

HomeVision and HomeVision-Pro Serial Protocol (Version 3.43)

Custom Solutions, Inc.

Copyright 1996 - 2005

Custom Solutions, Inc.
P.O. Box 33905
Indialantic, FL 32903
csi@csi3.com
www.csi3.com

DOCUMENT PURPOSE

HomeVision can be controlled by commands over the serial interface. It can also report events over the serial interface. Once you understand the protocol, you can create your own programs to control HomeVision and/or respond to messages from HomeVision. This file provides details of the interface protocol.

SERIAL PORTS SUPPORTING COMMUNICATIONS PROTOCOL

The protocol described here applies to the main HomeVision serial port (or HomeVision-Pro serial port 1). It defines how PC software connected to the serial port can communicate with the controller. Prior to PROM version 3.4, the other add-on serial ports (or HomeVision-Pro serial ports 3 and 4) did not support this communications protocol. Thus, most PC software had to use the PC serial port that was connected to the controller's main serial port. In some cases, users may want to run more than one program. For example, one might want the HomeVision software running (perhaps to use its web server), and also a voice recognition program. Each program expects to open a PC serial port connected to HomeVision, and will use the standard HomeVision serial protocol to communicate with it. If only one HomeVision serial port supported this protocol, users couldn't run both programs simultaneously. Now, beginning with PROM version 3.4, the add-on serial ports also support most of this standard serial communications protocol. This allows software programs to connect to other HomeVision serial ports. In general, we recommend that the HomeVision software use the main serial port, and other programs use the add-on ports.

Before an add-on port will use this standard protocol, you must enable it. This is done using the "Serial" tab on the Controller Settings Screen. When enabled, you can still transmit serial messages out the port (by using the appropriate commands in your schedule), and you can still process received messages (using the Serial Data Input Event).

Note that you only need to enable this feature for an add-on port if the port will connect to a PC running software that uses the HomeVision standard serial protocol. If instead you're connecting to external devices like thermostats, security systems, LCD screens, I/O boards, etc., you should leave this option disabled.

Beginning with PROM version 3.4, the add-on serial ports can optionally support most of the standard serial communications protocol. Following are the things the add-on ports cannot do:

- Load schedules into the controller.
- Learn infrared signals.
- Transmit "event reporting" messages. These are the messages that you can configure using the "Event reporting" tab on the Controller Settings Screen in the PC software. These messages can only be reported out serial port 1. Examples of these messages are shown below:

```
90 X-10 House/Unit : P 1
85 Time: 23:05:47 Date: 09/15/03

91 X-10 House/Func : P Off
85 Time: 23:05:47 Date: 09/15/03

98 Input Port Changed: #09 Low
85 Time: 23:05:53 Date: 09/15/03
```

Note that the "PC communications" reporting messages can be reported over the add-on ports. These are the messages that you can configure using the "PC communications" tab on the Controller Settings Screen in the PC software, and are described in the "Auto reporting" file. Most

PC software that relies on report messages from the controller uses this method instead of the “event reporting” messages mentioned above. Thus, these programs should function when connected to an add-on serial port.

SERIAL INTERFACE COMMANDS TO HOMEVISION

Nearly every command that can be included in a schedule can also be transmitted to the controller over the serial interface. When received by the controller, it executes the command and reports the results back over the serial interface.

This capability provides you powerful control over your home automation system. You can program the controller to run independently of your computer, but still be able to execute any command at any time from the computer. It’s also possible to design a complete automation system running on your computer. The computer could decide what to do and when, and then issue the appropriate commands to the controller.

COMMAND STRUCTURE

Commands issued to the controller must follow a specific format in order to be recognized. Although commands differ in the amount of data required, they all follow the same basic format:

Header Character (a comma)
Command Character
<Command Data>
Terminator Character (a carriage return)

The meanings of each portion of the command are provided below:

- Header Character: A single character (a comma) that tells the controller that a command will follow. All commands must start with this header character.
- Command Character: A single, unique character that identifies the type of command. There are different characters for X10 commands, infrared commands, flag commands, etc. The command character for each command type is given in Table 1. Note that the command character is case-sensitive.
- <Command Data>: Consists of 0 or more bytes of data that define what the command must do. For example, when commanding the controller to transmit an X10 signal, you must specify the house/unit code (such as A1, C7, etc.) and the function (On, Off, Dim, etc.). These are supplied in the “Data” field. The brackets <> indicate that the data length varies with different commands. Some commands require no data at all. For these, the “Data” field is skipped. The total command length (in bytes) and any required data bytes are provided in Table 1.
- Terminator: A single character (a carriage return) that completes the command and tells the controller to execute it.

COMMAND RESPONSE

If the controller successfully performs the command, it reports that the command was completed by transmitting the following message over the serial interface:

Done(Carriage Return)(Line Feed)(SOH)

Only the word “Done” will appear on a terminal emulator. You’ll notice that three bytes of information are transmitted after “Done”. The carriage return and line feed move the cursor to the start of the next row. SOH is the ASCII character with a binary value of 1. This byte is transmitted by the controller to make it easier for a PC program to determine when the command response is complete. When you see this value, you know that the response is complete. The only other time the controller would transmit this value is if your schedule includes a command to transmit the binary value of 1.

COMMAND GENERATION

There are several different ways to generate and send a serial command to the controller:

- 1) Use a communications or terminal emulation program (such as the “Terminal” program supplied with Windows). Almost all of these programs provide the capability to type data on the keyboard and have the character sent directly out the serial port. The programs also display any data received from the controller. All you have to do is configure the program (selecting the appropriate comm port, baud rate, etc.), then start typing.

Most programs send the data immediately after the key is pressed. Pressing the back space or delete keys cannot undo what has already been transmitted to the controller. If you make a mistake while transmitting data to the controller, it might execute an unplanned command. Be sure to read and understand the section on aborting commands before you try to send commands from a terminal program.

- 2) Use the controller command interface built into the PC software (located under the “Control” menu). Unlike terminal emulators, this command screen doesn’t transmit any data until the carriage return key is pressed. When it is pressed, the entire line is transmitted with a carriage return at the end. This allows you to see the entire command and verify it’s right before transmitting it. You should note several things about the screen:
 - The carriage return will not appear in the command line, but it will be automatically transmitted at the end of the line. This carriage return is interpreted by the controller as the command terminator.
 - The command line is not cleared after transmitting it. If you press the carriage return again, the same command will be transmitted again.
 - The window will display any data received from the controller. The CLEAR button will clear this window.
- 3) Send serial data from another application. Any program that can transmit data over the serial interface can transmit commands to the controller.

COMMAND PROCESSING

This section describes how the controller decodes the incoming serial data to identify the command.

Whenever the controller receives serial data, it checks to see if it follows the HomeVision command format. After a complete and valid command is received, it is executed. Any invalid data will terminate the receive cycle (all previously received data will be ignored) and the controller will start over waiting for the next command. The receive cycle steps through the following states:

- 1) The controller waits for a header character (a comma) to be received. Any data that is not a comma is ignored. Only when a valid header is received will it go to State 2.
- 2) The controller now expects to receive a command character. If the character immediately following the header is not a valid command character, the receive cycle is terminated and the controller returns to State 1 to wait for another header character. If it is a valid command character, it moves to State 3.
- 3) The controller is now expecting to receive a specific number of data bytes (which can be different for each command). If the command does not require any data, it immediately goes to State 4. Command characters that require data have a specific data length we'll call "L". The next L bytes received are stored in an internal buffer. After the last byte (byte number L) is received, the controller moves to State 4.
- 4) The controller now expects to receive the terminator code (a carriage return). If any other character is received instead, the receive cycle is terminated and the controller returns to State 1 to await another header character. The command is ignored. If the proper terminator code is received, the controller moves to State 5.
- 5) The controller acknowledges that a valid command was received by transmitting the following response over the serial interface:

Cmd :

In the response, the "##" will be replaced by the Command ID Number of the received command (as shown in Table 1). There is no carriage return or line feed at the end of the line. The controller then moves to State 6.

- 6) The controller now executes the command based on the command character and the associated data. If the command requires a data response from the controller (such as when you are reading the value of a flag or variable), it will be transmitted here. After completing the command, the controller moves to State 7.
- 7) The controller reports that the command was completed by transmitting the following message over the serial interface, as discussed previously:

Done(Carriage Return)(Line Feed)(SOH)

The command process is now complete and the controller returns to State 1 to wait for another command.

TO ABORT COMMANDS ENTERED IN ERROR

It's easy to make a mistake when entering data from a keyboard. If you're using a terminal emulation program that transmits each character as the key is pressed, this could cause unexpected results. If you start entering a command and realize you've made a mistake, the command can always be aborted. How this is done depends on where in the command entry sequence you made the mistake.

- If you enter the header character (a comma) by mistake, press any key that is not a valid command character. The space bar is probably the easiest choice. This immediately aborts the receive process.

- If you enter the header character and then a wrong command character, the controller will be expecting either data or the terminator. The only way to abort is to first enter the required amount of data (or more), then enter an invalid terminator character. Since different commands require different amounts of data, you may not be sure how many times to enter a keystroke. Some commands expect 16 bytes of data, so you need to press a key (any key except the carriage return) at least 16 times to fill up the data buffer. After the controller has received the amount of data it expects, it will be looking for the carriage return as a terminator. Pressing any other key at this point will abort the receive process.

The safest way to get out of this mess is to press any key EXCEPT FOR THE CARRIAGE RETURN 17 times. This will always abort the receive process regardless of which command character was entered (it does this by causing an incorrect terminator). However, if you know how many data bytes are required for the command character you entered inadvertently, you may not really need 17 keypresses. Simply enter 1 more byte than the required data length. For example, if the command requires 4 bytes, enter 5 keypresses. This will fill up the data buffer, and then abort when an invalid terminator character is received. Again, you can use any key except for the carriage return (the carriage return is the expected terminator).

- If you've already entered the header, command character, and its data, the controller is expecting the terminator. Press any key except for the carriage return and the receive process will be aborted.

The key to this process is to make sure not to press the carriage return (the terminator) when the controller is waiting for it.

CONTROLLER COMMAND FORMATS

Table 1 lists all of the controller command types and the format of the data in each. The meanings of the columns are explained below:

- "COMMAND DESCRIPTION"

This is a brief description of the command type. You'll notice several things about the available commands:

- Most command types are for specific objects, such as flags, variables, input ports, etc. Each object type has one or more commands to control them.
- Some object types have multiple commands, each requiring different amounts of data. For example, there is one command type for X-10 commands which require 4 bytes of data. There is a different command type for X-10 commands which require 6 bytes of data.

- "COMMAND CHARACTER / ASCII VALUE"

The character listed at the left of this column is the Command Character. If you are entering commands from a terminal emulator program, this is the key you would press immediately after the header character. The three digits at the right of the column show the ASCII value of the character. The "h" indicates the value is shown in hex format. If you are transmitting binary data from a PC program, this is the value to transmit (it's the same as transmitting the character).

There are other command characters not listed in Table 1. They are used to load schedules and read and write internal memory. Transmitting these commands could corrupt the schedule, requiring reloading. Therefore, command characters not listed in the table should never be sent.

- “COMMAND LEN”

This is the command data length. This many bytes (characters) must be provided in the <data> field of the command.

- “COMMAND DATA”

This column describes the order of the data in the <data> field (the actual data to be entered is described in the Appendix). The numbers such as “1/2” indicate the order in which the data is entered. The data is usually entered in pairs of characters. “1/2” indicates that characters 1 and 2 together make up the first value.

The data values always range from 0 to 255. Each value is entered as two characters in hex format. Here are some examples:

<u>To Enter This Decimal Value</u>	<u>Type These Characters</u>
0	00
1	01
9	09
10	0A
15	0F
16	10
17	11
32	20
48	30
64	40
80	50
255	FF

Notice that you must always enter two characters, using a leading zero where necessary. The characters A, B, C, D, E, and F needed to represent hex values are case-sensitive and must be capitalized.

Several points should be noted about the command data:

- Many command types require an object ID number (such as timer ID number, X10 module ID number, etc.). This is the same number listed on the object summary screens for each object. It ranges from 0 to 255. This specifies the object that the command will take action on.

For some commands, this field is not used, but you must still enter a value. For example, the command to clear all flags doesn't really need a flag ID number since the command affects all flags. Nevertheless, when entering this command you must still fill in this field. You can simply enter zero (actually two 0's).

For some commands, this field is used to provide data other than an object ID. The details in the appendix will indicate what you should provide in this field.

- Most command types require a command ID number. This indicates the specific command to be performed (such as which X10 command to perform or which output port command to perform). Most of the available commands are identical to the commands you could include in your schedule. In other words, the same commands you see listed in the actions entry screen

for each command type are available here. Refer to the appendix for the ID numbers and details of each command. Refer to the user's manual for descriptions of what each command does.

- "RESPONSE ID #"

Each command has a unique ID Number (we interchangably refer to this as the Command ID # or the Response ID #). This value is transmitted by the controller in response to the command. It is a two-digit hex number. You can read this value to verify that the proper command was performed by the controller.

- "RESPONSE DATA"

This column describes the data that will be transmitted by the controller in response to the command. Only a few commands (primarily those to read the current state of an object) provide response data. However, the controller always provides a acknowledgement of all commands, as described in "Command Processing".

COMMAND DETAILS

Table 1 shows the format of each command. However it doesn't give the details of exactly what the command data should be. For commands that require it, this information is provided in the appendix.

TABLE 1 - CONTROLLER COMMANDS FORMAT

COMMAND				RESPONSE		NOTES
DESCRIPTION	CHARACTER / ASCII VALUE	LEN	DATA	ID #	DATA	
X-10 Command - 4 bytes	X / 58h	4	1/2: module ID # 3/4: command #	28	Various	See appendix for command details
X-10 Command - 6 bytes	P / 50h	6	1/2: module ID # 3/4: command # 5/6: "##" data value	20	No Data	See appendix for command details
X-10 Command - 8 bytes	g / 67h	8	1/2: module ID # 3/4: command # 5/6: data byte 1 7/8: data byte 2	37	No Data	See appendix for command details
Input Port Command	: / 3Ah	4	1/2: port ID # 3/4: command #	0A	No Data	See appendix for command details
Output Port Command - 4 bytes	3 / 33h	4	1/2: port ID # 3/4: command #	03	No Data	See appendix for command details
Output Port Command - 6 bytes	M / 4Dh	6	1/2: output port ID # 3/4: command # 5/6: "##" data value	1D	No Data	See appendix for command details
Output Port Command - 8 bytes	Z / 5Ah	8	1/2: output port ID # 3/4: command # 5/6: "##" value MSB 7/8: "##" value LSB	2A	No Data	See appendix for command details
IR Command	; / 3Bh	4	1/2: IR signal ID # 3/4: command #	0B	No Data	See appendix for command details
IR Command to Transmit Multiple Times	< / 3Ch	4	1/2: IR signal ID # 3/4: # of times to xmit	0C	No Data	See appendix for command details
Transmit Standard IR Signal Once	T / 54h	4	1/2: device code 3/4: key code	24	No Data	
Macro Command	O / 4Fh	4	1/2: macro ID # 3/4: command #	1F	No Data	See appendix for command details

Phone Command	h / 68h	4	1/2: command # 3/4: data byte	38	No Data	See appendix for command details
Flag Command - 4 bytes	> / 3Eh	4	1/2: flag ID # 3/4: command #	0E	None except for read command which returns 1 byte as 2 ASCII/hex characters	See appendix for command details

COMMAND				RESPONSE		NOTES
DESCRIPTION	CHARACTER / ASCII VALUE	LEN	DATA	ID #	DATA	
Flag Command - 6 bytes	` / 60h	6	1/2: flag ID # 3/4: command # (must be 4) 5/6: "##" data value	30	No Data	See appendix for command details
Variable Command - 4 bytes	Y / 59h	4	1/2: variable ID # 3/4: command #	29	No Data	See appendix for command details
Variable Command - 6 bytes	? / 3Fh	6	1/2: variable ID # 3/4: command # 5/6: "##" data value	0F	None except for read command which returns 1 byte as 2 ASCII/hex characters	See appendix for command details
Timer Command	= / 3Dh	12	1/2 : timer ID # 3/4 : command # 5/6 : new centisecond value 7/8 : new second value 9/10 : new minute value 11/12: new hour value	0D	None except for read command which returns 1 byte as 2 ASCII/hex characters	See appendix for command details
Controller Command	G / 47h	2	1/2: command #	17	Various depending on command	See appendix for command details
Video Command - 4 bytes	L / 4Ch	4	1/2: command # 3/4: "##" data value	1C	No Data	See appendix for command details
Video Command - 15 bytes	d / 64h	15	1 : # of characters to write to screen 2-15: characters to write	34	No Data	See appendix for command details
Scheduled Event Command	H / 48h	4	1/2: scheduled event ID # 3/4: command #	18	No Data	See appendix for command details
Periodic Event Command	I / 49h	4	1/2: periodic event ID # 3/4: command #	19	No Data	See appendix for command details
Caller ID Command	R / 52h	14	1/2: command # 3 - 14: 12 ASCII characters	22	No Data	See appendix for command details

Thermostat Command - 2 bytes	c / 63h	2	1/2: command #	33	No Data	See appendix for command details
Thermostat Command - 6 bytes	e / 65h	6	1/2: command # 3/4: zone # 5/6: "##" data value	35	See appendix for response details	See appendix for command details

COMMAND				RESPONSE		NOTES
DESCRIPTION	CHARACTER / ASCII VALUE	LEN	DATA	ID #	DATA	
Multifunction Expansion Board Command	Q / 51h	4	1/2: command # 3/4: "##" data value	21	See appendix for response details	See appendix for command details
Read Date/Time From Controller	@ / 40h	0	No Data	10	1/2 : centisecond (00-99) 3/4 : second (00-59) 5/6 : minute (00-59) 7/8 : hour (00-23) 9/10 : day (1-7) 11/12: date (01-31) 13/14: month (01-12) 15/16: year (00-99)	Time/date data is reported in binary coded decimal format, not hex.
Set Controller Date/Time	A / 41h	14	1/2 : second (00-59) 3/4 : minute (00-59) 5/6 : hour (00-23) 7/8 : day (1-7) 9/10 : date (1-31) 11/12: month (1-12) 13/14: year (00-99)	11	No Data	Time/date data must be transmitted in binary coded decimal format, not hex. Monday = 1, Tuesday = 2, etc.
Set Scheduled Event Fixed Time	B / 42h	6	1/2: scheduled event ID # 3/4: hour value (0-23) 5/6: minute value (0-59)	12	No Data	Doesn't change random interval
Set Scheduled Event Relative Time	C / 43h	6	1/2: scheduled event ID # 3/4: base time: 00 = before sunrise 01 = after sunrise 02 = before sunset 03 = after sunset 04 = before dark 05 = after dark 5/6: minute offset (0 - 255)	13	No Data	Doesn't change random interval

Set Scheduled Event Random Time Interval	D / 44h	4	1/2: scheduled event ID # 3/4: interval in minutes (0-255)	14	No Data	
System Variable Command	^ / 5Eh	6	1/2: command # 3/4: variable data type 5/6: ## value	2E	See appendix for response details	See appendix for command details

COMMAND				RESPONSE		NOTES
DESCRIPTION	CHARACTER / ASCII VALUE	LEN	DATA	ID #	DATA	
Security System Command	l / 6Ch	4	1/2: command # 3-12: "##" data value	3C	See appendix for response details	See appendix for command details
Data Log Command	f / 66h	12	1/2: command # 3-12: variable data type	36	See appendix for response details	See appendix for command details
Simulate Infrared Remote Keypress	J / 4Ah	1	1: IR remote button code	1A	No Data	See appendix for command details
Miscellaneous Commands	[/ 5Bh	15	1: command # 2-15: see appendix for specific data	2B	See appendix for response details	See appendix for command details
Light Command - 4 bytes	i / 69h	4	1/2: light ID # 3/4: command #	39	Variable	See appendix for command details
Light Command - 6 bytes	j / 6Ah	6	1/2: light ID # 3/4: command # 5/6: "XX" data value	3A	No Data	See appendix for command details
Light Command - 8 bytes	k / 6Bh	8	1/2: light ID # 3/4: command # 5/6: "XX" data value 7/8: "YY" data value	3B	No Data	See appendix for command details

EXAMPLES

The appendix contains examples for each command which is described there. Examples of the other command types are shown below:

- To read the time and date from the controller, use this command:

,@

- To set the controller's time and date to 8:55 AM (and zero seconds), Tuesday, April 22, 1997, use this command:

,A00550802220497

Note that:

- The numbers are entered in decimal format, not hex.
- The number for Tuesday is 02.
- The number for April is 04.
- Only the last two digits of the year are entered.

To set the controller's time and date to 11:30 PM (and zero seconds), Sunday, December 25, 2001, use this command:

,A00302307251201

Note that:

- The hour value is entered in military (24-hour) time. For example, 11 PM is 23 hours.
- The number for Sunday is 07 (days are 01 (Monday) to 07 (Sunday)).
- The number for December is 12 (months are 01 (January) to 12 (December)).

- To set scheduled event 01 to be performed at 8:55 AM, use this command:

,B010837

Note that:

- The values are entered in hex format (not decimal format as in the previous example). Therefore, 55d is entered as 37h.

- To set scheduled event 01 to be performed at 11:30 PM, use this command:

,B01171E

Note that:

- The hour value is entered in military (24-hour) time. Therefore, 11 PM is 23 hours in decimal. However, this must be converted to hex, which is 17h.

- To set scheduled event 01 to be performed 5 minutes before sunset, use this command:
`,C010205`
- To set scheduled event 01 to be performed 60 (3C hex) minutes before sunrise, use this command:
`,C01003C`
- To set scheduled event 01 to have a random interval of +/- 30 (1E hex) minutes, use this command:
`,D011E`
- To transmit a standard IR signal with a device code of 24h and a key code of BCh, use this command:
`,T24BC`

EVENT REPORTING MESSAGES FROM HOMEVISION

When the controller is configured to report events, it will transmit a new message every time the event occurs. The message follows the following format:

`XX <Message> {CR} {LF} {Terminator}`

- “XX” is the Message ID Number. Each type of report message from the controller has a unique ID number in a 2-digit hexadecimal format. Each digit will be a hexadecimal value (numbers 0-9 or letters A-F). When converted to decimal format, they represent the numbers 0 through 255. By reading these two bytes, a user program can quickly identify the type of message from the controller.
- “<Message>” will contain a text description of the event.
- {CR} signifies that a carriage return is sent.
- {LF} signifies that a line feed is sent.
- {Terminator} signifies that a single-byte terminator code is sent. The byte has a value 1d, which is the ASCII code SOH (this is NOT the number 1, which is ASCII value 49d). The terminator is provided to make it easier for a computer program to read and interpret the messages from the controller.

NOTE: The carriage return and line feed do not actually appear on the screen if you're using a terminal emulation program. Rather, they cause the cursor to move to the beginning of the next line. The terminator code (ASCII code SOH) also will not appear on some terminal programs (although other programs may display it as a special symbol).

Table 2 shows the format of each message and gives examples.

TABLE 2 - CONTROLLER MESSAGE FORMAT

ID #	DESCRIPTION	NOTES	EXAMPLES
80	Power restored	Transmitted every time power is applied to unit.	80 Power Restored. {CR}{LF}{SOH}
81	Self-test passed	The self-test is run after power is applied. This message is transmitted if it passes.	81 Self-Test Passed!{CR}{LF}{SOH}
82	Self-test failed	The self-test is run after power is applied. This message is transmitted if it fails.	82 Self-Test Failed # 01 {CR}{LF}{SOH} 82 Self-Test Failed # 04 {CR}{LF}{SOH} NOTE: The numbers, when converted to binary, show which test(s) failed: Bit 0 = Port A/B Control Test Bit 1 = External RAM Test Bit 2 = Port C Test Bit 3 = Clock Test Bit 4 = Microcontroller Test Bit 5 = PROM Test Bit 6 = Not Used Bit 7 = Not Used
83	Error message	The controller continually checks for errors during operation. This message is transmitted any time a new error occurs. Repeats of the previous error are not retransmitted.	83 New Error Message = 26 {CR}{LF}{SOH} 83 New Error Message = 3D {CR}{LF}{SOH} NOTE: Refer to the owner's manual for a list of error numbers and descriptions.
85	Time and date	If time and date reporting is enabled, it will be transmitted after each message.	85 Time: 00:00:00 Date: 01/01/96{CR}{LF}{SOH} 85 Time: 12:55:30 Date: 10/15/00{CR}{LF}{SOH} 85 Time: 23:59:59 Date: 12/31/15{CR}{LF}{SOH} NOTE: These times and dates are: 12:00:00 AM, January 1, 1996 12:55:30 PM, October 15, 2000 11:59:59 PM, December 31, 2015
86	Controller reset	Transmitted if the controller resets due to an error	86 Controller reset. {CR}{LF}{SOH}

88	Scheduled event	Transmitted when a scheduled event occurs.	88 Scheduled Event Occurred: #00 {CR}{LF}{SOH} 88 Scheduled Event Occurred: #A4 {CR}{LF}{SOH} NOTE: The number is the scheduled event ID number, in hex.
----	-----------------	--	--

ID #	DESCRIPTION	NOTES	EXAMPLES
89	Periodic event	Transmitted when a periodic event occurs.	89 Periodic Event Occurred: #0A {CR}{LF}{SOH} 89 Periodic Event Occurred: #1D {CR}{LF}{SOH} NOTE: The number is the periodic event ID number, : hex.
8A	Timer event	Transmitted when a timer goes off (expires).	8A Timer Went Off: #00 {CR}{LF}{SOH} 8A Timer Went Off: #0B {CR}{LF}{SOH} NOTE: The number is the timer ID number, in hex.
90	X-10 house and unit code received	Transmitted when an X-10 signal consisting of a house code and a unit code is received.	90 X-10 House/Unit : A 1{CR}{LF}{SOH} 90 X-10 House/Unit : B 9{CR}{LF}{SOH} 90 X-10 House/Unit : G 10{CR}{LF}{SOH} 90 X-10 House/Unit : P 16{CR}{LF}{SOH} NOTE: The units codes are in decimal format, not hex.
91	X-10 house and function code received	Transmitted when an X-10 signal consisting of a house code and a function code is received.	91 X-10 House/Func : A On{CR}{LF}{SOH} 91 X-10 House/Func : B Off{CR}{LF}{SOH} 91 X-10 House/Func : C All Lights On{CR}{LF}{SOH} 91 X-10 House/Func : D All Lights Off{CR}{LF}{SOH} 91 X-10 House/Func : E All Units Off{CR}{LF}{SOH} 91 X-10 House/Func : F Bright{CR}{LF}{SOH} 91 X-10 House/Func : G Dim{CR}{LF}{SOH} 91 X-10 House/Func : H Status Request{CR}{LF}{SOH} 91 X-10 House/Func : I Status Is Off{CR}{LF}{SOH} 91 X-10 House/Func : J Status Is On{CR}{LF}{SOH} 91 X-10 House/Func : K Hail Request{CR}{LF}{SOH} 91 X-10 House/Func : L Hail Acknowl.{CR}{LF}{SOH} 91 X-10 House/Func : M Extended Code{CR}{LF}{SOH} 91 X-10 House/Func : N Extended Data{CR}{LF}{SOH}

ID #	DESCRIPTION	NOTES	EXAMPLES
92	X-10 function code only received	<p>Transmitted when an X-10 signal consisting of only a function code is received. Only the preset dim should fall into this category, as it is the only X-10 signal transmitted without a house code (the house code bits are used to hold the preset dim level instead).</p> <p>However, when HomeVision receives the other function codes, it will sometimes report them in message 92 instead of 91 (due to a bug). You should treat messages 91 and 92 exactly the same. In either case, you need to look at the words to determine what signal was received.</p>	<p>92 X-10 Func Only : A On{CR}{LF}{SOH} 92 X-10 Func Only : B Off{CR}{LF}{SOH} 92 X-10 Func Only : D All Lights On{CR}{LF}{SOH} 92 X-10 Func Only : C All Lights Off{CR}{LF}{SOH} 92 X-10 Func Only : E All Units Off{CR}{LF}{SOH}</p> <p>92 X-10 Func Only : Level 00 Preset Dim{CR}{LF}{SOH} 92 X-10 Func Only : Level 01 Preset Dim{CR}{LF}{SOH} 92 X-10 Func Only : Level 0A Preset Dim{CR}{LF}{SOH} 92 X-10 Func Only : Level 0D Preset Dim{CR}{LF}{SOH} 92 X-10 Func Only : Level 10 Preset Dim{CR}{LF}{SOH} 92 X-10 Func Only : Level 1F Preset Dim{CR}{LF}{SOH}</p> <p>NOTE: For the preset dim signals, the number represents the preset dim level (ranging from 0 to 1F), in hex.</p>
93	X-10 transmission failed	Transmitted when an X-10 transmission fails. This occurs when the controller transmits an X-10 signal the maximum specified number of times, but each attempt fails.	93 X-10 Transmit Failed (Collision){CR}{LF}{SOH}
98	Input port changed	Transmitted when an input port changes state.	<p>98 Input Port Changed: #00 Low{CR}{LF}{SOH} 98 Input Port Changed: #05 High{CR}{LF}{SOH}</p> <p>NOTE: The number is the input port ID number, in hex.</p>
A1	Standard format IR signal received	Transmitted when a "standard format" infrared signal is received. This occurs regardless of whether an IR signal that matches the device and key code is defined in the schedule.	<p>A1 Received IR Code = 0A F8 {CR}{LF}{SOH} A1 Received IR Code = BB 16 {CR}{LF}{SOH}</p> <p>NOTE: The first characters (0A and BB in these examples) are the device codes, and the last characters (F8 and 16) are the key codes. All values are in hex.</p>

A2	Defined IR signal received	Transmitted when a "standard format" infrared signal is received that matches the device and key code of a signal defined in the schedule.	A2 Matches IR Signal # = 04 {CR}{LF}{SOH} A2 Matches IR Signal # = 1C {CR}{LF}{SOH} NOTE: The number is the IR signal ID number, in hex NOTE: This message will always be preceded by an / message reporting the signal's device and key codes as shown: A1 Received IR Code = 0A F8 {CR}{LF}{SOH} A2 Matches IR Signal # = 04 {CR}{LF}{SOH}
----	----------------------------	--	---

ID #	DESCRIPTION	NOTES	EXAMPLES
A3	Unknown IR signal	Transmitted when an infrared signal is received that <u>doesn't</u> follow the "standard format". This could be another format of remote control signal, or perhaps stray IR emissions from a fluorescent light.	A3 Unknown IR Signal 30 {CR}{LF}{SOH} A3 Unknown IR Signal 31 {CR}{LF}{SOH} A3 Unknown IR Signal 3D {CR}{LF}{SOH} NOTE: The numbers 30, 31, 3D, etc. should be ignored. They are only used by the computer when learning IR signals.
B0	Phone rings	When using HomeVision-Phone/Serial or HomeVision-Phone/CID, occurs each time the phone rings and reports the ring count. It also occurs once after the phone stops ringing.	B0 Rings=## {CR}{LF}{SOH} NOTE: "##" will contain the current ring count in decimal format. It will range from "00" to "15".
B1	Phone touchtone received	When using HomeVision-Phone/Serial or HomeVision-Phone/CID, occurs each time a touchtone is received and reports the tone.	B1 Tone=# {CR}{LF}{SOH} NOTE: "#" will contain the tone (0-9, *, or #).
B2	Phone goes on-hook	When using HomeVision-Phone/Serial or HomeVision-Phone/CID, occurs each time a phone goes on-hook.	B2 On-hook {CR}{LF}{SOH}
B3	Phone goes off-hook	When using HomeVision-Phone/Serial or HomeVision-Phone/CID, occurs each time a phone goes off-hook.	B3 Off-hook {CR}{LF}{SOH}
B5	Caller ID information	When using HomeVision-Phone/CID, occurs each time valid caller ID information is detected.	B5 CID: #=xxxxxxxxx NAME=XXXXXXXXXXXXXXXXX {CR}{LF}{SOH} where: xxxxxxxxxx = caller's number XXXXXXXXXXXXXXXXX = caller's name

Note: In many of the examples, a blank space is transmitted immediately before the carriage return. Pay careful attention to this if you are counting the number of bytes in the message as a validity check.

APPENDIX

COMMAND FORMAT DETAILS

X-10 COMMANDS [Command types P (20), X (28), and g (37)]

- Module ID number (2 bytes). Module ID numbers start with zero representing A1 and progress upwards through FFh (255d), which represents P16. Values are entered in hex format, as shown in the following table:

		HOUSE CODE															
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
U N I T C O D E	1	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
	2	01	11	21	31	41	51	61	71	81	91	A1	B1	C1	D1	E1	F1
	3	02	12	22	32	42	52	62	72	82	92	A2	B2	C2	D2	E2	F2
	4	03	13	23	33	43	53	63	73	83	93	A3	B3	C3	D3	E3	F3
	5	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
	6	05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
	7	06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
	8	07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
	9	08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
	10	09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
	11	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
	12	0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
	13	0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
	14	0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
	15	0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE	BE	CE	DE	EE	FE
	16	0F	1F	2F	3F	4F	5F	6F	7F	8F	9F	AF	BF	CF	DF	EF	FF

- Command number (2 bytes). Commands without a "##" value are 4-byte commands (type X). Commands with a "##" value are 6-byte commands (type P).

Cmd # 00 (type X) = On
Cmd # 01 (type X) = Force On
Cmd # 02 (type X) = Off
Cmd # 03 (type X) = Force Off
Cmd # 04 (type X) = Set State To Neutral
Cmd # 05 (type X) = Dim
Cmd # 06 (type X) = Force Dim
Cmd # 07 (type P) = Dim ## Times
Cmd # 08 (type P) = Dim ## Times Slow
Cmd # 09 (type P) = Force Dim ## Times
Cmd # 0A (type P) = Force Dim ## Times Slow
Cmd # 0B (type X) = Brighten
Cmd # 0C (type X) = Force Brighten
Cmd # 0D (type P) = Brighten ## Times
Cmd # 0E (type P) = Brighten ## Times Slow
Cmd # 0F (type P) = Force Brighten ## Times
Cmd # 10 (type P) = Force Brighten ## Times Slow
Cmd # 11 (type P) = Set To Level ##
Cmd # 12 (type P) = Set To Level ## Slow
Cmd # 13 (type P) = Force To Level ##
Cmd # 14 (type P) = Force To Level ## Slow
Cmd # 15 (type X) = Send House/Unit Code Only

Cmd # 16 (type P) = Send House/Function Code ## Only

values are:

On	= 02
Off	= 0A
Brighten	= 06
Dim	= 0E
All Lights On	= 04
All Units Off	= 0C
All Lights Off	= 00
Status Request	= 09
Status Is On	= 07
Status Is Off	= 01
Hail Request	= 0D
Hail Acknowledge	= 05

NOTE: These commands do not send a unit code. When entering a module ID value, you can use the ID number of any of the 16 modules with the desired house code. For example, module ID numbers 00 through 0Fh (15d) all will send house code A.

Cmd # 17 (type X) = Toggle Module

Cmd # 18 (type X) = Enable Module

Cmd # 19 (type X) = Disable Module

Cmd # 1A (type X) = Set State To On

Cmd # 1B (type X) = Set State To Off

Cmd # 1C (type X) = Transmit Module Status Signal

Cmd # 1D (type P) = Transmit Preset Dim To Level ##

Cmd # 1E (type X) = Read Current State

NOTE: This command will cause the controller to transmit the current state over the serial interface. The value should be converted to 8 bits and interpreted as follows:

bit	0	: 0 = Disabled
	1	= Enabled
bits 1-2	: 00 = Off	
	01 = On	
	10 = Neutral	
bits 3-7	: current level (0-16)	

Note that the bits are ordered as follows: 76543210

Example: A light on at level 15 (decimal) and enabled will have this value:

01111011

This is the value 7Bh, and 7B will be transmitted.

Cmd # 1F (type X) = Refresh Module

Cmd # 20 (type X) = Refresh House Code

Cmd # 21 (type P) = Set PCS Light To Level ##

Cmd # 22 (type X) = Micro-Brighten PCS Light 1 Level (1/2%)

Cmd # 23 (type X) = Micro-Dim PCS Light 1 Level (1/2%)

Cmd # 24 (type P) = Put Current Level Into Variable ## {for normal X-10 commands}

Cmd # 25 (type P) = Put Received Preset Dim Level Into Variable ##

```

Cmd # 26 (type P) = Set To Level In Variable ##
Cmd # 27 (type P) = Force To Level In Variable ##
Cmd # 28 (type P) = Transmit Preset Dim To Level In Variable ##
Cmd # 30 (type X) = Enable Off Trigger
Cmd # 31 (type X) = Disable Off Trigger
Cmd # 32 (type X) = Enable On Trigger
Cmd # 33 (type X) = Disable On Trigger
Cmd # 34 (type X) = Clear "Status Request" Bit
Cmd # 35 (type X) = Clear "Status Is On" Bit
Cmd # 36 (type X) = Clear "Status Is Off" Bit
Cmd # 37 (type X) = Clear "Hail Request" Bit
Cmd # 38 (type X) = Clear "Hail Ack" Bit
Cmd # 39 (type X) = Clear "Preset Dim" Bit
Cmd # 3A (type X) = Clear All Advanced X-10 Bits
Cmd # 3B (type g) = Transmit Extended Data Bytes: DATA & COMMAND
Cmd # 3C (type g) = Transmit Extended Code Bytes: DATA & COMMAND
Cmd # 3D (type g) = Transmit Extended Code Bytes: Data in Variable ## & Command
                    in Variable ##
Cmd # 3E (type X) = Put Current Preset Dim Level (0-31) into Variable ##

```

For command type P (6-byte X-10 commands), 2 additional bytes are also required:

- 2 bytes for "##" data value required for the command (shown above).

For command type g (8-byte X-10 commands), 4 additional bytes are also required:

- 2 bytes for DATA value (or variable # holding DATA) required for the command (shown above).
- 2 bytes for COMMAND value (or variable # holding COMMAND) required for the command (shown above).

EXAMPLES:

- To send an ON command to module A1 (ID # 00), use the FORCE ON command (01):

 ,X0001
- To send an OFF command to module A2 (ID # 01), use the FORCE OFF command (03):

 ,X0103
- To send 5 DIM commands to module B2 (ID # 11), use command 7:

 ,P110705
- To send house and unit codes for module B3 (ID # 12), use command 15:

 ,X1215
- To send an ALL LIGHTS ON command to house code A, use command 16 with any module ID number from 00h to 0Fh:

,P001604 or ,P0A1604 or ,P0F1604

- To send an BRIGHT command to house code P, use command 16 with any module ID number from F0h to FFh:

,PF01606 or ,P091606 or ,P0C1606

LIGHTING COMMANDS [Command types i (39), j (3A), and k (3B)]

- 2 bytes for light ID number:
- 2 bytes for command:

Cmd # 00 (type i) = Off
Cmd # 01 (type i) = On
Cmd # 02 (type i) = Toggle
Cmd # 03 (type j) = Brighten XX times
 XX value specified in next 2 bytes
Cmd # 04 (type j) = Dim XX times
 XX value specified in next 2 bytes
Cmd # 05 (type k) = Ramp to level XX at rate YY
 XX value specified in next 2 bytes
 YY value specified in following 2 bytes
Cmd # 06 (type k) = Ramp to level in var XX at rate YY
 XX value specified in next 2 bytes
 YY value specified in following 2 bytes
Cmd # 07 (type j) = Select scene XX
 XX value specified in next 2 bytes
Cmd # 08 (type i) = Refresh light
Cmd # 09 (type j) = Put current level into var XX
 XX value specified in next 2 bytes
Cmd # 0A (type i) = Set state to OFF
Cmd # 0B (type j) = Set state to ON at level XX
 XX value specified in next 2 bytes
Cmd # 0C (type j) = Set state to ON at level in variable XX
 XX value specified in next 2 bytes
Cmd # 0D (type i) = Set state to ON at unknown level
Cmd # 0E (type i) = Set state to unknown
Cmd # 0F (type i) = Set state of current address to OFF *****
Cmd # 10 (type j) = Set state of current address to ON at level in var ## *****
 XX value specified in next 2 bytes
Cmd # 11 (type i) = Read state and level

***** These commands shouldn't be used over the serial interface.

For command type j (6-byte lighting commands), 2 additional bytes are also required:

- 2 bytes for "XX" data value required for the command (shown above).

For command type k (8-byte lighting commands), 2 additional bytes are also required:

- 2 bytes for "YY" data value required for the command (shown above).

EXAMPLES:

- Serial command to turn light 5 on:

,i0501<CR>

- Serial command to brighten light 6 4 times:

,j060304<CR>

- Serial command to set light 12 (0Ch) to level 2 at rate 1 :

,k0C050201<CR>

- Serial command to set light 12 (0Ch) to level 100 (64h) at rate 20 (14h):

,k0C056414<CR>

- Serial command to set light 0 to scene 9:

,j000709<CR>

INPUT PORT COMMANDS [Command type : (0A)]

- Input port ID number (2 bytes).
- Command number (2 bytes):

Cmd # 00 (type :) = Read 8-Bit Port Into Result Value

Reads all 8 bits of the specified port and puts result in system variable "Result Value" . The port to read is determined by the ID number. 0 is for Port B, 1 is for Port C.

Cmd # 01 = Not Used

Cmd # 02 = Not Used

Cmd # 03 (type :) = Disable Input Port

Cmd # 04 (type :) = Enable Input Port

Cmd # 05 (type :) = Read Input Port State Into Result Value

When this command is issued over the serial interface, the controller will also transmit the current state (2 LSBs) over the serial interface. The value is interpreted as follows:

00 = port low and disabled
01 = port low and enabled
02 = port high and disabled
03 = port high and enabled

Cmd # 06 = Not Used

Cmd # 07 (type :) = Enable Input Port Low Trigger

Cmd # 08 (type :) = Disable Input Port Low Trigger

Cmd # 09 (type :) = Enable Input Port High Trigger

Cmd # 0A (type :) = Disable Input Port High Trigger

Cmd # 0B (type :) = Read Port B Value Immediately Into Result Value

Cmd # 0C (type :) = Read Port C Value Immediately Into Result Value

EXAMPLES:

- To disable input port 07, use command 03:

,:0703

- To read the current state of input port 07 into Result Value and also transmit it over the serial interface, use command 05:

,:0705

- To read the state of all 8 lines of input port C into Result Value, use command 00:

,:0100

OUTPUT PORT COMMANDS [Command types 3 (03), M (1D), and Z (2A)]

- Output port ID number (2 bytes).
- Command number (2 bytes). Commands without "##" values are 4-byte commands (type 3). Commands with a "##" value are 6-byte commands (type M). Commands with a "####" value are 8-byte commands (type Z).

Cmd # 00 (type 3) = Set Output Port Low
Cmd # 01 (type 3) = Set Output Port High
Cmd # 02 (type 3) = Toggle Output Port
Cmd # 03 (type 3) = Disable Output Port
Cmd # 04 (type 3) = Enable Output Port

Cmd # 05 (type 3) = Read Output Port State Into Result Value
When this command is issued over the serial interface, the controller will also transmit the current state (2 LSBs) over the serial interface. The value is interpreted as follows:

00 = port low and disabled
01 = port low and enabled
02 = port high and disabled
03 = port high and enabled

Cmd # 06 = Not used

Cmd # 07 (type Z) = Pulse Output Port High For ##.### Seconds
Convert the time to milliseconds (0 - 65,535) and enter in the next 4 bytes, MSB first

Cmd # 08 (type Z) = Pulse Output Port Low For ##.### Seconds
Convert the time to milliseconds (0 - 65,535) and enter in the next 4 bytes, MSB first

Cmd # 09 (type M) = Set 8-Bit Port To ##
The port to write to is determined by the Output Port ID Number. 00 is for Port A, 01 is for Port C. The ## value is the binary values for the 8 lines of the selected port. Enter the ## value in the next 2 bytes.

For command type M (6-byte output port commands), 2 data bytes are also required:

- Enter "##" data value required for command 09.

For command type Z (8-byte output port commands), 4 data bytes are also required:

- Enter milliseconds value in 4 bytes, with the most significant byte first

EXAMPLES:

- To set output port 1 low, use 4-byte command 00:

,30100

- To set output port 0 high, use 4-byte command 01:

,30001

- To pulse output port 3 high for 5 milliseconds, use 8-byte command 07:

,303070005

To pulse it high for 1 second, first convert it to milliseconds. 1 second is 1000 milliseconds. Then convert 1000 (decimal) to hex: 03E8. The command is then:

,3030703E8

- Let's say we want to set all 8 lines of output port A simultaneously as follows:

Line A-0 = High (1)
Line A-1 = High (1)
Line A-2 = Low (0)
Line A-3 = High (1)
Line A-4 = Low (0)
Line A-5 = Low (0)
Line A-6 = Low (0)
Line A-7 = High (1)

First, we must determine the value which corresponds to the state of the 8 lines indicated above. The binary representation is as follows:

10001011

Note that the highest port (A-7) is first (the most significant bit) and the lowest port (A-0) is last (the least significant bit). This 8-bit binary number must then be converted to hex. The result is 8B (139d).

Thus, the full command is:

,M00098B

INFRARED SIGNAL COMMANDS [Command type ; (0B) and < (0C)]

- IR signal ID number (2 bytes) (except for commands 08 and 0A, where this should be the variable ID number). Not used with command 0A or 0B (enter zeros).
- Command number (2 bytes):

```
Cmd # 00 (type ;) = Transmit IR Signal ## Once
Cmd # 01 = Not used
Cmd # 02 (type ;) = Disable IR Signal ## Receive Mode
Cmd # 03 (type ;) = Enable IR Signal ## Receive Mode
Cmd # 04 (type ;) = Disable IR Signal ## Transmit Mode
Cmd # 05 (type ;) = Enable IR Signal ## Transmit Mode
Cmd # 06 = Not used
Cmd # 07 = Not used
Cmd # 08 (type ;) = Transmit IR Signal at Variable ## Once
Cmd # 09 (type ;) = Start Repeated Transmission Of IR Signal ##      ***
Cmd # 0A (type ;) = Stop Repeated Transmission Of IR Signals      ***
Cmd # 0B (type ;) = Start Repeated Transmission Of IR Test Signal  ***
Cmd # 0C (type ;) = Start Repeated Transmission Of IR Signal at Variable ## ***
```

*** Commands 09 through 0C only function with HomeVision-Pro, not HomeVision or HomeVision-PC.

- To transmit an IR signal multiple times, use command type <. Enter the signal ID # followed by the number of times to transmit it.

EXAMPLES:

- To transmit IR signal 20d (14h), use command 00:
 ,;1400
- To transmit IR signal 20d (14h) 7 times, use command type <:
 ,<1407
- To disable reception of IR signal 21d (15h), use command 02:
 ,;1502
- To start transmitting the IR test signal repeatedly, use command 0B:
 ,;000B
- To stop the repeated transmission of IR signals, use command 0A:
 ,;000A

MACRO COMMANDS [Command type 0 (1F)]

- 2 bytes which depend on the command being performed:
 - For command 0-2, use the macro ID number (2 bytes).
 - For command 3, use the variable ID number (2 bytes).
- Command number (2 bytes):

Cmd # 00 (type 0) = Do Macro
Cmd # 01 (type 0) = Disable Macro
Cmd # 02 (type 0) = Enable Macro
Cmd # 03 (type 0) = Do Macro Specified by Variable ##

EXAMPLES:

- To do macro 16d (10h), use command 00:

 ,01000
- To disable macro 17d (11h), use command 01:

 ,01101
- To do the macro specified by variable 5, use command 03:

 ,00503

PHONE COMMANDS [Command type h (38)]

- Command number (2 bytes):

Cmd # 00 (type h) = Dial Digit X
Cmd # 01 (type h) = Take Phone Off-Hook
Cmd # 02 (type h) = Put Phone On-Hook

- 2 data bytes:

- For command 0, use 2 bytes for the digit to dial:

<u>DIGIT</u>	<u>BYTES TO SEND</u>
0	00 (or 0A)
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
*	0B
#	0C

- For other commands, use 00 (two zeros).

EXAMPLES:

- To dial a 5, use command 00:

,h0005

- To take the phone off-hook, use command 01:

,h0100

- To put the phone on-hook, use command 02:

,h0200

FLAG COMMANDS [Command types > (0E) and ` (30)]

- Flag ID number (2 bytes). These bytes are used to hold different data, depending on the particular command you use:

- For commands 00-05, enter the ID number of the flag to use.
- For commands 06-08 (which affect all flags), these bytes will be ignored; you can enter any value.
- For commands 09-0C, enter the ID number of the variable.

- Command number (2 bytes):

Cmd # 00 (type >) = Clear Flag

Cmd # 01 (type >) = Set Flag

Cmd # 02 (type >) = Toggle Flag

Cmd # 03 (type >) = Neutral Flag

Cmd # 04 (type `) = Make Flag Equal To Flag ##

Second Flag ID number (##) specified in next 2 bytes

Cmd # 05 (type >) = Read Flag

This command will cause the controller to transmit the current state over the serial interface. The value is interpreted as follow:

00 = flag clear

01 = flag set

02 = flag neutral

Cmd # 06 (type >) = Clear All Flags

Cmd # 07 (type >) = Set All Flags

Cmd # 08 (type >) = Neutral All Flags

Cmd # 09 (type >) = Clear Flag At Variable ##

For this command, put the variable ID number in the flag ID number bytes

Cmd # 0A (type >) = Set Flag At Variable ##

For this command, put the variable ID number in the flag ID number bytes

Cmd # 0B (type >) = Toggle Flag At Variable ##

For this command, put the variable ID number in the flag ID number bytes

Cmd # 0C (type >) = Neutral Flag At Variable ##

For this command, put the variable ID number in the flag ID number bytes

For command type ` (6-byte flag commands), 2 data bytes are also required:

- 2 bytes to store ## value. This only applies to command number 04, where this holds the ID number of the second flag.

EXAMPLES:

- To set flag 00, use command 01:

,>0001

- To toggle flag 08, use command 02:

,>0802

- To make flag 02 equal flag 03, use the 6-byte command 04:

,`020403

- To clear the flag whose ID number is in variable 255d (FFh), use the 6-byte command 09:

,>FF09

VARIABLE COMMANDS [Command types ? (0F) and Y (29)]

- Variable ID number (2 bytes). These bytes are used to hold different data, depending on the particular command you use. For most commands, they hold the ID number of the variable. Exceptions to this are noted in the command list.
- Command number (2 bytes):
 - Cmd # 00 (type ?) = Variable = Constant ##
Constant ## value specified in next 2 bytes
 - Cmd # 01 (type Y) = Increment Variable [Stop at 255]
 - Cmd # 02 (type Y) = Increment Variable [Roll over at 255]
 - Cmd # 03 (type Y) = Decrement Variable [Stop at 0]
 - Cmd # 04 (type Y) = Decrement Variable [roll under at 0]
 - Cmd # 05 (type Y) = Read Variable
This command will cause the controller to transmit the current variable value over the serial interface
 - Cmd # 06 (type ?) = Variable = Variable + ##
value specified in next 2 bytes
 - Cmd # 07 (type ?) = Variable = Variable + Variable ##
Second variable ID (##) specified in next 2 bytes
 - Cmd # 08 (type ?) = Variable = Variable - ##
Constant ## value specified in next 2 bytes
 - Cmd # 09 (type ?) = Variable = ## - Variable
Constant ## value specified in next 2 bytes
 - Cmd # 0A (type ?) = Variable = Variable - Variable ##
Second variable ID (##) specified in next 2 bytes
 - Cmd # 0B (type Y) = Variable = Variable * 2
 - Cmd # 0C (type Y) = Variable = Variable / 2
 - Cmd # 0D (type ?) = Variable = Variable ##
Second variable ID (##) specified in next 2 bytes
 - Cmd # 0E (type Y) = Variable = Result Value
 - Cmd # 0F (type Y) = Set All Variables Equal To ##
Constant ## value stored in ID Number byte
 - Cmd # 10 (type ?) = Increment Variable (MSB) And Variable ## (LSB) (16-bits)
Put MSB variable ID number in Variable ID Number bytes and
put LSB variable ID number in next 2 bytes)

Cmd # 11 (type ?) = Decrement Variable (MSB) And Variable ## (LSB) (16-bits)
Put MSB variable ID number in Variable ID Number bytes and
put LSB variable ID number in next 2 bytes

Cmd # 12 (type Y) = Variable = Random Number

Cmd # 13 (type ?) = Variable = Variable @ Variable ##
Looks up ## value in next 2 bytes, then uses it as the ID
number of the variable to set the first variable (in
Variable ID Number bytes field) equal to

Cmd # 14 = Not Used

Cmd # 15 = Not Used

Cmd # 16 (type Y) = Variable = Current Centisecond

Cmd # 17 (type Y) = Variable = Current Second

Cmd # 18 (type Y) = Variable = Current Minute

Cmd # 19 (type Y) = Variable = Current Hour

Cmd # 1A (type Y) = Variable = Current Day Of Week

Cmd # 1B (type Y) = Variable = Current Date Of Month

Cmd # 1C (type Y) = Variable = Current Month

Cmd # 1D (type Y) = Variable = Current Year

Cmd # 1E (type Y) = Variable @ Variable ## = Current Centisecond

Cmd # 1F (type Y) = Variable @ Variable ## = Current Second

Cmd # 20 (type Y) = Variable @ Variable ## = Current Minute

Cmd # 21 (type Y) = Variable @ Variable ## = Current Hour

Cmd # 22 (type Y) = Variable @ Variable ## = Current Day Of Week

Cmd # 23 (type Y) = Variable @ Variable ## = Current Date Of Month

Cmd # 24 (type Y) = Variable @ Variable ## = Current Month

Cmd # 25 (type Y) = Variable @ Variable ## = Current Year

NOTE: For the above 8 commands (1E-25), put the variable ##
ID number in the Variable ID Number field

Cmd # 26 (type ?) = Variable @ Variable ## = Constant
Constant value ## stored in next 2 bytes

Cmd # 27 (type Y) = Increment Variable @ Variable ## [Stop at 255]

Cmd # 28 (type Y) = Increment Variable @ Variable ## [Roll over at 255]

Cmd # 29 (type Y) = Decrement Variable @ Variable ## [Stop at 0]

Cmd # 2A (type Y) = Decrement Variable @ Variable ## [roll under at 0]

NOTE: For the above 4 commands (27-2A), put the variable ##
ID number in the Variable ID Number field

Cmd # 2B (type ?) = Variable = Analog Input ##
Analog input signal ## stored in next 2 bytes. This command
is only valid when a multifunction expansion board is used.

Cmd # 2C (type ?) = Variable = Digital Temperature Sensor ##
Temperature sensor ID # specified in next 2 bytes. This
command is only valid when a multifunction expansion board
is used.

Cmd # 2D (type ?) = Variable = Variable * ##
 ## value specified in next 2 bytes

Cmd # 2E (type ?) = Variable = Variable / ##
 ## value specified in next 2 bytes

Cmd # 2F (type ?) = Variable = Variable * Variable ##
 Second variable ID (##) specified in next 2 bytes

Cmd # 30 (type ?) = Variable = Variable / Variable ##
 Second variable ID (##) specified in next 2 bytes

Cmd # 31 (type ?) = Variable = Byte ## from Text Storage Buffer
 Byte location in Text Storage Buffer(##) specified in next 2
 bytes

Cmd # 32 (type ?) = Byte ## in Text Storage Buffer = Variable
 Byte location in Text Storage Buffer (##) specified in next
 2 bytes

For command type ? (6-byte variable commands), 2 data bytes are also required:

- 2 bytes for "##" data value required for the command (shown above).

EXAMPLES:

- To set variable 06 equal to 100d (64h), use 6-byte command 00:
 ,?060064
- To increment variable 06 and roll over if necessary, use 4-byte command 02:
 ,Y0602
- To add 5 to variable 16d (10h), use 6-byte command 06:
 ,?100605
- To set all variables equal to 0, use 4-byte command 0F:
 ,Y000F
- To set the variable @ variable 03 equal to the current hour, use 4-byte
 command 21:
 ,Y0321

TIMER COMMANDS [Command type = (0D)]

- Timer ID number (2 bytes).
- Command number (2 bytes):

Cmd # 00 (type =) = Load Timer With HH:MM:SS:SS And Start
HH:MM:SS:SS value specified in next 8 bytes

Cmd # 01 (type =) = Halt Timer

Cmd # 02 (type =) = Resume Timer

Cmd # 03 (type =) = Stop And Clear Timer

Cmd # 04 (type =) = Reset All Timers

Cmd # 05 (type =) = Read Timer State

This command will cause the controller to transmit the current state over the serial interface. The value is interpreted as follow:

00 = timer stopped (not running or ringing)

01 = timer running

02 = timer ringing

Cmd # 06 (type =) = Put Timer Value Into 4 Variables

Cmd # 07 (type =) = Set Timer To Value Of 4 Variables

- Following 8 bytes:

For command number 00, store 8 bytes for time value (in hex format):

- 2 bytes for CENTI-SECONDS value (0-99)
- 2 bytes for SECONDS value (0-59)
- 2 bytes for MINUTES value (0-59)
- 2 bytes for HOURS value (0-255)

For command numbers 06 and 07, next 2 bytes hold value of first variable (in hex format). Remaining 6 bytes can be any value.

For all other commands, these 8 time value bytes are not used. You can enter any values in them.

EXAMPLES:

- To load timer 08 with 1 hour, 5 minutes, 0 seconds, and 50d (32h) centi-seconds and start, use command 00:

,=080032000501

- To halt timer 08, use command 01:

,=080100000000

Note that the last 8 bytes could contain any values, as they aren't used.

CONTROLLER COMMANDS [Command type G (17)]

- Command number (2 bytes):

Cmd # 00 (type G) = Check Communications

This command does nothing but respond over the serial interface

Cmd # 01 (type G) = Halt Controller

Cmd # 02 (type G) = Resume Controller

Cmd # 03 (type G) = Enable Master Report Mode

Cmd # 04 (type G) = Disable Master Report Mode

Cmd # 05 (type G) = Enable IR Report Mode

Cmd # 06 (type G) = Disable IR Report Mode

Cmd # 07 (type G) = Enable Input Port Report Mode

Cmd # 08 (type G) = Disable Input Port Report Mode

Cmd # 09 (type G) = Enable X-10 Report Mode

Cmd # 0A (type G) = Disable X-10 Report Mode

Cmd # 0B (type G) = Enable Controller Actions Report Mode

Cmd # 0C (type G) = Disable Controller Actions Report Mode

Cmd # 0D (type G) = Enable Scheduled Event Report Mode

Cmd # 0E (type G) = Disable Scheduled Event Report Mode

Cmd # 0F (type G) = Enable Periodic Event Report Mode

Cmd # 10 (type G) = Disable Periodic Event Report Mode

Cmd # 11 (type G) = Enable Timer Report Mode

Cmd # 12 (type G) = Disable Timer Report Mode

Cmd # 13 (type G) = Enable Reporting Of Time And Date Along With Events

Cmd # 14 (type G) = Disable Reporting Of Time And Date Along With Events

Cmd # 15 (type G) = Enable Infrared Receive Function

Cmd # 16 (type G) = Disable Infrared Receive Function

Cmd # 17 (type G) = Enable Input Port Function

Cmd # 18 (type G) = Disable Input Port Function

Cmd # 19 (type G) = Enable X-10 Receive Function

Cmd # 1A (type G) = Disable X-10 Receive Function

Cmd # 1B (type G) = Turn User LED On

Cmd # 1C (type G) = Turn User LED Off

Cmd # 1D (type G) = Read Error Message

Cmd # 1E (type G) = Put Error Message in Result Value

Cmd # 1F (type G) = Clear Error Message

Cmd # 20 (type G) = Transmit Time and Date

Cmd # 21 (type G) = Transmit Controller Status

Response data is a value between 0 and 255. When converted to binary, the meaning of each bit can be found as follows:

bit 0 : 0 = Running, 1 = Halted

bit 1 : 0 = OK, 1 = controller error

bit 2 : 0 = TW-523 OK, 1 = no TW-523

bit 3 : 0 = using RAM config, 1 = using default config

bit 4 : 0 = self test passed, 1 = self test failed

bit 5 : 0 = clock OK, 1 = clock error

bit 6 : 0 = RAM checksum OK, 1 = RAM checksum error

bit 7 : not used

Cmd # 22 (type G) = Transmit Sunrise Time (HH MM format)

Cmd # 23 (type G) = Transmit Sunset Time (HH MM format)

Cmd # 24 (type G) = Execute Self-Test

Cmd # 25 = Not used
Cmd # 26 = Not used
Cmd # 27 (type G) = Set Baud Rate to 1200
Cmd # 28 (type G) = Set Baud Rate to 2400
Cmd # 29 (type G) = Set Baud Rate to 4800
Cmd # 2A (type G) = Set Baud Rate to 9600
Cmd # 2B (type G) = Set Baud Rate to 19200

NOTE: All baud rate changes take effect after the current serial transmissions are completed.

NOTE: Command numbers 2C through 44 are for internal use only. DO NOT USE THEM. DOING SO COULD CORRUPT THE SCHEDULE.

Cmd # 45 (type G) = Clear Scrolling Video Display RAM
Cmd # 46 (type G) = Enable Logging of Received X-10 Signals
Cmd # 47 (type G) = Disable Logging of Received X-10 Signals
Cmd # 48 (type G) = Enable Logging of Input Ports
Cmd # 49 (type G) = Disable Logging of Input Ports
Cmd # 4A (type G) = Enable Logging of Scheduled Events
Cmd # 4B (type G) = Disable Logging of Scheduled Events
Cmd # 4C (type G) = Enable Logging of Scheduled Thermostat Changes
Cmd # 4D (type G) = Disable Logging of Scheduled Thermostat Changes
Cmd # 4E (type G) = Enable Logging of Time and Date With Events
Cmd # 4F (type G) = Disable Logging of RTime and Date With Events

NOTE: Command numbers 50 through 63 are for internal use only. DO NOT USE THEM. DOING SO COULD CORRUPT THE SCHEDULE.

Cmd # 64 (type G) = Set Baud Rate to 57600 {Currently not implemented}
Cmd # 65 (type G) = Request Status Report of All X-10 Devices
Cmd # 66 (type G) = Request Status Report of All Flags
Cmd # 67 (type G) = Request Status Report of All Variables
Cmd # 68 (type G) = Request Status Report of All Input Ports
Cmd # 69 (type G) = Request Status Report of All Output Ports
Cmd # 6A (type G) = Request Status Report of Controller
Cmd # 6B (type G) = Request Status Report of All Analog Inputs
Cmd # 6C (type G) = Request Status Report of All Digital Temperature Sensors
Cmd # 6D (type G) = Enable Phone Event Report Mode
Cmd # 6E (type G) = Disable Phone Event Report Mode

NOTE: Command number 6F is for internal use only. DO NOT USE.

Cmd # 70 (type G) = Set Daylight Savings Time Mode (does NOT change clock)
Cmd # 71 (type G) = Clear Daylight Savings Time Mode (does NOT change clock)
Cmd # 72 (type G) = Enable Serial Thermostat Polling
Cmd # 73 (type G) = Disable Serial Thermostat Polling

EXAMPLES:

- To enable the master report mode, use command 03:

 ,G03
- To disable the IR receive function, use command 16:

,G16

- To turn the user LED on, use command 1B:

,G1B

- To transmit today's sunset time, use command 23:

,G23

VIDEO COMMANDS - 4 BYTES [Command type L (1C)]

- Command number (2 bytes):

Cmd # 00 (type L) = Not used

Cmd # 01 (type L) = Set Cursor Row To ##

Cmd # 02 (type L) = Set Cursor Column To ##

Cmd # 03 (type L) = Do Line Feed And Carriage Return

Cmd # 04 (type L) = Clear Screen

Cmd # 05 (type L) = Turn On Screen Display ##

value meaning:

00 = Off

01 = On

Cmd # 06 (type L) = Set OSD To ## Video Background

value meaning:

00 = External

01 = Internal

Cmd # 07 (type L) = ## Video Mode

value meaning:

00 = Start External

01 = Start Internal

02 = Stop Video

Cmd # 08 (type L) = Start Video Menu System

Cmd # 09 (type L) = Display Video Screen ##

NOTE: For a listing of the video screens and their numbers, see the file "Serial" located on the installation disks.

Cmd # 0A (type L) = Set Character Blink ##

value meaning:

00 = Off

01 = On

Cmd # 0B (type L) = Set Row _____ Character Size To _____

value meaning:

Upper 4 bits contains character size

Lower 4 bits contains row number

Cmd # 0C (type L) = Display Character Number ## On Screen

This command is used over the serial interface to display characters on the screen. Each command sends a single character. ## are the two hex digits representing the ASCII value of the desired character. For example:

<u>## Value</u>	<u>Character Displayed</u>
30	0
31	1
39	9
3A	:

41	A
5A	Z
61	a
7A	z

NOTE: The ASCII values for the numbers and letters can be found in the appendix to the user's manual. However, many characters can be displayed that do not have a similar ASCII equivalent. A table is provided following the video command examples that shows the ASCII values for these characters.

Cmd # 0D (type L) = Display All 3 Digits Of Variable Number ## On Screen

Cmd # 0E (type L) = Display Result Value On Screen

Cmd # 0F (type L) = Display Text String Number ## On Screen

Cmd # 10 (type L) = Set Video Switch To ##

value meaning:

00 = External Video Position

01 = OSD Video Position

Cmd # 11 (type L) = Initialize OSD Circuits

Cmd # 12 (type L) = Set Character Blink Ratio To ##

value meaning:

00 = Off

01 = 1:3

02 = 3:1

03 = 1:1

Cmd # 13 (type L) = Set Character Background Style To ##

value meaning:

00 = None

01 = Outline

02 = Square

03 = Solid

Cmd # 14 (type L) = Select Current Screen Background To ##

value meaning:

00 = Black

01 = Blue

02 = Green

03 = Cyan

04 = Red

05 = Magenta

06 = Yellow

07 = White

Cmd # 15 (type L) = Set Vertical Screen Offset To ##

Cmd # 16 (type L) = Set Horizontal Screen Offset To ##

Cmd # 17 = Not used

Cmd # 18 = Not used

Cmd # 19 = Not used

Cmd # 1A = Not used

Cmd # 1B (type L) = Set Default Screen Background To ##

value meaning:

00 = Black
01 = Blue
02 = Green
03 = Cyan
04 = Red
05 = Magenta
06 = Yellow
07 = White

Cmd # 1C (type L) = Display Current Time On Screen

Cmd # 1D (type L) = Display Current Date On Screen

Cmd # 1E (type L) = Display Last 2 Digits Of Variable Number ## On Screen

Cmd # 1F (type L) = Display Last Digit Of Variable Number ## On Screen

Cmd # 20 (type L) = Log Time to Event Log

Cmd # 21 (type L) = Log Text ##### to Event Log

(1 byte for number of characters, then the characters)

Cmd # 22 (type L) = DO NOT USE

Cmd # 23 (type L) = DO NOT USE

Cmd # 24 (type L) = Log Time and Date to Event Log

Cmd # 25 (type L) = Display Value of Timer ## on Screen

- 2 bytes for ## value (as described in above command list). For commands without an ## value, enter 2 zeros.

EXAMPLES:

- To set the cursor to row 2, use command 01:

,L0102

- To clear the screen, use command 04:

,L0400

- To start the internal video mode, use command 07:

,L0701

- To stop the video mode, use command 07:

,L0702

- To start the video menu system, use command 08:

,L0800

- To display video screen 07 (the X-10 listing screen), use command 09:

,L0907

- To set row 1 size to 2, use command 0B:
 ,L0B21
- To set row 1 size to 3, use command 0B:
 ,L0B31
- To set row 10d (Ah) size to 4, use command 0B:
 ,L0B4A
- To display the number "5", use command 0C with an ASCII value of 3555:
 ,L0C35
- To display the letter "A", use command 0C with an ASCII value of 41:
 ,L0C41
- To display the letter "a", use command 0C with an ASCII value of 61:
 ,L0C61
- To display a comma, use command 0C with an ASCII value of 2C:
 ,L0C2C
- To display the telephone symbol, use command 0C with an ASCII value of 26:
 ,L0C26
- To set the video switch to the on-screen display position, use command 10:
 ,L1001
- To set the current screen background color to red, use command 14:
 ,L1404
- To display the last 2 digits of the value of variable number 06, use command 1E:
 ,L1E06

VIDEO CHARACTERS AND THEIR ASCII VALUES

DISPLAYED CHARACTER	ASCII VALUE	DESCRIPTION
.	2E	Period
,	2C	Comma
:	3A	Colon
;	3B	Semicolon
?	3F	Question Mark
-	2D	Dash
/	2F	Slash
See note	20	Blank Space
<	3C	Left Angled Bracket
>	3E	Right Angled Bracket
See note	60	Snowman
See note	21	Left Arrow
See note	40	Up Arrow
See note	23	Right Arrow
See note	24	Down Arrow
See note	25	Small Solid Square
See note	7C	Large Solid Square
.	5E	Point (like a period, but shifted up and left slightly)
x	2A	Multiply Sign (same as small letter x, but shifted up slightly)
See note	26	Telephone
See note	28	Man
See note	29	Woman
See note	5B	Reverse Symbol
See note	5D	Forward Symbol
See note	5C	Sun
See note	22	Your guess is as good as ours!
See note	3D	Umbrella
See note	27	Musical Note
See note	5F	No Display

NOTE: These characters could not be drawn in this table. You'll have to display them and see for yourself what they look like.

VIDEO COMMANDS - 15 BYTES [Command type d (34)]

This command is used to write up to 14 characters to the video screen.

- 1 byte indicating the number of characters to display. Valid values are 1 - 14. The byte value must be in hex format ("1" for 1 character, "9" for 9 characters, "A" for 10 characters, "E" for 14 characters, etc.).
- 14 bytes, 1 for each character to display. You must transmit all 14 bytes, even if you specify a smaller number to display. Each byte should be the actual character to display ("1" for the number 1, "A" for the capital letter A, h for the small letter h, etc.)

EXAMPLES:

- To write (display) "Happy Birthday", send this command:

`,dEHappyBirthday`

- To write (display) "Hello", send this command:

`,d5HelloXXXXXXXXXX`

Note that the last 9 characters can be anything, as they aren't used since the command is for only 5 bytes.

SCHEDULED EVENT COMMANDS [Command type H (18)]

- Scheduled event ID Number (2 bytes).
- Command number (2 bytes):

Cmd # 00 (type H) = Unused
Cmd # 01 (type H) = Enable Scheduled Event
Cmd # 02 (type H) = Disable Scheduled Event
Cmd # 03 (type H) = Do Scheduled Event Now

EXAMPLE:

- To do scheduled event 40d (28h), use command 03:

 ,H2803

PERIODIC EVENT COMMANDS [Command type I (19)]

- Periodic event ID Number (2 bytes).
- Command number (2 bytes):

Cmd # 00 (type I) = Enable Periodic
Cmd # 01 (type I) = Disable Periodic Event
Cmd # 02 (type I) = Do Periodic Event Now
Cmd # 03 (type I) = Set Periodic Event Rate To: Every Time Thru Loop
Cmd # 04 (type I) = Set Periodic Event Rate To: Every 1 Minute
Cmd # 05 (type I) = Set Periodic Event Rate To: Every 5 Minutes
Cmd # 06 (type I) = Set Periodic Event Rate To: Every 15 Minutes
Cmd # 07 (type I) = Set Periodic Event Rate To: Every 1 Hour
Cmd # 08 (type I) = Set Periodic Event Rate To: Every 2 Hours
Cmd # 09 (type I) = Set Periodic Event Rate To: Every 4 Hours
Cmd # 0A (type I) = Set Periodic Event Rate To: Every 8 Hours

EXAMPLES:

- To do periodic event 15d (0Fh), use command 02:

 ,I0F02
- To set periodic event 60d (3Ch) to be performed every hour, use command 07:

 ,I3C07

CALLER ID COMMANDS [Command type R (22)]

- Command number (2 bytes):

Cmd # 00 (type R) = Execute Caller ID Macro
Cmd # 01 (type R) = Set Line 1, First 12 Characters To #####
Cmd # 02 (type R) = Set Line 1, Last 12 Characters To #####
Cmd # 03 (type R) = Set Line 2, First 12 Characters To #####
Cmd # 04 (type R) = Set Line 2, Last 12 Characters To #####
Cmd # 05 - DO NOT USE
Cmd # 06 - DO NOT USE
Cmd # 07 (type R) = Put All 48 Bytes of Caller ID Data Into User Text Area
Starting at Byte ##

- For commands 01 through 04, enter 12 bytes for "##" text.
For command 00, enter any 12 characters after the zeros.
For command 07, enter binary value of start byte number into first byte,
and any 11 characters after that.

EXAMPLES:

- To execute the caller ID macro, use command 00:

,R0000000000000000

Note: The last 12 characters can be anything.

- To set the first line of the caller ID screen to display "Your Mother's Calling", use commands 01 and 02 as follows:

,R01Your Mother'

and

,R02s Calling____ (_ represents blank space)

- To set the second line of the caller ID screen to display "555-1234", use commands 03 and 04 as follows:

,R03555-1234 (with 4 blanks at the end)

and

,R04 (12 blanks at the end)

THERMOSTAT COMMANDS [Command type c (63) and e (65)]

- Command number (2 bytes):

Cmd # 00 (type c) = Set Zone 1 to Currently Scheduled Temperature
Cmd # 01 (type c) = Set Zone 2 to Currently Scheduled Temperature
Cmd # 02 (type e) = Set Zone ## System Mode to Off
Cmd # 03 (type e) = Set Zone ## System Mode to Heat
Cmd # 04 (type e) = Set Zone ## System Mode to Cool
Cmd # 05 (type e) = Set Zone ## System Mode to Auto
Cmd # 06 (type e) = Set Zone ## Fan Mode to On
Cmd # 07 (type e) = Set Zone ## Fan Mode to Auto
Cmd # 08 (type e) = Set Zone ## Control Mode to Run (Auto)
Cmd # 09 (type e) = Set Zone ## Control Mode to Hold (Off)
Cmd # 0A (type e) = Set Zone ## Temperature Setting to ##
Cmd # 0B (type e) = Set Zone ## Temperature Back
Cmd # 0C (type e) = Raise Zone ## Temperature 1 Degree
Cmd # 0D (type e) = Lower Zone ## Temperature 1 Degree
Cmd # 0E (type e) = Initialize Zone ## Thermostat
Cmd # 0F (type e) = DO NOT USE
Cmd # 10 (type e) = DO NOT USE
Cmd # 11 (type e) = DO NOT USE
Cmd # 12 (type e) = Request Zone ## Temperature Report
Cmd # 13 (type e) = Request Zone ## Temperature Setting Report
Cmd # 14 (type e) = Request Zone ## System Mode Report
Cmd # 15 (type e) = Request Zone ## Fan Mode Report
Cmd # 16 (type e) = Refresh Zone ## System Mode
Cmd # 17 (type e) = Refresh Zone ## Fan Mode
Cmd # 18 (type e) = Refresh Zone ## Temperature Setting

- For "type e" commands, 2 bytes for zone number ## minus 1. Zones are numbered 1 thru 16, but are stored as 0 thru 15.
- For "type e" commands, 2 bytes for new setting:
 - For command #0A, enter new temperature setting (0-99).
 - For other commands, enter two zeros.

EXAMPLES:

- To set the zone 1 thermostat to the currently scheduled temperature, use command 00:

,c00

- To set the zone 2 thermostat to the heat mode, use command 03:

,e030100

- To set the zone 6 thermostat fan to the auto mode, use command 07:

,e070500

- To set the zone 3 thermostat temperature setting to 75 degrees (6B hex), use command 0A:

,e0A026B

MULTIFUNCTION EXPANSION BOARD COMMANDS [Command type Q (51)]

- Command number (2 bytes):

Cmd # 00 = Read Analog Input ##

Value read in ranges from 0 to 255 (00h to FFh), which represents 0 to 5 volts. To determine the actual voltage from the value read, multiply by 0.01961V. The table below shows the relationship:

Value Read	Voltage
0	0.000
1	0.020
2	0.040
51	1.000
128	2.500
254	4.980
255	5.000

Cmd # 01 = Check For Digital Temperature Sensor At Port ##
(looks for presence pulse in single-drop mode)

Cmd # 02 = Read Digital Temperature Sensor At Port ##
(reads temperature in single-drop mode)

Cmd # 03 = Read 8-Bit Expansion Port ##

Each of the three 8-bit ports on the expansion board(s) can be read as a group. The result is read as an 8-bit value with one bit representing the current state of each of the eight lines in the port. A 0 (zero) bit indicates the corresponding input is low, while a 1 (one) bit indicates it is high. The highest port number is listed first. For example, if a command to read a port returns the value 147d, it is interpreted as follows:

The value 147d (93h) is 10010011 in binary. The eight inputs on the port are as follows:

Input signal 1 = 1 = high
Input signal 2 = 1 = high
Input signal 3 = 0 = low
Input signal 4 = 0 = low
Input signal 5 = 1 = high
Input signal 6 = 0 = low
Input signal 7 = 0 = low
Input signal 8 = 1 = high

Cmd # 04 = Select Infrared Transmit Zones ##

Cmd # 05 = Read Digital Temperature Sensor Address At Port ##
(reads the sensor's 8-byte address in single-drop mode)

Cmd # 06 = Read Digital Temperature Sensor ##
 (reads temperature from the sensor in multi-drop mode)
 Cmd # 07 = Report Digital Temperature Sensor ## Value Previously Read
 (reports the most recently read temperature in multi-drop
 mode; this command does not actually read the temperature
 sensor)

- Two more bytes must be entered. The data depends on the type of command:
 - For command 00, enter 2 bytes for the analog input ID #. The analog input ports are labeled as 1 thru 8 on each expansion board. However, the port ID numbers to enter for this command range from 0 to 15 (00 to 0F hex), as shown below:

```

Board 1, analog input 1 = 00
Board 1, analog input 2 = 01
Board 1, analog input 3 = 02
Board 1, analog input 4 = 03
Board 1, analog input 5 = 04
Board 1, analog input 6 = 05
Board 1, analog input 7 = 06
Board 1, analog input 8 = 07
Board 2, analog input 1 = 08
Board 2, analog input 2 = 09
Board 2, analog input 3 = 0A
Board 2, analog input 4 = 0B
Board 2, analog input 5 = 0C
Board 2, analog input 6 = 0D
Board 2, analog input 7 = 0E
Board 2, analog input 8 = 0F

```

- For commands 01, 02 and 05, enter 2 bytes for the port ID # the sensor is connected to. The digital temperature sensors connect to the analog input ports, so you use the analog input port number (00 to 0F hex), as shown above for command 00.
- For command 03, enter 2 bytes for the hardware address of the 8-bit port. This command reads all eight bits of the port. The port hardware addresses are as follows:

```

Board 1, digital input port 1 (called port D) = 00
Board 1, digital input port 2 (called port E) = 01
Board 1, digital input port 3 (called port F) = 02
Board 2, digital input port 1 (called port G) = 04
Board 2, digital input port 2 (called port H) = 05
Board 2, digital input port 3 (called port I) = 06

```

Note: The letters D thru I are a shorthand reference for each I/O port. Ports A thru C are located on the HomeVision controller.

- For command 04, enter 2 bytes to select the active infrared zones. Each of the eight zones is represented by one bit (a zero value means

the zone is off, a one value means the zone is on). The eight bits are converted to two hex bytes. Here are some examples:

<u>SELECTED ZONES</u>	<u>BINARY VALUE</u>	<u>HEX VALUE</u>
1	00000001	01
2	00000010	02
3	00000100	04
4	00001000	08
5	00010000	10
6	00100000	20
7	01000000	40
8	10000000	80
1, 2	00000011	03
1, 2, 3, 4	00001111	0F
2 through 8	11111110	FE
All 8 zones	11111111	FF

- For commands 06 and 07, enter the digital temperature sensor ID #. The ID number ranges from 00 to 3F hex. The sensor must be connected to analog input 1 on expansion board #1. Multiple sensors can be connected to the same port.

RESPONSE FORMAT:

Commands 00 and 03 report the value read back to the computer as follow:

- Command 00 (reading analog value) transmits this serial message:

21 Cmd: 00 ID VV Done(Carriage Return)(Line Feed)(SOH)

Where: 21 is the command ID # for all expansion port commands.
 00 is the command # for reading an analog input.
 ID contains the ID # of the analog input being reported.
 VV contains the analog input value.

See the examples below.

- Command 03 (reading expansion port) transmits this serial message:

21 Cmd: 03 ID VV Done(Carriage Return)(Line Feed)(SOH)

Where: 21 is the command ID # for all expansion port commands.
 03 is the command # for reading an expansion port.
 ID contains the ID # of the expansion port being reported.
 VV contains the expansion port value.

See the examples below.

EXAMPLES:

- To read analog input 2 on board 1, use command 00:

,Q0001

If the analog value is 45h, the controller response will be:

21 Cmd: 00 01 45 Done(Carriage Return)(Line Feed)(SOH)

- To read analog input 8 on board 2:

,Q000F

If the analog value is ABh, the controller response will be:

21 Cmd: 00 0F AB Done(Carriage Return)(Line Feed)(SOH)

- To read digital input port 1 on board 1, use command 03:

,Q0300

If the eight inputs are all high (the port value is FFh), the controller response will be:

21 Cmd: 03 00 FF Done(Carriage Return)(Line Feed)(SOH)

- To read digital input port 2 on board 2:

,Q0305

If the eight inputs are all low (the port value is 00h), the controller response will be:

21 Cmd: 03 05 00 Done(Carriage Return)(Line Feed)(SOH)

- To select IR transmit zone 1, use command 04:

,Q0401

- To select IR transmit zone 7:

,Q0440

- To select IR transmit zone 1, 2, 3, 4, and 7:

,Q044F

- To select all IR transmit zones:

,Q04FF

SYSTEM VARIABLE COMMANDS [Command type ^ (5E)]

NOTE: "System variables" are variables built into the controller which are used for storing function-specific data. Currently, the only variables related to weather information are available. Some of these variables are used for displaying information on the new weather conditions video screen. Other weather variables are not displayed, but can be used for other purposes. In future versions, we will be adding more commands to take advantage of these system variables.

- Command number (2 bytes):

Cmd # 00 (type ^) = Set System Variable XX Equal To Constant Value ##
Cmd # 01 (type ^) = Set System Variable XX Equal To Variable ##
Cmd # 02 (type ^) = Set Variable ## Equal To System Variable XX
Cmd # 03 (type ^) = Read Value Of System Variable XX Over Serial Interface

- System variable ID number (2 bytes). Each system variable has a unique ID number, as shown below:

00 = Current inside temperature (see note 1)
01 = Today's low inside temperature (see note 1)
02 = Today's high inside temperature (see note 1)
03 = Current outside temperature (see note 1)
04 = Today's low outside temperature (see note 1)
05 = Today's high outside temperature (see note 1)

NOTE 1: The temperatures can range from -99 to +154 degrees in increments of 1 degree. Temperatures are stored as actual temperature plus 100. HomeVision subtracts 100, then displays it, with two exceptions. On the weather conditions video screen, the value of 0 is displayed as -99 degrees instead of -100. A value of 255 is displayed as blank spaces instead of a number (this is so that temperatures that aren't available can be left blank). Here are some examples:

<u>STORED VALUE</u>	<u>DISPLAYED VALUE</u>
0	-99
1	-99
2	-98
99	-1
100	0
101	1
175	75
254	154
255	

06 = Current inside relative humidity (see note 2)
07 = Today's low inside relative humidity (see note 2)
08 = Today's high inside relative humidity (see note 2)
09 = Current outside relative humidity (see note 2)
0A = Today's low outside relative humidity (see note 2)
0B = Today's high outside relative humidity (see note 2)

NOTE 2: Relative humidities can range from 0 to 100 in increments of 1%. A value of 255 is displayed as blank spaces.

0C = Current barometric pressure - inches (see note 3)

0D = Current barometric pressure - hundredths (see note 3)

NOTE 3: Barometric pressure requires two variables for storage. Each variable can range from 0 to 99 in increments of 1%. All other values are displayed as blank spaces.

0E = Barometric Pressure Direction (0 = steady, 1 = falling, 2 = rising, anything else = blank spaces are displayed)

0F = Wind direction digit 1 (see note 4)

10 = Wind direction digits 2-3 (see note 4)

NOTE 4: Wind direction (in degrees from 0 to 359) requires two variables for storage. The first variable holds the first digit, which can range from 0 to 3. The second variable holds the second and third digits, which can range from 00 to 99 (in hex format). If the first variable is >3, then blanks will be displayed on the weather conditions TV screen.

11 = Current wind speed (see note 5)

12 = Today's low wind speed (see note 5)

13 = Today's high wind speed (see note 5)

NOTE 5: Wind speed can range from 0 to 254 in increments of 1. A value of 255 is displayed as blank spaces.

14 = Today's rainfall - inches (see note 6)

15 = Today's rainfall - hundredths (see note 6)

16 = Rainfall this week - inches (see note 6)

17 = Rainfall this week - hundredths (see note 6)

18 = Rainfall this month - inches (see note 6)

19 = Rainfall this month - hundredths (see note 6)

1A = Rainfall this year - inches (see note 6)

1B = Rainfall this year - hundredths (see note 6)

NOTE 6: Four different rainfall amounts can be tracked, with each requiring two variables for storage. The "hundredths" values can range from 0 to 99 in hex format. Only for the variable "Today's rainfall - inches", a value of >99 is displayed as blank spaces instead of the digits. The other rainfall "inches" amounts are not included on the weather conditions screen and can range from 0 to 255 inches.

- 2 bytes for "##" value required for the command:
 - For command 00, enter the constant value in hex format (from 00 to FF).
 - For commands 01 and 02, enter the variable ID # you want to use (in hex format). Note that this refers to the variables you created in your schedule.
 - For command 03, enter two zeros (00).

EXAMPLE:

- To set the system variable "Current outside temperature" to indicate 65 degrees (41 hex), use command 00:

,^000341

- To set the system variable "Today's low outside relative humidity" to the value contained in your variable #3, use command 01:

,^010A03

- To set your variable #8 equal to the the system variable "Current wind speed", use command 02:

,^021108

- To set the current barometric pressure to 30.08 inches and falling, use these commands:

,^000C1E	{30 inches = 1E hex}
,^000D08	{8 hundredths = 08 hex}
,^000E01	{01 = falling}

- To set the current wind direction to due south (180 degrees), use these commands:

,^000F01	{"1" first digit = 01 hex}
,^001050	{"80" last 2 digits = 50 hex}

- To set the total rainfall this year to 12.90 inches, use these commands:

,^001A0C	{12 inches = 0C hex}
,^001B5A	{90 hundredths = 5A hex}

DATA LOG COMMANDS [Command type f (66)]

- Command number (2 bytes):

Cmd # 00 (type f) = Write Variable ## To Data Log
Cmd # 01 (type f) = Reset Data Log Pointer To Start Of Log
Cmd # 02 (type f) = Clear Data Log
Cmd # 03 (type f) = Transmit Complete Data Log In ASCII Format
Cmd # 04 (type f) = Transmit Complete Data Log In Binary Format
Cmd # 05 (type f) = Transmit Partial Data Log In ASCII Format
Cmd # 06 (type f) = Transmit Partial Data Log In Binary Format
Cmd # 07 (type f) = Transmit Data Log Pointer Value (Location)

- 10 data bytes, depending on command number:

- For command 00, enter 2 bytes for the variable ID #. The remaining eight bytes should be zeros.
- For commands 01, 02 and 07, enter 10 zeros.
- For commands 03 and 04, enter two bytes to indicate the number of data log bytes to transmit on each line. The number can range from 01h to FFh (1 to 255 decimal). The remaining eight bytes should be zeros.
- For commands 05 and 06, enter:
 - o Two bytes indicating the number of data log bytes to transmit on each line. The number can range from 01h to FFh (1 to 255 decimal).
 - o Four bytes indicating the byte number in the data log to begin transmitting from. This can range from 0001h (the first byte) to FFFFh.
 - o Four bytes indicating the number of lines of data to transmit. This can range from 0001h (the first byte) to FFFFh. Note that you can request more data than is actually in the log. In this case, non-data log memory locations will be read and the data will be meaningless.

EXAMPLE:

- To write the value of variable #26 (1Ah) to the log, use command 00:

 ,f001A000000000
- After reading the data log, use command 02 to clear the log, then command 01 to reset the pointer to the start of the log:

 ,f0200000000000
 ,f0100000000000

- To transmit the complete data log in ASCII format using 8 bytes per line, use command 03:

,f0308000000000

- To transmit 25 (0019h) lines of text, each with 8 bytes of data, starting at the first byte in the data log (byte number 0001) in ASCII format, use command 05:

,f050800010019

- To transmit 1000 (03E8h) lines of text, each with 8 bytes of data, starting at byte 8000 (1F40h) in the data log in binary format, use command 06:

,f06081F4003E8

- To transmit the data log pointer value (which is equivalent to the number of bytes used in the log) use command 07:

,f0700000000000

SIMULATE INFRARED REMOTE KEYPRESS [Command type J (4A)]

These commands simulate pressing an IR remote button to control the video menu system. When you send a command, the controller acts on it as if it received the corresponding signal from an IR remote. This feature is provided to allow voice control of the video menu system. You can setup voice recognition software to send the serial command to HomeVision when you speak the corresponding word(s).

This command is sent as a single character representing each of the 16 remote buttons used for the video menu system:

<u>CHARACTER</u>	<u>REMOTE BUTTON</u>
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	Enter
B	Channel Up
C	Channel Down
D	Volume Up
E	Volume Down
F	Cancel

EXAMPLE:

- When the video system is off, the Channel Up button starts it. You can simulate this with this command:

,JB

- From page 1 of the video menu system, the number 1 selects the X-10 menu entry. Simulate this by:

,J1

- From the X-10 signal list, the number 1 selects X-10 signal A-1. Simulate this by:

,J1

- From the X-10 signal list, the Channel Down button exits the menu system. Simulate this by:

,JC

MISCELLANEOUS COMMAND - 15 BYTES [Command type [(5B)]

- Command number: 1 ASCII character. There is currently only 1 command available.

Cmd # 0 (ASCII zero) = Write 12 characters to RAM video screen number ##

Following the command number are:

- 1 binary byte for the RAM video screen ID #. The number ranges from 100 up to the last screen (a maximum value of 131).
- 1 ASCII character to indicate where the text should be written on the screen. The capital letters A through V are used as follows:

```
A = First half of row 1
B = Last half of row 1
C = First half of row 2
D = Last half of row 2
E = First half of row 3
F = Last half of row 3
.
.
.
U = First half of row 11
V = Last half of row 11
```

- 12 bytes representing the 12 characters to write.

EXAMPLE:

- Let's assume you want to write a screen like this:

```
CURRENT WEATHER REPORT
-----
Partially cloudy with a
40% chance of showers
late this afternoon.

Highs in the low 80's.
Lows near 70 tonight.

Turning cooler tomorrow.
-----
```

To do this, use a series of 22 commands to fill up the entire screen. It's difficult to show the commands because one of the characters is a binary value that doesn't have printable ASCII equivalents. The first command would write the first 12 characters of the first row. The command would be:

```
,[0*ACURRENT WEAT
```

The * represents the binary character which can't be printed. It is the desired video screen number (100 for the first screen).

The complete series of 22 commands would look like this:

,[0*ACURRENT WEAT	
,[0*BHER REPORT	{note 2 blanks spaces at the end}
,[0*C-----	
,[0*D-----	
,[0*EPartially cl	
,[0*Foudy with a	{note blank space at the end}
,[0*G40% chance o	
,[0*Hf showers	{note 3 blanks spaces at the end}
,[0*Ilate this af	
,[0*Jternoon.	{note 4 blanks spaces at the end}
,[0*K	{note 12 blanks spaces at the end}
,[0*L	{note 12 blanks spaces at the end}
,[0*MHighs in the	
,[0*N low 80's.	{note 2 blanks spaces at the end}
,[0*OLows near 70	
,[0*P tonight.	{note 3 blanks spaces at the end}
,[0*Q	{note 12 blanks spaces at the end}
,[0*R	{note 12 blanks spaces at the end}
,[0*STurning cool	
,[0*Ter tomorrow.	
,[0*U-----	
,[0*V-----	

SECURITY SYSTEM COMMANDS [Command type 1 (6C)]

- Command number (2 bytes):

Cmd # 00 (type 1) = Arm Partitions ##### in Home Mode
Cmd # 01 (type 1) = Arm Partitions ##### in Away Mode
Cmd # 02 (type 1) = Disarm Partitions #####
Cmd # 03 (type 1) = Toggle Zone ## Bypass Mode
Cmd # 04 (type 1) = Toggle Partitions ## Chime Mode
Cmd # 05 (type 1) = Toggle Partitions ## Instant Mode
Cmd # 06 (type 1) = Sound Partitions ##### Fire Panic
Cmd # 07 (type 1) = Sound Partitions ##### Medical Panic
Cmd # 08 (type 1) = Sound Partitions ##### Police Panic
Cmd # 09 (type 1) = DO NOT USE
Cmd # 0A (type 1) = Request Partitions Snapshot Status
Cmd # 0B (type 1) = Request Zone ## Snapshot Status
Cmd # 0C (type 1) = Turn Partitions ##### Chime On
Cmd # 0D (type 1) = Turn Partitions ##### Chime Off

- 2 bytes for "##" value required for the command:
 - For commands 0-2, 4-8, C, and D, enter the partition number in hex format (from 00 to FF). The command can control multiple partitions by setting the appropriate bits:

Bit 0 set if command should be performed on partition 1
Bit 1 set if command should be performed on partition 2
Bit 2 set if command should be performed on partition 3
Bit 3 set if command should be performed on partition 4
Bit 4 set if command should be performed on partition 5
Bit 5 set if command should be performed on partition 6
Bit 6 set if command should be performed on partition 7
Bit 7 set if command should be performed on partition 8
 - For commands 3 and B, enter the zone number minus one in hex format (from 00 to FF). In other words, zone 1 is "00", zone 2 is "01", etc.
 - For command A, enter two zeros (00).

EXAMPLE:

- To arm partition 1 in the HOME mode:

,10001
- To disarm partitions 1, 2, and 3:

,10207
- To turn all partitions (1-8) chime mode off:

,10DFF

- To toggle the zone 8 bypass mode:
 ,10307