

Ninebot ES Communication Protocol

Document revision records:

Version No.	Revised by	Time	Remarks
V1.4	Li Xingle	2018-3-16	Memory control table modified as the volume production version, center control modified as the external equipment
V1.5	Lester(Li Xingle)	2018-9-17	Add 0x80,0x81 bool function

Table of Contents

Ninebot ES Communication Protocol.....	1
1. Summary of communication protocol.....	2
2. Protocol format.....	2
2.1 Serial bus packet format.....	2
2.2 ID definition.....	2
2.3 Packet length.....	3
2.4 Command word definition.....	3
2.5 Data index description.....	4
2.6 Data segment description.....	4
2.7 Serial port check and description.....	5
3. Use case of Communication Protocol.....	5
3.1 Serial port reads memory control table.....	5
3.2 Serial port writes memory control table.....	6
4. Description of scooter memory control table.....	6
5. Description of battery memory control.....	13
6. Appendix1 - Serial port check code.....	16
7. Appendix 2 - Error code meaning and solution.....	17

This document is now in the public domain

Released by ScooterHacking.org on 27/9/2019

Original URL: <https://cloud.scooterhacking.org/release/nbdoc.pdf>

1. Summary of communication protocol

This protocol defines the general protocol format, with which Ninebot electronic control system realizes multi-node communication through physical serial ports and Bluetooth serial ports. Configuration of the serial port is 115200, 8 bytes, without verification and with 1 stop bit. Information query, control and parameter modification instruction can be realized through the read-write operation of the “memory control table” stored in the controller. Basic storage unit of the memory control table is 2 bytes of signed integer data (short).

All data longer than 1 byte is subject to the low-priority transmission and storage form.

Macro definition related to the scooter communication protocol is stored in the NinebotCmd.h file based on C programming language.

Note: Compared with the previous communication protocol *Bluetooth Serial Port Protocol*, ID and command word definition are changed greatly and CAN bus protocol is added in this version.

2. Protocol format

2.1 Serial bus packet format

Byte	Frame header (2Byte)		Packet length (1Byte)	Source ID (1Byte)	Target ID (1Byte)	Command word (1Byte)	Data index (1Byte)	Data segme nt [N Byte]	Checksum (2Byte)	
Value	0x5A	0xA5	N	0~63	0~63	0~255	0~255	Data	CsumL	CsumH

2.2 ID definition

Point-to-point communication

Name	ID
Master control of electric scooter (ES)	0x20
Bluetooth instrument of ES	0x21
Built-in battery of ES	0x22
External battery of ES	0x23
PC upper computer connected through serial port / CAN debugger/IoT equipment	0x3D
Mobile phone linked through Bluetooth serial port (BLE)	0x3E

Broadcast ID

Name	ID
Entire network broadcast	0x00

2.3 Packet length

Serial port data packet

Packet length is equal to the data segment length N

2.4 Command word definition

Control table access instructions

Definition of instruction	Function	Value
CMD_CMAP_RD	Read control table data	0x01
CMD_CMAP_WR	Write control table data, with reply.	0x02
CMD_CMAP_WR_NR	Write control table data, without reply.	0x03
CMD_CMAP_ACK_RD	Response packet to instruction reading.	0x04
CMD_CMAP_ACK_WR	Response packet to instruction writing	0x05

Instructions related to data download

Definition of instruction	Function	Value
CMD_IAP_BEGIN	Start frame of firmware download / with reply from scooter.	0x07
CMD_IAP_TRANS	Data frame of firmware download / with reply from scooter.	0x08
CMD_IAP_VERIFY	Check frame of firmware download / with reply from scooter.	0x09
CMD_MCU_RESET	Chip reset instruction / without reply from scooter.	0x0A
CMD_IAP_ACK	Response frame of firmware download.	0x0B

Instructions related to external equipment communication

define	function	command

<p>CMD_SCO_HEARTBEAT</p>	<p>Two-way heartbeat command Scooter need to receive a command which index = 0x7c,data = 0x7c as a heartbeat, if 30 second The scooter needs to receive the heartbeat command of index=0x7C and data=0x7C. If the heartbeat is not received for 30 seconds, it will alarm, and motor will be locked after 1 minute. For example, IoT sends heartbeat every 4S; The mainboard of the scooter sends out 13 bytes messages every 500ms: #define LOCK_STATUS 0 //lock status #define CHARGE_STATUS 1 //charge status #define BAT1_VOLT 2 //battery1 voltage V #define BAT2_VOLT 3 // battery2 voltage V #define BAT1_SOC 4 // battery1 State of Charge % #define BAT2_SOC 5 // battery2 State of Charge % #define BODY_VOLT_SOC 6 //total State of Charge % #define DRY_VOLT 7 //Voltage V #define ES_SPEED 8 //speed 0.1km/h #define BAT1_TEMP1 9 //Battery1 temperature -20°C #define BAT1_TEMP2 10 // Battery1 temperature -20°C #define BAT2_TEMP1 11 // EX. Bat temperature -20°C #define BAT2_TEMP2 12 // EX. Bat temperature -20°C</p>	<p>0x55</p>

2.5 Index description

In the instruction packet which reads and writes the memory control table and response packet with read operation, data index indicates the offset address of the accessed data in the table; in the response packet (command word 0x05) with response writing operation, data index 0 indicates successful write. Other values are defined as follows:

Index definition	Value	Meaning
CMD_CMAP_WRERROR_AUTH	0x01	No write permission for the write-in address.
CMD_CMAP_WRERROR_BUSY	0x02	The control table is under operation, and no write-in is allowed.
CMD_CMAP_WRERROR_SIZE	0x03	Write-in data is out of scope.
CMD_CMAP_WRERROR_FORM	0x04	Write-in data is in wrong form.

During firmware download (IAP), data index indicates natural number of the data packet under

continuous data download; when the firmware downloads related instruction response packets, data index 0 indicates download successful. Other values are defined as follows:

Index definition	Value	Meaning
IAPERROR_SIZE	0x01	Firmware is over-sized
IAPERROR_ERASE	0x02	Erase Flash failed
IAPERROR_WRITEFLASH	0x03	Write Flash failed
IAPERROR_UNLOCK	0x04	Scooter is unlocked / firmware can be updated
IAPERROR_INDEX	0x05	Data index error
IAPERROR_BUSY	0x06	IAP is busy (such as writing in Flash)
IAPERROR_FORM	0x07	Data format error (length of the data sent is not an integer multiple of 8)
IAPERROR_CRC	0x08	Data verification failure
IAPERROR_RESET	0x09	Other errors

2.6 Data segment description

Control table access instructions

Definition of instruction	Data segment content	Data segment length N (byte)
CMD_CMAP_RD	Length of data to be read (unit: Byte, a frame of data reads up to 255 bytes)	1
CMD_CMAP_WR	Write data content (maximum 255 bytes of data for a single write)	0~N
CMD_CMAP_WR_NR		
CMD_CMAP_ACK_WR	1- write successfully, 0- write failed	0
CMD_CMAP_ACK_RD	Data read	0~N

Instructions related to firmware/data download

Definition of instruction	Data segment content	Data segment length N (byte)
CMD_IAP_BEGIN	4 bytes (u32 type), length of firmware to be transmitted	4
CMD_IAP_TRANS	Firmware content	0 - N (integral multiple of 8; 128 if there is no special requirement)
CMD_IAP_VERIFY	4 bytes download data checksum	4

CMD_MCU_RESET	N/A	0
---------------	-----	---

2.7 Serial port check and description

Calculation method: add all the data **except the header** and then invert the result. The lower 16 bits shall prevail.

3. Use case of Communication Protocol

3.1 Serial port reads memory control table

If the upper computer wants to read current temperature of the scooter, as the index address of scooter temperature in the memory control table is 0x3E (refer to section 4.1), instruction type shall be “read instruction”, namely 0x01 and the data index is 0x3E. As the basic unit of the memory control table is a 16-bit integer data, the read length is 2 bytes. Data segment has only one byte with the value of 2 and the frame length is 1. Source ID is ID of the upper computer, 0x3D. Target ID is main control panel ID of the scooter, 0x20 (refer to section 2.3). Instruction packet sent by the upper computer is:

```
5A A5 01 3D 20 01 3E 02 60 FF
```

Upon receiving the data packet, the scooter will return current body temperature value. The main control panel will return data to the upper computer according to the source ID of the packet. Therefore, the target ID is the upper computer ID, 0x3D and source ID is main control panel ID of the scooter, 0x20. Instruction type is 0x05 (read response) and data index is 0x3E (body temperature). Data segment is the body temperature (as it is a 16-bit data, the data segment has two bytes with the low order in front. Assume current body temperature is 31.8°C, as the temperature unit is 0.1°C, the value read is 318, namely 0x136 in the hexadecimal system. Therefore, two bytes of the data segment are 36 01) with the frame length of 2. The response packet returned by the scooter is as follows:

```
5A A5 02 20 3D 04 3E 36 01 27 FF
```

In practical application, in order to save communication bandwidth, multi-data can be read in one time, such as:

to read 14-byte scooter serial number in one time. Start address of the serial number in the memory table is 0x10, with a size of 14Byte, from 0x10 to 0x16. During reading, data index of the instruction packet is set as 0x10 and the read length (namely data segment) is set as 14, the scooter will return continuous 14byte data segment, namely scooter serial number. Instruction packet sent by the upper computer is:

```
5A A5 01 3D 20 01 10 0E 82 FF
```

3.2 Serial port writes memory control table

When data is about to be written to the memory control table, instruction type of the instruction packet shall be set as “write instruction”, namely 0x02 (write instruction with response) or 0x03 (write instruction without response). For “write instruction with response”, the lower computer will return a response instruction indicating successful writing upon receiving the “write instruction”; for “write instruction without response”, the lower computer will not return such instruction, whether receiving the instruction or not. The former is used for write-in of some important data and the upper computer needs to confirm whether the lower computer receives the data or not, such as write-in of scooter speed limit value; the latter is used for the occasion having high requirement on real-time performance but paying less attention to the write-in, such as write-in of remote control target speed value under remote control mode.

For example: write scooter speed limit value 10kmeter/h under the speed limit mode with response.

Index of the speed limit value is 0x74 with the unit of 0.1km/h. So, the value written is 100, namely 0x0064 in the hexadecimal system. As the basic unit of the memory control table is a 16-bit integer data with the low order in front during writing, the data segment to be written is 0x64 0x00, and the whole instruction packet is:

5A A5 02 3D 20 03 74 64 00 C5 FE

In case of successful writing, the response packet returned by the lower computer is as follows:

5A A5 01 20 3D 05 74 01 27 FF

Like read operation, multi-data can also be written in the write operation, such as writing of Bluetooth pairing code 123456:

5A A5 06 3D 20 03 17 01 02 03 04 05 06 6D FF

4. Description of scooter memory control table

Information and control parameters of the controller form a one-dimensional control table. The upper computer gains scooter status and interacts with the scooter through read-write operation of the memory control table. Control parameters in the control table can be saved in case of power off.

“Data index” in the instruction refers to the data index of the memory control table.

Note: 0xb0~0xbd content is the commonly used scooter data, which is the same as partial content in 0x10~0x6F. For example, 0xbb/NB_QUK_BODY_TEMP and 0x3e/NB_INF_BODY_TEMP have the same content related to the scooter temperature. In this way, commonly used scooter information is available through one read instruction.

Content definition of the control table is as follows:

Address	Name	Command item	Type	Permis sion ^[1]	Initial value
---------	------	--------------	------	-------------------------------	------------------

0x00~ 0x0F	Reserved	Reserved	--	--	--
0x10~ 0x16	NB_INF_SN	Scooter serial number, 14Byte	7*2 U8	R	--
0x17~ 0x19	NB_INF_BT PASSWORD	Bluetooth pairing code [2], 6Byte	3*2 U8	R/W	000000
0x1A	NB_INF_FW_VERSI ON	Firmware version of master control [3]	U16	R	--
0x1B	NB_INF_ERROR	Error code	U16	R	0
0x1C	NB_INF_ALARM	Alarm code	U16	R	0
0x1D	NB_INF_BOOL	Boolean state word [4]	U16	R	--
0x1E	NB_INF_WORK_SYS	Current operation system, 1 or 2			--
0x1F	NB_INF_WORKMO DE	Current operation mode, 0 NORMAL; 1 ECO; 2 SPORT	U16	R	--
0x20	NB_INF_BTC_1	Volume of storage battery 1	U16	R	--
0x21	NB_INF_BTC_2	Volume of storage battery 2	S16	R	--
0x22	NB_INF_BTC	Battery percentage of the scooter, 0- 100	S16	R	--
0x23		Reserved	S16	R	--
0x24	NB_INF_ACTUAL_M IL	Actual remaining mileage, unit: 10m	S16	R	--
0x25	NB_INF_PRD_RID_ MIL	Predicted remaining mileage, 10m	S16	R	--
0x26	NB_INF_SPEED	Current speed, 0.1km/h	S16	R	--
0x27		Reserved			--
0x28		Reserved			--
0x29	NB_INF_RID_MIL_L	Lower 16 bits of the total mileage, m	S16	R	--
0x2A	NB_INF_RID_MIL_H	Higher 16 bits of the total mileage, m	S16	R	--
0x2B		Reserved	S16	R	--
0x2C		Reserved	S16	R	--
0x2D		Reserved	S16	R	--
0x2E		Reserved	S16	R	--
0x2F	NB_INF_RID_MIL_S IG	Single mileage, 10m	S16	R	--
0x30		Reserved	S16	R	--
0x31		Reserved	S16	R	--
0x32	NB_INF_RUN_TIM_ L	Lower 16 bits of the total operation time, sec	S16	R	--
0x33	NB_INF_RUN_TIM_ H	Higher 16 bits of the total operation time, sec	S16	R	--
0x34	NB_INF_RID_TIM_L	Lower 16 bits of the total riding time, sec	S16	R	--

0x35	NB_INF_RID_TIM_H	Higher 16 bits of the total riding time, sec	S16	R	--
0x36		Reserved			--
0x37		Reserved			--
0x38		Reserved			--
0x39		Reserved			--
0x3a	NB_INF_RUN_TIM_SIG	Single operation time, sec	S16	R	--
0x3b	NB_INF_RID_TIM_SIG	Single riding time, sec	S16	R	--
0x3c		Reserved	S16	R	--
0x3d		Reserved	S16	R	--
0x3e	NB_INF_BODY_TEMP	Scooter temperature, 0.1°C	S16	R	--
0x3f	NB_INF_BAT1_TEMP	Battery 1 temperature, 0.1°C	S16	R	--
0x40	NB_INF_BAT2_TEMP	Battery 2 temperature, 0.1°C	S16	R	
0x41	NB_INF_MOS_TEMP	MOS pipe temperature	S16	R	
0x42~0x46		Reserved			
0x47	NB_INF_DRV_VOLT	System driving voltage, 0.01V, also supply voltage of the central control	S16	R	--
0x48~0x4f		Reserved	S16	R	--
0x50	NB_INF_BAT2_TEMP2	External battery temperature 2, 1°C	S16	R	
0x51		Reserved	S16	R	--
0x52		Reserved	S16	R	--
0x53	NB_INF_MOT11_CURT_P	Motor phase current, 0.01A	S16	R	--
0x54		Reserved	S16	R	--
0x55~0x60	Reserved	Reserved	--	--	--
0x61		Reserved	S16	R	--
0x62		Reserved	S16	R	--
0x63		Reserved	S16	R	--
0x64		Reserved	S16	R	--
0x65	NB_INF_AVRSPEED	Average speed, 0.1km/h	S16	R	--
0x66	NB_INF_VER_BMS2	External battery version number [3]	U16	R	--
0x67	NB_INF_VER_BMS	Built-in version number [3]	U16	R	--
0x68	NB_INF_VER_BLE	Instrument panel firmware version	U16	R	--

		number [3]			
0x69~ 0x6f	Reserved	Reserved	--	--	--
0x70	NB_CTL_LOCK	Lock [5], send 1 for locking, scooter reset automatically	S16	R/W	0
0x71	NB_CTL_UNLOCK	Unlock [5], send 1 for unlocking, scooter reset automatically	S16	R/W	0
0x72	NB_CTL_LIMIT_SPD	Speed limit or speed limit release	S16	R/W	250
0x73	NB_CTL_NOMALSPEED	Speed limit value in normal mode, 0.1km/h	S16	R/W	150
0x74	NB_CTL_LITSPEED	Speed limit value in speed limit mode, 0.1km/h	S16	R/W	60
0x75	NB_CTL_WORKMODE	Operating mode: 0, NORMAL; 1, ECO; 2, SPORT	S16	R/W	0
0x76		Reserved	S16	R/W	0
0x77	NB_CTL_ENGINE	Start or shut down the engine	S16	R/W	0
0x78	NB_CTL_REBOOT	Restart system [6], write 1 for restart	S16	R/W	0
0x79	NB_CTL_POWEROFF	Shutdown [6], write 1 for shutdown	S16	R/W	0
0x7a		Reserved	S16	R/W	0
0x7b		Reserved	S16	R/W	0
0x7c	NB_CTL_CRUISE	Cruise control switch: 0 Off, 1 On.	S16	R/W	0
0x7e	Looking for scooter	0 :off 1:on (headlight and taillight will flash, and beep) , work once a command sent	S16	W	0
0x80	NB_CTL_FUN_BOOL_1	function setup bool 【7】	S16	R/W	0
0x81	NB_CTL_FUN_BOOL_2	function setup bool 【7】	S16	R/W	0
0x90	NB_CTL_HEADLIGHT_IOT_SW	headlight control on or off	S16	R/W	1
0x91	NB_CTL_BEEP_ALARM_SW	Beep alarm on or off	S16	R/W	1
0x92	NB_CTL_BEEP_TOTAL_SW	Beep control switch	S16	R/W	1
0xb0	NB_QUK_ERROR	Error code [10]	S16	R	0
0xb1	NB_QUK_ALARM	Alarm code [9]	S16	R	0
0xb2	NB_QUK_BOOL	Boolean state word [4]	S16	R	0
0xb3	NB_QUK_BTC_BAT1_2	Volume of battery 1 and battery 2. Lower 8 bits for battery 1 and higher 8 bits for battery 2, percentage 0-100	S16	R	0
0xb4	NB_QUK_BTC	Battery volume of the scooter, 0-100	S16	R	0
0xb5	NB_QUK_SPEED	Current speed, 0.1km/h	S16	R	0

0xb6	NB_QUK_AVRSPPEED	Average speed, 0.1km/h	S16	R	0
0xb7	NB_QUK_RID_MIL_L	Lower 16 bits of the total mileage, m	S16	R	0
0xb8	NB_QUK_RID_MIL_H	Higher 16 bits of the total mileage, m	S16	R	0
0xb9	NB_QUK_RID_MIL_SIG	Single mileage, 10m	S16	R	0
0xba	NB_QUK_RUN_TIM_SIG	Single operation time	S16	R	0
0xbb	NB_QUK_BODY_TEMP	Scooter temperature, 0.1°C	S16	R	0
0xbc	NB_QUK_CRT_LIT_SPEED	Current speed limit value, with the lower 8 bits for current speed limit value and higher 8 bits for the full range value, unit: 0.1km/h	S16	R	
0xbd	NB_QUK_SYS_POWER	Scooter power, unit: W	S16	R	
0xbe	NB_QUK_ALARM_DELAY	Alarm code for delay reset	S16	R	
0xbf	NB_QUK_PRD_RID_MIL	Predicted remaining mileage, 10m	S16	R	
0xc0~0xc5		Reserved	S16	R	0
		Reserved	S16	R	0
0xc6	NB_ONE_LED_MODE	Display mode of chassis lamp strip [8]			
0xc7	Reserved	Reserved	S16	R/W	0
0xc8	NB_LED_COLOR1	Color of chassis lamp strip 1 [8]	2*S16	R/W	0xA0F0 Blue
0xca	NB_LED_COLOR2	Color of chassis lamp strip 2 [8]	2*S16	R/W	0x50F0 Green
0xcc	NB_LED_COLOR3	Color of chassis lamp strip 3 [8]	2*S16	R/W	0x00F0 Red
0xce	NB_LED_COLOR4	Color of chassis lamp strip 4 [8]	2*S16	R/W	0xC8F0 Purple
0xd0~0xd9	Reserved	Reserved	S16	R	0
0xda	NB_CPUID_A		U16	R	0
0xdb	NB_CPUID_B		U16	R	0
0xdc	NB_CPUID_C		U16	R	0
0xdd	NB_CPUID_D		U16	R	0
0xde	NB_CPUID_E		U16	R	0
0xdf	NB_CPUID_F		U16	R	0

0xe0~0xff	Reserved	Reserved	--	--	--
-----------	----------	----------	----	----	----

Note:

- [1] Read-write access permission, with R meaning read only and R/W meaning read-write.
- [2] The Bluetooth pairing code default is 000000 and there is no need to input such pairing code in case of Bluetooth connection by default. If the pairing code is set as non-0 figures, pairing code shall be input during Bluetooth connection (there is no need of input in case of Bluetooth Dongle connection in PC). After successful setting of the pairing code, Reinitialize Bluetooth mark (index 0x4D) must be set as 1. On-board Bluetooth will be reinitialized in the next start and the pairing code will work.
- [3] Definition of the firmware version is: 16-bit figures are divided into 4 groups with 4-bit figures in each group. The highest 4 digits are PCB version number internally reserved and the rest 12 digits are firmware version number, such as 0x1101, indicating PCB version number 1 and firmware version number V1.0.1.
- [4] Each digit of Boolean variable indicates one variable of the scooter status. 1 indicates sport status and the specific definition is:

//BOOL variable Mark definition:

Definition	Bit Position	Instruction
NB_BOOLMARK_LIMITSPEED	((unsigned short)0x0001)	Speed limit or not
NB_BOOLMARK_LOCK	((unsigned short)0x0002)	Scooter lock or not
NB_BOOLMARK_BEEP	((unsigned short)0x0004)	Buzzer alarm or not
Reserved	((unsigned short)0x0008)	
Reserved	((unsigned short)0x0010)	
Reserved	((unsigned short)0x0020)	
Reserved	((unsigned short)0x0040)	
Reserved	((unsigned short)0x0080)	
Reserved	((unsigned short)0x0100)	
NB_BOOLMARK_BAT2_IN	((unsigned short)0x0200)	Battery 2 inserted
Reserved	((unsigned short)0x0400)	
NB_BOOLMARK_ACT	((unsigned short)0x0800)	Scooter activated or not
Reserved	((unsigned short)0x1000)	
Reserved	((unsigned short)0x2000)	
Reserved	((unsigned short)0x4000)	
Reserved	((unsigned short)0x8000)	

[5] Locking or unlocking is available only in non-riding mode. Corresponding mark will be reset after locking or unlocking ends.

[6] Restart and shutdown are available only in non-riding mode. Corresponding mark will be reset after completion of the operation.

[7] Function setting bool variables:



Define	Bit Position	Instruction	Default
NB_FUN_BOOLMARK_HEADLED_ALWAYS_ON	((unsigned short)0x0001)	Headlight always on/off	0
NB_FUN_BOOLMARK_HEADLED_FLASH	((unsigned short)0x0002)	Headlight flash	0
NB_FUN_BOOLMARK_BACKLED_ALWAYS_ON	((unsigned short)0x0004)	Taillight always on/off	0
NB_FUN_BOOLMARK_BACKLED_FLASH	((unsigned short)0x0008)	Taillight flash	0
NB_FUN_BOOLMARK_BAT2LED_ALWAYS_ON	((unsigned short)0x0010)	Ex. Battery light on/off	1
NB_FUN_BOOLMARK_NO_ALARM_AT_LOCK	((unsigned short)0x0020)	No alarm when locked	0
NB_FUN_BOOLMARK_SPEED_IN_MPH	((unsigned short)0x0040)	Use mph units	0
NB_FUN_BOOLMARK_UNIT_SW	((unsigned short)0x0080)	Display unit or not 1: yes 0:no	1
NB_FUN_BOOLMARK_SPEED_ICON_SW	((unsigned short)0x0100)	Display speed or not	1
NB_FUN_BOOLMARK_PANNEL_SW	((unsigned short)0x0200)	Panel display or not 1:on 0:off	1
NB_FUN_BOOLMARK_BLUETOOTH_SW	((unsigned short)0x0400)	Bluetooth broadcast or not When on,bt icon cannot control; When off,bt icon can define status:flash or always	0
NB_FUN_BOOLMARK_BT_ICON_ALWAYS_SW	((unsigned short)0x0800)	Bluetooth icon always on or off	0
NB_FUN_BOOLMARK_BT_ICON_FLASH_SW	((unsigned short)0x1000)	Bluetooth icon flash or off	0
NB_FUN_BOOLMARK_FAULT_ICON_SW	((unsigned short)0x2000)	Error icon on or off	1

NB_FUN_BOOLMARK_TEMP_ICON_SW	((unsigned short)0x4000)	Temperature error icon on or off	1
NB_FUN_BOOLMARK_BAT_ICON_SW	((unsigned short)0x8000)	Display battery level on or off	1
0x81:			
NB_FUN_BOOLMARK_BUTTON_SPDMODE_SW	((unsigned short)0x0001)	Button can change mode or not	1
NB_FUN_BOOLMARK_MODE_ICON_SW	((unsigned short)0x0002)	Display speed mode or not	1

[8] Chassis lamp strip control:

Define	Data	Mode	The number of color
MINI_BLED_MODE_CLOSE	0	Close	N/A
MINI_BLED_MODE_1COLOR	1	Single color breathing	NB_LED_COLOR1
MINI_BLED_MODE_ALLCOLOR	2	Full-color breathing	Non-adjustable
MINI_BLED_MODE_2C_APART	3	Double color apart	NB_LED_COLOR1&2
MINI_BLED_MODE_ALLC_APART	4	Full-color apart	Non-adjustable
MINI_BLED_MODE_1C_STAR	5	Single color flash	NB_LED_COLOR1
MINI_BLED_MODE_ALLC_STAR	6	Full-color flash	Non-adjustable
MINI_BLED_MODE_POLICE	7	Alarm mode 1	NB_LED_COLOR1&2
MINI_BLED_MODE_POLICE_2	8	Alarm mode 2	NB_LED_COLOR1&2
MINI_BLED_MODE_POLICE_3	9	Alarm mode 3	NB_LED_COLOR1&2&3&4

[9] Alarm code definition:

ALARM_CODE_LOCKED	9	//alarm for being pushed in lock mode
ALARM_CODE_HIGH_VOL	12	// alarm for high recoil energy during brake

[10] Error code: See chapter 7 for details.

5. Description of battery memory control

Address	Name	Command item	Type	Permis sion ^[1]	Initial value
0x00~ 0x0F	Reserved	Reserved	--	--	--
0x10~ 0x16	BAT_SN	Battery serial number, 14Byte	7*2 U8	R	--
0x17	BAT_SW_VER	Battery hardware and firmware version	U16	R	0
0x18	BAT_CAPACITY	Design capacity of the battery, mAh	U16	R	0
0x19	BAT_TOTAL_CAPACITY	Full charge capacity of the battery, mAh	U16	R	0
0x1A	BAT_DESIGN_VOLTAGE	Design voltage of the battery, 10mV	U16	R	0
0x1B	BAT_CYCLE_TIMES	Cycle times of the battery	U16	R	0
0x1C	BAT_CHARGE_TIMES	Battery charging times	U16	R	0
0x1D	BAT_CHARGE_CAP_L	Lower 16 bits of the accumulative charge capacity of the battery, 10mAh	U16	R	0
0x1E	BAT_CHARGE_CAP_H	Higher 16 bits of the accumulative charge capacity of the battery, 10mAh	U16	R	0
0x1F	BAT_OVER_DISCHARGE_TIMES	Battery overflowing and over- discharging times, with the lower 8 bits indicating the overflowing times and higher 8 bits indicating over- discharging times	U16	R	0
0x20~ 0x2f	Reserved	Reserved	U16	R	--
0x30	BAT_FUN_BOOLEAN	Boolean variable [1]	S16		
0x31	BAT_REMAINING_CAP	Current residual capacity, mAh	U16	R	--
0x32	BAT_REMAINING_CAP_PERCENT	Current residual capacity percentage, 0-100	S16	R	--
0x33	BAT_CURRENT_CUR	Current current, 10mA	S16	R	--
0x34	BAT_VOLTAGE_CUR	Current voltage, 10mV	S16	R	--
0x35	BAT_TEMP_CUR	Temperature, with the higher 8 bits and lower 8 bits indicating temperatures of two temperature sensors respectively, 0-119 corresponding to -20 - 99°C	S16	R	--

0x36	BAT_BALANCE_STATU	Balanced open status of the protective plate	S16	R	--
0x37	BAT_ODIS_STATE	Undervoltage condition of battery cell	S16	R	--
0x38	BAT_OCHG_STATE	Overvoltage condition of battery cell	S16	R	--
0x39	BAT_CAP_COULO	Coulombmeter capacity	S16	R	--
0x3A	BAT_CAP_VOL	Voltmeter capacity	S16	R	--
0x3B	BAT_HEALTHY	Health degree	S16	R	--
0x3C~ 3f	Reserved	Reserved	S16	R	--
0x40~ 0x4f	BAT_CORE_VOLT_1~ BAT_CORE_VOLT_16	Single cell voltage	S16	R	

The meanings of Boolean states meaning:

Boolean state	Bit Position	Instruction
BMS_BOOLMARK_PASSWORD	((unsigned short)0x0001)	Battery check code right or not
BMS_BOOLMARK_ACT	((unsigned short)0x0002)	Battery activated or not
BMS_BOOLMARK_CHG_PROTECT	((unsigned short)0x0004)	Battery charging protected or not
BMS_BOOLMARK_CMOS	((unsigned short)0x0008)	Charging MOS pipe open or not
BMS_BOOLMARK_WRITE_CMD	((unsigned short)0x0010)	Write lock open or not
BMS_BOOLMARK_DISCHARGE	((unsigned short)0x0020)	Discharging or not
BMS_BOOLMARK_CHARGE	((unsigned short)0x0040)	Charging or not
BMS_BOOLMARK_CHARGERIN	((unsigned short)0x0080)	Charger inserted or not
BMS_BOOLMARK_DISCOVER	((unsigned short)0x0100)	Over discharging or not
BMS_BOOLMARK_CHGOVER	((unsigned short)0x0200)	Over charging or not
BMS_BOOLMARK_VOERTEMP	((unsigned short)0x0400)	Over temperature or not
BMS_BOOLMARK_TEST_MODE	((unsigned short)0x0800)	Test mode or not
Reserved	((unsigned short)0x1000)	
Reserved	((unsigned short)0x2000)	
Reserved	((unsigned short)0x4000)	
Reserved	((unsigned short)0x8000)	

6. Appendix1- Serial port check code

Serial port check and C code

```
uint16_t CheckSum1ByteIn2ByteOut(uint8_t* data, int len)
{
    uint16_t checksum = 0;
    ASSERT(NULL != data);
    for (int i = 0; i < len; i++) {
        checksum += data[i];
    }
    checksum = ~checksum;
    return checksum;
}
```

7. Appendix 2- Error code meaning and solution

Error code	Description and solution
10	Abnormal communication between instrument panel and main control panel; please check related wiring.
11	Abnormal A phase current sampling of the motor, please check the main control panel.
12	Abnormal B phase current sampling of the motor, please check the main control panel.
13	Abnormal C phase current sampling of the motor, please check the main control panel.
14	Abnormal accelerator Hall, please check accelerator twist grip, instrument panel and related wiring.
15	Abnormal brake Hall, please check brake twist grip, instrument panel and related wiring.
16	Built-in battery abnormality in MOS switch, please check the main control panel.
17	External battery abnormality in MOS switch, please check the main control panel.
18	Abnormal motor Hall, please check motor, main control panel and related wiring.
19	Built-in battery abnormality in voltage detection, please check built-in battery, main control panel and related wiring.
20	External battery abnormality in voltage detection, please check external battery, main control panel and related wiring.
21	Built-in battery abnormality in communication, please check built-in battery, main control panel and related wiring.
22	Built-in battery password is wrong, please replace the battery.
23	Built-in battery is in default serial number, please replace the battery.
24	System voltage detection abnormality, please check the main control panel.
25	Undefined.
26	Flash save error, please check the main control panel.
27	Master control password is wrong, please replace the main control panel.
28	Short circuit in motor driven upper MOS bridge, please check the main control panel.
29	Failure in motor driven bottom MOS bridge, please check the main control panel.
30	Undefined.
31	Program skip error, please check the main control panel.

32	Undefined.
33	Undefined.
34	Undefined.
35	Vehicle is in default serial number, please replace the main control panel.
36	2+4 connector failure or external battery charging wire failure.
37	Charging base failure or built-in battery charging wire failure.
38	Undefined.
39	Built-in battery temperature sensor abnormality, please replace the battery.
40	Controller temperature sensor abnormality, please check the main control panel.
41	External battery temperature sensor abnormality, please replace the battery.
42	External battery abnormality in communication, please check external battery, main control panel and related wiring.
43	External battery password is wrong, please replace the battery.
44	External battery is in default serial number, please replace the battery.