



# 广州西博臣科技有限公司

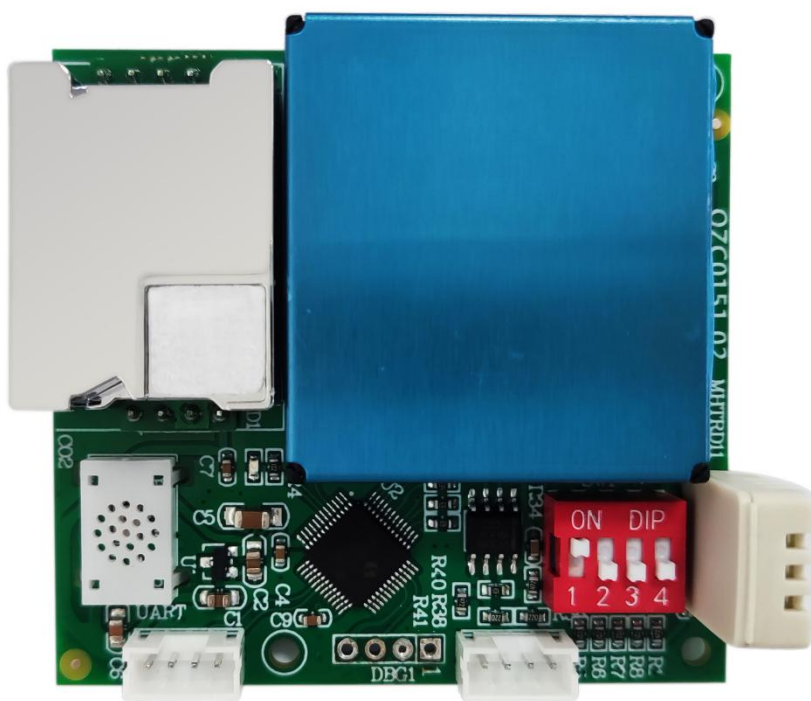
GUANGZHOU CYBERSEN TECHNOLOGY CO.,LTD.

## 七合一空气质量检测模块

CHTG-7

产品说明书

版本号：V1.0



## 一、产品描述

CHTG-7 七合一空气质量检测模块是一款高性价比的数字输出传感器模块，采用 UART 和 RS485 输出模式，集 CO<sub>2</sub>、甲醛、TVOC、激光粉尘 PM2.5/PM10 颗粒物、温度、湿度传感器于一体(可根据需求定制组合)。可对所处环境进行实时全面检测，具有良好的稳定性，非常方便客户使用。

## 二、应用领域

- 酒店房间空气质量监控
- 新风换气系统
- 空气净化器，空调
- 空气质量监测设备
- 厨卫换气控制系统
- 智能家居设备

## 三、产品特点

- 同时输出 CO<sub>2</sub>、甲醛、TVOC、PM2.5、PM10、温度、湿度七组数据
- 灵敏度高，数据稳定
- UART、RS485 输出
- 温度分辨率 0.1℃，湿度分辨率 0.1%RH
- 每 2 秒自动通过 UART 信号输出七组传感器监测数据

## 四、量程精度

类别	测量分辨率	测量范围	测量精度
CO <sub>2</sub>	1ppm	400ppm~5000ppm	±50ppm+5%读数
甲醛	1ug/m <sup>3</sup>	0ug~1000ug/m <sup>3</sup>	±25%
TVOC	1ug/m <sup>3</sup>	0ug~2000ug/m <sup>3</sup>	±25%
PM2.5	1ug/m <sup>3</sup>	0ug/m <sup>3</sup> ~999ug/m <sup>3</sup>	±10%
PM10	1ug/m <sup>3</sup>	0ug/m <sup>3</sup> ~1000ug/m <sup>3</sup>	±10%
温度	0.1℃	-20℃~60℃	±0.5℃
湿度	0.1%RH	5~95%RH	±5%RH

## 五、技术参数

接线端子	PH2.0 母座
输出信号	UART、RS485
工作电压	5.0±0.2VDC
平均工作电流	≤80mA
预热时间	2 分钟(甲醛和 TVOC 需要预热稳定，其它参数上电即显示)
工作温度	0℃~50℃
工作湿度	≤95%RH

外形尺寸	62*58*19mm (L×W×H)
使用寿命	5 年

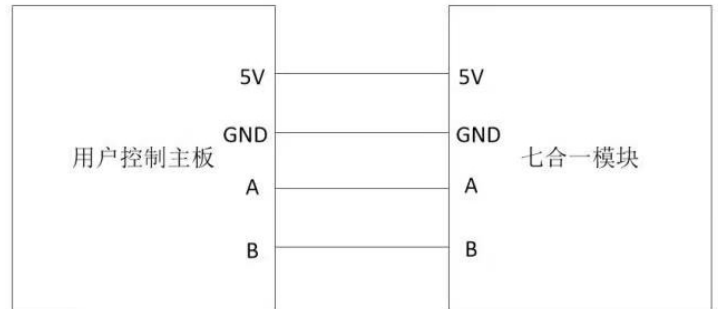
## 六、接口定义

### 1、RS485

接口	名称	功能
1	B	RS485B
2	A	RS485A
3	GND	电源地
4	VDD	接电源 5VDC

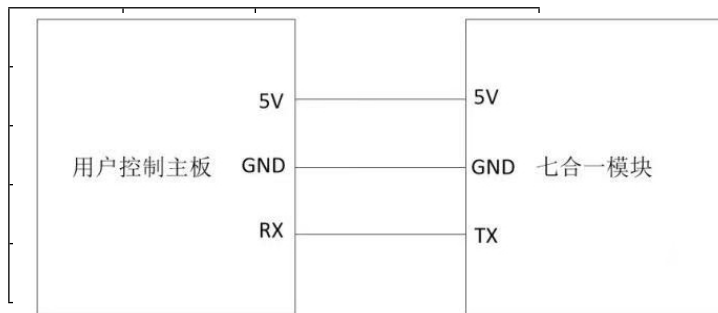
RS485 协议见附件

### 连接示意图



### 2、UART

### 连接示意图



串口数据流格式，波特率：9600bps；数据位：8 位；校验位：无；停止位：无。

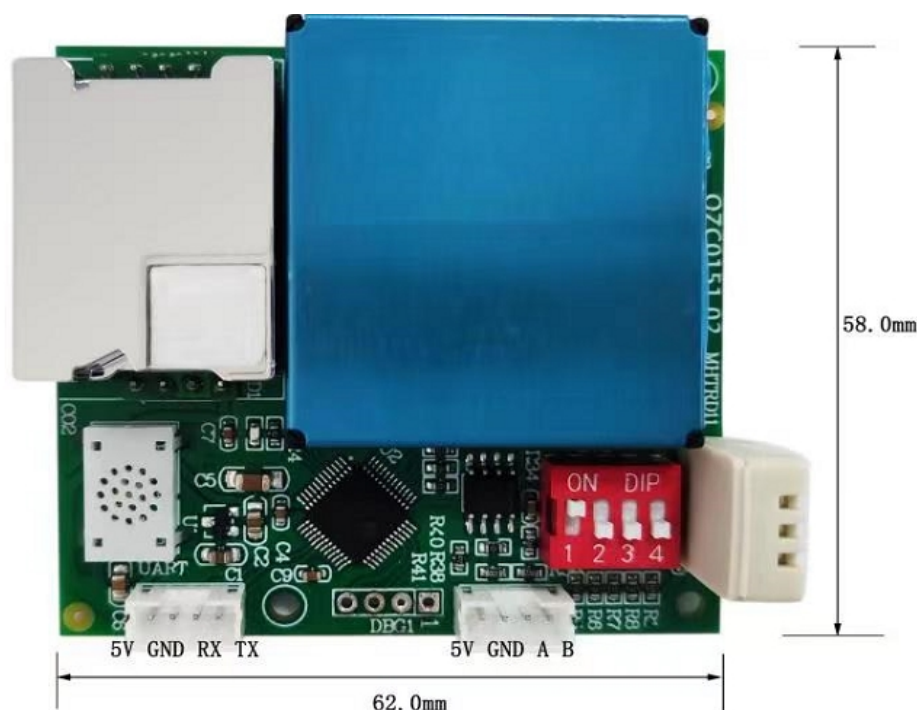
#### 2.1 UART 通信协议

字节	名称	说明
B0	帧头 1	固定值 3Ch
B1	帧头 2	固定值 02h
B2	数据	CO <sub>2</sub> 高字节
B3	数据	CO <sub>2</sub> 低字节
B4	数据	甲醛高字节
B5	数据	甲醛低字节
B6	数据	TVOC 高字节
B7	数据	TVOC 低字节
B8	数据	PM2.5 高字节
B9	数据	PM2.5 低字节
B10	数据	PM10 高字节

B11	数据	PM10 低字节
B12	数据	温度整数部分
B13	数据	温度小数部分
B14	数据	湿度整数部分
B15	数据	湿度小数部分
B16	校验和	湿度校验和

说明：校验和 B16 等于：B0+B1+B2+B3+ . . . . . B15 的值, 取低 8 位。当温度的数据 B12 的 bit7=1 时，代表是负温度，B12 的 bit7=0 时，是正温度。例如，当 B12=9Bh 时，此时 bit7=1，表示是负温度，此时实际温度就是-27℃；如果 B12=1Bh, 此时 bit7=0, 代表是正温度，此时实际温度就是 27℃。

## 六、外观尺寸图 【62\*58\*19mm (L×W×H)】



## 七、注意事项

- 初次上电使用需预热 2 分钟以上
- 请勿将该模块长时间置于高浓度有机气体及腐蚀性气体中
- 请勿将模块安装在强对流气体环境下使用
- 避免地面沙尘，飘絮物等大尘埃颗粒甚至絮状物污染导致风扇缠绕阻转，建议使用设备采取适当的预过滤处理
- 用户切勿拆解传感器，包括金属屏蔽壳，以防出现不可逆破坏
- 本传感器适用于普通室内环境

附：

## RS485 通讯协议

### Ver1.0

#### 1、概述

通信协议详细地描述了七合一空气质量检测模块 CHTG-7 输入和输出命令、信息和数据，以便第三方使用和开发。

物理接口：

连接上位机的主通信口，采用标准串行 RS-485 通讯口。

信息传输方式为异步方式，起始位 1 位，数据位 8 位，停止位 1 位，无校验。

数据传输缺省速率为 9600b/s

#### 2、通信协议详述

##### 2.1 通讯方式

- 1) 所有回路通信应遵照主/从方式。在这种方式下，信息和数据在单个主站和从站（监控设备）之间传递。
- 2) 支持广播模式。
- 3) 无论如何都不能从一个从站开始通信。
- 4) 若主站或任何从站接收到含有未知命令的包裹，则该包裹将被忽略，且接收站不予响应。

##### 2.2 返回数据帧结构描述

每个数据帧组成如下：（RTU 模式）

地址

功能代码

数据数量

数据 1

...

数据 n

CRC 16 位校验

#### 3、传输格式

##### 1) 主机发送读地址命令：

地址	功能码	寄存器地址 高位	寄存器地址 低位	数据个数 高位	数据个数 低位	CRC 16 位校验
00	03	01	00	00	01	Xxxx 低位在前

从机返回地址：

地址	功能码	字节长度	地址高位	地址低位	CRC 16 位校验
00	03	02	00	xx	Xxxx 低位在前

例：发送：00 03 01 00 00 01 84 27

响应：00 03 02 00 **addr** CRCL CRCH 若 **addr=2** 则读取的传感器地址为 2

## 2) 主机发送读数据命令:

地址	功能码	数据起始地址高位	数据起始地址低位	返回数据个数高位	返回数据个数低位	CRC 16 位校验
xx	03	00	02	00	07	低位在前

(目前仅支持读取所有数据, 从地址 0002 开始, 读取 7 个数据值), 只有 7 个传感器数据, 对应 7 个传感器地址, 数据起始地址高位永远为 0, 返回数据个数高位永远为 0.

内部报文信息

起始地址	数据个数	说明
只能是如下:		
0x0002	1	CO2 浓度
0x0004	1	甲醛浓度
0x0006	1	TVOC 浓度
0x0008	1	PM2.5 浓度
0x000A	1	PM10 浓度
0x000C	1	温度值
0x000E	1	湿度值

起始地址只能用以上列表中的地址, 否则不回应. 数据个数如果超出, 也不回应, 比如, 起始地址 0X000C, 可以返回的数据个数只可能是 1 或 2, 如果要回 3 个数据则不回应. 同理, 地址 0X000E, 返回数据个数只能是 1. 返回数据个数 0, 也不回应.

从机传感器返回值:

从机地址	功能码	数据个数	数据 N	CRC
XX	03	xx	xx	xxxx

字节长度仅指数据长度。

N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13
CO2 高位	CO2 低位	甲醛 高位	甲醛 低位	TVOC 高位	TVOC 低位	PM2.5 高位	PM2.5 低位	PM10 高位	PM10 低位	温度 高位	温度 低位	湿度 高位	湿度 低位

示例: 主机发送 01 03 00 02 00 07 CRCL CRCH

响应: 01 03 0E 02 26 00 26 00 34 00 0E 00 11 01 1C CRCL CRCH

## (3) 强制设置地址命令

此命令可以强制设置从机地址, 地址值 1 到 255, 此后, 从机地址就是主机设置的地址, 从机拨码地址设置无效, 目前暂不支持掉电保存, 重新上电后, 从机默认地址还是拨码开关的地址。

主机发送指令：

字节	名称	说明
B0	广播地址	00h
B1	功能码	06h
B2	寄存器地址高	01h
B3	寄存器地址低	00h
B4	从机地址高	00h
B5	从机地址低	从机地址(1~255)
B6	CRC16	低
B7	CRC16	高

从机应答：

字节	名称	说明
B0	广播地址	00h
B1	功能码	06h
B2	字节数	02h
B3	从机地址高	00h
B4	从机地址低	从机地址(1~255)
B5	CRC16	低
B6	CRC16	高

示例：00 06 01 00 00 **address** CRCL CRCH

从机应答：00 06 02 00 **address** CRCL CRCH

若 **address**==2 则设置的从机地址为 2 此后，拨码地址无效。

主机得到从机回应后，认为强制设置地址成功，以后首字节按照 **address** 设置的地址，发送读数据命令。

#### 4、主机数据采样频率：

读取温湿度传感器数据时，上位机读取数据每次间隔时间不小于 500ms，推荐值 1s。

#### 5、功能码

03:读取数据

02:读取地址

#### 6、命令举例：

串口设置：异步通讯，起始位 1 位，数据位 8 位，无校验，停止位 1 位

数据传输速率缺省为：9600b/s

上位机发送 01 03 00 02 00 07 CRCL,CRCH （读取地址 01 从 00 02 开始的 7 个数据，从机不检查检验码）

INS01020406070914-S 返回：

0x01, 0x01, 0x07, CO2H, CO2L, 甲醛 H, 甲醛 L, TVOCH, TVOCL, PM2.5H, PM2.5L, PM10H, PM10L, 温度 H, 温度 L, 湿度 H, 湿度 L, CRCL, CRCH

例 1:

TX: 01 03 00 02 00 07 CRCL, CRCH

RX: 01 03 07 01 E2 00 05 00 24 00 2D 00 01 0F 03 11 CRCL, CRCH

CO2 浓度值= CO2H x 256 + CO2L PPM (BYTE3 x256 + BYTE4)

甲醛浓度值 = 甲醛 H x256 + 甲醛 L ug (BYTE5 x256 + BYTE6)

TVOC 浓度值 = TVOCH x256 + TVOCL ug (BYTE7 x256 + BYTE8)

PM2.5 浓度值= PM2.5H X256 + PM2.5L ug (BYTE9 x256 + BYTE10)

PM10 浓度值= PM10H X256 + PM10L ug (BYTE11 x256 + BYTE12)

温度= ((BYTE13)X256 + (BYTE14)) % 10

湿度= ((BYTE15)X256 + (BYTE16)) % 10

以上 CO2=482, 甲醛=5, TVOC=36, PM2.5=45, PM10=56, 温度=27.1, 湿度=78.5

例 2:

TX:01 03 00 0C 00 02 CRCL, CRCH (10:43:52:001) (读取地址01 从00 0C 开始的2个数据) 0x000C开始的两个数据, 对应温度和湿度.

RX:01 03 02 01 27 02 45 03 57 (10:43:52:159)

返回温度值为 0X0127 对应 10 进制 295, 表示温度 29.5℃

返回湿度值为 0X0245 对应 10 进制 581, 表示湿度 58.1%RH

**注意:**

从机返回的温湿度数据分别用两个字节表示, 高位在前, 低位在后

返回数据范围 -32768---32767, 实际温湿度数据需要将返回值除以 10

**例如:**

返回湿度 16 进制数据: 0X0311, 对应十进制 785, 表示湿度为 78.5%RH

返回温度 16 进制数据: 0X00FF, 对应十进制 255, 表示温度为 25.5℃

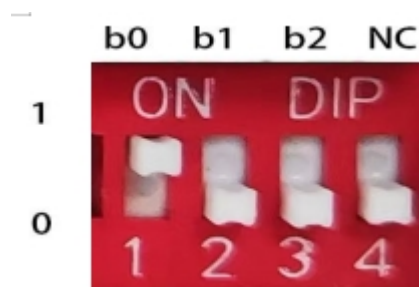
返回温度 16 进制数据: 0X8064, 最高位为 1 表示负数, 对应十进制-100, 表示温度为-10.0℃

读取地址值:

上位机发送: 00 02 00 00 00 01 CRCL, CRCH

变送器返回: 00 02 02 00, 地址 L, CRCL, CRCH

## 7、地址设置说明:



拨码开关共 4 位, 使用了 3 位, 第四位可以忽略, 单片机不读取。



第 1, 2, 3 脚, 分别对应 b0, b1, b2. 如上图所示, 向上拨码 (ON 方向), 对应的位 为高, 向下拨码为低。

地址码 = (b0 | b1<<1 | b2<<2) (**尽可能不要使用 00**) 00 是广播地址, 使用容易产生错误  
模块使用的地址码与发命令读取的地址码可能会有差异, 模块使用的地址码, 仅在上电时读取一次, 上电后, 如果改变地址, 模块的使用的地址码不会变化. 需重新上电才会变化;

发命令读取地址码是当前拨码开关的实时反应. 如果上电后, 没有改动地址, 模块使用的地址与发命令读取的地址是一致的, 如果发生改动, 发命令读取的时改动后的地址. 模块使用的是上电时的地址.

地址列表:

b2	b1	b0	ADDR
0	0	0	0x00
0	0	1	0x01
0	1	0	0x02
0	1	1	0x03
1	0	0	0x04
1	0	1	0x05
1	1	0	0x06
1	1	1	0x07

CRC 校验参考:

```
unsigned int GetCRC16(unsigned char *ptr, unsigned char len)
```

```
{
    unsigned int index;
    unsigned char crch = 0xFF; //高 CRC 字节
    unsigned char crcl = 0xFF; //低 CRC 字节
    unsigned char code TabH[] = { //CRC 高位字节值表
        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,
    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
    } ;
unsigned char code TabL[] = { //CRC 低位字节值表
    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
    } ;
while (len--) //计算指定长度的 CRC
{
    index = crch ^ *ptr++;
    crch = crcl ^ TabH[ index];
    crcl = TabL[ index];
}

```

```
return ((crch<<8) | crcl);  
}
```