Electronics and Computer Science
Faculty of Engineering and Physical Sciences
University of Southampton

Matthew Carmichael
Pragash Balachandran
Thomas Whyte-Venables
William Sparrow
Zhe Koh
15th January 2021

Low Cost Level Sensing for Domestic LPG Canisters.

Project Supervisor: Professor Steve Gunn
Second Examiner: Dr Yoshishige Tsuchiya

A Group Design Project Report submitted for the Award of
MEng Electrical Engineering
MEng Electronic Engineering
MEng Electronic Engineering
MEng Electrical and Electronic Engineering
MEng Electrical and Electronic Engineering

Abstract

This project aimed to design a bench-prototype of a relatively low-cost LPG level sensor. The purpose of this project was to aid LPG rental companies in reducing the number of customers calling when needing to replace their LPG cylinders.
The end product consisted of two main components; an internal level sensing module, and an external communications module. The internal module was a battery, and Bluetooth® microcontroller housed within an in-house made float sensor.
The external system was a strip board with all the modules required for communication, including Bluetooth® and LoRaWAN. In order to achieve communication between the internal and external systems, Bluetooth Low Energy was implemented. A satellite navigation device was used externally to obtain the location of the LPG cylinder. A LoRaWAN module was used to upload all gathered information to a cloud database.
This project tested and demonstrated a bench prototype of a fully working system, which was within budget. This report discusses all research and testing, and evaluates the results demonstrating the success of the project. This report also discusses further work required to progress to the next stage of development. Project management techniques used are also discussed, including the mitigation of the effects of COVID-19.

Contents

List of Figures

List of Tables

Acknowledgements

The team wish to thank Steve Braithwaite from ASH Wireless for his committment, guidance, and crucial knowledge which helped guide the team throughout the project. Without the support from Steve Braithwaite, this project would not have been as successful or enjoyable as it was.

The team would also like to thank 'V' Whittlesley for providing support for many students during the COVID-19 pandemic by keeping Level 2 of Building 16 open as long as possible, which was instrumental to the success of the team. We would also like to thank Jeff Hooker and Simon White for their dedication to helping many students during the pandemic, by procuring technical equipment as well as offering their technical expertise. This allowed our product to be tested thoroughly.

We would also like to thank Professor Steve Gunn and Dr Yoshishige Tsuchiya who provided valuable advice and feedback which was critical to the success of our project.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have used and modified code available on learn.adafruit.com and infocenter.nordicsemi.com

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

Chapter 1

Introduction

1.1. Problem Statement

The problem that this project aims to solve is the design and implementation of a low-cost Liquid Petroleum Gas (LPG) level sensor on consumer cylinders. The project seeks to aid rental companies with the process of recovering and exchanging LPG cylinders with their customers.
This project is a design exercise set by ASH Wireless to understand the feasibility of implementing a level sensor to help optimise the exchange process between the cylinder rental company and their customers. This design exercise was created after a company approached ASH Wireless with a similar idea, but ultimately chose a different solution.
This LPG level sensor must be battery operated and must last between 6 weeks and 6 months as part of the specification. The data from the cylinders must be available via a wireless link to a web or mobile application. The product follows the project specification as seen in Appendix A.1.
This project aims to design and build a bench prototype which will be able to operate under circumstances similar to those which the end product would see in a real-world scenario. Research into currently available level sensing technologies and the down selection process have been documented within this report. After performing this research the team was tasked with applying the level sensing technology to a design, and building a functional bench prototype as a proof of concept. The finished project was presented to Steve Braithwaite as the contact for ASH Wireless.

1.2. Problem Solution

The product created for this project was a fuel level sensor for measuring the amount of LPG stored inside a domestic LPG cylinder. It allows for remote level monitoring of the remaining fuel in rental cylinders over an area of 1200 square kilometres (km^2). The implementation of this system is in the form of a float level sensor sealed inside

Matthew Carmichael - 1.1, 1.2
Thomas Whyte-Venables - 1.2

the cylinder. It has a binary state of empty or full allowing for simple low-power electronics to be installed internally in a "sealed for life" configuration. The sensor reads the state of the fill level and sends the information via Bluetooth from the inside of the cylinder to an external mainboard. This mainboard then retrieves the cylinder's location and packages the information such that it can be sent over the Long Range Wide Area Network (LoRaWAN) and processed by an external cloud-based database.

1.2.1. Internal Operation

The float sensor is split into two modules; internal, and external. The internal module is mounted inside the cylinder where the float mechanism is situated. This mechanism fits through the standard cylinder aperture hole which has a diameter of 20.6mm. This mitigates the cost of additional machining of the cylinder and in turn allows for an "off the shelf" canister to be used.
The operational principle of the internal module is that when the cylinder is full, the float material will provide enough buoyancy to actuate a switch through a simple lever mechanism. This switch will be held down for the majority of the time as the level of the LPG will be high enough to cause the float to keep exerting pressure on the switch. Once the level of LPG reaches a threshold of 15cm above the bottom of the cylinder, the force produced by the buoyancy of the float will not be enough to actuate the switch any longer. Hence the resulting output of the fill level switch will be binary.
In order to communicate with the external module, a short-range Bluetooth® connection was used to connect the internal and external modules together. The microcontroller used for the internal and external boards, the Nordic Semiconductor® nRF52840, has integrated Bluetooth® connectivity which means that additional Bluetooth® transceivers were not required. Once per day, the external module connects to the internal module and is passed the current state of the switch, it then disconnects from the internal module. Once the external module has disconnected, the internal module enters a sleep mode to save power. After 22 hours have passed, the internal module wakes up and advertises itself via Bluetooth® for 25 milliseconds every 5 seconds. This is in order to mitigate the potential drift of the real-time clock in the microcontroller, caused by temperature differentials.

1.2.2. External Operation

The external mainboard pictured below in Figure 1.1 has three separate modules which together deliver the switch state of the float, and the location of the cylinder, through a local gateway to a web application.
From Figure 1.1, the module on the far left is the Grove - GPS (Air530) which provides the location information of the cylinder in the form of a longitude and latitude co-ordinate. The central module is the nRF52840 microcontroller which has built-in Bluetooth® connectivity to communicate with the internal module. The right-hand module handles interfacing with the LoRaWAN which is the method of sending data to the cloud.

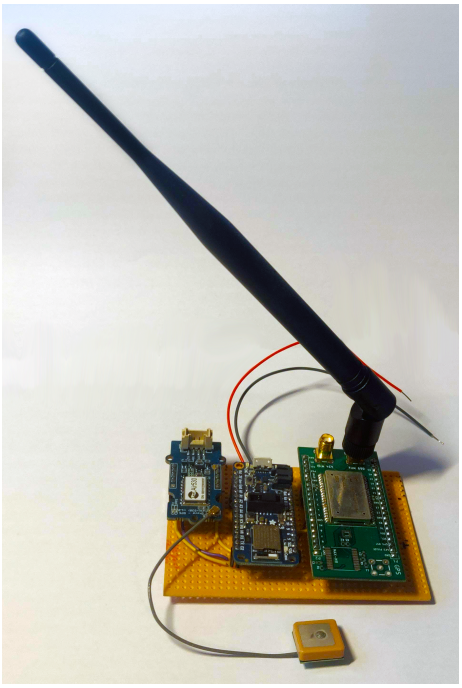


Figure 1.1: A photo of the external system soldered to stripboard

When the microcontroller wakes up from sleep after 24 hours; the Air530, Bluetooth®, and LoRa modules are initialised. The Bluetooth® will scan until it connects to the internal module, which advertises every 5 seconds. Once the Bluetooth® is connected, the switch state is retrieved and the connection to the internal module is disconnected, in order to save power. During this time the Air530 is configured, and the location co-ordinate is acquired. The switch state and coordinate are then sent over LoRaWAN to the nearest gateway, and are then uploaded to the web application. The LoRaWAN is used in conjunction with The Things Network (TTN) and the web application Ubidots to present the data in an easily interpretable format as seen below in Figure 4.3.

Chapter 2

Background

2.1. Why Should the Product Be Created?

2.1.1. Societal

The LPG level sensor aids in the widespread rental of domestic LPG cylinders in deprived areas where residents are still reliant on cheaper solid and liquid fuels, such as wood and kerosene, for cooking. These fuels generate large amounts of particulates which can affect people's health negatively. [1] It has been found that wood has the highest carbon dioxide emission when a natural draft is used (which is most common amongst poorer communities) followed by wood with a forced draft, then kerosene burning. [2] With the lowest carbon emissions being associated with the burning of LPG. LPG was also found to have the highest thermal efficiency out of these four methods. [2] These negative health effects can decrease people's lifespans and cause health problems such as respiratory conditions and cardiovascular problems. [3]
One major impact of being reliant on wood and other solid fuels is that they must be collected, which can lead to many hours of the day being taken up retrieving these fuels. [3] If LPG is available to these families then hours of the day can be freed up by the LPG being delivered by the rental company, which in turn can give a greater quality of life.

This project aims to create a low-cost method of sensing the level inside an LPG cylinder. This can help in reducing the possibility of families running out of LPG. Traditionally, the rental company would be contacted by the customer once the LPG cylinder had run out completely and would end up leaving the customer without any fuel to cook with. This would majorly impact those in developing countries who would then need to spend time retrieving an alternative fuel which may not be as efficient or clean burning [3]. With the introduction of the level sensor, the rental company would have an early warning to know when the level inside the cylinder is low. With this system of LPG level sensing in place, the customer is no longer

Matthew Carmichael - 2.1,2.2
Pragash Balachandran - 2.2
Thomas Whyte-Venables - 2.2, 2.4
William Sparrow - 2.2, 2.3

required to contact the rental company and the rental company can send a fully filled cylinder out to the customer before the customer uses all of the LPG left in the cylinder. This will improve the quality of life of the LPG users in developing countries as alternative fuels will no longer be required for cooking.

2.1.2. Environmental

LPG is a clean burning fuel which means that it has negligible particulates when compared to other dirtier fuels such as wood and coal. [4] This makes LPG's carbon footprint lower when compared to these other fuels which create particulates in the air.
There is also a global abundance of LPG, despite it be non-renewable, as it is a by-product of oil refining and hence should be used instead of being disposed of as it increases the efficiency of the extraction process. [4] Due to this global abundance and its highly efficient complete combustion, LPG is one of the best contenders to be used for household cooking in developing countries. [4] A higher demand for LPG will decrease the dependency on other solid fuels such as wood and coal.
The LPG level sensor produced in this project will help decrease the carbon emissions caused by the delivery vehicles. The location of the LPG cylinder will be sent with the fill level to enable the optimisation of journey planning and therefore the shortest feasible path can be used to deliver and pick up the LPG cylinders. This will help to decrease the emissions associated with logistics.

2.1.3. Economic

The economic impact of this project is primarily directed towards the rental companies that would implement these level sensors inside their LPG cylinders. The introduction of the level sensor will eliminate the need for the customer to report that their cylinders are running low on LPG, as it will be done automatically by the system. This will help the customer, as time does not have to be used calling the rental company. This means that received customer enquiries regarding changing their LPG cylinder would now be reduced in number, hereby reducing the need for call centres. This can reduce overhead costs such as the rental of office space, the salaries of call centre employees, and equipment costs. This will lead to savings for the rental companies which could increase profit margins.

2.2. Existing Products

There are many LPG level sensing products on the market available for consumers. They fall into two categories, continuous level sensing and point level sensing methods. For continuous level sensing, the level of the liquid is monitored and updated at set time intervals to continuously track the level of the liquid. This can be used for process management. Point level sensors have the ability to detect when the liquid reaches a pre-determined specified level and can be used for overflow control and empty cylinder prevention. Continuous level sensors can also be used to implement these measures, depending on how the system is configured. A sensor is either

invasive or non-invasive. Invasive methods have the sensor making direct contact with the liquid being measured, whereas non-invasive methods do not. The target market for these existing products are the owners of the LPG cylinders. Some examples of available products are listed below.

2.2.1. Invasive

The 6200 Series from Rochester Gauges [5] is a float sensor which can also be used to measure the level. This is an invasive implementation as the float will rest on the LPG itself, the movement of the float is then translated through gears and a magnetic coupling to allow the fill level to be shown on the dial. The product is mounted using a threaded hole that is drilled into the cylinder, and provides a continuous reading. However, the dimensions and design of the product mean that it cannot be implemented with all cylinders as it requires a wide interior diameter to allow the float to rotate freely around its axis. This presents for domestic cylinders as they are typically tall and thin.

2.2.2. Non-Invasive

The Mopeka Bluetooth® Level Sensor kit is a non-invasive sensor that attaches to the bottom of the cylinder and uses ultrasonic waves to measure the LPG level. [6] It also implements Bluetooth® communication to send the fill state to the user's phone. This product is a continuous level sensing method, the LPG level is updated in real time. It can measure the level from when the cylinder is full, until it is completely empty. The device measures the time taken for ultrasonic waves to propagate from the device, reflect off the surface of the LPG, and be received by the sensor.
There was also an attempt to create a magnetic level sensor that measured the thermal junction found at the boundary between liquid and gas, therefore identifying the level of the LPG. [7] Once the level had been identified, the strip would transmit the data to the user's phone via Bluetooth®. Unfortunately, the Kickstarter project for the device issued refunds in April 2018, citing manufacturing issues.

2.3. Internet of Things (IoT)

The Internet of Things (IoT) is a term used to describe the network of physical devices (Things) that are connected to the Internet. These devices share data and sensor readings with each other and central databases. [8] Any sensor that sends its data to a cloud system is classed as an IoT device. Since our project sends LPG level data to the cloud to be monitored at a central database, it is deemed to be an IoT device.
The number of IoT devices increased 31% year-on-year to 8.4 billion in the year 2017. [9] The ability to remotely monitor aspects of a project decreases the need for on-site engineers, and increases the number of diagnostic tools available during a repair. As sensors in devices become more and more prevalent, the devices containing the sensors become increasingly smaller as manufacturers look to find anything that might benefit from remote monitoring. With smaller devices come smaller power supplies, and with more remote locations comes less widespread internet connections.

Therefore, other wireless technologies must be considered, with particular emphasis placed on low-power consumption.

2.3.1. Long-range wireless

2.3.1.1. Low-Power Wide Area Networking (LPWAN)

Low-Power Wide Area Networking (LPWAN) is a catch-all term for wireless networks designed to allow long-range communication with a low data rate, allowing for a reduction in power and cost for transmission. There are many different LPWAN technologies and protocols, with LoRaWAN and SigFox being the two main competitors. [10] [11] LoRaWAN looks at a wider frequency spectrum than SigFox, and as a result tends to get more interference. However, since the information that a receiver is looking for is so specific, the coding benefits outweigh the noise found on the signal. At scale, practical set-up costs are around the same for both LoRaWAN and SigFox but unlike SigFox, the endpoint and base station of a LoRa network are relatively inexpensive. However, these costs become irrelevant at the industrial scale. It is generally accepted that LoRaWAN is more user-friendly, for whereas SigFox does not have worldwide coverage, a LoRaWAN network can be set up by anyone for only a few hundred pounds, which allows coverage to be established anywhere on the planet. [12]

2.3.1.2. Very Small Aperture Terminal (VSAT)

Very Small Aperture Terminal (VSAT) is a two-way satellite communication system that uses dish antennae pointed at access satellites in geostationary orbits. This is more useful for high data rate transmission than LoRaWAN, and while not specifically used for sensor data transmission their use in connecting devices over long ranges means they could technically be used for our purposes.

2.3.2. Short-range wireless

2.3.2.1. Bluetooth® Low Energy (BLE)

Bluetooth® Low Energy (BLE) is a subsection of Bluetooth® technology, used in applications such as healthcare and fitness, beacons, and security. For versions since Bluetooth® 4.0, both BLE and standard Bluetooth® protocols can be supported by one device. Using mesh profiles, BLE can be used by devices to communicate with each other. In a mesh profile, each device can pass information to other BLE devices, allowing for grouped actions such as lighting control of an entire building. [13] The range of BLE is theoretically up to 100m, but in practice, this is reduced by external factors such as walls and electrical interference to around 10m. [14] With a maximum throughput of 1.37Mbits per second the BLE protocol can send plenty of data over a connection.

2.3.2.2. Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) uses electromagnetic fields to track tags attached to objects. When triggered by an electromagnetic interrogation pulse, a small radio transponder transmits data back to the RFID reader. RFID is superior to barcodes as the tag does not need to be in line of sight of the reader, so it can

be embedded in an object. This reduces the chance of external damage rendering it unreadable, as a damaged barcode can not be read by a scanner. RFID could be useful for this project as a means of identifying cylinders.

2.3.2.3. Near-Field Communication (NFC)

Near Field Communication (NFC) is a short range communication protocol designed to allow electronic identification between two devices. It is most commonly found in contactless payment systems, and has a range of around 10cm. [15] NFC could be useful for this project as a means of taking payment for LPG gas refills, and as a means of identifying cylinders.

2.4. LPG Containers

LPG is a colourless gas often used in cooking, heating, and crop-drying. It is often supplied in two main forms: propane (C_3H_8) or butane (C_4H_{10}). [16] For the gas to become liquified it has to be pressurised to its vapour pressure. This typically occurs at a pressure of 220kPa at 20°C for pure butane and 900kPa at 20°C for pure propane. [17]
For the domestic storage of LPG the two main methods of storage are canisters and cylinders. 'Canister' typically refers to the small portable bottles used by those who engage in camping or long distance walking activities as they are lightweight and compact. For cylinders the two main options easily available are either made from steel or composite materials.

2.4.1. Steel

Steel cylinders are typically made of two discs of steel which are stamped into the shape of a cylinder. One half has a hole punched into the top into which the valve flange is inserted and welded in place. The ring used for standing the cylinder upright is also welded to the bottom half of the cylinder. These two halves are then welded together to create a sealed cylinder.
These cylinders are typically heavy with a 13Kg cylinder of propane having a gross weight of 25Kg. [18] Due to being made of steel they are also susceptible to the environment with corrosion being a major problem. Whilst often coated to prevent rust, over time this coating is often knocked and scraped off leaving the bare steel to corrode.
The major advantage to these cylinders is that of price, as they are very cheap to produce in mass quantities. This is because they involve basic metal fabrication procedures which can be performed almost entirely autonomously. So, for purposes which are mostly static and sheltered, the money saved in the price of the cylinder is often enough to offset the disadvantages.

2.4.2. Composite

Composite cylinders are typically made using a three part design. The inner part is a High Density Polyethylene (HDPE) liner which is blow moulded to create the cylinder pressure vessel shape. This then has an extremely long strand of glass fibre -coated in resin- wound around it multiple times. This layer gives strength to the

internal liner. The third part is another layer of HDPE which is moulded to create a protective case. It also serves to act as a stand to keep the cylinder upright. The valve threading is cast into the HDPE liner, allowing for different valve types to be set into the cylinder. [19]
The benefits of composite cylinders are; their resistance to corrosion, higher mechanical resistance, and reduced weight in comparison to steel. [20]
Composite cylinders do not corrode as none of the components react with the air and have high chemical resistance. This means that their structural integrity won't suffer if they come into contact with harmful chemicals. Their lower weight means that for the same weight load lorry, more individual cylinders can be taken, which increases the efficiency of the logistics. They also have a higher mechanical burst pressure than steel which means they are more resistant to failures caused by age. [20]
The main disadvantage to composite cylinders is their high cost in manufacture. This cost can be offset by the long lifetime of the cylinders, as well as costs saved in logistics for transporting and retrieving from customers.

2.4.3. Valves

As a method of interfacing with equipment for various different applications there exists a large array of standard valves. In each area of the world a different standard is prevalent. For Europe and the United Kingdom the standard most commonly used on domestic LPG cylinders is the POL 105 thread. [21] This is equivalent to a 5/8" British Standard Pipe thread which has a nominal diameter of 22.9mm. The thread has a symmetrical "V" shaped thread with rounded crests to ensure a maximum possible mating surface between the male and female components. This therefore means that the minor diameter of this size of thread is 20.587mm.

Chapter 3

Sensors

3.1. Sensor Research

As discussed in Section 2.2 there are many different level sensing technologies so it was decided that a large part of the project should be dedicated to researching each type of sensor.
After extensive research, the level sensing technologies which we explored for this project are as follows; Tuning fork, Capacitance, Laser, Float, Microwave, Pressure, Ultrasonic, Optical prism, Weight, and Piezo-resonant.

3.1.1. Tuning Fork

One method of invasive level sensing makes use of a tuning fork's resonant properties. A piezo-electric crystal acting as an actuator vibrates the fork at its natural frequency. When the material to be measured is between the prongs, these vibrations are dampened. This change of vibration is then detected by a second piezo-electric crystal acting as a sensor which then trips a switch. This 'present or absent' measuring system means that a tuning fork is a binary level sensor.
There are several advantages to a tuning fork level sensor, chiefly its ability to not be affected by turbulence, so long as the LPG is still in between the forks, the switch will not trip while the cylinder is in transit. With no mechanical moving parts, the maintenance of the sensor will be minimal. The sensor will also not be affected by being inside a pressurised vessel. [22]
The fact that the sensor can be used in pressurised cylinders is also a disadvantage in this instance. In order to mount the sensor, a secondary hole must be made in the cylinder wall, requiring custom design and pressure testing. The power requirement is a minimum of 12V, which is too high when aiming for a low-power device. [23] This is a disadvantage of this type of sensor. Another disadvantage for this type of the sensor is the price, which can be over £200 for a small sensor. [24]

Matthew Carmichael - 3.1.3, 3.1.4, 3.2, 3.3
Pragash Balachandran - 3.1.5, Original Author of 3.1.6
William Sparrow - 3.1.1, 3.1.2, 3.1.5, Main Editor of 3.1.6
Zhe Koh - 3.1.7, 3.1.8, 3.1.9, 3.1.10

3.1.2. Capacitance

A capacitive level sensor can be both invasive and non-invasive. When placed inside the container, a parallel plate capacitor is in contact with the liquid. As the level drops such that the liquid between the plates is replaced with air, the capacitance of the capacitor changes. This is due to the dielectric constant of air being different to that of the measured medium. The change in capacitance can be measured by applying a voltage across the capacitor. Typically this change in capacitance is gradual as the parallel plates extend the entire height of the cylinder. This means that the sensor is continuous in its measurements, as opposed to discrete. However, it would be possible to implement a small point level sensor using this method by simply shortening the length of the parallel plates.
The advantage of the invasive sensor is that there are no moving parts to require maintenance, and there is no effect of pressure or temperature on the sensor since no sensitive components are in contact with the sensor. The invasive sensor again presents the issue of custom machining costs, as well as one unit costing in excess of £140. [25] The minimum voltage required for use is 12V, this, along with the other disadvantages, prevents its implementation in this project.
For plastic-walled composite cylinders there exists an externally mountable sensor that can sense when there is a level change inside the cylinder. This works on the same principle as the invasive sensor, which is why it only works with plastic tank walls, as metal would interfere with the tank too much. The main drawbacks of this type of sensor are its cost, around £180 each, and the extensive calibration required to be accurate. [26] There is also a risk of the sensor being damaged, as it sits externally on the cylinder which by its very design will be in a harsh environment.

3.1.3. Laser

Laser level sensing is one of the more common approaches to measuring liquids in a pressure vessel. A laser is an invasive method which measures the time of flight of a laser pulse reflected off the surface of the liquid and back to a photoreceiver. The time it takes for the photoreceiver to detect a laser pulse can be used to determine the distance from the top of the pressure vessel to the surface of the liquid. This is true as long as the velocity of the speed of light is known through the gas inside the vessel.
The advantages of this sensor are that they are very precise and accurate. This sensor also has very simple inputs and outputs which can be easily interfaced with microcontrollers. Being chemically and temperature resistant from 25°C up to 80°C are also benefits of using this type of sensor. These sensors can be used both continuously and discretely which means that their use would be applicable to this project. The cost of such sensors is within the range of £35-£40 which is feasible and reasonably low-priced. [27]
The disadvantages of using lasers are that they are invasive which is not ideal for our purposes. The laser requires a centralised hole to allow an accurate distance between roof and liquid level to be determined even in the case that the cylinder is not located on level ground. This is very difficult without making changes to the valve itself which would incur high costs, due to the fact that domestic cylinders have hemi-spherical tops. In addition to this there are also power concerns as it

will use 2.5-10mA with a transient peak of up to 100mA which could be an issue for minimising power consumption. [28] Furthermore, the voltages used are usually 4.5V or greater, there are some that are feasible to use at 3.3V, but the availability of these devices and their cost is unknown. [29]

3.1.4. Float

A float level sensor works on the principle of buoyancy, where a material of lighter density than the liquid inside the pressure vessel will rise to the liquid level. This principle can be employed in various configurations, both point level, and continuous. The first is to have a set minimum value which will have a float pressing against a limit switch until the liquid level decreases such that the float can no longer provide enough force to hold the switch down, resulting in a binary value. On the other hand, a maximum value can be set where the float must rise with an increasing fill level until the maximum is reached. At this point the float closes a circuit which provides a binary signal which can then be read by a local microcontroller. A compromise between the two is also available where the float is used to send a continuous value of fill level. This method uses the track that the float runs along to create a change in impedance which can then be detected and used to ascertain the position of the float along this track. Once the position of the float is known, it is then simple to calculate the level of the LPG. Another option for float-based level sensing is the method discussed in Section 2.2.1. Whereby, a counterbalanced arm rests on the surface of the liquid, and the rotation of the arm is translated via a system of gears, and magnetic couplings to a linear level value.
Floats are very invasive and will be fully submerged within the liquid. Therefore, most commercially available floats are chemically resistant and can be very long lasting. Floats can orientate both horizontally and vertically which gives a choice of mounting position and are therefore more flexible in their design.
Many commercially available float sensors are designed for large pressure vessels which would not be applicable to this design. The commercial float sensors are sized for large industrial LPG containers, which are often static and are therefore not applicable to this project. The sensors designed for these large containers are often not space efficient, which poses a problem for small domestic cylinders. The power requirements for these commercial float sensors are quite high with a minimum voltage of 24V being required to operate them. This is high for the level sensor design, and, in combination with the fact that they are also expensive, makes commerical float sensors undesirable. [30]

3.1.5. Microwave

This method is based on the reflection of an electromagnetic pulse. There are several methods of using microwaves, including both invasive and non-invasive systems. In an invasive system, the pulse is emitted in the direction of a surface, and then is reflected by the measurand. The echo time of the pulse is proportional to the distance of the sensor from the surface of the liquid. The time of flight is then used to determine the liquid level by calculating the distance of the liquid from the sensor, using a known sensor height. Due to the nature of microwave divergence, the sensor must be close to the surface of the liquid, or the returning wave will be

heavily affected by noise. The main advantage of using microwaves is that due to the nature of the waves, the accuracy of the sensor is independent of external factors such as pressure, temperature, and dust. [31] The time of flight method of measuring produces a continuous level reading inside a defined range, but outside of this range it is effectively a point level measurement.
Microwaves can also be used in a non-invasive manner as a point level switch. The transmitter and receiver are adhered to either side, on the outside of the container. The presence of a material in the path of the microwave will dampen the received signal and indicate whether or not there is a liquid is present at a certain predetermined level.
In addition to the health issues surrounding microwaves, the power requirements in sending the wave, and the processing of the returned signal places this option out of the scope of the project. There is also a high cost consideration for an off the shelf design. [32]

3.1.6. Pressure

The pressure sensor is placed inside the liquid to be measured. The transmitter has a pressure diaphragm, the inside of which is kept at atmospheric pressure. The other side of the diaphragm is in contact with the liquid and the pressure exerted on the diaphragm, represented as strain on the diaphragm, can be used to calculate the weight of liquid in the container. From this, simple maths can be used to calculate volume of liquid left in the container.
The obvious flaw with using a pressure sensor in this design, is that LPG is pressurised and would therefore affect the reading. However, due to LPG vaporising below a certain pressure, the gas pressure inside the cylinder would remain constant. With a constant force applied to the surface of the LPG, an offset can simply be removed from the sensor reading to obtain the pressure exerted by the mass of the LPG.
The other main issue surrounding this sensor is the cost, with some models costing over £400. [33] We would be unable to manufacture the sensor, to circumvent this issue of cost, ourselves as the pressure sensor requires custom micromachining.

3.1.7. Ultrasonic

This type of sensor uses ultrasonic echo sounding to measure the liquid level. The sensor is usually mounted on top of the cylinder which means no custom-made cylinder is required for this type of sensor. The sensor will emit an ultrasound wave into the cylinder which will be reflected upon reaching the surface of the liquid. The liquid level is then computed by measuring the time taken for the reflected wave to be received. This is then converted into an electrical output signal. As the sensor uses a time of flight method, continuous liquid level can be measured. [34]
One of the advantages of this type of sensor is that it is non-invasive because it is mounted on the top of the cylinder as mentioned above. This type of sensor also works on both conductive and non-conductive liquid. Since the pulse wave emitted is at high frequency this type of sensor has high accuracy and its performance is not affected by a sealed pressurised vessel.
The major disadvantage of this type of sensor is that they are relatively high in cost compared to other types of sensor. Moreover, since the sensor is top-mounted, it

only works when the cylinder is in an upright position. This type of sensor also does not work in a vacuum as the emitted ultrasound cannot propagate through a vacuum. In addition, more signal processing is required in comparison with other types of sensor.

3.1.8. Optical Prism

This type of sensor operates using Snell's Law. Snell's Law is used to predict the behaviour of a light beam refracting from one substance to another. The difference in refractive index between different mediums causes different degrees of refraction. This can be detected by an optical sensor. [35] All transmitted light is reflected back to the receiver when the liquid is not in contact with the sensor. However, when the sensor is immersed in the liquid, the light disperses into the liquid, and less light is returned to the receiver. The amount of light returned to the receiver affects the output signal strength, hence point level sensing is possible.
This type of sensor is compact and small. This makes them able to measure the level even if there is only small amount of liquid. This type of sensor needs to be pressure and temperature resistant as the sensor needs to be submerged in the liquid to detect the difference in the refractive index. [36]
The major disadvantage of this type of sensor is that they are invasive due to the required contact with the liquid. [36]

3.1.9. Weight

Instead of using off the shelf level sensors, this technique uses a strain gauge. It is based on the concept that given the respective weights of an empty cylinder and a fully filled cylinder, the amount of liquid in the cylinder can be estimated by measuring the current weight of the cylinder.
The advantage of using this technique is that it is relatively low cost compared to other sensing methods. The disadvantages of this method include the fact that this method requires extensive and constant calibration to obtain an accurate result and yet even after calibration, it is still not as accurate as other methods. Factors such as sensor drift and uneven ground surfaces affect the accuracy of this measurement system.
Weighing the cylinder is the recommended method of detecting the amount of LPG left in a cylinder by most manufacturers. [18] However, this method is not suitable for our project, as discussed in Section 3.2.

3.1.10. Piezo-Resonant

This type of sensor creates an acoustic signal when it is excited which is a function of the natural resonance of the material. The generated signal is directed through the wall of the cylinder, and the sensor detects the reflected pulse to determine whether there is air or liquid on the opposite side of the cylinder wall.
The biggest advantage of piezo-resonant sensors is that they are non-invasive as they are to be used outside of the cylinder. Having a high accuracy is also an advantage to this type of sensors. The major disadvantage is that this type of sensor only works with plastic cylinders. This means that this type of sensors is unable to measure the liquid level in a steel cylinder. Moreover, the thickness of the wall also affects the

reading of this type of sensors as they only work on cylinders which the walls are not thicker than 6.5mm. Temperature is another issue with this type of sensors as they cannot be used above 70°C. [37]
There is another method of level sensing using piezo-resonance. By exciting to the cylinder and detecting its resonant frequency ,which changes with fill level, the amount of liquid left in the cylinder can be determined. This uses the principle that resonant frequency is equal to $2/\lambda$, where λ is the distance between the surface of the liquid and the top of the cylinder.

3.2. Sensor Down Selection

A summarised specification of each type of sensors are as shown in Table I:

Table I: Table of a summary of each sensor type

Type of Sensors	Type of Detection	Invasive?	Relative Cost	Requires Custom Valves?
Tuning Fork	Point Level	Yes	High	No
Capacitance	Continuous Level	No	High	No
Laser	Continuous Level	Yes	Low	Yes
Float	Point Level	Yes	High (if premade)	No
Microwave	Continuous Level	No	High	No
Pressure	Continuous Level	Yes	High	No
Ultrasonic	Continuous Level	No	High	No
Optical Prism	Point Level	Yes	Low	Yes
Weight	Continuous Level	No	Low	No
Piezo-Resonant	Point Level	No	Low	No

The sensors that have been chosen for down selection are the; float sensors, ultrasonic sensors, and piezo-resonant sensors. Each have their own properties that would make them suitable for this project.
The sensors that have been outlined already have drawbacks such as; requiring a change in cylinder design, and a lack of compatibility between the sensor and valve location. This is why sensors such as laser and optical prism have not been chosen. The change in manufacturing would have high start-up costs and these changes would have to be compatible with both types of cylinder. Some sensors are too power-intensive and may require additional current management devices which will have a constant current draw, significantly reducing the total run-time of the sensor module. These power-intensive sensors are capacitance, pressure, and microwave which will therefore not be used. Some sensors are too expensive such as tuning fork which has been discounted for this reason. Weight sensors may be fairly low cost, however due to the changing location of the cylinder, it cannot be guaranteed that the ground will be level. These conditions are not ideal for calibration of the force sensors. This method is only effective when the sensor used to take the reading has an opportunity to zero itself before measurement. This would not be an option in this project, as the sensor will have to remain attached to the cylinder at all

times. For this reason, weight sensors cannot be used in this application. This has eliminated the majority of the sensor choices and only three choices are left: ultrasonic, piezo-resonant, and float sensors.
Ultrasonic sensors can be used as an external sensor which is not invasive. This sensor will be mounted on the bottom of the cylinder and will send an ultrasonic pulse through the cylinder base. The sensor is mounted on the bottom of the cylinder to offer protection from the elements. The pulse will pass through the cylinder base and reflect off the boundary between the LPG and the gas inside the cylinder. The time of flight of the pulse will give an indication of height between the liquid-gas boundary and the base of the cylinder. In order to make sure that the values acquired are correct, an average of 8 pulses is taken. Each pulse can only start after the last one has been received. This means that the module will need to be active for a relatively long time to collect the pulse data. On the other hand, ultrasonic sensing is very precise and therefore can be used to give an exact fill level daily. This will also allow for data to be collected about user's usage habits and can be used to anticipate cylinder refills more efficiently. This sensing method is also non-invasive which means that a custom valve is not required to be incorporated which saves production costs. In addition, the ultrasonic sensor uses piezo-electric transducers which will generate the waveforms with a low energy cost. One major downside to ultrasonic sensors is that the echoes from the generated pulses are very noisy and complex. Parsing these echoes will be extremely power-intensive in terms of processing resources. Hence, due to this reason this sensor option was not chosen.

Piezo-resonant sensors are non-invasive level sensors which uses the resonant frequency of the cylinder to determine whether there is air or LPG on the other side of the cylinder wall. This provides an accurate point level reading. One of the main downsides of this sensor is that it is only compatible with the composite cylinders and it must be preconfigured to each type of composite cylinder because of the different thicknesses of the cylinder walls. By using piezo-electric transducers the cost of manufacturing this sensor is decreased as the transducers are low cost. The piezo-electric transducers also save in power consumption as they will require little current to induce oscillation. The main drawback of this method is the power consumption of the microcontroller used to generate the driving oscillations. In addition, the processing power required to decrypt the reflected oscillations will be very high to the point where this method is no longer viable. For this reason, this sensor was not chosen.

Floats are invasive sensors which usually have high power and monetary cost when in an industrial or commercial setting. However, if it were designed and made "in-house" it would be significantly cheaper and also would eliminate the negative aspects of the commercially available float sensors. These include high power requirements as they are sized with industrial equipment in mind. A separate hole would need to be drilled to accommodate any horizontally orientated floats hence, a vertical orientation would be required to mitigate additional machining costs to the cylinder. For vertically orientated floats, the valve hole would provide enough space to insert

a vertical float sensor. This means that custom valves or cylinders are not required and production costs are saved. For this project, the minimum set value method as mentioned earlier in this chapter would be most applicable to the project. In this method, a binary value is given which can be more easily processed. This value can easily be sent wirelessly to an external module mounted on the outside of the cylinder. This could then collect any other additional data and send it over a long-range communications network. This will give a system that has two different power requirements, one for the internal, and one for the external. As a result these modules will have different, appropriately sized, batteries. This method is viable under these circumstances and hence can be taken forward for further development.

3.3. Final Implementation of Sensor

The down selected float sensor will be made in-house and will therefore have various requirements that need to be met.
The first of these requirements is that the float must fit within the 20.6mm valve hole so that it does not require a custom valve aperture. With this in mind the circuit board and battery used should also be able to fit within this 20.6mm constraint. The float will operate on the basis that while the LPG level is above a set threshold, the float will actuate a switch, holding it closed. When the liquid level decreases below the threshold level, this switch will no longer be held closed which will tie a pin on the microcontroller to ground.
The density of LPG at vapour pressure at 15°C is 0.525g/ml [38]. The float material must be able to provide enough force to hold the switch closed. There are multiple types of float that can achieve this. The first being a hollow metal ball which is filled with air. This will displace enough LPG to overcome the weight of the metal shell and cause the hollow ball to rise. The challenge with this type of float is ensuring that the hollow ball is large enough to provide sufficient force to actuate the switch, while still fitting through the valve aperture. Another issue is creating an airtight seal to prevent loss of buoyancy. An additional type of float is made of Nitrile Butadiene Rubber (NBR). NBR is a low-density rubber which has a density of 1.15 g/ml, this is coupled with the injection of pores which are filled with air to provide greater buoyancy [39]. This helps it achieve a much lower density to be able to act as a float in a range of oils and liquids.
These float materials will be used to ensure the float has enough buoyancy to hold the switch closed. The float sensor will be weighted so that it sinks to the bottom of the LPG cylinder and will not rise and cause interference with the valve, or change the threshold level. The threshold for the float sensor will be about 15cm above the bottom of the cylinder. At this level there will some LPG remaining which will allow continued demand to be made on the old LPG cylinder until a new one is delivered. The height of the float sensor above the bottom of the cylinder can be varied in manufacture to provide the optimum lag time before delivery. Once a day, the switch state is checked and the binary value is sent to the external module. Due to there being no additional manufacturing to the cylinder, the internal module must be able to communicate with the external module wirelessly. The state of the

float switch is sent over a wireless communications channel to an external module. The external module will then gather any additional information and send it over a long-range wireless network to be configured onto a web application interface.

Chapter 4

Design

4.1. Internal

4.1.1. Mechanism

As discussed in Section 3.3 the float sensor is a sealed internal system, this means that a relatively simple mechanism can be used to detect the LPG level. To make the design as simple and efficient as possible, whilst maintaining low manufacturing costs, the end choice of this design is that of a switch actuated by a float. This switch can then be connected to a microcontroller, which can read the states to be either "full" or "empty" depending on the switch state.
Given this choice of float sensor design criteria, it resulted in a number of initial design ideas as shown in Figure 4.1.

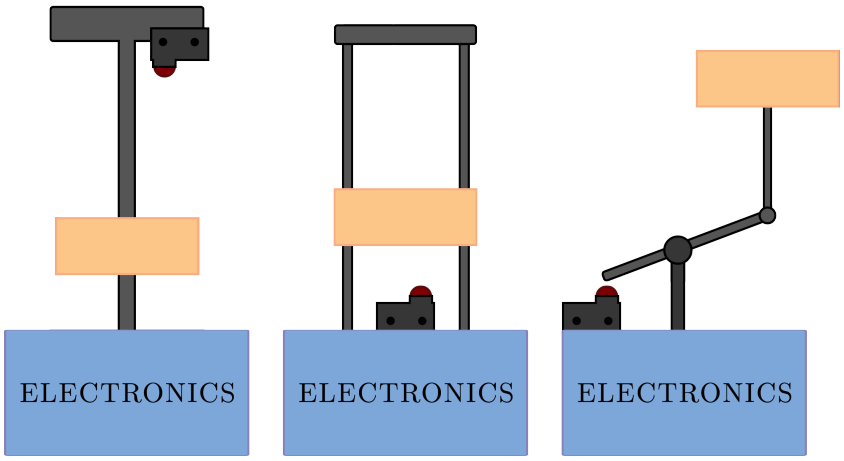


Figure 4.1: Different initial design ideas for float and switch placements.

Matthew Carmichael - 4.1.2, 4.1.3, 4.2.4
Thomas Whyte-Venables - 4.1.1
William Sparrow - 4.2.1, 4.2.2, 4.2.3, 4.3

The initial design was deemed to be too complicated to route the connection from the switch to the electronics below given the space constraints discussed in Section 3.3.
The second design is incredibly simple and cheap to implement meaning it shouldn't be subject to mechanical failure over the total lifetime of the product. However, it requires the need for NBR float material, which is easy to acquire in bulk but from University approved suppliers is only available in the form of sheet or o-rings and not the blocks required. These would be easy to buy for a large scale operation, but the only feasible way of obtaining it for the project would have been to buy an existing float sensor and dismantle it. The other issue with this design is related to the switch, as for the system to operate, a low density material of sufficient size is required, in order to have a large enough mass to operate the switch. This means either an ultra low force switch or a large block of material. Given the material supply issues discussed above, the switch was the only variable that could be changed. Finding switches in the ultra low force category proved difficult as most manufacturers stop their lower force limit at 1N. Any lower force often resulted in the use of lever actions or larger switch bodies which would not fit through the aperture of the valve.
This lead to the final design idea, which was the use of a mechanical linkage to allow the float to exert a larger force on the switch by exercising a mechanical advantage. By varying the placement of the pivot, simple mechanics could be employed to generate more force for the same size of float. Given the limitations of both manufacture and suppliers this was the design we took forward.

4.1.2. Communications

The float sensor on the inside of the cylinder must be able to communicate wirelessly with an external module. The external module then gathers additional environmental data and transmits this data, and the switch state to a web application. The wireless device required to connect the internal module to the external module would need to be short range and would need to produce a signal strong enough to pass through the cylinder wall.
A typical radio transmitter is a feasible method of communication, but it would have a much larger range than required as there is less than a metre distance between the internal and external modules. For this reason, a Bluetooth® transmitter can be used as it is specifically designed for short distance communications. [40] BLE is the one of the lowest energy consuming short range wireless communication protocols. [40] This is important as it will allow for the power consumption to be minimised for each module.
One of the initial microprocessors that was considered for this project was the EMB1061. [41] This microprocessor was initially chosen because it had integrated Bluetooth® connectivity, in addition to a small board size that would be able to fit within the valve aperture. However, once the team had purchased this microprocessor, it became apparent that it would not be able to meet the needs of this project. This is because the external module would require multiple UART connections, and this microprocessor only provided one. Furthermore, this chip had little to no support in how to program it too much time would be used learning how to configure the Bluetooth® connection for this board. Therefore this microprocessor was not used,

and additional research was undertaken to find a better suited microprocessor. [41] The chip chosen was a Nordic Semiconductor® chip with an ARM® Cortex-M4 processor, the nRF52840. This chip was chosen due to its low power support, including its low power mode which has a current consumption of 0.97μA. This particular microprocessor chip was chosen because it offered on-board BLE, as well as two UART ports which other versions did not offer. For prototyping purposes, the Adafruit Feather had been selected as the development board because it has many supported libraries in the Arduino Integrated Development Environment (IDE) as well as examples for use as a Bluetooth® device. This made the programming of the breakout boards easier.
One method of transmitting the fill level over the BLE protocol is to include the fill level in the advertisement packet. This reduces the time taken to send data, as the devices never need to connect. However, this method was not chosen because of the BLEUART protocol supported by Nordic Semiconductor® offered a much more reliable connection as an alternative. This choice made the current draw of the nRF52840 boards higher, but this increase was considered to be negligible when compared to the increase in connection stability.

4.1.3. Power

The specification for this project, as shown in Appendix A.1, dictates that this must be a battery operated sensor. It must be able to last between 6 weeks and 6 months before being taken back to the depot for replacement or recharging. The battery must therefore have enough capacity to be able to last the 6 month time period.
The power requirements of the internal battery will depend primarily on the power consumption of the microprocessor which includes the activation of BLE. By assuming that the maximum currents in each power mode are used, a worst-case scenario can be calculated in terms of power consumption. The highest current option while being in low power mode according to the nRF52840 datasheet is 17μA. [42] The current draw of the microprocessor whilst on and running code is 6.3mA and the highest reasonable transmission current is 8.6mA. [42] The power usage of the internal module per day is fixed. The internal module will be in low power mode for 22 hours of the day and then wake up from sleep to start advertising for 25 milliseconds and sleep for 5 second intervals.

Table II shows the total time throughout the day that is spent on each task for the internal module. The microcontroller sleeps for the majority of the day, and the Bluetooth® function and microcontroller will be in the scanning and normal operation modes for 45 seconds of the day. This includes every 25 millisecond instance where the internal module acts as an advertising beacon to connect to the external module. As well as an additional 5 seconds for while it is connected to the external module, until it is disconnected and called into low power mode.

Table II: Table of time spent per day for each internal component

Internal Component Task	Time /s	Current /A	Coulombs per Day /C
Microcontroller Sleep	86355.0	$1.7x10^{-05}$	1.47
Bluetooth® On	45.0	$8.6x10^{-03}$	0.39
Microcontroller On	45.0	$6.3x10^{-03}$	0.28
Total			2.14

The total theoretical daily charge consumption is 2.14 Coulombs (C). Over a period of six months the charge requirement is 390C which gives a capacity requirement of 108mAh. Batteries are usually sized much larger than this and it would be an inconvenience to have to remove the internal module from the cylinder on a regular basis. Hence the battery has been over-sized to last as long as possible without the need of removing the internal module. The internal battery should be sized to last between 7-10 years, after which the internal module will need replacing. This is because 7-10 years is the average shelf life of most batteries.
One of the first cells that was considered was the LR44 which has a small form factor and would be able to fit through the 20.6mm diameter hole for the cylinder valve. The downside to this cell is that its capacity is low at 175mAh, which would therefore mean that this battery would need replacing often. Therefore, this cell will not be used.
The CR20xx series of coin cells have a similar diameter to the cylinder opening which will make it difficult to fit through the valve aperture and therefore they are inappropriate for this project.
The LS14250 is a $\frac{1}{2}$AA battery with a capacity of 1.2Ah at 3.6V. The battery conforms to the 20.6mm diameter cylinder hole so it will be able to fit inside. However, due to the internal battery's charge needing to last its full useable lifetime, the capacity of this cell will not be enough to last the required 7 years with a daily usage of 2.14C.
The CR123A has the largest capacity out of this selection at 1.7Ah at 3V. It will be able to last over the required 7 years due to its larger capacity as well as the battery having a long shelf life. In addition, the diameter of this battery is 16.7mm and its length is 34.1mm. This means that this battery will easily fit through the valve aperture of 20.6mm diameter. Therefore, this is the battery that will be used inside the cylinder.

4.2. External

4.2.1. Requirements

The external module will sit on the outside of the cylinder and must; connect wirelessly to the module inside the cylinder, receive the fill level, package that together with the location of the cylinder, and send all the data to a cloud database. Once the data has been sent, the module and peripherals need to enter a low power state for 24 hours. Once 24 hours has elapsed, the module and peripherals must

wake up, and repeat the process again.

4.2.2. Satellite Navigation

The data being sent from the cylinder not only has to include the fill level of LPG, but also the global position of the cylinder. This is so that once the cylinder is empty, the refill logistics team know where they need to go to replace the cylinder. It also acts as a deterrent for thieves, and aids in stolen cylinder recovery.
There are four main satellite systems; Global Positioning System (GPS), Global National Satellite System (GLONASS), BeiDou, and Galileo. Each system has its benefits, and areas of constant coverage. BeiDou is structured to provide maximum coverage to Asia, GLONASS is a Russian satellite navigation system, and Galileo is a European-specific version of the more widely known American GPS system. [43] [44] [45] Most commercial satellite navigation systems use all the available satellite systems, and as a result GPS has become the de-facto reference name for all satellite navigation systems.
The GPS chip we use must not only work globally, but must also be very low power. Ideally we would be able to specify an interval for data transmission, either through a delay inside the chip, or by sending a sleep or wake command over a serial connection. We found the Seeed Studio Grove-GPS (Air530), which has a sleep current of 31μA and a positioning accuracy of 10m. [46] The configuration of this chip is discussed later.

4.2.3. Communications

Once the fill level from inside the cylinder has been received, and the GPS location has been acquired, the design needs to send the data to the cloud. The cylinder will be located in rural communities and may not have access to an internet connection. Therefore we must turn to other methods of long-range communications.
When global long-range communications are mentioned, most would think of satellite communications as the go-to option, however the power requirements of such devices places them outside of our scope, therefore we must look for other options.
One such method is to use the General Packet Radio Services (GPRS) structure. GPRS is a packet-based wireless communication service most commonly found in mobile phones. [47] The Arduino Cellular board is a low-power option that seamlessly connects with the Arduino IoT cloud, a very simple web dashboard allowing for basic board monitoring. [48] Although the GPRS protocol requires a SIM card, it provides near unlimited coverage. Even though the board is marketed as low power, the power draw associated with sending data over GPRS would mean the system would draw too much current to last the required lifetime.
The other low-power communications protocol available is the Long Range Wide Area Network (LoRaWAN). LoRaWAN is a low power protocol operating on licence-free frequency bands. Low cost and low power broadcast units can connect to a central base station up to 20km away. This central base station is connected to the internet and transmits the data provided by the broadcast units to the cloud. One base station can connect to over 100 end point modules, meaning that a single base station could cover an area with a radius of 20km and handle all of the cylinders in that area, as shown in Figure 4.2. Microchip Technology Inc.'s RN248 chip is an 868MHz module

based on the LoRa, with a sleep current of 1.6μA and a maximum transmit current of 44.5mA, this ultra low power chip is perfect for our use case. [49]

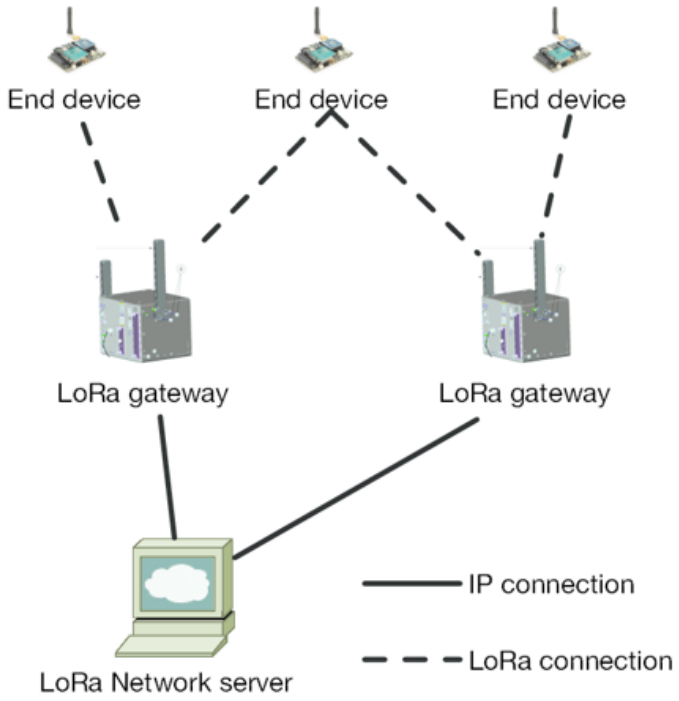


Figure 4.2: A diagram of a LoRa network [50]

4.2.4. Power

The external module is required to be powered between 6 weeks and 6 months in accordance with the specification, as show in Appendix A.1.
The components in the external module that require power are the microcontroller with integrated Bluetooth®, the GPS chip, and the LoRa chip. The current consumptions for the microcontroller's sleep mode, normal running mode, and Bluetooth® transmission mode are all the same as the internal module's with 17μA, 6.3mA and 8.6mA respectively. [42]
In addition to these values, the GPS and LoRa modules can be categorised into low power modes, and transmission modes. The GPS maximum transmission current is 42.6mA and its sleep current is 31μA. [46] The maximum transmission current for the LoRa chip is 44.5mA and the maximum sleep current is 1.6μA. [49]
The GPS and LoRa are both activated first after waking up due to their long start up times. The Lora module takes a maximum of 18 seconds to connect to a gateway, and the GPS takes a maximum of 28 seconds to initialise a connection. The Bluetooth® is initialised and starts scanning for the internal module. A connection is likely to be made in the first 5 seconds of scanning. Once it has connected, the external module will receive a binary value for the state of the switch within 5 seconds. The GPS will get the location co-ordinates in the form of latitude and longitude and the LoRa will send a data packet containing the GPS data and the switch state to the local gateway where it is submitted to the cloud.
Once this action is complete, the microcontroller, GPS, and LoRa chips will enter low power mode for a duration of 24 hours. Therefore, the total maximum time that

the microcontroller will be active for is 38 seconds. By using these timings and the currents of each component, the total charge consumption for the external module can be calculated. This is summarised in Table III.

Table III: Table of time spent per day for each external component

External Component Task	Time /s	Current /A	Coulombs per Day /C
LoRa On	18.0	4.45×10^{-02}	0.801
Microcontroller On	38.0	6.30×10^{-03}	0.2394
GPS On	28.0	4.26×10^{-02}	1.1928
Bluetooth® On	10.0	8.60×10^{-03}	0.086
Microcontroller Sleep	8362.0	1.70×10^{-05}	1.4682
GPS Sleep	8362.0	3.10×10^{-05}	2.6772
LoRa Sleep	8362.0	1.60×10^{-06}	0.1382
Total			6.6028

The theoretical daily charge consumption for the external module is 6.6C. The charge requirement over a 6-month time period is 1202C which translates to a battery requirement of 333mAh.
Due to the external module having easier accessibility, the battery chosen for this module should be rechargeable as it would allow for multiple uses and would decrease the environmental impact of e-waste. Three different batteries were investigated to determine the option best suited to this project.
The rechargeable C–cell runs at 1.2V with a capacity of 3Ah. The capacity for the C-cell is too large for the 6-month time period which results in wasted capacity and therefore cost. In addition, the form factor of this battery is large and would lead to a larger casing being required to fit to the cylinder.
The AAA rechargeable cell runs at 1.2V and has a capacity of 800mAh and will require 3 cells to reach 3.6V. The capacity of this battery is well suited to the needs of this project. However, the multiple cells add an extra cost to the battery as a whole, as well as the need for a battery holder.
The 700mAh battery pack operates at 3.6V and offers over double the capacity required. The cost of this battery pack is similar to the multiple AAA cells required and an additional battery holder is no longer required, which is better for the bench prototype. Therefore, the external battery choice is the 700mAh battery pack.

4.3. Cloud Infrastructure

Part of the original specification requires that the data from the cylinder be displayed to the user via a web or mobile application, as shown in Appendix A.1. The data from the LoRa module has to be transmitted to a local base station, which then transmits it to the internet. Once in the cloud, the next step is to manipulate the received data into a user-friendly format, and then show it in an application. The options for base stations are to either custom-build one using a Raspberry Pi or

similar device, or to use an existing network with free-to-use base stations. These open-source base stations are rate limited, and can be subject to fair-use policies. In the final project we would advise that custom base stations be used. However, for prototyping purposes we have used an open-source network.

4.3.1. The Things Network (TTN)

The Things Network (TTN) is a global, open, LoRa network, which removes the requirement for users to create their own base station. [51] TTN also has native support for Cayenne Low Power Payload (CayenneLPP), an open source method for sending sensor data over LoRaWAN. [52] CayenneLPP uses a single byte to signify a sensor's identifier, another byte to identify the data type, and then as many bytes as is required to send the value of the sensor. By compressing the data in this way, each payload can contain two sensors' data, and the overall transmit time is reduced. Once the data reaches TTN, the CayenneLPP transmission is unpackaged, and displayed in the console. This data can then be sent to third-party applications via one of TTN's integrations. For sending to web applications, TTN supports HTTP requests, as well as If This Then That (IFTTT) for simple mobile applications. [53] However, Ubidots is a service which creates both web and mobile applications in a widget format. [54] After testing all the integrations for usability and reliability, Ubidots was chosen for our project.

4.3.2. Ubidots

Ubidots is an IoT application framework that prides itself on not requiring coding or a software development team. It is a natively supported integration in the TTN console, and provides a synchronised web and mobile application. The web application is shown in Figure 4.3 and shows the fill level of the cylinder to be full or empty, as well as an interactive map of the GPS location of the cylinder. Ubidots also maintains a record of all previous locations and fill levels, something that TTN is unable to do. This allows for realtime and historical tracking of usage, and cylinder position.

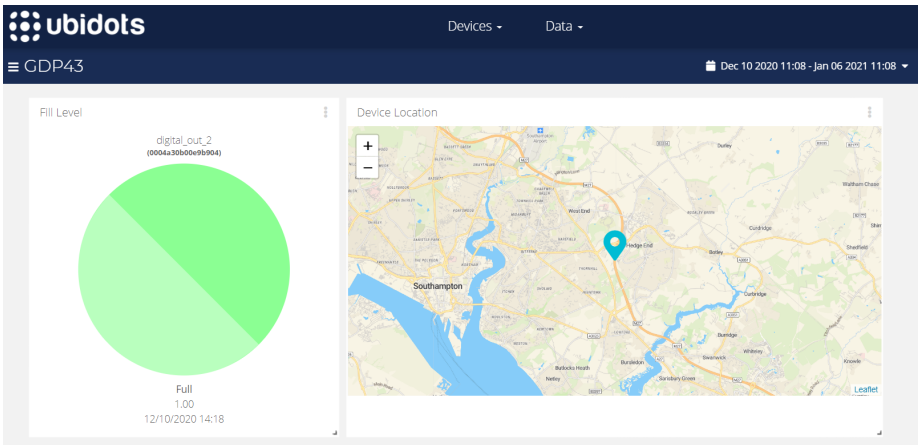


Figure 4.3: The web application for the project in Ubidots.

Chapter 5

Implementation

5.1. Implementation as a Bench Prototype

The specification of this project lists the final product as a bench protoype, as shown in Appendix A.1. This means that certain aspects of a professional product can be ignored, for example; where a professional product would use chips directly from the manufacturer, place them onto custom Printed Circuit Boards (PCB), and program them during construction. Instead, we use chips soldered to breakout boards that are built to do more tasks than we are asking them to do. We use a Nordic Semiconductor® nRF52840 chip for the 'brain' of the project, but it is mounted to an Adafruit Feather to aid prototyping and debugging. [42] This presents an issue when considering the module inside the cylinder, as the Feather is too wide to fit through the valve aperture. In this instance we have designed a custom PCB which would remove this issue. The GPS and LoRa systems are also on breakout boards, and while a production model would place them all on one PCB, for a prototype individual boards are fine.

5.2. Coding

The code for all of the project was written in the Arduino variant of C/C++, mainly because it is the language with which we have the most experience, and it gives us access to libraries which improve the readability and usability of our code.

5.2.1. Internal

The Bluetooth® modules communicate using BLE which is integrated into the Nordic Semiconductor® nRF52840 boards. The internal module acts as a peripheral device to the external module which acts as a central device. The internal module sets up an advertisement package which includes the BLEUART service. The BLEUART

Matthew Carmichael - 5.2.1,5.2.3
Pragash Balachandran - 5.3.2.3
Thomas Whyte-Venables - 5.3.2.1, 5.3.2.2
William Sparrow - 5.1, 5.2.2, 5.2.3, 5.3.1
Zhe Koh - 5.3.2.3

service allows UART communication between boards over BLE. By advertising the BLEUART service, a 128-digit Universally Unique Identifier (UUID) is sent as part of the packet. This service is unique to nRF52xxx boards and hence it can be used as an identifier by the external module. The internal module then sets up its advertising interval and its fast interval timeout. The advertisement then starts and is stopped after 25ms to save power. The start advertisement function is called every 5 seconds once the board has woken up from sleep. When connected to the external module, the internal module then reads the switch value and sends either 1 for full or 0 for empty. This is sent by writing the value to the BLEUART service.
The external module also sets up an advertisement which includes the BLEUART service as this allows UART communication over the BLE protocol to be established. Client UART is also established, which sets up an internal server for the BLEUART communication between the boards. This service can only be used on BLE central devices. An advertising packet is set up in the same way as the internal module and is set to scan for nearby Bluetooth® devices.
When a device is found, its advertising packet is checked for the BLEUART service within its UUID. If the service is present in the advertising packet, the external module will connect to the device. This means that only nRF52xxx boards are able to connect to each other. Once connected, the external module checks if the client UART is able to receive. The client UART then reads what has been sent by the internal module and outputs only the value 1 or 0 with the same designations as before by using a switch case. When the value has been received the external module disconnects from the internal module. The internal module then enters a low power mode upon disconnection.

This method of connection and disconnection was chosen because it would be able to reduce the power draw. The initial approach was to have the internal and external module be synchronised using their real-time clocks and waking up at the same time each day. However real-time clocks have some drift when being used over long periods of time. This drift can be increased by fluctations in temperature, this is a problem because the internal and external modules are subject to different temperature environments. Therefore, the real-time clock is being used over a shorter amount of time. On the other hand, the internal module would not be able to advertise for the entire time period without the battery needing to be replaced regularly. This is why a combination of both of these methods was utilised. This means that the boards are able to go into low power for the majority of the day and then for a short time window, of a few hours, the internal module will be advertising every 5 seconds.

5.2.2. External

The code for the external device is compiled onto an Adafruit Feather. The LoRa and GPS boards are connected using the inbuilt serial port and a software serial port that repurposes spare GPIO pins. This leaves one serial port free for sending debug data to a monitor, if required.

5.2.2.1. *setup*

The *setup* function initialises the various peripheral devices in order of how long it takes each of them to connect and initialise. First it calls *initialize_radio*, and then connects to the BLE chip inside the cylinder. Once it has connected to both the LoRaWAN and the BLE, the GPS is sent the initialise commands from inside *AIR530.cpp* that make it generate the correctly formatted National Marine Electronics Association (NMEA) data.

5.2.2.2. *AIR*530.*cpp*

When the GPS is first powered, the default mode is to transmit out every piece of data it can, as fast as possible. To stop it from doing this, we must send it some configuration commands, as shown in Table IV.

Table IV: Table of *Air*530.*cpp* configuration commands

Function	Command Text	Description
setsatellites()	$PGKC115,1,0,0,1	GPS and Galileo only
setnmea()	$PGKC242,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	GLL data format only
setinterval()	$PGKC101,10000	1000ms (10 seconds)
sleepGPS()	$PGKC105,4	4 specifies ultra-low power, wake on WAKE pin

The only other two aspects of the *.cpp* file are the function *wakeGPS()* which pulls pin 9 on the Feather low, and then high. Pin 9 on the Feather is connected to the GPS WAKE pin. And *calculatechecksum()* is used to calculate the correct checksum to add to the end of the configuration commands. To do this it performs an eXclusive OR (XOR) operation on consecutive values in the command.

5.2.2.3. *loop*

The *loop* function handles; getting the fill level, getting the GPS data, sending it all to TTN, shutting down all the components, and then waiting for 24 hours. Inside the function is a while loop that will only run if the chip is supposed to be awake. When it does run, the first thing it does is to assign the fill level passed over BLE to the variable *level*. Once it has done this, it will not check again, even if the while loop runs again while waiting for other things. This is essential, as the LoRa module will not send data while the BLE is connected.*SoftwareSerial* simply does not work. Once the fill level has been acquired, the function repeats, waiting for the GPS to send data. If it is not formatted correctly, then the GPS is re-initialised and tries again. This is a necessary step due to the temperamental nature of the GPS. In a production model, a more reliable GPS should be used. When correctly formatted data is sent from the GPS, the function passes off the raw Geographic Position-Latitude/Longitude (GLL) data to the *convertLatLong* function, which handles the parsing and sending of data.
As soon as correct data has been received, the GPS is then sent the command to sleep. This means that the GPS is on for the absolute minimum amount of time, which saves power.

Once the data has been passed to the *convertLatLong* function, the function waits for 24 hours. The use of the default delay function in Arduino is deliberate. When waiting, the Feather automatically places itself into ultra-low power mode.
Once 24 hours have elapsed, the function calls all the initialisation functions in the same order as in *setup*, except for the BLE which has a different function called, without the initial constructors present, as to recall those will cause the chip to malfunction.

5.2.2.4. *convertLatLong*

The *convertLatLong* function is called by the *loop* function in the main code block. Its role is to take the NMEA formatted data and convert it to separate latitude and longitude values. It does this by converting the NMEA from a collection of unsigned characters into a string. It can then use functions from the *string.h* library to trim each end of the NMEA data to just give the degrees and minutes data. NMEA data for latitude and longitude is given as **dddmm.mmm** with leading zeroes being inserted as required.
Once degrees and minutes have been isolated, they can be converted to decimal degrees ,the format used by CayenneLPP, by using the equation below.

$$latitude = degrees + \frac{minutes}{60}$$

The function also checks which hemisphere the values for latitude and longitude have come from, and applies a negative modifier as necessary. For this reason, we convert the strings to floating point values so that they can be manipulated by numerical operators.
The *convertLatLong* function then sends the values of fill level, latitude, longitude, and altitude to the *sendLoRa* function.

5.2.2.5. *sendLoRa*

The *sendLoRa* function is called by the *convertLatLong* function and is passed values for fill level, latitude, longitude, and altitude. TTN (the online service which receives our LoRaWAN broadcasts) natively supports a method of encoding called CayenneLPP which is specifically designed for encoding various sensor data. There is a library associated with CayenneLPP, and we use it in this function.
First, we reset the buffer in CayenneLPP, clearing any data from previous broadcasts, then we add a GPS sensor, passing it values for latitude, longitude and altitude. Next, we add a digital output to represent the fill level. Once this is all added into the buffer, we use the *myLora.txBytes* function to send the packet to TTN.

5.2.2.6. *Bleuart_receive*

The *Bleuart_receive* function handles the passing of the fill level over the BLE connection. It is a simple read in to a variable, followed by a set of if statements to check the variable value. During long term testing we encountered an issue where the code would get stuck inside this function. The BLE would disconnect prematurely,

and the fill level would never be received. To combat this, a watchdog timer was implemented in the code. If the code gets stuck in this function for 30 seconds, then the *BLEreconnect* function is called, which restarts the connection between the modules, and allows the fill level to be sent.

5.2.3. ASM Chart

The Algorithmic State Machine (ASM) chart for the design is shown in Figure 5.1. The ASM chart shows the sequential operation of the system and is instrumental in the planning stage of the Software Development Life Cycle (SDLC). [55] The oval states are actions performed by the code, and are performed in the order given in the diagram. The diamond states are questions, the code will check a variable or flag at these points, and will complete a different action depending on the result of the check.

5.3. Build

5.3.1. External

As already stated earlier in section 5.1, the chips we are using have been mounted on breakout boards for the ease of development, and in the case of the nRF52840 it has been mounted on the Adafruit Feather, which allows it to be programmmed using the Arduino IDE. Figure 5.2 shows the three chips soldered to a piece of stripboard.

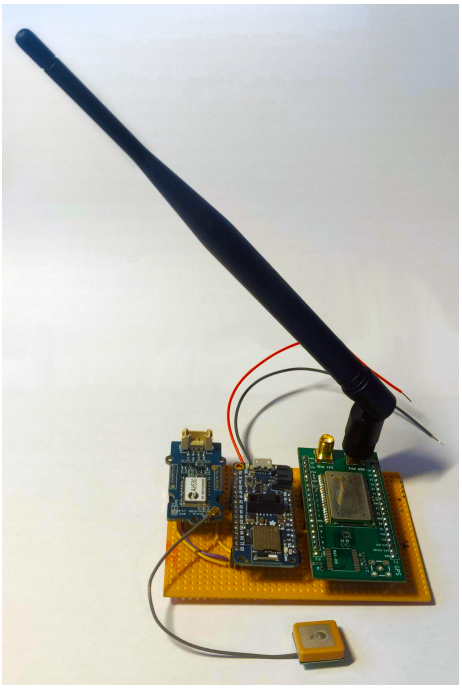


Figure 5.2: A photo of the external system soldered to stripboard

This is sufficient for a bench prototype, a complete product will have an enclosure protecting it from the elements. When this enclosure is created, the GPS antenna (the square on the end of a wire at the bottom of 5.2) will need to have sky access, otherwise the Air530 will be unable to get a GPS fix. In addition, the LoRaWAN

antenna (large black antenna) will need to be unobstructed so that it can transmit to the nearest base station.

5.3.2. Internal

5.3.2.1. CAD Work

The initial Computer Aided Design (CAD) work was done using the design choice mentioned in Section 4.1.1. The major design constraints were calculated by the width of the valve aperture using 5/8" British Standard Pipe threading. This has an internal diameter of 20.6mm so for tolerance the maximum size the sensor could have been was a diameter of 20mm. The height of the level at which the float needed to actuate the switch was calculated by creating the cylinder in CAD. The maximum fill level of a cylinder is 80-85% of the maximum volume of the cylinder so when it is deemed to be "low" on LPG that was selected to be 10% of this 85% maximum volume which puts the float at approximately 150mm above the bottom of the cylinder.
Once these critical dimensions were known, the float sensor system could be modelled around the design principles of the float discussed earlier. This resulted in the design as shown in Figure 5.3.

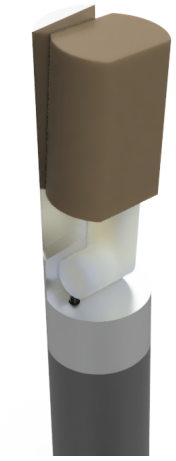


Figure 5.3: A rendering of the float sensor configuration with one side panel cut away to show the switch embedded in the body.

Once this design had been completed, the next challenge for the design was maintaining the position of the float inside the cylinder. As this sensor is dropped into the cylinder, and is removable, it isn't fixed in place. Given that the bottom of the cylinder is spherical due to the pressure vessel's requirements this makes it very difficult for the sensor to remain upright. The spherical nature of the cylinder bottom was used as a method of self centering the sensor in the cylinder.
To keep the sensor upright, a system of spring loaded arms was devised such that as soon as the sensor has passed the valve aperture the arms open up and hold the sensor upright against the sides of the cylinder. This is shown in Figure 5.4

Figure 5.4: A rendering of the extended arms to brace against the walls of the cylinder.

The materials for the arms had to be carefully selected such as to not damage the internal walls of the cylinder. HDPE is a fairly soft material meaning that the arms could not be made of metal, for the sake of the design the arms were selected to be made from HDPE as this would be unlikely to damage the internal walls due to having the same hardness.
The electronics and battery were stored in the base of the sensor as there was a requirement for material to exist there to raise the float to the correct height. This also had the added benefit of making the base of the sensor the most dense part of the entire assembly, increasing the stability of the whole system when it is being transported between locations. The base with a cutaway is shown in Figure 5.5 as it allows for the location of the Bluetooth® module and battery to be more easily visualised.

Figure 5.5: A rendering of the extended arms to brace against the walls of the cylinder.

5.3.2.2. 3-Dimensional (3D) Printing

For the bench prototype of the float sensor it was decided that manufacture would be mainly performed with 3-Dimensional (3D) printing, specifically Fused Deposition Modeling (FDM). Material selection for this was chosen to be Polyethylene Terephthalate (PETG) as it has very high tensile strength, high chemical resistance, and is easy to print relative to other engineering prototype filaments, such as Ultem and Nylon.
Due to the lack of easily obtainable NBR material, a test float print was made with

very high tolerance settings. This float test piece was then submerged in the testing medium (mineral oil) for several days to test how resistant it was to the ingress of oil. After the test, it was confirmed that it was possible to print a float for the bench prototype and a very low density float was printed.
Each part was printed and finished with sandpaper to ensure a smooth engagement of moving surfaces. The separate parts were then glued together using cyanoacrylate based adhesive.
Due to the constraints of 3D printing, and the breakout board for the Bluetooth® module being larger than the valve diameter, it was decided that the base for the Bluetooth® module and battery would not be manufactured, as it was a bench prototype. A battery holder was soldered to the breakout board and the switch was soldered to the appropriate pins so that testing of the float could proceed.

5.3.2.3. PCB

The PCB for the internal circuitry was to be designed as a proof of concept for the final product to show that the design is feasible and can be implemented with an actual LPG cylinder. The PCB to be designed should replace one of the breakout boards that was assembled and implemented for the bench prototype.
The breakout board to be replaced contains the Nordic Semiconductor® nRF52840 , as well as connections to the float switch. Since the Adafruit Feather to be replaced has additional features such as voltage regulators, Light-Emitting Diodes (LED), and other peripherals which are not used in this project, these can be omitted from the final design and a smaller PCB with only the required features can be designed for production.
The purpose of the board design is to show that it can fit through the valve aperture. It should be housed within the body of the float structure and as such, its main specification is that its width needs to be under 20mm.
The PCB design needs to achieve the same functionality as the prototype circuit, with a smaller form factor. The switch, situated within the cylinder during its operation, needs to connect to the board and be wired appropriately. The 2 additional through hole pads situated on the board allow for these wired connections from the float mechanism to connect to the chip and signal the change in state when the fill level drops below the threshold for refilling.
For the PCB proof of concept, the full Adafruit Feather module is not needed, as not all of its functionality is required by our product. For this, the MDBT50Q chip, which can be found on the Adafruit Feather, is used. The MDBT50Q is a Raytac derivative of the Nordic Semiconductor® nRF52840. [56] The chip allows us to input a signal as data from the switch and allows Bluetooth® transmission to the external mainboard. Its dimensions are 10.5 x 15.5 x 2.2 mm, which allows it to comfortably fit onto the PCB and still meet the maximum size specifications. Ports were added for the Joint Test Action Group (JTAG) interface for testing and debugging purposes. A pull-down resistor was also added after referring to Adafruit's schematic layout. Figure 5.6 shows both the schematic diagram and the board layout of the final PCB, as well as its dimensions.

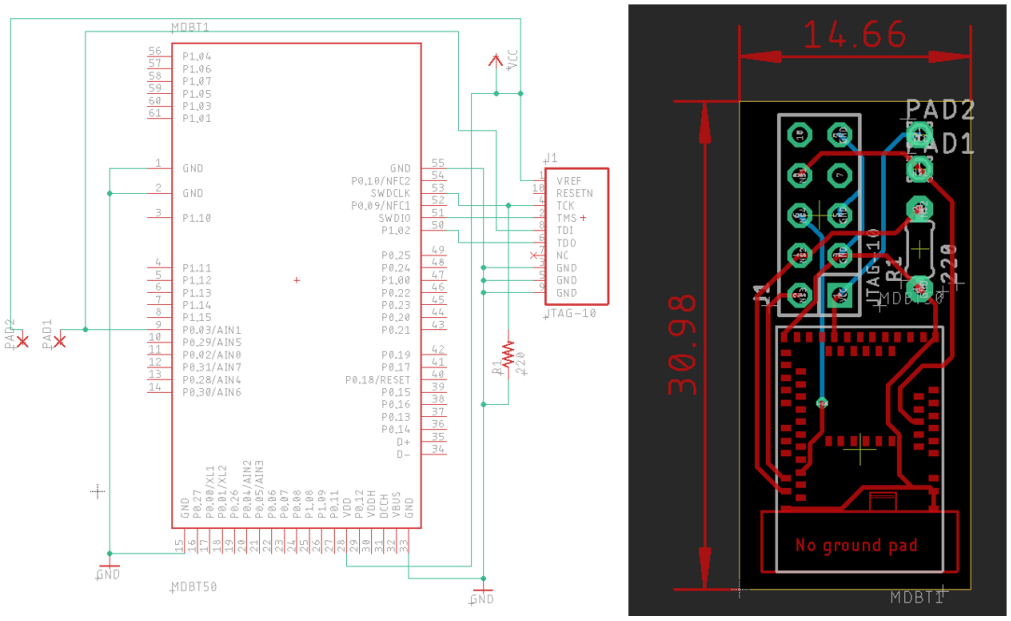


Figure 5.6: The schematic diagram and board layout of the PCB.

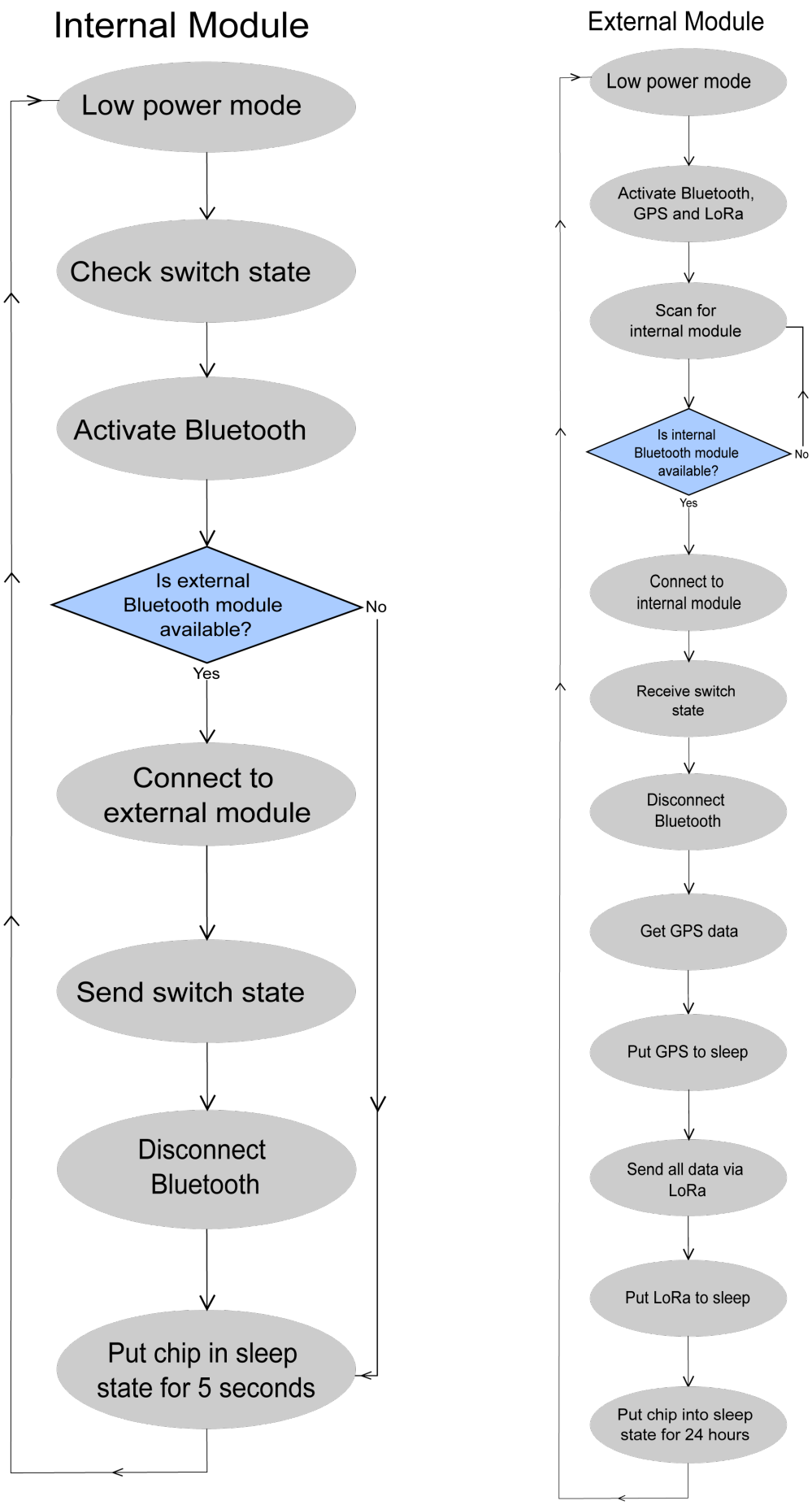


Figure 5.1: The Algorithmic State Machine (ASM) chart for the design

Chapter 6

Testing

6.1. Internal

6.1.1. Bluetooth® Power States

6.1.1.1. Signal Strength Testing

The Bluetooth® power states control the output signal strength from the Nordic Semiconductor® nRF52840 microcontroller. The different signal strengths require different amounts of power usage. The higher the signal strength the higher the power required to send the signal.
This project requires as little power as possible to send the data from the internal module to the external module. Therefore, the lowest stable signal strength will be used to send the state of the float switch. The set-up of this test includes various control variables. The first is the distance between the two nRF52840 boards being set to 1.5 metres. This value was chosen as it is an over-estimate of the distance that it would need to travel to the external module on the cylinder. Furthermore, the environment for the microcontroller for the internal module was set inside a plastic container to simulate the composite cylinder. Secondly, the advertising duration was kept constant at 500ms every 5 seconds, which is more than long enough to reliably connect every time, provided the signal is strong enough. The last control variable was adding a 2 second delay after disconnecting for both the internal and external modules. This was introduced to ensure that board had disconnected correctly. Without this delay, the boards could reconnect almost instantly without giving an indication that they had disconnected. By adding this delay, the BLE connection LED flashes on the external nRF52840 board, which indicates that it is scanning for a BLE device. This also gives time to start the stopwatch. Without this delay the boards would connect faster than human reflexes would allow.

The test included two nRF52840 boards, one of which was connected to the computer to monitor the switch state, this would act as the external module. On the other

Matthew Carmichael - 6.1.1
Thomas Whyte-Venables - 6.1.2
William Sparrow - 6.2, 6.3

hand, a plastic container housed the other board with a battery, at a fixed distance of 1.5m from the other module, this would act as the internal module.
The test started with the maximum transmit power which is +8dB, this value is set in the *setup* function of the internal module. The command *Bluefruit.setTxPower(8);* was used to set the signal strength of the advertisements being broadcast by the internal module. The external module had a fixed transmit power of +4dB throughout the test. This was done to see the effects of reducing the internal signal strength only.
During the test after the initial connection, the external module disconnected from the internal module at the push of a button. The time taken for the boards to reconnect was recorded and this was repeated 4 more times to give an average over five data points. Once all five tests were completed, and an average was taken, the internal module was reprogrammed with a change to the transmission power, reducing it to the next signal strength down from the previous value. This method was then repeated for all other signal strength levels that can be achieved on the nRF52840 board. The results can be found in Table V.

Table V: Table of time taken to connect when signal strength is decreased

	Time to Connect /s					
Signal Strength /dB	Test 1	Test 2	Test 3	Test 4	Test 5	Average
8	1.52	1.84	1.64	2.07	1.84	1.782
7	1.79	10.09	2.10	1.97	2.03	3.596
6	1.97	2.03	9.94	10.98	1.97	5.378
5	10.53	10.53	10.54	1.97	1.84	7.082
4	2.29	1.70	20.94	1.78	1.96	5.734
3	1.77	2.10	2.03	1.90	2.23	2.006
2	1.87	2.43	1.97	2.36	10.07	3.740
0	1.53	1.84	2.03	1.91	1.90	1.842
-4	1.93	1.85	1.90	10.51	10.29	5.224
-8	2.03	1.71	1.77	10.56	2.30	3.674
-12	21.77	2.17	1.91	2.16	21.54	9.910
-16	1.89	2.03	2.23	2.09	2.23	2.094
-20	43.90	2.57	10.89	2.24	10.27	13.974
-40	N/A	N/A	N/A	N/A	N/A	N/A

6.1.1.2. Advertisement Duration Test

The next test for the Bluetooth® power states was to determine the lowest feasible advertisement duration. This would also save power as the internal module would be advertising for a number of hours each day. The more time that can be spent in sleep mode, the longer the battery will last. This test is set up in the same way as the signal strength test, with the only change being that the transmission power is set to the lowest stable signal strength which is -16dB. The duration of the advertisement is controlled using a delay between starting the advertisement and forcing the advertisement to stop. The starting advertisement duration was 500ms, which is the same as the signal strength test. The external module disconnected five

times after the initial connection and the times for reconnection were recorded and averaged. The advertisement duration was then reduced to 250ms and the testing procedure was repeated, with each iteration getting shorter until a 1ms duration was reached. The results of this test can be seen in Table VI.

Table VI: Table of time taken to connect when advertising duration is decreased

	Time to Connect /s					
Advertisement Duration /ms	Test 1	Test 2	Test 3	Test 4	Test 5	Average
500	1.89	2.03	2.23	2.09	2.23	2.094
250	2.63	2.23	15.60	2.23	2.21	4.980
100	1.85	10.28	18.06	2.42	2.04	6.930
50	2.58	2.10	15.14	2.16	2.40	4.876
25	2.34	2.10	2.10	15.98	2.33	4.970
10	55.49	2.30	10.28	2.18	18.46	17.742
5	2.3	2.23	2.11	70.6	134.45	42.338
1	N/A	N/A	N/A	N/A	N/A	N/A

6.1.2. Float Testing

For the testing of the prototype float design, a testing jig was made from a large glass jar filled with a nonconductive medium, mineral oil. Mineral oil was chosen as it has a density of about 0.85g/ml, is nonconductive, and is easily available comercially. The density is not close to the desired 0.52g/ml because upon further research most liquids at this density either require pressurisation, or are extremely flammable. Pressurisation would be an issue as it would mean changing the environment and observing results would be very difficult, and flammable materials present a serious safety hazard. This difference in density was taken into account and the float was manufactured and modified to reflect the relative density change.
An Adafruit Feather was programmed to read the input of a pin and set an LED to be on or off depending on the switch state of the float. The Feather was powered through the use of the CR123A battery, in a battery holder. The program had pin *A5* set as an input with a pull-up resistor. This pin was connected through the float sensor to ground, such that when the switch in the float sensor was depressed, the pin was tied to ground. This pin was checked every 100ms for a change in the switch state, and then changed the LED state to reflect any changes. When tied to ground, the Feather would turn the red LED on, and when pulled high by the internal resistor, it turned the LED off.
As shown in Figure 6.1, when the level was below the actuation point the LED turned off. This indicated that the switch was not depressed by the float.

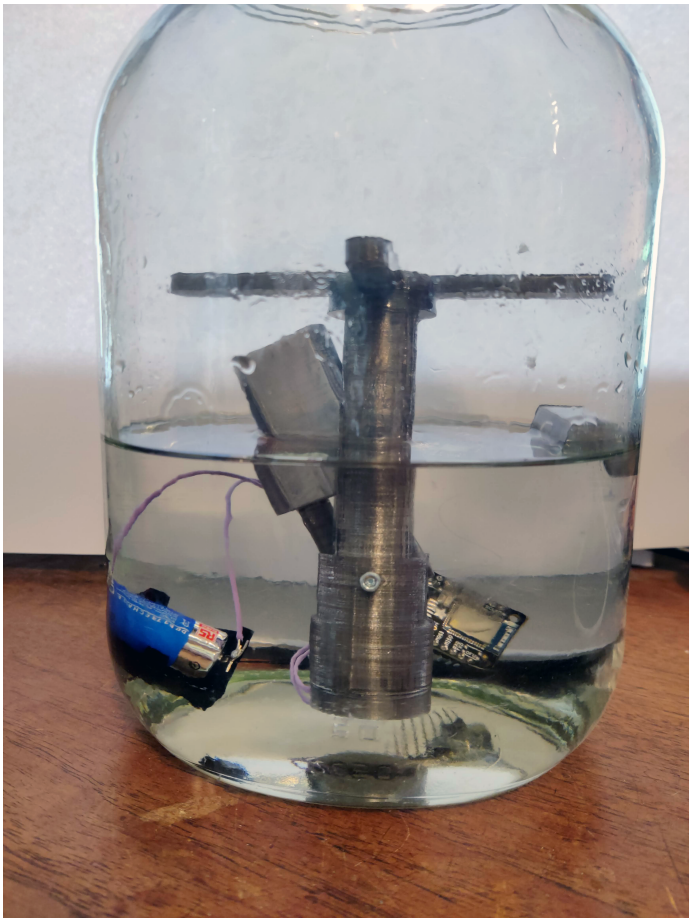


Figure 6.1: This picture shows the float sensor when the level is below the actuation point and as a result the LED is off.

As shown in Figure 6.2, when the level was greater than the height of the float the LED was on. This indicated that the float was depressing the switch.

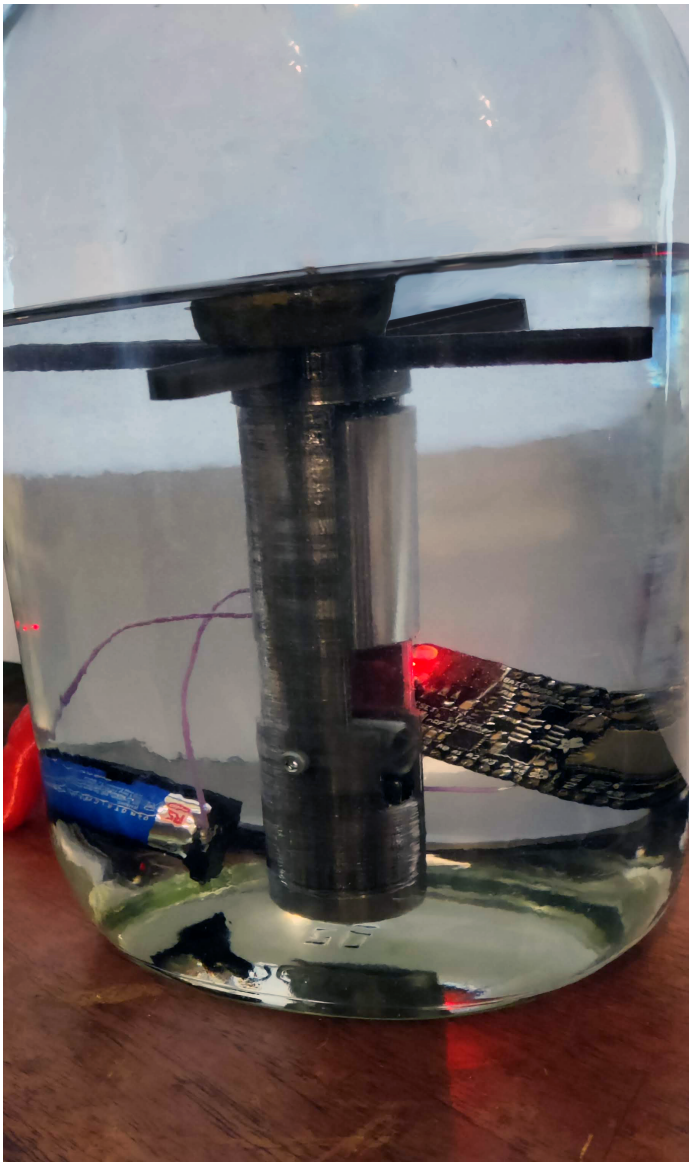


Figure 6.2: This picture shows the float sensor when the level is above the actuation point and as a result the LED is on.

6.2. External

6.2.1. Design Power Consumption

To evaluate the success of the project, the power consumption of the design must be evaluated, this has already been calculated theoretically, but it must also be calculated practically.
In order to do this, the completed design was connected to a Keithley Source Measurement Unit (SMU) and the current and voltage usage was measured over a period of 3 cycles, with a 30 second delay simulating the 24-hour period. The full set of SMU readings can be seen in Figure 6.3. The first cycle is the initialisation cycle, where the GPS is initialised, as well as other 'one time' tasks. Therefore, it does not follow the same pattern as the subsequent two cycles. The middle cycle is the first 'wake from sleep' cycle, and is shown in red in Figure 6.4. In order to

find the total current usage of the design, we must evaluate its current draw over 24 hours. The two states that the design can be in are *sleeping* and *awake*. We have the *awake* state in real-time, but the *sleeping* state must be extrapolated from the 30 second sleep to a 24 hour sleep.

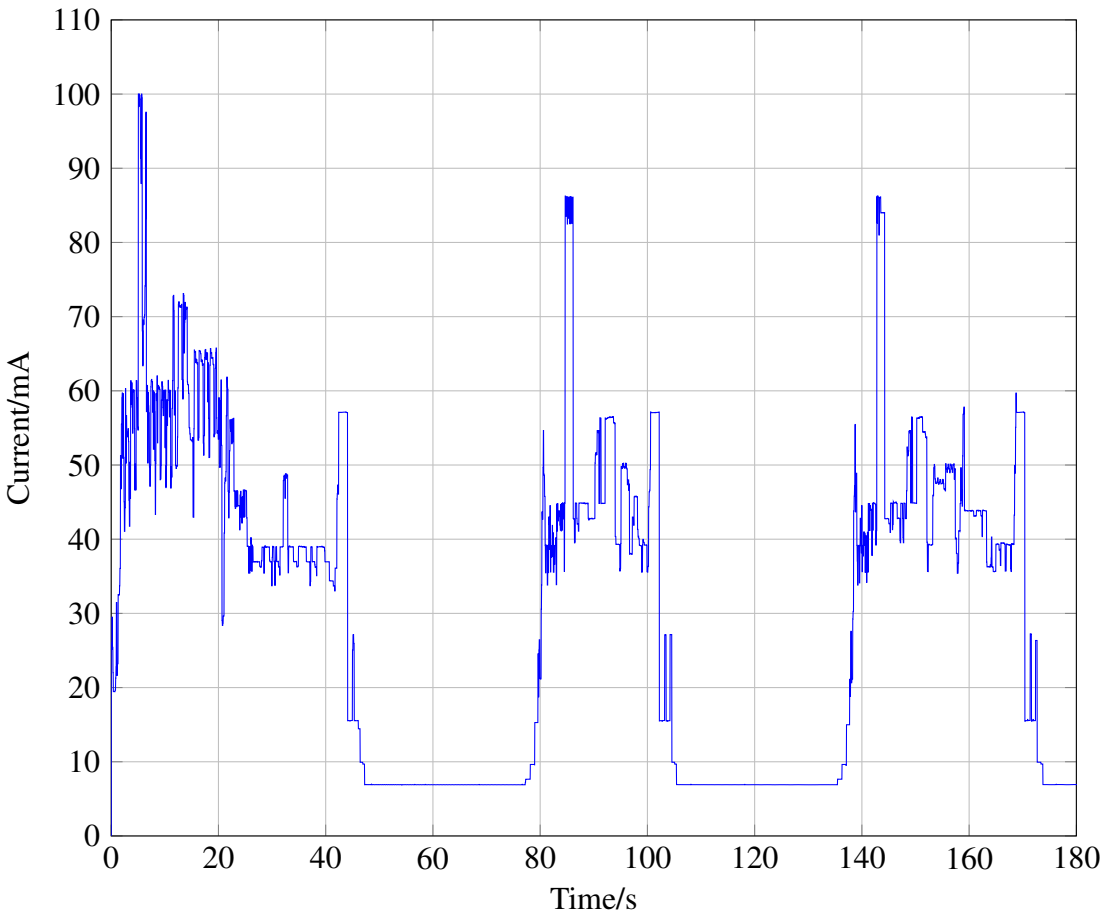


Figure 6.3: Graph of power output for three cycles of the design.

Using the middle cycle shown in Figure 6.4, we calculated the total charge used during the 28 second *awake* state to be 1.15C. Then, by finding the charge usage for one second of the *sleeping* state, which was 0.00492C, we calculated the charge usage for 24 hours of *sleeping* state to be 424.7C. Obviously, this was much higher than the theoretical value, and this was because of unnecessary components on the breakout boards, as opposed to just having the current draw of the chips.

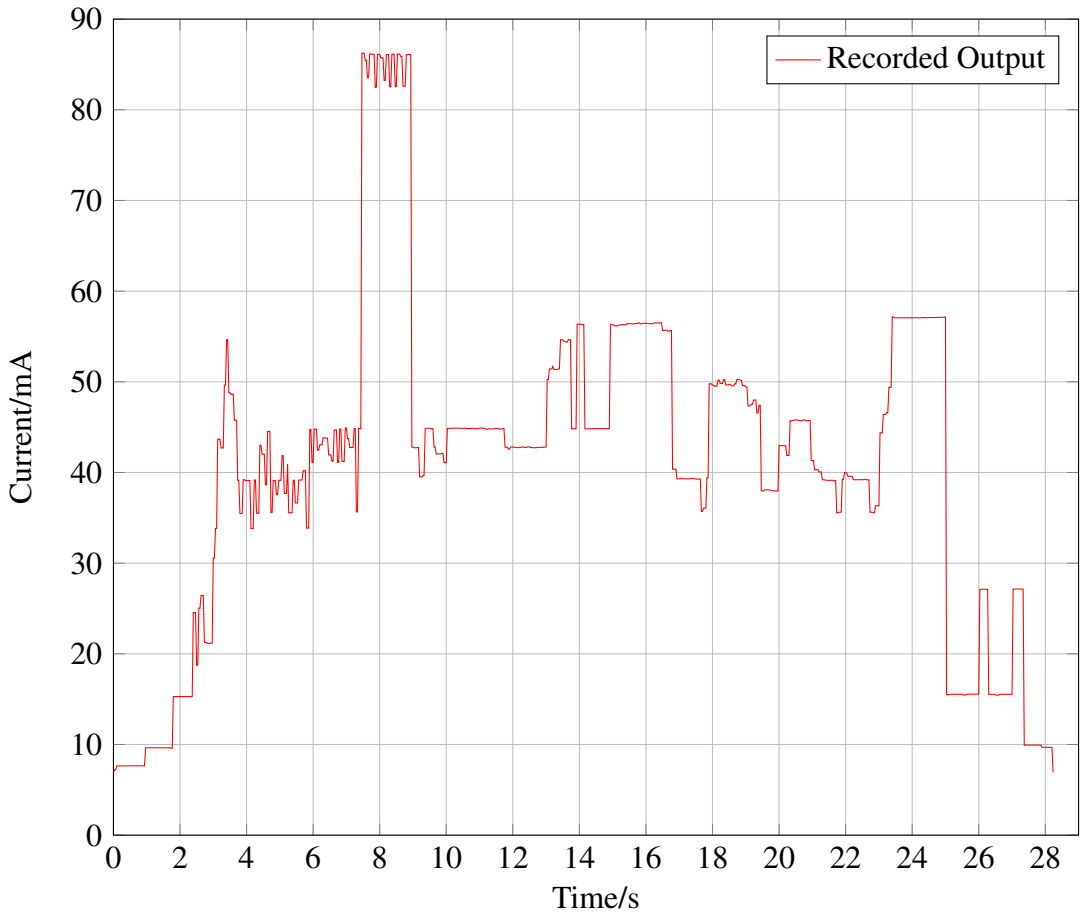


Figure 6.4: Graph of power output during the wake and transmit sequence.

6.3. Total System Testing

6.3.1. 6 Month Robustness Test

The specification states that the system must run for between 6 weeks and 6 months, (A.1), as this is how long a cylinder is expected to last before the LPG inside it needs to be refilled. As the project timeline does not allow for a real-time 6 month test, we changed the sleep delay from 24 hours to 30 seconds, meaning that a 6 month test (183 days) was simulated in a 3.5 hour test. Throughout the test, the fill state of the cylinder was periodically changed, resulting in Figure 6.5 which was made using data extracted from Ubidots.

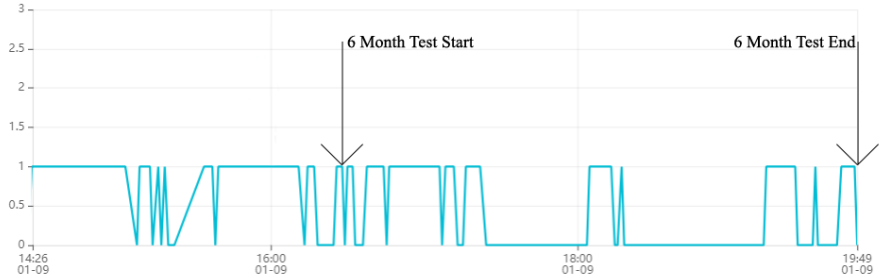


Figure 6.5: A graph of user controlled fill-level, taken from the Ubidots Dashboard

6.3.1.1. Active Time

By analysing the data sent by the system to the serial port, we identified the average time taken for an active cycle to complete. By checking the time difference between the message *sleeping* and the message *LoopNumber:#*, we determined the time the device was active for. This is because in the monitoring version of the code, these are the first and last messages sent when the system sleeps and wakes up. The results of this analysis are shown in Table VII.

Table VII: Table of averages for active system time

Averages	Time/s
Mean active time	38.39 $(2.d.p)$
Mode active time	31 (to the nearest second)

The graph of all the active times for a 6 month cycle are shown in Figure 6.6.

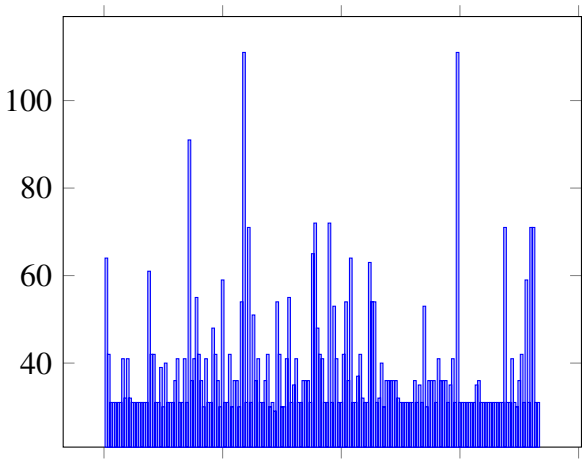


Figure 6.6: The graph of the duration of active times for a 6 month simulated cycle

Two of the active times were above the 100 second mark, both at 111 seconds. As well as one duration of 90 seconds, there were also a selection of durations around the 70 second mark. All of these will be discussed in our evaluation later.

6.3.1.2. Sleep Time

The serial port data also allowed us to see the average sleep time between active cycles. The worry here was that if there was a large amount of discrepancy when scaled up to 24 hours, then the external and internal modules may not wake up at the same time. This would lead to them wasting power searching for their inactive counterpart. However, we can observe from Figure 6.7 that there was little to no variation in the 30 second delay.

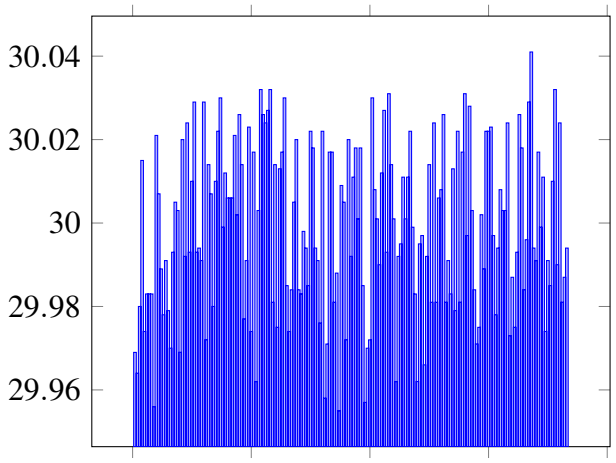


Figure 6.7: The graph of the duration of sleep times for a 6 month simulated cycle

With an average sleep time of 29.99 seconds, we can safely say that clock desync will not be an issue when we scale up the delay to 24 hours. When the Bluetooth® connects, the clocks of the external and internal modules are re-synced back together. Therefore, the maximum amount of time that the clock will be running while susceptible to desync would be 24 hours. With this test completed, we have discounted clock desync as a possible cause of system failure.

Chapter 7

Evaluation

7.1. Internal

7.1.1. Bluetooth® Power States

7.1.1.1. Signal Strength Test

The signal strength test was completed to calculate the lowest signal strength that could hold a stable connection to the external module. The results from Table V show the different signal strength levels and the time taken to reconnect to the external module. The shorter the time taken to reconnect, the more stable the connection. Therefore, the average time taken to reconnect can describe the stability of the connection.

Table V shows the trend that as the signal strength decreases, the longer average time it will take to reconnect. The signal strengths with the weakest stability were -40dB, -20dB and -12dB. This is due to the fact that -40dB was unable to connect at the 1.5m distance and the signal was too weak to penetrate through the plastic container walls and reach the external module. The -20dB and -12dB signal strengths both had consistent long reconnection times. The average reconnection times were both at and over the 10 second mark which is not acceptable. This means that it is likely that the boards would reconnect within 15 seconds which is a significant amount of time and battery capacity will be wasted on connecting the boards together. The internal and external modules must connect in the shortest times possible to reduce the power consumption.
The next set of signal strengths were fairly stable. However, their average reconnection times were over five seconds. The medium stability signal strengths included in this set were: +6dB, +5dB, +4dB and -4dB. This set of signal strengths are more feasible in this project as they are likely to connect in under 10 seconds. However, this was not the fastest reconnection time.

Matthew Carmichael - 7.1.1
Thomas Whyte-Venables - 7.1.2
William Sparrow - 7.2, 7.3, 7.4
Zhe Koh - 7.2

The fastest reconnection times were all at the 2 second mark, which was the same as the delay which was inputted to allow for the timer to be started. The average reconnection times for this set were all under 5 seconds and therefore it is likely that these signal strengths will connect within 5 seconds of disconnection. The most stable signal strengths were: +8dB, +7dB, +3dB, +2dB, 0dB, -8dB and -16dB. The lowest power of the most stable signal strengths tested was -16dB, which will be used in the final implementation of the design. This is to help reduce the internal current draw while the Bluetooth® advertising is active.

7.1.1.2. Advertising Duration Test

Changing the duration of advertisement is useful to reduce the power consumption of the internal module. By reducing the advertising duration, the total time that the microcontroller will be awake for is reduced, hence maximising the battery capacity. The results of the advertisement duration test are found in Table VI.
From these results, a trend can be seen that as the advertising duration decreases, the average reconnection time increases slowly until a threshold of 10ms is reached. Then, as the durations get even smaller, the average reconnection times increase significantly until no connection is made whatsoever. An advertising duration of 1ms is too short a pulse to be received by the external module and hence will not connect. 5ms and 10ms are not very stable either as on average it will take over 15 seconds to connect to either board which is not permissible for this project. The next fairly stable advertising duration is 100ms which is likely to reconnect within 10 seconds of disconnecting. However, this is not optimised for power consumption. The most stable advertising durations were 500ms, 250ms, 50ms and 25ms which all averaged under the five second mark for reconnection. The shortest of these reconnection times was 25ms, which will be implemented into the final design.

7.1.2. Floats

As shown in Section 6.1.2 the float performed exactly as expected. The change in level is reflected in the float orientation and as a result when the level falls below a certain point, the switch is released and the LED is turned off.

7.2. External

This test was carried out to determine the power consumption of the design. It can be seen in Figure 6.3 that the second and third cycles were similar to each other, and the first cycle was different. This is due to the first cycle initialising all the features, which consumed more power while the subsequent cycles only showed the power consumed after the system 'wakes' from 'sleep' state. The 'awake' power consumption is determined from the second cycle since this is the cycle that is going to be repeated. The measured power consumption was higher than the calculated theoretical power consumption. This is due to the Adafruit Feather having additional features such as voltage regulators, and a battery charger, which were not used in the design and are consuming additional power. We can discount these additional features from the calculations to get a more accurate estimate of power consumption in the final product. These are shown in Table VIII.

Table VIII: Table of discountable sleep current draws

Current Draws	Amount to discount /A
Charging LED	$1.53x10^{-03}$
AP2112 Voltage regulator Quiescent Current	$1.93x10^{-06}$
GD25Q16 SPI Sleep Current	$1x10^{-06}$
LiPo Charger MCP73831 Current	$53x10^{-06}$
GPS eco/ultra-low power mode	$819x10^{-06}$

Firstly, we discount some of the unnecessary components on the Feather (not all components are shown on the schematic); the charging LED, voltage regulator, the SPI controller, and the integrated LiPo charger. Finally, the GPS did not enter its sleep state correctly, so if we assume that it will do this in the final product, we can discount almost a whole milliamp of current draw. Once we have removed these unnecessary current draws, the charge usage per second drops to 3.99mC, and the 24-hour charge consumption drops to 344.5C. The adjusted power consumption for the middle cycle is shown in blue in Figure 7.1.

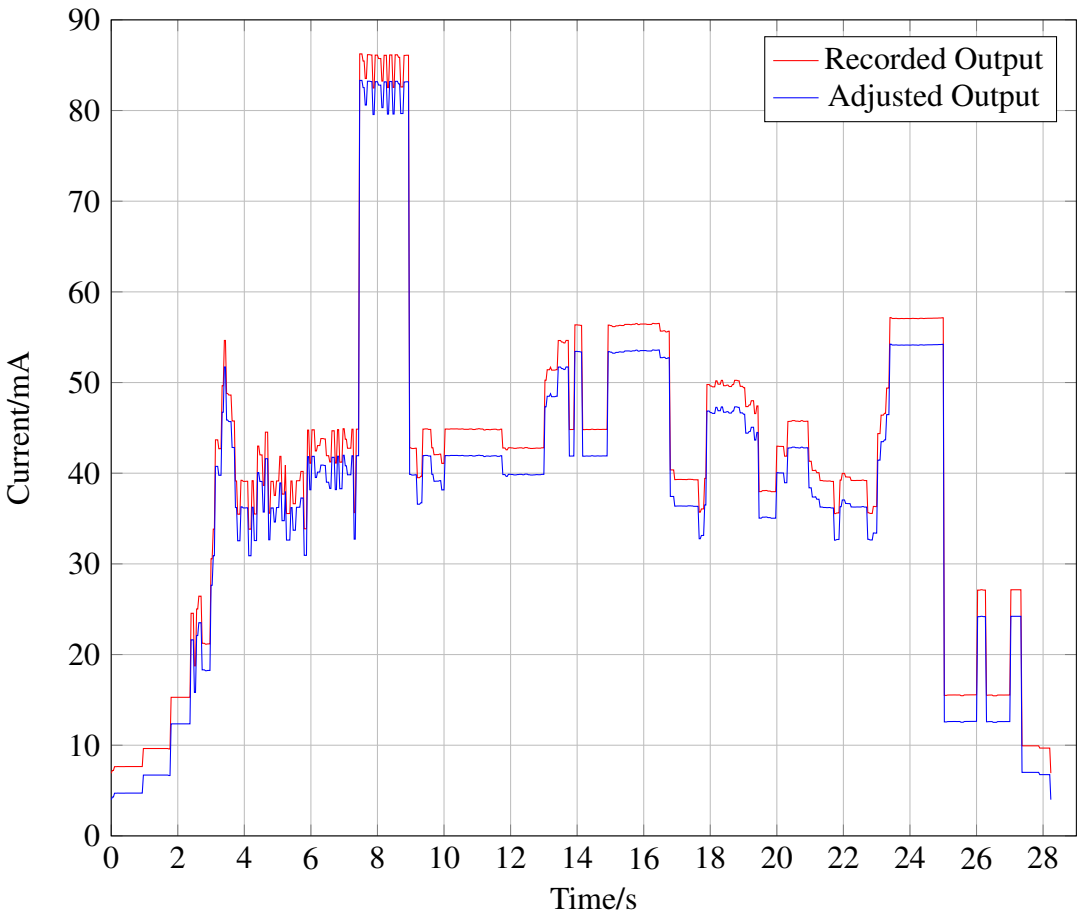


Figure 7.1: Graph of power output during the wake and transmit sequence.

7.3. Total System Testing

7.3.1. 6 Month Robustness Test

In Figure 6.6 there are several active times that are much higher than the mean active time of 38.39 seconds. Using the serial monitor timestamps, we looked at the reasons for this. First we considered the two 111 second cycles. The similar times suggested that both were as a result of the same issue. Figure 7.2 shows the serial readout from one of the 111 second runs. The reason for the 111 second cycle is

```
19:05:58.279 -> LoopNumber:147
19:06:00.058 -> Trying to join TTN
19:06:20.545 -> Unable to join. Are your auth keys correct, and do you have TTN coverage?
19:07:33.838 -> Successfully joined TTN
19:07:33.838 -> Entered Bleuart receive
19:07:38.952 -> connected BLE
19:07:39.868 -> Enable TXD's notify
19:07:39.868 -> Got Fill Level
19:07:39.868 -> 0
19:07:40.384 -> disconnected BLE
19:07:40.384 -> timeout checker
19:07:40.384 -> $PGKC001,105,3*29
19:07:40.384 -> $GPGLL,8960.0000,N,00000.0000,E,190734.485,V$GPGLL,8960.0000,N,00000.0000,E,190744.485,V,N*4GLL Data acquired
19:07:44.593 -> Northern Hemisphere
19:07:44.593 -> Eastern Hemisphere
19:07:44.593 -> Latitude: 90.00000
19:07:44.593 -> Longitude: 0.00000
19:07:44.593 -> TXing
19:07:48.374 -> TX Done
19:07:49.359 -> Slept lora
19:07:49.359 -> sleeping
```

Figure 7.2: The serial readout for loop 147 of the six month simulated test

in line 3 of the readout. The system failed to connect to TTN, which is the LoRa platform we use to send fill data to the cloud. In this case, the code instructs the system to wait for a minute before trying again. When we considered the initial connection time attempt of 20 seconds, plus the 60 second wait before retrying, and then add on the mode active time of 31 seconds, we reached 111 seconds exactly. When we check loop 57, the other 111 second cycle, we found that it also did not connect to TTN initially, and had to wait for an additional minute. Thanks to the Nordic Semiconductor® nRF52840's dynamic power handling, this sleep period will use minimal power as shown in Table III. However, in order to reduce the effect of this on the battery life, the sleep time after an unsuccessful TTN connection should be reduced to a smaller value of around 30 seconds.

The next highest value in Figure 6.6 is an active cycle of 90 seconds, this could not have been as a result of a TTN delay. This is because we have seen that the delay caused by this error was at least 80 seconds, and the rest of the process required at least 30 seconds to complete. Figure 7.3 shows the serial readout from the 90 second cycle. We can see in line 7, and line 10, that the prompt *timeout triggered* showed that the BLE disconnected before the fill level was sent by the internal module, a known bug. When this occurs, a timeout function is triggered after 30 seconds of inactivity. This watchdog timer re-calls the function that connects to the internal module, and the fill level is sent again. The watchdog timer triggered twice in a row for this cycle, so for each time *timeout triggered* is shown, a 30 second delay can be added to the mode cycle duration, as shown in Table VII. This gives a cycle duration of 91 seconds, to the nearest second, which is the same as the duration of loop 34.

The next set of results to discuss are all results that are higher than average because of a single BLE disconnect timeout. Because we know that a watchdog timer triggering introduces 30 seconds of delay, we checked all results that were greater than 50

```
16:55:16.462 -> LoopNumber:34
16:55:18.240 -> Trying to join TTN
16:55:31.543 -> Successfully joined TTN
16:55:31.543 -> Entered Bleuart receive
16:55:33.085 -> connected BLE
16:55:33.611 -> disconnected BLE
16:56:01.094 -> timeout triggered
16:56:03.067 -> connected BLE
16:56:03.578 -> disconnected BLE
16:56:33.591 -> timeout triggered
16:56:33.779 -> connected BLE
16:56:34.493 -> Enable TXD's notify
16:56:39.962 -> Got Fill Level
16:56:39.962 -> 1
16:56:40.485 -> disconnected BLE
16:56:40.485 -> timeout checker
16:56:40.485 -> $PGKC001,105,3*29
16:56:40.485 -> $GPGLL,8960.0000,N,00000.0000,E,165531.983,V$GPGLL,8960.0000,N,00000.0000,E,165642GLL Data acquired
16:56:42.317 -> Northern Hemisphere
16:56:42.317 -> Eastern Hemisphere
16:56:42.317 -> Latitude: 90.00000
16:56:42.317 -> Longitude: 0.00000
16:56:42.317 -> TXing
16:56:46.096 -> TX Done
16:56:47.093 -> Slept lora
16:56:47.093 -> sleeping
```

Figure 7.3: The serial readout for loop 34 of the six month simulated test

seconds in duration. We then identified those that experienced a timeout, and those which were simply a combination of multiple devices taking a longer than normal time to connect and send data. Figure 7.4 shows the serial readout of loop 133, which took 53 seconds to complete. In line 7 the watchdog timer was triggered, which explains the longer than normal duration.

```
18:50:29.609 -> LoopNumber:133
18:50:31.400 -> Trying to join TTN
18:50:44.680 -> Successfully joined TTN
18:50:44.680 -> Entered Bleuart receive
18:50:44.960 -> connected BLE
18:50:45.740 -> disconnected BLE
18:51:14.247 -> timeout triggered
18:51:15.002 -> connected BLE
18:51:15.734 -> Enable TXD's notify
18:51:16.263 -> Got Fill Level
18:51:16.263 -> 0
18:51:16.761 -> disconnected BLE
18:51:16.761 -> timeout checker
18:51:16.761 -> $PGKC001,105,3*29
18:51:16.761 -> $GPGLL,8960.0000,N,00000.0000,E,185045.327,V$GPGLL,8960.0000,N,00000.0000,E,185GLL Data acquired
18:51:18.199 -> Northern Hemisphere
18:51:18.199 -> Eastern Hemisphere
18:51:18.199 -> Latitude: 90.00000
18:51:18.199 -> Longitude: 0.00000
18:51:18.199 -> TXing
18:51:21.951 -> TX Done
18:51:22.970 -> Slept lora
18:51:22.970 -> sleeping
```

Figure 7.4: The serial readout for loop 133 of the six month simulated test

By contrast, Figure 7.5 shows the serial readout for loop 61, which did not trigger the watchdog timer, but still took 50 seconds to complete. However, we can see from looking at the timestamps that the time spent; trying to join TTN, waiting for the fill level to be sent, and waiting for the GPS to send data meant that the process took much longer than in Figure 7.6. Figure 7.6 shows the serial readout for a typical cycle, in this case loop 1, which took 30.7 seconds to complete.

```
17:28:26.149 -> LoopNumber:61
17:28:27.929 -> Trying to join TTN
17:28:41.213 -> Successfully joined TTN
17:28:41.213 -> Entered Bleuart receive
17:28:45.237 -> connected BLE
17:28:46.273 -> Enable TXD's notify
17:28:52.306 -> Got Fill Level
17:28:52.306 -> 0
17:28:52.770 -> disconnected BLE
17:28:52.770 -> timeout checker
17:28:52.770 -> $PGKC001,105,3*29
17:28:52.770 -> $GPGLL,8960.0000,N,00000.0000,E,172840.650,V$GPGLL,8960.0000,N,00000.0000,E,172901.736,V,N*48
17:29:06.122 -> $GPGLL,8960.0000,N,00000.0000,E,172906.026,V,N*49
17:29:12.208 -> $GPGLL,8960.0000,N,00000.0000,E,172912.094,V,N*45
17:29:12.349 -> GLL Data acquired
17:29:12.349 -> Northern Hemisphere
17:29:12.349 -> Eastern Hemisphere
17:29:12.349 -> Latitude: 90.00000
17:29:12.349 -> Longitude: 0.00000
17:29:12.349 -> TXing
17:29:16.093 -> TX Done
17:29:17.117 -> Slept lora
17:29:17.117 -> sleeping
```

Figure 7.5: The serial readout for loop 61 of the six month simulated test

```
16:19:55.468 -> LoopNumber:1
16:19:57.293 -> Trying to join TTN
16:20:10.580 -> Successfully joined TTN
16:20:10.580 -> Entered Bleuart receive
16:20:13.764 -> connected BLE
16:20:14.651 -> Enable TXD's notify
16:20:15.120 -> Got Fill Level
16:20:15.120 -> 0
16:20:15.637 -> disconnected BLE
16:20:15.637 -> timeout checker
16:20:15.637 -> $PGKC001,105,3*29
16:20:15.637 -> $GPGLL,5055.7340,N,00123.2628,W,162012.274,V$GPGLL,5055.7314,N,00123.2598,W,162022.388,A,A*4F
16:20:21.445 -> GLL Data acquired
16:20:21.445 -> Northern Hemisphere
16:20:21.445 -> Western Hemisphere
16:20:21.445 -> Latitude: 50.92886
16:20:21.445 -> Longitude: -1.38766
16:20:21.445 -> TXing
16:20:25.228 -> TX Done
16:20:26.211 -> Slept lora
16:20:26.211 -> sleeping
```

Figure 7.6: The serial readout for loop 1 of the six month simulated test

Where loop 1 only took 2 seconds from connecting BLE to receiving the fill level, loop 61 took 7 seconds. From GPS initialisation to the correct data being acquired, loop 1 took 6 seconds whereas loop 61 took 20 seconds. This shows that the cause of the discrepancy in timings is mainly the GPS module.

Throughout the project, the Air530 has performed poorly at best. The antenna struggles to get a fix, even outside without any interference, and it will randomly decide to remove that fix, and the stored location data, for no apparent reason. The data it sends can be configured using commands sent over a UART connection, but it will arbitrarily ignore these for no discernible reason. This lack of discipline surrounding commands is the main cause of every cycle that takes between 40 and 50 seconds, and without changing the GPS we cannot alter this.

7.4. Cost

The specification shown in Appendix A.1 states that the product should not only be low-power, but also low-cost. To evaluate the success of the project we must compare the cost of the system when scaled from prototype into industrial scale production. The total cost of the prototype was £264.89, a more detailed breakdown of the cost of the prototype is given in section 8.2. For these purposes we will assume

that 'industrial scale' is defined as units of production in the order of hundreds of thousands.

7.4.1. RN2483 (LoRa)

From Microchip Technology Inc, the RN2483 chip has a cost of £6.85 when bought in quantities greater than 5000, although more discounts are available for greater volumes. [57] Extrapolating the available data, when increasing by a factor of 10, the cost reduces by 16.2%. Meaning that for an order of 50,000 pieces we would expect a price per unit of £5.74. Doubling that order quantity we could comfortably estimate a cost of £5.00 per chip.

7.4.2. Air530 (GPS)

The Seeed Studio Air530 GPS comes in at a cost of $11.90 or £8.70. [58] When increasing from 1 unit to 10 units there is a 10% discount available, followed by a 5.61% discount for more than 20 units, and a further 6.32% discount when more than 50 units are purchased. This means that the minimum price shown on the manufacturer's website is £6.95, a 20% discount from the individual price. Extrapolating for an order of 100,000 we can estimate a discount of at least 50%, giving us a final price of £4.35.

7.4.3. nRF52840

The Nordic Semiconductor® nRF52840 chip when purchased from Avnet costs £1.73 for 100,000. [59] This is a significant saving on the Adafruit Feather we purchased, which cost £19.02 per item

7.4.4. Batteries

The batteries used in the project are the CR123A Manganese Dioxide battery, costing £2.53 per unit from RS and a 700mAh battery pack costing £6.86 for more than 5 units. [60] [61] Applying the usual bulk discount of around 40% we can cost the internal battery at £1.52 and the external battery at £4.11.

7.4.5. Comparison

Using the estimated chip and battery costs, we find that the external module will cost around £15.19. This is disregarding enclosures, as these can be manufactured from any material, and as a result cannot be costed effectively until a design and decision has been reached. The internal module will cost £3.25, plus the cost of the PCB, which cannot be known until a design has been sent to a manufacture. Adding the cost of the switch, which can be extrapolated from data to cost £0.40 per item when purchased in bulk, gives a total cost for the internal section of £3.65. [62]

The total cost for all the batteries, chips, and hardware comes to £18.84, adding on the cost of the PCB and enclosure we can conservatively estimate the manufacturing cost to be £25. This is 9.5% of the prototyping cost, and at a 100% profit margin, would retail for £50. This is in line with other LPG level sensing devices such as the GAS IT Mopeka level sensor which retails for £54.99. [6]

As the device is in line with other competing products, and is able to do more than them. We consider this product to be suitably low-cost to meet the specification.

Chapter 8

Project Management

8.1. Team Management

After the initial meeting with our external partner, the project specification document was drawn up. This outlined the overall scope of the project and what tasks would be required to consider the project complete. It became clear that there were three main areas that required focused effort. Those being; Internal Communications, External Communications, and Float Sensing Technology.

8.1.1. Communication

As will be discussed in Section 8.8 normal communication procedures were unavailable. This meant that the usual face-to-face group meetings, both internally and with external partners, were impossible to perform. This lead to the adoption of weekly meetings over Microsoft Teams with our industrial partner and daily meetings through a program called Discord, for Voice Over Internet Protocol (VOIP) calls. This allowed for easy discussion amongst the team members about problems faced, and for discussion related to project direction.
To establish a time and day for the weekly meeting, a schedule was drawn up. It utilised Excel and had the 5 days of the working week as columns, and then each row was a 30 minute block in which everyone could indicate whether they would be available, or occupied. This spreadsheet is shown in Figure 8.1 and resulted in a weekly meeting at 13:00 on a Monday.

8.1.2. Division of Responsibilities

After the initial meeting with Steve Braithwaite and receiving the project brief, the initial task was assigning a Project Manager. This was anonymously voted on by the team and Thomas Whyte-Venables was voted as project manager. The next step was to asses each team member's strengths and weaknesses. Hobbies were also added to the questionnaire as this allowed for people's personal interests to contribute to

---

Matthew Carmichael - 8.7
Thomas Whyte-Venables - 8.1, 8.2, 8.4, 8.5, 8.6, 8.7
William Sparrow - 8.3, 8.7, 8.8

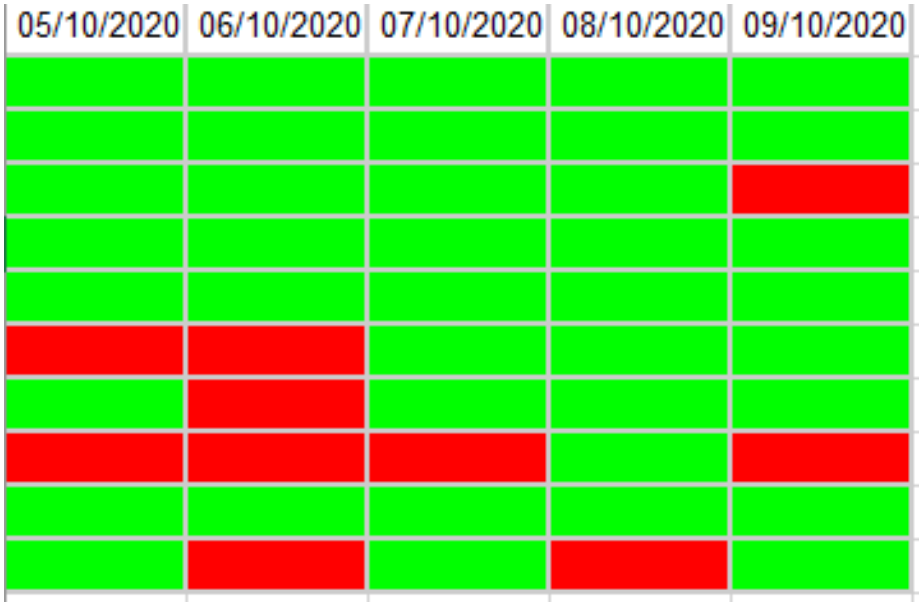


Figure 8.1: The timetable showing group availability each week

the end results. The results of this questionnaire are found in Appendix A.4. These results were then used to decide who would work best in various areas of the project. Tasks were assigned to team members using Microsoft To Do. This allowed for tasks to be assigned to specific people with the ability to add deadlines which were available to all team members. This meant that if a certain area was falling behind, the whole team would be able to see and could offer help to bring the project back up to schedule. The total list of tasks is included in Appendix A.6.

8.1.3. Project Plan

As part of writing the specificiation for this project a plan, was written up. This was done in the form of a Gantt Chart as shown in Appendix A.2.
This plan shows that we aimed to perform all hardware and software based tasks during the university term, and we aimed to have report writing performed through the Christmas break. This is because it did not require in-person interactions, or specialist equipment to perform.
The plan defined a work flow related to sensors and module research, followed by a mass acquisition and integration phase, and ending with testing and review of the hardware and software. Once all of the end information could be gathered, the need for testing or alterations during the report writing stages was removed. This meant that a more cohesive and effective report could be written.

8.1.4. Actual Project Timeline

During the course of the project, details of tasks performed were recorded in the form of a Gantt Chart, as shown in Appendix A.3.
Compared to the predicted progress, as shown in Appendix A.2, a lot of tasks were delayed in both their start and ending. Notable examples are that of "Research of Auxiliary Components" which continued further into the project that anticipated due to issues with Bluetooth® modules, as discussed in Section 4.1.2. Another is that of "Research into Float Material" which took several weeks to complete despite

the team thinking it would be a single day task. Despite all of these delays in starting various aspects of the project the task of report writing was actually only delayed by one day. This attests to the serious effort put in by the team to maintain schedule despite all of the issues that arose.

8.2. Budget

The budget for this project was set at £400. There was a prospective budget suggested in the project brief, a breakdown of which is shown in Table IX.

Table IX: Table of project brief budget guidelines

Item	Budget Associated/£
Off-the-shelf sensors	50
Canister	50
Electronics Modules	80
Misc components	100
Unassigned	120

When compared to our actual budget, as shown in Table X, certain areas were over budget compared to the predicted cost and other areas were significantly under budget.
The most significantly under budget item was that of the cylinder. Due to the nature of LPG cylinders it is nearly impossible to buy a cylinder that is both brand new and empty. As a result, we had to turn to the second-hand market which allowed us to source a second-hand cylinder for 14% of the original budget allocated.
The area which ended up being over budget was related to electronics modules. This was in part due to modules being bought and then considered redundant due to reasons discussed in Section 4.1.2. The breakout boards for the nRF52840 were reasonably expensive due to their rich feature set, which was not required for this project. This, in conjunction with acquiring enough to have redundancy in case of failure, meant that electronics modules ended up being over budget by 22.9%.

Table X: Table of actual budget used

Item	Budget Associated/£
FloGas Cylinder (Used)	7
Seeed Air530	11.18
EMB1061 Bluetooth®	4.9
2.4GHz Antenna	5.5
Microswitches	7.14
RN2483	22.83
Adafruit Feather nRF52840	95.10
CR123A Battery	10.15
Battery Holder	2.11
Battery Contacts	3.43
3D Printer Filament	84.83
Potting Compound	10.72
Total	264.89

At the end of the project there was remaining budget left over from the £400, in total £135.11, leaving 33.8% remaining. Compared to the project brief budget guidelines, assuming the unassigned budget was not designated to be used, the project still came in under budget by 12.6%.

8.3. Software Development

For development of code and file sharing, we used Microsoft OneDrive as an online shared drive mounted to each group member's computer. This allowed us to work collaboratively in Microsoft Office's suite of apps, and also to manage version control and edit history for all other files. Figure 8.2 shows the file system containing both the internal and external version controlled code.

For code development, we implemented the Software Development Life Cycle (SDLC) [55] process to make sure our code was as good as it could be. First we planned our code, using an ASM chart (Figure 5.1, and writing pseudo code so that our purpose and scope was clearly defined.
Next, we began the 'create' stage by deciding on our programming language, the Arduino variant of C/C++, and deciding how each individual chip will communicate with each other, via UART. The final section of 'create' was to begin to solve problems that we had anticipated through planning.
Once a vague framework had been created we began the 'developing' stage of the SDLC which was where we wrote the code, this was the longest stage of the process. Once the code had been written and compartmentalised into functions, .cpp and .h files, we progressed to the 'testing' stage where we began to identify, categorise, and minimise the number of glitches and bugs in the system.
The final stage of the SDLC is 'deployment' which is not a relevant stage for this project as there is no end-user to deliver the finished product to. We have created a bench protoype and will be simply handing all files to ASH Wireless.

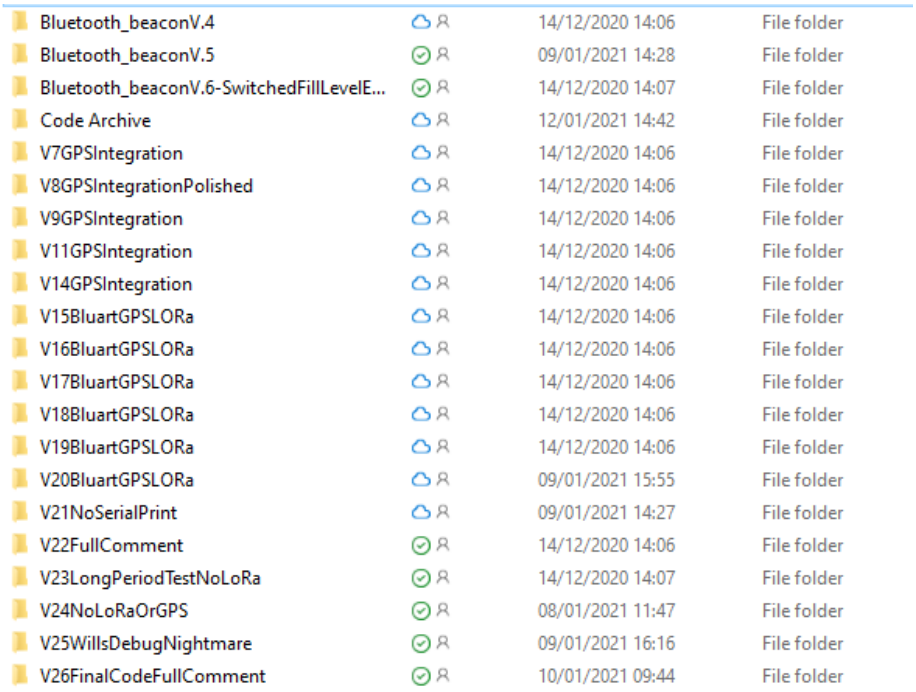


Figure 8.2: The file system of version controlled code

8.4. Hardware Development

For hardware development of the float, all CAD work was done in Autodesk® Fusion 360 which has full feature sets, free for students. As part of this it supports cloud saving version control which allows multiple users to access the file simultaneously. Each part for the design was created in an individual file and then included into a final build. This allowed for minor design changes to be made for individual components, which were then updated in the final design.
Once the design was finalised, most of the development was performed using a 3D printer as this allowed for quick prototyping of the design. A major part of planning for this build centered around the access to a team member's privately owned 3D printer. This meant that hardware development could continue even during the Christmas break, and when the University closed due to COVID-19 progress wasn't hindered.

8.5. Risk Analysis

To determine risks that could affect the progress of the project, a risk analysis was performed as shown in Table XI. This shows the possible risks, their likelihood of occurence, and how severe these could be in preventing further progress of the project. This was then used to calculate a Risk Exposure Score and those with higher scores were taken more seriously than the lower scored risks. These risks then had mitigation procedures created for them such that if an issue occurred it was easy to counteract the effects.

8.6. Stakeholder Analysis

As part of identifying project requirements, a stakeholder analysis was perfomed and the results of this are shown in Table XII.

Table XII: Table of stakeholders

Stakeholder Type	Stakeholder	Description
Primary	LPG Provider	Company responsible for the rental of LPG in areas where the system is going to be implemented.
Primary	Customer	Individual who will be renting the LPG for the domestic use.
Secondary	Logistics Company	Company responsible for delivering full canisters to consumers and collection of empty canisters for return to the refilling depot.
Secondary	The Things Network	Company who provides LoRaWAN support and wider integration in the World Wide Web.

This shows that for our purposes we were mostly concerned with the needs of the LPG provider and the customer, as these are the ones mostly likely to be directly affected by the design decisions we made.

8.7. Group Contribution

8.7.1. Research

Table XIII shows each item of research conducted by the team. Unless *even split* is listed below the item, the list of names is in order of contribution amount.

Table XIII: Table of research contribution

Item	Contributor
Power	Matthew Carmichael
Location Tracking	Thomas Whyte-Venables
Long-Range Communications	William Sparrow
LoRa	William Sparrow Thomas-Whyte-Venables
Short-Range Communications	William Sparrow
Bluetooth Module Options *(even split)*	Matthew Carmichael Thomas Whyte-Venables William Sparrow
Bluetooth	Matthew Carmichael Zhe Koh
Microcontroller *(even split)*	Matthew Carmichael Thomas Whyte-Venables William Sparrow
Cloud Data Processing	William Sparrow
Float Material	Pragash Balachandran
Sensor *(even split)*	Matthew Carmichael Pragash Balachandran Thomas Whyte-Venables William Sparrow Zhe Koh

8.7.2. Software

Table XIV shows the different software aspects of the product that have been created by the group. The names are listed in order of contribution amount.

Table XIV: Table of software contribution

Item	Contributor
LoRa	William Sparrow
Air530	Thomas Whyte-Venables William Sparrow
Air530 Datasheet Translation	Zhe Koh
Bluetooth	Matthew Carmichael Zhe Koh William Sparrow Thomas Whyte-Venables
Cloud	William Sparrow
Integration	William Sparrow Thomas Whyte-Venables
Testing	William Sparrow Thomas Whyte-Venables Matthew Carmichael

8.7.3. Hardware

Table XV shows the different software aspects of the product that have been created by the group. The names are listed in order of contribution amount.

Table XV: Table of hardware contribution

Item	Contributor
Soldering of Components	Thomas Whyte-Venables
Float Design and Build	Thomas Whyte-Venables
Float Testing	Thomas Whyte-Venables
Power and Battery	Matthew Carmichael
PCB Design	Zhe Koh Pragash Balachandran
Testing	Thomas Whyte-Venables William Sparrow
Testing Data Processing	William Sparrow Zhe Koh

8.7.4. Report

Individual contributions for the report are labelled in the footers with referenced sections next to the name of the contributors for each part.
If the reference is the highest level of section, that means that that person contributed on every subsection included by that section. If someone contributed on a single subsection but another person for the whole of the section, the person who contributed only a single subsection will only have the subsection attributed to them.
The LaTeX formatting and compiling was performed by Thomas Whyte-Venables and William Sparrow. The final spelling and grammar check in the report was conducted by; Matthew Carmichael, Thomas Whyte-Venables, William Sparrow, and Zhe Koh.

8.8. COVID-19

There are obvious issues that will be present in a project that is started and finished in varying severities of a global pandemic and the accompanying national and local lockdowns. From issues with delivery and supply chains, to the group being unable to meet in person, the project was severely impacted. Here we will attempt to qualify the problems we encountered, and the work-arounds and solutions we implemented.

8.8.1. Communications and Meetings

In September 2020, the University of Southampton, as mandated by the British government, reduced its on campus teaching and contact to a minimum. This impacted GDP groups who were told that meeting in person would not be allowed. Instead, other methods of communication were required. As a group we decided to use Discord, a free-to-use instant messenger that supports text, voice, and video messaging, for our communication between group members. [63] Although it was not a suitable replacement for face-to-face communication, it did allow for instant contact with all members of the group provided that the group members were online

and logged in to their account. This additional step did mean that we encountered issues surrounding group members not seeing messages in a timely fashion.
For 'in-person' communication with members of staff inside the University of Southampton, and with the external partner, we used Microsoft Teams for its video conferencing capabilities. [64] We scheduled a weekly meeting with our industrial partner to replace the normal face to face supervisor meetings that would normally have occured. These meetings served as a useful anchor in the week and allowed us to update the industrial partner on the progress we had made, and the challenges we had encountered. Where Teams fell short is the lack of in-person social pressure for group members that contributed less than others. We struggled to impress the seriousness of the lack of contribution from certain group members over a video call. If the meetings had been in person it would have been more easy to convey just how little they had contributed to the project, and the impact it was having on the amount of work being completed.
Where messages needed to be received immediately and Discord was not an option, for example if a group member did not have access to their computer, then we used Facebook Messenger to communicate. This was a rare occurence and was only used when absolutely necessary. We did not use Facebook for voice and video calls, as these were scheduled so there was no need for fast, instant communication.
For written communication with members of staff inside the University of Southampton and with the external partner, we used email. This allows for accountability and record keeping in the event of a dispute, and is the only form of communication used universally inside the professional world.
To manage the assignment of tasks, whereas in a normal year these would be assigned verbally and then checked on in the face-to-face meetings, we used Microsoft To Do. This is an online task assignment tool that synchronises to our university accounts. We encountered issues with this as certain members of the group did not check the list of tasks, and also did not tick off their tasks when they had completed them. This meant that there was a desynchronisation between task completion, and other team members knowing that they had been completed. This disconnect was only resolved at out weekly meetings. If we were to do this again, we would look to manually record task assignment at the project manager level, and check up on team members more frequently to manage their progress.

8.8.2. Software Development

Developing code was always going to be a challenge for us. Not only because of the lack of Computer Science students in our group, but also because integrating code that has been written by multiple people is a challenge when the authors are in the same room together. It is more of a challenge when it has to be done by one person on their own. The initial split of code development was that the Bluetooth® communication would be handled by one person, the LoRa would be handled by another, and the GPS would be handled by a third. In reality, the Bluetooth® was done by one person, and then the GPS and LoRa were programmed by a group. When it came to displaying the data in an application, we decided to handle that when we had everything else done.

Once the individual aspects of the design had been programmed, we had to integrate

all the systems. This would have been a challenge over Discord, or Teams, but we were able to spend some time in the Building 16 computer laboratory. This allowed us to work within earshot of each other, and was instrumental to the completion of this project. The idea for using Ubidots was also developed in Building 16, and without it we would not have been able to display the data received into The Things Network.

8.8.3. Hardware Development

The development of hardware without access to the University's resources is a significant challenge. We realised that if the country went into another lockdown, as it had already done in March 2020, that we would be unable to complete a prototype of the design. As a result of this we made sure to access Building 16 throughout December 2020 and were able to solder all the external components onto one piece of stripboard before the laboratory was closed as a result of the January 2021 lockdown. Because we knew that a team member had access to a 3D printer outside of the University, we were able to plan around the lockdown and could focus on more important aspects of the project, secure in the knowledge that we could still complete the physical components of the project external to University facilities if necessary. As it happened, we did need to use this facility, as the Building 16 lab was closed from Christmas until the project deadline.

Collaboratively developing hardware remotely is not something that is usually done. As a work-around for this issue we made use of Microsoft OneDrive and the files for the CAD drawings of the internal system were made available for anyone to work on at any time. Using version control we were able to keep track of changes and monitor for any changes that were incorrect. Upon finding a problem we could simply roll back to an earlier version without losing everything.

Table XI: Table of risk analysis for the project

Risk	Probability (1, low - 5, high)	Severity (1, low - 5, high)	Risk Exposure (E = P×S)	Mitigation
Permanent loss of team member	1	4	4	Speak with supervisor for advice and arrange to redistribute tasks amongst the team.
Lack of contribution due to illness	3	3	9	Illness severity dictates process for mitigation as work may need to be redistributed.
Project manager not assigning work	1	4	4	Method of keeping the project manager accountable to minimise the impact of them not working by changing to different team member.
Issues with communication due to COVID-19	4	2	8	Establish multiple methods of contacting each other i.e Discord, Facebook, Teams, and Email.
Loss of data	2	5	10	Use Microsoft OneDrive as part of the university as these provide cloud backups whenever a file revision is made. It also has the facility for restoring deleted files.
Issues with team supervisor	1	5	5	This is an area that needs to be discussed with module leader and potentially course leader
Logistics issues relating to COVID-19	2	4	8	Plan to order any components as early in the timeline as possible to minimise issues caused by delays.
Hardware failure affecting the project	2	4	8	Order enough components to have redundancy such that if components break there are replacements to hand.
Failure of team to understand core concepts	2	3	6	Contact industrial partner at ASH Wireless and ask questions for maximum clarification

Chapter 9

Conclusion

9.1. Proposal of further work and ideas

The final product of this project is a bench prototype capable of using float sensor technology to determine whether a cylinder of LPG is full or empty. To further this project and create an end product there are some factors that can be changed to develop this bench prototype further.

9.1.1. PCB

One of the main issues with the bench prototype is that the current microcontroller used has a width of 23mm which will not be able to fit within the valve aperture. However, if this product were to be mass produced, a custom designed PCB could be manufactured using the same Nordic Semiconductor® nRF52840 chip with only the necessary I/O pins traced to minimise the size of the PCB. Further information about the proposed PCB can be found in section 5.3.2.3.

9.1.2. GPS

This product has multiple elements that have to collaborate together to culminate in the final bench prototype. One key element of the system is the GPS which is necessary to give a location of the cylinder. This information is used by delivery drivers to exchange LPG cylinders with customers. The GPS module used was the Air530 which was able to function for the majority of the time. However, while testing the system as a whole, it became apparent that the GPS did not always comply with the code that was written. In the cases where the GPS got stuck in places, the code sent the GPS instructions to reconfigure itself. However, this did not always fix the issue, hence one of the major differences that would be made in this project would be to use a different GPS device, one with greater support.

Matthew Carmichael - 9.2,9.1.1, 9.1.2
William Sparrow - 9.1.3

9.1.3. TTN and Ubidots

The current method of displaying data from the cylinder is suitable for a domestic end user only. For management on an industrial scale, an Structured Query Language (SQL) database would be more suitable, as it allows for interaction with other programs and interfaces while also managing and storing lots of data. The Things Network has an integration that stores data for seven days and makes it available for querying using a Representational State Transfer (REST) Application Programming Interface (API). By using this integration and connecting it to an SQL database, multiple devices can be managed and displayed on a custom interface.
The custom interface would seek to replace the Ubidots interface, meaning it would need to display the fill level of the cylinder, and the location sent by the GPS. When the cylinder became empty, the system would need to send an alert, and notify a refill team with the cylinder's location. All of this can be accomplished at an industrial scale, but not for an individual prototype. Given more time, we could have developed the SQL integration and demonstrated a working prototype of the industrial system, but delays from COVID-19, and the lack of a Computer Science student meant that we were unable to achieve this.

9.2. Summary

This project investigates various approaches to be able to sense a liquid level inside a sealed, pressurised, and portable LPG cylinder. Previous methods were explored and the most cost and power effective methods were down selected. This project has offered an alternative to the method that rental companies currently use to retrieve and replace LPG cylinders. It has been achieved by rendering the need for the customer to call the rental company redundant. This will have a positive economic, environmental, and societal impact on the company and the local community as stated in Section 2.1.

Group 43 were able to produce a functional bench prototype of an LPG level sensor. This has been achieved using float technology to give an accurate point liquid level within the LPG cylinder. The system was split into internal and external modules. The internal module was used to read the state of the switch and send it to the external module. The external module was used to gain the GPS co-ordinates of the cylinder and send this data via LoRaWAN to a web application hosted by Ubidots. By undergoing various tests, the connectivity via Bluetooth® was verified as stable, and the system was identified as robust.
The product was able to successfully fulfil the specification set out in Appendix A.1. The bench prototype has achieved this by appropriately sizing the correct batteries to last for the 6 week to 6 month duration, as well as using as many off the shelf components as possible. This makes this product as reproducible and low cost as possible, as well as making the product reusable for multiple rental periods by using rechargeable batteries.
The objectives that were initially set out at the start of the project have also been fulfilled. The research and down selection of the main sensor that is used to track the level of the LPG inside the cylinder encompasses a wide range of different sensing

technologies to establish the most feasible technology for this project.
Off the shelf auxiliary components such as the microcontroller were selected, and integrated with the down selected float sensor. This was implemented as part of the internal module. A second subsystem, which included the GPS and LoRaWAN modules, was interfaced with the microcontroller to form the external module. By combining these two subsystems, the bench prototype was able to function as an LPG level sensor for man carriable canisters.
This project has fulfilled the basic specification and has extended to some of the stretch goals. However, this bench prototype can be developed further and improved by introducing the changes suggested in further work in Section 9.1. By implementing these changes, this project would be able to function as an end product.
The team was able to work towards the final product by setting communication channels early in the project. This was crucial to being able to work effectively to complete this project. Additionally, weekly meetings with the customer helped to ensure that the team adhered to the customer's needs. By working in an agile configuration, the members of the group could help each other on different tasks.
Overall, this project has enhanced the group's understanding and experience in the field of wireless connectivity. The team believes that this project has accomplished the problem statement and has proved fruitful for ASH Wireless as a customer.

Bibliography

[1] "Wood smoke and your health," Dec 2020. [Online]. Available: https://www.epa.gov/burnwise/wood-smoke-and-your-health

[2] G. Shen, W. Preston, S. M. Ebersviller, C. Williams, J. W. Faircloth, J. J. Jetter, and M. D. Hays, "Polycyclic aromatic hydrocarbons in fine particulate matter emitted from burning kerosene, liquid petroleum gas, and wood fuels in household cookstoves," *energy & fuels*, vol. 2017, no. 31, pp. 3081–3090, Jan 2017.

[3] S. C. Anenberg, K. Balakrishnan, J. Jetter, O. Masera, S. Mehta, J. Moss, and V. Ramanathan, "Cleaner cooking solutions to achieve health, climate, and economic cobenefits," *Environ. Sci. Technol*, vol. 2013, no. 47, pp. 3944–3952, April 2013.

[4] R. Van Leeuwen, A. Evans, and B. Hyseni, "Increasing the use of liquified petroleum gas in cooking in developing countries," *Livewire*, vol. 2017, no. 74, 2017. [Online]. Available: http://hdl.handle.net/10986/26569

[5] "6200 series, four bolt gauge for domestic tanks, dot cylinders and motor fuel," May 2018. [Online]. Available: https://rochestergauges.com/product/6200/

[6] "Single gas level bluetooth ultrasonic level sensor." Jan 2021. [Online]. Available: https://www.gasit.co.uk/single-gas-level-bluetooth-ultrasonic-gas-level-sensor.html

[7]

[8] "Internet of things (iot)," Jan 2021. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

[9] "Corporations unite against hackers," Feb 2018. [Online]. Available: https://www.faz.net/aktuell/wirtschaft/digitec/grosse-internationale-allianz-gegen-cyber-attacken-15451953-p2.html#pageIndex_1

[10] "Lora alliance," Jan 2021. [Online]. Available: https://lora-alliance.org/about-lorawan

[11] "Sigfox," Jan 2021. [Online]. Available: https://www.sigfox.com/en

[12] "The things indoor lorawan wifi gateway based on sx1308 – us915," Jan 2021. [Online]. Available: https://www.amazon.com/dp/B08L6BWNJR

[13] "Platforms for smart building services," May 2018. [Online]. Available: https://www.bluetooth.com/blog/platforms-for-smart-building-services/

[14] "Bluetooth & btle," Nov 2017. [Online]. Available: https://learn.adafruit.com/alltheiot-transports/bluetooth-btle

[15] "Nfc," April 2018. [Online]. Available: https://techterms.com/definition/nfc

[16] "Liquefied petroleum gas(lpg)," Jan 2021. [Online]. Available: https://www.calor.co.uk/lpg

[17] "Propane - vapor pressure," Jan 2021. [Online]. Available: https://www.engineeringtoolbox.com/propane-vapor-pressure-d_1020.html

[18] "How do i know if my cylinder is empty?" Jan 2021. [Online]. Available: https://www.flogas.co.uk/frequently-asked-questions/how-do-i-know-if-my-cylinder-is-empty

[19] "Our technology," Jan 2021. [Online]. Available: https://www.hexagonragasco.com/about/our-technology

[20] "Product benefits," Jan 2021. [Online]. Available: https://www.hexagonragasco.com/products/prodcuts-benefits/product-benefits

[21] "Introduction to calor cylinders, valves, regulators, hoses & safety devices," Jan 2021. [Online]. Available: http://www.adamsgas.co.uk/wp-content/uploads/2016/12/1-Introduction-to-Calor-Cylinders-Valves-Regulators-Hoses-Safety-Devices-1of3.pdf

[22] "Vibrating tuning fork liquid level switches," Dec 2020. [Online]. Available: https://www.sensorone.com/vibrating-tuning-fork-liquid-level-switches/

[23] "Model ctf mini tuning fork level switch," Dec 2020. [Online]. Available: https://www.dwyer-inst.co.uk/Product/Level/LevelSwitches/TuningFork/ModelCTF#specs

[24] "Vega vegaswing 51 series, tuning fork horizontal mounting level switch pnp output," Dec 2020. [Online]. Available: https://uk.rs-online.com/web/p/level-sensors/4936802

[25] "Siemens capacitance level probe relay output," Dec 2020. [Online]. Available: https://uk.rs-online.com/web/p/level-sensors/5132427/

[26] "Non-intrusive rf capacitance sensors," Dec 2020. [Online]. Available: https://www.omega.co.uk/pptst/LVP51.html

[27] "Gems sensors electro optic horizontal, vertical mounting level probe," Dec 2020. [Online]. Available: https://uk.rs-online.com/web/p/level-sensors/6163028/

[28] "Optical liquid level sensor- external m12 mount," Dec 2020. [Online]. Available: https://www.mouser.co.uk/datasheet/2/657/cynergy3_ols2_v3-1892296.pdf

[29] "Optomax basic range of liquid level switches," Dec 2020. [Online]. Available: https://sstsensing.com/product/optomax-basic-range-of-liquid-level-switches/

[30] "2048 series miniature float switch, cable type," Jan 2021. [Online]. Available: https://www.levelsensorsolutions.com/2048-series-miniature-float-switch-cable-type-c-21_8_35.html

[31] "Guided level radar," Jan 2021. [Online]. Available: https://www.pepperl-fuchs.com/great_britain/en/classid_493.htm

[32] "Microwave level sensor," Jan 2021. [Online]. Available: https://uk.rs-online.com/web/p/level-sensors/0488037/

[33] "Pressure level sensor," Jan 2021. [Online]. Available: https://uk.rs-online.com/web/p/level-sensors/8044099/

[34] "Ultrasonic liquid level sensors," Jan 2021. [Online]. Available: https://www.sensorsone.com/ultrasonic-liquid-level-sensors/

[35] "Optical liquid level sensors," Jan 2021. [Online]. Available: https://www.smdsensors.com/blog-optical-liquid-level-sensor/

[36] "Pro's and con's of common level sensors," Jan 2021. [Online]. Available: https://automationforum.in/t/types-and-pros-cons-and-application-of-common-level-sensors/5486

[37] "Liquid level sensing technologies," Jan 2021. [Online]. Available: https://www.gemssensors.co.uk/~/media/files/resources/na_english/whitepaper/whitepaper_liquidlevelsensingtechnologies.pdf

[38] "Lpg specification," Jan 2021. [Online]. Available: https://iocl.com/products/LPGSpecifications.pdf

[39] "float sensors," Jan 2021. [Online]. Available: http://float-sensor.net/floattypelevelswitch/float/

[40] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee 802.15.4," *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, vol. 2017, Jan 2012.

[41] "Emb1061," Jan 2021. [Online]. Available: https://files.seeedstudio.com/products/113990637/res/DS0080EN_EMB1061_V1.3.pdf

[42] "nrf52840," Jan 2021. [Online]. Available: https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840/GetStarted

[43] "Beidou satellite system," Jan 2021. [Online]. Available: http://en.beidou.gov.cn/

[44] "Glonass," Jan 2021. [Online]. Available: https://www.glonass-iac.ru/en/

[45] "Galileo satellite system," Jan 2021. [Online]. Available: https://www.gsa.europa.eu/

[46] "Seeed studio grove-gps (air530)," Jan 2021. [Online]. Available: https://www.mouser.co.uk/new/seeed-studio/seeedstudio-grove-gps-air530-module/

[47] "What is gprs (general packet radio services)?" May 2007. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/GPRS

[48] "Arduino sim, the new cellular connectivity service for the arduino iot cloud," May 2019. [Online]. Available: https://blog.arduino.cc/2019/05/21/arduino-sim-the-new-cellular-connectivity-service-for-the-arduino-iot-cloud/

[49] "Rn2483 chip," Jan 2021. [Online]. Available: https://www.microchip.com/wwwproducts/en/RN2483

[50] "Lora network structure," Jan 2021. [Online]. Available: https://www.researchgate.net/profile/Thomas_Heide_Clausen/publication/307965130/figure/fig1/AS:404728832905216@1473506281969/LoRa-network-architecture.png

[51] "The things network," Jan 2021. [Online]. Available: https://thethingsnetwork.org

[52] "Cayenne lpp," Jan 2021. [Online]. Available: https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload

[53] "Ifttt," Jan 2021. [Online]. Available: https://ifttt.com/home

[54] "Ubidots," Jan 2021. [Online]. Available: https://ubidots.com/

[55] "Software development life cycle," Jan 2021. [Online]. Available: https://blog.eccouncil.org/5-phases-of-the-secure-software-development-life-cycle-sdlc/

[56] "Raytac mdbt50q," Jan 2021. [Online]. Available: https://cdn-learn.adafruit.com/assets/assets/000/068/544/original/Raytac_MDBT50Q.pdf?1546346679

[57] "Microchip technology inc. - rn2483," Jan 2021. [Online]. Available: https://www.microchip.com/wwwproducts/en/RN2483

[58] "Seeed studio grove-gps (air530)," Jan 2021. [Online]. Available: https://www.seeedstudio.com/Grove-GPS-Air530-p-4584.html

[59] "Avnet nrf5240," Jan 2021. [Online]. Available: https://www.avnet.com/shop/us/products/nordic-semiconductor/nrf52840-qiaa-t-30744573456355206 76?CMP=EMA_OEMSecrets_inventoryfeed_VSE

[60] "Rs components -cr123a," Jan 2021. [Online]. Available: https://uk.rs-online.com/web/p/speciality-size-batteries/8010711/

[61] “Rs components -700mah pack,” Jan 2021. [Online]. Available: https://uk.rs-online.com/web/p/rechargeable-battery-packs/1769367/

[62] “Omron switch d2aw,” Jan 2021. [Online]. Available: https://www.mouser.co.uk/ProductDetail/Omron-Electronics/D2AW-EL003H/

[63] “Discord,” Jan 2021. [Online]. Available: https://discord.com/

[64] “Microsoft teams,” Jan 2021. [Online]. Available: https://www.microsoft.com/en-gb/microsoft-365/microsoft-teams/group-chat-software

Appendix

The Appendices contain files considered too large for the main report. This includes, but is not limited to, the Gantt Chart, the project specification, and the full archive of written code.

A.1. Project Specification and Project Plan

School of Electronics and Computer Science
ELEC6200 MEng Group Design Project

Project Specification and Plan

GDP Number and Title: 43 Low cost level sensing for domestic LPG canisters

Academic Supervisor(s): Steve Gunn

Team Members: Mathew Carmichael, Pragash Balachandran, Thomas Whyte-Venables, William Sparrow, Zhe Koh

External Partner (including affiliation): ASH Wireless

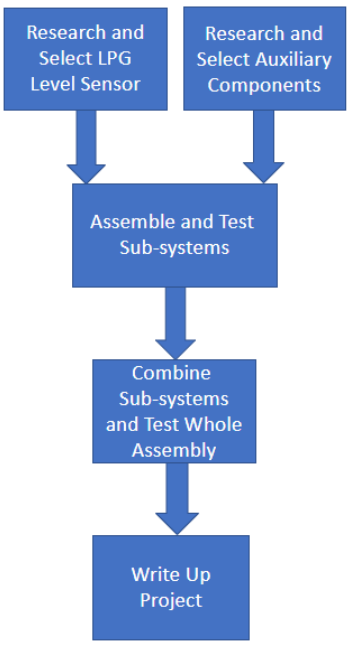
Project Specification:

- A battery powered sensor to report fill state and location of a man carriable gas canister.
- The product will report to a database weekly and will check the fill level daily.
- When the level is low the product will report to the database.
- The expected use time will be between 6-24 weeks, in a discontinuous manner.
- The product should use as many 'off the shelf' components as is practical, in order to minimise the cost and production time incurred by the company implementing the system.
- The product needs to be low power for extended use time with a low cost for later large-scale rollout to market.
- The product will be a bench prototype exploring the viability of the system in a real-world application.
- The product should be reusable and rechargeable when the gas canister is refilled.
- The functional size should be minimised to avoid complications related to environment.

Project Plan:

Objectives:

- Research all the current level sensing technologies.
- Down select the three most applicable technologies and conduct further research.
- Source the sensor with the greatest number of positive attributes.
- Research auxiliary componentry to be used for the total system.
- Interface the level sensor with the 'off the shelf' microcontroller.
- Interface the Global Positioning System (GPS) sensor with the 'off the shelf' microcontroller.
- Use the microcontroller to send data using LoRa network.
- Display the data on an external device.
- Combine all the subsystems into a working system.



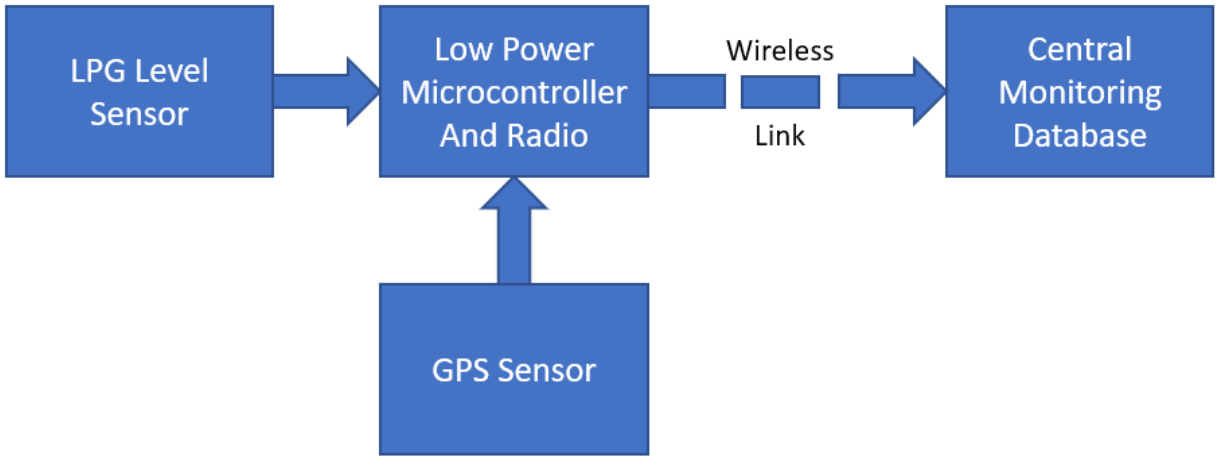
Stretch Objectives:

- Minimise the power wasted on unnecessary functions in the microcontroller.
- Implement accelerometer-based interrupts for triggering GPS reporting.
- Design and create housings.
- Implement method of switching programming between steel and composite canisters so both can be used without flashing of different software.
- Create a bespoke web interface.
- Develop a method of recharging when out on deployment.
- Report the data to a custom database.

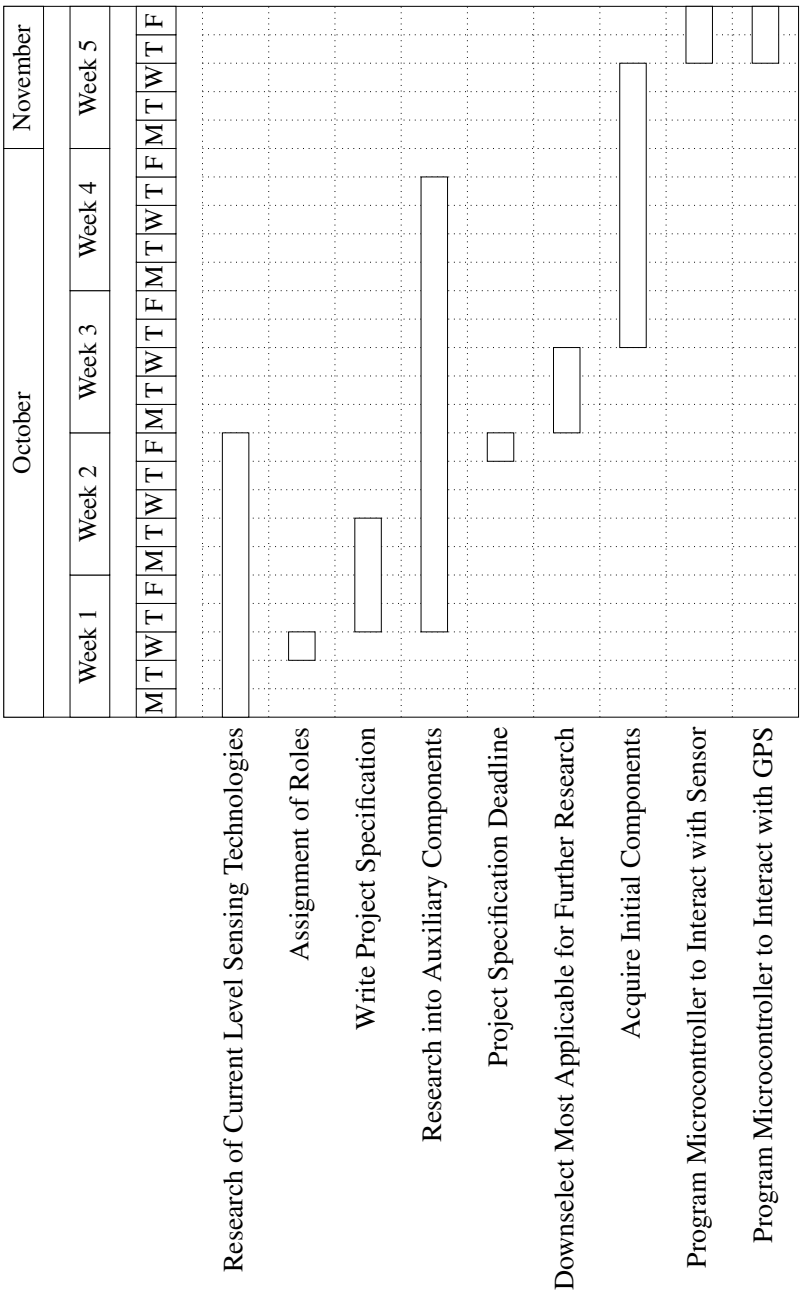
Deliverables:

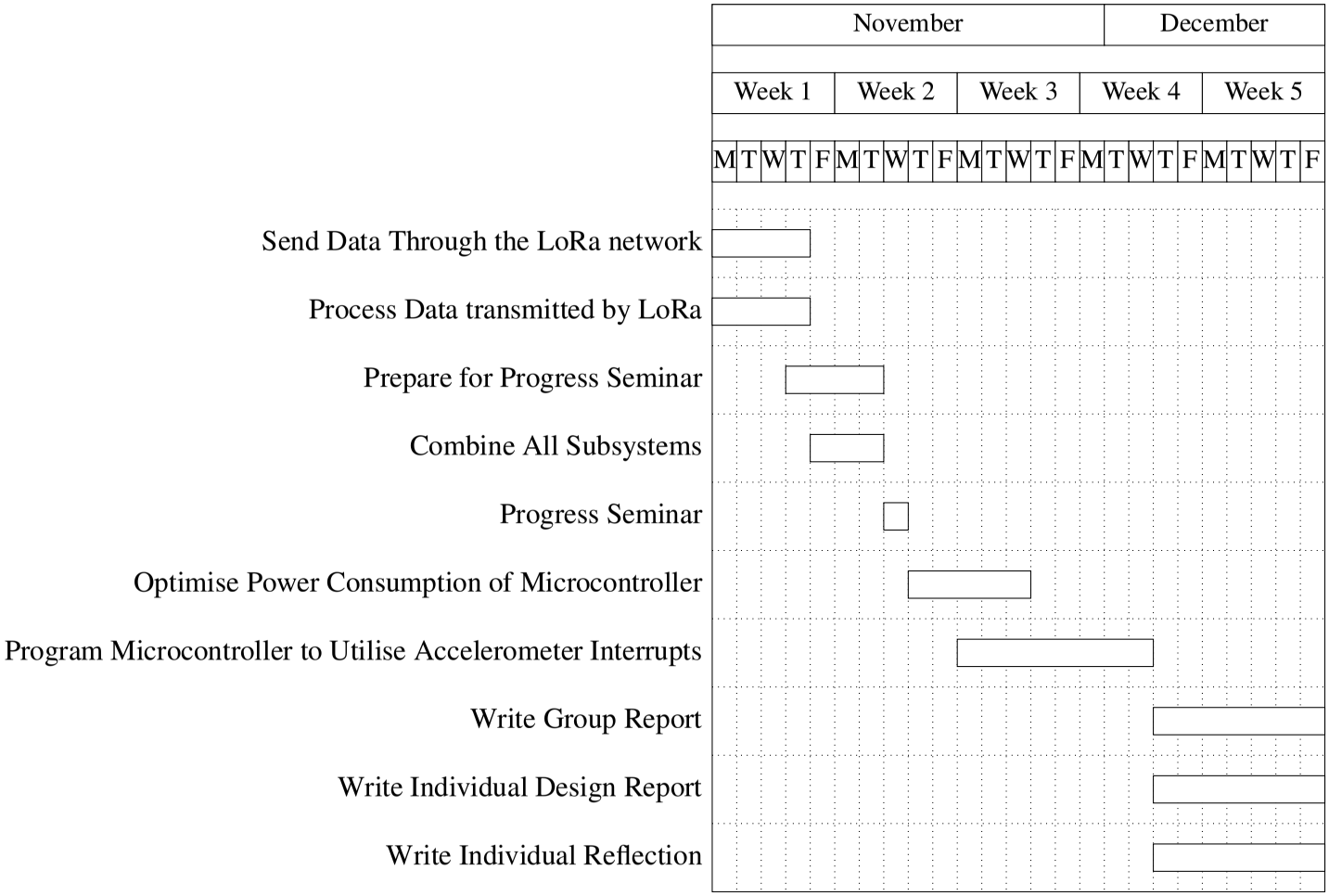
- Project Specification
- Project Plan
- Progress Presentation
- Project Report

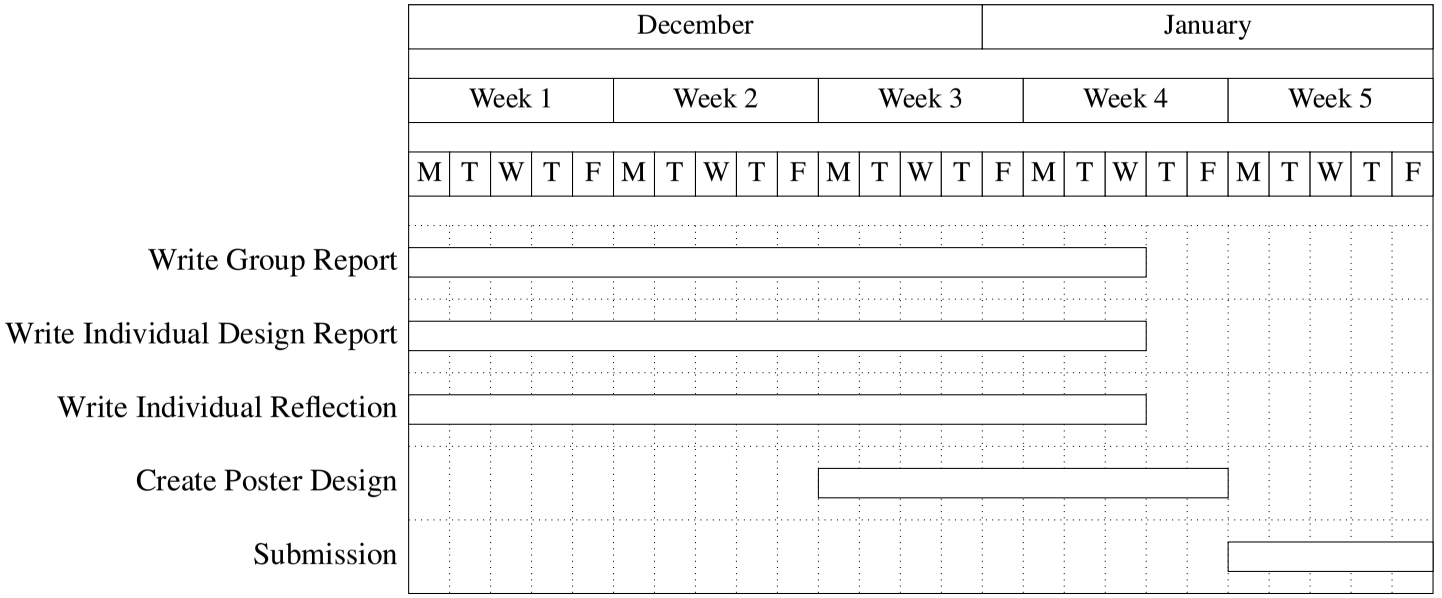
Subsystem Block Diagram:



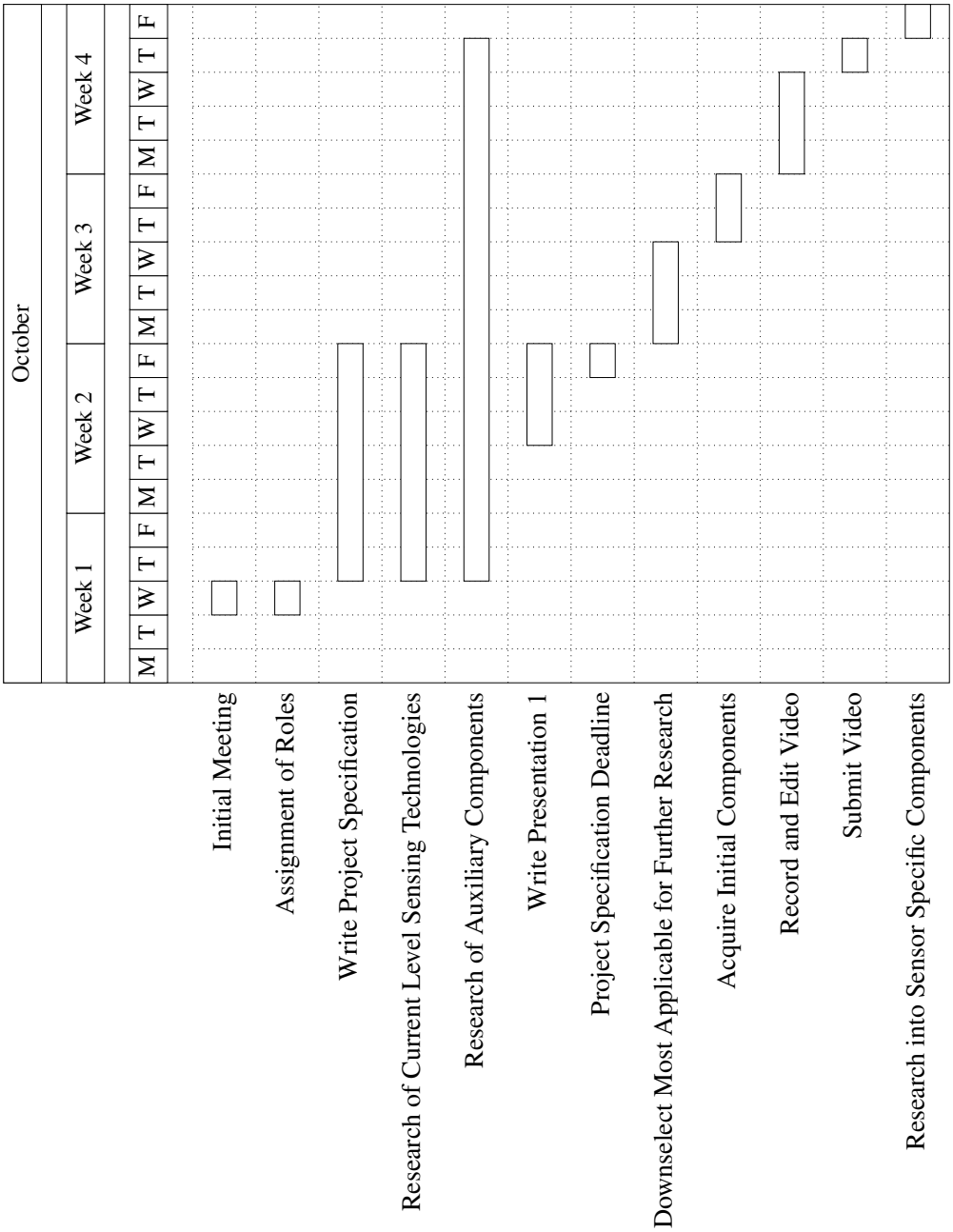
A.2. Proposed Timeline

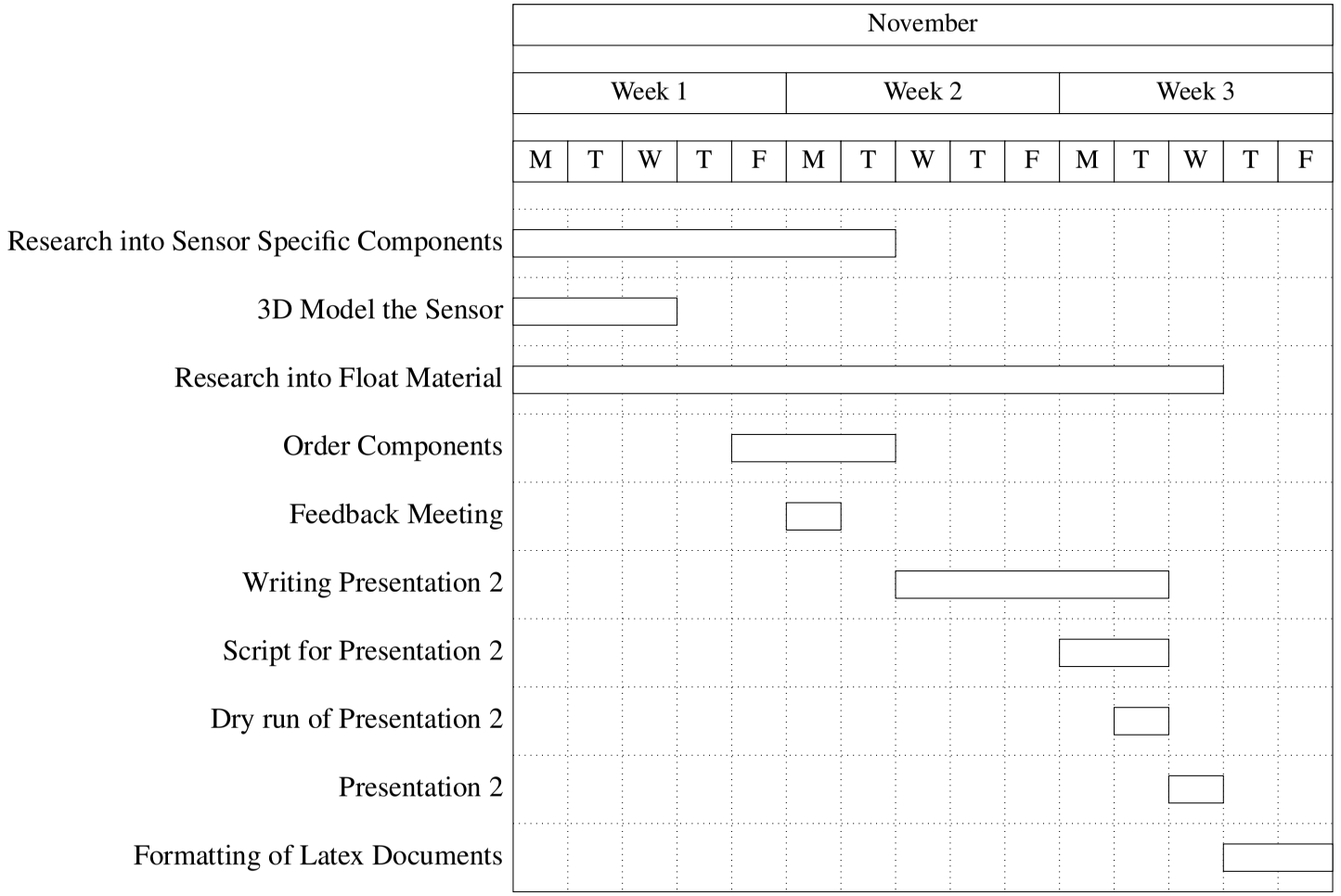


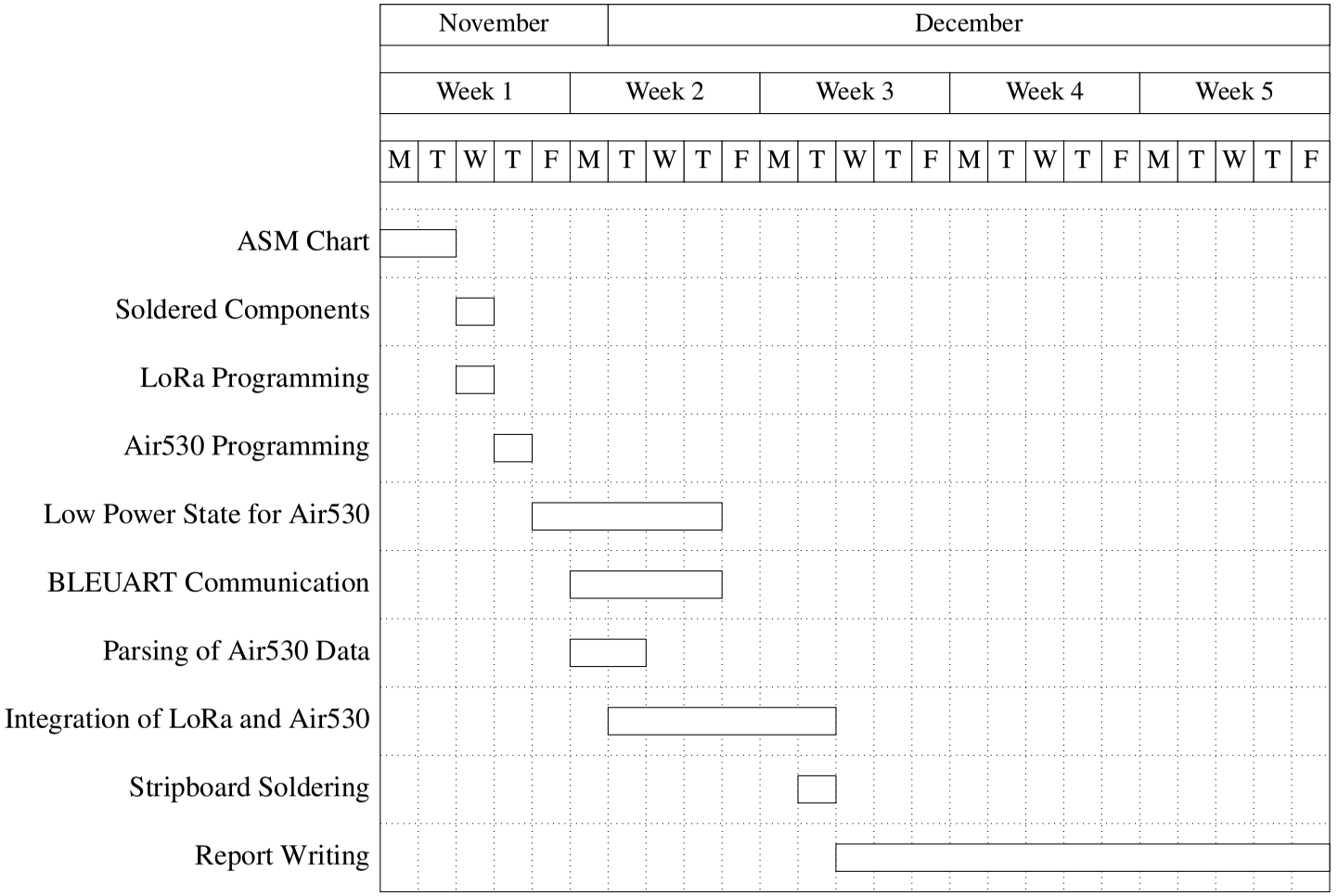

November
December
Week 1
Week 2
Week 3
Week 4
Week 5
M T W T F M T W T F M T W T F M T W T F M T W T F
Send Data Through the LoRa network
Process Data transmitted by LoRa
Prepare for Progress Seminar
Combine All Subsystems
Progress Seminar
Optimise Power Consumption of Microcontroller
Program Microcontroller to Utilise Accelerometer Interrupts
Write Group Report
Write Individual Design Report
Write Individual Reflection

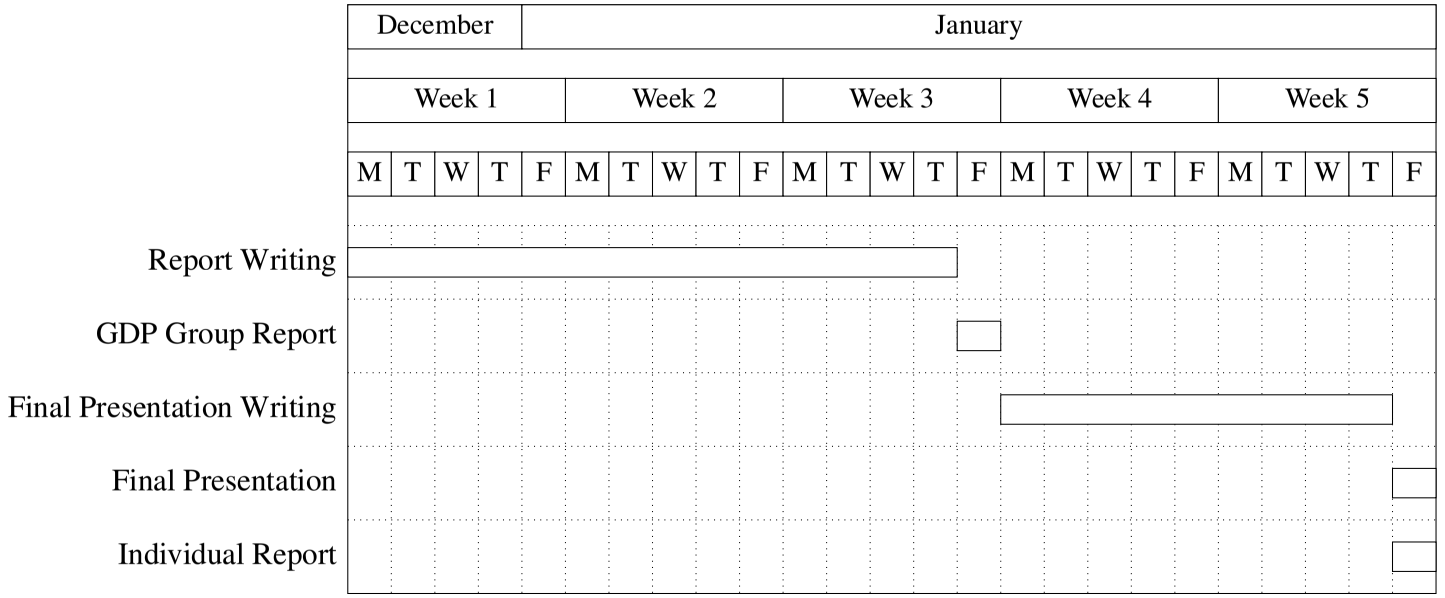

December
January
Week 1
Week 2
Week 3
Week 4
Week 5
M T W T F M T W T F M T W T F M T W T F M T W T F
Write Group Report
Write Individual Design Report
Write Individual Reflection
Create Poster Design
Submission

A.3. Actual Timeline




November
Week 1
Week 2
Week 3
M T W T F M T W T F M T W T F
Research into Sensor Specific Components
3D Model the Sensor
Research into Float Material
Order Components
Feedback Meeting
Writing Presentation 2
Script for Presentation 2
Dry run of Presentation 2
Presentation 2
Formatting of Latex Documents


November
December
Week 1
Week 2
Week 3
Week 4
Week 5
M T W T F M T W T F M T W T F M T W T F M T W T F
ASM Chart
Soldered Components
LoRa Programming
Air530 Programming
Low Power State for Air530
BLEUART Communication
Parsing of Air530 Data
Integration of LoRa and Air530
Stripboard Soldering
Report Writing


December
January
Week 1
Week 2
Week 3
Week 4
Week 5
M T W T F M T W T F M T W T F M T W T F M T W T F
Report Writing
GDP Group Report
Final Presentation Writing
Final Presentation
Individual Report

A.4. Skills Assessment

Table 1: Table showing the skills assessment conducted at the start of the project

Name	Degree Title	Strengths	Weaknesses	Hobbies
Matthew Carmichael	Electrical Engineering	Good knowledge of arduino and knowledge of bluetooth modules. I have some experiance with bending and forming using aluminium sheets. Always on time and will always make sure that the work is done to a good standard before submission.	Organisation, presentation skills can be lacking sometimes. Slow writer	I enjoy tinkering with an RC robot of my own design
Pragash Balanchandran	Electronic Engineering	Research, hardware integration	Microcontrollers, Software-end	Designing stuff, art and stuff
Thomas Whyte-Venables	Electronic Engineering	CAD, hardware based assembly, research, project management, hardware design	Programming	Machining and 3D Printing
William Sparrow	Electrical and Electronic Engineering	Manufacture, Web Design (HTML CSS JS), Arduino (kinda), some experience with communication over local WiFi networks between Arduino and a web page. Idk if it's a skill but I'm good at bodging stuff to work. Also good at bullshitting and public speaking	Complex Circuit Design, LaTeX	Whitewater kayaking - like a lot, Video editing and production
Zhe Hao Koh	Electrical and Electronic Engineering	Did FPGA coding, prototype design and assembly, PCB design in previous project	Procrastination	I assume nothing I do in my free time helps in this project (football, cooking)

A.5. Meeting Minutes

7/10/2020 Discord

Tom W-V elected as PM
Matt working on power
Research area assigned
Power- Matt
Communication - Will
Sensors, Canisters, Liquids, microcontroller - Pragash, Zhe
GPS/Project - Tom
Meeting arrange for 8/10/2020 at 15:00pm

8/10/2020 Discord
Discussed compartmentalizing aspects where you can upgrade individual components. Cost vs gain in power management in report.
Research about:

- Power- 3.7V battery outlined but could be undervoltage
  - Other options are 7.4V centered around the 3.7V multiples
  - Exploring making own Arduino as a stretch goal
  - Also Solar pack stretch goal but it was proposed difficulties
- Communications- worked to basic assumption of LE, Low cost and cellular.
  - LORA is expensive so for our setup not ideal and limited range
    - Will discuss limitations
  - SIGFOX explored but possible stretch goal at first
  - Arduino IOT suggested as an initial start
  - Research showed using accelerometer to detect long movement before activating high energy GPS
- S,C,L,M- Types of sensors looked
  - Tuning Fork – point level switch
  - Microwave
  - Laser
  - Ultrasonic
  - Optical Prism
  - Capacitance
  - Pressure
  - Floats
  - Seek to clarify the binary nature of full or not
  - Aim to look at gas canister choices ideally noninvasive
  - Discussed acquiring both types of cylinder empty on ebay
  - Stretch goal of custom microcontroller after Arduino.
- GPS- Two choices, both Arduino based and both cheap in power terms

Discussed writing of project spec for submission to Steve tomorrow

9/10/2020 Discord
Upon feedback from Steve and discussion the executive decision was made to switch to the LoRa network in total.

Decided again upon clarification from Steve to make device report weekly regardless of change in system status with the addition of updates upon change.
Decided to not pursue custom microcontroller board but to focus on disabling functions for energy saving.
More or less know GPS sensor
More or less solved Battery
More or less solved LoRa
Prioritise sensors
Then Arduino
Then LoRa.

12/10/2020 Teams Steve B
Mentioned LPG canister issues to Steve, he has said to link him some canisters for our needs.
Talked about comments relating to LoRa and he said it is not a main focus but be prepared to defend.
Discussed Down selection process and Steve Approved
Steve Suggested building sensor if necessary.
Discussed aspects of project managements.
Discussed the project planning flow chart, Steve suggested making a flow diagram of processes rather than of operation, he said make it a maximum of 5 tasks and suggested items can be in parallel in addition to what we have already. Add comms into microcontroller with original diagram.
Steve suggested emailing regarding LoRa, he mentioned fair usage rules.
https://www.thethingsnetwork.org/community/southampton/
Steve suggested for power dealing in Charge(coloumbs) vs watts when it comes to power assessment as it isn't an average.
Steve agreed with multiple microcontrollers
Questioned Arduino usage stating it is a prototyping tool mentioned ease of use removes features.
Using other microcontroller allows us to demonstrate skills in a more effective fashion.
Looking at temperature and or flow rate as an options.
Generate power from peltier?
Steve said why not can be more effective and useful rather than why you should.

12/10/2020
Assigned research areas:
- Ultrasonic and Optical Prism Zhe
- Laser and Floats Matt
- Microwave and Pressure Pragash
- Tuning fork and Capacitance Will

Canister size was defined as 13Kg capacity in both Composite and Steel.
Arranged meeting for tomorrow afternoon.

13/10/2020
- Matt's Research:

Most lasers are invasive but low power and cheap.
Can only be noninvasive using composite.
£35-40?
Floats

Power management is an issue as high voltage and high current.
So more than likely not viable
But can be cheap
Binary output
LPG is an insulator so a switch could be submerged

Ultrasonic discussed as one found from https://www.gasit.co.uk/
Using on the side gives a binary output removing need for calibration beyond initial startup.

Will's Research:
Tuning fork has to be invasive
Based on a system of a tuning fork which varies its vibration when fill level changes
Basically, dead before we start.

Discussed a system which is noninvasive measuring the frequency of canister which varies relative to fill level.

Capacitance
Invasive capacitor uses liquid level to vary dielectric constant and measure capacitance
Non invasive only works with composite.

Pragash's Research:

- Microwave sends the wave down a rod calculating with time of flight
- Pressure is invasive as well.

Downselection of Ultrasonic (research at all), float, frequency of canister.

Discussed finalising project plan by Wednesday 14th October
Slides to be written and compiled by Friday 16th @ 12:00pm

19/10/2020 Teams Steve B
Discussed Ultrasonic sensors with Steve and He mentioned hardware timers can provide mm resolution.
Suggested getting expensive sensor and then try with a cheaper one
He suggested setting ground rules about individual work and remind everyone that the mark includes individual marks.
Steve suggested reinvestigating capacitance as an invasive sensor type.
Specific heat capacity could be interesting
Suggested quantising data as late as possible in the chain.
Steve suggested composite prioritisation for sensor selection.
Steve also suggested substituting if the canister is unlikely.
Steve brought up weight but said bring up and rule out in report.
Lift up weighing testing using servos but power hungry.
Microchip RN2483 suggested by Steve with a breakout board.
Steve said gas flow is a last-minute resolution which works backwards rather than forward.
Steve suggested 2 times a day communication during this project as an average.
Steve suggested infra red lidar VL6180X as they have proved useful in the past.
Steve suggested drafting out sensor report so that we can think about it whilst writing.

Is LPG opaque at infrared and explore that for optical.
Steve agreed that talking about costings of total operations. For 100,000.
Tom agreed to email time and date of second presentation.

19/10/2020 Discord
Write presentation 1 and draft final by end of Wednesday.
Record and script on Thursday
Video edit Friday for submission
Recommend using audacity and OBS as free.
Discussed researching Ultrasonic Today
Research floats tomorrow
Result resonance sensing Wednesday.

26/10/2020 Teams
Talked about ultrasonic sensors, Steve mentioned ultrasonic transducers for cheaper from DigiKey
Steve mentioned decay time of transducer overwriting the output.
Mention patents in final writing and check what if sonic or ultrasonic.
Steve said research floodflash
And https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc1700-cortex-m3/512kb-flash-64kb-sram-ethernet-usb-lqfp100-package:LPC1768FBD100
Cortex M3 M4 and M0 are powerful signal processors.

Float sensor sealed for life system with radio protocol. Arranging listening protocols.

Resonance sensor, which nodes are being excited and the responses.

Determine the valve associated with the composite canisters available in the UK. And as a result fitting cell through hole.
Adding testing functionality.

2/11/2020 Teams
Mentioned float sensor being used.
Discussed valve sizes, Steve agreed with 5/8 BSP
CR2032 too wide for this
Button cells have poor pulse supply
Recommend LS14250

2/11/2020 Discord:
Start the writing process of the end presentation for the sensor choices.
Need to research shelf lives of cell choices mentioned.
Research of polystyrene in pressure chambers.
Look into mounting system, expanding arms seems best
Research into super low power Bluetooth or RF communication method for internals.
Look at super power-efficient microcontrollers as low processing power required.
https://www.silabs.com/documents/public/data-sheets/efm32g-datasheet.pdf

9/11/2020 Feedback Meeting

Presentation:

Wasn't overly impressed generally list of bullet points without extra information being added.
Block diagram would have been good
Disjointed with interconnectivity.
For next presentation make it more visual which compliments the script rather than just repeating the words Add diagrams.
Asked about next presentation date (18th)
Talk about requirements
Deciding on battery first is back to front
Specify the requirements for the sensor and comms
Good also to mention the actual design choices and advs and dis adv
Mention Hard constraints and soft constraints.
Convey the work in the background rather than the end result of the research.
Talk about range of devices.
Presentation has not conveyed our actual workflow.

Visuals?

Overall block diagram,
Capture all constraints first and then go from there rather than talk about individual parts.
Pictures and diagrams.

Tips?

Project manager kicks things off and overall block diagram
Just add slight animation.
Space constraints, protocols. Demonstrate the work in the background

Current Progress:

Make it clear what the resolution is
Talk about why not weight.
Talk about accuracy more
Justify why ideas suck
Steve wanted to sharpen up our answers.
https://files.seeedstudio.com/products/113990637/res/(RE)RM0095EN_EMB1061%20Firmware%20Programming%20Manual_V1.0.pdf
Programming manual for the BLE module.
Show optimisation for turning modules on and off with timing diagrams in the next presentation.
Highlight programming systems, ie interrupts and/or polling.

Recording a dry run for feedback from Steve G.
Insider info says our is at 2:40.

Mention in report separation issues with working. And risk assessment.

9/11/2020 Steve B Meeting

Look at Nordic BLE rather than this mxchip as mxchip is proving difficult already.

9/11/2020 Afternoon Discord Meeting

Discussed change of float design to a sideways float rather than vertical slide as this allows for more concentrated package of electronics.
Researched the power consumption of each individual nRF chip and decided on nRF52840 as it has dual UART ports.

10/11/2020 Discord
Created requisition for other auxiliary components and submitted.
Filled in SOP for building 16

11/11/2020 Discord
Buoyancy calculations related to float density
Started presentation 2 slides.

13/11/2020 Discord
Finalised the presentation slides for the next week.

16/11/2020 Discord
Process of script writing and tidying of slides started before later meeting.

16/11/2020 Teams with Steve B
Discussed ordering of components.
Mentioned cells selected.
What is the lifetime of external battery?
Discussed downsizing of the external battery.
Discussed cold start and hot start conditions mentioned it being purely cold start
Succinct battery capacity mentioned in terms of coulombs and specification for battery.
Draft numbers at a minimum.
Identify the important issues versus the non important.
Discussed size constraints for the Bluetooth board.
Steve said just email back digikey.
Add Steve b to presentation recording but do not expect response.
Prioritise what can be completed now.

16/11/2020 Discord
Assigned tasks to everyone for the script writing and information contained:
- Pragash: Floats
- Matt: Power

23/11/2020 Teams with Steve B
Discussed battery gas leakage.
Discussed components arriving
Do we have programming architecture outlined?
ASM chart discussed
Steve said look at real time embedded notes page.
Use interrupts sparingly only to post to state machine.
Steve mentioned doing physical testing with the cad design just to see how well it actually works

Run to one significant figure rather than multiple as it just adds nothing.
Forward power spreadsheet to Steve.
Discussed physical meetings
Discussed Christmas and cancelling the 28th of December meeting.

30/11/2020 Steve B Teams
Mentioned internal broadcasting every 5 seconds rather than synchronisation
Short advertisement .5ms length
Satellite location is worth exploring mentioning maybe switch to glonass. Look at patterns of coverage and inclination for signal strength.
Discussed contribution this week
Send Steve a contents list asap in the report writing stage.

10/12/2020 Steve Gunn
Steve said we took the feedback on and the presentation was really good
Steve said talk about robust nature of code to allow change gps or other components
Add a JTAG programmer setup to the PCB for the internals.
Steve suggested a testing jar for demonstration
Testing of incline as part of the testing procedure.
Look at using Overleaf.

14/12/2020
Talked about project state
Steve advised adding a Bluetooth advertising state having the data rather than connecting and disconnecting. At least look and discuss
Changing pulse length to 1 minute
Will mentioned transcribing of data from the videos.
Steve said a few days can be sufficient for desync testing.
Steve mentioned JTAG sizing for the internal PCB
Zhe said he would convert the images to current values.
Matt mentioned writing up of battery
Steve asked about what the end project is
Steve also asked about What3words integration. Worth a look at very least inclusion into report
With regards to testing he thinks we are okay make sure it is functional
Desync testing doesn't need to be so long a few days is manageable

21/12/2020
Pragash didn't attend but no statement as to why not.
Talked about state of project continuation with regards to testing, report writing, and mechanical design
Steve mentioned testing using a hardware written gps value
Asked steve about mentioning ASH in the document he also suggested the story related to Gas4All
Talked about Bluetooth advertising and inability to get it to work
In response steve said talk about it but say we didn't due to library choices.

No Minuteable Meetings conducted since 21/12/2020

A.6. Microsoft To Do Tasks

GDP43

15 January 2021

- [ ] Weekly Report
  0 of 5 • Fri 22 Jan
  - [ ] Will
  - [ ] Tom
  - [ ] Matt
  - [ ] Pragash
  - [ ] Zhe

Completed

- [x] ~~Record Video Components~~ ★
  ✓ 5 of 5
  - [x] Will
  - [x] Tom
  - [x] Pragash
  - [x] Matt
  - [x] Zhe
- [x] ~~Submit Project Specification~~ ★
- [x] ~~Write Powerpoint Slides~~ ★
  ✓ 4 of 4
  - [x] Product Specification
  - [x] Power Management
  - [x] Communication Requirements
  - [x] Project Management
- [x] ~~3D model of Sensor~~ WT
- [x] ~~Add the switch~~

Printed with Microsoft To Do 1/9

GDP43

15 January 2021

- ~~Arrange Meeting with Steves for feedback~~ WT
- ~~Arrange Pickup of Gas Canister~~ WT
- ~~ASM diagram~~
  ✓ 2 of 2
  - Basic aspects
  - Fully drawn
- ~~Battery Research~~ CM
- ~~BLUART~~
  ✓ 2 of 2
  - Create Functions
  - 5 second interval broadcasts
- ~~Bluetooth Power States~~
- ~~Collect From building 16~~
- ~~Comment all Code~~
  ✓ 5 of 5
  - Main setup and loop file
  - LORA.h
  - LORA.cpp
  - Air530.cpp
  - Air530.h
- ~~Complete 2nd Presentation Script~~
- ~~Complete 2nd Presentation Slides~~

GDP43

15 January 2021

- ~~Create a budget spreadsheet~~ WT
- ~~Create a Risk Management Plan~~ WT
- ~~Create a visualisation of data being transmitted~~
- ~~Draft up documentation for UART, pinouts, and communications.~~
  ✓ 3 of 3
  - GPS
  - LorA
  - BT
- ~~email digikey~~
- ~~Email Jeff Hooker about Ossila SMU~~
- ~~Email LoRa manager at soton~~
- ~~Email steve the minutes~~ WT
- ~~Establish LoRa comms with TTN~~
- ~~Expanded Polystyrene properties under high pressure for LPG~~ BP
- ~~Implement Low Power modes for GPS~~
- ~~Initial BLUART implementation~~
- ~~Integrate BLUART and GPSLoRa~~

GDP43

15 January 2021

- ~~Integrate GPS and LoRa~~
- ~~Look into meetings rooms in B16 and B100~~ WT
- ~~Mircontroller for interacting with LoRa and GPS~~ KZ
- ~~PCB Design~~ BP
- ~~Permanent Soldering using female headers~~
- ~~Power Consumption of schematics~~
- ~~Power Use Summary for all chips (Worst Case)~~
- ~~Reconnect BLE after '24hr delay' and get new fill level~~
- ~~Reply to Steve about pres 2~~
- ~~Report Sections Initial Write~~
- ~~Research and downselect Microcontroller~~
- ~~Research BLUART~~
- ~~Research into Microwave and Pressure~~ BP
  ✓ 2 of 2
  - Microwave
  - Pressure
- ~~Research into Tuning fork and Capacitance~~ SW

GDP43

15 January 2021

- [x] ~~Research into Ultrasonic and Optical Prism~~ KZ
- [x] ~~Research Laser and Floats~~ CM
- [x] ~~RF and BT Modules~~ SW
- [x] ~~Save Film of graph onto OneDrive~~
- [x] ~~Send eBay links to Steve B~~
- [x] ~~Send in SOP~~
- [x] ~~Sleep and wakeup lora chip~~
- [x] ~~Solder LoRa module~~
- [x] ~~Stripboard from breadboard~~
- [x] ~~Testing~~
  ✓ 3 of 3
  - [x] Float Sensor
  - [x] BT dB Power States
  - [x] 6 month test
- [x] ~~Tie GPS Wake Pin to a feather IO pin~~
- [x] ~~Update main report sensor document~~
  2 of 4 • Wed, 4 Nov 2020
  - [x] Matt
  - [ ] Zhe
  - [ ] Pragash

- [x] Will

- [x] ~~Weekly Report~~
3 of 5 • Fri, 13 Nov 2020

  - [x] Will
  - [x] Tom
  - [x] Matt
  - [ ] Pragash
  - [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 23 Oct 2020

  - [x] Will
  - [x] Tom
  - [x] Matt
  - [ ] Pragash
  - [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 6 Nov 2020

  - [x] Will
  - [x] Tom
  - [x] Matt
  - [ ] Pragash
  - [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 30 Oct 2020

  - [x] Will
  - [x] Tom
  - [x] Matt

- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri 8 Jan

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri 1 Jan

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 25 Dec 2020

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 18 Dec 2020

- [x] Will
- [x] Tom

- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 11 Dec 2020

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 4 Dec 2020

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 27 Nov 2020

- [x] Will
- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
3 of 5 • Fri, 20 Nov 2020

- [x] Will

- [x] Tom
- [x] Matt
- [ ] Pragash
- [ ] Zhe

- [x] ~~Weekly Report~~
  3 of 5 • Today

  - [x] Will
  - [x] Tom
  - [x] Matt
  - [ ] Pragash
  - [ ] Zhe

- [x] ~~Weekly Report~~
  3 of 5 • Fri, 16 Oct 2020

  - [x] Will
  - [x] Tom
  - [x] Matt
  - [ ] Pragash
  - [ ] Zhe

A.7. Code

as

```
/*---Custom .h files to include---*/
#include "LORA.h"

SoftwareSerial mySerial(10, 11);
//Set up a secondary serial port using pins 10 and 11 as RX and TX
rn2xx3 myLora(mySerial);
//Create an instance of the rn2xx3 library,
CayenneLPP lpp(51);
//Set the maximum buffer size of the CayenneLPP library

void sendLoRa(int fillLevel, float latitude, float longitude, int
altitude) {                          //Function to package the GPS data
and fill level and send it over the LoRa network
  lpp.reset();
//Remove any stored packets in the CayenneLPP buffer
  lpp.addGPS(1, latitude, longitude, altitude);
//Add GPS data to CayenneLPP
  lpp.addDigitalOutput(2, fillLevel);
//Add a digital output to CayenneLPP (Fill Level)
  myLora.txBytes(lpp.getBuffer(), lpp.getSize());
//Send the CayenneLPP packet over the LoRa network
}
bool initialize_radio() {
//Function to wake up the LoRa chip and connect to the Things Network
  mySerial.begin(9600);
//Open the secondary serial port to the LoRa chip
  pinMode(12, OUTPUT);
//Specify the pin on the Feather connected to the LoRa chip's reset as an
output
  digitalWrite(12, LOW);
//Pull the LoRa chip's reset pin low. This both disconnects any existing
conection
  delay(500);
//and wakes up the board from sleep. We will use this feature to
improvise a GPIO
  digitalWrite(12, HIGH);
//wake function on the LoRa board
  delay(100);
  mySerial.flush();
//Clear the Software Serial
  myLora.autobaud();
//Use the library's autobaud feature to set the baud rate on the LoRa
chip to 9600
  String hweui = myLora.hweui();
//Test the communication with the LoRa board by asking for its DevEUI
  while (hweui.length() != 16) {
//The DevEUI registered to the board should be 16 characters long
    delay(10000);
//Wait 10 seconds before trying again
    hweui = myLora.hweui();
//Try to get the DevEUI again
  }
  bool join_result = false;
//Initialise a connected flag to be false
  const char *appEui = "70B3D57ED00383AE";
//These are the keys generated in the Things Network console
  const char *appKey = "B50868053F67ECAC2FE104E1CD5E650B";
//They authenticate the connection request
```

```
  join_result = myLora.initOTAA(appEui, appKey);
//Try to join the LoRa network using the keys above. On success set the
flag true
  while (!join_result) {
//We only enter this loop if the connection isn't initially successful
    delay(10000);
//Delay 10 seconds before retrying
    join_result = myLora.init();
//Try to join the LoRa network again
  }
  return true;
//Return true, so that the while loop breaks where this function was
called
}

void convertLatLong(unsigned char* buffer, float latitude, float
longitude, int level, int altitude) { //Function to convert raw GPS data
into latitude and longitude
  /*The buffer will look like this when coming in from the GPS:
$GNGLL,5055.9050,N,00123.1481,W,000546.041,N,N*54*/
  String strLatDeg((char*)buffer);
//Convert the char buffer into strings to make it easier to manipulate
  String strLongDeg((char*)buffer);
//We use 4 different strings, one for degrees of latitude,
  String strLatMin((char*)buffer);
//one for minutes of latitude, one for degrees of longitude,
  String strLongMin((char*)buffer);
//and one for minutes of longitude
  int NSmodifier = 1;
//A variable to modify the sign of the latitude value
  int EWmodifier = 1;
//A variable to modify the sign of the longitude value
  if (strLatDeg.charAt(17) == 'S') {
//If the 18th character (0 indexed) in the string is an S
    NSmodifier = -1;
//Southern hemisphere reading, meaning negative latitude
  }
  else {
//If the 18th character is an N
    NSmodifier = 1;
//Northern hemisphere reading, meaning positive latitude
  }
  if (strLongDeg.charAt(30) == 'W') {
//If the 31st character (0 indexed) in the string is a W
    EWmodifier = -1;
//Western hemisphere reading, means negative longitude
  }
  else {
//If the 31st character is an E
    EWmodifier = 1;
//Eastern hemisphere reading, means positive longitude
  }
  strLatDeg.remove(16, 33);
//Trim ,N,00123.1481,W,000546.041,N,N*54 from the 1st string
  strLatDeg.remove(0, 7);
//Trim $GNGLL, from the 1st string, to leave latitude (5055.9050)
  strLatMin.remove(16, 33);
//Trim ,N,00123.1481,W,000546.041,N,N*54 from the 3rd string
```

```
  strLatMin.remove(0, 7);
//Trim $GNGLL, from the 3rd string, to leave latitude (5055.9050)
  strLongDeg.remove(29, 33);
//Trim ,N,N*54 from the 2nd string
  strLongDeg.remove(0, 19);
//Trim $GNGLL,5055.9050,N, from the 2nd string, to leave longitude
(00123.1481)
  strLongMin.remove(29, 33);
//Trim ,N,N*54 from the 4th string
  strLongMin.remove(0, 19);
//Trim $GNGLL,5055.9050,N, from the 4th string, to leave longitude
(00123.1481)
  strLatDeg.remove(2, 7);
//Trim 55.9050 from latitude to leave degrees only (50)
  strLongDeg.remove(3, 7);
//Trim 23.1481 from longitude to leave degrees only (001)
  strLatMin.remove(0, 2);
//Trim 50 from latitude to leave minutes only (5.9050)
  strLongMin.remove(0, 3);
//Trim 001 from longitude to leave minutes only(23.1481)
  latitude = ((strLatDeg.toFloat()) + (strLatMin.toFloat() / 60)) *
NSmodifier;                          //Add latitude degrees to minutes/60,
multiply by the NS modifier (1 or -1)
  longitude = ((strLongDeg.toFloat()) + (strLongMin.toFloat() / 60)) *
EWmodifier;                       //Add longitude degrees to minutes/60,
multiply by the EW modifier (1 or -1)
  sendLoRa(level, latitude, longitude, altitude);
//Call the funtion that sends the GPS data and fill level to the LoRa
network
  delay(1000);
//Delay to allow for that data to send
  sleeplora();
//Call the function that sleeps the LoRa module
}
void sleeplora() {
//Function to sleep the LoRa module
  myLora.sleep(20000000);
//We sleep the module for several days and rely on the reset pin to wake
it up early
}
```

```
/*---.h file boilerplate---*/
#ifndef _H
#define LORA_H
/*---.h files to include---*/
#include <rn2xx3.h>
#include <SoftwareSerial.h>
#include <CayenneLPP.h>
/*---List of functions inside LORA.cpp---*/
void sendLoRa(int fillLevel, float latitude, float longitude, int
altitude);
bool initialize_radio();
void convertLatLong(unsigned char* buffer,float latitude,float
longitude,int level,int altitude);
void wakelora();
void sleeplora();
#endif
```

```
/*---Libraries to include---*/
#include "Arduino.h"
#include "String.h"
/*---Custom .h files to include---*/
#include "AIR530.h"

void sendcmd(String cmd) {
//Function to send a packaged command to the GPS
  while (Serial1.available()) {                                    //While
data is coming from the GPS
    Serial1.readStringUntil('\n');                                 //Read
it until the end of the line. Prevents transmission both ways at the same
time
  }
  Serial1.print(cmd);                                              //Send
the packaged command to the GPS
}
String calculatechecksum(String cmd) {
//Function to calculate the checksum to be added to the end of commands
  uint8_t checksum = cmd[1];                                       //Set
the checksum to be the 2nd character in the cmd string
  char temp[5];
//Initialise a temporary buffer
  for (int i = 2; i < cmd.length(); i++) {                         //For
the length of the cmd string
    checksum ^= cmd[i];                                            //To
create the checksum, XOR (^) consecutive values in the cmd string
  }
  memset(temp, 0, 5);                                              //Fill
the temp buffer with integers
  sprintf(temp, "*%02X\r\n", checksum);                            //Fills
the temp buffer with the checksum, followed by \r\n, the required end of
line characters
  cmd += temp;
//Concatenates cmd and temp
  return cmd;
//Return the final package with checksum applied
}

void setsatellites() {
//Function to specify which satellite clusters we want to request data
from
  String cmd = "$PGKC115,1,0,0,1";
//Request GPS and Galileo only, no Beidou or GLONASS
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                    //Call
the function that sends the command
}
void setnmea() {
//Function to specify which NMEA data formats we want
  String cmd = "$PGKC242,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0";
//Request GLL Only
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                    //Call
the function that sends the command
}
```

```
void PPSoff() {
//Function to turn off the 1 pulse per second output on the GPS
  String cmd = "$PGKC161,0";                                    //0
means off, 1 means on
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                 //Call
the function that sends the command
}
void setinterval() {
//Function to specify the interval at which th GPS sends data
  String cmd = "$PGKC101,10000";                                //Set
interval to 10000ms (10 seconds)
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                 //Call
the function that sends the command
}

void GPSreset() {
//Function to send the system reset command
  String cmd = "$PGKC030,3,1";                                  //The 3
specifies a 'cold' restart. The 1 specifies a full software restart too
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                 //Call
the function that sends the command
}
void eraseGPS() {
//Function to erase the auxiliary positioning data in the flash memory of
the GPS
  String cmd = "$PGKC040";                                      //The
command has no arguments
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                 //Call
the function that sends the command
}
void sleepGPS() {
//Function to place the GPS into sleep mode
  String cmd = "$PGKC105,4";                                    //The 4
specifies ultra-low power mode. The GPS must be woken by pulling up the
WAKE pin
  cmd = calculatechecksum(cmd);
//Calculate appropriate checksum
  sendcmd(cmd);                                                 //Call
the function that sends the command
}
void wakeGPS() {
//Function to wake the GPS up
  digitalWrite(9, LOW);                                         //Write
the pin on the feather (Pin_9) connected to the GPS WAKE pin low
  delay(1);
//Minimal delay to save power
  digitalWrite(9, HIGH);                                        //Bring
the pin back up
}
```

```
void initGPS() {
//Function to configure the GPS, using the commands above
  delay(100);
  setsatellites();
//Specify the satellite clusters the GPS uses to get a fix
  delay(100);
  setnmea();
//Specify the data we want from the GPS, GLL (lat, long, UTC)
  delay(100);
  setinterval();
//Specify the interval between GPS data transmissions (10 seconds)
  delay(100);
}
```

```
/*---.h file boilerplate---*/
#ifndef GPS_Air530_H
#define GPS_Air530_H
/*---.h files to include---*/
#include "Arduino.h"
#include "String.h"
/*---List of functions inside AIR530.cpp---*/
void sendcmd(String cmd);
String calculatechecksum(String cmd);
void setsatellites();
void setnmea();
void PPSoff();
void setinterval();
void GPSreset();
void eraseGPS();
void sleepGPS();
void wakeGPS();
void initGPS();
#endif
```

```
/*
 ** Pin Connections**
  LoRa -- Feather
    TX -- 10
    RX -- 11
   Rst -- 12
   VCC -- 3V
   Gnd -- Gnd

   GPS -- Feather
    TX -- RX
    RX -- TX
   VCC -- 3V
   Gnd -- Gnd
  Wake -- 9
*/
/*---Libraries to include---*/
#include <Adafruit_GPS.h>
#include <Adafruit_PMTK.h>
#include <string.h>
#include <bluefruit.h>
/*---Custom .h files to include---*/
#include "AIR530.h"
#include "LORA.h"

BLEUart bleuart;
//Create an instance of the BLEUart library
BLEClientUart clientUart;
//Create an instance of the BLE Client UART library
/*---Variable declarations and initialisations---*/
char sw;
//Character to store the switch state of the float sensor
int level = 2;
//Initialise fill level to be an impossible value so we know when data
has been recieved
unsigned char buffer[64];
//Buffer array for GPS data recieved over the Serial1 port
int count = 0;
//Variable to store the length of the buffer array
float latitude = 0;
//Float variable initialised so we can pass latitude as a variable to
convertLatLong in LORA.cpp
float longitude = 0;
//Float variable initialised so we can pass longitude as a variable to
convertLatLong in LORA.cpp
int altitude;
//Initialise a variable for CayenneLPP to include as an altitude with GPS
packets.
int connectedBLE = 0;
//Flag to indicate BLE connection
int GLLworking = 0;
//Flag to indicate if the Air530 has received its config commands
int sleeping = 0;
//Flag to indicate if the boards are sleeping
int timeoutcheck = 0;
//Counter to check the number of seconds a function is unresponsive for
/*---Function declarations---*/
void clearBufferArray();
```

```
void scan_callback(ble_gap_evt_adv_report_t* report);
void connect_callback(uint16_t conn_handle);
void disconnect_callback(uint16_t conn_handle, uint8_t reason);
int Bleuart_recieve(void);

void setup() {
  Serial1.begin(9600);
//Open the Serial port to the Air530 GPS
  pinMode(9, OUTPUT);
//Initialise the GPS wake pin as an output
  altitude = 0;
//We increase the altitude variable by one each time we send data to
check that we aren't receiving old data
  while (!initialize_radio()) {
//Call the function initialize_radio, which connects to the LoRa network
    delay(100);
//Wait until the LoRa chip has connected before continuing
  }
  getBLEswitch();
//Call the function getBLEswitch, which starts advertising for a BLE
connection
  while (connectedBLE == 0) {
//Wait until the chip inside the canister has connected before continuing
    delay(100);
  }
  delay(100);
  initGPS();
//Call the function initGPS which sends config commands to the AIR530
GPS.
}
void loop() {
  while (sleeping == 0) {
//Only run this loop when the boards are awake
    while (level == 2) {
//If the fill level hasn't yet been received from inside the canister
      level = Bleuart_recieve();
//Assign the value being sent over BLE to the variable 'level'
      if (level != 2) {
//If the fill level has been received
        Bluefruit.disconnect(0);
//Disconnect the 1st (0-indexed) device, the feather in the canister.
        timeoutcheck = 0;
//Reset the timeout check counter to 0
      }
      delay(500);
//Delay to allow the timeout checker to count.
      timeoutcheck++;
//Increase the timeout check counter (every half a second)
      if (timeoutcheck == 40)
//If 20 seconds has elapsed
      {
        connectedBLE = 0;
//Reset the connectedBLE flag to be low as the chip is no longer
connected to BLE
        BLEreconnect();
//Call the BLEreconnect function which restarts BLE advertising for a
connection
```

```
        while (connectedBLE == 0) {
//Wait until the chip inside the canister has connected before continuing
          delay(100);
        }
        timeoutcheck = 0;
//Once the chip has reconnected, reset the timeout check counter to be 0
      }
    }
    if (Serial1.available()) {
//If data is coming into the Serial1 port ==> data is coming from GPS
      while (Serial1.available()) {
//For the whole time data is coming into the Serial1 port
        buffer[count++] = Serial1.read();
//Read each character that comes over the Serial1 port into consecutive
places in the buffer array.
        if (count == 64)break;
//If the buffer is full, stop reading in data
      }
    }
    String strbuffer((char*)buffer);
//Convert the buffer array into a string so that we can manipulate it
more easily
    delay(100);
    if ((strbuffer.startsWith("$GNGLL")
//If the GPS data is this acceptable format
         || strbuffer.startsWith("$GPGLL"))
//Or this acceptable format
        && strbuffer.length() > 34) {
//And hasn't been cut off (is long enough to contain lat and long data)
      /*We are looking for:
$GNGLL,5055.9050,N,00123.1481,W,000546.041,N,N*54*/
      sleepGPS();
//Call the function sleepGPS which sends the GPS into ultra-low power
(wake on WAKE pin) mode
      altitude++;
//Increase altitude to show that we have got new data
      convertLatLong(buffer, latitude, longitude, level, altitude);
//Call the convertLatLong function which parses the GPS data and sends it
over LoRa
      sleeping = 1;
//Set the sleeping flag high
      level = 2;
//Reset the fill level to an impossible value so that we can get a new
one after sleeping
    }
    if (strbuffer.startsWith("$GNGGA")) {
//If the GPS data is the incorrect format, the GPS didn't accept its
config commands
      initGPS();
//Call the function initGPS again to reconfigure the GPS
      GLLworking = 1;
//Set the GLLworking flag high to show that the GPS has been reconfigured
    }
    else if (GLLworking == 0) {
//Effectively an 'else' for when the GPS sends incoherent data
      initGPS();
//Call the function initGPS again to reconfigure the GPS
```

```
      GLLworking = 1;
//Set the GLLworking flag high to show that the GPS has been reconfigured
    }
    clearBufferArray();
//Call the function clearBufferArray which replaces all slots in the
buffer array with '0'
    count = 0;
//Reset the buffer array length variable to 0
  }
  delay(30000);
//24 hour delay which sets the feather into low power mode. For
demonstrations the delay is set to 30 seconds
  sleeping = 0;
//After 24 hours has elapsed, set the sleeping flag low so that we can
run the data acquisition loop again
  while (!initialize_radio()) {
//Call the initialize_radio function again to reconnect to the LoRa
network
    delay(100);
//Wait for the connection to be established
  }
  GLLworking = 0;
//Reset GLLworking flag to be low in case the GPS isn't configured
properly on wake-up
  wakeGPS();
//Call the function wakeGPS which pulls the GPS wake pin low and then
high
  BLEreconnect();
//Call the BLEreconnect function which restarts BLE advertising for a
connection
  while (connectedBLE == 0) {
//Wait until the chip inside the canister has connected before continuing
    delay(100);
  }
}
void BLEreconnect() {
//BLEreconnect function, used when reawakening from 24 hour sleep
  bleuart.begin();
//Start the bleuart service
  clientUart.begin();
//Start the clientUART service
  Bluefruit.setConnLedInterval(250);
//Set the blink interval for the 'advertising' LED
  Bluefruit.Central.setConnectCallback(connect_callback);
//Initialise a function to be called when a connection is established
  Bluefruit.Central.setDisconnectCallback(disconnect_callback);
//Initialise a function to be called when a device disconnects
  Bluefruit.Advertising.addService(bleuart);
//Advertise for bleuart enabled connections
  Bluefruit.Scanner.setRxCallback(scan_callback);
//Initialise a function to be called when a possible connection is
detected
  Bluefruit.Scanner.restartOnDisconnect(false);
//Don't restart scanning if a disconnect occurs, we will restart it
manually
  Bluefruit.Scanner.filterRssi(-80);
//Filter out packets with a RSSI lower than -80
```

```
  Bluefruit.Scanner.setInterval(160, 160);
//Set a duty cycle of 100% for advertising interval (intervals of
0.625ms), (interval,on time)
  Bluefruit.Scanner.useActiveScan(true);
//Request scan response data
  Bluefruit.Scanner.start(0);
//Don't timeout the scanning.
}
void getBLEswitch() {
  Bluefruit.begin(0, 1);
//Initialise Bluefruit with maximum connections as Peripheral = 0 and
Central = 1
  Bluefruit.setTxPower(4);
//Set the TX power to be 4, check bluefruit.h for supported values
  Bluefruit.setName("Ble_central");
//Set the device name
  bleuart.begin();
//Start the bleuart service
  clientUart.begin();
//Start the clientUART service
  Bluefruit.setConnLedInterval(250);
//Set the blink interval for the 'advertising' LED
  Bluefruit.Central.setConnectCallback(connect_callback);
//Initialise a function to be called when a connection is established
  Bluefruit.Central.setDisconnectCallback(disconnect_callback);
//Initialise a function to be called when a device disconnects
  Bluefruit.Advertising.addService(bleuart);
//Advertise for bleuart enabled connections
  Bluefruit.Scanner.setRxCallback(scan_callback);
//Initialise a function to be called when a possible connection is
detected
  Bluefruit.Scanner.restartOnDisconnect(false);
//Don't restart scanning if a disconnect occurs, we will restart it
manually
  Bluefruit.Scanner.filterRssi(-80);
//Filter out packets with a RSSI lower than -80
  Bluefruit.Scanner.setInterval(160, 160);
//Set a duty cycle of 100% for advertising interval (intervals of
0.625ms), (interval,on time)
  Bluefruit.Scanner.useActiveScan(true);
//Request scan response data
  Bluefruit.Scanner.start(0);
//Don't timeout the scanning.
}
void clearBufferArray() {
//Function to clear buffer array
  for (int i = 0; i < count; i++ {
//For the length of the buffer
  buffer[i] = 0;
//Replace each character in the buffer with 0
  }
}
void scan_callback(ble_gap_evt_adv_report_t* report) {
//Function to be called when a possible connection is detected
  if (Bluefruit.Scanner.checkReportForUuid(report, BLEUART_UUID_SERVICE))
{ //If advertising contains bleuart service
    Bluefruit.Central.connect(report);
//Connect to the device with bleuart UUID in advertising
```

```
  }
  else {
//If the advertising doesn't contain bleuat service
    Bluefruit.Scanner.resume();
//Keep scanning for a connection
  }
  Bluefruit.Scanner.resume();
//Keep scanning for a connection, in case the connection failed
}
void connect_callback(uint16_t conn_handle) {
//Function to be called when a connection is established
  delay(100);
  if ( clientUart.discover(conn_handle)) {
//Once the clientUART has established a connection
    clientUart.enableTXD();
//Turn on clientUART service TX
  }
  connectedBLE = 1;
//Set the connectedBLE flag high
}
void disconnect_callback(uint16_t conn_handle, uint8_t reason) {
//Function to be called when a device disconnects
  (void) conn_handle;
//Name of the device
  (void) reason;
//Reason for disconnect
}
int Bleuart_recieve(void) {
//Function to get the fill level from inside the canister
  while (clientUart.available()) {
//While the clientUART is receiving data
    sw = clientUart.read();
//Assign the value of sw to be the character being sent over BLE
    if (sw == '1') {
//If the value of sw is 1
      return (1);
//Return the function with a value of 1
    }
    if (sw == '0') {
//If the value of sw is 0
      return (0);
//Return the function with a value of 0
    }
    delay(500);
//500ms delay so that the timeout checker can count
    timeoutcheck++;
//Add one to the timeout checker
    if (timeoutcheck == 60) {
//If 30 seconds has elapsed (60 x 500ms)
      connectedBLE = 0;
//Set the connected BLE flag low, the chip is no longer connected
      BLEreconnect();
//Call the BLEreconnect function which restarts BLE advertising for a
connection
      while (connectedBLE == 0) {
//Wait until the chip inside the canister has connected before continuing
        delay(100);
      }
```

```
      timeoutcheck = 0;
//Once the chip has reconnected, reset the timeout counter to be 0
    }
  }
}
```