
RR508 485 LT 通信协议

David Zhao

zhaoming@rfdcn.com

上海锐帆信息科技有限公司

知识产权说明

本协议为我公司（上海锐帆信息科技有限公司）的知识产权，经授权后，可由被授权者使用。但不得将其泄露给任何未经授权的第三方，否则，可能引起的法律责任和后果由泄密者承担。

目录

RR508 485 LT 通信协议.....	1
知识产权说明.....	1
串口设置.....	3
通信包结构.....	3
下行 (PC to Device):	3
上行 (Device to PC):	3
通信命令.....	4
设置增益.....	4
获得增益.....	4
读取标签.....	4
注意事项.....	4
附: CRC7 定义—C 语言示例	6

串口设置

Baud rate = firmware dependent.
Data bits = 8
Parity = None
Stop bits = 1

通信包结构

上下行通信是相同的包结构

下行（PC to Device）:

数据（2 进制）	偏移	长度（字节）	含义	备注
02H	0	1	包起始标志	STX
0XXH	1	1	设备地址	ADDR
0XXH	2	1	命令字	CMD
0XXH	3	1	参数长度	LTH，可用值为 0~14H
PARAM	4	N	参数	PARAM，N 必须等于参数长度。 如果 N=0，则此字段不存在
0XXH	4+N	1	CRC7 校验	CRC
03H	5+N	1	包结束标志	ETX

上行（Device to PC）:

数据（2 进制）	偏移	长度（字节）	含义	备注
02H	0	1	包起始标志	STX
0XXH	1	1	设备地址	ADDR
0XXH	2	1	状态字	STT
0XXH	3	1	参数长度	LTH，可用值为 0~14H
PARAM	4	N	参数	PARAM，N 必须等于参数长度
0XXH	4+N	1	CRC7 校验	CRC
03H	5+N	1	包结束标志	ETX

Note:

CRC 为从 ADDR 到 PARAM 结束的所有 N+3 个字节的 CRC7 校验。

CRC7 的生成多项式为 $G(x) = x^7 + x^3 + 1$

通信命令

设置增益

下行:

CMD = 11H, LTH = 1, PARAM = 增益

上行:

STT = 0, LTH = 0, PARAM 不存在

获得增益

下行:

CMD = 12H, LTH = 0, PARAM 不存在

上行:

STT = 0, LTH = 1, PARAM = 增益

读取标签

下行:

CMD = 02H, LTH = 0, PARAM 不存在

上行:

若干个数据包（数量由固件配置决定）

包结构（包含一个标签信息）:

STT = 0, LTH = 4, PARAM = TAG_ID_B0, TAG_ID_B1, TAG_ID_B2,
TAG_ID_B3 (Big Endian)

其中, TAG_ID 如果为 0, 则表示没有标签。

注意事项

- I) 所有的通信由上位机主动。
- II) 因为 485 为半双工协议, 所以注意时序。

-
- III) 由于线路原因可能有丢包现象，所以命令可以重试。
 - IV) 建议尽量使用 115200 的波特率，这样比较容易保证带宽。
 - V) 任何未尽事宜，请 Email 至: zhaoming@rfidcn.com



附：CRC7 定义—C 语言示例

```
/*
   CRC7 - Used in SD Card.
   Optimized version
   By David Zhao
   zhaoming@rfdcn.com
*/

#ifndef BYTE
#define BYTE unsigned char
#endif //BYTE

const BYTE gucCrcTable[256] =
{
    0x00, 0x12, 0x24, 0x36, 0x48, 0x5a, 0x6c, 0x7e,
    0x90, 0x82, 0xb4, 0xa6, 0xd8, 0xca, 0xfc, 0xee,
    0x32, 0x20, 0x16, 0x04, 0x7a, 0x68, 0x5e, 0x4c,
    0xa2, 0xb0, 0x86, 0x94, 0xea, 0xf8, 0xce, 0xdc,
    0x64, 0x76, 0x40, 0x52, 0x2c, 0x3e, 0x08, 0x1a,
    0xf4, 0xe6, 0xd0, 0xc2, 0xbc, 0xae, 0x98, 0x8a,
    0x56, 0x44, 0x72, 0x60, 0x1e, 0x0c, 0x3a, 0x28,
    0xc6, 0xd4, 0xe2, 0xf0, 0x8e, 0x9c, 0xaa, 0xb8,
    0xc8, 0xda, 0xec, 0xfe, 0x80, 0x92, 0xa4, 0xb6,
    0x58, 0x4a, 0x7c, 0x6e, 0x10, 0x02, 0x34, 0x26,
    0xfa, 0xe8, 0xde, 0xcc, 0xb2, 0xa0, 0x96, 0x84,
    0x6a, 0x78, 0x4e, 0x5c, 0x22, 0x30, 0x06, 0x14,
    0xac, 0xbe, 0x88, 0x9a, 0xe4, 0xf6, 0xc0, 0xd2,
    0x3c, 0x2e, 0x18, 0x0a, 0x74, 0x66, 0x50, 0x42,
    0x9e, 0x8c, 0xba, 0xa8, 0xd6, 0xc4, 0xf2, 0xe0,
    0x0e, 0x1c, 0x2a, 0x38, 0x46, 0x54, 0x62, 0x70,
    0x82, 0x90, 0xa6, 0xb4, 0xca, 0xd8, 0xee, 0xfc,
    0x12, 0x00, 0x36, 0x24, 0x5a, 0x48, 0x7e, 0x6c,
    0xb0, 0xa2, 0x94, 0x86, 0xf8, 0xea, 0xdc, 0xce,
    0x20, 0x32, 0x04, 0x16, 0x68, 0x7a, 0x4c, 0x5e,
    0xe6, 0xf4, 0xc2, 0xd0, 0xae, 0xbc, 0x8a, 0x98,
    0x76, 0x64, 0x52, 0x40, 0x3e, 0x2c, 0x1a, 0x08,
    0xd4, 0xc6, 0xf0, 0xe2, 0x9c, 0x8e, 0xb8, 0xaa,
    0x44, 0x56, 0x60, 0x72, 0x0c, 0x1e, 0x28, 0x3a,
    0x4a, 0x58, 0x6e, 0x7c, 0x02, 0x10, 0x26, 0x34,
    0xda, 0xc8, 0xfe, 0xec, 0x92, 0x80, 0xb6, 0xa4,
    0x78, 0x6a, 0x5c, 0x4e, 0x30, 0x22, 0x14, 0x06,
    0xe8, 0xfa, 0xcc, 0xde, 0xa0, 0xb2, 0x84, 0x96,
    0x2e, 0x3c, 0x0a, 0x18, 0x66, 0x74, 0x42, 0x50,
    0xbe, 0xac, 0x9a, 0x88, 0xf6, 0xe4, 0xd2, 0xc0,
    0x1c, 0x0e, 0x38, 0x2a, 0x54, 0x46, 0x70, 0x62,
    0x8c, 0x9e, 0xa8, 0xba, 0xc4, 0xd6, 0xe0, 0xf2
};
```

```
BYTE    gucCrc7;
#define SdCrcIni()    gucCrc7=0
#define SdDoCrc(ucX)    gucCrc7=gucCrcTable[    gucCrc7^(ucX)]
#define SdFinishCrc()    gucCrc7 |= 0x01
#define SdGetCrc()    gucCrc7
```

```
BYTE SdBuildCrc(const BYTE * pucData, BYTE ucLth)
```

```
{
    SdCrcIni();
    while(ucLth--)
    {
        SdDoCrc(pucData[0]);
        pucData++;
    }
    SdFinishCrc();
    return SdGetCrc();
}
```