3. Implementation

3.1 Software

3.1.1 GUI

The GUI itself was built using Qt Creator v5.14 which allowed easy accessibility and modification of what is known as widgets, or objects that a user can interact and use on the screen. The usage of Qt falls under the GNU Public License which would allow us to make our project open source. Qt makes extensive use of inheritance, where all classes are derived from the QObject class which allows for easy manipulation and creation of widgets. It also provides the creator with a signal/slot system as a quick and easy method for implementing event handling and object communication, based on the observer pattern. Qt also provides other technicalities such as threadding support, openGL libraries and many more. All of the code was generated based on looking up documentation found on: https://wiki.qt.io/Main.

3.1.2 MPU6050

Located at “RTEP/Software/MPU6050” is example code that reads the data from the MPU6050’s accelerometer registers. This code has been based off of the “LSM9DS1_RaspberryPi_CPP_library” git repository created by Bernd Porr. It implements the WiringPi library to communicate over I2C with the MPU6050 board and a timer to control the sampling period. .

3.1.3 Unit Tests

Unit tests simulating the data received from the hardware were created since COVID-19 halted further development and assembly for the MPU6050 boards. The required files to run the unit tests are ExerciseTests.cpp and ExerciseTests.h.

This unit test output x,y and z axis data that would be expected from the accelerometers during exercise 1 and 2 described above in section 2.