

***SIN-10  
USERS GUIDE***

*Revision Date: 03/02/2010*

*Table of Contents*

Overview ..... 3  
Out of the Box ..... 4  
Hardware ..... 6  
Demo Power-up ..... 7  
Axis “Name” Assignment ..... 8  
Timeouts ..... 10  
Local Commands ..... 10  
Special Escape Commands ..... 13  
Watchdog Operation ..... 13  
Error Indicators ..... 14  
Round Robin Buffer ..... 14  
Data Rate Selection ..... 14



## Overview

The SIN-10 is an intelligent serial converter/buffer designed to interface a customer's host computer to a stepmotor subsystem consisting of AMS's motor controller products. The SIN-10 simplifies application software and improves performance by allowing commands and protocol macros (SIN-10 "local commands") to be sent in blocks (complete lines).

Communication between the customer's host computer and the SIN-10 takes place at various rates (switch selectable) from 1200 baud to 115k baud. A microprocessor integral to the SIN-10 buffers host command strings of up to 250 characters. This communication is passed on in turn by the SIN-10 to connected controller(s).

Communications between connected controller(s) and the SIN-10 is at 9600, 38.4k or a special 470k baud (also switch selectable). A second 250-character buffer handles controller response data from the "party line" serial bus, and this response data is relayed by the SIN-10 back to the host computer.

SIN-10 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
- Mates to USB serial adapters
- Replacement for standard serial adapters
- 250 character host buffer allows long command sequences
- 250 character "party line" buffer
- DB-9 (RS-232) 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 is difficult, for example).

The SIN-10 solves that problem because it is line driven, making it simple to send one, or a block of commands using the I/O built into any software that is capable of output to a COM port. With the SIN-10 there is no need for special echoed character handshake software.

Version 1.03 adds the following features:

1. Add NAME separators in returned data string.
2. Additional “&V” command to return SIN-10 version number.
3. Ping test to verify SIN-10 presence and current mode: "single" or "party line".

Version 1.04 adds the following features:

1. Implement watchdog recognition.
2. Improve error detection.
3. Clarify some manual descriptions.
4. Eliminate erroneous timeouts with different products.
5. Implement true “round robin” RXD buffer.
6. Two LED blink on error.

### ***Out of the Box***

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-10.

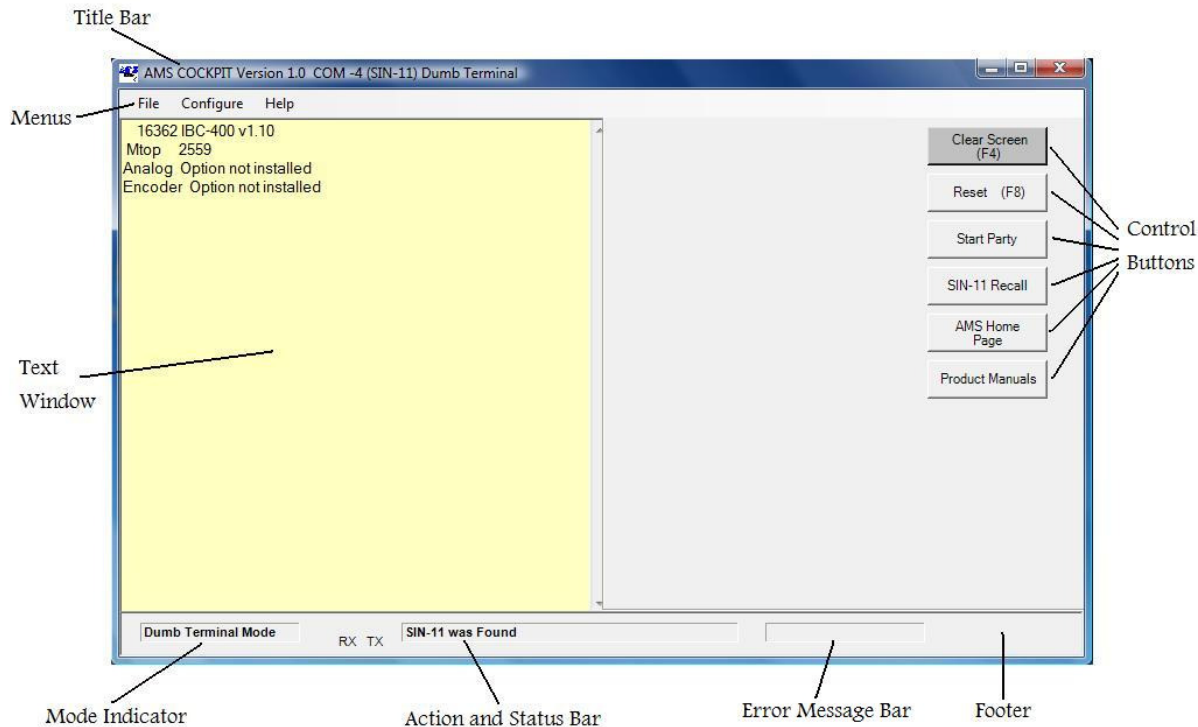
#### **Equipment Setup (quantity one each)**

- SIN-10
- DB-9 Serial cable (supplied with SIN-10)
- CAT-5 data (RJ-45) cable (supplied with SIN-10)
- AMS party line controller product, i.e. SAX, DAX, (C)MAX, DCB, etc.
- Computer with DB-9 serial port (obtain DB-25 adapter if required)

#### **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**

**Connection/Power-on**

1. Insure the AMS controller power is off.
2. Connect the RJ-45 cable of the SIN-10 to the serial INPUT connector of the AMS controller.
3. Connect the serial port end of the SIN-10 cable to computer.
4. It is desirable to have the SIN-10 switches and LED's visible (use a cable as required).
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.**

Unless a special baud rate is to be used ALL SIN-10 switches must be OFF. In any event the Party Line switch must always be off.

6. Start the terminal program. The standard settings are 9600, 8, N, 1 (9600 baud, 8 data, no parity, 1 stop bit).
7. Turn the controller power on. The SIN-10 is powered via the party line cable 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).

**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

1. Enter the command “& P1.” The SIN-10:
  - 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-10 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 “&E5” to turn on the echo.
3. Enter the command line “A+1000;AW0;AZ.”

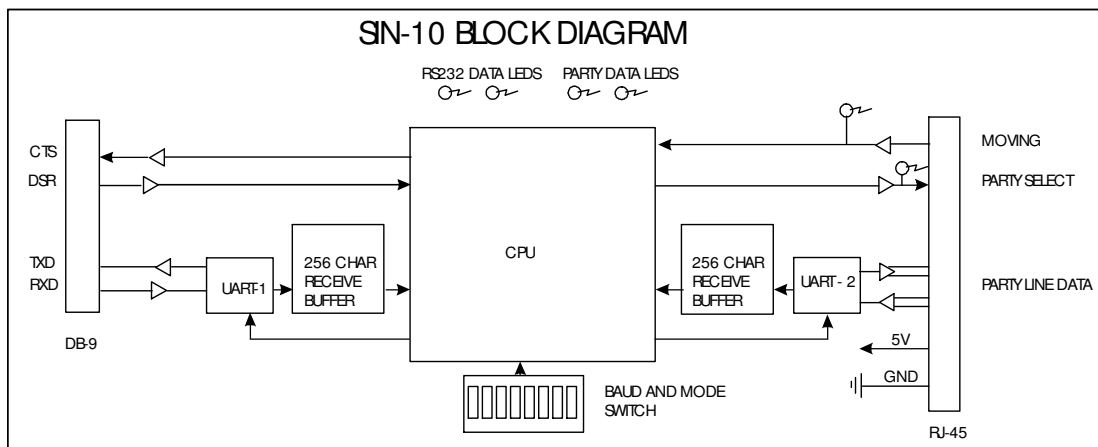
The motor should index 1000 steps (provided the parameter setup is correct), waits until motion is complete, and then reports the position (preceded by the axis name).

### 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.

Good Luck!

## Hardware



The SIN-10 is attached to the host computer via a free COM port. A 9-pin “D” connector plugs directly into a standard PC RS-232 serial port. (A 25-pin to 9-pin adapter may be necessary for some computers). Input to the first axis is through an RJ-45 connector. The SIN-10 obtains its 5-volt power from the INPUT serial connector of the first controller on the party line bus.

### Switches

S1, S2, and S3 set the host computer serial data rate.

S4 and S5 set the party line serial data rate.  
 S6, S7 are reserved for future use (currently "don't-care").  
 S8 selects the party line mode, (OFF or ON) implemented at power-up, i.e. the logic level present at pin 8 of RJ-45.

### LEDs

Several LEDs are provided for diagnostic and status purposes:

- “Party” (Orange) indicates "party line" mode.
- “Moving,” (Green) indicates one or more axes are moving.
- “RS-232” (Yellow) indicates serial RS-232 host activity.
- “RS-422” (Yellow) indicates serial RS-422 controller activity.

## Demo Power-up

The SIN-10 will operate in one of two distinct modes. The mode determines how the controller (party line) bus operates.

1. 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.
2. In the PARTY mode, multiple controllers may be daisy-chained on the party line bus. Pin 8 of the RJ-45 connector is pulled low.

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

The SIN-10 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-10, the following initialization takes place:

1. The SIN-10 is initialized.
2. All eight switches are read and the serial data rates are initialized.
3. The controller party line bus mode, as set by S8, is selected.
4. A "sign-on" message for the SIN-10 is sent to the host.

Connect the computer and a single controller. On power up of the controller, the SIN-10 reads and initializes the serial data rates based on the switch settings. A good starting point would be both ports set to 9600 baud (all switches off). Unless otherwise specified, most AMS party line products are factory set for 9600 baud and may or may not accept other data rates.

### Party Line Switch at Power-up = OFF

When the party line switch (S8) is in the off position, the power-up mode is 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 sends all input characters to the single connected controller, and echoes all characters received from the controller. Commands are executed on receipt of the enter (or line-feed) key.

This mode is useful for single axis debug, diagnostics, and assigning the axis name, which will be required for party line mode. Simple programs may be typed into a controller's NV memory in the single mode.

### Party Line Switch at Power-up = ON

When the S8 switch is in the on position, the power-up mode is party line. The axis must have a valid name (see: *Axis "Name" Assignment*). A low logic signal is asserted on the party select line (pin 8) that is contained in the serial cable to each controller. As the controllers initialize, they set their

internal mode based on this signal line. In this mode, like the single mode, the SIN-10 passes characters in both directions.

## Axis “Name” Assignment

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-10. Insure that the party line switch (S8) is off.
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-10.
4. Type the desired name character (A for our example), followed by the space key. The attached controller should “sign-on.” with "x v2.05<cr><lf>" or similar.
5. Enter the "Examine" command: X<cr>. The axis name will display at the end of line one.
6. Type S<cr> to store the name in non-volatile memory. The name is now saved.
7. Place the axis in party line mode (one axis connected):  
     Enter: &P1<cr> (scan 64 axes)  
           or  
           &P2<cr> (scan 1 axis)
8. Test the setup by typing <AZ<cr>, a request for controller A's position counter. The response should be: "Z 0<cr><lf>.
9. 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 "&P" scan command starts with upper case 'A' and scans through all upper case characters and then all lower case characters. But the names A, B, and C would be detected faster. The command "&P4" would then be a better choice.

As the SIN-10 finds controllers it displays the axis name. Dots are printed in place of non-present axes. 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-10. 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 250), 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 (up to 250 characters long) 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-10 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-10 memory (resulting from a prior &P1 command). The SIN-10 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-10 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-10 command.

## Timeouts

The SIN-10 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-10 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-10. 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-10 has a predefined name of “&.” The following commands are executed within the SIN-10:

### &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-10 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-10 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. 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).

#### **&P0<cr> (Single Mode Dumb Terminal)**

This mode is used in single axis mode:

1. Party line signal (pin 8) is off.
2. Characters from PC go directly to the motor controller.
3. Characters from controller pass directly to PC.
4. Any & commands are executed.
5. Serial data rate conversion is maintained.
6. Used to assign name and store it in memory.

#### **&P1<cr> (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. Dumb terminal mode is OFF.
3. Echo input is OFF.
4. Echo results is ON.
5. 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.

*Note: The "&P(n)" command must be executed as a single command line.*

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 either the name or a '.' is printed. Once this is completed a <CR> is emitted. The SIN-10 now has a list of the axis names that are used for error checking and global (name = \*) commands. For proper operation the party line switch (S8) should be left on when multiple axes are connected (in party line).

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.

#### **&P(n) <cr> (Start Party Line Operation)**

Start party line operation where n is a value between 2 and 65. The SIN-10 starts with upper case 'A', and then tests n-1 consecutive axes to discover all axis names.

Example

&P 3 (test for 2 axes, such as A and B)

&P 65 (test for the maximum number of axes, as in the "&P1" command)

This variation permits customizing the number of axes for your system; minimizing startup overhead.

#### **&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-10 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-10 to determine its status. This character will immediately echo a response character allowing determination of the SIN-10 state as follows:

1. No echo: The SIN-10 is not connected or power is missing.
2. Echo ^D (04d ♦): The SIN-10 is in single line mode.
3. Echo ^F (06d ♠): The SIN-10 is in party line mode.

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

This “ping” can be used to determine if the SIN-10 mode has changed or reset for some reason, enabling the host to recover from unexpected resets, power fail, or other bad events. Note that this command reports the current mode, not the state of S8, the "Party Line" switch.

### 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.

## Watchdog Operation

The MAX series controllers implement “watchdog” hardware that detect errors. Watchdog flags are set when unusual events such as electrical noise, under-voltages (brown outs) occur. One flag indicates that a brown out has occurred. A second flag indicates a timeout was probably caused by a CPU crash.

In party line mode the controller echoes a ! character in place of the name. Even with the ! echo, the commands will continue to execute. The flags can be read with "(name)] 0" command.

These flags **cannot** be reset (cleared) with either the (name)C, ^C or ESC command. In order to reset these flags, either the input power must be cycled off and on (hardware reset) or the "(name)]1" command must be used.

The special command "i 1" forces this error condition for testing.

## 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.
4. Watchdog timeout.

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-10 during its address scan, will cause the characters: "?<name>#<cr><lf>" to be returned to the host.*

If an address scan (e.g. "&P1,cr>") detects NO valid addresses, the string: "<cr><lf>" is returned, and the "Party" LED will be unlit.

## Round Robin Buffer

The host can send multiple command lines without waiting for the previous "lines" to complete. When the RS-232 receive buffer becomes full (250 characters), the CTS will stop character input flow. Each command line in will result in a returned "result" line, even if it is a blank line. Any returned data should be read, checked for applicable errors, and used as required.

*Note: The communications can become confused when multiple lines are sent.*

## Data Rate Selection

### Serial Data Rates To/From Computer

Baud	S1	S2	S3
9600	Off	Off	Off
115k	On	Off	Off
57.6k	Off	On	Off
38.4k	On	On	Off
19.2k	Off	Off	On
4800	On	Off	On
2400	Off	On	On
1200	On	On	On

### Serial Data Rates To/From AMS Controller(s) (Party Line)

Baud	S4	S5
9600	Off	Off
38.4k	On	Off
470k	Off	On

**AMS Product Selection/Serial Data Rate Guide**

Core Microprocessor	AMS Product Family	Available Data Rates		
		9600	38.4k	470k
SMC-C24	SAX/DAX, MCB, DCB-241	Yes	Yes	Yes*
SMC-25	CCB-25	Yes	Yes	Yes
SMC-26	CCB-26, DCB-261, DCB-264, DCB-274	Yes	Yes	Yes
SMC-2000	MAX, (C)MAX	Yes	No	Yes

\*Software V2.05 and above.

DCB-241 = 9,600 baud only. Other serial data rates require a hardware modification.