**Extra documentations**

**FLC3030 camera**

The PCIe card of the Stresing FLC3030 camera has two ports, called S1 and S2. The camera receives signals from the Faros laser through port S1. It begins integrating for each trigger signal received through port S1 to match the probe. Port S2 receives signals from the Pockels cell and stores the trigger states to label each measurement as either pump-on or pump-off.

The trigger states are stored in the data structure exported by the camera. The camera is controlled through the *camera* function in *camera.py*. You specify a number of shots , **N**. The camera will then perform Nx2 measurements, with each measurement triggered by a signal through port S1. Once all **Nx2** measurement are taken, the data is exported in a (**Nx2**, 1088) array, where each row corresponds to a specific measurement. Columns 12 up to and including 1034 contain active pixels that measure light intensity. Pixels 0 up to and including 11, and 1036 up to and including 1088 contain metadata.

The trigger states of the ports S1 and S2 are stored in the third column. Without any triggers the column reflects a value of 0. A trigger from port S1 raises the value by 49152 and a trigger from port S2 raises the value by 16384. The meaning of the combined signal states S1/S2 is as follows:

0: OFF/OFF

32768: ON/OFF

16384: OFF/ON

49152: ON/ON

Each time we call the *camera* function to capture light intensity measurements, the camera drivers must start-up again. This causes the first few measurements to contain some sort of calibration signals instead of measured light intensities. Therefore, we made sure the camera captures the instructed number of shots twice, separated into 2 blocks. This functionality is built into the camera itself via a parameter called *number of blocks (nob).* The first block contains the calibrations signals (as I refer to it), and the second block contains the actual light intensity measurements. We only export the second block.

This behavior can potentially be improved by starting the camera drivers only once at startup (if possible) and setting *settings.nob* in *camera.py* to 1.

I haven’t been able to figure out what the exposure time of the camera is per reading.

**Real time probe and dA windows.**

When the software is started and all trigger ports are properly connected, you should immediately see a live view of the probe and dA values. At start-up, the integrations time is set to 1000 shots, but can be changed via an input box above the probe window.