HomeVision-Pro 3.5

Custom Solutions, Inc.

Copyright 2004-2006

SUMMARY OF CHANGES – 3.5

This is a new version of the PC software and PROM. It makes several changes:

- The automatic daylight savings time feature uses the new start and stop dates that begin in 2007 for the United States. It now starts the second Sunday in March and ends the first Sunday in November. The dates for European users are unchanged.

- The "auto report feature" has been improved. This feature is enabled on the PC Communications tab of the Controller Settings screen (under the Configure menu). This new version adds automatic reporting of HVAC, Security System, and Custom Lighting changes. It also improves the reporting for Flags, Variables, and Analog Inputs. Refer to the "Auto Report" file for more details.

- Added a web browser to the software. It is accessed under the "Control" menu. It defaults to showing the built-in web pages (if you have this option enabled using the Web Server screen under the "Configure" menu). This is a simple browser mainly intended to provide quick access to the HomeVision web pages (although it can display other web pages as well).

- Modified the web pages to use "Cascading Style Sheets" to make it easier to modify the look of all the web pages at once. This new version of the software requires a new set of web page files installed in the HTML directory.

- Supports a web page to display status of and control custom lights. Refer to the Web Server chapter of the version 3.5 owners manual for details on customizing the web page. As a brief example, assume you want the web page index to link to the custom lighting web page. By default, the HVIndex.htm page has lines like the following for links to X10, macros, and other pages:

      <A HREF="HVTableX10.htm" target="main-frame">X-10</A>
      <A HREF="HVTableMacro.htm" target="main-frame">Macros</A>

  To add a link to the custom lighting web page, add this line:

      <A HREF="HVTableLight.htm" target="main-frame">Lights</A>

- Modified the "Run Program" command that can be transmitted to a PC serial port. The software will no longer convert the command to uppercase letters. This change allows parameters to be passed to the program without being converted to uppercase (as previous software versions did).

- The "Export" function under the file menu now includes custom lights in the ".HVX" file that it generates.

- Made software change to correct a bug with the data log that can occur if the controller's memory usage is very high.

- Added a "controller" command to reset the HomeVision-Pro unit. The effect is similar (but not identical) to a power cycle, and the controller will transmit the normal startup message after the reset is complete. Note that when this command is performed in a schedule, the controller is immediately reset and will not perform any actions that may have followed the reset command in the schedule. After the reset is complete, the controller resumes normal operations. Most users would never use this command – it's provided for the rare cases where a user desires a way to reset the controller without cycling power.

- Added "controller" commands to reinitialize the HomeVision-Pro serial port 3 and 4 chips. These commands cause the controller chips to get reset to the correct baud rate and duplex mode. Most users would never use these commands – they're provided for the rare cases where a user desires a way to reinitialize the ports without doing a complete controller reset.

SUMMARY OF CHANGES – 3.43a

New PC software that corrects a bug in the Thermostat Configuration Screen that could cause it to crash.

SUMMARY OF CHANGES – 3.43

This is a new version of the PC software and PROM. It makes several changes:

- A few text fields in the software were widened to show more characters when the user has the Windows font size set to the "large" or "extra large".
- Made slight improvement to X-10 transmission collision avoidance. With the new design, if another device transmits an X-10 signal very shortly before HomeVision-Pro is ready to transmit a signal, there is less chance of a collision.
- Modified the digital temperature sensor reading to work with the new, slower DS18S20 sensors. Previously, there could be occasional errors reading these sensors if the HomeVision-Pro schedule file ran extremely fast.
- "X-10 sequences" can now use "wildcards" to match a received X-10 signal. When entering X-10 sequences, there are three ways to use the wildcards:

    1. To match any unit or function code with a particular house code: Select "House / Unit Code", select the desired house code in the House Code list box, then select the "?" character in the Unit Code list box.
    2. To match any house code with a particular unit code: Select "House / Unit Code", select the "?" character in the House Code list box, then select the desired unit code in the Unit Code list box.
    3. To match any house code with a particular function code: Select "House / Function Code", select the "?" character in the House Code list box, then select the desired function code in the Function Code list box.

SUMMARY OF CHANGES – 3.42c

This is a new version of the PC software. It corrects one bug in the previous software related to how it exports data to an "HVConfiguration.txt" file. This file is only used with the Macromedia Flash movie that can be used to control HomeVision-Pro.

SUMMARY OF CHANGES – 3.42

NOTE: Schedule files saved with the version 3.42 software cannot be opened by older versions. Attempts to do so may cause the older software to lock up.

This is a new version of the PC software and PROM. It adds several new features:

- Support for "custom lighting" commands. Some users would like to use HomeVision-Pro to control lighting systems that don't use the X-10 protocol. To facilitate this, HomeVision-Pro now supports what we call "custom lighting" commands. The HomeVision-Pro controller does not include built-in control of other lighting systems; i.e., there are not specific commands for lighting systems such as Lutron, C-Bus, CentraLite, etc. Instead, the user (or someone else) must either write a HomeVision-Pro macro or external PC software that can control the lighting system. Once this is done, HomeVision-Pro will provide the following capabilities:

    - Define and name up to 255 lights.
    - Track light levels in an internal state table.
    - Enter commands in the schedule for these lights, such as "Light 3 (Porch light) On", "Light 7 (Kitchen light) to level 50", and "Light 45 (Master bath) select scene 6".
    - Use the HomeVision software, web browser, or other PC software to control the lights and read status.

- View a TV screen to control the lights and show their status.
- Check the current status in If-Then conditions.

Refer to the file "Custom Lighting" installed with the HomeVision-Pro software for details on how this works.

- The software now supports a serial port baud rate of 57,600. The 3.3 PROMs already supported this baud rate on the main serial, but the PC software didn't. We've seen that on some slower PCs, communications could be erratic at 57,600 baud. We recommend you test it by downloading your schedule several times to make sure your PC is fast enough.

- The standard HomeVision-Pro serial communications protocol (as described in the "Serial Protocol" file) can now be used over serial ports 3 and 4. Previously, only serial port 1 supported this.

- Added more commands for reading data received from a serial port. The new commands can read hex characters, and can read character locations specified by a variable value.

- Added commands to allow calculating a checksum value on data transmitted out a serial port. These may be useful if you need HomeVision-Pro to communicate with external devices that require a checksum.

- Added ability to detect and take action when *any* "standard" infrared signal is received. Normally, the controller is limited to 255 IR signals for which you can assign actions. This new feature allows you to create a periodic event that can detect when any of the 65,536 possible standard IR signals is received. You can then check the signal's device and key code to decide what action to take.

- Added infrared commands to allow transmission of signals loaded into a "test" area. This allows PC software to load a signal into the controller and transmit it, thereby overcoming the controller limit of 255 learned IR signals. Refer to the file "Using Extra IR Signals" installed with the HomeVision-Pro software for details on how this works.

- Allow both RCS X-10 and serial thermostats to be used at the same time.

- Added option for HomeVision-Pro serial ports 3 and 4 to receive any binary data. When disabled (the default setting), the ports will ignore binary zeros (the NULL character) and binary tens (line feeds). This works fine for most uses where ASCII data is received. However, if the data is in binary format and contains zeros or tens, this option must be enabled.

- Added infrared commands to allow repeated transmission of IR signals.

CHANGE DETAILS – VERSION 3.42

ALL SERIAL PORTS SUPPORT STANDARD COMMUNICATIONS PROTOCOL

The standard HomeVision serial communications protocol (as described in the "Serial Protocol" file) can now be used over serial ports 3 and 4. Previously, only HomeVision-Pro serial port 1 supported this.

Previously, 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, the other 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 other ports.

Before a 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 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.

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 use this method instead of the "event reporting" messages mentioned above. Thus, these programs should work OK when connected to an add-on serial port.

NEW SERIAL RECEPTION COMMANDS

Seven commands have been added for reading data received from a serial port. They are described below.

```
Put Value Of Received Hex Character # Into Result Value
Put Value Of 2 Received Hex Characters ## Into Result Value
```

These two commands look for a hex value at the specified location(s) in the serial data string, convert it to a number, and put it into the system variable "Result Value". After the value is put in "Result Value", you can use variable commands to move it to a variable and perform other operations on it.

If the specified locations are not valid numbers, Result Value will be set equal to 255 to indicate an error. The difference between these commands is how many characters are used to determine the value:

- The first command looks at a single character that must be a number between 0 and 9, a letter between a and f, or a letter between A and F. The result will be a value between 0 and 15.

- The second command looks at two consecutive characters, both of which must be a number between 0 and 9, a letter between a and f, or a letter between A and F. The result will be a value between 0 and 255.

```
Put Value Of Received Hex Character At Variable # Into Result Value
Put Value Of 2 Received Hex Characters At Variable # Into Result Value
```

These two commands are similar to the two commands above. The difference is that instead of specifying a fixed character location in the serial message, you specify a variable. When the command runs, it will use the variable's value as the character location to start reading from.

```
Put Value Of Received Character At Variable # Into Result Value
Put Value Of 2 Received Characters At Variable # Into Result Value
Put Value Of 3 Received Characters At Variable # Into Result Value
```

These three commands are similar to three existing commands that read in decimal values. The difference is that instead of specifying a fixed character location in the serial message, you specify a variable. When the command runs, it will use the variable's value as the character location to start reading from.

SERIAL TRANSMISSION CHECKSUM COMMANDS

There are two commands to allow calculating a checksum value on data transmitted out a serial port:

```
Clear Transmit Checksum Value
Put Transmit Checksum Value Into Result Value
```

These may be useful if you need HomeVision to communicate with external devices that require a checksum. HomeVision automatically calculates a running checksum on all data you send out a serial port using one of the serial port commands. If you need to calculate a checksum on the data you transmit, you would first use the "Clear Transmit Checksum Value" prior to starting the transmission. This resets the checksum value to zero. You would then transmit the data that must be checksummed. The controller will automatically calculate a running checksum on the data. After you've finished transmitting the data, use the "Put Transmit Checksum Value Into Result Value" command. As the name implies, this will put the value of the checksum into the system variable "Result Value". You can then use a variable command to put Result Value into one of your own variables, at which point you can do anything you like with it. Most commonly, you would transmit this value either as a single binary character or two hex characters.

Note several things about the checksum process:

- The controller maintains a single checksum value – there is *not* one for each serial port. Therefore, you must send all the data for a port and obtain its checksum before you transmit out another port. You don't have to worry about HomeVision automatically transmitting data out a serial port, as they won't be added to the checksum.

- The checksum is an 8-bit value, and is therefore limited to the range of 0 to 255. If the sum exceeds 255, only the lower 8 bits are kept. In effect, the total sum is divided by 256, and the remainder is the checksum.
- The checksum is calculated by summing the raw binary values of the transmitted data. For example, the capital letter "A" has a value of 65, and the number "1" has a value of 49. See Appendix B of the HomeVision manual for a table of the values of the standard ASCII characters.

As an example, assume you transmit "ABC123", followed by a carriage return and line feed. Here are the values that will be summed up:

```
A  = 65 decimal (41 hex)
B  = 66 decimal (42 hex)
C  = 67 decimal (43 hex)
1  = 49 decimal (31 hex)
2  = 50 decimal (32 hex)
3  = 51 decimal (33 hex)
CR = 13 decimal (0D hex)
LF = 10 decimal (0A hex)
```

The total sum is 371 decimal (173 hex). The checksum value is 115 decimal (73 hex). Thus, the "Put Transmit Checksum Value Into Result Value" command will put the value of 115 into Result Value.

RECEIVING ANY STANDARD INFRARED SIGNAL

Normally, the controller is limited to 255 IR signals for which you can assign actions (using the IR Signal Summary Screen under the Objects/Events menu). This new feature allows you to create a periodic event that can detect when any of the 65,536 possible standard IR signals is received. You can then check the signal's device and key code to decide what action to take. This is intended for advanced users, as it requires using special "memory access commands" that are normally hidden from users. Here's how this process works:

- Whenever the controller receives an IR signal, it will store up to three bytes of information about the signal in RAM.
- You create a periodic event that runs every loop. The periodic event uses the memory access commands to read the signal information into three of your own variables. You can then check these variables in If-Then statements and take action.
- After all periodic events have run, the controller will reset the signal information. This enables you to detect when a new signal is received.

Following is an example of the code that goes in the periodic event:

```
Memory access: Read data from address EF59h into variable #28 (IRRX signal type)
Memory access: Read data from address EF5Ah into variable #29 (IRRX device code)
Memory access: Read data from address EF5Bh into variable #30 (IRRX key code)
If
    Var #28 (IRRX signal type)  = 1
Then
    Serial port 1 transmit: "IRRX: "
    Serial port 1 transmit: Variable #28 (IRRX signal type) value as 2 hex bytes
    Serial port 1 transmit: " "
    Serial port 1 transmit: Variable #29 (IRRX device code) value as 2 hex bytes
    Serial port 1 transmit: " "
    Serial port 1 transmit: Variable #30 (IRRX key code) value as 2 hex bytes
    Serial port 1 transmit: Carriage return and line feed
End If
```

The first three commands are the memory access commands that read the three bytes containing the IR signal information into your variables (numbers 28, 29, and 30 in this example). If a standard IR signal was received, the "EF59" variable will contain the value of one, so check for this value. You can then take whatever actions

you want. In this example, we simply transmit the device and key codes out the serial port. Typically, users would check the device and key code in additional If-Then statements and take appropriate actions.

The "EF59" variable will contain one of four possible values:

> 0: Means no new IR signal was received.
> 1: Means a standard IR signal was received. The next two bytes will contain the device and key code.
> 2: Means a learned IR signal was received. The next byte will contain the ID # of the IR signal it matches.
> 3: Means an unknown IR signal was received. This will also happen if IR noise is received.

In order to enter the "memory access commands" in your schedule, this capability must be enabled. To do this, you must edit the Windows registry on your PC. This can be done using RegEdit. Open the registry and then open this path:

> HKEY_CURRENT_USER / Software / VB and VBA Program Settings / HomeVision

Under the "Miscellaneous" section, look for an entry with the name "MemoryAcess" and change its data value to "True". If it's not there, add an entry (using Edit/New/String Value) with the name "MemoryAcess" and set its data value to "True". The next time to start the HomeVision software, memory access commands will be enabled. You access them from the Actions Entry Screen via the "Other" button. Note that when entering the address to read or write, you enter two bytes in decimal format, but the command will display the address in hexadecimal format. For example, the hex address "EF59" shown in the example is entered as "239" for the Most Significant Byte and "89" for the Least Significant Byte.

REPEATED INFRARED SIGNAL TRANSMISSIONS

NOTE: This function is still in the Beta testing phase. If you wish to use it, please contact us so we can discuss some issues with it. The HomeVision-Pro PROM supports this function, but the software may require some slight modifications for your particular application.

Previously, the primary way to transmit an IR signal multiple times was with a command requiring you to specify the number of transmissions. Now, it is possible to command the controller to start transmitting a signal repeatedly, and then later perform another command to stop the transmission. Although these commands can be performed in your schedule, they were primarily intended to be sent to the controller via the serial interface.

Let's describe a scenario where this might be used. Assume a touchscreen controller is connected to a HomeVision-Pro serial port. The touchscreen has an infrared control screen to control a VCR. When the user touches the Fast Forward button, HomeVision-Pro must start transmitting a particular IR signal, and do so continually. When the user releases the Fast Forward button, or touches another button, HomeVision-Pro must stop transmitting. Previously, this would have been difficult to accomplish. Now, the touchscreen can send one serial command to HomeVision-Pro to start the transmissions, and another to stop them, greatly simplifying things.

New Schedule Commands

The new commands that can be used are described below:

```
Start Transmitting IR Signal ## Repeatedly
```

> This command starts the repeated transmission of the IR signal specified by ##.

```
Start Transmitting IR Signal At Variable ## Repeatedly
```

> This command uses the value of the variable as the ID number of the IR signal, and starts repeatedly transmitting it. This command cannot be used in a schedule – it must be sent to the controller over the serial interface.

```
Start Transmitting IR Test Signal
```

This command starts the repeated transmission of the IR signal loaded into the special test area. This area can hold one signal. It is normally used during the IR signal learning process, which allows the user to load the learned signal into the special test area, and then transmit it to verify it works. However, this test area can be used during normal operation as a means for PC software to load and transmit a signal not previously loaded in the controller. The PC software must first load the signal into the test area, and then set up three additional variables. The software can then use this command to start transmitting it repeatedly. Refer to the file "Using Extra IR Signals" for details. This command cannot be used in a schedule – it must be sent to the controller over the serial interface.

```
Stop Transmitting IR Signals
```

This stops the repeated transmission of IR signals. Note that the controller has a built-in safeguard to stop transmitting if 255 consecutive signals are sent without another start or stop command being received.

If you wish to send these commands to the controller over the serial interface, refer to the infrared command section of the newest "Serial Protocol" file for the format details.

Details

Here's how this repeated transmission works when controlled via the serial interface. First, you would send a command to the controller to specify the delay between signal transmissions (this will be discussed in detail below). Next, you would send one of the commands to start transmitting an IR signal. The signal would not be transmitted immediately, but only when the controller returns to running its main "loop". This would typically be within a few milliseconds of completing the serial command.

Once each loop, the controller will transmit the IR signal. It will also delay the amount of time specified by a "delay" variable, thereby putting a gap between IR signals. A gap is usually needed so that the IR receiver is able to decode and recognize the signal. The signal will continue to be transmitted each loop. Finally, you would send a command to stop the IR transmissions. If the controller transmits 255 times before receiving another Start or Stop command, it will abort the transmissions. This is to prevent the controller from getting stuck in an endless loop. Most IR signals are at least 50ms long, and with a typical gap of 20ms between transmissions, it would take about 18 seconds to transmit it 255 times. If you need it to transmit longer than this, you should send another start command every 17 seconds or sooner.

Delay Between Signals

During normal operation (when not transmitting), a loop typically takes between 5 and 30 milliseconds to complete (depending on the user's schedule and peripheral devices). While it is transmitting, the loop will slow down considerably. IR signals typically range in duration from 50 to 200ms. This time, plus the "delay" time (if any) will be added to the 5-30ms typical loop time.

The "delay" variable allows the user to control the minimum gap between transmitted IR signals. This variable should be set prior to starting the IR transmission. The variable is set using a special serial command that can write to the controller's internal RAM. The command format is as follows:

,6EF53DD

where: ,6 is the command to write a value to RAM.
EF53 is the RAM address to write to.
DD is the delay value, in milliseconds, in hex format.

For example, to put a delay of 50ms ("32" in hex format), use this command:

,6EF5332

Note that this delay will be in addition to the 5-30ms typical loop time. Therefore, you might want to reduce the delay value to account for this. For example, assume that the loop time for your schedule averages 20ms, and you want a 50ms total delay between signals. You should then set the delay variable to 30ms.