Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Interim Report 2017



Project Title: **Distributed Road Traffic Speed Monitoring**

Student: **Jeremy Chan**

CID: **00818433**

Course:  **4T**

Project Supervisor: **Dr. Ed Stott**

Second Marker:

Table of Contents

[I. Introduction 3](#_Toc472871757)

[II. Project Specification 3](#_Toc472871758)

[A. Project goals 3](#_Toc472871759)

[B. Advanced project goals 3](#_Toc472871760)

[C. Clarifications 3](#_Toc472871761)

[III. Related Work 4](#_Toc472871762)

[A. License plate recognition 4](#_Toc472871763)

[B. Peer to peer network 4](#_Toc472871764)

[C. Photo evidence publication 4](#_Toc472871765)

[IV. Estimation 4](#_Toc472871766)

[V. Background Reading and Literature Review 5](#_Toc472871767)

[A. License plate recognition 5](#_Toc472871768)

[B. Peer to peer network 6](#_Toc472871769)

[C. Photo evidence publication 6](#_Toc472871770)

[VI. Design 6](#_Toc472871771)

[A. License plate recognition 6](#_Toc472871772)

[B. Peer to peer network 6](#_Toc472871773)

[C. Photo evidence publication 6](#_Toc472871774)

[VII. Build 6](#_Toc472871775)

[A. License plate recognition 6](#_Toc472871776)

[B. Peer to peer network 6](#_Toc472871777)

[C. Photo evidence publication 6](#_Toc472871778)

[VIII. Testing 6](#_Toc472871779)

[A. License plate recognition 6](#_Toc472871780)

[B. Peer to peer network 6](#_Toc472871781)

[C. Photo evidence publication 7](#_Toc472871782)

[IX. Deployment and Maintainance 7](#_Toc472871783)

[X. References 7](#_Toc472871784)

[XI. Appendix 7](#_Toc472871785)

[A. Table of Figures 7](#_Toc472871786)

[B. Gantt Chart 8](#_Toc472871787)

# Introduction

* Project motivation
  + Lots of traffic speeding (ref the freedom of info report)
  + Find some info on lethality of speeding
  + Estimate on non motorway speeding?
* Project aim
* Accenture delivery method?

# Project Specification

From the project description, the goals of the project can be separated into the following:

## Project goals

* Implement a number plate recognition system using existing computer vision algorithms on a low-cost, readily available hardware platform.
* Set up a peer-to-peer network to share vehicle passing times and detect violations without the need for a central server.
* Publish photo evidence of any violations

## Advanced project goals

* Use the changes in the number plate geometry as the vehicle passes to detect the instantaneous speed of a vehicle. This provides a stand-alone mode that will aid adoption in areas where there isn't already an established network.
* Implement automatic peer discovery so that each device can find its neighbours and calculate the minimum legal transit time between them using a public mapping database.
* Add an encryption layer so that a hacker or rogue peer cannot use the network to track the movements of law-abiding vehicles.
* Package the system so that it can be easily installed in a home by an inexperienced user.

## Clarifications

After discussions with Dr. Stott, the following clarifications were made:

1. The number plate recognition system should be targeted at an off the shelf package, so there should be minimal setup and calibration done. This also means anyone, with the right equipment, should be able to download and compile the system if they have existing hardware.
2. The low-cost, readily available hardware platform will be a Raspberry Pi (RPi), with a camera attached to it. Using an RPi combines the best of cost (~£40 at time of writing), power (quad core CPU [1]), flexibility (camera can be any USB webcam or RPi’s official cameras), and support (development work on the RPi is extensive and there are ample tutorials/information online).
3. The peer to peer network should ideally be fully decentralised, so the system should be able to find peers without the help of a central server.
4. Publishing photo evidence will most likely be done onto a social network.
5. The public mapping database will be one accessible to most people – Google Maps API.
   1. https://developers.google.com/maps/documentation/roads/speed-limits
6. A hacker or rogue peer should not be able to extract license plates from the system remotely.
7. There should be a whitelist of emergency vehicles if possible in the case of there being an emergency vehicle over the speed limit.
8. The final package should be a software package uploaded onto GitHub, with clear instructions on how to compile and run the program with examples. As this project involves testing on hardware as well, a list of recommended hardware should also be provided (after successful testing), so the project can be easily replicated in the future.

# Related Work

Research into already existing products was done on the three main goals of the project, to judge the market feasibility of the project.

## License plate recognition

## Peer to peer network

## Photo evidence publication

Add justification on what to prioritise in the project, (network), and make sure this is the novel bit.

# Estimation

As the project timeline spans the entire duration of the 4th year, a plan to map out actions and results with reasonable time estimates is needed. This plan will be used to track progress throughout the project. However, given the open-ended nature of this design and build project beyond the defined project goals, there is a lot of design, build, and testing to be done. Hence, the estimates for design, build, and testing will inevitably overlap by a fair margin, thereby inflating the number of days to complete the project. The following table (Table 1) shows an estimate for each stage of project delivery, in days. A buffer is also included for unforeseen project issues that may push the timeline back. The corresponding Gantt chart is shown in section X.B.

Table : Estimation of time needed for each task in the project

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Time needed (days) | Buffer (days) | Actual time used (days) |
| Initial research about topic before first meeting | 2 | 0.5 | 2.5 |
| Clarify project aims and goals | 1 | 0.5 | 1 |
| Prioritise project goals | 1 | 0.5 | 1 |
| Define project requirements | 1 | 0.5 | 1.5 |
| High level design of license plate recognition | 5 | 2 |  |
| High level design of P2P network | 5 | 2 |  |
| High level design of security issues, encryption | 5 | 2 |  |
| Coding of license plate recognition | 10 | 5 |  |
| Coding of P2P network | 10 | 5 |  |
| Coding of privacy issues, encryption | 10 | 5 |  |
| Testing of license plate recognition | 5 | 2 |  |
| Testing of P2P network | 5 | 2 |  |
| Testing of privacy issues, encryption | 5 | 2 |  |
| Integration of all systems, along with possible hardware | 10 | 2 |  |
| Packaging and release on GitHub | 2 | 0.5 |  |
| Documentation of code | 2 | 1 |  |
| Report writing | 10 | 5 |  |
| Making of demo | 5 | 2 |  |
| TOTAL | 99 | 39.5 | 6 |

To track the progress of the project, a project management software was used. There available for free use. Asana (<https://asana.com/>) was previously used in the 3rd year project to great effect. However, as this project is not collaborative, the project pane of GitHub was used as a simple tracker (Figure 1). Four groups of tasks were made; the GitHub project tracker allows for easy moving of tasks from one group to another. Another advantage of using the GitHub project tracker is that everything regarding the project is in one place, accessible and modifiable from anywhere.

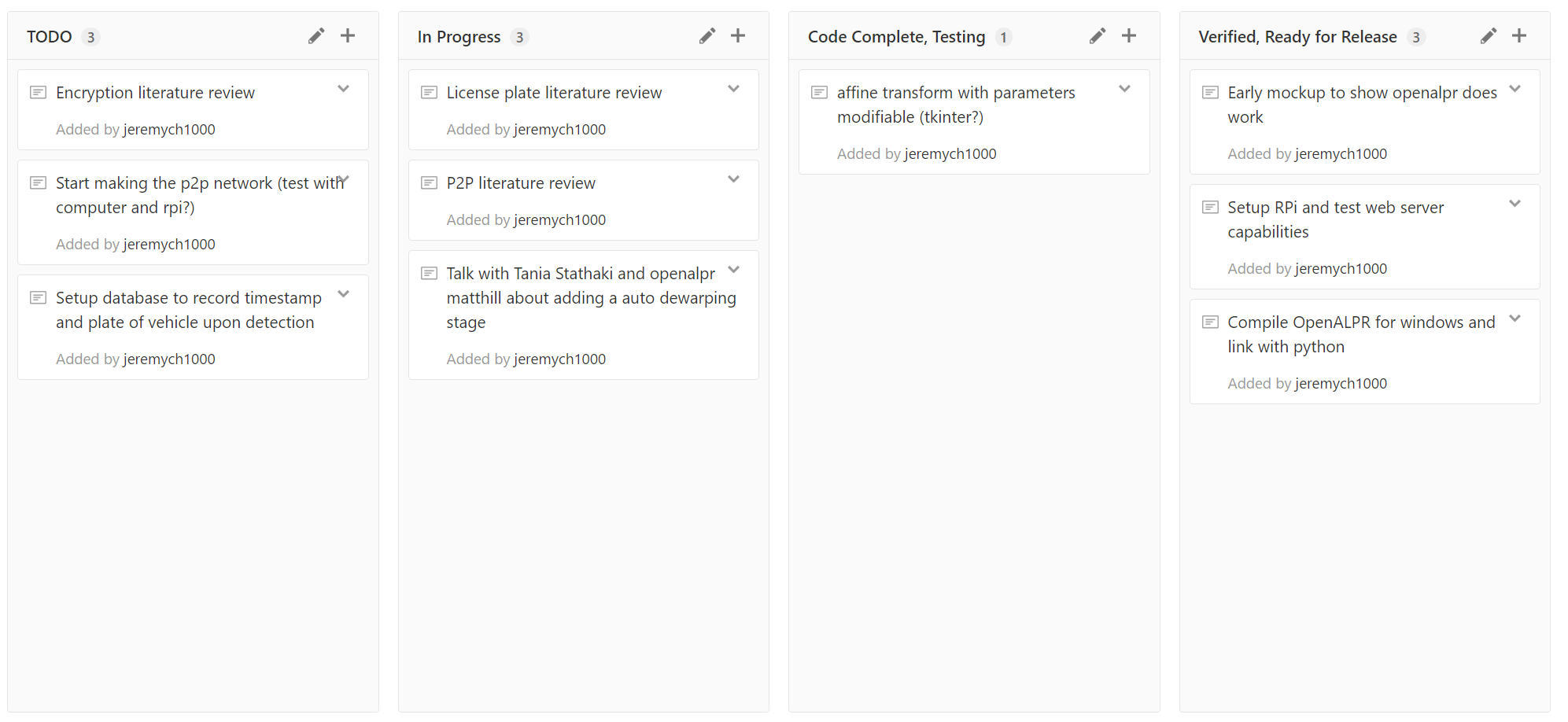


Figure : GitHub project tracker

# Background Reading and Literature Review

## License plate recognition

## Peer to peer network

## Photo evidence publication

# Design

## License plate recognition

## Peer to peer network

## Photo evidence publication

# Build

## License plate recognition

## Peer to peer network

## Photo evidence publication

# Testing

## License plate recognition

## Peer to peer network

## Photo evidence publication

# Deployment and Maintainance

# References

[1] ‘Raspberry Pi 3 Model B’, *Raspberry Pi*. .

# Appendix

## Table of Figures

[Figure 1: GitHub project tracker 4](#_Toc472805942)

## Gantt Chart

