

Week 3
Exercise 1

Simple State Machine

Aim

- **To understand the sequence of state machine**

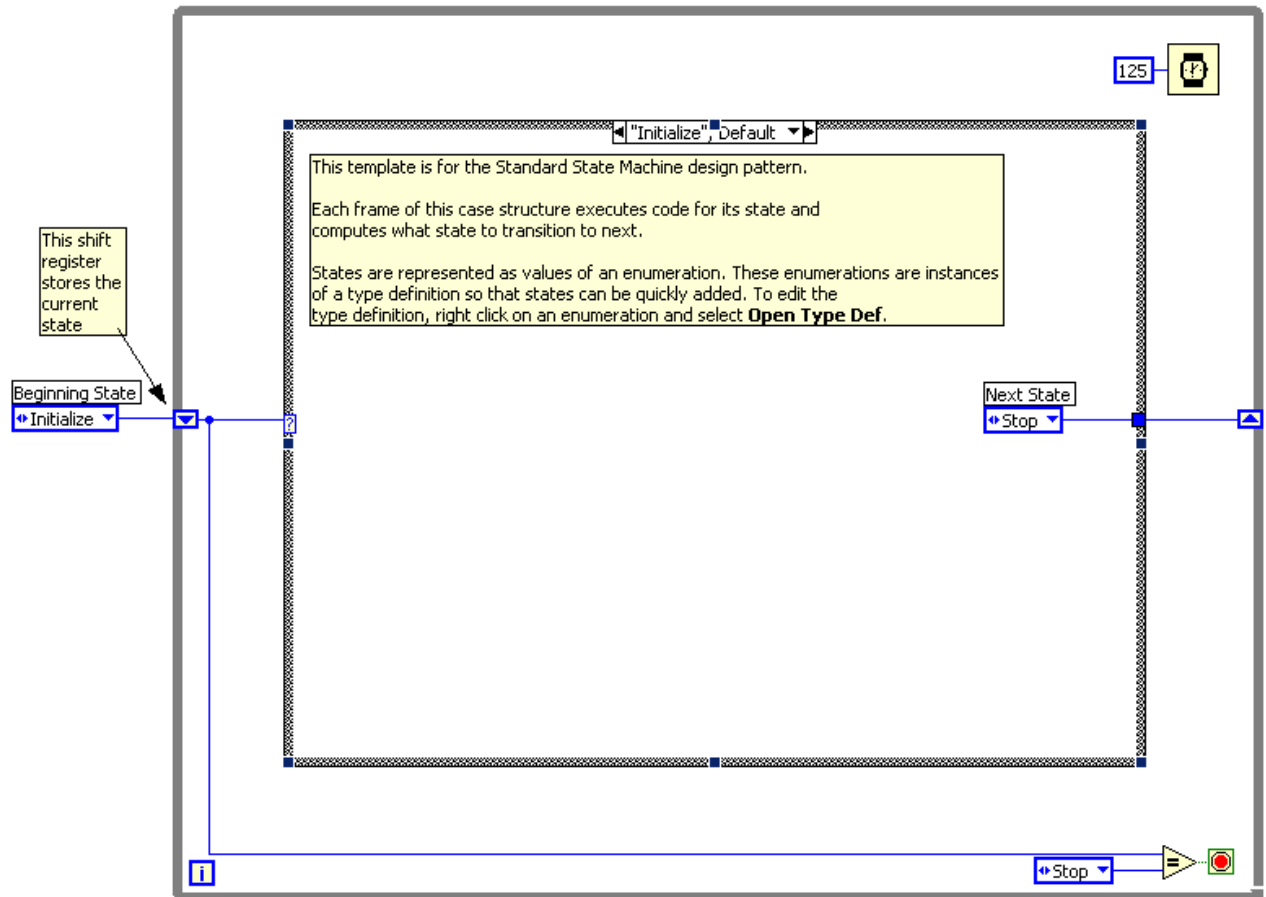
Create a VI using state machine architecture that simulates a simple test sequence. The VI will have an initial state, where it will display a pop-up message indicating that it is starting the test. Then it will proceed to the next case and then to the final state where it will ask the user whether to start over or end the test.

Front Panel

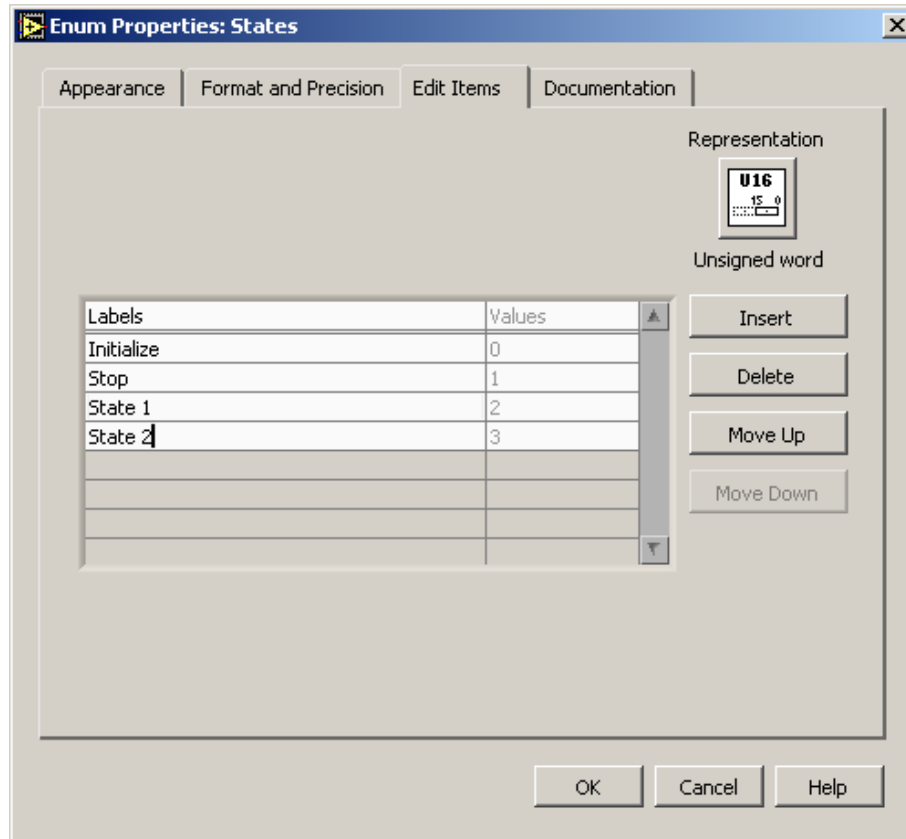
Rather than start from scratch, we will use a VI template to create our state machine.

1. From the initial Lab View screen click on **New...**, and choose Standard State Machine, which is located under the **VI from Template » Frameworks » Design Patterns** heading.
2. Examine the template, and then save it in another directory before you begin working on it.

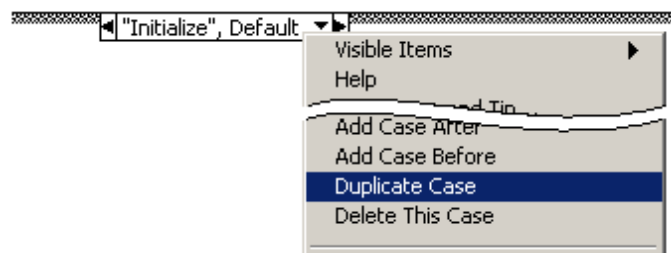
Block Diagram



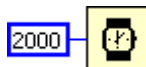
3. Right click on the enum constant labeled Next State and select Open Type Def.
4. On the front panel of the StateMachinesStates.ctl Type Def VI, right click on the States enum control and choose **Edit Items**.
5. Add two more states. Call them "State 1" and "State 2"



6. Close the State Machines.ctl Type Def Front panel and save the control with the default name when prompted.
7. Right click on the Case Selector Label of the case structure and choose Duplicate case. Do this one additional time so that there are four cases: Initialize, State 1, State 2, and Stop.

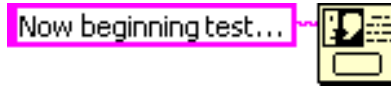


8. Change the value connected to the Wait function to 2000.

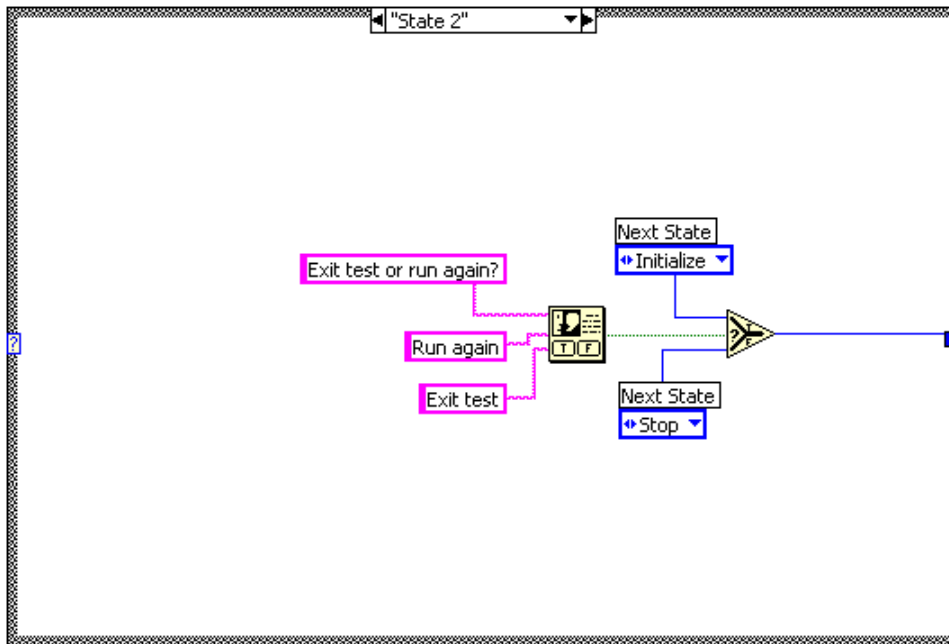


9. Right click on the shift register on the left side of the while loop and create an indicator. Change its name to "Current State".

10. In the “Initialize”, Default case place a One Button Dialog function and wire a string constant into the **Message** input. Type “Now beginning test...” into the string constant.



11. Change the enum constant labeled Next State to “State 1”.
12. Change to the next state in the case structure (“State 1”) and change the enum constant labeled Next State to “State 2”.
13. Change to the next state (“State2”) and add the following code.
 - a. Place a Select function and connect two enum constants (*Tip: Copy the enum constants from one of the previous cases*)
 - b. Place a Two Button Dialog and wire create the constants as illustrated below.



14. Run the VI.
15. Save and close the VI.

End of Exercise

Week 3
Exercise 2

Digital count VI

Aim

- **To write a VI that converts the number of events generated by the quadrature encoder to a digital number to display on the Led's on the DAQ signal accessory.**

Theory

Digital I/O

Each Led is wired to a digital line on the DAQ device. These lines are numbered 0, 1, 2, and 3, starting with the Led on the right. You can write to these lines individually or as a digital port. However, the digital port includes all 8 DIO lines. Because the quadrature encoder uses DIO6 for up/down counting, you cannot wire to DIO6. Therefore in this example you should write a Boolean array to digital lines 0-4.

Note: step by step procedure is not summarized in this exercise; hence students are encouraged to use their knowledge to build this VI.

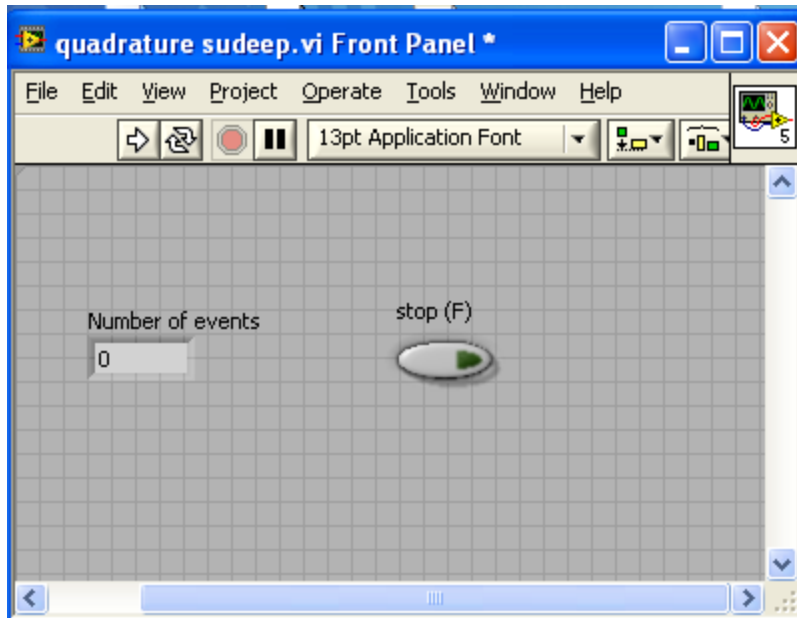
Procedure

Hardware settings

- Disconnect the wire from the analog input 1 to the sine wave.
- Connect a wire from the point **A** of the quadrature encoder to the **source 0** and a wire from **point B** to the **gate 0**

Front Panel

1. The front panel is very simple, and is shown in the figure below.



Block Diagram

2. In order to have a repetitive process a **while loop** is essential.
3. The internal working of the DAQ system can be summarized as follows. The quadrature encoder generates a signal as it is rotated and this signal is transmitted to the digital lines of the Led's. Therefore based on this phenomenon it is necessary to have two DAQ assistants on the block diagram.

Configuring DAQ assistant 1

Acquire signals>counter input>Edge count. Supported physical channel Ctr0.

In the second property box, under **active edge** select **falling** and under **count direction** select **externally controlled**.

Configuring DAQ assistant 2

Generate Signals> Digital output>Line output. Select physical channels **port0 to port 3**. Note that all the 4 channels can be selected by pressing ctrl key.

In the second property box, **check Invert line**, because Led's use negative logic.

4. The data obtained from the Quadrature encoder needs to be converted to dynamic data express VI. **Right click on the block diagram>Express>Signal Manipulation>TO DDT**. Under properties select **single scalar**.
5. Since the digital lines need a Boolean array. **Right click on the block**

Diagram>Boolean>Number to Array.

6. Array subset function is required to resize automatically to display .
Index inputs for each dimension in the array. **Right click on the block diagram> Array>Array subset. The length should be 4.**
7. In order to convert millisecond multiple to millisecond timer value,.
Wait until next ms Multiple is required. **Right click on the block Diagram>Timing>Wait until Ne....**
8. Wire the block diagram.
9. Save the VI
10. Run the VI.

End of Exercise

Week 3

Exercise 3

Introduction to Thermocouples (Challenge)

AIM

- **To understand temperature measurement using thermocouple**
- **To understand strings**

Note: Try to build the block diagram on your own.

Hardware settings

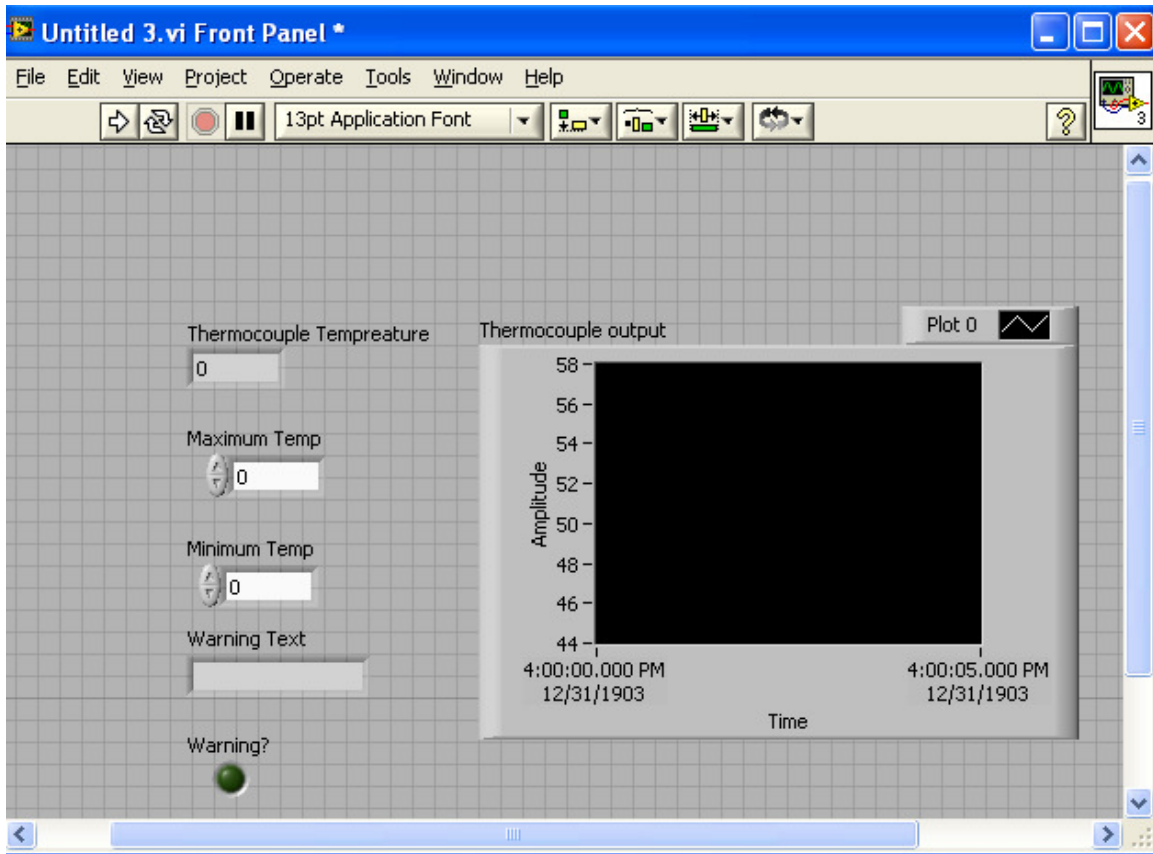
Connect the given thermocouple wire to channel 4 of the DAQ board. Also ensure that a wire is connected between Analog input 1 and the sine wave.

Front panel

Following components are required on the front panel

- Two numeric controls to enter the **maximum** and **minimum** temperature.
- One numeric indicator to record the temperature of the thermocouple.
- One string control to display the warning text
- One Boolean Led to display the warning light.
- One waveform chart to observe the variation in temperature from the thermocouple.

Front panel appears as follows.



Block Diagram

The block diagram consists of the following components

- DAQ assistant to record the thermocouple temperature.
DAQ board configuration is as follows **Acquire signals>Analog input>Temperature>Thermocouple**. Supported physical channel **ai4**. Thermocouple type is **K/J/R/S/T (depending on which you have got) CJC source is constant**
- Save the VI
- Run the VI