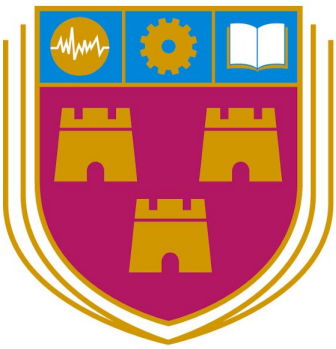
Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of* TECHNOLOGY CARLOW

At the Heart of South Leinster

# LifeBuoy Monitor
# Research Document

BSc (Hons) in Software Development

Name: Garry Byrne

Student ID: C00120055

Year: 4th Year

Supervisor: Dr. Oisin Cawley

Due date: 18-04-2018

# Table of Contents

**Abstract** **3**

**Introduction** **4**

What is the Sigfox network? 4

**Existing Solutions** **7**

**IOT Devices** **8**

Arduino 8

Raspberry Pi 9

Conclusion 10

**Sensors** **11**

Accelerometer 11

Altimeter 12

Magnetic Contact Switch 13

Conclusion
In the end i decided to go with the Altimeter/Accelerometer as it meet the requirements i had for this project. These were small size, low power requirements and low price. But the main benefit was the available libraries that would make the coding easier and quicker. 14

**Batteries** **15**

Lithium-ion 15

Alkaline 16

NiMH 16

Conclusion 17

**Power Requirements and Optimization** **18**

Hardware 18

Software 18

Conclusion 19

**Native versus Web Apps** **20**

Native 20

Web Apps 21

Hybrid App 22

Conclusion 22

**Development Tools** **23**
PlatformIO IDE 23
Embrio IDE 23
Programino IDE 24
Arduino IDE 25
Conclusion 25

**Databases** **26**
Relațional 26
NON-RELATIONAL 27
Conclusion 28

**Blbliography** **29**

# Abstract

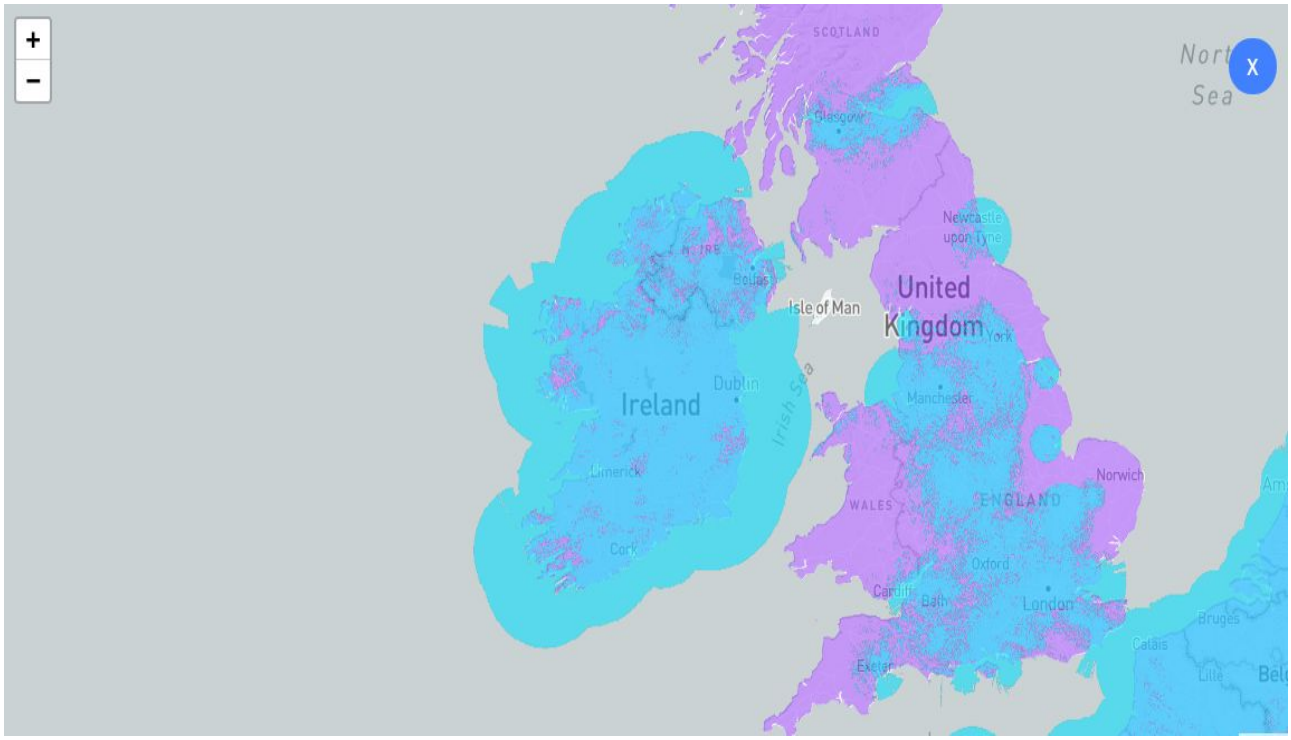
The purpose of this document is to list the research I completed and the information I gained throughout the duration of this project. The document goes through a variety of different technologies that I came across during the project along my reasons for choosing a particular technology. This document is to be updated throughout the entirety of this project.

# Introduction

The purpose of this fourth year project is to create a system, that will allow the user to monitor the status of all connected lifebuoys. There was only one existing system out there right now that I found during my research and this solution is expensive. This research document will look at a low cost solution to this problem, using IOT(Internet Of Things) prototyping boards like Arduino, Raspberry Pi. It will also have to communicate over the Sigfox network. I will also look into different types of databases along with different kinds of batteries to power the device.

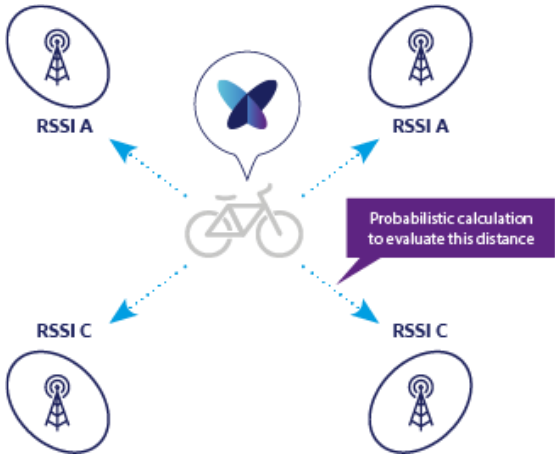
## What is the Sigfox network?

The Sigfox network is a wireless networks that allows low-energy devices(home automation, smart watches and remote health monitoring) to communicate over large distances, these devices need to be continuously on and emit small amounts of data. The network is a Low-power Wide-area network (LPWAN), this means that it is designed to allow long range communications at a low bit rate and low power consumption which means that a device can remain active for months with just a battery for power.The Sigfox network is said to have over 90% coverage in Ireland as you can see in the image below.




Sigfox uses a subscription based model, a subscription can be bought(yearly) or it can come included with certain devices(MKR 1200). The board I bought came with a two year subscription. Sigfox offer many benefits such as custom callbacks,location and statistic services. The most interesting service is callbacks, callbacks allow the user to receive data from a device and forward it to either their own infrastructure or on to a third parties infrastructure(AWS,Firebase).

Location is calculated with data from the Sigfox infrastructure, messages sent by a device are used to calculate the nearest base station. The method used is based on the signal strength (RSSI - Received Signal Strength Indicator) using a probability model.



This location method is not very precise but it can have between 1 km and 10 km accuracy, so depending on the use case of the device it can be acceptable eg, package tracking app that needs only to know when a package has reached a specific city..
The advantages of using the inbuilt location functionality of the Sigfox network is that

- No additional hardware is required.
- Battery consumption is not affected.

When you are sending a message through the Sigfox network you are limited to 140 messages a day which are limited to 12 bytes in size per message.
There are many other LPWAN’s available to use(LoRA, Weightless) but from looking at what Sigfox offers, it seems the best choice.

# Existing Solutions

There was only one similar system that I could find that is available in the market right now, which is the Lava Group's "Sentry" system. The Lava groups solution is to electronically monitor the lifebuoys. An alarm is triggered when a lifebuoy leaves its housing or "base". Sentry uses GSM or LPWAN technologies to transfer data. The Sentry solution was awarded the  UTV Business Eye Awards, R&D Project of the year.

Pros

- Remote monitoring of lifebuoys, instant alarms and notifications as soon as device is removed from its base station.
- Alerts and notifications can be viewed on multiple devices(PC, tablet, smartphone).

Cons

- The main downside of this solution is the cost associated with it. There are 10's of thousands of lifebuoys installed along Ireland's waterways. If each Sentry cost roughly 500 to install, it will cost 10's of millions to purchase and install them.

# IOT Devices

When looking at development boards, I had three criteria that I had to fulfill for meeting the guidelines of my project.
These were:

1. Size - The board must of small enough size so it can fit inside a lifebuoy.
2. Power Consumption - The board must function while drawing minimal power.
3. SigFox Compatible - The board must connect to the sigfox fox network, either through an in-built or external antenna.

I choose to concentrate on 2 different manufacturers of development boards. These are Arduino, Raspberry pi.

## Arduino

Arduino is a company who manufacturer hardware and software. The user community that has built up around the arduino company have a say in the creation of the company's future products. The hardware that arduino designs and manufactures include single-board microcontrollers and microcontroller kits. These products are distributed as open-source hardware and software

I looked at 3 arduino boards, these were the Arduino MKR FOX 1200, the Arduino Uno Rev3 and the Genuino ZERO.

- Arduino MKR FOX 1200 - The MKR FOX board uses a SAMD21 Cortex-M0+ as its cpu, which has a 32 bit architecture clocked at 48mhz. It also has 256kb of flash memory and 32 kb of SRAM. This board has a low power draw and operates on 3.3v. The dimensions of the board are 67.64 by 25mm and weighs 32g. This board comes with an antenna and 2 years subscription to the SIGFOX network. The cost of this board is £35.

- Arduino Uno Rev3 - The Uno board uses a ATmega328P as its cpu, which has an 8 bit architecture clocked at 16mhz. It also has 32kb of flash memory and 2kb of SRAM and 1kb of EEPROM. This board has a low power draw and operates on 5v. The dimensions of the board are 68.6 by 53.4mm and weighs 25g. This board needs an external antenna and a subscription to access the SIGFOX network. The cost of this board is £20.

- Genuino Zero - The Zero board uses a ATSAMD21G18 as its cpu, which has a 32 bit architecture clocked at 48mhz. It also has 256kb of flash memory and 32kb of SRAM. This board has a low power draw and operates on 3.3v. The dimensions of the board are 68.6 by 53.4mm and weighs just 23g. This board needs an external antenna and a subscription to access the SIGFOX network. The cost of this board is £39.

# Raspberry Pi

The Raspberry Pi Foundation is based in the UK, they develop small single board computers to help in the teaching of basic computer science. Education was its original purpose but because of its small size and low price, it was quickly adopted by creators and tinkerers who require more than a basic microcontroller (such as Arduino).
The Raspberry Pi is slower than modern laptops but it is still a complete Linux computer, this is of benefit because it has a much lower power consumption than many devices of its performance level.

I looked at 3 Raspberry boards, these were the RASPBERRY PI ZERO W, the RASPBERRY PI 2  and the RASPBERRY PI 3.

- RASPBERRY PI ZERO W - The Zero is the newest board offered by Raspberry Pi and was released in February 2017. It is powered by a Broadcom BCM2835 chip which is a quad core processor. The board has a clock speed of 900Mhz and has 1GB of RAM. The dimensions of the board are 65 by 30mm and weighs 9g. The cost of this board is £10. This board also needs a module(not included) to connect to the SIGFOX network which costs £66.

- RASPBERRY PI 2 - The Pi 2 was released in February 2015. It is powered by a Broadcom BCM2835 chip which is a single core processor. The board has a clock speed of 1Ghz and has 512mb of RAM. The dimensions of the board are 85 by 56mm and weighs 45g. The cost of this board is £34. This board also needs a module(not included) to connect to the SIGFOX network which costs £66.

- RASPBERRY PI 3 - The Pi 3 was released in February 2016. It is powered by a Broadcom BCM2837 chip which is a quad core processor. The board has a clock speed of 1.2Ghz and has 1GB of RAM. The dimensions of the board are 56 by 85mm and weighs 45g. The cost of this board is £32. This board also needs a module(not included) to connect to the SIGFOX network which costs £66.

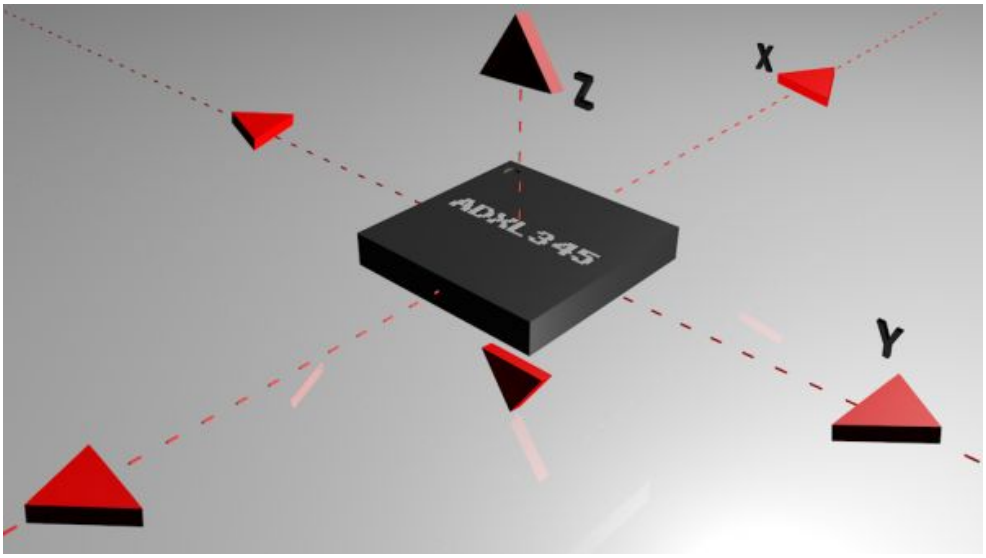
## Conclusion

In the end i decided to go with the MKR 1200 as it meet my needs for this project. The factors that the device had that met my requirements were price, formfactor, small power consumption(through arduino libraries) and out of the box connectivity to the Sigfox network.

# Sensors

## Accelerometer

An accelerometer is a device that can that sense either the static or dynamic forces of acceleration. An example of static and dynamic forces can be the force of gravity pulling at your feet and for dynamic forces you could physically move or vibrate the accelerometer. Accelerometers can measure acceleration on one, two or three axis. Accelerometers contain capacitive plates internally which can be fixed or can be attached to tiny springs that move internally as acceleration forces act upon the sensor.

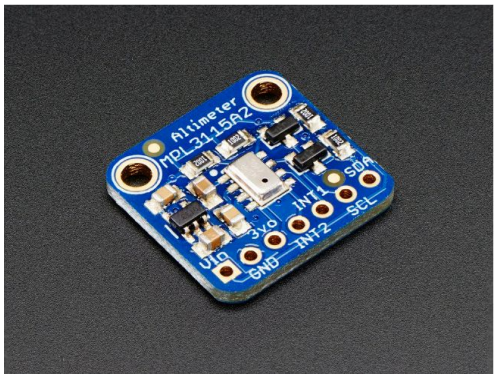



Accelerometers are usually low-power devices. The required current to run these devices are usually in the micro (μ) or milli-amp range, with a voltage requirement of 5V or less. The current consumption can vary depending on the settings (e.g., power saving mode versus standard operating mode). These modes can make accelerometers perfect for battery powered scenarios.
Accelerometers have a range of forces that they can measure. These ranges can vary from ±1g up to ±250g, the smaller the range, the more sensitive the readings will be from the accelerometer.


The accelerometer sensor I researched is the Adafruit LIS3DH, I picked this sensor because it had the lowest power draw of the sensors that I looked at.
The power draw is only 2uA which is good because part of the project is to try and the power draw as low as possible. I choose to research an accelerometer sensor as I thought this would be the sensor that would most fit the project as it can detect any movements of the lifebuoy.

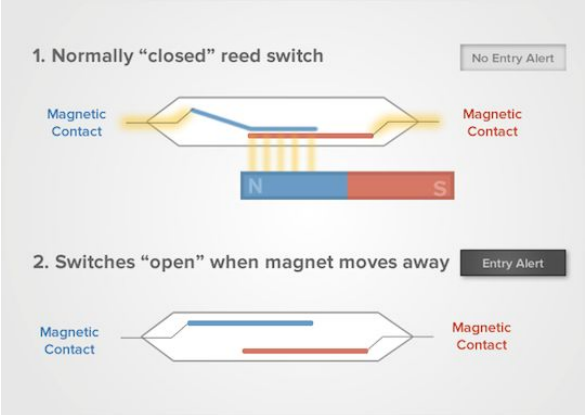
## Altimeter

An altimeter sensor measures the altitude of the device above a fixed level. The altitude measurement is based off of atmospheric pressure, as the pressure drops the altitude increases. The sensor I researched is the MPL3115A2, this sensor provides pressure, temperature, and calculates altitude (based on pressure) it also has a low power draw and is accurate to within 30cm, there are more accurate sensors available but they also have a much higher power draw.
I looked at the altimeter sensor because it seemed like a good alternative to the accelerometer sensor, although it would only give me a reading in the y axis..



## Magnetic Contact Switch

The last sensor/switch i looked at is a magnetic contact switch. Magnetic contact switches work by having a magnet and a reed switch. When a magnet is pressed against a reed switch it will close the circuit, as soon as this circuit is opened an event will be triggered. I decide to research magnetic contact switches as I thought this would be a simple solution to the problem. This was because if the circuit gets broken, the user will know that the lifebuoy was either used or moved.

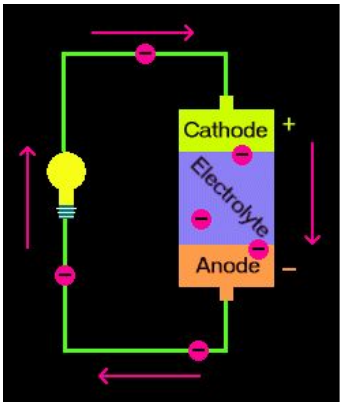



## Conclusion

In the end i decided to go with the Altimeter/Accelerometer as it meet the requirements i had for this project. These were small size, low power requirements and low price. But the  main benefit was the available libraries that would make the coding easier and quicker.

# Batteries

The batteries I chose to research were chemical batteries(Lithium, Alkaline and NiMH) because these are relatively cheap and the most common battery type available. A chemical battery works by a chemical reaction that happens within the battery, this reaction causes a buildup of electrons at the anode. This results in an electron difference between the anode and cathode, the electrons want to rearrange themselves to get rid of the difference but the electrolyte stops the electrons from going straight to the cathode.




In the above picture, the electrons go through the wire and allowing the electrons reach the cathode and along the way it lights the bulb.

## Lithium-ion

A lithium-ion battery is a type of rechargeable battery in which lithium ions move from the negative electrode to the positive electrode during discharge and back when charging.

Pros

- Can be made in various shapes and sizes, this is helpful because the lithium-ion battery can be made to perfectly fit the available space within a device.
- Low discharge rate compared to other battery types(5% compared to 30% per month)
- Lithium-ion batteries also have the greatest energy density. Energy density is the power per weight ratio of a battery.
- Can be recharged.

Cons

- Lithium is highly volatile and can catch fire.
- Some Lithium-ion batteries can lose 20% or more of their capacity a year irrespective of the amount of times they were charged.
- Lithium-ion batteries also cost more than other types of batteries.

## Alkaline

Alkaline batteries are disposable batteries with use zinc and manganese dioxide as its electrodes. The alkaline electrolyte used is either potassium or sodium hydroxide. These batteries have a steady voltage offering better energy density and leakage resistance than carbon zinc batteries.

Pros

- Comes in all sizes which is good for multiple uses.
- Cheap compared to other chemical batteries.
- Alkaline batteries are widely available.

Cons

- Cannot usually be recharged.
- Alkaline batteries are more prone to leaking when in use compared to other kinds of batteries.
- Alkaline batteries also have poor performance in high drain devices(digital cameras).

## NiMH

NiMH is a type of rechargeable battery. The chemical reaction at the positive electrode using nickel oxide hydroxide and the negative electrodes use a hydrogen-absorbing alloy instead of cadmium. A NiMH battery can have two to three times the capacity of an equivalent size NiCd, and its energy density can approach that of a lithium-ion battery.

Pros

- High capacity(up to 11,000mah).
- Contain no toxic materials, so easier to dispose of.
- Can also be recharged hundreds of times.
- Economically priced; NiCd is the lowest in terms of cost per cycle

Cons

- More expensive than alkaline batteries.
- Self-discharge quickly, for example if they are left unattended for a month or longer they can lose all charge and will then need to be recharged before use.
- Low cell voltage of 1.20V requires many cells to achieve high voltage

| Battery type | Capacity AA cell | Voltage | Self-discharge Capacity after 1 year storage | Runtime Estimated photos on digital camera |
|---|---|---|---|---|
| **NiMH** | 2,700mAh, rechargeable | 1.2V | 50% | 600 shots |
| **Eneloop*** | 2,500mAh, rechargeable | 1.2V | 85% | 500 shots |
| **Regular alkaline** | 2,800mAh; non-rechargeable | 1.5V | 95% 10 year shelf life | 100 shots |
| **Reusable alkaline** | 2,000mAh; lower on subsequent recharge | 1.4V | 95% | 100 shots |
| **Lithium** (Li-FeS2) | 2,500–3,400mAh (non-rechargeable) | 1.5V | Very low 10 year shelf life | 690 shots |

## Conclusion

In the end i decided to go with the Lithium-ion. I chose Lithium-ion because it has a low discharge rate compared to other battery types, they have the greatest energy density and can be recharged. The only concern of using Lithium-ion is that it is highly volatile and can easily catch fire, this is a concern as the device will be exposed to the elements

# Power Requirements and Optimization

The amount of power the development board uses is important for this project. The lifebuoy sensor will have to work for extended periods of time on a portable power source(battery).
From my research there is only one way of achieving this, through software and hardware changes to the development board.

## Hardware

I will list below the hardware power saving techniques I found during my research.

- LED - Many of the development boards out there come with 2 to 3 LEDs, these are usually used for showing when the board is getting power or for warning lights. These LEDs can use 10-15 mA, so just by removing these LEDs you can save up to 45mA. LEDs can be removed either by de soldering or by cutting the tracks on the board. Disabling the LEDs can also be a hindrance when trying to find a fault on the board.
- Barebones Build - A bare bones build is when you buy a breadboard and only the individual parts needed for the build. This cuts out unnecessary parts that would otherwise drain the batteries faster. This solution would not work for me as i do not have good enough knowledge in electronics to complete a bare bones build.
- Linear Regulator - Linear regulators change a higher voltage to a lower voltage, this change cause the linear regulator to heat up. This heat is wasted power, to get around this you could use switch mode power supplies which are up to 90% more efficient.

## Software

I will list below the software power saving techniques I found during my research.

- Lower Voltage - Most boards run on 5V but can still operate at 3.3V. When reducing the voltage to a board you might have to reduce the clock speed too or the board can behave in unexpected ways.
- Sleep mode - The biggest saving in power usage would be putting the processor to sleep and have it wake when an event occurs.

## Conclusion

In the end I decided to go with software power optimization over hardware because software optimizations can be accomplished through the use of libraries whereas hardware optimizations require you to physically modify the device and this is something with which i have no experience in. Although I believe the biggest power saving is to be made from hardware power notifications as i believe LEDs are one of the biggest power draws.

# Native versus Web Apps

## Native

A native application is an app developed specifically for a specific platform(Android or IOS). Since the apps are written for a specific platform, the app can easily use the devices features including Camera, Contacts, built in sensors and can access the notification system.
Native apps are downloaded from an application store including the Play store(Android) and Itunes(IOS). Examples of native apps would be MyFitnessPal, Snapchat and Whatsapp.

Pros

- Speed - Native apps offer the fastest and most responsive experience to the user because they are usually optimized to fully take advantage of the device's hardware and the features provided by the operating system.
- Ease of Discovery- Through the application stores built in search feature, it is easy to find an app that fits your needs.
- Offline - Many native apps can still provide their intended functionality even without internet access.
- If a user downloads an app from an application store, they can be reassured the app is safe. This is because for an app to be listed on the app store, they have to meet specific standards set.

Cons

- Multiple code bases - If a developer wants to release their app onto multiple platforms they will need to have multiple code bases. The developer will either have to have a team working on the app for each platform or they will have to concentrate on one platform and then when finished move onto the next platform. Either way this leads to an increase in time and cost.
- Cost of Upkeep and Approval - The cost of keeping an app maintained and updated is also higher for native apps as they need to stay current with new standards. Getting an app approved for an app store can prove to be long and drawn out for the developer and there is no guarantee of success when approval is granted.

## Web Apps

A web app is like a native app but it isn't developed to work on only one platform(Android, IOS), it can run on any device that can connect to the internet. Web apps can be nearly identical to the look of native apps, also web apps can access nearly all the devices features through use of an API(Application Programming Interface) like GPS, Accelerometer and Camera.

Pros

- Cost - An advantage of developing an web app is once developed is can run on any device connected to the internet. This leads to a cost and time saving for the developer.
- Access anywhere - An web app can be accessed anywhere(PC, Smartphone and Tablet) and with any device once internet access is available.
- Updates - Quick updates are possible as the developer does not have to worry about updates being held up by app stores enforced standards.
- Device Support - In the market today there is countless different devices and operating system with many versions. A web app can support all these devices and operating system versions once they are capable of connecting to the internet. This gives the developer access to a huge pool of potential users.
- Data - Since a web app isn't installed locally on a device, the data a user enters is processed and saved remotely. This means the user can access their data across multiple devices.

Cons

- Offline - If a web app cannot connect to the internet, the web app becomes virtually useless as the app isn't stored locally on the device.
- Quality Control - Unlike native apps that have an app store that enforce their own standards, web apps don't have one. This can lead to the web app being unsafe in terms of security for the user.
- Discoverability - Like the point above, web apps can be difficult to find as they have no curated store for the user to search through. This can lead to a useful app going unnoticed by users.

## Hybrid App

Hybrid apps is very similar to native apps but instead of been written in Java or Swift, it is written in HTML, CSS, and JavaScript(like a web app). The difference between hybrid apps and web apps is that hybrid apps are encompassed inside a native app which in turn uses the devices WebView. This in turn allows the hybrid app to access the functionalities of the device(camera,GPS and contacts), these are often inaccessible to web apps.
Most hybrid apps make use of Apache Cordova which is a platform that provides JavaScript APIs to access device capabilities, these are built with native code. Unlike web apps, hybrid apps can be published to an app store which makes it more like native apps.

Pros

- Lower requirements - For hybrid apps you don’t have to learn any of the specific languages for iOS or Android.
- Lower cost - Developing a hybrid app is an easier process than in developing a native app. You don't have to create separate versions of the app for iOS and Android, instead you will have one app that will work on both platforms.

Cons

- Performance Issues - A hybrid app will never offer the kind of seamless experience of native apps. They are less refined and might cause performance problems in certain high requirement apps.
- Lack of certain native features - Android has features which are only available for their native apps. One example of this is RenderScript.RenderScript is a framework for running computationally intensive tasks at high performance on Android. Hybrid apps have no access to such features and thus might be more difficult to use.

## Conclusion

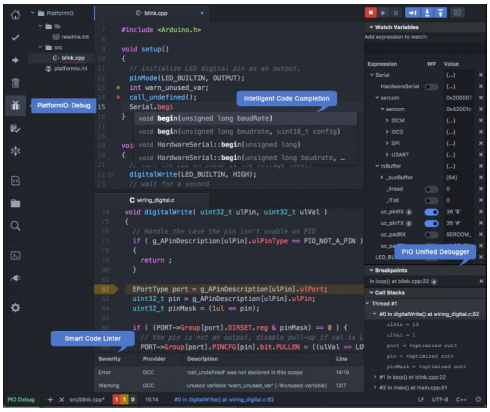
At the start I decided to create a web app, I created a basic version of the website with some functionality but as I got more into the project i realised that a native or hybrid app would suit my needs better. I decided to go with a hybrid app over native as I had more familiarity with HTML and Javascript and less experience with Android Studio and Java.

# Development Tools

From my research into IOT devices, I realized the best suited development board for my needs was the Arduino MKR FOX 1200. So when researching development tools, I concentrated on Arduino IDE's. I looked at 4 different IDE's, these were PlatformIO IDE, Embrio, Programino IDE and Arduino IDE.

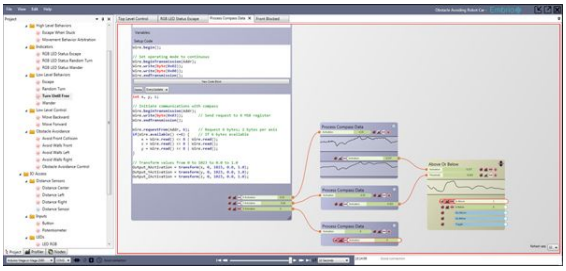
## PlatformIO IDE

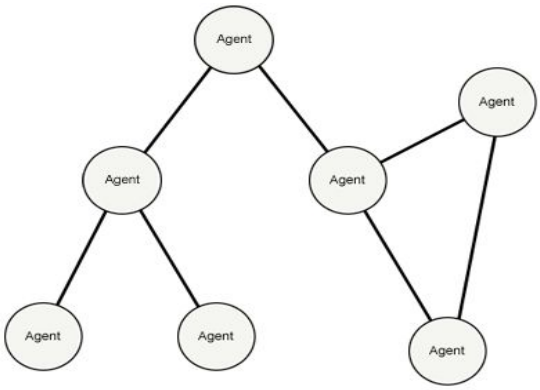
PlatformIO is described as the next generation integrated development environment for IOT devices. PlatformIO is open source supports over 400 development boards and over 20 development platforms.
This IDE includes intelligent code completion and a smart code linter which provides analysis of your code for potential errors. This IDE comes as a plugin for Github's Atom and Microsoft's VSCode. It runs on Linux, Windows and MacOS. The cost of PlatformIO is £0.




## Embrio IDE

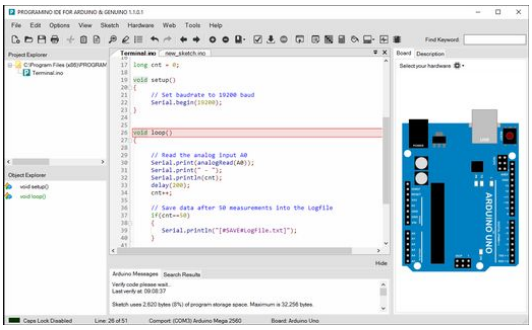
Emrio is a real time, agent based IDE for Arduino. Agent based programming is when each agent has one job and is always running, agents work together to solve the problem. This IDE comes as a standalone IDE and runs on Linux, Windows and MacOS. Embrio is not open source but has a free version which allows you to do everything except upload a final program to the Arduino. The yearly cost of Embrio is £70.





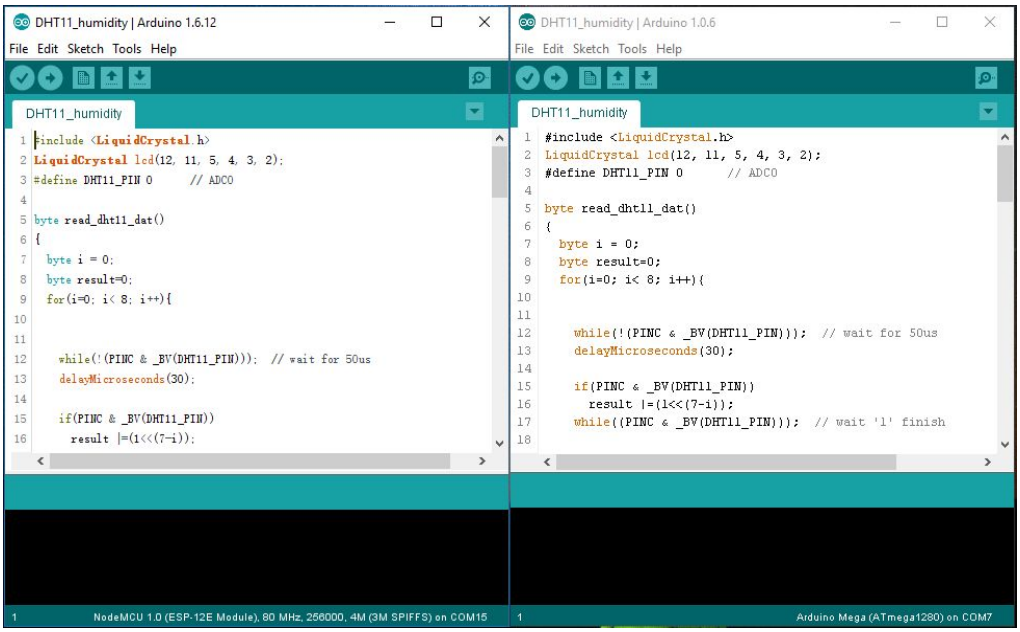

## Programino IDE

Programino is a low priced IDE for Arduino and is aimed at “Makers” and “Professionals”.  It comes with a hardware viewer so you always have an overview of your board when programming. This IDE comes as a standalone IDE and runs on Windows. Programino is not open source. The IDE requires a license key which costs £29.



# Arduino IDE

Arduino integrated development environment (IDE) is a cross-platform application. It It includes a code editor and provides simple one-click mechanisms to compile and upload programs to an Arduino board. The source code for the IDE is released under the GNU General Public License
The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures.




# Conclusion

In the end I decided to go with the Arduino IDE as they create the board I would be basing my project on. The IDE has all the features that I need such as ability to add in libraries, code in C++ and it is also free. Also the IDE is very lightweight in terms of laptop requirements and it has the ability to compile the program before uploading, this allows the users to see any mistakes before it gets uploaded to the board.

# Databases

A database is a large quantity of indexed data, also it can be searched, filtered and updated. A database provides an efficient way of handling large amounts of data. There are many types of databases(Relational, non-Relational and Hierarchical), I will be focusing on Relational  database's as these are the ones i have the most experience with.

## Relaţional

A relational database is a collection of tables that have a defined relationship and use a table/key model. These relationships can be either, One to One, One to Many, Many to One or Many to Many. Examples of relational databases are Microsoft Access, MySQL and Microsoft SQL Server.

Pros

- Known - Relational Databases are well established and known as they are around since 1979.
- Queries - Complex queries can be easily carried out.
- Data Duplication - In relational databases it is easy to avoid data duplication.
- Expandability - If data is in separate tables, it is easy to add data that isn't needed yet but may be required in the future.

Cons

- Cost - The cost of setting up and maintaining a relational database can be high depending on the size and complexity needed.
- Limits - Some relational databases have a limit on the field length, this can lead to a loss of data if the inputted info is longer than the given size.

# NON-RELATIONAL

Relational databases have tables whereas, NoSQL databases are document-oriented. This way, unstructured data (such as articles, photos, and video) can be stored in a single document that can be easily found but isn't necessarily categorized into fields like a relational database is.
There are four main kinds of NoSQL database

- Document database. Inserted data is stored in the form of free-form JSON structures or "documents". There is no inherent need to specify what fields, if any, a document will contain.
- Key-value stores. Free-form values are accessed in the database by way of keys.
- Wide column stores. Data is stored in columns instead of rows as in a conventional SQL system.
- Graph databases.  Data is represented as a network or graph of entities and their relationships, with each node in the graph a chunk of data.

A database provider I looked at was Firebase database, which is a realtime, NoSQL database, Data is stored as JSON and synchronized in realtime to every connected client.

Pros

- Big data - The amount of data that is being created and stored daily has increased hugely(Big Data). Relational databases capacity has been growing to match these increases, but they are getting to a point where it is becoming unmanageable to the owner. Today, the volumes of "big data" that can be handled by NoSQL systems, such as Hadoop, outstrip what can be handled by the best Relational database.

Cons

- Expertise - Nearly every developer will have some experience with Relational databases whereas many developers will have little to no experience with a NoSQL.
- Changing - With the speed of technological advancement, it becomes difficult to keep up with the functions and syntax required to use a NoSQL database. I have personal experience of this with Firebase, i haven't used it in a year but even in this time frame the changes are huge.

## Conclusion

In the end i decided to go with the Firebase Realtime database as i had some previous experience and it also allowed me to insert my data quickly and easily. It also provides the ability to create cloud functions which will benefit me in the sorting and filtering of data, as well as creating push notifications.

# BIbliography

Wikipedia. (). *Sigfox.* Available: https://en.wikipedia.org/wiki/Sigfox. Last accessed 03/11/2017.

Lava Group. (). *Sentry Lifebuoy Monitoring.* Available: http://www.thelavagroup.co.uk/site/assets/files/1512/sentry_print-1.pdf. Last accessed 01/11/2017
.
MicroChip. (). *ATSAMD21G18.* Available: http://www.microchip.com/wwwproducts/en/ATsamd21g18. Last accessed 01/11/2017.
Arduino. (). *Genuino Zero.* Available: https://store.arduino.cc/genuino-zero. Last accessed 02/11/2017.

Arduino. (). *Compare board specs.* Available: https://www.arduino.cc/en/Products/Compare. Last accessed 02/11/2017.

Cooking-Hacks. (). *Sigfox module for Arduino.* Available: https://www.cooking-hacks.com/sigfox-module-for-arduino-waspmote-raspberry-pi-intel-galileo-868-mhz-7184. Last accessed 02/11/2017.

Social-Compare. (). *RaspberryPI models comparison.* Available: https://socialcompare.com/en/comparison/raspberrypi-models-comparison. Last accessed 03/11/2017.

Dimension Engineering. (). *A beginner's guide to accelerometers.* Available: https://www.dimensionengineering.com/info/accelerometers. Last accessed 03/11/2017.

AdaFruit. (). *MPL3115A2 - I2C BAROMETRIC PRESSURE.* Available: https://www.adafruit.com/product/1893. Last accessed 03/11/2017.

Cara Giaimo. (2013). *Door Sensor Secrets: What They Are, How They Work & 6 Unexpected Uses.* Available: https://simplisafe.com/blog/door-sensor-secrets. Last accessed 04/11/2017.

Walker Rowe. (). *IoT battery outlook: Types of batteries for IoT devices.* Available: http://internetofthingsagenda.techtarget.com/feature/IoT-battery-outlook-Types-of-batteries-for-IoT-devices. Last accessed 04/11/2017.

qrg northwestern. (). *Power System.* Available: http://www.qrg.northwestern.edu/projects/vss/docs/power/2-how-do-batteries-work.html. Last accessed 04/11/2017.

Nick Gammon. (2012). *Power saving techniques for microprocessors.* Available: http://www.gammon.com.au/power. Last accessed 05/11/2017.

ALEX THE GIANT. (). *Reducing Arduino Power Consumption.* Available: https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption#reducing-the-clock-speed. Last accessed 05/11/2017.

Priya Viswanathan. (2017). *The Pros and Cons of Native Apps and Mobile Web Apps.* Available: https://www.lifewire.com/pros-and-cons-of-native-apps-and-mobile-web-apps-2373173. Last accessed 02/11/2017.

MobiLoud. (). *Native, Web or Hybrid Apps.* Available: https://www.mobiloud.com/blog/native-web-or-hybrid-apps/. Last accessed 04/11/2017.

Imobdevtech. (2016). *What is a web app? Pros & Cons of using them.* Available: https://www.imobdevtech.com/Blog/what-is-a-web-app-pros-cons. Last accessed 05/11/2017.

Techterms. (). *Web Application.* Available: https://techterms.com/definition/web_application. Last accessed 05/11/2017.

Arduino. (). *Development Tools.* Available: https://playground.arduino.cc/Main/DevelopmentTools. Last accessed 05/11/2017.

Embrio. (). *Agent Based Programming.* Available: http://www.embrio.io/documentation/agent-based-programming. Last accessed 05/11/2017.

Programino. (). *PROGRAMINO IDE for Arduino.* Available: http://www.programino.com/. Last accessed 05/11/2017.

Guru99. (). *What is Database? What is SQL?.* Available: https://www.guru99.com/introduction-to-database-sql.html. Last accessed 05/11/2017.

Computer Hope. (). *Database.* Available: https://www.computerhope.com/jargon/d/database.htm. Last accessed 05/11/2017.

Keith D. Foote. (2016). *A Review of Different Database Types: Relational versus Non-Relational.* Available: http://www.dataversity.net/review-pros-cons-different-databases-relational-versus-non-relational/. Last accessed 05/11/2017.

unknown. (2018). *Connecting Sigfox.* Available: https://build.sigfox.com/steps/study. Last accessed 05/11/2017.