



ModBus

进入词条

全站搜索

帮助

首页

分类

特色百科

用户

权威合作

手机百科

个人中心

Modbus通讯协议

编辑

同义词 [ModBus](#)一般指Modbus通讯协议

Modbus是由Modicon（现为施耐德电气公司的一个品牌）在1979年发明的，是全球第一个真正用于工业现场的总线协议。

ModBus网络是一个工业通信系统，由带智能终端的**可编程序控制器**和计算机通过公用线路或局部专用线路连接而成。其**系统结构**既包括硬件、亦包括软件。它可应用于各种数据采集和过程监控。

ModBus网络只有一个主机，所有通信都由他发出。网络可支持247个之多的远程从属控制器，但实际所支持的从机数要由所用通信设备决定。采用这个系统，各PC可以和中心主机交换信息而不影响各PC执行本身的控制任务。

中文名	Modbus 通讯协议	发明时间	1979年
外文名	Modbus protocol	定 义	一个工业通信系统
		连接组成	带智能终端通过公用线路连接

目录

- 1 [Modbus协议](#)
- 2 [特点](#)
- 3 [传输方式](#)
- 4 [数据校验方式](#)
- 5 [功能码定义](#)

Modbus协议

编辑

为更好地普及和推动Modbus在基于**以太网**上的**分布式应用**，目前施耐德公司已将Modbus协议的所有权移交给IDA（Interface for Distributed Automation，分布式自动化接口）组织，并成立了Modbus-IDA组织，为Modbus今后的发展奠定了基础。在中国，Modbus已经成为国家标准GB/T19582-2008。据不完全统计：截止到2007年，Modbus的**节点**安装数量已经超过了1000万个。

Modbus协议是应用于**电子控制器**上的一种通用语言。通过此协议，控制器相互之间、控制器经由网络（例如以太网）和其它设备之间可以通信。它已经成为一种通用工业标准。有了它，不同厂商生产的控制设备可以连成**工业网络**，进行集中监控。此协议定义了一个控制器能认识使用的消息结构，而不管它们是经过何种网络进行通信的。它描述了一个控制器请求访问其它设备的过程，如何回应来自其它设备的请求，以及怎样侦测错误并记录。它制定了消息域格局和内容的公共格式。

当在同一Modbus网络上通信时，此协议决定了每个控制器需要知道它们的设备地址，识别按地址发来的消息，决定要产生何种行动。如果需要回应，控制器将生成反馈信息并用Modbus协议发出。在其它网络上，包含了Modbus协议的消息转换为在此网络上使用的帧或包结构。这种转换也扩展了根据具体的网络解决节地址、路由路径及错误检测的方法。

此协议支持传统的RS-232、RS-422、RS-485和以太网设备。许多工业设备，包括PLC，DCS，智能仪表等都在使用Modbus协议作为他们之间的通信标准。

特点

编辑

Modbus具有以下几个特点：

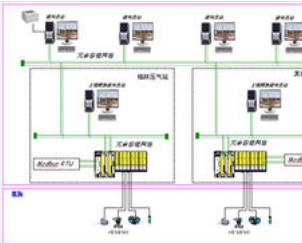
（1）标准、开放，用户可以免费、放心地使用Modbus协议，不需要交纳许可证费，也不会侵犯知识产权。目前，支持Modbus的厂家超过400家，支持Modbus的产品超过600种。

（2）Modbus可以支持多种电气接口，如RS-232、RS-485等，还可以在各种介质上传送，如双绞线、光纤、无线等。

（3）Modbus的帧格式简单、紧凑，通俗易懂。用户使用容易，厂商开发简单。

Modbus网络传输

标准的Modbus口是使用**RS-232-C**兼容**串行接口**，它定义了连接口的针脚、电缆、信号位、传输**波特率**、**奇偶校验**。控制器能直接或经由**Modem**组网。



Modbus通讯协议图册

V百科

往期



知名学者



郭天祥



谢希仁



周立

相关术语



uart



串口通讯



crc校验



现场总线



串口联网...



世电...

计算机类书籍



plc编程10...



visual c++...



linux网...



词条统计

浏览次数：988890次

编辑次数：53次历史版本

最近更新：2018-04-19

创建者：[...随缘...](#)

1 视频会议报价	12 自考本科
2 消防工程师	13 反无人机
3 车牌识别系统	14 空气源热泵
4 python培训	15 美国大学排名
5 透明图片	16 个人艺术写真
6 移动数据采集	17 小程序开发
7 can通讯协议	18 开发一个api
8 plc通讯协议	19 小程序开发
9 计算机编程入门	20 例题
10 ui设计需要学	21 儿童编程
11 代办上牌	22 明尼苏达大学

控制器通信使用主—从技术，即仅一设备（主设备）能初始化传输（查询）。其它设备（从设备）根据主设备查询提供的数据作出相应反应。典型的主设备：主机和可编程仪表。典型的从设备：[可编程控制器](#)。

主设备可单独和从设备通信，也能以广播方式和所有从设备通信。如果单独通信，从设备返回一消息作为回应，如果是广播方式查询的，则不作任何回应。**Modbus**协议建立了主设备查询的格式：设备（或广播）地址、功能代码、所有要发送的数据、一错误检测域。

从设备回应消息也由**Modbus**协议构成，包括确认要行动的域、任何要返回的数据、和一错误检测域。如果在消息接收过程中发生一错误，或从设备不能执行其命令，从设备将建立一错误消息并把它作为回应发送出去。

其它类型传输

在其它网络上，控制器使用对等技术通信，故任何控制器都能初始化和其它控制器的通信。这样在单独的通信过程中，控制器既可作为主设备也可作为从设备。提供的多个内部通道可允许同时发生的传输进程。

在消息位，**Modbus**协议仍提供了主—从原则，尽管网络通信方法是“对等”。如果一控制器发送一消息，它只是作为主设备，并期望从从设备得到回应。同样，当控制器接收到一消息，它将建立一从设备回应格式并返回给发送的控制器。

查询回应周期

（1）查询

查询消息中的功能代码告之被选中的从设备要执行何种功能。[数据段](#)包含了从设备要执行功能的任何附加信息。例如功能代码**03**是要求从设备读保持寄存器并返回它们的内容。数据段必须包含要告之从设备的信息：从何寄存器开始读及要读的寄存器数量。错误检测域为从设备提供了一种验证消息内容是否正确的方法。

（2）回应

如果从设备产生一正常的回应，在回应消息中的功能代码是在查询消息中的功能代码的回应。数据段包括了从设备收集的数据：像寄存器值或状态。如果有错误发生，功能代码将被修改以用于指出回应消息是错误的，同时数据段包含了描述此错误信息的代码。错误检测域允许主设备确认消息内容是否可用。

传输方式

 编辑

在**ModBus**系统中有2种传输模式可选择。这2种传输模式与从机**PC**通信的能力是同等的。选择时应视所用**ModBus**主机而定，每个**ModBus**系统只能使用一种模式，不允许2种模式混用。一种模式是**ASCII**（美国信息[交换码](#)），另一种模式是**RTU**（远程**终端设备**）。

用户选择想要的模式，包括**串口通信**参数（[波特率](#)、校验方式等），在配置每个控制器的时候，在一个**Modbus**网络上的所有设备都必须选择相同的传输模式和串口参数。所选的**ASCII**或**RTU**方式仅适用于标准的**Modbus**网络，它定义了在这些网络上连续传输的消息段的每一位，以及决定怎样将信息打包成消息域和如何解码。在其它网络上（像**MAP**和**Modbus Plus**）**Modbus**消息被转成与串行传输无关的帧。

传输模式特性

ASCII可打印字符便于故障检测，而且对于用高级语言（如**Fortran**）编程的主计算机及主**PC**很适宜。**RTU**则适用于**机器语言**编程的计算机和**PC**主机。

用**RTU**模式传输的数据是8位二进制字符。如欲转换为**ASCII**模式，则每个**RTU**字符首先应分为高位和低位两部分，这两部分各含4位，然后转换成十六进制等量值。用以构成**报文的ASCII**字符都是十六进制字符。**ASCII**模式使用的**字符**虽是**RTU**模式的两倍，但**ASCII**数据的译码和处理更为容易一些，此外，用**RTU**模式时报文字符必须以连续数据流的形式传送，用**ASCII**模式，字符之间可产生长达**1s**的间隔，以适应速度较慢的机器。

控制器能设置为两种传输模式（**ASCII**或**RTU**）中的任何一种在标准的**Modbus**网络通信。

ASCII模式

当控制器设为在**Modbus**网络上以**ASCII**（美国标准信息交换代码）模式通信，一个信息中的每8个比特作为2个**ASCII**字符传输，如数值**63H**用**ASCII**方式时，需发送两个字节，即**ASCII**"6"（**0110110**）和**ASCII**"3"（**0110011**），**ASCII**字符占用的位数有7位和8位，国际通用7位为多。这种方式的主要优点是字符发送的时间间隔可达到1秒而不产生错误。

代码系统

- 十六进制，**ASCII**[字符](#)0...9,A...F
- 消息中的每个**ASCII**字符都是一个十六进制字符组成

每个字节的位

- 1个起始位
- 7个**数据位**，最小的有效位先发送
- 1个**奇偶校验位**，无校验则无

1个停止位（有校验时），2个Bit（无校验时）
错误检测域
· LRC(纵向冗长检测)
RTU模式
当控制器设为在Modbus网络上以RTU模式通信，在消息中的每个8Bit字节按照原值传送，不做处理，如63H，RTU将直接发送01100011。这种方式的主要优点是：数据帧传送之间没有间隔，相同波特率下传输数据的密度要比ASCII高，传输速度更快 ^[1] 。
代码系统
8位二进制，十六进制数0...9，A...F
消息中的每个8位域都是一或两个十六进制字符组成
每个字节的位
1个起始位
8个数据位，最小的有效位先发送
1个奇偶校验位，无校验则无
1个停止位（有校验时），2个Bit（无校验时）

数据校验方式

 编辑

CRC

CRC域是两个字节，包含一16位的二进制值。它由传输设备计算后加入到消息中。接收设备重新计算收到消息的CRC，并与接收到的CRC域中的值比较，如果两值不同，则有误。

CRC是先调入一值是全“1”的16位寄存器，然后调用一过程将消息中连续的8位字节和当前寄存器中的值进行处理。仅每个字符中的8Bit数据对CRC有效，起始位和停止位以及奇偶校验位均无效。

CRC产生过程中，每个8位字符都单独和寄存器内容相异或（XOR），结果向最低有效位方向移动，最高有效位以0填充。LSB被提取出来检测，如果LSB为1，寄存器单独和预置的值或一下，如果LSB为0，则不进行。整个过程要重复8次。在最后一位（第8位）完成后，下一个8位字节又单独和寄存器的当前值相异或（XOR）。最终寄存器中的值，是消息中所有的字节都执行之后的CRC值。

CRC添加到消息中时，低字节先加入，然后高字节。

CRC-16错误校验程序如下：报文（此处只涉及数据位，不指起始位、停止位和任选的奇偶校验位）被看作是一个连续的二进制，其最高有效位（MSB）首选发送。报文先与 X_{16} 相乘（左移16位），然后看 $X_{16}+X_{15}+X_{12}+1$ 除， $X_{16}+X_{15}+X_{12}+1$ 可以表示为二进制数11000，0000，0000，0101。整数商位忽略不记，16位余数加入该报文（MSB先发送），成为2个CRC校验字节。余数中的1全部初始化，以免所有的零成为一条报文被接收。经上述处理而含有CRC字节的报文，若无错误，到接收设备后再被同一多项式（ $X_{16}+X_{15}+X_{12}+1$ ）除，会得到一个零余数（接收设备核验这个CRC字节，并将其与被传送的CRC比较）。全部运算以2为模（无进位）。

习惯于成串发送数据的设备会首选送出字符的最右位（LSB-最低有效位）。而在生成CRC情况下，发送首位应是被除数的最高有效位MSB。由于在运算中不用进位，为便于操作起见，计算CRC时设MSB在最右位。生成多项式的位序也必须反过来，以保持一致。多项式的MSB略去不记，因其只对商有影响而不影响余数。

生成CRC-16校验字节的步骤如下：

- ①例如一个16位寄存器，所有数位均为1。
- ②该16位寄存器的高位字节与开始8位字节进行“异或”运算。运算结果放入这个16位寄存器。
- ③把这个16寄存器向右移一位。
- ④若向右（标记位）移出的数位是1，则生成多项式10，1000，000，0000，001和这个寄存器进行“异或”运算；若向右移出的数位是0，则返回③。
- ⑤重复③和④，直至移出8位。
- ⑥另外8位与该十六位寄存器进行“异或”运算。
- ⑦重复③~⑥，直至该报文所有字节均与16位寄存器进行“异或”运算，并移位8次。
- ⑧这个16位寄存器的内容即2字节CRC错误校验，被加到报文的最高有效位。

另外，在某些非ModBus通信协议中也经常使用CRC16作为校验手段，而且产生了一些CRC16的变种，他们是使用CRC16多项式 $X^{16}+X^{15}+X^{14}+1$ ，单首次装入的16位寄存器为0000；使用CRC16的反序 $X^{16}+X^{14}+X^{11}+1$ ，首次装入寄存器值为0000或FFFFH。

CRC简单函数如下：

```
1 unsignedchar*puchMsg;/*要进行CRC校验的消息*/
2
3
4 unsignedshortusDataLen;/*消息中字节数*/
5
6 unsignedshortCRC16(puchMsg,usDataLen)
7
8
9 {
10
11
12 unsignedcharuchCRCHi=0xFF;/*高CRC字节初始化*/
13
14
15 unsignedcharuchCRCLo=0xFF;/*低CRC字节初始化*/
16
17
18 unsignedduIndex;/*CRC循环中的索引*/
19
20
21 while(usDataLen--)*/*传输消息缓冲区*/
22
23
24 {
25
```



ModBus

进入词条



```
31 uchCRCLo=auchCRCLo[uIndex];
32
33 }
34
35
36
37 return((uchCRCHi<<8)|uchCRCLo);
38
39 }
40
41 /*CRC高位字节值表*/
42
43 staticunsignedcharauchCRCHi[]={
44 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
45 0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
46 0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,
47 0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
48 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,
49 0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,
50 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,
51 0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
52 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
53 0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,
54 0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,
55 0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
56 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
57 0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,
58 0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
59 0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
60 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
61 0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
62 0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,
63 0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
64 0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
65 0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,
66 0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,
67 0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
68 0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
69 0x80,0x41,0x00,0xC1,0x81,0x40
70 };
71
72 //CRC低位字节值表
73
74 staticcharauchCRCLo[]={
75 0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,
76 0x07,0xC7,0x05,0xC5,0xC4,0x04,0xCC,0x0C,0x0D,0xCD,
77 0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,
78 0x08,0xC8,0xD8,0x18,0xD9,0x1B,0xDB,0xDA,0x1A,
79 0x1E,0xDE,0xDF,0x1F,0xDD,0x1D,0xDC,0x14,0xD4,
80 0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,
81 0x11,0xD1,0xD0,0x10,0xF0,0x30,0x31,0xF1,0x33,0xF3,
82 0xF2,0x32,0x36,0xF6,0xF7,0x37,0xF5,0x35,0x34,0xF4,
83 0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,
84 0x3B,0xFB,0x39,0xF9,0xF8,0x38,0x28,0xE8,0xE9,0x29,
85 0xEB,0x2B,0x2A,0xEA,0xEE,0x2E,0x2F,0xEF,0x2D,0xED,
86 0xEC,0x2C,0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,
87 0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,0xA0,0x60,
88 0x61,0xA1,0x63,0xA3,0xA2,0x62,0x66,0xA6,0xA7,0x67,
89 0xA5,0x65,0x64,0xA4,0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,
90 0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,
91 0x78,0xB8,0xB9,0x79,0xBB,0x7B,0x7A,0xBA,0xBE,0x7E,
92 0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,0xB4,0x74,0x75,0xB5,
93 0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,
94 0x70,0xB0,0x50,0x90,0x91,0x51,0x93,0x53,0x52,0x92,
95 0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,0x9C,0x5C,
96 0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,
97 0x99,0x59,0x58,0x98,0x88,0x48,0x49,0x89,0x4B,0x8B,
98 0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
99 0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,
100 0x43,0x83,0x41,0x81,0x80,0x40
101 };
```

LRC

LRC错误校验用于ASCII模式。这个错误校验是一个8位二进制数，可作为2个ASCII十六进制字节传送。把十六进制字符转换成二进制，加上无循环进位的二进制字符和二进制补码结果生成LRC错误校验（参见图）。这个LRC在接收设备进行核验，并与

分享



被传送的LRC进行比较，冒号（：）、回车符号（CR）、换行字符（LF）和置入的其他任何非ASCII十六进制字符在运算时忽略不计。

- 1 Modbus 协议
- 2 特点
- 3 传输方式
- 4 数据校验方式
- 5 功能码定义

功能码定义

编辑

表1 ModBus功能码

01	READ COIL STATUS
02	READ INPUT STATUS
03	READ HOLDING REGISTER
04	READ INPUT REGISTER
05	WRITE SINGLE COIL
06	WRITE SINGLE REGISTER
15	WRITE MULTIPLE COIL
16	WRITE MULTIPLE REGISTER

自动化通讯协定

- [BSAP](#)
- [CC-Link](#)
- [CIP](#)
- [CAN](#)

Baidu 百科

Modbus

进入词条

编辑

收藏

赞

程序自动化

- [HART Protocol](#)
- [MECHATROLINK](#)
- [PieP](#)
- [SERCOS III](#)
- [RAPIEnet](#)
- [Honeywell SDS](#)
- [MelsecNet](#)
- [Profibus](#)
- [Sinec H1](#)
- [HostLink](#)
- [Modbus](#)
- [PROFINET IO](#)
- [SynqNet](#)
- [INTERBUS](#)
- [Optomux](#)
- [SERCOS interface](#)
- [TTEthernet](#)

工业控制系统

- [OPC DA](#)
- [OPC HDA](#)
- [OPC UA](#)
- [MTConnect](#)

智能建筑

- [1-Wire](#)
- [BACnet](#)
- [C-Bus](#)
- [DALI](#)
- [DSI](#)
- [KNX](#)
- [LonTalk](#)
- [Modbus](#)
- [oBIX](#)
- [VSCP](#)
- [X10](#)
- [xAP](#)
- [ZigBee](#)

输配电通讯协定

- [IEC 60870-5](#)
- [DNP3](#)
- [IEC 60870-6](#)
- [IEC 61850](#)
- [IEC 62351](#)
- [Modbus](#)
- [Profibus](#)

智能电表

- [ANSI C12.18](#)
- [IEC 61107](#)
- [DLMS/IEC 62056](#)
- [M-Bus](#)
- [Modbus](#)
- [ZigBee Smart Energy 2.0](#)

车用通讯

- [CAN](#)
- [FMS](#)
- [FlexRay](#)
- [IEBus](#)
- [J1587](#)
- [J1708](#)
- [J1939](#)
- [Keyword Protocol 2000](#)
- [LIN](#)
- [MOST](#)
- [NMEA 2000](#)
- [VAN](#)

参考资料

1. Schneider-Electric.Modicon Modbus Protocol Reference Guide[R].1996

词条标签： 通信技术

猜你喜欢

[modbus协议转换器](#)

[郑多燕燕根粉正品包装](#)

[modbus转profibus](#)

[设计化肥包装](#)

[通讯协议转换器](#)

[平洲附近有没有电脑培训班](#)

[海尔空调电脑板多少钱](#)

[免费下载爱鲜蜂](#)

[下载爱鲜蜂](#)

[中发里二手房](#)



碳纤维布加固



车牌识别系统



儿童编程



虹吸同层排水



工业级交换机



小马达



移动数据采集器

新手上路

[成长任务](#)

[编辑规则](#)

[编辑入门](#)

[百科术语](#)

我有疑问

[我要质疑](#)

[参加讨论](#)

[在线客服](#)

[意见反馈](#)

投诉建议

[举报不良信息](#)

[投诉侵权信息](#)

[未通过词条申诉](#)

[封禁查询与解封](#)

分享



- 1 Modbus 协议
- 2 特点
- 3 传输方式
- 4 数据校验方式
- 5 功能码定义



ModBus

进入 词条

编辑

收藏

赞

分享