SmartMotor Communications at a Glance

Items covered in this section: page

SmartMotor Communications at a glance

Notes on Boot-Up sequence of SmartMotors with regards to communications:

They default to 9600 Baud, no parity 8 data bits and 1 stop bit. (9600,N,8,1).

All SmartMotors boot up in ECHO_OFF mode with **no address**.

This means they will respond to globally addressed commands, i.e. commands proceeded by dec128 or hex80.

With in the first 200msec's or so of power up, if they have not received any serial communications, they will begin executing code previously downloaded to them from the top down.

If the Code begins with RUN? , execution will stop at that line until a "RUN" is received via RS-232 serial port.

Multiple motors on a communications line

If more than one motor are to be placed on a communications line, they must be set up properly to avoid communications errors.

RS-232

If RS-232 is used from a host PC or other RS-232 compatible device, all motors must be in ECHO mode.

While in ECHO mode, all data reaching a motor's received port will be "echoed" back out it's transmit port. Since RS-232 serial lines must be daisy-chained together, the motors must be in ECHO mode to work properly.

RS-485

If RS-485 is used, all motors must be in ECHO-OFF mode. RS-485 is a parallel communications network. If any motor was to echo out commands received, it would cause all motors or any other devices on the network to get hit with the same data.

Note: SmartMotors use 2 wire RS-485 standards. This means line biasing determines whether or not the motor is in transmit mode or receive mode at the hardware level. To insure motors do not hang up in the transmit mode, there must be a minimum of a 200mVolt differential between RS-485 A and B channels.

This is easily achieved by placing a pull down resistor of about 500 Ohms from the B channel to ground somewhere on the RS-485 network.. All SmartMotors have a 5Kohm pull-up resistor on both A and B channels already. The 500 Ohm resistor will provide the enough biasing needed to make the hardware default to the receive state. If there are long distances between motors, it may be necessary to provide a resistance across channels A and B. A 200 Ohm resistor wired from A to B at the remote end of the RS-485 line should provide ample voltage drop for needed biasing.

If the above electrical rules are not applied, communications cannot be guaranteed to work.

back to top

Pre-Addressed Motors on Boot-Up

If it is desired for the motor to have an address on boot-up, the motor must have a program downloaded to it with the set address command at or near the top of the program.

Example:

SADDR1 will set the motor's address to 1.

The motor address command uses integer numbers 1-120, however to specifically address a particular SmartMotor, you must precede the desired command with decimal or hex equivalent addresses.

Example:

If a motor's program begins with SADDR1, then a command specifically meant for that motor must be preceded by dec129 or hex81.

Example:

To set commanded velocity on motor 1 to 1000 you would transmit the following:

hex81V=1000

To set commanded acceleration on motor 2 to 500 you would transmit the following:

hex82A=500

The commands must be followed by either a carriage return or space character (dec13 or dec10).

These same rules apply to one SmartMotor talking to another. This can be done via use of the PRINT commands in proper format.

Example: PRINT(#131,"A=1200",#13)

The above code sends out dec131 immediately followed by A=1000 immediately followed by a carriage return.

PRINT(#131,"A-1200 ") would give the same result because a space was included after A=1200.

back to top

Global addressing:

As mentioned at the beginning, all motors without an address respond to any command preceded by dec128 or hex80. This is also true for any motor with an address that is not in "sleep" mode.
If a motor is in Sleep mode, it will only start listening again if the WAKE command preceded by its address is sent to that motor.
Example:

If hex85SLEEP is transmitted, motor 5 will go into "sleep mode."
If hex80x=123 is transmitted, every motor on the serial line will have variable x updated to 123 except motor 5.
If hex85WAKE is transmitted, motor 5 will "wake up" and will respond once again to commands.

back to top

Addressed or De-Addressed state:

It is important to understand addressed or de-addressed states of SmartMotors. These states determine whether or not a SmartMotor will respond to commands.

Lets assume for example that we have 5 motors on a communications network.. All of them have programs downloaded with addresses 1-5 respectively via the SADDR command.

On Boot-up all are ready to listen. They will respond to either a globally addressed command or a specific motor will respond to a specifically addressed command.

Once a specific address is sent out on the line, that motor will be in the "addressed" state. All other motors will be in the "de-addressed" state. what this means id that from that point on, any command sent out to the motors without an address proceeding it, will be acted upon only by the motor in the addressed state. All other motors will basically ignore anything received.

By sending out a command preceded by the global address (hex80 or dec128), all motors will be placed into the "addressed state in will remain in that state until another specific address is transmitted. Under this case of 5 motors addressed 1-5 respectively, if a hex89 for example or any other address outside of hex80-hex85 is sent, al motors would become "de-addressed". No motor would respond to any command until an address within the range of motors on the line is received.

This does come in handy if other Non-SmartMotor devices are on the same network. For example: suppose you have the same 5 SmartMotors on line with a barcode reader or temperature controller. If you wanted to send set-up parameters to these devices but wanted to insure the SmartMotors would ignore the set-up parameters (since they are more than likely not even valid SmartMotor syntax), you could "de-address" all SmartMotors by sending out an address outside the bounds of any motor on the line.

Most RS-485 devices operate this way.

back to top

Addressing SmartMotors from a Host PC or other Serial Device

Note: This only applies to an RS-232 serial daisy chain. It will not work on an RS-485 network. Motors must be pre-addresses in downloaded programs for an RS-485 networks to work at all.

It is important to realize the boot-up state of SmartMotors from the first section to understand this sequence. Please review it if you have not already done so. Go there

Since SmartMotors without addresses default to address zero (hex80 or dec128), a sequence of commands must be issued in proper order to achieve addressing of the SmartMotors.

SmartMotors without programs downloaded into them will not retain addresses from this procedure upon loss and return of power!

The following is an example sequence of addressing 3 SmartMotors from the SMI software terminal screen. Assumptions are as follow:

1. Host PC is set up for 9600 Baud,N,8,1 since this is the power-up default for SmartMotors.
2. Three SmartMotors are wired in serial daisy chain with Tx of Host PC wired to Motor-1 Rx, Motor-1 Tx wired to Motor-2 Rx, Motor-2 Tx wired to Motor-3 Rx, Motor-3 Rx wired to Host PC Rx.

(Tx is RS-232 transmit, Rx is RS-232 Receive)

0ECHO_OFF	Places all motors in echo off
0SADDR1	Set first motor to address 1
1ECHO	Set it to echo mode so the next motor will be able to receive commands
1SLEEP	Set it to sleep mode so it will not act upon following commands
0SADDR2	Set next motor to address 2 and repeat sequence
2ECHO	
2SLEEP	
0SADDR3	
3ECHO	
3SLEEP	
1WAKE	Set all motors to wake status.
2WAKE	
3WAKE	

Note: SMI Software automatically replaces leading numbers in commands with a decimal offset of 128. In other words, 0ECHO_OFF resulted in “(dec128)ECHO_OFF” being transmitted.

This is the equivalent from any other software source:

(dec128)ECHO_OFF
(dec128)SADDR1
(dec129)ECHO
(dec129)SLEEP
(dec128)SADDR2
(dec130)ECHO
(dec130)SLEEP
(dec128)SADDR3
(dec131)ECHO
(dec131)SLEEP
(dec129)WAKE
(dec130)WAKE
(dec131)WAKE

Note: (hex80-83) is the same as (dec180-313), All lines must be terminated with either a carriage return (dec13) or space character.

back to top