Lab notebook for Merschjann spectrometer experiment

Gabriel Ehrlich

2015.07.01 – 2015.09.30

Contents

2015.07.01

I figured I should record what I've done somewhere. Here's the structure: this will be a single document, backed up online (starting soon; GitHub repo or something, after I find out from Christoph whether that needs to be private). It will be kept in a folder with images and other relevant stuff that you'd want in a lab notebook, which will be included inline in this document using standard tex stuff (or else labbook syntax, if it's special). Having it all in one document makes it searchable in case down the line I don't remember on which day I learned something important.

Questions for Christoph:

1. Do I need to install the Solis software once for each camera? This time through I selected "iDus", and I'm not sure if I need to do it again with "Newton".

2. I'd like to use GitHub to back up my work. Do I need to use a private repository, or is public okay?

Today I set up my TeX environment in my VM (latexmk with vim and zathura), installed Python (Anaconda) on the Windows part of my laptop, and installed the SOLIS/SDK packages that Christoph passed me. Next: look at Daniel's code and the SDK driver API and figure out how to use it.

2015.07.05, to make up for a couple of hours Thurs morning

Christoph's answers:

1. Nope.
2. Will decide when we meet.

Next I'll try to figure out Daniel's code. Here's a summary of the file structure, as far as I can tell:

LabControl.py: Probably the main file. Imports `delaystage`, `ThorlabsBeamshutter`, `PicardShutter`, `ThemisApplication`, `SaveSettings`, `qledindicator`, `Joblist`, `MeasurementItem.DataViewer`, `ValveControl.ValveControl`, `RotationStage`, and `keithley`.

LabControl(DataViewer): Probably inherits from a widget or set of widgets? Keeps track of a bunch of widgets in a window, and the log file.

main!!!: tries to load settings, then starts app.

Keithley.py: Contains classes for controlling devices. Imports `qledindicator`.

Keithley(QtGui.QWidget): Widget subclass that can interact with devices.

acquireThread(QtCore.Thread): Initialized with something named `keithley`, presumably a Keithley object (?). Looks like it continuously gets data from `keithley`until the LED is unchecked.

KeithleyWidget(Keithley): Even more specificity to the widget? Has a bunch more methods that control devices.

main!!: Run the widget until it's closed (?), then close it.

delaystage.py: DelayStage(QObject): Methods to move delay stage.

DelayStageWidget(QWidget): Class with Qt-decorated methods that wrap DelayStage methods.

LabControl(QWindow): probably a window within which to put the widget. Has an attribute `DelayStageDock`that is a `QDockWidget`instance.

main: create a `QApplication`instance (since this is only called when this file is main), create `LabControl`instance, and show it.

2015.07.05, to make up for a couple of hours Thurs morning

ThorlabsBeamshutter.py: Beamshutter(QObject): Methods to operate beam shutter.

BeamshutterWidget(QWidget) : Qt-decorated wrappers for Beamshutter methods.

main: Make app and Beamshutter widget and start it.

Also a bunch of other stuff that may or may not turn out to be relevant. What Daniel hasn't written up yet are the optical chopper and the I/O card, but it seems pretty straightforward based on what he's done. The tricky part will be figuring out the drivers.

2015.07.06

Clearly today is going to be spent installing stuff... I got Anaconda installed, and now I'm installing Qt designer at the suggestion of the Internet.

Questions for Christoph:

1. Is it okay to use Qt designer under an LGPL license? (Related to GitHub question) (I can switch later if necessary)

2. Should I use the same .exe files to install SOLIS etc. on the lab computer, in addition to my laptop?

I'll ask him these when he returns, rather than over email.

What I learned: Qt Designer creates only the XML file that holds the UI; the rest is done using Python. This makes the stuff that is easy easy, so it seems like a really good idea. But Qt Designer is integrated as part of Qt Creator, which in order to work seems to require stuff I wouldn't have unless I downloaded the whole package. So I installed the whole thing and I'll probably be using only the Qt Designer portion.

Bug: Qt Designer icon grayed out. Solution: Need to create a project first, then select it and open it with Qt Designer.

Got through designing the UI. Next is a bit tricker: finish going through the tutorial named "Your first GUI app with Python and PyQt" so that I can get the Python part set up. Honestly this doesn't look too bad, which given the usual rate of errors and bugs means it'll probably only take me all summer!

2015.07.07

Spent getting stuff for my living space.

2015.07.08

First I set up Internet for my laptop—apparently there was no Ethernet driver installed. It works now.

Then I figured out how to make the mounted folders on my Linux box writeable. The trick was setting the owner using uid=1000 (or whatever user id is correct). I also mounted my entire Dropbox so that I can get to it easily. Also installed Dropbox on the lab computer.