286-Telnet Example Application Note

The application note applies to:

- IP Communications
- Telnet
- NetLinx
- IP Settings

Description

This example shows the basic concepts of IP Communications. Specifically, the example creates an IP Client socket and connect to the NetLinx master using the Telnet protocol. The example issues a "GET IP" command which retrieves all the IP settings from the NetLinx master. Most of these settings are available using the library call GET_IP_ADDRESS function so this is not the easiest (or recommended!) way to retrieve the IP settings.

Note the most Telnet servers (i.e. device that accept telnet command) do require the use of the Telnet escape codes. It is possible that a Telnet server will not establish a connection until the Telnet negotiation sequence has been entered. How do you know if this is the problem you might be seeing? Telnet escape codes begin with the character 0xFF ($FF, 255 or HEX FF). If a device sends you this character and will not respond to any of the command you are sending, it probably requires the Telnet negotiation sequence. Most devices we have encountered so far do not require this: in our experience, the negotiation sequence is not required for the NetLinx master, Polycom codecs, All Windows Telnet servers or most Unix/Linux Telnet servers. See this link for more information on the Telnet protocol.

Notice the IP address used in this example? You might wonder what 127.0.0.1 is. This is the loopback address. IP implementations understand this to be the local machine. In fact, the request you send never makes it out across ethernet. The IP software simply "short-circuits" the request back to itself without sending it across the wire. The name "localhost" and "127.0.0.1" are synonymous

Example Code

```
(***********************************************************)
(* DEVICE NUMBER DEFINITIONS GO BELOW *)
(***********************************************************)
DEFINE_DEVICE
(*** NetLinx ***)
dvTELNET = 0:5:0
(***********************************************************)
(* CONSTANT DEFINITIONS GO BELOW *)
(***********************************************************)
DEFINE_CONSTANT
(* IP *)
IP_TCP = 1
IP_UDP = 2
nTELNET_PORT = 23
(***********************************************************)
```

```
(* TYPE DEFINITIONS GO BELOW *)
(***********************************************************)
DEFINE_TYPE
(***********************************************************)
(* VARIABLE DEFINITIONS GO BELOW *)
(***********************************************************)
DEFINE_VARIABLE
(* INFO *)
CHAR strTELNET_BUFFER[1000]
SLONG slRESULT
(***********************************************************)
(* SUBROUTINE DEFINITIONS GO BELOW *)
(***********************************************************)
(*************************************)
(* CALL NAME: GET_TELNET_PARAMETER *)
(* FUNCTION: GET PARAMETER FROM STR *)
(* PARAMS: BUFFER - IN *)
(* RETURN: PARAMETER *)
(*************************************)
DEFINE_FUNCTION CHAR[100] GET_TELNET_PARAMETER (CHAR strDATA[])
STACK_VAR

CHAR chCHAR
CHAR strPARAM[100]
{
(* CLEAN UP AND SHORT CIRCUIT *)
strPARAM = ""
IF (LENGTH_STRING(strDATA) = 0)
RETURN strPARAM;
(* KILL SPACES *)
WHILE (strDATA[1] = $20 && LENGTH_STRING(strDATA))
chCHAR = GET_BUFFER_CHAR(strDATA)
(* GET UP UNTIL CR *)
WHILE (strDATA[1] <> $0D && LENGTH_STRING(strPARAM) < 100)
strPARAM = "strPARAM,GET_BUFFER_CHAR(strDATA)"
(* DONE *)
RETURN strPARAM;
}
(***********************************************************)
(* STARTUP CODE GOES BELOW *)
(***********************************************************)
DEFINE_START
(* BUFFERS *)
CREATE_BUFFER dvTELNET,strTELNET_BUFFER
(***********************************************************)
(* EVENT PROCESSING ROUTINES BELOW *)
(***********************************************************)
DEFINE_EVENT
(**************************)
(*** TELNET MESSAGES *******)
```

```
(***************************)
DATA_EVENT[dvTELNET]
{
ONERROR:
{
SEND_STRING 0,"'TELNET EVENT: ERROR(',ITOA(DATA.NUMBER)"
strTELNET_BUFFER = ""
}
ONLINE:
{
SEND_STRING 0,'TELNET EVENT: ONLINE'
strTELNET_BUFFER = ""

}
OFFLINE:
{
SEND_STRING 0,'TELNET EVENT: OFFLINE'
strTELNET_BUFFER = ""
}
STRING:
{
(* VARS *)
STACK_VAR
CHAR strTEMP[100]
LONG lPOS
(* DEBUG *)
SEND_STRING 0,'TELNET EVENT: INFO'
(* TELNET MESSAGE? *)
IF (FIND_STRING(strTELNET_BUFFER,'Welcome',1))
{
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'Welcome',1)
strTELNET_BUFFER = ""
SEND_STRING dvTELNET,"'GET IP',13,10"
}
(* GET IP? *)
IF (FIND_STRING(strTELNET_BUFFER,'MAC Address',1) &&
FIND_STRING(strTELNET_BUFFER,'>',1))
{
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'IP Settings',1)
strTEMP = REMOVE_STRING(strTELNET_BUFFER,"$0D,$0A",1)
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'HostName',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'HOST NAME=',strTEMP"
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'Type',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'IP TYPE=',strTEMP"
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'IP Address',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'IP ADDRESS=',strTEMP"
```

```
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'Subnet Mask',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'SUBNET MASK=',strTEMP"

strTEMP = REMOVE_STRING(strTELNET_BUFFER,'Gateway IP',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'GATEWAY IP ADDRESS=',strTEMP"
strTEMP = REMOVE_STRING(strTELNET_BUFFER,'MAC Address',1)
strTEMP = GET_TELNET_PARAMETER(strTELNET_BUFFER)
SEND_STRING 0,"'MAC ADDRESS=',strTEMP"
strTELNET_BUFFER = ""
IP_CLIENT_CLOSE (dvTELNET.PORT)
}
}
}
(***********************************************************)
(* THE ACTUAL PROGRAM GOES BELOW *)
(***********************************************************)
DEFINE_PROGRAM
WAIT 200
{
slRESULT = IP_CLIENT_OPEN (dvTELNET.PORT,'127.0.0.1',nTELNET_PORT,IP_TCP)
IF (slRESULT < 0)
SEND_STRING 0,"'IP_CLIENT_OPEN ERROR: ',ITOA(slRESULT)"
}
(***********************************************************)
(* END OF PROGRAM *)
(* DO NOT PUT ANY CODE BELOW THIS COMMENT *)
(***********************************************************)
```