

This folder contains the source files for the Vertilon Drivers adapted for the Linux operating system developed using Visual Studio Code. To execute the program you must first install the hidapi library. From there, check the Makefile which is set up to compile and run the program, either with root access or without. If you want to execute the drivers without root access, you will need to create a udev rule.

To make a udev rule allowing your operating system to identify a PhotoniQ device, first execute the command `sudo nano /etc/udev/rules.d/99-Vertilon-HID.rules` in a terminal with root access. Then navigate to the file you just created and open it with a text editor. Add to the empty file `"SUBSYSTEM=="usb", ATTRS{idVendor}=="0925", ATTRS{idProduct}=="0480", MODE="0666"` and then save the file which may prompt you to input an admin password. From there, execute the commands `sudo udevadm control --reload-rules` followed by `sudo udevadm trigger` in the terminal to reload and activate the udev rule.

There is a chance that your PhotoniQ device will not share the same vendor ID or product ID. If the above instructions don't work, first try executing the command `lsusb`, while the hardware is connected through a USB port, in the terminal. This command will output a list of devices, their vendor, and product IDs connected through the USB of your computer. Find the device labeled "Lakeview Research PhotoniQ" and once you do, you should also see the device's vendor and product ID. If they are different to the ones given above in the line `"SUBSYSTEM=="usb", ATTRS{idVendor}=="0925", ATTRS{idProduct}=="0480", MODE="0666"`, then follow the instructions above again but swap the IDs out with the ones from the `lsusb` command.

The drivers include six pairs of header and source code files: main, Initialize, Control Interface, Data Interface, Close, and HID Globals. An interface to a PhotoniQ device can be created in the Initialization process and commands can be sent to a device via the control interface. The library used to perform IO is the hidapi library which is implemented based on the libusb library.

The Initialize files initializes and validates an interface with the PhotoniQ. The Control Interface executes a control operation to a previously initialized PhotoniQ. Many helper functions are included to make sending control operations as easy as possible and documentation is included to explain what each command does and what arguments are required.

The Data Interface is used to signal to the PhotoniQ to generate reports which are then parsed into events. Event data can then be written to a file. The Data Interface can also be signaled to stop generating reports based on certain conditions being met such as a specified number of triggers, an amount of time passing, or upon the change of a Boolean value. The contents of event data can be found in any PhotoniQ user manual.

The Close header and source file combines two separate driver functions of the LabVIEW drivers into one, dealing with both closing the connection to the PhotoniQ and also handling errors. A connection can be closed without throwing an error and should be done at the end of any program that interfaces with a PhotoniQ.