BC66&BC66-NA QuecOpen Low Power Management Application Note

NB-IoT Module Series

Rev. BC66&BC66-NA_QuecOpen_Low_Power_Management_Application_Note_V1.1

Date: 2020-06-04

Status: Released

www.quectel.com

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

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE 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.

About the Document

Revision History

Version	Date	Author	Description
1.0	2019-06-19	Allan LIANG	Initial
1.1	2020-06-04	Sheffer GU	1. Added the applicable module BC66-NA. 2. Added the description of Ql_DeepSleep_Register API in Chapter 3.3. 3. Added the description of power-on and wake-up reasons of the modules in Chapter 5. 4. Added power consumption values of BC66-NA QuecOpen module in Chapter 6. 5. Added two important notes in Chapters 7.11 and 7.12.

Contents

Table Index

Figure Index

1 Introduction

Low power consumption is increasingly demanded in embedded applications, especially in battery-powered systems/applications. Reduced power consumption helps extend battery life, which means reduced operation cost and improved market competiveness of the products.

Minimizing of power consumption can be realized through software and hardware designs, and this document mainly introduces how to minimize power consumption of the modules through software design.

In QuecOpen® Solution, BC66 and BC66-NA modules provide minimized power consumption in Deep Sleep mode in which the CPU switches off the power supply and only RTC keeps running. In software design, please pay attention to the application logic difference between Cold Start (the first time power-up, or resetting) and Deep Sleep Wakeup. The following chapters introduce methods and important notes relating to low power management of BC66 and BC66-NA modules in QuecOpen® Solution.

2 Basic Concepts

You may have to know the following basic concepts before managing power consumption of BC66 and BC66-NA modules in QuecOpen solution.

2.1. DRX

DRX: Discontinuous Reception. It is an operating mode of Modem.

In order to reduce power consumption, the module monitors a NPDCCH channel once every DRX cycle to detect if there is any paging.

As illustrated below, DRX paging cycle refers to the period after the Modem enters idle mode and before T3324 times out. DRX cycle is usually short, therefore if there is any incoming data during DRX cycle, the time delay of receiving data is short as well.

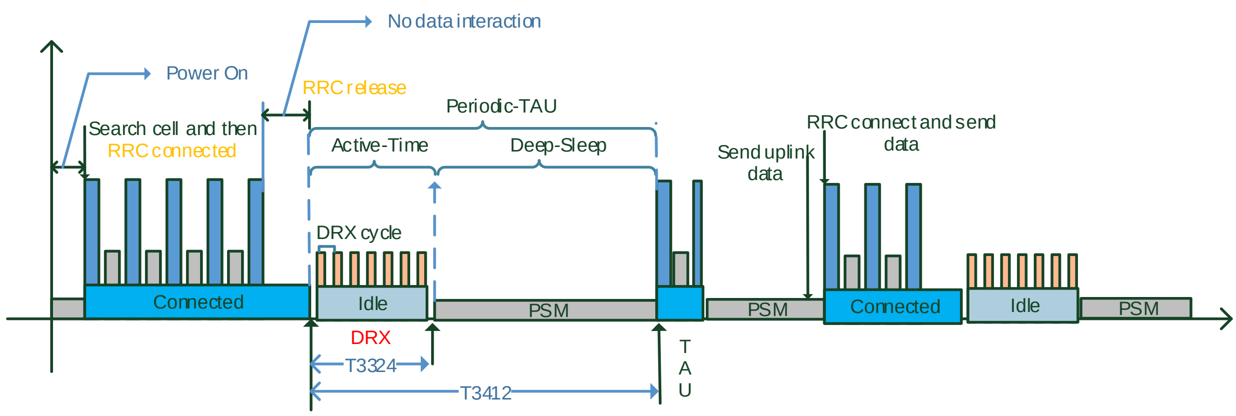

Figure 1: DRX Mode Illustration

NOTES

1. Typical DRX cycles are 1.28 s and 2.56 s.
2. DRX cycle is determined by network, and cannot be configured through the module.
3. DRX mode is suitable for applications with high real-time but low power-consumption requirements, such as intelligent street lighting systems.

2.2. eDRX

eDRX: extended DRX. It is an operating mode of Modem.

eDRX is a technology introduced in *3GPP Rel.13*. Compared with DRX, eDRX has a longer paging cycle, which allows the UE to further reduce power consumption but results in longer delay in downlink data receiving. The module only monitors NPDCCH channel and supports downlink transmission during PTW in eDRX mode. In other time durations, the module is in sleep mode, and does not monitor paging channels or receive downlink traffic.

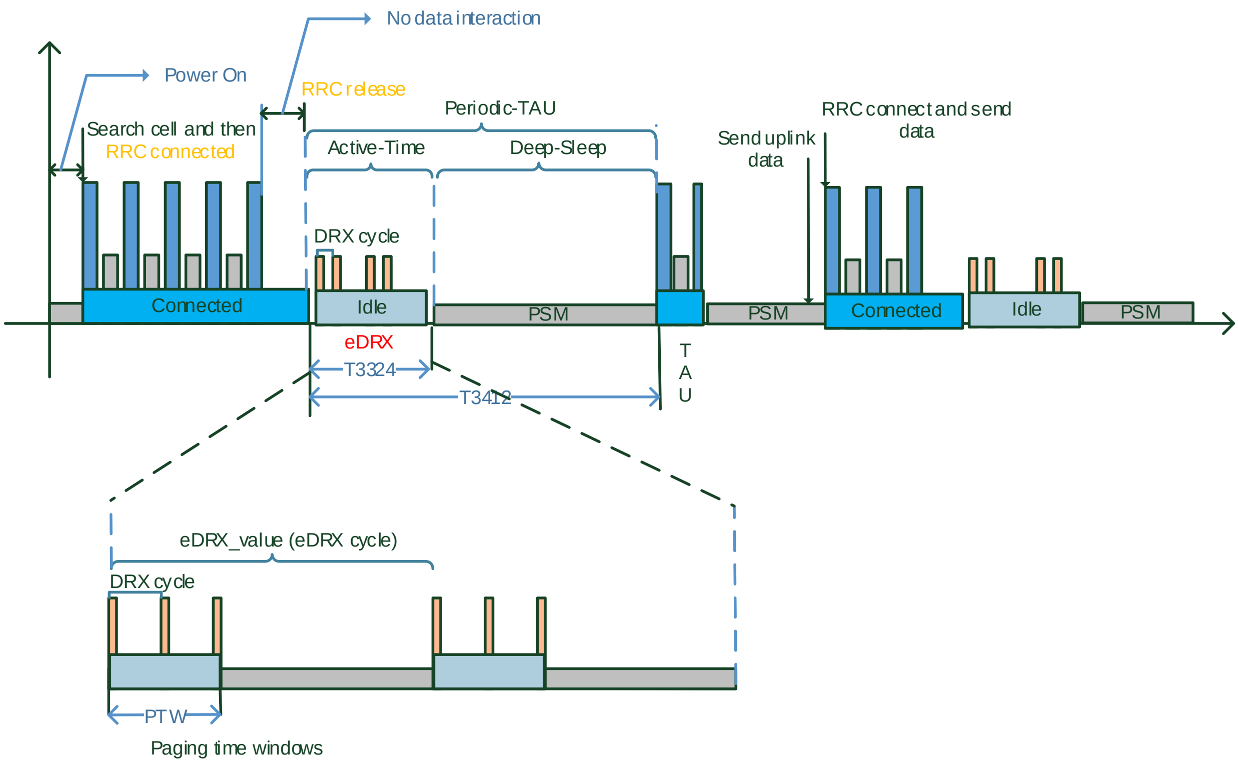

Figure 2: eDRX Mode Illustration

NOTES

1. Typical eDRX cycles are 20.48 s and 81.92 s.
2. BC66 and BC66-NA QuecOpen modules support eDRX cycle configuration with **AT+CEDRXS**, but the network determines whether the configuration will take effect. For more details of this command, refer to ***document [1]***.
3. eDRX mode is suitable for applications with high requirements in low power consumption but low requirements in real-timeliness, such as smart metering.

2.3. PSM

PSM: Power Saving Mode. It is an operating mode of Modem.

PSM is a technology introduced in *3GPP Rel.12*. When PSM is enabled, the module is disabled in RF signal receiving and transmitting as well as access stratum activities after T3324 times out, which greatly reduces power consumption. In PSM, the module cannot be paged and stops access stratum activities such as cell reselection, which indicates the module is unreachable in downlink.

In PSM, the module stops monitoring NPDCCH channel but remains registered on the network, which allows the module to send uplink data or signaling without performing ATTACH procedure.

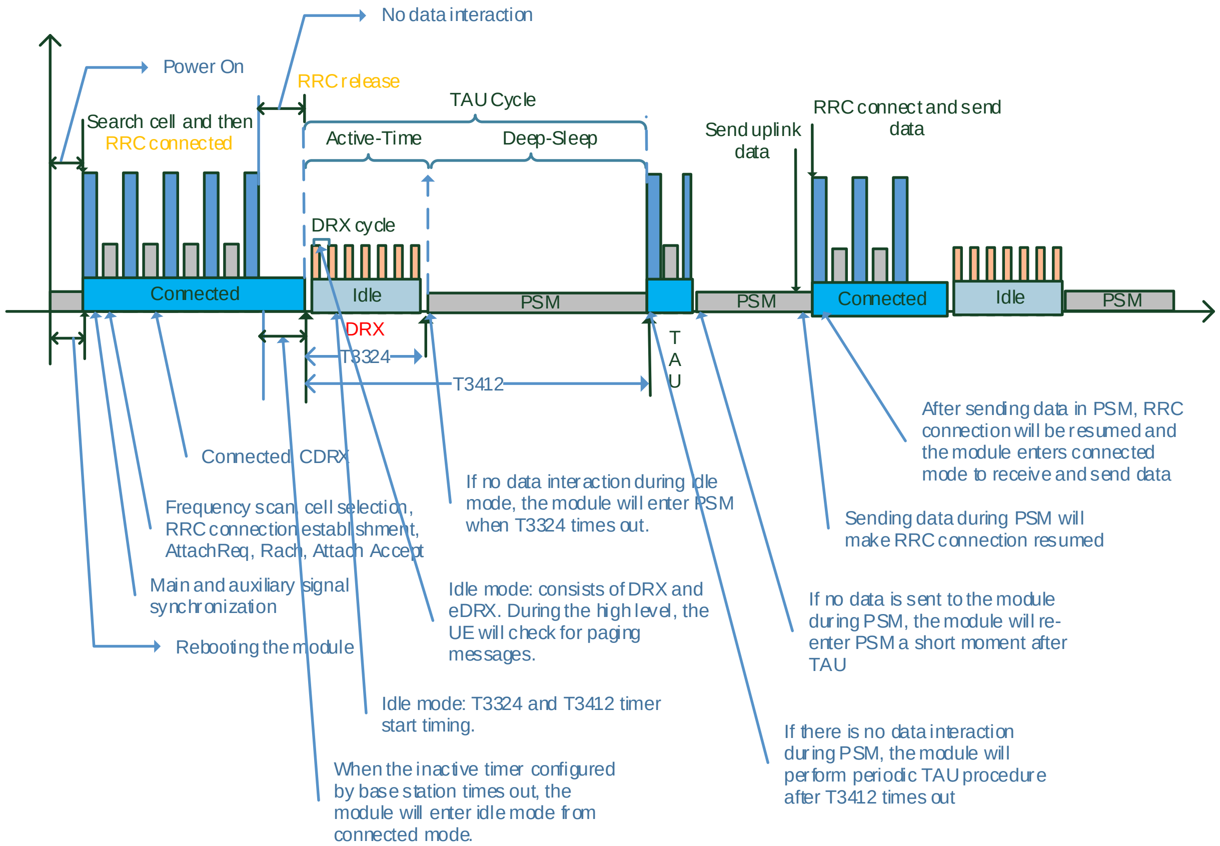

Figure 3: PSM Mode Illustration

NOTES

1. PSM can be enabled/disabled with **AT+CPSMS**. For more details of this command, refer to ***document [1]***.
2. The module provides lower power consumption in PSM than that in eDRX.

2.4. Idle (AP Operating Mode)

When all current tasks of the module are suspended, the AP enters idle mode to stop processing any service.

NOTES

1. The AP is enabled to enter idle mode automatically by default.
2. *Ql_SleepDisable* disables the AP from entering idle mode, and *Ql_SleepEnable* enables entering idle mode.
3. When the AP is disabled from entering idle mode by *Ql_SleepDisable*, the module cannot enter Light Sleep mode or Deep Sleep mode, but this does not affect tasks' running.

2.5. Deep Sleep

All "low power mode" introduced in this document means "Deep Sleep" mode in which the module provides minimized power consumption. In Deep Sleep mode, the CPU is powered off, all programs stop running, uplink/downlink data cannot be processed, serial ports are inaccessible, and only RTC keeps running (this is a similar situation as module power-off).

Generally speaking, when the following two conditions are met simultaneously, the module enters Deep Sleep mode automatically:

- Modem enters PSM
- AP enters idle mode, that is, all tasks are suspended

There are some exceptions as illustrated below, and in these special conditions, the module also enters Deep Sleep mode no matter in which mode the Modem/AP is.

- Executed **AT+CFUN=0** when USIM card is present
- When eDRX cycle is longer than 81.92 s
- Entered OOS status during network searching

The following circumstances may disable the module from entering Deep Sleep mode:

- PSM is disabled
- **AT+QSCLK=0** or **AT+QSCLK=2** is executed
- The AP is not in idle mode (i.e. when there are frequent interrupts, timers, etc. to be handled)
- TCP/MQTT connections are not disconnected timely after use
- PWRKEY is kept at low level all the time
- USB port keeps connected

- USIM card is absent

NOTES

1. When the Modem is disabled from entering PSM with **AT+CPSMS=0**, or the AP is disabled from entering idle mode, the module cannot enter Deep Sleep mode.
2. Distinguished feature of the module when it enters Deep Sleep mode is: serial ports are inaccessible and the VDD_EXT voltage is 0 V.
3. All data not saved to NVRAM will be lost when the module enters Deep Sleep mode. To avoid this, *Ql_Flash_Write* can be used to back up the important data before entering Deep Sleep mode and *Ql_Flash_Read* can be used to read the backup data subsequently. After the module wakes up, the programs will be reloaded and the tasks will be resumed.
4. After the module wakes up from Deep Sleep mode, it will reload the programs and resume the tasks.

3 Enter Deep Sleep Mode

The module is enabled to enter Deep Sleep mode automatically by default. If the Modem is disabled to enter PSM or the AP is disabled from entering idle mode, the following methods can enable the module to enter Deep Sleep mode again. If no modification is made to the default configuration, ignore this chapter directly.

3.1. AT+CPSMS

AT+CPSMS=1 can be used to enable entering PSM. In addition, **AT+CPSMS** can configure T3324 and T3412 values. For more details, refer to ***document [1]***.

NOTES

1. The validity of T3324 and T3412 values is determined by the network operators.
2. Due to different operator mechanisms in different regions, after configuring T3324 and T3412 values, first use **AT+CEREG=5** and then **AT+CEREG?** to query whether T3324/T3412 value is configured successfully. For more details about **AT+CEREG**, refer to ***document [1]***.

3.2. Ql_SleepEnable()

Ql_SleepEnable() can be used to enable the AP entering idle mode (enabled by default). The API description is provided below:

```
/*****************************************************************
* Function:          Ql_SleepEnable
*
* Description:
*                         Enable the module to enter Deep Sleep mode.
*
* Return:
*                       QL_RET_OK indicates this function successes.
*                       Ql_RET_NOT_SUPPORT indicates this function is not supported.
*****************************************************************/
  s32 Ql_SleepEnable(void);
```

NOTE

Ql_SleepEnable() only enables the AP to enter idle mode. The module will enter Deep Sleep mode after all tasks are suspended and the Modem enters PSM.

3.3. Ql_DeepSleep_Register()

In QuecOpen solution, the modules support Deep Sleep status indication. *Ql_DeepSleep_Register* can be called to register the callback function for Deep Sleep status indication. When the module is about to enter Deep Sleep mode, the callback function is triggered and you can print related log to indicate that the module enters Deep Sleep mode. This callback function in QuecOpen solution works as the URC **ENTER DEEPSLEEP** of the modules in standard solution, serving to indicate that the module has entered Deep Sleep mode.

Ql_DeepSleep_Register is defined as below:

```
/*****************************************************************
* Function:       Ql_DeepSleep_Register
*
* Description:
*                 This function registers a callback handler of Deep Sleep mode
*                 The callback is called when the module enters Deep Sleep mode.
*
* Parameters:
*                 callback_ds_event:
*                     [out] Notify the application when the module enters Deep Sleep.
*                 param:
*                     [in] User data.
*
* Return:
*                 QL_RET_OK indicates register ok;
*                 QL_RET_ERR_PARAM indicates the param error.
*****************************************************************/
typedef void (*Callback_DeepSleep_Func)(void *param);
s32 Ql_DeepSleep_Register(Callback_DeepSleep_Func callback_ds_event,void * param);
```

4 Disable Deep Sleep Mode

The module is enabled to enter Deep Sleep mode automatically by default. Use any of the following methods to disable the module from entering Deep Sleep mode.

4.1. AT+CPSMS

AT+CPSMS=0 can be used to disable the Modem from entering PSM, thus disabling Deep Sleep mode. For more details of this command, refer to ***document [1]***.

It is recommended to use **AT+CPSMS=0** to disable entering PSM for applications with high requirements in real-timeliness, low requirements in power consumption and with a demand for downlink data receiving.

4.2. Ql_SleepDisable()

Ql_SleepDisable() can be used to disable the AP from entering idle mode, thus disabling Deep Sleep mode. The API description is provided below:

```
/*****************************************************************
* Function:       Ql_SleepDisable
* Description:
*                 Disable the module from entering sleep modes.
* Return:
*                 QL_RET_OK indicates this function successes.
*                 Ql_RET_NOT_SUPPORT this function is not supported.
*****************************************************************/
 s32 Ql_SleepDisable(void);
```

NOTE

Ql_SleepDisable() also disables the module from entering Light Sleep mode.

5 Power-on and Wake-up Reasons

5.1. Power-on Reasons

Ql_GetPowerOnReason can be called to obtain the reason why the module is powered on. When it is called in the application, its return value explains the reason why the module is powered on.

QuecOpen solution provides the following 4 power-on reasons. Refer to the definition of *Enum_Ql_Power_On_Result_t* for details.

- QL_FIRST_BOOT_UP First power-on and is powered on by PWRKEY
- QL_DEEP_SLEEP Wakes up from Deep Sleep status
- QL_SYS_RESET Reboots by a command or the reset button
- QL_FORCED_SHUT_DOWN Powered on by PWRKEY after powered off with a command

```
typedef enum {
    QL_FIRST_BOOT_UP            = 0,    /* first boot up*/
    QL_DEEP_SLEEP               = 1,    /* wakeup from deepsleep */
    QL_SYS_RESET                = 2,    /* sys_reset */
    QL_FORCED_SHUT_DOWN         = 3,    /* forced shut down */
    QL_POWER_ON_RESULT_MAX = 3
} Enum_Ql_Power_On_Result_t;
```

Ql_GetPowerOnReason is defined as follows:

```
/*****************************************************************
* Function:       Ql_GetPowerOnReason
*
* Description:
*                 This function gets the reason for power on.
*
* Parameters:
*                 None
*
* Return:
*                 The reason of the Power on, one value of Enum_Ql_Power_On_Result_t.
*****************************************************************/
s32 Ql_GetPowerOnReason(void);
```

5.2. Wake-up Reasons

After the module wakes up from Deep Sleep, *Ql_GetWakeUpReason* can be called to obtain the reason.

QuecOpen solution provides the following 4 wake-up reasons. Refer to the definition of *Enum_Ql_Wake_Up_Result_t* for details.

- Ql_RTC_TC_WAKEUP Not supported currently
- Ql_PSM_EINT_WAKEUP Wakes up by the external interrupt pin PSM_EINT
- Ql_RTC_AlARM_WAKEUP Wakes up by TAU or RTC timer timing out
- Ql_POWER_KEY_WAKEUP Wakes up by PWRKEY pin (Not Recommended)

```
typedef enum {
    Ql_RESULT_NOT_WAKEUP    = 0,        /* power on. Not wakeup*/
    Ql_RTC_TC_WAKEUP        = 1,        /* rtc tc wakeup*/
    Ql_PSM_EINT_WAKEUP      = 2,        /* psm_eint wakeup*/
    Ql_RTC_AlARM_WAKEUP     = 3,        /* rtc timer wakeup*/
    Ql_POWER_KEY_WAKEUP     = 4,        /* power_key wakeup*/
    Ql_WAKE_UP_RESULT_MAX   = 4
} Enum_Ql_Wake_Up_Result_t;
```

Ql_GetWakeUpReason is defined as follows:

```
/*****************************************************************
* Function:     Ql_GetWakeUpReason
*
* Description:
*               This function gets a reason for wake up the module.
*
* Parameters:
*               None
*
* Return:
*               The reason of the wake up, one value of Enum_Ql_Wake_Up_Result_t.
*****************************************************************/
Enum_Ql_Wake_Up_Result_t Ql_GetWakeUpReason(void);
```

5.2.1. Ways of wake up from Deep Sleep

5.2.1.1. Wake Up by TAU

- Wake-up Solution Description

After the module enters Deep Sleep mode and waits until the timer T3412 times out, a TAU (Tracking Area Update) will be triggered. Then the module will be woken up, AP tasks start to run and the Modem exits from PSM.

- Recommended Application Scenarios

Users can configure different T3412 values (if supported by the network operator) to realize module wake-up from Deep Sleep mode, and this solution is recommended for applications with low real-time requirements, such as smart metering.

5.2.1.2. Wake Up by PSM_EINT Pin

- Wake-up Solution Description

The module provides an external interrupt pin PSM_EINT. Pulling down PSM_EINT (falling edge) will wake the module up from Deep Sleep.

- Recommended Application Scenarios

This solution is recommended for applications with an external MCU or external buttons/sensors, which are convenient to provide PSM_EINT a falling edge.

5.2.1.3. Wake Up by RTC Timer

- Wake-up Solution Description

Users can set a RTC timer to wake the module up from Deep Sleep. When the RTC timer times out, the module will be woken up.

- Recommended Application Scenarios

Initiating an RTC timer before the module enters Deep Sleep mode. If the timeout value is 10 minutes, then the module will be woken up every 10 minutes. Every time when the module is woken up, it will handle related services, and when the services are finished the module will enter Deep Sleep mode again for a period of timeout value.

NOTES

1. PSM_EINT falling edge solution and RTC timer solution will only wake up the AP. It is necessary to send uplink data to wake up the Modem from PSM.
2. In hardware wake-up solutions, PWRKEY is NOT recommended to be used to wake up the module but to power on the module.

6 Power Consumption

Table 1: Current Consumption of BC66 QuecOpen Module (3.3V VBAT Powered)

Deep Sleep						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.**	**Unit**
Idle	PSM			3.5		μA
Light Sleep						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.**	**Unit**
Idle	@ eDRX = 81.92 s, PTW = 40.96 s			288		μA
	@ DRX = 1.28 s			541		μA
	@ DRX = 2.56 s			434		μA
Active [1]						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.** [2]	**Unit**
Normal	Connected @ Single-tone (15 kHz subcarrier spacing)	B1 @ 23 dBm		100	285	mA
		B2 @ 23 dBm		103	294	mA
		B3 @ 23 dBm		107	308	mA
		B4 @ 23 dBm		107	307	mA
		B5 @ 23 dBm		107	303	mA
		B8 @ 23 dBm		113	325	mA
		B12 @ 23 dBm		134	393	mA
		B13 @ 23 dBm		111	319	mA
		B17 @ 23 dBm		133	392	mA
		B18 @ 23 dBm		110	316	mA

		B19 @ 23 dBm	109	311	mA
		B20 @ 23 dBm	109	301	mA
		B25 @ 23 dBm	103	293	mA
		B26* @ 23 dBm	TBD	TBD	mA
		B28 @ 23 dBm	128	375	mA
		B66 @ 23 dBm	109	312	mA
	Connected @ Single-tone (3.75 kHz subcarrier spacing)	B1 @ 23 dBm	193	302	mA
		B2 @ 23 dBm	187	296	mA
		B3 @ 23 dBm	215	335	mA
		B4 @ 23 dBm	237	311	mA
		B5 @ 23 dBm	215	330	mA
		B8 @ 23 dBm	224	344	mA
		B12 @ 23 dBm	250	395	mA
		B13 @ 23 dBm	203	316	mA
		B17 @ 23 dBm	258	409	mA
		B18 @ 23 dBm	198	313	mA
		B19 @ 23 dBm	198	314	mA
		B20 @ 23 dBm	215	329	mA
		B25 @ 23 dBm	187	297	mA
		B26* @ 23 dBm	200	313	mA
		B28 @ 23 dBm	250	398	mA
		B66 @ 23 dBm	200	316	mA

Table 2: Current Consumption of BC66-NA QuecOpen Module (3.3V VBAT Power Supply)

Power OFF (AT+QPOWD=0)						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.**	**Unit**
/	/			2.7		µA
Deep Sleep						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.**	**Unit**
Idle	PSM			3.5		µA
Light Sleep						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.**	**Unit**
Idle	@ eDRX = 81.92 s, PTW = 40.96 s			130		µA
	@ DRX = 1.28 s			520		µA
	@ DRX = 2.56 s			250		µA
Active [1]						
AP Mode	**Modem Mode**		**Min.**	**Typ.**	**Max.** [2]	**Unit**
Normal	Connected @ Single-tone (15kHz subcarrier spacing)	B1 @ 23 dBm		95	265	mA
		B2 @ 23 dBm		95	265	mA
		B3 @ 23 dBm		100	280	mA
		B4 @ 23 dBm		100	280	mA
		B5 @ 23 dBm		95	270	mA
		B8 @ 23 dBm		100	290	mA
		B12 @ 23 dBm		85	240	mA
		B13 @ 23 dBm		90	250	mA
		B17 @ 23 dBm		85	240	mA
		B18 @ 23 dBm		95	270	mA
		B19 @ 23 dBm		95	270	mA

		B20 @ 23 dBm	95	270	mA
		B25 @ 23 dBm	95	270	mA
		B26* @ 23 dBm	TBD	TBD	mA
		B28 @ 23 dBm	85	240	mA
		B66 @ 23 dBm	100	280	mA
		B71 @ 23 dBm	85	240	mA
		B85 @ 23 dBm	85	235	mA
Normal	Connected @ Single-tone (3.75kHz subcarrier spacing)	B1 @ 23 dBm	210	270	mA
		B2 @ 23 dBm	205	270	mA
		B3 @ 23 dBm	210	280	mA
		B4 @ 23 dBm	215	280	mA
		B5 @ 23 dBm	200	280	mA
		B8 @ 23 dBm	220	290	mA
		B12 @ 23 dBm	180	230	mA
		B13 @ 23 dBm	190	250	mA
		B17 @ 23 dBm	180	240	mA
		B18 @ 23 dBm	195	260	mA
		B19 @ 23 dBm	205	270	mA
		B20 @ 23 dBm	205	270	mA
		B25 @ 23 dBm	205	270	mA
		B26* @ 23 dBm	TBD	TBD	mA
		B28 @ 23 dBm	185	240	mA
		B66 @ 23 dBm	215	280	mA
		B71 @ 23 dBm	180	240	mA
		B85 @ 23 dBm	180	240	mA

NOTES

1. [1)] Power consumption under instrument test condition.
2. [2)] The “maximum value” in “Active” mode refers to the maximum pulse current during RF emission.
3. “*” means under development.
4. The current consumption values are excerpted from the module hardware design manuals. Refer to ***document [2]*** and ***document [3]*** for the latest values (if available).

7 FAQs and Important Notes

7.1. Wake Up from Light Sleep Mode

After the module enters Light Sleep mode, the UART should be woken up first before sending any data, otherwise the first packet of data will be lost. Users can wake up the module by sending a packet of data with two bytes (such as **AT**) via main UART first before sending any data.

7.2. How to Enter Deep Sleep Quickly

Frequent interrupts or timers make the AP involved in task handling all the time, and thus extend the time for AP to enter idle mode. If the module has to enter Deep Sleep mode quickly, disable the frequent interrupts and/or timers.

When the AP tasks are idle and the module has registered on network normally, the module enters Deep Sleep quickly after executing **AT+CFUN=0**.

7.3. Cannot Enter Deep Sleep Mode in 60 s after AT+CFUN=0

If the module has registered on network successfully, **OK** will be returned immediately after executing **AT+CFUN=0**, and then the module enters Deep Sleep mode. However, if the module is not registered on network successfully, **OK** will not be returned until the RRC connection expires (60 s to 80 s), and after that, the module enters Deep Sleep mode.

7.4. Purpose of Ql_OS_GetMessage

Ql_OS_GetMessage retrieves a message from the current task's message queue. When there is no message to be handled, the task will be suspended, which allows the AP to enter idle mode immediately.

7.5. How to Check Whether Modem is in PSM

- Execute **AT+QNBIOTEVENT=1,1** to enable the indication of PSM state through URC. When the Modem enters PSM, URC **\r\n+QNBIOTEVENT: "ENTER PSM"** will be reported.

- Get and review the GKI log to check whether the Modem has entered PSM. The following log indicates the Modem has entered PSM.

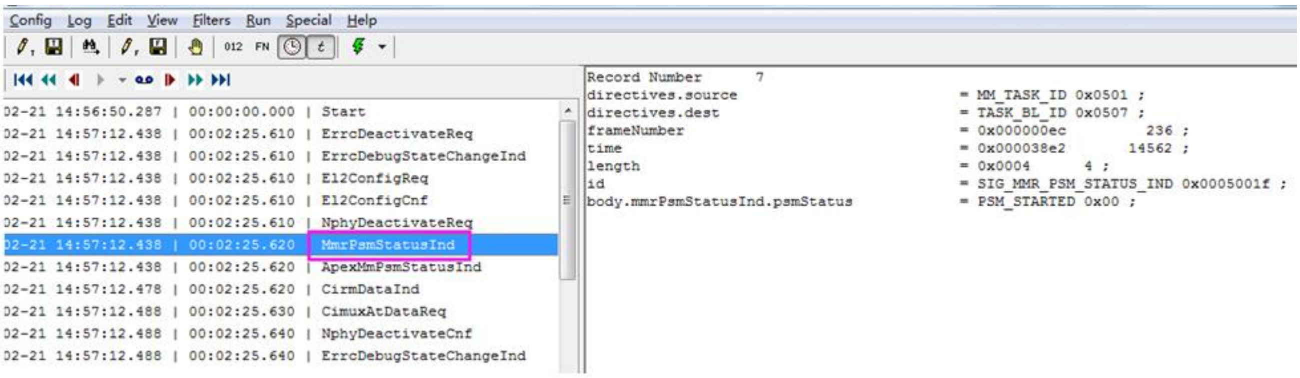

Figure 4: Modem Has Entered PSM

7.6. Can Enter Deep Sleep Only After TCP Connection Disconnected

When establishing a TCP connection, the module will be disabled from entering Deep Sleep mode. This is the reason why only after TCP connection is disconnected, the module can enter Deep Sleep mode.

But TCP connection establishment will not influence the module entering Light Sleep mode and the Modem entering PSM.

NOTE

MQTT connection is established over TCP, so the module cannot enter Deep Sleep mode either before MQTT connection is disconnected.

7.7. Capture Gennie Log via USB Port

When Gennie tool is connected to USB port for log capturing, the module will not be able to enter Deep Sleep mode because the USB port is always connected and in working mode.

7.8. Cold Start vs Wake-Up from Deep Sleep Mode

In cold start, the module will first initialize its PMU and then enters BootROM stage. In BootROM, the module will check whether there is a need for firmware/App bin download. If yes, the module will start firmware download, otherwise it enters BootLoader. During BootLoader, the module either performs DFOTA or enters FreeRTOS subsequently. In FreeRTOS, the module will initialize AP and Modem and starts to run tasks.

Wake-up from Deep Sleep shares almost the same procedures with cold start but there are two major differences as listed below:

- Wake-up from Deep Sleep mode does not need to check whether there is a need for firmware/App bin download.
- After entering FreeRTOS, the time needed for AP and Modem initialization is greatly shortened as the module is able to obtain the network information for previous network-connection via RTC RAM.

7.9. How to Release RRC Connection Quickly

This can be configured through **AT+QNBIOTRAI=<rai>**. For more information on this command, refer to ***document [1]***.

In current NB-IoT network, RRC connection is usually initiated by network, but the module can notify the network that UE does not have any data to transmit with the above-mentioned command. And then the network will determine whether to release RRC connection immediately or not.

Available values of **<rai>** include:

0 No information available (or none of the other options apply) (default)
1 TE will send only 1 UL packet and no DL packets expected
2 TE will send only 1 UL packet and only 1 DL packet expected

If RRC connection is intended to be released quickly, specify **<rai>** as 1 or 2:

1 The UE sends an uplink packet without expecting any downlink packet, and thus the RRC connection will be released immediately after uplink packet is sent. A typical example is UDP application.
2 The UE sends an uplink packet and also expects a downlink ACK packet, and thus the RRC connection will be released immediately after receiving the downlink packet. A typical example is PING command application. If **<rai>** is specified as 1 during PING, RRC connection will also be released after sending uplink packet. But as the network has to send the ACK packet, the UE will be paged again to re-establish the RRC connection.

7.10. How to Enter Deep Sleep Quickly after AT Command Sending

By default, the module will start a Sleep Lock timer (10 s by default) to prevent the module from entering Deep Sleep mode after the module sends any AT command through RIL APIs.

AT+QRELLOCK can be used to release the Sleep Lock after AT command sending, thus allowing the module to enter sleep mode immediately when there are no other sleep handles available to control the sleep state of the system. For more information on the command, refer to ***document [1]***.

7.11. Usage of PSM_EINT Callback Function

Ql_Psm_Eint_Register can be called to register PSM_EINT event notification. When the module is woken up by the external interrupt pin PSM_EINT, the application will call the callback function. Be noted that QL type functions cannot be called in the callback function, otherwise it may result in module crash or reboot.

7.12. Usage of RTC_Timer Callback Function

Ql_Rtc_RegisterFast can be called to register an RTC timer callback function. When the module is woken up due to RTC timer times out, the application will call the callback function. Be noted that among all FreeRTOS related APIs, only the ISR type functions can be called in RTC timer callback functions, otherwise it may result in module crash or reboot.

8 Appendix

8.1. Related Documents

Table 3: Related Documents

SN	Document Name	Remark
[1]	Quectel_BC66&BC66-NA_AT_Commands_Manual	BC66 and BC66-NA AT commands manual
[2]	Quectel_BC66-QuecOpen_Hardware_Design	BC66 QuecOpen hardware design
[3]	Quectel_BC66-NA-QuecOpen_Hardware_Design	BC66-NA QuecOpen hardware design

8.2. Terms and Abbreviations

Table 4: Terms and Abbreviations

Abbreviation	Description
AP	Application Processor
App	Application
Modem	Modulator-Demodulator
MQTT	Message Queuing Telemetry Transport
AS	Access Stratum
CPU	Central Processing Unit
DFOTA	Delta Firmware upgrade Over-The-Air
DL	Downlink

DRX	Discontinuous Reception
eDRX	extended Discontinuous Reception
IoT	Internet of Things
MCU	Microcontroller Unit
NPDCCH	Narrowband Physical Downlink Control Channel
NVRAM	Non-Volatile Random Access Memory
OOS	Out-of-Service
PDN	Public Data Network
PSM	Power Saving Mode
PTW	Paging Time Window
RAM	Random Access Memory
RF	Radio Frequency
RRC	Radio Resource Control
RTC	Real Time Clock
TAU	Tracking Area Update
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
URC	Unsolicited Result Code
USB	Universal Serial Bus
USIM	Universal Subscriber Identification Module