THE UNIVERSITY OF THE WEST OF ENGLAND

FACULTY OF ENVIRONMENT & TECHNOLOGY

FRENCHAY

Portable Air Quality and Pollution Monitoring Sensor

BSc Computer System Integration

UFCFR4-45-3 - Computing project

Merlin Roe

Student Number: 12005506

Supervisor : Craig Duffy

Second Reader : Julian Webb

Word Count : 14,203

Word Count (Excluding Tables) : 12,529

# Abstract

The projects purpose was to research and implement air quality and pollution tracking technologies within a portable embedded device. Exploring cheap hardware alternatives and free open source tools, to offer a thorough and cost effective form of monitoring.

While offering a simple yet comprehensive way to query and track the air quality, using an interactive heat map from the data collected.

# Acknowledgement

I would like to express my thanks to the following people, whom without this project would not have been completed.

Craig Duffy for his guidance and continues support through the project as my supervisor, and Ian Johnson and Julian Webb for giving me incredibly useful feedback during the first demonstration session.

I would like to thank my family Deana Roe, Anthony Roe and Sarah Roe for helping proof read parts of my project. Encouraging and Supporting me throughout its development and my studies.

I would also like to extend my thanks to the university technicians, for their help in the electrical side of implementation.

Finally, I wish to thank the Redbear Forum, Particle Forum and other online communities, for offering free help feedback and developing open source libraries to use and learn from.

Table of Contents

[1 Abstract 2](#_Toc478992516)

[2 Acknowledgement 3](#_Toc478992517)

[4 Introduction 8](#_Toc478992518)

[4.1 Background Research 8](#_Toc478992519)

[4.2 Aims 11](#_Toc478992520)

[4.2.1 Objectives 11](#_Toc478992521)

[4.2.2 Personal objectives 11](#_Toc478992522)

[5 Research 12](#_Toc478992523)

[5.1 Active solutions 12](#_Toc478992524)

[5.2 Internet of things 13](#_Toc478992525)

[5.3 Open Source 14](#_Toc478992526)

[5.4 Operating System 15](#_Toc478992527)

[5.5 Embedded Systems 15](#_Toc478992528)

[5.6 Embedded Systems options 15](#_Toc478992529)

[5.6.1 Raspberry Pi 3 – B 16](#_Toc478992530)

[5.6.2 Particle Electron 17](#_Toc478992531)

[5.6.3 Redbear Duo 18](#_Toc478992532)

[5.6.4 MSP-EXP430F5529LP 19](#_Toc478992533)

[5.7 Temperature and Humidity sensors 20](#_Toc478992534)

[5.7.1 Temperature and Humidity sensor comparison 20](#_Toc478992535)

[5.8 Particulate Matter Sensor 21](#_Toc478992536)

[5.8.1 Particulate Matter sensor comparison 22](#_Toc478992537)

[5.9 Gas Sensor 23](#_Toc478992538)

[5.9.1 Gas sensor comparison 24](#_Toc478992539)

[5.10 GPS 25](#_Toc478992540)

[5.10.1 GPS comparison 26](#_Toc478992541)

[5.11 Networking comparison 27](#_Toc478992542)

[5.11.1 Wi-Fi 27](#_Toc478992543)

[5.11.2 LoRaWAN (Low power wide area network) 28](#_Toc478992544)

[5.11.3 Bluetooth low energy 28](#_Toc478992545)

[5.11.4 GPRS 29](#_Toc478992546)

[5.12 Power source 30](#_Toc478992547)

[5.12.1 Alkaline batteries 30](#_Toc478992548)

[5.12.2 Lithium-ion Polymer battery (Li-Po) 30](#_Toc478992549)

[5.12.3 Power source comparison 31](#_Toc478992550)

[5.13 Database and Web server 32](#_Toc478992551)

[5.13.1 MySQL introduction 32](#_Toc478992552)

[5.13.2 Database and Web server comparison 33](#_Toc478992553)

[6 Requirements 34](#_Toc478992554)

[6.1 Introduction 34](#_Toc478992555)

[6.2 General overview 34](#_Toc478992556)

[6.2.1 Embedded Board hardware 35](#_Toc478992557)

[6.2.2 Temperature and Humidity sensor 35](#_Toc478992558)

[6.2.3 Particulate Matter sensor 36](#_Toc478992559)

[6.2.4 Gas sensor 36](#_Toc478992560)

[6.2.5 GPS 37](#_Toc478992561)

[6.3 Requirements 37](#_Toc478992562)

[6.4 Requirements Priority 37](#_Toc478992563)

[6.5 Requirements Table 38](#_Toc478992564)

[6.5.1 Post Requirements 38](#_Toc478992565)

[6.5.2 General Requirements 38](#_Toc478992566)

[6.5.3 Warnings and Fault Detection Requirements 39](#_Toc478992567)

[6.5.4 Database Requirements 39](#_Toc478992568)

[6.5.5 Website Requirements 40](#_Toc478992569)

[6.5.6 Security Requirements 40](#_Toc478992570)

[7 Design 41](#_Toc478992571)

[7.1 Introduction 41](#_Toc478992572)

[7.2 Overview 41](#_Toc478992573)

[7.3 Finite State Machine 43](#_Toc478992574)

[7.4 Hardware Design 44](#_Toc478992575)

[7.4.1 Embedded Board and Operating System 44](#_Toc478992576)

[7.4.2 Temperature and Humidity design 44](#_Toc478992577)

[7.4.3 Particulate sensor design 45](#_Toc478992578)

[7.4.4 Gas sensor design 45](#_Toc478992579)

[7.4.5 GPS Design 47](#_Toc478992580)

[7.4.6 Wi-Fi Design 47](#_Toc478992581)

[7.5 Battery Component 48](#_Toc478992582)

[7.6 Hardware schematic 49](#_Toc478992583)

[7.7 Database and Website 50](#_Toc478992584)

[7.7.1 Database 50](#_Toc478992585)

[7.7.2 Website 51](#_Toc478992586)

[8 Implementation 52](#_Toc478992587)

[8.1 Cross development 52](#_Toc478992588)

[8.1.1 Host machine 52](#_Toc478992589)

[8.1.2 Toolchain 52](#_Toc478992590)

[8.1.3 Version Control - GIT 52](#_Toc478992591)

[8.2 Implementation plan 53](#_Toc478992592)

[8.3 Cross compiling 53](#_Toc478992593)

[8.4 Temperature and Humidity - DHT22 54](#_Toc478992594)

[8.4.1 Testing 54](#_Toc478992595)

[8.5 GPS – GP-20U7 55](#_Toc478992596)

[8.6 Particulate sensor - PPD42NJ 55](#_Toc478992597)

[8.6.1 Testing 56](#_Toc478992598)

[8.7 CO2 gas sensor – MQ135 57](#_Toc478992599)

[8.7.1 Testing 58](#_Toc478992600)

[8.8 CO Gas sensor – MQ7 59](#_Toc478992601)

[8.8.1 Testing 60](#_Toc478992602)

[8.9 Voltage issues 61](#_Toc478992603)

[8.10 Networking 62](#_Toc478992604)

[8.10.1 Finding a network 62](#_Toc478992605)

[8.10.2 Uploading results 62](#_Toc478992606)

[8.11 Database 63](#_Toc478992607)

[8.12 Website 64](#_Toc478992608)

[9 Testing 65](#_Toc478992609)

[10 Conclusion 67](#_Toc478992610)

[10.1 Project goal 67](#_Toc478992611)

[10.2 Critical evaluation 67](#_Toc478992612)

[10.3 What did I learn 67](#_Toc478992613)

[10.4 Summary 68](#_Toc478992614)

[11 Future Work 69](#_Toc478992615)

[11.1 Switching to a Microcontroller 69](#_Toc478992616)

[11.2 Using Bluetooth 69](#_Toc478992617)

[11.3 Calibrating the Gas sensors 69](#_Toc478992618)

[11.4 Improved interface 70](#_Toc478992619)

[12 List of Figures 71](#_Toc478992620)

[13 List of Illustrations 72](#_Toc478992621)

[14 List of Tables 72](#_Toc478992622)

[15 References 73](#_Toc478992623)

[16 Appendices 78](#_Toc478992624)

# Introduction

## Background Research

Pollution and air quality has been worsening ever since the Industrial revolution, which is causing huge impacts on our environment and our health in the form of poor air quality. The World Health Organisation estimates air pollution contributes around 800,000 premature deaths a year. The problem is deeper than its commonly known, with many studies focusing on specific causes of effect due to air pollution. With air pollution contributing to cardiovascular and cardiovascular disease causing premature death, with a long-term effect leading up to a 76% effect on Cardiovascular mortality.

1. ( Anderson, et al., 2012)

As of 2016 our PPM (parts per million) for Carbone Dioxide (Co2) has reached a permanent high of over 400, to give some perspective of the increase the average PPM before 1950 was 300. Below in Figure 1 is a diagram from NASA who have been tracking Co2 levels, which accurately show the drastic increase in pollution and harmful particles within our air.

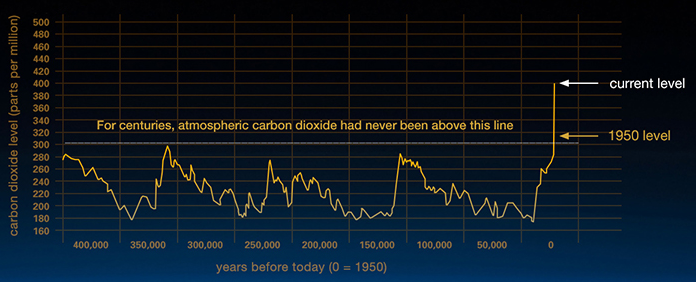
1. (David, n.d.)
2. 

Figure 1 Rise in Carbon Dioxide to 400ppm in 2016, Climate NASA. Available from: <http://climate.nasa.gov/climate_resources/24/> [Accessed 25 October 2016].

Co2 is the most widely known pollutant, however it’s not the only damage particle being generated. Nitrogen Oxides (NOx) is a generic term used for Nitric Oxide (NO) and Nitrogen Dioxide (NO2). The NOx group of particles also have a negative effect on our environment, however it also has a huge impact on our own health. Carbon Monoxide (CO) is a gas which is short lived however has harmful effects on humans.

NOx, CO2 and CO gasses are generated from nitrogen and hydrocarbons during combustion, and increase within higher temperatures; which is exactly how a cars combustion engines operate. There are several emission standards in place to help combat and minimise the pollution engines produce, however in dense urban areas where cars are frequent or often caught in traffic the emissions build up and stay low within our breathing air for long periods of time.

1. (Puri & Annamalai, 2006) (Kampa & Castanas, 2008)

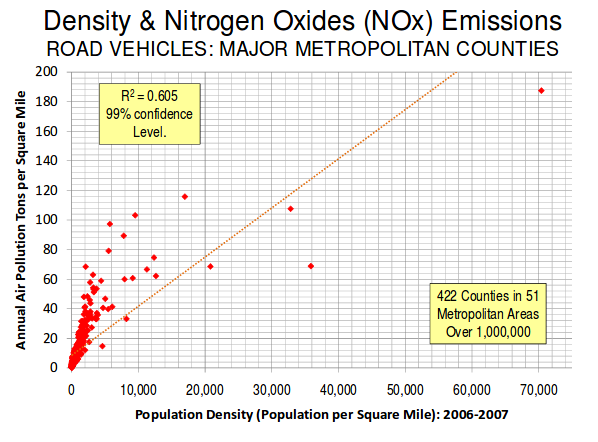


Figure 2 Density & Nitrogen Oxides (NOx) Emissions, Demographia. Available from: <http://www.demographia.com/> [Accessed 15 February 2017].

As seen in Figure 2, the population density has a significant increase with NOx particles present per square mile. While this has a negative effect on the environment, it’s the effect on air quality that’s most worrying. While NOx exposure has been associated with mortality, there is still a lack of evidence to draw a conclusive correlation between NOx exposure and mortality rates; this is primarily due to the complex relationships particles have within dense areas. NOx is however still responsible for the formation of smog and acid rain, with studies having proven that long-term exposure leads to decrease in lung function and an increased risk in respiratory symptoms such as phlegm and acute bronchitis. These effects are only worsened with more exposure and higher intake in air, which directly effects people’s daily exercise routines such as, cycling and jogging. While these activities are healthy, the effect is countered by the negative effect of the air quality they are breathing.

(WHO Working group, 2003)

Carbon Monoxide (CO) is colourless and odourless gas is generated from incompletion of the combustion engine, which lives temporarily within the atmosphere. CO is a highly toxic gas, effecting humans similar to CO2 by combining haemoglobin within red blood cells into carboxyhaemoglobin; which inhibits the space where oxygen is stored within red blood cells. Limiting the oxygen being passed through the body can have catastrophic health effects, a level of 50% carboxyhaemoglobin results in coma, seizures and possible fatality.

1. (Tikuisis, et al., 1992)

Air pollution is being actively countered from local authorities and from wider reaching groups such as the European Union. As previously mentioned car emissions are being regulated, these regulations are to set quantitative limits on permissible amount of pollutants released from specific sources over specific timeframes. For instance, the Euro-5 emission standards were adopted by the UK in 2009, this regulation ensures cars sold within Europe meet certain standards for exhaust emissions.

1. (Office, 2007)

The city of Bristol has been on the forefront within the UK at tackling air pollution. Bristol has been awarded the title of “European Green Capitals” in 2005 and in 2015, this is due to the large budget invested into energy efficiency and renewable energy; while continually lowering the Carbon emissions. This was achieved by multiple strategies such as increasing the amount of people opting to travel via bicycle’s over cars, which greatly reduce the number of cars on the road lowering the overall air pollution released.

“Carbon emissions have consistently reduced in Bristol since 2005, despite a growing economy. Bristol has the ambition of becoming a European hub for low-carbon industry with a target of 17,000 new jobs in creative, digital and low carbon sectors by 2030. Bristol demonstrated 4.7% growth in the green economy in 2012.”

1. (European.eu, 2017)

Bristol was awarded the 2015 Green capital partly based on the updated plan for limiting NO2 within Bristol’s urban areas *“Air Quality Plan for the achievement of EU air quality limit value for nitrogen dioxide (NO2) in Bristol Urban Area (UK0009)*”. This plan is an updated version earlier released in 2011.

(Department for Environment, 2015)



Bristol is currently monitoring its air quality and emissions using five sensors around the city established in 1994, all of which are focusing on monitoring NOx gases. These monitors are very accurate when taking readings but lack a wide area coverage, by only collecting within 20-30 metres and upwards of 3 metres. Limiting to area’s air quality is tracked within the city of Bristol.

(Bristol City Council, 2017)

Illustration 1 NOx Monitor in Fishponds, Bristol Air Quality, Avaiable from: <http://www.bristol.airqualitydata.com/cgi-bin/sites.cgi?1010> [accessed 22 February 2017]

## Aims

The aim of this project will be to explore the practicality and usage of small, portable and cheap air quality and pollution monitors, which can be disturbed around a wide and dense area. Specifically, something that can detect particles and or gasses in a small hand held size device.

The data monitored should result to an easy and cheap pollution map designed for tracking air quality within large densely populated areas. While offering individual people to track the amount of pollution they are exposed to on a daily basis, A combination of data can build up a city-wide map that can be used by local governments to tackle pollution and easily pin point areas of high pollution.

### Objectives

This project objectives are as follows:

* RTOS or Linux based controller
* Long lasting battery life
* Detect airborne particles
* Monitor pollutant gasses
* Track location
* Automation
  + Monitoring
  + Network connection
  + Data upload
  + Error and Fault detection
* Large online storage
  + Simple and effective interface

### Personal objectives

This project offers the opportunity to expand my own personal objectives, such as:

* To further develop my skills and understanding in the process of cross development
* Gain knowledge in controlling delicate hardware components such as gas sensors.
* Learn the process of developing a system from the hardware perspective.

# Research

## Active solutions

As mentioned within the background section, pollution and air quality have been issues of concern; this has led to numerous solutions for tackling and monitoring air quality; particularly within cities.

One of which was briefly discussed within the background section, that being the Bristol air quality monitoring (Bristol City Council, 2017)*.* This network of air quality sensors consists of five sensors located in and around the city of Bristol, four of these sensors detect the group of NOx gasses; while one of them measures the weather. These five sensors are the primary source of air quality monitoring within Bristol, which are also part of the “*Air Quality Action Plan”* (Department for Environment, 2015). These sensors are extremely accurate but are limited in number and range. The “*Wells Road monitoring station*” reads a sample of 1.5 metres high and 1 metre from the kerb which limits its city-wide coverage quite drastically.

Another active solution is the *AirBeam*, which is a commercial device by *AirCasting* designed to be a wearable, open source end to end active solution for consumers to collect and share their health and local air quality information. *AirBeam* has multiple sensors allowing it to monitor sound levels, temperature, humidity, fine particulate matter (PM2.5), carbon monoxide (CO), nitrogen dioxide (NO2) and the wearer’s heart rate. While *AirBeam* itself does not upload the results, it monitors the wearer and local area and sends this information to the consumer via Bluetooth. The aim of this product is primarily to monitor its wearer along with the air quality they are exposed too, with its target audience being “*Citizen scientists*” and “Changemakers”. The *AirBeam* is marketed as low cost and portable, portability side an individual unit costs $250.

(Heimbinderc, et al., 2016)

The *Mobile Air Quality Monitoring Network* (MAQuMoN) is an actively running project discussed in the conference in “*Air Quality Monitoring with SensorMap”* (Völgyesi, et al., 2008)*.* This network is built up from multiple air quality and weather sensors which are consistently moved around the city of Nashville, Tennessee, USA by attaching the devices to cars. The aim of this project was to resolve the issue of location-dependent air quality sensors, the current solution in the city was useful but failed to give enough data to evaluate air quality in any given neighbourhood.

1. (Institute for Software Integrated Systems, 2011)

## Internet of things

The *Internet of Things* (IoT) is the networking of physical to the internet, enabling communication with other devices to collect and exchange data. The aim of these devices is to make the internet even more immersive and pervasive, while enabling easy access and interaction with a wide variety of devices from home applicants such as central heating to monitoring sensors and vehicles. The applications of IoT devices is not limited and has a wide range of different domains. It can be incorporated into various areas such as home automation, industrial automation, medical aids, mobile healthcare, elderly assistance, intelligent energy, environment monitoring and traffic management.

1. (Zanella, et al., 2014)

“Generally speaking, IoT refers to the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence. IoT will increase the ubiquity of the Internet by integrating every object for interaction via embedded systems, which leads to a highly-distributed network of devices communicating with human beings as well as other devices.”

(Xia, et al., 2012)

As Cities develop the demand for adequate provision of services and infrastructure are increasing to meet the needs of the cities inhabitants, leading to a slow adoption of IoT devices into the infrastructure to aid in managing roads and traffic to managing public transport systems. According to a journal article called *“An Information Framework for Creating a Smart City Through Internet of Things*” posted in IEE (Jin, et al., 2014) By 2050 an expected 70% of the world’s population will be living in cities and the surrounding areas, for a city to survive such a large population it needs to be connected for economic, environmental and social growth; this inadvertently grows the IoT.

While the idea and implementation of a smart city is still being developed, there are several areas where its use is in effect. Within Bristol is the “*Bristol is open*” project (Bristol City Council, University of Bristol, 2017), the aim of the project is to connect many open and participating IoT sensors to an anonymised database. This includes energy, air quality, traffic flows and sensors on smart phones from willing participants, combined these can result in an open programmable city; with city information available freely.

The use of IoT devices to measure air quality within dense cities has been done before. The journal article “*Developed urban air quality monitoring system based on wireless sensor networks”* aim was to measure known polluted areas in real time, using cheaper small networked connected devices. This was achieved using WSN to connect the several sensor nodes to a gateway and GSM to communicate with a database.

(Ana & Octavian, 2014)

## Open Source

*Open source* refers to a piece of computer software along with the source code made publicly available with a license where the copyright holder provides the right to change, study and distribute the software to anyone for any purpose. *Open source* software can be developed by a single developer or be a collaborative project, which offer a range of benefits, most notably by developing a diverse scope of design perspective that a single company is incapable of sustaining. Most commonly known *open source* software is the Linux kernel, Firefox, PHP, MySQL, Gimp and Git.



Illustration 2 Open Source Initiative Logo From: <https://opensource.org/logo-usage-guidelines> [Accessed 2 March 2017]

*Open source* software is commonly released under a license which contributors are under when submitting changes, licenses vary depending on the project. Explicit licenses such as the Apache contributor license, which define that intellectual property has been contributed to the project or company. Implicit licenses such as open source licenses allow the software to be freely used modified and shared. Common open source licenses are, MIT license, BSD license and Mozilla Public license.

(Laurent, 2004)

Open source software is easier obtained then propriety software, this availability increases the adoption of an open source standard. Due to open source projects having unlimited developer contribution, software can be reliable, high quality and often produced quickly. This is due to the amount of independent developers testing and fixing bugs. The mix of developer input produce a flexible system allowing easier contribution, while also speeding up innovation.

Releasing open source software was initially a way to share software allowing to learn from one another, with the goal of advancing computing. Commercialising software began to take over which affected the sharing of software, eventually several groups emerged calling for free and open source software. Today the “*Open Source Initiative”* is widely recognised by governments internationally.

(Creative Commons Attribution 4.0 International License, 2017)

## Operating System

The operating system (OS) is the system software that manages the computer’s hardware and provide resources for the application software. While computers don’t necessarily require an OS to run, such as bare-metal system, they are useful to manage the resources and user’s application. This project will be considering two types of OS, firstly a real-time OS (RTOS) and Linux which is a general-purpose operating system (GPOS).

This project does not need to meet any real-time constraints, therefor the research will look at embedded boards using either a general-purpose operating system or a real time operating system.

(Sirio, 2017)

## Embedded Systems

“[Embedded systems] are electronic systems that execute a limited number of fixed tasks. Because the tasks do not change during the lifespan of an embedded system, it is not general programmable in the way a personal computer or workstation is.”

(Stravers, 1994)

Embedded systems differ to general purpose machines which have no specific task and instead can change its tasks based on the software. Embedded systems must often meet real-time constraints; while it’s not always required to be real-time, they are often required to meet some real-time constraint.

They are most often embedded inside larger systems for specific tasks, these commonly require real-time constraints, dependability and meet efficiency requirements. Within each of the following is multiple embedded systems such as inside car brakes and trains acceleration.

* Planes
* Cars
* Trains
* Traffic lights
* MRI
* Telecommunications

1. (Peter, 2011)

## Embedded Systems options

The project will need an embedded system to drive the components, below is a list that could potentially be used for this project. The list is composed of embedded systems that are relatively cheap, have a reasonable amount of documentation, require low power, wide range of connections and have a large community to aid in debugging.

### Raspberry Pi 3 – B

The Raspberry Pi 3 by the Raspberry Pi Foundation is the most popular hobby embedded board on the list, it uses its own Linux based general purpose OS. This board comes with a wide range of peripheral connections which are unlikely to be used, however is very powerful for price and size.

1. (Raspberry Pi Foundation, 2016)



Illustration 3 Raspberry Pi 3-B, Raspberry Pi, Available From: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> [ Accessed 20 February 2017]

|  |  |  |  |
| --- | --- | --- | --- |
| Specifications | Size | Misc | Connections |
| * 1.2GHz 64-bit quad-core Cortex A53 * 1GB RAM * General Purpose Operating system (GPOS) * Wireless LAN * Bluetooth | 85.60 mm × 56.5 mm | * 5V Power usage * Micro SD card slot | * 40 Total pins * 2 – 5V * 2 – 3.3V * 8 – GND * 2 – I2C EEPROM * 26 Digital pins |
| Strengths | | Weaknesses | |
| * Powerful * Built in Wi-Fi * SD card storage * Low cost | | * CPU to powerful * High power consumption * Large | |

Table 1 Raspberry Pi 3-B Specs

### Particle Electron

The *Particle Electron* by Particle Industries Inc is an embedded board with an embedded cellular modem, this allows communication over the particle gateway using GSM by default. This is the biggest benefit to the Electron; however the board cost is high and the need for a monthly sim card only increases that price.

1. (Particle Industries Inc, 2015)

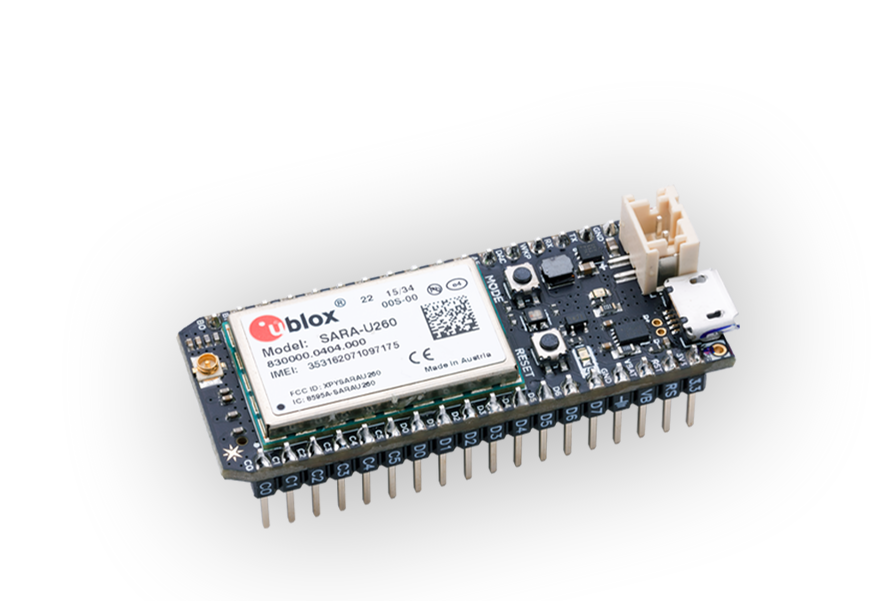


Illustration 4 Particle Electron, Particle, Available from: <https://store.particle.io/> [Accessed 20 February 2017]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specifications | Size | Misc | | Connections |
| * STM32F205 ARM Cortex M3 (120MHz) * 1MB Flash, 128K RAM * Cellular modem: U-Blox SARA G-series (2G) * Open source design * Real-time operation system (RTOS) | 50.8mm x 20mm | * 3.9V Power usage | | * 36 Total pins: * 28 GPIOs (D0-D13, A0-A13) * TX/RX * 2 GNDs * VIN * WKP * RST |
| Strengths | | | Weaknesses | |
| * Small * GPRS Ready * Low power | | | * Expensive * Restricted to web IDE / Particle IDE * Delivered from America (hard to get) | |

Table 2 Particle Electron Specs

### Redbear Duo

*Redbear Duo* developed by Redbear Limited has the most wireless communication modules with WiFi and Bluetooth. The Duo uses a forked version of the Electron’s firmware, while using the open source FreeRTOS.

1. (Redbear Limited, 2016)

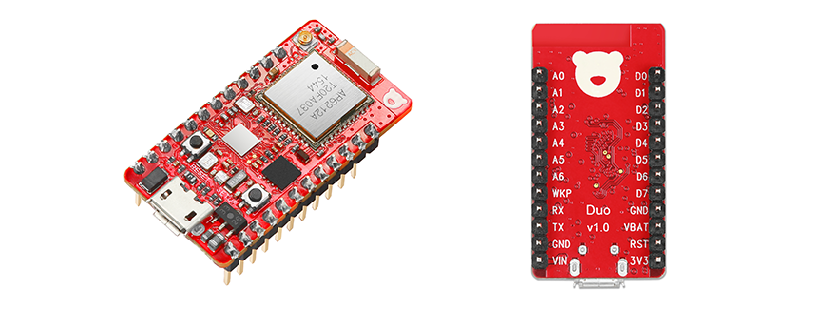


Illustration 5 Redbear Duo, RedBear, Available from: [https://github.com/redbear/Duo/blob/master/docs/duo\_introduction.md](https://github.com/redbear/Duo/blob/master/docs/duo_introduction.md%20) [Accessed 20 February 2017]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specifications | Size | Misc | | Connections |
| * STM32F205 ARM Cortex M3 (120MHz) * AP6212A Wireless Module (Wi-Fi + Bluetooth) * 1MB internal flash * 2MB external SPI flash * 128KB SRAM * Real-time operation system (RTOS) | 1. 38.9mm x 20.4mm | * 3V-3.6V * Supports multiple development options. | | * 24 Total pins * 18 x Digital I/O * 8 x ADC Input * 2 x DAC Output * 13 x PWM * 2 x UART * 2 x SPI * I2S, I2C, CAN * JTAG(SWD) debug port |
| Strengths | | | Weaknesses | |
| * Small * Cheap * Low power * Built in Wi-Fi | | | * Less powerful * Less pins for expansion | |

Table 3 Redbear Duo Specs

### MSP-EXP430F5529LP

The MSP developed by Texas Instruments is the only embedded board listed not using an Arm cortex CPU, instead it uses their own Texas instrument microcontroller; this does offer lower CPU power but not enough to be put off the list. The MSP is an entry level board, which means it offers lots of IO pins with a simple interface, along with a vast number of technical documents.

1. (Texas Instruments Incorporated, 2015)



Illustration 6 MSP-EXP430F5529LP, Texas Instruments, Available from: <http://www.ti.com/tool/MSP-EXP430F5529LP> [Accessed 20 February 2017]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specifications | Size | Misc | | Connections |
| * MSP430F5529 16-bit MCU (25-MHz) * 128KB of flash * 8KB of RAM | 1. 66mm x 64mm | * 1.8V-3.6V | | * 40 Total pins * Up to four serial interfaces (SPI, UART, I2C) * 12-bit analog-to-digital converter |
| Strengths | | | Weaknesses | |
| * Very low power * Very Cheap * Light weight | | | * Large * Slow CPU * SDK not Development friendly * Low support | |

Table 4 MSP-EXP430F5529LP Specs

## Temperature and Humidity sensors

The project requires the monitoring of temperature and humidity in the local vicinity for several key reasons. Firstly, the temperature and humidity will have direct effect on the data received from gasses and particles sensors, these sensors can only operate accurately within specific temperature and humidity conditions. Secondly, monitoring the effect of particles and air quality on humidity and temperature. Humidity monitoring will return a percentage value which indicts the level of water vapour present in the air. The temperature sensor will return the current temperature in Celsius. The two sensors readings can also calculate the dew point.

The key features:

* Accurate results with longevity in an outside environment
* Digital signal
* Small for mobility
* Cost effective
* Resistance to extreme city conditions

### Temperature and Humidity sensor comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor name | Price | Specifications | Pros | Cons | Source |
| SHT75 | £25 | * 2.4V-5V * Digital signal * sleep mode * Manufacture calibrated * -40°C - +123°C * ±1.8% Relative Humidity | * Low power * Already calibrated | * Very Expensive * Very small | (Farnell, 2015) |
| DHT22 | £11 | * 3V-5V * Digital signal * Reads every 2s * -40°C - +80°C * ±2% Relative Humidity | * Cheap * Low power * Small * Lots of support | * Limited to reading every 2s. | (Sparkfun, 2010) |
| HIH-6120-021-001 | £7 | * 2.3V-5.5V * Digital signal * -25°C - +85°C * ±4% Relative Humidity | * Low powerful * Cheap | * Very small * Smaller range | (RS Components Ltd, 2011) |

Table 5 Temperature & Humidity sensor comparison

## Particulate Matter Sensor

Particulate matter sensors are used to detect a wide range of particles that are commonly grouped by size, these range from 10 micrometres in diameter (PM10) to 2.5 micrometres (PM2.5). All particles 10 micrometres or smaller are very easy to inhale and it is these which cause multiple health problems.

Particulate Matter sensors are very expensive instruments while still not offering 100% accuracy. There are small and cheap alternatives, however these are far less accurate and become a balance of cost and accuracy. These cheaper alternatives also make it impossible to correctly differentiate between particles.

The primary focus in particulate matter sensor is the detection range, as mentioned, the ideal particle matter (PM) sizes to detect range from 10 micrometres(PM10) down to 2.5 micrometres (PM2.5).

1. ( Anderson, et al., 2012)

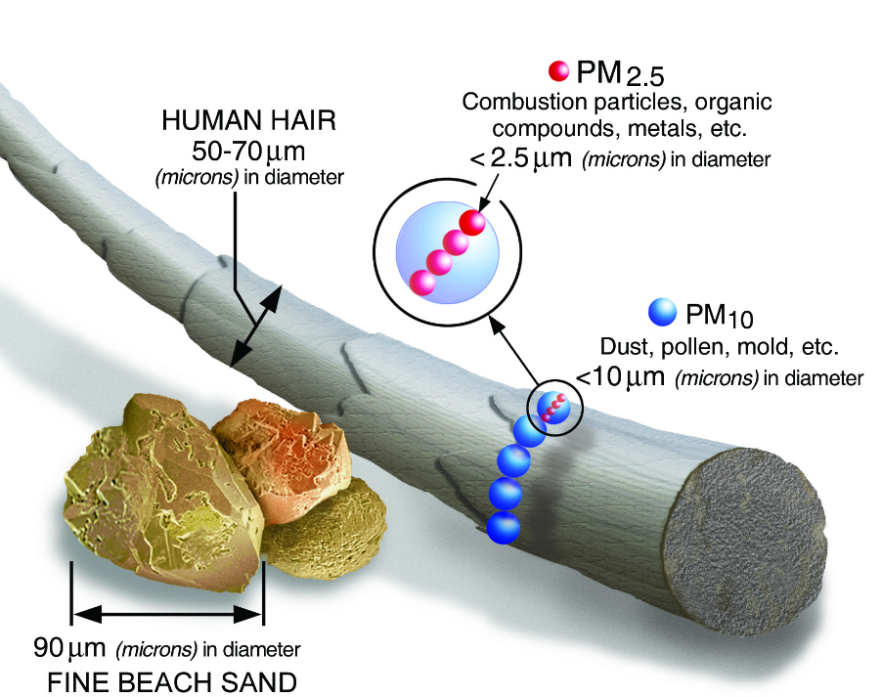


Figure 3 Size comparisons for PM particles, EPA From: [https://www.epa.gov/pm-pollution/particulate-matter-pm-basics#effects](https://www.epa.gov/pm-pollution/particulate-matter-pm-basics%23effects%20) [Accessed 14 January 2017].

# Particle matter (PM10)

Particles within the 10 micrometres in diameter range can be sourced from crushing or grinding operations, and dust stirred up by vehicles.

# Particle matter (PM2.5)

These are the smaller pollutants at 2.5 micrometres and smaller, are often a result from combustion, such as cars and power plants, forest fires. PM2.5 is the finer particles which are often more toxic heavy metals and hazardous organic pollutants.

1. (Libelium Comunicaciones Distribuidas, 2015)

### Particulate Matter sensor comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor name | Price | Specifications | Pros | Cons | Source |
| PPD42 Particle sensor | £13 | * 5V * Detect 1um+ * Digital pulse output | * Accurate * Well supported * Well suited for city usage | * Expensive * Large | (Shinyei Technology Co.,LTD, 2016) |
| Optical Dust Sensor GP2Y1010AU0F | £8 | * 3.3V * Fine particle detector * Analog output | * Cheap * Open source * Low power | * Unknown detection range * large | (SparkFun, 2014) |
| Grove - Dust Sensor | £15 | * 5V * Detect 1um+ * PWM output | * Accurate * Suited for city usage | * Expensive * Large | ( Seeed Development Limited, 2015) |

Table 6 Particulate Matter sensor comparison

## Gas Sensor

Gas sensors will be required in this project to get obtain an accurate reading of the air quality; however, air quality is usually effected by a combination of gasses. While gas sensors can detect a range of gasses which exist, it becomes difficult to accurately determine the concentration of these individual gasses.

This leads to focusing on a small set of gas sensors or using gas sensors which detect the larger range of gasses, with the aim of using the datasheet to find the concentrations.



Illustration 7 MQ7 Gas sensor, From: [https://www.sparkfun.com/products/9405](https://www.sparkfun.com/products/9405%20) [Accessed 14 January 2017].

The key gasses to consider are:

* Carbon Monoxide (CO)
  + An odourless gas produced by the incompletion of burning fuels, primary source is from car emissions. CO inhibits red blood cells ability to carry oxygen, this can cause haemoglobin.
* Carbon Dioxide (CO2)
  + CO2 is beneficial for life as we know it, however the abundance being created due to the burning of fossil fuels. Like CO the effects of CO2 can affect the blood flow while also causing lung damage in large quantities.
* Nitrogen Oxides (NOx)
  + NOx gasses are also created from car emissions, when the N2 gas combines with oxygen. In large quantities NOx can have negative effects on the lungs functionality.
    - Note: NOx sensors are incredibly expensive so unlikely to monitor.

(Kampa & Castanas, 2008)

### Gas sensor comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor name | Price | Specifications | Pros | Cons | Source |
| MQ-7 - Carbon Monoxide (CO) | £5 | * 5V * Analogue Signal * 20 - 2000ppm | * Accurate * Cheap | * Only detects CO | (Ardiuno, 2016) |
| MQ135 | £3 | * 5V * Digital + Analogue output * Detects, NH3, NOx, alcohol, benzene, smoke, CO2, etc * 10-1000ppm * 10-300ppm | * Digital output * Cheap * Detects many PM | * No specific PM detection * Too sensitive for outside | (Ardiuno, 2016) |
| MiCS-2614 – Ozone (O3) | £28 | * 2.5V * 10 – 1000ppm * Digital output | * Very Accurate | * Very expensive * Complicated | (MicroController Pros LLCc, 2017) |
| MG811 CO2 Carbon Dioxide Gas Sensor | £50 | * 6V * Analogue signal * 0 - 10000ppm | * Very accurate | * Expensive * Power hungry | (Hanwei Electronics, n.d.) |

Table 7 Gas sensor comparison

## GPS

One of the projects aim is portability, to accurately determine the location of each data reading, GPS can achieve this by finding the current coordinates of the device. Global positioning system (GPS) will be used to match data readings to specific locations. GPS is used for finding the positioning of itself, it does this by using multiple satellites around the earth. The GPS acts like a receiver picking up the positioning signals and time stamps from the satellites, and uses this information to triangulate its own position relative to the satellites. This can be seen below in Figure 4.

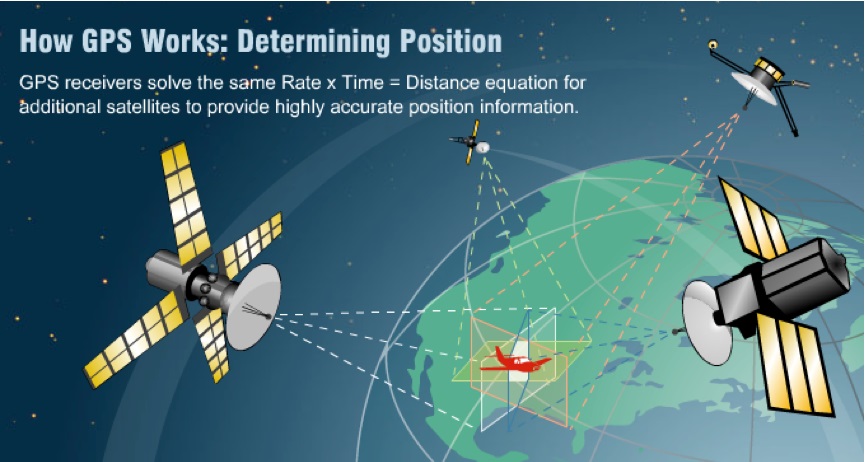


Figure 4 GPS Triangulation From: [http://www.avionicswest.com/Articles/howGPSworks.html](http://www.avionicswest.com/Articles/howGPSworks.html%20) [Accessed: 15 January 2017]

Using the data record from one device over a large area, or a network of said devices; the GPS coordinates can be used to build up a city-wide image of readings. Below is an image from the MAQOUM project (Institute for Software Integrated Systems, 2011).

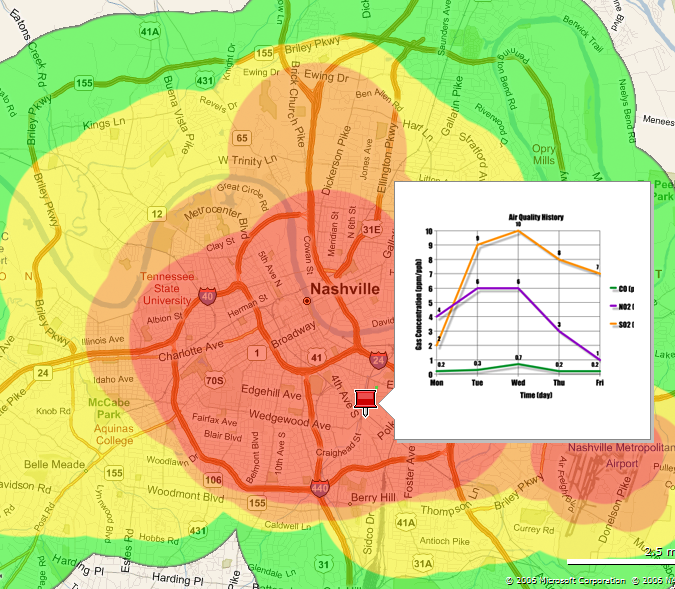


Figure 5 MAQUMON pollution map From: [http://www.isis.vanderbilt.edu/projects/maqumon](http://www.isis.vanderbilt.edu/projects/maqumon%20) [Accessed 16 January 2017]

1. The key features:

* Accurate results
* Small for mobility
* Cost effective
* Low power
* UART or L2C connection (preferably both)

### GPS comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor name | Price | Specifications | Pros | Cons | Source |
| PAM-7Q GPS Module | £40 | * 3.3V – 5V * 2 Meter position accuracy * UART connection * L2C connection | * Low power * UART or L2C connection * Small | * Expensive | (Parallax Inc, 2017) |
| GPS EM – 506 | £32 | * 5V * 2.5 Meter position accuracy * UART Connection | * Cheaper * Small | * Requires 5V * Being UART having both GPS and GPRS | (SparkFun Electronics, 2017) |
| DFRobot GPS | £16 | * 5V (supports 3.5v) * 2.5 Meter position accuracy * UART Connection | * Cheap * Small * Low power | * Can’t use this GPS and have GPRS | (DFRobot , 2017) |
| Adafruit FONA 808 GSM & GPS | £50 | * 5V * Send & receive GPRS data * Send & receive GSM data * Read GPS location * 2.5 Meter position accuracy * UART Connection | * Small * Does both GPS and GSM * UART Connection, no risk of no GPS | * Requires 5V * Expensive | (Makersify, 2017) |
| GPS - GP-20U7 | £17 | * 3.3V * 2.5 Meter position accuracy * UART TX only | * No setup, outputs GPS coordinates * Cheap * Low power | * No external antenna connection | (HobbyTronics Ltd, 2017) |

Table 8 GPS comparison

## Networking comparison

The project aims for portability; to achieve this some form of wireless communication will be required; allowing the data recorded to be uploaded onto a database. With effective networking the project can upload it’s results automatically when available; allowing for the possibility of live readings. The research will cover local areas networks and wide area networks, briefly describing their usage within the project. The options when it comes to wireless communication are as follows:

* Wi-Fi
* LoRaWAN
* Bluetooth
* GSM

### Wi-Fi

Wi-Fi uses the standard IEEE 802.11 to communicate over a Wireless local area network on a radio frequency(IEEE, 2012). Wi-Fi is trademarked by the Wi-Fi Alliance which consists of over 300 companies. Accessing a Wi-Fi network means the device needs to be equipped with a wireless network interface controller, and within range of a wireless network; which are often connected to the internet. Wi-Fi using 2.4 GHz can offer up to 54 Mpbit/s for devices such as, computers, phones, laptops, digital audio players and IoT devices.

Wi-Fi is commonly used within households, businesses and public hotspots with a range of 20 meters up to 100m, however using a semi-parabolic antenna the range can exceed 20 miles; it should be noted these large Wi-Fi antennas are not implemented within Bristol.

Wi-Fi offers high data throughput, however require access to a wireless network which are often protected by wireless protected access (WPA) protocols. A portable device may not be able to a connect to a network freely when desired, limiting its usage.

Opting for Wi-Fi means three possible use cases. First is depending on public Wi-Fi hotspots, this option can offer frequent internet connection. The issue is there is no guarantee of finding a public network, with many still requiring entering a unique password. Second is purchasing a hotspot pass from an internet service provider, this allows connecting to any of their networks using a specific login. While viable this option can be extremely expensive. Final option is storing multiple known networks and connecting when within range. This means the device is likely to upload results less frequently, however does offer a cheaper and more secure alterative.

1. (Geier, 2001) (Wi-Fi Alliance, 2017)

### LoRaWAN (Low power wide area network)

LoRaWAN (Low Power Wide Area Network) is an emerging new wireless communication managed by the LoRa Alliance, offering low power, low cost, high range and secure bi-directional communication. LoRaWAN is designed to support a large network of millions of devices, such as IoT devices, machine to machine and smart cities, offering a range of 15 km to 20km, depending on multiple variables such as hardware and area density; urban areas get around 5km range. LoRaWAN is an Low Power Wide Area Network (LPWAN) specification.

“The LoRaWAN Specification document describes the LoRaWAN™ network protocol including MAC layer commands, frame content, security, flexible network frequency management, device EIRP and TX dwell time, power control, relay protection and more.” (LoRa Alliance, 2017)

LoRaWAN network architecture usually uses star-of-stars topology, using gateways as a bridge to relay messages across end devices and a central network server; using standard IP connections. The data transfer rate is selected based on communication range and duration, data rates range from 0.3 kbps to 50 kbps*.* (LoRa Alliance, 2017)

LoRaWAN would be the ideal choice, however due to the current lack of gateways obtain a connection is tricky, due to this issue it’s the most unreliable option.

### Bluetooth low energy

Bluetooth low energy (BLE) is an advancement on classic Bluetooth, both are short range wireless personal area networks. BLE is designed to offer low energy consumption while offering similar communication range as its predecessor, developed to work perfectly on smaller applications such as: headsets, keyboards, wireless speakers and IoT devices. BLE offers a theoretical range of up to 100 metres, however practically the range will vary from 1m to 20m. Data throughput varies from the specific BLE module, ranging from 125 kbit/s to 2 Mbit/s.

BLE is a local area network, requiring connection to a gateway to upload the results. This issue is easily solved by connecting the device to the user’s phone, allowing for a consistent connection. The networking costs would be shifted onto the user’s phone, this option would also require the development of a mobile phone application.

1. (Bluetooth SIG, Inc, 2017) (STMicroelectronics, 2017)

### GPRS

General packet radio system (GPRS) is a packet oriented data transmission system, its used for 2G, 3G and 4G cellular communication; most commonly seen as part of the mobile phone networks. GPRS is an integrated component of the Global System for Mobile Communications (GSM) which extends its data capabilities, such as offering Short Message Service (SMS) and constant internet access however, GPRS does not have specific data transfer speeds, it has variable throughput and latency based on the number of users currently using the service.

The use of GPRS requires specific hardware, a device using the GPRS network must contain a GPRS module used to connect to the network, and a sim card to store a unique serial number and other important information; these are used to address the device. Accessing the GPRS and GSM network will need to be within reach of a base station antenna, this limits the reach of wireless communication; the range can vary from couple hundred meters to several tens of kilometres. The GSM network has developed quickly due to the spread of mobile phones, a connection can be made in any urban areas.

Below in Figure 6 is the GSM coverage within and around the city of Bristol, which clearly shows the signal that can be obtained using GSM within this project.

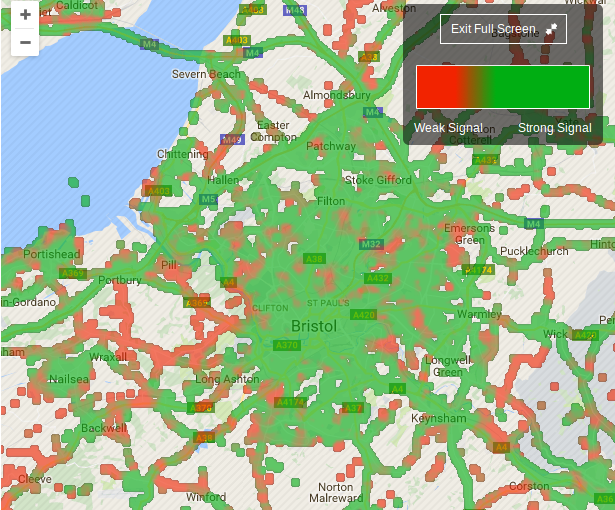


Figure 6 Bristol GSM coverage From: [http://www.which.co.uk/reviews/mobile-phone-providers/article/mobile-phone-coverage-map](http://www.which.co.uk/reviews/mobile-phone-providers/article/mobile-phone-coverage-map%20) [Accessed 16 March 2017]

While GSM offers a consistent connection to the internet, it requires a SIM card and a networking contract with a phone network such as EE. Accessing the GPRS network also requires specific hardware, while many Embedded devices contain Wi-Fi and BLE on board.

1. (Salkintzis, et al., 2002)

## Power source

Due to the projects aim of portability having a suitable power source is crucial. Ideally the device should be cable of being powered for an entire day, offering continuous monitoring without need of recharging or replacing the batteries. The only possible solutions are batteries and self-generating power source, such as solar power. The project is aiming for low power consumption, however particulate sensor and gas sensor require high voltage for accurate results.

### Alkaline batteries

Alkaline batteries are usually a primary cell battery, designed with a single cell to be used once and discarded once depleted. Rechargeable Alkaline batteries are second cell batteries which are capable of recharging the cells offering reusability. Alkaline batteries derive the name from its use in alkaline electrolyte, powering the device by the reaction between zinc and manganese dioxide.

There are Five types alkaline batteries with varying Voltage and milliampere an hour (mAh):

* D – 1.5 Volts & 15,000 mAh
* C – 1.5 Volts & 8350 mAh
* AA – 1.5 Volts & 2900 mAh
* AAA – 1.5 Volts & 1000 mAh
* 9V – 9 Volts & 595 mAh

1. (Batteries.com, 2017)

The alkaline batteries considered for this project are the 9V batteries and AA batteries. This project will require a minimum of 5 volts which the 9V battery offers, a combination of AA batteries in a pack can power up to 9v.

### Lithium-ion Polymer battery (Li-Po)

Lithium-ion Polymer batteries, also known as Li-Po and Li-Poly are second cell rechargeable batteries which are often housed within non-rigid pouches. These batteries consist of positive electrode, negative electrode, separator and electrolyte, by using the intercalation of the lithium ions of the negative and positive electrode it produces a charge in the liquid electrolyte.

Li-Po batteries aren’t produced to specific standards but are made based on need, each cell within a Li-Po battery offers 3.7v and a range of mAh; cells can be combined to offer higher voltage, for example:

* 1 Cell - 3.7 V
* 2 Cells - 7.4 V
* 3 Cells - 11.1 V

Li-Po batteries are a rechargeable battery option for the project, using a two cell battery the device can receive enough voltage with only the mAh having to be taken into consideration.

1. (Redline Batteries, 2007)

### Power source comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Price | Volts | Capacity | Pros | Cons | Source |
| 6x AA alkaline batteries pack | £3 + £4 | 7.2V-9V | 10,800mAh – 15,600mAh | * Very cheap * Easy * long runtime (mAh) | * Short term option * charging creates downtime * requires regulator | 1. (Makersify LTD, 2017) |
| 9V alkaline batteries | £6.7 | 9v | 595mAh | * Very cheap * Easy | * Short term option * requires regulator * Low runtime (mAh) | 1. (Amazon.com, Inc., 2017) |
| Li-Po battery | £20 | 7.4V | 2,200mAh | * Easy * Lots of documentation | * Charging circuit required * Consistent charging required * Low runtime (mAh) | 1. (Makersify LTD, 2017) |
| Li-Po power bank | £14 | 5v | 15,600mAh | * Easy * Long run time (mAh) | * Requires breakout board * Consistent charging required | 1. (Amazon.com, Inc, 2017) |

Table 9 Power Source option comparison

## Database and Web server

The data monitored from the device will need to be collected into an organised storage system, along with offering a simple interface to view and query the data gathered.

Due to the project using wireless communication to transfer the data, an online database is the most suitable solution for storage. A relational database would be required to link data recorded to a device, while not necessary for a single device it offers expansion. A website will be used to easily interface with the database, which require a webserver to process HTTP requests and basic network protocol.

It’s common practice to have these two components separate for security and expansion, however using one machine to handle both functions are possible and while be considered side by side.

### MySQL introduction

MySQL is an open source relational database system (RDS), which uses SQL (Structured Query Language) a programming language designed specifically for managing data held within a RDS. MySQL is extremely common in personal and business practices, due to the reliable and effective database system that’s open source, which can run on many systems; such as Linux, FreeBSD, Unixware and Microsoft Windows. MySQL is a central component of the open source LAMP stack, LAMP stands for, Linux, Apache, MySQL and PHP. LAMP is designed for building suitable dynamic websites and other web applications.

1. (Suehring, 2002)



Illustration 8 MySQL Logo From: [https://www.mysql.com/](https://www.mysql.com/%20) [Accessed 24 February 2017]

This project will be using MySQL for an online database management, making use of the basic functionalities such as, insertion, queries, data updates and deletion. The reasoning for using MySQL is the large community, most support and many services using MySQL as a standard database system.

1. (Oracle Corporation, 2017)

### Database and Web server comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Database option | Price | Specifications | Pros | Cons | Source |
| Raspberry pi – Home server | £25 | * 1.2GHz 64-bit quad-core CPU * 1GB RAM * 64GB storage | * Cheap * Complete control * Scalable | * Using home networked * Higher security threat * Personally, maintaining hardware | 1. (Timerlake, 2017) |
| Go daddy – web hosting | £2.99 per month | * 1 Website * 100GB storage * 10GB MySQL storage * unlimited bandwidth | * Cheap * Maintained hardware * 100% uptime * Expandable | * Low storage space * No access to hardware | 1. (GoDaddy Operating Company, LLC, 2017) |
| Amazon RDS – Free tier | £0 (first year) | * 20GB storage * 20GB backup storage * 750 hours of RDS instance | * Free first year * good starting storage * 100% uptime | * Expensive after first year * Complicated * No access to hardware | 1. (Amazon Web Services, Inc, 2017) |

Table 10 Database comparison

# Requirements

## Introduction

This section will be covering the project's requirements; these will be created based on the research section. The requirements will be split into hardware requirements with a further break down into system requirements. The purpose is to give an overview description of the project along with its capabilities, which will act as a guide during the design and implementation stages of the system.

## General overview

The aim of the device is a small lightweight Pollution and air quality sensor with the ability to also track temperature and humidity. The purpose is to aid in tracking and measuring pollution and air quality within dense urban areas. This will be achieved by tracking temperature, humidity, particulates and gasses in the vicinity. Along with reading and storing this data, it will track the device via GPS, this allows accurate monitoring of the area data was recorded. This makes it possible to track pollution levels over time; as well as across different locations in a city, with the idea of expansion using multiple devices passing through. Combining the results can result in a picture of city wide, maybe even countrywide; pollution and air quality.

1. Hardware requirements

The research has highlighted some key aspects and features to the design and hardware that will be required. The following is a list of the hardware along with its requirements based from the research section. Each piece of hardware will be picked from the research section to match all or most of these requirements.

### Embedded Board hardware

The embedded board is the core of this project; primarily it will need to have the required pins for all the sensors that will be attached. A board with several wireless communication methods will allow for more flexibility.

|  |  |
| --- | --- |
| **Hardware** | **Reason** |
| WiFi | For possible wireless communication to an online database. |
| Bluetooth | For possible wireless communication to an online database. |
| Serial port | Used for embedded system development and debugging. Most likely in the form of USB. |
| Low power < 5v | Device aims to be portable, so less power consumption means longer usage. |
| Digital pins | Pins used to connect to the sensors. |
| TX/RX pins | These pins are required for GPS communication. |

### Temperature and Humidity sensor

These two sensors usually can come in a single form factor, however both readings will need to be accurate. The readings from these will be record and will also have a direct effect on the operations of the particles and gas sensors.

|  |  |
| --- | --- |
| **Hardware** | **Reason** |
| Accurate Temperature (±1°C) | Accurate readings result it accurate results. |
| Accurate Humidity (±2%) | Correct readings will help data monitoring, especially since it effects particle sensor. |
| Wide Temperature range (-10°C - +50°C) | This allows it to work reliable in almost any location. |
| Digital Output | Simple digital outputs make reading values easier without ADC. |
| Low power < 3.3v | Device aims to be portable so less power consumption means longer usage. |

### Particulate Matter sensor

The particulate matter sensor will be monitoring the particles in the vicinity, alongside the gas sensor, this is the second most important piece of hardware. This component partially responsible for monitoring air quality therefor the will need to be accurate and suitable for outside usage.

|  |  |
| --- | --- |
| **Hardware** | **Reason** |
| Accurate particle detection 1.0µm <= | The more accurate the particle detection is the more reliable the results are. |
| Detect multiple particles | This allows one sensor to cover a wide range rather than have multiple sensors which increase size, cost and power usage. |
| Suitable for outdoor usage | This project aim is to be carried around urban areas. While many sensors are more accurate they are too sensitive for outdoor usage. |
| Digital Output | Simple digital outputs make reading values easier without ADC. |

### Gas sensor

This project is likely to require multiple gas sensors to cover a wider range of gasses, preferably having one for a specific gas such as CO; and another which monitors a wider range of gasses. These gas sensors require accuracy and monitor a large particle per million (ppm).

|  |  |
| --- | --- |
| **Hardware** | **Reason** |
| Accurate gas monitoring range  20ppm-1000ppm | These sensors need to be able to monitor the gas in small and large concentrations, higher ppm roof, the higher concentrations it can monitor. |
| Detect multiple particles | This allows the project to cover a wide range of gasses, allowing for better mapping of air quality. |
| Suitable for outdoor usage | This project aim is to be carried around urban areas. While many sensors are more accurate they are too sensitive for outdoor usage. |

### GPS

While the GPS coordinates do not need to be pinpoint accurate, a location is needed to place the readings on a map; allowing pollution mapping. GPS is also required as the source of accurate time stamps, as the device will be constantly sleeping with no other way to track the time.

|  |  |
| --- | --- |
| **Hardware** | **Reason** |
| Accurate ( < 5m) | Accurate readings allow a better air quality map to be created. |
| Digital Output | Simple digital outputs make reading values easier without ADC. |
| Low power < 3.3v | Device aims to be portable so less power consumption means longer usage. |

## Requirements

Below in Table 11**Error! Reference source not found.** is the list of requirements this project will need to accomplish the problem of portable, cheap, urban air quality monitoring. The requirements list covers areas from the hardware, software mechanics to the database; this will build up a clear picture of the projects capabilities and aid in the design and implementation of the project.

## Requirements Priority

The requirements are prioritised using the standard MoSCoW prioritisation method. This method is a popular and common form of prioritisation and is defined with four acronyms:

* Must have – MH
  + Must have are critical to the current project for it to be a success.
* Should have – SH
  + Should have are important but not necessary for delivery in the project
* Could have – CH
  + Could have are desirable but not necessary to the project.
* ~~Won’t have – WH~~
  + ~~Won't have are the least-critical or not appropriate at that time.~~

To streamline the requirements, the WH acronym will be ignored. While this acronym can be very important in larger projects, it has little use here.

(Brennan, 2009)

## Requirements Table

### Post Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| PO01 | System powers on. | None | MH |
| PO02 | System has no POST CPU faults. | PO01 | MH |
| PO03 | System has no POST RAM data faults. | PO01 | MH |
| PO04 | System has no POST I/O data faults. | PO01 | MH |

### General Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| GE01 | Device will read current time and date stamps from GPS. | PO01-PO04 | SH |
| GE02 | Battery can power device for 10-12 hours. | PO01 PO04 | SH |
| GE03 | Device will take readings every 5 minutes. | PO01-PO04 | MH |
| GE04 | Device will record particle count over 30 seconds. | PO01-PO04 | SH |
| GE05 | System will record gas ppm over 30 seconds. | PO01-PO04 | SH |
| GE06 | Device will store results when not connected to the internet. | PO01-PO04 | MH |
| GE07 | Device will change to a known network connection when possible. | PO01-PO04 | MH |
| GE08 | Device will upload all results to the database when connect to the internet. | PO01-PO04 | MH |
| GE09 | Allow user to disable the device without powering it off. | PO01-PO04 | CH |
| GE10 | Device Sleeps between data readings. | PO01-PO04 | SH |
| GE11 | If Device losses GPS signal, location will be last known if recent. | PO01-PO04 | CH |
| GE12 | If no time stamp can be read from GPS, then will time assumed between one before and after. | PO01-PO04 | CH |

### Warnings and Fault Detection Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| WF01 | System will indicate a warning and status with LEDs. | PO01-PO04 | MH |
| WF02 | System will have fault detection temperature and humidity reading error. | PO01-PO04, GE03, GE04 | SH |
| WF03 | System will have fault detection particle sensor reading error. | PO01-PO04, GE01 | SH |
| WF04 | System will detect loss of GPS connection. | PO01-PO04, GE05 | SH |
| WF05 | System will have fault detection for GPS reading error. | PO01-PO04, GE05 | SH |
| WF06 | System will indicate if connected to WiFi. | PO01-PO04, GE07 | SH |
| WF07 | System will warn user of low battery. | PO01-PO04 | CH |
| WF09 | Fault detection will result in self reboot. | PO01-PO04 | CH |

### Database Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| DB01 | Store Devices within a Devices table | None | MH |
| DB02 | Store Gas, Particle, Temperature, Humidity, GPS coordinates, Time stamps, Date stamps and Device ID within Data table. | GE01-GE07 | MH |
| DB03 | Database is not limited to local network access. | None | CH |

### Website Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| WI01 | Allow users to see all readings | DB01, DB02 | MH |
| WI02 | Allow users to search specific device readings | DB01, DB02 | MH |
| WI03 | Allow users to search specific days’ readings | DB01, DB02 | MH |
| WI04 | Allow users to search for specific areas. | DB01, DB02 | CH |
| WI05 | Allow readings to be seen overlaid on a interactive map | DB01, DB02, WI01, WI02 | SH |

### Security Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Func. Req #** | **Requirement Description** | **Dependencies** | **Priority** |
| SE01 | Webserver requires key to data upload. (no unauthorised uploads) | DB01 | SH |
| SE02 | Web interface and Database have SQL injection protection. | DB01 | MH |
| SE03 | Login to access results. | WI01 | CH |

Table 11 Requirements tables

# Design

## Introduction

The design will consist of several sections. First is the overview of the entire project, displaying how the it will function and interact with each other. Second is the hardware section, explaining the components chosen and designed within the system and hardware schematic. The last design section is the database and website, explaining how the database is setup.

## Overview

The layered architecture design can be broken down and listed into five areas which are described below:

1. Results stored in an online database.
2. Website created to upload results to the database, along with an interface to view air quality and track its effect over time. This layer is split into two sections however it’s still one layer being the website.
3. The software layer is relativity basic, it consists of reading air quality over a 30 second period; then either uploading or storing the readings. While searching for WiFi to upload any stored results.
4. The embedded board will be running an RTOS, which will run the application and pull readings from the multiple sensors. This will most likely be using the board’s pre-defined RTOS, or FreeRTOS.
5. This layer consists entirely of the sensors, these sensors will gather their readings and pass them to the RTOS system so it can be picked up. None of the sensors will be receiving information back.

Diagram can be seen below in Figure 7.

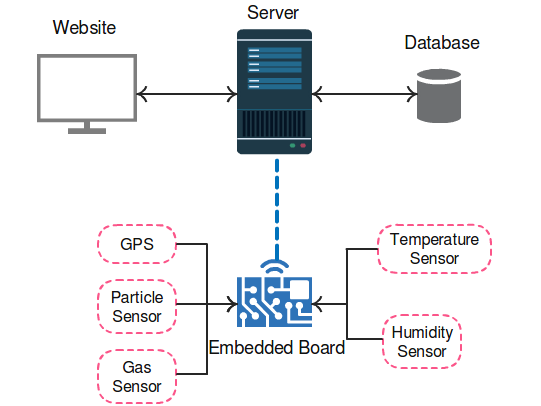


Figure 7 Layered Architecture Design

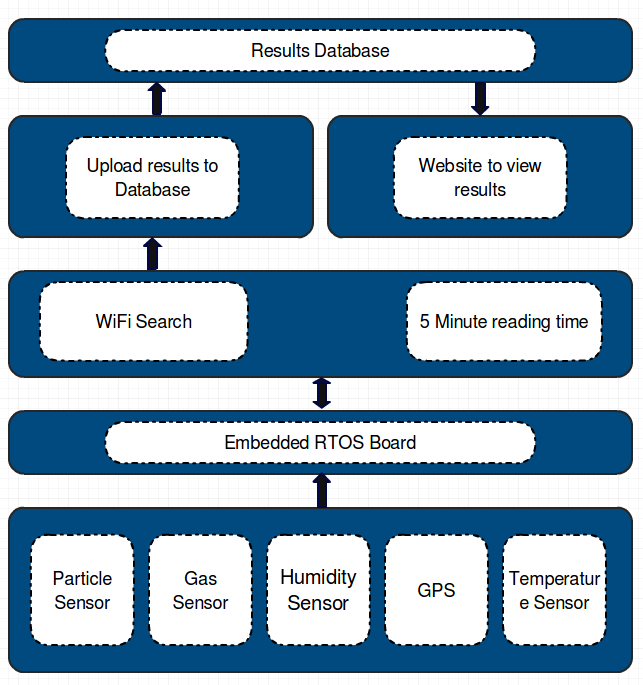


Figure 8 Layers within Architecture

## Finite State Machine

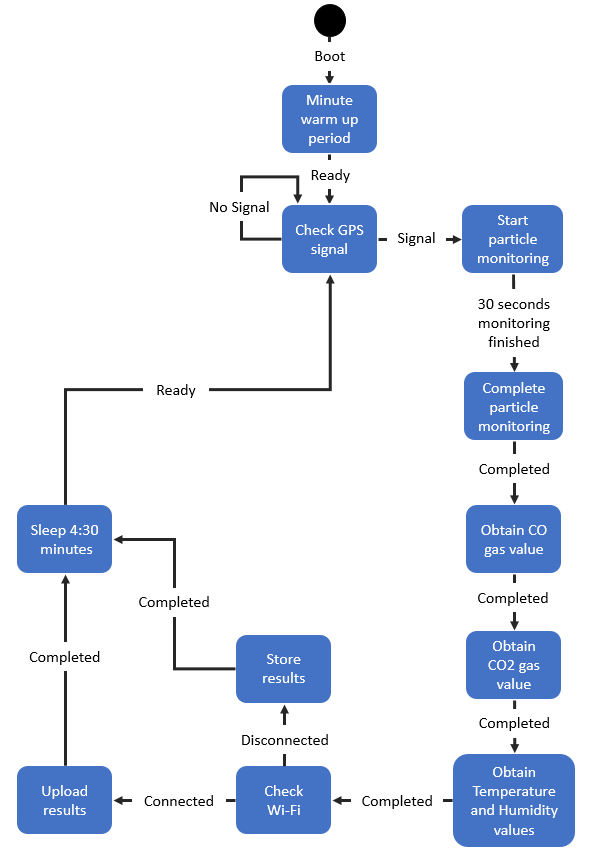


Figure 9 Finite State Machine

## Hardware Design

### Embedded Board and Operating System

This project will use a Redbear Duo as the embedded board. The Duo features an on board wireless module which supports Wi-Fi, which will be utilised for the wireless communication to the server. The low powered board has a working voltage of 3.3 volts, which is ideal for the project aims. The board can directly power several other components in this project, however components which require 5 volts must be powered directly from the battery.

The board runs a modified FreeRTOS operating system by Particle. The firmware is open source and available from the GIT repository, (Redbear Limited, 2016)*.* The firmware and user application will be cross compiled and flashed onto the Redbear. The Particle firmware uses its own hardware abstraction layer for interfacing with the pins, the API also offers functionality to the Particle cloud.

### Temperature and Humidity design

Temperature and Humidity will be monitored using the DHT22 sensor. This component runs at 0.5 Hz which limits polling to two seconds, this should not hinder the project considering readings are taken every 5 minutes. The DHT22 outputs all data through a single digital pin, which requires specific timing to acquire the correct readings and normalise the results.

Below in Figure 7 is the design for reading the temperature and humidity from the single digital input. While there are libraries for this particularly components, many of these libraries are designed for the Arduino which can’t function on the Redbear Duo.

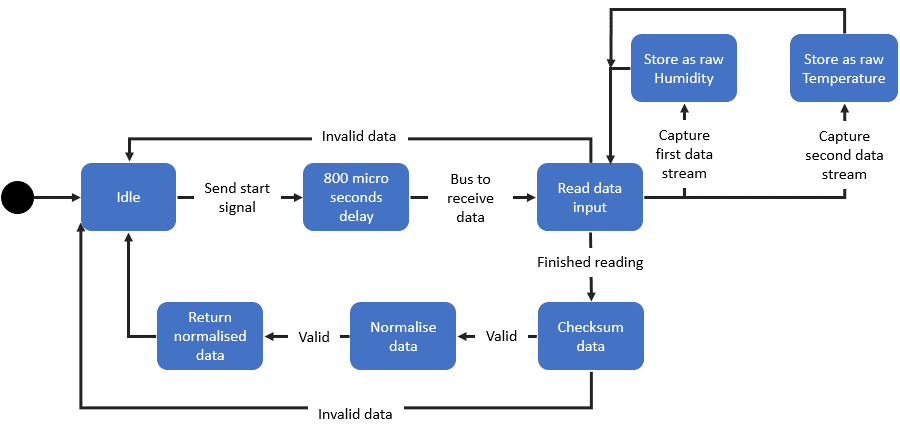


Figure 10 DHT22 Temperature and Humidity Design

### Particulate sensor design

The PPD42NJ will be used to monitor the airborne particles, this particulate sensor is a high voltage component requiring 5 volts. This sensor detects all airborne particulates which are larger than 1 micrometre, refer to the Research section for specifics on particles.

The monitoring is achieved by detecting particles based on light scattering, the single pin outputs a pulse based on this light scattering method. Monitoring the pulse is done by a Lo Pulse Occupancy time (LPO). Measuring the low pulse duration over a set specific time frame is in proportion to PM concentration.

Calculating the PPM or particles per 0.01 cubic feet requires some complicated mathematics. The grove dust sensor specification ( Seeed Development Limited, 2015) contains a snippet of code in which calculates the particles per 0.01 cubic feet.

The calculation using the spec sheet can be simplified to the following equations:

ratio = lowpulseoccupancy ÷ (sample Time \* 10.0)

concentration = 1.1 \* ratio3 - 3.8 \* ratio2 + 520 \* ratio + 0.62

### Gas sensor design

This project will use two gas sensors to measure gasses, the MQ7 will monitor carbon monoxide (CO) and the MQ135 will monitor carbon dioxide (CO2). Like the particulate sensor these sensors use a working voltage of 5 volts, being powered directly from the battery is necessary. These sensors detect gasses by using a heated coil, which is why they require higher voltage. The coil takes time to heat up which is why the device needs a minute warm up time before readings can be taken. These heated coil is the biggest power drain on the device.

#### MQ7

The MQ7 is designed specifically to read carbon monoxide, within a detection range of 20ppm to 2000ppm. The MQ7 design doesn’t need a LPO to measure the particles per million, however it has a 30 second polling time.

#### MQ135

This project will use the MQ135 to monitor CO2, however the MQ135 is an air quality sensor designed for detecting CO, CO2, NH3, Alcohol, Benzene and other gases. The design for this gas sensor is to output a single value which is a ppm for the combination of these gasses, this is problematic since this project requires individual values. Due to this problem, the project needs to derive the CO2 ppm from this combination, this can be achieved by using a correlation from the MQ135 specification curve chart. The curve chart is below in Figure 11.

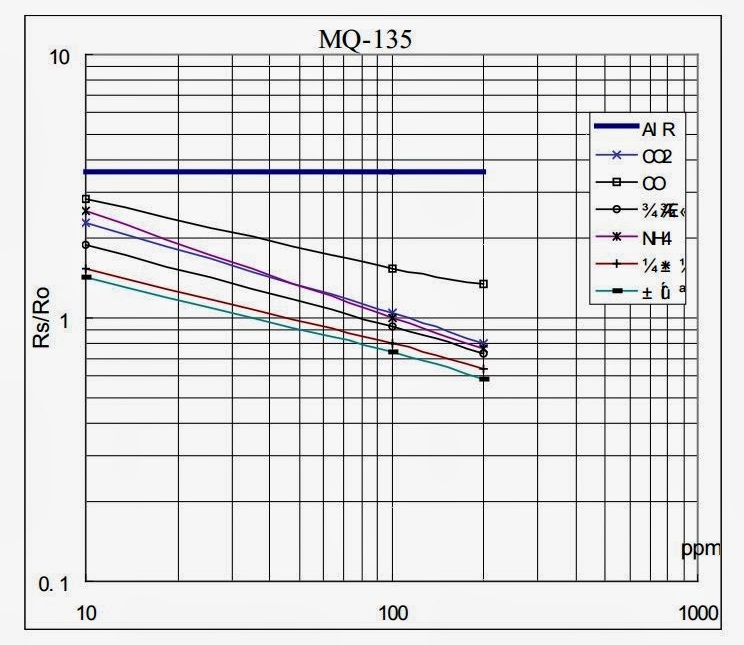


Figure 11 Sensitivity characteristics of the MQ135 From: MQ135 Specification

The curve chart is a power function, this means by using the power regression of a specific gas we can measure the PPM of CO2 within the combined value. The below piece of maths is a simplified equation to determining the PPM for specific gas, using the CO2 scaling factor.

ppm = scaling factor\*(Resistance/Ro)^Exponent

Using the MQ135 to detect CO2 is less accurate then a sensor designed to detect CO2, the MQ135 is built as an air quality sensor; the output is a collection of all these gases. This design is using a correlation to gather a fairly accurate CO2 ppm. An example is if there was zero CO2 and a vast amount of CO this would give a value of CO2 being detected, due to using the correlation.

This project is aimed to be used within densely populated areas, where its guaranteed to encounter a vast amount of CO2; which makes the MQ135 viable within this project. The maths is complicated, however using this component in this way was based off (Gironi, 2014) design using the MQ135 to detect CO2 using an ATmega. Gironi discussed and explained the mathematics throughout his personal blog, and offers open source access to the project.

### GPS Design

To match the recorded data to specific locations the device needs to know its own location and the time of reading. This will be achieved using a low powered miniature GPS called GP-20U7, this 3.3 volt GPS can be powered directly from the board. This GPS module uses a UART serial connection at 9600bps, however requires no input data; simply outputs a consistent stream of current coordinates.

The GP-20U7 outputs the coordinates in several formats called NMEA periodically. The format being picked up for this project is the “*Recommended minimum specific GNSS data*” (RMC). The RMC contains 12 values. Below is an example of RMC data, the table of values can be seen in the appendix.

$GPRMC,161229.487,A,3723.24756,N,12158.34162,W,0.13,309.62 050317,,\*10

This project does not require all the RMC values, focusing on 7 values which are:

* UTC position
  + Timestamp hhmmss.sss
* Status
  + A=data valid or V= Invalid
* Latitude
* N/S Indicator
* Longitude
* E/W Indicator
* Date
  + Date ddmmyy

### Wi-Fi Design

To achieve the projects aims, it will use the Wireless module on the Redbear Duo to upload results via Wi-Fi. However due to planned mobility, a Wi-Fi connection will not always be available. To ensure readings are uploaded the board will periodically scan for stored known Wi-Fi networks, where the data can then be uploaded.

The basis for this design relies upon the device coming within range of these known Wi-Fi networks, either at the end or throughout the device’s route. An example is the device being carried by postal workers as they deliver mail, returning to the post office at the end where it can upload the results.

This design can be achieved by scanning for Wi-Fi networks, which match any known stored networks by the SSID, every time new data has been gathered.

Figure 12 below is a finite state diagram showing the process.

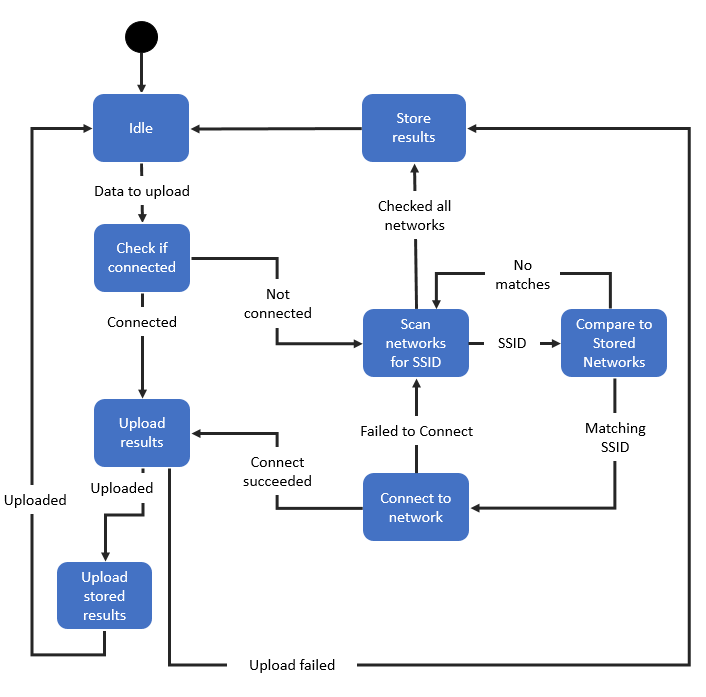


Figure 12 Wi-Fi Finite state diagram

## Battery Component

Due to the hardware components high voltage requirement, the battery being used throughout implementation will be a USB power bank; using a USB breakout board to deliver power. The aim for final product however, will be too use a two cell Li-Po battery to offer a smaller and easily rechargeable battery.

## Hardware schematic

Figure 13 is the hardware schematic for this project. The voltage difference makes it impossible to power all the components directly from the board. The solution is to power the Redbear Duo, particulate and gas sensors directly from the power supply, while powering the temperature and GPS from the board which work at 3.3 volts. The high voltage output of these components can damage the Redbear, we can level the voltage input by using a logic level convertor; which levels 5 volts down to 3.3 volts.

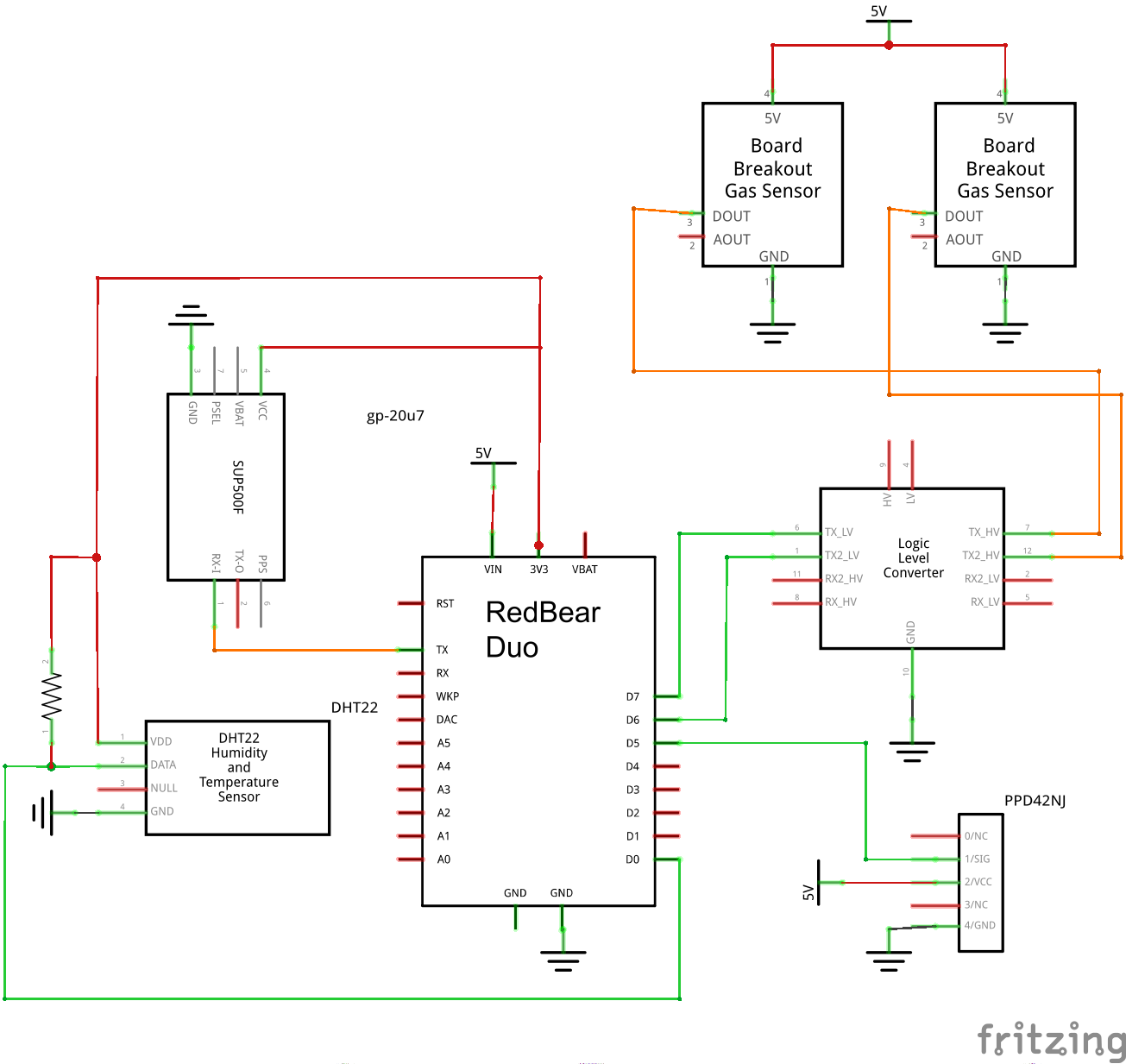


Figure 13 Hardware Schematic

## Database and Website

This project will use the service “*GoDaddy*” (GoDaddy Operating Company, LLC, 2017) for a domain name and the hosting of the website and database.

### Database

The database is designed to handle the data readings from multiple devices, it can manage this by using two tables. The first table is designed to handle the multiple devices that are uploading readings, while the second table houses the results. Using the device’s ID as a primary key we can link all readings to a specific device. This design allows searching for results to a specific device, keeping the ability to query all results to a specific location.

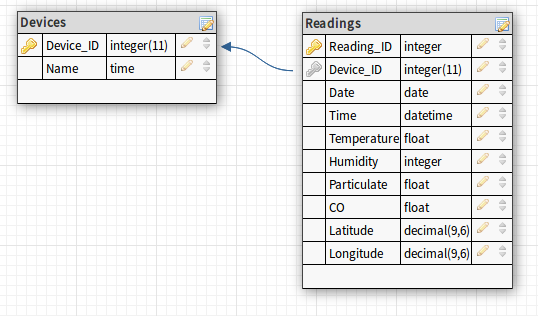


Figure 14 Database Design

### Website

The data monitored by the device will only be accessible through the website, as mentioned previously the aim is to offer a simple and effective way to query and visualise the air quality and pollution data. The ideal way to display this would be using a heat map, as seen in the “*MAQUMON*” project in the background research. A heat map can clearly show the levels and areas of pollution, however it does not give individual location data. To query specific data there will be several search options:

* By Date
* By Device
* By location

Below in Figure 15 is a rough design layout for the website using “*google maps*” (Google, 2017) to overlay and locate GPS coordinates.

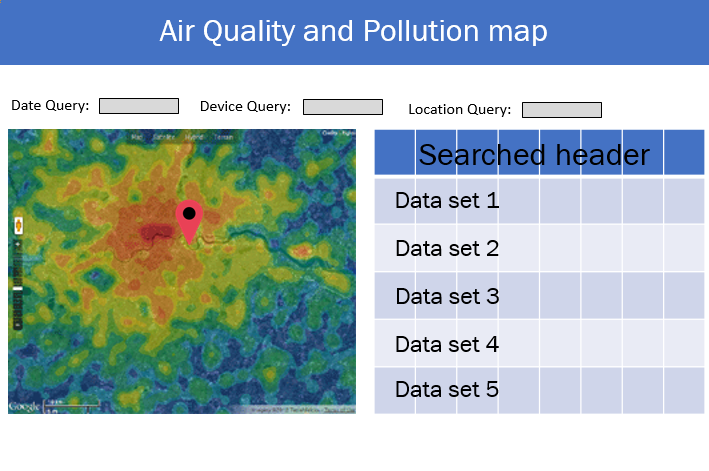


Figure 15 Website Design

# Implementation

This section will initially cover the tools used during the development and implementation, being written in order the implementation was done. The development is done as rapid prototyping. Any issues that arise during the implementation will be discussed within its section, and cover possible solutions and justification; each section is tested before moving on.

## Cross development

### Host machine

Developing applications for embedded systems often require the use of a host machine where the application will be written and then cross compiled to run the binaries on a specific embedded board. During this project the application will be written and compiled on a Linux Mint operating system, the binaries will be flashed to the Redbear Duo using “*DFU-util*”. Device Firmware Upgrade (DFU) is a tool used to flash over firmware and user application over serial.

The host machine will also be used for debugging and testing the embedded system. Using a USB serial monitor, the application can print out information of the current process, this will be used to give indication of its status; along with any warning or error messages.

A problem occurs when powering the Redbear from the battery. A micro USB is used for power and serial connection, however to power all the components a battery is needed. Powering the device using battery and the USB connection can short circuit the board. To solve this issue, a seral UART to USB will be used to monitor messages over the RX and TX pins when a battery is connected instead of the USB serial and power cable.

### Toolchain

To cross-compile the application for the Redbear Duo a toolchain is necessary. The Redbear Duo uses an ARM Cortex M3, cross compiling for this architecture means using the GNU tools for ARM processors “gcc-arm-none-eabi”.

This toolchain is open source and easily accessible, placed within my “/usr/local/” path to be accessed by the Make tool. The toolchain will not be permanently added to the hosts path to avoid any conflict, instead appended the toolchain when cross compiling.

### Version Control - GIT

Throughout the development of this project the versions and changes will be logged using GIT. GIT is an open source version control system, enabling the recovery of older versions of the application. Git is primarily used as a tool for collaboration, however in this project its purpose will be tracking changes to aid in debugging. The GIT repository for this project will contain the firmware and the user application.

## Implementation plan

The implementation will be carried out in a linear fashion which can be summarised as:

1. Cross compiling the Particle Firmware for the Redbear Duo
2. Implementing Temperature and Humidity sensor using Arduino library
3. Implementing GPS receiver
4. Implementing Particulate sensor
5. Implementing MQ135 as a CO2 gas sensor
6. Implementing MQ7 CO gas sensor
7. Networking
8. Building the database
9. Creating the website

## Cross compiling

Before flashing my application, the Redbear Duo’s firmware required an update, the board is shipped with version 0.2.4 however during the implementation the firmware will be 0.3.1. During the update using “*DFU-util*” is when a major issue occurred, while dumping the “*system-part2*” binaries into the internal flash storage; I miss typed the address space. This caused the “*system-part1*” space to be overwritten while flashing “*system-part2*” and leaving an invalid space occupying part of the internal flash where “*system-part2*” is expected.

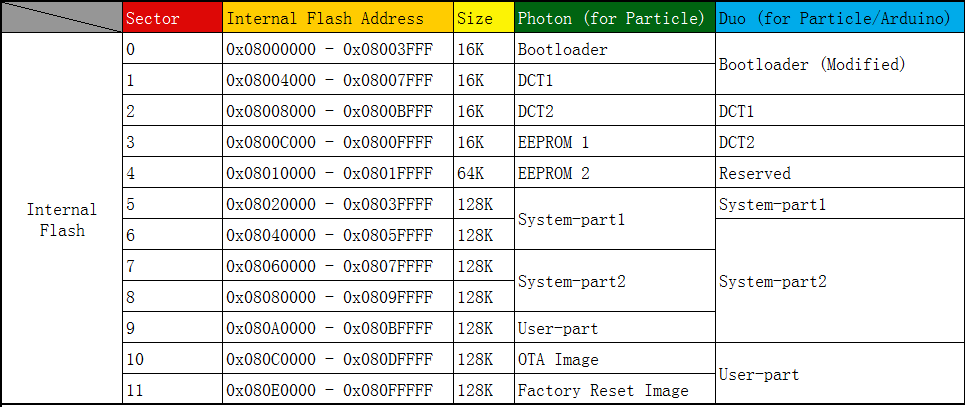


Figure 16 Internal memory map for Redbear Duo

Above in Figure 16 is a small section of the Duo’s memory map to give an idea of the mistake made during the firmware update. I became aware of the issue on reboot when the board flashed red LED’s for SOS due to the hard fault.

Unfortunately at this stage it was impossible to activate either safe mode, or the DFU flash mode to re-flash the board. While researching similar problems and possible solutions I came across one option that offered a possibility of fixing the hard fault, this required using the Redbear Duo’s breakout board which I would need to purchase. Using the breakout board, it was possible to flash the board directly through the I/O pins instead of over the USB serial, however there was no guarantee this would work and required purchasing the breakout board.

A new Redbear Duo cost a little more than the breakout board so the solution was to simply purchase another. After knowing the initial mistake flashing the binaries onto the new board worked flawlessly, with firmware 0.3.1 I could now flash my compiled application to the board without touching the firmware unnecessarily.

## Temperature and Humidity - DHT22

During the research of the DHT22 sensor I discovered a library designed for the Arduino to control the DHT11 and DHT22, this library was designed for the Arduino however could function on the Redbear Duo; ensuring the code calls the correct header.



### Testing

The DHT22 was tested against a thermometer, to measure its accuracy, the library file helps display the temperature and humidity which can be seen below in Figure 17.

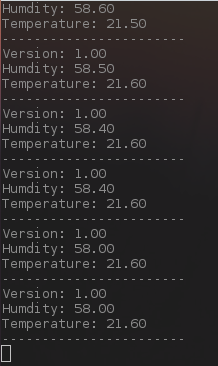


Figure 17 Temperature and Humidity implementation

The library designed by Mark Ruys (Ruys, 2013) uses the BSD licenses allowing free use given the copyright information remains intact.

## GPS – GP-20U7

I encountered a GPS problem very early on, the problem is the GPS picks up a signal but fails to return its location. The RMC data shows we are only receiving a valid date and time stamp, yet displaying the “V” status for void data. However, using a serial UART to USB I can view all the data coming out the GPS serial connection, the module is picking up multiple satellites in view (GSV).



Figure 18 GPS receiving no coordinates

A few steps were taken into solving this issue. Initially it was tested indoors, so the board and GPS was left outside for 5-10 minutes to obtain a reading. This benefited the data it was receiving, such as increasing the visible satellites, however the GPS coordinates were still blank. Secondly was ensuring the GPS module was receiving enough voltage, the working voltage is 3.3v give or take 0.1 volt; the GPS was being supplied 3.2v to 3.3v. This problem appears to be that the GP-20U7 internal antenna is too weak, this problem could potentially be fixed with the addition of an external antenna; however, the GP-20U7 has no socket for an external antenna.

Due to the problem being a lack of signal with no ability to extend the antenna, the solution is replacing the GPS module with an alternative. Ensuring the new board can have an external antenna this issue should not persist. Implementing the new GPS will be done before the demonstration of the project.

## Particulate sensor - PPD42NJ

During the hardware setup of PPD42NJ, the micro USB connected to the Redbear had to be disabled. This is because the PPD42NJ requires powering directly from the battery, to power both the board and the particulate sensor, 5 volts will be delivered to both using the VIN pin on the Redbear to deliver 5 volts which is regulated to 3.3 volts; and 5 volts directly to the PPD42NJ.

This meant it could no longer be used for the serial connection, instead the debugging information will be sent over the UART RX and TX using a USB to UART connection.

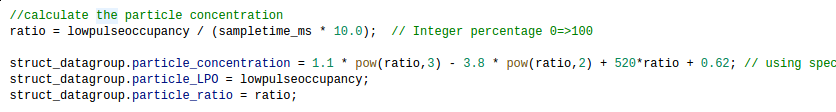


Figure 19 Particulate sensor PPM calculation

Above in Figure 19 is the calculation done to work out the particles within 0.01 cubic feet, this is done after the 30 second LPO time as described in the Design section.

### Testing

Testing was carried out by measuring the particles in an unventilated room compared to the same room and introducing ventilation by opening a large window. As the air ventilated the room, bringing in particles and having the wind move settled particles. The value received from the PDD42 raised accordingly. These values can be seen below in Figure 20.

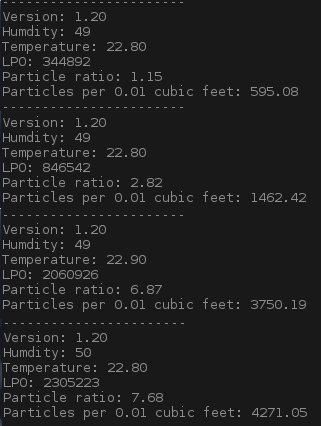


Figure 20 PPD42 Particulate Sensors Testing

## CO2 gas sensor – MQ135

Implementing the MQ135 was problematic, the first issue I encountered was my misinterpretation of the digital and analog output pins. The breakout board offers little explanation within the specification sheet, due to this I initially thought the two pins offered the same output. The MQ135 uses the digital output as a Transistor-transistor logic (TTL) pin, while the analog output is the voltage of 0 – 5 volts; the higher the voltage the higher the concentration of gasses.

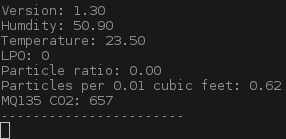


Figure 21 Raw ADC value from MQ135

Figure 21 is the output of the raw analog value from the gas sensor, after correcting this flaw within the schematic and using the Redbear Duo’s analog pin 0.

The raw analog value is not very useful. Within the Design section, I briefly explain the maths behind figuring out the PPM of specific gasses using the MQ135 raw value and the specification curve graph to calculate the correlation.



Figure 22 MQ135 calibration

While implementing the calibration and power regression I found an Arduino project based on (Gironi, 2014) within it by (empierre, 2017). By reading through both these projects I created my own library based heavily on these two for detecting the CO2 using the MQ135 gas sensor, using “*empierre’s*” library I can calibrate the MQ135 sensor before and optionally during the boot up of the device. The library function prototypes and defines are below in Figure 23.

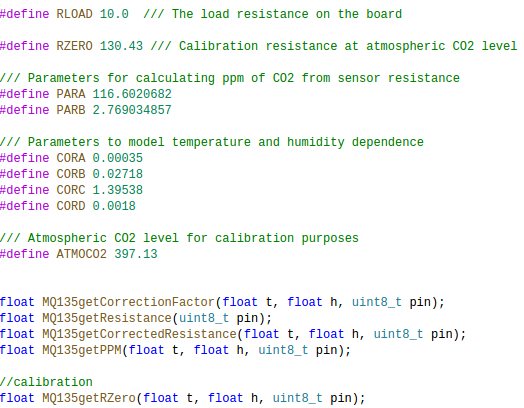


Figure 23 MQ135 Header file snippet

To calculate the CO2 particles per million (PPM), the MQ135 needs to be initially calibrated to a clean environment. This is achieved by initially finding the resistance of the sensor using “*MQ135getRZero*()”, then using “*MQ135getPPM()”* to obtain the raw value and calculate the resistance. While I don’t have access to a completely area for calibration, it was done within a closed off clean room.

### Testing

This calibration is then added to the project as a hardcoded calibration value, although this calibration can be done during the device boot up it requires a clean environment. Due to this requirement, the calibration is done ahead of time to allow start up and shuts down keeping the same resistance. While this can affect results due to changing resistance, having calibration ran while restarting mid journey will result in incorrect results.

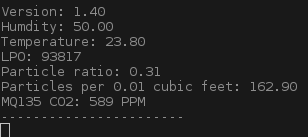


Figure 24 MQ135 Testing within a used room

Figure 24 is the working output after calibration in a relatively clean environment, during this quick test windows were closed and only a small reading was detected. This most likely due to using correlation of the curve graph.

## CO Gas sensor – MQ7

The MQ7 breakout board also uses a digital pin and analog pin, similar to the MQ135. The analog pin outputs the gas sensors voltage for concentration, while the digital pin is used to alternate the voltage being delivered to the gas sensor. The MQ7 requires a high 5 volts for 90 seconds; followed by 60 seconds of 1.4 volts.

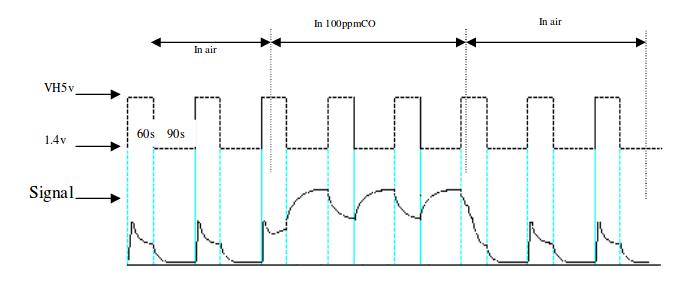


Figure 25 MQ7 Heating Cycle From: <http://www.sainsmart.com/zen/documents/20-011-959/MQ-7.pdf> [Accessed 15 March 2017]

The sensor will handle the voltage levels, I just need to control the digital pin. Which toggles the voltages levels, this process requires a high voltage for 60 seconds followed by 90 seconds of low voltage before the reading can be taken; this is demonstrated in Figure 25.

This pushes back the reading time to 150 seconds before collection instead of the initial 30 seconds. This issue will also effect the sleeping of device when no monitoring is to be taken place, a solution would be turning the voltage to low turning to sleep and initiating the cycle early as discussed.

Another issue using the MQ7 is the lack of libraries. I could only find one library to calculate the PPM from the raw ADC values, and one Arduino project outputting the raw values using the alternating voltages. To solve this issue, I will be developing a library to control the MQ7 gas sensor, which will be a combination of these two projects.

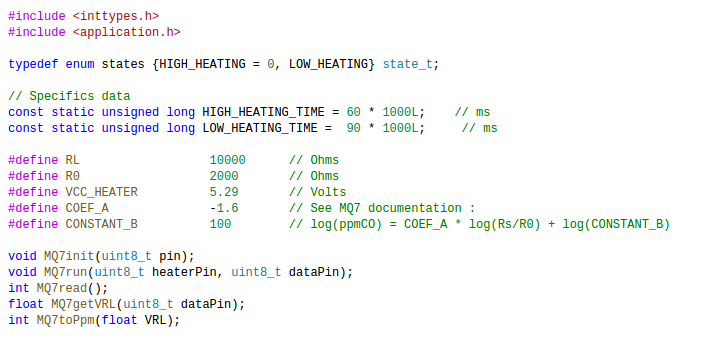


Figure 26 MQ7 CO PPM library

The library in Figure 26 is the implementation for MQ7. This library manages the voltage changes required for the MQ7 within “MQ7run()” as well as calculating the PPM value for carbon monoxide. The biggest issue being unable to create an accurate calibration function for this gas sensor, the solution was to use a “Ro” value from the projects this was based on.

### Testing

The readings from the MQ7 appear to be correct at times, however I started to notice problems with changes and fluctuations in values from both gas sensors and the particulate sensor. The addition of the MQ7 appears to have caused issue with other components, this can be seen below in Figure 27. The particle reading differs greatly between readings, and the CO2 value from the MQ135 is hugely out of proportion. The MQ7 is causing problems with the other sensors, which leads me to believe it’s a hardware problem. At this stage the MQ7 will be left in current state while debugging the problem.

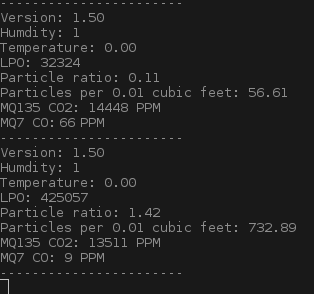


Figure 27 MQ7 Testing

## Voltage issues

During the testing of the MQ7 the readings from all components where returning incorrect values, due to its effect on all components the hardware was a point of interest. Testing began on individual sensors, in case a single one was at fault. This was carried out by removing all sensors and re-attaching them one by one, the issue started to occur when re-attaching the MQ7 last. I began the same process but changed the order of re-attachment, the same issue could be seen attaching the MQ135 at the end.

This lead me down the issue of power and voltage, the PPD42NJ, MQ7 and MQ135 use around 5 volts. The voltage running through the power supply wire is 4.23 volts, this can be seen below in Figure 28.



Figure 28 Low voltage problem

The problem is the high-power consumption from these components, specifically the two gas sensors. As mentioned the gas sensors work by heating coils, to keep these running is very power consuming. Lowering the power of these gas sensors during standby is difficult, due to the start-up time they require for initial heating.

There is no immediate solution to solve this power consumption problem. At this point the device is being powered by a USB power bank, going through a breakout board which delivers 5 volts and up to 2.1 amps. Lithium batteries are usually smaller and offer lower amps then the current battery. Due to this the MQ7 will be removed from the device to keep the other components functional, the CO levels could possibly be implemented using the MQ135 correlation; similar to detecting the CO2.

## Networking

### Finding a network

Connecting to any local Wi-Fi networks was achieved by following the design specification, with the aid of the firmware’s API scanning for local networks was achieved with ease. Storing the known Wi-Fi networks within a simple structure allowed looping through and comparing known networks to those within range. Figure 29 is the structure used to hold these networks.

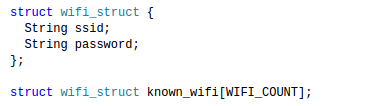


Figure 29 Known WiFi structure

A problem with this method however is the networks are hard-coded within the “WIFI.h” library, altering these networks means flashing the board with the updated code. A future solution would be a feature to send back the Wi-Fi list when the board next uploads results, for comparison and updating.

### Uploading results

Uploading the results consisted of managing the communication from the board, and the data into the server. The website will use “*GET*” to receive the results, which is just a way to transfer information to a webserver by placing it within the URL. A small PHP script called “*insertdata.php*” within the website which parses the information and uploads it as a new reading by the device. An example of URL it receives is the following:

www.merlinroe.com/insertdata.php?sensor\_ID=1&date=2017-03-22&time=12:00:32&temp=17&humid=75&CO=6&CO2=45&parti=3000&lat=423.32&lat\_ori=N&long=356.32&long\_ori=E

To build and send the data in this format the library HttpClient by (nmattisson, 2016) will be used, which offers a simple way pass the data collected over HTTP. If the device is connected to a Wi-Fi network and ready to upload data, it simply loops through all readings taken and runs them through this upload processes. The response from HTTP request return either a positive or negative value, which indicate if the data was uploaded successfully.

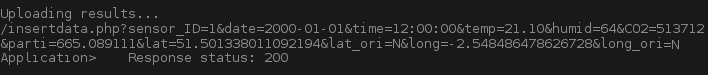


Figure 30 Positive response to webserver

Above in Figure 30 is the response received from the server when uploading a set of data. A positive response indicates a valid upload, while a negative response indicates a failed upload.

## Database

Implementing the database was made easier by using the tool “*PHPmyadmin*”, which is a tool written in PHP intended to handle MySQL. The only issue when creating the database was implementing the GPS coordinates, within the Design section its planned with two floating point values for longitude and latitude. This design does not store the orientation for these values, which is either N, E, S, W. While not all GPS coordinates use these orientations, the currently selected GPS requires them. The solution was to add two new values being “latitude\_orin” and “longitude\_orin” within the implemented database. These changes can be seen below in Figure 31.

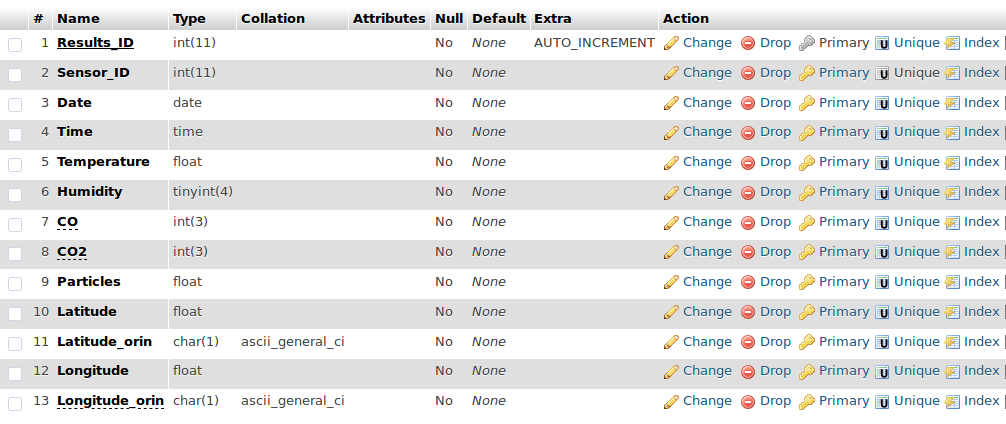


Figure 31 Database Implementation

## Website

Implementing the website was initially done on a Linux virtual machine using Apache for the HTTP server, MySQL for the database and PHP for the website and scripts. The purpose of the virtual machine is to implement and test the website and database on a closed local server, once the website was completed the scripts and web pages where transferred to the “*GoDaddy*” hosted server.

An issue while implanting the website was still having the lack of a GPS to fully test the heatmap. To solve this solution the data initially given was uploaded with estimated GPS coordinates, this allowed me to implement the map integration along with overlaying a heatmap of air quality.

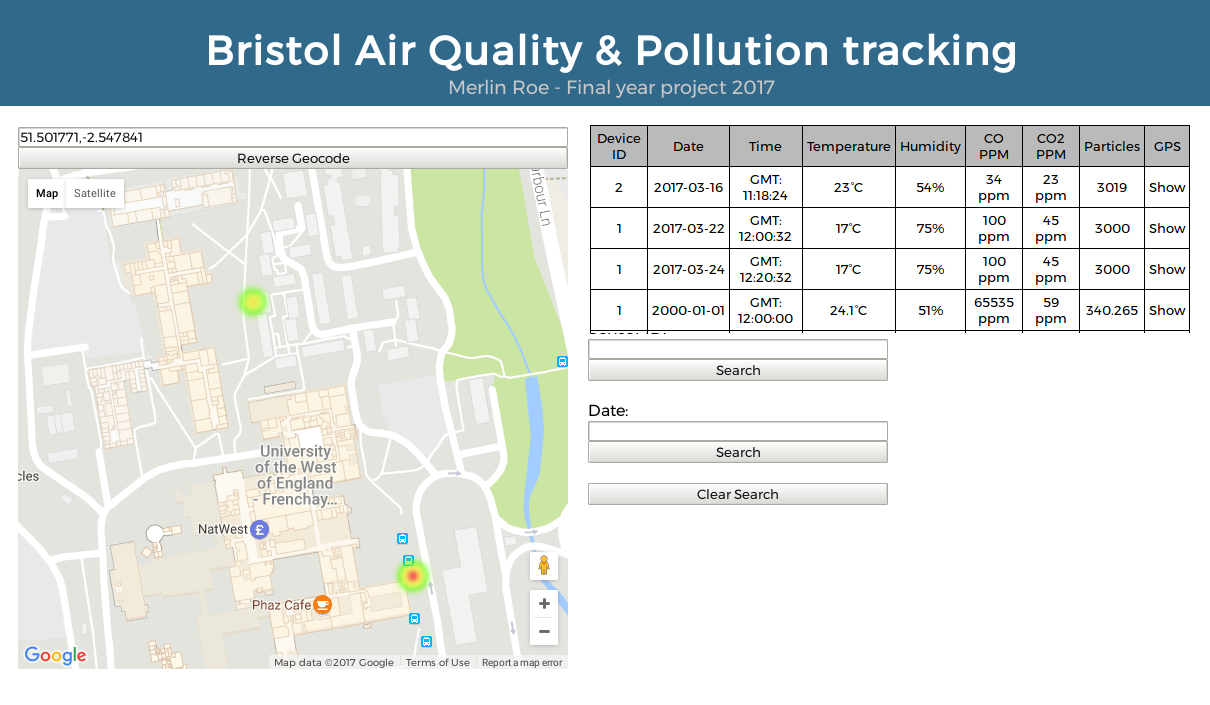


Figure 32 Implemented Website

Figure 32 is the website, staying in a very similar style to the design. You can see the two points on the heatmap overlaid on google maps, these two points correspond to the taken readings. The search features allow filtering by date or by device id, while currently only one device exists the addition was for the possibility of expansion.

The implementation is missing the search by location feature, which was added within the design. The issue is the locations are stored by the longitude and latitude, which makes searching tricky. A possible solution was searching by GPS and offering the closest reading to the input, however this still requires GPS coordinates inputted; which is not user friendly.

The alternative is to implement a way to automatically turn the searched location into GPS coordinate, and continue the search from there. As for this implementation section, it will not be included due to time restrictions, however more research will be done the inclusion within the demonstration.

# Testing

Due to testing being carried throughout the implementation of the project, this section will consist of a sample of acceptance tests, created from the Requirements section. The missing requirements are either lower priority, or the testing can be seen within implementation or as an extension to the following.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Expected result | Steps | Observed result | Pass/Fail |
| PO01-PO04 | System boots without any error’s. | 1: Power on board  2: Check for red LED error flashes. | Board boots and flashes green to indicate connecting to Wi-Fi network. |  |
| GE03 | Readings are taken every 5 minutes. | 1: Power on board | New results are uploaded every 5 minutes. |  |
| GE05 | System will record gas ppm over 30 seconds. | 1: Power on board | The MQ gas sensors do not require reading time, this was not implemented due to being unnecessary. |  |
| GE06 | Device will store results when not connected to the internet. | 1: Power on board  2: No Wi-Fi connection  3: Leave for 15 minutes  4: Enable Wi-FI | No results uploaded for first 15 minutes. When re-enabled Wi-Fi three new readings are uploaded. |  |
| GE08 | Device will change to a known network connection when possible. | 1: Power on board  2: Connect to mobile phone network  3: Disable mobile phone network | Device runs normally until phone network disabled. The device then scans and automatically connects to home network. |  |
| GE12 | If no time stamp can be read from GPS, then will time assumed between one before and after. | 1: Power on board  2: Take first reading  3: Disable GPS  4: Check new results. | Device uses last known GPS coordinates, does not assume location. |  |
| WF01 | System will indicate a warning and status with LEDs. | 1: Power on board  2: Remove Humidity and Temperature sensor to create error. | The board appears to run normally, but does not upload results. |  |
| WF06 | System will indicate if connected to WiFi. | 1: Power on the board  2: connect to Wi-Fi  3: Disconnect from Wi-Fi | The board’s LED is blue to indicate offline and green for online. |  |
| DB01- DB02 | Database to Store Gas, Particle, Temperature, Humidity, GPS coordinates, Time stamps, Date stamps and Device ID within Data table. | 1: N/A | There is a database which stores all readings. |  |
| WI01, WI02, WI03, WI05 | Website available to view all readings, query for specific readings, and view interactive heatmap. | 1: View website  2: View all results  3: Query day  4: Query device  5: Use interactive heatmap | All readings are available online and searchable, heat map is implemented using google maps API. |  |
| SE01-SE03 | Website has security such as: Login, SQL injection protection, verify upload | 1: View website | The site has SQL injection and verify uploads, but does not require login to view readings. |  |

Table 12 Acceptance tests selection

# Conclusion

## Project goal

The project goal was to design and implement a cheap portable air quality and pollution monitor, for tracking urban air quality levels and points of high concentration. The conclusion will evaluate the success and my personal thoughts of the project.

## Critical evaluation

The project was more challenging than initially expected, the major issue being controlling the delicate gas sensors and the maths to calculate the concentrations. The results seen are acceptable in achieving the primary aims laid out within the introduction and requirements, compromises had to be made for the gas monitoring and problems occurred which set back some initial design plans.

The MQ135 accuracy is hard to define, while relaying accurate changes based on the gas concentration; the returned PPM is hard to validated. Calibrating the gas sensor is done within an environment which is assumed to be relatively clean air, the resulting PPM value is calculated by using multiple online sources which are widely adopted; however can’t be locally validated.

The background research investigated the NOx gasses, it quickly became apparent during the primary research that these gasses where very expensive and power consuming; for this type of project. Subsequently CO and CO2 where chosen for measuring pollution while using particulate sensor for general air quality.

The GPS signal strength was a major setback, which cost time in debugging and ordering a replacement module. Researching more GPS options, and thoroughly investigating projects that used them; would have greatly helped in narrowing down the choices. Replacing the GPS module was a simple process, however it added extra time and increased the overall cost of the project.

## What did I learn

I have expanded my knowledge in many areas and learnt several key skills throughout the project. Below are a few of these summarised:

* Building an embedded system from start to finish taught me several key skills, such as working on a project of this scale with a limited time. I had to manage work between research, design, rapid prototyping and testing.
* Researching and implementing hardware for a specific project, this includes wiring and soldering the hardware. Throughout the Computer System Integration course, we learn to code for hardware however; I never learnt to select and implement the hardware which I write applications for.
* This project makes uses of multiple libraries, researching and reading them taught me what to look for when choosing a library. Such as looking for one that meets my exact needs, the licensing and possible examples.
* Researching was something done I had done only in small portions before. Having the opportunity to research a problem and its key areas, and then use that new information to design and implement a possible solution; was a new and a useful skill I learnt.
* Learning how important and how you can implement power efficiency, battery life and efficient code was a new experience for me. I had always known the benefits and reasoning for these, yet putting these practices into effect made the results and purpose much clearer.

## Summary

The project has been challenging, nonetheless it’s been very enjoyable and rewarding. Giving me areas of research and experience that I otherwise would not have engaged in. I believe the result matches my expectations, while the process itself was a new and exciting process. The research throughout the project has lead me to believe in the advancement in monitoring and tackling pollution, and the efforts put towards improving air quality is an ongoing battle.

# Future Work

Rapid prototyping was used to implement the project, due to this development process and limited time resulted in some features or decisions to be incomplete or unexplored. This section will cover additional features or areas to expand upon, and possible paths to explorer during a revision of the project.

## Switching to a Microcontroller

While researching the hardware, I focused on RTOS or Linux based boards. However during the design and implementation, I was thinking about power efficiency, portability and price, that using a microcontroller such as the Atmega 328 would have been a better solution.

A microcontroller has a lower working voltage compared to the Redbear Duo, the price of a microcontroller is around three pounds compared to thirty pounds. The drawbacks are the lack of wireless communication on board, implementing Wi-Fi or Bluetooth could prove to be harder than expected. Programming the board bare-metal compared to RTOS could be difficult. The difference in performance should have a very small impact on a project such as this. A revision of this project may not see a switch, however would see more research invested into microcontrollers.

## Using Bluetooth

While Bluetooth was an option during the research section it lost to Wi-Fi for its ease of use and range. Nonetheless during the implementation Bluetooth started becoming to look like a better option for wireless communication. Connecting to the user’s phone instead of known networks could solve and improve the project in several ways.

Removing the GPS module from the device and relying on the phones on board GPS module, means one less component on the device itself. Which in turn lowers the cost and improves battery life for more portability. The downside of using Bluetooth is adding an extra component in implementation, which is developing the phone application to receive the results, obtain GPS coordinates and then upload them as a complete set of data.

Nevertheless, the benefits would out way the negatives, the reason I did not switch was the limited time. A revision of this project would see a move in the wireless communication to Bluetooth from Wi-Fi.

## Calibrating the Gas sensors

As previously mentioned the gas sensors are calibrated to an assumed clean room. To improve accuracy of this project, having known PPM concentration of gases within a sealed area improve readings. While one of the harder enhancements for this project it is probably the most important improvement.

## Improved interface

The user interaction with the device is minimal, this is mostly due to rapid prototyping; leaving the least important requirements till last. Additional LED’s and buttons could be beneficial in power efficacy and usability. For instance, using a LED to indicate its state by colour, can be used by the user to know if its searching for Wi-Fi, Taking a reading, no GPS data, low battery, etc.

Additionally, an extra button could have been implemented to re-calibrate the gas sensors, or instead of scanning for Wi-Fi networks every 5 minutes; a button could be used by the user to start a scan. This would save power by only using the Wi-Fi module when needed.

# List of Figures

[Figure 1 Rise in Carbon Dioxide to 400ppm in 2016, Climate NASA. Available from: http://climate.nasa.gov/climate\_resources/24/ [Accessed 25 October 2016]. 8](#_Toc478992625)

[Figure 2 Density & Nitrogen Oxides (NOx) Emissions, Demographia. Available from: http://www.demographia.com/ [Accessed 15 February 2017]. 9](#_Toc478992626)

[Figure 1 Size comparisons for PM particles, EPA From: https://www.epa.gov/pm-pollution/particulate-matter-pm-basics#effects [Accessed 14 January 2017]. 21](file:///E:\project\Write_ups\finished\Final_year_project3.docx#_Toc478992627)

[Figure 4 GPS Triangulation From: http://www.avionicswest.com/Articles/howGPSworks.html [Accessed: 15 January 2017] 25](#_Toc478992628)

[Figure 3 MAQUMON pollution map From: http://www.isis.vanderbilt.edu/projects/maqumon [Accessed 16 January 2017] 25](file:///E:\project\Write_ups\finished\Final_year_project3.docx#_Toc478992629)

[Figure 6 Bristol GSM coverage From: http://www.which.co.uk/reviews/mobile-phone-providers/article/mobile-phone-coverage-map [Accessed 16 March 2017] 29](#_Toc478992630)

[Figure 7 Layered Architecture Design 42](#_Toc478992631)

[Figure 8 Layers within Architecture 42](#_Toc478992632)

[Figure 9 Finite State Machine 43](#_Toc478992633)

[Figure 10 DHT22 Temperature and Humidity Design 44](#_Toc478992634)

[Figure 11 Sensitivity characteristics of the MQ135 From: MQ135 Specification 46](#_Toc478992635)

[Figure 12 Wi-Fi Finite state diagram 48](#_Toc478992636)

[Figure 13 Hardware Schematic 49](#_Toc478992637)

[Figure 14 Database Design 50](#_Toc478992638)

[Figure 15 Website Design 51](#_Toc478992639)

[Figure 16 Internal memory map for Redbear Duo 53](#_Toc478992640)

[Figure 17 Temperature and Humidity implementation 54](#_Toc478992641)

[Figure 18 GPS receiving no coordinates 55](#_Toc478992642)

[Figure 19 Particulate sensor PPM calculation 56](#_Toc478992643)

[Figure 20 PPD42 Particulate Sensors Testing 56](#_Toc478992644)

[Figure 21 Raw ADC value from MQ135 57](#_Toc478992645)

[Figure 22 MQ135 calibration 57](#_Toc478992646)

[Figure 23 MQ135 Header file snippet 58](#_Toc478992647)

[Figure 24 MQ135 Testing within a used room 58](#_Toc478992648)

[Figure 25 MQ7 Heating Cycle From: http://www.sainsmart.com/zen/documents/20-011-959/MQ-7.pdf [Accessed 15 March 2017] 59](#_Toc478992649)

[Figure 26 MQ7 CO PPM library 60](#_Toc478992650)

[Figure 27 MQ7 Testing 60](#_Toc478992651)

[Figure 28 Low voltage problem 61](#_Toc478992652)

[Figure 29 Known WiFi structure 62](#_Toc478992653)

[Figure 30 Positive response to webserver 62](#_Toc478992654)

[Figure 31 Database Implementation 63](#_Toc478992655)

[Figure 32 Implemented Website 64](#_Toc478992656)

# List of Illustrations

[Illustration 1 NOx Monitor in Fishponds, Bristol Air Quality, Avaiable from: http://www.bristol.airqualitydata.com/cgi-bin/sites.cgi?1010 [accessed 22 February 2017] 10](file:///E:\project\Write_ups\finished\Final_year_project3.docx#_Toc478992657)

[Illustration 2 Open Source Initiative Logo From: https://opensource.org/logo-usage-guidelines [Accessed 2 March 2017] 14](#_Toc478992658)

[Illustration 3 Raspberry Pi 3-B, Raspberry Pi, Available From: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ [ Accessed 20 February 2017] 16](#_Toc478992659)

[Illustration 4 Particle Electron, Particle, Available from: https://store.particle.io/ [Accessed 20 February 2017] 17](#_Toc478992660)

[Illustration 5 Redbear Duo, RedBear, Available from: https://github.com/redbear/Duo/blob/master/docs/duo\_introduction.md [Accessed 20 February 2017] 18](#_Toc478992661)

[Illustration 6 MSP-EXP430F5529LP, Texas Instruments, Available from: http://www.ti.com/tool/MSP-EXP430F5529LP [Accessed 20 February 2017] 19](#_Toc478992662)

[Illustration 7 MQ7 Gas sensor, From: https://www.sparkfun.com/products/9405 [Accessed 14 January 2017]. 23](#_Toc478992663)

[Illustration 8 MySQL Logo From: https://www.mysql.com/ [Accessed 24 February 2017] 32](#_Toc478992664)

# List of Tables

[Table 1 Raspberry Pi 3-B Specs 16](#_Toc478992665)

[Table 2 Particle Electron Specs 17](#_Toc478992666)

[Table 3 Redbear Duo Specs 18](#_Toc478992667)

[Table 4 MSP-EXP430F5529LP Specs 19](#_Toc478992668)

[Table 5 Temperature & Humidity sensor comparison 20](#_Toc478992669)

[Table 6 Particulate Matter sensor comparison 22](#_Toc478992670)

[Table 7 Gas sensor comparison 24](#_Toc478992671)

[Table 8 GPS comparison 26](#_Toc478992672)

[Table 9 Power Source option comparison 31](#_Toc478992673)

[Table 10 Database comparison 33](#_Toc478992674)

[Table 11 Requirements tables 40](#_Toc478992675)

[Table 12 Acceptance tests selection 66](#_Toc478992676)

# References

Anderson, J. O., Thundiyil, J. G. & Andrew, A., 2012. Clearing the Air: A Review of the Effects of Particulate Matter Air Pollution on Human Health. *Journal of Medical Toxicology,* pp. 166-175.

Seeed Development Limited, 2015. *Grove - Dust Sensor.* [Online]   
Available at: https://www.seeedstudio.com/Grove-Dust-Sensor-p-1050.html#  
[Accessed 23 February 2017].

Amazon Web Services, Inc, 2017. *Amazon RDS.* [Online]   
Available at: https://aws.amazon.com/rds/  
[Accessed 2 March 2017].

Amazon.com, Inc., 2017. *Duracell Plus Power Type 9V Alkaline Batteries, Pack of 4 - Amazon.* [Online]   
Available at: https://www.amazon.co.uk/Duracell-MN1604-Plus-Power-Batteries-Pack/dp/B005KP74BI/ref=sr\_1\_2?s=electronics-accessories&ie=UTF8&qid=1488559933&sr=1-2&keywords=9v+battery  
[Accessed 2 March 2017].

Amazon.com, Inc, 2017. *Coolreall Power Bank 15600mAh 2 Port USB Compact: Amazon.co.uk.* [Online]   
Available at: https://www.amazon.co.uk/dp/B016Y0FO2I/ref=sr\_ph\_1?ie=UTF8&qid=1488560259&sr=sr-1&keywords=power+bank  
[Accessed 2 March 2017].

Ana, F. T. & Octavian, P., 2014. Wireless sensor network and web based information system for asthma trigger factors monitoring. *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings 2014 IEEE International,* pp. 1388-1393.

Ardiuno, 2016. *MQ Gas sensors.* [Online]   
Available at: http://playground.arduino.cc/Main/MQGasSensors  
[Accessed 23 February 2017].

Batteries.com, 2017. *Batteries.* [Online]   
Available at: https://www.batteries.com/#  
[Accessed 2 March 2017].

Bluetooth SIG, Inc, 2017. *Bluetooth Low Energy | Bluetooth Technology Website.* [Online]   
Available at: https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy  
[Accessed 3 March 2017].

Brennan, K., 2009. *A Guide to the Business Analysis Body of Knowledger.* s.l.:International Institute of Business Analysis.

Bristol City Council, University of Bristol, 2017. *Bristol is open.* [Online]   
Available at: http://www.bristolisopen.com/  
[Accessed 22 February 2017].

Bristol City Council, 2017. *Bristol Air Quality.* [Online]   
Available at: http://www.bristol.airqualitydata.com/index.cgi  
[Accessed 22 February 2017].

Creative Commons Attribution 4.0 International License, 2017. *International Authority & Recognition | Open Source Initiative.* [Online]   
Available at: https://opensource.org/authority  
[Accessed 3 March 2017].

David, C., n.d. *Scripps CO2 Program.* [Online]   
Available at: http://scrippsco2.ucsd.edu/history\_legacy/keeling\_curve\_lessons  
[Accessed 21c February 2015].

Department for Environment, F. &. R. A., 2015. Air Quality Plan for the achievement of EU air quality limit value for nitrogen dioxide (NO2) in Bristol Urban Area (UK0009). *Environmental quality,* p. 44.

DFRobot , 2017. *GPS Module with Enclosure-DFRobot.* [Online]   
Available at: https://www.dfrobot.com/product-1302.html  
[Accessed 4 March 2017].

empierre, E., 2017. *empierre.* [Online]   
Available at: https://github.com/empierre  
[Accessed 6 March 2017].

European.eu, 2017. *European Green Capital.* [Online]   
Available at: http://ec.europa.eu/environment/europeangreencapital/winning-cities/2015-bristol/  
[Accessed 23 February 2017].

Farnell, 2015. *SENSIRION SHT75 Humidity Sensor.* [Online]   
Available at: http://uk.farnell.com/sensirion/sht75/sensor-humidity-temp-3-3v/dp/1590514  
[Accessed 23 February 2017].

Geier, J., 2001. *IEEE 802.11 Services | Overview of the IEEE 802.11 Standard | InformIT.* [Online]   
Available at: http://www.informit.com/articles/article.aspx?p=24411&seqNum=7  
[Accessed 2 March 2017].

Gironi, D., 2014. *Davide Gironi: Cheap CO2 meter using the MQ135 sensor with AVR ATmega.* [Online]   
Available at: https://davidegironi.blogspot.co.uk/2014/01/cheap-co2-meter-using-mq135-sensor-with.html  
[Accessed 5 March 2017].

GoDaddy Operating Company, LLC, 2017. *GoDaddy.* [Online]   
Available at: https://uk.godaddy.com/  
[Accessed 2 March 2017].

Google, 2017. *Google Maps.* [Online]   
Available at: https://www.google.co.uk/maps/  
[Accessed 20 March 2017].

Hanwei Electronics, n.d. *MG811 Datasheet.* [Online]   
Available at: http://www.mouser.com/ds/2/321/CO2SensorDatasheetMG811-17599.pdfc  
[Accessed 23 February 2017].

Heimbinderc, H., Yap, R., Cosentino , C. & Berg, G., 2016. *AirBeam.* [Online]   
Available at: http://www.takingspace.org/aircasting/airbeam/  
[Accessed 7 November 2016].

HobbyTronics Ltd, 2017. *56 Channel GPS Receiver - GP-20U7 | GPS-13740 | SparkFun.* [Online]   
Available at: http://www.hobbytronics.co.uk/gps-gp-20u7  
[Accessed 4 March 2017].

IEEE, 2012. *IEEE Standards.* [Online]   
Available at: http://standards.ieee.org/getieee802/download/802.11-2012.pdf  
[Accessed 23 February 2017].

Institute for Software Integrated Systems, 2011. *Mobile Air Quality Monitoring Network.* [Online]   
Available at: http://www.isis.vanderbilt.edu/projects/maqumon  
[Accessed 22 February 2017].

Jin, J., Gubbi, J., Marusic, S. & Palaniswami, M., 2014. An Information Framework for Creating a Smart City Through Internet of Things. *IEEE Internet of Things Journal ,* 1(2), pp. 28-34.

Kampa, M. & Castanas, E., 2008. Environmental Pollutionc. *Environmental Pollution,* 151(2), pp. 362-367.

Kampa, M. & Castanas, E., 2008. Human health effects of air pollution. *Environmental Pollution,* 151(2), pp. 362-367.

Laurent, A. M. S., 2004. Understanding Open Source and Free Software Licensing. In: *Understanding Open Source and Free Software Licensing.* s.l.:O'Reilly Media, p. 4.

Libelium Comunicaciones Distribuidas, 2015. *Particle sensor.* [Online]   
Available at: http://www.libelium.com/particle-matter-dust-sensor-pm1-pm25-pm10-air-quality-smart-cities/  
[Accessed 23 February 2017].

LoRa Alliance, 2017. *LoRa Technology.* [Online]   
Available at: https://www.lora-alliance.org/What-Is-LoRa/Technology  
[Accessed 2 March 2017].

LoRa Alliance, 2017. *LoRaWAN For Developers.* [Online]   
Available at: https://www.lora-alliance.org/For-Developers/LoRaWANDevelopers  
[Accessed 2 March 2017].

Makersify LTD, 2017. *DFRobot 6xAA Battery Holder(double layer).* [Online]   
Available at: https://makersify.com/products/dfrobot-6xaa-battery-holderdouble-layer  
[Accessed 2 March 2017].

Makersify LTD, 2017. *DFRobot 7.4V Lipo 2200mAh Battery (Arduino Power Jack) - Makersify.* [Online]   
Available at: https://makersify.com/products/dfrobot-7-4v-lipo-2200mah-battery-arduino-power-jack  
[Accessed 2 March 2017].

Makersify, 2017. *dafruit FONA 808 - Mini Cellular GSM + GPS Breakout - Makersify.* [Online]   
Available at: https://makersify.com/products/adafruit-fona-808-mini-cellular-gsm-gps-breakout?variant=5057794305  
[Accessed 4 March 2017].

MicroController Pros LLCc, 2017. *Ozone (O3) Sensor, MiCS-2614, in MiCS-2610 Footprint, 10-1000ppb.* [Online]   
Available at: http://microcontrollershop.com/product\_info.php?products\_id=6791  
[Accessed 23 February 2017].

nmattisson, 2016. *nmattisson/HttpClient.* [Online]   
Available at: https://github.com/nmattisson/HttpClient  
[Accessed 22 March 2017].

Office, P., 2007. Regulation (EC) No 715/2007 — type approval of light passenger and commercial vehicles with respect to emissions (Euro 5 and Euro 6) and access to vehicle repair and maintenance information. *Official Journal of the European Union,* Volume 50, p. 45.

Oracle Corporation, 2017. *MySQL.* [Online]   
Available at: https://www.mysql.com/  
[Accessed 25 February 2017].

Parallax Inc, 2017. *PAM-7Q GPS Module | 28509 | Parallax Inc.* [Online]   
Available at: https://www.parallax.com/product/28509  
[Accessed 4 March 2017].

Particle Industries Inc, 2015. *Particle Datasheets.* [Online]   
Available at: https://docs.particle.io/datasheets/electron-datasheet/  
[Accessed 23 February 2017].

Peter, M., 2011. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems.* 2nd ed. s.l.:s.n.

Puri, I. K. & Annamalai, K., 2006. *Kalyan Annamalai.* s.l.:CRC Press.

Raspberry Pi Foundation, 2016. *Raspberry Pi 3 B.* [Online]   
Available at: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/  
[Accessed 22 February 2017].

Redbear Limited, 2016. *Redbear Duo.* [Online]   
Available at: https://github.com/redbear/Duo/blob/master/docs/duo\_introduction.md  
[Accessed 22 February 2017].

Redline Batteries, 2007. *manoonpong.* [Online]   
Available at: http://www.manoonpong.com/Other/main\_page=page\_2.pdf  
[Accessed 2 March 2017].

RS Components Ltd, 2011. *Honeywell HIH-6120-021-001 Temperature & Humidity Sensor | HoneyWell.* [Online]   
Available at: http://uk.rs-online.com/web/p/temperature-humidity-sensors/7811365/  
[Accessed 23 February 2017].

Ruys, M., 2013. *markruys/arduino-DHT: Efficient DHT library for Arduino.* [Online]   
Available at: https://github.com/markruys/arduino-DHT  
[Accessed 5 March 2017].

Salkintzis, A., Fors, C. & Pazhyannur, R., 2002. WLAN-GPRS integration for next-generation mobile data networks. *IEEE Wireless Communications ,* 9(5).

Shinyei Technology Co.,LTD, 2016. *PPD42NJ Paricle Sensor Unit.* [Online]   
Available at: http://www.shinyei.co.jp/stc/eng/optical/main\_ppd42.html  
[Accessed 23 February 2017].

Sirio, G. D., 2017. *ChibiOS free embedded RTOS - RTOS Concepts.* [Online]   
Available at: http://www.chibios.org/dokuwiki/doku.php?id=chibios:articles:rtos\_concepts  
[Accessed 2 March 2017].

SparkFun Electronics, 2017. *GPS Receiver - EM-506 (48 Channel).* [Online]   
Available at: https://www.sparkfun.com/products/12751  
[Accessed 4 March 2017].

Sparkfun, 2010. *DHT22.* [Online]   
Available at: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf  
[Accessed 23 February 2017].

SparkFun, 2014. *Optical Dust Sensor - GP2Y1010AU0F.* [Online]   
Available at: https://www.sparkfun.com/products/9689  
[Accessed 23 February 2017].

STMicroelectronics, 2017. *BlueNRG-MS - Bluetooth Low Energy Network Processor supporting Bluetooth 4.1 core specification - STMicroelectronics.* [Online]   
Available at: http://www.st.com/en/wireless-connectivity/bluenrg-ms.html?icmp=pf260894\_pron\_nb\_jan2015&sc=bluenrg-ms  
[Accessed 2 March 2017].

Stravers, P., 1994. *Embedded System Design.* Delft: s.n.

Suehring, S., 2002. *MySQL Bible.* 1st ed. s.l.:Wiley Publishing, Inc.

Texas Instruments Incorporated, 2015. *MSP430F5529 USB LaunchPad Evaluation Kit.* [Online]   
Available at: http://www.ti.com/tool/MSP-EXP430F5529LP  
[Accessed 23 February 2017].

Tikuisis, P. et al., 1992. Rate of formation of carboxyhemoglobin in exercising humans exposed to carbon monoxide. *Journal of Applied Physiology,* 72(4), pp. 1311-1319.

Timerlake, B., 2017. *LampHowTo.* [Online]   
Available at: http://lamphowto.com/  
[Accessed 2 March 2017].

Völgyesi, P., Nádas, A., Lédeczi, Á. & Koutsoukos, X., 2008. *Air Quality Monitoring with SensorMap.* s.l., IEEE.

Wi-Fi Alliance, 2017. *Certification | Wi-Fi Alliance.* [Online]   
Available at: https://www.wi-fi.org/certification  
[Accessed 2 March 2017].

W. W. g., 2003. *Health Aspects of Air Pollution with Particulate Matter, Ozone and Nitrogen Dioxide,* Bonn, Germany: WHO.

Xia, F., Yang, L., Wang, L. & Vinel, A., 2012. Internet of Things. *International Journal of Communication Systems,* 25(9), pp. 1101-1102.

Zanella, A. et al., 2014. Internet of Things for Smart Cities. *IEEE Internet of things Journal,* 1(1), pp. 22-33.

# Appendices

# Project Plan

Below in Figure 33 and Figure 34 is the project plan and Gant chart, which where outlined at the start of the project. This plan was followed throughout the documentation, however faltered near the end of the project due to time restrictions.

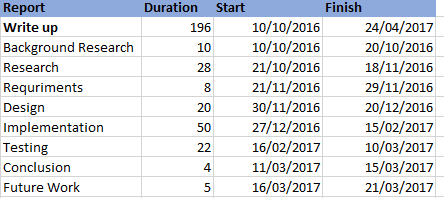


Figure 33 Project Plan

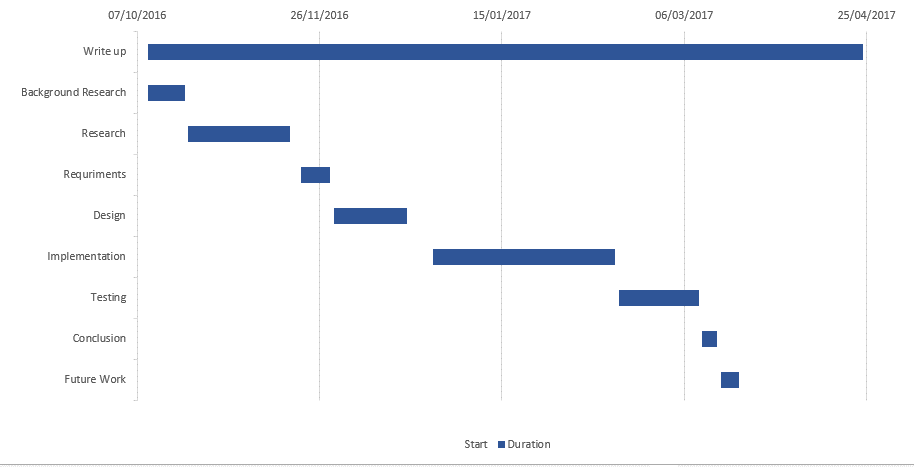
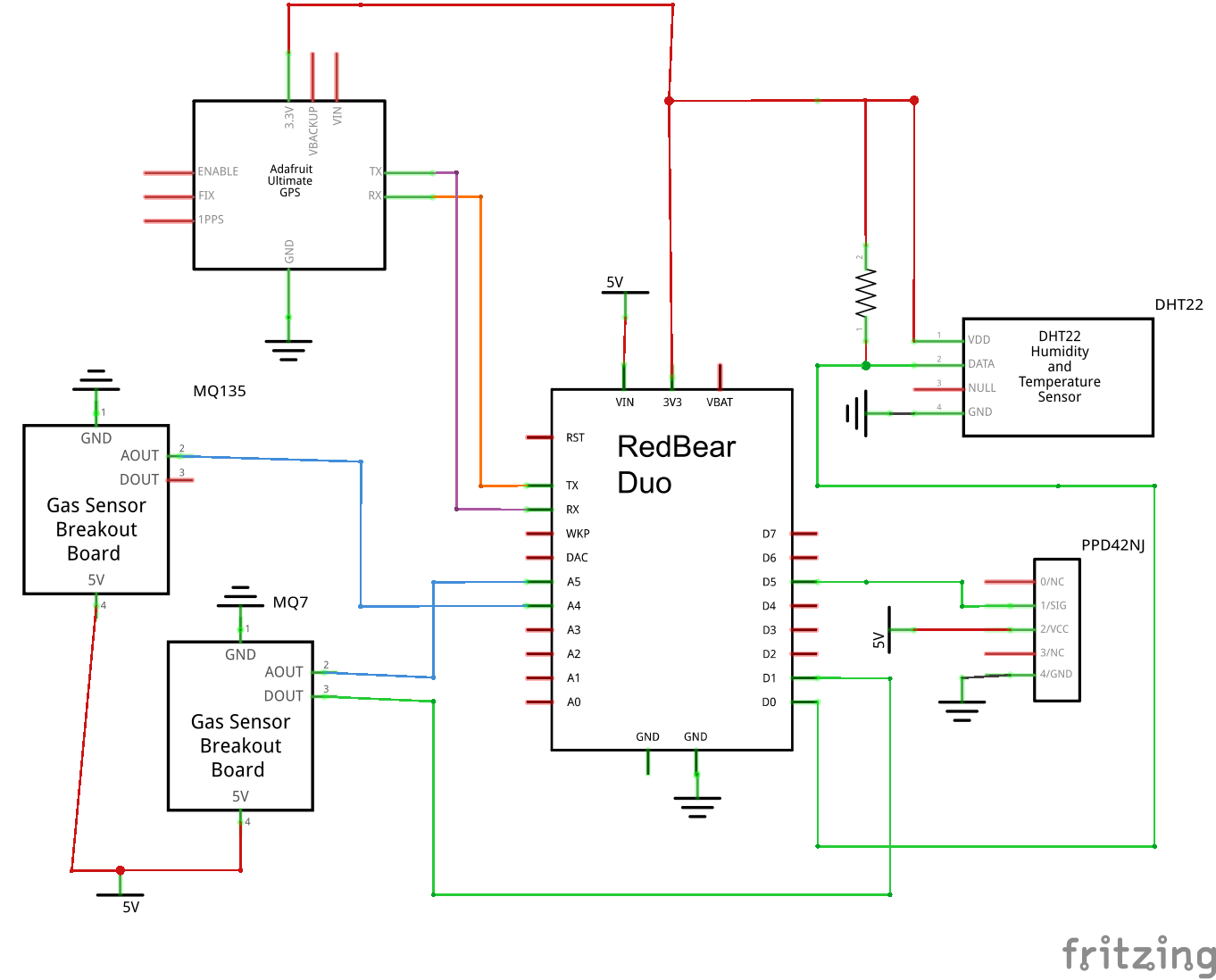


Figure 34 Project Gant chart

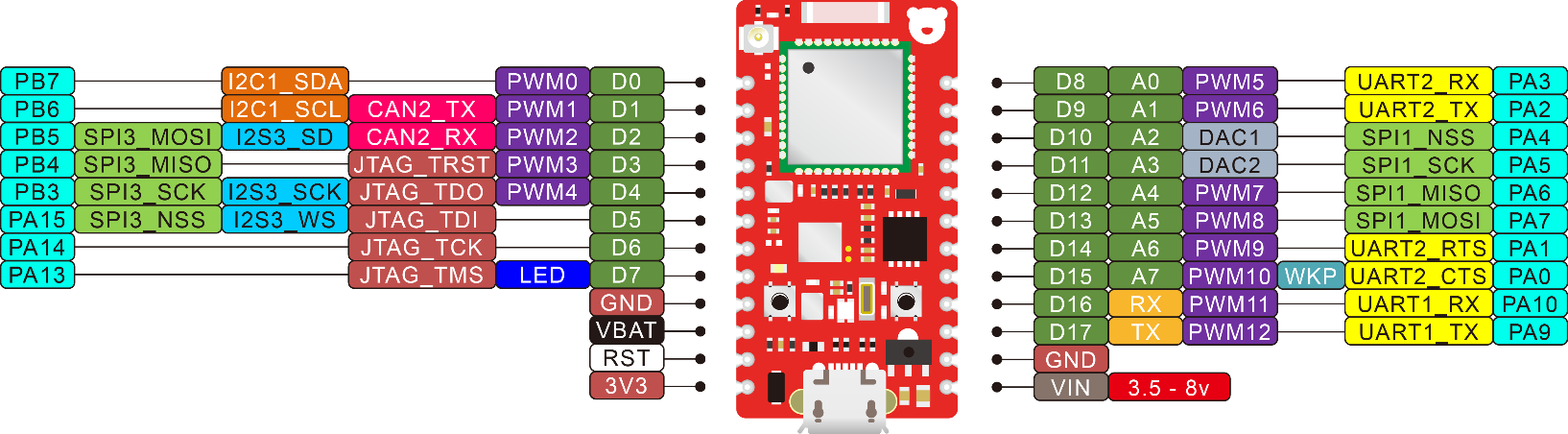
# Hardware Specification

Revised hardware schematic

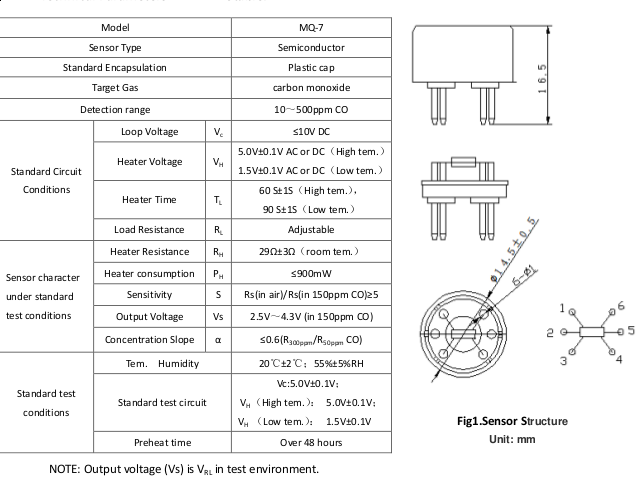
Below is the new hardware schematic being used, which was designed during and after implementation.

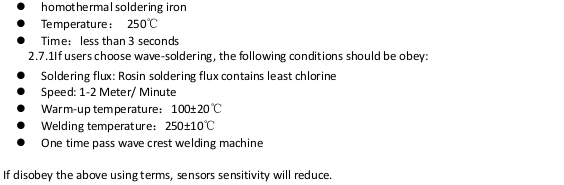


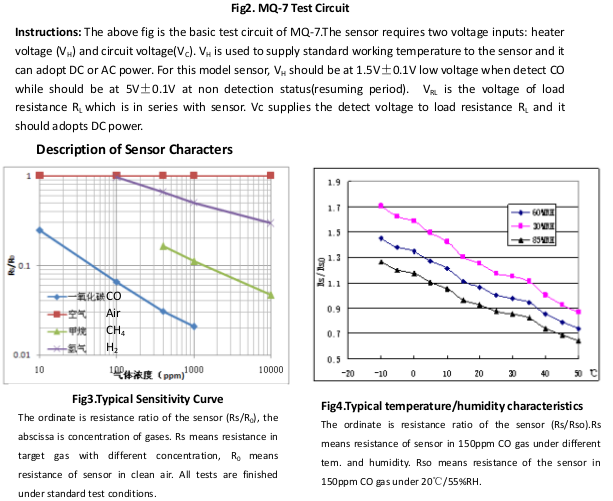
Redbear Duo

* STMicroelectronics STM32F205, ARM 32-bit Cortex-M3 @120 MHz
* AMPAK AP6212A Wireless Module (built around Broadcom BCM43438):
  + Wi-Fi 802.11b/g/n, working on 2.4 GHz ISM Band
  + Bluetooth 4.1 (Dual Mode), working on 2.4 GHz ISM Band
* Memories:
  + 1MB internal flash
  + 2MB external SPI flash
  + 128KB SRAM
* User Interface:
  + 1 x RGB LED
  + 1 x single LED
  + 1 x RESET button
  + 1 x user button
* I/O Capabilities:
  + 18 x Digital I/O
  + 8 x ADC Input
  + 2 x DAC Output
  + 13 x PWM
  + Connectivity
  + 2 x UART
  + 2 x SPI
  + 1 x I2S
  + 1 x I2C
  + 1 x CAN
  + 1 x High Speed USB
  + JTAG(SWD) debug port
* Development Platform:
  + Arduino IDE
  + Particle Build(Web IDE)
  + Espruino Web IDE
  + Broadcom WICED SDK
  + ARM GCC
* Real-time operating system (FreeRTOS)
* Hardware and software open source
* Single-sided PCBA for easy mounting on other PCB
* Breadboard frindly
* Alternative signal chip antenna or external antenna
* FCC and CE certified

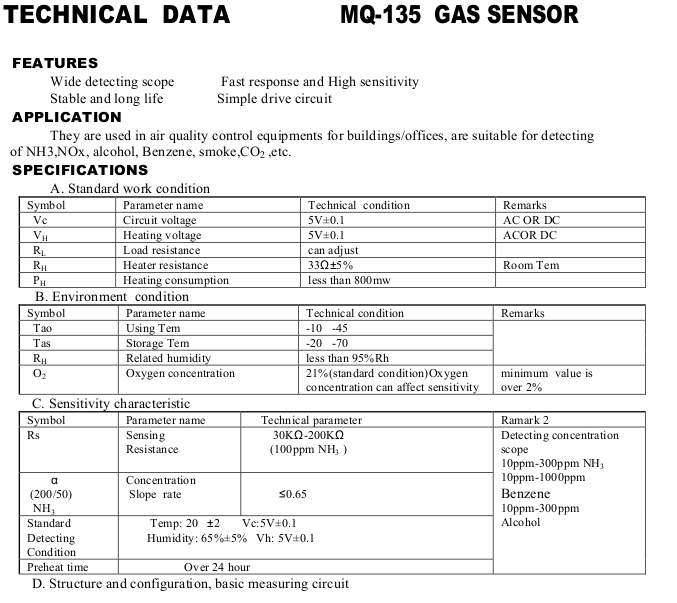
MQ7

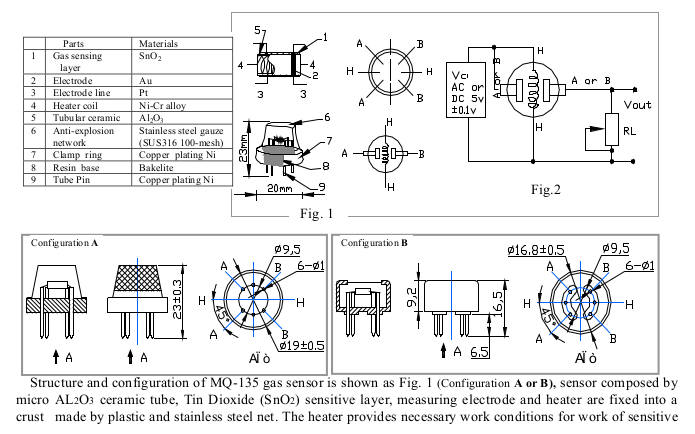


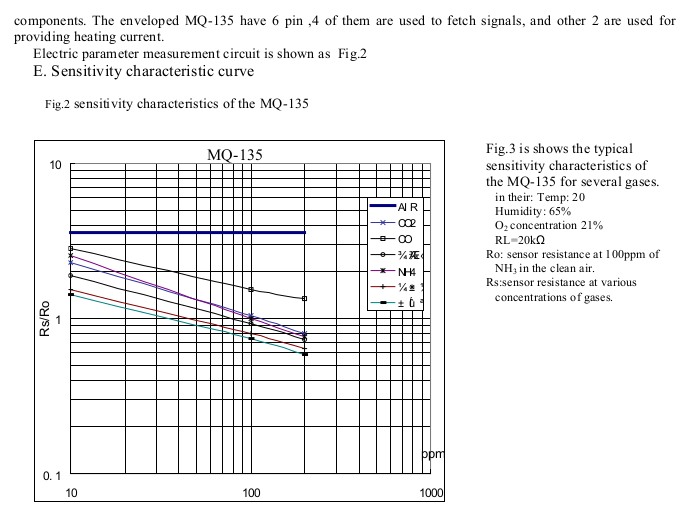




MQ135







PPD42NJ