

**Preliminary Report**

**Home Energy Management System (HEMS)**

by

Jack Harding

This Report is submitted in partial fulfilment of the requirements of the Honours Degree in Electrical and Electronic Engineering (DT021A) of the Dublin Institute of Technology

8 February 2019

Supervisor: John Dalton

List of Abbreviations

HEMS Home Energy Management System

HMS Home Management System

MQTT Message Queuing Telemetry Transport

API Applied Programming Interface

IoT Internet of Things

HAN Home Area Network

OH2 OpenHAB2

# Introduction

A study from consulting firm McKinsey found that in modern data centres only six to twelve percent of the total electrical power used was on computation-the rest being spent on standby. This gross waste of energy along with many modern households having the ability to generate green energy and run relatively independently from the grid, requires the need for a system to manage when and how electrical power comes into the home. [1]

Recently, there has been a surge in the number of smart home devices, due to the price of electronics reducing significantly, the increase has encouraged consumers to buy into the many of the home management systems available. Much of these systems provide connectivity between multiple platforms including user account data which gives the HMS the ability to tailor the needs of the user based on search and location history as well as passwords. HMS’s also offer voice control which is revered in the modern technology trends, this changes the ease at which commands can be given and expands the range of what can be done.

A solution could be to combine a consumer-focused HMS with an energy management system to give the user the information and control over their energy. The HEMS is required to collect data from the array of sensors throughout the home, this data can be given to the user to determine how proceed in saving energy. Another option may be to integrate it with an HMS (Alexa, Google Home) and it be controlled automatically from the cloud, this would allow for more intelligent usage of resources and might benefit the grid to have a large set of energy data, but at the cost of data privacy.

# Objective

The objective of this report is to design and build a prototype that monitors and controls both energy consuming and producing devices. This platform will provide energy management for Prosumers; who both produce and consume energy, it will be based upon a Raspberry Pi microcontroller with a framework to handle basic home devices as well as energy generation devices such as PV panels and wind turbines. The HEMS is to run alongside a Home Management System (HMS) such as Amazon Alexa and assist with the management of energy in a home.

The monitoring of the devices is to be done with multiple sensors connected via some wireless protocol across the environment, they will be measuring the power used and state each device, this information will be relayed to a common gateway. This is where much of the analysis will be done, the usage among devices needs to be categorised into ones that can have energy saved and devices that require constant power e.g. fridge, modem. The data received will allow the energy generated by the onsite renewables, to be used intelligently to maximise self-consumption. This can be achieved by using onsite batteries to store the energy generated when the wind blows and sun shines so that when the consumers needs to use the electricity, typically in the evening-its ready to go.

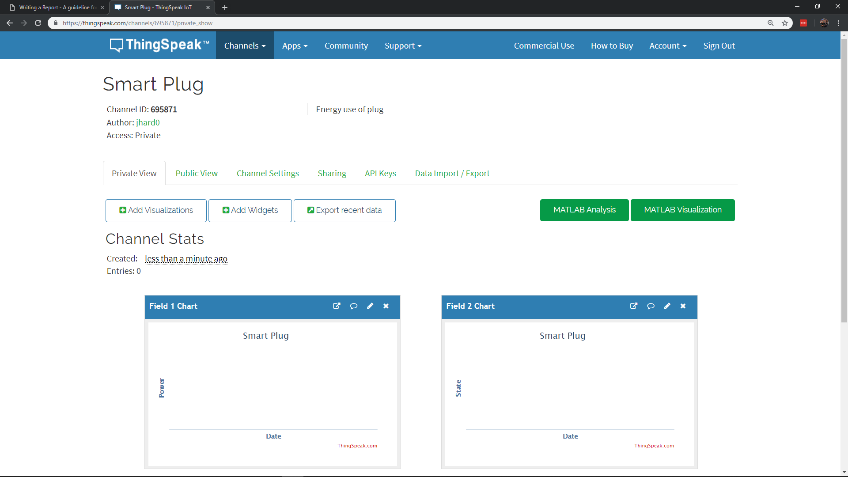
# Ethics

One of the main concerns of a home energy management system is data privacy. The idea of a service that monitors the usage and states of much of the devices in a private home could cause some potential users to be deterred especially when remote access is permitted. This could be the case for a “disconnected system” one where the energy management is done without interference from the service provider. This would offer

## Cloud/Persistence

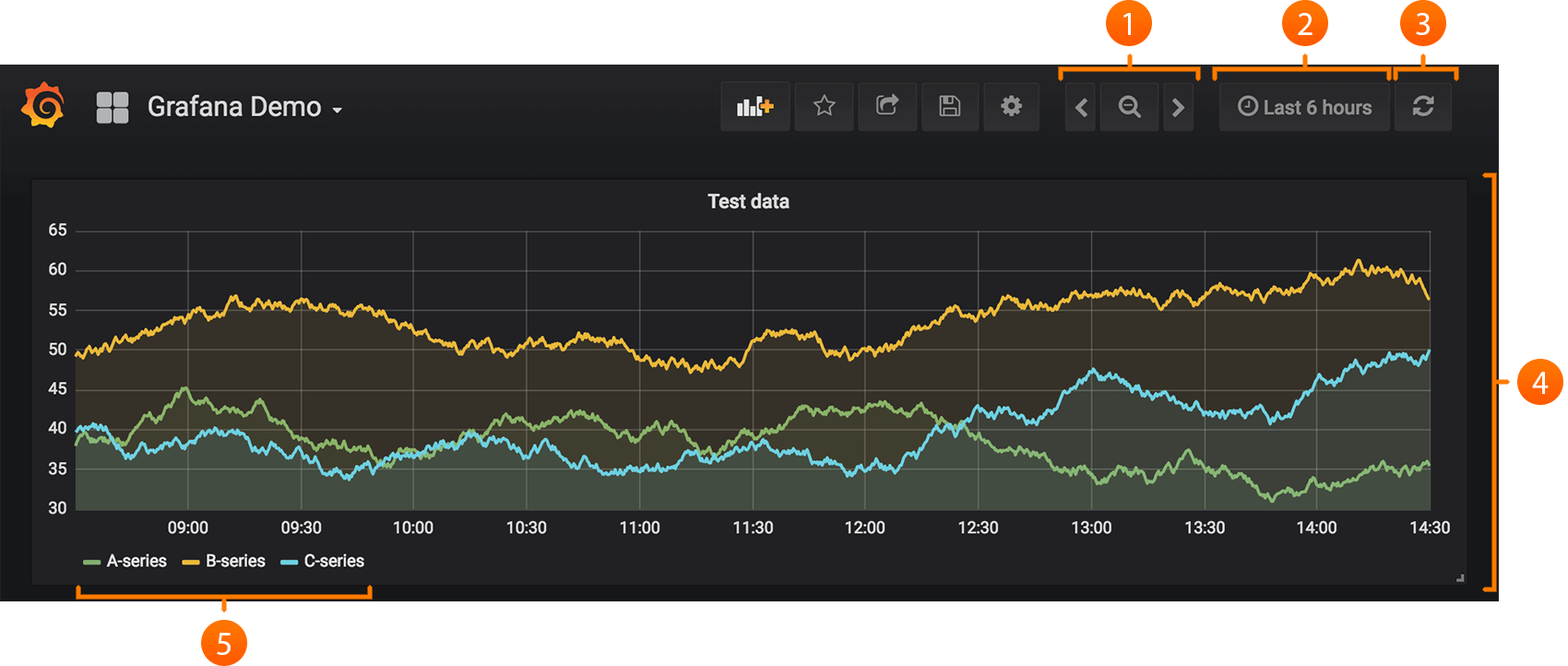
### ThingSpeak

One of the options for hosting the information online is ThingSpeak, an IoT platform that allows analysis, visualisation and storage of data collected from sensors, sent via Arduino or Raspberry Pi. This platform allows data logs from sensors to be used in the MATLAB environment, making complex analysis of historical data possible. To start, a channel must first be created, this contains data, location, and state fields.

This data can then be graphed to visualise the usage of a certain appliance in this case and adjust, to lower power usage. The first channel created is the Smart Plug, it is connected via Z-Wave and has two fields; state (is device on/off) and Power (how much has been used so far). ThingSpeak offers a MQTT API, this gives the ability to subscribe and publish to specific topics, broadening the capability of the HEMS.

### Influx & Grafana

InfluxDB is an open source time series database used for engineering applications for its high accuracy timing (within nanoseconds) and the primary key is time, so a timestamp comes with every entry into the database by default. The Time-Structured Merge Tree (TSM) was an engine made in 2015 to increase compression, throughput and query speed.

Grafana is also open source and is used for visualising time series data for sensors, weather applications and home automation. The Grafana dashboard is shown above, the period of graphs can be changed when needed and it can have multiple graphs and alerts, to let the user know about certain areas of a home requiring special attention.

## Gateway

The gateway is used as a central hub and a point where the devices such as sensors connect to the cloud, this gateway usually has a way of organising the data before uploading, this usually includes some form of encryption to ensure the security of that HAN. The choice of gateway is very important when it comes to usability, security and reliability.

### HomeSeer

HomeSeer is essentially a Raspberry Pi with proprietary software installed for ease of configuration. The HomeSeer offers straightforward connection to Z-Wave devices with its UZB transceiver, the dashboard is available after initial setup through a web browser, password is set and remote access for the gateway can be enabled too. This option offers both Android and iOS apps to control devices on the Z-Wave network. Although it does offer plugins like OpenHAB does, they are very limited and can only have a maximum of five plug-ins at a time, many of them aren’t free either.

### Raspberry Pi

This option allows for the highest degree of flexibility, it gives the choice of using OpenHAB or HomeAssistant and still has the full functionality of using a Raspberry Pi with all its libraries available to be used in conjunction with the project as well it being open source. The Pi’s allows for multiple wireless protocols to be used also with Zigbee, Z-Wave, and many others available. The Pi can be installed with either Raspbian or OpenHABian, the latter is an OS made by OpenHAB specifically for home automation and allows easy set up but at the cost of flexibility, Raspbian is the recommended OS with the capability of installing NodeJS for asynchronous events or the use of Python and its extensive range of libraries.

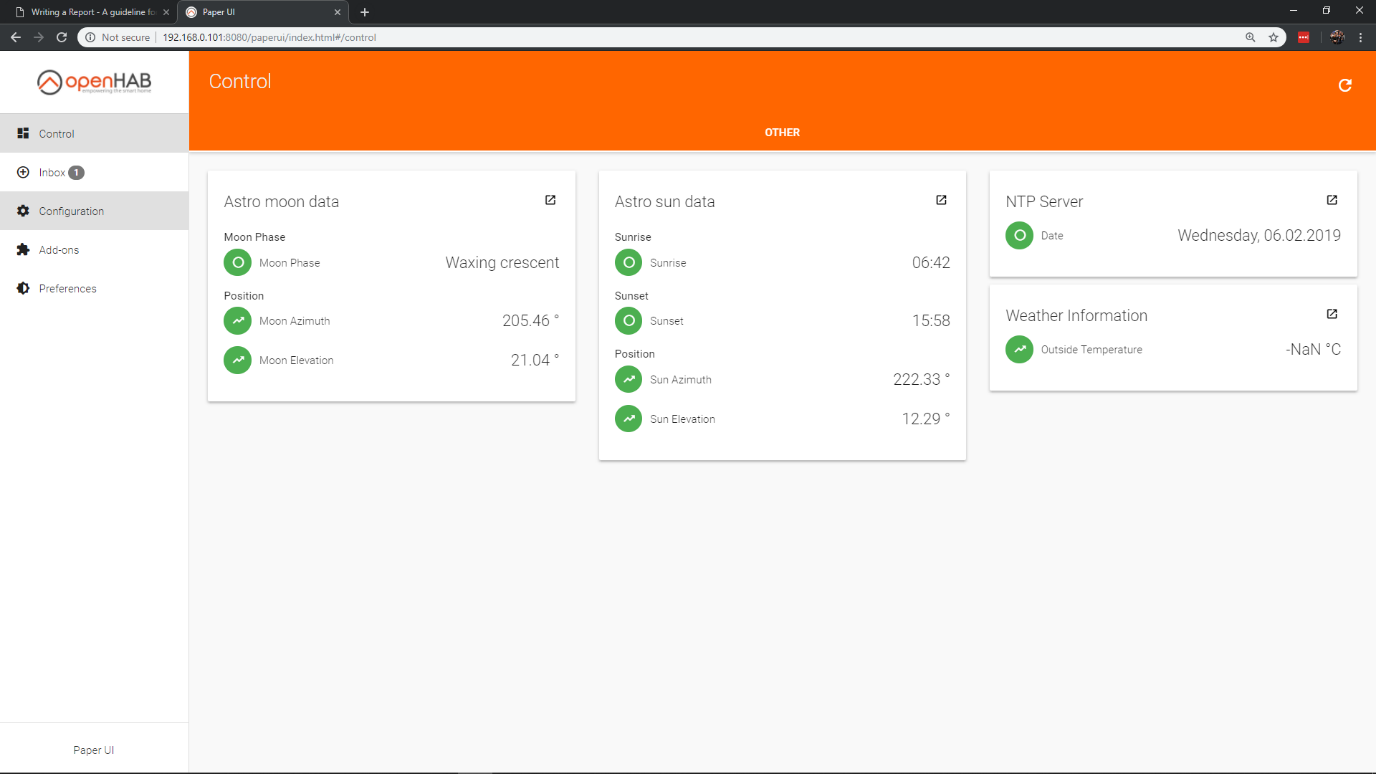
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mandatory** |  | **Desirable** |  |  | **HAN** |  |  |  |  |  |  |
| **Platform** | **Third party software ('apps') on Hub, or on low-latency real-time connection with Hub** | **Remote Software Management ('apps market place'), e.g. iTunes, Play Store.** | **Plug & Play pairing (if no missing software on Hub)** | **Automatic download of missing software during pairing (Control Protocols such as X10, Thingy 'drivers')** |  | **Ethernet** | **PLC** | **RS-485** | **Z-wave** | **Wi-Fi** | **Bluetooth** | **Zigbee** |
| **Samsung SmartThings** | N | N, maybe planned | N |  |  | y |  |  | y |  |  | y |
| **Apple HomeKit** | Y, no controls in portal | Y |  |  |  |  |  |  |  | y | y |  |
| **Somfy (DT-10 favourite)** | N | N |  |  |  |  |  |  | y |  |  |  |
| **Control4** | N | 4Store, no longer supported | Y | Y |  |  |  |  |  |  |  |  |
| **Creston** | N | Y |  |  |  | y |  | y |  | y |  |  |
| **HomeSeer (Hometroller)** | Y | Y (software updater) |  |  |  |  |  |  |  |  |  |  |
| **Iris** | No | N | N | N |  | y |  |  |  |  | y |  |
| **Vera** | Y (can run without internet) | y |  |  |  |  |  |  | y | y | y | y |
| **Nest** | Y, can run local some feature | Y | Y |  |  |  |  |  |  | y |  |  |
| **HomeAssistant** | Y | Y, with Watchtower app | Sometimes | N |  | y | y |  | y | y | y | y |
| **Hive Hub 360** | N | Y |  |  |  |  |  |  |  | Y | y | Y |

## HMS

The Home Management of the HEMS is essential to the operation of the system, it determines how the devices are connected and where they are connected to. There are many options out there, including the ones in the evolution matrix above, but the list has been narrowed to three options, OpenHAB and HomeAssistant for the Pi or the proprietary one of HomeSeer.

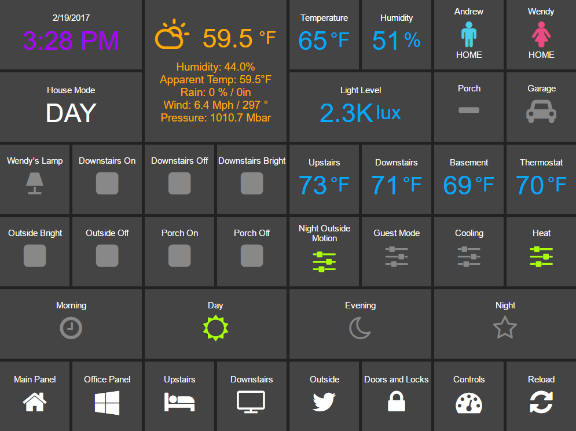
### OpenHAB

Manufacturers create smart devices to help automate the home but lacks something amalgamate capabilities, OpenHAB allows for integration of multiple devices and gives the user full control of their data, it is written in Java and based on the Eclipse SmartHome Framework. OpenHAB has the option of having an intranet approach, where only those on the network could have access to any of the devices, remote access is also an option. Below shows a typical OpenHAB2 dashboard.

 A core concept is “items”, this is essentially to interface the device with certain attributes so if it is changed to another (physical or virtual) it will work as before. There’s a vast library of bindings built into OpenHAB, including Z-Wave, InfluxDB and many more, these remove the need to develop software to allow different languages and frameworks to function.

### HomeAssistant

HomeAssistant is installed as a standalone operating system like OpenHABian and has all the capabilities of OH2 but with some advantages and drawbacks. HomeAssistant is written in Python 3 rather than Java and has a far faster development cycle allowing for quicker integration of new items but at a cost of stability.

 [x]

## Wireless Protocol

### Z-Wave

### Zigbee

# References

[1]https://www.mckinsey.com/~/media/mckinsey/dotcom/client\_service/sustainability/pdfs/a\_compelling\_global\_resource.ashx

<https://docs.influxdata.com/influxdb/v1.7/concepts/time-series-index/>

<https://www.openhab.org/about/who-we-are.html>

<https://appdaemon.readthedocs.io/en/latest/DASHBOARD_INSTALL.html> -HomeAss

<https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/>

Connectivity for HEMS pdf

Smart Homes Automation Z-Wave pdf

Review of Communication Zigbee pdf

Excel SpreadS

Respectful IoT

GDPR Iot pdf