Smart Energy Monitoring Network

Design Document

Team Number: May1725

Team Email: may1725@iastate.edu

Adviser: Nathan Neihart

Team Members:

Joseph Freeland (Co-Lead)

Milan Patel (Co-Lead)

Adam Cha (Communications Lead)

Adam Dau (Webmaster)

James Tran (Key Concept Holder)

Wei LinLin (Key Concept Holder)

Contents

1 Introduction

1.1 Project statement

We are trying to make a wireless power sensor that can monitor the power usage of different electronic devices that appear around the house. The sensor would then report that power usage back to the user via a user friendly web application. Our goal is to have multiple power sensors that can simultaneously monitor the power of different devices, and be monitored on the user interface all at once. Also there will be a central hub acting as the middleman between the sensors and web application. All this will be connected via the user's wifi network.

1.2 Purpose

Our device's purpose will mostly be decided by how the user would like to implement it. If they want to measure how much energy it takes to run a fan all day, as opposed to turning the A/C up, they could do that. Our power sensors would serve as a way to compare the two to decide which is more cost effective. Also if they would like to see how a device, let's say a toaster, compares to other toasters as far as energy consumption then they could do that. Maybe they would like to see how much energy leaving a night light on all night for their kids takes up, or if their is power being wasted by leaving their laptop charging all night. These are all ways that our power sensors can be used. Generally we are giving people an opportunity to have more knowledge and control of how they use electrical devices around the house.

1.3 Goals

First our main goal is to be able to build a working power sensor unit. That means that we have a sensor that can measure the power being used by a certain device. Our sensor will be plugged into an outlet along with the device it will be monitoring. We will need it to be able to take measurements periodically, and go into a low power mode between measurements in order to save energy.

Our next goal would be to have working communication between our database, power monitoring sensor, and web application. We plan on using a wifi connection to transfer data between these three sources. This also means that the users will need to have a working wifi network in order for our design to work. We will implement a way to connect to the wifi in the web application.

Next we would to have multiple devices plugged in and transmitting data at the same time with the user interface being able to show all the devices in use at once so the user can look at and compare devices at the same time. Also we would like the necessary equations to put the data in the format we want to be at least partly implemented in the power sensors circuitry.

Also we would like to have our web application compare the normal power usage for a certain device with what the user's device is using. We will either do this by comparing data from other people on our network that are also monitoring certain devices of the same type, or by determining a fixed number that would best represent the average for that device based on our own research.

2 Deliverables

There are three main deliverables that we must have to complete our objectives. First is the power measuring circuit that will plug in with the devices to measure the power they are using. This needs to be small and out of the way, and use up very little power itself. Otherwise the whole point of monitoring the system in order to to cut power usage is destroyed. Also we would like to create multiple power monitoring sensors so the user can have many different devices plugged in and being monitored at the same time.

Next we will need a central database that will store all the measurements that are taken from the power measuring circuit. We have created a database in SQLite that we will be able to use for this purpose. We have also had success in storing and retrieving data from the database, and our next step will be to have data from a circuit be received and stored by the database.

Lastly we need the User interface. Again this will be in the form of a web application, and will be able to show and compare multiple devices that the user has plugged into our sensors. From this user interface they will be able to name all the devices they are monitoring for easier identification, and see what different devices are consuming compared to the normal power consumption for that type of device.

3 Design

3.1 System Specifications

There are several commercially-available energy monitoring software systems that we looked at before starting design. We wanted to make sure that our software and hardware could match or exceed the specifications of the systems on the consumer market.

From a hardware perspective, most systems use high side current sensing measurement. We developed a circuit that measures current, directly, on the low side of the load. We do this to avoid the impractical cost that comes with a high common mode differential amplifier needed for high side measurement. Although we use a different method, our components are the same. We are making use of a current sensing resistor and differential amplifier. We then follow that with a series combination of auto-ranging, digitally-programmable amplifiers. This method allows for excellent sensitivity, but the main trade-off comes in the form of power/heat dissipation and excess components which increase the cost.

On the software side, most systems use a web application or a mobile application to present the user with their energy usage. After considering the advantages of several different approaches, we decided to focus our energy into a web-based application. This removes any OS-dependency, as a web application can be viewed on any device with a browser, opening our application up to many more devices than just Android or iOS. In addition, our design requires a central network server, and the app can easily be hosted on that central hub. An example of a system already created is the TED system [1](Refer to Appendix 1 for a picture of the system).

3.1.1 Non-functional

3.1.1.1 Web Application System Requirements

1. The web application shall allow the user to change the period of energy data collection
2. The web application shall show the user energy graphs over a selectable time range
3. The web application shall show a list of all connected monitoring stations
4. The web application shall allow the user to give each monitoring station a user-friendly name
5. The web application shall allow the user to turn off the AC power to individual energy monitoring stations
6. The web application shall allow the user to calculate the cost of any individual device
7. The web application shall retrieve it's data from a central database
8. The web application shall fulfill all these requirements in the Chrome browser

3.1.1.2 Central Data Processing Requirements

1. The data processor shall receive data from all connected monitoring stations
2. The data processor shall convert the data from X units into Y units
3. The data processor shall store each data point, along with a timestamp, into the central database
4. The data processor shall be able to receive simultaneous transmissions from at least 10 monitoring stations

3.1.1.3 Hardware System Requirements

1. The hardware shall not be a source of significant power drain (power consumption less than 5W).
2. The differential circuitry must be able to maintain a maximum bidirectional current with an RMS-amplitude of 15A.
3. The hardware will have a current floor of 100 mA RMS.
4. The hardware shall be sensitive to changes in current and voltage measurement to the value of at least 100mV/A.
5. The hardware shall not make the load operate in conditions that are harmful to it and/or affect performance of the circuit.
6. The hardware shall have a user-controlled switch within it.

7. The hardware shall be able to provide an open circuit in the event of operation outside absolute maximum ratings (max current is rated at 15A).
8. The hardware shall provide a output with minimal frequency modulation.
9. The hardware shall operate in temperate range from -25 to 80 degree Celsius.

3.1.2 Non Functional Requirements

3.1.2.1 Hardware Non-functional Requirements

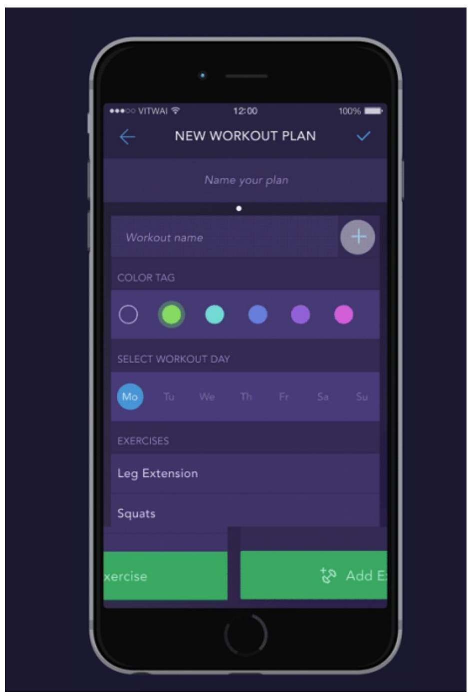
The hardware should be able to fit in a package that is non-intrusive to other devices on the electrical outlet. Additionally, we are aiming to be NEMA 5 compatible; as are all modern electrical sockets. This device should also not produce any obtrusive audible noise when it is in on, or standby mode. The overall goal is to produce a device that is negligibly intrusive to the user.

At this point of the design stage, we are confident that we have enough space to integrate all the components behind the wall outlet. This might be a breakthrough in terms of design because there is no wall outlet with power monitoring capability in the market. Most of commercial power monitor devices are extruded. Our goal for this particular design is to hide all the hardware components behind the wall outlet.

3.1.2.2 Web App Non-Functional Requirements

The web app should allow the user to view the power consumption of all devices in graph or text form. The web app should be modern and well-designed, with a sensible UI and easy to use controls. Commercially-available designs set a high standard for usability and our application should be no different. Below are some designs that we can elaborate from:

Example Design 1

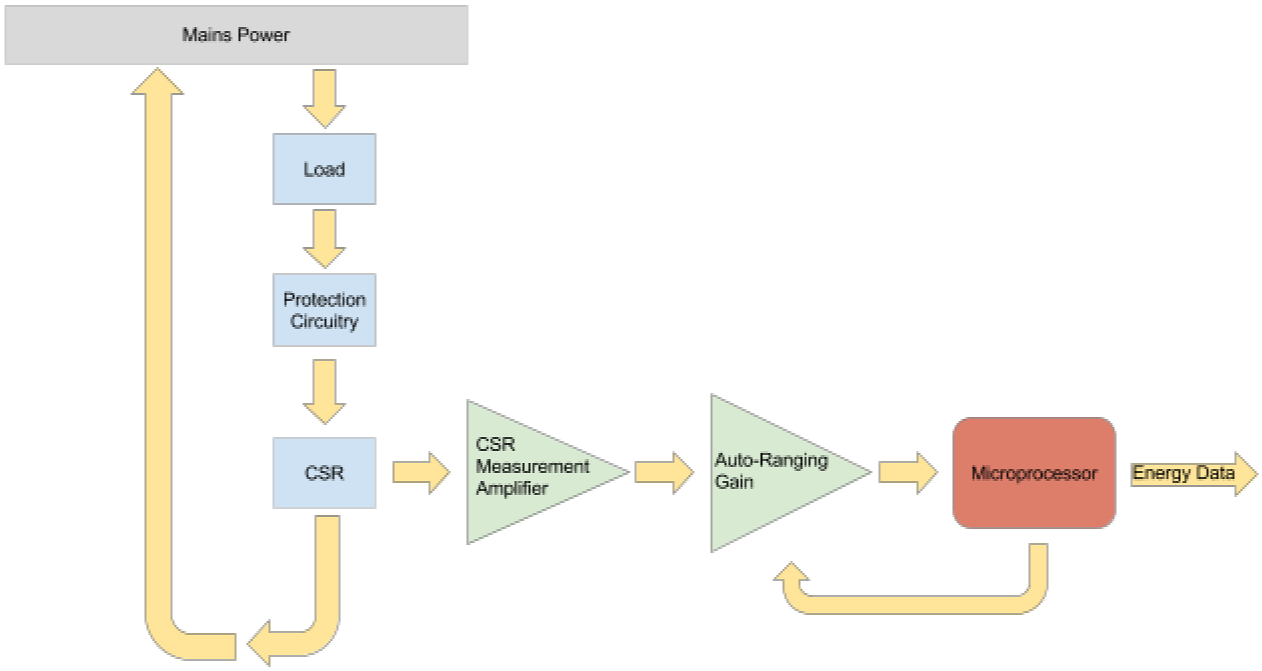


Example Design 2

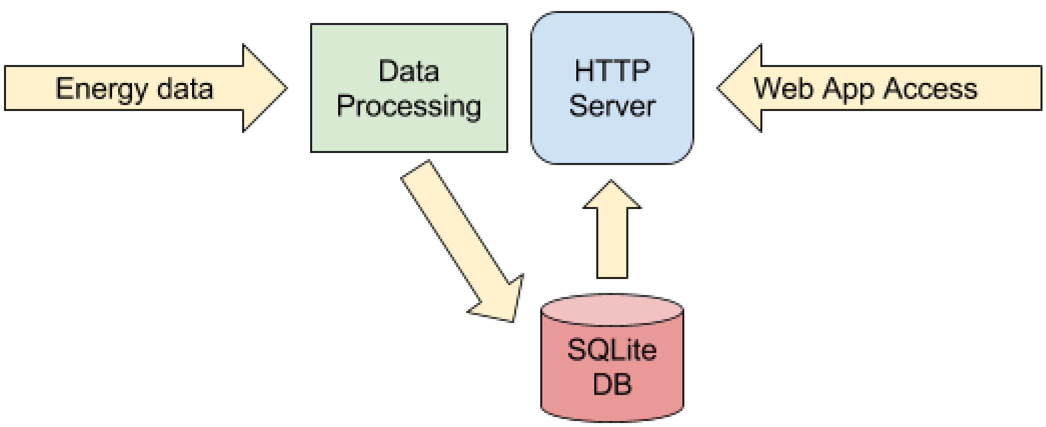
3.2 Proposed Design/Method

3.2.1 Proposed System Block diagram

Hardware Block Diagram



Central Server (Network) Block Diagram



3.3 Assessment/Design Analysis of Proposed methods

3.3.1 Hardware Assessment/Design Analysis of Proposed methods

Current sensing circuit:

The hardware portion of this project refers to the safe and accurate measurement of current & voltage. We have fuses and surge protection equipment on board to protect the user and their load in the case of a main breaker failure. We also provide a user controlled switch in the way of a transistor, if the user chooses to switch it off, remotely.

When it comes to power measurement, we chose to measure on the low side of the load. We do this as to not disturb the load (i.e. avoid brownouts). Originally, we had planned on using a Hall-Effect sensor as our means of power measurement. However, as we progress through the process, we began leaning toward the shunt current sensing because the Hall-Effect sensor was producing too much noise which in turn led to inaccurate measurements at low current levels. Regardless of high or low side, the load produces a certain current draw from the mains line, we then place a CSR in series with the load to measure the voltage produced across the CSR with respect to common. Achieving this is possible through Ohm's Law. The remaining voltage that drops across our Current Sensing Resistor(CSR) is where we begin our measurements. Here, we take the voltage of the CSR and amplify it using a fixed-gain differential amplifier. We then take this amplified signal and run it through two more inverting amplifiers. The reason we have two is for more options for gain settings; which in turn allows us to be more precise by allowing us more auto-ranging levels. Another reason we have two is so that we don't have an inverted output, which makes for more convenient calculations.

We mainly chose to use low-side measurement not just for brownout prevention, but rather due to product availability and cost. As we alluded to earlier, measuring on the high side entails a high common mode amplifier and, when we throw in all of our other specifications; the field of usable devices becomes non-existent. Therefore, we are forced to move forward measuring on the low side. However, this is not a huge issue. Usually, when one measures on the low side of the load, it will will cause some instability since they are lifting that terminal off of the common node. But the really worry comes from not being able to sense short circuits.

Normally, we would send a signal from the microprocessor to our digitally controlled switch to open if an excessive current was detected. But, we came to the conclusion we wouldn't be able to break the circuit before permanent damage was done to the load; therefore, being on the high or low side had no effect. Additionally, since we are dealing with household power, most circuits are self-regulating, in the fact that they will break the current path internally if a current greater than the max rated continuous current. Most of the single breakers are rated for 15A RMS, hence that is the max continuous current our device can handle. Nonetheless, some breakers are rated for 20A RMS+. To protect our device in those cases, we included our own internal fuse that takes action when a steady current greater than 15A RMS is detected. Now that this issue is taken car of, the measurement differences between high & low side are trivial.

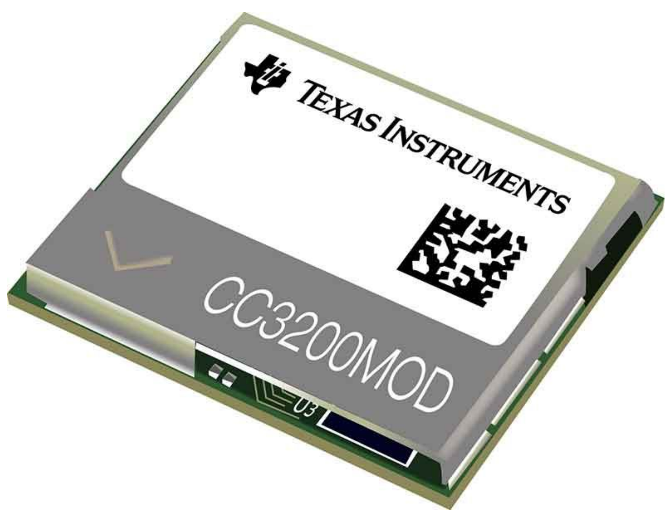
The remaining universal problem in our circuit is start-up transients. Since all loads have surge protection components on board, we just have to worry about protecting our own device. This becomes also trivial by adding a few components like capacitors, diodes, etc; just as our loads do..

Moving to the output of the first stage amplifier, we have a current dependent voltage waveform. We have built the differential amplifier so that a max current will correspond to a voltage amplitude equal to the max input voltage of our microprocessor's ADC, 1.4V. However, for currents less than the max rated continuous current, we implement the microprocessor controlled variable gain amplifiers.

When the microprocessor ADC voltage is less than 65% percent of 1.4V, the microprocessor will implement an auto-range algorithm to find the best gain to measure the waveform. This information is then feedbacked into our power calculations for accurate data logging. Going forward from here, all tasks are executed by the network.

Microcontroller Unit:

When designing a low-power consumption device, the MSP microcontroller from Texas Instruments is more than adequate for the job. However, we didn't just need processing power; we needed a communication apparatus to get the data to the central hub. Therefore, a low-power MCU with a wifi shield was needed. The CC3200MOD was chosen for meeting all of our design criteria and then some; while occupying a small footprint.



With the ARM Cortex-M4, we can easily handle all the computation, interrupts from the sensor, power management, and circuit control. Additionally, the built-in Wifi shield provides reliable connectivity and easy setup for the programmer.

3.3.2 Network Assessment of Proposed methods

We define the 'network' portion of the project to refer to the data processing and receiving code, HTTP server, and database system. As with all web-based applications, there are many available approaches that we could choose.

The data processing code is responsible for receiving (over WiFi) all the data packets from the monitoring stations. The data is sent over a network, so we essentially have two options for data transmission, UDP and TCP. TCP is a connection-oriented protocol that offers several error-handling utilities. UDP is connectionless and offers no guarantee for delivery, but it is the one we will be using. It is well-suited for data streaming and it is simpler and easier to implement.

The HTTP server portion of the system also includes the web application it is hosting. This part of the project is fairly straightforward, as there are a few industry-standard servers available, like Apache. The specific server we choose isn't important, so we will most likely stick with what is installed on our machine. We have chosen to host the server on a Raspberry Pi, a device that will most likely match the hardware in a commercially-available system, in both technical and physical specifications.

For the database system, we are presented with many different options. Eventually, we settled on using SQLite. SQLite is a single-file database that is accessed like a file, directly from the disk. This eliminates the need for a separate database server/connection, making our overall system simpler. It is also a single file, making it easy to backup and transfer between machines. To create a database, we store three variables in a file or separate files: timestamp, the id of the station that recorded it, and the raw value. We use integer type to store timestamp and id, real type to store raw value. The limit range of these values are 12 bits.

3.4 Testing & Validation

3.4.1 Network

To verify the network portion of the project, we will have several different testing strategies, one for each portion of bigger component.

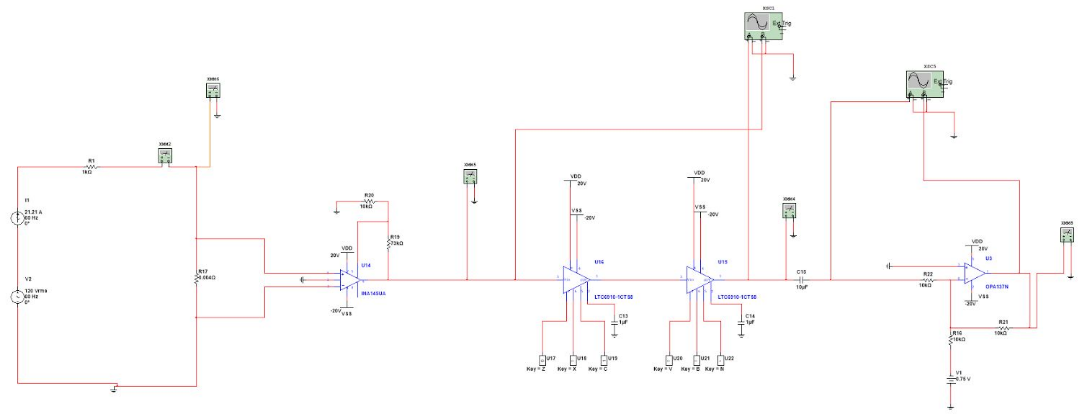
To test the data processing, we will send dummy packets to the central hub via some test device, at the specified UDP address. This will allow us to make sure the packets are being handled correctly, in addition to testing how many simultaneous packets can be received before overloading occurs.

The database and HTTP server portions aren't subject to the same testing, as they will basically either work or fail. Software testing of our actual web application will not be covered here as we don't have a detailed software plan yet. We will most likely utilize unit testing to test each component of the web app.

3.4.2 HARDWARE

Power monitoring component

Simulated Validation: Before building the prototype, we simulated the whole system with the Multisim software. With its comprehensive library of components, we were able to use Multisim to perform interactive simulation our proposed design.



Using the simulation, we had to determine the functionality and accuracy of our power measuring device and how it will rely on the performance of the current sensing component. Starting off, it is safe to assume that power from the utility company will maintain 120 Vrms voltage at 60 Hz, since they have very strict regulations on that. The validating process then becomes: 1) functionality, where we will be using commercial ACS712 sensor to compare the results between it and the CSR method; and 2) accuracy, where will be using Agilent 34401a multimeter to compare the accuracy of the CSR method.

Results:

These results are obtained from Multisim simulations and the Labview waveform viewer.

Load 1: 1200 ohm resistor - 1 mA(RMS)

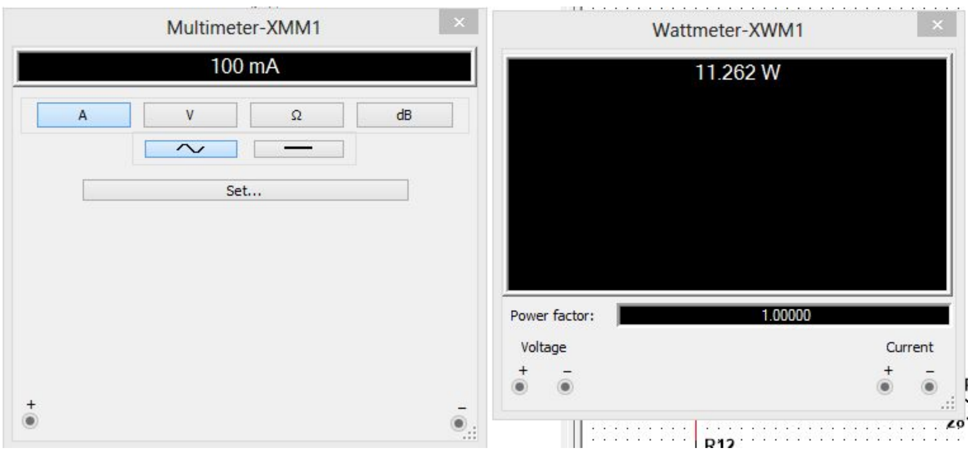


Figure : Actual current and power consumption of the load

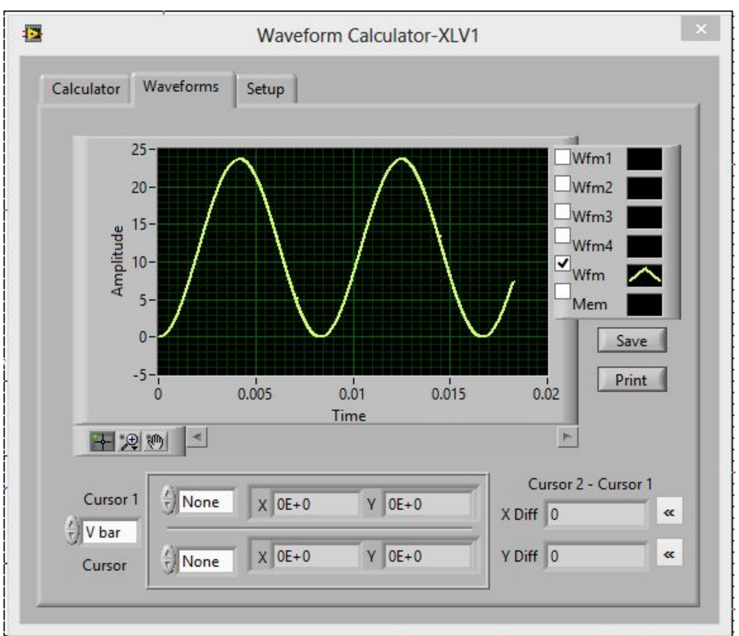


Figure : Instantaneous power (Average of 11 Watts)

Load 2: 8 ohms resistor - 15A(RMS)

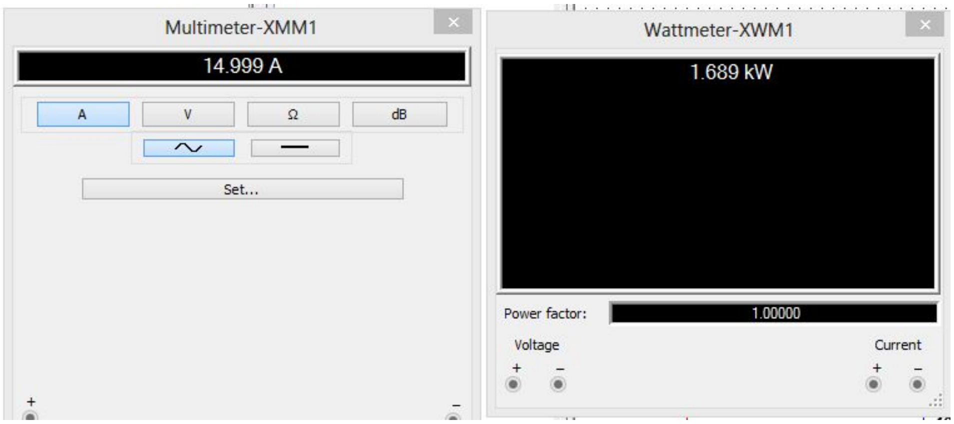


Figure 5.1-3: Actual current power consumption of the load

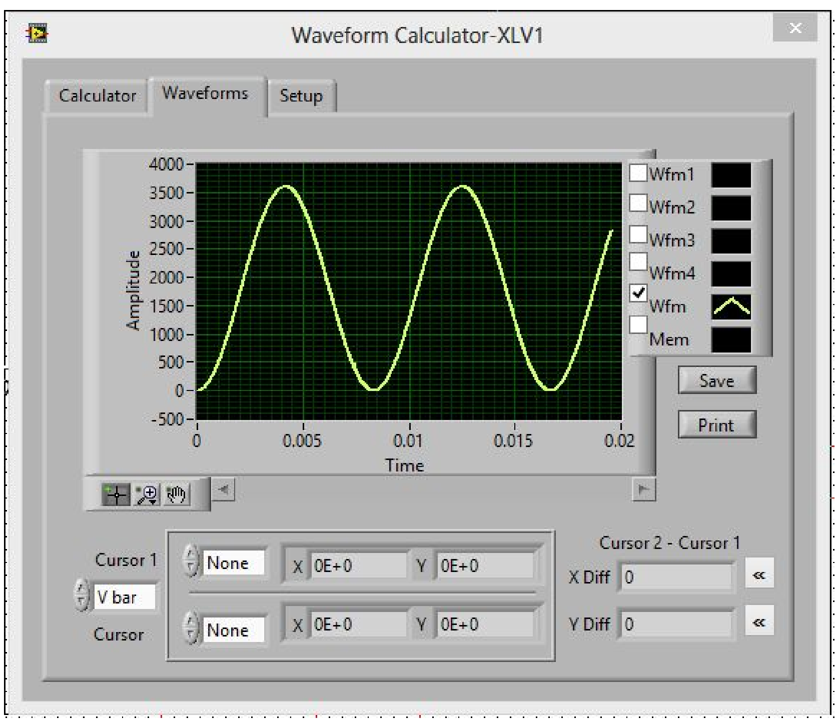
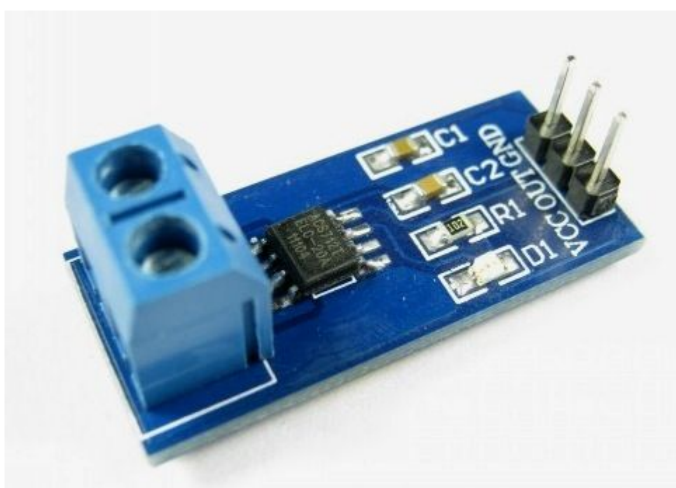


Figure 5.1-4: Instantaneous power (Average of 1.7 kW)

Comments:

- Using shunt resistors, we were able to measure current as low as 100 mA. This is something that we couldn't achieve using the ACS712. Additionally, the noise level at the output is significantly reduced. We are confident that our actual circuit will perform equivalently to the simulation because Multisim has very accurate amplifier models.
- As alluded to before, we decided on using shunt resistors to measure current. With programmable gain amplifiers that will work in tandem with the microprocessor to offer a better sensitivity at certain ranges. ed

Functionality: Even though the ACS712 (Hall-Effect Sensor) is not a part of our design anymore; it is still a reliable sensor to verify proper functionality of our current sensing resistor. We used an oscilloscope to acquire signal at the output pin of the Hall-Effect sensor, and compared it to the output of the CSR circuit. This way, we can compare the signals and see how improvement we get by using the shunt resistor current sensing.



Hall-Effect Sensor

Accuracy: One way to check the accuracy of the current sensing circuit is by comparing the measured current value with the multimeter 34410A. Verifying accuracy is the biggest challenge of our design. With the multimeter 34410A, we can validate the accuracy in the range from 0A to 3A (RMS). Current above 3A will be validated mathematically using component models obtained from lower range measurements. Basically, once we test the circuit in the range from 0A to 3A, we can characterize the resistor properly. At the same time, we will be able to account for the resistance of internal wires of the system, DC offset voltage of the amplifier, and shunt resistance degradation.

Agilent 34410A Multimeter

Set of components to be tested: there are two types of devices that we will test: analog and digital. The reason that we have two sets of components because of their power consumption patterns vastly vary over time. For most analog devices, they draw continuously. On the other hand, digital devices draw current at partial duty cycle.

Analog appliances:

1) Weller soldering gun. (Rated power consumption 50 W)

2) Incandescent light bulb (Rated power consumption 100 W)

3) Hair dryer (Rated power consumption 1000 W)

Digital Appliances:

1) iPhone charger (Rated power consumption 5W)

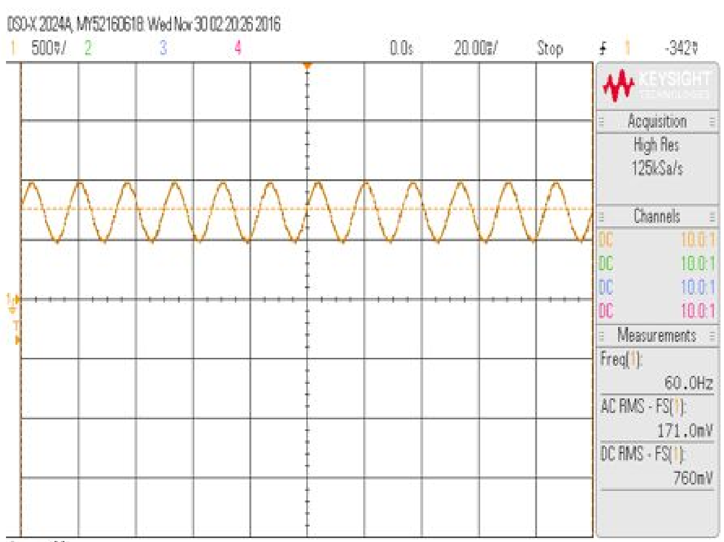
2) Macbook charger (Rated power consumption 85W)

Finally, once we obtain all the current measurements for above appliances, we will compute their power consumption. Then we will compare them with the rated power consumption given by manufacturers/companies. The difference will then be used to evaluate our design and see what need to be changed to improve the accuracy.

Results:

Analog:

1) Weller soldering gun. (Rated power consumption 50 W)



All RMS Values:

$$I = \frac{(171\ mV)}{10 * 8.3 * 0.004}$$

$$I = 516\ mA$$

Using Multimeter

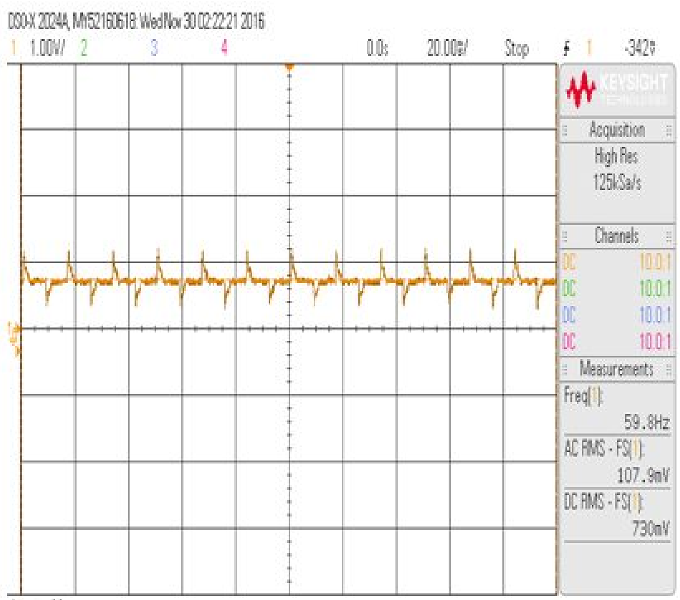
$$I = {\sim}510\ mA$$

Analog Known Load Test Results

Here we have our test results for the soldering iron. All though we tested all the load above, we chose to include only one results for the sake of redundancy. Here we have connected our CSR circuit in series with the load, the soldering iron. And on the output of the circuit, we can see an analog waveform with an AC voltage RMS-amplitude of 171mV. From here, if we work backwards and divide by the gain that were used for this scenario, 10 and 8.3; we can get the original voltage. Then if we divide by the resistor value, approximately 4mΩ, we get the amperage the load is currently pulling, 516mA. This was close to the value measured from the multimeter; within 2% to be precise. This is a value we are very happy with.

Digital:

1) iPhone charger (Rated power consumption 5W)



All RMS Values:

$$I = \frac{(107.9\ mV)}{50 * 8.3 * 0.004}$$

$$I = 65\ mA$$

Using Multimeter:

$$I = \sim 61\ mA$$

Here, the story repeats itself. We tested a number of digital loads, but only chose to include one for redundancy. Similar to the analog scenario, we have place the cell phone charger in series with our CSR and displayed the output on the oscilloscope. The results was just as expected. Digital devices tend to have this very low duty cycle draw pattern that we discovered early into this semester. As we can see, there is no frequency modulation; therefore, if we take the VAC RMS-amplitude and divide by the subsequent gains and the resistor values we arrive at a current of 65mA. This being close to the measured multimeter value of 61mA; within 7% to be precise.

Now that we know the circuit performs well with given loads, we have to take a second look at the system and be sure that it is prepared for all scenarios that it may run into during real-world usage.

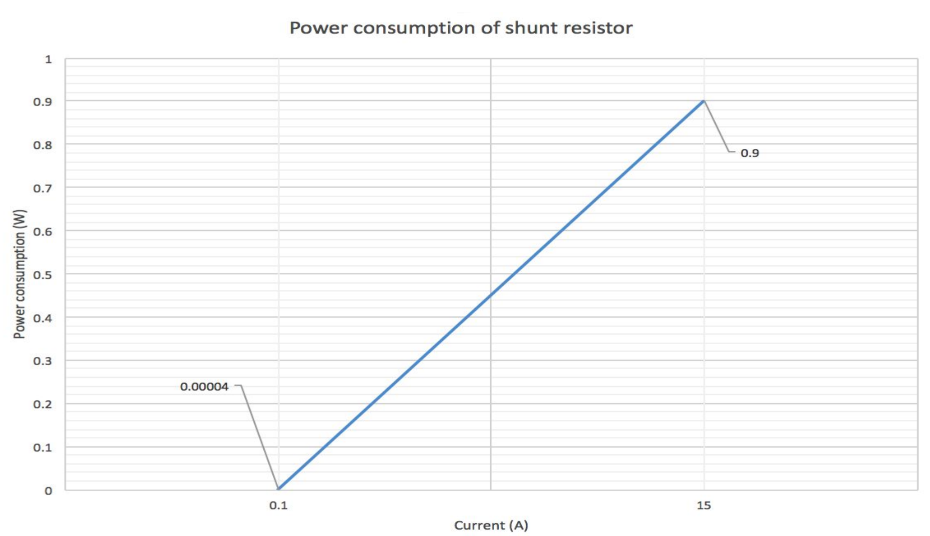
4 Challenges

Below are the main challenges we will be, or currently are, running into during the implementation of our device

Power Consumption:

Low-power consumption is a common characteristic amongst many electronic devices these days. Without a doubt, engineers have taken this challenge seriously, and a challenge it is. Everyday, engineers are experimenting with new ways to shave off all the milliwatts they can. Likewise with our design. We want to minimize the design so that the overall energy consumption of the device is lower than 5W. This number is based on the lowest power that is being consumed by a typical phone charger, and similar smart-plugs on the market today. To achieve this specification, we have to be very careful with passive components like resistors, microcontrollers, and wireless communication. The obvious trade-off would be between the functionality and power consumption; the more we worry about power-consumption, the more the circuit's capabilities are limited. Critical tasks like signal processing and data transferring can not be further simplified. Thus, we are pretty much relying on the software algorithm to improve power management.

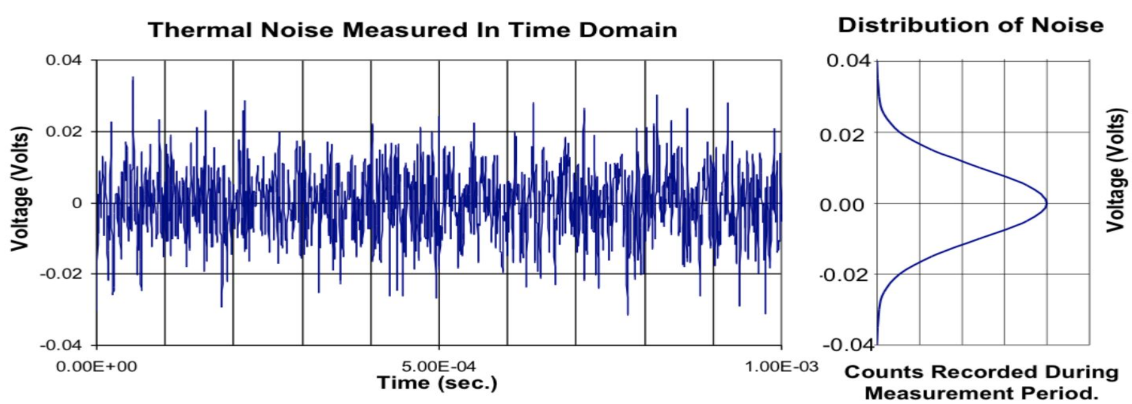
A rough estimation of power consumption will give us some ideas to see what we can do to improve the power consumption of the whole device. However, the power consumption of the shunt resistor will stay constant. The plot below shows the power consumption of a 4 mOhms resistor at the minimum current of 100 mA and maximum current of 15 A. At the max current measurement, the power consumption of the resistor will dominate (about 1W). At the small current measurement, the power consumption of the resistor is relatively small (around microwatt). The power consumption of the MCU will dominate so we can perform power gating the reduce power consumption.



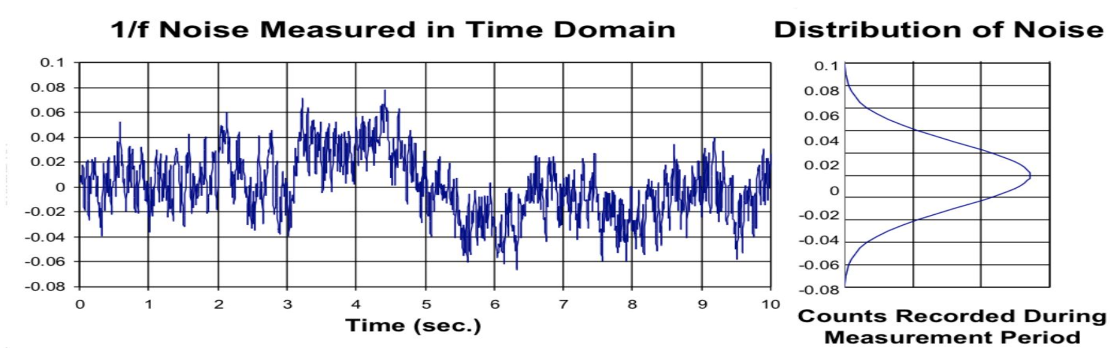
Noise:

When dealing with measuring signals for verification, noise is major issue. Noise mainly comes from components themselves. We started addressing the noise issue by replacing the ACS712 sensor with the resistor current sensing circuit. However, there is still background noise in the system. And, at low current levels, the noise level can be quite dominant. Our aim is to increase the signal-to-noise ratio enough to make accurate measurements possible at all of our specified ranges. In order to achieve this goal, we first need to account for all the different types of noise in the system. There are two prominent types of noise in our the current sensing circuit: broadband noise and flicker noise. Below plots demonstrate the distribution of noise in the system. Both types of noise have a mean of zero so we can apply an averaging filter at the output. However, we must design the filter carefully so that we don't lose any of our signal's data points.

1) Broadband noise



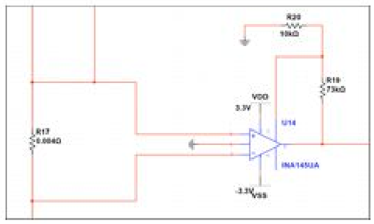
2) Flicker noise



Courtesy of Texas Instruments

Analysis: Using this knowledge we can begin the analysis of noise in our system. From the theorems of fundamental noise analysis, the majority of noise can be closely approximated by looking at the first stage amplifier. That amplifier in question is the TI INA145. The results can be seen below:

- First Stage Amplifier: INA145



We need a gain of 8.33

$$Bandwidth = \frac{500\ kHz}{8.33} = 60.02\ kHz$$

$$e_{bb} = 90 \frac{nV}{\sqrt{Hz}}$$

$$\mathrm{BW_n} = k_n f_h = 1.57 * 60.02\ kHz$$
$$= 94.23\ \mathrm{kHz}$$

Noise Analysis (1)

$$Broadband\ noise: E_{nBB} = 27.627\ uV$$

$$Flicker\ noise: E_{nFlicker} = 0.64509\ uV$$

$$Resistor\ noise: E_{nR} = 2.959\ uV$$

$$Total\ noise = \sqrt{27.627^2 + 0.64509^2 + 2.959^2} = 27.79\ uV$$

$$Simulation: 22.61\ uV$$

$$SNR = \frac{100\ mA\ * 4\ mOhms\ * 8.3}{27.79\ uV} = 14.39$$

Noise Analysis(2)

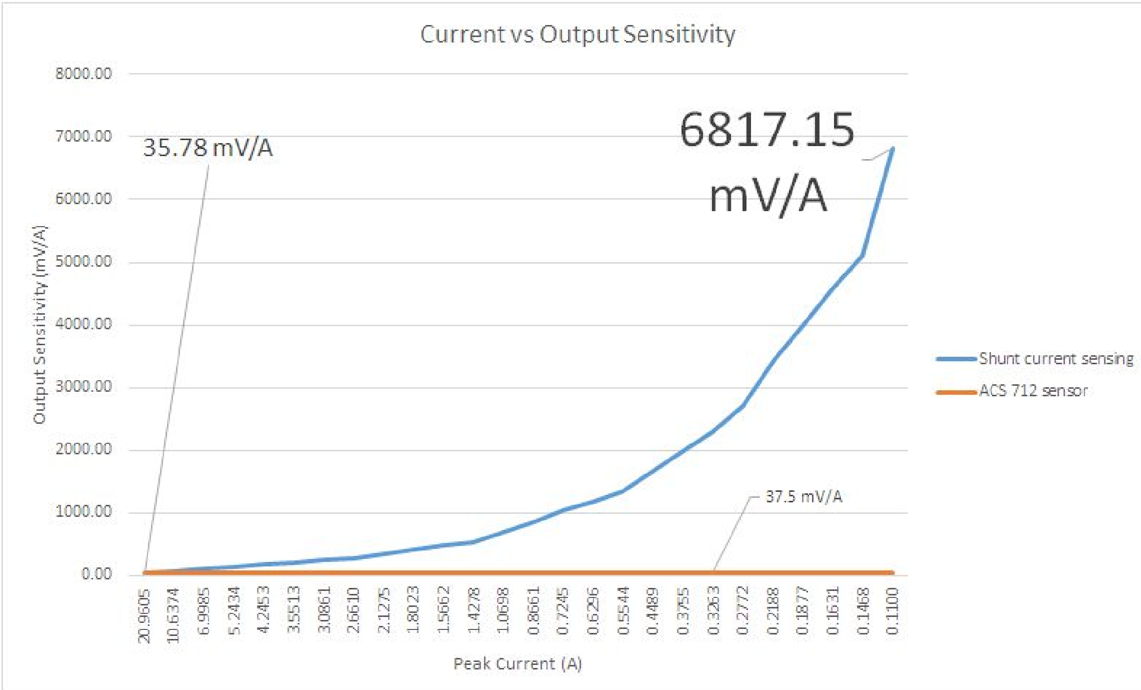
Here we used a combination of hand calculations, simulations, and physical measurement to model the noise in the system. Starting from the top, with the first stage amplifier having a fixed gain of 8.3, we can find the -3dB frequency by dividing the Gain-Bandwidth Product by our given gain. In this case it turned out such that the -3dB frequency is right at 60 Hz. Next, we pulled the noise spectral density from the data sheet, as you can see in the bottom left on the first image. Then the last equation on the first slide, in the bottom right, is an ideal brickwall approximation of our bandwidth. It takes the -3dB frequency and multiplies it by pi/2, which is an accepted theoretical factor for brickwall approximation.

On the following image, Noise Analysis (2), we calculated the total broadband noise using the spectral density and multiplying that number by the square root of the bandwidth, alluded to on the previous image. The flicker noise was calculated by hand by using the values from the datasheet. Since this noise varies directly with the inverse of frequency, we knew it would be more dominant in our low-frequency model. Fortunately, when multiplied the constant from the datasheet by the appropriate frequency, it was still relatively negligible. Lastly the, resistor noise comes from the thermal noise equation. We then found the magnitude of that noise, which was very similar to a simulated noise analysis on MultiSim; therefore we were confident in our calculations. Finally, calculating our signal-to-noise ratio at our current floor, where the noise will be the most dominant, we arrive a ratio of 14.39. This value is well above our target signal-to-noise ratio of 4; hence, moving forward noise will be a negligible issue.

Accuracy:

The energy measurement that our team is designing intends to handle the current ranging from -15A to 15A, RMS. This is a relatively large current range. Our approximate sensitivity is 4V/30A, which is approximately 133 mV/A. With that small of a sensitivity, the accuracy of the measurement relies mainly on the Analog-to-Digital converter, in the sense that the ADC must be capable of converting relatively small decimal voltages. If we measure high current values, there is nothing to be concerned about because the output analog signal can be easily read by the microcontroller. However, once we reach the level the of milliamp current, the accuracy reduces dramatically. This is the level where the noise dominates the actual current. To combat this effect, we have implemented two stages of amplification controlled by the MCU. Therefore, if the signal is not noticeable at the ADC's input, the MCU can respond by changing the gain, using its auto-ranging algorithm. This constraint will cause a few ranges to have lower resolution.

Analysis:



Measured Output Sensitivity Graph

As you can see above, we have plotted current versus the output sensitivity of our circuit. We first went about measuring the gain of the amplifier and cross-checking their values to the ideal values designated in the datasheet. Since there was a small percentage error; we had to take this into account. We measured a handful of differential voltages across our resistor, the output voltage, and took note of the gain specified. Once we tabulate this data in a spreadsheet, we were able to calculate the measured sensitivity for that current value. Once we got all the possible current values we could measure, we extrapolated the data to fit the measurements for which we did not have a comparable peak current. The extrapolation of the data was done minimally;therefore, we are very confident that the this graph will faithfully model the operating characteristics. This analysis serves the purpose of not only seeing what sort of accuracy is capable at different ranges, but it can help us down the road when we are writing functions on the microcontroller to compute power. Instead of importing a sizable look-up table on the microprocessor's memory, we can model this relationship using a polynomial and place that on the MCU. Then, the MCU will auto-range the gain such that the ADC can read the output from the hardware circuit. Once the MCU does this, it will look for the peak value, then it can relate it back to a point on this graph, pull out the corresponding sensitivity and make a quick n' easy calculation.

Network Connection:

Some challenges we will be having is the ability to connect to a wifi network that has a security passcode. For now we are assuming there will not be a security passcode, but we may need to figure out a way to incorporate to be able to please a wider range of users. Some other challenges we may have include the strength of the wifi connection to our CC3200 microcontroller. Often times it is difficult to establish a wifi signal on the controller and we have to reboot the controller multiple times to get the connection established. We are able to verify the wifi connection is established through the Tera Term shown in Appendix 5. The controller has to be close to the source of the connection due to a small power strength of the microcontroller and the size of the antenna on the microcontroller as well. To get around this problem, we are planning on using a separate antenna providing us a greater range, reliability, and stronger power to connect to the wifi.

Data Handshaking:

A large challenge will be when we have multiple energy measuring devices running at the same time. The challenge occurs when two or more units send data at the exact same point in time and how the receiver will be able to choose to interpret one, and when that's finished interpret the remaining. Otherwise only one signal could be recognized or there could be an error in the system not allowing the signal to go through.

ADC:

At the end of our hardware circuit we will have an analog signal. In order to store this data into our database, we will need to convert this signal from analog to digital. We do this through our CC3200 microcontroller. We will be using the datasheet to look at the appropriate channels to use for the ADC; however, some challenges may occur when dealing with high noise or low voltage signals as far as transferring that signal over.

5 Timeline

See Appendix 2.

5.1 First Semester

The first semester will be spent mostly in the conception, research, and prototype phases as seen in the timeline, located in appendix 2. Once the conception was complete we split into two groups: hardware and networking/software, with two members occupying the hardware and the remaining four being dispatched to the network/software side. At this point, each team researched their respective components to achieve our goal. For example, the hardware team looked at ways of power measurements and conceived a circuit that should theoretically output a voltage that is inputted into the microprocessor. From there the software team will write algorithms to compute average power and transmit the data. Likewise, the software team is researching libraries and tinkering with development kits to better understand how to process and transmit data all the way back to the user. Once the research is complete, we simulate the proposed methods. If all goes as planned, we will order parts to test their behavior with a physical prototype. From there, we plan on refining the prototype until the following semester; where we will move into final implementation.

5.2 Second Semester

The second semester will involve more software prototyping. At this stage, we should have a working sensor prototype that can send messages to a central hub. We will be testing the reception of the UDP server and the storage of the data into the database. Next, we will refine the prototype of the web application to include all of the user functionality we want. We will do UI testing and make sure that the aesthetics of the site are up to par. In addition, we will finalize the PCB design of our hardware section, which includes the power sensors and the microcontroller.

We will finish the semester with documentation and preparing for our final presentation. Documentation will include a complete explanation of our final approach and implementation. Preparing for our final presentation will include comparing our solution to commercial systems and also rehearsing for the oral presentation.

6 Conclusions

In conclusion, there are three main goals of our project and several steps to achieve these goals. The first goal is to build a working power sensor that measures and transfers data of the consuming power of a certain device. The second goal is to deal with data from multiple devices simultaneously and to display them properly on the user's interface. The third goal is to compare power consuming of the same kind of device from different users to present the average for that kind of device. To achieve these goals, we plan to use a Hall effect transimpedance amplifier to get a proportional output voltage or current which data can be processed and transferred through microprocessor. Then we build a network portion of the project which includes the data processing and receiving code, HTTP server and database system. At last, we will add functions on the web application to allow users to monitor and control the energy cost of every device.

7 References

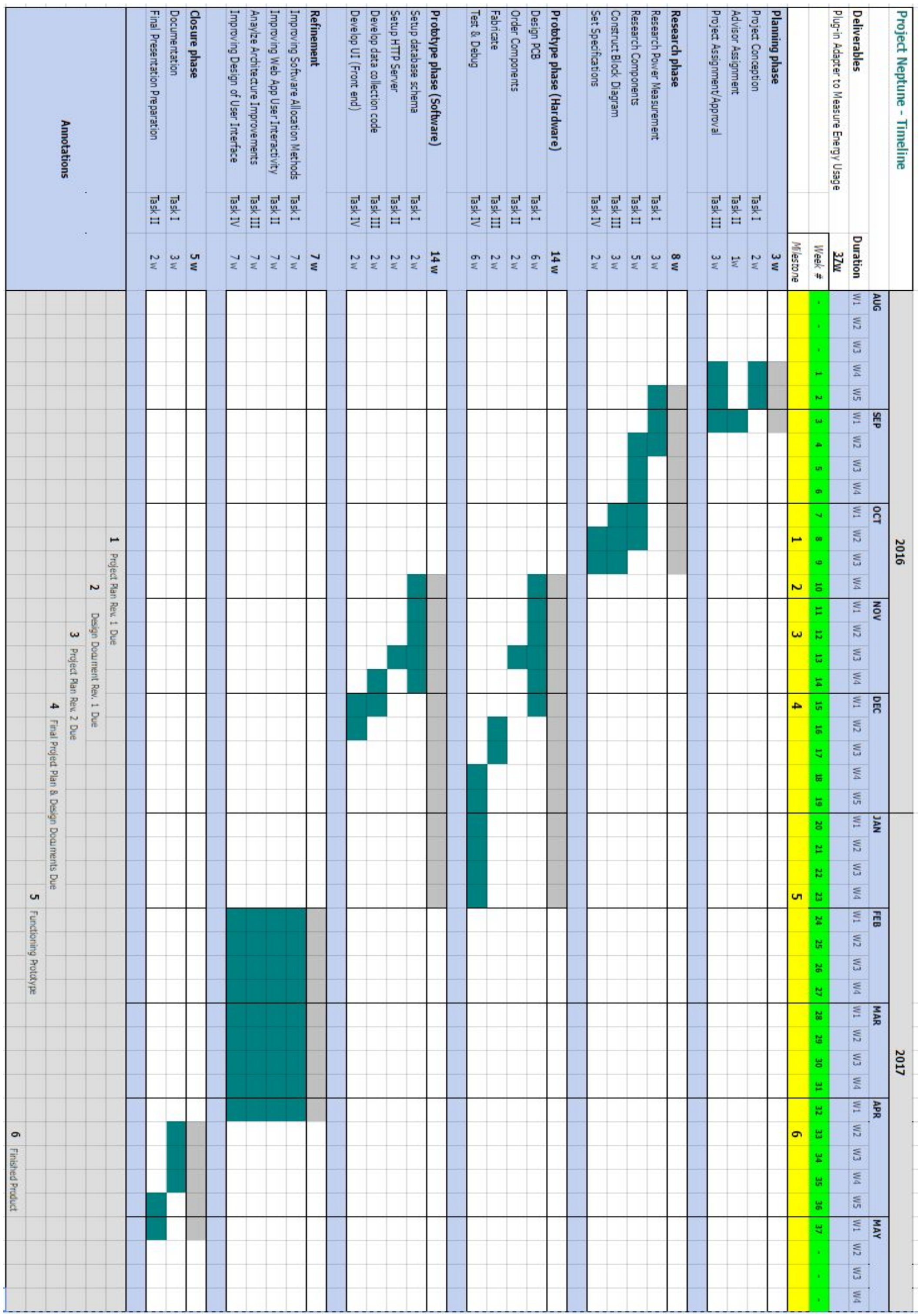
1. "The Energy Detective Electricity Monitor." *The Energy Detective Electricity Monitor*. TED, n.d. Web. 12 Oct. 2016. <http://www.theenergydetective.com/>.
2. "CC3200." *CC3200 | SimpleLink CC3x | Wireless MCUs | Description & Parametrics*. Texas Instruments, n.d. Web. 20 Nov. 2016. <http://www.ti.com/product/CC3200>.
3. Develco Products. (n.d.). *Smart Plug.* Retrieved from http://www.develcoproducts.com/media/1459/smart-plug-datasheet.pdf
4. Maxim Integrated. (2002). *Linear Charger for Single-Cell Li+ Battery - MAX1898.* Retrieved from https://datasheets.maximintegrated.com/en/ds/MAX1898.pdf
5. ROHM Semiconductor. (2015). *PSR400ITQFJ2L00 Datasheet.* Retrieved from http://www.mouser.com/ds/2/348/psr-951869.pdf
6. Stencel, L. (2014, Mar 4). Effective Surge And Lightning Strike Protection For AC And DC Power Line Applications. Retrieved from http://powerelectronics.com/circuit-protection-ics/effective-surge-and-lightning-strike-protection-ac-and-dc-power-line-applicat?page=2
7. Texas Instruments. (2000, September 27). *Programmable Gain Difference Amplifier - INA145.* Retrieved from http://www.ti.com/lit/ds/sbos120/sbos120.pdf
8. Texas Instruments. (2000, September 27). *Programmable Gain Difference Amplifier - INA146.* Retrieved from http://www.ti.com/lit/ds/symlink/ina146.pdf
9. Texas Instruments. (2014, December 5). *CC3200MOD SimpleLink Wi-Fi and Internet-of-Things Module Solution, a Single-Chip Wireless MCU.* Retrieved from http://www.ti.com/lit/ds/symlink/cc3200mod.pdf
10. Yarborough, B. (2012, Jan 6). Components and Methods for Current Measurement. Retrieved from http://powerelectronics.com/power-electronics-systems/components-and-methods-current-measurement

8 Appendices

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. You may also include your Gantt chart over here.

1. System design

2. Project Timeline



3. CC3200MOD datasheet

Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community

TEXAS INSTRUMENTS

CC3200

SWAS032F – JULY 2013 – REVISED FEBRUARY 2015

CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things Solution, a Single-Chip Wireless MCU

1 Device Overview

1.1 Features

- CC3200 SimpleLink Wi-Fi—Consists of Applications Microcontroller, Wi-Fi Network Processor, and Power-Management Subsystems
- Wi-Fi CERTIFIED™ Chip
- Applications Microcontroller Subsystem
  - ARM® Cortex®-M4 Core at 80 MHz
  - Embedded Memory
    - RAM (Up to 256KB)
    - External Serial Flash Bootloader, and Peripheral Drivers in ROM
  - 32-Channel Direct Memory Access (µDMA)
  - Hardware Crypto Engine for Advanced Fast Security, Including
    - AES, DES, and 3DES
    - SHA2 and MD5
    - CRC and Checksum
  - 8-Bit Parallel Camera Interface
  - 1 Multichannel Audio Serial Port (McASP) Interface with Support for Two I2S Channels
  - 1 SD/MMC Interface
  - 2 Universal Asynchronous Receivers and Transmitters (UARTs)
  - 1 Serial Peripheral Interface (SPI)
  - 1 Inter-Integrated Circuit (I^2C)
  - 4 General-Purpose Timers with 16-Bit Pulse-Width Modulation (PWM) Mode
  - 1 Watchdog Timer
  - 4-Channel 12-Bit Analog-to-Digital Converters (ADCs)
  - Up to 27 Individually Programmable, Multiplexed GPIO Pins
- Dedicated External SPI Interface for Serial Flash
- Wi-Fi Network Processor Subsystem
  - Featuring Wi-Fi Internet-On-a-Chip™
  - Dedicated ARM MCU
    Completely Offloads Wi-Fi and Internet Protocols from the Application Microcontroller
  - Wi-Fi and Internet Protocols in ROM
  - 802.11 b/g/n Radio, Baseband, Medium Access Control (MAC), Wi-Fi Driver, and Supplicant
  - TCP/IP Stack
    - Industry-Standard BSD Socket Application Programming Interfaces (APIs)
    - 8 Simultaneous TCP or UDP Sockets
    - 2 Simultaneous TLS and SSL Sockets
  - Powerful Crypto Engine for Fast, Secure Wi-Fi and Internet Connections with 256-Bit AES Encryption for TLS and SSL Connections
  - Station, AP, and Wi-Fi Direct® Modes
  - WPA2 Personal and Enterprise Security
  - SimpleLink Connection Manager for Autonomous and Fast Wi-Fi Connections
  - SmartConfig™ Technology, AP Mode, and WPS2 for Easy and Flexible Wi-Fi Provisioning
  - TX Power
    - 18.0 dBm @ 1 DSSS
    - 14.5 dBm @ 54 OFDM
  - RX Sensitivity
    - –95.7 dBm @ 1 DSSS
    - –74.0 dBm @ 54 OFDM
  - Application Throughput
    - UDP: 16 Mbps
    - TCP: 13 Mbps
- Power-Management Subsystem
  - Integrated DC-DC Supports a Wide Range of Supply Voltage:
    - V_{BAT} Wide-Voltage Mode: 2.1 to 3.6 V
    - VIO is Always Tied with VBAT
    - Preregulated 1.85-V Mode
  - Advanced Low-Power Modes
    - Hibernate: 4 µA
    - Low-Power Deep Sleep (LPDS): 250 µA
    - RX Traffic (MCU Active): 59 mA @ 54 OFDM
    - TX Traffic (MCU Active): 229 mA @ 54 OFDM, Maximum Power
    - Idle Connected (MCU in LPDS): 825 µA @ DTIM = 1
- Clock Source
  - 40.0-MHz Crystal with Internal Oscillator
  - 32.768-kHz Crystal or External RTC Clock
- Package and Operating Temperature
  - 0.5-mm Pitch, 64-Pin, 9-mm × 9-mm QFN
  - Ambient Temperature Range: –40°C to 85°C

4. CC3200MOD board

2 Getting Started

2.1 Download and Install Software

Download and install the following software:

- CC3200 SDK package.
  - This guide assumes the use of the default installation folder *C:\TI\CC3200SDK_1.2.0*.
- Tera Term (or similar software)
  - Tera Term link: http://en.sourceforge.jp/projects/ttssh2/releases/

2.2 Configure Board

The jumpers on the CC3200-LAUNCHXL should be connected as shown in Figure 1. It may be necessary to move a jumper from P58-VCC to SOP2.

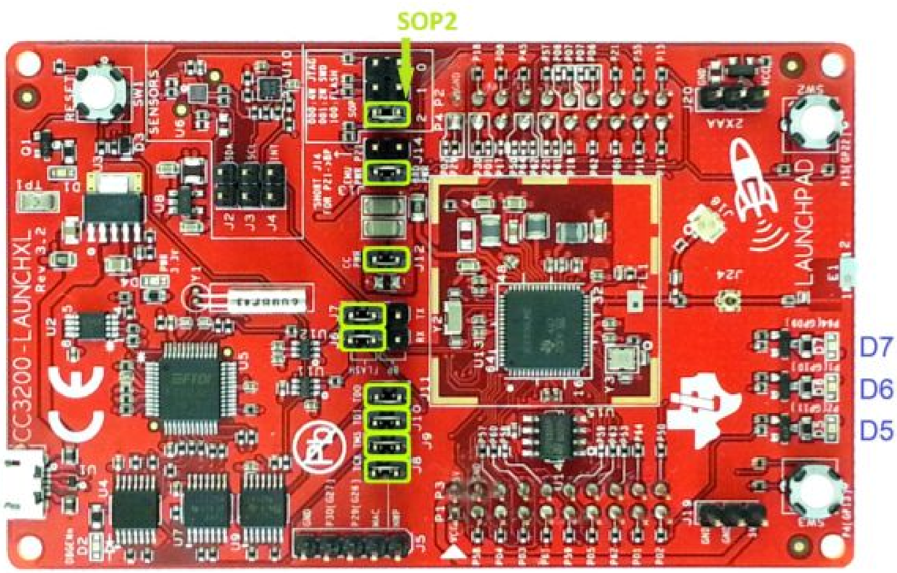


Figure 1. Jumpers on the CC3200-LAUNCHXL

1. Connect the CC3200-LAUNCHXL to the PC using the provided micro-USB cable.
2. The CC3200-LAUNCHXL is now visible in the Device Manager as shown in Figure 2. Note the COM port number that appears.

5. CC3200MOD wifi connection