IR-2110
8路隔离开关量输入数据采集模块

产品使用说明书

Ver 1.0

北 京 异 特 路 智 能 通 讯 科 技 有 限 公 司
Beijing itRob Intelligent Telecommunication Co , Ltd.

版本历史

版本号	发布日期	变更内容
Ver 1.0	2012.7.23	新产品发布。 产品版本号为：201201

目　　录

1. 产品介绍

1.1 概述

IR-2110为一款8路隔离开关量输入的分布式远程工业数据采集模块。

IR-2110的通信接口采用工业领域使用最为广泛的RS-485总线进行通讯与控制，可广泛应用于各种工业测量与控制系统中。控制主机或主控制器通过RS-485总线与IR-2110模块通讯，可实现对工业现场的开关量信号的采集，并将开关量信号反馈回（通过RS-485）主控计算机,主控计算机以此为依据而采取相应的控制输出。

IR-2110可同时支持两种通信协议，IRASCII协议与ModbusRTU协议。其中IRASCII协议与研华公司的ADAM协议以及弘格公司的DCON协议相兼容，其指令集兼容于NuDAM、ADAM等模块；而ModbusRTU协议是当前工控领域里使用最为广泛的协议。同时支持以上两种通信协议无疑使产品具有更广泛的适用性。实际应用时用户可根据需要自行设定模块的通信协议。

1.2 产品特点

◆支持双协议：IRASCII协议与ModbusRTU协议。可与采用类似协议的模块挂在同一个RS-485网络中，最大限度地给工程设计人员带来方便。

◆提供485(通讯状态)指示灯和PWR(电源)指示灯，方便用户查看模块工作状态。

◆RS-485接口具备防雷和光电隔离双重保护功能，使产品工作更加安全可靠。

◆模块支持在线设置，用户可通过RS-485对模块的工作参数进行设置。所有的设置参数全部保存于模块内部的EEPROM中，与那些需要打开产品外壳来通过跳线进行设置的产品相比更加简单、方便。

◆内置硬件看门狗,使系统工作更安全更可靠。

◆输入通道支持脉冲输入锁存功能，可以采集脉冲量信号，如安防报警系统的报警信号采集等。

◆本产品采用合金外壳，坚固耐用，也具备良好的防电磁干扰特性。

◆安装方式即可以选择标准DIN导轨式安装，也可以选择采用螺丝固定的壁挂式安装方式，具体方式由用户在购买产品时指定。

1.3 端子分布

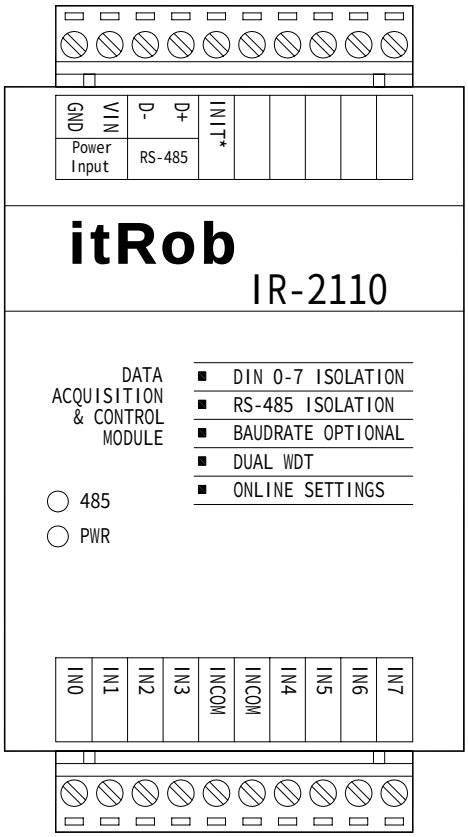


图1.3 IR-2110端子分布

1.4 特性与参数

输入通道	输入通道数量	8路
	隔离方式	单端隔离
	隔离电压	3000Vrms
	数字电平0	0～1V
	数字电平1	+4V～30V
	输入阻抗	3KΩ
最小无故障时间	100000小时	
RS-485接口	接口信号	D+、D-（两线半双工）
	通讯速率	1200、2400、4800、9600、19200、38400、57600、115200，（单位：bps）用户可通过命令设置模块的通讯波特率
	防雷保护电压	7～8V
	防雷保护容量	600W
	光电隔离保护	3000Vrms
	带载点数	一条RS-485总线可最多挂接32个模块，如果用户使用的模块数量超过32个，可通过RS-485中继器（如IR-1301S）或RS-485分配器（如IR-1305A）来扩展RS-485网络。
LED指示灯	PWR：电源指示灯（红色） 485：RS-485通信指示灯（绿色）	
通讯协议	同时支持IRASCII协议与ModbusRTU协议，可通过命令设置	
指令数量	IRASCII	10条
	ModbusRTU	11条
功耗	1.1W	
输入电源电压	+9V～+30VDC(具备电源反接保护功能)	
温度	-20℃～70℃	
湿度	5%～90% 无冷凝	
安装方式	标准DIN导轨卡装或壁挂式安装（由用户在购买产品时指定）	
体积/尺寸	（参见第5节外型及尺寸）	

表1.4 IR-2110参数列表

1.5 结构图

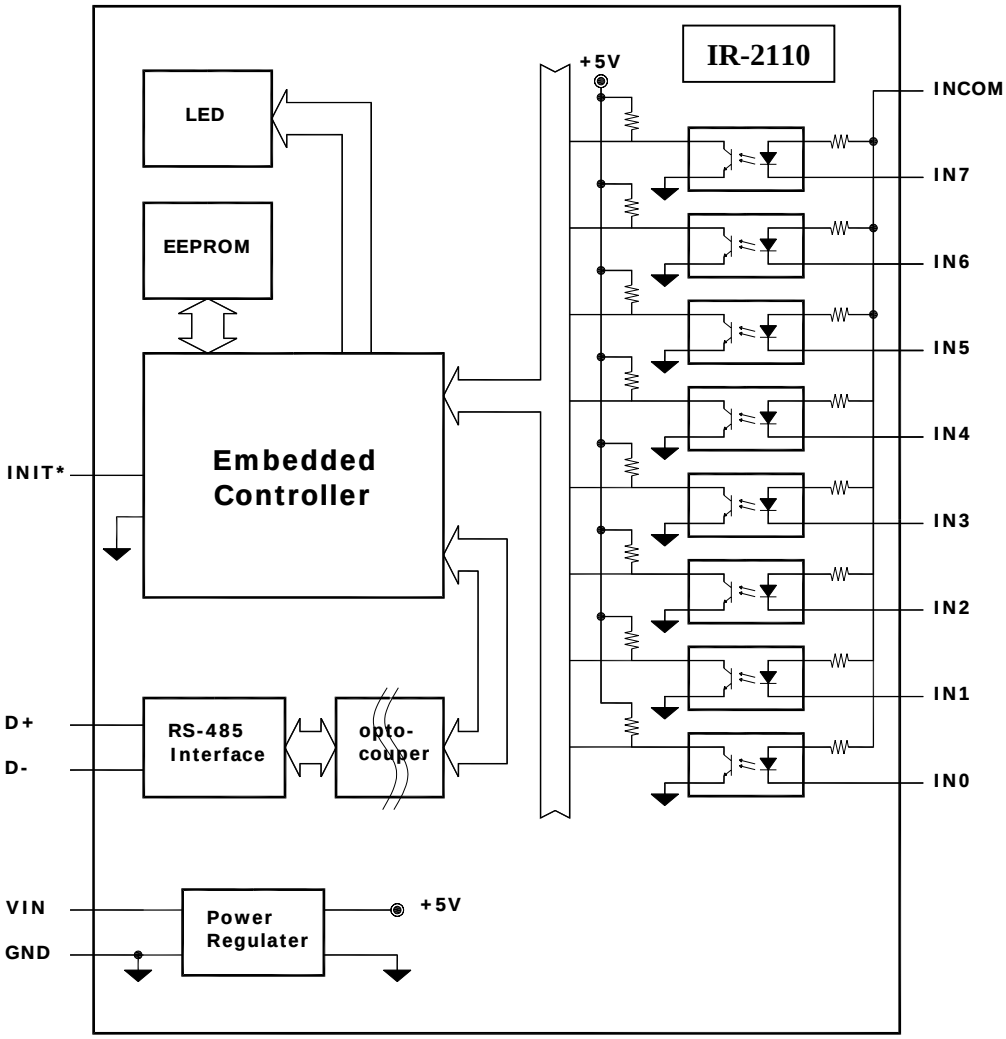


图1.5 IR-2110内部结构框图

1.6 接口与信号

端子名称	说 明		端子名称	说 明
VIN	电源正（+9V～+30VDC）		IN0	开关量输入端0
GND	电源负（地）		IN1	开关量输入端1
D+	RS-485接口D+		IN2	开关量输入端2
D-	RS-485接口D-		IN3	开关量输入端3
INIT*	初始化端子（输入）		INCOM	开关量输入公共端
	(空)		INCOM	开关量输入公共端
	(空)		IN4	开关量输入端4
	(空)		IN5	开关量输入端5
	(空)		IN6	开关量输入端6
	(空)		IN7	开关量输入端7

表1.6 接口与信号

1.7 接线说明

1.7.1 干节点信号输入

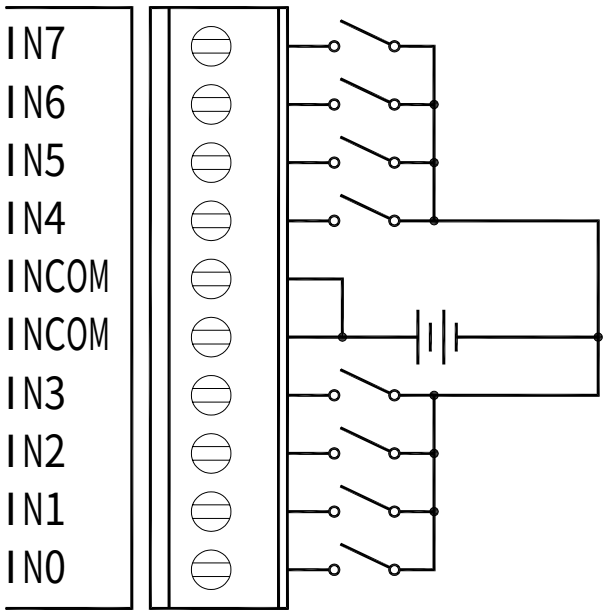


图1.7.1 IR-2110干节点信号输入接线

说明：如图1.7.1所示，当开关闭合时将使逻辑0输入到IR-2110；当开关断开时将使逻辑1输入到IR-2110。上图中INCOM端子建议的电源电压范围为+4V～30V。

1.7.2 TTL/CMOS信号输入

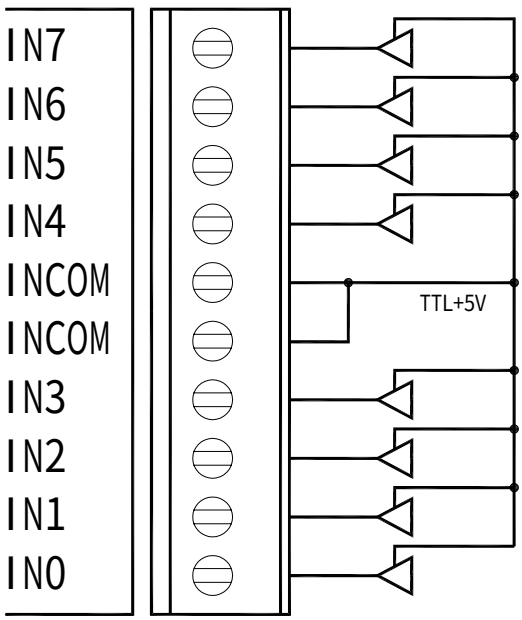


图1.7.2 IR-2110 TTL/CMOS信号输入接线

说明：如图1.7.2所示，当TTL/CMOS输出为低电平时将使逻辑0输入到IR-2110；当TTL/CMOS输出为高电平时将使逻辑1输入到IR-2110。上图中INCOM端的输入电压为+5VDC。

1.7.3 集电极开路(OC门)信号输入

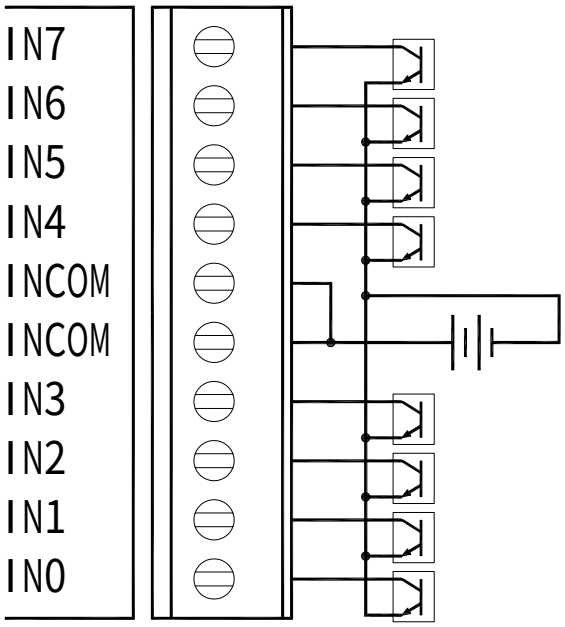


图1.7.3 IR-2110 集电极开路信号输入接线

说明：如图1.7.3所示，当NPN型三极管处于饱和(导通)状态时将使逻辑0输入到IR-2110；当NPN型三极管处于截止状态时将使逻辑1输入到IR-2110。三极管无须上拉电阻，因为在模块内部已有上拉回路。上图中INCOM端子建议的电源电压范围为+4V～30V 。

1.8 LED指示灯

IR-2110模块上有2个LED指示灯，PWR和485。

PWR灯为电源指示灯，红色。当模块供电正常时该灯常亮。

485灯为RS-485通信状态指示灯，绿色。当IR-2110模块收到主机发来的命令或数据时，即便该命令不是发给本模块的，该灯都会闪烁，同时，当模块返回数据给主机时该灯也会闪烁。此外，当模块刚上电或复位时，该指示灯会连续闪烁3次，以指示模块内部的MCU已正常启动。

1.9 应用领域

- ◆ 远程数据采集
- ◆ 过程监控
- ◆ 工业过程控制
- ◆ 能源管理
- ◆ 门禁控制
- ◆ 安防系统
- ◆ 实验室自动化
- ◆ 建筑自动化
- ◆ 产品测试
- ◆ 直接数据控制

2. 使用指导

2.1 系统需求

在使用IR-2110进行开关量的数据采集与监测之前，必须构建一个主从式结构的RS-485网络系统。该系统由一台主机、RS-485网络设备、远程I/O模块以及监视/控制对象、电源、通讯及配置软件等组成。如图2.1所示：

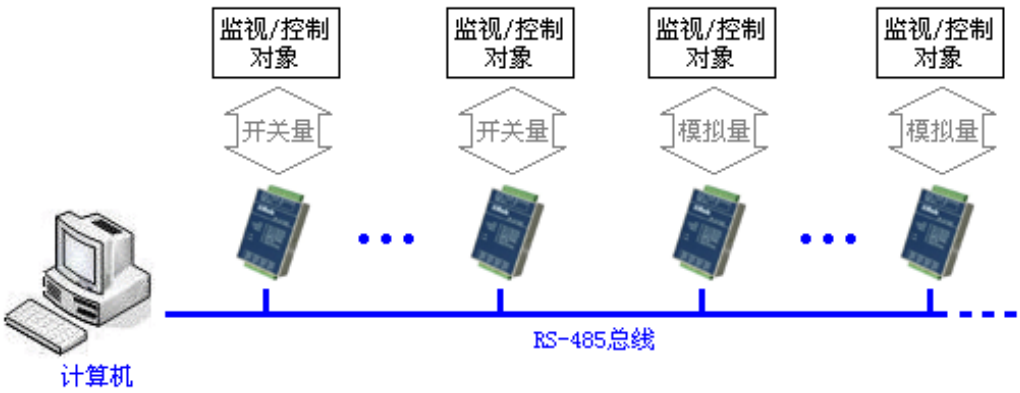


图2.1 基于RS-485网络的远程数据采集与控制系统

系统工作过程描述

采集数据时主机会在RS-485总线上通过轮询的方式逐个发送“读”命令给各控制点的I/O模块，当指定地址的I/O模块收到主机发来的读命令后便会立即返回该模块当前的I/O通道状态给主机，主机收到I/O模块返回的I/O状态信息后会根据这些信息采取相应的动作，如根据控制算法产生新的输出、在屏幕上显示I/O状态信息以及产生报警等。如果主机需要输出新的控制信息给控制对象，则必须发送“写”命令到指定地址的I/O模块。当指定地址的I/O模块收到主机发来的“写”命令时便会根据命令中的信息产生相应的输出。

注：

a).主机与下面的I/O模块之间采用的是单主结构的通信模式。单主就是指系统中只有一个主机，并且数据的传输必须由主机来控制，从机没经过主机的允许是不可以随机发送数据的。在实际应用中，主机发出的轮询数据/命令通过RS-485总线广播到整个RS-485网络上，网络上的所有I/O模块都会收到主机发来的轮询信息，由于轮询的信息中包含I/O模块的地址，虽然所有I/O模块都会收到该数据/命令，但只有地址匹配的I/O模块才会作出响应。

b).挂接于整个RS-485总线上的所有I/O模块都必须有一个唯一的模块地址，RS-485总线上的所有模块地址不能重复，否则系统将无法正常工作。

c). RS-485标准只是描述了物理层的电气标准，并没有规定上层的通信协议，因而不同厂家可以在此基础上设计自己的通信协议。这就意味着不同的厂家针对不同的设备及应用可能会设计很多不同的通信协议，这无疑给用户带来了不便。虽然目前已有基于RS-485的国际标准协议，如MODBUS协议等，但这是非强制性的，各厂家仍然可以设计自己的协议。有一点用户需要遵守的，就是在同一个RS-485网络系统内所有设备必须采用相同的通讯协议。

异特路公司的IR-2000系列远程I/O模块支持双协议：IRASCII协议与ModbusRTU协议。其中IRASCII协议与研华公司的亚当协议以及弘格公司的DCON协议相兼容，其通用指令集兼容于NuDAM、ADAM等模块；而ModbusRTU协议是当前工控领域里使用最为广泛的协议。同时支持以上两种协议无疑给产品带来了最广泛的适用性。

2.1.1 主机计算机

在基于RS-485网络的采集/控制系统中，主机是最重要也是最核心的部分,它是整个系统的控制中心。其主要作用是监视/采集整个控制系统的各种变量(模拟量、开关量等)，并根据控制算法产生需要的控制输出(模拟量、开关量等)。

由于主机计算机在系统中的重要作用，实际应用中我们建议采用标准的工控机作为系统主机(比如一台标准的IBM PC/AT兼容计算机)。毫无疑问，相对于普通PC机而言，工控机具备更高的稳定性和更高的可靠性。同时，该主机必须具备一个标准RS-232接口或RS-485接口，并可通过该接口发送或接收串行数据。

当主机只有RS-232接口时，这时就需要使用一个RS-232转RS-485转换器将RS-232信号转换成RS-485信号。此时可选用异特路公司的有源光电隔离转换器IR-1112B，该转换器可提供光电隔离保护，可有效保护您的设备。

2.1.2 RS-485网络设备

RS-485网络设备涉及RS-485网络的所有环节，包括通信电缆以及转换设备(可选)、中继设备等(可选)等。

通信电缆

虽然RS-485采用差分数据传输模式，具有很强的抗干扰能力，一般采用普通的双绞线即可满足要求，但在条件允许的情况下，我们仍然建议采用屏蔽双绞线。

转换设备(可选)

如上一节所述，当主机只有RS-232接口时，就需要一个RS-232转RS-485的转换器。实际的转换器在实现转换功能的同时还具备光电隔离保护功能，如异特路公司的IR-1112B(见图2.1.2a)。如果主机不具备RS-232接口，也不具备RS-485接口，但具备其它接口如USB接口，则可以选择USB转RS-485转换器，如异特路公司的IR-1401D，该转换器可通过计算机的USB口扩展出一个虚拟串行口，该串行口的物理接口可同时提供RS-232/422/485三种接口，并且具备光电隔离保护功能。同样可实现USB扩展RS-485接口功能的还有其它产品如IR-1401CG等。以上这些转换器产品都是针对工业应用环境要求高安全性和可靠性而专门设计的。

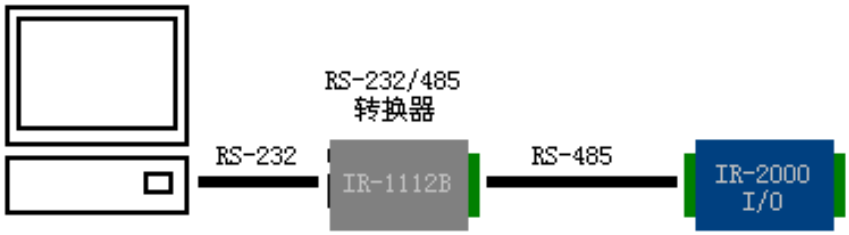


图2.1.2a 主机通过RS-232/485转换器与I/O模块相连

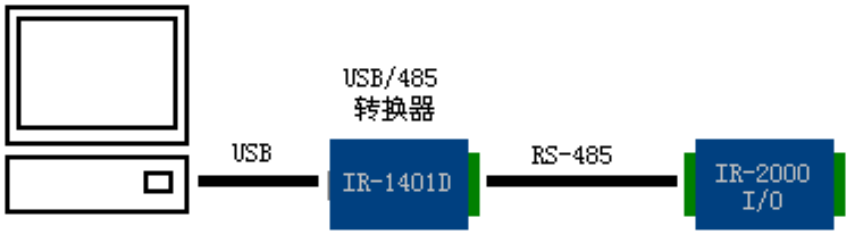


图2.1.2b 主机通过USB/485转换器与I/O模块相连

中继设备(可选)

RS-485的标准规定了最远传输距离为1200米，虽然在通信速率比较低以及挂接的设备数量不多的情况下可传输更远的距离，但有些应用场合下仍需要专门的设备延长RS-485的通讯距离。此时，最常采用的是RS-485中继器，如异特路公司的IR-1301S/D，该产品在实现RS-485信号中继的同时，还具备信号的光电隔离保护功能，可更有效的保护您的设备。理论上说，每挂接一个RS-485中继续，即可使RS-485通讯距离再延长1200米，并且RS-485总线上可挂接的中继续器的数量在理论上是没有限制的。另一方面，每增加一个中继器可使RS-485系统再多挂接32个I/O模块(所有IR-2000系列I/O模块485口均为标准负载，既单独的485接口可最多挂接32个IR-2000模块)。虽然可通过不断挂接中继器的方法来增加I/O模块的数量，但IR-2000系列I/O模块的通信协议规定了最多挂接I/O模块的数量。当协议为IRASCII时最多不能超过256个模块；当协议为ModbusRTU协议时最多不能超过247个模块。

在有些场合要求RS-485的通讯距离更远，并且线路经过的区域无法(或不方便)接中继器时，此时可采用光纤转RS-485转换器，如异特路公司的IR-1513S，可实现将RS-485信号转换成光信号经过单模光纤传输。一对这样的转换器最远可将RS-485信号延长20公里。

2.1.3 远程I/O模块

这里的远程I/O模块既指异特路公司的IR-2000系列远程数据采集/控制模块。根据现场要采集的数据种类以及控制对象接收的控制信号的不同，IR-2000系列远程I/O模块分为数字量I/O模块和模拟量I/O模块两大类。每个大类中包括许多种产品型号，具体内容请参见《异特路公司产品选型手册》。

2.1.4 监视与控制对象

监视与控制对象并不是本说明书的重点，因此这里只说明常见的监视与控制对象相关的信号类型，这些信号主要分数字量和模拟量两大类。常见的数字量有开关量、脉冲量、频率、计数等；而常见的模拟量如温度、流量、压力、湿度、液位、行程、速率、电压、电流等。而这些模拟量都需要转换成能够直接被I/O模块接收的标准信号，这些信号主要有电压(如：0～5V、0～10V、-5V～+5V、-10V～+10V等)和电流(如：4～20mA)信号。

2.1.5 电源

在应用于工业环境的情况下，IR-2000系列模块被设计成支持标准的工业24VDC非稳压电源。在使用直流电源供电时要保证电压范围在+9V～+30VDC范围内。当供电范围在+9V～+30VDC范围内时，电压纹波必须保持在4V的峰-峰值以内。在模块的连接端子处都标有电源接口的定义。当远程供电时，必须考虑直流电源的压降效应。

所有IR-2000系列模块内部都集成了输入电压范围在+9V～+30VDC的高效开关电源。以IR-2110为例，IR-2110的功耗为1.1W，假设用户的电源为24VDC，且模块与电源之间的距离不足以造成明显的压降，若要使模块正常工作，那么该24VDC电源所具备的最大输出电流不得低于1.1W/24V=0.046A。

当系统的规模较小时，可通过壁挂电源通过电源总线给各I/O模块供电。当系统规模较大或通讯距离较远时，建议每个模块通过本地的电源供电，这样会使系统工作更可靠。这些便宜的设备可以很容易的从电子零售店买到。

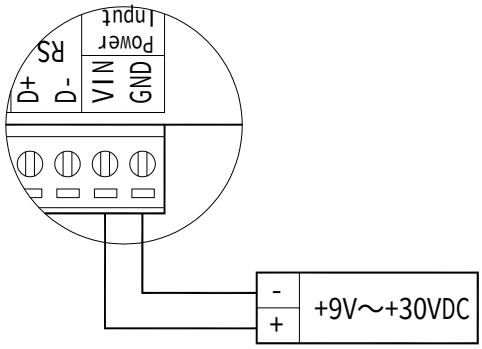


图2.1.5 IR-2110的电源接线

2.2 测试与控制功能软件IR2000Utility

IR-2110模块支持双协议：IRASCII协议与ModbusRTU协议。无论采用何种协议，在实际使用时都需要用户根据模块所选择的通信协议自行编程来实现对模块的控制。编程工具则可以选用Visual C++、C#、Visual Basic、Delphi等，具体语言及工具的选择由用户自己决定。虽然使用的编程工具和语言不同，但基本原理都是类似的，都是通过程序操纵计算机串口，通过串口发送和接收数据来完成对模块的设置和I/O控制。而发送或接收的数据的格式则要严格依照模块的通信协议。

IR-2000系列I/O模块包含一套可以对模块进行设置和测试的功能软件IR2000Utility。该软件提供了一个基于菜单的用户设置和操作界面，通过这个软件，用户可以很容易的对IR-2000模块进行设置和I/O控制；同时，该软件还提供了一个基于命令行的终端程序Terminal。通过该终端程序用户可直接输入IR-2000系列I/O模块的命令，同样可以实现对模块的设置和I/O控制。下面简单介绍IR2000Utility软件的用法。

2.2.1 运行程序

假设计算机的串口COM1已经连接了RS-232/485转换器，并且该转换器的485口已经与IR-2110模块正确连接。此时，在该计算机上运行IR2000Utility程序，初始界面如下图所示：

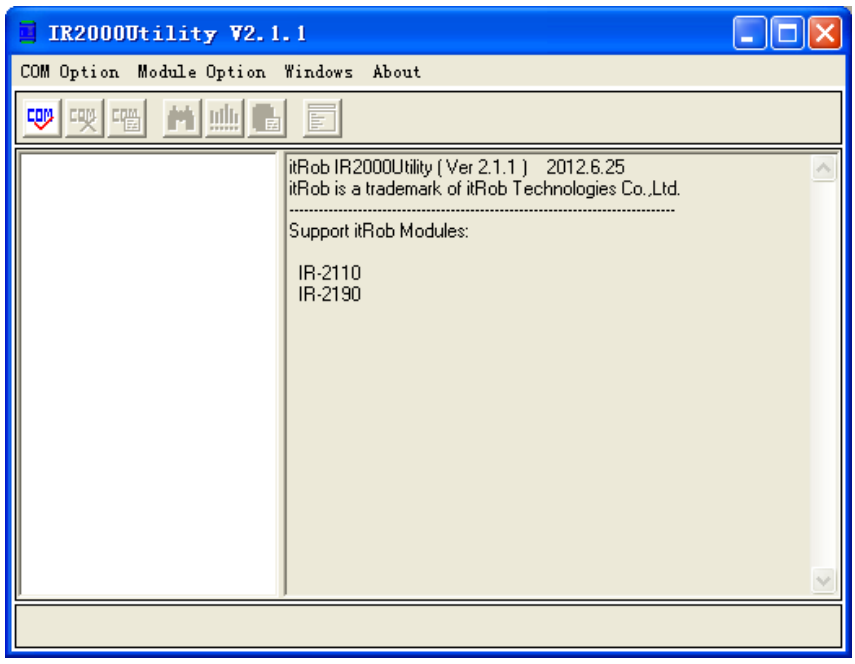


图2.2.1.a IR2000Utility程序主窗口界面

2.2.2 打开串口

在菜单项中选择COM Open或点击工具栏的 按钮，弹出串口设置对话框，如下图所示：

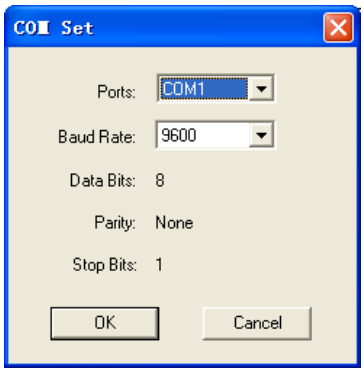


图2.2.2.a

选择串口号为COM1，通讯波特率为9600bps(该串口的数据位、奇偶校验位在这里是不能设置的，IR2000Utility会自动将数据位设成8位、奇偶校验方式设为无校验状态、1位停止位)，点击OK按钮，打开串口COM1。

成功打开串口后，IR2000Utility程序窗口下部的状态拦会显示当前已打开串口的相关信息，如图（2.2.2.b）所示：

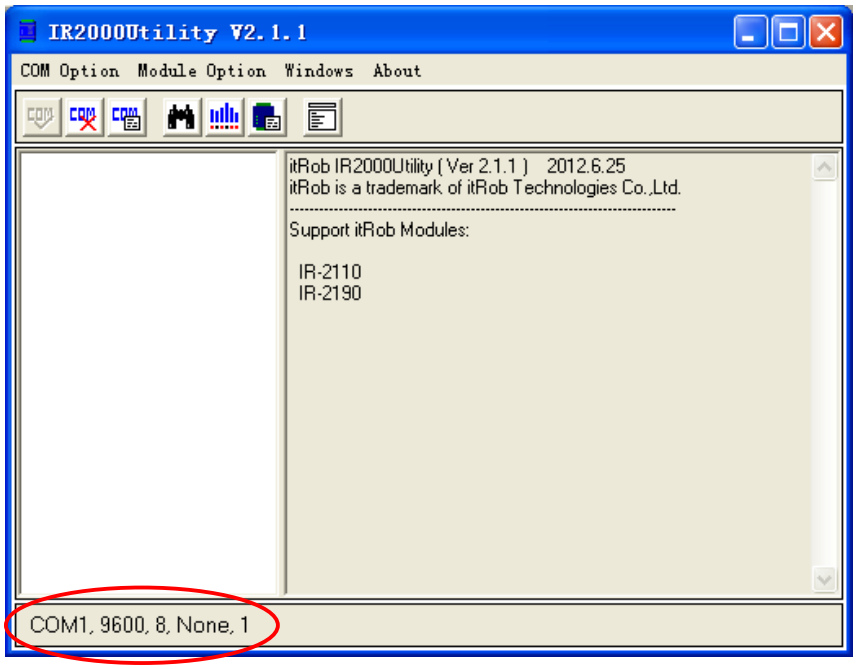


图2.2.2.b

2.2.3 搜索模块

此时在菜单中选择Module Search项或在工具栏中点击 按钮，弹出模块搜索设置对话框。如下图所示：

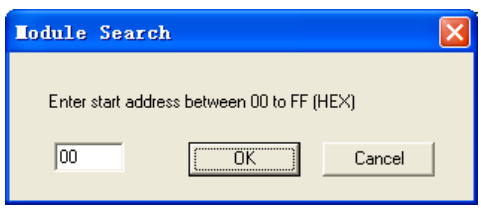


图2.2.3.a

在对话框中输入模块搜索的起始地址(两位十六进制数)，点击OK按钮，程序便从指定的起始地址处采用地址递增的方式在指定的串口(这里是COM1口)搜索模块，直到地址到达FF为止。搜索模块时会弹出模块搜索信息对话框，如下图所示：

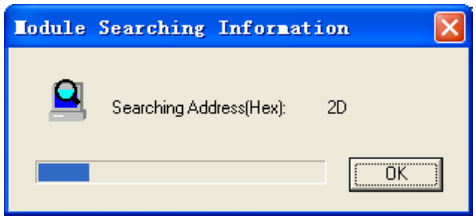


图2.2.3.b

搜索结果会在IR2000Utility程序工作区的左侧模块搜索列表中显示出来。如下图所示：

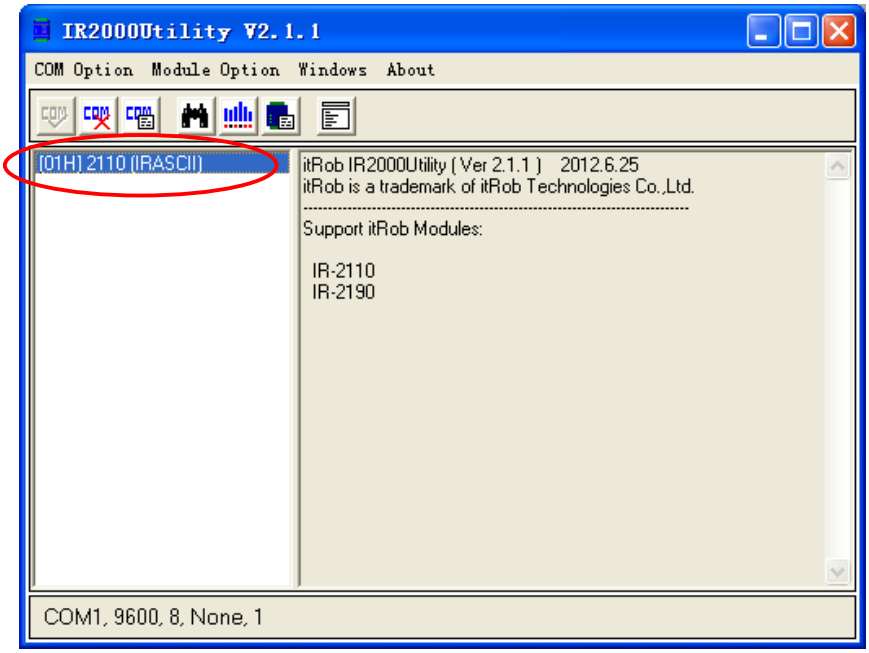


图2.2.3.c

该搜索列表中显示了已搜索到模块的相关信息，如：模块地址、模块型号以及该模块所使用的协议等信息。

如图(2.2.3.c)可知，搜索到地址为01的模块，型号为IR-2110，该模块所采用的协议为IRASCII协议(无检验和)。

2.2.4 IO控制

用鼠标左键双击该列表项，或先单击该表项使该搜索表项处于选中状态，然后再选择IO Control菜单(或点击 按钮)，便会弹出IR-2110模块的IO控制窗口，如下图所示：

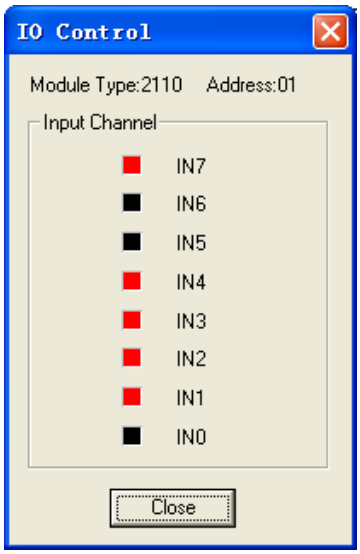


图2.2.4.a

对于IR-2110模块来说，其对应的IO控制对话框中除了显示模块型号、模块地址外，还显示IR-2110的8个输入通道的当前状态，如图(2.2.4.a)所示。

在Input Channel栏中显示的是8个输入通道的状态，红色代表该通道的当前状态为ON(高电平)状态；黑色代表该通道的当前状态为OFF(底电平)状态。如果此时IR-2110的输入通道状态发生变化，则通道的状态变化会立即在该窗口显示出来。如图可知，输入通道IN1、IN2、IN3、IN4、IN7的当前状态为ON，而输入通道IN0、IN5、IN6的当前状态为OFF。

2.2.5 设置通信参数

下面介绍如何通过IR2000Utility设置IR-2110的配置参数。

如图(2.2.3.c)所示，选择菜单中的Module Settings项或点击工具栏中的 按钮，将弹出模块设置对话框，如图(2.2.5.a)所示。

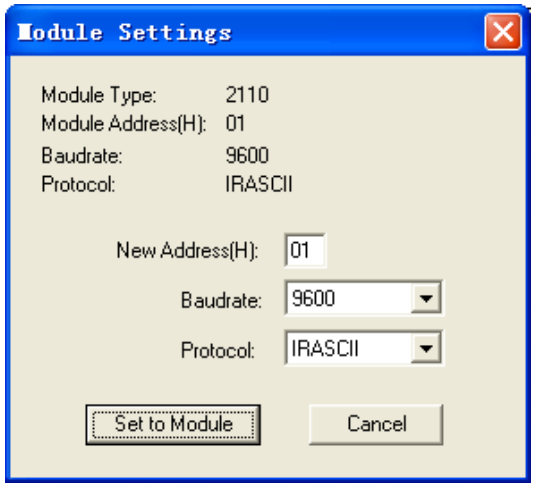


图2.2.5.a

通过该对话框可设置IR-2110模块的地址、波特率以及通信协议。

该对话框左上角显示的是模块当前的配置信息，有地址、波特率、协议。对话框下面是用户要设置的相应配置参数。

当用户设置好新的模块配置参数后，点击Set to Module按钮即可将新的配置参数写入IR-2110模块。

说明：协议下拉列表选项中IRASCII表示不带检验和的IRASCII协议；IRASCII-CHK表示带检验和的IRASCII协议；ModbusRTU表示ModbusRTU协议。

注意！如果用户试图更改模块的波特率和协议设置，必须使INIT*端子与GND短接。否则，IR2000Utility会提示设置失败。

关于IR-2110模块的通信参数设置有如下几点需要注意：

1、如果用户忘记了模块当前的通信参数设置，可以采用如下两种方法重新与模块建立连接：

方法1：用IR2000Utility软件在所有的波特率段全部搜索一遍(地址从00～FF)，如果模块正常则一定可以搜索到模块。但该方法显然很麻烦而且可能会用很长时间才能搜索到模块。

方法2：将模块的INIT*与GND端子短接，然后重新启动模块，此时模块会自动进入初始化状态，即：波特率9600bps、地址00、协议为IRASCII协议。此时用IR2000Utility软件可以立即搜索到模块并建立连接，然后在通过模块设置（Module Settings）对话框重新设置模块的通信参数。

2、在设置模块地址时要注意在IRASCII协议下有效模块地址范围为00～FF(十六进制)共256个地址，而在ModbusRTU协议下模块的有效地址范围为01～F7(十六进制)共247个地址。因此用户在将模块协议设置成ModbusRTU协议时，一定要同时检查地址是否符合ModbusRTU协议规定的地址范围，否则会返回设置失败的信息。

3、在用模块设置(Module Settings)对话框设置模块通信参数时，如果模块当前协议为IRASCII(或IRASCII_CHK)协议，而此时用户只想更改模块地址(波特率与协议都不变)的话，则不需要将模块的INIT*端子与GND端子短接就可成功设置；但如果模块当前协议为ModbusRTU协议，则必须先将模块的INIT*端子与GND端子短接再点击“Set to Module”按钮才可成功设置。原因是在IRASCII(或IRASCII_CHK)协议下模块通信参数设置命令只有一条，既：%AANNTTCCFF，该命令允许模块在INIT*端子与GND端子未短接且波特率和协议不变的情况下只设置地址且不返回错误信息。但在ModbusRTU协议下则需要2条命令完成，既：模块地址设置命令(功能码46、子功能码04)和波特率协议设置命令(功能码46、子功能码06)。模块地址设置命令不要求INIT*与GND短接，但波特率协议设置命令要求INIT*与GND必须短接且无论波特率或协议是否有变化。由于在用户点击“Set to Module”按钮后IR2000Utility会先发送模块地址设置命令先更改模块地址，然后再发送波特率协议设置命令，而后一条命令要求INIT*与GND端子必须短接，因此，如果此时INIT*与GND端子未短接则软件会提示如下设置结果信息：

图2.2.5.b

图2.2.5.b中的设置结果信息消息框显示模块新地址设置成功，但模块波特率和协议设置失败，模块返回异常码0x04(参见功能码46、子功能码06)。由于波特率和协议设置失败，因此这两个参数并没有改变。

4、若模块当前协议为ModbusRTU，如果此时用户想将模块设置成IRASCII (或IRASCII_CHK)协议，且新地址设置为00或F8～FF(在ModbusRTU协议下非法但在IRASCII协议下合法的地址)的话，必须按照如下步骤设置：

步骤一：将INIT*与GND短接，保持地址不变，先成功设置新的波特率和协议，断开INIT*与GND后重新启动模块。

步骤二：按照新的波特率重新搜索到模块并建立连接后再将模块地址设置成新的地址(INIT*与GND不必短接)。

5、一旦用户成功将模块设置成某种协议并重新启动模块后(INIT*与GND断开)，该模块将不再接收其它协议的命令。

2.2.6 终端的使用

IR2000Utility中还提供了使用户可以直接发送命令给模块的途径。终端(Terminal)对话框即可完成这个功能。

选择菜单的Terminal选项或点击工具栏的  按钮，即可弹出终端对话框。在Terminal窗口中，用户可以选择所要采用的通信协议(IRASCII协议、带检验和的IRASCII协议或ModbusRTU协议)，如图(2.2.6.a)、(2.2.6.b)、(2.2.6.c)所示。

选定协议后，用户可在Command Input框中输入要发送的命令。对于不同的协议会有不同的追加选项，分别描述如下：

◆选择IRASCII协议，则终端窗口会自动显示追加(Cr)字符的复选框，如图2.2.6.a所示。

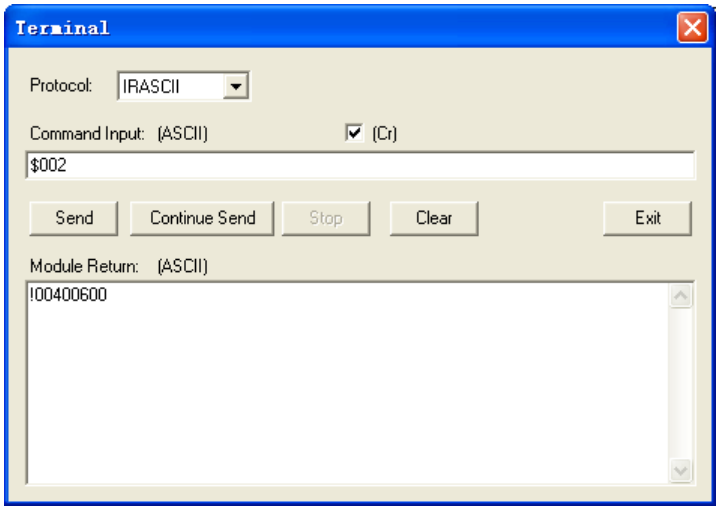


图2.2.6.a (IRASCII协议时的命令与返回信息)

后面的复选框(Cr)表示是否在命令后添加(Cr)字符。由于IRASCII协议的大多数命令后面都需要追加(Cr)字符，因此可以选中该复选框。图2.2.6.a中输入的命令为读地址为00的模块的配置信息命令,输入完命令后点击发送(Send)按钮将该命令发送到模块。正常情况下模块会立即返回配置信息。如图2.2.6.a可知，模块返回的信息为!00400600，该信息说明地址为00的模块的模块代码为40，波特率为9600bps，采用IRASCII协议(无检验和)。

当选择协议为IRASCII协议时，接收数据会自动以ASCII码格式显示。

◆选择带检验和的IRASCII协议(IRASCII_CHK)，则终端窗口会自动显示追加CHK的复选框和追加(Cr)字符的复选框，如图2.2.6.b所示。

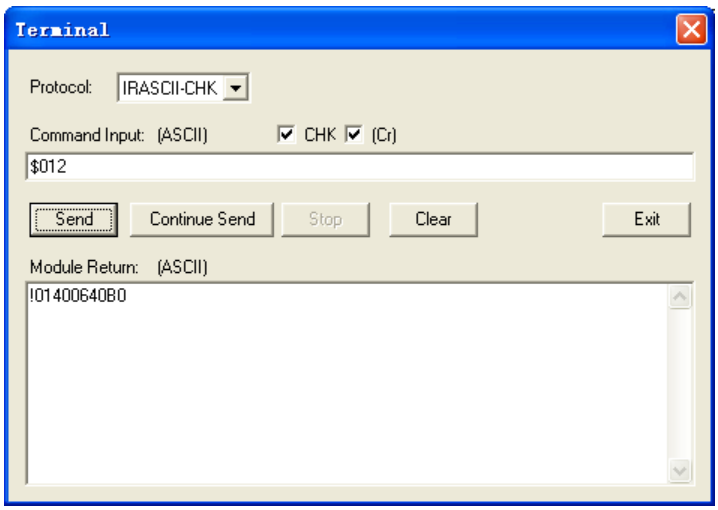


图2.2.6.b (IRASCII协议(带检验和)时的命令与返回信息)

若用户选择了CHK复选框，则终端程序会自动在用户输入的命令后追加检验和字段，而不必人工计算和追加，非常方便。图2.2.6.b中输入的命令为读地址为01的模块的配置信息命令,输入完命令后点击发送(Send)按钮将该命令发送到模块。正常情况下模块会立即返回配置信息。如图2.2.6.b可知，模块返回的信息为!01400640B0，该信息说明地址为01的模块的模块代码为40，波特率为9600bps，采用IRASCII协议(带检验和)。

如果CHK复选框未选中，则用户需要自行在输入的命令后添加检验和字段。

当选择协议为IRASCII_CHK协议时，接收数据会自动以ASCII码格式显示。

◆选择ModbusRTU协议，则终端窗口会自动显示追加CRC复选框，如图2.2.6.c所示。

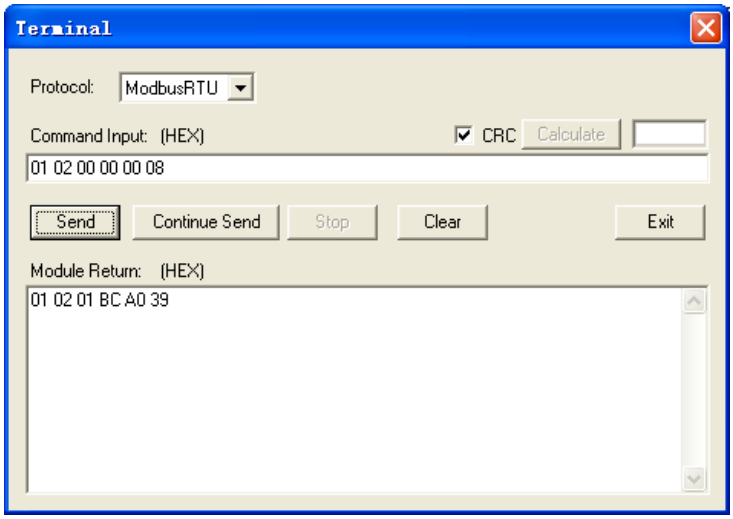


图2.2.6.c（ModbusRTU协议时的命令与返回信息）

如用户选中了CRC复选框按钮，则终端会在用户输入的十六进制格式的ModbusRTU命令后自动追加CRC。

如图2.2.6.c所示，用户在命令输入框内输入了读模块输入通道状态命令(模块地址01、功能码02、起始通道号0000、通道数量0008)后模块返回如下信息：01 02 01 BC A0 39 ，由返回信息可知模块的8个输入通道IN7、IN6、IN5、IN4、IN3、IN2、IN1、IN0的输入状态分别为ON、OFF、ON、ON、ON、ON、OFF、OFF。

注意！这里用户输入的命令中不包含CRC，因为CRC复选框为选中状态(自动追加)，而下方的模块返回信息框中则包含模块实际返回的带CRC的信息。

如果CRC复选框没有选中则终端不会在用户输入的ModbusRTU命令后自动追加CRC 。但此时在CRC复选框右侧会显示手动计算CRC的按钮“Calculate”。此时，用户点击该按钮后终端程序会计算用户输入的ModbusRTU命令的CRC值并在按钮右侧的文本框显示计算结果，如图2.2.6.d所示。

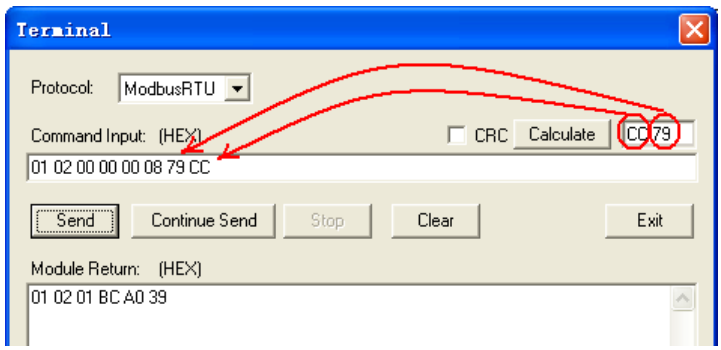


图2.2.6.d

如图2.2.6.d所示，当用户输入相同的读模块输入通道状态命令(01 02 00 00 00 08)后，点击Calculate按钮后的计算结果为CC79。此时，用户需要手工将计算的CRC添加到输入命令的末尾。**但要注意！ModbusRTU协议的命令格式规定CRC的低8位在前，高8为在后。**

“Calculate”按钮给用户提供了一条计算十六进制字节串的CRC值的途径，这无疑给基于ModbusRTU协议编程的工程技术人员带来了方便。

关于IR-2110模块的ModbusRTU协议命令的详细信息请参见4.4节ModbusRTU协议命令详解。

关于IR2000Utilty软件的使用参见《IR2000Utilty使用手册》。

2.3 用串口测试软件与模块通讯

除了IR2000Utilty软件外，用户还可以通过串口测试软件来与模块通讯，如异特路公司的IRCOMM Ver2.0、串口调试助手等。这些软件可以很容易的从网络上免费下载，而且种类众多。如图2.3.a所示为IRCOMM软件的发送窗口，该窗口的HEX框中已输入了读模块当前设置命令$002(由于命令后必须有Cr字符，所以该命令必须在HEX框中输入)，而图2.3.b为IRCOMM的主窗口接收到模块的返回信息，该信息说明地址为00H的模块的模块类型代码为40，波特率为9600bps，模块的通信协议为IRASCII协议，并且通信不启用检验和。

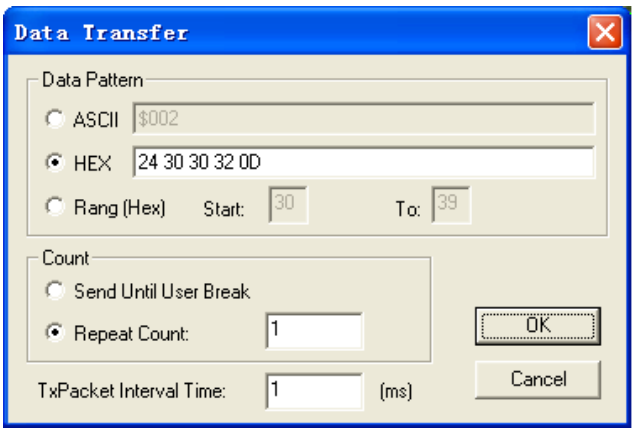


图2.3.a 数据发送窗口

图2.3.b 数据接收窗口

2.4 通信参数

IR-2110模块的用户可设置参数总共有4项，如下表(表2.4)：

编号	设置项名称	设置条件	使用命令	
1	地址	（无）	IRASCII	%AANNTTCCFF
			ModbusRTU	功能码46，子功能码04
2	波特率	INIT*与GND短接	IRASCII	%AANNTTCCFF
			ModbusRTU	功能码46，子功能码06
3	通信协议	INIT*与GND短接	IRASCII	%AANNTTCCFF
			ModbusRTU	功能码46，子功能码06

表2.4 IR-2110的可设置项列表

地址、波特率、通信协议都是用户必须的设置项。由于IR-2110为8路隔离开关量输入模块，其没有输出功能，因此IR-2110没有安全值输出功能，也就没有通信看门狗功能。

下面具体说明以上3个设置项。

2.4.1 模块地址

IR-2110模块的地址范围在不同的协议下略有差别。当模块协议为IRASCI协议时模块有效地址范围为00～FF(十六进制)共256个地址；当协议为ModbusRTU协议时模块的有效地址范围为01～F7(十六进制)共247个地址。当协议为ModbusRTU协议时，地址00规定为广播地址(如同步采样命令：地址00、功能码46、子功能码18)，而地址F8～FF(十六进制)为ModbusRTU协议标准规定的保留地址。用户不能将00和F8～FF范围内的地址设置到IR-2110模块，否则模块会返回错误信息。

另外，如果用户只更改模块地址的话，则不必在INIT*状态(INIT*端子与GND端子短路)下进行，并且地址更改后不必重新启动模块即可立即生效。

（关于在不同的协议下如何设置模块的地址请参考对应协议下的相关设置命令）

注！如果一个RS-485网络中挂接了多个IO模块，若要使系统正常工作必须保证所有模块的地址不能重复，既不允许有地址相同的模块挂在同一个RS-485网络上。

协议	IRASCII	ModbusRTU
地址范围	00～FF（共256个有效地址）	01～F7（共247个有效地址）

表2.4.1 IR-2110 在不同协议下的地址空间对照表

2.4.2 波特率

IR-2110模块的通信口为标准的RS-485接口，支持的通信波特率如下表(表2.4.2)所示：

代码	03	04	05	06	07	08	09	0A
波特率	1200	2400	4800	9600	19200	38400	57600	115200

表2.4.2 IR-2110 通信波特率(单位bps)

说明：上表中的代码对应相应的波特率数值。在IRASCII协议下对应%AANNTTCCFF命令中的CC字段；在ModbusRTU协议下则对应设置命令(功能码46、子功能码06)中的04号字节。

IR-2110的波特率的出厂值为9600bps。用户若要更改模块波特率必须先使模块的INIT*端子与GND短接，然后用相应协议下的设置命令设置模块的波特率。当用户成功更改模块波特率后，新设定的波特率不会立即生效，要想使新波特率生效，用户必须先断开INIT*端子与GND端子，然后重新启动IR-2110模块。

若要使控制主机(计算机)与IR-2110模块通信，必须保证主机与模块具有相同的波特率。如果用户忘记了IR-2110模块的当前波特率值，可以通过如下方法重新设置模块波特率：首先将模块的INIT*端子与GND端子短接，然后重新启动模块，这时IR-2110将进入INIT*状态（初始化状态），模块在INIT*状态下所有的设置参数都将载入出厂默认值，既：地址为00、波特率为9600bps、协议为IRASCII协议(无检验和)。基于以上设置，用户即可通过%AANNTTCCFF命令来重新设置模块的波特率。

注！IR-2110的波特率必须在INIT*状态下才可更改。

2.4.3 通信协议

IR-2110支持2种通信协议---IRASCII协议和ModbusRTU协议。其中，IRASCII协议又分2种---不带检验和与带检验和。因此，用户在IR2000Utility程序的模块设置对话框中会看到在协议下拉列表中有3个选项，其中IRASCII就是指不带检验和的IRASCII协议，IRASCII_CHK就是指带检验和的IRASCII协议。如表2.4.3所示：

协议名称	IRASCII（不带检验和）	IRASCII（带检验和）	ModbusRTU
下拉列表选项	IRASCII	IRASCII_CHK	ModbusRTU

表2.4.3 协议名称对照表

无论何种协议，都采用RS-485通信。IR-2110模块的通讯口为标准的RS-485接口，其数据格式符合串口通讯的标准数据格式，既：1位起始位，8位数据位，无奇偶校验，1/2位停止位。IR-2110的校验方式不采用面向每个字节的奇偶校验方式，而是采用面向整个数据包(字节串)的校验方式。在IRASCII协议下为检验和方式；ModbusRTU协议下为CRC方式。

关于IRASCII协议如何计算检验和的方法参见附录B。
关于ModbusRTU协议的CRC码的计算方法参见附录G。

2.5 启动过程及参数载入

IR-2110的启动过程主要有2种：正常启动与INIT*（初始化）启动。

2.5.1 正常启动

正常启动是指IR-2110在启动时INIT*引脚与GND引脚未短接的启动方式，此时IR-2110的所有可设置参数全部从位于其内部的EEPROM中载入。用户之前设置并保存的数据就存放于该EEPROM中。该方式也是IR-2110正常应用时的启动方式。

2.5.2 INIT*启动

INIT*启动也叫初始化启动，是指IR-2110在启动时INIT*引脚与GND引脚短接的启动方式，此时IR-2110的所有可设置参数全部自动载入出厂默认值，而不是从EEPROM中载入用户的设置参数。 INIT*启动后的载入的默认配置参数为波特率为9600、地址为00、协议为IRASCII(无检验和) 。该启动方式的主要用处是当用户忘记模块的设置参数时提供一个恢复出厂默认值的途径，以方便用户快速的与模块建立连接，以对模块进行新的设置。

IR-2110的启动流程与参数载入流程如下图(图2.5.2)所示：

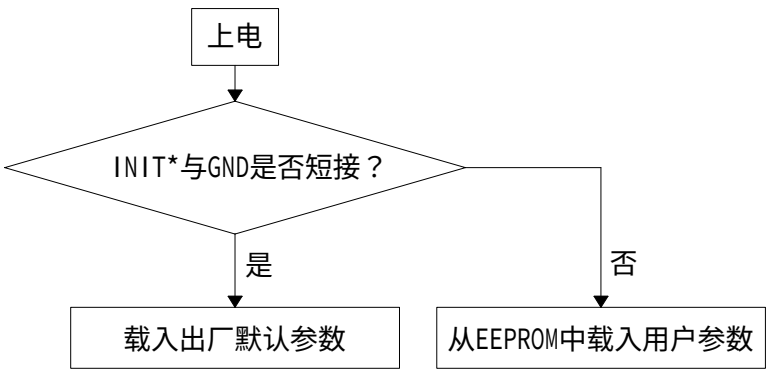


图2.5.2 IR-2110 启动过程及参数载入流程图

2.6 安装说明

IR-2000系列模块可方便的安装于任何面板、支架和标准DIN导轨上。用户可以根据应用需要选择安装方式,通过搭配不同的配件(这些配件需要用户在购买产品时指定),来适应不同的安装方式。

有2种可选的配件：塑料DIN导轨配件和金属壁挂配件。塑料DIN导轨配件可使模块轻松卡装于标准DIN导轨上(如图2.6.a所示)；搭配金属壁挂配件可使模块安装于面板和墙壁(如图2.6.b所示)。

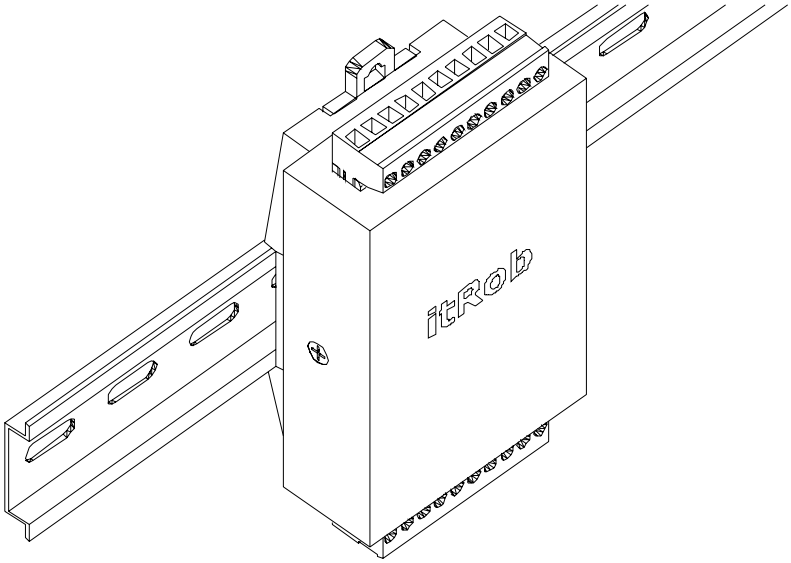


图2.6.a DIN导轨安装示意图

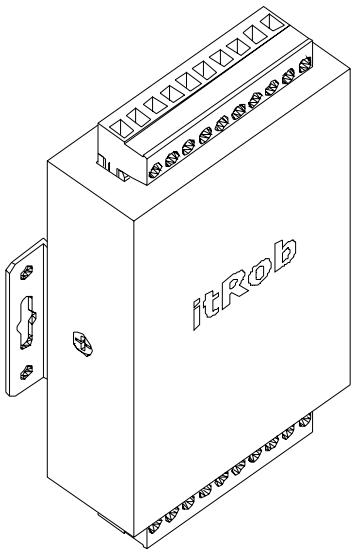


图2.6.b 面板安装示意图

以上两种安装配件都通过螺丝固定于模块背面的两个螺丝固定孔上，用户可自行安装及拆卸。

3. IRASCII协议指令系统

3.1 IRASCII协议命令列表

IR-2110的IRASCII协议指令系统主要由10条命令组成，如下表所示：

命令语法	命令名称	命令描述	页码
$AA2	读模块配置信息命令	读模块的配置信息，这些信息包括模块代码、波特率、协议	23
%AANNTTCCFF	写模块配置信息命令	设置模块的新地址、波特率、协议	24
$AAM	读模块名称命令	读指定地址模块的名称/型号	25
$AAF	读模块硬件版本号命令	读指定地址模块的硬件版本号	26
$AA6	读I/O通道命令	读指定地址模块的当前I/O通道的状态	27
#**	同步采样命令	命令所有已连接的模块将当前的I/O通道状态保存在内部的同步采样寄存器中	28
$AA4	读同步采样命令	读指定地址模块的同步采样数据，这些数据是模块之前在执行#**命令时保存于内部的同步采样寄存器中的数据	29
$AA5	读模块复位状态	读指定地址模块的复位状态信息	30
$AAL0	读输入脉冲锁存命令	读指定地址模块的输入通道脉冲锁存寄存器值	31
$AAC	清除输入脉冲锁存	将指定地址模块的输入通道脉冲锁存寄存器清零	32

表3.3 IR-2110 IRASCII协议命令列表

3.2 命令详解

$AA2

名称：读指定地址模块的通信配置参数命令。

说明：该命令执行成功后，指定地址的I/O模块会返回相应的通信配置参数。

语法：$AA2(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

2为配置状态命令代码。

(Cr)为终止字符，即回车(0DH)。

该命令请求地址为AA的I/O模块返回当前的通信配置参数。

返回：!AATTCCFF(Cr)命令有效时。

若模块检测到一个语法错误、非法命令或通讯错误，或命令指定的地址不匹配，那么模块将不会返回任何信息。

!为界定字符，该字符指示命令是有效的。

AA为返回信息的模块地址(2个字符的十六进制数)，取值范围为00～FF。

TT为代码类型，IR-2110的代码类型始终为40。

CC为波特率代码，如下表所示(单位bps)：

代码CC	03	04	05	06	07	08	09	0A
波特率	1200	2400	4800	9600	19200	38400	57600	115200

FF为模块协议字，为两个字符十六进制数，其对应的8位二进制数中的第2位和第6位描述了检验和和通讯协议状态信息，如下表所示：

F				F			
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

其中bit0、bit1、bit3、bit4、bit5、bit7保留不用为0。

bit2为协议位(0为IRASCII协议；1为MODBUS协议) 。

bit6为IRASCII协议的检验和选择位(0为不带检验和；1为带检验和)。

(Cr)为终止字符，即回车(0DH)。

例1(无检验和)：

命令：$582(Cr)

返回：!58400600(Cr)

该命令读地址为58H的模块的当前通信配置参数。

从模块的返回信息可知，该模块的地址为58H，代码类型为40，波特率为9600bps，通信协议为IRASCII(无检验和)。

例2(带检验和)：

命令：$122B9(Cr)

返回：!12400640B2(Cr)

该命令读地址为12H的模块的当前通信配置参数。

从模块的返回信息可知，该IO模块的地址为12H，模块代码为40，波特率为9600bps，通信协议为IRASCII(有检验和)。本命令检验和为B9H，返回数据检验和为B2H。

%AANNTTCCFF

名称：模块设置命令

说明：本命令将设置指定地址为AA的模块的新地址(NN)、模块代码(TT=40)、波特率(CC)、以及模块协议字(FF)。

语法：%AANNTTCCFF(Cr)

%为前导符

AA为要配置信息的模块的当前地址(2个字符的十六进制数)，取值范围为00～FF。

NN(00～FF)为设置的模块的新地址(2个字符的十六进制地址)。

TT为将要设置的模块的代码类型，对于开关量IO模块IR-2110来说TT=40(定值)。

CC为指定模块的通讯波特率代码，如下表：

代码CC	03	04	05	06	07	08	09	0A
波特率	1200	2400	4800	9600	19200	38400	57600	115200

FF为模块协议字，为两个字符十六进制数，其对应的8位二进制数中的第2位和第6位描述了检验和和通讯协议状态信息，如下表所示：

F				F			
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

其中bit0、bit1、bit3、bit4、bit5、bit7保留不用应为0。

bit2为协议选择位(0为IRASCII协议；1为MODBUS协议)。

bit6为IRASCII协议的检验和选择位(0为不带检验和；1为带检验和)。

(Cr)为终止字符，即回车(0DH)。

返回：!AA(Cr) 表命令有效,且已成功执行。

?AA(Cr) 表示设置参数非法/无效，或者当试图改变波特率或协议字设置时INT*端子没有与GND相连。另外，即便当INT*端子已经与GND相连的情况下，如果命令指定的模块代码TT有错误(不等于40)、波特率CC不存在(代码错误)或指定的协议字FF试图设置那些未定义的保留位(如：试图设置FF的第5位为1)时，模块都将返回?AA(Cr)。

如果IO模块检测到有语法错误或通讯错误，或指定地址与本地址不匹配时，将不返回任何信息。

!为模块接受到有效命令时响应命令的前导符。

?表示IO模块收到非法/无效命令时返回的前导符。

AA为返回信息的模块地址(2个字符的十六进制数)，取值范围为00～FF。

(Cr)为终止字符，即回车符(0DH)

说明：对于IR-2000系列I/O模块来说，只有地址可以动态(在线)设置。波特率与协议字只有当INT*接地(GND)时才可进行设置。

例1(无检验和)：

命令：%2324400600(Cr)

返回：!24(Cr)

该命令设置地址为23H的模块的新地址为24H，波特率设为9600bps，无检验和，协议为ASCII协议，(模块代码不变)。

从模块的返回信息可知该设置命令设置成功。

例2(有检验和)：

命令：%00004006000F(Cr)

返回：!0081(Cr)

该命令设置地址为00H模块的波特率设为9600bps，协议为IRASCII(无检验和)，地址和模块代码不变。

从模块的返回信息可知该设置命令设置成功。本命令检验和为0FH，返回数据检验和为81H。

$AAM

名称：读模块名称(型号)命令

说明：本命令请求模块返回它的名称(型号)

语法：$AAM(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

M为读模块名称(型号)命令代码。

(Cr)为终止字符，即回车(0DH)。

返回：!AA(Module Name)(Cr) 当命令有效时

若模块检测到一个语法错误或通讯错误，或命令指定的地址不匹配，那么将不会返回任何信息。

!为前导符，该字符指示一个有效的命令。

AA为返回信息的模块地址(2个字符的十六进制数)，取值范围为00～FF。

(Module Name)为地址为AA的模块的名称。对于IR-2110模块来说(Module Name)为2110。

(Cr)为终止字符，即回车(0DH)。

例1(无检验和)：

命令：$12M(Cr)

返回：!122110 (Cr)

该命令指定地址为12H的I/O模块返回它的模块的型号。由返回信息可知模块型号为2110。

例2(带检验和)：

命令：$00MD1(Cr)

返回：!00211045(Cr)

该命令指定地址为00H的I/O模块返回它的模块的型号。由返回信息可知模块型号为2110。

本命令检验和为D1H，返回数据检验和为45H。

$AAF

名称：读模块版本号命令

说明：请求地址为AA的模块返回该模块的版本信息。

语法：$AAF(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

F为读模块版本号命令。

(Cr)为终止字符，即回车(0DH)。

返回：!AA(Version)(Cr)　当命令有效时

若模块检测到一个语法错误或通讯错误，或命令指定的地址不匹配，那么将不会返回任何信息。

!为前导符，该字符指示一个有效的命令。

AA为返回信息的模块地址(2个字符的十六进制数)，取值范围为00～FF。

(Version)为模块的版本代码目前的版本代码为201201。其中2012表示最近一次更新该模块软件版本的年份，01表示子版本号。

(Cr)为终止字符，即回车(0DH)。

说明：关于IR-2110模块的版本升级信息请参见Page2(软件版本历史)。

例1(无检验和)：

命令：$58F(Cr)

返回：!58201201(Cr)

该命令指定地址为58H的模块返回它的模块的版本号。

从模块返回信息可知，该模块的硬件版本号为201201。

例2(带检验和)：

命令：$00FCA(Cr)

返回：!00201201A7(Cr)

该命令指定地址为00H的模块返回它的模块的版本号。

从模块返回信息可知，该模块的硬件版本号为201201。

本命令检验和为CAH，返回数据检验和为A7H。

$AA6

名称：读I/O通道命令

说明：该命令指定地址为AA的IR-2110模块返回输入通道的当前状态。

语法：$AA6(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

6为读I/O通道命令代码。

(Cr)为终止字符，即回车符(0DH)。

返回：!00(DI)00(Cr)　当命令有效

如果IR-2110模块检测到一个语法或通信错误，或命令指定地址不匹配，IR-2110将不返回任何信息。

(说明：不同的I/O模块对于该命令的返回信息的格式可能不同，本说明书在未作特别说明的前提下所有格式都是针对IR-2110模块的)

!为前导符，指示一个有效命令已被接收。

00为保留字节。返回信息中的第2、3、6、7个字符为0。

(DI)为两个字符的十六进制数，其对应的二进制数的各位分别指示IR-2110的8个数字输入通道的输入状态。由于IR-2110有8个输入通道(用8个二进制位表示)，因此(DI)的第1个字符(高4位)表示IN4～IN7的4个输入通道的状态；(DI)的第2个字符(低4位)表示IN0～IN3的4个输入通道的状态。(DI)的取值范围为00～FF。

（DI）各位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IR-2110输入通道	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
说明：每个通道的对应位为0表示输入通道的状态为OFF；为1表示输入通道状态为ON								

(Cr)为终止字符，即回车(0DH)。

例1(无检验和)：

命令：$006(Cr)

返回：!002A00(Cr)

该命令读指定地址为00H的IR-2110模块的I/O通道状态。

从模块返回的信息可知，(DI)=2AH，(对应8位二进制数00101010），可知输入通道IN0、IN2、IN4、IN6、IN7是OFF状态；输入通道IN1、IN3、IN5为ON状态。

例2(有检验和)：

命令：$006BA(Cr)

返回：!00000041(Cr)

该命令读指定地址为00H的IR-2110模块的输入通道状态。

从模块返回的信息可知，(DI)=00H，(对应二进制数00000000），指示所有8个输入通道均为OFF状态。

本命令检验和为BAH，返回数据检验和为41H。

* *

名称：同步采样命令

说明：命令所有I/O模块采样I/O通道的当前数值，并将该数值保存在模块内部的同步采样寄存器中。当模块收到该命令后将自动清除同步标志。

语法：#**

#为前导符。

**指示该命令为同步采样命令。

该命令不需要终止字符(Cr)，即回车字符(0Dh)。

(注意！实际应用中该命令后面无论有没有(Cr)字符都可以被模块成功接收)

返回：(无)

模块在接收到该同步采样命令后并不会返回任何信息。为了得到该同步采样数据，主机必须给模块发送一条读同步数据命令$AA4。

（关于同步采样的进一步信息请参见附录E）

$AA4

名称：读同步采样数据命令

说明：IR-2110模块收到该命令后会返回一个数据，该数据正好是之前该模块执行同步采样命令#**时已经保存于同步采样寄存器中的那个数值。

语法：$AA4(Cr)

$为前导符。

AA(00～FF)为模块地址(2个字符的十六进制数)。

4为读同步采样数据命令代码。

(Cr)为终止字符，即回车(0DH)。

返回：!(status)00(DI)00(Cr)

如果模块检测到了一个语法错误或通讯错误，或者命令所指定的地址不匹配，则不会返回任何信息。

!为界定字符，以指示一个有效的命令。

(status)为同步标志，指示返回的数据(data)是否为上一次同步采样命令#**被执行后第一次返回的数据。如果status=1，则说明该数据是第一次被读取的；如果status=0则说明该数据在上一次同步采样命令#**发出后已经被$AA4命令读走过至少一次(既:status=1表示读取的是最新数据，status=0表示读取的是旧数据)。模块上电复位时(status)缺省值为0。每当IR-2110收到读同步数据命令(#**)后，同步标志status便被清零。

(DI)为两个字符的十六进制数，其对应的二进制数的各位分别指示IR-2110的8个数字输入通道的输入状态。由于IR-2110有8个输入通道(用8个二进制位表示)，因此(DI)的第1个字符(高4位)表示IN4～IN7的4个输入通道的状态；(DI)的第2个字符(低4位)表示IN0～IN3的4个输入通道的状态。(DI)的取值范围为00～FF。

（DI）各位	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IR-2110输入通道	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
说明：每个通道的对应位为0表示输入通道的状态为OFF；为1表示输入通道状态为ON								

(Cr)为终止字符，即回车(0DH)。

说明：在IR-2110模块刚上电复位后，在没执行#**命令前执行$AA4命令，模块将返回!0000000(Cr)。

例1(无检验和)：

命令：$064(Cr)

返回：!1003C00(Cr)

该命令指定地址为06H的IR-2110模块返回它的同步采样数据，该数据为最后一次同步采样命令#**发出后保存于模块内部的同步采样寄存器中的数据。返回数据中状态(status)为1，表示该数据之前没有被#**命令读取过；(DI)=3CH，对应二进制数为00111100，可知输入通道2、3、4、5为ON状态；输入通道0、1、6、7为OFF状态。

例2(无检验和)：

命令：$004(Cr)

返回：!000FF00(Cr)

该命令指定地址为00H的IR-2110模块返回它的同步采样数据，该数据为最后一次同步采样命令#**发出后保存于模块内部的同步采样寄存器中的数据。返回数据中状态(status)为0，表示在上次发送同步采样#**命令后，同步采样数据已经被读取过至少1次。(DI)=FFH对应二进制数11111111，说明所有输入通道全都为ON状态。

例3(有检验和)：

命令：$004B8(Cr)

返回：!100120075(Cr)

该命令指定地址为00H的IR-2110模块回传它的同步采样数据，该数据为最后一次同步采样命令#**发出后载入到内部采样寄存器中的数据。模块返回信息中状态(status)为1，表示该数据是第一次被读取。(DI)=12H对应二进制数00010010，说明输入通道1、4为ON状态，0、2、3、5、6、7通道为OFF状态。

以上命令检验和为B8H，返回数据的检验和为75H。

$AA5

名称：读复位标志命令

说明：请求模块返回复位标志信息，该信息指示自上次读复位标志命令发出后该模块是否被复位过。

语法：$AA5(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

5为读复位标志命令代码。

(Cr)为终止字符，即回车(0DH)。

返回：!AAS(Cr) 当命令有效时

若模块检测到一个语法错误或通讯错误或命令指定的地址不匹配，将不会返回任何信息。

!为前导符，该字符指示该命令有效。

AA为返回信息的模块地址(2个字符的十六进制数)，取值范围为00～FF。

S为模块复位标志，S=0或1。如果S=1则表示该模块自上一次接收到读复位状态命令$AA5后，已经被复位了至少一次；如果S=0则表示模块自上一次收到读复位状态命令$AA5后没有被复位过。当模块上电复位或由硬件看门狗复位后，S被置1，每次执行完读复位标志命令$AA5后，S的值都被清0。

(Cr)为终止字符，即回车(0DH)。

（关于复位标志的进一步信息请参见附录C）

例1(无检验和)：

命令：$395(Cr)

返回：!390(Cr)

该命令指定地址为39H的模块返回它的复位状态信息。

从模块返回的信息可知复位状态S=0，说明该模块自上次复位状态命令($AA5)执行后没有被复位(人工复位或硬件看门狗复位)过。

例2(有检验和)：

命令：$005B9(Cr)

返回：!001B2(Cr)

该命令指定地址为00H的模块返回它的复位状态信息。

从模块返回信息可知复位状态S=1，说明该模块自上次复位状态命令($AA5)执行后被复位(人工复位或硬件看门狗复位)过至少1次。本命令检验和为B9，返回数据检验和为B2。

$AAL0

名称：读输入脉冲锁存命令

说明：本命令请求IR-2110模块返回输入脉冲锁存寄存器的内容。

语法：$AAL0(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

L0为读输入脉冲锁存命令。

(Cr)为终止字符，即回车(0DH)。

返回：!(Data)(Cr)当命令有效时

若模块检测到一个语法错误或通讯错误，或命令指定的地址不匹配，那么将不会返回任何信息。

!为前导符，指示该命令为有效。

(Data)为一个6位十六进制数，其中后2位始终为00，前4位十六进制数可最多表示16位二进制数，对应16个输入通道。但由于IR-2110只有8个输入通道，因此(Data)的前2个数位始终为0。既(Data)的实际取值为00XX00，其中XX的取值范围为00～FF，转换成8位二进制数后可表示8个输入通道的锁存状态。

XX的二进制位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
对应输入通道号	通道7	通道6	通道5	通道4	通道3	通道2	通道1	通道0
说明：每个通道的对应位为0表示该输入通道状态（电平）无变化；为1表示输入通道状态（电平）有变化								

(Cr)为终止字符，即回车(0DH)。

（说明：关于输入脉冲锁存的进一步信息请参见附录D）

例1(无检验和)：

命令：$12L0(Cr)

返回：!000100(Cr)

该命令指定地址为12H的IR-2110模块返回它的输入脉冲锁存寄存器的值。

从模块返回信息可知输入脉冲锁存寄存器的值为01（对应的二进制为00000001），因此可以判断输入通道0有脉冲发生，而输入通道1、2、3、4、5、6、7没有脉冲发生。

例2(带检验和)：

命令：$01L001(Cr)

返回：!00A30055(Cr)

该命令指定地址为01H的IR-2110模块返回它的输入脉冲锁存寄存器的值。

从模块返回信息可知输入脉冲锁存寄存器的值为A3（对应的二进制为10100011），因此可以判断输入通道0、1、5、7有脉冲发生，而输入通道2、3、4、6没有脉冲发生。

本命令检验和为01H，返回数据检验和为55H。

$AAC

名称：清除模块输入脉冲锁存寄存器命令

说明：本命令用于清除模块输入脉冲锁存寄存器的内容。清除后的脉冲锁存寄存器值为00。

语法：$AAC(Cr)

$为前导符。

AA为模块地址(2个字符的十六进制数)，取值范围为00～FF。

C为清除输入脉冲锁存寄存器命令。

(Cr)为终止字符，即回车(0DH)。

返回：!AA(Cr)当命令有效时

若模块检测到一个语法错误或通讯错误，或命令指定的地址不匹配，那么将不会返回任何信息。

!为前导符，指示该命令为有效。

AA（00～FF）为两字符十六进制数模块地址。

(Cr)为终止字符，即回车(0DH)。

例1(无检验和)：

命令1：$01L0(Cr)

返回1：!000F00(Cr)

命令2：$01C(Cr)

返回2：!01(Cr)

命令3：$01L0(Cr)

返回3：!000000(Cr)

命令1指定地址为01H的IR-2110模块返回它的输入脉冲锁存寄存器的值。

从模块返回信息1可知输入脉冲锁存寄存器的值为0F（对应的二进制为00001111），因此可知输入通道0、1、2、3有脉冲发生，而输入通道4、5、6、7没有脉冲发生。

命令2清除地址为01H的IR-2110模块的输入脉冲锁存寄存器。

从模块返回信息2可知清除操作成功。

命令3再次指定地址为01H的IR-2110模块返回它的输入脉冲锁存寄存器的值。

从模块返回信息3可知输入脉冲锁存寄存器的值为00（对应的二进制为00000000），因此可知自从上一个清除输入脉冲锁存命令执行以来，8个输入通道都没有脉冲发生。

例2(带检验和)：

命令：$01CC8(Cr)

返回：!0182(Cr)

该命令指定地址为01H的IR-2110模块清除它的输入脉冲锁存寄存器的值。

从模块返回信息可知清除操作成功。

本命令检验和为C8H，返回数据检验和为82H。

4. ModbusRTU协议指令系统

4.1 关于ModbusRTU协议

（注意！如果用户在使用IR-2110时采用的通信协议为IRASCII协议，则该部分可以跳过。）

IR-2000系列I/O模块的ModbusRTU协议设计完全符合标准的ModbusRTU协议国际标准规定的语法规则。关于ModbusRTU协议的详细内容请登陆异特路公司官网：www.itrob.com.cn 或 www.itrob.cn 获得。用户也可以登陆http://www.modbus.org获得关于Modbus协议的更详细的信息。

4.2 ModbusRTU协议命令列表

IR-2110的ModbusRTU协议指令系统主要由11条命令组成，如下表所示：

功能码（HEX）	子功能码（HEX）	命 令 描 述	页码
01	（无）	读（R）输入状态、输入脉冲锁存、输入同步采样	34
02	（无）	读（R）输入通道状态	37
46	00	读（R）模块名称/型号	39
	04	写（W）模块地址	41
	05	读（R）模块通信设置(波特率、协议模式)	43
	06	写（W）模块通信设置(波特率、协议模式)	45
	07	读（R）模块固件版本号	47
	08	读（R）模块复位状态	48
	17	清除输入脉冲锁存命令	49
	18	输入同步采样命令（广播命令）	50
	19	读（R）读同步数据标志	51

表4.2 ModbusRTU协议命令列表

说明：上表中功能码0x01、0x02为ModbusRTU协议标准规定的通用功能码，相关的命令数据格式请参考Modbus相关标准，而功能码0x46则是ModbusRTU协议标准规定的用户自定义功能码，其内部数据格式定义见对应命令。

4.3 ModbusRTU协议异常码列表

如果IR-2110模块无法正常执行命令，则可能返回的异常码信息如下表所示：

字节编号	名称	字节数	说 明
00	地址	1 Byte	从设备地址，范围（0x01～0xF7）
01	功能码	1 Byte	功能码的最高位MSB置1
02	异常码	1 Byte	参见Modbus协议标准定义 0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x02：非法数据地址 0x03：非法数据值 0x04：从设备错误
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

表4.3 ModbusRTU协议异常码列表

关于ModbusRTU协议CRC的计算方法请参见附录G。

4.4 ModbusRTU协议命令详解

功能码0x01(读模块通道状态)

该功能码用于读取输入通道的状态值、输入通道的脉冲锁存值以及输入通道的同步状态值。

当输入通道状态有变化时（比如由ON状态变成OFF状态，或反之）都会使对应输入通道脉冲锁存器置1并保持该1状态不变，直到发送清除锁存命令为止。

下面是功能码0x01所用的内部位寄存器的地址范围定义：

0x0020～0x0027：输入通道状态寄存器地址（分别对应输入通道0～7）

0x0040～0x0047：输入通道脉冲锁存寄存器地址（分别对应输入通道0～7）

0x0060～0x0067：输入通道同步状采样寄存器地址（分别对应输入通道0～7）

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x01
02	通道起始地址高8位	1	0x00
03	通道起始地址低8位	1	0x20～0x27：输入通道状态寄存器地址（输入通道0～7） 0x40～0x47：输入通道脉冲锁存寄存器地址（输入通道0～7） 0x60～0x67：输入通道同步状采样寄存器地址（输入通道0～7）
04	通道数高8位	1	0x00
05	通道数低8位	1	0x01～0x08
06	CRC-L	1	CRC低8位
07	CRC-H	1	CRC高8位

注1：区分该命令读取的是输入通道、输入脉冲锁存还是输入同步状态由命令中指定的通道地址决定。

注2：如果命令指定有效地址以外的通道地址，如0028，模块会返回异常码0x02（非法数据地址）。

注3：通道寄存器起始地址与通道数之和不能大于通道最大地址+1，否则模块将返回异常码0x03（非法数据值）。

注4：当命令指定读取的是输入通道同步采样寄存器时，执行该命令后模块会自动使同步采样标志清零。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x01
02	字节数	1	0x01
03	通道状态	1	当命令中的通道地址低8位为0x20～0x27时，本字节描述输入通道的状态；（0表示OFF；1表示ON） 当命令中的通道地址低8位为0x40～0x47时，本字节描述输入通道的脉冲锁存状态；（0表示状态无变化；1表示状态有变化） 当命令中的通道地址低8位为0x60～0x67时，本字节描述输入通道同步采样寄存器状态；（0表示OFF；1表示ON）
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	从设备地址，范围（0x01～0xF7）
01	功能码	1 Byte	功能码的最高位MSB置1
02	异常码	1 Byte	0x01：非法功能代码 0x02：非法数据地址 0x04：从设备错误
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

（说明：关于输入脉冲锁存的进一步信息请参见附录D）

例1：读输入通道状态值（以下命令及返回信息皆由十六进制数表示）

命令1：01 01 00 20 00 08 3C 06

返回1：01 01 01 8A D0 2F

命令2：01 01 00 24 00 04 7D C2

返回2：01 01 01 0A D1 8F

命令1中指定的模块地址为0x01，功能码为0x01，起始输入通道编号为0x0020（从输入通道0开始），要读取的输入通道的数量为0x0008（要读取8个输入通道的状态）。

既命令1指定地址为0x04的IR-2110模块返回输入通道0～7共8个输入通道的状态。

从模块返回信息1可知，数据字节的数值为0x8A，对应的二进制值为10001010。由于命令1指定读取从0开始的共8个输入通道的状态，因此根据Modbus RTU协议规定可知bit7～bit0分别表示输入通道7～0的状态。从返回数据可知输入通道1、3、7的状态分别为ON，而输入通道0、2、4、5、6的状态为OFF。

命令2中指定的模块地址为0x01，功能码为0x01，起始输入通道编号为0x0024（从输入通道4开始），要读取的输入通道的数量为0x0004（要读取4个输入通道的状态）。

既命令2指定地址为0x01的IR-2110模块返回输入通道4、5、6、7共4个输入通道的状态。

从模块返回信息2可知，数据字节的数值为0x0A，对应的二进制值为00001010。由于命令2指定读取从4开始的共4个输入通道的状态。因此根据Modbus RTU协议规定可知，数据字节的高4位为0（无意义），低4位从bit3～bit0分别表示IN7～IN4共4个输入通道的状态。从返回数据可知输入通道7、5的状态为ON，而输入通道6、4的状态为OFF。

命令1的CRC值为063C，返回信息1的CRC值为2FD0，命令2的CRC值为C27D，返回信息2的CRC值为8FD1。

（在实际应用中常采用一条命令一次读取所有的输入通道状态的方法，既命令1的方法）

例2：读输入通道脉冲锁存值（以下命令及返回信息皆由十六进制数表示）

命令1：07 01 00 40 00 08 3C 7E

返回1：07 01 01 18 51 0A

命令2：07 01 00 47 00 02 0D B8

返回2：07 81 03 E0 50

命令1中指定的模块地址为0x07，功能码为0x01，起始输入脉冲锁存通道编号为0x0040（从输入通道0开始），要读取的输入通道的数量为0x0008（要读取8个输入通道的脉冲锁存状态）。

既命令1指定地址为0x07的IR-2110模块返回输入通道0～7共8个输入通道的脉冲锁存状态。

从模块返回信息1可知，数据字节的数值为0x18，对应的二进制值为00011000。由于命令1指定读取从0开始的共8个输入通道的脉冲锁存状态，因此根据Modbus RTU协议规定可知，bit7～bit0分别表示输入通道7～0的脉冲锁存状态。从返回数据可知输入通道0、1、2、5、6、7的脉冲锁存状态为0（表示无电平变化），输入通道3、4的脉冲锁存状态为1（表示有电平变化）。

命令2中指定的模块地址为0x07，功能码为0x01，起始输入脉冲锁存通道编号为0x0047（从输入通道7开始），要读取的输入通道的数量为0x0002（要读取2个输入通道的脉冲锁存状态）。

既命令2指定地址为0x07的IR-2110模块返回输入通道7～8共2个输入通道的脉冲锁存状态。

从模块返回信息2可知，命令2有错误。由模块返回的错误代码0x03可知，命令2中有非法数据值。分析原因，由于IR-2110模块只有8个输入通道（编号从0～7），而命令2中指定通道的起始地址为0x0047，要读取的通道数量为0x0002（既2个通道），根据IR-2110模块的内部寄存器定义，0x0047对应输入通道7，已经是最后一个通道了，根据Modbus RTU协议可知，从这个位寄存器开始只能读取1个通道的数值。而命令中确指定读取2个通道，因此命令数据值非法。

命令1的CRC值为7E3C，返回信息1的CRC值为0A51，命令2的CRC值为B80D，返回信息2的CRC值为50E0。

（在实际应用中常采用一条命令一次读取所有的输入通道脉冲锁存状态的方法，既命令1的方法）

注意：一旦某个输入通道有电平变化并导致输入通道的脉冲锁存寄存器置1，这个锁存状态会一直保持下去，直到主机发送清除脉冲锁存命令（参见功能码0x46、子功能码0x17）为止，这个锁存状态才会被清零。

例3：读输入通道同步采样值（以下命令及返回信息皆由十六进制数表示）

命令1：**00 46 18 00 EB F1**

返回1：**（无）**

命令2：**03 01 00 60 00 08 3C 30** （注意！该命令将使同步采样标志清零）

返回2：**03 01 01 F0 50 74**

命令1中模块地址为0x00（为广播地址），功能码0x46，子功能码0x18，该命令为同步采样命令。既命令RS-485总线上所有的IO模块读取当前的输入通道状态并保存到同步采样寄存器中（参见功能码0x46之子功能码0x18）。

由于命令1为广播命令，因此模块不返回任何信息。

命令2中指定模块地址为0x03，功能码0x01，起始输入通道同步采样寄存器编号为0x0060（从输入通道0开始），要读取的同步输入通道的数量为0x0008（要读取8个输入通道的同步采样值）。

既命令2指定地址为0x03的IR-2110模块返回输入通道0～7共8个输入通道的同步采样值。

从模块返回信息2可知，数据字节的数值为0xF0，对应的二进制值为11110000。由于命令2指定读取从0开始的共8个输入通道的同步采样状态，因此根据Modbus RTU协议规定可知，bit7～bit0分别表示输入通道7～通道0的同步采样状态。从返回数据可知输入通道0、1、2、3的同步采样值为0（OFF），输入通道4、5、6、7的同步采样值为1（ON）。

命令1的CRC值为F1EB，命令2的CRC值为303C，返回信息2的CRC值为7450。

功能码0x02(读模块输入通道状态)

该功能码用于读取IR-2110模块的输入通道的状态值（ON/OFF）。

读取IR-2110模块输入通道状态也可以采用功能码0x01，但功能码0x01是多功能命令，而功能码0x02是专门用于读取IR-2110模块输入状态的命令。

对于IR-2110来说命令0x02使用的内部寄存器的地址排列如下：

0x0000～0x0007：输入通道状态寄存器地址（分别对应输入通道0～7）。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x02
02	通道起始地址高8位	1	0x00
03	通道起始地址低8位	1	0x00～0x07：输入通道状态寄存器地址（输入通道0～7）
04	通道数高8位	1	0x00
05	通道数低8位	1	0x01～0x08
06	CRC-L	1	CRC低8位
07	CRC-H	1	CRC高8位

注1：如果命令指定有效地址以外的通道地址，如0008，模块会返回异常码0x02（非法数据地址）。

注2：通道寄存器起始地址与通道数之和不能大于8，否则模块将返回异常码0x03（非法数据值）。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x02
02	字节数	1	0x01
03	通道状态	1	该字节的指定位分别对应指定输入通道的状态。 （1表示ON；0表示OFF）
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	从设备地址，范围（0x01～0xF7）
01	功能码	1 Byte	0x82 （功能码0x02的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码 0x02：非法数据地址 0x03：非法数据值
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：读输入通道状态值（以下命令及返回信息皆由十六进制数表示）

命令1：05 02 00 00 00 08 78 48

返回1：05 02 01 73 E1 5D

命令2：05 02 00 02 00 01 19 8E

返回2：05 02 01 00 A0 B8

命令1中指定的模块地址为0x05，功能码为0x02表示读取输入通道状态，起始输入通道编号为0x0000（从输入通道0开始），要读取的通道的数量为0x0008（要读取8个输入通道的状态）。

既命令1指定地址为0x05的IR-2110模块返回输入通道0～7共8个输入通道的输入状态。

从模块返回信息1可知，数据字节的数值为0x73，对应的二进制值为01110011。由于命令1指定读取从0开始的共8个输入通道的状态，因此根据Modbus RTU协议规定可知，数据字节的bit7～bit0分别表示输入通道7～0的状态。从返回数据可知输入通道2、3、7的状态为OFF，输入通道0、1、4、5、6的状态为ON。

命令2中指定的模块地址为0x05，功能码为0x02表示读取输入通道状态，起始输入通道编号为0x0002（从输入通道2开始），要读取的输入通道的数量为0x0001（要读取1个输入通道的状态）。

既命令2指定地址为0x05的IR-2110模块返回输入通道2的输入状态。

从模块返回信息2可知，数据字节的数值为0x00，对应的二进制值为00000000。由于命令2指定读取从IN2开始的共1个输入通道的输入状态，因此根据Modbus RTU协议规定可知，数据字节的bit7～bit1为0（无意义），bit0表示输入通道2的输入状态。从返回数据可知bit0=0，既输入通道2的状态为OFF。

命令1的CRC值为4878，返回信息1的CRC值为5DE1，命令2的CRC值为8E19，返回信息2的CRC值为B8A0。

（在实际应用中常采用一条命令一次读取所有的输入通道状态的方法，既命令1的方法）

功能码0x46 子功能码0x00(读模块名称)

该子功能码用于读取模块的名称/型号。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x00
03	CRC-L	1	CRC低8位
04	CRC-H	1	CRC高8位

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x00
03	(保留)	1	0x00
04	型号高2位	1	若模块为IR-2110，则该字节为0x21
05	型号底2位	1	若模块为IR-2110，则该字节为0x10
06	子型号	1	若模块为IR-2110， 则该字节为0x00
07	CRC-L	1	CRC低8位
08	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	从设备地址，范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：08 46 00 C2 62

返回：08 46 00 00 21 10 00 C1 AC

该命令指定地址为0x08的模块返回模块型号/名称。

从模块返回信息可知，模块的型号为2110。

以上内容中发送命令的CRC值为62C2，返回信息的CRC值为ACC1。

例2：（以下命令及返回信息皆由十六进制数表示）

命令：08 46 35 02 75

返回：08 C6 01 62 62

该命令指定地址为0x08的模块执行子功能码35的命令。

从模块返回信息可知命令无法正常执行（返回信息的功能码最高位为1），异常码为0x01，模块无法执行0x35子功能码，既0x35子功能码为非法子功能码。

以上内容中发送命令的CRC值为7502，返回信息的CRC值为6262。

例3：（以下命令及返回信息皆由十六进制数表示）

命令：01 48 00 16 00

返回：01 C8 01 B6 00

该命令试图指定地址为0x01的模块执行功能码0x48下的子功能码00的任务。

从模块返回信息可知命令无法正常执行（返回信息的功能码最高位为1），异常码为0x01，模块无法执行0x48功能码的00子功能，因为0x48功能码为非法功能码。

以上内容中发送命令的CRC值为0016，返回信息的CRC值为00B6。

功能码0x46 子功能码0x04(设置模块地址)

该子功能码用于设置(写)模块的地址。该命令成功执行后，新地址将立刻生效。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x04
03	新地址	1	模块新地址（有效范围0x01～0xF7）注意！不能设置超过范围以外的地址，否则设置失败
04	(保留)	1	0x00
05	(保留)	1	0x00
06	(保留)	1	0x00
07	CRC-L	1	CRC低8位
08	CRC-H	1	CRC高8位

注1：新地址不能超出ModbusRTU协议规定的合法范围（0x01～0xF7），否则模块将返回异常码0x03。

注2：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块新地址（有效范围0x01～0xF7）如果设置成功那么该字节为新设置的地址，既新地址立即生效！
01	功能码	1	0x46
02	子功能码	1	0x04
03	(保留)	1	0x00
04	(保留)	1	0x00
05	(保留)	1	0x00
06	(保留)	1	0x00
07	CRC-L	1	CRC低8位
08	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：非法数据值。当新设定的模块地址为非法或保留字节不为0x00。
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：A1 46 04 05 00 00 00 54 60

返回：05 46 04 00 00 00 00 B1 66

该命令设定地址为0xA1的模块的新地址为0x05。

从模块返回信息可知，新地址设置成功。

以上内容中发送命令的CRC值为6054，返回信息的CRC值为66B1。

例2：（以下命令及返回信息皆由十六进制数表示）

命令：3C 46 04 00 00 00 00 18 65

返回：3C C6 03 A2 6D

该命令试图将地址为0x3C的模块的新地址设定为0x00。

从模块返回信息可知，新地址0x00设置失败。模块返回0x03异常码，表示数据有错误或命令无法执行。原因是新地址0x00是广播地址，模块不能设置该地址，因此设置失败。

以上内容中发送命令的CRC值为6518，返回信息的CRC值为6DA2。

例3：（以下命令及返回信息皆由十六进制数表示）

命令：2A 46 04 02 0A 00 00 4E 1E

返回：2A C6 03 43 A9

该命令试图设定地址为0x2A的模块的新地址为0x02。（注意！在该命令中的第一个保留字节不是0x00，而是0x0A。）

从模块返回信息可知，命令无法成功执行，模块返回异常码为0x03，说明有非法数据值。

原因是命令中第5个字节为保留字节，该字节必须为0x00，而实际值为0x0A。此种情况模块会将其视为非法数据值，因此返回异常码0x03。

以上内容中发送命令的CRC值为1E4E，返回信息的CRC值为A943。

例4：（以下命令及返回信息皆由十六进制数表示）

命令1：02 46 04 03 00 00 00 C7 E2

返回1：03 46 04 00 00 00 00 D7 66

命令2：02 46 04 04 00 00 00 C6 96

返回3：（无返回信息）

命令1设定地址为0x02的模块的新地址为0x03。

从模块返回信息1可知，新地址0x03设置成功。

命令2试图再次指定地址为0x02的模块的新地址为0x04。

此时模块无任何响应。原因是命令1成功执行后模块的地址已经变成0x03了，而命令2中指定的0x02的地址不存在，因此模块不做任何响应。

以上内容中命令1的CRC值为E2C7，返回信息1的CRC值为66D7，命令2的CRC值为96C6。

功能码0x46 子功能码0x05(读模块通信参数)

该子功能码用于读取保存于模块内部的EEPROM中的通信设置参数，这些通信参数包括波特率和通信协议。

注意！该命令读取的是模块内部EEPROM中的通信参数，而不是模块当前正在使用的通信参数，这两种参数可能是不同的。以上情况的一个例子就是当用写通信参数命令（参见功能码0x46子功能码0x06）设置了新的参数到模块后，在模块没有复位前，新参数是无法生效的，既EEPROM中的新参数与模块当前的参数是不同的。此时可以采用本命令读取EEPROM中的通信参数，以确定新参数是否确实写入到EEPROM中了。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x05
03	(保留)	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x05
03	(保留)	1	0x00
04	波特率	1	0x03（1200）, 0x04（2400）, 0x05（4800）, 0x06（9600）, 0x07（19200）, 0x08（38400）, 0x09（57600）, 0x0A（115200）
05	(保留)	1	0x00
06	(保留)	1	0x00
07	(保留)	1	0x00
08	协议模式1	1	0x00:IRASCII协议 0x01:Modbus RTU协议
09	协议模式2	1	当协议模式1=0x00时才有效 0x00: IRASCII协议（无检验和） 0x01: IRASCII协议(有检验和)
10	(保留)	1	0x00
11	CRC-L	1	CRC低8位
12	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：非法数据值（当保留字节不为0x00时）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：**23 46 05 00 E9 25**

返回：**23 46 05 00 06 00 00 00 01 00 00 48 3B**

该命令指定地址为0x23的模块返回EEPROM中的通信参数设置信息。

从模块返回信息可知，模块的通信波特率为9600bps，通信协议为Modbus RTU协议。

以上内容中发送命令的CRC值为25E9，返回信息的CRC值为3B48。

例2：（以下命令及返回信息皆由十六进制数表示）

命令：**23 46 05 00 E9 25**

返回：**23 46 05 00 08 00 00 00 00 00 00 F6 3B**

该命令指定地址为0x23的模块返回EEPROM中的通信参数设置信息。

从模块返回信息可知，模块的通信波特率为38400bps，通信协议为IRASCII协议，且不带检验和。

注意！虽然模块当前工作于Modbus RTU协议下，但返回的信息显示模块的通信协议为IRASCII协议。可能的原因参见上文。

以上内容中发送命令的CRC值为25E9，返回信息的CRC值为3BF6。

例3：（以下命令及返回信息皆由十六进制数表示）

命令：**23 46 05 AA 69 5A**

返回：**23 C6 03 93 AB**

该命令指定地址为0x23的模块返回EEPROM中的通信参数设置信息。

从模块返回信息可知，该命令出现异常，无法正确执行。由返回异常码0x03可知，异常原因是命令中的保留字节非0x00导致的。

以上内容中发送命令的CRC值为5A69，返回信息的CRC值为AB93。

功能码0x46 子功能码0x06(设置模块通信参数)

该子功能码用于设置(写)模块的通信设置参数，这些通信参数包括波特率和采用的通信协议。

该命令成功执行的前提是必须使模块的INIT*与GND短接。当命令成功执行后，这些参数将保存于模块内部的EEPROM中，但不会立即生效。如果想使新设置的参数生效，必须将INIT*与GND断开，然后重新启动模块。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x06
03	(保留)	1	0x00
04	波特率	1	0x03（1200）, 0x04（2400）, 0x05（4800）, 0x06（9600）, 0x07（19200）, 0x08（38400）, 0x09（57600）, 0x0A（115200）
05	(保留)	1	0x00
06	(保留)	1	0x00
07	(保留)	1	0x00
08	协议模式1	1	0x00:IRASCII协议 0x01:Modbus RTU协议
09	协议模式2	1	当协议模式1=0x00时才有效 0x00: IRASCII协议（无检验和） 0x01: IRASCII协议(有检验和)
10	(保留)	1	0x00
11	CRC-L	1	CRC低8位
12	CRC-H	1	CRC高8位

注1：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

注2：当协议模式1=0x01时（表示Modbus RTU协议），协议模式2将被忽略，但必须为0x00或0x01，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x06
03	(保留)	1	0x00
04	(保留)	1	0x00
05	(保留)	1	0x00
06	(保留)	1	0x00
07	(保留)	1	0x00
08	(保留)	1	0x00
09	(保留)	1	0x00
10	(保留)	1	0x00
11	CRC-L	1	CRC低8位
12	CRC-H	1	CRC高8位

注意！新设置的波特率和协议要生效必须先将INIT*与GND断开，然后重新启动模块。

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：波特率、协议设置数据内容非法时以及保留字节不为0x00 0x04：参数正确但无法成功设置（比如INIT*引脚没有与GND短接）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：01 46 06 00 0A 00 00 00 01 00 00 30 B3

返回：01 46 06 00 00 00 00 00 00 00 00 CB 73

该命令设置地址为0x01的模块的波特率为115200bps，通信协议采用Modbus RTU协议。

从模块返回信息可知，通信参数设置成功。当重新启动模块后新的设置才会生效。

以上内容中发送命令的CRC值为B330，返回信息的CRC值为73CB。

例2：（以下命令及返回信息皆由十六进制数表示）

命令：01 46 06 00 06 00 00 00 02 00 00 0C B3

返回：01 C6 03 33 A1

该命令设置地址为0x01的模块的波特率为9600bps，而通信协议设置字节为0x02，为非法数据。

从模块返回信息可知，通信参数设置失败。原因是通信协议设置字节为0x02，为非法内容。此时模块会将其视做非法数据格式，因此返回异常码0x03。

以上内容中发送命令的CRC值为B30C，返回信息的CRC值为A133。

例3：（以下命令及返回信息皆由十六进制数表示）

命令：02 46 06 00 04 00 00 00 01 00 00 D0 37

返回：02 C6 04 82 63

该命令设置地址为0x02的模块的波特率为2400bps，通信协议设置为Modbus RTU协议。

从模块返回信息可知，通信参数设置失败。模块返回异常代码0x04，表示从设备错。

由于命令的格式及内容都是正确的，因此原因有可能是IR-2110模块的INIT*引脚与GND之间没有短接。由于要成功设置模块的波特率和通信协议的前提是必须先使模块的INIT*与GND短路，否则模块会返回设置失败的信息。

以上内容中发送命令的CRC值为37D0，返回信息的CRC值为6382。

功能码0x46 子功能码0x07(读固件版本号)

该子功能码用于读模块的固件版本号。

固件版本号由6位阿拉伯数字组成。前4位数字描述了固件版本号的年份信息，后2位描述了版本号信息。如201201表示版本号为2012年的第01版。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x07
03	CRC-L	1	CRC低8位
04	CRC-H	1	CRC高8位

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x07
03	版本1,2位	1	版本年份的前2位
04	版本3,4位	1	版本年份的后2位
05	版本5,6位	1	版本号
06	CRC-L	1	CRC低8位
07	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：03 46 07 F2 62

返回：03 46 07 20 12 01 44 39

该命令读取地址为0x03的模块固件版本号信息。

从模块返回信息可知，该模块的固件版本号为201201。

以上内容中发送命令的CRC值为62F2，返回信息的CRC值为3944。

功能码0x46 子功能码0x08(读复位标志)

该子功能码用于读模块的复位标志。

（关于复位标志的进一步信息请参见附录C）

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x08
03	（保留）	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x08
03	复位标志	1	0x00-表示没有复位；0x01-表示有复位
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

每当模块接收到读复位标志命令并正确返回标志值后，该复位标志便自动清零（复位标志=0x00）。

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：数据错（当保留字节不为0x00时）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令：08 46 08 00 E4 51

返回：08 46 08 01 25 91

该命令读取地址为0x08的模块的复位标志。

从模块返回信息可知，该模块的复位标志值为0x01，说明模块在该命令执行前被复位过。

以上内容中发送命令的CRC值为51E4，返回信息的CRC值为9125。

功能码0x46 子功能码0x17(清除输入脉冲锁存)

该子功能码用于清除IR-2110模块输入通道的脉冲锁存值。

每当IR-2110输入通道的电平发生变化（我们称此变化为脉冲），这个变化都会导致对应输入通道脉冲锁存寄存器置1，并且这个1状态会一直保留，直到模块接收到清除输入通道脉冲锁存命令为止。一旦该命令成功执行，IR-2110的所有输入通道的脉冲锁存状态全部清零，既8个输入脉冲锁存寄存器全部为0。

关于输入脉冲锁存的详细内容参见附录D。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x17
03	（保留）	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x17
03	（保留）	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注：一旦模块接收到清除输入通道锁存命令并成功执行后，所有的输入脉冲锁存寄存器全部清零。

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：数据错（当保留字节不为0x00时）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令1：08 46 17 00 EC 61

返回1：08 46 17 00 EC 61

命令1指定地址为0x08的模块清除所有的输入通道脉冲锁存寄存器。从模块返回信息1可知，该命令已成功执行。

以上内容中命令1的CRC值为61EC，返回信息1的CRC值为61EC。

功能码0x46 子功能码0x18(同步输入采样命令)

该子功能码用于命令RS-485总线上的所有的IR-2000模块读取输入通道的当前输入状态，并保存于同步采样寄存器中。

由于该命令为广播命令，因此该命令指定的地址必须是0x00，且模块接收并执行该命令后不返回任何信息。

该命令发送后所有模块都将收到该命令并产生相应的动作， 既读取模块将当前的输入通道状态，并保存于内部的同步采样寄存器中，以用于之后的读模块同步输入状态命令(功能码0x01)读取，同时还会将内部的同步采样标志置成0x01。模块执行该命令后不返回任何信息。

关于同步采样的进一步信息请参考附录E。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	必须为0x00
01	功能码	1	0x46
02	子功能码	1	0x18
03	（保留）	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注1：命令中的保留字节必须为0x00，否则模块将不会执行该命令。

注2：如果命令中地址字节不为0x00，而是某个模块的实际地址，那么该模块会将该命令视为非法命令，并返回异常码0x01。也就是说只有当指定地址为广播地址0x00时，该命令才会被模块正确接收。

返回信息格式：（无）模块收到该命令后不会返回任何信息。

错误响应：（无）

例1：（以下命令皆由十六进制数表示）

命令1：00 46 18 00 EB F1

返回1：（无）

命令1的CRC值为F1EB。

功能码0x46 子功能码0x19(读同步采样标志)

该子功能码用于读取IR-2110模块的同步采样标志。

同步采标志的主要作用是用于主机查询模块当前的同步采样数据是否为上次同步采样命令(参见功能码0x46，子功能码0x18)发送以来第一次被读取的，也就是该标志反映了模块的同步采样数据是否曾经被读取过。

模块返回的同步采样标志只有2种可能的数值0x00和0x01，含义如下：

0x00：表示同步采样数据已经被读取过了；0x01：表示同步采样数据还没有被读取过；

每当模块收到同步采样命令(参见功能码0x46，子功能码0x18)后，该标志便自动置成0x01。

每当模块收到读同步输入采样命令(参见功能码0x01，通道号010060～0x0067)并正常返回信息后，同步采样数据为新标志便自动清零(0x00)。 关于同步采样标志的详细信息参见附录F。

命令格式：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x19
03	（保留）	1	0x00
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

注：命令中的保留字节必须为0x00，否则模块将返回异常码0x03。

若命令成功执行，则返回信息格式如下：

字节编号	名 称	字节数	说 明
00	地址	1	模块地址（有效范围0x01～0xF7）
01	功能码	1	0x46
02	子功能码	1	0x19
03	同步采样标志	1	0x00-同步输入采样数据为旧数据，该数据已经被读取过； 0x01-同步输入采样数据为新数据，该数据还没被读取过；
04	CRC-L	1	CRC低8位
05	CRC-H	1	CRC高8位

错误响应：

字节编号	名 称	字节数	说 明
00	地址	1 Byte	范围（0x01～0xF7）
01	功能码	1 Byte	0xC6 （功能码0x46的最高位MSB置1） 当命令功能码非法时，该字节为命令功能码最高位置1。
02	异常码	1 Byte	0x01：非法功能代码（这里的功能代码既包括功能码也指子功能码） 0x03：数据错（当保留字节不为0x00时）
03	CRC-L	1 Byte	CRC低8位
04	CRC-H	1 Byte	CRC高8位

当发送命令出现CRC错误时，模块将不返回任何信息。

例1：（以下命令及返回信息皆由十六进制数表示）

命令1：1A 46 19 00 ED 79

返回1：1A 46 19 01 2C B9

命令1指定地址为0x1A的模块返回同步采样标志。从模块返回信息可知同步采样标志为0x01，既模块当前的同步采样数据为新数据，还没有被读取过。

以上内容中命令1的CRC值为79ED，返回信息1的CRC值为B92C。

5. 外型及尺寸

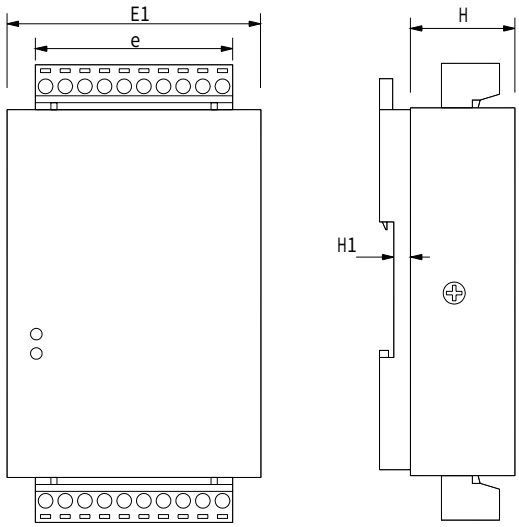


图5.a 搭配DIN导轨配件尺寸

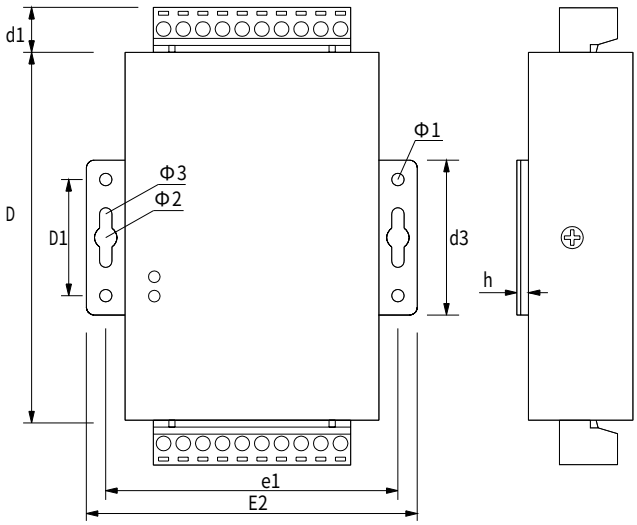


图5.b 搭配壁挂配件尺寸

表5 尺寸对照表(单位:mm)

E1	65.8	**d3**	40.0
E2	85.0	**H**	26.8
e	50.8	**H1**	4.8
e1	74.6	**h**	3.0
D	96.0	**Φ1**	3.3
D1	30.0	**Φ2**	5.5
d1	11.6	**Φ3**	3.8

附录A IRASCII协议语法介绍

IRASCII通信协议规定了主机向模块发送命令的指令语法格式以及模块返回数据给主机的语法格式。

命令格式：(Leading)(Address)(Command)(Data)[CHK](Cr)

响应格式：(Leading)(Address)(Data)[CHK](Cr)

语法格式中涉及的各字段含义。

(Leading)：为前导符。主机命令常用的前导符有'$'、'%'、'#'；从机响应常使用的前导符有'!'、'?'、'>'。

(Address)：为目标模块地址。由2个十六进制字符组成。如：00、A1、5B、FF等，其中字母必须为大写。有些命令没有地址字段，如同步采样命令#**。

(Command)：为命令代码。不同的主机命令使用不同的命令代码，如代码'2'表示读配置、代码6表示读I/O通道、代码X0表示写安全值等。不同命令的代码不同，代码的字符数可能也不同。另外，有些命令要求主机发送数据给从机的还需要在命令代码后面添加数据信息，如单通道输出命令#AA00(data)(Cr)；还有些命令没有命令代码，其代码已经隐含于前导符中了，如模块配置命令%AANNTTCCFF(Cr)。

(Data)：为数据。可以是1位、2位、3位或4位或更多位十六进制字符组成。如安全值设置命令$AAX0TTTTDDDD(Cr)中的(Data)字段由TTTTDDDD组成，其中TTTT由4个十六进制字符组成，DDDD也由4个十六进制字符组成，因此该命令的(Data)字段总共由8个十六进制字符组成。

[CHK]：为检验和(可选)。该字段由2个十六进制字符组成。如果用户选择了通信带检验和，那么要求主机发送的命令以及模块返回的信息中都必须带有检验和字段；如果用户选择了通信不带检验和，那么要求主机命令和模块返回信息都不带检验和。(关于检验和的计算方法见下节)

(Cr)：为命令结束符。主机命令和模块响应信息的最后一个字节都必须为(Cr)字符，既0DH。所有命令中只有同步采样命令#**后不需要(Cr)字符。

下面以读配置信息命令为例说明ASCII协议的命令格式。

假设I/O模块的地址为00H，波特率为9600bps，模块代码为40H，模块的协议字为00H(采用ASCII协议、不使用检验和)，若主机要读该模块的配置信息则需发送如下命令$002(Cr),如果正常模块应返回!00400600(Cr)。以上命令和响应信息中每个ASCII字符对应的十六进制值如下表所示：

主机发送的ASCII码命令	'$'	'0'	'0'	'2'	(Cr)
每个字符对应十六进制值	24H	30H	30H	32H	0DH

表3.1.a

模块返回的ASCII码信息	'!'	'0'	'0'	'4'	'0'	'0'	'6'	'0'	'0'	(Cr)
每个字符对应十六进制值	21H	30H	30H	34H	30H	30H	36H	30H	30H	0DH

表3.1.b

(注意！IRASCII协议中所有字母必须为大写)

附录B IRASCII协议检验和的计算

每条主机命令或模块返回信息的检验和的计算方法为将该条命令或返回信息的数据包中除了最后的(Cr)字符以外的所有ASCII字符对应的十六进制值相加，所得到的和(十六进制数)取最低2位既是检验和。

举例：比如读模块配置信息命令$002(Cr)的带检验和的形式为$002B6(Cr) ，其检验和的计算方法为将$002的每个字符对应的ASCII码值相加，既24H+30H+30H+32H=B6H，将该检验和以十六进制字符形式写入命令的[CHK]字段既为$002B6(Cr)。注意!最后必须有(Cr)字符。

如果模块不带检验和时返回信息为!00400600(Cr)，那么带检验和时应该返回!00400600AB(Cr)。计算方法同上，既：21H+30H+30H+34H+30H+30H+36H+30H+30H=1ABH，取最后2位既AB就是要求的检验和。

(注意！如果主机通讯带检验和，那么模块返回信息必须也带检验和。)

附录C 复位标志

当模块上电复位或被看门狗复位以后，复位标志被置位，并且该复位标志被读复位状态命令清零。如果复位标志被置为1，则说明模块已经复位了至少1次，并且输出有可能已经改变为上电初始值了。同时复位标志为1还说明在此期间模块被人为手动复位或由于模块程序故障导致模块被硬件看门狗复位的情况发生了至少1次。该项功能对于用户的意义在于，用户可以以此来检测模块的工作状态是否稳定。

一般情况下，用户的数据采集系统在刚启动时，控制主机要先向I/O模块发送一条读复位标志命令，这样做的主要目的是清除复位标志。在之后的程序中可以定期向模块发送读复位标志命令，以读取复位标志的值。如果系统正常，模块工作稳定，那么每次读取的复位标志都应该为0。如果在系统运行期间I/O模块工作不稳定，导致模块被内部的硬件看门狗自动复位（这种情况的发生是随机的不可预测的），那么当主机再次向模块发送读复位标志命令时，模块就会返回复位标志=1的信息。这时，主机就知道该模块曾经出现过至少1次故障。

附录D 输入脉冲锁存

对于具备开关量输入通道的IR-2000系列IO模块来说，输入脉冲锁存被定义为当输入通道电平状态发生变化时，模块内部寄存器对这一状态变化的保存。这里所说的输入通道电平状态发生变化包扩两种情况：一种是输入电平由高到低的变化，既下降沿；一种是输入电平由低到高的变化，既上升沿。就是说只要输入通道电平发生变化，模块就会把这种变化看作一个脉冲信号，并使与该通道对应的内部的输入脉冲锁存寄存器置1，之后无论该输入通道如何变化，该寄存器都会一直保存这个1状态不变。而要想清除这个1状态必须通过清除输入脉冲锁存命令来完成。

输入脉冲锁存命令的主要作用是使模块可以捕捉到输入信号在瞬间（短时间）发生的变化。而如果使用读输入通道状态命令，由于每次只能读取模块输入通道的当前状态（高/低电平），要想捕捉瞬间的脉冲信号则理论上必须至少2次读取通道状态（采样2次），既至少发送2条读输入通道状态命令。这一过程意味着主机要通过485总线发送2条命令，每条命令之间还要等待模块回传数据。由于485通讯速率以及主从应答通讯方式本身的时间消耗，这个过程很可能导致丢失这个脉冲信号。如图d.1所示，由于低电平脉冲时间可能很短，2个采样时间点可能正好处于A和B两点，采样的结果是两次采样都是高电平，得到的结论是没有发现低电平脉冲。由此可见，要想通过普通的输入通道采样命令采样到输入脉冲信号是很不可靠的。

输入脉冲锁存最常见的应用就是报警器报警信号的采集。在安防领域里常见的红外报警器的报警信号输出都采用脉冲输出。比如，正常情况下报警器会恒定输出一个高电平信号，如果有物体移动而触发报警，则报警器会输出一个低电平脉冲（宽度大约为若干毫秒）。如图d.1所示：

图d.1 脉冲波形

如果使用开关量输入模块采集这个报警信号，则可以使用输入脉冲锁存命令来实现。

报警监控计算机先发送一条清除输入脉冲锁存命令给模块以清除报警锁存，然后定时发送输入脉冲锁存命令。如果没有报警信号则每次模块返回的对应的输入脉冲锁存寄存器的值应该为0；而如果有报警发生，则对应位为1，这时系统便知道有报警发生了。

附录E 同步采样

IR-2000系列I/O模块的通信接口都采用RS-485接口。由于基于RS-485通信的分布式数据采集与控制系统无论其采用何种通信协议，都有一个共同的特点就是都采用轮询的方式访问下面的设备(I/O模块)。这种方式带来的一个主要问题就是当下面挂接的设备数量比较多的时候速率会比较慢。设想一下，当总共有200个模块挂接到485总线的时候，就算每访问一个模块只需要0.2秒的时间，将所有200个模块全部访问一便也需要至少40秒(02*200)的时间。而有些应用恰好就偏偏要求采集同一时刻的所有模块的输入值(注意！这里说的是同一时刻的所有模块的输入数值)。显然轮询的方式是不可能做到这点的。原因很容易理解，由于每访问一个模块都需要时间，轮询完所有模块需要比较长的时间，而在此时间段内，每个模块的输入都可能会发生变化。因此，最后采样得到的结果并不是同一个时间点的数值。

解决上述问题的方法就是采用同步采样。

同步采样的过程主要分2步：

第一步，主机发一个同步采样命令到RS-485总线上。由于同步采样命令是广播命令，所有挂接的模块都会接收该命令并产生动作，但并不返回任何数据。产生什么动作呢，就是对输入通道进行采样，并将采样的结果保存到模块内部的同步采样寄存器中，以等待主机读取。注意！在模块接收到下一条同步采样命令之前，同步采样寄存器中的内容始终不变，而不管在此期间输入通道的状态如何变化。

第二步，主机开始以轮询的方式逐一访问每个模块。但访问的命令并不是常规的读输入通道命令，而是读同步采样寄存器的命令。

上述方法很有效的解决了“同一时刻”的问题。但由于第二步的轮询方式当模块数量比较多时仍然需要较长的时间，因此在实际应用中，每个同步采样时间点之间的时间间隔可能仍然比较长。

附录F 同步采样标志

当模块收到同步采样命令后，除了对当前的输入通道进行采样并将结果保存到内部的同步采样寄存器以外，还要对模块内部的同步采样标志置位。同步采样标志的作用是指示模块的同步采样寄存器中的内容是否被读取过，也可理解为是否为新数据。

每当模块收到主机发来的读同步采样寄存器命令并返回信息后，该标志即被清零，以指示同步采样值已经被读取过了。

附录G ModbusRTU协议CRC的计算

关于ModbusRTU协议CRC值的计算方法及原理在很多技术文章中都有较详细的论述。因此，在这里只介绍ModbusRTU协议CRC值的计算方法。

计算CRC码的步骤为：

1、预置16位寄存器的初始值为十六进制FFFF（即全为1）。称此寄存器为CRC寄存器；

2、把第一个8位数据与16位CRC寄存器的低8位相异或，把结果放于CRC寄存器中；

3、把CRC寄存器的内容右移一位(朝低位)，用0填补最高位，检查刚移出的最低位；

4、如果最低位为0：重复第3步(再次移位)；

如果最低位为1：CRC寄存器与多项式A001(二进制表示为1010 0000 0000 0001)进行异或；

5、重复步骤3和4，直到右移8次，这样整个8位数据全部进行了处理；

6、重复步骤2到步骤5，进行下一个8位数据的处理,直到最后一个字节；

7、最后得到的CRC寄存器即为CRC码。

下面举例说明ModbusRTU协议CRC的计算过程：

假设需要计算CRC值的字节串有2个字节12 AB，下面的图表演示了CRC的计算过程。

计算第1个字节的CRC值。步骤如下：

	CRC高8位								CRC低8位								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	移出位
CRC初始值为FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
第1个字节为0x12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	
与第1个字节异或	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	
计算结果右移1位，高位补0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1（第1次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	
计算结果右移1位，高位补0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1（第2次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0	
计算结果右移1位，高位补0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0（第3次右移）
继续右移1位，高位补0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1（第4次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	
计算结果右移1位，高位补0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1（第5次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	0	
计算结果右移1位，高位补0	0	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	0（第6次右移）
继续右移1位，高位补0	0	0	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1（第7次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	0	0	1	1	0	1	0	0	1	1	1	1	1	1	0	
计算结果右移1位，高位补0	0	1	0	0	1	1	0	1	0	0	1	1	1	1	1	1	0（第8次右移）

已经累计右移8次，此时的CRC为第1个字节的CRC计算结果，此结果为：0x4D3F

如上表所示，第1个字节的CRC计算结果为0x4D3F，该值在计算第2个字节的CRC值时将作为CRC的初始值使用。接下来根据第1个字节的CRC计算结果(0x4D3F)来计算第2个字节的CRC值，见下页。

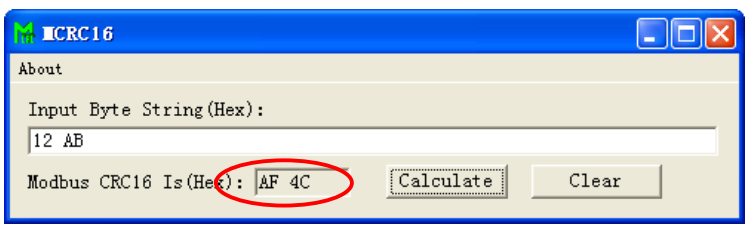
计算第2个字节0xAB的CRC值。步骤如下：

	CRC高8位								CRC低8位								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	移出位
CRC的值为4D3F	0	1	0	0	1	1	0	1	0	0	1	1	1	1	1	1	
第2个字节为0xAB	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	
与第2个字节异或	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	0	
计算结果右移1位，高位补0	0	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	0（第1次右移）
继续右移1位，高位补0	0	0	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0（第2次右移）
继续右移1位，高位补0	0	0	0	0	1	0	0	1	1	0	1	1	0	0	1	0	1（第3次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	0	1	0	1	0	0	1	1	0	1	1	0	0	1	1	
计算结果右移1位，高位补0	0	1	0	1	0	1	0	0	1	1	0	1	1	0	0	1	1（第4次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	
计算结果右移1位，高位补0	0	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0（第5次右移）
计算结果右移1位，高位补0	0	0	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0（第6次右移）
计算结果右移1位，高位补0	0	0	0	1	1	1	1	0	1	0	0	1	1	0	1	1	0（第7次右移）
计算结果右移1位，高位补0	0	0	0	0	1	1	1	1	0	1	0	0	1	1	0	1	1（第8次右移）
与A001异或	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
异或结果	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0	0	

已经累计右移8次，此时的CRC为第2个字节的CRC计算结果，此结果为：0xAF4C

如上表所示，第2个字节的CRC值计算结果为0xAF4C，这个值就是最终的计算结果，既字节串12 AB的CRC值为0xAF4C。

以上结果还可以通过异特路公司的专门用于计算ModbusRTU协议CRC值的软件MCRC16来验证：如下图所示：



图F.1 MCRC16的计算结果

上图中的计算软件为MCRC16，是异特路的一款专门计算ModbusRTU协议的CRC计算软件。该软件也是一款绿色的免费软件，可以通过异特路公司的官方网站免费下载。另外，IR2000Utility软件也可以计算CRC，但与MCRC16软件不同的是IR2000Utility必须在成功打开串口的前提下才可以计算CRC，而MCRC16不用。

上面的例子还有一点需要注意的是字节串12 AB的CRC计算结果为AF 4C，那么将该CRC计算结果添加到字节串12 AB的末尾以形成最终要发送的ModbusRTU协议包的字节顺序为：12 AB 4C AF。既CRC计算结果的低8位在前，高8位在后，这点用户一定要注意！

下面的C++函数是专门计算ModbusRTU协议CRC值的函数，MCRC16程序正是使用该函数来计算CRC的。

该函数有4个入口参数，介绍如下：

pchByteBuf ---指向字节串缓冲区的第一个字节的指针

dwByteNum ----缓冲区中的字节数

pchCrcH -----指向计算的CRC值的高8位(该字节要放入ModbusRTU协议数据包的倒数第1个字节)

pchCrcL -----指向计算的CRC值的低8位(该字节要放入ModbusRTU协议数据包的倒数第2个字节)

```
void CalModbusRTUCrc(char *pchByteBuf, DWORD dwByteNum, char *pchCrcH, char *pchCrcL)
{
    BYTE       chCrcH,chCrcL;              //CRC寄存器高8位和低8位
    BYTE       chChackH,chChackL;          //检测位(用于检测右移1位后的移出位)
    DWORD      i;
    DWORD      j;
    chCrcH=0xFF;
    chCrcL=0xFF;
    for( i=0; i<dwByteNum; i++ )
    {
     chCrcL = chCrcL ^ pchByteBuf[i];
     for( j=0; j<8; j++ )
     {
      chChackL = chCrcL & 0x01;
      chChackH = chCrcH & 0x01;
      chCrcL = chCrcL>>1;
      if( chChackH == 0x01 )
      {
       chCrcL = chCrcL | 0x80;
      }
      else
      {
       chCrcL = chCrcL & 0x7F;
      }
      chCrcH = chCrcH>>1;
      chCrcH = chCrcH & 0x7F;
      if( chChackL == 0x01 )
      {
       chCrcH = chCrcH ^ 0xA0;
       chCrcL = chCrcL ^ 0x01;
      }
    }
  }
    pchCrcH[ 0 ] = chCrcH;
    pchCrcL[ 0 ] = chCrcL;
}
```

上述函数CalModbusRTUCrc用户可以直接拿来使用。

下面的C++代码说明该函数的用法。

```
char achBuffer[1024];//定义一个缓冲区，大小为1024字节
char chCRCH,chCRCL;//定义2个字符变量，分别用于存放计算的CRC值的高8位和低8位。
/* 假设上述缓冲区中已经存放了10个字节的数据包(从0单元开始存放)，下面开始计算该数据包的CRC值 */
CalModbusRTUCrc( achBuffer, 10, &chCRCH, &chCRCL);
```

上述函数执行后字符变量chCRCH和chCRCL中将分别存放计算好的CRC值的高8位和低8位。

关于ModbusRTU协议下的CRC计算还有一种经常用的方法就是查表法，关于查表法的详细信息请用户参考《Modbus协议标准(应用)》，该文档可以从异特路公司的官网免费下载。

ITROB TECHNOLOGY DEPARTMENT DT：BZAB-ZG-ZF DB X.Q.

北 京 异 特 路 智 能 通 讯 科 技 有 限 公 司

TEL:010-62977213 FAX:010-62977237

www.itrob.cn

www.itrob.com.cn

E-mail:itrob@sina.com