Institiúid Teicneolaíochta Cheatharlach

INSTITUTE *of* TECHNOLOGY CARLOW

PiLYNK

Home Automation System

Research Manual

Project Name:	PiLYNK Home Automation
Student Name:	Keith Byrne
Student ID:	C00170460
Class Group:	Software Development
Supervisor:	Joseph Kehoe
Document:	Research Manual
Date:	15/04/2016

Table of Contents

1. Purpose

The aim of this research manual is to identify areas of research conducted prior to the development of the Pi LYNK Home Operating System (Home-OS) for remote administration of appliances and lighting from anywhere in the world and extended automation of aforementioned appliances and lighting.

This document will provide an audit trail of technologies considered, evaluated and possibly tested in some cases. Subsequently, a choice of the most ideal technologies that will be used in the development of the project will be made. Furthermore, any further research made into previously overlooked technologies will be included thereafter and details of such technologies will be detailed as encountered and researched.

2. Introduction

2.1 Overview

With the rising costs of electricity usage due to the dwindling supply of finite resources that provide a large portion of the world's energy supply, making small changes to our daily behaviour can have a large, positive impact on energy conservation. Many of us in our daily pursuits leave our homes without a care as to what unnecessary energy consumption is actually doing to the physical state of planet. For example, leaving appliances on because it's more convenient to simply leave them on, (phone chargers, standby's, microwave idle control, etc.) rather than going through the effort of turning it off and back on when it's ultimately necessary.

It is true that people have very hectic schedules and simply may not have the time available to be concerned about these seemingly trivial issues. However, upon analysing the actual costs of unnecessary power consumption, even with a small number of idle phone chargers, laptop chargers, one can understand the need for a change in how people view this issue. As these electrical items are still active and drawing a current.

The following points are a brief overview of the impact that even leaving the house for two hours can have:

- The increase of the carbon footprint and pollution of the ecosystem.
- Higher cost of living and inorganic demand on energy suppliers.
- Maintenance costs of appliances and increases in waste disposal.

The project idea aims to help people in reducing this impact by creating a system that makes it easy to control what's on and what's not, even when that person is not even in the home and also help inform and habituate homeowners to more sensible conversation tactics. Furthermore, to allow householders to monitor activity in an effort to cut costs of the electricity bill (appliance by appliance basis) and potentially regulate usage of certain systems, such as TV's after hours for children.

3. An Overview of Home Operating Systems

Modern homes can be viewed much like a fully functioning computer system, much like a desktop computer, laptop or smart phone. Homes are typically occupied with components that serve particular functions for home owners, ranging from toasters to thermostats, analogous to soundcards and ram modules or CPU's and PSU's. This leads to a new form of thinking in how we go about our daily lives and how our homes can work for us, and us for our homes. To achieve this almost symbiotic harmony, the following statement from Colin Dixon and co-authors certainly holds true:

"The home needs an operating system..." [1]**.**

Generally speaking, we view the home as a heterogeneous group of appliances, lighting options, heating components, etc. Subsequently, we argue that heterogeneity is hindering technological innovation in the home—homes differ in terms of their devices and how those devices are connected and used. To abstract these differences, a technical medium or bridge must be proposed.

The concept of smart, IoT connected homes has been around for well over twenty-something years. Pertaining to this vision is the concept of users being able to easily perform tasks that involve diverse sets of household devices in their home without needing to wade through a technical minefield of configuration, connectivity and the relevant programming requirements.

It may become common place in twenty years from now that all new homes will come smart home equipped. In which case, the need for such systems to be of a high standard is absolutely certain.

Appliances
Lighting
Measure
Global Control

4. Similar Existing Technologies

4.1 Belkin WeMo

The Belkin Wi-Fi enabled WeMo Switch lets you turn electronic devices on or off from anywhere. The WeMo Switch uses your existing home Wi-Fi network to provide wireless control of TVs, lamps, stereos, heaters, fans and more. Simply download the free WeMo App from the Google Play Store or the Apple App store, plug the switch into an outlet in your home, and plug any device into the switch. You'll be able to turn that device on or off using your smartphone or tablet (running Android 4.0 and later or iOS 5 or higher). You can set schedules for your devices and control them remotely using a mobile internet connection. You can also add additional switches to your home to control more devices [2].

The WeMo range is completely modular and stands out as the best currently marketed product in this line of Home Operating System services. However, the technology comes at a considerable price tag. A single Belkin WeMo Automation Switch current retails at sixty U.S dollars [2] [3], as seen in in Figure 4.a.

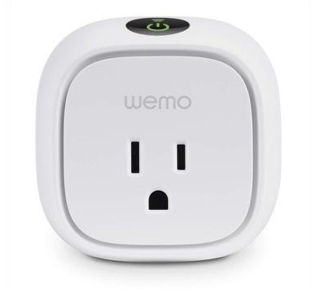


Figure 4.a - Belkin WeMo Automation Switch [3]

4.2 Light Wave RF

Light Wave RF is a wide scale Home-OS service manufacturer. Light Wave produce modular technologies for appliances, heating, energy, sensors and lighting [4]. In addition to this, Light Wave RF also provide high end products for commercial/corporate customers for offices, warehouses and other interior settings. Due to the larger scale building management solutions options, Light Wave RF is an unattainable benchmark. However, in terms of this projects scope, a small subset of their services can be used for inspiration.

Light Wave RF allows users to connect with a suite of various remote access devices such as smartphone and tablet for controlling the central hub of the Light Wave RF systems. Due to this, users are connected no matter where they are currently located. Light Wave RF products are sold through Maplin, due to this, prices are not static.

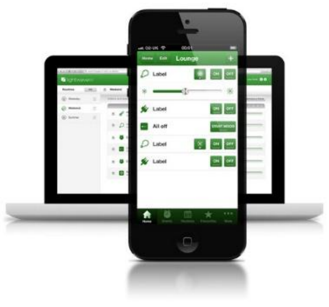


Figure 4.b - Light Wave RF Device Suite [4].

4.3 ADT Security Automation

ADT Security offer a very complex range of home automation systems ranging from: ADT Pulse Features - Security, Burglary, Fire & Flood Alarms security, fire alarms, carbon monoxide air monitoring systems, video surveillance, temperature control, locks and garage door automation, lighting, flood monitoring and voice automation services [5].

Although very specialist in nature compared to the nature of this project, the concept is similar. Automated security is also becoming an increasingly hot topic as trust in conventional services such as door triggered alarms relies on the intervention of policing authorities to have the systems maximum effect.

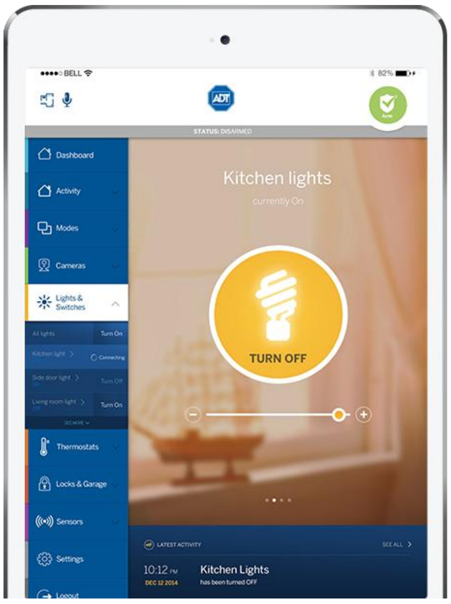


Figure 4.c - ADT Security Application Dashboard [5].

4.4 Philips Hue

Figure 4.d - Hue Bulb & Hub [6]

Philips Hue is a home automation system developed by Philips. This series of automation technology focuses on highly specialised lighting for homes. The technology offers home owners ambient, sensual and mood lighting via the use of Hue modules (Hue bulbs) that are controlled using a Hue hub [6] as illustrated in Figure 4.d. Hue also offers a series of very interesting features with the Hue, such as audio tempo controlled variable lighting and a lighting array containing sixteen million colours. Philips Hue currently has control integration with Apple HomeKit [7].

5. System Hardware – Big Versus Small?

For the system to be small enough to be hidden, quiet, reliable and yet powerful enough to operate as a central hub, receive network requests, broadcast radio frequencies accurately and precisely and allow for safe, persistent storage, two options were considered:

- Small-Scale m-ATX Form Factor desktop system.
- Single Board Micro-Computing System.

5.1 Small Scale Desktop Server System

This option faced many obvious downsides before research even began. The main downside of course being – cost. The system would perform its duties without any performance issues, downtime concerns or heating issues. Nevertheless, a barebones system would rack up a base price tag of anywhere between €300 and €600, depending on what dedicated hardware is to be included before any other critical components are even considered. Cheaper options are available, however this comes at the risk of high failure rates and damage to other components from typical usage.

5.2 Single Board Server System

A single-board computer (SBC) is a complete computer built on a simple foundation of a single circuit board, with the necessary microprocessor(s), memory components, input/output (I/O) components and other various features required of a functional computer.

Single-board computers are typically aimed at demonstration or development systems, for educational systems, or for use as embedded computer controllers. Many types of home computer or portable computer integrate all their functions onto a single printed circuit board.

Unlike many desktop personal computer, single board computers often did not rely on expansion slots for peripheral functions or expansion. Single board computers have been built using a wide range of microprocessors. Simple designs, such as built by computer hobbyists, often use static RAM and low-cost 8 or 16 bit processors. Other types, such as blade servers, include all the memory and processor performance of a server computer in a compact space-saving format. In the next section, various options will be explored.

5.3 Deciding Factors

It was concluded early on that the system should be as cost effective as possible. But this isn't the only weighted factor in deciding which option to choose. Other factors such as cost of operation on a daily basis, space consumption, heat emission, available processing vs required processing.

6. Single Board Technologies

Due to the benefits, ranging from size, minimal power consumption and general flexibility of single board computers, this option was decided and agreed upon at a very early stage. Much research into various board developers and specifications relating to both soon followed. Two of the biggest contenders were: Intel's Galileo and Raspberry Pi. The following sections will drill down on the features and considerations for each of the boards.

6.1 Intel Galileo

Galileo is a microcontroller board based on the Intel® Quark SoC X1000 Application Processor, a 32-bit Intel Pentium-class system on a chip. It's the first board based on Intel® architecture designed to be hardware and software pin-compatible with Arduino shields designed for the Uno R3. Digital pins 0 to 13 (and the adjacent AREF and GND pins), Analog inputs 0 to 5, the power header, ICSP header, and the UART port pins (0 and 1), are all in the same locations as on the Arduino Uno R3. This is also known as the Arduino 1.0 pinout [8]. The board is illustrated below in Figure 6.a and Figure 6.b

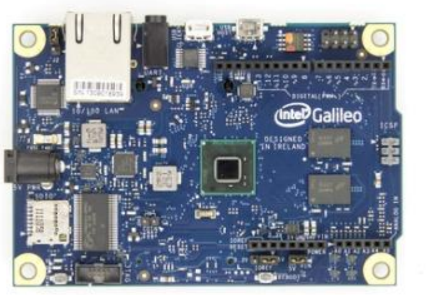


Figure 6.b - Intel Galileo (Front) [8]

Figure 6.a - Intel Galileo (Back) [8]

6.1.1 Relevant Features

One of the strongest advantages of the Galileo boards is the compatibility with the Arduino Software Environment, this allows users to become familiar with functionality and potential uses very quickly. In addition to Arduino hardware and software compatibility, the Galileo board has several PC industry standard I/O ports and features to expand native usage and capabilities beyond the Arduino shield ecosystem. A full sized mini-PCI Express slot, 100Mb Ethernet port, Micro-SD slot, RS-232 serial port, USB Host port, USB Client port, and 8MByte NOR flash come standard on the board [8].

6.1.2 Galileo Architecture

Certain important points can be considered from viewing the high level architecture of the board itself. The General Purpose Input/Output (GPIO) arrangement and availability is a factor as regardless which board is selected.

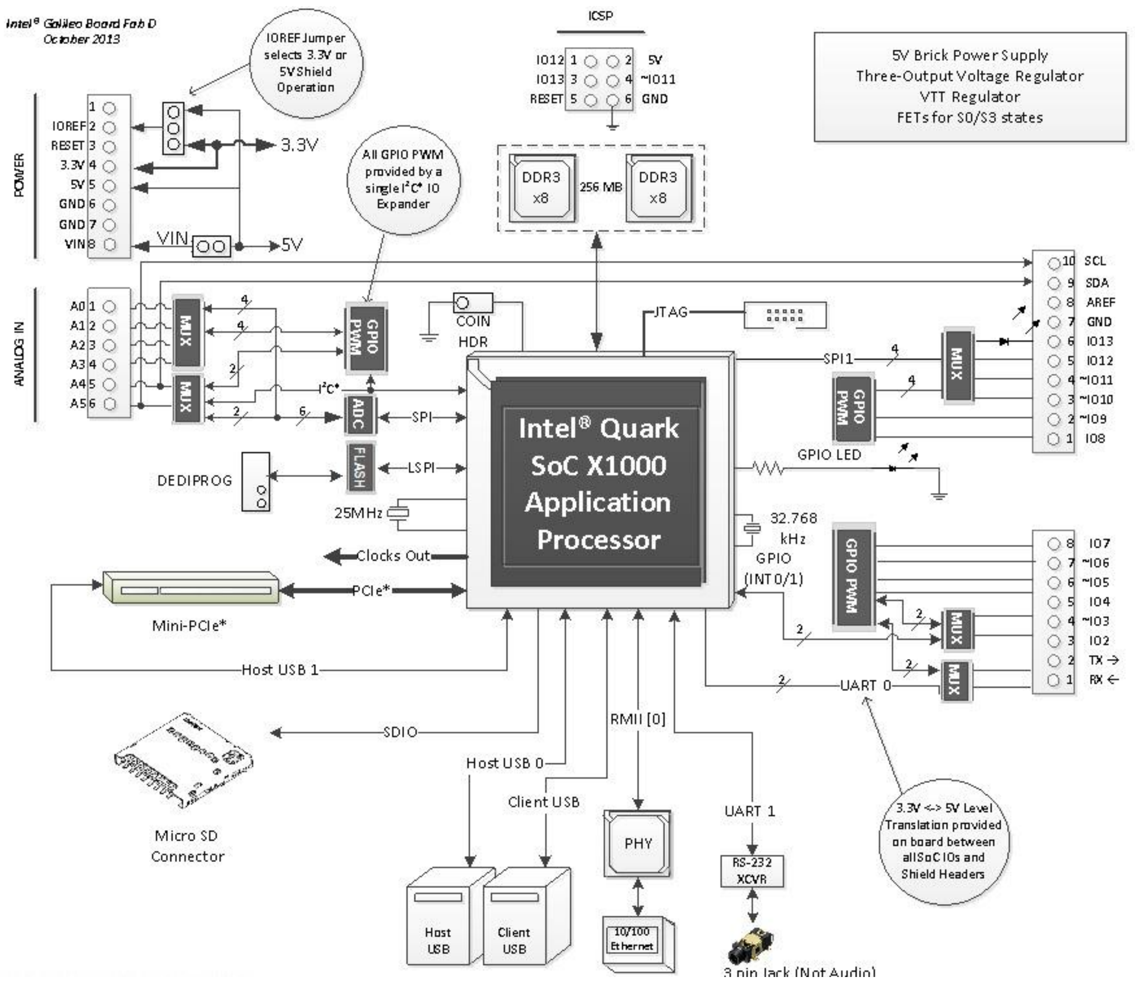


Figure 6.c - Intel Galileo Architecture

6.2 Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games [9].

The Raspberry Pi has gained much traction since its early inception and now many models are available for hobbyists, inventors, educators etc. all offering specific specifications to suit various needs.

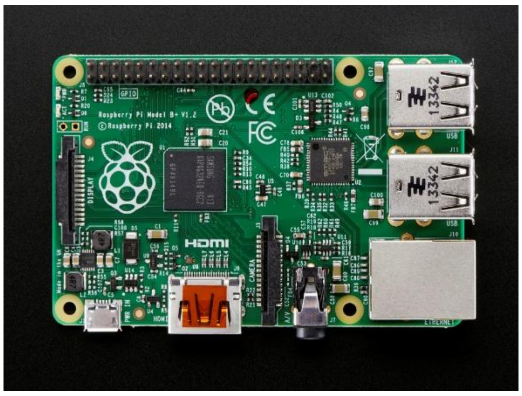


Figure 6.e - Raspberry Pi B+ (Front)

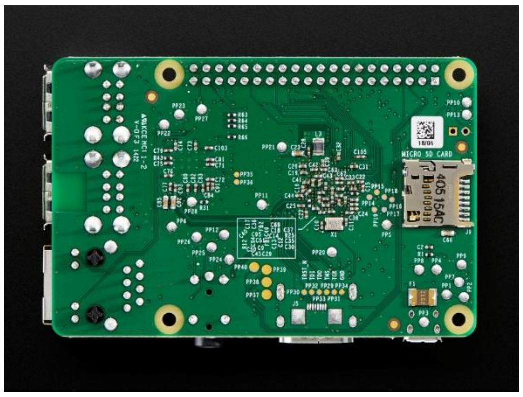


Figure 6.d - Raspberry Pi B+ (Rear)

6.2.1 Raspberry Pi Model Table

As of October, 2015, the following models are available for standard retail:

	R-Pi 1 Model A	**R-Pi 1 Model A+**	**R-Pi 1 Model B**	**R-Pi 1 Model B+**	**R-Pi 2 Model B**
Release Date:	*Feb 2012*	*Nov 2014*	*N/A*	*July 2014*	*Feb 2015*
Target price:	*US $25*	*US $20*	*US $35*	*US $25*	*US $35*

***Note**: For the sake of efficient research, the Raspberry Pi B+ model was selected due to availability and the lower price tag.*

6.2.2 Relevant Features

- More GPIO. The GPIO header has grown to 40 pins, while retaining the same pinout for the first 26 pins as the Model A and B.
- More USB. We now have 4 USB 2.0 ports, compared to 2 on the Model B, and better hot plug and overcurrent behaviour.
- Micro SD. The old friction-fit SD card socket has been replaced with a much nicer push-push micro SD version.
- Lower power consumption. By replacing linear regulators with switching ones we've reduced power consumption by between 0.5W and 1W.
- Better audio. The audio circuit incorporates a dedicated low-noise power supply.
- Neater form factor. Aligned the USB connectors with the board edge, moved composite video onto the 3.5mm jack and added four squarely-placed mounting holes.

6.2.3 Raspberry Pi Architecture *(Relative)*

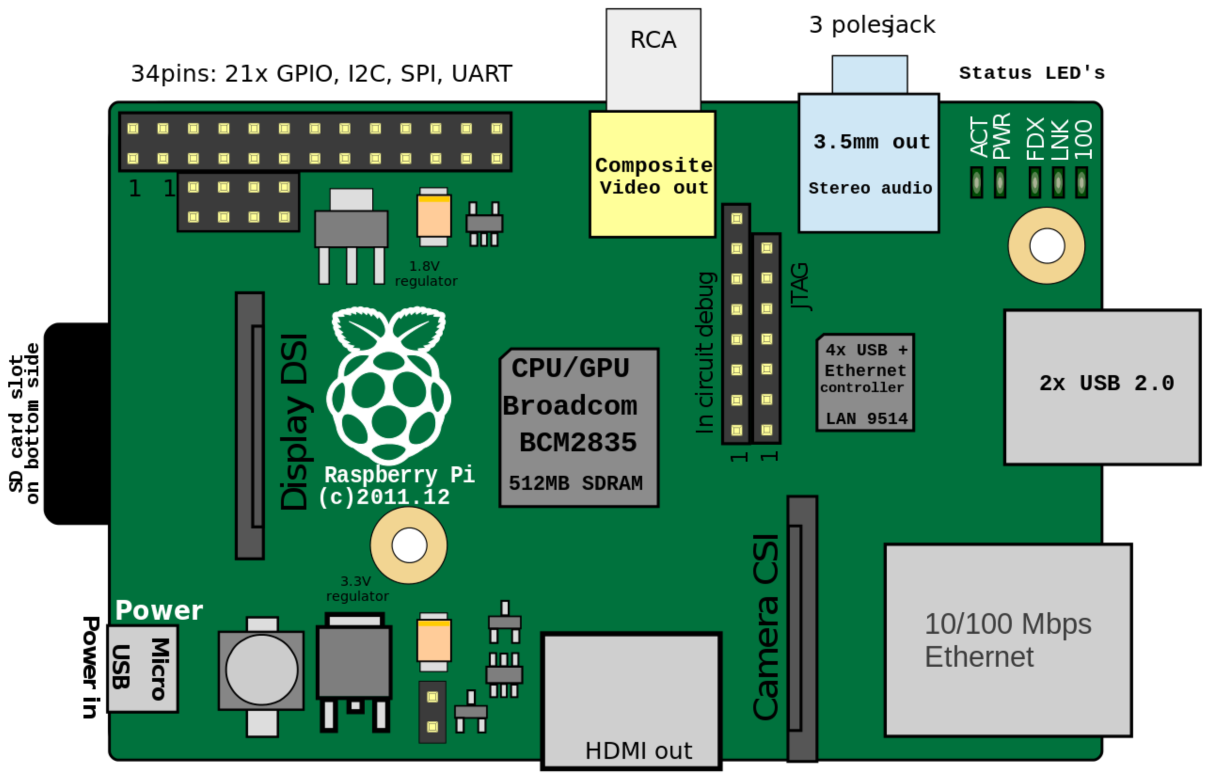


Figure 6.f - Raspberry Pi B+ Architecture

6.3 Intel Galileo vs Raspberry Pi B+

After considering all positive and negative points for both boards. It was decided that the Raspberry Pi B+ was to be the system's operating board. First however, it must be noted that the Raspberry Pi and Arduino based boards are not aimed to achieve the same results. Both boards are good at their indented roles with diverging specialities at certain points. The decision was made due to following points:

6.3.1 Cost

The Raspberry Pi (~€20) comes in at almost a third of the price of a Galileo (~€70) board and can be acquired through domestic distributors and also delivered in less than a week [10] [11].

6.3.2 Initial Setup

The Raspberry Pi can be setup and operational in less than thirty minutes providing the available requirements are on hand. This includes an SD Card for the operating system and a 5V micro USB power supply.

6.3.3 Community & Development Options

The Pi comes ready to run with a large array of various libraries built using Python that can be used for a number of exciting Pi related projects [12].

6.3.4 Recommended Choice

Initial feedback from the Pi community, academic advice and professional opinion suggest the Pi is the better choice due to familiarity with past projects and knowledge of the Pi itself. This will reduce major downtimes due to potential common errors should they occur, as knowledge exists in regards to remedying the problem.

7. Raspberry Pi Operating Systems

Raspberry comes built ready to run a selection of down sized, existing operating systems.

7.1 NOOBS

The NOOBS [13] installer allows a user to install from a pre-set list of operating systems including:

- Arch Linux ARM
- OpenELEC
- Pidora
- Puppy Linux
- Raspbmc including XBMC
- RISC OC
- Raspbian (Recommended OS for Pi 1 models)

This is not an exhaustive list however. Several forks also exist but are not available using the NOOBS installer. These forks are special purpose built for specific tasks, such as Ark OS for website and email self-hosting. Many of these forks may or may not have uses on this project. However the best potential candidates for the project are; ***Raspbian***, ***Snappy Ubuntu***, ***Pidora*** and ***Windows 10 IOT Core***.

7.1.1 Raspbian Pi

At present, initial setup, testing and familiarisation with Raspbian showed that it is suitable for the project and offers expansion options and easy upgrades to revised systems in the further. In addition to this, Raspbian is the recommended operating system for general use and the officially supported operating system by the Raspberry Pi Foundation [13], meaning optimisations can be made within the system. Two kernel versions of Raspbian are currently available:

- Raspbian Jessie (Kernel 4.1)
- Raspbian Wheezy (Kernel 3.18)

If any changes in the operating system are decided on, details on this changes will be detail herein.

7.1.2 Snappy Ubuntu Core

Snappy Ubuntu a specially optimised operating system for clouds and small devices. “*Snappy Ubuntu Core is a new rendition of Ubuntu with transactional updates - a minimal server image with the same libraries as today’s Ubuntu, but applications are provided through a simpler mechanism.*

The snappy approach is faster, more reliable, and lets us provide stronger security guarantees for apps and users — that’s why we call them “snappy” applications.” [14]

8. Raspberry Pi Projects

In this section, various Raspberry Pi projects designed and developed by the Pi community will be reviewed to further understand the potential of the Pi and how a simple single chip computer can become so much more.

8.1 Pi in the Sky

This project is an extension project for Raspberry Pi's GPIO break out pins. This add on board acts a GPS receiver and radio transmitter designed for tracking high altitude balloon flights. The board utilises an ultra-high frequency transmitter which is license-free in Europe. Together with the supplied, open-source software it embodies the experience of over 50 successful flights, about half of which have used the Raspberry Pi. The general features of the board include:

- Efficient built-in power regulator providing run time of over 20 hours from 4 AA cells
- Highly sensitive UBlox GPS receiver approved for altitudes up to 50km
- Temperature compensated, frequency agile, Radiometrix 434MHz radio transmitter
- Temperature sensor
- Battery voltage monitoring
- Sockets for external i2c devices, analogue input, external temperature sensor
- Allows use of Raspberry Pi camera
- Mounting holes and spacers for a solid connection to the Pi

Open source software also enables a wide range of additional functionality that can further utilise data recorded using the on board sensors [15].

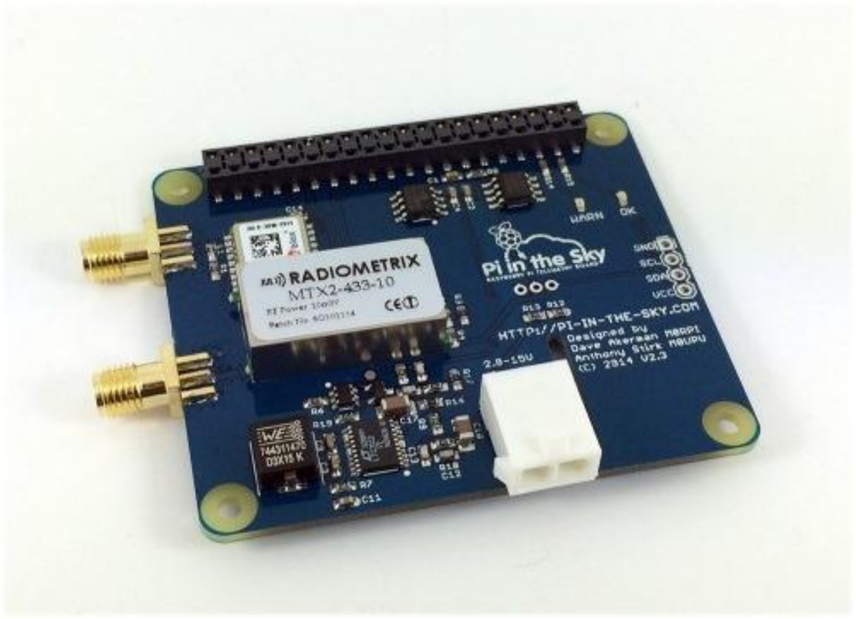


Figure 8.a - Pi in the Sky Board

8.2 Glasgow Raspberry Pi Cloud

“The University of Glasgow Raspberry Pi Cloud project is a teaching and research project focused around the construction of a “scale-model” of a warehouse-sized computer using only Raspberry Pi devices. The project is based in the School of Computing Science and is led by SICSA Research Fellows Dr David R. White and Dr Posco Tso…” [16].

The project continues to progress. The physical build is complete, and one of the students has spent months building a nice RESTful API to monitor and control the PiCloud through a web interface.

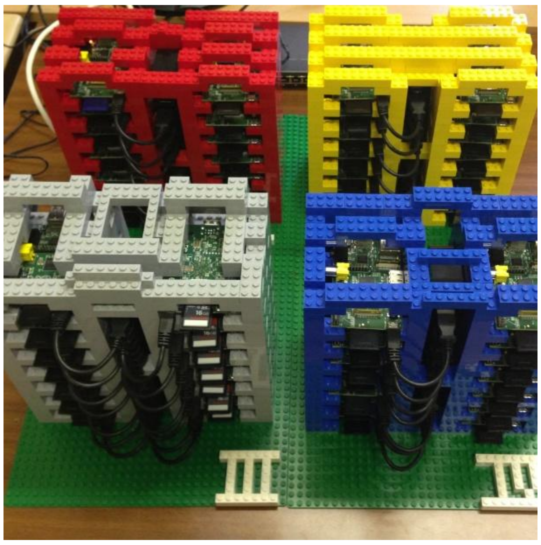


Figure 8.b - Glasgow R-Pi Cloud Housed in Lego

According to the recent updates from the project, LXC virtualisation has also been used and furthering building a software stack on top of that. Posco has also been experimenting with Hadoop and Software Defined Networking (SDN).

This project is an example that the Pi can achieve immense results, given the size and cost of materials, example shown in Figure 11.b where Lego is used as a construction bed.

8.3 Windows IoT Facial Recognition

This project is a subset of a series of ‘Internet of Things’ driven projects – Supported by Microsoft. The series is dubbed as the Hack the Home initiative, which provides makers with free, open-source components for effortless interfacing with devices and services that makers use most to up-tech their homes.

Security is a subject on the rise with projects involving boards such as the Raspberry Pi. A self-built system is not only less expensive than a bulky industry compliant installation, but it also allows for complete control and ultimate customization to suit the needs of the builder.

Microsoft’s Project Oxford allows for facial recognition applications to be more accessible to makers now, more than ever before. This project utilises a basic webcam, an internet connection and a door that unlocks itself via facial recognition.

8.4 Hobbyist Projects

In addition to the higher scale projects listed above, there exists a plethora of community driven projects that users with little to no education on microcontrollers, programming or electronics can dive into and see results immediately. Beginner packages exists for this niche collective.

9. Device Control

To achieve control of devices and lighting in the home, the system will need to be able to communicate with these devices and transmit protocol codes to control their status. This can be achieved using several different methods ranging from various complexities and prices. This section will cover a series of suitable technologies deemed suitable without in depth knowledge of electronics and advanced circuitry.

9.1 Static 433 MHz Sockets

This method of controlling devices relies entirely on being able to break a flow of electricity using an intermediate socket device with an internal switch that responds to a particular frequency, typically transmitted using a ~433.82 MHz remote control interface. 433 MHz is a pseudo industry standard used for this type of device.

The type of socket is placed between a mains outlet and the device plug itself. Using the given remote interface, a user can switch individual plugs on and off or all using the group protocol transmitter programmed onto the remotes microcontroller. Although these devices are useful, they lack the ability to automate at set times during the day and further extension requires a different remote configuration.

Figure 9.a – Energenie example socket component

9.2 433 MHz Receiver

To further understand the explicit nature of the codes being transmitted from the remotes themselves, it is necessary to read and interpret the RF transmission. For this, a simple low cost ~433 MHz receiver module is required. These units are incredibly inexpensive. They can be found for at a cost typically between €1 and €5, depending on reliability and build integrity. Signals can be caught using a variety of different methods, these will be covered at a later stage.

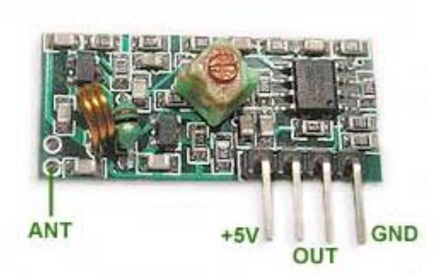


Figure 9.b - 433 MHz Receiver

9.3 433 MHz Transmitter (LPD 433MHz)

LPD433 (low power device operating at 433MHz) is an ultra-high frequency [17] band that can be operated without a license. These frequencies are in accordance with the region 1 industrial, scientific and medical (ISM) radio bands of 433.050MHz to 434.790MHz and operating is mainly limited to European Conference of Postal and Telecommunications Administration countries (CEPT) [18].

"LPD hand-held radios are authorized for license-free voice communications use in most of Europe using analog frequency modulation (FM) as part of short range device regulations, with 25 kHz channel spacing, for a total of 69 channels. In some countries, LPD devices may only be used with an integral and non-removable antenna with a maximum legal power output of 10 megawatts…"

9.3.1 Channel Distribution

Channel	Frequency (MHz)	Channel	Frequency (MHz)	Channel	Frequency (MHz)
1	433.075	**24**	433.650	**47**	434.225
2	433.100	**25**	433.675	**48**	434.250
3	433.125	**26**	433.700	**49**	434.275
4	433.150	**27**	433.725	**50**	434.300
5	433.175	**28**	433.750	**51**	434.325
6	433.200	**29**	433.775	**52**	434.350
7	433.225	**30**	433.800	**53**	434.375
8	433.250	**31**	433.825	**54**	434.400
9	433.275	**32**	433.850	**55**	434.425
10	433.300	**33**	433.875	**56**	434.450
11	433.325	**34**	433.900	**57**	434.475
12	433.350	**35**	433.925	**58**	434.500
13	433.375	**36**	433.950	**59**	434.525
14	433.400	**37**	433.975	**60**	434.550
15	433.425	**38**	434.000	**61**	434.575
16	433.450	**39**	434.025	**62**	434.600
17	433.475	**40**	434.050	**63**	434.625
18	433.500	**41**	434.075	**64**	434.650
19	433.525	**42**	434.100	**65**	434.675
20	433.550	**43**	434.125	**66**	434.700
21	433.575	**44**	434.150	**67**	434.725
22	433.600	**45**	434.175	**68**	434.750
23	433.625	**46**	434.200	**69**	434.775

9.3.2 Usage Proposal

This device allows for signals within the 433 MHz range to be transmitted. Like the receiver module, they can placed onto a breadboard for testing and work incredibly well with micro computing devices such as the Raspberry Pi and Arduino. Typically the receiver and transmitter work in pairs to allow for two way communication. These devices are also quite low in cost however and as a result are indiscriminate and will receive a fair amount of noise due to their simplicity and low cost construction.

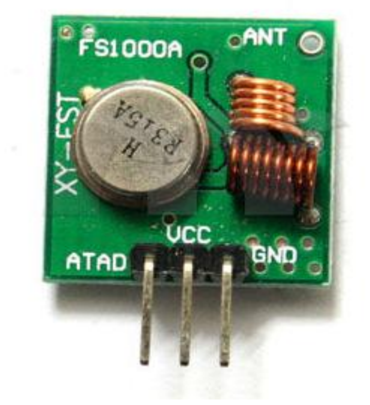


Figure 9.c - 433 MHz Radio Transmitter Module

The project will rely more on the transmission of codes than the interception of codes from source remotes. Subsequently, attention must be given to how the system will deal with the embedded codes retrieved from source devices. With a transmitter such as this, it's possible to communicate with and directly control the sockets mentioned above once the signals have been captured.

9.3.3 Additional Information

9.3.3.1 Surface Acoustic Wave

A surface acoustic wave (SAW) is an acoustic wave travelling through a medium through which is propagates. In the context of this project, air is the medium through which the oscillated sound waves propagate. SAW's were first explained by John William Strutt, also known as the 3rd Baron Lord Rayleigh in 1885 [19]. Named so after the discoverer, Rayleigh waves have a longitudinal and vertical shear component than can couple with any media in contact with a surface.

SAW resonators are used widely around the globe. This oscillation technique allows for the production of signals in the radio spectrum range of about 100 kHz to 100 GHz [20].

9.3.4 Mechanical Relay Switches

These small devices operate as electromagnetic breaker units that either complete or interrupt a current based on the state of the latching switch built into the housing. The relays are intended to interrupt high voltage lines with low voltage power driving them. They are relatively simple devices and provide a solution for handling mains lighting switching and a slightly more risky and technical solution for handling appliances. The proposed usage of relays should be carefully monitored throughout the project as work with them is generally quite dangerous as they are in contact with live mains supply during the operation.

9.4 X10 (Industry Standard) Experimental

X10 is pioneering protocol for communication among electronics devices, primarily used for home automation. To achieve communications, X10 uses power line wiring for signalling and subsequent control, brief radio frequency bursts representing digital media are transmitted. It was first introduced in 1975 and has seen decades of use from home automation enthusiasts all over the world. X10 can be used to control lamps, appliances, wall switches and can be used to monitor doors, windows and motion.

X10 for the time being is still a developing concept and somewhat expensive compared to the cheaper alternatives. Much difficult and risky work on integrating modules with mains power lines would be required and this would violate many safety standards and also conflict with conditions imposed by landlords for stakeholders that are not homeowners. Especially dangerous without proper qualifications to tamper with such high current lines.

9.5 Z-WAVE

Zwave (or Z wave or Z-wave) is a protocol for communication among devices used for home automation. It uses RF for signalling and control. Zwave was developed by Zensys, Inc. a start-up company based in Denmark.

Z-wave is the leading smart home technology that is inside everyday products like lights, locks, thermostats and much more. These products work together to create the backbone of a smart home and enable you to use your phone or tablet to create one-touch scenes that help with daily activities like: saving energy, keeping your home secure, looking after your loved ones and being more comfortable.

Zwave operates at 908.42 MHz in the US (868.42 MHz in Europe) using a mesh networking topology. A Zwave network can contain up to 232 nodes (estimate), although reports exist of trouble with networks containing over 30-40 nodes. Zwave operates using a number of profiles (think of them like languages), but the manufacturer claims they interoperate. The caveat is that care must be observed when selecting products as some products from certain manufacturers are not compatible with other manufacturers' products. Zwave utilizes GFSK modulation and Manchester channel encoding.

A central, bridging network controller, device is required to setup and manage a Zwave control network. Each product in the home must be “included” into this Zwave network before it can be manipulated via the Zwave control interface (and before it can assist in repeating/hoping within the mesh network). Each Z-Wave network is identified by a Network ID and each device is further identified by a Node ID [21].

The Network ID is the ubiquitous identification of all nodes belonging to any single logical Z-Wave network. Network ID has a length of 4 logical bytes and is assigned to each device by the primary controller when the device is added into the network.

The Node ID is the address of the device / node existing within network. The Node ID has a length of 1 byte.

10. Radio Wave Analysis

During this phase of research, a practical and primary form of research was conducted on the output waves of a typical 433 MHZ PT22XX series encoding microcontrollers in attempt to develop a module to replicate any 433MHZ frequency wave. For this, a combination of software based binary logic analysis and waveform analysis was required. This portion of research is the determining factor for the success of the project and provides the largest of key deliverables.

This was accomplished using Ondřej Staněk's open source IR Protocol Analyser. This software is a universal application for monitor and bit level decoding of several types of IR and radio wave packets. It can be used via microphone ports on any laptop or desktop system and signals received using any receiver capable of capturing radio or infrared waves [22]. For this application, the aforementioned 433 MHz receiver was used as a primary test device.

In addition to this, Reaper, a digital audio workstation was used to view the waveform and compare replicated signals against the original remote signals. REAPER is a complete digital audio production application for Windows and OS X, offering a full multitrack audio and MIDI recording, editing, processing, mixing and mastering toolset [23].

This software package was used mainly to ensure integrity of the replicated waveform and ensure data amplification and waveform peaks were maintained. Unsuccessful transmissions would occur if any of the waveform attributes were incorrect.

The following sections will cover the low level concepts of radio communications and formats relative to this project.

10.1 Manchester Codes

In data transmission, Manchester encoding is a form of digital encoding in which data bits are represented by transitions from one logical state to the other. This is different from the more common method of encoding, in which a bit is represented by either a high state such as +5 volts or a low state such as 0 volts.

When the Manchester code is used, the length of each data bit is set by default. This makes the signal self-clocking. The state of a bit is determined according to the direction of the transition. In some systems, the transition from low to high represents logic 1, and the transition from high to low represents logic 0. In other systems, it is reversed.

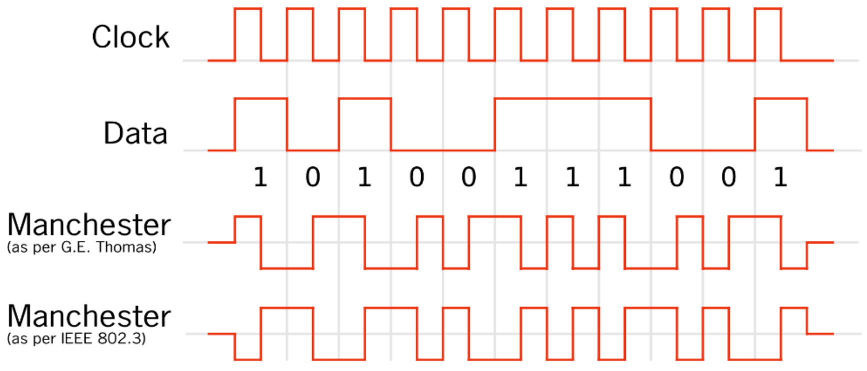


Figure 6 - Manchester Code Output Illustrated

10.2 Outputting Codes

To output any radio code, the Raspberry Pi must be used with a middleware library. For this project, there are several varieties of libraries that operate differently and offer support for GPIO control.

10.2.1 PIGPIO

PIGPIO (Pi General Purpose Input Output) is a library for the Raspberry which allows control of the General Purpose Input Outputs (GPIO). The library works on all versions of the Pi.

PIGPIO is written in C but may be used by other languages. In particular the PIGPIO daemon offers a socket and pipe interface to the underlying library [24]. Some of the key features of this library are as follows:

- Sampling and time-stamping of GPIO 0-31 between 100,000 and 1,000,000 times per second.
- Provision of PWM on any number of the user GPIO simultaneously.
- Provision of servo pulses on any number of the user GPIO simultaneously.
- Callbacks when any of GPIO 0-31 change state (callbacks receive the time of the event accurate to a few microseconds).
- Notifications via pipe when any of GPIO 0-31 change state.
- Callbacks at timed intervals.
- Reading/writing all of the GPIO in a bank (0-31, 32-53) as a single operation.
- Individually setting GPIO modes, reading and writing.
- Socket and pipe interfaces for the bulk of the functionality in addition to the underlying C library calls.
- The construction of arbitrary waveforms to give precise timing of output GPIO level changes (accurate to a few microseconds).
- Software serial links using any user GPIO.
- Rudimentary permission control through the socket and pipe interfaces so users can be prevented from "updating" inappropriate GPIO.
- Creating and running scripts on the PIGPIO daemon.

10.2.2 RPi.GPIO

RPi.GPIO is a Python module for controlling the GPIO on a Raspberry Pi. It was discovered mid-way through iteration one however that this module is unsuitable for real-time or timing critical applications. This is largely due to the fact that it is difficult to predict when Python will be busy garbage collecting and how Python deals with the Raspberry Pi’s own process management, as it runs directly below the Linux kernel. As Linux is a multitasking operating system, priority may be given to another process at any given time. A code output with any interrupt will result in a broken bit sequence and the Manchester code transmission will be invalid.

Arduino offers better real-time solutions for this purpose, however, the aforementioned PIGPIO retains a better control over process control as discovered on the reworking on the PiLYNK transmitter module. However, the module can be used for other sensors.

11. Sensors

Using sensors with the Raspberry Pi system will allow for measuring of real world data objects. The scope of this project will encompass motion and temperature and provide functionality relating to the data captured using these sensing mediums. Two main sensor types will be discussed in this section, ancillary internal sensors such as the Raspberry Pi's integrated CPU sensors will be discussed in the design documents.

11.1 PIR Motion Sensor

Pyroelectric (or Passive) Infrared Sensors (PIR) are sensors that measure infrared light radiating from objects in its field of view or sensor range [25]. They are implemented in many existing technologies for detecting motion and range from basic modules to more advanced, long range, high sensitivity models, typically used for military applications, avionics, domestic and commercial security.

This project will use a simplistic model using a BISS0001 chip, RE200B [26] sensing element and NL11NH [27] lens. The following image is an illustration of the unit researched for the project:

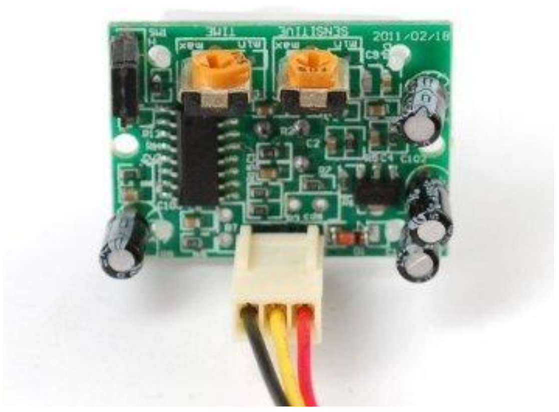


Figure 18 - PIR Sensor Rear-side

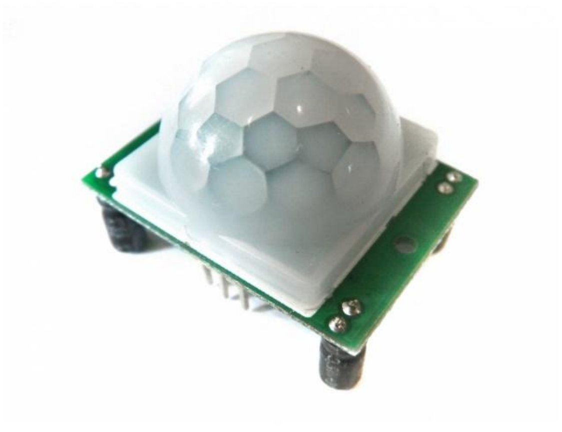


Figure 17- PIR Sensor Front-side

11.1.1 Understanding Radiation Detection Concept

The PIR sensor itself has two slots, each slot is made of a sensitive material that is particularly sensitive to infrared radiation (the same heat emitted from the human body). In its idle state, the PIR sensor both sensor slots can receive a view of the field of vision in its immediate front-area. Both slots detect the same amount of IR, this is typically the ambient radiation from the room or walls or outdoors. When a mass of warmth passes the sensor, it first triggers the first receptacle of the module, this causes a positive differential change between the two halves of the sensing pair. When the body mass exists the view of the sensing area, the module generates a negative differential charge [25].

These pulses of high low voltages are what can be detected using the Raspberry Pi's GPIO pins.

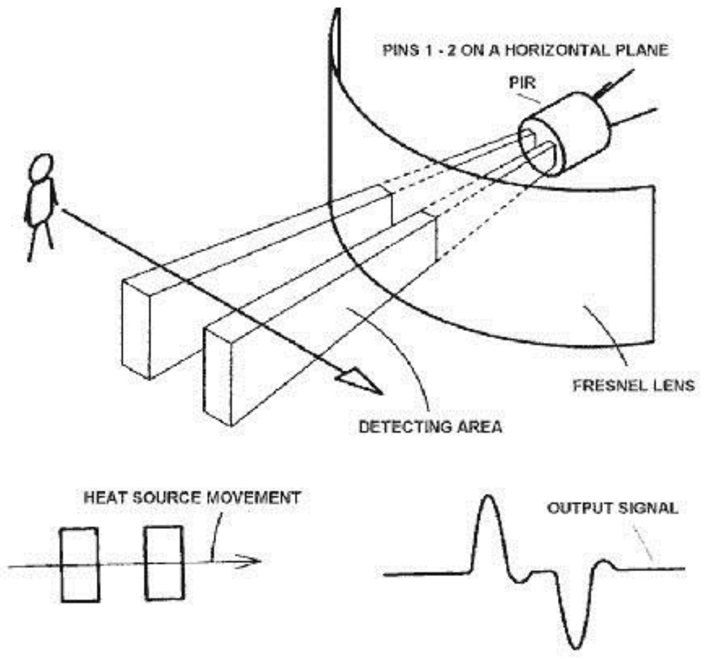


Figure 19 - PIR Detection Illustration [25]

As illustrated in the image [right], these devices exhibit high sensitivity and reliable performance made possible by Murata's ceramic technology and packaging technology developed over many years. IR(X) series sensors realise cost benefits and higher performance with a new improved material of infrared ceramics throughout the range [28]. As the cost of these devices is relatively low, using a wide array of these devices in a home automated security system is possible at a relatively low price.

11.1.2 The Sensor

The sensor itself is housed in a hermetically sealed metal containment unit to improve noise reduction and temperature/humidity immunity. On this unit is a window made of IR-transmissive material that is typically made of coated silicon, this protects the sensing element from harmful elements. Behind this small sensing window are the two aforementioned sensing elements.

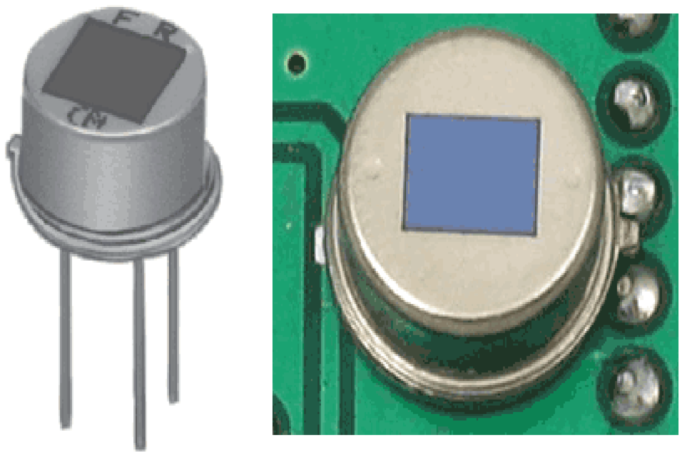


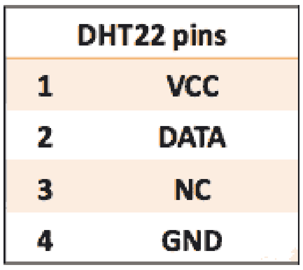
Figure 20 - Hermetically Sealed Sensor

11.2 Temperature Sensor

11.2.1 DHT22 AM2302

The DHT22 is a basic reading, low-cost digital temperature and air humidity percentage sensor. It uses capacitive humidity sensing and thermistor to measure the surrounding air. The sensor can then be read using the GPIO on the Raspberry Pi by reading the output data from the sensor on the sensors digital output pin (assuming power is flowing also). The sensors are relatively easy to use, however, it can be difficult to get the timings for readings exact [29].

The recommended approach is to connect the first pin on the left to a charged pin of between 3-5volts of power, the second pin is the data pin, this should be connected to one of the Raspberry Pi's GPIO data pins, and finally the ground pin (far-right) should be connected to the Raspberry Pi's ground pin array. Adafruit has written a module available for Python to read the DHT, however, for the sake of completeness, a new module will be written especially for this purpose.



DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

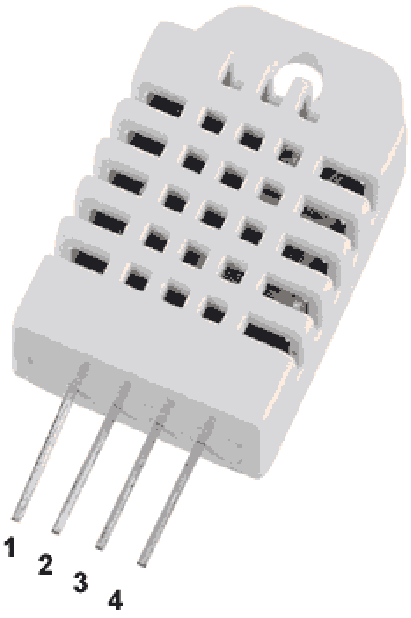


Figure 11 - DHT22, Including Pin Reference Datasheet [29]

11.2.2 DHT22 Technical Details

The following is a list of technical details sourced from Adafruit [29].

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
- 4 pins, 0.1" spacing
- Weight (just the DHT22): 2.4g

12. Client Side (Front End) Technologies

Client side or ‘Front End’ technologies refer to the viewable interface and standard client control model for a client-server system. Typically, client defines the web application and what a user will see when using the system. Actions such as inputting form information, credential validation, user and accounts details are all examples for typical client side behaviour.

In this particular system, the aim of the front end development was simple – provide a simple responsive UI that technocrats and technophobes alike can appreciate. Distributing a poorly designed UI for a system can undermine the concrete developments elsewhere and turn users away from repeat usage. To achieve this, a particular set of industry standard web application development standards and technologies are employed.

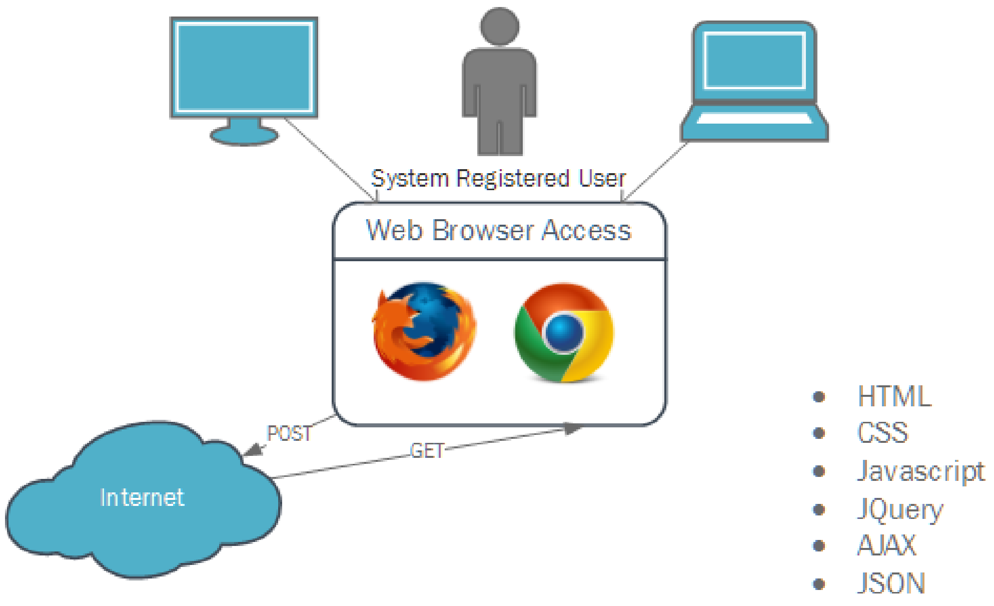


Figure 12.a Client Side Illustration

12.1 HTML

12.1.1 Description

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. HTML is the cornerstone technology bridging CSS and Javascript. It is used by most websites to create visually appealing and engaging webpages, UI’s and large scale web applications and recently a push for increased mobile web applications with the aid of bootstrapping for mobile friendly development.

HTML has been used since 1990. It is a product of the complication technical specification of markup languages known as the SGML (Standard Generalised Markup Language). Since its inception and wide scale roll out, HTML has been experienced a rapid revision history and subsequent has evolved at an increasing pace, at least compared to other technologies that exists today. On the 28th of October 2014, the World Wide Web Consortium (W3C) finalised and published HTML5

12.1.2 Structure

HTML documents are divided into three principal levels inherited from the HTML document object model (DOM):

- *<HTML>*
  - *<HEAD>*
  - *<TITLE>*
  - *<BODY>*
  -

Different artefacts are identified, retrieved and or displayed at each level. Typically at the <HEAD>, the web pages identifying information such as the websites main title and various important attributes. When a user request a page, they are presented with the pages title, which appears in the browsers tab, this acts a key identification for sites in bookmarks as all other information is hidden. <BODY> represents the bulk content of the web page such as graphics, text, forms etc. A full index of tags can be found here.

12.1.3 Example Usage

```
<!doctype html>

<html lang="en">
<head>
  <meta charset="utf-8">

  <title>A HTML Title</title>
  <meta name="description" content="A HTML Webpage">
  <meta name="author" content="site">

  <link rel="stylesheet" href="css/styles.css?v=1.0">

  <!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>

<body>
  <script src="js/scripts.js"></script>
</body>
</html>
```

12.2 CSS

12.2.1 Definition

CSS (Cascading Style Sheets) is a style language that defines the layout of HTML documents. CSS covers fronts, colours, margin, lines, heights, width, background, images, advanced positions and a wide array of other attributes. As HTML itself is very limited in its ways of altering its own aesthetics, thus CSS has become a critical component in displaying beautified web pages.

CSS first appeared in 1996 as a product of the W3C and was titled CSS1. CSS is currently on major version 3 and is titled CSS3 upon its release in 1999. The schema, or plans for CSS are divided into modules and the current version has forty of these units.

Initially, browsers did not support the standard or only supported limited parts. Even today, some browsers do not support style sheets the same and this cause bugs and interoperability issues when designing web pages. Programmers have developed work-arounds and hacks to support different browsers. Once browsers started accepting style sheets and presenting them as designed, further barriers to the implementation of CSS emerged. Many programmers and designers were reluctant to use the extended formatting abilities available. Progress to a full roll solution out was slowed. However, it is now considered the staple of web design. A full CSS reference index can be found here.

12.2.2 Advantages of CSS

- Control layout of many documents from a single style sheet.
- More precise control of layout
- Apply different layout to different media-types, such as mobile vs desktop.
- Numerous advanced and sophisticated techniques.
- Content CSS caching for quicker re-retrieval of webpages.

12.2.3 Disadvantages of CSS

- CSS works different on different browsers. Internet Explorer and Opera supports CSS as different logic.
- Browsers support proprietary tags.

12.2.4 Example Usage

```
/* Button Design CSS */
.onoffswitch {
   position: relative;
   width: 95px;
   -webkit-user-select:none; -moz-user-select:none; -ms-user-select: none;
}
.onoffswitch-checkbox {
   display: none;
}
.onoffswitch-label {
   display: block; overflow: hidden; cursor: pointer;
   border: 2px solid #000; border-radius: 1px;
}
```

12.3 Javascript

12.3.1 Definition

JavaScript (JS), not to be confused with Java, was created in 10 days in May 1995 by Brendan Eich. Originally named Mocha, it was renamed LiveScript and then renamed again to the current JS upon receiving a trademark license from Sun. The initial motivation for creating the language was to build a client-side language for Netscape. JS is not a programming language in a strict sense. Instead, it is a lightweight scripting language because it uses the browser to do the dirty work. The language, much like Java is a descendant of C and C++, but the languages have splintered quite significantly.

Today, JS has conquered the web. Party due to the fact that Java has suffered at the hands of internecine and highly publicized feuds between Sun and pretty much everyone. VBScript, Microsoft's answer to JS, is being abandoned by the company, but the compiled C# can't be expected to fill the ever-needed role of a scripting language.

12.3.2 Advantages of JavaScript

- **JS is executed on the client side** – The code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server.
- **JS is a relatively easy language** – As a language, it is relatively easy to learn and comprises of syntax that is high level (close to English). It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it a breeze to develop a script to solve a one for one problem.
- **Extended functionality to web pages** – Although initially JS was very difficult to extend and almost impossible at one stage, nowadays third party add-ons like Greasemonkey enable JS developers to extend functionality.

12.3.3 Disadvantages of JavaScript

- **Security issues** – JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system. While restrictions are set in place by modern web standards on browsers, malicious code can still be executed complying with the restrictions set.
- **JavaScript rendering varies** – Different layout engines may render JS different resulting in inconsistency in terms of functionality and interface.

12.3.4 Example Usage

```
function sumDigits(num)
{
    var i, sum = 0; // can declare two variables at once
    for (i = 1; i <= num; i++)
    {
      sum += i; // add each number to sum (ie, 1 + 2 + ...+ num)
    }
    // Display result
    alert("The sum of the digits from 1 to "+ num + " is:\n\n\t " + sum);
}
```

13. Additional Front End Technologies

Included in this section are technologies that are not exclusively noted as individual technologies, rather community implementations, extensions and libraries in respect of technologies that already exist and on which they are built upon.

13.1 JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. JQuery is the most popular JavaScript library in use today, with installation on 65% of the top 10 million highest-trafficked sites on the Web [30]. JQuery is free, open-source software licensed under the MIT License.

13.1.1 Advantages of JQuery

- **Ease of use** – This is the main advantage of JQuery, it is a lot easier to use compared to standard JS and other JS libraries. Apart from simple syntax, it also requires much less code to achieve the same feature in comparison.
- **Large library** – JQuery enables you to perform a plethora of functions in comparison to other JS libraries.
- **Strong Open Source community** – JQuery, while relatively new, has a strong following that religiously devote their time to develop and enhance the functionality of JQuery. Thus there are hundreds of prewritten plugins available for download of content delivery linking to instantly speed up the development process.
- **Great documentation and API**
- **AJAX support** – JQuery allows for the development of AJAX templates with ease, AJAX enables a sleeker interface where actions can be performed on pages without requiring the entire page to be reloaded.

13.1.2 Disadvantages of JQuery

- **Functionality maybe limited** - While JQuery has an impressive library in terms of quantity, depending on how much customization you require on your website, functionality maybe limited thus using raw JS may be inevitable in some cases.
- **JQuery JS file required** - The JQuery JS file is required to run JQuery commands, while the size of this file is relatively small (25-100KB depending on server), it is still a strain on the client computer and maybe your web server as well if you intend to host the JQuery script on your own web server.

13.1.3 Example Usage

```
……………………
   <a href="http://jquery.com/">jQuery</a>
   <script src="jquery.js"></script>
   <script>

   $( document ).ready(function() {
      $( "a" ).click(function( event ) {
         alert( "Prompt the user with a browser message" );
         event.preventDefault();
      });
   });

   </script>
………………………
```

```
$(document).ready(function() {
   $("a").click(function( event ) {
      alert("Thanks for visiting!");
   });
});
```

13.2 JQuery UI

JQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

13.3 Bootstrap

Bootstrap is a free and open-source collection of tools for creating websites and website applications. Bootstrap is the most popular HTML, CSS and JS framework for developing responsive, mobile first projects on the web [31]. Bootstrap was original created by a designer and a developer at Twitter. Prior to being an open-soured framework, Bootstrap was known as Twitter Blueprint.

Bootstrap was a critical selection choice on this project as a decision had to be made between mobile-application versus a ubiquitous web application. Bootstrap allowed for the latter to be developed and cater for desktop, tablet and mobile platforms all in one. Bootstrap contains features that can be largely utilised on this project such as:

Pre-processors: Bootstrap ships with vanilla CSS, but its source code utilizes the two most popular CSS pre-processors, Less and Sass. Quickly get started with precompiled CSS or build on the source.

One Framework, Every Device: Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.

Fully Featured: With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

13.3.1 Advantages of Bootstrap

- Works well in modern browsers.
- Mobile first approach.
- Normalises many CSS issues.
- Lightweight.
- Largely customisable.
- Encourages use of LESS CSS.
- Gives consistency.
- Easy to implement.
- Comes preloaded with JQuery plugins.
- Helps aesthetically-challenged developers.

13.3.2 Disadvantages of Bootstrap

- Weak when it comes to complex data entry screens. Of course, screens should be made simple for users, but that doesn't mean a form might not be technically complex. Sometimes, users are well-trained and *want* complex forms so that they can rapidly do work.
- JQuery plugins are limited. Certainly, you can integrate your own, find other 3rd party controls, etc. But out of the box, you are missing the heavy-hitters, like grids, tree views, drag and drop, etc.
- Naming conventions and semantics are mediocre. Example: using the <i> tag for icons. Class names like "pull-right".
- It's very customizable, but inevitably many sites start looking alike (just like WordPress—consider how many blog sites look the same).

Finally, Bootstrap has a common major pro **and** con: **it's someone else's code.** This means that if Bootstrap does not perform exactly as you wish, it is necessary to discover how to do so. This involves stripping back the code base and deviating from the established code stream.

13.3.3 Example Usage (HTML)

Note: Use of COL-MD-X Bootstrap classes for grid spacing

```
<div class="container">
   <!-- Example row of columns -->
   <div class="row">
     <div class="col-md-4">
       <h2>Heading</h2>
       <p>CONTENT</p>
       <p><a class="btn btn-secondary" href="#" role="button">View details &raquo;</a></p>
     </div>
     <div class="col-md-4">
       <h2>Heading</h2>
       <p>CONTENTp>
       <p><a class="btn btn-secondary" href="#" role="button">View details &raquo;</a></p>
     </div>
   </div>
</div>
```

13.4 AJAX

Ajax is not a programming language or a tool, but a concept. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. The best definition I've read for Ajax is "the method of exchanging data with a server, and updating parts of a web page - without reloading the entire page." Ajax itself is mostly a generic term for various JavaScript techniques used to connect to a web server dynamically without necessarily loading multiple pages. In a more narrowly-defined sense, it refers to the use of XmlHttpRequest objects to interact with a web server dynamically via JavaScript.

13.4.1 Benefits

Callbacks

Ajax is used to perform a callback, making a quick round trip to and from the server to retrieve and/or save data without posting the entire page back to the server. By not performing a full postback and sending all form data to the server, network utilization is minimized and quicker operations occur. In sites and locations with restricted bandwidth, this can greatly improve network performance. Most of the time, the data being sent to and from the server is minimal. By using callbacks, the server is not required to process all form elements. By sending only the necessary data, there is limited processing on the server. There is no need to process all form elements, process the ViewState, send images back to the client, or send a full page back to the client.

Making Asynchronous Calls

Ajax allows you to make asynchronous calls to a web server. This allows the client browser to avoid waiting for all data to arrive before allowing the user to act once more.

User-Friendly

Because a page post back is being eliminated, Ajax enabled applications will always be more responsive, faster and more user-friendly.

Increased Speed

The main purpose of Ajax is to improve the speed, performance and usability of a web application. A great example of Ajax is the movie rating feature on Netflix. The user rates a movie and their personal rating for that movie will be saved to their database without waiting for the page to refresh or reload. These movie ratings are being saved to their database without posting the entire page back to the server.

13.5 ChartJS

ChartJS is a beautiful open source front end graphing solution offered by community developer Nick Downie. The charts are developed with responsiveness in mind and resize for different viewport dimensions for maximum platform coverage. The library is modular offering a wide array of different chart types and can be programmed to be interactive on touch for improved user feedback.

13.6 CanvasJS

HTML5 JavaScript Charting Library with a simple API and 10x better performance. Charts are responsive & can run across devices including iPhone, Android, Desktops, etc. [32].

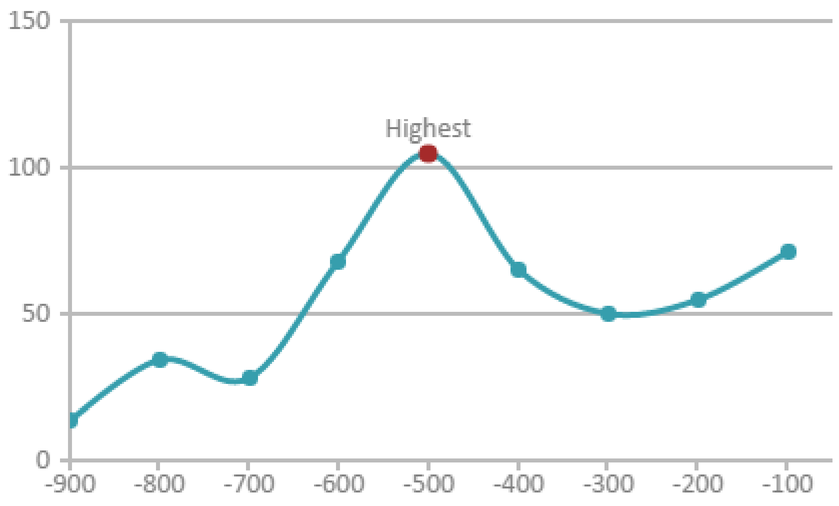


Figure 2 - Sample Usage Output of CanvasJS [32]

13.7 Jinja2 Templating Engine

Jinja2 is a modern and designer friendly templating language purpose built for Python extension. The engine is modelled heavily after Django's templates. It is fast, widely used and secure with optional sandboxed template execution environment [33].

Sample usage of the Jinja2 engine that greatly reduces the amount of front end markup required for well-formed web pages.

```
<title>{% block title %}{% endblock %}</title>
<ul>
{% for user in users %}
  <li><a href="{{ user.url }}">{{ user.username }}</a></li>
{% endfor %}
</ul>
```

14. Server Technologies

Server or 'Backend' refers to the lower layer of a web platforms architecture. This essentially means that it is the base from which client connections are served. Typically, a server is simply a program that runs on a machine purpose built for that usage, or specially configured, much like a Raspberry Pi server. The server acts as a control point between information available on lower layer that is typically a database (Mongo, MySQL, SQL or even text files). It is a safe approach as it forces client side manipulation of data to be only carried out by the instruction on the server. This allows for the continuity of data integrity and safety.

For this project, the server represents the heaviest lifting in the form of processing data, interpreting user data, manipulation of contextual data and in/out communication with underlying data stores.

Whilst the decision to proceed with Python was made at a very early stage, consideration was given to PHP as it is a mainstay of many successful web architectures used on a regular basis today. For example, Facebook runs a workhorse PHP architecture on its backend.

14.1 Python

Python is a dynamically-typed, high level, general-purpose, interpreted programming language. The core design philosophy of Python is to emphasis code readability. Due to this, programmers are able to expresses instructions in far less code than C based languages such as Java or C++. As stated:

"Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open." [34].

14.1.1 Advantages of Python

Python is widely used and its community documentation is outstanding, meaning that any reference to any API is quick and painless. Python web server programs can be distributed quickly compared to PHP and scalability of larger projects is considerably manageable.

Python has a large range of integrated frameworks that offer specialised support for specialised tasks. An example of this type of framework is the Flask micro-framework which will be discussed later. Django is perhaps the most popular of this type of framework and it is widely used for large scale web based applications.

Python can also be easily used through the WSGI (Web Server Gateway Interface). The WSGI is a specification that describes how a web server communicates with web applications and how web applications can be chained together to process requests. An example usage scenario would be running a Flask web application via an Apache web server using the Werkzeug-Serving platform.

14.1.2 Illustration of Usage

Because of the simplistic syntax of Python, code is very easily understood by humans. The following code sample illustrates the Python typography.

```
# Program to convert kilometers into miles
# Input is provided by the user in kilometers

# take input from the user
kilometers = float(input('How many kilometers?: '))

# conversion factor
conv_fac = 0.621371

# calculate miles
miles = kilometers * conv_fac
print('%0.3f kilometers is equal to %0.3f miles' %(kilometers,miles))
```

14.2 Python WiringPi

In order for Python to control the GPIO of the Raspberry Pi using the libraries mentioned under the radio control section and sensor arrow, the GPIO control interface must first be installed. This comes in the form of WiringPi.

WiringPi is a GPIO access library written in C for the BCM2835 used in the Raspberry Pi. It's released under the GNU LGPLv3 license and is usable from C and C++ and many other languages with suitable wrappers. It is designed to be familiar to people who have used the Arduino wiring system.

The original Raspberry Pi Model A and B version B1 was a $35 single board computer with a 26-pin General Purpose Input/Output (GPIO) connector and this carries a set of signals and buses. There are 8 general purpose digital I/O pins – these can be programmed as either digital outputs or inputs. One of these pins can be designated for hardware PWM output too. Additionally there is a 2-wire I2C interface and a 4-wire SPI interface (with a 2nd select line, making it 5 pins in total) and the serial UART with a further 2 pins. The following reference chart shows the GPIO pin layout in contrast to the BCM GPIO layout.

P1: The Main GPIO connector							
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin
		3.3v	1	2	5v		
8	Rv1:0 - Rv2:2	SDA	3	4	5v		
9	Rv1:1 - Rv2:3	SCL	5	6	0v		
7	4	GPIO7	7	8	TxD	14	15
		0v	9	10	RxD	15	16
0	17	GPIO0	11	12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v		
3	22	GPIO3	15	16	GPIO4	23	4
		3.3v	17	18	GPIO5	24	5
12	10	MOSI	19	20	0v		
13	9	MISO	21	22	GPIO6	25	6
14	11	SCLK	23	24	CE0	8	10
		0v	25	26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin

Figure 3 - WiringPi Layout on Raspberry Pi A

14.3 PHP Hypertext Processor

PHP is a server-side scripting language designed for web development. However it can and is also used as a general-purpose programming language. The PHP language admits an imperative, functional, object-oriented, procedural and reflective paradigm. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now has been changed to the recursive backronym PHP: Hypertext Preprocessor.

14.3.1 Advantages of PHP

PHP is widely known for being able to achieve anything (in a software context). Some of its key features are as follows:

- **Server-side scripting.** This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming. See the installation instructions section for more information.
- **Command line scripting.** You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks. See the section about Command line usage of PHP for more information.
- **Writing desktop applications.** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution. If you are interested in PHP-GTK, visit » its own website.

PHP can be used on all major operating systems, including Linux, many UNIX variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others [35].

An interesting fact to note about PHP is that it was initially developed without a functional specification [36].

14.3.2 Illustration of Usage

```php
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . sum(5, 10) . "<br>";
echo "7 + 13 = " . sum(7, 13) . "<br>";
echo "2 + 4 = " . sum(2, 4);
?>
```

15. Database Technologies

A database or data store, is an organised collection of data, in the context of relational databases is a collection of schemas, tables, queries, reports and related view objects.

In computing, databases are sometimes classified according to their organizational approach. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. Tabular database construction maintains data integrity, this essentially means that the data is always formed from a particular model. This model is defined through the conceptual, logical and physical data models.

The two main types of databases explored during this projects research is relational databases and object-oriented databases.

15.1 MySQL (Relational Choice)

MySQL is the world's most popular open source database [37]. It is currently maintained by Oracle enterprises. It is suited for high performance, scalable database applications. Some of the key points to note about MySQL:

15.1.1 Advantages of MySQL

- **Easy to Use** – MySQL is very easy to install and setup for first time users and existing users alike. Additionally, thanks to a bevy of third-party tools that can be added to the database, setting up an implementation is a relatively simple task. Also, in general, it's also an easy database to work with. As long as there is an understanding of the language, there shouldn't be too many problems.
- **Excellent Support** – As MySQL is open source, community support and official API support is quite strong. Due to this, any issues can be easily addressed and solutions found quickly.
- **Industry Standard** – Although MySQL's popularity has waned somewhat in recent years, it remains one of the most-used database systems in the world. It's compatible with virtually every operating system, and is more or less an industry standard. This is, of course, in spite of all the folks who say it's on the way out.

15.1.2 Disadvantages of MySQL

- **Stability Issues** – MySQL tends to be somewhat less reliable than its peers. These stability issues are related to the manner in which it handles certain functions (such as references, transactions, and auditing). While the database is certainly still usable in light of these problems, they do tend to make MySQL a poor choice for certain use cases [38].
- **Poor Performance Scaling** – Although MySQL is equipped to handle a virtually limitless volume of data, it has a troubling tendency to come grinding to a halt if it's forced to deal with too many operations at a given time. This relatively poor performance scaling means that anyone with high concurrency levels should probably look into an alternative. This issue however, does not relate to this project as the volume of data will never reach large scale enterprise volumes.

15.2 MongoDB (Object-Oriented NoSQL Option)

Before mentioned the points of MongoDB, it is important to focus on the concept of NoSQL first. NoSQL offers applications a more flexible, object-oriented interface to databases systems. This allows for an agile model. Relational databases were not designed to cope with the scale and agility challenges that face certain modern applications, nor were they built to take advantage of the commodity storage and processing power available today.

MongoDB itself is an open source, document-oriented database designed with both scalability and developer agility in mind. Instead of storing your data in tables and rows as you would with a relational database, in MongoDB you store JSON-like documents with dynamic schemas.

15.2.1 Advantages of MongoDB

- A document-based data model. The basic unit of storage is analogous to JSON, Python dictionaries, Ruby hashes, etc. This is a rich data structure capable of holding arrays and other documents. This means you can often represent in a single entity a construct that would require several tables to properly represent in a relational db. This is especially useful if your data is immutable.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- No schema migrations. Since MongoDB is schema-free, your code defines your schema.
- A clear path to horizontal scalability.
- Any data works. This means any type of data can be entered anywhere.

15.2.2 Disadvantages of MongoDB

- Data size in MongoDB is typically higher due to e.g. each document has field names stored it.
- Any data works. This means any type of data can be entered anywhere.
- Less flexibility with querying (e.g. no JOINs).
- No support for transactions - certain atomic operations are supported, at a single document level.
- At the moment Map/Reduce (e.g. to do aggregations/data analysis) is OK, but not blisteringly fast. So if that's required, something like Hadoop may need to be added into the mix.
- Less up to date information available/fast evolving product.

15.3 Deciding Factor

It is important to keep in mind that a non-relational database is ***not better*** than a relational one. If your database has a lot of relations and normalization, it might make little sense to use something like MongoDB. It's all about finding the right tool for the job. As mentioned, any data anywhere is both an advantage and disadvantage.

Due to the nature of the data related to the PiLYNK system, a relational option is a better choice. Performances factors were not considered, rather the scale of relationships between different data dimensions considered from pre-specification.

16. Final Thoughts

Research was is a huge part of a project of this nature. Without question, areas of the project would be impossible without exact understanding of low level technologies and high technologies alike. Manchester Codes are the newest concept researched in terms of previous projects, previous work and personal exposure alike.

Given the above research remarks and content covered, one of the most difficult factors of this project will be to successfully narrow and identify the project scope to ensure an efficient resulting product. Otherwise, too much with less quality will be the result. Furthermore, a sufficient amount of time will have to be given to creating a solution for outputting Manchester Codes and ensuring the reliability of the codes across the all devices. Should this aspect of the project be incomplete, further work may become incredibly difficult as control over devices will require further research for alternatives. However, given the obvious risk, the project research points should provide for a large base of technologies to work from and discover some solution eventually, a varied set of concepts to explore and finally, a satisfactory result at the end of the project life cycle.

17. References

[1] C. Dixon, “The Home Needs an Operating System (And an App Store),” Microsoft Research University of Washington, Washington, 2010.

[2] Belkin, “Belkin WeMo Products,” 20 September 2015. [Online]. Available: http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/.

[3] Belkin, “Belkin WeMo Switch,” Belkin, [Online]. Available: http://www.belkin.com/us/F7C029-Belkin/p/P-F7C029/. [Accessed 27 October 2015].

[4] L. W. R. PLC, “Light Wave RF,” Light Wave RF, 5 May 2015. [Online]. Available: http://lightwaverf.com/. [Accessed 2 October 2015].

[5] ADT, “ADT Security Specialists,” 2015 ADT LLC DBA ADT Security Services, 2015. [Online]. Available: http://www.adt.com/. [Accessed 1 October 2015].

[6] Philips, “Meet Hue,” Philips, 2015. [Online]. Available: http://www2.meethue.com/en-gb/about-hue/what-hue-does/. [Accessed 2 October 2015].

[7] Philips, “Philips Apple HomeKit,” Philips, 2015. [Online]. Available: http://www2.meethue.com/en-gb/friends-of-hue/apple-homekit/. [Accessed 2 October 2015].

[8] Intel, “Intel Galileo,” Intel, 2015. [Online]. Available: https://www.arduino.cc/en/ArduinoCertified/IntelGalileo. [Accessed 15 September 2015].

[9] R. P. Foundation, “What is a Raspberry Pi,” Raspberry Pi Foundation, 2015. [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed 10 September 2015].

[10] A. Fruit, “Raspberry Pi B+ Model,” Ada Fruit, 2015. [Online]. Available: https://www.adafruit.com/products/1914. [Accessed 5 September 2015].

[11] A. Retail, “Intel Galileo,” Amazon, 2015. [Online]. Available: http://www.amazon.co.uk/Intel-Galileo-Gen-400MHz-Motherboard/dp/B00MTKHIFE. [Accessed 1 October 2015].

[12] R. P. Foundation, “Python Pi,” Raspberry Pi Foundation, 2015. [Online]. Available: https://www.raspberrypi.org/documentation/usage/python/. [Accessed 1 October 2015].

[13] R. P. Foundation, “Raspberry Pi Downloads,” Raspberry Pi Foundation, 2015. [Online]. Available: https://www.raspberrypi.org/downloads/. [Accessed 15 September 2015].

[14] C. L. U. a. Canonical, "Snappy Ubuntu," 2015 Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd., 2015. [Online]. Available: https://developer.ubuntu.com/en/snappy/. [Accessed 15 October 2015].

[15] P. I. T. S. Project, "Pi In The Sky," Pi In The Sky Project, 2015. [Online]. Available: http://www.pi-in-the-sky.com/. [Accessed 15 October 2015].

[16] U. o. Glasgow, "Raspberry Pi Cloud," N/A, [Online]. Available: https://raspberrypicloud.wordpress.com/. [Accessed 12 October 2015].

[17] Common, "Ultra High Frequency," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Ultra_high_frequency. [Accessed 05 November 2015].

[18] Wikipedia, "CEPT," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/European_Conference_of_Postal_and_Telecommunications_Administrations. [Accessed 06 November 2015].

[19] L. Rayleigh, "Waves Propagated Along the Plane Surface of an Elastic Solid," London Math Society, London, 1885.

[20] D. Chattopadhyay, "Electronics - Fundamentals and Applications," New Age International, Delhi, 1996 - 2006.

[21] ZWave, "ZWave Smart Home Solutions," ZWave, [Online]. Available: http://www.z-wave.com/. [Accessed 15 January 2016].

[22] O. Staněk, "IP Protocol Analyzer," N/A, [Online]. Available: http://www.ostan.cz/IR_protocol_analyzer/. [Accessed 1 November 2015].

[23] Cuckos, "Reaper FM," Cuckos, [Online]. Available: http://www.reaper.fm/. [Accessed 1 November 2015].

[24] ABYZ, "GPIO," ZBYZ, [Online]. Available: http://abyz.co.uk/rpi/pigpio/index.html. [Accessed 5 11 2015].

[25] Adafruit, "How PIR's Work," Adafruit, [Online]. Available: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work. [Accessed 15 December 2015].

[26] Unknown, "PIR Sensor RE 200B," [Online]. Available: https://learn.adafruit.com/system/assets/assets/000/010/134/original/RE200B.pdf. [Accessed 2 January 2016].

[27] Adafruit, "Specification of Fresnel Lens," [Online]. Available: https://learn.adafruit.com/system/assets/assets/000/010/135/original/NL11NH.pdf. [Accessed 2 January 2016].

[28] Adafruit, "Pyro Electrics IRS-A200ST01," [Online]. Available: https://learn.adafruit.com/system/assets/assets/000/010/137/original/pyroelectrics21e.pdf. [Accessed 2 January 2016].

[29] Adafruit, “Adafruit DHT22,” Adafruit, [Online]. Available: https://www.adafruit.com/products/385. [Accessed 1 November 2015].

[30] W. Tech, “JavaScript Libraries,” W3Tech, 2015. [Online]. Available: http://w3techs.com/technologies/overview/javascript_library/all. [Accessed 15 November 2015].

[31] Boostrap, “Bootstrap,” Boostrap (MIT License) & Core Team, 2016. [Online]. Available: http://getbootstrap.com/. [Accessed 20 December 2016].

[32] Fenopix, “CanvasJS,” Fenopix, [Online]. Available: http://canvasjs.com/. [Accessed 15 01 2016].

[33] JinjaDevCommunity, “Jinja2,” Pocoo, [Online]. Available: http://jinja.pocoo.org/docs/dev/. [Accessed 10 10 2015].

[34] P. S. Foundation, “Python,” [Online]. Available: https://www.python.org/about/. [Accessed 01 08 2015].

[35] PHP, “What can PHP do,” [Online]. Available: https://secure.php.net/manual/en/intro-whatcando.php. [Accessed 08 11 2016].

[36] J. Jackson, “PHP Gets a Formal Specification, At Last,” [Online]. Available: http://www.itworld.com/article/2697195/enterprise-software/php-gets-a-formal-specification--at-last.html. [Accessed 10 11 2015].

[37] Oracle, “MySQL,” [Online]. Available: https://www.mysql.com/. [Accessed 01 08 2015].

[38] D. Ocean, “SQLite vs MySQL vs PostgreSQK,” [Online]. Available: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems. [Accessed 05 12 201].

[39] WeMo, “WeMo,” 1 October 2015. [Online]. Available: http://www.wemo.com/.

[40] T. Radar, “Tech Radar Computing Components,” Tech Radar, 2015. [Online]. Available: http://www.techradar.com/news/computing-components/motherboards/8-best-micro-atx-and-mini-itx-motherboards-923069. [Accessed 1 October 2015].

[41] R. P. Foundation, “Raspberry Pi Features,” Raspberry Pi Foundation, 2015. [Online]. Available: https://www.raspberrypi.org/products/model-b-plus/. [Accessed 1 October 2015].

[42] L. Midwestern Mac, “Introducing Dramble Raspberry,” 2015. [Online]. Available: http://www.midwesternmac.com/blogs/jeff-geerling/introducing-dramble-raspberry. [Accessed 12 October 2015].

[43] W. Schools, “HTML Element Reference,” W3 SChools, 01 01 2015. [Online]. Available: http://www.w3schools.com/tags/. [Accessed 11 November 2015].

[44] W. Schools, "CSS Reference," W3 Schools, 01 01 2015. [Online]. Available: http://www.w3schools.com/cssref/default.asp. [Accessed 11 November 2015].

[45] G. Fairhurst, "Manchester Encoding," Unknown, Unknown, 2007.

[46] Z-Wave, *Z-Wave Technologies,* N/A, 2015.

[47] SmartHome, "ZWave - The Basics," SmartHome, [Online]. Available: http://www.smarthome.com/sc-what-is-zwave-home-automation. [Accessed 20 September 2016].

[48] C. Dixon, "An Operating System for the Home," USENIX, N/A, 2015.

18. Table of Figures