Local Variable VI

Goal

Use a local variable to write to and read from a control.

Scenario

You have a LabVIEW Project that implements a temperature weather station. The weather station acquires a temperature every half a second, analyzes each temperature to determine if the temperature is too high or too low, then alerts the user if there is a danger of a heat stroke or freeze. The VI logs the data if a warning occurs.

Two front panel controls determine the setpoints-the temperature upper limit and the temperature lower limit. However, nothing prevents the user from setting a lower limit that is higher than the upper limit.

Use a local variable to set the lower limit equal to the upper limit if the user sets a lower limit that is higher than the upper limit.

Design

The VIs in this project have already been written. Your only task is to modify the VIs so that the lower limit is set equal to the upper limit when necessary.

State Definitions

The following table describes the states in the state machine.

State	Description	Next State
Acquisition	Set time to zero, acquire data from the temperature sensor, and read front panel controls	Analysis
Analysis	Determine warning level	Data Log if a warning occurs, Time Check if no Warning occurs
Data Log	Log the data in a tab-delimited ASCII file	Time Check
Time Check	Check whether time is greater than or equal to .5 seconds	Acquisition if time has elapsed, Time Check if time has not elapsed

Changing the value of the lower temperature limit control should happen after the user has entered the value but before the value determines the warning level. Therefore, make the modifications to the VI in the Acquisition or Analysis state, or place a new state between the two.

The files that you need to complete this exercise are here: `<NI eLearning>\LV Core 1\Variables\Exercise`.

1. Before determining which option to use, take a closer look at the content of the Acquisition and Analysis states:

   - ❑ Open `Weather Station.lvproj` in the `<Exercise>` directory.
   - ❑ Open `Weather Station UI.vi`.
   - ❑ Review the contents of the Acquisition and Analysis states, which correspond to the Acquisition and Analysis cases of the Case structure.

Design Options

You have three different design options for modifying this project.

Option	Description	Benefits/Drawbacks
1	Insert a Case structure in the Acquisition state to reset the controls before a local variable writes the values to the cluster.	Poor design: the acquisition state has another task added, rather than focusing only on acquisition.
2	Insert a new state in the state machine that checks the controls and resets them if necessary.	Ability to control when the state occurs.
3	Modify the Determine Warnings subVI to reset the controls.	Easy to implement because functionality is already partially in place. However, if current functionality is used, one set of data always is lost when resetting the lower limit control.

This exercise implements Option 2 as a solution.

New State Definitions for Option 2

The following table describes the new state definitions to implement.

State	Description	Next State
Acquisition	Acquire data from the temperature sensor on channel AI0 and read front panel controls	Range Check
Range Check	Read front panel controls and set the lower limit equal to the upper limit if upper limit less than the lower limit	Analysis
Analysis	Determine warning level	Data Log if a warning occurs, Time Check if no Warning occurs
Data Log	Log the data in a tab-delimited ASCII file	Time Check
Time Check	Check whether time is greater than or equal to .5 seconds	Acquisition if time has elapsed, Time Check if time has not elapsed

Implementation

1. If the `Weather Station.lvproj` is not already open, open it from the `<Exercise>` directory.

2. Add the Range Check state to the state machine.

   - From the **Project Explorer** window, open the `Weather Station States.ctl` by double-clicking the listing. This is the type-defined enumerated control that defines the states for the state machine.

   - Right-click the control and select **Edit Items** from the shortcut menu.

   - Insert an item and modify to match Table 1. Be careful not to add an empty listing.

Table 1. States Enumerated Control

Item	Digital Display
Acquisition	0
Range Check	1
Analysis	2
Data Log	3
Time Check	4

- ❑ Save and close the control.
- ❑ If the `Weather Station UI.vi` is not open, open it by double-clicking the listing in the **Project Explorer** window.
- ❑ Open the block diagram.
- ❑ Right-click the state machine Case structure and select **Add Case for Every Value** from the shortcut menu. Because the enumerated control has a new value, a new case appears in the Case structure.

3. Read the upper and lower limit controls in the Range Check state, instead of the Acquisition state, as shown in Figure 1.

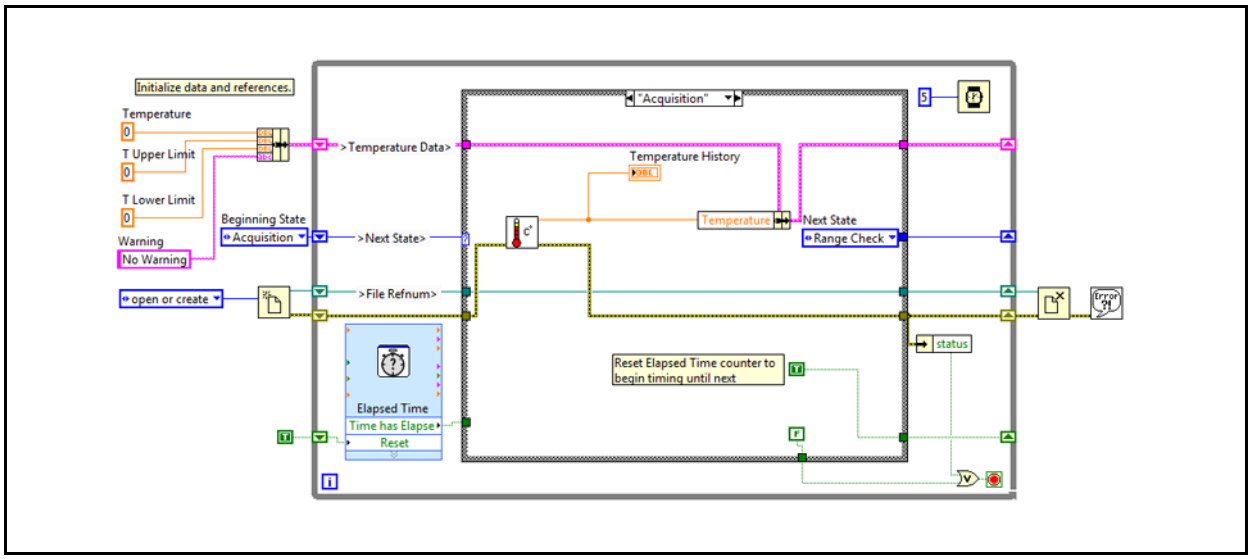

Figure 1. Completed Acquisition State

- ❑ On the block diagram of the Weather Station UI VI, select the **Acquisition** case in the state machine Case structure.

- ❑ Inside the Acquisition case, change the **Next State** enum to **Range Check**.
- ❑ Make a copy of the **Next State** enum by pressing <Ctrl> and dragging a copy outside the While Loop.
- ❑ Move the **Upper Limit** and **Lower Limit** numeric controls outside the While Loop.
- ❑ Resize the **Bundle by Name** function to one element, as shown in Figure 1.
- ❑ Select the **Range Check** case in the state machine Case structure.
- ❑ Move the **Upper Limit** and **Lower Limit** numeric controls and the **Next State** enum into the Range Check state.

4. Set the Range Check state to transition to the Analysis state.
   - ❑ In the Range Check case, wire the **Next State** enum to the **Next State** output tunnel.
   - ❑ Change the **Next State** enum to **Analysis**.
5. If the Upper Limit is less than the Lower Limit, use a local variable to write the Upper Limit value to the Lower Limit control, as shown in Figure 2.

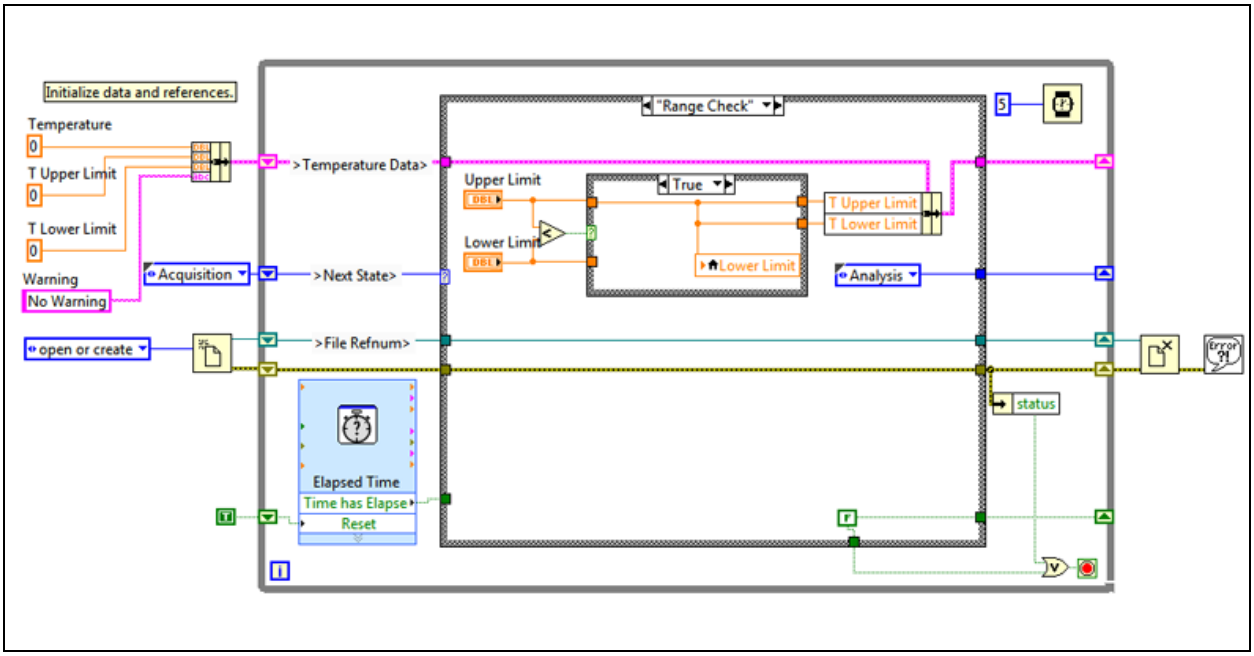

Figure 2. Completed Range Check State-True

- ❑ Add a **Less?** function to the Range Check state.

- ❑ Add a **Case structure** to the right of the Less? function.
- ❑ Wire the **Upper Limit** and **Lower Limit** controls to the **Less?** function and the **Case structure** as shown in Figure 2.
- ❑ Right-click the **Lower Limit** control and select **Create»Local Variable** from the shortcut menu.
- ❑ Move the local variable inside the True case of the Case structure.

- ❑ Add a **Bundle By Name** function to the right of the Case structure.
- ❑ Wire the **Temperature Data** cluster to the **input cluster** input of the Bundle By Name function.
- ❑ Expand the **Bundle By Name** function to two elements.
- ❑ Select **T Upper Limit** in the first element and **T Lower Limit** in the second element.

- ❑ Add a `False` constant to the outer Case structure.
- ❑ Wire the case as shown in Figure 2.

6. If the Upper Limit is equal to or greater than the Lower Limit, pass the values of the controls to the temperature cluster, as shown in Figure 3.

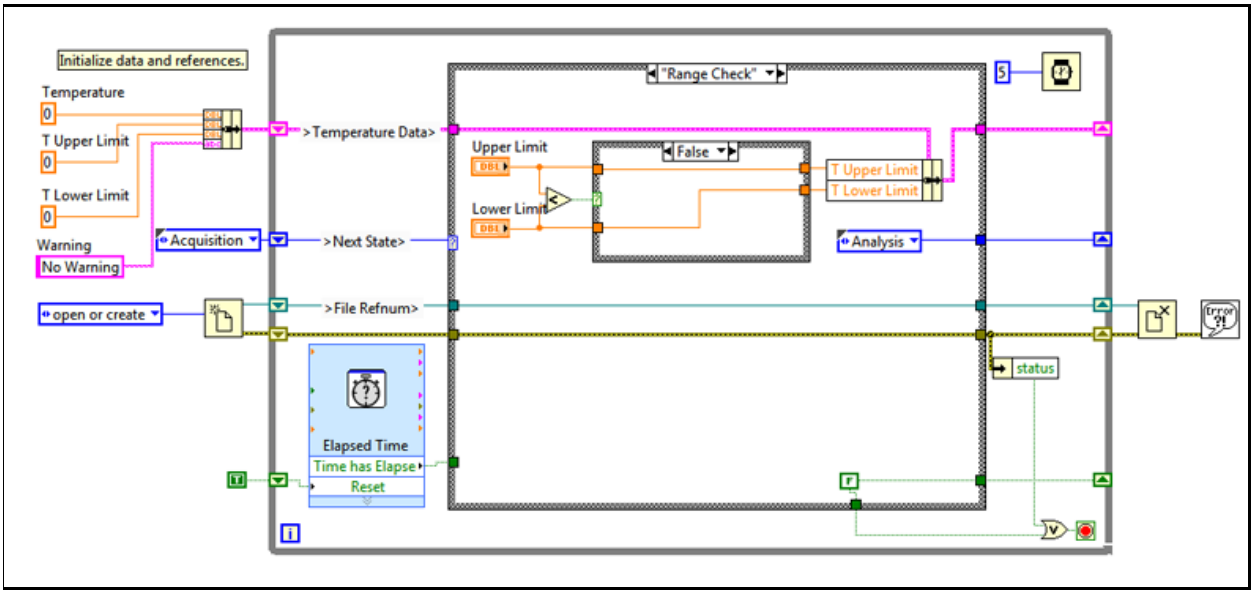

Figure 3. Figure 3. Completed Range Check State-False

- ❑ Switch to the **False** case of the interior Case structure.
- ❑ Wire the **Upper Limit** and **Lower Limit** data through the case.

7. Save the VI.
8. Save the Project.

Test

1. Run the VI.
   - ❑ Name the log file when prompted.
   - ❑ Enter a value in the **Upper Limit** control that is less than the value in the **Lower Limit** control. Does the VI behave as expected?
2. Stop the VI when you are finished.
3. Close the VI and the project.

End of Exercise

Notes