# Outlet-Level Power Monitoring and Point-of-Use Feedback for the Reduction of Power Consumption

Benjamin Hamner
Advisor: Dr. John Board

ECE 193 Independent Study
Smart Home Fellows Project, Fall 2010
Duke University

## 1 Abstract
Energy efficiency is currently a hot topic: part of the $787 billion dollar stimulus package included $38 billion of energy efficiency spending and $20 billion of tax incentives in the coming decade, and efficient use of energy is key to saving money and non-renewable resources devoted to energy production. However, consumers do not have the information they need to optimally reduce their power consumption: the vast majority only see their power usage on the monthly bill. This removes behaviors, such as leaving the heat on while away or leaving a computer on, from their eventual costs. This project focused on analyzing and prototyping potential methods to meter power usage on the device level and provide real-time and historical feedback on where a consumer's energy dollars are spent. Modified Kill-a-Watts were constructed and a custom prototype unit was designed to meter power outlets. Matlab and python scripts were written to calculate and record power usage received wirelessly from the outlets and upload this information to a server. A web-server with a PHP/MySQL backend was created to store this data and display it to the user. It was concluded that a custom-built energy metering outlet would be most suitable for device-level power metering and point-of-use feedback in the long term, and there is much room for future work in all areas of this field, especially with analyzing and visualizing energy usage data to optimally modify consumer behaviors.

## 2 Introduction
During 2009, US consumers used 1380 million MWh of electricity, or 35.3% of the overall electricity market, at a cost of 155.4 billion dollars. Since there were 125 million households in the US at the time, this comes down to an average electricity cost of approximately $1250 / household / year.[1] Additionally, over 90% of this energy came from non-renewable sources, and much of that originated from countries outside the United States. Thus there are strong economic incentives for citizens, along with political and environmental incentives for the nation, to reduce and optimize residential energy usage.

However, as energy is billed once a month, and the vast majority of consumers in the US do not see how much energy they have been using until the monthly bill comes, it is difficult for consumers to optimize their energy consumption beyond pointers such as "turn the lights off when you leave a room." The Energy Star program, which was started by the US government in 1992 to reduce energy consumption and greenhouse gas emissions, is an international standard for energy-efficient consumer products. According to the program's website, it helped save nearly 17 billion dollars on US consumers' energy bills in 2009.[2] While this helps, it still does not give residents the feedback required to make them aware how much each device is costing them in their electricity bills, and how they save energy and money via slight behavioral modifications.

There are a number of niche products on the market that allow customers to monitor their whole-house energy usage in real time. For example, the Wattson[3] and the Onzo[4] are two systems that include a sensor to monitor the energy usage via the lines feeding into the utility power meter, and a base station to display this in terms of dollars and power. While these systems enable consumers to determine when they are using more power than normal and provide real-time feedback for the change in energy usage, they do not inform consumers which of their devices and their behaviors are using most of their energy and dollars.

Electricity in the US is delivered to homes as alternating current with a root mean square (RMS) voltage of 120V and 240V, with most consumer appliances using 120V, and some systems, such as the HVAC system, typically using 240V. Power in the US is delivered at 60 Hz, which means the voltage waveform is a sinusoidal waveform with a RMS magnitude of 120 volts (an amplitude of approximately 170 V) that oscillates at a frequency of 60 Hz. When a load is connected to this supply, current can be drawn in different fashions. The most basic case is for a load with a purely resistive heating element, such as an incandescent light bulb. In this case, the current waveform is also sinusoidal and at 60 Hz, and it's amplitude is proportional to the resistive heating element by Ohm's law. Average real power is calculated by numerically integrating the instantaneous voltage and current in this and all cases, and, in this case, it is equal to the apparent power, which is the product of the RMS voltage and current. Some devices, such as refrigerators, may have a large inductive load, which offsets the phase of the current waveform from the voltage waveform. This decreases the real power relative to the apparent power, and the ratio between the two is termed the power factor. Another type of power supply is a switched power supply, which only draws power when magnitude of the voltage is above a certain threshold. This also decreases the power factor. It costs more and is less efficient to deliver energy to devices with lower power factors.

This project focused on analyzing and prototyping potential methods for metering power usage at the outlet level, along with creating a system to record and display the real-time and historical power usage to the consumer. The long-term vision of this project was to provide consumers with the information they need in a simple and effective manner in order to make behavioral and purchasing decision to reduce their energy costs and optimize their energy consumption.

## 3 Outlet Level Power-Metering

### 3.1 Tweet-a-Watt

The first possibility analyzed for wireless power metering at the outlet level was a modified Kill-a-Watt[5], a device used to display instantaneous power usage on an outlet, called a Tweet-a-Watt[6] (see Figure 1). Tweet-a-Watts could be constructed for $60 (see Table 1) and two hours of labor per device, and consisted of a Xbee wireless chip plus analog and digital circuit components soldered inside of the Kill-a-Watt. 10 Kill-a-Watts had previously been constructed by another student and were available for testing.


**Figure 1**. Kill-a-Watt modified into a Tweet-a-Watt. The LED at the top flashes each time the Tweet-a-Watt transmits a data packet.

**Table 1.** Cost of Tweet-a-Watt Components

| Component | Cost |
|---|---|
| Kill-a-Watt | ~$20.00 |
| Xbee module | $23.00 |
| Xbee adaptor | $10.00 |
| Analog circuit elements | $7.00 |
| **Total** | **$60.00** |

Each Tweet-a-Watt broadcast the instantaneous power consumption every two seconds to a base station, which calculated and displayed the Watts used and logged the average power usage every five minutes. The base station consisted of a laptop connected to an Xbee chip via a USB-to-Serial cable. Data was initially recorded via a modified version of the Python script available from the Tweet-a-Watt website. Later, a Matlab script was written to perform similar functions while enabling more complex real-time analysis of the data.

The first test performed on the Tweet-a-Watts involved measuring the power consumption of a known load, a 40W incandescent light bulb, with a two Tweet-a-Watts. An unmodified Kill-a-Watt verified that the 40W bulb actually drew 40W. As the data was coming in, it was noted that the individual 2-second power measurements varied by approximately 50%, from 20W-60W. For this test, Tweet-a-Watt #4 was connected to wall power, Tweet-a-Watt #2 was plugged into Tweet-a-Watt #4, and the light bulb was plugged into Tweet-a-Watt #2. The average power usage (of 150 power samples every two seconds) was recorded every five minutes for 3.5 hours. The results are shown in Figure 2.

The mean power usage as measured by Tweet-a-Watt #2 was 42.7W, and the mean power usage as measured by Tweet-a-Watt #4 was 49.5 W. The standard deviations of the 5 minute readings were 1.59 W and 1.92 W, respectively, and the ranges were 38.7 W – 45.6 W and 45.7 W – 54.6 W, respectively.

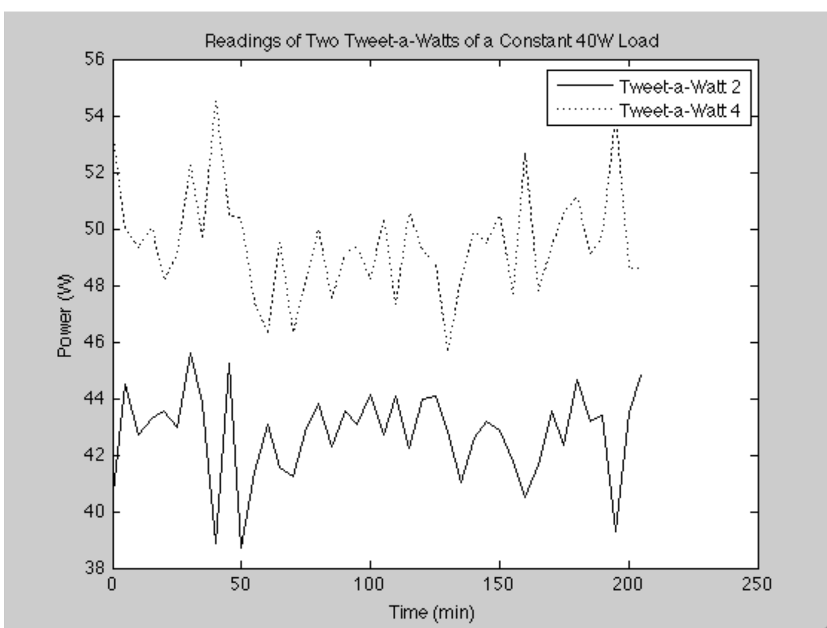


**Figure 2.** Power used by a 40W load as measured by 2 Tweet-a-Watts.

Next, the zero-load power usage was measured by a Tweet-a-Watt over a 50-hour period, with average readings recorded every five minutes. The Tweet-a-Watt was calibrated before the trial. The results are shown in Figure 3. The mean was 3.4 W, with a standard deviation of 1.3 W and a range from 0.8 W – 5.8 W.

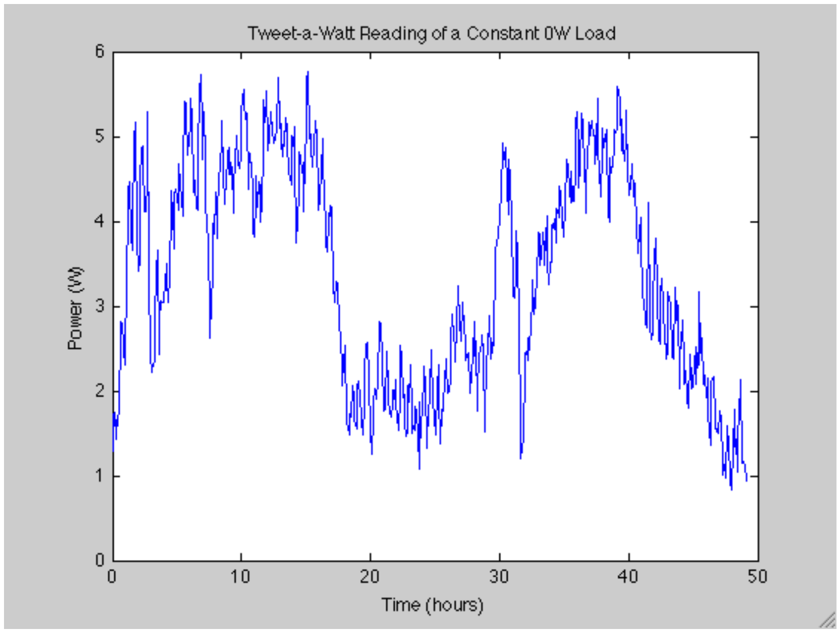


**Figure 3.** Power Usage of a 0W load as measured by 2 Tweet-a-Watts

Next, Tweet-a-Watt #9 and Tweet-a-Watt #10 were connected to a load with a switched power supply, which only drew current when the voltage was approximately at a maximum +170 V or minimum -170 V. An unmodified Kill-a-Watt measured the load of this to be constantly 40 W, with a power factor of 0.61. Figure 4 shows the average power readings, calculated every 5 minutes. The mean power measured by Tweet-a-Watt #9 and Tweet-a-Watt #10 was 61.3 W and 53.6 W, respectively. The standard deviations were 2.30 W and 1.95 W, and the ranges were 48.4 W-68.4 W, and 46.1 W – 61.1 W.

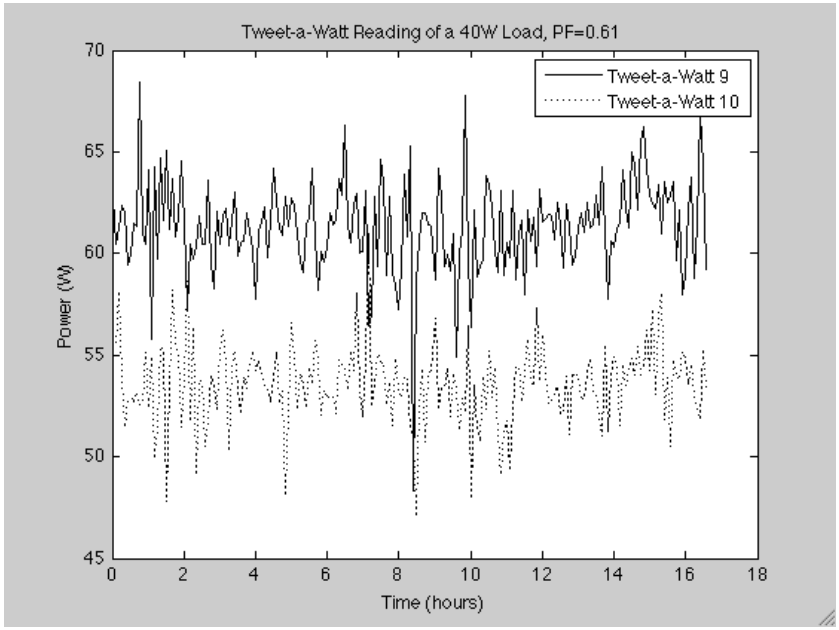


**Figure 4**. Power reading of 40 W switched power supply with a power factor of 0.61 from two Tweet-a-Watts.

For a last test of the Tweet-a-Watt's accuracy, Tweet-a-Watt #9 and Tweet-a-Watt #10 were connected to a power strip containing several computers and oscilloscopes, and a light. The computers were left at a baseline processing level, and set not to go to a screensaver and not to go to sleep. A Kill-a-Watt measured this load as varying between 350 W and 351 W, with a power factor of 0.82. The plot of the power, as measured by the Tweet-a-Watts for 6 hours for each 5 minute interval is shown in Figure 5. The mean power, as measured by Tweet-a-Watt #9 and Tweet-a-Watt #10, was 442.6 W and 401.1 W, respectively. The standard deviations were 9.9 W and 10.2 W, and the ranges were 419.4 W – 476 W, and 378,4 W – 439.1 W.

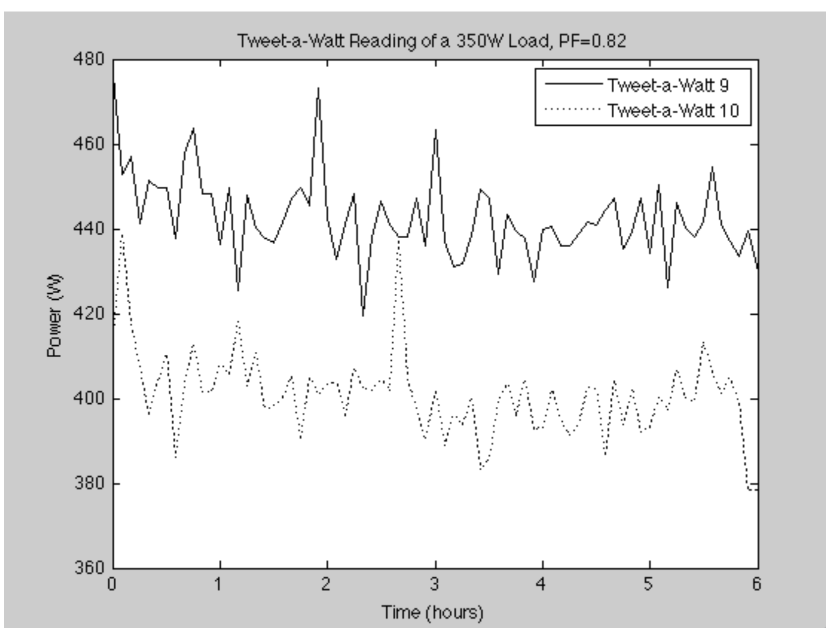


**Figure 5**. Power reading of 350 W load with a power factor of 0.82 from two Tweet-a-Watts.

Thus, the Tweet-a-Watts were able to give a rough order-of-magnitude estimate of the power draw of a connected device (10s of Watts vs. 100s of Watts), but not much more. All data shown here is for the average power used over a 5-minute interval, which consists of the average of 150 power measurements taken every 2 seconds. This begged the question of why the Tweet-a-Watt power measurements showed so much variation and were so far off, when the Kill-a-Watt measurements were consistent with each other and showed no variation.

This large error was due to the manner in which the Tweet-a-Watt took its data. Instead of interfacing with the Kill-a-Watt's own power calculations, the Xbee chip took 17 digital samples each of the voltage and current data. Each sample was taken 1ms apart over a 17ms time period, and digitized to a 10 bit value from 0-1023. The Xbee chip then transmitted these 17 voltage and 17 current samples, along with an identifier for the chip, to the base station, slept for two seconds, and repeated this. This 17ms period captured approximately one power cycle of the device, which occurs at 60 Hz, or about one every 16.7ms. The Xbee then slept through approximately 119 power cycles before recording another. When this data packet was received by the base station, the voltage was scaled to be 120 V RMS, the baseline was removed from the current data by a manually-calibrated constant, and then converted to amps via multiplying by another constant. The power used by the load in Watts was then calculated by numerically integrating these two sets of samples.

To improve the accuracy of the power measurements, first potential improvements to the Xbee's protocol were analyzed. Data could not be sampled by one analog port at a rate higher than 1 kHz, so that was already at its limit for the voltage and current samples. It also did not have the memory to send 34 samples each of voltage and current data back to the base station, so two power cycles could not be measured instead of one. The sleep time was reduced on the Xbee to below 2 seconds, but that prevented the capacitor used to power the Xbee's radio transmission from charging up between transmissions, making the data collection erratic. Thus, it was concluded that the Xbee parameters were set optimally, and the Tweet-a-Watt system was unable to take sufficiently accurate measurements of the real-time, outlet-level power usage.

### 3.2 Modified Tweet-a-Watt

Since the Tweet-a-Watt unable to sample and transmit raw voltage and current information fast enough to accurately calculate the power usage, additional modifications of the Kill-a-Watt were explored. A PIC18F1320 microcontroller was connected to the scaled voltage and current data output from the Kill-a-Watt, and then connected via its Universal Asynchronous Receiver/Transmitter (UART) port to the Xbee. Figure 6 shows the modified Tweet-a-Watt, with the Xbee and PIC microcontroller on a breadboard.

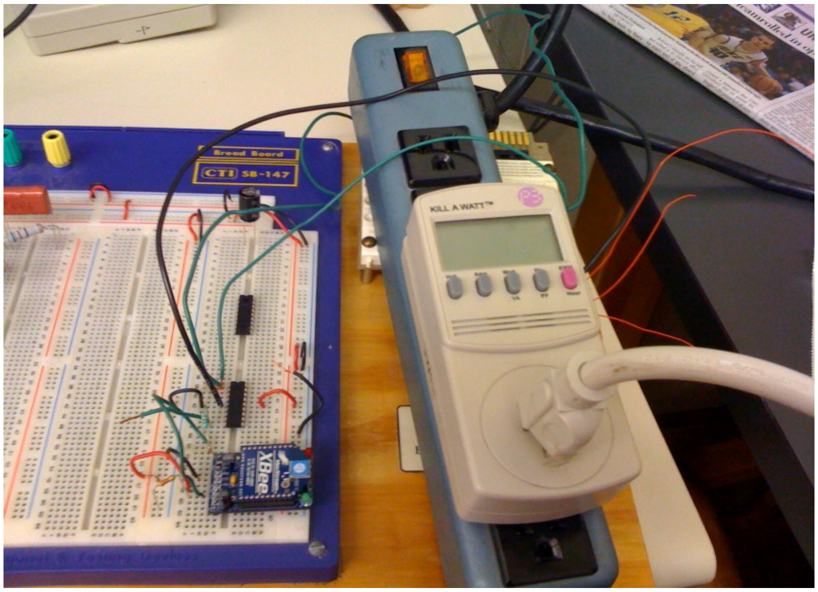


**Figure 6**. Tweet-a-Watt modified to have a hire sampling rate and perform the power calculations at the device level instead of at the base station.

The PIC18F1320 was then programmed using assembly language to do what the Xbee could not - sample the raw voltage and current data from the Kill-a-Watt at a high rate, and subsequently calculate the current power usage from this information. To do this, the PIC's internal clock was set as high as possible – to 8 MHz, which allowed the assembly program to utilize 2 million instructions / second. The raw voltage and current data from the Kill-a-Watt was connected to 2 of the analog input pins (RA0 and RA1) on the PIC. The pick then alternately sampled these pins at the maximum rate that it was possible to achieve high precision on the 10-bit A/D converter, and, at the same time, perform the necessary calculations with the data. The timing was corrected so that the PIC took precisely 3000 voltage samples and 3000 current samples every second, and it executed precisely two million instructions each second, with the precision of this limited only by the accuracy of the internal clock. This means that 50 voltage samples and 50 current samples are taken for each of the 60 power cycles that occurs each second.

After the first current and voltage samples are taken, the PIC stores these and initiates the A/D conversion for the next voltage sample. While it is taking this A/D conversion, it converts the unsigned voltage and current samples, which range from 0-1023, to signed 2-byte integers by subtracting the estimated mean value of the channel. The initial mean value of the channel is estimated to be 512. During the course of each power cycle, the PIC stores the maximum and minimum voltage and current, and estimates the new mean by averaging the each maximum and minimum, and then averaging this with the previous baseline estimate of the channel. This assumes that the mean voltage and mean current on each channel is zero, which was true for all devices tested. This also allows the PIC to rapidly converge on the mean of the raw channels, while adjusting for any DC offset present in the raw voltage and current samples and limiting the impact of noise on the estimated channel offset.

Once the PIC converts the voltage and current into 16-bit signed integers, it uses a 16-bit signed multiply routine to calculate the instantaneous power. It then saves this value, waits for the A/D conversion of the voltage sample to complete, and starts the A/D conversion of the current sample.

While the current measurement is being made, the PIC adds the previously calculated instantaneous power to a 6-byte variable representing the sum of the instantaneous power measurements since the last transmission. It then calculates the square of each of the previous voltage and current samples, and adds these to two variables representing the RMS voltage and current. It then waits for the current measurement to complete, and repeats these steps.

After each power cycle (after the PIC takes 50 voltage and 50 current samples), the new DC offsets for the voltage and current channels are calculated based on the maximum and minimum of each channel and the previous value. After data is taken for 120 power cycles (2 seconds), the PIC transmits the data over the serial transmission line to the Xbee, which then transmits it to the base station. The data transmitted to the Xbee is 6-bytes of power data, 2-bytes for the DC-offset of the voltage channel, 2 bytes for the DC-offset of the current channel, 5-bytes for the RMS voltage, and 5-bytes for the RMS current. These transmissions are made while A/D conversions of voltage and current samples are going on, so that they do not interfere with the timing of the data collection.

The Xbee was programmed to wake up when it received a transmission on the serial line, transmit the data to the base station, and then go back to sleep. On the PC side, first a Python script, and then a Matlab script, were written to receive and display the data. These scripts functioned by taking in the scaled RMS voltage, RMS current, and power values, multiplying them by experimentally determined constants to convert them into volts, amps, and Watts, respectively.

After calibration, additional measurements were taken for the 40W and 70W incandescent light bulbs with the modified Tweet-a-Watt. Each of the 2-second power measurements fell within one watt of the actual power usage, as shown by a Kill-a-Watt. The majority of these measurements agreed to within 0.2 W of each other.

### 3.3 Custom-Rolled Solution

Since a custom circuit board would be necessary to safely use a modified Tweet-a-Watt in a consumption environment, the feasibility of building an outlet-level power meter from scratch was analyzed as well. Advantages of this approach would be the production of a extensible platform for outlet-level power metering and point-of-use feedback, which would have potential commercial applications. Disadvantages included the additional time and cost of product design and development, along with the necessity of working directly with high voltages from wall outlets.

For a starting point, the design for a device used in a similar project at UC-Berkeley, the Green Soda project, was analyzed. Two types of outlet level power meters, ACme-A and ACme-B, were built over the course of this thesis project. The former used a shunt resistor to measure current being delivered to the device, along with direct rectification for the AC/DC power supply and a ADE7753 energy-metering IC to measure and calculate the power consumption. The ACme-B used a Hall-Effect sensor to measure the current conversion, a step-down transformer and a bridge rectifier as the power supply, and calculated the energy consumption using a microcontroller. Both devices used Berkeley's EPIC core as a microcontroller/radio transmitter.[7]

The features of the ACme-A were used to help design the analog front-end circuitry for this project, in order to convert the 120 V AC into 5 V DC to power the device and step the voltage down with a 600:1 voltage divider in order to measure it. A shunt resistor was used in order to measure the device's power consumption. Two options were built into the circuit for measuring the power: one used the ADE7753 energy-metering IC to sample and numerically integrate the voltage and current to calculate power. The other involved using the PIC18F1320 microcontroller to sample and calculate the power, in the same manner as in the modified Tweet-a-Watt. Table 2 shows the components used in the custom-rolled solution, along with price estimates for prototyping and manufacture. Figure 7 shows the schematic layout of the circuit, and Figure 8 shows the PCB layout. Each of these designs were made with the EAGLE layout editor. Figure 9 shows the 3D rendering of the PCB.

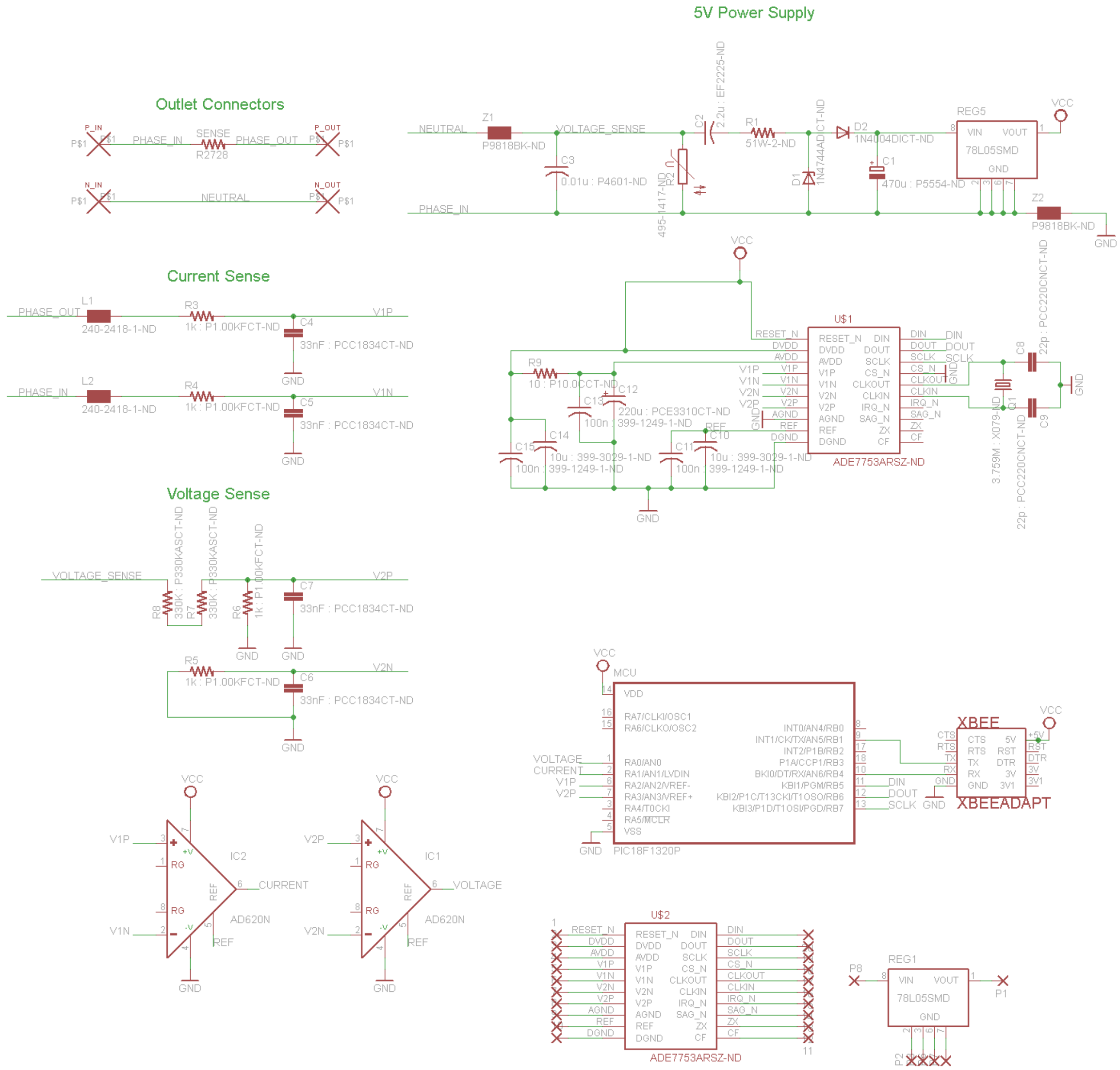


**Figure 7.** This figure shows the schematic for the custom-rolled PCB for an outlet-level power meter. The 5V power supply uses direct rectification to provide 5 V DC power from 120 V AC power, and is based on UC Berkeley's ACme-A design. The current sense uses a shunt resistor to convert the current being delivered to the load to a voltage while drawing little power itself. The voltage sensor consists of a voltage divider to step down the voltage to a level at which it can be read by the ADE7753 power metering IC. The op-amps were included to step up the voltage range to one that could be read in at high precision by the PIC microcontroller and correct for the negative values of the voltage and current relative to ground. The PIC

microcontroller is connected to raw the voltage and current data, the ADE7753 power-metering chip, and the Xbee in order to be able to broadcast one or both power measurements.

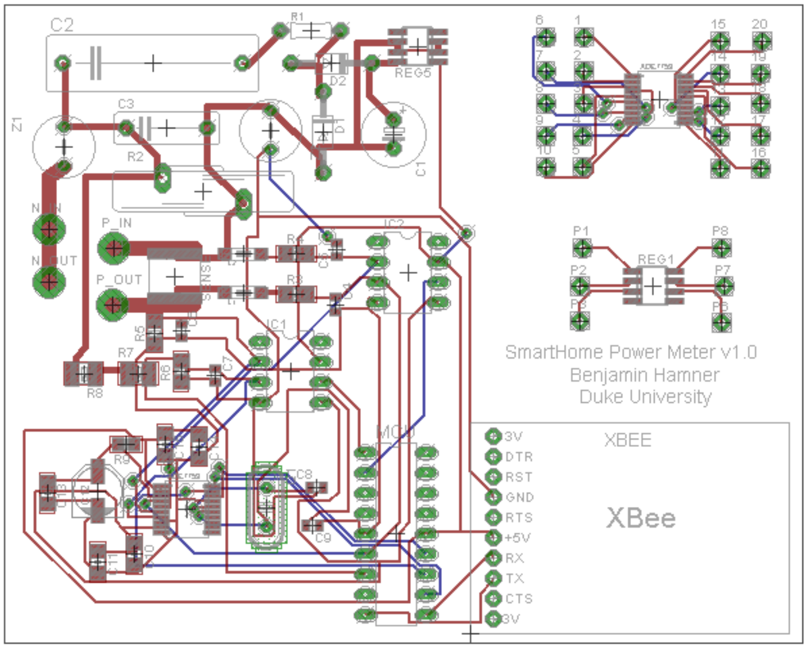


**Figure 8**. This figure shows the custom PCB layout. There are an additional ADE7753 and voltage regulator in the top-right corner so that these sections could be cut off the PCB and used to put a surface-mount package on a breadboard for testing.

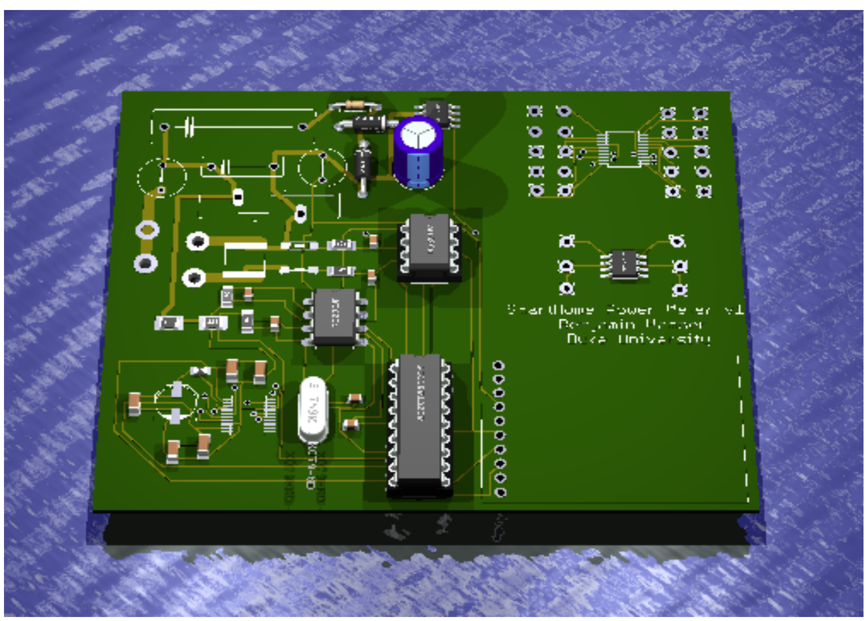

**Figure 9.** 3-D rendering of the custom PCB layout along with most components.

A few sub-circuits of the PCB were tested before the PCB was ordered. The relay circuit successfully cut a light on and off by controlling the port on the PIC microcontroller, which could be done remotely via the XBee. The circuit connecting the microcontroller and the Xbee functioned well for sending and receiving data. The performance of the shunt resistor was verified when it was connected between a 40W bulb and the outlet power supply using an oscilloscope: the sinusoidal voltage waveform was proportional to the current flowing through the resistor. However, as two of the key components, the ADE7753 power metering IC and the 5-volt regulator were surface-mount components with small pins, and their performance could not be easily evaluated without the PCB.

Unfortunately, by the time the PCB was designed and ordered, there were only several weeks left of the semester and the project. The PCB did not arrive until two weeks after the order was placed, during the middle of exams. The first attempt to solder components on the circuit had everything successfully mounted and connected, as verified with a multimeter, with the exception of the ADE7753, which had pins that were exceptionally small and close together. Bits of solder connected several of the pins in a single circuit, preventing its operation. This was chip was removed and the re-soldering process was started. It was left in the microwave a little too long due to a badly-timed phone call, and the chip burned, preventing a full test of the system. See Figure 10 for the burned chip.

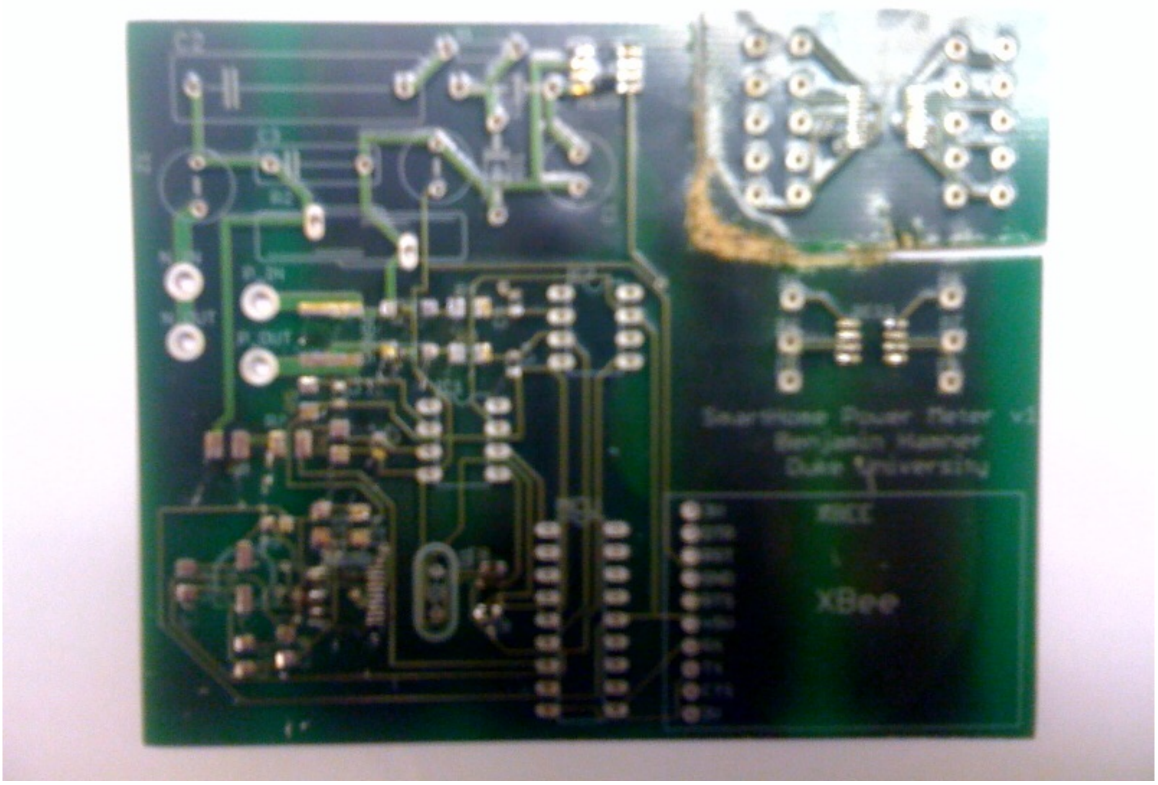


**Figure 10**. Burned PCB chip.

## 4 Data Logging and Display

Methods for logging and displaying the data were also analyzed, separately from the mechanism for collecting outlet-level power data. To log the incoming data, a server-based backend was written with PHP and MySQL. The basic MySQL database was designed with four tables: metercategories, meterrelations, meters, and powerdata. The table metercategories was for differentiating which categories, such as entertainment and lighting, associated meters fell under, and consisted of a category name and id. The table meterrelations was for establishing potential relations among power meters, and consisted of three fields: an id, a parent field, and a child field. For example, a whole-house meter would have many entries in this field, for each of the child meters directly under it. A meter connected to a power strip could have several entries, if devices connected to the power strip also had their own meters. This table provides a simple method to organize the relationship between power meters, and helps prevent data from being double-counted.

The table meters contained a list of all the meters that could be connected to the system. It had an id field, a name for each meter to specify what it was connected to (for example, "Desk Lamp" or "Televesion"), and a category field, linked to the category table via category id. This table formed the link between meters and devices, and devices and categories. The table powerdata contained instantaneous power readings (typically uploaded at one-minute intervals, and averaged over the entire minute). It included fields for an id, the timestamp at which the reading was taken, the id of the meter that took the measurement, and the power measurement.

These MySQL tables were manipulated via PHP scripts. Installdb.php read the specified table structure, as outlined above, from the tablestructure.php file, which was designed to allow the simple addition and modification of tables, and created the tables with the necessary properties. Db-logon.php contained the database login information, and was included by every script that accessed or modified data within the database. Renewtable.php deleted the data with the associated table (specified as renewtable.php?table=table_id) and updated the data structure of the table with that specified in the tablestructure.php file. Removedb.php removed all the tables in the powermeter database, along with the powermeter database itself, allowing the system to be cleanly wiped and reinstalled. Addpowerdata.php was used for adding a single power datapoint to the database in the cloud: a PHP or PERL script on a local machine could call the URL /powermeter/admin/addpowerdata.php?meter=meter_id&timestamp=datetime&power=powerdata to automatically upload the latest power data point to the server and save it in the database. Addrandompowerdata.php permitted random power data to be added to the database, allowing the site to be tested. Installedtablestructure displayed the table structure currently present in the database in HTML format, as shown in Figure 11.

**meters**

| | |
|---|---|
| id | INT NOT NULL auto_increment |
| name | VARCHAR(100) UNIQUE NOT NULL |
| category | INT NOT NULL |

**powerdata**

| | |
|---|---|
| id | INT NOT NULL auto_increment |
| meter | INT NOT NULL |
| time | DATETIME NOT NULL |
| power | DOUBLE NOT NULL |

**meterrelations**

| | |
|---|---|
| id | INT NOT NULL auto_increment |
| parent | INT NOT NULL |
| child | INT NOT NULL |

**metercategories**

| | |
|---|---|
| id | INT NOT NULL auto_increment |
| name | INT NOT NULL |

**Figure 11.** Table structure for power meter database, as shown by installedtablestructure.php.

The script edittable.php, run as edittable.php?table=tablename, was written to allow for the simple addition or modification of data within any tables over the web. The table was displayed in an HTML table, and any cell could be modified simply by clicking the contents of the associated cell and then clicking "save changes." A sample of this form, for the meters table, is shown in Figure 12.

| id | name | category |
|---|---|---|
| Save Changes | | Add Row |
| 1 | CIEMAS 1 | 2 |
| 2 | CIEMAS 2 | 2 |
| 3 | Power Strip 2 | 1 |
| 4 | Tweet A Watt #4 | 1 |
| 5 | MAC Laptop | 1 |
| 6 | Main Power Strip | 1 |
| 7 | TV | 1 |
| 8 | Dell Laptop | 0 |
| 10 | | 0 |

**Figure 12**. Simple way to add/edit data to table with an HTML interface in editable.php.

The main script, index.php, provided a basic web interface for displaying web history data. It pulled the power usage data for the selected power meter, and then used the Google

Annotated Timeline API to graph the data for the selected device. This interface is shown in Figure 13.

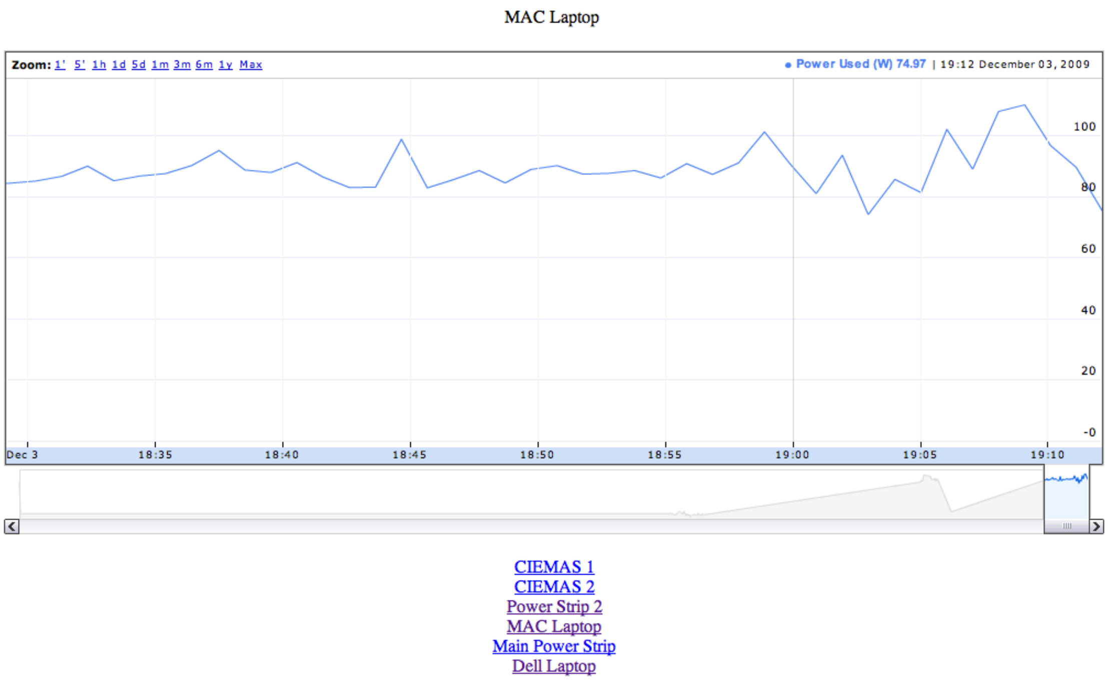


**Figure 13**. Basic web interface for displaying power meter history data.

### 5 CIEMAS power data

Since safety concerns preventing the use of the Tweet-a-Watts for long-term data collection, and the custom-designed devices would be unable to be designed to meet safety standards and produced to collect data within the time allotted for the project, power data from the CIEMAS buildings was gathered and analyzed over a four-month time interval. A script was written to poll data from the two power meters in CIEMAS every minute while the laptop was on, and this script was allowed to run from December 2009 to March 2010.

The average power usage for each minute of the day, over the course of the entire day, for the three month period is shown in Figure 14 for the two meters and in aggregate. One key points are that the daily power usage peaks between 10am and 4pm, when people are most likely to be in the buildings. The second is that the power usage generally varies over a relatively small range over the course of the day. The mean variation was from a minimum of 1300W at night to a maximum of 1400W during the day, or less than 10% of the baseline power consumption of 1300W. This means that the majority of the power consumed by CIEMAS is via devices and loads that are always on, as opposed to ones that are used marginally by students, faculty, and staff over the course of a day. However, this data does not help identify which devices and categories this power consumption falls under, or point to efficient ways to reduce it.

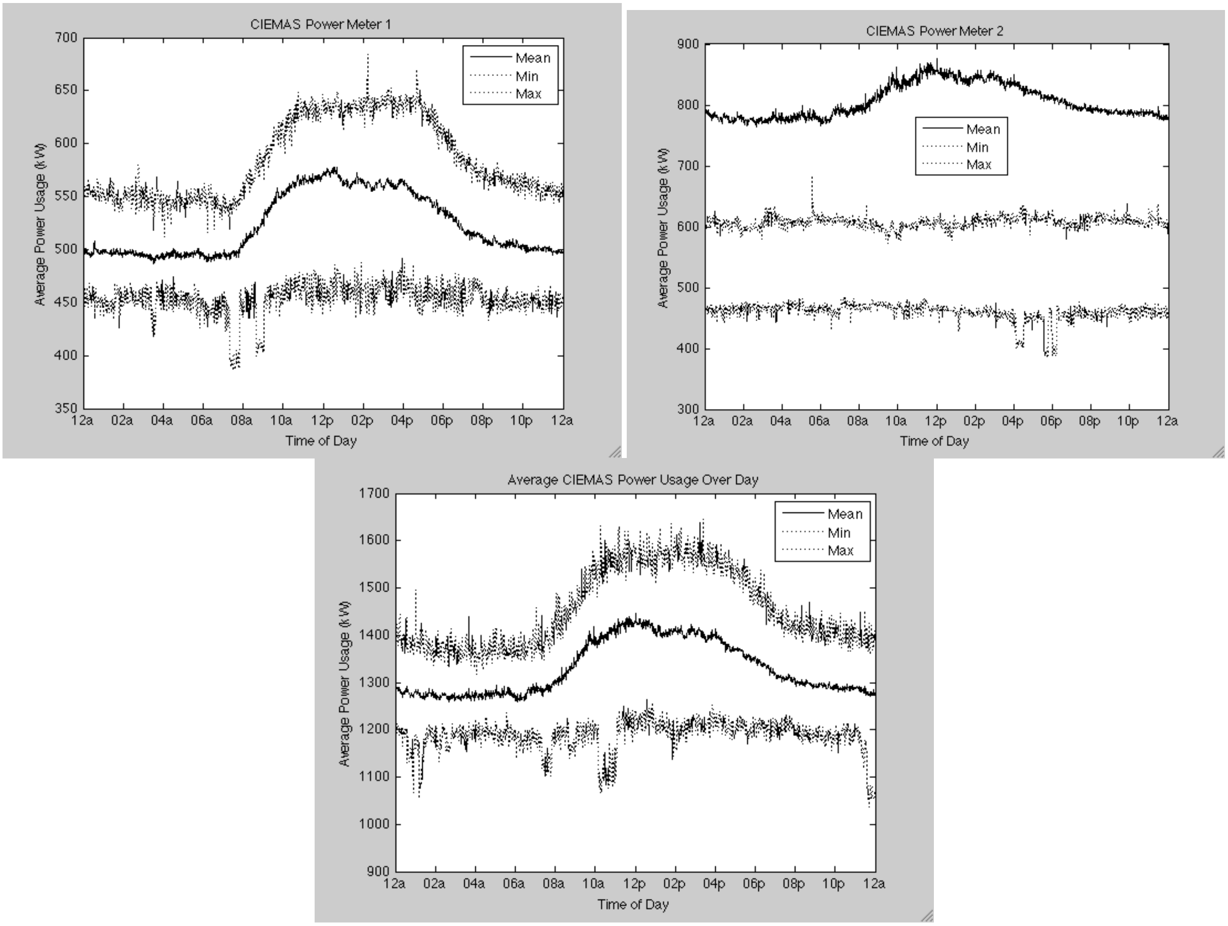


**Figure 14.** CIEMAS power usage over the course of an average day.

Next, the relationship between maximum daily temperature for Durham[8] and power mean power consumption between 10am and 5pm was analyzed. This data is shown in Figure 15. There is no correlation between the temperature and the power consumption for this data. This is not expected for typical buildings and residential homes, since heating and cooling utilizes energy. However, Duke uses central heating for many buildings from the steam plant, and thus the energy used to heat and cool CIEMAS was not reflected in the power meter data. There is significant information in this data: it is clustered into two groups, the larger of which has the mean daytime power consumption above 1350 kW, and the smaller of which has it below.

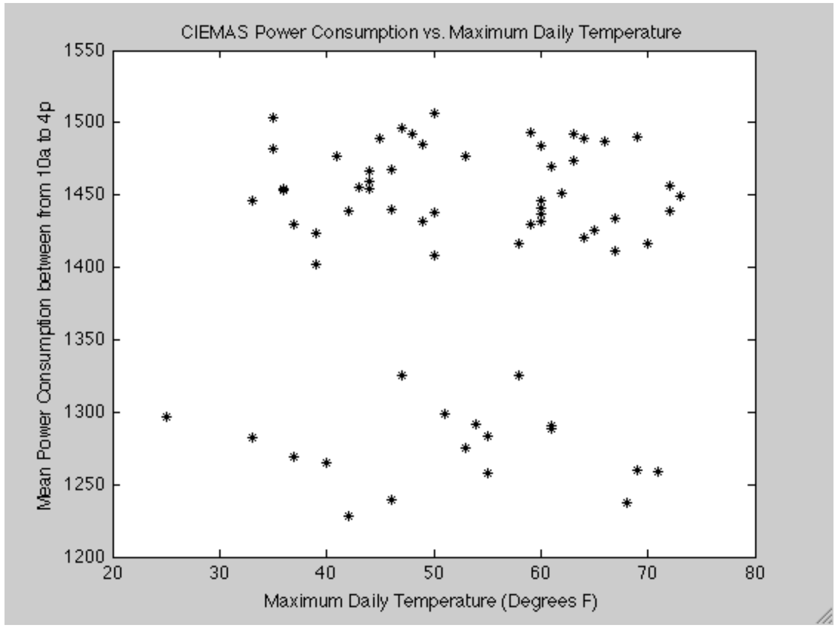


**Figure 15.** Mean CIEMAS power usage between 10am and 5pm vs. maximum daily temperature.

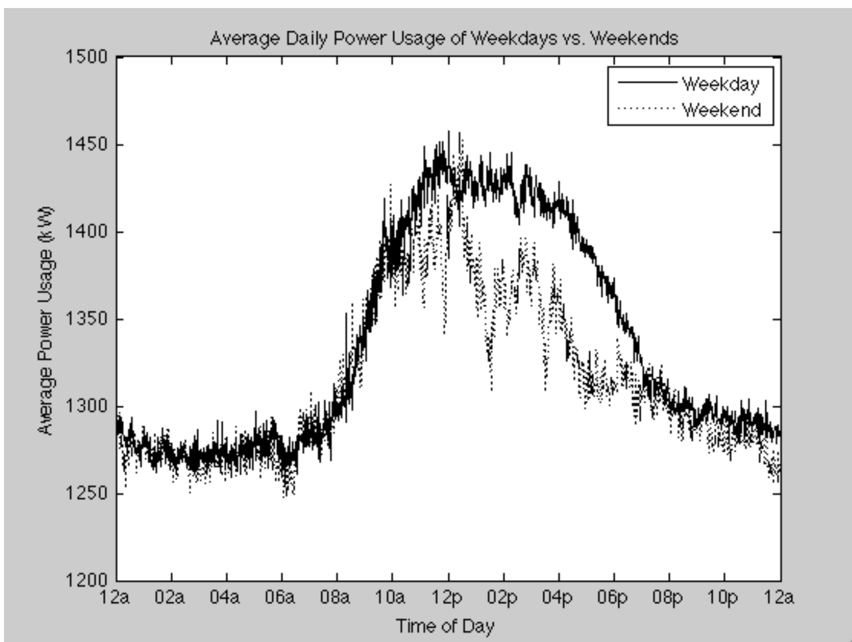


**Figure 16.** CIEMAS power usage for weekdays vs. weekends.

To determine the source of these two groupings in the data, the average temperate was analyzed as it varied over the course of the day for weekends and weekdays. This data is shown in Figure 16. For weekends, the daytime power usage was less than that for weekends. This indicates that the variation in power during the day was at least partially due to the marginal use of devices by humans during the day as opposed to at night, as opposed to the increase in temperature during the day or other causes.

## 6 Collaboration with Other Groups

Collaborative work was done with Token Energy, a student group started at NC State University that monitors the whole-building energy usage of two resident halls at NC State. This information is displayed to students in real time via TV screens around campus, and Token Energy has competitions to encourage residents to reduce their energy usage. Their areas of expertise are in the graphic design, marketing, and programming of the systems, and I provided a hard engineering take on their work and how to expand it in the future. Token Energy currently consists of students at the three major universities in the Triangle, Duke, UNC, and NC State, and is looking to expand these systems in each university in addition to other schools, with the goals of promoting energy awareness and saving money by reducing consumption. I helped facilitate discussions to bring this to Duke in the future, and one of their marketing teams analyzed the potential economic implications of my work for a class at NC State. Discussions were also held with multiple other groups on Duke's campus interested in device-level power metering.

## 7 Discussions

The three possibilities analyzed for continuously monitoring power usage at the outlet level analyzed were the Tweet-a-Watt, the modified Tweet-a-Watt, and the custom-rolled solution. The advantages of the Tweet-a-Watt were price and convenience: there were already 10 devices that had been built, and their use would not require additional design and development work. However, when it was noticed that the 2-second instantaneous power measurements reported by a Tweet-a-Watt for a 40-W bulb varied wildly from about 20 W to 60 W, this questioned the accuracy of the Tweet-a-Watts. Extensive testing of the Tweet-a-Watts demonstrated that all of them were unreliable for more than rough order-of-magnitude estimates of power usage, and that the error in the Tweet-a-Watt's measurements varied proportionally with power usage. Also, since the Tweet-a-Watt involved having loose solder and wires in the Kill-a-Watt, there were fire safety concerns.

The reason the Tweet-a-Watt failed to accurately measure power was that it only took 17 samples each of the voltage and current waveform over a single power cycle every two seconds. To fix this, a PIC microcontroller was programmed to sample the waveforms and calculate the power usage. This microcontroller took 6000 each of voltage and current samples over 120 power cycles in the same two seconds, allowing it to much more accurately determine the power being delivered to the load. The disadvantage of this approach, however, was that this modified system was not able to be contained within the Kill-a-Watt enclosure, at least without the creation of a custom PCB, and the possibility of more loose wires and solder hanging around created a fire hazard.

As a result, it was concluded that the best option for outlet-level power monitoring would be a completely custom-built device with the power supply, current and voltage sampling, power calculation, and wireless connection all on a single printed circuit board. While the prototype was not successfully built due to the burnt PCB, the lessons learned from modifying the Tweet-a-Watt, the functioning of the Kill-a-Watt, and the success of another similar project indicate that the design was solid and it would have functioned well. Proper certification of this device would handle the power safety concerns, and, additionally, this platform would be extensible and allow point-of-use feedback at the outlet level via LED lights and sounds.

A Dell laptop connected to a Xbee chip was used as the base station for this device. This functioned well in a prototype setting, but in a production setting, leaving a fully functional computer running 24/7 greatly increases the power use of the monitoring system, which does not aid the goal of reducing energy consumption. As a result, this system would be converted to run on a low-cost and low-energy netbook, or a custom base station that has the networking capabilities to upload data to the web and a display for showing real-time power usage information. Additionally, smart-phone applications for popular platforms (such as the iPhone and Android) would be written to interface with the database on the server, show real time, category-differentiated, and aggregate power consumption information, and enable consumers to switch devices on and off remotely via the relays in the meters.

The information the CIEMAS power data did not contain was at least as significant as the information it contained. While baseline power consumption was able to estimated, and strong trends were shown based on the day of the week and the time of the day, it was not useful for identifying where this energy was being used, and how this energy could be saved. Device-level energy usage information, both in the quantity the device consumes when it is on, and how often the device is on for, and category-level energy usage is key in making consumers aware of both where their energy dollars are being spent and how to optimize their behaviors to minimize their energy bills without significantly reducing their free time or quality of life.

## 8 Conclusions and Future Research

This project analyzed methods of measuring device-level power usage, storing this information, and displaying it to the user. A prototype system was developed that measured power usage, collected the data via Python and Matlab scripts in the base station, and then uploaded to a MySQL database online. Real-time information was displayed on the base station, and historical power usage could be viewed online. The base station and online database could aggregate information from various types of meters, including outlet-level meters via an Xbee interface and whole-building meters via an online interface. It was concluded that the whole-building power usage information, while valuable, is insufficient for consumers to optimize their power consumption. Also, the preferred method to measure outlet-level power usage is via a customizable and extensible platform that could also provide point-of-use feedback in the form of light and sounds.

There are a number of interesting avenues to explore in the future. On the power-meter hardware side, the outlet-level meters could be designed to a production-level form factor that fits smoothly over a normal outlet, accurately measures the power usage of connected devices, and broadcasts this information via a robust mesh network. Additionally, LED lights could be used to provide feedback on the power usage relative to what is normal for the device. There are a number of start-up companies developing similar hardware to solve this challenge, and this hardware could also be used to study other research questions once it is released to the market.

For the base station, hardware and software could be developed to aggregate, display, and store this information in a compelling and energy-efficient manner. The developed on the server side could be extended to allow for multiple users, buildings and households. Additionally, various forms of data visualization could be explored and tested for there impact on the resident's energy usage. Home automation techniques could be added through the ability to turn devices on and off remotely via the relays, providing additional convenience and power savings. Additionally, applications developed for smart phones could be valuable for alerting consumers about unexpected power spikes, while providing additional data visualization and home automation options.

One of the most interesting lines of future work would be the development of models taking into account home-size, weather, the date, the time of day, and other factors to estimate the power usage of typical residential homes at any given point in time. These models could be used to help identify when the homes are using more or less than the expected power, and encourage consumers to behave in manners that would optimize their power consumption. Additionally, this data and real-time feedback could be used by power companies to change cost of power over the course of the day, to help even the loads on the network and decrease the expensive need for extra capacity during peak hours. For example, real-time pricing and feedback could help encourage consumers to perform energy-intensive tasks such as washing and drying clothes later at night, when there is less load on the network and the power could be correspondingly cheaper.

## 9 Acknowledgements

I would like to thank my advisor for this project, Dr. John Board, for the idea and the tremendous amount of support he gave me along the way. I also would like to thank Mr. Dhruv Kshatriya, who constructed ten Tweet-a-Watts over summer as a part of his project and was always available to discuss various ideas. Mr. Aurel Seleazenu was valuable in providing me access to the power data for CIEMAS, in addition to helping with Token Energy. Mr. Anup Engineer and the Token Energy team were helpful with discussions involving the future of US energy in addition to helping a marketing team at NC State analyze the marketing implications for a class. Also, I would like to thank the Smart Home Director, Mr. Jim Gaston, for the support and connections he provided to me as a Smart Home Fellow.

**References**

1. "Electrical Power Annual – Summary Statistics for the United States." US Department of Energy. http://www.eia.doe.gov/cneaf/electricity/epa/epates.html
2. "Energy Star." http://www.energystar.gov/index.cfm?c=about.ab_index
3. "DIY Kyoto." http://www.diykyoto.com/uk/wattson
4. "Onzo." http://onzo.com/
5. "P3 – Kill A Watt." http://www.p3international.com/products/special/P4400/P4400-CE.html
6. "Tweet A Watt. A Safe and Simple Wireless Power Meter." http://www.ladyada.net/make/tweetawatt/
7. Jiang X et al. "Design and Implementation of a High-Fidelity AC Metering Network." University of California, Berkely. http://buzzing.cs.berkeley.edu/~xjiang/papers/jiang09acme.pdf
8. "State Climate office of North Carolina." http://www.nc-climate.ncsu.edu