

# *NPort Express* **User' s Guide**

- Ethernet Modem
- Advanced RAW Connection  
(TCP/UDP Support) With Data packing



## Introduction

NPort Express's new Ethernet Modem operation mode is designed to provide true modem emulation, in which NPort Express emulates a regular serial modem, accepting AT commands for dial-in/out service and then transparently passing raw data in/out after the user initiates the Ethernet Modem mode.

After a connection is established, NPort Express acts as a gateway between the serial line and Ethernet LAN, making it possible for raw RS-232/422/485 data to travel across a local LAN. After receiving raw serial data, NPort Express will prefix and postfix a TCP/IP header and trailer, and then send it out to the Ethernet medium. Once the control host receives the TCP/IP data frame, the NOS (Network Operating System) recovers the raw data by decoding the TCP/IP header and trailer. The user can easily capture the raw serial data from the Ethernet medium using Telnet or some customized TCP/IP socket program, providing an ideal long-distance serial data transmission solution between host and serial device.

## Application Architecture


Two typical applications:

(1)




(2)



 **Note:** If the DCD signal for carrier detection is required, you will need to use a cable to connect NPort Express's serial port to your device/PC, as shown here:



## NPort Express Configuration

 *Note: This function is available in firmware version 1.01.81 or above for NPort Express*

1. Enter NPort Express' s configuration screen using either Telnet over a LAN, or PComm Terminal Emulator via NPort Express' s RS-232 Console port.
2. Open the “serverConfig” screen, and then change the “Operating Mode” to [Ethernet Modem].
3. Open the “Serialport” screen and set “TCP port no.” and “Destination IP”.  
NOTES: 1. “TCP port no.” has default value = 4001  
2. Only the device with IP set to “Destination IP” will be allowed access or leave empty for all devices
4. Restart NPort Express' s operating system.

## Using Ethernet Modem

### [Dial-in]

NPort Express listens for a TCP/IP connection request from the remote Ethernet modem or host. NPort Express' s response depends on the ATSO value, as outlined below.

ATSO=0 (default): NPort Express will temporarily accept the TCP connection and then send the “RING” signal out through the serial port. The serial controller must reply with “ATA” within 2.5 seconds to accept the connection request, after which NPort Express enters data mode. If no “ATA” command is received, NPort Express will disconnect after sending three “RING” signals.


ATSO>=1: NPort Express will accept the TCP connection immediately and then send the “CONNECT <baud>” command to the serial port, in which <baud> represents the baud rate of NPort Express' s serial port. After that, NPort Express immediately enters data mode.

### [Dial-out]

NPort Express accepts the AT command “ATD <IP>:<TCP port>” from the serial port and then requests a TCP connection from the remote Ethernet Modem or PC. This is where <IP> is the IP address of the remote Ethernet modem or PC, and <TCP port> is the TCP port number of the remote Ethernet modem or PC. Once the remote box accepts this TCP connection, NPort Express will send out the “CONNECT <baud>” signal via the serial port and then enter data mode.

### [Disconnection request from local side]

When NPort Express is in data mode, the user can drive DTR signal to OFF or send the “+++” from local serial port to NPort Express. NPort Express will enter the command mode and return the “NO CARRIER” via serial port, and then input “ATH” for shot down tcp connection after 1 second.

 *Note: The “+++” cannot be divided, the “+” character can be changed in register S2 and the guard time, which prefixes and suffixes the “+++” in order to protect the raw data, can be changed in register S12.*

#### **[Disconnection request from remote side]**

After the TCP connection is shot down by the remote Ethernet modem or PC, NPort Express will send the “NO CARRIER” signal via serial port and return to command mode.

### **AT Commands Reference Info**

NPort Express supports the following common AT commands used with a typical modem:

No.	AT command	Description	Remarks
1	ATA	Manually answer	
2	ATD <IP>:<Port>	Dial up the IP address & Port No.	
3	ATE	ATE0=Echo OFF ATE1=Echo ON (default)	
4	ATH	ATH0=On-hook (default) ATH1=Off-hook	
5	ATI, ATI0, ATI1, ATI2	Modem version	reply “OK” only
6	ATL	Speaker volume option	reply “OK” only
7	ATM	Speaker control option	reply “OK” only
8	ATO	On line command	
9	ATP, ATT	Set Pulse/Tone Dialing mode	reply “OK” only
10	ATQ0, ATQ1	Quiet command (default=ATQ0)	
11	ATSr=n	Change the contents of S register	See “S registers”
12	ATSr?	Read the contents of S register	See “S registers”
13	ATV	Result code type ATV0 for digit code, ATV1 for text code 0=OK 1=connect (default) 2=ring 3=No carrier 4=error	
14	ATZ	Reset (disconnect, enter command mode and restore the flash settings)	

15	AT&C	Serial port DCD control AT&C0=DCD always on AT&C1=DTE detects connection by DCD on/off (default)	
16	AT&D	Serial port DTR control AT&D0=recognize DTE always ready (default) AT&D1, AT&D2=reply DTE when DTR On	
17	AT&F	Restore manufacturer' s settings	
18	AT&G	Select guard time	reply "OK" only
19	AT&R	Serial port RTS option command	reply "OK" only
20	AT&S	Serial port DSR control	reply "OK" only
21	AT&V	View settings	
22	AT&W	Write current settings to flash for next boot up	

## S Registers

No.	S Register	Description & default value	Remarks
1	S0	Ring to auto-answer (default=0)	
2	S1	Ring counter (always=0)	no action applied
3	S2	Escape code character (default=43 ASCII "+")	
4	S3	Return character (default=13 ASCII)	
5	S4	Line feed character (default=10 ASCII)	
6	S5	Backspace character (default= 8 ASCII)	
7	S6	Wait time for dial tone (always=2, unit=sec)	no action applied
8	S7	Wait time for carrier (default=3, unit=sec)	
9	S8	Pause time for dial delay (always=2, unit=sec)	no action applied
10	S9	Carrier detect response time (always=6, unit 1/10 sec)	no action applied
11	S10	Delay for hang up after carrier (always=14, unit 1/10 sec)	no action applied
12	S11	DTMF duration and spacing (always=100 ms)	no action applied
13	S12	Escape code guard time (default=50, unit 1/50 sec) to control the idle time for "+++"	

# **Advanced RAW connection (TCP/UDP support) with Data Packing**

---

## **Introduction**


NPort Express makes an ideal Ethernet gateway for serial RS-232/422/485 data, and support raw data transmission, making it possible for serial data to travel over a LAN. When NPort server receives raw serial data, a TCP/IP or UDP header and trailer are added, and then the resulting packet is sent out over the Ethernet medium. Once the control host receives the TCP or UDP frame, the NOS (Network Operating System) recovers the raw data by decoding the TCP or UDP header and trailer. This allows the user to easily capture the raw serial data via the Ethernet, using either Telnet or a customized TCP or UDP socket program, and provides an ideal solution for long-distance serial data transmission between host and serial device.

## **The Client-Server Principle**

In general, the “Client” is a program (e.g., Internet Explorer) which actively requests a specific service, with the service often located on a remote host. A “Server” on the other hand is a program that passively listens and responds to requests from Clients. It is often the case that Clients reside on an individual’s PC and Servers reside on larger and faster computers used specifically to run Server programs (in fact, this type of computer is itself often referred to as a Server).

## **The RAW connection TCP Server/Client**

The “raw connection” allows NPort Server to act as a passive server (TCP server) that listens for TCP connection requests from client hosts, or an active client (TCP client) that requests the TCP connection to the specific host for data transmission. The host reads or writes the serial data bi-directionally after the TCP connection has been established.

 **Note:** *The operating mode “RAW connection (TCP server)” replaces the older “RAW connection mode” with the powerful data-packing which helps data transmission more efficient.*

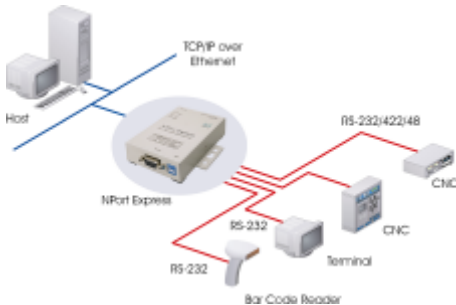
## **The RAW connection UDP Server/Client**

The “raw connection UDP server/client” supports both server and client ways for data transmission. NPort Express sequentially sends the serial data to the hosts configured in “Destination IP addresses”, and transfers the data from the hosts configured in “Source IP addresses” to serial port. NPort Express totally provides 4 IP ranges of both destination hosts and source hosts.

## The powerful “Data Packing” makes data transmission more efficient

NPort Express adds the “delimiter” and “force transmit time out” for customized data format. Once the user startup with one or two of these settings, NPort Express will automatically send the data from serial port to Ethernet after receiving the delimiter character(s) or receiving serial data time out. This function could meet your specified application requirements like card reader, copy machine, scanner. etc.

## The application Architecture



## Configuring NPort Express

### (1) NPort Express (applies to firmware v. 1.01.82 or above for DE-311/DE-311M)

- 1-1 Telnet NPort Express' IP address (default = 192.168.127.254) to access the Telnet console.
- 1-2 Choose the VT-100 console terminal type by pressing 1.
- 1-3 Move the cursor to “Server” and then press Enter to configure IP address, netmask, DHCP, and gateway.
- 1-4 Press ESC, move the cursor to “OP\_mode”, press Enter, and then select RAW connection TCP server, TCP client or UDP server/client mode.
- 1-5 Press Enter to set the serial port(s) to RAW connection.
- 1-6 Move the cursor to “Pure raw data mode” and then press enter for more settings, including TCP port No., etc.
- 1-7 Press ESC, move the cursor to “Save” to save the configuration, and then move to Restart to reboot the system.

### **Notes 1: for TCP server / client**

*Destination IP: Keep blank to allow access for all hosts, or enter an IP address to allow access to only one host. This IP MUST be assigned under TCP client mode.*

*TCP connect on: (for TCP client only) choose the "startup" for automatic TCP connecting to destination host after system startup and ready; or "Any character" for building up TCP connection after receiving serial character.*

*Inactivity time: (0~65535 ms) the serial port will be reset if there is no transmission during this time.*

*TCP alive check time: (0-99 min.) the TCP connection will be reset if there is no activity for a length of time equal to tcp\_time\_out.*

*Data Packing: Delimiter 1 (Hex): keep empty for none trail delimiter needed or input a character in Hex*

*Delimiter 2 (Hex): Keep empty or input the second trail character in Hex*

*Force transmit: (0~65535 ms) system sends the serial data to Ethernet or keep empty for none force transmit needed.*

### **Notes 2: for UDP server / client**

*Serial to Ethernet*

*Dest IP addr: 4 ranges of IP address which NPort Express transmits the serial data to with the respective port no.*

*Ethernet to Serial*

*Src IP addr: 4 ranges of source IP address which NPort Express accepts the data from and transmits to the serial port; Keep blank for accepting all hosts.*

*Inactivity time: (0~65535 ms) the serial port will be reset if there is no transmission during this time.*

*TCP alive check time: (0-99 min.) the TCP connection will be reset if there is no activity for a length of time equal to tcp\_time\_out.*

*Data Packing:*

*Delimiter 1 (Hex): keep empty for none trail delimiter needed or input a character in Hex*

## Delimiter 2 (Hex): Keep empty or input the second trail character in Hex

Force transmit: (0~65535 ms) system sends the serial data to Ethernet or keep empty for none force transmit needed.

## Introduction of socket interface

This interface provides easy-to-use commands for building up customized applications. When using the Socket Interface, connections are established with an IP address and port number. The IP address identifies a remote host and the port number identifies an application process (e.g., 80 for www browser service). The combination of <IP address : port number> allows the user to access the desired application process running on the remote host.

NPort Server will launch the bi-directional serial transmission service and then listen for requests from the Client via NPort server's IP and TCP port 4001 (the manufacturer's default value) after the user starts up the raw connection service. (The request is commonly made by application software built in or installed on the user interface PC or terminal—refer to the *Programming Example* given below for more details on how to use this operation mode.)

## Programming Example 1

The following example was developed under VC++ 4.2 for Windows 9x/NT/2000

```
////////////////////////////////////  
//NPort RAW mode Tx/Rx example program //  
// // //  
//Date: 08-30-2001 //  
// // //  
//Version: 1.0 //  
// // //  
//Program description: //  
// A dumb terminal emulation example program //  
// via NPort RAW mode //  
// 1. hit <ESC> to stop the program //  
// 2. program sends data from the keyboard to NPort //  
// 3. program prints to the screen any data //  
// read from NPort //  
// 4. Syntax: as_raw NPort_IP TCP_PORT //  
// 5. Program developed under VC++ 4.2 //  
// 6. May be used on Windows 9X/NT/2000 OS //  
////////////////////////////////////
```

```

#include <winsock2.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <string.h>
#include <conio.h>

#define IP_ERROR      0xFFFFFFFF // Invalid ip address
#define INTERVAL     100        // 10 msec
#define RETRY         50         // connect retry count

SOCKET              Fd;         // Socket descriptor used on data Tx/Rx

unsigned long        dot2ip(char *dot);
SOCKET               sioopen(unsigned long ipaddr,int p);

int main(int argc, char *argv[])
{
    int      port,i;
    unsigned long ip;
    WSADATA  wsaData;
    char     ch,len;
    char     buf[80];
    SOCKET   fd;

    if ( argc < 3 ) {
        printf("Syntax: %s NPortIP TCP_Port\n",argv[0]);
        return -1;
    }

    ip = dot2ip(argv[1]);
    if (ip == IP_ERROR) {
        printf("Invalid IP address %s!\n",argv[1]);
        return -2;
    }
    port = atoi(argv[2]);

    //
    // On windows we need to call WSASStartup before calling any SOCKET function
    //
    // If your project(VC++,VB,DELPHI) has include TCP/IP MODULE on it,
    // you do need not to call this function, because it is called automatically
    // when you select TCP/IP module.
    //
    if (WSASStartup(0x202,&wsaData) == SOCKET_ERROR) {
        fprintf(stderr,"WSASStartup failed with error
%d\n",WSAGetLastError());
        WSACleanup();
        return -5;
    }
    //

```

```

// connect to remote
//
printf("connecting to --> %s@s....",argv[2],argv[1]);
fd = sioopen(ip,port);
if (fd != INVALID_SOCKET) {
printf("ok\n");
printf("<ESC> = stop program.\n");
printf("Any key = send to remote.\n");
printf("Dumb terminal begin ... \n");
printf("\n");
sprintf(buf,"Welcome to NPort RAW mode example prog.\r\n");
send(fd,buf,strlen(buf),0); // Send welcome string to remote
for (;;) {
if (kbhit()) { // keyboard is hit
ch = getch();
if (ch == 27) { // user hit <ESC> --> exit
printf("\n");
break;
}
send(fd,&ch,1,0); // Send data to NPort
if (ch == '\n') {
send(fd,"\r",1,0); // send LF as CR-LF
}
if (ch == '\r') {
send(fd,"\n",1,0); // send CR as CR-LF
}
}
len=recv(fd, buf, sizeof(buf), 0);

if (len <= 0) // No data read
Sleep(10); // Prevent from wasting too much of CPU time
else {
for (i=0;i<len;i++)
printf("%c",buf[i]);
}
}
closesocket(fd);// Close TCP connection
} else {
printf("fail!\n");
}
}
//
// On windows we need to call WSACleanup to free SOCKET resource
// before exiting the program
//
WSACleanup();
printf("hit any key to stop program...\n");
getch();
printf("program exit.\n");
return 0;
}

```

```

//
//Convert dot notation to IP address
// ie: From "192.168.2.1" to 0x0102A8C0
//
unsigned long dot2ip(char *dot)
{
    unsigned long ip;
    unsigned char *c;
    int i, d;

    c = (unsigned char *)&ip;
    for (i = 4; i-- > 0; ) {
        d = *dot++ - '0';
        if (d < 0 || d > 9)
            return IP_ERROR;
        while (*dot >= '0' && *dot <= '9') {
            d = d * 10 + *dot++ - '0';
            if (d > 255)
                return IP_ERROR;
        }
        *c++ = d;
        if (*dot++ != '.')
            break;
    }
    if (*--dot || i)
        return IP_ERROR;
    return ip;
}
//
//Connect to remote TCP port
//
SOCKET sioopen(unsigned long ipaddr,int port)
{
    struct sockaddr_in des;
    int i,j,len;
    SOCKET fd;
    BOOL b = TRUE;
    ULONG mode = 1; /* set to non_delay mode */
    unsigned short p;

    p = htons((unsigned short)port);
//
// open socket
//
    fd = socket(AF_INET, SOCK_STREAM, 0);
    if ( fd == INVALID_SOCKET ) {
        return(fd);
    }
//

```

```

// Set SOCKET to No Delay mode
//
    if (ioctlsocket(fd,FIONBIO,&mode)) {
        closesocket(fd);
        return(INVALID_SOCKET);
    }
//
// Set remote IP address and port no
//
    des.sin_family = AF_INET;
    des.sin_addr.s_addr = ipaddr;
    des.sin_port = p;
    len = sizeof(struct sockaddr_in);
//
// connect to remote
//
    i = 0;
    for (;;) {
        j = connect(fd,(struct sockaddr *)&des, len);
        if (j == 0) // connected
            break;
        if (WSAGetLastError() == WSAEISCONN) { // already connected
            j = 0;
            break;
        }
        if (i++ >= RETRY) // Connected failed too many times --> give
up
            break;

        Sleep(INTERVAL); // Sleep for a while before trying it again.
        // Prevent from wasting too much of CPU time.
    }
    if( j != 0 ) { // Can't connect to remote
        closesocket(fd);
        return(INVALID_SOCKET);
    }
    return(fd);
}

```

## Programming Example 2

The following program was developed under VB6.0 with serial settings 38400, n, 8, 1

---

```
Private Sub cmdConnect_Click()  
    If txtIP.Text = "" Or txtPort.Text = "" Then Exit Sub  
    Winsock1.Connect txtIP.Text, txtPort.Text  
    txtStatus.Text = txtStatus.Text & "Connecting..." & vbCrLf  
    Timer1.Enabled = True  
End Sub  
  
Private Sub cmdDisconnect_Click()  
    Winsock1.Close  
    txtStatus.Text = txtStatus.Text & "Connection Close." & vbCrLf  
End Sub  
  
Private Sub cmdClose_Click()  
    If MsgBox("Are you sure to shutdown the Remote Server application?",  
vbQuestion + vbYesNo, "Shutdown") = vbNo Then Exit Sub  
    SendData ("Close:")  
    Winsock1.Close  
End Sub  
  
Private Sub cmdSendKey_Click()  
    Dim strMsg As String  
    strMsg = InputBox("Please enter any letters to send back to the Server.",  
"Send Key", "")
```

```

    If strMsg <> "" Then
        If Not SendData("Keyboard:" & strMsg) Then
            Winsock1.Close
        End If
    End If
End Sub

Private Sub Timer1_Timer()
    MsgBox "Client could not find server.", vbCritical
    If Winsock1.State <> sckClosed Then
        Winsock1.Close
    End If
    Timer1.Enabled = False
    txtStatus.Text = txtStatus.Text & "Connection Fail." & vbCrLf
End Sub

Private Sub Winsock1_Connect()
    Timer1.Enabled = False
    txtStatus.Text = txtStatus.Text & "Connection Established." & vbCrLf
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim strData As String
    Winsock1.GetData strData
    txtStatus.Text = txtStatus.Text & "Get data: " & strData & vbCrLf & vbCrLf
End Sub

```

```

Private Function SendData(sData As String) As Boolean

    On Error GoTo ErrorHandler

    Dim lngTime As Long

    blnReply = False

    Winsock1.SendData sData

    Do Until (Winsock1.State = 0) Or (lngTime < 10000)

        DoEvents

        lngTime = lngTime + 1

        If lngTime > 10000 Then Exit Do

    Loop

    SendData = True

    Exit Function

```

ErrorHandler:

```

    SendData = False

    MsgBox Err.Description, vbCritical

    Exit Function

```

End Function

---

```

Private Sub cmdConnect_Click()

```

```

    If txtIP.Text = "" Or txtPort.Text = "" Then Exit Sub

    Winsock1.Connect txtIP.Text, txtPort.Text

    txtStatus.Text = txtStatus.Text & "Connecting..." & vbCrLf

    Timer1.Enabled = True

```

```

End Sub

```

```

Private Sub cmdDisconnect_Click()
    Winsock1.Close
    txtStatus.Text = txtStatus.Text & "Connection Close." & vbCrLf
End Sub

Private Sub cmdClose_Click()
    If MsgBox("Are you sure to shutdown the Remote Server application?",
vbQuestion + vbYesNo, "Shutdown") = vbNo Then Exit Sub
    SendData ("Close:")
    Winsock1.Close
End Sub

Private Sub cmdSendKey_Click()
    Dim strMsg As String
    strMsg = InputBox("Please enter any letters to send back to the Server.",
"Send Key", "")
    If strMsg <> "" Then
        If Not SendData("Keyboard:" & strMsg) Then
            Winsock1.Close
        End If
    End If
End Sub

Private Sub Timer1_Timer()
    MsgBox "Client could not find server.", vbCritical

```

```

    If Winsock1.State <> sckClosed Then
        Winsock1.Close
    End If

    Timer1.Enabled = False

    txtStatus.Text = txtStatus.Text & "Connection Fail." & vbCrLf
End Sub

Private Sub Winsock1_Connect()
    Timer1.Enabled = False

    txtStatus.Text = txtStatus.Text & "Connection Established." & vbCrLf
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim strData As String

    Winsock1.GetData strData

    txtStatus.Text = txtStatus.Text & "Get data: " & strData & vbCrLf & vbCrLf
End Sub

Private Function SendData(sData As String) As Boolean
    On Error GoTo ErrorHandler

    Dim lngTime As Long

    blnReply = False

    Winsock1.SendData sData

    Do Until (Winsock1.State = 0) Or (lngTime < 10000)
        DoEvents
        lngTime = lngTime + 1
    Loop

```

```

        If lngTime > 10000 Then Exit Do

    Loop

    SendData = True

    Exit Function

ErrorHandler:

    SendData = False

    MsgBox Err.Description, vbCritical

    Exit Function

End Function

```

---

### **Programming Example 3**

The following UDP program was developed under VB6.0 with NPort Express  
192.168.206.106 port 4001

```

////////////////////////////////////
Objects:

Text1 --- for data sent to NPort Express

Text2 --- for data received from NPort Express

Command1 --- End

Command2 --- Send the contents of Text1

////////////////////////////////////

Private Sub Command1_Click()

If Winsock1.State <> sockClosed Then

    Winsock1.Close

End If

```

```
End
End Sub

Private Sub Command2_Click()
Winsock1.SendData Text1.Text
End Sub

Private Sub Form_Load()
With Winsock1
    .Protocol = sckUDPProtocol
    .RemoteHost = "192.168.206.106"
    .RemotePort = 4001
    .LocalPort = 4001
    .Bind
End With
Form1.Caption = "UDP transmission via port 4001"
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
Winsock1.GetData strData
Text2.Text = Text2.Text + strData
End Sub

-----
```