

**24G 毫米波雷达  
R24AVD1 人体存在雷达  
用户手册 V1.8**

## 目录

1. 产品概述.....	2
2. 工作原理.....	2
3. 硬件设计注意事项.....	2
3.1 电源可参考以下电路设计.....	3
3.2 使用接线图.....	3
4. 天线与外壳的布局的要求.....	4
5. 静电防护.....	4
6. 功能干扰项.....	4
6.1 无人状态，异常输出有人.....	4
6.2 有人状态，异常输出无人.....	5
7. 功能详解.....	5
7.1 标准功能点说明.....	5
7.2 体动幅度参数输出说明.....	6
8. 协议说明.....	6
9. 通讯命令及参数定义.....	6
9.1 帧结构定义及说明.....	6
9.2 地址分配及数据信息说明.....	7
附录 1: CRC 校验码参考解析代码.....	11
10. 历史版本更新说明.....	13

**说明:**

点击链接或扫描二维码确保您使用的是最新版本的文档:

[http://www.micradar.cn/go\\_file.php?id=171](http://www.micradar.cn/go_file.php?id=171)

**1. 产品概述**

本文档主要阐述该雷达使用事项，各个阶段需要注意的问题点，尽可能降低设计成本和增加产品的稳定性，提升项目的完成效率。

从硬件电路参考设计、雷达天线与外壳的布局要求、如何区分干扰和多功能的标准 UART 协议输出。

本雷达是一个自成体系的隔空感知传感器，由射频天线、雷达芯片和高速主频 MCU 一起组合而成的模组，依赖稳定灵活优越的算法架构核心，解决用户的各种场景探测需求，可搭载上位机或者主机灵活输出探测状态和数据，满足几组 GPIO 可供用户定制开发。

雷达天线发射电磁波信号，并同步接收目标反射后的回波信号，雷达处理器通过解析回波信号的波形参量，反馈目标的距离、方向、速度等信息。可以探测运动物体的状态和轨迹

**2. 工作原理**

雷达发射 24G 频段毫米波信号，被测目标反射电磁波信号，并于发射信号进行解调处理，进而通放大、滤波、ADC 等处理，得到回波解调信号数据。在 MCU 单元对回波信号的幅度、频率、相位进行信息解算，最终实现目标参数（呼吸、运动、微动等）测量及场景评估。

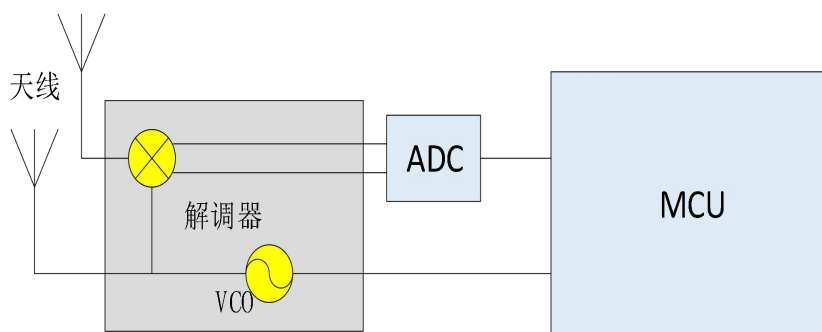


图 1: 工作原理图

**3. 硬件设计注意事项**

该雷达的额定供电电压需满足 4.9 - 6V，在正常工作情况下，额定电流要求 200mA

以上的输入。电源设计，电源纹波需  $\leq 100\text{mv}$ 。

### 3.1 电源可参考以下电路设计

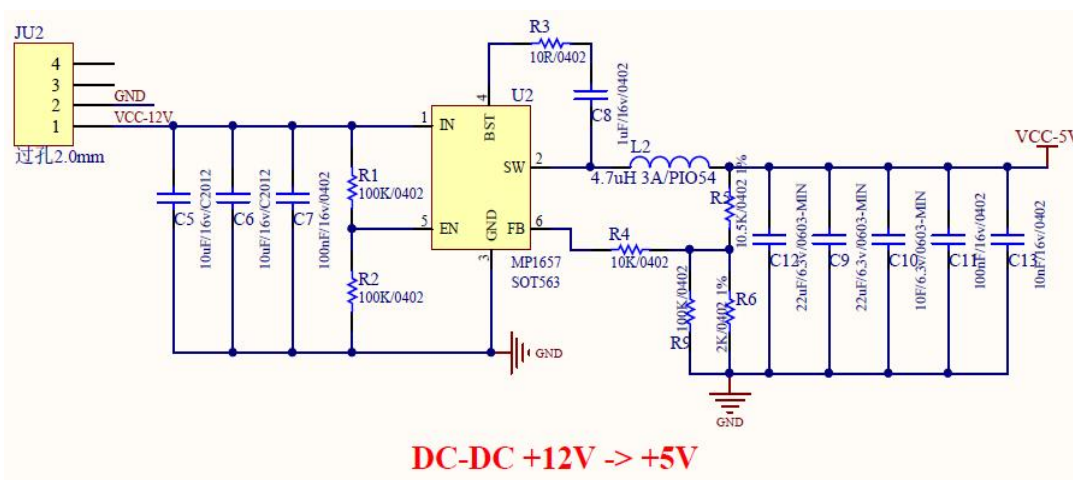


图 2

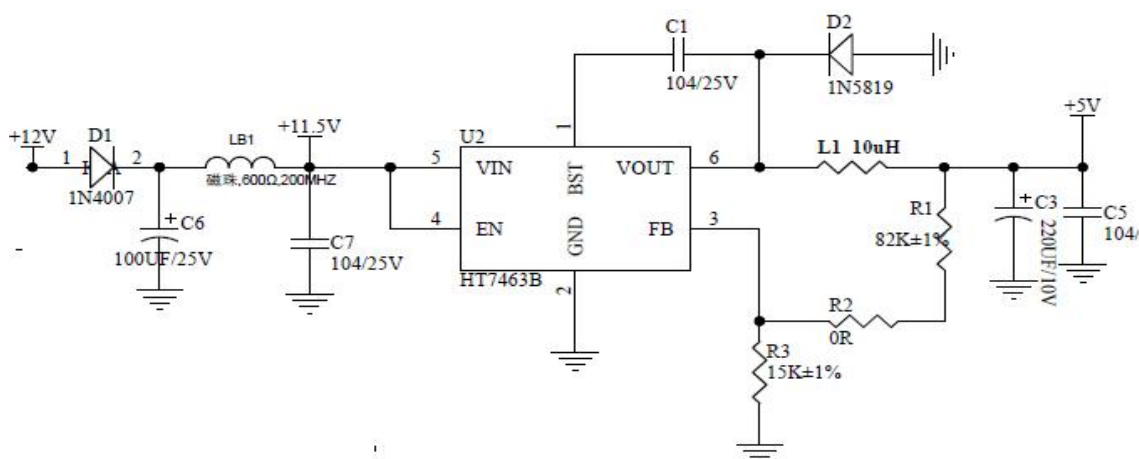


图 3

### 3.2 使用接线图

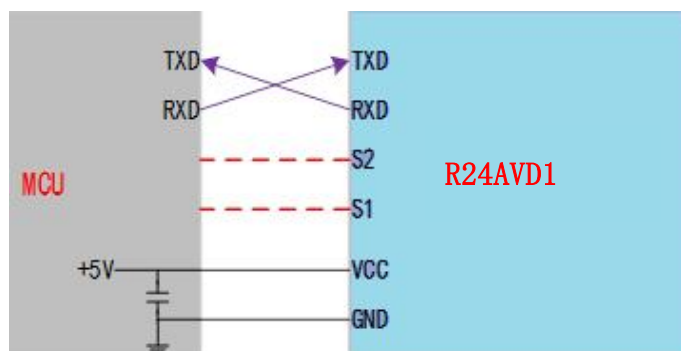


图 4 雷达模块与外设连线示意图

## 4. 天线与外壳的布局的要求

PCBA: 需要保持雷达的贴件高度比其他器件  $\geq 1\text{mm}$

外壳结构: 需要保持雷达天线面和外壳面有 2 - 5mm 距离

外壳探测面: 非金属外壳, 需要平直避免弯曲面, 影响整个扫面面积的性能

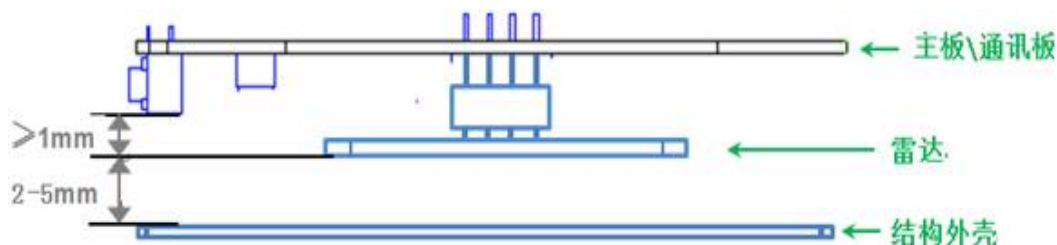


图 5: 天线与外壳的布局图

## 5. 静电防护

雷达产品内部具有静电敏感电路, 容易遭受静电危害, 因此需要在运输、存储、工作和拿取的过程中充分做好静电防护工作, 不要用手触摸抓取雷达模块天线表面和连接器管脚, 只能触摸其边角部分。对雷达传感器进行操作时, 请尽量带上防静电手套。

## 6. 功能干扰项

### 6.1 无人状态, 异常输出有人

正常状态, 雷达会精准判断人体静坐状态和睡眠的存在, 并输出相对应生命体征等信息。

A. 雷达扫描面积大, 门口, 木板墙的隔壁运动被探测到。

调整方法: 降低雷达灵敏度, 雷达提供场景设置。

B. 雷达下方正对运行中空调, 风扇。

调整方法: 调整雷达位置, 不要直接面对空调, 风扇。

C. 空调风引起的物体晃动。

调整方法: 棉质, 非金属物品不会引起误报, 金属物品需要固定。

D. 雷达没有固定, 振动导致误报。

避免支撑晃动, 震动。

E. 宠物, 飞鸟等偶尔运动物体。

由于雷达测量微动, 灵敏度很高, 无法排除此项干扰。

F. 电源干扰，导致偶尔有误判。

尽量保持供电电流稳定，减少纹波。

## 6.2 有人状态，异常输出无人

雷达通过电磁波收发，判断人体存在。距离雷达越近，精度越高。

A.人体在雷达范围之外：

雷达扫描范围，调整安装角度。

雷达测量范围，在不同环境，电磁波反射面积不同，扫描面积会有微小差异。

B.金属遮挡造成错误输出：

过厚的办公桌椅，金属座椅。会阻挡电磁波穿透，造成误判。

C.扫描角度差异：

雷达没有扫描到躯干部位,造成误判。

D.雷达灵敏度过低：

雷达提供参数调节，增加灵敏度改善。

## 7. 功能详解

### 7.1 标准功能点说明

功能点	状态变化时间/功能解释
DP1: 有人/无人	无人到有人，0.5s 内上报 有人到无人，50s 左右输出无人状态
DP2: 有人静止/有人活跃	静态动态切换，0.5 秒以内上报
DP3: 有人靠近设备/ 有人远离设备/有人无方向移动	0.5s 内上报；持续 3s 靠近/远离会上报 持续靠近/持续远离状态
DP4: 体动幅度参数 0 - 100	1 秒输出一次数据 参考（体动幅度参数输出说明）
DP5: 灵敏度设置 1 - 3 档	默认为灵敏度 3，可支持适配 3 个档位调节； 灵敏度越高，静止探测面积越大

DP6: 场景模式（默认，区域探测，卫生间，酒店，卧室，办公室）	按照面积大小，适配不同的场景
----------------------------------	----------------

## 7.2 体动幅度参数输出说明

体动幅度参数		
0%	无人	环境无人
1%	静止（睡眠）	只有呼吸而没有肢体运动
2%-30%	微动作	只有轻微头部或者肢体小运动
31%-60%	走动/快速肢体运动	比较慢速的身体移动
61%-100%	跑动/近距离大动作	快速身体移动

## 8. 协议说明

本协议应用于 24G 毫米波人体存在探测雷达与上位机之间的通信。

本协议概要介绍了雷达工作流程，对接口协议组成架构进行了简单介绍，并给出了相关雷达工作所需要控

制命令及数据,串口通信定义如下:

- 接口电平: TTL
- 波特率: 9600bps
- 停止位: 1
- 数据位: 8
- 奇偶校验: 无

## 9. 通讯命令及参数定义

### 9.1 帧结构定义及说明

#### A. 帧结构定义

起始码	数据长度		功能码	地址码1	地址码2	数据	校验码	
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.帧结构说明

- 起始码：1Byte，固定为 0X55。
- 数据长度：2 Byte，低字节在前，高字节在后（长度=数据长度+功能码+地址码 1+地址码 2+数据+校验码）
- 功能码：1Byte
- 读命令：0X01
- 写命令：0X02
- 被动上报命令：0X03
- 主动上报命令：0X04
- 地址码：地址码 1 表示功能分类，地址码 2 表示具体功能（见地址分配及数据信息说明）
- 数据：n Byte
- 校验码：2 Byte，低字节在前，高字节在后（采用 CRC16 校验，参考代码见附录 1。）

## 9.2 地址分配及数据信息说明

24G 生物感知雷达接口内容					
	功能码	地址码 1	地址码 2	数据	备注
雷 达 接 口 内 容	读命令 0X01	标识查询 0X01	设备 ID 0X01		
			软件版本 0X02		
			硬件版本 0X03		
			协议版本 0X04		
		雷达信息查询 0X03	环境状态 0X05		
			体征参数 0X06		
		系统参数 0X04	灵敏度设置 0X0C		
			场景设置 0X10		
			强制进入无人档位 0X12		



写命令 0X02	系统参数 0X04	灵敏度设置 0X0C	枚举范围 1~3	分别对应 1 2 3 档(默认 3 档) 档位越大, 越灵敏
		场景设置 0X10	默认模式 0X00	
			区域探测 (顶装) 0X01	
			卫生间 (顶装) 0X02	
			卧室 (顶装) 0X03	
			客厅 (顶装) 0X04	
			办公室 (顶装) 0X05	
			酒店 (顶装) 0X06	
		强制进入无人档位 0X12	不使用强制进入无人功能 0X00	
			10s 0X01	
	30s 0X02			
	1min 0X03			
	2min 0X04			
	5min 0X05			
10min 0X06				
30min 0X07				
其他功能 0X05	重启 0X04			
	开始 OTA 升级 0X08	4Byte 整形数据 (固件包大小) + 15Byte 固件信息		

			升级包传输 0X09	包偏移 (4Byte) + 数据包 (1024Byte)	前四个字节固定为包偏移, 后面为数据包内容
			升级结束信息 0X0A	固定字符 0X0F	
被动上报 命令 0X03	上报模块标识 0X01		设备 ID 0X01	12 Byte 数据	
			软件版本 0X02	10 Byte 数据	
			硬件版本 0X03	8 Byte 数据	
			协议版本 0X04	8 Byte 数据	
	上报雷达信息 0X03	环境状态 0X05		无人状态 00 FF FF	
				有人静止 01 00 FF	
				有人运动 01 01 01	
		体征参数 0X06	4 Byte Float 数据 (见附录 2)		
	上报系统参数 0X04		灵敏度设置 0X0C	当前档位值 (0X01~0X03)	
		场景设置 0X10		默认模式 0X00	
				区域探测 (顶装) 0X01	
				卫生间 (顶装) 0X02	
				卧室 (顶装) 0X03	
			客厅 (顶装) 0X04		
			办公室 (顶装) 0X05		
			酒店 (顶装) 0X06		
	强制进入无人档位 0X12	不使用强制进入无人功能 0X00			

				10s 0X01	
				30s 0X02	
				1min 0X03	
				2min 0X04	
				5min 0X05	
				10min 0X06	
				30min 0X07	
				60min 0X08	
		上报其他信息	反馈 OTA 升级开始 0X08	失败 0X00	
			反馈 OTA 传输 0X09	成功 0X01	
				固定字符 0X0F	
		上报模块标识 0X01	软件版本 0X02	15 byte 数据	
	主动上报 命令 0X04	上报雷达信息 0X03	环境状态 0X05	无人状态 00 FF FF	
				有人静止 01 00 FF	
				有人运动 01 01 01	
			运动体征参数 0X06	4 Byte Float 数据	
		接近远离状态 0X07	固定字符 0X01 0X01	无 0X01	
				接近 0X02	
	远离 0X03				
	持续接近 0X04				
	持续远离				

				0X05	
	上报其他信息 0X05	心跳包 0X01	无人状态 00 FF FF		
			有人静止 01 00 FF		
			有人运动 01 01 01		
		异常复位 0X02	0X0F	雷达重启或者重上电时会先上报异常复位指令再开始进行初始化过程最后上报初始化成功指令 代表雷达初始化成功开始进行正常运行	
	初始化成功 0X0A	0X0F			

说明：1) 读写命令为上位机向雷达发送指令。

2) 上报命令为雷达向上位机发送信息。

3) 人体灵敏度为 1-3 档，默认 3 档，档位越大越灵敏。

## 附录 1: CRC 校验码参考解析代码

1. `const unsigned char cuc_CRCHi[256]=`

```

2.     {
3.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
4.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
5.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
6.         0x00,  0xC1,  0x81,  0x40,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,
7.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
8.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
9.         0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
10.        0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
11.        0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
12.        0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
13.        0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
14.        0x00,  0xC1,  0x81,  0x40,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,
15.        0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
16.        0x00,  0xC1,  0x81,  0x40,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,
17.        0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,
18.        0x00,  0xC1,  0x81,  0x40,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,

```

```

19.      0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
20.      0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x00,  0xC1,  0x81,  0x40,
21.      0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
22.      0x00,  0xC1,  0x81,  0x40,  0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,
23.      0x00,  0xC1,  0x81,  0x40,  0x01,  0xC0,  0x80,  0x41,  0x01,  0xC0,  0x80,  0x41,
24.      0x00,  0xC1,  0x81,  0x40
1.      const unsigned char      cuc CRCLo[256]=
2.      {
3.          0x00, 0xC0, 0xC1,      0x01, 0xC3, 0x03,      0x02,  0xC2,  0xC6,  0x06,  0x07,  0xC7,
4.          0x05, 0xC5, 0xC4,      0x04, 0xCC, 0x0C,      0x0D,  0xCD,  0x0F,  0xCF,  0xCE,  0x0E,
5.          0x0A, 0xCA, 0xCB,      0x0B, 0xC9, 0x09,      0x08,  0xC8,  0xD8,  0x18,  0x19,  0xD9,
6.          0x1B, 0xDB, 0xDA,      0x1A, 0x1E, 0xDE,      0xDF,  0x1F,  0xDD,  0x1D,  0x1C,  0xDC,
7.          0x14, 0xD4, 0xD5,      0x15, 0xD7, 0x17,      0x16,  0xD6,  0xD2,  0x12,  0x13,  0xD3,
8.          0x11, 0xD1, 0xD0,      0x10, 0xF0, 0x30,      0x31,  0xF1,  0x33,  0xF3,  0xF2,  0x32,
9.          0x36, 0xF6, 0xF7,      0x37, 0xF5, 0x35,      0x34,  0xF4,  0x3C,  0xFC,  0xFD,  0x3D,
10.         0xFF, 0x3F, 0x3E,      0xFE, 0xFA, 0x3A,      0x3B,  0xFB,  0x39,  0xF9,  0xF8,  0x38,
11.         0x28, 0xE8, 0xE9,      0x29, 0xEB, 0x2B,      0x2A,  0xEA,  0xEE,  0x2E,  0x2F,  0xEF,
12.         0x2D, 0xED, 0xEC,      0x2C, 0xE4, 0x24,      0x25,  0xE5,  0x27,  0xE7,  0xE6,  0x26,
13.         0x22, 0xF2, 0xF3,      0x23, 0xF1, 0x21,      0x20,  0xF0,  0xA0,  0x60,  0x61,  0xA1,
14.         0x63, 0xA3, 0xA2,      0x62, 0x66, 0xA6,      0xA7,  0x67,  0xA5,  0x65,  0x64,  0xA4,
15.         0x6C, 0xAC, 0xAD,      0x6D, 0xAF, 0x6F,      0x6E,  0xAE,  0xAA,  0x6A,  0x6B,  0xAB,
16.         0x69, 0xA9, 0xA8,      0x68, 0x78, 0xB8,      0xB9,  0x79,  0xBB,  0x7B,  0x7A,  0xBA,
17.         0xBE, 0x7E, 0x7F,      0xBF, 0x7D, 0xBD,      0xBC,  0x7C,  0xB4,  0x74,  0x75,  0xB5,
18.         0x77, 0xB7, 0xB6,      0x76, 0x72, 0xB2,      0xB3,  0x73,  0xB1,  0x71,  0x70,  0xB0,
19.         0x50, 0x90, 0x91,      0x51, 0x93, 0x53,      0x52,  0x92,  0x96,  0x56,  0x57,  0x97,
20.         0x55, 0x95, 0x94,      0x54, 0x9C, 0x5C,      0x5D,  0x9D,  0x5F,  0x9F,  0x9E,  0x5E,
21.         0x5A, 0x9A, 0x9B,      0x5B, 0x99, 0x59,      0x58,  0x98,  0x88,  0x48,  0x49,  0x89,
22.         0x4B, 0x8B, 0x8A,      0x4A, 0x4E, 0x8E,      0x8F,  0x4F,  0x8D,  0x4D,  0x4C,  0x8C,
23.         0x44, 0x84, 0x85,      0x45, 0x87, 0x47,      0x46,  0x86,  0x82,  0x42,  0x43,  0x83,
24.         0x41, 0x81, 0x80,      0x40
25. };

```

```

1.  static unsigned short int  us_CalculateCrc16(unsigned char *lpuc_Frame, unsi
      gned short int  lus_Len)
2.  {
3.      unsigned char  luc_CRChI = 0xFF;
4.      unsigned char  luc_CRCLo = 0xFF;
5.      int li Index=0;
6.
7.      while(lus_Len--)
8.      {
9.          li Index = luc_CRCLo ^ *( lpuc_Frame++);
10.         luc_CRCLo = (t_BYTE)( luc_CRChI ^ cuc_CRChI[li_Index]);
11.         luc_CRChI = cuc_CRCLo[li_Index];
12.     }

```

```

13. return (unsigned short int)(luc CRCLo << 8 | luc CRChi);
14. }

```

示例：灵敏度查询读命令：

55 07 00 01 04 0C sum (2byte)  $\xrightarrow{\text{校验码解析}}$  55 07 00 01 04 0C EA DB

## 附录 2：运动体征参数解析代码

```

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);
}

```

## 10.历史版本更新说明

Revision	Release Date	Summary
V1.0_0212	2020/02/12	初稿
V1.1_0319	2021/03/19	重新调整
V1.2_0628	2021/6/28	加上 人体灵敏度档位说明
V1.3_0906	2021/9/06	人体灵敏度从 0-9 修改为 1-10
V1.4_0210	2022/2/10	增加初始化成功指令协议

V1.5_0221	2022/2/21	增加强制进无人档位协议
V1.6_0606	2022/6/6	调整文档封面及相关数据细节
V1.7_0223	2023/2/23	增加持续接近/持续远离协议 优化相关数据细节
V1.8_0207	2025/2/7	修改“阈值档位”为“灵敏度设置”