

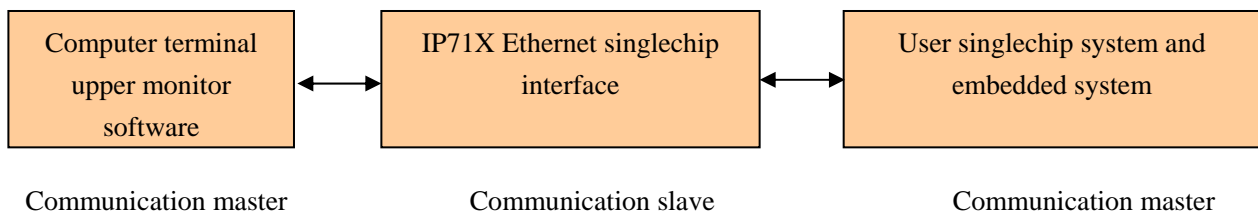
IP71X Ethernet Interface Communication Protocol V1.6

IP71X series Ethernet singlechip interface module connects with computer terminal; the computer initiates communication, and computer terminal is communication master. IP71X connects with user singlechip, and user singlechip initiates communication. IP71X inserts the middle of the computer terminal and singlechip, and communication slave is always IP71X.

The upper monitor software of computer terminal is called computer terminal for short.

IP71X series Ethernet singlechip interface module is called IP71X for short.

User singlechip system and embedded system are called singlechip for short.



The Ethernet singlechip communication connection that IP71X composes

The instruction packet defined by IP71X adopts Windows Socket transfers, and uses UDP protocol. Transfer speed is very fast.

IP71X uses IPB bus protocol, the first byte is length, the second byte is type, and the final byte is checking code.

IPB length can be divided into 3 kinds of formats at most, i.e. the most short 32 bytes, standard 138 bytes, and the most long 1040 bytes.

The maximal length of each product is not always same; users should refer to the user manual of each product respectively.

The length of each package instruction is 2 to 138 bytes, so sending and receiving buffer area length defined is 138 bytes.



IPB instruction packet format

Byte 1 is **instruction length**, its value is the length of each package instruction, include the length of byte 2, but don't include the length of checking code, effective range is the maximal length IPB bus defines, exceeding range will be thought that it is a invalid instruction packet.



Hard & Soft Technology Co., LTD.

<http://www.HSAV.com>

Address: second floor, No.199, Longyin 2nd Road, Xixiang, Shenzhen, China

TEL: 86-0755-27951479 27950879

FAX: 86-0755-27950879-213

Technology support: support@HSAV.com

Business contact: Sales@HSAV.com



Byte 2 is **instruction packet type**. Use numerical value 0xff to 0xf4 in system application, so the type range that users can use is 0xf3 to 0x00.

Byte n is **checking code**; its value is the sum adding from the byte 1 to the byte n-1. The receiving end can judge the validity of this packet according to this checking code. If this packet is not valid, this packet will be abandoned.

The unit of IP71X extended EEPROM memory is sector, and each sector is 512 bytes. When write sector, for instance 0x0000H、0x0200H, automatically erase corresponding sector if mark is valid.

The sending and receiving of Socket need to appoint IP address and port number, and port numbers from 0 to 1024 are always used by Windows. Please don't use the port number that other application software use usually.

IP71X transfer protocol instruction type is as follows:

Instruction type (byte 2)	Instruction	Transfer protocol
0xff	System setup and IP query kind instruction packet.	UDP
0xfe	HTTP hypertext transfer protocol instruction packet.	TCP
0xfd	FTP file transfer protocol instruction packet.	TCP
0xfc	SMTP message transfer protocol instruction packet.	TCP
0xfb	TFTP ordinary file transfer protocol.	TCP
0xfa	DNS domain name server protocol instruction packet.	TCP
0xf9	DHCP dynamic host configuration protocol instruction packet.	TCP
0xf8	TELNET virtual terminal access protocol instruction packet.	TCP
0xf7	ARP address resolution protocol, ICMP internet control messages protocol, and IGMP internet group messages protocol.	ICMP
0xf6 to 0xf4	System reserved.	TCP/UDP
0xf3 to 0x00	Transparent transmission protocol, user can define it at will.	TCP/UDP

System setup and IP query kind instruction packet are used for query and modification by IP71X system. Use UDP transfer protocol. UDP is the equal transfer protocol, and isn't divided into sever and client; all IP are equal, any IP can initiate instruction packet. IP71X requires that all instruction packets should be initiated by computer terminal. Analyze the relevant instructions after IP71X receive instruction packet, and transfer corresponding instruction packet to singlechip if there is need to transfer to singlechip, or respond to computer terminal after processing by itself. Computer always need to know the IP address and port number of IP71X, can use broadcast mode to send all IP to LAN or use unicast instruction to send to the corresponding IP while sending instruction packet.

IP71X has one "IP find" instruction packet used to find IP product by user mainframe, can accept the broadcast format of "255.255.255.255", port number can be any port number.

Computer terminal can first use "IP find" broadcast packet to find all IP products in the network. IP71X will respond the corresponding instruction packet to indicate the type of product and so on.

Computer obtains the information of all IP according to "IP find".



◆ System setup and IP query kind instruction instructions

Byte 1 is length.

Byte 2 is instruction packet type, it is fixed as 0xff.

Byte 3 is subclass selection.

The final byte is checking code; its value is the sum adding from the first byte to the final byte (don't contain checking code itself).

If 16-bit parameter is composed of two bytes, the first byte is low bit, and the second byte is high bit.

If 32-bit parameter is composed of four bytes, the first byte is low bit, and the fourth byte is high bit.

It is certain for IP address that high bit is always in front, for instance "192.168.0.1", the first byte is 192, and the fourth byte is 1.

0xnn means the value described is uncertain, and it may be any value. But its value is the range fixed originally, for instance, instruction length is 2 to 137.

B7 means the seventh bit of byte, and B6 means the Sixth bit of byte, the rest may be deduced by analogy.

◆ IP query kind instruction packet instructions

Instruction length	Subclass selection	Instruction byte 1 is instruction length , byte 2 is 0xff, byte 3 is subclass selection .
0x03	0xfe	<p>IP find, computer use broadcast to send all IP in general. It is used to confirm whether IP71X has connected.</p> <p>send: 0x03/0xff/0xfe</p> <p>Return: 0x1b/0xff/0xfe/parameter 0 to parameter 24 Parameter 0 to parameter 5 is the MAC address of IP71X; Parameter 6 to parameter 9 is the gateway address of IP71X, must be set when use the network connected by router; Parameter 10 to parameter 13 is the IP address of IP71X; Parameter 14 to parameter 15 is the port number of IP71X; Parameter 16 to parameter 19 is the address mask of IP71X; Parameter 20 to parameter 21 is communication interface mode selection indication, please see "singlechip communication interface mode" instructions; Parameter 22 to parameter 23 is the product type indication of IP71X, please see "the module model indication of IP71X" instructions; Parameter 24 is the module model indication of IP71X; please see "the module model indication of IP71X" instructions.</p> <p>attention: Only IP find instruction packet can accept "255.255.255.255" broadcast format, and only when IP address corresponds to port number the rest instruction packets are valid.</p>
0x12	0xfd	<p>IP address modification. It is suitable for computer terminal and singlechip terminal. Computer terminal must adopt unicast format to appoint the corresponding IP.</p> <p>send: 0x12/0xff/0xfd/parameter 0 to parameter 24 Parameter 0 to parameter 3 is the original IP address of IP71X; Parameter 4 to parameter 5 is the original port number of IP71X; Parameter 6 to parameter 9 is the IP71X gateway address waiting for modification; Parameter 10 to parameter 13 is the IP71X IP address waiting for modification; Parameter 14 to parameter 15 is the IP71X port number waiting for modification; Parameter 16 to parameter 19 is the IP71X IP address mask waiting for</p>



		<p>modification;</p> <p>Parameter 20 to parameter 21 is the IP71X communication interface mode waiting for modification, please refer to “singlechip communication interface mode” instructions;</p> <p>Parameter 22 to parameter 23 is the PI71X product type waiting for modification. Please refer to “the product type indication of IP71X” instructions.</p> <p>Parameter 24 is modification module permission code; its value is the complement of IP71X module model indication.</p> <p>Return:</p> <p>0x04/0xff/0xfd/parameter 0</p> <p>Mean instruction processing succeeds if parameter 0 is 0x00; mean instruction processing fails if parameter 0 is 0x01.</p>
0x03	0xfc	<p>IP synchronization. It is suitable for computer terminal.</p> <p>send:</p> <p>0x03/0xff/0xfc</p> <p>Return:</p> <p>0x03/0xff/0xfc</p> <p>attention:</p> <p>If computer terminal doesn't receive instruction packet after computer terminal communicates with IP for some time, the corresponding IP may be closed, so computer terminal may not receive the instruction packet IP71X sends. At this time, can use IP synchronization instruction to maintain communication between computer terminal and IP71X every few seconds.</p>

◆ IP product name and display kind instruction packet instructions

Instruction length	Subclass selection	Instruction byte 1 is instruction length , byte 2 is 0xff, byte 3 is subclass selection .
0x03	0xee	<p>Query the product name of IP71X; it is suitable for computer terminal and singlechip.</p> <p>send:</p> <p>0x03/0xff/0xee</p> <p>Return:</p> <p>0xnn/0xff/0xee/parameter 0 to parameter n</p> <p>Parameter 0 is the ISN byte length of product name.</p> <p>Parameter 1 is character ISN instructions, when it is 0x01 mean UNICODE ISN, at this time, all ISN including ASC II have 16 bits, and apply to the Windows platform that use UNICODE. Please note that length should divide by 2 when UNICODE. When it is 0x02 mean GB2313 ISN.</p> <p>Parameter 2 to parameter n is the ISN of product name, it must be double number when UNICODE. When it is 0x02 mean GB2313 ISN.</p>
0x03	0xed	<p>The product name modification of IP71X, it is suitable for computer terminal and singlechip.</p> <p>send:</p> <p>0x03/0xff/0xed/parameter 0~parameter n</p> <p>Parameter 0 is the ISN byte length of product name.</p> <p>Parameter 1 is character ISN instructions, when it is 0x01 mean UNICODE ISN, at this time, all ISN including ASC II have 16 bits, and apply to the Windows platform that use UNICODE. Please note that length should divide by 2 when UNICODE. When it is 0x02 mean GB2313 ISN.</p> <p>Return:</p> <p>0x04/0xff/0xed/parameter 0</p> <p>Parameter 0 is 0x00 means instruction processing succeeds; parameter 0 is 0x01 means instruction processing fails.</p>



Instruction packet that is not sent	0xec	<p>IP71X module debugging display, make debugging data that singlechip needs to display show on the computer terminal. It is suitable for computer terminal and singlechip.</p> <p>send: Singlechip terminal is sending, and computer terminal cannot send. 0xnn/0xff/0xec/parameter 0 to parameter n Parameter 0 to parameter n is the data that singlechip needs to display. Computer terminal uses instruction length to obtain the length of display; in general, computer terminal provide a display window, and make this instruction packet to be shown in hexadecimal, which can provide convenience for debugging the program of singlechip.</p> <p>Return: 0xnn/0xff/0xec/parameter 0 Mean instruction processing succeeds if parameter 0 is 0x00; mean instruction processing fails if parameter 0 is 0x01.</p>
-------------------------------------	------	--

◆ Ethernet connection and IP address kind instruction packet instructions

Instruction packet	Subclass selection	Instruction byte 1 is instruction length , byte 2 is 0xff, byte 3 is subclass selection .
Instruction packet that is not sent	0xde	<p>The IP address and port number of communication master, automatically return to singlechip when computer terminal changes. It is only suitable for singlechip terminal.</p> <p>send: Automatically return, the instruction packet that is not sent.</p> <p>Return: 0x09/0xff/0xde/parameter 0 to parameter 6 Parameter 1 to parameter 4 is the IP address of communication master, and parameter 5 to parameter 6 is the port number of communication master.</p> <p>attention: Need to be used when IP address and port change communication with multi-mainframe. For instance, 192.168.0.12 computer terminal mainframe communicates with IP71X formerly, but now IP71X receive 192.168.0.13 instruction packet, IP71X will automatically return it to singlechip. The computer terminal of the current IP address sends the instruction packet behind of this instruction packet. Computer terminal doesn't use this instruction packet, and only singlechip terminal uses it.</p>
0x0c	0xdd	<p>The IP address and port selection that singlechip chooses new computer mainframe. It is only suitable for singlechip.</p> <p>send: 0x09/0xff/0xdd/parameter 0 to parameter 5 Parameter 0 to parameter 3 is the IP address of computer terminal. Parameter 4 to parameter 5 is the port number of computer terminal.</p> <p>Return: 0x04/0xff/0xdd/parameter 0 Mean instruction processing succeeds if parameter 0 is 0x00; mean instruction processing fails if parameter 0 is 0x01.</p> <p>attention: Need to be used when IP address and port change communication with multi-mainframe. The computer terminal of the current IP address sends the instruction packet behind of this instruction packet. Computer terminal doesn't use this instruction packet, and only singlechip terminal uses it.</p>



Instruction packet that is not sent	0xdc	<p>IP71X and Ethernet connection status indication. It is only suitable for singlechip terminal.</p> <p>send: Automatically return, instruction packet that is not sent</p> <p>Return: 0x04/0xff/0xdc/parameter 0 If parameter 0 is 0x01 it means IP71X connects with Ethernet successfully, and if parameter 0 is 0x00 it means there is no connection.</p>
-------------------------------------	------	--

◆ IP71X expansion function kind instruction packet instructions

Byte 1 is **instruction length**, byte 2 is 0xff, byte 3 is **subclass selection**.

Instruction length	Subclass selection	Instruction byte 1 is instruction length , byte 2 is 0xff, byte 3 is subclass selection .
0x06	0xce	<p>IP71X module EEPROM memory read. It is suitable for computer terminal and singlechip.</p> <p>send: 0x06/0xff/0xce/parameter 0 to parameter 2 Parameter 0 to parameter 1 is 16-bit address, the effective range if the length of EEPROM. The effective range of IP71X is 0x0000 to 0x17ff. Parameter 2 is the length of data waiting for read. IP71X return corresponding data according to this length.</p> <p>Return: 0xnn/0xff/0xce/parameter 0 to parameter n Parameter 0 to parameter n is the data read.</p>
0x05	0xcd	<p>IP71X module EEPROM memory write. It is suitable for computer terminal and singlechip.</p> <p>send: 0x05/0xff/0xcd/parameter 0 to parameter n Parameter 0 to parameter 1 is the sector address waiting for write. Parameter 2 is the data length waiting for write. Parameter 3 to parameter n is the EEPROM data waiting for write.</p> <p>Return: 0x04/0xff/0xcd/parameter 0 Mean instruction processing succeeds if parameter 0 is 0x00; mean instruction processing fails if parameter 0 is 0x01.</p>
0x05	0xcc	<p>IP71X module EEPROM memory sector erase, it is suitable for computer terminal and singlechip.</p> <p>send: 0x05/0xff/0xcc/parameter 0 to parameter n Parameter 0 to parameter n is the sector address waiting for erasing.</p> <p>Return: 0x04/0xff/0xcc/parameter 0 If parameter 0 is 0x00 it means erasing sector succeeds, and if parameter 0 is 0x01 it means erasing sector fails.</p> <p>attention: Erasing sector instruction will erase all 512 bytes. The address erased must be the corresponding sector address. If not sector address, IP71X doesn't carry out erase action, at the same time responding erase sector fails. Before users use erase sector, if some data are in the middle of the sector waiting for erasing, the available data should be saved in the behind of the other sector, then erase the corresponding sector, such doing can make sue that the effective data is not lost.</p>



0x06	0xcb	<p>IP71X temperature indication. It is suitable for computer terminal and singlechip.</p> <p>send: 0x06/0xff/0xcb/parameter 0 Only read temperature value one time if parameter 0 is 0, and the rest value indicate IP71X to respond temperature value automatically at certain times. The unit is second.</p> <p>Return: 0xnn/0xff/0xcb/parameter 0 to parameter 1 Parameter 0 to parameter 1 is temperature value, and units digit is radix point.</p>
------	------	--

◆ IP71X and singlechip communication interface mode instructions

IP71X can use I²C, SPI and UART hardware communication modes to communicate with singlechip. I²C and SPI communication modes are master at singlechip terminal, can write instruction packet to IP71X at any time, or read the instruction packet saved in the IP71X.

The first parameter and the second parameter compose 16-bit value, the instruction of each bit is as follows:

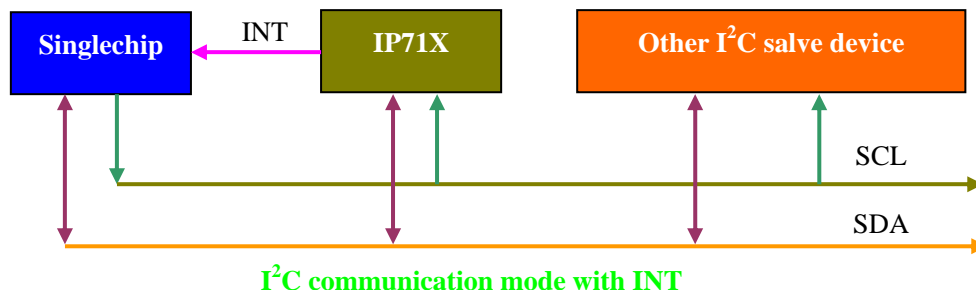
B15 is modification permission bit. Only when B15 is 1 can allow modify communication mode from computer terminal or singlechip terminal. When singlechip terminal modifies communication mode, modify the corresponding communication mode after singlechip read the response modified successfully. Then singlechip must adopt new communication mode for communication. Don't suggest in general that using singlechip modifies communication mode.

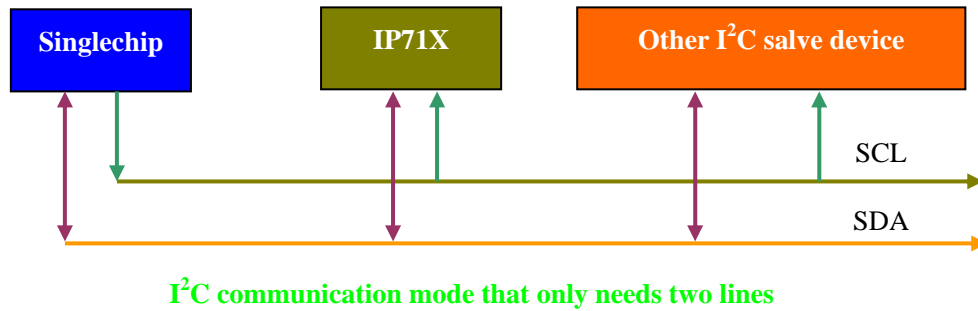
B11 to B8 is the hardware type selection of IP71X and singlechip interface. 0001 is using I²C hardware communication interface; 0010 is SPI communication interface I²C communication; 0100 is using UART hardware communication interface. The rest values reserved.

When hardware communication interface selected is I²C:

Use INT to generate interrupt on the singlechip if B4 is 0. If there is any change at the IP71X, INT will generate change and inform singlechip.

Only use SCL and SDA to connect singlechip if B14 is 1, and singlechip only get the circumstances of instruction packet through querying at certain times.





B7 to B0 is the address of I²C slave, and it is corresponding with the I²C standard address definition.

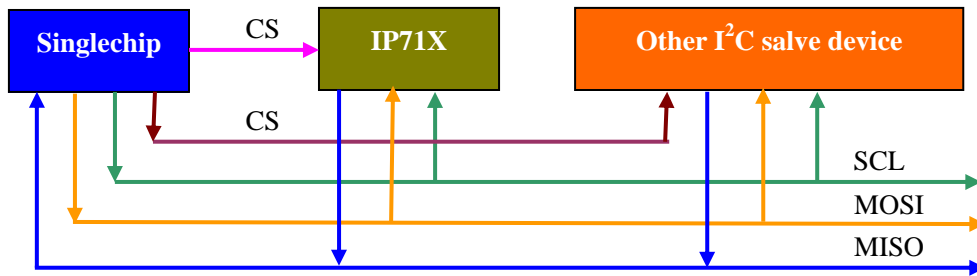
B0 is 0; if singlechip reads IP71X, B0 need to be written as 1 while sending slave address, this accords with I²C criterion.

When hardware communication interface selected is SPI, can use SCK, MOSI, MISO and CS chip selection 4 ports between IP71X and singlechip:

B9 and B8 are SPI mode selection, as shown below:

Mode	The first data driven	Other data driven	Data sampling	SCK level
00	Prior to the fist SCK rise edge	SCK fall edge	SCK rise edge	Low level
01	The first SCK rise edge	SCK rise edge	SCK fall edge	Low level
10	Prior to the fist fall edge	SCK rise edge	SCK fall edge	Low level
11	The fist SCK fall edge	SCK fall edge	SCK rise edge	Low level

SPI bus connecting with user mainframe is supported by IP71X, and using four ordinary IO ports is okay, as shown in the following illustration.



SPI bus consists of CS, SCK, MOSI, and MISO. If user mainframe has not built-in SIP control, can use software mode to generate SPI time sequence. When user mainframe system has many devices of SPI bus can connect in parallel SCK, MOSI, and MISO with the SCK, MOSI, and MISO of IP71X. When CS is high, the MISO of IP71X is high-resistance state. For the standard SPI device, all MISO are high-resistance state when CS chip selection is noneffective.

When use SPI bus, user mainframe regularly uses MOSI to write 0x00, and when the value read is not 0x00 means that IP71X has received the data packet sent by server. When MISO is 0x01 have a package of data packet that has not been read. When MISO is 0x02 have two packages of data packets that have not been read, and the rest may be deduced by analogy. After CS chip selection turns to low, sending the 0xff of one byte later will turn into high CS, and it is a read state register, after finishing, CS will turn to high, and return to the idle state of SPI.



User mainframe writes 0x00 and prepares to read the data of MISO after CS turns to low, the value of MISO read by the first byte is the counter of packet that has not been read, at this time, must keep CS low until all bytes have been read completely.

The second byte is the length of instruction packet, and accords with the length of instruction set; its valid value is 0x02 to 138. User mainframe reads all bytes according to the value of length. The value sent by MOSI should be 0xff while reading.

When user mainframe needs to send data packet, the first byte is length that the valid value is 2 to 138, and accords with the length of instruction set. User mainframe must write instruction packet data that is same as length.

When the first byte turning to low in the CS is 0xff means that user mainframe does not read the state of instruction packet, if it is 0x00 means that user mainframe reads instruction packet, and it is 0x02 to 0x8a means that user mainframe sends instruction packet. When only have 0xff, CS is an 8-bit time and low, after finishing it will turn into high CS. CS will become high after finishing read and writing while reading and sending instruction packet.

The highest clock of SPI clock is 6MHz. So the maximum value of communication speed adopting SPI is 6Mbps. Comparing with I²C and UART, this speed is the highest. If system has SPI components or user mainframe has system IO can have priority in using SPI bus to get faster communication speed.

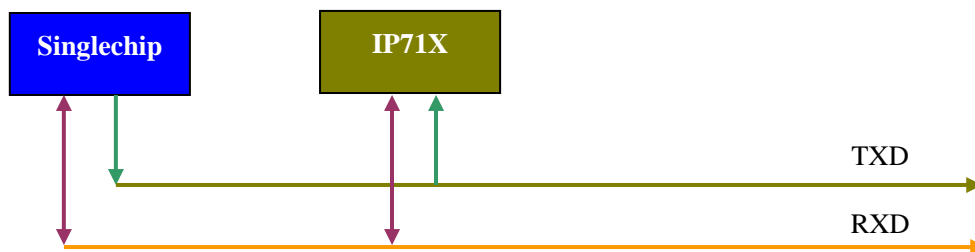
When select UART communication mode, can use RXD and TXD interface between IP71X and singlechip.

B7 to B5 is parity bit selection.

B3 to B0 is UART transfer data bit selection.

B4 is stop bit selection.

B14 is UART format selection. Use IP71X all protocols when it is 0, in this way singlechip can use IP71X all functions, and include instruction length, instruction type and checking code and so on; use transparent transmission when it is 1, and IP71X will leave out instruction length and so on, singlechip will not make any setting and query to IP71X.



Standard UART serial interface communication mod



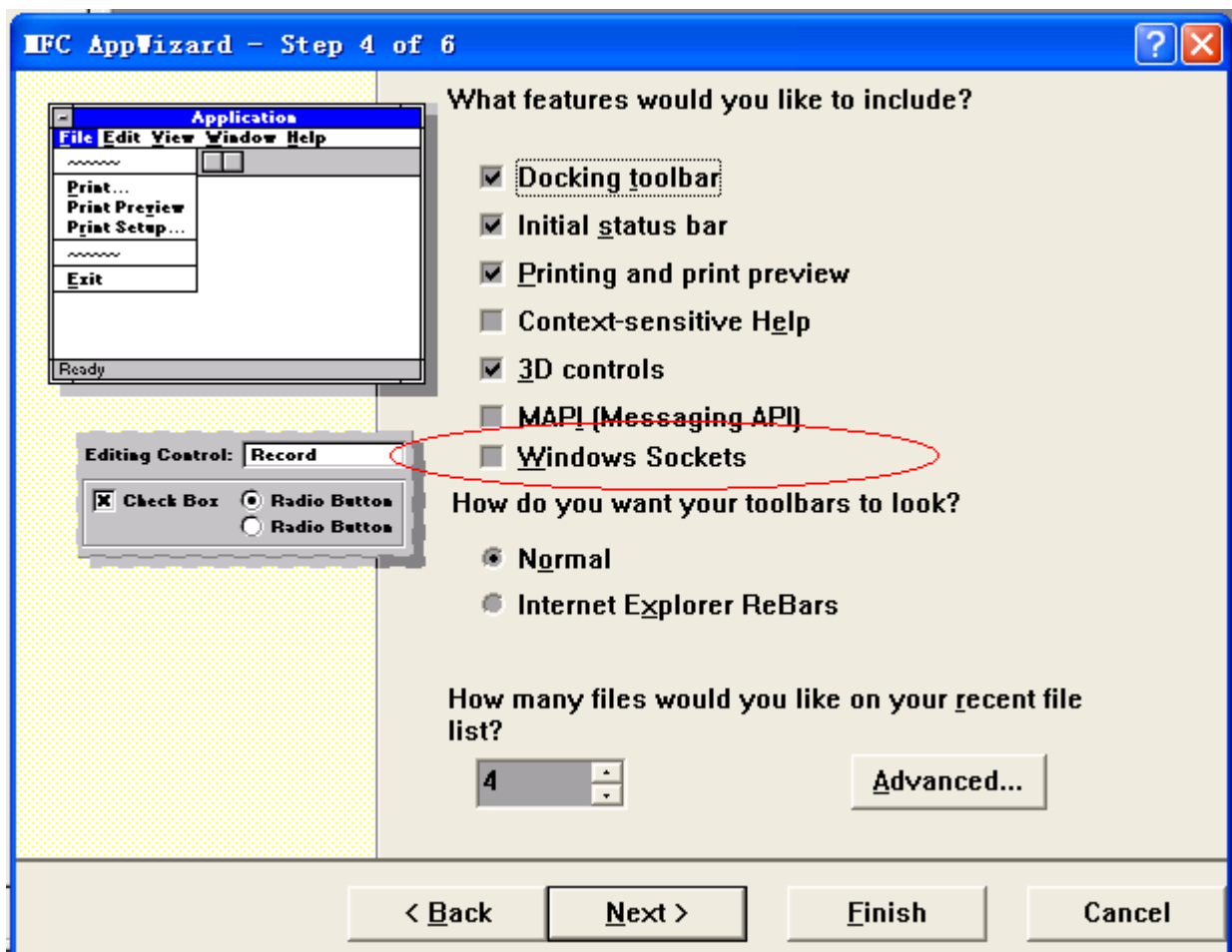
◆ Software compiling attention

Can use the Socket dynamic library of Windows while compiling Windows network program. Windows makes the detailed specification to Socket.

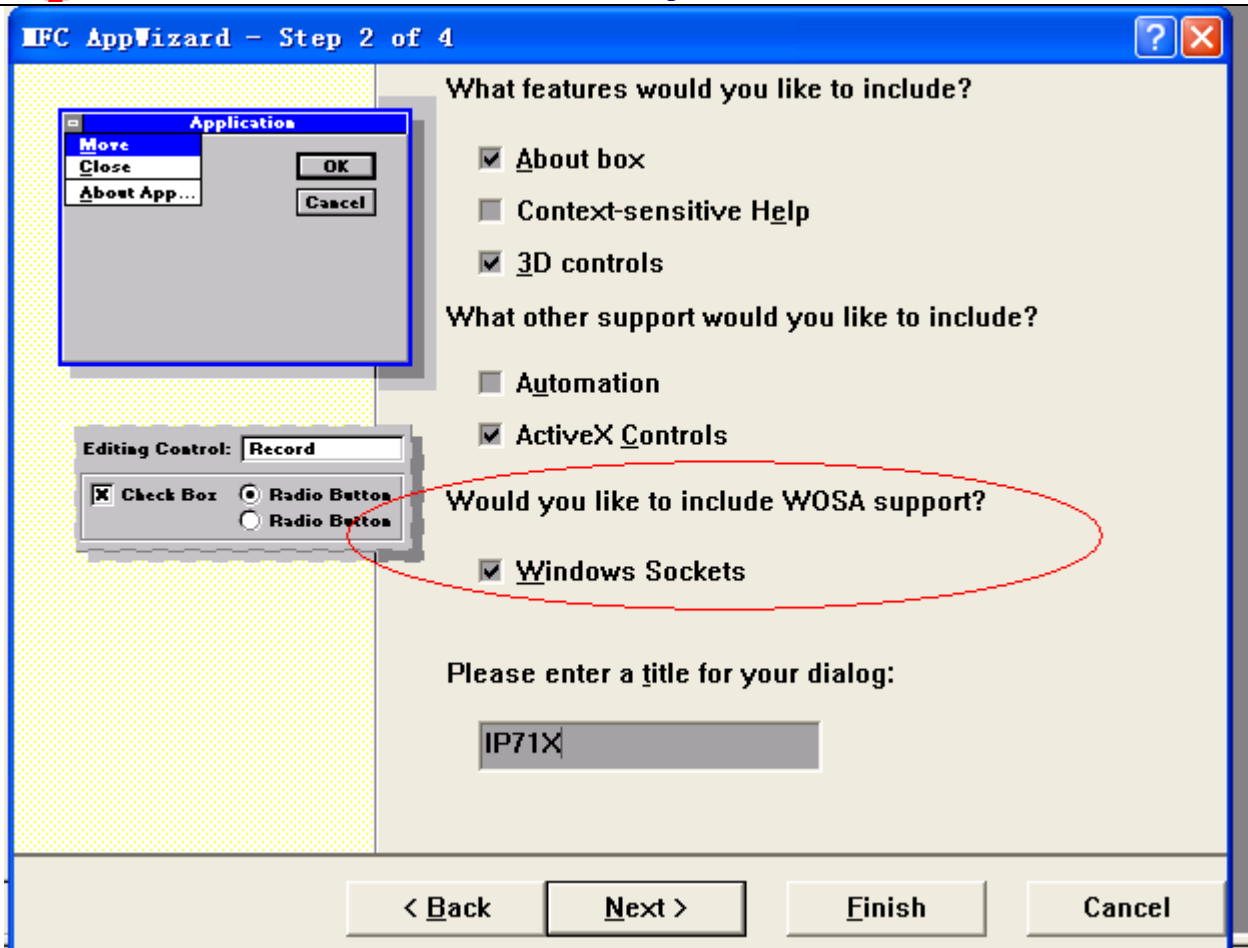
Using Socket is very simple, methods of operation are general: first select to read IP address, and then bind network. Directly using the relevant function to receive and send instruction packet is okay.

The following is the VC programming example that simply shows the compiling method of computer terminal program.

There are two kinds of methods using Socket. The first kind can get Windows Sockets support while newly building item. Please select “Windows Sockets” when Step 4 of 6, as shown below:



If is Dialog Based type, please select “Would you like to include WOSA support?” when Step 2 of 4, as shown below:



Adding in Socket in the old item can add in the following sentences in the InitInstance() of the item .cpp (please note that it is not Dlg.cpp).

```
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";    // The default address of computer terminal mainframe. Users can
    modify it as the default address belongs to you.
```

```
    m_constPort = 65533;    // Port number. It is the port number of IP71X default.
```

```
// Modifying according to the actual port number users can get the IP address of local host
```

```
// Initialization succeeds. Users set up a timer; it can read whether there is any new instruction packet at fixed time, if
there is any new instruction packet, it will make the corresponding processing. Broadcast mode sending can send the
one instruction packet to the all IP of the same network, and it can be used when confirm that the data packet waiting
for sending is all IP.
```

```
    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){ // initialize socket
    AfxMessageBox("initializing socket fails"); // if initialization fails will display indication dialog box.
    OnCancel(); // quit program
}
BOOL bBroadcast = TRUE; // set m_socket attribute, allow to broadcast
m_socket.SetSockOpt(SO_BROADCAST, &bBroadcast, sizeof(BOOL));
return;
}

void CSM311_PS2Dlg::METN_Polling(){ // timer
    CString fromIP; // opposite party IP
    UINT g2Local_1; // opposite party port

    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)){ // test whether there is any data that can be received.
        // buffer area used to receive data    buffer area size    source address    source port
        if (m_socket.ReceiveFrom(gIPB_RX_Buffer, 138, fromIP, g2Local_1)){ // receive data from on socket
            MIPB_RX_Buffer(fromIP, g2Local_1);
        }
    }
    return;
}

void CSM311_PS2Dlg::METN_TX_Command(CString gsLocal_1){
    if (gsLocal_1 == ""){ // broadcast
        m_socket.SendTo(
            LPCTSTR(gIPB_TX_Buffer), // datagram content
            (gIPB_TX_Buffer[0] + 1), // datagram length
            m_constPort, // the port of opposite party, this program use the unified port.
            NULL // NULL means datagram broadcast, the rest value is the
address of opposite party.
        );
    }
    else { // unicast
        m_socket.SendTo(LPCTSTR(gIPB_TX_Buffer), (gIPB_TX_Buffer[0] + 1), m_constPort, gsLocal_1);
    }
    return;
}
}
```



First use broadcast address to send "IP find" instruction packet when computer terminal doesn't know the status of IP71X connection.

It means connection succeeds after the relevant response instruction packet.

Can send the other instruction packet to carry out the corresponding action.