Ultrasonics Spectrometer Code Manual

Matthew Rothfuss*

Department of Animal Science and Food Industry, Kansas State University

May 10, 2015

*mrengr@ksu.edu, mrengr@phys.ksu.edu

Contents

Chapter 1

LabView Basics

1.1 Introduction

LabVIEW (short for **Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) is a development environment for visual programming, developed by National Instruments (www.ni.com). The code files (or program files) are identified by the **.vi** extension called **Virtual Instruments** or **VIs** for short. This graphical language is most commonly used for data acquisition, instrument control, signal processing (analysis), industrial automation, and more.

The next section will cover some basics of LabVIEW design and operation. For additional resources, the current (2013) LabVIEW Getting Started Manual is located here.

1.1.1 Additional Resources

[1]

Chapter 2

Theory of Operation

Define the background concepts of how/what this program is accomplishing. Make refs to papers but don't do the math here (don't have time for that). Just outline the basics of what we want to do, what goes into the system, what the system does (ref manuals and such for theory & papers), and what the system outputs.

Chapter 3

Code Structure

Theory of code operation goes here. ie case structure, state machine,

3.1 Main VI

Define the outline of the Main VI (the main program) and hit on each part of it. Don't spend time explaining the subvi's here since i'm doing that in the **Custom VI's** section. Make sure to to be thorough on all the code that is not included in the subvi section.

The main program **ASUDS_v13.vi** is contained within a Project file called (file name here). When the program is ran it will first load the auto generated system settings files (via the LC931C_Read.vi [3.2.1] or DPR300_Read.vi) [3.2]) from the last time the program ran. A set of default system settings files are included for first time use. Next the Oscilloscope and the JSR are both initialized via LC931C_Int.vi and DPR300_Int.vi [3.1, 3.2] with the loaded settings as-well-as set the front panel controls to the loaded settings. At the same time any leftover front panel controls are set to their default values and all block diagram cases are set to there initial positions.

Once the settings clusters have been set properly and

3.2 Custom VI's

List of custom VI's and a short description of what they do. In the next section we will take a deeper look into each of these subvi's.

These files load in most of the front panel controls, some are missing since more code has been added but the vi that saves the system settings has not been updated to include the new additions.

3.2.1 Oscilloscope VIs

LC931C_Read.vi

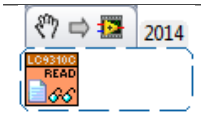
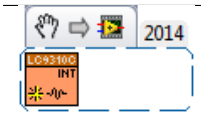
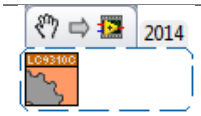
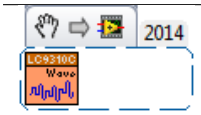
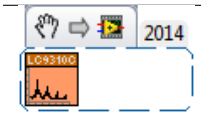
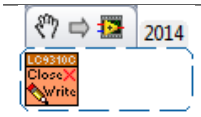
Oscilloscope		
VI	File Name	Description
	LC931C_Read.vi	Load Oscilloscope Setting from System Generated File
	LC931C_Int.vi	Initialize Oscilloscope Settings
	LC931C_settings.vi	Apply Settings to Oscilloscope
	LC931C_single-wave-output.vi	Acquire Single Wave from Oscilloscope and Average
	LC931C_norm-pad-hilbert.vi	Oscilloscope Tab Settings
	LC931C-Config-Write-Close.vi	Write Oscilloscope settings to System File and close Oscilloscope resources

Table 3.1: Oscilloscope Custom VI's

The LC931C_Read.vi reads the LeCroy 9310C oscillosope settings from file and loads the values into a **LeCroy 9310C Settings** cluster. The settings folder (**System_Settings**) is located in the root directory of the main VI. This VI requires the MGI Library. Figure (3.1) is the block diagram. It is setup as an error case structure. When an error is detected from the *error (in)* input then the code in the green box does not execute.

The for loop steps through each list section of the ".ini" file. Each list section corresponds to one of the input cluster constants *(LC930x_TimeBase, LC930x_Vertical_Setup, LC930x_Trigger_Edge_Setup, LC930x_Read_Wave)*.

3.3 Operation

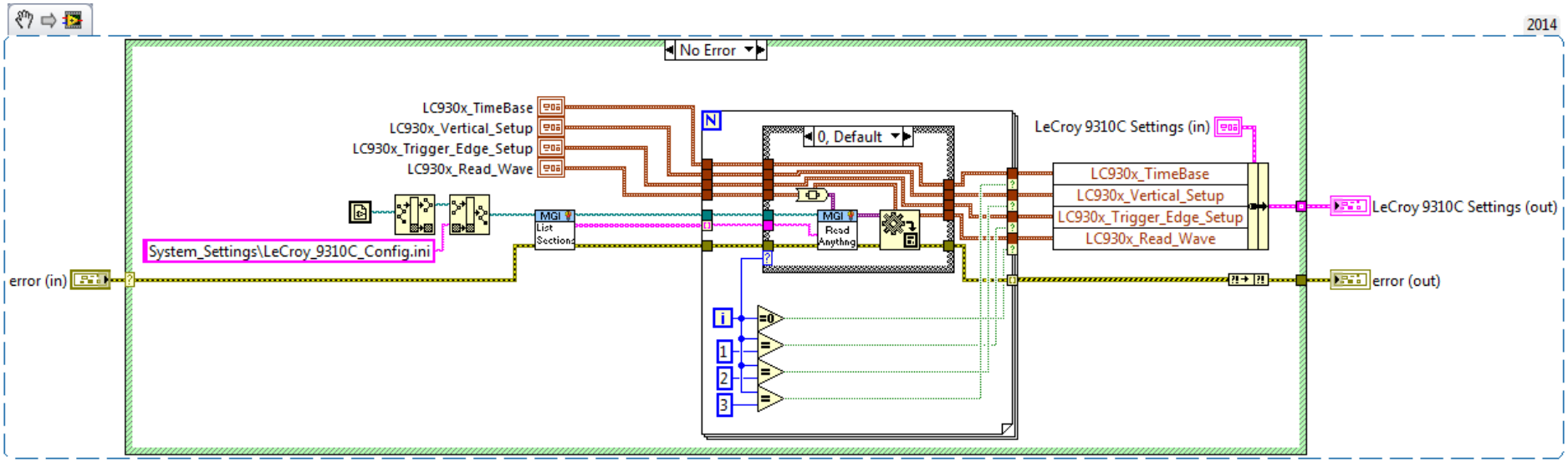


Figure 3.1: LC9310C_Read.vi

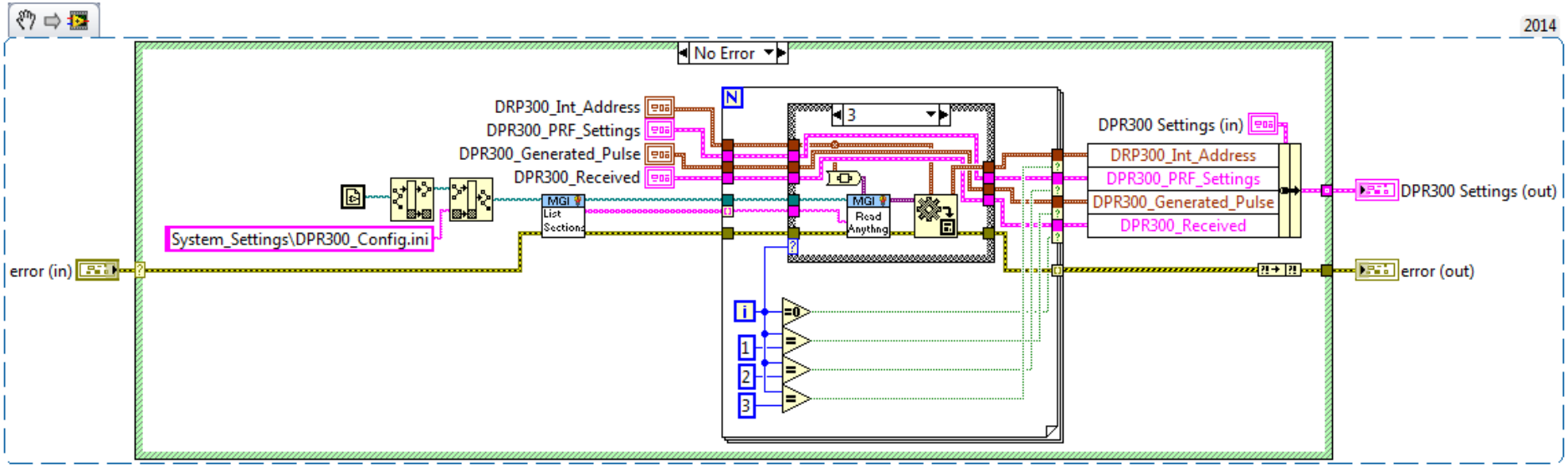


Figure 3.2: DPR300_Read.vi

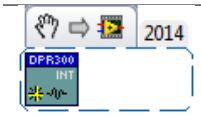
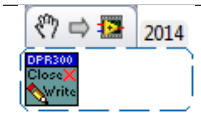
JSR Pulser/Receiver		
VI	File Name	Description
	DPR300_Read.vi	Load Pulser/Receiver Setting from System Generated File
	DPR300_Int.vi	Initialize Pulser/Receiver Settings
	DPR300_settings.vi	Apply Settings to Pulser/Receiver
	DPR300-Config-Write-Close.vi	Write Pulser/Receiver settings to System File and close Pulser/Receiver resources

Table 3.2: JSR Pulser/Receiver Custom VI's

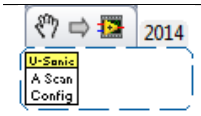
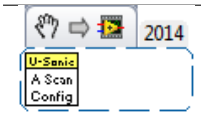
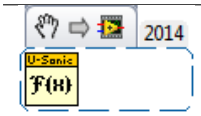
Ultrasonic Package		
VI	File Name	Description
	USonic-A-Scan-Config-edit.vi	Configure/Set Gates for Data Acquisition
	USonic-Gates-edit.vi	Pull Out Relevant Data from Gates for Data Acquisition
	USonic-FFT.vi	Process Gate For Quick Analysis

Table 3.3: Ultrasonic A-Scan Customized Package VI's

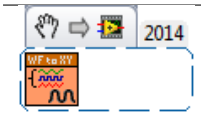
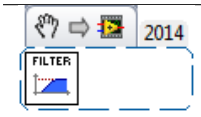
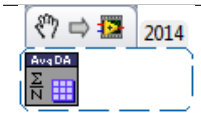
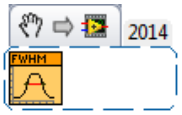
Math		
VI	File Name	Description
	Waveform-to-XY-Array.vi	Convert Waveform to XY-Array
	Filter_signal.vi	Filter Wave Signal for Oscilloscope Tab (does not affect Data Acquisition)
	Average-Dynamic-Array.vi	Take the Average of N elements in a Dynamic Array
	FWHM-Poly.vi	Compute the Full Width Half Max (FWHM) of either a Waveform, XY-Graph, or Waveform cluster

Table 3.4: Custom Math VI's

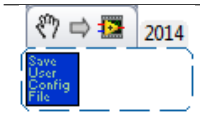
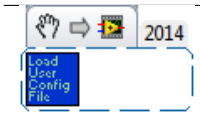
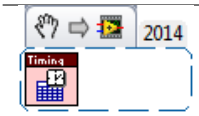
Miscellaneous		
VI	File Name	Description
	Save-User-Config-File.vi	Save all front panel controls to a user.ini settings file
	Load-User-Config-File.vi	Load the user.ini settings file
	Time-Data.vi	Load and Save Data Timing table
	Instrument-error-handler.vi	Pop-up error message for loss of Instrument signal

Table 3.5: Miscellaneous Custom VI's

Bibliography

[1] Tomás E. Gómez and Álvarez-Árenas. Air-coupled ultrasonic spectroscopy for the study of membrane filters. *Journal of Membrane Science*, 213(1-2):195–207, March 2003.