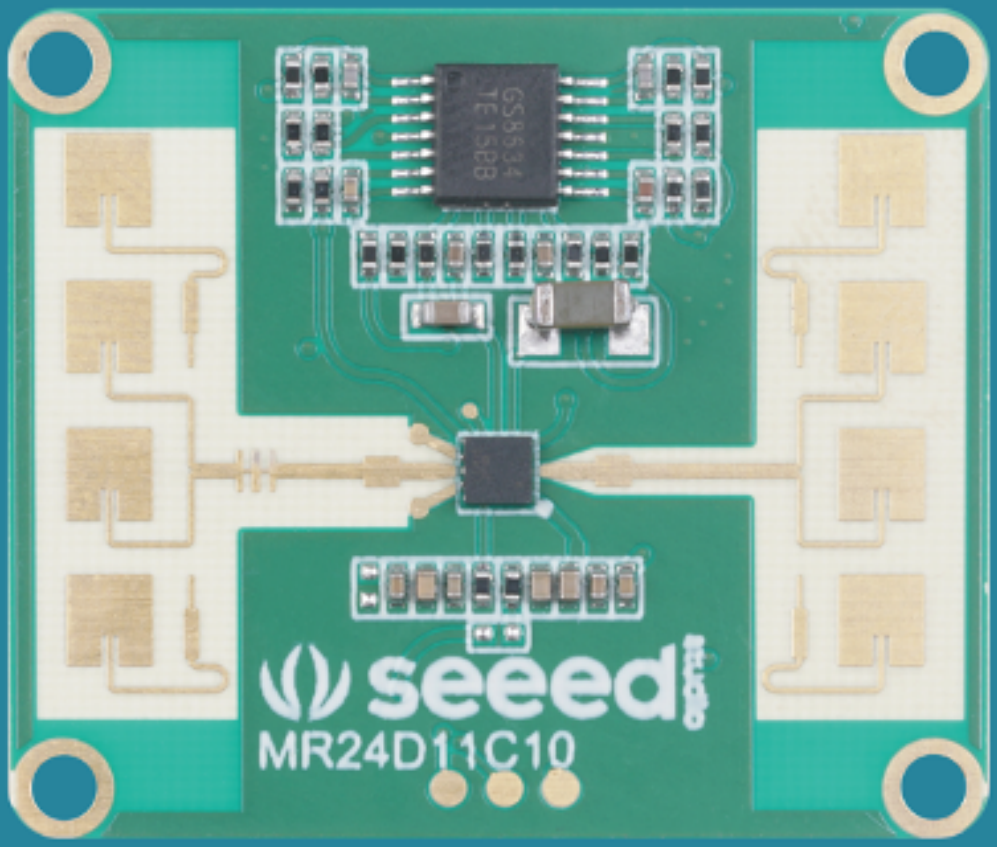
MR24HPB1

Human Presence Radar User Manual



CONTENTS

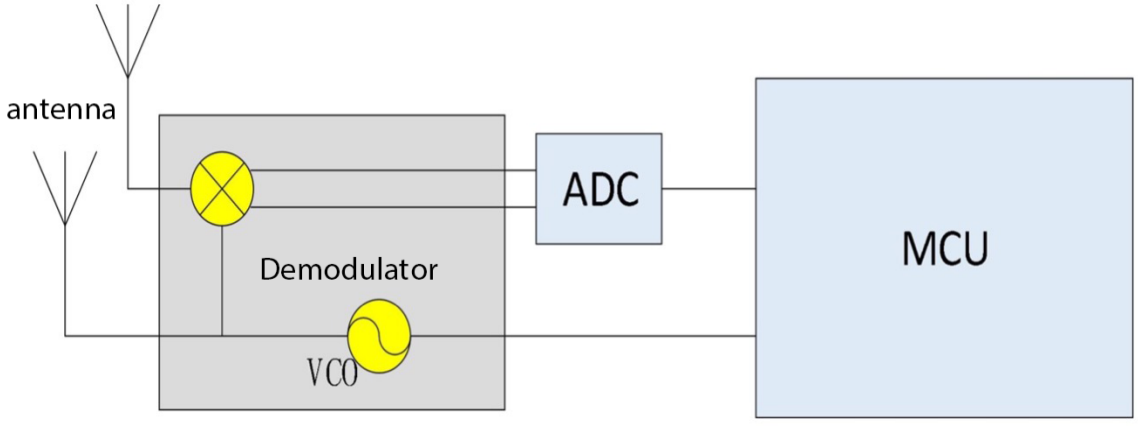
Overview

This document focuses on the use of the radar, the issues that need to be addressed at each stage to minimise design costs and increase product stability and improve the efficiency of project completion.

From hardware circuit reference design, radar antenna and housing layout requirements, how to differentiate between interference and multi-functional standard UART protocol outputs.

The radar is a self-contained space sensing sensor, consisting of RF antenna, radar chip and high speed main frequency MCU together with a module that relies on a stable and flexible superior algorithm architecture core to solve the user's various scenario detection needs, which can be equipped with a host computer or host computer to flexibly output detection status and data, meeting several groups of GPIOs for custom development.

1. Principle of operation



The radar transmits a millimetre wave signal in the 24G band, the measured target reflects the electromagnetic wave signal and demodulates the transmitted signal, which is then amplified, filtered and processed by ADC to obtain the echo demodulated signal data. The amplitude, frequency and phase of the echo signal are decoded in the MCU unit, which ultimately

enables the measurement of target parameters (breathing, movement, micro-motion, etc.) and scene evaluation.

2. Hardware Design Considerations

The radar needs to have a rated supply voltage of 4.9 – 6V and a rated current requirement of 200mA or more input under normal operating conditions. The power supply must be designed for a supply ripple of ≤ 100mv.

2.1. The power supply can be designed with the following circuit in mind

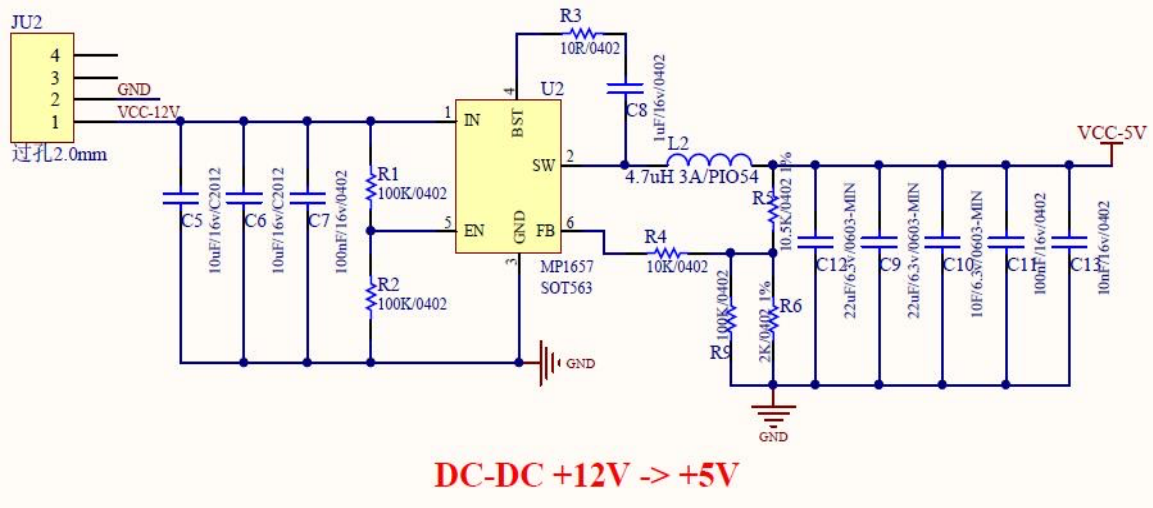


Fig. 1

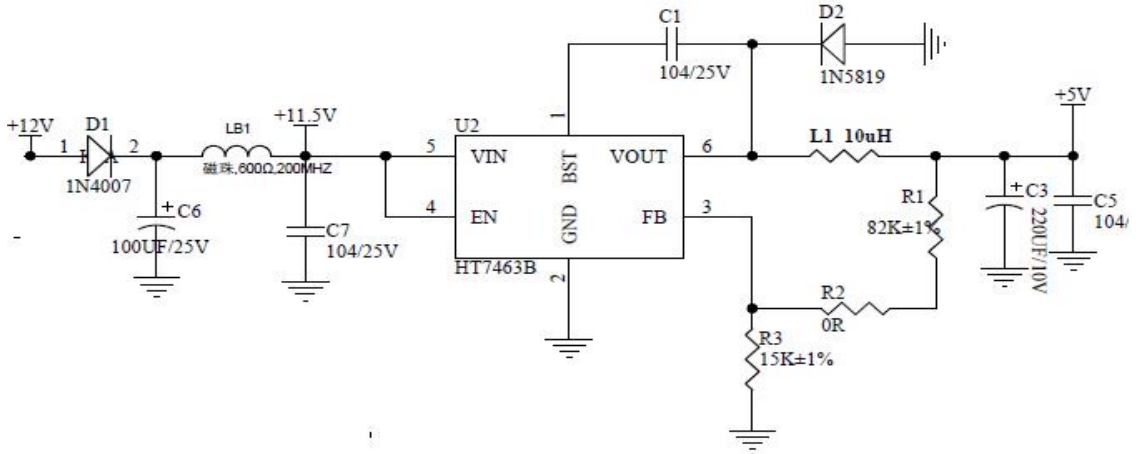


Fig. 2

2.2. using the wiring diagram

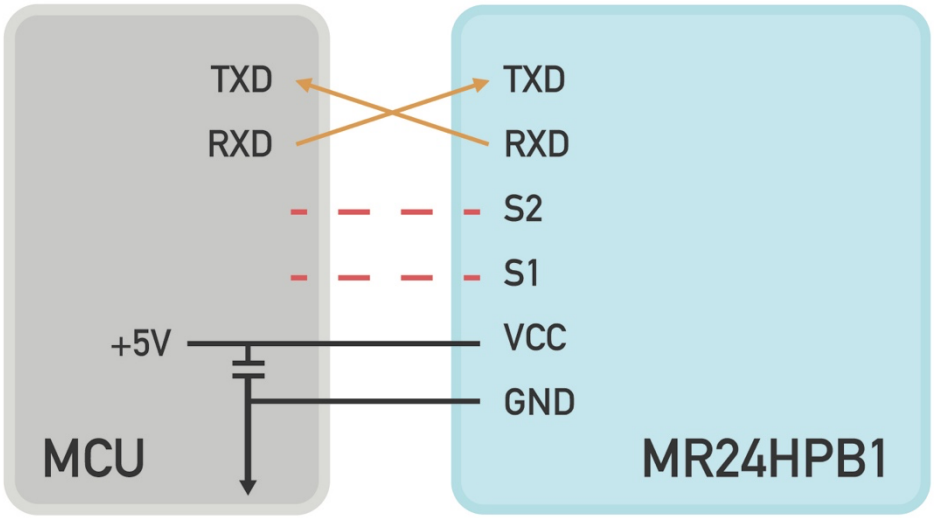


Fig. 3 Schematic diagram of the radar module and peripheral connections

3. Antenna and housing layout requirements

PCBA: the radar needs to be kept ≥ 1mm higher than the other components.

Housing construction: need to maintain a distance of 2 – 5mm between the radar antenna face and the housing face.

Housing detection surface: non-metallic housing, needs to be flat and straight to avoid curved surfaces which can affect the performance of the entire swept surface area.

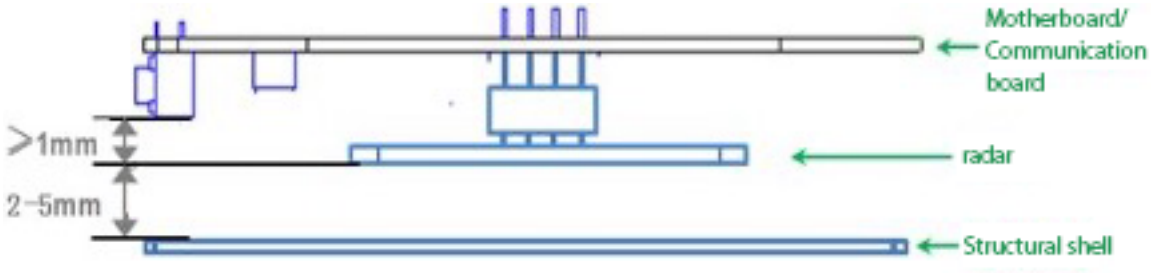


Fig. 4

4. Static Protection

Radar products have electrostatic sensitive circuitry and are susceptible to electrostatic hazards, therefore they need to be adequately protected from static electricity during transport, storage, work and handling.

When handling the radar sensor, please wear anti-static gloves if possible.

5. Functional disturbances

5.1 Unoccupied state, abnormal output occupied

In normal conditions, the radar will accurately determine the presence of a sitting or sleeping body and output information on falls, breathing, vital signs etc.

A. Radar scanning area is large, doorway, boarded wall next door movement is detected.

Adjustment method: reduce radar sensitivity, radar provides scene setting.

B. Radar underneath is facing a running air conditioner or fan.

Adjustment method: adjust the radar position so that it is not directly in front of the air conditioner or fan.

C. Shifting objects caused by air conditioning winds.

Adjustment method: cotton, non-metallic objects will not cause false alarms, metal objects need to be fixed.

D. The radar is not fixed, vibration causes false alarms.

Avoid supporting shaking and vibration.

E. Occasional moving objects such as pets, birds, etc.

As the radar measures micro-movements, the sensitivity is very high and this interference cannot be excluded.

F. Power supply interference, resulting in occasional false alarms.

Try to keep the power supply current stable and reduce ripple.

5.2 Manned status, abnormal output unoccupied

Radar determines the presence of a human body by sending and receiving electromagnetic waves. The closer you are to the radar, the more accurate it is.

A. Human body out of radar range

Radar scanning range with adjustment of mounting angle. Radar measurement range, in different environments with different electromagnetic wave reflection areas, the scanning area will vary slightly.

B. False output due to metal occlusion

Excessively thick desks and chairs, metal seats. It will block the electromagnetic wave penetration and cause misinterpretation.

C. Differences in scanning angles

The radar does not scan the torso area. This can lead to false positives.

D. Radar sensitivity is too low

The radar offers parameter adjustment to increase sensitivity for improvement.

6. Functions in detail

6.1. Function point descriptions

Functions	Status change time/function explanation
DP1: occupied/unoccupied	No one to occupied, report within 0.5s Manned to unoccupied, no status output in 1–2 minutes or so
DP2: Some people are stationary / Some people are active	Static dynamic switching, reporting within 0.5 seconds
DP3: Someone close to the device / someone moving away from the device / someone moving without direction	Status output once every 2 seconds
DP4: Body movement amplitude parameter 0 – 100	Data output once every 5 seconds Reference (description of output of body motion amplitude parameters)
DP5: Sensitivity setting 1 – 10 steps	Default scene mode, adapted to 10 positions of adjustment
DP7: Scene modes (bed, bathroom, hotel, bedroom, office, default mode)	Adapted to different scenarios according to the size of the area
DP8: No false alarm confirmation prompt	

6.2. Description of the output of the body motion amplitude parameter

Body movement amplitude parameters		
0%	None	Environmental unmanned
1%	Stationary (sleep)	Only breathing without body movement
2% – 30%	Micro-Movements	Only minor head or limb movements Movement
31% – 60%	Walking/fast body movements	Slower body movements
61% – 100%	Running/close range big moves	Rapid body movement

7. Description of the agreement

This protocol is used to communicate between a 24G millimetre wave sleep detection radar and a host computer.

This protocol outlines the radar workflow, provides a brief introduction to the interface protocol component architecture and gives the control commands and data required for the operation of the relevant radar, with the serial communication defined as follows.

- Interface level: TTL
- Baud rate: 9600bps
- Stop bits: 1
- Data bits: 8
- Parity: None

8. Communication commands and parameter definitions

8.1 Definition and description of the frame structure

A. Frame structure definition

Starting Code	Length of data		Function codes	Address code 1	Address code 2	Data	Check Code	
0X55	Lenth_L	Lenth_H	Command	Address_1	Address_2	Data	Crc16_L	Crc16_H
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	n Byte	1 Byte	1 Byte

B. Description of the frame structure

a. Start code: 1 Byte, fixed to 0X55.

b. Data length: 2 Byte, low byte before, high byte after.

Length = Data Length + Function Code + Address Code 1 + Address Code 2 + Data + Checksum.

c. Function code: 1 Byte

Read command: 0X01

Write command: 0X02

Passive report command: 0X03

Active report command: 0X04

d. Address code: Address code 1

indicates the function classification, address code 2 indicates the specific function.

See the description of address assignment and data information.

e. Data: n Byte

f. Checksum: 2 Byte, low byte before, high byte after.

CRC16 checksum is used, see Appendix 1 for reference code.

8.2. Description of address assignment and data information

24G Bio-aware radar interface content					
	Function Code	**Address code 1**	**Address code 2**	**Data**	**Notes**
1	Read command 0x01	Marking search 0x01	Device ID 0x01		
2			Software version 0x02		
3			Hardware version 0x03		
4			Protocol version 0x04		
		Radar Information Search 0x03	Environmental status 0x05		
11			Signs parameters 0x06		
12		System parameter search 0x04	Threshold gear 0x0C		
			Scene setting 0x10		
16			Threshold gear 0x0C	Enumeration range1~10	Corresponding to 1 2 3 4 5 6 7 8 9 10 gears (default 7) The higher the gear, the more

24G Bio-aware radar interface content					
	Function Code	**Address code 1**	**Address code 2**	**Data**	**Notes**
	copy order 0x02	System parameters 0x04			sensitive it is
			Scene setting 0x10	Default mode 0x00	
				Area detection (top loading) 0x01	
				Bathroom (top mounted) 0x02	
				Bedroom (top loading) 0x03	
				Living room (top mounted) 0x04	
				Office (top loading) 0x05	
				Hotel (top loading) 0x06	
		Other functions 0x05	Reboot 0x04		
17		Reporting module identification 0x01	Device ID 0x01	12 Byte data	
18			Software version 0x02	10 Byte data	
19			Hardware version 0x03	8 Byte data	
20			Protocol version 0x04	8 Byte data	
27		Report radar	Environment	Unoccupied 00 FF FF	

24G Bio-aware radar interface content					
	Function Code	**Address code 1**	**Address code 2**	**Data**	**Notes**
28	Passive reporting of orders 0x03	information 0x03	status 0x05	Someone is stationary 01 00 FF	
29				Some people exercise 01 01 01	
30			Signs parameters 0x06	4 Byte Float data (see appendix 2)	
		Reporting system information 0x04	Threshold gear 0x0C	Current gear value (0x01~0x0a)	
			Scene setting 0x10	Default mode 0x00	
				Area detection (top loading) 0x01	
				Bathroom (top mounted) 0x02	
				Bedroom (top loading) 0x03	
				Living room (top mounted) 0x04	
				Office (top loading) 0x05	
				Hotel (top loading) 0x06	
31			Environment status 0x05	Unoccupied 00 FF FF	
32				Someone is stationary 01 00 FF	
33				Some people exercise	

24G Bio-aware radar interface content						
	Function Code	**Address code 1**	**Address code 2**	**Data**		**Notes**
	Proactive reporting of commands 0x04	Report radar information 0x03		01 01 01		
34			Motor signs parameters 0x06	4 Byte Float data		
			Approaching away state 0x07	Fixed character 0x01 0x01	None 0x01	
					Close to 0x02	
					Stay away 0x03	
		Report other information 0x05	Heartbeat Pack 0x01	Unoccupied 00 FF FF		
				Someone is stationary 01 00 FF		
				Some people exercise 01 01 01		
			Abnormal reset 0x02	0x0F		

Description.

1) The read/write command is for the upper computer to send commands to the radar.

2) The report command is for the radar to send information to the upper computer.

3) Fall sensitivity is 1~10, default is 4, the higher the level, the more sensitive it is.

Appendix 1: CRC check digit reference parsing codes

```
const unsigned char cuc_CRCHi[256]=
{
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40
 };

const unsigned char  cuc_CRCLo[256]=
{
        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
 };


static unsigned short int us_CalculateCrc16(unsigned char *lpuc_Frame, unsigned short i
nt lus_Len)
 {
     unsigned char luc_CRCHi = 0xFF;
     unsigned char luc_CRCLo = 0xFF;
     int li_Index=0;
```

```
59.
60.      while(lus_Len--)
61.      {
62.           li_Index = luc_CRCLo ^ *( lpuc_Frame++);
63.           luc_CRCLo = (unsigned char)( luc_CRCHi ^ cuc_CRCHi[li_Index]);
64.           luc_CRCHi = cuc_CRCLo[li_Index];
65.      }
66.      return (unsigned short int )(luc_CRCLo << 8 | luc_CRCHi);
67. }
```

Appendix 2: Analysis codes for motor sign parameters

```
typedef union
{
    unsigned char Byte[4];
    float Float;
}Float_Byte;

void main()
{
    Float_Byte fb;
    fb.Byte[0] = 0x9A;
    fb.Byte[1] = 0xFB;
    fb.Byte[2] = 0xE7;
    fb.Byte[3] = 0x3F;
    printf("%f\ r\ n",fb.Float);
}
```

Historical version update notes

Revision	Release Data	Summary
V1.0_0212	2020/02/12	First draft
V1.1_0319	2021/03/19	Readjustment
V1.3_0628	2021/6/28	Add Human sensitivity explained and fall sensitivity explained
V1.4_0906	2021/9/06	Human sensitivity revised from 0–9 to 1–10