Boonton IVI Driver Files in LabVIEW

Introduction

The Boonton 55 Series Wideband USB Peak Power sensor comes with an installation software package which includes an API (Application Programming Interface) for remote programming. The API is designed to allow considerable programming flexibility for implementing user features by using Boonton IVI-COM and IVI-C driver files. This application note explains the benefits and features of IVI drivers and shows how to use the properties and methods they contain in a LabVIEW program as an example.

IVI Benefits and Features

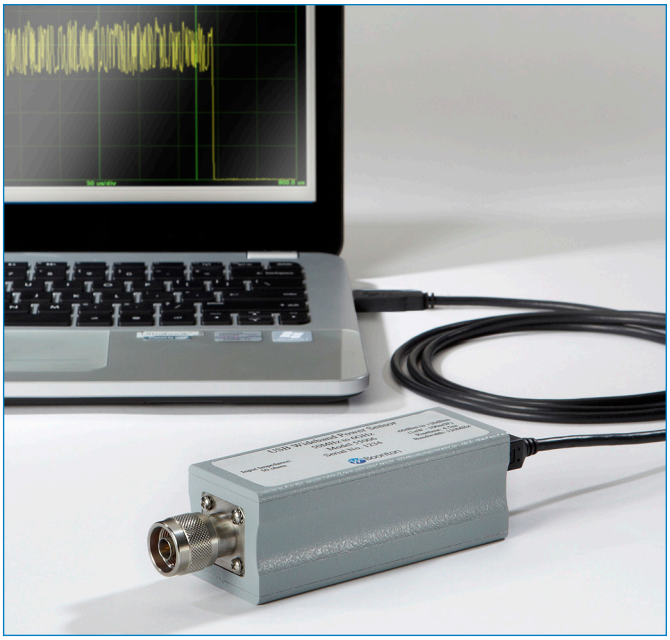
To support all popular programming languages and development environments, IVI drivers provide either an IVI-C or an IVI-COM (Component Object Model) API. Driver developers may provide either or both interfaces, as well as wrapper interfaces optimized for specific development environments.

Although the functionality is the same, IVI-C drivers are optimized for use in ANSI C development environments, while IVI-COM drivers are optimized for environments that support the Component Object Model (COM). IVI-C drivers extend the VXIplug&play driver specification and their usage is similar. IVI-COM drivers provide easy hierarchical access to instrument functionality through methods and properties. Boonton IVI driver files are compatible with LabVIEW Software including other programming language like MatLab, C Sharp, C++ and Visual Basic (VB.NET).

All IVI drivers communicate with the instrument through a low-level device driver, but it is not necessary for users to interface at the device level. Our example uses the LabVIEW, a widely used software program with standard library for communicating with instruments from a personal computer. Please note that 55 Series USB sensor is not a VISA device. Hence users need to import Boonton IVI driver files into LabVIEW Program and create LabVIEW Library files then use our provided functions and attributes to develop their own LabVIEW driver file.

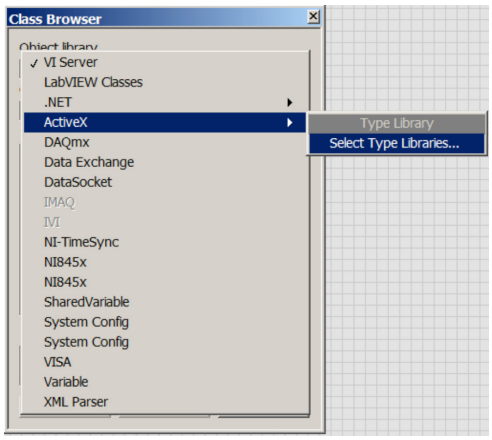
Advantages of IVI-COM driver files in LabVIEW:

- Readability of code using Invoke Nodes and Property Nodes is superior to that of wrapped driver VIs.
- All driver functionality, methods and properties, is accessible from the Class Browser.
- Navigation of driver functionality is clearer and more convenient in the Class Browser than in the Function palette.
- Property access is very easy to navigate from the block diagram.

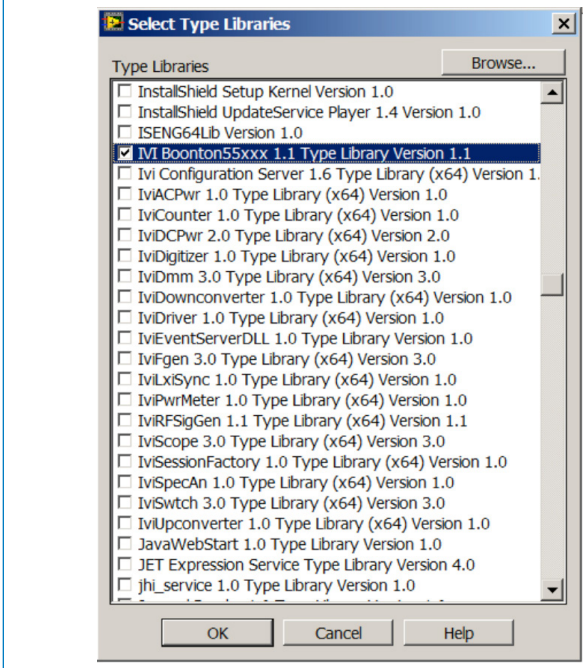


How to use Class Browser for accessing Boonton 55xxx Library

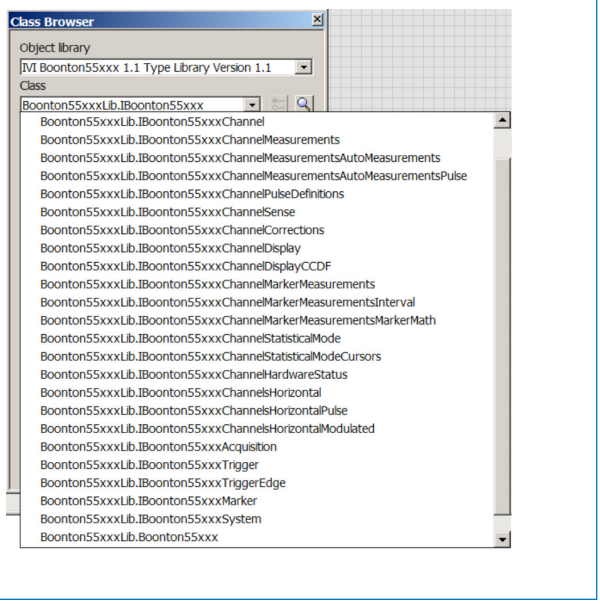
To select Boonton 55xxx IVI library files you can select ''Class Browser'' from View menu of LabVIEW software. Then follow the sequences shown below in a Block Diagram window of new VI in order to access related properties and methods:



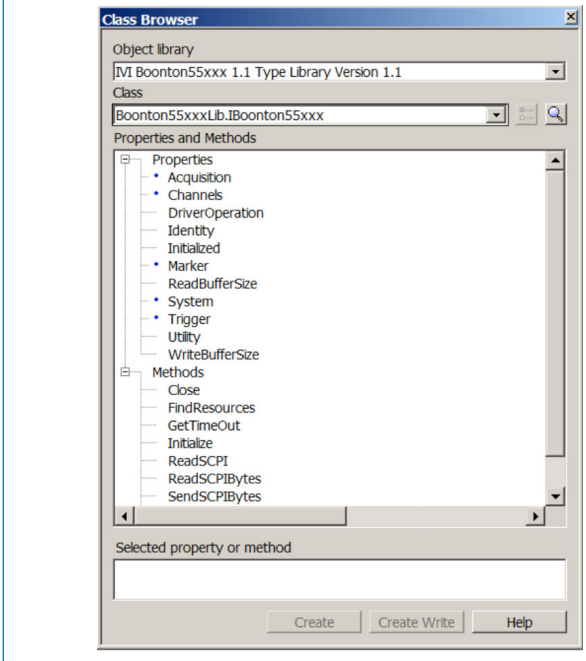
Step 1: From Class Brower window select Object Library > ActiveX and then ''Select Type Libraries''



Step 2: A new window will show up with a list of library files names. Select IVI Boonton55xxx Type Library from that list and click OK button.



Step 3: Now you will see a list of available interfaces under Boonton55xxx Object Library. Select any preferred interface name from the list.



Step 4: Once you select any specific class, you will find a list of properties and methods to be used with your preferred measurement mode

Boonton IVI-COM drivers: Properties and Methods

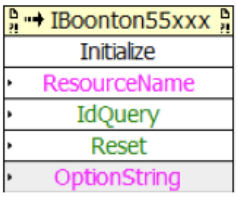
To use Boonton IVI-COM drivers in LabVIEW you will need to use the ActiveX functions and the Class Browser that are built-in to LabVIEW. Once you select new VI from LabVIEW, Front Panel and Block diagram windows will appear. By right clicking on Block diagram, you can select Function Palette as explained earlier in the above section. Now select the Connectivity sub-palette and then the ActiveX subpallete. From this palette, you can access ActiveX and COM objects including all Boonton IVI-COM drivers.

To write LabVIEW applications with an IVI-COM driver, you simply select a method or property from the Class Browser and drag it on to the diagram, as in the image below. Here are some example properties and methods used for the following LabVIEW example program.

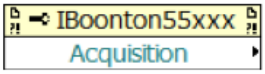
The Class Browser automatically creates a LabVIEW Invoke Node associated with the selected IVI-COM driver's Initialize method (for example). The Invoke Node is well designed, intuitive to use, and understandable when viewed on a real diagram. The Node clearly indicates the IVI-COM parent interface name (IBoonton55xxx), the method name (Initialize), and each of the input parameters (ResourceName, IdQuery, Reset, OptionString). If the method had any output parameters, these also would be shown.

Each parameter is conveniently color-coded to indicate the data type. For the Initialize node, the ResourceName and OptionString parameters are pink, indicating they are strings. Boolean parameters are indicated by the color green. The OptionString parameter has a slight but easily discernable gray background indicating that this parameter is optional.

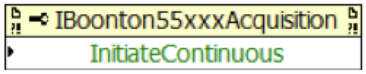
FindResources method returns a delimited string of available resources. which can be used in the initialize function to open a session to an instrument.



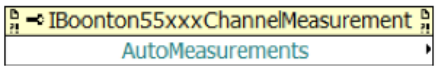
Initialize method is used to open the I/O session to the instrument. Several parameters used - ResourceName. IdQuery, Reset and optionString.



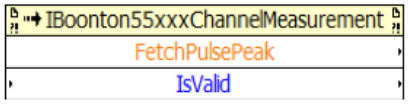
Acquisition function is used for Boonton USB sensor acquisition



InitiateContinuous property is used to set or return the data acquisition mode for single or free-run measurements.



AutoMeasurements property is used to make a reference to the IBoonton55xxx Measurements-AutoMeasurements interface.



FetchPulsePeak method is used to returns the peak amplitude during the pulse.

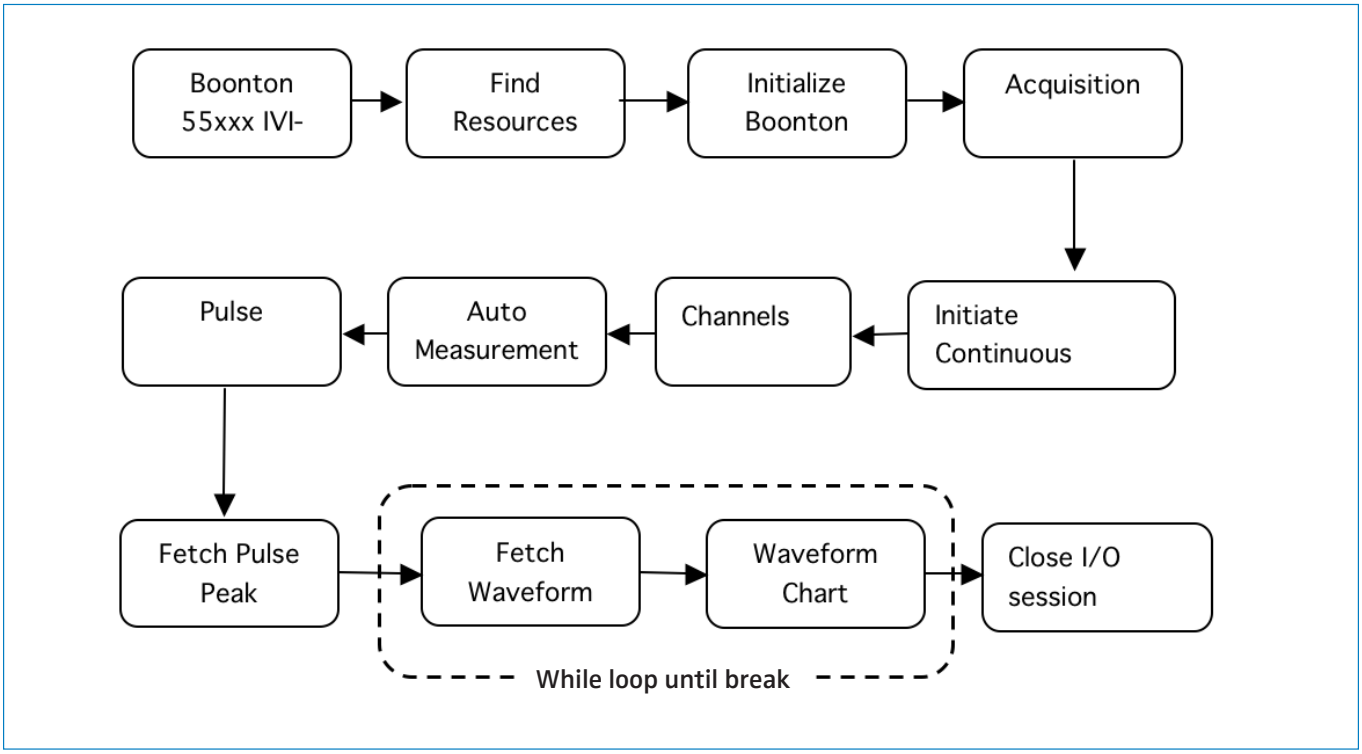
Example Requirements

Below are the minimum requirements for this example LabVIEW program:

- LabVIEW version 8 or later.
- Boonton 55xxx IVI driver files (installed with Boonton Peak Power Analyzer Suite software)
- IVI Shared Components (installed with Boonton Peak Power Analyzer Suite software)
- Boonton USB driver (installed with Boonton Peak Power Analyzer Suite software)
- Boonton 55 series Wideband USB Peak Power Sensor.

Understanding Example Program

The example LabVIEW program for Boonton 55 series USB sensor we provide can be explained by using a simple block diagram. Below is a sequence of some related Boonton IVI-COM properties or methods to understand how they are used in LabVIEW.



Block diagram for understanding LabVIEW example program.

Example Program in LabVIEW

LabVIEW code for accessing an IVI-C driver's attributes is identical to that shown above for IVI-COM. Property nodes are used in precisely the same fashion for both flavors of IVI drivers. Since real-world drivers consist primarily of properties, LabVIEW applications built on IVI-COM drivers will look very close to their IVI-C counterparts. As an example, the code below shows the same simple task of configuring an RF Pulse signal using an an IVI-COM driver.

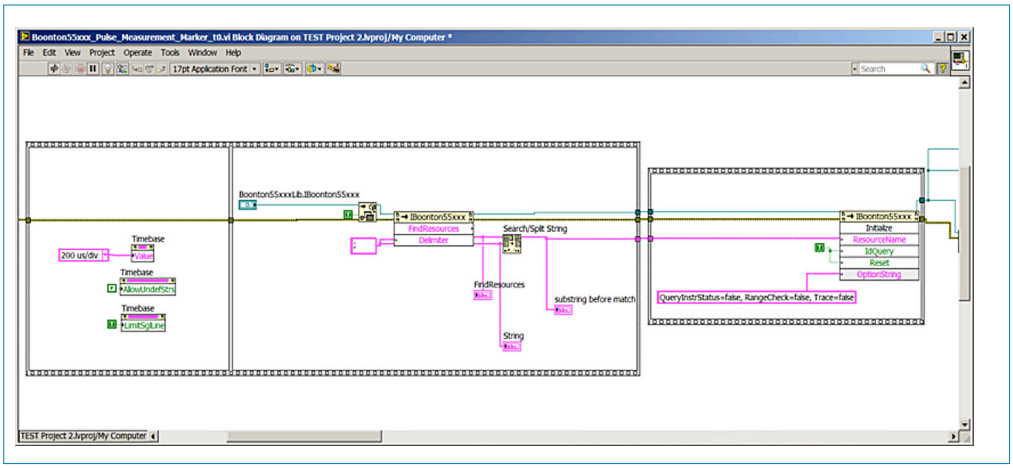


Figure 1. Screenshot of LabVIEW example for Pulse measurement.

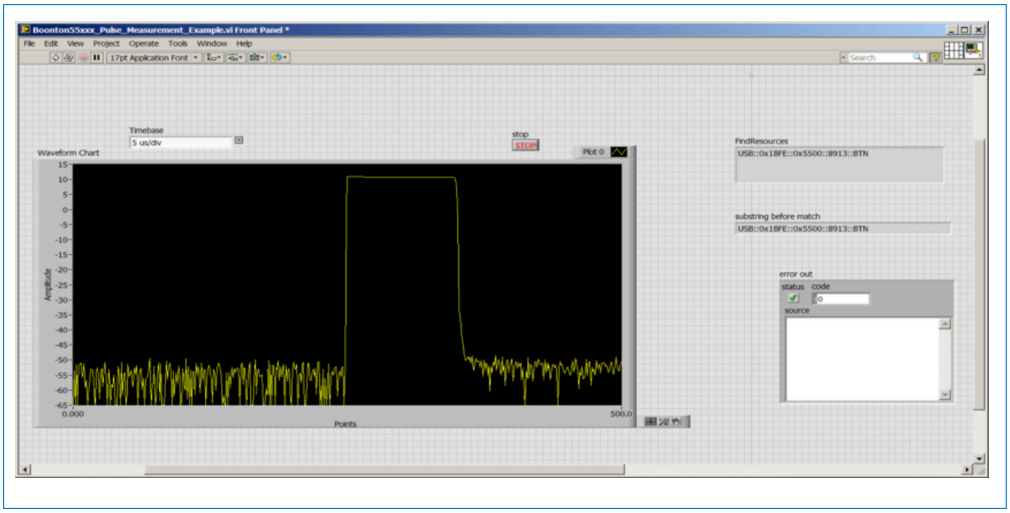


Figure 2. Output of the example LabVIEW program.

Resources

[1] Example LabView file for Boonton 55 Series USB Sensor

[2] Boonton 55 Series API Programming Reference

[3] Comparing IVI-COM and IVI-C by Pacific Mindworks

[4] Getting Started with IVI Drivers by IVI Foundation

Wireless Telecom Group
Boonton Microlab Noisecom

Wireless Telecom Group Inc.
25 Eastmans Rd
Parsippany, NJ
United States
Tel: +1 973 386 9696
Fax: +1 973 386 9191
www.boonton.com

Follow us on:

WTGinnovation

Wireless Telecom Group

blog.wtcom.com

WTGinnovation

B/WP/1214/EN