Instrumental Analysis laboratory
Chem 342/790.1, Spring 2009
Queens College, CUNY

Experiment 1: Using LabVIEW and DAQ to Build a Temperature Sensor

The lab will use insert-module approach in which students develop the LabVIEW code for temperature programming and insert it into a prebuilt VI template. This experiment involves measuring temperature using a thermocouple, and using feed back to control temperature ramp speed and ensure the actual temperature matches the desired value. Various tasks are involved in the VI, such as sampling analog signals, conversion of analog signal to a digital code, feedback control algorithm, digital signal output for operating a solid-state relay to switch the heater, plotting temperature vs. time, and date storage. Once the VI template is completed, students are required to expand the VI so that different types of thermocouples can be utilized and various temperature programming modes can be realized. This lab mimics the temperature programming in commercial gas-chromatographic instruments for retention time control. In this lab, students will obtain basic skills in LabVIEW environment navigation, application creation, data acquisition, and instrument control.

Thermocouple: A thermocouple is a device whose potential varies with temperature. A variety of thermocouples are available, suitable for different measuring applications. They are usually selected based on the temperature range and sensitivity needed. Thermocouples are most suitable for measuring over a large temperature range, up to 1800 °C. Type K (chromel–alumel) thermocouple is the most commonly used one for general purpose. The temperature range for Type K is from −200 °C to +1350 °C.

Purpose

1) Understanding of basics of LabVIEW virtual instruments (VIs) and Data Acquisition Devices (DAQ)
2) Create a LabVIEW subVI to measure the temperature using K thermocouple (i.e., convert the thermoelectric voltage to temperature reading).
3) Using temperature feed back to control temperature ramp speed and ensure the actual temperature matches the desired value.

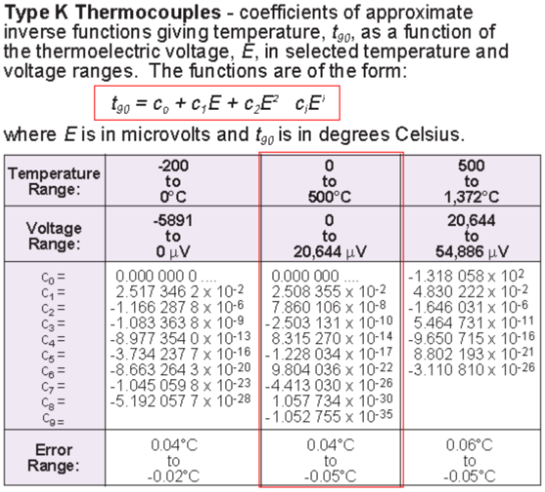
Equipment

Windows computer with LabVIEW software
National Instrument PCI 6229 DAQmax card and connector box
Thermocouple (type K)
Beaker with cold water
Stirrer
Solid State Relay
Jumper wires
Heating ring

Procedures

Part I. Build a SubVI to Convert Thermoelectric Voltage to Temperature

You will build a LabVIEW VI that will convert a raw voltage signal generated from thermocouple to a temperature reading. This VI will be placed inside of a top-level LabVIEW VI and used as a subVI in Step 2.
In the temperature measurement, the thermocouple will generator thermoelectric voltage cross the junction, and the following an inverse polynomial converts the thermoelectric voltage (V) to the temperature (°C).



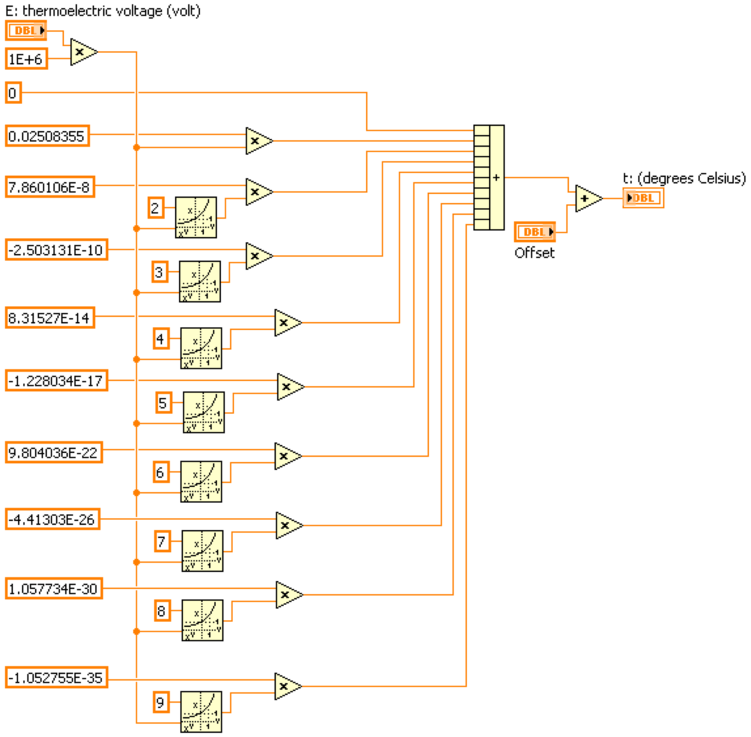
Type K Thermocouples - coefficients of approximate inverse functions giving temperature, t_{90}, as a function of the thermoelectric voltage, E, in selected temperature and voltage ranges. The functions are of the form:

$$t_{90} = c_0 + c_1 E + c_2 E^2 \quad c_i E^i$$

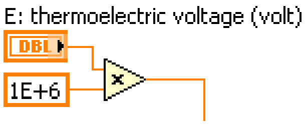
where E is in microvolts and t_{90} is in degrees Celsius.

Temperature Range:	-200 to 0°C	0 to 500°C	500 to 1,372°C
Voltage Range:	-5891 to 0 µV	0 to 20,644 µV	20,644 to 54,886 µV
c_0 =	0.000 000 0	0.000 000	$-1.318\,058 \times 10^{2}$
c_1 =	$2.517\,346\,2 \times 10^{-2}$	$2.508\,355 \times 10^{-2}$	$4.830\,222 \times 10^{-2}$
c_2 =	$-1.166\,287\,8 \times 10^{-6}$	$7.860\,106 \times 10^{-8}$	$-1.646\,031 \times 10^{-6}$
c_3 =	$-1.083\,363\,8 \times 10^{-9}$	$-2.503\,131 \times 10^{-10}$	$5.464\,731 \times 10^{-11}$
c_4 =	$-8.977\,354\,0 \times 10^{-13}$	$8.315\,270 \times 10^{-14}$	$-9.650\,715 \times 10^{-16}$
c_5 =	$-3.734\,237\,7 \times 10^{-16}$	$-1.228\,034 \times 10^{-17}$	$8.802\,193 \times 10^{-21}$
c_6 =	$-8.663\,264\,3 \times 10^{-20}$	$9.804\,036 \times 10^{-22}$	$-3.110\,810 \times 10^{-26}$
c_7 =	$-1.045\,059\,8 \times 10^{-23}$	$-4.413\,030 \times 10^{-26}$	
c_8 =	$-5.192\,057\,7 \times 10^{-28}$	$1.057\,734 \times 10^{-30}$	
c_9 =		$-1.052\,755 \times 10^{-35}$	
Error Range:	0.04°C to -0.02°C	0.04°C to -0.05°C	0.06°C to -0.05°C

The finalized VI will look like the figure below:

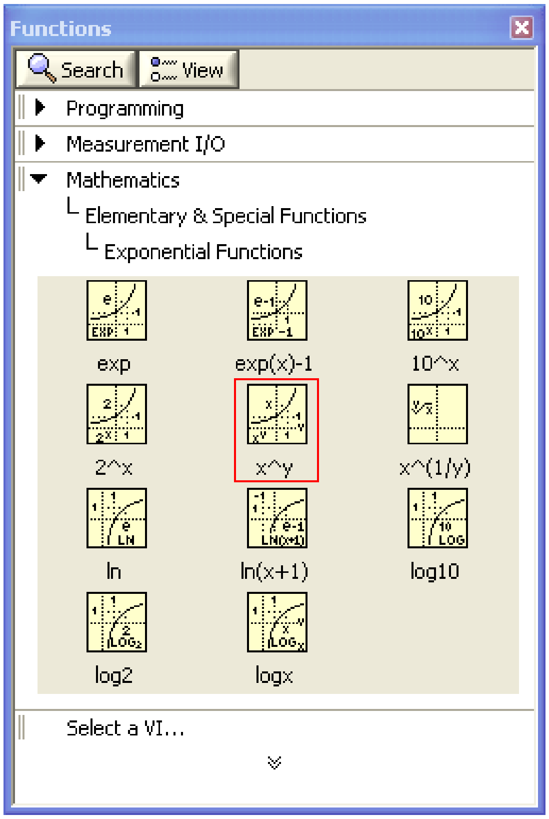


1. Open a new, blank VI
2. The first calculation converts the **voltage** reading to the **microvolt**. To perform the equation use the Multiply function found in the Numeric palette.

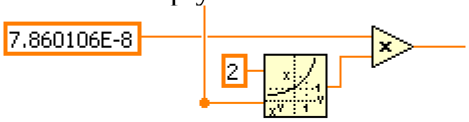


a. Place the Multiply function on the Block Diagram.
b. Right Click on the lower input node and select Create>Constant.
c. Make this value 1E+6 (the conversion ratio)
d. Right click on the upper input node and select Create>Control. Give this control a label of "E: thermoelectric voltage (volt)".

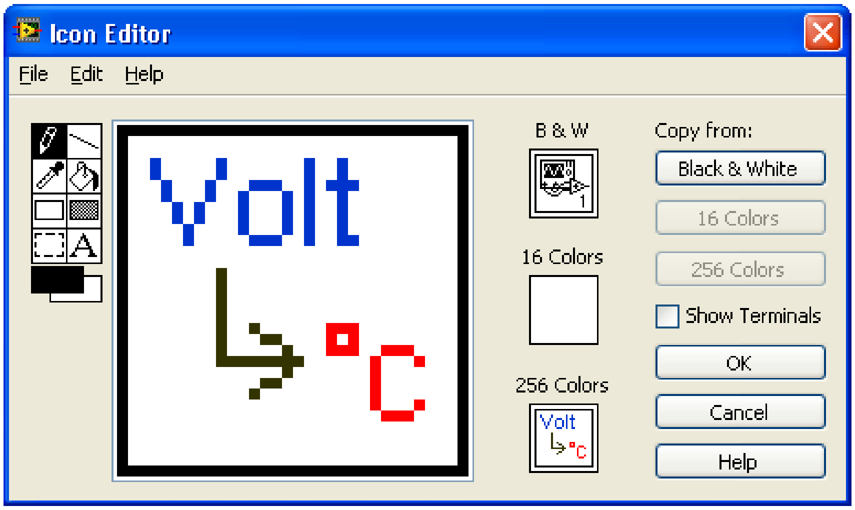
3. Place a "Power of X" function on the block diagram.



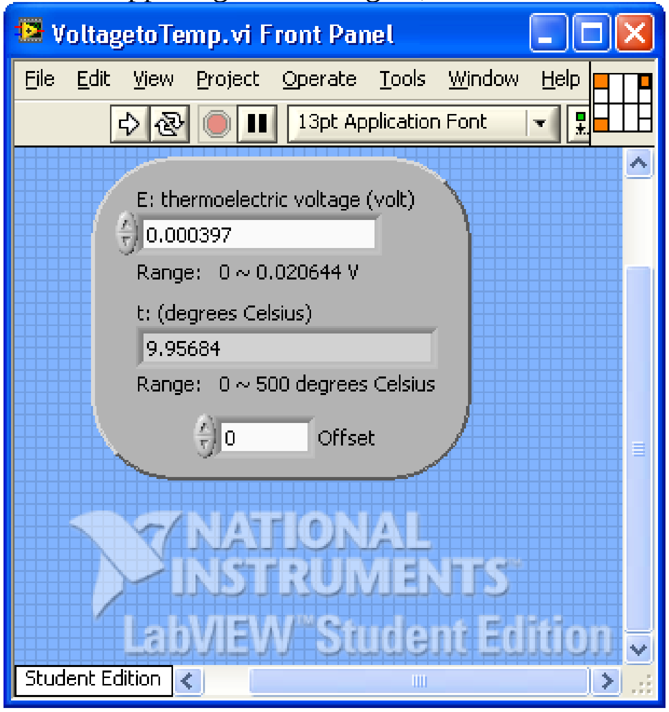
4. Place the Multiply function on the Block Diagram and wire them as illustrated.



5. Finish the rest part in the similar way.
6. Build the icon and connector pane so that you can use this VI as a subVI.



a. Go to the *front panel* and right click the icon in the upper right corner and select Edit Icon. The icon editor dialog box appears.
b. Use the Icon Editor tools to draw an icon or write descriptive text (like the icon above).
c. When the icon is completed, click the OK button.
d. Right click the icon in the upper right corner again, and select Show Connector.



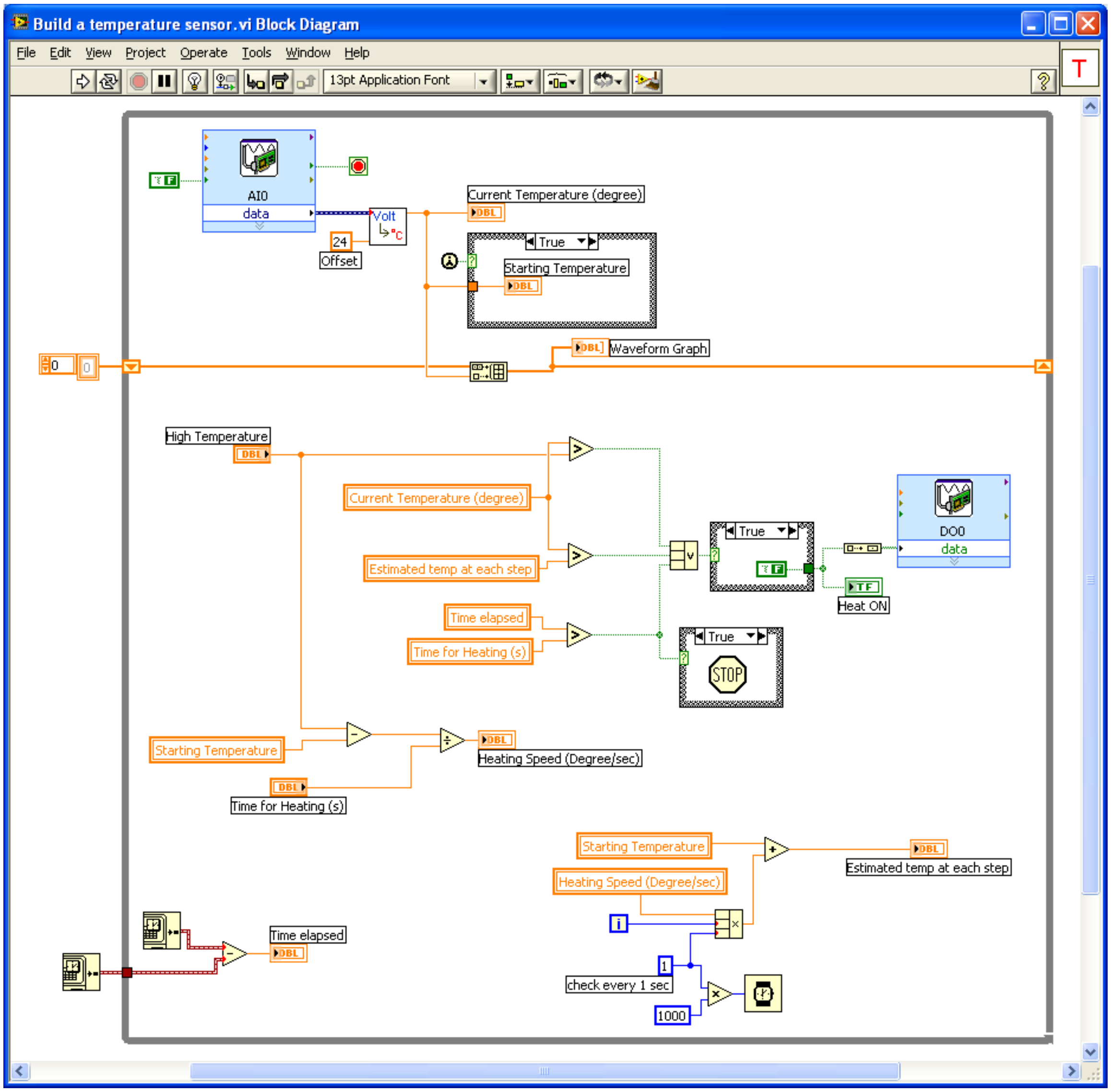
E: thermoelectric voltage (volt)
0.000397
Range: 0 ~ 0.020644 V
t: (degrees Celsius)
9.95684
Range: 0 ~ 500 degrees Celsius
0 Offset

e. Click the top left terminal and then bring your mouse into the front panel and click on the "E: thermoelectric voltage (volt)" control. The top left terminal should turn orange.
f. Click the bottom left terminal and then bring your mouse into the front panel and click on the "offset" control. The bottom left terminal should turn orange.
g. Click the top right terminal and then click on the "t: (degrees Celsius)" indicator. The top right terminal should turn orange.

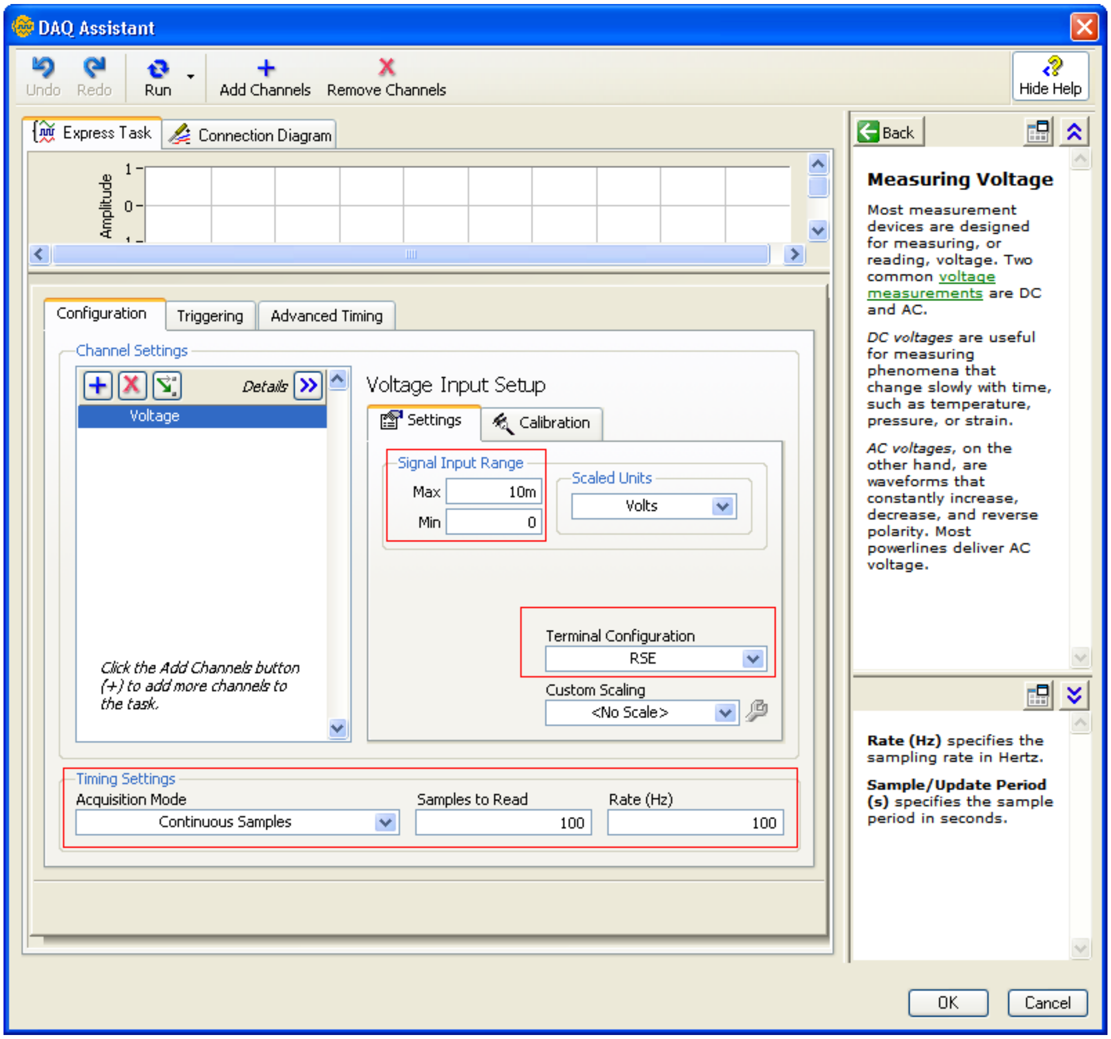
7. The VI has now been configured for use as a subVI. Select File>Save and name the VI.

Part II Insert the subVI into the Top-Level VI

In Part 1 you built a subVI. In this section you will take the subVI and place it into a new VI that collects voltage data from AI0, converts the AI0 voltage to temperature, outputs Digital Output 0 to control heating switch and displays the temperature-time plot on a single waveform. The final VI will look like the figure below:



1. Insert a DAQ Assistant that collects data from AI0. Data collection should continue until the user presses a stop button. Set the parameters as shown.



Voltage

Click the Add Channels button (+) to add more channels to the task.

Signal Input Range
Max 10m
Min 0

Scaled Units: Volts

Terminal Configuration: RSE

Custom Scaling: <No Scale>

Timing Settings

Acquisition Mode	Samples to Read	Rate (Hz)
Continuous Samples	100	100

Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common voltage measurements are DC and AC.

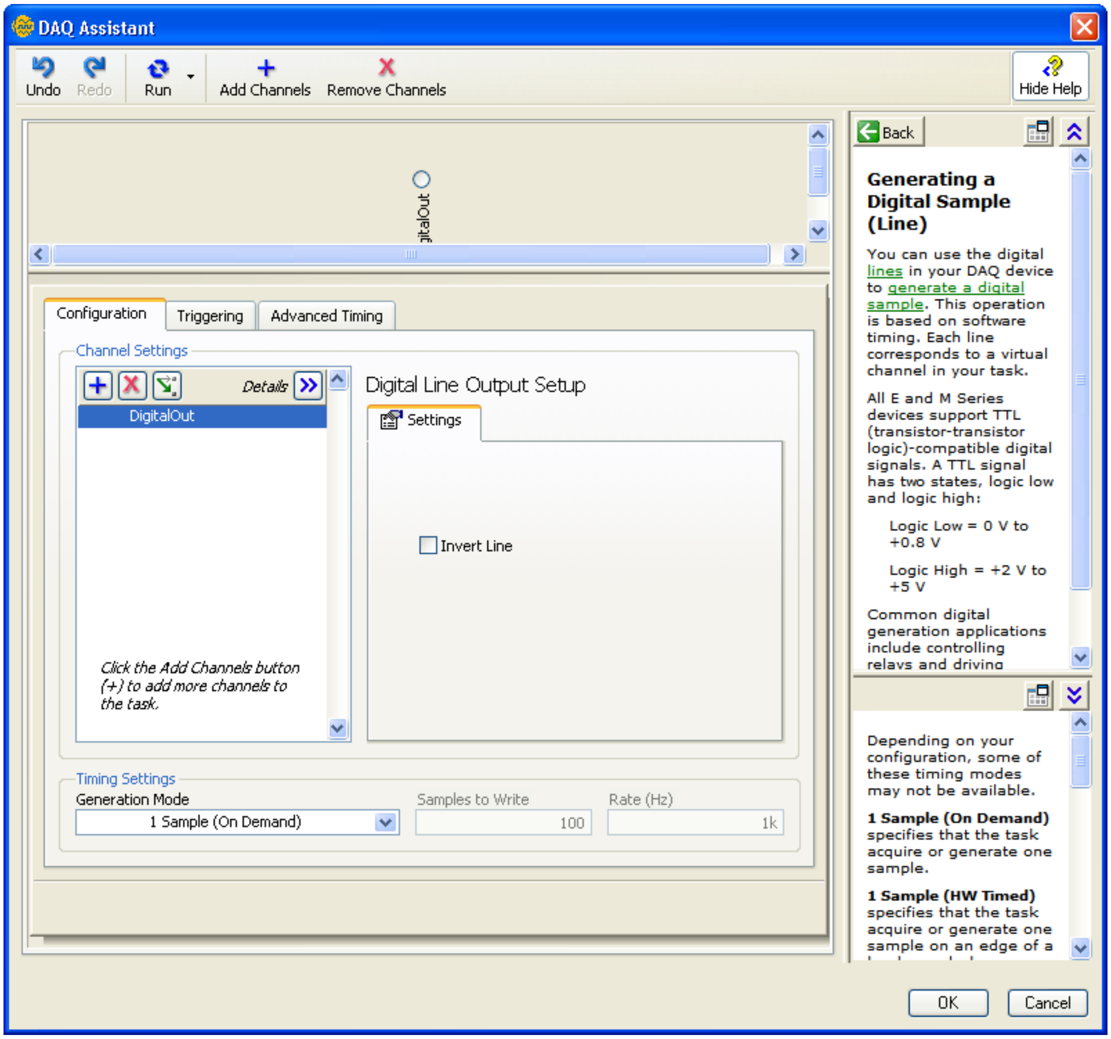
DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

Rate (Hz) specifies the sampling rate in Hertz.

Sample/Update Period (s) specifies the sample period in seconds.

2. Insert a DAQ Assistant that controls heating switch through DO0.



Channel Settings

DigitalOut

Click the Add Channels button (+) to add more channels to the task.

Digital Line Output Setup

□ Invert Line

Timing Settings

Generation Mode: 1 Sample (On Demand)

Samples to Write: 100

Rate (Hz): 1k

Generating a Digital Sample (Line)

You can use the digital lines in your DAQ device to generate a digital sample. This operation is based on software timing. Each line corresponds to a virtual channel in your task.

All E and M Series devices support TTL (transistor-transistor logic)-compatible digital signals. A TTL signal has two states, logic low and logic high:

Logic Low = 0 V to +0.8 V

Logic High = +2 V to +5 V

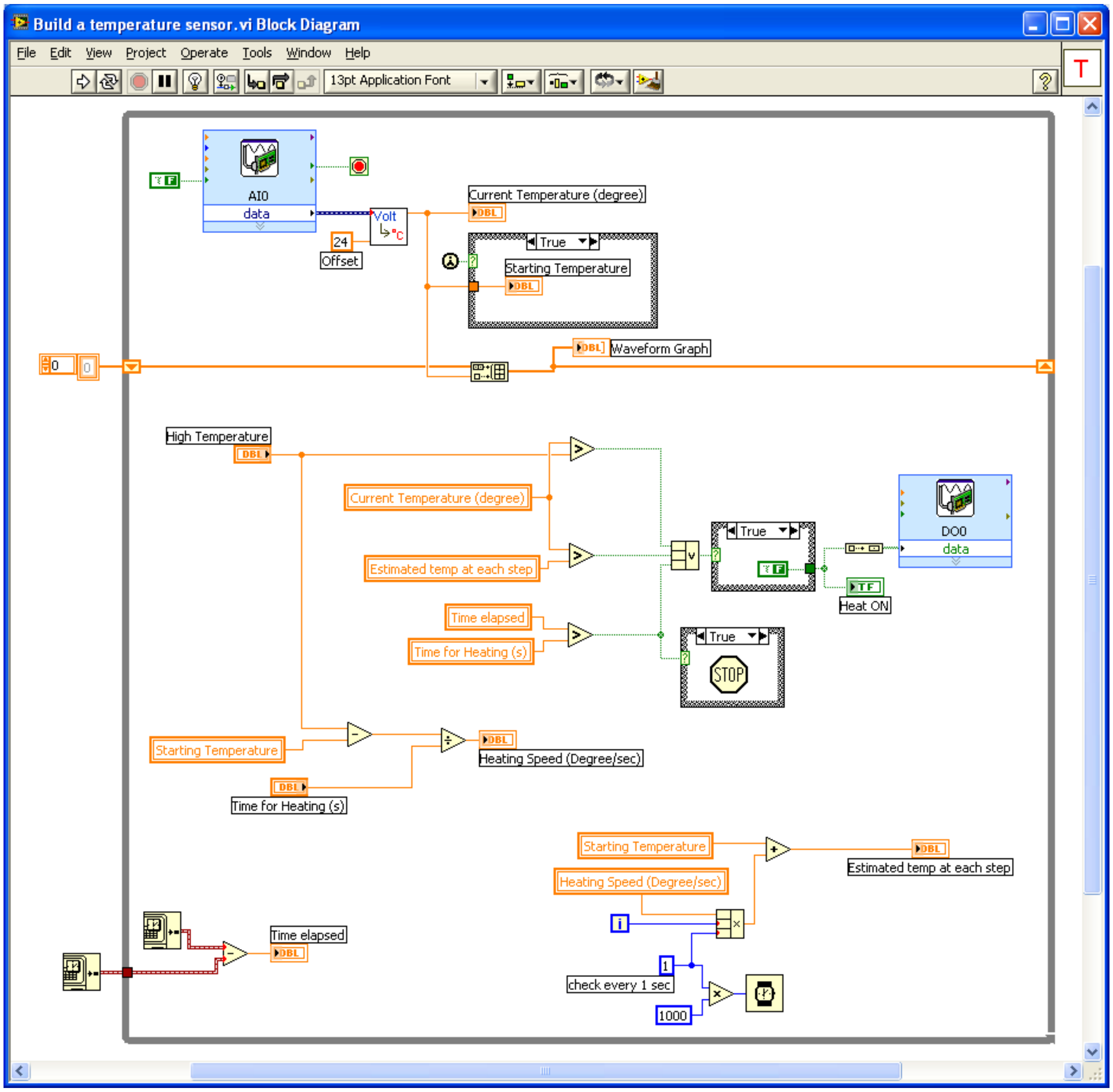
Common digital generation applications include controlling relays and driving

Depending on your configuration, some of these timing modes may not be available.

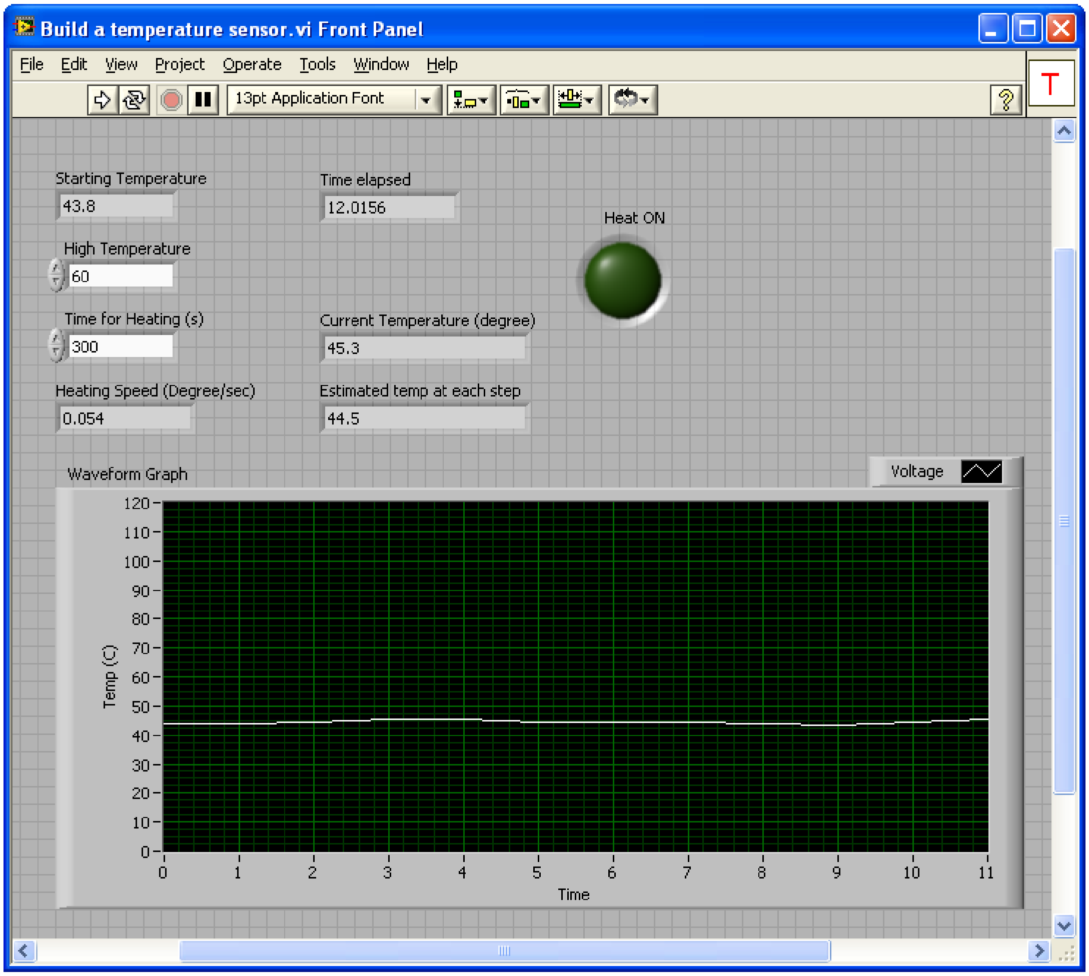
1 Sample (On Demand) specifies that the task acquire or generate one sample.

1 Sample (HW Timed) specifies that the task acquire or generate one sample on an edge of a

3. Finish the rest part as shown.



4. The VI is now complete. Modify the Front panel if needed and save the VI.



Starting Temperature: 43.8

Time elapsed: 12.0156

High Temperature: 60

Time for Heating (s): 300

Current Temperature (degree): 45.3

Heating Speed (Degree/sec): 0.054

Estimated temp at each step: 44.5

Part III Run the VI and Measure Temperature

1. Run the VI created in Part 2. Measure the temperature using the thermocouple Temperature connected to AI0. Having the temperature sensor connected to a multimeter allows for a direct comparison, making it much easier to troubleshoot your circuit and your program.
2. Check the calibration of your temperature sensor over a range of temperatures. Compare the readings of the two sensors in ice water, room temperature, and hot water. Make sure that you provide time for the reading to stabilize. After completing the measurements, dry off the thermocouple.

Features of Program

1. The program runs continuously until the heating time is up. Even if the high temperature is reached, the program keeps running and while the water cools down, the heat will be turned on automatically.

2. When the program starts, the first measured temperature is registered as the "Starting Temperature".

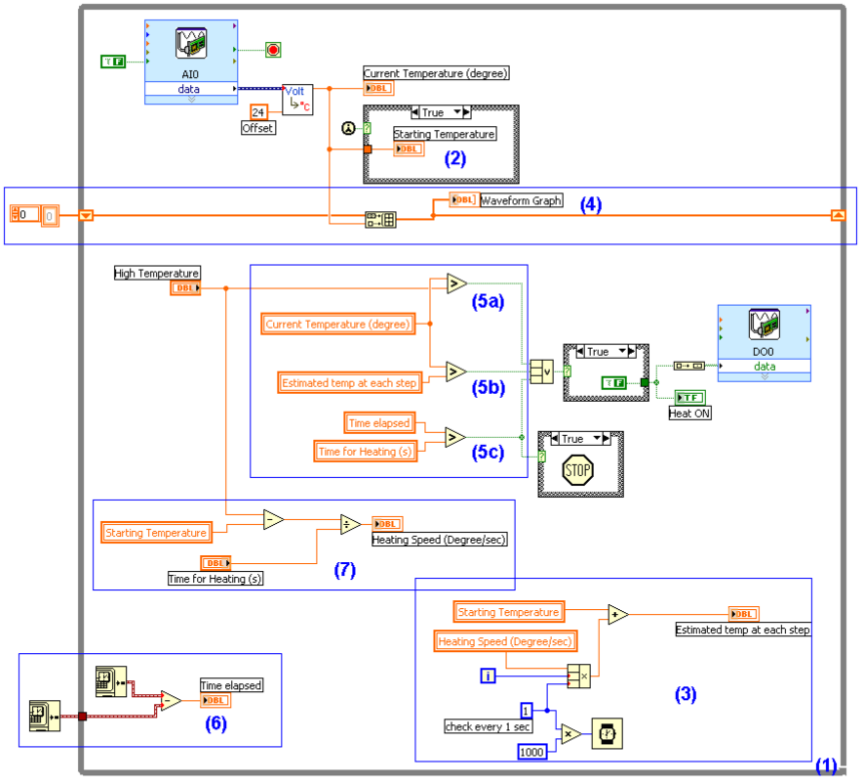
3. The estimated temperature at each step is calculated out by:
   Est. temperature at each step = Staring temperature + Heating speed × Time elapsed

4. The current temperature versus time is displayed in the waveform graph.

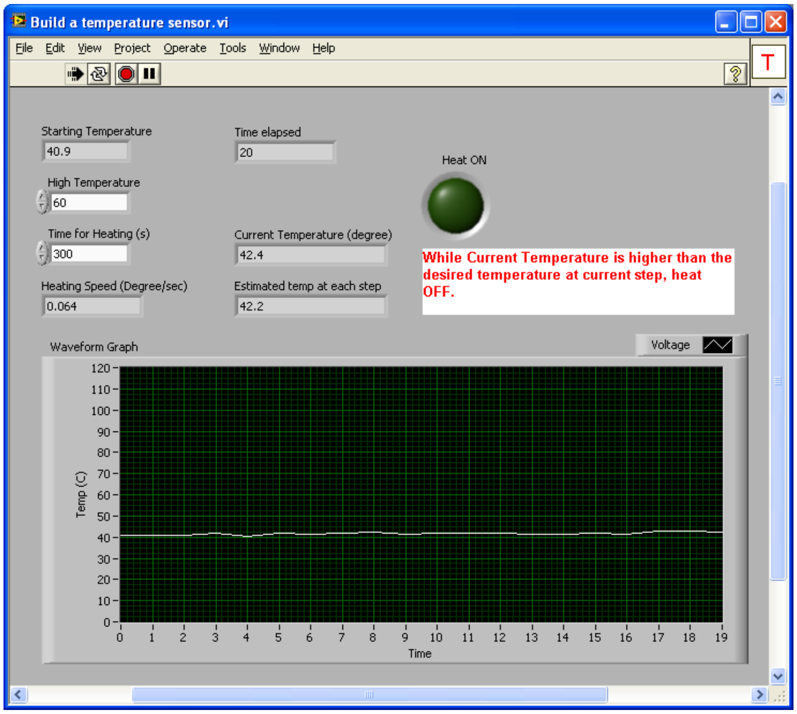
5. Only when at least one of the following three conditions is fulfilled, heating is off. Otherwise, it keeps on.
   a) *Current temperature* higher than *High temperature*, or
   b) *Current temperature* higher than the *estimated temperature at each step*, or
   c) *Heating time* is up.

6. Elapsed time is displayed.

7. Heating speed(degree/sec) is calculated out by:
   Heating speed = (High temperature – Starting temperature) / Time for heating



While Current Temperature is higher than the desired temperature at current step, heat OFF.



Starting Temperature 40.9

Time elapsed 20

High Temperature 60

Heat ON

Time for Heating (s) 300

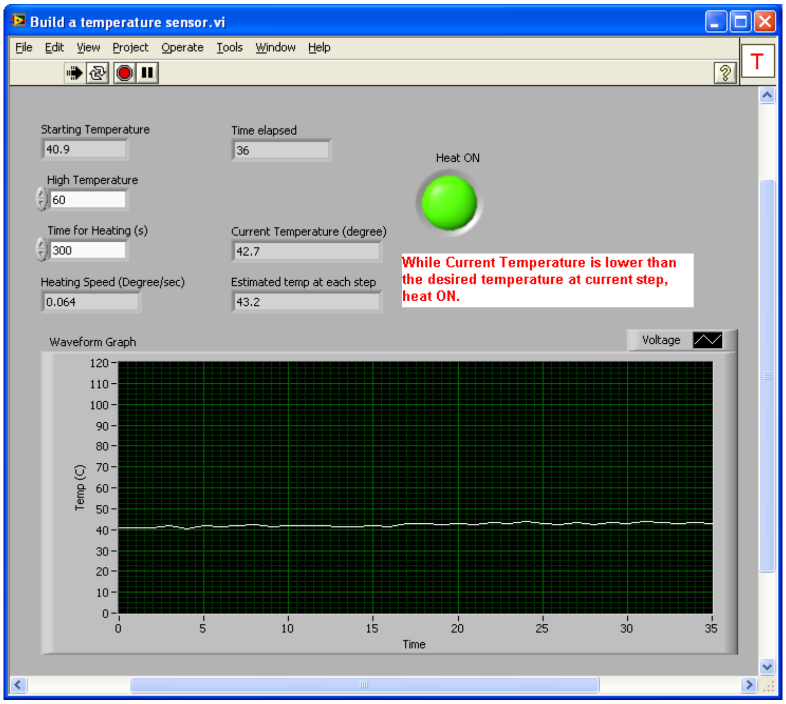
Current Temperature (degree) 42.4

Heating Speed (Degree/sec) 0.064

Estimated temp at each step 42.2

While Current Temperature is higher than the desired temperature at current step, heat OFF.

While Current Temperature is lower than the desired temperature at current step, heat ON.



Starting Temperature 40.9

Time elapsed 36

High Temperature 60

Heat ON

Time for Heating (s) 300

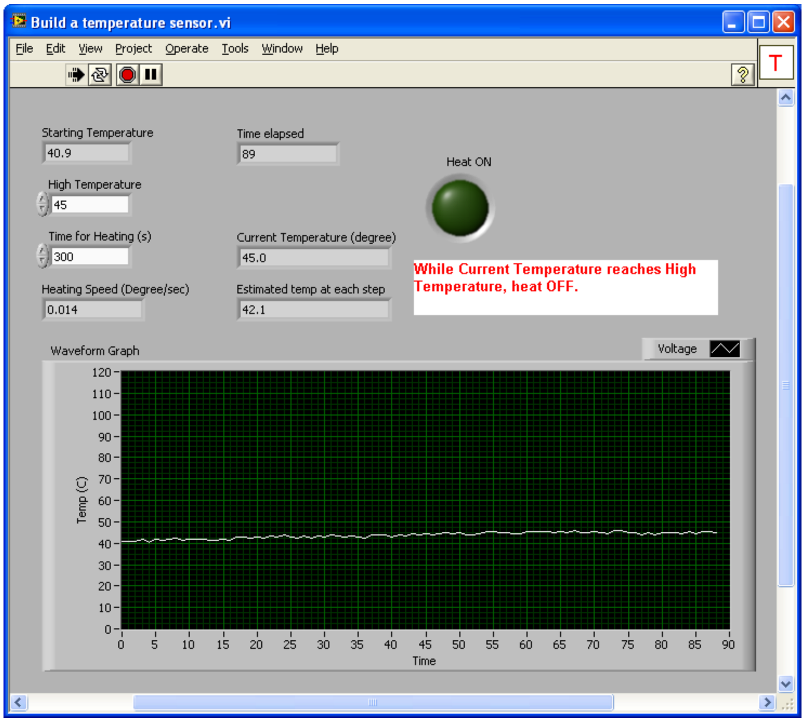
Current Temperature (degree) 42.7

Heating Speed (Degree/sec) 0.064

Estimated temp at each step 43.2

While Current Temperature is lower than the desired temperature at current step, heat ON.

While Current Temperature reaches High Temperature, heat OFF.



Build a temperature sensor.vi

Starting Temperature: 40.9

Time elapsed: 89

High Temperature: 45

Time for Heating (s): 300

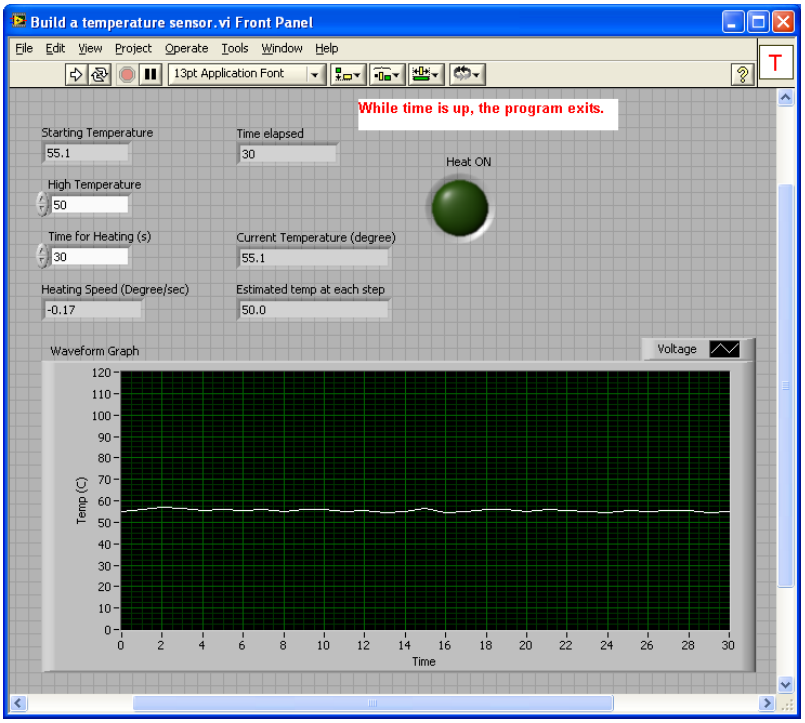
Current Temperature (degree): 45.0

Heating Speed (Degree/sec): 0.014

Estimated temp at each step: 42.1

While Current Temperature reaches High Temperature, heat OFF.

While time is up, the program exits.



Build a temperature sensor.vi Front Panel

While time is up, the program exits.

Starting Temperature: 55.1

Time elapsed: 30

High Temperature: 50

Time for Heating (s): 30

Current Temperature (degree): 55.1

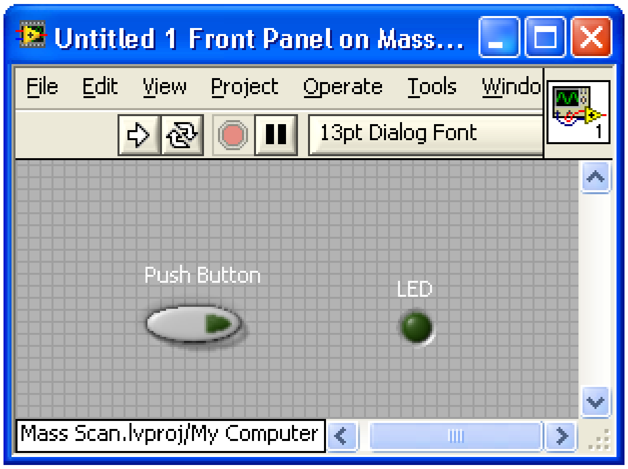
Heating Speed (Degree/sec): -0.17

Estimated temp at each step: 50.0

For the Report

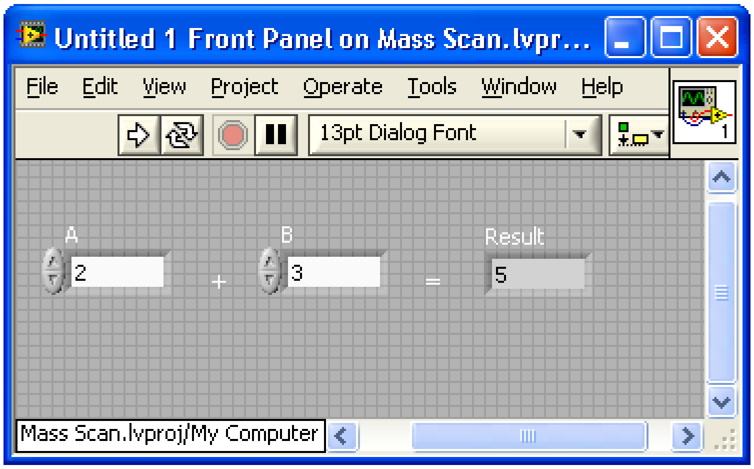
1. What's LabVIEW short for? What's VI?

2. Construct a VI that uses a round push-button control to turn on a square light indicator whenever the push button is depressed. The front panel is very simple and should look something like the one shown below.



Untitled 1 Front Panel on Mass...

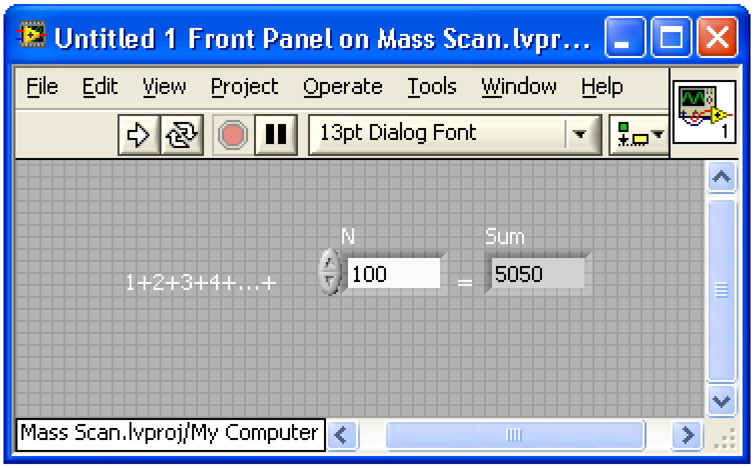
3. Construct a VI that could calculate additions shown like below.



Untitled 1 Front Panel on Mass Scan.lvpr...

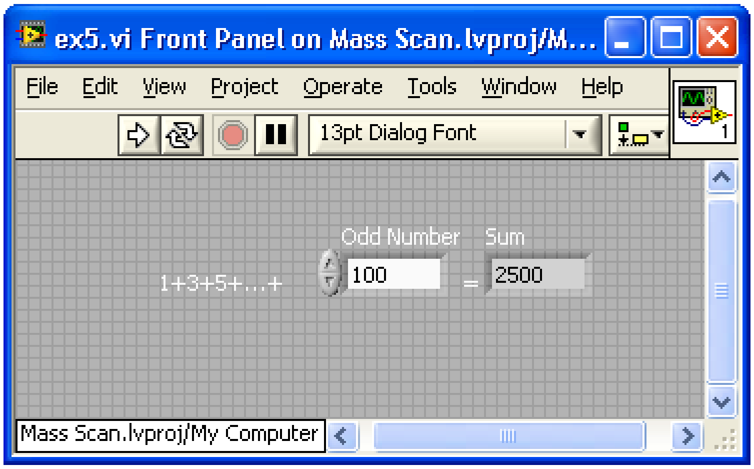
A: 2 + B: 3 = Result: 5

4. Construct a VI that could calculate the sum of a series of continuous integer numbers from 1 to given integer N. Sum=1+2+3+…+N


Untitled 1 Front Panel on Mass Scan.lvpr…
File
Edit
View
Project
Operate
Tools
Window
Help
13pt Dialog Font
Mass Scan.lvproj/My Computer

1+2+3+4+…+ N 100 = Sum 5050

5. Construct a VI that could calculate the sum of a series of continuous odd numbers from 1 to given odd number N. Sum=1+3+5+…+ N


ex5.vi Front Panel on Mass Scan.lvproj/M…
File
Edit
View
Project
Operate
Tools
Window
Help
13pt Dialog Font
Mass Scan.lvproj/My Computer

1+3+5+…+ Odd Number 100 = Sum 2500

Tip: For ex4 & ex5, “For Loop” or “While Loop” may be needed for calculation.