

76-81GHz 连续调频波雷达液位计

产品使用说明书



上海胜蚌电子信息科技有限公司

1. 产品介绍

1.1 特点

产品是指工作在 76-81GHz 的调频连续波 (FMCW) 雷达产品。产品量程可以达到 65m，盲区在 10cm 以内。由于它工作频率更高，带宽更大，测量精度更高。产品提供支架的固定方式，无需现场布线使得安装便捷简易。

主要优势如下：

- 基于自研的 CMOS 毫米波射频芯片，实现更紧凑的射频架构，更高的信噪比，更小的盲区。
- 5GHz 工作带宽，使产品拥有更高的测量分辨率与测量精度。
- 最窄 6° 天线波束角，安装环境中的干扰对仪表的影响更小，安装更为便捷。
- 一体化透镜设计，体积精巧。
- 低功耗运行，寿命大于 3 年。
- 水位超上、下限（可配置）上传报警信息。

2. 技术规格

雷达液位计技术规格

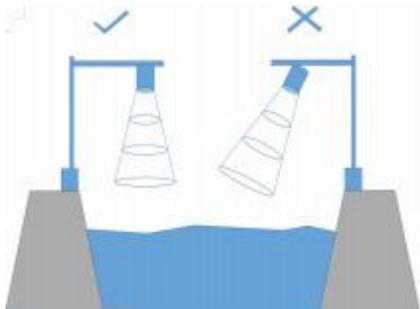
发射频率	76GHz~81GHz
测量范围	0.1m ~70m
测量精度	± 1mm
波束角	6°
供电范围	9~36 VDC
通讯方式	RS485
	-40~85°C
外壳材质	PP / 铸铝 / 不锈钢
天线类型	透镜天线
推荐线缆	7*0.3mm:
防护等级	IP68
安装方式	支架 / 螺纹

3. 安装

安装需要注意的两点：（1）保证仪表垂直于水面（2）避免发射波束照射到干扰物，产生虚假回波。

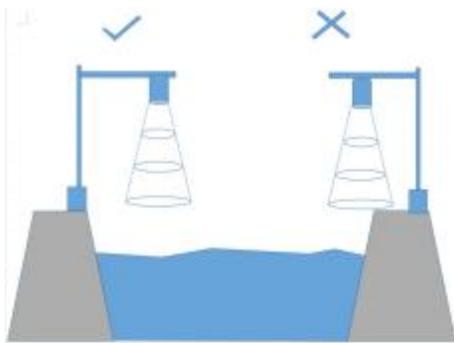
典型工况参见以下几点。

- 保证水位计垂直于水面安装，倾斜将接受信号幅度变弱，影响正常测距。



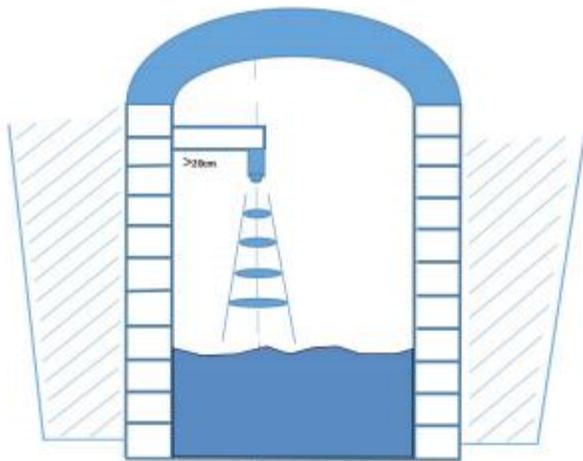
仪器安装位置示意图

- 保证波束范围内没有干扰物，如河岸岸边。



仪器安装位置示意图

- 仪器安装至少离侧壁 20cm ，地下管网安装尽量靠近下水井中央，否则井壁容易产生干扰信号，影响测量判断。



安装至少离容器壁 20cm

4. 连接

4.1 应用场景 4：485 传输（红：+ 蓝：- 绿：485A 黄：485B）



产品接线图

当用户使用 RS485 传输时，液位计通过防水接头引出 4 根线，按照图中的线序连接用户的产品，然后对插锁定即可，图中 B 端线的长度根据用户需求可定制。串口参数：RS485，波特率 --9600，数据位 --8，停止位 -- 1，校验位 -- 无。注意设备供电电压在 9V--36V 范围内，建议使用 12V 供电。

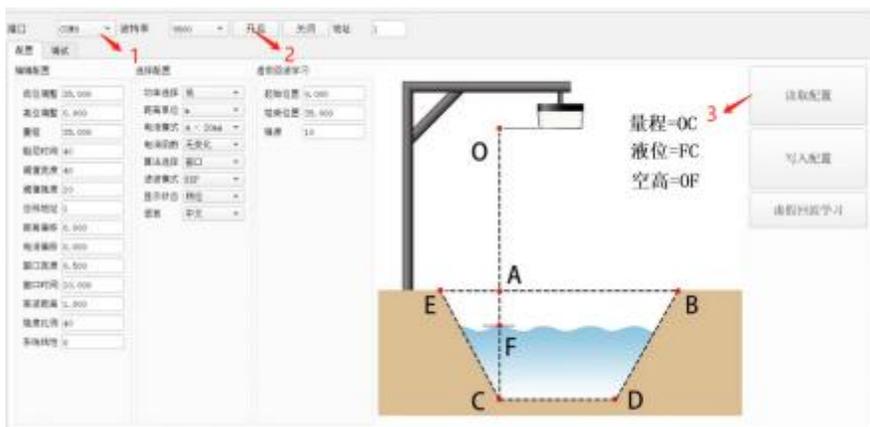
5. 液位计操作

5.1 显控说明

显控用来显示液位值，当按下按键后开始测量并显示，20 秒无操作显控熄屏。

5.2 PC 上位机界面说明

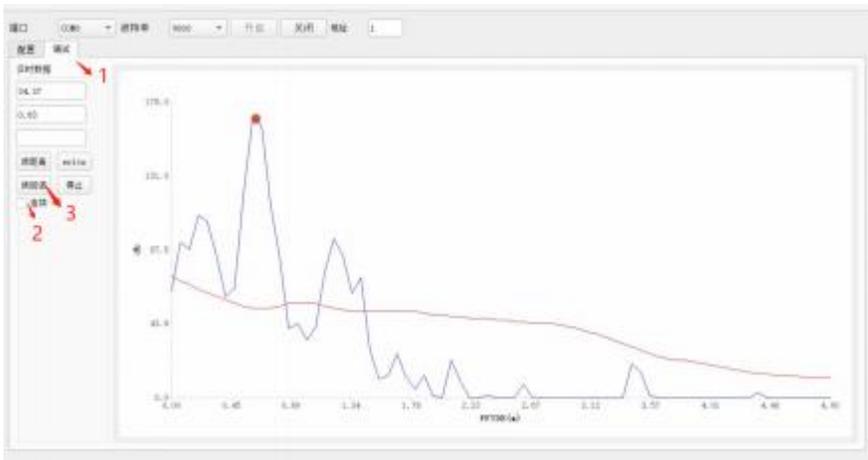
5.2.1 软件设置



配置上位机串口参数

按照图中顺序配置串口参数，波特率 9600，连接成功后双击“读取配置”可以设置和读取参数。

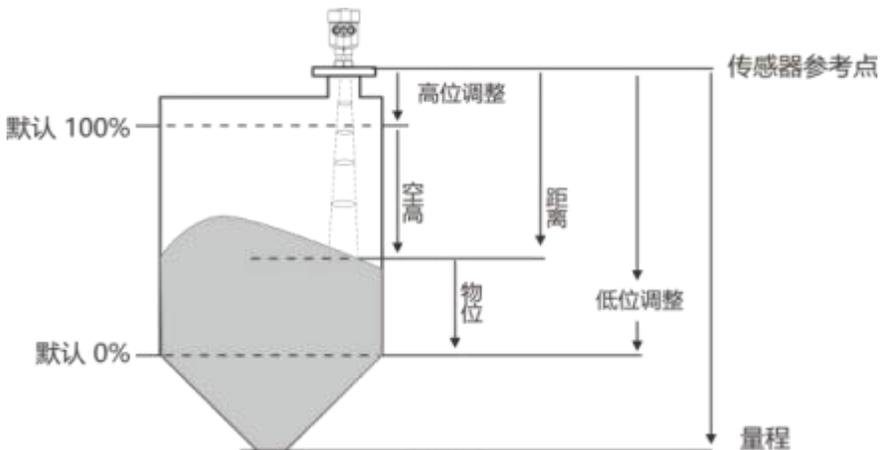
5.2.2 回波曲线



上位机波形界面

5.2.2.1. 【测量类型】

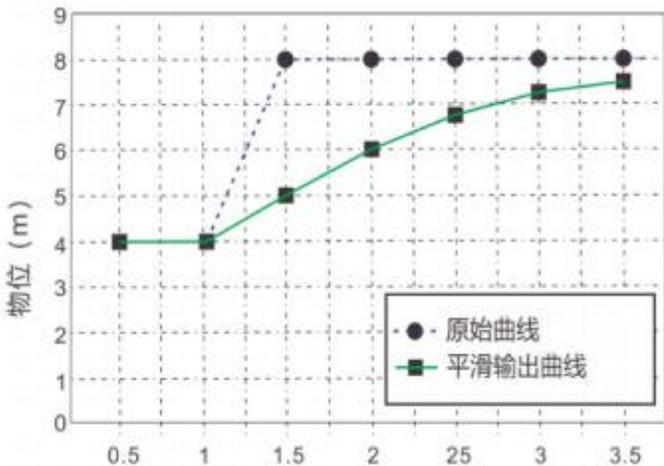
【测量类型】可以根据现场需要，选择距离信息输出形式。核心参数是传感器参考点到待测物的测试距离。给出了实时值具体的计算公式。其中高位调整默认是 0。



参数名称	显示
默认值	物位
关联配置	无
选项意义	物位模式：实时值 = 低位调整 - 距离 (最小为 0) 空高模式：实时值 = 距离 - 高位调整 (最小为 0)
特别事项	如果距离 > 低位调整，物位 = 0； 如果距离 < 高位调整，空高 = 0；

5.2.2.2. 【阻尼滤波】

【阻尼时间】的作用是平滑测量结果中的突变，也就是阻尼滤波器。例如，阻尼时间为 2 秒，被测物体位置在 t 时刻发生阶跃变化，测量输出值会缓慢发生变化，在第一个 2 秒内，完成 63.2% 的变化，并在第 10 秒 (5 倍的设置值) 跟随到实际位置，如下图所示。



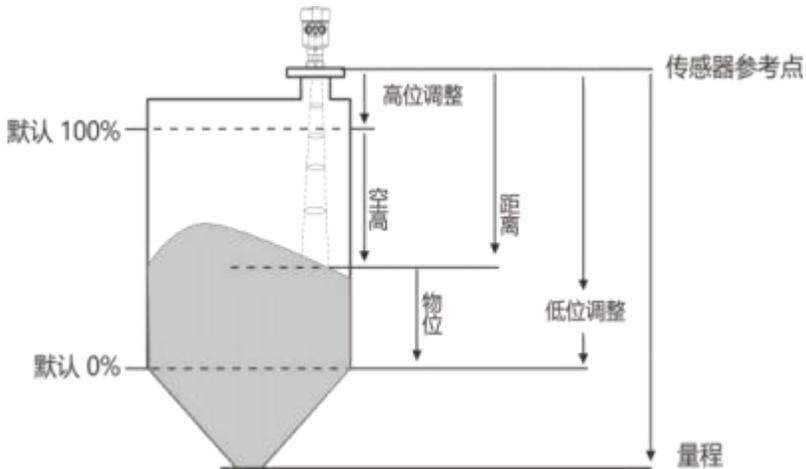
阻尼时间编辑界面与含义

阻尼时间说明

参数名称	阻尼时间
参数范围 (S)	0 ~100
默认值 (S)	40
关联配置	无
选项意义	阻尼输出，提升信号稳定性
特别事项	由于测量时间短，暂时未使用此参数

5.2.2.3. 【低位调整】

【低位调整】即低位调整点，具体的定义参见下图。



低位调整编辑界面与定义

低位调整说明

参数名称	低位调整
参数范围 (m)	0.1~ 量程
默认值 (m)	30
关联配置	如果设置的低位调整 < 高位调整 +0.1，则低位调整 = (高位调整 +0.1)；
特别事项	低位调整点与量程无关，只影响物位的计算

5.2.2.4. 【高位调整】

【高位调整】即高位调整点。

高位调整说明

参数名称	高位调整
参数范围 (m)	0 ~ (低位调整 -0.1)
默认值 (m)	0
关联配置	如果设置的高位调整 > (低位调整 -0.1) , 则高位调整 = (低位调整 -0.1) ;
特别事项	高位调整与盲区设定无关, 只影响空高的计算

5.2.2.5. 【距离偏移】

【距离偏移】用于修正传感器的参考点。仪表默认的参考点在出厂时被调校到如下图 a 点所示的位置。如果想将参考点向下调校到 b 点, 则在设置中输入 h1。



距离偏移编辑界面

距离偏移说明

参数名称	距离偏移
参数范围 (m)	(- 内置偏移)~10m
默认值 (m)	0
关联配置	无
选项意义	修正传感器的参考点零点。传感器输出值的范围仍为【盲区】~【量程】之间，实际传感器测量范围归一化到初始参考点为：【距离偏移 + 盲区】~【距离偏移 + 量程】。
特别事项	

5.2.2.6. 【量程设定】

为了测量得到正确的结果，需设置仪表的量程范围，具体含义参见下表。

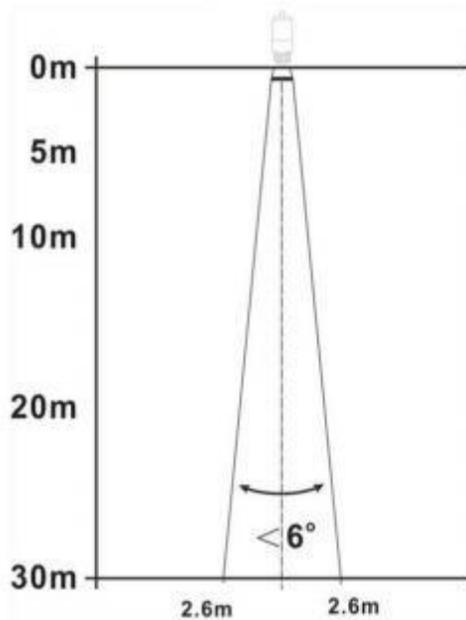
表 6 - 10 量程说明

参数名称	量程
参数范围 (m)	1~65
默认值 (m)	35
关联配置	如果设置量程 < (盲区 + 0.1m)，则量程自动设置为 (盲区 + 0.1m)。
选项意义	算法处理时会忽略量程之外的回波，合理设置量程可以避免多次反射干扰以及可能的范围之外的干扰信号。
特别事项	此量程并非指仪表的远端测量极限，只用作限定算法区域。仪表测量极限请参见技术规格一节。

注：盲区与量程决定了算法应用的具体范围，可以通过合理设置，规避干扰与虚假回波，实现快速且稳定的测量。

6. 附录 B：术语表

波束角：以比最大值低 3dB 作为界限的波束宽度。雷达液位计系列波束角 6°



仪表的雷达波束空间几何形状示意图

距离分辨率：距离分辨率是指两个靠在一起的物体间隔多远距离时，物位雷达能够分辨出是两个物体而不是一个物体且能够测出它们各自的距离。如果两个物体的距离间隔小于物位雷达的距离分辨率，那么雷达只能测得一个距离值，此距离值不等于其中任何一个物体的距离值，而是两个物体距离值的综合。雷达液位计的调频带宽 $B=5\text{GHz}$ ，

最小距离分辨率 = 光速 / 工作带宽 / 2 \approx 3cm。

测量精度：如果只有一个物体且这个物体移动了很小的距离，物位雷达是否能识别距离变化。分辨出单个物体移动距离的指标叫做精度。雷达液位计的中频信号进行自有算法分析，测量精度为 0.5mm。

盲区：(1) 指仪表的近端的测量极限，盲区内仪表无法测量

回波：雷达接受到的反射的信号。

发射锥体：天线波束角度的延伸。

虚假回波：任何不是所需目标产生的回波。一般来说，虚假回波由容器的障碍物产生。多重回波：在目标回波距离出现的多次反射回波，可能为 2 次，3 次。

量程：(1) 指仪表的最远测量极限 (2) 特殊的，指人为设定的最远距离，该距离以外，仪表处理数据的时候不考虑。

重复性：在相同的情况下测量同一个变量的多次测量的偏差程度。

阈值曲线：一个随时间变化的曲线，作为阈值，超过其的回波被认为是有效的。

一、通信协议硬件接口参数

水位计采用串口通信，默认参数如下：

通信参数	串口电平	波特率 /bps	奇偶校验	数据长度 /bit	停止位 /bit
串口	TTL	9600	无 (none)	8	1

每帧间隔超时时间 50ms。

二、通信协议格式

水位计对外通信采用 ModbusRTU 通信协议，每一个完整的数据帧包含：地址域、功能码、数据和校验。其中校验为数据帧的 CRC16 校验数据，低字节在前，高字节在后。水位计雷达出厂默认地址为 1。

请求命令格式和雷达回复数据格式说明如下：

查询参数格式：功能码 0x03 请求：

回复：

设备地址	功能码	数据长度	寄存器值	CRC
(1 字节)	(1 字节)	(1 字节)	(2xN 字节)	(2 字节)

N：寄存器数量

查询参数格式：功能码 0x04 请求：

回复：

设备地址	功能码	数据长度	寄存器值	CRC
(1 字节)	(1 字节)	(1 字节)	(2xN 字节)	(2 字节)

N：寄存器数量设置参数格式：功能码 0x10 请求：

回复：

设备地址	功能码	起始地址	寄存器数量	CRC
(1 字节)	(1 字节)	(2 字节)	(2 字节)	(2 字节)

三、通信协议命令说明

以下对每个命令进行详细说明：

1、设置指令：功能码 0x10

读取命令：功能码 0x03

起始地址 0x2000

偏移地址	命令名称	数据格式	单位
0	低位调整	0-65000	mm
1	高位调整	0-65000	mm
2	量程	0-65000	mm
3	阻尼时间	0-100	无
4	阈值宽度	10-200	无
5	阈值强度	0-200	Db
6	总线地址	1-32	无
7	距离偏移	-10000-10000	mm
8	电流偏移	-10000-10000	mA/1000
9	窗口宽度	0-35000	mm
10	窗口时间	0-65000	ms
11	首波距离	0-35000	ms
12	强度比例	10-90	百分比
13	物料选择	0- 液体，1- 固体	

14	距离单位	0-m , 1-cm , 2-mm	
15	电流模式	0-(4-20mA),1-(20 - 4mA)	
16	电流仿真	默认 0	
17	电流函数	0- 无变化,1-22mA,2-3.6mA	
18	算法选择	0- 窗口 ,1- 首波 ,2- 强度	水位计选择 2
19	滤波模式	0-EXP , 1-KEF	
20	显示状态	0- 物位 , 1- 空高	
21	语言	0- 中文 , 1-English	

注释：每一个参数占用一位寄存器地址，数据为 U16 或 I16 格式

举例：

读低位调整 命令如下：

设备地址	功能码	起始地址	寄存器数量	CRC
0x01	0x03	0x2000	1	(2 字节)

写低位调整 命令如下：

设备地址	功能码	起始地址	寄存器数量	CRC
0x01	0x10	0x2000	1	(2 字节)

CRC 码的计算规则：

预留 16 位寄存器为十六进制 FFFF(即全为 1)。称此寄存器位 CRC 寄存器；

把第

一位 8 位数据与 16 位 CRC 寄存器的地位相异或，把结果放于 CRC 寄存器：

检查最低位是否为 0，如为。则把寄存器的内容右移一位（朝低位），

用填补高位；

如为 1 把寄存器的内容右移一位（朝低位），用填补高位，然后 CRC 寄存器与多项式 A001

（1010 0000 0000 0001）进行异或；4.3.4、重复步骤 3，直到右移 8 次，这样整个 8 位数据全部进行了处理：

重复步骤 2 到步骤 4，进行下一个 8 位数据的处理；

最后得到的 CRC 寄存器即为 CRC 码。将 CRC 结果放入信息帧时，将高地位交换，低位在前。

通讯协议范例：

主机发送数据：

站号	功能码	起始地址	读取点数	校验码	意义
01	03	0000	0001	840A	读取空高，单位 cm
01	03	0001	0001	D5CA	读取空高，单位 mm
01	03	0002	0001	25CA	读取液位高，单位 cm
01	03	0003	0001	740A	读取液位高，单位 mm

附录：CRC 校验

```
static const uint8_t aucCRCHi[] ={  
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
```

```
};
```

```
static const uint8_t aucCRCLo[] = {
```

```
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

};

```

uint16_t usCRC16(uint8_t* pucFrame, uint16_t usLen)

{

uint8_t   ucCRCHi =0xFF;

uint8_t   ucCRCLo =0xFF;

uint16_t   iIndex;

while( usLen-- )

{

iIndex = ucCRCLo ^ *(pucFrame++);

ucCRCLo = ( uint8_t)( ucCRCHi ^ aucCRCHi[iIndex] ); ucCRCHi =
aucCRCLo[iIndex];

}

return ( uint16_t)( ucCRCHi << 8 | ucCRCLo);

}

```



