



SIM800 Series_Embedded AT _Compiling Environment_V1.02



Document name	SIM800 Series_Embedded AT_Compiling Environment
Version	1.02
Date	2015-08-10
Status	Released
Doc Control ID	SIM800 Series_Embedded AT_Compiling Environment_V1.02

General Notes

Simcom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by Simcom. The information provided is based upon requirements specifically provided to Simcom by the customers. Simcom has not undertaken any independent search for additional relevant information, including any information that may be in the customer's possession. Furthermore, system validation of this product designed by Simcom within a larger electronic system remains the responsibility of the customer or the customer's system integrator. All specifications supplied herein are subject to change.

Copyright

This document contains proprietary technical information which is the property of SIMCOM Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2015

Content

Version History	4
1 Tool and Code Architecture.....	5
2 Compiling.....	6
2.1 Compiler Environment.....	6
2.2 Compiling Process Introduction.....	6
2.3 Client Configuration.....	7
2.4 Add New File to Project.....	7
3 Download	9
4 Debug	10
4.1 EAT_UART_DEBUG_MODE_UART mode.....	10
4.2 Catcher Tool Operation	10

Version History

Date	Version	modification	Owener
2013-7-11	1.01	Original	Bin Mao
2015-8-10	1.02	Update scope	Bin Mao

Scope

This document is applicable to SIM800 series Embedded AT module, include SIM800W, SIM840W, SIM800V, SIM800H, SIM800, SIM800M64, SIM808 and SIM800C.

This document describes the development guide of Embedded AT and relative notes.

1 Tool and Code Architecture

Compiler: RVCT

Version: ARM/Thumb C/C++ Compiler, RVCT3.1 [Build 569]

Compiling tool: winmake (under ..\build\)

Download tool: SIMCom_SIM800H_EAT_flash_Tool_V1.01

Folder list:

build	compiling settings & makefile script
core	Head files & libraries
demo	Demo source code
Doc	Documentations
output	Compiled result
SIMCom_SIM800H_EAT_flash_Tool_V1.01	FW update tool
src	Customer's source code
SIM800H32.bat	Compiling script executed with command line

2 Compiling

2.1 Compiler Environment

Once RVCT compiler installed successfully, armcc command will show you correct version info as following.

```
f:\Module\SIM800_52D\MT6252MOD\eat\app>armcc
ARM/Thumb C/C++ Compiler, RVCT3.1 [Build 569]

Usage:          armcc [options] file1 file2 ... fileN
Main options:
```

SIM800H E-AT project will be compiled with script line.

```
f:\Module\SIM800_52D\MT6252MOD\eat\app>build_SIM800W64.bat
```

2.2 Compiling Process Introduction

Build list description.

File	Function	User-configuration
user.mak	Configuration file for compiling path,source code folder,and compiling case,etc.	Yes
Makefile	System entrance	No
option.mak	Configure compiling environment	No
app_build.mak	Compile source code	No

Compiling process is as following.

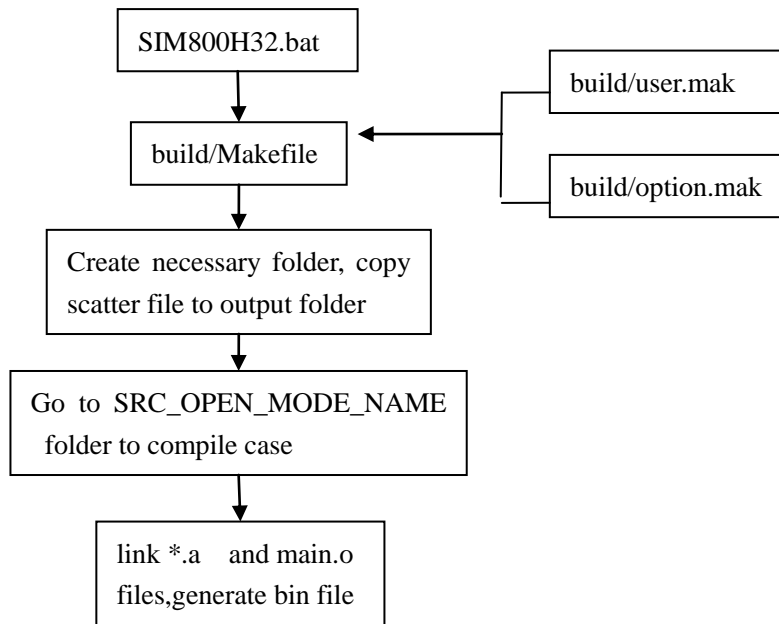


Figure 1 compiling process

Bin file is available in folder defined by OUTPUT_BIN_DIR (user.mak file defined) .
 SIM800H and SIM800 app file will be generated to path core/SIM800H32_EMBEDDEDAT/.
 Downloading core and app file need SIM800H32_EAT.cfg file. If just update app file, just check app option.

2.3 Client Configuration

Client could modify some settings for user.mak.

```

#RVCT 3.1 Root Directory
DIR_ARM = C:\Progra~1\ARM\RVCT
#source file folder
SRC_DIR = demo
#current case to be compiled
SRC_OPEN_MODE_NAME = adc
  
```

2.4 Add New File to Project

Here, we add a new test case, for example sms case, source code includes main.c, app_sms.c and app_sms.h.

Note: case source should include main.c file for app entrance.

Steps for this case:

- 1) Create sms folder;

- 2) Generate main.c, app_sms.c and app_sms.h files.
- 3) Copy Makefile from demo/adc to root sms folder.

Two options here to organize source files.

- a. compile all *.c files (ensure main.c included)

```
#First method#  
S1:= $(wildcard *.c)  
SOURCE :=$( S1)
```

- b. specify source file to compile

```
#Second method#  
SOURCE := main.c \  
app_sms.c
```

*Note, we recommend main.c file for app entrance in EAT project. If there has no main.c but different source file only, should modify following info to link to main.o file.

```
MAIN_SRC := app_demo_uart.c  
MAIN_OBJ := main.o
```

- 4) modify user.mak file.

```
# root directory  
SRC_DIR = src  
# case folder to be compiled  
SRC_OPEN_MODE_NAME = sms
```

- 5) call “sim800H32.bat all” to compile or “sim800H32.bat clean” to clean generated files.

3 Download

Objective file includes two sections. One is core file for simcom module; another is app file for EAT case.

Normally, core is here, just update app file. But if need new core version to build project, should download these two sections together. Make sure app file is compiled based on this new core file.

Note:

- 1. If SIMCom offer new CORE files to customers, CORE must be upgraded. Besides, APP has to be compiled again with new CORE and be downloaded.***
- 2. Download tools will be different according to different requirements.***

4 Debug

Three methods are available to view logs.

- eat_uart_write() interface to print log info
- Configure DEBUG port as EAT_UART_DEBUG_MODE_UART mode, and eat_trace() interface print log info which could be visible with hyperterminal
- Configure DEBUG port as EAT_UART_DEBUG_MODE_TRACE mode, and eat_trace() interface prints log info which only be visible with Catcher tool provided by SIMCom.

4.1 EAT_UART_DEBUG_MODE_UART mode

```
app_func_ext1():
    EatUartConfig_st cfg =
    {
        EAT_UART_BAUD_115200,
        EAT_UART_DATA_BITS_8,
        EAT_UART_STOP_BITS_1,
        EAT_UART_PARITY_NONE
    };
// configure EAT_UART_2 as debug
eat_uart_set_debug(EAT_UART_2);

// configure DEBUG port as UART mode.
eat_uart_set_debug_config(EAT_UART_DEBUG_MODE_UART, &cfg);
```

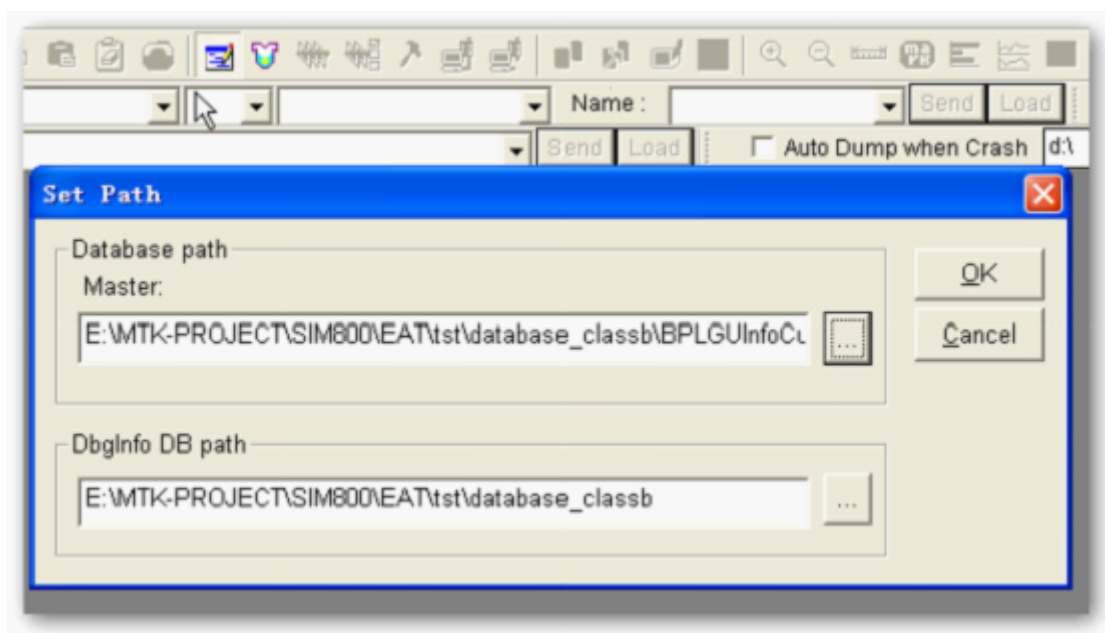
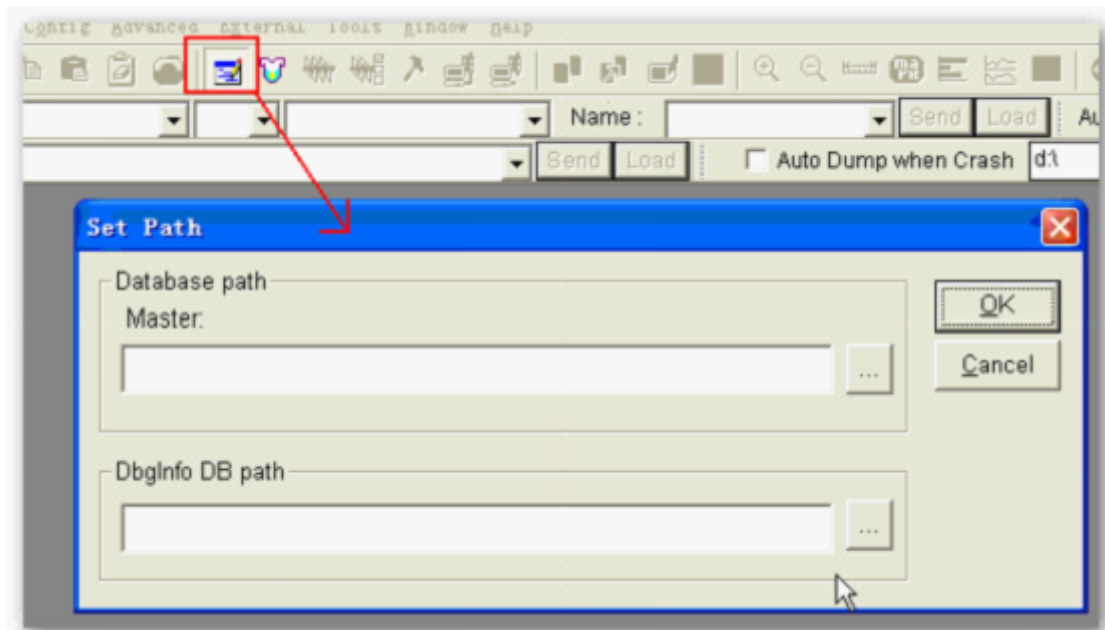
4.2 Catcher Tool Operation

- 1) Configure UART2 output log info.

```
void app_func_ext1(void*data)
{
    eat_uart_set_debug(EAT_UART_2);
    .....
}
```

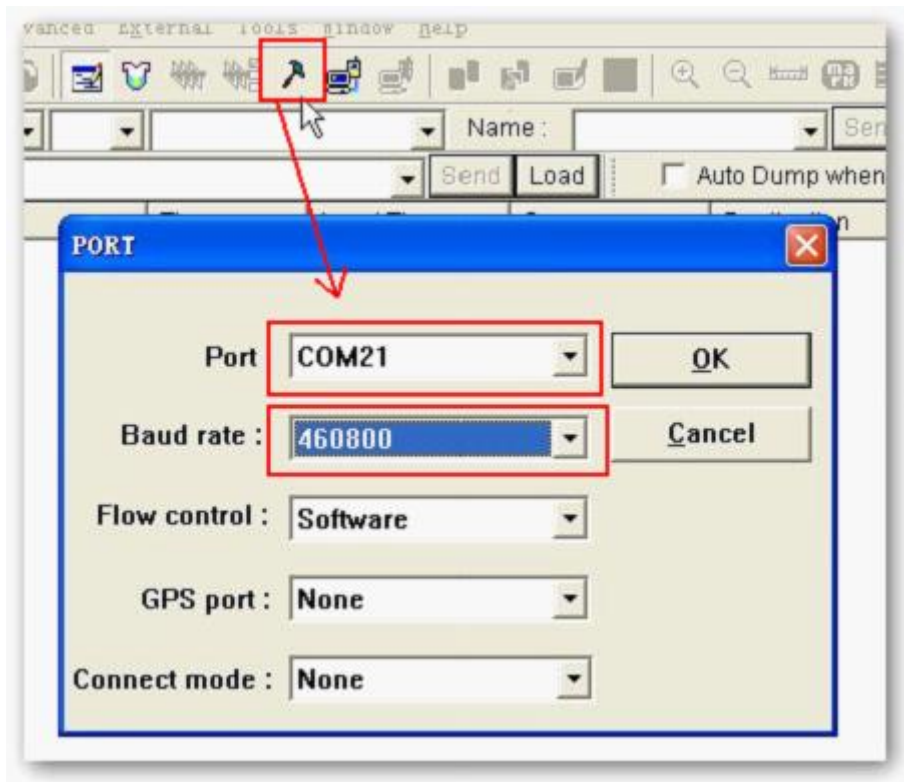
- 2) Connect UART2 to PC, and launch on Catcher tool.
- 3) Pick up data base file.

Database file is provided by SIMCom, such as BPLGUInfoCustomAppSrcP_XXX_XXX_EMBEDDEDAT.

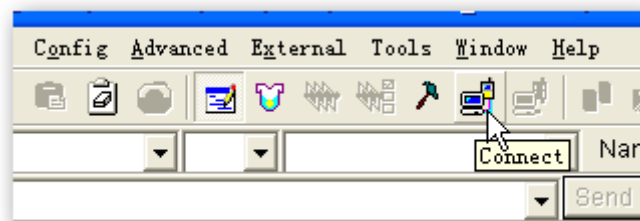


4) Configure port

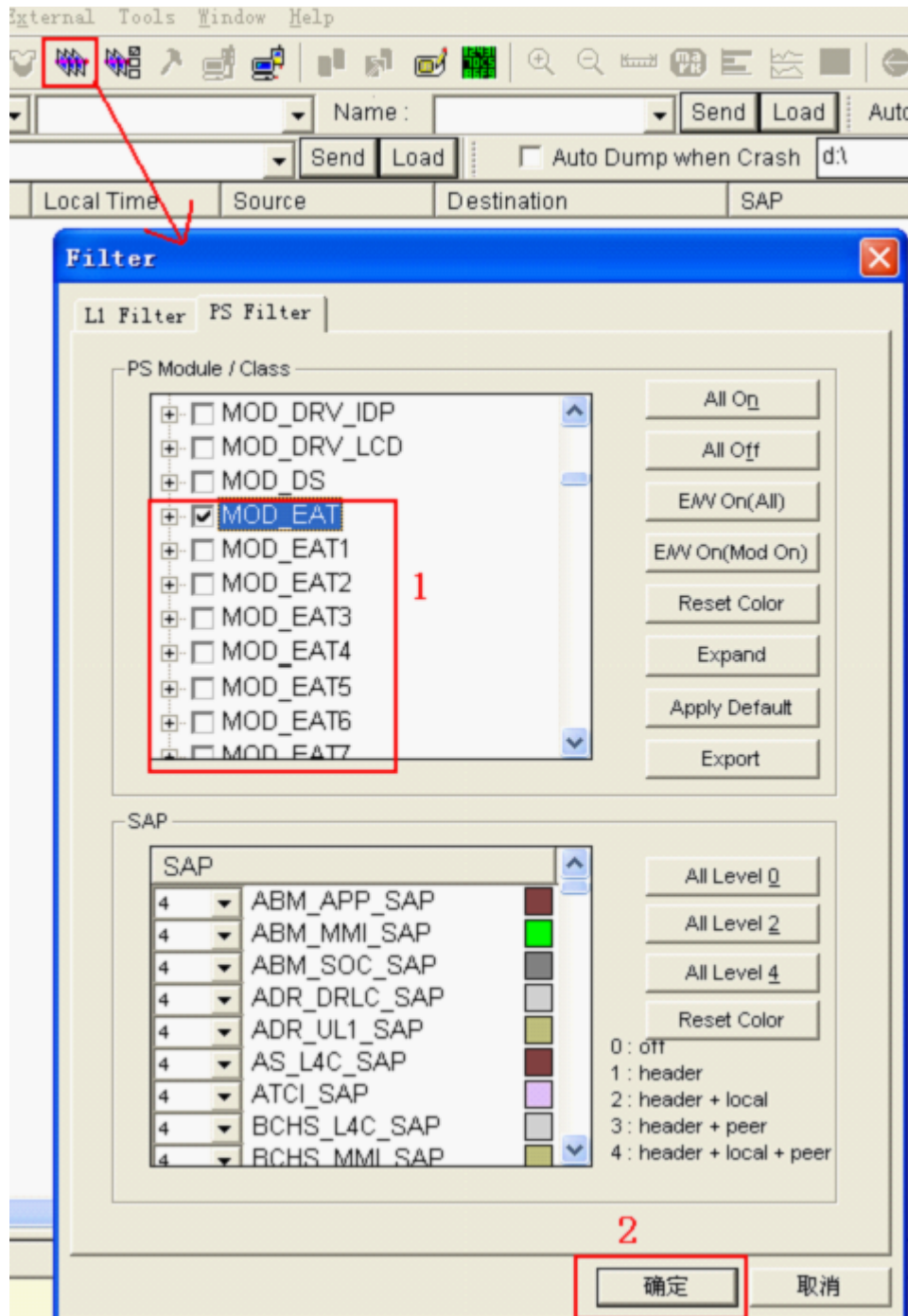
Pick up correct COM port, choose 460800 baud rate.



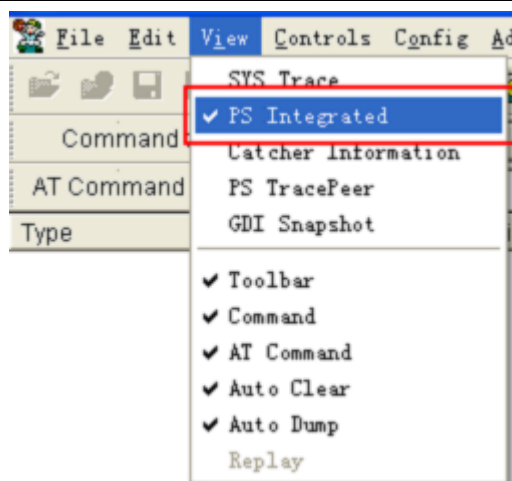
- 5) Hit "Connect" button to continue.



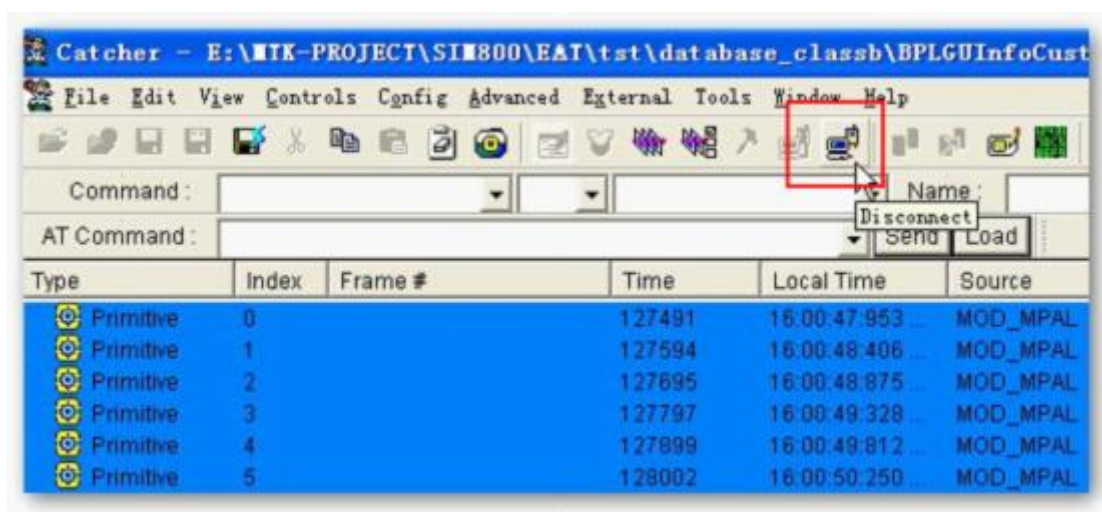
- 6) Configure filter.



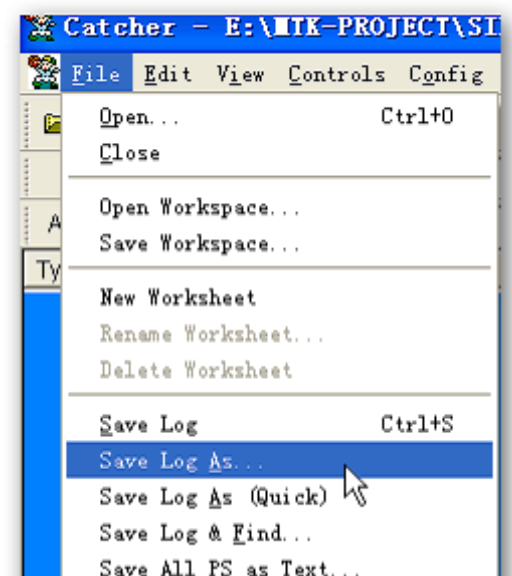
- 7) Ready to catch log info. Hit “View-> PS Integrated”, log trace will be visible.



- 8) Disconnect communication to save log.



- 9) Hit "File->Save log as..." option to save log.



10) Log trace is like following.

Primitive	298	1636	15:33:15.380	...	MOD_DRV_HISR	MOD_EAT	DRIVER_PS_SAP	MSG_ID_UART_READY_TO_READ_IND
Trace	299	1636	15:33:15.380	...	MOD_EAT			eat_core_uart.c:62:when=1 len=0
Trace	300	1636	15:33:15.380	...	MOD_EAT			can_write_len=1
Trace	301	1636	15:33:15.380	...	MOD_EAT			tail=11 head=11
Trace	302	1636	15:33:15.380	...	MOD_EAT			main.c:292:msg 7
Trace	303	1636	15:33:15.380	...	MOD_EAT			get_uart0_buf.
Primitive	304	1636	15:33:15.380	...	MOD_EAT	MOD_ATCI	INVALID_SAP	MSG_ID_UART_READY_TO_READ_IND
Primitive	305	1637	15:33:15.380	...	MOD_ATCI	MOD_EAT	INVALID_SAP	MSG_ID_UART_READY_TO_READ_IND
Primitive	306	1637	15:33:15.380	...	MOD_ATCI	MOD_DS	INVALID_SAP	MSG_ID_READY_UART_WRITE_IND
Trace	307	1637	15:33:15.380	...	MOD_EAT			main.c:292:msg 4

Contact us:

SIMCom Wireless Solutions Co.,Ltd.

Address: Building A, SIM Technology Building, No. 633, Jinzhong Road, Shanghai,
P. R. China 200335

Tel: +86 21 3252 3300

Fax: +86 21 3252 3020

URL: www.sim.com/wm