



M10

Quectel Cellular Engine

Open CPU Tutorial

Tutorial_V1.00



Document Title:	Custom Application Building Tutorial
Revision:	1.00
Date:	2009-7-18
Status:	Draft
Document Control ID:	Tutorial_V1.00

General Notes

Quectel offers this information as a service to its customers, to support application and engineering efforts that use the products designed by Quectel. The information provided is based upon requirements specifically provided to Quectel by the customers. Quectel 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 Quectel 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 Quectel 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 Quectel Wireless Solutions Ltd. 2009

Contents

Contents	2
0. Revision History	3
1. Introduction.....	4
1.1. Purpose.....	4
1.2. References	4
1.3. Glossary.....	4
1.4. Abbreviations	4
2. Build Directory	5
3. Compile Environment	5
4. Create the First Project.....	5
4.1. Copy the project files	5
4.2. Add the source files	6
4.3. Edit the build files	6
4.3.1. Build files in root directory.	6
4.3.2. Build files in customer directory.	6
4.4. Build Embedded Application	7
4.5. Download Embedded Application image.....	8
5. Startup Embedded Application.....	8
6. Debugging	8

0. Revision History

Revision	Date	Author	Description
V1.00	2009-08-07	Willis YANG	Initial.

1. Introduction

1.1. Purpose

This document describes the process to develop an Embedded Application based on Open CPU. This document demonstrates the necessary steps to build an application from scratch, and tends to be a guide for future developments.

1.2. References

SN	Document name
[1]	OPEN_CPU_DGD
[2]	Open CPU Specification

1.3. Glossary

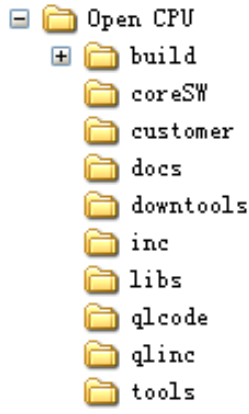
GLOSSARY	Description
AT commands	Set of standard modem commands.
Embedded Application	User created application that utilizes Embedded API functions to interact with Quectel core software, only to run on a Quectel product.
Quectel library	Library delivered by Quectel for Embedded application sources with interfaces of core system.
Core system	The Core system released by Quectel, which includes the core binary file and Quectel library.
Downloader	A Windows program allows developer to download software images to Quectel module via serial cable.

1.4. Abbreviations

Abbreviation	Description
API	Application Programming Interface
CPU	Central Processing Unit
FCM	Flow Control Manager
KB	Kilobyte
OS	Operating System
RAM	Random-Access Memory
ROM	Read-Only Memory
SDK	Software Development Kit

2. Build Directory

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



“**Open CPU**”-- the root directory of customer project

“**CoreSW**” – Quectel core system image

“**customer**” – Embedded Application code

“**build**” – Built binary file, scatter file etc.

“**code**” -- customer source code

“**inc**” -- head file of customer code.

“**docs**” --All related documents of Open CPU.

“**downtools**” – The program used to download binary file to Quectel modules.

“**qlcode**” -- codes provided by Quectel.

“**qlinc**” -- The header files that give the declaration of all interfaces provided by Quectel.

“**libs**” contains Quectel library.

“**tools**” -- Contains different tools provided by Quectel.

3. Compile Environment

ADS(ARM Developer Suite) is mandatory in order to build embedded applications. Catcher tool is used for advance debug, it can be found in tools directory of Open CPU package.

4. Create the First Project

4.1. Copy the project files

Developer should copy the folder “**Open CPU**” 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.

4.2. Add the source files

1. Create the source files in folder “E:\Open CPU\customer\”.
2. Create the header files in folder “E:\Open CPU\inc\”.

4.3. Edit the build files

4.3.1. Build files in root directory.

Root directory contains three build files, “makefiledef”, “makefile”, “config”.

“**makefiledef**”-- defines some global compile rules, normally, it should not be modified.

“**config**”-- defines input source file directory, output files and some predefine macro.

BIN_DIR	the path which built binary file will be created in.
ELF_OUTPUT	the path of the built elf file.
BIN_OUTPUT	the path of the built binary file, which is the image file of Embedded Application.
LIS_OUTPUT	the path of lis file, which contains detailed information of object files in the image file.
SYM_OUTPUT	the path of the symbol table file.
SCATTER_FILE	the path of scatter file, which defines memory layout of image file. It will be used by download tool when downloading image file to module.
MOD_LIST	the paths of the source code directories of Embedded Application.
SYM_INPUT	the path of the input symbol table file. It is necessary for building Embedded Application image with ARM linker.

```

BIN_DIR=build
ELF_OUTPUT=$(strip $(BIN_DIR))\ql_custom.elf
BIN_OUTPUT=$(strip $(BIN_DIR))\ql_custom.bin
LIS_OUTPUT=$(strip $(BIN_DIR))\ql_custom.lis
SYM_OUTPUT=$(strip $(BIN_DIR))\ql_custom.sym
SCATTER_FILE=$(strip $(BIN_DIR))\scat6223.txt

SYM_INPUT=libs\M10_MODEM_08A_PCB01_gprs_MT6223_S00.sym
MOD_LIST=customer

```

“**makefile**”-- define detailed compile and link steps. normally, customer need not to modify it

4.3.2. Build files in customer directory.

makefile-- define source file and include path etc.

INCLUDES	the path where the referenced head files locate.
TARGETLIB	name of the library built from the source files in customer directory.
OBJS	list object files to be built. Every .obj file comes from the corresponding

source file (.c file).

```
INCLUDES = -I.|| -I.\inc\ -I.\qlinc\
TARGETLIB= customer.lib

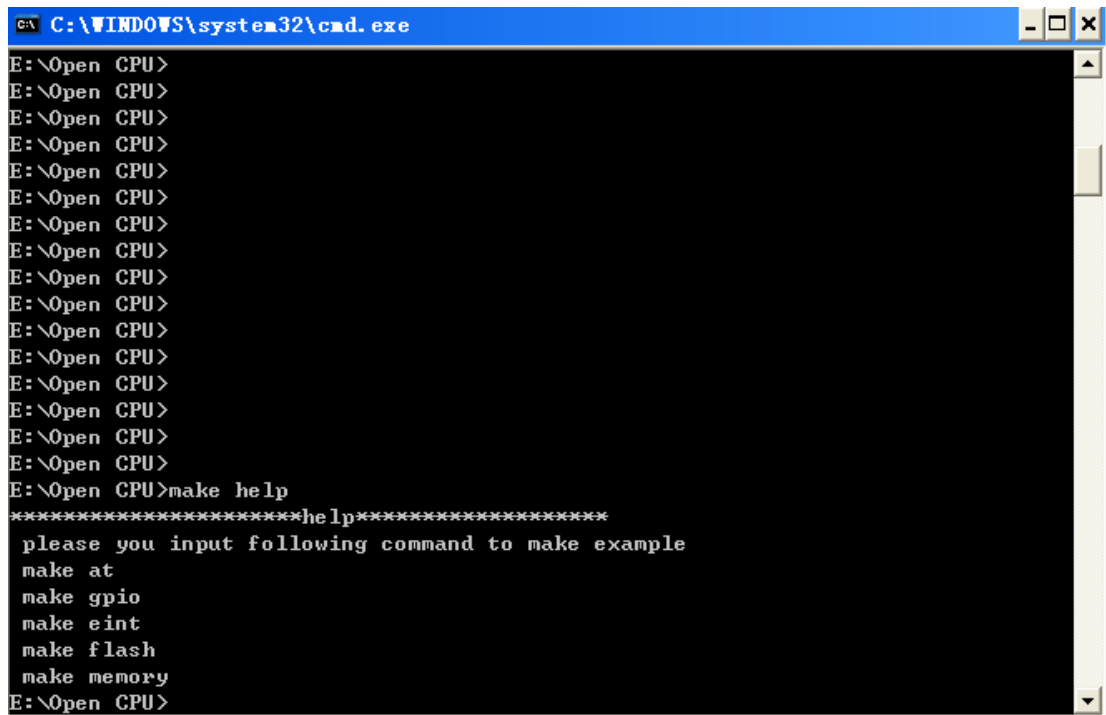
OBJS=example_at.o
```

Note: Customer can set up several directories for customer code. After the directories are created, please add the directory names to **MOD_LIST** and create corresponding **makefile** in every source code directory.

4.4. Build Embedded Application

The steps to build Embedded Application are listed as following.

1. Run CMD and go to the “**Open CPU**” directory.
2. Run “make clean” to clean up all old output files.
3. Run “make help” to show how to make examples Open CPU provides.



```

C:\WINDOWS\system32\cmd.exe

E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>
E:\Open CPU>make help
*****help*****
please you input following command to make example
make at
make gpio
make eint
make flash
make memory
E:\Open CPU>
```

4. Run “make at” to compile and build the example code about sending and receiving AT commands into Embedded Application.


```

C:\WINDOWS\system32\cmd.exe
E:\Open CPU>
E:\Open CPU>make at
*****make at example*****
tools\make.exe[1]: Entering directory `E:/Open CPU'
tools\make.exe[2]: Entering directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Nothing to be done for `all'.
tools\make.exe[2]: Leaving directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Entering directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Nothing to be done for `all'.
tools\make.exe[2]: Leaving directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Entering directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Nothing to be done for `all'.
tools\make.exe[2]: Leaving directory `E:/Open CPU/qlcode'
tools\make.exe[2]: Entering directory `E:/Open CPU/customer'
tools\make.exe[2]: Nothing to be done for `all'.
tools\make.exe[2]: Leaving directory `E:/Open CPU/customer'
*****LINK LIBS to ELF*****
"build\scat6223.txt", line 19: Warning: L6314W: build\scat6223.txt(line 19, col
24) No section matches pattern ql_main.*o(ZI).
Warning: L6305W: Image does not have an entry point. (Not specified or not set d
ue to multiple choices.)
Finished: 0 information, 2 warning and 0 error messages.
*****CREATE BIN FILE*****
c:\progra~1\arm\advu1_2\bin\fromelf.exe -output build\ql_custom.bin -
bin build\ql_custom.elf
tools\make.exe[1]: Leaving directory `E:/Open CPU'
E:\Open CPU>

```

Build successfully!

4.5. Download Embedded Application image

Please refer to the document “Upgrade_FW_Tools_UGD”.

5. Startup Embedded Application

In default setting, Embedded Application will not startup. To startup Embedded Application, please following steps as below.

1. Power on module, execute “AT+QLOCPU=1” to enable Open CPU function.
2. Reset module, Open CPU will be enabled and Embedded Application will be executed.

If customer wants to disable Open CPU, customer should send “AT+QLOCPU=0\n” to module with FCM API `Ql_SendToModem` in Embedded Application.

6. Debugging

Open CPU provides two debug mode, basic mode and advance mode. Debug mode can be set with API `Ql_SetDebugMode()`.

In basic mode, customer debug messages will be output as text to debug serial port(UART port 2). In advance mode, customer debug messages and system debug messages will be output to debug serial port in special format. Quectel will also provide the catcher tool to capture and analyze these messages. The default debug mode is advance mode.

Customer can call API **QI_DebugTrace()** to output debug message. For example,
“**QI_DebugTrace(“debug information: line %d is %s\r\n”, linenumber, linecontent);**”
”

QUECTEL



Shanghai Quectel Wireless Solutions Ltd.

Room 801, Building E, No.1618, Yishan Road, Shanghai, China 201103

Tel: +86 21 5108 2965

Mail: info@quectel.com