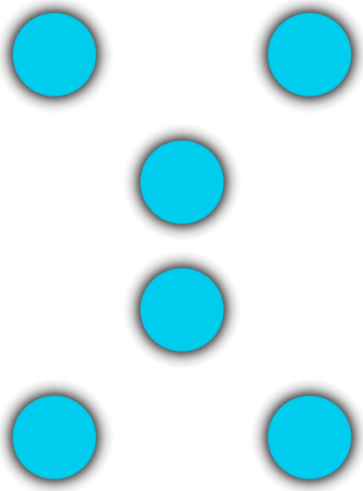
Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

PiLYNK

Home Automation System



Final Project Report

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

Table of Contents

1 Introduction

This document contains the final report for the PiLYNK home automation system, developed from October 2015 to mid-April 2016 as part of the 4th year Software Development course (Hons BSc.) at the Institute of Technology, Carlow.

The project was constructed in three iterations, divided up into relatively equal time frames throughout the project life cycle. All iterations presented their own set of unique challenges and fair share of stresses that greatly tested my commitment and dedication to achieving the desired end result.

The following sections will include a detailed description of the project, the challenges faced throughout the project, conformance to specification, learning outcomes (both technical and personal), a final review of the project and final acknowledgements for any invested parties or any individual person who contributed to the success of the project.

1.1 PiLYNK Home Automation System

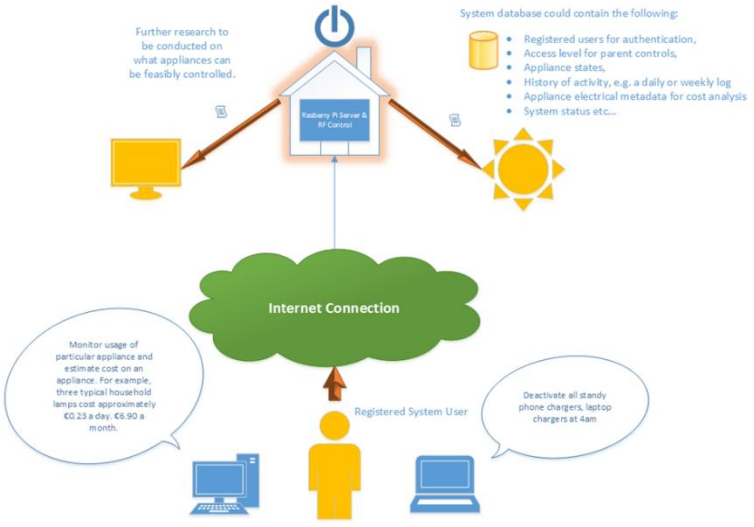
1.2 What is PiLYNK?

PiLYNK Home Automation is a fully integrated home operating system which allows any property/premise dweller, whether tenant, owner, business, guest or in house carer, the ability to interact with devices in their residence remotely, from anywhere in the world, at any time, under the assumption internet access is available.

Some of the example usage scenarios of PiLYNK:

1. Automating light control when on long vacations for sense of security.
2. View device usage displays and cost analysis to identify cost reduction strategies and micro manage individual devices.
3. Enable security modes to alert administrative users by email and or text when intruders are detected (this should be used in pair with an existing security system to ensure maximum peace of mind).
4. Automating timeouts on central device nodes for infant rooms.
5. View temperature metrics to analysis the right time to enable home heating.
6. Create device groupings to macro manage several devices.
7. Enable a motion controlled lighting solution to reduce the amount of wasted electricity for needless light usage.
8. Add guests to allow shared control, useful for large home environments.
9. Keep your data safe – all data stored on PiLYNK is encrypted with an SHA1 encryption algorithm. However, owners can permit open API access for publishing usage data for analysis (meta-project feature).

The main focus of the project, at least initially, was to allow for wireless control of devices in the home. This was the foundation upon which the rest of the key features were identified. The image below illustrates a conceptual overview of PiLYNK.



1.3 Motivation

The key goal over all was to empower people to make a more knowledgeable effort to reduce the unnecessary consumption of finite resources on the earth and subsequently hinder the growing issue of the human carbon footprint show to be increasing at alarming rates. As illustrated below in an excerpt from the Environmental Protection Agency's reports on Carbon emissions:

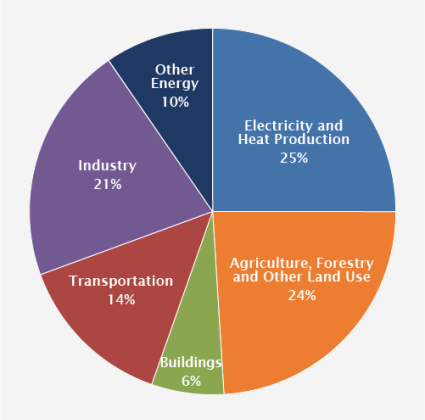


Figure 1.a - Illustration of Electrical Output [1]

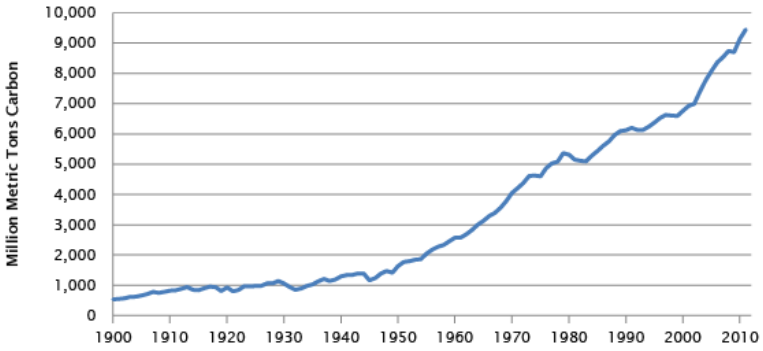


Figure 1.b - Rise in Carbon Emissions since 1900 [1]

The hope is that intelligent control over a homogeneous group of devices throughout a typical home or premise may help in achieving this by encouraging greener habits through the use of technology, particularly for the younger generation.

The motivation was the simple idea: one individual person can generally only make a small change, but many people making small changes will achieve a lasting change. As quoted from Tony Robbins.

"By changing nothing, nothing changes."

---

2 System Overview

PiLYNK is a mixture of not only software disciplines, it also touches on network engineering, a fundamental understanding of electrical engineering, circuitry and microcontrollers, and to some degree, data analytics and business reporting concepts.

The following section will explore the breakdown of low level communication hardware components, front end presentation software aspects and the server operating model.

2.1 Low Level Components

At the lowest layer of the PiLYNK, there exists two sets of grouped components:

- Radio communications array – 433MHz radio transmitter and receiver.
- Sensor network – Motion sensor and temperature sensor.

These devices are built upon integrated circuits and microcontrollers.

2.2 Server Hardware

At the heart of the whole project is the single board computer, the Raspberry Pi, which is about the height and width of a credit-card and depth of a standard scientific calculator. It is very much the core of the project's operating infrastructure as all development branches depend on it to function.

The Pi acts as the web server, the database housing hardware, the server application controller, the communication driver and the sensor job manager.

The Raspberry Pi runs a subset Debian/Linux distribution known as Raspbian (JESSIE revision [Kernel 4.1]). It is available as an open source operating system, no different to other Linux distributions. The operating system is stored on a micro secure digital memory unit. This unit operates with approximately 16 GB of memory. Plenty for the usage of this project and enough to satisfy large data sets that will accumulate over time. Moreover, expansion options exist that extend this memory even more.

The server itself runs on a Python codebase with Flask and Werkzeug. This allowed for a huge amount of integrated tools to be explored for improving and expanding the initial concepts. The server itself runs over a MySQL database. MySQL performed exceptionally well throughout the project, even given the lower clock speeds and RAM on the Pi, compared to large scale dedicated server machines and even desktops, it ran flawlessly and communication with the server has been completely reliable to date.

The last element to mention is the DDNS support for allowing network access to the Pi. This came in the form of a NoIP client package linked to the Pi's own network control. This allowed for the chosen domain – keithbyrnepi.ddns.net, to be statically linked to the dynamic public IP that the Raspberry Pi is given via the internet service provider's (ISP) dynamic host control protocol (DHCP). As there is little to no control over how an ISP manages IP's, this was the best option. It worked quite well as it provided an inexpensive remote test platform before committing to another option.

2.3 Software

The software dimension involved during the project represented the largest workload throughout, without a doubt. As it is a software development project, main focus was given to this area.

Software drives the server, controls the GUI, controls the communications array, communicates with the sensor network and maintains the persistent data store. The Python code was built with the Model View Controller (MVC) design pattern governing the data, user interface and business logic interaction process.

Beneath the interactions of the MVC. The PiLYNK runs with a series of scheduling operations which handle time sensitive aspects such as schedules or listening services for sensors. All these services represent a cyclical operating model of one service helping other services to continue serving their functions. For example, the scheduling job operates on an hourly trigger interval, sweeping the database for upcoming schedules, queuing the jobs for switching devices at their designated date-time signatures which feeds the cost metrics area with new data to report.

These services, must run without error or the system will eventually fall into a state of disrepair and become unpleasant and unreliable. This presented a large test of patience, which will be discussed later.

A huge amount of new technology was discovered on this development journey. These will also be discussed later in this document, although not completely exhaustive, it is a breakdown of software tools used, explored and in some cases, cursed for hours due to errors or poor API's.

2.4 Graphical User Interface (Presentation)

The presentation layer of PiLYNK represents what the user sees and how it interacts with the lower layers. This is often expressed as the Graphical User Interface (GUI). This layer is constructed using Flask and Jinja2 with HTML templates at its simplest. The 'views' that the users see are served by the Flask controller, the C and the V in the MVC. The system is secured by a login system so access is heavily locked down and can only be achieved through administrative approval. This was necessary as risks from malicious activity with devices in the home was a real concern. For the purpose of the user side demonstration, sample credentials will be used.

Mobile first development was a large concern with the design of the front end of the system, therefore, in the following demonstration, both mobile and desktop usage scenarios will be shown.

Note: *Not every aspect of the front end will be explored, mainly key features that are of exceptional importance and somewhat functionally informative.*

2.4.1 Logging In

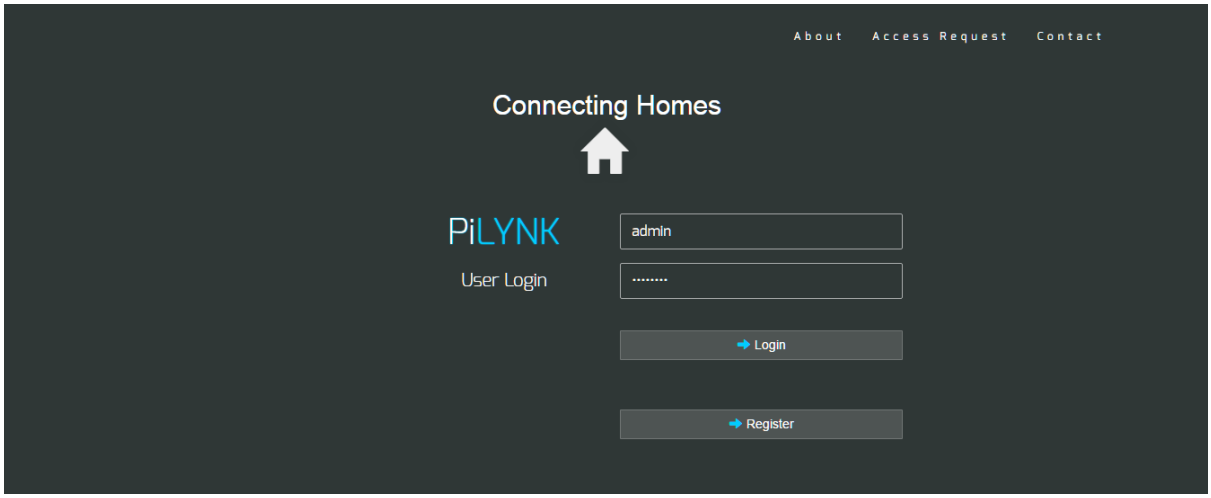


Figure 2.a - PiLYNK Login

Once users have successfully entered their details and submitted the login request. The system approves and navigates the user to the home screen.

2.4.2 Dashboard (Home)

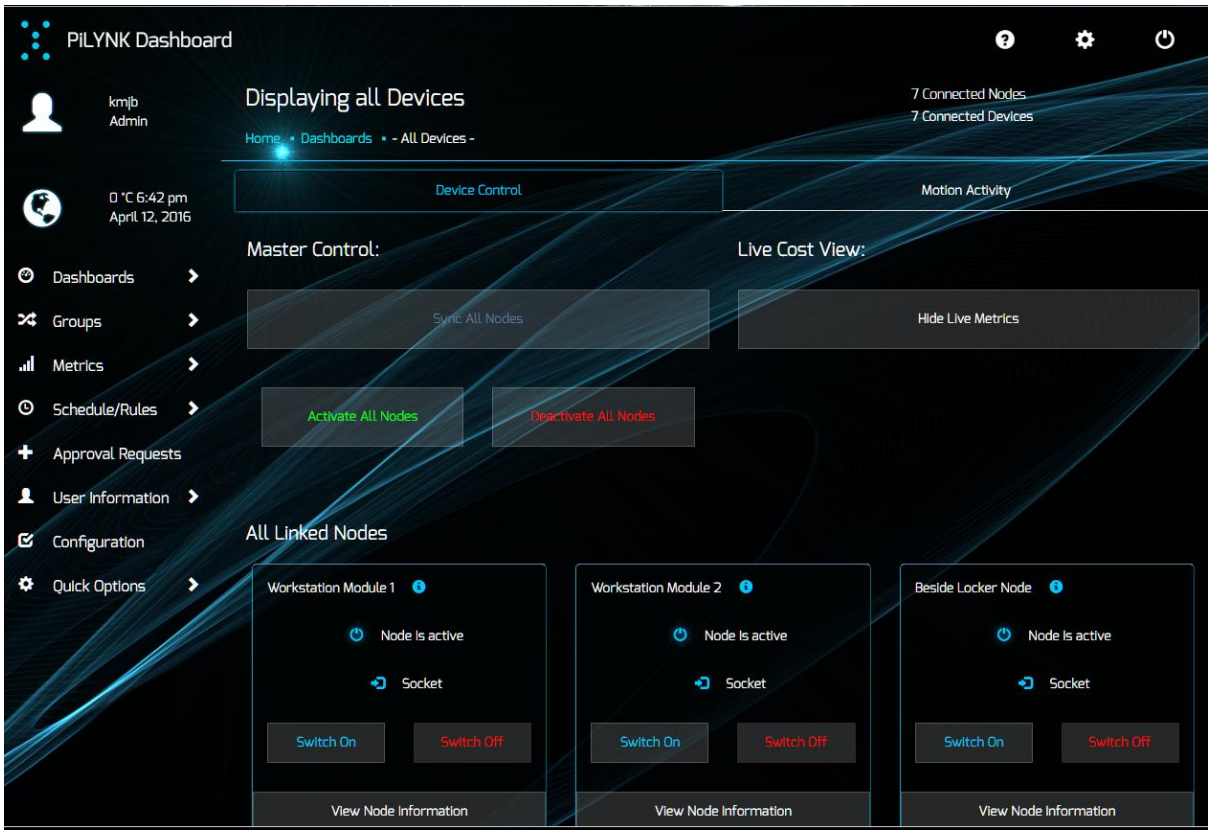


Figure 2.b - PiLYNK Dashboard

The dashboard is the main control interface for the core function of controlling devices and later revised for control of the motion control also. From this view, users can switch on and off linked nodes, view live cost estimation metrics and control the nodes via master control (all on, all off, sync).

2.4.3 Viewing and Controlling Linked Nodes

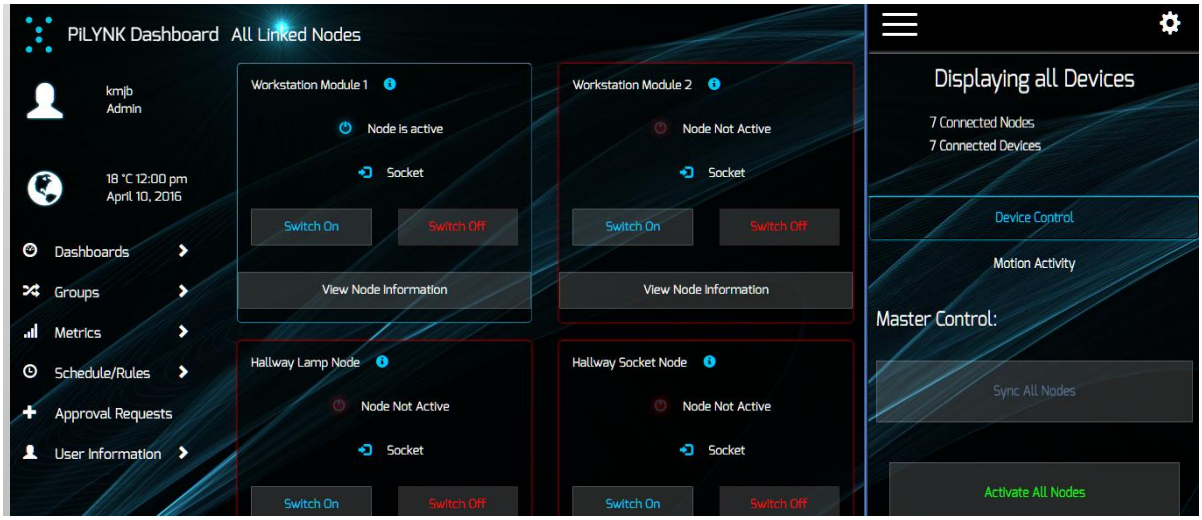


Figure 2.c - Dashboard Focused on Nodes Desktop & Mobile

The above focused view shows all the nodes that are linked to the PiLYNK system. Each of these nodes represent a single control object in the node network. Users can select switch on or switch off or extend the details of the node or edit the node.

When a user switches any given node, the update is instantly received from the system and subsequently the node display statuses are accurately reflected.

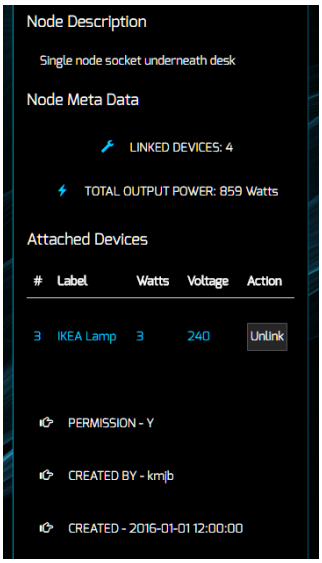


Figure 2.d - Node Information

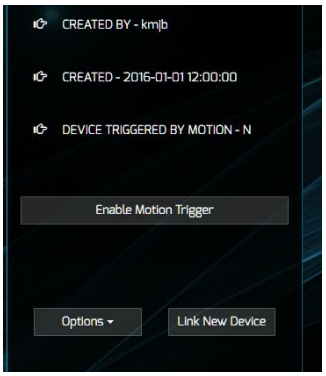


Figure 2.e - Node Options

Figure 2.f - Node Options

2.4.4 Viewing Live Metrics

By selecting the View Live Metrics option on the main dashboard, users will be presented with a live cost and watt output estimation display. This information is live and updates every second. It is useful for viewing the usage for manual record keeping of watts and costs.

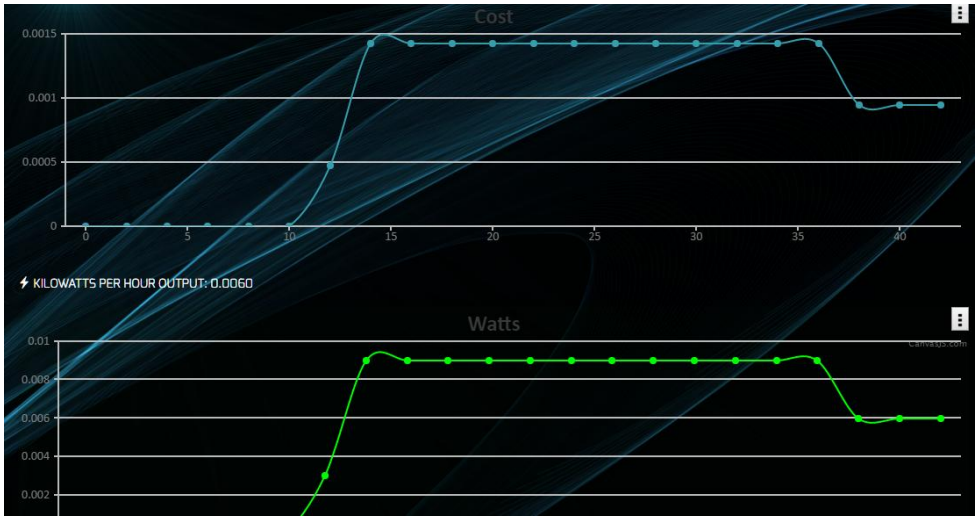


Figure 2.g - Live Metrics View

2.4.5 Basic Details

In addition to these functional points, basic node information can be viewed at all times from the main dashboard for quick reference at the top of the view display:

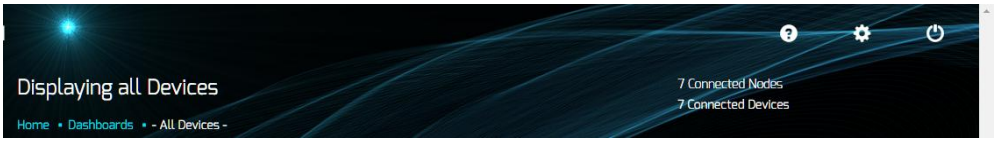


Figure 2.h - Master Node Data

2.4.6 Controlling Motion Sensor

The lockdown mode, which alerts users if motion has been detected, can be controlled from the motion activity tab. This tab allows users to manipulate the motion sensor. Options are illustrated below:

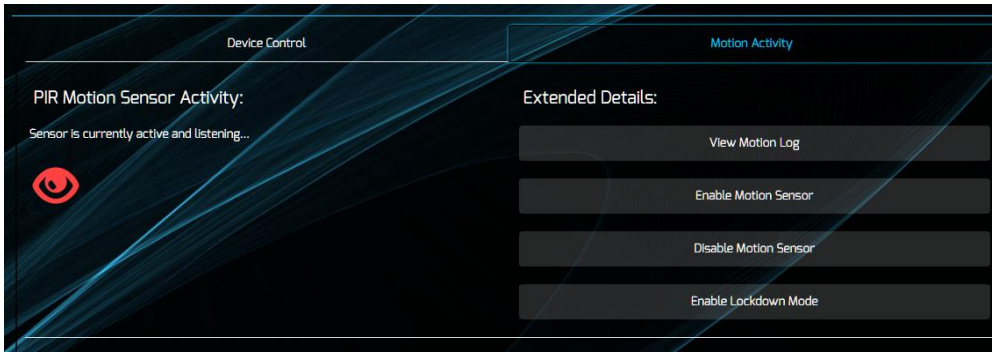


Figure 2.i - Motion Activity

2.4.7 Navigation

The left most navigation bar allows users to navigate through the different pages, this is broken down into Dashboard types, Groups, Metrics, Schedules, Approval Requests, User Info, Configuration and Quick Options. As this is hidden from standard mobile view, both menus are illustrated below:

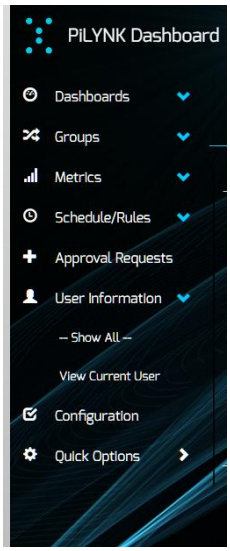


Figure 2.j - Desktop Navigation

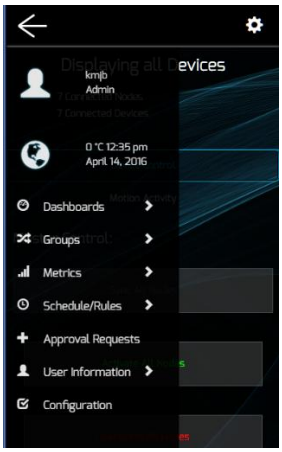


Figure 2.k - Mobile Navigation

Within the navigation pane is a self-user view display and also live temperature updates represented by the head icon and globe icon respectively. Here users can update their password, add a new address or view their own details.

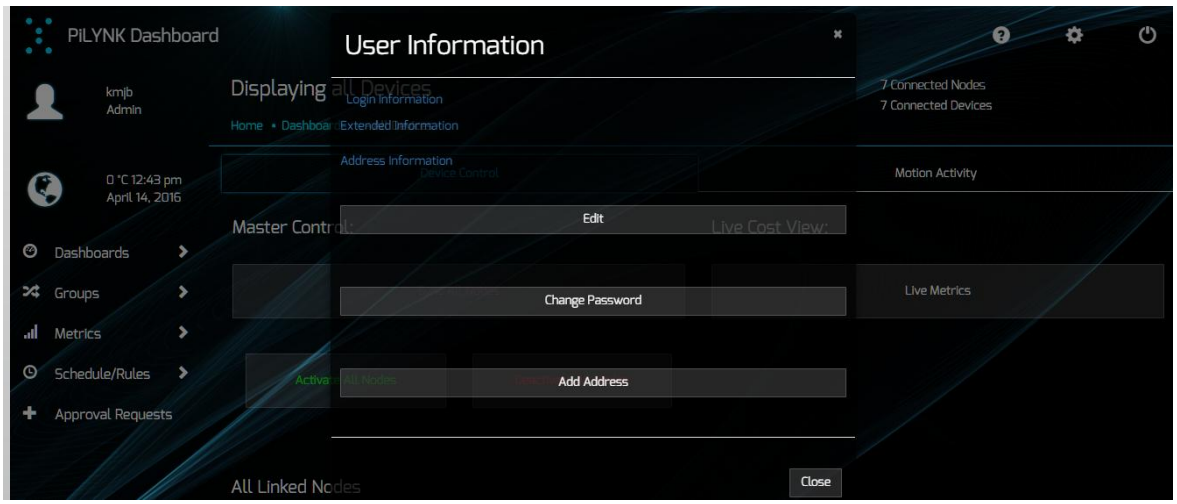


Figure 2.l - User View

2.4.8 Viewing Metrics

2.4.9 Temperature

The temperature metric view is a pre-set data view with charts generated from the past twenty four hours, weekly (Monday to Sunday) and Monthly (each month). These sub sections can be selected from the master navigation tab at the uppermost section of the view. The daily report is set by default.

Figure 2.m - Climate Metric Options

The charts illustrates both the recorded temperature and the air humidity percentage.

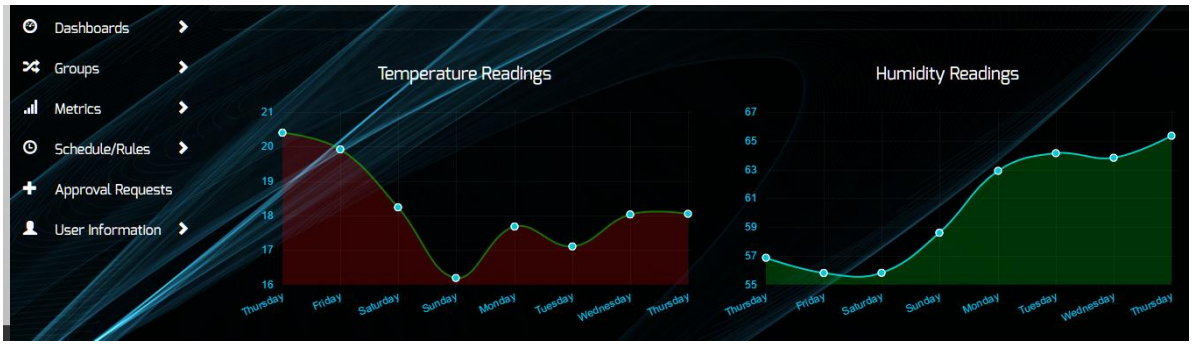


Figure 2.n - Weekly Chart Data

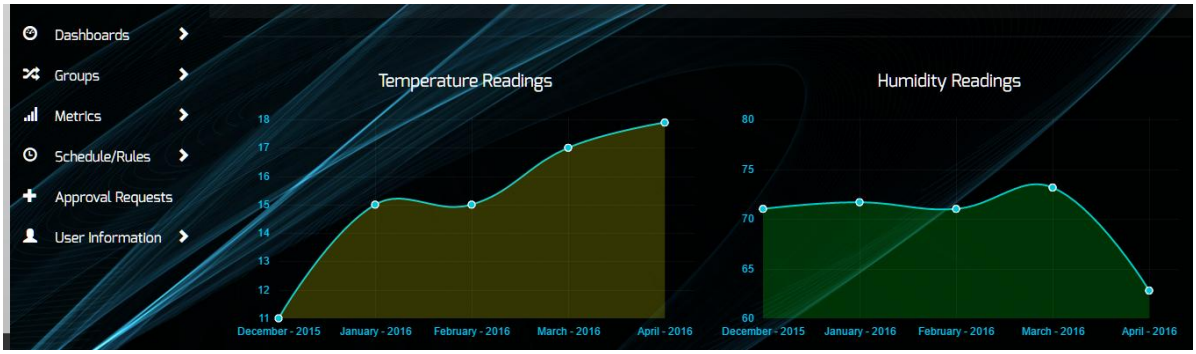


Figure 2.o - Monthly Chart Data

2.4.10 Motion Metrics

The motion sensor captures key data elements when a motion event has been detected. The data is illustrated below. Note: the lockdown record feature shows if the lockdown mode was enabled or not when the event occurred. The data is in tabular form and can be filtered for data refined and number of events expanded.

Figure 2.p - Motion Sensor Activity Data

2.4.11 Cost Metrics

Cost Metrics offer the most informative of all the metric types for general users. This is broken into two categories, live running costs for today (since midnight) and a ranged analysis for weekly, monthly and custom period. The custom period offers the most interactive aspects as will be illustrated below.

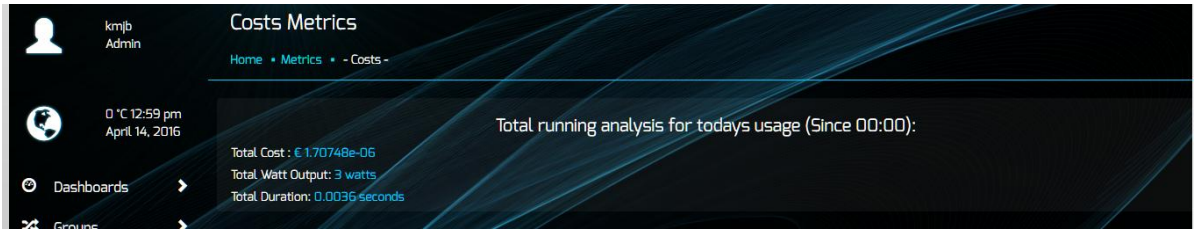


Figure 2.q - Today's Cost Analysis

Users can select a custom period from date-time **x** to date-time **y.** This allows users to view how much costs have changed since past records. This data is relevant for nodes and all devices attached to nodes at the time of record.

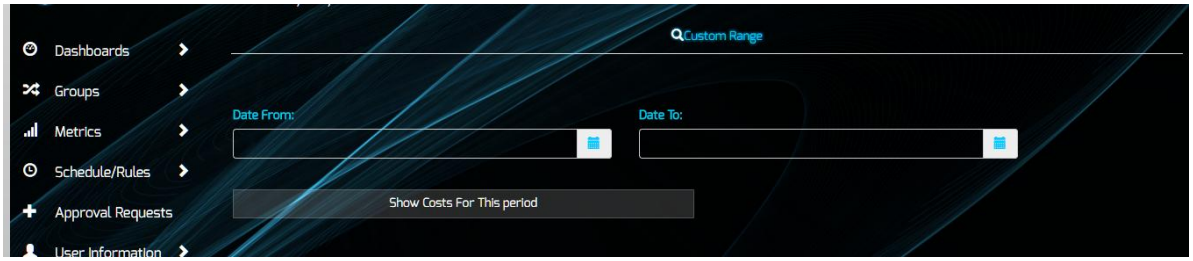


Figure 2.r - Data for Time Period Entered

For illustration purposes, the manual input date from will be the 2nd of September 2015 to today's date.

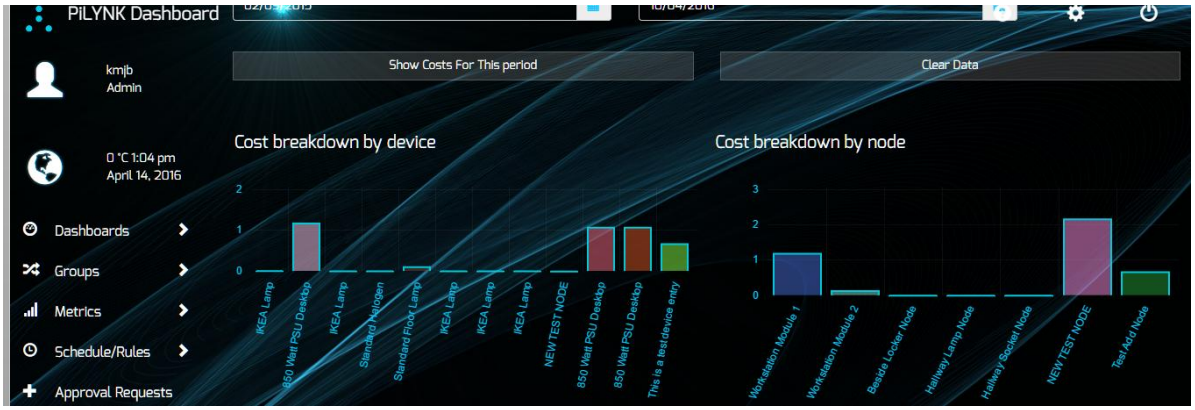


Figure 2.s – Data for time period x to y

2.4.12 Schedule Manager

The schedule manager view allows users to view currently queued schedules and also view passed schedules. This feed is live also and times are reflected as countdowns on the base information panel. More details are available from the view button.

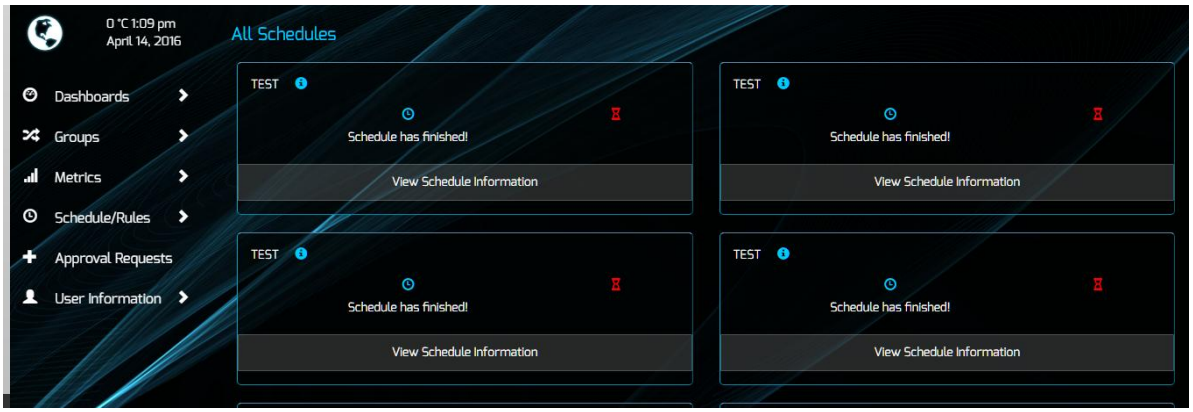


Figure 2.t - Schedule Viewer

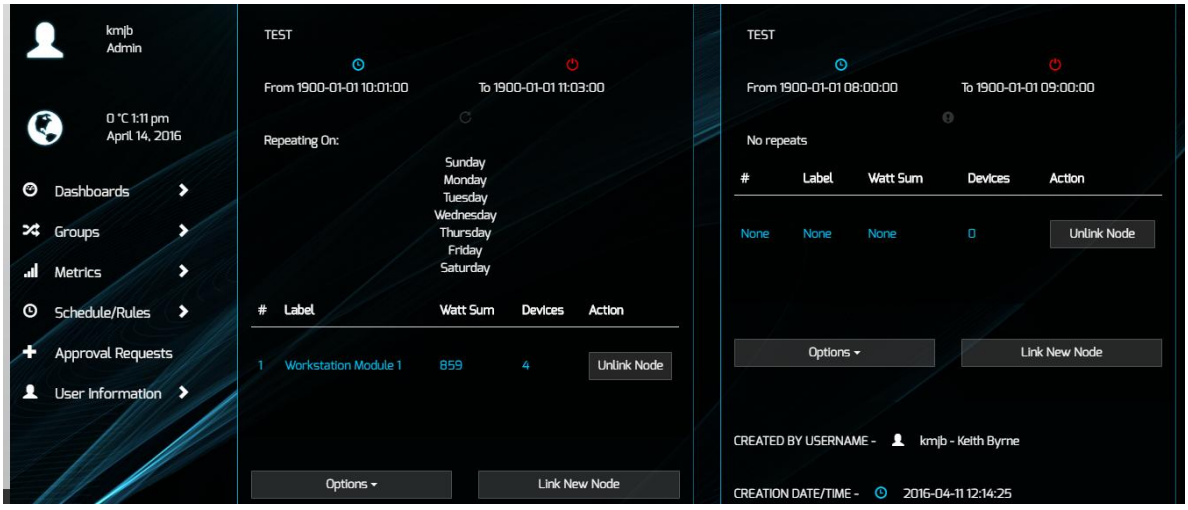


Figure 2.u - Expanded Information

Furthermore, in this expanded view there is also the daily repeat pattern (if it is a repeating schedule) information and details regarding the nodes linked to the schedules. Nodes can be added or removed from the schedule mapping here and schedules also cleared or disabled.

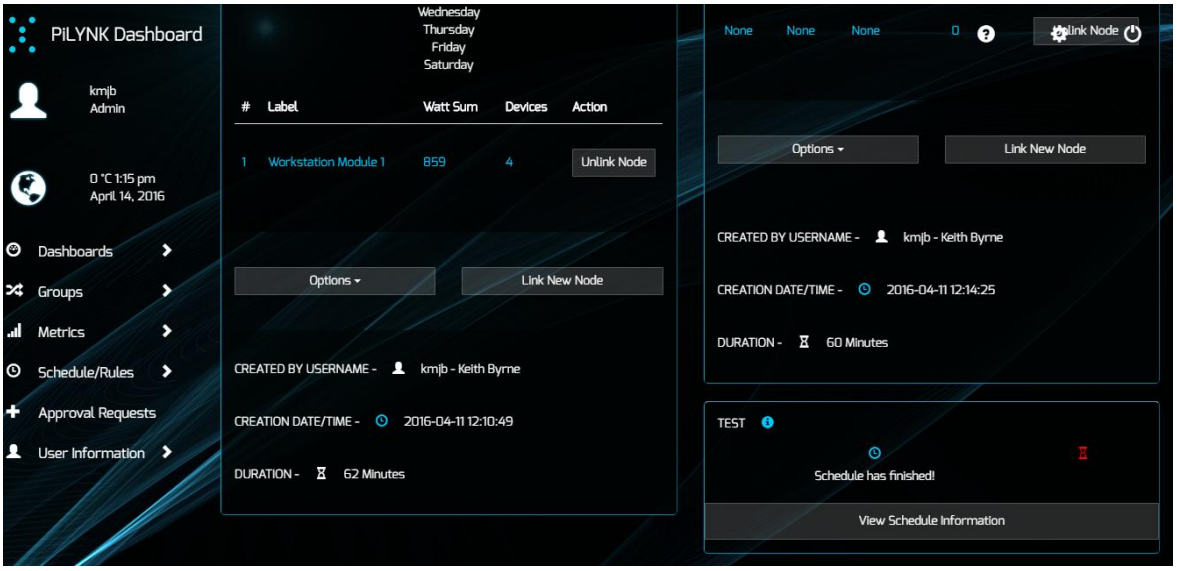


Figure 2.v - Schedule Information

2.4.13 Adding Devices

The system relies on the linking of nodes for most of the functionality to extend from. Initially, manual linking of a set number of nodes was the way forward. However, after much consideration in iteration two, I decided to link nodes dynamically using a simple training algorithm for nodes. This means that users can add nodes themselves with ease. The below demonstration shows this process in three steps: data entry, training/testing and finally submitting. The following graphic shows the form entry section.

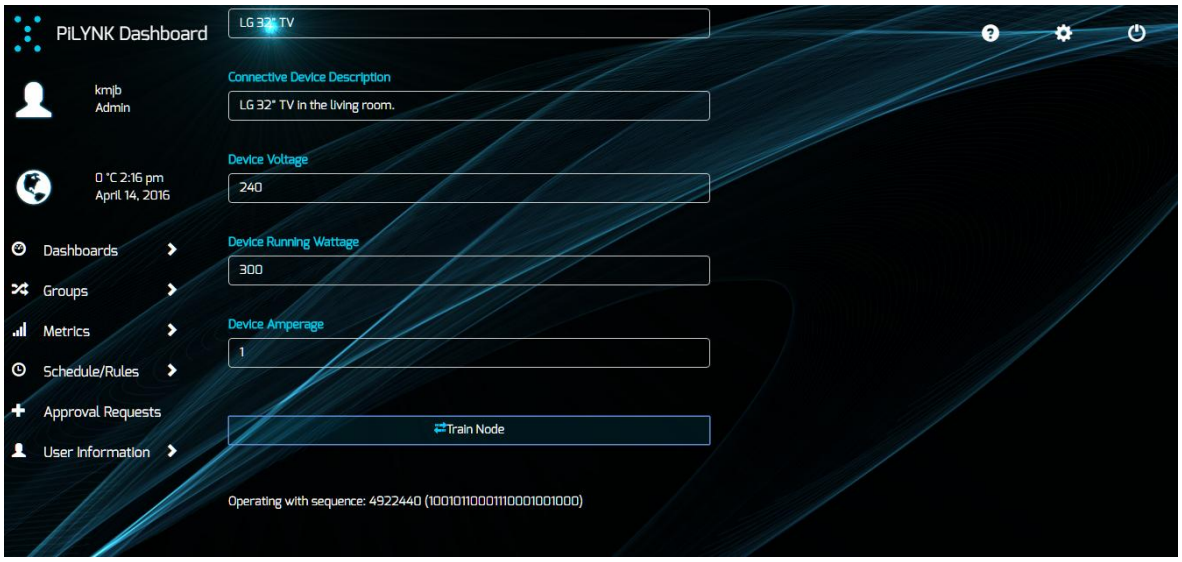


Figure 2.w - Example Form Details

Next the user selects the train node option, this will require to user to put their node into study mode outside of the system. The system will then test the node by publishing the new codes for activation and deactivation intermittently for ten seconds.

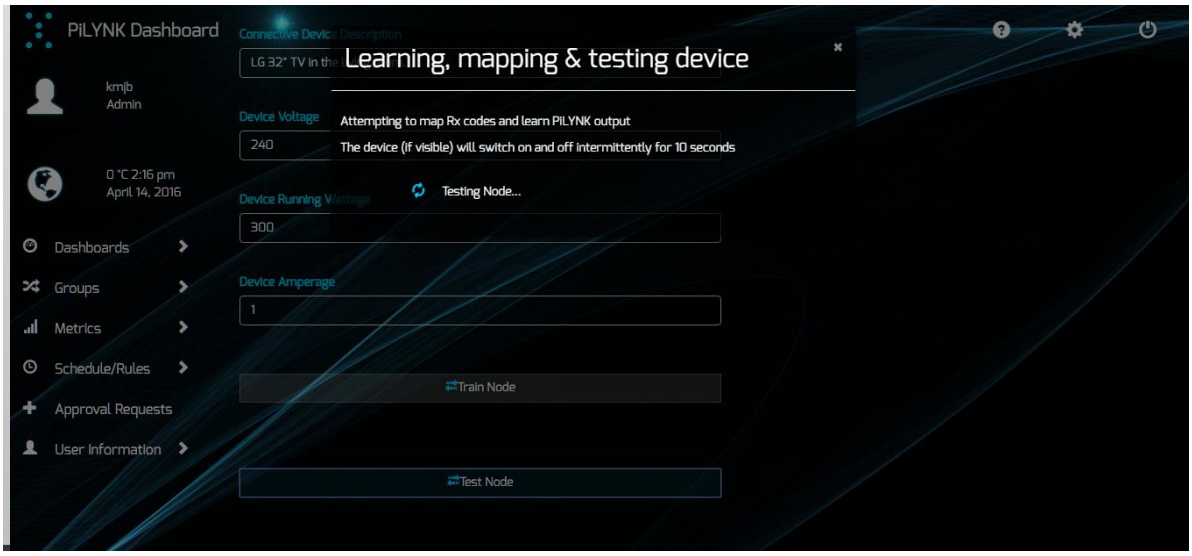


Figure 2.x - Testing Display

Once the testing process has lapsed, the user will be prompted with a confirmation message stating the test process has been successful on the system side.

Note: *This is an area where the system can be misused and no protection against malicious behaviour can be put in place here. It is the priority of the end user to ensure that the node linking process is correctly adhered too, meaning nodes should be placed into study mode at this point. However, if such a misuse should occur, node linking can still be performed post-submission by simply entering study mode and selecting off from the main dashboard for the respective node.*

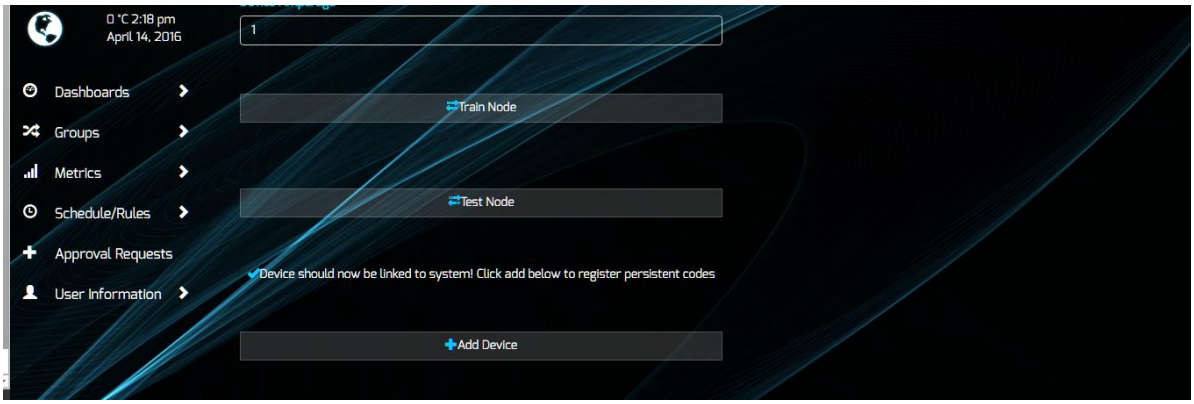


Figure 2.y Successful Test Operation Prompt

This concludes the sample user demonstration for the PiLYNK system.

3 Conformance to Functional Specification & Design

This section will detail the contrast between end product and specified product during the early stages of the project and also design stages during the middle stage of the project.

As my project was proposed and subsequently approved by my supervisor, I had the duty of proposing my own scope and this was subject to review. Thus, I found there were many changes mid-way through iteration one in terms of how far I can bring the project and with what hardware, at my own discretion.

Initially, controlling devices was the aim. However, as my understanding of the platform I was working with grew rapidly with the sheer quantity of active research, I discovered more options as time went on. Because of this, the security branch was added post-spec during iteration one. The documents are however updated to reflect these changes.

My main aim naturally was to incorporate as much technology as possible, get familiar with as many concepts as possible and present an intelligent solution based off of my work. Even though there may have been small changes, nevertheless, the end product is very much what was promised in the initial specification. To aid in this contrast effort, I have attached the initial proposal for assistance.

…….My proposed 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. 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, for example.

Physical System

I would consider using a Raspberry Pi or Intel Arduino. At the moment the Pi has much broader community support and less reinventing will be needed for handling the general purpose input output pins.

Front-End

At the moment, I would consider firstly a browser for handling the user's interaction with the system, with the option of an Android application if the supervision considers it feasible.

Backend/Server

The list of options for Raspberry Pi is rather long and I have yet to consider a viable candidate for this. Plenty of frameworks are available and also plenty of operating systems are available for the Pi itself, with Ubuntu Mate being the preferred so far.

Although, I feel there were certainly some unavoidable changes, additions and exclusions that had to occur to ultimately improve the outcome of the project. Certain CRUD operations were required that were initially excluded due to the workload being too much in foresight. The aforementioned security branch was an unmentioned feature initially, temperature metrics was unmentioned and providing a solution for aggregated two way communications, however low level it may be, was a solution that was created out of the need to solve the problem of providing outbound signals.

In terms of how the technology changed from initial specification discussions: The first prototype for transmitting Manchester Codes was written and developed in C++. The solution was redesigned for Python soon after. Ultimately, this provided a much more reliable solution. This will be discussed in more detail later in the document.

Finally, I can conclude that the promised product was delivered with a small number of bonus features that can appeal to a wider set of users. The following is an unsealed version of a prototype built during iteration three, nicknamed ‘Albion’.

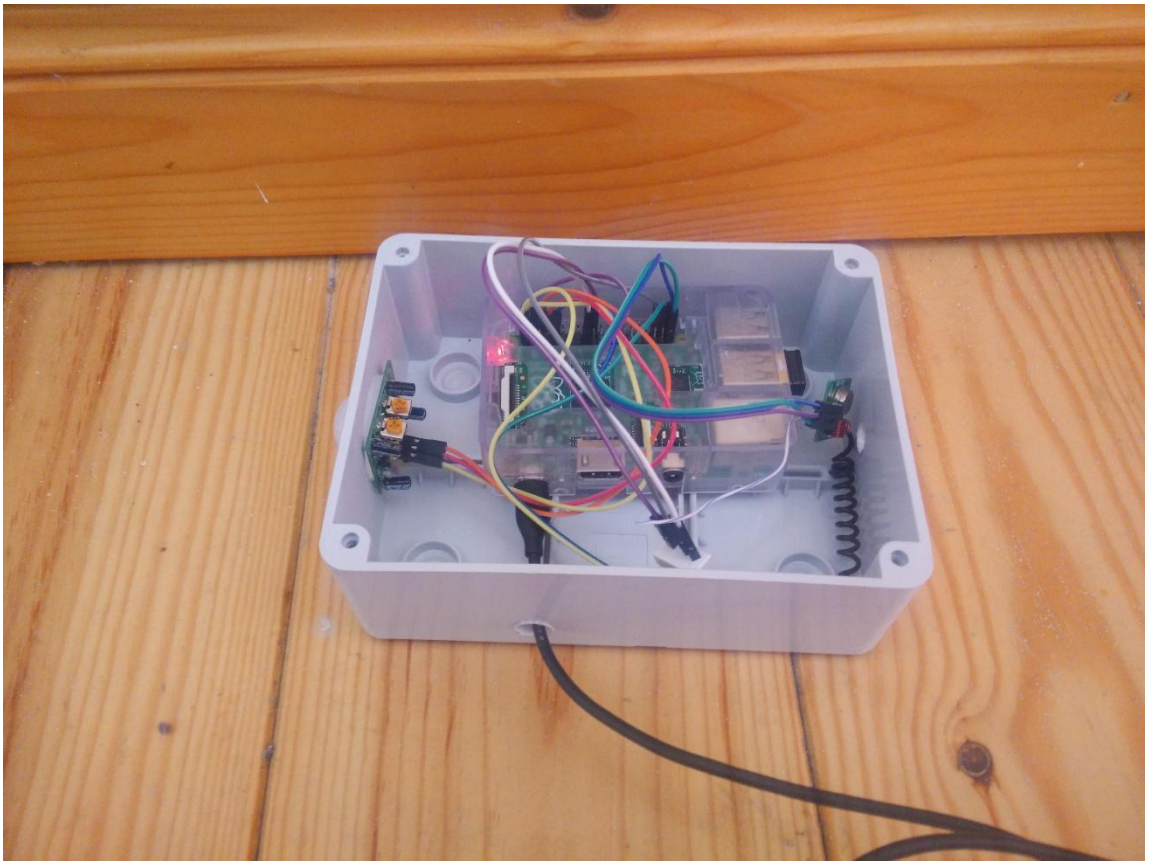


Figure 3.a - Prototype Model

4 Learning Outcomes

4.1 Personal

Firstly, I must mention that this project was hugely important for me. Not just that is was my sole responsibility to achieve a successful result but because it was a huge personal investment for me. I grew with the project as time went on and it became a part of my life as a lot of the technology I researched and implemented was completely out of my comfort zone. Bridging the gap between the realm of software development and the realm of electronic hardware required a lot of rapid education in a field of which the majority for me was completely unknown. This often resulted in my limits being tested.

In respect to this, my project was quite a risk at the beginning as I was not sure it was even possible for me to implement a solution. This caused many sleepless nights throughout October and November with several failed solution attempts. However, with great risks often come even greater reward. Nonetheless, great reward seldom comes without hard work and dedication towards a goal. As a result of this, I have gained a large amount of self-belief in my abilities to solve challenges and overcome obstacles that are seemingly unsurmountable. All things in hindsight, I would not change my approach. Additionally, I believe I have gained some new personal attributes and also improved existing attributes that were not so refined.

4.1.1 Time Management

In order to efficiently manage my time throughout the project life cycle, I had to assign my time carefully to certain priority tasks. I learned to take segments of work and divide them out into less daunting sub segments in order to check them off the list of outstanding tasks to complete. At the beginning, this was very much an imperfect skill as I was feeling constantly overwhelmed by the sheer quantity of work and the lack of time. However, not long into the first iteration I was approaching the project with a strategy in mind, and tactically dealing with the tasks and setting myself time limits in order to remove the risk of becoming over encumbered with incomplete workloads. This allowed me to foresee an end in sight and view the project as a functioning process with a real time frame rather than a chaotic problem. The following quote sums up my rational towards this quite well:

“He who fails to plan is planning to fail.”

Winston Churchill

4.1.2 Prioritising Tasks

A lot of the time I felt I was trying to do everything at once, and when one aspect failed to run properly, the entire load would fail also. In addition to managing time, I learned when to stop working on tasks that simply weren’t of a high enough priority at the given time. This was sometimes a personal difficulty weighing interest vs. purpose. Throughout the project the UI became the main focus even when it shouldn’t have, drawing time from more critical jobs like improving the server codebase and logic.

4.2 Technological

I personally feel that jumping into a final year project and not coming out the other side without trying something new or massively refining an existing skill is an opportunity lost. Under supervised learning and direction where necessary, it is a chance to explore new areas and a playground of discovery in which to roam with a reduced set of restrictions.

Technically speaking, for me, I dived into a huge set of new technologies and refined a number of existing ones also. MySQL and data modelling being an exception as it was the foundation of my work placement programme. However, web development technologies were relatively new to me and this area was an immense pool of candidate technologies to use. My decision to use Python was out of personal interest and I am personally glad I choose it. The following is a shortened list of technologies and software concepts that I wasn't overly familiar with and also technologies I was familiar with, in no particular arrangement.

- ***MySQL***
- ***HTML***
- ***CSS***
- ***JQuery***
- ***AJAX***
- ***WTForms***
- ***Raspberry Pi***
- ***Debian***
- ***Microcontrollers***
- ***Radio Technology***
- ***Sensor Technology***
- ***Analysis***
- ***HTTP***
- ***Model View Controller***
- ***ParsleyJS***
- ***CSS***
- ***Flask***
- ***Jinja2***
- ***Python***
  - ***APScheduler***
  - ***Werkzeug***
  - ***SHA Encryption***
- ***Debian/Linux***
- ***Bootstrap***
- ***ChartJS***
- ***CanvasJS***
- ***Javascript***
- ***Data Modelling***
- ***Solution Design***
- ***GitHub***

4.3 Research and Development

Developing a concept from the ground up using primary and secondary research was something that I wasn't too well versed in before this project, at least on the same scale. I found the sheer amount of research frightening at times. Stripping down irrelevant information whilst retaining valuable information is a skill well served in any industry and this was often the case throughout the project. My research even led me to some conclusions that I did not like and I was forced to make some decisions that were against my wishes or reservations in how I was approaching my solutions. That, I believe was an effective use of using data for the purpose of personal advancement and improvement rather than just validating what I already knew and believed to be truth. Contradictions in how I should approach problems led me to solving the problem.

This project and each of its stages, difficulties, successes and failures to a lesser extent, has given me a great opportunity to use my skills acquired from my several years of study to date and really integrate what sometimes I felt were lost ideologies and skills. This is a huge boost for supporting my abilities and realising that I can put a vast amount of knowledge to use in a real project and that I can truly rely on myself to get to an end point without cracking under the pressures of deadlines or frustrations that will always arise in a project scenario. This, I wholeheartedly believe, will be hugely beneficial to me, not just within the industry but also with interpersonal aspects of my life in regards to communicating problems when issues become too much, to rely on the wisdom of others when your mind may be clouded and not allowing the struggles to hinder personal progression. For this project and many other projects alike, the requirement is dedicating yourself to a cause and maintaining steadfast resolve and focusing on the greater goal in mind.

With this confidence I feel I can better judge where I would like to be, professionally, in the future. Having this achievement behind me and supporting me will remind me in times of doubt that staying on the path will lead to success, and no matter how difficult things may seem, they could always be far worse than they may appear.

5 Project Review

The following sections describe the various aspects in regard to the greatest difficulties faced, what went right versus what went wrong and the review on what changes I would make if I was to attempt the project from the beginning again.

5.1 What Went Right

Overall, I can safely state that the project as a whole went accordingly to plan, even with the challenges faced. Nevertheless, the concept from beginning to end was absolutely terrifying, knowing that it represented such a large portion of the final year made the pressure really begin to stack during the first iteration when I was still very much in unknown territory and could not see any end in sight. However, once a number of my initial milestones were reached, my sense of direction began to come back into phase and my orientation towards what needed to be done also started to align with that of my initial plan. This was key to staying on track and not straying too far from the plan. To break my final thoughts on the projects main success point down I will categorise the key areas.

5.1.1 Graphical User Interface

Graphical User Interface (GUI) design is often difficult for software developers as it is common for developers to be somewhat aesthetically challenged. I admit, I am often in this category. However, for this project, it felt it necessary to design it as I would like a system to look if I was an end user. Subsequently, I spent a large amount of time refining the core GUI and I am relatively satisfied with the outcome, excluding one or two minor cutbacks due to time that I would have liked. The design is informative yet simple, responsive, aesthetically elegant and most importantly, easy to use.

The key success point however, was the successful mobile first design. Creating a system that runs on several platforms offers a more pleasant end result.

5.1.2 Manchester Codes Algorithm

During the weeks in October, I feared this section would have belonged in the '*what went wrong'* category. The reworking of the Manchester Codes radio output module caused a lot of frustration and halted the project for several weeks due to faulty output. This required a large amount of work and was the key deciding factor between success and total failure. To better understand the early stages of this process I have included the analysis I performed on audio waveforms from a standard RF socket remote control. To briefly sum this process up, I gathered existing radio devices, deduced how they operate, performed research on how the electronics operate, performed in depth analysis into a traced form of the data output by the controller, generated an approach based on my findings and then hit the drawing board for a programmatic solution based on my findings.

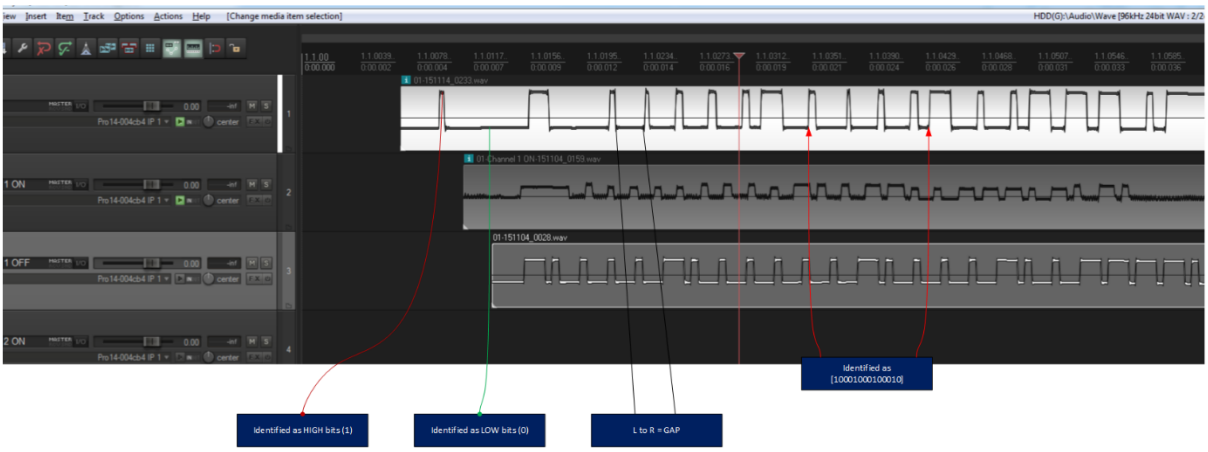
First, I will detail the rough edge approach I started with. This first involved capturing the radio outputs from a radio controller and analysis of the oscillator frequency modulation to determine the state of the binary code. At first glance, this all appeared as noise and random bits appearing on a waveform. In truth, for several days, this ***was***

all just noise. Primarily due to the fact that there was a large amount of interference from the central processing unit (CPU) in my desktop. The CPU often emits low amounts of electromagnetic radiation. It was located quite close to my work area at the time and it wasn't something I immediately identified.

To capture the codes, I simply attached a standard 1/8" microphone jack to one of the data pins and from the ground pin on a very inexpensive 433MHz radio *receiver*, and the power and ground pin back to the Raspberry Pi, which provided a safe 5v power supply to the receiver. Being careful to isolate the ground to the lower section of the jack, as the data pin outputs a 'hot' current and can damage the microphone ports on the machine used, or the resulting waveform will be a series of complete noise.

Figure 5.a - 1/8th "Audio Jack Setup

Then, I directed the remote antenna towards the receiver and captured the content using a digital audio workstation (Cubase, Reaper, ProTools etc.). From here, it was possible to view the nature of the codes. As illustrated, it is evident where the codes admit a 0 or a 1. The flat 'low' pulses represent 0, and the peaking 'high' pulses represent a 1. Initially, this was recorded upside down due to some phasing issues on input but I soon discovered the correct orientation for the code.



Attributes were gathered from the analysis above that contributed directly to the modelling process for the node and radio relationship. A key point to note is that the distance between each high and low in the bit sequence is measured in microseconds (µs – meaning one millionth of a second).

Once this step was accomplished, it was time to move onto replicating the radio transmissions using software. This was a daunting task. I was absolutely stumped by this challenge and it represented the greatest difficulty throughout the project.

Initially, I developed the module in C++ as it is a relatively stronger language for communicating with low level hardware, as it is directly descended from C. However, as time went on, I began to notice issues with the outputs and the algorithm only successfully communicated with devices about 50% of the time. This was simply not acceptable. Therefore, I decided to attempt a second solution using Python and integrated the module directly into the servers codebase rather than call C functions using the operating system. The following illustration shows an example of the difference between early code outputs in comparison to captured codes, replicated codes being the uppermost and captured being track three, four, five and six, with track two being noise from earlier tests. Notice the distortion compared to the capture.

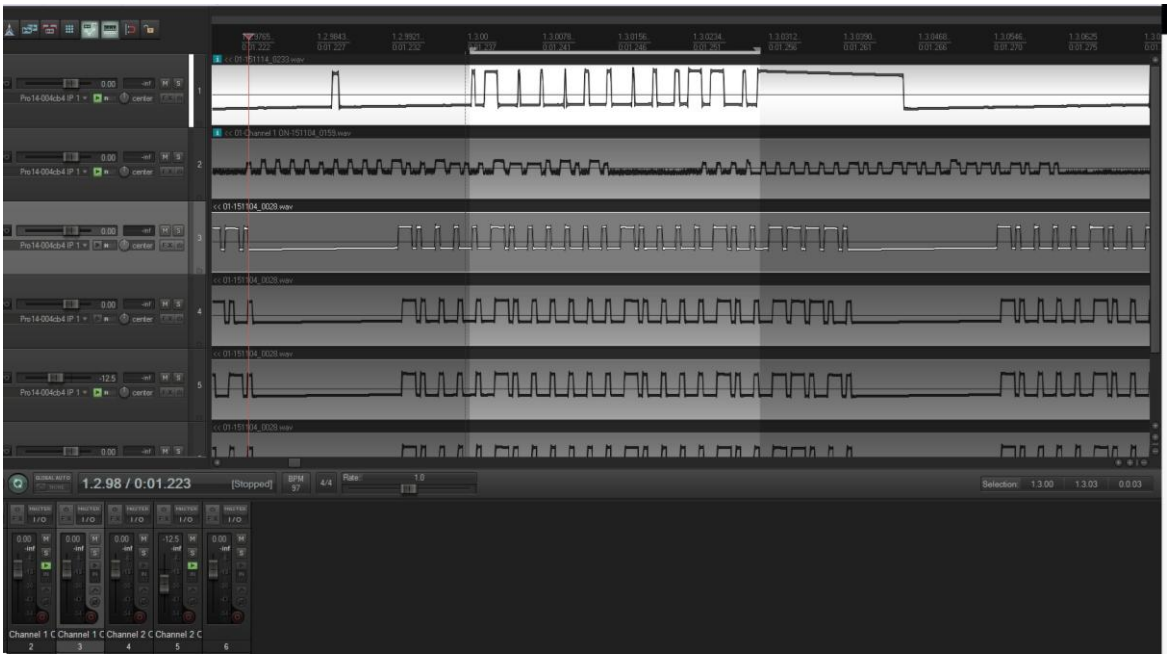


Figure 5.b - Replicated Transmission Compared to Capture

After many long hours of research and development I eventually started to notice positive results. The first prototype admitted a 20% fail rate, which was 30% better than the C++ alternative. And then working a production solution, I achieved a 92% success rate at the end of iteration two. I calculated this result from running a test code one hundred times on a preliminary test node. The test failed eight of those times. This was a milestone and considered a sufficiently low dropout out rate as the project is not a completely safety critical system. To combat the eight percent handicap, I simply programmed the output to loop with slight thread delays in between each iteration.

Finally, this allowed for the horizontal scalability of the node network, which is a key feature in the end product.

5.2 What Went Wrong

Fortunately, nothing major went wrong with the project. However, I feel it is important to mention that the time spent on GUI would have perhaps been better served focusing on cementing key internal algorithms and perfecting error handling. The quantity of CSS code is a representation of the amount of personal styling performed on the dashboard UI. I would not express it as a regret, however. Rather a concern to note if a similar situation arises in the future.

5.3 What Could Not Be

This was presented later iteration two as a possible functional introduction. The idea of allowing nodes in the network to communicate back to PiLYNK itself, in order to achieve full communication and constant monitoring capabilities. However, the complexity of circuitry required for this task was simply too difficult, given the time frame and other responsibilities. This idea intrigues me greatly. Nevertheless, I had to put it to rest or risk being distracted from the main goal. Perhaps a revision for some time in the near future!

5.4 Taking a Different Approach

If I was to return to a foreseen form of myself in the past and redo the project from scratch, I do not believe I would attempt anything differently. I performed accurate research where necessary, form reasonable conclusions and made decisions regarding every aspect of the project. I believe my technology choices were exceptionally well suited to the task they were employed for.

My design for PiLYNK was aimed to be a producible end result, in the sense that copies can be made relatively easily. This was an exciting concept also, in being able to to physically make a technological product.

5.5 Advice for Others

My most truthful piece of advice is to simply believe in yourself. Believe that you can overcome obstacles as long as you work towards it.

Additionally, I would express the importance of valid research over a trial and error approach. It proved to be a huge factor in how I proceeded with my project milestones and a trial and error approach would have cost a huge amount of time that was very much in short supply.

Don’t allow yourself to become too engrained in any set of values just because you believe them to be true. Allow for contradictions to better you approach and allow for judgement of peers. Be ready to take negative and positive criticism alike and expand on the feedback to ultimately achieve constant betterment.

Speak with your supervisor often. This can’t be expressed enough. An on demand resource of invaluable knowledge is a fortunate gift to have. It will only improve your work and engagements are generally positive as the common ground after all, is software and technology. Shared interests blend together.

6 Acknowledgements

First and foremost, I would like to thank my project supervisor, Joseph Kehoe, for his time and patience with my regular updates and not always concise series of questions sent to him throughout the year relating to scope and direction that he gladly responded with no hesitation. And of course, for never being a dull presence!

I would like to thank the other 4th year lectures for clarifying some issues that were related to their field and for any observations made by them throughout the year, not even specifically related to this project but in handling other subject matters that would've otherwise interfered with the project itself.

I would also like to extend my thanks to some of my friends and family for being supportive when I refused to stop talking about what I am working on. Not to mention constant encouragement and motivation to consider the bigger picture and not lose sight when things got tough.

And so, the journey has come to its end.