

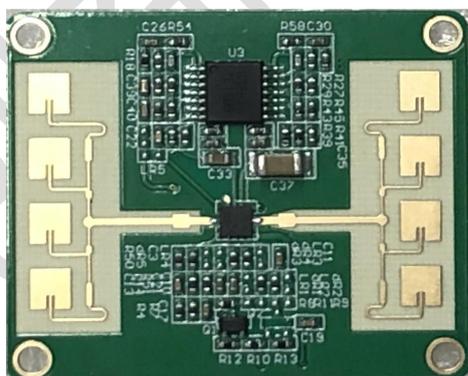


睡眠监测雷达 R24SM1B

数据手册 (V1.0)

产品特点

- 24GHz 雷达传感器
- 人体生命体征输出
- 基于增强多普勒雷达技术，实现二维区域人员感知功能；
- 实现呼吸监测及运动感知功能；
- 呼吸次数探测最大距离： ≥ 2 米
- 天线波束宽度： $90^\circ - 100^\circ / 30^\circ$
- 具备场景识别能力，识别有人/无人及人员活动状态
- 不受温度、湿度、噪声、气流、尘埃、光照等影响，适合恶劣环境；
- 输出功率小，长时间照射对人体无伤害；
- 无人到有人探测时间：0.5 秒以内
- 有人到无人探测时间：大于 1 分钟



产品应用

- ◇ 睡眠监控
- ◇ 呼吸功能监测
- ◇ 老人看护
- ◇ 婴幼儿看护
- ◇ 家庭康养



产品封装

- ✚ 体积： $\leq 35\text{mm} \times 30\text{mm} \times 5\text{mm}$
- ✚ 接口：Pitch 2.0mm 双排插针

立创商城专用



目 录

1. 概述.....	4
2. 电气参数.....	4
3. 模块尺寸及引脚说明.....	5
3.1. 模块尺寸.....	5
3.2. 引脚说明.....	5
4. 模块工作模式.....	6
4.1. 雷达模块探测范围.....	6
4.2. 主要功能.....	7
4.3. 安装方式.....	7
4.4. 雷达模块工作模式.....	8
4.5. 安装方式.....	8
4.5.1. 倾斜安装.....	8
4.5.2. 置顶安装.....	9
5. 典型应用模式.....	10
5.1. 卧室安装及应用.....	10
6. 模块接口协议.....	11
6.1. 接口介绍.....	11
帧结构定义及说明.....	11
A、帧结构定义.....	11
B、帧结构说明.....	11
7. 注意事项.....	13
7.1. 启动时间.....	13
7.2. 有效探测距离.....	13
7.3. 雷达生物探测性能.....	13
7.4. 电源.....	13
8. 常见问题.....	13
9. 免责声明.....	13
10. 版权说明.....	14
1. 联系方式.....	14
云帆瑞达科技（深圳）有限公司.....	14
附录1.....	14



1. 概述

R24SM1B 雷达模块是基于毫米波多普勒雷达体制，实现的人体呼吸感知及人体运动感知的雷达探测模块。本模块基于增强多普勒雷达信号处理体制，通过对人员运动的多普勒参数及人员的呼吸参数同步感知技术，实现特定场所内人员睡眠状态的无线感知。

本雷达模块具有如下工作特点：

- ◇ 本模块将检测对象限制于人员（运动或静止），剔除环境内其它无生命物体的干扰；
- ◇ 本模块实现运动人员及静止人员的同步感知功能；
- ◇ 本模块对非生命类物体干扰有效剔除，也可实现非生命类运动物体检测；
- ◇ 产品支持二次开发，适应多种场景应用；
- ◇ 本输出功率小，对人体不构成危害；
- ◇ 本模块不受温度、光照、粉尘等因素影响，灵敏度高，应用领域广泛。

2. 电气参数

参数	最小值	典型值	最大值	单位
工作性能				
呼吸感应最大距离		1.5	2	米
运动探测最大距离	4	5	6	米
雷达探测角度（水平）		90-110		度
雷达探测角度（俯仰）		30-40		度
工作参数				
工作电压（VCC）	4.5	5.0	6	V
工作电流（I _{CC} ）	90	93	100	mA
工作温度（T _{OP} ）	-20		+60	°C
存储温度（T _{ST} ）	-40		+80	°C
发射参数				
工作频率（f _{TX} ）	24.0	24.1	24.25	GHz
发射功率（P _{out} ）		6	8	dBm



3. 模块尺寸及引脚说明

3.1. 模块尺寸

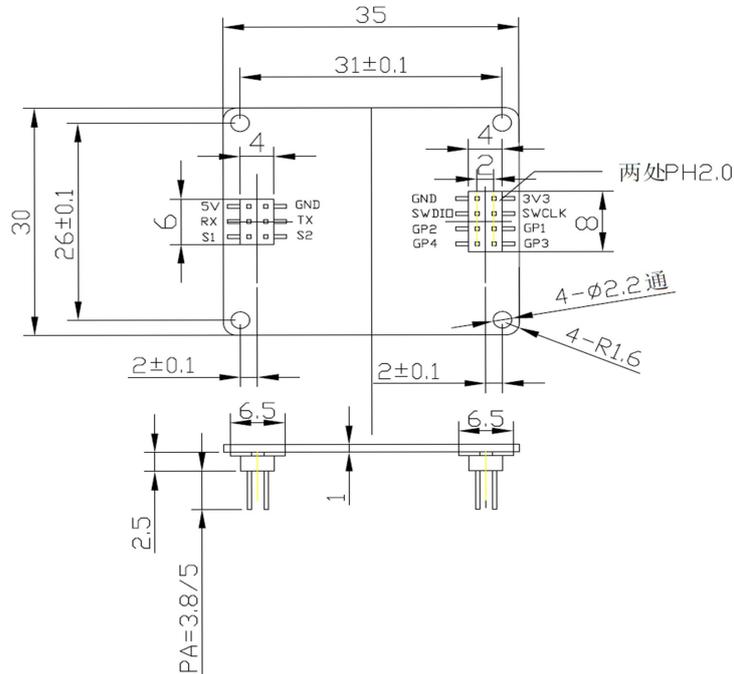


图 1 雷达模块结构示意图

3.2. 引脚说明

接口	引脚	描述	典型值	说明
接口 1	1	5V	5.0V	电源输入正端
	2	GND		地
	3	RX		串口接收
	4	TX		串口发送
	5	S1	3.3V/0V	有人/无人
	6	S2	3.3V/0V	静止/活跃
接口 2	1	3V3	3.3V	输出电源
	2	GND		地
	3	SL		保留
	4	SD		保留
	5	GP1		备用扩展引脚
	6	GP2		备用扩展引脚
	7	GP3		备用扩展引脚
	8	GP4		备用扩展引脚

- 注：
- 1) S1 输出：高电平-有人，低电平-无人；
 - 2) S2 输出：高电平-活跃，低电平-静止
 - 3) GP1~GP4 为参数选择控制端，可根据用户需求重定义。



使用接线图

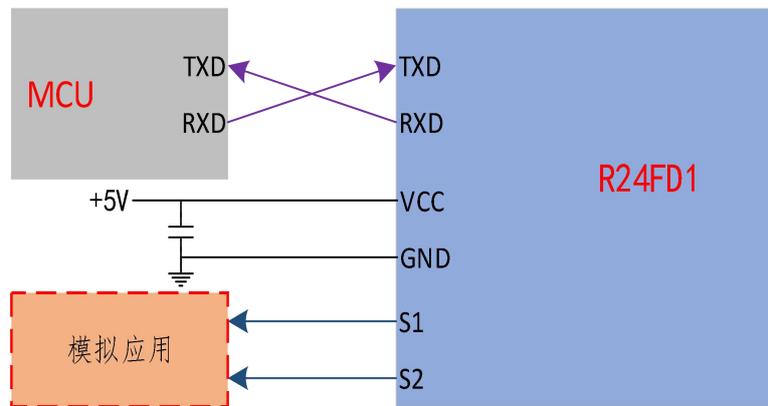


图 2 雷达模块与外设连线示意图

4. 模块工作模式

4.1. 雷达模块探测范围

本雷达模块天线方向图如下图所示。雷达水平及俯仰 3dB 波束宽度为 100° 。

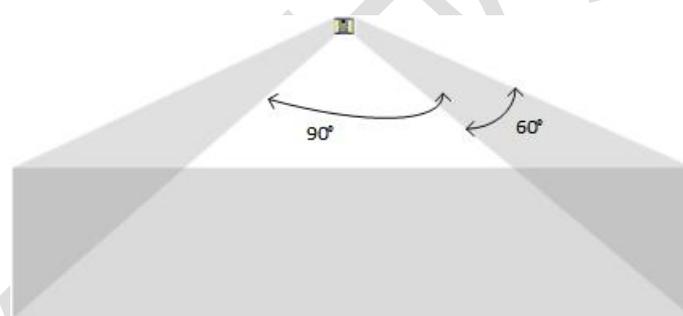


图 3 雷达波束仿真图

雷达置顶安装模式下，雷达安装高度 H 为 2~2.5 米，雷达有效波束投影半径 R 为 2~4 米，如下图所示。

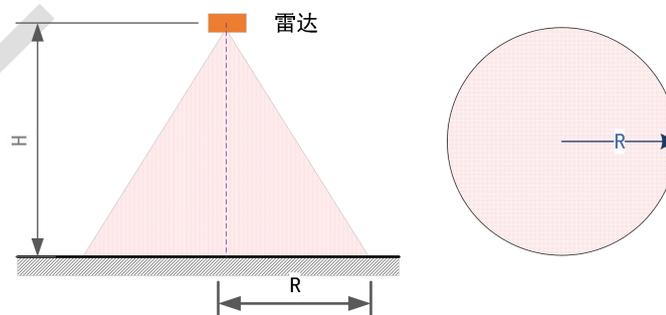


图 4 雷达安装示意图

注：随着雷达安装高度、安装场景的改变，雷达有效作用范围有相应的改变。



4.2. 主要功能

本雷达模块主要功能包括：

- A、运动检测功能；
- B、微动检测功能；
- C、呼吸检测功能；
- D、睡眠状态评估功能；

4.3. 安装方式

本雷达模块建议安装方式为置顶安装。如下图所示为置顶安装。

雷达垂直安装，水平偏离角度 $\leq 5^\circ$ ，保证雷达主波束覆盖探测区域；雷达安装高度建议为 2-3 米；雷达前面无明显遮挡物及覆盖物。

受雷达安装高度及雷达波束范围影响，水平作用区域为雷达投影半径 2~4 米区域。

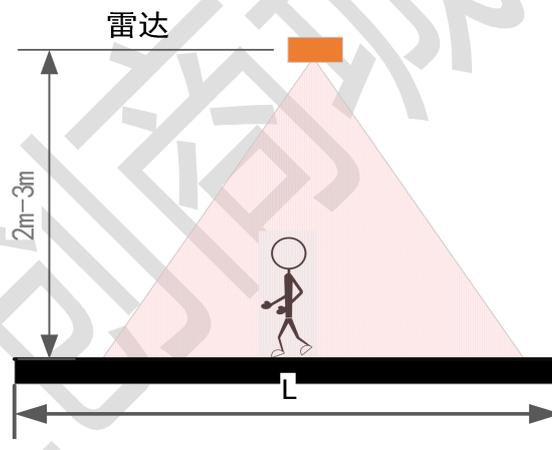


图 5 雷达安装示意图

注意：

- A、雷达安装方式，需要雷达主波束覆盖人体主要活动区域，并尽可能法线方向正对；
- B、模块工作时，模块表面不应该有金属物遮挡；
- C、受电磁波传输特性影响，雷达作用距离与目标 RCS、目标覆盖物材质及厚度相关联，雷达有效作用距离会有一定程度变化。
- D、对应静止状态人体探测，不同体位会对雷达作用距离有影响，雷达不保证



所有状态均达到最大作用距离。

4.4. 雷达模块工作模式

本雷达模块可以在参数设置控制下，实现参数运行模式或状态运行模式。

◇ 参数运行模式

参数运行模式是直接见雷达探测参数输出，用户可以在输出参数基础上开发相关应用；

◇ 状态运行模式

状态运行模式是雷达模块通过统计分析处理后，综合评估当前检测区域人员状态，用户可以直接利用该结果。

雷达输出参数主要包括如下表所示参数：

	输出参数	参数说明
1	呼吸参数	对探测到的人体呼吸频率、呼吸时间间隔等参数实时测量，并周期性输出；
2	运动参数	对探测区域内人体运动频率、运动强度等实时监测，并进行参数评估后周期性输出；
3	场景状态评估	基于人体呼吸参数、运动参数等，周期性场景评估。

注：雷达参数输出时间间隔预设 10 分钟，可以根据用户需求调整时间间隔。

4.5. 安装方式

本雷达模块建议安装方式包括倾斜安装及置顶安装。

4.5.1. 倾斜安装

如图 5 所示为倾斜安装。本安装方式主要用于床旁按照。

雷达安装高度建议为床以上高度 0.5-1.5 米；雷达下视倾斜角度范围为 $10^{\circ} \sim 30^{\circ}$ ，雷达前面无明显遮挡物及覆盖物，雷达正向波束能够覆盖被测人员胸腔。雷达与人体间最大距离不超过 2 米。

该模式下，雷达正下方及邻近区域可能存在监视盲区。

随着下视倾角增加，静态人体探测距离会明显压缩。

受雷达天线辐射特性影响，偏离雷达法线方向位置，雷达有效作用距离会降低。

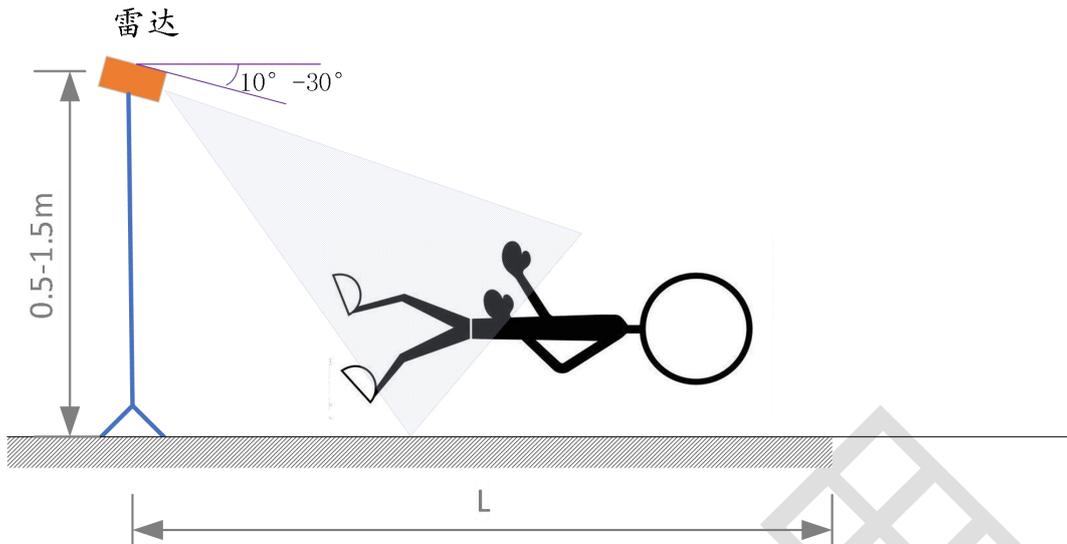


图 6 斜下视安装示意图

4.5.2. 置顶安装

如图 4 所示为置顶安装。本安装方式主要针对平躺状态下的人体监测，比如卧室、养老场所、病床等。

雷达垂直安装，水平偏离角度 $\leq 5^\circ$ ，保证雷达主波束覆盖探测区域；雷达安装高度建议为床以上 1~2 米；雷达前面无明显遮挡物及覆盖物。

受雷达安装高度及雷达波束范围影响，水平作用区域长度 $L \approx 3$ 米~5 米。

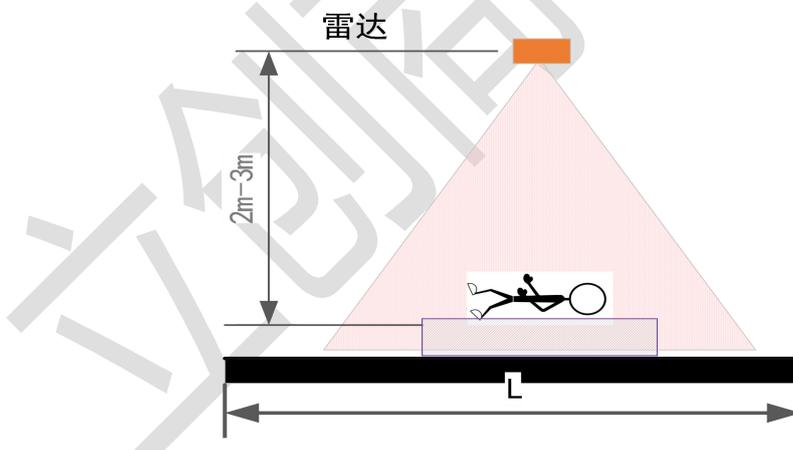


图 7 斜下视安装示意图

注意：

- E、上述不同安装方式，均需要雷达主波束覆盖人体主要活动区域，并尽可能法线方向正对；
- F、斜下安装时，由于覆盖区域水平投影变化，水平作用距离将对应减小；
- G、模块工作时，模块表面不应该有金属物遮挡；



H、受电磁波传输特性影响，雷达作用距离与目标 RCS、目标覆盖物材质及厚度相关联，雷达有效作用距离会有一定程度变化。

I、对应静止状态人体探测，不同体位会对雷达作用距离有影响，雷达不保证所有状态均达到最大作用距离。

5. 典型应用模式

5.1. 卧室安装及应用

针对特定应用，实时卧床人员相关信息，比如有人/无人、睡眠状态、睡眠深度、运动信息等，进而给出相关信息，实现特定应用。该模式下，雷达需要置顶安装。

基于该模式应用，可以实现应用包括

- ◆ 老人看护
- ◆ 康养看护
- ◆ 酒店应用
- ◆ 家庭健康



6. 模块接口协议

6.1. 接口介绍

雷达模块与上位机采用串口通信模式，串口通信定义如下：

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

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

- a. 起始码：1Byte，固定为0X55。
- b. 数据长度：2 Byte，低字节在前，高字节在后。
 $\text{长度} = \text{数据长度} + \text{功能码} + \text{地址码1} + \text{地址码2} + \text{数据} + \text{校验码}$ 。
- c. 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：
 $0X04$
- d. 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。
- e. 数据：n Byte
- f. 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



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
1	读命令 0x01	标识查询 0x01	设备 ID 0x01		
2			软件版本 0x02		
3			硬件版本 0x03		
4			协议版本 0x04		
11		雷达信息查询 0x03	体征参数 0x06		
12		系统参数 0x04	阈值档位 0x0C		
16	写命令 0x02	系统参数 0x04	阈值档位 0x0C	枚举范围 1, 2, 3, 4, 5	分别对应 1 2 3 4 5 档
			无人累计时间调节 0x09	1Byte 数据	以分钟为单位
17	被动上报命令 0x03	上报模块标识 0x01	设备 ID 0x01	12 Byte 数据	
18			软件版本 0x02	10 Byte 数据	
19			硬件版本 0x03	8 Byte 数据	
20			协议版本 0x04	8 Byte 数据	
27		上报雷达信息 0x03	环境状态 0x05	无人状态 00 FF FF	
28				有人静止 01 00 0F	
29				有人运动 01 01 01	
30			体征参数 0x06	4 Byte Float 数据	
31	主动上报命令 0x04	上报雷达信息 0x03	无人状态 00 FF FF		
32			环境状态 0x05	有人静止 01 00 0F	
33			有人运动 01 01 01		
34			运动体征参数 0x06	4 Byte Float 数据	
35	睡眠雷达数据 上报 0x05	呼吸参数 0x01	呼吸频率 0x01	1Byte 整型数据	单位：秒
36			最大呼吸间隔 0x03	1Byte 整型数据	
37		体动参数 0x02	体动幅度 0x04	1Byte 整型数据	0-100%
38		场景评估 0x03	入床/离床 0x07	离床 0x00	
39				入床 0x01	
40			睡眠评估 0x08	清醒 0x00	
41				浅度睡眠 0x01	
42		深度睡眠 0x02			
43		呼吸异常预警 0x09		呼吸正常 0x00	
44				呼吸报警 0x01	



7. 注意事项

7.1. 启动时间

由于本模块在初始上电开始工作时，需要对模块内部电路完全复位，并对环境噪声进行充分评估，才能保证模块正常工作。因此模块初始上电工作时，需要开机稳定时间 $\geq 30s$ ，才能保证后续输出参数的有效性。

7.2. 有效探测距离

雷达模块的探测距离与目标 RCS、环境因素关联较大，有效探测距离可能随着环境及目标改变而变化，本模块暂时不具备测距功能，因此有效探测距离在一定范围波动属于正常现象。

7.3. 雷达生物探测性能

由于人体生物特征属于超低频、弱反射特征信号，雷达处理中需要相对长时间累积处理，在累积过程中，可能诸多因素影响雷达参数，因此偶发性的探测失效是正常现象。

7.4. 电源

雷达模块对电源品质的要求，高于常规低频电路。在对模块供电时，要求电源无门限毛刺或纹波现象，且有效屏蔽附件设备所带来的电源噪声。

雷达模块需良好的接地，由于其他电路带来的地噪声，也可能引起雷达模块性能下降甚至工作异常；最常见的是导致探测距离变近或误报率增加。

为了保证模块内部 VCO 电路的正常工作，对本模块供电要求为+5V~+6V 供电，特别是电源电压不能低于 5V。

外部电源必须提供足够的电流输出能力和瞬态响应能力。

8. 常见问题

9. 免责声明



我认为，在出版时尽量做到文档描述的准确无误。考虑到产品的技术复杂性及工作环境的差异性，但仍难以排除个别不准确或不完备之描述，故本文档仅作为用户参考之用。我公司保留在不通知用户的情况下对产品作出更改的权利，我公司不做任何法律意义上的承诺和担保。鼓励客户对产品和工具最近的更新提出意见。

10. 版权说明

本文档所提及的元件及器件，皆为对其版权持有公司所公布的资料之引用，其修改和发布的权利均属于其版权持有公司，请在应用时通过适当的渠道确认资料的更新情况以及勘误信息，我公司不对这些文档具有任何权利和义务。

1. 联系方式

云帆瑞达科技（深圳）有限公司

电子邮箱：sales@iotrada.com

lg@iotrada.com

电话：0755-88602663

地址：深圳市福田区车公庙苍松大厦北座 509B

附录1

```
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  
25. };
```



```
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, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 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, unsigned
   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. }
```