

Using VISA

Goal

Use the VISA API functions to communicate effectively with an instrument.

Description

During this exercise, you create a VI that can communicate with different types of hardware. The VI will follow the general VISA API programming flow.

The VI will connect to the instrument, write data to the instrument, read the response, and then close the connection when it is finished.



Note The Agilent 34411A digital multimeter is used in this exercise. You can use any message-based instrument to complete this exercise

Figure 1 and Figure 2 show an example of the completed front panel and block diagram respectively.

Design

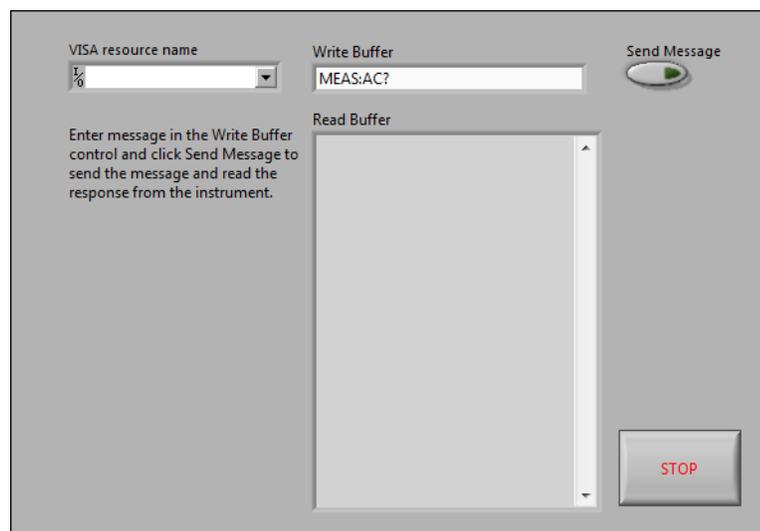


Figure 1. VISA Write Read Buffer Front Panel

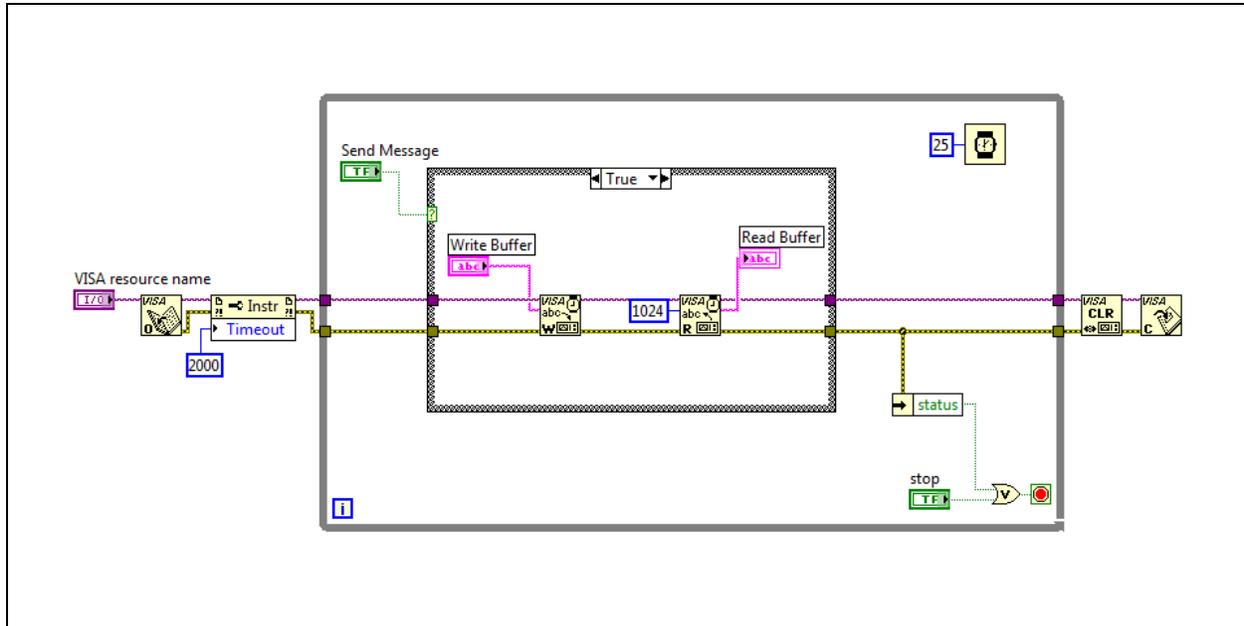
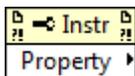


Figure 2. VISA Write Read Buffer Block Diagram

Implementation

The files that you need to complete this exercise are here: <NI eLearning>\LV Core 1\Instr Ctrl Prog\Exercise.

1. Open a blank VI.
 - Open LabVIEW.
 - Select **File»New VI**.
2. Place a VISA Open function onto the block diagram.
 - Press <Ctrl-E> to switch to the block diagram.
 - Select **View»Functions Palette**.
 - Select the **Instrument I/O** palette.
 - Select the **VISA** category, then select the **VISA Advanced** sub-category.
 - Select and drag the **VISA Open** function onto the block diagram.
3. Place a VISA Property Node onto the block diagram.
 - Select and drag the **VISA Property Node** onto the block diagram.



4. Configure the placed VISA functions.
 - Right-click the VISA resource name terminal on the VISA Open function and select **Create»Control**.
 - Wire the VISA resource name out terminal of the VISA Open function to the reference input terminal of the VISA Property Node.
 - On the VISA Property Node click on property and select **General Settings»Timeout Value**.
 - Right-click the Timeout terminal on the VISA Property Node and select **Change All to Write**.
 - Right-click the Timeout terminal on the VISA Property Node and select **Create»Constant**.
 - Change the value to 2000.
 - Wire the error out terminal from the VISA Open function to the error in terminal of the VISA Property Node.

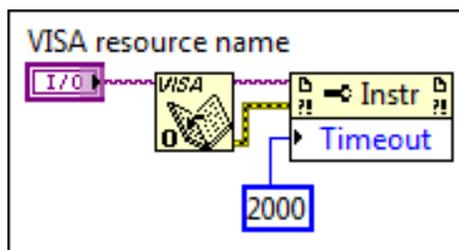


Figure 3. Initialize the Session

5. Place a While Loop onto the block diagram.
 - Select the **Structures** category from the **Programming** palette.
 - Select and drag the **While Loop** onto the block diagram.
 - Expand the size of the While Loop to give enough room to create the rest of the code.
6. Place a Case structure inside the While Loop.
 - Select the **Structures** category.
 - Select and drag the **Case structure** inside the While Loop.
 - Expand the size of the Case structure to provide enough room to create code inside the Case structure.



7. Create the condition for the Case structure.
 - Right-click the Case Selector of the Case structure and select **Create»Control**.
 - Rename the control to *Send Message*.
 - Press <Ctrl-E> to switch to the front panel.
 - Right-click the Send Message button and select **Mechanical Action»Latch When Released**.

8. Modify the True case.
 - Press <Ctrl-E> to switch to the block diagram.
 - Select the **Instrument I/O** palette and the **VISA** category.
 - Place a **VISA Write** function onto the block diagram in the Case structure.
 - Place a **VISA Read** function onto the block diagram in the Case structure.
 - Right-click the write buffer terminal of the VISA Write function and select **Create»Control**.
 - Right-click the read buffer terminal of the VISA Read function and select **Create»Indicator**.
 - Right-click the byte count terminal of the VISA Read function and select **Create»Constant**.
 - Change the value to 1024.



9. Wire the VISA resource name terminals and error terminals as shown in Figure 4.

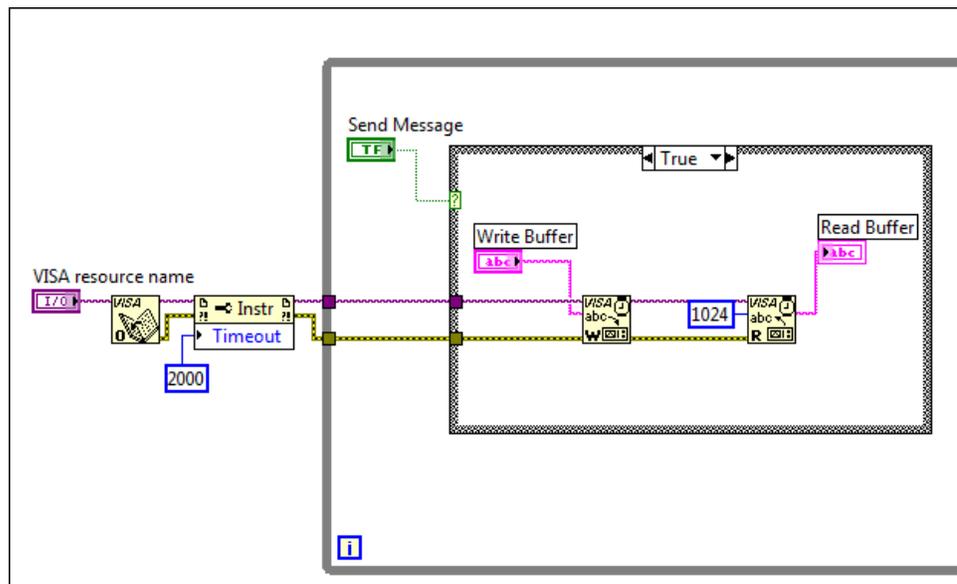


Figure 4. Case Structure Code

10. Add a delay in the While Loop.

- Select the **Timing** category from the **Programming** palette.



- Select and drag a **Wait (ms)** function onto the block diagram in the While Loop.
- Right-click the milliseconds to wait terminal of the Wait (ms) function and select **Create»Constant**.
- Change the value to 25.

11. Place VISA Clear and VISA Close functions.

- Select the **Instrument I/O** palette and the **VISA** category.



- Select and drag the **VISA Clear** function onto the block diagram.

- Select the **Instrument I/O** palette, the **VISA** category, and the **VISA Advanced** sub-category.



- Select and drag the **VISA Close** function onto the block diagram.



- Select and drag the **Or** function onto the block diagram.
- Wire the status output of the Unbundle by Name function into the x terminal of the Or function.
- Press <Ctrl-E> to switch to the front panel.
- Select **View»Controls Palette**.
- Select the **Boolean** category.



- Select and drag the **Stop Button** onto the front panel.
- Press <Ctrl-E> to switch to the block diagram.
- Wire the Stop button to the y terminal of the Or function.
- Wire the output of the Or function to the conditional terminal of the While Loop.

14. Arrange the controls on the front panel as seen in Figure 1.

15. Save the VI.

- Select **File»Save**.
- Save the VI as `visa read write buffer.vi`.

Test

1. On the VISA resource name, select the device that is connected to your computer. If you do not have an instrument, skip the remaining steps.



2. Click the **Run** button.
3. If your instrument is capable of reading an AC Voltage measurement, like the Agilent 34411A digital multimeter, type `MEAS:AC?` in the Write Buffer control. Otherwise type `IDN?`.
4. Click the **Send Message** button and the response should show up in the Read Buffer indicator.
5. Click the **Stop** button to stop the program.

End of Exercise

Notes
