Intersection Traffic Control System for Autonomous Vehicles

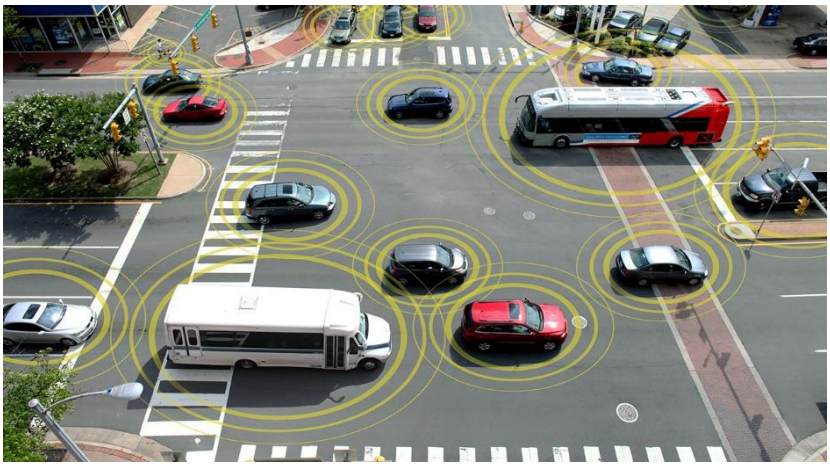


Figure 1: A future where vehicles and static road entities inter-communicate.

PROJECT GROUP 09:

MOHIKA GUPTA (CID: 01056243)

PIERS DE BASTO (CID: 01051176)

ASHISH SUNDAR (CID: 01058844)

HAARIS OSMAN MEHMOOD (CID: 01063083)

ARTHUR STEMMER (CID: 01070496)

MICHAL MYSIOR (CID: 01104972)

WALEED EL-GERESY (CID: 01054800)

SUPERVISOR: DR. BRUNO CLERCKX

13/03/17

Table of Contents

1 Abstract

Autonomous cars are starting to become a reality, which will reduce traffic accidents by eliminating human error. Presently, traffic at intersections is largely managed by traffic lights, which are time and fuel inefficient. For example, cars may be required to wait even when the intersection is empty or if all cars had the same destination lane.

Assuming a future where autonomous cars have become a reality, this project (ITCS) aims to improve the management of traffic for single lane intersections. This report outlines the development of the project.

The group developed a software simulation and built hardware (a transceiver and microcontroller) to implement mesh network and logic system (software), as well as the physical communication device (hardware). To implement the intersection management system, algorithms used data collected about each car to determine the optimal timing for the vehicles to cross the intersection safely. Using an algorithm, it was seen that several different cars could traverse the intersection simultaneously and thereby increase the throughput of the intersection in comparison to the current system of traffic lights. To allow the vehicles to share the data needed to implement these algorithms, a mesh network to allow vehicle-to-vehicle communication was developed. It was observed that the hardware devices were capable of setting up an ad hoc mesh network which could be used to exchange information between the vehicles.

2 Introduction

2.1 Background

With autonomous cars already starting to become a reality in the form of self-driving cars [1], the current traffic system will soon become redundant. This project aims to provide an intersection management system for autonomous cars.

Presently, there is an intersection control paradigm for Autonomous Intersection Management (AIM) being tested and implemented [2]. This project utilises an intersection manager to process requests and issue reservations. The risk of using a Vehicle- to –infrastructure (V2I) approach like in AIM is if communication with the intersection manager and the vehicle fails, there is no way to ensure the intersection is managed safely. ITCS tackles this problem by enabling cars to communicate with each other via Ad-Hoc mesh networks.

To develop a mesh network for intersection management, each car is represented as a node in the network. A mesh network enables the transmission of data over a large topology without the need for a central hub. This is achieved by 'hopping' packets from one node to the next, reducing data loss, noise and overall transmission power. Mesh networks are also resilient to node failures since multiple paths to connect two nodes can exist. These properties of the mesh network topology make it an ideal solution for managing cars that are close enough to communicate between each other yet far away from any infrastructure, such as rural areas.

In the interim report, design concepts for the different subsections of this project were considered. This report discusses the design choices made, further concept designs considered, how the final designs were implemented and suggests further work for this project.

2.2 Main Modules

Given that the final product will communicate with multiple entities and perform data processing, the most appropriate model for inspiration was the Open System Interconnection (OSI) 7-layer model (see appendix section 11.1). This model is a widely used, general networking framework that describes the different sub-systems required when transmitting and receiving data over any medium and forming any type of network. Due to time-constraints, the model was simplified to focus on the aspects that are most important for the project. As a result, the design and development of the product was spilt across three distinct layers.

2.2.1 Physical Layer

The physical layer consists of several hardware modules: communication modules, sensors and a power supply.

Communication involves interfacing with radio transmitter and receiver modules; sensors enable real-time data logging of acceleration and bearing. The power module involves all hardware required to provide a stable power supply for each node.

2.2.2 Network Layer

This is the backbone layer of the project; it incorporates a variety of algorithms and coded implementations that enable peer-to-peer communications as well as ensuring that the network is scalable and adaptable. It will also be responsible for reliably receiving data packets from other nodes in real-time and decoding them into useful information.

2.2.3 Application Layer

The application layer builds on the physical and network layer to implement the intersection traffic management system. Data collected in the Physical layer is propagated through the network via the Network layer. The Application layer then uses this data to implement the algorithms used for intersection management. These algorithms aim to maximise throughput whilst allowing vehicles to navigate the intersection safely.

3 Design Criteria

As detailed in the Interim Report, the following are seven criteria that will help team members decide on what ideas to implement, trade-offs to consider and how to select the most effective design choices. These criteria touch all subsystems of the project i.e. the physical, mesh and application layer.

1. Reliability
2. Performance + Response
3. Adaptability
4. Cost Effectiveness
5. Self-sufficiency
6. Reproducibility + Manufacturability (Design)
7. Unique Selling Point

For more information on these design criteria, please refer to appendix section 11.1

4 Concept Design & Selection

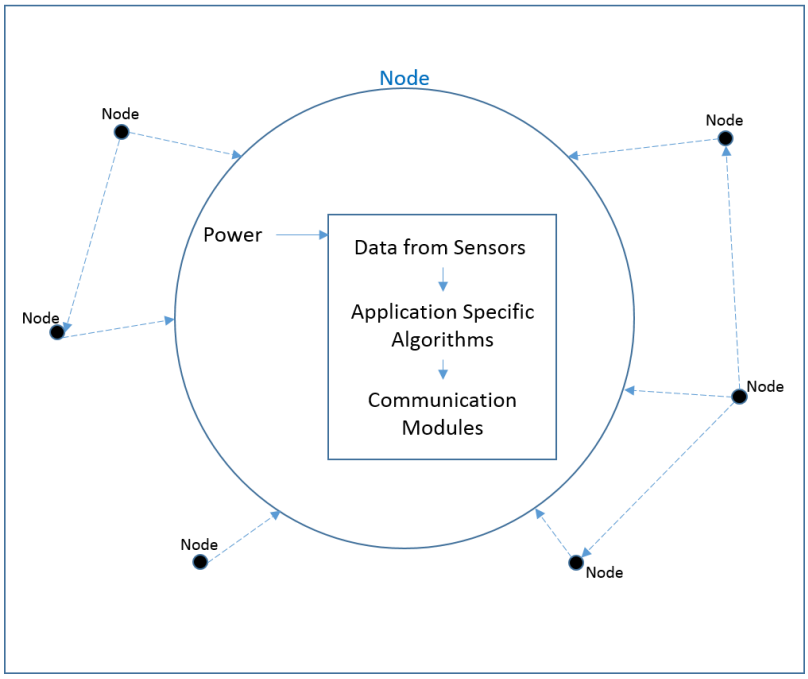


Figure 2: High level diagram for the product implementation.

As outlined in the Interim Report, the overall structure of the project is as shown in figure 2.
As described in section 3.3, this overall structure was split into 3 modules for which different design concepts were considered

4.1 Physical Layer

4.1.1 Communications

For the purpose of a building a wireless mesh network, a few communication media were considered.

It was important to use a module that provided an interface simple enough so that the group would not have to focus too much on low-level data transmission algorithms. On the other hand, the hardware had to provide enough flexibility to allow mesh network implementation.

A few existing wireless standards were considered, namely Bluetooth, Wi-Fi and ZigBee. Unfortunately, all of them turned out to have disadvantages that disqualified them from being used in the network:

- Bluetooth – it has sufficient data rates but rather low range. Also, it is designed for exchanging data between paired devices, meaning that it is not suitable for dynamic network used in traffic control
- Wi-Fi – similarly to Bluetooth, it requires pairing.
- ZigBee – it is a mesh network standard oriented towards low power devices for this reason it might not allow enough flexibility for the group to create optimal network implementation. Moreover, modules using ZigBee are quite expensive, and their range is limited [3].

For those reasons, it was decided that best suited would be a simple wireless module with all RF components and a controllable chip on board, which however does not implement any specific wireless standard.
There are a few such modules on the market, mostly using radio frequencies 433 MHz, 868 MHz and 2.4 GHz (which do not require license in Europe). Examples of such modules are nRF24L01+, RFM12B

or RFM22. As all of these modules fit the desired requirements for a communication module, the final decision was to use the RFM12B module as it seemed to be the best documented (though informally) out of all the options.

The chosen version of RFM12B [4] is a transceiver in a surface-mount package and uses 868 MHz frequencies. It communicates with the MCU via SPI interface and has all necessary Radio Frequency (RF) components, meaning further hardware is limited to a simple antenna.

Criteria	Bluetooth	WiFi	ZigBee module	RFM12B (and similar)
Hardware Setup Cost	-	-	--	0
Requires pairing?	-	-	0	0
Range	-	0	-	0
Hardware Setup Time	+	+	+	0
Flexibility in use	--	--	-	0
Scalability	--	-	0	0
Reliability	+	+	+	0
Final Score	-5	-3	-2	0
Rank	4	3	2	1
Continue?	No	No	No	Yes

4.1.2 Sensors

In order to measure and store physical quantities such as velocity, position, etc. it was imperative that every node had its own sensor module. Each node would measure and record data about itself to accurately determine the time taken to reach the entrance of an intersection. A similar method will also be used to calculate time to completely cross an intersection. For these purposes two main approaches were considered:

1. The first method involved using a GPS receiver and linking it with a microcontroller. The GPS would then accurately provide the position and speed of the node. This information would be mapped to a centralised coordinate system. Assuming the location of intersections are fixed and pre-determined from local maps, the calculation for 'time to intersection' can easily be performed.
2. The second method took a more laid-back approach, using an accelerometer and magnetometer to record 3D acceleration and 3D magnetic flux density. Data from the accelerometer would then be integrated once to find the absolute value for speed, and integrated twice to determine displacement. The magnetometer would be used to calculate the heading, which would determine the direction in which the car is travelling with respect to north. This method assumes all autonomous cars have access to built-in GPS modules and hence the distance data from an accelerometer could be converted to relative distance from the origin. Similar to the previous method, speed and distance would then be used to calculate time to reach or cross the intersection.

Criteria	GPS	eCompass
Setup Cost	+	0
Setup Time	+	0
Working?	+	0
Scalability	0	0
Reliability	+	0
Final Score	3	0
Rank	1	2
Continue?	No	yes

4.2 Network Layer

As stated in the Interim Report, the basis of all V2V communication is to form a mesh network representing each car as a node in the network. Two main functions will allow for communication of nodes across the network:

- Communicate – A function to allow a given node to communicate with its neighbouring nodes. It will either receive data from different nodes and broadcast it or send the data for analysis to the relevant function if the current node itself was the recipient.
- Path – A function, which calculates the shortest path to connect to nodes. This will allow any two nodes in the mesh network to communicate (relevant algorithms can be discussed in the design concept section)

4.3 Application Layer

Several different designs were considered for the application layer. These were as follows:

1. A straightforward switching system

This system operates upon two key principles. Priority is always given to the car that arrives at the intersection 1st, regardless of other parameters (such as car speed). A maximum of 1 car is allowed to traverse the intersection at any time, forming a simple collision avoidance algorithm. The disadvantage of this system is that the throughput (the number of cars that traverse the junction per unit time) could be improved by allowing more cars to traverse the intersection simultaneously (see Figure 3).

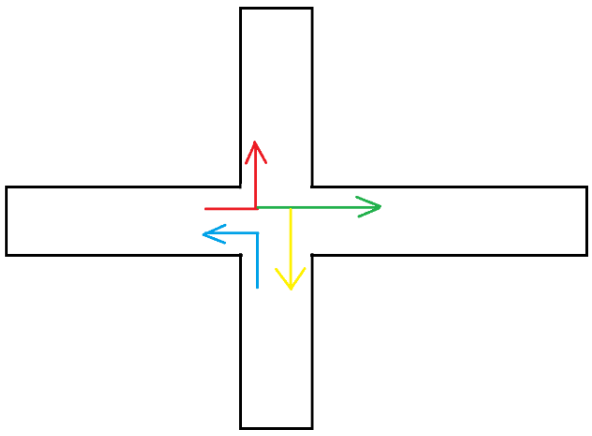


Figure 3: The blue path can be traversed while any of the other three coloured paths are traversed

2. Train grouping paradigm (BATCH)

This solution relies on grouping cars with the same route through the intersection. If several cars entering one lane are destined for the same final lane, they can essentially be considered to be one longer vehicle, termed a 'macro car'. If the simple algorithm described in the 1st point is then applied to the 'macro cars', throughput can be increased. However, the problem of increased waiting time for cars with different routes arises.

3. Numerical optimisation of a 3D space-time graph

This optimisation relies on plotting the intersection x and y position coordinates against time on a 3-D graph. Paths of cars in space and time will be plotted with a spatial radius corresponding to the dimensions of the car. Paths In space-time for each car will be plotted and numerical techniques will be used to optimise the number of paths that can be plotted on the graph without overlap (see Figure 4). This method is the optimal technique, however it requires significant computational power to process such large amounts of data. Furthermore, any changes in the system require full re-computation of the numerical optimisation, increasing the computation time and power required.

Figure 4 visualises the numerical optimization process. If the red and blue paths intersect then the paths of the cars are impossible. The figures on the left show the x, y view, with the projected paths of the red and blue cars. The right figures show the problem with these paths in time, with an intersection between the two paths evident. The gradient of each path with respect to time is representative of the speed.

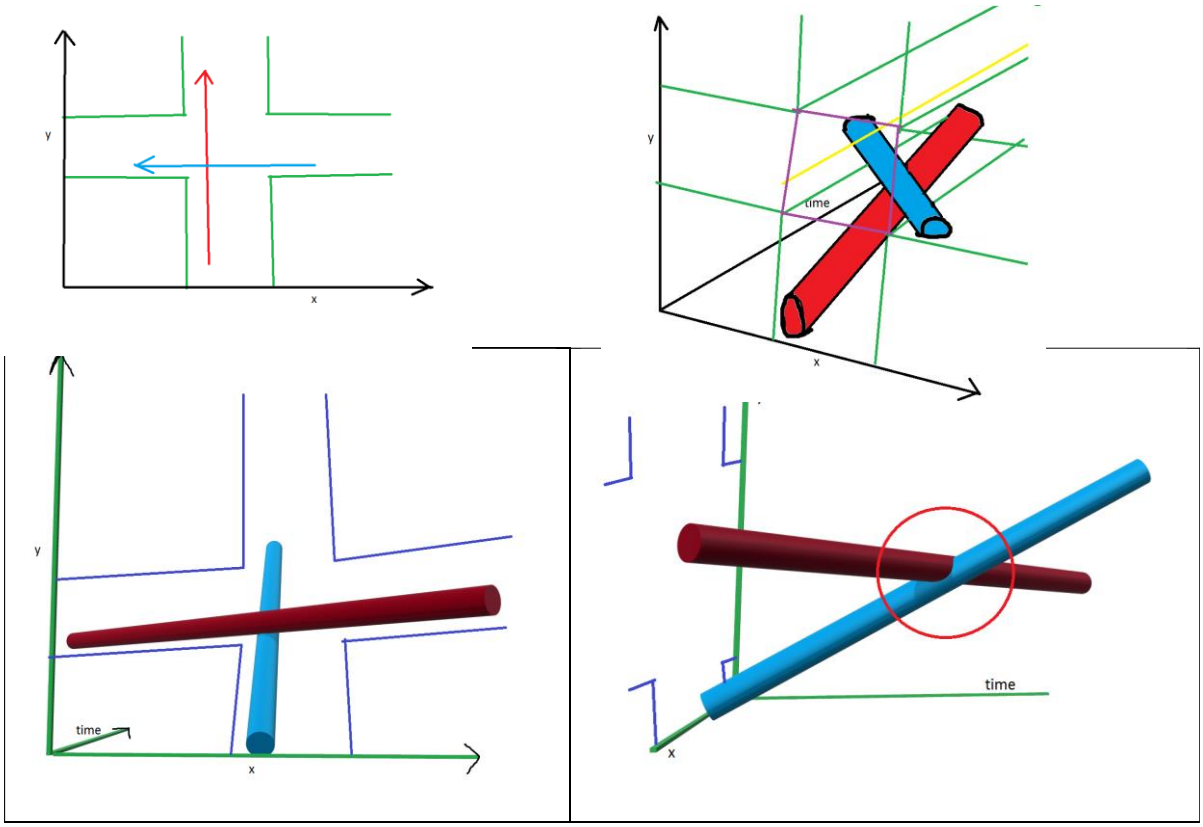


Figure 4: A visualisation of the numerical optimisation process.

4. Absolute priority function with master node

This method uses a combination of methods 1 and 2. Each car sends a calculated value called priority according to various different parameters (detailed in the concept development section). Priority will determine the order in which the cars are allowed to traverse the junction and will be deterministic such that the nodes will be able to determine whether or not they can traverse the junction themselves.

5. Scheduling

This method is an alternative to the priority method. The problem with the priority method is that priority is constantly changing up until the car reaches the intersection. Therefore, whether or not the car can pass through the intersection is only known for certain at the time of arrival at the intersection. This method instead assigns a schedule according to the priority function and means that cars should know in advance their predicted time of intersection arrival and traversal. This should enable them to adjust their speed in advance to reduce the time the cars are waiting at the intersection, increasing the efficiency. Cars will individually submit a timeframe in which they are able to arrive at the intersection based on the distance to the junction and the range of speeds the car is allowed to travel at. Priority factors, including whether the car can turn at the same time as another waiting car (see figure 3 for elucidation), are then used in order to decide upon the order the cars are allowed to traverse.

Factor	Switching	Train Grouping	Numerical Optimisation	Absolute Priority	Schedule
Local processing required	+	0	-	-	-
Data transmission packet size	0	0	-	+	-
Local storage needs	+	0	-	+	0
Simplicity	+	0	-	0	0
Fairness	-	-	+	+	+
Fuel efficiency	-	0	+	0	+
Car Throughput	-	0	+	+	+
Real-time Flexibility	+	0	-	+	+
Notice of changes	-	0	-	-	+
Total	0	-	-	3+	3+

As can be seen in the above decision table, although each of the methods had unique advantages, overall it was decided that the absolute priority function and the scheduling method were the methods with the best overall performance in multiple areas. A scheduling function which schedules cars based on various factors determined by the priority was implemented. The details of this are outlined in concept development section 6.3.

5 Concept Development

5.1 Physical Layer

5.1.1 Sensors Pre-Processing

This is a function that is called before transmitting a data packet over the network. It gathers raw acceleration data from the internal sensor of a given node, removes effects of gravity along each of the axis and stabilises it using a moving average algorithm. It then converts it into velocity and displacement and then finally produces speed and distance travelled respectively using numerical step integration [5].

5.1.1.1 Inputs

3D Acceleration (g), 3D Magnetic Flux Density (G) (Data from sensors)
For detail on how the raw data was processed to extract relevant information see appendix section 11.2.

5.1.1.2 Output

The final output of this function is 'time to intersection', which is given by the distance to intersection divided by the speed. Assuming that all autonomous cars will be able to use GPS, for the purposes of simulation it is assumed that the GPS coordinates can be mapped to the inputs used for the algorithms developed.

5.1.1.3 Testing

From tests conducted, a major issue in integrating the data from the accelerometer to obtain displacement was discovered: small errors accumulate progressively through the double integration of the initial data. Within a short span of time the data becomes quite different from its true value. This is a well-known problem known as integration drift due to sensor bias. The only viable method to solve this problem is to use GPS instead to determine position and velocity. Since the product is designed for autonomous cars of the future it is safe to assume that the cars will have some form of GPS receiver built-in and hence will be able to make far more accurate data measurements. Unfortunately, due to a strict budget and specific testing conditions, namely low displacement and indoor environment, the group was unable to use GPS modules and hence decided to use the accelerometer for demonstration purposes. For short demonstrations the error bias should be within acceptable levels.

5.1.2 Radio Modules

Initially it had been decided to use the RFM12B for the reasons specified previously (see section 5.1.1). However, trials to get a stable link between modules resulted in a failure. None of the different libraries found enabled the modules to communicate properly; it was hard to achieve a stable connection between the module and the mbed, let alone make radio communication work.

In light of this, it was decided to use a different, slightly cheaper nRF24L01+ module which has less informal documentation, but was well documented on the mbed developer website [6], which was found to be more accurate than the informal documentation found for the RFM12B. These modules turned out to be easy to launch and flexible and allowed for different data rates and variable frequencies allowing multiple channels amongst other features. Therefore, they are the final option chosen.

Criteria	nRF24L01	RFM12B
Setup Cost	+	0
Setup Time	+	0

Working?	+	0
Scalability	0	0
Reliability	+	0
Final Score	3	0
Rank	1	2
Continue?	yes	no

1.1.3 Power

For reasons specified in the interim report, it was decided to use a Buck SMPS to power each node (i.e. the microcontroller, communications modules and sensors). The concept design initially used was only partially implemented, since some of the design constraints were changed.

5.1.2.1 Chip selection

The LM2574N was selected as it can handle low power and voltage while implementing feedback to sustain the output voltage (at any load, when tested from 20 mA to 200 mA only 0.3 volts of change was observed). It runs at 5 kHz, which is ideal. At higher frequencies, noise and fluctuations are observed and at lower frequencies wide fluctuations of the power supply are observed. It also has an option of using other fixed voltages, so the same chip can be used even if the power requirements change. This feature proved to be useful as the communications chip was changed to another that needed a 5 volt supply line, as well as the mbed requiring a board to power it which also needed a 5 volt supply line.

The Mbed has an additional internal voltage regulator as well, which will be used to provide the sensor and the communications chip with a 3.3 V supply [7].

5.1.2.2 Calculations

Please refer to appendix section 11.3 for details of how component values were selected.

5.1.2.3 Circuit Used

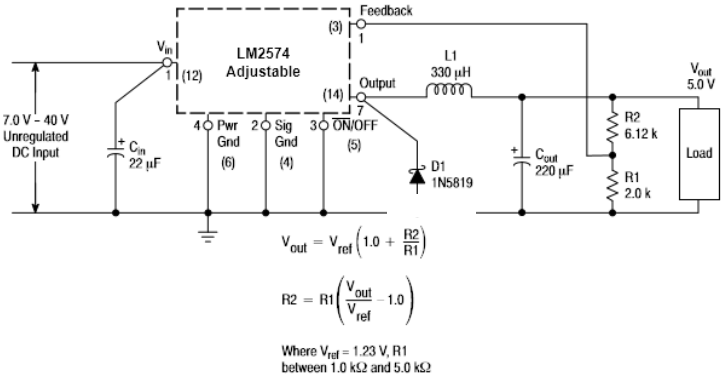


Figure 5: Final Circuit Design [8]

5.1.2.4 PCB Consideration

This chip is a high frequency (52 kHz) oscillator which means it can input a lot of noise and transients through stray capacitances, so the tracks must be isolated enough to avoid this. Single point grounding is also a viable means to avoid this, as the electrical noise would not have any viable path to recirculate in the circuit. In this essence, the power rail must also be kept as short as possible to avoid any stray capacitances.

5.1.2.5 Critical Analysis

To test to see what Vref is, as a small error was observed (an output voltage of 5.04 was obtained instead of an output voltage of 5.01 (using a test value of 5.0 volts and resistor values of 4k7 ohms for R2 and 15k ohms for R1)), two different combinations of resistors were input to observe the output voltage. The ones mentioned above yielded a Vref value of 1.2186, and using a R1 value of 5k and a R2 value of 11k ohms yielded a Vref value of 1.2192 V. From this testing, a Vref value of 1.22 can be assumed.

With testing for the output voltage waveform, 150 mV to 200 mV of peak-to-peak noise was observed, with significantly more overshoot on the positive side than the negative (150 mV plus and 50 mV minus). Therefore, the output value attempted for must be reduced to 4.9 volts, to avoid any potential overshoot, as well as to avoid amplifying the positive overshoot when the communications module switches from transmitting to receiving (as receiving draws less current).

Using this value and the fact that R1 can be a maximum of 5K ohms (maximise the resistance to minimise the power loss), an R2 value of 13K3 ohms was obtained to yield an output voltage of 5.0 volts. However, as 13K3 isn't a standard resistor value, 13K was used as a substitute, to yield 4.91 V as an output voltage, which is just under the ideal value of 5.0 volts.

While testing for potential input voltages, a 9 volt input voltage had more noise than a 4.5 volt input voltage. Putting a 1Ω resistor in series between the input and the chip input pin showed that an extra 5 mA was being used up for the 9V supply, which equates to an extra 22.5 mW being used. Therefore, even though the 9V battery is smaller and requires less housing space, for the above reasons and the fact that they outweigh the positives, the 4.5 volt battery combination rather than the 9 volt battery was used.

5.1.2.6 Motor Considerations

Since motors will be used in the prototype, this also needs to be considered as it can easily transmit back EMF into a chip and render it unusable. To prevent this, there will need to be a fly back diode connected in forward bias between the negative terminal of each motor and the positive rail to reduce the effect of back EMF.

5.1.2.7 Estimated Load & Battery Life

The estimated load and thus battery life can be calculated from the following. ARM mbed processor LPC11U24 estimated to run at max frequency (50 MHz) - draws 7mA and runs at 3.3 volts (the board has its own power control so it can run with a ripple as long as the supply is above 4.5 volts) [9].

Communication module nRF24L01+ is estimated to run on the setting for transmitting at 0dBm output power, drawing ~11.3 mA, and is estimated to receive at a rate of 2 Mbps, drawing ~13.5 mA, overall 24.8 mA [10].

The sensor LSM303DLHC - draws 110uA and runs at 3.3 volts. [11]

5.1.2.8 Power Drawn

The current drawn by the above components sum to 19.51 mA, and with a supply voltage of 3.3 volts, that sums to 64.38mW (P=IV). If one battery provides 2600mWh [12] then one battery alone would provide about 40 hours of runtime. 4 batteries are needed as higher input voltages provide greater efficiency with this SMPS chip, so ideally 12V is required but portability is also a consideration, so compromising between the two results in using 4 batteries.

The SMPS chip itself draws 15uA, so overall power used is 64.47mW, and with ~70% efficiency at this voltage, 4 batteries are expected to last about 110 hours [13].

5.2 Network Layer

For the final implementation it was decided to use a network that sends the (processed) sensor data to all the surrounding nodes. Each packet does not have a destination address and therefore the network will keep propagating the packet until the max-hops value reaches zero at which point the packet will not be propagated any longer. This is to ensure that there are no packets that keep floating around on the network for no reason.

Each packet has the following structure:

BYTE [0-5]	BYTE [6]	BYTE [7]	BYTE [8-11]	BYTE[12-15]	BYTE [16]	BYTE [17]	BYTE[18]
MAC[0-5]	LaneID	IntendLaneID	SPEED[0-3]	DIST[0-3]	PRIORITY	MAXHOPS	PNUM

BYTE[0-5] MAC: Each mbed had its own unique MAC address which can be accessed through the mbed library. Instead of defining an addressing system this predefined feature was used. The MAC address has 6 bytes of data.
BYTE[6] LaneID: This is the lane on which the car is currently located. This is necessary for surrounding cars to make a decision.
BYTE[7] IntendedLaneID: Together with the LaneID this value is used to determine what the path of the car will be over the junction. Both variables can only have 8 values.
BYTE[8-11] Speed: This is the speed at which the car is approaching the junction. Internally it is a float and so the IEEE-754 is encoded into four bytes for transmission and decode the data on the receiving end.
BYTE[12-15] Distance: Together with the speed it is possible to use this data to determine the time until the car reaches the junction. However, it has been decided that to accurately control the car it would insufficient to only send the time until the car reaches the junction.
BYTE[16] Priority: An integer number to indicate the priority with which the cars can enter the junction
BYTE[17] Maximum Hops: To make sure that the packets do not keep propagating through the network endlessly, this parameter sets the maximum number of hops a packet can do.
BYTE[18] Packet Number: To ensure that the same packets are not received more than once a packet identifier is added. This together with the MAC address provides a unique ID for each packet. That is, assuming there are less than 256 packets from one node propagating through the network at once.

5.2.1 Implementing the mesh network:

Using a 'receive and scatter' method where when a node receives a signal it sends the signal to all nodes within range, the next node does the same and so on.
For the purposes of the prototype the assumption that most of the nodes function without many problems (colliding packets etc.) needs to be relied on. That could be solved by having a better, more expensive transceiver. Nevertheless it is possible to identify a couple of issues that would arise looking at the system purely from the perspective of the network layer. The most important problem with the mesh topology is that the amount of data sent through the network becomes very large very quickly. To reduce this problem and still keep the network functioning properly two solutions have been implemented. The first is to only allow packets to hop a certain number of nodes (MAXHOPS) and the second is to remove the packet from the network if it is received by the receiving node for the second time by simply not retransmitting it. The first solution is very simple, if

the MAXHOPS parameter becomes zero the packet will not be retransmitted because it has been in the network for too long. The second works as follows: after the node has received a signal, all subsequent messages with the same MAC address and packet ID will be considered invalid and will not retransmitted.

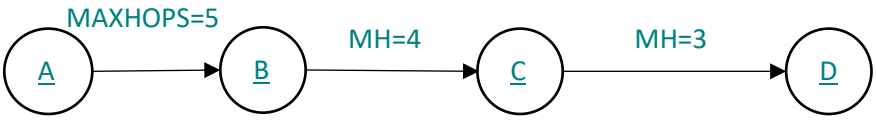


Figure 6: A visual representation of MaxHops

To do this each node will need to keep a buffer of all the most recent message packet IDs and sender MAC addresses. Thankfully this will likely not require very much memory because each packet only requires 7 bytes of storage. To make this system efficient each node will need to be able quickly look through this list of MAC addresses and packet IDs therefore it would be beneficial to use a hash table for storage (quick lookup).

5.3 Application Layer

The application layer is what implements the intersection traffic control. The application layer runs algorithms that use the data received by a node which is stored in the data packet (see Network Layer) to determine if it is safe for the node to navigate the intersection.

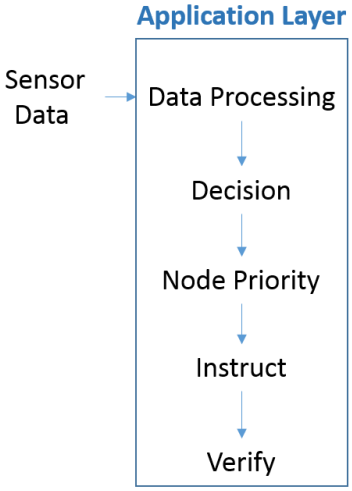


Figure 7: Outline of Application Layer

Figure 7 shows the structure of the application layer as outlined in the Interim Report, however, after further consideration (see section 5.3) the application layer algorithms were developed as detailed below.

5.3.1 Inputs

The data that the main algorithm would receive would be:

1. **CarID** – This is to uniquely identify each car, this can also be the MAC address of each mbed used for prototyping.
2. **LaneID** – This will determine which lane the car is currently travelling to. Assuming a maximum of 8 lanes, each lane is given an identifier starting from the 'North Outward' lane and going in counter-clockwise direction until the last lane ('West Inward') is reach.
3. **IntendedLaneID** – This is an important piece of information and is used by the algorithm to determine where the car intends to travel. Depending on the direction the car is heading for, the priority of the car and the time it takes to traverse through the lane would be changed.
4. **Distance from the junction** – Used by the scheduler to determine the order of each car in the each lane.
5. **TimeEarly** – The earliest possible time the vehicle can arrive at the junction based on the current and maximum speed of the vehicle.

5.3.2 Process

As mentioned briefly in section 4.3, the team initially looked at using a purely deterministic optimisation algorithm (henceforth referred to as A) which obtained various parameters for each car

and determined the optimum route for each car. This outputs a vector of displacement profiles, one for each car. One of the benefits of this approach is that it can be modelled as a single function in simulation, allowing all nodes in the network to reach the same decision with no ambiguity. The disadvantage is that each car requires a nontrivial amount of computing power and each node must replicate the same calculations. Even though the current projection of self-driving cars predicts that each car would have powerful hardware [14], the team leaned towards a different approach. Another reason as to why the team decided against this approach was since a large amount of car parameters would have to be transmitted over the network to maximise performance of the algorithm such as car efficiency graphs, acceleration profiles etc. It was thus decided that the approach outlined below should be taken instead.

The final approach was to split the junction into a 2x2 grid and give each tile an identifier. Cars would then book time for each tile depending on their desired turn location so a car turning left would book a single tile, a car going straight would book the two adjacent tiles to get it across the intersection with a staggered pair of times, a car turning right would book the three tiles it requires to traverse the intersection in staggered time slots. The scheduler would determine a time slot that was free and communicate this with the car.

From the perspective of the car:

1. Determine current speed and distance from the intersection and calculate the earliest possible time the current car can arrive at the intersection.
2. Send this time to the distributed scheduler. The scheduler will return with a time slot.
3. Change speed to make that time slot. Attempt to stay as close to the car in front as is safe.

From the perspective of the scheduler:

1. Receive the earliest time that a car can arrive.
2. Schedule the car into the earliest possible free time slot. Recalculate time slots if necessary.
3. Look for any optimisation opportunities.
4. Communicate this data back to the car that has book the time slot as well as all other cars whose time slots have been changed.

Safety considerations:

To ensure that the highest safety standards are adhered to, the algorithm will leverage and rely upon the self-driving technology of the cars themselves to stay safe. Each car will be responsible for detecting and preventing collisions. This is achieved by loosely coupling the intersection and self-driving car algorithms where the intersection algorithms provide a service rather than instructions.

The team designed this algorithm as an amalgamation of research concepts that have already been implemented. The core algorithm concepts most closely resemble the Hybrid Approach that H. Kowshik, D. Caveney and P. R. Kumar detailed [15]. The team decided to innovate by taking a completely distributed approach using distrusted database and ad-hoc elected master based on the distance to the intersection. This was seen as a good idea as the mater node will thus be approximately central and there will be enough time to elect a new master by the time the master has left the junction range. This slot based system and election of a master ensures that deadlock scenarios cannot happen

Clearing a scheduled system in minimal time is an NP-Hard problem as proven by A. Giridhar and P. R. Kumar [16]: in fact, Job scheduling with conflicts is well studied, and is known as one of the 21 famous NP-Complete problems. The team decided to loosely book time slots based on a FCFS (First

Come, First Served) basis in order to prevent situations where a long stream of cars would make a lone car wait for a large amount of time to cross. The team aimed to make the intersection fair, in that the maximum waiting time should be prioritised over throughput. However, the team experimented the BATCH algorithm [17] in a limited capacity with the aim to increase throughput. Batches of nodes were limited to sizes of 5 to limit the maximum wait time.

This algorithm decouples the scheduling from the base algorithm. This means that the scheduling algorithm and optimisation strategy can be separately scaled with advances in the academic space. The team researched an evaluated a variety of different scheduling algorithms: AIM based reservation [18], Genetic algorithms [19], SchIC [20] and VANET [21]. The team were reluctant to train a genetic algorithm and thus the option was considered but not implemented due to a lack of real world training data. A reservation system where a car books a slot and the intersection controller either confirms or denies requests restricts the optimisation abilities.

As suggested in VANET, the team used a combination of an Oldest first job algorithm and a greedy strategy to assign slots to each car. The car sends the earliest slot that it can possibly make to each junction segment that it needs to pass through and then the scheduling system assigns a slot closest to the car's ideal speed. Cars are added to the schedule when they are approximately 200 m from the intersection. The system checks for conflicts by using a discrete set of time intervals. This was determined to be the best strategy over the use of scheduling algorithms defined by SchIC due to the simplicity of the solution matching the project's time constraints.

The algorithm then checks for conflicts and shifts cars slots around to squeeze all cars in to the schedule for each junction square and then updates all cars whose slots have been adjusted. Cars with the same lane and destination are batched together in groups whose maximum size is 5. 5 was determined to be an appropriate trade-off between efficiency and average wait time but this can be adjusted based on real world feedback when the solution is deployed. The scheduler has other constraints built in such as ensuring that a car will never be required to overtake another car in the same lane. Also, cars crossing multiple lanes will have directly staggered slots across the 4 intersection junctions to ensure that a car is not forced to wait in the intersection and thus block other routes.

5.3.3 Performance:

To evaluate the performance of application level algorithms, a group of performance metrics need to be decided upon. The team decided to choose the following:

- Computational Requirements: It is preferable to use an algorithm that has a lower computational requirement. This lowers the power demands of the appliance and/ or frees up computational budget for other areas.
- Reliability: Safety is a major consideration and thus the algorithm must be defined for all cases.
- Wait time: Little's Law [22] dictates that $L = \lambda W$ where L is the Number of customers in the in intersection, W is the average time a car is in the intersection and λ is the effective arrival rate. In order to minimise L and thus maximise the intersection throughput, one needs to minimise W. The goal is to minimise the mean wait time which in turn increases throughput, although, the maximum wait time will have to be establish to prevent the system from over optimising a stream of cars and neglecting smaller [17] intersecting streams of cars.
- Fuel Efficiency: By minimising the number of times the car has to stop and promoting smooth velocity transitions, each car can maximise its fuel efficiency and reduce the wear and tear of the vehicle.

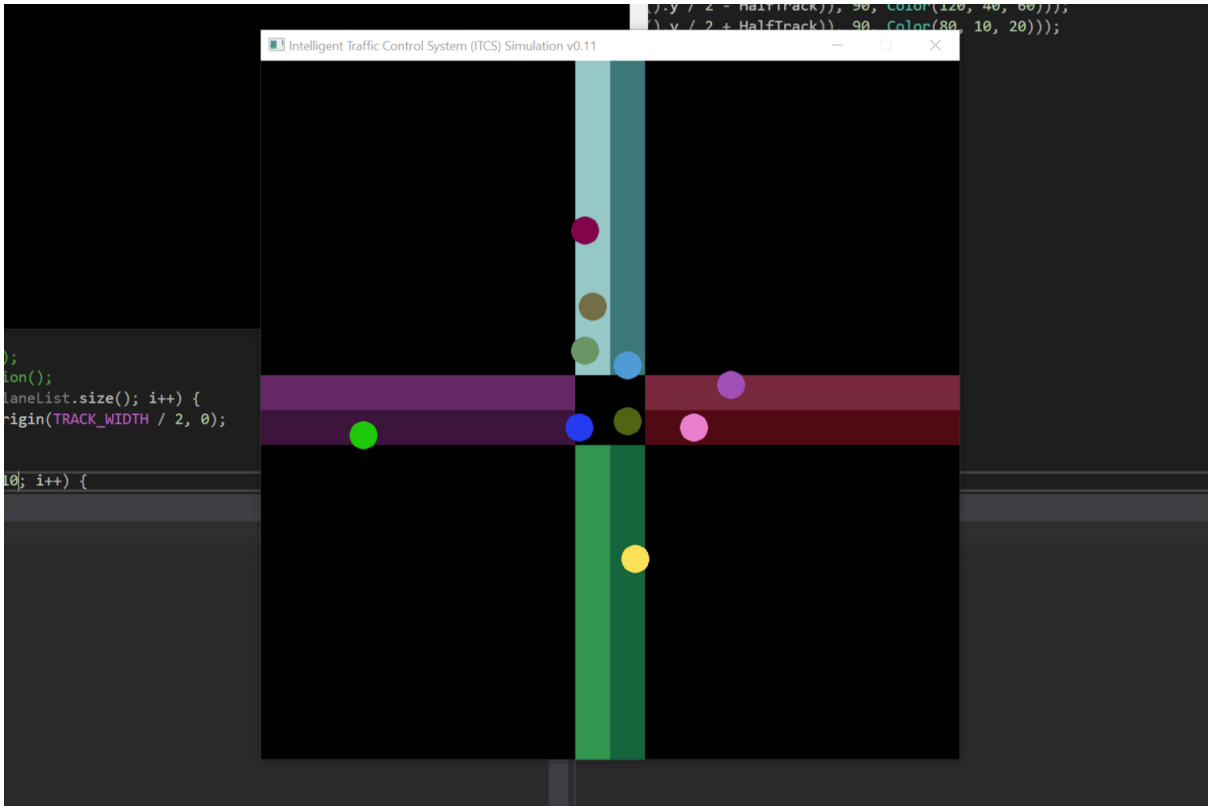
5.3.4 Simulation

In order to test and evaluate the viability and performance of the high level algorithms, the team decided to construct a simulation environment. This system was designed to be modular so that different algorithms can be swapped in and out. The benefit of a simulation rather than testing on hardware is that the team can scale the simulation in order to better simulate the real world as well as maintaining a separation of concerns: the application algorithms can be tested in isolation before they are integrated into the whole system.

To do the simulation, the team constructed a test environment with SFML [23], a graphics library written in C++. C++ was chosen for a variety of reasons: the mbed environment is C++-based, writing the simulation in C++ means algorithms could be ported to the environment with minimal effort; the team are all familiar with C++ which means that everybody can contribute; when advancing beyond basic test cases and introducing more real world parameters, performance should not be significantly affected; in the event that the team has to transition to new underlying hardware, there is a high probability that there exists a C++ compiler for the platform in question.

In the simulation, all vehicles are assumed to go a maximum of 25km/h and start off approximately 100m from the junction. The number of vehicles on in a 100m radius of the junction can be controlled to simulate different traffic flow densities. Each car is randomly assigned a lane and a destination. They then book a slot with the intersection controller as defined by the team's algorithm and edit their based on the time slot assigned to them. All cars would then attempt to pass through the junction at approximately 25 km/h. The maximum breaking ability was assumed to be 9 m/s^2 which was determined to be an accurate experimental value [24].

Some assumptions were made about the environment that simplified the model due to time and complexity constraints. The situation where a car must cancel its time slot due to a breakdown or object blocking its path has not been extensively testing or simulated.



5.3.5 Results/Observations

The simulation revealed that the group's application level algorithm worked: the number of collisions was zero. Also, the mean wait time of a single car was typically less than that of a traffic light. It was observed that in high traffic densities, the traffic patterns did approach that of a traffic light based intersection. The largest benefit was when there was a very low traffic density as expected. The mean wait time of a car was often 0 rather than a randomly distributed variable $\in [0, T]$ where T is the traffic light period. It was noticed that the deceleration that some cars experienced would likely cause passenger discomfort in the real world. The algorithm will have to be refined to reduce this effect.

5.4 Consumer Consideration

5.4.1 Socioeconomic Impact

The Mesh network based traffic light control would improve the context which it is placed in in many ways, one of which is that it makes travelling and traffic lights much more efficient, in that there would be less traffic jams, resulting in the fact that people who use it have more time and are less stressed, as a result of being late to their appointments or just wasting time idling on the road when they needn't be.

Additionally, if more time was provided and this project was improved upon in that many intersections / traffic control systems are interconnected, it could also redirect traffic based on which paths are being used the most and which are being used the least. For a busy city like London especially, this would be very helpful in that it reduces the drive for resources to manage high car density on the streets and redirection of traffic.

The finished product in making travel more efficient will encourage people to go out more often, from which spending will be increased. This is a positive note for the economy, as well as the fact

that consumers will generally go out more and engage in more activities than they otherwise would have, which is a societal improvement.

Another economic benefit is that there would not be any need for traffic signals in the future if this product is fully implemented; if traffic signals are removed then their maintenance and the costs associated with them are also removed, as well as any traffic jams caused by their errors.

Yet another benefit is that emergency vehicles can get through traffic signals a lot quicker as there is less congestion in general, resulting in a societal and social improvement.

5.4.2 Environmental Impact

Furthermore, as it reduces traffic jams and therefore engine idling time, it cuts down on emissions, as most fuel is mainly used while idling or when start-stopping, as the rapid acceleration uses more energy than needed (considering it stops again soon after) as well as the fact that when the engine is not fully running, incomplete combustion takes place and thus results in gases such as nitrous oxides, carbon monoxide, and unburnt hydrocarbons being released. If this stop-starting and idling is reduced by the traffic control device, then these gases could be prevented and thus it positively contributes to the environment.

Since it saves travel time and engine idling time, it therefore also saves a lot of fuel [25], which is both an economic and a social benefit in that it's cheaper for the user overall.

5.4.3 Manufacturing Considerations

Since the design is meant to be used in driverless cars, the size should not play that large of a role in the industrial design (could get in the way if it's too big in a manual car, by obstructing the drivers vision or just getting in the way) if it fulfils the criteria of being easy to move around and swap between cars as well as somehow being connected to the outside world, so a maximum size of approximately 150mmx100mmx50mm (length, width, and height, respectively) will be used.

To be able to be moved about easily yet still be stable when a position is selected, the housing will need a method of being secured to the car. Two possibilities were considered to resolve this issue – one of them was to use elastic straps with the device to connect it to the rear-centre console, but this method was disregarded because of the lack of standardization in cars regarding the rear centre console (in terms of height, distance to a window, clear radio line of sight (if sheet metal is used in the seats for example, which would block out the RF waves from that direction)), which results in an unpredictable signal strength, which might interfere with distance calculations or even overall communications. Another possible idea was to use a suction mount to mount the device to the bottom right corner of the windscreen to ensure minimum interference for the device from the car as well as obtaining standardised position data from all cars, as there is only minimal change between cars regarding the windscreen – the main variable expected to change is the height. To have a secure connection with the windscreen, four suction mounts will be provided. In terms of safety, since the front of the device is attached to the car, in any front facing crash at high speed, the resultant force on the device will force it to go further into the windshield rather than away, so there is very little danger of the device bouncing around and causing damage in a crash. Crashes where the cars travel at a high speed and collide with something in front that mainly throw about unrestrained objects at a high enough speed for it to cause significant damage, as side crashes or crashes into the back of a car don't have enough sustained force in a particular direction to undo the suction holders [26].

The main material used for the housing will be plastic, as it is easy to manufacture on a large scale as well as being easy to machine and customise the aesthetics of. Furthermore, it also allows RF waves

to penetrate through the housing easily. Regarding user appeal, five different colours will be used to match the user's car's interior. The four colours are, black, white, grey, cream, and brown, as these are the main dashboard colours and a large proportion of people would be satisfied with these colours. Edges that don't make contact with the windscreen will be chamfered and an overall sleek minimalist device will be shipped.

Regarding ergonomics, since this device is designed to be used with fully autonomous cars, ergonomics only comes to play in two main areas- battery replacement and the on off switch. To make thinks simple, the on off switch will be at the left side of the device so the user can turn it on as soon as they enter the vehicle with ease and turn it off when they are leaving without paying too much thought to the process.

For replacing the batteries easily, there will be an access hatch provided that will allow easy battery replacement. For this, part of the housing will be cut away (with laser cutters) and the hatch to replace it will be machined by using injection moulding on the Terramac plastic.

The material chosen for housing the PCB and the sensors is a biodegradable plastic resin called terramac. This material was chosen as it is environmentally friendly (completely biodegradable and minimal carbon footprint) [27]. It can be processed on a large scale and is available in a variety of colours.

Blow moulding will be used to manufacture the housing, as this plastic resin was designed to be used with that as well as being an economically viable [28] as well as a very energy efficient [29] way to produce mass numbers of hollow plastic products, like this housing. Specifically, the process used will be continuous extrusion blow moulding, similar to the way plastic bottles are produced.

To make the actual PCB, large scale automated PCB printing would be used to print the boards, and then silk screen printers deposit solder paste onto the boards where components are due to be mounted, after which pick and place machines place the surface mount components onto the right locations. Reflow ovens are then used to melt the solder past to create a permanent bond, after which they are automatically quality controlled by using automated optical inspection [30].

Terramac plastic foam will also be provided to hold the PCB and the sensors in the right position inside as well as protect the internal parts from impacts. For this, injection moulding will be used and the PCB will be secured to the foam by using plastic glue, all done autonomously, as well as combining the outer housing and the PCB-plastic foam combination.

5.4.4 Pricing

Currently the main components for the prototype are:

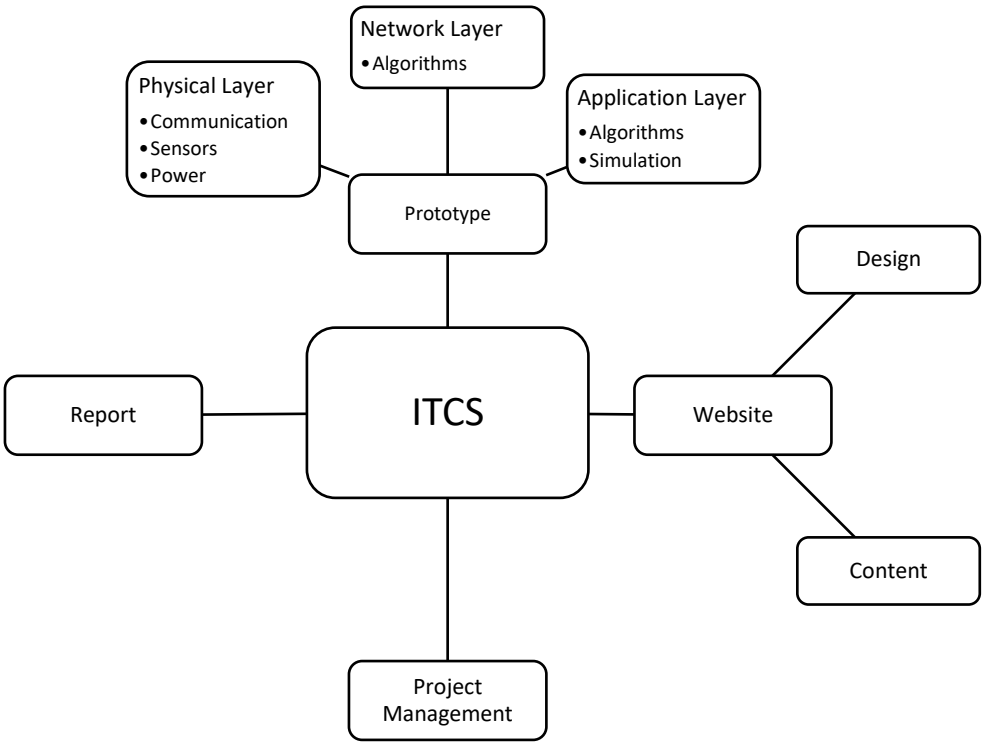
Name	Function	Price (£)
nRF24L01+	Communications module	1.96
LM2574N	SMPS voltage regulator	1.22
Passive components	Powering module	0.73
LSM303DLHC	Sensor module	5.95
ARM mbed module	Microcontroller module	46.80

The total cost per prototype module is £56.66. However, the retail price for the product is to be much higher, mainly due to the need to use much more accurate and reliable (and so expensive)

components, as well as additional sensors (such as GPS receiver or speed sensor). Moreover, labour costs need to be covered.

6 Project Management

The work for this project was divided based on each member's strengths (e.g. members proficient in programming were allocated software based tasks). The project was split into four main aspects: prototype, website, project management and report. The key working areas for each aspect of the project are outlined below:



In alphabetical order the contribution of each team member for different aspects of the project.

PROTOTYPE	Physical Layer	*Communication*	Arthur	Michal		
		Sensors	Haaris	Piers		
		Power	Ashish			
	Network Layer	*Algorithms*	Arthur	Michal	Mohika	Waleed
	Application Layer	*Algorithms*	Haaris	Mohika	Piers	Waleed
		Simulation	Haaris	Piers	Waleed	
WEBSITE	**Design**		Haaris	Piers		
	Content		Piers	Waleed		
PROJECT MANAGEMENT	**Task Distribution**		Haaris	Mohika		
	Team Organisation		Mohika			
REPORT	**Content**		All Members			
	Formatting + Structure		Haaris	Mohika		

The team held weekly meetings to compare each subgroup's progress with the project plan (see appendix section 11.4). To ensure every meeting was productive and applicable to all team

members, an agenda based on current project progress was created before each meeting to ensure any issues and miscommunications could be dealt with quickly. If any subgroups were unable to keep to the timings outlined in the project plan, team members making progress ahead of time and members with a lighter workload would be tasked with ensuring all subgroups were making good progress.

7 Future Work

- Test the mesh network with a greater number of nodes
- Use data from GPS modules to determine location
- Investigate efficiency
- Further develop the application layer to work for multi-lane and staggered intersections.
- Acceleration/deceleration
- Variable speed in lane

8 Conclusion

The report has presented tests and designs leading to implementing a sample mesh network system for intersection traffic control. It included the approach leading to creating a simulation of a quite complex system, as well as hardware implementation of a much simpler one. It was shown that such a system is feasible to implement, and that its use can enhance performance of autonomous vehicles, which are undoubtedly the future of transportation.

Of course, to be suitable for that, the network, its algorithms and the hardware it uses must all be extremely reliable, which is the case for all equipment used in the automotive industry. Therefore, in the future the group will focus on moving the system to a more dependable hardware platform and improving efficiency and accuracy of intersection management algorithms to achieve a final aim of having a system that can be used in real cars.

9 References

[1] T. T. Team, “All Tesla cars being produced now have full self-friving hardware,” [Online]. Available: https://www.tesla.com/blog/all-tesla-cars-being-produced-now-have-full-self-driving-hardware. [Accessed 10 03 17].

[2] “Autonomous Intersection Management: Multi-Intersection Optimization,” [Online]. Available: http://www.cs.utexas.edu/~aim/papers/IROS11-hausknecht.pdf. [Accessed 05 02 2017].

[3] Wikipedia, “ZigBee,” [Online]. Available: https://en.wikipedia.org/wiki/ZigBee. [Accessed 12 2 2017].

[4] TEM, “HOPE MICROELECTRONICS RFM12B 868S2,” [Online]. Available: http://www.tme.eu/gb/details/rfm12b_868s2/rf-communication-modules/hope-microelectronics/rfm12b-868s2/. [Accessed 01 03 2017].

[5] CHRobotics, “Using Accelerometers to Estimate Position and Velocity,” [Online]. Available: http://www.chrobotics.com/library/accel-position-velocity. [Accessed 2017 03 05].

[6] O. Edwards and ARM, “ARM mbed developer website,” [Online]. Available: https://developer.mbed.org/components/nRF24L01/. [Accessed 27 2 2017].

[7] Farnell, “mbed NXP LPC11U24,” [Online]. Available: http://www.farnell.com/datasheets/1520737.pdf?_ga=1.156453821.740724533.1489402657. [Accessed 20 02 17].

[8] T. Instruments, “LM2574x Datasheet,” [Online]. Available: http://www.ti.com/lit/ds/symlink/lm2574.pdf. [Accessed 01 03 17].

[9] NXP, “LPC11U2x,” [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC11U2X.pdf. [Accessed 15 01 2017].

[10] “nRF24L01+ Datasheet,” [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf. [Accessed 2017 02 15].

[11] STMicroelectronics, “LSM303DLHC Datasheet,” [Online]. Available: http://bsfrance.fr/documentation/10351_GY511_LSM303DLHC/LSM303DLHC.pdf. [Accessed 20 01 2017].

[12] “Insteon,” [Online]. Available: https://en.wikipedia.org/wiki/Insteon. [Accessed 12 1 2017].

[13] “LM2574, NCV2574,” [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/14f5/0900766b814f5606.pdf. [Accessed 2017 02 11].

[14] Nvidia, “Self-Driving Vehicles Development Platform,” [Online]. Available: http://www.nvidia.com/object/drive-px.html. [Accessed 06 03 2017].

[15] D. C. a. P. R. K. H. Kowshik, “Provable Systemwide Safety in Intelligent Intersections,” *IEEE Transactions on Vehicular Technology,* vol. 60, no. 3, pp. 804-818, 2011.

[16] A. G. a. P. R. Kumar, “Scheduling Automated Traffic on a Network of Roads,” *IEEE Transactions on Vehicular Technology,* vol. 55, no. 5, pp. 1467-1474, 2006.

[17] P. S. ,. S. S. L. I. R.-C. E. F. D. H. C. R. Remi Tachet, “Revisiting Street Intersections Using Slot-Based Systems,” 2016.

[18] D. G. M. I. H. M. Z. a. C.-N. C. Kartik Pandit, “Adaptive Traffic Signal Control With Vehicular Ad hoc Networks,” *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY,* vol. 62, no. 4, p. 1459, 2013.

[19] G. JANSSON, “Traffic Control with Standard Genetic Algorithm,” *CHALMERS,* vol. Report No. 2010:127, 2010.

[20] S. F. S. L. L. G. J. B. Xiao-Feng Xie, “Schedule-driven intersection control,” *Transportation Research Part C,* no. 24, p. 168–189, 2012.

[21] H. V.Hemakumar, “Optimized Traffic Signal Control System at Traffic Intersections,” *IOSR Journal of Computer Engineering (IOSR-JCE),* vol. 15, no. 3, pp. 36-43, 2013.

[22] J. D. L. a. S. C. Graves, “Little's Law,” [Online]. Available: http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf. [Accessed 05 03 2017].

[23] SFML, “SFML,” [Online]. Available: http://www.sfml-dev.org/index.php. [Accessed 28 02 2017].

[24] N. Kudarauskas, “Analysis of emergency braking of a vehicle,” [Online]. Available: http://www.tandfonline.com/doi/pdf/10.1080/16484142.2007.9638118. [Accessed 08 03 17].

[25] G. Smith, “Why does my car use less fuel on the highway than it does around town?,” [Online]. Available: https://www.carsguide.com.au/car-advice/why-does-my-car-use-less-fuel-on-the-highway-than-it-does-around-town-35369. [Accessed 06 03 2017].

[26] M. Y. A. L. Daniel Avrahami, “The danger of loose objects in the car,” [Online]. Available: https://pdfs.semanticscholar.org/b1f9/a9d35ffee36fe4ba28265852fc92be8b7eb2.pdf.

[27] UNITIKA, “What's Terramac?,” [Online]. Available: https://www.unitika.co.jp/terramac/e/what/index.html. [Accessed 10 03 17].

[28] C. Technology, “Moulding the Future,” [Online]. Available: http://webcache.googleusercontent.com/search?q=cache:YvCmWEnfb8QJ:www.cleanroomtechnology.com/technical/article_page/Moulding_the_future/52816+&cd=4&hl=en&ct=clnk&gl=uk. [Accessed 06 03 2017].

[29] P. News, “Blow molding equipment makers turn up the efficiency,” [Online]. Available: http://www.plasticsnews.com/article/20150225/NEWS/150229960/blow-molding-

equipment-makers-turn-up-the-efficiency. [Accessed 06 03 2017].

[30] Zentech, “Surface Mounted Technology Makes Manufacturing Processes Smooth,” [Online]. Available: http://info.zentech.com/blog/bid/248318/Surface-Mounted-Technology-Makes-Manufacturing-Processes-Smooth. [Accessed 06 03 2017].

[31] Road Cover, “Intersections and Safe Driving,” 1 January 2016. [Online]. Available: http://www.roadcover.co.za/intersections-safe-driving/. [Accessed 29 January 2017].

[32] TechTarget, “Network Node,” [Online]. Available: http://searchnetworking.techtarget.com/definition/node. [Accessed 11 01 2017].

[33] H. RF, “RFM12B Datasheet,” [Online]. Available: http://cdn.sparkfun.com/datasheets/Wireless/General/RFM12B.pdf. [Accessed 25 01 2017].

[34] Nortek-AS, “Acoustic Doppler Velocimeters,” [Online]. Available: http://www.nortek-as.com/en/products/velocimeters. [Accessed 22 1 2017].

[35] BitBox, “Battery Showdown Results: Low Drain,” [Online]. Available: http://www.batteryshowdown.com/results-lo.html. [Accessed 05 02 2017].

[36] Z. Alliance, “ZigBee IP and 920IP,” [Online]. Available: http://www.zigbee.org/zigbee-for-developers/network-specifications/zigbeeip/. [Accessed 9 1 2017].

[37] *. J. E. N. a. Ó. G. Felipe Jiménez, “Autonomous Manoeuvring Systems for Collision Avoidance on Single Carriageway Roads,” *Sensors (Basel),* vol. 12, no. 219587, 2012.

[38] U. Government, “GPS Accuracy,” [Online]. Available: http://www.gps.gov/systems/gps/performance/accuracy/. [Accessed 25 01 2017].

[39] Lifewire, “Z-wave,” [Online]. Available: https://www.lifewire.com/what-is-z-wave-817700. [Accessed 12 1 2017].

[40] U. D. o. Transportation, “Traffic Safety Facts,” [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115. [Accessed 5 02 2017].

[41] M. Yan, “Dijkstra's Algorithm,” 26 March 2016. [Online]. Available: http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf. [Accessed 29 January 2017].

[42] Chuck, “Robotics Stack Exchange,” [Online]. Available: http://robotics.stackexchange.com/questions/9988/instantaneous-velocity-calculation-from-accelerometer. [Accessed 01 03 2017].

[43] “Inertial Navigation System,” [Online]. Available: https://en.wikipedia.org/wiki/Inertial_navigation_system. [Accessed 01 03 2017].

[44] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/ZigBee.

[45] Webopedia. [Online]. Available: http://www.webopedia.com/quick_ref/OSI_Layers.asp.

[46] "The 7 Layers of the OSI Model," [Online]. Available: http://www.webopedia.com/quick_ref/OSI_Layers.asp. [Accessed 05 03 2017].

[47] "Understanding ESR in Electolytic Capacitors," [Online]. Available: https://abacus.avnet.com/wps/portal/abacus/article/understanding%20esr%20in%20electrolytic%20capacitors. [Accessed 2017 02 14].

[48] "0.5 A, Adjustable Output," [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/14f5/0900766b814f5606.pdf. [Accessed 2017 02 20].

10 Appendix

10.1 7 Layer OSI Model for Main Modules

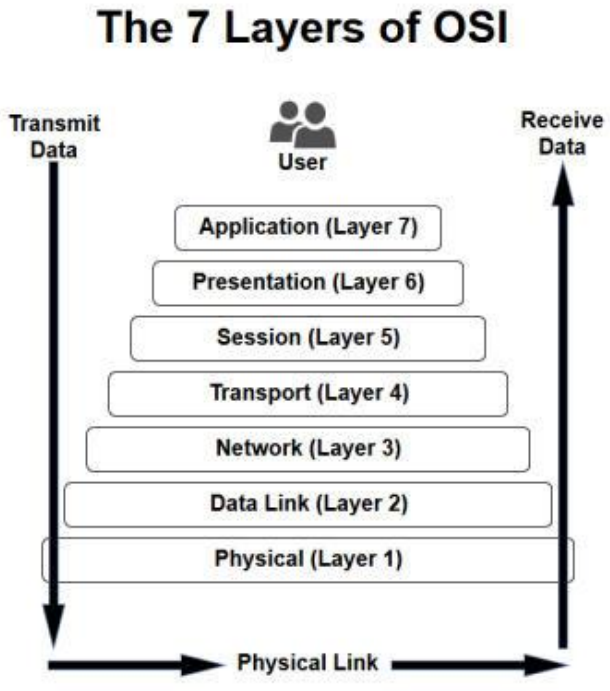


Figure: Image describing the 7-layer OSI model

10.2 Processing of Raw Sensor Data

1. Get Yaw (Heading), Pitch and Roll
   These three quantities determine the rotation of an object. It is necessary to perform tilt compensation. This is because even minor tilts cause the contact force F_c to produce a component along either of the x, y, or z-axis. This contact force exists as a direct reaction to the Earth's gravitational pull. In a 3D space there can be two main types of rotations (pitch and roll) that might affect A_{raw} at rest. The image below illustrates each of these rotations for a formula

1 car. Even though cars in general won't have large pitch or rolls, there can be some rotation due to tilts in the road design for safety and other purposes.

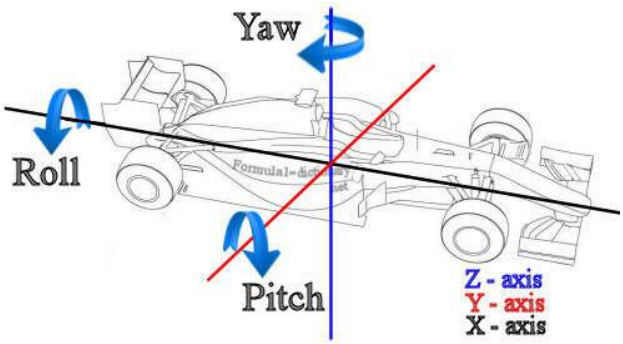


Figure 8: Yaw, Pitch and Roll (ψ-ϑ-φ)) for a car

If the pitch, roll and yaw angles are represented as $\theta, \phi \; and \; \psi$ respectively, the sensor at rest outputs:

$$A_{raw} \approx \begin{bmatrix} sin\theta \\ sin\phi \\ cos\theta \end{bmatrix}$$

For reference, there is no component of gravity along the x or y axes on a flat surface and hence no contact force. The only contact force is the normal contact force which is along the z-axis and has magnitude of $\cos(0°) = 1$ $(here \; 1 = 1[g] = 9.81[ms^{-1}])$.

The pitch (θ) can be calculated as following:

$$\tan(\theta) = \frac{-A_x}{\sqrt{(A_y)^2 + (A_z)^2}}$$

The roll (ϕ) can be calculated as following:

$$\tan(\phi) = \frac{A_y}{A_z}$$

The yaw (ψ) which is also called the 'heading' can be calculated as following (M is the magnetic flux density vector):

$$\tan(\psi) = \frac{M_y}{M_x}$$

2. Calibrate error
   Once θ and ϕ are calculated a bias vector is generated when the car is at rest (with same values as A_{raw}). The bias vector is then separated from the main acceleration vector to give only the linear acceleration. Due to physical constraints bias values can only be calculated when there are no external forces.

3. Numeric Step Integration

A simple numeric step integration method is used to calculate velocity and displacement from acceleration. The formulas are:

$$V_{i+1} = V_i + A_{i+1} \cdot g \cdot \delta t$$
$$X_{i+1} = X_i + V_{i+1} \cdot \delta t$$

Where δt is defined as the time-step and is determined by the wait time for any given loop iteration and $A_{i+1}, V_{i+1}\ and\ X_{i+1}$ are the new acceleration, velocity and displacement vectors after time t= δt. The acceleration vector is scaled by a factor of 'g' as the sensor outputs data in terms of g.

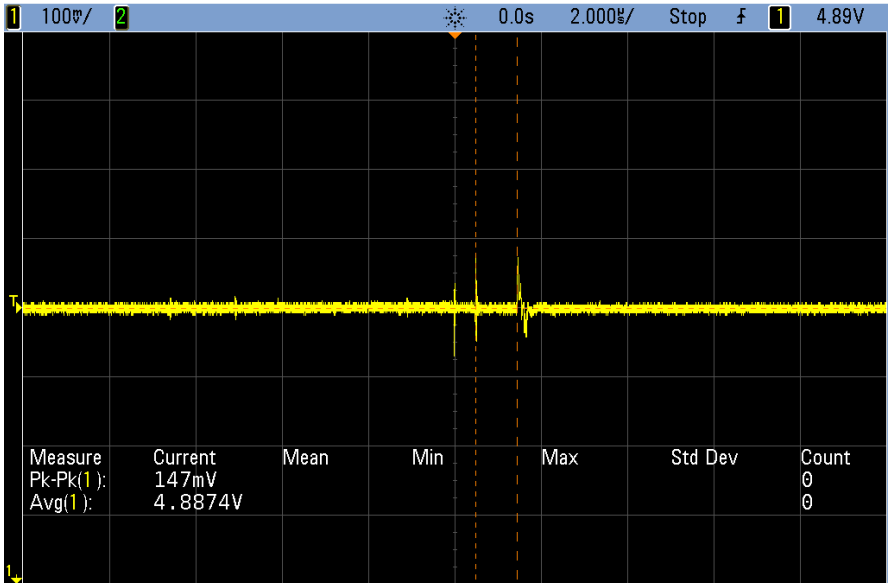
4. Speed and Distance
   The final part of the processing involves getting the absolute value of the velocity and displacement only in the XY plane since it can be assumed that cars don't travel along the z-axis. The reason why direction can be ignored is because it can be determined from heading which is given by yaw.

10.3 Power Calculations:

Using the formula for keeping it above discontinuous conduction, $L \geq \frac{V_o\left(1-{}^{V_o}/_{V_i}\right)}{2fI_o}$, an inductance value of 400 µH is derived, however, as the chip uses feedback to match the output power to the size of the load, slightly smaller inductor values will also be fine without leading to discontinuous conduction. The value for 3.3 volts, the initial test value, was 120 µH, which corresponds with the value provided in the datasheets. Testing at this value however, provided a lot of noise, so when shifting the output level to 5 volts, the inductors were also changed to 330 µH (also the recommended value for a 5 V output according to the datasheet), which was just slightly less than what the above equation suggested. This value yielded much lower noise.

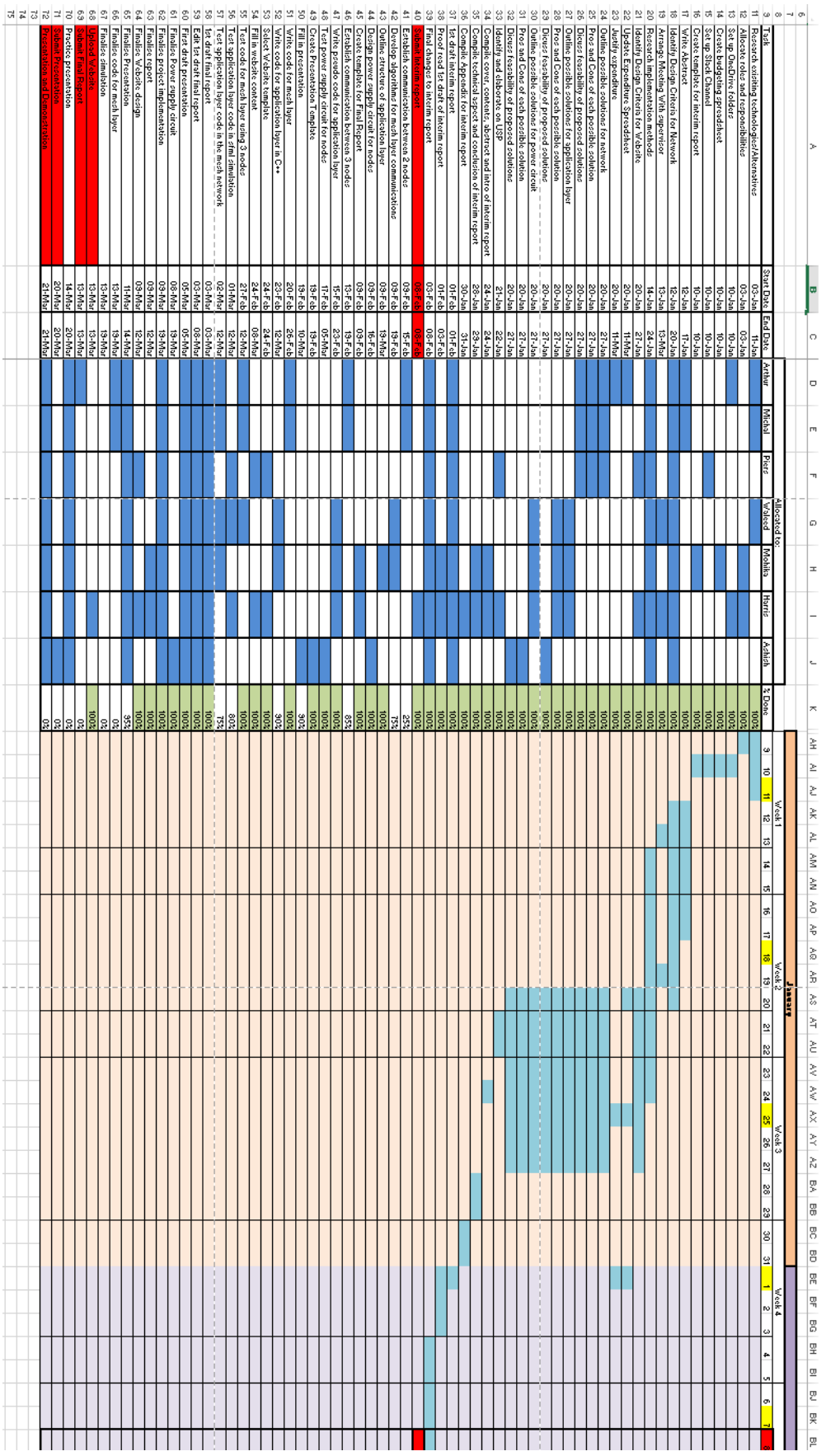
The effective series resistance of a normal electrolytic capacitor from 22 µF to about 500 µF amounts to somewhere between 0.1 – 0.05 at 50 kHz [4], and at a current draw of 20 mA or so the voltage ripple held across the ESR of a capacitor doesn't amount to much (2mV – 1mV (using the formula VESR=I*RC)), so this can be safely ignored.

The voltage ripple caused by the capacitance can be calculated by using this formula: $\Delta V_c = \frac{\Delta I}{8fC}$, from which using 20mA as the current drawn, 52 kHz as the frequency and 220 µF (the recommended value in the datasheet), the voltage ripple is found to be 213mV, but only 200 mV was observed in practice. However, in order to sustain any additional current from the Mbed board or to sustain the potential use of motors in the prototype, a 470 µF capacitor was used, which yielded an ideal voltage ripple (using the above values) of 99.7 mV.

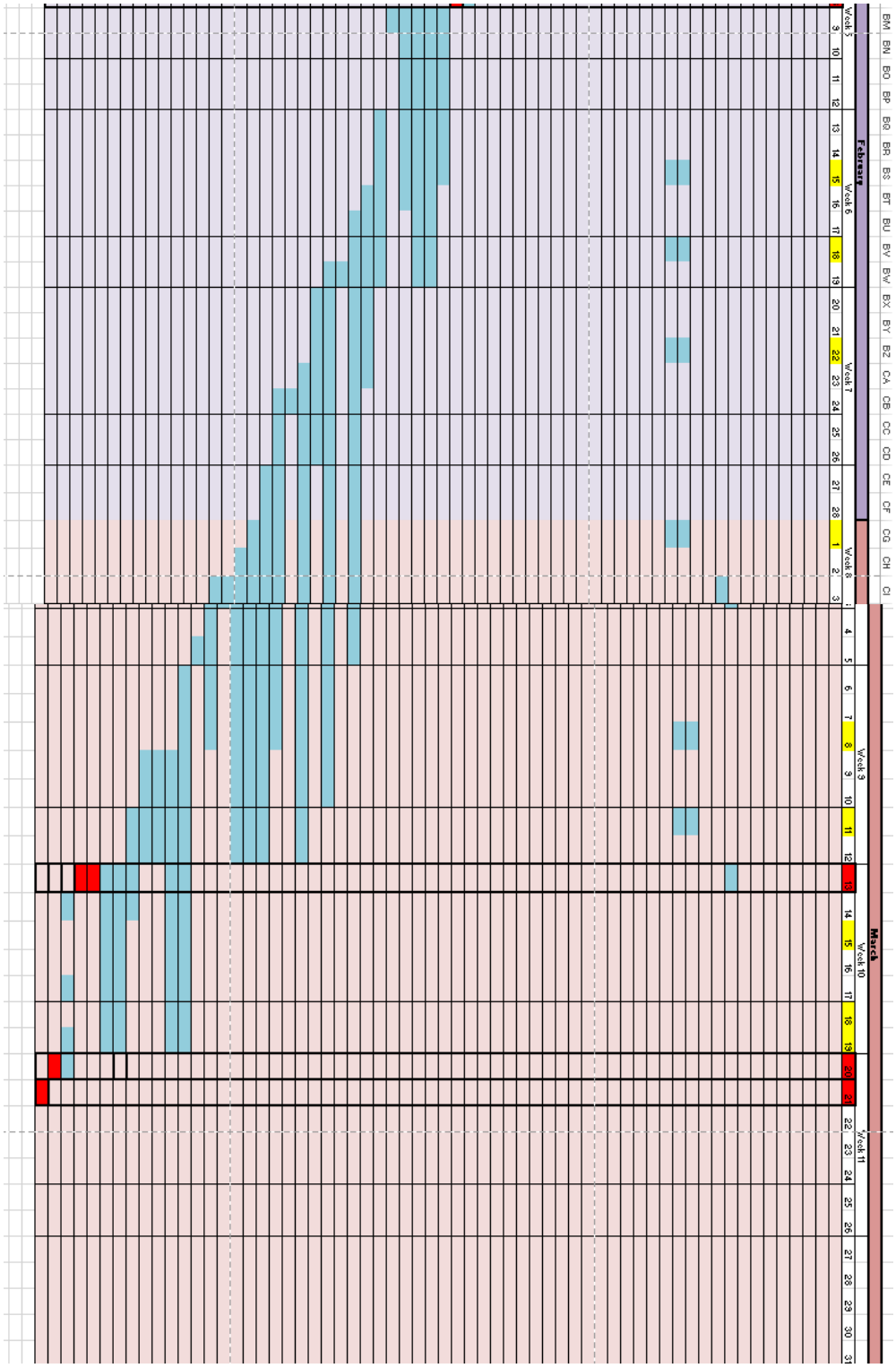


This is the waveform obtained when used with a 5V output, 6V input, 330 µH inductor and a 220 µF capacitor.

10.4 Project Plan



Task	Start Date	End Date	Allocated to: Arthur	Michal	Piers	Waleed	Mostiks	Harris	Ashish	% Done
Research existing technologies/Alternatives	03-Jan	11-Jan								100%
Allocate roles and responsibilities	03-Jan	03-Jan								100%
Set up OneDrive folders	10-Jan	10-Jan								100%
Create budgeting spreadsheet	10-Jan	10-Jan								100%
Set up Slack Channel	10-Jan	10-Jan								100%
Create template for interim report	10-Jan	10-Jan								100%
Write Abstract	12-Jan	17-Jan								100%
Identify Design Criteria for Network	12-Jan	20-Jan								100%
Arrange Meeting with supervisor	13-Jan	13-Mar								100%
Research implementation methods	14-Jan	24-Jan								100%
Identify Design Criteria for Website	20-Jan	27-Jan								100%
Update Expenditure Spreadsheet	20-Jan	11-Mar								100%
Justify expenditure	20-Jan	27-Jan								100%
Outline possible solutions for network	20-Jan	27-Jan								100%
Pros and Cons of each possible solution	20-Jan	27-Jan								100%
Discuss feasability of proposed solutions	20-Jan	27-Jan								100%
Outline possible solutions for application layer	20-Jan	27-Jan								100%
Pros and Cons of each possible solution	20-Jan	27-Jan								100%
Discuss feasability of proposed solutions	20-Jan	27-Jan								100%
Outline possible solutions for power circuit	20-Jan	27-Jan								100%
Pros and Cons of each possible solution	20-Jan	27-Jan								100%
Discuss feasability of proposed solutions	20-Jan	27-Jan								100%
Identify and elaborate on USP	21-Jan	22-Jan								100%
Compile cover contents, abstract and intro of interim report	24-Jan	24-Jan								100%
Compile technical aspect and conclusion of interim report	28-Jan	29-Jan								100%
Compile Appendix for interim report	30-Jan	31-Jan								100%
1st draft interim report	01-Feb	01-Feb								100%
Proof read 1st draft of interim report	01-Feb	03-Feb								100%
Final changes to interim report	03-Feb	06-Feb								100%
Submit interim report	06-Feb	06-Feb								100%
Establish communication between 2 nodes	03-Feb	15-Feb								100%
Develop algorithms for mesh layer communications	03-Feb	19-Feb								75%
Outline structure of application layer	03-Feb	19-Mar								100%
Design power supply circuit for nodes	03-Feb	16-Feb								100%
Create template for Final Report	03-Feb	03-Feb								100%
Establish communication between 3 nodes	15-Feb	19-Feb								85%
Write pseudo code for application layer	15-Feb	23-Feb								100%
Test power supply circuit for nodes	17-Feb	05-Mar								100%
Create Presentation Template	19-Feb	19-Feb								100%
Fill in presentation	19-Feb	10-Mar								30%
Write code for mesh layer	20-Feb	26-Feb								100%
Write code for application layer in C++	23-Feb	12-Mar								30%
Select Website template	24-Feb	24-Feb								100%
Fill in website content	24-Feb	06-Mar								100%
Test code for mesh layer using 3 nodes	27-Feb	12-Mar								100%
Test application layer code in final simulation	01-Mar	12-Mar								80%
Test application layer code in the mesh network	02-Mar	12-Mar								75%
1st draft final report	03-Mar	03-Mar								100%
Edit 1st draft final report	03-Mar	06-Mar								100%
First draft presentation	05-Mar	05-Mar								100%
Finalise Power supply circuit	06-Mar	19-Mar								100%
Finalise project implementation	09-Mar	19-Mar								100%
Finalise report	09-Mar	12-Mar								100%
Finalise Website design	09-Mar	12-Mar								100%
Finalise Presentation	11-Mar	14-Mar								35%
Finalise simulation	13-Mar	19-Mar								0%
Finalise code for mesh layer	13-Mar	19-Mar								0%
Upload Website	13-Mar	13-Mar								0%
Submit Final Report	13-Mar	13-Mar								100%
Practice presentation	14-Mar	20-Mar								0%
Submit Presentation	20-Mar	20-Mar								0%
Presentation and Demonstration	21-Mar	21-Mar								0%


February
March
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11