

IP71X 以太网接口通讯协议 V1.8

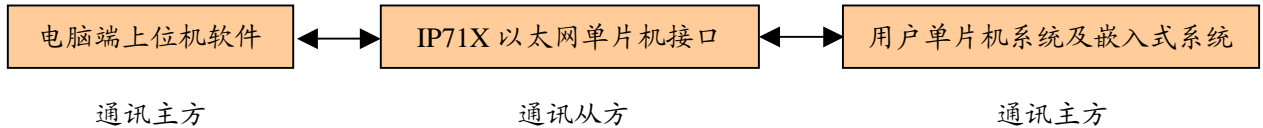
2008 年 08 月 02 日

IP71X 系列以太网单片机接口模块与电脑端连接，通讯由电脑端发起，电脑端为通讯主方。IP71X 与用户单片机连接，通讯由用户单片机发起，IP71X 插入电脑端与用户单片机之间，总是充当通讯从方的角色。

电脑端的上位机软件简称为电脑端。

IP71X 系列以太网单片机接口模块简称为 IP71X。

用户单片机系统及嵌入式系统简称为单片机。



IP71X 组成的以太网单片机通讯线路示意图

IP71X 定义的指令包采用 Windows Socket 传输，使用 UDP 协议，传输速度极快。

IP71X 使用 IPB 总线协议，第 1 个字节为长度，第 2 个字节为类型，最后 1 个字节为检验码。

IPB 的长度最大分为最短 32 字节、标准 138 字节及最长 1040 字节 3 种格式。

每个产品的最大长度不一定相同，以各自产品的用户手册为准。

每包指令的长度为 2~138 字节，故定义的发送及接收缓冲区长度为 138 字节。



IPB 指令包格式示意图

字节 1 为**指令长度**，其值为每包指令的长度，包括字节 2 本身的长度，但不包括检验码的长度，有效范围为 IPB 总线所定义的最大长度，超出范围将被视为无效指令包。

字节 2 为**指令包类型**。在系统应用中使用了数值 0xff~0xf4，故用户可使用的类型范围为 0xf3~0x00。

字节 n 为**检验码**，其值从第 1 字节至第 n-1 字节相加之和。接收方可以根据这个检验码判断这个包的合法性。如果不合法则放弃这个包。

IP71X 扩展的 EEPROM 记忆体以扇区为单位，每个扇区为 512 字节。在写入扇区的地址时，例如在 0x0000H、0x0200H 等时，如果标志有效则自动擦除相应的扇区。

Socket 的发送及接收需要指定 IP 地址及端口号，其中端口号 0~1024 总是被 Windows 使用。修改端口号时请避免使用其他应用软件常用的端口号。

IP71X 传输协议指令类型如下表所示：



深圳市龙珠科技有限公司

Hard & Soft Technology Co., LTD.

<http://www.HSAV.com>

地址:深圳市西乡龙珠路 99 号 2 楼

技术支持: support@HSAV.com

hsavd110.pdf

电话/传真:0755-27951479 27950879

业务联系: sales@HSAV.com

2008 年 08 月 02 日



指令类型(字节 2)	说明	传输协议
0xff	系统设置及 IP 查询类指令包。	UDP
0xfe	HTTP 超文本传输协议指令包。	TCP
0xfd	FTP 文件传输协议指令包。	TCP
0xfc	SMTP 邮件传输协议指令包。	TCP
0xfb	TFTP 普通文件传输协议指令包。	TCP
0xfa	DNS 域名服务器协议指令包。	TCP
0xf9	DHCP 动态主机配置协议指令包。	TCP
0xf8	TELNET 虚拟终端访问协议指令包。	TCP
0xf7	ARP 地址解析协议、ICMP 网际控制及 IGMP 网际组报文协议。	ICMP
0xf6 至 0xf4	系统保留。	TCP/UDP
0xf3 ~ 0x00	透明传输协议, 用户可以自行定义。	TCP/UDP

其中系统设置及 IP 查询类指令包为 IP71X 系统用于查询及修改之用。使用 UDP 传输协议。UDP 为平等的传输方式, 没有服务器及客户端之分, 所有 IP 都是平等的, 可以由任意 IP 发起指令包。IP71X 规定所有指令包都是由电脑端发起。IP71X 接收后, 分析相关指令, 如果需要传输给单片机, 则将相应的指令包转送给单片机, 否则自行处理完后回应给电脑端。电脑端总是需要知道 IP71X 的 IP 地址及端口号, 发送指令包时, 可以使用广播形式发送给局域网所有 IP, 或者使用单播指令发送给相应 IP。

IP71X 有 1 个“IP 查找”指令包是用于用户主机查找 IP 产品之用, 可以接受“255.255.255.255”的广播形式, 而且端口号可以为任意。

电脑端可以先使用“IP 查找”广播包来查找网络之中的所有 IP71X 产品。IP71X 会回应相应的指令包指示其产品的类型等信息。

电脑端依据“IP 查找”获得所有 IP 的信息。

◆ 系统设置及 IP 查询类指令说明

字节 1 为长度。

字节 2 为指令包类型, 这里固定为 0xff。

字节 3 为子类选择。

最后一个字节为检验码, 其值为从第 1 个字节到最后字节 (不含检验码本身) 相加之和。

如果是两个字节组成 16 位的参数, 则第 1 个字节为低位, 第 2 个字节为高位。

如果是 4 个字节组成 32 位的参数, 则第 1 个字节为低位, 第 4 个字节为高位。

IP 地址总是高位在前, 例如“192.168.0.1” 则第 1 个字节为 192, 第 4 个字节为 1。

0xnn 表示所描述的值不确定, 可能为任意值。但其值为原先约定的范围, 例如指令长度为 2~137。

B7 表示位于字节的第 7 位, B6 表示位于字节的第 6 位, 以此类推。



◆ IP 查询类指令包说明

指令长度	子类选择	说明 字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择
0x03	0xfe	<p>IP 查找, 电脑端一般使用广播方式发送给所有的 IP。用于确定 IP71X 是否连接。</p> <p>发送: 0x03/0xff/0xfe</p> <p>回应: 0x1b/0xff/0xfe/参数 0 ~ 参数 24 参数 0 ~ 参数 5 为 IP71X 的 MAC 地址; 参数 6 ~ 参数 9 为 IP71X 的网关地址, 在使用路由器连接的网络时必须设置; 参数 10 ~ 参数 13 为 IP71X 的 IP 地址; 参数 14 ~ 参数 15 为 IP71X 的端口号; 参数 16 ~ 参数 19 为 IP71X 的 IP 地址掩码; 参数 20 ~ 参数 21 为通讯接口方式选择指示, 见“单片机通讯接口方式”的说明; 参数 22 ~ 参数 23 为 IP71X 的产品类型指示, 见“IP71X 的产品类型指示”的说明。 参数 24 为 IP71X 的模块型号指示, 见“IP71X 的模块型号指示”的说明。</p> <p>注意: 只有 IP 查找的指令包接受“255.255.255.255”这种广播形式。其余指令包必须要 IP 地址及端口号对应才为有效的指令包。</p>
0x12	0xfd	<p>IP 地址修改, 适用于电脑端与单片机端。电脑端必须采用单播形式指定相应的 IP。</p> <p>发送: 0x12/0xff/0xfd/参数 0 ~ 参数 24 参数 0 ~ 参数 3 为 IP71X 原来的 IP 地址; 参数 4 ~ 参数 5 为 IP71X 原来的端口号; 参数 6 ~ 参数 9 为 IP71X 待修改的网关地址; 参数 10 ~ 参数 13 为 IP71X 待修改的 IP 地址; 参数 14 ~ 参数 15 为 IP71X 待修改的端口号; 参数 16 ~ 参数 19 为 IP71X 待修改的 IP 地址掩码; 参数 20 ~ 参数 21 为 IP71X 待修改的通讯接口方式, 见“单片机通讯接口方式”的说明; 参数 22 ~ 参数 23 为 IP71X 待修改的产品类型, 见“IP71X 的产品类型指示”的说明。 参数 24 为修改模块允许码, 其值为 IP71X 的模块型号指示的反码。</p> <p>回应: 0x04/0xff/0xfd/参数 0 参数 0 为 0x00 表示指令处理成功; 参数 0 为 0x01 表示指令处理失败。</p>
0x03	0xfc	<p>IP 同步。适用于电脑端。</p> <p>发送: 0x03/0xff/0xfc</p> <p>回应: 0x03/0xff/0xfc</p> <p>注意: 有些电脑端在与 IP 通讯一待时间没有收到指令包后会关闭相应的 IP, 导致从 IP71X 发送的指令包在电脑端收不到的现象。这时, 可以每隔几秒钟时间使用 IP 同步指令来维持电脑端与 IP71X 的通讯。</p>



◆ IP 产品别称及显示类指令包说明

指令长度	子类选择	说明 字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择
0x03	0xee	<p>查询 IP71X 的产品别称, 适用于电脑端与单片机。</p> <p>发送: 0x03/0xff/0xee</p> <p>回应: 0xnn/0xff/0xee/参数 0~参数 n 参数 0 为产品别称的内码字节长度; 参数 1 为文字内码说明, 为 0x01 表示使用 UNICODE 的内码, 这时所有的内码包括 ASC II 都有 16 位的, 适合于使用 UNICODE 的 Windows 平台。注意在 UNICODE 时注意将长度除以 2。为 0x02 为 GB2313 内码; 参数 2~参数 n 为产品别称的内码, 在 UNICODE 时必须为双数。为 0x02 为 GB2313 内码。</p>
0x03	0xed	<p>IP71X 的产品别称修改, 适用于电脑端与单片机。</p> <p>发送: 0x03/0xff/0xed/参数 0~参数 n 参数 0 为产品别称的内码字节长度; 参数 1 为文字内码说明, 为 0x01 表示使用 UNICODE 的内码, 这时所有的内码包括 ASC II 都有 16 位的, 适合于使用 UNICODE 的 Windows 平台。注意在 UNICODE 时注意将长度除以 2。为 0x02 为 GB2313 内码;</p> <p>回应: 0x04/0xff/0xed/参数 0 参数 0 为 0x00 表示指令处理成功; 参数 0 为 0x01 表示指令处理失败。</p>
没有发送的指令包	0xec	<p>IP71X 模块调试显示, 将单片机需要显示的调试数据在电脑端显示, 适用于电脑端与单片机。</p> <p>发送: 单片机端为发送, 电脑端不能发送。 0xnn/0xff/0xec/参数 0~参数 n 参数 0~参数 n 为单片机需显示的数据。电脑端使用指令长度来获得显示的长度, 通常电脑端提供一个显示窗口, 将本指令包的数据以 16 进制的方式显示, 这样可以方便地调试单片机的程序。</p> <p>回应: 0xnn/0xff/0xec/参数 0 参数 0 为 0x00 表示指令处理成功; 参数 0 为 0x01 表示指令处理失败。</p>



◆ 以太网连线及 IP 地址类指令包说明

指令长度	子类选择	说明 字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择
没有发送的指令包	0xde	<p>通讯主方的 IP 地址及端口, 在电脑端主机变化时, 自动回传给单片机, 只适用于单片机端。</p> <p>发送: 自动回传, 没有发送的指令包。</p> <p>回应: 0x09/0xff/0xde/参数 0~参数 6 参数 1~参数 4 为通讯主方的 IP 地址, 参数 5~参数 6 为通讯主方的端口号。</p> <p>注意: IP 地址及端口在变化与多主机通讯时需要使用。例如原来是 192.168.0.12 电脑端主机与 IP71X 通讯的, 现在 IP71X 收到 192.168.0.13 的指令包, 则 IP71X 会自动回传给单片机。 在这个指令包后面的指令包为当前 IP 地址的电脑端发出的。 电脑端不使用这个指令, 只有单片机端有效。</p>
0x0c	0xdd	<p>单片机选择新的电脑端主机方的 IP 地址及端口选择, 只适用于单片机端。</p> <p>发送: 0x09/0xff/0xdd/参数 0~参数 5 参数 0~参数 3 为电脑端的 IP 地址; 参数 4~参数 5 为电脑端的端口号。</p> <p>回应: 0x04/0xff/0xdd/参数 0 参数 0 为 0x00 表示指令处理成功; 参数 0 为 0x01 表示指令处理失败。</p> <p>注意: IP 地址及端口在与多主机通讯时需要使用。 在这个指令包后面的指令包为当前 IP 地址的单片机发出的。 电脑端不使用这个指令, 只有单片机端有效。</p>
没有发送的指令包	0xdc	<p>IP71X 与以太网连线状况指示, 只适用于单片机端。</p> <p>发送: 自动回传, 没有发送的指令包。</p> <p>回应: 0x04/0xff/0xdc/参数 0 参数 0 为 0x01 表示 IP71X 与以太网连线正确; 为 0x00 表示没有连线。</p>



◆ IP71X 扩展功能类指令包说明

字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择

指令长度	子类选择	说明 字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择
0xnn	0xce	<p>IP71X 模块 EEPROM 记忆体读出, 写入, 适用于电脑端与单片机。</p> <p>发送: 0xnn/0xff/0xbe/参数 0~参数 n 参数 0 为 0 : 电脑端读 EEPROM 数据。 参数 0 为 1 : 电脑端向 EEPROM 写数据。 参数 1~参数 2 为 16 位的 EEPROM 地址, EEPROM 地址的有效范围为 0x0000~0x19ff. 参数 3 为每次读取或写操作时数据的长度, 参数 4~参数 n 为电脑端向 EEPROM 写入的数据在读取时不需要。</p> <p>回应: 0xnn/0xff/0xbe /参数 0, ... 参数 0 为 0 : 单片机 EEPROM 向电脑端回转数据. (数据 1, 数据 2...) 参数 0 为 1 : 电脑端向 EEPROM 写数据正确, (后面没有数据)</p> <p>注意: 擦除扇区指令将擦除全部的 512 字节。擦除的地址必须为相应的扇区地址, 如果不是扇区地址则 IP71X 不执行擦除动作, 同时回应擦除扇区不成功。用户使用擦除扇区之前, 如果有一些有用的数据在待擦除的扇区之中, 则应该将有用的数据先保存在其他扇区之后, 再擦除相应的扇区, 这样保证有效的数据不会丢失。</p>
0x06	0xcb	<p>IP71X 温度指示, 适用于电脑端与单片机。</p> <p>发送: 0x06/0xff/0xcb/参数 0 参数 0 为 0 则只读取一次温度值, 其余值则指示 IP71X 定时自动回应温度值, 以秒为单位。</p> <p>回应: 0xnn/0xff/0xcb/参数 0~参数 1 参数 0~参数 1 为温度值, 个位为小数点。</p>

◆ IP71X 远程端口类指令包说明



字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择

指令长度	子类选择	说明 字节 1 为指令长度, 字节 2 为 0xff, 字节 3 为子类选择
0xnn	0xbe	<p>读取或以修改远程端口的主机列表, 适用于电脑端。</p> <p>发送: 0xnn/0xff/0xbe/参数 0~参数 n, 参数 0 为 0x00 表示电脑端从 IP71X 读取主机列表 参数 0 为 0x01 表示电脑端更新主机列表到 IP71X 参数 1~参数 6 为第 1 组主机列表的 4 个字节 IP 地址及 2 个字节的端口号。 参数 n 为以 6 个字节为 1 组的主机列表, 最大支持 8 组的主机列表。 当为读取主机列表时参数 1~参数 n 不存在。</p> <p>回应: 0xnn/0xff/0xbe/参数 0~参数 n, 参数 0 为 0x00 表示从 IP71X 返回主机列表, 这时参数 1~参数 n 为主机列表, 以 6 个字节为 1 组, 最大为 8 组。 参数 0 为 0x01 则为电脑端写入主机列表 IP71X。这时参数 1 为 0 表示更新列表成功。</p>
0x06	0xbd	<p>输出数据到远程端口, 适用于电脑端与单片机。</p> <p>发送: 0x05/0xff/0xbd/参数 0~参数 1, 参数 0 为远程端口的输出端口输出电平, B7 对应 GP7, B6 对应 GP6 以此类推。 参数 1 为远程端口的控制有效标志。 其中 B7 及 B5 保留为 0, B4 为 1 表示修改输出端口的值有效。 即必须 B4 为 1 时, 参数 0 的 B7~B0 及参数 1 的 B3~B0 才有效。 参数 1 的 B3 对应 GP11 的输出电平, B2 对应 GP10, B1 对应 GP9, B0 对应 GP8。</p> <p>回应: 0xnn/0xff/0xbe/参数 0~参数 n, 参数 0 为 0x00 表示指令处理成功; 参数 0 为 0x01 表示指令处理失败。</p>
0x03	0xbc	<p>从远程端口读取数据, 适用于电脑端与单片机。</p> <p>发送: 0x03/0xff/0xbc 没有参数, 如果需要随机读取远程端口的数据则需要发送。平时远程端口有任何数据变化时会主动返回, 无需电脑端发送本指令包。</p> <p>回应: 0x07/0xff/0xbc/参数 0~参数 5 参数 0 为远程端口输入的电平值, B7 对应 GP7, B6 对应 GP6 以此类推。 参数 1 的 B7 保留, B6 为 1 表示所回应的数据中, 有效部分包括第一个键盘的键值, B5 为 1 表示有效部分包括第二个键盘的键值, B4 为 1 表示为通用输入端口的值。 B6、B5 及 B4 可能同时为 1 或为任意的组合, 当电脑端发送读取端口时, 这 3 个位都为 0。 B7 为 1 表示指令包是收到电脑端的指令而回传的, 并不是数据有变化主动回传的。除非有特别的应用, 一般不建议使用电脑端读取数据的方式, 因为这样无法判断键盘是更新值或者是原来的值。 参数 2~参数 3 为第一个 PS2 键盘的键值。 参数 4~参数 5 为第二个 PS2 键盘的键值。</p>

◆ IP71X 与单片机通讯接口方式说明

IP71X与单片机之间可以选择使用I²C、SPI及UART等三种硬件通讯方式，其中I²C及SPI通讯方式在单片机端为主方，可以随时向IP71X写入指令包，或者读取保存在IP71X的指令包。

第1个参数与第2个参数组成16位的值，每个位表示如下：

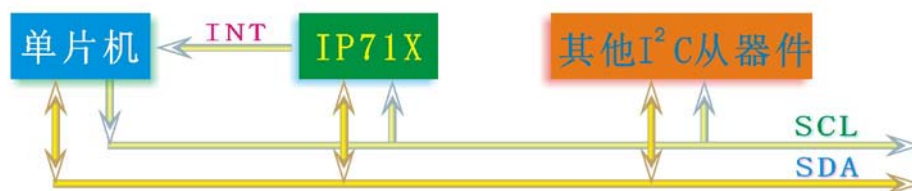
B15为修改允许位。只有B15为1时才允许从电脑端或单片机端修改通讯方式。当从单片机端修改通讯方式时，单片机读取到修改成功的回应后修改相应的通讯方式。接着单片机必须采用新的通讯方式进行通讯。一般不建议使用单片机修改通讯方式。

B11~B8为IP71X与单片机接口使用的硬件类型选择。0001为使用I²C硬件通讯接口；0010为使用SPI硬件通讯接口I²C通讯；0100为使用UART硬件通讯接口。其余值保留。

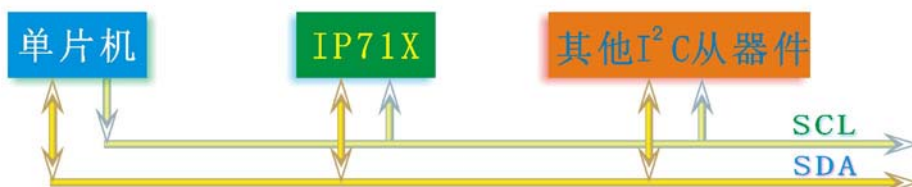
当硬件通讯接口选择为I²C时：

B4为0则使用INT对单片机产生中断，在IP71X有指令包变化时，INT变化通知单片机。

B14为1则只使用SCL及SDA与单片机连接，单片机只能定时查询来获取指令包变化的情况。



带INT的I²C通讯方式



只需两根线的I²C通讯方式

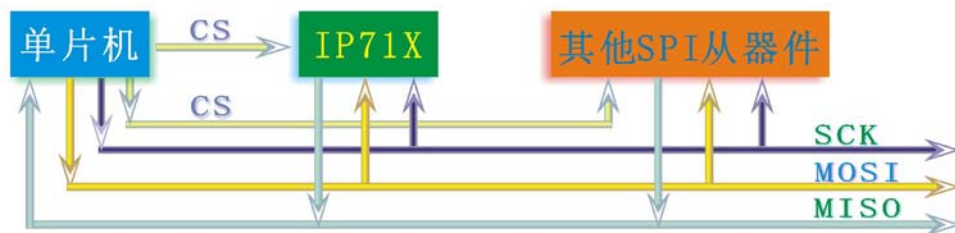
B7 ~ B0 为 I²C 从机的地址, 与 I²C 标准的地址定义想对应。

B0 位为 0, 如果单片机读 IP71X 时, 则需在发送从机地址时 B0 写为 1, 这个与 I²C 规范是一致的。

当硬件通讯接口选择为 SPI 时, IP71X 与单片机之间使用 SCK、MOSI、MISO 及 CS 片选共 4 个端口: B9 与 B8 为 SPI 模式选择, 见下表所示。

模式	驱动的第一个数据	驱动的有关数据	采样的数据	SCK 电平
00	在第 1 个 SCK 上升沿之前	SCK 下降沿	SCK 上升沿	低电平
01	第 1 个 SCK 上升沿	SCK 上升沿	SCK 下降沿	低电平
10	在第 1 个 SCK 下降沿之前	SCK 上升沿	SCK 下降沿	低电平
11	第 1 个 SCK 下降沿	SCK 下降沿	SCK 上升沿	低电平

IP71X 支持 SPI 总线与用户主机相连, 用户主机只需 4 根普通的 I/O 口即可实现。两者的连线如下图所示:



高速 4 线 SPI 通讯方式示意图

SPI 总线使用 CS、SCK、MOSI、MISO 4 根线, 如果用户主机不内置 SPI 控制, 可以使用软件方式产生 SPI 时序。当用户主机系统中有多多个 SPI 总线的器件时, 可以将 SCK、MOSI、MISO 与 IP71X 的 SCK、MOSI、MISO 并联使用。在 CS 为高时, IP71X 的 MISO 为高阻状态。标准的 SPI 器件在 CS 片选为无效时, MISO 都为高阻状态。

当使用 SPI 总线时, 用户主机定时使用 MOSI 写入 0x00, 读到 MISO 的值不为 0x00 时表示 IP71X 已经接收到由服务器发送的数据包。MISO 为 0x07 时有 1 包未读的指令包, 0x02 为 2 包, 如此类推。在 CS 片选变低后发送 1 个字节的 0xff 后变高 CS, 则只是读取状态寄存器, 完成后 CS 变高恢复 SPI 的空闲状态。当 CS 变低后用户主机写入 0x00 则准备读取 MISO 的数据, 第 1 字节读回 MISO 的值为未读指令包的计数器, 这时, 必须保持 CS 为低, 直到所有的字节都读完为止。

第 2 个字节为指令包的长度, 与指令集的长度相一致, 其有效值为 0x02 至 138。用户主机根据长度的值读完所有的字节。在读取时, MOSI 发出的值应为 0xff。

用户主机需要发送数据包时, 第 1 个字节为长度, 有效值为 2 至 138, 与指令集的长度相一致。用户主机必须写入相同于长度的指令包数据。

在 CS 变低的第 1 个字节为 0xff 时, 表示用户主机未读取指令包的状态为 0x00 时表示用户主机读取指令包, 0x02 ~ 0x8a 表示用户主机发送指令包。只有 0xff 时, CS 为一个字节 8 位的时间为低, 完成后即变高 CS。在读取或发送指令包时都必须完成所有的读写后 CS 才变高。

SPI 时钟最高时钟为 6MHz。故采用 SPI 的通讯速率最高为 6Mbps。与 I²C 及 UART 相比速度是最高的。



当系统中有其他SPI器件或用户主机有多余IO端口的情况下，可以优先使用SPI总线获得更快的通讯速度。

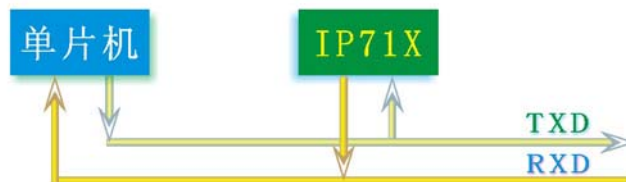
当选择为 UART 通讯时，IP71X 与单片机之间使用 RXD 及 TXD 接口。

B7 ~ B5 为奇偶检验位选择。

B3 ~ B0 为 UART 传输数据位数选择。

B4 为停止位选择。

B14 为 UART 格式选择。为 0 时使用 IP71X 全部的协议，这样，单片机可以使用 IP71X 的所有功能，传输时包括指令长度、指令类型及检验码等；为 1 时使用透明传输，IP71X 将略去指令长度等，单片机将不能对 IP71X 作任何设置及查询。



标准 UART 串行接口通讯方式



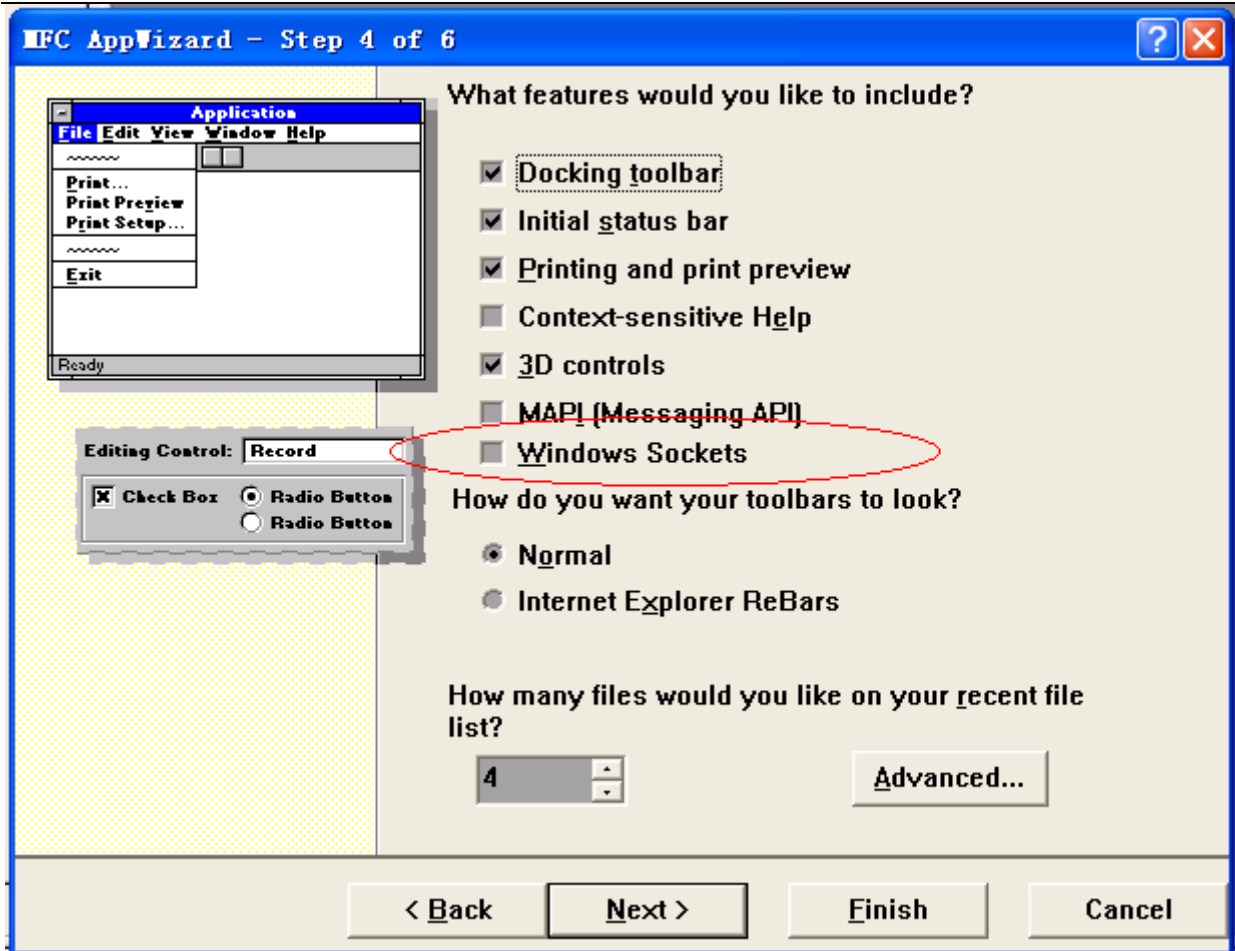
◆ 软件编写注意

编写 Windows 网络程序时，可以使用 Windows 的 Socket 动态库。Windows 对 Socket 作了详细的规范，可适合 VC、VB 等对网络直接编程。用户不需理会网络通讯的细节。

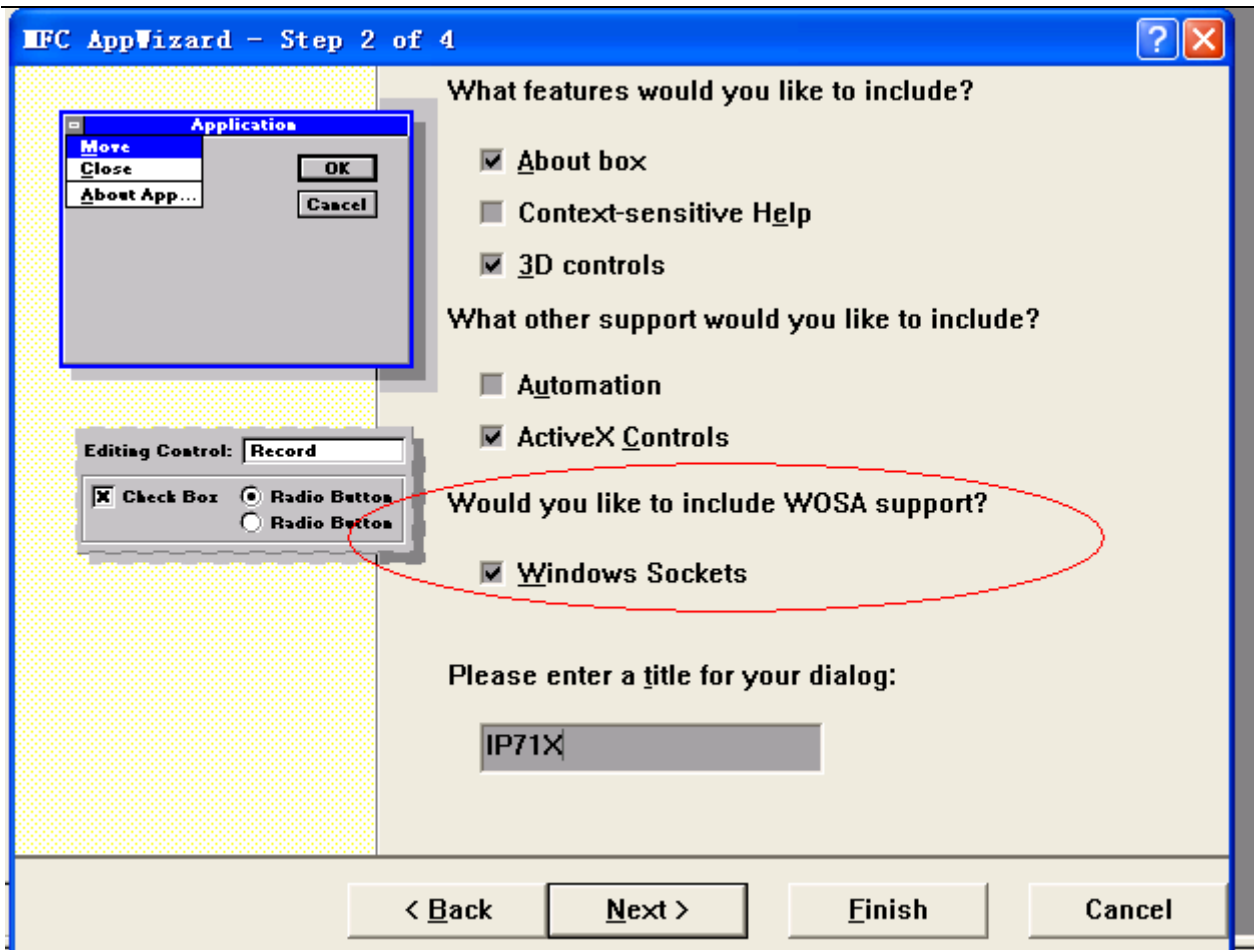
使用 Socket 非常简单，一般使用方法为选读取本机的 IP 地址，然后绑定网络即可使用 Socket。直接使用相关的函数发送及接收指令包即可。

以下以 VC 编程方式作为例子，简单地说明电脑端程序的编写方法。

使用 Socket 的方法分为两种，第一种可以在新建项目时 Windows Sockets 的支持，当使用为 Multiple documents 时，在 Step 4 of 6 处勾上 “Windows Sockets” 选项，如下图红色圈处所示。



如果为 Dialog Based 类型的，在 Step 2 of 4 时在“Would you like to include WOSA support?” 的选项处勾选“Windows Sockets” 选项，如下图红色圈处所示。



如果是在旧的项目之中加入 Socket，可以在项目.cpp(注意不是项目 Dlg.cpp)的 InitInstance()之中加入以下语句：

```
if (!AfxSocketInit()) {  
    AfxMessageBox(IDP_SOCKETS_INIT_FAILED);  
    return FALSE;  
}
```

```
void CSM311_PS2Dlg::METN_Initialize(){  
    char hostname[100];  
    gethostname(hostname, 1000);
```

```
    m_strLocalIP = "192.168.0.1"; // 电脑端主机的缺省地址。用户可以修改为自己的缺省地址。
```



```
m_constPort = 65533; // 端口号。这是 IP71X 缺省的端口号。  
// 用户可以根据实际的端口号进行修改获得本机的 IP 地址。实际的 IP 地址允许使用广播的方式。  
  
// 初始化完成。用户设置一个定时器，在固定的时间读取是否有新的指令包，有则接收并作相应的处理。  
广播方式发送可以将同一个指令包发送给同一个网络内的所有 IP，在确定待发送的数据包为所有 IP 都需接收时可以使用。
```

```
struct hostent *host = gethostbyname(hostname);  
if (host != NULL){  
    m_strLocalIP = inet_ntoa(*(in_addr*)host->h_addr);  
}  
if (m_socket.Create(m_constPort, SOCK_DGRAM, m_strLocalIP) == FALSE){ // 初始化 socket  
    AfxMessageBox("初始化 socket 失败"); // 如果初始化失败，显示提示对话框  
    OnCancel(); // 退出程序  
}  
BOOL bBroadcast = TRUE; // 设置 m_socket 属性，允许广播  
m_socket.SetSockOpt(SO_BROADCAST, &bBroadcast, sizeof(BOOL));  
return;  
}  
  
void CSM311_PS2Dlg::METN_Polling(){ // 定时器  
    CString fromIP; // 对方 IP  
    UINT g2Local_1; // 对方端口  
  
    fd_set in_set = {0};  
    timeval t = {0,0};  
    FD_SET(m_socket.m_hSocket, &in_set);  
    if (select(1, &in_set, NULL, NULL, &t)){ // 测试是否有可以接收的数据  
        // 用于接收数据的缓冲区地址 缓冲区大小 源地址 源端口  
        if (m_socket.ReceiveFrom(gIPB_RX_Buffer, 138, fromIP, g2Local_1)){ // 从 socket 上接收数据  
            MIPB_RX_Buffer(fromIP, g2Local_1);  
        }  
    }  
    return;  
}  
  
void CSM311_PS2Dlg::METN_TX_Command(CString gsLocal_1){  
    if (gsLocal_1 == ""){ // 广播  
        m_socket.SendTo(  
            LPCTSTR(gIPB_TX_Buffer), // 数据报内容
```




```
(gIPB_TX_Buffer[0] + 1),           // 数据报长度
m_constPort,                       // 对方的端口, 本程序使用统一的端口
NULL                                // NULL 表示数据报广播, 其余值为对方的地址
);
}
else {                               // 单播
    m_socket.SendTo(LPCTSTR(gIPB_TX_Buffer), (gIPB_TX_Buffer[0] + 1), m_constPort, gsLocal_1);
}
return;
}
```

电脑端在未知道 IP71X 产品连接的情况时, 先使用广播地址发送“IP 查找”指令包。
收到相关回应指令包后表示连接成功。
可以发送其他指令包进行相应的动作了。