Design Report – Team 02

ELETENG209: Analogue & Digital Design

*Design of a Smart Energy Monitor*

Krithik Lekinwala

Hao Lin

Ankush Patel

Ruskin Swedlund

Department of Electrical, Computer, & Software Engineering

# Abstract

This report describes the design, validation and operation of a smart energy monitor capable of measuring the power utilised by an appliance with the use of hardware and firmware implementations.

# Introduction

By establishing our goals, ideas, and opinions from the very beginning of this project, we were able to stay productive, motivated, and effective with our work. Our main objective of undertaking this project was to gain valuable experience of how things operate in the industry and to physically represent our knowledge by creating a practical energy monitor, utilising software, and hardware features.

We are creating a Smart Energy Monitor which can measure the amount of power that an appliance consumes by outputting the voltage, current, and power simultaneously. The Analog part creates, filters, and amplifies our signal so it is better suited for an input to the Digital Processing of the project. The Software implementations take care of re-creating the signal and outputting to a display, smartphones via Bluetooth, and computers using a USB to Serial cable.

# Diagram Description automatically generated

Fig. 1: A conceptual system diagram [1]

# Literature Research

Table I: A Comparison of existing energy meters

|  |  |  |
| --- | --- | --- |
| Parameter | Neurio Smart Monitor | Efergy Pro |
| Operating voltage range | 90 – 130V | 110-300V AC |
| Accuracy | ±1% of reading | 98% |
| Power Consumption | < 2 W | 0.2 W |
| Communication Protocol | Wi-Fi, Bluetooth, wired | Wi-Fi |
| Peak Current | --- | 95 mA |
| Operating Temperature | 0 - 65°C | 0 - 35°C |
| Measurements | Irms, Vrms, W, X, Wh | Energy (Wh) |

# Design Specifications

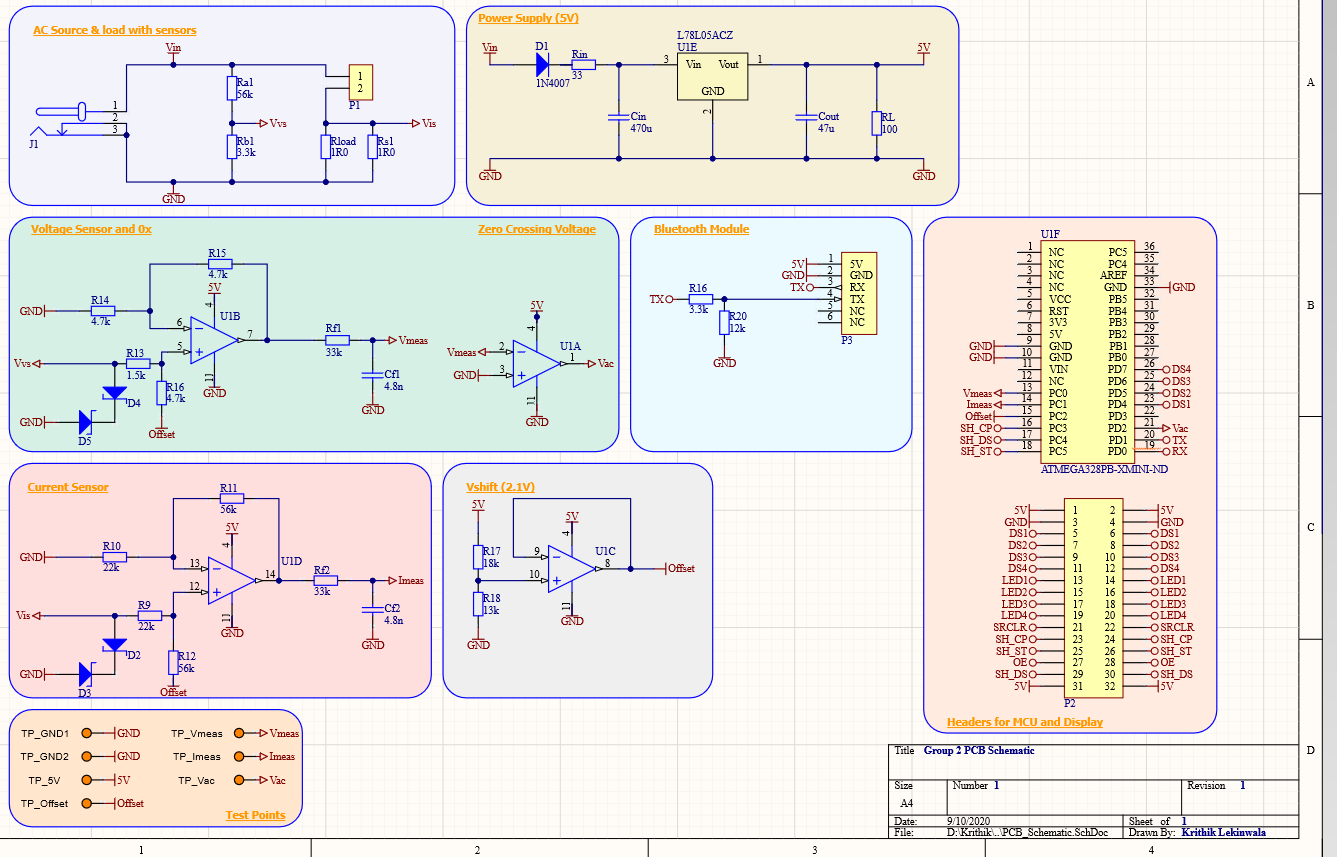
The key differences are in the communication type, operating voltage, and types of measurements. The Neurio monitor measures all the units our monitor can but in addition measures reactive power (X). Efergy Pro only measures Energy. Our Energy monitor is more comprehensive in providing power, voltage and current. Neurio and ours have Bluetooth and hard-wired connectivity, but the Neurio also has Wi-Fi capability.

Table II: Design Specifications

|  |  |
| --- | --- |
| Parameter | Configuration |
| Source Voltage | 14VRMS ± 10% |
| Source Frequency | 50Hz ± 2% |
| Load Range | 2.5VA to 7.5VA |
| Load Power Factor | 0.75 to 0.99 |
| Measurement Accuracy | 5% of full-scale reading |
| ADC Conversion Rate | 1kHz or slower |
| LCD Display Information | Voltage, current, power and energy |
| LCD Display Units | VRMS, Apk, W and Wmin |
| LCD Scroll Rate | 1s |
| UART Baud Rate | 9600 Baud |
| Information Transferred Via UART | Voltage, Current, Power and Energy |
| PCB Size | 200 mm­­­­2 |
| PCB Technology | Double layer with PTH |
| Device Technology | TH or SMT |

# The Analogue Design

## The Schematic

We maximized the gain of our captured voltage and current signals from the differential amplifier input by optimizing the resistor values we choose in order to represent our 5V ADC reference voltage more effectively. The optimisation resulted in an approximate increase of 50%.

## The PCB

We decided to add an additional current limiting resistor to the output of the linear regulator to ensure correct operator of the regulator package.

## Design Validation

Complete Table II (change table numbering as required) to highlight key design parameters and to show that your design functions as predicted by your calculations. For naming convention refer to <https://ee209-2020class.github.io/presentations/DigitalL4/presentation.html#5>. If you had developed hardware and tested it, add a column to include experimental data. Include a short paragraph detailing special features of the design.

Table III: Key design parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Calculated | Simulated |
| Gvs | (3300/56000 + 3300) = 0.056 | |
| Vvs when VAC is 15.4 Vrms | 0.989 V | 986 mV |
| Vvs when VAC is 12.6 Vrms | 0.81 V | 805 mV |
| Gis | 0.56 | |
| Vis when IAC is 0.60 Arms | 0.47 V | 0.446 V |
| Vis when IAC is 0.16 Arms | 0.127 V | 0.125 V |
| Gvo | (4700/4700) = 1 | |
| Vvo when VAC is 15.4 Vrms | 3.09 V | 3.09 V |
| Vvo when VAC is 12.6 Vrms | 2.91 V | 2.91 V |
| Gio | (56000/22000) = 2.55 | |
| Vio when IAC is 0.60 Arms | 3.1 V | 3.08 V |
| Vio when IAC is 0.16 Arms | 2.37 V | 2.37 V |
| Gvf | 1 | |
| Vvf when VAC is 15.4 Vrms | 3.32 V | 3.29 V |
| Vvf when VAC is 12.6 Vrms | 3.1 V | 3.08 V |
| Gif | 1 | |
| Vif when IAC is 0.60 Arms | 3.31 V | 3.29 V |
| Vif when IAC is 0.16 Arms | 2.42 V | 2.43 V |
| ΔVin of 5V regulator | 2.13 V | 1.56 V |
| ΔV5V of 5V regulator | 0 V | 0 V |

# The Embedded Software Design

## Flowchart

Add a detailed flowchart to explain the functionality of your embedded software. Use appropriate symbols. The flowchart should clearly indicate how your embedded software work so someone else can use it to modify your software without having to talk to your team. You may divide the flowchart into a few flowcharts if required. Also, if needed include a short paragraph and/or pseudo code to provide details of any special features of the design.

## Peripheral Configurations

Add tables summarizing how each peripheral used is configured. As an example, Table III (change table numbering as required) show the UART configuration.

Table IV: UART configuration

|  |  |
| --- | --- |
| Parameter | Configuration |
| Number of data bits | 8 |
| Number of stop bits | 1 |
| Baud rate | 9600 |
| Parity mode | None |
| Transmission mode | Simplex (transmit) |
| Transmit mode | Polling (UDRE0) |
| Transmit voltage | 5 V |

Table V: Peripheral 2

|  |  |
| --- | --- |
| Parameter | Configuration |
| Number of data bits | 8 |
| Number of stop bits | 1 |
| Baud rate | 9600 |
| Parity mode | None |
| Transmission mode | Simplex (transmit) |
| Transmit mode | Polling (UDRE0) |
| Transmit voltage | 5 V |

Table VI: Peripheral 3

|  |  |
| --- | --- |
| Parameter | Configuration |
| Number of data bits | 8 |
| Number of stop bits | 1 |
| Baud rate | 9600 |
| Parity mode | None |
| Transmission mode | Simplex (transmit) |
| Transmit mode | Polling (UDRE0) |
| Transmit voltage | 5 V |

Table VII: Peripheral 4

|  |  |
| --- | --- |
| Parameter | Configuration |
| Number of data bits | 8 |
| Number of stop bits | 1 |
| Baud rate | 9600 |
| Parity mode | None |
| Transmission mode | Simplex (transmit) |
| Transmit mode | Polling (UDRE0) |
| Transmit voltage | 5 V |

## Design Validation

Add information to validate the functionality of your embedded software modules. Show that they work as intended. For example, you can include the following information gathered from Proteus (also include experimental results if you have developed hardware),

* Oscilloscope capture of UART transmitting a few bytes of data to validate correct UART settings
* Screen capture to show messages printed on terminal
* Validate ADC by applying a DC voltage and graphing conversion results
* Validate timer(s) using oscilloscope capture of for example a pin toggle
* Validate ISR(s) using oscilloscope capture of for example a pin toggle
* Validate 7-segmet display operation using screen capture of the display and oscilloscope capture of data in to and out of the shift-register

Include text as needed to explain your results.

# Performance of the Energy Monitor

## Design Validation

Add 2 plots of ADC data points obtained for each voltage and current signals from Proteus to show your program captures data as intended. In these plots, indicate sketches of the voltage and current waveforms you were measuring as obtained from LTspice/Proteus. If you had developed hardware and tested it, include your ADC data points obtained experimentally for each voltage and current signals in 2 separate plots and compare these against voltage and current waveforms obtained from an oscilloscope. You can use Excel or MATLAB for plotting.

## The Accuracy

Add 3 plots to show accuracy of your voltage, current and power measurements. First plot should have x-axis as ideal values of voltage measured and y-axis as error in voltage measurement. Second plot should have x-axis as ideal values of current measured and y-axis as error in voltage measurement. Third plot should have x-axis as ideal values of power measured and y-axis as error in voltage measurement. Each plot should have at least 5 data points. You can use Excel or MATLAB for plotting.

# Conclusions

# In a single short paragraph tell the conclusions of this report.

# References

1. D. J. Thrimawithana, Class Lecture, Topic: "Analogue & Embedded Software Design: An Introduction to the Course" ELECTENG 209, Department of Electrical, Computer, and Software Engineering, The University of Auckland, Auckland, October 2020.

**Dep. of Electrical, Computer, & Software Engineering**

The University of Auckland   
20, Symonds Street

Auckland, New Zealand

**T** +64 9 373 7599

**W** auckland.ac.nz