Wireless Energy Measurement System

DESIGN DOCUMENT

Team Number: May1725

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)

Team Email: may1725@iastate.edu

Contents

1 Introduction

1.1 Project statement

With our project 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 up 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 is connected to a user interface. That means that we have a sensor that can measure the power being used by a certain device. Send that data via wifi to a central hub that can then take that data and again send it to a web application that can present that data in a easy to use and understand format.

Our next goal would be 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 need to have a central hub that will take and store all the data to be used that has been gathered from the sensor. The central hub will be in the form of either a raspberry pi or ISU server. Here the data will be added to a database that the user interface will communicate with in order to get the information needed to present to the user.

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

Include any/all possible methods of approach to solving the problem. Discuss what you have done so far. What have you tried/implemented/tested etc. We want to know what you have done.

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.

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 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 IP22 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.

3.1.1.2 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.

3.1.2 Functional

3.1.2.1 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

3.1.2.2 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

3.1.2.3 Hardware System Requirements

1. The hardware shall not be a source of significant power drain (power consumption less than 5W).
2. The current sensor must be able to measure bidirectional current with an RMS-amplitude of 15A.
3. The hardware shall be sensitive to changes in current and voltage measurement to the value of at least 100mV/A.
4. The hardware shall not make the load operate in conditions that are harmful to it and/or affect performance.
5. The hardware shall have a user-controlled switch within it.
6. 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).
7. The hardware shall provide a sinusoidal output with minimal phase shift and no frequency modulation.
8. The hardware shall operate in temperate range from -25 to 125 degree Celsius.

3.2 Proposed Design/Method

3.2.1 Proposed System Block diagram

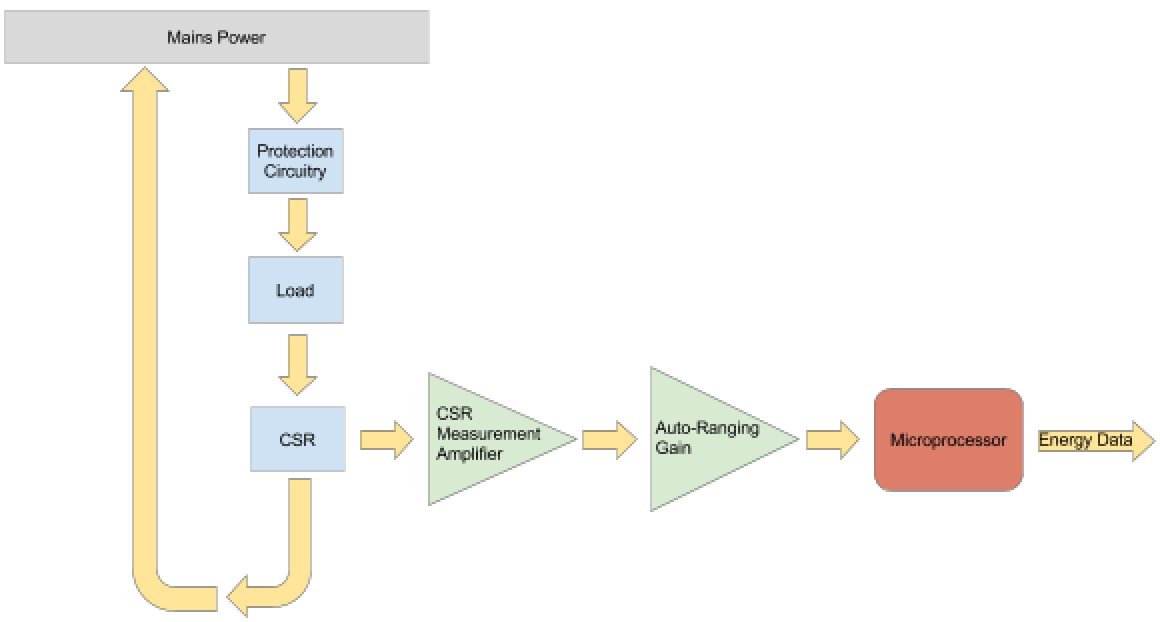


Figure 3.2-1: Hardware Block Diagram

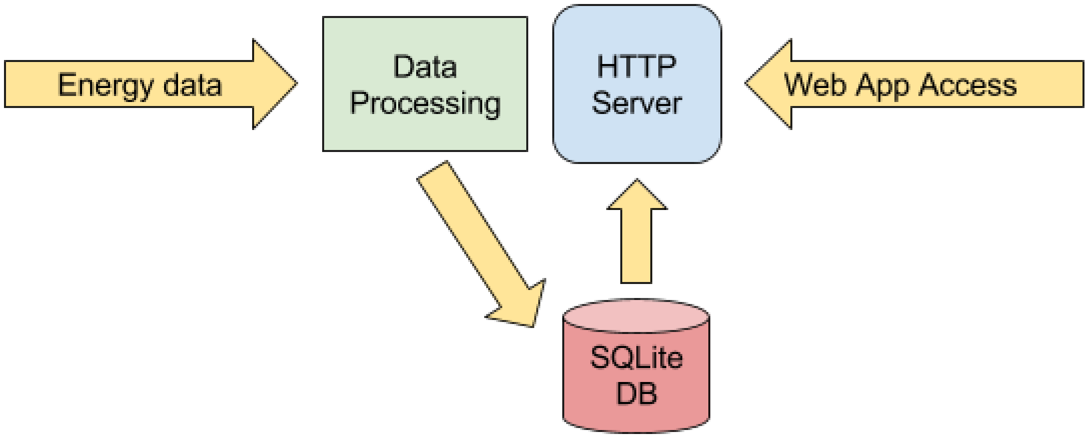


Figure 3.2-2: Central Server (Network) Block Diagram

3.2.2 Hardware Assessment of Proposed methods

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, it the user chooses to switch it off personally. When it comes to power measurement, we choose to measure on the low side of the load. We do this as to not disturb the load (i.e. avoid brownouts). Once the load has consumed the power, we measure the remaining voltage 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 differential amplifier. We then take this amplified signal and run it through two more inverting amplifiers. The reason we have two is for 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 chose to use low-side measurement mainly 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 current greater than 15A RMS is detected. Once this issue is covered, the measurement differences between high & low side are trivial.

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

Moving to the output of the differential 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, 1.8V. 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 40% percent of 1.8V, 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, all tasks are executed by the network.

3.2.3 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.

3.3 Design Analysis

3.3.1 Software Design Analysis

Originally, we were using the MSP430 microcontroller from Texas Instruments. We originally thought this would be a suitable microcontroller to use. However, we realized that we came across some difficulties with connecting to the wifi and so we had to do some more reasearch on a better controller that would enable us to connect to wifi easier. So we eventually came up with using a CS3200 microcontroller from TI that has the ability to connect to wifi very easily. To connect to the wifi we had to download all the necessary software provided by TI on their website and in the instruction manual. We then followed the rest of instructions provided by TI to implement the code they require on the TI software system to allow the microcontroller to pick the wifi network and then bypass the security on the wifi to connect the microcontroller to the network. We got that to work; however, the reliability of connecting to the network is not that great. We are having to possibly go to a a different antenna on the microcontroller since the one on the CS3200 is extremely small or put more power into the microcontroller to create a stronger signal output. We are planning on using a SQLite database, but we still need to set that up and have the microcontroller send data to the database. We also are still going to need a webapp as well that we will create using Javascript and Python.

3.3.2 Hardware Design Analysis

A lot of things have changed during the design phase for analog measurement. Originally, we were using a Hall Effect sensor, the ACS712. This sensor was placed in series on the high side of the load. This method included running the current through a very small 1.2mΩ equivalent resistor. Though the physics of the Hall Effect, the ACS712 would produce a proportional voltage output. This differs from our current method that uses a differential amplifier in that the proportional voltage that the ACS712 produces was completely isolated from the load circuit. This allowed for high voltage and current measurement, if we so desired. It also greatly reduced power consumption and component count. Therefore, we thought of it as a catchall solution. Through implementation and testing we found that the sensor did not provide the sensitivity to changes in current or accuracy in the low range of current measurement we desired. The Hall Effect naturally produces a lot of noise to due to limitations in the dielectrics. Hence, when we tried to measure currents around 100mA, we received a noise envelope that made it impossible to retrieve any data from.

Knowing we would not be able to use this method anymore we deferred to the conventional method of using a current sensing resistor and cascaded amplifiers. We are in the initial testing phases of this method but, so far this method provides better accuracy and a better range of measurement. The current sensing resistor method allows for sensitivities of 10x that of the ACS712, and the current floor for the current sensing resistor method is much lower than the ACS712, due to the fact the resistor nor the amplifier introduce noticeable noise into the system. The only drawbacks are a miniscule, but not relatively negligible, increase in power consumption and an increase in the number of components. Moving forward we plan on finding ways to decrease total power consumption and sensitivity.

4 Testing/Development

4.1 Interface Specifications

Our hardware/software interfacing consists of two interfaces: the analog-to-digital conversation at the microcontroller's ADC, and the network interface between each sensor node and the base station. Each interface is detailed below.

Our current sensing circuit consists of a current-sensing resistor and a set of amplifiers. The output of the amplifiers goes into the ADC of our microcontroller, the TI CC3200. The reference voltage of the ADC is about 1.4V and it has 12-bit resolution. This is more than enough to satisfy our technical requirements.

The second interface is between the CC3200 and the base station. This connection is achieved over the homeowner's local network. Data packets are sent through the WiFi radio of the CC3200 over UDP. The packets arrive at the base station and are decoded by a UDP server. The decoded samples are passed into the database for later use.

4.2 Hardware/software

4.2.1 Software

Some of the software we will be using in our project used in the testing phase. For our CS3200, we had to include the TI CS3200 SDK, Tera Term, and Code Composer Studio 2.0. We also will be using SQLite software to create our database. The CS3200 SDK and Tera Term and the tools needed to connect with the CS3200 and the Code Composer Studio is where we write the code for the microcontroller. The results of the code/microcontroller are displayed on the Tera Term.

The base station is a Raspberry Pi, running a variant of Linux. This will host our UDP server and the HTTP server. The UDP server is responsible for receiving packets from the sensing stations. The HTTP server hosts the web application that the user accesses. We will likely be using some sort of web application framework like Django or AngularJS.

4.2.2 Hardware

For current measurement, we have two alternatives: hall-effect sensor and shunt current resistor. We try to exploit the most performance out of these devices to obtain the most accurate current measurement as much as possible. However, in the end we decided shunt current resistors were the best option.

Hall-effect sensor (ACS712)

The chip above is ACS712, a very common and cost effective current sensor. The basic mechanism of the sensor is that it senses magnetic field generated by the current flowing through the internal conductor. This sensor has a really small internal resistance (1.2 mOhms). Because of this characteristic, ACS712 can handle high current.

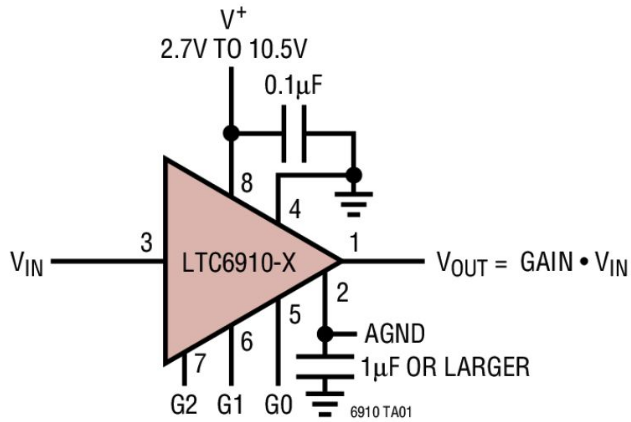
Shunt resistor (1 mOhms)

In shunt current measurement, resistors plays a key role in modeling Ohm's law. If both voltage and resistance are known, one can calculate the current easily. However, easiness generates challenges as well. Most resistors are temperature-dependent. So having a constant-resistance resistor is extremely important in current measuring. The above resistor is from ROHM semiconductor, designed for low-variation, high power-rating. This resistor type is chosen because it has low level of error in current measurement when comparing it against its carbon counterparts.

Difference Amplifier (INA145)

Measuring the voltage across a device can be done by using a difference amplifier. The output of the difference amplifier is proportional to the difference between two input terminals. However with INA145, the gain can be adjusted by varying the resistive feedback network. This feature gives our design more flexibility in terms of gain.

Programmable gain amplifier (LTC6910)



Even with the gain from the difference amplifier, the output voltage is relatively small (approximately uV for 100mA). The programmable gain amplifier is used to overcome this issue. With the LTC6910 amplifier, we can easily set the gain by controlling three digital inputs. This makes auto-ranging possible in our design.

4.3 Process

4.3.1 Software Process

The testing of the c3200 chip from TI we are using was pretty simple. TI provided a guide to help with connecting to the internet. We had to download CCS, Tera Term, and the CC3200 SDK in order to configure the board to be able to connect to the internet. After that we accesed some provided test code that helped us to contact the wifi network. We had to customize a few bits of code during this step, in order to give the system passwords for getting into the network. Most of our problems we faced in this process were due to the inconsistency of our chip being able to connect to a wifi network. This is a problem we will still have to figure out in the next few weeks.

Now we are in the process of creating a server and web app that will store data and interact with the user. We plan on using SQLite as our server in conjunction with Python. The web application will be written using the common combination of Javascript, CSS, and HTML. As we sure up the wifi connectivity of the cc3200 these two projects will become more of a priority, but for now we are making sure that what we are working on is bug free.

4.3.2 Hardware Process

a. Hall-effect sensor (ACS712 sensor)

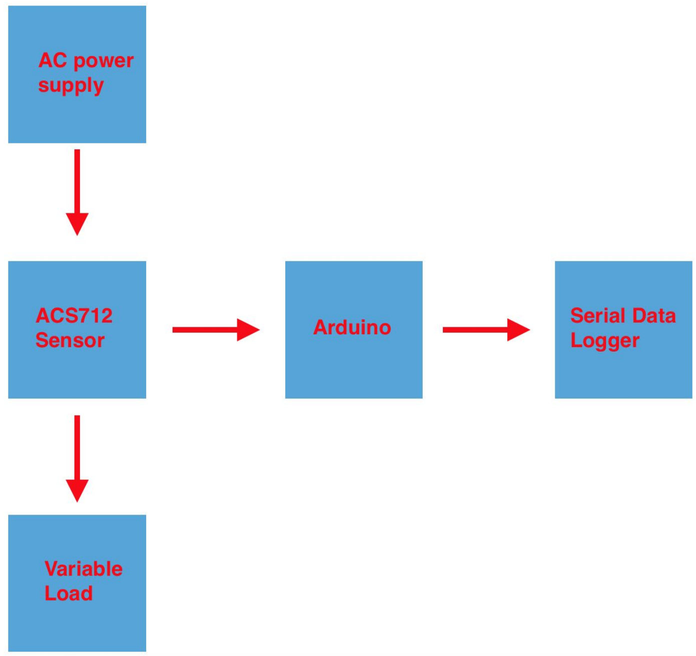


Figure 4.3-1: Testing ACS712 sensor flow chart

Testing the ACS712 sensor is done by simply connecting the sensor in series with the power supply and load. We modified the AC plug socket so that we can switch between different loads for different measurements. The output pin of the ACS712 will be connected directly to the analog pin of the Arduino microcontroller. Arduino is used because our CC3200 is not fully functional at this stage of the design. In addition, Arduino interface provide serial monitor, which makes it easier to obtain raw data. During the testing phase, we connect different appliances (ranging from high to low power consumption device) to the modified AC plug socket. Then we record the data through serial port at 250 kBaud rate.

b. Current sensing using shunt resistor (1 mOhms resistor)

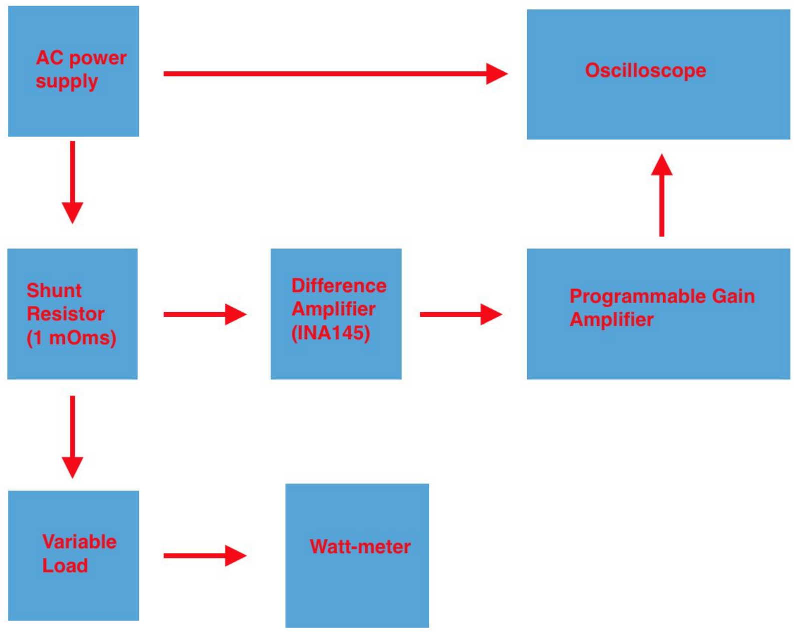


Figure 4.3-2: Current sensing using shunt resistor flowchart

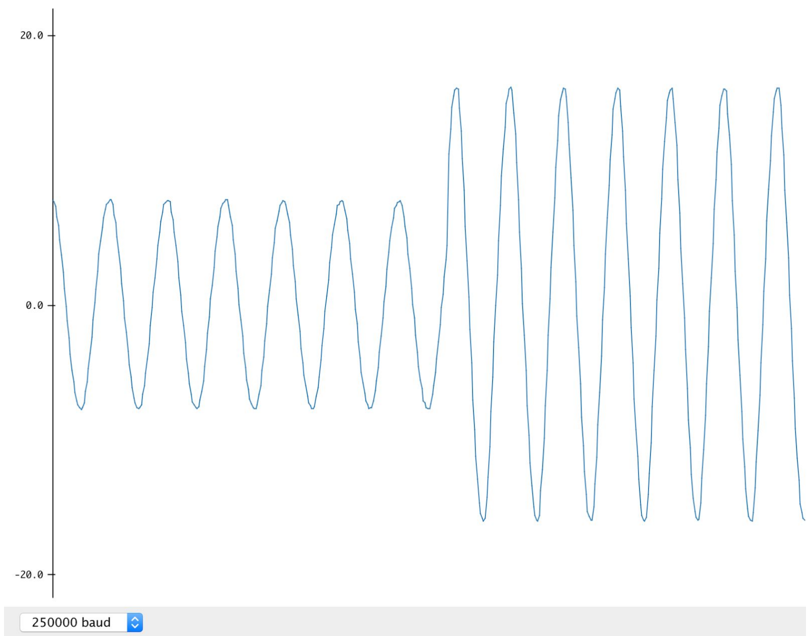
Testing current sensing using the shunt resistor is a bit more complicated than the previous hall-effect sensor. First, we connect the series shunt resistor in series with AC power supply and load. Then, we connect both input terminals of the difference amplifier to the shunt resistor. Because the voltage drops across the resistor is relatively small (~microvolts), it is very insufficient to look at the voltage using the oscilloscope. The next step would be amplifying the voltage difference using the programmable gain amplifier. The output voltage is now large enough to be observed using the oscilloscope. In addition, we will observe the AC voltage. In most cases, the voltage provided by power company doesn't vary much. However, during the testing stage, we want to make sure everything is accurate as much as possible. Once we obtain the voltages from voltage and current sensing, we can mathematically analyze the data to revert back to the actual values. The oscilloscope that we use in the lab has this math capability so we can calculate the instantaneous power directly. The final step of testing would be comparing the measured power with the value from from Wattmeter. The testing cycle will repeat for each load.

5 Results

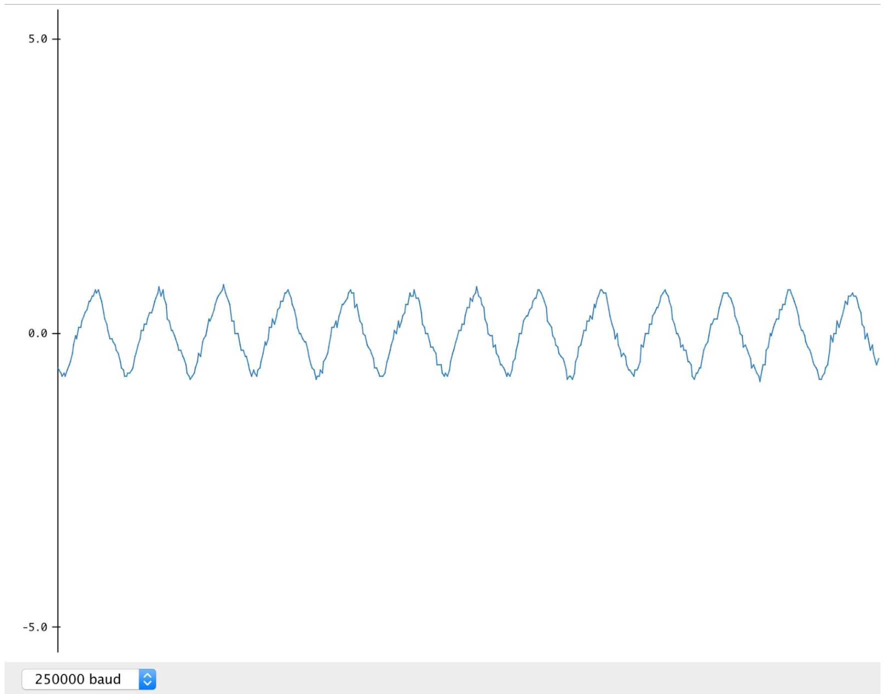
5.1 Hardware results:

a. Current Measurement Using the ACS712 Sensor

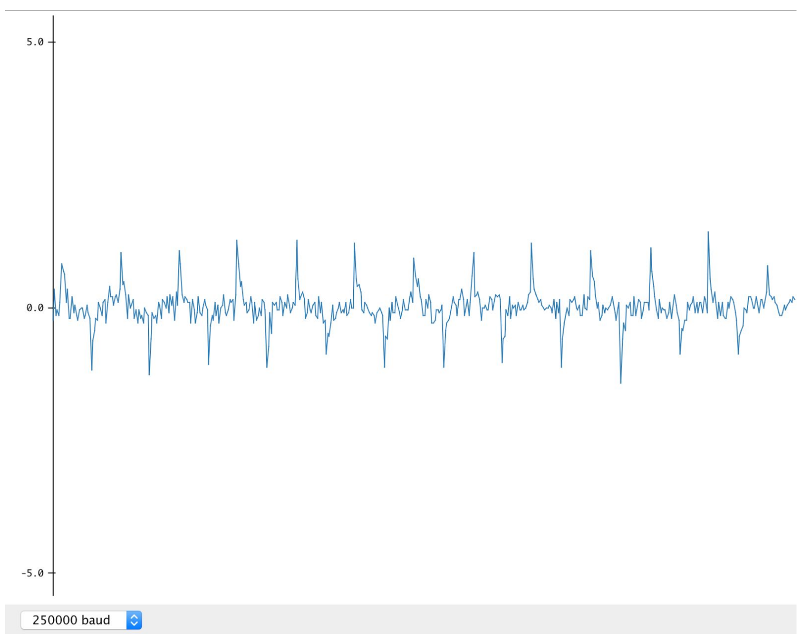
We tested different loads in series with ACS712 to obtain sample current measurements. Our aim is to verify what current level the sensor is capable of measuring. Even though, these tests doesn't verify the accuracy of the sensor; it does provide us with some visualizations of current pattern. (note: the y-axis for these plots below is in ampere unit)



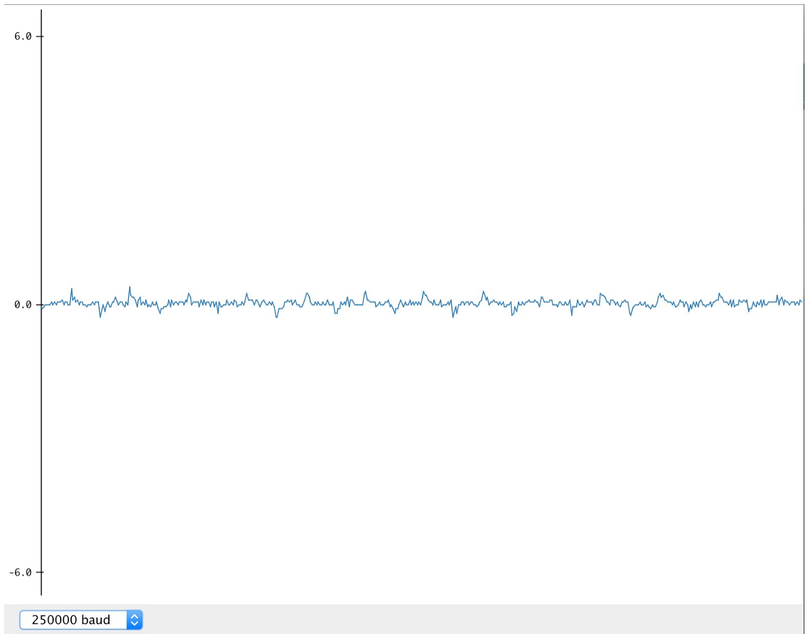
Load 1: 1000 W heater



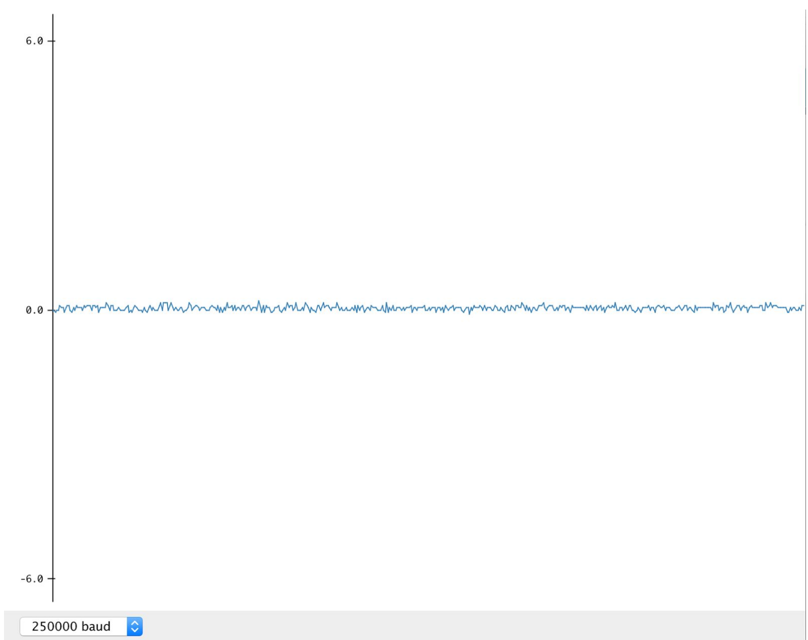
Load 2: Mini table fan



Load 3: 27 W fluorescent light bulb



Load 4: iPhone charger



Load 5: No load

Comments:

- From this set of the data, we learned that most resistive loads such as: heaters, incandescent bulbs, and water boilers draw sinusoidal current. On the other hand, devices like phone charger draws current at different duty cycle. This result fundamentally changed how we sample data. With sinusoidal current, low sampling rate can be applied to calculate the current because the current varies in a predictable way. However, if the current changes randomly, a higher sampling rate is needed in order to capture all the data points.

- Secondly, we found that ACS712 is capable measuring high current level without any issue. However, once the current level drop below 1A, ACS712 seems to perform poorly due to the noise level. This is something that we can't improve much because it is a common characteristic of the Hall-effect sensor.
- The failure of the ACS712 is that it is sensitive to magnetic field. We thought about shielding the sensor but considering its performance, ACS712 might not be the best current method.
- The main target of the design is to account for low-power consumption appliances. However, with the test results from the ACS712, we have to defer to the shunt resistor method to achieve better a better current measurement range.

b. Shunt Current Resistor

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

<u>Load 1: 1200 ohm resistor - 1 mA(RMS)</u>

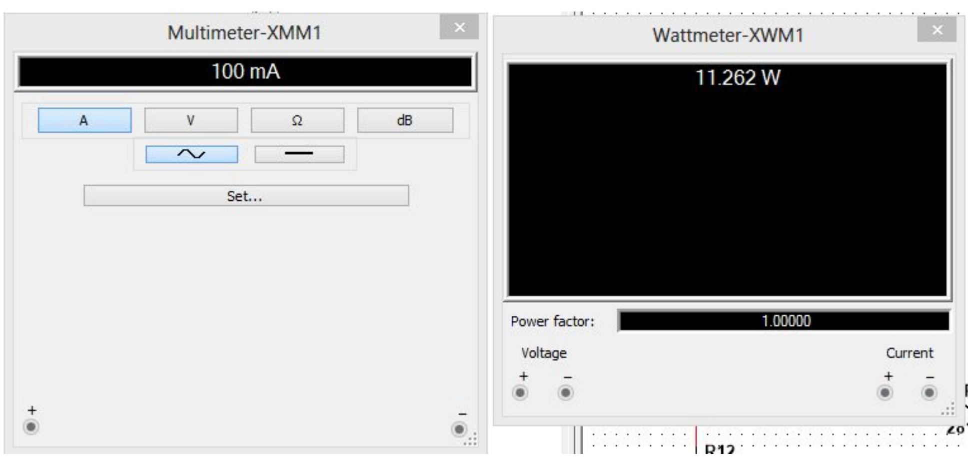


Figure 5.1-1: Actual current and power consumption of the load

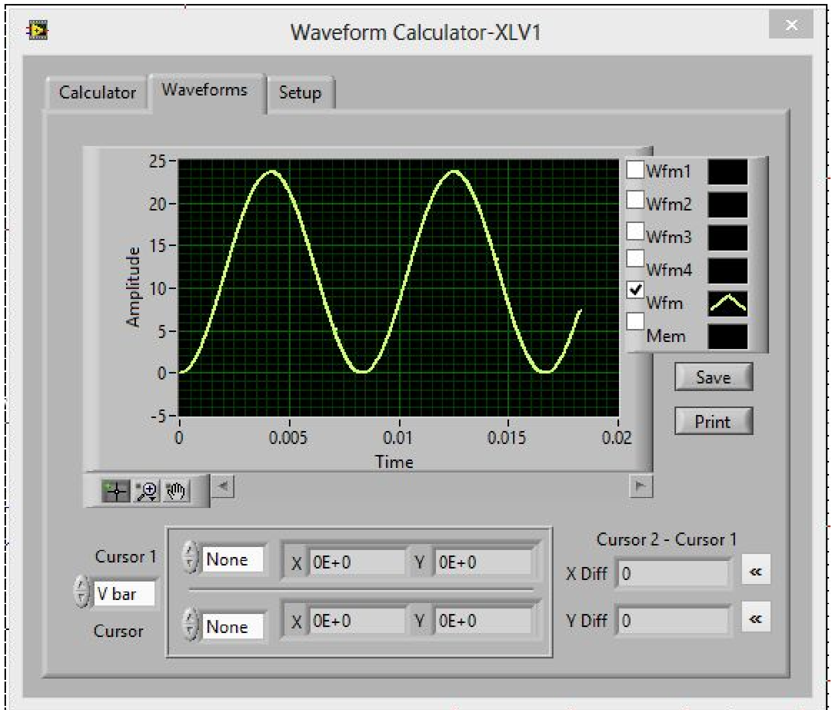


Figure 5.1-2: Instantaneous power (Average of 11 Watts)

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

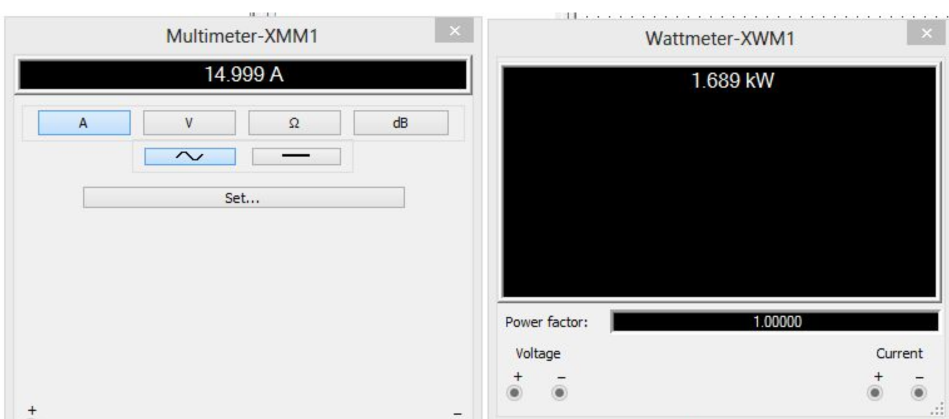


Figure 5.1-3: Actual current power consumption of the load

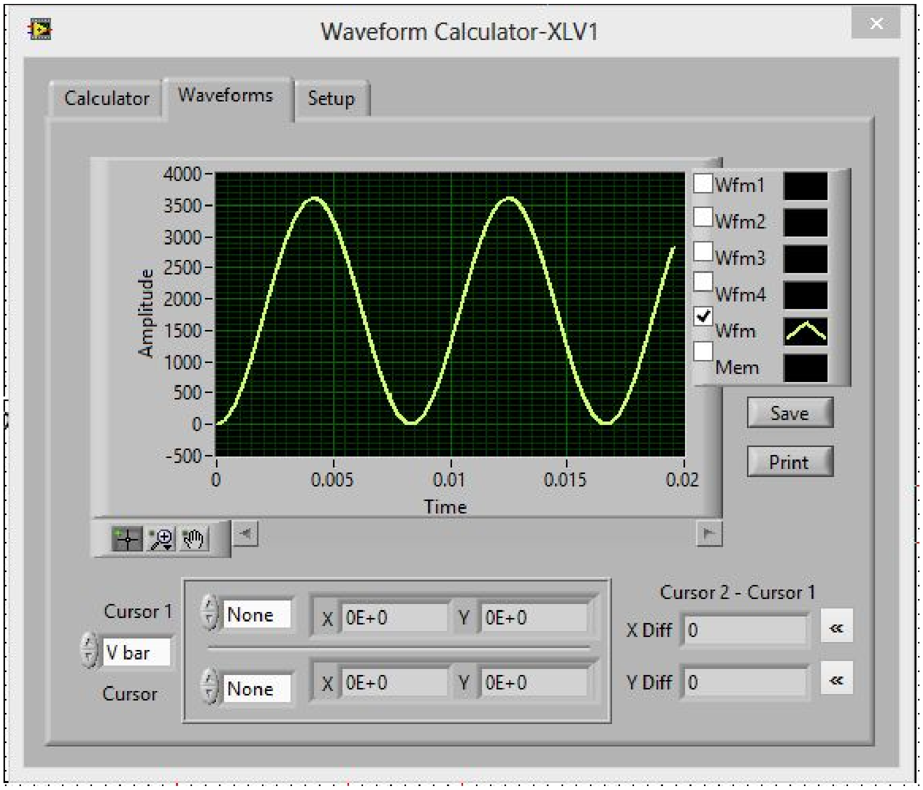


Figure 5.1-4: Instantaneous power (Average of 1.7 kW)

<u>Comments:</u>

- 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 the programmable gain, we can divide the current range into multiple ranges. By doing this, better resolution can be achieved

5.2 Software Results:

For the software side we have obtained the cc3200 board (pictured below) for connecting our measuring circuit to wifi. We have programed the board to connect to wifi, and are in the process of creating a server that can communicate with the board through its' wifi link.

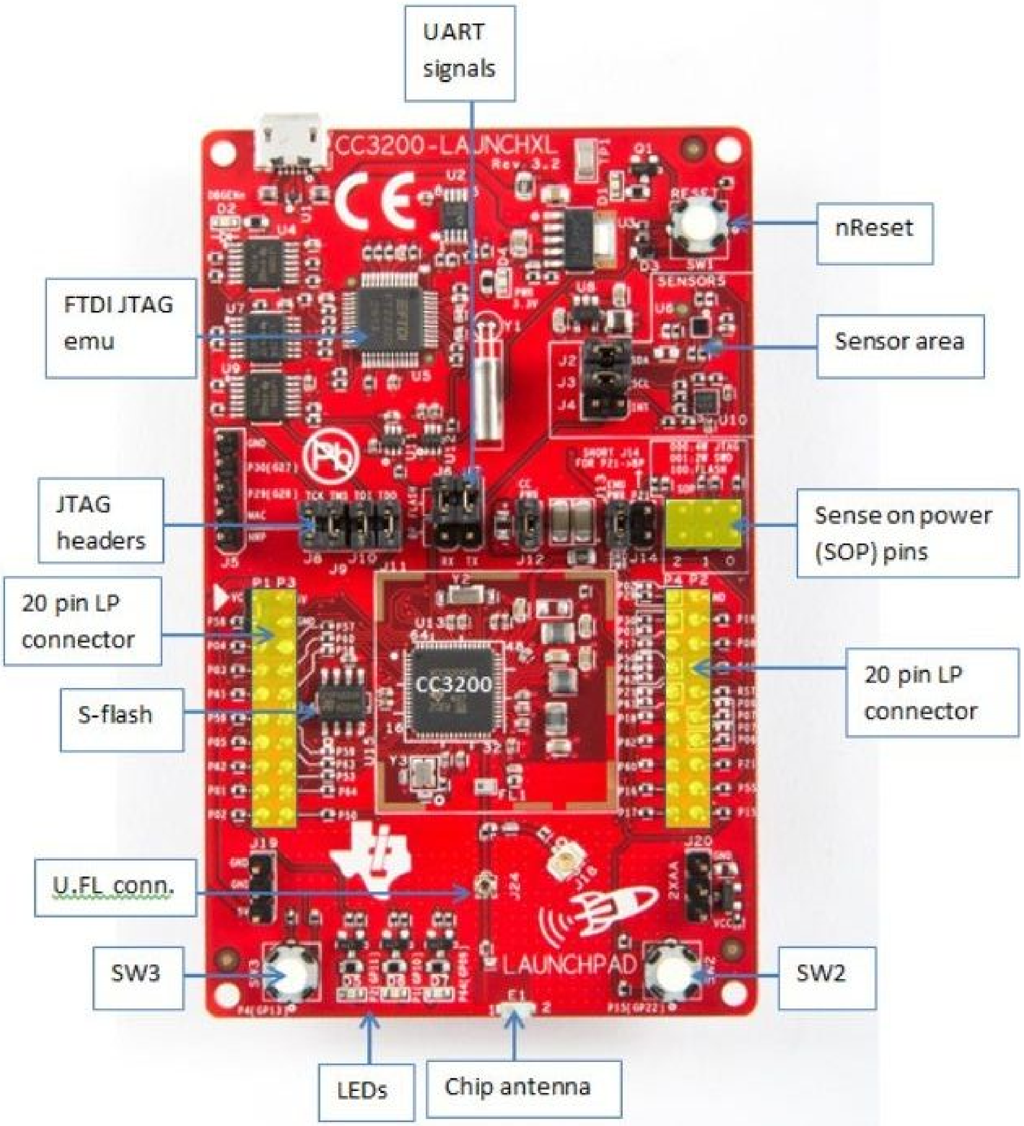


Figure 5.2-1: CC3200 launchpad

6 Conclusions

From now on, we have designed the circuit which is used to measure the consuming power and the actual components that are used to build the circuit. Also, we obtained and programmed the cc3200 board which is used to connect measuring circuit to Wifi. We are dealing with the specific parameters setting, database building and web application designing. First our main goal is to be able to build a working power sensor unit that is connected to a user interface. Our next goal would be 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 to have our web application compare the normal power usage for a certain device with what the user's device is using. To achieve our goals, we compared many plans and picked the best ones. First, we choose low-side measurement mainly due to product availability and cost. Then, we choose UDP as our data transmission rather than TCP, because it is well-suited for data streaming and it is simpler and easier to implement. And we use shunt resistor because it can measure current as low as 100 mA.

7 References

7.1 Software References

TI CS3200 Microcontroller Datasheet and SDK Software Download:

"SimpleLink Wi-Fi CC3200 Software Development Kit (SDK)." *CS3200*. Texas Instruments, n.d. Web. 04 Nov. 2016. <http://www.ti.com/tool/cc3200sdk>.

Tera Term Software Download:

"Download File List - Tera Term - OSDN." *Download File List - Tera Term - OSDN*. OSDN, n.d. Web. 04 Nov. 2016. <https://osdn.net/projects/ttssh2/releases/>.

Code Composer Studio 2.0 Download:

"Category:Code Composer Studio V6." *Texas Instruments Wiki*. N.p., 12 Sept. 2016. Web. 04 Nov. 2016. <http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v6>.

7.3 Hardware References

Develco Products. (n.d.). *Smart Plug.* Retrieved from http://www.develcoproducts.com/media/1459/smart-plug-datasheet.pdf

Maxim Integrated. (2002). *Linear Charger for Single-Cell Li+ Battery - MAX1898.* Retrieved from https://datasheets.maximintegrated.com/en/ds/MAX1898.pdf

ROHM Semiconductor. (2015). *PSR400ITQFJ2L00 Datasheet.* Retrieved from http://www.mouser.com/ds/2/348/psr-951869.pdf

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

Texas Instruments. (2000, September 27). *Programmable Gain Difference Amplifier - INA145.* Retrieved from http://www.ti.com/lit/ds/sbos120/sbos120.pdf

Texas Instruments. (2000, September 27). *Programmable Gain Difference Amplifier - INA146.* Retrieved from http://www.ti.com/lit/ds/symlink/ina146.pdf

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

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

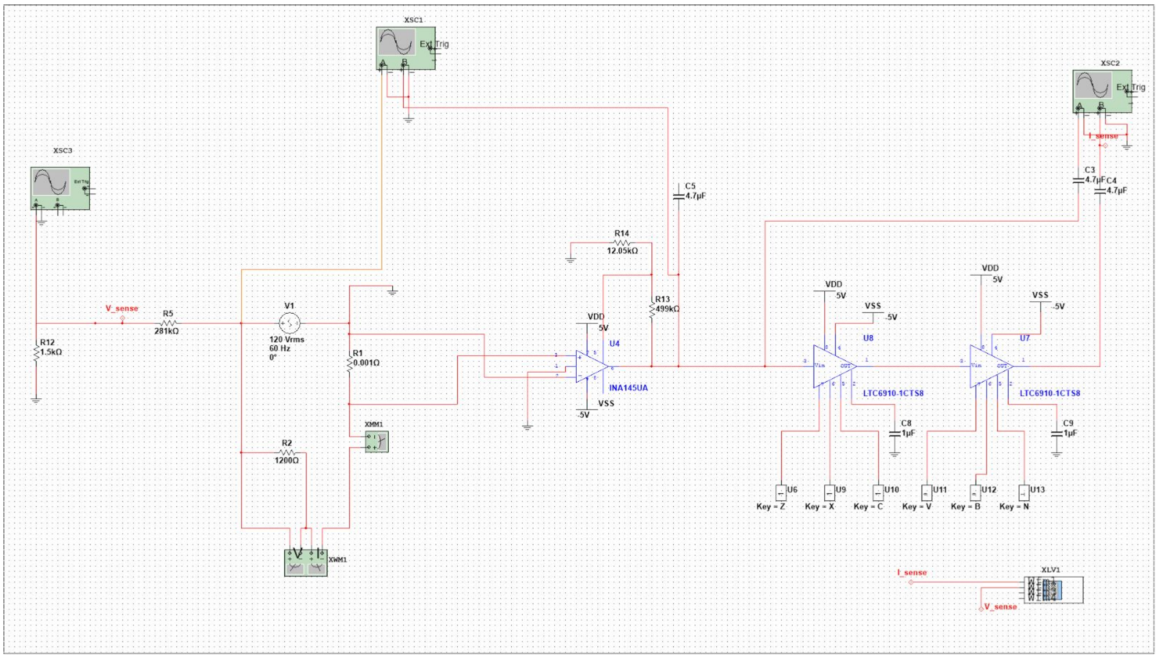


Figure A-1: Shunt Resistor Current Sensing Circuit

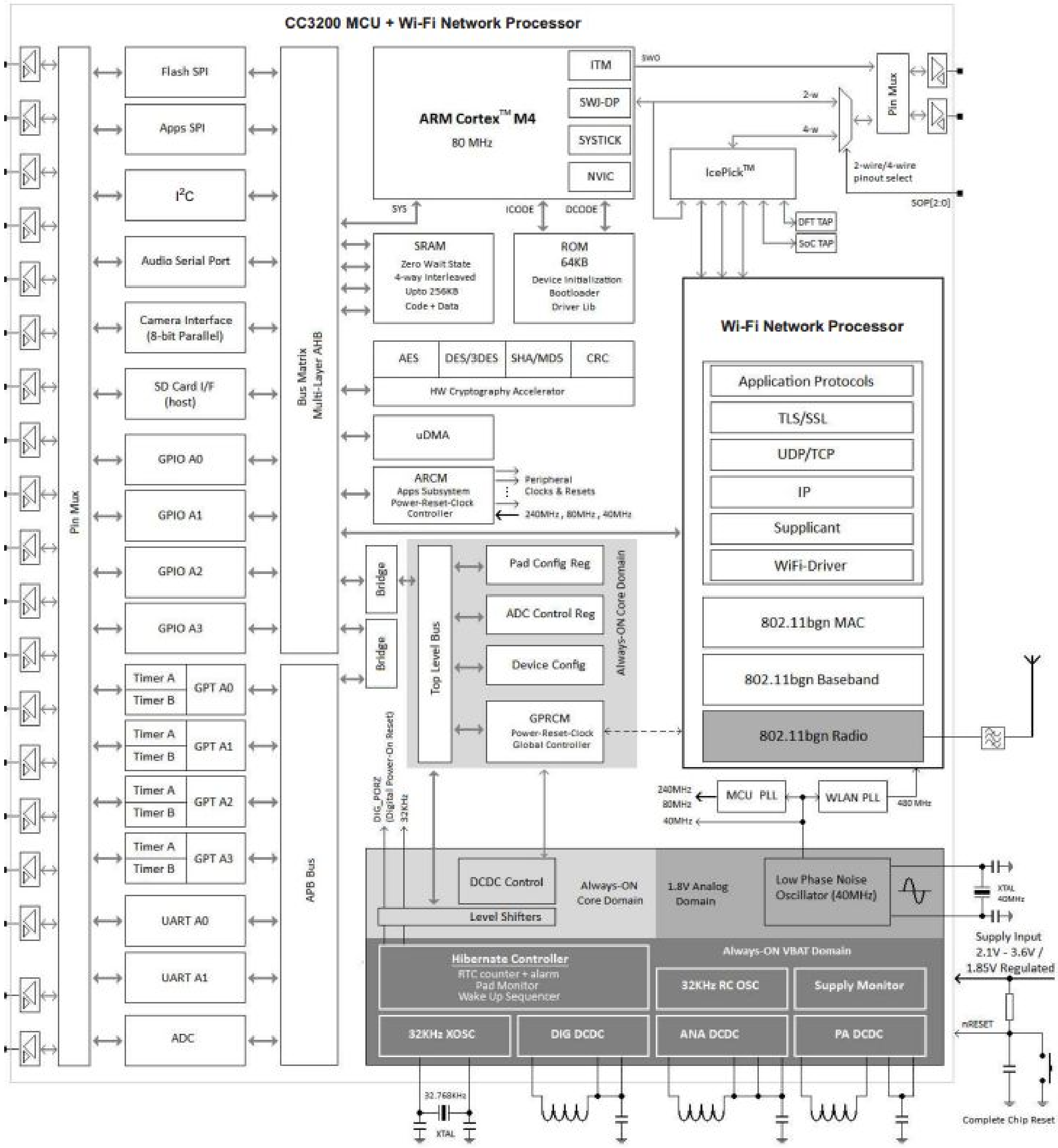


Figure A-2: CC3200 Wireless Module Functional Block Diagram