

# **Design and implementation of a dual WiFi/cellular linked smart energy meter**

Final Report

**J Wagener**  
16056214

Submitted as partial fulfilment of the requirements of Project EPR400  
in the Department of Electrical, Electronic and Computer Engineering

University of Pretoria

November 2022

Study leader: Dr. JH van Wyk

## Part 1. Preamble

This report was compiled to outline the research and production process of developing a non-intrusive load detection energy meter for appliance identification (this device being my final year project submission).

### *Project proposal and technical documentation*

This main report contains an unaltered copy of the approved Project Proposal (as Part 2 of the report).

Technical documentation appears in Part 4 (Appendix).

All the code that I developed appears as a separate submission on the AMS.

### *Project history*

The work of Gouhua *et al* in [1] was used as a basis and point of departure for this project. However, though the method of identification used is the same as that of Gouha *et al*, the implementation of their work in the creation of the device appears to be unprecedented and original. This project makes extensive use of existing algorithms on decision trees - many of which were adapted from [2], though other authors' work was also relied upon (and appropriately cited).

I reused FFT code from a library and did not develop this myself.

The FIR-Filter design (Type,Window,Order) is my own. I reused the FIR implementation code from a library and did not develop this myself.

I reused HTML framework code from a library and did not develop this myself.

The rest of the work reported on here, is entirely my own.

### *Language editing*

This document was edited by a qualified language editor and writing guide. By submitting this document in its present form, I declare that this is the written material that I wish to be examined on.

My language editor was \_\_\_\_\_ **Daniëlle van der Watt**.



Language editor signature

2022 - 11 - 07

Date

*Declaration*

I, Jonathan Wagener understand what plagiarism is and have carefully studied the plagiarism policy of the University. I hereby declare that all the work described in this report is my own, except where explicitly indicated otherwise. Although I may have discussed the design and investigation with my study leader, fellow students or consulted various books, articles or the internet, the design/investigative work is my own. I have mastered the design and I have made all the required calculations in my lab book (and/or they are reflected in this report) to authenticate this. I am not presenting a complete solution of someone else.

Wherever I have used information from other sources, I have given credit by proper and complete referencing of the source material so that it can be clearly discerned what is my own work and what was quoted from other sources. I acknowledge that failure to comply with the instructions regarding referencing will be regarded as plagiarism. If there is any doubt about the authenticity of my work, I am willing to attend an oral ancillary examination/evaluation about the work.

I certify that the Project Proposal appearing as the Introduction section of the report is a verbatim copy of the approved Project Proposal.

Jonathan Wagener  
J Wagener

2022 - 11 - 07

Date

## TABLE OF CONTENTS

---

<b>Part 1. Preamble</b>	<b>i</b>
<b>Part 2. Project definition: approved Project Proposal</b>	<b>vii</b>
1. Project description	
2. Technical challenges in this project	
3. Functional analysis	
4. System requirements and specifications	
5. Field conditions	
6. Student tasks	
<b>Part 3. Main Report</b>	<b>xiv</b>
<b>1 Literature study</b>	<b>1</b>
<b>2 Approach</b>	<b>5</b>
<b>3 Design and implementation</b>	<b>7</b>
3.1 Design summary	7
3.2 Theoretical analysis	8
3.3 Simulation	18
3.4 Hardware design and implementation	23
3.5 Software design	28
3.6 Software implementation	39
3.7 Final system integration and testing	40
3.8 Statistical analysis	41
<b>4 Results</b>	<b>44</b>
4.1 Summary of results achieved	44
4.2 Tools developed for qualification testing	45

4.3 Qualification tests	46
<b>5 Discussion</b>	<b>56</b>
5.1 Interpretation of results	56
5.2 Critical evaluation of the design	56
5.3 Design ergonomics	57
5.4 Health, safety and environmental impact	58
5.5 Social and legal impact of the design	59
<b>6 Conclusion</b>	<b>60</b>
6.1 Summary of the work completed	60
6.2 Summary of the observations and findings	60
6.3 Contribution	60
6.4 Future work	61
<b>7 References</b>	<b>62</b>
<b>Part 4. Appendix: technical documentation</b>	<b>64</b>
<b>HARDWARE part of the project</b>	<b>65</b>
Record 1. System block diagram	65
Record 2. Systems level description of the design	65
Record 3. Complete circuit diagrams and description	65
Record 4. Hardware acceptance test procedure	66
Record 5. User guide	67
<b>SOFTWARE part of the project</b>	<b>68</b>
Record 6. Software process flow diagrams	68
Record 7. Explanation of software modules	68
Record 8. Complete source code	68
Record 9. Software acceptance test procedure	69
Record 10. Software user guide	69

**EXPERIMENTAL DATA** **70**

Record 11. Experimental data 70

## LIST OF ABBREVIATIONS

---

<b>ELF</b>	Electric load forecasting
<b>ENOB</b>	Effective number of bits
<b>RMS</b>	Root mean squared
<b>PRU</b>	Programmable real-time unit
<b>LED</b>	Light-emitting diode
<b>ID3</b>	Iterative Dichotomiser 3
<b>SPI</b>	Serial Peripheral Interface
<b>NILM</b>	Non-Intrusive Load Monitoring

## **Part 2. Project definition: approved Project Proposal**

This section contains the problem identification in the form of the complete approved Project Proposal, unaltered from the final approved version that appears on the AMS.

For use by the Project lecturer	Approved	Revision required
<b>Feedback</b>	 <div style="border: 1px solid green; padding: 5px; text-align: center;">Approved</div>	

To be completed by the student					
<b>PROJECT PROPOSAL 2022</b>			Project no	JHvW9	Revision no
Title Mr	Surname <b>Wagener</b>	Initials <b>J</b>	Student no 16056214	Study leader (title, initials, surname) Dr JH van Wyk	
Project title <b>Design and implementation of a dual WiFi/cellular linked smart energy meter</b>					

Language editor name Jonathan Wagener	Language editor signature 
<b>Student declaration</b> I understand what plagiarism is and that I have to complete my project on my own.	<b>Study leader declaration</b> This is a clear and unambiguous description of what is required in this project. Approved for submission (Yes/No)
Student signature 	Study leader signature and date  28 JULY 2022

## 1. Project description

What is your project about? What does your system have to do? What is the problem to be solved?

With the growing popularity of connected devices and services, and amidst consumers' rising concerns of re-sustainable energy usage, there has been a growing desire for more information about the energy consumers use. Current "smart" energy meters which are accessible to consumers are limited as they are only able to report the total amount of energy consumed by all the devices connected to the circuit. If a consumer wanted to measure the energy consumption per load on the same circuit simultaneously it would require the use of multiple "smart" energy meters (one per load). The goal of this project is to design a single energy meter that can measure per device energy consumption on the same circuit when multiple loads are simultaneously in operation.

The system will have to identify individual loads using the energy profile/fingerprint of the individual loads when multiple loads are simultaneously in operation. This will then be used to calculate the energy consumed per load. The energy consumption data will be accessible to a user via a web page accessible over WiFi or cellular interface.

## 2. Technical challenges in this project

Describe the technical challenges that are *beyond* those encountered up to the end of third year and in other final year modules.

### 2.1 Primary design challenges

The main design challenge is designing an algorithm that can identify individual loads using the energy profile/fingerprint of the individual loads when multiple loads are in operation simultaneously. The design of this algorithm also needs to account for devices (such as dishwashers and washing machines) that go through cycles and, as a result, the energy profile/fingerprint and power usage of those devices is not always constant.

Many loads in households require little power to operate (LED lights) while some require large amounts of power to operate (kettles). The design challenge is the creation of a sampling subsystem that can account for this vast range of current draw while maintaining SNR at the low end. The subsystem needs to measure this small current draw signal reliably.

### 2.2 Primary implementation challenges

- 1) A live 230V service is dangerous – thus correct isolation is required. Safety measures should be implemented to ensure the safety of both the user and to protect the device. Should a failure occur, dangerous voltages and current should not be permitted to transfer to the rest of the device.
- 2) The target user of this device is likely not to have in-depth knowledge of energy or its measurement, so it should be intuitively usable and the data should be presented in such a way that laypersons will understand and interpret it.
- 3) Interfacing the machine learning model on an embedded platform to identify devices from the sampled data may present certain challenges.

## 3. Functional analysis

### 3.1 Functional description

Describe the design in terms of system functions as shown on the functional block diagram in section 3.2. This description should be in *narrative format*.

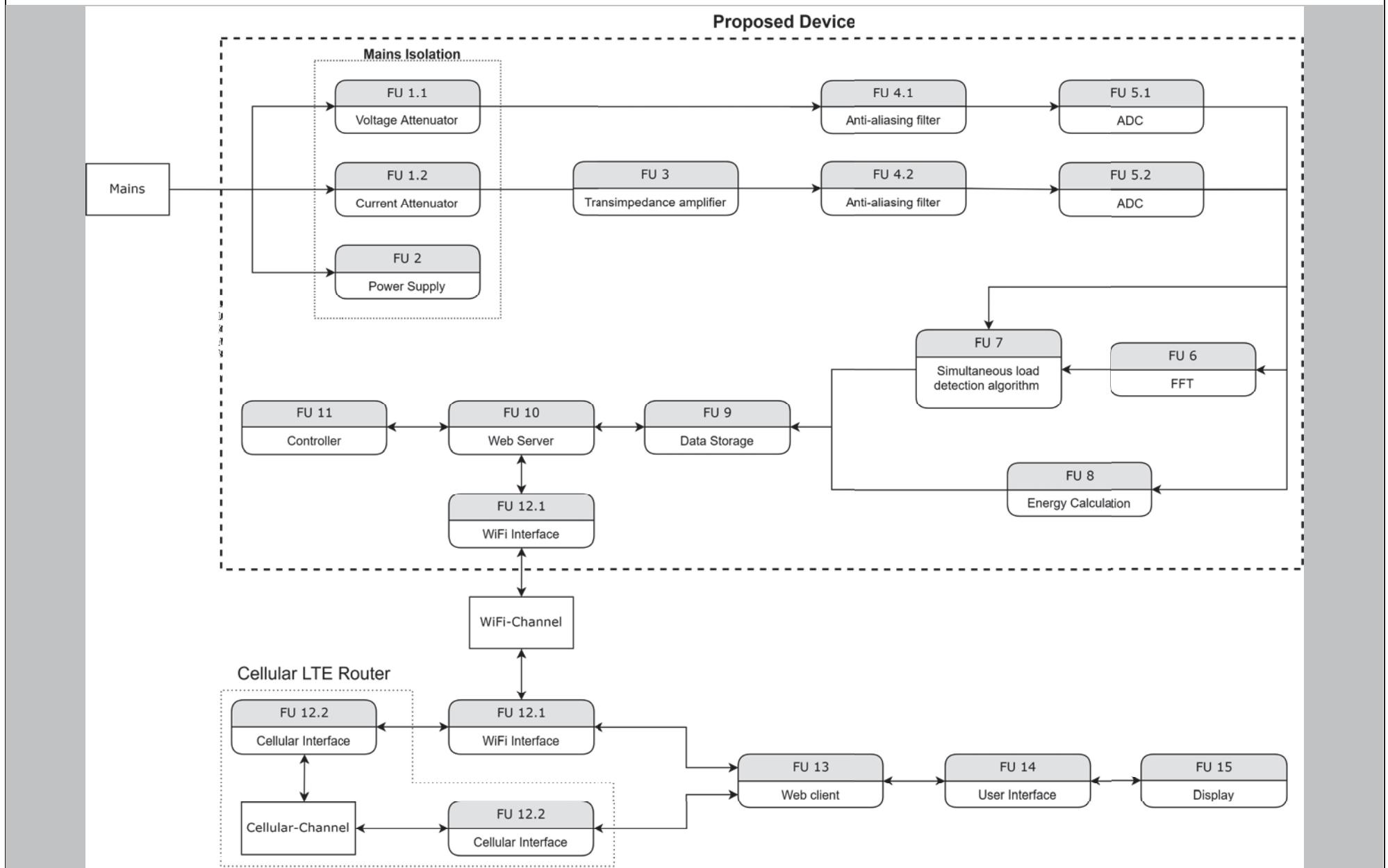
The device is connected to mains via a voltage attenuator (FU 1.1) and the current attenuator (FU 1.2) as well as a power supply (FU 2).

The voltage and current attenuators isolate the voltage and current, and convert it to levels that can be used to obtain the voltage and current measurements. The voltage signal is then passed through an anti-aliasing filter (FU 4.1) before it is sampled by an ADC (FU 5.1). The attenuated current signal follows the same steps as the voltage signal, but it is first converted to a voltage signal using a transimpedance amplifier (FU 3), then passed to the AA Filter (FU 4.2), and then sampled by an ADC (FU 5.2). The sampled voltage and current data is then passed through an FFT (FU 6).

Each of the loads that the device is designed to detect has a unique energy fingerprint. Functional Unit 7 (FU 7) is an algorithm which has been trained via machine learning to use the aforementioned energy fingerprints to detect which loads are simultaneously consuming energy. This is done by feeding the voltage (FU 5.1), current (FU 5.2), and FFT (FU 6) data into the model. FU 7 then processes this data and produces an output array containing the probabilities relating to the loads actively consuming energy at that moment in time.

The energy currently being consumed per identified load is calculated (FU 8). The consumed energy (FU 8) data and device identification (FU 7) data is then stored in a database (FU 9) which is then accessible through a web server (FU 10) and controlled by a controller (FU 11). The web server (FU 10) is accessible through a WiFi interface (FU 12.1) which is also connected to a cellular interface via an external WiFi Cellular Router - the latter also has a cellular interface (FU 12.2). The web client (FU 13) allows the user to interact via a user interface (FU 14) - the user can view the resulting data through a display (FU 15).

### 3.2 Functional block diagram



## 4. System requirements and specifications

These are the core requirements of the system or product (the mission-critical requirements) in table format IN ORDER OF IMPORTANCE. Requirement 1 is the most fundamental requirement.

	<b>Requirement 1: the fundamental functional and performance requirement of your project</b>	<b>Requirement 2</b>	<b>Requirement 3</b>
<b>1. Core mission requirements of the system or product.</b> Focus on requirements that are core to solving the engineering problem. These will reflect the solution to the problem.	Simultaneously and correctly differentiate between any combination of 5 - 10 devices.	Comply with the requirements stipulated by the IEC/AS 62053-21 to be classified as a class 2 energy meter.	Comply with the IEC 60038 Voltage Standards
<b>2. What is the target specification</b> (in measurable terms) to be met in order to achieve this requirement?	The device must be able to distinguish any combination of 5 - 10 household appliances that can be turned on simultaneously at unpredictable times, with a true positive rate (TPR) of 95%	The error in the power measurement of any combination of the loads that draw between 1.2 - 40 A connected should not exceed that of $\pm 2.5\%$ .	Able to measure a voltage in the range of 230V $\pm 10\%$ at 40 A (Most homes do not use more than 40 A at any one moment)
<b>3. Motivation:</b> how or why will meeting the specification given in point 2 above solve the problem? (Motivate the specific target specification selected)	A TPR of 95% will ensure that the correct loads are detected 28/30 times per minute, meaning that an incorrect detection lasts 2-4 seconds. The impact of a false positive identification on the total energy reported per load is, thus, negligible.	Meeting the above specifications will ensure that the energy measurement device is recognized as a class 2 energy meter and that it meets or exceeds the current standard for residential applications.	The voltage line can vary between 207 - 253 Vrms. This means that the device needs to be able to measure over this range to comply with the above specification.
<b>4. How will you demonstrate at the examination</b> that this requirement (point 1 above) and specification (point 2 above) has been met?	The device will be connected to multiple loads in random unknown states. The device will then be powered on and report the states of the connected loads. The states of the loads can then be altered at any point which will be reflected by the device	The combined energy measurement of the multiples of different device types connected will be confirmed to not exceed an error of $\pm 2.5\%$ regardless of the connected devices impedance.	The device will be set up over a number of days to monitor the power line. This will ensure that there is enough voltage variation to confirm that the device can tolerate the voltage fluctuations common in residential supplies.
<b>5. Your own design contribution:</b> what are the aspects that you will design and implement yourself to meet the requirement in point 2? If none, remove this requirement.	The creation, implementation, training and deployment of the machine learning model used to identify loads that are consuming energy simultaneously.	Interfacing and accounting for tolerance errors in the mains interface and the ADC sampling stage.	The attenuation and sampling stage will be designed to meet or exceed the specifications by selecting the correct components.
<b>6. What are the aspects to be taken off the shelf</b> to meet this requirement? If none, indicate "none"	External device data sets will be used to develop the device fingerprinting. The embedded platform. The FFT algorithm.	Appliance/load with known and or measured energy consumption parameters.  Discrete time based energy consumption formula.	The voltage and current interface devices will be taken off the shelf as well as the ADC.

## System requirements and specifications page 2

	<b>Requirement 4</b>	<b>Requirement 5</b>	<b>Requirement 6</b>
<b>1. Core mission requirements of the system or product.</b> Focus on requirements that are core to solving the engineering problem. These will reflect the solution to the problem.	The system should be able to identify loads with low energy consumption.	The change in energy consumption of any connected device should be reflected in near realtime.	
<b>2. What is the target specification</b> (in measurable terms) to be met in order to achieve this requirement?	The device must be able to detect a load that consumes more than 2W of power.	The status of the energy consumption of any connected load should be updated in under 2 seconds.	
<b>3. Motivation:</b> how or why will meeting the specification given in point 2 above solve the problem? (Motivate the specific target specification selected)	Identifying loads that draw little power will allow the system to identify devices that are left on but, because they draw such little power, are often overlooked. This is problematic as such devices will still add to energy consumption over time.	Updating the energy consumption of any active device in under 2 seconds will result in an accurate energy meter that is able to allow a user to recognise active loads and their energy consumption near instantly.	
<b>4. How will you demonstrate at the examination</b> that this requirement (point 1 above) and specification (point 2 above) has been met?	The system will be connected to a load that is known to draw 2W of power along with other larger loads that are on simultaneously. The system will need to correctly identify and report the energy consumption of this load.	Multiple devices will be powered on. A load with the ability to alter its power draw will be monitored. (e.g the power draw of a hair dryer) The device will report the change in power draw in under 2 seconds.	
<b>5. Your own design contribution:</b> what are the aspects that you will design and implement yourself to meet the requirement in point 2? If none, remove this requirement.	The detection and power measurement methods.	The detection algorithm.	
<b>6. What are the aspects to be taken off the shelf to meet this requirement?</b> If none, indicate "none"	The 2W load.	The clock source.	

## 5. Field conditions

These are the REAL WORLD CONDITIONS under which your project has to work and has to be demonstrated.

	Field condition 1	Field condition 2	Field condition 3
<b>Field condition requirement.</b> In which field conditions does the system have to operate? Indicate the one, two or three most important field conditions.	The system should be isolated from all other electrical loads which the system is not trained to detect.	The system should not be overloaded.	
<b>Field condition specification.</b> What is the specification (in measurable terms) for this field condition?	Only the specified loads which the system is designed to identify should be connected.	The combinations of loads connected to system should not exceed that of 8000W.	

## 6. Student tasks

### 6.1 Design and implementation tasks

List your primary design and implementation tasks in bullet list format (5-10 bullets). These are *not* product requirements, but *your* tasks.

- The design of a voltage attenuator.
- The design of a current attenuator.
- The design of a data logger subsystem to create data sets with known appliances.
- The design and training of a machine learning algorithm to detect individual appliances.
- The implementation of a machine learning algorithm on an embedded platform.

### 6.2 New knowledge to be acquired

Describe what the theoretical foundation to the project is, and which new knowledge you will acquire (*beyond* that covered in any other undergraduate modules).

- Require proficiency in professional level PCB design for final product to ensure a high quality device is delivered.
- Machine learning and model implantation on micro-controllers to correctly identify devices.
- Understanding and evaluating performance of identification models.
- Understanding safety standards (SANS, ISO, etc.) and designing the product accordingly.
- Professional documentation of the design process, as well as documentation regarding device usage.
- Modular component design to allow for simple integration and overall design.

## **Part 3. Main Report**

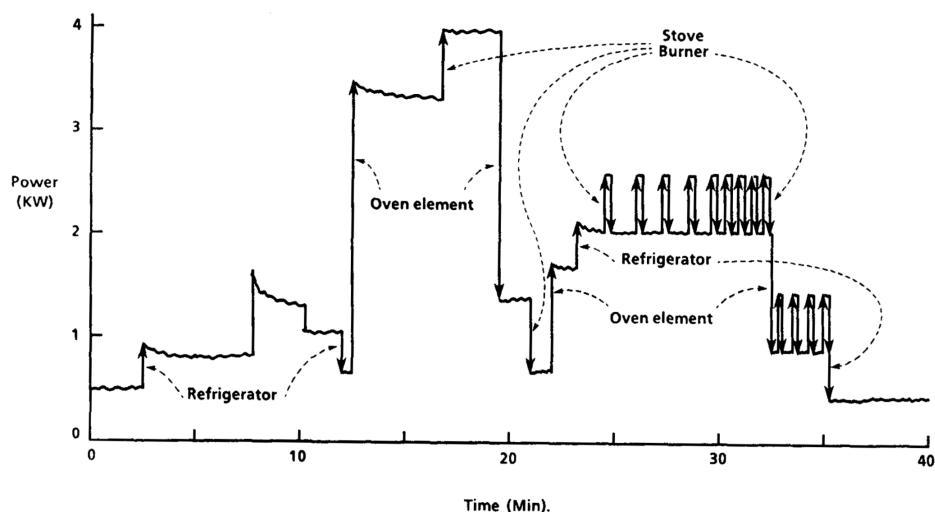
## 1. Literature study

---

An energy meter is a device used to measure the energy consumption of the active loads on a circuit. Most energy meters provide the total energy used, but do not provide per-load energy consumption. This information can be important to both consumers and energy providers. Consumers are able to better understand how they consume energy. The desired granularity of how the total energy is consumed is only achievable in three ways: first, the actual device can report on its own energy consumption; secondly, an energy meter may be installed for each device; and, finally, Non-Intrusive Load Monitoring (NILM) could be used.

Energy providers are able to manage supply and demand more effectively which is necessary for the functionality of any energy grid [3]. Using modern prediction algorithms to analyse the granular information regarding how consumers use energy throughout the day can improve the accuracy of electric load forecasting (ELF), thus enabling energy providers to manage (and meet) the energy supply and demand more effectively [4].

All electrical loads will produce a rising or falling edge when there is a change to their working state - this is indicated by the arrows pointing upwards or downwards in Figure 1. The state between the two transient edges is the steady state [5]. Features that define a change to the working state can be defined as transient signatures and steady state signatures [6]. In the 1980s, George Hart [7] presented a method of load identification which allowed him to distinguish between multiple loads with one energy monitoring device. He pioneered the term Non-Intrusive Load Monitoring (NILM). Hart's method made use of the transients or (in his words) "step changes".



**Figure 1.**  
Examples of load signatures of household appliances as illustrated in [7].

As shown in Figure 1, Hart's method is only able to identify loads when a transition event that causes a significant change in the power draw is detected. The method's failure to make use of the steady state to identify loads reduces its effectiveness.

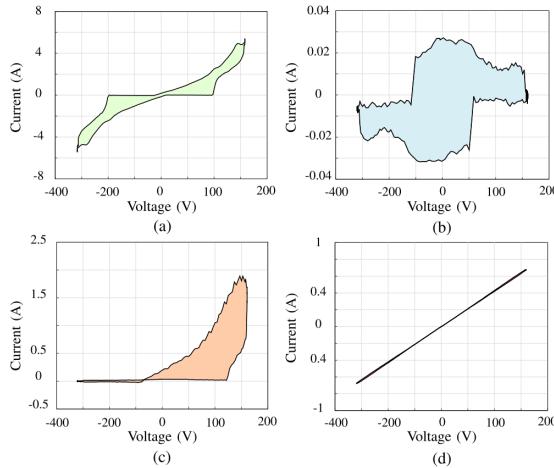
Hart's original system was later improved in a number of ways. The first refinement resulted from innovation in data acquisition systems used for energy monitoring which has allowed for smaller changes in power to be measured at an increased sample rate. Most modern off-the-shelf energy monitoring systems make use of analogue-to-digital converters (ADCs) that have a resolution exceeding 10-bits [8]. This is one of the most notable improvements. The increase in ADC resolution enables an energy monitor to measure a change in power exceeding 11.23 W, with some energy monitors being able to measure a change as low as 0.17 W.

Another improvement to the data acquisition systems was an increase in the sample rate - this has resulted in more harmonic information to identify unique loads [9]. In order to distinguish between 20 - 40 loads, a sample rate of between 10 – 40 kHz is required [9]. This allows for the identification of small power draw loads that would not have been detectable using Hart's original method.

Saleh makes a case in favour of using specialised mathematical transforms for transient feature extraction - such as the Huang Hilbert Transform (HHT) with Ensemble Empirical Mode Decomposition (EEMD) ([10]). While the author persuasively argues that this is an accurate and effective method of feature extraction, he does not expand much on the practical implementation of his method.

The steady state can be disaggregated into many features that are unique to the particular state of a load. Some of the most common features include power variation, and time and current harmonics. These steady state features are currently being researched and tested [11]. One of the most notable features that has recently been studied is the V-I trajectory feature.

V-I trajectory allows for the clustering of similar load signatures in feature space. This is accomplished by projecting the instantaneous voltage and current signals to an XY-cartesian plane over one complete cycle. In Figure 2, some loads from the Controlled ON-OFF Loads Library (COOLL) are mapped in this way. The figure shows that every load's projection is distinct.



**Figure 2.**

Examples of load projection from the COOLL data set as illustrated in [11].

Mulinari *et al* integrated the features that came about as a result of the new feature extraction techniques into three existing detection algorithms (Ensemble, kNN and SVM) ([11]). They recorded that the accuracy of the detection methods could vary from between 70 % and 99.64 % when identifying the loads contained in the COOLL dataset. The most significant result was that - even though all the features (existing and new) were combined by the detection algorithm - the accuracy of the prediction never fell below 96.35 %.

The detection methods used by Mulinari *et al* ([11]) are not the only algorithms that have been used in NILMs. Methods which use Hidden Markov models are still actively being researched. Two papers using Hidden Markov models were recently published in 2015 and 2018 [5, 12]. The Hidden Markov model (the original method used by Hart) uses statistical predictions to identify the current state of a load [7]. As was the case in the original work of Hart, the results presented by Zhihao *et al* [5] and Jung *et al* [12] also only pertained to loads that caused significant power draw changes when a change in operation occurred. The research did not make any predictions or comment on the effectiveness of the device in a situation where it was used with loads which cause nominal power changes. The effectiveness of the device is given in Bit-Error-Rate (BER). However, the authors do not provide any insight as to how to relate the BER to allow for a comparison to be drawn with the other methods discussed in the literature. The usefulness of these results as a benchmark against which other methods can be compared is therefore limited.

The research of Zhihao *et al*, [5] and Jung *et al*, [12] was focused solely on theoretical application and only tested in simulations. Mubarok *et al*, [13] conducted similar research with the focus being on practical implementation. Their aim was to develop an algorithm that was simple to implement on an embedded processing platform. Mubarok *et al* implemented a method referred to as the radial basis function - this method is used to derive an unknown function from a set of inputs and known outputs. The authors use the FFT of the current signal as the input to the radial basis function. Once the function has been derived, it is implemented on the STM32F407VGT6 embedded platform. The implemented radial basis function was able to identify the loads with an error margin of merely 0.14 % to 4.84 %. These results demonstrated that the practical radial basis function was able to detect loads accurately in most instances with an error rate that was commendably low. These findings were limited as the function needs to be trained for every possible combination comprised of the unique loads, and these combinations increase exponentially. In other words, if the number of unique loads was doubled, the resulting number of combinations would be 256. The radial basis function is an approximation function which uses a large number of linear matrix operations in order to make a prediction. Consequently, it has the disadvantage of being slowed down by the number of combinations due to the complexity cost being between  $O(n^2)$  and  $O(n^3)$  (with n being the number of samples).

A recent paper published in 2020 by Guohua *et al* [1] may provide a solution for this problem. They propose using a random forest algorithm - an algorithm which is commonly used to overcome classification and regression problems in a machine learning context. Unlike the radial basis function, the decision tree only makes use of a number of checks - no mathematical operations are performed when it makes its prediction. Here the complexity cost is  $O(n \log(n))$ . This is substantially lower than that of the radial basis function. Unfortunately, this method resulted in a slight decrease in accuracy. Using the same number of loads as

Mubarok *et al* as well as the large power loads used in [11, 5, 12], the resulting largest measured error margin was 14 % [13]. As with the research done by Mubarok *et al*, the use of a random forest algorithm in NILM has also only been simulated, its practical implementation not yet having been tested [13].

Non-Intrusive Load Monitoring has seen significant improvement since Hart's original design in the 1980s [7]. Most of these improvements and related innovations occurred within the last 10 years due to renewed interest in the field of energy consumption. There are two areas of focus where researchers improved the NILM design. Firstly, the features used for identification and the detection methods used were improved upon. Mulinari *et al* [11] showed that by implementing new identification features into existing detection methods, the accuracy of these methods can be significantly improved. Furthermore, it was discovered that combining new features with new or improved detection methods also resulted in improved accuracy [1, 5, 12, 13]. Most of the newly developed algorithms' results were obtained via simulations and not through practical implementation, as previously mentioned. The exception to this was the work of Mubarok *et al*, [13], who successfully implemented a NILM on an embedded platform. However, it was discussed that the method used was limited as the complexity and time needed to perform an identification increases exponentially with the number of loads. On the other hand, Guohua *et al*, [1] demonstrated that a NILM using a random forest algorithm can produce similar levels of accuracy as that of the method used by Mubarok *et al* [13]. The advantage of the random forest algorithm was that it needed significantly less time to compute, the time taken scaling with  $O(n \log n)$  as opposed to  $O(n^2)$ .

Upon reviewing the relevant literature, it can be concluded that the work of Mubarok *et al* [13] (where the method's functionality was only demonstrated through simulations) may be improved upon by using the methods of Guohua *et al* [1] for practical implementation purposes. This can be done by implementing a decision tree on an embedded platform. It is proposed that these improvements will result in the accuracy of the prediction method increasing to 95 %.

## 2. Approach

---

The goal was to develop a smart energy meter to identify the real-time state of the connected loads. The energy meter was produced to meet a number of requirements - systems and subsystems aimed at fulfilling each requirement respectively were combined to create a fully functioning end product.

The point of departure for the design process was to split the energy meter into two distinct subsystems. The first was the analogue front end which facilitated a safe tie-in with the live 230V service. The energy meter and user was galvanically isolated from mains electricity supply. The second subsystem processed the sampled data, and created the interface through which the user can view and interact with this processed data.

The first part of the design (discussed in more detail in section 3, below) focused on the ADCs design requirements. An ADC and embedded platform were selected once the ENOB and sample rate were determined. The preferred outcome was to obtain an embedded system with an onboard ADC. However, such a platform which meets the design requirements was found to be unjustifiably expensive given its intended application. An alternative solution was thus to select the embedded platform and external ADC which still meets the design requirements but which would be more cost-effective.

Once the ADC was selected, a circuit was built to galvanically isolate the device from mains electricity, to convert voltage, and to current signals to a range the ADC can measure. Transformers (safety and current clamp), optocouplers and sense resistors were considered as potential isolation methods. The safety transformer was determined to be the most beneficial method with the least negligible drawbacks for voltage measurements. Meanwhile (as will be discussed in more detail later), the current clamp transformer was found as the best solution for the current measurement. Thereafter the complementary circuitry was designed to interface the transformer and current clamp to the respective ADCs.

The completed analogue circuitry was successfully simulated, after which the circuit schematics were transformed into hardware schematics. The hardware schematics were then used to create the physical circuitry. Here a modular design was implemented to allow for individual sections of the circuit to be modified or changed if necessary as opposed to the entire circuit having to be re-designed or -constructed following possible change.

The embedded platform and ADCs were interfaced with a serial communication protocol which allowed the two ADCs to sample the voltage and current simultaneously, reducing the need for phase compensation when calculating the power. The raw binary data that the ADC produced was converted to a floating point format, and calibrated to reflect the actual values of the voltage and current.<sup>1</sup>

Two digital FIR filters - an Anti-aliasing and a 50 Hz low-pass filter - were designed from first principles and then applied to the raw signal sampled by the ADCs. These filters used different window functions to fulfil different functions. The Anti-aliasing filter was used to prevent the occurring of unwanted spectral components in the frequency system of interest as a result of the sampling process. The 50 Hz low-pass filter was designed to isolate the 50 Hz

---

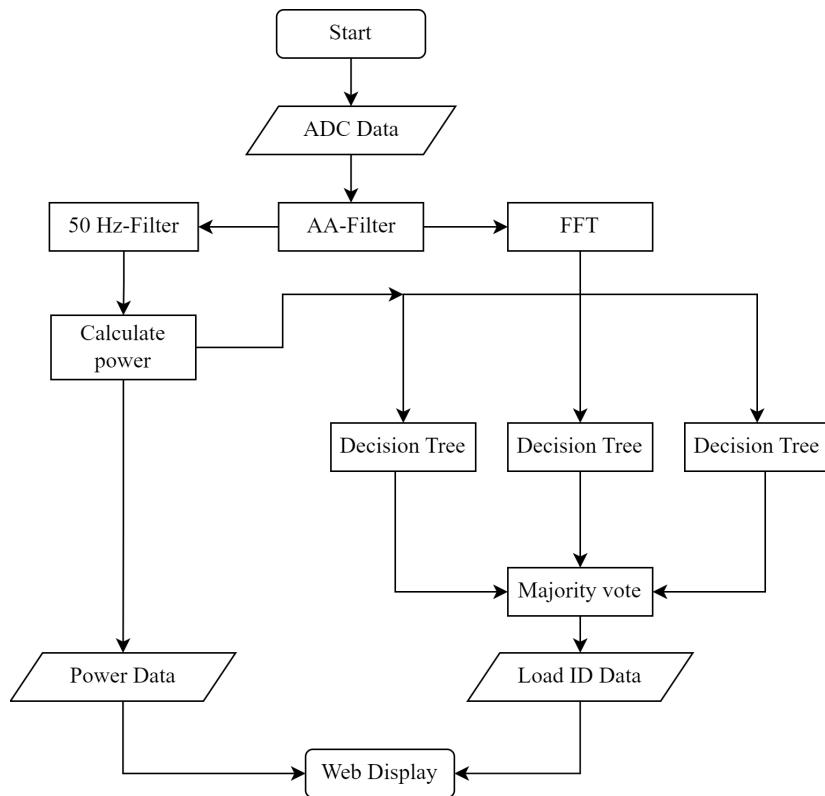
<sup>1</sup> Implemented on embedded platform

power component from the harmonics to make the power measurements more accurate.<sup>1</sup>

Thereafter, a data collection program was developed to collect the necessary features for the identification of loads. These features included the real and reactive power components, as well as the harmonics of the current load which were produced by performing the FFT transform on the current signal.

A detection algorithm was then developed, the main goal being to obtain an algorithm which offered sufficient accuracy with the lowest possible performance overhead. A number of method algorithm options were considered, namely classification decision trees, neural networks and K-Nearest Neighbors. The investigation found that multiple decision tree classifiers for the creation of a random forest was the method which best fulfilled the necessary requirements. As such, the collected data was used to train the decision trees and create the forest.<sup>2</sup>

The hardware and software were then combined. The figure 3 depicts the final interfacing of the main program, which called all the above software, resulting in a completed system that performed the energy calculation and load detection. The main program then presented the data on a web page for the user to view in accordance with the flowchart depicted in figure 3



**Figure 3.**  
**Flow chart of software functions integrated into a complete detection system.**

---

<sup>2</sup>Implemented on personal computer

### 3. Design and implementation

---

#### 3.1 Design summary

Deliverable or task	Implementation	Completion of deliverable or task, and section in the report
Design of mains voltage interface and attenuation circuit	The student completed the design from first principles	Complete Section 3.2.1
Design of mains current interface and attenuation circuit	The student completed the design from first principles	Complete Section 3.2.3
Design of differential ADC input driver	The student completed the design from first principles	Complete Section 3.2.4
Design of ±2.5 V reference	The student completed the design from first principles	Complete Section 3.2.5
Hardware design and implementation voltage interface	The student implemented the hardware design on veroboard from first principles	Complete Section 3.4.1
Hardware design and implementation current interface	The student implemented the hardware design on veroboard from first principles	Complete Section 3.4.2
Hardware design and implementation ADC input driver	The student implemented the hardware design on veroboard from first principles	Complete Section 3.4.3
Hardware design and implementation ±2.5 V reference	The student implemented the hardware design on veroboard from first principles	Complete Section 3.4.4
PRU ADC Interface Software	The student implemented a custom version of the SPI protocol from first principles to interface the ADC and PRU	Complete Section 3.5.1 and 3.5.2
Digital FIR Filter Design	The student designed the filter from first principles The implementation was done using a library	Complete Section 3.5.3
Development of the data collection system	The student designed and implemented a custom automated data collection system from first principles. The GUI used to control the system was implemented using a library	Complete Section 3.5.5
Development of the decision tree	The student implemented a custom version of the SPI protocol from first principles to interface the ADC and PRU	Complete Section 3.5.6
Training of the decision tree and creation of the forest	The student implemented a decision tree algorithm from first principles with the particular implementation of the decision tree forest being completely original	Complete Section 3.6.1
Full system integration GUI development	The system was fully integrated as a Dash Application. The GUI components provided by Dash are taken off the self. All the functions that generate the information that GUI displays are designed and integrated using first principles	Complete Section 3.7

**Table 1.**  
**Design summary.**

## 3.2 Theoretical analysis

### 3.2.1 ADC Sample rate and resolution requirements

All ADCs have three main specifications: the voltage range they can measure, the number of bits they can use (resolution) to represent that voltage, and the speed (sample rate) at which the conversion takes place. The accuracy of the measurement is affected by all three of these specifications. The effective number of bits (ENOB) is the measure of how much resolution an ADC will have in a particular system, this being dependant on all three of the above specifications. Increasing the ENOB measure also increases accuracy and allows smaller signals to be measured. However, this increase comes at a trade off for every bit increase to ENOB. The design complexity increases exponentially.

The following calculations were used to determine the minimum ENOB required for the voltage signal. The specification for household voltage supplied in South Africa is  $\pm 230 \text{ V} \pm 10\%$ , meaning that the range over which the samples needed to be taken can be calculated by equations 1 and 2

$$\begin{aligned} V_{min} &= 230 \text{ V} \times (100 - 10\%) \\ &= 230 \text{ V} \times (90\%) \\ &= 207 \text{ V}, \end{aligned} \tag{1}$$

$$\begin{aligned} V_{max} &= 230 \text{ V} \times (100 + 10\%) \\ &= 230 \text{ V} \times (110\%) \\ &= 253 \text{ V}. \end{aligned} \tag{2}$$

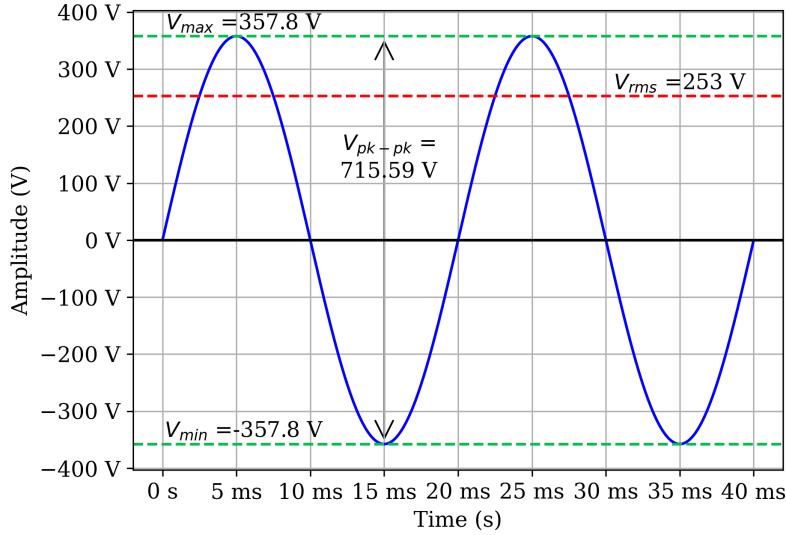
The root-mean-square (RMS) Voltage was used to calculate the values above. The definition of effective or RMS value that was provided in [14] is that the effective value of a periodic current is the DC current that delivers the same average power to a resistor as the periodic current. Using the RMS value would result in a design that cannot measure the AC voltage waveform over the desired range as this is not the peak-to-peak voltage of the signal. The actual peak-to-peak values of the voltage signal were derived using the equations presented below:

$$\begin{aligned} V(max)_{pk-pk} &= [V_{max} - (V_{min})] \times \sqrt{2} \\ &= [253 \text{ V} - (-253 \text{ V})] \times \sqrt{2} \\ &= 715,59 \text{ V}, \end{aligned} \tag{3}$$

$$\begin{aligned} V(min)_{pk-pk} &= [V_{max} - (V_{min})] \times \sqrt{2} \\ &= [207 \text{ V} - (-207 \text{ V})] \times \sqrt{2} \\ &= 585.48 \text{ V}. \end{aligned} \tag{4}$$

Figure 4 is a mathematical simulation of a 50 Hz sine wave with the largest possible amplitude the system was expected to measure. Indicated in the figure are the  $V_{rms}$  and  $V_{max}$  values - the

$V_{max}$  value is noticeably larger than  $V_{rms}$  value.



**Figure 4.**  
**Mathematical simulation of expected voltage signal.**

A design error of 1.0 % was chosen for the voltage measurement in order to prevent the desired error of the power measurement from exceeding  $\pm 2.5 \%$ . This allows for a  $\pm 1.5 \%$  error in the rest of the system. Equation 5 below was used to convert the 1.0 % error to a voltage value in relation to the smallest expected peak-to-peak voltage signal

$$\begin{aligned}\Delta V &= 1\% \times V(\min)_{pk-pk} V \\ &= 1\% \times 585.48 V \\ &= 5.85 V.\end{aligned}\tag{5}$$

Using the result from equation 5, the number of quantisation levels required to express the maximum peak-to-peak voltage while maintaining the design error was calculated using equation 6,

$$\begin{aligned}Levels &= \frac{V(\max)_{pk-pk}}{\Delta V} \\ &= \frac{715.59}{5.85} \\ &= 122.22 \\ &\approx 123.\end{aligned}\tag{6}$$

Using the number of quantisation levels from equation 6 , the minimum ENOB were obtained

using equation 7,

$$\begin{aligned}
 ENOB &= \log_2(\text{Levels}) \\
 &= \log_2(123) \\
 &= 6.93 \\
 &\approx 7 \text{ bits}.
 \end{aligned} \tag{7}$$

Therefore, only a 7-bit ADC is required to sample the voltage signal while maintaining an error of less than  $\pm 1.0\%$ .

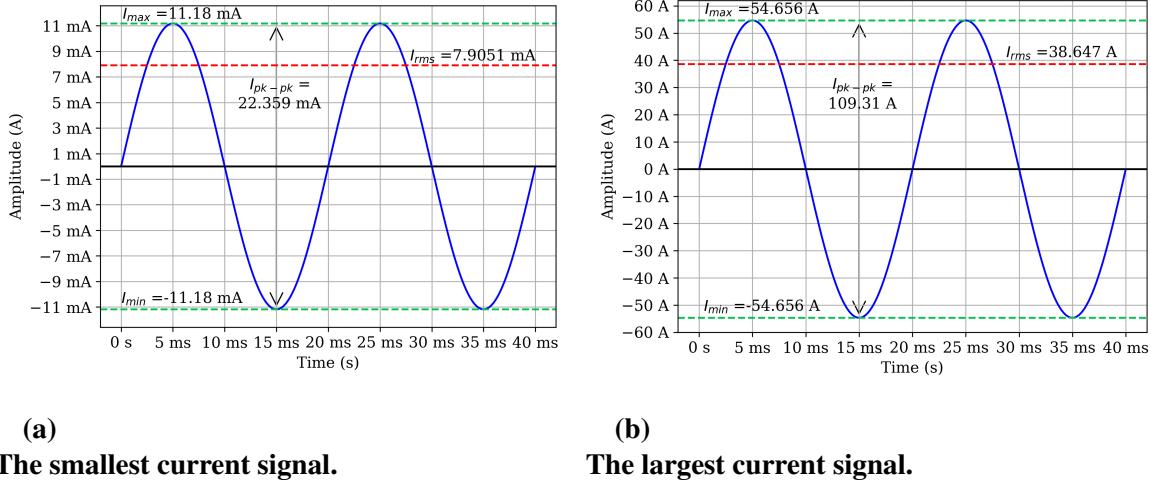
A similar process was followed to calculate the ENOB required to represent the current signal as the process used for sampling the voltage signal. The load's current draw is dependant on its voltage and power rating. This means that both the voltage and power draw values must be accounted for when calculating the range of current values needing to be measured.

Equation 8 and equation 9 are used to calculate the minimum and maximum current values that the system is expected to be able to measure,

$$\begin{aligned}
 P_{min_{rms}} &= V_{max_{rms}} \cdot I_{min_{rms}} \\
 2 \text{ W} &= 253 \text{ V} \cdot I_{min_{rms}} \\
 \therefore I_{min_{rms}} &= \frac{2 \text{ W}}{253 \text{ V}} \\
 &= 7.9051 \text{ mA},
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 P_{max_{rms}} &= V_{min_{rms}} \cdot I_{max_{rms}} \\
 8000 \text{ W} &= 207 \text{ V} \cdot I_{max_{rms}} \\
 \therefore I_{max_{rms}} &= \frac{8000 \text{ W}}{207 \text{ V}} \\
 &= 38.647 \text{ A}.
 \end{aligned} \tag{9}$$

The Figure 5 depicts two current signals. The left Figure 5a is the smallest possible signal that the system is expected to measure. The right Figure 5b is the largest possible signal that the system is expected to measure.

**Figure 5.**

**The smallest and largest current signals that are expected to be measured.**

The data in Figure 5a shows that the smallest possible current value is 11.18 mA. The largest possible value is 54.656 A. However, unlike the voltage, this accuracy value only applies to loads that draw over 1.2 A. Thus, in order to maintain a measurement error of less than  $\pm 2.5\%$ , a design error of  $\pm 1.0\%$  is expected - the value of which is calculated by equation 10,

$$\begin{aligned}\Delta I &= 1\% \times 1.2 \times \sqrt{2} A \\ &= 1\% \times 3.39 A \\ &= 33.94 mA.\end{aligned}\tag{10}$$

Since the above calculated value of 33.94 mA is larger than 11.18 mA calculated in equation 11 above, the ENOB can be calculated, as in

$$\begin{aligned}Levels &= \frac{I(max)_{pk-pk}}{I(min)} \\ &= \frac{109.31}{11.18 mA} \\ &= 9777.28 \\ &\approx 9778.\end{aligned}\tag{11}$$

Using the number of levels calculated by equation 11 above, this can then be expressed as

ENOB using equation 12,

$$\begin{aligned}
 ENOB &= \log_2(Levels) \\
 &= \log_2(9778) \\
 &= 13.26 \\
 &\approx 14 \text{ bits.}
 \end{aligned} \tag{12}$$

Comparing the results from equations 7 and 12, it is clear that an ADC which matches or exceeds 14 bits is required. As most ADCs do not come with a resolution of 14 bits, a 16-bit ADC is selected.

The sample rate of the ADC was selected to exceed that of 49 kHz. This is because the main features required for detecting loads are the harmonics a device will create while in operation. The harmonics will be calculated using an FFT transform on the current signal. Only the odd harmonics are used. With a sample rate of 49 kHz, this implies that the FFT transform will have a frequency range of 24.5 kHz. Since only the odd harmonics are used for load identification, a total of 245 harmonics can be used to fingerprint loads. Following the above criteria, the MCP33131D-10 from Microchip was selected based on the specifications summarised in Table 2 below:

MCP33131D-10		
Parameter	Minimum requirement	Actual
Sample Rate	$\pm 49 \text{ kHz}$	$\pm 1 \text{ MHz}$
Resolution	14-bits	16-bits
Input Range	$\pm 5.0 \text{ V}$	$\pm 5.1 \text{ V}$

**Table 2.**  
**Summary of the ADC specifications that are required to meet the needs of the application [15].**

Since the ADC is a precision 16-bit ADC, it requires an external voltage reference that maintains the same level of accuracy. Therefore, the MAX6250 (a precision 5.0 V) was selected as the external voltage reference. This is because the voltage reference has a very low noise level of  $\pm 1.5 \mu\text{V}$  peak-to-peak - this being far below the ADC's quantisation error.

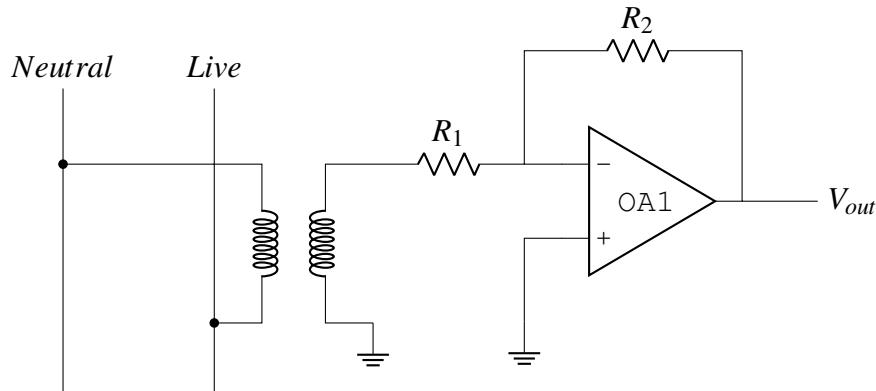
### 3.2.2 Mains Voltage Signal Isolation and Interface

There are many known methods for measuring a live voltage signal using a microcontroller. These include using a shunt resistor in series to create a voltage drop that is low enough for the ADC to measure. However, it offers little to no protection for the device and the user. One way of mitigating this is to create a voltage drop using multiple resistors in series. Implementing this design can isolate the device and user in the event that a resistor fails. However, this is still not a perfect solution as it results in the addition of multiple points of failure.

There are two methods which may be used to mitigate this potential danger by completely isolating the system from the potentially dangerous live voltage supply. The first method uses

an optocoupler to completely isolate the live supply from the device performing the voltage measurement. The device makes use of a LED with a brightness output which is proportional to the supplied voltage. A photodiode is then used to convert this light to a voltage signal, the signal being dependent on the brightness of the LED. One advantage of this method is that there is no interference with the actual voltage supply. However, the disadvantage is that there is no linear output proportional to the voltage input, there thus needing to be a calibration step in software or hardware to produce this linear output.

The Figure 6 below shows a second alternative method via which to isolate and step down the dangerous voltage level using a safety transformer.



**Figure 6.**  
**Voltage interface circuit.**

The safety transformer depicted in the circuit above is rated to transform a 230 V down to 12 V. As such, the max peak-to-peak voltage is 37.34 V. Since the voltage is still too high for the specified ADC(MCP33131D-10) even after implementing the safety transformer, the voltage must be transformed to a range of  $\pm 2.5$  V. This transformation is affected using a voltage amplifier comprised of an operational amplifier and two resistors, the amplification factor being governed by equation 13

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1}, \quad (13)$$

respectfully [16].

Substituting  $V_{in} = 37.34$  V,  $V_{out} = 2.5$  V and selecting a value of  $10\text{ k}\Omega$  for  $R_1$ , results in a value of  $1250\ \Omega$  for  $R_2$ .

### 3.2.3 Mains Current Signal Isolation and Interface

As mentioned in the section above, there are many ways in which to sample a live 230 V current signal. One method is to insert a shunt resistor between the current supply and the load on the live voltage wire. By measuring the voltage over the shunt resistor (which has a precise resistance value that is known), the current flowing to the load can be calculated using

ohms law. However, this method has the potential to place the device and its user in danger should the resistor fail.

A safer solution is to use a current clamp, a transformer which typically has a split core and can wrap around the live wire. A current is produced through the wire by placing the core around it. This current can then be converted into a voltage using a resistor. However, not all current clamps are necessarily equally suited to perform this function. Current clamps can vary in three distinct ways:

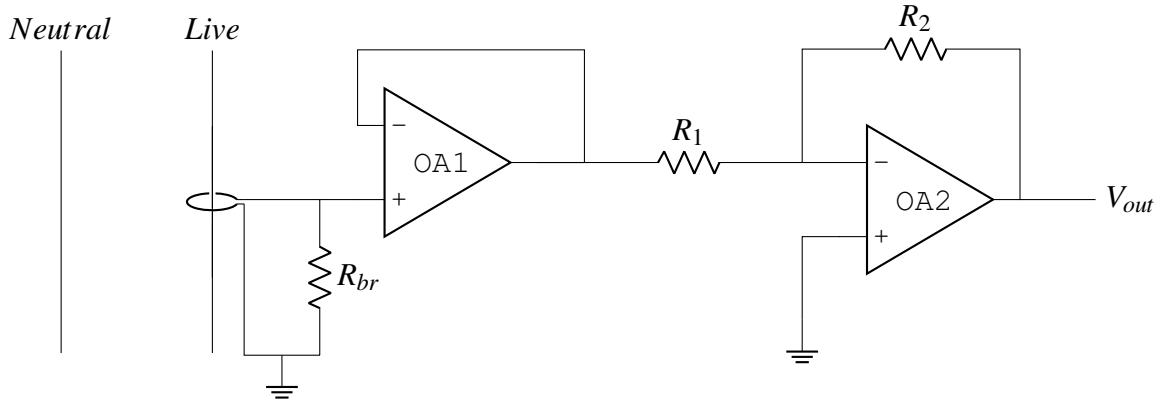
- First, the number of windings between the primary coil and the secondary coil may vary. This number is directly responsible for the range of currents which the transformer is rated to measure at its specified accuracy level.
- Secondly, the magnitude of the burden resistor may differ, this having a direct impact on the linear region and accuracy of the current signal produced by the transformer.
- Finally, the accuracy class of the current clamps may vary - this accuracy being determined by the maximum rated error that occurs in the linear measurement region of the current clamp.

Having taken all of the above factors as well as the desired accuracy specified in the design requirements into account, the ATO-75-B1I-D10 current clamp from LEM was selected as most suitable for the design. The Table 3 below is a summary of requirements that the ATO-75-B1I-D10 met or exceeded resulting in it being selected as most suitable for this particular application.

ATO-75-B1I-D10 with $R_{br} = 4 \Omega$		
Parameter	Minimum requirement	Actual
Primary current ( $I_p$ )	$\pm 75 \text{ A}$	$\pm 125 \text{ A}$
Rated primary current ( $I_{pr}$ )	$\pm 75 \text{ A}$	$\pm 75 \text{ A}$
Frequency bandwidth (-3 dB)	49 kHz	500 kHz
Ratio error ( $\epsilon$ )	$\pm 2 \%$	$\pm 1.0 \%$
Linearity error ( $\epsilon_L$ )	$\pm 0.5 \%$	$\pm 0.1 \%$
Rated secondary Current ( $I_{sr}$ )	NA	$\pm 75 \text{ mA}$
Rated transformation ratio ( $K_{ra}$ )	NA	1000 A/A

**Table 3.**  
**Summary of the specifications that are required to meet the needs of the application [17].**

The circuit diagram in Figure 7 below depicts the current clamp and the burden resistor which create a voltage signal proportional to the current flowing through the live supply wire.



**Figure 7.**  
**Current interface circuit.**

With the burden resistor  $R_{br} = 4 \Omega$ , the ATO-75-B1I-D10 current clamp indicated in the circuit schematic presented in Figure 7 is rated to achieve the error values stated in Table 3 above. This resistor is nominal compared to the others used in the circuit. As such, the voltage signal produced over  $R_{br}$  is susceptible to loading, which could make the amplification of the signal all but impossible. A buffer amplifier was placed between the burden resistor and the amplification circuit in order to mitigate the loading of the signal. The signal is then amplified using a voltage amplifier comprised of an operational amplifier and two resistors, with the amplification factor being governed by the equation 14

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1}, \quad (14)$$

respectfully [16]. Where  $V_{in}$  is defined by equation 15

$$V_{in_{max}} = I_{sr} \cdot 4\Omega. \quad (15)$$

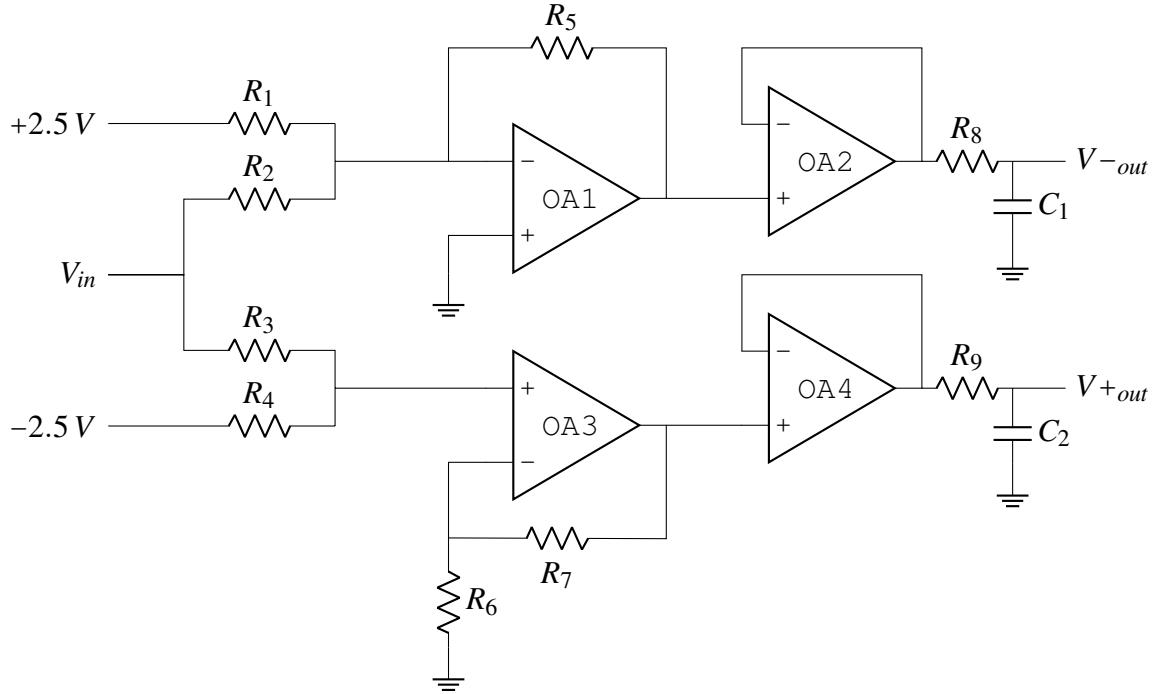
Substituting  $I_{sr} = 106.07 \text{ mA}$  into equation 15 above yields,  $V_{in_{max}} = 424.26 \text{ mV}$ . With  $V_{out_{max}} = \frac{V_{ref}}{2} = 2.5 \text{ V}$ . Selecting a value of  $32 \Omega$  for  $R_1$ , results in a value of  $826 \Omega$  for  $R_2$ .

### 3.2.4 Differential ADC input driver

The differential ADC works by sampling the difference between two signals. Thus, the original signal needs to be converted to a differential signal in order to make use of the full 16-bit range of the MCP33131D-10 ADC. It is imperative to note that, while the data sheet of the MCP33131D-10 states that it can measure  $\pm V_{ref}$ , this specification is somewhat ambiguous and potentially misleading. The specification actually refers to the difference between the "positive" and "negative" inputs to the ADC - not to the inputs themselves. The ADC is also only designed to have a positive voltage applied to both inputs. Thus, the original signal needs to be modified to use the ADC's entire range.

The before-mentioned modification was done by designing a differential ADC driver (depicted in Figure 8) to modify the signal. The purpose of the differential ADC driver is to take a signal and apply  $-\frac{V_{ref}}{2}$  offset - after which the modified signal is then inverted using an inverting

unity gain amplifier. Offsets are applied to ensure that the signal is positive and in the expected range for the ADC. An offset of  $\frac{V_{ref}}{2}$  is thus also applied to the original signal using a normal unity gain amplifier.



**Figure 8.**  
**Differential ADC input driver.**

The Figure 8 above depicts a circuit schematic that was designed. The non-inverting unity gain amplifier is comprised of an operational amplifier (OA1) and resistors  $R_1, R_2$  and  $R_3$ . The inverting unity gain amplifier is comprised of an operational amplifier (OA2) and resistors  $R_4, R_5$  and  $R_6$ . Since this ADC driver is comprised of unity gain amplifiers thus the resistors  $R_1$  through  $R_6$  are all ideally the same value of  $10\text{ k}\Omega$ . The differential signals are then buffered by the operational amplifiers (OA3 and OA4) to prevent loading. This signal is then run though a passive first order anti aliasing RC filter who's cut off frequency is defined by the following equation 16

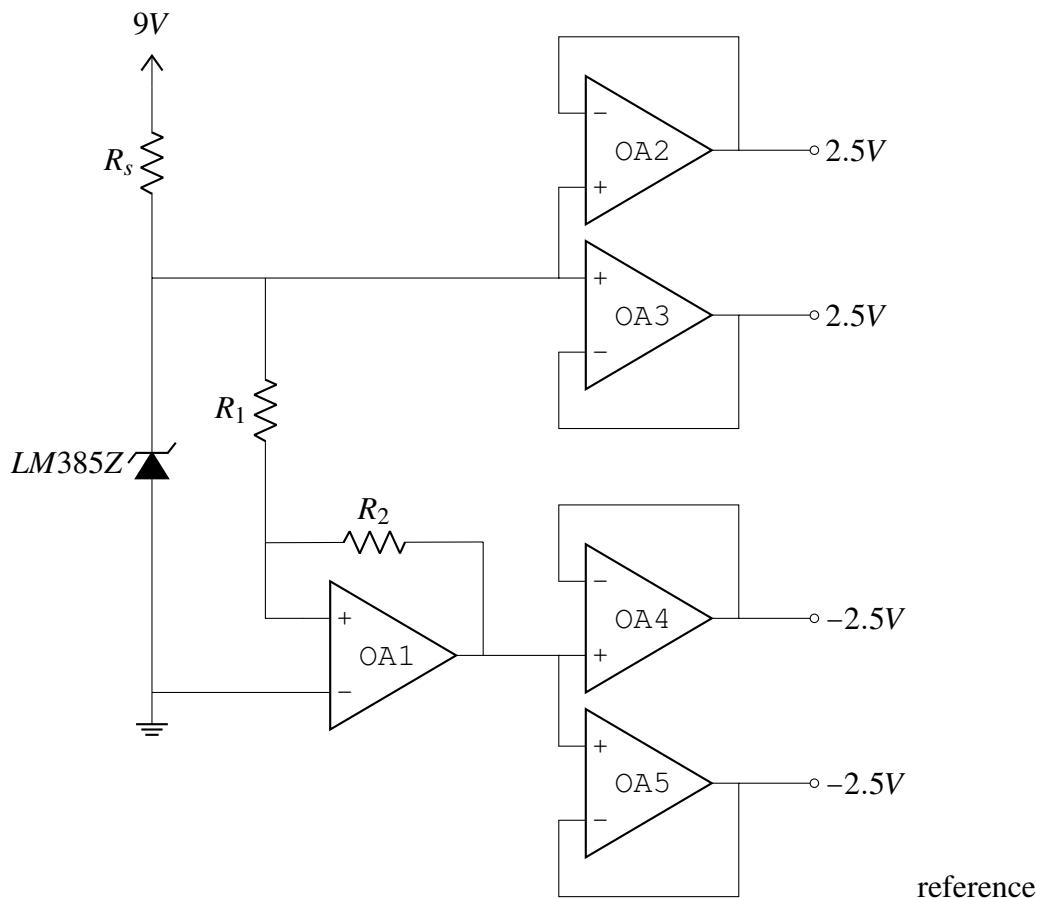
$$f_c = \frac{1}{2\pi RC}, \quad (16)$$

respectfully [16]. With a desired cutoff frequency of  $20\text{ kHz}$  the  $2.2\text{ nF}$  capacitors were specified  $C_1$  and  $C_2$ . The resistors  $R_8$  and  $R_9$  were then calculated to have a value of  $3617\text{ }\Omega$  using equation 16. Since  $3617\text{ }\Omega$  is not a standard resistor value a resistor value of  $3300\text{ }\Omega$  was selected. Equation 16 was then used again to calculate the as design new cut off frequency of  $21.9\text{ kHz}$ .

### 3.2.5 Voltage Reference

The Figure 8 depicted above is the differential ADC driver. On the left, the design requires both a positive and negative  $\pm 2.5$  V references to offset the signals. Since this design is used for both voltage and current signals, two of each reference are required.

In order to create these four references, buffers are used instead of creating a duplicate of the left side of the circuit depicted below in Figure 9.



**Figure 9.**  
 $\pm 2.5$  V reference used as the offset for the circuit in Figure 8.

As one can see from the circuit depicted above makes use of the LM385Z 2.5 V precision voltage reference from Texas Instruments. This particular voltage reference creates a controlled voltage drop over the diode that is controlled by the current that follows through the

resistor  $R_s$ . The value of  $R_s$  is defined by the equation 17

$$\begin{aligned} R_s &= \frac{V_s - V_{ref}}{I_L + I_{ref}}, \\ &= \frac{9\text{ V} - 2.5\text{ V}}{2.0\text{ mA}}, \\ &= 3300\text{ }\Omega, \end{aligned} \tag{17}$$

respectfully [18].

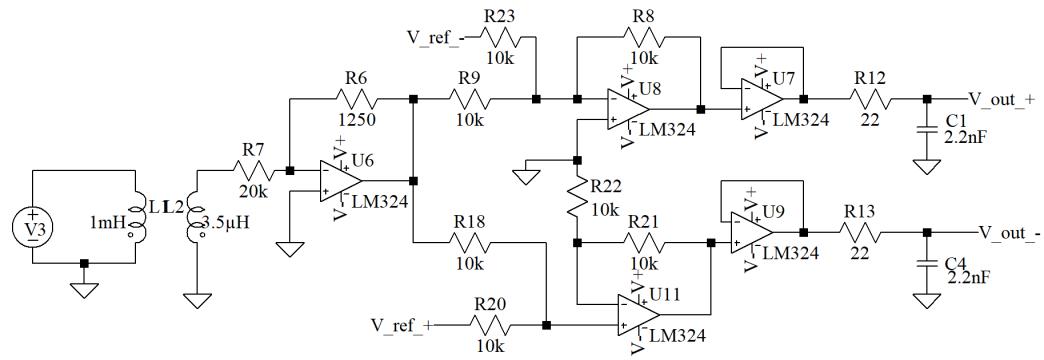
In order to prevent the large current draw and thus affect the stable 2.5 V at the base of  $R_s$ , significant resistance is required for  $R_1$ . The voltage produced at the base of  $R_s$  is then inverted using an inverting amplifier comprised of the operational amplifier (OA1),  $R_1$  and  $R_2$ . The purpose of the inverting amplifier is not to alter the magnitude of the voltage but rather create a negative voltage of the same magnitude  $R_1 = R_2$ . Since this is an amplifier design the general rule of keeping the resistors below 100 k $\Omega$  was followed and a resistance value of 47 k $\Omega$  is selected for resistors  $R_1$  and  $R_2$ .

### 3.3 Simulation

The simulations depicted in this section of the report were all performed on the SPICE simulation software known as LTspice. The raw file outputs were then imported into python to produce the figures you see in section below. Since the intended application of this circuit is power measurement and feature extraction the choice of operational amplifier choice has little impact on the performance of operation. What was considered was the packaging of the operational amplifier. The differential ADC diver makes use of four operational amplifiers. Using the LM324 from Texas Instruments which comes as a quad packaged operational amplifier integrated circuit (IC) greatly simplifies the hardware design and implementation described in section 3.4.

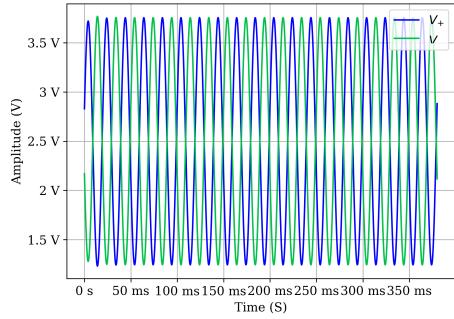
#### 3.3.1 Mains Voltage Signal Sampling

The circuit diagram in Figure 10 is the complete circuit diagram used to simulate the voltage sampling circuit. The circuit is the combination of circuits depicted in Figures 6 and 8. A mutual inductor is used to simulate the safety transformer. The simulation is performed twice with the input voltage varied to simulate the maximum and minimum voltages that can be expected when implemented in a practical application.



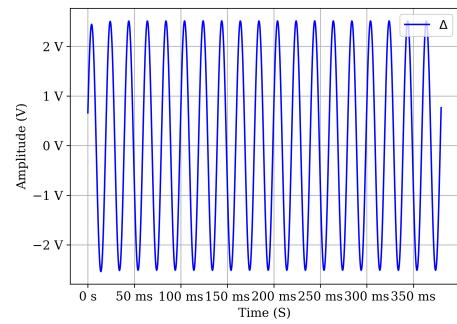
**Figure 10.**  
**Complete voltage interface circuit with differential ADC driver.**

The sub figures depicted in Figure 11 are used to confirm that the design voltage interface circuit is able to convert the large voltage signal from the mains supply to voltage range that is able to leverage the full 16-bit resolution of the ADC.



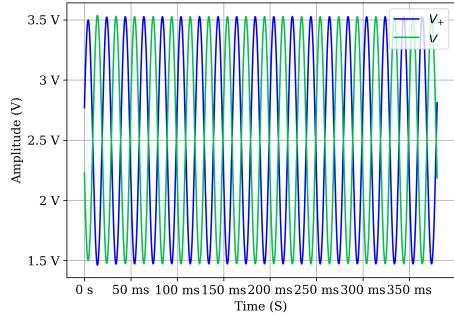
(a)

**Positive and negative outputs of the voltage interface circuit with the maximum expected input.**



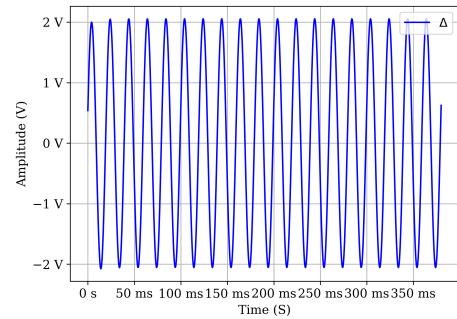
(b)

**Difference of outputs seen in Figure 11a.**



(c)

**Positive and negative outputs of the voltage interface circuit with the minimum expected input.**



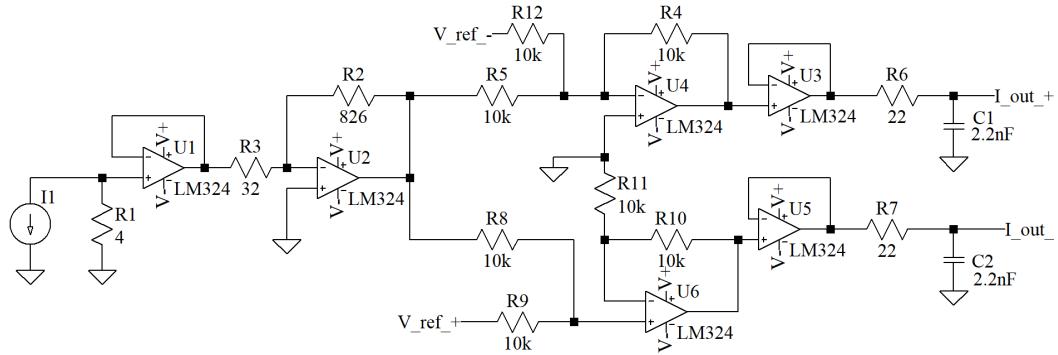
(d)

**Difference of outputs seen in Figure 11c.**

**Figure 11.**  
**Simulation of the voltage response of the circuits in Figure 10**

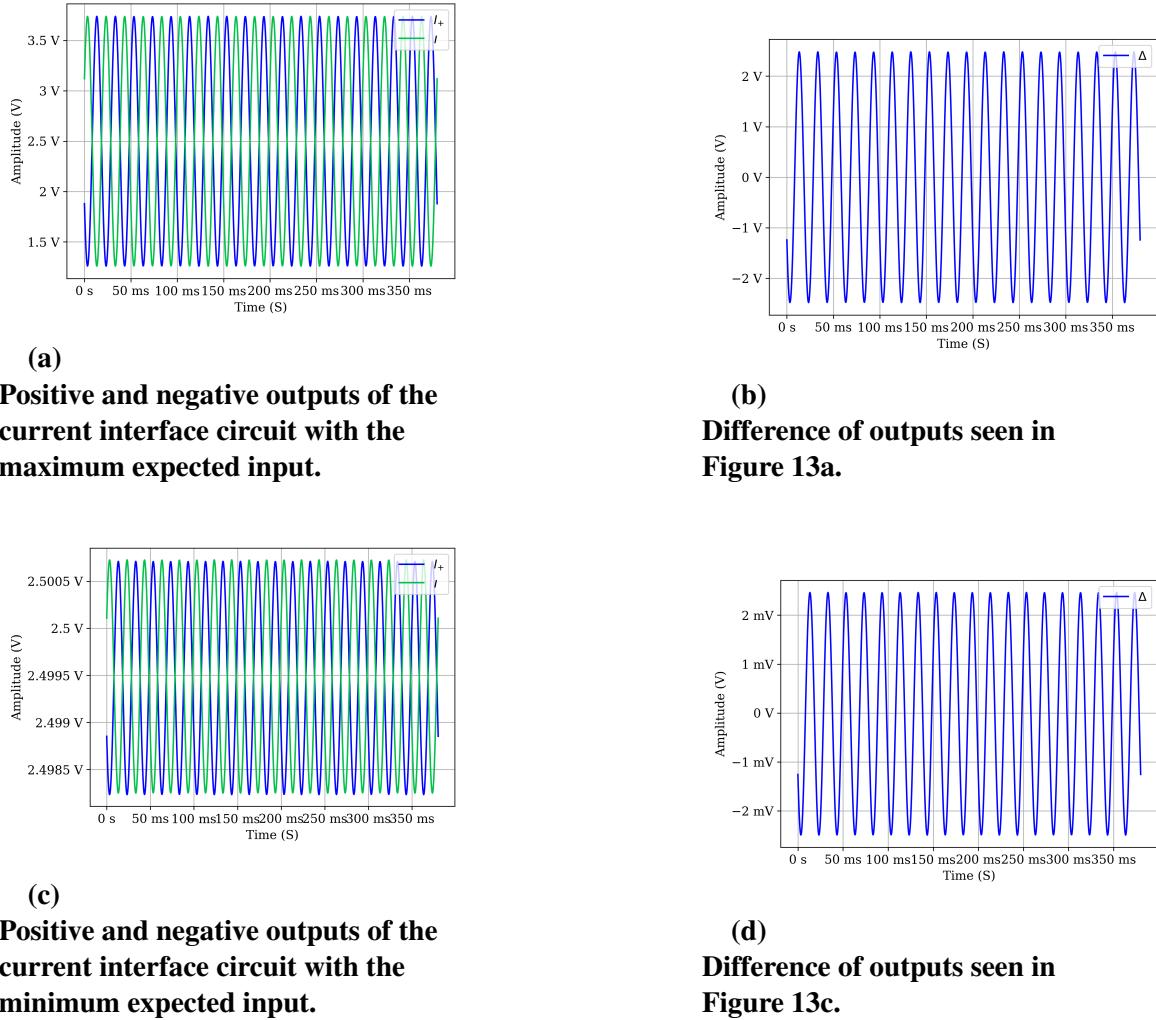
### 3.3.2 Mains Current Signal Sampling

The circuit diagram in Figure 12 is the complete circuit diagram used to simulate the current sampling circuit. The circuit is the combination of circuits depicted in Figures 7 and 8. A current source is used to simulate the current created by the ATO-75-B1I-D10 current clamp. The simulation is performed twice with the input current varied to simulate the maximum and minimum currents that can be expected when implemented in a practical application.



**Figure 12.**  
**Complete current interface circuit with differential ADC driver**

The sub figures depicted in Figure 13 are used to confirm that the design current interface circuit is able to leverage the full 16-bit resolution of the ADC over the full range of expected current draws from devices in a practical application.

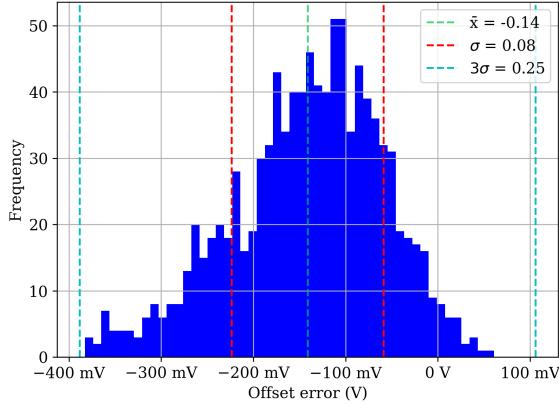


**Figure 13.**  
**Simulation of the voltage response of the circuit in Figure 7**

The simulations of both the complete voltage and current circuits were done with ideal resistors. While this is easily corrected by making a use of a potentiometer for the amplification stages of both circuits in Figures 6 and 7 respectively. This method is no simple to implement for circuit depicted in Figure 8.

In order to mitigate the effects of the non-ideal components in the ADC driver circuit the results Monte Carlo simulation Figure 14 that swept the resistors values with their specified ranges. As can be seen with all the resistors having a specified tolerance of 1 % the mean offset error between the positive and negative values is 140 mV. While this does not seem like a large value converting it to the number of ADC steps (1835) that will be lost as result it has a meaningful impact on the range of values.

### 3.3.3 Differential ADC input driver



**Figure 14.**  
**Histogram depicting the expected offset error**

## 3.4 Hardware design and implementation

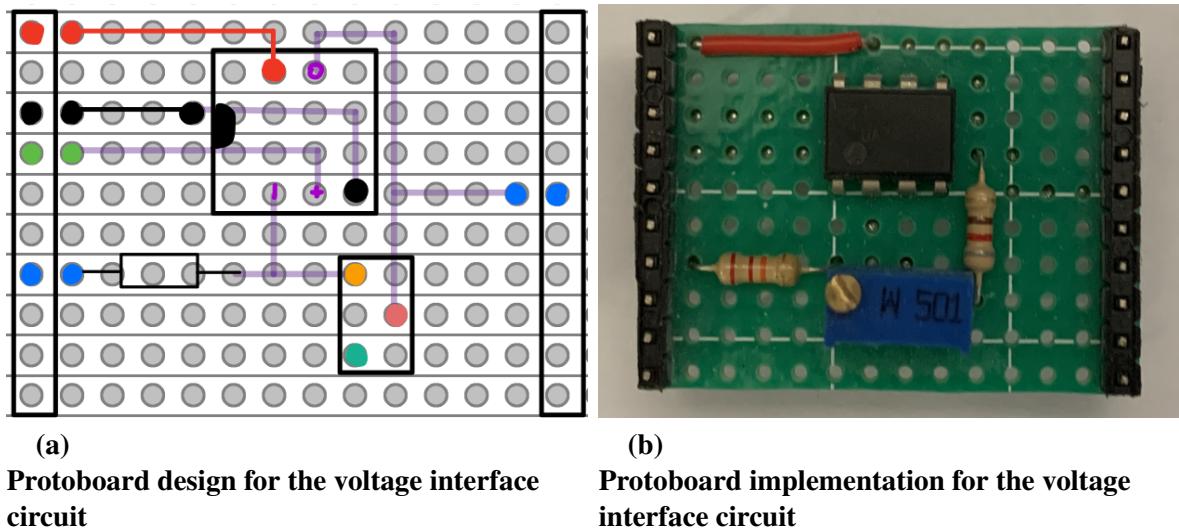
As the device is in the prototyping phase a modular design principle is followed. The advantage of using the modular design is that it subdivides the system into smaller subsystems. These smaller subsystems are then transformed into modules. The modules are independently created, modified and replaced as needed.

The general method to transform a circuit schematic to a physical hardware design is described in the following steps:

- Step 1: The component symbols used to represent individual components in the circuit schematic must be converted to their equivalent physical package dimensions and pin-outs.
- Step 2: The packages must then be arranged and the routing for the connections planned on a blank schematic of the prototyping board.
- Step 3: Make the necessary modifications (if any) due to any physical layout compatibility or routing issues and update the schematic from step 2.
- Step 4: Place the physical components on the board and check for any compatibility or routing issues.
- Step 5: Solder components in place.
- Step 6: Test the circuit to ensure that all connections are correct and that the circuit performs according to the simulation.

### 3.4.1 Mains Voltage Signal System

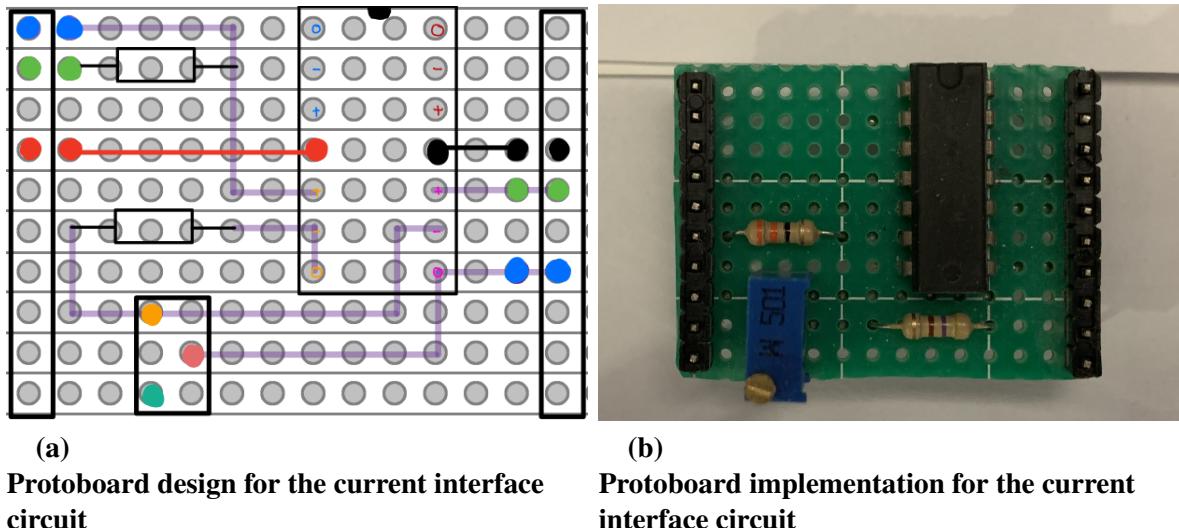
Following the general steps 1 through 3 described above the Figure 15b is created from the circuit schematic in Figure 6. Since the circuit is comprised of non-ideal components, a potentiometer is used to replace the resistor  $R_1$  in Figure 6. Making this modification allows for a calibration that compensates for the tolerances found in non-ideal components.



**Figure 15.**  
**Hardware design and implementation of the voltage interface module**

### 3.4.2 Mains Current Signal System

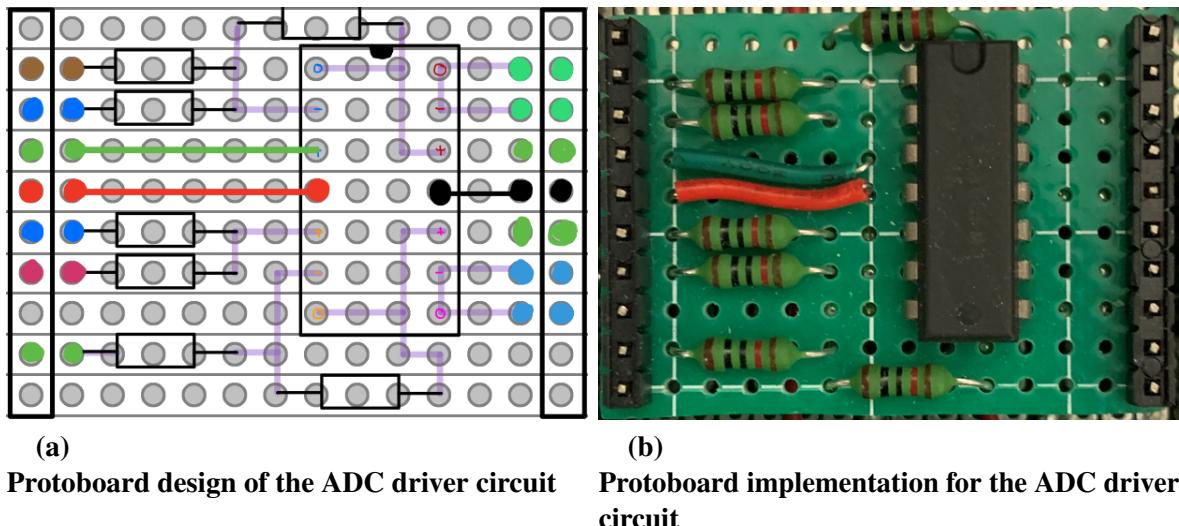
Following the general steps 1 through 3 described above, the Figure 16b is created from the circuit schematic in Figure 7. As with the voltage interface circuit, the current interface circuit is comprised of non-ideal components. Using the same design choice to replace resistor  $R_1$  in Figure 7 with a potentiometer allows for the calibration to compensate for the non-ideal components.



**Figure 16.**  
Hardware design and implementation of the current interface module

### 3.4.3 Differential ADC input driver

Following the general steps 1 through 3 described above the Figure 17b is created from the circuit schematic in Figure 8.



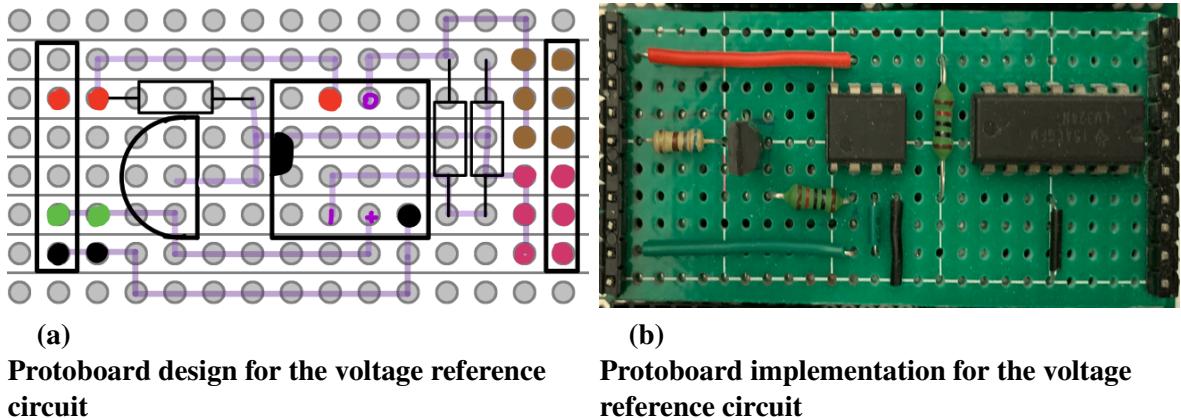
**Figure 17.**  
Hardware design and implementation of the ADC driver module

In the previous two cases variations in component values were compensated for by using a potentiometer - however, this is an impractical method for the circuit depicted in figure 17. The

resistors in the circuit have been specified as having a tolerance of no greater than  $\pm 1$  percent. This variance in the resistance still results in a large variance in the offset error between the two signals, as seen in Figure 14. In order to mitigate this error a "real world" adaptation of the Monte Carlo simulation used to produce Figure 14. 100  $10\text{k}\Omega \pm 1\%$  is used as the batch size. Using the same circuit presented in Figure 6 with the voltage source replaced with the MAX6250 precision voltage reference used for the 16-bit ADC.  $R_2$  is then replaced randomly and sorted by the voltage gain measured at the output. Resistors that produced the same voltage were grouped together and then used to reduce variance in the voltage offset.

### 3.4.4 Voltage Reference

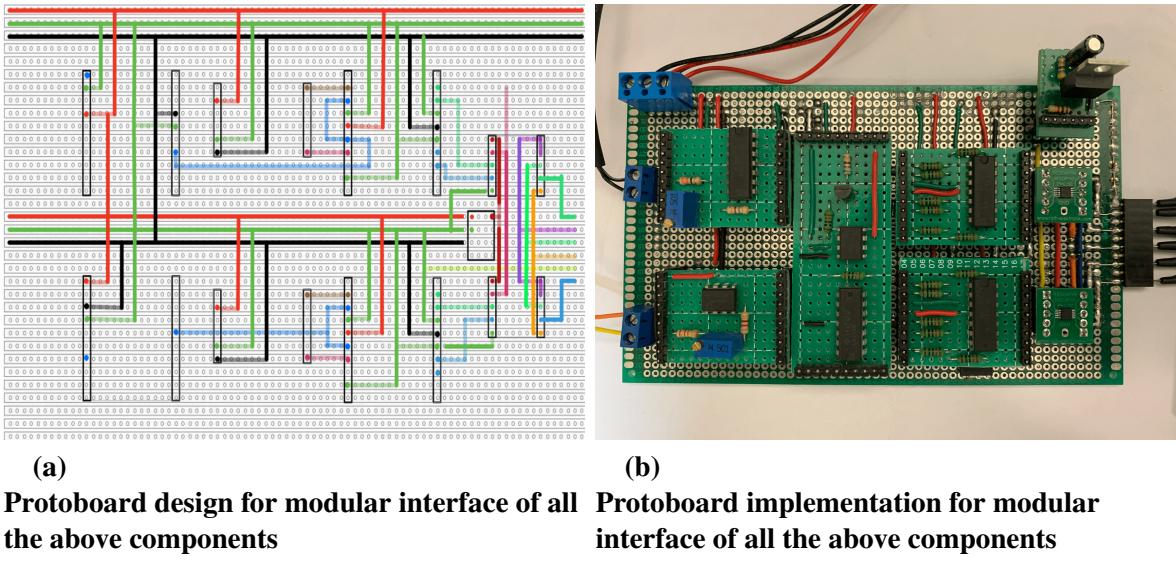
The 2.5 V reference circuit used to create the circuit depicted 9 in Figure 18a was constructed using the same method as was used above. It may be observed that the additional LM324 quad operational amplifier package that is seen in the Figure 18b has been omitted. However, this operational amplifier package is only used to buffer the positive and negative references, thus its omission is of minor importance.



**Figure 18.**  
**Comparison between the design and hardware implementation of the voltage reference circuit**

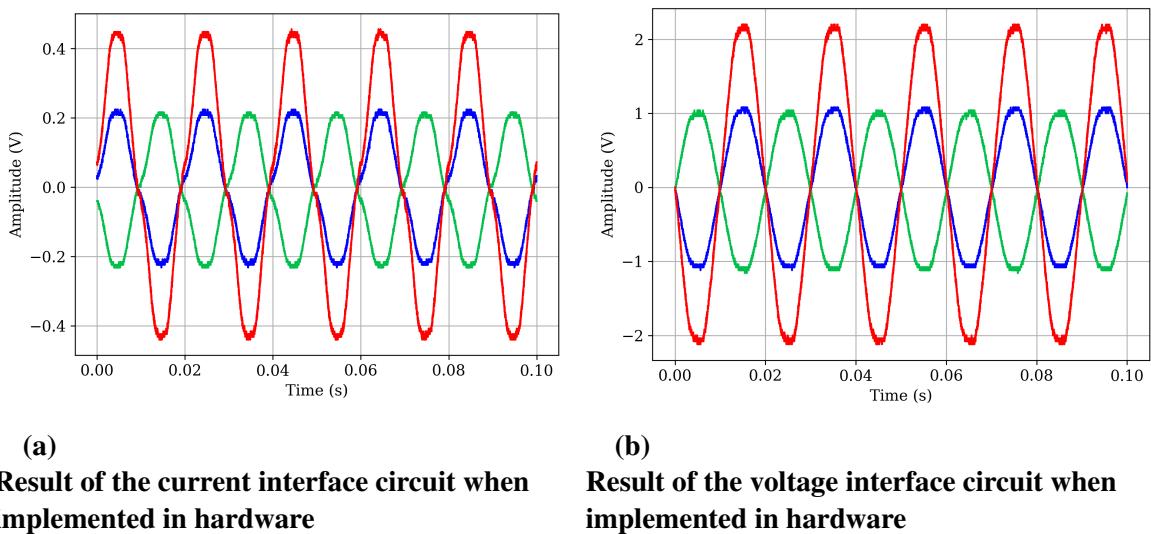
### 3.4.5 Full system integration

The circuit board layout depicted in Figure 19a serves as the base or motherboard to which all the modules connect and interface with one another. The design splits the voltage and current sampling subsystems at the top and bottom, respectfully. This results in a clean layout and no interference between the two independent signals. This division of the signals produces the physical layout seen in Figure 19b - the split is distinct. It can also be observed that the design requires two of the ADC driver modules depicted in Figure 17. This is because the voltage and current signals have separate independent ADCs.



**Figure 19.**  
Comparison between the design and hardware implementation of the modular interface of all the above components

To confirm the that the voltage and current interfaces are working as designed the the outputs of both circuits were captured using an oscilloscope. The results of which can be seen in Figure 20. For both the positive and negative traces ('green','blue') the DC offset was removed for illustration purposes but the offset is not removed from the difference between the two signals ('red'). Clearly depicted in the figures there is a negligible DC offset that comes as a result of differences between the positive and negative outputs from the differential ADC.



**Figure 20.**  
Current and voltage signals captured on an oscilloscope.

## 3.5 Software design

All the software designed below is intended to be deployed on the Beaglebone AI which is based on the AM5729 Sitara processor from Texas Instruments.

### 3.5.1 PRU ADC Interface

The pseudocode below outlines the steps need for the implementation of a bit banging program that is used to trigger a sample on both ADCs simultaneously using the Serial Peripheral Interface (SPI) protocol. The code below is also responsible for transferring the result from both ADCs to a 32-bit shared memory location that can be accessed by other programs using the address of the shared memory.

The need for a custom implementation of the SPI protocol is two-fold. Firstly, the ADC makes use of a non-standard chip select function to trigger a conversion. Secondly, the standard library implementations of the SPI protocol do not offer the data transmission speed or the ability to receive data from two slave SPI devices simultaneously.

The PRU-core on the Beaglebone runs independently of all other systems and has low level control of the Beaglebone pins. This allows for the ADC to perform sampling independently from the programs running on the Beaglebone. The advantage of this implementation is that the sample rate is kept consistent regardless of the processing load.

---

**Algorithm 1** PRU ADC SPI Sample Loop

---

```

while Infinite Loop do
    SPI chip select is set 'high'
    Reset temp1
    The upper 16-bits of temp 1 result of ADC2 copied from the temp variable
    The lower 16-bits of temp 1 result of ADC1 copied from the temp variable
    Delay of 10 µs
    SPI chip select is set 'low'
    Delay of 10 µs
    Temp1 is copied to the shared memory address
    Delay of 10 µs
    Reset temp
    for 16-bits of ADC2 and ADC1 do
        SPI clock is set 'high'
        Delay of 10 µs
        SPI clock is set 'low'
        Delay of 10 µs
        The nth samples from the ADCs are appended to temp
        Delay of 10 µs
    end for
    Delay of 10 µs
end while

```

---

The operations below describe the sequence of commands that need to be issued over the SPI protocol to issue a re-calibrate command.

---

**Algorithm 2** PRU ADC SPI Calibration Protocol

---

```

Reset i
for  $i < 1024$  do
    SPI clock is set 'high'
    Delay of 10 µs
    SPI clock is set 'low'
    Delay of 10 µs
    Increment i by 1
end for
Delay of 10 µs

```

---

### 3.5.2 Sampling Subroutine and Data processing

---

**Algorithm 3** 2's Complement to double array

---

**Require:** Memory address of 32-bit ADC result

```

function C - S A M P L E(Shared Memory Address ,number of samples, voltage,current)
    i = 0
    while i < number of samples do
        if Previous result ≠ Current result then
            Previous result = Current result
        for Voltage do
            Convert two's compliment binary to decimal
            Convert Decimal to floating point in range –5 V to 5 V
            Place result in array
        end for
        for Current do
            Convert two's compliment binary to decimal
            Convert Decimal to floating point in range –5 V to 5 V
            Place result in array
        end for
        i = i + 1
    end if
    end while
end function

```

---

### 3.5.3 Digital Filter

The Table 4 below is a summary of the most popular FIR window filter design functions.

Window Type	Peak Sidelobe Amplitude (Relative, dB)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log(\delta)$ (dB)
Rectangular	-13	$\frac{4\pi}{M+1}$	-21
Bartlett	-25	$\frac{8\pi}{M}$	-25
Hann	-31	$\frac{8\pi}{M}$	-44
Hamming	-41	$\frac{8\pi}{M}$	-53
Blackman	-57	$\frac{12\pi}{M}$	-74

**Table 4.**  
**Table of common FIR Windowing methods and their performance/design metrics adapted from [19].**

- Filter Type = Low Pass

- $F_{Sample} = 49 \text{ kHz}$
- $F_{Pass} = 20 \text{ kHz}$
- $F_{Stop} = 21 \text{ kHz}$

Since the anti-aliasing filter (AAF) does not require an aggressive stopband attenuation factor, a Hanning window is selected. As a result, fewer taps will be required, thereby reducing the computation overhead required to process the filter.

In order to calculate the filter order required to implement the specified Hanning window FIR filter, the Lobe width is defined as the difference between the stopband and passband, as in equation 18

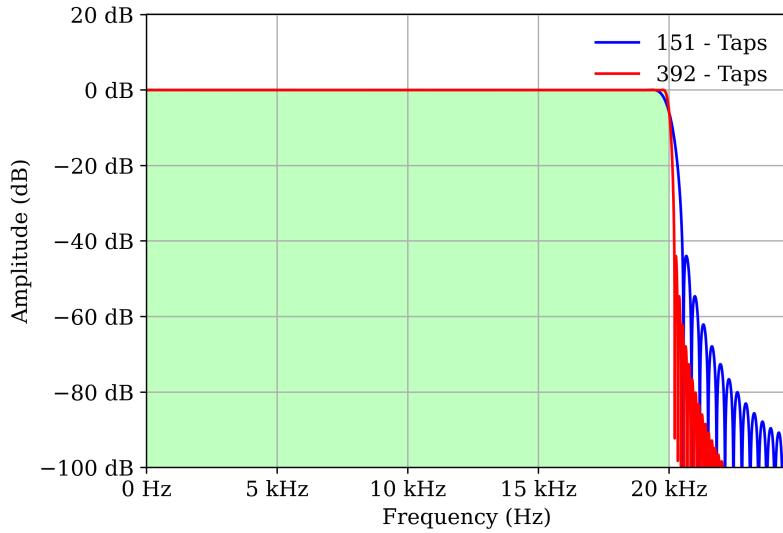
$$Lobe\ Width = (F_{Stop} - F_{Pass}). \quad (18)$$

Where filter order is defined as M, and using the table in Figure 4 above, the following equation is derived:

$$(F_{Stop} - F_{Pass}) = \frac{8 \cdot F_{sample}}{M}, \quad (19)$$

Substituting the desired values for the passband and stopband frequencies into equation 19 above, M is found to have a value of 392. This high filter order comes at a trade off in that it adds a delay to the signal and adding processing overhead.

Since the intended deployment of this filter is on an embedded device where processing power is limited (execution time being of major concern) a more conservative order of 151 is selected. Implementing the low pass AA-filter with the reduced number of taps for this particular application has had a negligible effect on the filter's frequency response, as is shown in Figure 21. The designed filter with the calculated number of taps 'red' and the implemented filter with reduced number of taps 'blue' are both functionally identical to the ideal filter response 'green'.



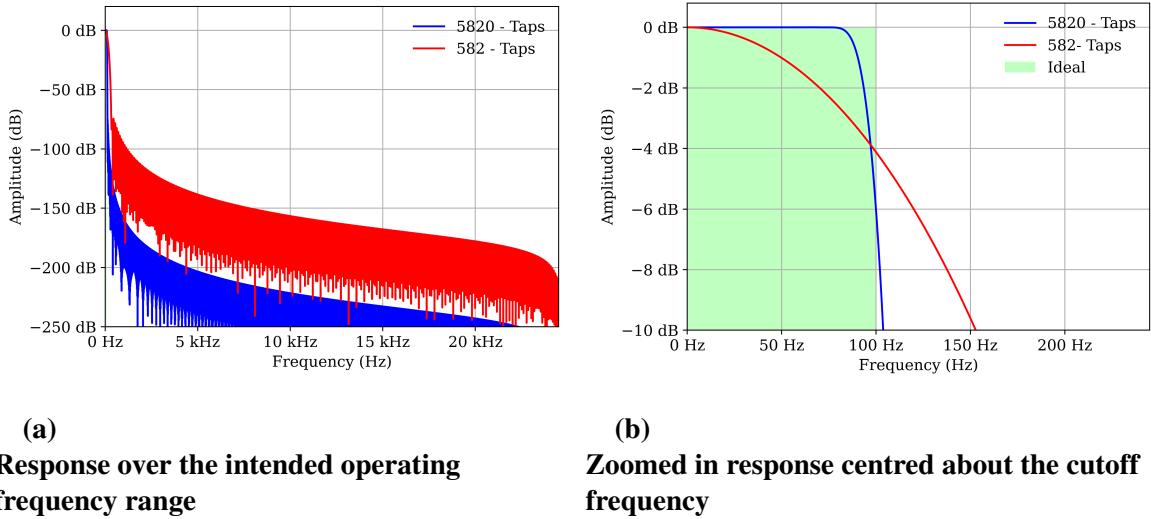
**Figure 21.**  
**Frequency response of the designed AAF**

- Filter Type = Low Pass
- Sampling Frequency = 49 kHz
- Passband = 50 Hz
- Stopband = 100 Hz

Since the sample rate is exceptionally high relative to the cut-off frequency, the Blackman window type is selected for 50 Hz filter using the table above as this has the most significant stopband attenuation of all the above filters. The process followed to calculate the filter order required is almost exactly the same as with the AAF, the only difference being that equation 19 is modified with the new passband and stopband values as well as the new formula for the lobe width. Thus, equation 20 below is derived

$$(F_{Stop} - F_{Pass}) = \frac{12 \cdot F_{sample}}{M}, \quad (20)$$

Substituting the desired values for the passband and stopband frequencies into equation 20 above results in M having a value of 5820. When simulating the frequency response of the designed filter 'blue', and the same filter with an order of 582 'red', with that of the ideal response in Figure 22b, the filter with more taps has a much better frequency response. However, the computation costs far exceed the benefits, especially when comparing it to a filter that requires one tenth that cost. The reason the lower order filter is just as effective in this case is because the intended application is the accurate measurement of real power.



**Figure 22.**  
**Frequency response of the designed low pass 50 Hz FIR filter**

The pseudocode below summarises the functions that are performed on the raw sampled data that was produced by algorithm 3. The signal is first calibrated and scaled from the  $\pm 5$  V to the original signal's respective value. The sampled signal is then passed through the AA-Filter designed in section 3.5.3. The DC-offset is also removed from the signal that results from the Differential ADC input driver design. The signal is then windowed to have 3-consistent periods over which the rest of the signal processing can be performed. The reason only 3-consistent samples were deemed sufficient and not the entire signal is that the latency between when the signal changed and the device reflecting the change is 1 s. Thus, the small amount of time by which the signal is delayed will have a negligible effort on the accuracy of the power and energy measurements.

---

#### Algorithm 4 Data pre-processing

---

Scale the  $\pm 5$  V to the actual voltage signal value  
 Scale the  $\pm 5$  V to the actual current signal value  
 Apply AA-Filter to voltage signal  
 Apply AA-Filter to current signal  
 $\text{Voltage signal} = \text{Voltage signal} - \text{mean}[\text{Voltage signal}]$   
 $\text{Current signal} = \text{Current signal} - \text{mean}[\text{Current signal}]$   
 Window Voltage signal to 3 periods  
 Window Current signal to 3 periods

---

### 3.5.4 Power Calculations

One of the most important features to extract for appliance classification is the amount of apparent and real power consumed, these being calculated as follows

$$\text{Instantaneous Power} = p(t) = v(t) \cdot i(t), \quad (21)$$

$$\text{Average Power} = P = \frac{1}{T} \int_0^T p(t) dt, \quad (22)$$

$$X_{rms} = \sqrt{\frac{1}{T} \int_0^T x^2 dt}, \quad (23)$$

$$\text{Apparent Power} = S = V_{rms} \cdot I_{rms}, \quad (24)$$

respectively [14].

The algorithms 5 - 8 are the pseudocode that explain the process that is needed to implement the above equations in any programming language. As presented in the algorithms, the continuous signals are first made discrete to the sample based equivalent formula.

---

#### Algorithm 5 Instantaneous Power Calculation

---

```
function INST_POWER( $V, I$ )
    Calculate the instant power:  $P_{inst} = V \cdot I$ 
    Apply 50Hz filter:  $P_{inst}$ 
    return  $P_{inst}$ 
end function
```

---



---

#### Algorithm 6 Real Power Calculation

---

```
function REAL_POWER( $P_{inst}$ )
    Find sum of the instant power:  $S_P = \sum_{i=0}^n P_{inst_i}$ 
    Find the mean of the sum :  $\bar{S}_P$ 
     $P = \bar{S}_P$ 
    return Real Power :  $P$ 
end function
```

---



---

#### Algorithm 7 RMS - Root mean square calculation

---

```
function RMS( $x$ )
    Square the input array:  $x^2 = x \cdot x$ 
    Find sum of the squared input array:  $S_{x^2} = \sum_{i=0}^n x_i^2$ 
    Find the mean of the sum :  $\bar{S}_{x^2}$ 
    Calculate the square-root value of the mean:  $RMS = \sqrt{\bar{S}_{x^2}}$ 
    return  $RMS$ 
end function
```

---

**Algorithm 8** Apparent Power Calculation

---

```

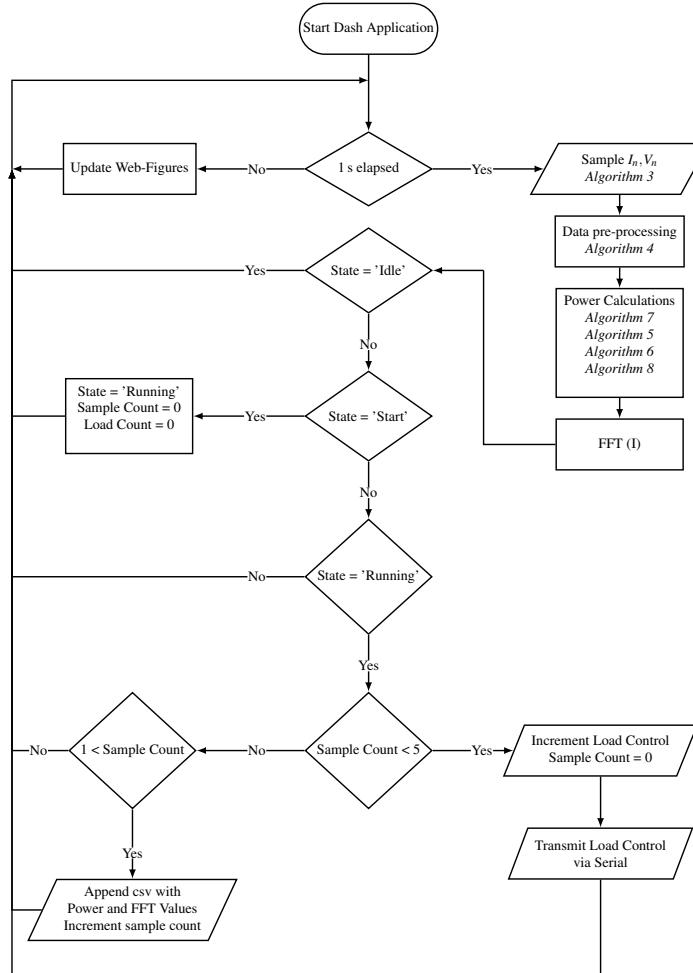
function APPARENT_POWER( $V_{rms}, I_{rms}$ )
     $S = V_{rms} \cdot I_{rms}$ 
    return Apparent Power : $S$ 
end function

```

---

**3.5.5 Data Collection**

The pseudocode below describes the automatic data collector's function to be the collection of the data required to train the decision trees used to create the random forest. The code makes use of an external 8-way relay module connected to an Arduino mega. The Arduino mega is loaded with a simple program that converts an 8-bit serial value to relay commands. The serial commands are issued by the Beaglebone AI through a USB to serial interface. The entire program is an amalgamation of all the functions that are controlled by a dash application.



### 3.5.6 Decision Tree Design

A decision tree is a series of decisions that map the probable outcome of events based on the information presented. The tree is constructed out of a root decision node that connects to other decision nodes. These connections continue until they reach a decision - also known as the leaf or terminal node. A decision tree can be categorical or continuous. Since the application is to categorise an appliance's current state, all the design choices were made for a categorical decision tree. In order to create these decision nodes the tree must be trained. There are a number of algorithms that are used to train decision trees (ID3 , C4.5 C5.0). This process of training is accomplished by fitting the tree to a training set of data for which the classification is already known. This is done by creating decisions that split the training set continuously until the split results in a group that only contains one type of classification data.

There are two measures that are available to measure how good a split is: the Gini index and the Shannon entropy. In [20], Aznar concludes that the two methods perform very similarly, although entropy results in slightly higher accuracy at the cost of it requiring more computation time. Based on his findings it was decided that using entropy (which can be calculated as

$$H(x) = - \sum_{i=1}^n P_i \log_2(P_i), \quad (25)$$

reflectively [21]) would be a better choice for the intended application since accuracy is the most important design criteria. The pseudocode describing the general implementation of the formula 25 is relatively simple and is only comprised of a few repetitive steps, as seen below

---

#### Algorithm 9 Entropy Calculation

---

```

Calculate the total occurrences
Calculate the total occurrence of unique categories
Calculate the probability of each unique categories with regard to the total occurrences
Reset the entropy variable  $H(x)$ 
for Every probability value  $P_i$  do
     $H(x_i) = -(P_{x_i} \cdot \log_2 P_{x_i})$ 
     $H(x) = H(x) + H(x_i)$ 
end for
return The entropy variable  $H(x)$ 

```

---

The algorithm described the psudocode presented below are the general steps followed to calculate the information gained per split.

---

**Algorithm 10** Information Gain

---

**Require:** Threshold value  $Threshold$   
**Require:** Total entropy before the split  $T_{entropy}$   
**Require:** Total number entries in the dataframe  $T_{entries}$

Split dataframe by the threshold value into left and right dataframes  
Calculate the weighted entropy of the left split  
Calculate the weighted entropy of the right split  
Sum the two weighted entropies  $Sum_{entropy}$   
Information gain =  $T_{entropy} - Sum_{entropy}$   
**return** information gain

---

The pseudocode below describes the process followed to find the best split from the features provided in the training data. Each feature is extracted and then the average between the two adjacent values in the feature is calculated. The data is then split and the information gained from the split calculated. This is repeated for every feature until the best split is discovered.

---

**Algorithm 11** Best Split

---

**Require:** Dataframe where column 1 is category the rest are id features

Calculate the total entropy using column 1  $T_{entropy}$   
Calculate the total number entries in the dataframe  $T_{entries}$

**for** every id feature **do**

    Sort dataframe in ascending order by id feature  
    Create a threshold array of averages between two adjacent id features

**for** every threshold value **do**

    Calculate the information gain for the threshold value

**end for**

**if** max information gain < current information gain **then**

    Store id feature  
    Store threshold value  
    Store current information gain

**end if**

**end for**

**return** id feature,threshold value,max information gain

---

The pseudocode below describes the process of creating and training a decision tree from a data set. The function makes use of recursion to calculate the best splits using the information gained pseudocode above. This continues until either the max depth maximum number of decisions is reached or the data has been completely split. The split feature and value are then stored as a python dictionary. This dictionary can later be used to process new data and make load predictions.

---

**Algorithm 12** Decision Tree creation and training

---

```

function TRAIN(data,max depth,information gained)
  if max depth has not been reached then
    Call the best split function with the current dataframe
    if information gained is larger than the minimum information gain then
      Call the make split function which returns split dataframes
      Create the split question - feature and value
      Call the TRAIN Function on the split dataframes
      if if the split results in the same number of category's being identified then
        Append only the one question node to Tree
      else if if the split results in different category's being identified then
        Append all question nodes to Tree
      end if
    else if information gained is smaller than the minimum information gain then
      Return maximum number of category's
    end if
  else if max depth has been reached then
    Return maximum number of category's
  end if
  return Tree
end function

```

---

The pseudocode below describes the methods used to implement the random forest algorithm used for training of trees used in a random forest algorithm.

---

**Algorithm 13** Bagging/Bootstrap aggregating

---

**Require:** Dataframe where column 1 is category the rest are id features

```

  Shuffle the dataframe
  Split the dataframe into a train and test subsets
  for number of desired trees in forest do
    Create a random with replacement sample that has 70% of the training sample size
    Train the decision tree on the random sample data
  end for
  Combine the created trees into a forest the output of which is the majority decision of all
  the trees in the forest
  return Majority decision

```

---

## 3.6 Software implementation

### 3.6.1 Decision Tree Deployment

The function described below is the method used to make identification and append the result to a binary list that can be used to display and verify the accuracy of the detected loads.

---

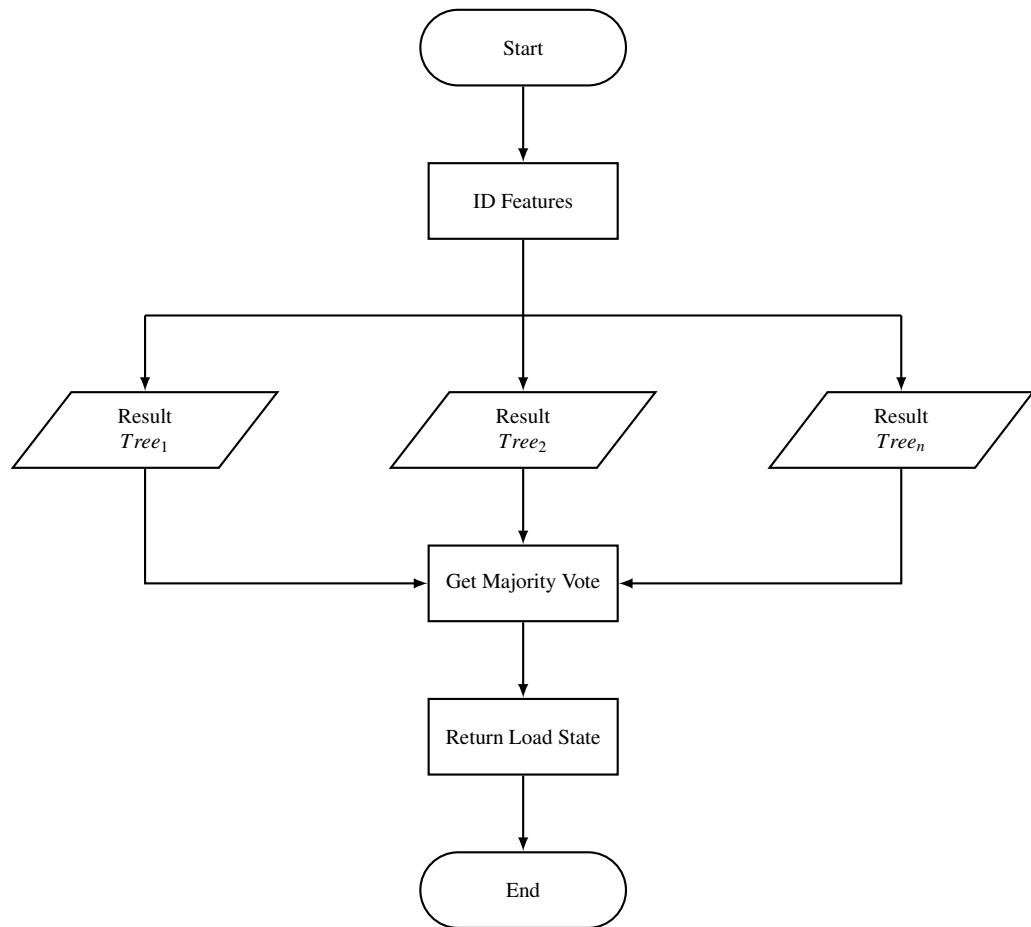
#### Algorithm 14 Decision tree forest used for classification of loads

---

```
function CLASSIFICATION(ID_Features,Decision_Trees,Load_ID)
    for number of unique loads do
        Call the function presented in Figure 23
        Append the load Load_ID with the state of the current load
        Load_ID['Load'] = status
    end for
    return Load_ID
end function
```

---

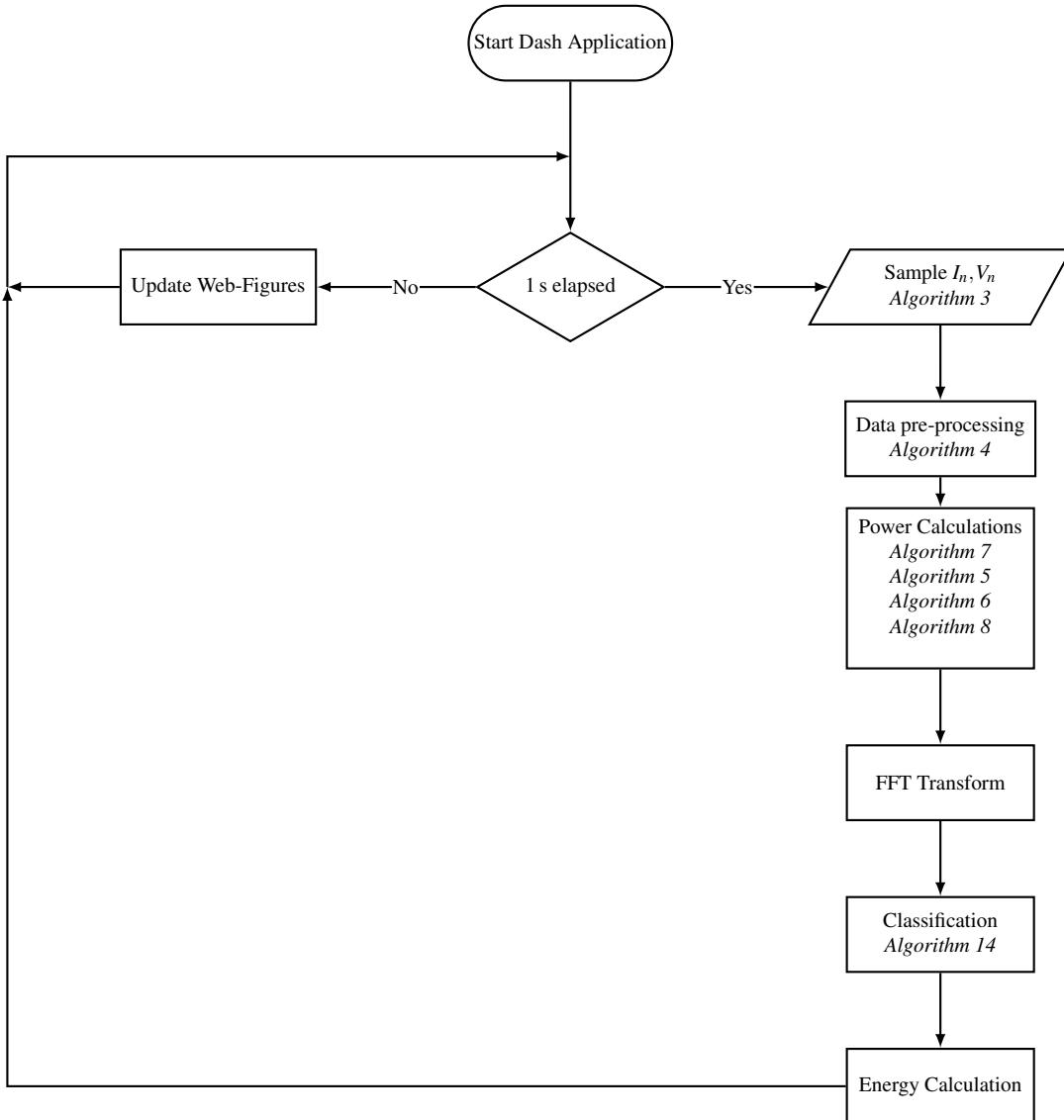
The code described by the flow diagram in Figure 23 is the process used to make a decision based on the majority vote of all the individual tree results.

**Figure 23.**

**Flow diagram of the describing the implementation random forest algorithm where any number of trees can be added.**

### 3.7 Final system integration and testing

The system described by the Figure 24 is the order of integration of how the individual designed pieces of software are interfaced together to create a full functioning system that is able to measure the energy and identify the loads that currently using said energy.



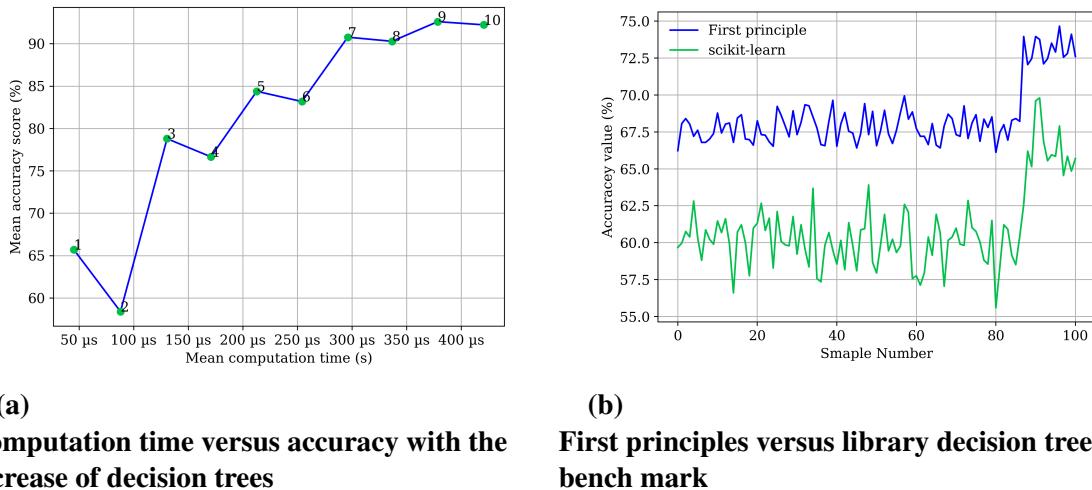
**Figure 24.**  
**Full software flow diagram depicting the integration of the sub-process.**

## 3.8 Statistical analysis

### 3.8.1 Decision Tree Performance

As with all software the larger and complex the function the longer the time needed to compute the function. The Figure 25a was used to measure the the accuracy benefit of more tress versus the computation cost. As can be clearly seen the worst combination is having only two tress this is because if there is no consensus the base case is returned. It can also be noted that the computation time increases at a linear rate but the performance gain has diminishing

returns the more trees there are.



**Figure 25.**  
**Decision tree performance measurements used in the design**

Using the above information the following process was used to create all the trees and where necessary a random forest algorithm was used:

1. The main data-set is split and copied into a data-set for every load.
2. This data-set was split into a training and evaluation data.
3. A decision tree was then trained and evaluated.
4. If the tree was below the threshold accuracy of 90 % the depth was increased.
5. The steps in 3 and 4 are repeated till the accuracy is achieved or the depth is exceed.
6. If the depth is exceed algorithm 13 is used to increase the accuracy.

Following the steps outlined above the Table 5 below shows the decision tress and random forests that are combined to form the final load detection algorithm.

Decision tree design metrics			
Load Name	Number of features	Depth	Number of Trees
Fan Speed 1	32	10	3
Fan Speed 2	32	10	3
CFL	10	4	1
2W	25	10	3
4W	25	10	3
Toaster	30	3	1
70W	20	8	1
2 x 70W	20	8	1

**Table 5.**  
**Decision tree design metric used in the final detection algorithm**

## 4. Results

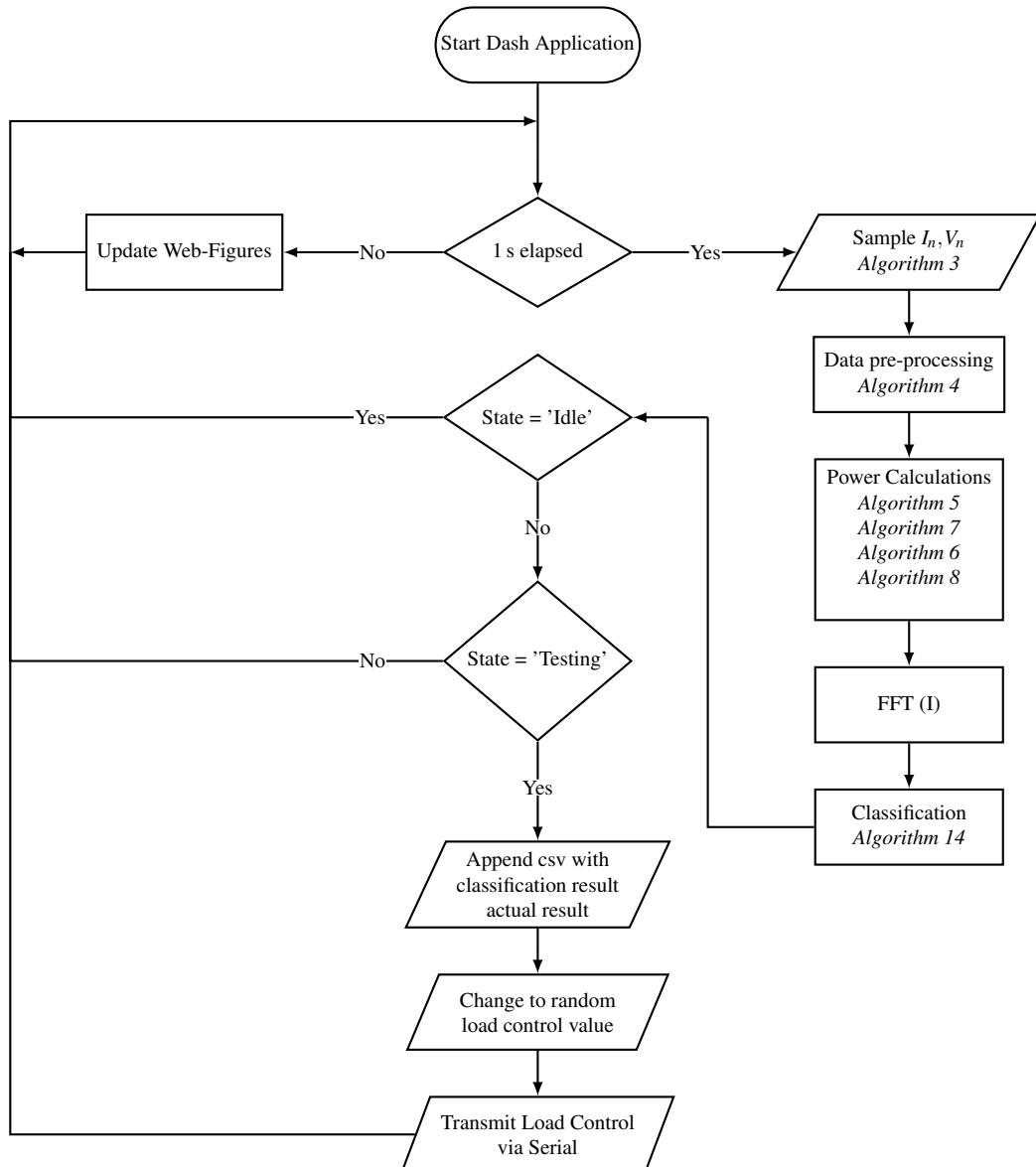
---

### 4.1 Summary of results achieved

Intended outcome	Actual outcome	Location in report
<b>Core mission requirements and specifications</b>		
The device must be able to distinguish between any combination of 5 - 10 household appliances that can be in operation simultaneously at unpredictable times, with a true positive rate (TPR) of 95%	The system should be able to identify 6 unique loads simultaneously	Section 4.3.1
The error in the power measurement of any combination of the household appliances that draw between 1.2 – 40 A connected should not exceed that of $\pm 2.5\%$	The error of the power measurement was discovered to be no greater than $\pm 1\%$	Section 4.3.2
Able to measure a voltage in the range of $230V \pm 10\%$ at 40A	The system was able to accurately measure over a voltage range of 207 – 240V The system was never exposed to larger voltages in the labs	Section 4.3.3
The device must be able to detect a load that consumes more than 2W of power	The system is able to correctly identify an LED which is rated to draw 2W	Section 4.3.1
The status of the energy consumption of any connected load should be updated in under 2 seconds.	The system updates all information once per second	Section 4.3.4
<b>Field condition requirements and specifications</b>		
Only the specified loads which the system is designed to identify should be connected	The system was able to correctly identify the loads specified.	Section 4.3.1
The combinations of loads connected to the system should not exceed 8000W.	The cable that the prototype used is only rated for 16 A. This limit was not exceeded as a safety precaution but proof of sufficient headroom in the design is shown.	Section 4.3.5

**Table 6.**  
**Summary of results achieved.**

## 4.2 Tools developed for qualification testing



**Figure 26.**  
**Software flowchart describing the evaluation program implemented on the Beaglebone AI for qualification testing.**

## 4.3 Qualification tests

### 4.3.1 Qualification test 1: Load identification accuracy test

#### *Objectives of the test or experiment*

The objective of the test described below is to verify whether the load detection method of using a random decision tree forest is able to correctly identify the current state of any connected load with a true positive rate of 95 %.

#### *Equipment used*

- Beaglebone AI
  - \* PRU core is running the SPI subroutine
  - \* Arm core is running the test software described in Figure 26
- Mains electricity sampling module *Figure 19*
- Arduino Mega - *Running load control program*
- FTDI Chip TTL-232RG-VSW5V-WE (RS 730-0155) +5V UART signalling
- 8-way Relay module
- Test loads:

Summary of load parameter				
Load Number	Appliance	Power consumption (W)	Manufacture	Model Number
1	LED Light	2	Lightworx	C352WCE27
2	LED Light	4	Lightworx	C354WCE27
3	CFL Light	20	Eurolux	G330
4	Halogen Light	70	Eurolux	G877BP
5	Halogen Light	70	Eurolux	G877BP
6	Fan	15	Safeway	PIA1821
7	Toaster	850	Pineware	PET201

**Table 7.**  
**Summary of the loads used in the testing of the detection algorithm.**

#### *Test setup and experimental parameters*

The Beaglebone and Arduino are loaded with the required test software. The loads are connected to the relay module. The specific loads must be connected in the correct sequence to ensure that the predicted load and the actual load correlate. This specific sequence of loads is to ensure consistency in the multiple runs performed. This ensures that the experiment has two parameters: the actual load which is controlled by the relay and the identified load.

#### *Steps followed in the test or experiment*

- Step 1: The test program and the control dashboard is accessed through the local IP Address presented in the terminal.
- Step 2: The voltage and current waveforms are checked to confirm that the PRU and test program is able to receive information and that the PRU is running.
- Step 3: The start test is then selected *Figure 36*.
- Step 4: The test program changes the load configuration and issues the command to relay module every 5-seconds.
- Step 5: The predicted and actual load states are appended to a csv file.
- Step 6: The csv file is then downloaded off the Beaglebone and imported into python.
- Step 7: The data is then plotted as a confusion matrix and the performance metrics are calculated.
- Step 8: Steps 1-7 are repeated at different times of the day and on different days to ensure that the results are consistent.

### *Results or measurements*

Figure 27 is a combination of all the results from the multiple runs of the test described by step 7 above.

		Predicted Class														
		CFL (20W) OFF	CFL (20W) ON	Fan Speed 1 OFF	Fan Speed 1 ON	Fan Speed 2 OFF	Fan Speed 2 ON	Halogen (70W) OFF	Halogen (70W) ON	LED (2W) OFF	LED (2W) ON	LED (4W) OFF	LED (4W) ON	Toaster OFF	Toaster ON	
True Class	CFL (20W) OFF	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	CFL (20W) ON	0.01	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fan Speed 1 OFF	0.00	0.00	0.93	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fan Speed 1 ON	0.00	0.00	0.17	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fan Speed 2 OFF	0.00	0.00	0.00	0.00	0.95	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fan Speed 2 ON	0.00	0.00	0.00	0.00	0.13	0.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Halogen (70W) OFF	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Halogen (70W) ON	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	LED (2W) OFF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.06	0.00	0.00	0.00	0.00	0.00
	LED (2W) ON	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.94	0.00	0.00	0.00	0.00	0.00
	LED (4W) OFF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.10	0.00	0.00	0.00
	LED (4W) ON	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.89	0.00	0.00	0.00
	Toaster OFF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
	Toaster ON	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Accuracy=94.8498%; Miss Classification=5.1502%

**Figure 27.**  
**Confusion matrix depicting the results of the implemented load prediction algorithm.**

### *Observations*

As can be seen in Figure 27 the developed algorithm was able to predict the correct state

of all the loads with 94.85 % accuracy. The main loads that the detection method had more trouble detecting were (as predicted) the loads with small power draws. What is interesting to note is that the detection accuracy of the 4 W LED was lower than that of the 2 W LED. The other interesting observation is that the power used by the load did not necessarily impact the accuracy of the detection (with the obvious exception of the toaster). This is evident when comparing the results from the 70 W Halogen and the 20 W CFL light, with the latter having much higher identification accuracy.

#### **4.3.2 Qualification test 2:Power accuracy test**

##### *Objectives of the test or experiment*

The objective of the test described below is to verify that the designed system meets the requirements of a class 2 energy meter. To meet these requirements the device must be able to measure and report the power draw with an error of no greater than  $\pm 2.5\%$  when the combined current draw of the loads connected exceeds that of 1.5 A.

##### *Equipment used*

- Beaglebone AI
  - \* PRU core is running the SPI subroutine
  - \* ARM core is running the test software described in Figure 26
- Multimeter: Techgear TG451DL
- Oscilloscope: RS-PRO IDS-2204E
- DeLonghi 1800 W heater.
- Mains electricity sampling module *Figure 19*
- Web-browser

##### *Test setup and experimental parameters*

The Beaglebone is loaded with the test software. The test application is then accessed via a web-browser. The heater is then turned on to max heat for 1 minute. This is to ensure that the heater reaches a steady state before the experiment begins as the power draw is far beyond the specified 1800 W. By doing this when the test is started the only parameter that is being measured is the power upon which the accuracy depends.

##### *Steps followed in the test or experiment*

- Step 1: The 1.8 kW heater is turned on and left running for 1 minute.
- Step 2: The test is started and the results are recorded as soon as the website that is hosted by the Dash application is accessed via the local IP Address presented in the terminal.
- Step 3: The following checks are performed every 5 min to ensure that the results from the test are valid.
  - \* The current waveform as it displays on the GUI is verified with the waveform displayed on the Oscilloscope.

\* The voltage and current measurements are confirmed using the multimeter.

Step 4: Once the test is completed the csv is transferred into python.

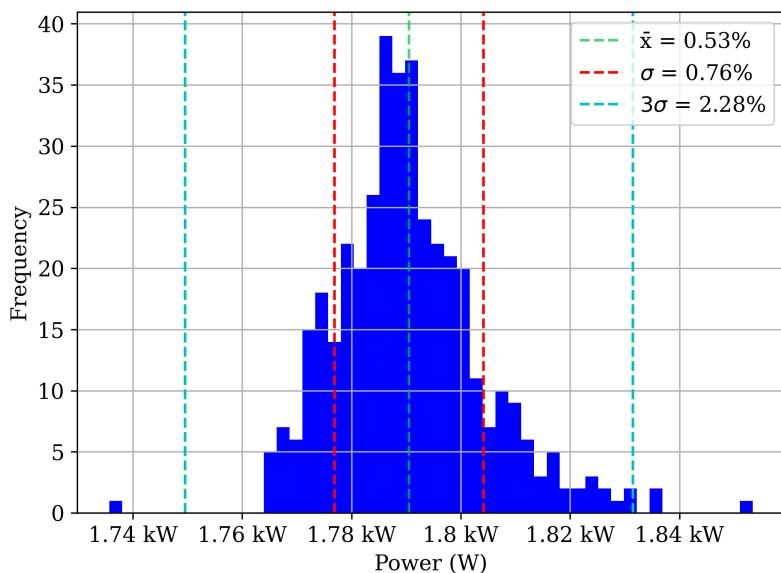
Step 5: A histogram that depicts the measured real power from the test is generated.

Step 6: The measured power fluctuations are converted to an error percentage.

Step 7: The results are depicted in a figure for simple analysis.

### *Results or measurements*

The measurements contained in the csv are the real power values that the device reported after being tested.



**Figure 28.**

**Histogram depicting variations in the power measurement of a known load.**

### *Observations*

As can be seen from the legend depicted in Figure 28, the power measurement had an average mean error of 0.53 % with one standard deviation of 0.76 %. From these two metrics it can be derived that 68.26 % of the measured results never exceeded an error of 0.76 % with 99.72 % of all measurements never having an error that exceeds 2.28 %, this being below the designed metric. This means that the device meets - and in some aspects exceeds - the requirements for it to be classified as a class 2 energy metering device.

#### **4.3.3 Qualification test 3: Voltage variation test**

##### *Objectives of the test or experiment*

The objective of the test described below is to verify that the designed systems complies

with the IEC 60038 voltage standard. This standard stipulates that the voltage supplied to households in South Africa is kept in the range of  $230\text{ V} \pm 10\%$ . To meet this requirement, the system must be able to measure the voltage as it fluctuates with the range stipulated above.

#### *Equipment used*

- Beaglebone AI
- Techgear TG451DL Multimeter
- Deye SUN-8K-SG01LP1-EU Hybrid inverter
- Custom variation of the code described in section 4.2 where the voltage and time values are appended to a csv instead of the identified load information.
- Web-browser

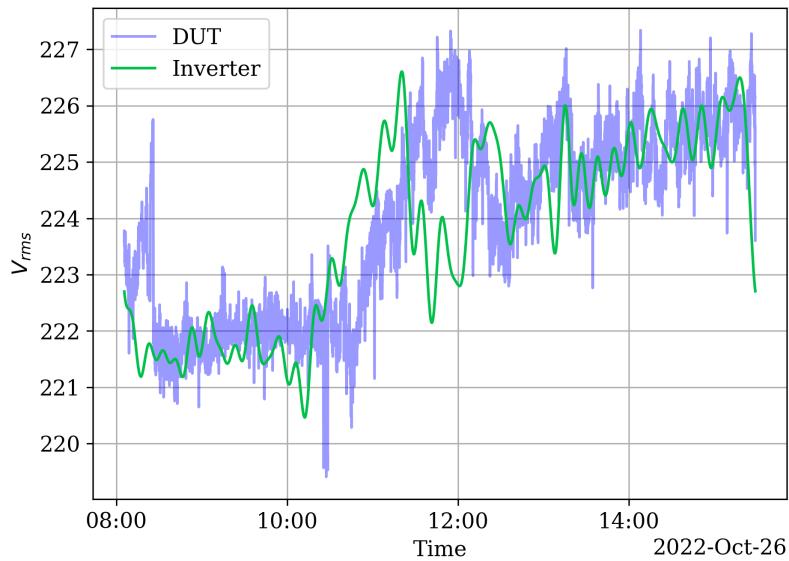
#### *Test setup and experimental parameters*

The device is setup with no load connected. The Beaglebone is loaded with the test software. The test application is then accessed via a web-browser. The measured voltage is confirmed to be correct before the test is ready to begin. The only parameter being measured here is voltage with respect to time.

#### *Steps followed in the test or experiment*

- Step 1: The 1.8 kW heater is turned on and run for 1 minute.
- Step 2: The test starts and results are recorded as soon as the website that is hosted by the Dash application is accessed via the local IP Address presented in the terminal.
- Step 3: The test is run for a duration of 30 min, during which the following checks are performed every 5 min to ensure that the results from the test are consistent/valid:
  - \* The current waveform as displayed on the GUI is verified with the waveform displayed on the Oscilloscope.
  - \* The voltage and current measurements are confirmed using the multimeter.
- Step 4: Once the test has concluded, the csv is transferred into python.
- Step 5: A histogram that depicts the measured real power from the test is generated.
- Step 6: The measured power fluctuations are converted to an error percentage.
- Step 7: The results are depicted in a figure for simple analysis.

*Results or measurements* The two traces in the Figure 29 and the report voltage from the inverter and the voltage from the device under test.

**Figure 29.**

**Comparison between the voltage measured by the inverter and the voltage measured by the device**

*Observations* As seen in the results depicted above, the voltage reported by the device follows the same curve as that of the inverter. The only exception to this trend occurred during a scheduled power failure. While this did effect the trend, the device was able to follow the trend correctly again once the normal power was restored. It can also be noted that, due to more frequent updates, the DUT voltage appears noisy when it is in fact more accurate, as confirmed by the spot checks done with the multimeter.

#### 4.3.4 Qualification test 4:Update latency test

##### *Objectives of the test or experiment*

The objective of the test described below is to verify that the latency between the actual state chaining of the load and the reported state of the load does not exceed that of 2 seconds.

##### *Equipment used*

- Beaglebone AI
  - \* PRU core is running the SPI subroutine
  - \* ARM core is running the test software described in Figure 26
- Mains electricity sampling module *Figure 19*
- Web-browser

*Test setup and experimental parameters* The device is setup with no load connected. The Beaglebone is loaded with the test software. The test application is then accessed via a web-browser. The main program with load identification is used to ensure realistic cpu load. The time of each measurement is taken. Thus the experimental parameter's update time is only dependant on how long the system takes to report an update.

#### *Steps followed in the test or experiment*

Step 1: The 1.8 kW heater is turned on and left running for 1 minute.

Step 2: The test is started and results are recorded as soon as the website that is hosted by the Dash application is accessed via the local IP Address presented in the terminal.

Step 3: The test is run for a duration of 30 min, during which the following checks are performed every 5 min to ensure that the results from the test are consistent (valid):

- \* The current waveform as displayed on the GUI is verified with the waveform displayed on the Oscilloscope.
- \* The voltage and current measurements are confirmed using the multimeter.

Step 4: Once the test has concluded the csv is transferred into python.

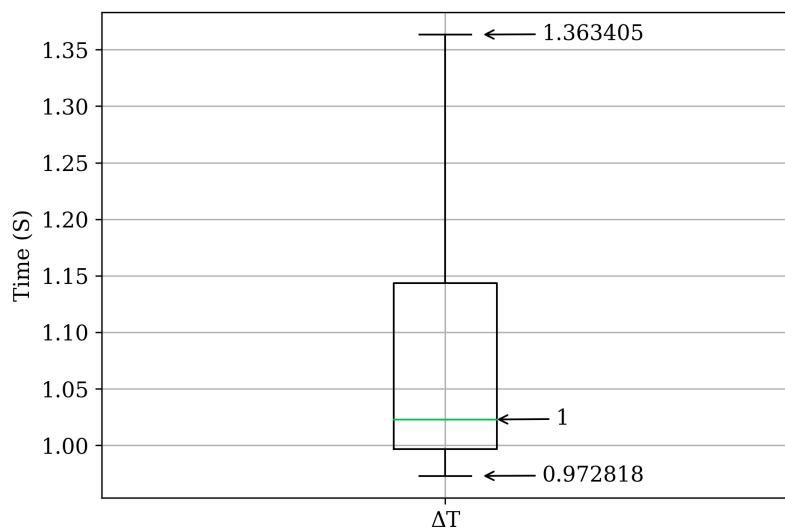
Step 5: A histogram that depicts the measured real power from the test is generated.

Step 6: The measured power fluctuations are converted to an error percentage.

Step 7: The results are presented as a figure for simple analysis.

#### *Results or measurements*

The box and whisker diagram depicted in the Figure 30 shows the difference between the update times extracted from the csv.



**Figure 30.**

**Box and whisker plot depicting the variation in the measurement update latency.**

### *Observations*

As can be seen in the results above, the update time has an average of one second and a maximum value of 1.3 seconds. This means the device meets specifications as the latency between updates never exceeded the specified two seconds as per the design requirements.

### **4.3.5 Qualification test 5:Maximum load test**

#### *Objectives of the test or experiment*

The objective of the test described below is to verify that the designed system is able to measure up to the rated max combined power draw of 8000 W.

#### *Equipment used*

- Beaglebone AI
  - \* PRU core is running the SPI subroutine
  - \* ARM core is running the test software described in Figure 26
- DeLonghi 1800 W heater
- Mains electricity sampling module *Figure 19*
- Web-browser

#### *Test setup and experimental parameters*

The Beaglebone is loaded with the test software. The test application is then accessed via a web-browser. The heater is then turned on to max heat for 1 minute. This is to ensure that the heater reaches steady state before the experiment begins as the power draw is far beyond the specified 1800 W. In doing this when the test is started the only parameter that being measured is the power upon which the headroom of the ADC needs to be measured against.

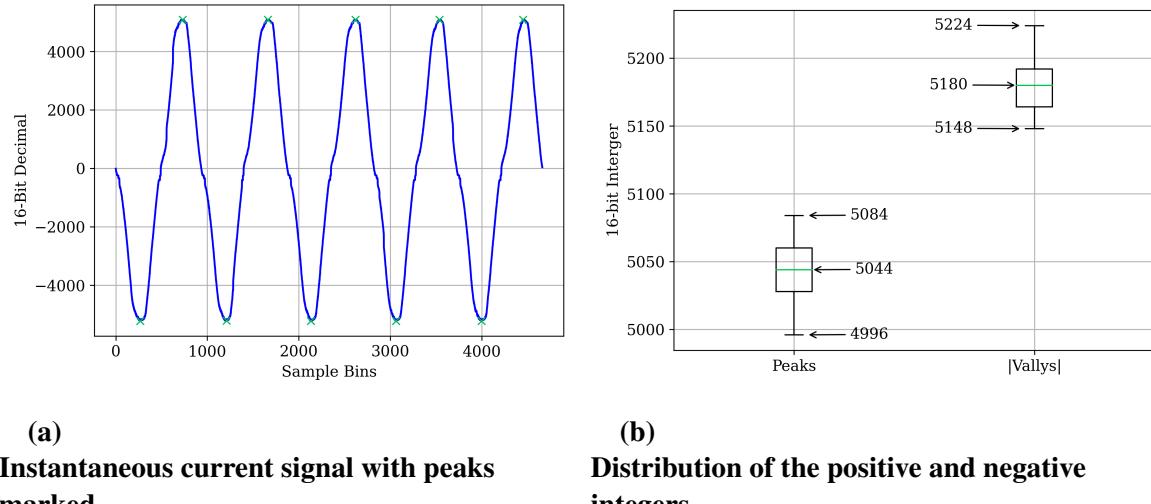
#### *Steps followed in the test or experiment*

- Step 1: The 1.8 kW heater is turned on and left running for 1 minute.
- Step 2: The test is started and results recorded as soon as the website that is hosted by the Dash application is accessed via the local IP Address presented in the terminal.
- Step 3: The test is run for a duration of 30 min, during which the following checks are performed every 5 min to ensure that the results from the test are consistent (valid)
  - \* The current waveform as displayed on the GUI is verified with the waveform displayed on the Oscilloscope.
  - \* The voltage and current measurements are confirmed using the multimeter.
- Step 4: Once the test has concluded the csv is transferred into python.
- Step 5: A histogram that depicts the measured real power from the test is generated.
- Step 6: The measured power fluctuations are converted to an error percentage.

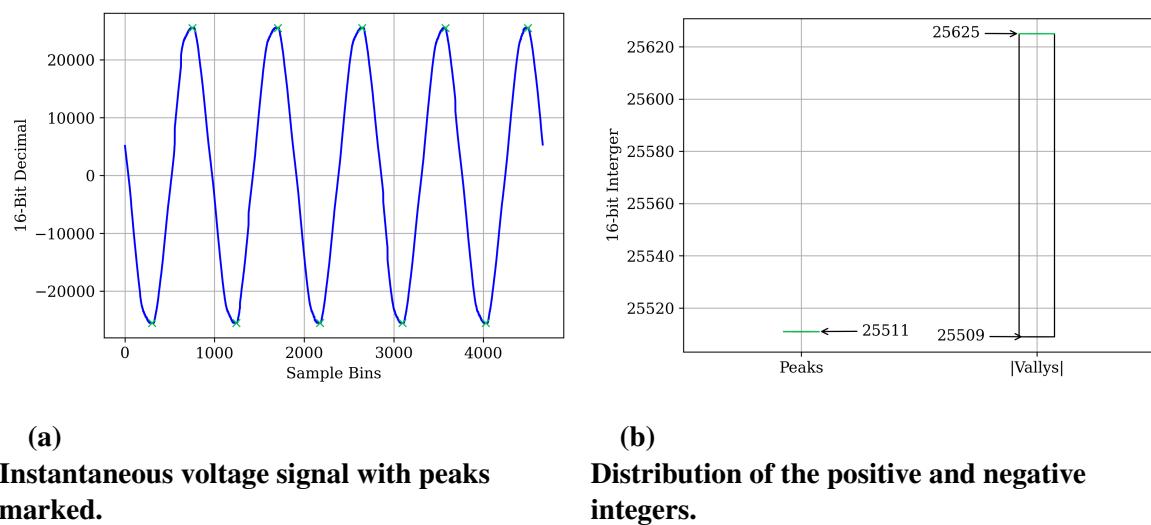
Step 7: The results are presented as a figure for simple analysis.

#### *Results or measurements*

The Figure 31 and Figure 32 show the raw binary value produced by the ADC converted to an integer value. The peaks and troughs are all detected and the statistics are displayed on the right of the raw signal.



**Figure 31.**  
Results used to confirm the available ADC headroom.



**Figure 32.**  
Results used to confirm the available ADC headroom.

*Observations*

As can be seen from Figure 31b, the largest integer that represented the known 1800 W load is 5224. Since the ADC is a differential ADC, the number of distinct positive and negative values is  $2^{15}$ . In dividing the the max integer with the known number of steps, the measurement is 16 % of the total range. Repeating this calculation with the 1800W of the load and 8000W gives a result of 22 %. Thus, the system clearly has sufficient headroom to measure combined loads that can draw a total load of 8000 W.

## 5. Discussion

---

### 5.1 Interpretation of results

The results presented in section 4 confirm that the device is able to identify the current state of loads correctly on the connected circuit. In other words, this result confirms that the design functions as intended. However, it must be noted that the tests were only performed in one location, thus a change in location with different voltage and noise may have an impact on these results.

The results also indicate that the detection accuracy of the result is not solely dependant on the size of the power draw. This becomes evident when comparing the true positive detection rates of the CFL and the Halogen lights as the CFL was correctly identified 100 % of the time.

The results also indicate that the latency between when loads are identified never exceeds two seconds. This ensures that the total reported energy consumed by the identified loads is accurate. The error in the power measurement was also proven to never exceed that of the 2.5 %.

### 5.2 Critical evaluation of the design

#### 5.2.1 Strong points of the current design

The design presented above was able to meet or exceed the design requirements summarised in Table 6. This is primarily the result of implementing a garbage-in, garbage-out design philosophy. That is to say, if the data fed into a device is poor, the output or result produced by the device is also expected to be poor. With this in mind, a large portion of the design process involved designing the data acquisition system. The design parameters were tightened resulting in more significant margins of error when the design was implemented in hardware. Both the ADCs were linked by their chip-select pins, allowing samples of both the current and voltage signals to coincide. Consequently, no phase correction was needed in the software to compensate for a delay.

#### 5.2.2 Aspects to be improved

Although (as mentioned) the device met all the stipulated design requirements, some design decisions can still be improved.

Data acquisition accuracy was the main design focus during the design phase. While this resulted in a design which was quite accurate, little to no attention was given to energy consumption. As a result, the two 9V batteries that power the circuitry had to be replaced frequently. A smaller safety transformer is another design modification that would drastically

improve the operating power of the device. The current design uses a transformer rated to output 3 A. This results in a constant 5 W of power being consumed while the device is in operation.

The most notable aspect which can be improved upon is the sample errors that occur as a result of interfacing a function coded in C and called from python to read a memory location which is being altered by another program. While the C code and the PRU sampling code are error-free, sample errors can result in the false identification of some loads when integrated into python.

The main program that runs all the software could be made to perform identifications and keep track of the energy consumption while the GUI was not open on a web browser.

### **5.2.3 Design functioning limitations**

The specific implementation of the detection algorithm based on a random forest algorithm is limited in its current state. The training of the individual trees required a dataset that was comprised of every possible load combination that can be expected. The six unique appliances used in testing resulted in 127 unique combinations of loads. The manual creation of this training set with all those combinations was impractical. It justified the design of supporting hardware to automate this task. While this resulted in an accurate detection algorithm, this method's scalability becomes more complex with the addition of every new unique load. This is a result of all the trees needing to be retrained with every new possible combination in order to maintain the accuracy mentioned in the results.

The physical construction of the device also led to it being more restricted in its functionality than the original design specification, the latter of which called for the device to be able to measure a power draw of 8000 W. While proof of sufficient ADC headroom is provided in the results section and the device is able to measure said power draw, the physical cables and sockets that allow for the measurements to be made is only rated for a maximum power draw of 3680 W.

## **5.3 Design ergonomics**

The target market for this device is the general public - in other words, it will be marketed to the average consumer with no technical expertise. For many consumers energy consumption - and, by extension, the cost of that energy in the form of electricity - is an abstract concept. Consumers often leave appliances on unnecessarily due to them not knowing that doing so leads to substantial energy wastage. When used properly, this device can inform the average consumer about their appliances' energy consumption and the associated cost, leading to them cutting down on energy consumption by turning off unnecessary appliances.

The data is presented to the user through a dashboard accessible as a web page. This dashboard is designed to simplify the relevant information to make it easily understandable. Gauges with clear colour distinctions (green, orange and red) are used to help users interpret raw values such as voltage, current, power and the power factor. The raw numeric values are thus given

meaning and contextualised. While the average consumer may not necessarily have a full understanding of what the values mean, the device will still give them a general indication of how energy can be used more sparingly and in a more cost-effective manner.

Another design choice which was made with the aim of simplifying the information's presentation is that all the respective power draw traces are shown on one graph throughout the day. This graph will thus allow the user to see how much power is consumed, the time of operation, and the duration of the energy consumption. In addition to this graph, a pie-chart showing the numeric proportion of each individual appliance's energy consumption over a period of time is also displayed. The graph and pie-chart are easy to interpret, and will thus inform consumers about which of their appliances are most energy consuming resulting in them using their appliances more efficiently.

## 5.4 Health, safety and environmental impact

### 5.4.1 *Health and Safety*

The device is interfacing with the mains electricity supply thus creating a potentially hazardous situation. As such, mandatory safety precautions had to be taken to prevent user injury or death as a result of accidental interaction. The high voltage and high current supply is therefore isolated through the use of an extension cable and plug - the transformer and plug are connected within isolated plastic housing, thereby significantly reducing any risk of injury as a result of exposure to the mains electricity supply.

The above-mentioned process of isolation thus means that the only way in which a user could possibly still be exposed to dangerous voltage is through the secondary stage of the safety transformer. According to the research presented in [22], a 60 Hz alternating current is potentially lethal at a value of 20 mA. In the same paper, the authors further found that human skin can have varying resistance depending on whether the particular skin has come into contact with water. Resistance ranges from a maximum of  $100\text{ k}\Omega$  when the individual's skin is dry, to a minimum of  $1\text{ k}\Omega$  when the skin is wet. Applying ohms law to the worst case scenario (i.e., where the user's skin offers the lowest possible resistance to the value of  $1\text{ k}\Omega$ , and is then exposed to the most significant voltage potential of 15 V), the largest possible current that can be conducted through the skin is 15 mA. This value is close to a lethal amount. However, it must be noted that this is a worst-case scenario as well as a very unlikely one. The user will be warned that the device is never to be exposed to water or operated in wet conditions, thus making it very unlikely that a situation in which they will be exposed to lethal voltage through the secondary stage of the safety transformer will arise. There is thus no inherent risk of energy or death for a user operating the device in accordance with clear instructions.

### 5.4.2 Environmental impact

As was found by Eskom (South Africa's only public energy provider) in its 2021 integrated report, (and as is indicated by the table in figure 33), 1.08 kg of CO<sub>2</sub> emissions are released per kilowatt hour of energy generated.

	Factor 1 (total energy sold)	Factor 2 (total energy generated)	If electricity consumption is measured in:			
			kWh	MWh	GWh	TWh
Coal use	0.55	0.54	kilogram	ton	thousand tons (kt)	million tons (Mt)
Water use <sup>1</sup>	1.41	1.39	litre	kilolitre	megalitre (Mℓ)	thousand megalitres
Ash produced	161	158	gram	kilogram	ton	thousand tons (kt)
Particulate emissions	0.37	0.37	gram	kilogram	ton	thousand tons (kt)
CO <sub>2</sub> emissions <sup>2</sup>	1.08	1.06	kilogram	ton	thousand tons (kt)	million tons (Mt)
SO <sub>x</sub> emissions <sup>3</sup>	8.36	8.24	gram	kilogram	ton	thousand tons (kt)
NO <sub>x</sub> emissions <sup>3</sup>	4.19	4.13	gram	kilogram	ton	thousand tons (kt)

**Figure 33.**

**Table of approximate resource cost per unit of energy adapted from [23]**

A single tree is estimated to absorb an average of 20 kg of CO<sub>2</sub> emissions per year ([24] and [25]). It is thus arguable that every 18.51 kWh can be counteracted by the planting of a tree. However, the device in itself leads to a reduction in energy consumption as the consumer is made to be more aware of the unnecessary appliances in use in their homes. For example, if a user has accidentally forgotten their pool or underfloor heating on, the device will bring this to their attention leading to them switching it off to save electricity (and money). The global effect is thus a general reduction in the electricity consumption in each home and, consequently, in CO<sub>2</sub> emissions and the burning of fossil fuels. The table above provides further information in this regard, it being argued that the usage of natural resources and harmful emissions will be reduced overall if consumers become more aware about their energy consumption.

### 5.5 Social and legal impact of the design

The designed device will need to be registered in accordance with the designs act of ACT195 of 1993. If the device is marketed to general public as it is intended to be it must comply with the consumer protection act 68 of 2008. The device must also meet the national environment act of 59 of 2008. This will ensure that the device will not become e-waste rusting in negative impacts to the environment. The device must comply with the IEC/AS 62053-21 which prescribes to the standard of energy meters as the device is designed to meet the requirements to classified as class 2 energy meter. The device must comply with the IEC 60038 which prescribes to the standard preferential values for the nominal voltage of electrical supply systems. Compliance with this standard ensures that the deices is able to measure that voltages in all standard locations. This device if eventually marketed will give consumers another tool with which they are able to reduce their electricity. This will allow consumers to make better more informed decisions on how they consume energy ultimately resulting the reduction in their monthly electricity cost.

## 6. Conclusion

---

### 6.1 Summary of the work completed

This report summarises work carried out to design a Non-Intrusive Load Monitoring (NILM) device. The objective was to improve upon known methods to create an identification algorithm better suited for implementation on embedded platforms.

A literature survey was conducted to obtain information about identification features and methods that have been used in this context by other researchers. The software and hardware for a NILM device was then designed from first principles. The design made use of a Beaglebone AI as the embedded platform. C code was developed for the ARM core and the PRU core. The C code was designed to be called from the main Python program. The system was then fully integrated. Supporting hardware and software was then developed to facilitate automated data collection. The random forest algorithm was then trained, and the trees were integrated into the full system. A number of tests were subsequently carried out. An automated test system was set up to automatically switch loads on or off and record the result of the prediction. The main result is shown in section 4.3.1.

### 6.2 Summary of the observations and findings

A load identification algorithm was successfully practically implemented on an embedded platform which then improved. It was found that it was possible to increase the accuracy of identification by creating separate trees for every load. The random forest algorithm was used to identify more complex loads. It was found that, by implementing the detection algorithm using this method, the accuracy achieved was 94.84 %. This is an increase of 8.84 % in relation to the method used by Gouhua *et al* [1].

An important discovery is that, for every unique load, there is an optimal decision tree design that benefits from an optimal number of features and depth. One implication of this is that the design of the random tree forest can be optimised for every load. Care needs to be taken when performing this optimisation as limiting the features or restricting the depth might increase the speed and reduce the complexity, but this will come at the cost of decreased identification accuracy.

### 6.3 Contribution

The use of a Beaglebone AI (Linux based embedded platform) is completely new and thus needed to be mastered before any software could be created. This is not one of the development platforms used in the undergraduate program. Linux is an open source operating system. While students might have previously been exposed to the customisation of the operating system, customisation for implementation on an embedded platform is completely novel.

Specifically, the Beaglebone and Beaglebone AI make use of capes which are also known as device tree overlays. These overlays are required to enable certain functionality of the Beaglebone, such as pin control. While the Beaglebone has an active development community meaning there are many examples of and resources about Beaglebone available, the Beaglebone AI has a much smaller development community and, as such, much more development work is required to achieve the same functionality as its non-AI counterpart.

The Beaglebone also features a multicore design. It has two dedicated ARM cores for general processing and 4-PRUs that are 200 MHz microprocessors. Unlike the ARM cores', the PRU's have low level control over the pins meaning alteration of the state of the pins can be made much faster than the general ARM core. In order to interface the ARM core and the PRU core, the PRU cookbook provided on the Beaglebone website was consulted.

Code was mostly developed by the student, though there was some reliance on existing libraries. The only library functions that were primarily made use of were those that facilitated a speed increase to well-documented mathematical transforms, like the FFT.

## 6.4 Future work

Although the algorithm is able to accurately and reliably predict the state of connected loads, it has limited expansion with regard to the number loads. The algorithm did not make use of any of the new detection features or the other detection algorithms discussed in literature. Future work should be aimed at combining the other features and methods to create a detection algorithm that is not limited in the number of loads that it is able to identify.

The processing platform used (the Beaglebone AI) was not an optimal choice. Selecting a new processing platform that features two independent onboard 16bit-ADCs may prove particularly difficult.

The limited testing done to prove the functionality in this report was as a result of only using one location and power-grid. Future work should include investigating the reliability of the system in different areas - possibly also in different countries.

## 7. References

---

- [1] W. Guohua, Y. Diping, Y. Jiayao, Z. Wenhua, D. Peng, and X. Yiqing, “Research on non-intrusive load monitoring based on random forest algorithm,” in *2020 4th International Conference on Smart Grid and Smart Cities (ICSGSC)*, 2020, Conference Proceedings, pp. 1–5.
- [2] A. F. Jauregui, “How to code decision tree in python from scratch,” Aug 2022. [Online]. Available: <https://anderfernandez.com/en/blog/code-decision-tree-python-from-scratch/>
- [3] T. Anwar, B. Sharma, K. Chakraborty, and H. Sirohia, “Introduction to load forecasting,” *International Journal of Pure and Applied Mathematics*, vol. 119, pp. 1527–1538, 10 2018.
- [4] J. Zheng, C. Xu, Z. Zhang, and X. Li, “Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network,” in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 2017, pp. 1–6.
- [5] Z. Zhihao, L. Jian, Z. Zhen-Yuan, H. Qi, and Q. Hedi, “Non-intrusive load identification methods based on li-norm and hidden markov chain model,” in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2020, Conference Proceedings, pp. 1875–1879.
- [6] G. A. Raiker, S. B. Reddy, L. Umanand, A. Yadav, and M. M. Shaikh, “Approach to non-intrusive load monitoring using factorial hidden markov model,” in *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)*, 2018, Conference Proceedings, pp. 381–386.
- [7] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [8] A. Haq and H.-A. Jacobsen, “Prospects of appliance-level load monitoring in off-the-shelf energy monitors: A technical review,” *Energies*, vol. 11, p. 189, 2018.
- [9] K. Ehrhardt-Martinez, K. Donnelly, and J. Laitner, “Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities,” *American Council for an Energy-Efficient Economy*, 2010.
- [10] A. Saleh, “Signal processing techniques for disaggregation of smart-meter energy data techniques de traitement du signal pour la désagrégation des données énergétiques des compteurs intelligents,” Thesis, Université de Haute Alsace - Mulhouse, 2019, engineering Sciences [physics]/Electric power Computer Science [cs]/Data Structures and Algorithms [cs.DS] Engineering Sciences [physics]/Signal and Image processingTheses. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-03576804>
- [11] B. M. Mulinari, D. P. de Campos, C. H. da Costa, H. C. Ancelmo, A. E. Lazzaretti, E. Oroski, C. R. E. Lima, D. P. B. Renaux, F. Pottker, and R. R. Linhares, “A new set of steady-state and transient features for power signature analysis based on v-i trajectory,” in *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, 2019, pp. 1–6.

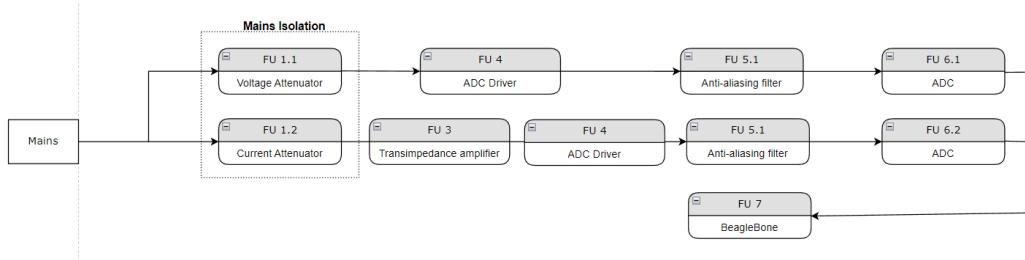
- [12] D. Jung, H. H. Nguyen, and D. K. Y. Yau, “Tracking appliance usage information using harmonic signature sensing,” in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, Conference Proceedings, pp. 459–465.
- [13] A. F. Mubarok, T. Octavira, I. Sudiharto, E. Wahjono, and D. O. Anggriawan, “Identification of harmonic loads using fast fourier transform and radial basis function neural network,” in *2017 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2017, Conference Proceedings, pp. 198–202.
- [14] C. K. Alexander and S. M. N. O., *Fundamentals of Electric Circuits*. McGraw-Hill, 2021.
- [15] *1 Msps 16-Bit Differential Input SAR ADC*, Microchip, 3 2018, rev A.
- [16] S. Franco, *Design with operational amplifiers and Analog Integrated Circuits*. McGraw-Hill Education - Europe, 2014.
- [17] *Current Transducer*, LEM, 5 2020, rev. 3.0.
- [18] *Adjustable Micropower Voltage References*, National Semiconductor (TI), 2 2005.
- [19] S. Arar, “Design examples of fir filters using the window method - technical articles,” Jul 2016. [Online]. Available: <https://www.allaboutcircuits.com/technical-articles/design-examples-of-fir-filters-using-window-method/>
- [20] P. Aznar, “Decision trees: Gini vs entropy,” Dec 2020. [Online]. Available: <https://quantdare.com/decision-trees-gini-vs-entropy/>
- [21] B. P. Lathi and Z. Ding, *Modern digital and Analog Communication Systems*, 2nd ed. Oxford University Press, 2009.
- [22] R. M. Fish and L. A. Geddes, “Conduction of electrical current to and through the human body: a review,” *Eplasty*, vol. 9, p. e44, Oct. 2009.
- [23] Eskom, “Eskom - integrated report,” p. 66, Mar. 2021. [Online]. Available: <https://www.eskom.co.za/wp-content/uploads/2021/08/2021IntegratedReport.pdf>
- [24] P. b. J. M. Stancil, A. s. reader, B. Weaver, Jessica, R. Hoffman, Jacqueline, Selcuk, Aaron, F. Opperman, R. Baradhwaj, and et al., “The power of one tree - the very air we breathe,” Jun 2019. [Online]. Available: <https://www.usda.gov/media/blog/2015/03/17/power-one-tree-very-air-we-breathe>
- [25] EcoTree, “How much co2 does a tree absorb?” [Online]. Available: <https://ecotree.green/en/how-much-co2-does-a-tree-absorb>

## **Part 4. Appendix: technical documentation**

## HARDWARE part of the project

---

### Record 1. System block diagram



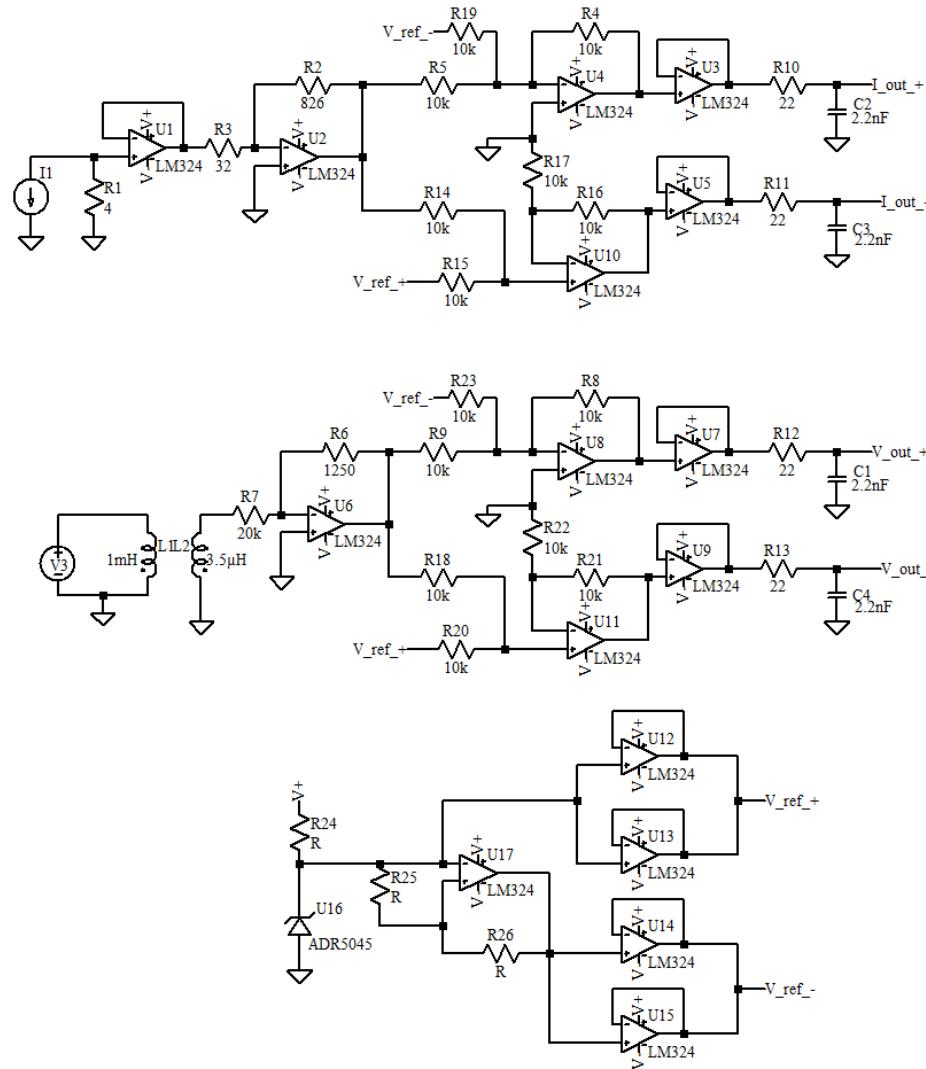
**Figure 34.**  
**System Block Diagram**

### Record 2. Systems level description of the design

The hardware described by the block diagram in figure 34 show the hardware used to measure that voltage and current of common house hold appliances. The first part of the design is the galvanic isolation of the device from the rest of the hardware. This is implemented using transformers. The current signal is then converted to a voltage signal by means of a transimpedance amplifier. The two signals are then amplified or reduced to a voltage range that the ADC is able to measure by means of the ADC Driver. These signal are then parsed through a first order low pass AA filter before being sampled by the ADCs and the result transfer to the Beaglebone.

### Record 3. Complete circuit diagrams and description

The Figure 35 is the full circuit schematic of the analogue front. The equations on how the values are calculated can be found in section 3.2.2 to 3.2.5. These circuits describe the method of safely isolating the user and the device from the dangerous mains voltage that is to be sampled. They are both powered by 2 9 volt batteries.



**Figure 35.**  
**Complete Circuit used for the mains interface**

The above circuits are then implemented on varioboard with all sub-circuits implemented as replaceable modules.

#### Record 4. Hardware acceptance test procedure

1. Once the system has been connected mains along with the two 9V batteries it is ready to be calibrated.
2. Two probes for an oscilloscope are connected to the voltage signal differential amplifier output. The potentiometer used to trim the voltage to a safe level for the ADC.
3. The above item must be repeated for the current measurement.
4. Once both the voltages are in the  $\pm 2.5$  V range of the ADC the hardware is ready to be used.
5. The beaglebone interface lead can now be connected and interfaced.

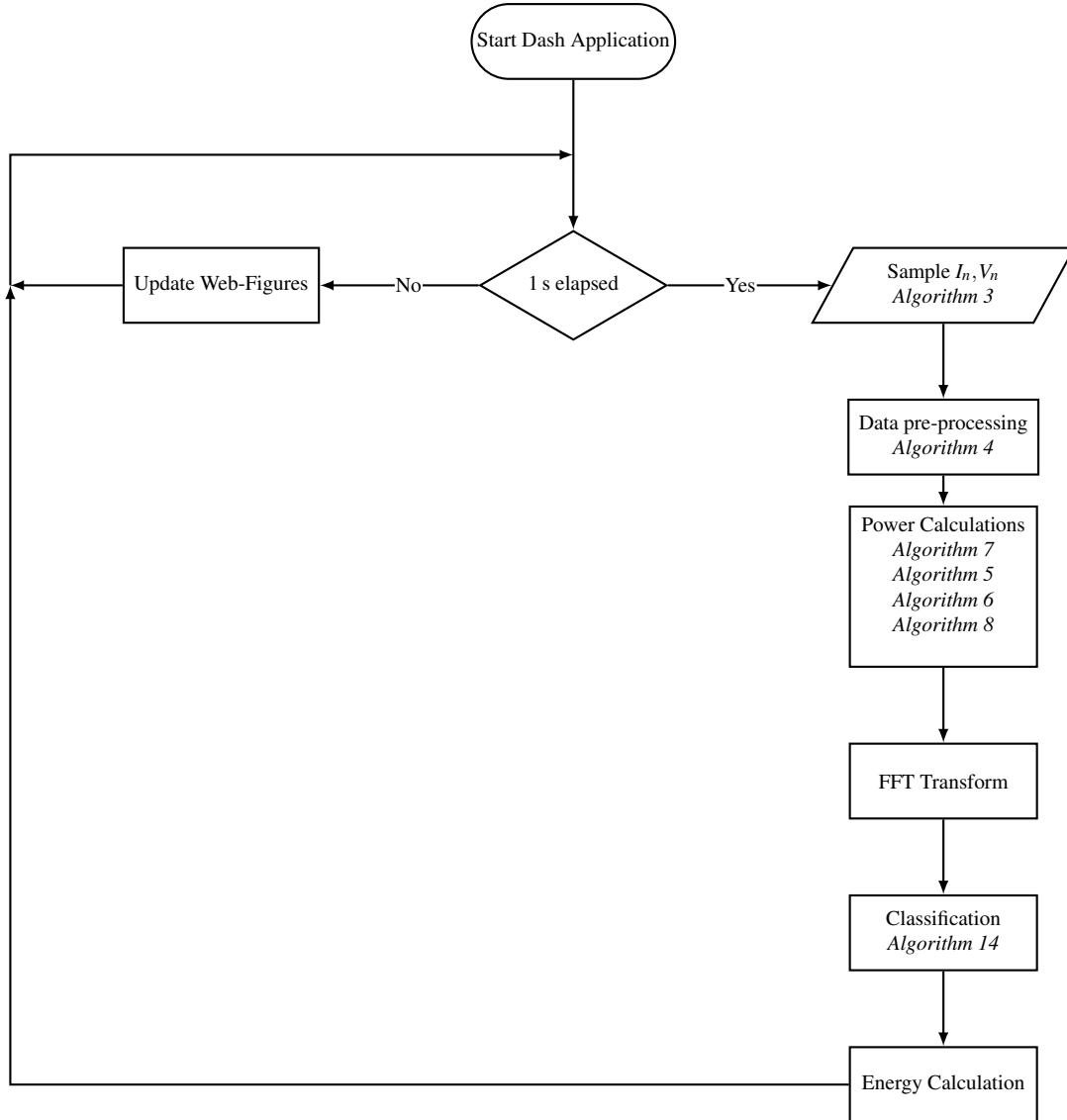
## Record 5. User guide

1. The 9V batteries must be connected first.
2. The system can then be connected to main via the plug.
3. The Beaglebone can then be turned on.
4. Loads can be connected and disconnected at the users discretion.

## SOFTWARE part of the project

---

### Record 6. Software process flow diagrams



### Record 7. Explanation of software modules

The program described by the flow diagram above depicts how all the pieces of individual software interact with one another to create the final product. The first step is performed using a dash call back that occurs every second. This functions similar to an interrupt on a microcontroller. If one second has elapsed all the power values are calculated and the FFT is performed on the current signal. This information is then sent to the identification algorithm. Once the identification has been made the system updates the GUI.

### Record 8. Complete source code

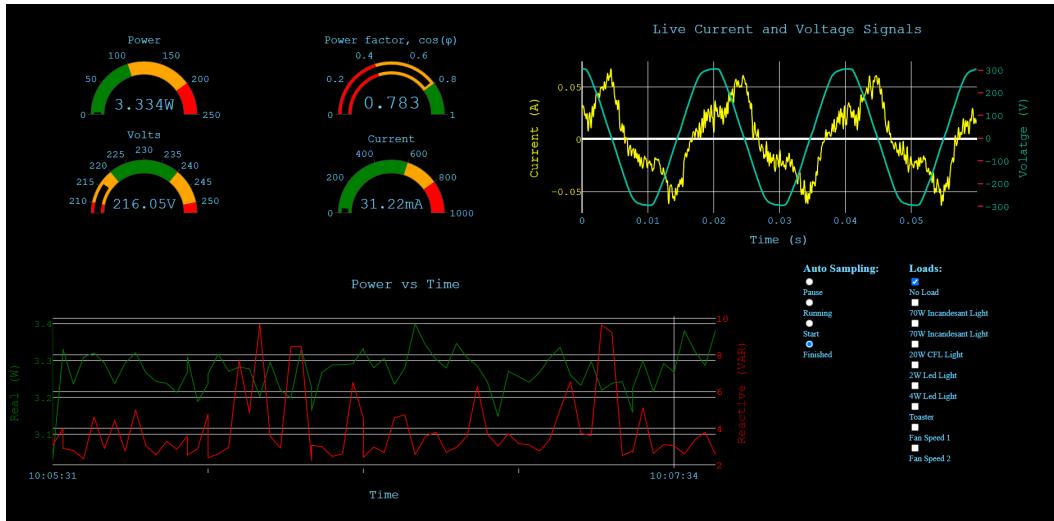
Complete code has been submitted separately on the AMS.

## Record 9. Software acceptance test procedure

After the Beaglebone could9 development IDE is accessed via the online interface. The PRU Program is launched. The system terminal should indicate running after this has been completed successfully. The main GUI.py program is then launched. An IP-Address will appear in the terminal. Assessing this IP address should load the GUI. If the voltage and current waveforms should look like sine wave and mountain if correct. There should also be a reported PF of around 0.7

## Record 10. Software user guide

1. Start the Beaglebone.
2. Open the cloud9 IDE.
3. Launch the PRU sample subroutine.
4. Launch the GUI.py application.
5. Open the website by entering the IP Address into a browser on the same network. The GUI 36 should appear.
6. Confirm the values presented.
7. Leave the web-page open for as long as you want the system to keep running.



**Figure 36.**  
The Dashboard used to control the testing and data collection

## **EXPERIMENTAL DATA**

---

### **Record 11. Experimental data**