

<	2018年5月						>
日	一	二	三	四	五	六	
29	30	1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	

搜索

找找看

我的标签

Modbus 物联网 软件开发 编程 C/C++(2)

日语 汉字 假名 转换(1)

随笔档案(11)

2017年3月 (6)

2009年3月 (1)

2008年10月 (2)

2008年9月 (2)

最新评论

1. Re:推荐一本书：清华出版的《Modbus软件开发实战指南》

楼主您好，我最近正在学习Modbus通讯，要是方便的话您能把这本书的电子版发我一份吗，我的邮箱是：1048262574@qq.com。谢谢！

-- 专注一事

2. Re:推荐一本书：清华出版的《Modbus软件开发实战指南》

代码都是开源的，书本上提供了下载方法。

最好学编程是跟着敲一遍代码，从头到尾学习，印象深刻，完成后，真正变成自己的知识，成为精通Modbus的大师水平。

-- Magic2008

3. Re:推荐一本书：清华出版的《Modbus软件开发实战指南》

@牛腩谢谢支持...

-- Magic2008

4. Re:推荐一本书：清华出版的《Modbus软件开发实战指南》

支持支持

-- 牛腩

5. Re:第一弹:发布原创日语汉字转假名小工具

需要密码谢谢楼主。

34643200@qq.com

-- liuliu159

阅读排行榜

1. 第二弹:升级原创日语汉字转假名小工具1.02(3176)

Modbus-RTU

一、数据分析

两个设备（单片机）通讯，用的是Modbus协议。

在单片机中拿出一部分内存（RAM）进行两个设备通讯，例如：

INT8U	OX[20];	// 定义8位的数组变量。	输出线圈	功能码：0x01,0x05,0x0F	地址：0x
INT8U	IX[20];	// 定义8位的数组变量。	输入线圈	功能码：0x02	地址：1x
INT16U	HoldDataReg[30];	// 定义16位的数组变量。	保持寄存器	功能码：0x03,0x06,0x10	地址：4x
INT16U	InDataReg[30];	// 定义16位的数组变量。	输入寄存器	功能码：0x04	地址：3x

说明：

OX[20] 代表是输出线圈，用功能码 0x01, 0x05, 0x0F 访问， 开头地址是 0 （这个后续说明）

IX[20] 代表是输入线圈，用功能码 0x02 访问，开头地址是 1 （这个后续说明）

另外两个一样的道理。

注意：所谓的“线圈”“寄存器”就是“位变量”“16位变量”。之所以称“线圈”因为Modbus最初是施耐德公司为其PLC制定的一种通讯协议很自然Modbus协议中的很多术语是和PLC中的术语相关的。

二、报文分析

简单分析一条Modbus-RTU报文，例如： 01 06 00 01 00 17 98 04

01	06	00 01	00 17	98 04
从机地址	功能号	数据地址	数据	CRC校验

这一串数据的意思是：把数据 0x0017(十进制23) 写入 1号从机地址 0x0001数据地址。

一个报文就是一帧数据，一个数据帧就一个报文： 指的是一串完整的指令数据，就像上面的一串数据。

98 04是它前面的数据（01 06 00 01 00 17）通过一算法（见附录2）计算出来的结果，其实就像是计算累加和那样（就是010600010017加起来的值，它的算法就是加法运算）

1、主机对从机写操作

如果单片机接收到一个报文那么就对报文进行解析执行相应的处理，如上面报文：

01	06	00 01	00 17	98 04
从机地址	功能号	数据地址	数据	CRC校验

假如本机地址是 1，那么单片机接收到这串数据根据数据计算CRC校验判断数据是否正确，如果判断数据无误，则结果是：

HoldDataReg[1] = 0x0017;

Modbus主机就完成了一次对从机数据的写操作，实现了通讯。

2、主机对从机读操作

主机进行读HoldDataReg[1] 操作，则报文是：

01	03	00 01	00 01	D5 CA
从机地址	功能号	数据地址	读取数据个数	CRC校验

那么单片机接收到这串数据根据数据计算CRC校验判断数据是否正确，如果判断数据无误，则结果是：返回信息给主机，返回的信息也是有格式的：

返回内容：

01	03	02	0017	F8 4A
----	----	----	------	-------

- 2. 第一弹:发布原创日语汉字转假名小工具 (3079)
- 3. 推荐一本书: 清华出版的《Modbus软件开发实战指南》(2152)
- 4. Modbus软件开发实战指南 之 开发自己的Modbus Poll工具 - 2(1798)
- 5. Modbus软件开发实战指南 之 开发自己的Modbus Poll工具 - 3(1547)

评论排行榜

- 1. 第一弹:发布原创日语汉字转假名小工具 (85)
- 2. 推荐一本书: 清华出版的《Modbus软件开发实战指南》(4)
- 3. 善用佳软推荐几个很好用的软件以备使用(1)

推荐排行榜

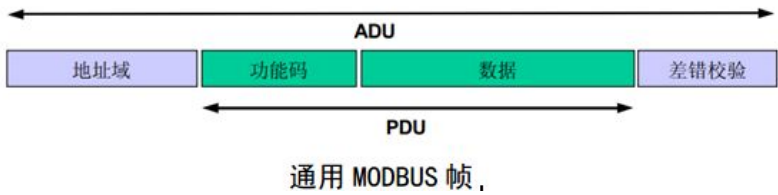
- 1. Modbus软件开发实战指南 之 开发自己的Modbus Poll工具 - 3(1)

从机地址 功能号 数据字节个数 两个字节数据 CRC校验

Modbus主机就完成了一次对从机数据的读操作，实现了通讯。

三、Modbus报文模型

以上了解到了Modbus的一帧报文是如何通讯的，其实每个报文的格式都基本一样的。

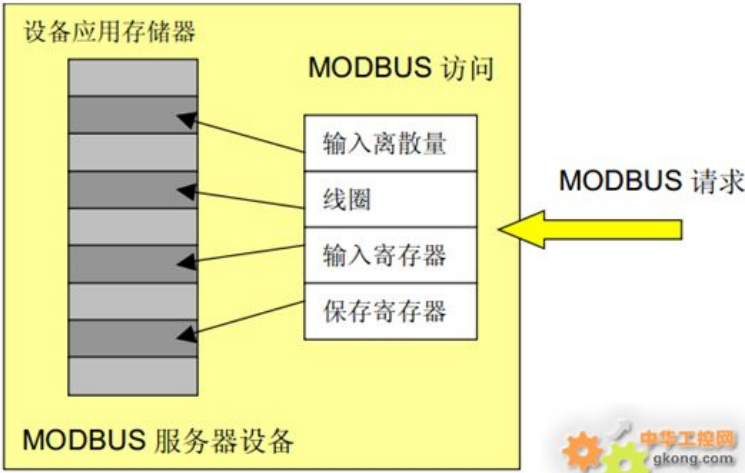


ADU： 应用数据单元
PDU： 协议数据单元

四、Modbus数据模型

MODBUS 以一系列具有不同特征表格上的数据模型为基础。四个基本表格为：

基本表格	对象类型	访问类型	内容
离散量输入	单个比特	只读	I/O 系统提供这种类型数据
线圈	单个比特	读写	通过应用程序改变这种类型数据
输入寄存器	16-比特字	只读	I/O 系统提供这种类型数据
保持寄存器	16-比特字	读写	通过应用程序改变这种类型数据



五、Modbus事务处理

下列状态图描述了在服务器侧Modbus事务处理的一般处理过程。



六、Modbus请求与响应

看Modbus协议手册，中文第 10 页开始，英文第 24 页开始。手册非常详细举例说明了Modbus协议各个功能号的请求与响应。

Modbus任务处理函数（在单片机上实现过程）

```

1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
/*-----*/
/* Function Name : ModbusHandle
 * Input       :
 * Return      :
 * Description  : ModBus 处理函数
 *-----*/
void ModbusHandle(void)
{
    if(RcvBuf[0]!=LOCALADDR){ // 比较地址，是否是本机地址
        UartClearBuffer(); // 不是本机地址，清空接收缓冲区
        return;
    }
    switch(RcvBuf[1]){
        case 0x01: ReadCoilState('O'); // 读取输出线圈状态
        }break;
        case 0x02: ReadCoilState('I'); // 读取输入线圈状态
        }break;
        case 0x03: ReadRegiState('H'); // 读取保持寄存器
        }break;
        case 0x04: ReadRegiState('I'); // 读取输入寄存器
        }break;
        case 0x05: SetSingleCoil(); // 设置单个线圈
        }break;
        case 0x06: SetSingleRegVal('H'); // 设置单个寄存器
        }break;
        case 0x0F: SetMultCoil(); // 设置多个线圈
        }break;
        case 0x10: SetMultRegVal('H'); // 预置多个寄存器
        }break;
        default:{
            RcvBuf[1] |= 0x80;
            RcvBuf[2] = 1;
            UartSend(RcvBuf, 3); // 返回错误码(不支持的功能号)
        }break;
    }
    UartClearBuffer(); // 报文处理完成
}

```

函数中，RcvBuf 为串口接收缓冲区，如果接收了一个报文则，RcvBuf[0] 为从机地址，RcvBuf[1] 为MODBUS功能号。根据功能号做出响应，而具体的操作根据功能号在各自的函数中执行，相当于解析接收到的数据。

附录1: Modbus-RTU功能码

最常用功能码:

下面“线圈”“寄存器”其实分别直的就是“位变量”“16位变量”

- | | |
|-----------|--------|
| 01 (0x01) | 读线圈 |
| 02 (0x02) | 读离散量输入 |
| 03 (0x03) | 读保持寄存器 |
| 04(0x04) | 读输入寄存器 |
| 05 (0x05) | 写单个线圈 |
| 06 (0x06) | 写单个寄存器 |
| 15 (0x0F) | 写多个线圈 |
| 16 (0x10) | 写多个寄存器 |

- 01 Read Coil Status
- 02 Read Input Status
- 03 Read Holding Registers
- 04 Read Input Registers
- 05 Force Single Coil
- 06 Preset Single Register
- 07 Read Exception Status
- 11 (0B Hex) Fetch Comm Event Ctr
- 12 (0C Hex) Fetch Comm Event Log
- 15 (0F Hex) Force Multiple Coils
- 16 (10 Hex) Preset Multiple Regs
- 17 (11 Hex) Report Slave ID
- 20 (14Hex) Read General Reference
- 21 (15Hex) Write General Reference
- 22 (16Hex) Mask Write 4X Register
- 23 (17Hex) Read/Write 4X Registers
- 24 (18Hex) Read FIFO Queue



附录2: CRC Generation

CRC Generation Function

```
unsigned short CRC16(puchMsg, usDataLen)

unsigned char *puchMsg ;           /* message to calculate CRC upon */
unsigned short usDataLen ;         /* quantity of bytes in message */

{
    unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
    unsigned uIndex ;               /* will index into CRC lookup table */

    while (usDataLen--)             /* pass through message buffer */
    {
        uIndex = uchCRCHi ^ *puchMsgg++ ; /* calculate the CRC */
        uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex] ;
        uchCRCLo = auchCRCLo[uIndex] ;
    }

    return (uchCRCHi << 8 | uchCRCLo) ;
}
```



顶
1
踩

好文要顶

关注我

收藏该文

Magic2008
关注 - 3
粉丝 - 3

[+加关注](#)

10

« 上一篇: [Modbus软件开发实战指南 之 开发自己的Modbus Poll工具 - 2](#)

» 下一篇: [【HAL库每天一例】freemodbus移植](#)

posted @ 2017-03-15 11:31 Magic2008 阅读(1547) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！

【推荐】华为云7大明星产品0元免费使用

【推荐】腾讯云如何降低移动开发成本

【大赛】2018首届“顶天立地”AI开发者大赛



最新IT新闻：

- 苹果CarPlay已成为400多款汽车标配
 - 天猫发布以旧换新服务规范
 - Epic CEO：《堡垒之夜》团队达500人，Steam不发行因抽成高
 - 腾讯水逆：大公司走向伟大公司的必经之路？
 - 自媒体创业泡沫消亡史
- » 更多新闻...



最新知识库文章：

- 你可以把编程当做一项托付终身的职业
 - 评审的艺术——谈谈现实中的代码评审
 - 如何高效学习
 - 如何成为优秀的程序员？
 - 菜鸟工程师的超神之路 -- 从校园到职场
- » 更多知识库文章...