Type	For Customers
Date	August 3th, 2024
Version	Ver : 2.0.7
Dep	Engineer Dept.

Horseshoe Lock Device (Bike Version)

TCP +BLE Interface Protocol

Author:

Date:

Trial Nuclear:

Version Revision Record:

Version	Time	Change description	Author
V2.0.0	2021-01-22	Horseshoe Lock Protocol Second Edition	
V2.0.1	2021-01-28	D0 Command add response	
V2.0.2	2021-04-16	Modify W0 response	
V2.0.3	2022-02-16	Modify the U1 command	
V2.0.4	2022-03-16	Modify B0 command, add latitude and longitude	
V2.0.5	2023-05-04	L5 command add battery lock and helmet lock control command	
V2.0.6	2023-08-03	L5 command add OC33 lock cable status report and inquiry	
V2.0.7	2023-08-03	L5 command add OC33 lock cable status report and inquiry	

CONTENT

1.TCP Communication Protocol Description

1.1 Command format

Note: commands are in string form, each item is separated by',', and each command ends with a new line ('\n'). and When the server sends commands, it needs to add 0xFFFF before the command header, HEX form

<1> <2> <3> <4> <5> <6> <7> <8>		
0xFFFF*CMDS,OM,123456789123456,200318123020,XX,DDD#< New Line >		
Item	**Content**	**Note**
1	0xFFFF	Two bytes start bit, HEX form, (non-"0xFFFF "string), please refer to Appendix I
2	*CMDS	Command header server -> lock use *CMDS, lock -> server use *CMDR
3	OM	Manufacturer code
4	123456789123456	Lock unique ID number, using lock communication module IMEI number (15 bits)
5	200318123020	Time, format yyMMddHHmmss ,200318123020->2020-03-18 12：30：20
6	XX	Type of command
7	DDD	The command bikeries content items, which may have multiple items, separated by',',
8	#	The end of command
9		Command check value, *
10	#< New Line >	End of command #, end with newline'\n'

1.2 Command List

Serial number	command	command description
1		
2		
3		

1.3 Details of command

1.3.1 Q0 (check-in)

	<1>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,Q0,412#<LF>**
1	Lock current voltage, unit 0.01 V ,412->4.12 V (range :320-420) Reference for voltage and electricity percentage mapping list by Appendix II
Server -> lock	**No response**

1.3.2 H0 (heartbeat)

Note: Horseshoe locks are sent once every 4 minutes by default H0, to maintain TCP connections

	<1><2> <3>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,H0,0,412,28#<LF>**
1	Lock status 0-> unlock status 1-> lock status
2	Lock current voltage, unit 0.01 V,412->4.12 V (range :320-420) Reference for voltage and electricity percentage Appendix II
3	Current network signal value, the larger this value, the better the signal (range :2-32)
Server -> lock	**No response**

1.3.3 L0 (Unlock commands)

	<1> <2> <3>
Server -> lock	***CMDS ,OM,123456789123456,200318123020,L0,0,1234,1497689816#<LF>**
1	0->Reset the lock riding time when unlocking (lock internal timing) 1-> Do not reset the riding time when unlocked (for temporary parking, continue timing when unlocked again)
2	User ID (range :0-4294967295)
3	Current timestamp (timestamp accurate to second range :0-4294967295)
	<1> <2> <3>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,L0,0,1234,1497689816#<LF>**
1	Unlock result returns 0-> success 1-> failure
2	Customer ID(sent with server)
3	Operation timestamp (sent with server)
Server -> lock	***CMDS ,OM,123456789123456,200318123020,Re,L0#<LF>(increase server response to ensure server has received unlock return)**

1.3.4 L1 (Lock command, lock active report)

	<1> <2> <3>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,L1,1234,1497689816,3#<LF>**
1	User ID(Unlock command)
2	Operation timestamp (same unlock command)
3	Cycle time unit: minutes
Server -> lock	***CMDS ,OM,123456789123456,200318123020,Re ,L1#<LF>(increase server response to ensure server has received unlock return)**

1.3.5 D0 (Acquisition of positioning commands, single)

Server -> lock	***CMDS,OM,123456789123456,200318123020,D0#<LF>**
	<1> <2> <3> <4> <5> <6> <7><8> <9> <10> <11><12><13>
Lock -> server	***CMDR, OM, 123456789123456, 200318123020,D0, 0,124458.00,A,2237.7514,N,11408.6214, E,6, 0.21,151216,10, M,A#<LF>**
1	0： command acquisition location upload identification 1: location tracking upload location identification
2	UTC time, hhmmss(minutes) format
3	Location status, A= effective location, VA= invalid location
4	Latitude ddmm.mmmm (degrees) format (the previous 0 will also be transmitted)
5	Latitude hemisphere N(northern hemisphere) or S(southern hemisphere)
6	longitude dddmm.mmmm (degrees) format (the previous 0 will also be transmitted)
7	Longitude hemisphere E(E) or W(W)
8	Number of satellites searched
9	HDOP(positioning accuracy)
10	UTC date, ddmmyy(date,month,year) format
11	Altitude
12	Height unit M： m
13	Mode indication (A= autonomous location, D= difference, E= estimation, N= data invalid)
Server-> lock	***CMDS,OM,123456789123456,200318123020,Re,D0#<LF>** (Increase the server response to ensure that the server has received)

Note 1: After the horseshoe lock receives the D0 command, it takes 20-180 seconds to respond (the lock needs to obtain multiple location information to filter and calculate an accurate location). The lock uploads positioning information every 1 hour by default.

Note 2: Invalid location in a similar format may occur

*CMDR,OM,123456789123456,200318123020,D0,0,033724.00,V,,,,,,120517,,,N#

Note 3:

If the obtained latitude and longitude coordinates are the original coordinates, the coordinates converted into the degree format are the coordinates of the WGS84 coordinate system. This coordinate can be used directly on a map using a WGS84 coordinate system, such as a google map.

Latitude conversion algorithm: the lat=dd+mm.mmm/60. represents positive or negative according to N or S, where the N is positive and the S is negative.

Longitude conversion algorithm: lng=ddd+mm.mmmm /60. Represents positive or negative according to E or W, where E is positive and W is negative.

That is, the original coordinates of latitude obtained in the example of the above table are 2237.7514..

The lat= converted to WGS84 coordinates is 22+37.7514/60=22.62919, which is positive for the Northern Hemisphere.

The lng= converted to WGS84 coordinates is 114+08.6214/60=114.14369, and the E is positive.

Note that WGS84 coordinates can not be directly used on domestic maps such as Amap or Baidu Maps, and need to be converted to the coordinates used in the corresponding maps. Please refer to the corresponding map API documents for the conversion algorithm.

1.3.6 D1 (Location tracking command, location use D0 command upload)

Note: only OC32 lock supports this command. When the location tracking is turned on, the GPS positioning module will be in a normal state, the power consumption of the lock will increase, greatly reducing the lock life time, and it is not recommended to trun on the location tracking in normal state.

	<1>
Server -> lock	***CMDS,OM,123456789123456,200318123020,D1,60#<LF>**
1	Upload location interval unit: seconds, when this value is 0, turn off tracking (range :0-4294967295)
	<1>
Lock -> server	***CMDR,OM,123456789123456,200318123020,D1,60#<LF>**
1	Upload location interval unit: seconds, turn off trace when this value is 0

1.3.7 S5 (Access to Lock Information)

Server -> lock	***CMDS,OM,123456789123456,200318123020,S5#<LF>**
	<1><2><3><4><5>
Lock -> server	***CMDR,OM,123456789123456,200318123020,S5,412,30,5,0,0#<LF>**
1	Lock current voltage, unit 0.01 V,412->4.12 V (range :320-420) Reference for voltage and electricity percentage Appendix II
2	Current network signal value, the larger this value, the better the signal (range :2-32)
3	Current number of GPS satellites searched
4	Lock status 0-> unlock status 1-> lock status
5	Retention, fill in 0

1.3.8 S8 (Bike search commands)

	<1><2>
Server -> lock	***CMDS,OM,123456789123456,200318123020,S8,8,0#<LF>**
1	Ring times, when the number is 0, the default ring 8 times (range :0-255)
2	Reserve
Lock -> server	***CMDR,OM,123456789123456,200318123020,S8,8,0#<LF>**
1	Ring times
2	Reserve

1.3.9 G0 (Acquire to Lock Firmware Information)

Server -> lock	***CMDS,OM,123456789123456,200318123020,G0#<LF>**
	<1> <2>
Lock -> server	***CMDR,OM,123456789123456,200318123020,G0,XX_110,Jul 4 2018#<LF>**
1	XX -> device identification code, different versions of horseshoe lock identification code 110->V1.1.0 software version number

2	Date of Software Compilation

1.3.10 W0 (Alarm command)

	<1>
Lock -> server	***CMDR,OM,123456789123456,200318123020,W0,1#<LF>**
1	1： Illegal movement alarm 2: Fall alarm 6: Clear fall alarm (bike picked up)
Server -> lock	***CMDS,OM,123456789123456,200318123020,Re,W0#<LF>**

1.3.11 U0 (Detection upgrade/start upgrade)

Note: horseshoe locks send U0 detection updates 24 H each interval, and services can also actively send U0 start upgrades

	<1> <2> <3>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,U0,A 1,110,1101#<LF>**
1	Lock device software version 110->V1.1.0
2	Horseshoe Lock device identification code
3	Date of firmware compilation
	<1> <2> <3> <4> <5>
Server -> lock	***CMDS ,OM,123456789123456,200318123020,U0,220,128,32434,A1,C7qn#<LF>**
1	Upgrade data package (1-128 Byte per package, surplus plus 1 package)
2	Data length per package Range: 1-128Byte
3	Total upgrade data CRC16 check value, decimal form (for CRC16 calculations, see Appendix IV）
4	Upgrade file corresponding device identification code (support to upgrade horseshoe lock, corresponding horseshoe lock device identification code contact Omni acquisition) Equipment identification code for upgrading cable locks :30
5	Upgrade key (sample key is “Vgz7” by default, batch order will have random generation,please contact Omni sales to get the upgrade key)

1.3.1 2 U1 (Acquisition of upgrade data)

	<1><2>
Lock -> server	***CMDR ,OM,123456789123456,200318123020,U1,100,A1#<LF>**
1	Which packages upgrade files to get (from 0)
2	Device identification code corresponding to the acquired upgrade file
	<1> <2> <3> <4>
Server -> lock	***CMDS,OM,123456789123456,200318123020,U1,100,1234,DATA#<LF>**
1	Which packages of upgrade files (from 0)
2	CRC16 check values per packet of upgrade data (for CRC16 calculation methods see Appendix IV）
3	Upgrade data (corresponding RAW data intercepted in upgrade file, non-string form)

1.3.1 3 U2 (Notification of upgrade results)

	<1><2>
Lock -> server	*CMDR ,OM,123456789123456,200318123020,U2,A1,0#<LF>
1	Upgrade device identification code
2	Upgrade result 0-> success 1-> failure
Server -> lock	No response

1.3.1 4 K0 (Set/get BLE 8 byte communication KEY)

	<1> <2>
Server -> lock	*CMDS,OM,123456789123456,200318123020,K0,1,yOTmK50z#<LF>
1	Set or read ID 0-> read 1-> settings (when 0, item 2 is empty ,',' retained)
2	BLE 8 byte communication KEY
	<1>
Lock -> server	*CMDR ,OM,123456789123456,200318123020,K0,yOTmK50z #<LF>
1	BLE 8 byte communication KEY

1.1.35 I0 (Obtain SIM biked ICCID number)

Server -> lock	*CMDS,OM,123456789123456,200318123020,I0#<LF>
	<1>
Lock -> server	*CMDR,OM,123456789123456,200318123020,I0,123456789AB123456789#<LF>
1	SIM card ICCID(generally 20 digits)

1.3.16 M0 (Get lock Bluetooth MAC address)

Server -> lock	*CMDS,OM,123456789123456,200318123020,M0#<LF>
	<1>
Lock -> server	*CMDR,OM,123456789123456,200318123020,M0,12：34：56：78：90：AB#<LF>
1	MAC address

1.3.17 S0(Shutdown command)

Note: this command is used for shutdown transportation, need to unlocked the device before shutdown, the way to boot the devvice is through charging or lock manually.

Server -> lock	*CMDS,OM,123456789123456,200318123020,S0#<LF>
Lock -> server	No respond

1.3.1 8 S1 (Reboot command)

Server -> lock	***CMDS,OM,123456789123456,200318123020,S1#<LF>**
Lock -> server	**No**

1.4 Extended Functional command

1.4.1 L5 (External device control)

Note: This command applies only to horseshoe locks that support external cable lock.

<1>	
Server -> lock	***CMDS,OM,123456789123456,200318123020,L5,1#<LF>**
1	Operation 1->Battery lock unlock 2->Helmet lock unlock 3->Cable lock unlock 4->Hub lock unlock 17> Battery lock locked 18->Helmet lock locked 19->Cable lock locked 20->Hub lock locked 33-> Access battery Lock status 34->Access helmet Lock status 36->Access Cable Lock status 37->Access OC32 cable Lock status
<1><2>	
Lock -> server	***CMDR,OM,123456789123456,200318123020,L5,1,0#<LF>**
1	Operation 1->Battery lock unlock 2->Helmet lock unlock 3->Cable lock unlock 4->Hub lock unlock 17> Battery lock locked 18->Helmet lock locked 19->Cable lock locked 20->Hub lock locked 33-> Access battery Lock status 34->Access helmet Lock status 36->Access Cable Lock status 37->Access OC32 cable Lock status
2	Result 0-> success 1-> failure 2-> communication timeout 16-> lock state 17-> unlock state

1.4.2 G1 (Get cable lock firmware version)

Note: This command applies only to horseshoe locks that support external cable lock.

Server -> lock	***CMDS,OM,123456789123456,200318123020,G1#<LF>**
<1><2><3><4><5>	
Lock -> server	***CMDR,OM,123456789123456,200318123020,G1,0,0,0,110,0#<LF>**
1	Reserve
2	Reserve
3	Reserve
4	Cable Lock Software Version 110->V1.1.0
5	Reserve

1.4.3 B0 (Beacon validation)

Note: this command is only suitable for supporting Beacon function horseshoe lock, Beacon verification function is used to return the bike at the designated location, the return point needs to place Omni Beacon device, the horseshoe lock will continue to search the device for 5 seconds after closing the lock, and upload the search results. The server determines whether the fee is settled according to the upload results.

	<1><2> <3> <4> <5><6>
Lock -> server	***CMDR,OM,123456789123456,000000000000,B0,1,10,12:34:56:78:90:AB,1578386704,50,0#**
1	Search for Beacon Device 0: Beacon Device not searched 1: Searched for Beacon Device
2	Beacon Device Type Identification Code
3	Beacon device MAC address
4	Beacon device timestamp
5	Beacon device remaining battery power,percentage form
6	Beacon Device Longitude
7	Beacon device latitude
8	Reserve
Server -> lock	***CMDS ,OM ,123456789123456,000000000000,B0,0#**
1	Validation result 0-> validation 1->MAC invalid 2-> time error 3-> Search Beacon Device again 4-> Stop Search Device

1.4.4 C0 (RFID card unlock request)

Note: This command applies only to horseshoe locks that support RFID functions

	<1><2> <3>
Server -> lock	***CMDR ,OM,863725031194523,000000000000,C0,0,0,000000001A2B3C4D #**
1	Action requested 0-> unlock operation
2	Retention, fill in 0
3	biked number, hexadecimal form (range 00000000000000- FFFFFFFFFFFFFFF)
Lock -> server	**After receiving the command verification biked, the server sends the L 0 command to unlock**

1.4.5 C1 (Management RFID number setting)

Note: This command applies only to horseshoe locks that support RFID functions

	<1> <2>
Server -> lock	***CMDS,OM,863725031194523,000000000000,C1,0,000000001A2B3C4D#**
1	Request operation 0-> Set administrator biked number 1-> Query administrator biked number 2-> Remove administrator biked number
2	Administrator biked number set, hexadecimal form (value range 00000000000000- FFFFFFFFFFFFFFF) When requests are 1 and 2, fill in 0 here

	<1>
Lock -> server	*CMDR ,OM,863725031194523,000000000000,C1,000000001A2B3C4D#
1	Administrator biked number stored in the current lock in hexadecimal form (value range 00000000000000-FFFFFFFFFFFFFFFF) When the administrator biked number is removed, the biked number becomes the default value 00000000000000000
2	Reserve, fill in 0

1.4.6 D2 (WIFI Positioning command)

Note: This command applies only to horseshoe locks that support WIFI positioning functions

	<1><2>
Server ->IoT	*CMDS ,OM,863725031194523,000000000000,D2,0,3#<LF>
1	Parameter Description: 0-> access to WIFI location information 1-> set to not upload WIFI location information automatically(default) 2-> set to upload when GPS data in D0 command is invalid 3-> set to upload D0 commands
2	Scan WIFI AP time, unit second, valid range :2-15. Out of valid range, default value 3 seconds
	<1><2> <3> <4> <5> <6>...<13>
IoT -> server	*CMDR,OM,863725031194523,000000000000,D2,1,3,A1B1C1D1E1F1-30,A2B2C2D2E2F2-40,A3B3C3D3E3F3-50,0,0,0,0,0,0,0,0#<LF>
1	Current positioning mode: 1-> does not upload WIFI location information (default) 2-> upload GPS data in D0 command when invalid 3-> upload D0 command actively
2	Current scan WIFI AP time
3-13	Near IoT WIFI AP data, up to 10 data, less than 10 filled with 0. Data description, example: A1B1C1D1E1F1-30" "A1B1C1D1E1F1"->WIFI hotspot MAC address, hex string form ,"-30"->RSSI(signal strength)

2.BLE Communication Protocol Description

2.1 Packet format

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random number, generated by the data sender, used to encrypt data
2	KEY	The communication secret key is randomly generated by the bike lock， APP obtained by the (0x11） command
3	CMD	Command Words
4	LEN	Data length
5	DATA	Data
+LEN 5	CRC	CRC previous data encrypted CRC16 check value
+LEN 6		

2.2 Data encryption process

Encryption composition: random number, KEY.

Encryption process:

1、NUM of random numbers

2、Randomized variant NUM_1=NUM +0x32

3、Fill the NUM_1 into the first byte of data

4、CRC the previous plaintext data and backfill the results after the NUM is different or (^) NUM respectively

5、CRC16 check the data before CRC and fill in the check value to the CRC position.

Note: See detailed examplesAppendix III

2.3 APP Communication process with locks

1.Bluetooth connection APP and bike lock

2.APP send (0x11) command to bike lock to obtain communication secret key KEY

3.bike lock returns communication secret key KEY,APP need to save secret key for subsequent communication

4.APP communication with bike locks

Note: The secret key KEY is retrieved only when a Bluetooth connection between the APP and the bike lock is established, and communication remains unchanged thereafter

2.4 UUID used

OC30 horseshoe lock UUID ：

Service UUID ： 0783b03e-8535-b5a0-7140-a304d2495cb7

characteristic under the service

Characteristic UUID	Type of operation	Note
0783b03e-8535-b5a0-7140-a304d2495cba	Write	Write commands to hardware
0783b03e-8535-b5a0-7140-a304d2495cb8	Notify	Information returned by hardware

OC32 horseshoe lock UUID ：

Service UUID ： 6e400001-b5a3-f393-e0a9-e50e24dcca9e

characteristic under the service

Characteristic UUID	Type of operation	Note
6e400002-b5a3-f393-e0a9-e50e24dcca9e	Write	Write commands to hardware
6e400003-b5a3-f393-e0a9-e50e24dcca9e	Notify	Information returned by hardware

2.5 Command details and examples

2.5.1 Access to communication KEY commands (0x11)

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	0x00
3	CMD	0x11
4	LEN	0x08
5-12	DATA	Device identification KEY,8 bytes (sample default "yOTmK50z", batch random generation need to contact Omni acquisition)
13	CRC	CRC16 check value of encrypted data before CRC (0-12)
14		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x11
4	LEN	0x01
5	DATA	secret key (KEY), KEY for communication
6	CRC	CRC16 check value of encrypted data before CRC (0-5)
7		

2.5.2 Unlock command (0x21)

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x21
4	LEN	0x09
5-8	DATA	APP End User ID Number
9-12	DATA	Unlock timestamp 4 bytes, high in front
13	DATA	Unlock type (0: inside fence ,1: outside fence)
14	CRC	CRC previous data encrypted CRC16 check value
15		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x21
4	LEN	0x05
5	DATA	Return value 0: Unlock success 1: Unlock failure
6-9	DATA	Unlock timestamp 4 bytes, high in front
10	CRC	CRC previous data encrypted CRC16 check value
11		

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x21
4	LEN	0x00
6	CRC	CRC previous data encrypted CRC16 check value
7		

2.5.3 Lock command (0x22)

Note: if the lock sends a lock return and the timeout does not receive a APP reply, an old cycling record can be obtained by 0x51 command

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x22
4	LEN	0x09
5	DATA	0： lock success 1: lock failure
6-9	DATA	Unlock timestamp
10-13	DATA	Ride time 4 bytes (in minutes)
14	CRC	CRC previous data encrypted CRC16 check value
15		

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: xFE 0
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x22
4	LEN	0x00
5	CRC	CRC previous data encrypted CRC16 check value
6		

2.5.4 Query lock status (0x31)

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x31
4	LEN	0x00
5	CRC	CRC previous data encrypted CRC16 check value
6		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x31
4	LEN	0x07
5	DATA	Lock status 0: Unlock 1: Lock
6		Battery power example :37(decimal） 3.7V
7		0:No data uploaded 1: No
8-11		Time stamp 4 bytes
12	CRC	CRC previous data encrypted CRC16 check value
13		

2.5.5 Access to unuploaded data (0x51)

Note: this data is the data not uploaded to the APP or server after the user is locked, including the user ID and the time of the vehicle, which is used to settle the fee

APP read this data need to send 0x52 command clear

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x51
4	LEN	0x00
5	CRC	CRC previous data encrypted CRC16 check value
6		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x51
4	LEN	0x012
5-8	DATA	timestamp used by the user when unlocking
9-12	DATA	User riding time unit: minutes
13-16	DATA	ID of users
17	CRC	CRC previous data encrypted CRC16 check value
18		

2.5.6 Clear unuploaded data in bike locks (0x52)

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x52
4	LEN	0x00
5	CRC	CRC previous data encrypted CRC16 check value
6		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x52
4	LEN	0x01
5	DATA	Return value :0: Success 1: Failure
6	CRC	CRC previous data encrypted CRC16 check value
7		

2.5.6 Cable lock control (0x81)

APP-> bike lock

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x81
4	LEN	0x01
5	DATA	Operation 0x03-> Cable lock unlocking 0x23-> Acquisition of cable lock status
6	CRC	CRC previous data encrypted CRC16 check value
7		

bike lock ->APP

bytes	Item	Note
0	STX	Header/frame head fixed value: 0xFE
1	NUM	Random numbers
2	KEY	Communication secret key
3	CMD	0x81
4	LEN	0x02
5	DATA	Operation 0x03-> Cable lock unlocking 0x23-> Acquisition of cable lock status
6		Operational results 0x00-> Success 0x01-> Failure 0x02-> Communication timeout for device 0x10-> Lock status 0x11-> Unlock status
7	CRC	CRC previous data encrypted CRC16 check value
8		

Appendix I: Note 0xFFFF before command header

String format：*CMDS ,OM,863158022988725,000000000000,L 0,20,0,1554862465#

HEX format：FF FF 2A 43 4D 44 53 4F 4D 38 36 33 31 35 38 30 32 32 39 38 38 37 32 35 30 30 30 30 30 30 30 30 30 30 30 30 4C 30 32 30 30 31 35 35 34 38 36 32 34 36 35 23 0A

JAVA command package:

```
private byte[] getSendOrder(){
        // command
        byte[] order ="*CMDS,....#\n".getBytes();
        // add 0xFF,0xFF
        return addByte(new byte[]{(byte) 0xFF,(byte) 0xFF},order);
    }

    private byte[] addByte(byte[] b1,byte[] b2){
        byte[] b = new byte[b1.length+b2.length];
        System.arraycopy(b1, 0, b, 0, b1.length);
        System.arraycopy(b2, 0, b, b1.length, b2.length);
        return b;
    }
```

Appendix II: Table for Lithium Battery Voltage and Electricity Percentage

Voltage (unit：0.01V)	Percentage of electricity
4.2	100%
4.116	95%
3.9561	90%
3.8687	80%
3.7946	70%
3.7344	60%
3.6982	50%
3.6548	40%
3.632	30%
3.5966	20%
3.5473	10%
3.4	0%

Appendix III: Bluetooth Encryption, Decryption Process

1. encryption: take app to the lock to obtain the operation KEY ,0x11 commands as an example.

item	index	hex (original)	hex (+0 x32)	hex (xor34)	calc CRC
data2	0	**FE**	FE	FE	FE
num	1	**34**	66	66	66
key	2	**0**	0	34	34
cmd	3	**11**	11	25	25
len	4	**8**	8	3C	3C
data	5	**79**	79	4D	4D
data	6	**4F**	4F	7B	7B
data	7	**54**	54	60	60
data	8	**6D**	6D	59	59
data	9	**4B**	4B	7F	7F
data	10	**35**	35	1	1
data	11	**30**	30	4	4
data	12	**7A**	7A	4E	4E
crc	13				B4
crc	14				F6

2. decryption: take the KEY of the lock return operation to the app as an example

item	index	hex	step1	step2	step3	step4	step5	step6
stx	0	**FE**	FE	**FE**		FE	FE	FE
num	1	**3C**				3C	3C	A (3C-32)
key	2	**F5**				F5	F5	FF (F5^A)
cmd	3	**1B**				1B	1B	11(1 B^A)
len	4	**0B**		**0B**	1(B^(3 C-32))	0B	0B	1(B^A)
data (key)	5	**F5**				F5	F5	FF (F5^A)
crc	6	**7C**				7C	7C	
crc	7	**C1**				C1	C1	
?	8	**0(IF HAVE)**						
?	9	**A (IFH A VE)**						
?	10	**B (IFHAVE)**						
?	11	**C (IFHAVE)**						
?	12	**D (IFHAVE)**						
?	13	**E (IFHAV E)**						

Step1： find FE address

Step2： find len location

Step3： decrypted The value of the len, the encrypted value is different or the processed random number is

1=0x0B^(0 x3C-0x32)

Step4： get the command according to the starting position and length of the command

Step5： detection CRC value

Step6： parse the command to get the desired value.

Appendix IV: CRC 16 calculation code (C code as an example)

```
unsigned char CRCHi[] = {
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
};

  unsigned char CRCLo[] = {
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
};

 unsigned int CRC16( unsigned char * pucFrame, unsigned int usLen)
{
    unsigned char    ucCRCHi = 0xFF;
    unsigned char    ucCRCLo = 0xFF;
    unsigned int iIndex=0x0000;

    while(usLen--)
    {
        iIndex = ucCRCLo ^ *(pucFrame++);
        ucCRCLo = ucCRCHi ^ CRCHi[iIndex];
        ucCRCHi = CRCLo[iIndex];
    }
    return (unsigned int)(ucCRCHi << 8 | ucCRCLo);
}
```

Appendix V: Bluetooth Broadcast Data Description

Manufacturer data Data Description

Example: 0xFFFFD713315DDBF685002900

Data	Note
FFFF	ID
D713315DDBF6	D7： 13： 31： 5D ： DB ： F6 MAC address
85	Low byte device type
00	High byte device type
29	Lock operating voltage 0x29->41->4.1 V
00	Lock status 0-> lock status 1-> unlock status