FitRider Scooter

Network Module and Server Protocol

No.	Modifier	Updated Time	Remark
1	Yang QuanWen	2019-06-25	The 5th Version

Customer provides parameters: IP (domain name), port number, SIM card APN

1、create a connection authentication

When a client creates a connection to the server, the client needs to provide username and password. The server needs to verify that username and password match.

Convention:

Vehicle client: username is prefix 0+imei, password is AES128(0+imei), client Id is the vehicle number.

2、 TOPIC related

When the client sends data to the server:

The TOPIC name of the vehicle client when sending data to the server is " bike" .

When the server sends data to the client:

The TOPIC name when the server sends data to the vehicle client is the vehicle number of the vehicle client.

FitRider Scooter

Network Module and Server Protocol

Instruction list：

No.	command	Transmission direction (Up: vehicle to server ; Down: Server to vehicle)	Remarks
1	power on	Down	
2	Scooter report confirmed power on	Up	
3	shutdown	Down	
4	Scooter report confirmed shutdown	Up	
5	Clear a single ride mileage	Down	
6	Clear a single riding mileage report	Up	
7	Clear single ride time	Down	
8	Clear a single ride time report	Up	
9	Clear total mileage	Down	

10	Clear the total mileage report	Up	
11	Clear the total riding time	Down	
12	Clear the total riding time report	Up	
13	Set the speed limit data	Down	Set the vehicle's max speed data
14	Set the speed limit data report	Up	
15	Get vehicle parameters	Down	
16	vehicle parameters report	Up	Status, speed limit value, current speed, vehicle body fault code, power, total riding time, total riding mileage, single riding

			time, single riding mileage
17	Vehicle failure report	Up	
18	Instruct upload latitude and longitude	Down	
19	GPS location report	Up	
20	Restart	Down	
21	Requested hardware information	Down	
22	Hardware information reporting	Up	
23	Upgrade firmware	Down	
24	Request SIM card number	Down	
25	Reply SIM card number	Up	
26	Enter	Down	

	transportation mode		
27	Heartbeat	Up	
28	Alarm	Down	
32	NFC Data	Up	
88	Universal response	Down	

---

power on (down)
Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 1, // integer ， unlock
}
Example: To start a scooter, just send {"a":1}.

Scooter report confirmed power on (up)
Topic ： bike
QoS ： 1
Payload:
{
"a": 2, // integer ， Confirm power on
"i": <12AB>, // string ， Vehicle number
"t": 1503046415, // long Actual power on time
"s": <status>, // integer ， status： 0：success 1：hardware fault 2：Firmware upgrading
"n": 0 // integer ， power on number
}

Shutdown (down)
Topic: Vehicle number
QoS ： 1 or 2
Payload:
{
"a": 3, // integer ， power off
}
Example: To turn off the scooter, just send {"a":3}

Scooter report confirms shutdown (up)
Topic ： bike
QoS ： 1
Payload:
{
"a": 4, // integer ， Confirm shutdown
"i": <12AB>, // string ， Vehicle number
"t": 1503046415, // long Actual shutdown time

"s": <status>, // integer ， Status: 0: Success 1 : Hardware failure 2 : Firmware upgrade
"n": 0 // integer ， Shutdown serial number
}

Clear one-time riding mileage (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 5 // integer
}
Example: To clear a single mileage, just send {"a":5}

Clearing a single riding mileage report (upward)
Topic: bike
QoS ： 1
Payload:
{
"a": 6 // integer
"i": <12AB>, // string ， vehicle number
"t": 1503046415, // long time
"s": <status>, // integer ， status 0:sucess 1:hardware fault 2:firmware upgrading 3:already later than the latest shutdown time
"n": 0 // integer serial number
}

Clear single ride time (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 7 // integer
}
Example: To clear a single ride time, just send {"a":7}.

Clearing a single ride time report (upward)
Topic: bike

QoS ： 0
Payload:
{
"a": 8 // integer
"i": <12AB>, // string ， vehicle number
"t": 1503046415, // long time
"s": <status>, // integer ， status: 0:sucess 1:hardware fault 2:firmware upgrading 3:already later than the latest shutdown time
"n": 0 // integer serial number
}

Clear total mileage (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 9 // integer
}
Example: To clear the total mileage, just send {"a":9}

Clear total mileage report (up)
Topic: bike
QoS ： 0
Payload:
{
"a": 10 // integer
"i": <12AB>, // string ， vehicle number
"t": 1503046415, // long time
"s": <status>, // integer ， status: 0:sucess 1:hardware fault 2:firmware upgrading 3:already later than the latest shutdown time
"n": 0 // integer serial number
}

Clear the total riding time (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{

"a": 11 // integer
}
Example: To clear the total ride time, just send {"a":11}.

Clear the total riding time report (up)
Topic: bike
QoS ： 0
Payload:
{
"a": 12 // integer
"i": <12AB>, // string ， vehicle number
"t": 1503046415, // long time
"s": <status>, // integer ， status: 0:sucess 1:hardware fault 2:firmware upgrading 3:already later than the latest shutdown time
"n": 0 // integer serial number
}

Set the speed limit data (down)
- Instruction format:

Topic: Vehicle number
QoS ： 1
Payload:
{
"a": 13, // integer
"k": 1500 // integer 0---1900
}

The K value and the speed roughly correspond to the following relationship:

600	4km/h
700	6 km/h
750	7 km/h
800	8 km/h
900	10 km/h
1000	11 km/h
1100	13 km/h
1200	15 km/h
1300	16 km/h
1400	18 km/h
1500	19 km/h
1550	20 km/h
1600	21 km/h

1800 23 km/h
1900 25 km/h
Example: To set the speed to 20km/h, just send {"a":13,"k":1550}

Set the speed limit data report (up)
Topic: bike
QoS ： 0
Payload:
{
"a": 14 // integer
"i": <12AB>, // string ， vehicle number
"t": 1503046415, // long time
"s": <status>, // integer ， status: 0:sucess 1:hardware fault 2:firmware upgrading 3:already later than the latest shutdown time
"n": 0 // integer serial number
}

Get the vehicle parameters (down)
• Instruction format:
Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 15 // integer
}
Example: To get the body parameters, just send {"a":15}

Vehicle parameter reporting (up)
Topic: bike
QoS ： 0
Payload:
{
"a": 16 , // integer
"i": <12AB>, // string ， vehicle number
" s ": 8 , // integer state
" k ": 8 , // integer speed limit value KM
" p ": 8 , // integer current speed KM
"e": 8 , // integer body fault code (0: no fault; 1: motor Hall fault; 2: speed throttle fault; 4: motor stalling; 5: overcurrent; 6: motor over temperature ;7: controller over temperature; 8: battery undervoltage; 9: battery overvoltage; 10: communication failure)

" b ": 8 , // battery (percentage of charge)
"y": 8 , // integer total riding time seconds
"q": 8 , // integer total riding mileage KM
"w": 8 , // integer single ride time seconds
"z": 8 // integer Single riding distance KM
"c": <status> // integer status 0:power on 1:shutdown
}

---

vehicle failure report (upward)
Topic: bike
QoS : 0
Payload:
{
"a": 17, // integer
"i": <12AB>, // string ， vehicle number
"e": 0 // integer (0: no fault; 1: motor Hall fault; 2: speed throttle fault; 4: motor stalling; 5: overcurrent; 6: motor over temperature; 7: controller over Temperature; 8: battery undervoltage; 9: battery overvoltage ; 10: communication failure)
"p": 0 // integer 0: no vibration 1: vibration
}

Indicates upload latitude and longitude (down)
Topic: Vehicle number
QoS : 0
Payload:
{
"a": 18 // integer , issue the gps command
}
Example: To get the latitude and longitude, just send {"a":18}, but this command is generally not used, because when the scooter is successfully positioned, it will be reported at the active interval. The default is 30 seconds. This interval can also be used. Set by instruction 33

GPS location report (up)
Topic: bike
QoS ： 1
Payload:
{
"a": 19, // integer
"i":<12AB>, //string, vehicle number

"g": "latitude,longitude,business_type,location_type",
//"3020.5887,12006.4229,1" business_type:0 Normal, 1 alert , location_type : 0 GPS,1 LBS
"i":<lockCode>, //string, vehicle number
"t": 1503046415 // long GPS reporting time
}

Restart (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 20 // integer
}
Example: To restart the network module, just send {"a":20}

Requested hardware information (down)

- Instruction format:

Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 21 // integer
}
Example: To get the network module hardware information, just send {"a":21}

Hardware information reporting (up)
Topic: bike
QoS ： 1
Payload:
{
"a": 22, // integer
"i":<12AB>, //string, vehicle number
"m": "V0.0.3|sim|1 | V0.0.1", // string hardware version number | SIM card number | scooter status | GPS version number
(Hardware version number, SIM card number, scooter status, GPS version number are separated by a vertical line |.
If the SIM card number is not available, the 0 is used. scooter status: 0: Powered on, 1: shutdown)

"t": 1503046415 // long reporting time
}

Upgrade firmware (down)
The server notifies the scooter to upgrade the firmware.
- Instruction format:

Topic: Vehicle number
QoS : 0
Payload:
{

"a": 23 , / / integer,
"l": 1000 , / / integer, firmware package distribution times
}
The scooter reports that the upgrade begins (up)
Topic : bike
QoS : 1
Payload:
{
"a": 66, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer, 0 can be upgraded , 1 no upgrade
}
Upgrade firmware data (down, up to 1024 bytes of firmware data at a time)
- Instruction format:

Topic: Vehicle number
QoS : 0
Payload:
{
"a": 67, // integer,
"x": 55, // integer, firmware package number
"l": 55, // integer, data length
"d": 012z / / firmware data hex
}
Scooter report confirms receipt of data (up)
Topic : bike
QoS : 1
Payload:
{
"a": 68, // integer ,
"i": "12AB", // string , vehicle number

"x": 55, // integer, firmware package number
"s": 0, // integer,0 The firmware package is successfully received (can send the next packet of data) 1 The firmware package fails to receive (the server needs to resend the packet data, and the module restarts after more than 5 failures to upgrade again)
}

After the scooter receives all the data packets, it reports the upgrade status (upstream)
Topic : bike
QoS : 1
Payload:
{
"a": 69, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer,0 upgrade succeeded 1 upgrade failed
}

Request SIM card number (down)
• Instruction format:
Topic: Vehicle number
QoS ： 0
Payload:
{
"a": 24 // integer
}
Example: To get the SIM card number, just send {"a":24}

Reply to the SIM card number (up)
• Instruction format:
QoS ： 0
Payload:
{
"a": 25, // integer
"i":<12AB>, //string, vehicle number
"h":123456789888888,//SIM card number
"t": 1503046415 // long Reply time
}

Heartbeat (Upstream module will send a blank data to the server for 5 seconds to maintain the connection between the module and the base station. The topic is bike5. The heartbeat packet such as battery power will be sent to the server every 2

minutes. The content is as follows)

- Instruction format:

Topic : bike
QoS : 1
Payload:
{
"a": 27, // heartbeat
"i": "12AB", // string , vehicle number
"b": <battery>, // integer, percentage of charge, from 0-100
"c": <status> // integer , status: 0 : boot, 1 : shutdown
}

Ringing (downstream, this command is not uploaded)
Topic: Vehicle number
QoS : 0
Payload:
{
"a": 28 // integer , ring
}
Example: If you want the network module horn to ring, just send {"a":28}.

Set user parameters (downstream)

- Instruction format:

Topic: Vehicle number
QoS : 0
Payload:
{
"a": 33, // integer
"u": 018.180.156.177,1844,123456,30 // string User parameters IP, PORT, vehicle number, GPS reporting interval
}

Example: I want to set the network module's new IP to 123.456.789.123, PORT to 1234, license plate number to 000003, GPS reporting interval to 45 seconds, only need to send {"a":33,"u":123.456.789.123,1234 , 000003, 45} is OK, this command will be actively connected to the new IP, PORT after the successful execution, this process needs to wait for a while, because the module has to be reconnected.

Scooter report confirms setting user parameters (upstream)
Topic : bike

QoS : 0
Payload:
{
"a": 34, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}
Get user parameters (downstream)

- Instruction format:

Topic: Vehicle number
QoS : 0
Payload:
{
"a": 35, // integer
}
Example: To get user parameters, just send {"a":35}

Scooter reporting user parameters (upstream)
Topic : bike
QoS : 0
Payload:
{
"a": 36, // integer ,
"i": "12AB", // string , vehicle number
"u": 018.180.156.177,1844,UNINET,123456,30 // string User parameter IP, PORT, vehicle number, GPS reporting interval

}
Headlight switching configuration (downstream)
Topic: Vehicle number
QoS : 0
Payload:
{
"a": 43, // integer , headlight switch
"j": 0// integer 0: command control 1: always on
}
Example:

If you want to set the headlights to be on, you only need to send {"a":43, "j":1} in the car off state, so that the headlights will be turned on before the car is turned on, and it is not commanded. Control, that is, sending the command 37 to control the light on and off is useless, the light will always be on

If you want to set the headlights to be controlled by the command, you only

need to send {"a":43, "j":0} in the car off state, so that the headlights are off when the car is turned on, you need to send the command 37 Control light is on and off

Headlight configuration report (upstream)
Topic : bike
QoS : 0
Payload:
{
"a": 44, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}

Headlight control (downstream)
Topic: Vehicle number
QoS : 0
Payload:
{
"a": 37, // integer , headlight control
"d": 0// integer 0: turn off the light 1: turn on the light

}
example:

This command only sends {"a":43 when the car is off, "j":0} is set to pass the car to control the light to be used, the command {"a":43,"j":0 } Once you set it up, you don't need to repeat the settings, the module will save automatically.

To light the headlights only need to send the command {"a":37,"d":1}, you only need to send the command {"a":37,"d":0} to turn off the headlights. The default headlights are off

Headlight control report (upward)
Topic : bike
QoS : 0
Payload:
{
"a": 38, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}
Vibration configuration (downstream)
Topic: Vehicle number
QoS : 0
Payload:

{
"a": 39, // integer , vibration configuration
"v": 0// integer 0: turn off vibration >0: vibration sensitivity
}
example:

The vibration here means that the car detects the vibration caused by the outside world in the closed state. When the number of consecutive vibrations reaches a certain value, the car horn will ring, and an alarm will be generated. At the same time, the car will open and enter the motor lock state, that is, the motor will block and prevent. When someone else taxis the car, when no vibration is detected within 120 seconds, the car will exit this state.

The value of 0 after this command v is to turn off the vibration detection. At this time, the external vibration car will not have any response. The specific sending command is {"a":39, "v":0}

The value after v is greater than 0 is the sensitivity to adjust the detection vibration. The larger the value, the less sensitive. The maximum value is 99. The default value is 25. If you want to set the sensitivity value to 10, you only need to send the command {"a":39,"v ":10}

Vibration configuration report (upstream)
Topic : bike
QoS : 0
Payload:
{
"a": 40, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}
Kilometer miles switching configuration (downstream)
Topic: Vehicle number
QoS : 0
Payload:
{
"a": 41, // integer, kilometer mile switch
"f": 0// integer 0: kilometer 1: mile
}
example:

If you want to set the meter display speed in kilometers, you only need to send {"a":41, "f":0} when the car is off.

If you want to set the meter display speed in miles, you only need to send {"a":41, "f":1} when the car is off.

Kilometer miles switching configuration report (upstream)
Topic : bike
QoS : 0
Payload:
{
"a": 42, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}

Boot test configuration
- Instruction format:

Topic: Vehicle number
QoS : 0
Payload:
{
"a": 51, // integer
"w": 0 // string 0: close the self-test program 1: turn on the self-test program
}

Note: If you re-power the scooter and the horn will ring twice, the meter will light up, indicating that you have entered the self-test program. At this time, shake the scooter within ten seconds until the tail light is on, indicating that the self-test is successful. A long ring of the speaker indicates that the self-test failed. As long as the self-test succeeds once, the self-test program will never enter the program again; here, you can also enable or disable the self-test program by sending the 51 command.

Boot test configuration report (upstream)
Topic : bike
QoS : 0
Payload:
{
"a": 52, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure

Set APN
- Instruction format:

Topic: Vehicle number

QoS : 0
Payload:
{
"a": 53, // integer
"z":AT+QICSGP=15,1,"apn","username","password",0 // string
}

example:

I want to set the SIM card APN to 123, the username to 456, and the password to 789. Just send {"a":53, "z":AT+QICSGP=15,1,"123","456","789" , 0} is fine.

If the username and password are not available, the corresponding location is empty. For example, the APN is 123, the username is empty, and the password is empty. You only need to send {"a":53, "z":AT+QICSGP=15,1,"123 ","","",0}

Scooter report confirms setting APN parameters (upstream)
Topic : o
QoS : 0
Payload:
{
"a": 54, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // integer 0: success 1: failure
}

Scooter reporting status parameters (upstream)
Topic : o
QoS : 0
Payload:
{
"a": 55, // integer ,
"i": "12AB", // string , vehicle number
"s": 0, // 0: The scooter didn't fall down 1: The scooter fell
"r": 36.8 //Battery voltage Unit V
}
Note: The above parameters indicate that the scooter does not fall, the battery voltage is 36.8V

Special ringing (downstream, this command is not uploaded)
Topic: Vehicle number
QoS : 0
Payload:

{
"a": 58 , // integer , ring
"v": 5, //number of rings
"i": 100, / / horn ringing time unit ms
"L": 200 //Time when the speaker does not ring Unit ms
}
Note: The above parameters indicate that the speaker will ring for 100ms, then it will not ring for 200ms, so alternate 5 times.

a:action type
i: vehicle number (10 digit code)
t: time cut (month-day-hour-minute-second (long type))
s: state
d: headlight control（0: off 1: on）
c: status (0: power on; 1: shutdown)
k: speed limit data KM
h: SIMcard number
p: current speed KM
y: total riding time seconds
l: Data length
x: number
d: data
q: Total riding mileage KM
w: single riding time seconds
z: Single riding mileage KM
p: vibration state 0: no vibration, 1: vibration
e: Vehicle fault code (0: no fault; 1: motor Hall fault; 2: speed throttle fault; 4: motor stalling; 5: overcurrent; 6: motor over temperature; 7: drive over temperature; 8: battery undervoltage; 9: battery overvoltage; 10: communication failure)
n: serial number (unlocking service serial number)
b: power (% of electricity)
u: user parameter
g: GPS information (latitude and longitude + report type)
m: hardware information
(Hardware version number | SIM card number | scooter status | GPS version number
(Hardware version number, SIM card number, scooter status, GPS version number are separated by |.
Empty ,if the SIM card number is not available.
vehicle status: 0: turned on, 1: turned off))
o:topic