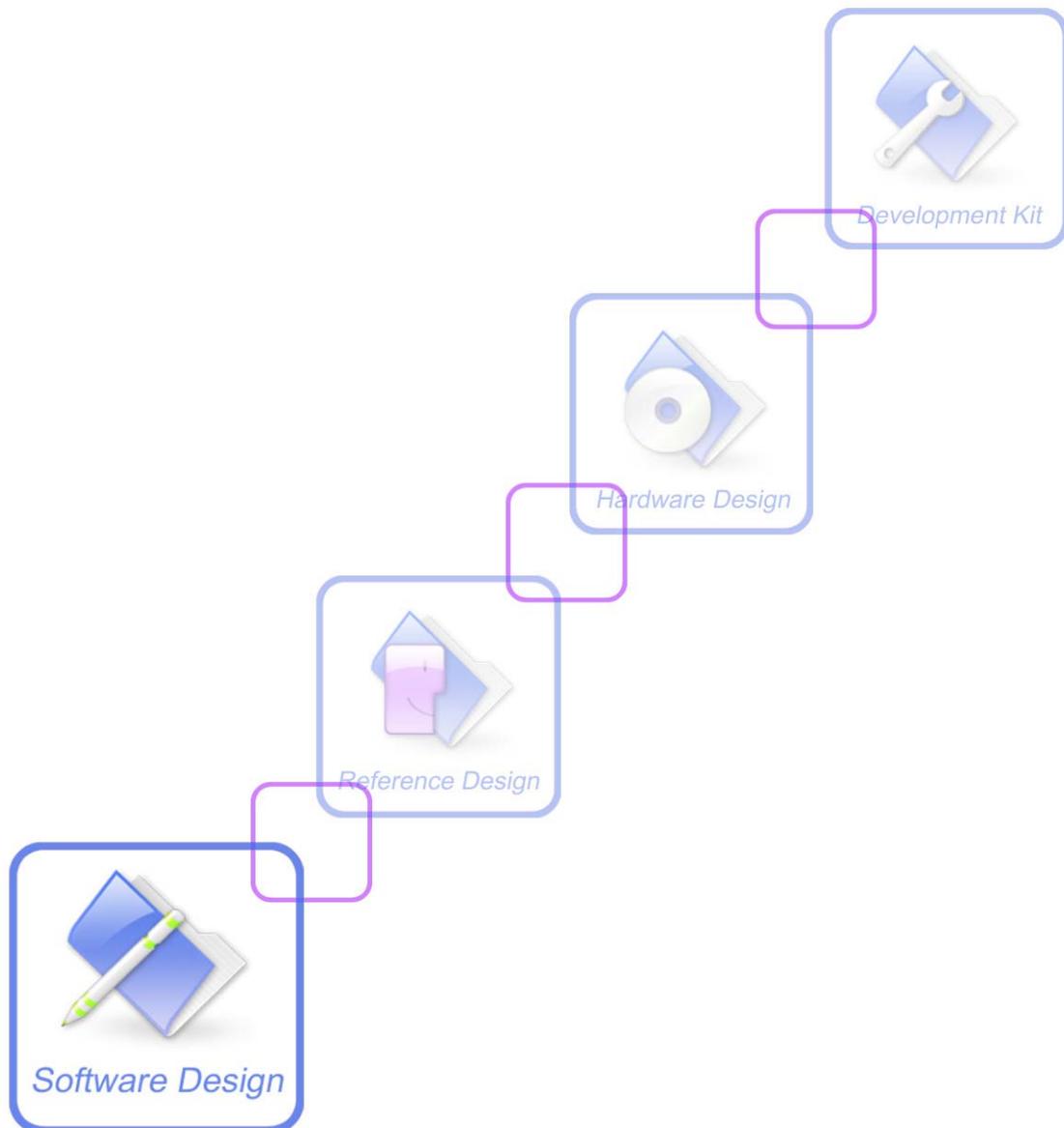




SIM900_Customer
Building
Note_V1.00

Application
Tutorial_Application



Document Title:	SIM900 Custom Application Building Tutorial Application Note
Version:	1.00
Date:	2010-9-16
Status:	Released
Document Control ID:	SIM900_Customer Application Building Tutorial_Application Note_V1.00

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 Limited. 2010

Contents

Version history	4
1 Introduction.....	5
1.1 Purpose.....	5
1.2 References	5
1.3 Glossary	5
1.4 Abbreviations	5
2 Build Directory.....	6
3 Compile Environment	8
4 Create the First Project.....	9
4.1 Copy the project files	9
4.2 Add the source files.....	9
4.3 Add a global macro	9
4.4 Build embedded software.....	10
4.5 Download embedded software and SIMCom core software.....	10
5 Debugging	12

Version history

Date	Version	Description of change	Author
2010-09-01	V1.00	Origin	MXN

1 Introduction

1.1 Purpose

This tutorial describes the process to develop a customer application using embedded AT APIs. This document demonstrates the steps necessary to build an application from scratch, and tends to be a guide for future development.

1.2 References

Sim900 SIMDevIDE Application Note V1.00

1.3 Glossary

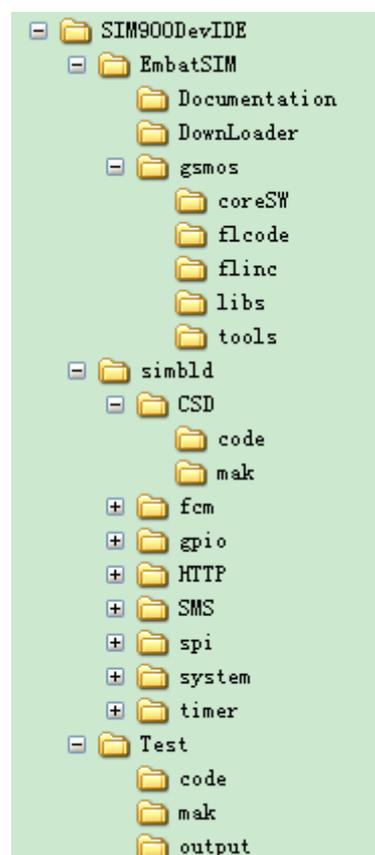
AT commands	Set of standard modem commands.
Embedded application	Customer developed applications running on a SIMCom GSM device that utilizes SIMCom embedded AT APIs.
Embedded software	Software which comprises the Embedded application and the SIMCom library
SIMCom library	Library delivered by SIMCom for Embedded application sources with interfaces of SIMCom Core Software.
SIMCom core system	SIMCom core operating system, in one binary file.
SIMCom downloader	A Windows program which allows developer to download software images to SIMCom module via serial cable.

1.4 Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
OS	Operating System
PDU	protocol data unit
RAM	Random Access Memory
ROM	Read Only Memory
RTK	Real Time Kernel
SDK	Software Development Kit

2 Build Directory

The following figure gives the typical directory structure of an embedded application.



“SIM900DevIDE”-- the root directory of customer project

“EmbatSIM” -- SIMCom folder

“Documentation” -- All related documents of Embedded AT.

“DownLoader” -- Download tools

“gsmos” -- SIMCom core system and library

“coreSW” -- SIMCom core system and spytrace configuration files.

“flcode” -- Codes provided by SIMCom.

“flinc” -- The header files that give the declaration of all interfaces provided by SIMCom.

“libs” -- Which contains SIMCom library.

“tools” -- Which contains different tools provided by SIMCom

“simbld” -- Which contains different examples provided by SIMCom

“example” -- Small example of embedded at.

“code” -- Example code

“mak” -- Make files

“CSD” -- CSD example

“FCM” -- Fcm module example

“GPIO” -- GPIO module example

“HTTP” -- Http module example

“SMS” -- SMS module example

“SPI” -- Spi module example

“SYSTEM” -- System API example

“TIMER” -- Timer module example

“Test” -- The folder created when a new project is created. (The name is depend on the project's name)

“code” -- Customer source code

“mak” -- Make files

“output” -- Binary file of built applications, the “”

3 Compile Environment

RVCT2.2 2_2b616 is mandatory in order to build embedded applications.

After installing RVCT, it is important to configure the compiler path of the RVCT. And the file “opt.mak” in “mak” folder should be modified in the project folder. It is also very important to configure the standard library path.

Assuming RVCT is installed in the directory C:\, the environment variables should have the following values:

```
#####
##$%^&%&^%*****&^*&*&^*&*&^*&*&^*&*&^%&^
#EmbeddedAT900 APPLication Project
# brucefan
# 2010-7-13
#Added by brucefan 2010-07-30 merge gcc & rcvt
##$%^&%&^%*****&^*&*&^*&*&^*&*&^*&*&^%&^s
#####
#####
#Compiler Tools
# You Can Choose :  RVCT  or  GCC
COMP=RVCT
#####
# IF you use RCVT, Please tell me The Compiler ROOT Path
COMPATH=C:\ARM\RVCT\Programs\2_2b616
#####
# IF you use RCVT, Please tell me The Standard Library Header Files ROOT Path
SLIBH=C:\ARM\RVCT\Data\2_2b576\include\windows
```

Note:

- 1 *SIMCom does not supply nor support third party software.*
- 2 *It is forbidden to include space in the name of the path to setup RVCT.*

4 Create the First Project

4.1 Copy the project files

1. Developers should copy the folder “**SIM900DevIDE**” from the CD-ROM to E: (of course, it is OK to copy it to a different path). This tutorial assumes E: is the project directory.

Important Note:

It is recommended to leave folder names unchanged, as they are often referenced by the mak file.

2. Copy the folder “**D:\SIM900DevIDE\simbl\example**” to “**D:\SIM900DevIDE**”, and rename the “**example**” folder, for example “**Test**”, (it depends on the project’s name,), modify “**D:\SIM900DevIDE\Test\mak\opt.mak**” as mentioned above.

4.2 Add the source files

1. Create the source file “**fl_customer.c**” in directory “**code**”, if it does not exist. Note that all the source files must be created in the directory “**code**” otherwise it cannot be compiled successfully.
2. Create the header file “**fl_customer.h**” in the directory “**flinc**”, if it does not exist. Note that all the header files must be created in the directory “**flinc**” otherwise it cannot be compiled successfully.
3. Modify “**D:\SIM900DevIDE\Test\mak\srclist.mak**” and add the source file “**fl_customer.c**” into the “**srclist.mak**”.

```
#####THUMB SRC LIST#####
THUMB_SRC += fl_customer.c
#####
ASM_SRC +=
```

4.3 Add a global macro

Modify the “**D:\SIM900DevIDE\Test\mak\generalfeatures.opt**” to add the macro which user wants to add, as shown in the following:

```
-DSIMCOM
-DSIMCOM_EMBEDDED_AT
```

Note:

“-D” should be added in front of the macro which user wants to add. When user wants to add the second macro, the second macro should be added in the second line. That means the two macros should be separated by “\r” and “\n”. This is very important.

4.4 Build embedded software

The steps to build embedded software are listed as following.

1. Run CMD and go to the “**Test**” directory.



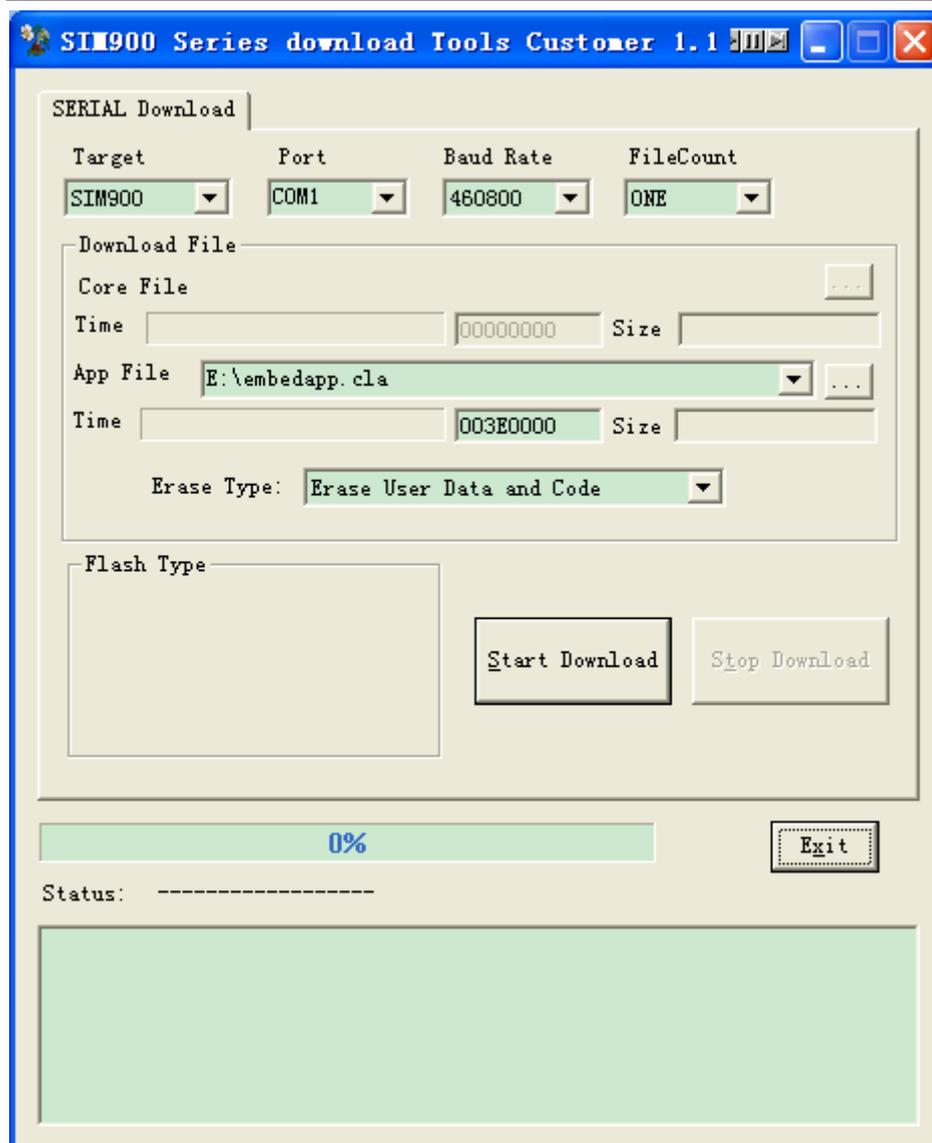
```
D:\SIM900DevIDE\Test>
```

2. Run “**run.bat**” and the firmware will be generated in “D:\SIM900DevIDE\Test\output”

4.5 Download embedded software and SIMCom core software

SIMCom downloader **SIM900 Series download Tools Customer.exe** is used to download embedded software and SIMCom core software onto SIMCom modules, and it is in the directory “**D:\SIM900DevIDE\EmbatSIM\DownLoader**”.

The steps to download software are as following.

**Important:**

1. Before start downloading the software, it is better to power on the module to make it reset by itself to download software, otherwise, it is necessary to reset the module manually to download software.
2. Before downloading SIMCom core mot file, “Erase Type” must be set to “Erase User Data and Code”.
3. “Erase Type” must be set to “Erase Source Code” before downloading the customer cla. Make sure that the address of the app file is 0x003E0000.

5 Debugging

Debugging SIMCom Embedded Application is easy:

1. fl_EnterDebugMode()

The function fl_EnterDebugMode() enters debug mode, directs all the debug information to the debug port. Developers can use terminal applications linked to the debug port such as Hyper Terminal in Windows to view debug information.

2. fl_DebugTrace()

3. The function fl_DebugTrace() prints out debug information whatever developer chooses to be.

It is virtually same as printf(), except fl_DebugTrace() can print out to module's debug port, while printf() can only embed output in system signals.

Example :

```
fl_DebugTrace("debug information: line %d is %s\r\n", linenumber, linecontent);
```

Contact us:

Shanghai SIMCom Wireless Solutions Ltd

Add: SIM Technology Building A,

No. 633, Jinzhong Road, Shanghai, P. R. China 200335

Tel: +86 21 3252 3300

Fax: +86 21 3252 3020

URL: www.sim.com