SIN-11-B
USERS GUIDE

Revision Date: 4/14/2011

Overview

The SIN-11-B, is one of Advanced Micro System's latest developments, offering you the exciting possibility of communicating with your AMS control hardware via Bluetooth wireless technology without the need for a hardwired connection. Advantages include:

- No wired connection is required between the AMS controller and the host computer enabling simpler and tidier wiring.
- Up to a 300ft wireless range in an open and unrestricted space.
- All functionality of the AMS product is maintained, including Party Line configuration to be able to control multi-axis wirelessly using only a single SIN-11-B.
- Change your host computer without the need for any wiring or rewiring.
- Utlizing bluetooth protocol, you are able to utilize your computers wifi connection without interruption.
- Password protected connection inhibits unauthorized access to the SIN-11-B.

These are just a few of the advantages of AMS' exciting new wireless motion controller technology!

The SIN-11-B is an intelligent serial converter/buffer designed to interface a 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-B "local commands") to be sent in blocks (complete lines). The features of the SIN-11-B are very similar to those of the SIN-11 with the difference that signals are transmitted via an air link rather than a cable.

SIN-11-B features:

- Wireless communication between host and AMS controllers
- Performs all necessary handshake operations
- Diagnostic LED's
- Powered from motor controllers – no separate power supply required
- 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
- 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-B 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.

Included in the Box:

The product shipment includes:

- The actual Bluetooth adapter model labeled SIN-11-B
- A Bluetooth dongle that can be plugged into a USB port of a computer
- A 6" cord (BLC-51-0.5) to connect the SIN-11-B to the serial IN of your first axis
- A plug (TERM-2) to terminate the serial OUT of the last axis of your system

Hardware Set-Up

Step 1 Set-up the AMS controller based product as described in the respective product manual.

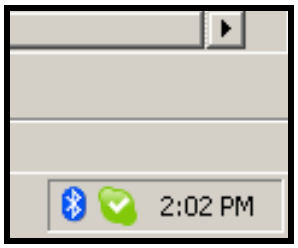
Step 2 Install a terminator plug (TERM-2) into J2 "Party Line Serial Output" of the last axis. Note that all axes need to have unique names. In case the name assignment has not been performed yet, use only one controller based product at this time. Regarding the termination plug, the controller will work without it, however reliability may be compromised, based on the environment (noise) and length of the interconnect cables.

Step 3 Connect the SIN-11-B to the serial in of the first axis using the CAT5E cable provided with the SIN-11-B (or any other CAT5E cable).

Step 4 Power up the AMS controller product or products

Bluetooth Driver Installation

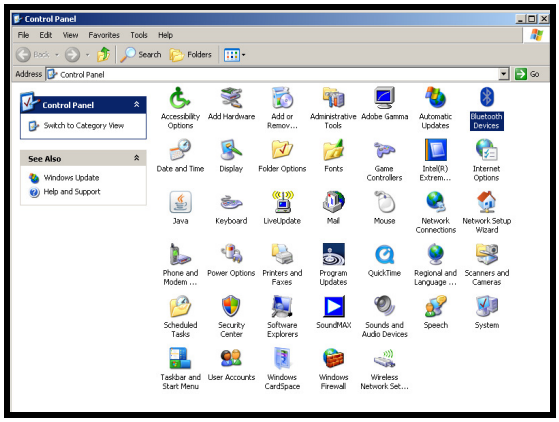
Step 5 Insert the Bluetooth Dongle into your computer. In case Windows does not install the driver automatically, install the driver that can be downloaded from the AMS web page (www.stepcontrol.com). Please contact us at (603) 882 1447 in case you need support.



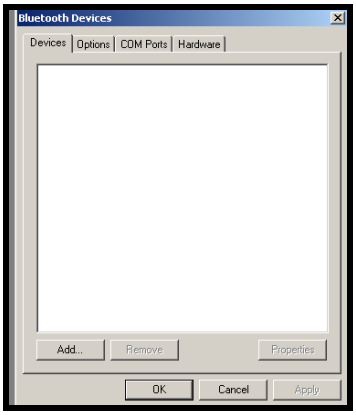
**The Bluetooth symbol will appear in your system tray if windows automatically recognizes your dongle

Adding the Bluetooth Device

Step 6 Double click icon "Bluetooth Devices" or find this in your control panel



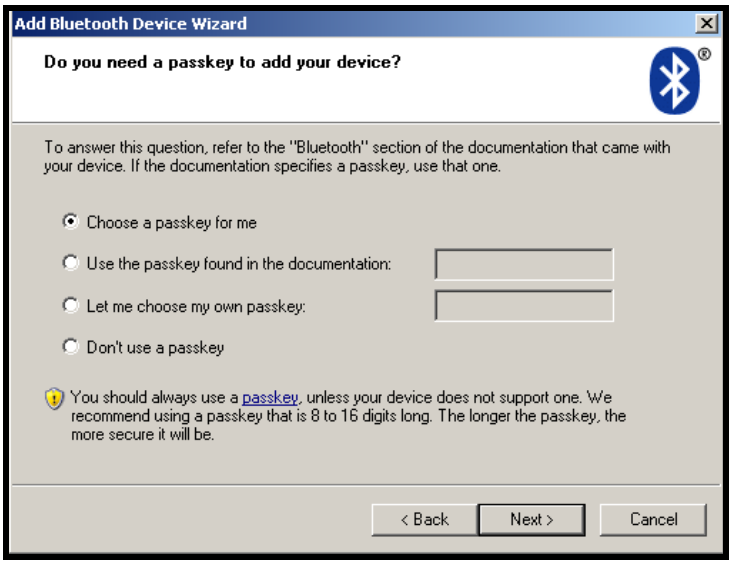
Step7 Click the "Add" button to add your device



Step 8 Select the "My device is set up and ready to be found." ****Be sure to have the AMS product you are using powered up and connected to the Bluebird.**

.

Step 9 Select "Use the passkey found in the documentation." The default passkey is **1234**.



Step 10 Once the device is added you will get a prompt stating it is successfully connected to your computer. ****Be sure to record the outgoing COM port that has been assigned to your device, as you will need this for using AMS Cockpit or any other COM port software you will be using.**

Communication Start-Up

To communicate with the SIN-11-B/Controller you will need a terminal program on your computer. AMS provides the AMS Cockpit software for Windows platforms free of charge that can be downloaded from the AMS web site. However, you may use other terminal programs such as HyperTerminal. There are also terminal programs available for other platforms such as MAC OS. See the AMS web site for more information.

Step 11 Once you have selected and - if needed - installed the communication software, make sure that the active COM port matches the assigned COM port of the SIN-11-B (see section "Adding the Bluetooth Device".

Step 12 Ensure that the port parameters are selected as follows: bits/s: 9600, data bits 8, parity: none, stop bits:1, flow control: hardware.

Sign-On (Dumb Terminal Mode; only one axis connected)

Step 13 Ensure that the SIN-11-B/Controller is powered up.

Assuming you have only one axis connected, you can work in the "Dumb" communications mode. This mode will always be the default after powering up – no matter which terminal program you are using. Strike the SPACE BAR key. The controller should sign on with a message including "Advanced Micro Systems, Inc.". If not, enter a (^C) (reset) and strike the SPACE BAR key again. Striking the ENTER <CR> key should result in an echo of "# " characters, further indicating communication is established.

In case you have multiple axes connected and each has been assigned a unique axis name, you cannot use the "Dumb" communications mode but will need to switch to Party Line mode. In order to achieve this, please issue "&". The SIN-11-B will perform the following actions:

a. Turn on the party line mode.
b. Reset 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-B RAM.
e. Prepares to receive command lines.
f. Turns the "Party" LED on (orange LED on SIN-11-B).

All axes that the SIN-11-B identified will be listed. Note that when you issue commands in Party Line mode every command needs to be preceded by the name of the axis it is intended for. For example if you would like axis B to execute an M1000 command, you need to issue: "BM1000".

You can now issue commands through the wireless interface to your AMS controller products(s).

Communication Modes

There are two methods (protocols) used to send and receive command and data from an AMS controller (axis):

1. "Dumb" Communications Mode

This is accomplished by connecting one single axis to the computer. Commands can be typed in and the controller will execute them. The designer can also enter program sequences into the NV memory and execute them. Virtually every capability can be explored. It is a "human friendly" interface and **NEVER** a computer controlled operation.

At start-up:

1. Hit the SPACE BAR key to sign on.

In Dumb communications mode, you can do a number of useful things:

- Assign "name" character (not necessary if using daisy chain). The dumb terminal mode must be used for name assignment – it cannot be done in Party Line mode.
- Tweaking speed and acceleration parameters
- Experimenting with commands
- Development of program sequences
- Storing motion sequences for non-hosted applications

Note: Single axis mode should never be used in a computer or PLC hosted applications. Single axis functions are suited for programming using the keyboard with visual screen "feedback."

2. Party Line Mode

Party line mode is intended for computer-controlled designs. A computer (usually a PC) can address one or more axes. All axes are interconnected in a chain configuration using "mini drop" network implemented with CAT-5 network cable.

Between 1 and 32 axis are configured as "slaves." Unlike the "Dumb" mode, a proper character by character echoed protocol is necessary for proper operation.

At start-up:

1. Issue an "&" command to enter Party Line mode.

Note that in Party Line mode every command issued needs to be preceded by the name of the axis which is intended to process the command. For example, if the user intends to issue the command M1000 to axis A, the following command line needs to be issued: "AM1000".

Communications Software

AMS offers the "AMS Cockpit" software that can be downloaded for free from the AMS website to assist customers in the implementation of their projects. It is compatible with Windows Operating systems. In addition to enabling communication, it includes some customized functionality, such as the downloading of programs into the non-volatile memory of the controller. This code is not intended to operate as an end user application program, but rather to allow familiarization, evaluation and programming of the AMS products.

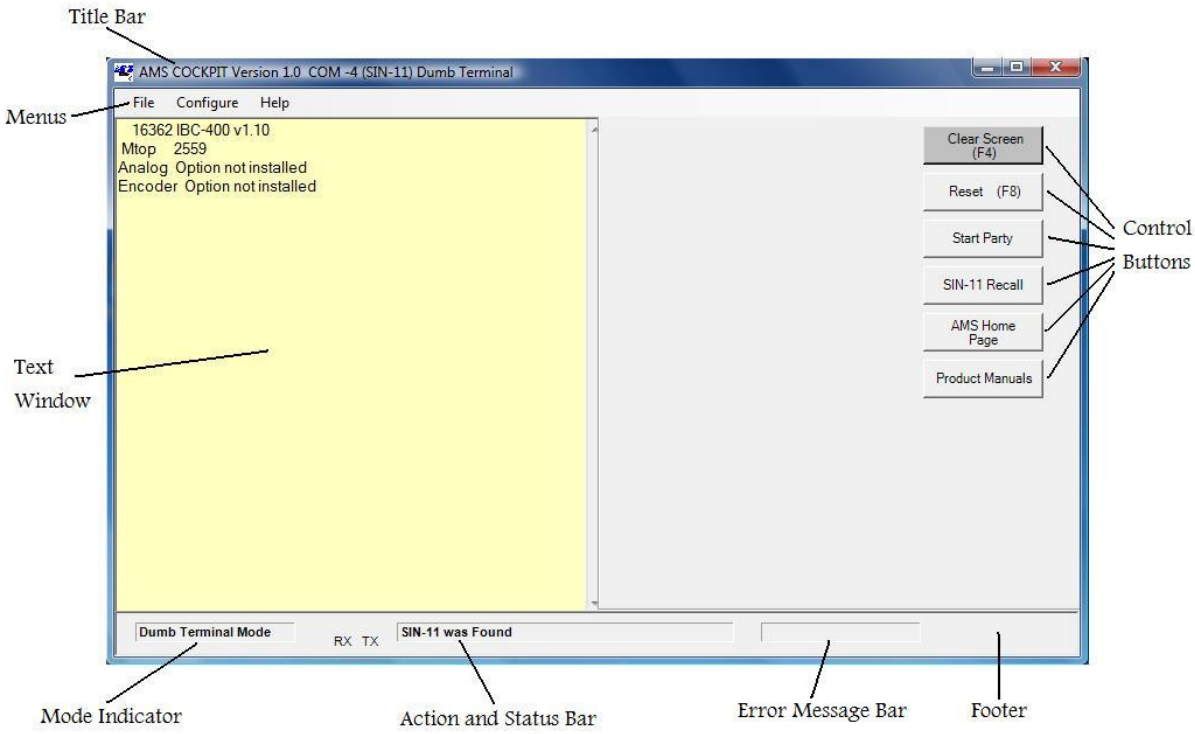
Other terminal programs, such as "HyperTerminal" may also be used. "HyperTerminal" by Microsoft is delivered with Windows Operating systems up to Windows XP. Unfortunately, it is no longer included in Vista and Windows 7.

There is a variety of other terminal programs available that may be used. There are freeware terminal programs available for MAC computers. See the AMS website for more information.

The default baud rate for all AMS products is factory set to 9600 baud. When using a third party software to communicate with the controller, it needs to be ensured that the port settings are as follows: data rate: 9600b/s, data bits: 8, stop bits: 1, parity: none, flow control: hardware. This represents the default for most PC's.

See the chapter entitled Getting Started for initiating the communication between computer and the AMS controller.

Also, there is an extensive manual for AMS Cockpit available from the AMS website.



Main Screen of AMS Cockpit Software

Changing Parameters of the Bluetooth Radio

The user may change some of the parameters of the Bluetooth radio within the SIN-11-B. In order to do so, type "$$$" in the terminal program within 60s after powering up the unit. This command will not be accepted if the unit has been powered up for more than 60s! You should see "CMD" returned. This indicates, that the parameter setting mode has been entered successfully. Valid command will return an "AOK", invalid ones "ERR". If a command is not recognized, a "?" is returned.

To exit the parameter setting mode, type "---"<CR> (three minus signs). Once you exited the parameter setting mode, commands can be issued to the controller again.

Assuming you are still in the parameter setting mode, you may list the main parameters using the "D" command.

Bluetooth Device Name

To change the device name use the "SN" command. The name may have up to 20 characters.

Example: "SN,MySIN-11-B"

When you issue the D command you should see the new name listed under the parameter "BT-Name"

Note that depending on the operating system you are using you may first need to remove the device from the list of Bluetooth devices and add it again in order to see the new name.

Passkey

The passkey can be shanged. Use the "SP" command to set it. The default passkey when shipped or restored to factory settings is: "1234"

Example: "SP,TheSIN-11-Bismine"

To verify that the new passkey has been set you may issue the "D" command. The parameter "PinCod" should show your new password.

Note, after you power down and power up again your BLUEMAX, the new passkey is required to establish a connection. If your operating system does not ask you for the new passkey, you may simply remove the Bluetooth device and add it again while providing the new passkey. See also the Section "Adding the Bluetooth Device" in the Chapter "Installation".

IMPORTANT: Make sure that you retain the information on the passkey. If you changed the passkey from its default and do not remember it, you will need to restore the SIN-11-B to the factory settings in order to establish communication again. See section "Restore factory settings".

Baud Rate

Note that the baud rate cannot be changed using a soft set command alone. In case you require your SIN-11-B to communicate with a higher baud rate to the host PC and or the AMS controllers, please contact AMS.

Exiting Parameter Setting Mode

To exit the parameter setting mode, type "---"<CR> (three minus signs). Once you exited the parameter setting mode, commands can be issued to the controller again.

LEDs

The SIN-11-B features 2 LEDs:

- The red LED indicates that a wireless connection between the SIN-11-B and your computer has successfully been established.
- The orange LED is on if the SIN-11-B has been set to "Party Line" mode by issuing the "&" command.

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-B.
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-B.

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-B 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-B 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-B 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-B. 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-B 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-B memory. The SIN-11-B 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-B command.

Timeouts

The SIN-11-B 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-B 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-B. 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-B has a predefined name of "&." The following commands are executed within the SIN-11-B:

&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-B 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-B 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-B 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-B 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-B to determine its status. This character will immediately echo a response character allowing determination of the SIN-11-B state as follows:

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

Line Buffer

The SIN-11-B 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).

Contact AMS

www.stepcontrol.com
Email: support@stepcontrol.com
Phone: 603-882-1447
Address: 29 River Road Suite B Essex Junction, VT 05452