HomeVision "How To" Information

Custom Solutions Inc.

Topics covered:

- Using two-way X-10 modules with HomeVision
- Controlling the RCS TX10-B bi-directional thermostat
- Using a rain gauge with HomeVision
- Pop-up surveillance camera on your TV
- How to tell when it's a holiday
- Using the Xantech 791-40 amplified connecting block
- Creating your own custom video screens
- Restoring X-10 lights to their prior level
- Using CompCo Engineering, Inc. probes with HomeVision
- Using an analog temperature sensor with HomeVision
- Displaying messages on your TV
- Sending FAXes with HomeVision
- Using Elk sensitive relays with HomeVision
- Tips on transmitting serial data
- Uses for temperature information
- Receiving infrared sequences

NOTE: Although these tips were written for the HomeVision controller, most apply to the HomeVision-Pro controller also.

Using Two-Way X-10 Modules with HomeVision

Module Description

X-10 recently introduced several modules (such as the LM14A lamp module) that can transmit their status via X-10. When these modules receive an X-10 "Status Request" signal, they will respond with either a "Status is On" or "Status is Off" signal. This allows HomeVision to determine whether the module is actually on or off. Note that there is no way to determine what level the module is at if it's on.

Overview

Using the two-way capability of these modules requires two steps in your HomeVision schedule:

1) Requesting the module status.
2) Reading the module response and taking the appropriate action.

To request the status, use these two X-10 commands:

```
X-10: A 1 (Lamp module) Transmit House/Unit Code only
X-10: Transmit House Code 'A' STATUS REQUEST
```

When module A-1 receives this signal, it will transmit a single X-10 signal, specifically, either:

```
House Code A - Status is On
```

or

```
House Code A - Status is Off
```

There are two ways you can use HomeVision to read this response and take action on it:

1) Check for the X-10 signal with two If-Then statements where the condition is an "X-10 sequence", like this:

```
If
   X-10 sequence [A-Status is On] is received within 5 seconds
Then
   ; Do whatever you want when the module is ON
End If
If
   X-10 sequence [A-Status is Off] is received within 5 seconds
Then
   ; Do whatever you want when the module is OFF
End If
```

An X-10 sequence condition is normally used to receive and respond to multiple successive X-10 signals. However, in this case we're only looking for one signal. Note that the module's response does not contain a unit code (only the house code and status). Therefore, if we have more than one two-way module with the same house code, we need to know which one we requested the status of. The examples given later show how to handle this.

2) Check the module's status bits which indicate that a "Status is On" or "Status is Off" signal was received, like this:

```
If
   X-10: A 1 (Lamp module) "Status is On" bit is set
Then
   ; Do whatever you want when the module is ON
End If
If
   X-10: A 1 (Lamp module) "Status is Off" bit is set
Then
   ; Do whatever you want when the module is OFF
End If
```

Whenever the controller transmits a "Status Request" signal for a specific module, then receives a "Status is On" or "Status is Off" signal from the same house code, it sets the corresponding status bit (note that this is only true for HomeVision PROM version 2.5 or later). This allows HomeVision to determine the module's house and unit codes, even though the module only transmits its house code. The code shown above checks the status bits to determine which, if any, status response was received.

Examples

There are two general ways to use these concepts in a schedule:

1) Request the status when you need it, wait a few seconds for the response, then do whatever you want based on the response. With this method, we'll check for the response with an "X-10 sequence" condition.

2) Request the status when you need it, then read the response in a periodic event and use it to update HomeVision's X-10 state table. With this method, we'll check the X-10 status bits to determine the response.

The first method works well in most cases, but there are times you might prefer the second. Examples of each are given below.

Method 1

This approach is to use a single event to request the module status and decode it's response. Here's an example macro that reads the status of lamp module A-1, then sets HomeVision's internal state table to match the actual module state:

```
X-10: A 1 (Lamp module) Transmit House/Unit Code only
X-10: Transmit House Code 'A' STATUS REQUEST
Wait 0:00:05.00 with timer #0 (Status request timer), Then:
   If
      X-10 sequence [A-Status is On] received within 5 seconds
   Then
      X-10: A 1 (Lamp module) Set state to ON
   Else
      If
         X-10 sequence [A-Status is Off] received within 5 seconds
      Then
         X-10: A 1 (Lamp module) Set state to OFF
      Else
```

```
         ; No response received from module - Do whatever you want.
      End If
   End If
End Wait
```

This macro first transmits the "Status Request" signal for device A-1. A wait timer is then set to 5 seconds to allow time for the transmission and the response. When the timer goes off, we use two If-Then statements to determine if we received back a "Status is On" or "Status is Off" signal. The condition is an "X-10 sequence", which in this case is only one X-10 signal in length. If we receive one of these signals, we set HomeVision's internal state table to ON or OFF to indicate the correct module state.

You might want to do something in addition to updating HomeVision's state table. For example, if the module is on, you could turn it off, but if it's off, you might do nothing. However, if you want HomeVision's internal state table (and the X-10 TV screen) to be correct, you should also set the state to ON or OFF as done here.

Notice the last "Else" statement in this example. It will occur if no response is received by the time the timer expires. This could happen if there are other X-10 transmissions going on in the house when HomeVision tries to transmit. HomeVision may have to wait or retry multiple times, meaning the timer could expire before the module has responded. It could also happen if any of the X-10 signals didn't make it to their destination. In this case, you might want to request status again and give some sort of error indication.

Improved Macro

One disadvantage of the above example is that it takes 5 seconds before HomeVision checks for the response and takes any action. You could shorten the wait timer to 2 seconds to speed it up, as the transmission and response will normally complete in less than 2 seconds. However, this increases the chance that the timer will go off before the response is received, resulting in missing the response.

A clever way to improve on this is to have the macro check for the response after 2 seconds, and if it wasn't received, wait several more seconds and check again. This is shown in the following macro. This has the advantage of providing a quick response most times, and only taking longer if the X-10 transmission was delayed a while. It is, however, more complex to write.

```
X-10: A 1 (Lamp module) Transmit House/Unit Code only
X-10: Transmit House Code 'A' STATUS REQUEST
Wait 0:00:02.00 with timer #0 (Status request timer), Then:
   If
      X-10 sequence [A-Status is On] received within 5 seconds
   Then
      X-10: A 1 (Lamp module) Set state to ON
   Else
      If
         X-10 sequence [A-Status is Off] received within 5 seconds
      Then
         X-10: A 1 (Lamp module) Set state to OFF
      Else
         ; No response received from module, so wait 3 more seconds:
         Wait 0:00:03.00 with timer #0 (Status request timer), Then:
            If
               X-10 sequence [A-Status is On] received within 5 seconds
            Then
               X-10: A 1 (Lamp module) Set state to ON
            Else
```

```
               If
                  X-10 sequence [A-Status is Off] received within 5 seconds
               Then
                  X-10: A 1 (Lamp module) Set state to OFF
               Else
                  ; No response received from module - Do whatever you want.
               End If
            End If
         End Wait
      End If
   End If
End Wait
```

Method 2

The main problem with the previous method is the delay in receiving the module's response. One way around this is to transmit the status request whenever you need to (as shown previously), but check for the response in a separate periodic event. Set the periodic event to run "each loop" and it will catch the X-10 response almost immediately after it's received. This provides the fastest possible response time.

One obstacle with this is that the module does not transmit its unit code in its response. How can the periodic event know which unit code the response belongs to? If you have only one two-way module on each house code that you request status for, then it's easy: the response must be from that module. But what if you have more than one? The best solution is to use the controller's X-10 status bits. Whenever the controller transmits a "Status Request" signal for a specific module, then receives a "Status is On" or "Status is Off" signal, it assumes the signal is from the requested module and sets the corresponding status bit. Here's an example of how you would use these in a periodic event:

```
If
   X-10: A 1 (Lamp module) "Status is On" bit is set
Then
   X-10: A 1 (Lamp module) Set state to ON
   X-10: A 1 (Lamp module) Clear "Status is On" bit
   ; Do whatever else you want
End If
If
   X-10: A 1 (Lamp module) "Status is Off" bit is set
Then
   X-10: A 1 (Lamp module) Set state to OFF
   X-10: A 1 (Lamp module) Clear "Status is Off" bit
   ; Do whatever else you want
End If
```

Note that we clear the "Status is On" and "Status is Off" bits so the If-Then statement isn't repeatedly true. We only want the condition to be true after the module transmits the corresponding X-10 signal.

If you have multiple two-way modules, you can check for all the responses in one periodic event, like this:

```
If
   X-10: A 1 (Lamp module) "Status is On" bit is set
Then
   X-10: A 1 (Lamp module) Set state to ON
   X-10: A 1 (Lamp module) Clear "Status is On" bit
```

```
End If
If
   X-10: A 1 (Lamp module) "Status is Off" bit is set
Then
   X-10: A 1 (Lamp module) Set state to OFF
   X-10: A 1 (Lamp module) Clear "Status is Off" bit
End If
If
   X-10: A 2 (Family room lamp) "Status is On" bit is set
Then
   X-10: A 2 (Family room lamp) Set state to ON
   X-10: A 2 (Family room lamp) Clear "Status is On" bit
End If
If
   X-10: A 2 (Family room lamp) "Status is Off" bit is set
Then
   X-10: A 2 (Family room lamp) Set state to OFF
   X-10: A 2 (Family room lamp) Clear "Status is Off" bit
End If
If
   X-10: F 11 (Bedroom lamp) "Status is On" bit is set
Then
   X-10: F 11 (Bedroom lamp) Set state to ON
   X-10: F 11 (Bedroom lamp) Clear "Status is On" bit
End If
If
   X-10: F 11 (Bedroom lamp) "Status is Off" bit is set
Then
   X-10: F 11 (Bedroom lamp) Set state to OFF
   X-10: F 11 (Bedroom lamp) Clear "Status is Off" bit
End If
```

Conclusion

There are several different ways to use these two-way modules with HomeVision. Choose whichever method best fits your needs. You can even use these to detect when a module is not responding and take appropriate action.

CONTROLLING THE RCS TX10-B BI-DIRECTIONAL THERMOSTAT

NOTE

This article was written for the version 2.3 software. Version 2.3 did not contain any commands to directly control thermostats. Instead, it required you to use your own variables and macros, a fairly involved task. Version 2.5 added automatic control of RCS TX10 and TX10-B thermostats, as well as commands to directly control them that you can include in your schedule. Therefore, most users will never need to do what this article describes. However, if you want to modify the way the automatic control works, or control different thermostats, this article provides a good idea of the process you should follow.

INTRODUCTION

The RCS TX10-B bi-directional thermostat can be controlled and read via X-10. This article describes how to use this thermostat with HomeVision to control your Heating, Ventilation, and Air Conditioning (HVAC) system. After setting it up, you will be able to:

- Use X-10, infrared, etc. to change the thermostat temperature setting, mode, or fan.
- Read the temperature setting, actual temperature, mode, or fan state.
- Control it from a TV screen.
- Control it from a PC.
- Automate it to a time-temperature schedule.
- Change the time-temperature schedule from a TV screen.

There are so many things that HomeVision can do that it will take a while for you to implement them all. We suggest you allow 3-4 hours to implement and test everything. It's especially important to thoroughly test each macro. One typo and things may not work properly. These errors will be much easier to find if you test each macro immediately after writing it. Otherwise, you may wake up in the middle of the night with the house at the wrong temperature and have no idea why.

SCHEDULE OVERVIEW

This is a summary of the process you'll follow to create your schedule:

- Create variables and flags to store the current state of the HVAC system.

- Create a variety of macros to control the system. There will be four groups of macros:

  1. Macros which send commands to the thermostat.
  2. Macros which request status from the thermostat. These macros do not read the response from the thermostat (reading is done with a periodic event that runs continually).
  3. Macros which control how the thermostat TV control screen responds when the user presses a button on their remote control.
  4. Macros which can be included anywhere in the schedule to change the temperature setting.

- Create periodic events which periodically request status from the thermostat and continually check for X-10 reports from the thermostat.

- Configure several items in the PC software.

- Enter the initial thermostat time/temperature schedule in the PC software.

IMPORTANT NOTES

There are several important things to note about this example schedule:

- All the objects (variables, macros, etc.) are numbered 1 and up. You will probably be using different numbers and must adjust the commands accordingly.

- The thermostat is assumed to be on house code J.

- This example covers a single-zone HVAC system. For additional zones, you must duplicate most of the flags, variables, periodic events, and macros for each zone. You can control as many zones as you want in this way. However, the TV control screens and the automatic time/temperature schedule only supports two zones.

- This article assumes you are familiar with how the TV thermostat control screens work in conjunction with your macros. If you are not clear on this, please read the file "Screens" provided on installation disks. The file also explains how we choose the variable values (0-Off, 3=Cool, etc.) so that they will be displayed as the proper words on the TV screen.

- The documentation provided with the TX10-B thermostat shows X-10 preset dim levels of 1 to 32. HomeVision numbers the levels as 0 to 31. Thus, preset dim level 1 in the TX10-B documentation corresponds to level 0 for HomeVision, level 2 for the TX10-B corresponds to level 1 for HomeVision, etc.

- This example uses the X-10 preset dim capabilities of the TX10-B thermostat to communicate with it. It does not use the "On"/"Off" commands.

- You can modify this example schedule to suit your precise needs. This will be easy once you understand how it works. There are some features in it you might not use. Review it to learn how it works, then decide what portions to incorporate.

CREATING THE SCHEDULE

Step 1 - Create Variables

Create these variables. The numbers shown for initial value are recommended values. They will probably not agree with your thermostat's actual settings when you first download the schedule. However, they will be corrected when HomeVision reads the status from the thermostat. You could also go to the TV thermostat screen and use your remote to make the correct settings.

#	NAME	INITIAL VALUE	DESCRIPTION
1	HVAC Control Mode	0	Indicates whether HomeVision should control the thermostat to the time/temperature schedule: 0 = Off (HomeVision doesn't change the temperature) 2 = Auto (HomeVision runs the time/temperature schedule) This variable should be set to 2 when you want HomeVision to control the thermostat temperature in accordance with the

			time/temperature schedule. Set it to 0 when you want to hold the current temperature (just like the "Hold" or "Vacation" mode on standard thermostats).
2	HVAC System Mode	0	Indicates the current state of the HVAC system: 0 = Off 2 = Auto 3 = Cool 4 = Heat These values allow the state to be displayed properly on the thermostat TV screen.
3	HVAC Fan Mode	2	Indicates the current state of the HVAC fan: 1 = On (on continually) 2 = Auto (fan runs under control of thermostat) These values allow the state to be displayed properly on the thermostat TV screen.
4	HVAC Temp Setting	75	Holds the thermostat temperature setting.
5	Current Temp	75	Holds the current temperature (read from the thermostat).
6	Thermostat Data Request	1	Used by the periodic event "Request Thermostat Data" to determine what type of status to next request from the thermostat. The periodic event requests different data each time it runs based on these possible variable values: 1 = Read current temperature 2 = Read temperature setting 3 = Read HVAC mode 4 = Read fan mode
7	Preset Dim Level	0	Used to temporarily hold the received preset dim level. We can then check it in If-Then statements or save it to another variable.

Step 2 - Create Flags

Create these flags:

#	NAME	DESCRIPTION

1	Reading Temperature Setting	Used to indicate whether we're reading the temperature setting or the actual temperature from the TX10-B thermostat:
		SET = Reading HVAC temperature setting
		CLEAR = Reading current temperature
		This is necessary because both types of reports from the thermostat use the same X-10 house and unit codes. We must know which temperature we requested in order to know what the report means.

2	HVAC Setback Mode	Indicates whether the thermostat's "setback" mode is on or off:
		SET = On
		CLEAR = Off

		The TX10-B's built-in setback mode is not used in this example. It is provided only for completeness. You may skip it.

Step 3 - Create Macros

Create these macros:

#	NAME	DESCRIPTION
1	Request Current Temp	Requests the thermostat to report the current temperature.
2	Request Temp Setting	Requests the thermostat to report the temperature setting.
3	Request HVAC Mode	Requests the thermostat to report the HVAC mode.
4	Request HVAC Fan Mode	Requests the thermostat to report the fan mode.
5	Request HVAC Setback Mode	Requests the thermostat to report the setback mode.
6	Request HVAC Setback Delta	Requests the thermostat to report the setback temperature delta. This command is not used in this example. It is provided only for completeness. You may skip it.
7	Set HVAC Temperature	Commands the thermostat to the desired temperature. Variable #4 (HVAC Temp Setting) must contain the desired temperature before this macro is run.
8	Set HVAC Mode	Commands the thermostat to the desired mode. Variable #2 (HVAC System Mode) must contain the desired mode before this macro is run.
9	Set HVAC Fan Mode	Commands the thermostat fan to the desired mode. Variable #3 (HVAC Fan Mode) must contain the desired mode before this macro is run.
10	HVAC Screen Control Mode	Runs when the user presses button #1 while viewing the thermostat control TV screen. It cycles the HVAC Control Mode variable (#1) between Off and Auto.
11	HVAC Screen System Mode	Runs when the user presses button #2 while viewing the thermostat control TV screen. It cycles the HVAC System Mode variable (#2) between Off, Auto, Cool, and Heat. It then calls macro #8 to set the thermostat to the new mode.
12	HVAC Screen Fan Mode	Runs when the user presses button #3 while viewing the thermostat control TV screen. It cycles the HVAC Fan Mode variable (#3) between Auto and On. It then calls macro #9 to set the fan to the new mode.
13	HVAC Screen Temp Setting	Runs when the user presses button #4 while viewing the thermostat control TV screen. HomeVision automatically sets the HVAC Temp Setting variable (#4) to the temperature entered by the user. This macro then calls macro #7 to set the thermostat to the desired temperature.
14	Increase HVAC Temp 1 Deg	Commands the thermostat to increase the temperature setting 1 degree (unless it's already at or above 90 degrees).
15	Decrease HVAC Temp 1 Deg	Commands the thermostat to decrease the temperature setting 1 degree (unless it's already at or below 60 degrees).
16	Setback HVAC Temperature	Changes the thermostat setting to an energy saving temperature while the house is unoccupied. It "holds" the same temperature and prevents HomeVision from changing the temperature at the scheduled times. This macro should be run when the homeowner leaves home.
17	Resume HVAC Control	Sets the thermostat to the temperature specified in the time/temperature schedule. It also resumes controlling the thermostat to the time/temperature schedule. This macro should be run when the homeowner returns home.
18	Initialize HVAC System	Initializes the thermostat operating modes (as discussed under "Step 7").

		This normally has to be run just once. However, you may run it at other times if you think the thermostat has inadvertantly changed modes.

Step 4 - Create Periodic Events

Create these periodic events:

#	NAME	RATE	DESCRIPTION
1	Read Thermostat Data	Each loop	Repeatedly checks to see if any X-10 preset dim commands were received from the thermostat. If so, it determines the signal's meaning and updates the corresponding flags and variables. This is where all messages from the TX10-B are received.
2	Request Thermostat Data	1 minute (or slower if desired)	Runs every minute and requests status from the thermostat. This event doesn't read the thermostat's response (periodic event #1 does that). This event cycles through requests for different status each time. This is necessary because we can't read all the thermostat's status at once. Variable #6 tells us what data to request each time.

Step 5 - Enter Actions for Macros and Periodic Events

The commands to enter in each macro and periodic event are provided below:

MACRO #1: Request Current Temp

Clear flag #1 (Reading Temperature Setting)
X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
X-10: Transmit 'Preset Dim' to level 0

MACRO #2: Request Temp Setting

Set flag #1 (Reading Temperature Setting)
X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
X-10: Transmit 'Preset Dim' to level 1

MACRO #3: Request HVAC Mode

X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
X-10: Transmit 'Preset Dim' to level 2

MACRO #4: Request HVAC Fan Mode

X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
X-10: Transmit 'Preset Dim' to level 3

MACRO #5: Request HVAC Setback Mode

```
    X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 4
```

MACRO #6: Request HVAC Setback Delta

```
    X-10: J 5 (Thermostat Request) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 5
```

MACRO #7: Set HVAC Temperature

```
    Var #7 (Preset Dim Level) = var #4 (HVAC Temp Setting)
    If
        Var #4 (HVAC Temp Setting) >= 40
        And Var #4 (HVAC Temp Setting) <= 67
    Then
        X-10: J 2 (Thermostat Set Point) Transmit House/Unit Code only
        ;Subtract 36 to convert from temperature to a preset dim level (# 0-31):
        Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) - 36
    End If
    ;
    If
        Var #4 (HVAC Temp Setting) >= 68
        And Var #4 (HVAC Temp Setting) <= 99
    Then
        X-10: J 3 (Thermostat Set Point) Transmit House/Unit Code only
        ;Subtract 68 to convert from temperature to a preset dim level (# 0-31):
        Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) - 68
    End If
    ;
    If
        Var #4 (HVAC Temp Setting)  <= 39
        Or Var #4 (HVAC Temp Setting) >= 100
    Then
        ;These temperatures are invalid.  Force to 75 to be safe:
        Var #4 (HVAC Temp Setting)  = 75
        Var #7 (Preset Dim Level) = 7
        X-10: J 3 (Thermostat Set Point) Transmit House/Unit Code only
    End If
    ;
    ;Now that we've found the necessary preset dim level, we can transmit it:
    X-10: Transmit 'Preset Dim' to level in variable #7 (Preset Dim Level)
```

MACRO #8: Set HVAC Mode

```
If
    Var #2 (HVAC System Mode)  = 0
Then
    ;Command the thermostat to OFF:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 0
End If
;
If
    Var #2 (HVAC System Mode)  = 2
Then
    ;Command the thermostat to AUTO:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 3
End If
;
If
    Var #2 (HVAC System Mode)  = 3
Then
    ;Command the thermostat to COOL:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 2
End If
;
If
    Var #2 (HVAC System Mode)  = 4
Then
    ;Command the thermostat to HEAT:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 1
End If
```

MACRO #9: Set HVAC Fan Mode

```
If
    Var #3 (HVAC Fan Mode)  = 2
Then
    ;Command the fan to AUTO:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 5
End If
;
If
    Var #3 (HVAC Fan Mode)  = 1
Then
    ;Command the fan to ON:
    X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
    X-10: Transmit 'Preset Dim' to level 4
End If
```

MACRO #10: HVAC Screen Control Mode

```
;  This macro runs when the user presses button #1 while viewing the thermostat control TV screen.
;  It cycles the HVAC Control Mode variable (#1) between Off (0) and Auto (2).
;
If
    Var #1 (HVAC Control Mode) = 0
```

```
Then
    ;Change the mode from Off to Auto:
    Var #1 (HVAC Control Mode) = 2
Else
    ;Change the mode from anything else to Off:
    Var #1 (HVAC Control Mode) = 0
End If
```

MACRO #11: HVAC Screen System Mode

```
;  This macro runs when the user presses button #2 while viewing the thermostat control TV screen.
;  It cycles the HVAC System Mode variable (#2) between Off (0), Auto (2), Cool (3), and Heat (4).
;  You may want to modify this to skip the Auto mode if you're not using it.
;
If
    Var #2 (HVAC System Mode) = 0
Then
    ;Change the mode from Off to Auto:
    Var #2 (HVAC System Mode) = 2
    Do macro #8 (Set HVAC Mode)
Else
    If
        Var #2 (HVAC System Mode) = 2
    Then
        ;Change the mode from Auto to Cool:
        Var #2 (HVAC System Mode) = 3
        Do macro #8 (Set HVAC Mode)
    Else
        If
            Var #2 (HVAC System Mode) = 3
        Then
            ;Change the mode from Cool to Heat:
            Var #2 (HVAC System Mode) = 4
            Do macro #8 (Set HVAC Mode)
        Else
            If
                Var #2 (HVAC System Mode) = 4
            Then
                ;Change the mode from Heat to Off:
                Var #2 (HVAC System Mode) = 0
                Do macro #8 (Set HVAC Mode)
            Else
                ;If we get here, the requested setting is invalid.  Don't do anything.
            End If
        End If
    End If
End If
```

MACRO #12: HVAC Screen Fan Mode

```
;  This macro runs when the user presses button #3 while viewing the thermostat control TV screen.
;  It cycles the HVAC Fan Mode variable (#3) between Auto (2) and On (1).
```

```
;
If
    Var #3 (HVAC Fan Mode) = 2
Then
    ;Change the mode from Auto to On:
    Var #3 (HVAC Fan Mode) = 1
Else
    ;Change the mode from anything else to Auto:
    Var #3 (HVAC Fan Mode) = 2
End If
;
;Now run the macro to send the command to the thermostat:
Do macro #9 (Set HVAC Fan Mode)
```

MACRO #13: HVAC Screen Temp Setting

```
;  This macro runs when the user presses button #4 while viewing the thermostat control TV screen.
;  HomeVision sets the HVAC Temp Setting variable (#4) to the temperature entered by the user.
;  This macro verifies the value is OK, then sets the thermostat to it.
;
;First, ensure the desired temperature is between 60 and 90 degrees:
If
    Var #4 (HVAC Temp Setting) < 60
Then
    ;Don't let user set temperature below 60:
    Var #4 (HVAC Temp Setting) = 60
End If
;
If
    Var #4 (HVAC Temp Setting) > 90
Then
    ;Don't let user set temperature above 90:
    Var #4 (HVAC Temp Setting) = 90
End If
;
;Now run the macro to send the command to the thermostat:
Do macro #7 (Set HVAC Temperature)
```

MACRO #14: Increase HVAC Temp 1 Degree

```
;Raise setting unless it's already at or above 90 degrees:
If
    Var #4 (HVAC Temp Setting) < 90
Then
    Increment var #4 (HVAC Temp Setting) [roll over at 255]
    ;Now run the macro that actually changes the thermostat setting:
    Do macro #7 (Set HVAC Temperature)
End If
;
;Note that we're not using the TX10-B commands "Increase Setpoint" or "Decrease Setpoint".
;You may prefer to use those commands instead.
```

MACRO #15: Decrease HVAC Temp 1 Degree

```
;Lower setting unless it's already at or below 60 degrees:
If
    Var #4 (HVAC Temp Setting) > 60
Then
    Decrement var #4 (HVAC Temp Setting) [roll under at 0]
    ;Now run the macro that actually changes the thermostat setting:
    Do macro #7 (Set HVAC Temperature)
End If
```

MACRO #16: Setback HVAC Temperature

```
;  This macro sets the thermostat back.
;  If the HEAT mode is on, it lowers the temperature to a predefined level.
;  If the COOL mode is on, it raises the temperature to a predefined level.
;  If in AUTO mode or OFF, nothing happens.
;  This macro should be run when the homeowner leaves home.
;
If
    Var #2 (HVAC System Mode) = 3
Then
    ;It's in the COOL mode (variable = 3), so use a high setting (84 degrees):
    Var #4 (HVAC Temp Setting) = 84
    Do macro #7 (Set HVAC Temperature)
Else
    If
        Var #2 (HVAC System Mode) = 4
    Then
        ;It's in the HEAT mode (variable = 4), so use a low setting (65 degrees):
        Var #4 (HVAC Temp Setting) = 65
        Do macro #7 (Set HVAC Temperature)
    End If
End If
;
;Now set the control mode variable to OFF (0) so the controller "holds" the current temperature.
;If we didn't do this, it would run the thermostat to the time/temperature schedule.
Var #1 (HVAC Control Mode) = 0
```

MACRO #17: Resume HVAC Control

```
;  This macro sets the thermostat to the temperature specified in the time/temperature schedule.
;  It also resumes controlling the thermostat to the time/temperature schedule.
;  This macro should be run when the homeowner returns home.
;
Set zone 1 thermostat to scheduled temperature
Var #1 (HVAC Control Mode) = 2
```

MACRO #18: Initialize HVAC System

```
;This macro sets the thermostat up to work with HomeVision
;
;First, enable the thermostat's "Preset Dim Command Mode":
X-10: J 4 (Thermostat Command) Transmit House/Unit Code only
X-10: Transmit 'Preset Dim' to level 18
;
;Next, disable the "Ack Message Mode":
X-10: Transmit 'Preset Dim' to level 21
;
;Next, disable the "Echo Command Mode":
X-10: Transmit 'Preset Dim' to level 23
;
;Next, disable the "Safe Command Mode":
X-10: Transmit 'Preset Dim' to level 25
;
;Finally, disable the "Unit Code On/Off Mode":
X-10: Transmit 'Preset Dim' to level 17
```

PERIODIC EVENT #1: Read Thermostat Data

```
;  This macro repeatedly checks to see if any X-10 preset dim commands were received from
;  the thermostat.  If so, it determines the signal's meaning and updates the corresponding
;  flags and variables.  This is where all messages from the TX10-B are received.

;This first If-Then statement will check for a status report from the thermostat
If
    X-10: J 6 (Thermostat Response) 'Preset Dim' bit is set
    ;If set, we know a J 6 preset dim signal was received
Then
    ;Store the received preset dim level in a variable:
    X-10: J 6 (Thermostat Response)-Put received preset dim level into variable #7 (Preset Dim Level)
    ;
    ;Now determine the meaning of the received preset dim level:
    If
        Var #7 (Preset Dim Level) = 0
        ;This report means the thermostat is now off
    Then
        ;Set the system mode variable to 0 to indicate we're in the off mode:
        Var #2 (HVAC System Mode) = 0
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 1
        ;This report means the thermostat is now in the heat mode
    Then
        ;Set the system mode variable to 4 to indicate we're in the heat mode:
        Var #2 (HVAC System Mode) = 4
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 2
        ;This report means the thermostat is now in the cool mode
    Then
```

```
        ;Set the system mode variable to 3 to indicate we're in the cool mode:
        Var #2 (HVAC System Mode) = 3
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 3
        ;This report means the thermostat is now in the auto mode
    Then
        ;Set the system mode variable to 2 to indicate we're in the auto mode:
        Var #2 (HVAC System Mode) = 2
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 4
        ;This report means the thermostat fan is now on
    Then
        ;Set the fan mode variable to 1 to indicate it's on:
        Var #3 (HVAC Fan Mode) = 1
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 5
        ;This report means the thermostat fan is now in auto mode
    Then
        ;Set the fan mode variable to 2 to indicate it's in auto mode:
        Var #3 (HVAC Fan Mode) = 2
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 6
        ;This report means the thermostat setback mode is now on
    Then
        ;Set the setback mode flag to indicate it's on:
        Set flag #2 (HVAC Setback Mode)
    End If
    ;
    If
        Var #7 (Preset Dim Level) = 7
        ;This report means the thermostat setback mode is now off
    Then
        ;Clear the setback mode flag to indicate it's off:
        Clear flag #2 (HVAC Setback Mode)
    End If
    ;
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 6 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
;
;These next If-Then statements will check for a temperature report from the thermostat
If
    X-10: J 11 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is between -60 and -29 degrees
Then
```

```
    ;Now check flag #1 to see if this report is the actual temp or the temp setting.
    ;Then set the corresponding variable to 0 degrees (don't allow negative temperatures).
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = 0
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = 0
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 11 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
If
    X-10: J 12 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is between -28 and +3 degrees
Then
    X-10: J 12 (Thermostat Response)-Put received preset dim level into variable #7 (Preset Dim Level)
    If
        Var #7 (Preset Dim Level) <= 28
    Then
        ;Value is <=28, so temperature is negative.  Limit it to 0 (no negative numbers)
        Var #7 (Preset Dim Level) = 0
    Else
        ;Subtract 28 degrees to convert the preset dim level to the temperature:
        Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) - 28
    End If
    ;
    ;Now check flag #1 to see if this report is the actual temp or the temp setting
    ;Then set the corresponding variable to 0 degrees (don't allow negative temperatures)
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = Var #7 (Preset Dim Level)
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = Var #7 (Preset Dim Level)
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 12 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
If
    X-10: J 13 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is between 4 and 35 degrees
Then
    X-10: J 13 (Thermostat Response)-Put received preset dim level into variable #7 (Preset Dim Level)
    ;Add 4 degrees to convert the preset dim level to the temperature:
    Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) + 4
    ;
    ;Now check flag #1 to see if this report is the actual temp or the temp setting
    ;Then set the corresponding variable to the reported temperature
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = Var #7 (Preset Dim Level)
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = Var #7 (Preset Dim Level)
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 13 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
If
    X-10: J 14 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is between 36 and 67 degrees
Then
    X-10: J 14 (Thermostat Response)-Put received preset dim level into variable #7 (Preset Dim Level)
    ;Add 36 degrees to convert the preset dim level to the temperature:
    Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) + 36
    ;
    ;Now check flag #1 to see if this report is the actual temp or the temp setting
    ;Then set the corresponding variable to the reported temperature
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = Var #7 (Preset Dim Level)
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = Var #7 (Preset Dim Level)
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 14 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
If
    X-10: J 15 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is between 68 and 99 degrees
Then
    X-10: J 15 (Thermostat Response)-Put received preset dim level into variable #7 (Preset Dim Level)
    ;Add 68 degrees to convert the preset dim level to the temperature:
    Var #7 (Preset Dim Level) = var #7 (Preset Dim Level) + 68
    ;
    ;Now check flag #1 to see if this report is the actual temp or the temp setting
    ;Then set the corresponding variable to the reported temperature
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = Var #7 (Preset Dim Level)
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = Var #7 (Preset Dim Level)
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 15 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
;
If
    X-10: J 16 (Thermostat Response) 'Preset Dim' bit is set
    ;This indicates the temperature is >=100 degrees
Then
    ;Now check flag #1 to see if this report is the actual temp or the temp setting
    ;Then set the corresponding variable to 99 degrees (don't allow numbers >99)
    If
        Flag #1 (Reading Temperature Setting) is set
    Then
        ;We requested the temperature setting
        Var #4 (HVAC Temp Setting) = 99
    Else
        ;We requested the current temperature
        Var #5 (Current Temp) = 99
    End If
    ;Now clear the "Preset Dim" bit and the received level so we can receive again:
    X-10: J 16 (Thermostat Response) Clear 'Preset Dim' bit and level
End If
```

PERIODIC EVENT #2: Request Thermostat Status

```
;Use the value of variable 6 to determine what type of data to request this time:
If
    Var #6 (Thermostat Data Request) = 1
Then
    Do macro #1 (Request Current Temp)
End
;
If
    Var #6 (Thermostat Data Request) = 2
Then
    Do macro #2 (Request Temp Setting)
End
;
If
    Var #6 (Thermostat Data Request) = 3
Then
    Do macro #3 (Request HVAC Mode)
End
;
If
    Var #6 (Thermostat Data Request) = 4
Then
    Do macro #4 (Request HVAC Fan Mode)
End
;
;Now change the variable so this event will request different data next time.
```

```
;These commands cycle the variable between 1, 2, 3, and 4.
If
    Var #6 (Thermostat Data Request) = 4
Then
    Var #6 (Thermostat Data Request) = 1
Else
    Increment var #6 (Thermostat Data Request) [roll over at 255]
End If
```

NOTE: If you are using two different zones, each with its own thermostat, you should use only one periodic event to read the status from them. Change the actions here to cycle the Thermostat Data Request variable from 1 through 8 instead of 1 through 4. The first 4 values should work just as shown here, requesting the 4 types of data for zone 1. Values 5 through 8 should request the same data for the zone 2. In this way, only one of the thermostats is used each time the periodic event occurs. If you requested status from both thermostats at the same time, they would transmit their responses at approximately the same time, causing X-10 collisions.

Step 6 - Put Other Commands In Schedule

You will probably want to put commands elsewhere in your schedule to control the thermostat. Here are some examples.

- When the house is unoccupied, you'll want to set the temperature back and stop controlling the thermostat to the time/temperature schedule. This can be done with a single command anywhere in your schedule:

  Do macro #16 (Setback HVAC Temperature)

- When the homeowner returns home, you'll want to return the thermostat to the currently-scheduled temperature and resume controlling the thermostat to the time/temperature schedule. This can be done with a single command anywhere in your schedule:

  Do macro #17 (Resume HVAC Control)

- You may want users to be able to increase or decrease the temperature via an X-10 signal (perhaps from a 'mini-controller' by the bed). You might use an X-10 ON signal to raise the temperature 1 degree, and an X-10 OFF signal to lower the temperature 1 degree. To do this, include the following command in the "On Actions" for the desired X-10 signal:

  Do macro #14 (Increase HVAC Temp 1 Deg)

  In the "Off Actions" for the desired X-10 signal, include this command :

  Do macro #15 (Decrease HVAC Temp 1 Deg)

Step 7 - Configure The TX10-B Thermostat

The TX10-B thermostat must be set up in the following manner (these are the thermostat's default settings).

- The "Preset Dim Command Mode" must be ENABLED.

- The "Ack Message Mode" may be either ENABLED or DISABLED. However, we recommend DISABLING it to reduce the number of X-10 signals on the powerline.

- The "Echo Command Mode" may be either ENABLED or DISABLED. However, we recommend DISABLING it to reduce the number of X-10 signals on the powerline.

- The "Safe Command Mode" must be OFF. If you wish to use the thermostat in the Safe Command Mode, you will have to modify the macros in this example to send each X-10 signal twice instead of just once.

- The "Unit Code On/Off Mode" mode may be either ENABLED or DISABLED. However, we recommend DISABLING it.

Macro 18 "Initialize HVAC System" makes these settings by sending the appropriate commands to the thermostat. This normally only needs to be done once. However, you may want to run the macro periodically to ensure the thermostat hasn't inadvertantly been changed. You could include it in the power failure recovery event, but it will slow down the recovery process. Alternatively, you might put it in a scheduled event that runs at, say, 3:00 AM each day.

Step 8 - Configure PC Software

- From the "Configure" menu, choose "Thermostat". Select the number of zones you want HomeVision to be able to automatically control. If you need more than two zones, HomeVision will not be able to run a time/temperature schedule for the additional zones. For each zone, select "RCS TX10" (there is not a separate selection for the TX10-B thermostat). Finally, select the house code for each thermostat (Note: this house code is only used when you're using the thermostat control screen in the PC software to directly control the thermostat).

- If you want to be able to control the thermostat(s) from your TV, choose "AC/Heating TV Screen" from the "Advanced" menu. Check the "Enable Video (TV) Control Screen" box. Select the variables which contain the HVAC status and the macros which control the TV screen. For this example, you would select the following variables and macros:

  Zone 1 Variables:

1 - Control Mode	=	#1: HVAC Control Mode
2 - HVAC Mode	=	#2: HVAC System Mode
3 - Fan Mode	=	#3: HVAC Fan Mode
4 - Current Temp Setting	=	#4: HVAC Temp Setting
Current Temperature	=	#5: Current Temp

  Zone 1 Macros:

1 - Change Control Mode	=	#10: HVAC Screen Control Mode
2 - Change HVAC Mode	=	#11: HVAC Screen System Mode
3 - Change Fan Mode	=	#12: HVAC Screen Fan Mode
4 - Change Current Setting	=	#13: HVAC Screen Temp Setting

- If you want to set a time/temperature schedule for the thermostat(s), select "Thermostat Schedule" from the "Advanced" menu. Enter the time/temperature schedule for each mode (Heat, Cool, and Auto) of each zone.

- Download the schedule to the controller.
- Enjoy!

Using a Rain Gauge with HomeVision

Several companies make rain gauges that can be used with HomeVision. This article explains how to set up the "Rainew Tipping Bucket Rain Gauge" from RainWise. The same approach can be used with other gauges.

Rain Gauge Description

The RainWise rain gauge is a large bucket with a "rocker" in the base that changes position each time 0.01 inches of rain is received. The rocker movement triggers a relay that briefly shorts two wires together. These wires can be connected to a HomeVision input port and used to track the amount of rainfall.

Unfortunately, the wires are only shorted together for a small amount of time, possibly under 20ms. HomeVision could miss such a short pulse if it were performing some other functions at that precise time. This would cause HomeVision to under-measure the amount of rain. This may or may not be a problem, depending on how accurate you want the measurement. This article shows two different setups, the second of which overcomes this problem by the use of a timed relay.

Configuration 1

The easiest configuration is to connect the rain gauge wires directly to a HomeVision input port. One wire would go to the desired input, and the other would go to the HomeVision ground.

In this configuration, the wires will normally be "open", allowing them to be pulled high (to 5V) on HomeVision. Thus, the input will be read as a "high". When the rain gauge triggers, the wires will briefly short, causing the HomeVision input to go "low".

Configuration 2

This configuration adds a timed relay from Elk Products (part number ELK-960). This relay will act as a "pulse stretcher", lengthening the rain gauge pulse to ensure that HomeVision detects it.

Connections are as follows:

- The relay must be powered by a 12 or 24 VDC power source (actually, 9VDC will usually work as well). Connect the power lead to the relay's "+" input, and the ground lead to the "-" input.

- Connect one of the rain gauge wires to the power supply ground wire, and the other wire to the relay's trigger input.

- Connect the relay to HomeVision as follows: connect one wire from the relay's common output to a HomeVision ground point; connect another wire from the relay's "normally open" output to the desired HomeVision input port.

Make the following settings on the relay:

- JP1: Seconds
- JP2: 1-shot mode
- JP3: Begin
- JP4: Mode B
- JP5: Negative trigger

- JP6: 12 or 24 V, depending on your power source
- Adjust R3 to a timeout setting of about 5 seconds

Here's how this will work. The rain gauge wires will normally be "open", which causes the relay to be off. Since the "normally open" relay output (which is open when the relay is off) is connected to HomeVision, the open HomeVision input will be pulled high (to 5V) by HomeVision. Thus, the input will be read as a "high".

When the rain gauge triggers, its wires will briefly short. This will cause the timed relay to trigger, and it's "normally open" output will close to ground, in turn causing the HomeVision input to go "low". The rain gauge wires may very quickly open again, but the relay will remain triggered for the duration of it's timeout period. Since we set it to 5 seconds, the relay will remain triggered for 5 seconds. When this time is up, the relay goes off and the HomeVision input port returns high.

With this setup, HomeVision can only detect one pulse within the relay timeout period of 5 seconds. However, it would take a VERY heavy rainstorm to trigger the rain gauge more quickly than this. 0.01 inches of rain every 5 seconds equates to 36 inches per hour!

Software Setup

In the HomeVision software, you should put your rain tracking code in the "low" actions for the input port. Each time 0.01 inches of rain occurs, the port will go low, and your actions will be performed.

Let's assume you want to track the total amount of rainfall each day. To do this, create two variables called:

- Today's rain: Inches
- Today's rain: Hundredths

Put the following commands in the "low" actions for the input port:

```
Increment var #1 (Today's rain: Hundredths) [roll over at 255]
If
   Var #1 (Today's rain: Hundredths) = 100
Then
   Var #1 (Today's rain: Hundredths) = 0
   Increment var #2 (Today's rain: Inches) [roll over at 255]
End If
```

Each time this runs, the hundredths value will be incremented. When it reaches 100 (meaning it has rained a full inch), we reset the hundredths value to zero and increment the inches value.

You should also create a scheduled event that runs every day at midnight and resets the variables to zero, like this:

```
Var #1 (Today's rain: Hundredths) = 0
Var #2 (Today's rain: Inches) = 0
```

You can also display the data on HomeVision's weather conditions TV screen. Add these two weather conditions commands to the end of the input port actions shown previously:

```
Today's rainfall (hundredths) = var #1 (Today's rain: Hundredths)
Today's rainfall (inches) = var #2 (Today's rain: Inches)
```

These commands put your variables into the corresponding "weather variables" that are automatically displayed on the TV screen.

Conclusion

It's quite easy for HomeVision to track rainfall. You could extend this by tracking weekly, monthly, or annual rain amounts as well, just be adding a few variables and commands. You could also use this information to adjust your lawn watering schedule.

Pop-up Surveillance Camera On Your TV

Would you like to automatically see who's at your front door, even before they ring the doorbell? It's easy to do! Just follow the simple steps described below.

CAMERA

First, a camera must be installed. The camera is mounted so that a person standing near the front door can be seen. If the camera will be exposed to the rain, you should use a specially designed outdoor camera. Alternatively, you could mount the camera in a weatherproof enclosure. If the camera is in a well protected area, you can get by with a standard camera. However, you may want to seal all the openings with a clear silicone caulk as a precaution. As long as you can turn on an outside light, you do not have to be concerned about low-light sensitivity when choosing a camera.

A cable is run from the camera video output to a television or a video modulator. I recommend using RG-6 coax, although shielded twisted pair may be sufficient. A second cable is required if the camera has audio output. Low voltage power cable (9 to 12 VDC for most cameras) must be run to the camera's power input.

MOTION SENSOR

A motion sensor is positioned to detect someone approaching the front door. If possible, avoid aiming it at trees or plants, as movement caused by the wind can trigger some sensors.

The motion sensor also requires low voltage power. In most cases, the camera and motion sensor can be run off the same power cable.

The motion sensor's output must be routed to the home controller. There are several ways to accomplish this. Wire can be run to connect the motion sensor output directly to a digital input on HomeVision. Or, the output of the motion sensor can be connected to an X-10 Powerflash module. This module generates X-10 On and Off signals when the motion sensor contacts open and close.

Of course, the Powerflash module must be connected to the AC power line in order to generate the X-10 signal. If AC power is not available at the motion sensor's location, wire can be run to a convenient location where the Powerflash module can be installed. For instance, in our home we ran wire from the motion sensor to the security system closet where AC power is available, and installed the Powerflash module there.

There is one potential drawback to using the X-10 approach: the Powerflash module continues to output X-10 signals each time motion is detected. If you plan to send other X-10 commands in response to the motion detector (such as turning on a light or sounding a chime), the Powerflash signals may flood the power line, causing a delay in the X-10 light or chime command being received. This has not been a major problem for us, however.

TELEVISION

There are also different options for supplying the camera video to the television. The camera output can be connected directly to an auxiliary input on the TV, or a video modulator can be used to add the camera video on a TV channel. In this case, the camera output is connected to a video modulator. The modulator is then set up to place the camera output on an unused channel and add it to the cable signal. By modulating the camera video, it can also be distributed to other TVs in the house.

There are several ways to configure the IR transmission in order to control the TV. HomeVision's IR output can be routed through an IR connecting block to a mini-emitter, which is attached to the television's IR receive window. Alternatively, the IR output can be routed to an IR "blaster" (a transmitter that floods the whole room with IR). The transmitter must be positioned to face the TV. However, some of these devices are no longer available, as the trend is towards mini-emitters. If you want to control TVs throughout your house, you'll also need an IR distribution system.

Finally, for effective IR control, HomeVision must be able to detect the current state of the TV (on or off). This can be accomplished by using a high frequency probe (such as the CompCo Engineering TV “Sniffer” Probe). The probe is located behind the TV, where it can detect high frequency elecromagnetic energy that emanates from the TV when it is on. The probe has a digital output, and also requires 5 VDC and ground. The output is attached to a digital input on HomeVision. 5 VDC and ground can be provided from Port C on HomeVision.

OUTSIDE LIGHT

The outside light should be X-10 controlled. The easiest was to do this is to replace the existing switch with an X-10 wall switch.

HOME CONTROLLER

In this example, a Powerflash module is used to transmit the motion detector output, and the camera video is modulated onto channel 72.

IR Learning

HomeVision must be able to transmit the following IR signals:

- TV On
- TV Last Channel
- TV Number 7
- TV Number 2

To learn these IR signals, first select "IR Signals" from the HomeVision "Objects/Events" menu, and add as many IR signals as you plan to learn. It is best to give each one a meaningful name. After you have added the IR signals, click on "Done", and then select "IR Learn" from the "Other" menu. Click the "Learn" button and follow the on-screen instructions. When the signal has been learned, select the IR signal number/name where you wish to save the signal and click on the "Save" button. Repeat the process for each signal.

Timer

A timer is needed because the motion detector continues to send signals as long as something is moving. We don’t want to keep executing the actions over and over when it is still the same person at the door. In our example, the timer is set for one minute, so that after one minute goes by with no motion detected, the camera video display is shut off.

To create the timer, select "Timers" from the "Objects/Events" Menu. Select a timer for use or add a new one by clicking the "Add" button. Enter a name for the timer in the name field. We used timer number 1 and named it "Motion Sensor". For this application, timer actions do not need to be defined and enabled. Click the "Done" button to exit.

Flag

A flag is used to remember the initial state of the TV so it can be returned to that state when the event is over. The flag is set up by selecting "Flags" from the "Objects/Events" menu. Select or add a flag and name it. Click "Done" to exit.

Digital Input for the TV Probe

The digital input for the TV probe must also be configured by opening the "Input Port" window and selecting, naming and enabling the correct port.

X-10 Actions

From the "Objects/Events" menu select "X-10 Modules". Select the Powerflash module's house and unit code. Establish actions for the Powerflash module's X-10 Off signal by double-clicking on the "Off Actions Defined" field in the "X-10 Modules" screen. The code used to define the appropriate actions is shown in the box accompanying this article. When you have finished entering the action code, make sure the "Module Enabled" and "Off Actions Enabled" fields are checked in the X-10 Modules screen.

Note several features of this setup:

- It turns the TV on if its not already on. After motion stops, it returns the TV to its previous state.
- It turns the outside light on if it's dark, then turns if off after 1 minute with no motion being detected.
- It sounds a chime module the FIRST time motion is detected

CONCLUSION

We hope you will enjoy using this home automation application. Of course, there are numerous modifications that can be made to customize this idea for your particular situation. For instance, a similar function could be implemented using a driveway sensor. Have fun!

Example Parts List
Camera
Motion Sensor
Powerflash module
HomeVision controller
IR connecting block
Mini-emitter
TV sniffer probe
X-10 chime module
X-10 wall switch

Sample Schedule Code

NOTE: This code is located in the Powerflash module's "Off Signal" actions and is performed each time the corresponding X-10 Off signal is received.

```
;The front door outside motion sensor just detected motion.
If
   Timer #1 (Motion Sensor) is not running
   ;This timer will be running if we previously detected motion and are waiting
   for it to time out before turning the light back off and returning the TV to
   its previous state.
Then
   ;This must be the first detected motion.
   ;Turn the TV to channel 72 to view the front door camera.
   If
```

```
      Input Port C-1 (TV On/Off Probe) is low
      ;The port is low if the TV is currently on.
   Then
      Set flag #1 (TV On/Off State)
   Else
      ;The TV is off, so turn it on.
      Clear flag #1 (TV On/Off State)
      Transmit IR signal #10 (TV Power) once
   End If
   Transmit IR signal #7 (TV 7) once
   Transmit IR signal #2 (TV 2) once
   If
      Time is dark (after sunset & before sunrise)
   Then
      ;It's dark outside, so turn the porch light on.
      X-10:A 1 (Outside Entry Light) FORCE ON
      ;Wait 1 second then try turning light on again.  (In case motion sensor
      is flooding the power line with X-10 signals)
      Delay for 1.000 seconds
      X-10:A 1 (Outside Entry Light) FORCE ON
   End If
   ;Sound chime module to alert us to the visitor.
   X-10:A 2 (Chime Module) FORCE ON
End If
;
;Load timer to time out after 1 minute without motion.
;Note that this command resets the timer each time motion is detected.
Wait 0:01:00.00 with timer #1 (Motion Sensor), Then:
   ;These actions are only performed after 1 minute without motion:
   If
      Flag #1 (TV On/Off State) is cleared
   Then
      ;Turn TV back off (its original state).
      Transmit IR signal #10 (TV Power) once
   Else
      ;Return TV to previous channel which the viewer was watching.
      Transmit IR signal #11 (TV Channel Return) once
   End If
   If
      Time is dark (after sunset & before sunrise)
   Then
      X-10:A 1 (Outside Entry Light) FORCE OFF
   End If
End Wait
```

QUESTION: How can I know if it's a holiday? I'd like to perform (or skip) certain events on some days.

ANSWER: If the date is the same each year, it's easy. Simply use a "Date" condition in an If-Then statement. In the date condition, specify the month and date of the month. For example, to know if it's New Year's Day, check the boxes for the month "January" and the date "1". This condition will only be true when the date is the first and the month is January. This is how it would appear in your schedule:

```
If
   Date is: (Dates= 1) (Months= Jan)
Then
```

Here are some of the holidays:

New Year's Day :	January 1	St. Patrick's Day :	March 17
Groundhog Day :	February 2	Flag Day :	June 14
Lincoln's Birthday :	February 12	Halloween :	October 31
Valentine's Day :	February 14	Veteran's Day :	November 11
Washington's Birthday :	February 22	Christmas Day :	December 25

What if the date moves around? For example, Labor Day is the first Monday in September, and will occur on a different date each year. Here's how to handle this:

1) Determine the range of dates it could fall on. In this case, the first Monday of a month always falls on the first through seventh day of the month.
2) Enter a date condition with the following items selected:
   - All seven of the allowed dates (in this example, that's 1, 2, 3, 4, 5, 6, and 7).
   - The day of the week (Monday)
   - The month (September)

This condition will only be true on a Monday in September where the date is the first through the seventh. Since this is the definition of Labor Day, it will be true every Labor Day, but never on any other day. This is how it would appear in your schedule:

```
If
   Date is: (Days= M) (Dates= 1,2,3,4,5,6,7) (Months= Sep)
Then
```

This same method can be used for other holidays. The key is determining what dates of the month the holiday can fall on. The following table shows some holidays that move around each year:

HOLIDAY	WHEN IT OCCURS	POSSIBLE DATES
M. L. King Birthday	3rd Monday in January	15-21
President's Day	3rd Monday in February	15-21
Mother's Day	2nd Sunday in May	8-14
Memorial Day	Last Monday in May	25-31
Father's Day	3rd Sunday in June	15-21
Labor Day	1st Monday in September	1-7
Thanksgiving (Canada)	2nd Monday in October	8-14
Columbus Day	2nd Monday in October	8-14
Election Day	Tuesday after 1st Monday in Nov.	2-8
Thanksgiving (U.S)	4th Thursday in November	22-28

Why isn't Easter listed? Well, Easter occurs on the first Sunday after a full moon on or after March 21. It can fall anywhere from March 22 to April 25. The only way to handle this is to get a calendar for the next several years and look up the dates. Then, enter separate If-Then statements for the actual date each year. To check the year, put the current year (its last two digits) into a variable, then check the value of the variable. For example, in 1996, Easter occurs on April 7. Here's how to enter the condition:

```
Var #1 (Year) = current year
If
   Var #1 (Year) = 96
   And Date is: (Dates= 7) (Months= April)
Then
```

With these tricks, HomeVision can determine any holiday and act accordingly!

Using the Xantech 791-40 Amplified Connecting Block with HomeVision

The HomeVision infrared transmit jack (IR TX) can directly drive the Xantech 791-40 amplified connecting block.

Requires: Cable with 3.5mm mono or stereo plug (mono preferred) on one end, and wires on the other end.

Connections: Plug the plug into HomeVision's IR TX jack. Connect the wires to the 791-40 terminal block as follows:

For Mono Plug:
- Striped wire (the plug's tip) to "Input".
- Plain wire (the plug's body) to "Ground".

For Stereo Plug:
- Red wire (the plug's tip) to "Input".
- Ground wire (the plug's body) to "Ground".
- Gray wire (the plug's center ring) is left unconnected.

NOTE: If the wires don't match these descriptions, use an ohmmeter to determine which wires connect to the specified portions of the plug.

WARNING:

DO NOT CONNECT HOMEVISION TO THE 791-40 USING A CABLE WITH 3.5mm PLUGS ON EACH END. THIS COULD DAMAGE HOMEVISION, THE 791-40, AND/OR THE 791-40's POWER SUPPLY.

The 791-40's "Aux" input jack accepts a 3.5mm stereo plug (the body is 12V, center ring is ground, and the tip is the IR TX input). This "Aux" input **cannot** be used with HomeVision for the following reasons:

- If you plug a mono plug into the jack (regardless of whether the other end of the cable is connected to anything or not), **it will short the 791-40's power supply to ground**. It will overheat and may be permanently damaged.
- If you plug a stereo plug into the jack and the other end of the cable into HomeVision, improper operation will result. The cable will connect the 791-40's 12V line to HomeVision's ground, causing significant voltage shifts. The 791-40 may still work erratically and/or it's LED may light when HomeVision is transmitting to it, but this setup should never be used.

Refer to Note 5 in the 791-40's installation instructions for more information.

CREATING YOUR OWN CUSTOM VIDEO SCREENS

INTRODUCTION

With HomeVision, you can create your own video screens. You can even use your remote control to control them. This article shows how to create a simple screen that displays the time and date. It also shows how to create a "menu" that you select from with your remote. Your remote can perform different functions depending upon which screen is being displayed, allowing virtually unlimited control options. This article also shows how to extend this to create virtually any screen you can think of.

OPERATION OVERVIEW

Here's a summary of how the custom screens will work once they're set up:

- The screen drawing commands will be contained within a separate periodic event for each screen. The periodic event will be set to run "each loop", but will initially be <u>disabled</u>.

- To display the screen in your schedule, you'll:

  1) Set the variable "Custom TV Screen #" to the ID number of the screen you want to display (you'll have a unique # for each screen).
  2) Run the macro "Start Custom TV Screen". This macro will:

     - Initialize the video system and create a blank screen to draw on.
     - Disable all the periodic events that draw custom screens (this ensures they won't be drawn simultaneously).
     - Enable the periodic event that draws the desired screen (as specified in the variable "Custom TV Screen #").

     The enabled periodic event will then draw the desired screen each time it runs (about every 20ms). By repeatedly drawing the screen in the periodic event, any data which changes will be immediately updated.

- Actions to perform when the user presses an infrared remote button while viewing the screen will be contained in the IR signal actions for the respective button. This could display a different custom or built-in screen, run a macro, shut off the video system, or just about anything else you can think off.

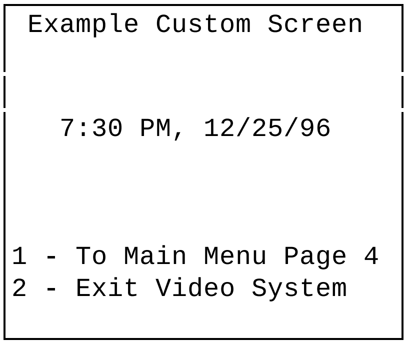
- To shut off the custom screen in your schedule, run the macro "Stop Custom TV Screen". This macro will:

  - Disable all the periodic events that draw custom screens.
  - Set the variable "Custom TV Screen #" to 0 to indicate that no custom screens are being displayed any more.

- The "power failure recovery" event will also run the macro "Stop Custom TV Screen" every time power is restored to the unit. This ensures it starts up properly.

EXAMPLE SCREEN

This article develops a simple screen to the display the time and date and a short menu, as shown below:



```
 Example Custom Screen


   7:30 PM, 12/25/96



1 - To Main Menu Page 4
2 - Exit Video System
```

This screen is included in the example schedule shipped with HomeVision. The object numbers (macros, periodic events, etc.) are different than those used here, but their names and the concepts are the same. You can view it from main menu page 4 by pressing button #2.

CREATING THE SCHEDULE

Step 1 - Create Variable

Create one variable:

#	NAME	INITIAL VALUE	DESCRIPTION
1	Custom TV Screen #	0	Indicates the number of the custom screen being displayed. Each screen must have a unique ID # (preferably starting at 1 and progressing up). It will be set to 0 (zero) when no custom screens are being displayed. Note that this is different from the built-in screen numbers.

Step 2 - Create Periodic Event

Create one periodic event:

#	NAME	RATE	INITIAL STATE	DESCRIPTION
1	Custom TV Screen #1	Each loop	Disabled	Repeatedly writes all the desired information for this custom screen. By repeatedly drawing the screen, any data which changes will be immediately updated. This event must initially be disabled so that it does not draw anything. It will enabled when it's time to start drawing. This event should not clear the screen, as flickering will result when it is repeatedly cleared and redrawn.

NOTE: If you want to create more than one custom screen, use a separate periodic event for each.

Enter these actions for the periodic event:

Video: Set cursor row to 1

```
    Video: Set cursor column to 2
    Video: Display text 'Example Custom Screen'
    ;
    Video: Set cursor row to 4
    Video: Set cursor column to 3
    Video: Display current time
    Video: Display text ', '                              {note that there's a space after the comma}
    Video: Display current date
    ;
    Video: Set cursor row to 8
    Video: Set cursor column to 1
    Video: Display text '1 - To Main Menu Page 4'
    ;
    Video: Set cursor row to 9
    Video: Set cursor column to 1
    Video: Display text '2 - Exit Video System'
```

NOTE: After you enter these and leave the actions entry screen, the event will be enabled automatically. Be sure to disable it.

Step 3 - Create Macros

Create these macros:

#	NAME	DESCRIPTION
1	Start Custom TV Screen	Starts displaying the custom screen specified in variable #1 (Custom TV Screen #).
2	Stop Custom TV Screen	Stops all custom screens from being displayed by disabling the corresponding periodic events.

Enter these actions for the macros:

MACRO #1: Start Custom TV Screen

```
    ;In this example, we'll display the screen on a blank background (not on incoming video):
    Video: Start internal video mode
    ;
    ;Display built-in screen #0, which is a blank screen to write your own information on:
    Video: Display video screen #0
    ;
    ;Next, we disable all the periodic events that are used to draw custom screens:
    Disable periodic event #1 (Custom TV Screen #1)
    ;
    ;Finally, we determine which custom screen to display and enable its periodic event.
    ;The desired screen # must be in variable #1(Custom TV Screen #) before this macro runs.
    If
        Var #1(Custom TV Screen #) = 1
    Then
        Enable periodic event #1 (Custom TV Screen #1)
    End If
```

NOTE:

In this example, there's only 1 custom screen. Therefore, there's only one command to disable it, and one If-Then statement to enable it. In this case, it seems odd to disable the event, then re-enable it. However, this concept makes much more sense when you have multiple custom screens, and we wanted the example to show this. Here's how this macro would look if you had three custom screens:

```
;In this example, we'll display the screen on a blank background (not on incoming video):
Video: Start internal video mode
;
;Display built-in screen #0, which is a blank screen to write your own information on:
Video: Display video screen #0
;
;Next, we disable all the periodic events that are used to draw custom screens:
Disable periodic event #1 (Custom TV Screen #1)
Disable periodic event #2 (Custom TV Screen #2)
Disable periodic event #3 (Custom TV Screen #3)
;
;Finally, we determine which custom screen to display and enable its periodic event.
;The desired screen # must be in variable #1(Custom TV Screen #) before this macro runs.
If
    Var #1(Custom TV Screen #) = 1
Then
    Enable periodic event #1 (Custom TV Screen #1)
End If
If
    Var #1(Custom TV Screen #) = 2
Then
    Enable periodic event #2 (Custom TV Screen #2)
End If
If
    Var #1(Custom TV Screen #) = 3
Then
    Enable periodic event #3 (Custom TV Screen #3)
End If
```

MACRO #2: Stop Custom TV Screen

```
;First, we disable all the periodic events that are used to draw custom screens:
Disable periodic event #1 (Custom TV Screen #1)
;
Then, set variable a to 0 to indicate that no custom screens are being displayed:
Var #1(Custom TV Screen #) = 0
```

NOTE: As noted with the first macro, you would disable the other periodic events here also if you have more than one custom screen.

Step 4 - Create Infrared Signals

In this example, we're only using two infrared buttons (1 and 2). However, we recommend you set up all 16 of the signals that are used to control the video system. This will make it easier to later assign actions to them for other purposes.

To set up these signals, you must know the device code and key code for each button on your remote. If you're not sure what this means, refer to the owners manual chapter 14 (under "IR Signal Reception Details" and "Configuring the Controller to Control the Video Screen System"). Once you schedule has been set up to control the video system, you can view the actual device codes and key codes on the "Video Remote Control" screen located under the "Configure" menu in the PC software.

The default HomeVision device and key codes are shown in the following table (these are for a Hitachi TV, One-for-All remote code 145). These are the same as used in the example schedule. If you've remapped any of the keys (such as the "enter" or "cancel" keys), or selected a different remote device, use their actual codes, not the ones shown here.

NOTE: All of these signals have these same settings:

- IR Signal Type: Standard
- Device Code: 10

#	NAME	KEY CODE	DESCRIPTION
0	Remote Input 0	48	This is the number 0 button.
1	Remote Input 1	176	This is the number 1 button.
2	Remote Input 2	112	This is the number 2 button.
3	Remote Input 3	240	This is the number 3 button.
4	Remote Input 4	56	This is the number 4 button.
5	Remote Input 5	184	This is the number 5 button.
6	Remote Input 6	120	This is the number 6 button.
7	Remote Input 7	248	This is the number 7 button.
8	Remote Input 8	32	This is the number 8 button.
9	Remote Input 9	160	This is the number 9 button.
10	Remote Input Channel Up	152	This is the "channel up" button.
11	Remote Input Channel Down	24	This is the "channel down" button.
12	Remote Input Volume Up	72	This is the "volume up" button.
13	Remote Input Volume Down	168	This is the "volume down" button.
14	Remote Input Enter	80	This is the "enter" button.
15	Remote Input Cancel	208	This is the "cancel" button. Since some remotes don't have a button labeled "cancel", this may be assigned to the "mute" button.

Enter these actions for these two IR signals:

INFRARED SIGNAL #1: Remote Input 1

```
If
    Var #1(Custom TV Screen #) = 1
Then
    ;Do whatever you want to happen when displaying custom screen #1 and button 1 is pressed.
    ;In this example, we want to go to main menu page 4, which is built-in screen #55.
    ;But first, we must stop the custom screen from being displayed:
    Do macro #2 (Stop Custom TV Screen) once
    Video: Display video screen #55
```

```
End If
```

INFRARED SIGNAL #2: Remote Input 2

```
If
    Var #1(Custom TV Screen #) = 1
Then
    ;Do whatever you want to happen when displaying custom screen #1 and button 2 is pressed.
    ;In this example, we want to shut the video system off entirely.
    ;But first, we must stop the custom screen from being displayed:
    Do macro #2 (Stop Custom TV Screen) once
    Video: Stop video mode
End If
```

NOTES:

It's important to understand when the controller will execute these actions. If these buttons were not the same ones used to control the video menu system, then they would execute every time the buttons were pressed. However, buttons set up to control the video menu system behave differently:

- When no video screens are being displayed (the video system is turned off), the actions are performed whenever the corresponding button is pressed.
- When you're displaying your own custom screen using built-in screen #0, these actions are also performed whenever the corresponding button is pressed.
- When you're displaying any of the built-in screens (except for #0), these actions ARE NOT PERFORMED. Instead, the buttons are used to control the built-in screens.

HomeVision is set up this way so the buttons can control the built-in video system when you are using it, AND do other things (like control your own screens or perform macros) when you're not using it.

In this example, when the custom screen is being displayed, these infrared signal actions will be performed. We use the If-Then statement to determine which custom screen is being displayed, then perform whatever actions we want. Of course, in this example there's only one custom screen. But if you had more, you would have one If-Then statement for each screen, like this:

```
If
    Var #1(Custom TV Screen #) = 1
Then
    ;Do whatever you want to happen when displaying custom screen #1 and button 2 is pressed.
End If
;
If
    Var #1(Custom TV Screen #) = 2
Then
    ;Do whatever you want to happen when displaying custom screen #2 and button 2 is pressed.
End If
;
If
    Var #1(Custom TV Screen #) = 3
Then
```

```
        ;Do whatever you want to happen when displaying custom screen #3 and button 2 is pressed.
    End If
```

With this setup, the buttons will behave differently depending on which screen you're currently watching. This give you tremendous power when creating your own screens.

Step 5 - Power Failure Recovery Event

Enter the following command in the power failure recovery event:

```
    Do macro #2 (Stop Custom TV Screen) once
```

This is done in case power goes out while you're displaying a custom screen. When the controller restarts, the periodic event for the screen will still be enabled, and it will write to the screen. When you then start the video menu system, the periodic event will overwrite it, causing flickering (or alternating between the two screens). Running this macro prevents this by disabling all the periodic events.

CONTROLLING YOUR SCREEN

We're now ready to use this screen in a schedule. Here's how to start and stop displaying it.

Starting a custom screen

Anywhere in your schedule you want to start this screen, enter these commands:

```
    Var #1(Custom TV Screen #) = 1
    Do macro #1 (Start Custom TV Screen) once
```

That's all there is to it. First, you set the variable to the screen number you want to display, then you run the macro to start it.

You could also start the screen using main menu page 4 of the built-in video menu system. For details on using main menu page 4, refer to the file "Screens" provided on the HomeVision installation disks. Basically, it displays options on the screen and allows you to perform a macro when the corresponding button is pressed. To display a custom screen from there, create a macro (perhaps called "Start Custom TV Screen #1"), and include in it the two commands shown above. Then, use the "Custom TV Menu" screen, located under the "Advanced" menu in the PC software, to assign this macro to the desired entry on menu page 4. When the user presses the corresponding button, the macro will be performed and the custom screen appears. This is the method used in the example schedule provided with HomeVision to start the custom screens.

Stopping a custom screen

Whenever you want to stop any custom screen, enter this command:

```
    Do macro #2 (Stop Custom TV Screen) once
```

In this example, we did this in the two infrared signal events. After stopping the custom screens, you might also want to:

- shut off the video system (like we did for IR signal #2), or

- go back to the menu system (like we did for IR signal #1), or
- go to another custom screen (by setting variable #1 to the new screen number and running macro #1 (Start Custom TV Screen) again).

CUSTOM SCREEN NOTES

- Remember that you must always display built-in screen #0 when showing your own screen. Screen 0 is a blank screen for you to write on. If you display any other built-in screen, HomeVision will repeatedly redraw the specified screen (as it is part of the video menu system), making it impossible for you to write on. In fact, if you write data to one of these screens, it will probably be immediately overwritten and never be seen.

- When you're first creating your own screens, you'll probably run into this problem. Either when you first run the schedule, or when you try to display the menu system, you'll see the screen start flickering or alternating between two screens. What's happened is that one of the periodic events that draw your screens is enabled at the same time as the built-in menu system is active. Since both repeatedly write to the screen, they overwrite each other, resulting in a mess.

  If you entered the power failure recovery action we recommended, you can remove and reapply power to stop this. However, you'll need to figure out where you went wrong and correct it. Check the periodic events in your schedule and make sure they are disabled (note that when you close the actions entry screen, the periodic event will be enabled automatically, so this is a common source of this problem).

- You can use an If-Then statement to determine what to write to the screen. For example, you could look at the state of a flag to determine the security system status to display, like this:

  ```
  Video: Display text 'System is '                {note blank space at end}
  If
      Flag #1 (Security System State) is set
  Then
      Video: Display text 'armed   '              {note 3 blank spaces at end}
  Else
      Video: Display text 'disarmed'
  End If
  ```

  If the status changes while you're watching the screen, you'll see it change. When doing this, be sure that the possible text values are the same length. That way, they completely overwrite what was previously on the screen. In this example, consider what could happen if 'armed   ' did not contain 3 blank spaces at the end. If the system was initially disarmed, then got armed while displaying this screen, the word 'armed' (without the blanks) would not completely overwrite the word 'disarmed', resulting in:

  ```
  System is armedmed
  ```

  being displayed. By putting the three blank spaces at the end, we completely overwrite the previous value.

- Refer to the file "Screens" for more information on the video screens. It includes a complete listing of the built-in screens and their numbers.

CONCLUSION

You can use these methods to create virtually any screen you want. With some work and creativity, you can create screens to:

- Show your own menu system (bypassing the built-in system entirely if you like).
- Perform your favorite macros.
- Play different radio stations.
- List the compact disks in your 300 disk changer by title and play the selected one.
- Report the last few times the security system was armed and disarmed.
- Show status of your home (alarm status, thermostat mode, house mode, # of phone calls since you last left home, etc.)
- Display weather information (from a directly-connected weather station or downloaded over the serial interface from your PC).

Good luck and have fun!

RESTORING X-10 LIGHTS TO THEIR PRIOR LEVEL

INTRODUCTION

Many people like to use their home automation system to temporarily turn lights on or off, then return them to their prior level. Here are some common uses:

- Have an “All Lights On” command which you run if you hear a noise and want to scare off a potential intruder. But later, you’ll want to put the lights back where they were. You probably don’t want to just do “All Lights Off”, as that will turn everything off.

- A porch light is left on at a dim level throughout the night. But when a motion sensor detects motion, the light goes to full brightness. After a few minutes, the light should return to the previous level (not just be turned off).

Both of these scenarios require the automation system to store the initial level of the light(s), then later return the lights to the same level. HomeVision is one of the very few systems that provide this capability. Most other systems would, at best, require you to use flags to keep track of the on/off state of each light, then turn the light on or off based on these flags. You could never keep track of the level of a dimmed light and restore it with such systems.

HOMEVISION COMMANDS

HomeVision has two primary commands to implement this feature:

```
X-10 Module ## - Put Current Level Into Variable ##
X-10 Module ## - Set To Level In Variable ##
```

The first command puts the X-10 light’s current level into the specified variable. Use it to store the light level prior to changing it. The second command sets the light to the level contained in the specified variable. Use it to restore the light to the previous level.

EXAMPLE

Here’s an example of how to use these commands. It shows how to implement a simple “All Lights On” macro and a “Cancel All Lights On” macro.

Assume you have three X-10 light modules:

A-1 = Family Room Light
A-2 = Dining Room Light
A-3 = Bedroom Light

First, create three variables to temporarily hold the light levels:

Variable #1 = Family Room Level
Variable #2 = Dining Room Level
Variable #3 = Bedroom Level

Next, create the “All Lights On” macro with these commands in it:

```
X-10 A-1 (Family Room Light) - Put Current Level Into Variable #1 (Family Room Level)
```

X-10 A-2 (Dining Room Light) - Put Current Level Into Variable #2 (Dining Room Level)
X-10 A-3 (Bedroom Light) - Put Current Level Into Variable #3 (Bedroom Level)
X-10: Transmit House Code ‘A’ ALL LIGHTS ON

Finally, create the “Cancel All Lights On” macro with these commands:

X-10 A-1 (Family Room Light) - Set To Level In Variable #1 (Family Room Level)
X-10 A-2 (Dining Room Light) - Set To Level In Variable #2 (Dining Room Level)
X-10 A-3 (Bedroom Light) - Set To Level In Variable #3 (Bedroom Level)

Here’s how this works. The “All Lights On” macro first stores the initial light levels into the three variables, then turns them all on. When you later run the “Cancel All Lights On” macro, it will set the lights to the levels specified in the three variables, thereby putting them back where they were.

TECHNICAL NOTE

Different variable values are used to indicate the module's current state (On, Off, or Neutral), as follows:

Off	= 255
Neutral	= 254
On	= 0 to 16 (depending on the light's level)

You don't have to be concerned with this if you use the "Put Current Level Into Variable ##" to get the current level and then use the "Set To Level In Variable ##" command.

CONCLUSION

As you expand your home’s lighting system controls, you’ll run into many other cases where you want to restore a light to a previous level. HomeVision provides this ability, allowing you to develop an elegant, professional lighting system.

Using CompCo Engineering, Inc. Probes with HomeVision

Note: These probes are no longer available. Probes from other manufacturers work similarly, but the wiring colors may be different.

The HomeVision input ports can interface with the various status-sensing probes made by CompCo Engineering, Inc.

Requires: A cable with a 3.5mm stereo jack on one end and wires on the other end.

Connections: Plug the probe's 3.5mm plug into the cable's jack. Connect the cable's bare wires into HomeVision's port B or C terminal block as follows:

- Red wire (the plug's tip) to the 5V position.
- Ground wire (the plug's body) to the ground position.
- Gray wire (the plug's center ring) to one of the eight input positions.

NOTE: If the wire colors don't match these descriptions, use an ohmmeter to determine which wires connect to the terminal block.

Notes:

- The TV Sniffer Probe™ usually has a several second delay after the TV goes off before the probe's output changes. The probe is usually faster on HomeVision port B than on port C. If you're using the Multifunction Expansion Board, port 1 will be faster than ports 2 or 3.

- If you choose to cut the probe's plug off and wire it directly to HomeVision, you will have to identify which wire is which. On one probe we tested, the wires were as follows:

  - Green wire = ground
  - Copper-colored wire = 5V
  - Red wire = output signal

  We found these wires to be difficult to work with. We had to solder another short wire onto each wire before we got electrical continuity. Therefore, we recommend using a separate cable with a jack.

Using an Analog Temperature Sensor with HomeVision

The HomeVision Multifunction Expansion Board's analog inputs are commonly used with analog temperature sensors. This article explains how to read temperature with an LM34 temperature sensor. Other sensors can be used in a similar fashion.

Sensor Description

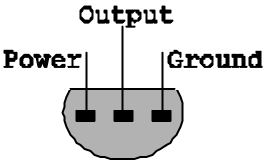
The LM34 is the most commonly used sensor, although other types are available. It can run off of 5-30 VDC, and the output is 10mV per degree F. It comes in several packages, although the three-lead plastic package (TO-92) is suggested. Three part numbers are available:

- LM34DZ
- LM34CZ
- LM34CAZ

The LM34DZ is adequate for most applications; the LM34CZ works over a wider temperature range; and the LM34CAZ is most accurate (but is considerably more expensive). They are available from a variety of sources, including Digi-Key (800-344-4539). An LM35 is also available, but will have lower temperature resolution (10mV per degree C).

Electrical Connections

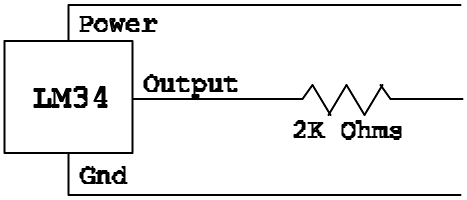
The LM34 requires three wires for power, ground, and signal. The following drawing shows the lead orientation as viewed from the bottom of the TO-92 package.



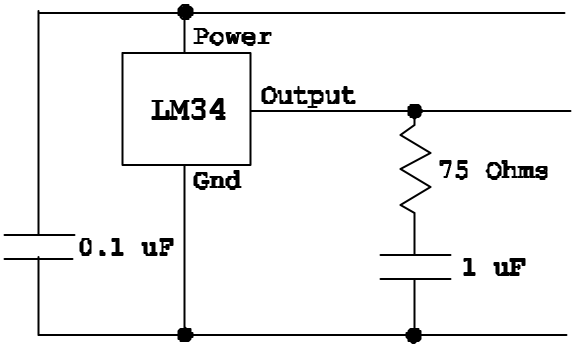
Bottom View

Connect the power and ground wires to the respective positions on the analog input terminal block. Connect the LM34 output to the desired analog input port. Twisted-pair wiring is recommended, with Cat5 cables preferred.

If you're driving a long cable (>40 feet), use either of the following two circuits for better performance:



Circuit For Improved Performance



Circuit For Maximum Performance

The first circuit is adequate in most applications, but optimum performance is achieved with the second circuit. It provides a decoupling capacitor for the power supply, and allows the LM34 to easily drive a large capacitive load (i.e., a long cable). If you're using a pre-made temperature sensor assembly (commonly packaged in a wall outlet package), it probably contains circuitry similar to this.

Software Setup

Use the HomeVision Analog Inputs Summary Screen under the Objects/Events menu to configure the analog input. The analog inputs work from 0 to 5 volts, and are read as 256 discrete voltage levels (or 255 "steps"). Each step is 19.6078mV (5V / 255). You will need to set the "Gain" and "Offset" factors to match the input to the sensor's output. With the proper settings, the value read by HomeVision (0 to 255) will equal the temperature. This process is described below.

Gain

Select the correct gain factor with the following equation:

$$\text{Gain} = \frac{19.6078}{XX}$$

where XX = the sensor's output in mV per degree F

The LM34's output is 10mV per degree F, resulting in a gain of 1.96, as shown:

$$\text{Gain} = \frac{19.6078}{10} = 1.96078 \quad \text{(rounded off to 1.96)}$$

Here's how this will work: The input value read by HomeVision increases by 1 each time the input voltage increases by 19.6mV, and 19.6mV corresponds to a temperature rise of 1.96 degrees F. Thus, we use a gain of 1.96 to multiply the input value change by so that it matches the temperature change. HomeVision will round off the final result to the nearest integer, but you should still use a gain of 1.96 and not 2.00 to minimize the error.

Because the analog input resolution (19.6mV) is coarser than the temperature sensor's (10mV), you will usually see the temperature change in 2 degree increments. This is shown below, where the right-hand column indicates the final value read by HomeVision and equals the reported temperature. Note the this value is never more than 1 degree off from the actual temperature.

Actual	LM34	Input	With 1.96
Temperature	Output	Value	Gain

0	0.000	0	0
1	0.010	0	0
2	0.020	1	2
3	0.030	1	2
4	0.040	2	4
.			
.			
.			
50	0.500	25	49
51	0.510	26	51
.			
.			
.			
100	1.000	51	100

Offset

For the LM34, set the offset factor to zero. Since the LM34 outputs 0 volts at zero degree F, there is no need for an offset. Other sensors may output a different voltage at 0 degrees, necessitating an offset factor. Offset is simply an addition to or subtraction from the value read in.

Calibration

You may want to use a thermometer or other temperature sensor to calibrate the LM34's temperature reading. If you need to adjust the LM34's reading, change the offset factor to make the temperature correct. Do not adjust the gain factor.

Using the Temperature in Your Schedule

First, create a variable to hold each temperature value. If you have multiple sensors, use consecutive variables so they can all be viewed on the same TV screen page.

Anywhere in your schedule you want to read the current temperature from the sensor, use this variable command:

```
Var #1 (Outdoor Temp) = Analog input #1 (Outdoor Temp Sensor)
```

This command reads in the analog input and puts it into the specified variable. Since we set the input's Gain and Offset factors properly, variable #1 now equals the actual temperature. You can read the analog input each time you need to know the temperature, or you can read it periodically. We recommend creating a periodic event that performs the above command. Set the event's rate to "every loop" or "every minute", depending upon how rapidly the temperature might change.

After putting the analog input into a variable, you can use the variable anyplace you want to know the temperature. For example, if you want to turn on a fan when the temperature exceeds 90 degrees, you could do this:

```
If
   Var #1 (Outdoor Temp) > 90
Then
   X-10: A1 (Ceiling Fan) ON
End If
```

That’s all there is to it!

Displaying Messages on Your TV

You can use HomeVision's video system to display messages on your TV. You can do this with commands in your schedule, or by sending serial commands from an attached computer. This article shows how to do both.

We recommend that you first read the video chapter of the owner's manual to ensure you understand how the video system works. If you want to send the commands from a computer, you should also read the serial protocol documentation.

Displaying Messages In Your Schedule

These video commands display the message "Happy Birthday" in the center of the TV screen for 30 seconds:

```
Video: Start internal video mode
Video: Set cursor row to 5
Video: Set cursor column to 6
Video: Display text "Happy Birthday"
Wait 0:00:30.00 with timer #1 (TV message timer), Then:
   Video: Stop video mode
End Wait
```

That's really all there is to it. You start the video system, position the cursor, then write the message. You can shut off the screen using a Wait Timer as shown here, with infrared signals, or a number of other means. You could also display the message overlayed onto incoming video, blink certain characters, change the character size, and much more.

Displaying Messages With Commands Over The Serial Interface

Step 1) Start the video system. The following command initializes it for internal video (text displayed on a solid color background):

```
,L0701  <carriage return>
```

The following command initializes it for external video (text overlayed on the incoming video):

```
,L0700  <carriage return>
```

Note that both of these commands also clear the screen and place the cursor in the top left corner of the screen.

Step 2) Position the cursor. Use the following command to set the cursor to a different row:

```
,L01XX  <carriage return>

where XX is the row number (1 to 11) in Hex format (01 - 0B).
```

Use the following command to set the cursor to a different column:

```
,L02XX  <carriage return>
```

```
where XX is the column number (1 to 24) in Hex format (01 - 18).
```

Step 3) Write the characters. Use the following command to display a single character:

```
,L0CXX  <carriage return>
```

where XX is the ASCII code for the desired character in Hex format. For example:

CHARACTER	XX VALUE
0	30
1	31
9	39
A	41
B	42

Commands over the serial interface can only send a single character. Therefore, you must loop through the message transmitting one character at a time. The cursor automatically moves after a character is written, so you don't have to reposition the cursor. The hardest part of this process is if the message extends beyond one line. You'll have to figure out where to break the line so it doesn't split a work in two.

Step 4) When you're done, use this command to shut off the video system:

```
,L0702  <carriage return>
```

Sending FAXes with HomeVision

Introduction

HomeVision can perform many different actions if you leave it connected to a running computer. One useful function is to send a FAX when an important event occurs. HomeVision does not contain any built-in FAX capabilities; instead, it uses your FAX software by sending the appropriate commands to it.

How To Send A FAX

HomeVision has a variety of commands to support sending a FAX:

- Run program "program.exe"
- Activate window "window name"
- Send keys "keystrokes"

You should review the HomeVision documentation for these commands so you understand them thoroughly before proceeding. With these commands, the controller can perform just about any action you can with the keyboard. There are two different ways to use these to send a FAX:

1) Use these commands to send all the needed keystrokes directly to the FAX software.

2) Use a software "macro" or "batch file" program. These programs can be configured to perform a sequence of actions upon command. Common capabilities include running programs, activating windows, sending keystrokes and mouse movements, waiting for a previous action to complete, and much more. Many of these are available as shareware or even freeware. You could use one of these programs and configure a macro to send the desired FAX. Then, have HomeVision send a single command to run the macro. This approach is more powerful than the first approach because these programs typically provide more commands than HomeVision to control the FAX program.

No matter which approach you choose, you will need to determine the correct sequence of commands needed to send the FAX. The required commands (or keystrokes) will vary greatly depending on your FAX software and how your computer is configured. The best way to determine them is by sending a text FAX with your software and making a record of all the keystrokes required. Here are some tips:

- It usually works best to keep your FAX software running all the time. This way HomeVision does not need to start and stop it. This simplifies timing of the commands, as there's no need wait for the program to start before sending the next commands. You should put the FAX software in your Windows startup folder so it will starts itself automatically.

- In your HomeVision schedule, put all the commands in a macro. Then you can easily test the macro by having it run when you receive a certain X-10 or infrared signal.

- Test the commands under all possible conditions to determine which may not work properly. For example:

    - Does it work OK if the FAX application is minimized? The "activate window" command selects the program icon, but doesn't open it. It stays minimized and subsequent keystrokes don't work as expected. Some programs allow you to restore or maximize the window with keystrokes. For example, Microsoft Word uses the keys Alt-F5 to restore the window and Alt-F10 to maximize the window. To determine if your program can do this, minimize the window, then click on the icon. A popup window will appear that shows the available keystrokes. If available, you can send these keystrokes after using the "activate window" command to restore the window, then proceed with the rest of the keystrokes.

- What happens if the program isn't running? If you don't start it, it obviously won't send a FAX. However, will something else improper happen instead? If another program is open, the keystrokes could cause it to do something you didn't intend.

- What happens if the phone line is busy? Most FAX programs will automatically redial several times, so this usually isn't a problem.

What to FAX

There are two different ways to send information via FAX:

1) FAX a file:

   With this method, you have a file containing the desired information already on the computer. You then tell the FAX software to FAX this file. But how do you create this file? One approach is to create in advance a variety of files each with a different message. One file might be a simple text file reading "Burglar alarm is going off at your house!". Another might be a graphics file containing a fancy picture of your home, along with the message "Water has been detected in the basement". Another approach is to use HomeVision's capability to create text files and write text to them. In this way, the file could contain variable information, such as "The indoor temperature has dropped to 40 degrees !". You could even have HomeVision log all serial messages to a file, then FAX the complete log file.

2) Enter the information directly into the cover sheet:

   Many FAX programs allow you to include text messages on the cover sheet. You can use HomeVision's "send keys" command to write this message.

Examples

These examples assume you leave your FAX software running all the time, which is our recommended method. However, if you don't, you can use a command such as this to start it:

```
Run program "C:\programs\FaxTalk\FaxTalk.exe"
```

Example 1

This example shows how to FAX a file using the QuickLink software supplied with some U.S. Robotics modems. The controller sends commands to the QuickLink program to FAX a file named "Alarm.txt", which already exists on the computer.

Step 1 - Activate the FAX software window

You need to activate the program window so that subsequent keystrokes are sent to it:

```
Activate window "QuickLink"
```

Step 2 - Send keystrokes

Now you send all the keystrokes needed to send the FAX. The notes at the end of each line explain what each command does.

```
Send keys "%+S"                           (selects New FAX from menu)
Send keys "c:\Alarm.txt{ENTER}{ENTER}"    (specifies file name to FAX)
Send keys "5551234"                       (number to send FAX to)
Send keys "{TAB}John Doe"                 (name of FAX recipient)
Send keys "%S"                            (sends FAX now)
```

These commands work with QuickLink; your program will likely be different. You should also note that you could put all these keystrokes in a single "send keys" command. However, spreading them out as done here makes it a little easier to understand what it's doing.

Example 2

This example shows how to send a FAX using the FaxTalk For Windows software. The controller sends commands to the FaxTalk program to FAX a preset coversheet which includes a comment field where new text can be added.

Step 1 - Activate the FAX software window

You need to activate the program window so that subsequent keystrokes are sent to it:

```
Activate window "FaxTalk"
```

Step 2 - Send keystrokes

Now you send all the keystrokes needed to send the FAX.

```
Send keys "%FW"                      (selects New FAX from menu)
Send keys "%N"                       (moves to "Phonebook Names" field)
Send keys "{UP}"                     (selects first entry as recipient)
Send keys "%A"                       (selects "Add to recipient list")
Send keys "%E"                       (opens "Edit Cover Sheet" window)
Send keys "{TAB}X"                   (selects template file "X")
Send keys "{TAB 7}"                  (7 tabs to reach "Comment" field)
Send keys "The alarm is going off"   (enters comment)
Send keys "%O"                       (selects "OK" and closes window)
Send keys "%S"                       (sends FAX)
```

Note several things about this FaxTalk procedure:

- We're selecting the recipient from a "phonebook" using a list box. It is difficult to select a desired entry with only keyboard strokes, as typing the recipient's name does NOT work. Instead, we simply select the first name from the list. You should therefore set up the phonebook so the first entry is the correct destination. You should also choose a name for the entry that will always be first on the alphabetical list (perhaps by starting it with an underscore character). Otherwise, you might later add a phonebook entry at the top of the list, and they receive your HomeVision FAX!

- FaxTalk uses coversheet templates that contain information you would otherwise enter manually. This example selects template "X", which was preset with text indicating that HomeVision is the FAX sender. We then enter a specific message in the comment field.

Example 3

EtCetera is a useful shareware program that performs "batch" commands to control other programs. Listed below is an EtCetera batch file that sends a FAX using QuickLink. This is very similar to example 1, except here EtCetera controls the QuickLink software instead of HomeVision doing it directly.

```
;Start QuickLink. If it's already running, this will activate it:
Run "C:\QLMCW\QLMCW.EXE"
;Wait 5 seconds for it to start up:
Wait 5
;Select the "Send FAX" command from the menu (by entering Alt-Shift-S):
SendKeys "/A/SS"
;Enter the file name to FAX, including its path:
SendKeys "c:\alarm.txt"
SendKeys "{ENTER}{ENTER}"
;Enter the phone number and recipient's name:
SendKeys "5551234"
SendKeys "{TAB}John Doe"
;Finally, send the FAX (by entering Alt-S):
SendKeys "/AS"
```

Once you create the batch file (a simple text file), HomeVision can run it with a single command. For example, if the batch file name is "sendfax.etc" and the EtCetera program is named "Etc_130.exe", you would use this command:

```
Run program "C:\Etc_130.exe sendfax.etc"
```

There are several advantages to having EtCetera control the FAX process instead of HomeVision:

- EtCetera's "Run" command works regardless of whether QuickLink is currently active, minimized, or not running at all.
- Wait statements in the batch file are easier to use than putting them directly in your HomeVision schedule.
- You can modify the batch file at any time without having to reload the HomeVision schedule.

EtCetera and other similar programs are available at many shareware sites on the internet.

Using Elk Sensitive Relays with HomeVision

Elk Products, Inc. makes a variety of sensitive relays that can be controlled by HomeVision output ports A and C. This article explains how to connect these to HomeVision.

Relay Description

This article covers the ELK-924 sensitive relay, but other relays in the same family work the same way. This relay requires only a low (1.2mA) trigger current to switch the relays, which can be provided by HomeVision. The relay also requires a 12VDC or 24VDC power source to power the relay coil. This must be provided by an external power source (not from HomeVision).

The relay has two control (or "trigger") inputs, a positive trigger and a negative trigger. These determine whether the relay is switched by a positive or negative input. Only one of these should be connected. The negative trigger will be used when driven by HomeVision port A, and the positive trigger will be used when driven by HomeVision port C.

Connecting to HomeVision Output Port A

Make these two connections:

- HomeVision Port A ground (terminal block position 10) to the negative power input ("Neg") on the relay. This provides a common ground between HomeVision and the relay. Note that this same point on the relay will also be connected to your external power source.

- Desired HomeVision Port A output to the negative trigger input ("-T") on the relay.

Here's how this is controlled by HomeVision:

- When HomeVision's output port is set high, the output connection is switched to ground. Since this is connected to the relay's negative trigger (which triggers on a low, or ground, input), the relay is energized.

- When HomeVision's output port is set low, the output connection is opened. An open circuit on the negative trigger input causes the relay to become de-energized.

Connecting to HomeVision Output Port C

Driving this relay from output port C is slightly more complicated than using port A. We must use the positive trigger input for the relay to operate properly. However, HomeVision port C cannot provide enough output current to drive the relay this way. Port C can sink up to 20mA to ground, but can only provide 0.3mA at 5 volts. Since the relay requires 1.2mA at 5V, the output is not strong enough. We overcome this problem by adding a resistor to provide the necessary current

Make these two connections:

- Desired HomeVision Port C output to the positive trigger input ("+T") on the relay.

- A resistor (470 to 1000 ohms) from the Port C output to 5 volts (terminal block position 9).

Here's how this is controlled by HomeVision:

- When HomeVision's output port is set high, the output is 5 volts. Since this is connected to the relay's positive trigger (which triggers on a high input), the relay is energized.

- When HomeVision's output port is set low, the output is at ground. This causes the relay to become de-energized.

Other Connections

Refer to the relay's documentation for connecting to the relay outputs.

Tips on Transmitting Serial Data

HomeVision can transmit serial data to an attached computer or other device. Common uses include:

- Logging data to a computer text file
- Using a text-to-speech program to “speak”
- Playing WAV files
- Issuing commands to a thermostat or security system

In order to do these things, you must be able to transmit the required information. That’s quite easy if it’s simple ASCII text. However, it’s often necessary to transmit variable values or binary data. This article shows how to accomplish this. It also provides some tricks on how you might collect and use certain data.

Text To Speech

The most popular use of serial transmissions is for “text to speech”. There are several computer programs that can read in serial data, and then “speak” it using the computer’s sound card. If you route the audio output into your home’s audio distribution system, HomeVision can speak to you anywhere in your house! Home Voice is a popular program that works with HomeVision to do this (and also gives you voice control of your home).

Whether you use Home Voice or another program, the concept is the same: you transmit the text you want to speak to the PC, in the proper format. For example, to have Home Voice say “alarm is armed”, you need to transmit this:

```
speak alarm is armed  <carriage return>
```

This is done using the following two serial commands:

```
Serial xmit: "speak alarm is armed"
Serial xmit: Carriage return and line feed
```

You’ll probably also want to be able to speak variable values (such as the current temperature, thermostat setpoint, etc.). HomeVision has a command to transmit a variable value as ASCII text. Here’s an example:

```
Serial xmit: "speak the temperature is "
Serial xmit: Var #0 (Temp) value as 3 ASCII bytes
Serial xmit: " degrees"
Serial xmit: Carriage return and line feed
```

If the variable value is 78, this will transmit the following (the carriage return and line feed are not shown):

```
speak the temperature is 078 degrees
```

Note that when HomeVision transmits a variable value as ASCII, it always transmits three bytes. If the variable is less than 100, it sends a leading zero. For example, a variable value of 78 is transmitted as:

```
078
```

and a value of 5 is transmitted as:

```
005
```

Depending upon the PC program, this may or may not work right because of the leading zero(s). To overcome this problem, HomeVision version 2.6 (currently in Alpha testing) has a new command that will omit the leading zeros. If you need this capability prior to the version 2.6 release, let us know and we can send you an advance version.

Transmitting The Time

It's often useful to transmit the current time or the time of a prior event. HomeVision has a built-in command that transmits the current time and date together on one line. However, you'll probably want to use a different format. For example, you may want to transmit the time like this:

```
11:05 AM
```

Here's how to do this. First, use two variables to hold the desired time (you could use three variables if you need the seconds value). For example, to store the current time, perform these two commands:

```
Var #1 (Hour) = current hour
Var #2 (Minute) = current minute
```

The hour value will be in 24-hour format (i.e., 0 = 12:00 AM, 1 = 1:00 AM, …. 23 = 11:00 PM), so you'll need to convert it to AM/PM format. We recommend you do this with a macro so you can use it elsewhere in your schedule also. Here's a macro to do this:

```
Macro #1 (Convert time to AM/PM):

If
   Var #1 (Hour) = 0
Then
   Var #1 (Hour) = 12
   Set flag #1 (Time is AM)
Else
   If
      Var #1 (Hour) >= 1
      And Var #1 (Hour) <= 11
   Then
      Set flag #1 (Time is AM)
   Else
      If
         Var #1 (Hour) = 12
      Then
         Clear flag #1 (Time is AM)
      Else
         If
            Var #1 (Hour) >= 13
            And Var #1 (Hour) <= 23
         Then
            Var #1 (Hour) = var #1 (Hour) - 12
            Clear flag #1 (Time is AM)
         Else
            ; ERROR
         End If
      End If
   End If
   End If
```

This macro converts the value in variable #1 into AM/PM format. Initially, variable #1 ranges from 0 to 23. When the macro is done, the variable is between 1 and 12 (unless there's an error), and flag #1 indicates whether the result is AM or PM (Set = AM, Clear = PM).

Once the hour is in AM/PM format, use a second macro to transmit it out the serial port, like this:

```
   Macro #2 (Transmit time):

   Serial xmit: Var #1 (Hour) value as 3 ASCII bytes
   Serial xmit: ":"
   Serial xmit: Var #2 (Minute) value as 3 ASCII bytes
   If
      Flag #1 (Time is AM) is set
   Then
      Serial transmit: " AM"
   Else
      Serial transmit: " PM"
   End If
```

This macro transmits the hour and minute values contained in the two variables, separated by a colon. It then transmits either "AM" or "PM" based on the state of the flag. Note that if you are sending this to a text-to-speech program, you may need to replace the colon with a blank space. The colon works properly with Home Voice's text-to-speech.

To summarize, first set variables #1 and #2 to the desired time, then run macro #1 to convert it to AM/PM format, then run macro #2 to transmit it. Here's how you would transmit the current time:

```
  Var #1 (Hour) = current hour
  Var #2 (Minute) = current minute
  Do macro #1 (Convert time to AM/PM) once
  Do macro #2 (Transmit time) once
```

Storing An Event Time

It's often useful to store the time an event occurs. For example, you could determine the time that your mail arrives by placing a switch on your mailbox door. To store this time, first create variables to hold it (one for the hour, one for the minute, and, if desired, one for the second). When the event occurs, put the current time into the variables, like this:

```
  Var #3 (Mail hour) = current hour
  Var #4 (Mail minute) = current minute
```

Now let's assume that sometime later you need to transmit this time out the serial port. You would use these commands:

```
  Var #1 (Hour) = Var #3 (Mail hour)
  Var #2 (Minute) = Var #4 (Mail minute)
  Do macro #1 (Convert time to AM/PM) once
  Do macro #2 (Transmit time) once
```

This puts the mail arrival time into variables #1 and #2, then runs the macros described previously. This is quite easy since we're reusing macros written previously. The only trick is to remember that the macros use the values in variables #1 and #2, so we need to set the variables prior to running the macros.

Sunrise And Sunset Times

HomeVision calculates sunrise and sunset times each day for your latitude and longitude, and you can easily perform events based on these times. It could also be useful to transmit these times. For example, in the morning you could have your computer speak to you and tell you what time sunset is today. HomeVision doesn't have a command to transmit the sunset time, but here's a way you can do it. Create a scheduled event that runs at sunset and include these two commands in it:

```
Var #5 (Sunset hour) = current hour
Var #6 (Sunset minute) = current minute
```

Since the event runs only at sunset, the current time will be the sunset time. You simply store the current time in two variables. Later, you can transmit these variable values as described previously.

Transmitting Binary Values

HomeVision can transmit binary data in addition to ASCII text. Binary data is transmitted in a two step process, as shown in this example:

```
Var #7 (Binary #) = 13
Serial xmit: Var #7 (Binary #) value as 1 binary byte
```

This example sets the variable to 13, which is the binary value for a carriage return. It then transmits the variable value as a single binary byte. Since you can set variables to any value from 0 to 255, you can transmit any binary value. If you need to do this multiple times in your schedule, you can create a macro and then run it whenever you need to.

Note that HomeVision already has a command to transmit a carriage return and line feed. You'll usually want to do this at the end of each line of text you send. However, there may be times you want to transmit only a carriage return, as shown in the above example.

Accessing Other PC Serial Ports

HomeVision has the unique capability to allow the controller to access multiple serial ports through a PC. To do this, the HomeVision software must be running using the serial port the controller is connected to. The controller can then send commands to the PC to open another serial port, send data to it, and close it. While the port is open, any data coming into it is forwarded to the HomeVision controller.

The desired comm port is opened by sending ASCII text to the PC, like this:

```
Open comm port "2,19200,N,8,1,1"
```

In your schedule, you transmit it with this serial command:

```
Serial xmit: "Open comm port "2,19200,N,8,1,1""
```

Refer to the owner's manual for details on the command parameters. After opening the port, you can send data to it like this:

```
Serial xmit: "To comm port "Text to send""
```

When the PC software receives the text:

```
  To comm port "Text to send"
```

it takes the text between the double quotes and sends it to the open serial port. It also automatically puts a carriage return and line feed on the end (remember this, it's important!).

Now let’s assume you want to transmit a variable value to the serial port. For example, you may have a thermostat connected that you want to change to a temperature setpoint of 78 degrees. Let’s assume the thermostat must receive the command in this format:

```
  Setpoint=078<carriage return>
```

The easiest way to do this is with this command:

```
  Serial xmit: "To comm port "Setpoint=078""
```

Since HomeVision automatically puts a carriage return on the end, this command successfully changes the thermostat setpoint.

In this example, the 78 degree number is a fixed value, not a variable as we’d like. So how do we send a variable value? Well, here’s the first thing most users would try, but it won't work:

```
  Serial xmit: "To comm port "Setpoint=""
  Serial xmit: Var #0 (Temp) value as 3 ASCII bytes
  Serial xmit: Carriage return and line feed
```

The problem with this is that the HomeVision PC software puts a carriage return after the first command when sending it to the thermostat. Here’s what actually gets transmitted to the thermostat (assuming variable #0 is 78):

```
  Setpoint=<carriage return>078<carriage return>
```

To leave off the carriage return after the "Setpoint=" text, we use a trick. We leave off the ending double quote from the first command, send the variable value, then send a final double quote. Here are the commands to do that:

```
  Serial xmit: "To comm port "Setpoint="
  Serial xmit: Var #0 (Temp) value as 3 ASCII bytes
  Serial xmit: """
```

Here’s what these commands actually transmit to the PC:

```
  To comm port "Setpoint=
  078
  "
```

Note that there are no carriage returns on any line. As far as the PC is concerned, all the data comes in together, like this:

```
  To comm port "Setpoint=078"
```

This is exactly what we want. When the PC receives this, it takes the text between the double quotes and sends it to the thermostat.

You can use this same trick for any commands where the PC is looking for text between double quotes. These commands are:

- Open comm port "2,19200,N,8,1,1"
- To comm port "Text to send"
- Write to file filename.txt "Text to write"
- Send keys "Keystrokes"
- Activate window "Window name"
- Display message "Message to display"

Conclusion

HomeVision can transmit any combination of fixed text, variable values, and binary data out the serial port. Since variables can hold just about any information you want (temperature, time, date, etc.), there are very few limits to what you can do.

Tracking Temperatures with HomeVision

There are many things HomeVision can do with temperature data. Here are a few ideas.

Temperature Sources

Before you can do anything with temperature information, you have to get it into HomeVision. There are several ways to do this:

- Analog temperature sensors (like the widely used LM34) connected to the HomeVision multifunction expansion board. These sensors usually have 2 degree F resolution and cannot read temperatures below zero. They require a few resistors and capacitors connected at the sensor to achieve optimum performance.

- Digital temperature sensors (DS1820 from Dallas Semiconductor) connected to the expansion board. These sensors have 1 degree F resolution and can read from -50 to +205 degrees F. Unlike analog sensors, they do not require any additional components. Also, you can attach multiple sensors to the same three wires, which greatly simplifies wiring. This is possible because each device has a unique address and HomeVision can read them individually.

- Thermostats. HomeVision can read the actual temperature back from the RCS TX10-B thermostat using X-10 signals. Other thermostats can be connected to HomeVision's serial port, allowing HomeVision to read their temperature as well.

- A computer. If you can get temperature data into your computer (from a weather station or other device), you can transmit it to HomeVision.

What can you do with this information? Here are a few things.

Track High and Low Temperatures

It's often useful to track daily high and low temperatures. To do this, create a variable for each. Then, update these each time you read in a new temperature. For example, the code below can be used in a periodic event that reads an analog temperature sensor every minute.

```
Var #1 (Current temp) = Analog input #1 (LM34)
If
   Var #1 (Current temp) > var #2 (Daily high temp)
Then
   Var #2 (Daily high temp) = var #1 (Current temp)
End If
If
   Var #1 (Current temp) < var #3 (Daily low temp)
Then
   Var #3 (Daily low temp) = var #1 (Current temp)
End If
```

You must also reset the high and low temperature variables at the end of each day. This is best done in a scheduled event that performs the following commands at midnight:

```
Var  #2 (Daily high temp) = var #1 (Current temp)
Var  #3 (Daily low temp) = var #1 (Current temp)
```

Weather Conditions TV Screen

The weather conditions TV screen can display a variety of weather data, including temperatures. An example screen is shown below:

```
   WEATHER CONDITIONS
             NOW  LO  HI
Inside Temp   72  70  74
Outside Temp  77  68  82
Inside RH     65  60  70
Outside RH    85  79 100
Wind Speed    16
Wind Direct  340 NNW
Rain Today  0.58 Inches
Pressure   30.05
PRESS ANY KEY TO RETURN
```

You would use the following three weather data commands to put the inside temperature readings into this screen:

```
Current inside temp = var #1 (Current temp)
Today's high inside temp = var #2 (Daily high temp)
Today's low inside temp = var #3 (Daily low temp)
```

You can do the same thing with the outside temperature readings as well. Note that if you don’t have access to the non-temperature data, you can leave them blank on the screen.

Display Temperature on the TV

Another use is to display the current temperature overlayed onto the TV channel you’re watching. You can create a periodic event that runs every hour and put the following commands in it:

```
Video: Start external video mode
Video: Display video screen # 0
Video: Set cursor row to 11
Video: Set cursor column to 1
Video: Display last 2 digits of var #1 (Current temp)
Wait 0:00:30.00 with timer #1 (Display temp), then:
   Video: Stop video mode
End Wait
```

These commands start the external video mode and display a blank screen (# 0) for you to write on. They put the cursor in the lower left screen corner, then write the current temperature. Finally, they start a timer which will shut the screen off after 30 seconds. Thus, the periodic event runs every hour, each time displaying the temperature overlayed onto the video source for 30 seconds.

Automatic HVAC Heat/Cool Changeover

HomeVision can control multiple thermostats to different daily time/temperature schedules, and set them back when you leave home. One useful thing you may want to also do is automatically switch your HVAC system into the proper mode based upon the actual temperature. The example below shows one way to do this.

```
If
   HVAC zone 1 current temp is > 78
Then
   ; Ensure we're in the cool mode:
   If
      HVAC zone 1 system mode is off
      Or HVAC zone 1 system mode is heat
      Or HVAC zone 1 system mode is auto
   Then
      Set HVAC zone 1 system mode to cool
      ; Ensure temperature setpoint is correct:
      If
         Flag #1 (At Home) is set
      Then
         Set HVAC zone 1 to scheduled temp
      Else
         Set HVAC zone 1 temp setpoint back
      End If
   End If
End If
;
If
   HVAC zone 1 current temp is < 70
Then
   ; Ensure we're in the heat mode:
   If
      HVAC zone 1 system mode is off
      Or HVAC zone 1 system mode is cool
      Or HVAC zone 1 system mode is auto
   Then
      Set HVAC zone 1 system mode to heat
      ; Ensure temperature setpoint is correct:
      If
         Flag #1 (At Home) is set
      Then
         Set HVAC zone 1 to scheduled temp
      Else
         Set HVAC zone 1 temp setpoint back
      End If
   End If
End If
```

It’s best to put these commands in a periodic event that runs each minute. If the temperature is above 78, it makes sure the HVAC is in the cool mode. If not, it puts it in the cool mode, then makes sure the temperature setpoint is correct. If we’re at home, it’s set to the temperature specified by the current schedule. If we’re away from home, the setback temperature (a higher temperature to save energy) is used. The heat mode is similar.

Note that the setback temperatures and the temperature schedule are specified elsewhere; this code just shows how the changeover is made.

Overtemperature and Undertemperature Monitoring

You can easily check to see if the temperature reaches an extreme that requires you to take special action. For example, you could monitor the indoor temperature while away from home to see if it drops below freezing. Or you could put sensors in your refrigerator and freezer to detect failures.

You might sound an alert if any of the temperatures exceed preset limits. This works well if someone is at home to see or hear the alert, but what if no one's home? How can you report these conditions? Here are a few ways:

- Connect HomeVision to a modem and have HomeVision dial a pager. It can also enter a unique code for different alarm conditions so you can determine the cause.
- Connect a HomeVision relay output to an input on your security system. Program the security system to dial out and play a message whenever the input is triggered. Note that not all security systems have this capability.
- Connect a HomeVision relay output to an auto-dialer.
- Connect HomeVision to a computer and have the computer:
  - Dial a pager (using the computer's modem).
  - Send a FAX (as described in the previous issue of Home Automator).
  - Send an e-mail message.

Report Data to Computer

HomeVision can transmit the temperature data to a computer and write it to a text file. Create a periodic event that runs every 5 minutes and transmits the temperature and current time. If you write it in the proper format, you can then import the file into an Excel spreadsheet. Since HomeVision can control programs on the computer, you could even automate this. Every day at 6:00 AM HomeVision could import the latest file into Excel, graph it, then print it for you to review later!

Conclusion

This only scratches the surface of how you might use temperature data in home automation. Let us know if you come up with other interesting ideas.

Receiving Infrared Sequences

Introduction

HomeVision can be controlled from a standard infrared (IR) remote. Any remote button can trigger a complex sequence of actions with just a single key press. Some users, however, like to use a *sequence* of consecutive key presses to trigger an action. There are several advantages to this:

1) A single remote can provide many more options. For example, using single key presses, the 10 number keys can only trigger 10 different actions. A sequence of 3 key presses, in contrast, can trigger 1000 different actions.

2) Inadvertent actions can be reduced. An action that triggers on a single key press could be caused by accidentally pressing one button. Using a sequence, however, makes it much less likely that someone will accidentally trigger it.

3) The user can protect certain actions with a password. For example, the user may require a 5-digit sequence to arm or disarm the alarm. This provides a high degree of security.

While HomeVision can easily receive IR signals and use them to trigger actions, it cannot automatically receive "infrared sequences". Instead, you will need to set up a few variables and events to implement this function. This article explains how to do this and provides an example. You may use this "as is" with HomeVision, or modify it to add even more capabilities.

Example Overview

The example included here can receive a sequence of either two or three key presses to trigger actions. Each key must be pressed within two seconds of the previous key. If not, the sequence is canceled. It uses one If-Then statement to check for each user-defined sequence. Note that a 2-button sequence cannot be a subset of a 3-button sequence (for example, if you have a 2-button sequence of "3-5", you can't have a 3-button sequence of "3-5-7").

Schedule Setup

Set up your schedule as follows:

1) Use the Variable Summary Screen to add the following five variables:

   Variable 1 = New IR Button
   Variable 2 = First Received IR Button
   Variable 3 = Second Received IR Button
   Variable 4 = Third Received IR Button
   Variable 5 = Number of IR Keypresses

   Set the "Initial State" and "Power Failure State" of each variable to zero.

2) Use the Timer Summary Screen to add a timer named "IR Sequence Timer".

3) Use the Infrared Signal Summary Screen to add the IR signals you want to receive. You will probably want 10 signals, one for each of the number keys. Define them to be "standard signals" and set their device and key codes to match your remote control. Refer to the infrared chapter of the owner's manual for assistance.

4) For each of the IR signals, enter the actions shown in the example.

5) Use the Macro Summary Screen to add a macro named “IR Button Pressed”. Enter the actions shown in the example.

6) Download the schedule into the controller and test it.

How It Works

When an IR key is pressed, the corresponding IR event runs. This set the “New IR Button” variable equal to the value of the key pressed, then runs the “IR Button Pressed” macro.

The macro first increments the “Number of IR Keypresses” variable, which is initially zero. This variable keeps a running count of how many keys have been pressed in this sequence. Next, the macro stores the number of the button just pressed into a variable. If this is the first button in the sequence, the value is stored in the “First Received IR Button” variable; if it’s the second button, it’s stored in the “Second Received IR Button” variable, and so forth.

If this is the second key press, the macro then checks to see if the received sequence matches a user-defined 2-button sequence. This is done with If-Then statements that checks the “First Received IR Button” and “Second Received IR Button” variables against the user-defined values (8 and 9 in this example). If they match, the desired actions are taken (in this example, it simply sends a message out the serial port). At this point, the sequence is complete, so the “Number of IR Keypresses” variable is reset to zero (so that a subsequent key press starts a new sequence). If there is no match to a 2-button sequence, the variable is not reset, so it can receive a third key press.

If this is the third key press, the macro checks to see if the received sequence matches a user-defined 3-button sequence. If so, the desired actions are performed. However, regardless of whether the 3-button sequence matches or not, the sequence is terminated by resetting the “Number of IR Keypresses” to zero. This ensures that invalid 3-button entries are cleared and that the system is ready to receive the next sequence.

Finally, the macro starts the “IR Sequence” timer at two seconds. This will time out and abort the IR sequence if the user doesn’t press another key within two seconds. You can adjust this value to work best for you. Note that this timer is started even if the sequence is complete, which isn’t really necessary, but it’s the easiest way to do it.

Conclusion

The IR sequence capability can be extremely powerful. You can add as many sequences as you like, or extend the sequence beyond three key presses. You’re limited only by your imagination!

EXAMPLE SCHEDULE EVENTS

IR SIGNAL EVENT #0 'IR Remote Button 0'

Var #1 (New IR Button) = 0
Do macro #1 (IR Button Pressed) once

```
IR SIGNAL EVENT #1 'IR Remote Button 1'

Var #1 (New IR Button) = 1
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #2 'IR Remote Button 2'

Var #1 (New IR Button) = 2
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #3 'IR Remote Button 3'

Var #1 (New IR Button) = 3
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #4 'IR Remote Button 4'

Var #1 (New IR Button) = 4
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #5 'IR Remote Button 5'

Var #1 (New IR Button) = 5
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #6 'IR Remote Button 6'

Var #1 (New IR Button) = 6
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #7 'IR Remote Button 7'

Var #1 (New IR Button) = 7
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #8 'IR Remote Button 8'

Var #1 (New IR Button) = 8
Do macro #1 (IR Button Pressed) once

IR SIGNAL EVENT #9 'IR Remote Button 9'

Var #1 (New IR Button) = 9
Do macro #1 (IR Button Pressed) once


MACRO EVENT #1 'IR Button Pressed'

;First, increment the keypress counter:
Increment var #5 (Number of IR Keypresses)

;Now determine how many keys have been pressed in this sequence:
If
    Var #5 (Number of IR Keypresses) = 1
```

```
Then
    ;This is the first IR keypress in the sequence
    Var #2 (First Received IR Button) = var #1 (New IR Button)
Else
    If
        Var #5 (Number of IR Keypresses) = 2
    Then
        ;This is the second IR keypress in the sequence
        Var #3 (Second Received IR Button) = var #1 (New IR Button)

        ;Look for any 2-button sequences we’re interested in:
        If
            Var #2 (First Received IR Button) = 8
            And Var #3 (Second Received IR Button) = 9
        Then
            Serial transmit: "IR SEQUENCE 89 RECEIVED!"
            ;We're done with the sequence, so reset the keypress counter:
            Var #5 (Number of IR Keypresses) = 0
        End If
    Else
        If
            Var #5 (Number of IR Keypresses) = 3
        Then
            ;This is the third IR keypress in the sequence
            Var #4 (Third Received IR Button) = var #1 (New IR Button)

            ;Look for any 3-button sequences we’re interested in:
            If
                Var #2 (First Received IR Button) = 1
                And Var #3 (Second Received IR Button) = 2
                And Var #4 (Third Received IR Button) = 2
            Then
                Serial transmit: "IR SEQUENCE 122 RECEIVED!"
            End If

            If
                Var #2 (First Received IR Button) = 1
                And Var #3 (Second Received IR Button) = 2
                And Var #4 (Third Received IR Button) = 3
            Then
                Serial transmit: "IR SEQUENCE 123 RECEIVED!"
            End If

            ;We're done with the sequence, so reset the keypress counter:
            Var #5 (Number of IR Keypresses) = 0
        End If
    End If
End If

;Start timer which will timeout if no keys are pressed in 2 seconds:
Wait 0:00:02.00 with timer #1 (IR Sequence Timer), Then:
    ;User hasn't pressed a key, so abort this IR sequence:
    Var #5 (Number of IR Keypresses) = 0
End Wait
```