HomeVision Hints

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