FIBOCOM
Perfect Wireless Connection

G510 Hardware OpenCPU User Manual

Version: V1.0.3

Date: 2014-04-14

Confidential Material

This document contains information highly confidential to Fibocom Wireless Inc. (Fibocom). Fibocom offers this information as a service to its customers, to support application and engineering efforts that use the products designed by Fibocom. The information provided is based upon requirements specifically provided to Fibocom by the customers. All specifications supplied herein are subject to change. Disclosure of this information to other parties is prohibited without the written consent of Fibocom.

Copyright

Trademarks Notice

Revision History

Version	Date	Remarks
V1.0.0	2013-06-21	Initial Version
V1.0.1	2013-07-31	Change M2M to OpenCPU; Update some descriptions
V1.0.2	2014-03-10	Add GPIO application spec,and need specially choose and use
V1.0.3	2014-04-14	Add OpenCPU modules and INT interface function.

Applicability Table

No.	Type	Note
1	G510-Q50-00	Standard model, it can be upgraded by integrated software
2	G510-Q50-90	Integrated model number, the external label is different from other models’
3	G510S-Q50-00	CE certification included, it can be upgraded by integrated software

Contents

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 *G510 Hardware User Manual* and *G5-Family OpenCPU API User Manual*, it introduces some methods and notes of hardware interface design and functional design when you use OpenCPU module.

1.2 OpenCPU features

Item	G510	
Hardware	Base Band	RDA8851(MIPS), 208MHz
	Flash Code Space（BYTE）	512K
	Flash Data Space（BYTE）	512K
	RAM（BYTE）	512K
	UART	3
	GPIO	5
	INT	2
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 UAR trace

2 Hardware Design

2.1 Power On/Off

The module power on and off is the two primary phases, which are related at the interface connector by the hardware signals POWER_ON, VDD.

The POWER_ON signal is a 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 (2.85V), module is powered-on.

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

Pin No.	Signal Name	Description
14	POWER_ON	Power on and off module Low level activated
13	VDD	Illustrating module status LDO power output 0V : module is off LDO power output 2.85V : 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, 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
19	UART1_TXD	Module Transmitted Data	G510 Transmitted Data
18	UART1_RXD	Module Received Data	G510 Received Data
21	UART1_CTS	Module Clear To Send	G510 Switch To Received Mode
20	UART1_RTS	Request To Send	G510 Notice DTE Requested To Send

Recommended connection:

Application MCU	Direction	Module	
TXD	→	Pin 18	UART1_RXD
RXD	←	Pin 19	UART1_TXD
RTS	→	Pin 21	UART1_CTS
CTS	←	Pin 20	UART1_RTS

All flow control handshakes are supported: hardware or none.

The UART1 is configured to be115200bps, 8 data bits, 1 stop bit and no parity.

2.2.2 UART2

UART2 communicates with outside.

Pin No.	Signal Name	Description	Feature
40	UART2_TXD	Module Transmitted Data	G510 Transmitted Data
39	UART2_RXD	Module Received Data	G510 Received Data

Recommended connection:

Application MCU	Direction	Module	
TXD	→	Pin 39	UART2_RXD
RXD	←	Pin 40	UART2_TXD

The UART2 is configured to be115200bps, 8 data bits, 1 stop bit and no parity.

2.2.3 OpenCPU Application Note for UART1&UART2

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

UINT16 len)	
Void (*uart_input) (INT32 uid, UINT8 *data, UINT16 len)	<uid> UART ID <data> UART data pointer <len> UART data length

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

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

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 11	HST_RXD
RXD	←	Pin 12	HST_TXD

2.2.5 OpenCPU Application Note for HOST UART

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

Interface Function	Input Parameter
INT32 sys_eventTrace (UINT32 value)	<value> 32 bytes data
INT32 sys_textTrace	Parameters like printf

(INT8 *fmt, ...)	

2.2.6 OpenCPU Application Note for Virtual UART

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

GPIO	Standard Module	OpenCPU Module	Description
IO0 (pin 22)	UART1_RING	GAPP_IO_0	Only as output,and refer to the notes below for detailes.
IO1 (pin 17)	UART1_DCD	GAPP_IO_1	Only as output,and refer to the notes below for detailes.
IO2 (pin 16)	UART1_DSR	GAPP_IO_2	Only as output,and refer to the

			notes below for detailes.
IO3 (pin 15)	UART1_DTR	GAPP_IO_3	Input/output,low level for default.
IO4 (pin 38)	LPG	GAPP_IO_4， LPG (by default)	Input/output,it is available after set by sys_set, there is a LPG CONTROL option in sys_set to control LPG.
IO5（pin20）	UART1_RTS	GAPP_IO_5	Output only
IO6（pin21）	UART1_CTS	GAPP_IO_6	Input only

Note：

G510，the reused GPIO below,need choosen according to the actual situation.

UART1_RING –IO0, only as the output,when power on,the progress of module initialization will last 1.4s uncontrolled low level,while the external pull-up is invalid.

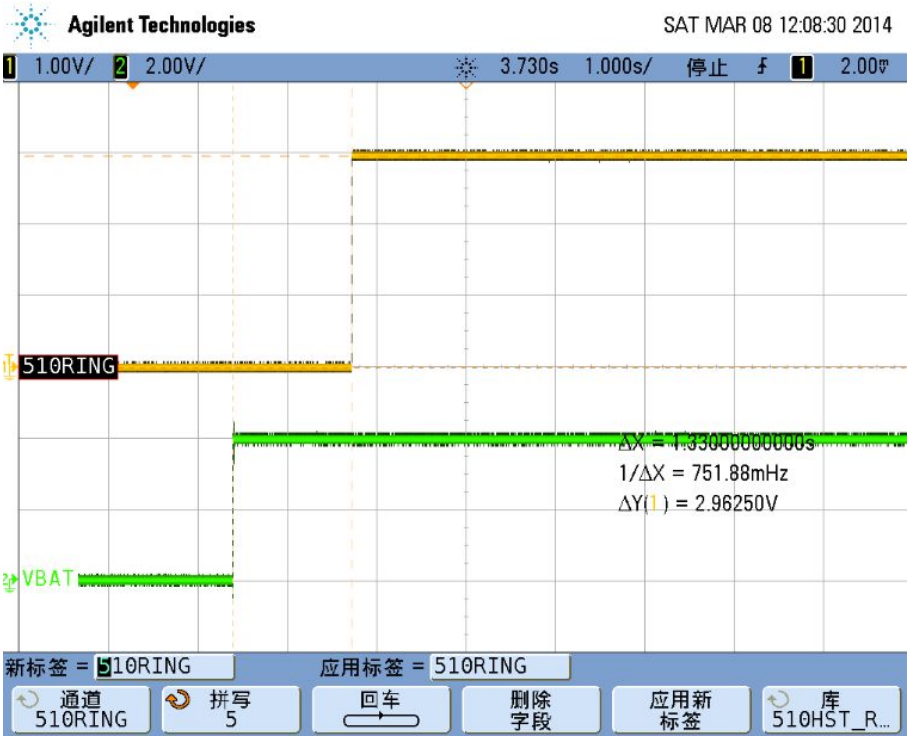

Figure 2.1

UART1_DCD –IO1, only as the output,when power on,the progress of module initialization will last 1.4s uncontrolled “first hign last low” level,while the external pull-up and pull-down are all invalid.

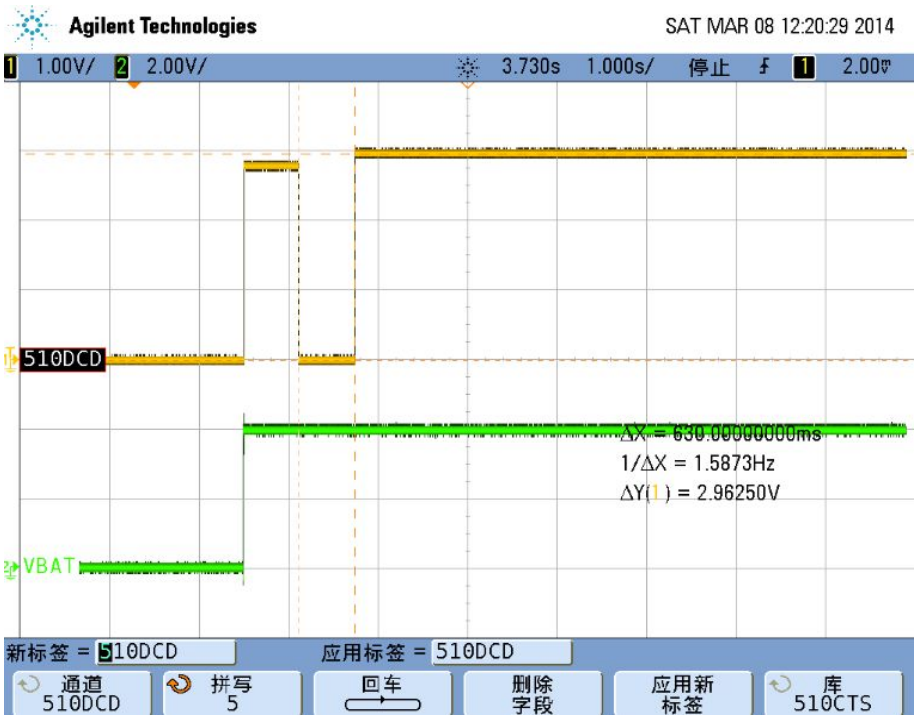

Figure 2.2

UART1_ DSR –IO2, only as the output,when power on,the progress of module initialization will last 1.4s uncontrolled continued hign level,while the external pull-down is invalid.

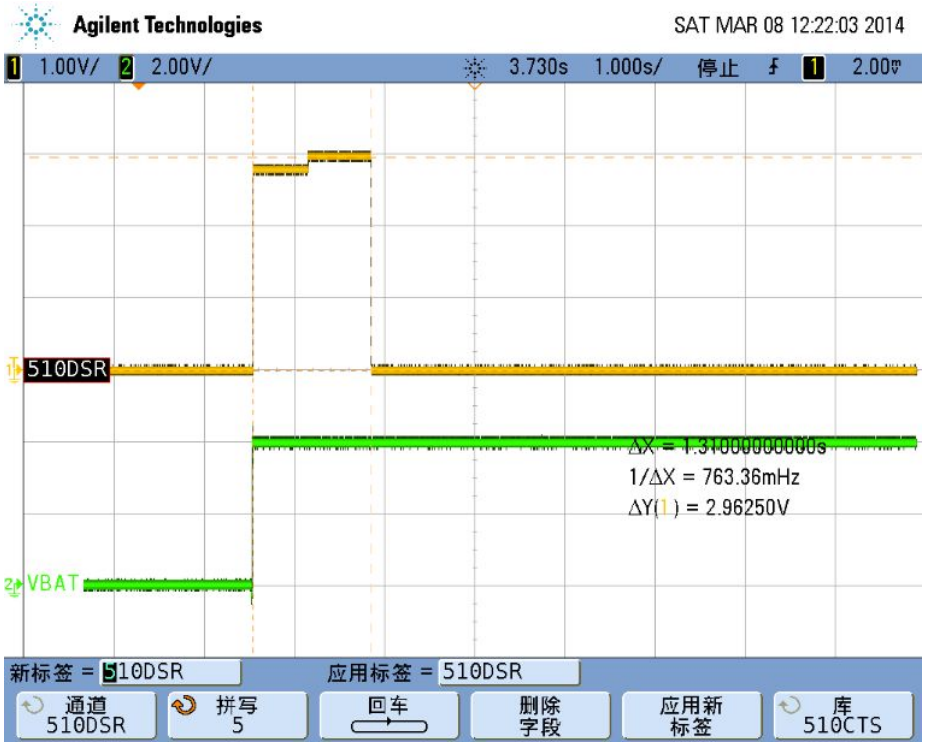

Figure 2.3

UART1_ DTR –IO3,as input or output,and when power on,the progress of module initialization will last low level, while the external pull-up is available.

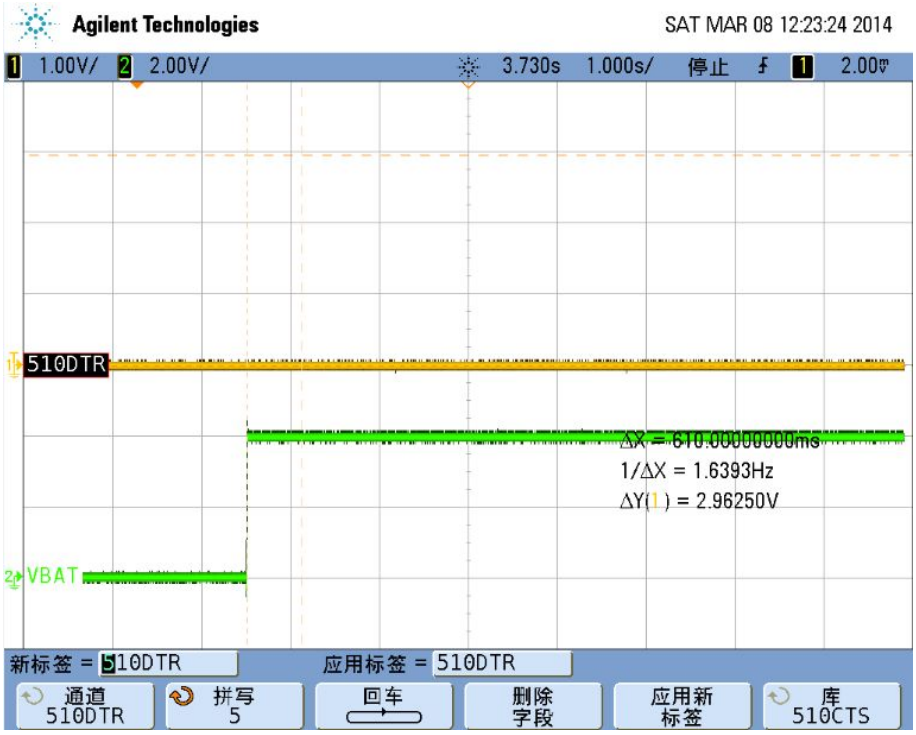

Figure 2.4

2.5 INT Interface

GPIO	Standard Module	OpenCPU Module	Description
PIN41	WAKE_UP		falling edge effective, need external pull-up
PIN27	SIM_CD		falling edge effective, need external pull-up

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

INT32 sys_gpio_get (GAPP_GPIO_ID_T id, UINT8 *level)	<id>IO number，9 means SIM_CD，10 means WAKE_UP <level>return status，means the current level,high or low 0 means low level, 1 means high level

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 *G5-Family OpenCPU API User Manual*.