C400 USB Labview 2011 VIs

Table of Contents

1. Introduction

The nPont C400 controller USB interface is manufactured by Future Technology Devices International (FTDI). FTDI provides communication example VIs for Labview 7, available to download at http://www.ftdichip.com/Support/SoftwareExamples/CodeExamples/LabVIEW.htm

nPoint has created a set of C400 USB VIs, including an example that shows how to use the nPoint VIs together with a basic user interface. The document "C400 Advanced Controller Communication.pdf" describes the communication protocol and controller memory addresses used for the nPoint sub VI block diagrams.

2. nPont C400 Example VI

2.1 Front Panel Overview

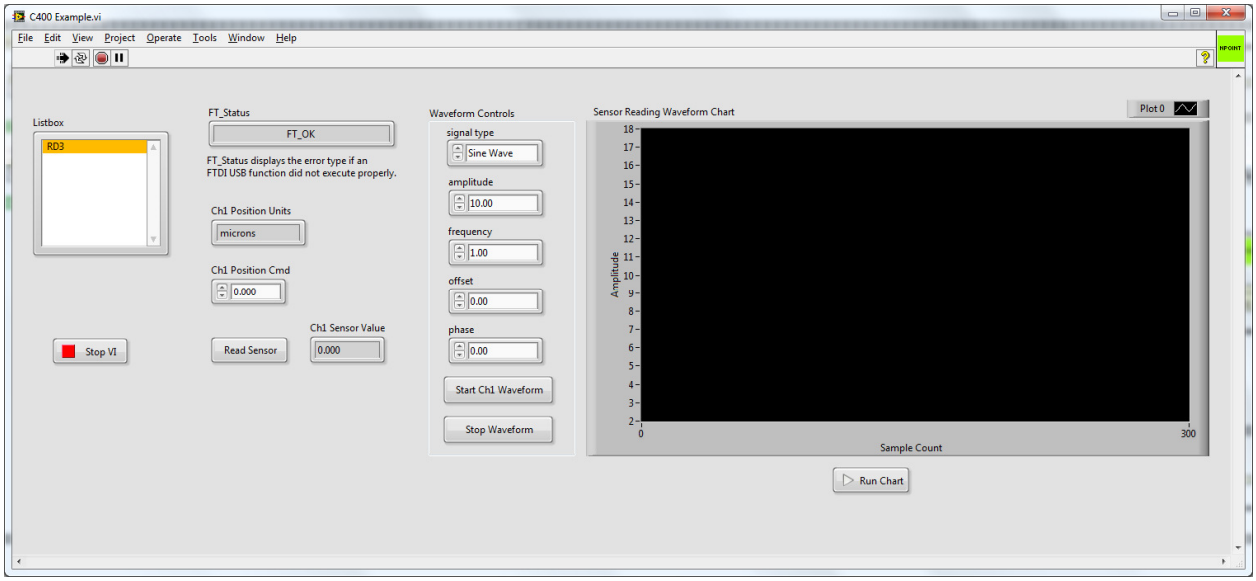


Figure 1 - C400 Example VI front panel

With a C400 controller connected and powered on, run the C400 Example VI. The Listbox will be populated with the serial number for any C400 controllers connected to the PC. Select a serial number from the listbox by clicking it, and then click the "Connect to selected C400 controller" button.

When the VI successfully connects to the C400 controller, several controls will become visible along with a text box showing the unit of measure for the controls (typically microns). The user interface controls are hard coded to Channel 1 of the controller to help keep the example block diagram simple. Each of the sub VIs has a channel input so that they can be used for any channel. The digital position command can be changed by typing in a position or clicking the up/down arrows. The Channel 1 sensor can be read a single time by clicking the Read Sensor button, or the sensor can be read continuously and plotted by clicking the Run Chart button. Click the Run Chart button again to stop continuously reading the sensor. A periodic waveform can be sent to the controller and started by entering waveform

parameters and clicking the Start Ch1 Waveform button. The static digital position input command is summed with the digital waveform and will act as a separate offset from the waveform offset control. Click the Stop VI button to disconnect from the controller and stop the VI.

2.2 Block Diagram Overview

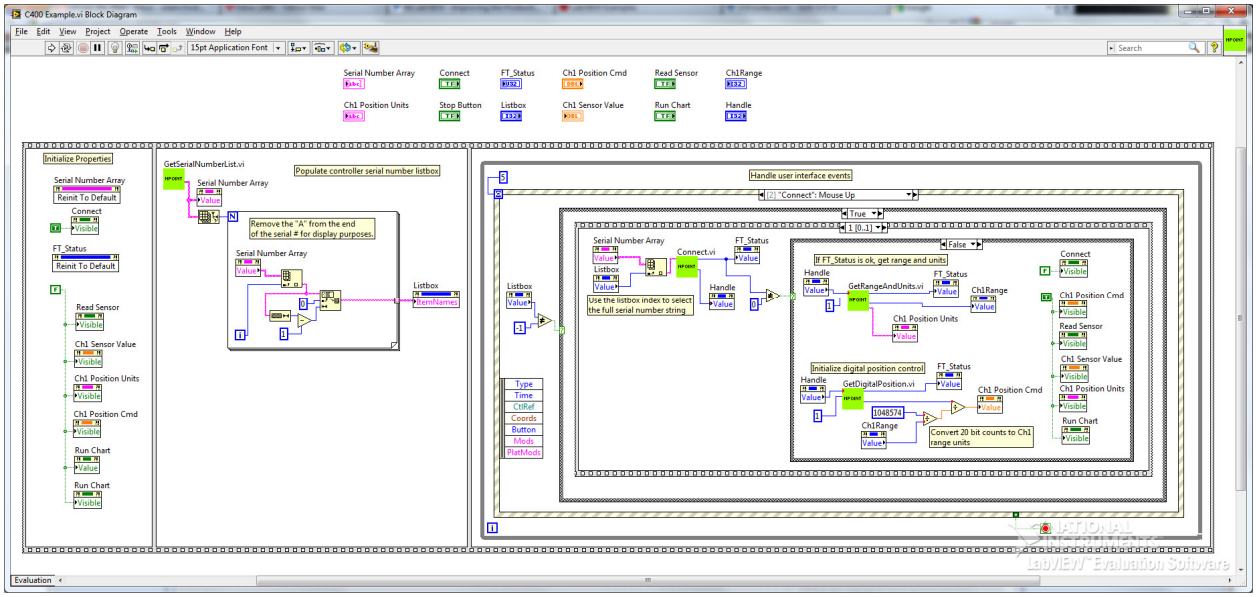


Figure 2 - C400 Example VI block diagram

The C400 Example VI block diagram has three main sequence frames. The first frame initializes properties to default values.

The second frame gets a list of C400 controller serial numbers, stores them in an array, and populates the list box with the "A" character removed from the end of the serial number so that it matches nPoint documentation. The C400 controller FTDI chip has two ports. Only the A port is used, the B port is not physically connected. The "A" or "B" character is added to the end of the device serial number to differentiate the ports since the two ports use the same serial number.

The third frame contains a **while loop** around an **event structure**. The event structure handles user interface events such as button clicks, and the while loop runs until the Stop button mouse up event is triggered.

2.2.1 Connect Event

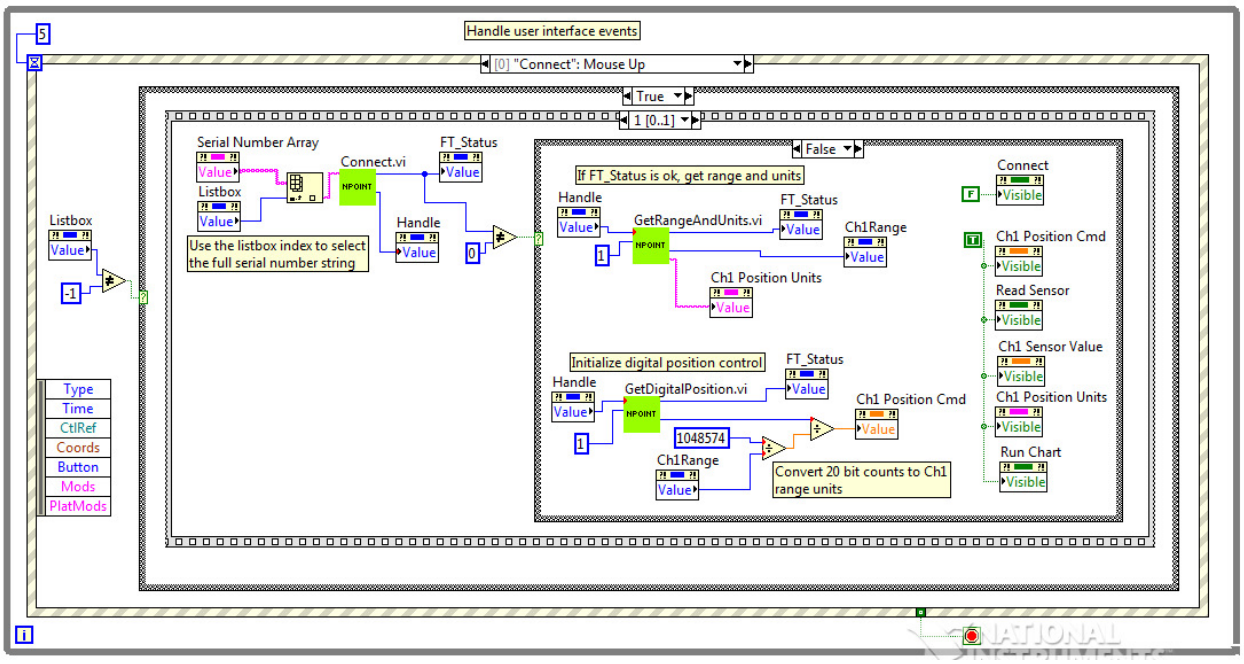


Figure 3 - Connect event in C400 Example VI

When the "Connect to selected C400 controller" button is clicked, the Connect event is triggered. If the listbox output value is -1, no controller is selected and the user is prompted to select a controller. If the listbox output value indicates an item is selected, the listbox output is used as an index for the Serial Number Array to call the nPoint Connect VI. The Connect VI waits for the controller to be ready if it was recently powered on, so the Connect button text is set to "Waiting for C400 to be ready" as a visual indicator for the user. If the FT_Status output of the Connect VI is a value of zero for FT_OK enumeration, the GetRangeAndUnits VI is called to get the Channel 1 range and units string. The GetDigitalPosition VI is used to initialize the Ch1 Position Cmd control to match the value in the controller. The Connect VI sets the Handle value that used as a reference to the device connection for all further communication until the device is closed.

2.2.2 Stop Button Event

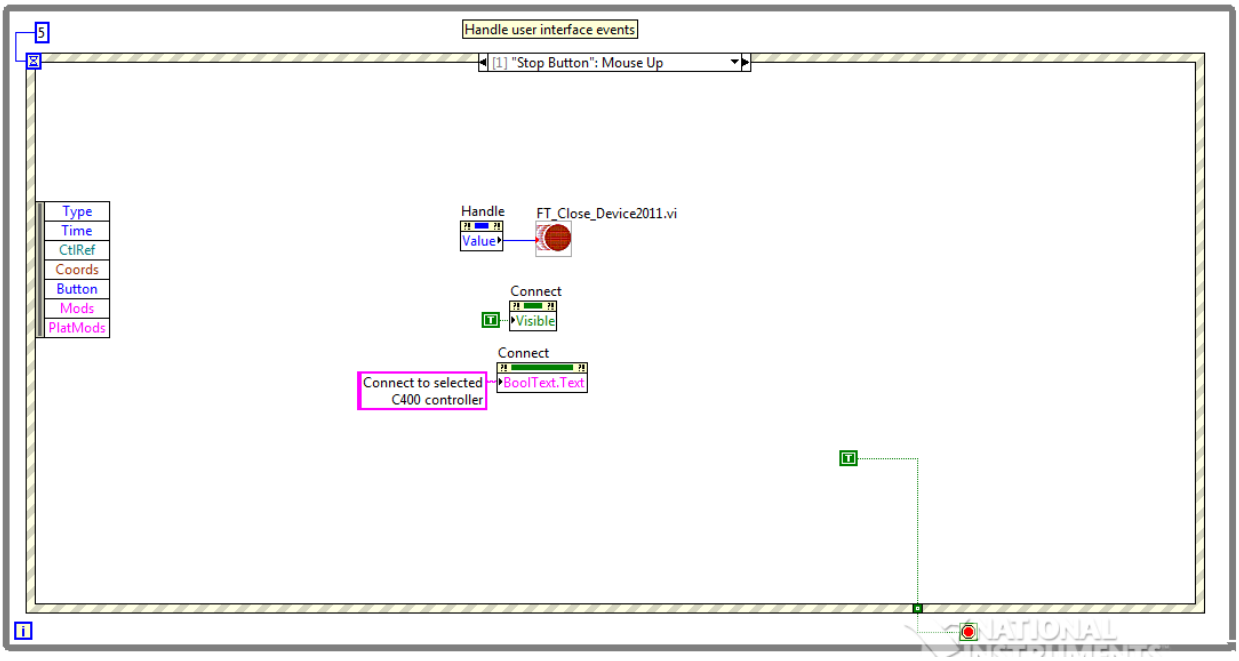


Figure 4 - Stop Button event in C400 Example VI

When the Stop button is clicked, the Stop Button event is triggered. The device connection is closed with the FT_Close_Device2011 VI provided by FTDI and upgraded to Labview 2011. The connect button is made visible, and the default button text is set. A true value is set to stop the while loop from running, ending execution of the C400 Example VI.

2.2.3 Ch1 Position Command Event

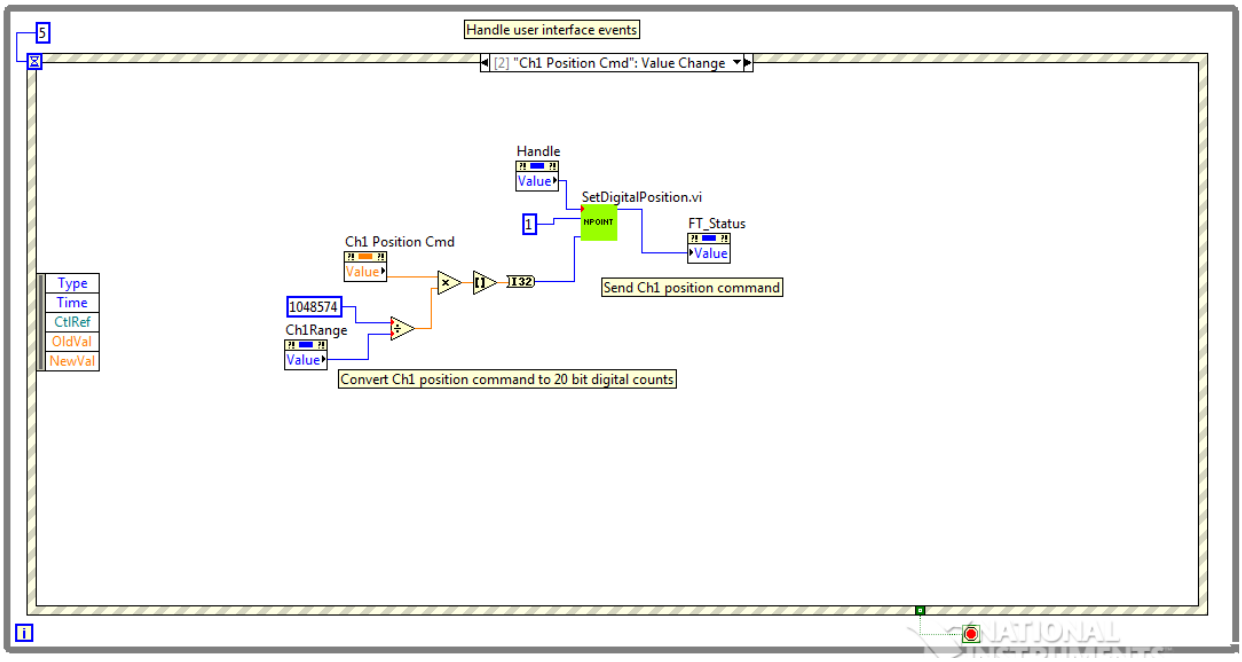


Figure 5 - Ch1 Position Cmd event in C400 Example VI

When the Ch1 Position Cmd numeric control is changed, the Ch1 Position Command event is triggered. The Ch1Range value is used to scale the user input from “range units” (typically microns) to 20 bit controller position counts, and the nPoint SetDigitalPosition VI is called.

2.2.4 Read Sensor Event

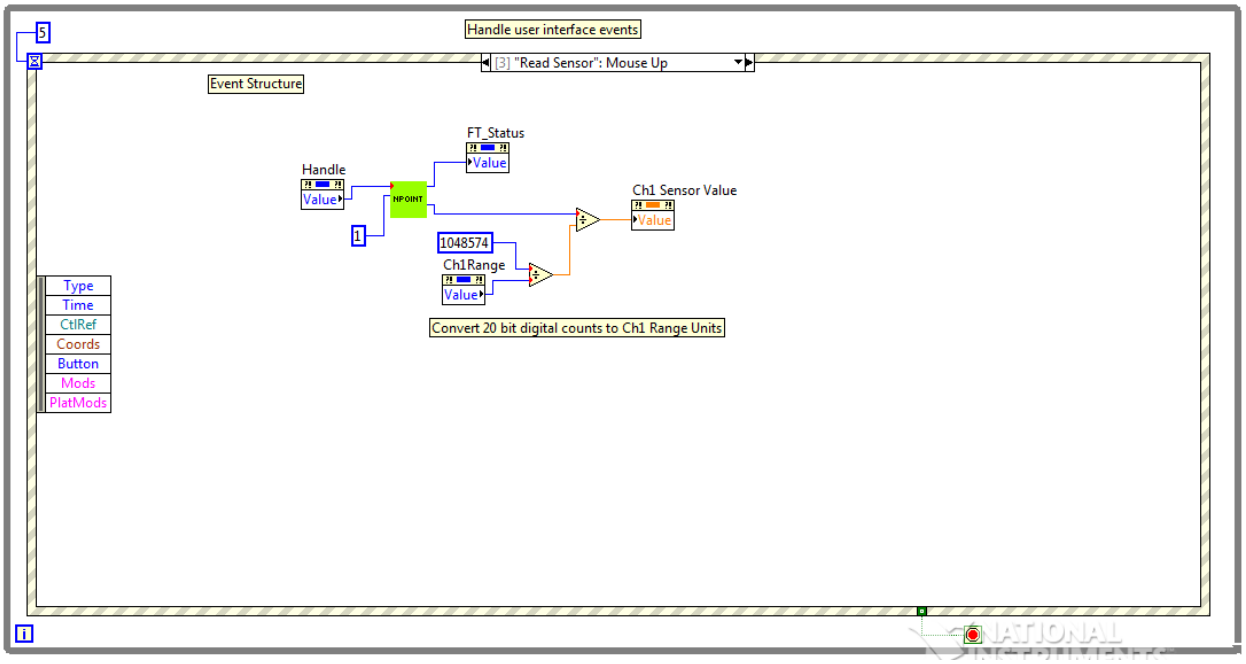


Figure 6 - Read Sensor event in C400 Example VI

When the Read Sensor button is clicked, the Read Sensor event is triggered. The nPoint GetSensorReading VI is called, and the Ch1Range value is used to scale the reading from 20 bit controller position counts (typically microns) to “range units” when displayed to the user via the Ch1 Sensor Value numeric indicator.

2.2.5 Timeout Event

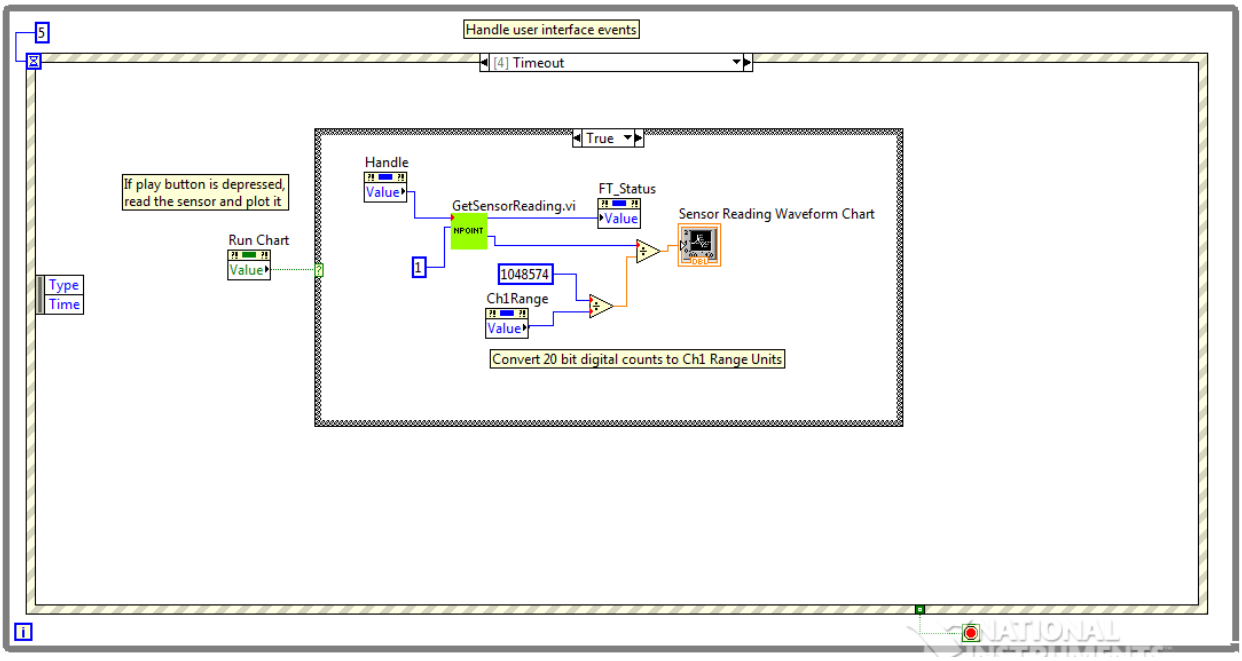


Figure 7 - Timout event in C400 Example VI

The timeout event is set to trigger every 5 milliseconds if no other event is triggered. The value of the Run Chart button is checked, true indicating that the button is depressed. If the Run Chart button is true, the nPoint GetSensorReading VI is called, and the Ch1Range value is used to scale the reading from 20 bit controller position counts (typically microns) to “range units” when displayed to the user via the Sensor Reading Waveform Chart. As values are added to the chart, the chart will update and begin scrolling when the chart area is filled.

2.2.6 Start Waveform Event

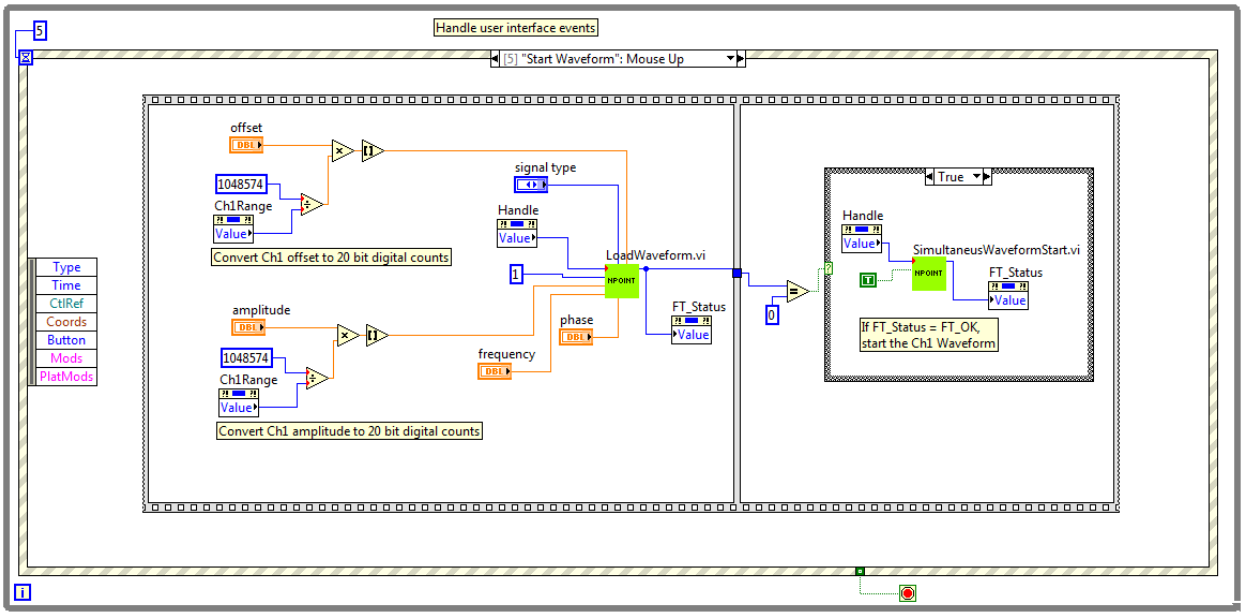


Figure 8 – Start Waveform event in C400 Example VI

When the Start Ch1 Waveform button is clicked, the Start Waveform event is triggered. The waveform parameters are sent to the LoadWaveform VI which writes the waveform points to C400 internal memory buffers. The amplitude and offset inputs for the LoadWaveform VI are scaled to 20 bit digital counts. In the second frame of the sequence the SimultaneousWaveformStart VI is called which will start waveforms for all channels at the same time for maintaining accurate phase relationship.

2.2.7 Stop Waveform Event

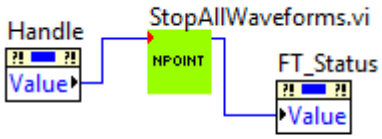
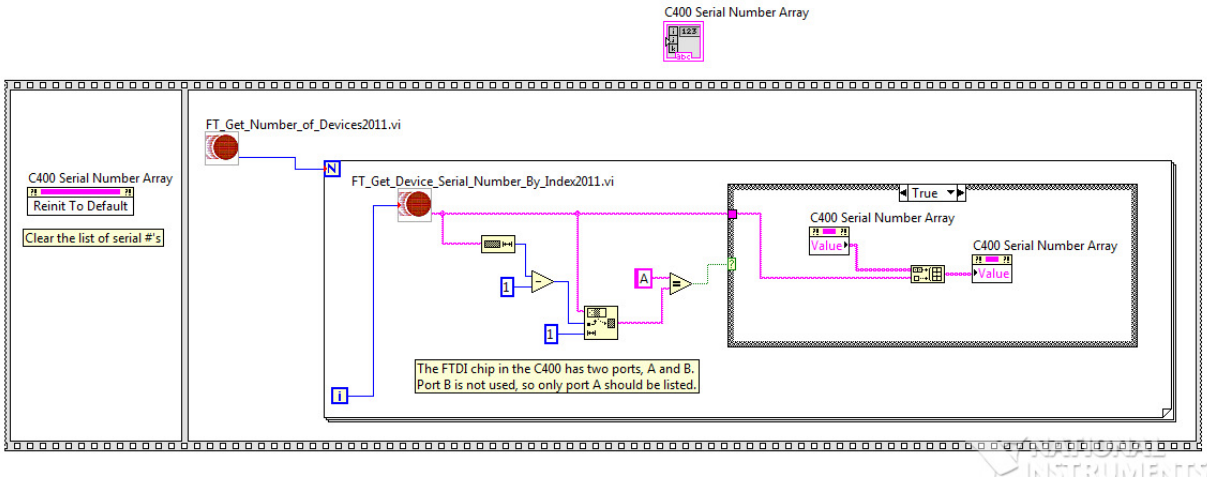


Figure 9 - Stop Waveform event in C400 Example VI

When the Stop Waveform button is clicked, the Stop Waveform event is triggered. The StopAllWAveforms VI is called which will stop the digital waveform for all controller channels.

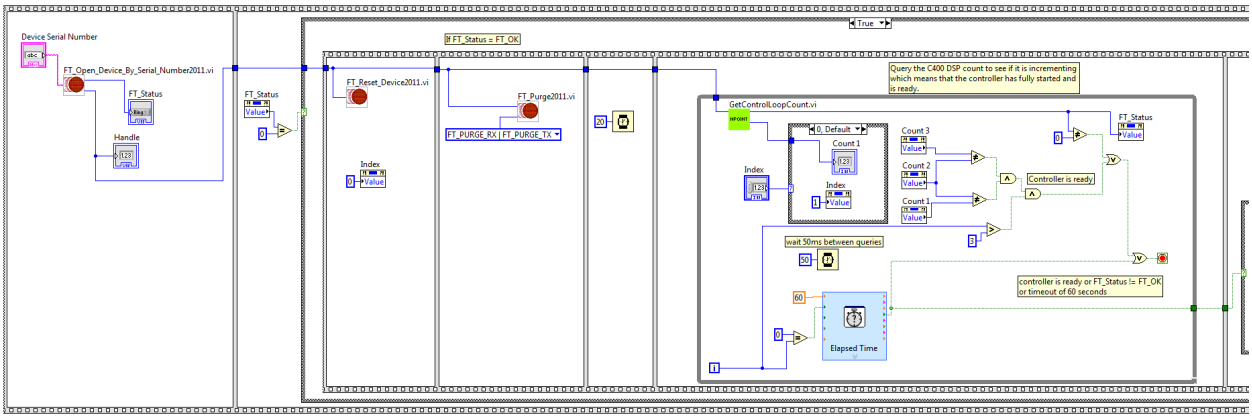
3. nPoint C400 USB Sub VIs

3.1 GetSerialNumberList VI



The GetSerialNumberList VI gets the number of FTDI devices connected to the computer, and then loops through the devices getting the serial number to populate a string array. The C400 controller will show as two devices with the same serial number and an "A" or "B" character appended. The B port of the FTDI chip in the C400 controller is not wired to anything, so only devices with an "A" at the end of the serial number are added to the list.

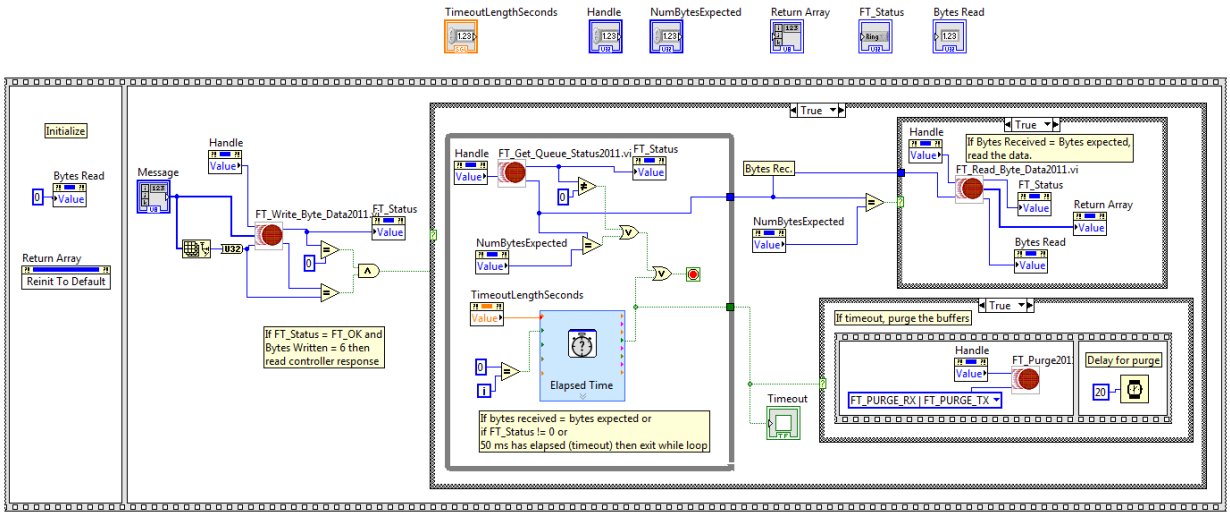
3.2 Connect VI



The connect VI uses a serial number string input to open a connected device. It then resets the device and purges the send-and-receive buffers. After waiting 20ms for the reset and purge to complete, the control loop count is queried until the last three readings are different from each other indicating that the control loop is now running. When the C400 controller is powered on, it needs to go through startup and initialization routines until it is ready for the PC program to read and write controller memory locations. Three control loop count readings are used so that the loop doesn't stop when the DSP initializes the memory value to zero, but is not ready yet. After the control loop is confirmed

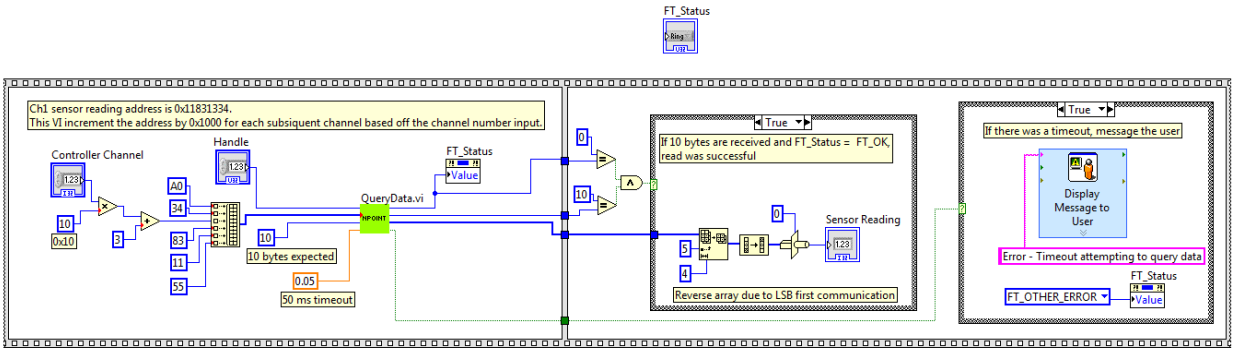
running, the program waits another 500 ms for final steps in the C400 power on sequence. If the process takes longer than 60 seconds, a timeout is issued.

3.3 QueryData VI



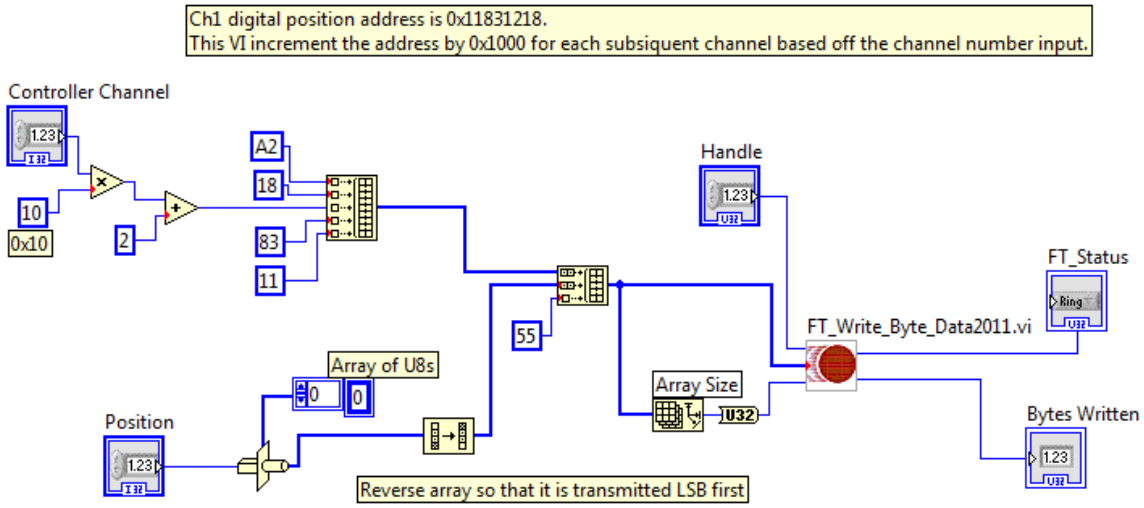
The QueryData VI sends a message input for which the user expects a controller response of a specific number of bytes. The VI loops until the specified number of bytes is received or a specified timeout is reached.

3.4 GetSensorReading VI



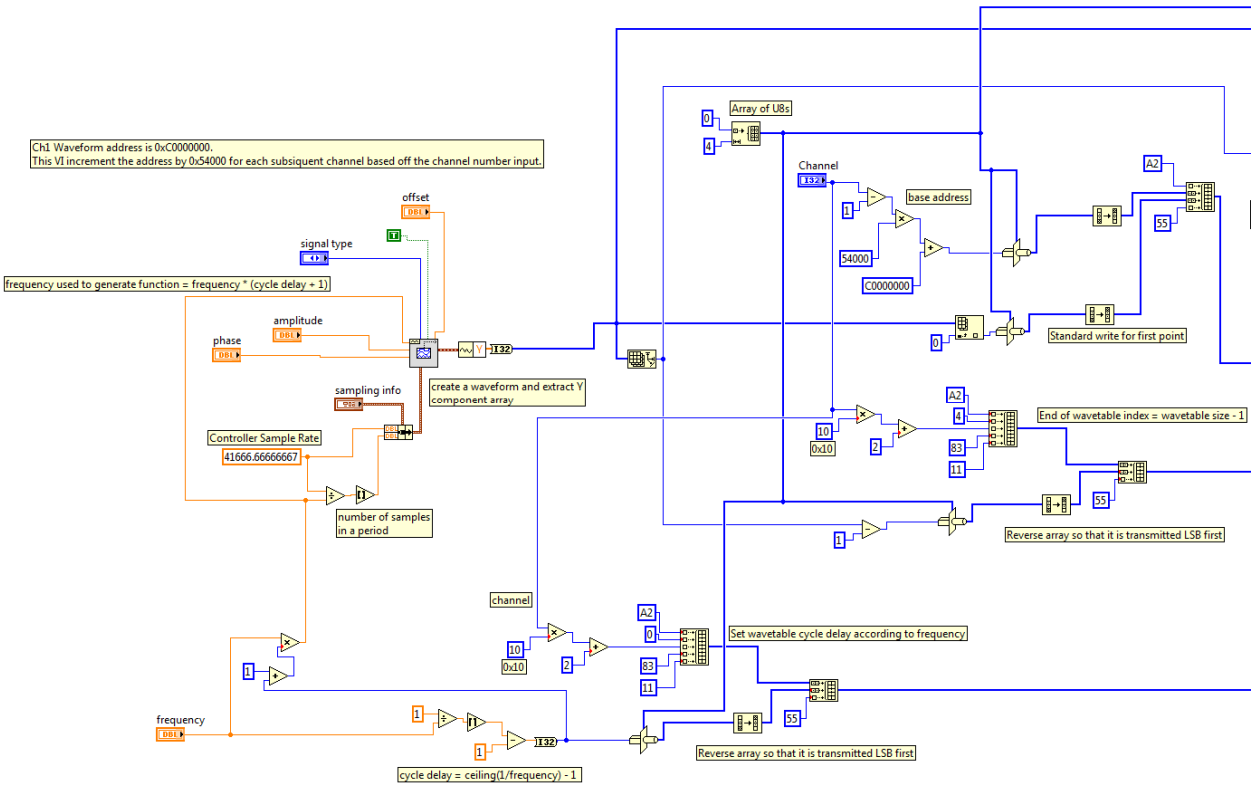
The GetSensorReading VI constructs a byte array according to the C400 communication specification detailed in the document “C400 Advanced Communication.pdf”. The QueryData VI is called and, if the correct number of bytes is received, the return message is cast to the Sensor Reading integer control. If there is a timeout, a message is sent to the user and FT_Status is set to FT_OTHER_ERROR. Other “Get” sub VIs are similar.

3.5 SetDigitalPosition VI

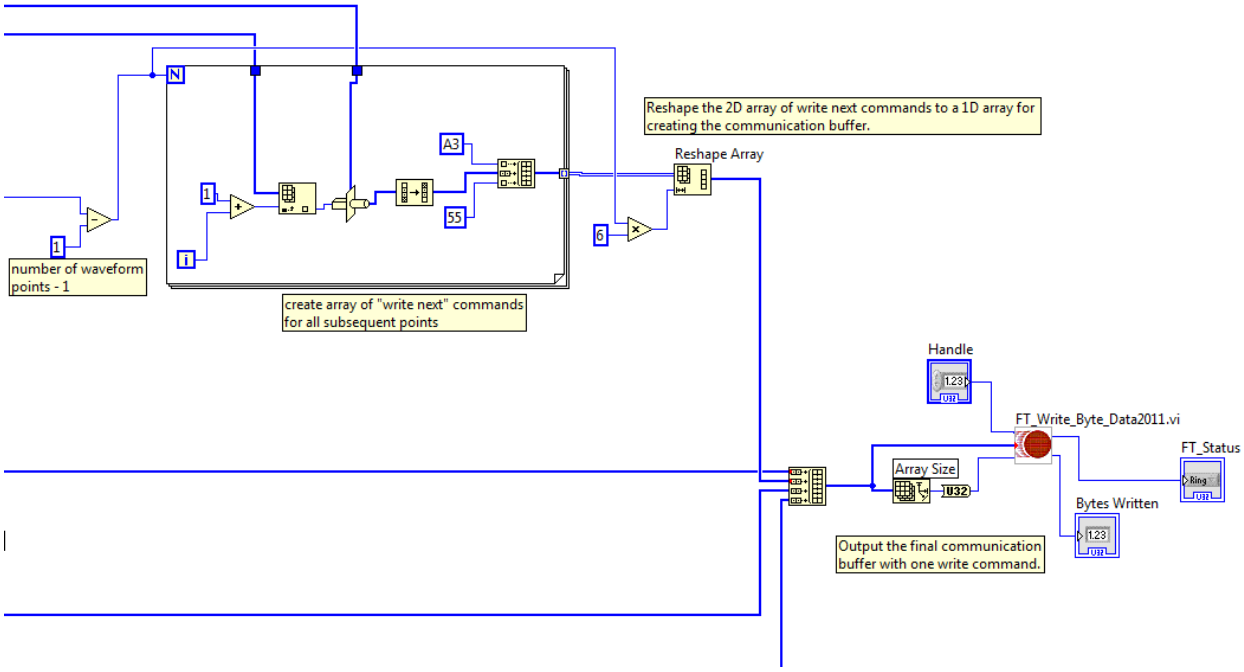


The SetDigitalPosition VI constructs a byte array according to the C400 communication specification detailed in the document “C400 Advanced Communication.pdf”, and sends it to the controller. Other “Set” sub VIs are similar.

3.6 LoadWaveform VI

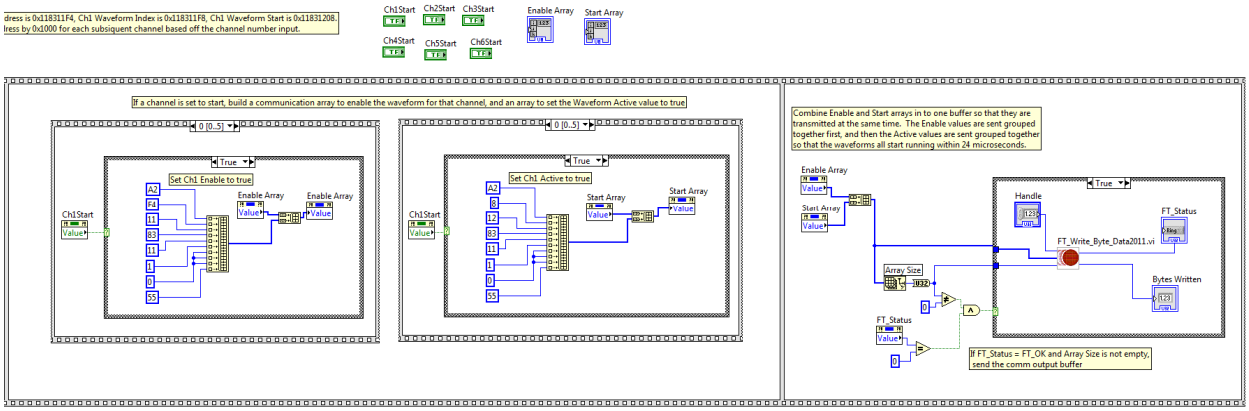


The first part of the LoadWaveform VI creates a waveform point array using the NI Basic Function Generator VI. If the frequency parameter is below 0.5 Hz, the cycle delay parameter is set and the frequency is multiplied accordingly so that the correct number of waveform points are generated. If the frequency parameter is above 0.5Hz, the cycle delay is not needed and frequency is not modified. A standard single write command is created for the end of wavetable index with a value of wavetable size minus one. A standard single write command is created for the first point in the wavetable to set the memory address for the “write next” commands that will be used with all subsequent wavetable points.



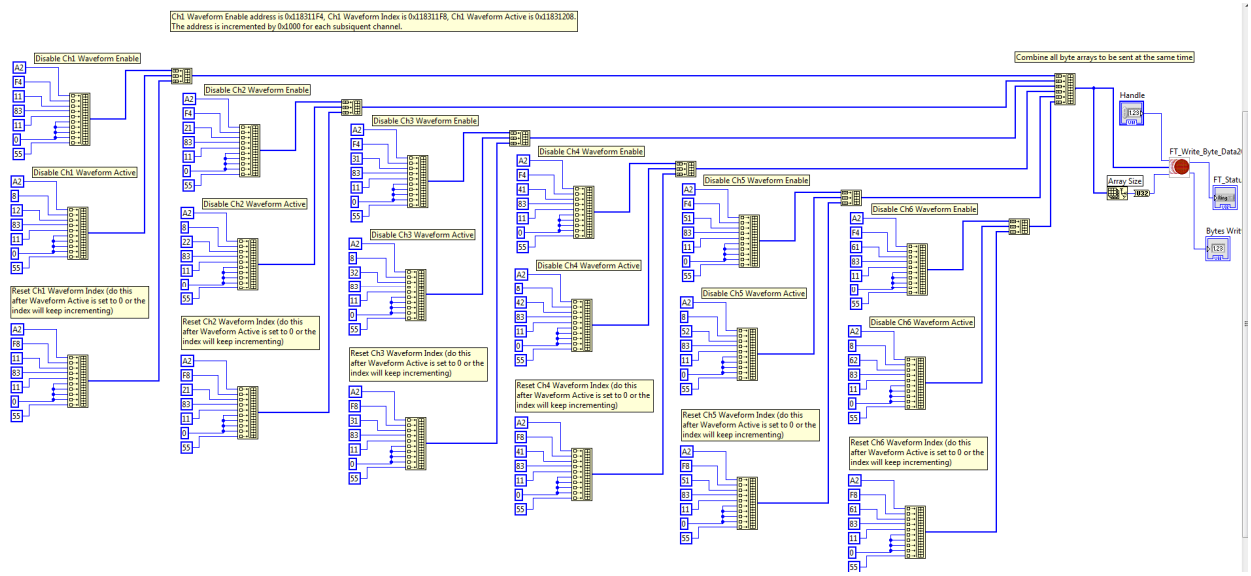
The second part of the LoadWaveform VI creates a 2D array of “write next” commands for wavetable points. Creating a 1D array inside the For loop in Labview 2011 was taking a significant amount of execution time due to memory usage for each iteration of the loop, so the 2D array is reshaped after the For loop completes. Finally, the standard write commands are combined in a single 1D array with the “write next” commands to be sent at the same time with the FT_Write_Byte_Data VI.

3.7 SimultaneousWaveformStart VI



The SimultaneousWaveformStart VI first calls the StopAllWaveforms VI to set wavetable Enable, Active, and Index values to zero for each channel. Then for any channel that has True wired to the start input, a command is added to an array to set the Enable and Active values to 1. Finally, all the commands are combined in a single array and sent to the controller with a single call to the FT_Write_Byte_Data VI.

3.8 StopAllWaveforms VI



The StopAllWaveforms VI creates a write command to set wavetable Enable, Active, and Index values to zero for each channel. The commands are combined in a single array and sent to the controller with a single call to the FT_Write_Byte_Data VI.