

Debugging Tips

Here are a few tips that may help you when debugging your schedule.

Tip 1 - Serial Messages

Use serial commands in your schedule to track what the controller is doing. On an attached PC, watch the messages in the terminal emulator window or log them to a file for later viewing. Put the messages anywhere you're not sure what the controller is actually doing. For example, if an event looks like this:

```
If
  Var #1 (Motion sensor count) > 5
Then
  X-10: A-1 (Porch light) ON
Else
  X-10: A-1 (Porch light) OFF
End If
```

Change it to this:

```
If
  Var #1 (Motion sensor count) > 5
Then
  X-10: A-1 (Porch light) ON
  Serial Transmit: "Condition is true. Variable #1 value ="
Else
  X-10: A-1 (Porch light) OFF
  Serial Transmit: "Condition is false. Variable #1 value ="
End If
Serial Transmit: Variable #1 (Motion sensor count) value as 3 ASCII bytes
Serial Transmit: Carriage return and line feed
Serial Transmit: Transmit time and date
```

If an error is occurring, we'd like to know when and how often it happens. One way to do this is to have a periodic event running every loop. Put the following code (or something similar) in it.

```
If
  Controller error
Then
  ; Report error number over serial interface:
  Controller command: Put error message in Result Value
  Var #1 (Error number) = Result Value
  Serial port 1 transmit: "ERROR: "
  Serial port 1 transmit: Variable #1 (Error number) value as 3 ASCII bytes
  Serial port 1 transmit: Carriage return and line feed
  Serial port 1 transmit: Time and date
  ; Now clear the error:
  Controller command: Clear error message
End If
```

This code will report every error over the serial interface, so you'll need to leave the HomeVision Terminal Emulator Screen open or enable logging of serial data (by clicking the Logging button at the bottom of the main HomeVision software screen). Note that the IF-THEN condition and the controller commands are located under the "Controller" button on the Actions Entry Screen Toolbar.

If you want to get fancier, you could even count the number of errors, as shown below:

```
If
  Controller error
Then
  ; Increment error counter:
  Increment 16-Bits: var #3 (Error count MSB) MSB & var #2 (Error count LSB) LSB
  ; Report error number over serial interface:
  Controller command: Put error message in Result Value
  Var #1 (Error number) = Result Value
  Serial port 1 transmit: "ERROR: "
  Serial port 1 transmit: Variable #1 (Error number) value as 3 ASCII bytes
  Serial port 1 transmit: Carriage return and line feed
  Serial port 1 transmit: Time and date
  ; Now clear the error:
  Controller command: Clear error message
End If
```

Tip 2 - Video Event Log

Use the video event log to determine when events are happening. Many events can be automatically written to the log, and you can also write your own messages. If you've got a lot of X-10 signals filling up the log, disable their automatic logging. Then add your own commands to write to the log in the specific X-10 signal events you're interested with.

Tip 3 - Controller LED

Use the controller commands that turn the user LED on and off. If you need to know when an event happens, have it turn the LED on. This is also convenient for determining when a timer is running. Turn the LED on when you start the timer, and turn it off when the timer expires. The LED commands are especially useful if you cannot leave the controller connected to a PC to receive serial messages.

Tip 4 - Debug Mode Flag

Often, you'll put some commands in your schedule while testing it, but don't want them executing all the time. A simple solution is to put all your debugging commands in an If-Then statement. Create a flag named "Debug Mode", and use the condition "If flag (Debug Mode) is set" in the If-Then statement. Whenever you need to debug the schedule, set the flag. When you're done, clear it and none of the debugging commands will execute.

Tip 5 - X-10 Transmissions

X-10 signal loss, phase coupling, or noise often cause communications problems. To help test for this, you may want the controller to repeatedly transmit X-10 signals so you can check various outlets with a signal meter or light. Here's one way to do it.

First, create a periodic event that is initially DISABLED and set it to run "each loop". In the actions for the event, put a command to toggle X-10 address P-1 (or any other address). Then create a macro called "Start Test Transmissions" and put a command in it to enable the periodic event. Next, create a macro called "Stop Test Transmissions" and put a command in it to disable the periodic event.

Whenever you want to start the test, run the "Start Test Transmissions" macro. You could do this from the serial interface, with an infrared command, or through an input port. The periodic event will then alternate between sending P-1 ON and P-1 OFF commands. Set up another event to run the "Stop Test Transmissions" so you can shut it off. Note that you can't use an X-10 signal to trigger the "Stop Test Transmissions" macro because the power line will be clogged up with the controller's transmissions.

Note that you don't really have to create macros as described here. Instead, you could directly enable and disable the periodic event (you could even do this from the video screen). You could also have the transmissions automatically stop after a certain amount of time by using a Wait Timer.