| Type | |
|---|---|
| Date | 2024/10/10 |
| Ver. | Ver : 2.1.8 |
| Dept. | R&D Dept |

# Full-featured parking lock communication protocol

Writer：
Date：
Audit：

## Revision history record

| Version | Content modification | Date |
|---|---|---|
| V2.1.0 | The new parking lock adopts microwave detection method, and adds the status of whether there is a car above, lock lever position detection, and HTTP network upgrade to the old version of the protocol. The new protocol will be used uniformly in the future. The old protocol functionality is no longer maintained. (Blocking and not blocking cars are updated to switch locks) | 2023.11.23 |
| V2.1.3 | Change the description for alarm type | 2024.01.23 |
| V2.1.4 | Add BLE command Query Cover state information (0x32) refer 1.7.8.2<br>Add the method to obtain the whole information(refer 1.9.4. How to obtain Parking lock full information) | 2024.01.26 |
| V2.1.5 | Add TCP command S6 for Setting the automatic lock function and H0 | 2024.03.08 |
| V2.1.6 | Fix the description for Q0 command and H0 command.<br>Add Autolock statu in H0 and S5 | 2024.03.14 |
| V2.1.7 | Add lock calibration instructions | 2024.06.24 |
| V2.1.8 | Adjust the parking lock structure. Remove related commands<br>Remove Bluetooth command: 0X04 0X06 0X42 0X45 Bluetooth remote control related commands | 2024.10.10 |

Contents

# 1.Bluetooth communication protocol

## 1.1 Overview

This protocol is used to describe the communication protocol between the parking lock and the server, the parking lock and the APP, and the APP and the parking lock remote control.

## 1.2 Communication package format

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | 0xA3A4 Data head / frame head fixed value: 0xA3A4 |
| 2 | LEN | Data length (the length of the DATA) |
| 3 | RAND | Random number, generated by the data sender, used to encrypt data |
| 4 | KEY_once | The communication secret key is randomly generated by the parking lock and obtained by the APP through the (0x01) command. |
| 5 | CMD | Command word |
| 6 | DATA | Data |
| 6+LEN | CRC | CRC CRC8 check value after encrypting the previous data |

## 1.3 Data encryption process

Encryption composition: random number, KEY.

Encryption process:

1. RAND Generate random numbers RAND
2. Generate random number variant RAND_1 = RAND + 0x32
3. Fill RAND_1 to the 3rd byte of data
4. Use RAND to XOR (^) the plaintext data before CRC after RAND and backfill the result accordingly.
5. Perform CRC8 check on the data before CRC, and fill in the check value into the CRC position.

**For Bluetooth encryption and decryption, please refer to Appendix 1.**

## 1.4 Communication process between APP and Parking Lock

1. Establish Bluetooth connection between APP and Parking Lock
2. APP sends (0x01) command to the parking lock to obtain the communication key KEY
3. The parkinglock returns the communication secret key KEY, and the APP needs to save the secret key for subsequent communication.
4. Communicates with the parking lock

**Note: The secret key KEY is only reacquired when the APP establishes a Bluetooth connection with the parking lock, and the communication remains unchanged thereafter.**

## 1.5 UUID used by the parking lock

Service UUID :6e400001-b5a3-f393-e0a9-e50e24dcca9e
The characteristic under this service

| characteristic UUID | operation type | illustration |
|---|---|---|
| 6e400002-b5a3-f393-e0a9-e50e24dcca9e | Write | Write instructions to hardware |
| 6e400003-b5a3-f393-e0a9-e50e24dcca9e | Notify | Information returned by the hardware |

When Android registers notifications, the UUID of the descriptor under characteristic needs to be used: 00002902-0000-1000-8000-00805f9b34fb
Not required for iOS.

## 1.6 UUID used by the remote control

Service UUID :6F400001-b5a3-f393-e0a9-e50e24dcca9e
The characteristic under this service

| characteristic UUID | operation type | illustration |
|---|---|---|
| 6F400002-b5a3-f393-e0a9-e50e24dcca9e | Write | Write instruction to hardware |
| 6F400003-b5a3-f393-e0a9-e50e24dcca9e | Notify | Information returned by the hardware |

When Android registers notifications, the UUID of the descriptor under characteristic needs to be used:00002902-0000-1000-8000-00805f9b34fb
Not required for iOS.

## 1.7 Parking lock command details and examples

1.7.1 Verify device KEY to obtain communication KEY command (0x01)

#### 1.7.1.1 APP->Parking lock

When the App connects to the Bluetooth device, it must first verify the device KEY through the 0x01 command to obtain the KEY for communication with the Bluetooth device. The equipment KEY of each parking lock is different, and users can also define the equipment KEY by themselves. When the connected Bluetooth device does not send the 0x01 command within 5 seconds, or the verification pairing password is incorrect, the Bluetooth device will automatically disconnect from the app.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x08 |
| 3 | RAND | random number |
| 4 | KEY | 0x00 |
| 5 | CMD | 0x01 |
| 6-13 | DATA | Device KEY, 8 bytes (OmniW4GX) |
| 14 | CRC | CRC CRC8 check value after encrypting the previous data |

If the user is connecting to Bluetooth, the device KEY is [0x4F6D6E6957344758].
Therefore, DATA[0]=0x4F, DATA[1]=0X6D, DATA[2]=0x6E, DATA[3]=0x69
DATA[4]=0x57,DATA[5]=0X34,DATA[6]=0x47,DATA[7]=0x58

#### 1.7.1.2 Parking Lock->APP

After receiving the command to obtain the communication KEY, the parking lock returns the KEY used for communication in DATA. The communication KEY obtained is always valid during this connection.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x02 |
| 3 | RAND | Random number 0x00 |
| 4 | KEY | KEY Communication key 0x00 fills in the current KEY |
| 5 | CMD | 0x01 |
| 6 | DATA | Verification identification: (1: success, 0: failure) |
| 7 | DATA | KEY Communication key (KEY), used for current communication |
| 8 | CRC | CRC CRC8 check value after encrypting the previous data |

1.7.2 Command error prompt command (0x10)

#### 1.7.2.1 Parking lock->APP

After connecting to the device, the user directly sends 0x12, 0x13 and other commands to communicate with the device without obtaining the communication KEY through the 0x01 command. The device prompts the user through this command. The user has obtained the communication KEY, but when sending other commands, the communication KEY used is incorrect, and this command is also used to prompt.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | random number |
| 4 | KEY | Communication key 0x00 or current KEY |
| 5 | CMD | 0x10 |
| 6 | DATA | Error message:<br>1：CRC authentication error<br>2：Communication KEY not obtained<br>3：The communication KEY has been obtained, but it is wrong |
| 7 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.7.3 Unlock command (0x05)

#### 1.7.3.1 APP->Parking lock

The APP sends unlocking control to the parking lock. The operation type is 1.

Use the 1 control command to unlock. After receiving the unlocking command, the parking lock will perform the unlocking operation.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x0A |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x05 |
| 6 | DATA | Instruction type (1: control instruction) |
| 7-10 | DATA | User ID (four bytes random user ID) |
| 11-14 | DATA | Operation timestamp |
| 15 | DATA | Reservation type |
| 16 | CRC | CRC CRC8 check value after encrypting the previous data |

#### 1.7.3.2 Parking lock->APP

After the parking space lock performs the operation, it returns to the operating state. When the parking space lock receives the control command sent by the APP, it returns control result 0; when the parking space lock completes execution, it returns operation success 1; when the

parking space lock executes the operation, it returns 2.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x05 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x05 |
| 6 | DATA | Control result (0: Execute unlocking, 1: Unlocking successful, 2: Unlocking timeout) |
| 7-10 | DATA | Operation timestamp |
| 11 | CRC | CRC CRC8 check value after encrypting the previous data |

#### 1.7.3.3 APP -> Parking lock

After receiving the operation status command from the parking lock (1: operation successful, 2: operation timeout), the APP replies to the parking lock. If it does not reply, the parking lock uploads the operating status (L0 command) through the network.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x05 |
| 6 | DATA | Command type (2: reply command) |
| 7 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.7.4 Lock command (0x15)

#### 1.7.4.1 APP->Parking Lock

The APP sends lock control to the parking space lock. Operation type is 1.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x15 |
| 6 | DATA | Instruction type (1: control instruction) |
| 7 | CRC | CRC CRC8 check value after encrypting the previous data |

#### 1.7.4.2 Parking lock->APP

After the parking lock performs the operation, the lock result is returned.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |

| | | |
|---|---|---|
| 2 | LEN | 0x09 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x15 |
| 6 | DATA | Control result (0: locking starts, 1: locking successful, 2: locking timeout) |
| 7-10 | DATA | Last unlock timestamp |
| 11-14 | DATA | reserved |
| 15 | CRC | CRC CRC8 check value after encrypting the previous data |

#### 1.7.4.3 APP -> Parking lock

After receiving the lock status command from the parking lock (1: operation successful, 2 operation timeout), the APP replies to the parking lock. If it does not reply, the parking lock uploads the operation status (L1 command) through the network.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x15 |
| 6 | DATA | Command type (2 reply command) |
| 7 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.7.5 Obtain the parking lock MAC address (0x03)

#### 1.7.5.1 APP -> Parking lock

Obtain the MAC address of the parking lock itself.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |
| 5 | CMD | 0x03 |
| 6 | DATA | Operation type (1: control instruction) |
| 7 | CRC | CRC CRC8 check value after encrypting the previous data |

#### 1.7.5.2 Parking lock->APP

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x07 |
| 3 | RAND | random number |
| 4 | KEY | Communication key |

| | | |
|---|---|---|
| 5 | CMD | 0x06 |
| 6 | DATA | 1: Obtain successful, 0: Obtain failed |
| 7-12 | DATA | MAC address |
| 13 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.7.6 Query of parking lock information (0x31)

#### 1.7.6.1 APP -> Parking lock

Can obtain information such as the device switch status and battery version.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x07 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x31 |
| 6 | DATA | Command type (1: control instruction) |
| 7 | CRC | CRC8 check value after data encryption before CRC (0-5) |

#### 1.7.6.2 Query Cover state information (0x32)

This command can get the cover state

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x32 |
| 6 | DATA | Command type (1: control instruction) |
| 7 | CRC | CRC8 verified value after encrypted previous CRC data (0-5) |

### 1.7.7 Modify Equipment KEY (0x33)

**APP -> Parking lock**

Modify the 8-byte device KEY verified when connecting the device.

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x33 |
| 6-13 | DATA | New verification KEY |

| | | |
|---|---|---|
| 14 | CRC | CRC 8 verified value of encrypted CRC data |

### 1.7.8 Set an alarm(0x40)

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x40 |
| 6 | DATA | 0x01 Normal Alarm sound 0x00 Mute 0X02 No more alarm after 3 abnormal sounds are detected |
| 7 | CRC | CRC 8 verified value of encrypted CRC data |

### 1.7.9 Set automatic locking (0x41)-- New

Description of the automatic lock function: The device automatically locks in the following two cases

1: After unlocking, if the car does not enter within 3 minutes, the device will automatically lock when it detects that there is no car above.

2: After unlocking, the car enters the parking space. The next time the car leaves, the device detects that there is no car above it and automatically locks.

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x41 |
| 6 | DATA | 0x01 Enable the automatic locking function<br>0x00 Disable the automatic locking function |
| 7 | CRC | CRC8 check value of encrypted data before CRC |

### 1.7.10 Find Parking lock (0x43)-- New

Note: Executing this command will trigger the device alarm once

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |

| | | |
|---|---|---|
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x43 |
| 6 | DATA | 0x01 |
| 7 | CRC | CRC8 check value of encrypted data before CRC |

### 1.7.11 Restart the Parking lock (0x44)--New

Note: After this command is executed, the device will restart directly without reply

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x44 |
| 6 | DATA | 0x01 |
| 7 | CRC | CRC8 check value of encrypted data before CRC |

### 1.7.12 Set the address of 485 (0x70)--New

Note: This command is used to set the address of 485 slave, the default factory setting is 1

**APP -> Parking lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data head / frame head Fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret key |
| 5 | CMD | 0x70 |
| 6 | DATA | Device 485 postal address (range：1-254) default setting is 1 |
| 7 | DATA | New 485 postal address range：1-254 |
| 8 | CRC | CRC8 check value of encrypted data before CRC |

## 1.8 Bluetooth modification\Instructions for obtaining Parking Lock information

Information format:

"Information item: data, information item: data, information item: data, information item:,"

Example:

"APN:CMNET, PIN:, IPMODE:0, IP:120.24.228.19, PORT:9666, VCODE:OM, KEY:12345678, "

Information item description

| Item | Explanation |
|---|---|
| APN | SIM card APN |
| PIN | SIM card PIN |
| IPMODE | Server address mode: 0: IP 1: Domain name |
| IP | Server address or domain name |
| PORT | Server port |
| VCODE | Manufacturer's name (OM) |
| KEY | Bluetooth communication verification key |
| UKEY | Upgrade verification key |
| VERSION | Software version |
| VDATA | Compilation date |
| POWER | Battery power |
| | |

Note 1: When a certain piece of information is not set or not sent, the information item can be omitted or the information data can be left blank.

## 1.8.1 APP requests to obtain Parking Lock information command (0xFA)

**APP->Parking Lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x01 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret |
| 5 | CMD | 0xFA |
| 6 | CRC | CRC CRC8 check value after encrypting the previous data |

**Parking Lock->APP**

After receiving the 0xFA command, the Parking Lock sends the 0xFB command to start data transmission.

## 1.8.2. Start data transfer command (0xFB)

**APP->Parking Lock**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x0a |

| | | |
|---|---|---|
| 3 | RAND | Random number |
| 4 | KEY | Communication secret |
| 5 | CMD | 0xFB |
| 6 | TYPE | Transmission data type (0: upgrade file 1: system information) |
| 7-8 | N_PACK | Total number of data packets (2byte) High byte first |
| 9-10 | DATA_CRC | Total transmitted data CRC check value (2byte) High byte first |
| 11 | DIV_TYPE | Default 0x86 (1byte) |
| 12-15 | VIF_KEY | The total size of the file (4byte) , high byte first, as in the previous example, 0x000ffff, means that the file size is 65535 bytes. Note: the size does not contain the last packet of zero-padding data |
| 16 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.8.3. Get data command (0xFC)

**Parking Lock->APP**

| SN | ITEM | Explanation |
|---|---|---|
| 0-1 | STX | Data header/frame header fixed value: 0xA3A4 |
| 2 | LEN | 0x03 |
| 3 | RAND | Random number |
| 4 | KEY | Communication secret |
| 5 | CMD | 0xFC |
| 6-7 | N_PACK | Which packet of data to obtain (2 bytes), high byte first |
| 8 | D_TYPE | Equipment type |
| 9 | CRC | CRC CRC8 check value after encrypting the previous data |

### 1.8.4. How to obtain Parking lock hardware information

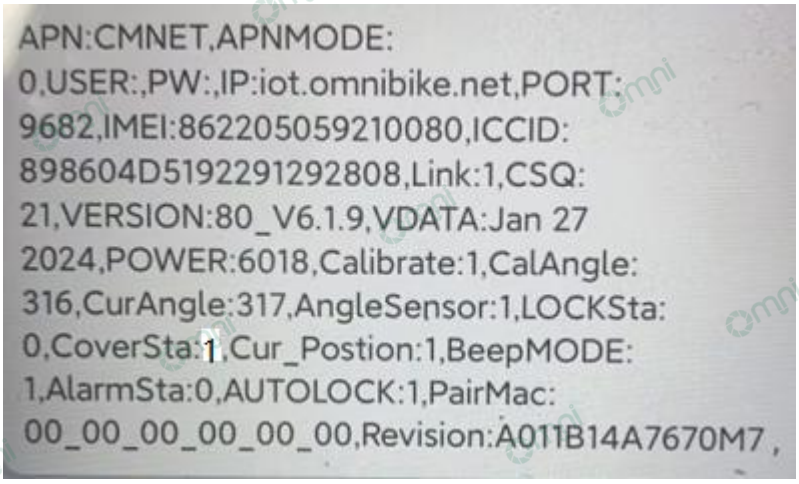
Step 1: APP send FA command ( refer to FA command)
APP send : a3 a4 0 67 d fa

Step2:Lock response by FB (To tell app how many bytes need to obtain)
Lock reponse: a3 a4 a df d fb 1 0 16 c0 98 80 0 0 0 0

Step3:APP Send FC( to obtain full information)
APP send : a3 a4 3 79 d fc 0 0 80

.....

.....




| KEY word explation | |
|---|---|
| PORT | Server port |
| IMEI | Model IMEI number |
| Link | 1 :Connected 0: Disconnect |
| Power | Battery power |
| Calibrate | 1: Have calibrated 0: No calibrate |
| CoverSta | 1: Covered 0: No cover |

## 2. TCP communication protocol description

Note: The instruction is in string form, each item is separated by ', ', and each instruction ends with a newline (' \n ')

0xFFFF, in HEX format, must be added before the command header when the server sends commands

| <1> <2> <3> <4> <5> <6> <7><8> | | |
|---|---|---|
| **0xFFFF*BGCS,OM,123456789123456,XX,DDD#<line feed>** | | |
| **SN** | **ITEM** | **Explanation** |
| 1 | 0xFFFF | Two-byte starting bit, HEX, non-0xFFFF string |
| 2 | *BGCS | Header server -> Locks use *BGCS, lock -> servers use *BGCR |
| 3 | OM | Vendor code |
| 4 | 123456789123456 | The lock unique ID number, using the IMEI number of the lock communication module (15 digits) |
| 5 | XX | Command type |
| 6 | DDD | Contents carried by the command, there may be multiple items, separated by ', ' |

| | | |
|---|---|---|
| 7 | # | Command end |
| 8 | <line feed> | End with newline '\n' |

## 2.1 Sign-in command (Q0)

| | <1> <2> <3><4><5> |
|---|---|
| Lock->server | *BGCR,OM,866266066379439,Q0,6023,E7:0F:B7:FD:A6:74,1,2,2#<LF> |
| 1 | Device current working voltage |
| 2 | Device MAC address |
| 3 | reserved |
| 4 | reserved |
| 5 | The current position of the lock lever<br>1: the lock lever falls horizontally (unlocking position)<br>2: the lock lever stands up (locking position)<br>3: other position (not unlocking or locking position) |
| | |
| Sever-> Lock | No response |

## 2.2 Heartbeat command (H0)

| | <1> <2> <3><4><5><6><7><8> |
|---|---|
| Lock->server | *BGCR,OM,866266066379439,H0,1,6023,31,0,1,1,1,1#<LF> |
| 1 | Device status(0-unlock，1-lock) |
| 2 | Current working voltage |
| 3 | Current network value ranges from 2 to 32. The larger the value, the better the signal |
| 4 | Whether there is a car over it（1: there is a car 0:no car）Locked status this flag is invalid |
| 5 | Lock lever Current position :<br>1: lock lever horizontal down (unlocking position)<br>2: lock lever upright (locking position)<br>3: other position (not unlocking or locking position) |
| 6 | reserved |
| 7 | reserved |
| 8 | Automatic lock status 1: enabled 0: disabled |
| | |
| Sever-> Lock | No response |

## 2.3 Obtain device information and status command (S5)

| Sever-> Lock | *BGCS,OM,123456789123456,S5#<LF> |
|---|---|
| <1> <2> <3><4><5><6> <7> <8> <9> <10> | |
| **Lock->Server** | *BGCR,OM,866266066379439,S5,6028,48,30,1,0,2,898604D5192291255340,CMIOT,E7:0F:B7:FD:A6:74,1#<LF> |
| 1 | Current working voltage |
| 2 | Percentage of battery capacity |
| 3 | Current network value ranges from 2 to 32. The larger the value, the better the signal |
| 4 | Lock status (1:lock， 0:unlock) |
| 5 | Whether there is a car over it（1: there is a car 0:no car）Locked status this flag is invalid |
| 6 | Lock lever Current position :<br>1: lock lever horizontal down (unlocking position)<br>2: lock lever upright (locking position)<br>3: other position (not unlocking or locking position) |
| 7 | SIM card ICCID |
| 8 | SIM card APN |
| 9 | Bluetooth MAC address |
| 10 | Automatic lock status 1: enabled 0: disabled |

## 2.4 Operation request command (R0)

| | <1> <2> <3> <4> |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,R0,0,300,20,1700638984#<LF> |
| 1 | Requested operation 0-> Unlock operation 1-> Lock operation |
| 2 | KEY Valid time unit: second (0-65535) |
| 3 | User ID (The value ranges from 0 to 4294967295, generated by the server) |
| 4 | Current timestamp (accurate to second) |
| | <1><2> <3> <4> |
| Lock->Server | *BGCR,OM,123456789123456,R0,0,55,1234,1700638984#<LF> |
| 1 | Requested operation 0-> Unlock operation 1-> Lock operation |
| 2 | Operation KEY. The value ranges from 0 to 255 and is generated randomly by the lock. This KEY is required when the server sends the unlock/lock operation |
| 3 | User ID (same as the server sends） |
| 4 | Operation timestamp (Unix timestamp, 10 digits)[Operation serial number] |

## 2.5 Unlocking operation command (L0)

| | <1> <2> <3> |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,L0,24,20,1497689816#<LF> |
| 1 | Operation KEY, obtained by the R0 command |
| 2 | User ID (The value ranges from 0 to 4294967295, generated by the server) |

| | |
|---|---|
| 3 | Current timestamp (accurate to second) |
| | <1> <2> <3> |
| Lock->Server | *BGCR,OM,123456789123456,L0,0,1234,1497689816#<LF> |
| 1 | The status returns<br>0-> Unlocking succeeded 1-> Unlocking failed 2-> The KEY is incorrect or invalid |
| 2 | User ID (The value ranges from 0 to 4294967295, generated by the server) |
| 3 | Current timestamp (accurate to second) |
| | |
| Server->Lock | *BGCS,OM,123456789123456,Re,L0#<LF> (Server response） |

## 2.6 Lock command(L1)

| | |
|---|---|
| | <1> |
| Server->Lock | *BGCS,OM,123456789123456,L1,55#<LF> |
| 1 | Operation KEY, obtained by the R0 command |
| | <1> <2> <3> <4> |
| Lock->Server | *BGCR,OM,123456789123456,L1,0,1234,1497689816,5#<LF> |
| 1 | The status returns<br>0-> Unlocking succeeded 1-> Unlocking failed 2-> The KEY is incorrect or invalid |
| 2 | User ID (The value ranges from 0 to 4294967295, generated by the server) |
| 3 | Current timestamp (accurate to second) |
| 4 | Duration, 4 bytes of data, in the form of a string (unit: minutes) |
| | |
| Server->Lock | *BGCS,OM,123456789123456,Re,L1#<LF> (Server response） |

## 2.7 Firmware Version command (G0)

| | |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,G0#<LF> |
| | <1> <2> <3> |
| Lock->Server | *BGCR,OM,123456789123456,G0,V1.0.0,May172017,SIM7670V01#<LF> |
| 1 | Device software version |
| 2 | Device software compilation date |
| 3 | Network module version information |

## 2.8 Alarm command (W0)

| | |
|---|---|
| | **<1>** |
| Lock->Server | *BGCR,OM,123456789123456,W0,1#<LF> |
| 1 | Alarm content<br>1: Manually pull the alarm 2: reset to the lock/unlock state failed<br>3: Lock with shelter (Foreign matter detected) |
| | |
| Server->Lock | *BGCS,OM,123456789123456,Re,W0#<LF>( Server response） |

## 2.9 Modify Server command (S3)

| | <1> <2> |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,S3,omniserver.ombike.com,9681#<LF> |
| 1 | IP information |
| 2 | PORT information |

## 2.10 Remote Upgrade command (U0)

The server actively sends the upgrade, and the device responds to the upgrade status; the device does not need to return data to the server and can request the firmware directly. The new full-function parking lock network adopts the HTTP upgrade mode. The TCP upgrades are no longer supported

| | <1> <2> |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,U0,http://file/GEN4_V3_86_V1.0.0.2.bin,35212#<LF> |
| 1 | Http upgrade address |
| 2 | Checksum (sum of upgrade file in HEX format, decimal format) |
| <1> | |
| Lock->Server | *BGCR,OM,123456789123456,U0,220#<LF> |
| 1 | Result 100 The upgrade request is received. 101 The upgrade is rejected. 200 Do not perform any operation when the upgrade begins. The upgrade process ends. The locator automatically restarts |

## 2.11 Restart the Parking Lock command (S1)

| | |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,S1#<LF> |

## 2.12 Lock-finding command (S8)

| | |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,S8#<LF> |

## 2.13 Automatic lock command (S6)

| <1> |
|---|

| | |
|---|---|
| Server->Lock | *BGCS,OM,123456789123456,S6,1#<LF> |
| 1 | 1: Enable the automatic lock function. 0: disables the automatic lock function |
| <1> | |
| Lock->Server | *BGCR,OM,123456789123456,S6,1#<LF> |
| 1 | Automatic lock function 1 or 0 |

# Appendix I: Bluetooth encryption, decryption process

1. Encryption: Take app to lock acquisition operation KEY, 0x01 instruction as an example.

4F6D6E6957344758

| item | index | hex(original) | hex(+0x32) | hex(xor 34) | calc CRC |
|---|---|---|---|---|---|
| STX | 0 | **A3** | A3 | A3 | A3 |
| STX | 1 | **A4** | A4 | A4 | A4 |
| len | 2 | **08** | 08 | 08 | 08 |
| rand | 3 | **1E** | 50 | 50 | 50 |
| key | 4 | **00** | 0 | 1E | 1E |
| cmd | 5 | **01** | 01 | 1F | 1F |
| data | 6 | **4F** | 4F | 51 | 51 |
| data | 7 | **6D** | 6D | 73 | 73 |
| data | 8 | **6E** | 6E | 70 | 70 |
| data | 9 | **69** | 69 | 77 | 77 |
| data | 10 | **57** | 57 | 49 | 49 |
| data | 11 | **34** | 34 | 2A | 2A |
| data | 12 | **47** | 47 | 59 | 59 |
| data | 13 | **58** | 58 | 46 | 46 |
| crc | 14 | | | | 01 |

2. Decrypt: Take KEY that returns the operation to the app as an example ()

| item | index | hex | step1 | step2 | Step3 | Step4 | Step5 |
|---|---|---|---|---|---|---|---|
| stx | 0 | **A3** | **A3** | | A3 | A3 | A3 |
| stx | 1 | **A4** | **A4** | | A4 | A4 | A4 |
| len | 2 | **2** | | **2** | 2 | 2 | 2 |
| rand | 3 | **B0** | | | B0 | B0 | 7E(B0-32) |
| key | 4 | **27** | | | 27 | 27 | 59(27^7E) |
| cmd | 5 | **7F** | | | 7F | 7F | 01(7F^7E) |
| data (f) | 7 | **7F** | | | 7F | 7F | 01(7F^7E) |
| data (key) | 8 | **27** | | | 27 | 27 | 59(27^7E) |
| crc | 9 | **1A** | | | 1A | 1A | |
| ? | 8 | **0(IF HAVE)** | | | | | |
| ? | 9 | **A(IF HAVE)** | | | | | |
| ? | 10 | **B(IF HAVE)** | | | | | |
| ? | 11 | **C(IF HAVE)** | | | | | |
| ? | 12 | **D(IF HAVE)** | | | | | |
| ? | 13 | **E(IF HAVE)** | | | | | |

Step1:Find the A3 A4 subscript address

Step2: Find out where len is.

Step3:Get the command according to the starting position and length of the command

Step4:Detecting CRC values

Step6:Parse the command to get the desired values.

## Appendix II: CRC 8 Calculation Code (C code as an example)

```
const char CRC8Table[]={
   0, 94, 188, 226, 97, 63, 221, 131, 194, 156, 126, 32, 163, 253, 31, 65,
   157, 195, 33, 127, 252, 162, 64, 30, 95, 1, 227, 189, 62, 96, 130, 220,
   35, 125, 159, 193, 66, 28, 254, 160, 225, 191, 93, 3, 128, 222, 60, 98,
   190, 224, 2, 92, 223, 129, 99, 61, 124, 34, 192, 158, 29, 67, 161, 255,
   70, 24, 250, 164, 39, 121, 155, 197, 132, 218, 56, 102, 229, 187, 89, 7,
   219, 133, 103, 57, 186, 228, 6, 88, 25, 71, 165, 251, 120, 38, 196, 154,
   101, 59, 217, 135, 4, 90, 184, 230, 167, 249, 27, 69, 198, 152, 122, 36,
   248, 166, 68, 26, 153, 199, 37, 123, 58, 100, 134, 216, 91, 5, 231, 185,
   140, 210, 48, 110, 237, 179, 81, 15, 78, 16, 242, 172, 47, 113, 147, 205,
   17, 79, 173, 243, 112, 46, 204, 146, 211, 141, 111, 49, 178, 236, 14, 80,
   175, 241, 19, 77, 206, 144, 114, 44, 109, 51, 209, 143, 12, 82, 176, 238,
   50, 108, 142, 208, 83, 13, 239, 177, 240, 174, 76, 18, 145, 207, 45, 115,
   202, 148, 118, 40, 171, 245, 23, 73, 8, 86, 180, 234, 105, 55, 213, 139,
   87, 9, 235, 181, 54, 104, 138, 212, 149, 203, 41, 119, 244, 170, 72, 22,
   233, 183, 85, 11, 136, 214, 52, 106, 43, 117, 151, 201, 74, 20, 246, 168,
   116, 42, 200, 150, 21, 75, 169, 247, 182, 232, 10, 84, 215, 137, 107, 53
};

 unsigned char CRC8_Table(unsigned char *pucFrame, char usLen)
{
      unsigned char crc8 = 0;

            while(usLen--)
                  crc8 = CRC8Table[crc8^*(pucFrame++)];

      return(crc8);
}
```

# Appendix III: Panel status light / horn indication

**Note: Factory manual calibration is required before delivery!.**

| Lock state | Red light instructions | Green light instructions | Horn |
|---|---|---|---|
| Lock not calibration / waiting for calibration (default calibration before delivery) | Flash twice for every 10S | NULL | NULL |
| The lock calibration was successful | NULL | Flash 1 time | Ring 1 sound |
| Lock calibration failed | Flash 1 time | NULL | Ring 2 sound |
| The lock was successfully unlocked | NULL | Flash 2 times | NULL |
| The lock was successfully locked | NULL | Flash 1 time | NULL |
| Uncalibrated Status Switch Lock | Flash 1 time | NULL | Ring 1 sound |
| The Lock lock failure (blocked above) | NULL | NULL | Ring 2 sound |
| Pull alarm | Flash 1 time | NULL | Ring 1 sound |
| Received remote control | | Flash 3 times | NULL |
| Send a lock search instruction | Flash 2 times | NULL | Long ring 1 sound |