WCDMA UGxx
Power Management Application Note

UMTS/HSPA Module Series

Rev. WCDMA_UGxx_Power_Management_Application_Note_V1.2

Date: 2015-04-01

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Mail: info@quectel.com

Or our local office, for more information, please visit:
http://www.quectel.com/support/salesupport.aspx

For technical support, to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
Or Email: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2015-01-16	Wythe WANG	Initial
1.1	2015-03-06	Wythe WANG	Changed the document name from “UG95” to “UGxx”.
1.2	2015-04-01	Wythe WANG	Updated applicable modules.

Contents

Table Index

Figure Index

1 Introduction

When the module is embedded into host system, it will increase the whole power consumption of the system. Therefore, Quectel provides several power management solutions on the module for the host to lower the overall power consumption by managing module working mode.

This document is applicable to UGxx modules.

The module power management solutions mainly consist of the following mechanisms:

- Host controls module to enter into power saving mode
- Host wakes up module
- The module wakes up host

This document illustrates hardware interface and the URC event in module power management first, and then elaborates typical application scenarios about power saving solutions. In the end, it briefly describes the sleep and wakeup software mechanism of module.

2 Hardware Interface

The communication between module and host is mainly realized through USB and UART port. Besides, the module also provides other functional pins for host to manage power.

2.1. Pin Assignment

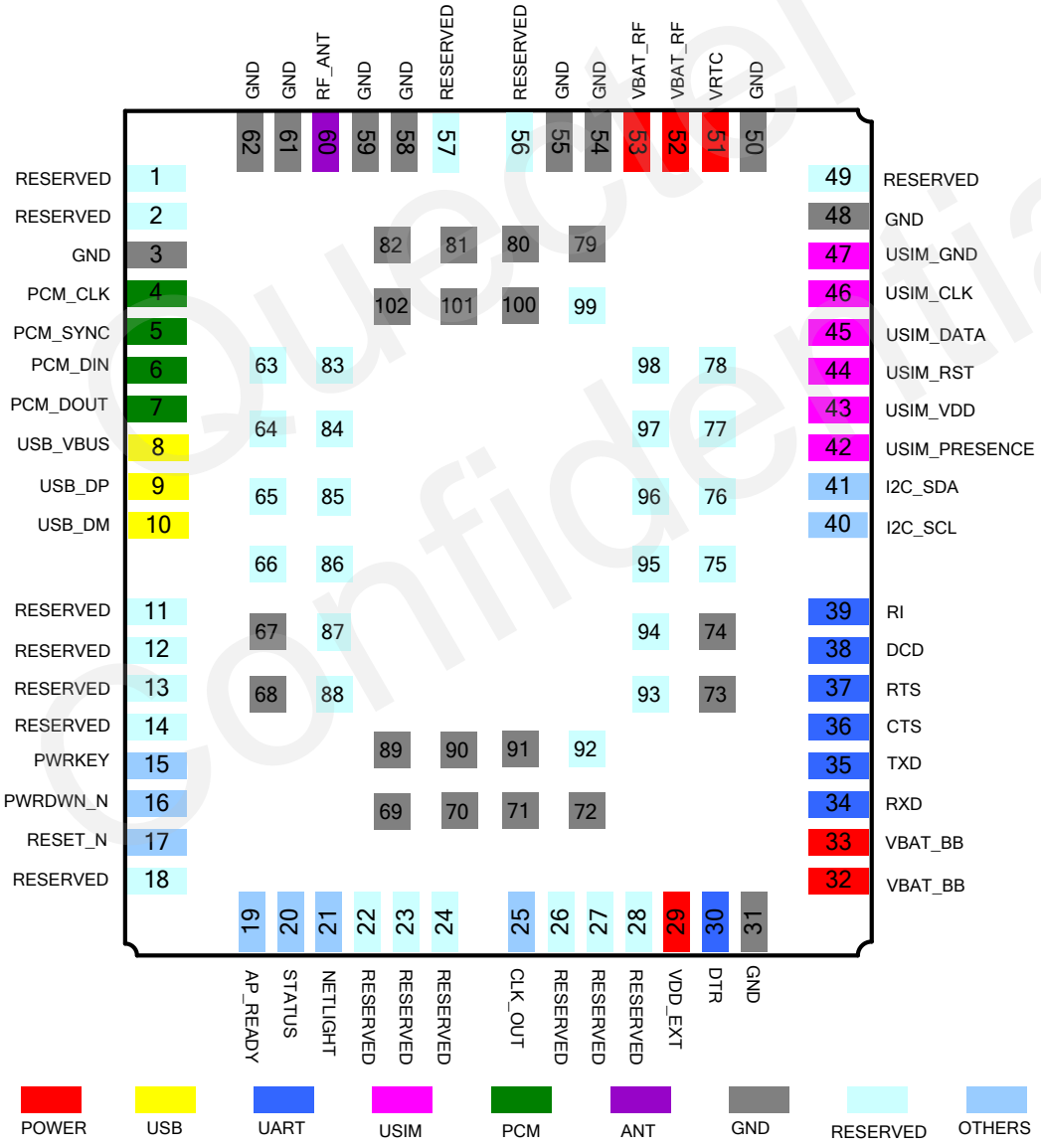

Figure 1: Pin Assignment (Top View)

2.2. Power Management Pin

Table 1: Power Management Pin

Pin Name	Input/Output	Description
DTR	IN	Used for host to wake up module from power saving mode or allow module to enter into the power saving mode. When it is in high level, it means host allows module to enter into power saving mode. When the module is in sleep mode, pulling down DTR will wake up module. DTR has been pulled up by default.
RI	OUT	Used for module to inform the host there is URC to report. When the module has no URC to report, RI will keep in high level. When there is URC to report, RI will output a low pulse. If host is in sleep mode, it will be waked up and process URC.

2.3. DTR

Pulling up DTR indicates the module is allowed to enter into power saving mode (To detect whether the module has been in power saving mode already, other conditions are needed to be met). If DTR is pulled down, module would not enter into power saving mode. When it has entered into power saving mode, you can pull down DTR to wake it up.

2.4. RI

When module needs to send URC to host, RI pin of the module will generate a low pulse (It will last 120ms by default, and the time can be configured by AT command) to inform the host. When the host is in power saving mode, you can wake it up by detecting the low pulse.

Furthermore, in order to ensure the host can handle URC efficiently, you can set to delay URC sending by AT command. For more details about AT command, please refer to ***Chapter 5***.

3 Introduction to URC

In normal data interaction process, the module not only responds the request from host, but also takes the initiative to report external events to host, such as incoming call and short message. In general, the unsolicited information reported by the module is called URC (Unsolicited Result Code).

The processing of URC is embodied in two aspects: report event information to host actively and generate low pulse on RI pin. In module power management, RI pin can inform the incoming URC to host in the form of pulse after the host enters into sleep mode. For the module supports several types of URC, frequent URC report will repeatedly wake up the host from the sleep mode, which will reduce the power management efficiency of the whole system and reduce the effect of power managing. In order to avoid this, host should select corresponding URCs based on its own needs.

3.1. URC Event

The module contains various URCs. For example, network status URC is used to report the current network status to host. Short message URC is used to report the new incoming short messages. And phone call URC is used to report the call state. For most URCs, you can configure whether to report them or not by AT command. So host can open or close URC report in different system status according to its own needs. For instance, when you first start the system, you can set to actively report the URC relevant to network, short message and phone call by AT command. Before the host enters into sleep mode, you can forbid the report of URC related to network via AT command, but only report the URC relevant to short message and phone call. After the host has entered into sleep mode, it will only be waked up by URC relevant to short message and phone call. In this way, the basic requirements of the host have been met, and the power management efficiency of whole system has been improved as well.

3.2. Example of URC Configuration

```
//The following example shows how to configure the relevant URCs:
AT+CREG=1 or AT+CREG=2          //Enable to actively report the CS network registration status
                                  (Disabled by default)
OK
AT+CREG=0                       //Disable to actively report the CS network registration status
OK
AT+CGREG=1 or AT+CGREG=2        //Enable to actively report the PS network registration status
```

```
                                    (Disabled by default)
OK
AT+CGREG=0                          //Disable to actively report the PS network registration status
OK
```

NOTE

For more details about AT commands, please refer to *Quectel_WCDMA_UGxx_AT_Commands_Manual*.

4 Application Scenarios for Power Saving Solutions

4.1. Power Saving Solutions Based on UART Port

4.1.1. Requirements

When the device is implementing power saving solutions, it is required that both host and module are able to enter into power saving mode. Host is communicated with module by UART port, not by USB port. This chapter illustrates the detailed implementation method based on these requirements.

4.1.2. Hardware Connection Configuration

1. Host is connected with module by UART port.
2. Host I/O is connected with module RI pin, and the pin connected with the host must be able to wake up.
3. Host I/O is connected with module DTR pin.

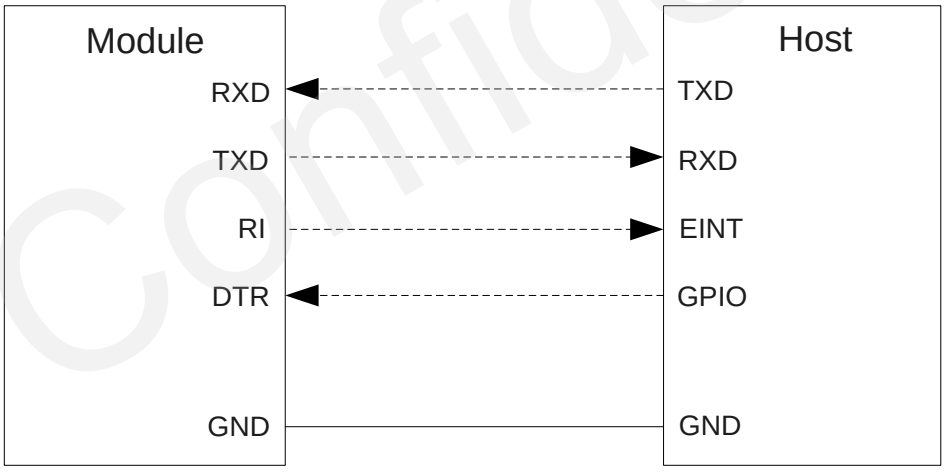

Figure 2: Connection Diagram Based on UART

UART port is an important channel for the communication between module and host. Through reusing the DTR functional pin in UART port, host can manage the module power easily. When DTR is in high level, the module will be allowed to enter into power saving mode. When DTR is in low level, the module cannot enter into power saving mode.

NOTE

The connection in dotted line in Figure 2 indicates that the level match between the two systems should be paid attention to in actual application. The connection in solid line means they can be connected directly. Same as the lines in the following connection diagrams of this chapter.

4.1.3. Software Initialization Configuration

1. Configure **AT+QSCLK** control variables to enable module to enter into power saving mode by **AT+QSCLK=1**;
2. Configure **AT+QCFG="uart/power",0,0** to set main port on auto-close mode.

4.1.4. Implementation Method

4.1.4.1. Module Enters into Power Saving Mode

When host enters into sleep mode, pulling up DTR pin enables the module to enter into power saving mode.

4.1.4.2. Host Wakes up Module

Host pulls down DTR pin to indicate host is in ready mode now.

4.1.4.3. Module Wakes up Host

When the module has URC to report, its RI pin will generate a low pulse (Duration time is 120ms by default, and it can be configured by AT command) to inform the host that URC is coming. When the host is in sleep mode, it is required to ensure that the change of RI pin level can wake up the host itself. After host is waked up, DTR pin should be set as low to enable main port.

4.2. Power Saving Solutions Based on USB Remote Wakeup

4.2.1. Requirements

When the device is implementing power saving solution, it is required that both host and module are able to enter into power saving mode, and host should support USB suspend/resume and remote wakeup functionality. This chapter illustrates detailed implementation method based on these requirements.

4.2.2. Hardware Connection Configuration

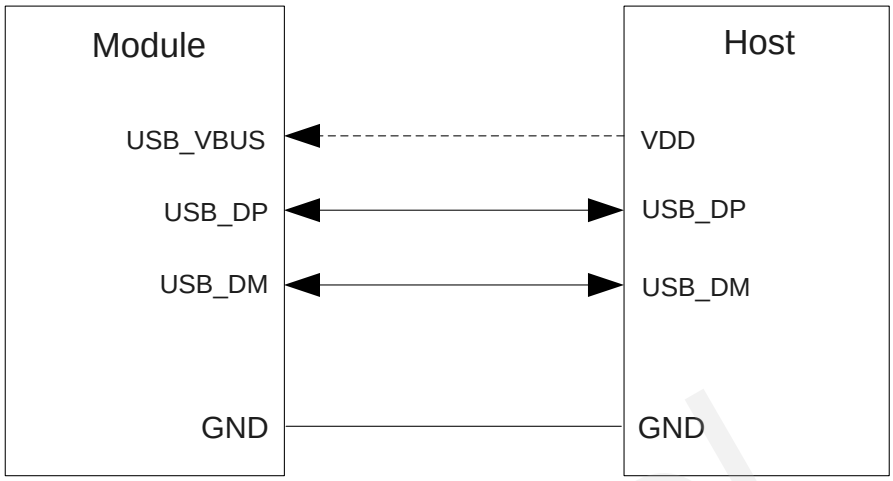

Figure 3: Connection Diagram Based on USB Remote Wakeup

4.2.3. Software Initialization Configuration

1. Configure **AT+QSCLK** control variables to enable module to enter into power saving mode by **AT+QSCLK=1**;
2. Configure **AT+QCFG="uart/power",0,0** to set main port on auto-close mode;
3. Configure the corresponding system and driver settings in host system. (For details, please contact Quectel technical support.)

4.2.4. Implementation Method

4.2.4.1. Module Enters into Power Saving Mode

Host sends suspend command to module on USB BUS, and module enters into power saving mode.

4.2.4.2. Host wakes up Module

Host sending data to module through USB will change USB bus from suspend status to resume status to wake up the module.

4.2.4.3. Module Wakes up Host

When the module has URC to report, module will send remote wake-up signals to host on USB BUS. Host USB should be able to be waked up through these signals.

4.3. Power Saving Solutions Based on USB Disconnection

4.3.1. Requirements

When the device is implementing power saving solution, it is required that both host and module are able to enter into power saving mode, and host should not support USB suspend/resume and remote wakeup functionality. So host should disconnect the USB when the device enters into power saving mode. This chapter illustrates detailed implementation method based on these requirements.

4.3.2. Hardware Connection Configuration

1. Connect host with module via USB;
2. Host I/O is connected with module RI pin, and the pin must be able to suspend wakeup.

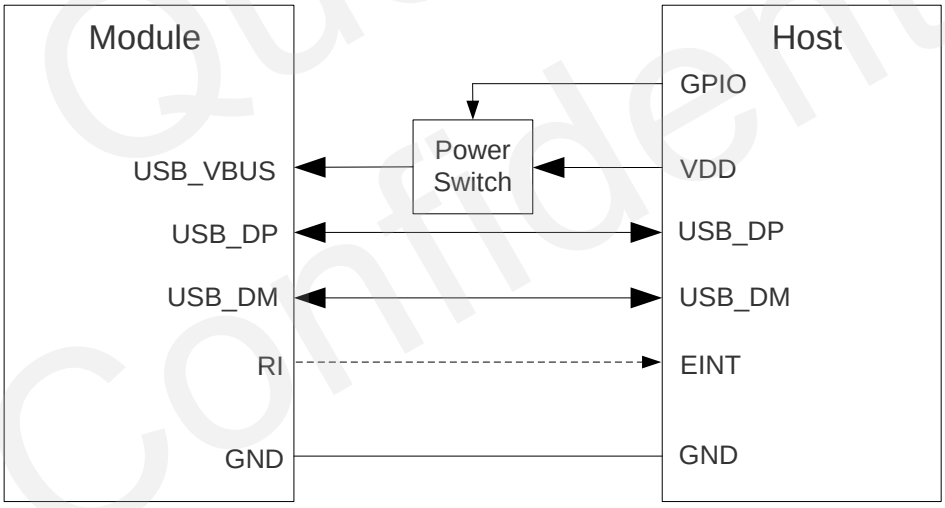

Figure 4: Connection Diagram Based on USB Disconnection

4.3.3. Software Initialization Configuration

1. Configure **AT+QSCLK** control variables to enable module to enter into power saving mode by **AT+QSCLK=1**;
2. Configure **AT+QCFG="uart/power",0,0** to set main port on auto-close mode.

4.3.4. Implementation Method

4.3.4.1. Module Enters into Power Saving Mode

Host pulls down module's USB_VBUS.

4.3.4.2. Host Wakes up Module

Host pulls up module's USB_VBUS to wake up the module. Then host begins to send data to the module.

4.3.4.3. Module Wakes up Host

When the module has URC to report, RI pin will generate a low pulse (Duration time is 120ms by default, and it can be set by AT command). Host must be able to be waked up by RI pin. After the host has been waked up, pull up module's UAB_VBUS to indicate the module outputs URC to host.

4.4. Power Saving Solutions Based on USB Suspend/Resume and RI Pin

4.4.1. Requirements

When the device is implementing power saving solution, it is required that both host and module are able to enter into power saving mode. Meanwhile, host should support USB suspend/resume, but need not support USB remote wakeup functionality, and module RI pin should be the external wakeup source of the host. So, host should disconnect the USB when the device enters into power saving mode. This chapter illustrates detailed implementation method based on these requirements.

4.4.2. Hardware Connection Configuration

1. Connect host with module via USB;
2. Host I/O is connected with module's RI pin, and the pin must be able to suspend wakeup.

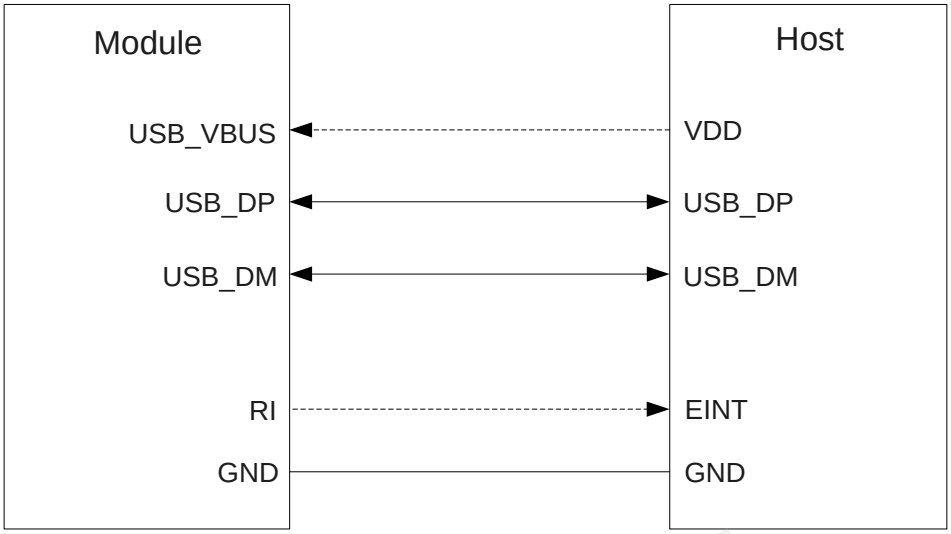

Figure 5: Connection Diagram Based on USB Suspend/Resume and RI Pin

4.4.3. Software Initialization Configuration

1. Configure **AT+QSCLK** control variables to enable module to enter into powering saving mode by **AT+QSCLK=1**;
2. Configure **AT+QCFG="uart/power",0,0** to set main port on auto-close mode.

4.4.4. Implementation Method

4.4.4.1. Module Enters into Power Saving Mode

Host sends suspend command to module on USB BUS to make module enter into powering saving mode.

4.4.4.2. Host Wakes up Module

Host sends resume command to module through USB BUS to wake up the module.

4.4.4.3. Module Wakes up Host

When the module has URC to report, RI pin will generate a low pulse (Duration time is 120ms by default, and the time can be set by AT command). Host must be able to be waked up by RI pin.

5 AT Command

5.1. AT+QCFG="uart/power" Turn on/off the UART Port

AT+QCFG="uart/power" Turn on/off the UART Port	
Test Command **AT+QCFG=?**	Response **……** **+QCFG: "uart/power", (0,1),(0-2)** **……** **OK**
Write Command **AT+QCFG="uart/power"[,<mode>]**	Response If <mode> is absent: **+QCFG: "uart/power",<uart>,<mode>** **OK** If not absent: **OK** **ERROR**
Reference	

Parameter

<uart>	0	UART1
	1	UART3
<mode>	0	Auto turn on/off the UART port
	1	Turn on the UART port
	2	Turn off the UART port

NOTE

The UART3 does not support mode 0.

Example

```
AT+QCFG="uart/power",0, 0          //Auto turn on/off the UART 1 port

OK
```

5.2. AT+QCFG="urc/ri/ring" Ring Line Behavior of RING

AT+QCFG="urc/ri/ring", **AT+QCFG="urc/ri/smsincoming"** and **AT+QCFG="urc/ri/other"** control the behavior of ring line when URC is reported. These configurations will be stored into NV automatically. The ring line is active low. **AT+QCFG="urc/ri/ring"** specifies the behavior of ring line when RING is presented to indicate an incoming call.

The sum of parameter **<activeduration>** and **<inactiveduration>** determines the interval time of "RING" indications when a call is coming.

AT+ QCFG="urc/ri/ring" Ring Line Behavior of RING	
Write Command **AT+QCFG="urc/ri/ring"[,<typeri>[,<pulseduration>[,<activeduration>[,<inactiveduration>[,<ringnodisturbing>]]]]]**	Response If configuration parameters are omitted (**+QCFG="urc/ri/ring"**), return current configuration: **+QCFG: "urc/ri/ring",<typeri>,<pulseduration>,<activeduration>,<inactiveduration>,<ringnodisturbing>,<pulsecount>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms

Parameter

<typeri>	The behavior of the ring line when URCs are presented	
	"off"	No change. Ring line keeps inactive
	"pulse"	Pulse. Pulse width determined by **<pulseduration>**
	"always"	Change to active. You may restore to inactive by **+QRIR**
	"auto"	When "RING" is presented to indicate an incoming call, ring line changes and keeps active. When ring of the incoming call

		ends, either answering or hanging up the incoming call, ring line will change to inactive.
	"wave"	When “RING” is presented to indicate an incoming call. The ring line outputs a square wave. Both **<activeduration>** and **<inactiveduration>** are used to set parameters of the square wave. When the ring of incoming call ends, either answering or hanging up the incoming call, ring line will change to inactive.
<pulseduration>	Set the width of pulse. Value ranges from 1 to 2000ms and default is 120ms This parameter is only meaningful when **<typeri>** is “pulse”. If this parameter is not needed, you can set it as null.	
<activeduration>	Set the active duration of the square wave, value ranges from 1 to 10000ms, and the default is 1000ms.This parameter is only meaningful when **<typeri>** is “wave”.	
<inactiveduration>	Set the inactive duration of the square wave, value ranges from 1 to 10000ms, and the default is 3000ms. This parameter is only meaningful when **<typeri>** is “wave”.	
<ringnodisturbing>	Set whether the ring line behavior could be disturbed. This parameter is only meaningful when **<typeri>** is configured to “auto” or “wave”. For example, when **<typeri>** is configured to “wave”, if you need the square wave not to be disturbed by other URCs (including SMS related URCs), you should set **<ringnodisturbing>** to "on".	
	"off"	It can be disturbed by other URCs of which the behavior of ring line is caused by an incoming call ringing.
	"on"	It cannot be disturbed by other URCs of which the behavior of ring line is caused by an incoming call ringing.
<pulsecount>	The count of pulse. This parameter is only meaningful when **<typeri>** is “pulse”. Value ranges from 1 to 5 and default is 1. The interval time between two pulse is equal to **<pulseduration>**.	

5.3. AT+QCFG="urc/ri/smsincoming" Ring Line Behavior of Incoming SMS

AT+QCFG="urc/ri/smsincoming" specifies the behavior of ring line when related incoming message URCs are presented. Related incoming message URCs list: **+CMTI,+CMT,+CDS,+CBM**.

AT+ QCFG="urc/ri/smsincoming"	Ring Line Behavior of Incoming SMS
Write Command **AT+QCFG="urc/ri/smsincoming"[,<typeri>[,<pulseduration>]]**	Response If configuration parameters are omitted (**+QCFG="urc/ri/smsincoming"**), return current configuration: **+QCFG: "urc/ri/smsincoming",<typeri>,<pulseduration>,<pulsecount>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms

Parameter

<typeri>	The behavior of the ring line when URC are presented "off" No change. Ring line keeps inactive "pulse" Pulse. Pulse width determined by **<pulseduration>** "always" Change to active. You should restore to inactive by **+QRIR**
<pulseduration>	Set the width of pulse. Value ranges from 1 to 2000ms and the default is 120ms. This parameter is only valid when **<typeri>** is "pulse".
<pulsecount>	The count of pulse. This parameter is only meaningful when **<typeri>** is "pulse". Value ranges from 1 to 5 and default is 1. The interval time between two pulse is equal to **<pulseduration>**.

5.4. AT+QCFG="urc/ri/other" Ring Line Behavior of Other URCs

AT+QCFG="urc/ri/other" specifies the behavior of ring line when other URCs are presented.

AT+ QCFG="urc/ri/other"	Ring Line Behavior of Other URCs
Write Command **AT+QCFG="urc/ri/other"[,<typeri>[,<pulseduration>]]**	Response If configuration parameters are omitted (**+QCFG="urc/ri/other"**), return current configuration: **+QCFG:**

	"urc/ri/other",<typeri>,<pulseduration>,<pulsecount> **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms

Parameter

<typeri>	The behavior of the ring line when URCs are presented "off" No change. Ring line keeps inactive "pulse" Pulse. Pulse width determined by **<pulseduration>**
<pulseduration>	Set the width of pulse. Value ranges from 1 to 2000ms and the default is 120ms. This parameter is effect only when **<typeri>** is “pulse”.
<pulsecount>	The count of pulse. This parameter is only meaningful when **<typeri>** is “pulse”. Value ranges from 1 to 5 and default is 1. The interval time between two pulses is equal to **<pulseduration>**.

5.5. AT+QCFG="urc/delay" Delay URC Indication

AT+QCFG="urc/delay" can delay the output of URC indication until ring line pulse ends.

AT+QCFG="urc/delay" Delay URC Indication	
Write Command **AT+QCFG="urc/delay"[,<enable>]**	Response If configuration parameter is omitted: **+QCFG: "urc/delay",<enable>** **OK** If configuration parameter is entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Reference	

Parameter

<enable>	0	URC indication will be output when ring line pulse starts
	1	URC indication will be output when ring line pulse ends (only the type of ring line is “pulse”, refer to **AT+QCFG="urc/ri/ring"**, **AT+QCFG="urc/ri/smsincoming" and AT+QCFG="urc/ri/other"**)

6 Software Mechanism

6.1. Module Power Saving Mechanism

There is a lower-priority sleep mission within module, which is used to detect whether module can enter into power saving mode or not. Other service missions (such as RF/USB/UART and so on) and power saving control variables have rights to vote on the power saving mission to decide whether or not the module can enter into power saving mode. When other service missions and power saving control variables both agree module to enter into power saving mode, power saving mission will be executed and module will enter into power saving mode. It should be noted that when module enters into power saving mode, RF will not be shut down but enter into DRX mode.

6.2. Module Power Saving Process

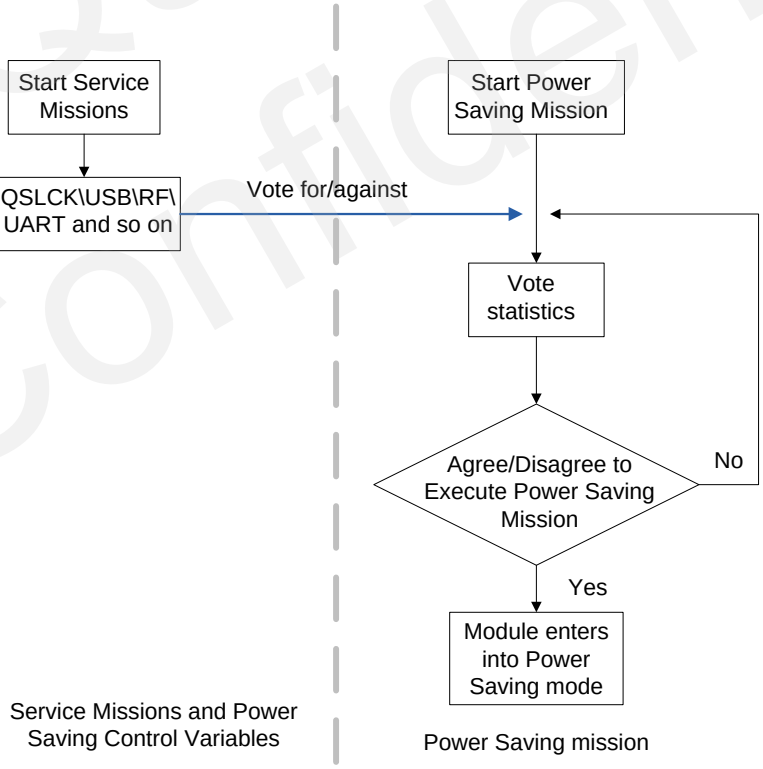

Figure 6: Module Power Saving Process

6.3. USB Power Saving Mechanism

When USB is working normally, root hub in hub or HCD will send SOF package periodically (full speed USB sends one package at an interval of 1ms, high speed USB sends one package at an interval of 125μs). At this time, USB on module will vote against the module entering into power saving mode.

According to the description of suspend in USB standard protocol, when the host system sets the port attached with USB device in hub or root hub as suspend, hub or root hub will stop sending SOF package and USB bus will enter into suspend mode. Then USB on module will enter into suspend mode, and vote for the module to enter into power saving mode.

NOTES

1. Quectel provides USB suspend solutions of module for different host systems. You can control module to enter into power saving mode on host system by these solutions.
2. If you use the USB driver developed by your own, the driver should support both global suspend and selective suspend functions.

6.4. Module Wakes up Host

6.4.1. Host Wake-up Event

Host wake-up event is the event that module takes the initiative to report information to host and wake up host when host is in sleep mode. Host wakeup event is also called URC.

6.4.2. USB Remote Wakeup

When host enters into sleep mode, and host USB bus is in suspend mode, if module has URC to report, it will send remote wakeup signals (duration time>3ms) to inform the host to resume USB.

The key points of module remote wakeup are listed as follows:

- Host USB controller must support remote wakeup, and it can wake up host. If USB controller does not support remote wakeup, it will not process remote wakeup signals received from the module. Similarly, if USB controller cannot wake up the host, then host will not be waked up from sleep mode by USB remote wakeup.

- As for host, module's remote wakeup function can be set by USB standard request. Host can enable remote wakeup function by "SET_FEATURE" and disable it by "CLEAR_FEATURE". Before host enters into sleep mode, module's remote wakeup function must be enabled.

- When module sends remote wakeup signals to host, host must keep resume signals for at least 20ms. After that, USB bus will enter into idle state. And before USB bus enters into idle state, USB controller must retrieve to send SOF package on USB bus within 3s. Otherwise, the module would enter into suspend mode again.

7 Appendix A Reference

Table 2: Terms and Abbreviations

Abbreviation	Description
USB	Universal Serial Bus
URC	Unsolicited Result Code
DRX	Discontinuous Reception
I/O	Input/output