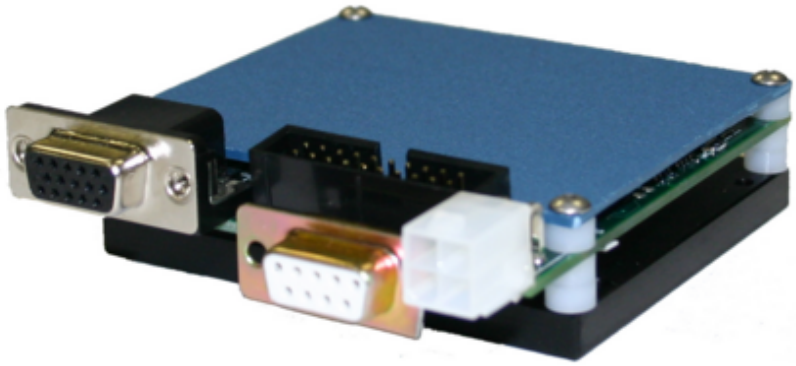
# SCA814 Servo Controlled Amplifier



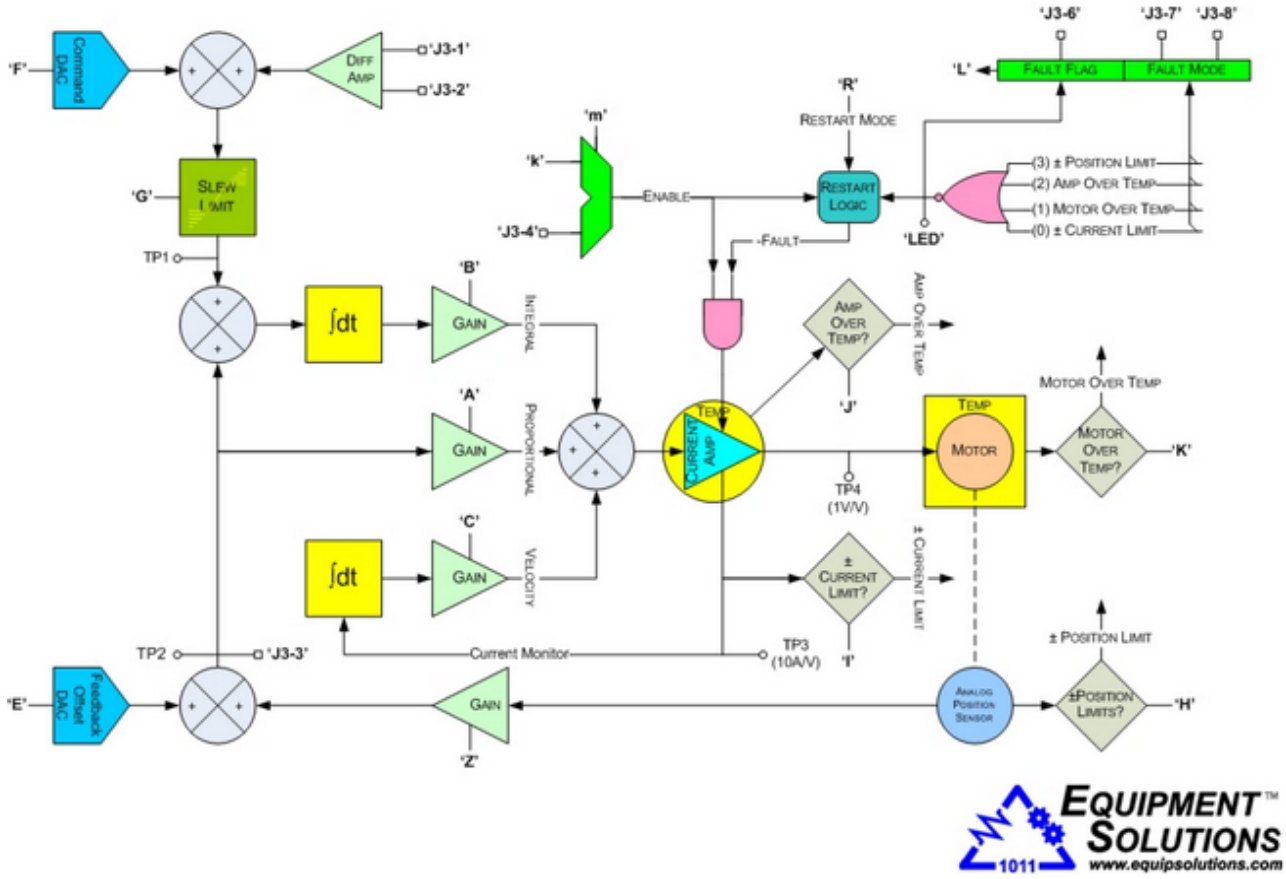
# Table of Contents

# Limited Warranty

Equipment Solutions warrants that its product will be free of defects in material and workmanship for a period of one year from the date of shipment. Equipment Solutions will repair or replace at its expense defective products returned by the Customer under a Return Authorization number issued by Equipment Solutions. This warranty is void if the product is damaged by "misuse" or "mishandling" by any party not under the control of Equipment Solutions. Equipment Solutions will determine misuse or mishandling. Misuse includes use of Equipment Solutions products with incompatible products resulting in damage to the Equipment Solutions product. The customer is responsible for charges for returning product for repairs. Equipment Solutions is responsible for charges for shipping product repaired under warranty back to the customer when Equipment Solutions is allowed to choose the carrier and level of service. The Customer is responsible for repair charges and all shipping charges for non-warranty repairs. Equipment Solutions's sole liability for any use of its product, regardless of the operating condition of such product, is limited to repair or replacement of the product. The Customer holds harmless and indemnifies Equipment Solutions from any and all other claims resulting from the use of Equipment Solutions products.

-o-

# SCA814 Servo Controlled Amplifier
## Servo Diagram




-o-

The SCA814 Servo Controlled Amplifiers have the unique and potentially high useful ability to be networked together. This peer-to-peer like architecture is depicted in Figure 1: SCA814 Servo Controlled Amplifier Network. In this configuration each servo controller is given a unique address that it will respond to. The range of valid address is between 129 and 255 inclusively. Address 128 is a global address that all servo controllers will respond to. Since all servo controllers share the same RS232 transmit line a conflict can exist if multiple servo controllers are on the network with the same address or the global address is used. At the precise moment when a new servo controller address is selected using the Controller ID command all servo controllers except the one matching the included address will disconnect their transmit line from the network

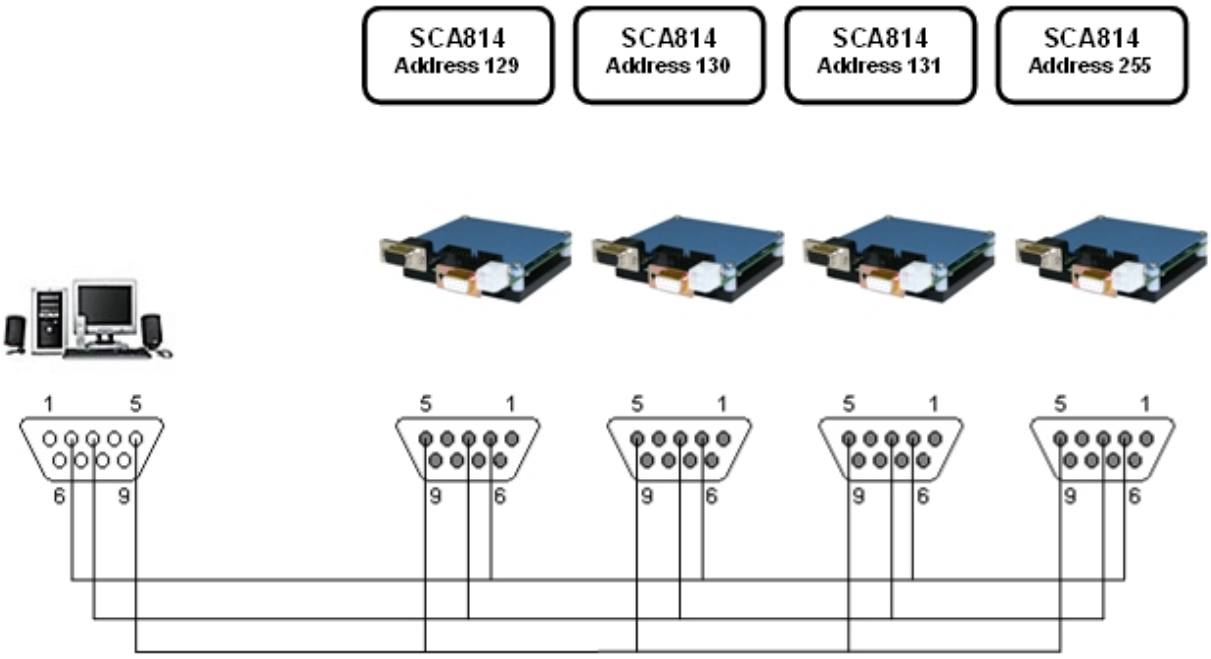



**Figure 1: SCA814 Servo Controlled Amplifier Network**

-o-

The following procedure outlines the process to configure a SCA814 Servo Controlled Amplifier with a unique network address.

1. With all other SCA814 Controllers disconnected from the host PC connect the SCA814 Servo Controlled Amplifier to be configured to the host.
2. Send the global address "128" to the autocollimator. In HyperTerminal this can be achieved by pressing and holding the ALT key while pressing and releasing the keys 0, 1, 2 and 8 from the numeric keypad.
3. Change the network address using the Controller ID command as described in this manual. Make sure that the address given is not the same as another SCA814 Controller already on the target network. Note that the address is automatically saved within the SCA814 Controller every time that it is updated.

-o-

The following procedure describes a process for determining a SCA814 Servo Controlled Amplifier's network address.

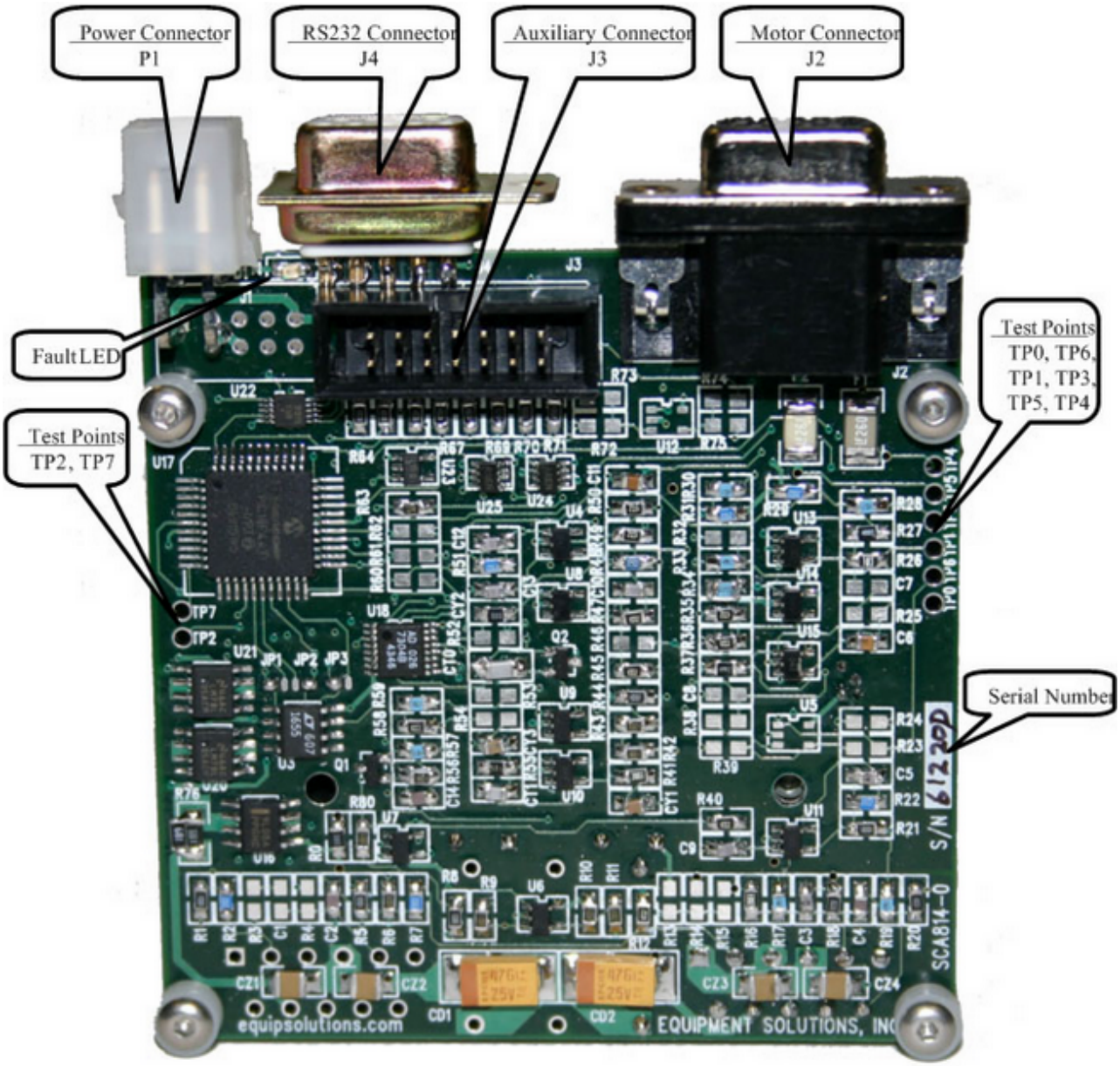
1. With all other collimators disconnected form the host PC connect the SCA814 Servo Controlled Amplifier to be evaluated to the host.
2. Send the global address "128" to the autocollimator. In HyperTerminal pressing and holding the ALT key while pressing and releasing each of the 0, 1, 2 and 8 keys from the numeric keypad can achieve this.
3. Send the Controller ID command to the autocollimator without the optional address. The autocollimator will respond with its network address expressed as a hexadecimal value in a ASCII string format.

-o-

# Networking basics for the SCA754 Servo Controlled Amplifier

When the SCA754 Controller is initially powered up or reset it enters into a state where it both listens (as a Listener) for commands and echoes them (as a Talker) back to the host. When multiple controllers are networked this situation will cause communications crashes as well as possible erratic behavior. To avoid this situation it is recommended that the first thing sent out is a valid controller address. When a valid network address is sent out only the controller with that address will echo that address character. This behavior provides a simple means to scan through all network address and determine which and how many controllers are present. When a controller detects that its network address has been sent it will automatically place itself into both a Listener and Talker mode. All other controllers on the network will reconfigure their Talker status so that they will not respond. If the previous network address was the Global Network Address (128) then all other controllers will remain in their Listener state. This is a good way to synchronize events across the network such as coordinated motion. If the previous network address was a specific controller address then that controller and all others not addressed will not respond in any way to commands that follow.

-o-


Power Connector
P1
RS232 Connector
J4
Auxiliary Connector
J3
Motor Connector
J2
Fault LED
Test Points
TP2, TP7
Test Points
TP0, TP6,
TP1, TP3,
TP5, TP4
Serial Number
S/N 61220D
SCA814-0
equipsolutions.com
EQUIPMENT SOLUTIONS, INC

-o-

## Power Connector (P2)

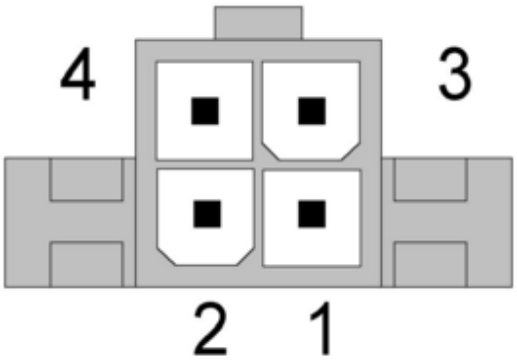



**Figure 3:Power Connector (P2)**

One connector is the Power Connector (P2). It is used to supply power to the SA754 and its daughter cards. The electronic architecture of the SA754 requires that a bipolar (±DC Voltage) be supplied to it. The SA754 has onboard regulators to make all other necessary voltages. A mating connector and pins are typically provided. Pre-made power cables of various lengths as well as Power Supplies are available from Equipment Solutions. The mating four-circuit plug is a Mini-Fit, Jr. connector made by Molex. The part number is either Non-strain relief 39-01-2040 (94V-2), 39-01-2045 (94V-0) or, strain relief 39-01-3042 (94V-2), 39-01-3048 (94V-0). These connectors use a crimp female style pin (Molex part number 39-00-0039).

The SA754 is typically configured to use ±24VDC supplies. For many applications ±15VDC is sufficient. The SA754 can be upgraded to use up to ±48VDC supplies. For most scanning applications the amplifier draws large current peaks when the motor accelerates then draws very little current at constant position or constant velocity. The current peaks can be as high at ±12 amps for a couple milliseconds, but ±4 amps for one millisecond is more typical. The average current required is rarely more than 2 amps but the supply(s) must be capable of supplying the peaks. Equipment Solutions recommends using linear supplies rather than switchers. The linear supplies typically have lower noise and are equipped with large output capacitance.

**Table 1:Power Connector (P2) Pinout**

| Pin # | Signal | Description |
|---|---|---|
| 1 | GND | Ground |
| 2 | GND | Ground |
| 3 | +V | Plus motor supply voltage to the Power Amplifier Card. |
| 4 | -V | Minus motor supply voltage to the Power Amplifier Card. |

-o-

# Motor Interface (J2)

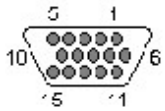



**Figure 4:Motor Connector (J2)**

The Motor Connector (J2) is used to interface the motor and it's sensors to the SA814. The connector is a High-Density DB-15 style female connector. A mating high-density 22D type plug is available from several manufacturers including AMP under part number 748364-1 using crimp style pin part number 748333-4. Determining the correct polarity of both the motor and associated position sensor is often accomplished through a trial-and-error process. Usually only one of the two sources (motor or sensor) should be varied. Fortunately the motor usually associated with the SA814 is a single-phase motor. That means that there are only two possibilities one correct and the other incorrect. When experimenting with these connections the precious motor loads (such as mirrors) should be considered. This procedure is only necessary if the SCA814 is to be connected to a non ESI motor since ESI insures this level of compatibility. A fuse for the motor is provided on the SA814. If the motor does not respond please consider the possibility that the fuse has blown and refer to the servicing procedures for further information.

**Table 2:Motor Connector (J2)**

| Pin # | Signal | Description |
|---|---|---|
| 1 | A- | Digital quadrature encoder phase A- input to the controller. |
| 2 | -SIG / A+ | Position feedback signal for stages configured with an analog feedback sensor OR digital quadrature encoder phase A+. Note that the controller does not use the digital quadrature encoder phase A+. |
| 3 | INT2 / B- | Primarily provides a interface between a motor with an integrated temperature sensor and the digital controller. Alternatively used as general-purpose analog (0-5V) input. |
| 4 | INT1 / B+ | Provides a general-purpose digital input or output pin. This line can source or sink up to 20 ma. |

| | | |
|---|---|---|
| 5 | PROM / I- | Provides an interface to the ESI Plug-N-Go Motor interface. If the motor does not support this feature this line may be alternatively defined as either a general-purpose digital input or output signal. This line can source or sink up to 20 ma. In the Plug-N-Go configuration the line is normally tri-stated. |
| 6 | MTR+ | Plus Motor Supply Current. |
| 7 | MTR+ | Plus Motor Supply Current. |
| 8 | MtrTemp | |
| 9 | +5VDC | Feedback sensor power. |
| 10 | +SIG / I+ | |
| 11 | MTR- | Minus Motor Supply Current. |
| 12 | MTR- | Minus Motor Supply Current. |
| 13 | GND | Ground. |
| 14 | GND | Ground. |
| 15 | GND | Ground. |

-o-

# AUXILIARY HEADER (J3)

The SC814 Board has a 14 pin Auxiliary Header that provides the user with access to auxiliary control and IO signals. The connector is a male two-row header that mates to industry standard IDC ribbon cable connectors. Pin 1 of the header is indicated on the header shell by a triangle. The odd number pins are on one side while the even pins are on the other side with pin 2 next to pin 1.

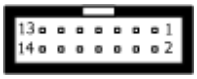



**Figure 6 J3 - Auxiliary Header**

**Table 6 Auxiliary Header Description**

| Pin # | Signal | Description |
|---|---|---|
| 1 | +Cmd | Analog input voltage that is directly proportional to commanded position. This pseudo differential ±2.5V DC signal, in reference to J3-2 is directly added to the software commanded Position level. As such, the user will typically want to set the software command position level to 32767, or midscale while using this input. Please contact the factory if the specified voltage range is incompatible with your application requirements. |
| 2 | -Cmd | Please refer to the description of J2-1. |
| 3 | AGnd | Analog Ground |
| 4 | Fdbk | ±4V voltage level that is directly proportional to motor position. Note that this signal is identical to that described in the Test Points section. Alternatively, the feedback signal level can be reported using the Motor Position command. |
| 5 | -Enable | Depending on the state of the Enable Source this TTL level input could be used to express whether the amplifier should be in its enabled (logic low) or disabled (logic high) state. Note that this pin has a weak pull-up associated with it and so is normally in the disabled state. |

| | | |
|---|---|---|
| 6 | Fault | This same information can be acquired through use of the Amp Fault command |
| 7 | Fault1 | The combination of these two TTL signal lines are used to communicate the cause of the latest fault. For the following descriptions consider Fault1 as the LSB and Fault2 as the MSB. A value of zero (0) indicates an over current condition, a value of one (1) indicates a motor over temperature condition, a value of two (2) indicates a board over temperature condition and a value of three (3) indicates a position limit condition. This same information can be acquired through use of the Amp Fault command.<br><br>**NOTE** : This hardware feature was deleted in firmware version date code 12-Sep-08 as follows; |
| 8 | Fault2 | |
| 9 | IO1 | Programmable user digital IO pin capable of sourcing or sinking 10mA. Note that the IO1 command can be used for configuration and state of this signal line. |
| 10 | IO2 | Programmable user digital IO pin capable of sourcing or sinking 10mA. Note that the IO2 command can be used for configuration and state of this signal line. |
| 11 | Reset | A low on this normally high pin causes the microprocessor to be reset. |
| 12 | TX | RS-232 transmit line. |
| 13 | RX | RS-232 receive line. |
| 14 | GND | Digital Ground |

-o-

# Test-Points

The Amplifier board contains three test-points on a short edge near the edge with the connectors. These test points are labeled GND, TP1, TP2, TP3 and TP4.

**Table 3: SCA814 Test-Points**

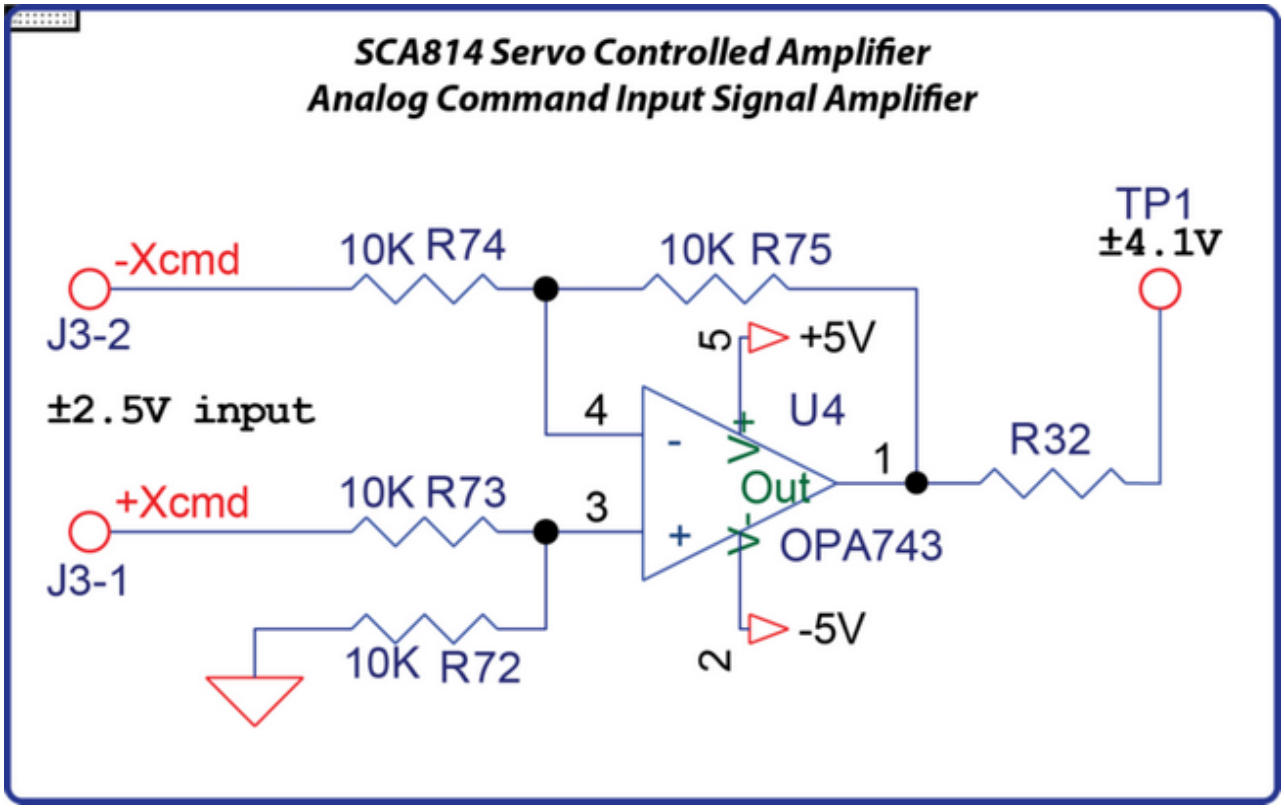
| TP# | Description |
|---|---|
| TP0 | Ground |
| TP1 | Command, after Slew Rate Limiting scaled at ±4V |
| TP2 | Feedback Position scaled to ±4V. |
| TP3 | Motor Current scaled at 10A/V |
| TP4 | Motor terminal voltage scaled at 1V/V. |

-o-

# Analog Position Input Interface

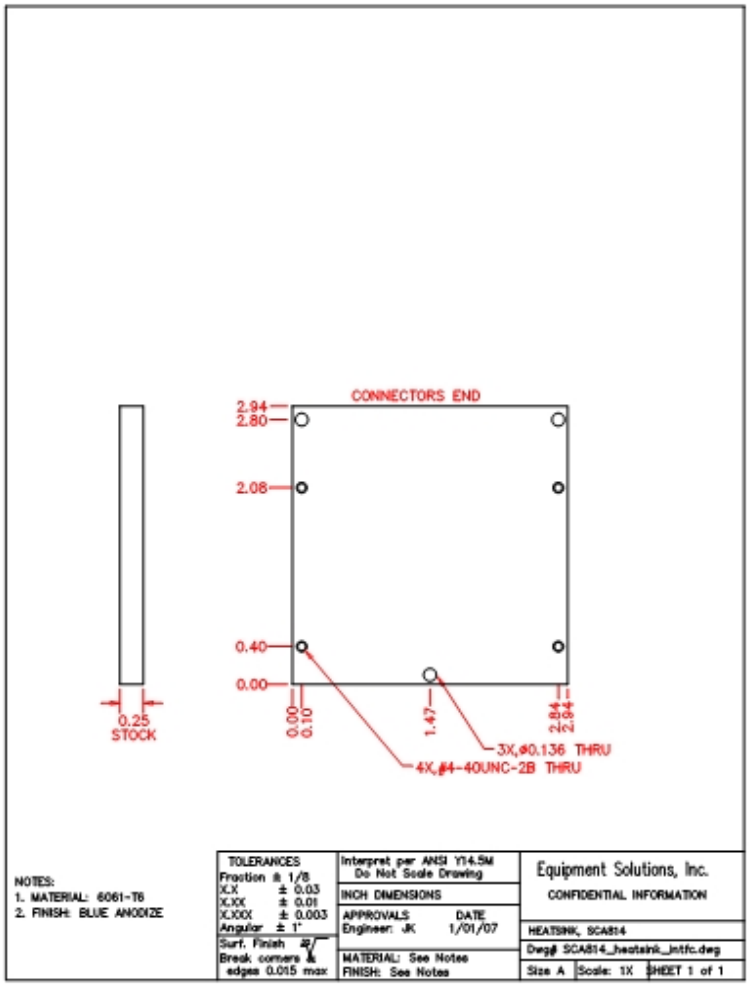
The SCA814 Servo Controlled Amplifier provides for a means to control the motor position through a simple analog input. For situations where the position needs to be updated very quickly, in excess of 1kHz, this control method should be considered. Based on the SCA814 Servo Controlled Amplifier Feedback Signal Configuration, in its standard configuration, the input signal is designed to accept a ±2.5 volt DC signal. That voltage will map directly to the complete travel of the stage it is connected to. For example if the SCA814 Servo Controlled Amplifier is connected to a VCS-10 Voice Coil Stage, which has 10 millimeters of travel, then -2.5 volts will represent on extreme of its travel and +2.5 volts the other extreme. By definition therefore, zero volts will correspond to the stages center of travel. This means that there is approximately a 5V/10mm or 0.5V/mm or 0.5mV/um translation.

It is possible through adjustment of onboard resistors to scale the input to a different voltage range. Note that the range needs to be centered about zero volts. Contact Equipment Solutions to discuss providing a pre-configured custom Analog Command Input Signal.




-o-

The SCA814 is provided with a 0.25 inch thick plate. This plate provides two important functions 1) heat dissipation for the on board high current motor amplifier and 2) a convenient mounting interface. If at all possible, it is recommended that the SCA814 is directly mounted a heat conducting surface to further promote heat dissipation. If this is not possible, some form of ventilation may be required for reliable continuous use.




-o-

# Interfacing to the SCA814 through HyperTerminal

Communications between the SCA814 and a Windows based system can be easily achieved using the Windows supplied HyperTerminal application. Note that sometimes this application is not loaded as part of the "Normal" Windows build process. Refer to the Control Panel "Add/Remove Programs" section if the application is not present.

The first screen presented during the HyperTerminal startup process is the "Connect To" screen as shown in Figure 6: HyperTerminal "Connect To" Screen. The only item of interest on this screen is the "Connect using" list box. Select the Serial Communications port number that is connected to the SCA814 from the list. Make sure that no active application such as "HotSync" is using the target port, as it will prevent a successful connection to the SCA814. When done click on the "OK" button.


**Figure 6: HyperTerminal "Connect To" Screen**

The HyperTerminal application will then display the "Properties" screen as shown in Figure 7: HyperTerminal "Properties" Screen. Every field within this screen is relevant and critical to the successful communications with the SCA814. The "Bits per second" should be set to 115200, the "Data bits" set to 8, the "Parity" set to None, the "Stop bits" set to 1 and finally the "Flow control" set to None. Figure 7: HyperTerminal "Properties" Screen shows the proper settings. Once all settings have been properly set the "OK" button should be pressed.

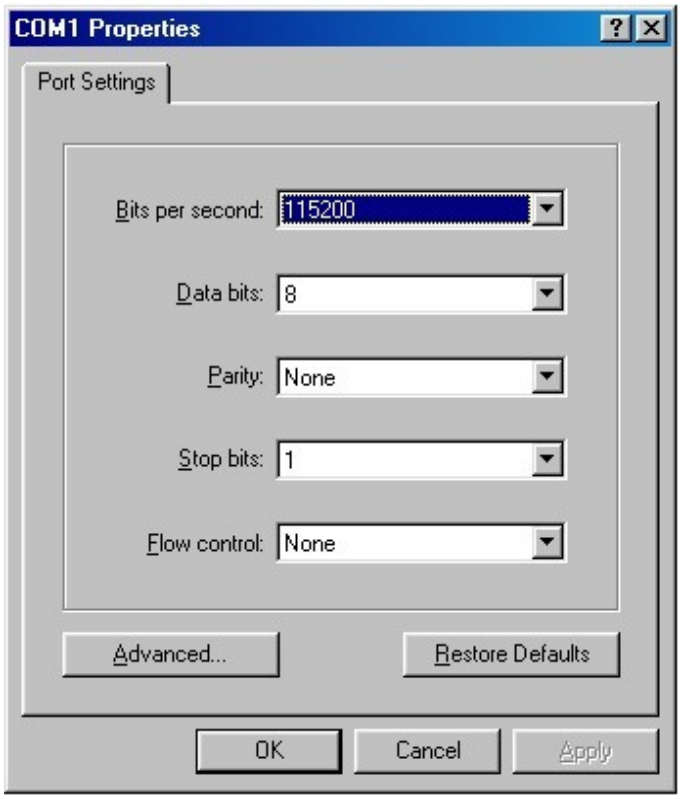



**Figure 7: HyperTerminal "Properties" Screen**

The HyperTerminal application will then present its "Terminal" page as shown in Figure 8: HyperTerminal "Terminal" Screen. The application will also automatically connect the assigned serial port to the SCA814. At this point communications with the SCA814 should be possible. As a quick test of the communications link a character can be typed at the PC. If the communications link is valid the SCA814 should receive that character and echo it back to the PC. The HyperTerminal application should intercept that character and display it on its Terminal screen.

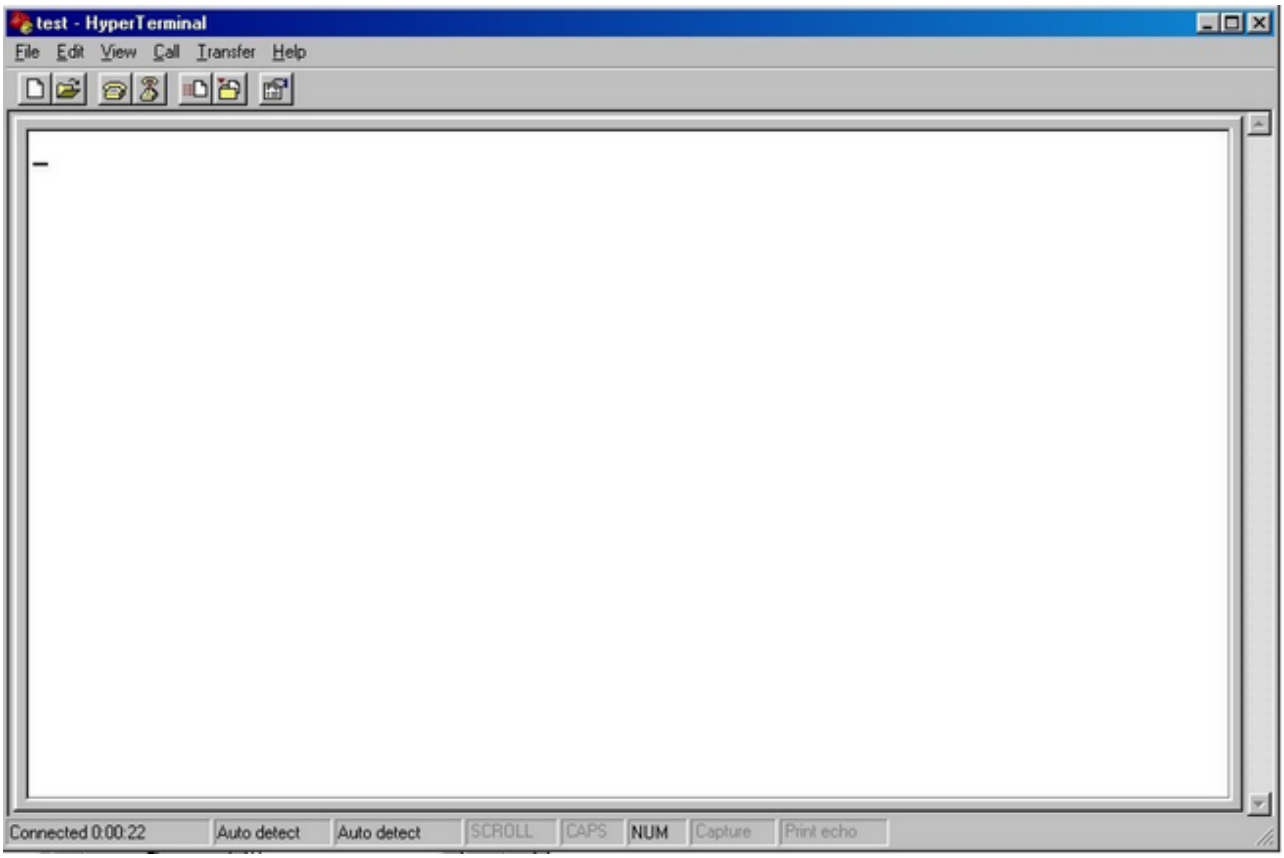
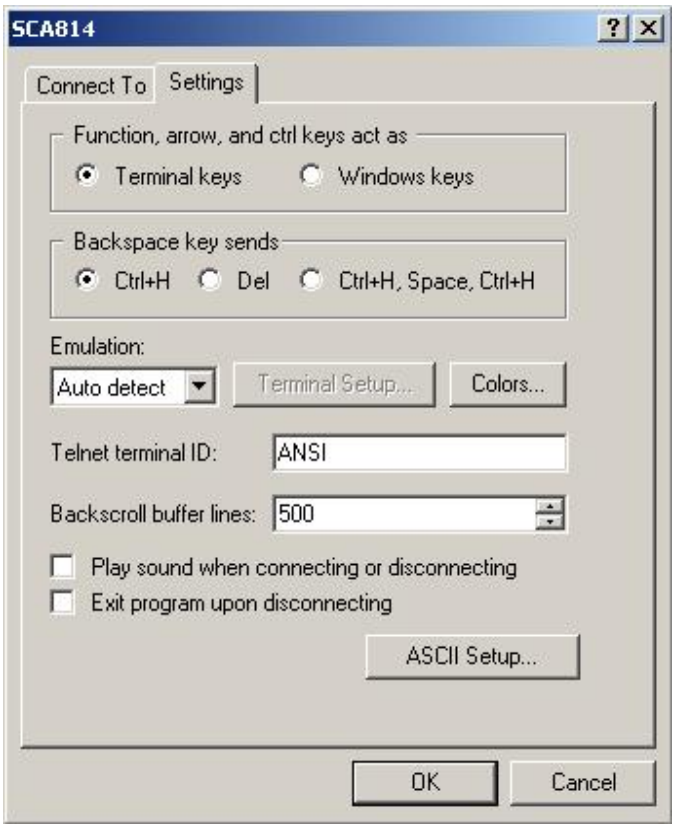


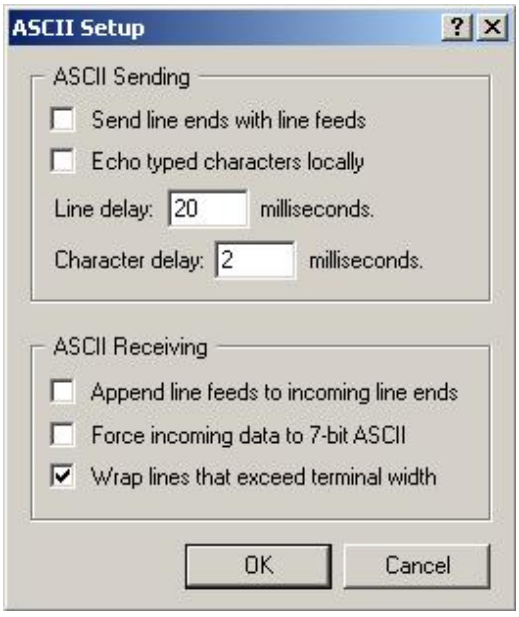

**Figure 8: HyperTerminal "Terminal" Screen**

If the entered character is not displayed the failure can be explained by several possibilities including; the physical port connected to the PC does not match the port just selected, the port was not properly configured, another application is using the port, the serial cable is not wired correctly, the SCA814 is not powered up or functional.

In some special cases such as downloading profile data or firmware updating it is necessary to set certain delays. These delays provide the controller with time to process the information sent. To set these delays select the Properties entry under the File menu. The Properties page will be displayed and will have two (2) tabs associated with it. Click on the Settings tab.




Near the bottom of the page will be a button labeled ASCII Setup...,click on it and a new window identified as ASCII Setup will be displayed.




Enter a value of 20 into the Line delay field and 2 into the Character delay field. Click on the OK button and then OK at the bottom of the original Properties page. At this point the assigned delays will be

applied to end-of-line and end-of-character processes. If there is any hint of communications failure such as early termination or error messages you may want to increase the previously set values.

-o-

# Interfacing to the SCA814 through a Custom Application

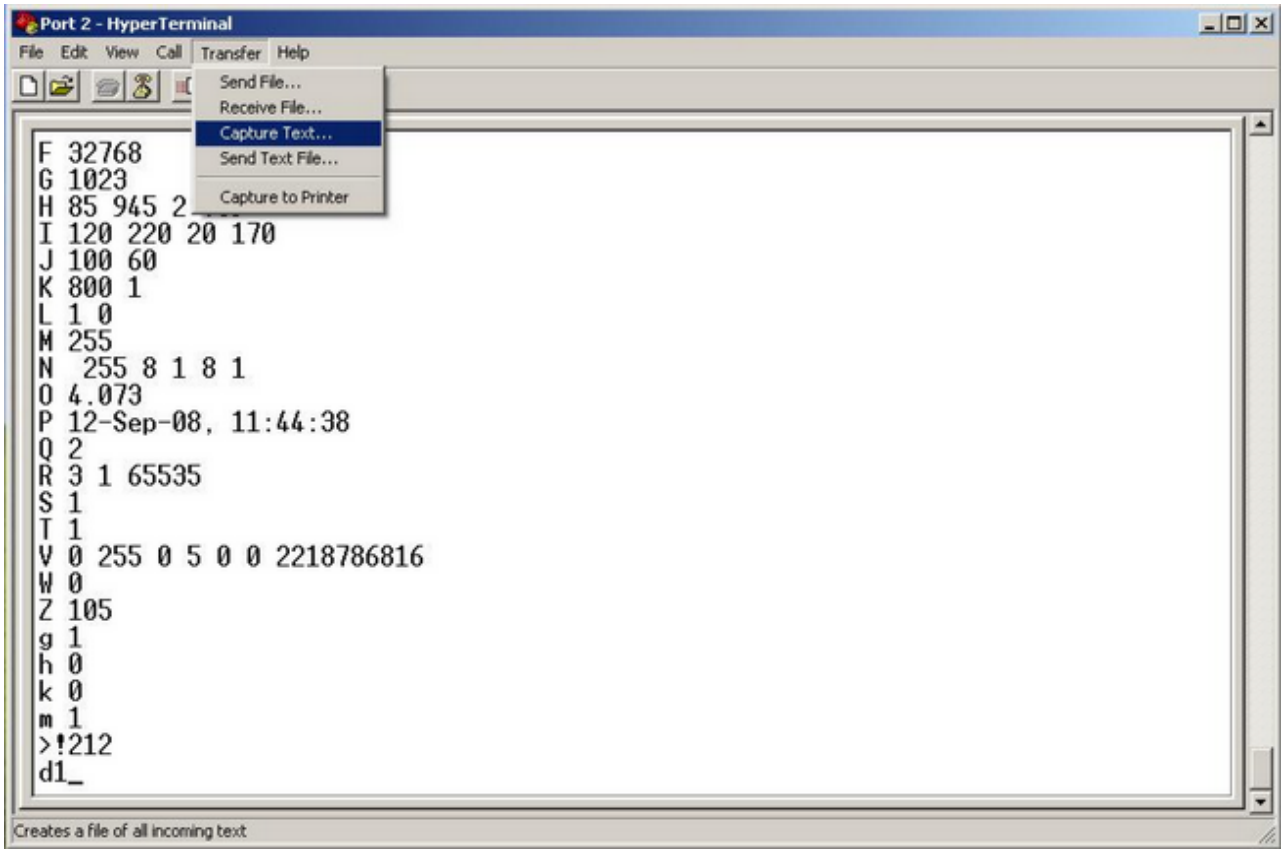
The SCA814 has a single character input buffer that can be overflowed if the proper steps are not taken. To avoid overflowing the input buffer the user should send a single character at a time and wait for that same character to be echoed back by the controller. While not necessary, it is advisable to verify that the character received from the controller is the same character sent. Once the character is received the next character can be processed. The exception to this process occurs when the carriage return is sent. The carriage return indicates an end-of-command which causes the controller to begin deciphering and processing the message. The controller will send a ">" character as an indicator that it has entered this phase of the command processing. Depending on the command, many more characters may be sent. When the command has been completely processed it will then send a Carriage Return and Line Feed character. The application should wait for this response before sending the next command.

-o-

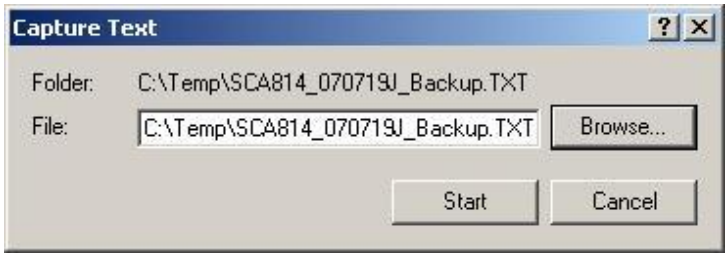
# Backing-Up the SCA814 through HyperTerminal

As described in various parts of this manual, the SCA814 Servo Controled Amplifier has many user configurable parameters. These parameters play a vital role in the amplifiers operating characteristics. With the introduction of the dump command it is now possible to save all SCA814 Servo Controlled Amplifier configuration information. The purpose of this section is to describe a semi-automatted method of saving that data. A subsequent discussion, Restoring the SCA814 through HyperTerminal, will describe how to download the information aquired here into the SCA814 Servo Controlled Amplifier.

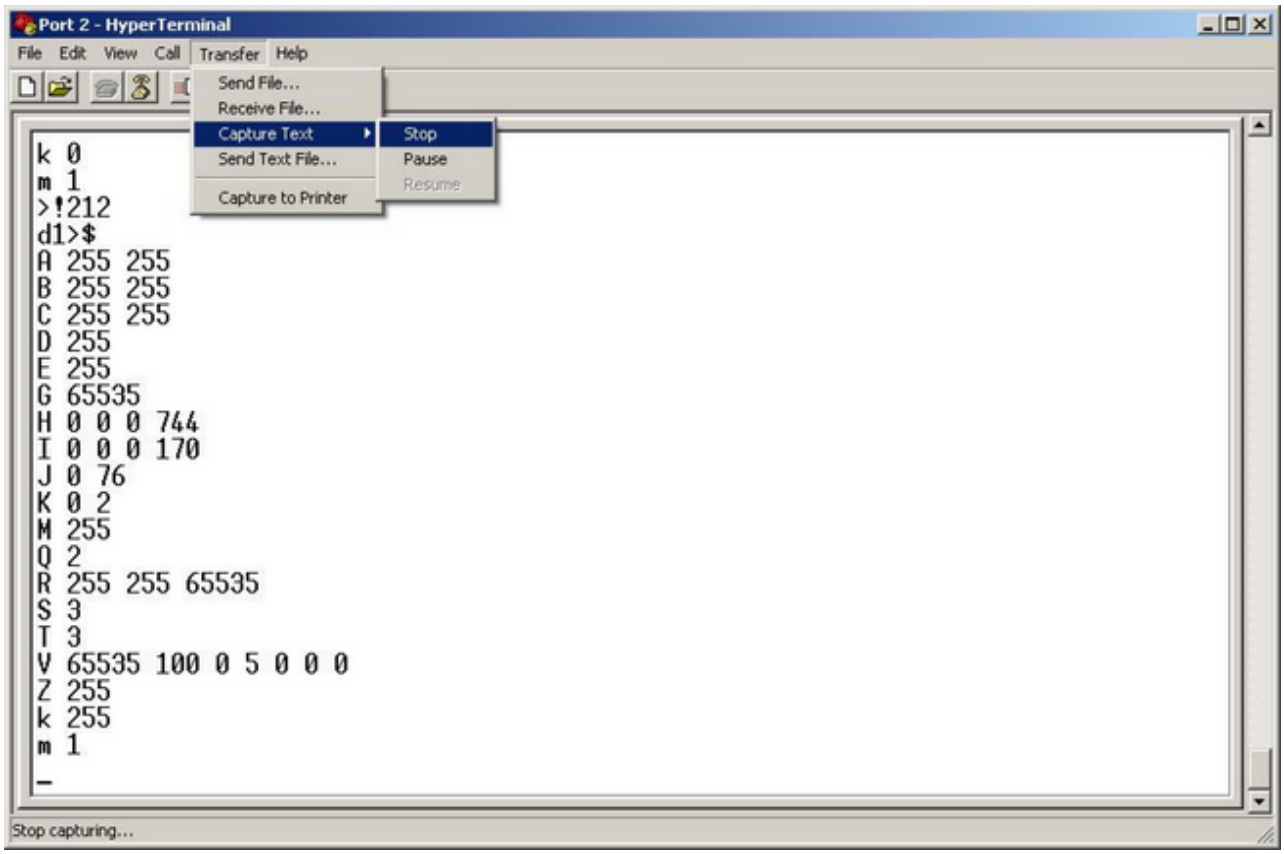
With the SCA814 Servo Controlled Amplifier powered-up and its serial communications established, as described in Interfacing to the SCA814 through HyperTerminal, enter the characters "d" and "1" at the terminal window as shown then select by clicking on the "Capture Text..." menu item found under the Transfer menu.



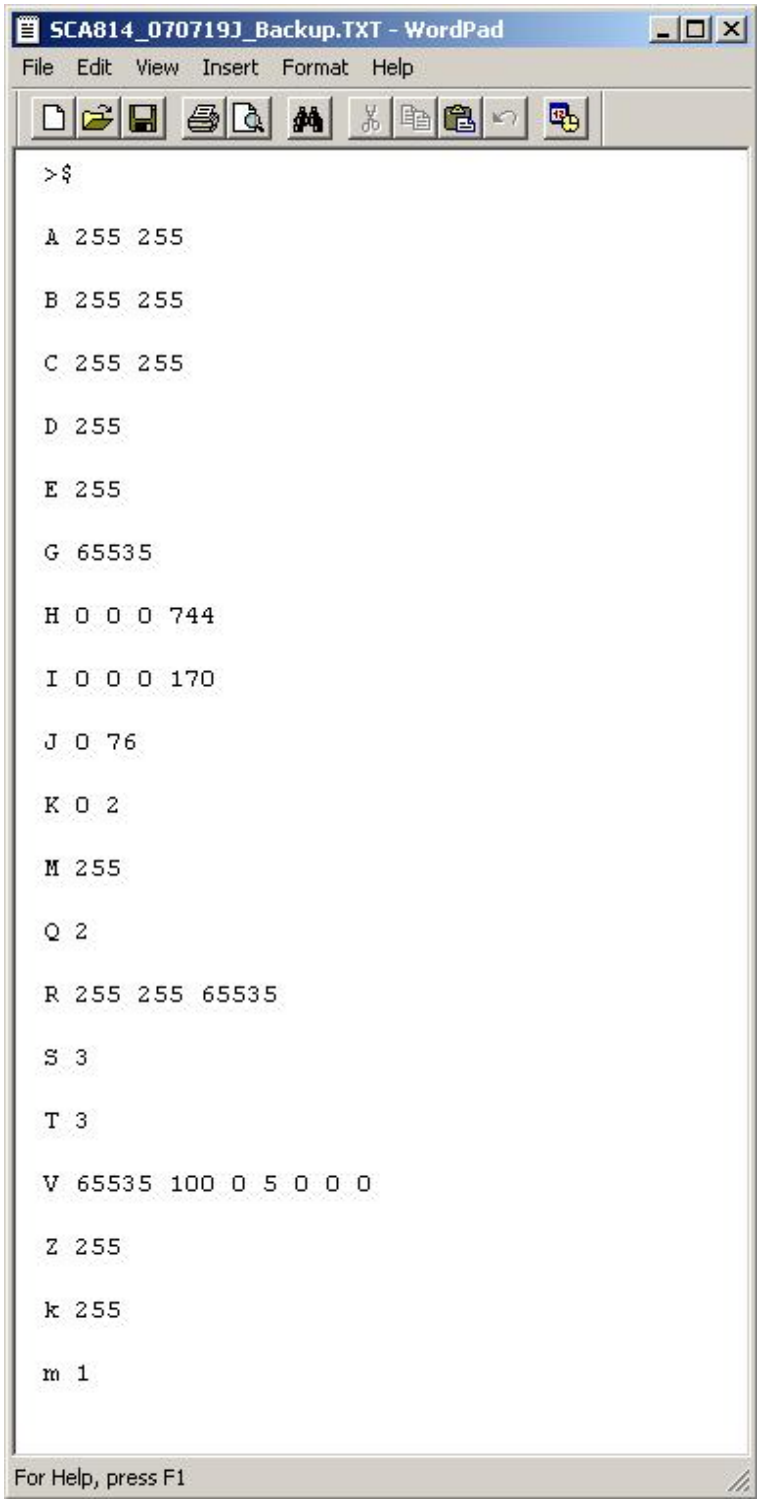

At this point a new window will be displayed that allows the user to select a file name under which all data to be captured will be saved. Select a path and file name and click on the "Start" button.




With the cursor active inside the terminal window, press the "Enter" key. This will cause the previous "d1" command to be executed. Parameter labels and their associated values will be sent to the terminal window. When all information has been sent, click on the the "Transfer" menu and select the "Capture Text" pull right item and finally click on the "Stop" menu item as shown below.

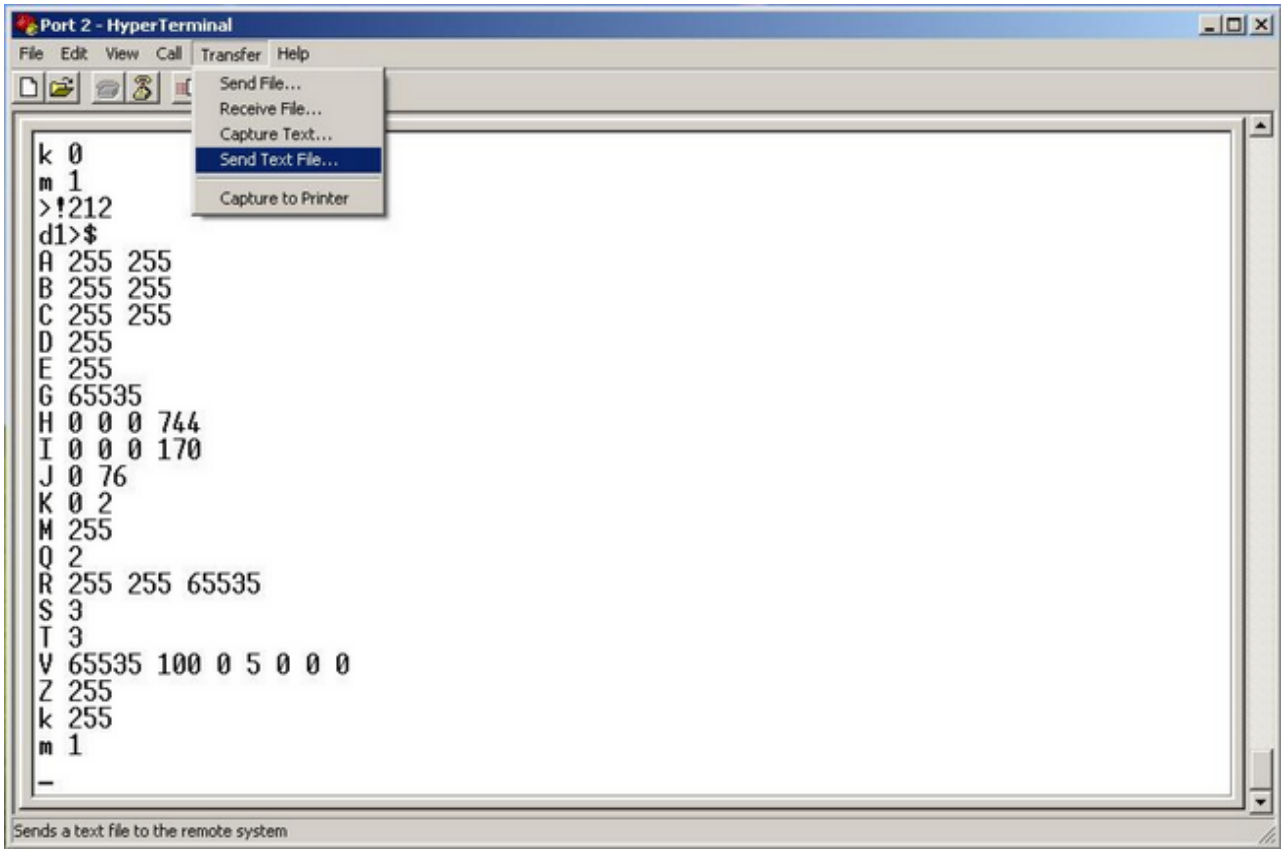


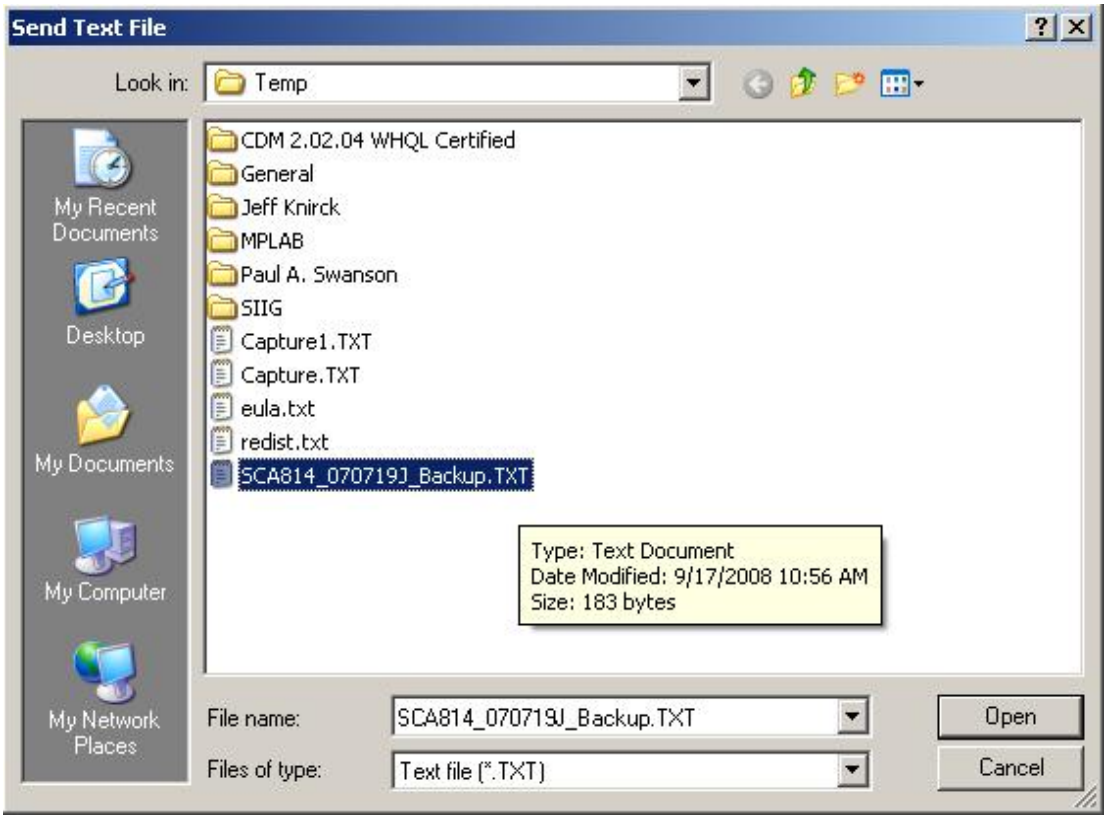

At this point, a file containing all the configuration data has been generated. It can be viewed by opening the file using Wordpad or another similar text processing application. An example of a configuration data backup file is shown below.


SCA814_070719J_Backup.TXT - WordPad
File Edit View Insert Format Help
>$
A 255 255
B 255 255
C 255 255
D 255
E 255
G 65535
H 0 0 0 744
I 0 0 0 170
J 0 76
K 0 2
M 255
Q 2
R 255 255 65535
S 3
T 3
V 65535 100 0 5 0 0 0
Z 255
k 255
m 1
For Help, press F1

-o-

# Restroring the SCA814 through HyperTerminal



-o-

# Profile Generator

The SCA814 profile generator provides a means to generate a position path at a fixed rate. A 128 byte section of non-volatile memory is used to store this position profile. Another section of non-volatile memory is also used to store related configuration data. At power-up all this information is automatically transferred into the SCA814's high speed memory.

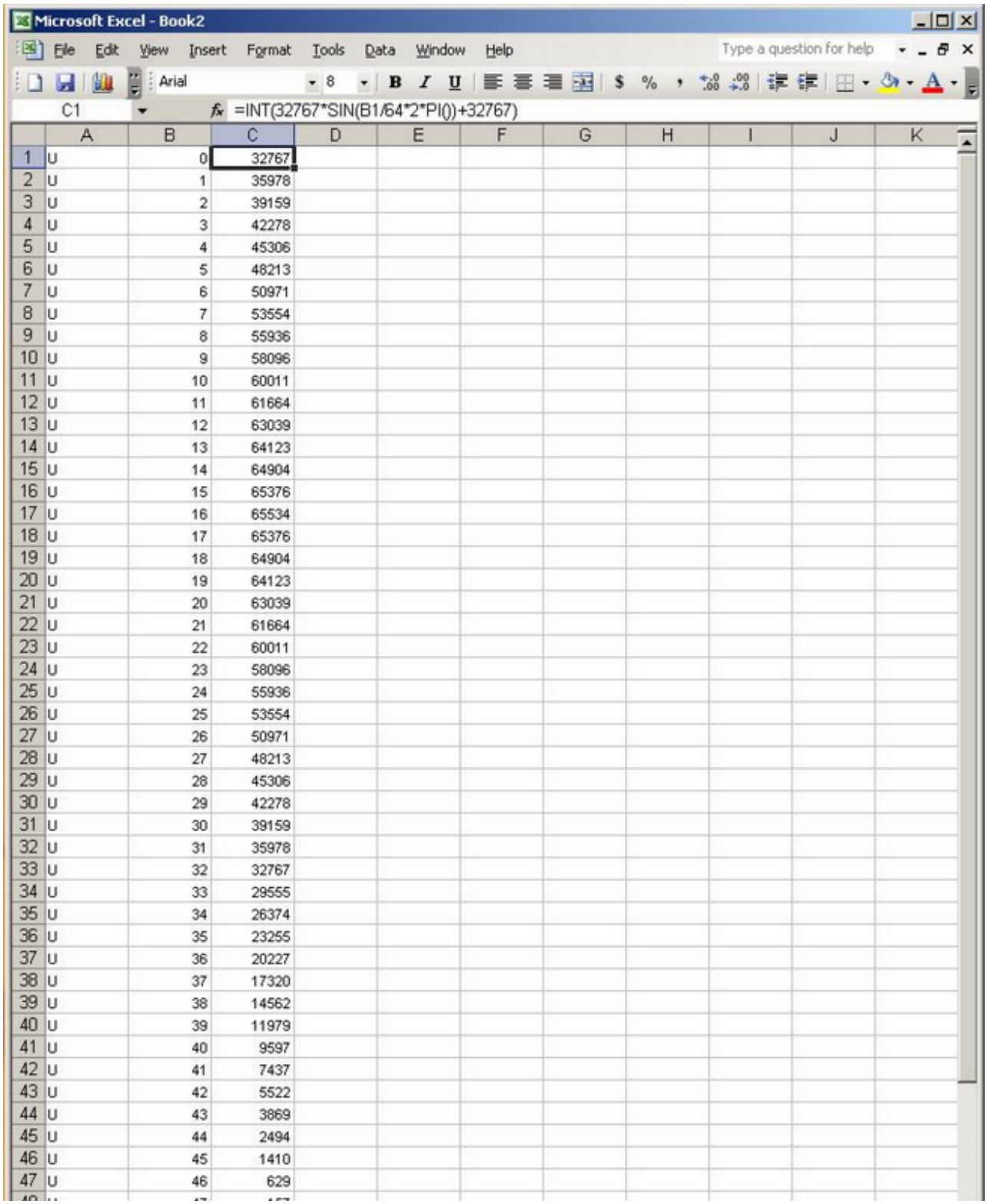
The profile generator can be operated in one of two different modes as described in the profile configuration command. Mode zero (0) or point to point mode provides a means to describe a 64 point position path. It is usually best to only consider this mode for non-linear position paths. An example of such a position path might be one described by a sine function. Each of the points is expressed as a 16-bit DAC value. Using the profile data command each profile point is entered into the SCA814.

-o-

# Creating and Downloading a Point to Point Profile

A convenient way to create and enter such a profile is by using a combination of Microsoft Excel and HyperTerminal. The following process describes in detail a method for generating a 64 point sine wave position path for the SCA814.

1) Creating a Position Profile through Microsoft Excel
    a) Open Microsoft Excel.
    b) In cell B1 enter the value "0".
    c) With cell B1 selected, and the mouse over the "Edit" menu, select "Fill" then "Series".
    d) Click the "Columns" from the "Series In" pane.
    e) Set "Step Value" to 1 and the "Stop Value" to 63.
    f) In cell A1 enter the character "U"
    g) Copy cell A1 and paste into cells A2 through A64
    h) In cell C1 enter the following "=int(32767*sin(B1/64*2*PI())+32767)
    i) Copy contents of cell C1 and paste into cells C2 through C64
    j) At the File menu select "Save As"
    k) In the "file name" field enter a name for this data set.
    l) In the "Save as type" field select "Text(Tab delimited)(*.txt)".
    m) Exit Microsoft Excel.



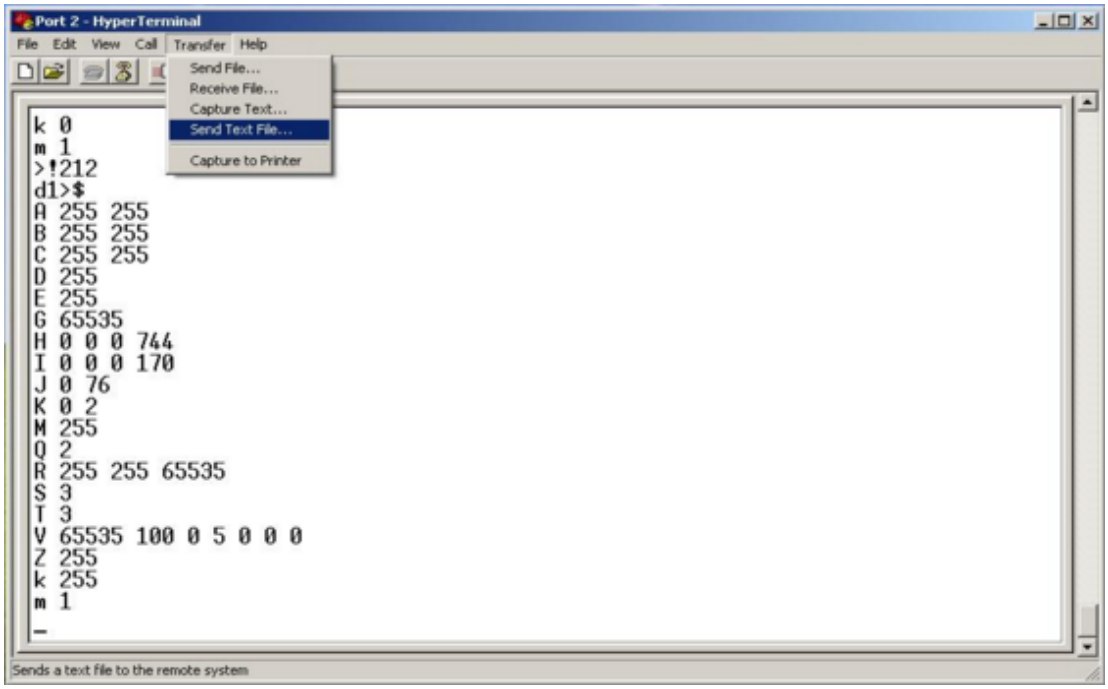

| | A | B | C |
|---|---|---|---|
| 1 | U | 0 | 32767 |
| 2 | U | 1 | 35978 |
| 3 | U | 2 | 39159 |
| 4 | U | 3 | 42278 |
| 5 | U | 4 | 45306 |
| 6 | U | 5 | 48213 |
| 7 | U | 6 | 50971 |
| 8 | U | 7 | 53554 |
| 9 | U | 8 | 55936 |
| 10 | U | 9 | 58096 |
| 11 | U | 10 | 60011 |
| 12 | U | 11 | 61664 |
| 13 | U | 12 | 63039 |
| 14 | U | 13 | 64123 |
| 15 | U | 14 | 64904 |
| 16 | U | 15 | 65376 |
| 17 | U | 16 | 65534 |
| 18 | U | 17 | 65376 |
| 19 | U | 18 | 64904 |
| 20 | U | 19 | 64123 |
| 21 | U | 20 | 63039 |
| 22 | U | 21 | 61664 |
| 23 | U | 22 | 60011 |
| 24 | U | 23 | 58096 |
| 25 | U | 24 | 55936 |
| 26 | U | 25 | 53554 |
| 27 | U | 26 | 50971 |
| 28 | U | 27 | 48213 |
| 29 | U | 28 | 45306 |
| 30 | U | 29 | 42278 |
| 31 | U | 30 | 39159 |
| 32 | U | 31 | 35978 |
| 33 | U | 32 | 32767 |
| 34 | U | 33 | 29555 |
| 35 | U | 34 | 26374 |
| 36 | U | 35 | 23255 |
| 37 | U | 36 | 20227 |
| 38 | U | 37 | 17320 |
| 39 | U | 38 | 14562 |
| 40 | U | 39 | 11979 |
| 41 | U | 40 | 9597 |
| 42 | U | 41 | 7437 |
| 43 | U | 42 | 5522 |
| 44 | U | 43 | 3869 |
| 45 | U | 44 | 2494 |
| 46 | U | 45 | 1410 |
| 47 | U | 46 | 629 |

2) Downloading a Position Profile through HyperTerminal

   a) At a minimum, set the Profile Configuration such that the Cycles is zero (0), Start and Stop are 0 and 63 respectively and the Mode is set to zero (0).

   b) Follow all steps listed in Interfacing to the SCA814 through HyperTerminal section.

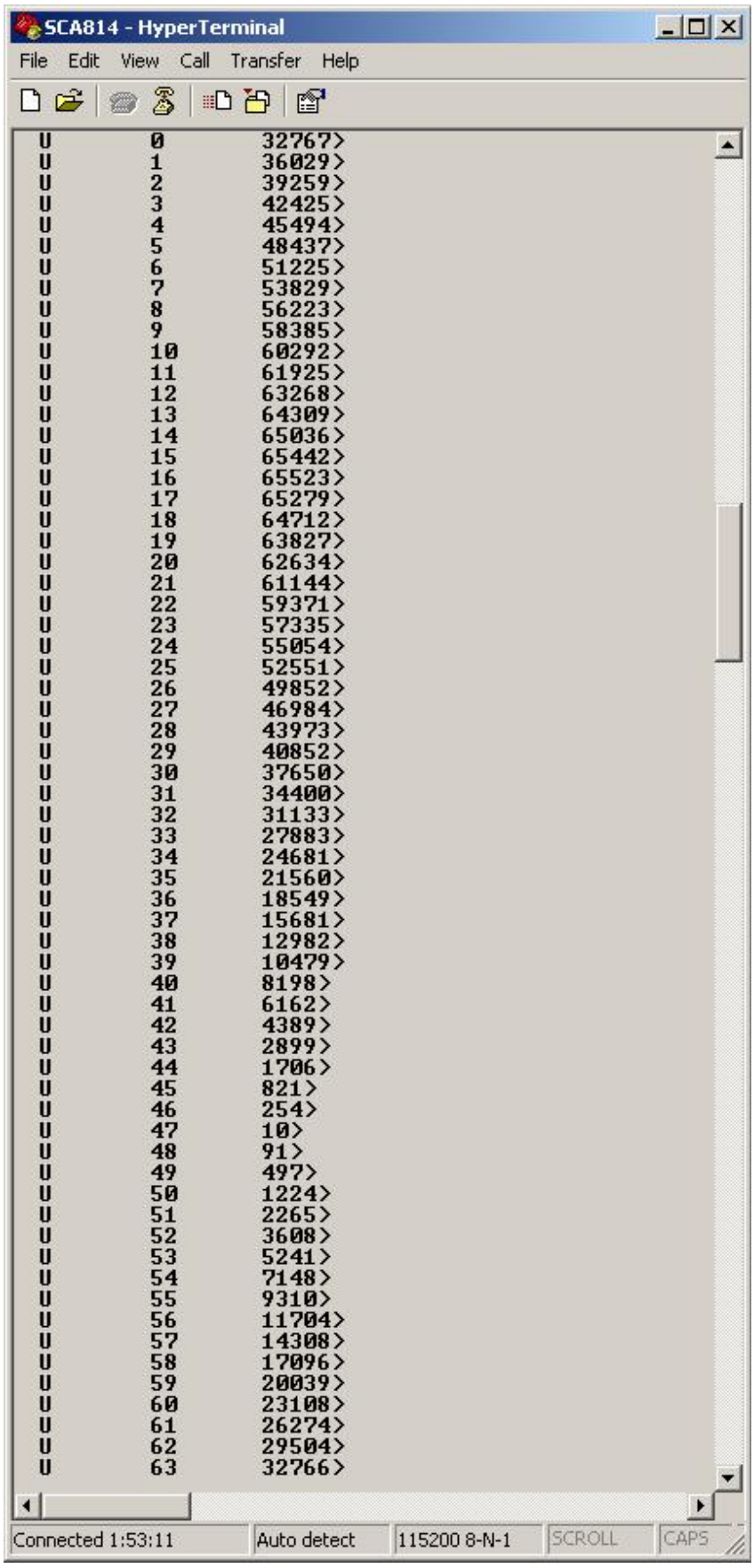
   c) From the Transfer menu select the "Send Text File..." item.




d) At the File Name field enter the name given during the Excel profile development above.

e) Click the Open button.

At this point data should start streaming from the PC, through its RS232 interface into the controller. You should not observe any thing other than the information contain in the text file itself. Witnessing "!400" within the data is suggestive that the delay times referred to in the Interfacing to the SCA814 through HyperTerminal section are not large enough.




To run the profile, set Cycles to a non-zero value. With a 64 point profile as described above, setting

Ticks to one (1) will cause the profile generator to produce a 610Hz sine wave profile. This waveform can be observed using an oscilloscope connected to Test Point #1.

-o-

# Creating a Few Examples of Linear Profiles

In order to properly tune a servo it is useful to feed it a square wave profile of both small and large amplitudes. This can be easily done using either the Point-to-Point or Linear modes. Below is an example of how to implement it in the Linear mode.

*A Small Amplitude Square Wave*

- V 0 255 0 3 1

The above Profile Configuration command causes the profile generator to stop running if it is running, set the Tick update rate at once every 6.5 milliseconds

- U 0 32000 0
- U 1 0 4
- U 2 34000 0
- U 3 0 4

The above Profile Data commands cause the profile generator to set the DAC to 32000, wait for 25msec, set the DAC to 34000 and then wait another 25msec.

- V 65535 255 0 3 1

The above Profile Configuration command cause the profile generator to begin running the above described profile for an indefinite number of cycles.

*A Large Amplitude Square Wave*

- V 0 255 0 3 1

The above Profile Configuration command causes the profile generator to stop running if it is running, set the Tick update rate at once every 6.5 milliseconds

- U 0 5000 0
- U 1 0 77
- U 2 60000 0
- U 3 0 77

The above Profile Data commands cause the profile generator to set the DAC to 5000, wait for 500msec, set the DAC to 60000 and then wait another 500msec.

- V 65535 255 0 3 1

The above Profile Configuration command cause the profile generator to begin running the above described profile for an indefinite number of cycles.

*A Triangle Wave*

- V 0 10 0 1 1

The above Profile Configuration command causes the profile generator to stop running if it is running, set the Tick update rate at once every 256 microseconds.

- U 0 10 100
- U 1 -10 100

The above Profile Data commands cause the profile generator to start add 10 DAC counts to the current DAC position 100 times for a total gain of 1000 DAC counts over 26msec. It then begins decrementing the DAC position by 10 counts and thus returning it back to its original start position with a total cycle time of 51msec or 20Hz.

- V 65535 10 0 1 1

The above Profile Configuration command cause the profile generator to begin running the above described profile for an indefinite number of cycles.

-o-

# Start-up Messages

## Start-up Messages

During the start-up phase of the SCA814 a RS-232 serial message will be sent out that indicates the reason for the start-up. They are as follows;

| | |
|---|---|
| WDT_TIMEOUT | 207 |
| MCLR_FROM_SLEEP | 211 |
| MCLR_FROM_RUN | 215 |
| NORMAL_POWER_UP | 212 |
| BROWNOUT_RESTART | 214 |
| WDT_FROM_SLEEP | 203 |
| RESET_INSTRUCTION | 200 |

NOTE: This feature was introduced in build 12-Apr-07 14:59:31. Please refer to the Compile Date Time message to learn if your controller has this feature.

-o-

## ASCII Command Set

The SCA754 Servo Control Amplifier supports many ASCII commands that can be issued across a simple RS-232 interface. To establish this communications link both hardware and software must be properly configured. To learn more about how to properly configure the hardware interface refer to the RS-232 Communications Cable section of this manual. With the hardware interface properly established a generic terminal application such as HyperTerminal can be launched. The Interfacing to the SCA814 through HyperTerminal section of this manual can be referenced for this purpose.

-o-

| **A** | **Read or Set Proportional Gain** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 65 | 0...255<br>0..255 | A127<cr> | > ###<cr> | Y |

The upper case **'A'** provides a means to read or set the proportional gain associated with the servo control law.

**NOTE:** Command was modified to include Startup Gain in firmware date codes of 31-Mar-08 and later

Startup Gain was introduced as a second parameter of this command to make startup more reliable under very aggresive tuning conditions. This second value is an initial value used during the servo Enable process. It is set to a value that will create a relative soft or "mushy" response to any servo error disturbance.

-o-

| **B** | **Read or Set Integral Gain** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 66 | 0...255 | B127<cr> | > ###<cr> | Y |

The upper case **'B'** provides a means to read or set the integral gain associated with the servo control law.

**NOTE:** Command was modified to include Startup Gain in firmware date codes of 31-Mar-08 and later

Startup Gain was introduced as a second parameter of this command to make startup more reliable under very aggresive tuning conditions. This second value is an initial value used during the servo Enable process. It is set to a value that will create a relative soft or "mushy" response to any servo error disturbance.

-o-

| C | Read or Set Velocity Gain | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 67 | 0...255 | C127<cr> | > ###<cr> | Y |

The upper case **'C'** provides a means to read or set the velocity gain associated with the servo control law.

**NOTE:** Command was modified to include Startup Gain in firmware date codes of 31-Mar-08 and later

Startup Gain was introduced as a second parameter of this command to make startup more reliable under very aggresive tuning conditions. This second value is an initial value used during the servo Enable process. It is set to a value that will create a relative soft or "mushy" response to any servo error disturbance.

-o-

| **D** | **Read or Set Slave Amp Trim** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 68 | 0...255 | D127<cr> | > ###<cr> | Y |
| The upper case **'D'** provides a means to read or set the trim compensation for the salve amplifier, if present. | | | | | |

-o-

| **E** | **Read or Set Feedback Offset** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 69 | 0...255 | E127<cr> | > ###<cr> | Y |
| The upper case **'E'** provides a means to read or set the feedback offset associated with the servo control law. | | | | | |

-o-

| **F** | **Read or Set Commanded Position** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 70 | 0...65535 | F127<cr> | > #####<cr> | N |
| The upper case **'F'** provides a means to read or set the commanded position associated with the servo control law. | | | | | |

-o-

| **G** | **Read or Set Slew Gain** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 71 | 0...1023 | G127<cr> | > ###<cr> | Y |
| The upper case **'G'** provides a means to read or set the slew gain associated with the servo control law. A larger value corresponds to a faster rise time. | | | | | |

-o-

| H | **Read or Set Motor Position** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 72 | 0...65535,<br>0...65535,<br>0...255,<br>0...1023 | H5000 60000 3<cr> | > ##### ##### ### #####<cr> | Y |

The upper case **'H'** provides a means to read or set the motor position fault conditions as well as the reading of the motor position. The first value corresponds to the negative limit, the second to the positive limit and the third a position limit count. Note that the first parameter should always be less than the second parameter. The position limit count represents the number of consecutive times that either negative or positive limit can be violated before generating a fault condition. By setting the negative limit value equal to the positive limit value or setting the position limit count to zero (0), position fault limit evaluation will be disabled. The fourth parameter is a representation of actual position as reported by the feedback sensor and is read only.

-o-

| **I** | **Read or Set Motor Current** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 73 | 0...65535,<br>0...65535,<br>0...255,<br>0...1023 | I5000 60000 3<cr> | > ##### ##### ### #####<cr> | Y |
| The upper case **'I'** provides a means to read or set the motor current fault conditions as well as the reading of the actual motor current. The first parameter is the negative current limit and the second the positive current limit. Note that the first parameter should always be less than the second parameter. The third parameter is the fault error count is the number of consecutive times that either negative or positive current limit can be violated before generating a fault condition. By setting the negative current limit equal to the positive current limit or setting the fault error count equal to zero (0), motor over currrent fault evaluation will be disabled. The fourth parameter is a sample of the actual current being applied to the motor and is read only. | | | | | |

-o-

| **J** | **Read or Set Board Temperature** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 74 | 0...1023,<br>0...1023 | J127<cr> | > ####<br>####<cr> | Y |

The upper case **'J'** provides a means to read or set the board temperature fault level as well as read the actual board temperature. The first parameter is the board temperature fault level. The second parameter is the actual board temperature and is read only. Note that only the first parameter is saved. A single count is nearly equivalent to 0.5 degrees Celsius. It is recommended that the board temperature not exceed 50 degrees Celsius or approximately 120 counts.

-o-

| **K** | **Read or Set Motor Temperature Fault Level** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 75 | 0...1023,<br>0...1023 | K512<cr> | >####<br>####<cr> | Y |
| The upper case **'K'** provides a means to read or set the motor temperature fault level as well as read the motor temperature. The first parameter is the motor temperature fault level. The second parameter is the actual motor temperature and is read only. Note that only the first parameter is saved. | | | | | |

-o-

| L | **Read Amplifier Fault State** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 76 | 0...1,<br>0...3 | L<cr> | >$ # #<cr> | N |

The upper case **'L'** provides a means to read if the amplifier has faulted and if so what the cause of the fault was. The first value returned is either zero (0) indicating that the amplifier is not faulted or one (1) in which case the amplifier is faulted. The second value returned is used to communicate what type of condition created the fault where;

- (0) indicates a over current condition as compared to a set of threshold values described in the Motor Current command.
- (1) indicates a motor over temperature condition as compared to a threshold value set in the Motor Temperature command.
- (2) indicates a amplifier over temperature condition as compared to a threshold value set in the Board Temperature command.
- (3) indicates a over travel condition as compared to a set of threshold values described in the Motor Position command.

Note that this mapping of values closely approximates that of Fault1 and Fault2 electrical lines on the Auxiliary Header.

**NOTE** : This command was modified in firmware version date code 12-Sep-08 as follows;

- (0) indicates a no fault condition
- (1) indicates a over current condition as compared to a set of threshold values described in the Motor Current command.
- (2) indicates a motor over temperature condition as compared to a threshold value set in the Motor Temperature command.
- (3) indicates a amplifier over temperature condition as compared to a threshold value set in the Board Temperature command.
- (4) indicates a over travel condition as compared to a set of threshold values described in the Motor Position command.\
- (5) indicates a motor break failure and will be only reported if properly configured.

Note that the previously described Fault1 and Fault2 electrical line functionality was eliminated as part

-o-

| **M** | **Read or Set Controller ID** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 77 | 128...255 | M129<cr> | > ###<cr> | Y |

The upper case **'M'** provides a means to read or set the network address for the currently active controller. Each controller can have a unique address value between 129 and 255. Please refer to the Networking the SCA814 section for further details.

-o-

| | **Read the Product Identification Number (PIN)** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| **N** | 78 | 0..255<br>0..255<br>0..255<br>0..255<br>0..255 | N<cr> | >$ ### ### ### ### ###<cr> | Y |
| The upper case **'N'** provides a means to read the Product Identification Number. The Product Identification Number consists of five values that can each assume a value of between zero (0) and 255. | | | | | |

-o-

| **O** | **Read the Compiler Version Number** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 79 | | A127<cr> | > #.###<cr> | Y |
| The upper case **'O'** provides a means to read the version number of the compiler used to create the firmware within the controller. | | | | | |

-o-

| **P** | **Read the Firmware Date and Time** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 80 | | P<cr> | > ##-##-## ##:##:##<cr> | Y |
| The upper case **'P'** provides a means to read the date and time that the firmware inside the controller was created. | | | | | |

-o-

# Communications Configuration

| Q | Communications Configuration | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 81 | | Q<cr> | ><br>#####<cr> | N |

**NOTE** : This command was introduced in firmware versions with date codes of 24-Apr-07 and later.

The upper case **'Q'** provides a means to set or read the communications cofiguration word. That sixteen-bit word contains various configuration data represented by various bit states. Those bit states are as follows;

| Bit # | Description |
|---|---|
| 0 | Baud Rate where 115200 baud when bit is cleared (0) and 230400 baud when the bit is set (1). |
| 1 | A command message is normally terminated with a Carriage Return. Immediately after the Carriage Return is detected the firmware enters the command parsing phase. To indicate this, a '>' character is sent. When this bit is set (1) this character is not sent. |
| 2 | When a command has been completely processed the Carriage Return original sent is finally reflected back to signify the completion of that command. In addition to the Carriage Return a Line Feed is normally sent. When this bit is set (1) the Line Feed is not sent. |
| 3 | Each time a character is sent to the controller it is normally echoed back as an acknowledgement of its reciept. This is a good method to regulate the host transmission rate and in sure faithful communications. The penalty of this method is that it doubles the transaction time. When this bit is set (1) the echoing feature is disabled. With this bit set great care should be taken to avoid overrunning the controllers single byte character buffer. |
| 4 | In order to not overrun the controllers single byte communications buffer an exaggerated pause needs to be supplied after each character transmission or the controller needs to acknowledge its reciept. One such reciept method is monitor the characters echoed back (bit3=0). When bit three (3) of the communications controfiguration is set (1) then this is not an option and this bit offers an alternative. When this bit is set (1) then IO1 is reconfigured to act as a Clear To Send (CTS). The host can monitor this pin and know that when this pin is set (1) it is OK to send at least one more character. When the CTS pin is cleared it is an indication that the controller is still processing the character or command, that the recieve buffer is full and that further communications should be suspended until the CTS returns to its normally HIGH state. Note that setting this mode will directly conflict with the functionality of the IO1 command. |

It is important to note that this parameter is not saved as many parameters normally are. When the controller starts/restarts this parameter is initialized to zero (0).

-o-

| R | Read or Set the Restart Mode and Source | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 82 | 0...1<br>0..1 | R0<cr> | > #<cr> | Y |

The upper case **'R'** provides a means to read or set the Restart mode of operation. There are two restart modes, auto and manual where auto is equated zero (0) and manual to one (1). In the auto mode the controller will try to restart the servo amplifier once every second until a successful restart condition is encountered. When in manual restart mode the Enable state must be toggled from its high state to a low state and then back to its high state. This toggling process is either done through a TTL electrical line or software depending on the state of the Enable Source.

**NOTE:** Command was modified to include Restart Source in firmware date codes of 29-May-07 and later

The second parameter of this command is used to identify which method will be used to cause the SCA814 Servo Controlled Amplifier to be restarted from a faulted state. Early generations of the SCA814 Servo Controlled Amplifier were equipped with special circuitry for this purpose and to use that method, this parameter should be set to zero (0). Later generations of the SCA814 Servo Controlled Amplifier had the aforementioned circuitry removed to further enhance servo performance and required firmware to assume the restart task. The Restart Source parameter should be set to one (1) to indicate that firmware should be used to restart the servo.

-o-

| **S** | **Read or Set IO1** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 83 | 0...3 | S2<cr> | > #<cr> | Y |

The upper case **'S'** provides a means to read or set the mode and state of IO1 pin. The least significant bit of this parameter is used to communicate the state and the second least significant bit is used to communicate the mode of operation. Note that this command can be in direct conflict with the CTS mode described in the Communications Configuration command. The mode of operation can be interpreted

-o-

| **T** | **Read or Set IO2** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 84 | 0...3 | T1<cr> | > #<cr> | Y |
| The upper case **'T'** provides a means to read or set the mode and state of IO2 pin. The least significant bit of this parameter is used to communicate the state and the second least significant bit is used to communicate the mode of operation. The mode of operation can be interpreted | | | | | |

-o-

| U | Read or Set Profile Data | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 85 | 0...63<br>0...65535<br>0...65535 | U1<cr> | > 1 500 6<cr> | Y |

**NOTE** : This command has been introduced in firmware versions with date codes of 16-Mar-07 and later.

The upper case **'U'** provides a means to read or set profile data. The format of the entered by this command is dependent on the active profile mode. This mode can be determined by reading the Profile Configuration.

When in the Point-to-Point mode the first parameter represent a Point address and can have a range of between zero (0) and 63 inclusively. The second parameter is the Point Position and has a range of zero (0) to 65535 inclusively. The on-board 16-bit DAC will be set to this value when referring to this Point address.

When in the Linear mode the first parameter represents a Slope address and can have a range of between zero () and 31 inclusively. The second parameter is the Step Size and has a range of -32768 to 32767 inclusively. The third parameter is the Step Count and has a range of zero (0) to 65535. The firmware will read the second and third parameters and respond in the following manner; from the current motor position it will add Step Size DAC counts during each tick period for a total of Step Count periods. Thus, if the motor was sitting at 32000 DAC counts and Step Size was set to -100 and Step Count to 10 then the motor position would be decremented by -100 DAC counts every tick period for 10 periods. At the end of 10 periods the motor would be sitting at 31000 DAC Counts and the next Linear address would be read and interpreted. There are two special cases to this intepretation processs that should be discussed here. The first case is when Step Size is set to zero (0). Under this condition a delay of Step Count tick periods is created where the motor position does not change. The second case is when Step Size is non-zero and Step Count is zero (0). In this situation the Step Size is evaluated as an absolute position. The on-board 16-bit DAC is set to a value directly corresponding to Step Size. Under such conditions, the Step Size should have a range of zero (0) to 65535 to correspond to the DAC limits.

-o-

| | Read or Set Profile Characteristics | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| **V** | 86 | 0...65535<br>0...255<br>0...63<br>0...63<br>0...1<br>0...2 | V1<cr> | > <cr> | Y |

**NOTE** : This command has been introduced in firmware versions with date codes of 16-Mar-07 and later.

The upper case **'V'** provides a means to read or set the profile configuration.There are a total of five parameters used to configure the profile. They are as follows;

| Parameter | Description |
|---|---|
| Cycles | A profile cycle is described through a series of consecuative memory cells. This parameter suggests the number of times that series will be scanned. There are two special cycle values that should be highlighted here. Setting cycles to zero (0) will cause the profile generator to terminate any currently active profile generation. Setting this parameter to its maximum value of 65535 will cause the the profile generator to run indefinitly. When set to 65535 the profile generator will also start runniing automatically after power-up or reset condition. |
| Ticks | Number of 25.6 microsecond periods between profile updates. |
| Start | The first and lowest memory address used to descibe a profile cycle. This parameter has a range of zero (0) to 63 when in Mode zero (0) and zero (0) to 32 when in Mode one (1). In both cases this parameter should be less than the Stop parameter. Setting the Start and Stop values equal is effectively equivalent to setting cycles to zero (0). |
| Stop | The last and highest memory address used to descibe a profile cycle. This parameter has a range of zero (0) to 63 when in Mode zero (0) and zero (0) to 32 when in Mode one (1). In both cases this parameter greater than the Start parameter. Setting the Start and Stop values equal is effectively equivalent to setting cycles to zero (0). |
| Mode | There are currently two (2) different ways to describe a profile. These modes are as follows; setting this parameter to zero (0) causes the Mode to be set to Point-to-Point and setting this parameter to one (1) causes the Mode to be set to Linear. A more complete discription of these modes is given in the Profile Data section. |
| Tick Source | **NOTE:** Command was modified to include Tick Source in firmware date codes of 9-Sep-08 and laterThere are currently two (2) different tick sources to select from. Setting the Tick Source to zero (0) will cause the profile generator to use the standard internal hardware timer. Setting this parameter to one (1) will cause the profile generator to use a software command as a source of the clocking. A value of two (2) is reserved for future source. |

-o-

| **V** | **Dump Motor Position and Current** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 86 | | V<cr> | > #####<br>#####<cr> | N |

The upper case **'V'** provides a means to read the first 50 measured position and current levels found after the last Command Position. The first value sent is the measured current and the second the position. Each of the 50 samples reported is separated by a Carriage Return and Linefeed.

**NOTE:** This command has been eliminated in firmware versions with date codes of 21-Feb-07 and later.

-o-

| **W** | **Read or Set Save Mode** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 87 | 0...1 | W1<cr> | > #<cr> | N |
| The upper case **'W'** provides a means to read or set the save mode where setting this parameter to a one will cause all attempts to save to be inhibited. The controller is initializes this parameter to zero (0). | | | | | |

-o-

| X | **Read or Set Encoder Position** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 88 | 0...65535 | X32767<cr> | > #####<cr> | N |

The upper case **'X'** provides a means to read or set the digital quadrature encoder position. Note that the encoder position is initialized to 65535 upon startup of the controller. This command is only applicable to stages equipped with a digital quadrature encoder.

**NOTE:** This command has been eliminated in firmware versions with date codes of 12-Sep-08 and later.

-o-

| **Y** | **Read Encoder Index Position(s)** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 89 | 0...2,<br>0...65535,<br>0...65535 | Y0<cr> | >$ #####<cr> | N |

**NOTE** : This command was removed in firmware versions with date codes of 12-Sep-08 and later.

The upper case **'Y'** provides a means to read the encoder index position(s). There are two edges to the index pulse, the rising edge and the falling edge. The parameter associated with this command is used to describe which of the two edges are to be reported. If a value of one (1) is sent the command will return the rising edge. If a value of two (2) is sent the command will return the falling edge. If any other value or no value at all is sent then this command will return both the rising and falling edges. This command is only applicable to stages equipped with a digital quadrature encoder.

**NOTE:** This command has been eliminated in firmware versions with date codes of 12-Sep-08 and later.

-o-

| **Z** | **Read or Set PSD Gain** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 90 | 0...255 | Z127<cr> | > ###<cr> | Y |
| The upper case **'Z'** provides a means to read or set the PSD gain associated with the servo control law. | | | | | |

-o-

# Command Position using Binary Data

| a | **Set Binary Commanded Position** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 97 | 0...65535 | a%&<cr> | > #####<cr> | N |

**NOTE** : This command was introduced in firmware versions with date codes of 24-Apr-07 and later.

The lower case **'a'** provides a means to set the commanded position associated with the servo control law using binary data as opposed to the more normal ASCII type data. This command requires that exactly two bytes are sent immediately following the 'a' character. These bytes are the binary encoded position where the MSB is sent first, followed by the LSB. As an example if the DAC needs to be set to 9510 (2526 hex) then 'a' will be followed immediately by a '%' (ASCII 25 hex) and then a '&' (ASCII 26 hex). Immediately after receiving the third and final byte of this command the controller will set the command position to the requested level. This command is otherwise identical to the Command Position command.

-o-

# Command Position using Binary Data with Checksum

| **b** | **Set Binary Commanded Position with Checksum** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 98 | 0...65535 | a%&*<cr> | > #####<cr> | N |

**NOTE** : This command was introduced in firmware versions with date codes of 24-Apr-07 and later.

The lower case **'b'** provides a means to set the commanded position associated with the servo control law using binary data as opposed to the more normal ASCII type data. This command requires that exactly three bytes are sent immediately following the 'b' character. The first two bytes are the binary encoded position where the MSB is sent first, followed by the LSB. The third byte is an eight bit check sum of the two byte binary position. As an example if the DAC needs to be set to 9510 (2526 hex) then 'b' will be followed immediately by a '%' (ASCII 25 hex) and then a '&' (ASCII 26 hex). Finally, since 0x25 + 0x26 = 0x4B then a 'K' (ASCII 46 hex) will be sent. When the controller receives the two byte position it will frist add their values together and compare it to the eight bit checksum. If the sum does not equal the checksum then the controller will not set the command position. This command is otherwise identical to the Command Position command.

-o-

# Configuration Byte

| **C** | **Read or Set Configuration Byte** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 99 | 0...255 | c127<cr> | > ###<cr> | Y |

The lower case **'c'** provides a means to read or set the configuration byte. The configuration byte uses various bits within its byte to annotate various configuration information. That information is represented as follows;

- bit 0: Brake Fault Check - Setting this bit to one (1) will cause the controller to continuously monitor the state of the Brake and cause a fault should the brake indicate its engaged. Setting this bit to zero (0) will turn this feature off.

-o-

# Dump Parameters and Profile

| **d** | **Read or Set Enable State** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 107 | 0...2 | d0<cr> | >$ #<cr> | N |

**NOTE** : This command was introduced in firmware versions with date codes of 09-Sep-08 and later.

The lower case **'d'** provides a means to dump the configuration parameters or profile data depending on the mode selected. The optional parameter associated with this command determines the dump mode and can assume a value betwwen zero (0) and two (2) inclusively. Mode zero (0), the power-up default, will cause the SCA814 to announce the values with both read-only and read-write parameters. Mode one (1), "d 1", will cause the SCA814 to send only the writable parameters. This mode is particularily useful if the infomation is to be restored at a later date. Mode two (2), "d2", will cuase the controller to send the entire profile memory in a format consistant with the profile configuration currently selected. Sending a "d" command without a parameter will cause the controller to use the mode specified most recently. For a more detailed discussion of this feature refer to Backing-Up the SCA814 through HyperTerminal and Restoring the SCA814 through HyperTerminal.

-o-

| **k** | **Read or Set Enable State** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 107 | 0...3 | A0<cr> | >$ #<cr> | Y |

The lower case **'k'** provides a means to read or set the enable state associated with the servo control law. Note that this command will only have effect if the Enable Source is set to software as opposed to hardware. Setting this parameter to one (1) will cause the amplifier to be enabled and conversely setting it to zero (0) will disable it.

**NOTE:** Command was modified to save the state in firmware date codes of 9-Sep-08 and later

Setting bit 1, zero based, will cause the enable state to be saved in non-volatile memory such that when powered-up the controller be set to the appropriate state. As an example, setting 'k' to 3 will cause the enable state to be saved as enabled and therefor enabled during power-up.

-o-

| **m** | **Read or Set Enable Source** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 109 | 0...1 | m0<cr> | >$ #<cr> | Y |

The upper case **'m'** provides a means to read or set the enable source associated with the servo control law. When this parameter is set to a zero (0) the electrical input on the Auxiliary Header is used. When this parameter us set to a one (1) the enable state is set through the Enable command.

-o-

| **n** | **Read or Set Tuning Profile Parameters** | | | | |
|---|---|---|---|---|---|
| | ASCII | Data | Example | Response | AutoSave |
| | 110 | 0...65535,<br>0...65535,<br>0...65535 | n127 32000 33000<cr> | >$ ### ##### #####<cr> | Y |

The upper case **'n'** provides a means to read or set the parameters associated with the tuning profile. The first parameter is the count-down delay time before the commanded position is updated. The second and third parameters are the two endpoint's that will be alternately commanded after count-down time.

**NOTE:** This command has been eliminated in firmware versions with date codes of 16-Mar-07 and later.

-o-

# Index

## - A -



## - B -



## - C -



## - D -



## - E -



## - F -



## - I -

# - M -



# - N -



# - P -



# - R -



# - S -



# - T -



# - V -

# - W -