Project Notes

**05/06/21 – 7 Hours**

* Refreshed on LabView basics
* Learnt basic ThorLabs motor control using LabView
* Encountered many problems trying to communicate with the motors and them having homing problems

**06/07/21 – 6 Hours**

* Successfully homed the X translation motor without finding out the proper procedure and solutions to our problems.
* Once homed, the program was created to move the angle motor for every x translation
* More bugs surrounding the communication between the controllers and PC were discovered. The computer goes into a “not responding” state when homing as the homing process can take a while. The PC cannot also have the APT User/Server applications running whilst the LabView program is and vice versa as the controllers can only talk to one application at a time.
* Application was setup to take input from the user for the steps and range of the translations.

**07/07/21 – 7 Hours**

* Controllers experienced homing issues again. It was found that the direction and homing limit switch were incorrect which was causing the issues. It was also concluded that a fix was undone when the controllers were turned off and on again. The fix was then hard coded into the program so the scanner will automatically home to the correct position on reboot.
* The last and external stepper motor controlling the Y translation was then introduced to the system. Liam gave me a run down on how the motor itself works and how it can be communicated using the serial comms. Liam also gave me a direction on how to implement the motors control into the main program. This implementation was found to be complex as many issues were encountered (i.e incorrect Baud rate and command structure). However, after approximately 4 hours I got the motor to drive from LabView. Proper motor control code was not written or integrated with the main program.

**11/08/2021 – 5 Hours**

* Due to Covid preventing access to the Lab at Uni, I had to work from home which limited what I could work on.
* This lead to me starting to familiarise myself with Python and looking at how I might make a 3D volumetric plot from the scanned data.

**17/08/2021 – 3 Hours**

* Looked into the best way to produce interactive 3D plots using Python.
* **Python Libraries for modelling volumetric data Matploblib, Ipyvolume,  [yt](http://yt-project.org/),**[**VTK**](https://www.vtk.org/)**and/or [Mayavi](http://docs.enthought.com/mayavi/mayavi/).**
* The two main ways which I have narrowed it down to is using Matplotlib and it’s built in displayer or using Ipyvolume through Jupyter Notebook and displaying it on a webpage.I am first going to experiment using Matplotlib/Pycharm as it seems like an easier route and as I like using an IDE rather than Jupyter Notebook and it may integrate nicer with being executed from LabView.
* However, it is noted that Ipyvolume is created uses WebGL in some compacity which suggests and seems that it runs smoother and faster. This means if we were to encounter performance issues it may be a viable option to transfer over to.
* Ipyvolume was found to run smoother and share a similar API as Matplotlib and therefore has been chosen to be used in testing.
* Next Steps would be to:
* 1. Convert test image to NP array
  + <https://www.pluralsight.com/guides/importing-image-data-into-numpy-arrays>
* 2. Display NP array on 3D Volume using Ipyvolume
* 3. Extra points for removing the black background
* 4. Maybe try creating some fake scan images and modelling them