Problem Statement: Design a program to do a loop-back test for the analog portion of the NI USB 6008 DAQ unit. The program should have a user interface to provide dynamic data to control the analog output of the unit. There should be an appropriate display showing analog control data and the analog output response to that data in 'real time'. The display should be around 100 data samples wide in order to clearly show I/O timing differentials and transmission errors.

Problem Solution: The NI USB 6008 is a multifunction device that can provide both analog and digital I/O operations. The analog operations involve analog to digital conversions, ADC, and digital to analog conversions, DAC. The digital operations involve accepting or putting out digital (binary) information compatible with computer processing or instrument control. Each I/O port is called a channel. All these operations have become associated with the term, data acquisition, or DAQ. All DAQs have specifications that tell the user how the device works and what the application limitations are in order to ensure proper performance. Operating the device outside of specified limits will result in meaningless results and device failure. The user's manual for your DAQ is posted on Manhattan.

A loop-back test involves driving an output port of a device or system with known data and having that output drive the input port of the device/system to compare the original data with the data that went through the device. The test can check for transmission errors and delays. The two NI USB 6008 outputs can only process one data point at a time. This means that the user can only supply a single value to the device for processing and presenting to the output port. A numeric control provides a single value. Data at an input port of the NI USB 6008 can be monitored by LabVIEW software. A waveform chart can be used to provide a visual record of user input data compared to device output data (looping back to the input). To check device behavior under changing input data the loop-back test should be running continuously or running under an iteration count long enough for the user to supply appropriate test data. A While or For loop provides iterative capabilities.

The I/O terminals of the unit are shown in tables 1 and 2 below. You need this information in order to properly interconnect your signal paths.

Table 1: Analog Terminals

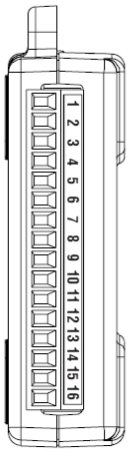
Module	Terminal	Signal, Single-Ended Mode	Signal, Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0–
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1–
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2–
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3–
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

Table 2: Digital Terminals

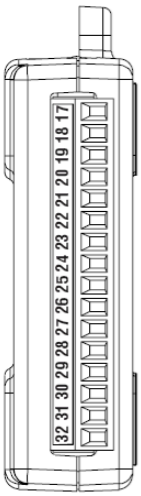
Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0 6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
	31	+5 V
	32	GND

Table 1 shows that you can work with either eight, single-ended inputs or four differential inputs and have two single-ended outputs to work with. The voltage range for the analog inputs is -10 volts to +10 volts. The voltage range for the analog outputs is 0 to 5 volts. Since you are doing a loop-back test (an output drives an input) you will not be able to test an input over its full specified range.

- Connect the USB DAQ to your computer (necessary for DAQ assistants to function).
- Start LabVIEW and open a blank VI (Front Panel and Block Diagram). Save the VI as **DAQtest.vi**.
- Go to the menu bar (either window) and open the **Tools** palette. Make sure **Automatic Tool Selection** is 'lit'.
- One way of providing a 'dynamic data' interface is to use a slider numeric control. Put one on the Front Panel. Label the slider **Dynamic Data** and set its limits from 0 to 5.
- To satisfy displaying dynamic data in real time put a **Waveform Chart** on the Front Panel.
- To give the user proper halt control over the program put a **Stop** button on the Front Panel. Label the button, **Halt Test**.
- To allow the program to run continuously and process changing input data, put a **While Loop** in the Block Diagram and wire the Stop button to it.
- To get the slider numeric control to communicate with the DAQ, place an **Express Output DAQ Assistant** to the right of the slider control icon in the Block Diagram. When the DAQ Assistant dialog opens select **Generate Signals >> Analog Output >> Voltage**. When the **Physical** channel window opens (with the DAQ model showing) select **ao0** (first output channel) then click on **Finish** to close the task window and open the **Output Voltage** setup window. In the **Generation Mode** pull-down menu select **1 Sample (On Demand)**. Set the **VoltageOut** variable to **0** then click on **OK** to close the Assistant dialog and have it build the interface. When the build is complete, wire the output of the slider numeric control to the data input of the DAQ Assistant icon. Note that using a DAQ Assistant in repeating-loop operations produces relatively slow program execution because the Assistant opens and closes the specified task each time the Assistant is called.

- To properly document the program so far, go to the **Tools** palette and click on the **Text Edit** button. Click on blank space to the right of the Dynamic Data icon and type in a text box, *User data to drive the DAC*. In a blank space over the Assistant icon create a text box containing, *DAQ Interface to drive analog output*. Put the Tools palette back into Automatic mode.
- To have the program sense analog loop-back data coming into the DAQ, place an **Express Input DAQ Assistant** to the left of the Waveform Chart icon in the Block Diagram. When the Assistant dialog opens select **Acquire Signals >> Analog Input >> Voltage**. When the **Physical** channel window opens (with the DAQ model showing) select **ai0** (first input channel) then click on **Finish** to close the task window and open the **Voltage Input** setup window. In the **Acquisition Mode** pull-down menu select **1 Sample (On Demand)**. Set the **Voltage** variable to 0. Set the **Terminal Configuration** pull-down menu to **RSE**. Set the **Signal Input Range** to **min = 0**, **max = 5**. Click on the **Finish** button to close the Assistant dialog and build the interface. Again note that the Assistant is being used in a repeating-loop operation.
- Document the addition of the input assistant with a text box containing, *DAQ interface sensing analog input*.
- In order to plot both the user data and the loop-back data on the same chart go to the Functions palette and select **Express >> Sig Manip >> Merge Signals** and place the **Combined Signal** icon to the left of the Waveform Chart icon and wire them together.
- Wire the output data port of the DAQ Assistant 2 to the bottom input of the signal combiner. Label this wire, *Loop-back Signal*, with a text box. Wire the output data port of the slider numeric control to the top input of the signal combiner.
- Go to the Front Panel and briefly run the program to check for any errors and to allow the display chart to configure itself for a double-trace plot.
- Right click on the Waveform Chart. When the dialog opens find **Chart History Length…** and set its value to 100.
- Click on the Waveform Chart legend box and expand it to show two plot legends.
- Open the **Properties** dialog for the Waveform Chart object in the Front Panel. In the **Display Format** tab select **Automatic formatting** and **2 Digits of precision** for the x axis. In the **Plots** tab select a dotted plot for **Plot 0** and make the dots open, white circles. For **Plot 1** select a dotted plot and make the dots closed, red circles. Change the label **Plot 0** to **Original In**. Change the label **Plot 1** to **DAQ Out**. In the **Scales** tab select **Autoscale** for the **Time (X-axis)**, set the x-axis style to unlabeled major and minor tic marks, and set **min=0, max=5** for the **Amplitude (Y-axis)** with **Autoscale deselected**. Close the Properties dialog. When you are finished you should have something like that shown below.

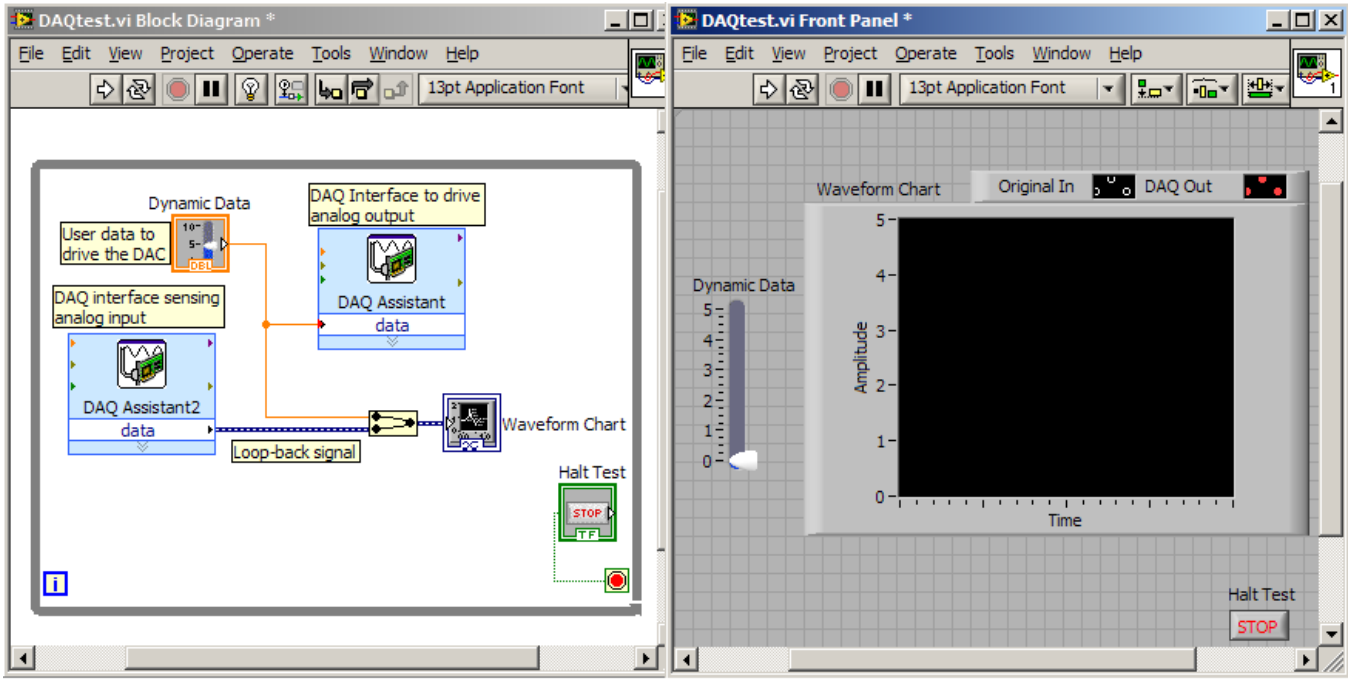


Fig. 6
Completed VI

- Connect a jumper wire from terminal **14 (AO0)** to terminal **2 (AI0)** of the 6008 DAQ. Make sure the green LED on the DAQ is flashing.
- Start the program by clicking on the Run arrow. Move the slider up and down. The chart display should show patterns of dots following the motion of the slider. For rapid motions of the slider you should see that there is a noticeable horizontal separation between white and red dot pairs. This separation is the result of processing delays between the input and the output of the DAQ. To test other loop-back combinations you need to right-click on the appropriate DAQ assistant, open the Properties dialog, and change the appropriate physical channel under **Channel Settings**. For example, run the test using AI2 with AO1.

The block diagram below shows an alternative problem solution. Its front panel has additional controls to select Input Channel, Output Channel, and Input Terminal Configuration options which were embedded in the DAQ Assistant interface. The program uses individual DAQ objects which are found in the **Measurement I/O >> DAQmx-Data Acquisition** palette. The operations involving setting up and shutting down I/O tasks are put outside the looping operation since we need those operations only once per program run. The actual reading and writing tasks are kept inside the loop since we are trying to test dynamic behavior. The result is a program that executes at least five times faster than the one using the DAQ Assistant.

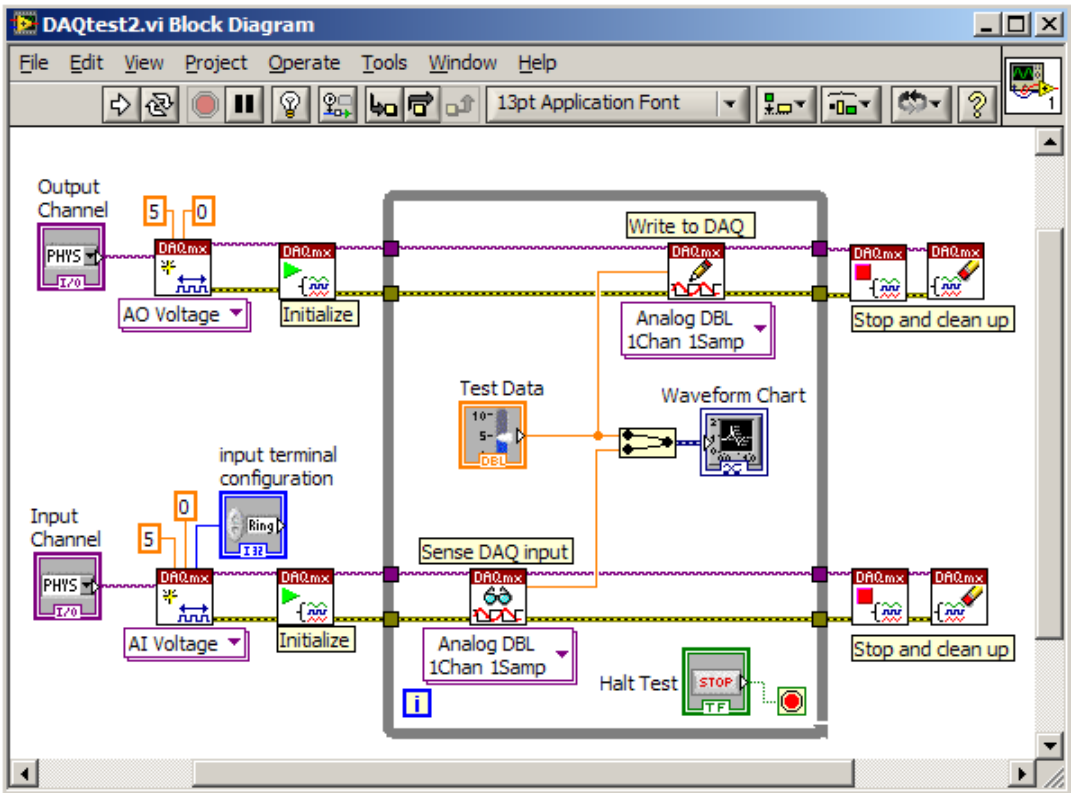


Fig. 7
Alternate Problem Solution