LabVIEW Generator Controls Workshop

Learning Objectives

The objective of the LabVIEW Powerplant Controls Workshop is to develop understanding of a control system and how to develop, implement, and fine-tune it. Through interactive, hands-on lessons, participants will focus on:

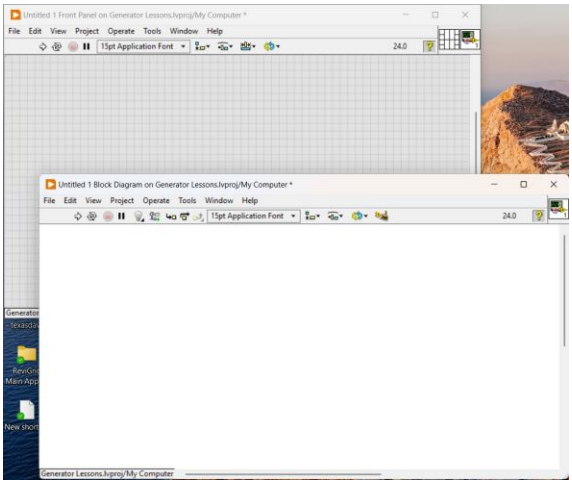
- **Achieving Precision through Closed-Loop Control:**
    - Explore and implement PID (Proportional-Integral-Derivative) control strategies to stabilize system outputs and fine-tune motor and load levels under varying conditions.
    - **Tuning Control Parameters for Optimal Performance:**
        - Adjust and refine control parameters to achieve desired system responses, minimize oscillations, and enhance system stability and accuracy.
- **Implementing System Operations with LabVIEW:**
    - Develop routines to initialize, operate, and collect performance data from a physical generator model, creating a foundation for real-world control applications.
- **Executing Communication Protocols for Device Integration:**
    - Apply basic communication protocols to configure, read from, and write to external devices, facilitating seamless interaction with system components.
- **Building a Functional Control System:**
    - Learn to create control system logic, applying structures like loops and conditionals to effectively manage system behavior.
    - Design intuitive user interfaces for monitoring and adjustment, using real-time charts and control elements.

Each lesson builds toward mastery in developing, executing, tuning and understanding dynamic control systems. By the end, participants will be prepared to design, implement, and optimize control systems, integrating practical programming skills with control engineering principles to address real-world challenges.

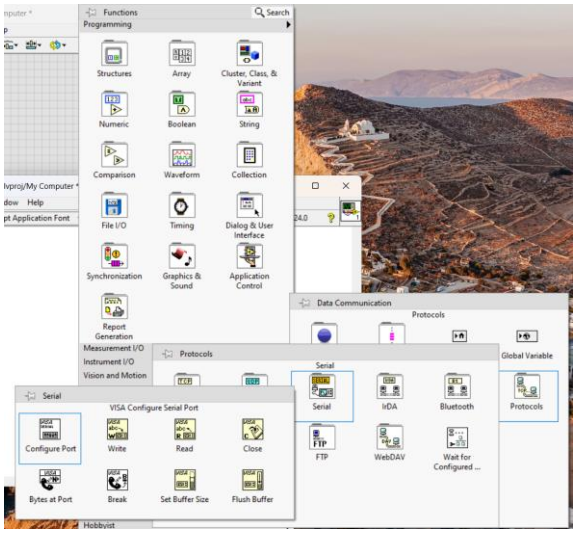
Power Generator Mini-System Plant

Lesson 1: Opening a Serial Port & Communicating with Generator Model

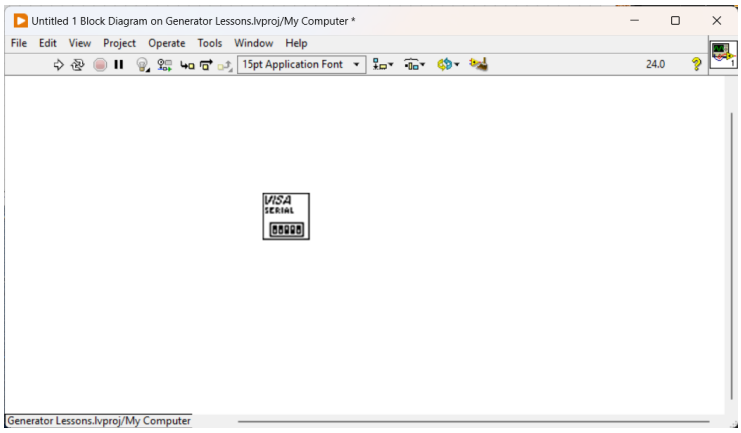
1. **Create a new VI**



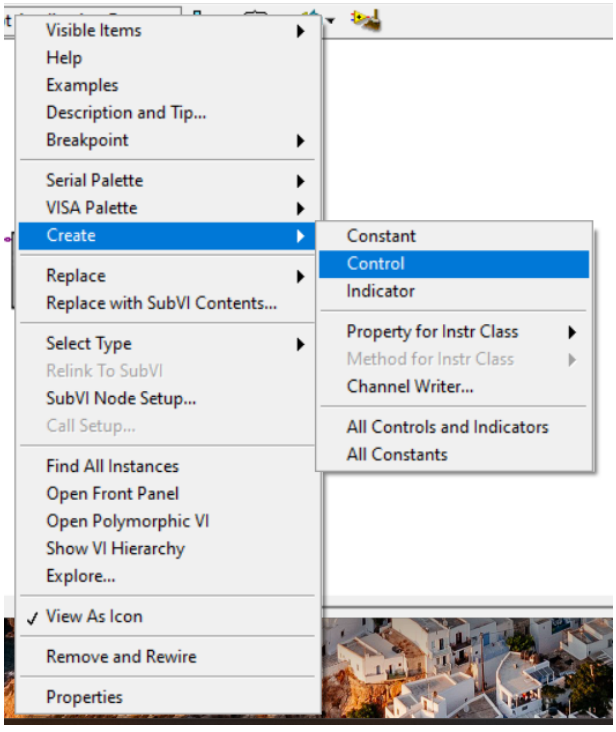
2. **Configure Serial Port**:
    a. **Diagram**: Right-click to access Data Communication>>Protocols>>Serial>>Configure Port.



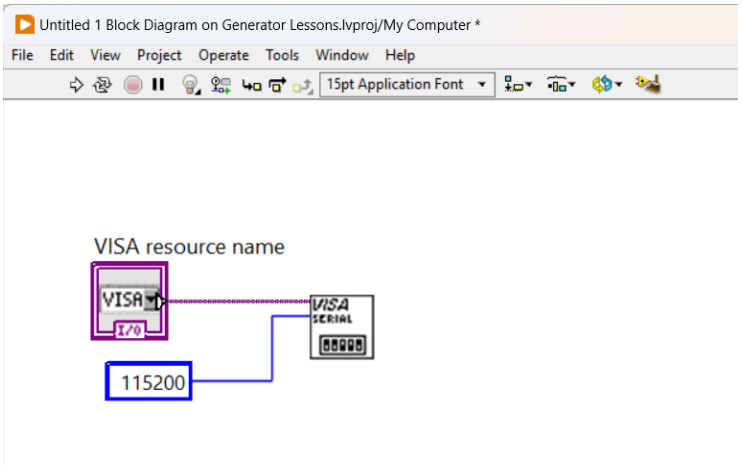
b. Place the Configure Port VI on the diagram.



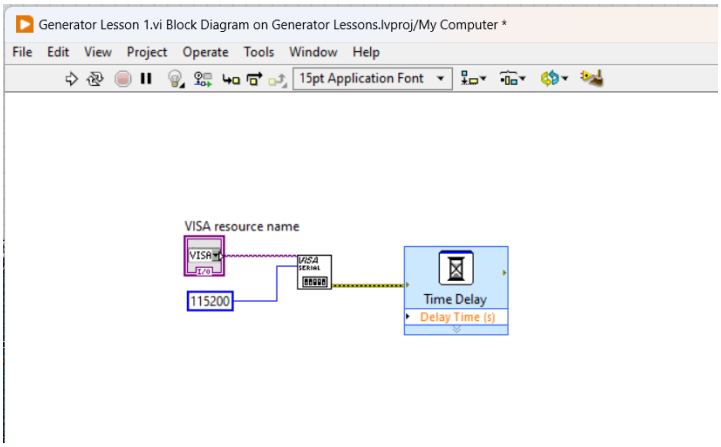
c. Right-click on the Visa resource name terminal and select Create>>Control.



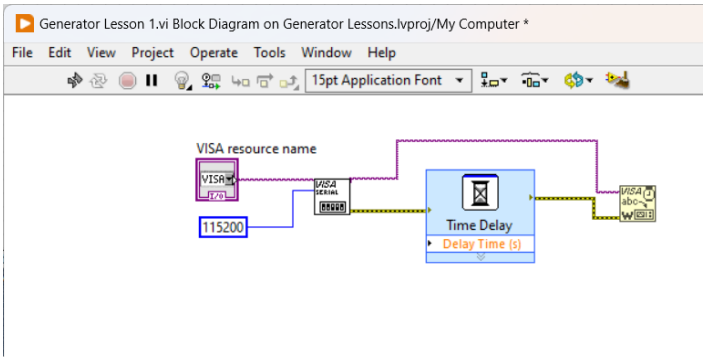
d. Right-click on the baud rate terminal, select Create>>Constant, and change the value from 9600 to 115200.



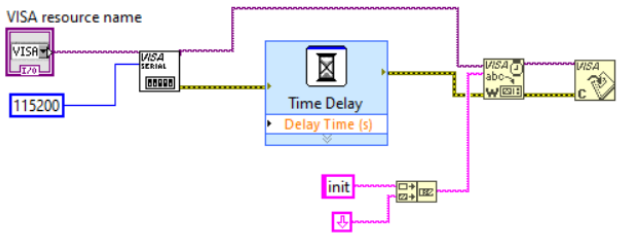
3. **Add Time Delay**:
   a. Right-click on the diagram: Timing>>Time Delay.
   b. Wire error data from the Configure Port to the error input terminal of the Time Delay VI.
   c. Double-click on Time Delay and set it to 3 seconds.



4. **Add Visa Serial Write Command**:
    a. **Add Write Command**: Right-click Data Communications>>Protocols>>Serial>>Visa Write.
    b. Wire the Visa resource name and error info to the Visa Write VI.



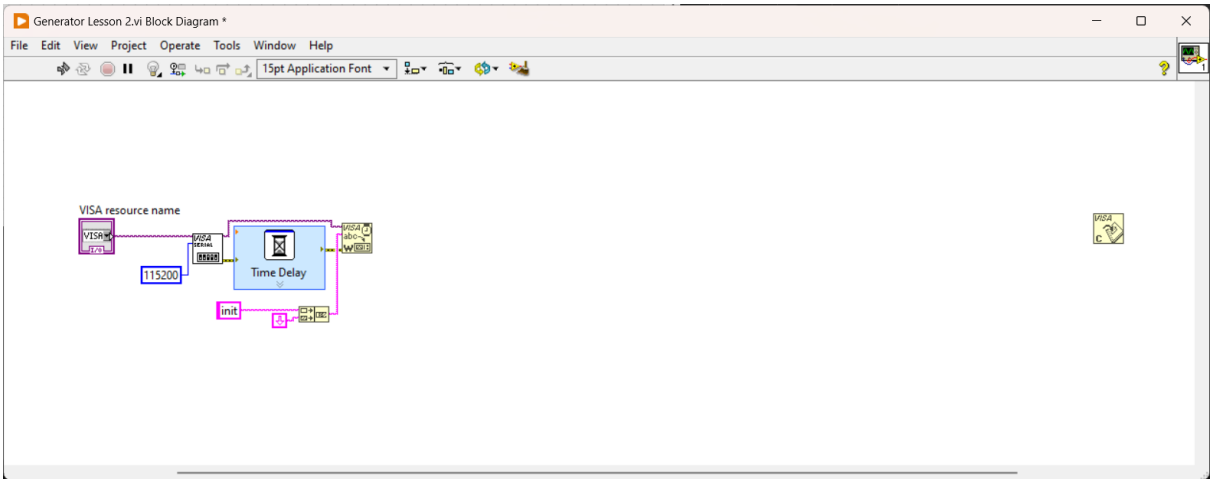
5. **Add String Commands & Visa Close**:
    a. Right-click: String>>Concatenate Strings, String>>String Constant, & String>>Line Feed Constant.
    b. Enter text “init” into the String Constant & wire it to Concatenate Strings VI.
    c. Connect Line Feed constant to bottom terminal of Concatenate Strings VI.
    d. Wire the output to the write buffer terminal of Visa Write VI.
    e. **Close the VISA Port**:
    f. Right-click: Data Communication>>Protocols>>Serial>>Visa Close.
    g. Wire Visa resource and error info to Visa Close.



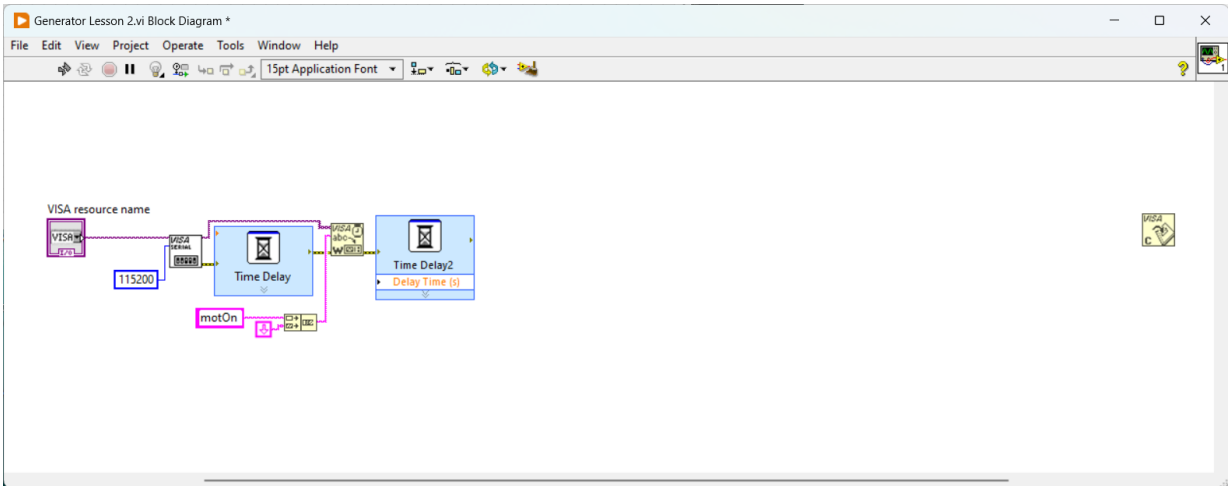
6. **Select COM Port and Run the VI**:
    a. Go to the front panel, select the com port for the Power Plant generator model, and click the Run arrow.

Lesson 2: Collecting and Plotting Data from the Generator Model

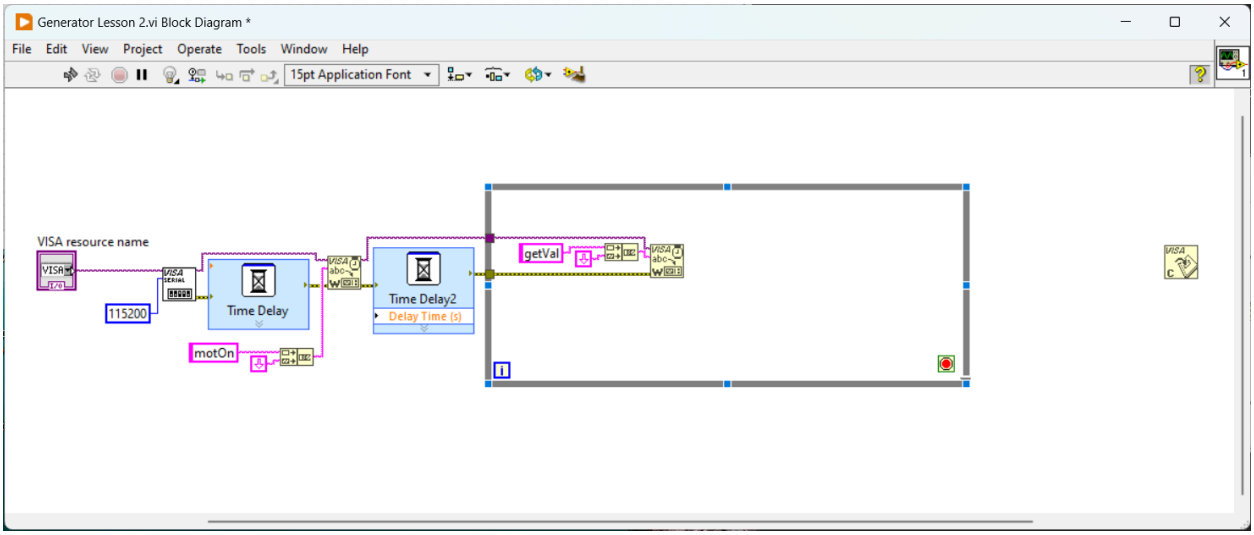
1. **Open Diagram from Lesson 1**.
    a. Disconnect the wires between Visa Write and Visa Close. Move Visa Close to the right to add functions.



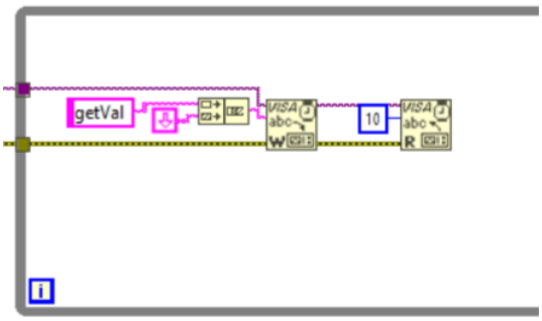
2. **Copy and Modify Time Delay**:
    a. Copy Time Delay function, paste it next to Visa Write VI, and change “init” to “motOn.”
    b. Wire error info from Visa Write to the Time Delay function.



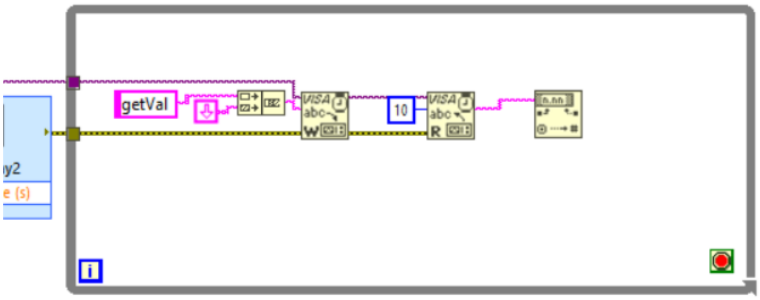
3. **Add While Loop & Visa Write**:
    a. **Diagram**: Right-click Structures>>While Loop.
    b. Copy and paste Visa Write VI, String Constant, Line Feed constant, & Concatenate String group inside the While Loop.
    c. Change “motOn” to “getVal”.



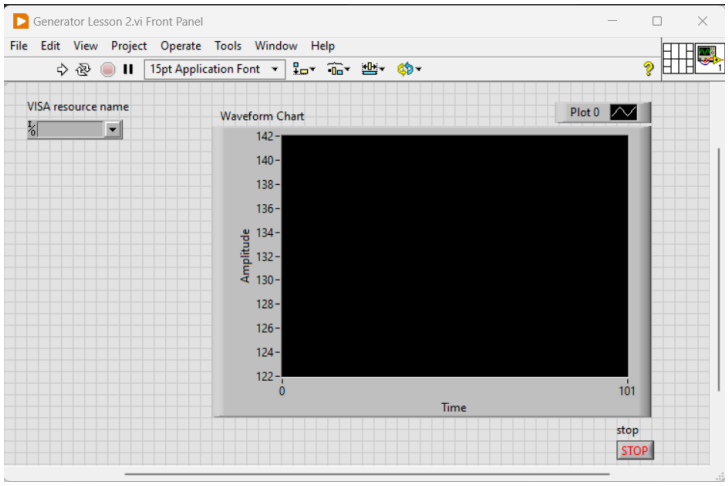
4. **Add VISA Read**:
    a. Right-click: Data Communication>>Protocols>>Serial>>Visa Read.
    b. Wire Visa resource name and error info to Visa Read VI.
    c. Right-click byte count terminal, Create>>Constant, and enter “10”.



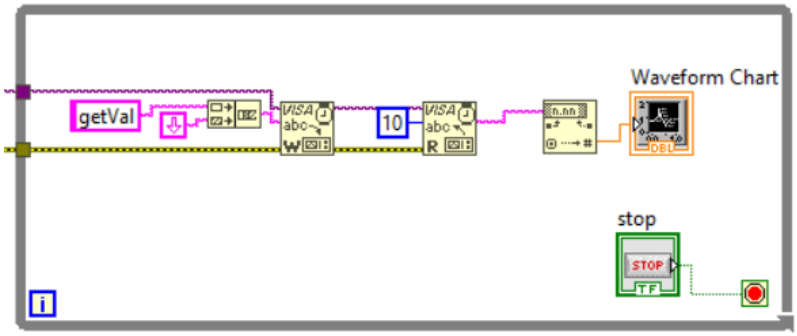
5. **Convert and Plot Data**:
    a. **Diagram**: Right-click String>>Number/String Conversion>>Fract/Exp String to Number.
    b. Wire output of Visa Read to Fract/Exp String to Number VI.



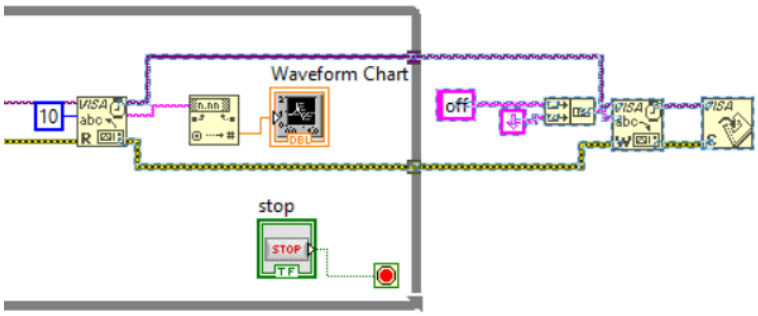
    c. **Front Panel**: Right-click Graph>>Waveform Chart, Boolean>>Stop Button.



d. **Diagram:** Wire numerical output to Waveform Chart and Stop Button to the Loop Condition.



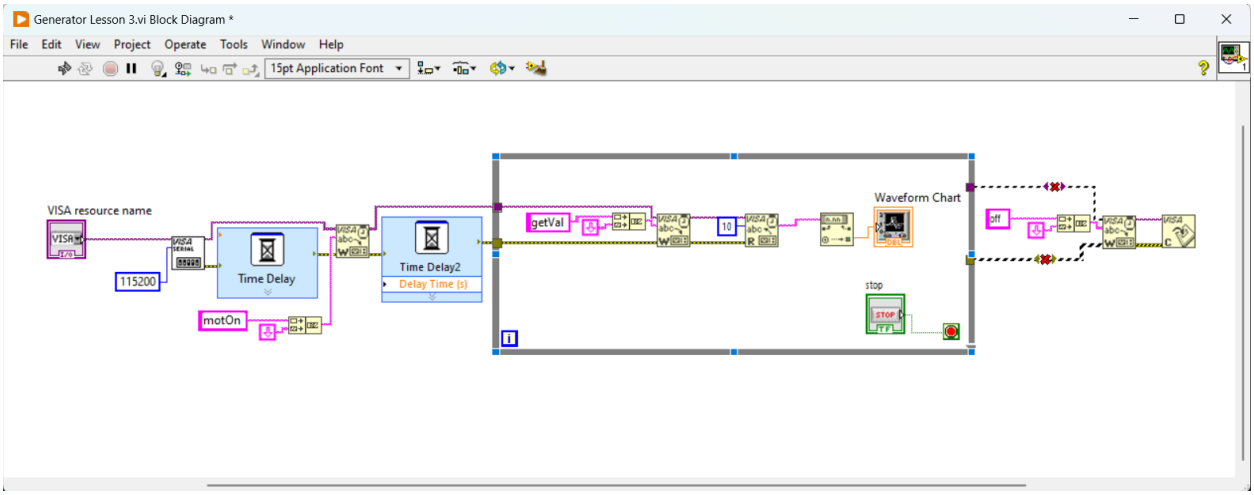
6. **Add Off and Close VI**:
    a. Copy “getVal” constant, Line Feed, Concatenate String, and Visa Write group. Paste between While Loop and Visa Close VI and change “getVal” to “off.”
    b. Wire Visa resource name and error info through the while loop border to the Write and Close VIs



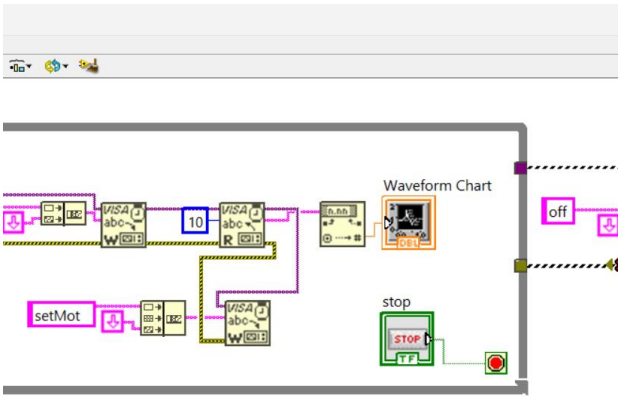
7. **Select Com Port and Run**
    a. **Front Panel:** Select com port Chart.
    b. Click Run arrow. Adjust to see voltage on Waveform

Lesson 3: Controlling the Generator Motor Speed

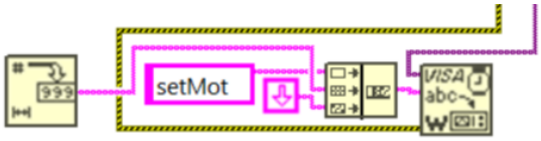
1. **Open Diagram from Lesson 2.**
    a. Delete Visa resource name and error wires from Visa Read VI.



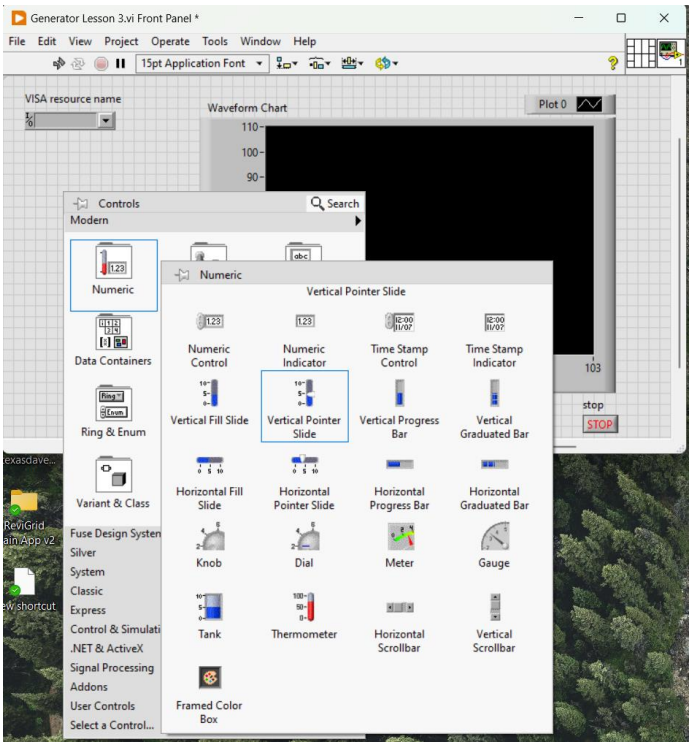
2. **Copy and Modify String Group & add Write:**
    a. Copy the group of VIs: getVal string constant, Line Feed, Concatenate Strings, & Write paste inside the While Loop.
    b. Change string from “getVal” to “setMot ” (include a space after setMot). Right click on the top terminal of Concatenate Strings and Add Terminal
    c. Wire the Visa resource name and error info the the Write VI:



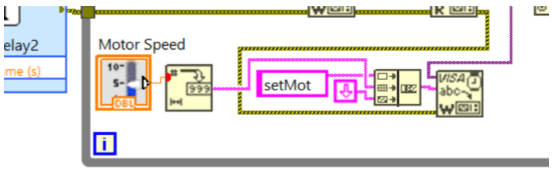
3. **Add Number to Decimal String VI**:
    a. **Diagram**: Right-click String>>Number/String Conversion>>Number to Decimal String.
    b. Wire output of Number to Decimal String VI to Concatenate Strings middle terminal.



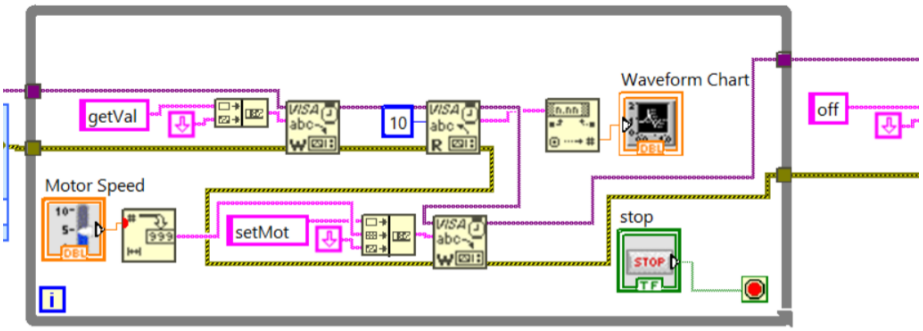
4. **Add Motor Speed Control**:
    a. **Front Panel**: Right-click Numeric>>Vertical Pointer Slide, set max to 255 and label it “Motor Speed.”



b. **Diagram**: Wire Motor Speed to Number to Decimal String VI.

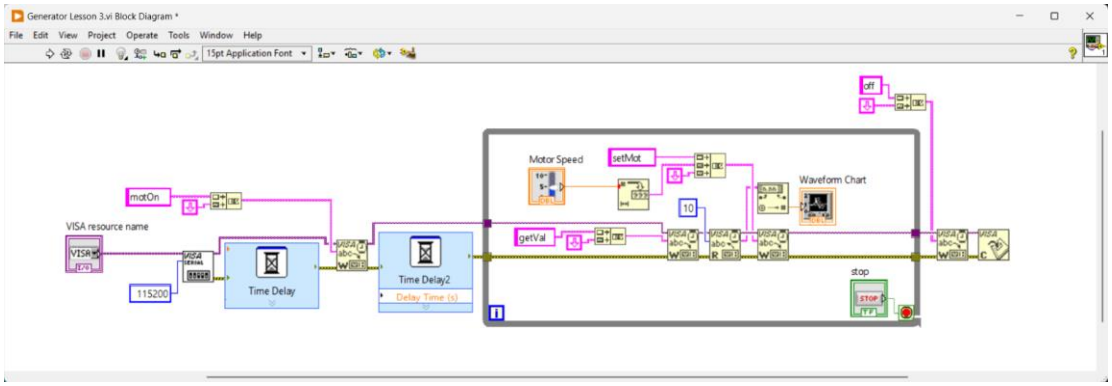


c. Wire the **Visa resource name** &**error** info of the **Visa Write** VI to the **While Loop** terminals that connect to the **Visa Write** VI that is outside the loop:



5. **Clean and Run**:
   a. Clean up diagram (Ctrl+U), run VI, and control motor speed with Motor Speed slider.

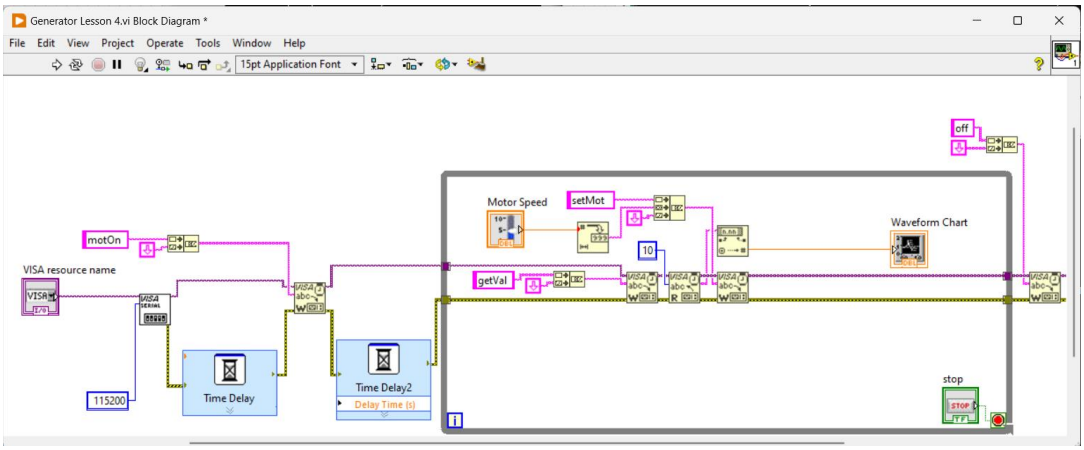
   b. The diagram looks like this:



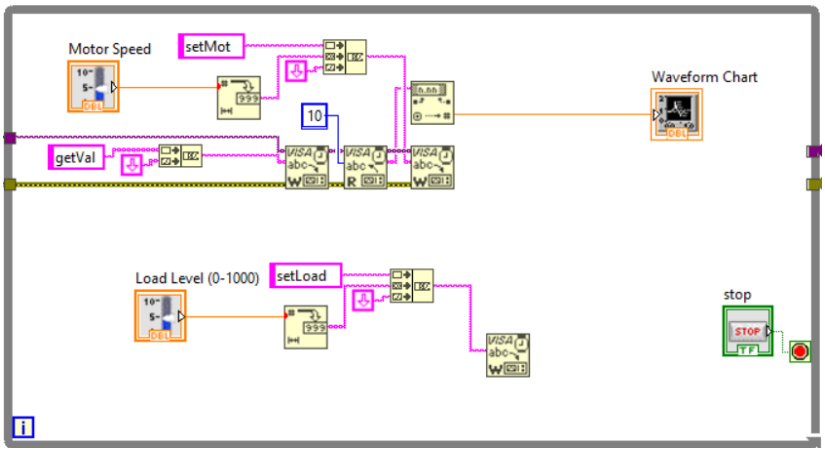
   c. Go to the front panel and run the VI.

Lesson 4a: Controlling Speed and Load

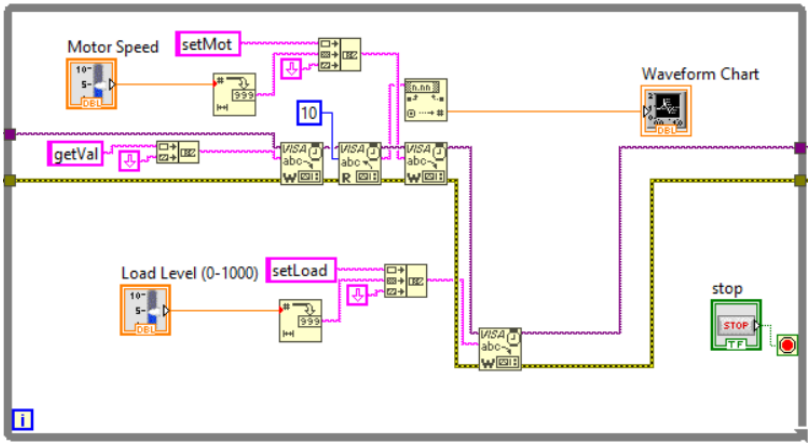
1. **Expand Diagram from Lesson 3**.
    a. Expand the diagram. Use **ctrl+left mouse click** and **stretch** to the bottom right



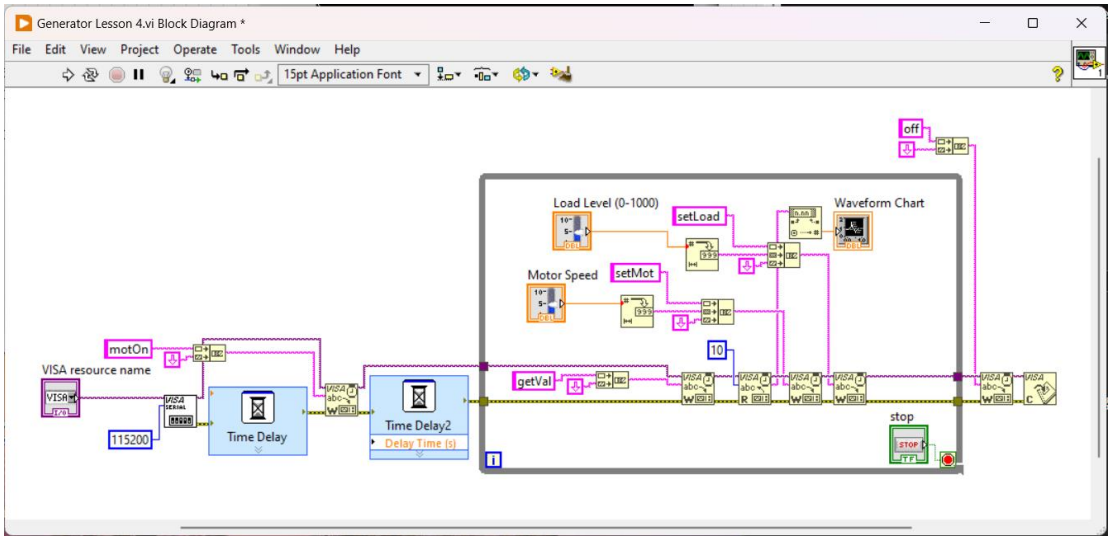
2. **Set Load Level**
    a. Copy the elements of group: **Motor Speed numeric control, Fract/Exp number to string, Line Feed Constant** and **Visa Write**
    b. Paste them into the **While Loop**
    c. Rename Motor Speed control to “Load Level.”
    d. Change string constant to “setLoad ” (include space).



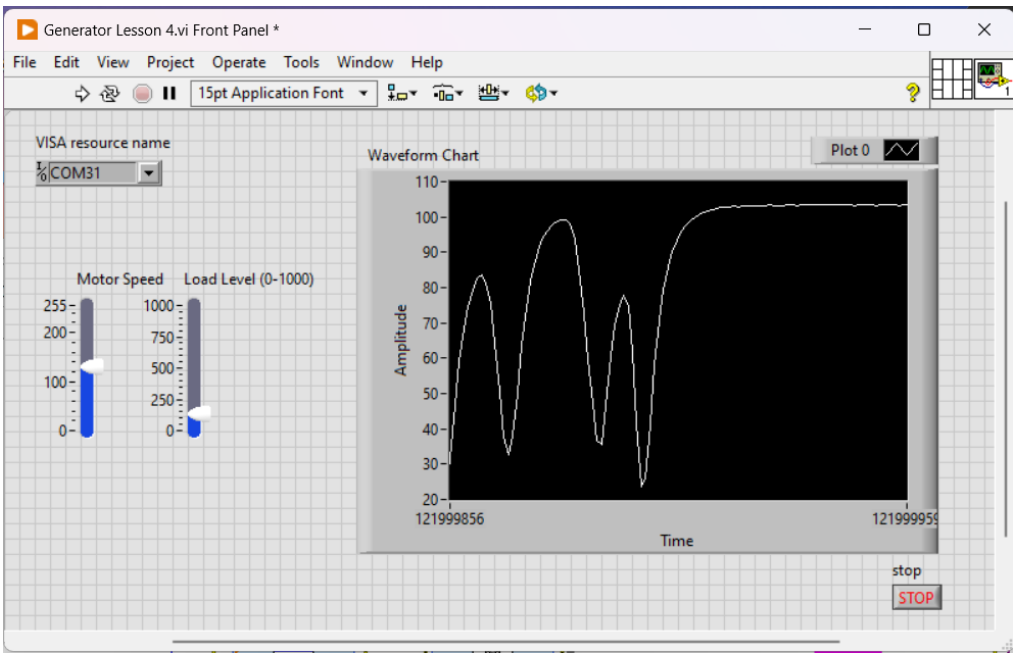
    e. Wire Visa resource name & error info to next Visa Write VI & While Loop terminals.



3. **Clean and Run**:
    a. Clean up VI, go to the front panel, set Load Level slide's max to 1000, and select com port.

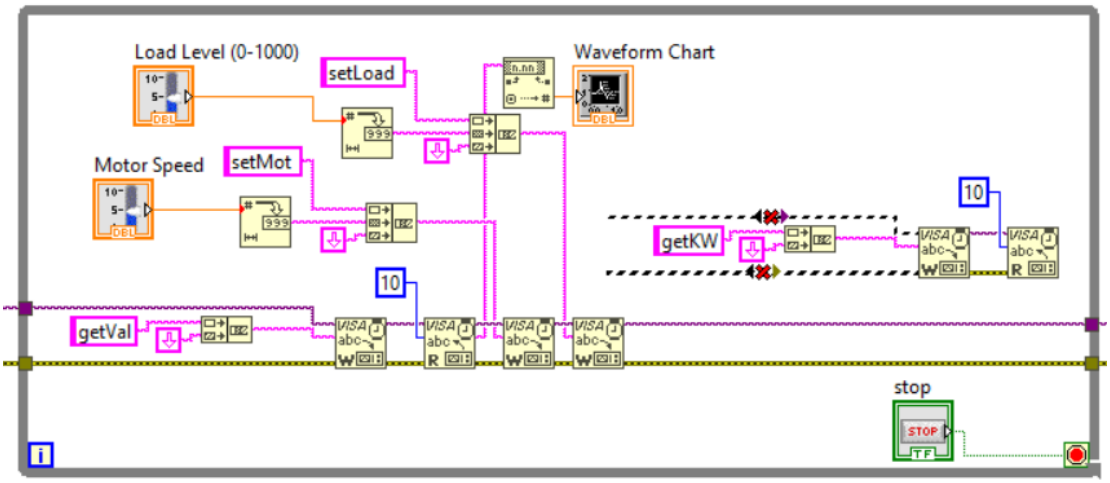


b. Select the com port for the Generator model by clicking the down arrow in the **Visa resource name** control
c. **Run the VI** by clicking the **run arrow**
d. Operate the Power Plant's drive motor and electronic load by adjusting the Motor Speed and the Load Level slides

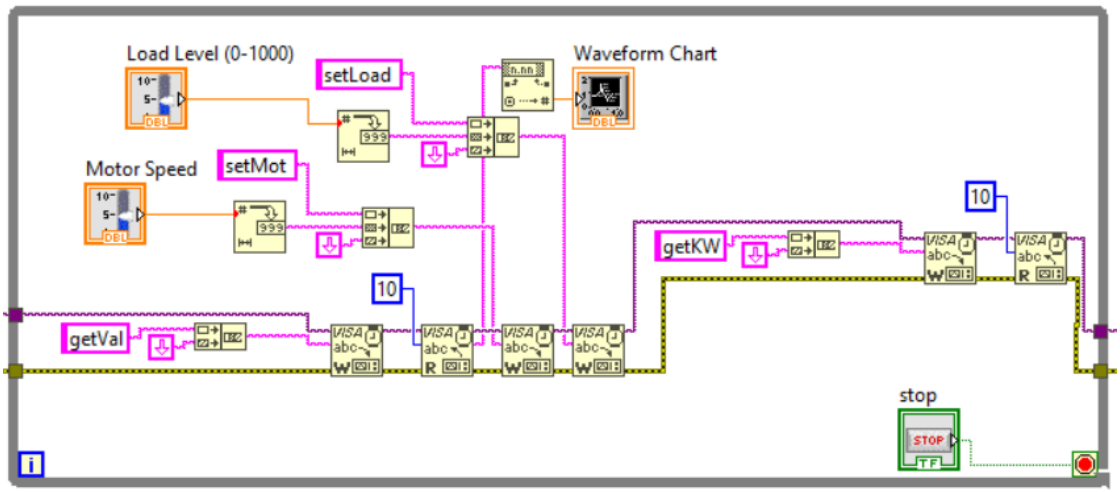


Lesson 4b: Read the kW Value

1. **Expand Diagram**:
    a. Copy “getVal” string constant, Line Feed Constant, Concatenate Strings, Visa Write, Visa Read, and “10” constant, paste in While Loop.
    b. Change “getVal” to “getKW.”

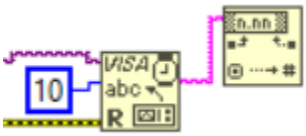


    c. Rewire the **Visa resource name** and **error** info to the **Visa Write** VI, then wire the same info from the **Visa Read** to the terminals on the **While Loop**:

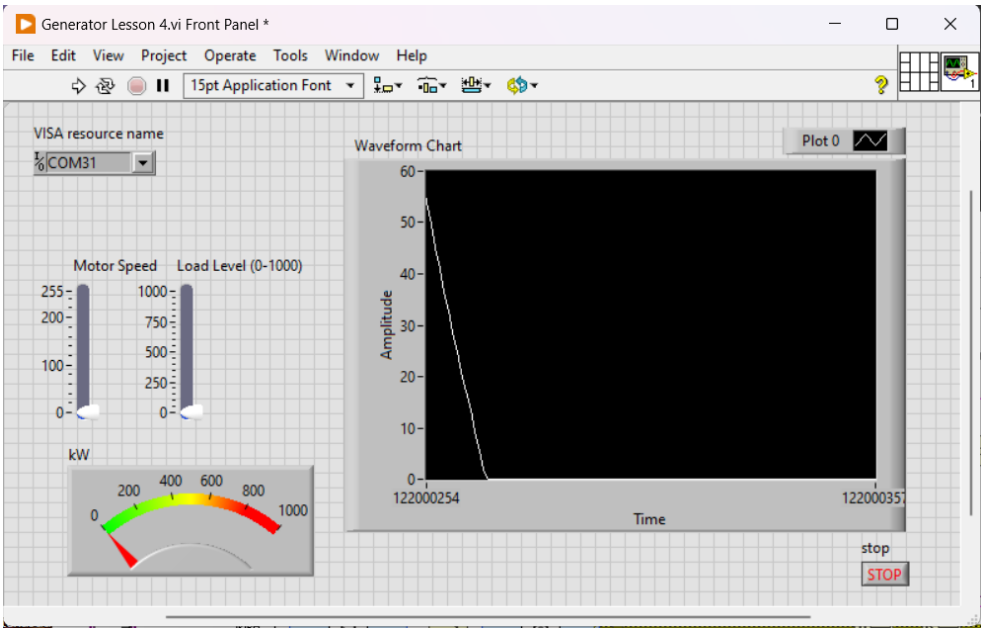


2. **Convert String to Number**:

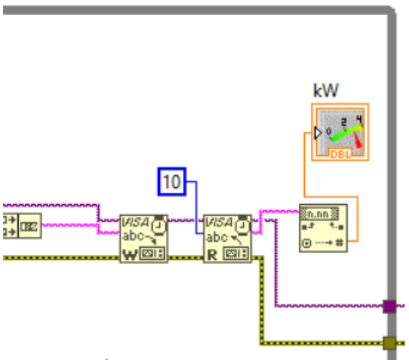
a. **Diagram**: Right-click String>>Number/String Conversion>>Fract/Exp String to Number.



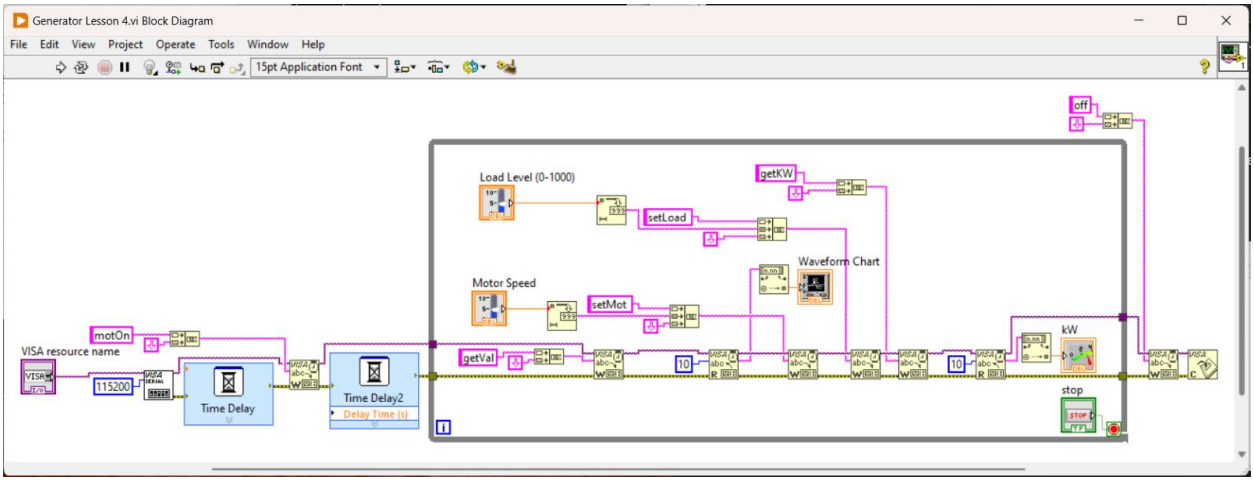
3. **Add Dispay**
   a. **Front Panel**: Right-click Numeric>>Meter, label “kW,” set range 0-1000.



4. Wire Meter
   a. Diagram: wire output of **Fract/Exp String to Number** to **meter**:

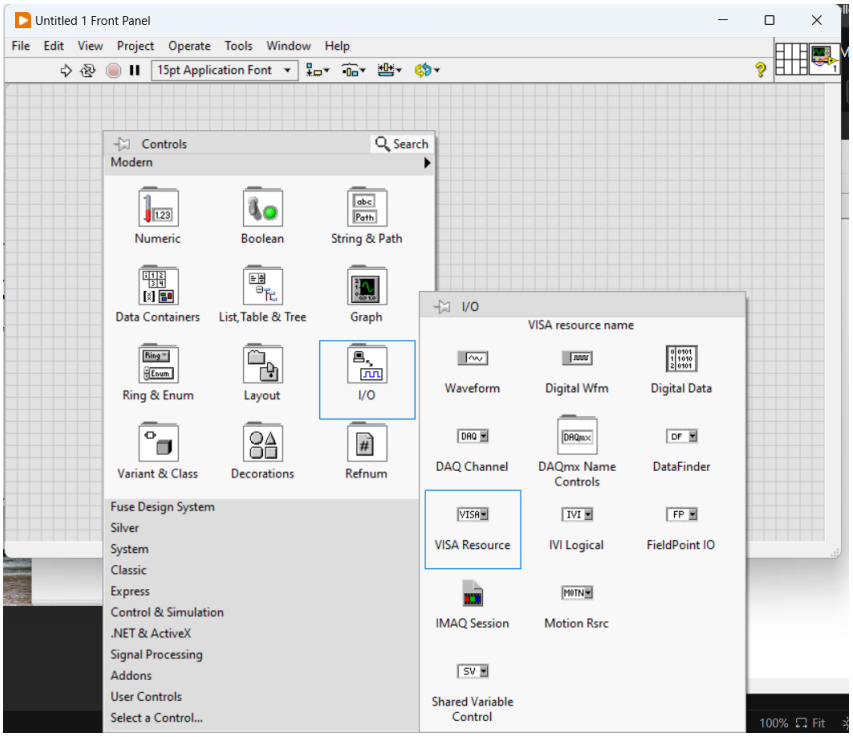


5. **Clean and Run**:
    a. Clean up VI, run the VI, adjust Motor Speed and Load Level, and observe kW output on Meter.

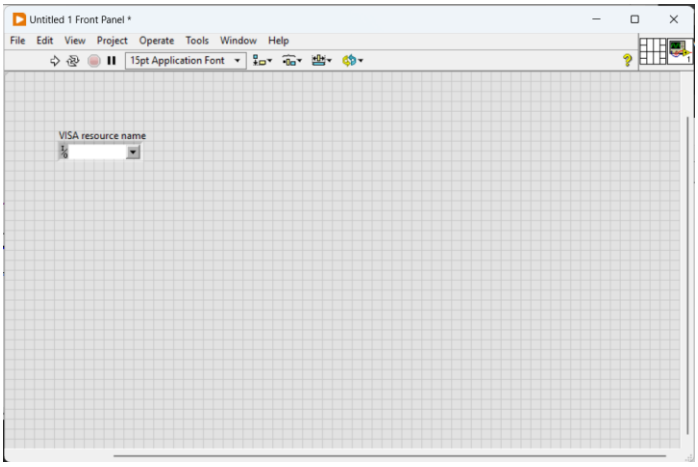


Lesson 5: PID Closed Loop Control Development

1. **Create a New VI**
    a. Go to the front panel and add a **Visa Resource Name Control**

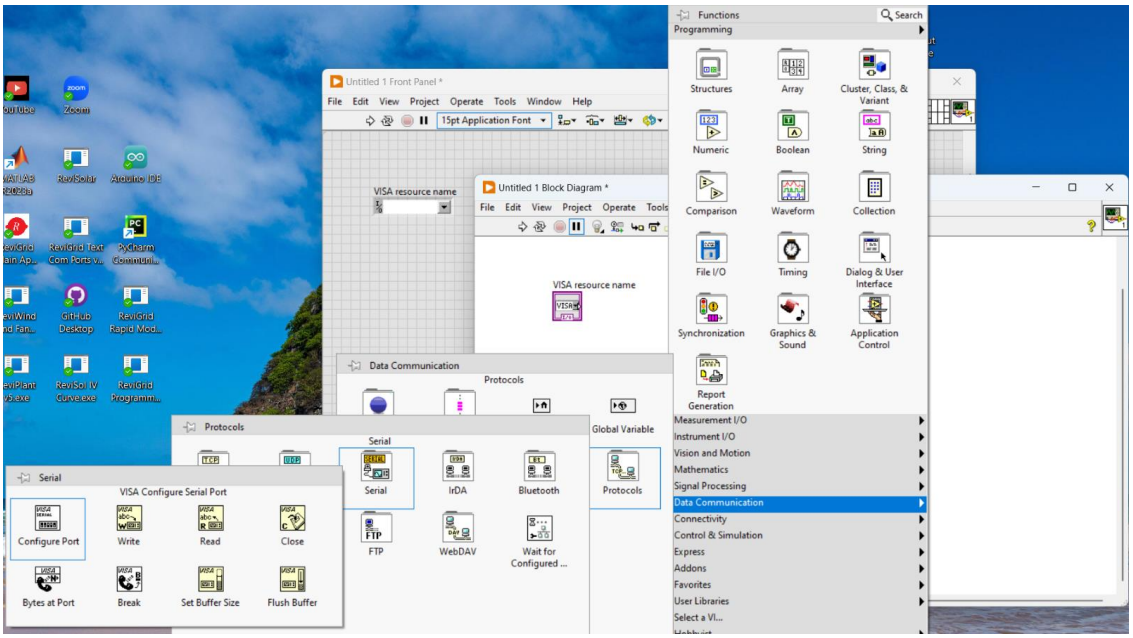


    b. The front panel looks like this:



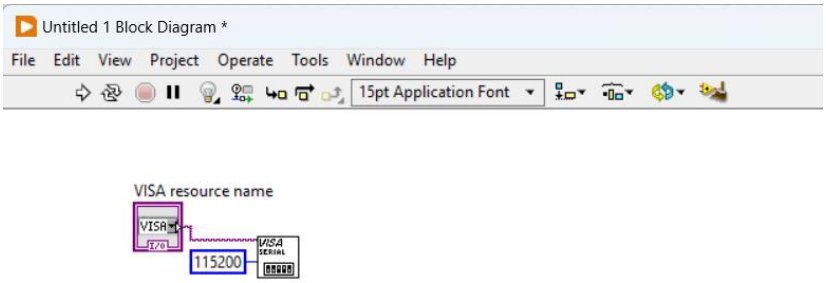
2. **Add a Visa Config Port VI**
    a. Go to the block diagram.

b. Right click: **Data Communication >> Protocols >> Serial >> Configure Port**



c. Wire the **Visa Resource Name Control** to the **Configure Port VI**.

d. Right click on the baud rate terminal and choose **Create** and **Constant**. Then type **115200** for the constant value:

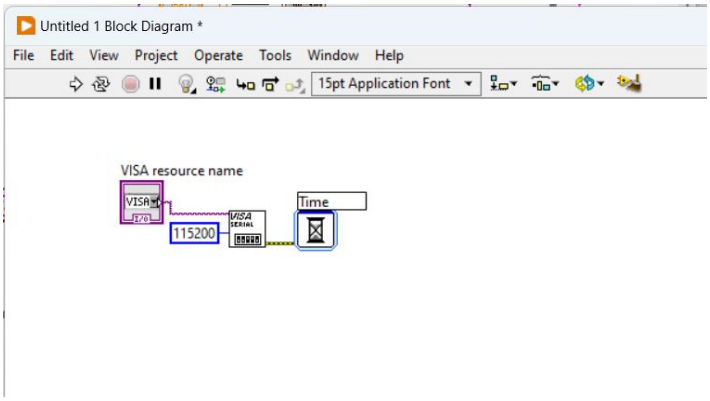


3. **Add a Time Delay VI**

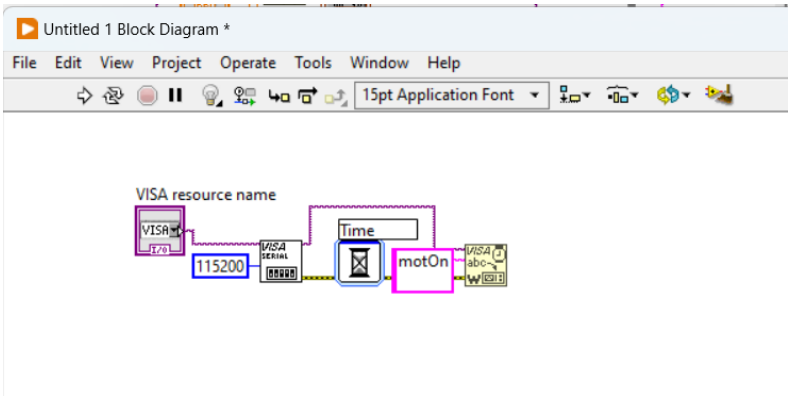
a. Right click: **Timing >> Time Delay**

b. The VI will automatically open a prompt box. Type in **2 seconds** for the delay value

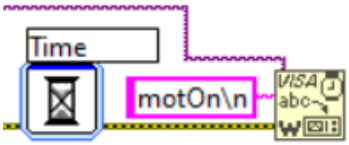
c. Next, click on the **Time Delay VI** and choose **View as Icon**
d. Wire the error info from the **Configure Port** to the **Time Delay VI**.



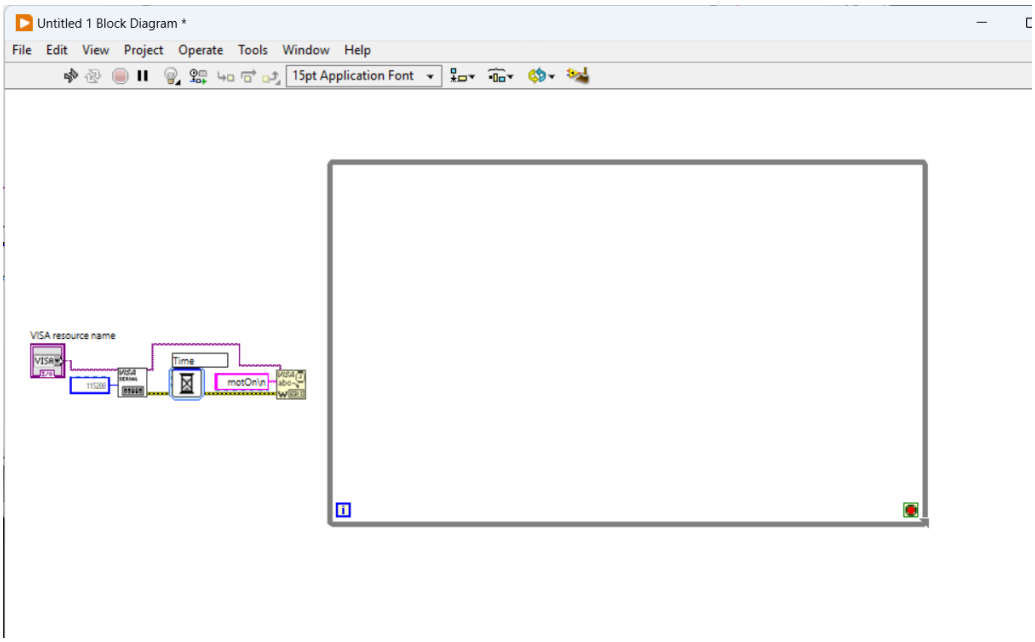
4. **Add Write VI and Command String a**dd a **Write VI**
   a. Right click: **Data Communications >> Protocols >> Serial >> Write**
   b. Wire error info from **Time Delay** and the **Visa Resource Name** to the **Write VI**
   c. Right click on write buffer terminal of **Write VI** and select **Create >> Constant**
   d. Type **motOn** and **Shift+Enter**



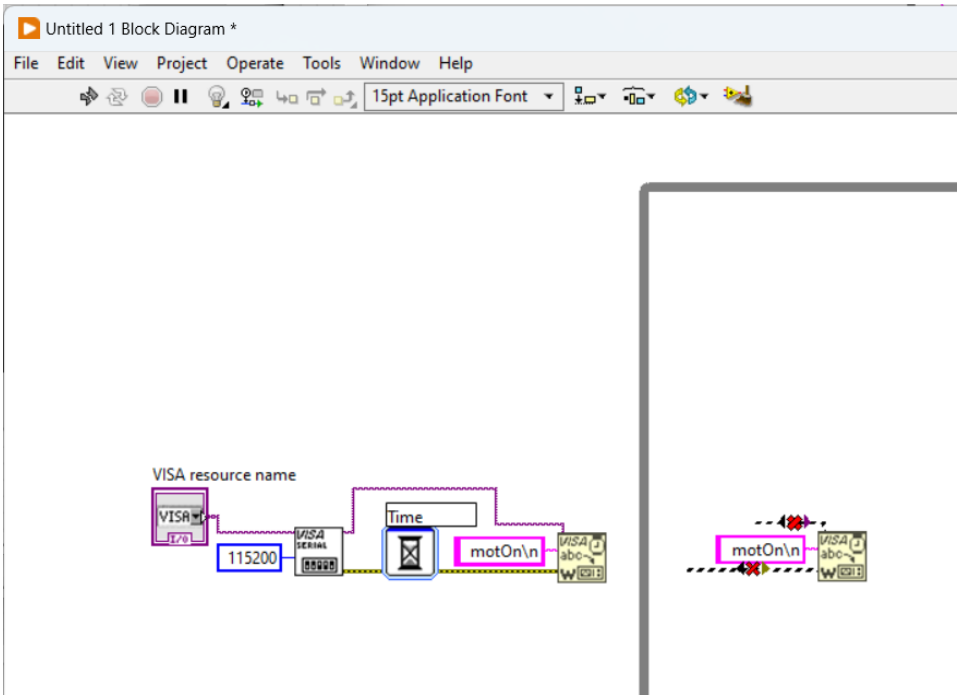
5. **Enable special character codes**
    a. Right click on the **“motOn”** string constant and select **‘\’ Codes Display**
    b. This will display special characters, like \n (Line Feed)



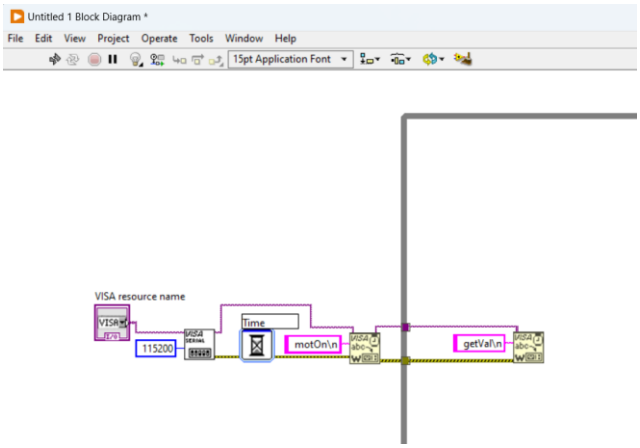
6. **Add a While Loop**
    a. Right click: **Structures >> While Loop**



7. **Copy Write in loop**
    a. Copy **Write VI** & **motOn\n** string constant group &paste inside loop



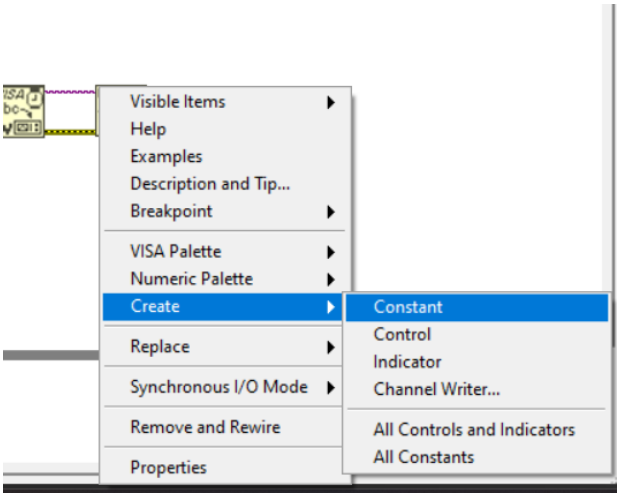
8. **Wire connections in the loop and Modify command**
    a. Wire **Visa Resource Name** & error info from **first Write VI** to the new Write VI
    b. Change string from motOn\n to **getVal\n**
    c. Wire **Visa resource name** and **error** info from the first **Write** to this **Write** VI:

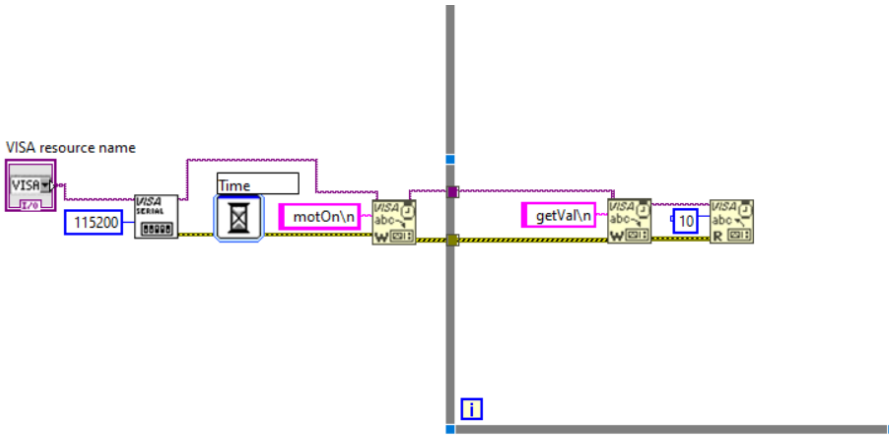


9. Add a **Visa Read VI**
    a. Right click: **Data Communication >> Protocols >> Serial >> Read**

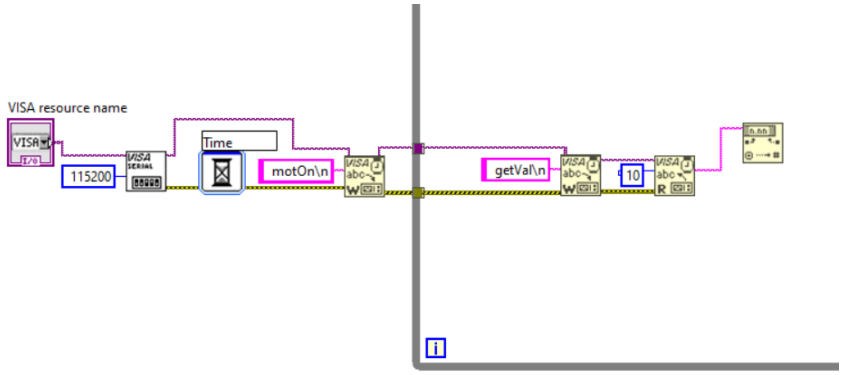
b. Wire the **Visa Resource Name** and error info into the **Read VI**

c. Right click on byte count terminal, select **Create >> Constant**, and type **10**

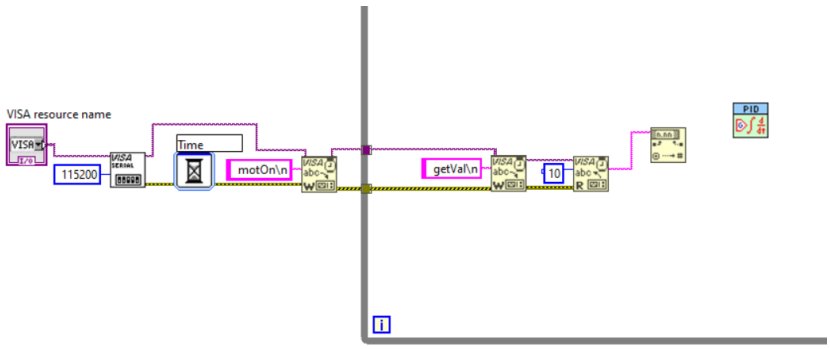




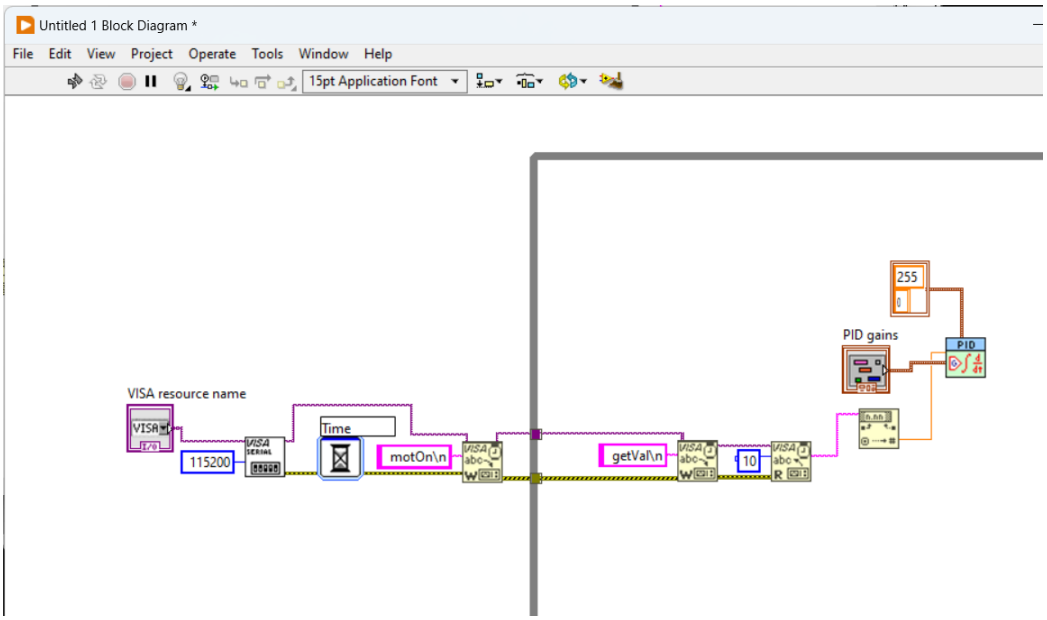
10. Add **Fract/Exp String to Number.vi**
    a. a. Right click: **String >> Number/String Conversion >> Fract/Exp String to Number**
    b. Wire **Read Buffer** output to the input of the **Fract/Exp String to Number VI**



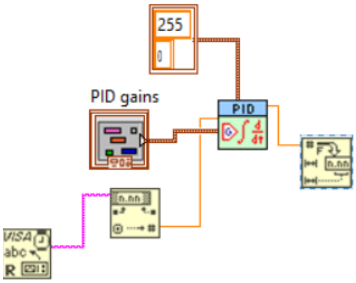
11. **Add PID.vi**
    a. a. Right click: **Control & Simulation >> PID >> PID.vi**



12. **Set the PID parameters**
    a. Right click on the **Output Range** terminal and set values to **0** and **255**
    b. Right click on the **PID Gains** terminal and select **Create >> Contro**
    c. Wire the **Fract/Exp String to Number VI** output to the **Process Variable** input of the **PID.vi**

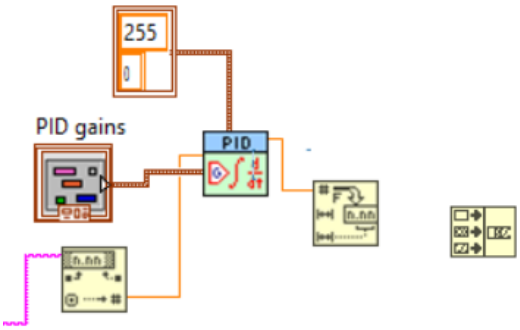


13. **Add Number to Fractional String.vi**
    a. Right click: **String >> String Conversion >> Number to Fractional String.vi**
    b. Wire the output of **PID.vi** to the **# input terminal of Number to Fractional String.vi**

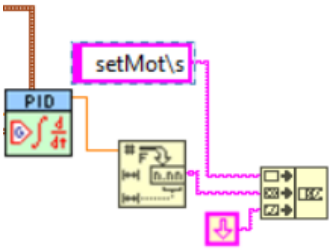


14. **Add Concatenate Strings.vi**
    a. Right click: **Strings >> Concatenate Strings.vi**

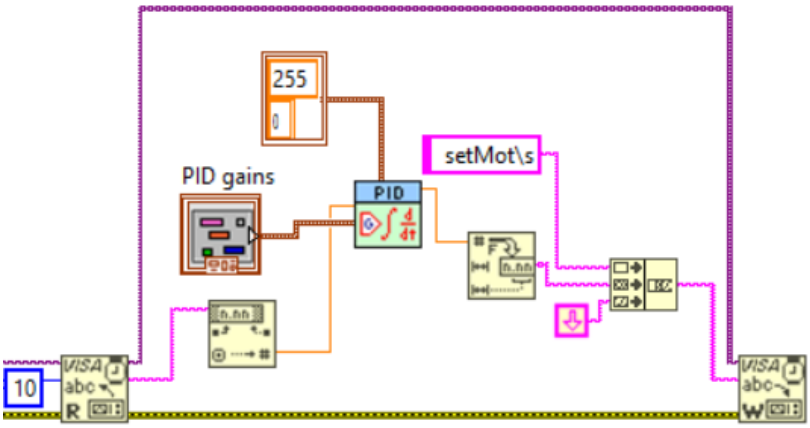
b. Left click to stretch down and add an additional input



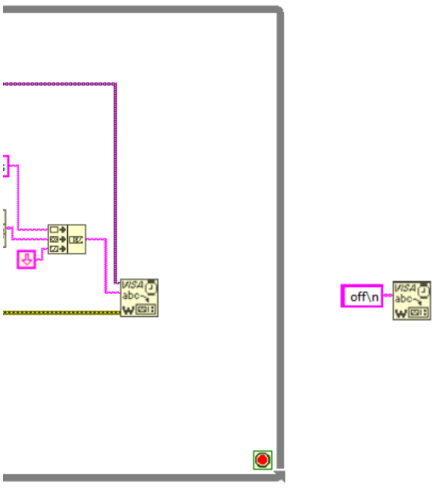
15. **Set the command string**
    a. Right click on the **top left terminal** of **Concatenate Strings.vi**, choose **Create >> Constant**
    b. Type: **setMot** (with a trailing space)
    c. Right click on the string constant and select **'\' Codes Display**
    d. Wire the output of **Number to Fractional String.vi** to the center terminal and **Line Feed Constant** to the bottom terminal



16. **Add a final Write VI**
    a. Right click: **Data Communication >> Protocols >> Serial >> Write.vi**
    b. Wire the **Concatenate Strings.vi** output and the **Visa Resource Name** and error info from the previous **Read VI**



17. **Copy Write and modify outside loop**
    a. Copy **getVal String** & **Write VI** group, paste outside loop, & change string to **off\n**

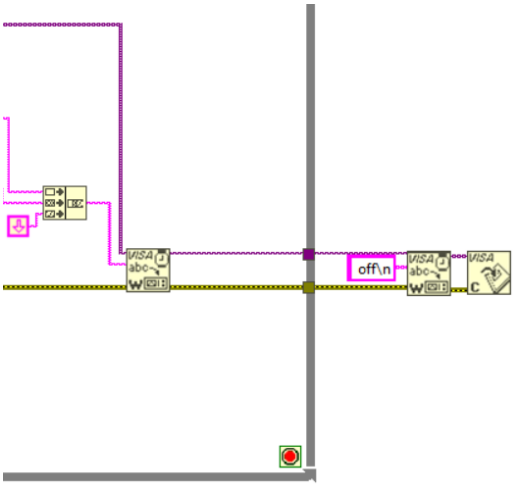


18. **Wire to Write and add Visa Close VI**

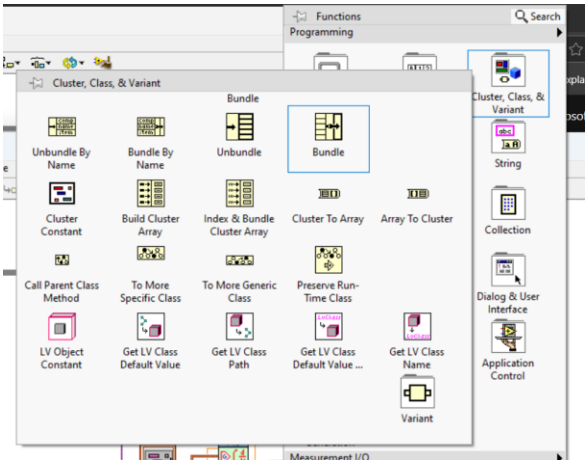
a. Wire **Visa Resource Name** & error info to input terminals of **Write VI**

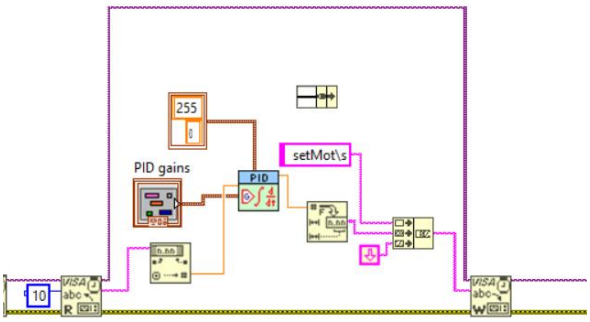
b. Right click: **Data Communications >> Protocols >> Serial >> Close.vi**

c. Wire the **Visa Resource Name** and error info to **Close.vi**

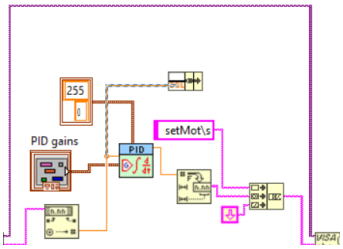


19. **Add a Bundle**
    a. Right click: **Cluster, Class & Variant >> Bundle**
    b. Wire the input value to the **PID.vi Process Variable** into the bottom terminal of the **Bundle**

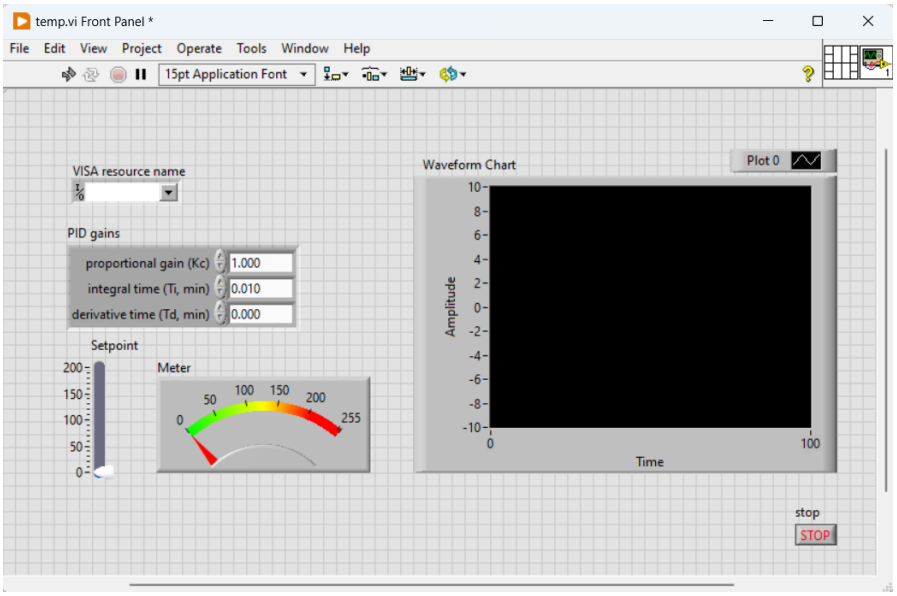




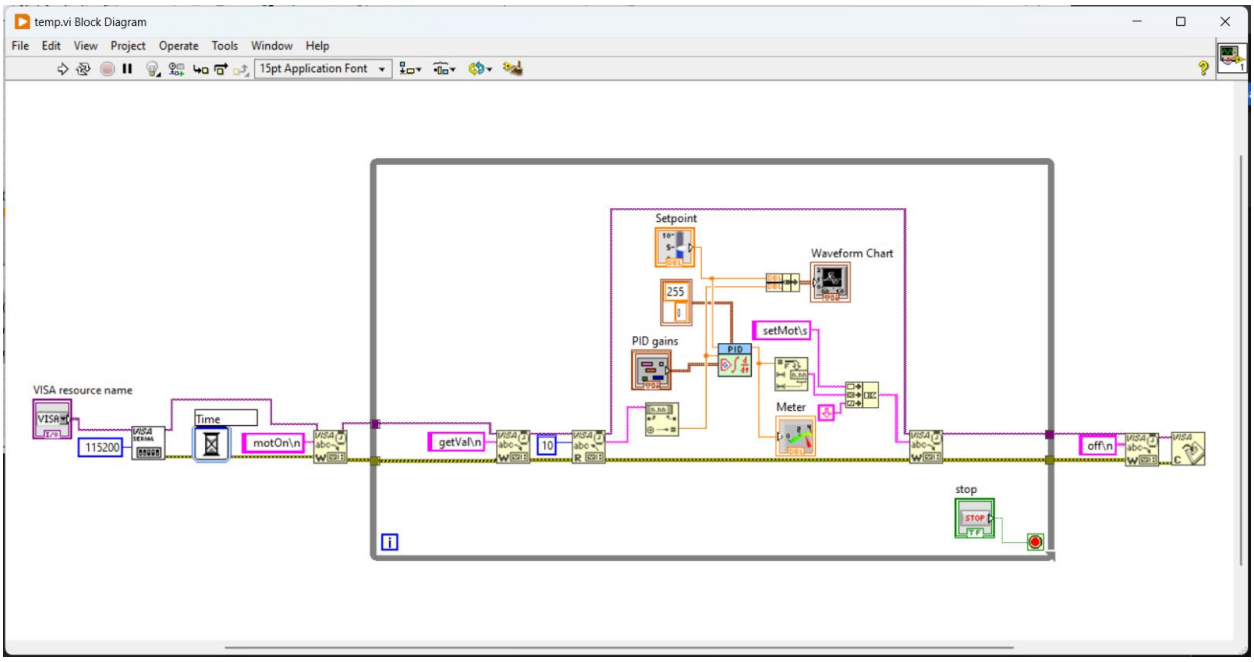
20. Wire the Values
    a. Wire the value from the input to the **PID**.vi **process variable** into the bottom terminal of the **Bundle**:



21. Go to the front panel and add the following controls and indicators:
    a. **Vertical Pointer Slide**: Right click: **Numeric >> Vertical Pointer Slide** (name it **SetPoint** and set max to **200**)
    b. **Waveform Chart**: Right click: **Graph >> Waveform Chart**
    c. **Numeric Meter**: Right click: **Numeric >> Meter** (name it **Output** and set max to **255**)
    d. **Stop Button**: Right click: **Boolean >> Stop Button**



22. Go to the diagram and make the following connections:
    a. Wire the **Setpoint Control** to the top terminal of **Bundle** and the **Setpoint terminal of PID.vi**
    b. Wire the **Bundle output** to the **Waveform Chart**
    c. Wire the **PID.vi output** to the **Meter Indicator**
    d. Wire the **Stop Button** to the **While Loop** conditional terminal



Run the VI

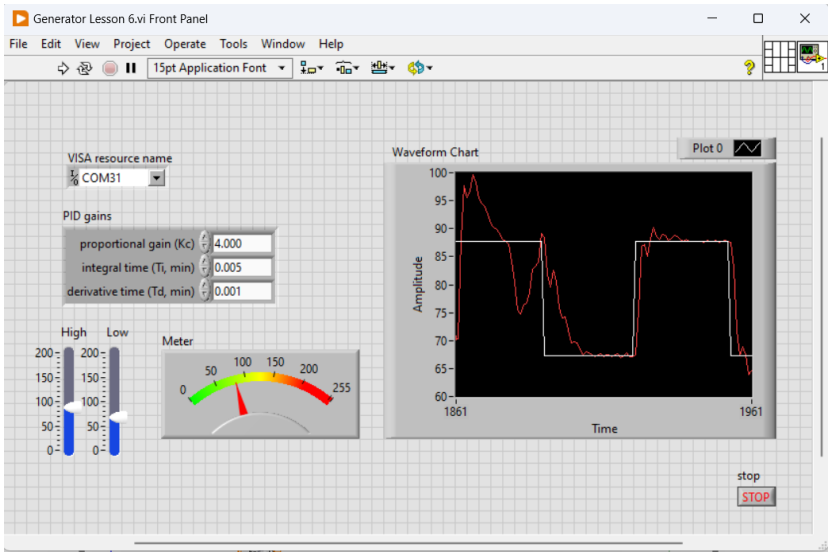
23. Go to the **front panel** and run the VI. Adjust the **Setpoint** and **PID coefficients** to observe the generator's response.

End of Lesson 5

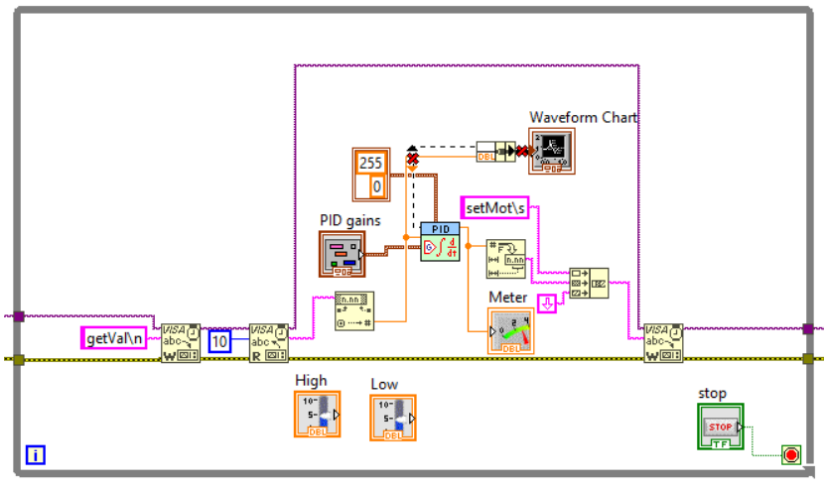
Lesson 6: PID Closed Loop Control Tuning

Start with the VI from Lesson 5

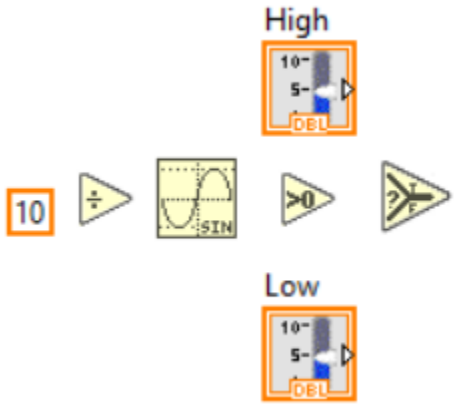
1. Add another vertical slide and rename them
    a. Go to the front panel and copy the vertical slide and paste the copy next to it
    b. Rename the Slides “High” and “Low”



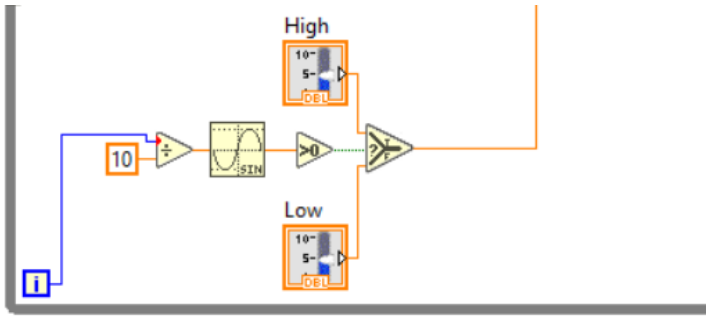
    c. Go to the diagram and delete the wire from the High slider



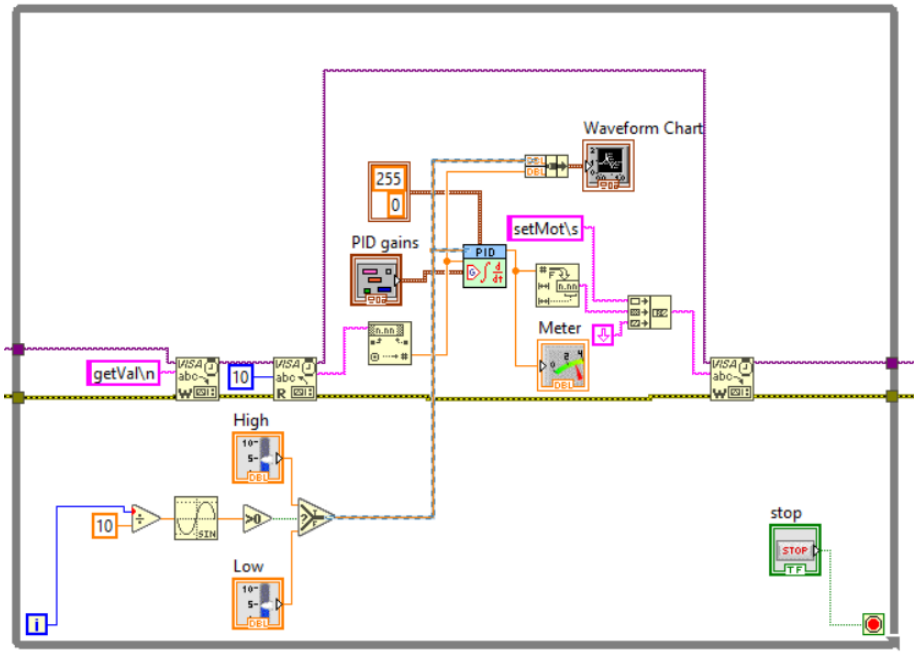
2. Add the logic to toggle between the High and Low slider values
    a. Go to the Diagram and divide, numeric constant, sin, greater than 0, & Select
    b. Right click: Numeric>>Constant (type 10)
    c. Right click: Numeric>>Divide
    d. Right click: Comparison>>Greater than 0
    e. Right click: Mathematics>>Elementary>>Trigonometry>>sin
    f. Right click: Comparison>>Select



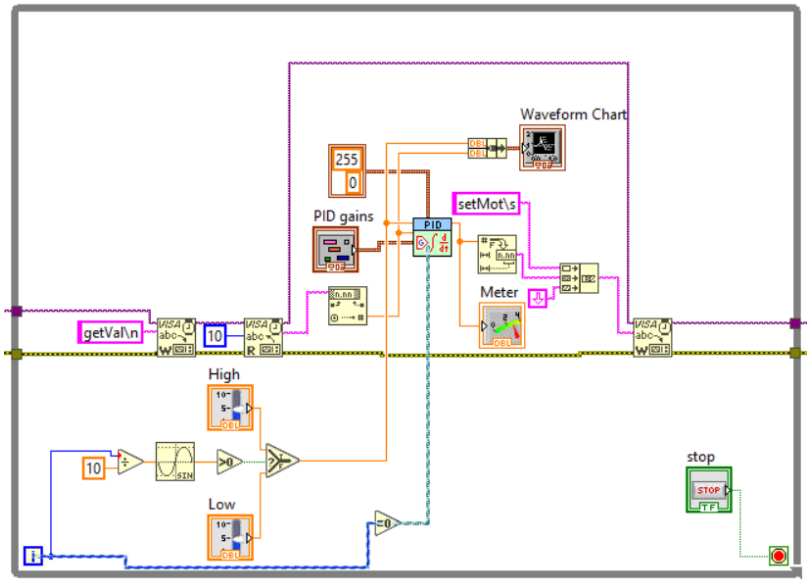
3. Wire the toggle logic
    a. Wire the loop counter to the top of the divide
    b. Wire the 10 constant to the bottom of the divide
    c. Wire the output of divide to sin
    d. Wire sin to Greater than 0
    e. Wire Greater than 0 output to the boolean input of select
    f. Wire the “High” slider to the top of the select
    g. Wire the “Low” slider to the bottom of the select



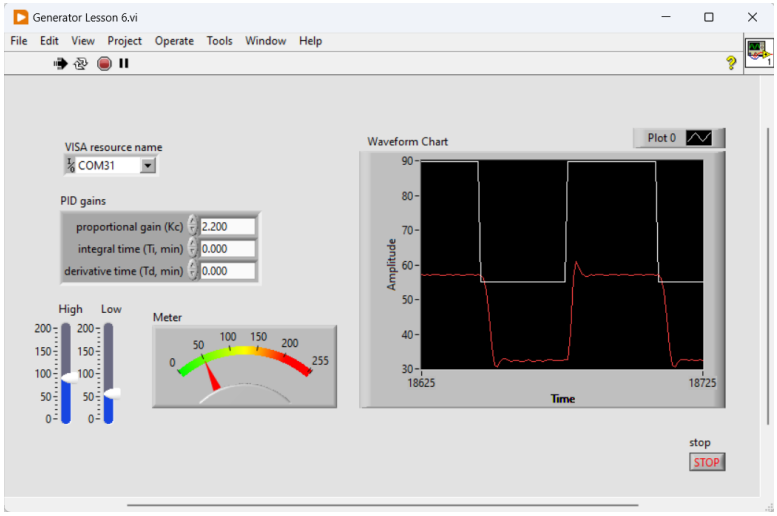
h. Wire the output of Select to the PID setpoint and Bundle input terminals



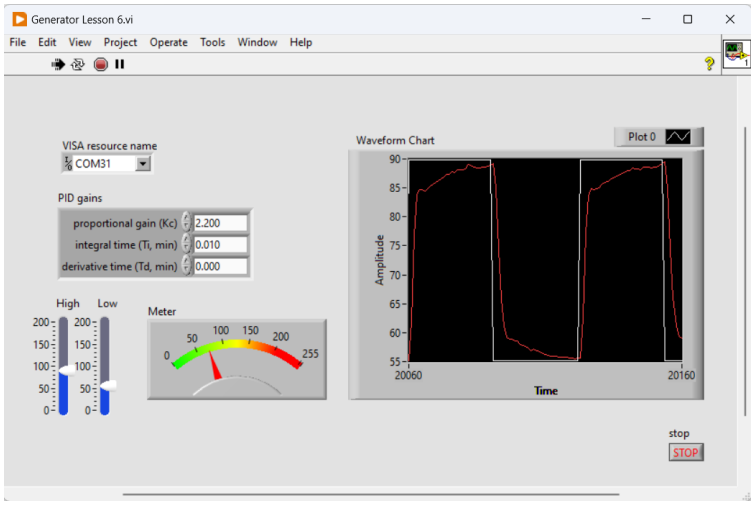
4. Wire the PID reset
    a. Add Equal to 0 to the diagram
    b. Right click: Comparison>>Equal to 0
    c. Wire from the Loop index to the Equal to 0
    d. Wire the output of Equal to 0 to tine reset terminal of the PID VI



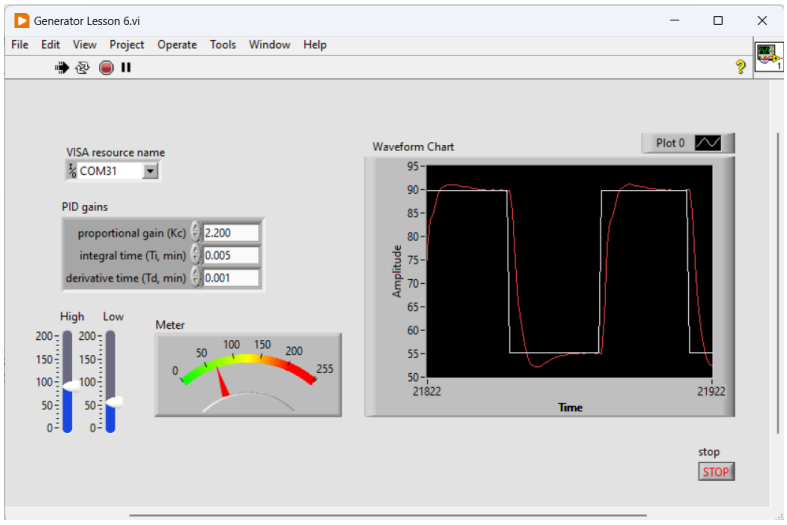
5. Run the VI and Tune
    a. Go to the front panel
    b. Select the Com port assigned to the power plant model
6. **Set Initial Conditions:**
    a. Begin with **P = 0**, **I = 0**, and **D = 0**.
    b.
7. **Adjust Proportional (P) Gain:**
    a. Gradually **increase the P Gain** from 0, observing the system's response for quick responsiveness without excessive oscillations.
    b. **Goal for P Gain:** Adjust until you achieve a fast response without instability. If oscillations become too high, reduce the P gain slightly.



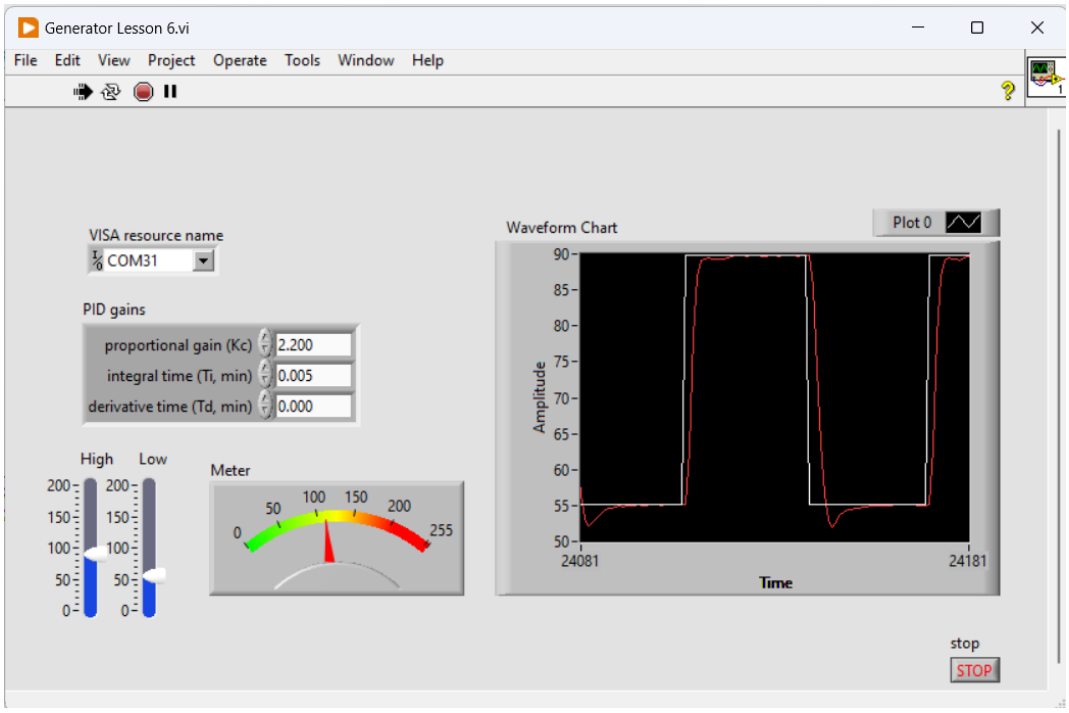
8. **Fine-Tune Integral (I) Gain (Starting from 1):**
    a. With I initially set to 1, **decrease it if necessary** to increase the integral effect.
    b. Lowering the I gain value enhances its power to reduce steady-state error.
    c. **Goal for I Gain:** Find a balance where the I term eliminates steady-state error without causing instability or overshoot. If oscillations occur, slightly increase the I gain (closer to 1).



9. **Set Initial Derivative (D) Gain to 0.01:**
    a. Change **D from 0 to 0.01** to start introducing the derivative effect.
    b. **Observe** the system's response. If oscillations occur immediately, this starting value can be reduced slightly.



10. **Gradually Decrease D Gain for Damping:**
    a. If D at 0.01 introduces oscillations, **decrease D slightly** to add damping without making the response overly sluggish.
    b. **Goal for D Gain:** Set it just low enough to dampen oscillations and stabilize the system response.
11. **Fine-Tuning and Testing:**
    a. Make incremental adjustments to each gain for optimal performance.
    b. Use the **Waveform Chart** to ensure a smooth response with minimal overshoot and steady-state error.
12. **Run Final Tests:**
    a. Test the system under various conditions to confirm it remains stable and responsive.
13. This sequence, with D starting at 0.01, should allow for effective damping without excessive initial impact on system stability.



End of Lesson 6