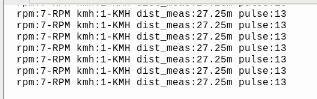
**ENGINEERING JOURNAL TEMPLATE**

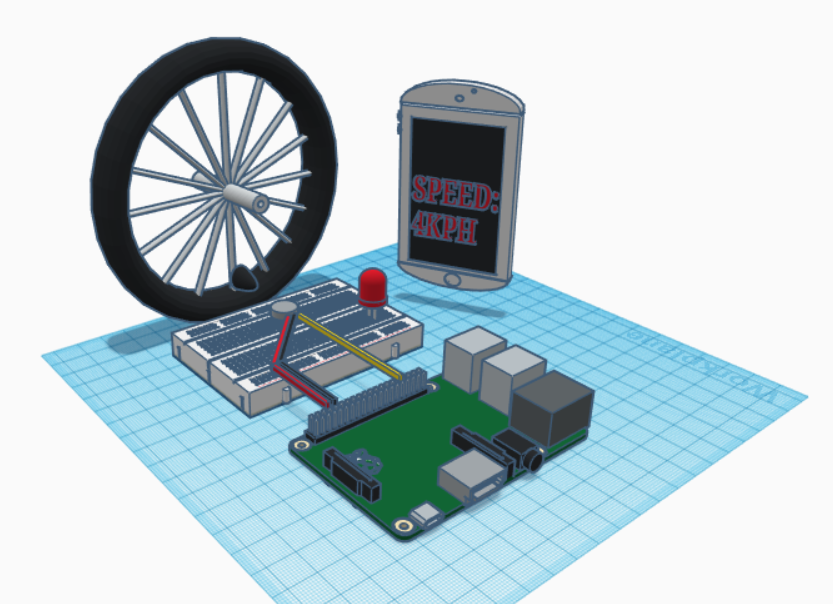
# Date

25th Nov- 29Nov

# Tasks

* After much research I am going to use cloudmqtt as the mqtt broker. Very simplistic and designed for sending small bits of data topics which is what my aim is.
* I want to print the data on a graph but I may have to save the data first and mqtt does not save its data. Possibly using SQLite.
* Python code for calculating the speed is working the way I want it now



* 
* CAD design of how I envisage to demonstrate my project. A magnet connected to the whell of a bike. As the wheel spins, the magnet will turn the hall effect sensor off. The time it takes for the magnet to do this every rotation is what we need to know in order to calculate speed

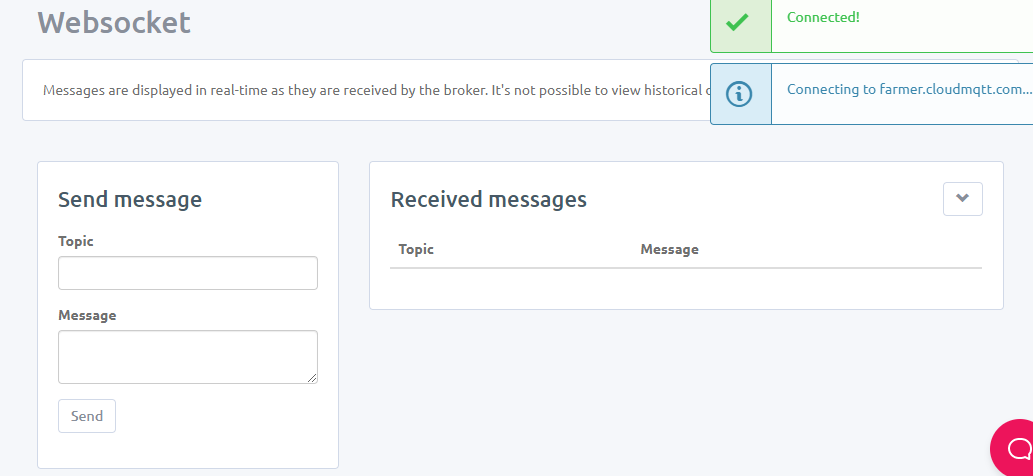
# Reflection

# Happy to get user interface on app the way I want it, speed activity can read hardcode data from cloudmqtt.com.

# Issues:

*Software:*

* + Connecting data from sensor to the app. I can hardcode data from cloudmqtt websocket and that will print on the app



# Solutions

*Hardware:*

* Working on software this week

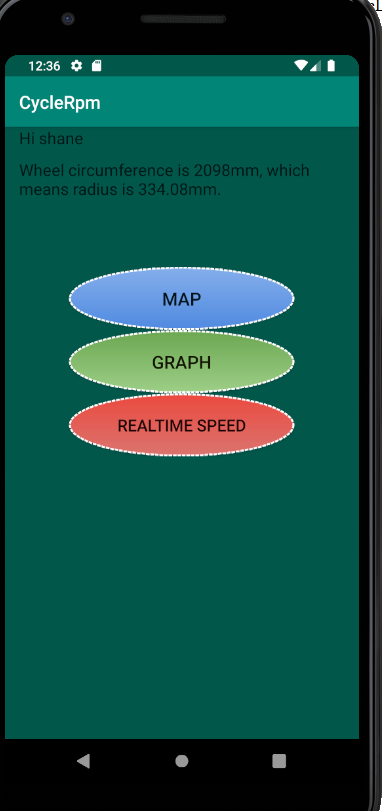
*Software:*

* + Made front end changes to the app as follows and added in the speed.
  + Created an android map api key that is letting me display my location on the map. API key is stored in the meta data in the androidmanifest.xml folder. Using an example online to help me with this. Only seems to work on the phone and not the emulator.

This is the first screen, I have made it so the details have to be entered to move to next activity.



This is next screen that prints name wheel circumference, and converts it into radius



You can then click on any button you want

I created a map function for riders to see their surroundings in case they are unsure of the route they want to take on their bike ride



This is the speed activity. At the moment I can hardcode data in from the mqtt websockets menu, picture above. I am happy that it is connecting to the broker, I will try to get the python code connecting to the broker on the other end.

