Project Plan

Wireless Energy Measurement System

MAY1725

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 pie 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

3.1 Previous work/literature

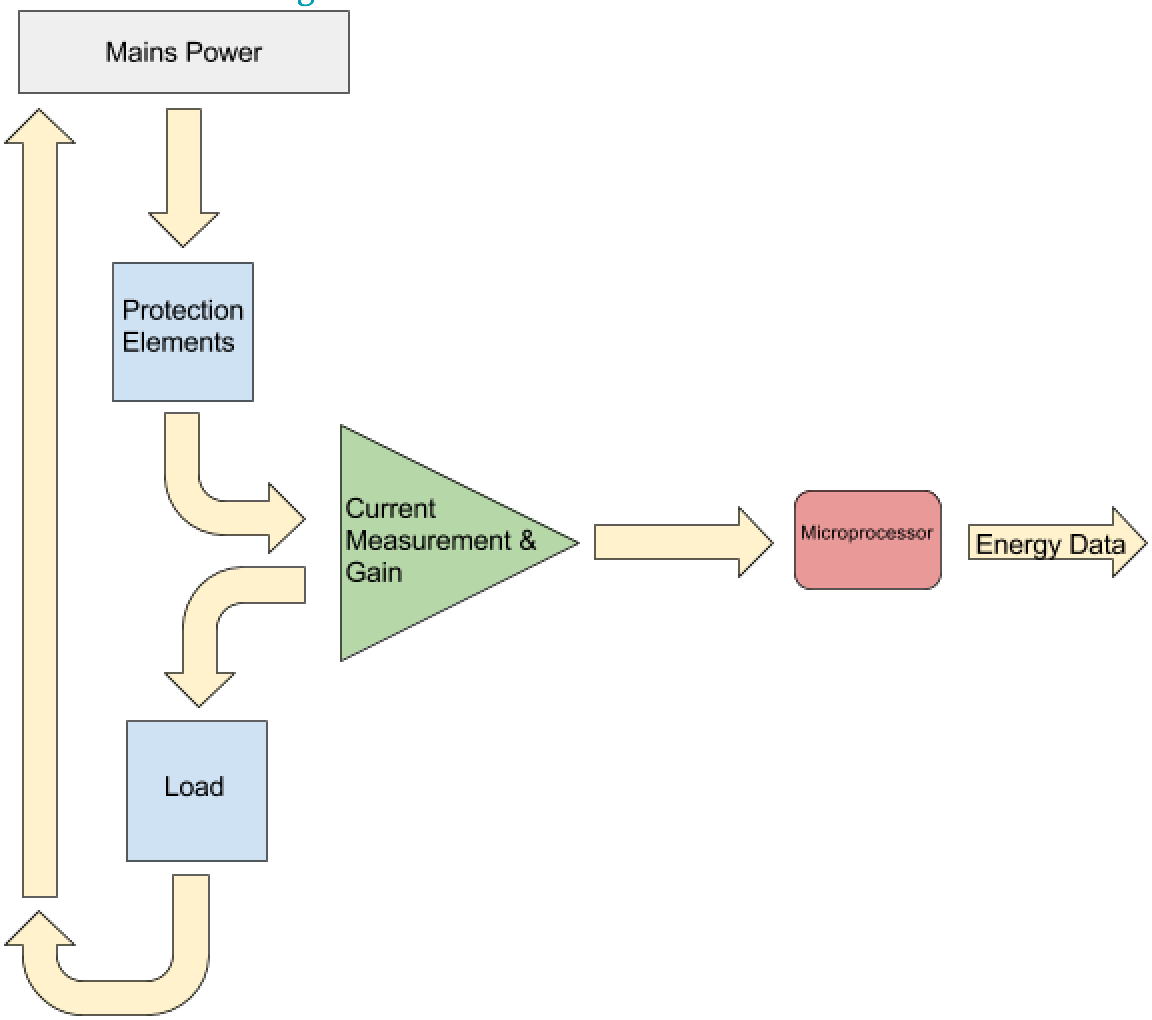
There are several commercially-available energy monitoring software systems that we looked at before starting design. We wanted to make sure that our software and hardware could match or exceed the specifications of the systems on the consumer market.

For hardware, most systems use a similar setup to our proposed setup. We develop a circuit that measures current directly on the high side of the load. This current produces a proportional output, either current or voltage. Then that data is received and processed by a microprocessor and the data is transmitted. However, where our solution differs is in the way of current measurement. As most proposed solutions use a current sensing resistor and differential amplifier, we choose to use a Hall Effect transimpedance amplifier. This method greatly cuts down on power consumption, but the main trade-off comes in the form of sensitivity to changes in current.

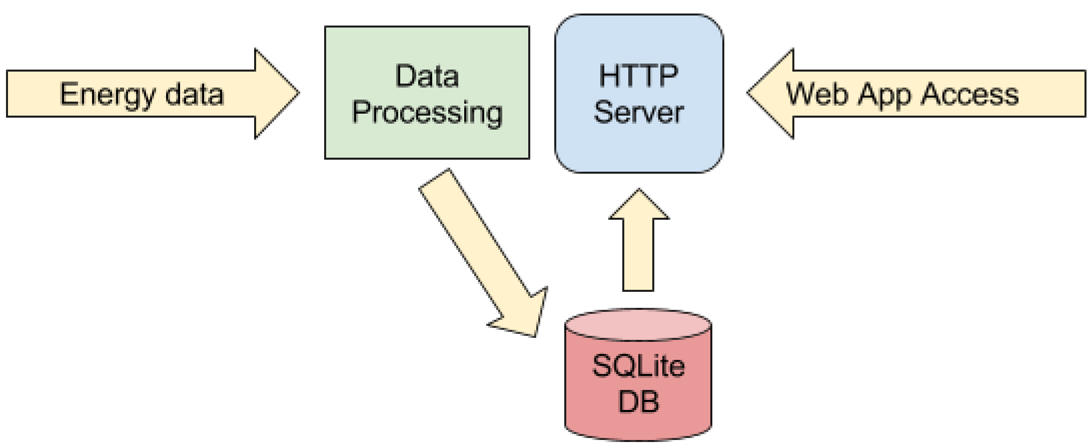
On the software side, most systems use a web application or a mobile application to present the user with their energy usage. After considering the advantages of several different approaches, we decided to focus our energy into a web-based application. This removes any OS-dependency, as a web application 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).

3.2 Proposed System Block diagram

Hardware Block Diagram



Central Server (Network) Block Diagram



3.3 Assessment of Proposed methods

3.3.1 Hardware Assessment of Proposed methods

The hardware portion of this project refers to the safe and accurate measurement of current & voltage. We have proposed a way of doing so that involves using transimpedance amplifier that is attached to the high side of the circuit. The current arrives at the amplifies (after going through protection elements such as a fuse, and a user controlled transistor switch) and through the Hall Effect is able to produce a proportional voltage at the output. The current then flows back into the mains line where it will travel to the load and complete the circuit. We chose to use high-side measurement due to the popularity of the method amongst similar devices. This popularity is due to the fact that if we were to measure current on the low side; we can unstabilize the ground point causing performance issues with the load. However the high side has cons as well.

Due to the fact that we are measuring power directly before the load receives it; we have to take precautions to achieve minimal power consumption. The fuse has a relatively small impedance; therefore, that component is not a grave concern. Similarly, the transimpedance amplifier has a resistance on the order of 0.1 milliohms. The major relative impedance will come from the transistor switch and the surge protection capacitors. That being said, the capacitors are an issue for a petite time window (only to handle start-up and stop transients). The remaining issue of the transistor is taken care of by choosing the switch we did which controls up to 15A with minimal power consumption. All these factors combined insure that we expect brown-out cases to not be an issue.

Moving to the output of the transimpedance amplifier. We expect to receive a proportional voltage with respect to the current. However, we will choose to pass that output through another non-inverting gain amplifier to better our sensitivity capabilities. Once the ideal sensitivity is reached, the voltage can pass into the microcontroller where the software team processes the data to obtain average power and the necessary inbetween steps to transmit the data back to the user.

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.

3.4 Validation

3.4.1 Network Validation

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 Validation

For hardware validation, we will need to construct a test circuit and operate under close to ideal conditions. This will most likely entail room temperature measurements using the lab bench.

The first hurdle includes validating that our start-up and start transients are controlled. We will take the switch and the respective protection circuits on either side of the switch and isolate them. Once that is done we can simulate a start-up and stop power transients by modulating the signal generator. However, we will not be able to get up to our max rated current of 15A. Therefore, we will have to theoretically approximate the transients at lower currents and experimentally verify. If the verifications are validated, we will test multiple points as close as possible to the our rated values. We will be able to develop a relationship and predict how it will behave.

Next, we will need to validate the current-to-voltage sensitivity on the transimpedance amplifier. Again we can do this with a function generator and cross check outputs obtained on the oscilloscope with that of the expected output. Here we will need to see how much noise will be an issue. The input should be rather clean as the local utilities tend to do a good job at providing a clean sine wave. The noise will come from the amplifier internals. If there is an issue, we will have to look into using a peak detector to eliminate noise and provide a clean input to the microprocessor.

Lastly, we will have to verify that our measurement circuit allows for optimal power transfer to the load. The process is similar to the ones prior. Test the circuit at multiple points with a common household load model, and verify that the load still receives the necessary voltage and current to operate.

4 Project Requirements/Specifications

4.1 FUNCTIONAL

4.1.1 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

4.1.2 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

4.1.3 Hardware System Requirements

1. The hardware shall not be a source of significant power drain (power consumption less than 2W).
2. The current sensor must be able to measure bidirectional current from -20A to 20A.
3. The hardware shall be sensitive to 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.

4.2 Non Functional Requirements

4.2.2 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.

4.2.2 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.

5 Challenges

Three biggest challenges that we will be facing when implementing the hardware part of the energy measurement device are:

Power Consumption:

Low-power consumption is a common characteristic of the many electronic devices these days. Without a doubt, engineers have been taken this challenge seriously. In our project, however, building energy measurement is much more challenging because we want to reduce the power consumption to the lowest level as possible. We want to minimize so that the overall energy consumption of the device must be lowered than 2W. This number is based on the lowest power that is being consumed by a typical phone charger. To achieve this specification, we have to be very careful with passive components like resistors, microcontroller, and wireless communication. The obvious trade-off would be between the functionality and power consumption. 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.

Noise:

When dealing with measuring equipment, noise is the common issue. Noise mainly comes from components themselves. We will be using Hall-effect sensor to measure the AC current. The basic mechanism of this sensor is to sense the generated magnetic field around the transmission line, which then can be convert back to current value. It sounds very practical in theory, but in reality, the sensor also picks up noise and amplify it. In addition, we will be embedding wireless shield into the device. Consequently, this shield will interfere with the Hall-effect sensor and will cause fluctuation in the measurement. The challenge here is that we can do filtering, however, it will cost the device significant amount of power consumption. So we aim to research more about component shielding and apply it to the design so that interference will be minimized.

Accuracy:

The energy measurement that our team is designing intends to handle the current ranging from -15A to 15A. This is a relatively large current range. If we would transform current value to voltage value, it would be 4V/30A, which is approximately 133 mV/A. The ACS712, the current sensor that we will be using has the sensitivity of 100 mV/A. With a small sensitivity, the accuracy of the measurement relies pretty much on the Analog-to-Digital converter, in the sense that the ADC must be capable of converting relatively small decimal voltage. If we measure high current value, 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.

The two biggest challenges that we will be facing when implementing the network part of the energy measurement device are:

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.

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.

6 Timeline

See Appendix 2.

6.1 First Semester

The first semester will be spent mostly in the conception, research, and prototype phases as seen above. Once the conception was complete we split into two groups: hardware and software, with three members on each team. 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. However, the software team is reaching 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 order sample parts to test their behavior. If as expected, or close enough, we will develop a prototype by the end of the semester. That being said, the prototype will still be in testing phases until the following semester.

6.2 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.

7 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.

8 References

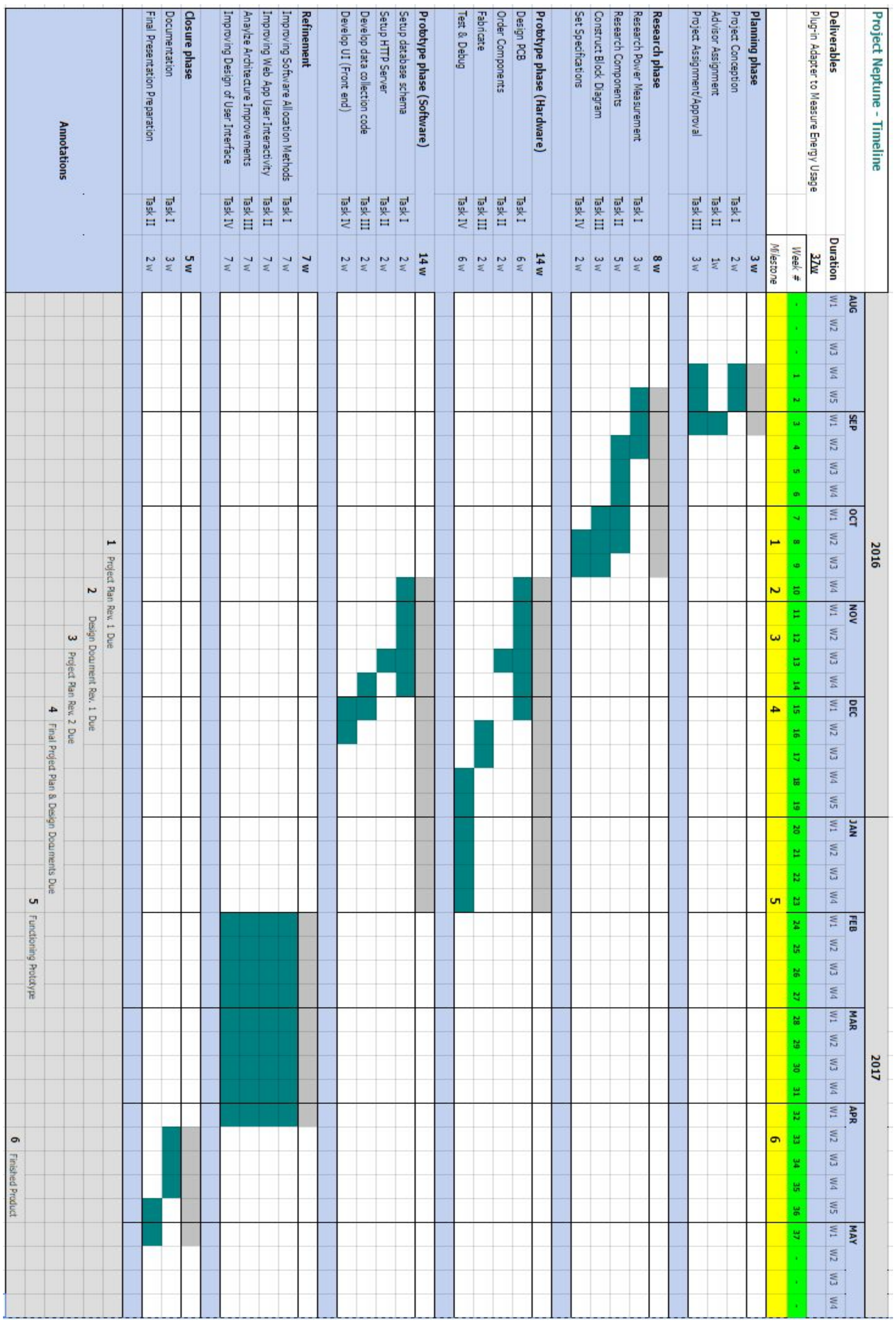
1. "The Energy Detective Electricity Monitor." *The Energy Detective Electricity Monitor*. TED, n.d. Web. 12 Oct. 2016. <http://www.theenergydetective.com/>.

9 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.

2.



Project Neptune - Timeline
Deliverables
Plug-in Adapter to Measure Energy Usage
Duration
37w
Week #
Milestone
2016
2017
AUG
SEP
OCT
NOV
DEC
JAN
FEB
MAR
APR
MAY
Planning phase
3 w
Project Conception
Task I
2 w
Advisor Assignment
Task II
1w
Project Assignment/Approval
Task III
3 w
Research phase
8 w
Research Power Measurement
Task I
3 w
Research Components
Task II
5 w
Construct Block Diagram
Task III
3 w
Set Specifications
Task IV
2 w
Prototype phase (Hardware)
14 w
Design PCB
Task I
6 w
Order Components
Task II
2 w
Fabricate
Task III
2 w
Test & Debug
Task IV
6 w
Prototype phase (Software)
14 w
Setup database schema
Task I
2 w
Setup HTTP Server
Task II
2 w
Develop data collection code
Task III
2 w
Develop UI (Front end)
Task IV
2 w
Refinement
7 w
Improving Software Allocation Methods
Task I
7 w
Improving Web App User Interactivity
Task II
7 w
Analyze Architecture Improvements
Task III
7 w
Improving Design of User Interface
Task IV
7 w
Closure phase
5 w
Documentation
Task I
3 w
Final Presentation Preparation
Task II
2 w
Annotations
1 Project Plan Rev. 1 Due
2 Design Document Rev. 1 Due
3 Project Plan Rev. 2 Due
4 Final Project Plan & Design Documents Due
5 Functioning Prototype
6 Finished Product