SKP 系列模组MODBUS 协议

森库莱萨（深圳）

目录

1 串口通讯说明

1.1 协议出厂默认参数

● 通讯接口： RS-485。

● 通讯格式： 1 个起始位，8 个数据位，无校验，1 个停止位。

● 波特率： 9600

● 地址： 1（可软件更改） 。

● 通讯方式： 监控主机与本装置采用一对一（或一对多）主从查询方式

● 数据协议： MODBUS-RTU

1.2 MODBUS-RTU 协议说明

数据格式：标准的 MODBUS 协议 RTU 方式，16 进制编码，若数字不是一个字节，则高字节在前低字节在后，主从式传输，每一帧有固定的格式，包括几个字节的数据，主机发送帧每个字节的意义如下：

从机地址（1Byte）	功能码（1Byte）	寄存器地址（2Byte）	寄存器数目（2Byte）	数据区（nByte）	CRC 校验（2Byte）
addr	Function	regH regL	numH numL	d0-dn	

***注：若不使用标准 MODBUS-RTU 协议可以忽略校验**

若需要返回数据，从机返回帧每个字节的意义如下：

从机地址（1Byte）	功能码（1Byte）	数据长度（1Byte）	数据区（nByte）	CRC 校验（2Byte）
addr	Function	len	d0-dn	crcH crcL

***注：若不使用标准 MODBUS-RTU 协议可以忽略校验**

若无需返回数据，从机返回帧每个字节的意义如下：

从机地址（1Byte）	功能码（1Byte）	寄存器地址（2Byte）	寄存器数目（2Byte）	CRC 校验（2Byte）
addr	Function	regH regL	numH numL	crcH crcL

***注：若不使用标准 MODBUS-RTU 协议可以忽略校验**

●从机地址 要询问传感器的地址。每个传感器可以有 1-247 个地址，若主机询问的地址为 0，则为广播式，每个从机都要响应命令，但是无需回复报文。 （本版本中无广播功能）

●命令 用来指示从机要执行的功能。

●寄存器地址 对应命令下需要访问的寄存器的起始地址，有两个字节，高字节在前。不同命令下可以有相同的地址，但并不是指向同样的寄存器。

●寄存器数目 对应命令下需要访问的寄存器的数目，有两个字节，高字节在前。若数目为 0，则填充 0x00 0x00。

●数据区 每个命令下数据的意义不同，参考下面各条命令的详细解释。若无数据，则此处没有，无需填充 0。

●数据长度 数据区的字节数。

●CRC 校验 用来校验一帧数据的正确性，防止传输过程中出错。CRC 校验有两个字节，高字节在前，所校验的字节包括本帧 CRC 前面所有的字节，CRC 校验字产生的算法参考后面附录。 3

1.3 MODBUS-RTU 寄存器说明

系统寄存器有如下定义（红色字体为默认值）：

寄存器名称	地址	类型	说明(十六进制)

模块地址寄存器	0x00000	W 写	模块地址寄存器 模块地址,取值范围 1-255 默认值：1
距离寄存器	0x40000	R 读	对应 MODBUS 的保持寄存器，存储着当前的测量距离，单位毫米（mm）

1.4 MODBUS-RTU 寄存操作说明：

对于本测距雷达模块，MODBUS-RTU 协议只涉及到两个操作

1.读保持寄存器（功能码 0x03）

2.写寄存器（功能码 0x06）

注：寄存器地址说明

40000 —— 距离信息

40001 —— status 状态信息

40002 —— ID 信息

40003 —— 波特率信息

例如，如果模块地址是 1，波特率是 9600，无奇偶校验，数据位是 8 位我们如果想全部读出这些寄存器，做如下操作：

主机发送： 01 03 00 00 00 04 44 09

从机响应： 01 03 08 07 3C 00 00 00 01 00 06 F9 F0

其中距离值为 07 3C，转换成十进制为 1852，即 1.852 米。其中 status 状态为 00 00，转换为十进制为 0，即工作正常。其中 ID 信息为 00 01，转换为十进制为 1，即从站地址为 1。

其中波特率信息为 00 06，转换为十进制为 6，即波特率为 9600（请参考数据手册修改波特率章节）。

请求（主机发送）十六进制		响应（从机发送）十六进制	
从机地址	01	从机地址	01
功能码	03	功能码	03
寄存器地址（Hi）	00	数据长度	08
寄存器地址（Lo）	00	40000 数据（Hi）	07
数量（Hi）	00	40000 数据（Lo）	3C
数量（Lo）	04	40001 数据（Hi）	00
校验码（Hi）	44	40001 数据（Lo）	00
检验码（Lo）	09	40002 数据（Hi）	00
		40002 数据（Lo）	01
		40003 数据（Hi）	00
		40003 数据（Lo）	06
		校验码（Hi）	F9
		检验码（Lo）	F0

持寄存器（设置系统的参数,以及控制系统）（0x06）

寄存器说明已经在上面提到，特别注意的是，设置完毕后系统重新上电。

例如，当前模块的地址是 01，我们想设置到 02

主机发送： 01 06 00 02 00 02 A9 CB

从机响应： 01 06 00 00 00 02 A9 CB（响应完成后即可完成设置）

请求（主机发送）十六进制		响应（从机发送）十六进制	
从机地址	01	从机地址	01

功能码	06	功能码	06
寄存器地址（Hi）	00	寄存器地址（Hi）	00
寄存器地址（Lo）	02	寄存器地址（Lo）	02
40002 数据（Hi）	00	40002 数据（Hi）	00
40002 数据（Lo）	02	40002 数据（Lo）	02
校验码（Hi）	A9	校验码（Hi）	A9
检验码（Lo）	CB	检验码（Lo）	CB

附录 1.MODBUS-RTU 中 CRC 校验 C 程序

CRC 简单函数如下：

```
unsignedchar*puchMsg;/*要进行CRC 校验的消息*/

unsignedshortusDataLen;/*消息中字节数*/

unsignedshortCRC16(puchMsg,usDataLen)
{
unsignedcharuchCRCHi=0xFF;/*高 CRC 字节初始化*/ unsignedcharuchCRCLo=0xFF;/*低
CRC 字节初始化*/ unsigneduIndex;/*CRC 循环中的索引*/ while(usDataLen--)/*传输消息缓
冲区*/
{ uIndex=uchCRCHi^*puchMsg++;/* 计 算 CRC*/
uchCRCHi=uchCRCLo^auchCRCHi[uIndex]; uchCRCLo=auchCRCLo[uIndex];
} return((uchCRCHi<<8)|uchCRCLo);
}
/*CRC 高 位 字 节 值 表 */
staticunsignedcharauchCRCHi[]={ 0x00,0xC1,0x81,0x40,0x
01,0xC0,0x80,0x41,0x01,0xC0,
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
};
//CRC 低位字节值表
staticcharauchCRCLo[]={
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
};
```