EC2x&AG35-Quecopen NF3303 Bluetooth Module User Guide

LTE Standard/Automotive Module Series

Rev.EC2x&AG35Quecopen_NF3303_Bluetooth_Module_User_Guide_V1.1

Date: 2018-09-06

Status: Preliminary

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 LV	Initial
1.1	2018-09-06	Grady QUAN	Updated GPIO pins and some chapters contents

Contents

Table Index

1 Introduction

The document is designed to help customers to quickly use the Bluetooth function of NF3303 WIFI/BT module on Quectel EC2x&AG35-Quecopen platform to achieve business needs.

This document mainly applies to overseas market, The LTE Standard/Automotive module currently supporting includes:

- EC2x:EC 20 R2.1/EC25/EC21
- AG35

2 Hardware Interface

NF3303 Bluetooth module communicates with EC2x or AG35 via hardware-flow-controlled four-wire serial port. In addition, EC2x or AG35 needs reserve one GPIO as power supply control pin for NF3303 Bluetooth module. If customers want it to support sleep and wake up function, two more pins also need to reserved, one is input interruption for EC2x or AG35, another is the general GPIO output pin.

2.1. The Pin Relationship Between EC2x and NF3303

Table 1: The pin relationship between EC2x and NF3303

EC2x		NF3303	
Pin No.	Definition	Pin No.	Definition
2	GPIO_10(Output)	40	BT_DEVWAKE (Input)
3	GPIO_42 (Input)	39	BT_HOST_WAKE_UP(Output)
139	PMU_GPIO1019(Output)	24	BT_EN (Input)
37	UART_RTS_BLSP6/GPIO_22(Input)	10	BT_UART_RTS_N (Output)
38	UART _RXD_BLSP6/GPIO_21(Input)	7	BT_UART_TXD (Output)
39	UART _TXD_BLSP6/GPIO_20(Output)	6	BT_UART_RXD (Input)
40	UART _CTS_BLSP6/GPIO_23(Output)	11	BT_UART_CTS_N (Input)
26	PCM_SYNC/GPIO_79	34	AUDIO_FSYNC_BT
27	PCM_CLK/GPIO_78	33	AUD_CLK_BT
25	PCM_OUT/GPIO_77	32	AUD_IN_BT
24	PCM_IN/GPIO_76	35	AUD_OUT_BT

2.2. The Pin Relationship Between AG35 and NF3303

Table 2: The pin relationship between AG35 and NF3303

AG35		NF3303	
Pin No.	Definition	Pin No.	Definition
150	GPIO_42(Output)	40	BT_DEVWAKE (Input)
62	GPIO_74 (Input)	39	BT_HOST_WAKE_UP(Output)
3	PMU_GPIO1019(Output)	24	BT_EN (Input)
166	UART_RTS_BLSP5/GPIO_10(Input)	10	BT_UART_RTS_N (Output)
165	UART _RXD_BLSP5/GPIO_9(Input)	7	BT_UART_TXD (Output)
163	UART _TXD_BLSP5/GPIO_8(Output)	6	BT_UART_RXD (Input)
164	UART _CTS_BLSP5/GPIO_11(Output)	11	BT_UART_CTS_N (Input)
77	PCM_1_SYNC/GPIO_20	34	AUDIO_FSYNC_BT
80	PCM_1_CLK/GPIO_23	33	AUD_CLK_BT
79	PCM_1_DOUT/GPIO_22	32	AUD_IN_BT
78	PCM_1_DIN/GPIO_21	35	AUD_OUT_BT

3 Driver Adaptation

According to Chapter 2, it needs adaptation serial port driver, BT_EN pin ordinary GPIO output driver, BT_HOST_WAKE_UP pin interruption input driver and BT_DEVWAKE ordinary GPIO output driver.

3.1. Serial Port Driver Adaptation

There are 4 serial ports on EC2x-QuecOpen Module: main serial port, debug serial port, serial port 1 and serial port 2. Serial port 1 and serial port 2 have the same function, both support RTS/CTS, can be serve as a peripheral communication serial port. Among them, RTS/CTS of serial port 1 is multiplexed with I2C, serial port 2 is multiplexed with SPI. The recommendation is choosing main serial port or serial port 2 as communication serial port of Bluetooth.

As the serial port 2 is multiplexed with the SPI, so it is not turned on under defaulted status, and the opening method is as follows:
Please close spi_6 and open blsp1_uart6 in the mdm9607-mtp.dtsi

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

3.2. BT_EN Pin Driver Adaptation

QuecOpen SDK already support most pins in EC2x&AG35, to realize output function for general GPIO through application layer configuration. Please confirm the GPIO used according to the actual hardware. EC2X/AG35 needs to pull up or pull low BT_EN pin according to the power-on time sequence requirement of NF3303 module. The specific process is to pull low BT_EN -> keep 200ms -> pull up BT_EN. Please according to actual hardware, modify the functions *ql_bt_en_pin_init* and *ql_bt_module_enable* which in the file ql_nf3303_ble_common.c in the path of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/source.* Example is shown as follows.

```
/*Operate BT_EN to enable BT Module*/
void ql_bt_module_enable()
{
    /*Pull Up PM_ENABLE to Power the Module*/
    system("echo 1 > /sys/kernel/debug/regulator/rome_vreg/enable");
    /*Reset BT Module*/
    ioctl(bt_power_fd, 0);    //pull Down BT_EN
    usleep(200000);
    ioctl(bt_power_fd, 1);    //pull Up BT_EN
}
```

3.3. BT_HOST_WAKE Pin Driver Adaptation

QuecOpen SDK already support most pins in EC2X&AG35 to realize interruption function through application layer configuration. Please confirm the GPIO used according to the actual hardware. The modification path is file ql_NF3303_ble_sleep.c in *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/source*, macro definition BT_HostWakePin is the actual hardware used pin. Example is shown as follows.

```
#if defined(__QUECTEL_PROJECT_AG35C__) || defined(__QUECTEL_PROJECT_AG35CE__) \
    || defined(__QUECTEL_PROJECT_AG35CEN__)|| defined(__QUECTEL_PROJECT_AG35CEVBM__)
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
static Enum_PinName BT_DevWakePin = PINNAME_GPIO6;
#else
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
static Enum_PinName BT_DevWakePin = PINNAME_GPIO2;
#endif
```

3.4. 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 confirm GPIO used according to actual hardware. The modification path is ql_NF3303_ble_sleep.c in the path of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/source*, macro definition BT_DevWakePin is the actual hardware used pin. Example is shown as follows.

```
#if defined(__QUECTEL_PROJECT_AG35C__) || defined(__QUECTEL_PROJECT_AG35CE__) \
    || defined(__QUECTEL_PROJECT_AG35CEN__)|| defined(__QUECTEL_PROJECT_AG35CEVBM__)
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
static Enum_PinName BT_DevWakePin = PINNAME_GPIO6;
#else
static Enum_PinName BT_HostWakePin = PINNAME_GPIO3;
static Enum_PinName BT_DevWakePin = PINNAME_GPIO2;
#endif
```

4 Sleep Wake-up Function Instruction

The NF3303 Bluetooth module uses two pins to implement sleep wake-up function. BT_HOSTWAKE_UP is an interruption input pin for the EC20&AG35 and an output pin for the NF3303 Bluetooth module. BT_DEVWAKE is an ordinary output pin for EC20&AG35 and an input pin for the NF3303 Bluetooth module.

4.1. BT_HOST_WAKE_UP Pin Function Instruction

When there is no data interaction, BT_HOST_WAKE_UP is low-level. After the mobile phone sends out a request of read/write data interaction, NF3303 send data to EC2x or AG35 and pull up BT_HOST_WAKE_UP pin, i.e. rising edge interruption occurs to the EC2X or AG35. After the data is sent, pull low the BT_HOST_WAKE pin to generate a falling edge interruption, that is, a falling edge interruption is sent to EC2X or AG35.

4.2. BT_DEVWAKE Pin Function Instruction

When EC2x or AG35 sending data to NF3303, Bluetooth protocol stack will generate callback function, according to this callback function, EC2x or AG35 pulls up BT_DEV_WAKE pin. After the data is sent, Bluetooth protocol stack will also generate another callback function, EC2x or AG35 pulls low BT_DEV_WAKE according to this callback function.

4.3. EC2x&AG35 RTS Pin Instruction

The judgement of Bluetooth module whether sending data to EC2X/AG35 is based on its CTS, namely, whether RTS of EC2x or AG35 is low-lever. If it is low-level, Bluetooth module will send data to EC2x or AG35. Therefore, when EC2X/AG35 in sleep mode, please make sure that the RTS output is high-level in case of the data lose.

4.4. Code Ideas of Sleep Wake-up Function

4.4.1. Initialization Phase of Sleep Wake-up

Register interruption service function of BT_HOST_WAKE_UP. Create the wakelock which is required to 4G module sleep and wake-up function. Pull up BT_DEV_WAKE pin.

4.4.2. Receiving Rising Edge Interruption Code Logic of BT_HOST_WAKE_UP

First to check whether wakelock is lock, if yes, don't need to lock, if no, lock it.

4.4.3. Receiving Falling Edge Interruption Code Logic of BT_HOST_WAKE_UP

A cycle timer with an interval of 5 seconds is started at system startup. When this timer timeout, it will firstly to check whether the BT_HOST_WAKE_UP falling edge interrupt is received, and then to check whether wakelock is unlocked, if yes, don't need to unlock, if no, unlock it.

4.4.4. Code Logic of BT_DEV_WAKE Pin

If receiving Bluetooth protocol stack callback function *NFBT_WARNING_BT_ALLOW_SLEEP*, BT_DEV_WAKE will be pulled low. If receiving Bluetooth protocol stack callback function *NFBT_WARNING_BT_WAKE*, BT_DEV_WAKE will be pulled up.

5 Instruction of Bluegate Protocol Stack

Bluegate protocol stack provided by NF3303 module manufacturer is to simplify the design of Bluetooth function and improve the development efficiency. Developers can directly operate Bluegate to implement most Bluetooth functions. As a middleware, Bluegate implements a variety of commonly used protocol stacks and profiles, providing a set of application programming interfaces (API) for developers to call. Please refer to ***NFore Bluegate Programming Guide.pdf*** to develop application program according actual application requirements

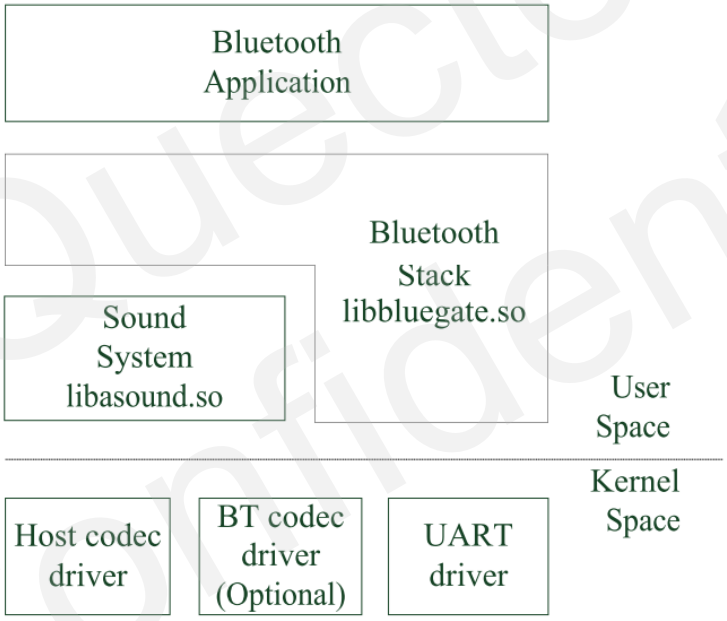
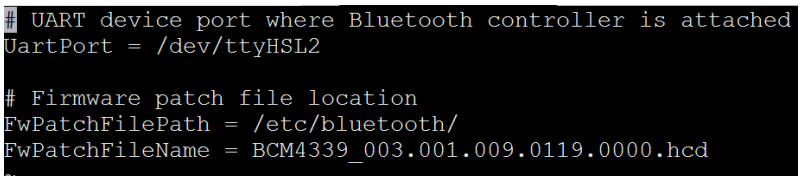


Figure 1: Bluegate Frames Diagram

5.1. Configuring Communication Serial Port

Under the premise that the serial port driver has been configured correctly, import bluegate_hw.conf which is in the path of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/lib/firmware* to directory of /etc/bluetooth/. Bluegate_hw.conf file information is shown as below.



The text information of the modified configuration file is as follow.

```
# UART device port where Bluetooth controller is attached
UartPort = /dev/ttyHSL2

# Firmware patch file location
FwPatchFilePath = /etc/bluetooth/
FwPatchFileName = BCM4339_003.001.009.0119.0000.hcd
```

"UartPort" must be consistent with the device information of the actual serial port. "FwPatchFilePath" should be same with the actual file path of BCM4339_003.001.009.0119.0000.hcd.

5.2. Importing Chip Firmware and Bluetooth Protocol Stack.

5.2.1. Importing Chip Firmware

After EC2x or AG35 module established communication with Bluetooth module, Module will transmit data to Bluetooth module based on HCI command. Thus, the chip firmware needs to be pre-placed in the file system. Please use ADB or other tools to push *BCM4339_003.001.009.0119.0000*.hcd under the path of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/lib/firmware* to /etc/bluetooth/ directory of EC20 or AG35. The reference command is shown as follows.

```
$ adb push BCM4339_003.001.009.0119.0000.hcd /etc/bluetooth/
```

5.2.2. Importing Bluetooth Protocol Stack

The Bluetooth protocol stack, (libbluegate.so) is provided as a dynamic library, and running of client applications program needs to depend on this library file. Therefore, the Bluetooth protocol stack needs to be pre-placed in the file system. Please use ADB or other tools to push *libbluegate.so* under the path of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303/lib/* to directory of EC20 or AG35. The reference command is shown as follows.

```
$ adb push libbluegate.so /lib/
```

6 Demo_UI Operation

6.1. Compiling Demo_UI

Firstly, get the latest SDK, configure the cross-compilation environment, for more information, please refer to the ***SDK Manual***.

Enter into directory of *ql-ol-sdk/ql-ol-extsdk/example/bt/nf3303*
Start to compile:

```
# make clean
# make
```

```
edison@edison-VirtualBox:~/9X28/ql-ol-sdk_/ql-ol-extsdk/example/bt/nf3303$ ls
demo_UI  include  lib  Makefile  README  source
```

Copy the executable program demo_UI generated after compiling above to the rootfs file system.

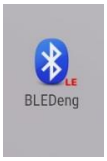
6.2. Running Demo_UI

Firstly to input the command in command line: **amix 'SEC_AUX_PCM_RX Audio Mixer MultiMedia1',**
Run demo_UI, the user can input "help" to check the operation command.

```
/data # ./demo_UI
Register CBK success


************************************************************
****** nFore BlueGate Bluetooth Middleware (v0.3.2.75) *****
************************************************************

Enable BT stack success

Enter the command to execute:
BT Adapter must reset hardware.

Get BT Userial fd: 16

Local device infos:
    Name:   [Bluegate]
    Addr:   [A4:04:50:30:26:31]
    Device Sleep


Wake BT controller right now!

bt state: enable

Local device infos:
    Name:   [Bluegate]
    Addr:   [A4:04:50:30:26:31]
    Device Connectable


Client Register status=0
Service Register status=0
Allow BT controller into sleep

help

****************************************** Help ******************************************
bt_enable(c1)           : Enable BT stack.
bt_disable(c2)          : Disable BT stack.
localinfos(c3)          : Get local device infos.
setname(c4)             : Set local device name.
setscan(c5)             : Set scan mode.
scan(c6)                : Scan nearby bluetooth devices.
stop(c7)                : Stop scan nearby bluetooth devices.
pairlist(c8)            : List paired remote device.
pair(c9)                : Pair with remote device.
delete(c10)             : Delete paired remote device.
connect(c13)            : Connect to paired remote device.
```

6.3. Opening BLE Broadcast Function

Mobile app test tool download:

Platform	APP	Picture Reference
Android	BLEDeng	
IOS	LightBlue	

The following is to open BLE operation by inputing an instruction("->" means the next step)

```
# c345 -> 0
# c328
# c311 -> 1 -> 1
```

For more information, please refer to the following figure:

```
getring(c215)                  : Get ringtone file path.
setringpth(c216)               : Get ringtone file path.
****************************************** Help End ******************************************


Enter the command to execute:

c345


Set GATT role to Peripheral(0), Central(1), or -1 to cancel.

0
Gatt set role : status=0, role=0

Enter the command to execute:
c328

Enter the command to execute:
Gatt service added : status=0 srvc_handle=40
                   : service UUID=000018FF00001000800000805F9B34FB
Gatt characteristic added status : status=0, srvc_handle=40, char. handle=42
                   : characteristic UUID=00002AFF00001000800000805F9B34FB
Gatt descriptor added : status=0, srvc_handle=40, descr_handle=43
                   : descriptor UUID=00002902000010008000000805F9B34FB
Gatt characteristic added status : status=0, srvc_handle=40, char. handle=45
                   : characteristic UUID=00002AFE00001000800000805F9B34FB
Gatt descriptor added : status=0, srvc_handle=40, descr_handle=46
                   : descriptor UUID=00002902000010008000000805F9B34FB
Gatt service start : status=0, srvc_handle=40
c311


Enter a number between 0 to 1 to stop/start listen incoming connect, or -1 to cancel.

1


Enter a number between 0 to 1 to non-connect/connect advertising type, or -1 to cancel.

1
GATT is now prepare listening to role=Peripheral
Listen operation succeed

Enter the command to execute:
Wake BT controller right now!

GATT is now listening to role=Peripheral
Allow BT controller into sleep
```

If the above-mentioned operations are successful, users can search Bluegate by opening the app with mobile phone. Please refer to the following picture.

6.4. Add BLE Security Key Function

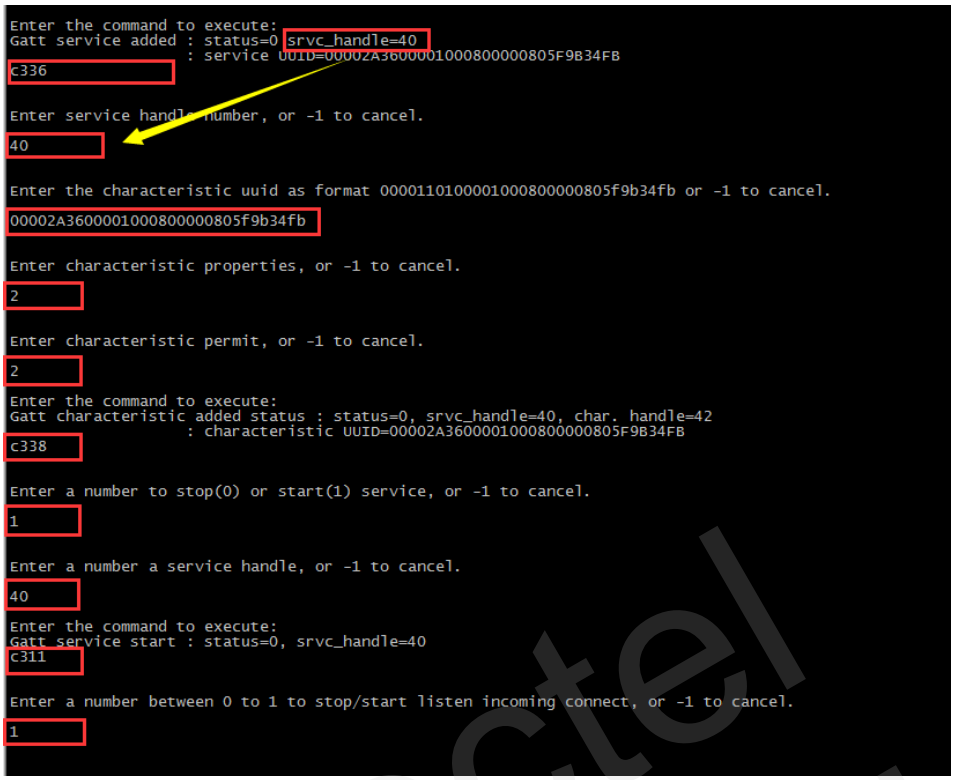
The following is to open BLE operation by inputting instruction ("->" means the next step)

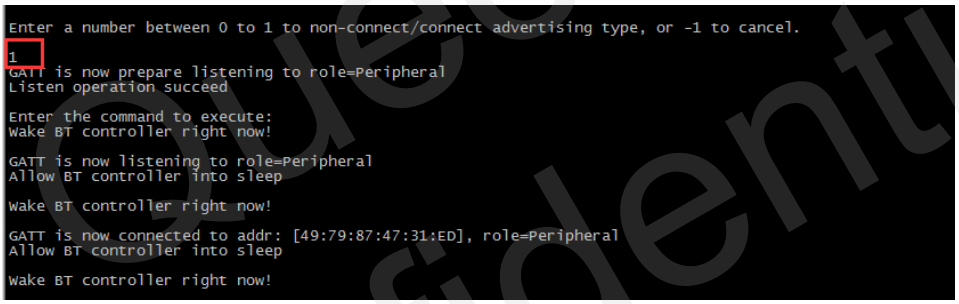
```
# c345 -> 0
# c334 –> 00002A3600001000800000805f9b34fbc336
```

After inputting c334, user can see srv_handle=? by printing, which is for c336 inputting

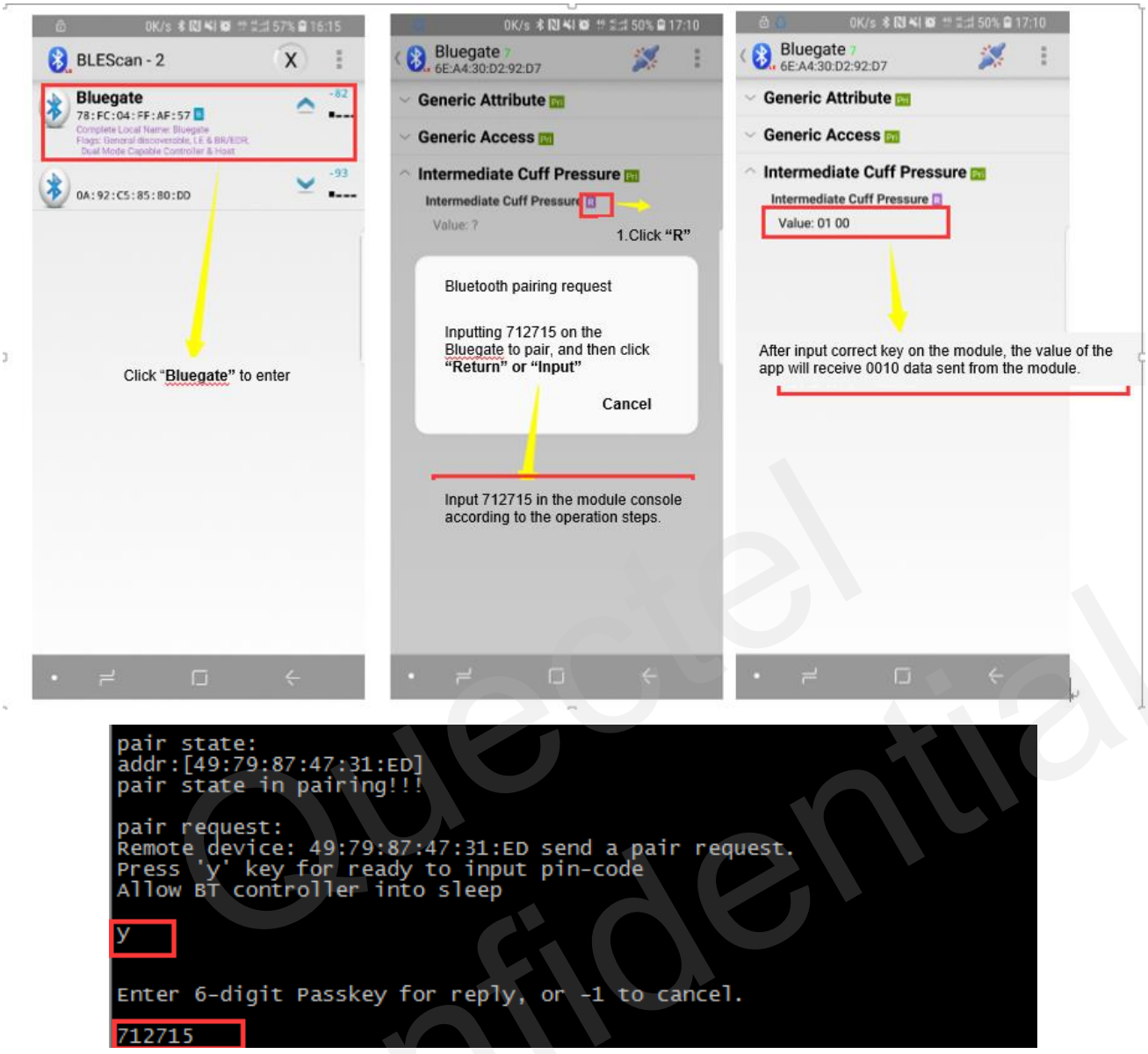
```
# c336 -> srv_handle –> 00002A3600001000800000805f9b34fbc336 -> 2 -> 2
# c338 -> 1 -> srv_handle
# c311 -> 1 -> 1
```

Please refer to the following console operation picture:





After the operation, the mobile phone opens BLEDeng to enter the Bluegate, click on **"R"**, and according to the pairing request, input y, 7127715 respectively in the module console for pairing, if the mobile terminal is successfully paired, then the data sent by the module will be received.



6.5. BLE Data Transmission (peripheral->central)

The current test indicates that the data is sent from the module to the mobile terminal.

The following is to open BLE operation by inputting an instruction ("->" means the next step)

```
# c345 -> 0
# c328
# c311 -> 1 -> 1
```

After the c311 command input completed, the mobile phone will receive the broadcast Bluegate, click to enter and open 18FF, click the RF icon on the right to become solid

The next step is to execute data transmission operations, please referr to the following console screenshots:

```
# c329
# 2222
```

```
c329

Enter the data in Hex for maximum length 512, or -1 to cancel.
2222
HEX: 22, 22,
```

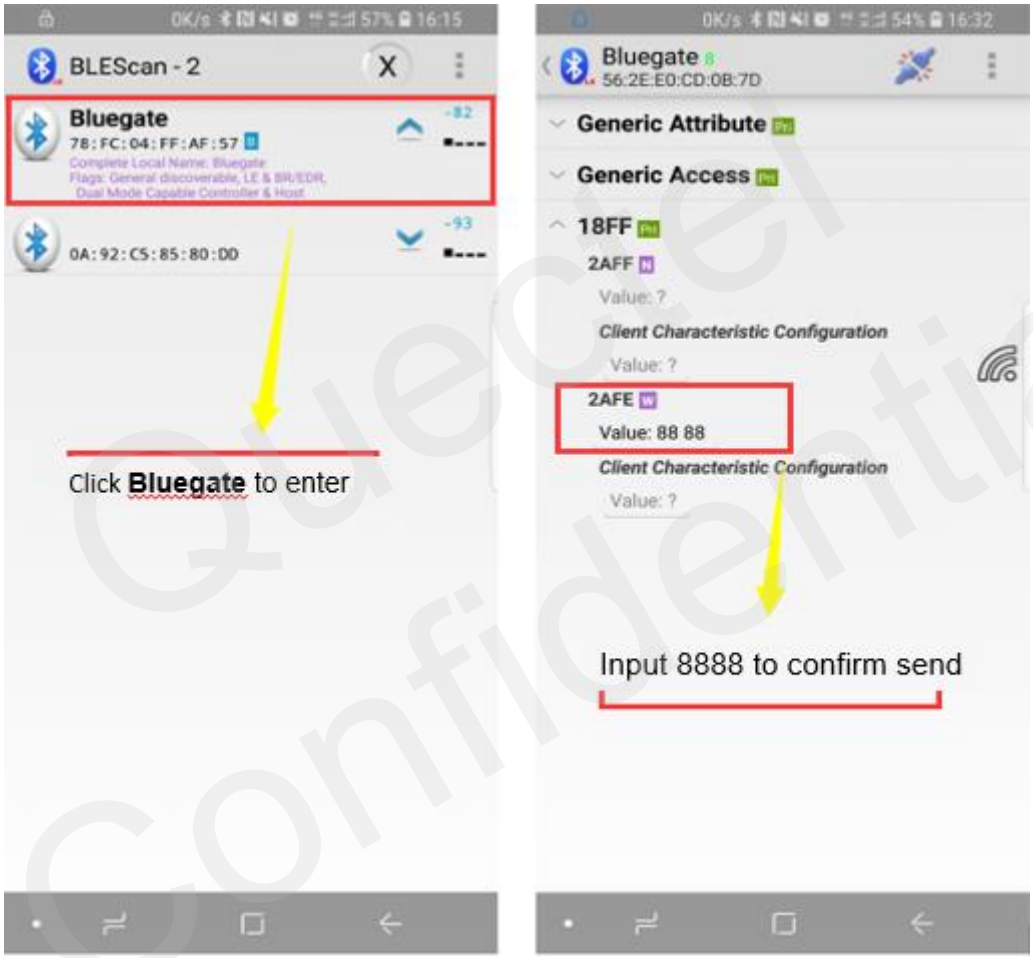
Please refer to the screenshot below, users can see that value=2222 will be received by mobile APP.

6.6. BLE Data Transmission(central -> peripheral)

The current test indicates that the data is sent from the mobile to the module terminal.
The following is to open BLE operation by inputting an instruction ("->" means the next step)

```
# c345 -> 0
# c328
# c311 -> 1 -> 1
```

Please refer to the picture below, select **"Bluegate"** in the left figure and then enter pattern on the right.



After input 8888 to confirm the transmission, users can see that the module terminal received the data=8888 from the log of the console.

```
Gatt Request write event :  [5F:3D:88:BC:85:2D], trans_id=1, attr_handle=45, offset=0, is_prep=0, length=2, need_rsp=1
                         : data=8888
Wake BT controller right now!

Allow BT controller into sleep
```

6.7. BT Pairing and Testing

Play music test: Firstly input command in command line: **amix 'SEC_AUX_PCM_RX Audio Mixer MultiMedia1' 1**
Run demo_UI, the user can input "help" to check the operation command

```
********************************************** Help **********************************************
bt_enable(c1)           : Enable BT stack.
bt_disable(c2)          : Disable BT stack.
localinfos(c3)          : Get local device infos.
setname(c4)             : Set local device name.
setscan(c5)             : Set scan mode.
scan(c6)                : Scan nearby bluetooth devices.
stop(c7)                : Stop scan nearby bluetooth devices.
pairlist(c8)            : List paired remote device.
pair(c9)                : Pair with remote device.
delete(c10)             : Delete paired remote device.
connect(c13)            : Connect to paired remote device.
disconnect(c14)         : Disconnect to paired remote device.
switch(c15)             : Switching current remote device.
setprofile(c16)         : set enable profile
getprofile(c17)         : get enable profile
address(c18)            : get bluetooth address of local device
********************************************** HFP **********************************************
```

Please refer to figure above, input pair or "c9" in command line of the console and then input the Bluetooth MAC address of the mobile phone, click to confirm on the mobile phone to complete the pairing, and input the character "y" in the console to complete the final pairing.

EVB needs to plug in the codec and headphone, and then turn switch to the CODEC mode, the users play music by mobile phone, only like this, music can be heard by EVB headphones.

Phone test: As shown above, turn switch to BT mode. Before running demo_UI, input the following voice recording command in the command line:

```
# amix 'MultiMedia1 Mixer AUX_PCM_UL_TX' 1
# arec -C 1 -R 8000 /data/record.wav &
# /data/demo_UI
```

The next step is to conduct Bluetooth pairing connection, and dials the 10086 for testing by the mobile phone.

The telephone test point is mainly to record the downlink voice. After the phone is powered off, the user can play */data/record.wav* to test whether the voice recording of the phone is successful.

```
# aplay /data/record.wav
```

PS: Since the recording operation is performed before the demo_UI running, the recorded audio will be silent at the beginning.

7 Sleep Wake-up Function Test

Plug in the Bluetooth antenna before testing to ensure good signal quality.

7.1. NF3303 Wake-up EC2x/AG35 Test

Unplug the USB cable and execute the demo program. After the initialization is completed, the autosleep function is enabled due to the demo program. At this time, the EC2x/AG35 will enter the sleep mode, and the Serial port terminal can no longer perform data interaction. At this point, the phone can still scan the NF3303 Bluetooth module. When the mobile phone initiates a connection with the Bluetooth module, the Bluetooth module will pull up the BT_HOST_WAKE pin to generate a rising edge interruption, which lead to wake up EC2x/AG35, and the client application program holds wakelock . Once the data interaction is completed, the Bluetooth module will pull low the BT_HOST_WAKE pin to generate a falling edge interruption, and the client application program releases wakelock that is held. If no other tasks hold the wakelock, the EC2x/AG35 will enter sleep mode and Serial port terminal can not perform data interaction.

7.2. EC2xAG35 Wake-up NF3303 Test

When there is no data interaction, the Bluetooth protocol stack generates a callback function notification. After receiving the notification, the customer application program will pull low the BT_DEVWAKE pin to allow the Bluetooth module to sleep. When EC2x or AG35 wants to interact data with the Bluetooth module, the Bluetooth protocol stack generates a callback function notification. After receiving the notification, the client application program will pull up the BT_DEVWAKE pin to wake up the Bluetooth module. user can use the oscilloscope to monitor the level change of the BT_DEVWAKE pin and test the Bluetooth module current consumption in real time to verify whether the Bluetooth module sleep and wake up normally.

8 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]	SA-HRD-211 NF3303 Module Hardware Specification V1.1	NF3303 WIFI/BT Module Spec
[4]	NFore Bluegate Programming Guide 0.10.4	Bluegate Stack API Programming Guide

Table 4: Terms and Abbreviations

Abbreviation	Description
BLE	Bluetooth Low Energy
EVB	Edge Virtual Bridging
MAC	Media Access Control
USB	Universal Serial Bus
APP	Application
IOS	IPhone Operating System
SDK	Software Development Kit
HCI	Human-Computer Interaction
ADB	Android Debug Bridge
API	Application Program Interface
GPIO	General-Purpose Input/Output)
SPI	Serial Peripheral Interface

WIFI	Wireless-Fidelity
UART	Universal Asynchronous Receiver/Transmitter
RXD	Receive External Data
RXD	Receive External Data
TXD	Transmit Data
UI	User Interface