

External Control of an NK Series Routing System using the NK-SCP/A

Introduction

An NK Series routing system has several different control options including hardware control panels, software control from a computer using the Phoenix NK Switchboard, the NK-GPI General Purpose Interface and serial ASCII control using the NK-SCP/A Serial Control Interface.

The ability to control an NK Series routing system from other manufacturer's equipment allows seamless integration into many different operating scenarios. For example, an automation system may need to control a router to automatically switch program to air from a number of different sources at pre-scheduled times, or a sophisticated AV presentation system in a conference center may require control of signal routing, lights, projectors etc all from a master control panel via a system controller.

The Codan Broadcast NK-SCP/A Serial Control Interface allows an NK Series Routing System to be controlled by any device capable of transmitting and receiving serial ASCII encoded data. Codan Broadcast have developed a simple ASCII protocol which is openly available for use with third party products and to software developers who wish to take advantage of the ability to externally control an NK Series routing system.



Figure 1 – NK-SCP/A Serial Control Interface

The NK-SCP/A – How It Works

The NK-SCP/A translates communications between the NK Series protocol transported by the native T-Bus and the ASCII protocol used by devices connected to the RS232 port on the NK-SCP/A. In this document the term “ASCII device” is used to describe any device connected to the NK-SCP/A for the purpose of controlling or receiving status from an NK Series routing system.

Connections to the NK-SCP/A are simple; RJ45 loop through ports connect to the routing system T-Bus via Cat-5 cable, which also provides phantom power to the unit, and a DB-9 RS232 port connects to the ASCII device via a standard RS232 cable. The RS232 pin assignments for the NK-SCP/A 9-pin connector are shown below:

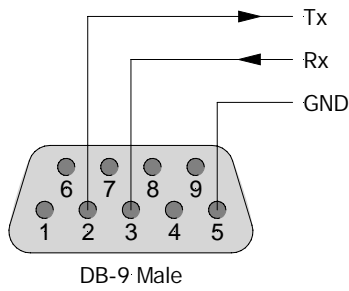


Figure 2 – NK-SCP/A Serial Port Pin Assignments

The diagram below shows how an ASCII device is connected via the NK-SCP/A to an NK Series routing system:

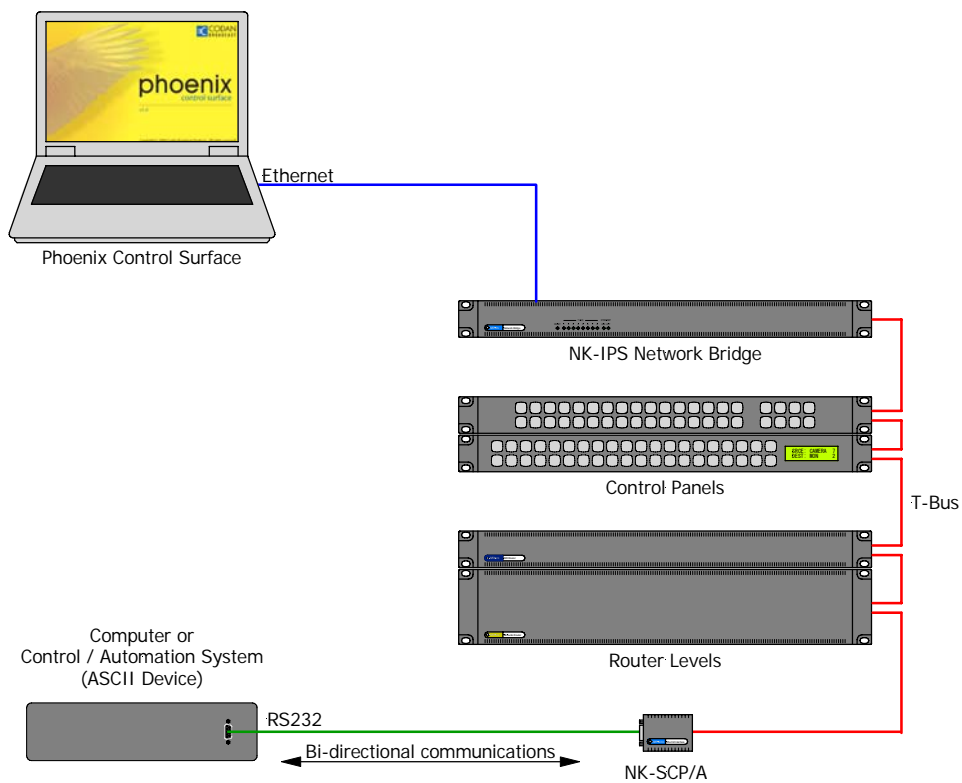


Figure 3 – NK Series Routing System with External RS232 Control

There are three commands that can be sent from an ASCII device to the NK-SCP/A, two of which will switch routers, and one which will request router status, summarised in the table below:

Command	Description
<X>	Switch an input to an output using a breakaway
<Z>	Switch multiple inputs (one per level) to an output
<R>	Read the current status of an output

Table 1 – NK-SCP/A ASCII Protocol Commands

After receiving a command string, the NK-SCP/A sends a response back to the ASCII device acknowledging that a valid data string has been received, and in the case of a valid request for router status, it returns the status for each of the 8 levels. Details of the ASCII protocol for the NK-SCP/A are included in the Appendix of this document.

Note that the NK-SCP/A will only send a response to a command first initiated by the ASCII device. The sequence of command and response strings is shown below:

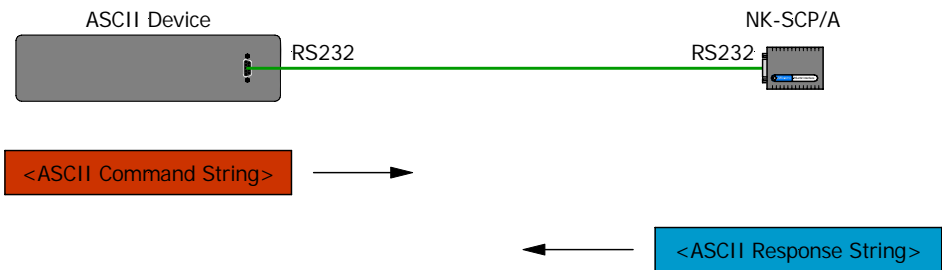


Figure 4 – ASCII Communication between ASCII Device and NK-SCP/A

ASCII strings are made up of 8-bit ASCII characters, each with 1 stop bit.

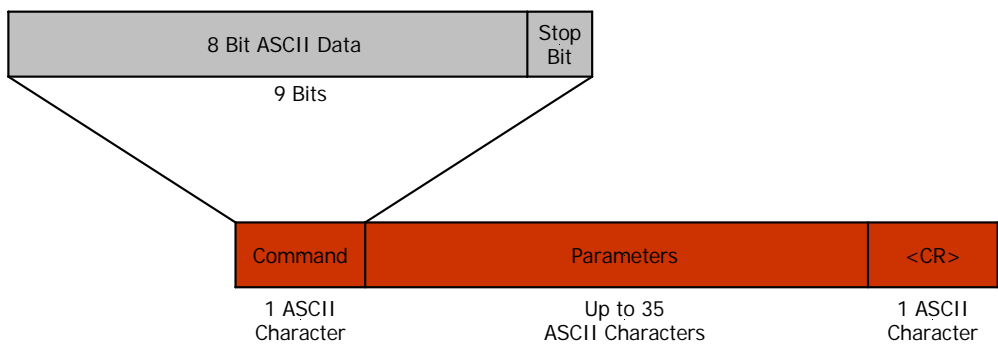


Figure 5 – ASCII Command String Format sent by ASCII Device

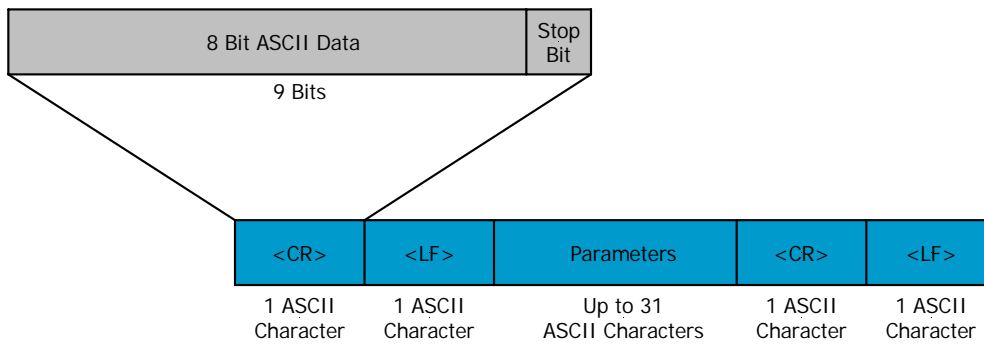


Figure 6 – ASCII Response String Format sent by NK-SCP/A

To demonstrate how an ASCII device can control an NK Series routing system via the NK-SCP/A we will use a computer running a simple ASCII terminal program (Windows XP HyperTerminal) to send and receive ASCII characters via the computer's serial port.

In most applications which use a computer to serially control the routing system via the NK-SCP/A custom software will be developed or application specific third party software will be used which implements the NK Series ASCII protocol through a third party programming interface.

For this demonstration, assume we are controlling the NK Series system shown in Figure 7 below comprising a 16 x 16 SDI router (assigned to Level 1), a 16 x 16 AES router (assigned to Level 2) and an RCP-NK1 control panel (providing local control).

The same computer running HyperTerminal can be used to configure, control and monitor the NK Series devices in the system using Phoenix via an NK-IPS as shown below. Router status can be monitored using the NK Switchboard in Phoenix.

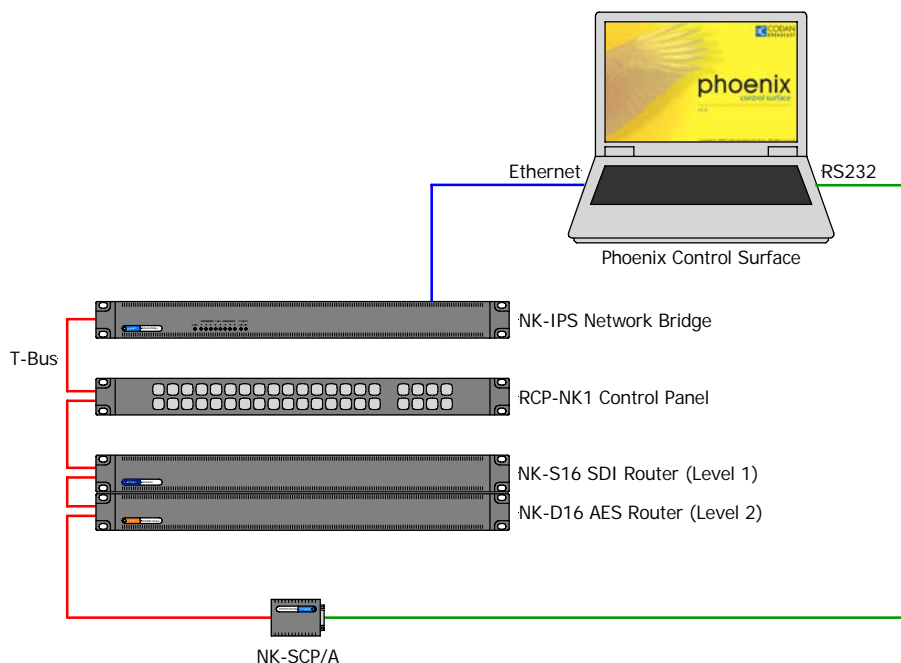


Figure 7 – NK Series Routing System with Control via an NK-SCP/A

Configuring the NK-SCP/A

The NK-SCP/A, like all other NK Series devices, is configured via an NK-IPS Network Bridge from a web browser or Phoenix software. The default NK-SCP/A configuration in Phoenix is shown below:

NKseries
SCP/A
Serial Num: 21536 Version: 2.12

Device Details
Name: NK - SCP / A Details:
Group: Address: 210

Configuration

ASCII Port Baud Rate: 9600

Breakaway Level Map:

Breakaway num	level mask (L...L +8)
1	1 2 3 4 5 6 7 8
2	1 2 3 4 5 6 7 8
3	1 2 3 4 5 6 7 8
4	1 2 3 4 5 6 7 8
5	1 2 3 4 5 6 7 8
6	1 2 3 4 5 6 7 8
7	1 2 3 4 5 6 7 8
8	1 2 3 4 5 6 7 8

First Input: 1
First Output: 1
Level Position: 1->8
Startup Delay (s): 5

Figure 8 – Default NK-SCP/A Configuration

The Breakaway Level Map in the NK-SCP/A configuration uses a level mask to define which levels will be switched when a particular breakaway is included in an ASCII switch command string sent from the ASCII device. Up to 8 breakaways can be defined in the NK-SCP/A Breakaway Level Map, with up to 8 levels included in each breakaway. The Level Position parameter allows the 8 levels to be mapped within the 32 levels supported by the NK Series control system.

Assigning Levels to Breakaways in the Breakaway Level Map

The Breakaway Level Map needs to be configured to include only levels which actually exist on the routing system, or the NK-SCP/A will return an ERROR. For the example system shown in Figure 7, the Breakaway Level Map below is configured with breakaways to make a tied switch (both Levels 1 and 2) and to switch each level alone:

Breakaway Level Map:

Breakaway num	level mask (L...L+8)
1	1 2 3 4 5 6 7 8
2	1 2 3 4 5 6 7 8
3	1 2 3 4 5 6 7 8
4	1 2 3 4 5 6 7 8
5	1 2 3 4 5 6 7 8
6	1 2 3 4 5 6 7 8
7	1 2 3 4 5 6 7 8
8	1 2 3 4 5 6 7 8

Figure 9 – Breakaway Level Map for NK Series System shown in Figure 2

Using Windows XP HyperTerminal

The Windows XP HyperTerminal program allows character based (ASCII) strings to be sent and received using a simple dialogue window.

Before starting HyperTerminal, first determine the computer’s COM Port used for communication with the NK-SCP/A.

From Windows XP select **Control Panel > System > Device Manager > Ports (COM & LPT)**. In this example a Belkin F5U409 USB / Serial Adaptor was used to emulate the serial COM4 port as shown below:

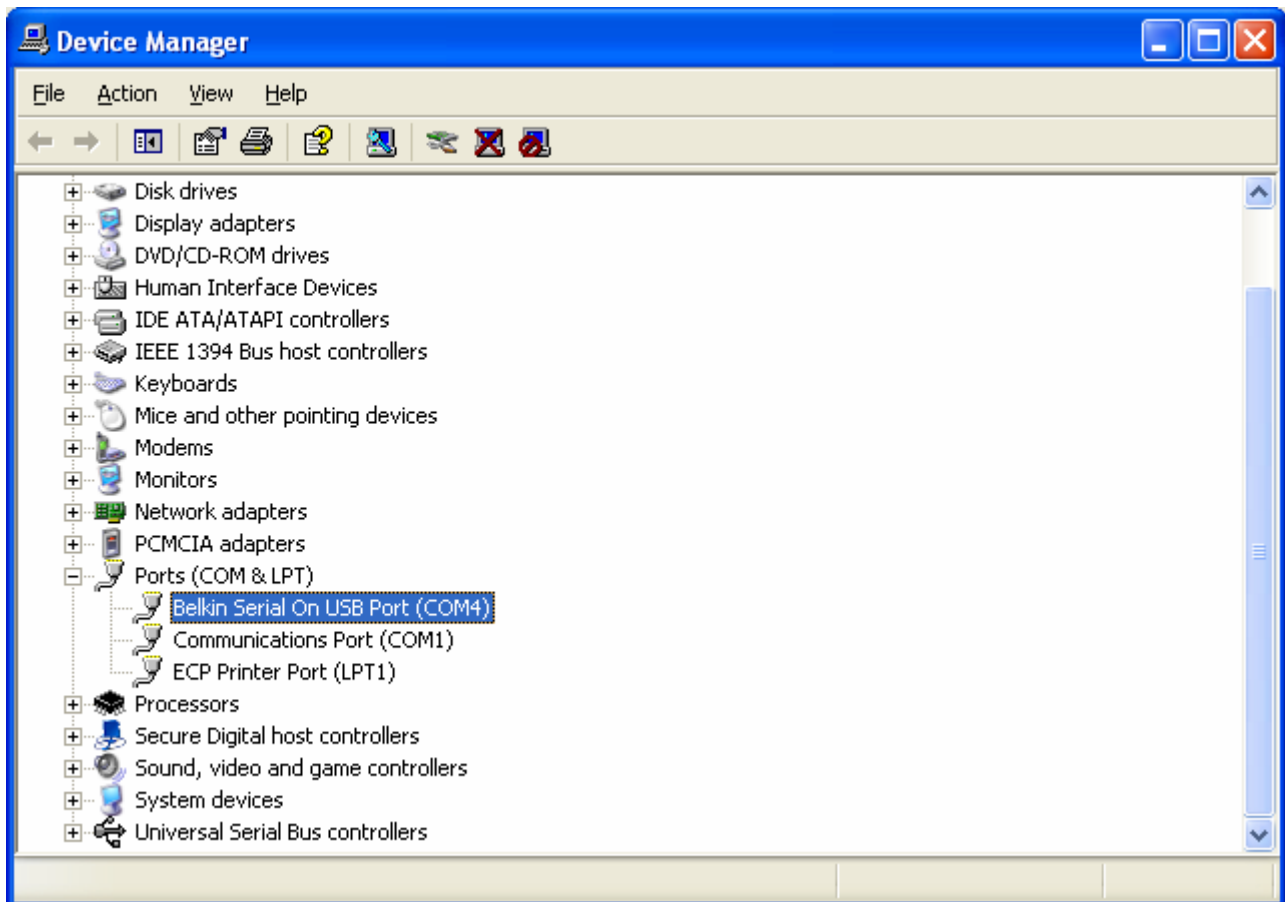


Figure 10 – Windows XP Device Manager COM ports

Open Hyper Terminal from Windows XP by selecting **Programs > Accessories > Communications > Hyper Terminal**. Enter a **Name** for the connection in the **Connection Description** dialogue box as shown below:



Figure 11 – HyperTerminal Connection Description Dialogue Box

Select the COM Port previously determined from Device Manager in the **Connect To** dialogue box as shown below:



Figure 12 – HyperTerminal Connect To Dialogue Box

Port Settings:

The ASCII Port Baud Rate can be set in the NK-SCP/A configuration. Default is 9600. The ASCII Port parameters in the table below are fixed for the NK-SCP/A and must be set correctly in the RS232 port parameters for the equipment which is connected to the NK-SCP/A RS232 port.

Parameter	Setting
Parity	None
Stop Bits	1
Data Bits	8
Characters	ASCII

Figure 13 – NK-SCP/A ASCII Port Fixed Parameters

Set the **Bits per second** parameter in the **COM Properties** dialogue box the same as the ASCII Port Baud Rate setting in the NK-SCP/A configuration and **Flow control** to None as shown below:

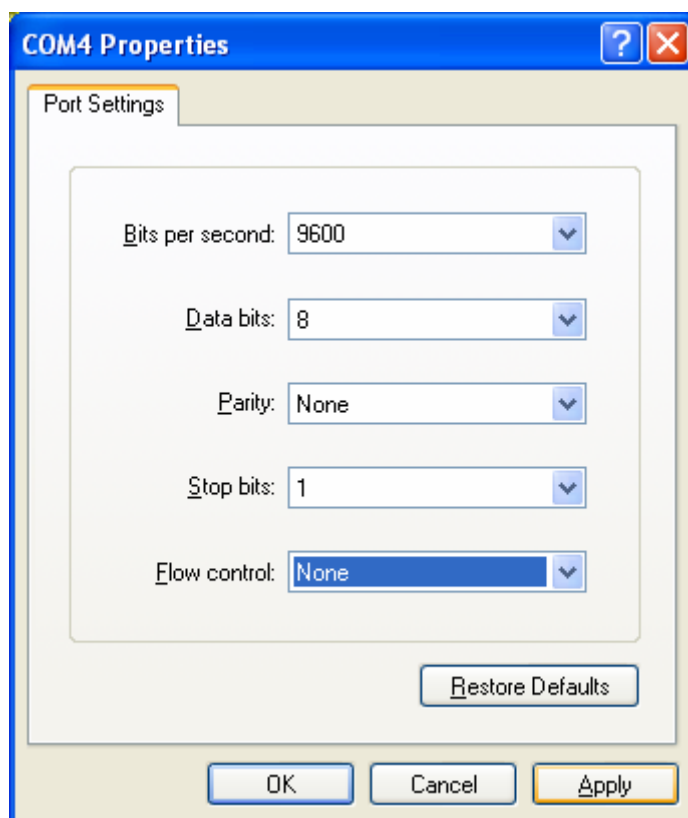


Figure 14 – Hyper Terminal COM Port Properties Page

Apply the settings, then click **OK**. The HyperTerminal window shown below can now be used to type ASCII command strings from the keyboard:

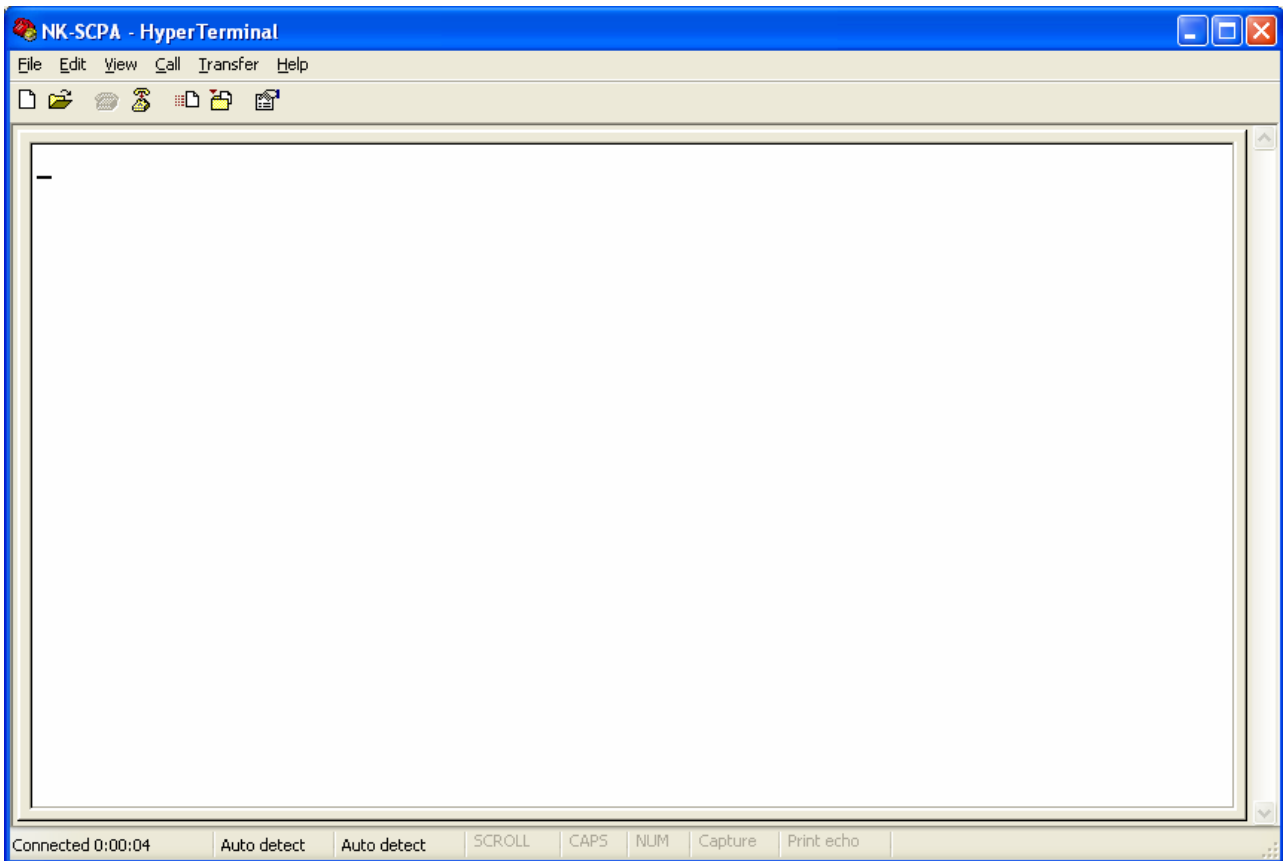


Figure 15 – Hyper Terminal Window

Using the ASCII Protocol

<X> Command - Switching an input to an output using a defined breakaway

The first command we will demonstrate switches a physical input to a physical output on specified levels. The levels switched are specified by the breakaway number whose level assignments are defined in the Breakaway Level Map of the NK-SCP/A, previously configured via Phoenix.

The ASCII command string to switch an input to an output using a defined breakaway has the general form:

<X> <ddd> <,> <sss> <,> <CR>

where

ddd is the **destination** or output number from 000 to 254 (which addresses Outputs 1 to 255)

sss is the **source** or input number from 000 to 254 (which addresses Inputs 1 to 255)

b is the **breakaway** number from 1 to 8

CR is a carriage return (or ENTER on the computer keyboard)

Example 1a - Switch Input 2 to Output 15 on both Levels 1 and 2

The following conceptual diagram shows the switches we wish to make on the example system in Figure 7:

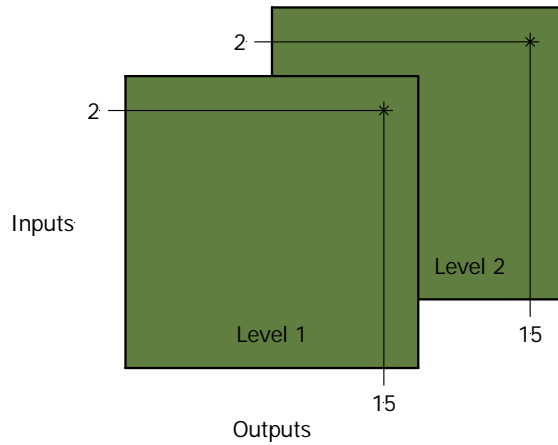


Figure 16 – Router Crosspoint Switch for Example 1a

To send this command from HyperTerminal to the NK-SCP/A, type the following characters in the HyperTerminal window, then press <ENTER>:

X001,014,1

Pressing ENTER appends the carriage return character <CR> and sends the ASCII string from the computer's serial COM port. Note that the <CR> character is a non printable ASCII character and does not display in the HyperTerminal window.

The NK-SCP/A will respond with the following, indicating a valid data string has been received:

OK

The general form of the ASCII response string for this command is:

<CR> <LF> <OK> <SP> <CR> <LF>

Note once again that the ASCII characters <LF> (line feed) and <SP> (space) are non printable ASCII characters and do not display in the HyperTerminal window, even though they are received by the computer's serial COM port. These characters allow HyperTerminal to display the received response in a format which is easily viewed in the HyperTerminal window. Other programs may handle the non-printable ASCII characters differently.

The HyperTerminal session for the above example is shown below:

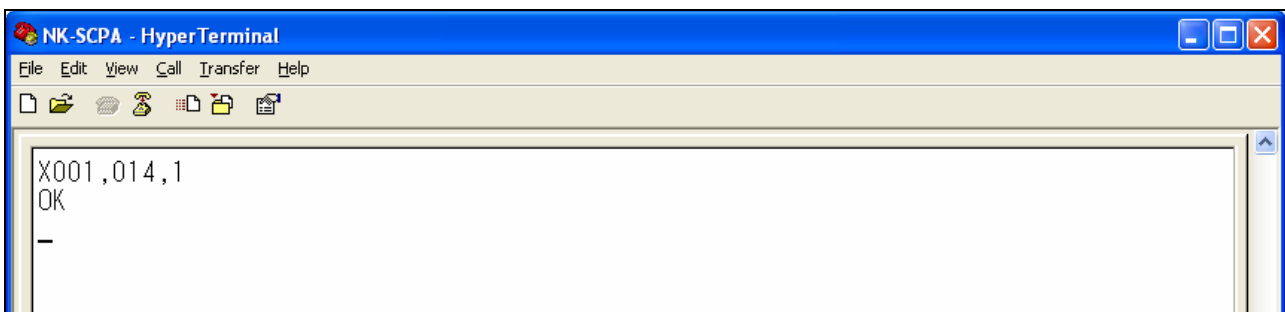


Figure 17 – HyperTerminal Session for Example 1a

Example 1b - Switch Input 2 to Output 24 on Level 1.

The HyperTerminal session for this example is shown below:

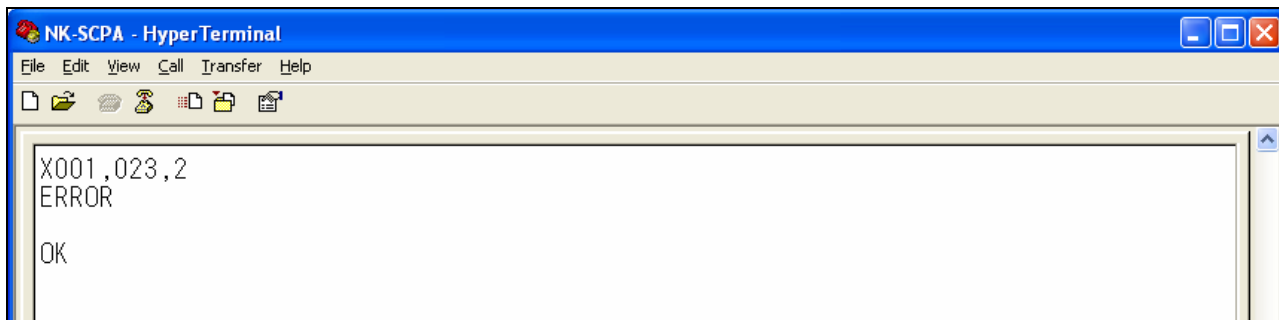


Figure 18 – HyperTerminal Session for Example 1b

The NK-SCP/A has returned an ERROR because Output 24 does not exist on the router assigned to Level 1. The NK-SCP/A also responds with OK indicating that a valid string was received.

Example 1c – Sending an invalid string

The following HyperTerminal sessions are examples of invalid strings received by the NK-SCP/A:

- X001, 014, Breakaway omitted at end of string
ERROR

- X001014, 1 Comma omitted between destination and source
ERROR

- x001, 014, 1 Lower case command character
ERROR

- X1, 14, 1 No leading zeros on destination and source values
ERROR

<Z> Command - Switching multiple inputs (one per level) to an output

Suppose we wanted to switch *different* inputs to the same output on both Levels 1 and 2. The <Z> command specifies the source to be switched to an output for each of 8 levels, with a <----> character sequence sent to not switch a level. The <Z> command string has the general form:

L1 L2 L3 L4 L5 L6 L7 L8

<Z> <ddd> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <CR>

Example 2 - Switch Input 5 to Output 15 on Level 1 and Input 2 to Output 15 on Level 2.

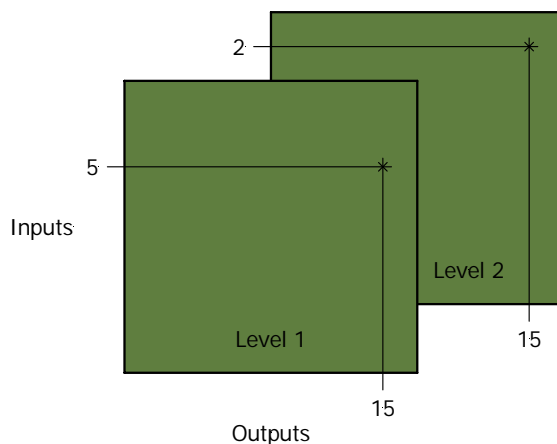


Figure 19 – Router Crosspoint Switch for Example 2

The HyperTerminal session for this example is shown below:

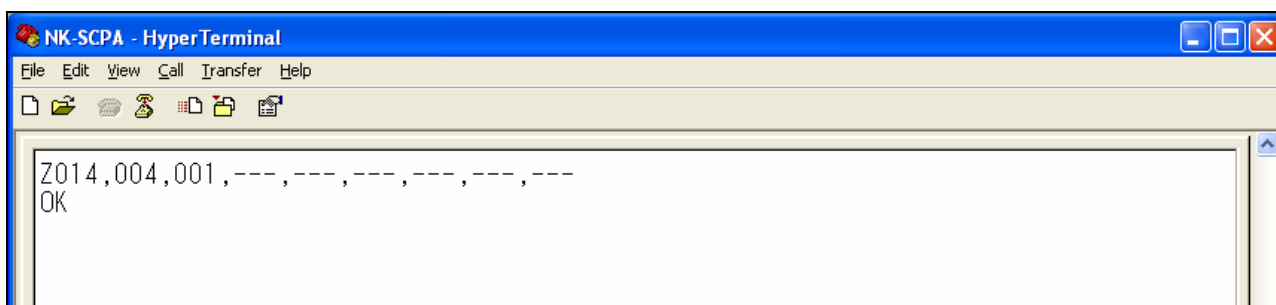


Figure 20 – HyperTerminal Session for Example 2

<R> Command – Read the current status of an output

The <R> command requests the status of an output (i.e. which input is routed to that output) for each of 8 levels. The <R> command string has the general form:

<R> <ddd> <CR>

Example 3 – Request the status of Output 15 for Levels 1 and 2.

After sending the command sequence in Example 2, send the following command sequence followed by ENTER:

R014

This command will return the following response from the NK-SCP/A:

004, 001, ---, ---, ---, ---, ---, ---

OK

The general form of the ASCII response string for this command is:

L1 L2 L3 L4 L5 L6 L7 L8

<CR> <LF> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <CR> <LF>

<CR> <LF> <OK> <SP> <CR> <LF>

Note that a <----> character sequence indicates that there is either no response from a level (i.e. there is no actual router connected), or there is no route made to the specified output on that level.

The HyperTerminal session for this and the previous example is shown below:

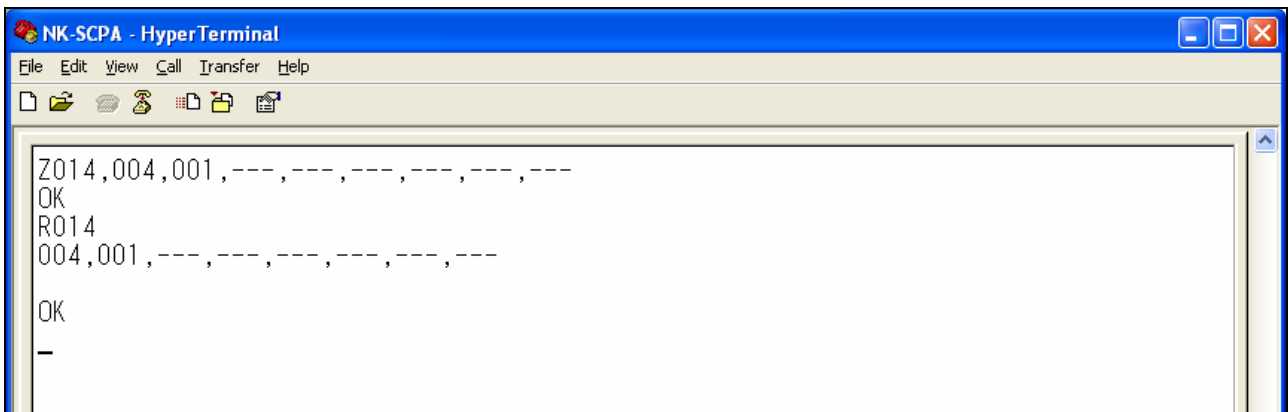


Figure 21 – HyperTerminal Session for Example 3

Appendix - NK-SCP/A ASCII Protocol

Characters within parentheses < > represent ASCII characters

Commands

The table below describes the protocol commands.

ASCII Protocol Commands

Command	Description
<X>	Switch an input to an output using a breakaway
<Z>	Switch multiple inputs (one per level) to an output
<R>	Read the current status of an output

ASCII Protocol Command Parameters

The table below describes the command parameters.

Command	Description	Value	Note
<ddd>	Destination – output number	000 to 254	1
<sss>	Source – input number	000 to 254	1
	Breakaway number	1 to 8	2
<--->	Blank entry (in place of actual output socket number)		
<CR>	Carriage return character		
<LF>	Line feed character		
<SP>	Space character		

Note 1 - An offset of -1 is required so that the lowest value of 000 points to the router input or output number 1. In this way inputs and outputs from 1 to 255 can be addressed with dialogue values from 000 to 254.

Note 2 - The breakaway number refers to the breakaway patterns as configured from the IPS. A breakaway may be active on more than one level.

Switch an input to an output using a defined breakaway

Start the sequence (11 bytes) with an upper case <X> followed by the destination, source, breakaway and carriage return as detailed below. Do not include the parentheses.

<X> <ddd> <,> <sss> <,> <CR>

Response from SCP

- If the data string is valid, the SCP returns response **(B)**.
- If the data string is invalid, the SCP returns response **(A)**.
- If the data string is valid but the source number is unavailable or any of the parameters are incorrectly specified, i.e. greater than 254 (255 to 999) for destination and source

and greater than 1 to 8 (0 or 9) for breakaway, the SCP returns response **(A)** then response **(B)**.

Response (A) <CR> <LF> <ERROR> <CR> <LF>

Response (B) <CR> <LF> <OK> <SP> <CR> <LF>

Switch multiple inputs (one per level) to an output

Start the sequence (37 bytes) with an upper case <Z> followed by the destination, then a source (or <---> to not switch a level) for each of 8 levels and carriage return as detailed below. Do not include the parentheses.

L1 L2 L3 L4 L5 L6 L7 L8

<Z> <ddd> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <CR>

Note: Send <---> in place of <sss> to not switch that level.

Response from SCP

- If the data string is valid, the SCP returns response **(B)**.
- If the data string is invalid, the SCP returns response **(A)**.
- If the data string is valid but the source number is unavailable or any of the parameters are incorrectly specified, i.e. greater than 254 (255 to 999) for destination and source and greater than 1 to 8 (0 or 9) for breakaway, the SCP returns response **(A)** then response **(B)**.

Response (A) <CR> <LF> <ERROR> <CR> <LF>

Response (B) <CR> <LF> <OK> <SP> <CR> <LF>

Read the current status of an output

Start the sequence (5 bytes) with an upper case <R> followed by the destination and carriage return as detailed below. Do not include the parentheses.

<R> <ddd> <CR>

Response from SCP

- If the data string is valid, the SCP returns response **(C)** then **(B)**.
- If the data string is invalid, the SCP returns response **(A)**.
- If the data string is valid but the destination is incorrectly specified, i.e. greater than 254 (255 to 999), the SCP returns response **(A)** then response **(B)**.

Response (A) <CR> <LF> <ERROR> <CR> <LF>

Response (B) <CR> <LF> <OK> <SP> <CR> <LF>

Response (C)

L1 L2 L3 L4 L5 L6 L7 L8

<CR> <LF> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <,> <sss> <CR> <LF>