TouchWin

信捷触摸屏 自定义通讯协议

操作手册

无锡信捷电气股份有限公司

资料编号 HC083 20120924 2C6

目录

1 概述 3
2 协议设置 4
2-1 了解协议设备对象 4
2-2 注册及创建协议文件 5
2-3 编写协议文件 7
2-3-1 简介 7
2-3-2 在哪里编写协议代码 7
2-3-3 协议代码的结构 8
2-3-4 设置步骤 8

1 概述

■ 什么叫自定义通讯协议？

自定义通讯协议的意思就是:借助触摸屏自定义协议方式，编写协议驱动文件，以实现与触摸屏软件中不支持设备的通讯。

■ 为什么要使用自定义通讯协议？

通信协议又称通信规程，是指通信双方对数据传送控制的一种约定。约定中包括对数据格式，同步方式，传送速度，传送步骤，检纠错方式以及控制字符定义等问题做出统一规定，通信双方必须共同遵守。也就是说触摸屏与设备之间的沟通必须讲述相同的语言，才能互相传输信息。

在触摸屏编辑软件 Touchwin 里，已自带了市面上主流 PLC，变频器和仪表等设备的通讯协议，用户只需要针对自己的设备在触摸屏里选择相应的协议即可实现通讯。



图示：设备列表

但是如果用户的设备不在通讯设备选择列表中，请先阅读该设备所支持的通讯协议，因为该设备的协议可能与设备列表中已存在的设备协议相同，例如被广泛使用的 modbus 协议。

如果确认在设备列表中没有支持该设备的协议，这种情况，就必须借助触摸屏自定义协议方式，编写协议驱动文件。

2　协议设置

2-1　了解协议设备对象

为了更好的理解自定义协议的制作过程，这里我们以仪表 V900（注：V900 是一个虚构的名称）的自由格式通讯协议的编写过程为例：

仔细阅读仪表 V900 的通讯协议，确定需要发送和接受的代码信息。根据用户的要求，需要在触摸屏上显示“当前重量”，“流量”显示 和“目标量”。

V900 的地址空间分布			
当前重量	H42	流量	H43
目标量	H44		

V900 读寄存器 通讯格式				
发送信息	站号	读指令功能码 03	目标寄存器地址	和校验
V900 回复信息	站号	寄存器内容高字节	寄存器内容低字节	和校验

V900 的站号设置为 1，波特率设为 9600，偶校验，8 位数据位，1 位停止位。根据上表中 V900 的通讯格式，下面枚举了“发送数据帧内容”及“回应的格式”。

1、读当前重量 ：H01 H03 H42 sumcheck;

回复：01 （回复内容高字节）（回复内容低字节） sumcheck;

2、读流量：　　H01 H03 H43 sumcheck;

回复：01 （回复内容高字节）（回复内容低字节） sumcheck;

3、读目标量：　　H01 H03 H44 sumcheck;

回复：01 （回复内容高字节）（回复内容低字节） sumcheck;

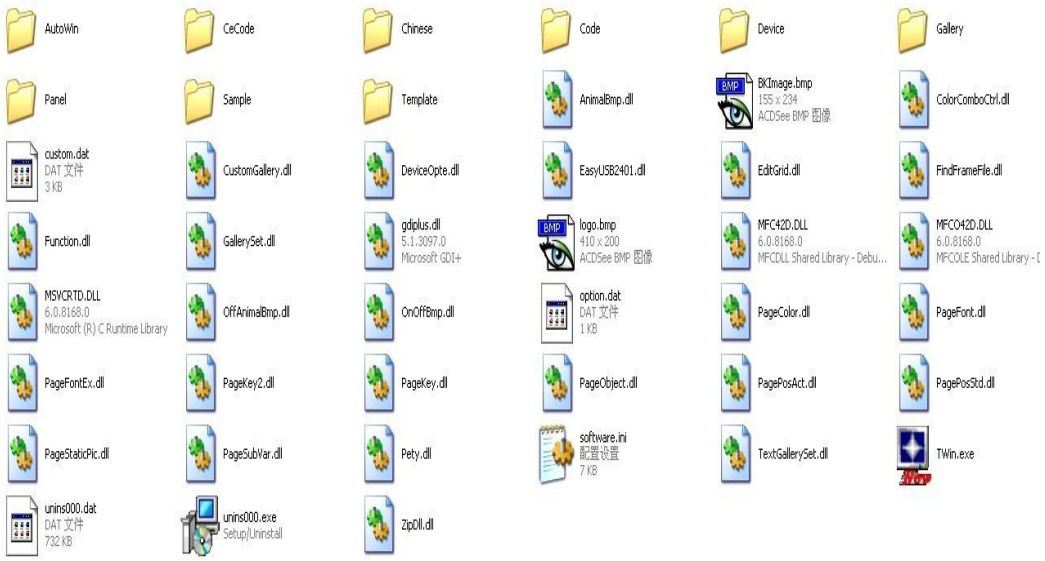
2-2 注册及创建协议文件

■ 简介

在编辑软件的设备列表中注册 V900 的选项，并且建立协议的载体文件。

■ 步骤

1、打开 Touchwin 软件安装的根目录：



2、在根目录里找到 custom.dat 的文件并用记事本打开，在文件的末尾加“V900= V900 仪表”，“=”左边为配置文件名称，“=”右边为设备列表中显示的名称；设置完成后保存并退出。（注：如果无法保存，请将该 dat 文件属性中的保密属性去掉）

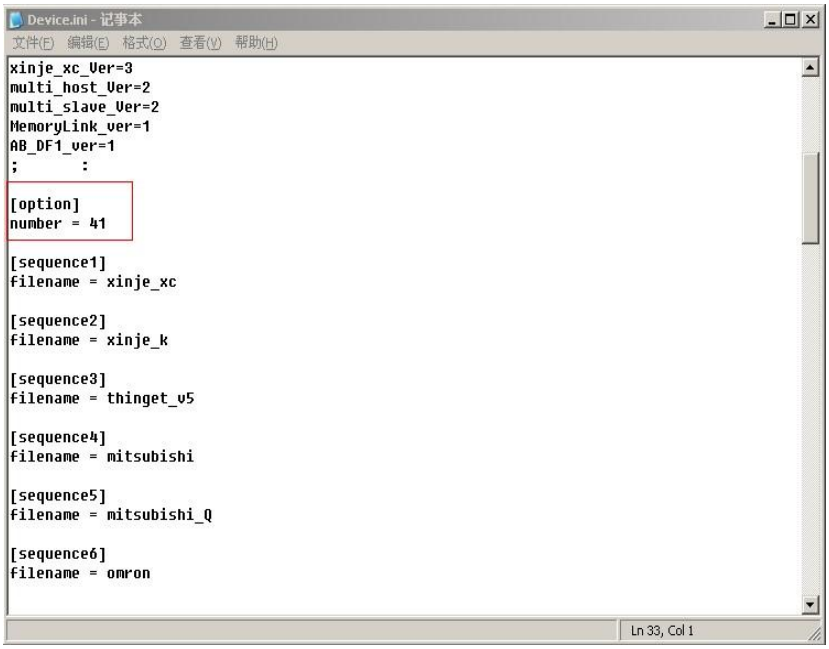


3、在根目录下找到 Device 的文件夹，打开该文件夹，找到并用记事本打开配置文件 Device.ini，在原文中找到：

[option]

number = n

在该例中，n=41，因为要新增一个项目 V900，那么 n+1 该值变成 42；

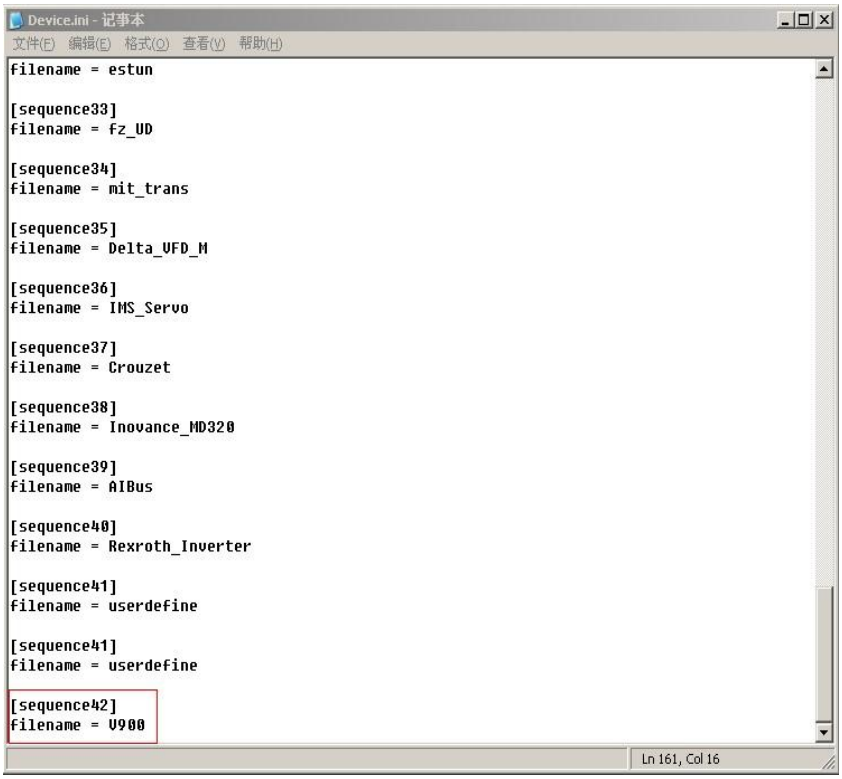


4、在配置文件 Device.ini 中，将滑动条拖移到底部，在最后位置增加两行文字：

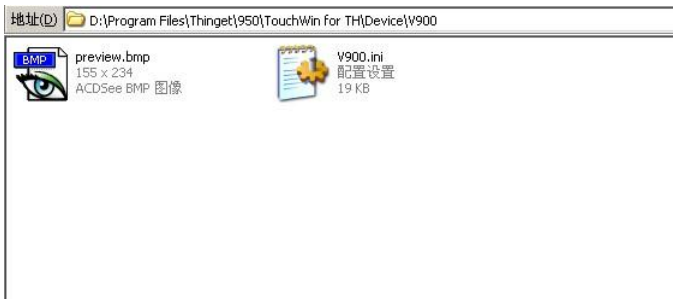
[sequence42]

filename = V900

输入结束后，保存文件并退出。



5、在根目录下找到并打开 Device 文件夹，新建一个名称为 V900 的文件夹，并在该文件夹里建立下面两个文件。(注：简便的方法是从别的设备文件夹中复制粘贴过来，然后将 ini 配置文件的名称修改为 V900)



6、双击打开 Touchwin 软件,在设备列表中你可以看到 V900 已经被列入到选型列表之中。到此，注册及创建协议文件的过程就结束了。



2-3 编写协议文件

2-3-1 简介

在上一节中已经完成注册及创建协议的过程，那么根据设备的通讯要求我们需要在协议文件里编写协议的代码。

2-3-2 在哪里编写协议代码

经过上一章的创建之后，在根目录下找到并打开 Device 文件夹下的 V900 文件夹，再双击打开 V900.ini 配置文件，然后在这个配置文件中输入自定义协议的代码。

2-3-3 协议代码的结构

这些协议的代码就是触摸屏进行通讯运作的规则。依照先后顺序协议共分成以下几大部分：

- 文件描述
- 缺省通讯参数设定
- 缺省站号设定
- 数据对象描述
- 通讯规则
- 对象优化

2-3-4 设置步骤

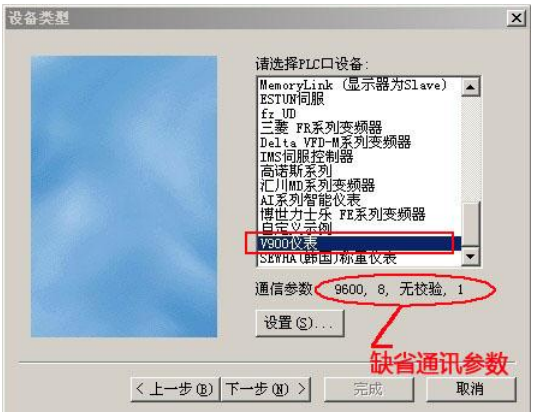
- **Descript　文件描述**

在 V900.ini 文件中编写以下内容。其为固定格式，套用即可。“文件描述”代码内容 如下：

```
[descript]
DeviceModelCode=22
FirmwareName=UserDefine
DownLoadDll = \Device\UserDefine\DownLoad.dll
SpecInfoDll = \Device\UserDefine\DownLoad.dll
DeviceType = 100
```

- **Communication　缺省通讯参数设定**

在软件中选择设备时其缺省的通讯参数。如下图：



“缺省通讯参数设定”代码内容：

```
[Communication]
BaudRate=9600          ;波特率
DataBits=8             ;数据位
Parity=2               ;奇偶校验   0-None, 1-Odd, 2-Even
StopBits=0             ;停止位     0-1 bit, 1-1.5 bits, 2-2 bits
```

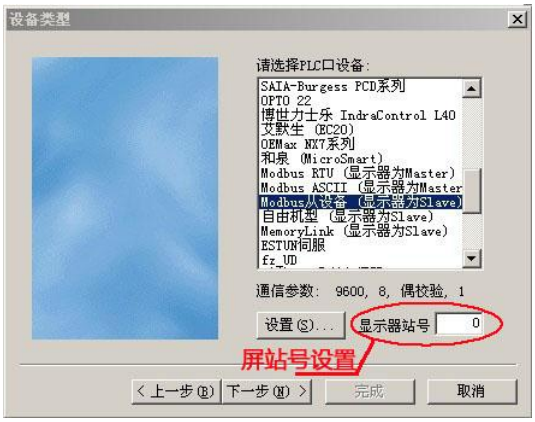
● **StationNo　缺省站号设定**

两部分分成屏的站号和设备站号。代码内容如下：

```
[StationNo]
PanelWithStationNo=0        ;0：屏站号不可设置 1：可设置          【注 1】
PanelDefaultStationNo=      ;屏缺省站号  本例屏的站号为不可设置，则此空。
PanelMinStationNo=          ;屏最小站号  本例屏的站号为不可设置，则此空。
PanelMaxStationNo=          ;屏最大站号  本例屏的站号为不可设置，则此空。
DeviceWithStationNo=1       ;0：设备站号不可设置   1：可设置   【注 1】
DeviceDefaultStationNo=1    ;设备缺省站号为 1
DeviceMinStationNo=0        ;设备可设置的最小站号
DeviceMaxStationNo=255      ;设备可设置的最大站号
```

注释：

【注 1】：触摸屏作为通讯从机的时候，PanelWithStationNo=1，如下图，在设备列表的下面显示器站号变为可设。当 PanelWithStationNo=0，此处即无此设置。DeviceWithStationNo 的功能类此。



● **Object 数据对象描述**

要读取 V900 的当前重量，那么就要描述“当前重量”所占用的空间大小，所对应通讯规则等。代码内容入下：

```
[Object]
ObjectNum=13    ；所描述的总的数据对象个数。
[Object1]       ；第一个数据对象
CanAct=1        ；数据对象占用空间基本单位 0-位, 1-寄存器, 2-寄存器组   【注 1】
IDSymbol=当前重量   ；在软件部件里的选项名称   【注 2】
TypeNo=0        ；其对应的“对象优化”号
bitlength = 16  ；位长度，“当前重量”占用了 16 个位
DescripNum=1    ；对象描述的段数   【注 3】
CanSelectType=4 ；数据类型的可选择种类（0000 0100）;5-3 bytes,4-n Regs,3-dword,
                  2-word,1-byte,0-bit【注 4】
ReadWord = 1    ；读寄存器调用通讯规则号

[Object1Descrip1]   ;对象描述第一段
Caption=            ;固定格式
Type=0              ;软件中数据输入对应类型：
```

0-Number,1-(0-7),2-(00-07),3-(0-15),4-(00-15),
5-(0-F),6-(00-0F),7-(0-31)　　【注 5】

DefaultNumber=0　　;软件中数据输入缺省数据
MinNumber=0　　;软件中数据输入最小限制
MaxNumber=0　　;软件中数据输入最大限制
format=10　　；软件中数据输入的进制
NumberStep=1　　；固定格式

注释：

【注 1】：Touchwin 软件将根据该选项来区分位,寄存器及寄存器组。如果在这里选择位，那么在软件中的按钮，指示灯等位操作部件对象类型选项里出现这些对象名称。如果在这里选择寄存器，那么在软件中的数据输入，数据显示等字操作部件对象类型选项里出现这些对象名称。如果在这里选择寄存器组，那么在软件中的字符显示，中文输入等寄存器组操作部件对象类型选项里出现这些对象名称。

【注 2】：如下图在对象类型的选项框里包含了当前重量这个选项。



【注 3】：对象描述的段数，在上例中段数为 1，也即在对象类型名称后只有一个描述值。如上图。但在某些设备协议里，需要两段描述值。在欧姆龙 PLC 的描述协议里，对象 D 的位操作，如 D100B,其特殊之处在其数字值最后一位的范围在 0-F，而其它的位则为十进制格式。在代码的描述中，我们将把 100 与 B 作为两段分开描述，代码如下:A 部分为最后一位之前的值 格式的描述，B 部分为最后一位值格式的描述。效果如下图。

[Object9]
CanAct=0
IDSymbol=D
TypeNo=0

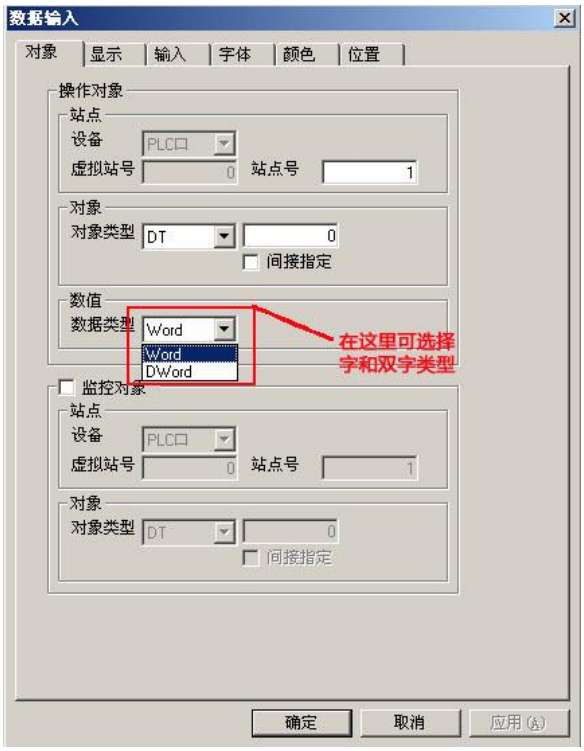
bitlength = 16
DescripNum=2
CanSelectType=1
ReadBit=16
SetBitOn=23
SetBitOff=23

[Object9Descrip1]　A 部分　　;
Caption=
Type=0
DefaultNumber=0
MinNumber=0
MaxNumber=99999
format=10
NumberStep=1

[Object9Descrip2]　　B 部分
Caption=.
Type=3
DefaultNumber=0



【注 4】: 数据类型的可选择种类 ，选择数据种类由 8 位二进制数组成，第 5 位-3 bytes,第 4 位-n Regs, 第 3 位-dword,第 2 位-word,第 1 位-byte,第 0 位-bit，如果需要字和双字可选，组成二进制数为 00001100，那么转换成十进制则为 12，CanSelectType=12，效果如下图。



【注 5】：软件中地址输入对应类型：

选择号码	数据输入对应类型
0	0 到限定输入最大值
1	(0-7)
2	(00-07)
3	(0-15)
4	(00-15)
5	(0-F)
6	(00-0F)
7	(0-31)

在地址类型为 0-F 时



● **CommuRule 通讯规则**

屏在同设备通讯时所发送数据帧的具体格式。这一步是协议的核心部分，要求对设备协议有详细的了解。代码如下：

```
[CommuRule]
CommuRuleNum = 19   ；通讯规则的个数
IsUseASC = 0        ；是否使用 asc 码转化   0:No use ASC, 1: usc ASC as Transmit Data
LowBitToASC = 0     ；是否忽略高位          0:NO, 1:YES     【注 1】
IsHightBitNext = 0  ；使用 asc 转化前提下，在字节中高位在前还是低位在前    0:NO, 1:YES
                    【注 2】
IsHightByteNext = 0 ；基本单位为字，高字节在前还是低字节在前       0:No, 1:YES
IsHightWordNext = 0 ；基本单位为双字，高字在前还是低字在前         0:NO, 1:YES

[CommuRule1]
IsInherit = 0       ；该通讯规则是派生还是原生    原生=0    派生=1    【注 3】
GroupNum = 2        ；发送和接受块数，通讯过程中一般都是一发一回的过程，所以该值一般为 2
Group1Type= 0       ；第一组定义为发送数据帧
Group1Blocks = 4    ；第一组块数为 4
Group2Type = 1      ；第二组定义为接受数据帧
Group2Blocks = 3    ；第一组块数为 3
Group1Block1 = 4    ；第一组第一块数据：设备站号        【注 4】
Group1Block2 = 5    ；第一组第二块数据：命令码          【注 4】
Group1Block2_CmdNo = 3
Group1Block3 = 1                ；第一组第三块数据：静态数据块    【注 4】
Group1Block3_StaticValue = 0x42 ；
Group1Block4 = 11               ；第一组第四块数据：校验码        【注 4】
Group1Block4_StartBlock = 1
Group1Block4_EndBlock = 3
Group1Block4_SelectMethod = 1
Group2Block1 = 4                ；第二组第一块数据：设备站号      【注 4】
Group2Block2 = 10               ；第二组第二块数据：数据区        【注 4】
Group2Block2_DataAreaType = 1
Group2Block3 = 11               ；第二组第三块数据：校验码        【注 4】
Group2Block3_StartBlock = 1
Group2Block3_EndBlock = 2
Group2Block3_SelectMethod = 1

[CommuRule2]
IsInherit = 1                   ；该通讯规则是派生还是原生    原生=0    派生=1
InheritRule = 1                 ；表示该条规则是从哪条规则中派生出来的。
Group1Block3 = 1
Group1Block3_StaticValue = 0x44   } 枚举出需要重新定义的块
```

注释：

【注 1】：在使用 ASC 转换，举例：0x56 ASC 转化后为 0x35,0x36，对具体的块可指定只表示低位忽略高位，如果 IsHightBitNext = 1，那么 0x56 ASC 转化后即为 0x36。

【注 2】：高低字节倒置，在使用 ASC 转换，举例：0x56 ASC 转化后为 0x35,0x36，如果 IsHightBitNext = 1，那么 0x56 ASC 转化后即为 0x36，0x35。

【注 3】：

原生：通讯规则中对于每个块都有定义。

派生：通讯规则与指定的某条原生通讯规则类似，在编写派生通讯规则，只需要枚举出需要重新定义的块即可。

【注 4】：在大多数的通讯协议中，也就是在一个通讯数据帧，都会包含几个内容：目标站号，功能代码，数据区，数据个数，校验码等。那么在 Twin 自定义协议中，是怎样来对这些内容进行区分定义的呢？下面代码是对“效验码”的一个描述：

Group1Block4 = 11
Group1Block4_StartBlock = 1
Group1Block4_EndBlock = 3
Group1Block4_SelectMethod = 1

Group1Block4 = 11 是对第一组第 4 块内容的一个职能定义，代号 11 即指定该功能块为校验码。Group1Block4_StartBlock ，Group1Block4_EndBlock，Group1Block4_SelectMethod 都是对第一组第 4 块的校验码的扩展定义。可以看出，每个块的组成一般包含两部分：职能定义 和 扩展定义。对于某些内容简单的职能块，“扩展定义”是没有的，比如：Group1Block1 = 4，即定义为设备站号。

下面对职能定义的特殊代号及其补充说明进行说明：（X：组数，Y：块数）

职能定义号	1
代表含义	静态数据块 ：某些块在其值已经确定的情况下使用。
扩展定义	Group X Block Y _StaticValue

职能定义号	2	
代表含义	块的长度 ：指定开始块 到 结束块 之间占用的字节数。	
扩展定义	Group X Block Y _LowBitToASC	ASC 转化后只保留低位. 举例：0x56 ASC 转化后为 0x35，0x36，经过 LowBitToASC 那么结果就只有 0x36 留下。
	Group X Block Y _StartBlock	开始块的标号，统计时包含此块
	Group X Block Y _EndBlock	结束块的标号，统计时包含此块

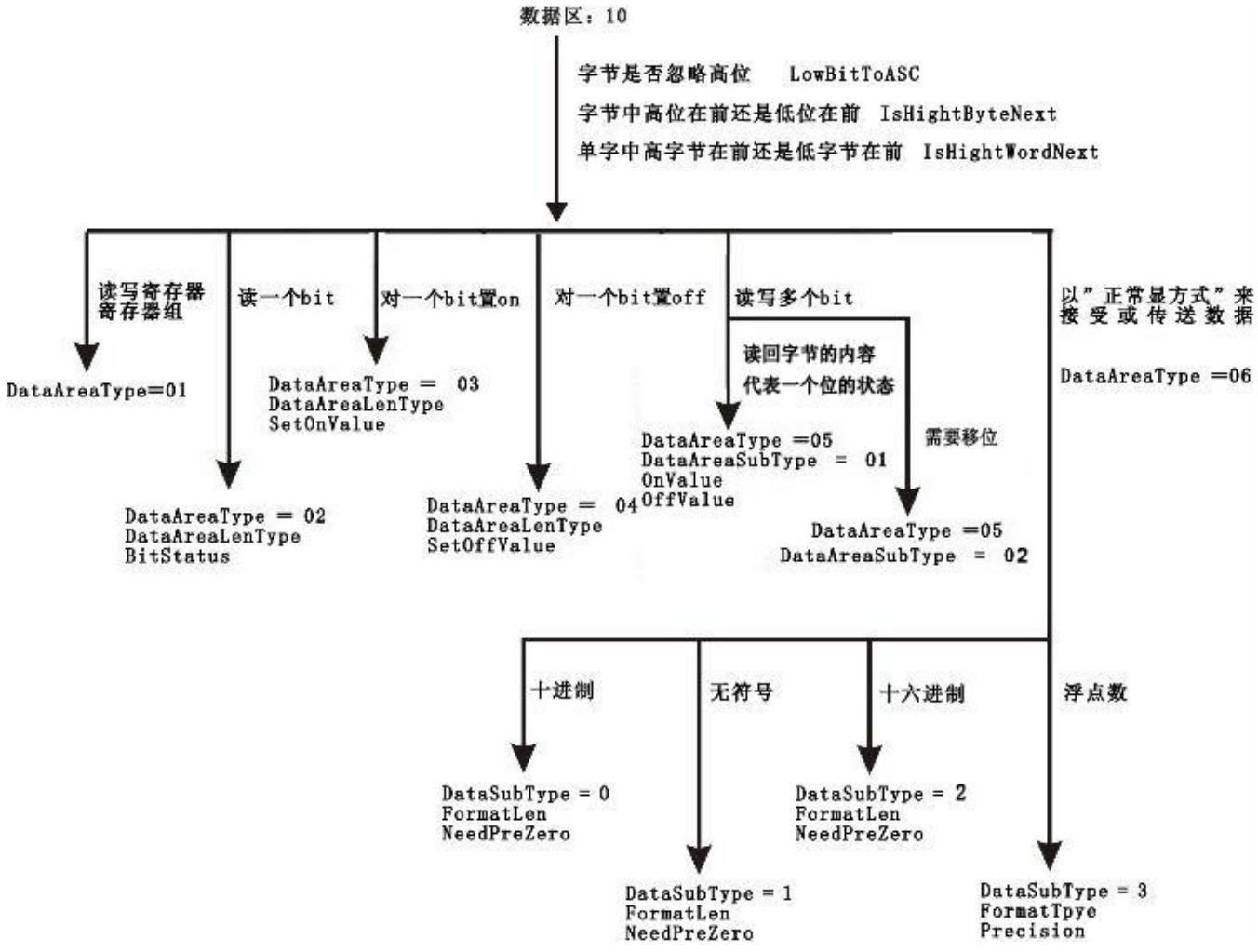
职能定义号	4	
代表含义	设备站号：所连接设备的通讯站点号，也称局号。	
扩展定义	Group X Block Y _LowBitToASC	ASC 转化后只保留低位

职能定义号	5	
代表含义	命令码：在通讯规则中，一般分为写命令操作，读命令操作 等，不同的命令操作会有不同的命令码。	
扩展定义	Group X Block Y _LowBitToASC	ASC 转化后只保留低位.
	Group X Block Y _CmdNo	命令码指定。比喻在 modbus 协议中多字读的功能码为 3 ，那么可表示为 Group X Block Y _CmdNo=3

职能定义号	7		
代表含义	参数地址： 例如：在通讯中要 读取 D1000 地址 里面的内容 ，那么我们需要在软件中的对象类型中输入 D1000，在这里 1000 就是参数地址。如下图所示：  参数地址在软件中设置为 1000，如果在通讯帧中将该参数地址转化成 16 进制的话，即为 03E8。 参数地址在软件中设置为 1000，如果在通讯帧中将该参数地址转化成 ASC 的数据格式输出，即为 31 30 30 30		
扩展定义	参数地址转化成 16 进制	Group X Block Y _ DispMode=0	模式选择(DispMode=0 地址转化成 16 进制)
		Group X Block Y _LowBitToASC	ASC 转化后只保留低位.
		Group X Block Y_IsHightByteNext	在单字中，高字节在前还是低字节在前
		Group X BlockY _IsHightWordNext	在双字中，高字在前还是低字在前
		Group X Block Y _ ParaAddress	地址转化表达式
		为什么需要转化表达式？因为有的情况下，面向用户的参数地址与通讯帧中的参数地址是不一样地，比如：在信捷 PLC 中，参数地址 TD0 在通讯中的实际地址编号为 H3000，那么在由 TD0 向 H3000 的转化就需要通过一个表达式来进行。如果在软件中，输入参数 TD N，那么 TD N 在通讯帧中的实际地址编号则应为 N+0x3000。那么这一步我们可以写成 Group X Block Y_ParaAddress = N+0x3000 。	
		Group X Block Y _HoldSpaceSize	（1：Byte；2：Word；3：DWord）在通讯帧中，该参数所占用的空间大小。Group X BlockY_HoldSpaceSize=2 说明占用空间为一个字大小。
	参数地址转化成 ASC 数据格式	Group X Block Y _ DispMode = 1	模式选择（DispMode=1：转化成 ASC 数据格式）
		Group X Block Y _LowBitToASC	ASC 转化后只保留低位.
		Group X Block Y_IsHightByteNext	在单字中，高字节在前还是低字节在前
		Group X BlockY _IsHightWordNext	在双字中，高字在前还是低字在前
		Group X Block Y _FormatLen	在通讯帧中该参数地址占用多少 Byte
		Group X Block Y _NeedPreZero	（0：No need；1：Need） 是否需要前导 0
		Group X Block Y _ ParaAddress	地址转化表达式

职能定义号	8	
代表含义	参数个数：就是要执行操作参数的个数。在一条通讯帧中，为了提高通讯的效率，可能会对连续的几个参数地址进行操作。	
扩展定义	Group X Block Y _LowBitToASC	ASC 转化后只保留低位.
	Group X Block Y_IsHightByteNext	在单字中，高字节在前还是低字节在前
	Group X BlockY _IsHightWordNext	在双字中，高字在前还是低字在前
	Group X Block Y _HoldSpaceSize	（1：Byte；2：Word；3：DWord） 在通讯帧中，该参数所占用的空间大小。 Group X Block Y _HoldSpaceSize=2 说明占用空间为一个字大小。
	Group X Block Y _CountMethod	（1：BITS；2：BYTES；3：WORDS；4：DWORDS）

数据区的描述比较复杂（职能定义号为 10），其结构图如下：



职能定义号	10	
代表含义	数据区：存储 接受和发送数据的地址区。例如：要将值 100 写入到地址 D100 中，那么该值 100 就是数据帧中数据区所存储的内容。另外数据区还负责存储读取的数据。	
	Group X Block Y _LowBitToASC	ASC 转化后只保留低位.

<table>
<tr><td rowspan="15">扩展定义</td><td>Group X Block Y_IsHightByteNext</td><td>在单字中，高字节在前还是低字节在前</td></tr>
<tr><td>Group X BlockY
_IsHightWordNext</td><td>在双字中，高字在前还是低字在前</td></tr>
<tr><td>Group X BlockY
_DataAreaLenType</td><td>（1：byte；2：word；3：dword）
Group X BlockY _HoldSpaceSize= （2，3，4）有效</td></tr>
<tr><td>Group X BlockY _BitStatus</td><td>Group X BlockY _HoldSpaceSize= 2 有效，读回的位值</td></tr>
<tr><td>Group X BlockY _SetOnValue</td><td>Group X BlockY_DataAreaType= 3 有效</td></tr>
<tr><td>Group X BlockY _SetOffValue</td><td>Group X BlockY_DataAreaType= 4 有效</td></tr>
<tr><td>Group X BlockY
_DataAreaSubType</td><td>Group X BlockY_DataAreaType= 5 有效</td></tr>
<tr><td>Group X BlockY _OnValue</td><td>Group X BlockY_DataAreaType= 5
Group X BlockY _DataAreaSubType=1 有效
对应字节与 OnValue 的值相等则状态为 on,否则为 off.</td></tr>
<tr><td>Group X BlockY _OffValue</td><td>Group X BlockY_DataAreaType= 5
Group X BlockY _DataAreaSubType=1 有效</td></tr>
<tr><td>Group X BlockY _DataSubType</td><td>（0：十进进制 1：无符号 2：十六进制）
Group X BlockY_DataAreaType= 6 有效</td></tr>
<tr><td>Group X BlockY _FormatLen</td><td>Formatlen 描述的是格式化的最大长度;
Group X BlockY_DataAreaType= 6
Group X BlockY _DataSubType=（0，1，2）有效
注：Formatlen 描述的是格式化的最大长度;当为 0 时，表示数据的长度完全自适应,此时 NeedPreZero 不起作用。</td></tr>
<tr><td>Group X BlockY _NeedPreZero</td><td>（0：No need；1：Need）
Group X BlockY_DataAreaType= 6
Group X BlockY _DataSubType=（0，1，2）有效</td></tr>
<tr><td>Group X BlockY _FormatTpye</td><td>（0：表示以 dddd.dddd 格式输出）
Group X BlockY_DataAreaType= 6
Group X BlockY _DataSubType= 3 有效</td></tr>
<tr><td>Group X BlockY _Precision</td><td>指定小数位后的精度
Group X BlockY_DataAreaType= 6
Group X BlockY _DataSubType= 3 有效</td></tr>
<tr><td>Group X BlockY _ModNumber</td><td>数据移位
Group X BlockY _DataAreaType=5
Group X BlockY _DataAreaSubType=2 有效</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">GroupXBlockY _DataAreaType：由于在数据帧里面存储的数据多种多样，可能是位、字、字节或者其他类型。为了保证通讯中数据交换的可靠性，需要对数据帧中存储数据的类型进行详细描述定义。以下就是这些描述的分类。</td></tr>
<tr><td colspan="2"><table>
<tr><th>赋值</th><th>含 义</th><th>解 释</th></tr>
<tr><td>1</td><td>读写寄存器、寄存器组</td><td></td></tr>
</table></td></tr>
</table>

		2	读一个bit	从读回的word或byte里，取出需要的bit。 以与信捷V5变频器通讯为例，我们需要读回变频器的正转/反转的状态。 存储变频器状态的地址为H2101,该地址的第三位作为变频器的正转/反转的状态指示，当该位为0值的时候，表示变频器处于正转状态，当该位为1值的时候，表示变频器处于反转状态。 以下为读正反转的通讯代码： [CommuRule12] ；读正反转 IsInherit = 1 InheritRule = 1 Group1Block3 = 7 Group1Block3_HoldSpaceSize = 2 Group1Block3_ParaAddress = N+0x2101 Group2Block4 = 10 Group2Block4_DataAreaType = 2 Group2Block4_DataAreaLenType = 2 Group2Block4_BitStatus = P & (1<<2) 重点关注红色的代码内容，在这里，会首先读回整个变频器状态H2101里面的值，但是如何从H2101中挑出第三位的值来呢？ Group2Block4_BitStatus = P & (1<<2) 返回的数据值我们默认为符号P,这里P代表H2101里面的值，（1<<2）的意义就是将0000 0001 左移2位，也就变为0000 0100，然后将0000 0100 与P与 运算。这样子就可以得到返回值H2101中第三位的值。 如果P & (1<<2)=0;表明读回位状态为OFF 如果P & (1<<2)=非0; 表明读回位状态为ON 【表达式使用 详见 本章“表达式”一节】
		3	对一个bit置ON	这种方式在控制变频器时使用极多，在触摸屏中，可设置一个按钮，该按钮在处于按下状态(ON)时,变频器启动，该按钮在处于释放状态（OFF）时，变频器停止。 控制变频器正转的通讯命令是：将值H0002写入H2000; 控制变频器停止的通讯命令是：将值H0007写入H2000; 就是当控制“正转/停止”的按钮在处于按下状态（ON）时，触摸屏通过串口发送“将值H0002写入H2000的命令码”，按钮处于释放状态（OFF）时，触摸屏通过串口发送“将值H0007写入H2000的命令码” 以下是信捷V5变频器的“正转/停止”的控制协议： [Object12] ；正转/停止 CanAct=0 ; Bit IDSymbol= TypeNo=11 DescripNum=1 bitlength = 1 CanSelectType=1

				ReadBit=13 SetBitOn=17 SetBitOff=18 ----------中间程序省略---------- [CommuRule17] ；设置正转 IsInherit = 1 InheritRule = 15 Group1Block3 = 7 Group1Block3_HoldSpaceSize = 2 Group1Block3_ParaAddress = N+0x2000 Group1Block4 = 10 Group1Block4_DataAreaType = 0x03 Group1Block4_DataAreaLenType = 0x02 Group1Block4_SetOnValue = 0x0002 Group2Block3 = 7 Group2Block3_HoldSpaceSize = 2 Group2Block3_ParaAddress = N+0x2000 [CommuRule18] ；减速停机/停止 IsInherit = 1 InheritRule = 15 Group1Block3 = 7 Group1Block3_HoldSpaceSize = 2 Group1Block3_ParaAddress = N+0x2000 Group1Block4 = 10 Group1Block4_DataAreaType = 0x04 Group1Block4_DataAreaLenType = 0x02 Group1Block4_SetOffValue = 0x0007 Group2Block3 = 7 Group2Block3_HoldSpaceSize = 2 Group2Block3_ParaAddress = N+0x2000
		4	对一个bit置OFF	使用方法同上
		5	读写多个bit	1、DataAreaSubType = 0x01 当读回到数据区中一个字节表示一个bit的状态,（强调：读回的字节只能代表一个bit的状态，比喻：返回字节值H24代表位状态为ON，返回字节值H45代表位状态为OFF。） 以下为代码内容： DataAreaSubType = 0x01 OnValue = 0x24 OffValue = 0x45 2、DataAreaSubType = 0x02 在信捷PLC当中，触摸屏发出读取位M0命令时，其会读回一个字

				节的位数据到缓存器中，并且触摸屏从缓冲器中第一位取值作为M0的显示值，此时如果缓冲器的第一位刚好对应M0，那么就不需要移位 ，即Group2Block4_ModNumber = 1；如果缓冲器的第n位对应M0，那么就需要移位 ，即Group2Block4_ModNumber = n； 以下是信捷PLC，读取线圈内容M的代码： Group2Block4 = 10 Group2Block4_IsHightByteNext = 1 Group2Block4_DataAreaType = 5 Group2Block4_DataAreaSubType = 2 Group2Block4_ModNumber = 1
		6	以”正常显示方式”来接受或传送数据	以“正常显示方式”来接受或传送数据。比如：在表示十进制数3000，一般的情况是把其转化成十六进制BB8,然后将BB8拆分成高低字节，也就是0B，B8两个字节。 而在这里所说的正常显示方式 指： 十进制3000先拆开变为 3 0 0 0 ；再转化成ASC码后变为 33 30 30 30； 这种正常显示方式分成两大类： 1）十进进制 ，无符号 ，十六进制 显示 2）浮点数显示 ● 十进进制 ，无符号 ，十六进制 显示 举例 1： 读取某仪表内部时间数据，比如 数据帧中返回 时间数据 为 31 31 30 31 34 32 ，表示读回的时间是：11 点 01 分 42 秒，那么在触摸屏上如何显示出 110142 这个时间呢？代码及释义如下： Group2Block7=10 ；定义接受区第七块为数据区 Group2Block7_DataAreaType = 0x06 ;定义该块 以”正常显示方式”来接受数据 Group2Block7_DataSubType = 1 ;定义接受的数据以无符号数 显示 Group2Block7_FormatLen = 6 ;定义接受的数据长度为 6 位 举例 2：NeedPreZero 在读数据的时候没有意义，但是在写数据的时候作用很大。 比如设置某仪表中的参数，设置该值 100，代码如下： Group1Block7 = 10 ;定义接受区第七块为数据区。 Group1Block7_DataAreaType= 6 ;定义该块以”正常显示方式”来接受数据。 Group1Block7_DataSubType= 1 ;定义接受的数据以无符号数显示。 Group1Block7_FormatLen= 6 ;定义接受的数据长度为 6 位。 Group1Block7_NeedPreZero = 1 ;定义是否需要前导 0（0：不需要，1：需要）。如果 Group1Block7_NeedPreZero = 1，那么发送的数据结构为：30 30 30 31 30 30；但如果 Group1Block7_NeedPreZero = 0，那么发送的数据结构为：31 30 30，其结果则完全不一样。

				● 浮点数显示 举例：读取某仪表内部数据，比如在数据帧中数据为 31 34 33 37 32 32 36 34 2E 31 30，表示读回的数据内容是：14372264.10，那么在代码中如何描述该数据区呢？代码及释义如下： Group2Block18= 10 ;定义接受区第 18 块为数据区。 Group2Block18_DataAreaType = 6 ;定义该块以“正常显示方式”来接受数据 Group2Block18_DataSubType = 3 ;定义接受的数据以浮点数 显示出来 Group2Block18_FormatTpye = 0 ;定义为 dddd.dddd 输出格式。设置该值对于数据读没有效果，只对数据写发生作用。 Group2Block18_Precision = 2 ;定义小数位后有两位

职能定义号	11	
代表含义	校验码：系统内部预先设置根据校验方法所导出的校验公式编制成的校验程序，当带有校验码的代码输入系统时，系统利用校验程序对输入的本体码进行运算得出校验结果之后，再将校验结果与输入代码的校验码进行对比来检测输入的正确与否。如果两者一致，则表明代码输入正确，系统允许进入，如果不一致，则表明代码输入有误，系统拒绝进入，并要求代码重新输入。 CRC 校验举例： Group1Block7 = 11 Group1Block7_StartBlock = 1 Group1Block7_EndBlock = 6 Group1Block7_SelectMethod = 3 Group1Block7_HoldSpaceSize = 2 Group1Block7_InitValue = 0xffff	
扩展定义	Group X Block Y _LowBitToASC	ASC 转化后只保留低位.
	Group X Block Y _ IsHightByteNext	在单字中，高字节在前还是低字节在前
	Group X Block Y _ IsHightWordNext	在双字中，高字在前还是低字在前
	Group X Block Y _ StartBlock	开始块的标号，统计时包含此块
	Group X Block Y _ EndBlock	结束块的标号，统计时包含此块
	Group X Block Y _SelectMethod	校验的方式（0：Lrc；1：Sum；2：-Sum；3：CRC_Modbus；4：CRC_IBM；5：CRC_ITU） 注：-Sum 校验是指前面数据和加上它正好为零，结果类型为字节。

职能定义号	13
代表含义	不确定数据块：当某数据块不需要在意或者不能确定的情况下，使用不确定数据块。

● **ObjectType ： 数据对象优化**

优化的作用是整合通讯规则，在达到同样通讯效果的前提下极力减少数据的通讯量。比如：在触摸屏里，如果在一个画面有地址连续的 5 个寄存器，没有通讯优化，触摸屏每次扫描都需要发送 5 次读指令数据帧，有通讯优化，触摸屏就会自动选择多个字内容读数据帧，一次性将 5 个寄存器读回来。下面内容是描写数据优化的一串完整代码。

```
[ObjectType]
ObjectTypeNum = 2     ；题头 ： 优化对象类型的个数

[ObjectType0]          ；优化对象类型 1   【注 1】
IsConvert = 0          ；需要转化为别的对象类型？（0：不需要 1：需要）
Regs = 1               ；该对象类型所选用第几条通讯规则，例子中选择的是第一条通讯规则
MaxLength = 32         ；【注 2】
BitLength = 16         ；该对象类型的位长个数

[ObjectType1]          ；优化对象类型 2   【注 3】
IsConvert = 1          ；需要转化为别的对象类型？（0：不需要 1：需要）
ToObjectType = 0       ；转化成第几号对象类型，例子中右边的值为 0，代表将当前 ObjectType1 转
                         化为 ObjectType0
TargetAddress = H+0x3000    ；目标地址的转化公式 ，
BitLength = 16              ；该对象类型的位长个数
```

注释：

【注 1】：在上节内容“Object 对象类型”描述中，我们知道，每个对象都对应于一个对象优化的描述，如下例：

```
[Object1]
CanAct=1
IDSymbol=D
TypeNo=0                    其对应的“对象优化”号，在这里其对应优化对象 ObjectType0;
bitlength = 16
DescripNum=1
CanSelectType=12
```

【注 2】：在进行通讯优化的过程中，最大读取对象的个数，这里不宜设的过大，因为很多通讯设备的地址不够，或者不允许单次这么大的通讯量，设的过大，会造成设备之间通讯不上等影响。如果触摸屏的通讯负担不多不至于影响操作使用，可以将其设置成 1。

【注 3】：大家很容易理解优化类型 ObjectType0 的结构，而对 ObjectType1 的描述就感觉很奇怪，为什么要将 ObjectType1 转化成 ObjectType0？这里以信捷 XC PLC 中的一段协议作为例程。

```
[Object1]
CanAct=1
IDSymbol=D
TypeNo=0
bitlength = 16;
```

DescripNum=1
CanSelectType=12
ReadWord = 1
ReadDWord = 1
WriteWord = 47
WriteDWord = 8

[Object2]
CanAct=1
IDSymbol=TD
TypeNo=1
bitlength = 16;
DescripNum=1
CanSelectType=12
ReadWord = 2
ReadDWord = 2
WriteWord = 9
WriteDWord = 9
----------中间代码省略----------
[ObjectType]
ObjectTypeNum = 2

[ObjectType0]
IsConvert = 0
Regs = 1
MaxLength = 32
BitLength = 16

[ObjectType1]
IsConvert = 1
ToObjectType = 0
TargetAddress = H+0x3000
BitLength = 16

注意：对于带有下划线的红色代码内容，普通寄存器 D 对应优化类型 0，而存储时间数据的寄存器 TD 对应优化类型 1。首先，TD 和 D 这两种寄存器在 PLC 程序应用中的作用是有区别的，但是从物理空间上来看，他们的地位是完全相同的。参看信捷《XC 系列可编程控制器用户手册》中的《Modbus 通讯功能》一章，通讯地址：寄存器 D 的 Modbus 地址空间是从 0 开始的，D0 对应地址 4x0，而 TD 的 Modbus 地址空间是从 4x3000 开始的，也就是说 D0 对应地址 4x0，TD0 对应地址 4x3000，而且 D、TD 在被进行读写的时候，通讯帧的格式是完全一样的，也就是说，在和外部设备通讯，交换数据的层面上，D 和 TD 没有任何的区别，只是地址不一样。那么在通讯优化的时候，经过类型的转化，可以达到更好的优化效果。

■ 表达式

举例：0x1600+N/8

用到的算术符号：+，-，*，/，%，(，)，<<，>>，&，|

数字表达：无符号十进制数（ 123 ）；
十六进制数（ 0xe1；0XE1 ）
注：不支持负数的表达，如‘-123’非法

特殊符号：N（读写元件第一段描述值）；
M（读写元件第二段描述值）；
R（寄存器组数）；
P（“数据区”中返回的数据）
注：表达式中不分大小写。

算术符号的运算级别（ 高->低 ）：第一级：（ , ）
第二级：* , / , %
第三级：+ , -
第四级：<< , >>
第五级：&
第六级：|

无锡信捷电气股份有限公司
江苏省无锡市蠡园开发区
创意产业园 7 号楼四楼
邮编： 214072
电话： (0510)85134136
传真： (0510)85111290

Xinje Electronic Co., Ltd.
4th Floor Building 7,Orignality Industry park, Liyuan Development Zone, Wuxi City, Jiangsu Province
214072
Tel: (510)85134136
Fax: (510)85111290