

<https://www.halvorsen.blog>



# LabVIEW State Machine

Creating LabVIEW Applications using the State Machine Principles

Hans-Petter Halvorsen

# LabVIEW

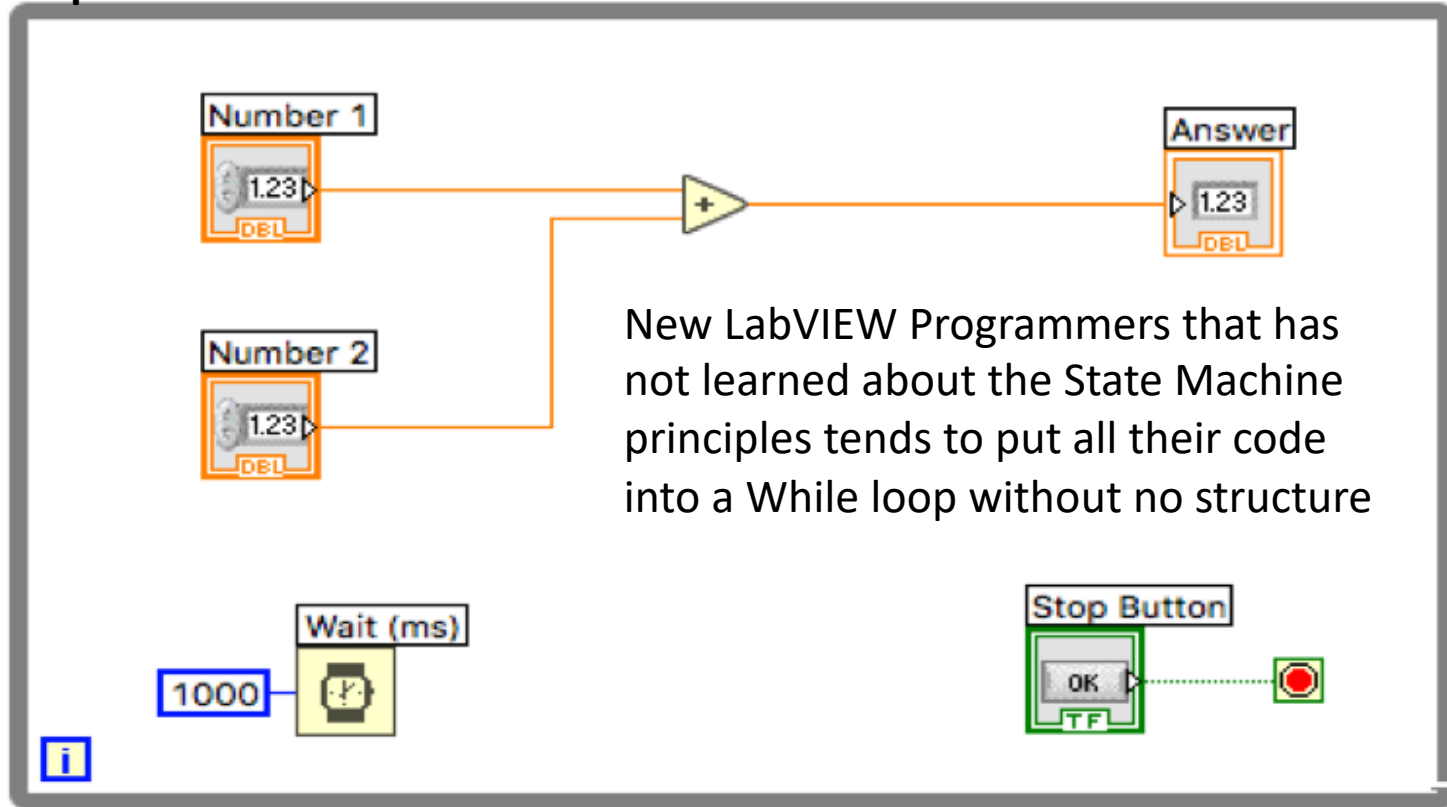
- LabVIEW is a graphical programming language
- LabVIEW has powerful features for Simulation, Control, Vision and DAQ Applications

## Resources:

- <https://halvorsen.blog/documents/programming/labview/>
- [https://halvorsen.blog/documents/teaching/courses/labview\\_automation.php](https://halvorsen.blog/documents/teaching/courses/labview_automation.php)

# Basic LabVIEW Example

## While Loop



New LabVIEW Programmers that has not learned about the State Machine principles tends to put all their code into a While loop without no structure

# Simple LabVIEW VIs vs. LabVIEW Applications

- Typically engineers often create simple LabVIEW VIs that eventually grow out of control, because they don't have the proper structure and best practices.
- The solution to this problem is organizing your code and data in a way that enables modularity, readability, and reuse.
- Using a state machine approach is a good way to make it right from the early beginning.

















# State Machine

The **state machine** is one of the fundamental architectures LabVIEW developers frequently use to build applications

In LabVIEW software, you can create a basic state machine with a **While loop**, a **Shift Register**, a **Case Structure**, and some form of case selector


















## Programming

↑ Search Customize

 Structures	 Array	 Cluster, Class, & Variant
 Numeric	 Boolean	 String
 Comparison	 Waveform	 Collection
 File I/O	 Timing	 Dialog & User Interface
 Synchronization	 Graphics & Sound	 Application Control
 Report Generation		

## Structures

↑ Search Customize

 For Loop	 While Loop	 Timed Structures
 Case Structure	 Event Structure	 In Place Element Structure
 Flat Sequence	 Formula Node	 MathScript
 Diagram Disable Structure	 Conditional Disable ...	 Type Specialization ...
 Shared Variable	 Local Variable	 Global Variable
 Decorations		 Feedback Node

# While Loop

# Case Structure

Initialize

## Shift Register

100

"Event Loop", Default

# Event Structure

Event Loop

Source  
Type  
Time

100 Wait (ms)

Tab Control

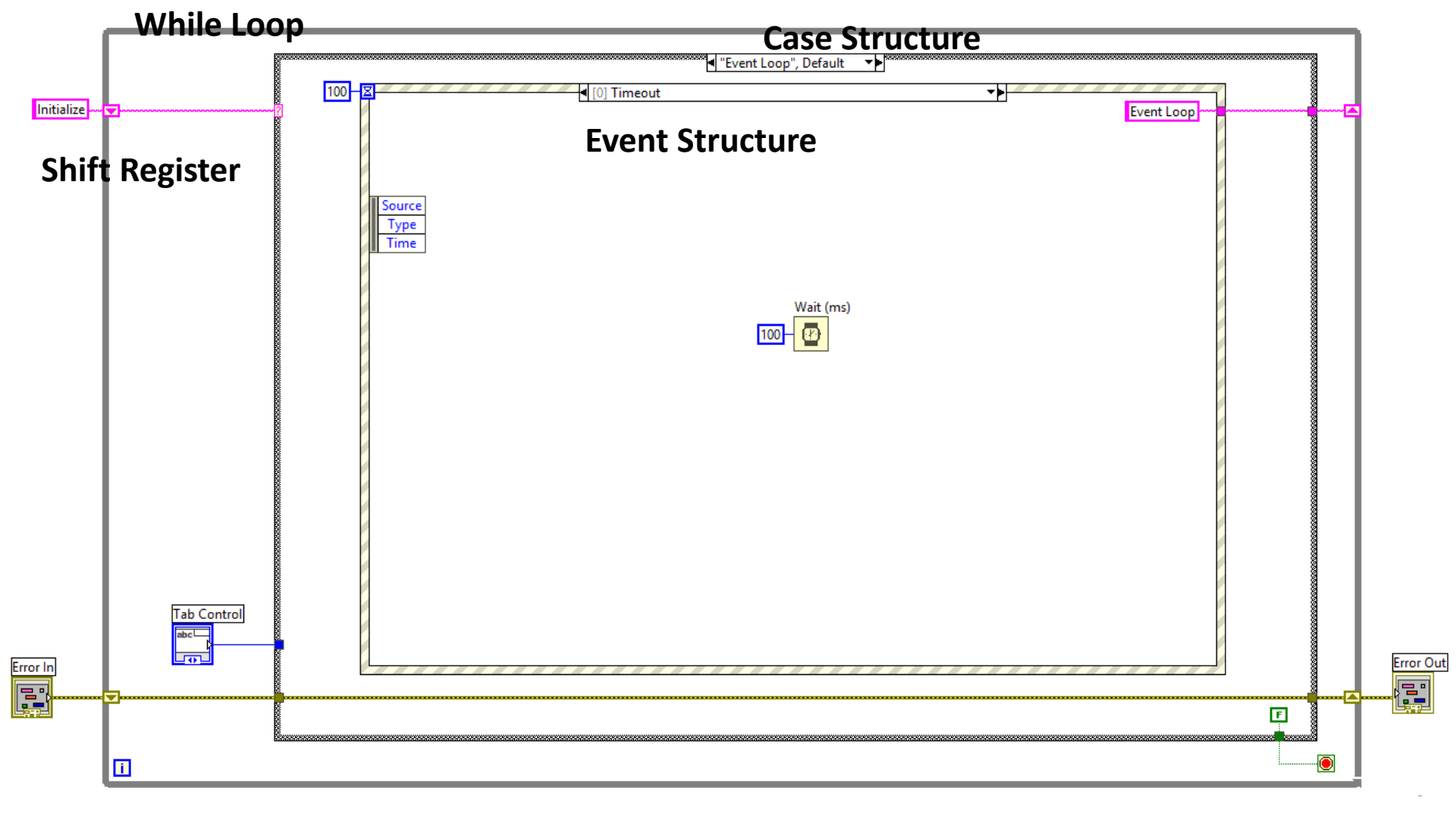
abc

Error In

Error Out

1

F

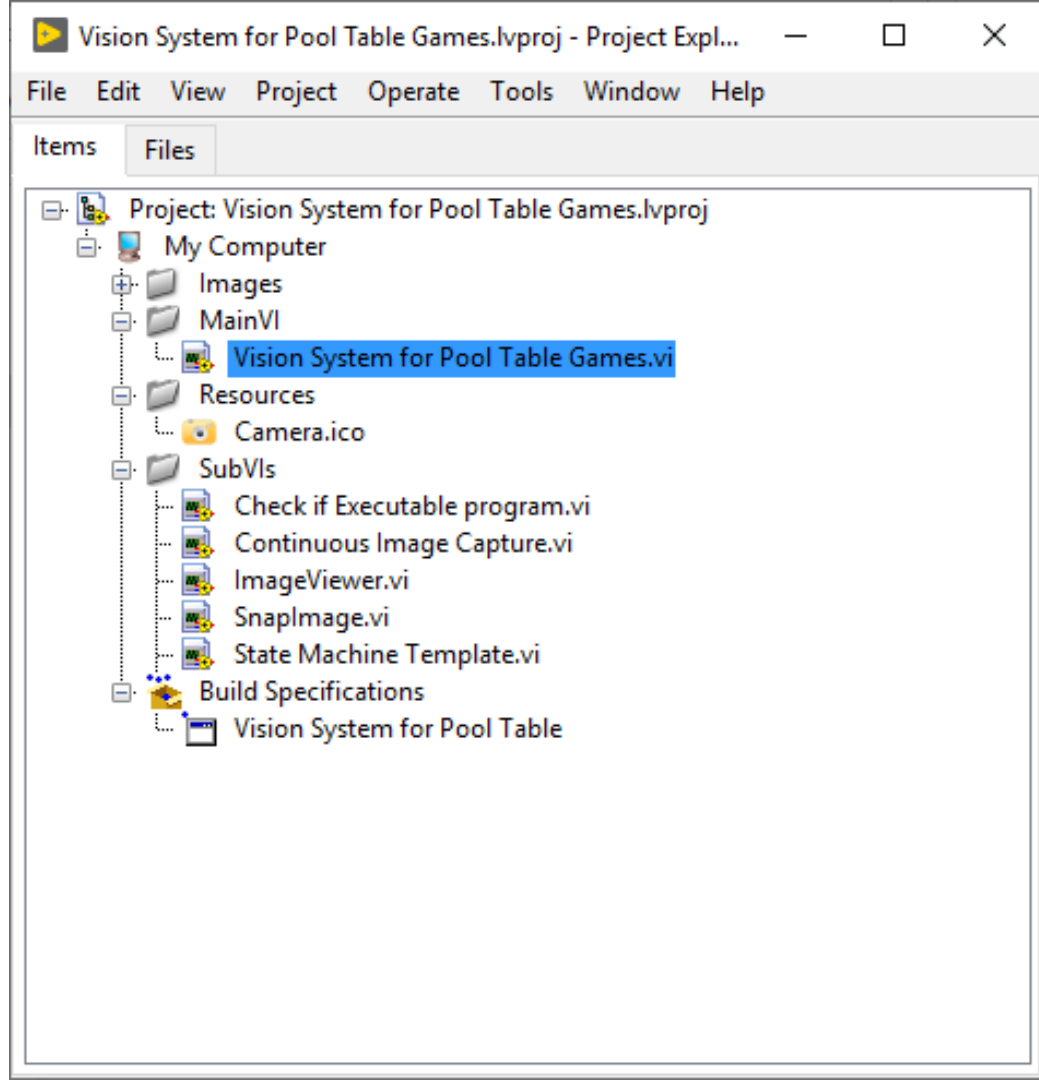


# Structure your Code!

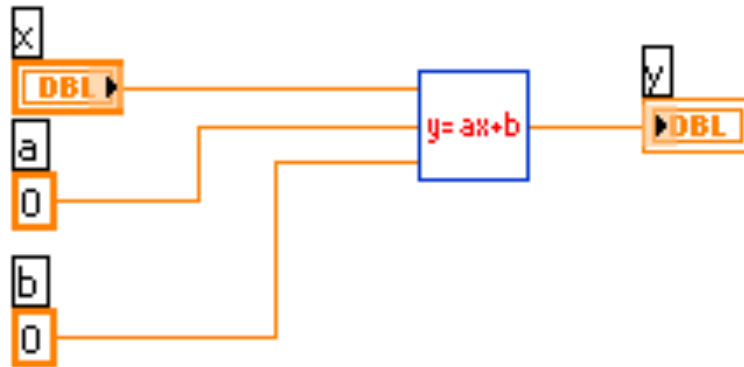
- Use the State Machine principles
- Use the Project Explorer
- Create and use SubVIs



# Project Explorer



# SubVIs



<https://www.halvorsen.blog>



# LabVIEW Example

Hans-Petter Halvorsen

LabVIEW Main Application.vi Front Panel on LabVIEW State Machine.lvproj/My Computer

File Edit View Project Operate Tools Window Help

15pt Application Font Search

SensorList

- Temp1
- Temp2
- Temp3
- Temp4

Celsius

Fahrenheit

14.9

58.7

New

Exit

LabVIEW State Machine.lvproj/My Computer

The image shows a LabVIEW front panel with a light gray background. At the top is a menu bar with 'File', 'Edit', 'View', 'Project', 'Operate', 'Tools', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons, a font size dropdown set to '15pt Application Font', and a search box. The main area contains a 'SensorList' window with a list of 'Temp1', 'Temp2', 'Temp3', and 'Temp4', where 'Temp4' is highlighted in yellow. To the right are two vertical thermometers: 'Celsius' with a scale from 10 to 30 and a yellow fill up to 14.9, and 'Fahrenheit' with a scale from 0 to 100 and a red fill up to 58.7. Below the thermometers are two buttons: 'New' with a plus sign and 'Exit' with a red square. The status bar at the bottom shows the file path 'LabVIEW State Machine.lvproj/My Computer'.

While Loop

Case Structure

Event Structure

"Wait", Default

[1] "New Sensor Button": Value Change

New Sensor Button



- Source
- Type
- Time
- CtlRef
- OldVal
- NewVal

New Sensor

Initialize

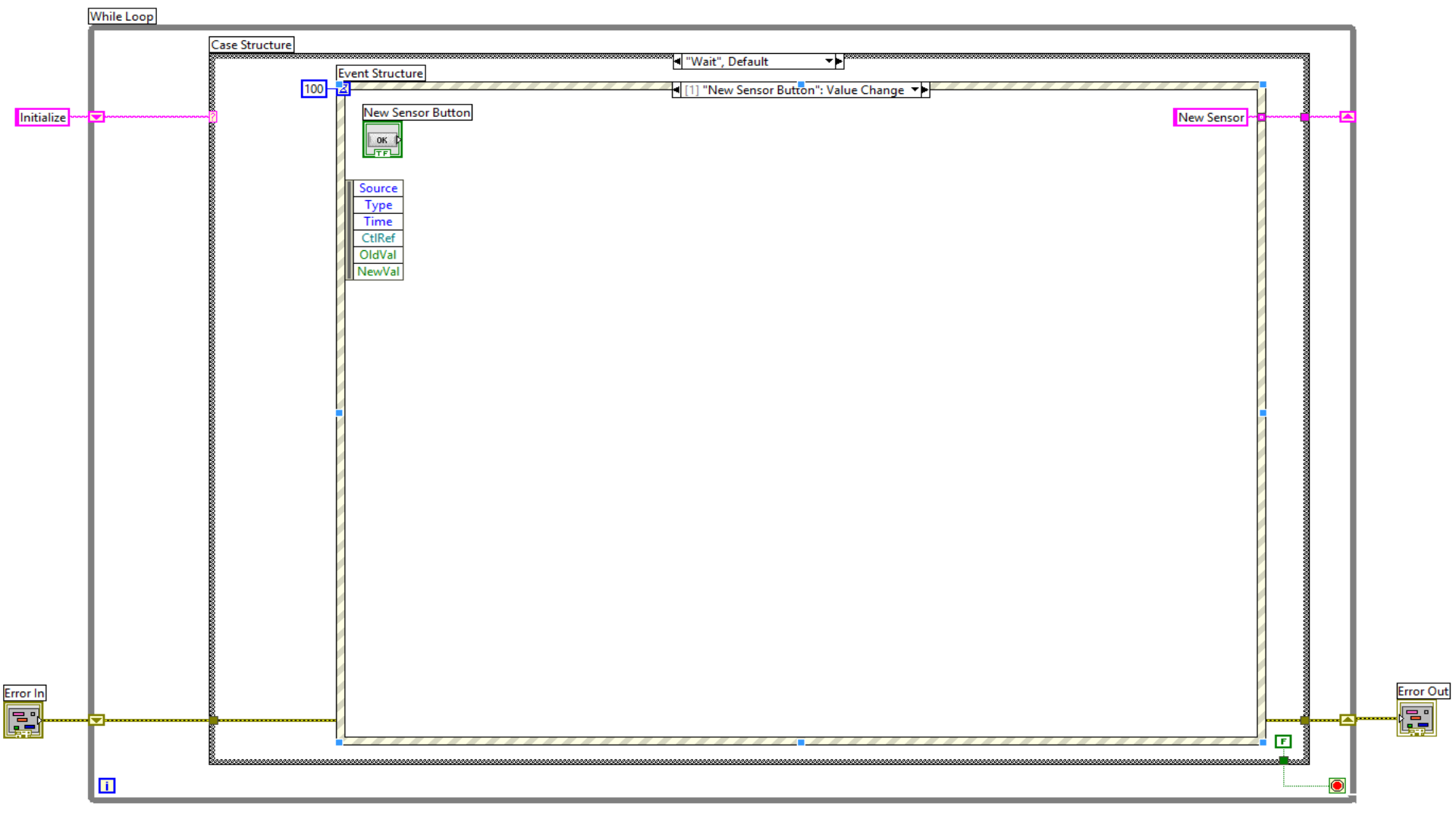
Error In

Error Out

I

F

●



- Project: LabVIEW State Machine.lvproj
  - My Computer
    - Images
    - Temperature.ico
    - MainVI
      - LabVIEW Main Application.vi
    - Resources
    - SubVIs
      - Convert to Fahrenheit.vi
      - New Sensor.vi
    - Build Specifications
      - LabVIEWApp

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

