University of Southern Queensland
Faculty of Health, Engineering & Sciences

Wireless Mountain Bike Safety System

A dissertation submitted by

Simeon Kelly

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Honours) (Electrical & Electronic Engineering)

Submitted: October 2023

Abstract

Mountain biking has always been considered a dangerous sport, as the rider not only has to navigate the trail safely but must also consider the other trail users. In some parks, two-way or multi- purpose tracks are often used. This adds an extra level of danger as riders need to be very mindful of oncoming traffic or hikers going in the same direction as them. According to a typology of bike crashes, 58% of crashes occur as the rider does not have time to manoeuvre and over 45% of them are crashes into another vehicle. (Billot-Grasset et al., 2014) This project focuses on the design of an indication system to alert bikers when there are other vehicles or cyclists ahead, especially when narrow and technically challenging portions of trail are concerned.

This project has an important contribution to make to the safety of trail users, without them needing special equipment or to remember to do something. Existing systems to help improve trail safety have required the rider to wear something to be tracked while systems for counting bikes have been focused on applications such as traffic monitoring and are not particularly useful at measuring direction or to a high accuracy (Ohlms et al., 2019). This system shall aim to provide a passive method to accurately determine to the presence and direction of mountain bikers.

The approach taken has been to identify potentially suitable sensor technologies and determine the requirements for their use in this context. Based on the preliminary analysis, three sensor types have been tested under constrained conditions. These were the vision, audio, and vibration sensors. All three were tested using a bike past them and had varying levels of success. The audio system was the only one which was able to pick up the direction of the bicycle however the vision was the most consistent at picking up movement. All three sensor types have possible ways of improving them, however the most recommended solution is the audio sensor. The major caveat with this is that is was only tested on one bike, and may not work reliably on all bikes.

The key outcome from this project is to be able to reliably detect presence and direction of mountain bikes on a trail. This system will be used to warn the cyclists of oncoming traffic and would be best suited to installation on both ends of a technically difficult portion of trail.

University of Southern Queensland

Faculty of Health, Engineering & Sciences

ENG4111/2 Research Project

Limitations of Use

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged. I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Simeon Kelly

Acknowledgments

My deepest gratitude goes to my parents, Peter and Jennifer. You kept me on task and exited to complete this and it would not have been possible without you both.

Secondly, I would like to thank my good friends Jazz Isaac and Matthew, who are always willing to study and push me through this challenging time.

I would also like to thank Chase Richardson for the idea of this project.

Finally, to my UniSQ supervisor Catherine Hills. The support, guidance and feedback I received from has been outstanding throughout this year, and I thank you for your care.

Simeon Kelly

Contents

Introduction

Background information

Mountain biking has always been considered a dangerous sport, and rightly so, as the rider not only has to navigate the trail safely but must also consider the other trail users. In less popular parks, two-way or multi- purpose tracks are often used.

This adds an extra level of danger as riders need to be very mindful of oncoming traffic or hikers going in the same direction as them. According to a typology of bike crashes, 58% of crashes occur as the rider does not have time to manoeuvre and over 45% of them are crashes into another vehicle.(Billot-Grasset et al., 2014).

This thesis focuses on the construction of an indication system to help bikers identify when there are other vehicles and cyclists ahead, especially when narrow and technically challenging portions of trail are concerned.

To accomplish this, the presence and direction of cyclists and other vehicles must be identified, and their presence must be signalled to anyother users on the path in front to notify them of the hazardous conditions.

The proposed system consists of multiple nodes along a mountain bike trail. These nodes would consist of sensors to detect the presence and direction of a bike, and a piezo buzzer or speaker to notify the cyclist of other trail users. An internal microcontroller would process the data and transmit to other nodes via a communication module All being powered by a battery, which is charged by an external solar. This must all be enclosed inside an enclosure which is both weatherproof and able to withstand mild vandalism attacks.

For example, a separate node could be installed at each end of a technical passage of mountain bike trail. When one node senses a vehicle passing into that section, it transmits a message to the node on the other end of the section, which would then alert any rider entering the trail from that end. To prevent confusion, the system should indicate both the presence or absence of other trail users on the section.

The most critical part of this system is being able to reliably sense the presence and direction of any passing bike reliably and in a power efficient manner. A robust system should not be triggered by erroneous signals from wildlife or movement by wind and should be able to detect in all weather and lighting conditions.

This shall be the first objective of this thesis, as other than this safety system implementation, a configuration of sensors which are able to detect the presence and direction of cyclists reliably and accurately can have many uses in data collection and larger systems.

Many systems have been implemented for counting bikes using various sensors, however these systems are focused on counting mass numbers of bikes and less on being very reliable at counting singular bikes as shown by a review completed of the current practice in the USA over the last decade (Ohlms et al., 2019). They are also often quite expensive to implement and require a live view of the count of bikes.

This thesis shall use the sensor configurations used in the past as inspiration for sensors which may be helpful, however none of the entire sensor configurations can be entirely copied.

Project Aims

In this thesis, I shall be ideally creating a system which reliably warns bikers of oncoming traffic and other riders. This is a very large final goal, so I have broken it up into a few objectives which I may or may not include in the final thesis as they may or may not be completed.

The first stage of the project is to conduct research into the previous systems used to sense bicycles and their current applications. Once these have been reviewed a sensor technology or multiple technologies which can be used to implement the bike sensing effectively. This will include deciding the location and type of sensors to be used. A variety of different sensors shall be ordered, and differing configurations shall be tested for their effectiveness and efficacy in sensing riders' presence and direction.

Ideally the sensors should be able to be installed together and factors such as computational power required, the sensors power draw and their robustness shall be considered.

Once the configuration and type of the sensors has been determined, a suitable supporting microcontroller or Single board computer should be chosen for the application. The power draw of these devices is also considered as upon implementation they will need to be powered form a battery and solar setup.

Once the computing board and sensors have been setup, the system shall be tested for efficacy in terms of its reliability to sense the bikes and its robustness.

If it is successfully good at sensing the presence and direction of the bike, a communication technology may be chosen to connect between the differing nodes and differing network configurations shall be considered, Power use and reliability shall be considered when deciding the best communication technology to use.

If this can be implemented successfully a prototype may be packaged in a watertight container and take to the field to be used.

These are very broad aim of this thesis but below I shall state the specific objectives these aims translate into and give good reasons for their use.

Project Objectives

The specific project objectives which have been laid out in the project specification shall be numbered and explained below. Please note that some of these have been allocated to be prioritised last as they would only make sense if time and resources permit within the resources allocated to this thesis.

1. *Conduct initial background research bike sensing technologies and their applications in current research, ideally finding a technology which has been used to sense speed and direction of bikes.*

This initial research shall be presented in the final thesis as a completed literature review and is presented below as a partially completed literature review.

2. *Review existing bike sensing systems and their underlying technologies. Consider the use case of the technologies and their running requirements and decide on one or more sensing technologies which would be appropriate to use in this system.*

This shall also be covered in the literature review but should include specific sensors to buy and configurations to test. A weighted matrix shall be used to decide on the technologies which shall be used in the final project.

3. *Conceptualize a suitable configuration for the position and type of sensors to reliably sense the presence and direction of bikes passing in front of them with minimal false positives and good reliability.*

This should be after the sensors have been ordered, a mounting system should be developed for them. a system for testing and ranking their efficacy shall also be used to determine the most reliable combination of sensors in the real world.

4. *Assess sensor requirements for necessary capability of system and costs.*

Once a sensor topography and technology has been chosen, a supporting mobile system should also be implemented. This shall depend on the chosen sensors as some may require more processing than others, such as image or audio sensors. The power consumption and ease of programming shall also be considered as devices such as a pic microcontroller or industrial PLC's are technically able to control the sensor however they require quite expensive development packages.

5. *Select sensors, a software development environment and microcontroller.*

A final system configuration should be chosen with all the features required. A final sensor configuration and software setup should be decided upon as well.

6. *Construct an initial prototype to facilitate sensor data collection.*

This prototype should be able to be run independently of a computer however may need to be connected to a laptop for data analysis and troubleshooting.

7. *Inspect data and develop program to analyse it for bike presence and direction.*

Using the chosen development environment, the sensor data should be analysed and converted into a presence and a direction digital signal with minimal false positives or missed bicycles.

8. *Evaluate sensor effectiveness at determining the presence and direction of passing bikes.*

Once the prototype is effectively collecting data and has been able to detect the presence and direction of passing bikes, the effectiveness of this system should be considered as ere may be many false positives or false negatives depending on the distance away from, speed and shape of the bike and rider. It should also be tested to see it waving branches or pedestrians will trigger it.

If time and resource permit:

9. *Deploy a suitable prototype into the field and record the data output from riding bikes past it as differing distances and situations.*

Ideally once the prototype has effectively been tested in controlled conditions it can be implemented using a battery pack or other power source on an actual trail. This can be used to test its effectiveness in differing weather conditions, at night, and also to test its ability to withstand the elements.

10. *Develop a safety system based on this sensing system, to let bikers know of upcoming traffic on the trail.*

Once a system has been effectively tested, a networking technology and topology should be considered in order to make this an effective indication system, and the design and implementation of the ideal final design should be considered.

Possible Limitations

The Limitations of this thesis are that it is quite unknown how difficult it will be to implement an effective and reliable bike monitoring system, and so significant time has been allocated to this.

The budget and time available to be dedicated to this project are also limited so the sensors used should be relatively cheap and easy to procure. The limited budget also dictates that the software and development environment used should be generally free or cheap, as it is not feasible to purchase an expensive debugging kit or software just for one prototype.

Literature review

This literature review is the compilation of the research into the previous sensing and computational system used to complete similar bike sensing systems. Below, a section has been dedicated to each the sensor technologies, communication technologies and computational technologies.

Previous sensor technologies

There have been many studies done on the population and frequency of travel of bicycle riders on various cycling lanes and around various cities. They have been thoroughly tested however usually have an error rating of at least 1% and do not sense for the direction of the rider. Below I will discuss the uses of these technologies and how they have impacted the choice of sensors for this installation.

I initially found ten sensor technologies which had been used in varying ways to detect the passing of bicycles. These were put into a weighted matrix to decide the most suitable ones for this project.

Video Processing

The use of video analysis is discussed by (Ryus et al., 2014), and has been successfully implemented in the past for use in cities and highways. Some of the benefits of this system is that it is quite a reliable method and works well in many different conditions. It is very fast at detecting as it is analysing many images per second. It is quite reliable at detecting bikes as well, as each picture is analysed separately. Disadvantages of this system include that it requires a lot more computational power than the other systems to run and is quite complex to create. Another example of Video detection of bicycles was demonstrated by (Cho et al., 2010). In this paper the researchers discussed the use of a tracking algorithm to track to bikes after they had been detected. This was proven to work quite well in the context of their research, which was tracking and counting bikes with video from dashcams or from traffic monitoring systems.

Figure 1 - Example of a Video Detection System (CT Technology 2023)

This research was completed to help in the research of self-driving cars, so focuses on the tracking of the bikes quite heavily. The computational power this algorithm requires mean it is less suitable for the use case proposed in this thesis. The tracking algorithm used in this paper would be considered as it can be generally applied to 2D inputs if they correlate. As shown below, as I have defined my system to be a portable low power unit, a continual monitoring of a live video feed is not feasible. An adaptation of this though, where a camera takes a photograph every half second or when triggered by another sensor may be feasible. This would have to depend on the computational power available.

Image

Image detection is quite similar to video detection however, it is slower and requires less computational power.

It does not have the tracking capabilities of video detection and is not as fast at detection bikes, however focuses more on the geometric relationship of the two wheels and the bike design. As shown by (Lin & Young), using the geometric relationship of the bicycles and its tyres is a suitable method for detection most types of bikes, and it is able to guess the bike directions with high accuracy. The upsides of this robust image detection system are that it is able to detect bicycles even if they are partially covered by the rider or are not directly side on to the camera. These researchers in particular tested the image recognition to at least 45 Degrees of angle on either side of directly side on in the X, Y and Z dimensions. It was able to detect the bicycles in each of these positions.

Figure 2 - Raspberry Pi Camera (Raspberry Pi Foundation 2023b)

The downsides of this method of detection, as it was based on the detection of the wheels and the relationship between them, and the rest of the frame were that the system struggled when presented with pictures which did not include the entire. It also struggled if the design of the bike did not conform to the conventional top tube design, such as folding bikes, or electric bikes. which is also a symptom of the design of the image detection system. This was the detection algorithm created by the above researchers, however there is also a few other image detection systems which are able to detect the presence of bicycles. These include the You Only Look Once (YOLO), Histogram of Oriented Gradients (HOG) and Discriminatively Trained Part-Based Model (DPM) methods. These three other methods have been tested extensively by (Redmon et al., 2016) (Laptev, 2009)&(Felzenszwalb et al., 2010) to varying success in detecting the presence of bicycles, however, are generally still very computationally expensive. The robustness and price of this kind of system should also be considered as the goal is to have a standalone system, which inherently means it is likely to be vandalised given that it is not designed to be placed in a public place. All this has been considered below in the weighted matrix.

Active Infrared

Infrared systems are similar to the image recognition technology discussed above however have a few unique benefits and drawbacks.

Firstly, it is important to make the distinction between active and passive infrared systems. Active systems use an infrared emitting device to detect the distance to given objects, and then use this distance information to determine the object. This is used in systems for traffic counting often, however is ideal if it is placed in a position above the intended counting zone as shown by (Noyce et al., 2006). This is because the beams have a wide Field of view in one direction only and therefore are always looking at a cross sectional area of the bike or traffic item in question. The main benefits of an active system are that they are able to monitor a wide path if positioned correctly, and most active infrared systems incorporate multiple beams, thus being able to detect the direction of the bicycle easily with a single sensor node. They are also less computationally expensive than pure image recognition as it is able to ignore visual background movement such as tree branches moving and is a lot less likely to be falsely triggered. We shall discuss passive infrared below.

LiDAR

A lot of active detection methods would not be feasible due to their power and computational requirements if not paired with another sensor to activate or deactivate them. Out of the three of these technologies, active Infrared or an infrared camera or a LiDAR sensor would be most appealing as they are not affected by the time of day and are less affected by the current weather conditions as well. These have been proven to be quite effective Systems and are in use today (Bernas et al., 2018).

The issue of bike detection systems being designed for mass traffic counting and not for minimal error rates is an issue that I believe will prevent the best system from being implemented by simply copying the sensor configuration of other researchers. The researched systems are almost always on a footpath of bicycle lane as well.

Some studies focusing more on the mass counting of bicycles and pedestrians, such as researched by (Turner et al., 2007) , use a multi sensor approach, one for detecting a passing object, and one to determine the identity of the object. However, as the use of either in-pavement or temporary on-pavement sensors is used in these systems, they cannot be implemented in my system for two main reasons.

Pressure pads Pressure Tubes

Pressure pads or pressure tubes will not be considered further in this literature review as firstly, given the rough terrain of installation and the likelihood of a gravel path, the sensors will not last as well as other sensing systems. Secondly, given that the system will ideally be put up on an actual track, minimal modifications to the track are preferred and are necessary. The last reason these were not considered was because the system was supposed to cheap, and these are commercial systems which cost thousands of dollars.

Figure 3 - Pressure Tubes for counting bicycles (Eco-Counter, 2022)

Peizo Electric Strips

Peizo Electric strips are another technology which has been considered, however the price of the systems, along with the nature of its installation and the inability to gain any directional information prevent it from being considered further.

Beam Break

Beam break sensors are used to count the passing of bikes by (Baas et al., 2016). They operate quite simply and are generally similar to pressure tubes or in that they can be placed a certain distance apart to determine speed and direction of the bikes.

The downside of the pressure tubes is that they suffer some of the same issues as the pressure pads sensors in that they are physically on the trail and so easily damaged and they are prone to not sensing bikes as they pass over as mountain bikes have a lot wider tires than typical bikes and are often not traversing on flat ground so may not get a consistent weight applied on them.

Beak Break sensors do not suffer from the same troubles and are different from active Infrared, LiDAR and Ultrasonic measuring devices because they do not return a distance to the item which passes them rather returns a digital output if there is an item in front of them or not. This is useful as it requires no signal processing however is pretty useless as it also provides very little information for disallowing false positives. This makes it a great secondary system however it may not be suitable for a primary system as any object in front of the beam break sensor would trigger it.

The beam break sensors also have their downsides as they can be very easily set off by local wildlife and possible pedestrians and require a post to be set up on each side of the trail so that the reflector may be hung. Beam break sensors may be paired with other sensor types such as infrared cameras or a LiDAR system to help determine the bicycles direction and discard any false positives which did not detect a rider travelling past them.

This technology is often used in conjunction with pressure tubes or piezo tubes to create a system which counts both pedestrians and bicycles as show below.

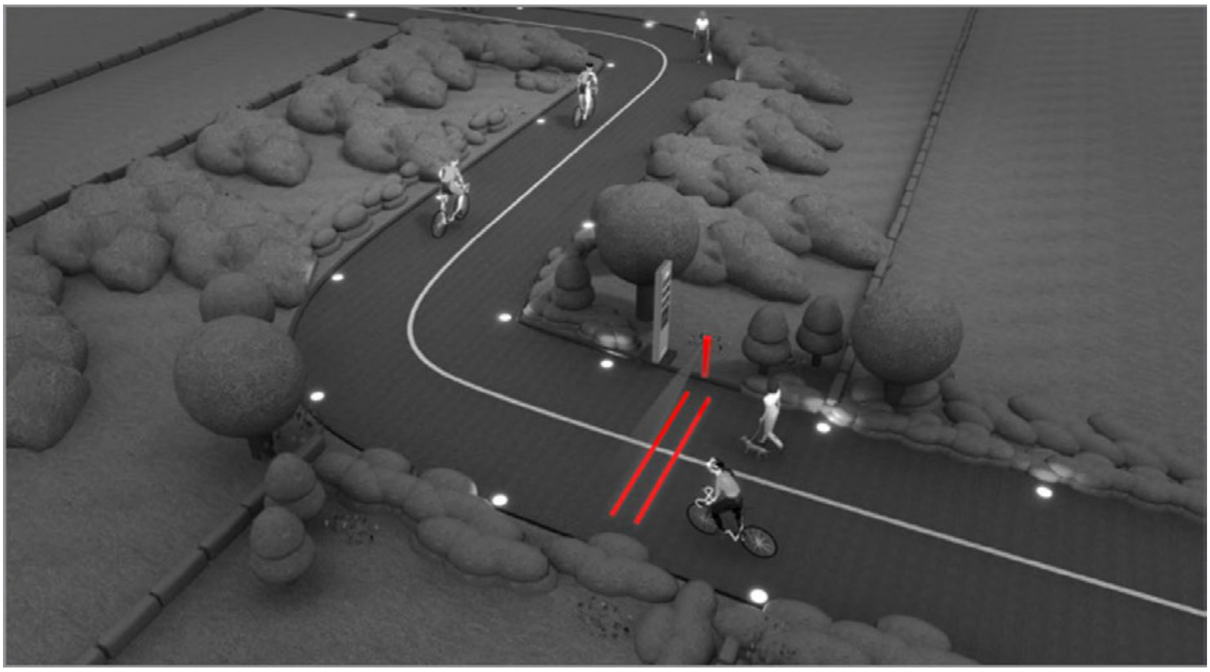


Figure 4 - Dual Mode system for sensing bicycles and pedestrians(q-free 2023)

Passive Infrared

Next we shall passive infrared, which although similar to active infrared, does not use an active light or laser source. This means that instead of an image which resembles a black and white image and is not affected by the night-time, the image is basically just reading the heat signatures of the images items and so doesn't require as much power to run. This methods of observation has already been effective in identifying the types and presence of pedestrians walking along a bush trail as shown by (Abildso et al., 2021). This technology may be suitable for the detection of presence and direction of mountain bikers on a trail also. The use of a camera would however raise concerns of the computational power required as a microcontroller would not be capable of processing the data quick enough to be useful.

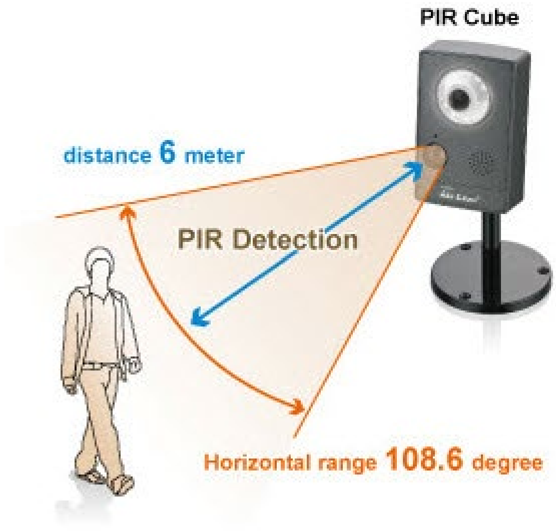


Figure 5 - Example PiR sensor (seeedstudio 2019)

RFID

Another technology which has been implemented in the past and could be considered is RFID tags. These are very simple to install and sense, and with a combination of two sensor the directional information may be determined by using the strength of the RFID signature. This has already been implemented in the past for the timing of athletes, and can be used to build a low cost timing solution (Tsai, 2011). The obvious problem with this however is that all users of the trail must wear an RFID tag and if they did not the system would be useless. As such this technology was not researched further.

Apart from the technologies used in most previous systems, three novel sensor technologies will be experimented with for this system. These are passive infrared, vibration monitoring and Time of Flight (distance) sensors.

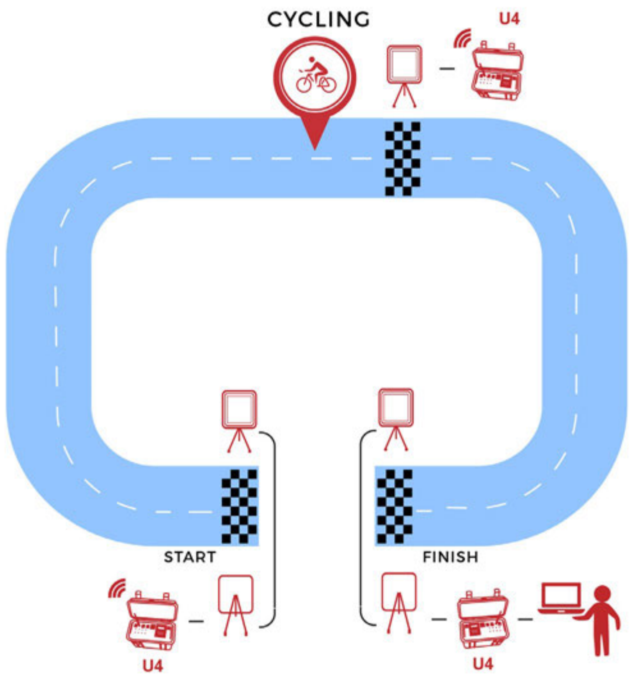


Figure 6 - RFID system used in bicycle race (rfidtiming 2017)

Audio Processing

Audio Processing may not seem like an obvious choice to reliably detect the presence and direction of bicycles, however it has been implemented with great success by (Zu et al., 2017) and (Djukanović et al., 2020). These systems where implemented on car traffic situations granted, however in instances where the background noise of the mountain biking trail can be assumed to be minimal, a system such as this can be implemented as directional microphones such as the ones used by (Zu et al., 2017), can be easily obtained. The main trouble with this sensor is the software design as it must be tuned to the sounds around it so to not have any false positives, and it may not work during wet weather if the microphones are not sufficiently covered.

Vibration

Monitoring of the ground vibrations can also be an effective way of counting bikes. In their study, (Rivas et al., 2016) concluded that bicycle could be detected using this method, even when using the same roadway as cars. The system I propose for monitoring vibration is a stake driven into the ground, with a sensitive accelerometer attached securely to it. Audio waves is also a possible source of bicycle detection, however as this system is going to be implemented in a low power situation and audio inputs need to be analysed, which is to computationally intensive to implement in this system. If a design for an audio could be implemented which did not require much computation it may be investigated later in the project.

As the direction of the vehicle is mandatory for this system, and as the systems need to be as close to 100% accurate as possible, sensors of each kind shall be used for testing, as well as perhaps multiple sensors used in unison.

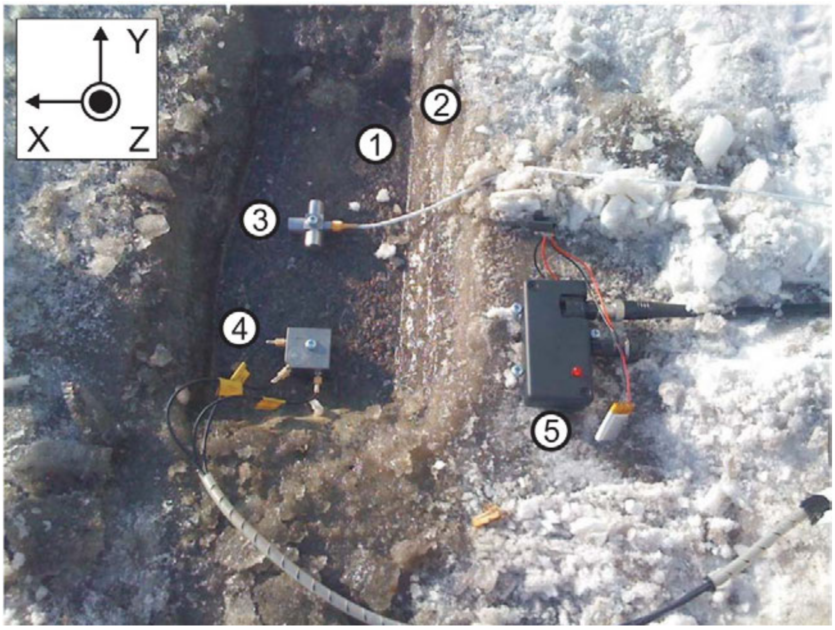


Figure 7 - Accelerometers used to measure roads vibration(Hostettler, 2009)

Products already on the market

Research was also done to see if there were any products already available which could complete the sensing function needed for this project. As it was decided that the sensors would all need to be wireless, that ruled out any pneumatic tube, piezo-electric or and they needed to determine direction reliably, no product could be found to suit other than a PYRO box by (Eco-Counter, 2022), which is designed for counting pedestrians and is a sealed unit, so is hard to integrate into the system and adjust it sensitivity.

Computational technologies

In the implementation of this system there must be some computational power to facilitate the sensor data processing if necessary, and eventually facilitate the communication between system nodes once the system is fully implemented. Important factors to consider when choosing a device is computing power, size, power consumption and cost, these shall be used in a weighted matrix below to decide on the final technology to be used. There have been multiple options for computational devices, and the only requirements which shall be considered is that they have enough analog and digital inputs and outputs that are easily usable, and that they are able to be deployed in a relatively low power use case.

Arduino Nano

The first few options investigated are Arduino microcontrollers. These can come in various sizes with differing capabilities. The first of the three styles investigated is the Arduino Nano, which is based upon the ATmega328 processor (Nano, 2018). This is a very capable platform which is the cheapest of the three investigated. This Arduino has 22 digital I/O pins, 7 of which are capable of PWM for analogue output or for running communication technologies. There are 6 of these pins which can be configured to run in analogue input mode, along with 2 other pins which can only be used in analog mode for a total of 8 pins of analogue pins with a resolution of 10 bits.

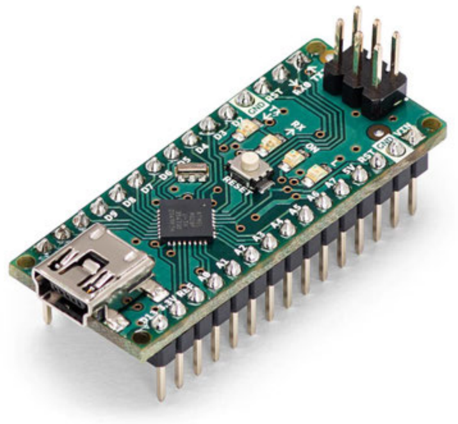


Figure 8 - Arduino Nano (S. A. Arduino, 2023)

These boards, along with the rest of the Arduino boards are very easy to develop code for and debug as there is a massive hobbyist community surrounding them. another advantage of this board is that they can be purchased as a combined package with different communication modules such as Wi-Fi, Bluetooth, Lora WAN cellular built in and so requires minimal setup and are ideal for integrating these communication technologies.

Some of the disadvantages of this system are that they are it has very little ability to either sink or source large currents, with it only being rated for 20mA(Babiuch et al., 2019). It also has very little onboard memory at 32KB, which makes logging and storing the sensor data trickier as it would need to utilise external storage.

Arduino Uno

The Arduino Uno is the midsize variant of the Arduino lines up so has a few more capabilities than the Nano. It is based on a similar microprocessor to the Nano, an ATmega328P, which consumes less power and is more efficient variant of the ATmega328 (Arduino). It Offers the same amount of Analog pins as the Nano, however these are now dedicated pins, and there are 14 dedicated digital I/O pins.

Figure 9 - Arduino Uno (Arduino, 2023b)

It has the same ability to drive and sink current of its smaller variant however is more robust in its allowed input voltages, which can now go up to 20V. The memory situation is also the same.

The last great advantage of the uno is its expandability, as the design allows ‘shields’ to be added onto the device with functionality such as communication systems, current switching, analog to digital converters and more.

In summary this is the same microprocessor as the Nano, simply presented in a more robust and configurable package.

Arduino Mega

The Arduino Mega is the biggest and most capable of the Arduino family which will be investigated. It is based upon an ATmega2560 microprocessor. This supports many more input and output pins, with 54 digital I/O pins, and 16 Analog input pins (Kumar et al., 2015). The design of this board also supports the use of shields the same as the Arduino Uno, and ones which are specifically designed for the Mega.

Figure 10 - Arduino Mega (Arduino, 2023a)

The advantages of the different microprocessor are that it has 256KB of memory, which allows it to store larger programs, and capture images, or analyse more data samples than the Uno. The downsides are that this is quite an expensive product, and although it has a great deal of I/O, that is not as important as the data analysing capabilities of the processor.

Raspberry Pi

The next option is the Raspberry Pi family of single board computers. These are very capable machines which would be able to process images and other signals easily, however, may not be up to processing video live. The Raspberry Pi 2 model B, which is the Pi investigated in this research and is based upon an integrated System on a Chip (SoC), namely the Broadcom BCM2836 (Raspberry Pi Foundation 2023a). This SoC is equipped on this pi with 1GB of RAM, native SD card support, and support for inputs from USB devices, 3.5mm Headphone devices, and General purpose I/O pins.

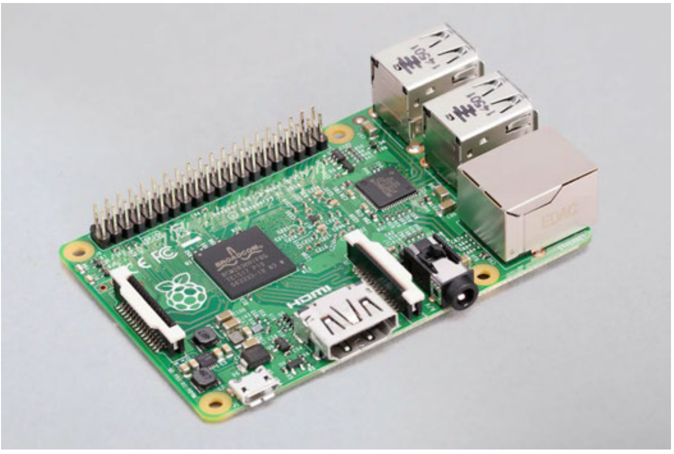


Figure 11 - Raspberry Pi 2 Model B (Raspberry Pi Foundation 2023a)

Similar to the Arduino Uno, there are many 'shield' available for the Pi which expand its functionality and allow it to communicate using many technologies. Another advantage of this system is that it runs its own Operating System based on Linux so the testing and development of the system would be much simpler as programs such as Octave could be used.

The Pi is also quite cheap for the amount of computing power available, which is also beneficial. The main downside of the Pi is its power consumption; however, this is not very large when it is not processing and is suited to a wireless sensor node application as shown by (Vujović & Maksimović, 2014).

Jetson Nano

The Jetson nano is the most computationally capable of the computational technologies investigated. It has a dedicated 128-core Maxwell GPU, and a quad-core ARM A57. It is able to encode and analyse video streams live and so would be necessary if using the video analysis method of bicycle detection. This Single Board Computer is designed for machine vision applications.

Figure 12 - Jetson Nano (NVIDIA, 2023)

These boards are very useful for robotics applications and computing large neural networks, however they are quite power hungry. They are the only boards investigated which need heatsinks to prevent overheating. This points to the second issue with this board, the power consumption. It used a minimum of 5W when processing video, which is significantly more than any of the other technologies. These boards are also quite expensive and given the data analysis needed for this system are very overpowered (Süzen et al., 2020).

PIC18

There have been PIC18 8-bit based wireless sensors constructed, however this integrated chip technology does not support much data processing much and has no advantages over Arduino in this situation. It also requires an expensive development board (Yussoff et al., 2010). It would be useful if the sensor node was going into mass production and should be investigated more if the Arduino is chosen for the computational technology.

Figure 13 - PIC18 development board with debugger (Microchip, 2023)

PLC

Programmable Logic Controllers or PLCs are industrial applications of PIC processors and are very reliable and easy to code. They come with the ability to add expansion cards, some of which enhance its ability to process data. The downside of this system is the cost, power consumption and the lack of flexibility with the programming environment. These have been used in some sensor applications however really shine in automation work (Stankovski et al., 2022).

Figure 14 - Click PLC (Automation Direct 2023)

All the above computational technologies shall be ranked based on six criteria in a weighted matrix. These are the Power Use, I/O Capability, Expandability, Robustness and Price. The results of these are shown below in Table 2.

Communication technologies

In order to implement this system fully, There must be communication between the nodes at the end of each section, and perhaps a connection to the internet to update the amount of bikes using the trail however this is outside of the scope and will not be considered further.

In terms of communication technologies there is two differing classes, one where the nodes directly communicate with other nodes in a peer-to-peer network and one where they all connect to an online server through various ways, in a master-slave relationship.

Below I will discuss the benefits and challenges facing each connection type from my research and shall also make a recommendation for the type of communication I shall use.

Peer to Peer technologies

A peer-to-peer communication is inherently more reliable and secure as it is not relying on other companies' infrastructure to remain operational. The concern with this communication type however is the distance able to be covered between the two nodes, and power concerns as long-range two-way communication can sometimes consume a fair bit of power. There are a few traditional technologies which have recently made adaptations to be considered for long range applications. Since the introduction of IoT there has been a lot more research conducted into the wireless technologies available for long range and low power use cases. Even if an 'IoT' system architecture isn't adopted these technologies shall be useful as they fit the similar requirements to this system (Foubert & Mitton, 2020).

LR-WLAN

The first of these is long range Wi-Fi connections. These are quite common in networking situation however require much power al their specifications are designed for a high rate of data transfer. As shown by a research group, this can be effective for up to 800m however requires significant power to transmit those distances (Qiao et al., 2018). These researchers also concluded that it can be configured to be used as an effective IoT communication channel.

LR-BTe

The recent Bluetooth 5 standard also has the ability for a low data rate, long range mode. This has been considered by researchers as a good technology for IoT however isn't often adapted as it is still relatively short range (Collotta et al., 2018).

Recently there has been much progress made to the low power and long-range communication standards as and these new technologies shall be discussed below.

Lora WAN

Long range wide area network or Lora WAN for short is a relatively recent technology which has become very popular in the IoT space as it requires no licencing fees, is very low power and can communicate at ranges of up to 40KM in rural areas. Given the type of terrain around mountain biking trails it is unlikely that 40KM will ever be reached, however as it can be used up to 10 KM away in urban environments and has been tested in bushlands range is not an issue(Grunwald et al., 2019). The downside of this technology is that it is very low bitrate and may be slightly tricky to setup (Silva et al., 2017). This technology also requires specialised equipment as well however companies are already manufacturing these relatively cheaply (Wisen, 2022).

NB-IoT

Narrow Band Internet of things communication is similar to that of Lora WAN however is licensed, and as such provides better reliability and range, but at the cost of battery life and cost (Sinha et al., 2017). As it is like Lora WAN but is more expensive, in the confines of this thesis I shall not consider it further.

Zigbee

Zigbee is an established IoT technology however it focuses more on the bandwidth and less on distance as it is designed for in home applications, ,the benefit for this technology is that security can be improved immensely however as this system is not dealing with sensitive data this is not considered to be of much importance (Fan et al., 2017).

UHF

Ultra-High Frequency or UHF communications are most commonly used for the transmission of voice signals however it has been used for IoT applications in the past (Bedeer et al., 2017). The main downside of UHF is that it inherently loses signal fast and so any long-range communication using it would need to be quite powerful.

Master Slave technologies

Another network topology to consider is Master Slave connections. The obviously need a base station, and so are suited for use in more developed area where you can piggyback off of another companies' infrastructure. A few inherent downsides to this are the unreliability of the network, the security vulnerabilities that come from being connected to the internet and the subscription fees that you will pay for these services.

Sigfox

Sigfox is a company which specialises in IoT infrastructure and distributes a technology very similar to Lora WAN. Sigfox connection devices connect to the Sigfox base station if nearby(Lavric et al., 2019). Base stations have a range of around 40KM, and so if this system where setup in a town this may be a viable option. However, as it is going to be setup rurally and there is no guarantee of a Sigfox base station nearby this cannot be used.

Satellite / Cellular

Existing technologies such as satellite and cellular communications have been considered, however as they have ongoing costs, require significant power, and do not provide any significant benefits which are useful for this system.

After reviewing all available communication technology options, it seems obvious that a Peer-to-peer connection type would be best, however I shall leave it up to the experimentation phase to make the final decision of a suitable technology.

Consequences/ ethics

This thesis is looking at constructing a system which is ideally going to increase the rider awareness of upcoming dangers and reduce crash rates when in use. This is of great benefit when the system is functioning properly however sometimes the system may fail to recognise a rider or may give a false positive when a rider is not there. This is a concern as the system if marketed as a safety system must be useful as a safety system or else it is a useless system.

In accordance with the engineers Australia code of ethics point 3.3, unless a 100% bike sensing capability is ensured, the stake holders (in this case the users of the system) must be informed that it is not a safety system and should not be relied upon for 100% effectiveness (Australia, 2019). This may be done through the labelling of the system as an 'indication system' not as a 'safety system' and using signs and such.

Another consideration is that the system ideally will be installed on a mountain biking track, and although it will need little maintenance, it should still be checked periodically, and a plan should be made to decommission the system after it has served its useful life. This is in agreeance with point 4.3 of the code of ethics, and simply consists of a plan to return the area where the system was installed to its original condition without leaving debris behind.

During the construction and implementation of this thesis, many risks shall be taken, although not particularly large. The risk management plans for these activities are attached below in Appendix B, although may be modified when the actual implementation and experimentation is more defined in the future.

The ethical responsibility of creating a safety system is quite large and should not be underestimated since it would need to be 100% effective, as such the system should not be described as a safety system however should be an indication system. Once the system is implemented, and end of life plan should also be created to ensure that the system does not have a permanent impact on the environment it is installed in.

Methodology

Overview

The objective of this thesis is to develop a mountain bike sensing system which is suitable for deployment onto a trail, and which can be confidently relied upon to reliably function.

There are a few steps which are necessary to complete this design. Firstly, the sensor technology must be chosen and acquired. Next, the computing hardware to support this must be acquired, and a system for processing the data must be decided upon.

Once these have been acquired the effectiveness of the sensors and their configuration should be understood. This shall be done through the testing of a chosen group of sensor technologies.

Deciding Sensing Technology

Below in Table 1 and 2 I shall use weighted decision matrices to decide the sensor and communication technologies I shall test during my project.

Firstly, the sensor technologies. The most important factors to consider when choosing these have been added to a weighted decision matrix below in Table 2, which has been weighted accordingly.

The above technologies have all been collated and ranked 1-5 on the following 8 criteria. These criterion were weighted 1-3 based on how important they were to the goal of the project, which is to make a power efficiency sensing system.

These items were evaluated in the weighted matrix shown in the Table 1 below.

Power efficiency

Power efficiency is quite important to this system as the final goal for implementation is a standalone system. This disadvantages a few technologies which are designed for urban deployments, as the mountain bike trails which this system will be deployed on will not have any access to external power. Thus, this criterion has been weighted the highest weight of three.

Reliability

The reliability of the proposed system is quite paramount as it is ultimately a safety system. The reliability of the system in this context is defined as the likelihood that it will not trigger the sensor when a bike rides past it or that it will falsely trigger when a bike is not passing it. It is obviously more beneficial to have a system which triggers more false positives than one which triggers more false positives within reason. As such, this criterion has also been weighted a three.

Directional

The system will ultimately need to be directional to be the most useful as the hazards the safety system Is warning of (oncoming riders) are not stationary, and thus only the bikers coming up to an oncoming bike should be warned. Most sensors are directional if you have enough of them placed far enough apart, but this criterion weighs how many sensors are needed to implement a directional sensor network. Thus, this criterion has also been given the weighting of three.

Timely

This may seem an odd criterion however, with the limited computational power available, some sensor technologies will take a while to compute if a bicycle is detected or not. The sensors can also be untimely by not being able to tell the when the bicycle is nearby. An example of this may be the RFID system of not configured correctly could trigger the system when the bike is very close to the antenna or when the bike is further away. This can be solved by simply moving the location of the sensors to give more time to warn the riders ahead, however it is more ideal to simply have sensors which sense the presence of bicycles in a timely manner. As such, this criterion is only weighted a two.

Processing Power

The processing power required to detect the bikes is quite an important consideration when considering the sensor configuration. The system will be standalone as mentioned above and will have to have its own power supply, in the shape of a solar panel probably. The greater the processing power required to detect the bike, the greater the power usage and the larger the power supply must be. It also required more expensive computational components such as a jetson nano or a raspberry Pi instead of an Arduino or similar microcontroller. This brings the computational power to a weighting of two.

Simplicity

The simplicity of the system, while not directly a safety issue, will greatly improve the ability for the system to be scrutinised for any potential bugs. It also makes the sensor systems easier to implement in the limited time available. As this is ultimately a safety system, the ability to scrutinise the code and to clearly see how it functions is valuable in ensuring its reliability. This is therefore weighted as a three.

Robustness

The robustness of a sensor system is not something which is discussed in many research papers, however, is important in the choosing of this sensor technology. This is because sensors which need physical contact; i.e. pressure pads, tubes and strips, will need more maintenance and are more likely to become damaged and cease working. This is not an issue in the urban environments which they are generally installed in however the gravel environment, especially considering that the bikers may be sliding when they hit the sensor. Another consideration for robustness is the effect of weather on the operation of the sensor. How moisture in the air and on the ground affects sensor performance shall not be discussed in this project other than in the hypothetical as it would require extensive testing. Overall, as this criterion affects the long-term usability of the sensor system and can be mitigated by protection methods it is weighted at a two.

Low Price

The price of the system is considered when choosing a sensor technology as this project is looking at a price efficient, reliable system. That is not to exclude expensive technologies outright, but just to try and use the resources give to the best of their abilities. This is therefore considered to be one of the least important criterions and is weighted at a one.

From this table the top six technologies where Pressure pads, Pressure tubes, Beam break sensors, Vibration sensors, audio sensors and photograph sensors.

As pressure pads, beam break sensors and pressure tubes have already been well documented in their use for counting bikes, and are all quite expensive, this project shall focus on the next three best options, which are vibration, audio, and photo sensing technologies, how these shall be approached is shown in the below sections.

Table 1 – Weighted Decision matrix for sensor technologies.

Technology	Power efficiency	Reliability	Directional	Timely	Processing Power	Simplicity	Robustness	Low Price	TOTAL
Weighting	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**1**	
Audio	4	2	1	2	3	3	2	4	**48**
Video Processing	1	3	5	2	1	1	1	2	**40**
Photo/Infrared	3	2	2	1	2	3	1	3	**41**
LiDAR	2	3	2	2	3	1	1	1	**37**
Pressure Pads	3	3	1	3	4	5	3	1	**57**
Pressure tubes	2	3	1	3	5	3	2	1	**48**
Beam Break	2	2	1	4	5	4	2	4	**53**
RFID	1	4	1	1	3	2	1	1	**35**
Passive Infrared	2	1	2	1	2	2	3	2	**53**
Vibration	4	4	2	2	3	2	4	4	**58**

Audio Sensing Technology

To Sense the sound, firstly a USB microphone has been used to start the audio analysis. It was connected to a stake in the ground and a bike was driven past, while recording. As there was little to no ambient noise there was a spike in the volume from the microphone, however simple amplitude analysis would not be sufficient as then any type of noise would trigger the sensor.

It was decided that a Fourier transform would be applied to the wave to determine the specific frequencies that would be present when the bike went past. This could also determine the difference between a branch cracking or an animal walking past or a cyclist.

Vibration Sensing Technology

Vibration sensing was tested using accelerometers attached to stakes in the ground to sense to sense the vibration amplitude. This is the simplest way to do this however more complex methods are being investigated.

Infrared sensing technology

The infrared Raspberry Pi camera was tested by driving a bike in front of it. The data was analysed by using a subtractive method to determine movement in the frame, this was not very effective as it sensed all of the movement from leaves and such in the background.

Deciding Computational Technology

In the literature review above seven computational technologies were investigated. These shall be ranked based on the below criteria.

Power Use

The power use of the computational technology is generally proportional to the amount of data processing needed. However, with the PLC, where power consumption is not a major consideration concern in the design, it has a much higher power consumption than would be on another chip with the same computational power. As such this criterion has been weighted as a two.

I/O Capability

The input and output capability of the computational technology chosen is an important factor to consider, as it allows for more sensors to be connected. However, as long as it is able to be connected to multiple sensors at once and have enough output capability to connect via the chosen communication type there is little need for may more such at the Arduino Mega boasts. As such, this criterion has been weighted as a one.

Expandability

The ease of expandability of the system refers more to the community surrounding the product and ease at which additional functionalities can be implemented. This includes the communication technology, or additional analog-digital converters, or outputs capable of driving larger loads. This is quite critical in the implementation of this project and so has been weighted as a three.

Work environment

The work environment is also a factor to consider when picking the computational technology. It is similar to the expandability of the system however considered the development environments capabilities, ease of use, availability and price. For example, the PIC18 requires an expensive debugger board to implement its code. As such this criterion has been weighted as a two.

Robustness

The robustness of all of these systems is both their ability to withstand bumps and knocks, but also their ability to exist in high or low temperatures. The only item which struggles in this case is the Jetson nano, which would need airflow to keep the GPU cool. This is not a critical criterion and as such had been weighted a one.

Price

The price of the computational technology is self-explained, and significant criterion so is weighted as a two. This is due to the limited recourses available during completing this research.

Table 2 – Deciding Computational Technology

Technology	Power Use	Input Capability	Output Capability	Expandability	Work Environment	Robustness	Price	TOTAL
Weighting	**2**	**1**	**1**	**3**	**2**	**1**	**2**	
Arduino Nano	5	3	2	2	5	2	5	**43**
Arduino Uno	5	2	2	4	4	3	4	**45**
Arduino Mega	4	5	4	3	4	5	2	**43**
Raspberry Pi	3	4	3	5	4	4	3	**46**
Jetson Nano	1	3	2	2	3	2	3	**27**
PIC18	4	3	3	3	2	3	4	**38**
PLC	2	2	5	4	3	5	2	**38**

Sourcing Parts

After the parts had been chosen, they had to be specifically found and sourced. Firstly the parts for the audio sensing technology.

Small electret microphones were an obvious choice for this application, however alone they are difficult to work with as they have a very small output voltage. Thus it was decided to purchase two connected to a small pre-amplifier, specifically the using the Maxim MAX4466 instrumentation amplifier as pictured below.

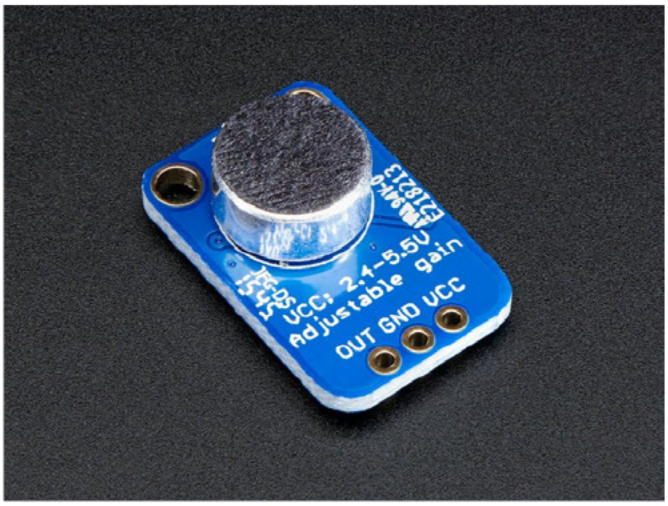


Figure 15 - Electret Microphone with Pre-Amplifier(Adafruit 2023b)

As this still outputs an analog voltage, an analog to digital converter is needed to input into the Raspberry Pi. As it is an audio signal a sampling rate of at least 48kHz is preferred, and so the

PCM1808 ADC from Texas instruments was purchased on a breakout board as shown below. This is an overly capable chip for this job as it is able to sample at 96kHz and is also 24-bit precision.

As these parts were shipped in, while they were on their way testing was begun using a USB connected webcam microphone, which helped the development of the sensing system.

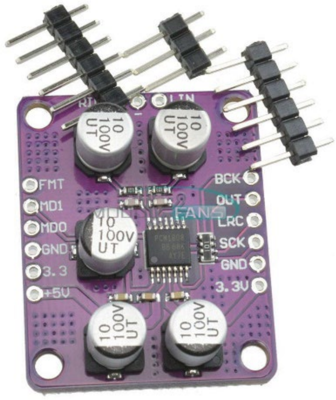


Figure 16 - PCM1808 ADC on breakout board (Texas Instruments 2023)

Next, the parts for the video sensing technology. The only part required was a Pi Camera as shown above in Figure 2. This was sourced from the university and so was able to be used right away.

The components for the vibration sensing were then sourced. Initially a digital accelerometer was chosen to monitor the vibrations, and the ADXL345 digital accelerometer, pictured below, was chosen as it would be able to connect directly with the Raspberry Pi.

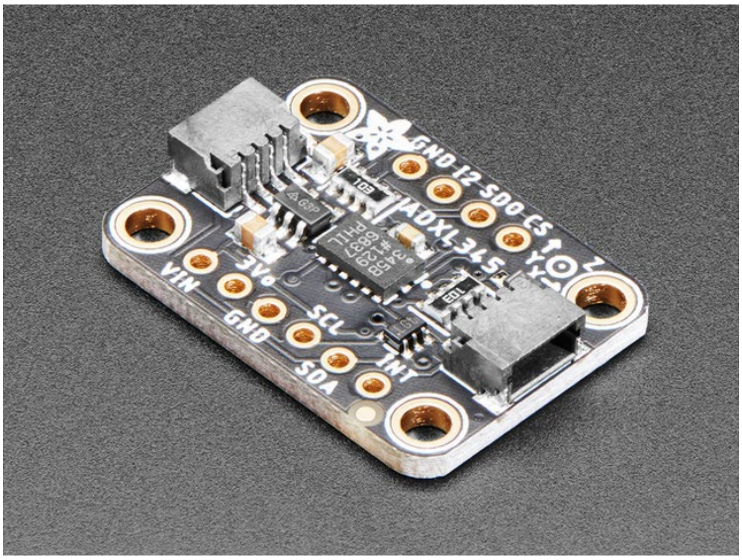


Figure 17 - ADXL345 Digital accelerometer (Adafruit 2023)

After some testing with this however it was not able to pick up the vibrations from far enough away, and I was decided to purchase a Geophone. This is also shown below, however as it output an analog signal another analog to digital converter must be used. This was chosen as the ADS1115 as this has an onboard amplifier and so the signal from the geophone would not have to be pre-amplified.

Figure 18 - SM-24 Geophone (SparkFun 2023)

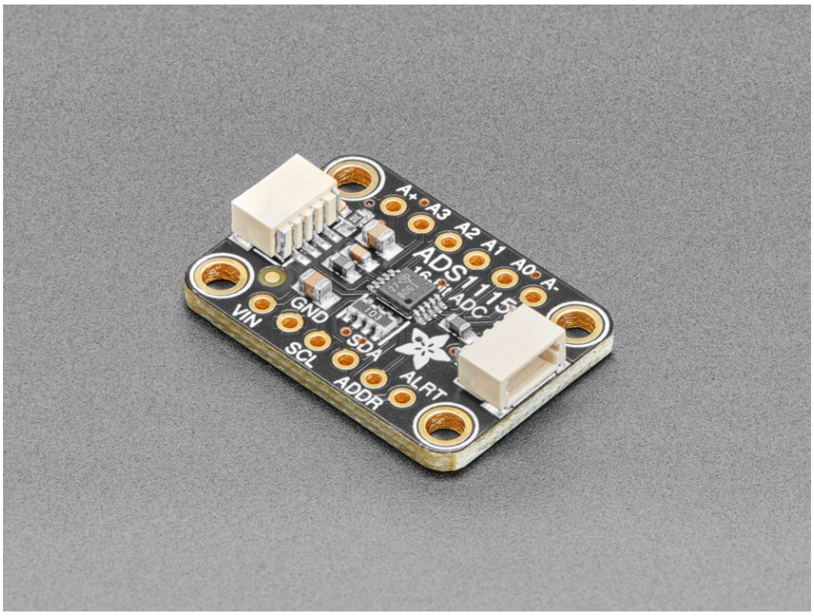


Figure 19 - ADC with input gain (Adafruit 2023a)

These components took a while to arrive and therefore were not able to be properly utilised, however from initial measurements they are a lot more sensitive than the previous setup with the digital accelerometer.

Once all the parts were sourced the experimentation could begin in full as detailed below.

Audio experimentation procedure

Audio Testing Process

Of the above methods, the Audio Sensing method was chosen to be tested first.

Below I shall overview the testing approach, methodology and techniques used.

Testing Theory

Below in Figure 21 - *Block Diagram of the Audio Testing Setup* is shown a block diagram with the proposed layout of the microphone sensors for the testing of the audio sensing technology. I shall be using two microphones to attempt to sense directionality as well as presence of a bicycle.

The two electret microphones were purchased on boards with a preamplifier, specifically based around the MAX4466 Op-AMP(Adafruit 2023b). This particular product has adjustable gain and shall be gone into with more depth below when the design of the system is justified.

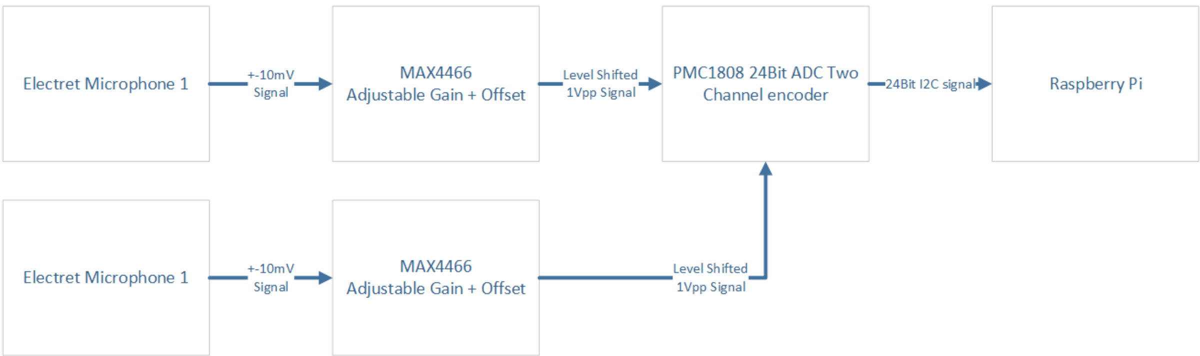


Figure 20 - Audio Signal Overview Before Entering Pi

These produce a 0-5V analog signal which is unreadable by the Raspberry Pi as it does not have analog inputs, rather it is fed through a PCM1808 analog to digital converter, which is able to sample this signal at up to 96kHz. This is a 24-bit decoder, and is able to communicate to the Raspberry Pi on the I2S protocol (Texas Instruments 2023). The two microphones were fed into the left and right channels of this ADC, which was able to the be input onto one of the GPIO pins of the Raspberry Pi.

Initially, a USB Logitech C270 microphone was used to record the sound, as the microphone modules had not arrived, however when they arrived the testing was redone as the microphones could be setup to have a much higher sampling rate.

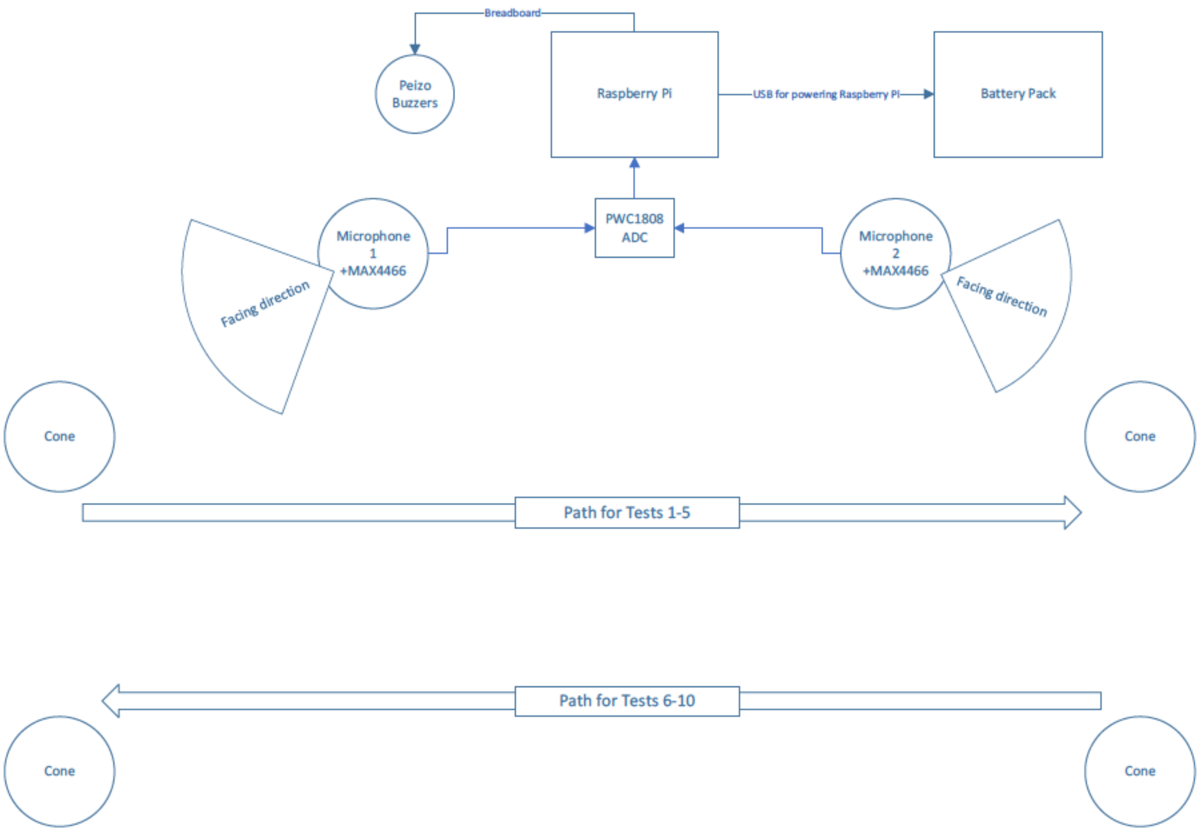


Figure 21 - Block Diagram of the Audio Testing Setup

Overview of Code

Once the amplifier was adjusted and the data was input to the Pi, it captured and analysed it using FFT. The below block diagram is a basic overview of the structure of this code, and over the next section I shall run through how specifically this code works. The layout for this code is also shown below in Figure 3.

Firstly, the data was captured from the I2C bus, and appended to the buffer until it filled (once every half second).

This audio was then run through the NumPy fast Fourier transform function, to analyse the frequency components of the input. From analysis of all of the waves, three 'trigger' frequencies were chosen, as these were always present when the bike passed.

These trigger frequencies were 572Hz, 2.7 kHz and 21.551kHz. The below figure was plotted in MATLAB to demonstrate the frequency analysis.

Once these Frequencies were detected, there uniqueness value was also calculated. This is the ratio of the power of this frequency compared to the surrounding frequencies.

This was all implemented using the code below. The full code is available in Appendix B.

```
# attempt to read byte from the device
```

```
indata = bus.read_byte(DEVICE_ADDRESS)

#print("Received data:", data)


        # if buffer is empty, this is first cycle, fill buffer

if !buffer

            buffer = indata

else if len(buffer) <= Fs/2

#if the buffer is over half a second long, save

buffer.append(indata);

else

# now doing an FFT on this buffer.

bufferFFT = np.fft.fft(buffer);
```

The Next step is the audio analysis. This was done by grouping the results from the FFT into 'bins' which surrounded frequencies of interest. These were created and used to determine both the presence and the uniqueness of the trigger frequencies. Once these had been defined and the average magnitude of the frequencies surrounding the trigger frequencies had been calculated, the Peizo Buzzer was activated.

This was single channel operation, however with dual channels, the operation of the code was similar however it simply uses two buffers and activates different piezo buzzers for either microphone which is triggered.

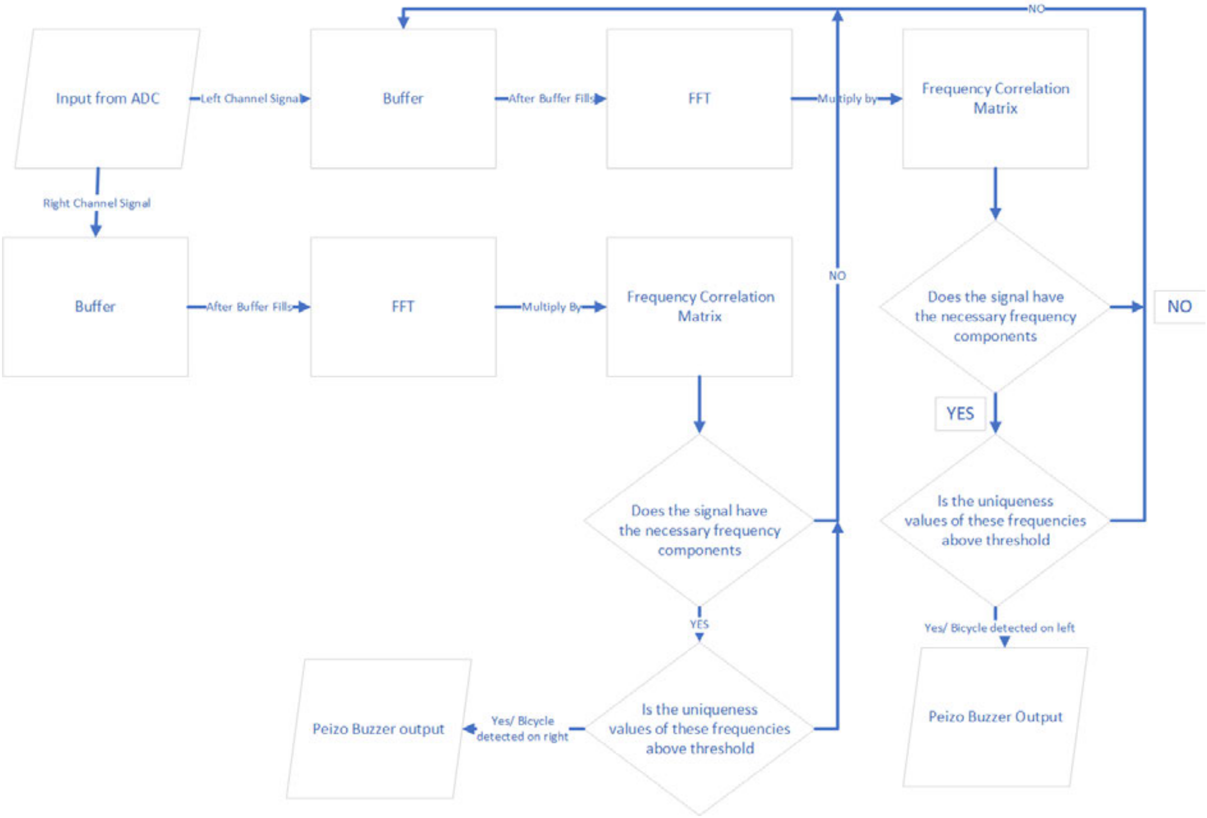


Figure 22 - Overview of Code Structure for Audio Analysis

Below is also set out the Circuit Diagram of the input microphones.

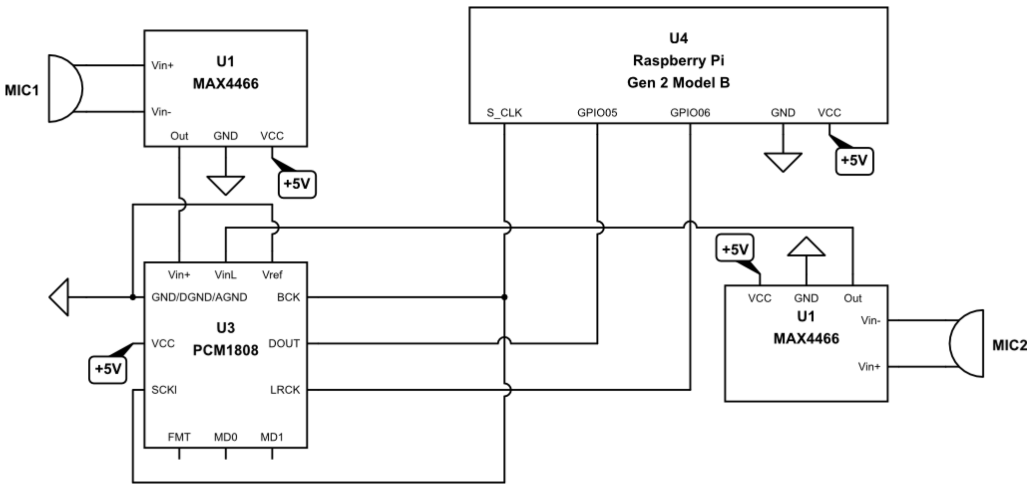


Figure 23 - Circuit Diagram of Audio Testing Setup

Testing

The above sections detailed how the testing of the audio sensing system was set up to capture the audio from the microphones, and how the code analysed this input to determine the passing of a bicycle.

The below section will detail how this setup was tested, the results of this testing and limitations of this testing.

This testing was setup using the setup detailed above and was initially done on flat ground. The cones as seen below set out a path for the bike to pass between, and the sensing capabilities for both directions were tested by riding the bike both directions five times.

While testing, a microphone was also attached to a nearby laptop to record alongside the microphones attached to the Raspberry Pi.

This was done to assist in the adjustment of the trigger frequencies as the audio recorded on the laptop could be easily run through a spectrum analyser to determine its component frequencies.

Initial Testing was completed using a USB microphone attached to the Raspberry Pi, and while this was somewhat successful, the USQB microphone had some limitations for running smoothly at high sample rates and consistently connecting on startup. This made the choice to use a simpler communication method such as I2C easy.

Initially the testing was not very successful, however adjustment of the microphones trigger frequencies, and adding the uniqueness detection rather than simply checking for the amplitude of the input all helped to increase its capabilities and reliability.

These methods increased the reliability of the sensing system, however there is still a few issues for future work.

In my testing, there was sometimes in which the device would be triggered unintentionally. After listening back to the recording, this was determining to be background noise from car driving in the nearby carpark. Once the location was moved, the sensing system was much more reliable.

Below is pictured a typical plot of a triggering FFT output vs a non-triggering output. As can be seen at each of the trigger voltages the bicycle data was higher, and as can easily be seen at the centre trigger frequency, this is a unique frequency to the bicycle.

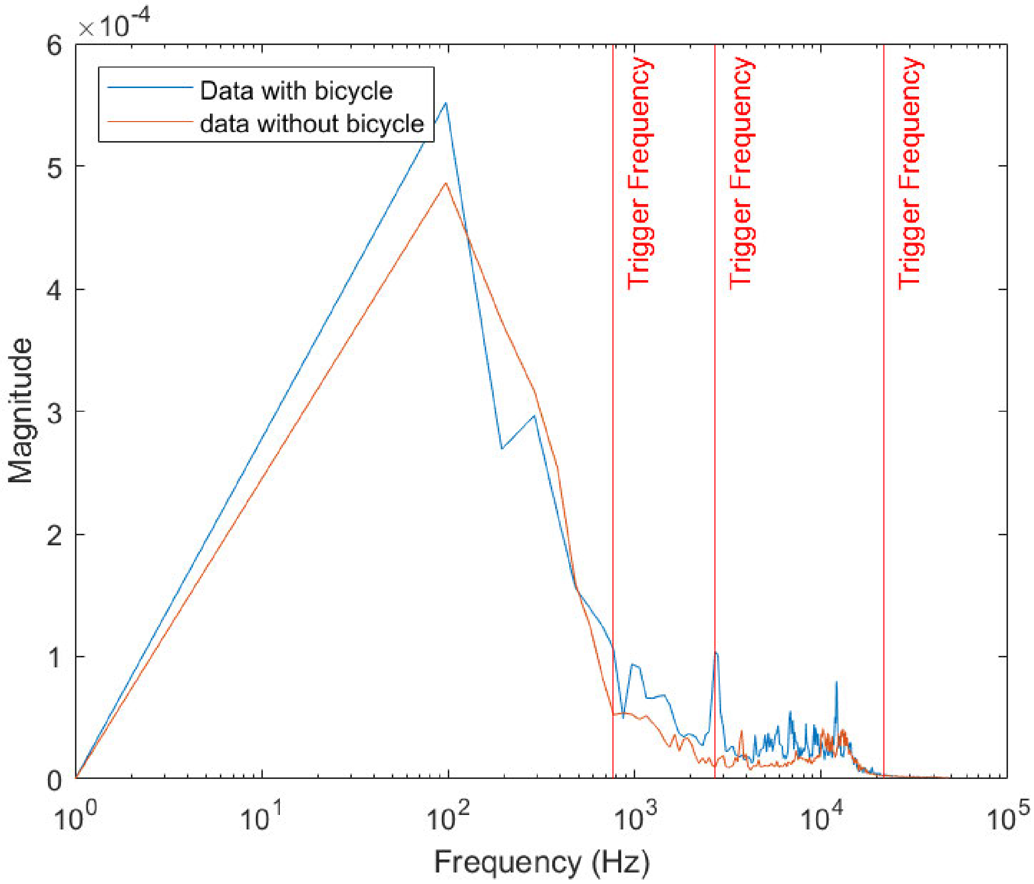


Figure 24 - Example of audio output from FFT of triggering signal

Some limitations of the testing procedure were that only a single bicycle was used. If more resources allowed, more bicycles should be used in the testing of this system, so that the system could become more robust and able to reliably detect many bicycles.

Another flaw in the system was the testing location. Although two locations were tested to attempt to account for changes in the audio profile of the ground, more work could be done to improve the reliability of the system in this way.

Below is pictured one of the locations used in the testing of this system. I shall discuss the results from testing of the system with the results from all of the testing methods below.

Figure 25 - Audio Sensing Setup for Testing

Vision Experimentation

For the Vision detection, the camera designed for the Raspberry Pi was used, as this was easily controllable by the Pi. In the below figures, the image setup can be seen. I chose a similar setup to the audio and vibration setup; however, the sensor was setup a lot closer to the Raspberry Pi as it is connected via a ribbon cable.

Figure 26 - Photograph of vision detection setup

This was how the vision setup was implemented. As the camera was set put quite low, the majority of the background was open sky, which helped out the motion detection.

The testing procedure was similar to the others, with the bicycle being ridden past the camera 5 times in each direction, recording when it triggered the piezo buzzer for an alarm.

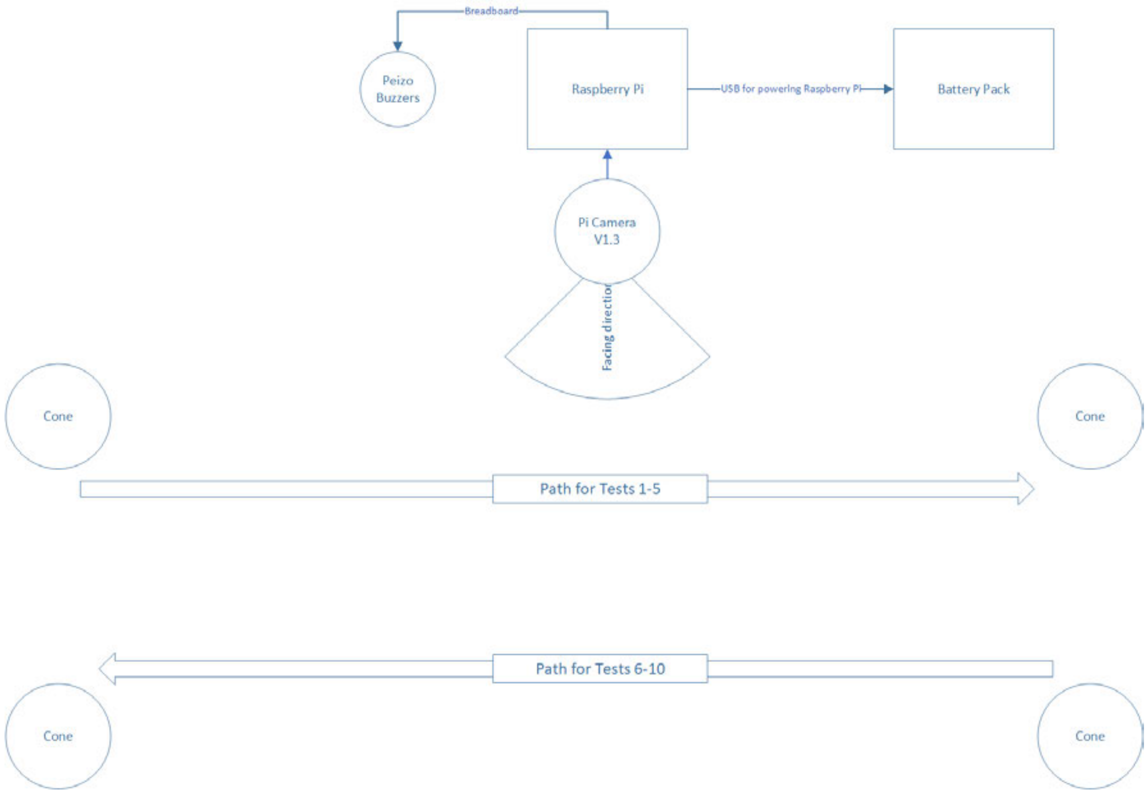


Figure 27 - Block diagram of vision setup

Below can be seen the structure for the code which was used. I shall also go over the code briefly, however all of the code shall be in appendix B if required.

This system was implemented in python using NumPy and the PiCamera Libraries.

Firstly, the camera was set to the correct resolution. The model used (Pi Camera v1.3) is capable of taking still images at a resolution of up to 2592 × 1944 pixels (Raspberry Pi Foundation 2023b). A resolution of 1920x1080 was chosen as this allowed the Pi to process the image in a more timely fashion and did not affect the sensing performance. Image showing the images converted to grayscale can be seen below.

The GPIO pins were also setup with the Peizo Buzzer on pin 23.

A continuous while loop was then implemented to continuously capture photographs every 0.5 seconds. Inside this, the camera was sent to capture an image at the specified resolution and convert it to grayscale. If this was the first image, the program then waited to implement the sensing until two photographs were taken.

This was implemented using this code.

```
#forever while loop
while(1)
   #capture the photograph to an array
   camera.capture(picamera.array.PiRGBArray(camera),output)
   #convert photograph to grayscale by averageing the RGB values
   greyscale = np.mean(output, axis=2, dtype=output.dtype)
   #now save to a buffer, for subtraction later if it is the first image (buffer is empty)
   if !buffer
      imageBuffer = greyscale;
```

Once two photographs were taken, the second one was compared to the first, using the NumPy function .isclose. This returns a matrix of the same size with a true element of if the corresponding elements in matrices are within a threshold of 50 and a false element if they are not. By inverting this matrix and summing all of the pixels where change was detected, it could be simply seen if there was a significant amount of movement in the picture. This was implemented as shown below.

```
   #if not the first image, subtract the two images
   difference = greyscale-imageBuffer;
   # if there are more than 20% of the image changed significantly
   # (over 50 change in colour) detect the image as having movement
changebyElement =
np.invert(np.all(np.isclose(greyscale,imageBuffer,rtol=50,equal_nan=FALSE)))*1;
   totalChange = np.sum(changebyElement);
```

If there was significant motion detected, the piezo buzzer would sound for 0.5 seconds before the code restarted to take a picture. If there was no motion detected, the code would wait for 0.5 seconds before retaking another photograph. This was implemented as shown below.

```
if  totalChange >= threshold
    #output to piezo buzzer for 0.5 seconds
    GPIO.output(peizoBuzzer,GPIO.HIGH)
    sleep(0.5) # Delay in seconds
    GPIO.output(peizoBuzzer,GPIO.LOW)
else
#if there is no movement detected, sleep until another photograph can be taken
    sleep(0.5)
```

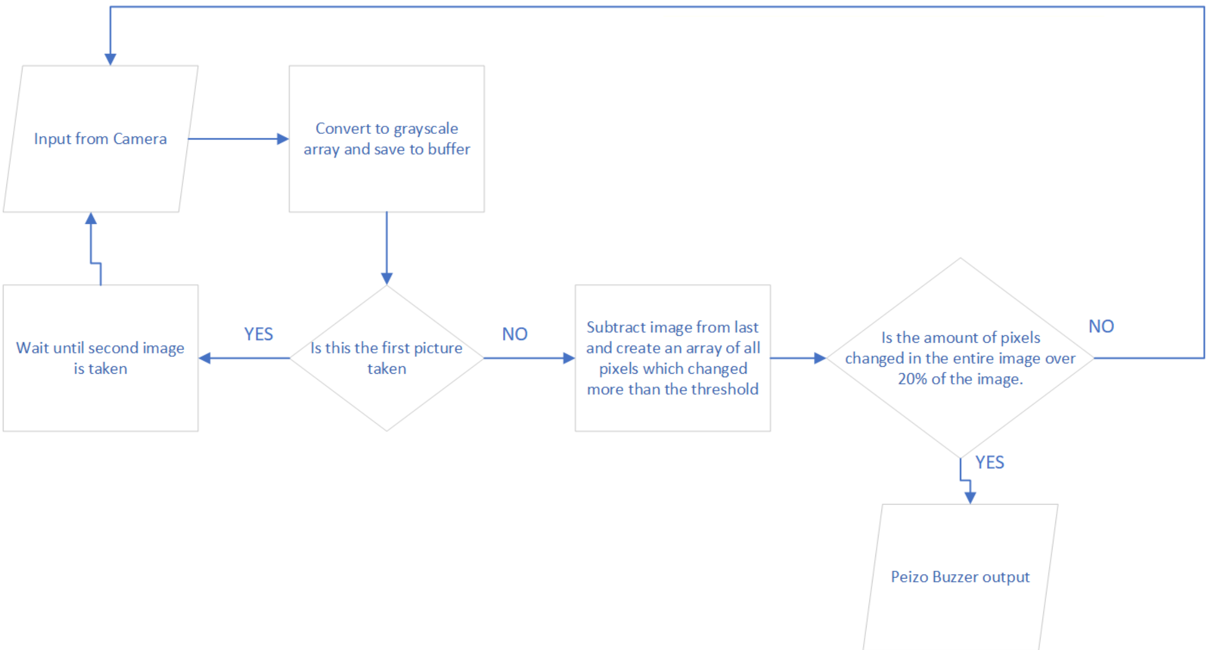


Figure 28 - Overview of working Vision sensing code

Below are some example results from this image detection method, the results from the testing shall be presented in the relevant section, however here some limitation from this process and ways to fix them.

Firstly, when I was testing in a different location, with a moving tree in the background, there was some difficulty calibrating the system to not trigger falsely, and also to trigger every time a bike passed. Secondly, this system really only detects movement at the moment, and does not differentiate between animal or bicycle or branch. Lastly, this image processing signal does not detect direction.

The first two of these issues can be mitigated by using a more advanced image classification algorithm. One as complex as a YOLO or similar system would require training and may be too complex for the computational power available. Much work has been done in this by many researchers, however the work of Lin and Young was useful, as it showed that a bicycle could be detected using a single side on photograph (Lin & Young, 2017). There method was based upon detecting the ellipses of the wheels, then using the geometric relationship between them and the tubes of the bicycles to determine the bicycles presence. The main issues they found were that if the image did not contain the entire image of the wheels, or even if a little bit of the wheels were out of frame, the ellipse detection did not work at all. This would be an issue for this system as well as the sides of the bike are often not in frame as the photograph is only take every 0.5 seconds.

A modified version of this detection method could be implemented to simply robustly determine if there were any ellipses in the picture, as there is not likely to be any other source of ellipses other than bikes being photographed.

Figure 29 - Example images from setup turned grayscale for image processing.

The third issue of the lack of directional sensing could also be rectified by splitting the resultant image into ‘zones’ horizontally, so that if the image detected more movement in the right zones first, and then detected movement in the left zones, the bike would be assigned a direction of right to left.

This was attempted to be implemented in the code in appendix B however was never utilised.

A final way in which the system could be improved is by limiting the vertical field of view which the camera captures as this would reduce the computation power required for detection. This would ideally be dynamically done within the code, however, could also be done manually.

One idea would be that the field of view is drastically reduced in the vertical direction, however if motion is detected, another picture is immediately taken to determine if it is a bicycle or not.

Figure 30 - Resultant image of motion detected.

Figure 31 - Resultant image when no motion is detected

Vibration Experimentation

The Testing Setup for the vibration sensing system was very similar to the sensing of the Audio, as it also used an Analog to Digital converter and the I2C transfer protocol to communicate to the

Raspberry Pi, as shown in the Figure below. As can be seen, in the ideal system two SM-24 Geophones would be connected to a 16-bit analog to digital converter which is also capable of amplifying the input voltages. These would then be connected to the Raspberry Pi via a I2C connection.

In My testing, I utilised an accelerometer which had an onboard ADC, and was able to communicate directly with the Pi. This showed some success, however more sensitivity was needed as the sensor only triggered when riding close beside it. To help with this issue a Geophone was purchased. Geophones are typically used to measure seismic activity and consist of a metal weight suspended by springs. This weight has copper windings around it and is surrounded by large magnets. When vibrations reach the device, it causes the weight to move, inducing a voltage in the copper windings.

Commercial versions of these devices can be very expensive however a hobby version was purchased for this experimentation. This was quite hard to find in stock however and only arrived in the past few weeks, so time to experiment with it has been very limited. Signals from the Geophone were able to be received from the Pi, and analysed over time to determine if there were continuous vibrations in the ground consistent with a bicycle.

The Code used to complete this is shown below.

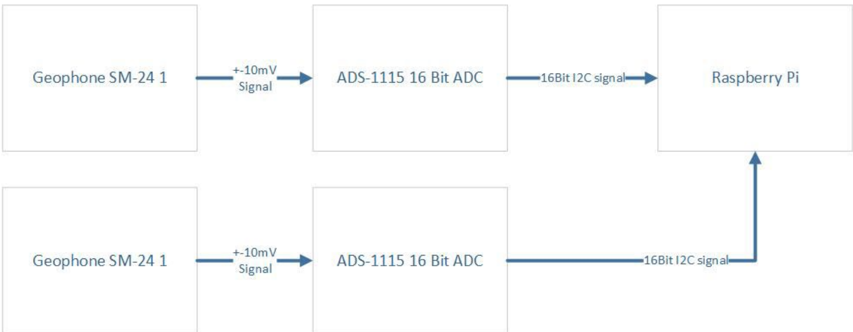


Figure 32 - Signal pathway for the vibration sensing system

The below Figure shows the layout of the code used in this project, and I shall explain the working of the code, using snippets of the code as necessary. The entire code is available in Appendix B below.

Firstly when the data is input from the vibration sensor, it is added to a buffer and compared to a threshold value. If the measured vibration is above that, a count is added to the vibration count, which ensures that it is not just a random measurement. If the vibration count is not added to consistently, it shall not trigger that a bike has been detected, as every time a sample comes through which is below the threshold the count is decreased by two.

If the counter reaches the count of 10, the output Peizo buzzer is activated and the count is reset.

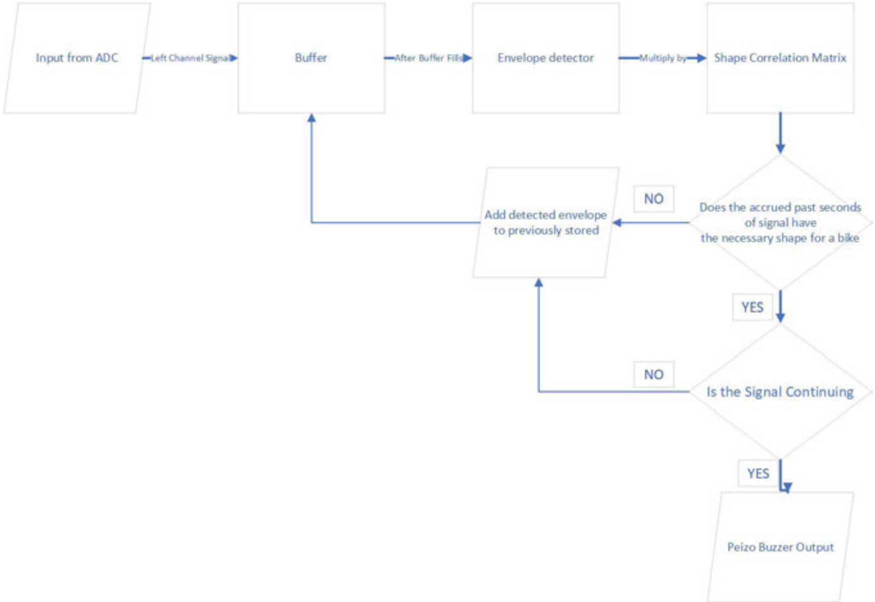


Figure 33 - Structure For code used to analyse vibration

Below is Pictured the Vibration Sensing setup, it was moderately successful in its ability to sense passing objects, and was able to moderately tell if the passing object was a human or a bicycle as long as the steps of the human were far enough apart.

A few ways to improve on this sensing system would be to store the vibration value and match it to a given profile recorded when a bicycle passes.

Another issue with the current setup is that it does not sense direction at all. This can be rectified with the purchase of another Geophone, however this was beyond the budget of this project.

Considerations when installing this project are that the geophone needs to be firmly placed on the ground as this was an issue I encountered when attempting to experiment with it.

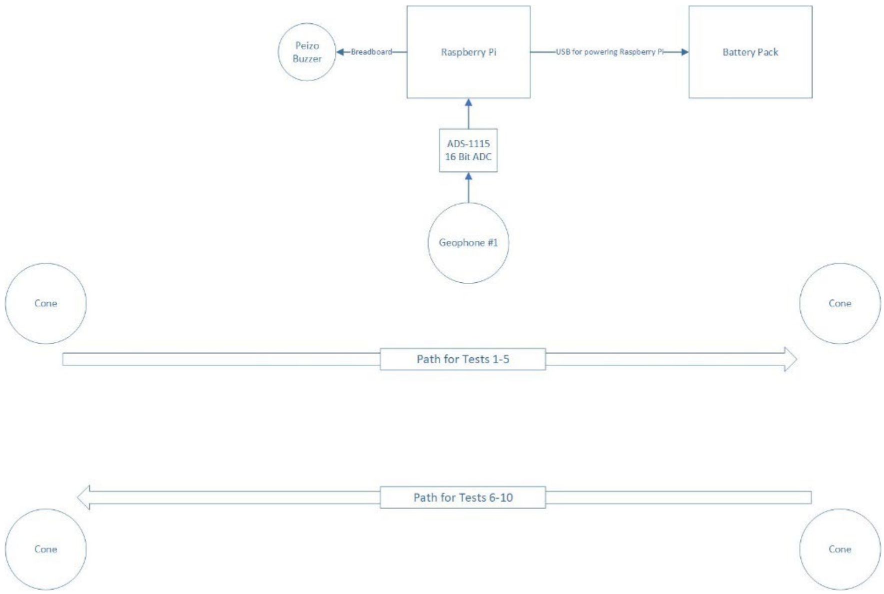


Figure 34 - Setup for the Vibration experiments

Results

In the above sections the experimentation layout for each of the three technologies has been explained, along with how this data was processed by the Raspberry Pi. This section shall Explain the testing procedure in greater detail and summarise the results. Below, possible reasons for each of these results shall be expanded upon and ways to improve the sensors performance are explored. After that the outcome of this project of summarised and future work is laid out to be completed.

Firstly, the results from the tests can be seen Below in Table 3. Each of the sensing systems were tested at least three times, with the audio system being tested many more times than that in its calibration.

To conduct the tests, the bicycle was ridden past the sensor in the setups listed above, and the piezo buzzers were used to determine if the bicycle was successfully detected or not. For each of the tests a recording device was left by the Raspberry Pi to ensure that the correct amount of passes were detected. This was finally repeated by a pedestrian to test the systems effectiveness at sensing pedestrians.

The results of these three tests are shown below as a score out of ten, as for each test, the bicycle was ridden in front of the sensor system five times in each direction. Finally, to test the ability of these systems to reject false positives, a pedestrian walked along the same path as the bicycle 10 times to see if the sensor system would detect that. A higher score on that test means that the system did not detect that the pedestrian had passed. This is useful information for future work as if this was implemented as a safety system, it would be best if pedestrians set it off as well.

Table 3 – Results From Testing Sensing systems

Technology	Bike Passing Test #1	Bike Passing Test #2	Bike Passing Test #3	Bike Passing Test #4	Total False Positives	Pedestrian Passing Test #1	Directional	TOTAL
Audio	6	5	8	9	-1	3	**Y**	**34**
Visual Sensing	10	8	9	9	-7	10	**N**	**29**
Vibration	6	5	7	8	0	5	**N**	**31**

Visual Sensing

The visual sensing method was able to sense that a bicycle was passing it correctly 90% of the time, and the pedestrian passing it 100% of the time however had four false positives over this time.

In the current setup, this sensing system is very effective at sensing any movement in front of it. It was able to Detect the most bicycles out of any of the systems however also had the most false positives out of the three sensing systems. On the whole this sensing method was very effective at detecting movement with how it was implemented in the current configuration, however also has also of potential for improvement.

Areas for improvement include the possibility of using an infrared camera and light source for nighttime sensing. This could also help alleviate the issue of the background changing and triggering the sensor falsely.

Another avenue for improvement is the motion detection algorithm. The current one used only looks for a change in the picture and is not able to tell direction from this or determine if the object in front of it is a bicycle.

On way to implement this change would be to reduce the aspect ratio of the camera temporarily so that it is being used mostly as a motion sensor. Once motion is detected however two full frame photographs are captured and analysed to determine of the object is a bicycle and its direction.

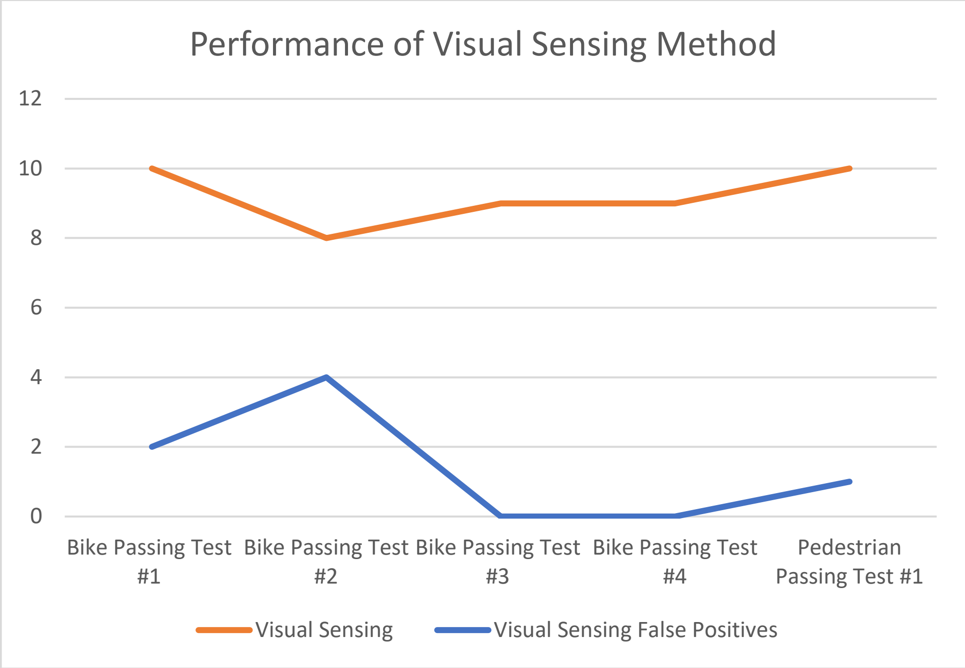


Figure 35 - Summary of Results from visual tests

Audio Sensing

As shown above, the audio sensing system was able to detect the bicycle 77.5% of the times the bicycle passed, was able to detect the pedestrian 30% of the times they passed, and only had one false positive during that time. Its performance was especially good in the last two tests when the trigger frequencies were more dialled in.

This was the only one of the systems was able to be setup and get the directionality functionality working using two microphones. It also had one of the lowest false positives of all the sensing methods, however it also did not trigger when the pedestrian walked past, and so would have to be modified to sense pedestrians reliably. Its reliability in detecting bikes could also be improved by using a confidence-based model over multiple seconds to determine the presence of a bicycle rather than simply using one section of samples lasting under a second.

This approach could also possibly determine the direction of a bicycle from one microphone based on the profile of the confidence level.

Another spot for improvement in this system would be the use of multiple bicycles in the testing phase. The chosen trigger frequencies work well with the chosen bicycle; however, no testing has been done to determine if they would work on multiple bicycles or all bicycles.

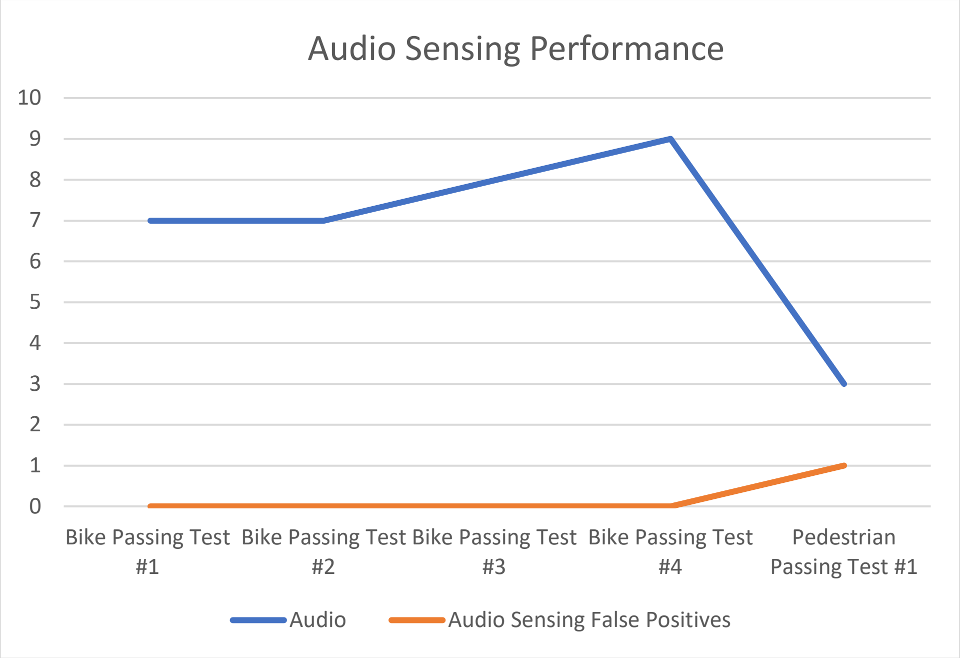


Figure 36 - Performance of Audio Sensing

Vibration Sensing

The vibration sensing method was able to sense that a bicycle was passing it correctly 65% of the time, and the pedestrian passing it 50% of the time however had no false positives over this time.

Vibration Sensing is a promising technology using the right equipment, however unfortunately for this project the right sensor was available for long enough to develop a good system surrounding it.

This system also did not sense the direction of the bicycle, however adding a secondary vibration sensor could be implemented in a similar way to the audio sensing system. The chosen approach of simple averaging the vibration magnitude in a given window and comparing it to a threshold works, however similar to the audio sensing system, if another bike was tested, perhaps it would not produce the same vibrations. Thus, a confidence based system similar to that proposed for the audio system could be implemented, as this would allow for more sensitive threshold tolerances. Another issue with this system is that it depends upon the sensor being firmly mounted onto the ground. Ideally this would be done with stakes into the ground however that was not available on the soil chosen.

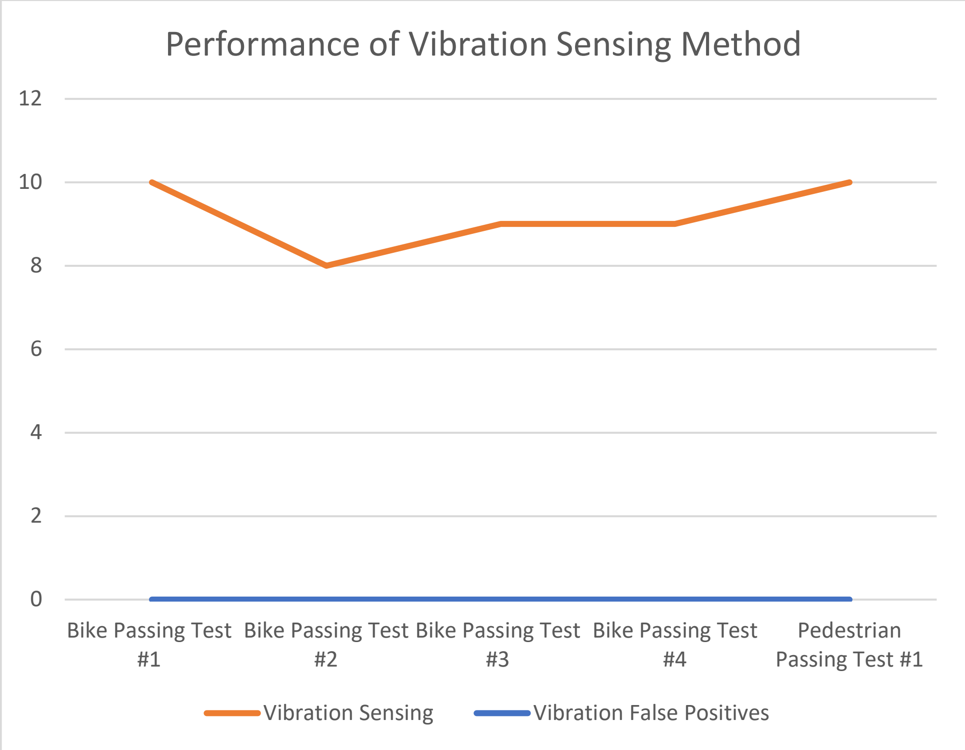


Figure 37 - Performance of Vibration Sensor System

Summary of Results

The results of my experimentation were a solid system for sensing bikes presence and direction in the audio sensing system, and two other solid sensing systems if optimised.

In the below Figure, the results are summarised and compared.

This project has conducted background research into the use of many technologies used for sensing bicycles and have reviewed existing bike sensing systems and their technologies. Once three sensing technologies were decided upon to test in more detail, a configuration was conceptualised for each, including the type and position of the sensors.

Once suitable sensors had been selected, a software development environment and computational technology were chosen in Python running on the Raspberry Pi.

Initial prototyping for the audio sensing technology was conducted using a USB webcam microphone. This to collect sensor data for analysis and the time when a bicycle passed was able to be determined based upon the presence of certain 'trigger' frequencies.

After receiving all of the required sensors, testing was continued along with refinement of the sensing algorithms. Unfortunately, a prototype was unable to be deployed into the field to gather more data.

All of the sensing methods investigated can be improved upon through methods listed above, however they are working to the best of their current ability given the resource constraints of this project. The below section contains recommendations for avenues of future work, and advice as to which paths may be not worth investigating.

Conclusions

This project set out to

1. Conduct initial background research bike sensing technologies and their applications in current research, ideally finding a technology which has been used to sense speed and direction of bikes.
2. Review existing bike sensing systems and their underlying technologies. Consider the use case of the technologies and their running requirements and decide on one or more sensing technologies which would be appropriate to use in this system.
3. Conceptualize a suitable configuration for the position and type of sensors to reliably sense the presence and direction of bikes passing in front of them with minimal false positives and good reliability.
4. Assess sensor requirements for necessary capability of system and costs.
5. Select sensors, a software development environment and microcontroller.
6. Construct an initial prototype to facilitate sensor data collection.
7. Inspect data and develop program to analyze it for bike presence and direction.
8. Evaluate sensor effectiveness at determining the presence and direction of passing bikes.

This thesis addressed requirements 1 and 2 during the literature review, as well as researching computational and communication technologies. The bike sensing technologies listed in them were decided upon in the first part of the methodology. The methodology also completed Requirements 3-7 were completed in the methodology, and throughout the experiments undertaken. Requirement 8 was completed in the above section assessing the sensors performance. Below future work relating to this thesis shall be described, followed by the references and appendices of this project.

Future Work

Each one of these sensing technologies can be improved substantially, given the right amount of resources.

For the Visual sensing method, it seems that reducing the vertical aspect ratio temporarily so that the entire frame does not have to be processed could improve the ability of the system to determine the object in front of it as it would then be able to take a full frame shot. Another way to improve its performance would be to implement an ellipse detection method, however this would only improve its ability to sense bicycles. An option I have not explored is to use an image classification method such as YOLO to sense whatever objects are in the frame.

For the audio sensing method, two possibilities of future work include implementing a confidence-based algorithm and extending the testing of different bicycles. Of these two the testing of different bicycles would be the most important, however in order to be able to sense the widest array of bicycles possible, a more complex detection algorithm will need to be utilised, based upon many more trigger frequencies. An avenue which may not be worth exploring is seeing if it is possible to detect the direction of the bicycle using only one microphone as the microphones are by far the cheapest part of the system.

For the Vibration sensing system, there are two main areas which could be studied in the future. These are similar to the audio sensing method, the first being to implement a confidence-based algorithm to determine of there is a bicycle approaching. This would allow the threshold vibration to be lower and therefore be more accurate. The second of these areas is to test different terrain in different weather conditions. It was initially planned to test this system in the same location when it was wet and dry, however resources ran out to do this. This is a unique problem to mountain biking as most previous vibration detection systems are implemented in urban areas where the water runs off easily and quite fast. One area not worth working on more would be the use of accelerometers to detect vibrations, in this project it was decided that the bicycle needed to be so close to the vibration sensor that it posed a safety risk unless the sensor was very well protected.

Finally, in general the biggest portion of future work is the turning of these sensing systems into a safety system to be implemented on mountain bike trails. This would include choosing a suitable communication method, power supply and rider alert system. In the future these sensing systems could be used in a safety system however they are not robust enough to implement immediately.

Appendix B – Code

Audio Code

```
import smbus

import time

import numpy as np


# Initialize the I2C bus

bus = smbus.SMBus(1)

#sampling frequency

Fs = 48000;

# Address of the I2C device

DEVICE_ADDRESS = 0x68

str buffer = NULL

#trigger frequiencies

triggerFs = [570,4900,21500];


#trigger threshold

triggerThresh = 1.3;


try:

    while True:

        # attempt to read byte from the device

        indata = bus.read_byte(DEVICE_ADDRESS)

        #print("Received data:", data)


                    # if buffer is empty, this is first cycle, fill buffer
```

```
    if !buffer
                    buffer = indata
    else if len(buffer) <= Fs/2
    #if the buffer is over half a second long, save
    buffer.append(indata);
    else
    # now doing an FFT on this buffer.
    bufferFFT = np.fft.fft(buffer);
    #converting to single sided, non complex

    n = buffer.size
    timestep = 1/Fs
    # now splitting into seperate bins
    freq = np.fft.fftfreq(n, d=timestep)
    # now analysing this, remembering the frequencies chosen
    if

    #sleep before repeating cycle
              time.sleep(0.1)
except KeyboardInterrupt:
  print("Program stopped by user")
```

```
finally:
    bus.close()
```

Audio Plotting Code

```
clc,clear


[y,Freq] = audioread("Recording (4).wav");
mod(length(y),Freq)
offset =25;
shortY = y(12*Freq:15*Freq,:);
lengthSection = Freq/2;
F = Freq*(0:(lengthSection/2))/lengthSection;
for i = 1:100
 analysisSection(i,:) = y((offset+i-1)*lengthSection+1:(offset+i)*lengthSection);


audioFFT(i,:)  = abs(fft(analysisSection(i,:)))/lengthSection;
audioFFTsingle(i,:) = audioFFT(i,1:lengthSection/2+1);


end


for i = 2:100
    if audioFFTsingle(i,6)>1.6*audioFFTsingle(i-1,6) &&
audioFFTsingle(i,23)>1.6*audioFFTsingle(i-1,23)
disp('bike')
else disp('no bike')
    end
```

```
end

%plot(F,audioFFTsingle)


bins = round(linspace(1,12000,500));

for i = 1:100

for j = 2:length(bins)

 averageBins(i,j) = mean(audioFFTsingle(i,bins(j-1):bins(j)),"all");


end

end

 range = linspace(1, 48000,500);

figure


semilogx(range,averageBins(1,:),range,averageBins(10,:));%,range,averageBins(9,:),range,averageBins(4,:));%,...

   % range,averageBins(5,:),range,averageBins(6,:),range,averageBins(7,:),range,averageBins(8,:));

   %     1:500,averagebins(9:),1:500,averagebins(10,:));

xline(770,'r','Trigger Frequency')

xline(2700,'r','Trigger Frequency')

xline(21500,'r','Trigger Frequency')


   legend('Data with bicycle','data without bicycle','location','nw');

   xlabel('Frequency (Hz)');

   ylabel('Magnitude')
```

Vision Code

```
    # Import neccesary packages

    import RPi.GPIO as GPIO
```

```
from picamera import PiCamera as camera
from picamera import picamera.array
from time import sleep
import numpy as np
# For peizo buzzer setup
#Select GPIO mode
GPIO.setmode(GPIO.BCM)
#Set buzzer - pin 23 as output
peizoBuzzer=23
GPIO.setup(peizoBuzzer,GPIO.OUT)
#set the resolution value
camera.resolution = (1920, 1080)

#forever while loop
while(1)
  #capture the photograph to an array
  camera.capture(picamera.array.PiRGBArray(camera),output)
  #convert photograph to grayscale by averageing the RGB values
  greyscale = np.mean(output, axis=2, dtype=output.dtype)
  #now save to a buffer, for subtraction later if it is the first image (buffer is empty)
  if !buffer
    imageBuffer = greyscale;

  #if not the first image, subtract the two images
  difference = greyscale-imageBuffer;
  # if there are more than 20% of the image changed significantly
```

```
    # (over 50 change in colour) detect the image as having movement
    changebyElement =
np.invert(np.all(np.isclose(greyscale,imageBuffer,rtol=50,equal_nan=FALSE)))*1;
    totalChange = np.sum(changebyElement);
    #compute the threshold
    threshold = 0.2*1920*1080;
    if  totalChange >= threshold
      #output to peizo buzzer for 0.2 seconds
      GPIO.output(peizoBuzzer,GPIO.HIGH)
      sleep(0.2) # Delay in seconds
      GPIO.output(peizoBuzzer,GPIO.LOW)
      #sleep for 0.3 seconds to take another photograph

    else
    #if there is no movement detected, sleep until another photgraph can be taken
      sleep(0.5)
```

Vibration Code

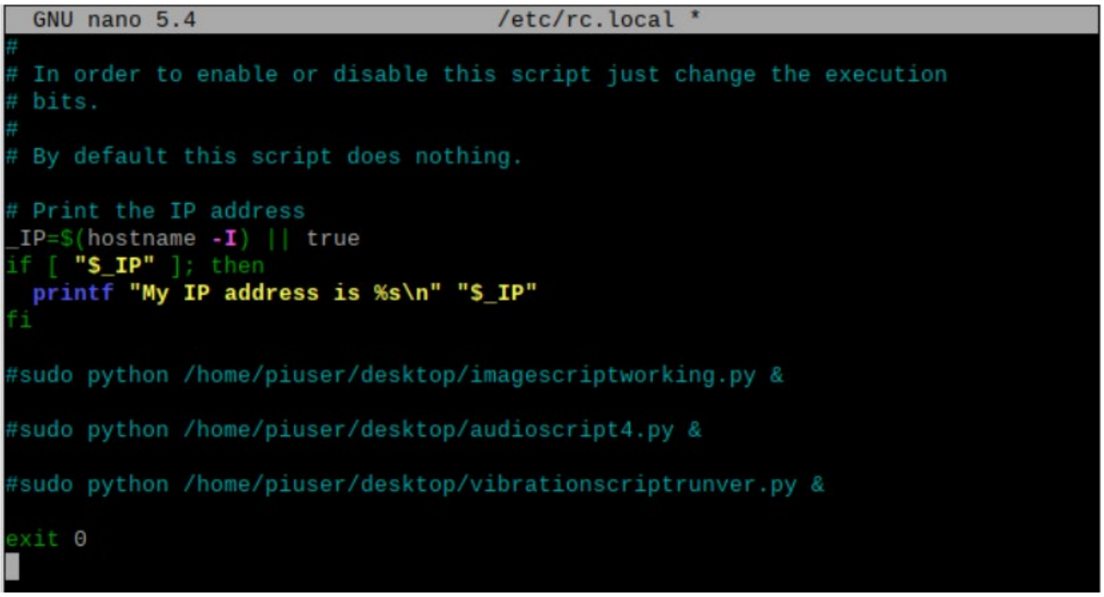
```
# Import neccesary packages
    import RPi.GPIO as GPIO
    from picamera import PiCamera as camera
    from picamera import picamera.array
    from time import sleep
    import numpy as np

            # For peizo buzzer setup
    #Select GPIO mode
    GPIO.setmode(GPIO.BCM)
    #Set buzzer - pin 23 as output
    peizoBuzzer=23
    GPIO.setup(peizoBuzzer,GPIO.OUT)

    # Initialize the I2C bus
    bus = smbus.SMBus(1)
    #sampling frequency
    Fs = 1000;
    # Address of the I2C device
    DEVICE_ADDRESS = 0x68;

    # threshold value for vibrations
    threshold = 640;
    # count for number of times vibration has been sensed
```

```
vibrationCount = 0;
try:
while True:
    # attempt to read byte from the device
    indata = bus.read_byte(DEVICE_ADDRESS)
    #print("Received data:", data)

                    # if buffer is empty, this is first cycle, fill buffer
    if !buffer
                            buffer = indata;
    else
    #if buffer already has items in it, calculate the average value of these items and compare to the
threshold value
    if buffer > threshold
        # add 1 to count of vibrations measured
        vibrationCount ++;

        if vibrationCount > 10
        #output to peizo buzzer for 0.2 seconds
        GPIO.output(peizoBuzzer,GPIO.HIGH)
        sleep(0.2); # Delay in seconds
        GPIO.output(peizoBuzzer,GPIO.LOW)
        # reset the vibration count
        vibrationCount = 0;

    else
```

```
            if vibrationCount >=2
                vibrationCount = vibrationCount-2;


        # assign input data to buffer
        buffer = indata;
        #sleep before repeating cycle
                time.sleep(0.1)
except KeyboardInterrupt:
    print("Program stopped by user")
finally:
    bus.close()
```

Bash Code

Below is the Bash Code For running scripts on startup of Pi.



```
  GNU nano 5.4                    /etc/rc.local *
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

#sudo python /home/piuser/desktop/imagescriptworking.py &

#sudo python /home/piuser/desktop/audioscript4.py &

#sudo python /home/piuser/desktop/vibrationscriptrunver.py &

exit 0
```

Figure 38 - BASH code for running of python script

References

Abildso, C. G., Haas, V., Daily, S. M., & Bias, T. K. (2021). Field Test of a Passive Infrared Camera for Measuring Trail-Based Physical Activity [Methods]. *Frontiers in Public Health*, *9*. https://doi.org/10.3389/fpubh.2021.584740

Adafruit (2023a). *ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier*. Retrieved 20/10/2023 from https://www.adafruit.com/product/1085

Adafruit (2023b). *Electret Microphone Amplifier - MAX4466 with Adjustable Gain*. Retrieved 18/10/2023 from https://www.adafruit.com/product/1063

Adafruit (2023). *ADXL345 Digital Accelerometer*. Retrieved 20/10/2023 from https://learn.adafruit.com/adxl345-digital-accelerometer/overview

Arduino. (2023a). *Arduino Mega Rev3*. Retrieved 17/10/2023 from https://store-usa.arduino.cc/products/arduino-mega-2560-rev3

Arduino. (2023b). *Arduino Uno Rev3*. Retrieved 17/10/2023 from https://store.arduino.cc/products/arduino-uno-rev3

Arduino, S. A. (2023). Arduino. *Arduino LLC*, *372*.

Australia, E. (2019). Code of ethics and guidelines on professional conduct. *Available at [Accessed 12 January 2021]. https://www. engineersaustralia. org. au/sites/default/files/resource-files/2020-02/828145% 20Code% 20of% 20Ethics*, *202020*.

Automation Direct (2023). *C0-00DD1-D Click PLC Product Overview*. Retrieved 17/10/2023 from https://www.automationdirect.com/adc/shopping/catalog/programmable_controllers/click_plcs_(stackable_micro_brick)/plc_units/c0-00dd1-d

Baas, J., Galton, R., & Biton, A. (2016). FHWA bicycle-pedestrian count technology pilot project : summary report [Tech Report]. https://rosap.ntl.bts.gov/view/dot/12416

Babiuch, M., Foltýnek, P., & Smutný, P. (2019). Using the ESP32 microcontroller for data processing. 2019 20th International Carpathian Control Conference (ICCC),

Bedeer, E., Pugh, J., Brown, C., & Yanikomeroglu, H. (2017). Measurement-based path loss and delay spread propagation models in VHF/UHF bands for IoT communications. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall),

Bernas, M., Płaczek, B., Korski, W., Loska, P., Smyła, J., & Szymała, P. (2018). A Survey and Comparison of Low-Cost Sensing Technologies for Road Traffic Monitoring. *Sensors*, *18*(10), 3243. https://www.mdpi.com/1424-8220/18/10/3243

Cho, H., Rybski, P. E., & Zhang, W. (2010, 19-22 Sept. 2010). Vision-based bicycle detection and tracking using a deformable part model and an EKF algorithm. 13th International IEEE Conference on Intelligent Transportation Systems,

Collotta, M., Pau, G., Talty, T., & Tonguz, O. K. (2018). Bluetooth 5: A concrete step forward toward the IoT. *IEEE Communications Magazine*, *56*(7), 125-131.

CT Technology (2023). *ITERIS: VANTAGE NEXT® VIDEO DETECTION*. Retrieved 18/10/2023 from https://ct-technologyinfo.com/vehicle-bicycle-and-pedestrian-detectors/

Djukanović, S., Matas, J., & Virtanen, T. (2020, 19 Oct.-13 Nov. 2020). Robust Audio-Based Vehicle Counting in Low-to-Moderate Traffic Flow. 2020 IEEE Intelligent Vehicles Symposium (IV),

Eco-Counter. (2022). *PYRO-Box*. https://www.eco-counter.com/en/produits/pyro-range/pyro-box/

Fan, X., Susan, F., Long, W., & Li, S. (2017). Security analysis of zigbee. *MWR InfoSecurity*, *2017*, 1-18.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627-1645. https://doi.org/10.1109/TPAMI.2009.167

Foubert, B., & Mitton, N. (2020). Long-Range Wireless Radio Technologies: A Survey. *Future Internet*, *12*(1), 13. https://www.mdpi.com/1999-5903/12/1/13

Grunwald, A., Schaarschmidt, M., & Westerkamp, C. (2019, 15-16 May 2019). LoRaWAN in a rural context: Use cases and opportunities for agricultural businesses. Mobile Communication - Technologies and Applications; 24. ITG-Symposium,

Hostettler, R. (2009). Traffic counting using measurements of road surface vibrations. In.

Kumar, R., Roopa, A., & Sathiya, D. P. (2015). Arduino ATMEGA-328 microcontroller. *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng*, *3*(4), 27-29.
Laptev, I. (2009). Improving object detection with boosted histograms. *Image and Vision Computing*, *27*(5), 535-544.
Lavric, A., Petrariu, A. I., & Popa, V. (2019, 29-30 Aug. 2019). SigFox Communication Protocol: The New Era of IoT? 2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI),
Lin, Y.-B., & Young, C.-P. (2017). High-precision bicycle detection on single side-view image based on the geometric relationship. *Pattern Recognition*, *63*, 334-354. https://doi.org/https://doi.org/10.1016/j.patcog.2016.10.012
Microchip. (2023). *PICDEM PIC18 Explorer Demonstration Board*

User's Guide. Retrieved 17/10/2023 from https://ww1.microchip.com/downloads/en/DeviceDoc/50001721C.pdf
Nano, A. (2018). Arduino Nano. *A MOBICON Company*.
Noyce, D. A., Gajendran, A., & Dharmaraju, R. (2006). Development of Bicycle and Pedestrian Detection and Classification Algorithm for Active-Infrared Overhead Vehicle Imaging Sensors. *Transportation Research Record*, *1982*(1), 202-209. https://doi.org/10.1177/0361198106198200125
NVIDIA. (2023). *Jetson Nano Developer Kit*. Retrieved 17/10/2023 from https://developer.nvidia.com/embedded/jetson-nano-developer-kit
q-free (2023). *HI-TRAC® CMU*. Retrieved 20/10/2023 from https://www.q-free.com/product/hi-trac-cmu-bicycle-and-pedestrian-monitoring-system/
Qiao, L., Zheng, Z., Cui, W., & Wang, L. (2018, 20-22 Oct. 2018). A Survey on Wi-Fi HaLow Technology for Internet of Things. 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2),
Raspberry Pi Foundation (2023a). *Raspberry Pi 2 Model B*. Retrieved 17/10/2023 from https://www.raspberrypi.com/products/raspberry-pi-2-model-b/
Raspberry Pi Foundation (2023b). *Raspberry Pi Documentation* Retrieved 18/10/2023 from https://www.raspberrypi.com/documentation/accessories/camera.html
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition,
rfidtiming (2017). *Cycling - RFID race timimg systems*. Retrieved 19/10/23 from https://rfidtiming.com/project-details/cycling/
Rivas, J., Wunderlich, R., & Heinen, S. J. (2016). Road vibrations as a source to detect the presence and speed of vehicles. *IEEE Sensors Journal*, *17*(2), 377-385.
Ryus, P., Ferguson, E., Laustsen, K. M., Prouix, F. R., Schneider, R. J., Hull, T., & Miranda-Moreno, L. (2014). *Methods and technologies for pedestrian and bicycle volume data collection*. Citeseer.
seeedstudio (2019). *PIR SENSOR: OVERVIEW, APPLICATIONS AND PROJECTS*. Retrieved 20/10/2023 from https://www.seeedstudio.com/blog/2019/08/03/pir-sensor-introduction-and-how-pir-motion-sensor-works-with-arduino-and-raspberry-pi/
Silva, J. d. C., Rodrigues, J. J. P. C., Alberti, A. M., Solic, P., & Aquino, A. L. L. (2017, 12-14 July 2017). LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities. 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech),
Sinha, R. S., Wei, Y., & Hwang, S.-H. (2017). A survey on LPWA technology: LoRa and NB-IoT. *ICT Express*, *3*(1), 14-21. https://doi.org/https://doi.org/10.1016/j.icte.2017.03.004
SparkFun (2023). *Geophone - SM-24, with Insulating Disc*. Retrieved 20/10/2023 from https://www.sparkfun.com/products/11744

Stankovski, S., Ostojić, G., Baranovski, I., Tegeltija, S., & Smirnov, V. (2022). Robust automation with PLC/PAC and edge controllers. *IFAC-PapersOnLine*, *55*(4), 316-321. https://doi.org/https://doi.org/10.1016/j.ifacol.2022.06.052

Texas Instruments (2023). *Single-ENDED,ANALOG-input24-Bit,96-kHzSTEREO A/D converter - ti.com*. Retrieved 18/10/2023 from https://www.ti.com/lit/ds/symlink/pcm1808.pdf

Tsai, W.-F. (2011). *A Low Cost RFID Tracking and Timing System for Bike Races* The Ohio State University]. http://rave.ohiolink.edu/etdc/view?acc_num=osu1298028620

Turner, S., Middleton, D. R., Longmire, R., Brewer, M., & Eurek, R. (2007). *Testing and evaluation of pedestrian sensors*.

Wisen. (2022). *WHISPER NODE - LORA*. Retrieved 10/10/2022 from https://wisen.com.au/store/products/whisper-node-lora/

Yussoff, Y., Abidin, H. Z., Rahman, R. A., & Yahaya, F. H. (2010, 16-18 April 2010). Development of a PIC-based wireless sensor node utilizing XBee technology. 2010 2nd IEEE International Conference on Information Management and Engineering,

Zu, X., Zhang, S., Guo, F., Zhao, Q., Zhang, X., You, X., . . . Yuan, X. (2017). Vehicle Counting and Moving Direction Identification Based on Small-Aperture Microphone Array. *Sensors*, *17*(5), 1089. https://www.mdpi.com/1424-8220/17/5/1089

Appendix C – Risk Management Plan

University of Southern Queensland **Offline Version**

USQ Safety Risk Management System

Note: This is the offline version of the Safety Risk Management System (SRMS) Risk Management Plan (RMP) and is only to be used for planning and drafting sessions, and when working in remote areas or on field activities. It must be transferred to the online SRMS at the first opportunity.

Safety Risk Management Plan – Offline Version			
Assessment Title:	Experiments required for BENH Thesis	Assessment Date:	12/10/2022
Workplace (Division/Faculty/Section):	Engineering Division	Review Date:(5 Years Max)	12/10/2023
Context			
Description:			
What is the task/event/purchase/project/procedure?	Construction and Implementation of parts for thesis proposal		
Why is it being conducted?	To ensure the proper functioning of componenets aquired for physical implementation of thesis project		
Where is it being conducted?	In the MakerSpace Collaboration Space R104, Outside		
Course code (if applicable)	ENG4111/ENG4112	Chemical name (if applicable)	
What other nominal conditions?			
Personnel involved	Simeon Kelly [redacted] Catherine Hills (Supervisor)		
Equipment	Multimetre, Magnifying Glass, Voltage Source, Breadboard, Bicycle,		

Environment	USQ Toowoomba Campus, Off campus
Other	
Briefly explain the procedure/process	The parts ordered for the project must be tested and have their operation verified. This involed assembling a test circuit of a breadboard and testing all of the functions of the device.
Assessment Team - who is conducting the assessment?	
Assessor(s)	Simeon Kelly [redacted]
Others consulted:	

Eg 1. Enter Consequence

Eg 2. Enter Probability

Eg 3. Find Action

	Consequence				
Probability	**Insignificant** No Injury 0-$5K	**Minor** First Aid $5K-$50K	**Moderate** Med Treatment $50K-$100K	**Major** Serious Injuries $100K-$250K	**Catastrophic** Death More than $250K
Almost Certain 1 in 2	M	H	E	E	E
Likely 1 in 100	M	H	H	E	E
Possible 1 in 1000	L	M	H	H	H
Unlikely 1 in 10 000	L	L	M	M	M
Rare 1 in 1 000 000	L	L	L	L	L

Recommended Action Guide
E=Extreme Risk – Task ***MUST NOT*** proceed
H=High Risk – Special Procedures Required (See USQSafe)
M=Moderate Risk – Risk Management Plan/Work Method Statement Required
L=Low Risk – Use Routine Procedures

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard without existing controls in place?	***Consequence:*** What is the harm that can be caused by the hazard without existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability = Risk Level			***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
				Probability	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Testing of components for experiments	Explosion of components/overheating of components leading to burning or inhaling toxic fumes	Moderate	PPE including Safety Glasses, protective clothing and gloves.	Rare	Low	Yes		Moderate	Rare	Low	Yes
Use of Soldering iron to construct project	Severe burns from soldering tip, Electrical shock from accidental melting of insulation surrounding wire, Inhaling fumes from soldering work	Major	PPe Including Safety glasses and boots. Holder for soldering iron to sit on when hot. Extraction fan to remove toxic fumes from workspace.	Rare	Low	No	Clearing of the workspace of any cords before working , adding grounding mat to workspace.	Major	Rare	Low	Yes
Use of screwdriver	Puncturing of skin, pinching, blunt trauma.	Moderate	PPE including safety glasses and gloves.	Rare	Low	Yes		Moderate	Rare	Low	Yes
Use of Hammer to set up for experiments	Blunt trauma to hands or feet, injury via fragments of flying debris from object hit	Moderate	PPE including safety glasses, boots, long pants and gloves	Unlikely	Moderate	Yes		Moderate	Unlikely	Moderate	Yes

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard without existing controls in place?	***Consequence:*** What is the harm that can be caused by the hazard without existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability = Risk Level			***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
				Probability	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Use of drill press to construct box for final experiment	Risk of tangling clothes in machine, being hit by objects jammed in drill bit, being cut by sharp edges left around holes and by shavings	Major	Use of PPE including safety glasses, boots, long pants and shirt, gloves. Use of belt and buttons to keep clothing close to body. Use of vice to hold object while drilling. Use of Vaccuum cleaner to clean up shavings after finishing.	Unlikely	Moderate	Yes		Major	Unlikely	Moderate	Yes
Use of bicycle to test sensors in all experiments	Risk of crashing into setup sensors, injuring bystanders or damaging equipment	Minor	Use of PPE including heltmet and enclosed shoes. Clearing of onlookers. Ensuring test equipment is clearly visible and not in dangerous location	Unlikely	Low	Yes		Minor	Unlikely	Low	Yes
Use of ladder to install final project near trail	Injury from falling, pinching finger in ladder	Moderate	Use of SOP for ladder including ensuring a second person holds onto the ladder, the operator is wearing PPE including a helment and enclosed shoes.	Unlikely	Moderate	Yes		Moderate	Unlikely	Moderate	Yes

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard without existing controls in place?	***Consequence:*** What is the harm that can be caused by the hazard without existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability = Risk Level			***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
				Probability	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
			Wear gloves while carrying the ladder.								
Manual carrying of ladder	Injury due to incorrect lifting or handling techiques, or tripping while handling the ladder	Moderate	Trolley or multiple people to be used as required.	Rare	Low	Yes		Moderate	Rare	Low	Yes
Extreme weather while running experiments	Injury due to increased chance of slipping, Illness from overexposure to exreme heat of cold.	Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
***Hazards*:** From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard without existing controls in place?	***Consequence:*** What is the harm that can be caused by the hazard without existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability = Risk Level			***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
				Probability	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 5 - Action Plan (for controls not already in place)			
Additional controls:	*Resources:*	*Persons responsible:*	*Proposed implementation date:*
Clearing of the workspace of any cords before working , adding grounding mat to workspace.	Grounding Mat, additional space on workstation.	Simeon Kelly	12/02/2023
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.

Step 6 - Approval			
Drafter's name:		Draft date:	Click here to enter a date.
Drafter's comments:			
Approver's name:		Approver's title/position:	
Approver's comments:			

I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.			
Approver's signature:		Approval date:	Click here to enter a date.