Bluetooth communication protocol

Applicable instrument: **HW8076**

Table of contents

Send the upgrade package information

Send the upgrade file

CRC

Data encryption

Encryption

Encrypted byte description

Modify the record

Protocol version	Date	Modify the content
V1.0	**2022-01-17**	First release
V1.1	**2022-02-25**	Added VCU upgrade, added AT commands related to changing passwords, and changed the SN code acquisition method
V1.2	**2022-09-13**	Modified cruise control speed limits
V1.3	**2022-10-13**	Modified the software version number on page 11 and the software version number on page 21 BYTE4 and BYTE5
V1.4	**2023-01-30**	Modified the MCU upgrade protocol
V1.5	**2023-03-04**	Added a MAC address acquisition interface

Glossary

Acronym	Definition
BMS	**Battery Management System**
MCU	**Motor Controller Uint**
RPM	**Revolution Per Minute**
SN	**Serial Number**
VCU	**Vehicle Control Unit**
CMD	**Command**

Instruction set

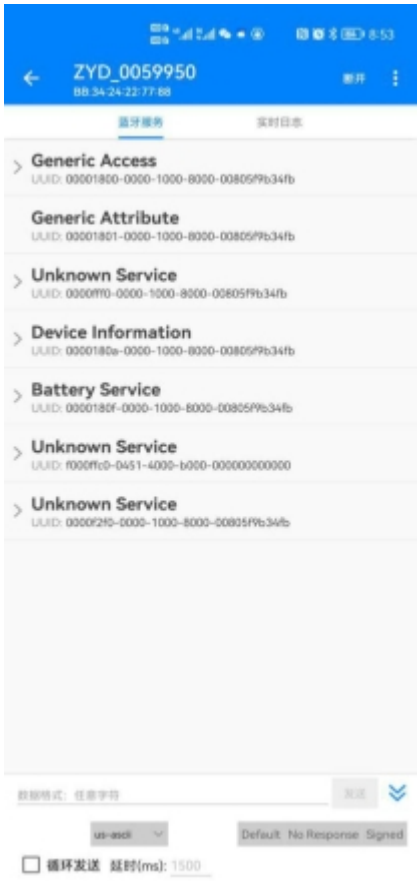
Script	Description	Note:
0xA0	Read the scooter operation data instructions	**The app does not read the** meter will also be automatically uploaded
0xA1	Read battery data instructions	**The app does not read the** meter will also be automatically uploaded

Script	Description	Note:
0xB0	Read version information and model instructions	Meters are not automatically uploaded

Script	Description	Note:
0xC0	**APP writes instructions**	Set function switch signals, speed limit values, etc

Script	Description	Note:
0xD0	Query if the controller allows upgrades	
0xD1	Erase the controller flash command	
0xD2	Upgrade data	
0xD3	The upgrade is complete	

1. Bluetooth service UUID



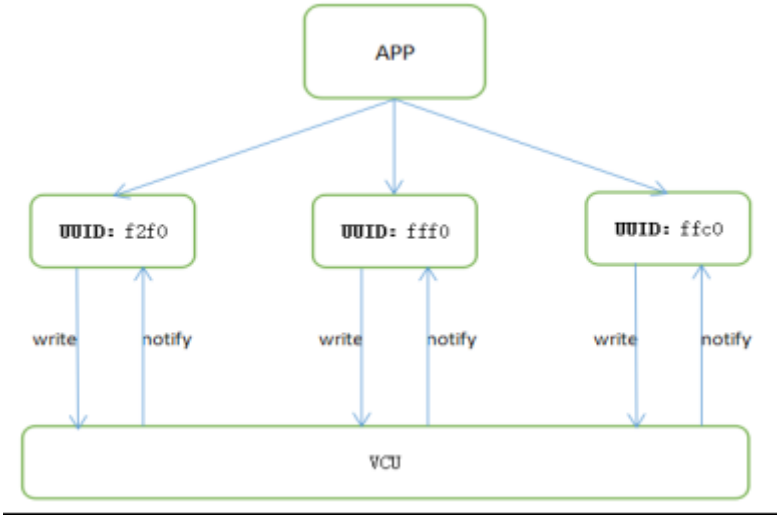
The above picture shows all the service channels of our Bluetooth chip, and in this project, we need to use 3 of them in addition to Device Information, namely:

UUID：0000fff0-0000-1000-8000-00805f9b34fb

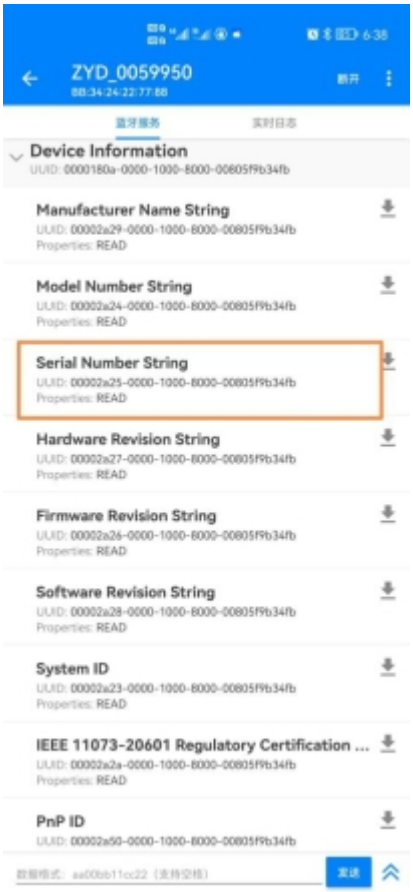
UUID：0000f2f0-0000-1000-8000-00805f9b34fb

UUID：f000ffc0-0451-4000-b000-000000000000

The communication framework diagram is as follows:

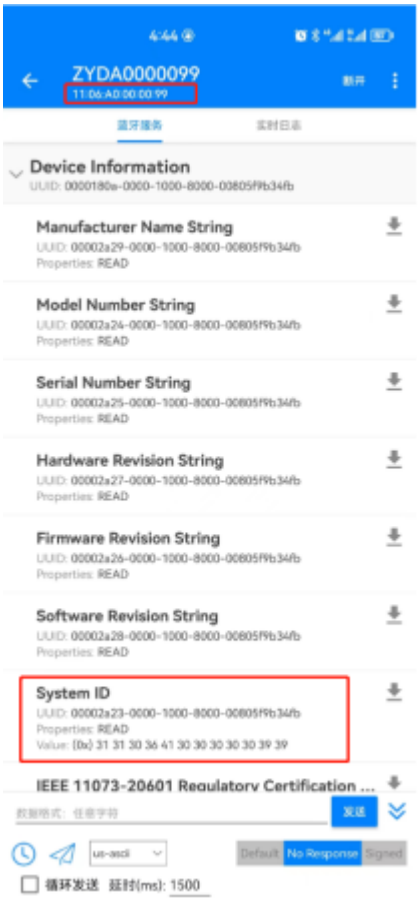


2.Device Information服务



SN码通过Device Information的Serial Number String获取

SN code is in ascii code format, 13 characters



MAC通过Device Information的System ID获取

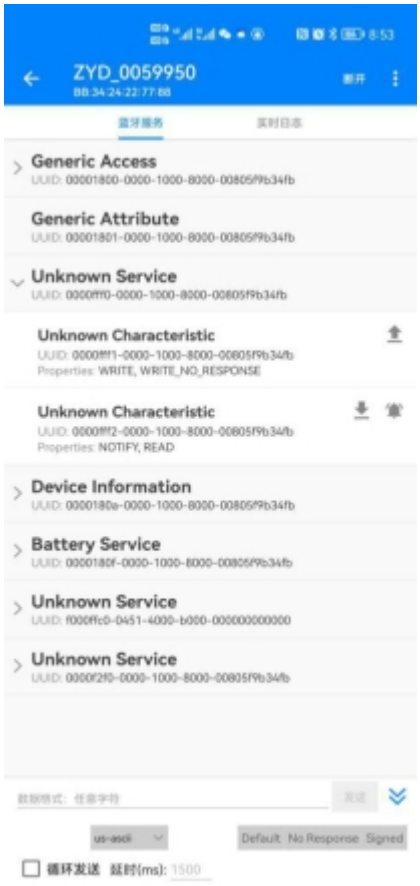
MAC is in ascii code format, 12 characters

3. Transparent service

透传服务UUID：0000fff0-0000-1000-8000-00805f9b34fb

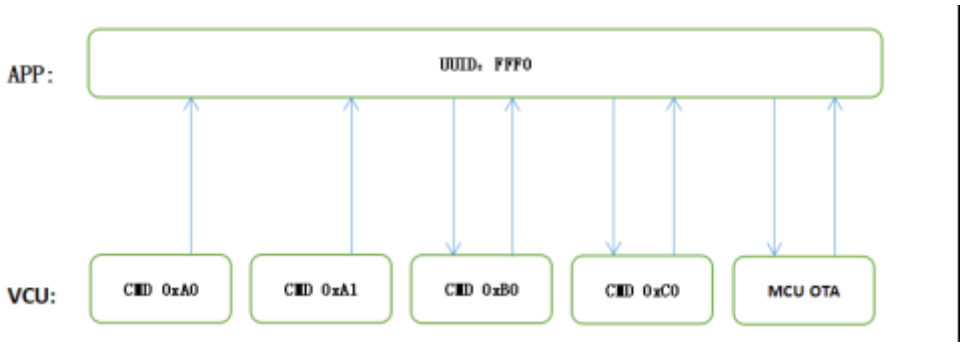
透传接收UUID：0000fff1-0000-1000-8000-00805f9b34fb（write）

透传发送UUID:0000fff2-0000-1000-8000-00805f9b34fb(notify)



The app can obtain scooter operation data, BMS data, control scooter status and upgrade controllers through this service.

The frame diagram is as follows:



The app reads the scooter operation data script 0xA0

The app does not need to send this command to read the data, because the meter will actively upload this frame data at regular intervals.

Example:

Write: F0 A0 05 C9 F0

Notify: F0 A0 19 00 00 00 00 19 10 19 0A 00 00 10 00 C8 19 19 00 C8 00 00 00 03 4F

The parse of each byte looks like this:

app send:

Serial number	Byte definition		Example
0	Baotou		0XF0
1	Script		0xA0
2	Total number of bytes in the frame		0x05
3	CRC16 low bytes		0xC9
4	CRC16 High Bytes		0xF0

Meter reply:(This package meter will be actively uploaded to the app at a scheduled time of 200ms).

Serial number	Byte definition	Example	Content definition	Note:
0	Baotou	0XF0		
1	Script	0xA0		
2	Total number of bytes in the frame	0x19		0x19=25
3	Fault code high bytes	0x00		See definitions below
4	Fault code low byte	0x00		See definitions below
5	Control function high bytes	0x00		See definitions below
6	Control function low byte	0x00		See definitions below
7	Cruise control speed setting	0x19	5-25km/h can enter the fixed speed cycle	0x19=25km/h
8	Current speed	0x10		0x10=16km/h
9	Maximum speed limit	0x19	Maximum speed 25km/h	0x19=25km/h
10	Subtotal mileage	0x0A	Mileage of this ride	0x0A=10km
11	Total mileage is high bytes	0x00		Km
12	Total mileage is low bytes	0x00		Km

13	Remaining range	0x10	Based on the power estimate	0x10=16km
14	The current limit value is high bytes	0x00	Unit 0.1A	0x00C8=200 represents 20A
15	The current limit value is low bytes	0xC8		
16	Current motor temperature	0x19	Range 0-255° C	25℃
17	Current controller temperature	0x19	Range 0-255° C	25℃
18	The current motor speed is high in bytes	0x00	Units RPM	200RPM
19	The current motor speed is low bytes	0xC8		
20	Retained	0x00		
21	Retained	0x00		
22	Retained	0x00		
23	CRC16 low bytes	CRCL		MODBUS——CRC16
24	CRC16 High Bytes	CRCH		

Control function byte definition:

Bit7-Bit6	Bit5	Bit4	Bit3-Bit2	Bit1-Bit0
Turn signals: 0x0: Off 0x1: Turn right 0x2: Turn left	Cruise control switch: 0: Off 1: On	Headlights: 0: Off 1: On	Retained	Gear: 0x0: 1 gear 0x1: 2 gears 0x2: Gear 3

Bit15-Bit11	Bit10	Bit9	Bit8
Retained	Boot mode: 0: Glide start 1: Zero start	Unit: 0:Km 1:Mile	Locked status: 0: Unlocked 1: Locked

Trouble code byte definition:

Fault code	Description
Bit0	
Bit1	0: Normal 1: Brake finger derailment is faulty
Bit2	0: Normal 1: Throttle finger flip fault

Bit3	0: Normal 1: Instrument and controller communication failure
Bit4	0: Normal 1: Controller overcurrent
Bit5	Retained
Bit6	Retained
Bit7	0: Normal 1: Hall fault
Bit8	Retained
Bit9	0: Normal 1: Op amp bias fault
Bit10-bit15	To be defined

E0: Motor Temp Out of RangeE1: Brake FaultE2: Throttle FaultE3: Communication FaultE4 Overcurrent FaultE5: Undervoltage FaultE6: Overvoltage Fault

E7: Motor Hall FaultE8: Motor Phase Fault
E9: MCU Over TemperatureE10: MCU Temp Out of RangeE11: Motor Over TemperatureE12: Bluetooth FaultE13: Battery Fault

The app reads the scooter BMS data script 0xA1

The app does not need to send this command to read the data, because the meter will actively upload this frame data at regular intervals.

Example:

Write: F0 A1 05 C8 60

Notify: F0 A1 19 00 00 00 0A 01 A4 64 00 10 0F A0 0F A0 19 00 00 00 00 00 00 A2 C0

The parse of each byte looks like this:

app send:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xA1
2	Total number of bytes in the frame	0x05
3	CRC16 low bytes	0xC8
4	CRC16 High Bytes	0x60

Meter reply: (This package meter will be actively uploaded to the app at a scheduled time of 200ms).

Serial number	Byte definition	Example	Content definition	Note:
0	Baotou	0XF0		
1	Script	0xA1		
2	Total number of bytes in the frame	0x19		
3	The battery status is high bytes	0x00		See definitions below
4	Battery status low bytes	0x00		
5	The battery current is high in bytes	0x00	Symbolized unit 0.1A	0x000A = 10 1.0A
6	Low battery current	0x0A		
7	The current voltage of the battery is high bytes	0x01	Unit 0.1V	0x01A4 = 420 42.0V
8	The current voltage of the battery is low bytes	0xA4		
9	Current battery level	0x64	Unit 1%	100%
10	The number of charging cycles is high	0x00		0x0010 = 16 times
11	The number of charging cycles is low bytes	0x10		
12	The battery capacity is high bytes	0x0F	Unit mAh	
13	Battery capacity is low bytes	0xA0		
14	The remaining capacity is high bytes	0x0F	Unit mAh	
15	Low bytes of remaining capacity	0xA0		
16	Battery temperature	0x19	There is a sign unit of ° C	0x19 = 25 25℃
17	Retained	0x00		
18	Retained	0x00		
19	Retained	0x00		
20	Retained	0x00		

21	Retained	0x00		
22	Retained	0x00		
23	CRC16 low bytes	CRCL		MODBUS——CRC16
24	CRC16 High Bytes	CRCH		

Note: Instruction 0xA#packet app can read without issuing commands The meter will automatically poll and upload these data

Battery status byte definition:

bit15-bit1	bit0
Retained	Charging flag: 1 - Charged 0 - Not charged
Bit32-bit16	
Retained	

The mobile phone sends the command code to read the skateboard information 0xB0

Example:

Write: F0 B0 05 C4 30

Notify: F0 B0 20 00 01 30 30 30 30 30 30 30 30 30 30 30 30 00 00 00 10 10 20 20 30 30 00 00 00 F6 9E

The parse of each byte looks like this:

app send:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xB0
2	Total number of bytes in the frame	0x05
3	CRC16 low bytes	0xC4
4	CRC16 High Bytes	0x30

Meter reply:

Serial number	Byte definition	Example	Content definition	Note:
0	Baotou	0XF0		

1	Script	0xB0		
2	Total number of bytes in the frame	0x19		0x19=25
3	The model is high bytes	0x00	Model 0x0001=S008 Base	
4	Model low byte	0x01		
5	Instrument hardware version	0x10	Indicates version 1.0	V1.0
6	Instrument software version	0x22	Indicates version 2.2	V2.2
7	Controller hardware version	0x20	Indicates version 2.0	V2.0
8	Controller software version	0x20	Indicates version 2.0	V2.0
9	BMS hardware version	0x30	Indicates version 3.0	V3.0
10	BMS software version	0x30	Indicates version 3.0	V3.0
11	Retained	0x00		
12	Retained	0x00		
13	Retained	0x00		
14	Retained	0x00		
15	Retained	0x00		
16	Retained	0x00		
17	Retained	0x00		
18	Retained	0x00		
19	Retained	0x00		
20	Retained	0x00		
21	Retained	0x00		
22	Retained	0x00		
23	CRC16 low bytes	CRCL		
24	CRC16 High Bytes	CRCH		

The APP issues the control data script 0xC0

The app can be used to control the scooter gear, headlight switch, cruise control switch, lock status, unit, start mode, speed limit value, etc.

Example:

Write: F0 C0 0F 00 00 19 19 00 00 00 00 00 00 39 E8

Notify: F0 C0 06 00 30 B8

The parse of each byte looks like this:

app send:

Serial number	Byte definition	Example	Content definition	Note:
0	Baotou	0XF0		
1	Script	0xC0		
2	Total number of bytes in the frame	0x0F		0x0A=10
3	Control function high bytes	0x00		
4	Control function low byte	0x00		
5	Cruise control speed setting	0x05	Speed>=5km/h can enter the fixed speed cycle	0x05=5km/h
6	Maximum speed limit	0x19	Maximum speed 25km/h	0x19=25km/h
7	Retained	0x00		
8	Retained	0x00		
9	Retained	0x00		
10	Retained	0x00		
11	Retained	0x00		
12	Retained	0x00		
13	CRC16 low bytes	CRCL		MODBUS——CRC16
14	CRC16 High Bytes	CRCH		

Data Reception Success Meter Response:

Serial number	Byte definition	Example

0	Baotou	0XF0
1	Script	0xC0
2	Total number of bytes in the frame	0x06
3	State bytes	0x00
4	CRC16 low bytes	CRCL
5	CRC16 High Bytes	CRCH

Status byte: 0x00--data received successful 0x01--CRC check-error

Control function byte definition:

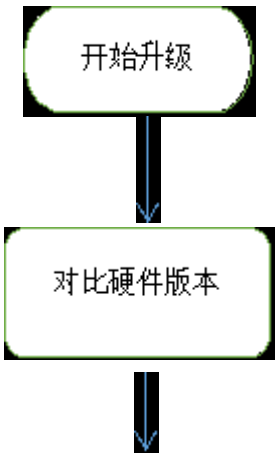
Bit7-Bit6	Bit5	Bit4	Bit3-Bit2	Bit1-Bit0
	Cruise control switch: 0: Off 1: On	Headlights: 0: Off 1: On	Retained	Gear: 0x0: 1 gear 0x1: 2 gears 0x2: Gear 3

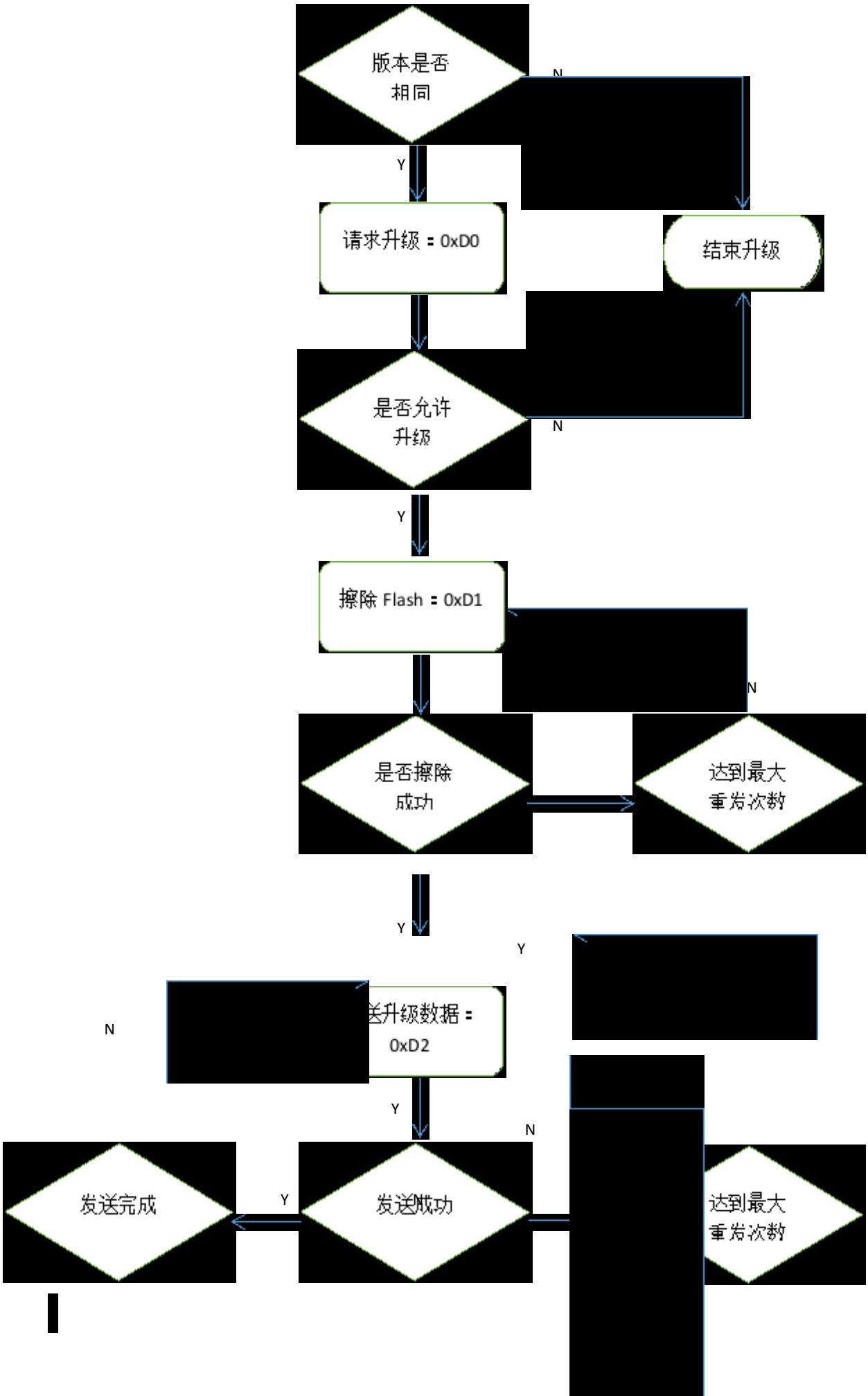
Bit15-Bit11	Bit10	Bit9	Bit8
Retained	Boot mode: 0: Glide start 1: Zero start	Unit: 0:Km 1:Mile	Locked status: 0: Unlocked 1: Locked

Controller MCU upgrade

During the upgrade process, if the instruction does not reply in 1s, resend it, and the upgrade is considered to have failed if the number of sends exceeds 5 times

MCU upgrade flow chart:




版本是否
相同
N
Y
请求升级 = 0xD0
结束升级
是否允许
升级
N
Y
擦除 Flash = 0xD1
N
是否擦除
成功
达到最大
重发次数
Y
Y
N
送升级数据 =
0xD2
Y
N
发送完成
Y
发送成功
达到最大
重发次数

Before upgrading, you need to compare the hardware version of the MCU with the hardware version of the upgrade file, and the hardware version can only be upgraded if the hardware version is the same.

The maximum number of reoccurrences was 5 times, with a time interval of 1s.

Request an upgrade script 0xD0

App sends:

Serial number	Byte definition	
0	Baotou	0xF0
1	Script	0xD0
2	CRC16 low bytes	0x44
3	CRC16 High Bytes	0x2C

Meter reply:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD0
2	State bytes	0x00

3	CRC16 low bytes	CRCL
4	CRC16 High Bytes	CRCH

Status bytes: 0x00 - Upgrade allowed 0x01 - Upgrade not allowed

擦除MCU flash指令码 0xD1

App send:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD1
2	CRC16 low bytes	0x85
3	CRC16 High Bytes	0xEC

Meter reply:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD1
2	State bytes	0x00
3	CRC16 low bytes	CRCL
4	CRC16 High Bytes	CRCH

Status bytes: 0x00--erase succeeded 0x01--erase failed

Send the upgrade data script 0xD2

The MCU upgrade file size is generally more than 40K, and the app needs to split the upgrade package and send it to the VCU in the following format.

The upgrade data is split into 128 bytes per pack.

App send:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD2
2	The package number is high bytes	0x##

3	The package number is low byte	0x##
4	The number of bytes of data is high	0x00
5	The number of bytes of data is low	0x80
6	Data 0	0x##
7	Data 1	0x##
8	Data 2	0x##
…	…	…
…	…	…
132	Data 126	0x##
133	Data 127	0x##
134	CRC16 low bytes	CRCL
135	CRC16 High Bytes	CRCH

Meter reply:

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD2
2	The package number is high bytes	0x##
3	The package number is low byte	0x##
4	State bytes	0x00
5	CRC16 low bytes	CRCL
6	CRC16 High Bytes	CRCH

The package number is the current package number received

State Bytes:

0x00--data received successfully 0x01-CRC error

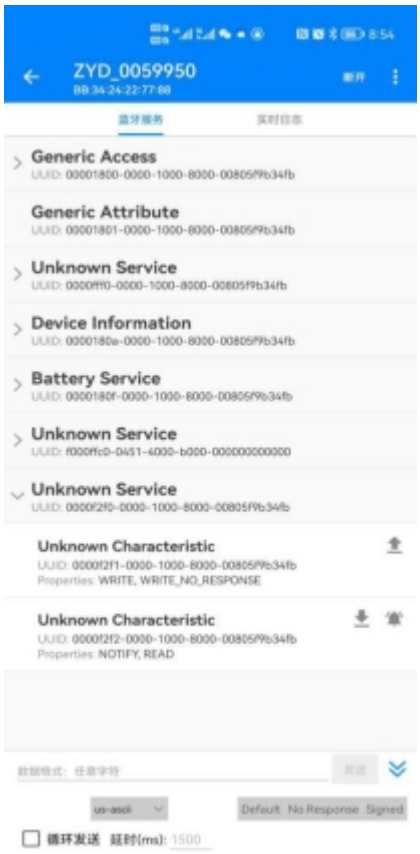
The upgrade is completed 0xD3 the script

Serial number	Byte definition	Example
0	Baotou	0XF0
1	Script	0xD3
2	CRC16 low bytes	0x04
3	CRC16 High Bytes	0x2D

The command meter did not reply, and the instruction was sent once after all upgrade files were sent for 1 sent.

4.AT Instruction Service

AT服务 UUID：0000f2f0-0000-1000-8000-00805f9b34fb
AT接收 UUID：0000f2f1-0000-1000-8000-00805f9b34fb（write）
AT发送 UUID：0000f2f2-0000-1000-8000-00805f9b34fb（notify）



AT commands are used to set some parameters of Bluetooth devices, which are sent and received in ASCII code format

Bluetooth Broadcast Name Query and Settings

directive	Answer	Parameters
Query: AT+NAME?	OK+NAME:Para	Para: Module name Up to 11 characters are allowed, including letters, numbers, underscores 默认 To=HW_ZAxx
设置:AT+NAME[For]	OK+NAME:Para	

Set/query encryption mode

directive	Answer	Parameters
Enquiry: AT+TYPE?	OK+TYPE:Para	Para: encryption mode A: No verification required B: Application-layer password encryption Default: A
设置:AT+TYPE[For]	OK+TYPE:Para	

Set/look up passwords

directive	Answer	Parameters
验证密码:AT+PWD[For]	验证成功:OK+PWD:Y 验证失败:OK+PWD:N	Para: Password6 digits Default: 888888
设置密码:AT+PWDM[For]	回复 OK+PWDM:Para	

Example (without encryption):

Check the Bluetooth broadcast name:

The command format uses ASCII code to send and receive.

Write: AT+NAME?

Notify: OK+NAME:ZYD_0059950

Modify Bluetooth Broadcast Name:

The command format uses ASCII code to send and receive.

Write: AT+NAME[AAABBBCCCDD]

Notify: OK+NAME:AAABBBCCCDD

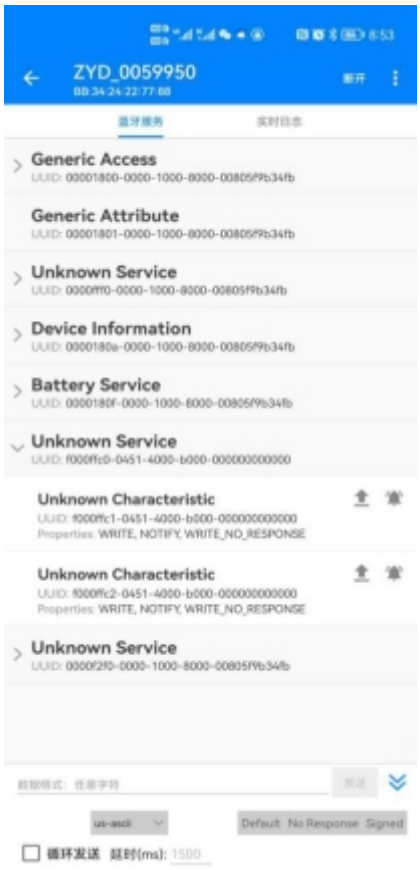
Successful modification:

5. VCU upgrade service

Upgrade the service UUID:f000ffc0-0451-4000-b000-000000000000

Upgrade to receive UUID:f000ffc1-0451-4000-b000-000000000000

Upgrade to send UUID:f000ffc2-0451-4000-b000-000000000000



Meter upgrades through the service

The upgrade file is a .bin file, where the first 16 bytes of the bin file are the upgrade firmware information, and the first 16 bytes are the upgrade data.

The following figure shows the first part of the bin file:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	6F	3C	D4	A2	03	01	B4	22	42	42	42	42	FF	FF	12	00
0010	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5
0020	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5	18	F0	9F	E5
0030	9D	64	84	E2	01	00	40	E2	01	00	44	E2	01	00	48	E2
0040	01	00	4C	E2	01	00	54	E2	01	00	50	E2	01	00	58	E2

The first 16 bytes are parsed as follows (low bytes first):

Byte0-Byte3 upgrades the checksum bytes

Byte4-Byte5 待升级vcu版本号

Data 0x00 0x22 corresponding version number V2.2

Byte6-Byte7 upgrade file length in words (4 bytes).

Data 0xB4 0x22 Corresponding length 0x22B4

byte8-byte11 升级类型

Data 0x42 0x42 0x42 0x42 Upgrade Meter Program

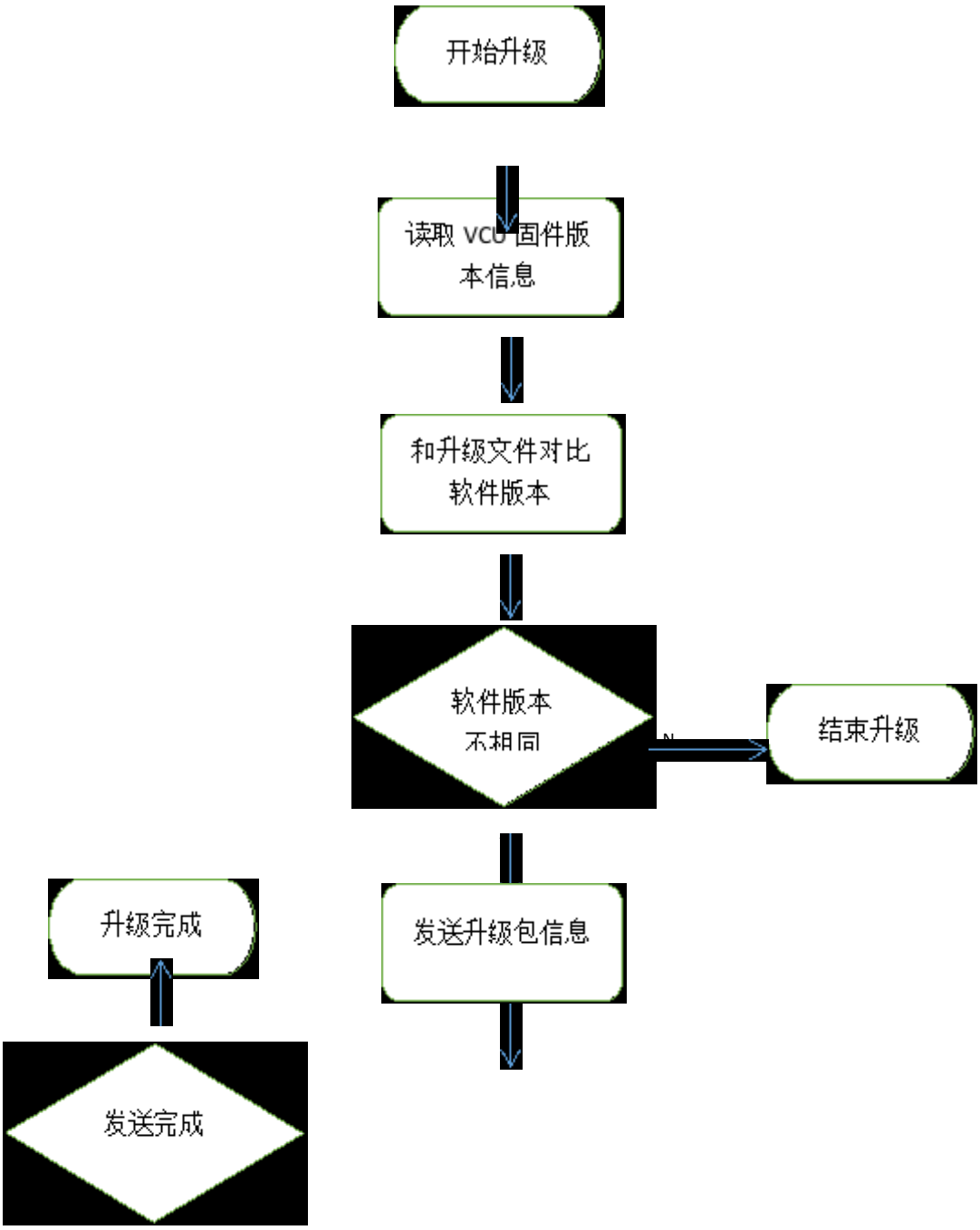
Byte12-Byte13 Reserved

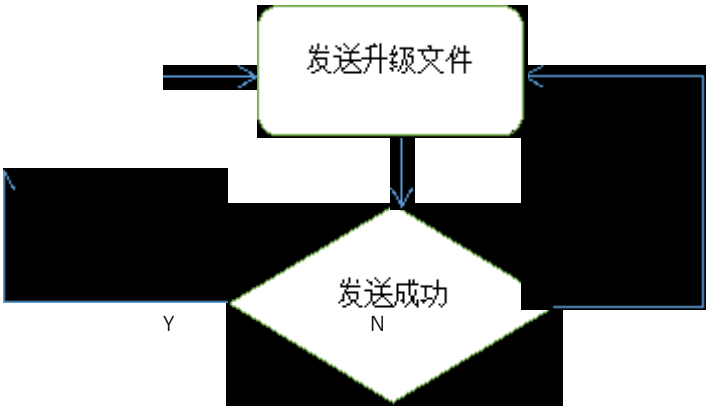
byte14-byte15 蓝牙协议栈版本

The data 0x12 0x00 corresponding version number V0.12

The app needs to check whether the version number of the instrument to be upgraded is different from the current version number of the instrument.

Instrument upgrade flow chart:





Read VCU firmware version information

App发送: UUID:f000ffc1-0451-4000-b000-00000000000000

Serial number	Byte definition	Example
0	directive	0x00

VCU回复: UUID：f000ffc1-0451-4000-b000-000000000000

Serial number	Byte definition	Example
0	VCU version number is 8 digits lower	0x1A
1	The VCU version number is 8 digits higher	0x00
2	The firmware package length is low bytes	0xB4
3	The length of the firmware package is high bytes	0x22
4	Uid0	0x42
5	Uid1	0x42
6	Uid2	0x42
7	Uid3	0x42
8	The BLE stack version is 8 bits lower	0x12
9	The BLE stack version is 8 bits higher	0x00

The app can only use the VCU version, and you don't need to pay attention to other data.

Send the upgrade package information

The content sent is the first 16 bytes of the upgrade package

App发送：f000ffc1-0451-4000-b000-000000000000

Serial number	Byte definition	Example
0	Firmware package verification bytes	0X6F
1	Firmware package verification bytes	0x3C
2	Firmware package verification bytes	0xD4
3	Firmware package verification bytes	0xA2
4	VCU version number is 8 digits lower	0x03
5	The VCU version number is 8 digits higher	0x01
6	The firmware package length is low bytes	0xB4
7	The length of the firmware package is high bytes	0x22
8	Uid0	0x42
9	Uid1	0x42
10	Uid2	0x42
11	Uid3	0x42
12	Retained	0xFF
13	Retained	0xFF
14	The BLE stack version is 8 bits lower	0x12
15	The BLE stack version is 8 bits higher	0x00

VCU回复： f000ffc2-0451-4000-b000-000000000000

Serial number	Byte definition	Example
0	The package number is 8 digits lower	0x00
1	The package number is 8 digits higher	0x00

It should be noted that the VCU reply is via FFC2

Send the upgrade file

App发送：f000ffc2-0451-4000-b000-000000000000

The app needs to split the entire upgrade file into 16 bytes per package and send it to VCU, and the package number needs to be automatically added by 1, and the app can be sent at intervals of 10ms per package

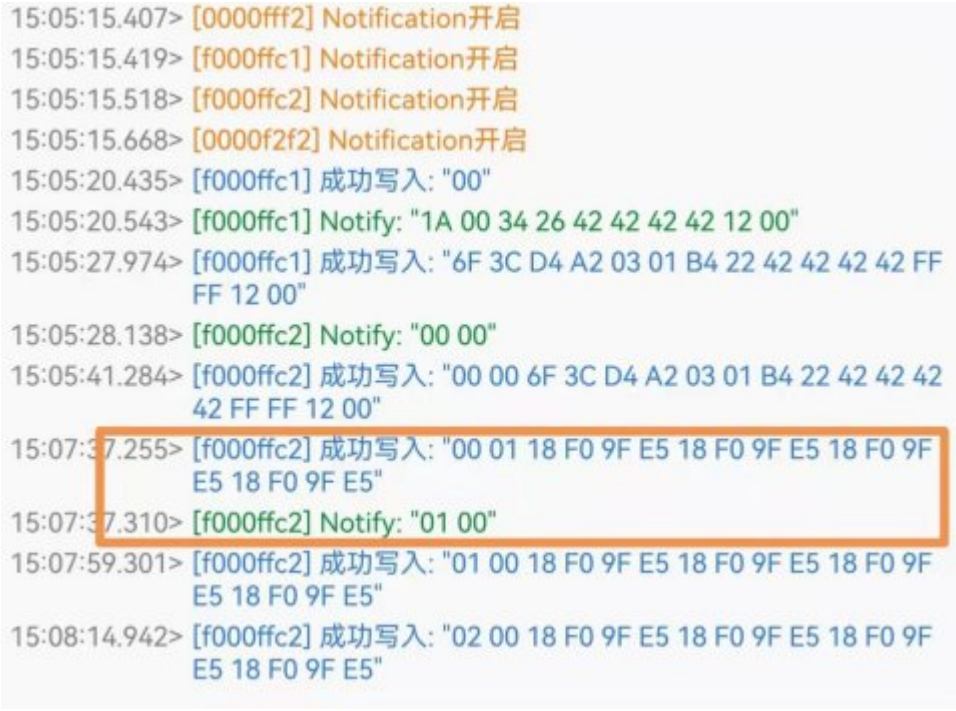
Serial number	Byte definition	Example
0	The package number is 8 digits lower	0x00
1	The package number is 8 digits higher	0x00
2-17	Upgrade data 16 bytes	

Vcu回复： f000ffc2-0451-4000-b000-000000000000

Only the VCU will receive an error in the data will reply The reply content is the package number, and the app needs to continue sending it down from the location of the received package number after receiving the reply

Serial number	Byte definition	Example
0	The package number is 8 digits lower	0x##
1	The package number is 8 digits higher	0x##

Example (without encryption):



If the app in the box is in the wrong format, the VCU will reply to the package number and remind the app that it needs data with the package number 0x01 0x00.

CRC

The CRC calculation range includes all bytes except CRC bytes, and the CRC calculation can directly call CalculateCRC16(), and the return value is the calculated CRC value.

```
const unsigned char CRCH[] =
{
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
    0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,
    0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
    0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,
    0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,
    0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
    0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
```

```
        0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
        0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
        0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
        0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
        0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
        0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
        0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
        0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
        0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
        0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
        0x40
};

const unsigned char CRCL[] =
{
        0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,
        0x04,0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,
        0x08,0xC8,0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,
        0x1D,0x1C,0xDC,0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,
        0x11,0xD1,0xD0,0x10,0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,
        0x37,0xF5,0x35,0x34,0xF4,0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,
        0x3B,0xFB,0x39,0xF9,0xF8,0x38,0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,
        0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,
        0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,
        0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,
        0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,0x78,0xB8,0xB9,0x79,0xBB,
        0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,0xB4,0x74,0x75,0xB5,
        0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,0x50,0x90,0x91,
        0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,0x9C,0x5C,
        0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,0x88,
        0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
        0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,
        0x40
};
/*********************************************************************
* *msgPtr needs to calculate the first address of the CRC data
* msgLen needs to calculate the length of the CRC data
```

```
* return calculated CRC value
*************************************************************************/
unsigned short int  CalculateCRC16(unsigned char  *msgPtr, unsigned short int msgLen)
{
   unsigned char crcHigh = 0xFF; //high byte of CRC initialized
   unsigned char crcLow = 0xFF; //low byte of CRC initialized
   unsigned char index; //will index into CRC lookup table

   while (msgLen--) //pass through message buffer
   {
      index = crcLow ^ (*(msgPtr++)); //calculate the CRC
      crcLow = crcHigh ^ CRCH[index];
      crcHigh = CRCL[index];
   }
   return (unsigned short int)((unsigned short int)(crcHigh<<8) | crcLow);
}
```

Data encryption

Encryption

We use encryption similar to single-table substitution, replacing 0-F with random numbers.

For example:

Key[16]={0x02,0x03,0x07,0x09,0x0A,0x01,0x04,0x08,0x0F,0x06,0x0D,0x0C,0x05,0x0B,0x00,0x0E};

0->2 1->3 2->7 3->9 4->A 5->1 6->4 7->8 8->F 9->6 A->D B->C C->5 D->B E->0 F->E

明文：tab[3] = {0x0F,0x12,0xE3}

密文：tab[3] = {0x2E,0x37,0x09}

For enhanced encryption, we employ 16 sets of keys

```
unsigned char const Key[16][16]={
{0x02,0x03,0x07,0x09,0x0A,0x01,0x04,0x08,0x0F,0x06,0x0D,0x0C,0x05,0x0B,0x00,0x0E},//0
{0x01,0x06,0x0A,0x05,0x07,0x02,0x0E,0x00,0x0C,0x03,0x0B,0x0F,0x09,0x0D,0x08,0x04},//1
{0x0F,0x03,0x09,0x07,0x0E,0x04,0x01,0x08,0x02,0x0D,0x06,0x00,0x05,0x0B,0x0C,0x0A},//2
{0x01,0x06,0x0F,0x09,0x0A,0x02,0x04,0x0D,0x07,0x03,0x08,0x00,0x05,0x0B,0x0C,0x0E},//3
{0x00,0x03,0x07,0x0E,0x08,0x01,0x0B,0x0A,0x0F,0x06,0x0D,0x0C,0x05,0x04,0x02,0x09},//4
{0x0F,0x00,0x0A,0x02,0x07,0x03,0x04,0x08,0x09,0x06,0x0C,0x0D,0x05,0x0B,0x01,0x0E},//5
{0x03,0x02,0x09,0x07,0x01,0x0A,0x0F,0x06,0x04,0x08,0x00,0x0C,0x0E,0x0B,0x0D,0x05},//6
{0x07,0x03,0x02,0x09,0x0F,0x0D,0x0A,0x08,0x0B,0x06,0x01,0x0C,0x05,0x04,0x00,0x0E},//7
{0x02,0x03,0x0B,0x01,0x0A,0x09,0x05,0x08,0x0C,0x07,0x0D,0x0F,0x04,0x06,0x0E,0x00},//8
{0x01,0x03,0x04,0x09,0x05,0x02,0x07,0x00,0x0E,0x06,0x0D,0x08,0x0A,0x0B,0x0C,0x0F},//9
{0x00,0x07,0x0D,0x0B,0x0A,0x06,0x04,0x08,0x0F,0x01,0x03,0x0C,0x05,0x0E,0x02,0x09},//A
{0x0F,0x03,0x0E,0x0B,0x06,0x08,0x04,0x01,0x02,0x0C,0x0D,0x0A,0x05,0x00,0x09,0x07},//B
{0x08,0x0C,0x07,0x0E,0x0A,0x01,0x06,0x02,0x0F,0x04,0x0D,0x03,0x00,0x0B,0x05,0x09},//C
{0x02,0x0E,0x0A,0x07,0x09,0x06,0x04,0x08,0x00,0x01,0x05,0x0C,0x0D,0x0B,0x0F,0x03},//D
{0x0E,0x00,0x0B,0x05,0x0C,0x0D,0x0F,0x06,0x04,0x08,0x01,0x0A,0x09,0x07,0x03,0x02},//E
{0x01,0x0F,0x03,0x07,0x06,0x00,0x0D,0x0C,0x02,0x09,0x0E,0x05,0x08,0x0A,0x0B,0x04},//F
};
```

Select the key by accumulating and lowering the four digits by Serial Number

For example, SN is "0123456789123"

秘钥为 Key[keytype][16]

Sum = 0x30+0x31+0x32+0x33+0x34+0x35+0x36+0x37+0x38+0x39+0x31+0x32+0x33

Sum = 0x02A3

keytype= Sum & 0x000F keytype= 0x0003

Ki[keytipe][16] = ki[3][16] =

{0x01,0x06,0x0F,0x09,0x0A,0x02,0x04,0x0D,0x07,0x03,0x08,0x00,0x05,0x0B,0x0C,0x0E},//3

/*Encrypted*/

```
/*****************************************************************************
*parameter *tabIn The data header address (plaintext) that needs to be encrypted
*Parameter *tabOutThe first address (ciphertext) of the encrypted data is stored
*Parameter len The length of the data that needs to be encrypted
*Parameter type key selection
*****************************************************************************/
void encrypt(unsigned char *tabIn,unsigned char *tabOut,unsigned char len,unsigned char type)
{
        unsigned char i=0;
        unsigned char codeL,codeH;

        for(i=0; i<len; i++)
        {
                codeL = tabIn[i]&0x0F;
                codeH = (tabIn[i]&0xF0)>>4;
                tabOut[i] = 0;
                tabOut[i] |= key[type][codeL];
                tabOut[i] |= (key[type][codeH]<<4);
        }

}
```

```
/*decrypt*/
/*****************************************************************************
*parameter *tabIn  The data header address (ciphertext) that needs to be decrypted
*parameter *tabOut: The first address of the decrypted data (plaintext).
*The parameter len requires the length of the data to be decrypted
*Parameter type key selection
*****************************************************************************/
void decrypt(unsigned char *tabIn,unsigned char *tabOut,unsigned char len,unsigned char type)
{
	unsigned char i=0,j=0;
	unsigned char codeL,codeH;

	for(i=0; i<len; i++)
	{
		codeL = tabIn[i]&0x0F;
		codeH = (tabIn[i]&0xF0)>>4;
		tabOut[i] = 0;
		for(j=0; j<16; j++)
		{
			if(codeL==key[type][j])
			{
				tabOut[i] |= j;
			}
			if(codeH==key[type][j])
			{
				tabOut[i] |=(j<<4);
			}
		}
	}
}
```

Example:

Send the mobile phone to send the command code to read the skateboard information:

Before encryption: F0 B0 05 C4 30

Encryption is used with the key [3][16]

After encryption: E1 01 12 5A 91

Encrypted byte description

1. Device Information的Serial Number 不用加密
2. When VCU is upgraded, the package number of the upgrade data is not encrypted
3. When the VCU is upgraded, the number of bytes sent or replied to is less than 3 bytes is not encrypted

In addition to the above points, all other data needs to be encrypted.