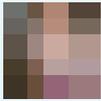


发表于 2010-5-8 10:24:00



• loveplc

该协议实际上适用于 PLC 编程端口以及 FX-232AW 模块的通信。

通讯格式:

命令	命令码	目标设备
DEVICE READ CMD	"0"	X,Y,M,S,T,C,D
DEVICE WRITE CMD	"1"	X,Y,M,S,T,C,D
FORCE ON CMD	"7"	X,Y,M,S,T,C
FORCE OFF CMD	"8"	X,Y,M,S,T,C

传输格式: RS232C 波特率: 9600bps 奇偶: even 校验: 累加方式 (和校验) 字符: ASCII

16 进制代码:

ENQ	05H	请求
ACK	06H	PLC 正确响应
NAK	15H	PLC 错误响应
STX	02H	报文开始
ETX	03H	报文结束

三菱 FX 系列 PLC 编程口通信协议举例

1、DEVICE READ (读出软设备状态值)

计算机向 PLC 发送:

始	命令	首地址	位数	终	和校验
STX	CMD	GROUP ADDRESS	BYTES	ETX	SUM

2、DEVICE WRITE (向 PLC 软设备写入值)

始	命令	首地址	位数	数据				终	和校验
STX	CMD	GROUP ADDRESS	BYTES	1ST DATA	2ND DATA	LAST DATA	ETX	SUM

3、位设备强制置位/复位

FORCE ON 置位

始	命令	地址	终	和校验
STX	CMD	ADDRESS	ETX	SUM
02h	37h	address	03h	sum

FORCE OFF 复位

始	命令	地址	终	和校验
STX	CMD	ADDRESS	ETX	SUM
02h	38h	address	03h	sum

说明：

1. 帧中的 BYTES 表示需要读取或者写入的字节数。
2. 地址算法上有说明。
3. 累加和是从 STX 后面一个字节开始累加到 ETX 的和。

这个就是网上流传的的比较完整的通讯协议说明了，可能对刚刚入门的朋友有点难度。因为光看这些对一个编程的初学者不知道如何下手，那么我将在下面的文章把不同地址的如何读写详细说明一下。

上次的文章主要讲述了三菱 PLC 的协议以及一些格式。但是如何来利用这个协议来写自己的程序呢？

我现在就讲述利用 VB 如何对 PLC 的地址读写和置位与复位等等。

首先，在这里我先讲述一下对 bit 的地址置位和复位操作。我要对 bit 位进行置位复位我们就要了结一下 bit 位的地址概念。

bit 位包括 S,X,Y,T,M,C 这些位，那么这些地址在 PLC 里面是如何存放的呢？学过计算机的人都应该知道计算机里面只有 0 和 1，把这些 0 和 1 按不同的顺序排列就能组成要计算机语言，在计算机里面也有地址，PLC 也是一台简单的计算机所以和它的原理是一样的这些 bit 位都是被有序的存储起来的。通过我的试验发现三菱 FX 系列 PLC 地址如下：

S 就是系统位，在三菱 FX 系列里面开始是 S0 点被存储在地址 0008 (H)，X0 是 0408 (H)，Y0 是 0508，T0 是 0708，M0 是 0808，C0 是 0D08。这样就对 PLC 的地址有了一定的了解，知道的 bit 位的地址我们如何来置位复位呢？对地址的通讯协议是这样的：

位设备强制置位/复位

FORCE ON 置位

始	命令	地址	终	和校验
STX	CMD	ADDRESS	ETX	SUM
02h	37h	address	03h	sum

FORCE OFF 复位

始	命令	地址	终	和校验
STX	CMD	ADDRESS	ETX	SUM
02h	38h	address	03h	sum

PLC 返回

ACK(06H) 接受正确 NAK(15H) 接受错误

有了格式我们就很清楚的来实现置位复位功能了。那么我们先做一些准备工作。上面的协议可能读者对和效验不太清楚那么我来解释一下，所谓和效验就是为了保证通讯的正确性，下面就是一个例子：

例子：

STX ,CMD ,ADDRESS, BYTES, ETX, SUM

02H, 30H, **31H,30H,46H,36H**, 30H,34H, 03H, 37H,34H

SUM=CMD.....ETX;

30h 31h 30h 46h 36h 30h 34h 03h=74h;

累加和超过两位取低两位

累加和是从 STX 后面一个字节开始累加到 ETX 的和

那么我们做一个小程序来实现它

```
*****
```

```
* 和校验
```

```
*Check_FCS(mStr)
```

```
*mStr 需要检查的字符串
```

```
*****
```

```
Public Function Check_FCS(mStr As String) As String
```

```
Dim K As Integer, mTest As Integer
```

```
For K = 1 To Len(mStr)
```

```
mTest = mTest Asc(Mid(mStr, K, 1))
```

```
Next
```

```
Check_FCS = Right(Hex(mTest), 2)
```

```
End Function
```

现在可以对 bit 位进行置位了程序如下：

```
*****
```

```
*强置 bit 位地址
```

```
*SetM(address)
```

```
*****
```

```
Public Function SetM(address As String) As Boolean
```

```
Dim Q As Long, myTest As String, myHex As String, QQ As String,
```

```
mStr As Integer, Num As Integer
```

```
On Error GoTo SetMErr
```

```
select Case Left(address, 1)
```

```
Case "M"
```

```
mStr = 2048
```

```

Case "X"
mStr = 1024
Case "Y"
mStr = 1280
Case "S"
mStr = 0
Case "T"
mStr = 1536
Case "C"
mStr = 3584
End Select
If mStr = 1024 Then
Q = OCT_to_DEC(Mid(address, 2)) mStr
Elseif mStr = 1280 Then
Q = OCT_to_DEC(Mid(address, 2)) mStr
Else
Q = Val(Mid(address, 2)) mStr 这个算法只能写到 M1741 点 address *
256 8 只能写到 M255 点
End If
QQ = Hex(Q)
If Len(QQ) = 3 Then
QQ = "0" & QQ
Elseif Len(QQ) = 2 Then
QQ = "00" & QQ
Elseif Len(QQ) = 1 Then
QQ = "000" & QQ
End If
QQ = Right(QQ, 2) & Left(QQ, 2) 换位操作 地位在前高位在后
myHex = "7" & QQ & Chr(3) 置位操作 7 是命令置位命令 则 8 是复位操
作
myHex = Chr(2) & myHex & Check_FCS(myHex)
Form1.MSComm.Output = myHex
Call Sleep(Tim)
myTest = Form1.MSComm.Input
If Asc(myTest) = 6 Then SetM = True
Exit Function
SetMErr:
SetM = False
End Function
*****

*复位 bit 位地址
*ResetM(address)
*****

Public Function ResetM(address As String) As Boolean
Dim Q As Long, myTest As String, myHex As String, QQ As String,
mStr As Integer
On Error GoTo ResetMErr

select Case Left(address, 1)
Case "M"
mStr = 2048
Case "X"
mStr = 1024
Case "Y"
mStr = 1280
Case "S"

```

```

mStr = 0
Case "T"
mStr = 1536
Case "C"
mStr = 3584
End Select

If mStr = 1024 Then
Q = OCT_to_DEC(Mid(address, 2)) mStr
Elseif mStr = 1280 Then
Q = OCT_to_DEC(Mid(address, 2)) mStr
Else
Q = Val(Mid(address, 2)) mStr 这个算法只能写到 M1535 点 address *
256 8 只能写到 M255 点
End If
QQ = Hex(Q)
If Len(QQ) = 3 Then
QQ = "0" & QQ
Elseif Len(QQ) = 2 Then
QQ = "00" & QQ
Elseif Len(QQ) = 1 Then
QQ = "000" & QQ
End If
QQ = Right(QQ, 2) & Left(QQ, 2)
myHex = "8" & QQ & Chr(3)
myHex = Chr(2) & myHex & Check_FCS(myHex)
Form1.MSComm.Output = myHex
Call Sleep(Tim)
myTest = Form1.MSComm.Input
If Asc(myTest) = 6 Then ResetM = True
Exit Function
ResetMErr:
ResetM = False
End Function

```

这里我写成了一个函数形式，方便以后随时调用下篇将继续介绍对 D 的地址位的读写操作等。

这次我们来说一下对三菱 FX 系列 PLC 的 D 地址的读写操作，先来看看他的通讯协议如下：

1、DEVICE READ（读出软设备状态值）

计算机向 PLC 发送：

始	命令	首地址	位数	终	和校验
STX	CMD	GROUP ADDRESS	BYTES	ETX	SUM

例子：从 D123 开始读取 4 个字节数据

02h	30h	31h,30h,46h,36h	30h,34h	03h	37h,34h
-----	-----	------------------------	---------	-----	---------

地址算法:address=address*2+1000h

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/147062056101006133>