# Tips for Event Driven LabView Virtual Instruments

*George Suárez*

Department of Computer and Electrical Engineering

University of Maryland, Baltimore County

CMPE 650 Digital Systems Design

george.suarez@umbc.edu

February 25, 2008

## Some Tips for Event Driven Virtual Instruments

Event driven in LabView is very similar to the coding style in embedded system. One simple way of doing this is by using three simple structures:a *while loop* controlled by a *boolean (Stop) control*, a *Case* controlled by an *Enum constant* with predefined values, and an *Event structure* that reacts on the programmed events. The continuous *While loop* ensures that the VI always runs until the stop signal is detected inside the *Event structure*. The *Case structure* normally contains three cases: Initialize, Run, and Quit (see Figures 1-**??**). By using the *Enum constant* with these three cases it is possible to control the flow of events on the VI. The initialization contains commands/assignments that are normally needed only at the beginning of execution. Inside the initialization the *Case structure* is forced to go to the Run case/state. The value of this signal representing the state of the VI is held by a shift register inside the *While loop* (small arrows on *While loop* left and right borders). Moreover, the *Event structure* lies inside the Run case. It is here where all the *event driven actions* are programmed similarly to interrupts in micro controllers. More information is available at the National Instruments website.
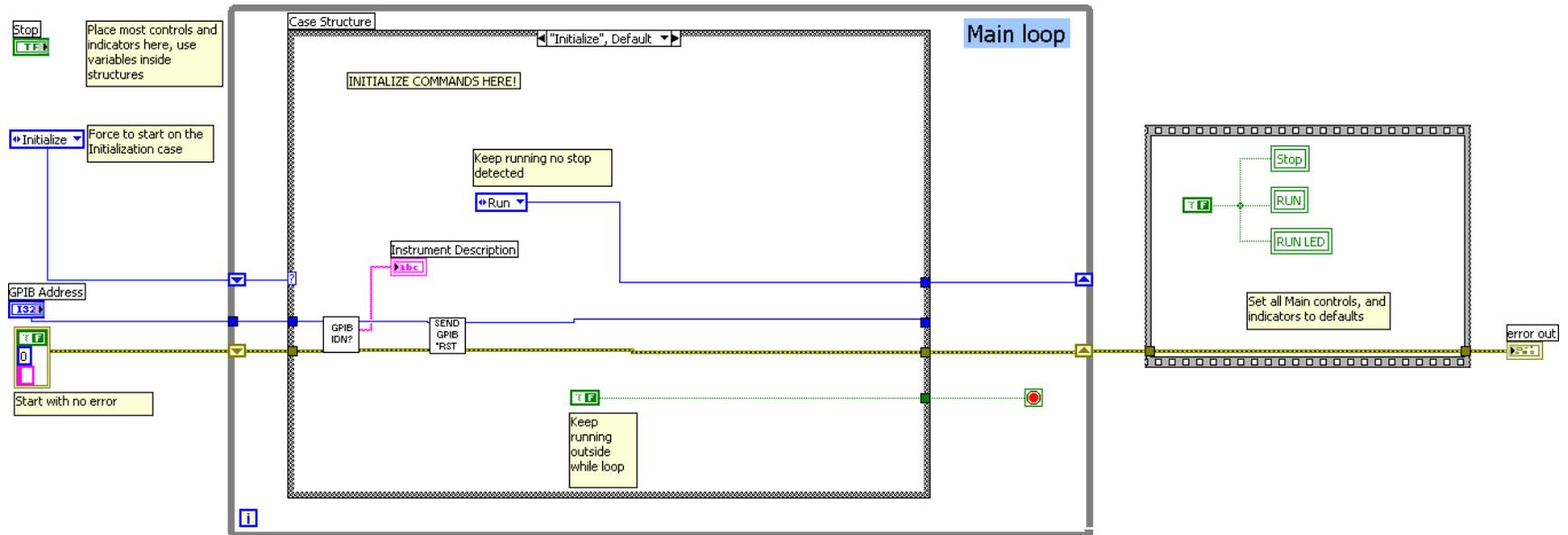
http://zone.ni.com/devzone/cda/tut/p/id/3331
http://zone.ni.com/devzone/cda/tut/p/id/5223

Figure 1: Initialization case (forced by enum constant)
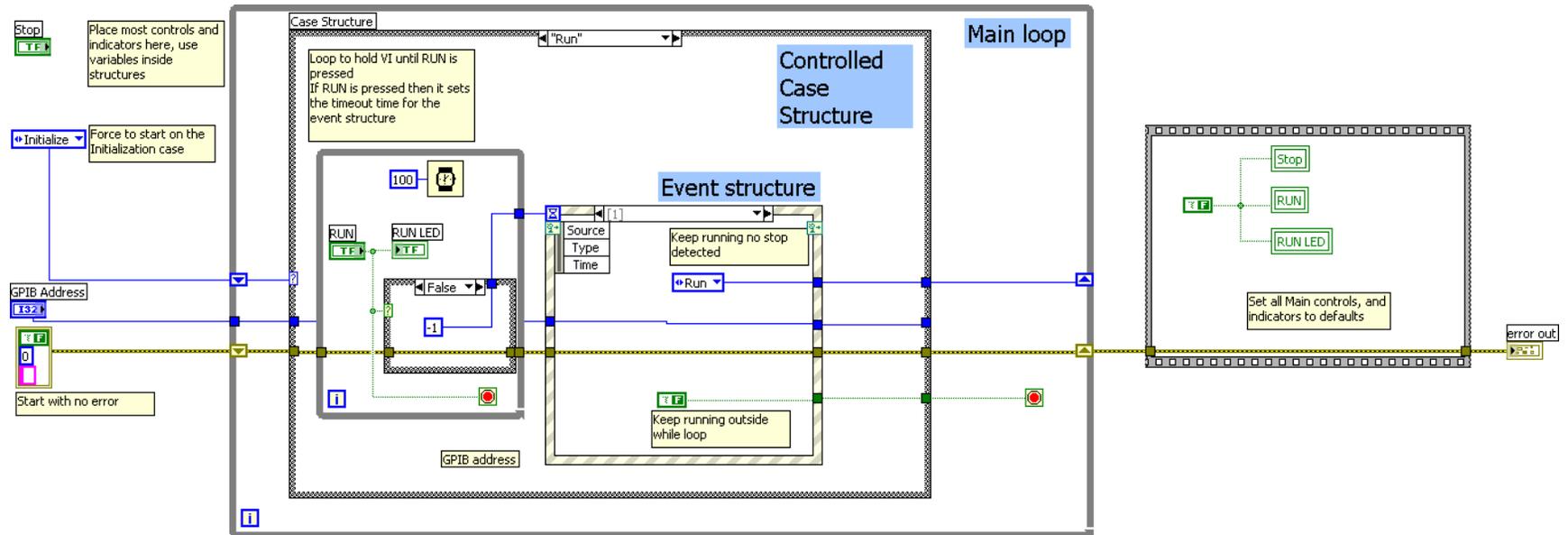
Figure 2: Run case (forced to keep running unless Stop is detected)

Stop

Place most controls and indicators here, use variables inside structures

Case Structure

Loop to hold VI until RUN is pressed
If RUN is pressed then it sets the timeout time for the event structure

◄ "Run" ►

Main loop

Controlled Case Structure

Initialize

Force to start on the Initialization case

100

RUN

RUN LED

Event structure

◄ [2] "Stop": Value Change ►

Stop detected force to go to Quit state

Quit

◄ True ►

Timeout time (ms)

GPIB Address

Stop

RUN

RUN LED

Start with no error

Source
Type
Time

Keep running outside while loop

GPIB address

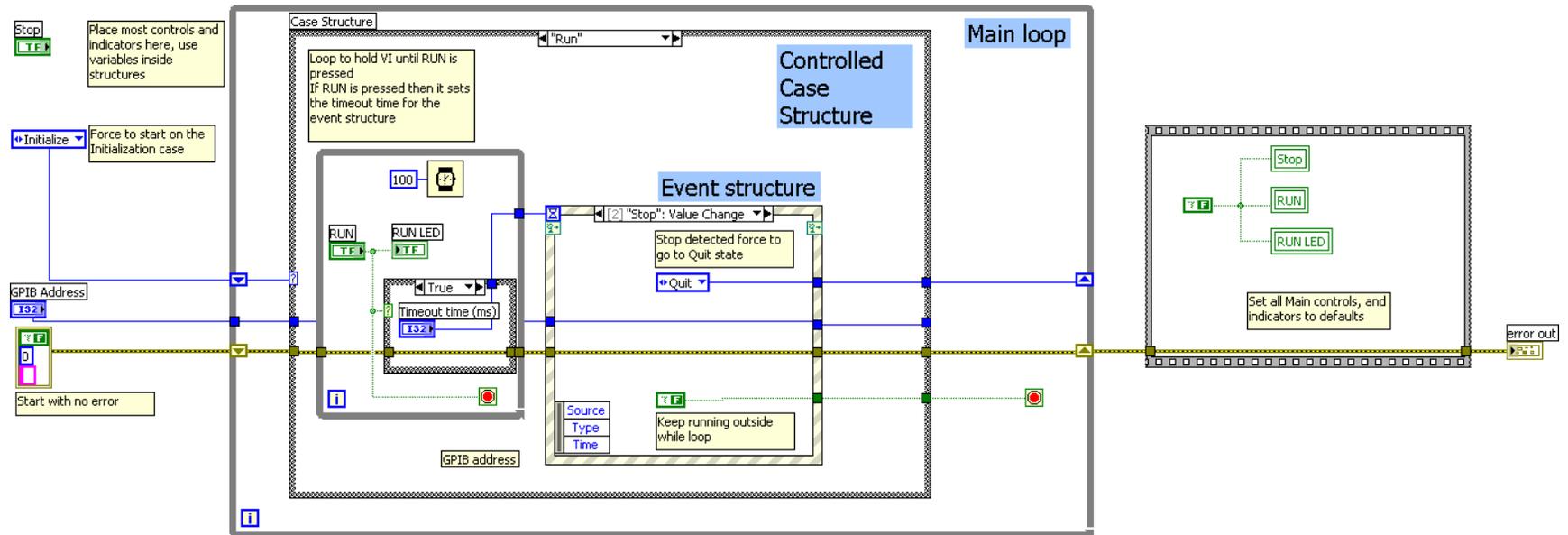Set all Main controls, and indicators to defaults

error out
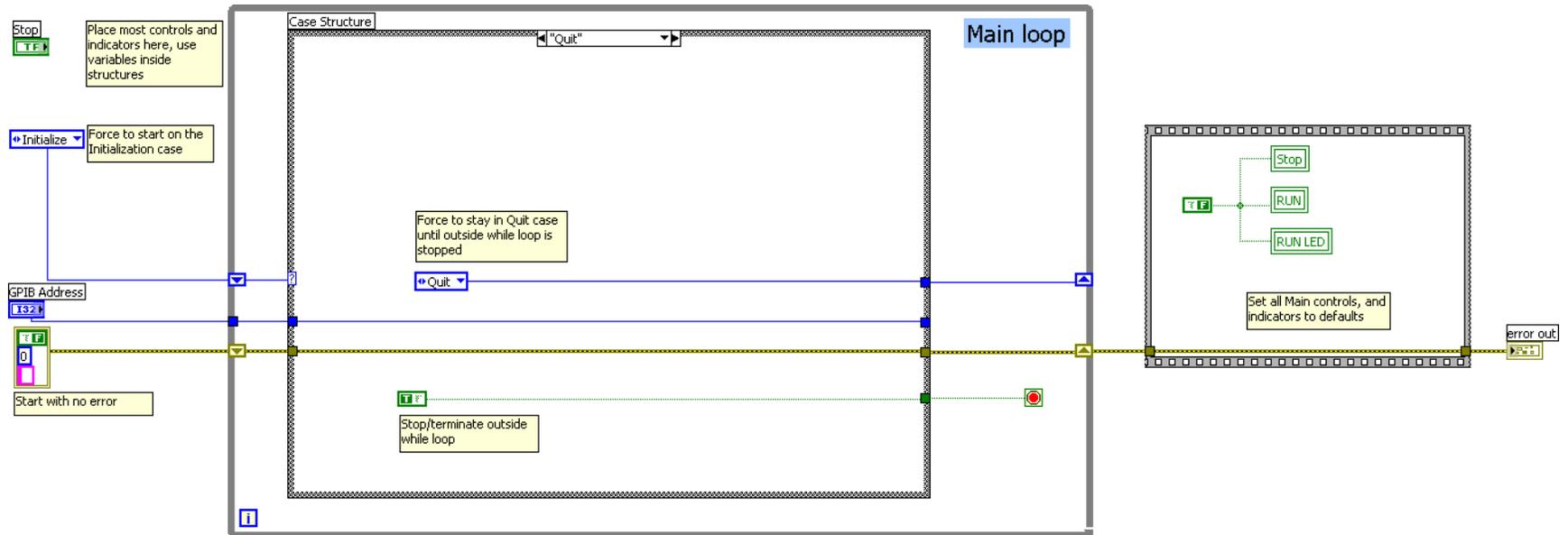
Figure 3: Event structure timeout case controlled by Timeout (ms) control

Figure 4: Quit case (forced by event structure when detects the Stop control)
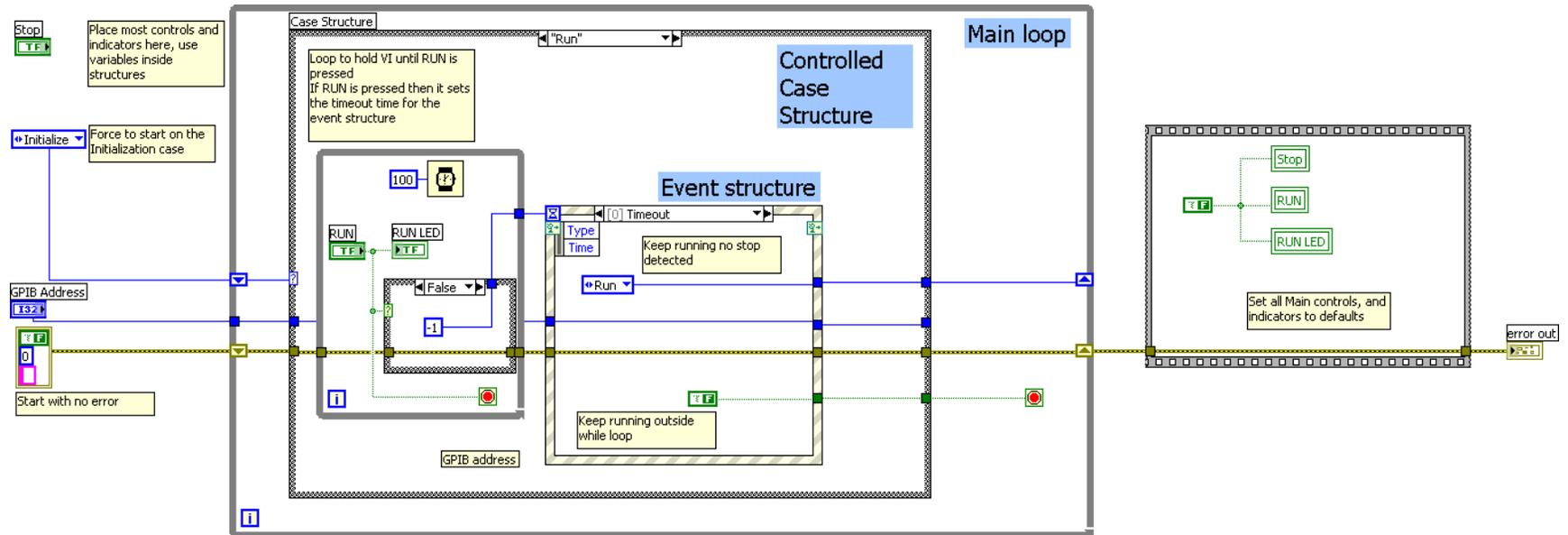
Stop

Place most controls and indicators here, use variables inside structures

Initialize — Force to start on the Initialization case

GPIB Address

Start with no error

Case Structure

Loop to hold VI until RUN is pressed
If RUN is pressed then it sets the timeout time for the event structure

"Run"

Controlled Case Structure

Main loop

100

Event structure

[0] Timeout

RUN

RUN LED

Type

Time

Keep running no stop detected

False

Run

-1

GPIB address

Keep running outside while loop

Stop

RUN

RUN LED

Set all Main controls, and indicators to defaults

error out

Figure 5: Event Driven