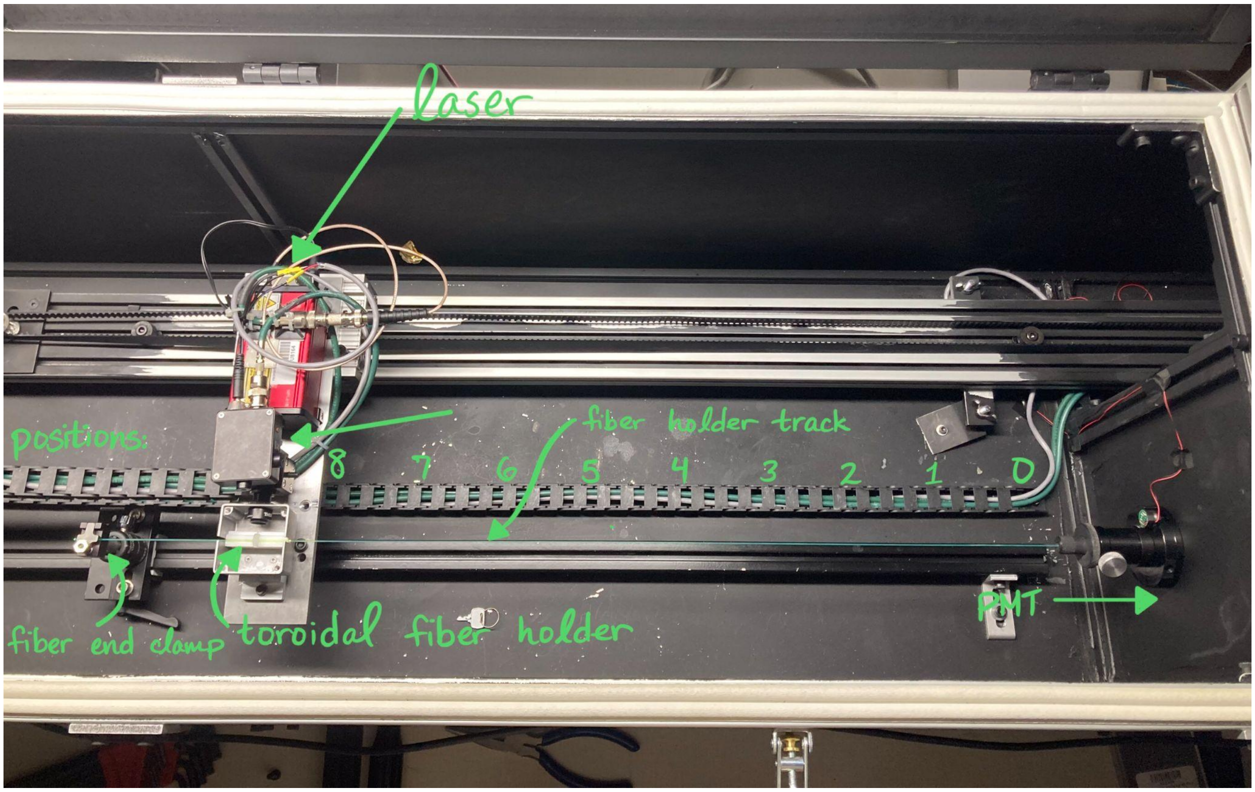

How to test WLS fibers in University of Rochester detector lab:

- follow Tejin's documentation to set up apparatus, PMT, and picoammeters
- log into computer - user is *fibertest* and password is *stepper*
- turn on laser and PMT
    - laser has a warmup time of 30-60 seconds
    - PMT has a warmup time of 30-60 minutes
- open terminal and navigate into DAQ folder
- move the laser to the position farthest from the PMT by running *shakeitup.py*
- remove your desired fiber from its packaging
- *when handling the fiber be careful not to touch the connector end of the fiber expose it to fluorescent light or sunlight, or otherwise damage it*
- turn the PMT iris off before opening the black box by flipping the switch on top of the PMT to "off"

- thread non-connector end through cylindrical PMT fiber holder, with the metal side of the holder facing the connector end
- carefully manipulate fiber so that the connector is firmly inside the PMT fiber holder, not resting on top
- insert fiber holder and connector end of fiber into PMT
- slide the free end of the fiber through the center of the toroidal fiber holder (remove the lid with the arrow if needed)
- secure the fiber by clamping the end with the magnetic end clamp
- replace the lid on the box, with the arrow facing the laser
- close the box lid, replace all three clamps on the outside of the box, close PMT switch in the center of the box, and open PMT iris by flipping the switch back to "on"
- run fiber test with *RunDAQ_v3.py*
    - I used 5 rail passes by adding '*--rep 5*' and each rail pass moved the laser towards the PMT, then away. I specified this by adding '*--pos 8,7,6,5,4,3,2,1,0,0,1,2,3,4,5,6,7,8'*. I usually ID'ed the fiber by the time I was testing it, by adding '*--id 1200'*, and specified the type of fiber by *'--suffix type_of_fiber'*.
- the program will deposit the CSV file produced in *DAQ/data/[today's date]/fibertest_….csv*
- move the fiber data CSV into the notebook *analysis_notebook.ipynb* in the "WLS-fiber-analysis" folder and analyze
    - choose the fit form for each fiber—noteworthy functions are "fit_data_double_exp" which fits the data with a curve of the form $D(x) = Ae^{-\alpha x}+Be^{-\beta x}$, and "fit_data_step2" which fits the data with a curve of the form $C\big(D(x) - R\cdot D(2L - x)\big)$ but does require that you specify values of A, α, B, and β
    - currently the value of L is hard-coded as 31.5"
    - run your function of choice with the fiber data, background data, and a title for the plot: for example, *fit_data_double_exp('data.csv', 'background.csv', 'Plot Title')*

    - the program will print the fit parameters, error, ^{2} of the fit, and a plot of the data and fit line
- remove the fiber by releasing the end clamp, sliding the fiber out of the toroidal holder, removing the PMT holder from the PMT, and sliding the PMT holder off the fiber
- repackage the fiber, with the connector end facing away from the opening of the bag

special fiber tests:

- background: Run the *RunDAQ_v3.py* program with the laser and PMT on, but no fiber inserted.
- destroyed: Use a razor blade to cut the non-connector end of the fiber at as shallow an angle as possible. *Before painting*, insert the fiber into the PMT and slide the end through the toroidal holder. Carefully lifting the end of the fiber, dab black paint onto the end of the fiber with a Q-tip or small brush. Secure the end of the fiber with the end clamp, replace the lid with the arrow, and run as usual. To remove the fiber, cut off any piece of the fiber which has paint on it with scissors before sliding the fiber out of the toroidal holder. Use a Kimwipe and alcohol to clean the end clamp before testing another fiber.

Attached below: Tejin Cai's documentation about *RunDAQ.py*. *RunDAQ_v3.py* has the same specifications, with the additional *--pos* command to specify the path of the laser

Contact Miriam Moore (miriamlynnnm@gmail.com) with further questions or for clarifications.

Operation of fiber testing box

Setup

The fiber testing box is a rectangular black box 2.55 x 0.40 x 0.34 m in dimension. An Arduino board controls the motion of a aluminum platform with a laser (NPL41B), splitter, test box system. The laser operates at 405 nm, the beam is split by a beam splitter cube allowing half of the beam to be reflected vertically and monitored by a photodetector. The horizontal beam then passes into the test box, illuminating a diffuser that then light up a wavelength-shifting (WLS) fiber.

Light from the WLS is sent to a PMT. The PMT is in a light-tight compartment with an iris in front of the PMT. The iris can be shut off with a switch, which prevents any light to reach the PMT when needed.

Note about the system

1) Shut off iris when opening the black box to prevent damage to PMT.
2) The laser has a violet indicator light at the back that shines steadily when the laser is on. I have covered it with a black tape but some light may still escape – so the black box is not totally dark. Consider completely erase the light.
3) When the platform moves, the detector signal may continually drop. The baseline signal established right now is about 4.6 μA. The signal could drop to as low as 3.8 μA. When the detector signal drops, the PMT signal measured at the same point also drops. I do not know exactly what's causing the issue. Some possibilities are
    1) Power to laser drops decreasing intensity.
    2) Platform motion causes beam to point away from the diffuser – less light illuminates the fiber.
4) Shaking the laser or the photodetector by hand don't affect the detector signal. So I tend to think the problem is in the laser. I have not tinkered with the laser power yet. Maybe it could help.
5) The problem can be "corrected" by instructing the platform to move towards the far side when it's already at the far side. This forces the bell to keep moving and "shake" the platform. I have observed doing so will cause the detector and PMT readings to go back up.

Software Instruction

Log on to **FiberTest**, open a terminal and navigate to DAQ folder. Inside there are a couple python packages:

1. fibertest.py
2. ft_arduino.py
3. DAQ.py
4. am_reader.py
5. RunDAQ.py
6. shakeitup.py

fibertest and **ft_arduino** are the original softwares used to control the platform.

am_reader is a simple utility to read the picoammeter signal output.

DAQ took functionalities from **ft_arduino** to improve the control of the platform, **am_reader** to read signal and has codes to export readings to csv files.

RunDAQ is a standalone python program to perform fiber measurement and automatically saves output to predefined directories.

Do

>> ./RunDAQ.py -h

will return

```
Process Fiber Test
optional arguments:
 -h, --help              show this help message and exit
 --id ID                 Set fiber ID, default to -1
 --suffix SUFFIX         Suffix to file name prefix_id_suffix.csv, default toNone
 --prefix PREFIX         Prefix to file name prefix_id_suffix.csv, default to fibertest
 --save_dir SAVE_DIR     Saving directory, default to ./data/{today's date}/
 --rep REP               Number of rail passes default to 5
 --n N                   Number of accumulated measurements default to 5
```

For example

./RunDAQ.py --id 20

will measure a fiber with id 20, run 5 rail passes, and at each point read ammeter 5 times and report the average. The output will be saved to ./data/20210612/fibertest_000020.csv

The tester can control the exact save directory, the prefix and a suffix in case a need arises.

The output file records data in columns of:

pmt01, pmt02… pmtNN, det01, det02,…,detNN, pmt_avg, pmt_stdev, det_avg, det_stdev

where 01, 02.. NN indicates the number of passes and each row contains the measurement at a point along the fiber. In order of furthest towards the PMT. So the first row record the measurements furthest from the PMT and the last row is the point closest to the PMT.