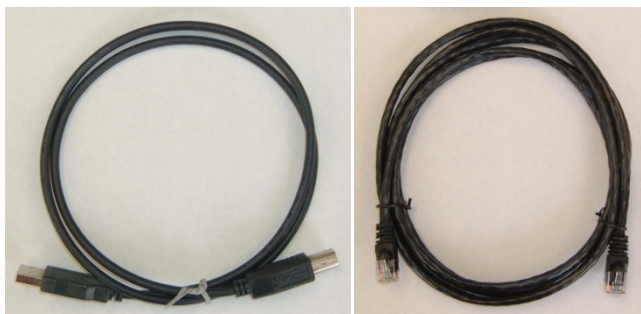


***SIN-11-USB
USERS GUIDE***

Revision Date: 05/06/2010

Overview 3
Getting Started 4
Hardware 6
Demo Power-up 7
Axis “Name” Assignment and Starting Party Line Mode. 7
Timeouts 10
Local Commands 10
Special Escape Commands 12
Error Indicators 13
Line Buffer 13



SIN-11-USB, USB cord and BLC-51-7 are all included when ordering the SIN-11-USB. Also included is a termination plug (not shown) that is inserted into the serial out of the last axis connected

Overview

The SIN-11-USB is an intelligent serial converter/buffer designed to interface a customer's host computer to a step motor subsystem consisting of AMS's motor controller products. This device simplifies application software and improves performance by allowing commands and protocol macros (SIN-11-USB "local commands") to be sent in blocks (complete lines). The features of the SIN-11-USB are very similar to those of the SIN-11 with the difference that it features a USB port instead of an RS-232 port.

SIN-11-USB features:

- Performs all necessary handshake operations
- Diagnostic LED's
- Powered from motor controllers
- Scans for one to 64 axes
- Built in macros such as; "wait while any axis moving"
- Simple to use
- 128 character host buffer allows long command sequences
- USB connection for host computer
- RJ-45 (RS-422) party line connection to AMS motor controllers

Note: The terms "AXIS," "CONTROLLER," and "MOTOR CONTROLLER" are synonymous in this manual.

With the complexity of today's computer operating systems, and the security built into the operating systems shell, it is becoming increasingly difficult for programmers to perform the direct interface with hardware such as UARTs (Windows NT/2000 is difficult, for example).

The SIN-11-USB solves that problem because it is ASCII “line” driven, making it simple to send one, or a block of commands using one text line - there is no need for special echoed character handshake software.

Getting Started

These steps will let you start communications with any AMS “Party Line-RS-422” product. Some of these instructions will be redundant with the more detailed information that follows in this manual.

Only a single axis is required but up to 32 axes can be connected together from one serial port. Most AMS products are supplied with, or can be ordered with, an RS-422 interface. RS-232 models are not compatible with the SIN-11-USB.

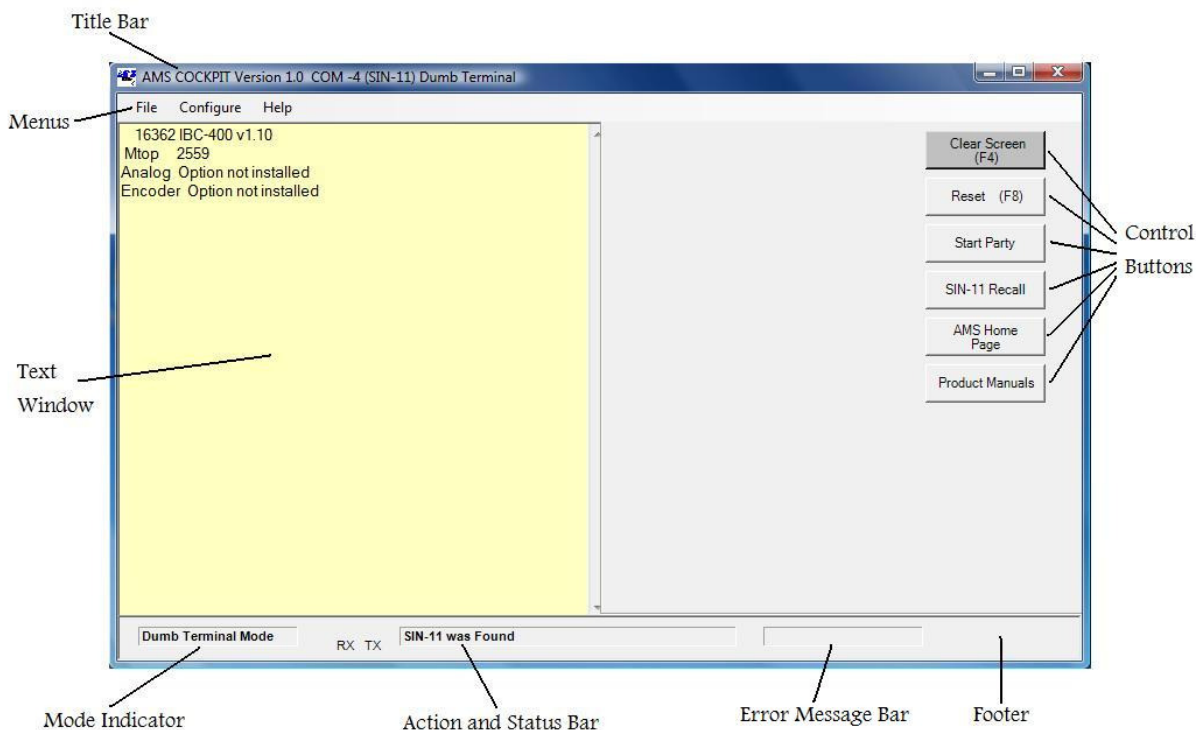
Equipment Setup (quantity one each)

- SIN-11-USB
- USB cord (supplied with SIN-11-USB)
- CAT-5 data (RJ-45) cable (supplied with SIN-11-USB)
- AMS party line controller product, i.e. IBC-400, mSTEP407, MAX-410, DCB-241, etc.
- Computer with USB port

Communication Software

For these tests only a basic terminal program is necessary, such as:

- “AMS Cockpit” software (can be downloaded for free from the AMS web site).
- Microsoft HyperTerminal, supplied with Windows for operating systems up to Windows XP.
- A third party terminal program.



Main Screen of AMS Cockpit Software

Driver Installation

1. Download the zip file comprising the USB drivers from the AMS web site (www.stepcontrol.com) to your computer and save the file on your computer. The zip file can be accessed either via the Sin-11-USB page in the “Products” Section or the “Tech Documents” page.
2. Extract all files from the driver zip file at any location of your choice. In doing so the directory substructure will be recreated. This is important for the installation to be performed successfully.
3. Start the driver installation executable that matches your Windows operating system. The driver needs to be installed in order to operate the SIN-11-USB.
4. Connect the SIN-11-USB to a USB port of your computer using the cord provided with the unit.
5. The red LED (SUSPEND mode) should light up after a few seconds. This means that your computer has recognized the USB device. If the red LED does not light up this means that your computer has not recognized the SIN-11-USB device and it is in SUSPEND mode. You will not be able to send data through it in this case. Please review the driver installation and call our tech support at (603) 882 1447 if needed.
6. Open the Windows Device Manager (The path to get there depends on your operating system. For example, in Vista you will find it directly in the Control Panel. In XP select “System” in the Control Panel. Then select the Hardware Tab. You should now see the Device Manager button to start it up). Expand the category “Ports”. You should see the SIN-11-USB listed along with the port number your computer has assigned to the SIN-11-USB. It is important when you later interface with the SIN-11-USB that you make sure your terminal program is set to this port number.

Connection/Power-on

Always start with only one controller attached, as they must be named first.

1. Ensure the AMS controller power is off.
2. Connect the RJ-45 cable of the SIN-11-USB to the serial INPUT connector of the AMS controller.
3. Ensure the driver software for the SIN-11-USB has been installed (see previous section).
4. Connect the USB port end of the SIN-11-USB cable to the computer if not connected already. The Red LED should turn on after a few seconds. It indicates that the device is not in SUSPEND mode. If in SUSPEND mode, the LED is off. In case the red LED does not turn on, your computer does not recognize the device. The red LED must be on in order to proceed – no communication is possible if the device is in SUSPEND mode. Please see the section Driver Installation.
5. A motor may be attached. Set the current to a low value if required.

Never connect or disconnect any motor while power is on!

6. Start the terminal program. Whichever terminal program you are using, make sure that you have the port set to the one which your computer has assigned to the Sin-11-USB (See point 6 in the Section Driver Installation).
7. The communication parameters need to be set to 9600, 8, N, 1 (9600 baud, 8 data, no parity, 1 stop bit).
8. Turn the controller power on. The SIN-11-USB is powered via the party line cable (with exception of the USB interface circuitry) and should “sign-on” as soon as controller power is applied. Some LED’s should blink. If this does not occur check:
 - a. That the correct serial port is chosen. Conflicts can arise from internal devices such as modems.
 - b. That the RJ-45 cable is connected to the serial input (when LED’s don’t flash).

Note that if you are using AMS Cockpit, the software will identify the device as a “SIN-11”. AMS Cockpit does not differentiate between the SIN-11 and the SIN-11-USB as there is no need for this.

Note: If you plan to disconnect the SIN-11-USB from your computer, you are advised to first close AMS Cockpit or the Terminal application that you are using before physically unplugging the SIN-11-USB. Otherwise the application will continue to search for the SIN-11-USB and in the worst case needs to be stopped via the close button on the upper right hand corner.

Controller Sign-on (note, various LED's flash to indicate serial data activity)

1. Hit the "spacebar" key as the first key after power on.
2. The motor controller should sign-on with the product model and software version.
3. Check the name assignment by entering "X<cr>" (Examine command). The name will be the end character of the first parameter line. It is usually "A." If it is not a readable character or you want a different name, follow the "Axis Name Assignment" procedure described further on in this manual.
4. Verify commands using a command such as "M 1000" to move and "M0" to stop.

Now is a good time to setup motor speeds, currents and other parameters for your application.

Party Line Start-up

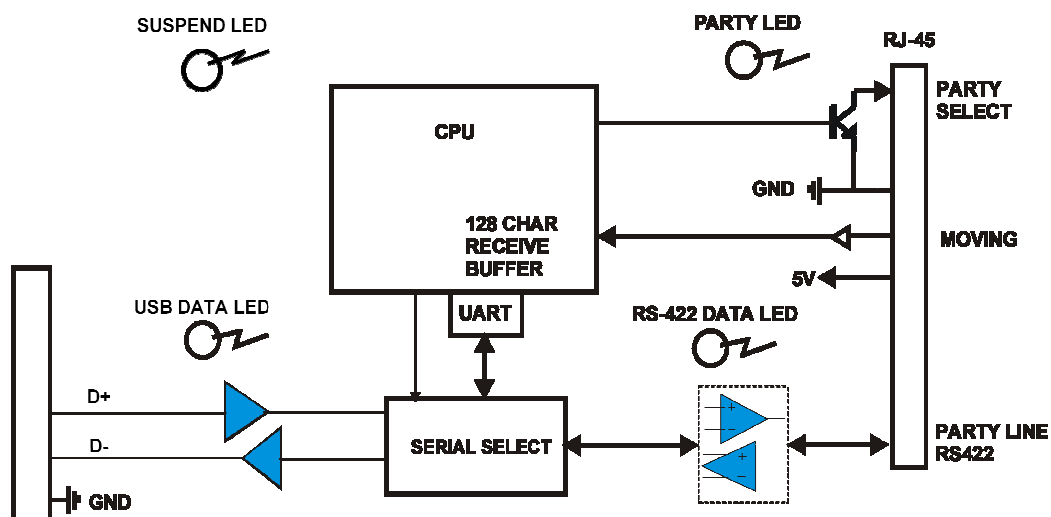
If more than one axis is to be used, each must have a different name assigned.

1. Enter the command "&" The SIN-11-USB:
 - a. Turns on the party line mode.
 - b. Resets the AMS controllers (^C).
 - c. Scans the RS-422 bus for axis names between A and z and responds with the name(s).
 - d. Map's the axis(s) within the SIN-11-USB RAM.
 - e. Prepares to receive command lines.
 - f. Keeps the "Party" LED on.

If the party line scan fails, the Party LED will be turned off, and an error condition indicated. In party line mode the character echo is set off. Computers like this, but humans need feedback.

2. Enter the command line "A+1000;AW0;AZ."

Hardware



SIN-11-USB Block Diagram

The motor should index 1000 steps (provided the parameter setup is correct), waits until motion is complete, and then reports the position (preceded by the axis name).

Flashing LED's

Two LED's flashing indicate that some error was detected during the previous command. The command may be from an invalid axis name, time out, or other condition. Fear not; the flashing in no way prevents subsequent commands and the flashing will stop with the next valid command.

The SIN-11-USB is attached to the host computer via a free USB port. Input to the first axis is through an RJ-45 connector. The SIN-11-USB obtains its 5-volt power from the INPUT serial connector of the first controller on the party line bus.

LEDs

Several LEDs are provided for diagnostic and status purposes:

- “Party” (Orange) indicates "party line" mode.
- “USB” (Yellow) indicates activity on the USB bus.
- “RS-422” (Yellow) indicates serial RS-422 controller activity.
- “SUSPEND” (Red) off if in SUSPEND mode

Demo Power-up

The SIN-11-USB will operate in one of two distinct modes. The mode determines how the controller (party line) bus operates.

- In the SINGLE mode only a single controller may be connected to the party line bus. Only in this mode can a controller's address be assigned and saved. Pin 8 ("party") of the RJ-45 connector is not asserted.
- In the PARTY mode, multiple controllers may be addressed on the party line bus. Pin 8 of the RJ-45 connector is pulled low and a ^P has been sent for the appropriate models.

Note that parameters, programs, and motion commands may be sent to a connected controller in either mode.

The SIN-11-USB is powered with 5 volts from the first controller's INPUT serial communication connector. It should be noted that although the length for RS-422/485 can be very long, many cables have small conductor sizes and the voltage drop may cause problems.

When power is applied to the SIN-11-USB, the following initialization takes place:

1. The SIN-11-USB is initialized.
2. A "sign-on" message for the SIN-11-USB is sent to the host terminal.
3. Strike the space bar, the controller signs on.

The power-up mode is always single (dumb terminal). Here, under Windows, you can use the hyper terminal or any other dumb terminal application. The terminal should allow control characters to be transmitted.

The single mode is a “straight through” connect. All received characters (except the ‘&’ character) pass to the RS-422 bus and received characters are returned to the terminal. This mode is useful for single axis debug, diagnostics, and assigning the axis name, which will be required for party line mode. Programs may be typed into a controller's NV memory in the single mode.

Axis “Name” Assignment and Starting Party Line Mode.

The following discussion will assume that a standard PC (or clone) is being used. For this discussion, we will use the character "A" as the axis name.

Refer to the specific AMS product manual for a detailed description this procedure.

1. Connect one axis only to the SIN-11-USB.
2. Start the dumb terminal program.
3. Apply power to the controller (or if power is already on, cycle the power to the controller). You should see a sign-on message from the SIN-11-USB.

There are now two methods to assign names, the “software” method and “hardware” method

Software Method

This is a new method using control characters as commands to assign names and initiate party-line operation. The advantage is that the hardware “switch” is no longer necessary. These features will be introduced into product as revisions are implemented. The SIN-11-USB will accommodate either protocol and any mix of old and new products. It operates as follows

1. Enter Ctrl N (^N) <cr> to assign the name. When the “name?” prompt is displayed, key in the desired name character (A-z) and immediately that character will be stored in memory.
CAUTION – Never do this when more than one controller is attached or all names in all controllers will become the same.
CAUTION – While the controller can accept other characters, including non pre-defined control characters, punctuation, numbers, etc, the SIN-11-USB only scans the 64 possible names between ‘A’ and ‘z.’
2. When the name is captured, the controller will restart. Use a space character to sign-on again.

Hardware Method

1. After power on (reset), type the desired name character (A for our example), followed by the space key. The attached controller will “sign-on.” with "x v2.05<cr><lf>" or similar.
2. Enter the "Examine" command: X<cr>. The axis name will be displayed at the end of line one.
3. Type S<cr> to store the name in non-volatile memory. The name is now saved.
4. Place the axis in party line mode (one axis connected):
 - a. Enter: &<cr> (scan 64 axes)
 - b. Test the setup by typing AZ<cr>, a request for controller A's position counter. The response should be: "Z 0<cr><lf>."
 - c. Enter a command: A+1000<cr> (axis ‘A’ should move 1000 steps).

Choosing Names

Any ASCII letter between A and Z (upper and/or lower case) can be used. For example A, Y, & z are all valid, because the "&" scan command starts with upper case ‘A’ and scans through all upper case characters and then all lower case characters.

As the SIN-11-USB finds controllers, it displays the axis name. The scan can take several seconds. During this procedure the “Party” (party line) LED will be on.

Multiple Axes Connected in Party Line

One to 32 axes (or more, dependant on cable lengths and type) may be addressed by one SIN-11-USB. Each axis must have a **unique** pre-assigned name stored in its NV memory. If you try to address an axis that does not currently exist, the returned string will contain a “?” and the unknown name character.

Party Line Command Syntax

Commands are sent to the party line controller bus in the form of a single string of characters (up to 127), terminated by a line-feed <lf> or carriage return <cr>. The input line may be edited using backspaces. On receipt of the terminator (<cr> is generally used) the buffered line is executed. On completion, it returns <cr><lf>, even if echo is disabled.

Some general rules to follow:

1. Only use spaces between 2 data numbers, i.e., XK10 20<cr>.
2. Enable echo of input only for debug purpose.

3. Send one complete line, then wait for cr/lf.
4. Scan each returned line for error indicators.

At this point, the “host” knows (if there are no errors) that the command(s) have been sent and received. However, unless the command string has included a "wait" command (such as &W*) as the last command, individual axes may still be moving. See "&W" under Local Commands.

Multiple commands can be placed on a line.

Command string example

```
X+1000; Y-2000; Z+8000<CR>
```

On receipt of this string, all three axes (X, Y, Z) will initiate the specified index motion. The host must wait until the command string has been processed, as indicated by the response of line end <cr><lf>.

Note, multiple commands are separated by semicolons (;). Spaces should not be used except between the numbers of commands that use two parameters, such as "K10 5."

Spaces are allowed before the delimiter (;), but only take up unnecessary buffer space. The maximum length of a single “axis” command is limited to 14 characters.

Commands that return result values send the axis “name” followed by the data as they are executed:

```
O0      (set origin of axis X to zero)
+1000   (axis to index 1000 steps)
&W      (SIN-11-USB wait until stopped)
Z       (read back axis position)
```

Example for axis X:

```
Host sends:  XO0;X+1000;&WX;XZ<cr>
Returns:     X1000;<cr><lf>
```

Attach three axes, named X, Y, and Z. Index all three with one command line, then wait until all are stopped.

```
X+1000; Y-2000; Z+8000;&W* ;XZ;YZ;ZZ<CR>
X 1000Y-2000Z8000<cr><lf>
```

The Global Command

Identical commands may be sent to all axes registered in the SIN-11-USB memory. The SIN-11-USB will send the same command to every available axis at the highest possible rate.

```
*+1000;&W* <cr>
```

This command will index all attached axes by 1000 steps, then wait for completion before sending the echo line.

Special keys will produce unique actions:

1. **^C** (control C) acts as a reset. The ^C is sent to reset any axis on the party line and the SIN-11-USB itself.
2. **Esc** (escape) aborts any in-process command.
3. **Backspace** (←) edits input lines in a conventional fashion.
4. **&** preceding a command signals a “local” SIN-11-USB command.

Timeouts

The SIN-11-USB uses a timer as a method to prevent “hang-ups” from axes that are busy or crashed. A timeout will occur (with certain exceptions) when:

1. Characters are not echoed within 150 milliseconds.
2. Acknowledge delay exceeds two seconds.

&A0 Mode

A most common cause for timeouts would be more than one consecutive “index” command to the same axis, without performing a “wait until stopped” function, i.e.,

```
X+8000;X-8000<cr>      (timeout is generated)
X+8000;XW0;X-8000     (timeout is not generated)
```

When a timeout occurs, an idle sequence (<lf>) to the offending axis is sent and a character (# or !) is echoed. Some products will echo a dollar (\$) character when busy, while others will not echo anything until the command has been completed. The programmer must be aware of these situations. The SIN-11-USB will automatically use the \$ echo with the "&W" command.

Note: The timeout function is disabled during the W command. Timeouts are completely disabled when in the &A1 mode (power up default).

Local Commands

Special commands can be executed by the SIN-11-USB. These commands set the mode, enable or disable echoes and perform functions such as scan for axes, wait while moving, etc.

Each motor axis is assigned a single character name, between “A-Z” or “a-z,” whereas the SIN-11-USB has a predefined name of “&.” The following commands are executed within the SIN-11-USB:

&A (Timeout Control)

This command defines the behavior when an axis is busy. A busy condition occurs when a command such as the "GO" command has not fully completed. If the controller processor is other than an SMC-C24 (see AMS Product Selection/Data Rate Guide at the end of this manual for control processor families), the axis emits a \$ in place of the normal “name” character. For the SMC-C24 the axis will not echo the character when “busy” and the SIN-11-USB must wait as long as required for the executing command to finish.

&A1

This command will enable the automatic “wait while busy” mode. When in this mode, commands to any axis are tested for the “busy” condition. When a busy condition is found, the SIN-11-USB will wait until the busy condition has finished, then execute the specified command. This is the default at power up (&A1).

&A0

Turning this mode off will result in the skipping of any “busy” axis. The response line will contain a \$ symbol. To determine if the SMC-C24 controller is busy, the timeout is used to determine that the controller is “busy.” The special case of the W command will temporarily force the “wait while busy” (disable timeouts) mode.

Not all AMS products feature the “\$” indicator. Products with the SMC-C24 based microprocessor require a timeout as a busy indicator, and will be slower to respond.

The “abort” (ESC) or reset (^C) character will break any busy polling “hang-up” and clear any commands in process or pending.

&D (Set Long Delay Time- 1-255)

This delay (default 150) specifies the delay to be used after a reset. The default of 1.5 seconds is adequate for most products. A symptom of a delay time too short is a missing axis name during the party line scan.

&E (Echo Mode Command)

This command permits enable or disable of characters sent to the PC. The number following the E represents a binary “map” of enable bits, i.e., bits 1 and 2 would be turned on with the E3 value. The most useful debug mode is &E5 where characters are echoed as they are typed (with editing backspace), and as result data is generated.

&E1<cr> (Echo Input Characters)

Characters received from the PC are echoed as they are received. In dumb terminal mode this may result in a double printout of each character.

In party line mode the typed characters will not be echoed unless this is enabled. The echo must be disabled when the computer program is rapidly sending strings.

&E2<cr> (Echo Commands)

Echo commands as they are executed (for debug).

&E4<cr> (Echo Results, Enabled after &P(n) command)

This function is enabled in the party line mode and will echo the values returned from commands that have a result (such as "z," the position counter request). This is the default at power-up.

&E5<cr> Combination E1 and E4(most useful for debug)

Input characters and output result strings are echoed (in party line).

& (Start Party Line Operation)

The following actions take place:

1. A logic low is placed on pin 8, the party line enable pin (Party LED on).
2. ^P is sent for later models.
3. Dumb terminal mode is OFF.
4. Echo input is OFF.
5. Echo results is ON.
6. An axis scan is done (takes several seconds).
7. <cr><lf> all commands echo a CR - LF sequence.

If the operation is unsuccessful (no axis is found), the single mode is forced, and a “?” character is echoed.

During an axis scan all possible names (A – Z and a – z) are tested for existence. Those that exist are recorded in memory and used for subsequent command error tests.

During the axis scan names are printed as found. Once this is completed a <CR> is emitted. The SIN-11-USB now has a list of the axis names that are used for error checking and global (name = *) commands.

For debug, a subsequent &E5 command will turn on full echo. Using the full echo with a computer batch load will generally cause UART over-run and/or slow operation.

&R n (Repeat Command String)

This command will cause the command string to repeat execution “n”+1 times, starting from the beginning of the line. By way of example:

```
X+800;Y+800;&W*;&R9<cr>
```

Axes X and Y will index 800 steps, a total of 10 times (repeat =9 + first execution), to the position of 8000. Specify n as 255 and the repeat will continue indefinitely, stoppable with the ESC character.

&V (“What Version” Command)

This returns a number specifying the SIN-11-USB software version and is useful for determining software compatibility.

&W (“Wait Until Motion Stopped” Commands)

&W<cr> is a hardware wait command that tests the moving signal on pin 1 of the RJ45 connector. It is very fast but not the most reliable method, due to possible "race" conditions.

Note: Not all products implement a hardware moving signal. And because the moving signal is “wire-or’ed” it is not possible to determine which axis is moving.

&W*<cr> waits for all listed (scanned) axes to finish moving by polling the status of all listed axes (and is thus completely accurate, but slower).

&W(axis name)<cr> waits for the named axis to finish moving. This command should be executed prior to another motion command on the same axis:

```
X+1000;&WX;X-1000<cr>
```

Special Escape Commands**Ping Command**

The single control character ^E (05d ♣) can be sent to the SIN-11-USB to determine its status. This character will immediately echo a response character allowing determination of the SIN-11-USB state as follows:

1. No echo: The SIN-11-USB is not connected or power is missing.
2. Echo ^K (011d ♂): The SIN-11-USB is present.

The response is almost instantaneous. These characters are never sent to the party line bus.

Control C (^C) (Reset Command)

The ^C character is reacted to immediately, regardless of other operation. The following events take place.

1. All character buffers are cleared.
2. Any looping (such as while busy) is terminated.
3. The global reset ^C command is sent to all connected axes.
4. The "loop if busy" &A1 mode is enabled.

These actions also take place at power-up. The party mode switch may be changed prior to asserting the ^C command.

The serial data rate switches are sampled only at power-up and not as a result of the control-c command.

ESC (Abort Command)

Like the ^C command, this one character command is immediate. This command aborts actions but does not trigger a “reset.” On receipt of the ESC command the following actions take place:

1. All character buffers are cleared.
2. Any looping (such as while busy) is terminated.
3. The global abort ESC command is sent to all connected axes.

Any motion controller that is moving or running sequences (triggered by a “GO” input) is halted. Command buffers are cleared and new command entry may be resumed.

Error Indicators

When an error is detected, two LED’s will start blinking (such as an invalid axis request). Subsequent operation is not affected. The host is not able to poll this error, but should have determined the cause during its error check procedure.

Error causes:

1. Communication timeout.
2. Invalid axis name.
3. Bogus command character.

The indicator is reset at the beginning of each command line, thus fast, repetitive command lines may not allow time for “blinking” to become apparent.

Note: A command sent to an axis, whose "name" has not been recorded by the SIN-11-USB during its address scan, will cause the characters: "?<name>#<cr><lf>" to be returned to the host.

If an address scan (e.g. "&") detects NO valid addresses, the string: "?<cr><lf>" is returned, and the "Party" LED will be unlit.

Line Buffer

The SIN-11-USB can receive one command line then must wait for the line execution to complete. If the receive buffer becomes full (127 characters), no more characters will be accepted and exclamation points ‘!’ are returned, regardless of the echo settings. During execution, the line buffer is spooled to the controller(s).