

LabVIEW Exercise 1

(credit: Brett Meyer)

Implementing the FSM in LabVIEW

Now you will implement the FSM you designed in LabVIEW.

- 1.) Open LabVIEW and click on the “Create Project” button in the left pane.

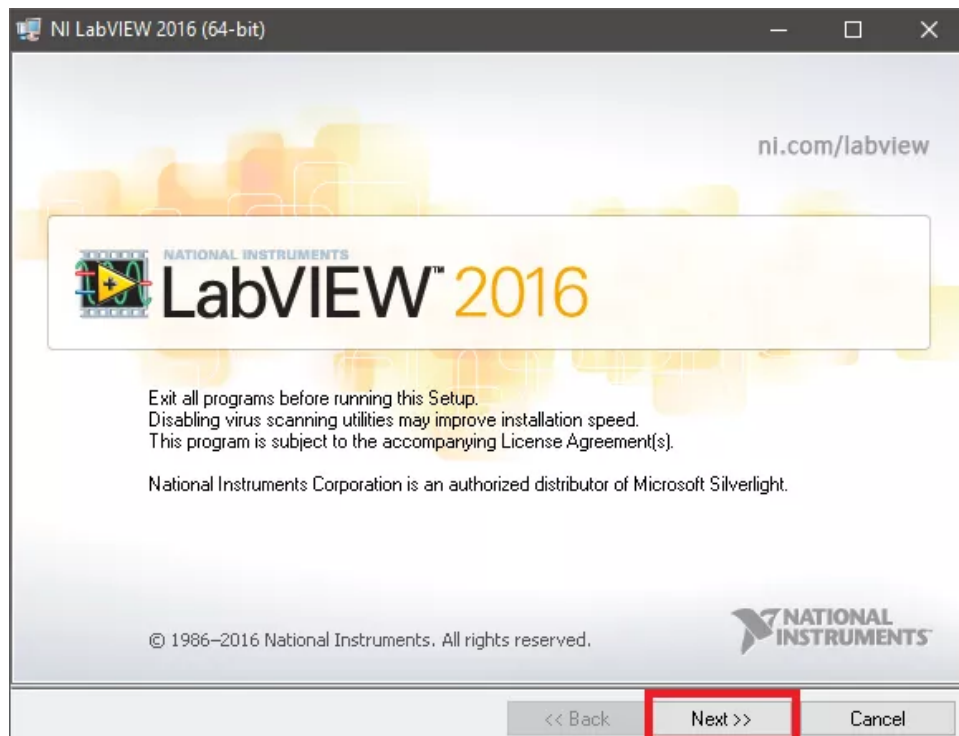


Figure 1: LabVIEW start window.

- 2.) Select “Blank Project” and press “Finish.”

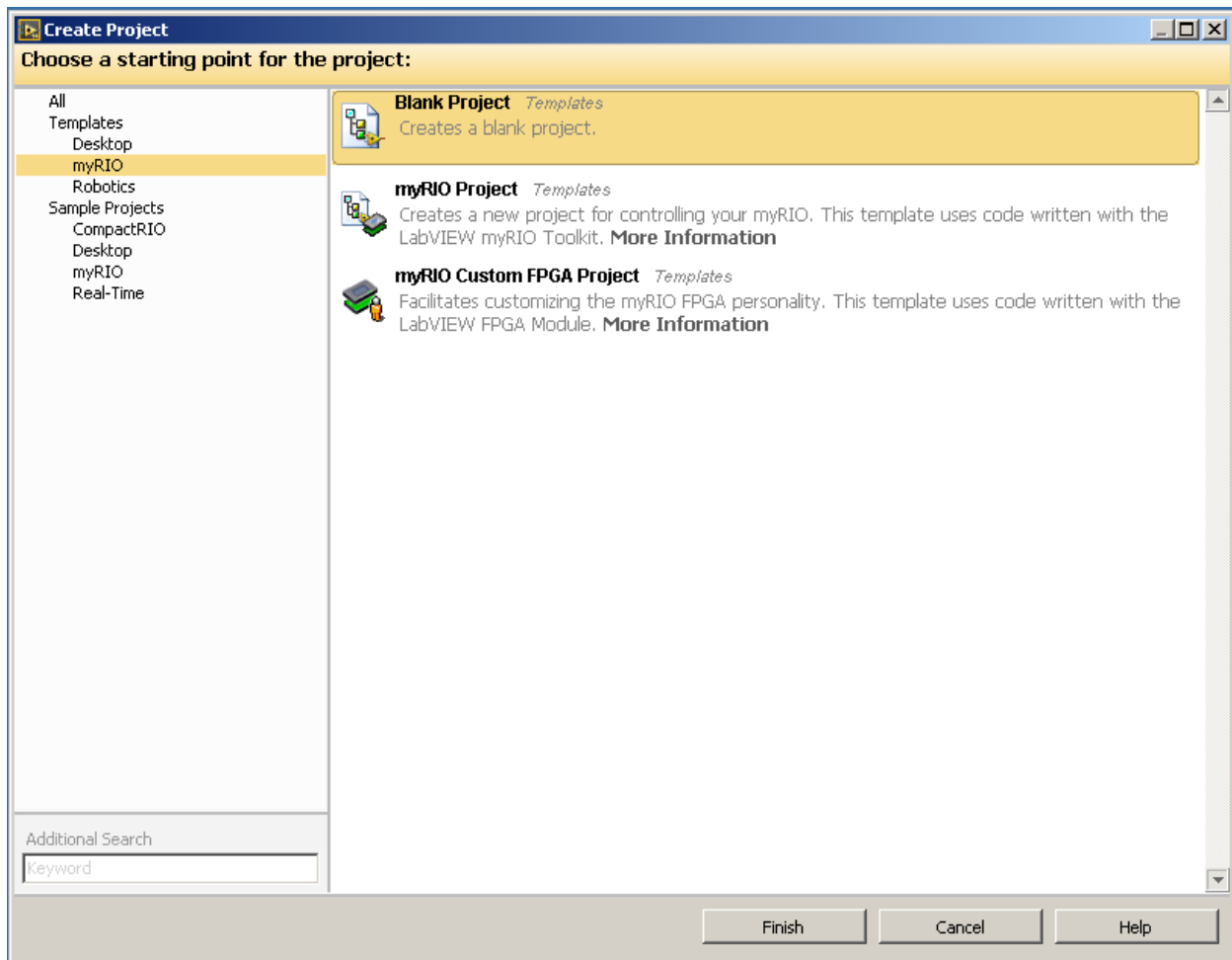


Figure 2: New Project Creation Screen

- 3.) In the Project Explorer, right click on “My Computer”, then “New”, then “VI”, to create a new Virtual Instrument (VI).

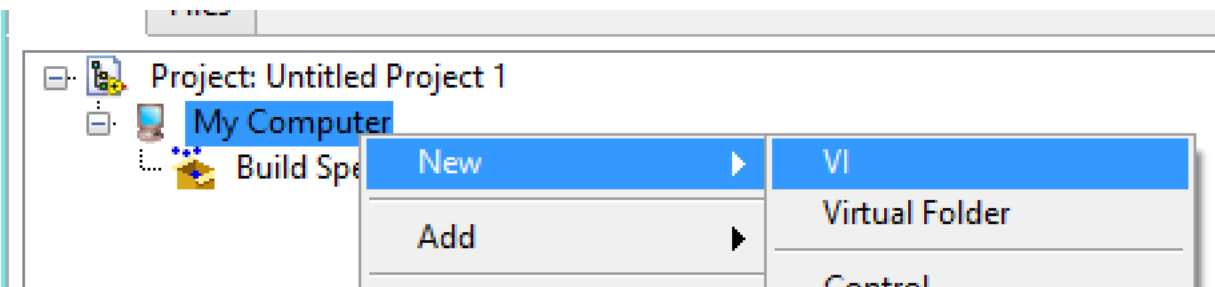


Figure 3: Create new VI

- 4.) The front panel and block diagram should then open up. Make a front panel with (Boolean/OK) buttons representing the user entering a nickel, dime, and quarter. There should be a string output for the total change deposited and also the amount of change to be returned. You can use a (Boolean/Square LED) LED to indicate whether soda has been dispensed.

Also, use a “Take Soda” button to reset the machine after the user has taken their soda.

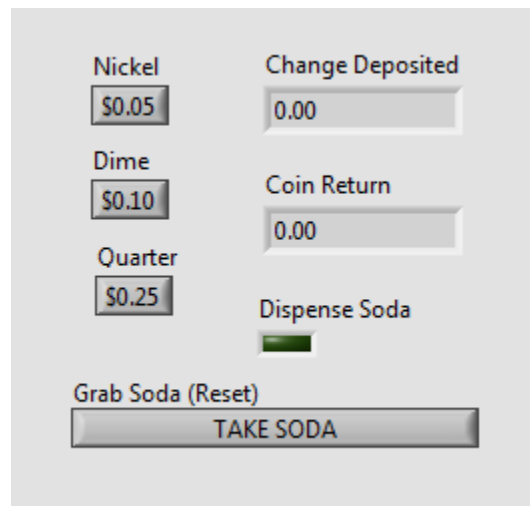


Figure 4: Sample Front Panel

- 5.) Just like in C, an FSM can be implemented as a case structure inside a while loop. The case structure can be found under Programming -> Structures -> Case Structure. Place the case structure inside of a while loop. The state can be kept track of using a shift-register (the little up and down arrows at the edge of the while-loop). To get these, right click on the loop boundary and select “Add Shift Register.” The initial state is piped into the shift register from outside the while-loop. In the code below, an enumeration constant is used so that each state has a unique name. This constant is available on the Numeric palette.

Note that all the enum constants used in the vending machine VI must be identical, and exactly match the names of the cases in your case structure. The easiest way to go about this is to make the first enumeration constant outside of the while loop, and populate it with all of the 10 states. Then, make copies of the first enum constant for all of the other instantiations in the VI, and select the desired case from the drop-down menu, which should reflect all of the previously entered states.

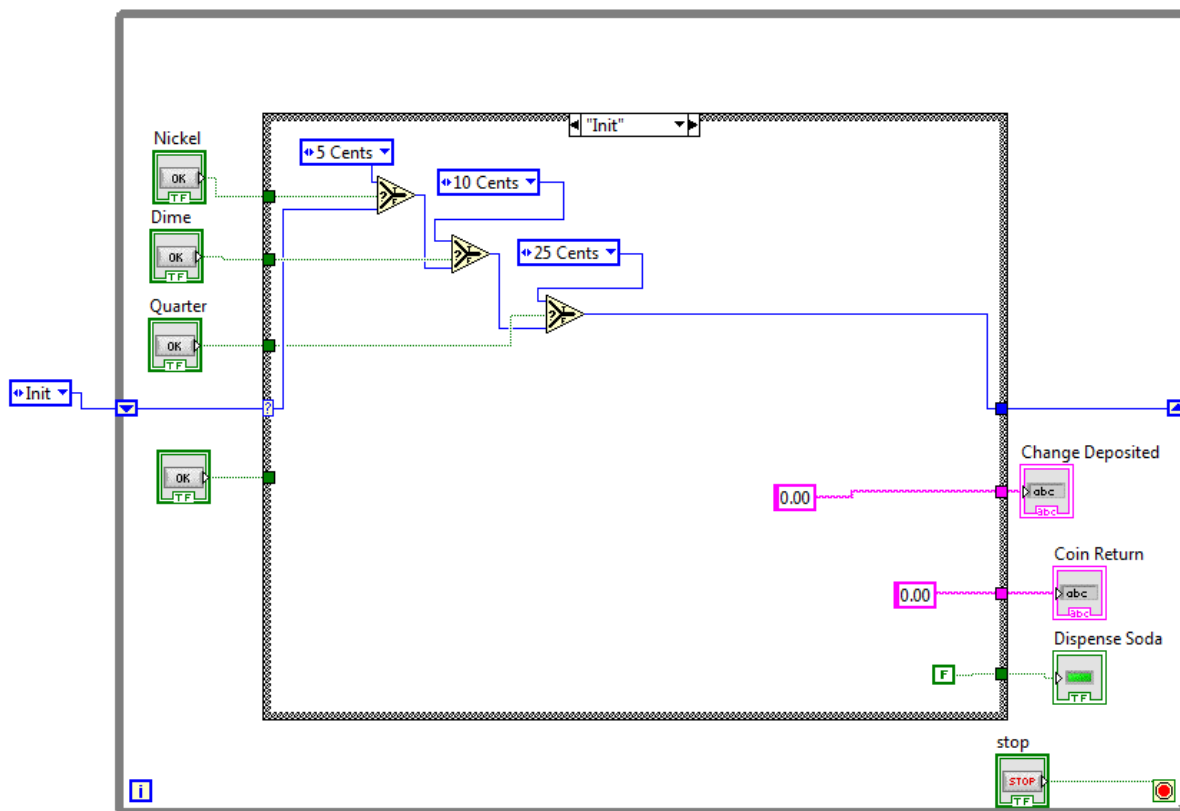


Figure 5: FSM in LabVIEW

- 6.) Connect the Shift register to the little question mark tab on the outside of the case structure. You can add additional cases by right clicking at the case selector on top of the structure and selecting "Add Case After."
- 7.) Fill in the logic for each state. It should calculate the next state based on inputs and should calculate the correct outputs as well. You may find the select block useful (Programming -> Comparison -> Select). The select block is essentially a Multiplexer: The Boolean input decides whether to pass the input wired to the true or false terminal.

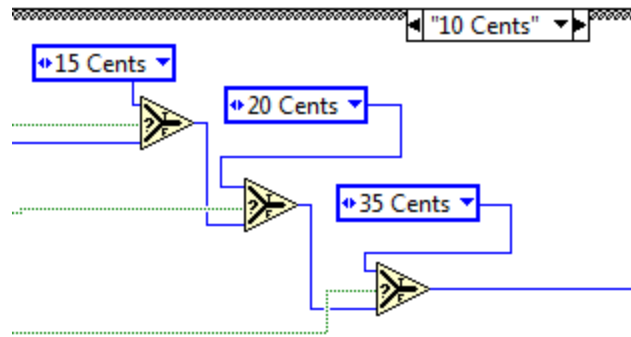


Figure 6: Select Block in Action

Validate

Make sure that your LabVIEW FSM behaves correctly, according to the problem specification. When you are confident that it does, submit your LabVIEW project through the class Moodle.