



Perfect Wireless Experience
完美无线体验

G610 Hardware OpenCPU User Manual

Version : V1.0.2

Date : 2015.08.25



Copyright

Copyright ©2015 Fibocom Wireless Inc . All rights reserved.

Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or transmit the document in any form.

Attention

The document is subject to update from time to time owing to the product version upgrade or other reasons. Unless otherwise specified, the document only serves as the user guide. All the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Trademark



The trademark is registered and owned by Fibocom Wireless Inc.

Revision History

Version	Date	Remarks
V1.0.0	2013-11-12	Initial Version
V1.0.1	2014-04-14	Update section 2.4
V1.0.2	2015-08-25	Update the logo.

Applicability Type

No.	Type	Note
1	G610-A20-00	Auto power on
2	G610-A20-01	Non -auto power on

Content

1 Preface.....	5
1.1 Overview.....	5
1.2 Specification.....	6
2 Hardware Design.....	7
2.1 Power On/Off.....	7
2.1.1 Turning on the Module.....	7
2.1.2 Turning off the Module.....	7
2.2 UART.....	7
2.2.1 UART1.....	8
2.2.2 UART2.....	8
2.2.3 OpenCPU Application Note for UART1&UART2.....	9
2.2.4 HOST UART.....	10
2.2.5 OpenCPU Application Note for HOST UART.....	10
2.2.6 OpenCPU Application Note for Virtual UART.....	10
2.3 Sleep Mode.....	11
2.4 GPIO Interface.....	11
2.5 INT Interface.....	12
2.6 Watchdog.....	12

1 Preface

1.1 Overview

Definition:

“Standard module” means the software of the module supports standard AT command, which doesn't support OpenCPU function;

“OpenCPU module” means the software of the module supports OpenCPU function. Customers can make secondary development based on the module, download and run customer software. After the module is powered up, besides its own software, you can also load customer software.

“Module Firmware” means the software comes with the module; usually it is already downloaded by default.

“Customer Software” means the secondary development software, usually customer need to download it to the module.

Some interface definitions of the module are different due to different software.

This document is based on *G610 Hardware User Manual* and *OpenCPU API User Manual*, it introduces some methods and notes of hardware interface design and functional design when you use OpenCPU module.

1.2 Specification

Item	G610	
Hardware	Base Band	RDA8851(MIPS), 208MHz
	Flash Code Space (BYTE)	512K
	Flash Data Space (BYTE)	512K
	RAM (BYTE)	512K
	UART	3
	GPIO	10
	INT	3
Software	File System	No limits with file number. File storage space is 512k; each file cannot exceed 100k.
	Software Timer	Unit: ms (higher than 500ms is recommended) You can enable at most 25 software timers at the same time.
	Thread	5
	Watchdog	Supported
	TCP Server	Supported
Support	UART1 Upgrade module firmware	Provided
	UART1Upgrade customer software	Provided
	Remote Upgrade customer software	Supported
	Debug Method	UART1, UART2 print and HOST UART trace

2 Hardware Design

2.1 Power On/Off

The module power on and off related to two hardware signals: POWER_ON and VDD.

The POWER_ON signal is an important signal.

The VDD signal indicates whether module is powered on or off. When this signal is disabling (0V), module is powered-off. When it is output (2.8V), module is powered-on.

The following table shows definitions of the pins for Power on/off.

Pin No.	Signal Name	Description
10	POWER_ON	Power on and off module Low level activated
9	VDD	Illustrating module status When level is low, module is off When level is high, module is on

2.1.1 Turning on the Module

For “Standard module”, the module won’t start directly after you powered up, it is turned off.

For “OpenCPU module”, normally external circuit is not controlled by any MCU or AP, so after the module is powered up, it automatically powered on. When you design the hardware, for G610-A20-01, we suggest you connect POWER_ON pin to the ground via 470ohm.

2.1.2 Turning off the Module

For “OpenCPU module”, if external circuit is not controlled by any MCU or AP, please do not use any operation or AT command which will cause the module power off.

2.2 UART

For “Standard module”, it has 3 UART ports.

UART1 is a completely independent 4 wire serial bus interface. This is the main UART.

UART2 is a 2 wire serial bus interface.

HOST UART is a debug UART, which is used for downloading, calibrating, trace and so on, it doesn't support any AT command.

For "OpenCPU module", it has one more UART which is Virtual UART; it is used for sending AT command from "Customer software" to module "Module firmware".





2.2.1 UART1

UART1 is used for all the communications with module, it can program, upgrade "Module firmware" and "Customer software".

The module is defined as a DCE device, and the user application is defined as the DTE device. These definitions apply for the UART signals naming conventions, and the direction of data flow, as described in the following tables.

Pin No.	Signal Name	Description	Notes
45	RXD_N	Module Transmitted Data	G610 Transmitted Data
44	TXD_N	Module Received Data	G610 Received Data
42	RTS_N	Module Request To Send	G610 Switch To Received Mode
43	CTS_N	Clear To Send	G610 Notice DTE Requested To Send

Recommended connection:

Application MCU	Direction	Module	
TXD		Pin 44	TXD_N
RXD		Pin 45	RXD_N
RTS		Pin 42	RTS_N
CTS		Pin 43	CTS_N

All flow control handshakes are supported: hardware or none.

The UART1 is configured to be 115200bps, 8 data bits, 1 stop bit and no parity.

2.2.2 UART2

UART2 communicates with outside.

Pin No.	Signal Name	Description	Feature
30	UART2_TXD	Module Received Data	G610 Transmitted Data
29	UART2_RXD	Module Transmitted Data	G610 Received Data

Recommended connection:

Application MCU	Direction	Module	
TXD		Pin 30	UART2_TXD
RXD		Pin 29	UART2_RXD

The UART2 is configured to be 115200bps, 8 data bits, 1 stop bit and no parity.

2.2.3 OpenCPU Application Note for UART1&UART2

Interface Function	Input Parameter
<pre>Void sys_uart_output (INT32 id, UINT8 *buff, UINT16 len)</pre>	<pre><id> UART ID, 0 means UART1, 1 means UART2 <buff> data pointer <len> data length</pre>
<pre>Void (*uart_input) (INT32 uid, UINT8 *data, UINT16 len)</pre>	<pre><uid> UART ID <data> UART data pointer <len> UART data length</pre>

You can set the baud rate of these two UART ports by API interface:

Interface Function	Input Parameter
<pre>INT32 sys_set (GAPP_OPTION_ID_T id, Void *arg,</pre>	<pre><id> operation ID <arg> parameter pointer <len>parameter length</pre>

<pre> UINT16 len) </pre>	
---------------------------	--

Note: Do not use AT command to set baud rate.

2.2.4 HOST UART

HOST UART is a debug UART, which is used for downloading, calibrating, trace and so on; it doesn't support any AT command. This interface is only used when debugging, users only need to connect to the test point.

Application MCU	Direction	Module	
TXD	→	Pin 48	HST_TXD
RXD	←	Pin 55	HST_RXD

2.2.5 OpenCPU Application Note for HOST UART

HOST UART can only output debug data, there are two formats.

Interface Function	Input Parameter
<pre> INT32 sys_eventTrace (UINT32 value) </pre>	<value> 32 bytes data
<pre> INT32 sys_textTrace (INT8 *fmt, ...) </pre>	Parameters like printf

2.2.6 OpenCPU Application Note for Virtual UART

Interface Function	Input Parameter
<pre> INT32 sys_at_send (UINT8 *cmd, UINT16 len) </pre>	<pre> <cmd> AT command string (including 0x0d) <len> string length </pre>

)	
Void (*at_resp) (UINT8 *rsp, UINT16 rsplen)	<rsp> AT port data starting address <rsplen> the data length

2.3 Sleep Mode

“OpenCPU module” supports sleep mode.

It goes into sleep mode by ATS24 command via Virtual UART.

When “Customer software” invokes the timer of “Module firmware”, it would affect the sleep mode,

This could further affect the consumption of the module.

The other operations of sleep mode are the same with “Standard module”.

2.4 GPIO Interface

G610 integrates 10 GPIO interfaces, users can control them by AT command, configure them to be output or input, output high level or read the external input level.

Pin No.	Pin Name	Standard Module	OpenCPU Module
35	GPIO01		Input/output
34	GPIO02		Input/output
33	GPIO03		Input/output
32	GPIO04		Input/output
31	GPIO07		Input/output
54	GPIO36		Input/output
38	DSR_N	Data Set Ready	Output
39	RING_N	Ring Indicator	Output
41	DCD_N	Data Carrier Detect	Output
49	LPG	LPG (by default)	This is available after set by sys_set, there is a

			LPG CONTROL option in sys_set to control LPG.
43	UART1_CTS	UART1_CTS	Output only
42	UART1_RTS	UART1_RTS	Input only

2.5 INT Interface

Pin No.	Pin Name	OpenCPU Module	Description
40	DTR_N		Rising edge or falling edge effective
3	SIM_CD		Low level effective
47	HS_DET		Low level effective, need external pull-up

The usage as listed below:

Interface Function	Input Parameter
<pre> INT32 sys_set (GAPP_OPTION_ID_T id, Void *arg, UINT16 len) </pre>	<pre> <id> operation ID <arg> parameter pointer <len>parameter length </pre>

2.6 Watchdog

“OpenCPU module” supports watchdog function, which avoid module crashes or other abnormal situation occurs. This function is disabled by default, you can invoke API to enable it and feed the dog. For details, please refer to *OpenCPU API User Manual*.