**ENGINEERING JOURNAL TEMPLATE**

# Date

* 5/1/2020

# Tasks

* Creating a function on the front end to take the output from the LIDAR driver and translate it to a point on the GUI.

# Reflection

* First close inspection of the code in the driver was required to find what types were used in the data output of the LIDAR unit, after a good while inspecting the code in the basic example in the SDK for the LIDAR unit, I found that the application uses an array of type “rplidar\_response\_measurement\_node\_t” called “nodes” that is 8192 in size. After inspecting a few definitions I found out “rplidar\_response\_measurement\_node\_t” is a structure returned form a LIDAR measurement, nodes is 8192 in size because at its highest speed the LIDAR unit can take ~8000 measurements in one rotation, therefore the array nodes contains a full frame within. After finding the “rplidar\_response\_measurement\_node\_t” structure deceleration I found 3 variables within, A char “Sync\_quality” that stores the quality of the measurement, a short “angle\_q6” that stores the angle the unit encoder was at the time of the measurement, and another short “distance\_q6” that stores the distance in mm that was measured.
* Secondly, I created a function and its deceleration in my “scanMapArea.cpp” and “scanMapArea.h” class and header files called “paintPoint” that takes 2 arguments, int distanceMm and int theta. I then had to do research to figure out how I can use a distance and angle to find a X and Y point on my GUI.

# Issues:

*Software:*

* Further research is required to find out how exactly I can get the function to paint a point on my GUI using the angle and distance measurements, I know some form of trigonometry is required, but don't yet know how to implement it in software.