UNIVERSITY OF CENTRAL FLORIDA

Wattzilla

EEL 4914 – Senior Design

Group 3
Matthew Beck
Kyle Ferguson
Travis Helms
Dejan Lee-Hue

Table of Contents

1 Introduction

1.1 Executive Summary

This project is going to be a home automation and power monitoring system that will give a user more control over their home and keep them more aware of their power use. Every month, countless people are surprised by their power bill, partly due to the fact that they just aren't aware of how much power they are using on a daily basis. Occasionally, in situations such as vacations or day trips, or even just daily life, someone may accidentally leave something on in their house that contributes to higher electricity bill, or maybe they just want to remotely turn on or off their lamps. Perhaps they aren't sure if they remembered to lock up their doors and want to make sure, but they can't just go back home to check. The project is designed to help with all of these issues.

The group plans to design a system that monitors total home power usage to display back to the user, as well as monitoring the individual power usage of whatever is plugged in to the custom-designed power outlets or of any lights that are wired into the system. The usage statistics will be provided to the user in a number of different methods, including a website, a phone app and possibly even SMS text message.

In this paper they will discuss the components of the system in detail. The research section will explain what they have found out about the concepts related to each component and what they have found as their available options for implementing these components in the project. They will discuss the pros and cons of the more viable options they find, but final decisions will not be discussed here.

The design sections will consist of the final decisions on what resources the group will use, along with descriptions of how they will implement the components in their final design. The prototyping section will be more detailed than the design sections, discussing exact details of each of the components such as pins and pseudo-code and dimensions, that sort of thing. Testing will of course contains the test plans for each component, with test plans for each individual component as well as a final overall test plan for the project when it is put together.

The point of this project is to be a culmination of the years spent in each group member's respective major, requiring them to fall back on all of the different things they have learned, both electrical- and computer-oriented. Since the team consists of two electrical engineers and two computer engineers, they feel that the scope of the project provides the right balance between the focuses of each of those majors. This project contains plenty of electrical engineering stuff in components like the outlet, light switch and door lock, plus plenty of computer engineering stuff like the image processing software, web site and the central hub that communicates with the other devices Many of these concepts even straddle the boundaries between the two majors, since they are so closely related to begin with. They believe that this project will be a fine way to put their skills to the test and show off what they can accomplish with what they have learned.

1.2 Goals, Objectives and Specification Requirements

1.2.1 Goals

This project has two main goals in mind. The first goal is combat that surprise and shock that someone might have when they get their electricity bill each month. The group wants to keep the user aware of their power use at all times. They plan to accomplish this by monitoring their power use and displaying it back to them in a convenient manner, so that they always know what to expect from their power bill. They also want users to be able to get an idea of where their power use is coming from by offering monitoring of individual devices as well as the entire home. While the main goal is to just keep the user aware, they hope that by using this system, a user would be able to actively reduce wasteful power use and reduce their monthly bill.

The second objective for the project is to provide convenient control over parts of a user’s home. The group plans to provide a user with the ability to turn on or off lights or outlets in their home remotely using a website or a phone app, or possibly even using SMS text messaging. They hope that this will assist the user with the first goal of reducing energy waste, but also that it will provide the user with the kind of convenience that they expect in today’s world. With these goals in mind, they are setting out to create a truly great smart home system.

1.2.2 Objectives

In order to monitor the power use for a home or device, the system will need to be able to access this data. Splicing monitoring hardware into home power supply lines could work, but that idea is dangerous, since the group members don’t have the training for it, and they could end up with costly damage to their homes or even potentially lethal injuries. They still need to be able to find out how much power is being used though, and fortunately there is a device already built in to every home that can tell them just that: a residential power meter. They decided that they want to use this as their method of monitoring the power use for an entire home. In order for the system to get this data in a completely non-intrusive way, they will use a camera to record the meter, and they will create image processing software to “read” the meter. The group will design the image processing software themselves, and have it run from their project web server. The software will include error-checking capabilities and would be able to handle rollover, should a meter get to that point. The software will not be specific to one style of meter, and will be designed to handle different styles of digits, whether a meter’s display is digital/7-segment or analog.

For individual electronic device monitoring, the scale is much smaller than for an entire home, so they can monitor the power use directly. They plan to develop a type of power-strip-style outlet that plugs into an existing outlet. This outlet will allow the system to both monitor the power usage of the device plugged into the outlet as well as control power to the outlet, allowing the outlet to be turned off. This would be useful for finding out how much of your power bill comes from any particular item, such as a refrigerator or a washing machine (of course this wouldn't work for things that are hard-wired into the house's electrical lines, like central air conditioning or built-in lighting). It also allows for remotely turning on or off a device such as a lamp or a fan.

In order to provide a solution for hard-wired devices, they will also be developing a light switch device with a push-button switch that can be controlled remotely. The light switch device will monitor the power of the light that it controls and will relay this back to the user, as well as telling the user whether the light is on or off and allowing the user to turn it on or off from the website or app. This will provide a proof of concept that could later be adapted to other devices, such as the central air conditioning mentioned earlier.

Finally, the group will also be developing door lock bolt that can be controlled from the website and app, or manually with a key. The lock will be able to report its state to the website and app, so that it can still know if the lock was locked or unlocked manually.

In order for a user to view their power usage, the group plans to develop a website and a phone app for the system. The user would provide the site with their specific power costs, including any variables such as a higher price after a certain amount of power and the day of each month that readings are taken by the power company for the bill, in order to provide a real time estimate for the user's current power bill for the month and possibly a projected estimate of the final bill. The user could of course also view the actual amount of power used in kWh, and a record of past months' power usage/bill amounts.

The website/app would also provide controls for any of the custom outlets, lights or door locks that are part of the system, allowing them to turn on or off power to the individual outlets and lights or lock or unlock the door. The user could label particular devices for ease of knowing what device they're controlling, and of course would be able to see the individual power usage for each outlet or light. A timer feature may even be implemented to turn on/off certain outlets at pre-defined times. The system may include a feature to allow the user to receive a daily SMS text message to their phone with their current power bill, or alert them if the system needs their attention. The interface for the website/app will be user-friendly and easy to understand. While the real-time power information should be fairly accurate, any power bill information will be pure estimation based on user input.

1.2.3 Goals and Objectives

1. Camera
   1.1. Must be waterproof
   1.2. Must be able to read the meter at night
   1.3. Must be able to provide a video feed to the internet
   1.4. Wireless capability is preferred
2. Image Processing Software
   2.1. Reading
      2.1.1. Must be efficient enough to provide virtually real-time readings
   2.2. Error Detection and Avoidance Requirements
      2.2.1. Must be able to account for rollover, should it occur
      2.2.2. Must be able to determine if a reading makes sense compared to recent readings
3. Central Hub
   3.1. Microcontroller
      3.1.1. Must be able to handle programming to control communication with devices and the web server
   3.2. Power Source
      3.2.1. Must regulate voltage
   3.3. Communication
      3.3.1. Must be able to communicate with devices wirelessly
      3.3.2. Must be bidirectional
      3.3.3. Must support multiple nodes
      3.3.4. Must be able to connect to the internet over Ethernet
4. Data Storage
   4.1. Must be a reliable and effective storage method
   4.2. Must provide easy interface with the parts of the system that will be storing and accessing data
5. Outlet/Light Switch
   5.1. Microcontroller
      5.1.1. Must have bidirectional communication with central hub
   5.2. Voltage Monitoring
      5.2.1. Must convert analog output to digital
   5.3. Current Monitoring
      5.3.1. Non-intrusive method is preferred
      5.3.2. Must correct DC offsets if significant
      5.3.3. Must convert analog output to digital
   5.4. Power Control
      5.4.1. Must use a relay circuit
      5.4.2. Toggle logic is preferred so that a power failure will not change the state
      5.4.3. If a light is turned on or off from the switch, it must be able to notify the rest of the system so that the saved state can be changed.
   5.5. Power Source
      5.5.1. Must be able to regulate voltage
6. Door Lock

6.1. Must follow the same requirements as lights, except for power monitoring which is not applicable.

7. Website

7.1. Must be user-friendly and intuitive

7.2. Must display current power use and estimated bill for the month so far

7.3. Must display power use and state (on/off) of system devices

7.4. Must provide the option to turn on or off devices and lock or unlock door locks

7.5. Must allow the user to name devices and door locks

7.6. Must be able to save data into and receive data from the database

7.7. Must be dynamic and be able to update in real time without a refresh

8. SMS Text Messaging (Optional component)

8.1. Must be free

9. Overall Project

9.1. Must keep spending as low as possible, because there are no sponsors, so everything the group spends is out-of-pocket

9.2. Must protect against power surges

1.2.4 Specifications and Requirements

1. Central Hub

1.1. Power source must run on 9 volt battery

2. Outlet/Switch

2.1 Microcontroller needs at least 2 analog-to-digital converters

2.2 Will use voltage divider for accuracy, cannot assume 120V/240V

2.3 Power source must run on 9 volt battery

3. Overall Project

3.1 Spending should not exceed project budget of $400

2 Research

2.1 Outlet Subsystem

2.1.1 Current Monitoring

The first step to power metering is to understand how to calculate the power usage in a device. Since this calculation follows Ohm's law, I need not state it. Instead, the group must find out how the power company charges the public for power, and then apply their method in the design of the group's power monitoring circuit. Power companies charge the consumer based on the kilowatt hours used in the current billing month. Since the consumer is charged per kilowatt hour, it's obvious that they must be by the amount of real power (P=I*V) used in the home. This calculation assumes that the power factor (see 2.7.4 for an explanation) is one. Now that it's clear how a power company bills its customers, the group can investigate ways to safely monitor the power usage in a device plugged into an outlet that they will be creating.

Since power usage is dependent on the total amount of current and voltage seen by the outlets in a home (or business), it's obvious that the group must first devise a way to measure the amount of current running through a device. Presently, there are two ways to measure both of these variables: directly, and indirectly. The term directly refers to the act of "plugging" live wires into a circuit they will then use to calculate the voltage (or current) used. While indirectly, means just that; indirectly sensing the current running through a device. While both methods are valid, they both have appeals and drawbacks associated with them, and thus deserve a thorough discussion on each.

The first method to investigate is that of directly measuring the current running to a device. This can be done by connecting a measurement circuit directly to a live wire to obtain a calculation. This method comes with many advantages and disadvantages. One advantage comes from the familiarity (gained in the many labs each of the group members have taken at UCF) of breaking a circuit and then using a current meter to re-complete the circuit to obtain a measurement. Another advantage is that of cost, which will become clear when the group begins to discuss specific devices used to accomplish the group's task of directly calculating the current running to a device. On the other hand, one major disadvantage of a direct connection is that of safety. Due to the high voltage and high current nature of the outlets the group will be dealing with, if they overlook an element in the circuit (such as correctly wiring the circuit) the outcome could be catastrophic. If the wires are underrated for the current they will be carrying, the wire could get too hot and lead to fire. Though these risked are minimized through careful planning and design of a circuit (and the wiring associated with that circuit), it is still important to note as a disadvantage. Another disadvantage is that of a minor power loss in the wires leading to the measurement device, and the power dissipated in the device. As you can see, though they favor this direct method due to the group’s comfort, there are associated cons with the method and thus must investigate the other method of indirect current measurement. These pros and cons are compared in Table 2-1.

Table 2-1 - Direct Current Measurement Pros and Cons

Pros	Cons
Familiarity with connections	Safety Issues
Cost	Power Loss

When it comes to direct current measurement, the group are presented with a few different approaches to accomplishing the task. One approach is to use a step down transformer. A second approach is to use a current shunt. A third approach is to use an Integrated Circuit (IC) or a System on a Chip (SOC) to measure the current. It is important to note that there are many ways to approach this problem, but the group has narrowed it down to three of the most common methods for the purpose of simplicity.

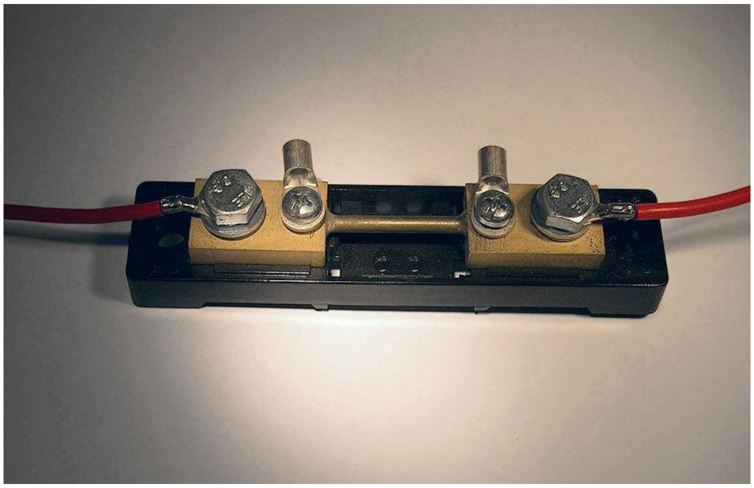
When first presented with the task of designing a circuit that could accurately (and safely) measure a large amount of current, the first method that came to mind was that of using a step down transformer to step down the high voltage to a level the group is comfortable working with. A step down transformer basically does just that: transforms a high voltage circuit into an equivalent lower voltage circuit. One advantage to this method is the ease of design associated with this fairly basic circuit element. On the other hand, a few disadvantages appear with the use of this element. One disadvantage is the size of the windings associated with the transformer. Another minor disadvantage is that of the risk that if for some reason the covering of the transformer were to come off, the high voltage located across the winding could easily arc a charge to one of the surrounding elements, or a surrounding person! The pros and cons are weighed in Table 2-2. Overall, this method is a little outdated and thus the group will continue searching for an alternative.

Table 2-2: Use of a Transformer

Pros	Cons
Ease of design	Safety issues
Basic circuit analysis used	Outdated method
Cost	Power loss

A second method of measuring the current running through a device is by using a current shunt. A current shunt is basically a low valued resistor, with a known scale of current vs. voltage drop, that can be used (by determining the voltage across the shunt) to tell how much current is running through the circuit. The shunt, shown in Figure 2-1, allows for a low voltage measurement across the element, and thus (typically) provides a linear method of calculating the current running through a circuit (without the need for a meter that can measure high voltages). This element provides another level of comfort with obtaining a current measurement from dangerous current supply. But, just as with the method of using a transformer as a means to transform a circuit to a much safer current value, the current shunt presents a number of safety issues. One such issue is that of the heat associated with passing a lot of current through a resistor. This heat can reach unsafe levels if a correctly rated current shunt is not used. Also, it is important to note that some current shunts only operate safely (at their rated current) at very low temperatures, and thus one must be careful to note the temperature of the location of the circuit on which the shunt will be placed.

Figure 2-1: Shunt Resistor



A third method of measuring the current running through a device is by using either an Integrated Circuit (IC) or a System on a Chip (SOC). The advantages of both of these devices are obvious: they are typically smaller, use less power, and are cheaper to make. Once such IC the group found, was that of the ACS-712 (produced by Allegro Microsystems and shown in Figure 2-2). This IC is designed to sense current from ±5 to 30 Amps, and is based on the Hall Effect principle. The low cost, tiny size, low power use, and simplicity of use of the device make this a heavy favorite (in the “direct” category) in that it combines advantages of both direct and indirect current measurement methods, while limiting the associated disadvantages of both.

As you can see, the method of directly measuring the current coming from a high voltage outlet is fairly well documented and thus presents many different ways to accomplish this task. Each method comes with advantages and disadvantages (discussed above and tabled in Table 2-3), but as a whole the idea of having to connect high voltage wires to a measurement device does not appeal to the group. This conclusion brings the group to the remaining method: indirect current measurement.

The other method of measuring current from an outlet is that of using indirect current monitoring devices. Indirect monitoring refers to the act of avoiding the use of high voltage (“live”) wires as an input to a device or other type of circuitry. This method is mostly predicated on sensors that use the Hall Effect to deduce the amount of current running through the tested wire. Though similar in the fact that they all rely on the Hall Effect, these “indirect” devices vary in the way that they output the results. Before the group can pick which sensor to use, they must first understand how the Hall Effect can be used to successfully quantify the current running in the wire of interest and the allowed accuracy of the sensor.

Figure 2-2: Allegro Microsystems ACS-712. Reprinted with legal authorization listed on Allegro Microsystems website.

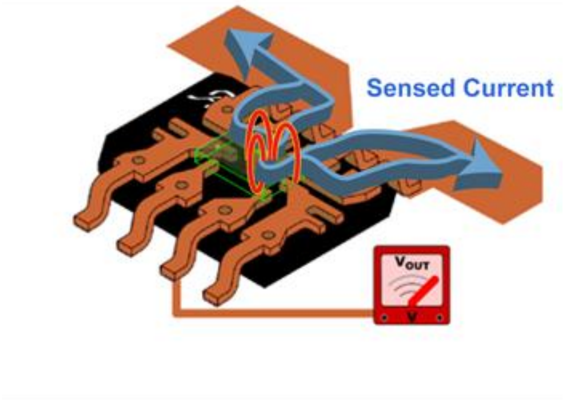


Table 2-3: Advantages and Disadvantages of overall method

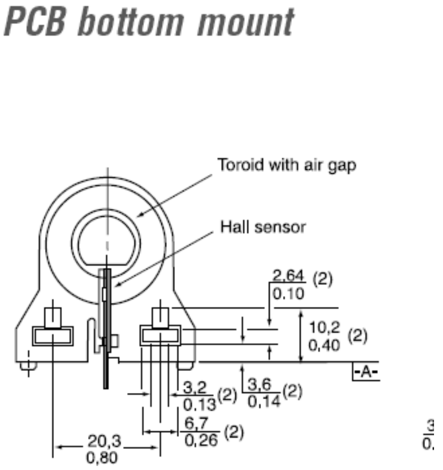
Advantages	**Disadvantages**
Fairly basic wiring	Danger of working with HV lines
Low Cost (per part)	Temp. of area must be considered
Basic Design needed	Potential damage to electric system

First, it is important to note that the Hall Effect relies on the principles of Faraday’s law. This basis on Faraday’s findings implies that the Hall Effect uses properties of magnetic fields to deduce a current (when placed inside the field).

The Hall Effect basically states that when a wire carrying current is placed inside a magnetic field, the electrons in the current will be deflected to different sides of the conductor (in 45 degree angles; demonstrated below in figure 1-5). The effect also states that based on this difference in resistance across the conductor, you can figure out the amount of the current running through the wire (while keeping the magnetic field constant) (Hall, 1879). This allows for a sensor to base itself on the Hall Effect principle, which ultimately allows for a high precision way of quantifying the current running through a wire without disturbing the circuit in which the current runs!

Now that the group has established the precision of Hall Effect sensors, they will discuss two such sensors that appear most favorable. One sensor that appeals to the group is the Honeywell CSLA2CD Hall Effect current sensor (shown in Figure 2-3). This sensor is favorable to others they have researched because of the linear (steady) voltage output, its ability to be mounted in different fashions to a PCB, and its low cost nature. It is able to sense up to 72 Amps of current indirectly, while requiring only between 5-12 DC volts of power to the device. It is also important to note that the linear DC voltage output (example of output shown in Figure 2-4) allows for it to be directly fed to an “Analog to Digital” converter without the need for conditioning the signal first. The only drawback that was found, when researching this sensor, was the apparent size of the device. This drawback is considered a minor one because of the added safety coming from not having to directly connect to a high voltage line, and the ease of use of the device.

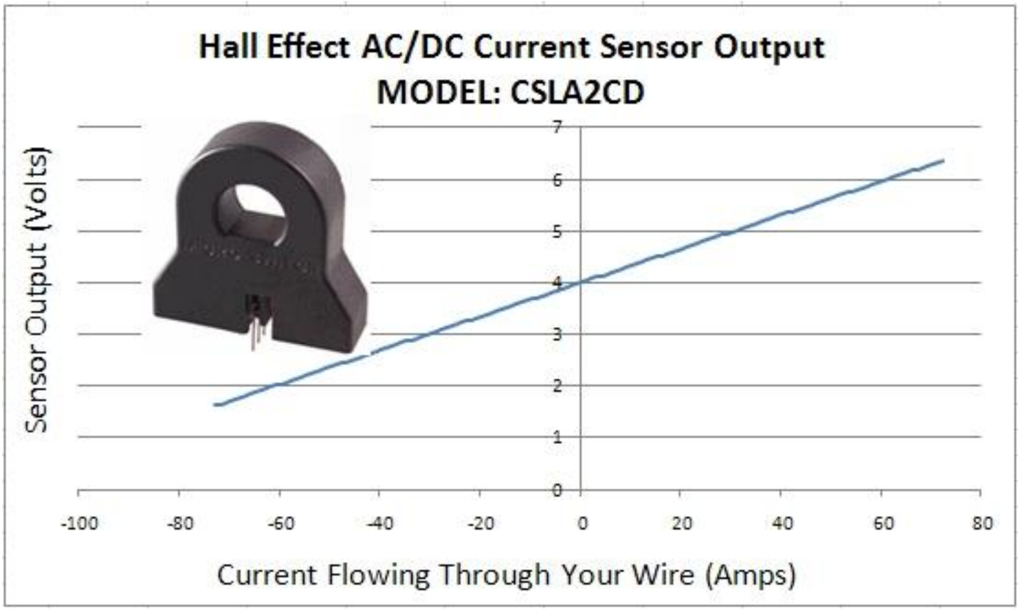
Figure 2-3: Honeywell CSLA2CD. Reprinted with legal permission listed on legal portion on Allied Electronics website



Once fed to the ATD converter, the group must determine the voltage output of the current sensor when the current is zero. This Zero current value and corresponding voltage will give them the DC offset of the output, and thus the group can subtract (or calibrate) the offset via software to read 0 volts at zero amps. Once calibrated, they will need to map a couple of different devices so that they can figure out the level of current that corresponds to the level of voltage output. Since this can easily be done, they will forego a discussion now and instead will discuss in the implementation section.

Another sensor that appealed to the group was that of the “Closed Loop Precise Hall Current Sensor” CYHCS-SH by Chen Yang technologies. This sensor provides similar package size, and a very stable linear output. Along with these two advantages, come more disadvantages. Two important disadvantages were that of high cost and high input voltage need (roughly double the one by Honeywell). If the group was going to be designing the power monitoring circuits for “high current” applications, this would be the wise choice. But since most household devices use less than 15 or 20 amps, they will not be using this sensor.

Figure 2-4: Current Sensor Linear Output Example



2.1.2 Voltage Monitoring

As the group seen, before one can perform a power calculation you must first find the current running through the device and the voltage seen by the device. As they will now be able to calculate the current running through the device, the group now turns their attention to the voltage seen by the device. When it comes to low voltage circuitry, you can directly measure (by way of a voltmeter, or other device) the voltage seen by the circuit element. But, as previously mentioned, high voltage circuitry complicates this once simple calculation. Also previously mentioned, the group prefers the indirect method to obtaining voltage and current measurement as a way of shortening the list of necessary safety precautions. Short of directly connecting a resistor to the high voltage circuit and then indirectly measuring the heat given off from the resistor (which as most engineers know leads to a large amount of power loss), there doesn't appear to be a safe and simple way to indirectly measure high voltage. This leaves one answer to the question of how to determine the voltage of a household outlet: voltage assumption.

Before they can discuss the validity of a voltage assumption, the group must first discuss the regulation of household power. It is a commonly known fact that a standard household outlet in the United States is supplying 120V (at 60Hz). In order to maintain safety and usage of a device, the power at the outlet must be fairly well regulated. That is to say, it's safe to assume that no matter which (labeled) 120V outlet you plug into in the U.S. you will most likely receive around the labeled voltage. In as early as 1989, a list of the range of voltage that should be seen by a device plugged into a 120 volt outlet was released (outlined below in Table 2-4). This range basically says that the voltage at a 120 volt outlet should be somewhere between 110 and 125 volts as a minimum b (Acceptable Voltage Ranges). It's important to note that as of late it would make monetary sense for power companies to regulate this voltage even further to avoid wasted (or unnecessary) power in devices.

Table 2-4: Range 'A' Voltage Regulation

Nominal Voltage	**Minimum Voltage**	**Max. Voltage**	**~= +/- %**
120	110	125	5

Now that the group have discussed a range of voltages they can expect at an outlet they will be monitoring, they are again faced with numerous methods for calculating the voltage, of which they will be further researching two. The first method is that of assuming the voltage at the outlet to be 120V. A second method is that of connecting a voltage divider circuit, and adjusting it so that the range of voltage seen matches that of the "A to D" converter which will be monitoring the voltage. Just as for current monitoring, both of the methods discussed are valid but have drawbacks that they must consider in order to be thorough with the group's investigation.

The first voltage measurement method to discuss is that of assuming the outlet to be 120V. At first glance, this method appears to be the preferred method for its simplicity and a decent accuracy (demonstrated in Table 2-5). This solution quickly gets passed over because of the allowance for even higher accuracy given from recent "high precision/low power" devices. Also, it is important to note that although the power company is usually extremely precise with its power delivery, one thing they need improvement on is the handling of ever growing urban areas. This high demand leads to uneven power distributions which can then cause lower voltages ("brown outs") seen by the devices in the home. These brown outs (and even power outages!) could lead the devices to a drastic loss of precision since the device wouldn't be able to calculate the power based on the lower voltages and thus would give the group false readings. It is for all of these reasons that the group has decided to monitor the voltage seen at the metered outlet.

Table 2-5: Voltage Assumption Validity of Normal Operation

Actual Voltage	Nominal Voltage	Error Calc	Precision (%)
110	120	$\frac{120-110}{120}$	+/- 8.33
125	120	$\frac{125-120}{120}$	+/- 4.17

The second method to calculating a voltage at an outlet is by way of a voltage divider. This method is fairly basic, since it relies on the fundamental concept taught in many freshman Electrical Engineering classes. A voltage divider does not necessarily mean that the group can just hand a lower AC voltage to the ADC though, since they will be required to do some signal conditioning to arrive at a steady voltage capable of being ready and translated into a binary value for computation. Along with signal conditioning, the group must take into account the direction of the current flow and could thus possibly need to use a diode or two to control it. Another potential issue the group must consider is that of a major power flux. A flux in the power could fry the microprocessor's ATD converter if not properly handled, which would mean they might have to use a Zener diode with a turn on voltage below the limit of the ATD to thus allow a path to ground, circumventing the risk. One final issue with this method is the need for resistors made to handle high voltages, and wires capable of carrying a high current away from the ATD to the load. Although this method comes with quite a few things to consider (listed for completeness in Table 2-6), it is still the most simple and accurate method available to safely measure a high voltage.

Table 2-6: Method 1 vs. Method 2

Method	Pros	Cons
1	Simplicity	Lower Accuracy, no accounting for voltage flux
2	Basic Concept, High precision, voltage flux accounted for	Many circuit elements, obvious higher cost, high attention to detail needed

As you can see, the two main methods of obtaining a voltage seen by the monitoring device are the assumption method and the voltage divider method. In order to increase precision, the group has decided to use the voltage divider method to pass a value to the ATD converter for reasons previously mentioned. Although this adds to the device's part count, it does not necessarily increase the group's budget because of the fact they should be able to obtain sample parts from various vendors in the hopes of the group using their parts in the group's final production.

Now that the group has determined how they are going to obtain a raw current and raw voltage measurement for the device plugged into the outlet monitoring device outlet, they must now prepare the signals to be safely passed to the ATD's.

2.1.3 Circuit Control

One of the most important decisions the group will face in the design of the outlet and door controllers is that of how exactly they will control each device. Since each of the devices has a different set of associated criterion, they will not necessarily assume that they will use the same type of control for each and thus must investigate the devices separately. In regards to each device, there appears to be two options for control: Electromechanical relay control, or solid state relay control. As with the various sensing techniques for current, each method of control has its advantages and disadvantages, and will now be discussed.

Of the two methods, it is obvious that the electromechanical relay has been around the longest. In an electromechanical relay, an input voltage is applied to the coil. This voltage induces a magnetic field which, in turn, pulls the switching arm to it (Solid State Relays vs. Electromechanical Relays). Because of the need to use a physical switching element with traditional contact pads, a bounce is destined to occur (shown in Figure 2-5). This bounce will then most likely add a level of noise to the output signal and thus must be dealt with. A second implication of using physical moving parts, is that of a device lifetime. A typical Electromechanical relay has a lifetime of around 100-200k switching “times” before the device starts to degrade and must be replaced. This short lifetime is a serious drawback for the group's application because the group doesn't want to have to worry about how many times they turn the device off/on versus how many switching “times” they have left. Another implication of using an electromechanical relay is that of the typical package size being (on average) 25mm by 10mm by 25 mm, which is a large device when compared to other circuit elements they will be using. A third implication of using an electromechanical relay is that since the input to control the electromechanical relay is not a low level input, they must use a transistor so that they can accept a logic level input from the microcontroller, which will be controlling the relay. It is important to note that due to the nature of a relay, both devices the devices will be investigate will be electrically isolated from the other circuit (which is a major advantage in the group's application. One final advantage of using this type of relay is that when the relay is in the off position, no current flows (thus saving energy). But, because of all of the listed drawbacks (listed in Table 2-7), they will consider the other method for circuit control: Solid State Relays.

Figure 2-5: Typical Electromechanical Relay

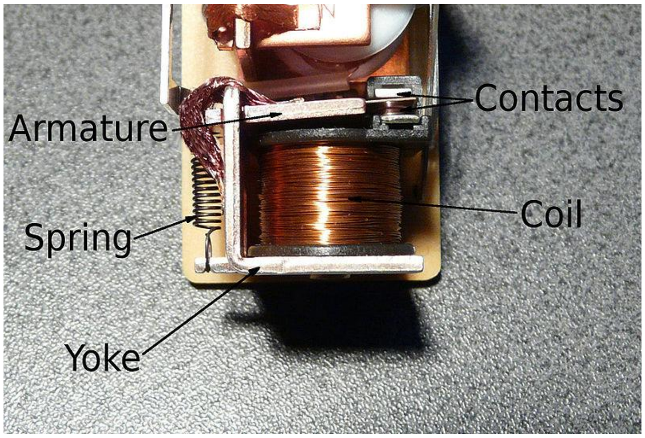
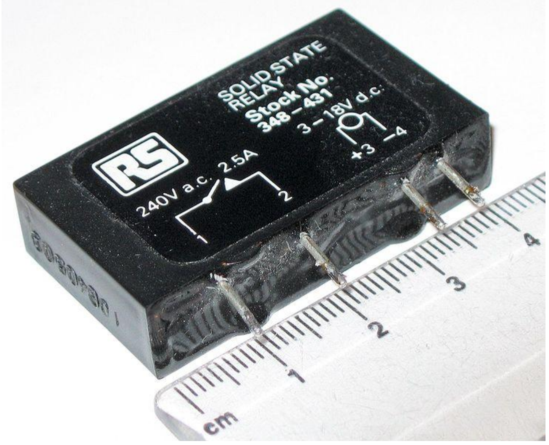


Table 2-7: Electromechanical Relay Pros and Cons

Pros	Cons
Electrically Isolated	Package Size
Safely handles high voltages	Short Lifetime
No current when off	Not logic Level input

The second method for switching a high voltage device on/off is by using a Solid State Relay. A solid state relay (shown in Figure 2-6) uses an LED to power a set of optically powered diodes, which in turn controls the circuit. Like electromechanical relays, Solid state relays are electrically isolated. Solid State relays are also bounce free since there are no mechanical (moving) parts. This lack of moving parts gives nearly 10x the lifecycle of an electromechanical relay (Solid State Relays vs. Electromechanical Relays). The control of the relay is based on a logic level input, and thus eliminates the need for other switching transistors to control the relay. Another advantage of the Solid State Relay is the tiny package size (roughly 1/2 the size of an electromechanical relay). Though there are many advantages, there are also a few disadvantages associated with solid state relays. One disadvantage is the need for a large heat sink when controlling large currents (above 15A) with a solid state relay which essentially negates the smaller package size. Another disadvantage is the fact that a minor power loss occurs when the device is off. Overall, the heat sink need for large current is an extreme drawback and thus will lead them to choosing and using an electromagnetic relay.

Figure 2-6: Example of a Solid State Relay



When it comes to searching for an electromechanical relay to use, there are a few options to consider. The first option to consider is the amount of current that the relay will be switching. A second option is the voltage required both to power the relay and switch the relay on or off. A third decision to be made is whether or not they want the relay normally open, or normally closed and then what combinations and switches are used to complete the relay. Since they will be designing circuits with a number of applications, they must consider each application and then make the corresponding decisions need to safely and accurately choose a relay for each.

The first circuit they must choose a relay for is that of the outlet monitor. The outlet monitor requires a relay capable of holding up to 20 amps, since a typical household outlet is rated for 20 amps. For this application, the group would like the relay to run on less than 9volts, and be switched by as low of a voltage as possible. Since they are not using a solid state relay, they must use a transistor to allow control of the electromechanical relay by an I/O pin of the microcontroller. Since they don’t want the power to the outlet to go off when the microcontroller goes off (due to a low battery) they must select a “normally closed” relay. On top of all these requirements, they also require that the relay be as small as possible, and cost as little as possible. These restrictions lead the group to the choice of using a T90 series relay. This relay (pictured in Figure 2-7, with the specifications shown in Table 2-8) is a form C relay, which means it is a Single Pull double throw capable of being both normally open and normally closed. It is also rated for 30Amps normally open, or 20 amps normally closed. On top of the mentioned capabilities, it comes with a nominal coil voltage of between 3-6 volts (making it ideal for matching the range of the battery the group will be using to power the microprocessor) and a small package design and price. All of these features make it ideal for use in the outlet monitor and any potential “add ons” they might choose for the group’s project!

Figure 2-7: T90N relay. Pending permission from www.onlinecomponents.com

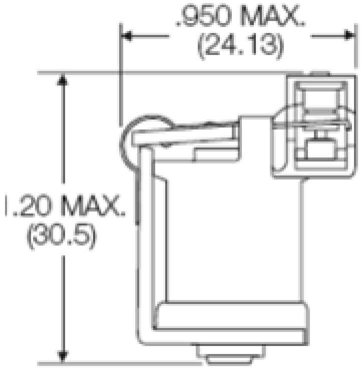


Table 2-8: Decision based on these specs, continued on next page

Manufacturer:	TE Connectivity / AMP Brand
Part No:	9-1393208-5
RoHS:	Yes
Type:	Power Relay
DC Coil Voltage:	6 V
Maximum AC Voltage Rating:	277 V
Family:	T90
Relay Construction:	Non-Latching
Maximum DC Voltage Rating:	N/A
AC Coil Voltage:	N/A
Coil Current:	150 mA
Coil Resistance:	40 Ohm
Contact Arrangement:	SPDT
Maximum Current Rating:	30 A

Pin_Count:	6
Maximum Pick-Up Voltage:	4.5 VDC
Minimum Dropout Voltage:	0.6 VDC
Mounting:	Through Hole
Maximum Power Rating:	N/A
Operating Temperature:	-55 to 85 °C
Dimension:	30.5 x 24.13 x 17.27 mm

2.2 Website

2.2.1 Web Development

The project requires remote access to a network. The network will be able to control electronic devices throughout a house. In order to make this control available to a user at any time while also allowing constant updates to the power consumption, there needs to be a website.

The internet will allow the user access to the home system network via a website which communicates with the central hub. The central hub acts as a switch allowing the user to turn on or off any device connected to the central hub.

When developing a website it is important to pick the right programming language. For the purpose of this project it is necessary to choose a programming language that is diverse with web development. For efficiency purposes it would be ideal to choose a language that can not only be used for web development, but also for other software components as well. The programming language should also make communication with the central hub device smooth and non problematic. There are various languages to choose from. C # is great for web development; It is an object oriented language that can be used to write ASP.Net dynamic web pages and XML web services.

PHP is another great option. PHP has its benefits over other languages due to its embedding characteristics and open source availability. Scripts written in PHP also can execute faster than other scripting languages. It is also known to be a language that is simple to use. This is important since all group members would need to be able to understand a new language. This language is also open source, which can be useful when dealing with complicated algorithms. PHP is an extremely versatile language, with access to various databases. It can also access C libraries and java classes.

Python is also another option. Python is mainly useful for its compatibility with Bluetooth. Bluetooth was of one of the groups' options for wireless communication. This language would prove to be very useful if the Bluetooth rout is taken for communication. Python is also a versatile language. It supports object oriented and structured programming, but is a very unfamiliar language and could present complications..

2.3 SMS

SMS or (text-messaging) is an extremely popular form of communication. There are many uses for SMS, but for this project the group plans on using text messaging for notifying users when certain triggers are set of in the software and on the website. Some triggers that could be used are when a device turns on or off, or when the current or projected bill reaches a certain dollar amount as set by the user.

There are a few ways to send SMS. Instead of sending an SMS via e-mail, the group can use the GET and POST HTTP methods for PHP. If the group end up coding the webpage in PHP, this method would be the most convenient. When setting up an SMS call from a web page, an SMS Gateway must be chosen. The gateway handles the message delivery process. They just need to pass the message details to the chosen gateway. The group can send the request using PHPs curl library. Curl is a library which allows communications to different types of servers with different protocols.

If the website is developed in C#, then sending a text via an email could be a great way to go. C# has existing classes for sending email, and the group could just send the message to the user's phone number at their provider's text domain, so they would receive the email via text. The email can be sent through any smtp server that the group is able to access, whether they create their own private smtp server, or use a free existing one such as Gmail.

2.4 Servers/Hosting

Creating a website is only part of what is needed to create an online user interface. The group also needs to decide on a server to host the website. Two general options are available for hosting the website: pay for a hosting service, or create their own personal server.

The easier option is to pay for hosting service. Sites like GoDaddy.com and many others provide website hosting as well as a domain name for accessing your website for a monthly fee. The monthly fee is usually fairly low, around five dollars per month for most sites' basic hosting service. Most services offer support for a wide range of programming languages, including PHP which the group members are interested in using on their site, as well as SQL database capabilities. All of the hosting sites they have explored provide more than enough storage and bandwidth for their needs, and email accounts are not a concern, so the choice comes down to ease of use, language/functionality support and price. A table showing a comparison of relevant features of some popular hosting services is available in Table 2-9

Table 2-9: Comparison of hosting services

	GoDaddy	**Yahoo**	**bluehost**	**iPage**
Price	$4.99/mo	$5.99/mo	$3.95/mo	$3.50/mo
PHP support	Yes	Yes	Yes	Not listed
SQL support	MySQL, MS SQL Server; 10 1GB MySQL DBs, 1 200MB MS SQL Server DB pre-provided	MySQL, phpMyAdmin, number of DBs only limited by storage space	MySQL, PostgreSQL, no reported limit, assumed limited by storage space	MySQL, phpMyAdmin, no reported limit, assumed limited by storage space
ASP.NET support	Yes	Not listed	Not listed	Not listed

While paying for hosting service may be easier than configuring their own server, hosting their own server would provide more direct control of the server. The biggest benefit of course is the cost, or lack-thereof (although a broadband internet connection is required, all group members have this already). Creating a personal web server from scratch is beyond the scope of this project however, so an existing server software would be used to create a server on a personal computer. One of the group members has experience hosting a website on a personal server in a previous school project. He used Aprelium's Abyss Web Server for hosting and Dyn (DynDNS at the time) for free DNS service to make the website accessible via a domain name. Abyss Web Server still exists in just about the same form as when the one group member used it before, but Dyn has since greatly reduced their free services, requiring at the very least a free trial using a valid credit card for access to one free hostname, and that one free hostname expires every thirty days. There are of course a number other options available for hosting and accessing a personal web server, some of which will be explored in the next few paragraphs.

Abyss Web Server is a small web server that is installed on your computer and requires almost no setup beyond the initial installation. There are of course many options that can be configured based on what the server is needed for, but once installed and started up, the server can be accessed by typing either "localhost" or "127.0.0.1" into the address bar of a web browser on the computer the server was installed on. The server can also be accessed by typing the computer's IP address into a web browser on any other internet-connected computer or device. This IP address may change, however, so connecting by typing in the server computer's IP address is not a very practical access method. The best solution to this problem is Dynamic DNS (DDNS) service from companies like Dyn mentioned previously, and will be covered later on.

Once the server is set up and accessible, there are a large amount of options available for creating the website. Abyss offers support for PHP and ASP.NET, among other languages, plus SQL database support. The server is accessed on port 80 by default, allowing external access via IP address without specifying a port, but this can be configured if necessary. In order to host an actual website, any website files can be put in the directory on your computer that Abyss points to for this purpose, where the main default page that should come up first must be called "index.html" or "index.htm". Any other file can be accessed directly, however, by typing in the URL (IP address or domain name provided by DDNS) followed by "/<file name>". Website files can also be placed in subfolders of the root site folder, if desired. With these features in mind, the only factors limiting the capabilities of the website are the functionality provided by the supported languages, the space available on the computer's hard drive, and the reliability of the computer itself and its internet connection.

An alternative popular option for personal server hosting is Apache HTTP Server. The documentation and user interface is slightly less user friendly than Abyss. It is however open source and therefore has a lot of external support, with users creating many different add-on features for it. The official Apache software offers support for PHP, and there are add-ons available to support SQL and ASP.NET. Apache works in much the same way as Abyss, providing access to the server with the same IP address methods, as well as using a directory on the hosting computer as the root folder for the website, where all of the website files must be held.

Another option is using Windows Server. Microsoft offers Internet Information Services (IIS) as an add-on for Windows Server 2008 R2, which provides a web server for hosting websites. While Windows Server 2008 R2 is technically a paid product, a free 180-day trial is available for evaluating IIS 7.5, which is enough time to last for the duration of the group's senior design project. IIS supports both PHP and ASP.NET, as well as MySQL and MS SQL Server. IIS also boasts about its media capabilities, which may be a useful feature for allowing users to view their power meter video stream if they decide to implement that feature. IIS offers integration with Visual Studio, allowing website publishing to the server directly from the IDE. Alternatively, IIS offers a user interface specifically for configuring website files. In either case, there's no need to just find the root site directory and drop in files.

As mentioned previously, accessing the website by IP address is not the best approach. IP addresses are not easy to remember, and the host computer's IP address could even change periodically. Ideally they would like to access the website with domain name style address like any other site you might access. The solution to this problem is Dynamic DNS. DDNS providers offer a service that points a domain name URL to the supplied IP address, with the ability to detect if that IP address changes and update the reference IP accordingly. As with any software solution, there are many DDNS service providers to choose from, both free and paid. Because the project does not need anything more than basic DDNS service for a single IP address, the group will only be considering free DDNS services to minimize project costs. A summary comparison of two services to be discussed is here in Table 2-10

Table 2-10: Comparison of DDNS service from Dyn and No-IP

	Dyn	**No-IP**
Free hostnames	1	3
IP update	Application or router	Application
Credit card	Required	Not required
Account Expiration	30 days of inactivity	Never
Hostname expiration	Every 30 days	Never

The DDNS provider mentioned previously, Dyn, is a very popular choice for DDNS, and has been used successfully in one of the group members' previous school projects. Dyn has recently scaled back their free service offerings, made it relatively difficult to find, and even going so far as to require signing up for a free trial of their minimum paid service with a valid credit card for automatic billing (although the automatic billing can be removed once the trial is activated). Free accounts expire after 30 days of inactivity, activity being considered either login in to the account on Dyn's website or the dynamic IP address being updated. The free DDNS service offers domain-to-IP hostname redirection for a single domain and IP address pair, providing the option to use a free domain provided by Dyn such as .dyndns.org, or purchasing an available URL through them in a more common domain such as .com. Even if a free account is kept active, the one free DDNS hostname will expire every 30 days and must be recreated. Dyn offers a downloadable application that tracks the computer's IP address and updates the one Dyn's service points to if it changes. Alternatively, many routers offer built in support for sending IP address updates to Dyn.

Considering how restricting Dyn has made their free DDNS service, an alternative site to be considered is No-IP. No-IP offers a free DDNS service for up to 3 IP addresses and, like Dyn, they offer an application that can update an IP address if it changes. Unlike the inconveniences of Dyn's free DDNS service, information on No-IP’s free service is easy to find, no credit card information is required, and accounts and hostnames do not expire. Other than those conveniences, No-IP’s service works in the same way as Dyn, so either is a viable option.

2.5 Database

For the project, the group will need to store data somehow. They need to store meter readings and meter details, power bill details, and potentially many other things. Because the system will ideally be account based, all of this data will need to be stored for every account. With that in mind, the most obvious solution is SQL. While there are other database solutions available, such as Oracle Database, one of the group members is very familiar with SQL, using is every day at work, and having used it in a previous school project.

SQL provides a table structure for storing data. Tables have columns which represent different pieces of data, and they have rows or records which each have some data in those columns. Tables usually have a column that is used to identify each record, called a primary key, and each record must have different data for the primary key. Tables can be related by having a column called a foreign key that relates to another table’s primary key, relating the data of a record in one table to the data in a record on another table. Using these relationships, lots of different data can be related in an efficient way, especially if many data items repeat frequently.

There are actually a few options when it comes to SQL, as there are different kinds of SQL solutions. The most popular SQL solution is Microsoft’s SQL Server, the current version of which is SQL Server 2012. SQL Server is what the experience group member uses at work, and while the paid versions can be pricey, there is a free version, SQL Server Express, that is meant for small desktop or web server applications, such as their project. Previous experience with the software could provide an advantage when developing the project.

The other main option for SQL is MySQL, an open source SQL server solution from Oracle. MySQL was used in the experienced group member’s previous school project, so he has some experience with it as well, although not as much as with SQL Server. MySQL has a free version of its SQL software and offers the same SQL functionality as SQL Server, so either one is good option.

2.6 Image Processing

There are two major things to consider about the image processing portion of the project. A camera is necessary to obtain an image in the first place, and it must be able to interface with the rest of the system. Once the group members have an image, the image must be processed through software that they will design. The output will be the power meter reading that will be used in determining the current power usage/bill.

2.6.1 Camera

The goal of the project is to digitize the output reading of a home power meter. The group wants to be able to read a value at any given time from a power meter and send that information in a digital form to the system. This information will be sent to a user in the form of a text message or via a smart phone app and website. The information about power consumption should be accessible to a user and should refresh constantly throughout the day so that the user can get an accurate reading of the power consumption of their home.

The problem is that this project needs to access the information from the power meter without actually hacking into system. Tampering with a power meter is not ideal. For this situation a camera is useful, because with a camera, the system can have the ability to transfer data in the form of an image or video.

The group's first thought was a USB camera, because building a camera is beyond the scope of this project. A USB camera would require the central hub to be USB-capable, and it would have to be able to run drivers to interface with the camera. A USB camera would also require a potentially very long USB wire, which may not be possible due to limitations to USB connectivity.

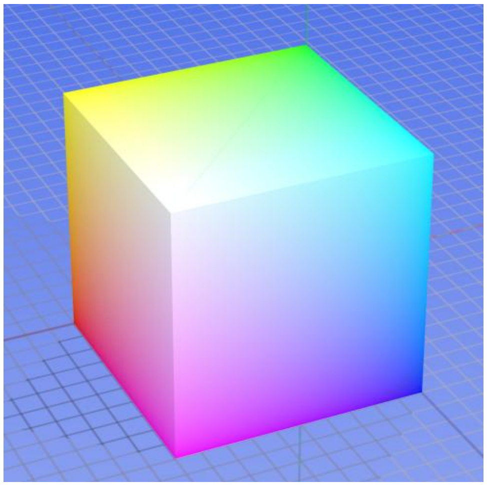
Another option is a wireless IP camera. This would allow the camera to send its video signal to the internet, without worrying about running wires around a house, or worrying about central hub compatibility. It would however require the user to have a wireless router, which may be prohibitive for some users.

2.6.2 Image Processing Software

Image processing software typically works with 2D images in the form of two-dimensional arrays of values representing the pixels of the image in some way. Individual pixels may be represented in a number of ways depending on both the image itself and the preference of the software designer. Pixels of monochromatic images may be represented as simply as an integer value within a range from a high value to a low value, representing the brightness or intensity of the pixel. Color image pixels are slightly more complex to represent however, because unlike brightness or intensity, color does not lend itself as easily to linear representation.

Two color representation models often used in computer vision are Red/Green/Blue (RGB) color space and L*a*b color space. RGB color is a widely known color model, combining various amounts of red, green and blue light together to produce a wide range of colors. In a color image, a pixel may be represented by a three-dimensional vector whose three components represent the value of red, green and blue color in the pixel as integer values, often ranging from 0 to 255. A visual representation of the RGB color model is shown in Figure 2-8, with the three color channels represented as axes.

Figure 2-8: 3-dimensional representation of the RGB color model using 3 axes to represent color channels. The origin (not shown) where all channels have value 0 is black, while the opposite corner where all axes are at their max value is white. Reproduced under Wiki Media Commons license

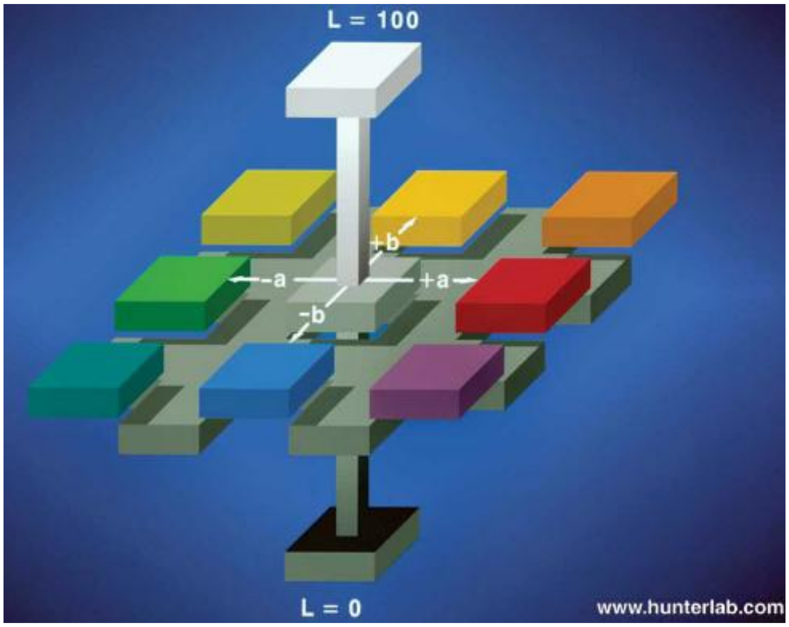


L*a*b color space is less commonplace but like RGB, it uses three values to represent color. Unlike RGB, the L*a*b color model uses the concept of opposite, mutually exclusive colors, allowing two opposite colors to be represented by a single value. L*a*b assigns red and green to the a value and blue and yellow to the b value, positive values representing one color of a pair while negative values represent the other color. The L value represents brightness and can only range from 0, darkest or black, to 100, brightest or white. A visual representation of the L*a*b color model is shown in Figure 2-9 below, with the three values represented as axes, similar to the RGB example.

Once a computer vision program has an image represented in such a way that it can work with it, all it can do is find patterns. Patterns are all that computer vision software can work with, but this pattern-finding can be done in many different ways to perform many different types of tasks. These tasks can range from something as simple as recognizing a specific shape, or as complex as motion tracking of an object across the multiple images of a video by finding the pattern of that object in each image.

In the case of this project, the group members need to recognize the pattern of digits in an image of a residential power meter. This can be complicated however, because there are a few different ways that a meter can show its power use reading. Some meters use seven-segment displays to just display a number directly. Others use a rolling display like that found in the odometer of an older car. Still others use a completely different method where there are several small clock-like circles on the meter with the numbers 0 through 9 around the circle. The current number that the circle represents is pointed to by a small hand the way a clock hand points to the current minute or hour. Each circle represents a digit of the full meter reading. This variety of meter will be referred to in this paper as a clock-digit meter. Examples of each type of meter are in Figure 2-10.

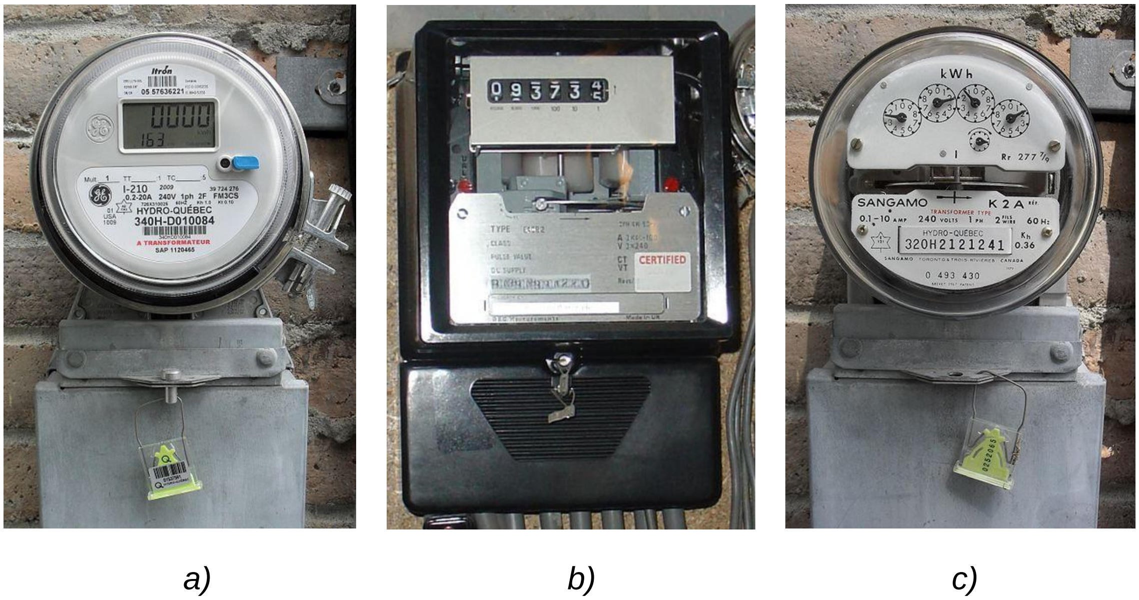
*Figure 2-9: 3-dimensional representation of the L*a*b color model. The L axis represents luminance or brightness and can have values only from 0 and 100, 0 being black (no brightness) and 100 being white (maximum brightness). The a axis represents red/green color and has no numerical limits, with positive values representing red and negative values representing green. The b axis represents blue/yellow color and also has no numerical limits, with positive values representing yellow and negative values representing blue. Permission pending from www.hunterlab.com*



Because the different styles of meter displays are so different, it becomes more complicated to read the meter visually. It should also be noted that the clock-digit meter may have its clock-digits oriented slightly curved as shown in Figure [K3], or aligned horizontally, which may also present its own challenge. These differences had to be taken into account when creating the image processing software.

For the purposes of this project, the group members wanted to build a program that will take an image as input, a frame of a video feed of a home power meter in this case, and process it to “read” the meter, outputting the meter reading to be used in bill calculation and any other functions that might need it. This program could either be hosted on the user interface website, or programmed directly onto a microcontroller.

Figure 2-10: Three different types of residential power meters: a) digital meter with seven-segment display, b) analog meter with rolling digits, c) analog meter with clock-like digits. All images reproduced under Wiki Media Commons license



a) *b)* *c)*

That said, there are a lot of computer vision software options available, including creating their own software from scratch or using the many pre-existing open-source libraries and code. The group members feel that what they end up using will likely be a combination of borrowed libraries and new code. The following details some available computer vision tools that could be built upon.

OpenCV was their first thought for computer vision software. OpenCV is a free, open source programming library for computer vision software. It was originally written in C but has since been converted to C++, which all new development is done in. An interface for Python has been created as well. Among other things, OpenCV's libraries provide functionality such as object/feature detection, identification, segmentation and recognition, plus general image processing, which is what they need for their digit recognition software. The software would still need to be coded by us; OpenCV simply provides a framework of functions to be used in creating it.

General image processing functions available in OpenCV include conversion between color models, plus conversion from color to grayscale or even to binary images. A binary image is an image where pixels have one of only 2 values, usually determined by a threshold on the original pixel values. Other available general image processing functions are image resizing and cropping, which may come in handy.

A feature detection function called Canny is available and uses the Canny edge detection algorithm to find edges in an image. And edge is considered to be a sudden change in color or brightness, relative to two thresholds determined by the user. The larger threshold is used to find what the algorithm considers to definitely be edges, based on the difference between the pixel and its adjacent pixels compared to the threshold. After the definite edges are found, any pixels at the end of a line of definite pixels have their difference compared to the lower threshold and if they pass, they are also considered edges. This lower threshold process is repeated until no more pixels pass the threshold. The result of the function is an image showing the detected edges, which should make it easier for the developed software to find digits in an image.

Another interesting function OpenCV offers is called matchShapes and is fairly self explanatory: it compares two images to match the shapes in them, giving a double that represents the likelihood that the images are a match, with lower values meaning more likely. This could be used in conjunction with some sample input images of digits that could be compared to digits in the image of the power meter, giving the actual digit value after the individual digits are found.

One more very interesting function in OpenCV is called grabCut which will crop or segment out a part of an input image based on a mask. This could be used with a mask that approximates what the digits section of the meter looks like in order to segment it and use the segmented part alone for more simplified digit identification.

The fact that OpenCV is a C++ library makes it seem like a decent choice for a microcontroller project because C++ is becoming more common as an option for programming microcontrollers. If the software is run on the web, however, C++ is a viable solution for that as well. One major concern with OpenCV is that it uses floating point math, while most common microcontrollers aren’t capable of. Other than buying an expensive microcontroller capable of using floating point math, using OpenCV on a microcontroller would require either floating point emulation or extensive tweaking of the existing code to remove dependence on floating points. On the other hand, using OpenCV in a web-hosted program wouldn’t pose a problem. The pros and cons of OpenCV are presented in Table 2-11.

Table 2-11: Pros and cons of OpenCV as related to the project

Pros	Cons
Wide range of functionality	Lots of dependency
Includes functions specific enough to be used for project needs	Large amount of files, requiring lots of space
Vast amount of comprehensive documentation and online help	Floating point math may pose problems for microcontrollers
Free and open source	

than OpenCV but with many of the same features that could be useful to the project. CVIPtools was developed and is maintained at the Southern Illinois University at Edwardsville. Like OpenCV, it includes image manipulation functions like resizing and color space conversion, as well as object/feature detection functions such as an edge detection function, segmentation, and pattern matching. Unlike OpenCV, however, CVIPtools is smaller in scale and therefore in file size. Also unlike OpenCV, detailed documentation on the functionality is found in a textbook rather than online.

Like OpenCV, the image processing functions of CVIPtools are written in C++ and use floating points, although not as heavily as OpenCV. Because of the language, CVIPtools is flexible in where it can be used. Because it doesn't rely as heavily on floating point math as OpenCV, it is more suitable for use on microcontrollers. Like OpenCV, CVIPtools as it exists on its own cannot be used directly for their purposes, but would rather provide a framework on which to build their own software. A summary of the pros and cons of CVIPtools is presented in Table 2-12.

Table 2-12: Pros and cons of CVIPtools as related to the project

Pros	Cons
Fewer files, requires less space	Less robust than OpenCV
Includes functions specific enough to be used for project needs	Documentation is available in a textbook rather than online
C++ language can be used flexibly	
Less dependency on floating point math allows easier use on microcontrollers	

One more preexisting image processing library is AForge.NET. As suggested by the name, AForge.NET is a .NET framework for developing in C#, and would be a great option for developing their software for a web server environment. AForge.NET offers many of the same image manipulation and object recognition tools as the previous two options, but it also provides a very useful method that the others don't have. AForge.NET has a build in function for grabbing a JPEG video stream directly from a given URL. This method would be extremely useful if the power meter video feed is linked to the web rather than directly to the system. This feature is AForge.NET's stongest advantage over OpenCV and CVIPtools, and since it depends on the code having direct access to the internet, it would only make sense to use it in a web server environment. If that is the route they take, then AForge.NET is likely to be the group members' library of choice. A summary of the pros and cons of AForge.NET is presented in Table 2-13.

Table 2-13: Pros and Cons of AForge.NET as related to the project

Pros	Cons
Fewer files, requires less space	Less robust than OpenCV
.NET framework/C# very useful for web server environment	C# is not as commonly supported on microcontrollers
Built-in JPEG fetching from URL	

Instead of using these complex preexisting image processing libraries, the group members could build all of these functions on their own, most likely using the existing functions as reference, but not using them outright. If that is the route they choose, then it would be a good idea to at least use an existing tool that provides an interface for working with images. CImg is just that; it is a single C++ header file that provides functionality for working with image files into any C++ project it is included in. It turns an input image into a usable data type and provides basic functions for image resizing, plus it gives access to individual pixels, making grayscale or binary color conversion easy to implement. The CImg header file is written to be very generic in regards to the operating system (or lack thereof) that it is used on, accounting for three types: Windows, Unix-based, or other/none. This makes it extremely flexible in how it can be used, whether on a microprocessor or a web server. A summary of the pros and cons of CImg is presented in Table 2-14.

Table 2-14: Pros and cons of CImg as related to the project

Pros	Cons
No other required dependencies outside of standard files	Provides no image processing functionality
Contained in a single 2MB header file	
Provides some optional plugins for additional functionality	
C++ language can be used flexibly	

Whether using the preexisting image processing functions of OpenCV or CVIPtools, or creating their own image processing functions using CImg as an interface for images, the options for reading the digits in an image remain largely the same.

2.7 Central Hub Communication

As described in the executive summary of this document, there were multiple different nodes that need to communicate to each other in order for the system to function properly. The topology of the network was a star configuration, with each element communicating with the central hub. Since each of these elements do not necessarily need to communicate with another element other than the central hub, they do not need to operate on the same bus. Therefore, more than one communication technology may have been used for each different connection. However, having many different types of communication could be cumbersome, so it has been a goal to minimize the number of communication technologies being implemented.

There are at least two different ways to categorize communication. The first of which is whether the communication is serial or parallel. Parallel communication allows for multiple bits to be transmitted simultaneously, while serial communication allows for only one bit to be transmitted at a time. Though parallel communication allows for higher transmission rates, serial communication is much easier to handle, and exists in many more useful forms. Hence, only serial communication has been considered for this project. Another way to categorize communication is by whether it is wired or wireless. Each of the above has their own advantages and disadvantages which are described in the following section.

2.7.1 Wired Communication

The following section describes various different wired communication techniques which were considered for the project.

2.7.1.1 Recommended Standard

There are many different protocols that could have been implemented for a hard wired physical layer. One of the most common of the "Recommended Standard" protocols is the RS-232(C) protocol. Its main advantage is that it allows simplicity when connecting to a personal computer. However, one of the goals of this project was to make a system that is stand alone, so that no other hardware will be necessary in order for the system to operate. Other drawbacks of the RS-232(C) protocol include its maximum cable length of 15 meters, its transmission rate of 20 kbps, its ±20 volt voltage range, and the fact that it only allows for two digital devices to communicate in the system; one transmitter and one receiver.

Other “Recommended Standard” protocols alleviate these issues include, but are not limited to, RS-423A, RS-422A, and RS-485. All of these protocols have much higher transmission rates, longer maximum cable lengths, lower voltage swings, and allow for more than two digital devices to be connected to the network at once. Communication protocols like the RS-485 operate with voltage differences as small as ±1.5 volts, allowing for interfacing directly to transistor-

transistor logic without the need for components to step up the voltages from the digital microprocessors.

2.7.1.2 Universal Serial Bus (USB)

Universal Serial Bus is another communication technology that was considered. One useful attribute of the Universal Serial Bus is that most digital embedded devices have Universal Serial Bus connectors. This could have simplified the process of selecting a development environment significantly, given that most development boards come readily equipped with Universal Serial Bus connectors. With one less feature to limit their selection to, more features and possibilities could have been explored.

Furthermore, most personal computers have Universal Serial Bus connectors, giving flexible opportunities for debugging. Two possible debugging modes include transmitting and receiving information. When the personal computer's connection is in transmitting mode, the receiving functionality of the device under test could be observed. Conversely, when the personal computer's connection is in receiving mode, the transmitting functionality of the device under test could be observed.

One of the most appealing aspects about choosing the Universal Serial Bus communication technology, however, was the fact that most webcams communicate using Universal Serial Bus already. This is particularly true for cheap webcams. Many cheap webcams start off at around twenty dollars, and are readily equipped with Universal Serial Bus connectors to transmit data about the image that it obtained through its lens. Considerations for the project were limited by costs, and not significantly by image quality (as long as a power usage reading could be obtained algorithmically from the captured image). Also, the fact that Universal Serial Bus communication technology is one to one (having one transmitter and one receiver) persuaded its use for communication with the camera focused on the power meter, given that there is only one of them.

One of the major limiting factors of the Universal Serial Bus communication technology was that the maximum lengths of connection cables are 5 meters (16.4 feet) for Universal Serial Bus 2.0 hi-speed applications, and 3 meters (9.9 feet) for Universal Serial Bus 1.1 low-speed devices. Solutions to these limitations included Universal Serial Bus hubs, Universal Serial Bus active extension cables, and Universal Serial Bus over Ethernet.

Universal Serial Bus active extension cables posed as the most practical of these solutions for the limited Universal Serial Bus cable length of 5 meters (16.4 feet). Up to 5 Universal Serial Bus active extension cables could have been connected, end to end, allowing for a maximum length of up to 80 feet. The cost for these would have been about 15 to 20 dollars each, and Universal Serial Bus ready cameras are generally about 20 to 30 dollars (for cheap ones); therefore the entire cost for communicating from the camera to the central hub using Universal

Serial Bus communication technology would have been at least 110$. This ended up being a detrimental figure to USB technology when compared with other wireless alternatives.

2.7.1.3 Power Line Communication

So far, all of the wired technologies discussed have one major issue: in order to use them, there would either have had to have been unattractive wires laying around the house running to the equipment, or they would have had to have been buried into the walls which would either have had to be done at the time of the house's construction, or the installation would be much more time consuming and costly. There is a wired technology, however, that could have avoided this cumbersome and inconvenient predicament. Power Line Communication is that technology. The idea of Power Line Communication primarily revolves around communicating using the copper which delivers power to appliances that get plugged into wall outlets. Along with this physical layer consideration, there are many proprietary protocols for this communication technology, including X-10, Universal Powerline Bus (UPB), and Insteon. After much research, the availability of development tools for these existing technologies is extremely limited, due to the proprietary nature of the products; the companies would rather have somebody just buy the whole product itself, rather than hacking it themselves, thereby actually owning the technology, and being able to use it for whatever they want.

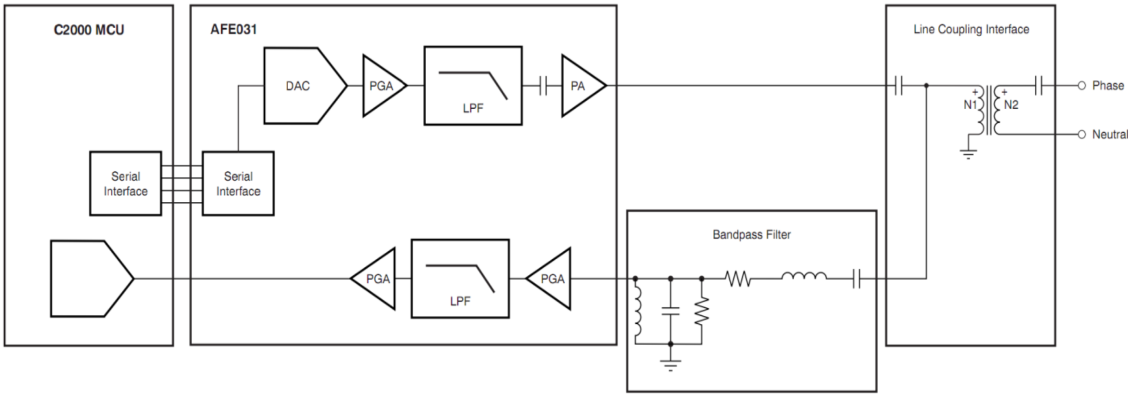
If this is the case, then it appears that the wheel would have needed to be reinvented. In order to transmit a message signal through the power mains, the message signal would have needed to be superimposed with the existing 60 Hz power signal (50 Hz in other parts of the world). It would have been assumed, and hopefully confirmed through testing, that the message signal will magnetically induce itself onto both phases in a typical split-phase power distribution system that exists in most residential circumstances. In order to receive a message signal through the power mains, the 50 or 60 Hz power signal would be filtered out with a high pass filter, or a bandwidth filter, in order to extract the message signal from whatever combined signal would be propagating through the mains. There were many filter designs that were considered for implementation in the application. In fact, the signal could have even been filtered on the digital side or the analog side. The most intuitive filter would most likely have been on the analog side so that the signal that would have required decoding would have been more manageable.

Both modes of transmission require some devices for interfacing between the digital microcontroller and the analog power lines, and all devices in the final design should support both modes of transmission. For the transmitting mode, a Digital to Analog Converter (DAC) would have been needed, and for the receiving mode, there would need to have been an Analog to Digital Converter (ADC). Also, both mode of transmission would have needed power amplifiers to amplify the message signal for reading or for sending.

There also would have needed to be a line coupling interface between the high voltage power lines and the discrete logic devices that will process the signal. A common implementation of this includes a transformer to provide electrical isolation from the power mains, and a high voltage capacitor to filter out some of the 50 or 60 Hz power signal.

Texas Instruments produces a relatively high performing analog front end device which allegedly “simplifies the design process dramatically,” called the AFE031. Along with its high functionality, the AFE031 allows for fine tuning to various CENELEC standards. It has included on it various filters, power amplifiers, a Digital to Analog Converter, a Serial Peripheral Interface (SPI), and a Zero Crossing Detector Block. While these embedded devices provide some functionality to the design, there still would have been many devices needed to complete the design. On the microprocessor side, there would need to be both a Serial Peripheral Interface and an Analog to Digital Converter (ADC). On the power line side, there would need to be a band pass filter, some type of surge protection circuitry, and a line coupling interface. There were also external capacitors and resistors needed to correctly filter the signals or correct impedances, all of which requiring to be chosen with care. The circuit in Figure 2-11 exemplifies the usage of the AFE031.

Figure 2-11: AFE031 Implementation



Contrary to most communication techniques, the carrier frequency for PLC is much lower than the frequency of communication. Also, it would not have been wise to disturb the power signal too greatly, due to unintended effects on other nearby appliances. In this case, the message signal would have needed to have much less power (much less of a voltage swing) than the 120 or 240 volt signal being delivered by the power grid. Many established Power Line Communication protocols use precisely timed impulses to send messages through the power lines. These impulses are referenced to the zero crossing point of the power signal.

While Power Line Communication technology is a unique and interesting notion, it would have put a tremendous strain on development, being that its development would have been long and arduous. It would have also necessitated that the microcontroller in the central hub and the microcontroller in the outlet monitor and controller both have an Analog to Digital Converter (ADC) devoted to Power Line Communication, and Serial Peripheral Interface (SPI) module. This introduced serious design limitations. It would not have been impossible to do, and was considered for the project, but it would have required too large a portion of the time and money budget.

2.7.2 Wireless Communication

Wireless communication was an excellent alternative to most wired communications. Similar to Power Line Communication, a wireless physical layer allowed for communication without needing to bury extra wires into the wall, or having them running along the walls, creating an unsightly appearance of one’s home. Also, most wireless communication technologies at least seem to have much less proprietary restrictions to development environments than Power Line Communication, making them an ideal choice for minimizing design obligations, and freeing up time to develop other potentially more interesting features.

There were three wireless communication protocols that were considered for the project. They were IEEE 802.11 (Wi-Fi), IEEE 802.15.1 (Bluetooth), and IEEE 802.15.4 (ZigBee). Interestingly, all three of these technologies communicate over the same frequency. However, each had its own collision avoidance and interference mitigating capabilities.

There were many positive aspects about wireless communication such as freedom and device mobility. It was also a much more aesthetic alternative to running wires all along walls inside and outside of the house. However most of these devices all communicate in the 2.45 GHz frequency band. Attenuating interference was a high priority in the pursuit of wireless communication.

2.7.2.1 IEEE 802.15.1 (Bluetooth)

IEEE 802.15.1 specifies a short range communication technology centered in the 2.45 GHz frequency band known as Bluetooth. Bluetooth was intended to replace RS-232 and Universal Serial Bus wired connections for devices within close range (10 meters) of a personal computer. One of its original applications was to mitigate the pain of actually having to hold a phone next to one’s face when speaking to someone remotely, even though to people nearby the user would appear to be a schizophrenic. Its innovation and robust design has made the technology an ideal candidate for other close range wireless applications such as remote controls for video game consoles, and interfaces for computer inputs such as keyboards and mice. Bluetooth is a relatively quick data transmission protocol comparable to User Datagram Protocol (UDP) as opposed to Transmission Control Protocol (TCP). This makes Bluetooth an ideal solution for voice and video data transmission. Bluetooth's topology is star-like, and is referred to as a piconet, which consists of a master and up to 7 slaves surrounding the master.

One interesting Bluetooth functionality that was given consideration was the collision avoidance algorithm that implements a Frequency-hopping spread spectrum (FHSS), in which its carrier frequency jumps around between 79 different frequency channels at an extremely rapid rate (1600 hops per second). Each new channel that it switches to is determined by a pseudorandom algorithm that has a repetition period of 23 hours. While this would have been generally sufficient in avoiding data collisions with all of the other devices that could potentially be operating somewhere in the 2.45 GHz frequency band, newer devices have implemented what is called Adaptive Frequency Hopping (AFH) to further improve the technology's collision avoidance capabilities. This is an algorithm that makes a note of which frequencies the Bluetooth devices experienced a collision, and updates the pseudorandom channel selection algorithm by not allowing those frequencies to be selected again.

One of the most useful benefits for the consideration of Bluetooth was that many personal computers have Bluetooth transceivers, which would have allowed for convenient and robust debugging opportunities. Since every Bluetooth device can operate as the master or the slave, the personal computer can be set as whichever mode it needs to be in, and the corresponding mode on the device under test could have been observed, allowing its functionality to be evaluated. Furthermore, when the personal computer connection transmits a signal, the receiving functionality of the device under test could have been observed. Conversely, the personal computer's connection could be in receiving mode, and the transmitting functionality of the device under test could be observed.

One of the biggest disadvantages of Bluetooth technology was its mere 10 meter range. Bluetooth was not designed for long range applications. Most houses are more than 20 meters across, making Bluetooth an impractical solution for the application under development. This was true unless, however, the network topology was to change from a piconet to what is called a scatternet. A scatternet is formed when one of the slaves from one piconet also serves as a master on an entirely different piconet; thus increasing the range of the network. This would have required much more complex software to manage the different networks simultaneously without interference. Furthermore, even with this solution, the network could still not have any gaps between two nodes larger than 10 meters.

2.7.2.2 IEEE 802.15.4 (ZigBee)

IEEE 802.15.4 specifies a low power, low cost, low rate, yet reliable set of standards for wireless communication networking. It was designed specifically to mitigate the issues of using existing wireless communication protocols, i.e.

Bluetooth, for sensor monitoring and control. It allows many more devices to be connected to a single network (about 18,450,000,000,000,000,000 total devices). It also has a typical range of 50 meters, but can transmit up to 500 meters if the environmental conditions are right. It has a hand-shake protocol which makes the data transmission reliable, and encryption capabilities to make the information transactions more secure.

As opposed to Bluetooth and other Wireless Personal Area Networking (WPAN) protocols, ZigBee operates as a Low-Rate Wireless Personal Area Network (LR-WPAN), as defined by IEEE 802.15.4. This means that ZigBee networks sacrifice some of their transmission speeds in exchange for a much more reliable, low powered and longer ranged transmission capabilities. Each ZigBee network must have at least one slightly higher power consuming node referred to as a Full Functioning Device (FFD) to operate as the Personal Area Network (PAN) Coordinator. Other nodes may either be Full Functioning Devices (FFDs) or Reduced Functioning Devices (RFDs) depending on whether the node needs to communicate with nodes other than the Personal Area Network (PAN) Coordinator, i.e. a peer to peer network.

The desired network topology implementation in the design for this project is a star topology. The central hub would then serve as the Personal Area Network (PAN) Coordinator. This being the case, the central hub would have to have a Full Functioning Device (FFD) node type. The outlet monitoring and control assemblies do not need to route any ZigBee packets, and only needs to communicate with the central hub. For this, either a Full Functioning Device (FFD) or a Reduced Functioning Device (RFD) could be used. However, since a Reduced Functioning Device (RFD) is cheaper in just about every aspect, that was the ideal node type for implementation in the outlet monitoring and control assemblies.

There were only two disadvantages to using ZigBee as opposed to other wireless protocols. The first disadvantage was the reduced data rate. This, however, was insignificant for communication with the outlet monitoring and control assemblies, being that it will be unnecessary to have to send large amounts of data very quickly. The second disadvantage, and much more severe, was the lack of debugging capabilities with personal computers and other earlier obtained equipment that were Bluetooth and Wi-Fi ready. ZigBee is a well-known protocol, however, which meant that there were sufficient development resources available online. Like many technologies though, ZigBee does have proprietary characteristics, which motivated us to initiate the development early to attenuate any potential delays.

2.7.2.3 IEEE 802.11 (Wi-Fi)

IEEE 802.11 is a standard wireless protocol that has many possibilities for design. It is the protocol that Wi-Fi brand networks operate on. Due to the fact that most homes already have Wi-Fi routers, the assumption has been made that whatever home this product will be installed in will already have existing Wi-Fi brand networks for internet access. This brings up two main issues: first, the system would have been designed so to not interfere significantly with these existing networks, and second is that the option would have manifested to bypass the central hub and send raw data directly to a remote server to be processed. Though we did not use the Wi-Fi protocol, it did allow for interesting alternatives in the event that whatever other chosen communication technology would have failed.

The 802.11 protocol has been subdivided into more specific protocols over the years. Of these more specific protocols, 802.11b and 802.11g are the most prevalent in modern day homes. 802.11b was an upgrade from the original 802.11 standard that allowed for a bandwidth of up to 11 Megabits per second (Mbps); a substantial increase from the original 2 Megabits per second network bandwidth. A few years later, the 802.11g standard was created which made further improvements on the technology. 802.11g can transmit data with a bandwidth of 54 Megabits per second, a significant increase from 802.11b. Another very convenient aspect about hardware that supports 802.11g data transmission is that it is backwards compatible with 802.11b devices. Therefore, it would have been advantageous to purchase an 802.11g network adapter, which could also be used with either an 802.11g network or an 802.11b network, whereas an 802.11b network device can only operate on an 802.11b network.

The range of the Wi-Fi network depends greatly on the router, network access devices, and other environmental factors. On the high end of this variation, one could have 120 feet of connectivity without purchasing high powered antennas. At the very least, a range of 50 to 65 feet could have been anticipated, in the worst of typical conditions. Wi-Fi can communicate without a line of sight, and can even transmit a signal through a wall.

The internet protocol used for websites most web sites is The Transmission Control Protocol (TCP). This protocol uses checksums in each header of each packet of information in order to verify that information is correct. Hence, if any of the data was corrupted by conflicting signals in the same exact frequency range, it would try again, until the packet goes through, or there was a timeout on the packet. This error detection algorithm guarantees that no corrupted information would be passed through the network. Wi-Fi networks, though they communicate on the same frequency as many other devices, rarely experience interference so bad that the communication is completely stalled, however if this would have happened, it could have been handled sufficiently at the application layer in a manner which would have been much better and easier to do than dealing with potentially corrupted information. The User Datagram Protocol (UDP) would not be necessary, because there would not be large amounts of information being transferred in short amounts of time.

One negative aspect that would have arisen had we implemented Wi-Fi devices for communication to and from sensors is that the costs for these parts are relatively high. Along with the fact that they are about twice as expensive as their ZigBee counterparts, they consume a lot more power as well. This made Wi-Fi modules unintuitive candidates for facilitating the communication between sensors installed with the intent of lowering power consumption. However, there are certain applications for which the Wi-Fi framework is quite ideal for. Such applications include image transmission with an IP camera.

2.8 Power Supply Considerations

There were many options to consider when designing the source of power for each network node in the project. Each node was required to have a supply voltage of 5 volts for the ATMega328, and 3.3 volts for the XBee module. Once a 5 volt power source was obtained for a given node, that voltage was simply regulated down to the 3.3 volts needed to power the XBee module. This has been implemented in the XBee mount reference design with a 78XXL voltage regulator, and either an MC33269ST-3, or a MC33269D-3 in the Ethernet port reference design. Only one of these options was needed in the final design, and was selected on a basis of lowest price at the time of PCB prototyping.

To obtain a 5 volt power source in the first place, each node had several different options; for instance, the Central Hub had the opportunity to be powered by the Ethernet cable running from the modem that it will be connected to, while the outlet monitor and controller could have potentially been powered directly by the hot lines running from the outlet. At the same time, however, any network node could have been powered either by a power adapter that plugs into a wall, or by a battery.

The first option that was available for the Central Hub was the Power over Ethernet module that was used in the development environment. This provided a simple design solution, due to the fact that once it was configured and installed, it required no further maintenance. However, there were a few disconcerting aspects of obtaining power through the Ethernet cable. Primarily, this would have made the Central Hub dependent on the router’s power supplying capabilities. Routers are not traditionally intended to serve as power supplies, so this could present issues. Secondly, the Power over Ethernet module used in development was not open sourced hardware, and the reference design used in development was incomplete for the Power over Ethernet module. Obviously, it was still capable of being accomplished, but it would have presented design problems that must have been solved, which would potentially have taken time away from designing other more interesting features.

Another option to provide power was by tapping directly into the 120V power lines running in the electrical outlets. This would have been convenient for the outlet monitor and controller, being that these devices already had power lines running through them. It could also have been replicated easily by any other node by adding a power cable to it. A complicated design process quickly presented itself, however. Trying to power discrete logic devices from a 120 volt alternating current power source, is like trying to fill a thimble with water from an opened fire hydrant. It can be done with a network of power transformers to isolate the circuit and high voltage capacitors to filter out the alternating current; in fact, most laptop chargers do exactly that. At the time of development, the task seemed daunting; however, after the completion of the project, it appears that these issues could have been easily dealt with.

To simplify the design, a power adapter like the laptop charger could have been implemented instead. This would have provided a reliable source of energy in the same fashion as the development environment. However, like in the situation above, a decent amount of energy would have been dissipated continuously, all day and all night. This would not have been consistent with the design motivation of building a smarter, more efficient home.

The last, but not least, solution considered was to use the power supplied by a battery or a series of batteries. The main drawback from using a battery or series of batteries to power any of the devices was that they will have to be replaced every so often (probably every couple of months, like a typical smoke detector). If and when the battery dies, however, the application layer would become aware of it, because it would stop receiving a signal from whichever device ran out of battery charge, and was be able to alert the user.

There were many benefits for using a battery, however. The first benefit was that the design was significantly simpler than the previously discussed methods. All that was needed is the battery source, and a voltage regulator to output a constant 5 volt signal for the board. This was very similar to the method used in the development environment, and was easily implemented in the same manner on each network node. Another advantage was that there is a smaller voltage drop from actual voltage source to 5 volt net voltage required by the circuitry. This means that the potential for energy dissipation was much lower.

2.9 Development Environment

Being that this project had no strings attached to a specific microprocessor; the convenience of choice as to which development environment became available. There were two options: the first was to build a development environment to evaluate a microcontroller for this project. This would have been risky and time consuming to accomplish being that none of the group members had experience specifically in doing this. Conveniently, there were pre-made development boards that already existed to evaluate various different microcontrollers. There were a few things to consider when choosing the development environment for any project.

The main issue for this project is whether or not the development board would meet the outlined specifications necessary to complete every required task. The most complex task that had to be performed by the microprocessor on the outlet monitoring and controlling unit was the wireless communication. Without this capability, the project would not work. Another significant capability needed on the development board was at least one analog to digital converter (ADC). This was needed to develop the board and software necessary for monitoring.

The second most important issue for this project with regard to the development environment was the amount of support that would have been available. If a development board is more obscure, less user friendly, has capabilities locked or not released for development, it could have overcomplicated the simplest of problems. In choosing a popular development board, the group hoped to accomplish the requirements of the project more easily. Not only did this decision help the project to be done on schedule, but new functionalities were explored with ease. Since a major goal for this project was scalability, this improved the chances of a more complex design with extra functionalities to be feasible.

The third most significant issue was cost. As the project was not able to raise any external financial support from sponsors or investors, every aspect of the project was funded by the group members. Being that the group members were all college students, there was not a large availability of finances to purchase large and costly development boards with excessive amounts of additional features. Additionally, the final design implementation was to be cost efficient for whatever complete final functionality it intended to serve. With that in mind, choosing an excessive development environment was unnecessary, wasteful, and even irresponsible.

The fourth most important issue was flexibility. This project needed to be a scalable as possible so that various functionalities could have been added or dropped at any time. It would have been advantageous for a board to have two analog to digital converters (ADC's) so that both current and voltage could have been monitored seamlessly (although an external multiplexor could have made one analog to digital converter suffice). Any wireless communication capabilities would have been beneficial; whether they were peripheral devices included on the development board itself, or available as a quick external add-on built specifically for the given development board. Zero crossing detection was an additional capability that would have allowed for calculation of power factor, which would have been a useful functionality to have as well.

2.9.1 PandaBoard

The PandaBoard was considered for use to help develop the central hub. With a seamlessly endless array of capabilities, this board definitely would have served the purpose of providing numerous scalability options. Also, being that it is a popular high-end development board, there appeared to be a large community of developers and projects to help with troubleshooting and support.

With a 1 gigahertz (GHz), multi-core processor, it would certainly have been capable of processing images. It even has an on board Wi-Fi and Bluetooth transceiver. There are multiple Universal Serial Bus (USB) ports and an Ethernet port which could come in handy for different possible design implementations and scalability options. It also has video capabilities like a High-Definition Multimedia Interface (HDMI), a Digital Visual Interface (DVI-D), and a liquid crystal display (LCD) expansion header. One of these video capabilities could be used for a monitor, or if that option is not chosen, it would be completely useless in the final design. It has audio in and out, which could be used for voice commands if the digital signal processing software libraries were present. It has a massive 1 Gigabyte low power Double Data Rate Rapid Access Memory (DDR RAM). Along with this memory is a full sized Secure Digital/MultiMediaCard (SD/MMC) cage for high speed and high capacity Secure Digital cards. While these memory capabilities would have been excessive for the basic design, they could have been useful if a more complex design was implemented with additional features.

The price for the basic PandaBoard was $174, and the upgraded version, The PandaBoard ES, was $182; a modest increase in cost for a modest increase in capabilities. The processor on the PandaBoard ES would have had a slightly faster processor, and could have encoded/decoded High Definition (1080p) video. These additional features are completely unnecessary for this project. Hence, only the $174 PandaBoard would be considered. However, while the board did provide a lot of scalability possibilities, many of the basic features were unnecessary. The price was relatively high for this project, but allowed for great scalability.

The PandaBoard is semi-open sourced; while the bill of materials, PCB and Gerber files are all available, the main processor on the board (OMAP4430/OMAP4460) was not purchasable in quantities of less than one thousand, meaning that it could not have been used in a prototype PCB layout. Also, Texas Instruments did not release the complete documentation for the processor to the public. One requirement for this project is to design and layout a specialized PCB. Given the circumstances, this would not have been possible to accomplish with ease.

2.9.2 BeagleBoard

The BeagleBoard was another development environment created by Texas Instruments. It was under consideration for use as the central hub as well. Like the PandaBoard, the BeagleBoard claimed to be open source hardware; however the same fine print issues existed like those of the PandaBoard. Hence, the central hub would essentially have had to have been to actually be the BeagleBoard itself.

There are three different BeagleBoards, all with less capabilities and cheaper than both PandaBoards. The BeagleBoard-xM is the highest end model of the BeagleBoards at $149. However, it would not come with Wi-Fi or Bluetooth; an essential capability for the central hub. At this point in the development environment consideration, purchasing the PandaBoard instead would have been cheaper, less time consuming, and less risky by not having to worry about compatibility issues. The original BeagleBoard was priced at $125, and the BeagleBone is priced at $89. However, without the peripherals for the additional scalability, purchasing an upper-mid range board seemed pointless when a much cheaper board could have been bought that would have done the same thing.

2.9.3 Texas Instruments' eZ430 product line

Texas Instruments has a collection of various wireless development tools offered by their eZ430 product line. In this MSP430 based product line, there are many different highly specialized boards, including the RF256x, RF2500, RF2500-SEH, and the RF1101T. The eZ430 RF256x is a Bluetooth evaluation kit. The eZ430 RF2500 is an evaluation tool for Texas Instruments' 2.4 GHz proprietary network protocol, SimpliciTI. The eZ430 RF2500-SEH is similar to the eZ430 RF2500 but includes a small (2.25x2.25) but efficient solar panel that be powered by indoor fluorescent lighting. The eZ430 RF1101T is another potential option provided by the product line which communicates over a 900 MHz frequency channel. This could have reduced interference with any Wi-Fi signals that may have been present.

The eZ430-RF2500 appeared to be the simplest and most practical solution for the project. The $49 development kit appeared easy to set up, would have come with a USB programming interface and emulator, providing quick and easy implementation. The kit would also have included two identical eZ430-RF2500Ts (wireless target boards), one eZ430-RF USB debugging interface, and one AAA battery pack that can power one of the target boards. While the evaluation kit seemed comprehensive, there also seemed to be limitations on flexibility, such as Texas Instruments' proprietary wireless communication stack, SimpliciTI, the lack of Eagle files to reference for final design, and Texas Instruments' proprietary Spy-Bi-Wire interface. With all these strings attached, it might have been hard to add functionalities using any products other than ones purchased from Texas Instruments.

2.9.4 Arduino

The most obvious advantage for choosing an Arduino development board for this project was the vast amounts of resources available for just about any project imaginable; online and for free. The community for Arduino is unreal. It has become an electronic staple in every upcoming hacker's diet. It is notoriously easy to use, and very versatile.

Besides making the project simpler to solve by being able to research countless other projects that have used this board, there were many other advantages as well. Everything on this board was open sourced. Software was very well documented and widely available and full datasheets are available for the Atmel microprocessors used on each of the different Arduino boards. Most importantly, the Eagle files were readily available for every model of Arduino that has ever been released. This allowed for a layout of a Printed Circuit Board (PCB) for the prototype based confidently on the reference design that had been rigorously tested up to that point.

2.9.5 XBee

XBee is a brand of radio frequency communication devices which operate using a modified ZigBee stack, based on the IEEE 802.15.4 communication standard. The devices are essentially antennas. This allows for any wires that would have been necessary to communicate between different nodes in the network to be eliminated.

XBee devices are well integrated into markets. There are many compatibility devices intended to interface XBee devices to whatever it needs to be interfaced with. These compatibility devices include XBee Explorers, XBee shields, and even XBee breakout boards. XBee Explorers come in many flavors; some that have Universal Serial Bus (USB) connectors, others with RS-232 connectors. Some XBee Explorers have voltage regulators, while others do not. There are also XBee Shields which are compatible with most Arduinos on the market. Finally, there are cheap XBee breakout boards if either cost became a severe issue, or the desire to do a lot of soldering became present. There were plenty of choices of XBee interfaces, which helped keep the design process easy to accomplish.

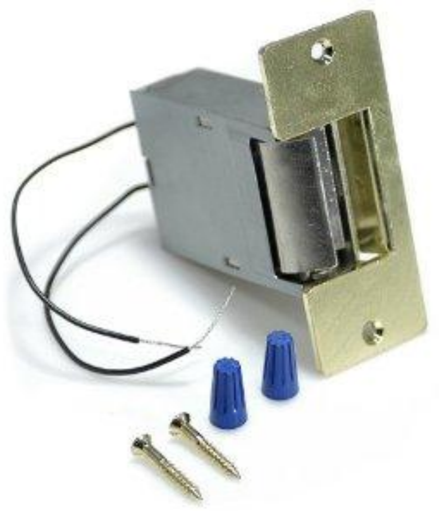
There were different classifications of XBee devices to consider. The first classification was whether the device is XBee Series 1 or XBee Series 2. Series 1 devices are basically easier to use. They would not have had to be configured before communication to another node. This made Series 1 devices ideal for simplifying the development process. Series 2 devices are much more advanced products that require setup, and could have allowed for optimization of things like power consumption. This made Series 2 devices superior choices for more advanced users who want more complex functionalities and more advanced customizability.

Another way that XBee devices are classified is whether the device is just a regular XBee, or an XBee Pro. The four main differences between the two are length, power consumption, range, and cost. The XBee Pro devices are slightly longer, consume more power, have a better range, and are more expensive than regular XBee devices. Hence, the only benefit from choosing an XBee Pro device over a regular XBee device would have been the range. The range on regular XBee devices is around 300 feet, and the range on XBee Pro's is on the magnitude of 1 mile. The 300 foot range achieved by the regular XBee modules is more than enough for residential applications, being that most homes are smaller than a football field, and the 1 mile range achieved by the XBee Pro modules is highly excessive for this project.

2.10 Door Strike

As part of the projects' home automation the group has decided to increase functionality by adding a door lock to the network. The door lock will be controlled by the website, but attached to the central hub, similar to the outlet and light switch. The project will allow a user to lock or unlock the door latch in the house at any given time. For this lock, there will be an electric door strike as seen in Figure 2-11. The group will use a mortise type door strike with a variable voltage of 3-6 volts and 1.25 – 1.8 amps. The mortise remains in a locked state until a voltage is applied. Once that voltage is applied the lock opens until the power source is cut off. The lock will only need to be supplied a voltage for a small period of time in order for it to work. The lock with communicate wirelessly with an android app or website through the hub, but will be connected using the positive and negative wires. The group will supply voltage to the lock via a 9 volt battery. Once the signal is sent from the website to the board a time delay will be introduced in order for the lock to remain unlocked for a few seconds.

Figure 2-11: Electric door strike



3 Design

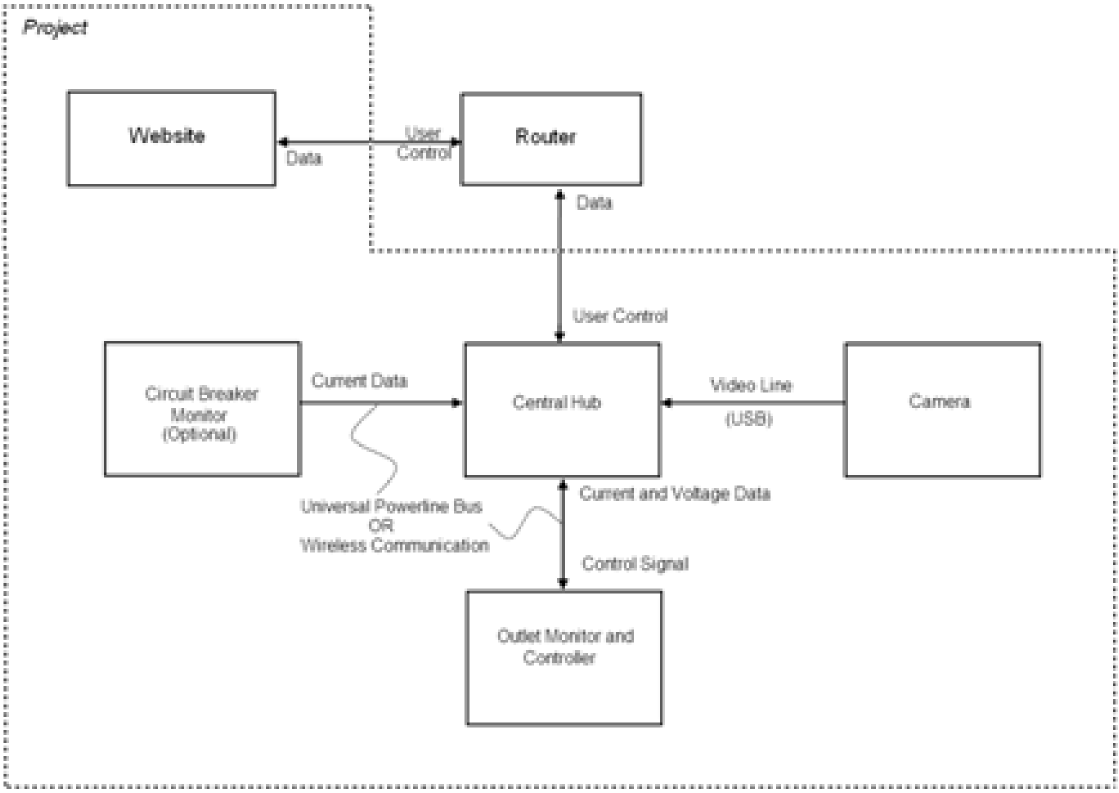
There were many different considerations that went into how precisely the system was to be arranged. For the most part, there were three essential components to the project: the camera which records the data outputted by the energy meter, the outlet circuit which allowed for user control and monitoring, and the website which provided an interface for the user to observe activity, and adjust settings. Other potential components included current clamps on various hot wires in the distribution board, a door lock control latch (electric strike), and an Android application. There were also theoretical applications which would have been convenient to have the option to implement as well, such as light switch and garage door monitoring and control.

There were many different possible arrangements of the components listed above, and many different ways to communicate between all of them. There were different network topologies to consider, each of which having their own advantages and disadvantages. Each network could have been arranged in a linear fashion, star configuration, or in a daisy chain. A daisy chain would only have been more useful than a linear or star network if the range of the communication technology being used is insufficient with regard to the distance between the furthest two nodes that need to communicate. It would have required each node to act as a repeater if necessary, which involves more advanced hardware and software. This approach was avoided.

The most intuitive approach for the network implementation of this project was to use a combination of star networks and linear bus networks. Star networks allocate each network node to a different input/output on the main processor. This was useful when communicating with different types of nodes such as the outlet monitoring and control unit and the door lock control unit. Linear networks use a single bus and an addressing protocol in the header of each packet of information that goes through transmission. This type of network was implemented for communication with nodes of the same type, such as multiple outlet monitoring and control units.

The original block diagram can be seen in Figure 3-1. Initially, it was thought that the central hub would receive the video feed, convert it to process discrete images every few seconds, and output the characters detected on the energy meter along with the other data that was obtained from the outlet monitoring and control unit. However, the idea was dropped quickly after further research. As can be seen in section 0, Universal Serial Bus (USB) communication technology was not sufficient for this task. Also, these types of processing capabilities required an advanced processor, and video decoding software libraries that may be proprietary or difficult to obtain. In theory, it was a simple concept, and it was most likely a completely possible design to implement, however more convenient, intuitive ways to complete the exact same functionality exist were explored so that the optional features could be investigated and added to make the design more well-rounded and impressive.

Figure 3-1: Original block diagram

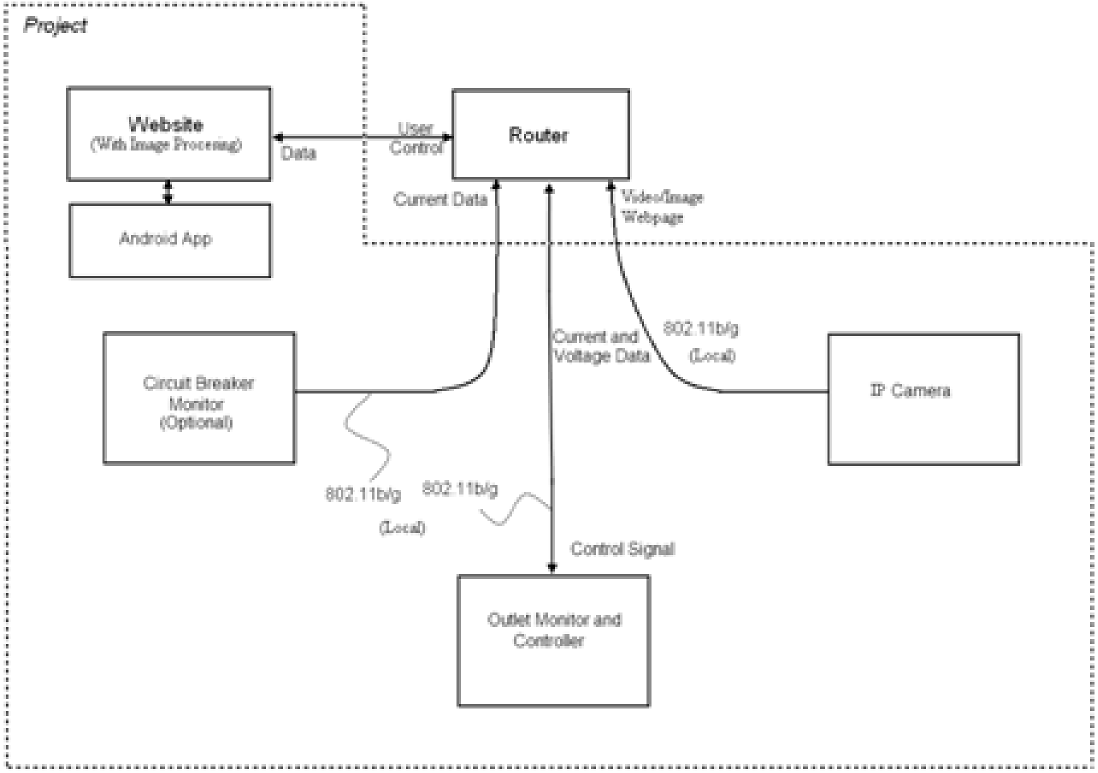


While researching about how to solve the image obtaining and processing conundrum encountered in the initial design, Internet Protocol (IP) cameras were soon discovered. These alleviated issues associated with wired communication protocols such as Universal Serial Bus (USB). They provided an intuitive, convenient, and straightforward methodology for obtaining images to be processed by software.

After further consideration of the 802.11 protocol, it was realized that the entire project could now be conceivably done without a central hub. This network implementation could be seen as not necessarily removing the central hub; it only moves it to a remote server with the help of the internet. Figure 3-2 is a direct implementation of this methodology. It can be seen how convenient and simple the design is. 802.11 is a convenient technology that could have allowed for a practically endless amount of nodes that can be moved with ease at any time, communicate through walls, and communicate over relatively long distances.

Just like the paradigm shift from analog to digital that has taken place over the last few decades, a new paradigm shift can be seen which allows for memory storage and application execution to take place “In the cloud.” This new strategy allows for much simpler hardware and software on client devices that use these new cloud computing capabilities.

Figure 3-2: Block diagram with no central hub

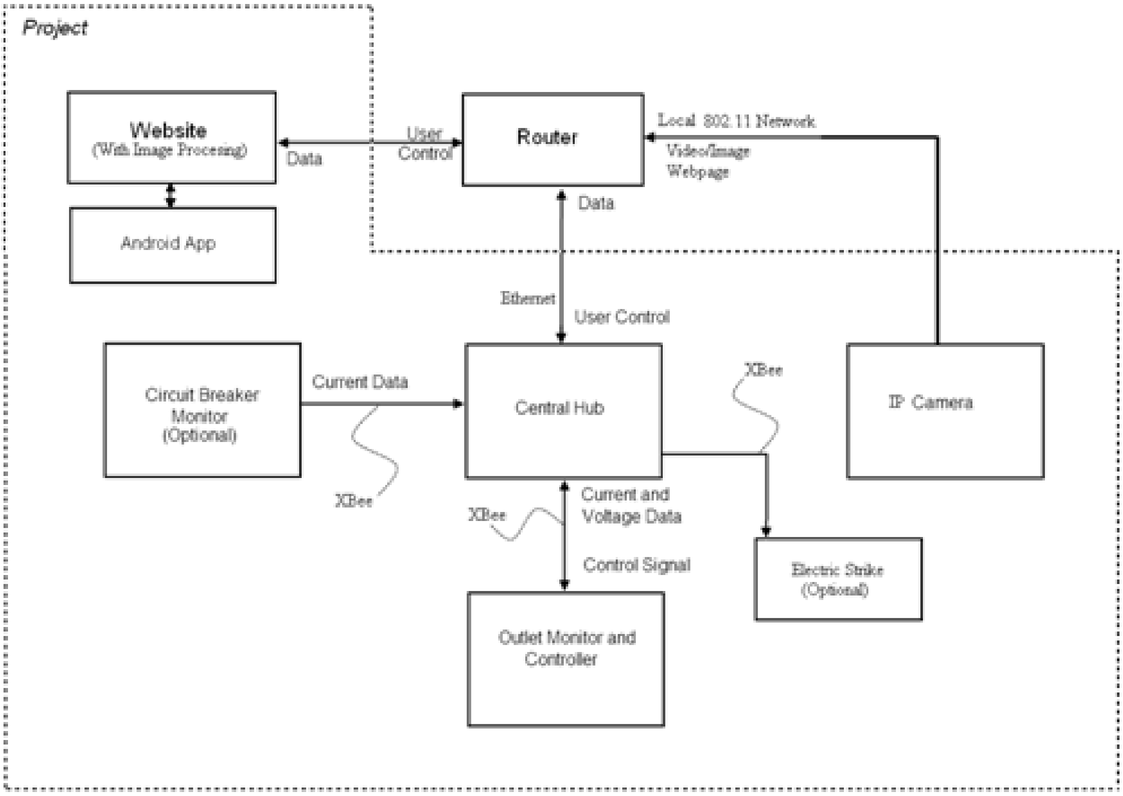


However, a few major issues remain with this design. One of these issues is that all flexibility vanishes. It would no longer have been so easy to simply add the remote door locking mechanism, unless that too has an IP address. It also would have relied heavily on the network, and for any amount of time that the network would be down, the data would be unrecoverable. The final, possibly most significant concern is that the project would have been too simple in terms of electrical engineering design problems required.

The central hub was therefore included once again, along with the use of an IP camera. There were two major options as to how the signal would be obtained from the IP camera. The two most sensible options were to send the 802.11 signal to the same router that the central hub is connected to, and connecting directly to the central hub through a completely separate Wi-Fi network. The latter would have necessitated an 802.11 communications port on the central hub, software to handle the wireless internet protocol, and a more sophisticated processor to handle all of this and the incoming images. With this in mind, it made much more sense to have the IP camera send signals directly to the router, which was designed to forward information wherever it is needed. With this implementation, the image could be processed online, or on the central hub.

However, it was still ideal to process the incoming images on the same remote web server which was hosting the site where the information was accessed and edited for the system so that additional storage and processing capabilities are not required on the central hub. This design implementation can be seen in **Error! Reference source not found.**.

Figure 3-3: Final block diagram



Another feature that can be seen in this design is the addition of the Electric Strike, a wirelessly controlled deadbolt assembly. This component was convenient to access through the internet or a cell phone application in case the physical keys were lost. In order to implement this, some kind of encryption scheme could have been employed so that not every person who can communicate in the 2.4 GHz frequency range has entry access to the home which the system is installed on.

Since construction of a Wi-Fi router was not included in this project, yet its functionality is. Therefore, it is shown in the block diagram, but not within the dashed perimeter labeled project. There are technically other network devices involved in the transmission of data to the website from the router, but those are omitted from the block diagram entirely, as they are also not part of the project either.

The Circuit Breaker Monitor and the Electric Strike are both labeled as optional entities on in the system. In the final design, only the Electric Strike was implemented, and the Circuit Breaker Monitor was omitted. Both of these components only needed information to flow in one direction. The Circuit Breaker Monitor only needed to send current data to the central hub from the site under surveillance, and the Electric Strike only needed to receive information regarding whether to lock or unlock the deadbolt.

The Android Application can be seen as only talking to the website. However, this interface could have been used to provide a seamless interface with the hardware located at the house in which the system is installed. Essentially, the web site could serve as a link in a daisy-chain topology between the Android Application, and the hardware it was intended to control. This would necessitate sophisticated software operation to provide information to be accessed by the Android Application without interrupting the web site application, which the user may also use. Also, the website must be able to receive data from the Android Application, and forward it to the system hardware without interruption.

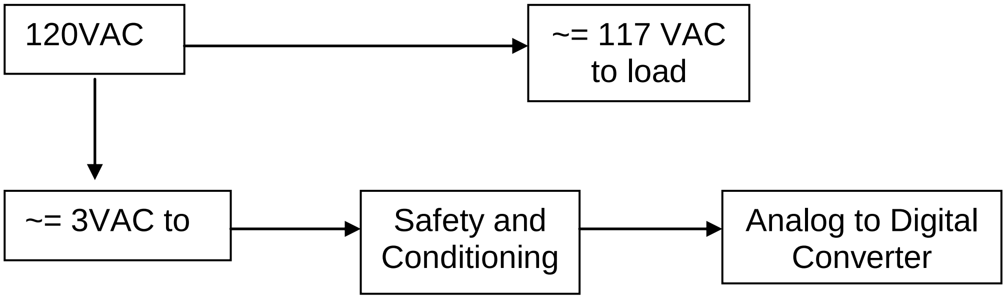
In conclusion of the design, multiple different specific design implementations were considered before choosing the optimal block diagram shown in **Error! Reference source not found.**. In this design the three essential project components were implemented; the image was obtained with an IP camera, the Outlet Monitoring and Control Unit communicates with the Central Hub through an XBee network, and the website was accessed through an existing router, which was extraneous to the project. Additionally, there could be an android application communicating only with the website, and an Electric Strike communicating directly to the Central Hub. Incidentally, with all of these network connections, the three major different types of network topologies were implemented. A linear network was implemented with the XBee network serving as the bus for which the Outlet Monitoring and Control Unit and either the Circuit Breaker Current Monitor, or Electric Strike are all connected to and have an address on. The Central Hub was also thought of being the center of a star network, communicating to the devices on the XBee network and the router through a separate Ethernet port. Finally, a daisy chain network could be seen connecting the Android Application to the rest of the system through the website.

3.1 Outlet Subsystem Design

3.1.1 Voltage Monitor Circuit Design

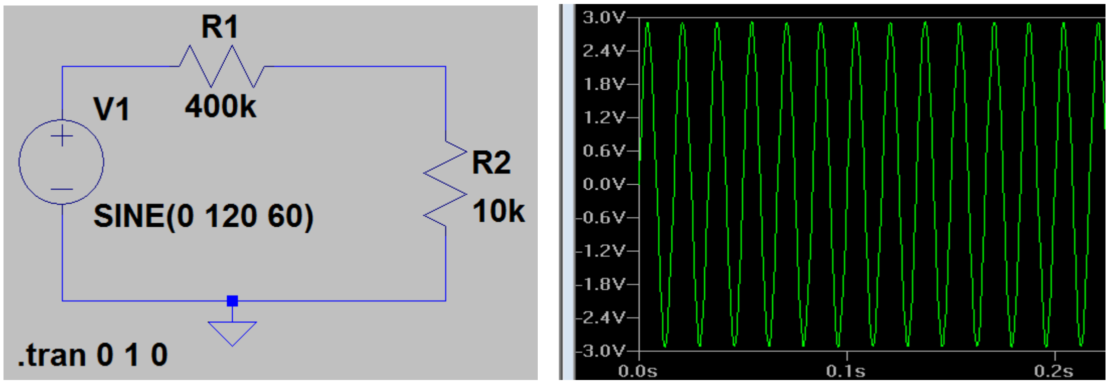
Now that the group has decided on directly measuring the voltage seen by the outlet monitoring device, it is time to start designing the circuit to obtain the voltage. It is important to note that the group will be placing control to the outlet before the signal is conditioned, but for clarity this will be covered in a following section. Now, since they are going to be converting this analog voltage to a digital bit representation, they must first bring this voltage down to a safe level to be seen by the ATD's (while still allowing the device plugged into the outlet to have as close to a true value of the actual voltage transmitted). Now in the range of the ATD's, the group will want to place a potentiometer before the ATD to allow for calibration of the signal (based on a known power requirement, such as a 100W light bulb), and add a few other elements to ensure the safety of the device, and more importantly: the microcontroller. Although this appears to be an involved process, the group will tackle each step one-at-a-time (with the block diagram shown in Figure 3-4), so that they end with a desired output to pass to the ATD.

Figure 3-4: Block Diagram of Signal Conditioning



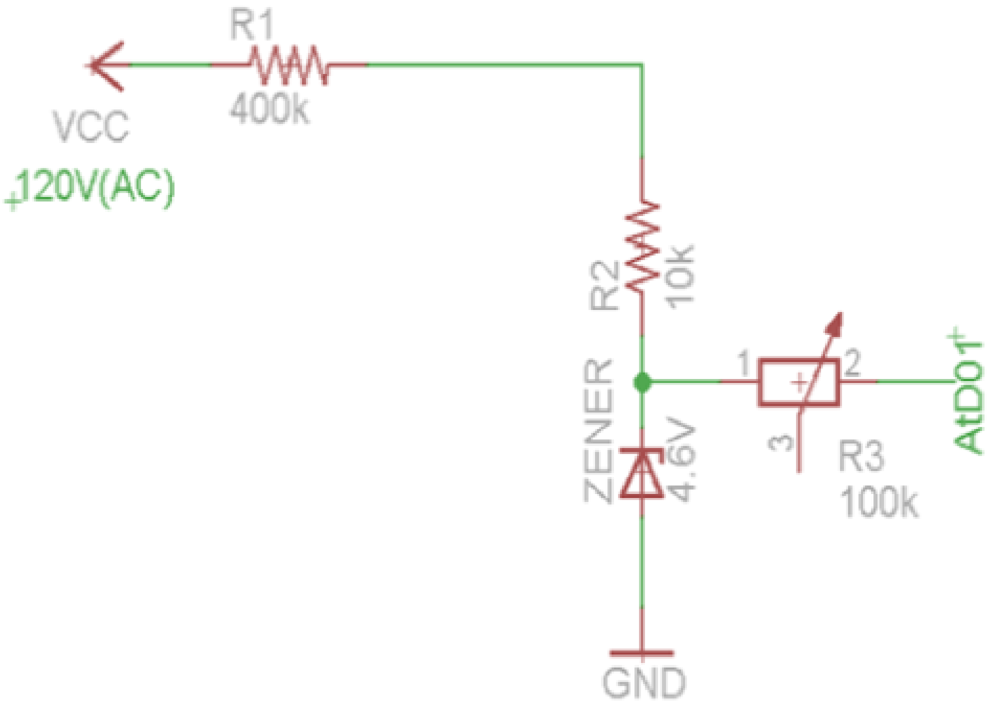
The next step in the group's quest to condition the high voltage signal is to use a simple voltage divider to send roughly 117V to the device, while sending an average of 3V to the input of the ATD. This voltage was chosen based on the input range of the ATD they will be using to perform the measurement. This is a relatively basic circuit, thus it needs no discussion as to how it will be designed (though the schematic and outputs are shown in Figure 3-5, as a proof of concept and design). It is important to note that they must choose resistors capable of handling this high of voltage, although most resistors today are capable, since safety is always a concern. Due to the simple nature of the voltage division step, the group will continue to discussion of the next step in signal conditioning: the envelope detector.

Figure 3-5: Voltage divider schematic and output



Now that the signal has been conditioned, the group must now put the finishing touches on the group's circuit used for obtaining the voltage at the outlet. Because of the fact that microcontrollers (and other "computers") used today are extremely sensitive to voltage, they must put in place something to protect the microcontroller. Since the voltage range given by the A/D is 0-5V (with the absolute max coming at 6V) the group has decided to put in place a "surge" protecting device: a Zener diode. They will choose this Zener to have a break down voltage as near to 5 volts as they can get. They will then place this device in series with the 10k resistor in the voltage divider, and will then place the load (the AtD input of the Atmega) across the Zener. When the DC voltage nears the upper limit of the range (5V) the Zener will basically push the higher voltage to ground. Because of the placement of the average DC output (under 3V @120AC), this Zener basically gives the device protection from extremely high voltages that would surely fry the Atmega's Input pin (final schematic shown in Figure 3-6). This simple circuit element also increases the linearity of the output due to the nature of the Zener Diode. Also, in series with the load, the group has chosen to place a 100K potentiometer. This potentiometer allows the group to the freedom to calibrate the output voltage seen by the Atmega to a base voltage that they choose (to +/- .5 volts).

Figure 3-6: Final Voltage Calculation Circuit



Once the signal has been conditioned for safety, the group is now faced with two options as to how they will be calculating an RMS value of the voltage. One way is by use of an envelope detector. Another way is to use various software techniques to sample the voltage at a certain rate and use an algorithm to determine a fairly precise reading of the voltage seen by the device. The latter method represents the cleanest and most technologically advanced way to accomplish the given task, and thus will discuss the algorithm later in the implementation section.

3.1.2 Current Monitor Circuit Design

Now that the voltage monitoring circuit has been designed, the group will move to the final piece to the power monitoring design problem: Current monitoring. Because of the reasons stated earlier, along with the fact that large amount of current are extremely dangerous, they have chosen to use a Hall Effect current sensor. First it must be noted that it is the preference of the group that led the group to the use of a 9 volt power supply to power the outlet monitor. This was chosen because of the size, convenience, and price of a standard 9 volt battery. Now, as they saw from the data sheet, the Hall Effect current sensor needs a voltage between 6-12 volts to operate. It is also important to note that since this device gives a voltage output, they must take into consideration the superimposing of the voltage input signal onto the voltage output signal. This superimposition could cause problems with the accuracy of the device, and thus have chosen to give it a clean 7 volts. In order to decouple the noise (as much as possible), they decided to use a 7 volt voltage regulator along with a couple capacitors and a diode for good practice (shown in Figure 3-7).Since the Current sensor is capable of producing between 0-7Volts as an output, and the voltage range of the Atd’s on the Atmega is between 0-5 V, they must use a voltage divider. Inside of this voltage divider, they have chosen to use a potentiometer to allow them to finely tune the voltage seen by the AtD.

Now that the specific components of the outlet monitor have been created, they are ready to combine them with the processor they will be using. It is important to note that the price list for this schematic is around what the group has budgeted for the outlet monitoring circuit (shown in Table 3-1 with development cost factored in). For the individual schematics, and the total schematic, they have chosen to use Eagle Cad so as to grant them the opportunity to ultimately design and layout boards to be used in the demo of the group’s project. The group feels that this design is critical to the completion of the group’s Electrical and Computer Engineering degrees because of the need to know how when they ultimately reach industry. Before they journey down the path of choosing which software to ultimately get the group’s PCB’s printed, they will show an excerpt of how they will complete the outlet monitoring circuit while omitting the placement of the relay so as to avoid complications associated with creating complex parts in Eagle (shown in Figure 3-8).

Figure 3-7: Current Sensor Power Supply and Output Schematic

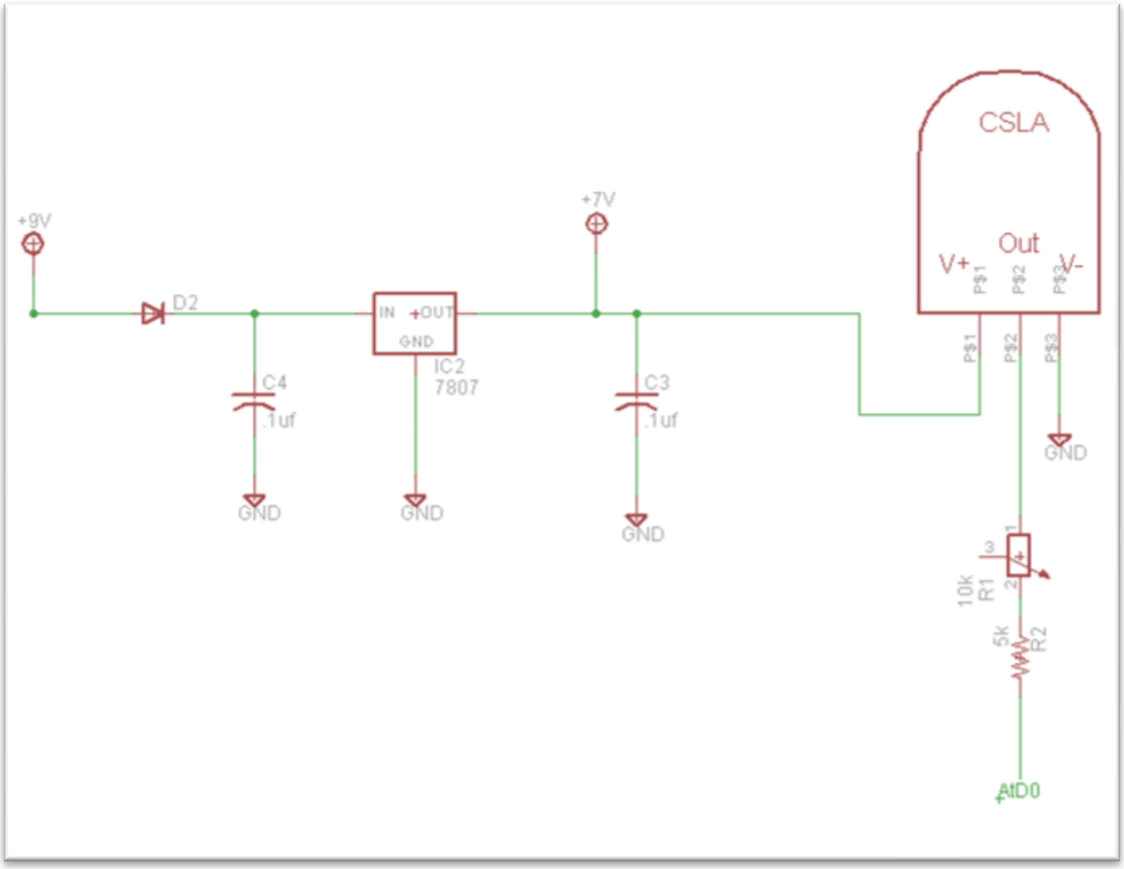
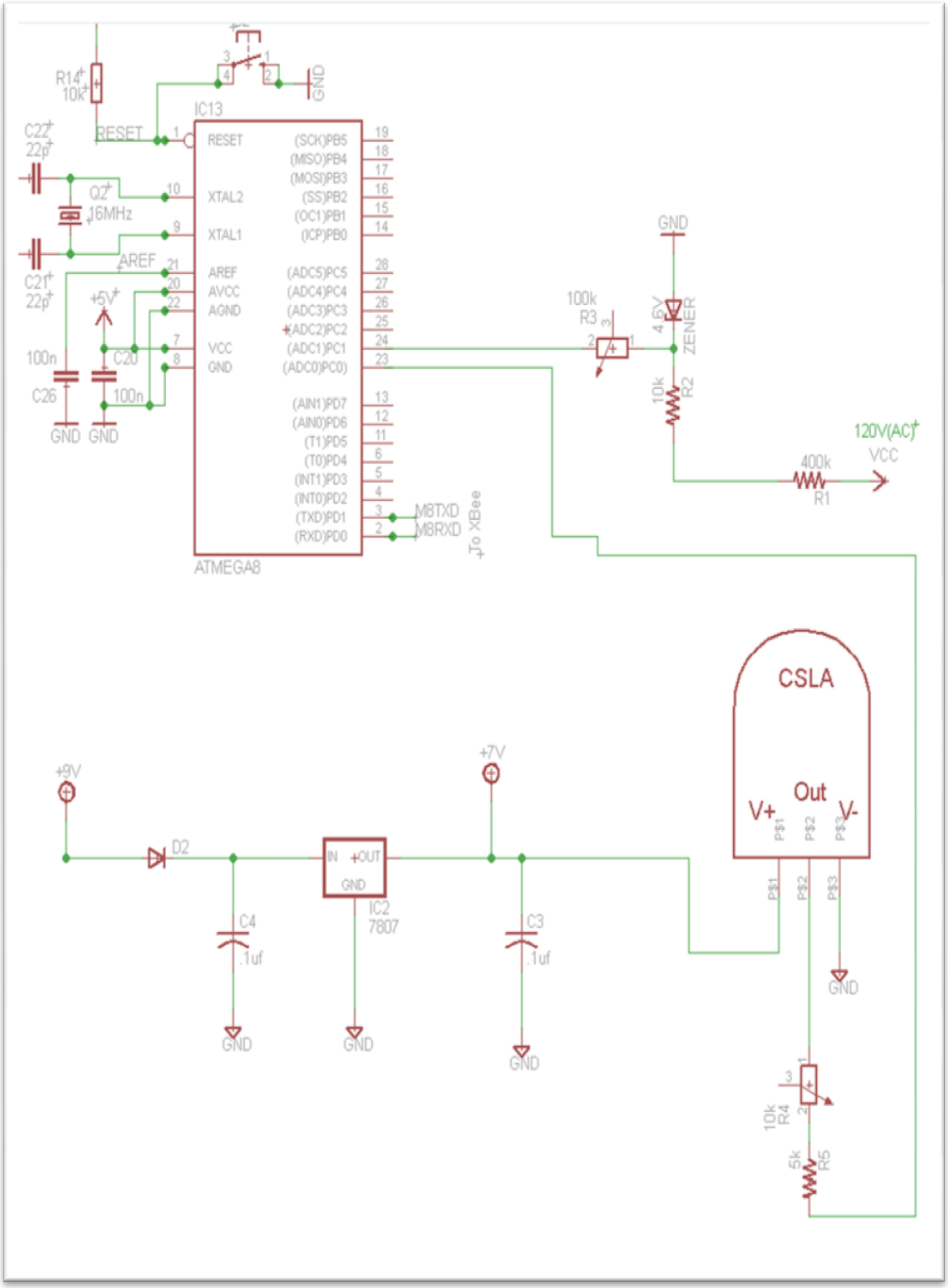


Table 3-1: Cost of Outlet Monitor

Component	**Quantity**	**Cost per unit($)**	**Total($)**
Voltage Regulator	2	1.50	3.00
Honeywell CSLA2CD Current Sensor	1	18.50	18.50
Resistors	3	.10	.30
Potentiometers	2	2.00	4.00
9Volt Battery	1	3.00	3.00
Atmega Processor	1	2.88	2.88
Wireless Comm	1	~20.00	20.00

Development	-	40.00	40.00
Relay	1	3.00	3.00
Wires and Housing	1	10.00	10.00
Transistor	1	2.00	2.00
TOTAL	14		106.68

Figure 3-8: Completed connection of Outlet monitoring Sensors



3.1.3 Outlet Software Design

The group decided to forgo turning the voltage output given from the smaller resistor in the voltage divider into a DC value (via an envelope detector), they then began investigating how they could handle the voltage calculation necessary to obtain a fairly accurate voltage (via software). They first had to determine that the ATD's on the processor they will be using (Atmega328) have a resolution of 10 bits (or 1024 different values) and a voltage range of 0-5 volts. Simple math tells a person that this means each step (from 0-5V) equals 5/1024 volts. This step means the device will be able to determine a difference in voltage as long as the delta is .005 or more. Since the group chose resistors that effectively place roughly 3 volts at the input of the Atmega's analog to digital pin this means that the value is received from sampling the analog to digital input will give the user an extremely precise value.

Since they have considered the precision of the results, the group must now consider how they will handle the time dependant nature of an AC signal. From the data sheet of the Atmega it is apparent that the processor is capable of reading a voltage from any of the analog to digital converter pins every 100 micro seconds (or every .0001 seconds). Since the signal the device will be reading has a frequency of 60 hz, the group members know that it will take 16.666ms to complete one wave, thus placing 8.33 ms between "zero crossing" points.

Also stated in the data sheet is the fact that when the input to any of the analog input pins drops below zero volts, the pin will clamp until a positive voltage returns. This clamping of the voltage makes the group's life much easier because of the fact they are able to deal with only positive values of the sine wave, and can thus focus on the algorithm for obtaining the peak RMS voltage seen by the outlet. Because of the fact that, no matter where in the time domain the device chooses to begin sampling, the device will return to the first sampled value every 16.66ms, the device must only sample for that amount of time to receive a precise value for the RMS voltage. Now that the group has discussed both the features of the analog input pin on the Atmega and the characteristics of a 60 hz signal, they can now specifically talk about how they will compute the voltage seen by the device.

Since the group will be using Arduino development boards to program the processor, they must first research how it is the operator can read a pin via the compiler specific to the Arduino. From the Arduino website (licensed to public thus no need for a reference), the group sees that an analog pin read can be accomplished with the process shown in Figure 3-9.

Now that there is a loop with which the system is able to sample the analog input, it can simply just have the loop run 167 times (giving a total sample time of 17 ms). Note in Figure 3-10, every time the loop is run, if the current value is larger than the last than it will be stored (thus giving us an accurate RMS max).

Figure 3-9: Read an Analog Pin

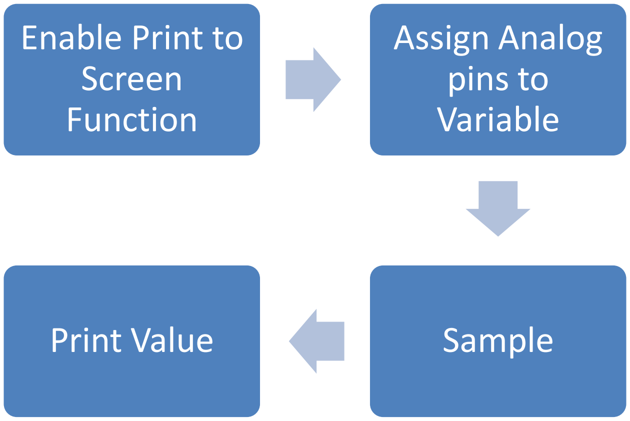
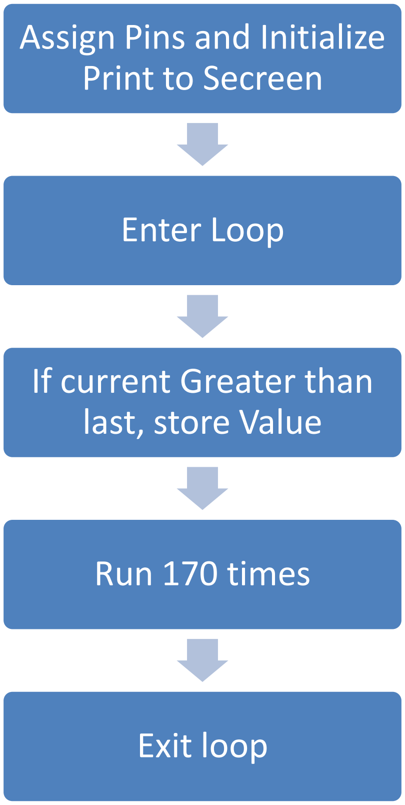

Enable Print to Screen Function
Assign Analog pins to Variable
Sample
Print Value

Figure 3-10: Steps to obtain Max RMS voltage


Assign Pins and Initialize Print to Secreen
Enter Loop
If current Greater than last, store Value
Run 170 times
Exit loop

As you can see, the circuit will be attempting to run a basic bit search in which the software will keep running (up to 167 times) and will be placing the maximum value in the integer RMSVoltage for storage. Now that they have an accurate RMS voltage (between 0-5 volts), they must do some multiplication to get it to be representative of the voltage seen by the outlet. This can be accomplished by just simply dividing RMSVoltage by the ratio of the resistor from which they are obtaining the voltage (.025). After this division, they will have successfully obtained the true RMS voltage seen by the device!

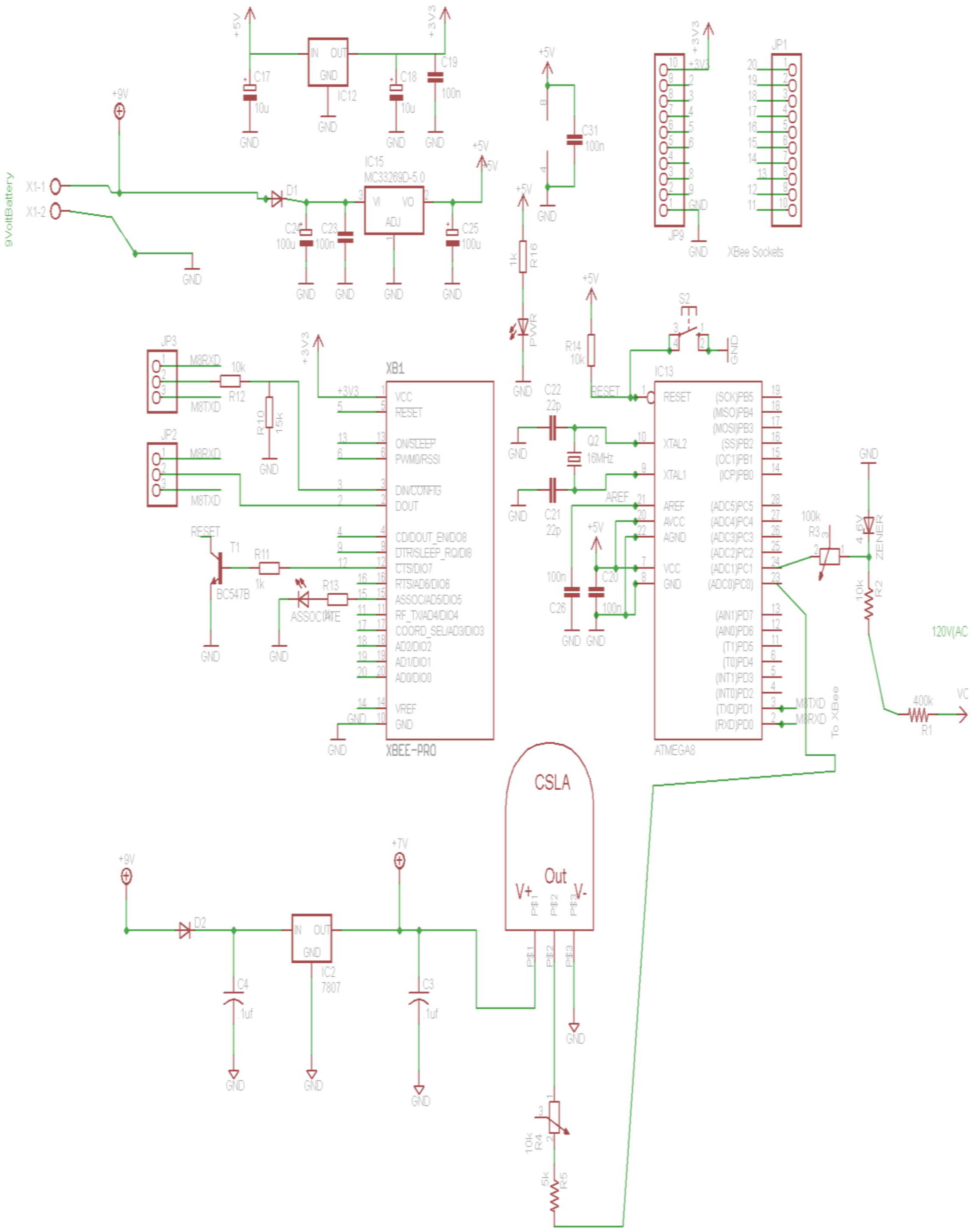
With regards to measuring the voltage output from the current sensor, the software coding needed is a bit simpler than the code need to obtain a voltage seen by the outlet. Since the output of the sensor is a linear voltage somewhere between 0 and 7 volts, they must first start with the potentiometer turned all of the way up to 10000 ohms. This is done so that they can see what voltage corresponds to a high current item (such as a blow dryer) without going over the 5 volt limit of the ATDs. Once they determine the output is within the limit, they can now work on figuring out the values of the voltages that correspond to the amount of current passing through the wire. This can be done by unplugging the device to be measured (so that they will know the amount of current running through the wire is equal to zero) and obtaining the voltage seen by the ATD. They can then pick a device (whose current is known) and plug it in, giving a second data point. Now that the group members have two data points, since the output is linear, they will have both the Y asymptote, along with the slope of the line. This can then be plugged into the equation of a line (y = mx+b) to determine a formula for obtaining the current of a device!

Once they have obtained both the current and voltage seen by the device, they can now obtain the power used. This calculation is simple: P = V*I. After the power is obtained, the device must transmit the integer representative of the power used by the device to the central hub in order for it to be uploaded to the web.

3.1.4 Power Control Design

Now that the monitoring circuit has been designed (final schematic including communication shield but without power supply shown in Figure 3-11) and the relay has been picked, the group must now research the specifics on how they will be gaining a control over the outlet. For all intensive purposes of this class project, the group will only be tackling a one outlet receptacle monitoring system because of the fact that adding another circuit does not prove the group’s concept any more than a one outlet plug will, as it is simply just double the part count and the exact same circuit. It has been the group’s vision to provide a free standing outlet that someone can simply just plug into the wall without having to change any of the existing circuitry in the home. This means that the group’s device must have a single plug receptacle that has been tied (appropriately and safely) to a 3 prong plug so as to provide the user with a “plug and play” device. It is worth noting that the group members have chosen to design the monitoring system for a 20 Ampere circuit because of the fact most 120v housing outlets have 20A fuses connected to them, meaning that the outlet can be used without worry of burning the house down. Now, the group members feel that the simplest way to accomplish this is by purchasing a three prong extension cord (one receptacle) and simply cutting it in half, exposing the live, neutral and ground line. Once this is done, they plan to attach the wires coming from the receptacle

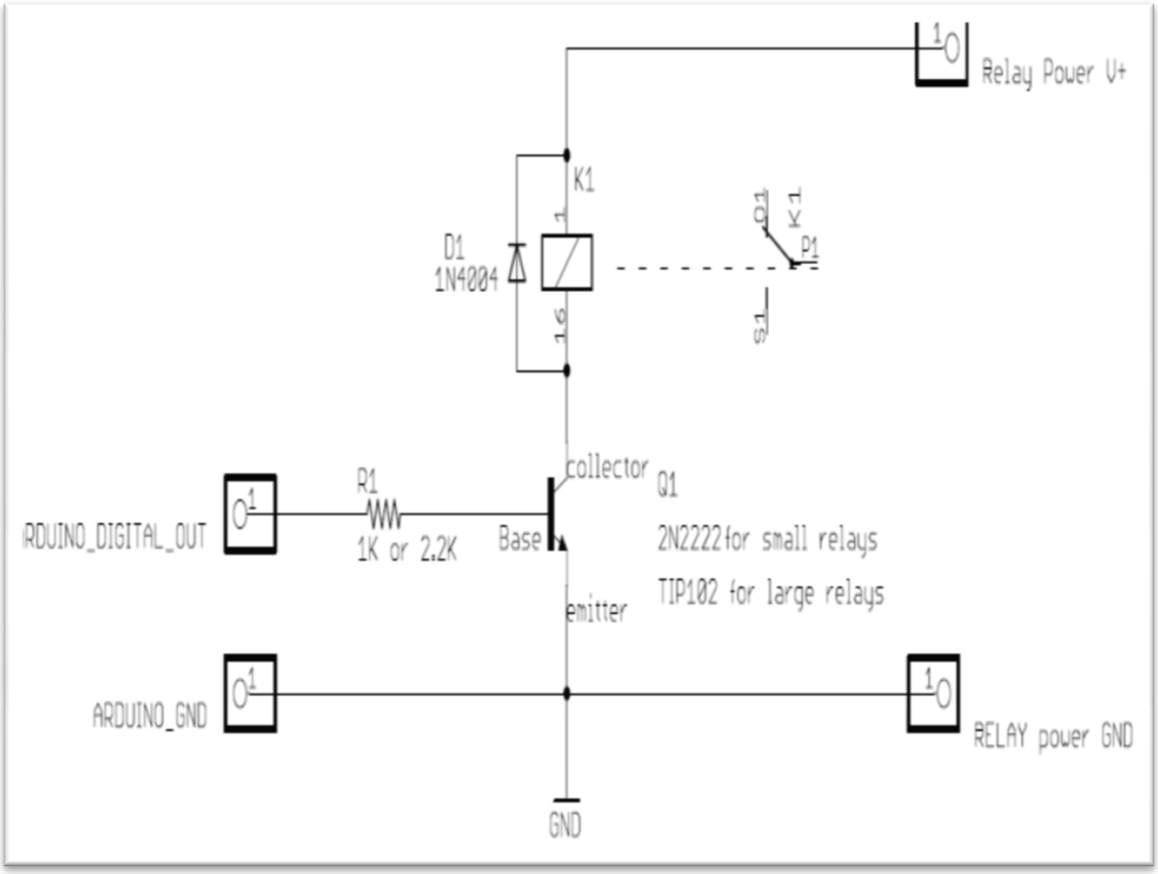
Figure 3-11: Final central hub schematic without power supply



to the normally closed side of the relay, with the normally open side being tied to ground. Before the group can move on, they must first discuss the specifics of the small circuit that will grant control of the outlet.

Using the relay discussed when designing the outlet monitoring device, the group will be implementing the design shown in Figure 3-12. This design effectively protects the relay from the surge associated with inductive loads by use of a diode to slow the spike in voltage to a level the relay is able to handle (without the dissipation of power associated with using a large resistor). They will be then be using a transistor as a switch to allow the switching of the relay from the small current provided by the output pin of the Atmega processor, thus granting the user control over the outlet. Since the relay has a 6 volt DC coil, they are granted numerous options to provide power to it but will most likely pull from the already regulated voltage given to the current sensor. All in all, the group members feel that this design is the most efficient and simplest way to grant the user control of the outlet, while also allowing for the state to be monitored and reported to the web (if they choose), and as such they will plan to implement a similar design.

Figure 3-12: Interfacing Relay to Atmega (Reprinted from http://www.arduino.cc/)

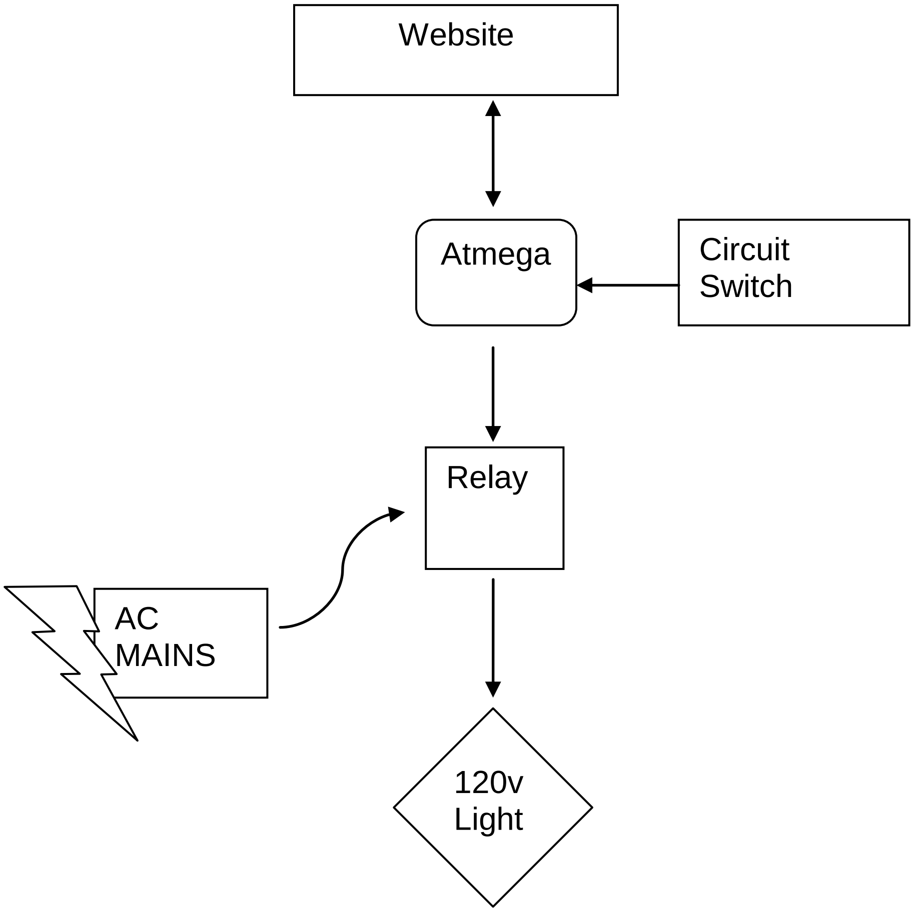


3.2 Light Control Design

The last device the group must design is the final scalability option: the Digital and Analog light switch. The idea behind this switch is that the user should be able to push a physical button, and have the light change states, while still being able to change the state via a mobile device. The group members feel that this would be a great way to showcase the group's project in that they would effectively be granting the user full control over their home (without being tied to only web control). This add-on allows the group to create an interface on the web that the user would be able to go to and place their lights on a timer (say for their front porch lights), or turn them off to save power! It's important to note that this will be the last device they will be tackling when it comes time to begin building and testing, but they feel that due to the well documented nature of this design problem that they should be able to successfully complete this challenge and thus they will begin discussing the design.

When the group members started thinking of ways to design this device they envisioned building a device with a male end (plug connection) on one side of the enclosure, with a light switch on the front, and with wires coming out of the top of the enclosure going to a light socket (design shown in Figure 3-13). This type of connection allows the group to demo this at the group's final presentation without having to modify an existing structure at UCF. Now that the vision of the device has been discussed, they will discuss in more detail the physical components need to make this vision a reality.

Figure 3-13: Light Control Block Diagram



This device will be similar to the outlet monitoring device they have built, in that the wires going to the light will be connected to a relay, which will have the same surge protecting diode along with the pin of the Atmega controlling a transistor (used as a switch). Although a similar design will be used, the group will need to pick a physical switch to allow the user to have manual control over the light. When it comes to the switching of the light, the main design problem the group will have is that of allowing the user to toggle the light via a manual switch, or by way of digitally controlling the light. This effectively rules out the use of a traditional light switch because of the fact that light switches are meant to be free standing physical switches meaning that they need no microprocessor to operate. This is great for industrial buildings and residential homes because you can simply plug them into to the wiring of a light and have immediate control without the need of other technology or programming experience. Since they want to be able to also toggle the switch using a mobile device, they cannot have a physical switch in the way because the group would have no way to reconnect the circuit allowing the light bulb to light. This leaves only one option: a manual circuit switch.

As the group has shown, a manual switch must be used so that they will be able to always have control of the outlet. In order to accomplish this, they will be using a microcontroller (the Atmega), a relay and a switch. For control of the 15 amp lighting circuit they will still be wiring the light directly to the relay, which they will still be controlling by a voltage (seen by a transistor) placed on the output pin of the Atmega. Although similar in nature, the group must alter the design slightly. They plan to connect one side of the pin to an input of the Atmega, with the other side connected to a resistor and a small voltage source (shown in Figure 3-14). This will allow the group to use software (example code flow chart shown in Figure 3-15) to monitor the state (high or low) seen by the input pin of the Atmega simply because when the switch is closed the voltage will be seen by the input pin (and when the switch is open the input pin will go low). Along with monitoring the state of the switch, the group must still have the Atmega looking for a user command from the web to change the state of the relay, thus allowing the group multiple ways to toggle the light! One benefit of the simplicity of design in this case allows them the option to either directly solder the circuit together, or create a schematic in Eagle (and layout the board) to ultimately allow for them to have the board professionally manufactured to minimize overall package size, and increase productivity (if the group decided to take the smart house devices to market).

Figure 3-14: Example Schematic of interfacing Button with Atmega

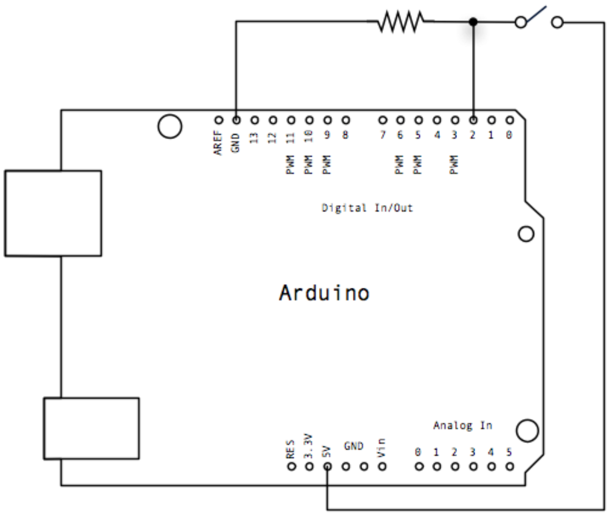
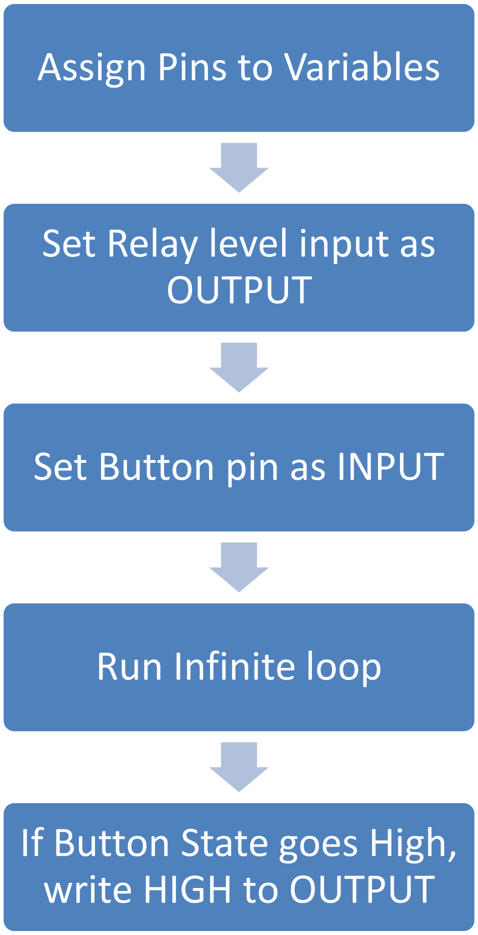


Figure 3-15: Example Code Flow Chart to monitor switch


Assign Pins to Variables
Set Relay level input as OUTPUT
Set Button pin as INPUT
Run Infinite loop
If Button State goes High, write HIGH to OUTPUT

3.3 Website Design

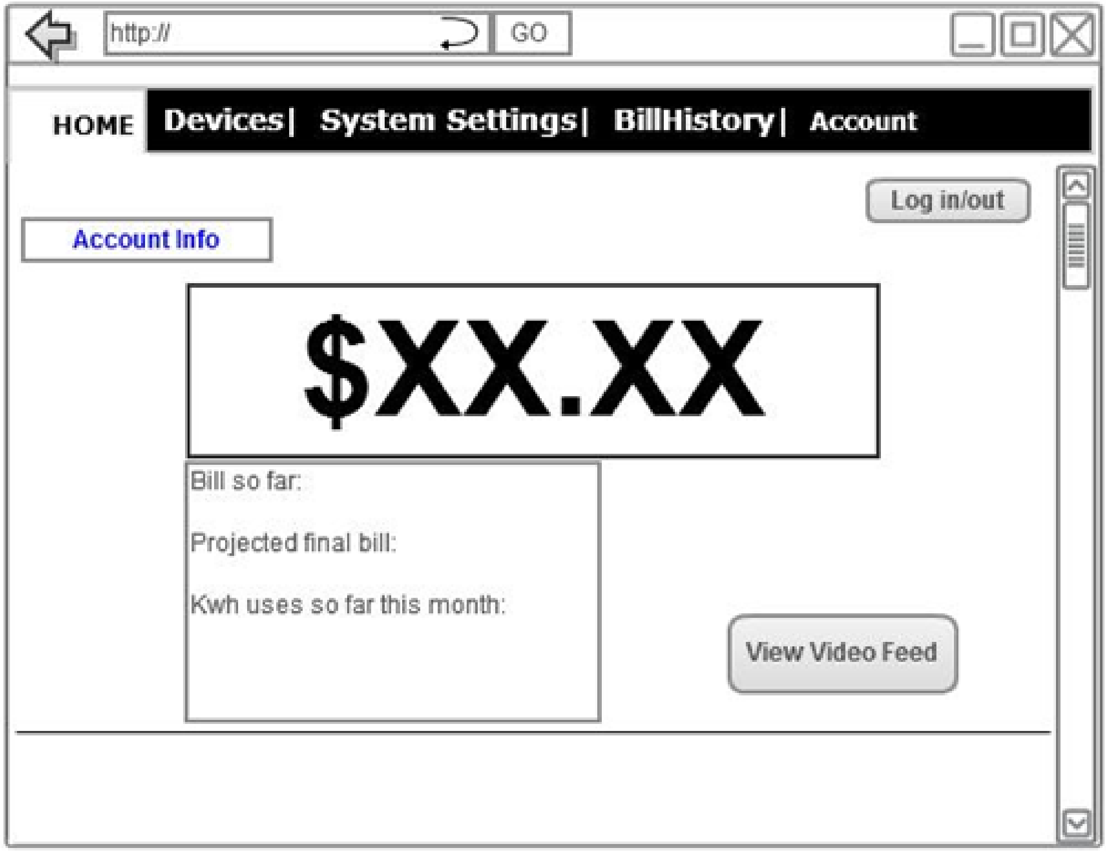
The website is one of the most important parts of the project, because it is where the user will actually interact with the system. Without the website, the system is rendered useless, because a user cannot see his or her bill and power usage, nor can they control the devices in their home. With this in mind, there was a lot of effort put into the website.

The website defines the functionality of the system. Anything that does not appear in some form on the web site might as well not even be part of the system. So, the user has to have some sort of display and/or control for each component of the system. Because the system uses a website that is hosted on a computer that is not necessarily itself part of the system, an end user will not be using their own computer as a web server, but connects to the project's web server. This means that if this system were developed for sale, multiple users would be connecting to the same server, and in order to display the correct information for each user, we needed to implement a user account system. Our user account system is fairly straightforward. A user has to register on our web site, using an email address and creating a username and password that they will use to login in the future, before they can use the system. Once a user logs in, they are able to configure their system for use.

The system home page shows the current monthly power use and bill for the whole house as the main component of the page, and updates in real time without the need to reload the entire page. The home page also shows the date of the next bill and a projected final bill for the month based on power usage for the month so far. The power used is stored in the database, but the bill calculation is done directly by the website software using the stored power use and billing rates. If you, the reader, check section 3.7.1, you will find that the group decided to use an IP camera for their camera feed, which provides a video feed accessible with a URL. The homepage includes a link to allow the user to view this video feed if they wish. The homepage is only one of several pages available to the user via links on a tool strip at the top of the page. Other pages include a page for devices, a page for calibration/system settings, and a page for account settings. An example of what the homepage was planned to look like is in Figure 3-16, while the final result is in Figure 3-17

The devices page shows a list of the devices that the user has connected to their system. Each device is listed by its user-define name and if it is an outlet, it has its current monthly power use and cost displayed by it. Each device has buttons for the appropriate actions the user can select, such as toggling power to a light or outlet on or off, or unlocking a door lock. The option to change a device name is also on this page, near each device, and there is an option to add a new device to the system. An example of the Devices page is in Figure 3-18, and the final page on the website is in Figure 3-19

Figure 3-16: Example of the home page



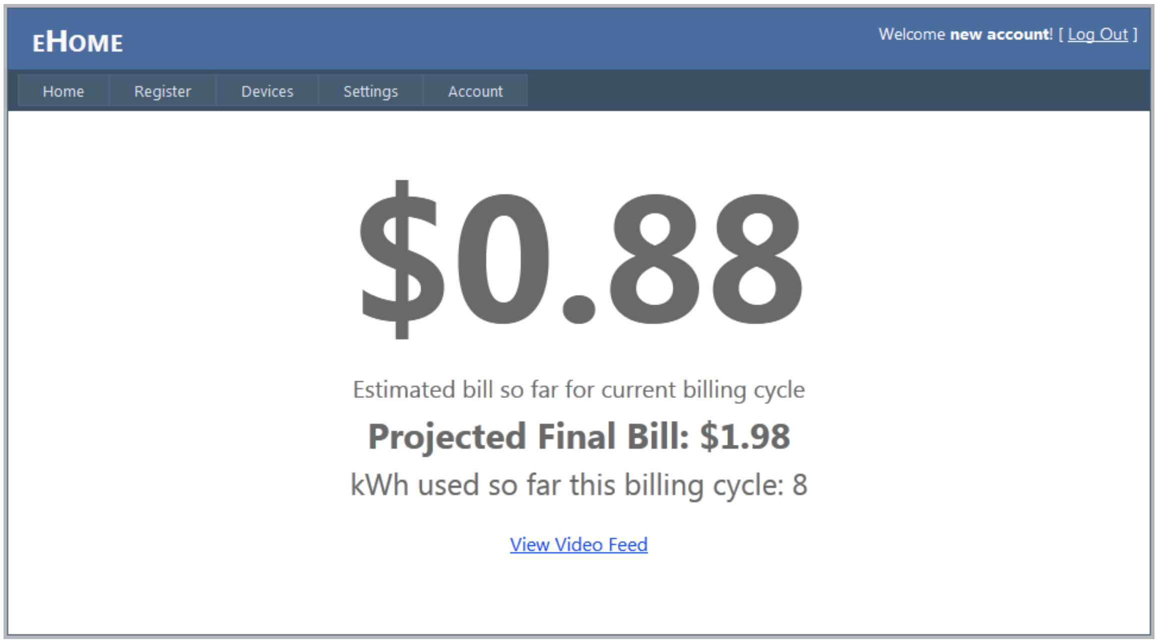
$XX.XX

Bill so far:

Projected final bill:

Kwh uses so far this month:

Figure 3-17: final home page



$0.88

Estimated bill so far for current billing cycle

Projected Final Bill: $1.98

kWh used so far this billing cycle: 8

Figure 3-18: Example of the devices page

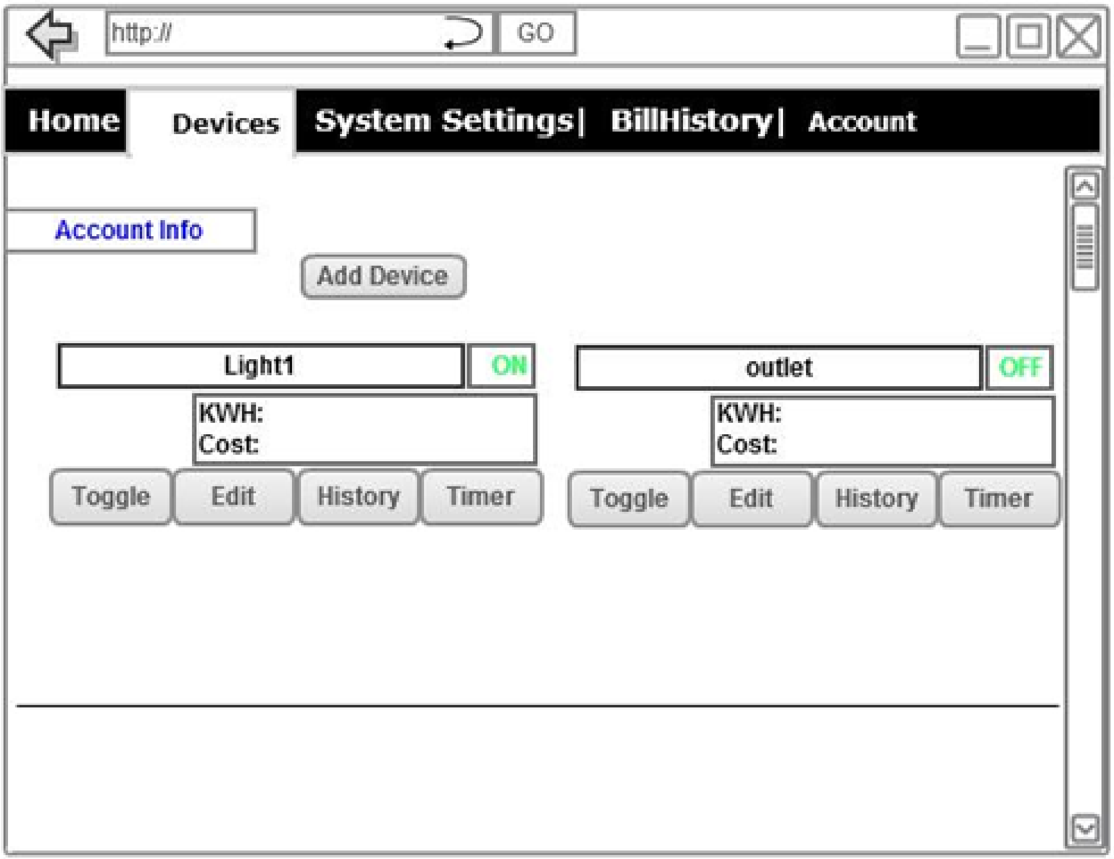
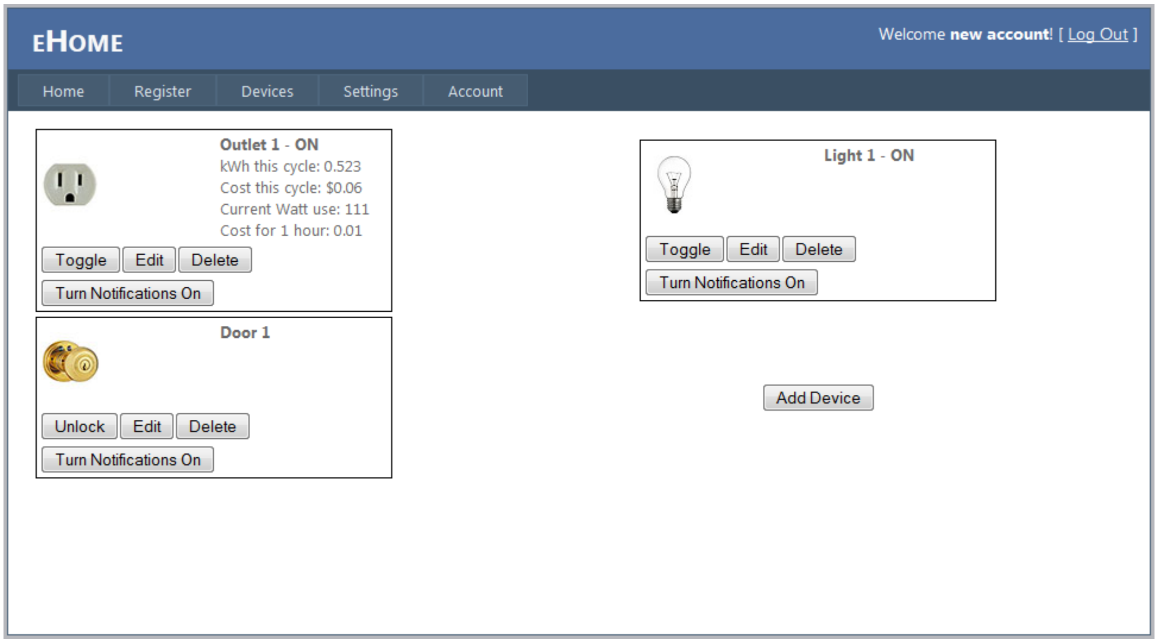


Figure 3-19: Final devices page



Outlet 1 - ON
kWh this cycle: 0.523
Cost this cycle: $0.06
Current Watt use: 111
Cost for 1 hour: 0.01

The calibration/system setting page is where the user can set all of their options for their system's behavior, such as the billing rates and the image processing software calibration. In order for the website to be able to calculate the estimated monthly bill so far, as well as the monthly cost so far for devices, the user has to define their power bill rates on this page. The user also has to define a usage range where the rate is applicable, because some power companies charge more per kWh after a certain amount of power has been used. For example, ten cents per kWh up to a thousand kWhs, then fifteen cents per kWh up to fifteen hundred kWhs, and then twenty cents per kWh beyond fifteen hundred. The user is able to enter as many of the rates and ranges as they need to in order to accurately estimate their monthly bill. They also need to enter the date of the month that their bill is calculated, so that the system knows when to start a new billing month.

The system settings page also lets the user tell the system how many decimal places, if any, their meter displays, because the meter reading software will not be able to detect this on its own and will just read the whole meter. So, if a user enters one decimal place, then the last digit of the meter reading will be a decimal place for display and bill calculation purposes. Also on this page are the settings for the system's camera IP address and central hub IP address.

The final option on the system settings page is the meter calibration section. This is where the user can set up the digit locations for their system. There will be an image of their meter taken from the camera, and the user will set each digit by clicking the 'set digit' button, clicking the top left of the digit area on the image, and then clicking the bottom right of the digit area. They do this for each digit, and then save the digit locations. An example of the system settings page is in Figure 3-20, and the final developed page is in Figure 3-21

The account settings page is very typical. On this page, the user is able to change their associated email address, username or password. This page is also where the user will set up their SMS settings. The user has to enter their cell phone number on this page, and has to verify it with a code sent to the phone number, so that they can't enter a wrong or random person's phone number. An example of the account setting page is in Figure 3-22, and the final page that ended up on the website is in Figure 3-23.

That concludes the features of the website, but the group also had to decide how the website was developed. The web server that was chosen is Abyss Web Server run on one of the group members' own computer, so this is where the website is hosted. Abyss offers support for PHP, javascript and ASP.NET, along with standard HTML of course. Given that the group member who worked on the website has experience with C# and ASP.NET, the website was developed using that, providing easy access to the MSSQL database. While the group member's experience has mostly been for console applications, he has had some experience with ASP.NET websites as well. ASP.Net provides the ability to have update panels on a website, with timers that refresh the panel every so often using AJAX scripts. The pages in the website are divided up as classes, and an example class diagram is shown in Figure 3-24.

Figure 3-20: Example of the system settings page

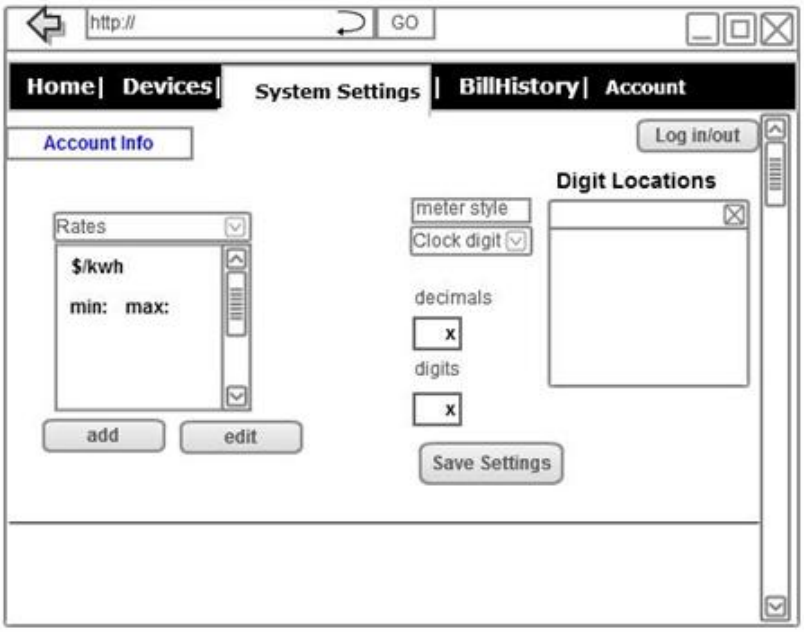
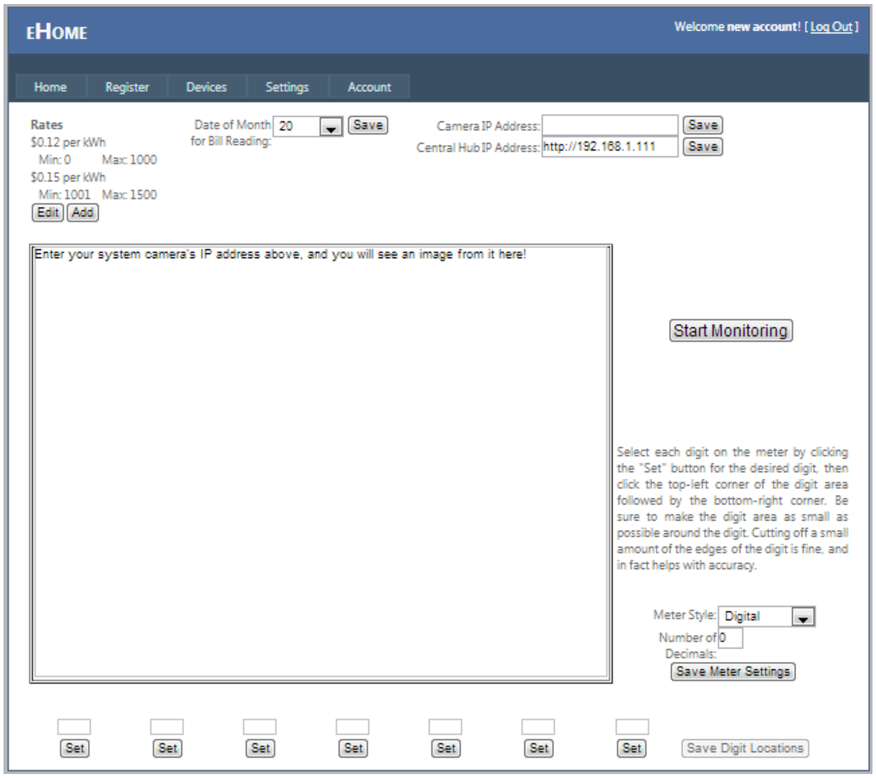


Figure 3-21: Final system settings page



Rates
$0.12 per kWh
Min: 0 Max: 1000
$0.15 per kWh
Min: 1001 Max: 1500

Date of Month for Bill Reading: 20

Camera IP Address:
Central Hub IP Address: http://192.168.1.111

Enter your system camera's IP address above, and you will see an image from it here!

Select each digit on the meter by clicking the "Set" button for the desired digit, then click the top-left corner of the digit area followed by the bottom-right corner. Be sure to make the digit area as small as possible around the digit. Cutting off a small amount of the edges of the digit is fine, and in fact helps with accuracy.

Meter Style: Digital
Number of Decimals: 0

Figure 3-22: Example of the account settings page

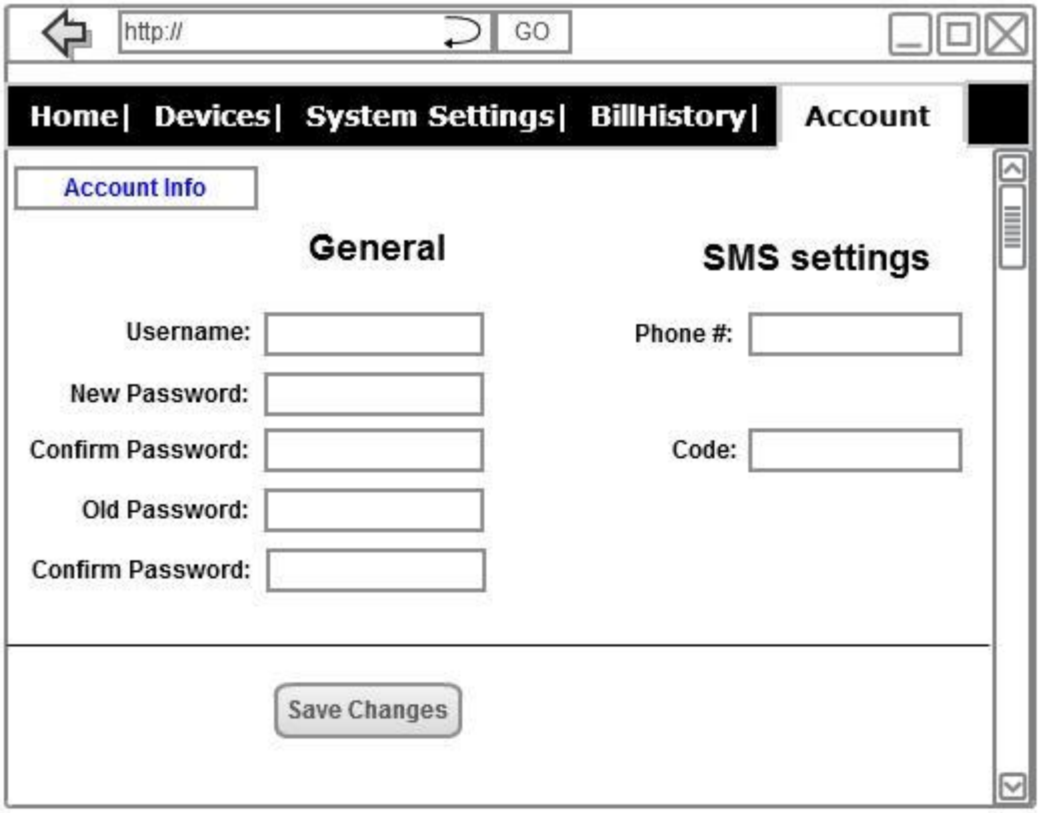
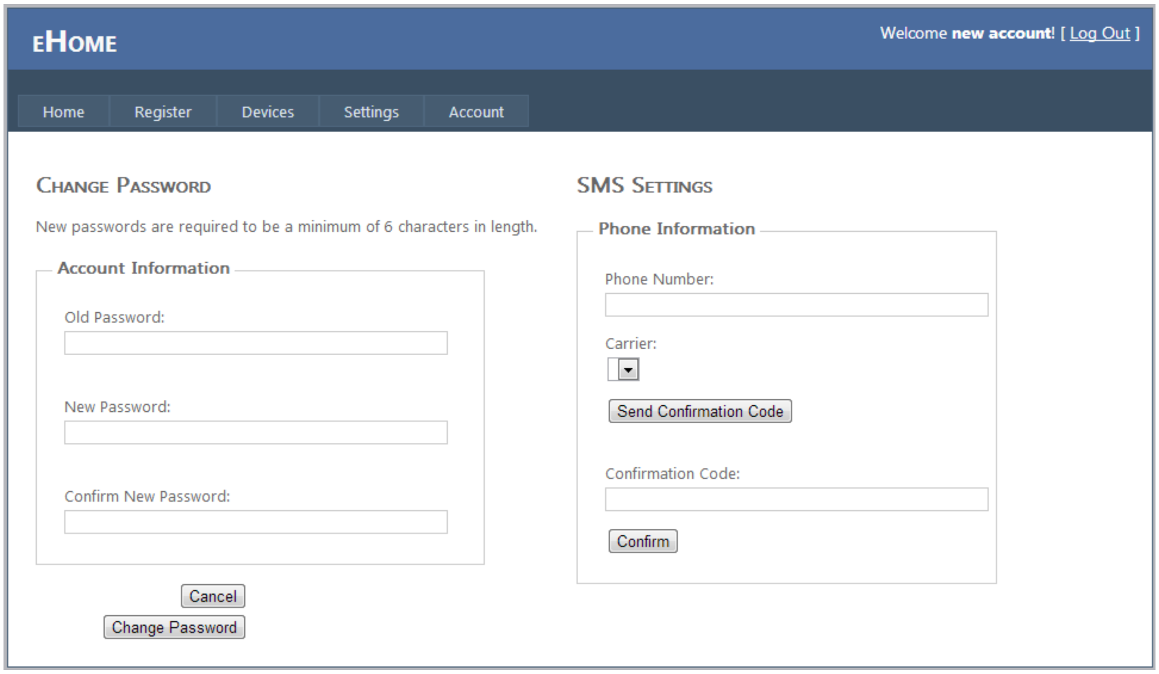
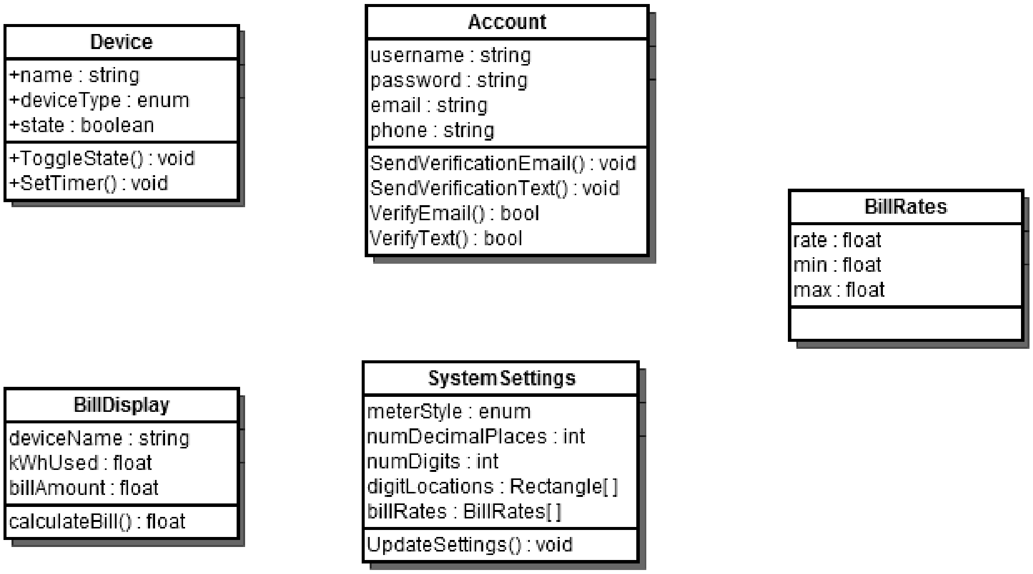


Figure 3-23: Final account settings page



New passwords are required to be a minimum of 6 characters in length.

Figure 3-24: Class diagram for website. Classes are independent of each other, and are just used by the website, so no relationships exist between them.



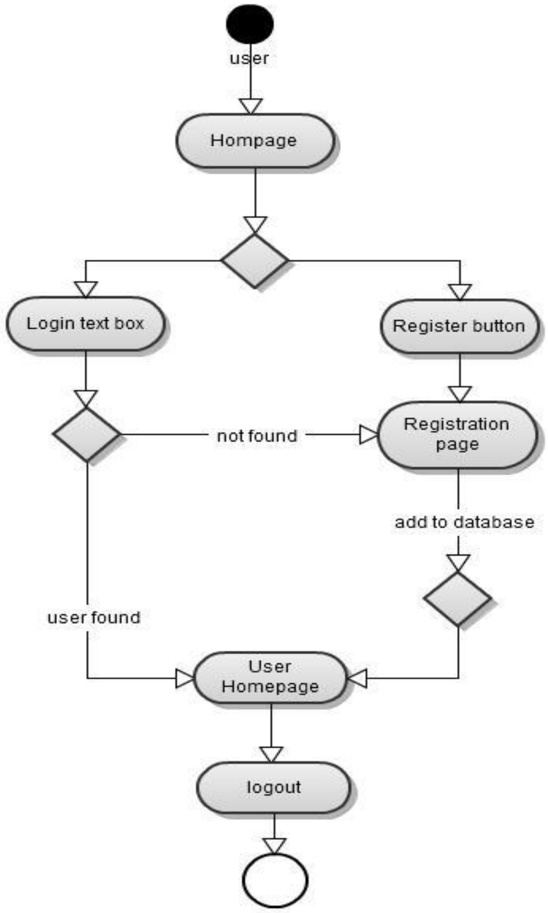
3.3.1 User Accounts

One of the issues with creating a system network is granting access to the network. The project needed to make sure only one person can control a network at a time. The group wanted to avoid multiple requests from different users at the same time. To do this the group planned on creating user accounts with password protection. When creating a website project in Visual C#/ASP.NET in Visual Studio 2010 with framework 4.0, the project is automatically set up with full account functionality, such as registration and logging in, and showing pages a certain way depending on if a user is logged in or not. Setting up the MSSQL database to use this functionality is also very simple, all that's needed is to run a setup file that is included in the SQL Server installation, and it sets up the database to have tables and stored procedures and everything else necessary for user accounts created and managed from the auto-generated ASP.NET 4.0 website.

Users use the auto-generated registration page to register a new account with a username, email address and password. There is no built-in check on email address format, so the user can enter invalid email addresses, but the group decided not to worry about checking anything like that, since they did not include any email functionality in their system. Once a user has an account, they can login from the login page, which has your standard username and password prompts.

When a user is logged in, they can access their account settings page. The account settings page is a modified version of the auto-generated password change page. The group added the ability to set SMS settings onto this page to make it more general purpose for account settings. Figure 3-25 shows a flow chart for the user account functionality.

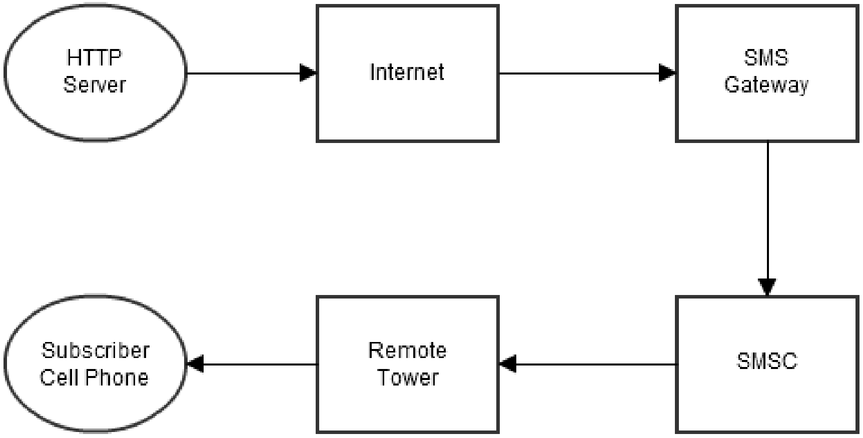
Figure 3-25: Flow chart for using a user account



3.4 SMS

The software for the SMS notification is built into the monitoring/image processing software. While the monitoring software runs through its loop, it checks for certain triggers that the user may have set that they want text alerts about, such as a device turning on or off. If the user has specified that they want an alert, then the software creates an email and sends it to the user's text email address, which is their ten digit phone number at their service provider's text message domain. The email is sent through a Gmail account set up for the project, by sending the mail to Gmail's smtp mail server with the Gmail account's login credentials. When the user receives the text, it is in an email-via-text format, and shows that it is from the Gmail account. The process can be seen in Figure 3-26.

Figure 3-26: Process for sending a text message from the website to a phone



3.5 Server/Hosting Design

After considering the website hosting options available to the group, as well as the project as a whole, the best hosting option available to them seemed to be hosting their own server. Hosting their own server is cost effective, which is important, because the rest of the project is going to be costly, so minimizing cost wherever possible was a concern. In addition, because one of the group members has experience successfully hosting a website on a personal web server running on his own computer at home, the group knew that it would work. This will provide them with plenty of freedom as far as how they structure the website and how much space they needed for it, since the only limit is the hard drive space on the hosting system. This also proved useful for development, as all four of them could have potentially had the server software set up on their machines, so they would have been able to develop parts of the hosted software individually without interfering with anyone else, if that had been necessary.

Now that the group had decided to host their own server, there were two more choices to be made: which server software to use, and which DDNS service to use. In both of these choices, previous experience with one of the choices was the deciding factor. They used Abyss Web Server for their hosting software, and Dyn for their DDNS service. In addition to previous experience, Abyss trumps Apache by being more user-friendly and having already-built-in support for SQL and ASP.NET, which are crucial for the project. Abyss beats out Windows Server by being completely free, not just a free trial. If they have any interest in keeping the system running after the senior design period is over, Windows Server is not a good idea unless they plan to pay for it when the trial ends. Dyn was chosen over No-IP despite its inconveniences due to previous experience, as well as having more of a reputation and a very professional look to the website. No-IP seemed legitimate enough, but before any research for this project, nobody in the group had heard of it, and the website doesn’t feel as professional as Dyn’s.

3.6 SQL Database Design

MySQL and SQL Server are evenly matched for how well they would have fit into our project. Both solutions offer free versions and the same simple database functionality that they need. Ultimately, the choice came down to previous experience, and the group member who was working with SQL the most has more experience with Microsoft SQL Server, so that was the group’s choice for the SQL software that the group uses. Now that they knew which software they will be using, they needed to determine what data they need to be storing and how they would store it.

Data that needed to be stored for an account includes account names and/or email accounts, and passwords for login purposes and account differentiation. These should have their own table for security purposes. Each user may have a different price per kilowatt-hour for the power bill, and that rate may even change depending on how much power is used, so that will have to be stored as well. Due to there being more than one rate possible per account, as well as multiple accounts having the same rates, rates should be stored in a separate table with a linker table to relate them. The date of the month that power bill is finalized should be saved, as well as the specific date of the last bill. These two things have their own table and relate to the account table. The IP address of the network that the user’s central hub is on should be saved in order to facilitate sending signals to the system from the website, and it should have its own table.

Other information that must be saved was for any devices such as outlets or light switches that the user has linked to their account. For each device, the current power use for the month so far must be saved, as well as the current state (on or off). The system allows users to give custom names to their devices in order make it easier for the user to know which one is which, so this name, the current monthly use so far and state can all be saved on one table. Device types are stored in a separate table in order to allow for the addition at any time of any devices that are incorporated into the project.

The information that needs to be stored for the image processing software must also be saved for and linked to each account, and includes the meter style, digit locations and latest meter reading. Style and readings can be stored in one table, while digit locations should be stored in a separate table, since there are multiple locations per meter.

All of these tables are related using primary and foreign keys. The database is accessed from the website and from the image processing software, both of which were written in C#/ASP.NET, a language that support SQL queries and updates. Integrating SQL into the rest of the project was not any problem.

3.7 Image Processing Design

3.7.1 Camera Design

After severe flaws were observed for attempting to transmit images over a Universal Serial Bus cable, other alternatives were considered. IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 (ZigBee) were both potential candidates for connecting the camera to the network, but not for long due to limited capabilities and resources. Soon, it was realized that IEEE 802.11 protocol cameras (IP cameras) would be ideal for this type of image transferring.

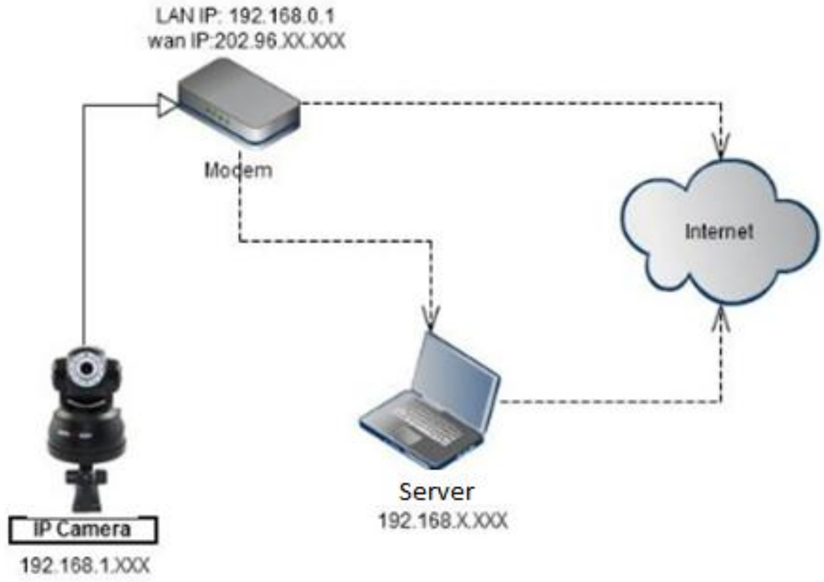
IP cameras actually host a web-page themselves where they upload a real-time video or image stream. This allows the images to be accessed from any point that has an internet connection. It can communicate through walls, which is necessary being that the electric meter is usually outside of the house, and the Wi-Fi router will most likely be inside with the Central Hub. On top of this, there would be no cables running through along any walls because it's wireless.

The advantage of using an IP camera is its ability to immediately create a digital image. The digital processing takes place within the camera. Another advantage of the IP camera is its ability to be found within a network. A wireless network can simply search for the IP address of the camera to begin communication. Internet protocol cameras also offer FTP and HTTP protocol to upload individual images, which is exactly what the project needed. The group just needed the image file so that the image processing program could try to interpret a numerical value from the image. These images can be streamed as motion JPEG or MPEG using file transfer protocol. As long as the system has the image file, the camera had done its job. The image processing software does the rest.

The project needed the ability to continuously refresh the reading from the power meter. This camera also takes care of that problem. An IP camera can be set up to record continuously or even be scheduled to record for only certain intervals of time. This way the system could have a power reading in real time and have the user see an accurate reading.

Since the government regulates the frequency and strength of analog wireless devices, problems can occur with interference from other wireless devices. Simple devices such as cell phones can create interference. The digital transmission of the protocol camera has no interference from analog devices, and with the 802.11x communication standard there is encryption built in. Given the expected distance between the camera and the network, the ability to avoid interference is extremely important. Quality of the digital image needed to be as accurate as possible in order for the software to interpret an accurate value. IP cameras can also offer images in megapixels of quality as high as some digital cameras. Figure 3-27 is a simple diagram of how the IP camera can be used in a network by accessing address of devices.

Figure 3-27: IP camera in the system network



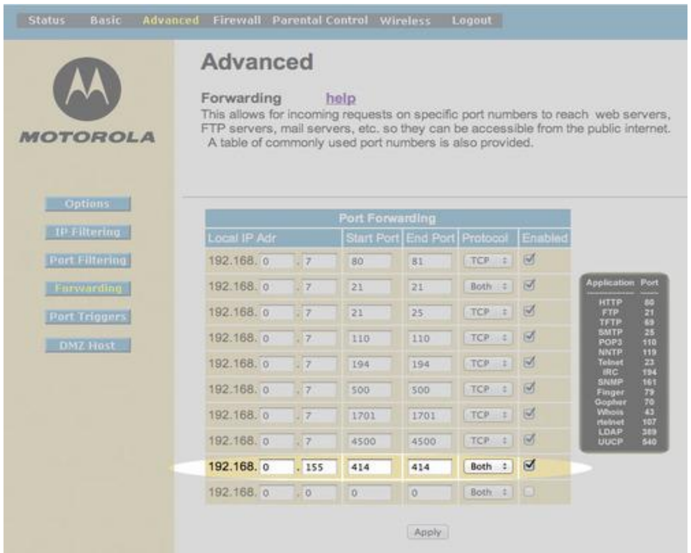
With the advancement of technology in internet protocol camera networks, the price of an IP camera was no longer an issue. Companies like D link and Wansview produce cameras that are not only inexpensive but easy to work with. These wireless cameras have software capable of communicating directly with a computer, phone app, or tablet.

When choosing and IP camera for the groups' specific purposes, the group had a few specifications. Primarily, the project needed the camera to be durable and capable of operating night or day. The group chose an IP camera that offers waterproof encasing. This was necessary for testing where the camera must be positioned outside in front of a power meter. Being outside, the camera needed to be able to withstand the humid and rainy weather conditions of Florida and still operate without any hindrance in performance. For testing and proof of concept the group selected a camera with night vision capabilities. The concept of the project required a camera that operates at all times. The camera needed the night vision functionality to ensure that the camera would take clear pictures regardless of the time of day. The specifications this group was looking for also included FTP still picture uploading and downloading with 802.11 protocol. This was crucial for the image recognition process. Although IP cameras are great for recording video, the group only needed to be able to pull a single image from the camera. The image recognition just needed to handle a single jpeg image and find the integer value on the image.

There are several companies that produce IP cameras. Not only did the group need to weigh the options based on specifications, but they also needed to consider budget. These devices can range from fifty to several hundred dollars. The waterproof capabilities that were needed also brought up the cost. The Wansview product line was the most appealing due to the fact that their cameras could fulfill the given specification needs. Depending on the model, these cameras could also be found online between the prices of $50 to $70. These cameras offered M-JPEG video compression, 24 infrared LEDs for night vision, IP66 waterproof grade enclosure, 802.11 protocol support, FTP picture upload/download, free DDNS by manufacturer, and free software.

These cameras come with software to set up a router for port-forwarding. The system could set up the IP camera to accept a request from the projects' web server or the hub by accessing port numbers. Communication could begin by assigning a port number to the camera with the DevFind.exe application and typing in the cameras IP address. As you can see in Figure 3-28 below there are also options for TCP protocol and IP filtering

Figure 3-28: TCP protocol and IP filtering options



Advanced

Forwarding help

This allows for incoming requests on specific port numbers to reach web servers, FTP servers, mail servers, etc. so they can be accessible from the public internet. A table of commonly used port numbers is also provided.

Port Forwarding

Local IP Adr			Start Port	End Port	Protocol	Enabled
192.168.	0	7	80	81	TCP	☑
192.168.	0	7	21	21	Both	☑
192.168.	0	7	21	25	TCP	☑
192.168.	0	7	110	110	TCP	☑
192.168.	0	7	194	194	TCP	☑
192.168.	0	7	500	500	TCP	☑
192.168.	0	7	1701	1701	TCP	☑
192.168.	0	7	4500	4500	TCP	☑
192.168.	0	155	414	414	Both	☑
192.168.	0	0	0	0	Both	□

Application	Port
HTTP	80
FTP	21
TFTP	69
SMTP	25
POP3	110
NNTP	119
Telnet	23
IRC	194
SNMP	161
Finger	79
Gopher	70
Whois	43
rtelnet	107
LDAP	389
UUCP	540

3.7.2 Image Processing Software Design

The image processing software needed to be able to take in a picture of a residential power meter and produce a number representing the reading of the meter. Because of the difficulty and time investment required to read the clock-digit style and analog meters, the group decided that they would not be able to develop that functionality, but would instead focus on the easier to work with and increasingly more widespread digital meters. Unfortunately, due to a lack of distinctive features of digital digits for pattern recognition software algorithms to recognize, they were also not able to make the digit location detection automatic either. Digit locations were to be given to the program manually, with the user clicking on an image to provide the locations. Reading digital digits once the locations are known is a fairly straightforward automatic process after that. If the camera was moved however, the digit locations would need to be given again.

The group decided that having their software run on the same system as their web server would be of greater benefit to the project than running it on a microcontroller. The project needs a website whether the image processing software is on it or not, and there is no limiting factor to why the group shouldn't run the software on the server. Running the software on a computer also gets rid of any concerns about using floating point math, a limiting factor of many lower-cost microcontrollers. As an added bonus, the group member who was developing the majority of the image processing software has lots of experience with C# and ASP.NET, which are supported natively by Abyss, the chosen web server (see Section 3.2.2), and is the language and platform that the software was developed in.

After considering the computer vision software options discussed in section 2.3.2, it was decided that writing their own custom software will provide the group with the best results. They borrowed the JPEG-fetch-from-video, grayscale conversion and image resizing functions from the AForge.NET library in order to aid in obtaining an image ready to be processed, but everything else was custom-made, using the existing libraries as reference material only. Creating their own software allowed greater flexibility and more control over the way the software performs. Rather than trying to adapt the broader functionality of OpenCV, CVIPtools or AForge.NET's entire libraries to their needs, writing the software themselves with the express purpose of reading a residential power meter allowed their code to be more concise and to the point, with no unnecessary extras to worry about taking up time or system resources. Theoretically, the custom software should be able to read a power meter faster than if they were to adapt the existing libraries, because there are no extra steps bloating the process.

Reading the meter at night is a concern that might not seem obvious at first. It is, of course, a concern, because a home power monitoring system should be able to read the meter at any time of day or night, in order to provide accurate power use and bill data at all times. The simplest solution is to provide light for the camera at night, whether the camera itself has built in lights, or a light is added to the camera setup. The one drawback of this simple solution is that some users might just not like having a light on all night on the side of their house. The only other solution that still allows reading the meter visually is some sort of night-vision camera, but that could have been cost prohibitive. See section 3.3.1 for the camera setup design for the chosen solution.

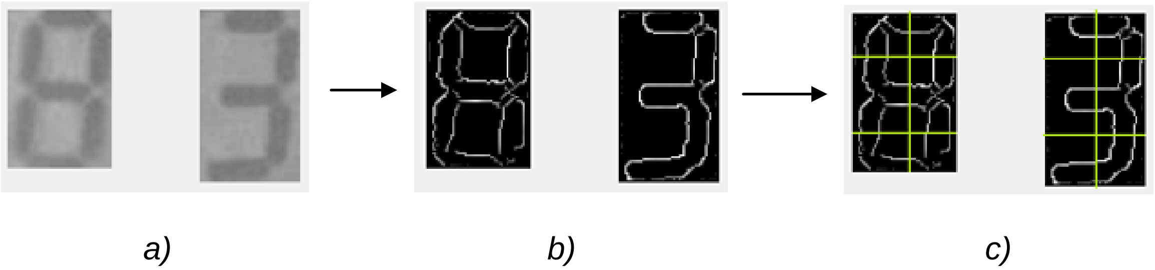
A quick internet search will show that residential power meters typically have color schemes of black, white and gray. There is typically little to no color on a power meter, and even when there is, it is not significant to how the meter would be read. Due to this, It was determined that for the project's purposes color information is unnecessary, so upon obtaining an image for processing it is immediately converted from color to grayscale using the RGB-to-grayscale function borrowed from AForge.NET.

Once the image processing software determines that a user's system is ready for monitoring and the user has told the system to start, it follows the same cycle of processing until it is told to stop. The first step is to obtain an image from the camera. This is done using AForge.NET's JPEGStream class. The image is converted into grayscale, again using AForge.NET. At this point, the program accesses the database to get the digit locations for the particular system being monitored. For each digit location, a section of the meter image is cropped out and sent on for "reading" individually.

For a single digit image, the process for determining its value is very straightforward. A Canny Edge Detector, provided by AForge.NET, is used on the image at first. A Canny Edge Detector does exactly what it says: it detects edges in an image, based on finding abrupt changes in brightness in a grayscale image. There is a high and a low threshold for how much difference there must be between pixel brightness. At first, the detector just checks each pixel to see if there is an abrupt color change of value greater than or equal to the high threshold, either horizontally, vertically, or diagonally. Any pixels that pass this threshold are considered to definitely be edges. All pixels adjacent to definite edges are potential edges. For each potential edge, the detector will now check for brightness changes compared to the low threshold. Any potential edges that pass the low threshold are now considered edges as well, and all pixels adjacent to them are not potential edges. This process continues until no more potential edges remain, only edges and non-edges. This process is represented visually by making all non-edge pixels have a black color, while all edge pixels have a brightness proportional to their brightness change value while the Edge Detector was running. This representation is what is used in the next step of the digit reading process.

Determining the value of a digit is very simple. The technique used is simply to detect whether or not a particular segment of the seven segments that make up the digit are showing (i.e. are black or just generally darker than the background). If a segment is shown, then the Canny Edge Detector should have marked the edges of the segment with white, otherwise the whole area of the segment should be uniformly black throughout. This is checked by just finding if there is an edge in the area of each segment. A check is done vertically down the middle of the digit image, checking the top third for edges for the top segment, the middle third for the middle segment, and the bottom third for the bottom segment. Then, for the top left and right segments, a check is done horizontally a quarter of the way down the image, and then the same thing is done three-quarters of the way down for the bottom left and right segments. A visual example of this is shown in Figure 3-29. For each segment, it is recorded whether that segment was detected as there or not. After the segments are detected, each of the ten possible configurations is checked for a match to all seven segments' states. If a match is found, that value is returned as the digit value. If nothing is found, then the value ten is returned, meaning an invalid reading.

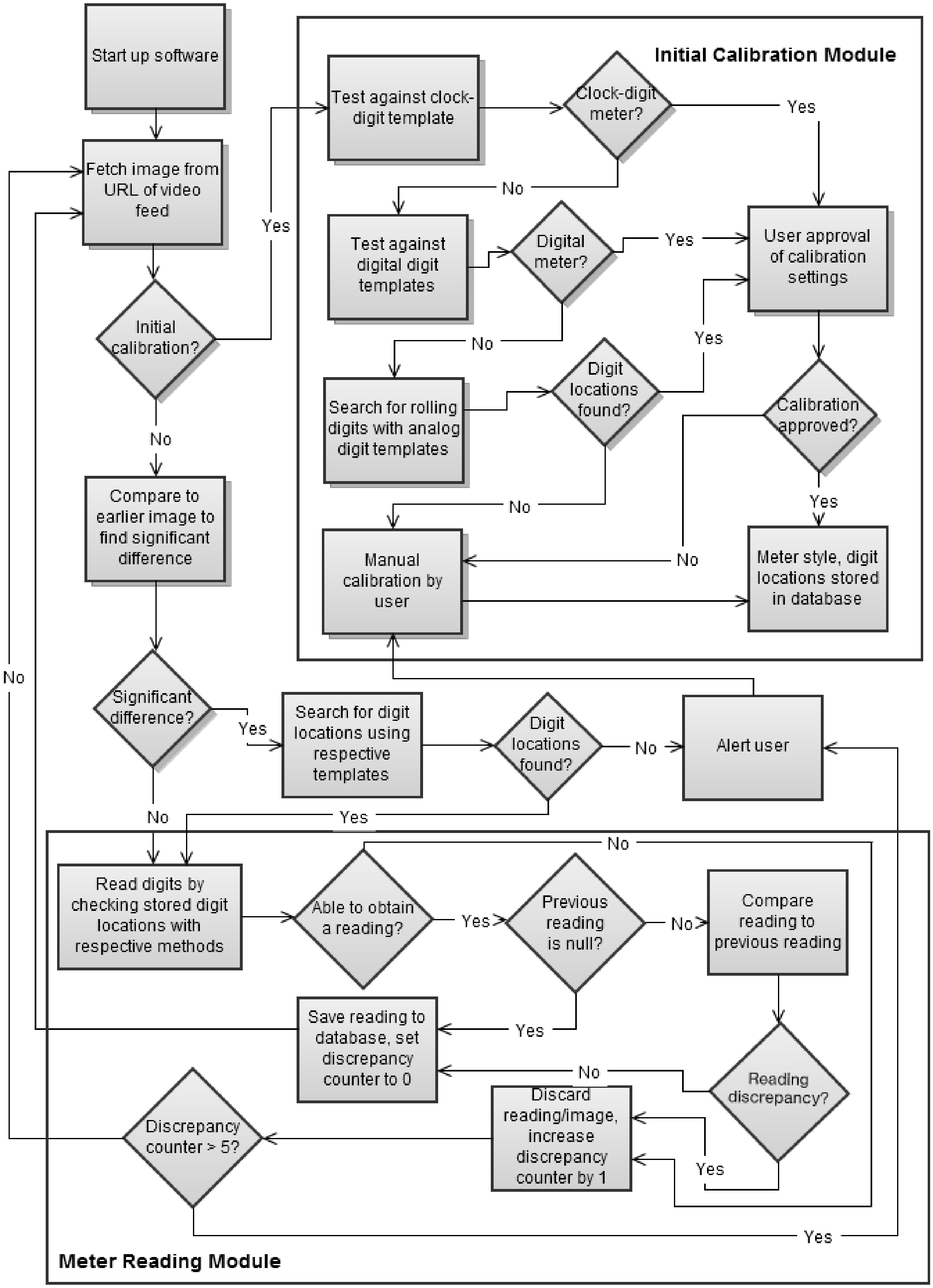
Figure 3-29: Visual example of how digital digit reading is done. a) original cropped image. b) Canny Edge Detector result. c) the lines along which edges are checked for to determine segments shown.



That process is repeated for each digit, and then the readings obtained from each digit are added together to get a meter reading. Because power meters increase their value very slowly, the only values that are valid are either the same value as the previously valid reading, that value plus one, or a value of zero if the last value was all nines, the highest the meter can display. Because this check is done on a fairly frequent basis while the meter will display the same value for some time, then if something happens to cause an invalid reading, the process still has plenty of chances to obtain a correct reading, provided that the error was due to the camera being moved.

The only exception to the valid/invalid readings rule is if the system monitoring was previously turned off, and is then turned back on. Any amount of time could have passed since the last reading, so it is necessary to allow for a gap in values. A flowchart of this operation is shown in Figure 3-30.

Figure 3-30: Flowchart of software operation


Start up software
Fetch image from URL of video feed
Initial calibration?
Yes
No
Compare to earlier image to find significant difference
Significant difference?
Yes
No
Initial Calibration Module
Test against clock-digit template
Clock-digit meter?
Yes
No
Test against digital digit templates
Digital meter?
Yes
No
User approval of calibration settings
Yes
Search for rolling digits with analog digit templates
Digit locations found?
No
Calibration approved?
Yes
No
Manual calibration by user
Meter style, digit locations stored in database
Search for digit locations using respective templates
Digit locations found?
No
Yes
Alert user
No
Read digits by checking stored digit locations with respective methods
Able to obtain a reading?
Yes
No
Previous reading is null?
No
Yes
Compare reading to previous reading
Reading discrepancy?
No
Yes
Save reading to database, set discrepancy counter to 0
Discard reading/image, increase discrepancy counter by 1
Discrepancy counter > 5?
Yes
Meter Reading Module

3.8 Central Hub Design

3.8.1 Central Hub Development

Given that the ATMega328 was sufficient for every aspect of the project, not to mention that it allowed for many useful scalability options, it was selected as the microprocessor for the project. One of the most appealing aspects for choosing the ATMega328 as the main processor for the project was the Arduino development environment. The time that can be saved not getting stuck on little details was spent scaling the project. The ATMega328 has many capabilities, including serial communication, and Serial Peripheral Interface (SPI) Bus master operation capabilities. It has 14 digital input/output pins, 6 Analog to Digital Converters (ADC's), and 6 Pulse Width Modulation (PWM) outputs, which was more than enough for the given applications under development.

An entire development environment for the ATMega328 was bought for less than $20. Also, given that the ATMega328 could perform each of the tasks required by each node in the sensor and control network, it easily served as the microcontroller for each of these circuits. This simplified the design process greatly, being that the processor will be programmed in the same fashion for each circuit, and worked similarly for each task it has been assigned to fulfill.

Another exciting aspect for development using the ATMega328 development environment discussed above was that it was highly compatible with the Arduino Ethernet Shield, which provided a reference design for Ethernet communication. This was extremely handy, being that nobody in the group has directly had experience setting up this kind of circuitry. Furthermore, this component was stacked simultaneously with an Arduino XBee shield. Looking at the circuitry, most of the connections conducted through each layer of silicon. The Ethernet was a priority to stack directly to the Arduino, so that the Serial Peripheral Interface (SPI) Bus could pass through to it. Otherwise, the XBee could not pass the Serial Peripheral Interface (SPI) Bus, due to a lack of the proper connectors. However, if the XBee Shield was stacked on top of the Ethernet shield, on top of the Arduino, the XBee communication lines would pass through the J3 connector on the Ethernet Shield. There were two more signals that are needed: the first was a grounding signal. This was important to make sure that the XBee Shield was not floating. The second signal that was required by the XBee Shield, but not supplied by the Ethernet Shield, was a 5 volt power source. This was found in multiple places by observing the trace carrying 5 volts on the Ethernet board. The first location where this was done is by running a wire from the pin header location shown in Figure 3-31.

In the event that a wire cannot fit between that location and the XBee board, a wire could just be directly soldered to the top of the In Circuit Serial Programming (ICSP) header, as shown in Figure 3-32.

Figure 3-31: 5V source pin on Ethernet Shield

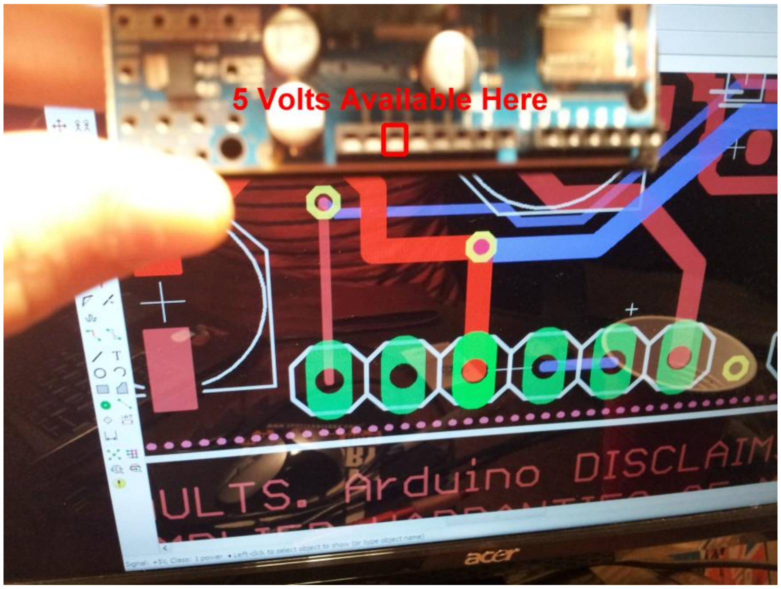
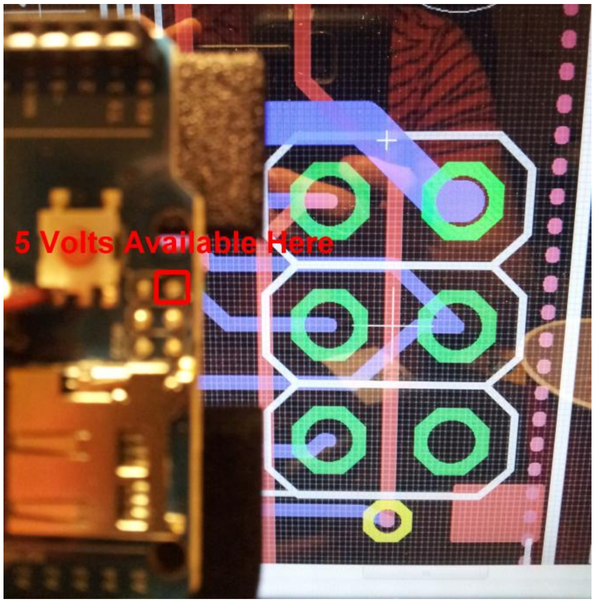


Figure 3-32: 5V input pin on XBee Shield



3.8.2 Final Hardware Design

The primary and most notable component on the central hub was the ATMega328, which served as the main processor. This part was capable managing and controlling all the communication that was fed through the central hub. The ATMega328 was powered by a 5 volt signal, provided by the MC33269D-5.0 voltage regulator, and had to be grounded so that no charge would accumulate on it. It had all the necessities to communicate with the internet through the W5100 Ethernet chip, and all the necessities to communicate with the XBee boards that will make up the sensor and control network. It had up to 6 Analog to Digital Converters (ADC's), and at least 6 output pins, which made it a suitable candidate for the Outlet Monitor and Controller circuit, Lock Controller circuit, and the Circuit Breaker Monitor circuit as well.

There were three different package configurations available for the ATMega328 microcontroller. The MEGA8-AI package corresponds to the Thin Plastic Quad Flat Package (TQFP), and would have to be soldered directly to the board. The MEGA8-MI package corresponds to the Micro Lead Frame Package (MLF), was a little bit smaller than the Thin Plastic Quad Flat Package, would still have had to have been soldered to the board (but not manually, since there are basically no leads), and also had the option of a thermal pad which can transfer heat from away from the chip to avoid overheating. Finally, the MEGA8-P package corresponds to the Plastic Dual In-line Package (PDIP) which was the same package used in development, and allowed the implementation of a socket, from which the ATMega328 would be removable. For this reason, the Plastic Dual In-line Package, while the Thin Plastic Quad Flat Package and Micro Lead Frame Package both had their advantages, was far superior in that the part can be easily replaced, in the event of a chip failure, without having to spin a brand new printed circuit board. Also, the ATMega328 microcontroller could now be programmed in the development environment, and transferred easily to the end prototype without any difficulty.

The ATMega328 had a 16 Megahertz (MHz) external clock wired between pins XTAL1 and XTAL2 to supply clock signal. The 16 Megahertz (MHz) crystal was the fastest clock speed that is specified in the ATMega328 datasheet. This was sufficient for communicating to the XBee module communicating to the W5100 Ethernet chip, and running any other peripheral functions that would take place throughout the execution of code. The crystal is stabilized by two grounding capacitors, with values of 22 picofarads each, as indicated by the ATMega328 datasheet.

The ATMega328 had a reset pin mapped to the top-leftmost pin (pin 1). This pin is activated by being pulled low for longer than the minimum pulse length. Once this would occur, the software would be interrupted, without making a note of where to return, and executes the Reset Vector. In the case of this project, that means that the board would simply reboot. This could be pulled low two different components on the board: The first is switch S2, which pulls the reset line low by tying it directly to ground when the momentary switch is depressed. The second way in which the reset line could be pulled low is by the CAT811TTBI-CT3 voltage supervisory circuit. This component monitors the 5 volt power line, and would pull the reset line low when a voltage is observed below a set reference voltage to protect the integrated circuits from suffering damage or presenting bugs if brown out conditions became present.

The ATMega328 handled communication with the W5100 Ethernet chip through a Serial Peripheral Interface (SPI) Bus. It performed this by running in SPI master operation mode. The ports used for this of a Serial Clock (SCK – PB5) line which were controlled by the ATMega328, a Master Input Slave Output (MISO – PB4) line which transmitted information from the W5100 Ethernet chip to the ATMega328, a Master Out Slave Input (MOSI – PB3) line which transmitted information from the ATMega328 to the W5100 Ethernet chip, and a Slave Select (SS – PB2) line which was just set low throughout the communication, since there was only one slave device; the W5100 Ethernet chip.

The 74LVC1G125 Bus buffer/line drivers were connected to each of the Serial Peripheral lines Master Output Slave Input (MOSI), Set Clock (SCK) and chip select lines (SCS). There was also a buffer for the SEN pin on the W5100. Each of these buffers is powered by a 3.3 volt line outputted by the MC33269D-3 voltage regulator. This was because the W5100 Ethernet Chip is powered by 3.3 volts, and the logic level of these signals had to be reduced from the 5 volts logic signals that the ATMega328 produces.

There were a series of Light Emitting Diodes (LED's) connected to various networks to show statuses, or signals. Many of these LED's were buffered through Chip Resistor Arrays, instead of normal resistors, in order to save board real-estate. There is an LED indicating both the 5 volt status, and 3.3 volt status. These were useful in letting the person debugging the circuit know which nets had power at a given moment. Also, the LED's connecting the communications lines indicated these connections, and verified whether or not there even was a connection.

There were two major ports on the board. First is the POE-RJ45, which is the Ethernet cable connector. This had to be placed on the board with the dimensions of the project enclosure known. It had to hang off the printed circuit board slightly, so that it could be flush with the enclosure. The next set of ports was the XBee socket, and is labeled accordingly in the schematic. This was where the XBee chip was plugged into the main printed circuit board. The dimensions were carefully mapped out, so the chip could fit on top of the board properly.

There was a solder jumper pad located on the schematic, running from interrupt line on the W5100 Ethernet Chip to the interrupt line input on the ATMega328. This provided a way to send interrupts to the ATMega328 from the W5100 Ethernet Chip. This would allow any change detected by the W5100 to be sent to the ATMega328, where it could be handled. The terminals of the solder jumper could be shorted together, or it might have had to go through an amplifier to jump the logic level back up to 5 volts for the ATMega328.

The MC33269D-5.0 and MC33269D-3.3 voltage regulators on the boards are described in the Power Supply Considerations section of this document.

The serial communication lines Transmit Data (TXD) and Receive Data (RXD) on the ATMega328 are connected to two jumpers. These jumpers allowed the circuit debugger to change the configuration of which lines were going to be driven as a transmit line or a receive line.

The schematic for the central hub can be seen over the next two pages. Sheet one, shown in Figure 3-33, has the main processor ATMega328, and XBee controller, and voltage regulators. Sheet two, Figure 3-34 on the following page, has the W5100 Ethernet Chip, the line drivers for it, the POE-RJ45 Ethernet port.

Figure 3-33: First sheet of central hub schematic

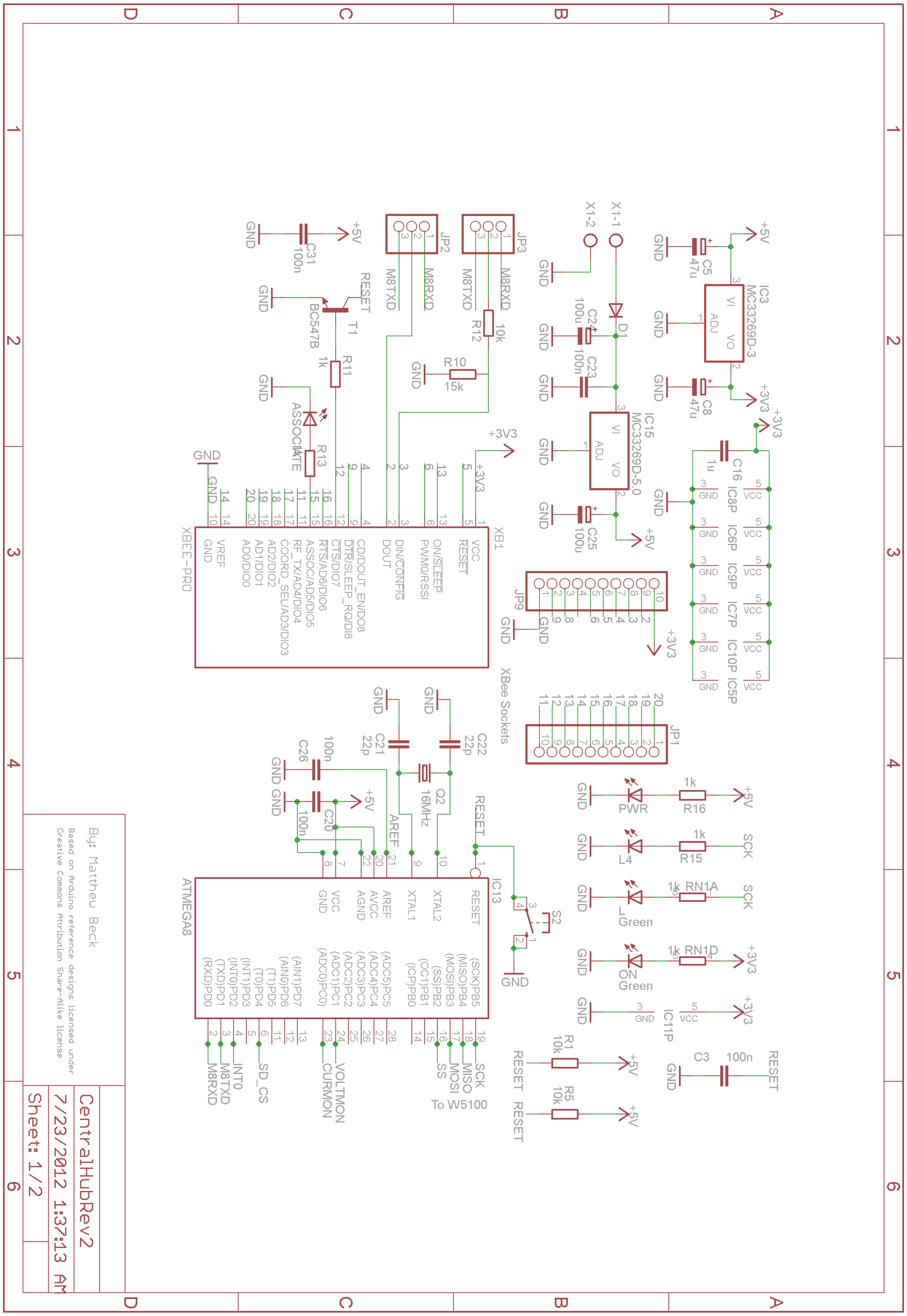
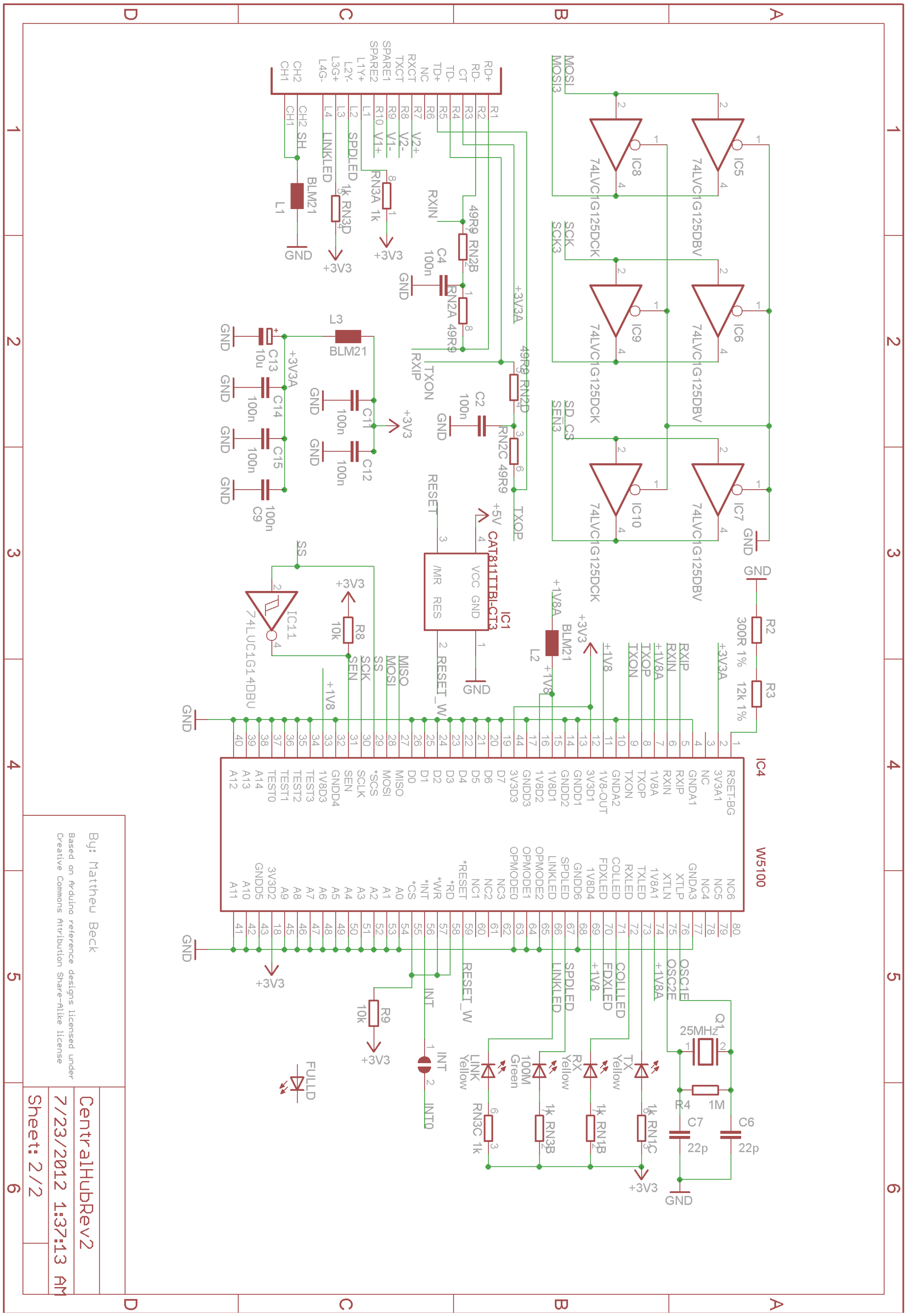


Figure 3-34: Second sheet of the central hub schematic



3.8.3 Ethernet Communication

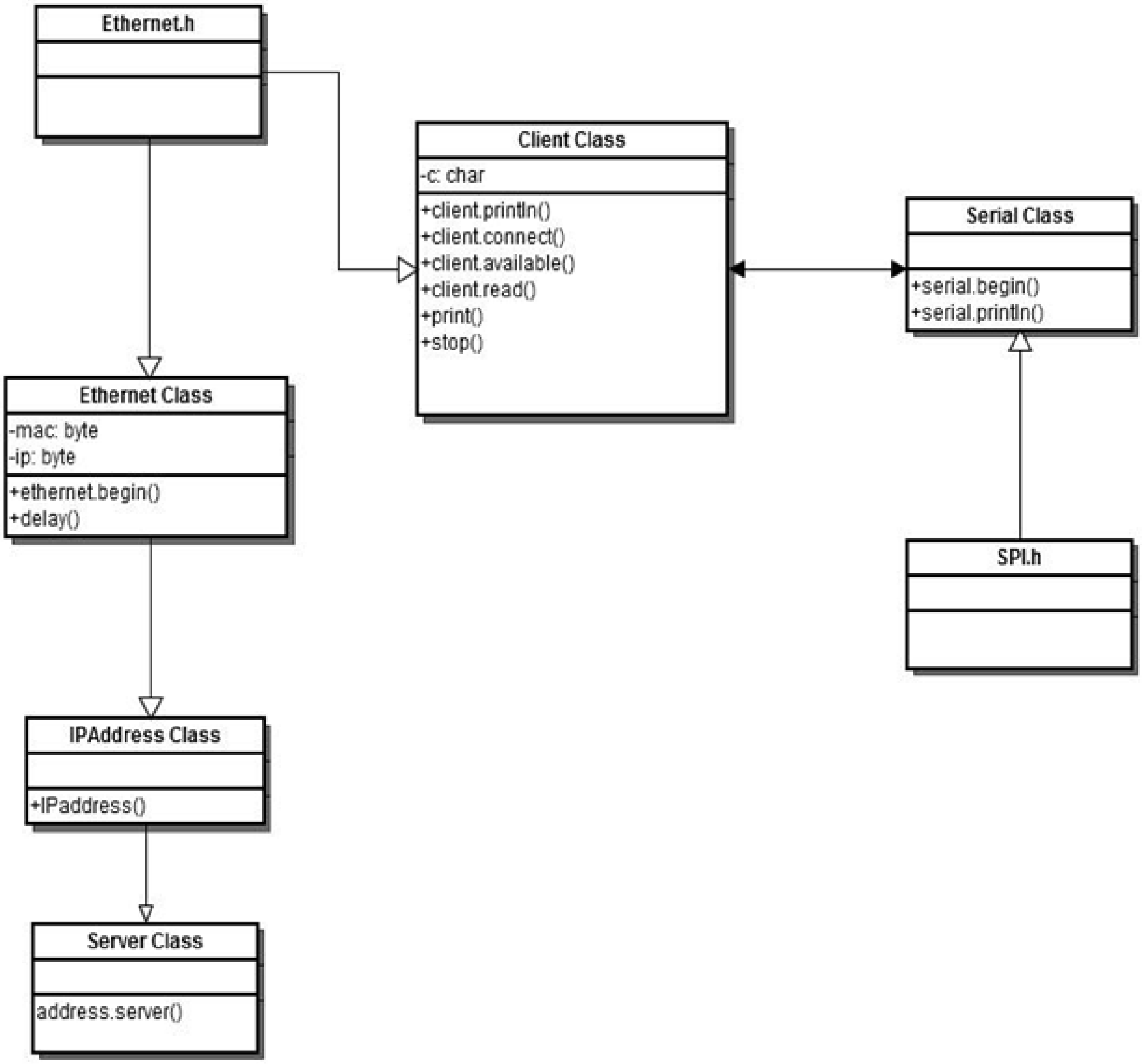
For the Ethernet shield the Ethernet library is one of the main resources.

In order to use the Atmega328 the group will need to program the device. The programming for the Atmega328 will be done using the arduino 1.0.1 software. The arduino1.0.1 is an open source environment designed for writing code and uploading to the i/o board of the Atmega328. The open source environment is written in java and can be run on Windows, Mac, and Linux. The Arduino1.0.1 software is downloaded off of the arduino.cc website. The arduino software contains an uploader which allows you to compile code and upload it to the I/O board. The processor will need to be programmed to communicate with the web and this software is also convenient for this specific purpose of the project. The program supplies an edit feature which allows the user to copy code to the user interface clipboard as HTML. This feature is suitable for embedding into web pages. Another convenient aspect of this software is the ability to compile in C. Since the group can code the Atmega328 in C it eliminates a huge learning curve for the group and allows us to simply focus on the libraries supplied by Arduino.

For this project the group will need to configure the processor to send and receive data from the internet as well as other devices within the Xbee network. The first task is configuring internet communication between the central hub and the website. The arduino1.0.1 allows you to make an HTTP request using the Ethernet shield which will be attached to the Atmega328 board. The purpose of the Ethernet shield is to connect the WizNet Ethernet controller to the Atmega328 via the SPI bus, which will use pins 10 11, 12, and 13. The only issue with the Ethernet board will be the physical connection of the Ethernet cable. The Central hub will have to be positioned close to an Ethernet outlet or the project will need a long enough cord to support a connection.

To access a specified website the code will be developed in c. Included in the programming process will be the libraries SPI.h and Ethernet.h. The code will require the mac address of the controller and IP address of the server for the website to be entered. Then the group can make use of the open source arduino libraries. The program is represented by a class diagram in Figure 3-35. This will represents how the group plans on using the atmega processor to access a webpage using the internet.

Figure 3-35: Ethernet connection class diagram (Reprinted from arduino.cc/)

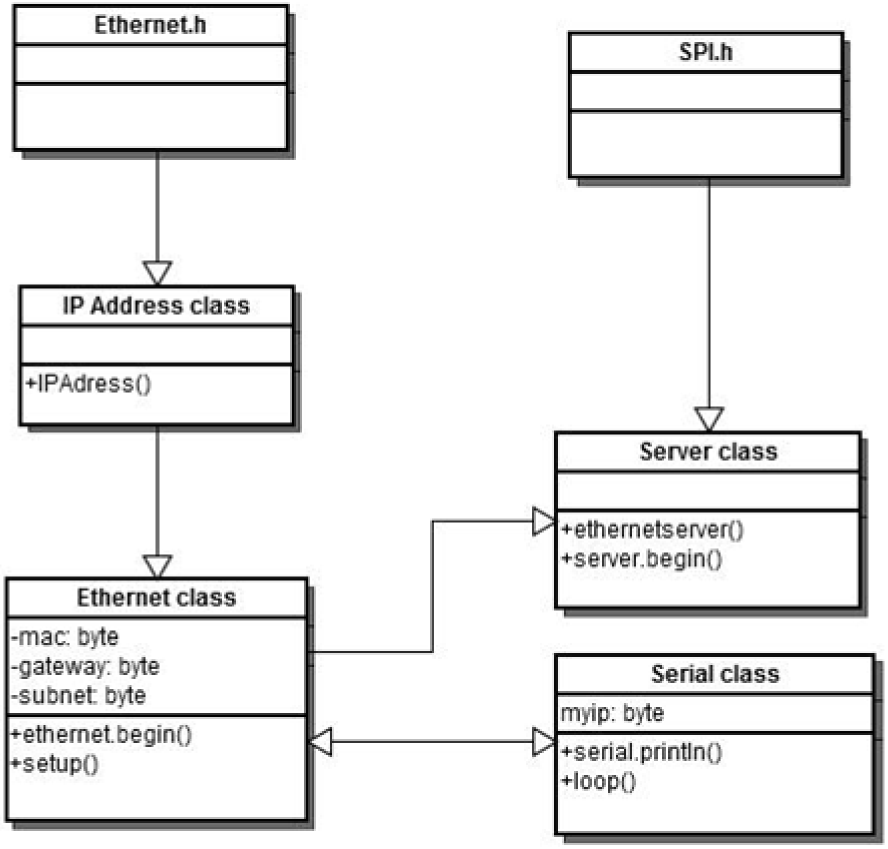


3.8.4 Xbee Networking

The Xbee shield is used to enable wireless communication with the Atmega328. It can communicate up to 300 feet outdoors(line-of-sight). The Xbees configuration involves connecting the device with the Atmega328 and Ethernet shield. The Xbee shield does not operate on its own. Attaching the xbee shield to the Atmega328 is the first step in setting up the network. Without the Atmega328 the xbee wireless communication cannot be configured. In order to set up communication between xbee modules, all modules must be on the same network. The ID parameters(0-0x16bits) need to be set and all modules must be set to the same channel(oxoB-ox1A). Every module has DH and DL parameters associated with the data destination address for the device. This address is composed of 16 bits. There are 3 main settings for the DH (0 or 1) which represents the first bit and the DL (represents last 16bits). The DL is used as a comparison value between devices. If DH = 0 and DL < oxFFFF then the data transmitted will be received by any module whose address matches the DL. If DH = 0 and DL = oxFFFF then all devices may receive data being transmitted. If DH !=0 and DL > oxFFFF then transmission will only be received by the device whose serial number equals the transmitting devices destination address.

The arduino1.0.1 has many functions to utilize for communication between devices and the internet. One function in particular that will be very useful to us is the IPaddress() function. The function is defined as: IP address(camera, hub, server) for communication purposes. Below is a basic example of its use in the arduino1.0.1 environment. The group plans on using the code from http://arduino.cc/en/Reference/Libraries when programming the network. The code is expressed in the class diagram in Figure 3-36 below.

Figure 3-36: Class diagram for XBee connection



3.9 Electric Door Lock Design

The Lock Controller schematic is very similar to the Central Hub Schematic. The same ATMega328 microcontroller is used as the central processor, the same MC33269D-5.0 and MC33269D-3.3 voltage regulators are used, and the same XBee chip/XBee connector is used. However, the W5100 Ethernet chip is nonexistent, being that the Lock Controller will not connect directly to the internet with an Ethernet connection. This obviously means that there will also be no POE-RJ45 Ethernet port, or buffers to adjust the logic level to 3.3 volts for a W5100 Ethernet chip that does not exist.

The additional hardware that the Lock Controller has that the Central Hub does not can be seen in section 5 of the schematic in **Error! Reference source not found.**. There are two screw terminals on a PTR Connector, part number AK300/2 located in the con-ptr500 library in Eagle. This is to connect the actual wires running to the electric door lock to the printed circuit board to allow for control by the ATMega328.

The type of electric door lock being used is an Electric Door Strike - Mortise Type. This particular electric door lock needs to be powered by a 5 volt power source, which is convenient being that the ATMega328 is also powered by a 5 volt power source. The typical operating current for this device is 1.5 Amperes. This current greatly exceeds the absolute maximum current rating of 40 milliamperes which can be sourced by the ATMega328. The most electrically intuitive way of addressing this problem is to simply add a metal–oxide–semiconductor field-effect transistor (MOSFET) to switch the power to the door lock on and off.

The metal–oxide–semiconductor field-effect transistor chosen for this task is the BUZ10 N-Channel Enhancement MOSFET, which is rated for 50 volts and a 20 ampere current. This greatly exceeds the load requirements presented by the Electric Door Strike - Mortise Type which has been implemented into the design. Being an N-channel transistor, the electric strike can be wired to the high side of the voltage drop. The logic control for this configuration is a high voltage on the first pin of port C (PC0) to open the lock, and a low voltage on the same pin to close the lock.

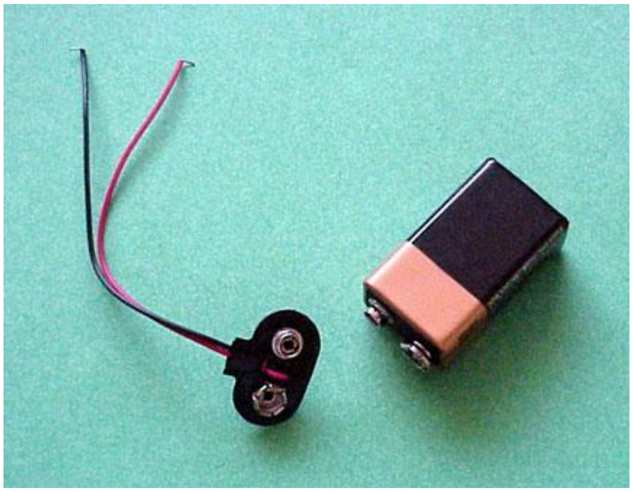
3.10 Power Supply Design

Since a steady 5 volt direct current signal was required by the circuit, a voltage regulator had to be chosen that could provide this consistently. The MC32269 voltage regulator series provided low dropout, positive fixed and adjustable voltage regulators. There was a 5.0 volt output version, and a 3.3 volt output version, which would supply the whole project with all of its voltage regulating needs. The MC32269 voltage regulator series also had a 1.0 volt dropout voltage, which would necessitate a minimum of a 6.0 volt input for the 5.0 volt regulator, and a minimum of a 4.3 volt input for the 3.3 volt regulator.

The voltage regulator that required a minimum of 6.0 volts at the input was powered easily by a 9 volt battery, shown with connector in Figure 3-37. This allowed a 3 volt margin of error on either the battery side, or the voltage regulator side of the equation. Furthermore, with a 9 volt battery, there was no need for a battery holder, since the tops of these batteries have terminals that can snap to specialized connectors, and the battery itself could be placed wherever it fits. An example of this connector can be seen in **Error! Reference source not found.**. 9 volt batteries are very common batteries that can be found in most stores, so it would serve as ideal option for most end users. This connector was purchased separately very inexpensively, and connected to the board by solder point where the 9 volt signal can be fed to the initial voltage monitor.

Another positive aspect about the 1.0 dropout voltage on these components was that the 3.3 volt voltage regulator could be powered by the 5.0 volt output from the other regulator, which would be a consistent, regulated voltage which is sufficiently above the required minimum 4.3 volts, but not so much higher that it will waste large amounts of energy.

Figure 3-37: 9V battery with connector



4 Prototyping

4.1 Hardware Prototyping

There were several different components that made up this project. Along with the central hub, there were four potential network nodes at the moment: these included, but were not limited to, an outlet monitor and controller, an Internet Protocol (IP) camera, a circuit breaker monitor, an electric door strike, and a light switch controller. Each of these devices had their own dimensions, which were relevant considerations as they relate to packaging considerations and logistics. Minimization of these dimensions was a design goal, but was secondary to the actual functionality of these devices.

Along with the electronic devices themselves, there were batteries that had to be included in each final prototype, an Ethernet cable to connect the central hub to a router or modem, and other essential components required for mounting and installation for each network node. Some wiring solutions, including the XBee communication networks and the Internet Protocol camera, reduced the need for hard wiring, and minimize the final packaging dimensions of the final product under development. The outlet monitor and controller necessitated a three prong AC power socket for which to plug in the end devices that served as the target for monitoring and control.

Once the final board dimensions were compiled and laid out, four drill holes were added to the board in each corner in order to mount the PCB to the final project enclosure. This took up valuable board real-estate, and was taken into consideration when designing the final Printed Circuit Boards (PCB's). The sizes of the drill holes were dependent upon the selected project enclosure, and the size of the project enclosure was dependent on the size of the final printed circuit boards; thus, developments for these factors were taken into consideration simultaneously.

The drill holes in the final printed circuit boards were large enough to fit the proper screws through, but not so large that the screw head failed to lock the board in place securely. The drill holes were also spaced in such a way that the final printed circuit board could be mounted properly into the project enclosure. Another issue that presented itself when fastening the printed circuit board to the project enclosure with metal, conducting screws was that this created the potential for short circuiting. One strategy was to have the screws floating, not connected to any net. This prevented drainage of any electric charge build-up from these floating nodes, which presented safety issues. Another way to manage these issues was to connect the screws to the ground network, or a ground plane. This allowed any built-up charge to quickly and safely dissipate through the ground network. This is a common solution for many applications, and was a proper solution for the project under discussion.

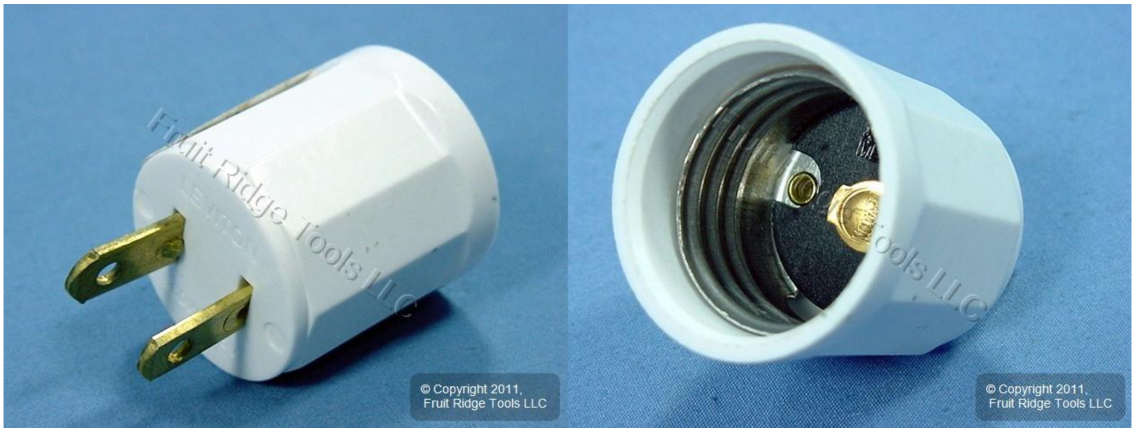
The project enclosures had dimensions which were large enough to house its corresponding printed circuit board and any other components or wires that are necessary for that particular node as well. There were screw mounts with the exact dimensions as those on the printed circuit board; these mounts were positioned so that the printed circuit board was not flat against any surface. This was done with spacers, and properly sized nuts.

The final prototype of the Central Hub needed to be directly connected to a router or modem. There were two ways to make this connection with an Ethernet cable. The easy way was to have an Ethernet cable sticking directly out of the project enclosure through a grommet. This method has many disadvantages, however. First of all, the cable would not be easily replaceable, which would be a significant disadvantage. The second, less significant disadvantage is that the device could not be stored as easily as it would if the Ethernet cable could be detached. Another, more desired approach was to have the Ethernet socket, which will be mounted directly to the board, to be available on the outside of the device. In order to make this interface flush, the Ethernet socket would have to hang off the board slightly. These measurements required precise measuring, and calculations of offsets. To be sure the dimensions were correct, the project enclosures were purchased first, measured with calipers, and finally the final printed circuit board could be laid out accordingly.

The final prototype for the Outlet Monitor and Controller had a 120 volt AC power plug directly mounted on one side, and a 120 volt AC power socket directly mounted on the other side. When designed in this manner, the device was able to plug directly into the wall, and any electric appliance which would have plugged directly into that same wall, could now plug directly into the prototype.

The final prototype of the Light Controller was very similar to the Outlet monitor and Controller. It, too, had a 3 prong AC power plug, but did not have an output 3 prong AC power socket. Instead, the Light Controller had a switch and a light fixture. The light fixture was made with a PVC frame, which fastened into a wall with angle brackets. In this scenario, the wires bringing power to the light socket simply run through the PVC piping. The light socket shown in **Error! Reference source not found.** fit snuggly into the end of a PVC pipe. This obviously means that the light socket was wired before inserting it into the PVC pipe. No lampshades or opaque glass were implemented as part of the project; the main focus of the project was for electrical design, not interior design.

Figure 4-1: Light socket to fit into PVC pipe



The final prototype for the Electric Door Strike was designed so that the controller box could be accessed easily from the inside of the door. This would be useful for changing the battery, and easily updating the software and/or installing additional features such as key card reading and biometric scanners. This is an issue when security comes into play, so it was kept in mind when designing the features. With customizability comes hack-ability, so a secure system would be completely closed, with no programming ports, and a microcontroller that is soldered to the board, so it is not removable like the others. The door jamb was fitted to properly house the Electric Door Strike. These dimensions were carefully measured with calipers during development, and documented in the final report.

Additional wiring was necessary for the outlet monitor and controller, the circuit breaker monitor, the electric door strike, and the light switch controller. It was, however, advantageous to reduce the amount of wiring as much as possible to simplify manufacturing and assembly processes. For instance, the Honeywell CSLA2CD Hall Effect current sensor could be loosely packaged in the final assembly, and connected to the printed circuit board with wires, or it could be physically mounted to the board. The second option is clearly the optimum solution, and was implemented in the final prototype. The dimensions of the Honeywell CSLA2CD Hall Effect current sensor were noted in the datasheet, downloaded in the Drawing Exchange Format (DXF) data file, and measured physically with calipers upon product arrival. A combination of all of these methods were utilized, given the lack of experience regarding this aspect of development, and narrowing any margins of error that could prove to be costly in terms of both time and money. Once the dimensions for the terminals on the Honeywell CSLA2CD Hall Effect current sensor were known, corresponding solder pads were added to the board to provide the mounting capabilities desired for the final prototype.

The Outlet Monitor and Controller needed wiring between the various components that make it up. These components consisted of the main circuit board, the 120 volt AC power plug, the 120 volt AC power socket, battery, the relay and the current monitor. Each of the wires for these had specific current specifications. The wires running to between the 120 volt AC power lines were capable of handling a 20 ampere load. According to the 2008 Nation Electric Code, all 20 ampere networks must be wired with 12 AWG wire or larger, hence, 12 AWG wire were used for the 120 volt AC power lines. The wire diameter was much smaller for the wires connecting the 9 volt battery and the analog and logic lines between the relay and notional current monitor, none of which exceeding 50 milliamperes under max current conditions. These lines were wired with 24 AWG wires.

A grounding plane was implemented in each prototype, either as a metal, conducting shield fastened to the apparatus, as a copper plane poured onto the circuit board, or even both. This grounding plane was directly connected to the ground prong that plugs in with the 120 volt AC power lines. There was also a grounding socket for the power output, allowing the end device to be connected to the earth ground as well through the grounding plane. This grounding plane was the best approach to reduce the possibility of ground loops, which could have negative effects, compromising the system.

During the development phases of the project, it was advantageous to cut an extension cord in half, providing an AC power plug from which to obtain a 120 volt AC power signal, and an AC power socket from which to output the monitored and controlled 120 volt AC power signal going to the end devices.

4.2 PCB Prototyping

When the group initially set out to design a Smarter House, they wanted the devices and overall design to be as clean as possible. They wondered why they couldn't have full control of their house from their mobile phone. They then wondered why they had to always carry around a key to open their front door. They grew tired of always worrying about what their power bill would be at the end of each month. All of these wonders and worries led them to the idea of designing devices to eliminate "all of the above"! Part of the vision they had was that they would design devices that would be as "free standing" as possible, and have as minimal a package footprint as possible. This "minimalist" approach led them to combine all of the circuit devices they had been learning about, with the processor and wireless technology the group members have seen around the internet into several schematics that accomplish their various design problems.

Once designed, they began to think about ways to take the group's circuits and make them as small as possible. This led them to the investigation of various companies who promise fast, cheap, and concise printed circuit boards based off of the various device schematics. When they began to research the apparent leaders in all three of those promised fields, they came across: PCB123, Custom PCB, and Express PCB. One thing became clear: all of these companies offer the chance to layout the various boards as the group members want, and enable the group to receive the smallest package possible for all of the devices. In order to properly compare these companies, the group will set the design needs as follows: simple design, 3 Copies of the same board with silk screen for simplicity, and 2 layered boards (3 by 3 inches in size). It's also important to note that the group must factor in the required board designing software (or lack thereof) and the ease of use pertaining to the program (along with ease of design of the boards).

The first company they investigated was PCB123. This was the company they had heard about from past senior design classes and was thus a front runner. When they began to search the site for information regarding printed circuit boards they found that the company offers four levels of pricing: ValueProto, PCBexpress, Full Feature, and custom Quote. ValueProto obviously was the cheapest of the four options, and thus immediately caught the group's eye, leading the group to investigate this price level first. The company states that this level (summarized in Table 4-1) is for those that don't require a quick turnaround time, and desire low quantities of two sided boards with low pricing. They then entered in the dimensions and received an immediate quote: 146.00 (with silk screening listed as an option, which means more money). They were then prompted to enter in the Gerber and Excellon files, which means that they don't require the use of their programs (which is a huge advantage to the group because of the fact they will be designing their boards using EagleCad). This level also promises a ship date of two weeks, with the boards being shipped via a ground carrier method which undoubtedly adds 1-2 weeks of additional ship time.

Overall, this method meets the group’s requirements at the lowest price offered by the site, and thus will continue the investigation with “Custom PCB”’s site.

Table 4-1: Summary of PCB 123

Criteria	Met?
Quick Turnaround	No (3-5 weeks)
Low Price	Yes (105$)
Simplicity of Design	Yes
Silk Screen	No (Extra)

The second company they researched was Silver Circuits at CustomPCB.com. This company offered two different levels of board manufacturing: Production and Prototyping. The production level had a minimum order of 100 boards, which is obviously 96 more boards then they need. It is important to note that though they required 100 boards to be bought, they only charged about 4 dollars per board with a onetime fee of 75 dollars (making this a great choice if the group decides to take the devices to production) and promised a speedy turnaround time. The second level they offered was that of Prototyping. This option allowed for two layered boards to be printed (up to 6 by 4 inches in size), with a minimum purchase quantity of only four boards. Along with a larger than needed board, they offered free silk screening on the top layer of each board. They promised a one week lead time, meaning that the boards should arrive within 2 weeks of the day the group were to order them. On top of all of this, they do not specify which software to use, and thus the group would be able to use Eagle to design the schematics and boards. Once the group members were quoted a price (within a minute), they were guaranteed 4 boards within two weeks for only 85 dollars! All of these features, both for prototyping (summarized in Table 4-2) and for production, helped the group rule out the use of PCB123 for board manufacturing, and place CustomPCB at the top of the list! Before they made up their mind on which company to use if they decide to ultimately have the device boards produced with, they must discuss the final option: Express PCB.

Table 4-2: Prototyping with Custom PCB

Criteria	Met?

Quick Turnaround	Yes (2 weeks)
Low Price	Yes ($85)
Simplicity of Design	Yes

The final site the group will be considering for the probable production of the group's prototype boards is that of Express PCB. Express PCB offers six different levels of board manufacturing: Standard, MiniBoard, Production, 4 Layer Production, ProtoPro, and 4 layer ProtoPro. Since the group members are not interested in 4 layered boards, they will only consider the use of MiniBoard, Production, Standard and ProtoPro. The first to consider is the lowest cost level: standard. Standard offers double sided boards, no silkscreen, and fast turnaround time (orders less than 50 are shipped the next business day! The price is based on a formula, which when the group chose enter in a quantity of 3 boards and 9 square inches per board, the group was given a total of 100 dollars (after shipping charges). This price guarantees the group same week delivery because of the quantity required and the overnight shipping. The next most affordable option is that of MiniBoard Service. This level guarantees the group a 3.8*2.5 inch board in a quantity of 3, a silkscreen, and a 4 day turnaround for 85 dollars which is surprisingly cheaper than the lower level ("Standard"). When the group researched the production and ProtoPro levels, they are both the same level of service as the MiniBoard level except that the ProtoPro offers one more board for around 70 more dollars, and the production level costs around 300 dollars for up to 10 boards. All of these features (summarized in Table 4-3) combined make this site a valid contender for the prototyping of the group's boards, but is effectively ruled out if the group decided to take their boards to production.

Table 4-3: Prototyping with Express PCB

Criteria	Met?
Quick Turnaround	Yes (<1 week)
Low Price	Yes ($85)
Simplicity of Design	Yes
Silk Screen	Yes

Now that the group members have effectively researched the leading PCB manufacturers, they are able to make a conclusion about which service they will be using if they decide to have the group's boards manufactured. The site that appealed to the group most was definitely Custom PCB. Custom PCB met all of the group's basic needs, and gave the boards 4 more inches of space with which to lay out the various device's boards. This extra space, coupled with the quick turnaround, low price, and freedom of design has led the group to the decision to use their company to prototype the group's 3 PCB's (which the group will be designing via Eagle's Cad software)!

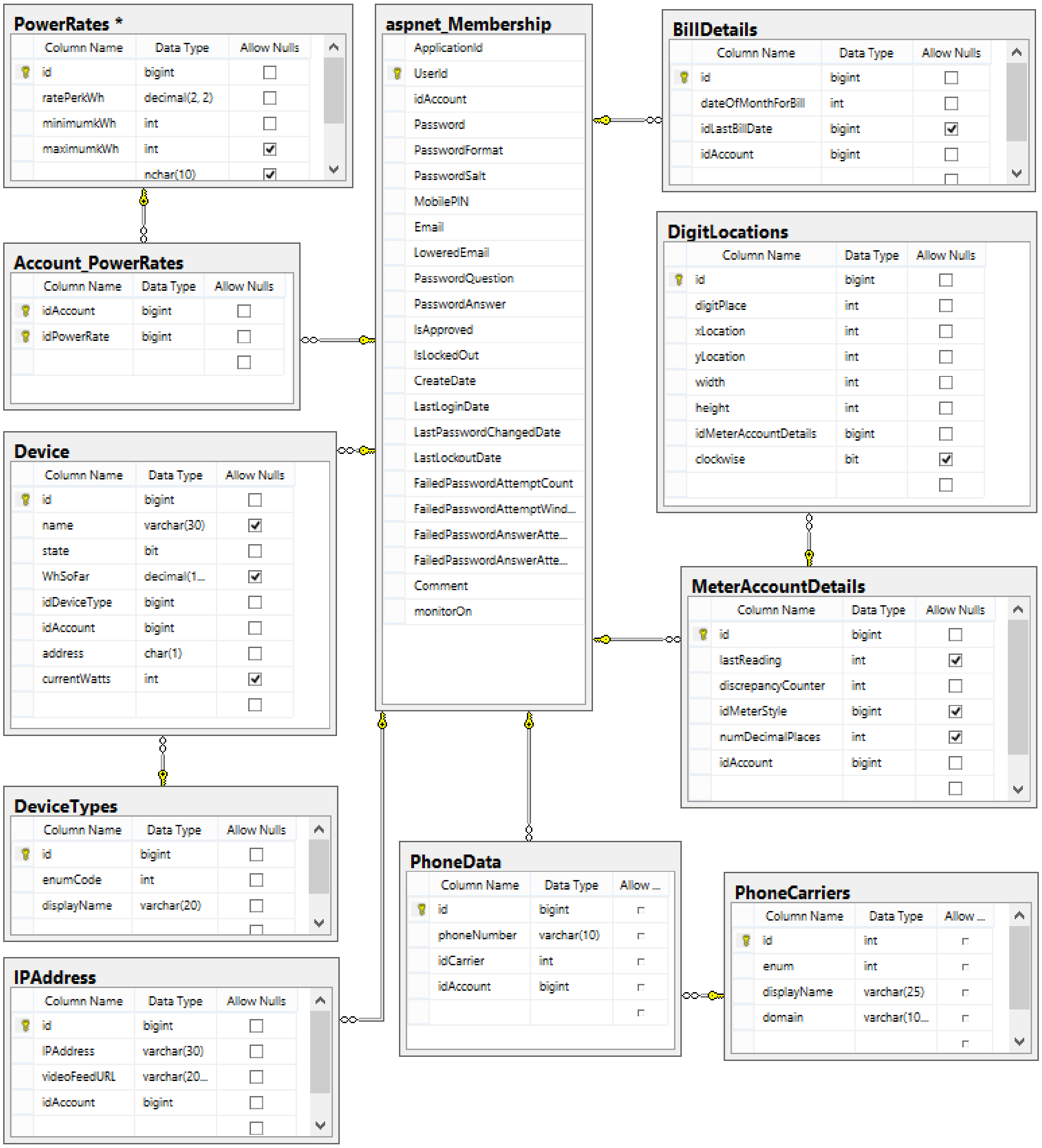
4.3 SQL Database Prototyping

The design section explained the data that needs to be stored and which items will be grouped together into tables. For prototyping, the group members provided a diagram of the database schema, including the tables, their columns and the data types of those columns, which columns are primary and foreign keys, and the relationships between tables. Foreign key columns are named in the format id[Table Name]. This diagram is in Figure 4-2.

Connecting to the SQL Server database should be fairly straightforward. Both ASP.NET and PHP offer pre-existing methods for connecting to SQL Server, so these will make connecting from the website and the image processing software easy.

For ASP.NET, there is a SqlClient namespace that includes several classes for interacting with SQL Server, including SqlConnection to open a connection and SqlCommand for building complex commands with input and output parameters. The SqlConnection class can take in a large number of parameters, but the ones that matter for this project are the server name, user ID, password, and connection timeout.

Figure 4-2: A diagram of the SQL Server database schema for the project. This database will hold data for the website and the image processing software.



PowerRates *

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	ratePerkWh	decimal(2, 2)	☐
	minimumkWh	int	☐
	maximumkWh	int	☑
		nchar(10)	☑

aspnet_Membership

	ApplicationId
🔑	UserId
	idAccount
	Password
	PasswordFormat
	PasswordSalt
	MobilePIN
	Email
	LoweredEmail
	PasswordQuestion
	PasswordAnswer
	IsApproved
	IsLockedOut
	CreateDate
	LastLoginDate
	LastPasswordChangedDate
	LastLockoutDate
	FailedPasswordAttemptCount
	FailedPasswordAttemptWind...
	FailedPasswordAnswerAtte...
	FailedPasswordAnswerAtte...
	Comment
	monitorOn

BillDetails

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	dateOfMonthForBill	int	☐
	idLastBillDate	bigint	☑
	idAccount	bigint	☐
			☐

Account_PowerRates

	Column Name	Data Type	Allow Nulls
🔑	idAccount	bigint	☐
🔑	idPowerRate	bigint	☐
			☐

DigitLocations

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	digitPlace	int	☐
	xLocation	int	☐
	yLocation	int	☐
	width	int	☐
	height	int	☐
	idMeterAccountDetails	bigint	☐
	clockwise	bit	☑
			☐

Device

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	name	varchar(30)	☑
	state	bit	☐
	WhSoFar	decimal(1...	☑
	idDeviceType	bigint	☐
	idAccount	bigint	☐
	address	char(1)	☐
	currentWatts	int	☑
			☐

MeterAccountDetails

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	lastReading	int	☑
	discrepancyCounter	int	☐
	idMeterStyle	bigint	☑
	numDecimalPlaces	int	☑
	idAccount	bigint	☐
			☐

DeviceTypes

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	enumCode	int	☐
	displayName	varchar(20)	☐
			☐

PhoneData

	Column Name	Data Type	Allow ...
🔑	id	bigint	☐
	phoneNumber	varchar(10)	☐
	idCarrier	int	☐
	idAccount	bigint	☐
			☐

PhoneCarriers

	Column Name	Data Type	Allow ...
🔑	id	int	☐
	enum	int	☐
	displayName	varchar(25)	☐
	domain	varchar(10...	☐
			☐

IPAddress

	Column Name	Data Type	Allow Nulls
🔑	id	bigint	☐
	IPAddress	varchar(30)	☐
	videoFeedURL	varchar(20...	☐
	idAccount	bigint	☐
			☐

4.4 Image Processing/Power Monitoring Software Development

The image processing software functions as described in the design section, 2.6.2. For the prototyping, some of the functionality is described in detail to show exactly how and why it should work. The main functions to be covered are the digit reading functionality, and the outlet power monitoring, as well as touching on the state monitoring for each device.

The software was written as a Windows service in C#, using Visual Studio 2010. As a service, there is a main loop that runs over and over again in five-second intervals. This loop checks the SQL database to get each account stored in it, and then runs through a loop for each account. In this loop, it checks if the account has set monitoring service on or off since the last time the loop ran, and only does something if the state has changed. If monitoring is being turned on for the first time, it creates an instance of a meter reading object class and adds it to a list of meter reading objects that the main loop keeps track of, and then it sets a variable in that turns the meter reading loop inside that object on. If the service was previously on for this device but is now off, it just sets the variable to turn the service back on. It only does this if all of the necessary info for meter reading has been provided to the system by the user, otherwise it moves on through the loop. Next, it does the same thing with a device monitoring object class, creating it and turning it on for first time use, or just turning it back on if it was previously on, but only if the necessary data for device monitoring has been entered. When the entire Windows service itself is stopped, if stops all monitoring objects and deletes them before ending.

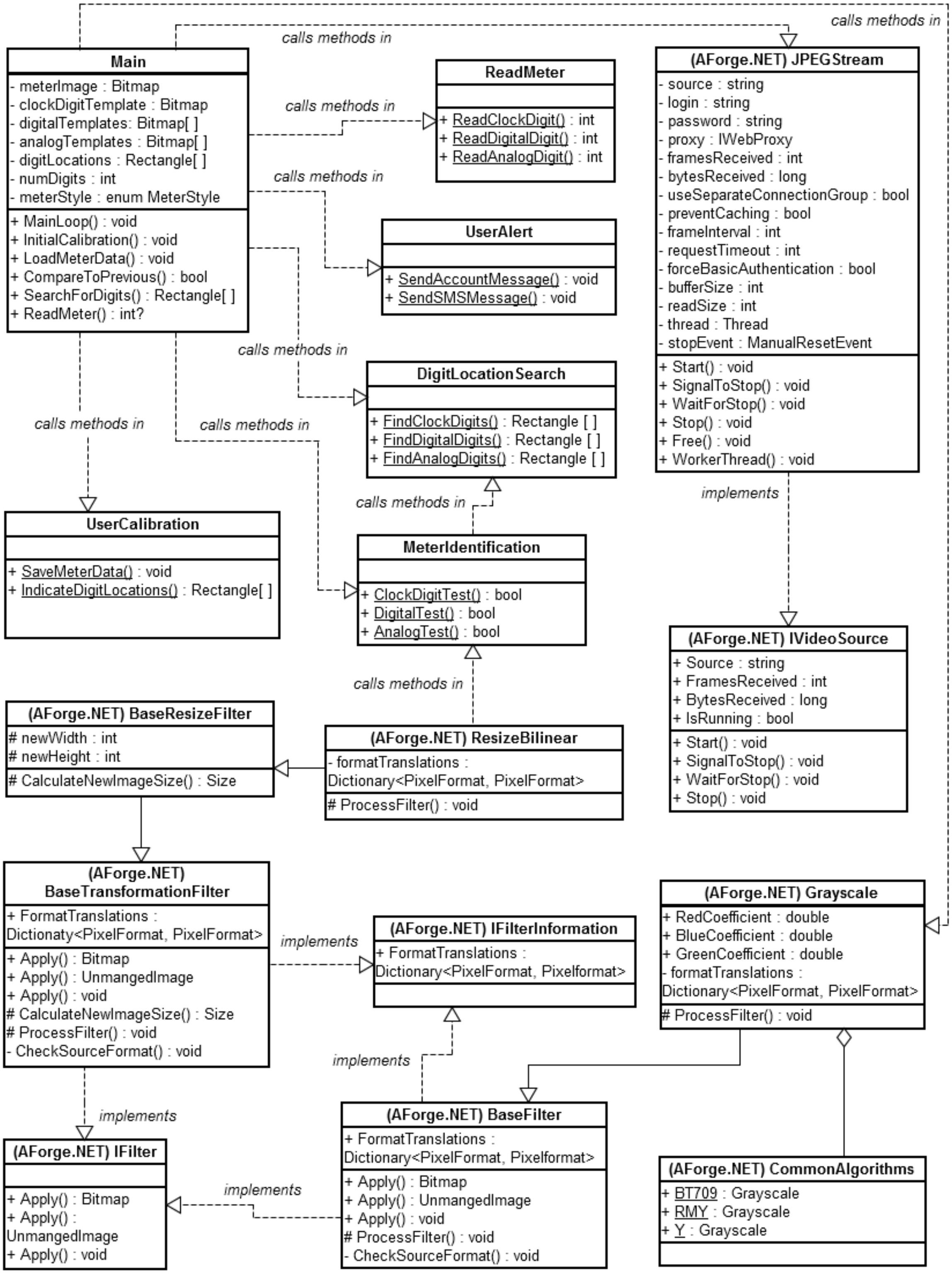
The meter reading object class contains a background worker that runs on a different thread, allowing it to run while the main loop also continues to run. In this background worker, there is a loop that runs once per minute to read the meter. The loop runs only once per minute to reduce the stress caused to the system as a whole, since each account has a background worker thread running. Because of this minute delay between reads, it is possible that the bill might not be completely real-time, but since meters increment very infrequently, it will not make much difference overall. Within the loop, the software performs its process of getting a frame from the camera, ‘reading’ the digit at each digit location as defined in the database, comparing the new reading to previous readings to check for erroneous readings, and if all is well, saves the reading to the database for the website to use. This process is described in detail in the image processing design section, 3.7.2.

Similar to the meter reading object class, the device monitoring object class also has a background worker that runs once per minute. The device monitoring class checks the database to get info on each device that the user has registered on their system. First, it checks the outlets. Because of the inconsistency of the current sensor, the software gets ten current readings from the outlet and uses the average as the current reading. It then gets a voltage reading, which is much more consistent. It uses these two readings to find the current power use at the moment. The object stores the last time that a power reading was obtained, and to get a watt-hours usage for the outlet, it multiplies the watts currently used by the time since the last reading (in hours), and just assumes that it has had that watt level for the whole time. Since it is only a minute between checks, and the system is only meant to provide an estimate, this is good enough for our purposes.

Other than the power reading for the outlet, the loop performs the same operations on all of the device types. It checks the state and compares it to the previous state, which it has stored. If the state has changed, it checks if the user wants to be notified of a change for that device, and if so, it sends the text notification as an email. Every time the loop runs, it gets all device information again, so the user does not have to stop and start monitoring again any time they make a change to the system configuration.

These few details of the program should provide insight into the most complicated part of the software. Everything else – the image retrieval, main program loop, grayscale conversion, user alert, SQL commands and all of the other little things – should all be pretty straightforward, taking more time to actually write than to design. As such, showing pseudo-code for those parts would be a waste.

Figure 4-3: Class diagram for the image processing software



4.5 Presentation Prototype

For the overall demonstration the group plans on building a simple box with several compartments or rooms. Each compartment will hold a different device.

The group will try to replicate a miniature home environment where several devices are within a reasonable distance. This box will contain a door lock with a latch system, push button light, and a power outlet with an external device plugged in. Several devices will be demonstrated with the outlet to show a difference of power consumption. Since the system needs to mimic the floor plan of a home, the group wants to place all the devices in a manner to simulate a real house. The lock will not just be a stand-alone device.

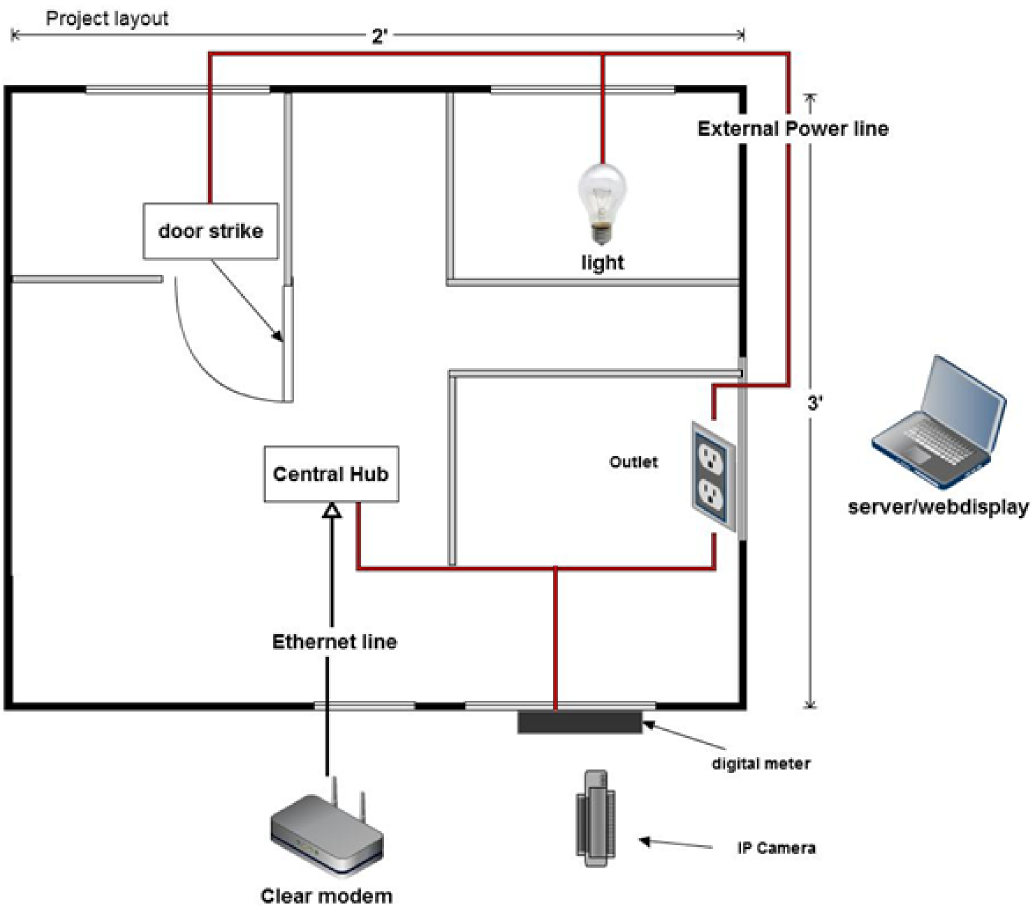
The plan is to construct a door model within the display. The lock will have a deadbolt with a strike. The deadbolt will be attached to a wall leading to one of the rooms of the display. When the strike is signaled the section of the wall will be designed to open. The lock will also have a key option for locking and unlocking. The light bulb will function in a similar manner. The light bulb will be attached to a wall. Next to the light will also be the push button mechanism. The light will be operational from the physical button as well as the website. The presentation will also include the central hub, which will be the center of the system. Attached to the outside of this box will be a small replica of a digital meter. The IP camera will sit outside of the box with the lens facing the digital meter. The router will sit outside the system with an Ethernet connection leading through the wall and to the central hub. Not included in the home replica, but still a part of the demonstration will be a laptop and a smart phone of our choosing. The website will be demonstrated on a laptop as well as the android app for the smart phone. The laptop is also necessary for the system to work. The group has set up a server on the computer. The system will not work without the laptop being on during the presentation. The group plans on making the dimensions of the display to be approximately 2'X3'X6". Throughout the floor plan power will need to be supplied. There will be an external cord running through the house to power all the devices. The floor plan is subject to change due to the fact that the group has not acquired all parts as of yet and will still need to decide what size is most feasible. A design layout is displayed in the figure below.

The group would like to make this home automation project mobile for demonstration purposes. It is imperative for the project to have internet access at the time of demonstration. The home network needs to be operational regardless of location. To do this it would be ideal to have a consistent internet connection to one modem throughout the testing and demonstration procedure. There are several options for a cable modem. The option that stands out the most is Clearwire which offers access to a 4g mobile network. The devices offered by this company can work in any location. A clear modem device can grant the subscriber internet access without having to do anything other than plugging into an outlet. It is essentially a plug and play modem with no need for a cable or phone outlet. The group can basically set up a wifi network without resetting a modem or plugging into a cable or phone line. This is convenient for the project because a great deal of the programming relies on network addressing. Using the same router for testing and during the project can allow the group to avoid resetting the system and any new network configuration for the devices. The system becomes truly mobile with this adjustment in the modem choice. In order to use the clearwire service a subscription for a device must be purchased. Fortunately, the group already has a member which owns a device with these capabilities and will offer use of the modem for the project presentation.

The project has a specific component which posses a unique problem for presentation purposes. The image processing procedure requires a power meter. During testing and evaluation, the group can simply access any home power meter. The convenience of an IP camera allows performance tests without purchasing or building a device to replicate the display of a power meter. Unfortunately the group cannot access a real power meter during the presentation of the project. This situation requires us to somehow replicate the digital reading for mobile purposes. The group needs a replication of the meter which will be within a close proximity of the rest of the project to demonstrate the use of the system. For this, the plan is to build a digital display. The digital display will replicate a power meter outputting the same amount of digits as a household meter. The group wants to be able to control the display as well. For the demonstration the group will increase the digital display and show how the image from the IP camera changes the value displayed on the website. The display will sit on the outside of the house model as a real meter would.

The presentation will start with a demonstration of concept. The group will first run through the individual devices and run physical testing of the equipment before demonstrating wireless functionality. The mechanics of the lock will be demonstrated using a key to open and close. The light and outlet will also be demonstrated. The group will then display the functions of the website. The outlet connection will be alternated with different devices to show the changes in output on the website. The functionality of switch buttons on the webpage will be displayed along with the user account option. A test run will be simulated for a new user where an account will be created. Once the account is created the sms function can be demonstrated along the power bill display. The group will vary the value of the sms trigger and also the value on the digital display for the IP camera. All of these demonstrations can be replicated on a smart phone as well.

Figure 4-4: Layout of presentation prototype


Project layout
2'
External Power line
light
door strike
3'
Outlet
Central Hub
server/webdisplay
Ethernet line
digital meter
IP Camera
Clear modem

5 Testing

5.1 Outlet Monitor Testing

The process of integration for the Outlet Monitoring and Controlling device is able to be broken down into many small steps. The first step will be to wire and verify the voltage divider to an AtD pin on the development board. The group will then provide stable power to the current sensor, and connect to a second AtD pin on the development board. Once completed, they will then calibrate each of the sensors separately. The final step in the integration of this device is the verification of correct wiring of the relay, then placing the output into the voltage divider mentioned earlier. Although this sounds somewhat straight forward, the group will be breaking each of these steps down into many smaller circuits and steps, and will thus discuss this process.

The first step in the integration of the outlet monitor and controller is the wiring of the voltage divider circuit. The group members intend to wire the voltage divider and safety circuit on a breadboard and provide a DC voltage of 10V. Once they can verify the ratio they have attempted to achieve (120:3 (volts)), they will then wire an AC signal of 2v (the largest amplitude that can be achieved in the labs at UCF) and again verify. Next, they will replace the 2 volt AC input with the wires they will be using to connect to the outlet receptacle to the voltage dividing circuit. On the other end of the wires which will now be connected to the voltage divider, will be a 3 prong outlet plug. Once connected, they will turn off the power to a 120V outlet located in the lab (or a team member’s house) and will plug in the 3 prong plug. They will proceed to turn on the power to the outlet and carefully check the voltage located across the Zener diode and verify it is between 2-3 volts (AC). Once verified, they will again kill power to the circuit and run a wire from the Anode of the Zener to an analog to digital pin on the Atmega. Once wired to the Atmega, they will have the software print voltage seen by the device to the screen (using the code shown in the design portion) and verify that is provides the group a somewhat stable voltage between 115 and 125 volts.

The next step in this process will be to wire the voltage regulator to the current sensor, and apply power. The group will then connect the voltage divider (noted in the design portion of the current measurement) to the output of the sensor, and will pass the voltage across the smaller resistor to another analog to digital pin on the development board. They will then proceed to calibrate the current sensor output by obtaining the voltage output given when no wire is run through the sensor, and when a wire with a known current is running through the sensor. Once calibrated and verified, they will then complete the integration process with the final step: adding the control element to the circuit.

The final step in the integration and testing process for the outlet monitor and controlling device is that of verifying the relay is correctly wired. This can be done by first wiring a small signal input to the “normally closed” path of the relay, while wiring the “normally open” portion directly to ground. Next, they must connect the surge protecting diode across the relay, and the switching transistor (to allow control of the relay) to the coil of the relay. Once connected, they will connect loose wires to the output of the relay, and provide a voltage to turn the transistor on. They must then verify that the output voltage is zero (meaning that the relay has been switched successfully). They will then disconnect the voltage source from the transistor and verify that the output of the relay is now the small signal input to the relay. Once verified, they can now wire an output pin of the Atmega to the corresponding location on the transistor, and then connect the relay to the wires coming from the plug used in the calibration of the voltage measuring circuit. This will then finally be tested by placing a one on the corresponding output pin of the microcontroller, and verifying that the ATD connected to the voltage divider shows a voltage of 0. If the voltage is zero, the outlet monitor and controller and been successfully integrated!

5.2 Light Control Testing

The process for the integration and testing of the light controlling circuit was extremely similar to that of the outlet monitor (and controller) device, with two key differences: a physical switch, and an output to a 120V (AC) light socket. Still, it was important to provide a detailed procedure which will allow for successful implementation of the device! The first major step in the process was to verify the correct hookup of the relay. The second step was to wire the switch to the corresponding Atmega pins. Finally, the group connected the light controlling circuit to the wires running to a live light socket (obviously switched off via the circuit breaker).

First, the tester must verify correct wiring to the relay. Similar to the procedure for verifying the outlet monitoring and controlling circuit’s relay, they will wire a 120 V(AC) input to the input of the relay. They will then connect the normally closed path to an output wire, and will connect the normally open path to a floating point. Next, they wired a transitor (along with the surge protecting diode) to the relay coil (with the wire to the transistor running to an output pin of the Atmega). They then place a one on the output pin of the Atmega and verify the output wires have no voltage running through them, along with conversely placing a Zero and verifying a 120 V (AC) output. Once this was verified, they then implemented this hookup on the circuit board they designed and verified that they were able to control the switching of the light with the Atmega.

The next step in the integration of this circuit is that of the implementation of the switching circuit detailed in the corresponding design section. Since it was a simple circuit setup, the group will forgo an explanation, and skip to the verification. Once wired, they will verify using code that when the switch was pressed, the corresponding input to the Atmega changes state. Once they can verify this state change, they were able to add code that effectively changes the state of the relay to match the state change of the output of the switching circuit.

Finally, the group combined the code discussed in the design portion of the outlet monitor and controller, along with the code to equate a state change of the switch with the state change of the light. They then verified that they had effectively granted control via a “digital switch” and an analog switch. Once verified, the group can assume that once they have the device connected to the central hub (via Zigbee and ultimately the web) that a user will be able to change the state of the light via his or her mobile device (along with a simple button press) and have thus successfully integrated the light controlling circuit!

5.3 Door Lock Testing

The performance testing procedure for the electronic door lock will be subdivided into three different levels. The lowest level of testing will be on simply the electric locking device, the second level will test the metal–oxide–semiconductor field-effect transistor (MOSFET), and the final level of performance testing will test the ATMega328 in conjunction with the other components. After the performance testing, the device will be analyzed for various failure modes.

The door lock that has been selected for this project is the Electric Door Strike - Mortise Type. This lock has been rated for 5 volts to be applied to the terminals, so that is exactly what the first test will be. The 5 volts can be obtained easily in the development environment by connecting to the 5 volt socket in the power header. The motor and board must be grounded to the same ground while doing this. Once the latching and unlatching capabilities and methods have been fully discovered and understood, the load current will be observed. Hopefully, this current should be around 1.5 Amperes, however the metal–oxide–semiconductor field-effect transistor being used to drive this device can handle up to 20 Amperes of current. If there is anywhere near this much current, however, that will raise serious concerns.

The metal–oxide–semiconductor field-effect transistor (MOSFET) is the next component to be tested. This was done by powering the board, thereby powering the MOSFET, connecting the door strike, and connecting a 5 volt logic signal to the gate. This should create a relatively short circuit across the source and drain terminals, which effectively provides a path for the electricity to travel through the electric door strike with. One way to provide 5 volts to the transistor gate is to remove the ATMega328, and connect a wire to the socket which runs to the MOSFET gate with the 5 volt socket in the power header. This will test both the MOSFET and the door strike.

Finally, the logic control handled by the ATMega328 was tested. This will be done by programming one of the ATMega328’s to toggling the first pin of port C (PC0) every few seconds. Next, the ATMega328 must be plugged back into the circuit board for the door lock controller. Then, the functionality of the MOSFET with the Electric Door Strike - Mortise Type in response to the digital level changes may be observed.

The final stage in testing was the power failure testing. This procedure is intended to find any destructive or interesting failure modes during a power failure. To conduct this test, the same software should be run that was toggling the first pin of port C (PC0) every few seconds. At different time intervals, the power was immediately disconnected, and the results were observed. If this caused the door latch to become unlocked, it was to be of serious concern. Since none of these issues cropped up, we were able to infer that the door lock was able to be used as a secured device!

5.4 Communication Testing

One of the most significant aspects about the project under development was the communications mechanisms between each of the various types of nodes. There was Ethernet Internet Protocol (IP) communication between the central hub and the modem or router, XBee Radio Frequency (RF) transmissions in the 2.45 gigahertz (GHz) frequency band between various local network nodes, and communication between the different high level user interfaces, including the website and android application.

The communication testing functionality testing was the most crucial part, of the project, and had very few ways of verifying results. Since the XBee chips took the serial communication lines hostage, there was no way to debug whether the message was being sent, and what it actually looked like. Every device used the XBee chips, so none of the devices could be debugged with a serial port. This prompted meticulous attention to detail when writing protocols.

The first type of communication was the Ethernet Internet Protocol communication between the Central Hub, and the wireless router or modem which provides an access point to the internet. There were initial setup functions to correctly setup the local network configurations. This included proper configurations of the subnet mask layers, and usage of the proper Internet Protocol addresses in order to be recognized by the system. These setup functions were run once in the beginning, and once only. The next major stage for development of the Ethernet communication network included in the final project design was to test the sending and receiving capabilities of the Central Hub through the W5100 Ethernet chip.

In order to verify the sending capabilities of the W5100 Ethernet chip an external computer had to be used. This computer attempted to read data which is uploaded to the network through the Central Hub's Internet Protocol address. To verify that the correct message is retrieved, a unique string was sent from the Central Hub. The string could be read in an internet browser on the testing computer. The webpage was refreshed, and different strings were tested under different circumstances to ensure that the system was working as planned.

Once the sending capabilities were verified, the next step was to test the receiving capabilities of the Central Hub through the Ethernet cable. This required the Central Hub to request packets from a given local server, which hosts another secret, unique number. Once this secret number was read by the W5100 Ethernet chip, an LED would be activated to verify that a message was received.

After the local network testing, the Central Hub was ready to send and receive data to and from the website created for the project. This was tested in a similar method that was used for the sub-netted Local Area Network (LAN). In this stage, however, the information going to and from the Central Hub on the Internet was controlled by the webpage/server which provided data structures for the information being passed to and from the Central Hub. The same transmission took place to and from the Central Hub, however now; the data structures on the webpage server were updated.

The second major communication technology that was developed and tested was the XBee communication protocol. This was verified between two XBee network nodes, as opposed to two Internet Protocol nodes. For this, two Arduino development boards were placed in close proximity initially, and a secret string was sent from one to another. Once this was verified, the node that just received the information sent its own secret string back to the first node. Once this new secret string was verified by the recipient node, short range communication will have been verified.

At this point, the maximum range capabilities of the XBee nodes could be tested. For this, both chips were put in loops sending and receiving an incrementing number 5 times per second. The number was incrementing, so that the software knew if a signal got lost. If the signal got lost, an LED lit up to alert the testers. Information about which number was missed, the time that it occurred, and other relevant parameters, including network parameters, were stored in an array of data structures. The XBee range test included sending signals over 10 meters, sending signals around corners, sending signals through plywood walls, and sending signals through concrete block walls. Once any error occurred in communication, the data structures holding information about the errors were observed. With this information, the range of communication was uncovered, understood, and improved through different fixes.

Once the XBee and Ethernet communications had been executed and verified, it was be time to test the complete development of the Central Hub. To do this, the Ethernet shield and XBee Wireless shield had to be attached to the same Arduino development board. Given the shape of the board, the XBee Wireless shield needed to be stacked on top of the Ethernet shield. The Wireless shield will then need a 5 volt power line and ground line from the Arduino board or the Ethernet board. The source for these lines was discussed in section 3.8.1.

Once the hardware is connected securely, software routines were required to manage the sending and receiving of both the Ethernet and the XBee protocols.

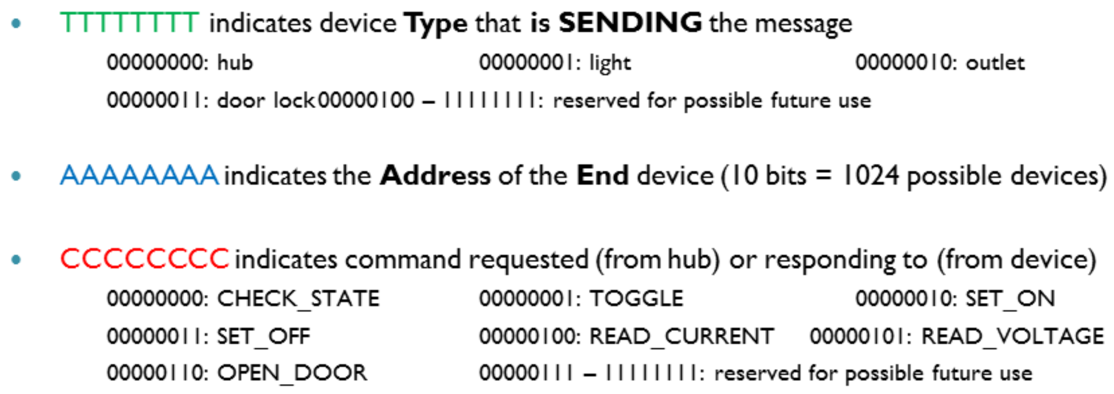
The process of sending and receiving data from multiple devices was prone to collision errors. When two devices would try to send a message at the same exact time to one device, data would be lost and possible cause the system to crash. To handle this error, the software was written in each of the devices such that the devices would only respond to the central hub. The central hub would only send out one request for information at a time, and wait until it received the response before sending out any more requests for information. This reduced the maximum rate of data flow, and also guaranteed no collisions. Since the project required minimal amounts of data flow, it was irrelevant that the maximum rate of data flow was reduced.

The protocol implemented for the Ethernet and XBee communication was as shown in Figure 5-1.

Figure 5-1: Ethernet/XBee communication protocol

<'$'> <TTTTTTTT> <AAAAAAAA> <CCCCCCCC>

Where:



- TTTTTTTT indicates device **Type** that **is SENDING** the message

  00000000: hub 00000001: light 00000010: outlet

  00000011: door lock 00000100 – 11111111: reserved for possible future use

- AAAAAAAA indicates the **Address** of the **End** device (10 bits = 1024 possible devices)

- CCCCCCCC indicates command requested (from hub) or responding to (from device)

  00000000: CHECK_STATE 00000001: TOGGLE 00000010: SET_ON

  00000011: SET_OFF 00000100: READ_CURRENT 00000101: READ_VOLTAGE

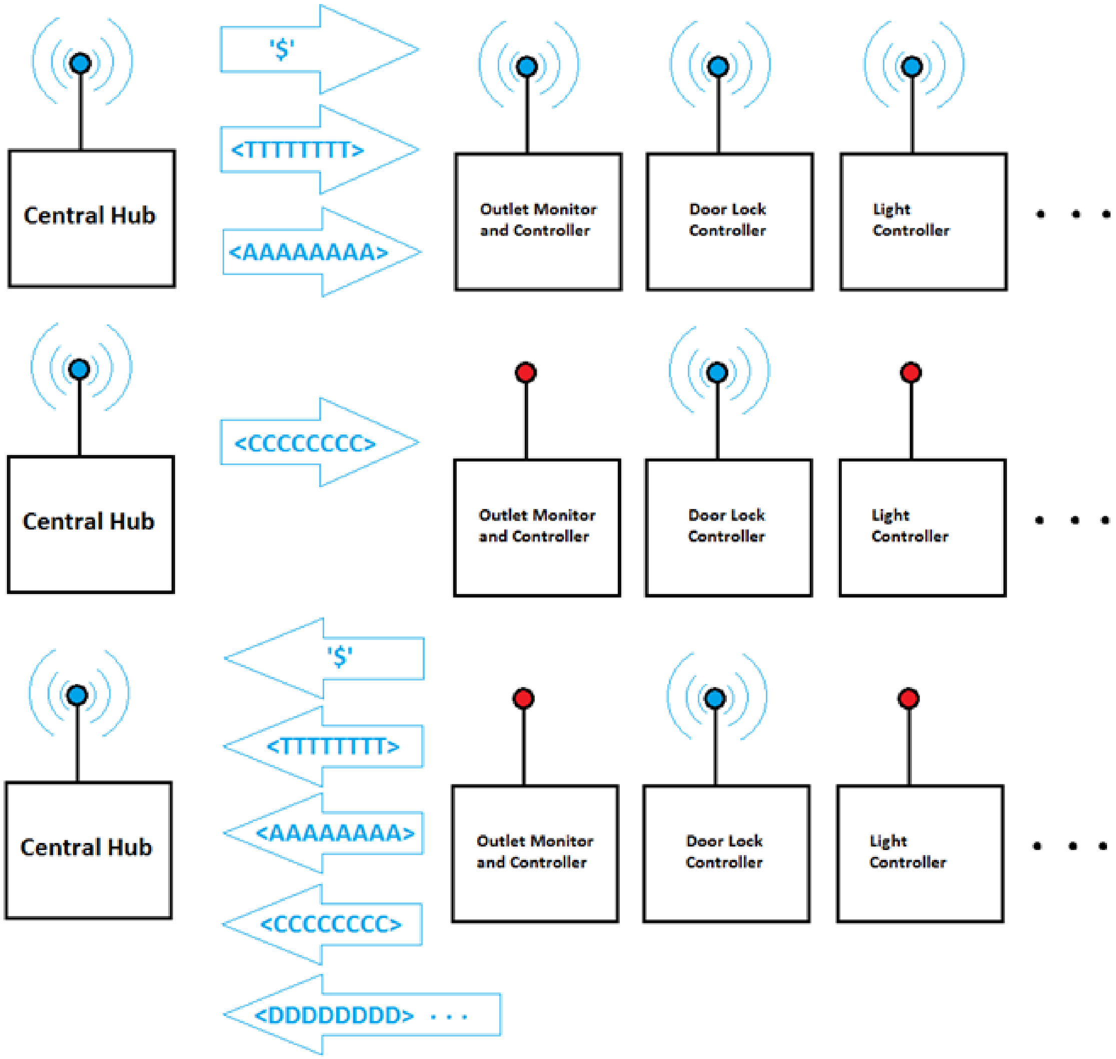
  00000110: OPEN_DOOR 00000111 – 11111111: reserved for possible future use

The website would send a request in this format, and it would be forwarded to the end devices, using a handshake shown as shown in Figure 5-2.

The end devices would respond to every command sent to it. If too much time went by with no response from the end device, a timeout occurred, and an error message was sent up the stack. The Central Hub's sole duties were to relay messages from the website to the end devices, and vice-versa. It performed no higher level logic; the system was controlled at the website level.

In order to verify communication, the IP address of the Central Hub would be typed into the URL search bar, followed by a "/$", followed by a command based on the protocol, and #defined parameters in the device software. The response to this message would be forwarded to the site, and printed on the webpage. This required quite elaborate software to control all of the wait periods, and handle every unique specific situation that could have occurred, but the process was fully debugged in time for the final presentation.

Figure 5-2: Hub/Device handshake process



5.5 Server/Hosting Testing

The Abyss Web Server software test consisted of installing the software on a group member's computer and making sure that it can be accessed from the web. The server connection could be tested as soon as the software was installed and the server was running, because it installs itself to be accessible from port 80 by default, and installs a placeholder page to show that it is working when accessed. The access port can be changed at any point, but port 80 is the port that is accessed by default by any incoming request over the internet. Once this initial test page was tested for accessibility, the group members added their own web page files in order to test that the ASP.NET code was working correctly.

Once they were sure that ASP.NET was able to work on the server as well as they need it to, the Abyss Web Server was considered functional for their project, and any further testing was done in conjunction with testing the web site and image processing software, which was run on the server.

Dyn's DDNS service was very simple to test. They just needed to create a hostname redirection service, and have the redirection point from the provided free URL to the IP address of the computer that the server was being hosted on. When typing the URL into a web browser brought up whatever is on their server, then they knew the service worked. This functionality is simple, and required no further testing.

5.6 SQL Database Testing

The SQL Database was tested in two stages. For the first stage, the group could test the database on its own, and the second stage was testing it from within the website and image processing software.

The first stage of testing was more for testing the table relationships, making sure that they can access the correct data with the given input. Based on Figure [K10] back in section 4.3, every piece of data should have been accessible by specifying only an account id. For devices, rates and digit locations, while they could access all of the ones associated with an account from just an account id, they were able to access a specific record by also passing in a detail for the one they want along with the account id. In order to make sure that this all worked correctly, they manually put data into the tables with insert statements. They used various select queries to test that the correct data was accessed.

Testing the database from within the web site and the image processing software was more focused on testing the connection with the database itself. The group needed to make sure that any connections they made were correct and stable. They tested both insert statements and select queries from within the website and the image processing software. When the table reflected inserts correctly, and the website and software were receiving the proper data back, then the connections were considered correct. When the website and software was able to run while using these connections for an extended period of time, then the connections were considered stable.

5.7 Image Processing Software Testing

Testing the image processing software involved testing of individual sections of the software, as well as the program as a whole once the rest of the software, such as the web site and the IP camera's video feed, was set up. The video feed image capture functionality in particular could not be tested without a video feed URL, but that functionality comes from the AForge.NET existing code, so it should not pose a problem. Similarly, any data saving with SQL will need the SQL database to be available. The user calibration of the system will not be entirely testable until the website is available to provide an interface, although the digit location interface can be simulated using a Windows form test application. This will be easier than trying to manually enter digit locations into the database. The grayscale conversion, resizing and digit reading functionality should be testable on their own, and the digit reading will be the most crucial thing to be tested, since it will require a lot of tweaking to make sure its thresholds and algorithms are just right.

Testing of the digit location calibration was done using a Windows form with an image object on it. The image had an event handler for when it was clicked on. Clicking a button to set a digit would cause the image-click event to record a first click as a rectangle’s start position on the image, in pixels, so this click should be at the top left of the digit area. The second click sets the bottom right of the digit area, and is used to set the rectangle’s width and height values. The location, width and height of the rectangle could then be used to crop out a section of the base image, preferably containing a digit, and then the image processing software could attempt to read that digit. The group used several different digital digit images found on the internet, since most digital digits are about the same, so the example test images did not necessarily need to be a power meter.

The digit reading algorithms were very straightforward for testing. As mentioned, any digital digit images were fed into the test form, and the digit reading results were returned, showing how well the algorithm was doing at reading digits. The form was even set to show the altered Canny Edge Detector result image when it read the digit, so that the group would know exactly what was going on under the hood, so to speak.

Once the website, camera feed and SQL database were all up and running, the components that depend on them could be tested. The website configuration interface software was tested by entering calibration data on the website and checking that it gets stored correctly in the database. The camera feed was tested, just to ensure that it was integrated properly with the software, by having the program grab images from the video feed and checking to make sure that the images were being grabbed correctly. The SQL database code was checked by having the software store some data, such as readings and digit locations, and ensuring that it made it to the database correctly.

Once everything was tested on its own, then the system was tested as a whole by setting the IP camera up to record a meter, and checking the website to make sure it prompts the user for calibration. After that point, they checked that the website began to show real-time updates from the meter based on the data that the software was providing, which meant that the software was working. When all of these functions were running as they should be, then that meant that the software worked. This whole-system test was performed several times in the presentation location to ensure that they would work properly for the presentation.

6 Administrative Content

6.1 Budget and Financing

There is no sponsor for this project, and so all of the expenses are coming out of pocket for the group members. However, while they want to minimize cost, they still need to make the project work, so the cost will still be considerable. They initially estimated that a $100 contribution from each of them, for a total of $400 available, would be enough to complete the project. At the very beginning, before any real research other than average prices for the parts they assumed they would need, they came up with an estimated cost of $270 for everything. This estimate included a camera, a USB cable that they will no longer be using, a single microprocessor, three PCBs, three wireless adapters and miscellaneous electronics parts. At this point in the project, at the end of Senior Design 1, they have found that they forgot to take development boards into consideration, and the estimated cost will almost certainly be exceeded by the end of the project. They thought that despite this oversight, they should have been able to remain on budget.

Along with the oversight of the high price of the group's development environment, the group did not realize how much electrical items and enclosures cost via MSRP (versus wholesale). These two oversights caused the group to spend roughly 100$ more than they had budgeted (final budget shown in table 6-1). The group feels that this "over-spending" will prove to be valuable engineering experience because it taught them to take into account development costs in order to give and accurate picture of how much a specific project will cost!

Table 6-1: Final Cost vs Budget

Part Name	Cost/Unit	Total Units	Total Cost
Arduino Duemilanove	$19.50	3	$58.50
Ethernet Shield W/ POE	$59.95	1	$59.95
Wireless Shield	$14.95	3	$44.85
Xbee Antennae	$22.95	3	$68.85
IP Camera	$49.99	1	$49.99
Door Strike	$29.46	1	$29.46
Current Sensor	$14.08	1	$14.08

Relays	$2.82	3	$8.46
Switch	$2.99	1	$2.99
Wire	$3.99	50’	$3.99
MOSFET	$1.99	1	$1.99
Paper	$21.57	1	$21.57
PCB Layout	*$30*	*3*	*$85*
Components	*$20*	*X*	*$20*
Enclosures	*$8*	*3*	*$24*
Total Expenditures (Projected):			**$493.68**

6.2 Milestones

When the group finally got started on their project, the group members decided to divvy up the sections of the project. While they would of course still communicate extensively to make sure that each of the parts would work together in the end, this approach allowed for each of them to focus on their own parts, rather than have multiple people work on the same thing and potentially wasting time with redundancy. For the Senior Design 1 final paper, they initially planned out their schedule so that they should each write 5 pages per week, finishing in 6 weeks and giving them two more weeks to compile everything and possible turn their paper in early. For Senior Design 2, they planned their schedule out by the week. This initial schedule is presented in Table 6-3 (initial dates accurate, while description was modified).

Table 6-3: Initial Weekly vs Final Schedule, continued on next page

Week	Week- Ending Date	Tasks
1	**6/10/2012**	Research; Begin paper
2	**6/17/2012**	Research; 5 Pgs. written each
3	**6/24/2012**	Research; 10 Pgs. written each
4	**7/1/2012**	15 Pgs. written each
5	**7/8/2012**	20 Pgs. written each; Part ordering
6	**7/15/2012**	25 Pgs. written each; Part acquisition
7	**7/22/2012**	30 Pgs. written each; Begin development board use

8	**7/29/2012**	Development board use
9	**8/5/2012**	**SD1 Paper Due (8/3); Begin PCB design; Software Design**
10	**8/12/2012**	PCB design; Software design
11	**8/19/2012**	~~Order PCB's; Software design~~ Cheaper methods researched
12	**8/26/2012**	~~Receive PCB's; Begin Assembly~~ Decided to make board
13	**9/2/2012**	~~Assembly~~ Researched Parts need
14	**9/9/2012**	~~Assembly/Troubleshooting~~ Ordered parts
15	**9/16/2012**	~~Assembly/Troubleshooting~~ Received Parts
16	**9/23/2012**	~~Assembly/Troubleshooting~~ Started build
17	**9/30/2012**	~~Troubleshooting;~~ Continued build
18	**10/7/2012**	~~Finish Up Design~~ Finished design of devices
19	**10/14/2012**	~~Finish Up Design~~ Designed Central hub
20	**10/21/2012**	Finish Up Design
21	**10/28/2012**	~~Finish Project~~ Integration to web
22	**11/4/2012**	~~Prepare Presentation~~ Website design
23	**11/11/2012**	~~Prepare Presentation~~ Verify project works, update CDR/Practice
24	**11/19/2012**	**Final Presentation**
~~**25**~~	~~**11/25/2012**~~	~~Practice!~~
~~**26**~~	~~**11/30/2012**~~	~~**Show project/Presentation**~~

As you can see, the group was roughly two weeks behind the planned schedule but caught up in mid-October. The original schedule was optimistic and included time at the end of the schedule for if a problem developed. Another problem that came up for the group was that of the presentation being roughly 10 days earlier than originally planned and thus lost 10 days of practice. Despite these setbacks, the group still had a few days to practice in the room they presented the presentation to Dr. Richie, Dr. Llewellyn, Dr. Bauer, and Dr. Lobo.

7 Conclusion

With Senior Design 1 at an end, the group members believe that this past semester has ultimately been a success. They had a few bumps, as any group can expect to, but they finished the necessary prep work and are ready to take on Senior Design 2, where they will put all of this research and design to use in building their final project. They got to know each other better, the way each of them goes about doing a project of this size, and they are ready to adapt to what they have learned and avoid any more bumps.

The hardware and software portions of the project are very different beasts, and each of these parts was approached in a different way. The hardware required a lot of time consuming research, including a huge chunk of time spent learning how to use the PCB design software. While there certainly are multiple ways to do something, the hardware portion of the project is very rigid beyond the choice of which parts to use. Despite this constraint, the group was able to figure out what they need to do and how they need to do it, and the fruits of their labor is evident in this paper.

The software part of the project is much more open to choices. While there was still plenty of research, the group members working on the software portion are already familiar with software coding, so more focus could be given to exploring options rather than figuring out how to even begin doing what they need to do. Even after choosing which tools to use, such as hosting options and programming languages, there was still plenty of freedom to choose how to write the software and design the website.

The experience has been enriching so far, and all of the group members are learning new things from both their individual research and from each other’s research. With this semester under their belts, the group is looking forward to completing their project and graduating from their degree programs.

8 Appendices

8.1 Part Lists

8.1.1 Central Hub Part List

Exported from CentralHub2.sch at 7/28/2012 6:20:00 PM
EAGLE Version 6.2.0 Copyright (c) 1988-2012 CadSoft

Part	Value	Device	Package	Library	Sheet
100M	Green	LEDCHIPLED_0805	CHIPLED_0805	led	1
ASSOCIATE	LED3MM	LED3MM	led	1	
C1	100n	C-EU	C0805RND	rcl	1
C2	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C3	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C4	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C5	47u	CPOL-EUD	PANASONIC_D	rcl	1
C6	22p	C-EUC0603	C0603-ROUND	SmartPrj	1
C7	22p	C-EUC0603	C0603-ROUND	SmartPrj	1
C8	47u	CPOL-EUD	PANASONIC_D	rcl	1
C9	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C11	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C12	100n	C-EUC0603	C0603-ROUND	SmartPrj	1

C13	10u	CPOL-EUSMCB	SMC_B	rcl	1
C14	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C15	100n	C-EUC0603	C0603-ROUND	SmartPrj	1
C16	1u	C-EUC0603	C0603-ROUND	SmartPrj	1
C20	100n	C-EU	C0805RND	rcl	1
C21	22p	C-EU	C0805RND	rcl	1
C22	22p	C-EU	C0805RND	rcl	1
C23	100n	C-EU	C0805RND	rcl	1
C24	100u	CPOL-EUD	PANASONIC_D	rcl	1
C25	100u	CPOL-EUD	PANASONIC_D	rcl	1
C26	100n	C-EU	C0805RND	rcl	1
D1	DIODE-SMB	SMB	diode	1	
FULLD	LEDCHIPLED_0805	CHIPLED_0805	led	1	
IC1	CAT811TTBI-CT3	STM812	SOT143	SmartPrj	1
IC3	MC33269D-3	MC33269D-3.3	DPACK	linear	1
IC4	W5100	W5100	SQFP-S-10X10-80	SmartPrj	1
IC5	74LVC1G125DBV	74LVC1G125DBV	SOT23-5	74xx-little-us	1
IC6	74LVC1G125DBV	74LVC1G125DBV	SOT23-5	74xx-little-us	1
IC7	74LVC1G125DBV	74LVC1G125DBV	SOT23-5	74xx-little-us	1
IC8	74LVC1G125D	74LVC1G125D	SC70-5	74xx-little-us	1

	CK	CK			
IC9	74LVC1G125DCK	74LVC1G125DCK	SC70-5	74xx-little-us	1
IC10	74LVC1G125DCK	74LVC1G125DCK	SC70-5	74xx-little-us	1
IC11	74LVC1G14DBV	74LVC1G14DBV	SOT23-5	74xx-little-de	1
IC13	ATMEGA8	AT90S4433P	DIL28-3	atmel	1
IC15	MC33269D-5.0	MC33269D-5.0	DPACK	linear	1
IC16	LM358D	LM358D	SO08	linear	1
INT	SJ	SJ	jumper	1	
JP1	PINHD-1X10	1X10	pinhead	1	
JP2	PINHD-1X3	1X03	pinhead	1	
JP3	PINHD-1X3	1X03	pinhead	1	
JP9	PINHD-1X10	1X10	pinhead	1	
L	Green	LEDCHIPLED_0805	CHIPLED_0805	led	1
L1	BLM21	WE-CBF_0805	805	wuerth-elektronik	1
L2	BLM21	WE-CBF_0805	805	wuerth-elektronik	1
L3	BLM21	WE-CBF_0805	805	wuerth-elektronik	1
L4	LED0805ROUND	0805RND	led	1	
LINK	Yellow	LEDCHIPLED_0805	CHIPLED_0805	led	1
ON	Green	LEDCHIPLED_0805	CHIPLED_0805	led	1
PWR	LED0805ROUND	0805RND	led	1	

Q1	25MHz	XTAL/S	QS	special	1
Q2	16MHz	CRYTALHC49S	HC49/S	crystal	1
R1	10k	R-EU_0603-RND	R0603-ROUND	SmartPrj	1
R2	300R	1%	R-EU_0603-RND	R0603-ROUND	Smart Prj
R3	12k	1%	R-EU_0603-RND	R0603-ROUND	Smart Prj
R4	1M	R-EU_0603-RND	R0603-ROUND	SmartPrj	1
R5	10k	R-EU_0603-RND	R0603-ROUND	SmartPrj	1
R8	10k	R-EU_0603-RND	R0603-ROUND	SmartPrj	1
R9	10k	R-EU_R0603	R0603-ROUND	SmartPrj	1
R10	15k	R-EU_0207/10	0207/10	rcl	1
R11	1k	R-EU_0207/10	0207/10	rcl	1
R12	10k	R-EU_0207/10	0207/10	rcl	1
R13	1k	R-EU_0207/10	0207/10	rcl	1
R15	1k	R-EU_	R0805RND	rcl	1
R16	1k	R-EU_	R0805RND	rcl	1
RN1	1k	4R-NCAY16	CAY16	resistor-dil	1
RN2	49R9	4R-NCAY16	CAY16	resistor-dil	1
RN3	1k	4R-NCAY16	CAY16	resistor-dil	1
RX	Yellow	LEDCHIPLED_0805	CHIPLED_0805	led	1
S2	10-XX	B3F-10XX	switch-omron	1	
T1	BC547B	BC547B	TO92	transistor-npn	1

TX	Yellow	LEDCHIPLED_0805	CHIPLED_0805	led	1
U$1	POE-RJ45	POE-RJ45	POE-RJ45	SmartPrj	1
X1	AK300/2	AK300/2	con-ptr500	1	
XB1	XBEE-PRO	XBEE-PRO	XBEE-PRO	maxstream	1

8.1.2 Lock Controller Part List

Exported from LockController.sch at 7/31/2012 8:49:34 AM
EAGLE Version 6.2.0 Copyright (c) 1988-2012 CadSoft
Assembly variant:

Part	Value	Device	Package	Library	Sheet
ASSOCIATE		LED3MM	LED3MM	led	1
C5	47u	CPOL-EUD	PANASONIC_D	rcl	1
C8	47u	CPOL-EUD	PANASONIC_D	rcl	1
C20	100n	C-EU	C0805RND	rcl	1
C21	22p	C-EU	C0805RND	rcl	1
C22	22p	C-EU	C0805RND	rcl	1
C23	100n	C-EU	C0805RND	rcl	1
C24	100u	CPOL-EUD	PANASONIC_D	rcl	1
C25	100u	CPOL-EUD	PANASONIC_D	rcl	1
C26	100n	C-EU	C0805RND	rcl	1
D1		DIODE-SMB	SMB	diode	1
IC3	MC33269D-3	MC33269D-3.3	DPACK	linear	1
IC13	ATMEGA8	AT90S4433P	DIL28-3	atmel	1
IC15	MC33269D	MC33269D-5.0	DPACK	linear	1

	-5.0				
JP1		PINHD-1X10	1X10	pinhead	1
JP2		PINHD-1X3	1X03	pinhead	1
JP3		PINHD-1X3	1X03	pinhead	1
JP9		PINHD-1X10	1X10	pinhead	1
PWR		LED0805ROUND	0805RND	led	1
Q1	BUZ10	BUZ10	TO220	transistor-fet	1
Q2	16MHz	CRYTALHC49S	HC49/S	crystal	1
R1	10k	R-EU_	R0805RND	rcl	1
R10	15k	R-EU_0207/10	0207/10	rcl	1
R11	1k	R-EU_0207/10	0207/10	rcl	1
R12	10k	R-EU_0207/10	0207/10	rcl	1
R13	1k	R-EU_0207/10	0207/10	rcl	1
R16	1k	R-EU_	R0805RND	rcl	1
S2		10-XX	B3F-10XX	switch-omron	1
T1	BC547B	BC547B	TO92	transistor-npn	1
X1		AK300/2	AK300/2	con-ptr500	1
X2		AK300/2	AK300/2	con-ptr500	1
XB1	XBEE-PRO	XBEE-PRO	XBEE-PRO	maxstream	1

8.2 Permissions

8.2.1 Fruit Ridge Tools

RE: Permissions
Fruit Ridge Tools LLC [sales@fruitridgetools.com]
Sent:Monday, July 30, 2012 7:23 AM

To: Matthew Beck [mbeck@knights.ucf.edu]

Matthew—
It is no problem to use the pictures. It is appreciated that you keep the copyright/watermarking on it. I wasn't sure if you were saying you were in need of one of these as a sample. If that is the case, let me know. I can send one to you .

Regards,
Tim Sears
Fruit Ridge Tools, LLC
Controller

716-484-3985 (tele)
815-642-4489 (fax)
tsears@fruitridgetools.com

From: Matthew Beck [mailto:mbeck@knights.ucf.edu]
Sent: Friday, July 27, 2012 8:52 PM
To: sales@fruitridgetools.com
Subject: Permissions

Hello,

I am interested in using the following part in a Senior Design class at UCF.

http://www.fruitridgetools.com/storefrontprofiles/processfeed.aspx?sfid=136763&i=242998779&mpid=8171&dfid=1

We are currently writing a 120 page paper, and would like to use the following pictures in it:

http://www.fruitridgetools.com/fullView.asp?sf=y&nb=1&id=%27242998779%27&img=http://images.fruitridgetools.com/Images/L61-W-EA-NOBOX-3.JPG

Could we have your permission to do so? We will leave your copyright on it. Is there any other particular way you would like to be credited for these images?

We are designing devices for a smarter homes (as part of our Senior Design project). We intend to grant user control of lights (with timers available), allow them to monitor power usage of an outlet (while turning the outlet on or off), unlock the front door, and monitor their whole house power bill; all from their mobile phone! Any help you can provide in the way of samples would really be appreciated, since we are poor college kids (NOTE: We are about to enter industry, and thus are not likely to forget a manufacturer that helps us out; meaning we are likely to return with much larger budgets!!!).

Thank you for your help!!!!

Matthew Beck
SI Leader - C Programming
Student Academic Resource Center
University of Central Florida

(863) 414-0273
UCF Stands for Opportunity

No virus found in this message.
Checked by AVG - www.avg.com
Version: 2012.0.2197 / Virus Database: 2437/5159 - Release Date: 07/27/12

8.3 Bibliography

Acceptable Voltage Ranges. (n.d.). (Signals Power and Grounding Specialists Inc.) Retrieved July 1, 2012, from http://www.spgs-ground.com/information/acceptable-voltage-ranges

Hall, E. H. (1879). On a New Action of the Magnet on Electric Currents. *American Journal of Mathematics* , 287-292.

Solid State Relays vs. Electromechanical Relays. (n.d.). (Solid State Optronics) Retrieved July 15, 2012, from http://www.ssousa.com/appnote040.asp

HunterLab. Hunter Associates Laboratory, Inc. Web. 27 June 2012. <http://www.hunterlab.com/>.

OpenCV Wiki. OpenCV. Web. 29 June 2012.
<http://opencv.willowgarage.com/wiki/>.

CVIPtools Home. CVIPtools Development Team, 17 June 2011. Web. 29 June 2012. <http://cviptools.ece.siue.edu/>.

AForge.NET :: Computer Vision, Artificial Intelligence, Robotics. Web. 10 July 2012. <http://www.aforgenet.com/>.

The CImg Library - C Template Image Processing Toolkit. David Tschumperlé, Oct. 2004. Web. 29 June 2012. <http://cimg.sourceforge.net/>.

Domain Names | *The World's Largest Domain Name Registrar - GoDaddy.com*. Go Daddy Operating Company, LLC. Web. 1 July.
2012.<http://www.godaddy.com/>.

Yahoo! Web Hosting: Unlimited Web Hosting | *Web Hosting Features* | *Web Design Tools*. Yahoo! Inc. Web. 1 July 2012.
<http://smallbusiness.yahoo.com/webhosting/>.

Web Hosting, Domain Names, ECommerce. Bluehost.com. Web. 1 July 2012. <http://www.bluehost.com/>.

IPage. IPage. Web. 1 July 2012. <http://www.ipage.com/ipage/index.html>.

Dyn - Managed DNS and Email Delivery. Dyn, 15 June 2012. Web. 2 July 2012. <http://dyn.com/>.

No-IP.com. No-IP.com. Web. 2 July 2012. <http://www.no-ip.com/>.