CLD Exercise 2: Elapsed Time Express VI FGV

Objective

Develop a simple Functional Global Variable (FGV) Timer based on the express VI and the given application front panel. Use the provided Enum with three states: Reset, Set Auto Reset, and Read Status.

The FGV is not an application in itself. Therefore, use the Test VI application front panel (Figure 1) to test the FGV.

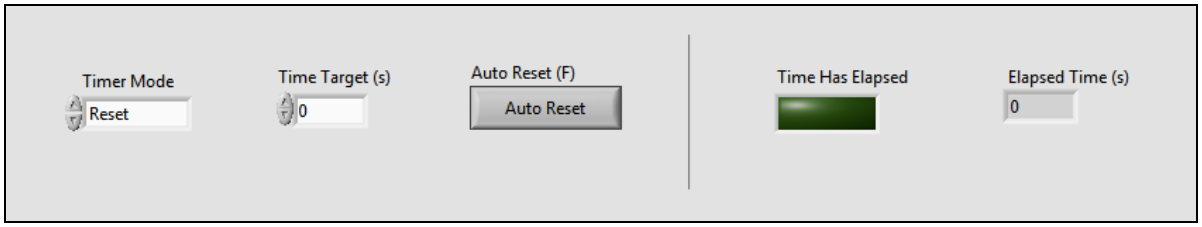

Figure 1. Test VI Application Front Panel

General Operation

The Test VI must continuously loop and call the FGV timer. The FGV timer must count up from zero to the **Time Target** while displaying the elapsed seconds in the **Elapsed Time** indicator. When the elapsed time has expired, the **Time Has Elapsed** LED must turn ON. The reset and auto reset modes are used to control the timer.

Application Terminology

FGV Timer

The functional global variable VI uses the Elapsed Time Express VI and the provided **Timer Mode** control.

Timer Mode

The Enum used for the FGV. This Enum has three values.

- **Reset:** Used to reset the express VI
- **Set Auto Reset:** Used to change the value of the auto reset shift register.
- **Read Status:** Used to read the outputs of the express VI

Test VI

This VI uses the FGV timer as a subVI in a loop.

Time Target

The time, in seconds, used for the timer application.

Auto Reset

The default value for the **Auto Reset** button is ON. When the **Auto Reset** button is ON, the timer must immediately begin a new timing cycle upon expiration of elapsed time.

When the **Auto Reset** button is OFF, the timer must continue to count elapsed time and keep the **Time Has Elapsed** indicator ON.

Elapsed Time

This indicator must continuously display the elapsed time in seconds.

Time Has Elapsed

This indicator turns ON when the time has expired. It is OFF whenever the time has not yet elapsed**.**

Initialization

The Test VI must initialize as shown in Figure 1, and the front panel controls and indicators must be in the following states.

- **Time Target:** Set to 4 seconds
- **Auto Reset:** Set to ON
- **Time Has Elapsed:** Set to OFF
- **Timer Mode:** Set to Reset

Operation

VI Run

The Test VI starts in Reset mode.

Set Timer Mode to Reset Status

With the **Timer Mode** set to Reset, the following actions must occur in the FGV timer:

- The Test VI time target is saved to the FGV timer
- The timer is reset.
- Auto reset is set to FALSE

Set Timer Mode to Read Status

With the Test VI **Timer Mode** set to Read Status, the Test VI must continuously read the status of the FGV timer. The following actions occur as the Test VI is reading.

When the Time Target is reached, the **Time Has Elapsed** LED must turn on.

If the **Auto Reset** is ON:

- The **Time Has Elapsed** LED turns ON
- The timer begins a new timing cycle
- The **Time Has Elapsed** LED turns OFF

If the **Auto Reset** is OFF:

- The **Time Has Elapsed** LED turns ON
- The timer continues counting up

Set Timer Mode to Set Auto Reset

With the Test VI in Set Auto Reset mode, the Test VI must pass the Auto Reset value to the FGV timer. The Test VI must read the status of the FGV timer.

When the **Auto Reset** shift register is changed the FGV timer must immediately respond.

- If the **Time Has Elapsed** is OFF the application must continue operation regardless if the **Auto Reset** is on or off.
- If the **Auto Reset** is changed to ON while the **Time Has Elapsed** is ON, the timer begin a new timing cycle.
- If the **Auto Reset** is changed to OFF while the **Time Has Elapsed** is ON, the timer must continue operation with the time elapsed.

Questions

How does the reentrancy of the express VI affect the testing of the action engine?

How does using a FGV simplify timing in an application?

Challenge Exercise

Develop a simple timer application that can pause timing, using the Elapsed Time Express VI and the given Application front panel (Figure 2).

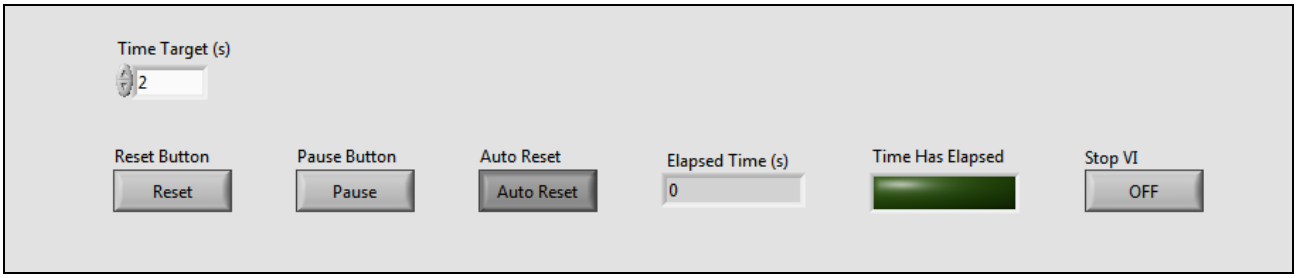

Figure 2. Pause Functionality Application Front Panel

General Operation

This application must operate the same as CLD Exercise 1, except when the **Pause** button is pushed and the timer must pause and display the elapsed time at the time of the pause. Upon releasing the **Pause**, the timer must continue from the point of the previous elapsed time.

Questions

The addition of the pause time requires the addition of what type of memory storage?

Is there more than one possible method to store data for use with the pause functionality?