Build a Smarter World

EC2X&AG35-QuecOpen Low Power Consumption Solution

LTE Module Series

Rev. EC2X&AG35-QuecOpen_ Low_Power_Consumption_Solution

Date: 2018-09-02

Status: Preliminary

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.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

For technical support, or to report documentation errors, please visit:
http://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.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2017-11-30	Gale GAO	Initial
1.1	2018-01-30	Gale GAO	Add the selection of driver(chapter 5)
1.2	2018-05-02	Gale GAO	Add the reasons of module canot enter sleep mode(chapter 7) Add the wakeup reasons(chapter 8) Add the indicator of sleep electricity consumption(chapter 9)
2.0	2018-05-15	Gale GAO	Update low power consume driver to V2.0, compile to be a kmod into rootfs by default, insmod it directly.
3.0	2018-09-02	Gale GAO	Optimize QuecOpen LPM feature.

Contents

1 Summary of QuecOpen LPM

OuecOpen LPM low power and consumption solution applies to the product with mcu externally, typical is the vehicle T-Box.

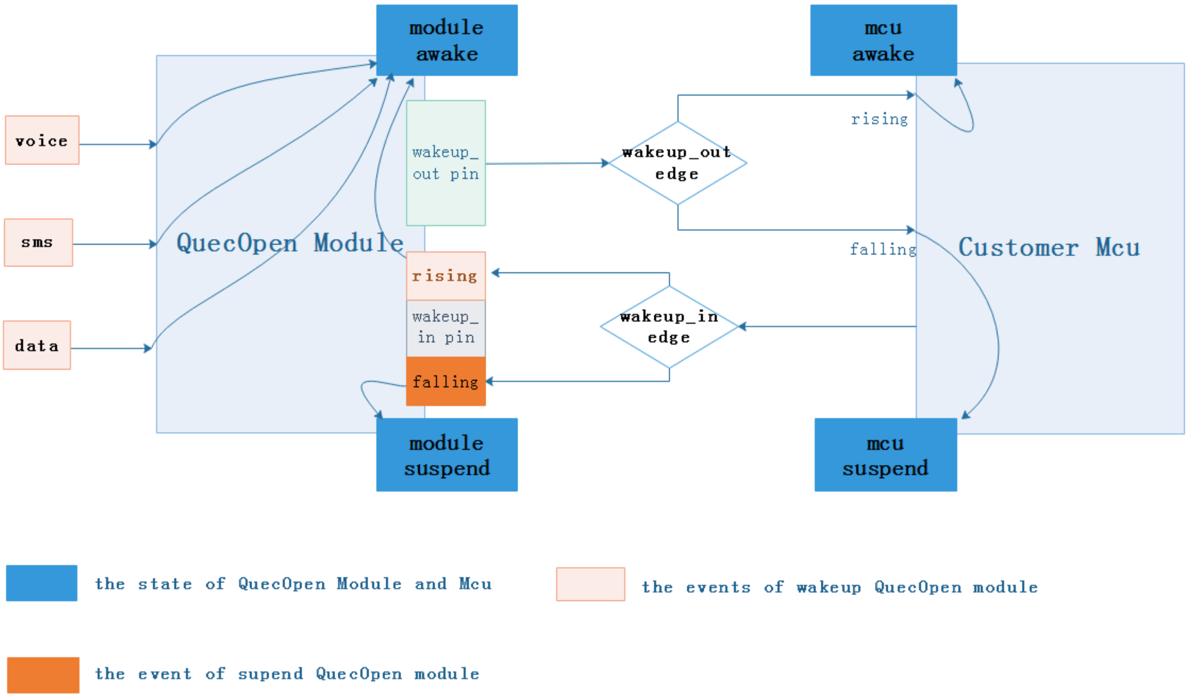
QuecOpen low power and consumption solution is developed by following Linux technology:

1. Autosleep: After enabling autosleep, there are tendency that freezing process, suspend peripheral and cpu go sleeping mode.
2. Wakelock: If kernel or any process hold one or several wakelock, module sleep will be restrained.

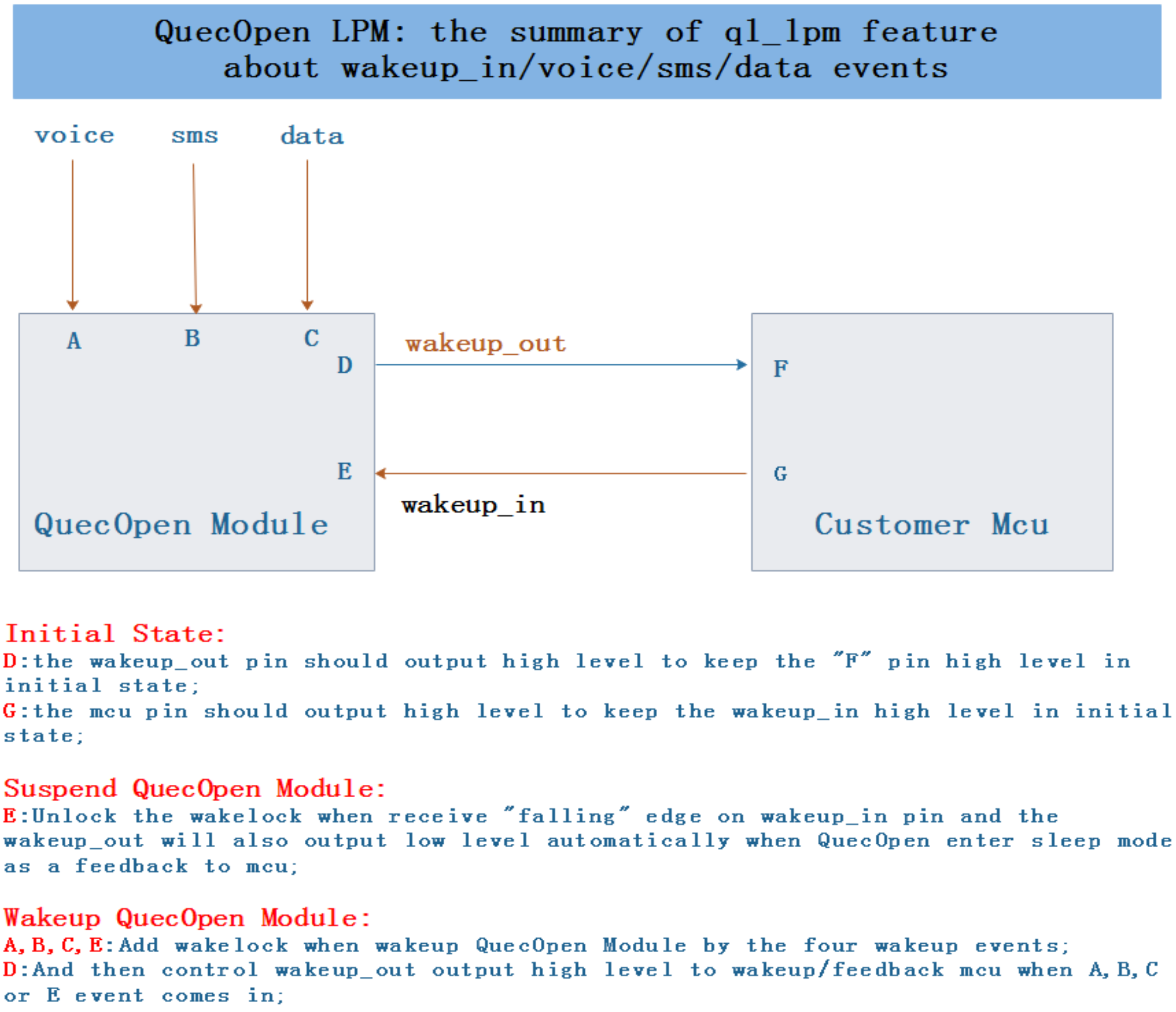
Autosleep and wakelock form a contradictory community to realise wakeup and sleep feature.

2 QuecOpen LPM State Introduction

2.1. QuecOpen LPM State



2.2. QuecOpen LPM Solution Description


QuecOpen LPM: the summary of ql_lpm feature about wakeup_in/voice/sms/data events
voice
sms
data
A
B
C
D
E
QuecOpen Module
wakeup_out
wakeup_in
F
G
Customer Mcu

Initial State:
D:the wakeup_out pin should output high level to keep the "F" pin high level in initial state;
G:the mcu pin should output high level to keep the wakeup_in high level in initial state;

Suspend QuecOpen Module:
E:Unlock the wakelock when receive "falling" edge on wakeup_in pin and the wakeup_out will also output low level automatically when QuecOpen enter sleep mode as a feedback to mcu;

Wakeup QuecOpen Module:
A,B,C,E:Add wakelock when wakeup QuecOpen Module by the four wakeup events;
D:And then control wakeup_out output high level to wakeup/feedback mcu when A,B,C or E event comes in;

Initial State:
D: The wakeup_out pin should output high level to keep the “F” pin high level in initial state.
G: The MCU pin should output high level to keep the wakeup_in high level in initial state.

Suspend QuecOpen Module:
E: Unlock the wakelock when receive “falling” edge in wakeip_pin and the wakeup_out will also output low level automatically when QuecOpen enter sleep mode as a feedback to MCU.

Wakeup QuecOpen Module:
A, B, C, E: Add wakelock when wakeup QuecOpen Module by the four wakeup events;
D: And the control wakeup_out output high level to wakeup/feedback MCU when A, B, C or E event comes in.

3 QuecOpen LPM API

QuecOpen LPM provides the following application layer APIs to control autosleep/wakeup.

```
typedef void (*QL_Lpm_Handler_T) ( ql_lpm_edge_t lpm_edge);
```

Feature: Register callback function through QL_Lpm_Init, callback function will be triggered when wakeup_in level changes.
Parameter:
lpm_edge: The edge of wakeup_in pin reported by underlying changed.
Enumeration Value: E_QL_LPM_FALLING or E_QL_LPM_RISING, please refer to head file ql_lpm.h.

```
int QL_Lpm_Init(QL_Lpm_Handler_T ql_lpm_handler, QL_Lpm_Cfg_T *ql_lpm_cfg);
```

Feature: Initialize Quectel LPM feature, auto load driver ql_lpm, at the same time listen wakeup_in state change and inform ql_lpm_handler.
Parameter:
ql_lpm_handler: User callback, will be triggered when wakeup_in level changes.
ql_lpm_cfg: User parameter data structure, it recommended that incoming NULL and use Quectel default pin and trigger mode. Extended support for users to fill the optional pins and trigger mode.
Returned Value: 0-success, -1-failure.

```
int QL_Lpm_Deinit();
```

Feature: Cancel Quectel LPM feature, unload ql_lpm kernel, cancel handler.
Returned Value: 0-success, -1-failure.

```
int Ql_Autosleep_Enable(char enable);
```

Feature: Enable AutoSleep, then system will go sleep mode automatically when conditions are met.
Parameter:
1-enable autosleep. 0-disable autosleep. (The parameter 0 is generally not passed, if need to keep wakeup, please use the following interface Ql_SLP_WakeLock_Lock to lock.)
Returned Value: 0-success, -1-failure.

```
int Ql_SLP_WakeLock_Create(const char *name, size_t len);
```

Feature: Create wakelock, and the owner of the wakelock created by this interface is the current process. Locks can be created up to 512.
Parameter:
Name: The name of wakelock (shown in /sys/kernel/debug/wakeup_sources)
Len: the length of name and can be up to 28 characters.
Returned Value: The wakelock descriptor, otherwise returns -1.

int Ql_SLP_WakeLock_Lock(int fd)

Feature: To lock wakelock created by above interface, after that Linux side cannot enter sleep mode.

Parameter:

fd: wakelock descriptor

Returned Value: 0-success, -1-failure.

int Ql_SLP_WakeLock_Unlock(int fd)

Feature: Unlock specified wakelock.

Parameter:

fd: wakelock descriptor

Returned Value: 0-success, -1-failure.

int Ql_SLP_WakeLock_Destroy(int fd)

Feature: Destroy wakelock.

Parameter:

fd: wakelock descriptor

Returned Value: 0-success, -1-failure.

NOTE

After Ql_Autosleep_Enable enable Autosleep, it's recommended that keeping system wakeup through Ql_SLP_WakeLock_Lock rather than Ql_Autosleep_Enable(0), Ql_SLP_WakeLock_Unlock can release wakelock and give up sleep locking. Process will be frozen after system sleep, and continue running after system wakeup.

Example

External pin interruption.
ql-ol-extsdk/example/low_power_consume_app/example_lpm.c

Internal interruption.
ql-ol-extsdk/example/low_power_consume_app/example_lpm_all.c

4 Events Supported by QuecOpen LPM

4.1. Sleep Event

4.1.1. Pin "Falling" Edge Interrupt Events Trigger Module Sleep.

Pin definition

EC2X		AG35	
wakeup_in	Pin62	wakeup_in	Pin61
wakeup_out	Pin5	wakeup_out	Pin147

Wakeup_in: Module input pin, when bootup, mcu need control this pin to be high level. The "falling" edge trigger app to release wakelock to enter sleep mode.

4.2. Wakeup Event

4.2.1. Pin "Rising" Edge Interrupt Event

Pin definition

EC2X		AG35	
wakeup_in	Pin62	wakeup_in	Pin61
wakeup_out	Pin5	wakeup_out	Pin147

Wakeup_in: Module input pin, when bootup, mcu need control this pin to be high level. When receiving "falling" edge module enter sleep mode, when receiving "rising" edge module wakeup.
Wakeup_out: Module output pin, after enabling ql_lpm, output high-level by default. Output low-level after module entering sleep mode, output high-level after module wakeup.

Example

```
ql-ol-extsdk/example/low_power_consume_app/example_lpm.c
```

4.2.2. Call, SMS and Data Wakeup Event

Call QuecOpen api in advance to register the corresponding callback, waiting for event triggered to wakeup module.

Example

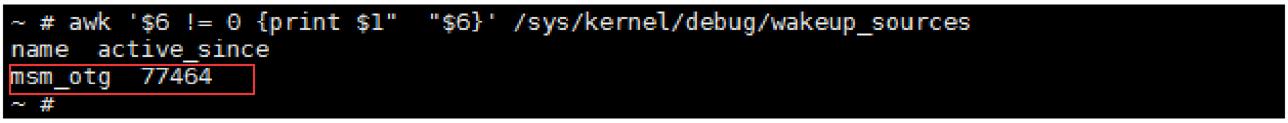
ql-ol-extsdk/example/low_power_consume_app/example_lpm_all.c

5 The Debug that Module Cannot Sleep

If users already released wakelock in app, but the module still cannot sleep, please query the wakelock which current system hold via below command

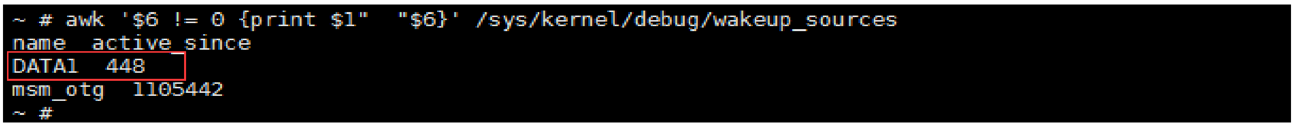
```
awk '$6 != 0 {print $1"    "$6}' /sys/kernel/debug/wakeup_sources
```

(1) If show msm_otg, it indicates that usb_vbus is in high-level state, please pull it low to go sleep mode.



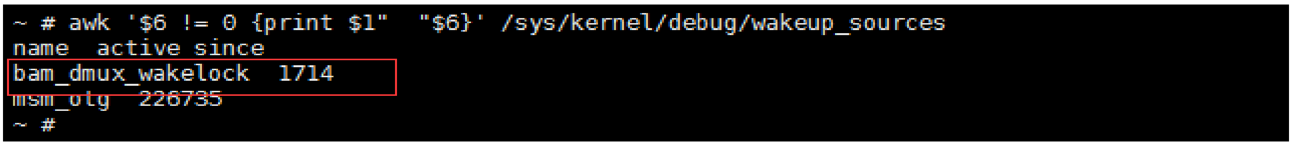
```
~ # awk '$6 != 0 {print $1"  "$6}' /sys/kernel/debug/wakeup_sources
name  active_since
msm_otg  77464
~ #
```

(2) If show DATA1, it indicates GPS data is preventing sleep, before sleep need to call QL_LOC_Stop_Navigation() to turn off GPS.



```
~ # awk '$6 != 0 {print $1"  "$6}' /sys/kernel/debug/wakeup_sources
name  active_since
DATA1  448
msm_otg  1105442
~ #
```

(3) If show bam_dmux_wakelock, it indicates there's data interaction on APN port, please close it.



```
~ # awk '$6 != 0 {print $1"  "$6}' /sys/kernel/debug/wakeup_sources
name  active_since
bam_dmux_wakelock  1714
msm_otg  226735
~ #
```

(4) If WiFi feature is used, before sleep please call ql_wifi_disable() interface to turn off WiFi.

(5) If Ethernet feature is used, before sleep please call ql_sgmii_disable() interface to turn off Ethernet.

6 Troubleshooting of Unexpected Wakeup

Users can through consumption capture diagram of power consumption catcher to check whether there's unexpected wakeup, also can use debug uart to capture log. The steps are as following.

(1) Execute below steps.

```
~ # echo 1 > /sys/module/printk/parameters/perf_mode_console
~ # echo 1 > /sys/module/msm_show_resume_irq/parameters/debug_mask
~ # echo 0x2 > /sys/module/ipc_router_core/parameters/debug_mask
```

(2) Then let module go sleep, check serial port log. If there is log like following, the module has woken-up.

```
[   113.386694] gic_show_resume_irq: 57 triggered qcom,smd-modem
[   113.386694] gic_show_resume_irq: 200 triggered qcom,smd-rpm
[   113.386694] resume cycles:           2542257600
[     113.388512]   [IPCRTR]     CLI   RX   Len:0xd   T:0x1   CF:0x0   SVC:<0x3:0x1>   SRC:<0x3:0x11>
DST:<0x1:0x43> DATA: 51000b04 13000600
[   113.388520] PM: noirq resume of devices complete after 0.975 msecs
[   113.389994] PM: early resume of devices complete after 1.088 msecs
```

(3) Analysis of wakeup log

1) gic_show_resume_irq: 57 triggered qcom,smd-modem: irg 57 indicates that modem send QMI message to AP side via smd channel. (If that is 58, indicates there is ip message sent to AP side.)

2) [IPCRTR] CLI RX Len:0xd T:0x1 CF:0x0 SVC:<0x3:0x1> SRC:<0x3:0x11> DST:<0x1:0x43> DATA: 51000b04 13000600:

CLI RX: QMI client received message, maybe response or indication, distinguished via DATA field.

SVC: <0x3,0x1>: 0x3 indicates QMI msg id: NAS.

DATA: 51000b04 13000600: This field reversed order is 13000600 51000b04, 0x04 indicates indication, 0x0051 indicates QMI_NAS_SIG_INFO_IND according to Chapter 8.1, that is, report QMI msgs that changed state due to changes in signal strength.

Then judge whether it was an unexpected wakeup and deal with it.

7 Consumption Index

7.1. Consumption Index

The module is powered by 3.8V.

Sleep Consumption:

Generally sleep consumption is about 4mA;

After AT+CFUN=0 turn off RF, sleep consumption is about 1.x mA;

Idle Consumption:

Consumption is about 25mA;

Data transmission or voice communication:

Consumption is generally a few hundred milliamperes, different under different standards.

Please refer to Table 1 or the hardware manual.

Parameter	Description	Condition	Typical Value	Unit
IVBAT	Shutdown mode	When the module is shut down	12	uA
	Sleep Mode	**AT+CFUN=0** (USB disconnect)	1.11	mA
		EGSM @DRX=2 (USB disconnect)	2.21	mA
		EGSM @DRX=5 (USB disconnect)	1.67	mA
		EGSM @DRX=5 (USB suspend)	1.91	mA
		EGSM @DRX=9 (USB disconnect)	1.51	mA
		DCS @DRX=2 (USB disconnect)	2.02	mA
		DCS @DRX=5 (USB disconnect)	1.45	mA
		DCS @DRX=5 (USB suspend)	1.64	mA
		DCS @DRX=9 (USB disconnect)	1.32	mA
		TD-SCDMA Band A @PF=64 (USB disconnect)	2.03	mA
		TD-SCDMA Band A @PF=128 (USB disconnect)	1.67	mA
		TD-SCDMA Band A @PF=256 (USB disconnect)	1.56	mA

		TD-SCDMA Band A @PF=512 (USB disconnect)	1.42	mA
		BC0 @SCI=1 (USB disconnect)	3.45	mA
		BC0 @SCI=1 (USB suspend)	3.74	mA
		WCDMA @PF=64 (USB disconnect)	2.02	mA
		WCDMA @PF=64 (USB suspend)	2.17	mA
		WCDMA @PF=128 (USB disconnect)	1.71	mA
		WCDMA @PF=256 (USB disconnect)	1.42	mA
		WCDMA @ PF=512 (USB disconnect)	1.33	mA
		LTE-FDD @PF=32 (USB disconnect)	3.37	mA
		LTE-FDD @PF=64 (USB disconnect)	2.27	mA
		LTE-FDD @PF=64 (USB suspend)	2.53	mA
		LTE-FDD @PF=128 (USB disconnect)	1.86	mA
		LTE-FDD @PF=256 (USB disconnect)	1.52	mA
		LTE-TDD @PF=32 (USB disconnect)	3.41	mA
		LTE-TDD @PF=64 (USB disconnect)	2.27	mA
		LTE-TDD @PF=64 (USB suspend)	2.51	mA
		LTE-TDD @PF=128 (USB disconnect)	1.71	mA
		LTE-TDD @PF=256 (USB disconnect)	1.42	mA
	Idle Mode	EGSM @DRX=5 (USB disconnect)	17.54	mA
		EGSM @DRX=5 (USB connect)	27.67	mA
		BC0 @SCI=1 (USB disconnect)	18.92	mA
		BC0 @SCI=1 (USB connect)	29.08	mA
		TD-SCDMA Band A @PF=64 (USB disconnect)	17.61	mA
		TD-SCDMA Band A @PF=64 (USB connect)	27.60	mA
		WCDMA @PF=64 (USB disconnect)	17.92	mA

		WCDMA @PF=64 (USB connect)	28.00	mA
		LTE-FDD @PF=64 (USB disconnect)	17.84	mA
		LTE-FDD @PF=64 (USB connect)	27.94	mA
		LTE-TDD @ PF=64 (USB disconnect)	18.11	mA
		LTE-TDD @ PF=64 (USB connect)	28.08	mA
	GPRS data transmission (GNSS off)	GSM900 4DL/1UL @32.62dBm	246.8	mA
		GSM900 3DL/2UL @32.45dBm	418.3	mA
		GSM900 2DL/3UL @30.73dBm	513.2	mA
		GSM900 1DL/4UL @29.75dBm	594.3	mA
		DCS1800 4DL/1UL @29.57dBm	170.8	mA
		DCS1800 3DL/2UL @29.45dBm	274.9	mA
		DCS1800 2DL/3UL @29.28dBm	374.8	mA
		DCS1800 1DL/4UL @29.11dBm	475.5	mA
	EDGE data transmission (GNSS off)	GSM900 4DL/1UL @27.24dBm	157.3	mA
		GSM900 3DL/2UL @27.14dBm	258.8	mA
		GSM900 2DL/3UL @27.01dBm	358.3	mA
		GSM900 1DL/4UL @26.91dBm	461.0	mA
		DCS1800 4DL/1UL @25.85dBm	143.4	mA
		DCS1800 3DL/2UL @25.57dBm	235.2	mA
		DCS1800 2DL/3UL @25.55dBm	323.7	mA
		DCS1800 1DL/4UL @25.22dBm	415.7	mA
	CDMA/TD-SCDMA data transmission(GNSS off)	BC0 @23.98dBm	600.7	mA
		TD-SCDMA Band A @23.42dBm	130.6	mA
		TD-SCDMA Band F @23.32dBm	131.9	mA
	WCDMA data	WCDMA B1 HSDPA @21.06dBm	503.8	mA

	transmission(GNSS off)	WCDMA B1 HSUPA @20.56dBm	500.6	mA
		WCDMA B8 HSDPA @21.16dBm	469.5	mA
		WCDMA B8 HSUPA @20.83dBm	527.2	mA
	LTE data transmission (GNSS off)	LTE-FDD B1 @22.04dBm	709.7	mA
		LTE-FDD B3 @22.87dBm	717.1	mA
		LTE-FDD B5 @22.11dBm	609.6	mA
		LTE-FDD B8 @22.40dBm	609.4	mA
		LTE-TDD B38 @22.75dBm	434.4	mA
		LTE-TDD B39 @22.90dBm	336.5	mA
		LTE-TDD B40 @23.04dBm	360.5	mA
		LTE-TDD B41 @22.95dBm	403.8	mA
	GSM voice communication	GSM900PCL=5 @32.71dBm	244.4	mA
		GSM900PCL=12 @19.53dBm	111.7	mA
		GSM900PCL=19 @5.69dBm	81.2	mA
		DCS1800 PCL=0 @29.64dBm	165.6	mA
		DCS1800 PCL=7 @16.66dBm	126.4	mA
		DCS1800 PCL=15 @0.41dBm	105.0	mA
	CDMA voice communication	BC0 @24.09dBm	686.3	mA
		BC0 @-60.12dBm	114.3	mA
	WCDMA voice communication	WCDMA B1 @23.01dBm	607.9	mA
		WCDMA B8 @22.57dBm	542.3	mA

8 Analysis of High Sleep Consumption

If it is confirmed that the module is in sleep mode but the average current consumption is high, there are three reasons, unexpected wakeup, base current and RF (such as standard, band)

(1) Users can through consumption capture diagram of power consumption catcher to check whether there's unexpected wake-up, also can use debug uart to capture log. The steps are as following.

1) Execute below steps.

```
~ # echo 1 > /sys/module/printk/parameters/perf_mode_console
~ # echo 1 > /sys/module/msm_show_resume_irq/parameters/debug_mask
~ # echo 0x2 > /sys/module/ipc_router_core/parameters/debug_mask
```

2) Then let module go sleep, check serial port log. If there is log like following, the module has woken-up.

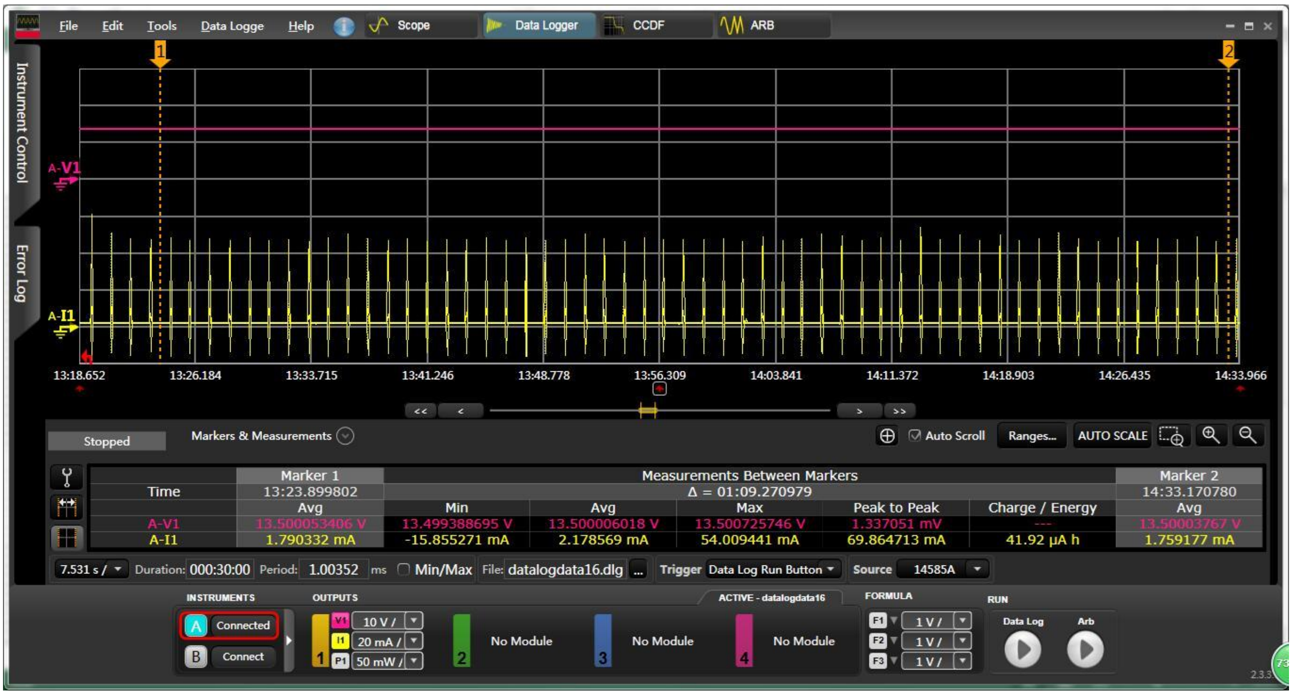
```
[   113.386694] gic_show_resume_irq: 57 triggered qcom,smd-modem
[   113.386694] gic_show_resume_irq: 200 triggered qcom,smd-rpm
[   113.386694] resume cycles:           2542257600
[   113.388512] [IPCRTR] CLI RX Len:0xd T:0x1 CF:0x0 SVC:<0x3:0x1> SRC:<0x3:0x11>
DST:<0x1:0x43> DATA: 51000b04 13000600
[   113.388520] PM: noirq resume of devices complete after 0.975 msecs
[   113.389994] PM: early resume of devices complete after 1.088 msecs
```

3) Then judge whether it was an unexpected wakeup and deal with it.

(2) If no abnormal wakeup is found in step (1), check if the base current is high.

Take AT+CFUN=0 to off RF interference, check whether base current is high, lower than 1.5mA is normal. If the base current is too high, it is necessary to check whether the pin configuration forms a leakage circuit with the external circuit.

(3) If the first two steps are normal, but the sleep consumption is still high, use power consumption catcher to capture accurate consumption chart, as shown below.



	Marker 1	Measurements Between Markers					Marker 2
Time	13:23.899802	Δ = 01:09.270979					14:33.170780
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-V1	13.500053406 V	13.499388695 V	13.500006018 V	13.500725746 V	1.337051 mV	---	13.50003767 V
A-I1	1.790332 mA	-15.855271 mA	2.178569 mA	54.009441 mA	69.864713 mA	41.92 µA h	1.759177 mA

7.531 s / Duration: 000:30:00 Period: 1.00352 ms □ Min/Max File: datalogdata16.dlg ... Trigger Data Log Run Button Source 14585A

Users can check that cause of high current consumption is the period of the DRX (Discontinuous Reception) is too small or the peak of the DRX is too high.