

机器人本体运动控制板与工控机 232 通信接口自定义协议

修订时间	修订人	修订版本	描述规范
2022-10-25	郑爱和	V1.0	初版
2022-10-26	郑爱和	V1.1	1. 增加 crc 计算规范 2. 增加通信异常响
2023-12-14	杨崇军	V1.2	增加新协议

机器人本体运动控制板与工控机之间通信采用 232 接口通信，串口配置格式位：波特率 115200，数据位 8 位，无硬件校验，停止位 1 位。

通信协议采取自定义通信帧格式，包括帧头，帧类型，有效数据，有效数据长度，以及数据校验码部分，通信模式分为工控机主动下发，下位机响应和下位机主动上报两种模式。

一通信帧格式

帧头：(2 字节)	AA 55
帧类型：(1 字节)	Type：数据帧类型定义
有效数据长度（1 字节）	N
有效数据(N 字节)	D1……DN，高字节在前，低字节在后
校验码(2 字节)	数据类型，数据长度和有效数据的 CRC 冗余校验，高字节在前，低字节在后

```
typedef struct
{
```

```

uint8_t head_tag1;//0xAA
uint8_t head_tag2;//0x55
uint8_t frame_type;
uint8_t data_length;
uint8_t data[];
uint8_t crc_high_byte;
uint8_t crc_low_byte;
}

```

1.1 电机控制

1.1.1 电机使能_01

模式：工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x01

有效数据长度：1

数据定义：data0：

0：禁能

1：使能

```

typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x01
    uint8_t data_length;//0x01
    uint8_t motor_enable;//0:禁能，1：使能
    uint16_t crc_value;
}

```

2) 下位机响应

帧类型 ID: 0x01

有效数据长度：1

数据定义：数据 0

例： 工控机下发：AA 55 01 01 01 50 E0

下位机响应：AA 55 01 01 01 50 E0

1.1.2 速度模式控制_02(下发数据底盘不做任何处理)

模式： 工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x02

数据长度：5

数据定义：D0: 3 速度模式

D1-D2: 左轮电机转速，单位：r/min

D3-D4: 右轮电机转速，单位：r/min

typedef struct

```
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x02  
    uint8_t data_length;//0x05  
    uint8_t motor_run_mode;//3:速度模式  
    int16_t left_motor_spd;  
    int16_t right_mode_spd;  
    uint16_t crc_value;  
}
```

2) 下位机响应：

下位机返回上位机下发的原始数据

例：工控机下发： AA 55 02 05 03 00 64 00 64 A1 3C

下位机响应： AA 55 02 05 03 00 64 00 64 A1 3C

下发速度 0： AA 55 02 05 03 00 00 00 00 95 7C

1.1.3 力矩模式控制_03

备注：目前未使用

模式： 工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x03

数据长度：4

数据定义：D0: 4 力矩模式

D1: 目标力矩，百分比 0-100

D2-D3: 速度限制，单位：r/min

typedef struct

```
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x03  
    uint8_t data_length;//0x04  
    uint8_t motor_run_mode;//3:速度模式  
    uint8_t motor_aim_torque;;  
    uint16_t motor_limit_spd;;  
    uint16_t crc_value;  
}
```

2) 下位机响应：

下位机返回上位机下发的原始数据

例：工控机下发： AA 55 03 04 04 20 00 40 E2 F0

下位机响应： AA 55 03 04 04 20 00 40 E2 F0

1.2 充电控制指令_04

备注：第二版硬件不支持，暂时未使用，第三版硬件支持

模式： 工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x04

有效数据长度：1

数据定义：1：开启充电，0：关闭充电

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x04
    uint8_t data_length;//0x01
    uint8_t charge_reley_ctrl;
    uint16_t crc_value;
}
```

2) 下位机响应：

下位机返回帧类型数据原始数据

例：工控机下发： AA 55 04 01 01 51 F0

下位机响应： AA 55 04 01 01 51 F0

1.3 参数配置_05(暂时未使用)

模式： 工控机下发， 下位机响应

1) 工控机下发

帧类型 ID：0x05

数据长度：9

数据定义：D0-D1: 轮子间距，高位在前，低位在后，单位：mm

D2-D3: 轮子直径，高位在前，低位在后，单位：mm

D4: 超声波盲区设置，单位: mm

D5: 超声波避障距离，单位：mm

D6: 超声波模块使能，最多配置 8 路，bit0-bit7，默认使能 4 路 0x0F

D7: 超声波节点地址

D8: 风扇工作温度

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x05
    uint8_t data_length;//0x09
    uint16_t wheel_distance;
```

```

uint16_t wheel_diameter;

uint8_t ultrasonic_evadible_distance;

uint8_t ultrasonic_blind_distance;

uint8_t ultrasonic_mode_enable;

uint8_t config_ultrasonic_nodeid;

uint8_t temperature_limit;

uint16_t crc_value;
}

```

2) 下位机响应：

下位机返回帧类型数据原始数据

1.4 清零编码器数据_06

模式：工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x06

有效数据长度：1

数据定义：data0：

```

typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x06
    uint8_t data_length;//0x01
    uint8_t motor_enable;//0:
    uint16_t crc_value;
}

```

2) 下位机响应

帧类型 ID: 0x06

有效数据长度：1

数据定义：数据 0

例：工控机下发：AA 55 06 01 00 51 90

下位机响应：AA 55 06 01 00 51 90

1.5 使能急停和防撞条_07

模式： 工控机下发， 下位机响应

1) 工控机下发

帧类型 ID：0x07

有效数据长度：1

数据定义：1：使能， 0：失能

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x07
    uint8_t data_length;//0x01
    uint8_t enable_ctrl;
    uint16_t crc_value;
}
```

2) 下位机响应：

下位机返回帧类型数据原始数据

1.6 灯带控制_08

模式： 工控机下发， 下位机响应

1) 工控机下发

帧类型 ID：0x08

有效数据长度：1

数据定义：1：亮红灯， 充电中， 0：亮绿灯， 充电完成， 正常状态

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x08
    uint8_t data_length;//0x01
```

```

uint8_t RGB_ctrl;

uint16_t crc_value;
}

```

2) 下位机响应：

下位机返回帧类型数据原始数据

1.7 使能 IMU 数据上报_09(新增协议)

模式：工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x09

有效数据长度：1

数据定义：data0：

0：禁能

1：使能

typedef struct

```

{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x09
    uint8_t data_length;//0x01
    uint8_t motor_enable;//0:禁能， 1：使能
    uint16_t crc_value;
}

```

2) 下位机响应

帧类型 ID: 0x01

有效数据长度：1

数据定义：数据 0

例： 工控机下发：AA 55 09 01 01 92 61

下位机响应：AA 55 09 01 01 92 61

1.8 固件版本号读取_10

模式：工控机下发查询指令，下位机响应

1) 工控机下发

帧类型 ID：0x10

数据长度：01

数据定义：D0：00

typedef struct

```
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x10  
    uint8_t date_length;//0x01  
    uint8_t data0;//0x00  
    uint16_t crc_value;  
}
```

2) 下位机响应：

帧类型 ID: 0x10

数据长度：06

有效数据：D0:主版本号

D1:次版本号

D2:修订版本号

D3:版本修改年，时间年的末尾 2 位，例如 2022 是 22

D4:版本修改月

D5:版本修改日

typedef struct

```
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x10  
    uint8_t date_length;//0x06  
    uint8_t firmversion1;
```

```

uint8_t firmversion2;
uint8_t firmversion3;
uint8_t firmversion4;
uint8_t firmversion5;
uint8_t firmversion6;
uint16_t crc_value;
}

```

例：工控机下发：AA 55 10 01 00 95 71

下位机响应：AA 55 10 06 01 01 01 16 0A 19 44 3C

//返回版本号 01.01.01.22.10.25

1.9 查询状态数据_11

模式：工控机下发查询指令，下位机响应

1) 工控机下发

帧类型 ID：0x11

数据长度：01

数据定义：D0：00

typedef struct

```

{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x11
    uint8_t data_length;//0x01
    uint8_t data;//0x00;
    uint16_t crc_value;
}

```

2) 下位机响应：

帧类型 ID: 0x11

数据长度：17

有效数据：

D0:温度数据，单位：摄氏度

D1:湿度数据，0-100

D2:设备状态数据：00
D3:前防撞条
D4:后防撞条
D5:急停信号
D6:开关机按键
D7:电池充放电状态：0-放电，1-充电
D8:电池电压数据：0-65535(10mV)
D9:电池电流数据：0-65535(10mA)
D10:电池电量百分比 SOC：0-100(%)
D11:电池健康系数 SOH：0-100(%)
D12:电机使能状态，1：使能，0：失能
D13: IMU 故障，0 代表无故障，1 代表故障
D14：驱动器故障，0 代表无故障，1 代表故障
D15：驱动器具体故障信息(uint8_t)

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x11
    uint8_t data_length;//0x12
    uint8_t temprature;
    uint8_t humidity;
    uint8_t status1;
    uint8_t input_status;
    .....
    .....
    uint16_t crc_value;
}
```

例：工控机下发：AA 55 11 01 00 55 20

下位机响应：AA 55 11 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 22 66

5.1 驱动器故障代码及解决方法

故障代码	释义	故障原因	解决方法
01	编码器 ABZ 报警	a. 编码器接线有误 b. 编码器损坏 c. 噪声干扰严重	a. 确认接线可靠、无误 b. 返厂检修 c. 远离大电流配线
02	编码器 UVW 报警	a. 编码器接线有误 b. 编码器损坏 c. 噪声干扰严重	a. 确认接线可靠、无误 b. 返厂检修 c. 远离大电流配线
03	位置超差	a. 位置指令频率过高 b. 位置环增益过小 c. 位置超差设定值过小 d. 电机或编码器接线错误 e. 电机转矩不足或负载过大	a. 调整输入指令脉冲频率 b. 重新调整相关参数 c. 重新调整位置超差设定值 d. 检查接线 e. 增大电机功率或减小负载
04	失速	电机转速过高	减小速度指令
05	ADC 零点异常	电机电流反馈通道异常	返厂检修
06	过载	a. 负载过大 b. 电机震荡 c. 机械制动器未释放 d. 电机及编码器接线错误	a. 更换大功率驱动器和电机 b. 重新调整增益 c. 检查机械制动器 d. 检查电机及编码器接线
07	功率电源欠压	a. 功率电源电压过低 b. 驱动器电压测量回路故障	a. 检查供电电压 b. 返厂检修
08	功率电源过压	a. 功率电源电压过高 b. 驱动器电压测量回路故障 c. 电机带载启停频繁	a. 检查供电电压 b. 返厂检修 c. 加装泄放模块
09	过流	a. 驱动器损坏 b. 电机动力线 UVW 短路 c. 电机损坏 d. 功率模块自动保护 e. 伺服电机剧烈震荡	a. 更换驱动器 b. 检查电机接线 c. 更换电机 d. 重新上电 e. 调整伺服增益参数

故障代码	释义	故障原因	解决方法
0A	瞬时放电报警	瞬时放电功率过大	a. 检查电源网压 b. 更换驱动器 c. 加装泄放模块
0B	平均放电报警	平均放电功率过大	a. 选择合适的再生放电电阻 b. 更换驱动器 c. 加装泄放模块
0C	参数读写异常	驱动器配置参数读写异常	返厂检修
0D	输入端口重复定义	输入端口功能定义有重复	重新调整输入端口功能定义
0E	断线保护	通讯故障	检查通讯线路
0F	电机温度报警	电机温度过高	检查线路、负载
10	电机温控断线报警	电机 NTC 未接	检查 NTC 接线
11	协同模式报警	-	-
12	驱动器温度报警	驱动器温度超过最大使用温度	a. 检查散热条件 b. 降额使用

2.0 避障数据上报_20(变更协议内容，针对电应普超声模块， 导向超声模块采用原来协议)

导向超声模块协议

模式：下位机主动上报，上位机不响应

上报周期：100ms

1) 下位机上报：

帧类型：0x20

数据长度：08

有效数据：

D0-D1: 前左超声波数据，单位:mm

D2-D3: 前中超声波数据，单位:mm

D4-D5: 前右超声波数据，单位:mm

D6-D7: 后超声波数据，单位：mm

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x20
    uint8_t data_length;//0x08
    uint16_t ultrasonic1_diatance;
    uint16_t ultrasonic2_diatance;
    uint16_t ultrasonic3_diatance;
    uint16_t ultrasonic4_diatance;
    uint16_t crc_value;
}
```

电应普超声模块协议

模式： 下位机主动上报，上位机不响应

上报周期：100ms

1) 下位机上报：

帧类型：0x20

帧 ID：

数据长度：02

有效数据：

D0-D1: 超声波数据，单位:mm

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x20
    uint8_t data_length;//0x03
    uint8_t frame_id;//1 代表后超声模块，2 代表左前超声模块，3 代表右前超声模块
    uint16_t ultrasonic_diatance;
    uint16_t crc_value;
}
```

2.1 运动数据上报_21(变更协议内容，上报数据底盘控制板不做任何处理)

模式：下位机主动上报，上位机不响应

上报周期：20ms

2) 下位机上报：

帧类型：0x21

帧 id:

数据长度：12

有效数据：

D0-D1: 左轮电机转速，单位:r/min

D2-D3: 右轮电机转速，单位:r/min

D4-D8: 左轮编码器值，编码器累加值

D9-D12: 右轮编码器值，编码器累加值

typedef struct

```
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x21
    uint8_t data_length;//0x0D
    uint8_t frame_id; //1 代表数据有效，0 代表数据无效
    int16_t left_motor_speed;
    int16_t right_motor_speed;
    int32_t left_motor_encoder;
    int32_t right_motor_encoder;
    uint16_t crc_value;
}
```

2.2 电池故障信息上报_22

模式：主动触发上报故障信息，上报周期 500ms

帧类型：0x22

数据长度：0x04

有效数据：

- data0：电池系统状态
- data1：电池告警标志
- data2：电池保护标志
- data3：电池故障状态
- data4: IMU 故障，0 代表无故障，1 代表故障
- data5：驱动器故障，0 代表无故障，1 代表故障

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x22
    uint8_t date_length;//0x0A
    uint16_t data0;
    uint16_t data1;
    uint16_t data2;
    uint16_t data3;
    uint8_t data4;
    uint8_t data5;
    uint16_t crc_value;
}
```

1 告警标志（1：发生告警，0：未发生告警）

bit0	单体过压告警	bit8	充电高温告警
bit1	单体欠压告警	bit9	放电高温告警
bit2	整组过压告警	bit10	充电低温告警
bit3	整组欠压告警	bit11	放电低温告警
bit4	充电过流告警	bit12	环境高温告警
bit5	放电过流告警	bit13	环境低温告警
bit6	预留	bit14	MOSFET 高温告警
bit7	预留	bit15	SOC 过低告警

表 2 保护标志（1：发生保护，0：未发生保护）

bit0	单体过压保护		bit8	充电高温保护
bit1	单体欠压保护		bit9	放电高温保护
bit2	整组过压保护		bit10	充电低温保护
bit3	整组欠压保护		bit11	放电低温保护
bit4	充电过流保护		bit12	环境高温保护
bit5	放电过流保护		bit13	环境低温保护
bit6	短路保护		bit14	MOSFET 高温保护
bit7	预留		bit15	预留

表 3（故障状态）

bit0	充电时 MOSFET 故障		bit8	预留
bit1	放电时 MOSFET 故障		bit9	预留
bit2	温度传感器 NTC 故障		bit10	预留
bit3	预充 MOSFET 故障		bit11	预留
bit4	电芯故障		bit12	预留
bit5	BMS 芯片故障		bit13	预留
bit6	预留		bit14	预留
bit7	预留		bit15	加热功能故障

表 4（系统状态）

bit0	预留		bit8	预留
bit1	充电 MOSFET 状态		bit9	预留
bit2	放电 MOSFET 状态		bit10	充满状态
bit3	预留		bit11	待机状态
bit4	预留		bit12	预留
bit5	预留		bit13	预留
bit6	预留		bit14	预留
bit7	加热启动状态		bit15	预留

2.3 IMU 数据上报_23(第二版机器使用)

模式：主动触发上报信息，上报周期 10ms

帧类型：0x23

typedef struct

```
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x23  
    uint8_t data_length;//0x5C  
    uint8_t data0;  
    uint8_t data1;  
    uint8_t data2;  
    uint8_t data3;  
    .....  
    .....  
    uint16_t crc_value;  
}
```

IMU 数据描述参考，实际内容取决于 IMU 配置

描述	接收到的数据	原始数据	转换的数据	乘法因子	真实数据	单位
包头	3A					
Sensor ID	01 00	0001	1			
指令号	09 00	0009	9			
数据长度	5C 00	005C	92			
时间戳	C6 A7 00 00	0000A7C6	42950	0.002	85.9	seconds
原始的 加速度计数据 X	00 00 3C BC	BC3C0000	-0.01147161	1	-0.01147161	g
原始的 加速度计数据 Y	00 00 80 BC	BC800000	-0.015625	1	-0.015625	g
原始的 加速度计数据 Z	00 88 80 BF	BF808800	-1.00415	1	-1.00415	g
校准后的 加速度计数据 X	E5 39 2A BC	BC2A39E5	-0.01038978	1	-0.01038978	g
校准后的 加速度计数据 Y	76 14 64 3B	3B641476	0.003480223	1	0.003480223	g
校准后的 加速度计数据 Z	96 F2 7E BF	BF7EF296	-0.9958891	1	-0.9958891	g
原始的 陀螺仪数据 X	29 5C 0F 3E	3E0F5C29	0.14	1	0.14	dps
原始的 陀螺仪数据 Y	33 33 33 3F	3F333333	0.7	1	0.7	dps
原始的 陀螺仪数据 Z	3E 0A D7 BE	BED70A3E	-0.42	1	-0.42	dps
静止偏差校准后的 陀螺仪数据 X	29 5C 0F 3E	3E0F5C29	0.14	1	0.14	dps
静止偏差校准后的 陀螺仪数据 Y	33 33 33 3F	3F333333	0.7	1	0.7	dps
静止偏差校准后的 陀螺仪数据 Z	3E 0A D7 BE	BED70A3E	-0.42	1	-0.42	dps
坐标轴校准后的 陀螺仪数据 X	24 63 18 3E	3E186324	0.1488157	1	0.1488157	dps
坐标轴校准后的 陀螺仪数据 Y	6A AD 32 3F	3F32AD6A	0.6979586	1	0.6979586	dps
坐标轴校准后的 陀螺仪数据 Z	64 A0 D2 BE	BED2A064	-0.4113799	1	-0.4113799	dps
四元数 w	AE 6E 7B 3F	3F7B6EAE	0.9821576	1	0.9821576	-
四元数 x	35 A5 6A BC	BC6AA535	-0.01432161	1	-0.01432161	-
四元数 y	06 79 47 BD	BD477906	-0.0486994	1	-0.0486994	-
四元数 z	ED 6C 39 3E	3E396CED	0.1810796	1	0.1810796	-
欧拉角数据 X	99 BB 1A 3F	3F1ABB99	0.604425	1	0.604425	degree
欧拉角数据 Y	2C 37 B9 40	40B9372C	5.787985	1	5.787985	degree
欧拉角数据 Z	55 E5 A6 C1	C1A6E555	-20.86798	1	-20.86798	degree
LRC 校验和*	9A 24	249A				
包尾	0D 0A					

2.4 红外数据上报_24(硬件不支持了)

模式：下位机主动上报，上位机不响应

上报周期：50ms

3) 下位机上报：

帧类型：0x24

数据长度：2

有效数据：单位：cm

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x24
    uint8_t data_length;//0x02
    int8_t distance_high_bit;
    int8_t distance_low_bit;
    uint16_t crc_value;
}
```

2.5 工控机电源控制_25

模式：工控机下发，下位机响应

1) 工控机下发

帧类型 ID：0x25

有效数据长度：1

数据定义：data0：1：断电

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0x25
    uint8_t data_length;//0x01
```

```
uint8_t ipc_power;//1：断电  
uint16_t crc_value;  
}
```

备注：当嵌入式收到该指令之后，会断工控机电源 5s，5s 之后会重新上电

2.6 异常响应帧(暂时不使用)

下位机响应异常数据帧

下位机回复：

帧类型：0x80

数据长度：0x02

有效数据：data0: 错误码 1

data1：错误码 2

```
typedef struct  
{  
    uint8_t head_tag1;//0xAA  
    uint8_t head_tag2;//0x55  
    uint8_t frame_type;//0x80  
    uint8_t data_length;//0x02  
    uint8_t error_code1;  
    uint8_t error_code2;  
    uint16_t crc_value;  
}
```

2.7 心跳帧_A0

下位机上报，上位机响应，上报周期 100ms，上位机响应原路返回

下位机上报：

帧类型：0xA0

数据长度：0x01

有效数据：data0:0x01

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0xA0
    uint8_t data_length;//0x01
    uint8_t data;//0x01
    uint16_t crc_value;

}
```

上位机响应：

帧类型：0x80

数据长度：0xA0

有效数据：data0: 0x01

```
typedef struct
{
    uint8_t head_tag1;//0xAA
    uint8_t head_tag2;//0x55
    uint8_t frame_type;//0xA0
    uint8_t data_length;//0x01
    uint8_t data;//0x01;
    uint16_t crc_value;

}
```

2.8 CRC 校验码计算

采用 modbus crc-16 冗余校验方式

```
uint16_t Asp_Get_CRC16(volatile uint8_t *ptr,uint32_t len)
{
    uint8_t i;
```

```

uint16_t crc = 0xFFFF;

if(len==0)

{
    len = 1;
}

while(len--)

{
    crc ^= *ptr;
    for(i=0; i<8; i++)
    {
        if(crc&1)
        {
            crc >>= 1;
            crc ^= 0xA001;
        }
        else
        {
            crc >>= 1;
        }
    }
    ptr++;
}

return(crc);
}

```

