Controlling PlutoSDR

At this point you should have PlutoSDR connected with the latest firmware installed on the device. As a prerequisite for this workshop you should have installed the libIIO drivers, if you have not done so go the [libIIO Release page] and install the latest drivers.

To make sure we can see and connect to PlutoSDR open a terminal or command prompt. Type “iio_info -s” to scan for devices. You should see an output similar to Figure 1.

Figure 1

Your device should have a unique serial number and URI. The URI is specific to the interface, here it is displayed as [usb;X.X.X], and can be used to interface with a specific device when required. When you have a network based device, like a FMComms board, the URI will be associated with an IP address such as: [ip:XXX.XXX.XXX.XXX]. If you do not observe a device, make sure PlutoSDR is connected to the correct USB port and has a one solid LED and one flashing LED. “iio_info” is a useful tool for identifying devices locally connect to connected on the network as well.

Another useful libIIO command line tool is “iio_attr”, which can be used to inspect, as well as read and write attributes. The help for this tool is shown in Figure 2, which outlines that ““iio_attr” can be used access different types of attributes including: device, channel, debug, buffer, and context.

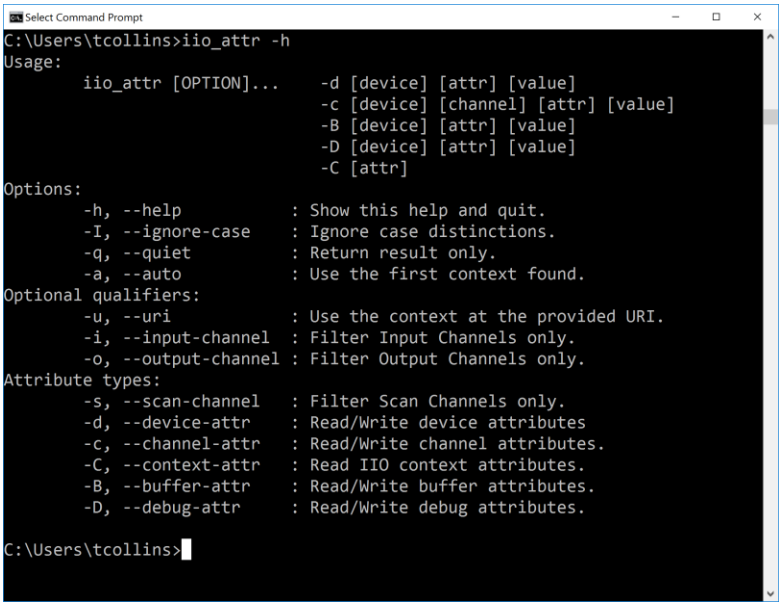

Figure 2

The first step when using “iio_attr” is to select a device to inspect. Since PlutoSDR is connected locally we can use the “auto” context feature and list the devices using the command “iio_attr -a -d”. Use this command to list your devices as shown in Figure 3.

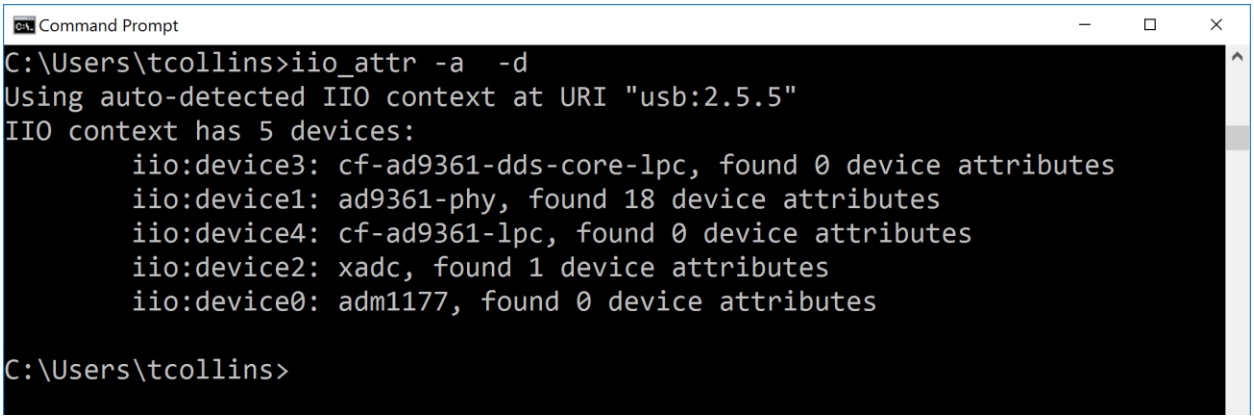

Figure 3

This command also prints some context information. Alternative, try using the “uri” tag to address your PlutoSDR. For the case in Figure 3, the new command would be “iio_attr -u usb:2.5.5 -d”. Construct this command and run it yourself to check if the devices match.

Next, we will examine some of the channel attributes for a given device. Run the command “iio_attr -a -c ad9361-phy”, as done in Figure 4, which lists the channels associated with the PHY driver. These channels are related to certain aspects of the transceiver like TX, RX, temp sensors, and even some custom filters in the FPGA data path.

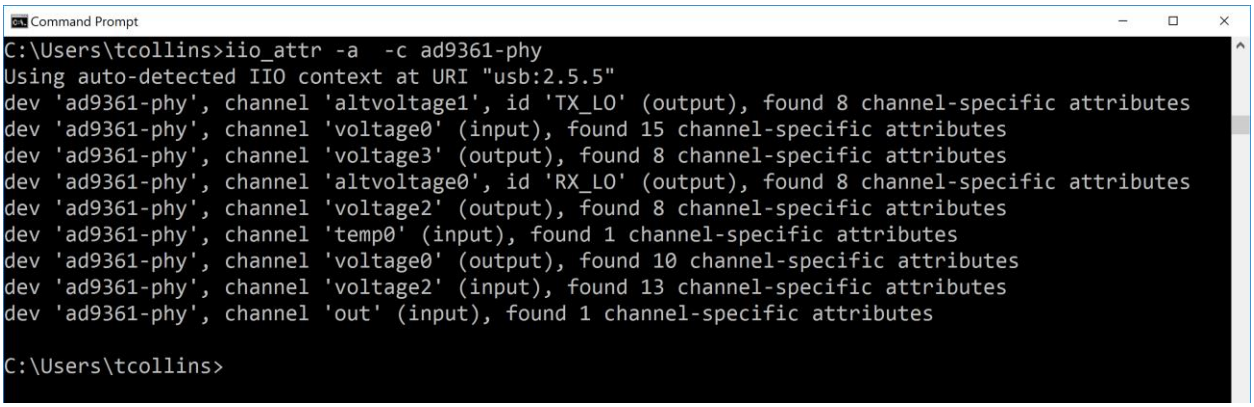

Figure 4

By selecting one of these channels, in this case voltage0 which is the real channel names for both TX and RX, we can view some of the attributes related to those channels. For reference, input channels are with respect to RX and output channels are with respect to TX. To list these attributes run the command “iio_attr -a -c ad9361-phy voltage0”.

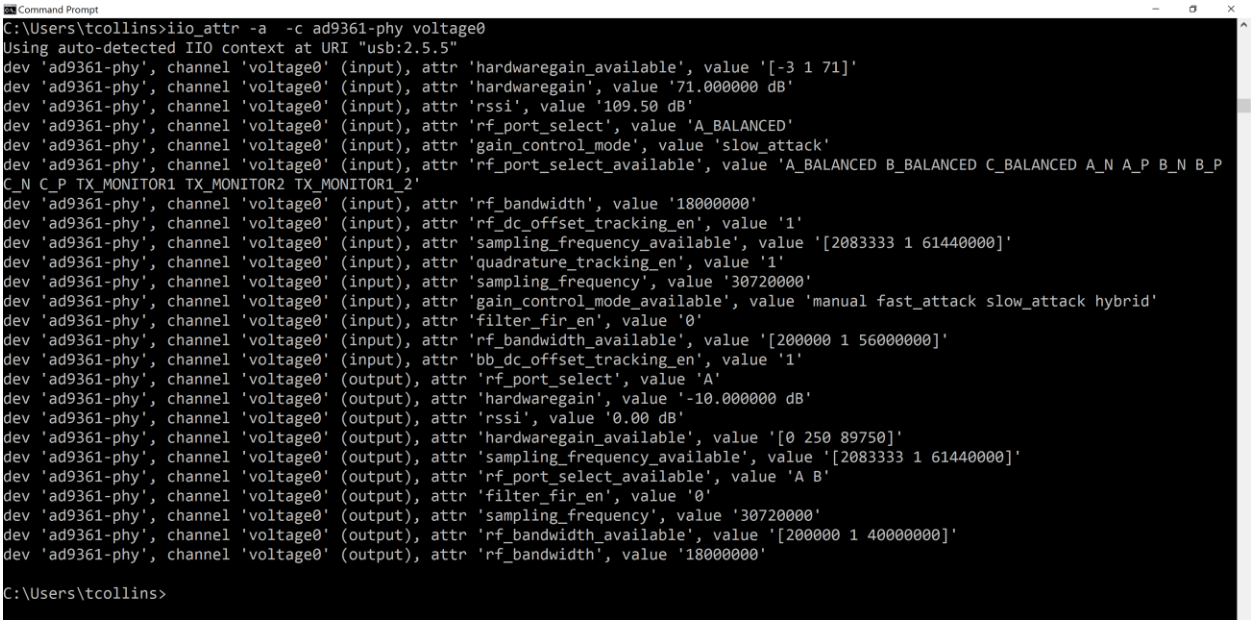

Figure 5

To filter out one of these attributes simply add it to the end of the last used command. For example, inspecting the hardwaregain would be called as “iio_attr -a -c ad9361-phy voltage0 hardwaregain”, as shown in Figure 6.

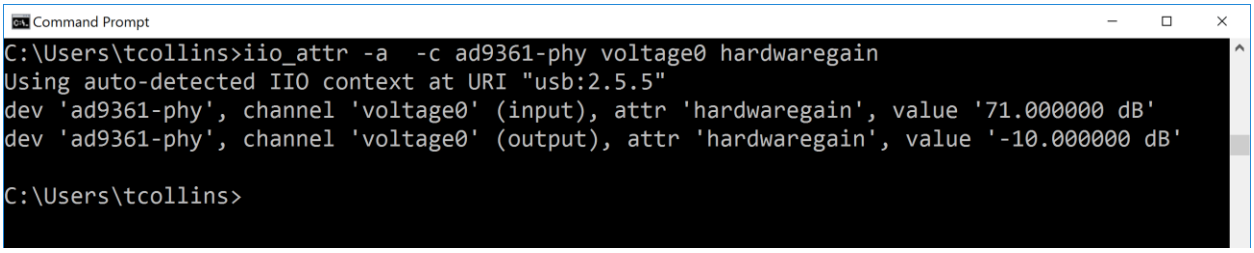

Figure 6

Next, we want to only view output signals (those related to the transmitter), which we can do by adding the output filter tag “iio_attr -a -o -c ad9361-phy voltage0 hardwaregain”.

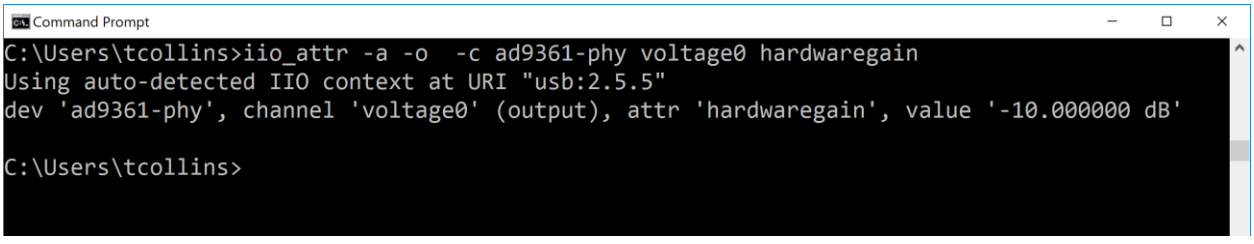

Figure 7

Finally, lets actually write to the attribute by adding a value to the end of the last command “iio_attr -a -o -c ad9361-phy voltage0 hardwaregain -30”. When values are written, they are first always read from beforehand, and then read from after the update is applied. If the value supplied is invalid, you will get an error.

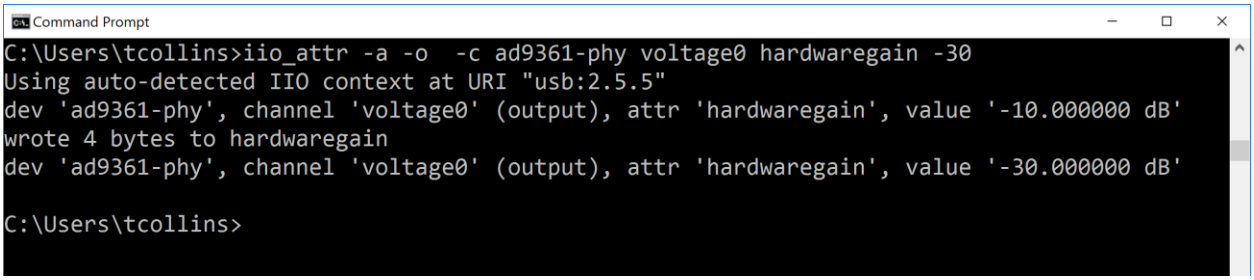

Figure 8

Going through the context, to the devices, to their channels, and finally to channel attributes, you can see how the driver controlling PlutoSDR is structed. We will build off of these ideas in following labs, but first we will do something more interesting.

Using the command line alone is pretty slow, non-intuitive, and repetitive. You can write your own applications, like “iio_attr” in C, or in other languages like Python or MATLAB. However, there exists a specific GUI application for interacting with IIO devices called IIO-Oscilloscope (OSC), which is may more civilized than poking around at attributes through “iio_attr” all day.