



Tutorial for Open AT[®] IDE 1.02

Revision: 005
Date: November 2006



wavecom[®]
Make it wireless

Tutorial for Open AT[®] IDE 1.02

Revision: 005
Date: November 20, 2006
Reference: WM_DEV_OAT_UGD_016

Overview

This tutorial describes the process to develop an Open AT[®] application using the Open AT[®] Project Wizard & Visual C++ Development environments, and how to download & start the Open AT[®] application on a Wavecom product.

Trademarks

[®], WAVECOM[®], WISMO[®], Open AT[®], Wireless CPU[®], Wireless Microprocessor[®], and certain other trademarks and logos appearing on this document, are filed or registered trademarks of Wavecom S.A. in France or in other countries. All other company and/or product names mentioned may be filed or registered trademarks of their respective owners.

Copyright

This manual is copyrighted by WAVECOM with all rights reserved. No part of this manual may be reproduced in any form without the prior written permission of WAVECOM.

No patent liability is assumed with respect to the use of the information contained herein.

Document History

Index	Date	Versions	
001	May 12, 2006	Creation	
002	August 2006	Update	
003	October 12, 2006	Update for Open AT [®] IDE 1.01	
004	October 30, 2006	Update	
005	November 20, 2006	Update for IDE 1.02	

Table of Contents

1	INTRODUCTION	8
1.1	References.....	8
1.2	Glossary	8
1.3	Abbreviations	9
2	CREATE APPLICATIONS WITH THE OPEN AT[®] PROJECT WIZARD	10
2.1	Creating an application with the Open AT [®] Project Wizard	10
2.1.1	Step 1: Launching the Open AT [®] Project Wizard	10
2.1.2	Step 2: Project location	11
2.1.3	Step 3: Project API.....	12
2.1.4	Step 4: Project Type.....	12
2.1.5	Step 5: Project Base.....	12
2.1.6	Step 6: Development environment.....	13
2.1.7	Step 7: Creating project	14
2.2	Directory Architecture.....	15
3	GENERATE AND RUN OPEN AT[®] APPLICATIONS IN REMOTE MODE	16
3.1	Generate the Open AT [®] remote application	16
3.1.1	Within Eclipse.....	16
3.1.2	Within Visual C++ 6.0.....	17
3.1.3	Within Visual C++ .NET	17
3.2	Running the Remote Application	18
4	GENERATE AND RUN OPEN AT[®] APPLICATIONS IN TARGET MODE	22
4.1	Build target mode application using a Development Environment	22
4.1.1	Within Eclipse.....	22
4.1.2	Within Visual C++ 6.0.....	23
4.1.3	Within Visual C++ .NET	23
4.2	Build target mode applications from the Open AT [®] IDE command line.....	23
4.3	Generated binaries.....	24
4.4	Download.....	25
4.5	Run the Embedded Application.....	30
4.5.1	With HyperTerminal.....	30
4.5.2	With the Terminal Emulator	30

List of Figures

Figure 1: Open AT® Project Wizard shortcut in the Start menu	10
Figure 2: Open AT® Project Wizard	11
Figure 3: Project path	11
Figure 4: Project API	12
Figure 5: Project type.....	12
Figure 6: Project base	13
Figure 7: Project name is automatically updated.....	13
Figure 8: Development environment selection	13
Figure 9: Create folder message box.....	14
Figure 10: Project files creation process.....	14
Figure 11: Directory Structure	15
Figure 12: Make Targets view	16
Figure 13: RTE project & configuration selection (Visual C++ 6.0)	17
Figure 14: Configuration selection (Visual C++ .NET)	17
Figure 15: Auto Detect toolbar button	18
Figure 16: Init Target toolbar button	18
Figure 17: Open TRACE toolbar button.....	18
Figure 18: Auto Detect toolbar button	18
Figure 19: Terminal Emulator	19
Figure 20: RTE Monitor window.....	20
Figure 21: Traces in Target Monitoring Tool	21
Figure 22: Strings displayed in Terminal Emulator.....	21
Figure 23: Make Targets view	22
Figure 24: Target project & configuration selection (Visual C++ 6.0)	23
Figure 25: Configuration selection (Visual C++ .NET)	23
Figure 26: Connection Type.....	25
Figure 27: Phone Call Destination.....	25
Figure 28: COM1 Serial Port Features.....	26
Figure 29: Modem Connection Check	27
Figure 30: File Download with HyperTerminal	28
Figure 31: File Selection.....	28
Figure 32: Application Download process	29

Tutorial for Open AT[®] IDE 1.02

Figure 33: Application is running on Hyper Terminal	30
Figure 34: Auto Detect toolbar button	31
Figure 35: Init Target toolbar button	31
Figure 36: Open TRACE toolbar button.....	31
Figure 37: Auto Detect toolbar button	31
Figure 38: Set and Request toolbar button	31
Figure 39: Set and Request window.....	32
Figure 40: Terminal emulator interface	33
Figure 41: Embedded application traces	33

1 Introduction

1.1 References

[1]	Tools Manual for IDE 1.02 (ref WM_DEV_OAT_UGD_018)
[2]	ADL User Guide
[3]	AT Command Interface Guide

1.2 Glossary

AT commands	Set of standard modem commands.
AT function	Software that processes AT commands and AT subscriptions.
Embedded application	User application sources to be compiled and run on a Wavecom product.
Embedded OS	Software that includes the Embedded application and the Wavecom library.
Target	Open AT® compatible product supporting an Embedded Application.
Target Monitoring Tool	Set of utilities used to monitor a Wavecom product.
Receive command pre-parsing	Process that intercepts AT responses.
Remote Application	Set of libraries enabling the User to run his application on a PC.
Send command pre-parsing	Process that intercepts AT commands.
Wavecom library	Library delivered by Wavecom to interface Embedded application sources with Wavecom OS functions.
Wavecom OS	Set of GSM and open functions supplied to the User.

1.3 Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
IDE	Integrated Development Environment
IR	Infrared
KB	Kilobyte
OS	Operating System
PDU	Protocol Data Unit
RAM	Random-Access Memory
ROM	Read-Only Memory
RTK	Real-Time Kernel
SMA	SMall Adapter
SMS	Short Message Services
SDK	Software Development Kit

2 Create Applications with the Open AT® Project Wizard

2.1 Creating an application with the Open AT® Project Wizard

This Wizard provides a user-friendly way to create an Open AT® project. The different steps to create an Open AT® project are listed in the following paragraphs.

2.1.1 Step 1: Launching the Open AT® Project Wizard

The Open AT® Project Wizard may be launched from the "Start->Wavecom->Open AT IDE 1.XX.XX ->Open AT Project Wizard" shortcut.

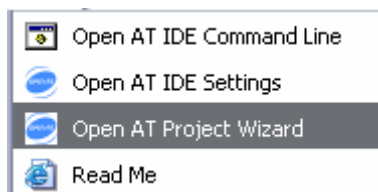


Figure 1: Open AT® Project Wizard shortcut in the Start menu

Tutorial for Open AT® IDE 1.02

The Open AT® Project Wizard window is then displayed.

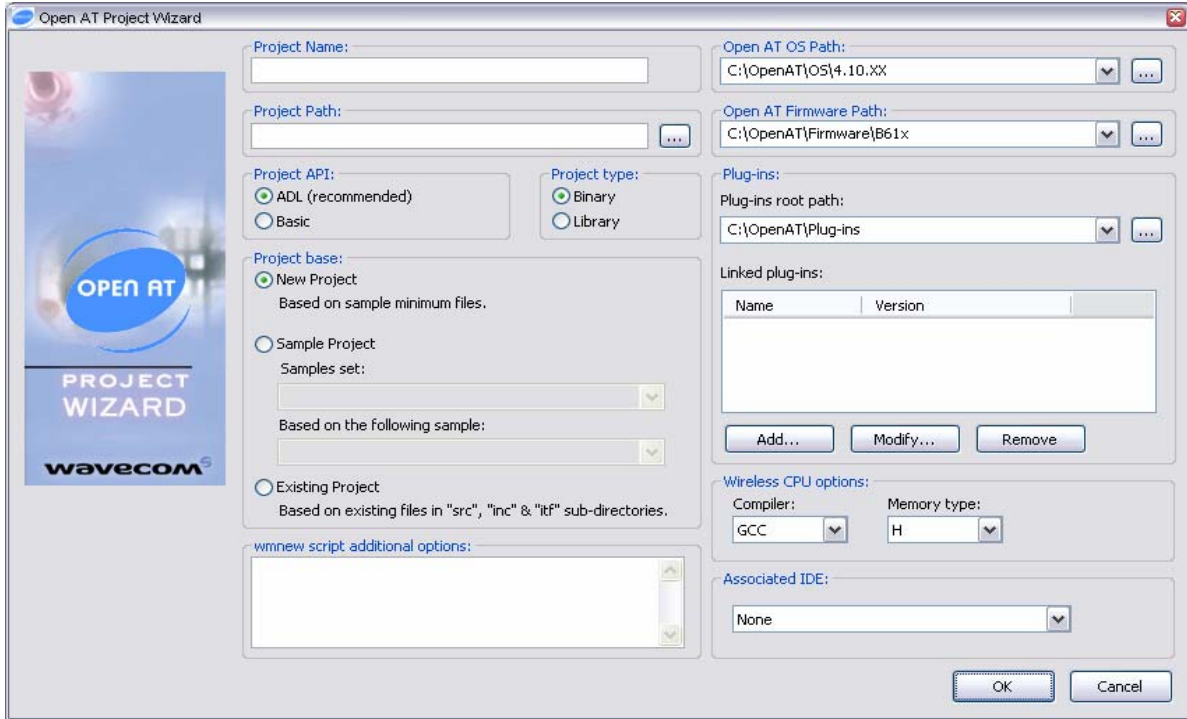



Figure 2: Open AT® Project Wizard

2.1.2 Step 2: Project location

Then, the project location has to be set. The corresponding text field may be updated to set this parameter. The  button may be used to access to an existing directory. New directory may be added manually in the text field (new directories will be automatically created by the setup).

Please note that the selected directory will be the real project root directory, so it should not contains any file before creating the new project.

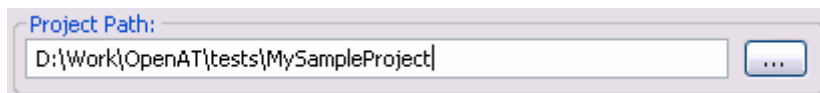


Figure 3: Project path

2.1.3 Step 3: Project API

We can now choose the project API. The ADL API is strongly recommended, since the Basic API is reserved for advanced users, and was kept for ascendant compatibility with older versions.

For this sample, the ADL interface will be kept.



Figure 4: Project API

2.1.4 Step 4: Project Type

Here we define if the generated project will be an Open AT® library or binary. Open AT® libraries are just C standard libraries, using the Open AT interfaces. Open AT® binaries are the Open AT® applications themselves.

For this sample, the Binary type will be kept.

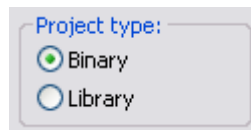


Figure 5: Project type

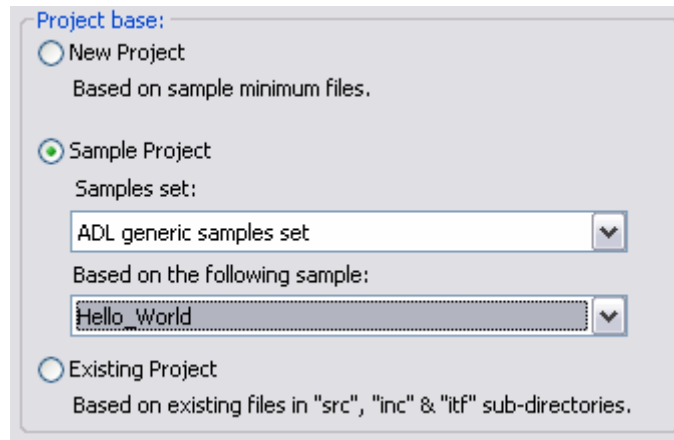
2.1.5 Step 5: Project Base

Open AT® projects are created from New Project or Samples Project bases, and may be updated using the Existing Project option.

- ❑ **New Project:** The Wizard will insert template source files into the project with the basic functions (see the minimum embedded application code in the Development Guides).
- ❑ **Sample Project:** The Wizard will create a new project, with a copy of a sample provided on the Open AT® CD-ROM. The user can choose the sample he wants to use for this project.
- ❑ **Existing Project:** no file will be added to the project. This option is useful when the user wants to create a new project using existing application code files. Existing files in. /src and. /inc will be added to the project.

Tutorial for Open AT® IDE 1.02

For this sample, the Hello_World Sample Project base will be used



Project base:

New Project
Based on sample minimum files.

Sample Project
Samples set:
ADL generic samples set

Based on the following sample:
Hello_World

Existing Project
Based on existing files in "src", "inc" & "itf" sub-directories.

Figure 6: Project base

Please note that when using the Sample Project base, the project name is automatically updated with the currently selected sample name.

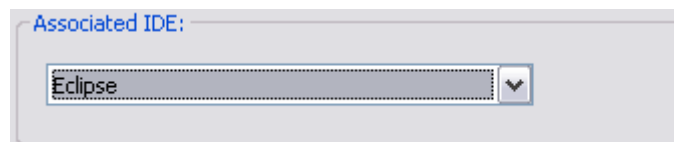


Project Name:
Hello_World

Figure 7: Project name is automatically updated

2.1.6 Step 6: Development environment

In order to launch a supported development environment, this one has to be selected in the drop-down box (only installed environments are listed), and the Launch checkbox has to be clicked.



Associated IDE:
Eclipse

Figure 8: Development environment selection

Tutorial for Open AT® IDE 1.02

2.1.7 Step 7: Creating project

The project files creation process will start as soon as the OK button is clicked.

If the selected path does not exist, the Wizard will prompt the user to create the directory.

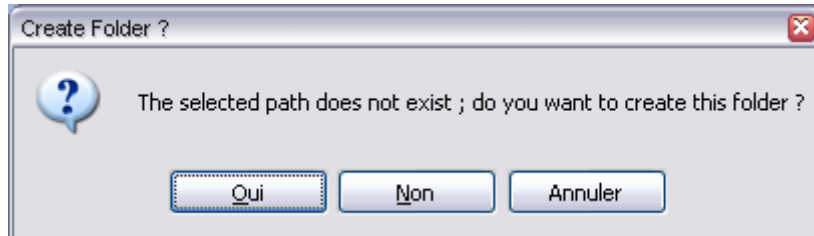


Figure 9: Create folder message box

Then the project files creation process will start. Process evolution is displayed in a bash shell text window. At the end of the process, the Open AT® Project Wizard is closed, and the selected Development Environment is launched.

```
C:\WINDOWS\system32\cmd.exe
Loading Open AT IDE...
Done.
Cleaning existing workspaces...
Copying Hello World sample files...
Project name : Hello World
Browsing used root libraries list...
Browsing used sample libraries list...
Browsing provided full paths list...
Building Hello World project settings file...
Set project options...
Updating flags list...
Building no pre-processing include paths table...
Building include paths table...
Updating include paths list...
-
```

Figure 10: Project files creation process

Note: If project creation problems are encountered, please refer to section 10 Annex: environment & compilation issues in the Tools Manual for IDE 1.02 [1] to solve them.

2.2 Directory Architecture

The typical directory tree structure of the application generated by the Open AT® Project Wizard is shown below:

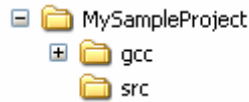


Figure 11: Directory Structure

The "**Root**" directory is the directory entered in the "**Location**" text field in the Open AT® Project Wizard step 2.

It is made up of the following sub-directories:

- ❑ **rte:** will contain the Remote Application binaries, which will be generated by a "**Build**" command,
- ❑ **gcc/out:** will contain the Embedded Application Software binary (generated with the GCC compiler), ready to be downloaded into the Target,
- ❑ **src:** contains the User Open AT® sources,

For this example, the "**src**" directory contains one file:

- hello_world.c

- ❑ **inc:** contains the User Open AT® headers,

For this example, the "**inc**" directory contains no files.

3 Generate and run Open AT® applications in Remote mode

3.1 Generate the Open AT® remote application

See the **Development Guides** and the **AT Command Interface** to create & develop the embedded application.

With this example, an "Hello World" string is displayed on serial link and on Target Monitoring Tool.

The remote application binary is generated as described in the **Tools Manual for IDE 1.02** [1].

3.1.1 Within Eclipse

- Display the "Make Targets" view
- Launch the build by clicking the "Build (RTE)" item.

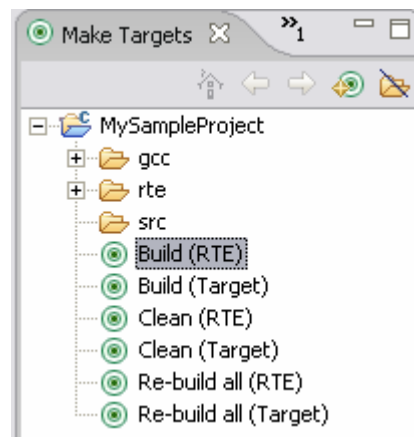


Figure 12: Make Targets view

Tutorial for Open AT® IDE 1.02

3.1.2 Within Visual C++ 6.0

- Select the RTE (Remote Task Environment) project, and the Debug configuration from the build toolbar.

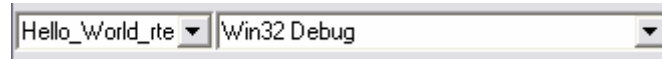


Figure 13: RTE project & configuration selection (Visual C++ 6.0)

- Use the "Build" -> "Build Hello_World_rte.dll" command (or F7 key).

3.1.3 Within Visual C++ .NET

- Select the Debug configuration from the build toolbar.



Figure 14: Configuration selection (Visual C++ .NET)

- In Visual C++ .NET, use the "Build" -> "Build solution" command (or Ctrl+Shift+B key combination).

If any compilation or link error occurs, please refer to section 10 Annex: environment & compilation issues in the Tools Manual for IDE 1.02 [1] to solve them.

3.2 Running the Remote Application

In the Target Monitoring Tool

Start the Target Monitoring Tool (from the Windows Start Menu).

Select **"Commands"** -> **"Auto Detect"** to detect the target speed (or use the toolbar button).



Figure 15: Auto Detect toolbar button

Select **"Commands"** -> **"Init Target"** to initialize the target in debug mode (or use the toolbar button).

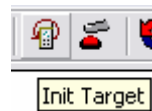


Figure 16: Init Target toolbar button

Select **"Traces"** -> **"Open"** to open the trace window (or use the toolbar button), if it is not already opened.

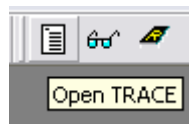


Figure 17: Open TRACE toolbar button

Select **"Commands"** -> **"Get Informations About Target"** to initialize the trace mode (or use the toolbar button).

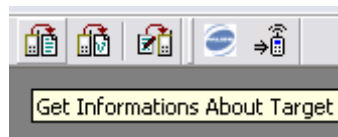


Figure 18: Auto Detect toolbar button

Tutorial for Open AT® IDE 1.02

In the Terminal Emulator

From the Target Monitoring Tool toolbar, start the Terminal Emulator.

Type the **"AT"** command in the Terminal Emulator window.

The Wireless CPU® should answer the **"OK"** string, in blue characters. If not, check your target connection or check if another program is using the serial port. Then, retry the operation.

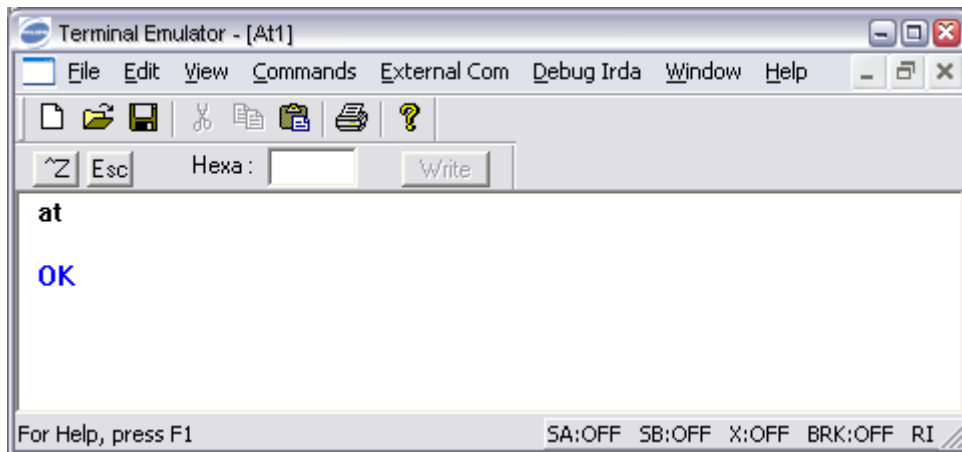


Figure 19: Terminal Emulator

In the IDE

Launch the application with:

- the toolbar button in Eclipse,
- the "Build" -> "Start Debug" -> "Go" command in Visual C++ 6.0,
- the "Debug" -> "Start" command in Visual C++ .NET.


The Remote Application Controller window appears.

Tutorial for Open AT[®] IDE 1.02

In the Remote Application Controller

Select the "CUSTOMER trace levels" you want to show in the Target Monitoring Tool.

For this example, the Hello World application uses Level 1 of CUS4 task. You can check this levels in the Remote Application Controller window.

Select the  button.

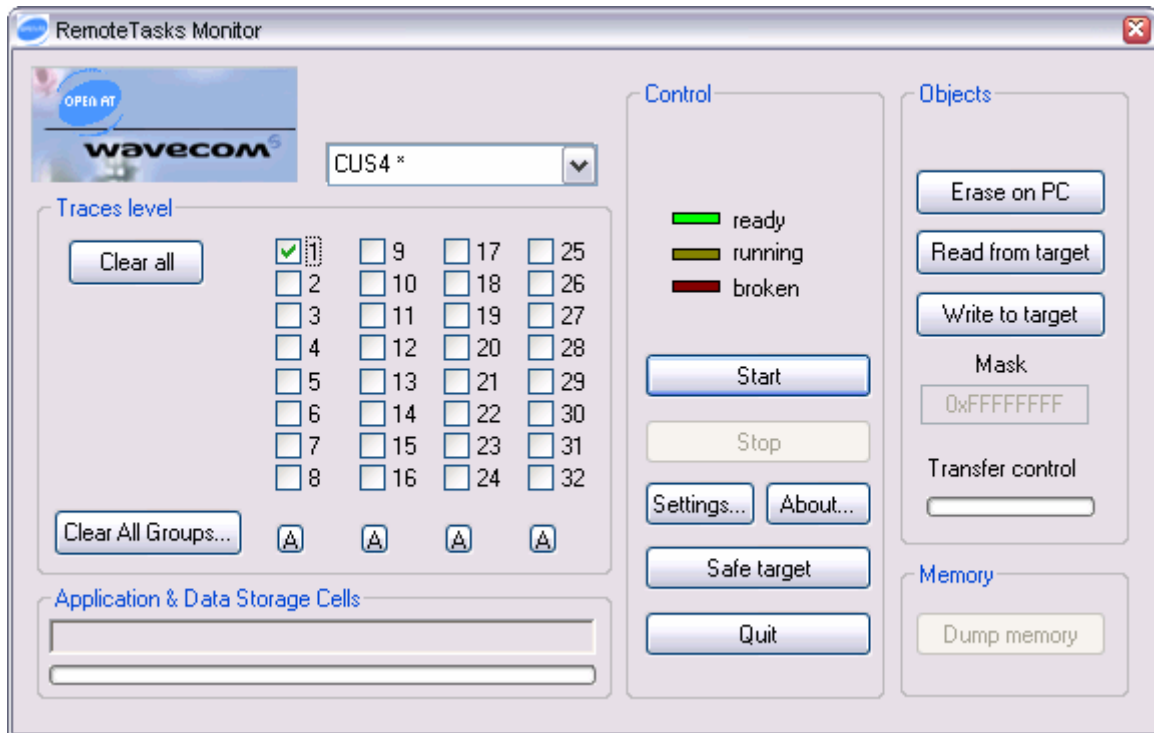


Figure 20: RTE Monitor window

Tutorial for Open AT® IDE 1.02

In the Target Monitoring Tool

Traces should appear in blue in the trace window, indicating that the remote application is running.

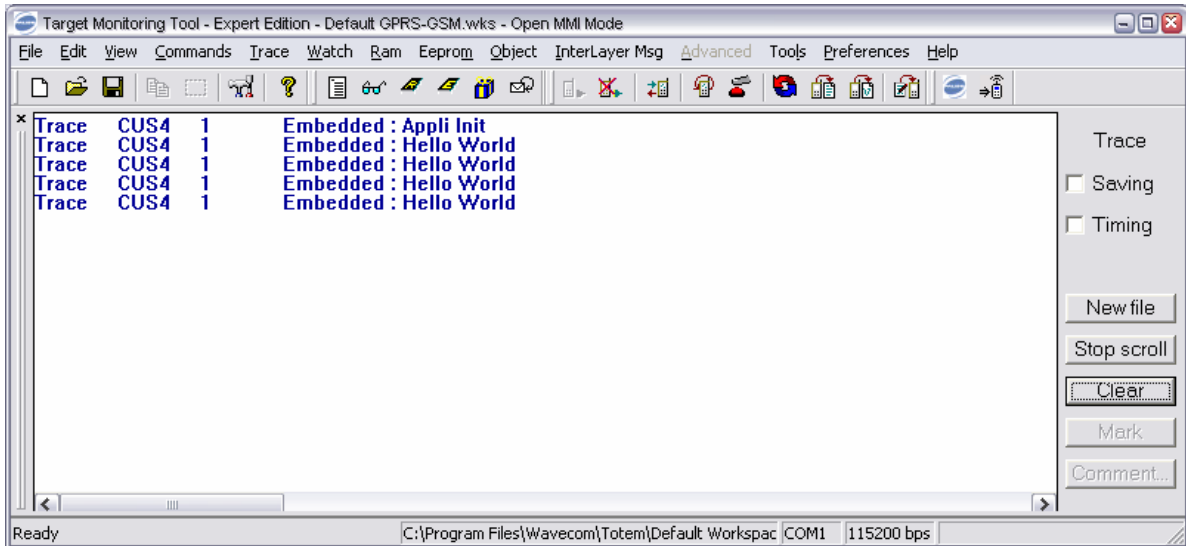


Figure 21: Traces in Target Monitoring Tool

In the Terminal Emulator

The modem is ready to receive AT commands, and displays strings sent by the Open AT® application.

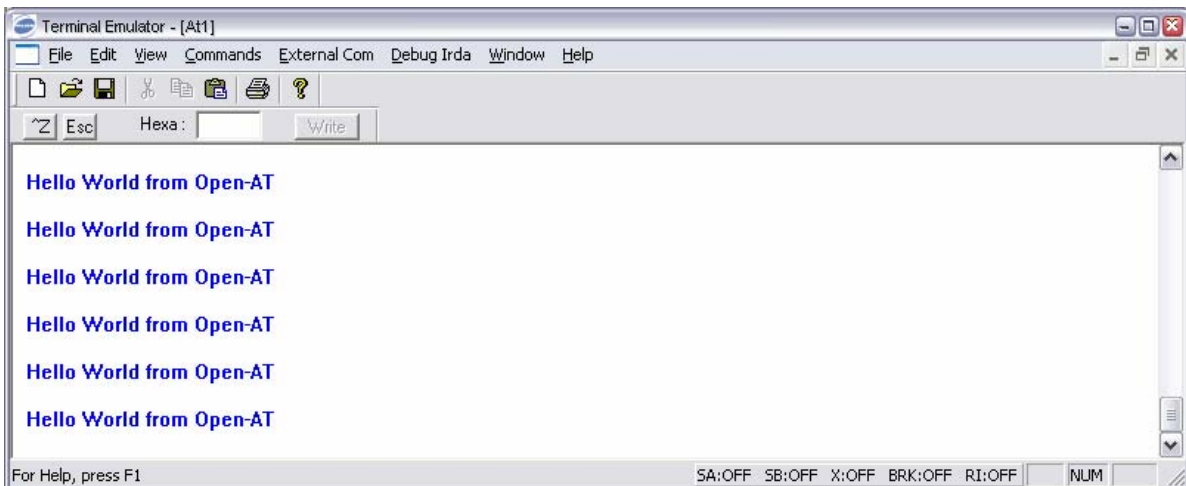


Figure 22: Strings displayed in Terminal Emulator

4 Generate and run Open AT® applications in Target mode

The target application binary is generated as described in the **Tools Manual for IDE 1.02** [1].

It may be generated using one of the supported IDE, or from the Open AT® IDE command line.

4.1 Build target mode application using a Development Environment

4.1.1 Within Eclipse

- Display the "Make Targets" view
- Launch the build by clicking the "Build (Target)" item.

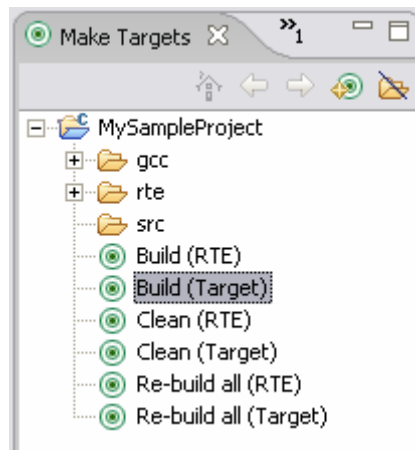


Figure 23: Make Targets view

4.1.2 Within Visual C++ 6.0

- Select the target project, and the Wismo_Target configuration from the build toolbar.

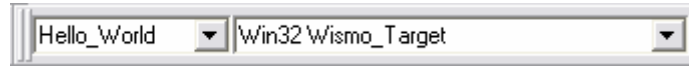


Figure 24: Target project & configuration selection (Visual C++ 6.0)

- Use the "Build" -> "Build Hello_World" command (or F7 key).

4.1.3 Within Visual C++ .NET

- Select the Wismo Target configuration from the build toolbar.

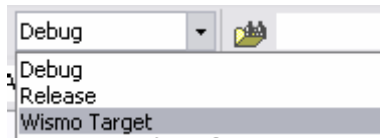


Figure 25: Configuration selection (Visual C++ .NET)

- In Visual C++ .NET, use the "Build" -> "Build solution" command (or Ctrl+Shift+B key combination).

If any compilation or link error occurs, correct them and restart the build operation.

4.2 Build target mode applications from the Open AT® IDE command line

- In a Open AT® IDE command window, call the "wmmake xxx" command in the root directory, where "xxx" is the application makefile name (without the ".mak" extension).

For this example, call the "wmmake Hello_World" command, from the "Hello_World" directory.

- Check in the created "[compiler]/out" directory for the new ".obj" compiled files. If a compilation error occurs, these ones will be displayed in the terminal window.

4.3 Generated binaries

Check in the "[compiler]/out" directory for the new "[compiler]_Hello_World_[mem].wpb.dwl" & "[compiler]_Hello_World_[mem].dwl" files.

Where:

- [compiler] is the current target compiler for the project, set in the Open AT[®] Project Wizard.
- [mem] is the current target memory type for the project, set in the Open AT[®] Project Wizard. With a Firmware allowing a single link address, the generated application binary is independent of the Wireless CPU[®] memory type. In this case the generated binary files names will be [compiler]_Hello_World.wpb.dwl & [compiler]_Hello_World.dwl.

These files are the Embedded Application binary, ready to be downloaded to the target.

The ".wpb.dwl" one is a compressed version of the ".dwl" one. This file will be smaller and so faster to download on the target.

4.4 Download

In order to download the ".wpb.dwl" file to the target, follow these steps:

1. Launch HyperTerminal;
2. Configure it to open the COM port (see following figures);

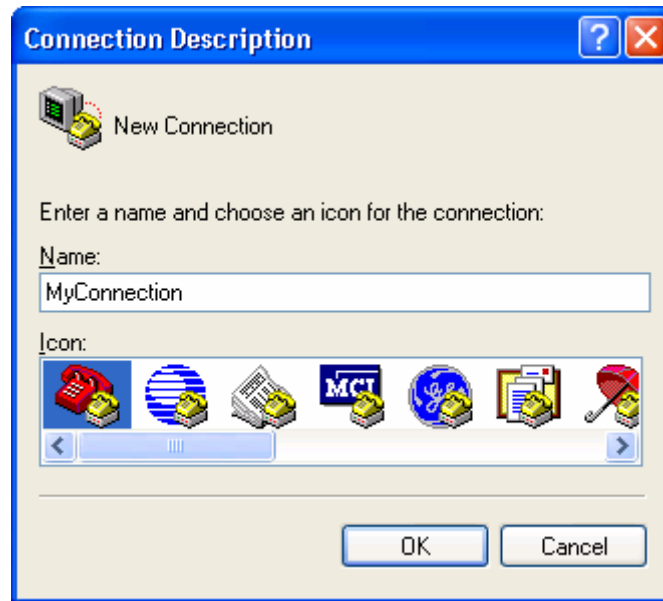


Figure 26: Connection Type



Figure 27: Phone Call Destination

Tutorial for Open AT® IDE 1.02

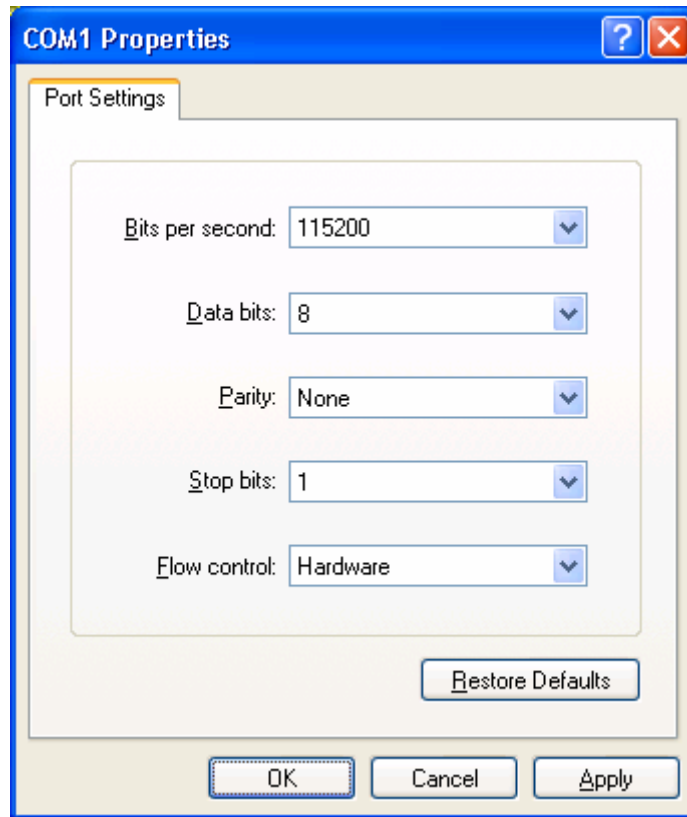


Figure 28: COM1 Serial Port Features

3. Type the "AT" command in the HyperTerminal window. The response must be "OK" (see figure below); The modem will echo characters on the serial link only if it is actually configured (In the figure, the "aaaaaaa" string is typed and the modem echoes it: the modem is connected);

Tutorial for Open AT® IDE 1.02

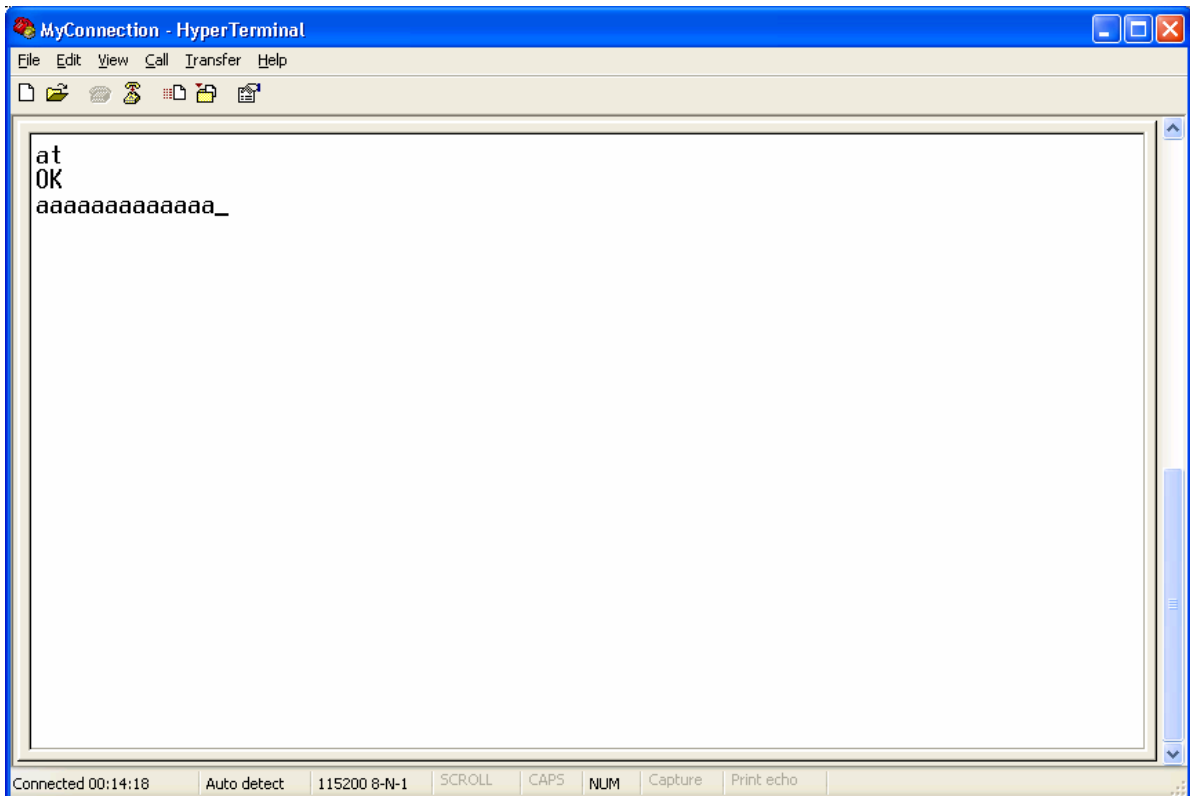


Figure 29: Modem Connection Check

4. Type the "**at+wdwl**" command to start the download mode. A "**+WDWL: 0**" response is given, and a set of characters should appear: the Wireless CPU® has acknowledged the download mode (see Figure below);

Tutorial for Open AT® IDE 1.02

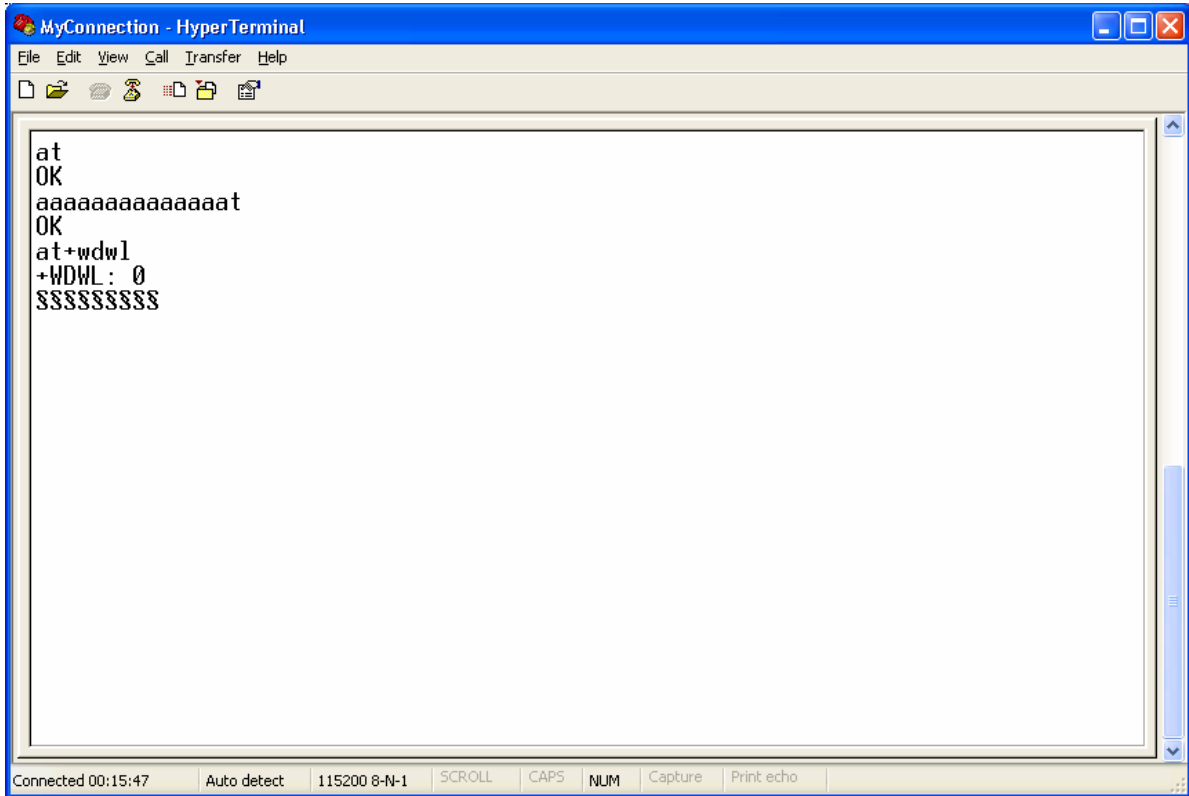


Figure 30: File Download with HyperTerminal

- From the "Transfer" menu, select "Send a file..." (see Figure 8);

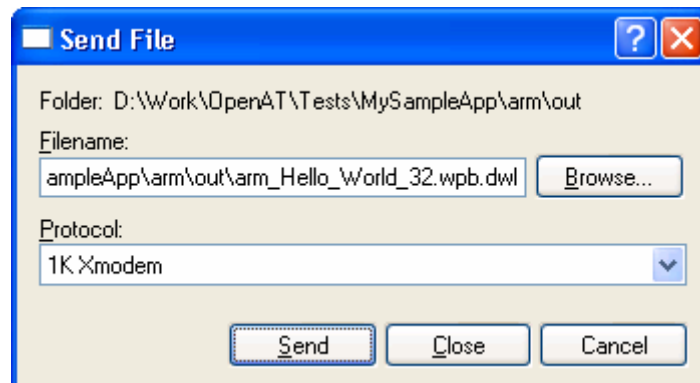


Figure 31: File Selection

- Select the "[compiler]/out/[compiler]_Hello_World_[mem].wpb.dwl" file, and the "1K Xmodem" protocol. Press the "Send" button (see Figure below);

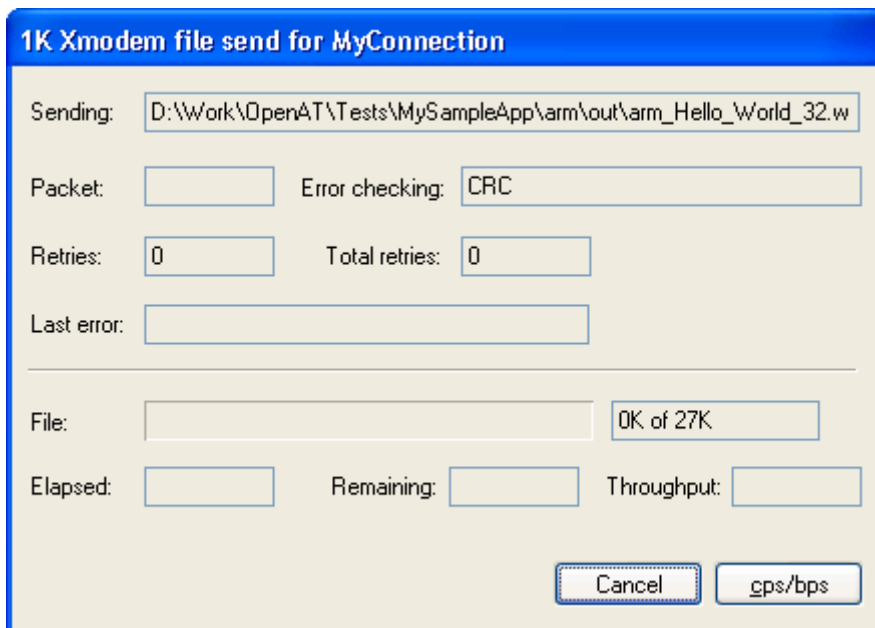


Figure 32: Application Download process

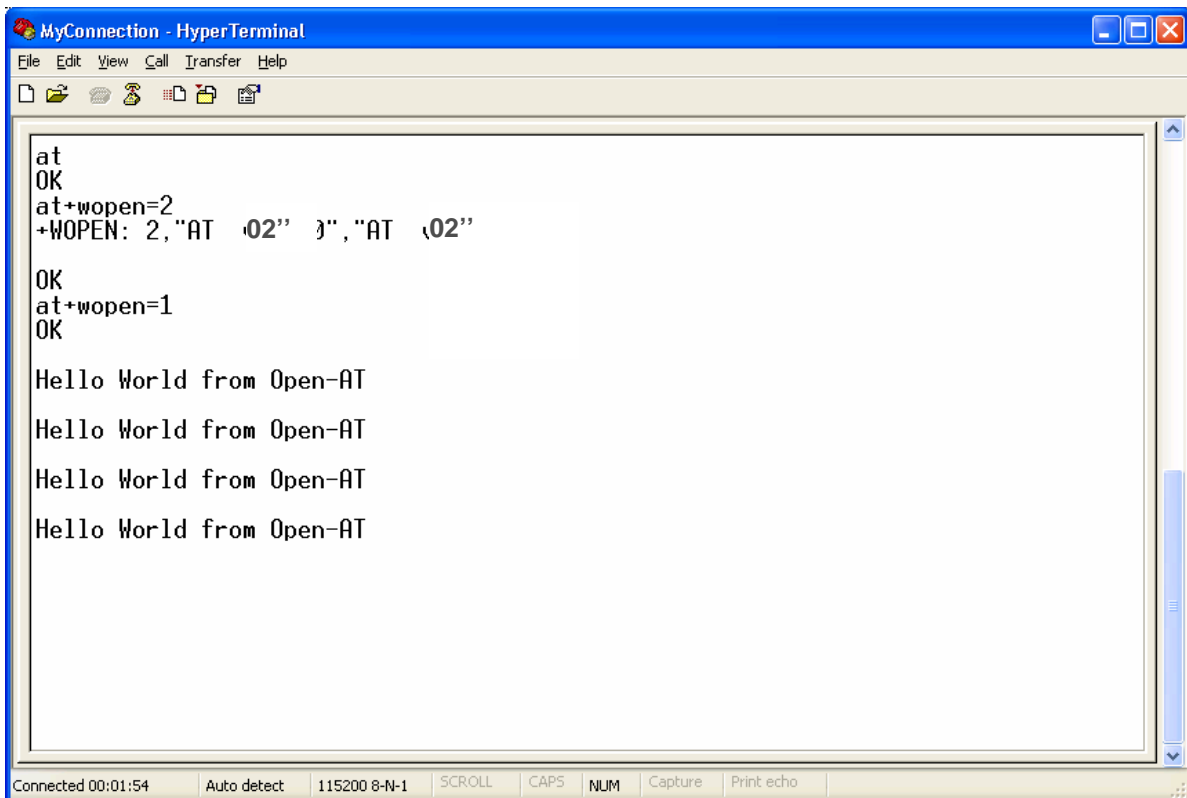
7. As soon as the download process is terminated (the download window disappears), reset the target with the "AT+CFUN=1" command.

4.5 Run the Embedded Application

4.5.1 With HyperTerminal

Type the "at+wopen=1" command, to start the embedded application (see **AT Commands Interface** for more information).

Your application is now running in nominal mode.



```
MyConnection - HyperTerminal
File Edit View Call Transfer Help
at
OK
at+wopen=2
+WOPEN: 2, "AT 02" }", "AT 02"
OK
at+wopen=1
OK
Hello World from Open-AT
Hello World from Open-AT
Hello World from Open-AT
Hello World from Open-AT
Connected 00:01:54 Auto detect 115200 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Figure 33: Application is running on Hyper Terminal

4.5.2 With the Terminal Emulator

Close the HyperTerminal program.

In the Target Monitoring Tool

Start the Target Monitoring Tool (from the Windows Start Menu).

Select "Commands" -> "Auto Detect" to detect the target speed (or use the toolbar button).

Tutorial for Open AT® IDE 1.02



Figure 34: Auto Detect toolbar button

Select **"Commands"** -> **"Init Target"** to initialize the target in debug mode (or use the toolbar button).

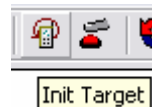


Figure 35: Init Target toolbar button

Select **"Traces"** -> **"Open"** to open the trace window (or use the toolbar button), if it is not already opened.

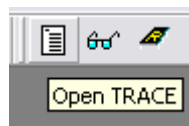


Figure 36: Open TRACE toolbar button

Select **"Commands"** -> **"Get Informations About Target"** to initialize the trace mode (or use the toolbar button).

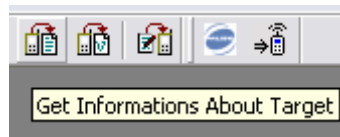


Figure 37: Auto Detect toolbar button

Select **"Traces"** -> **"Set Diagnose Levels."** Select **"CUS4"** in the **"Parameter"** list, check trace levels in the **"Bitmap"** zone, and use the **"Send Level"** button before closing the dialog box.



Figure 38: Set and Request toolbar button

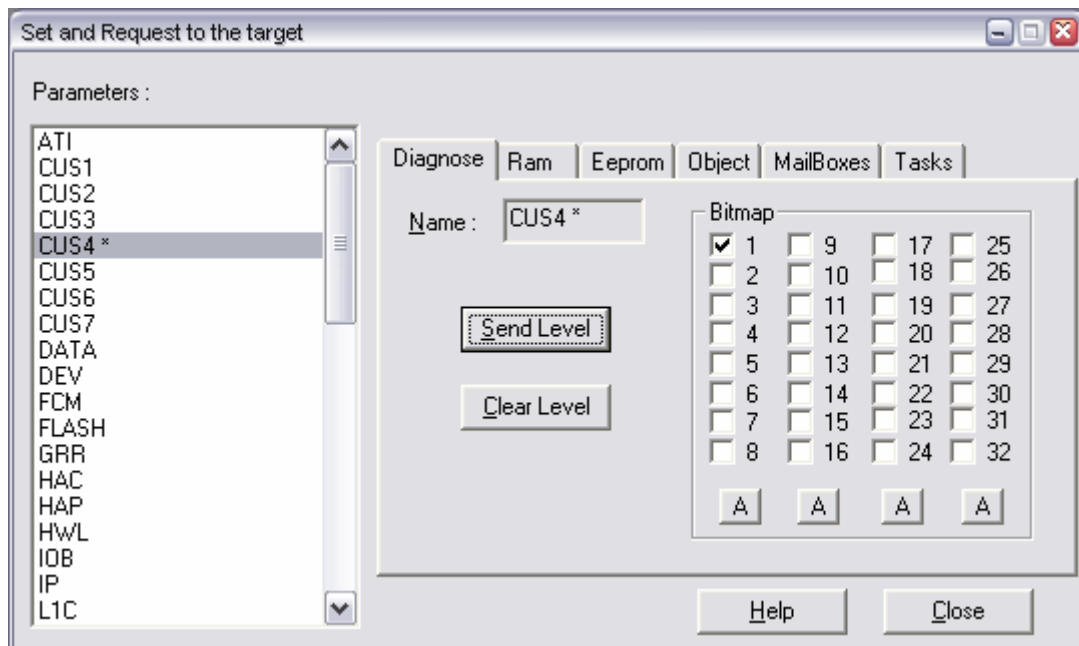


Figure 39: Set and Request window

For this example, the Hello World application uses Level 1, so check this levels in the Set Diagnostic Levels window.

In the Terminal Emulator

From the Target Monitoring Tool toolbar, start the Terminal Emulator.

Type the **"at"** command in the Terminal Emulator AT window.

The Wireless CPU® should answer the **"OK"** string in blue characters. If not, check your target connection, or if another program using the serial port is not running, then retry the operation.

Type **"at+wopen=1"** command in the Terminal Emulator AT window (see **AT Commands Interface** for more information).

Since switching between 0 and 1 states of the +WOPEN command reset the product, re-launch an "init target" process, by the **Ctrl+I** key combination, or from the **"Commands->Init Target"** menu.

The Wireless CPU® is then ready to receive AT commands, and displays the strings sent by the embedded application.

Tutorial for Open AT® IDE 1.02

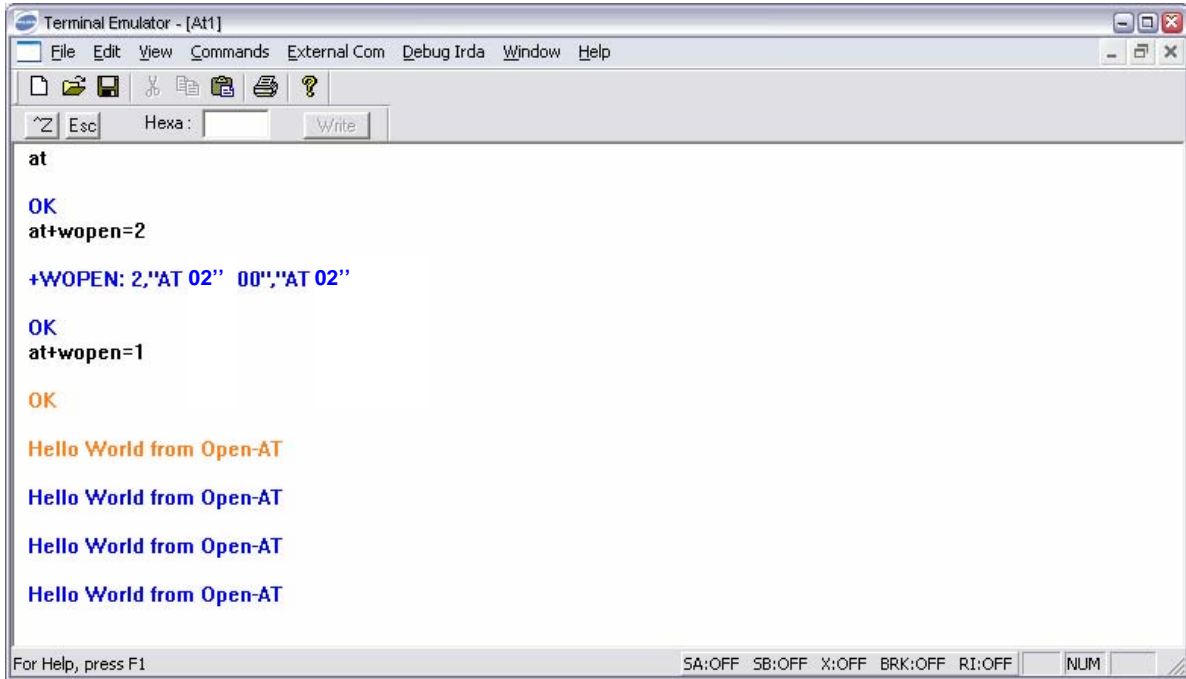


Figure 40: Terminal emulator interface

In the Target Monitoring Tool

Traces should appear in black in the trace window, indicating that the remote application is running.

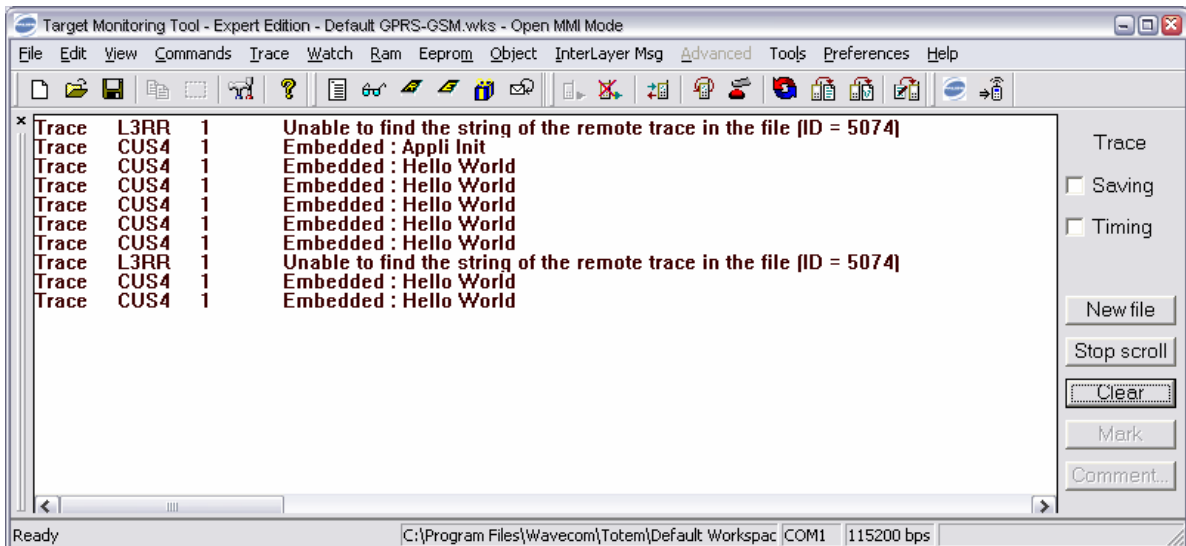
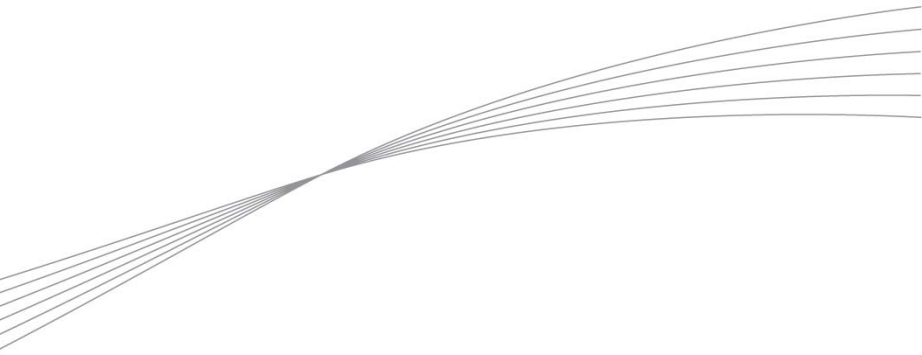


Figure 41: Embedded application traces



wavecom 

Make it wireless

WAVECOM S.A. - 3 esplanade du Foncet - 92442 Issy-les-Moulineaux Cedex - France - Tel: +33(0)1 46 29 08 00 - Fax: +33(0)1 46 29 08 08
Wavecom, Inc. - 4810 Eastgate Mall - Second Floor - San Diego, CA 92121 - USA - Tel: +1 858 362 0101 - Fax: +1 858 558 5485
WAVECOM Asia Pacific Ltd. - Unit 201-207, 2nd Floor, Bio-Informatics Centre - No.2 Science Park West Avenue - Hong Kong Science Park, Shatin
- New Territories, Hong Kong

www.wavecom.com