Exercise 10-3: Streaming N-Channels Acquisition Data to a Text File

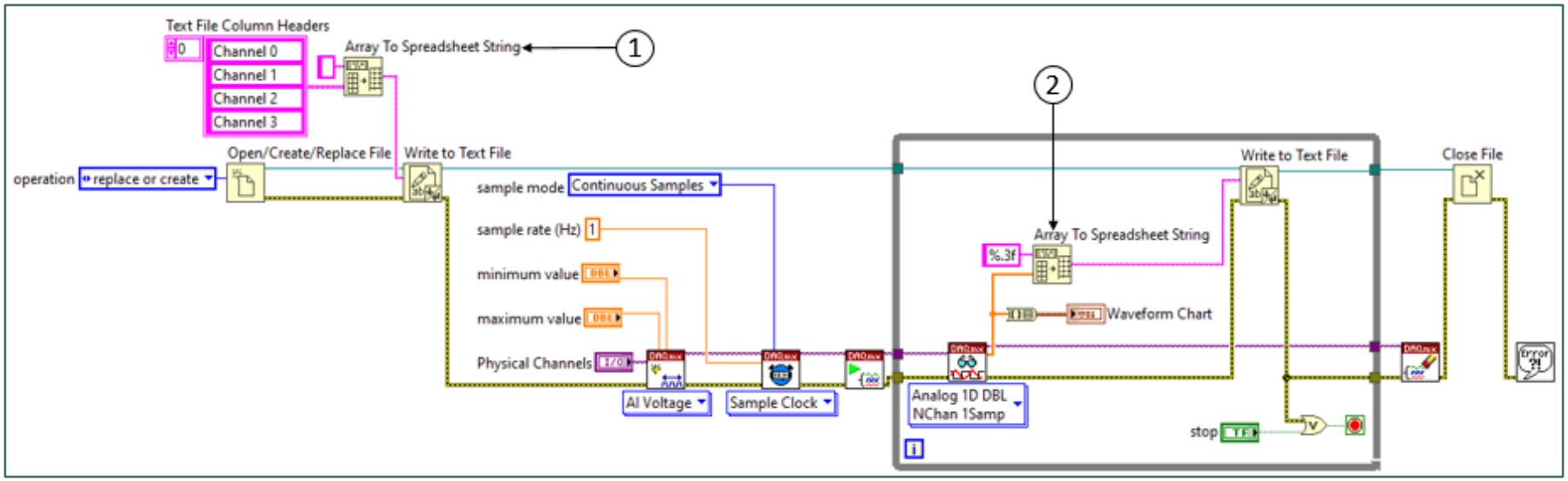
Goal

- Explore examples that stream *N*-channels, 1-sample and *N*-channels, *N*-samples acquisition data to a file.

Instructions

N-Channels, 1-Sample (1D DBL) Streaming Example

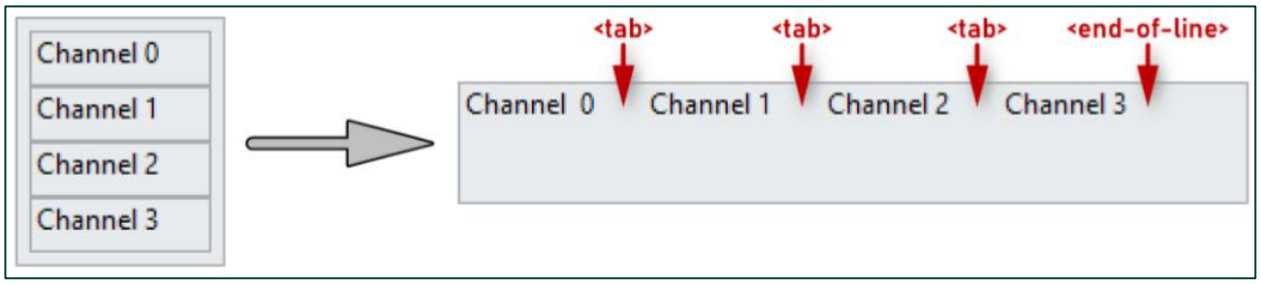
1. Open the following project: `C:\Exercises\LabVIEW Core 1\Low-Level Stream to Text File (NChan)\Low-Level Stream to Text File (NChan).lvproj`.
2. From the **Project Explorer** window, open the Low-Level Stream to Text File (NChan 1Samp) VI.
3. Explore the Block Diagram.



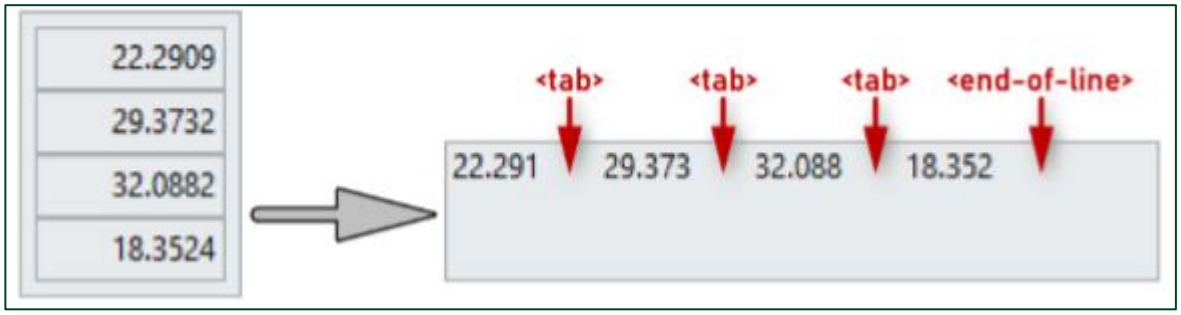
---

1. **Array to Spreadsheet String function** – This function is configured to create a spreadsheet string, which adds a delimiter between each array element and adds an end-of-line constant at the end of the string. In this VI, it turns the incoming string array into the following string:
   `Channel 0<tab> Channel 1<tab> Channel 2<tab> Channel 3<end-of-line>.`
2. **Array to Spreadsheet String function** – Use this function to convert the 1D DBL array data into a string data type because the Write to Text File function requires a string input.

- The first Array to Spreadsheet String function will create the following result.



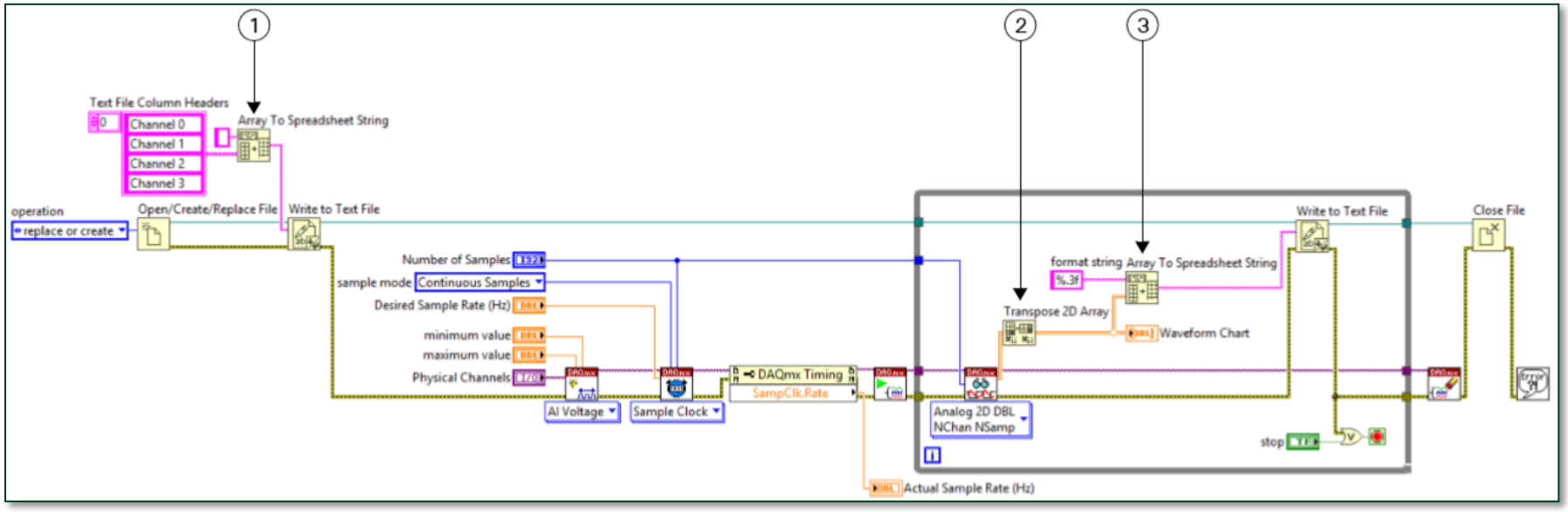
- The second Array to Spreadsheet String function, as configured in this exercise, would format a 1D DBL input to a spreadsheet string, as shown in the following example.



4. Test the VI
   - Run the VI.
   - In the file dialog box, save the log file as `C:\Exercises\LabVIEW Core 1\ Low-Level Stream to Text File (NChan)\Stream Data (NChan 1Samp).txt`.
   - Click the **Stop** button after approximately 10 seconds.
   - In Windows Explorer, open and explore the text file.
5. Use a probe to view wire values to better understand the functionality of the functions and VIs. You can also create indicators of wires and view the indicator values on the front panel, if you prefer.

N-Channels, *N*-Samples (2D DBL) Streaming Example

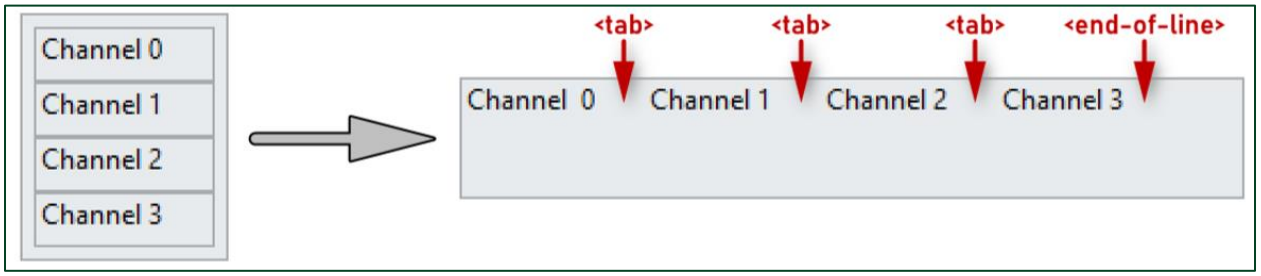
1. From the **Project Explorer** window, open the Low-Level Stream to Text File (NChan NSamp) VI.
2. Explore the block diagram



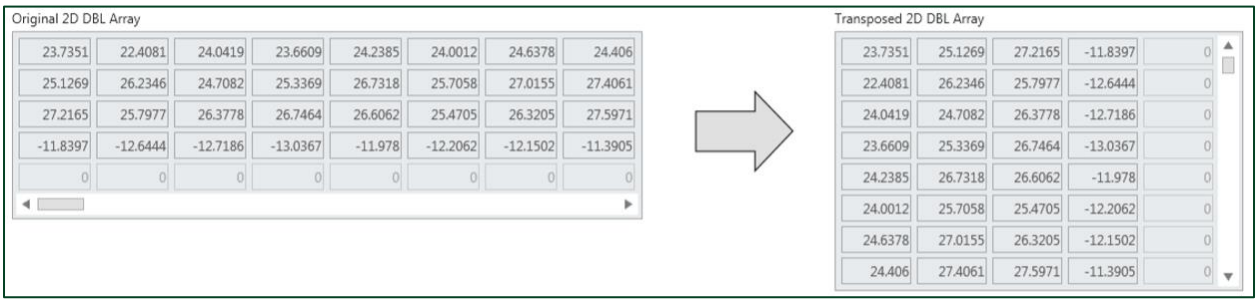
---

1. **Array to Spreadsheet String function** – In this VI, the Array to Spreadsheet String function is configured to create a spreadsheet string, which adds a delimiter between each array element and adds an end-of-line constant at the end of the string. In this example, it turns the incoming string array into the following string:
   `Channel 0<tab> Channel 1<tab> Channel 2<tab> Channel 3<end-of-line>.`
2. **Transpose 2D Array function** – Rearranges the elements of 2D array such that 2D array becomes transposed array.
3. **Array to Spreadsheet String function** – Use this function to convert the 2D DBL array data into a string data type because the Write to Text File function requires a string input.

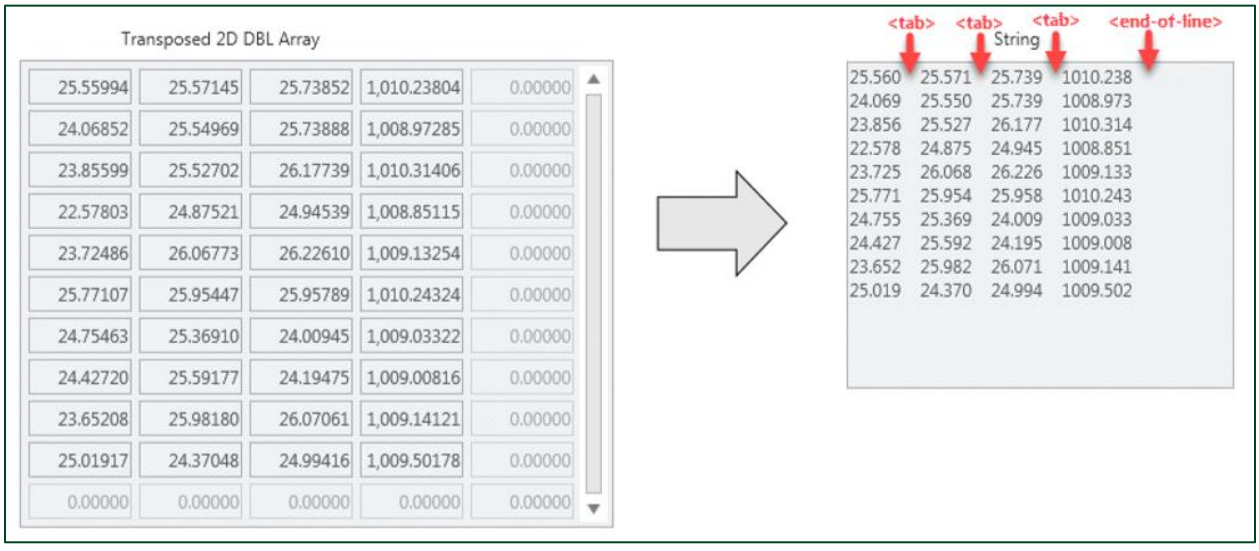
- The first Array to Spreadsheet String function, as configured in this exercise results in the following output.



- The Transpose 2D Array function rearranges the elements of the array, as shown in the following figure.



- The Second Array to Spreadsheet String function, as configured in this exercise, would format the output of the Transpose 2D Array function to a spreadsheet string, as shown in the following figure.



3. Test the VI.
   - Run the VI.

- In the file dialog box, save the log file as `C:\Exercises\LabVIEW Core 1\ Low-Level Stream to Text File (NChan)\Stream Data (NChan NSamp).txt`.
- Click the **Stop** button after approximately 10 seconds.
- In Windows Explorer, open and explore the text file.

4. Use a probe to view wire values to better understand the functionality of the functions and VIs. You can also create indicators of wires and view the indicator values on the front panel, if you prefer.

On the Job

Answer the following questions for your own applications.

1. Do you need to continuously stream multi-channel data to a text file?

   _____
   _____
   _____
   _____

2. If so, which of the above example is closest to meeting your needs?
   - *N*-Channels, 1-Sample (1D DBL) Streaming Example

     OR
   - *N*-Channels, *N*-Samples (2D DBL) Streaming Example

3. What modifications do you need to make to these examples to meet your needs?

   _____
   _____
   _____
   _____
   _____
   _____

End of Exercise 10-3