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/Display Method | Wi-Fi, Bluetooth, wired | Wi-Fi |
| Peak Current | --- | 95 mA |
| Operating Temperature | 0 - 65°C | 0 - 35°C |
| Measurements | Irms, Vrms, W, X, Wh | W, Irms, Vrms, 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 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

In the UART implementation, we implemented a custom print function to reduce clutter, total memory used, and for easier use. This function can transmit directly to the terminal by using

*print(“xyz: %d %f”, current, voltage);*

Additionally, we also used cubic interpolation and Simpsons Rule to integrate points for power and RMS voltage. Cubic interpolation is used to increase the accuracy of detecting peaks for RMS voltage and peak current values.



## Peripheral Configurations

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: ADC configuration

|  |  |
| --- | --- |
| Parameter | Configuration |
| Prescaler | 4 (sim), 128 (hardware) |
| Timer0 ADC mode | CTC A auto trigger |
| ADC interrupt time | 1ms |
| Voltage reading | ADC channel 0 |
| Current reading | ADC channel 1 |
| OCR0A | 99 (sim), 124 (hardware) |
|  |  |

Table VI: Display Configuration

|  |  |
| --- | --- |
| Parameter | Configuration |
| 7-Segment pin operation | Common Cathode |
| Units display | Ds4 |
| Clock Pulse bit | SH\_CP |
| Timer2 interrupt time | 1ms |
| Timer2 OCR2A | 99 (sim), 124 (hardware) |
| Timer2 display update interval | 10ms |
| 4 Display operation | Shift Register |

Table VII: I/O pins

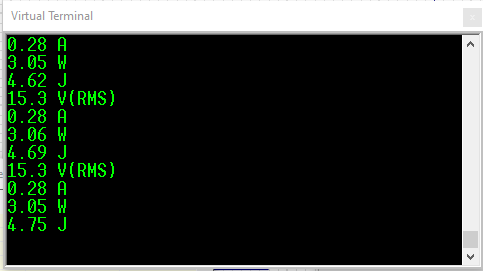
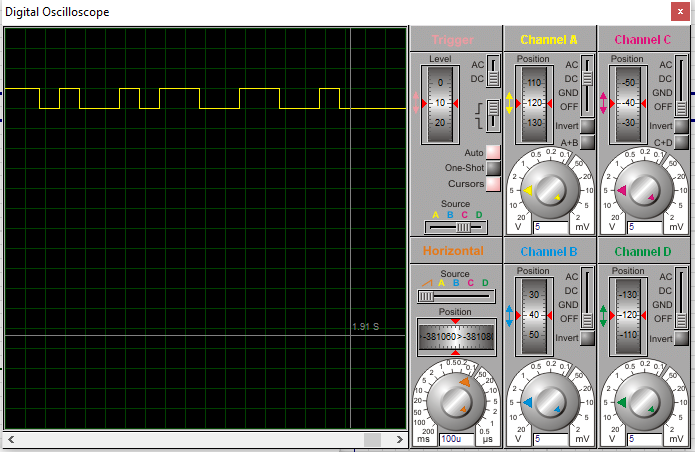
|  |  |
| --- | --- |
| Parameter | Configuration |
| 7 Seg Display Output | Ds1, Ds2, Ds3, Ds4 (PD4, PD5, PD6, PD7) |
| Reading Shift Register | SH\_DS (PC4) |
| Toggle shift register | SH\_ST (PC5) |
| LED toggle for testing operation | LED (PB5) |
|  |  |
|  |  |
|  |  |

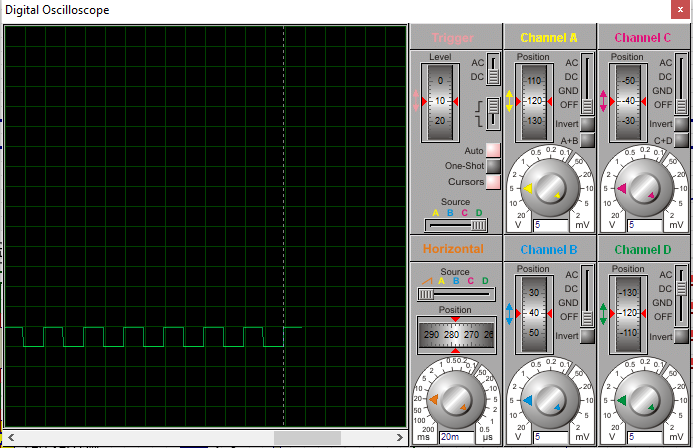
## Design Validation

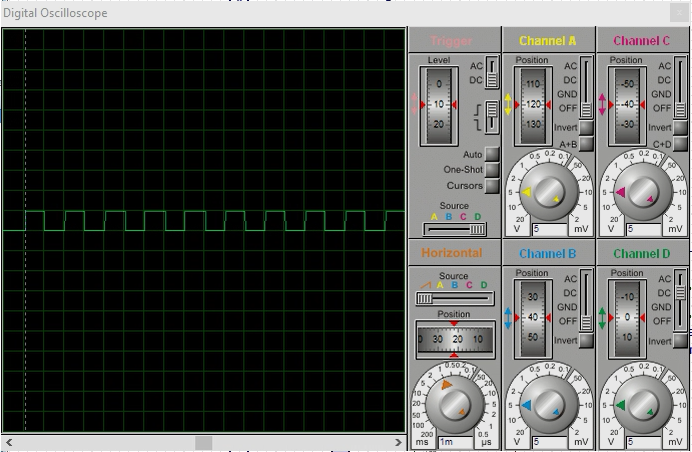
Add information to validate the functionality of your embedded software modules. Show that they work as intended. For example, you can include the following information gathered from Proteus (also include experimental results if you have developed hardware),

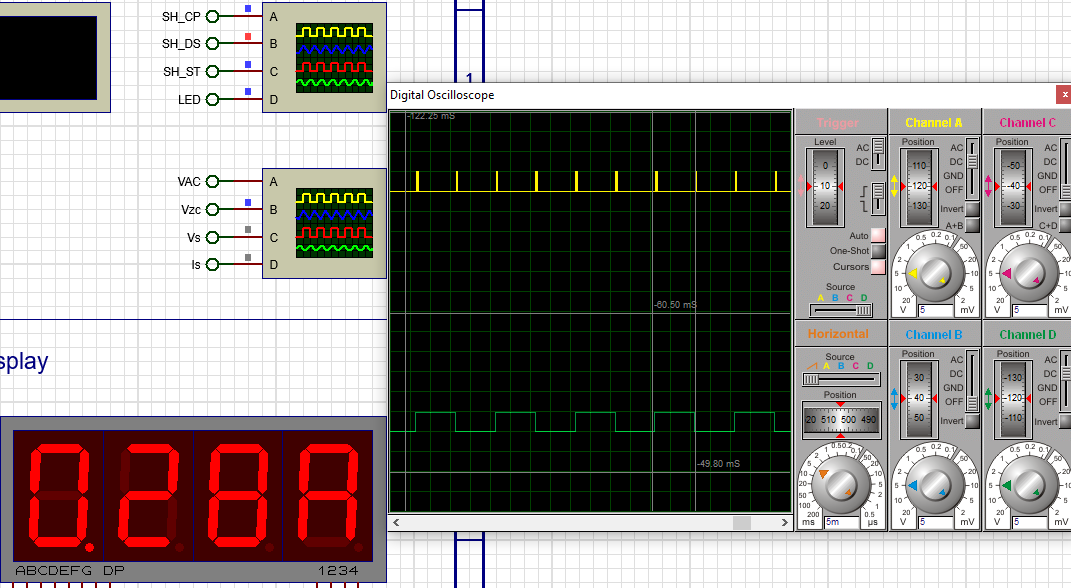
* Oscilloscope capture of UART transmitting a few bytes of data to validate correct UART settings
* Screen capture to show messages printed on terminal
* Validate ADC by applying a DC voltage and graphing conversion results
* Validate timer(s) using oscilloscope capture of for example a pin toggle
* 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.

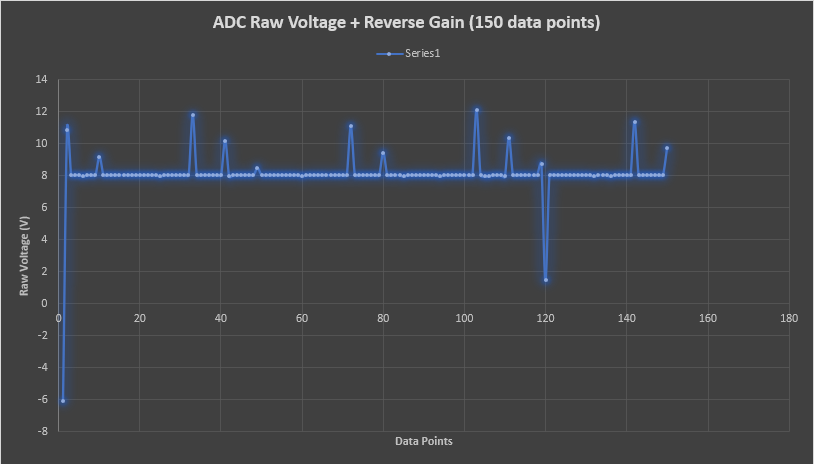
Oscilloscope capture of UART transmitting onto oscilloscope and terminal. The yellow waveform represents the voltage, and the blue waveform represents the zero crossing. Every time the 2nd zero crossing is triggered, the UART transmits the voltage at that point with a slight delay. Every first zero crossing, we sample and calculate, then transmit on the 2nd zero crossing. You can see the TX pin pulsing every 100uS when there is a transmission.

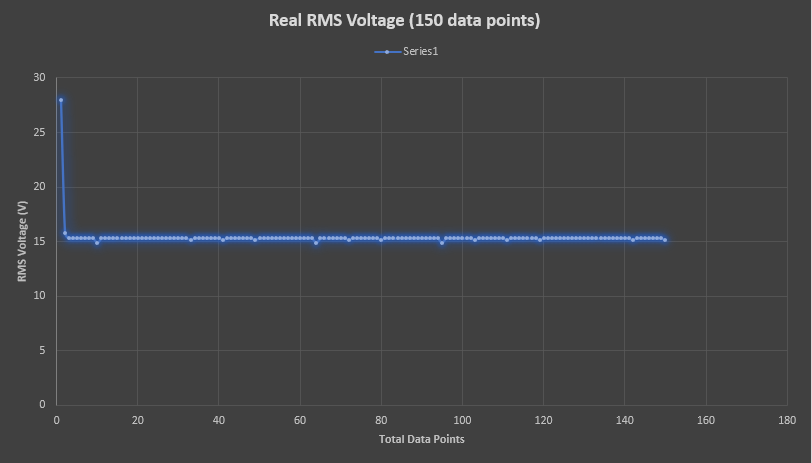
The zero crossing ISR (INT0) triggers every 20ms and the LED triggers when output compare match A is successful. From this screen shot, you can see that the LED turns on and off every 20ms.

The ADC sampling ISR (ADC\_vect) samples the ADC channel and triggers every 1ms.

Timer2 triggers approximately every 10.8ms which updates the display accordingly. We chose to use a faster update to stop flickering effects. SH\_CP triggers for every bit shift (yellow waveform).

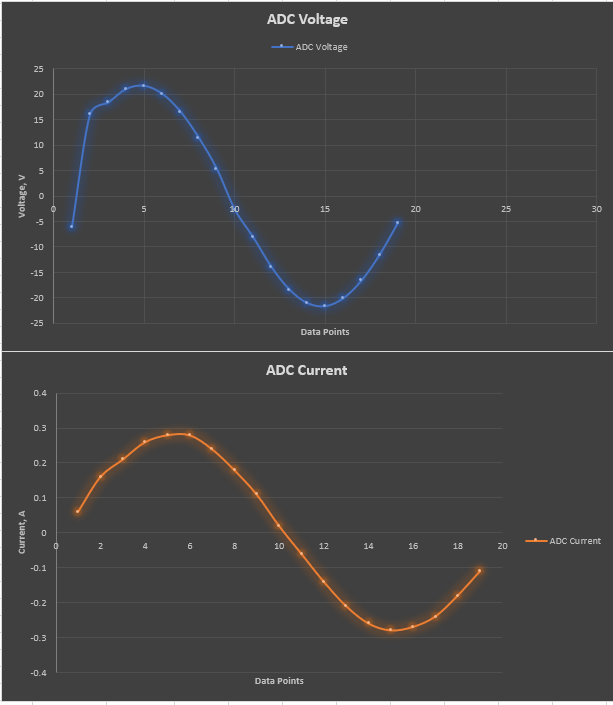
When applying a 15.4Vrms DC voltage, the ADC voltage conversion appears mostly constant, with a few imperfections. It fluctuates mainly by a few mV. After numerical integration and cubic interpolation, our results are much more consistant.

**BEFORE**

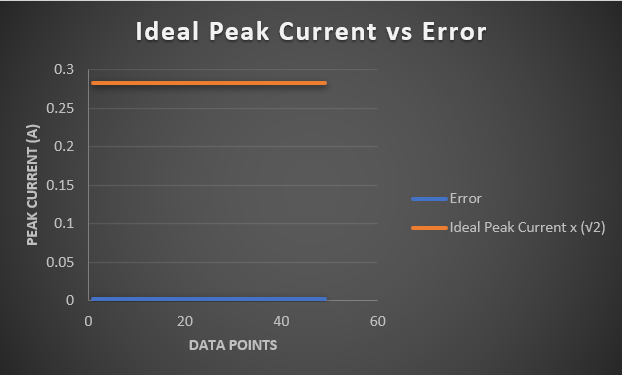
**AFTER**

# Performance of the Energy Monitor

## Design Validation



## The Accuracy

The current measured from Proteus was not the peak current, hence it was multiplied by √2 so it better compares with our calculated values.

# Conclusions

# In conclusion to our Energy Monitor Project, our report discusses how our Energy Monitor operates in terms of firmware and hardware, accuracy, workability, and comparisons to commercial grade products. We were able to achieve extremely accurate calculations with the use of software techniques. We hope that this report has given the reader some insight on our Energy Monitor and how it can be used as a real world example.

# 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