRQ_STACK_OVERFLOW	192	IRQ system stack overflow
RTK_EXCPT_FIQ_STACK_OVERFLOW	193	FIQ system stack overflow
RTK_EXCPT_SVC_STACK_OVERFLOW	194	SVC system stack overflow
RTK_EXCPT_STACKOPENIRQ	195	Open IRQ stack overflow
RTK_EXCPT_BAD_SEM_PRODUCE_NUMBER	201	Bad semaphore value (@ref RTKSemProduce)
RTK_EXCPT_BAD_SEM_CONSUME_NUMBER	202	Bad semaphore value (@ref RTKSemConsume)
RTK_EXCPT_BAD_SEM_SONSUME_DELAY_NUMBER	203	Bad semaphore value (@ref RTK SemConsumeDelay)
RTK_EXCPT_BAD_SEM_CONSUME_TEST_NUMBER	204	Bad semaphore value (@ref RTKSemTestConsume)
RTK_EXCPT_CONSUME_OVERFLOW	205	Semaphore value overflow (@ref RTKSemConsume)
RTK_EXCPT_SEM_CONSUME_DELAY_OVERFLOW	206	Semaphore value overflow (@ref RTKSemConsumeDelay)

RTK Exception	Value	Cause
RTK_EXCPT_NON_ALLOWED_REGION	211	Not allowed to consume a semaphore if schedule is deactivated
RTK_EXCPT_NON_ALLOWED_CONSUME_OPBGT	212	Not allowed to consume a semaphore if schedule is deactivated (in @refRTKSemConsumeDelay function)
RTK_EXCPT_NON_ALLOWED_CONSUME_REGION	213	Not allowed to consume a semaphore if schedule is deactivated
NON_ALLOWED_CONSUME_DELAY_REGION	214	Not allowed to consume a semaphore if schedule is deactivated (in @refRTKSemConsumeDelay function)
RTK_EXCPT_BAD_START_TASK_NUMBER	215	Bad task timer
RTK_EXCPT_NON_ALLOWED_CONSUME_DELAY_OPBGT	216	Bad task number
RTK_EXCPT_BAD_GET_TASK_STATE	217	Bad task number
BAD_SET_TASK_STATE	218	Bad task number
BAD_GET_TASK_NUMBER	219	Bad task number
RTK_EXCPT_BAD_SET_TASK_NUMBER	220	Bad task number

7.3 Module initialization faults

The table below lists stack error codes that can be output during the initialization of the module.

Fault number	Description
1	EEPROM definitely corrupt
2	No EEPROM found
3	Unknown flash device
4	Bad object format parameters
5	Dead flash device
6	Collision between objects and code

8 Software Compatibility Matrix

Firmware	Open AT Framework	Libraries
7.x	Open AT Framework 2.X	Internet Library 4.x or newer Developer Studio 1.2 or newer

9 Support

For direct clients: contact your Sierra Wireless FAE

For distributor clients: contact your distributor FAE

For distributors: contact your Sierra Wireless FAE

10 Document History

Level	Date	History
001	February 22, 2012	Creation

11 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 blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. 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.

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.

Copyright

Trademarks

AirCard® is a registered trademark of Sierra Wireless. Sierra Wireless™, AirPrime™, AirLink™, AirVantage™, Watcher™ and the Sierra Wireless logo are trademarks of Sierra Wireless.

wavecom®, , , ®, inSIM®, WAVECOM®, WISMO®, Wireless Microprocessor®, Wireless CPU®, Open AT® are filed or registered trademarks of Sierra Wireless S.A. in France and/or in other countries.

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

Macintosh and Mac OS 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 the respective owners.