EC2X&AG35-QuecOpen Cm256sm BLE User Guide

LTE Module Series

Rev.EC2X&AG35-QuecOpen cm256sm BLE User Guide_V1.1

Date: 2018-09-04

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	2018-03-21	Quinn ZHAO	Initial
1.1	2018-09-04	Grady	Updated BT_HOST_WAKE GPIO Pin Updated BT_WAKE GPIO Pin Updated UART Pin

Contents

Table Index

1 Introduction

The document is designed to help customers to quickly use the Bluetooth function of cm256sm WIFI/BT module on Quectel EC2X&AG35-QuecOpen platform.

2 Hardware Interface

Cm256sm Bluetooth module communicates with EC2X or AG35 via hardware-flow-controlled 4-line UART. In addition EC2X and AG35 need reserve one GPIO as power control pin for cm256sm. If sleep and wake up function is required, also need to reserve two pins more, one of them is input interruption for EC2X or AG35, another is the general GPIO output pin.

2.1. The Pin Relationship Between EC2X and cm256sm

Table 1: The pin relationship between EC2X and cm256sm

EC2X		cm256sm	
Pin No.	Definition	Pin No.	Definition
2	GPIO_10(Output)	6	BT_WAKE(Input)
3	GPIO_42(Input)	7	BT_HOST_WAKE(Output)
139	PMU_GPIO1019(Output)	34	BT_REG_ON(Input)
37	UART_RTS_BLSP6/GPIO_22(Input)	41	BT_UART_RTS_N(Output)
38	UART_RXD_BLSP6/GPIO_21(Input)	42	BT_UART_TXD(Output)
39	UART_TXD_BLSP6/GPIO_20(Output)	43	BT_UART_RXD(Input)
40	UART_CTS_BLSP6/GPIO_23(Output)	44	BT_UART_CTS_N(Input)

2.2. The Pin Relationship Between AG35 and cm256sm

Table 2: The pin relationship between EC2X and cm256sm

AG35		cm256sm	
Pin No.	Definition	Pin No.	Definition

150	GPIO_42(Output)	6	BT_WAKE(Input)
62	GPIO_74(Input)	7	BT_HOST_WAKE(Output)
139	PMU_GPIO1019（Output）	34	BT_REG_ON(Input)
166	UART_RTS_BLSP5/GPIO_10(Input)	41	BT_UART_RTS_N(Output)
165	UART_RXD_BLSP5/GPIO_9(Input)	42	BT_UART_TXD(Output)
163	UART_TXD_BLSP5/GPIO_8(Output)	43	BT_UART_RXD(Input)
164	UART_CTS_BLSP5/GPIO_11(Output)	44	BT_UART_CTS_N(Input)

3 Driver Adaptation

According to Chapter 2, UART driver adaptation is required, BT_REG_ON pin is common GPIO output driver, BT_HOSTWAKE pin interruption input drive, BT_DEVWAKE is ordinary GPIO output driver.

3.1. UART Driver Adaptation

There are 4 UART on EC2X-QuecOpen Module: main UART, debug UART, UART 1 and UART 2. UART 1 and UART 2 have same function, both support RTS/CTS, can be as a peripheral communication UART. Which, RTS/CTS of UART 1 is multiplexed with I2C, UART 2 is multiplexed with SPI. It's recommended to choose main UART or UART 2 as BLE communication UART.

Since UART 2 is multiplexed with SPI, it is not turned on by default. The method to enable it is as following.
In the file mdm9607-mtp.dtsi, disable spi_6, enable blsp1_uart6.

```
&spi_6 {
-     status = "ok";
+     status = "disabled";
};
&blsp1_uart6 {
-     status = "disabled";
+     status = "ok";
};
```

3.2. BT_REG_ON Pin Driver Adaptation

QuecOpen SDK already support most pins in EC2X&AG35, to realize output function for general GPIO through application layer configuration. Please according to actual hardware to select GPIO. The details please refer to Chapter 5.1.

3.3. Sleep Wake-up Driver Transplantation

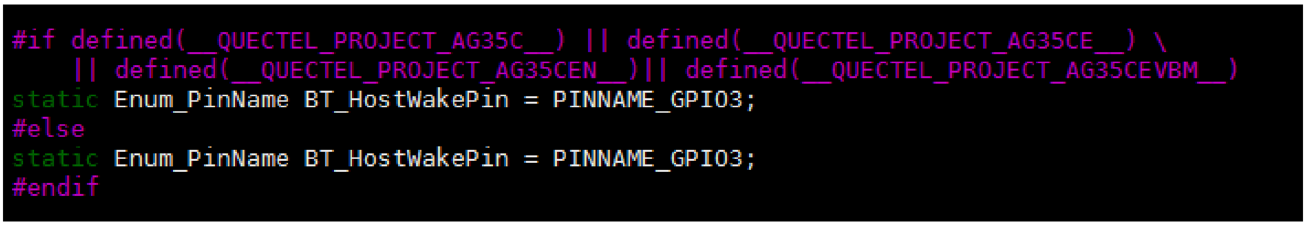
BSA protocol stack required the GPIO ID of BT_HOSTWAKE & BT_WAKE be respectively written to path **/sys/class/bluetooth/wake_gpios/host_wake** & **/sys/class/bluetooth/wake_gpios/dev_wake.** Therefore, need create two readable files **/sys/class/bluetooth/wake_gpios/host_wake** & **/sys/class/bluetooth/wake_gpios/dev_wake.**

Please modify file **hci_sysfs.c** of path is **/sys/class/bluetooth/wake_gpios/dev_wake,** codes added are shown as following.

```
--- /home/quinn1/Develop/SDK/EC20CEFAG/ql-ol-sdk/ql-ol-kernel/net/bluetooth/hci_sysfs.c    2018-03-07 20:28:15.000000000 +0800
+++ hci_sysfs.c   2018-02-05 22:54:06.000000000 +0800
@@ -6,6 +6,10 @@
 #include <net/bluetooth/hci_core.h>

 static struct class *bt_class;
+static struct device *bt_wakeup_gpio;
+
+static struct device_attribute dev_wake;
+static struct device_attribute host_wake;

 static inline char *link_typetostr(int type)
 {
@@ -201,9 +205,70 @@
     device_initialize(dev);
 }

+
+static int dev_wake_value = -1;
+
+static ssize_t
+show_dev_wake(struct device *dev, struct device_attribute *attr, char *buf)
+{
+    return snprintf(buf, PAGE_SIZE, "%d\n", dev_wake_value);
+}
+
+static ssize_t
+store_dev_wake(struct device *dev, struct device_attribute *attr,
+            const char *buf, size_t count)
+{
+    kstrtou32(buf, 0, &dev_wake_value);
+    return count;
+}
+
+static int host_wake_value = -1;
+
+static ssize_t
+show_host_wake(struct device *dev, struct device_attribute *attr, char *buf)
+{
+    return snprintf(buf, PAGE_SIZE, "%d\n", host_wake_value);
+}
+
+static ssize_t
+store_host_wake(struct device *dev, struct device_attribute *attr,
+            const char *buf, size_t count)
+{
+    kstrtou32(buf, 0, &host_wake_value);
+    return count;
+}
+
+
 int __init bt_sysfs_init(void)
 {
+      int ret = -1;
+
     bt_class = class_create(THIS_MODULE, "bluetooth");
+             bt_wakeup_gpio = device_create(bt_class, NULL, 0, NULL, "wake_gpios");
+    if (IS_ERR(bt_wakeup_gpio))
+          printk("[Quinn] Create Failed\n");
+
+    dev_wake.show = show_dev_wake;
+    dev_wake.store = store_dev_wake;
+    sysfs_attr_init(&dev_wake.attr);
+    dev_wake.attr.name = "dev_wake";
+    dev_wake.attr.mode = S_IRUGO | S_IWUSR;
+    ret = device_create_file(bt_wakeup_gpio, &dev_wake);
+             if (ret) {
+          printk("[Quinn] Create dev_wake Failed\n");
+          device_remove_file(bt_wakeup_gpio, &dev_wake);
+    }
+
+    host_wake.show = show_host_wake;
+    host_wake.store = store_host_wake;
+    sysfs_attr_init(&host_wake.attr);
+    host_wake.attr.name = "host_wake";
+    host_wake.attr.mode = S_IRUGO | S_IWUSR;
+    ret = device_create_file(bt_wakeup_gpio, &host_wake);
+             if (ret) {
+          printk("[Quinn] Create dev_wake Failed\n");
+          device_remove_file(bt_wakeup_gpio, &dev_wake);
+   }

    return PTR_ERR_OR_ZERO(bt_class);
 }
```

3.4. BT_HOST_WAKE Pin Driver Adaptation

QuecOpen SDK already support most pins in EC2X&AG35 to realize interruption function through application layer configuration. Please according to actual firmware to select the pin and modified file ql_cm256sm_ble_sleep.c in ql-ol-sdk/ql-ol-extsdk/example/bt/cm256sm/app_ble_test/source. Example is shown as following.



```
#if defined(__QUECTEL_PROJECT_AG35C__) || defined(__QUECTEL_PROJECT_AG35CE__) \
    || defined(__QUECTEL_PROJECT_AG35CEN__)|| defined(__QUECTEL_PROJECT_AG35CEVBM__)
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
#else
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
#endif
```

3.5. BT_DEVWAKE Pin Driver Adaptation

QuecOpen SDK already support most pins in EC2X&AG35, to realize output function for general GPIO through application layer configuration. Please according to actual hardware to select GPIO. The details please refer to Chapter 5.2.

4 Sleep Wake-up Function Introduction

The cm256sm module use two pins to implement the sleep wake-up function. BT_HOSTWAKE is an interruption input pin for the EC20&AG35 and an output pin for the cm256sm Bluetooth module. BT_DEVWAKE is an ordinary output pin for the EC20&AG35 and an input pin for the cm256sm module.

4.1. BT_HOSTWAKE Pin Function Introduction

When there is no data interaction, BT_HOSTWAKE is high-level. After the mobile phone sends out a read/write data interaction request, BT_HOSTWAKE will be pulled low by the Bluetooth module to be low-level. According to this interruption, EC2X or AG35 applies for holding a wakelock to keep EC2X or AG35 awake. After the phone is disconnected from the Bluetooth module, the EC2X or AG35 releases the wakelock, allowing the EC2X or AG35 to go sleep mode.

4.2. BT_DEVWAKE Pin Function Introduction

When there is no data interaction, BSA protocol stack will pull up BT_DEVWAKE pin, allowing Bluetooth module go sleep mode. When there is data interaction, BAS protocol stack will pull low BT_DEVWAKE pin, waking up Bluetooth module.

4.3. Introduction EC2X&AG35 RTS Pin Introduction

The judgement of Bluetooth module whether send data to EC2X/AG35 is according to its CTS, i.e. whether RTS of EC2X and AG35 is low-level. If it’s low-level, will send data to EC2X /G35. Therefore, when EC2X/AG35 in sleep mode, please maintain the RTS output is high-level in case the data lose.

5 BSA Protocol Stack Introduction

BSA (Bluetooth Simple API) protocol stack from Broadcom, is to simplify the design of Bluetooth function and improve the development efficiency. BSA is based on classical C/S structure, and executable program "bsa_server" running in the background of EC2X or AG35 exists as server. This executable program is a Bluetooth protocol stack, and implements a plurality of commonly used protocol stacks and profiles, and performs data interaction with the Bluetooth module based on the HCI command through the UART. The user writes one or more client programs according to actual needs and communicates with the "bsa_server" via the socket file.

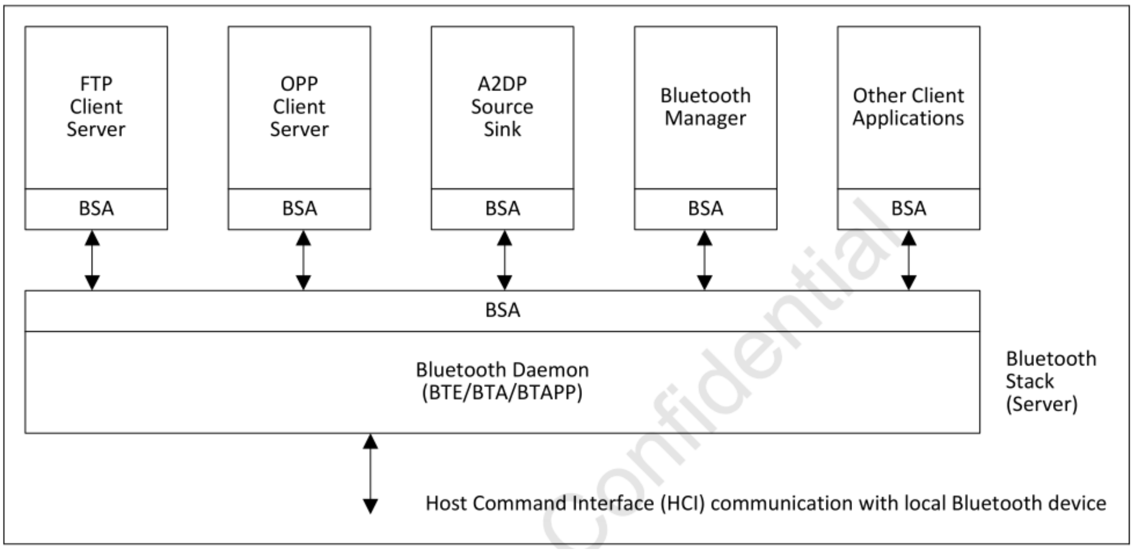


Figure 1: BSA Frames Diagram

5.1. Power-on Reset

EC2X/AG35 is according to the power-on time sequence requirement of cm256sm module to pull up or pull low BT_REG_ON pin. The specific process is: pull low BT_REG_ON -> keep 1-3s -> pull up BT_REG_ON. Script example is shown as following.

```
echo 1019 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio1019/direction
echo 0 > /sys/class/gpio/gpio1019/value
sleep 3
```

```
echo 1 > /sys/class/gpio/gpio1019/value
```

Please refer to ***Quectel_AG35-QuecOpen_Hardware_Design*** & ***Quectel_EC20_R2.0-QuecOpen_Hardware_Design*** and actual circuit design to select corresponding pin.

5.2. Sleep Wake-up Pin Configuration

BSA protocol stack requires GPIO ID of BT_HOSTWAKE & BT_WAKE respectively written to path **/sys/class/bluetooth/wake_gpios/host_wake** & **/sys/class/bluetooth/wake_gpios/dev_wake.** BT_DEVWAKE pin initialization should be low-level. Example is shown as following.

```
echo 10 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio10/direction
echo 0 > /sys/class/gpio/gpio10/value
echo 10 > /sys/class/bluetooth/wake_gpios/dev_wake
echo 42 > /sys/class/gpio/export
echo 42 > /sys/class/bluetooth/wake_gpios/host_wake
```

5.3. Execute bsa_server

Run the bsa_server reference SDK path is in README in "**ql-ol-sdk/ql-ol-extsdk/example/bt/cm256sm**", and please also import files bsa_serve & BCM434545.hcd to EC2X or AG35.

bsa_serve supports import multiple parameters to realize different functions. Use command **./bsa_server --hep** to get the detailed introduction of parameters. Command example is shown as following.

```
./bsa_server -d /dev/ttyHSL1 -p BCM434545.hcd -lpm -b /tmp/btsnoop_lpm.cfa -all=6 >
/tmp/bsa_server_lpm.log &
```

In which, "-d" specifies the device path and name of the communication UART used by EC2X or AG35 and cm256sm. It needs to be filled in according to the actual hardware and driver design. "-p" specifies the absolute path where the Bluetooth firmware is located. "-lpm" indicates that the low power consumption or sleep wakeup mode is enabled. Both "-b" and "-all" are related to log collection.
After Bsa_server is executed normally, a socket file named "bt-daemon-socket" will be generated. If no such file is generated, please check the hardware design and pin configuration again for any problems.

5.4. Execute bsa_client

Please according to actual application requirements and refer to ***BSA_API_Programming_Guide.pdf***, to develop bsa_client program. The connection is established by socket files generated by bsa_server executing. Quectel has already provided demo program, path is **ql-ol-sdk/ql-ol-**

extsdk/example/bt/cm256sm, the reference SDK path is README in **"ql-ol-sdk/ql-ol-extsdk/example/bt/cm256sm"**

6 Testing and Verifying

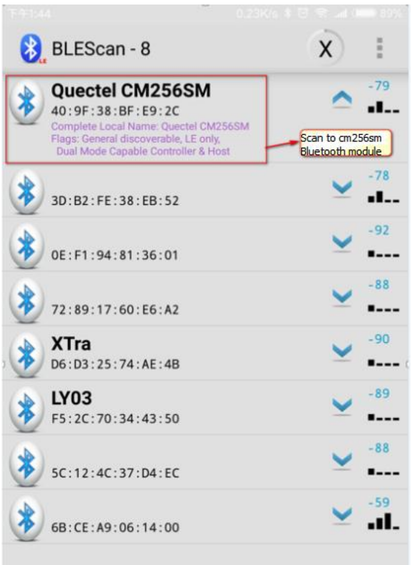
BLE is turned off on mobile phone by default, therefore mobile phone terminals need to download APP for function verification. If it's Android platform, please download BLE Deng APP. If it's iphone platform, please download LightBlue APP. This document is based on Android BLE Deng APP.

6.1. Basic Function Testing

The Chapter 6 mainly introduces the scan, connection and data interaction testing. Please turn on Bluetooth function and BLE Deng APP.

6.1.1. Scan Testing

Turn on BLE Deng App and it will automatically scans surrounding BLE device.



Quectel CM256SM
40:9F:38:BF:E9:2C
Complete Local Name: Quectel CM256SM
Flags: General discoverable, LE only,
Dual Mode Capable Controller & Host
-79

Scan to cm256sm Bluetooth module

3D:B2:FE:38:EB:52 -78

0E:F1:94:81:36:01 -92

72:89:17:60:E6:A2 -88

XTra
D6:D3:25:74:AE:4B -90

LY03
F5:2C:70:34:43:50 -89

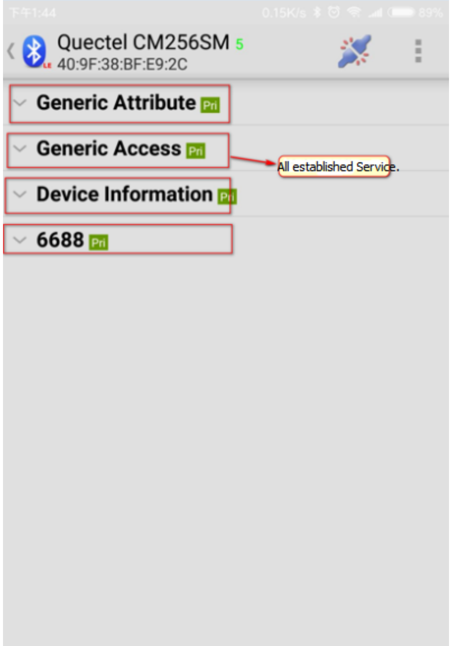
5C:12:4C:37:D4:EC -88

6B:CE:A9:06:14:00 -59

As shown above, mobile phone scanned to cm256sm Bluetooth module successfully.

6.1.2. Connection Testing

Please click 'Quectel CM256SM', connect mobile phone with cm256sm. After the connection is established, BLE Deng App will does 'Service Discovery', then scans all Service on cm256sm, shown as following.



下午1:44 0.15K/s 89%

Quectel CM256SM 5
40:9F:38:BF:E9:2C

Generic Attribute Pri

Generic Access Pri

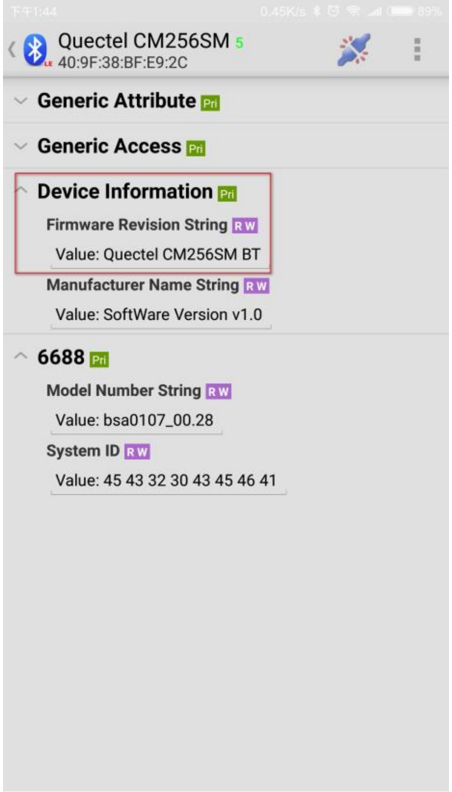
Device Information Pri

6688 Pri

All established Service.

6.1.3. Data Interaction Testing

Please click ‘Device Information’ Service, read characteristic data in ‘Firmware Revision String’. Shown as following.



下午1:44 0.45K/s 89%

Quectel CM256SM 5
40:9F:38:BF:E9:2C

Generic Attribute Pri

Generic Access Pri

Device Information Pri

Firmware Revision String R W
Value: Quectel CM256SM BT

Manufacturer Name String R W
Value: SoftWare Version v1.0

6688 Pri

Model Number String R W
Value: bsa0107_00.28

System ID R W
Value: 45 43 32 30 43 45 46 41

6.2. Sleep Wake-up function Testing

Before testing please insert bluetooth antennas to ensure good signal quality.

6.2.1. Cm256sm Waking-up EC2X/G35 Testing

Pull out USB cable, execute demo program, after initialization is done, EX2X & AG35 will go sleep mode as demo program enables autosleep function, data interaction can't be done under UART terminals. At this time, mobile phone still can scan cm256sm Bluetooth module, after mobile phone connecting with cm256sm, Bluetooth module will pull low BT_HOST_WAKE pin and generate interruption to wake up EC2X/G35. When the mobile phone disconnect with cm256sm and there's no any task holding wakelock, EC2X/G35 will go sleep mode, data interaction can't be done under UART terminals.

6.2.2. EC2X/AG35 Waking-up cm256sm Testing

When there is no data interaction, BSA protocol stack will pull up BT_DEVWAKE pin, allowing Bluetooth Module go sleep mode. When EC2X/AG35 trying to interacting with cm256sm, BSA protocol stack will pull up BT_DEVWAKE pin to wake up cm256sm. The oscilloscope can monitor the level change of BT_DEVWAKE pin and test the consumption of Bluetooth module in real time, and verify whether the Bluetooth module is sleeping or waking up.

7 Appendix A References

Table 3: Related Documents

SN	Document Name	Remark
[1]	Quectel_AG35-QuecOpen_Hardware_Design	AG35 Hardware Design Guide
[2]	Quectel_EC20-QuecOpen_Hardware_Design	EC20 Hardware Design Guide
[3]	AW-CM256SM_DS_Rev 14_CYW	Cm256sm WIFI/BT Module Spec
[4]	BSA_API_Programming_Guide	BSA Stack API Programming Guide

Table 4: Term and Abbreviation

Abbreviation	Description
BLE	Bluetooth Low Energy