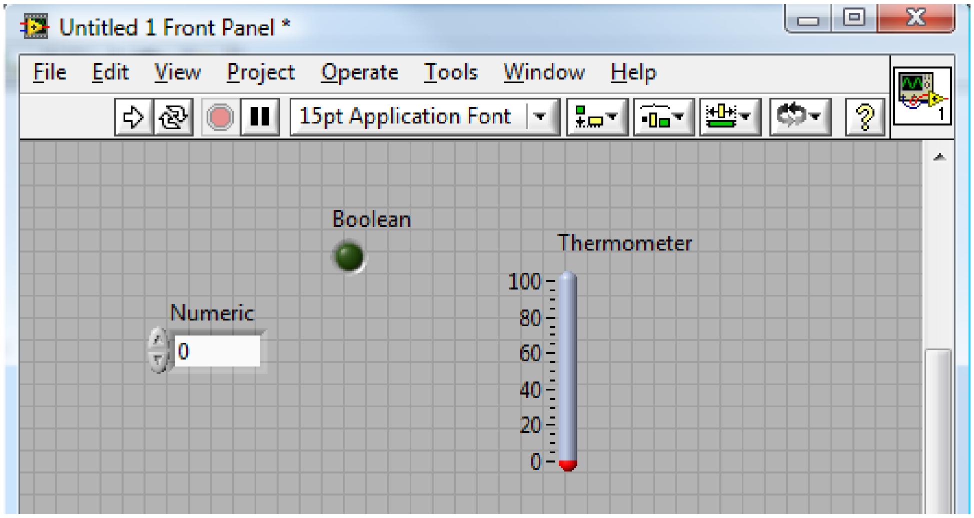
Chapter
5

Presenting Results

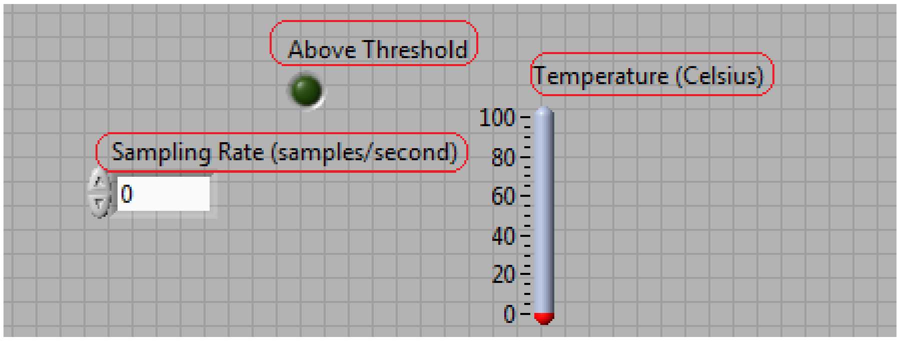
FRONT PANEL DESIGN

As discussed in previous sections, controls and indicators are front panel items that allow the user to interact with your program to both provide input and display results. When designing your front panel, you access controls and indicators by right-clicking the front panel to show the controls palette.

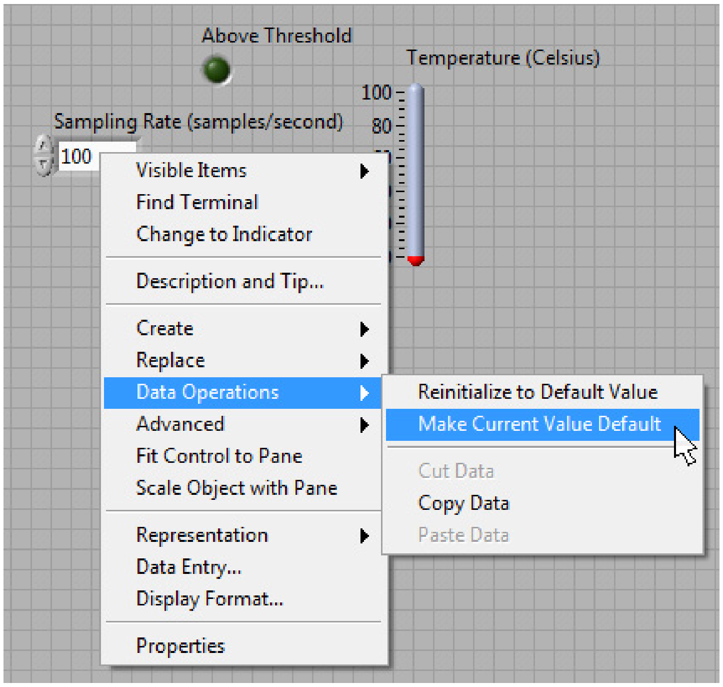
It is important to realize that the front panel gives the user a way to interact with the source code. It allows the user to change the values passed to the source code and see the data that the source code computes. When designing the user interface, choose controls and indicators that are appropriate for the task you want to perform. For example, when you want to set the sampling rate, use a numeric control that shows the increment/decrement button. When you want to indicate that a threshold value has been met, choose a Boolean LED indicator. When you want to display temperature, choose a chart or thermometer indicator.



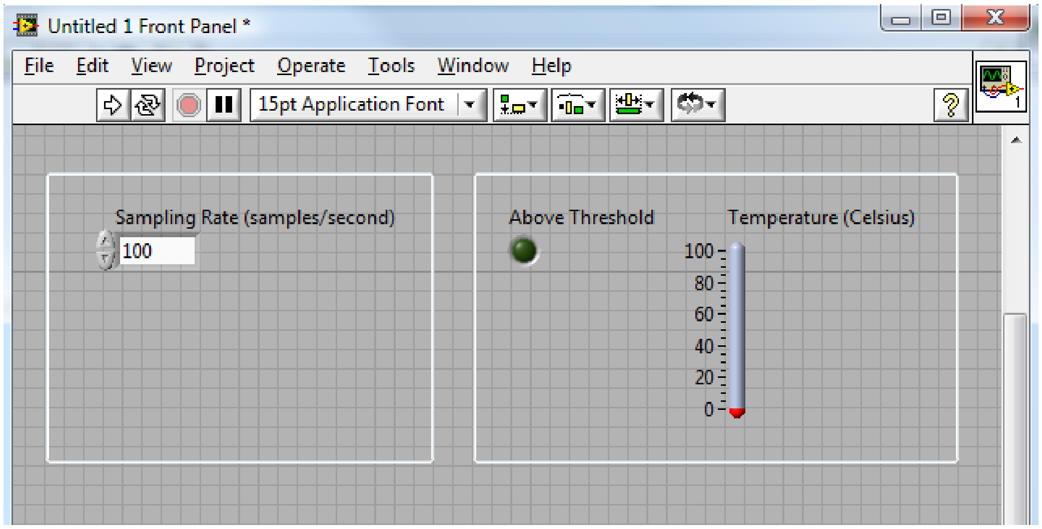
Make sure to label controls and indicators clearly. These labels help users identify the purpose of each control and indicator. Also, labeling helps you document your code on the block diagram. Control and indicator labels correspond to the names of terminals on the block diagram.



Spend time designing your front panel. LabVIEW offers many options for setting the properties of controls and indicators. You can view these properties by right-clicking the object and browsing through the shortcut menu and submenus. For example, sometimes you might want your LabVIEW VI to open with default values for your controls. In this example, if you want the Sample Rate control to load with a default value of 100 samples/second, first change the value of this control to 100 and then set the property that makes the current value as the default value.



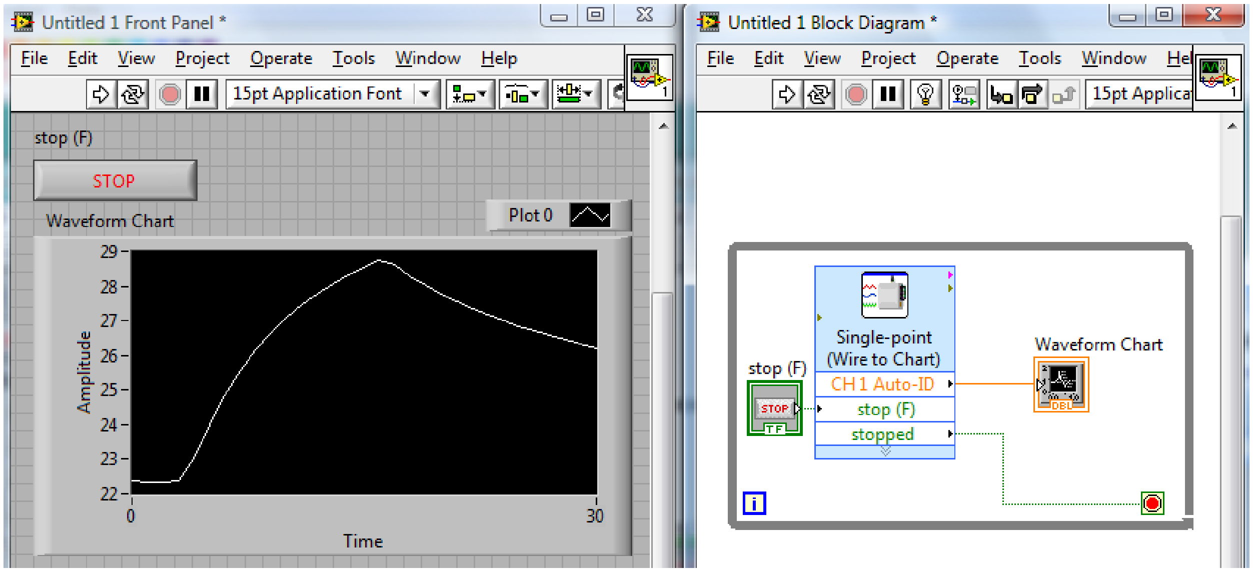
White space and alignment are probably the most important techniques for grouping and separation; the more items that your eye can find on a line, the cleaner and more cohesive the organization seems. When items are on a line, the eye follows the line from left to right or top to bottom. When you design the front panel, consider how users interact logically with the VI. Group controls and indicators, and consider the use of decorative borders (found in the Decoration subpalette). Leave some blank space between objects to make the front panel easier to read. Blank space also prevents users from accidentally clicking the wrong control or button. In general, spend some time spacing and aligning objects to provide users with a clean, easy-to-use interface.



GRAPHS AND CHARTS

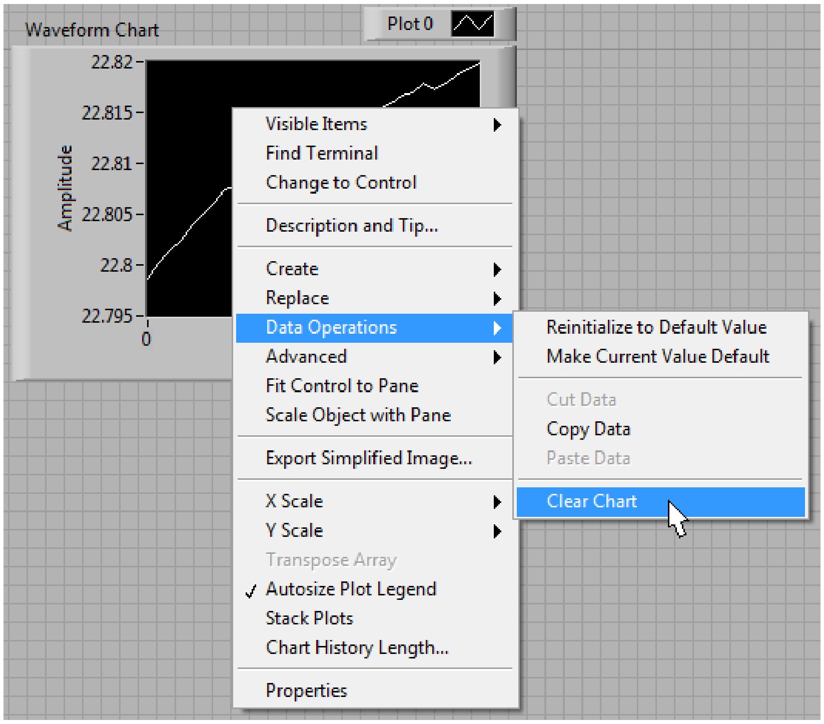
When displaying sensor data, the most commonly used indicators are the graph and chart. A plot line on a chart or graph is a powerful visual display of your sensor measurements. Understanding the differences between a chart and graph is important in order to properly design your LabVIEW program.

In general, a chart is used when performing slow data acquisition, updating the plot every time a data point is retrieved (point-by-point data acquisition). The plot on a chart is appended with the new data point to create a history. This allows the user to see the current reading or measurement in context with data previously acquired. When more data points are added than can be displayed on the chart, the chart scrolls so that new points are added to the right side of the chart, while old points disappear to the left.

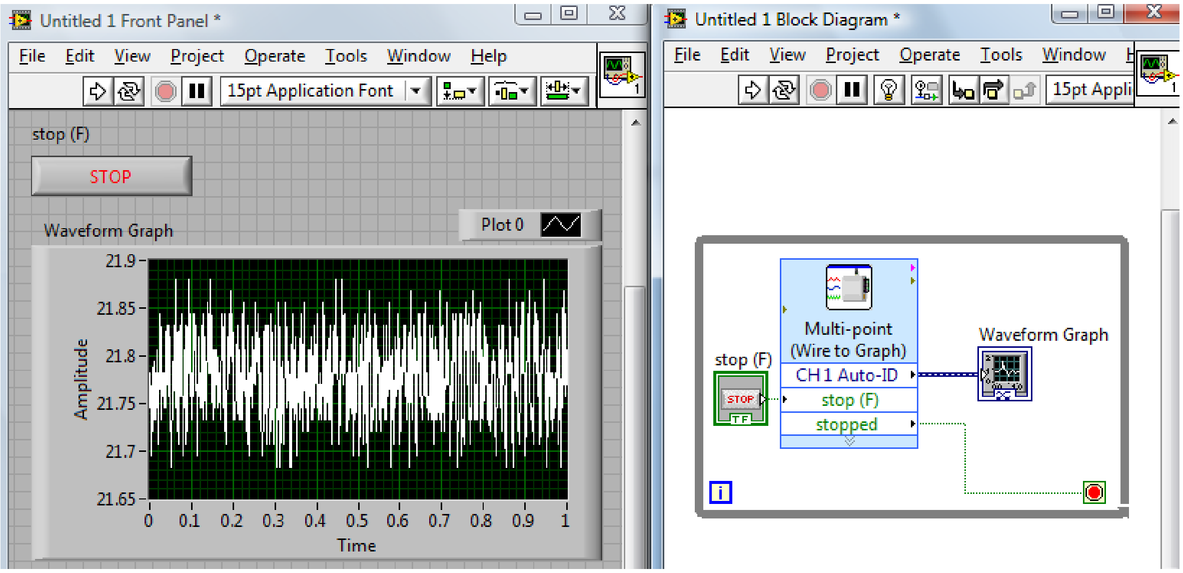


The program shown in the figure uses the Analog Express VI to collect single-point data at a rate of 1 sample/second for 30 seconds. With each iteration of the While Loop, a new data point is collected and appended to the plot line of the front panel chart.

An important thing to realize about a chart is that the history of data points is stored on the plot between runs of a LabVIEW VI. This means that if you run a VI and have a 30 second plot on your chart, the next time you run the VI, the new data points will be appended. The plot does not automatically clear and start over at time of zero when you click the Run button. One method to clear the chart between runs is to go to the shortcut menu, and select Data Operations ► Clear Chart.



A graph, on the other hand, does not store previous data points. When the data are plotted, the graph discards the previously plotted data and displays only the new data. Only showing the new data makes a graph the correct choice of indictor when you are collecting and displaying large sets of data points (multi-point data acquisition) at a fast rate. A chart would not work for fast, multi-point acquisition, because it would continually be appending these large data sets. This creates memory issues, and does not provide the user with a helpful visual representation of the data.

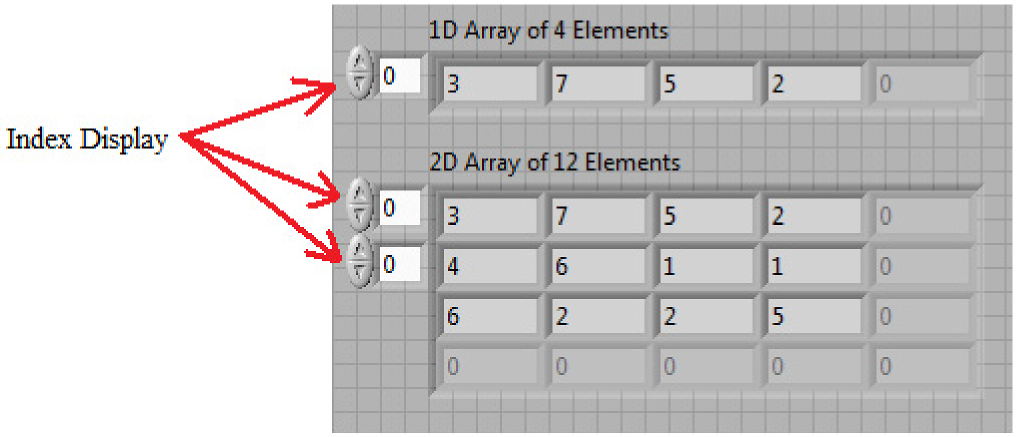


The program shown in the figure uses the Analog Express VI to repeatedly collect multi-point data at a rate of 1000 samples/second and a length of 1 second. When the loop iterates, the graph is updated with the most recent 1000 data points, providing the user with a constantly updated snapshot of the data.

The default look of a chart window is black with no gridlines. The default window of the graph window is with gridlines. However, these are properties that can be modified. In fact, the two indicators may be very hard to tell apart. If you are unsure, access the shortcut menu (right-click on the object) and open the Properties dialog box. The title of the dialog box will tell you whether it is a graph or a chart.

BUILDING ARRAYS WITH LOOPS

An array is an ordered, indexed list of data elements of the same type, similar to a matrix. An array, which consists of elements and dimensions, is either a control or an indicator—it cannot contain a mixture of controls and indicators. Elements are the data or values contained in the array. A dimension is the length, height, or depth of an array. Arrays are very helpful when you are working with a collection of similar data, and when you want to store a history of repetitive computations.



1D Array of 4 Elements

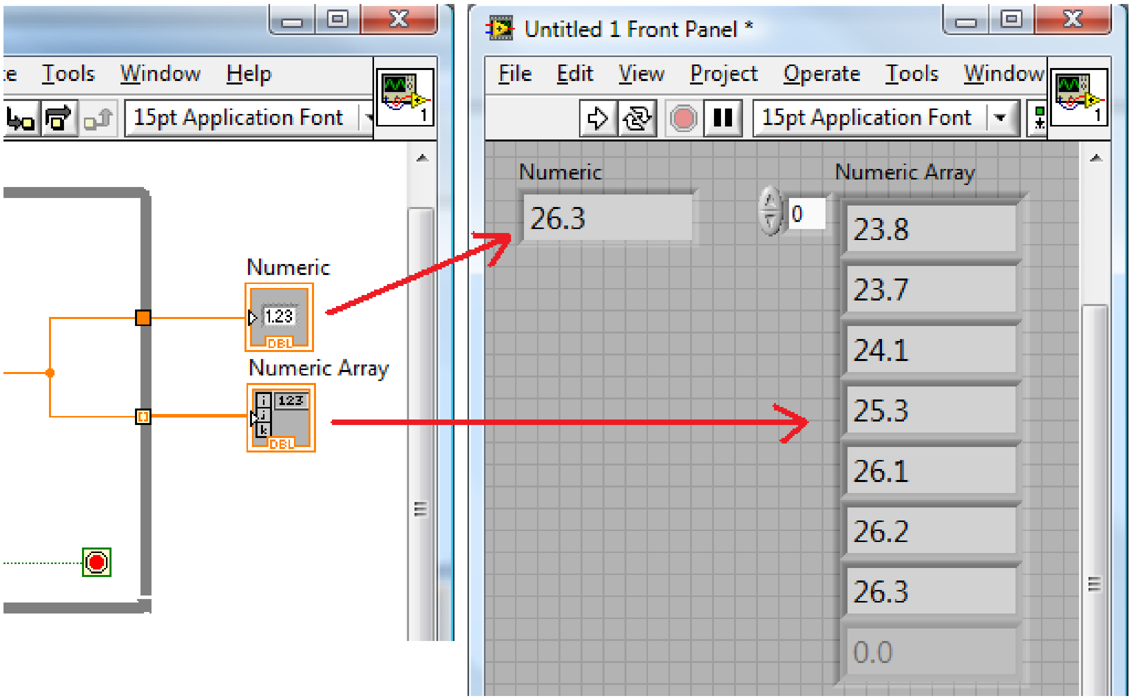
3	7	5	2	0

2D Array of 12 Elements

3	7	5	2	0
4	6	1	1	0
6	2	2	5	0
0	0	0	0	0

The figure above shows two front panel array indicators with numeric data elements. Array elements are ordered. The index display has up/down arrows that allow you to navigate to a particular element in the array. In LabVIEW software, the array index is zero-based. This means that if a one-dimensional (1D) array contains n elements, the index range is from 0 to n – 1, where index 0 points to the first element in the array, and index n – 1 points to the last element in the array.

For Loops and While Loops can index and accumulate arrays at their boundaries. This is known as auto-indexing. The indexing point on the boundary, called a tunnel, is created every time a wire crosses a loop boundary. The For Loop default is auto-indexing enabled. The While Loop default is auto-indexing disabled. The state of this tunnel is changed with the shortcut menu that appears when you right-click on the tunnel. Visually, the tunnel is solid when auto-indexing is disabled, and bracketed when auto-indexing is enabled. The wire exiting the loop is thicker when you use auto-indexing, because the wire contains an array instead of a single value.



An example of the use of auto-indexing would be to perform single-point data acquisition within a loop (temperature data, for example), and build the array by wiring the data point to the loop's border and auto-indexing the tunnel. Every data point that occurred (one point for each loop iteration) is stored in the tunnel. Upon exiting the loop, all data points are now stored in an array. This array can then be used in the next step of your program to perform analysis on the data or save to a file.

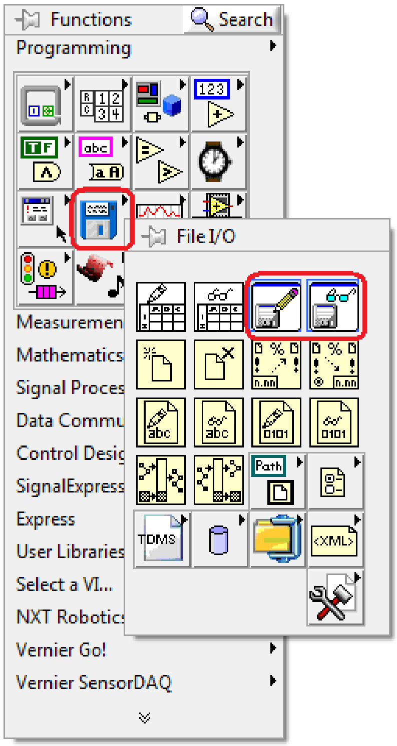
HIGH LEVEL FILE I/O FUNCTIONS

File I/O operations pass data from memory to and from files. High Level File I/O functions provide a higher level of abstraction to the user by opening and closing the file automatically before and after reading or writing data.

In LabVIEW, you can use File I/O functions to perform many operations on files, including:

- Open and close data files
- Read data from and write data to files
- Read from and write to spreadsheet-formatted files
- Move and rename files and directories
- Change file characteristics
- Create, modify, and read a configuration file

The Write to Measurement File Express VI and Read from Measurement File Express VI are easy-to-use Express VIs that are excellent for simple applications. Use these to write data to a text-based measurement file (.lvm) or a binary measurement file (.tdm) format, and read back the data from the generated measurement file. You can specify the file name, file format, and segment size.


Functions
Search
Programming
Measuremen
Mathematics
Signal Proces
Data Commu
Control Desig
SignalExpress
Express
User Libraries
Select a VI...
NXT Robotics
Vernier Go!
Vernier SensorDAQ
File I/O
Path
TDMS
<XML>