CLD Exercise 4: Event Structure Time Out

Objective

Develop a simple state machine, with an event structure as the timer, using the given application front panel (Figure 1). The application must maintain a count based on the **Increment** and **Decrement** buttons being pressed, and turns the **Time Has Elapsed** LED ON if there is inactivity for a time period greater than the **Time Target**.

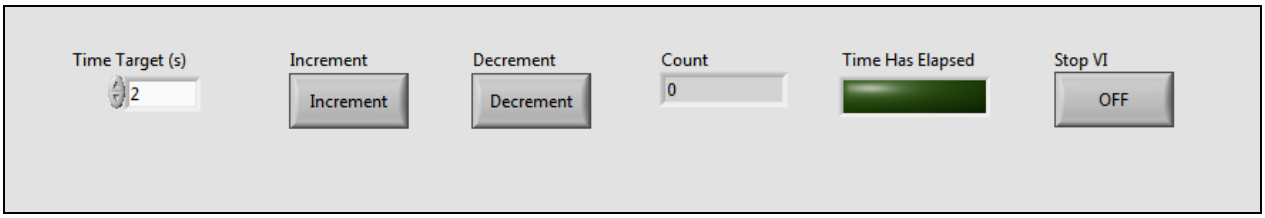

Figure 1. Application Front Panel

General Operation

The application maintains a count that starts at zero and increments or decrements when the **Increment** or **Decrement** buttons are pressed. When there is a period of inactivity on the front panel that is longer than **Time Target**, the **Time Has Elapsed** indicator must turn on and the **Count** resets to zero.

Application Terminology

Count

A count based on the **Increment** and **Decrement** buttons. The **Count** is an unsigned long integer and when decrementing below zero its value "rolls" to the highest integer value and counts down.

Increment and Decrement

The controls must increase or decrease the count number when pressed.

Time Target

The time, in seconds, used for the timer application. If this time elapses without any UI activity the count resets to zero.

Time Has Elapsed LED

This LED must turn ON when no UI activity has occurred for a length of time equal to the time target. The indicator remains ON until a UI activity occurs.

Initialization

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

- **Time Target:** Set to 2 seconds
- **Count:** Set to zero
- **Time Has Elapsed:** Set to OFF

Operation

VI Run

When started, the VI begins timing. If no actions are taken for a period of the **Time Target** the VI must set the **Count** to zero and turn ON the **Time Has Elapsed** LED.

Press Increment or Decrement buttons

The VI maintains a running count based on pressing the **Increment** and **Decrement** buttons.

Set Time Target

When the **Time Target** is changed, the timer must immediately respond to the new value of the **Time Target**.

Questions

Is a timeout case of an event structure a good timer for measuring elapsed time?

What are the strengths and weaknesses of the time out constant of an event structure time out case?

Challenge Exercise

Develop a timer using the time out functionality of an event structure. The application also acts as a bit counter and displays changing indicators corresponding to four bits. As there are four bits, all time steps must be in increments of one sixteenth of the **Time Target**.

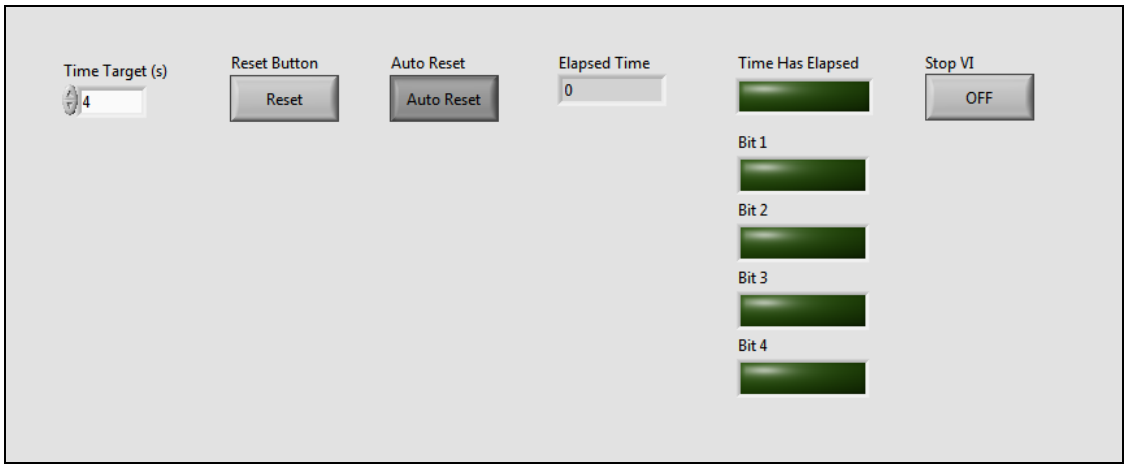

Figure 2. Event Structure Timer and Bit Counter

General Operation

The application measures elapsed time and displays the amount of elapsed time in milliseconds. The application uses the timeout value of the event case to measure time. The time target must be divided into 16 steps with the four indicator bits counting each step in binary.

Application Terminology

Time Target

The time in seconds used for the timer application. If this value is changed, the new **Time Target** must immediately become effective.

Time has Elapsed

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

Bits 1 - 4

These indicators are the four bits that must turn on and off. **Bit 1** is the least significant; **Bit 4** is the most significant digit. The **Time Target** must be divided by 16 in order for the 16 binary states to fit the time target.

For example, if the **Time Targe**t is 4 seconds, then each quarter second the indicators must change. In Figure 3 the time elapsed is 2.5 seconds, the 10^{th} step in 16 steps of a quarter second each, as displayed by the binary number 0101.

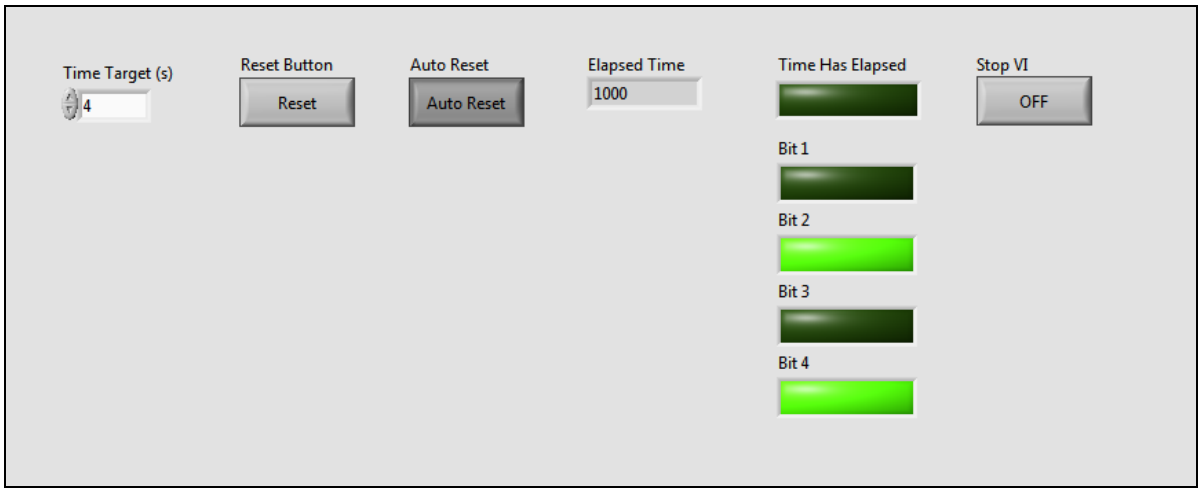

Figure 3. Bit Counter Example.

Reset

When the **Reset** button is pressed, the timer must start timing at zero and reset all the indicators to FALSE, and begin a new timing cycle.

Auto Reset

If this is TRUE, the timer must start again at zero when time elapses. If it is set to FALSE, the application must stop when the time elapses.

Initialization

The application must initialize as shown in Figure 2, and the front panel controls and indicators must be in the following states.

- **Time Target:** Set to 4 seconds
- **Auto Reset:** Set to ON
- **Elapsed Time:** Set to zero
- **Time Has Elapsed:** Set to OFF
- **Bits 1 – 4:** Set to OFF

Operation

VI Run

The VI must start timing, the **Elapsed Time** counts up and the **Time Has Elapsed** LED must be OFF. The **Bit** LEDs must start turning ON and OFF to simulate a bit counter.

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 must reset to zero and begin counting up
- The **Time Has Elapsed** LED turns OFF

If the **Auto Reset** is OFF:

- The application must stop

Set Time Target

When the **Time Target** is changed the timer must immediately respond to the new value of the **Time Target**.

Set Auto Reset

When the **Auto Reset** button is pressed the application must immediately respond.

- If the **Time Has Elapsed** is OFF the application must continue operation regardless if the **Auto Reset** is on or off.

Stop VI

The application must stop within one time step reaction time.

Questions

What is the greatest drawback to using this type of timing method?