



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.....	4
1.1 Executive Summary.....	4
1.2 Goals, Objectives and Specification Requirements.....	5
1.2.1 Goals.....	5
1.2.2 Objectives.....	5
1.2.3 Specification Requirements.....	6
2 Research	9
2.1 Outlet Subsystem.....	9
2.1.1 Current Monitoring.....	9
2.1.2 Voltage Monitoring.....	15
2.1.3 Circuit Control.....	17
2.2 Website	21
2.2.1 Web Development.....	21
2.3 Phone.....	22
2.3.1 Android App.....	22
2.3.2 SMS.....	23
2.4 Servers/Hosting	23
2.5 Database.....	27
2.6 Image Processing	28
2.6.1 Image Processing Software.....	31
2.6.2 Camera.....	35
2.7 Central Hub Communication	38
2.7.1 Wired Communication	39
2.7.2 Wireless Communication.....	42
2.8 Power Supply Considerations	46
2.9 Development Environment.....	48
2.9.1 PandaBoard	49
2.9.2 BeagleBoard.....	50
2.9.3 Texas Instruments' eZ430 product line.....	50
2.9.4 Arduino	50
2.9.5 XBee.....	51
2.10 Door Strike	52
3 Design	53

3.1 Outlet Subsystem.....	57
3.1.1 Voltage Monitor Circuit Design	57
3.1.2 Current Monitor Circuit Design	60
3.1.3 Outlet Software.....	64
3.1.4 Power Control.....	67
3.2 Light Control.....	69
3.3 Website Design.....	73
3.3.1 User Accounts	75
3.4 Phone Design	76
3.4.1 Android App.....	76
3.4.2 SMS.....	76
3.5 Server/Hosting Design	77
3.6 SQL Database Design	78
3.7 Image Processing	81
3.8 Central Hub.....	90
3.8.1 Hardware Design.....	92
3.8.2 Ethernet Communication	97
3.8.3 Xbee Networking	99
3.9 Electric Door Lock Design.....	100
4 Prototyping	102
4.1 Hardware	103
4.2 PCB Prototyping	106
4.3 SQL Database Prototyping.....	109
4.4 Image Processing Software Prototyping	111

1 Introduction

1.1 Executive Summary

Our 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. Our project is designed to help with all of these issues.

We plan 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 our custom-designed power outlets or of any lights that are wired into our 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 we will discuss the components of our system in detail. The research section will explain what we have found out about the concepts related to each component and what we have found as our available options for implementing these components in our project. We will discuss the pros and cons of the more viable options we find, but final decisions will not be discussed here.

The design sections will consist of our final decisions on what resources we will use, along with descriptions of how we will implement the components in our 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 contain our 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 our years spent in our respective majors, requiring us to fall back on all of the different things we have learned, both electrical- and computer-oriented. Since our team consists of two electrical engineers and two computer engineers, we feel that the scope of our project provides the right balance between the focuses of each of those majors. Our 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. We

believe that this project will be a fine way to put our skills to the test and show off what we can accomplish with what we 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. We want to keep the user aware of their power use at all times. We 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. We 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, we hope that by using our system, a user would be able to actively reduce wasteful power use and reduce their monthly bill.

The second objective for our project is to provide convenient control over parts of a user's home. We plan 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. We 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, we 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, we 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 we don't have the training for it, and we could end up with costly damage to our homes or even potentially lethal injuries. We 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 us just that: a residential power meter. We decided that we want to use this as our 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, we will use a camera to record the meter, and we will create image processing software to "read" the meter. We will design the image processing software ourselves, and have it run from our 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 we can monitor the power use directly. We plan to develop a type of power-strip-style outlet that plugs into an existing outlet. This outlet will

allow us 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, we 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, we 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, we plan to develop a website and a phone app for our 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. We 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 Specification Requirements

1. Camera Requirements
 - 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 Requirements
 - 2.1. Reading Requirements
 - 2.1.1. Must be able to read different styles of meters, digital and analog
 - 2.1.2. Must be able to distinguish between meter reading digits and other extra digits, such as serial numbers, etc
 - 2.1.3. Must be efficient enough to provide virtually real-time readings
 - 2.1.4. Must be able to adjust to sudden changes to camera orientation
 - 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
 - 2.2.3. Must be able to alert the user if it is unable to solve its issues automatically
3. Central Hub Requirements
 - 3.1. Microcontroller Requirements
 - 3.1.1. Must be able to handle programming to control communication with devices and the web server
 - 3.2. Power Source Requirements
 - 3.2.1. Must run on 9V battery
 - 3.2.2. Must be able to regulate voltage
 - 3.3. Communication Requirements
 - 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 Requirements
 - 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 Requirements
 - 5.1. Microcontroller Requirements
 - 5.1.1. Must have bidirectional communication with central hub
 - 5.1.2. Must have at least 2 analog-to-digital converters
 - 5.2. Voltage Monitoring Requirements
 - 5.2.1. Must use voltage divider for accuracy, cannot assume 120V/240V
 - 5.2.2. Must convert analog output to digital
 - 5.3. Current Monitoring Requirements
 - 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 Requirements
 - 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 Requirements
 - 5.5.1. Must run on 9V battery, rather than outlet's own power, to reduce chance of overloading the circuitry
 - 5.5.2. Must be able to regulate voltage
- 6. Door Lock Requirements
 - 6.1. Must follow the same requirements as lights, except for power monitoring which is not applicable.
- 7. Website Requirements
 - 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 our database
 - 7.7. Must be dynamic and be able to update in real time without a refresh
- 8. Phone App Requirements
 - 8.1. Same requirements as website
 - 8.2. Must run on android devices
- 9. SMS Text Messaging Requirements (Optional component)
 - 9.1. Must be free
 - 9.2. Must be able to receive texts back for device control and allowing user to get information on demand via text
- 10. Overall Project Requirements
 - 10.1. Must keep spending as low as possible, because we have no sponsors, so everything we spend is out-of-pocket
 - 10.2. Must protect against power surges

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 risks 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.

Figure 1-1

Direct Current Measurement Pros and Cons

<u>Pros</u>	<u>Cons</u>
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! Overall, this method is a little outdated and thus the group will continue searching for an alternative.

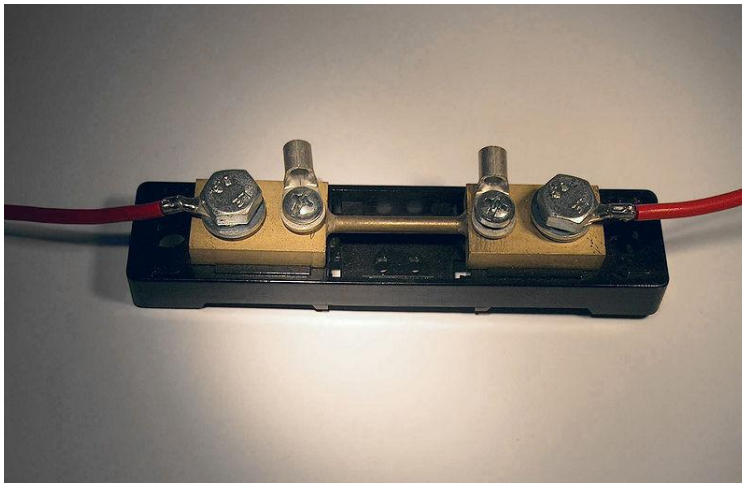
Figure 1-2: Use of a Transformer

<u>Pros</u>	<u>Cons</u>
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

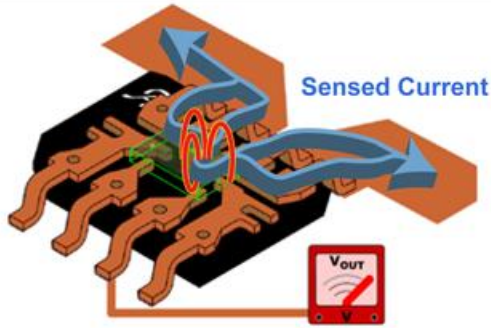
scale of current vs. voltage drop, that can be used (by determining the voltage across the shunt) to tell how much current is running through the circuit. The shunt (shown in figure 1-3) allows for a low voltage measurement across the element, and thus (typically) provides a linear method of calculating 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 1-3: 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 1-4). 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.

Figure 1-4 (Reprinted with legal authorization listed on Allegro Microsystems website)



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 below in Figure 1-5), 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.

Figure 1-5: Advantages and Disadvantages of overall method

<u>Advantages</u>	<u>Disadvantages</u>
<ul style="list-style-type: none"> Fairly basic wiring 	<ul style="list-style-type: none"> Danger of working with HV lines
<ul style="list-style-type: none"> Low Cost (per part) 	<ul style="list-style-type: none"> Temp. of area must be considered
<ul style="list-style-type: none"> Basic Design needed 	<ul style="list-style-type: none"> Potential damage to electric system

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.

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)[1]. 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 1-6). 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 1-7) 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.

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.

Figure 1-6: Honeywell CSLA2CD (taken with legal permission listed on legal portion on Allied Electronics website)

PCB bottom mount

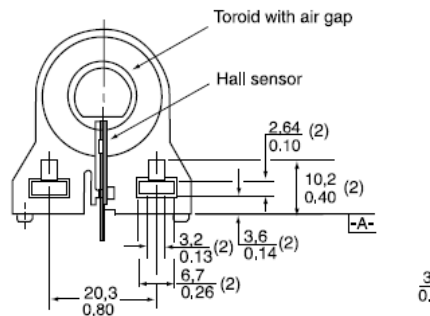
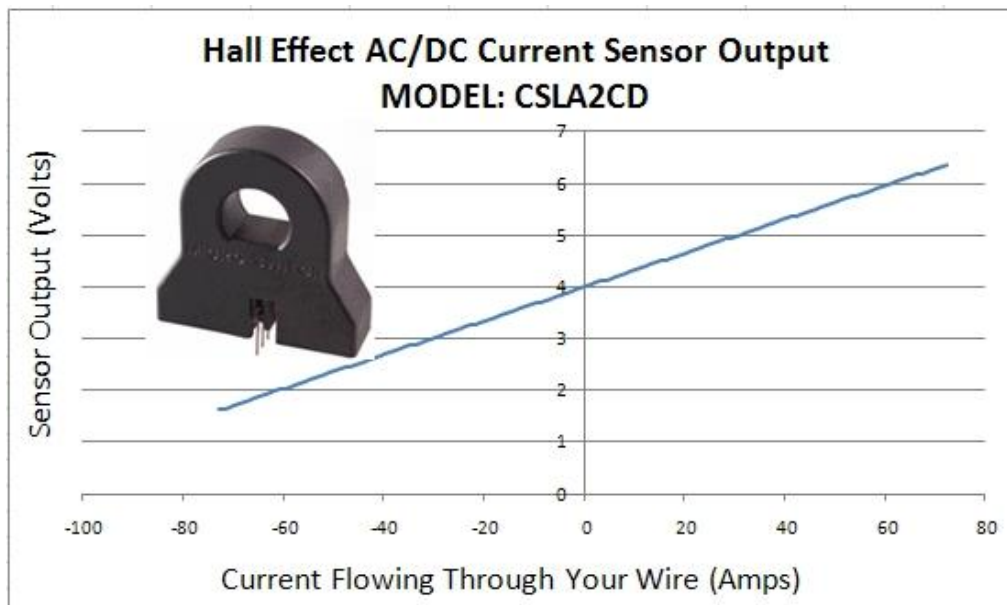


Figure 1-7: Current Sensor Linear Output Example



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.

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 figure 1-8). This range basically says that the voltage at a 120 volt outlet should be somewhere between 110 and 125 volts as a minimum b [2]. 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.

Figure 1-8: 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 Figure 1-9). 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.

Figure 1-9 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 figure 1-10), it is still the most simple and accurate method available to safely measure a high voltage.

Figure 1-10: 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 [1]. Because of the need to use a physical switching element with traditional contact pads, a bounce is destined to occur (shown in Figure 1-1). 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 Figure 1-2), they will consider the other method for circuit control: Solid State Relays.

Figure 1-1: Typical Electromechanical Relay

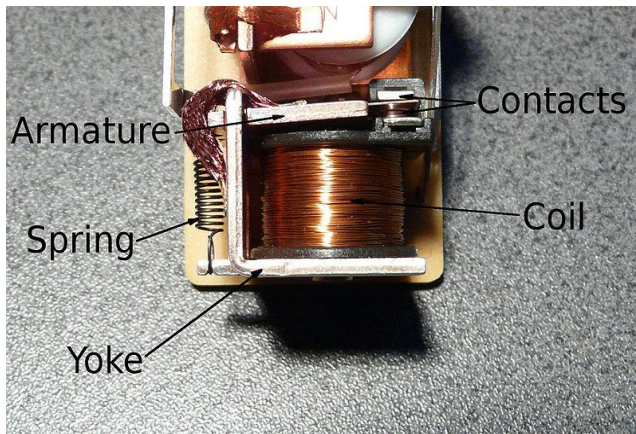


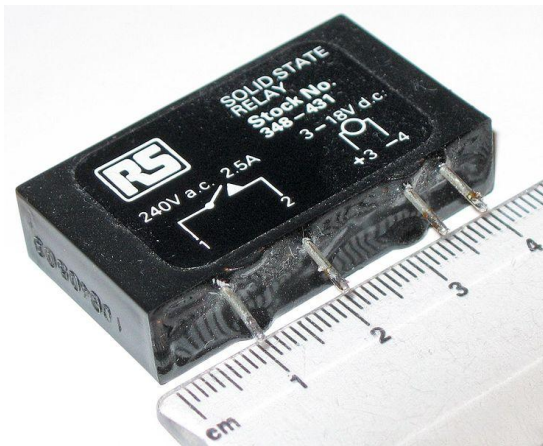
Figure 1-2: Electromechanical Relay Pros and Cons

<u>Pros</u>	<u>Cons</u>
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 1-3) 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 [1]. 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 electromechanical relay.

Figure 1-3: 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 1-4, with the specifications shown in Figure 1-5) 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 1-4:

Pending permission from http://www.onlinecomponents.com/TEConnectivityAMPBrand_9-1393208-5.html?p=10517136

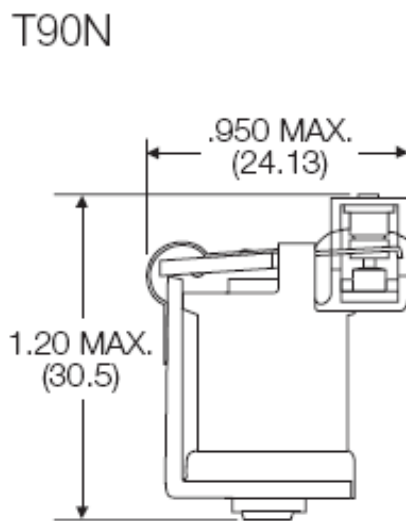


Figure 1-5: Decision based on highlighted Specs

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 over a system which can control electronic devices throughout a house. In order to make control available to a user at any time while also allowing constant updates to the power consumption, there needs to be a website.

The website will allow the user access to the home system network via a phone app. The phone application will update through the website. The phone app controls the website which communicates with the central hub. The central hub acts as a switch allowing the user to turn on or off device which will be wirelessly 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 its best to choose a language that can not only be used for web development, but also phone app programming. The programming language should also make communication between devices 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 speed. Scripts written in PHP 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. PHP is also open source, which can be useful when dealing with complicated algorithms. Php is an extremely versatile language, with a access to various databases. It can also access C libraries and java classes.

Python was also another option. Python is mainly useful for its compatibility with Bluetooth. Bluetooth was of one of our options for wireless communication and would prove to be very useful if that rout is taken. Python is also a versatile language. Python supports object oriented and structured programming. Java seems to be the most popular option for developing the smart phone app. Most android apps are written using the java sdk. It is also the most familiar language amongst the group.

Before developing the website we also need to decide on a server host provider for the website or building our own server. The simplest option would be to use an existing server hosting service. The problem with this option would be the monthly payment. Another problem with using a paid service is also the limitation to programming libraries. Some of the provided servers only allow a website to be programmed using certain libraries. To avoid any extra expenses and possible functionality issues with the site and the server, we will use a personal laptop to host the server. We will simply set up the server hosting software on a laptop and keep the laptop on when updating or accessing the website. The only issue with hosting a server on our own will be keeping the laptop on. The computer must be on in order to access the website or make changes to it. Nothing can work without the laptop being on. Keeping the laptop on will not be an issue since we will only need it on for development, testing, and presenting the project. Server hosting would only be an issue if we were using the site for the general public.

2.3 Phone

2.3.1 Android App

The basis of the project is the ease of use and convenience of home automation. The goal is to make home automation as convenient as possible. This can be done by allowing remote access to devices within the home. There is nothing more convenient than allowing control through a cell phone. This is where the phone app comes into play. The phone app will give a user access to the control through the central hub. We will set up the phone app to communicate with the

website in order to access the central hub. We will create an Application Programming Interface (API) for the website. The API will allow the phone App to pull information from the website. The phone application will not only display data from the website to the user, but will also send information to the website. The app should be able to change the state of the device.

The app will be designed using java. Java eclipse has a android sdk associated with it that allows the development of android app. Java is the most common programming language for phone apps and will present the least problems when developing the application for our specific purpose. The diagram below shows how the phone communicates with the website api and the website will allow updates from the android phone. Bidirectional communication between phone and website.

2.3.2 SMS

SMS or (text-messaging) is an extremely popular form of communication. There are many uses for SMS, but for this project we plan on using text messaging for notifying users within the database of the website. On the website, SMS will be a function that waits for a specified value from the projected bill display. Most likely we will use an if statement which accepts the value from the bill display each time it is uploaded to the page. `IF int bill_value >= int minimum then SMS()`. Any user with an account will be able to set a minimum value of the power bill. The integer value will then trigger the SMS message. All user accounts will have a username, password, phone number, email, and notification value associated with their profile.

There are a few ways to send SMS. Instead of sending an SMS via e-mail, we plan on using GET and POST HTTP methods for PHP. We are already coding the webpage in PHP so this method seems 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. We just need to pass the message details to the chosen gateway. We can send the request using PHP's curl library. Curl is a library which allows communications to different types of servers with different protocols.

2.4 Servers/Hosting

Creating a website is only part of what is needed to create our online user interface. We also need to decide on a server to host the website. Two general options are available for hosting our website: pay for a hosting service, or create our 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 our site, as well as SQL database capabilities. All of the hosting sites we have explored provide more than enough storage and bandwidth for our 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 [K4].

Table [K4] – 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 our own server, our 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 our 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 our 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 our 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 our 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 our 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 we do not need anything more than basic DDNS service for a single IP address, we will only be considering free DDNS services to minimize project costs. A summary comparison of two services to be discussed is here in Table [K5]

Table [K5] – 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 our 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 our project, we will need to store data somehow. We need to store meter readings and meter details, power bill details, and potentially many other things. Because our 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 our group members is very familiar with SQL, using it 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 our project. Previous experience with the software could provide an advantage when developing our 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.

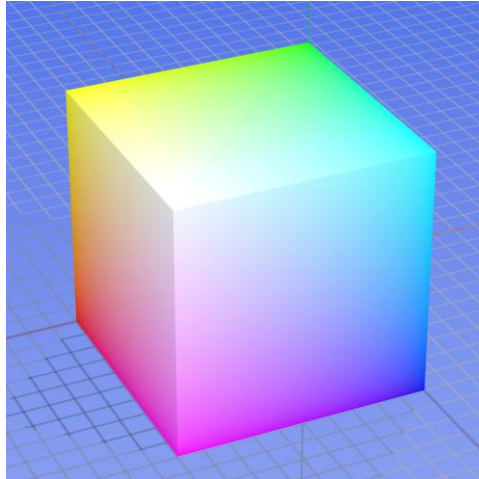
2.6 Image Processing

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.

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.

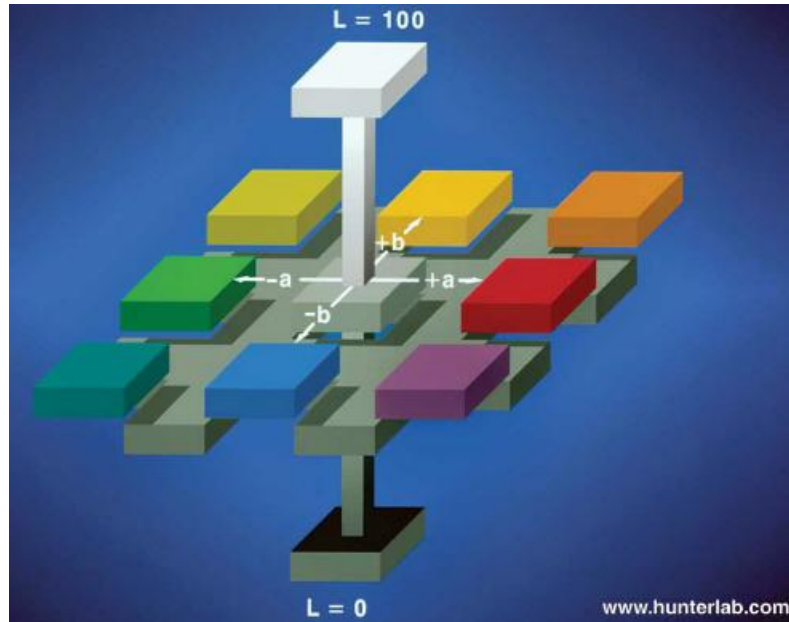
Two color representation models often used in computer vision are Red/Green/Blue (RGB) color space and L*a*b color space. RGB color is a widely known color model, combining various amounts of red, green and blue light together to produce a wide range of colors. In a color image, a pixel may be represented 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 [K1], with the three color channels represented as axes.

Figure [K1] - 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 [K2] below, with the three values represented as axes, similar to the RGB example.

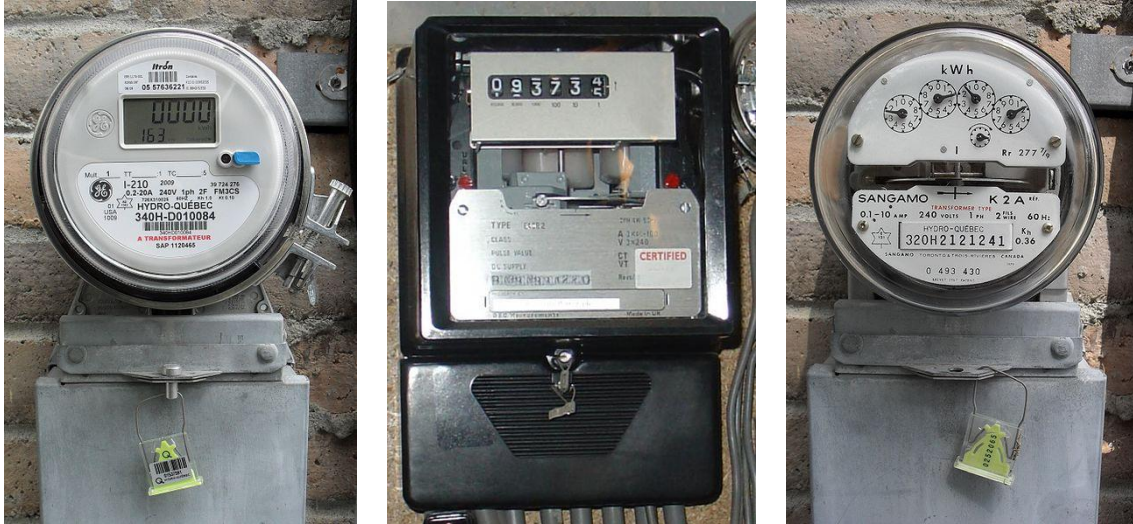
*Figure [K2] – 3-dimensional representation of the L*a*b color model. The L axis represents luminance or brightness and can have values only from 0 and 100, 0 being black (no brightness) and 100 being white (maximum brightness). The a axis represents red/green color and has no numerical limits, with 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*



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 here in Figure [K3]

Figure [K3] - 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)

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 will have to be taken into account when creating the image processing software.

2.6.1 Image Processing Software

For the purposes of our project, the group members need 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.

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. An 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 our 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. A summary of the pros and cons of OpenCV is presented in Table [K1].

Table [K1] – Pros and cons of OpenCV as related to our project

<u>Pros</u>	<u>Cons</u>
Wide range of functionality	Lots of dependency
Includes functions specific enough to be used for our 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	
C++ language can be used flexibly	

Another preexisting computer vision library is CVIPtools, a less robust library than OpenCV but with many of the same features that could be useful to our 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 our purposes, but would rather provide a framework on which to build our own software. A summary of the pros and cons of CVIPtools is presented in Table [K2].

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 our 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 built in function for grabbing a JPEG video stream directly from a given URL. This method would be extremely useful if our power meter video feed is linked to the web rather than directly to our system. This feature is AForge.NET's strongest 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 [K3]

Table [K2] – Pros and cons of CVIPtools as related to our project

<u>Pros</u>	<u>Cons</u>
Fewer files, requires less space	Less robust than OpenCV
Includes functions specific enough to be used for our 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	

Table [K3] – Pros and Cons of AForge.NET as related to our project

<u>Pros</u>	<u>Cons</u>
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 our 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 [K3].

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

Table [K3] – Pros and cons of CImg as related to our project

<u>Pros</u>	<u>Cons</u>
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	

2.6.2 Camera

Our goal is to digitize the output reading of a home power meter. We want to be able to read a value at any given time from a power meter and send that information in a digital form to our central hub. This information will be sent to a user in the form of a text message or via our 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 we need to access the information from the power meter without actually hacking into system. Tampering with a power meter is not ideal. For this situation an internet protocol camera is useful, because IP cameras have the ability to transfer data in the form of an image or video. The transfer of an image and or video can also be done wirelessly using an IP camera. Our intent is to have the Central Hub communicate with the IP camera wirelessly. We will then construct software within the Hub that can interpret the image file sent by the IP camera and output a numeric value to the website.

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 will 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.

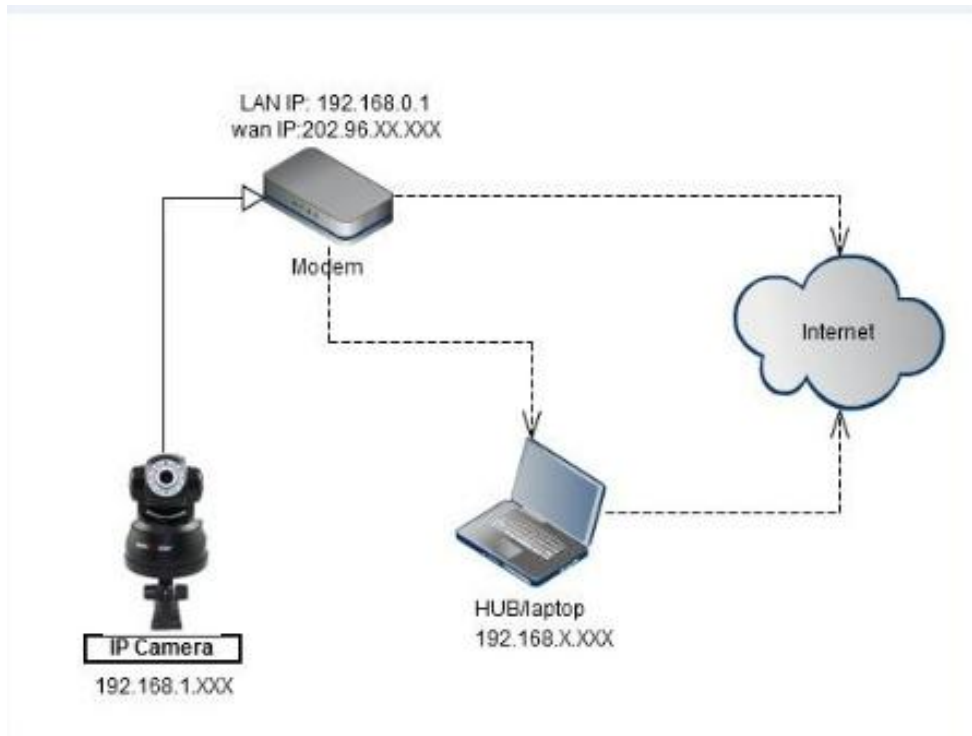
The digital processing takes place within the camera. Another advantage of the IP camera is its ability to be found within a network. A wireless network can simply search for the IP address of the camera to begin communication. Internet protocol cameras also offer FTP and HTTP to upload individual images, which is exactly what we need. We just need the image file so that our image processing program can 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 we have the image file, the camera has done its job. The image processing software will do the rest.

We also need 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 we can 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 such as cell phones.

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 needs 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. Below is a simple diagram of how we can use the IP camera in our network by accessing address of devices. All arrows represent wireless connectivity



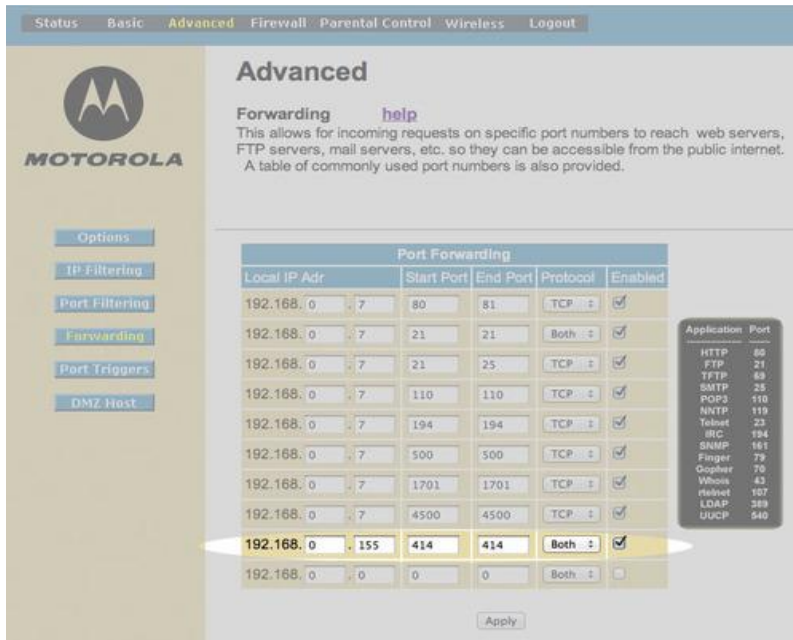
With the advancement of technology in internet protocol camera networks, the price of an IP camera is no longer an issue. Companies like D link and Wansview produce cameras that are not only inexpensive but easy to work with. These wireless cameras have software capable of communicating directly with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

When choosing an IP camera for our specific purposes, we need a few specifications. Primarily we need the camera to be durable and capable of operating night or day. We will choose an IP camera that offers waterproof encasing. This is necessary for testing where we plan to position the camera outside in front of a power meter. Being outside, the camera needs 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 we plan on selecting a camera with night vision capabilities. The concept of the project would require a camera that operates at all times. We need the night vision functionality to insure the camera will take clear pictures regardless of the time of day. The specifications we are looking for also include FTP still picture upload and downloading with 802.11 protocol. This is crucial for the image recognition process. Although IP cameras are great for recording video, we only need to be able to pull single image from the camera. The image recognition just needs to handle a single jpeg image and find the integer value on the image.

There are several companies that produce IP cameras. Not only do we need to weigh our options based on specifications, but we also need to consider budget. These devices can range from fifty to several hundred dollars. The waterproof

capabilities we need will also bring up the cost. The Wansview product line is the most appealing due to the fact that their cameras can fulfill our specification needs. Depending on the model, these cameras can also be found online between the prices of \$50 to \$70. These cameras can offer 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. We can set up the IP camera to accept a request from our web server or our hub by accessing port numbers. Communication can 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 the there are also options for TCP protocol and IP filtering



2.7 Central Hub Communication

As described in the executive summary of this document, there are multiple different nodes that need to communicate to each other in order for the system to function properly. The topology of the network is 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 be used for each different connection. However, having many different types of communication could be cumbersome, so it will be 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 will be considered for this project. Another way to categorize communication is by whether it is wired or wireless. Each have their own advantages and disadvantages which are described in the following section.

2.7.1 Wired Communication

2.7.1.1 Recommended Standard

There are many different protocols that could be 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 is to make a system that is stand alone, so that no other hardware is 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 has been considered. One useful attribute of the Universal Serial Bus is that most digital embedded devices have Universal Serial Bus connectors. This would simplify 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 our selection to, more features and possibilities can be 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 can be observed. Conversely, when the personal computer's connection is in receiving mode, the transmitting functionality of the device under test can be observed.

One of the most appealing aspects about choosing the Universal Serial Bus communication technology, however, is 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 are limited by costs, and not significantly by image quality (as long as a power usage reading can 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) makes it ideal 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 is 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 include Universal Serial Bus hubs, Universal Serial Bus active extension cables, and Universal Serial Bus over Ethernet.

Universal Serial Bus active extension cables are probably 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 can be connected, end to end, allowing for a maximum length of up to 80 feet. The cost for these are 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 will be at least 110\$. This will be an important figure to be 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 to be unattractive wires laying around the house running to the equipment, or they would have to be buried into the walls which would either have 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 would avoid 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

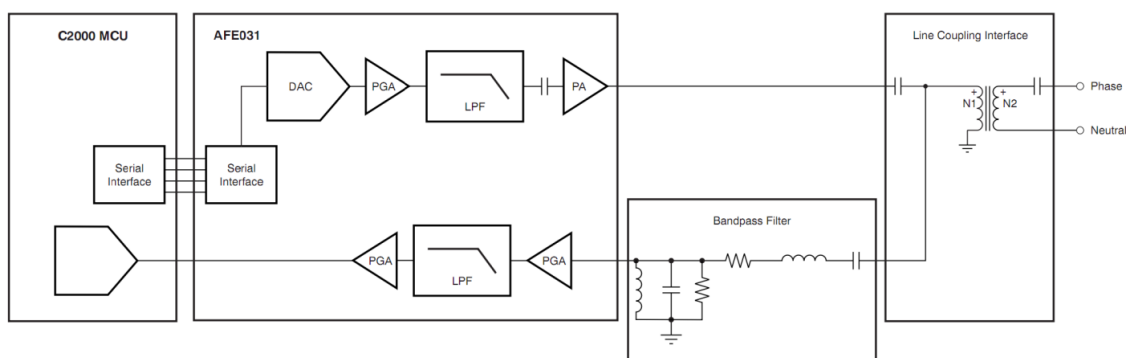
communication technology, including X-10, Universal Powerline Bus (UPB), and Insteon. After much research, the availability of development tools for these existing technologies is extremely limited, due to the proprietary nature of the products; the companies would rather have somebody just buy the whole 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 must be reinvented. In order to transmit a message signal through the power mains, the message signal needs to be superimposed with the existing 60 Hz power signal (50 Hz in other parts of the world). It will be 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 must be filtered out with a low pass filter, or a bandwidth filter must be implemented to extract the message signal from whatever combined signal is propagating through the mains. There are many filter designs that can be considered for implementation in the application; and in fact the signal could be filtered on the digital side or the analog side. The most intuitive filter will likely be, however, on the analog side so that the signal that needs to be decoded will be more manageable.

Both modes of transmission will 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, there will need to be a Digital to Analog Converter (DAC), and for the receiving mode, there will need to be an Analog to Digital Converter (ADC). Also, both mode of transmission will need power amplifiers to amplify the message signal for reading or for sending.

There will also need 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 are still many devices that are needed to complete the design. On the microprocessor side, there needs to be both a Serial Peripheral Interface and an Analog to Digital Converter (ADC). On the power line side, there needs to be a band pass filter, some type of surge protection circuitry, and a line coupling interface. There are also external capacitors and resistors needed to correctly filter the signals or correct impedances, all of which must be chosen with care. The circuit in Figure 1 exemplifies the usage of the AFE031.

Contrary to most communication techniques, the carrier frequency is much lower than the frequency of communication. Also, it would not be wise to disturb the power signal too greatly, because it is hard to know what the effects will be on certain appliances. In this case, the message signal will need 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 put a tremendous strain on development, being that its development would be long and arduous. It would also necessitate that the microcontroller in the central hub and the microcontroller in the outlet monitor and controller must both have an Analog to Digital Converter (ADC) devoted to Power Line Communication, and Serial Peripheral Interface (SPI) module. This introduces serious design limitations. It would not be impossible to do, and will still be considered, but it would require a large portion of the time budget.

2.7.2 Wireless Communication

Wireless communication is an excellent alternative to most wired communications. Similar to Power Line Communication, a wireless physical layer allows 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 are three wireless communication protocols that will be considered for the project. They are 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 has its own collision avoidance and interference mitigating capabilities.

There are many positive aspects about wireless communication such as freedom and device mobility. It is 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 is a high priority if this is the route that is pursued.

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 functionality that Bluetooth has is a 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 is 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 about Bluetooth is that many personal computers have Bluetooth transceivers, allowing 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 can be observed, allowing its functionality to be evaluated. Furthermore, when the personal computer connection transmits a signal, the receiving functionality of the device under test can be observed. Conversely, when the personal computer's connection is in

receiving mode, the transmitting functionality of the device under test can be observed.

One of the biggest disadvantages of Bluetooth technology is its 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 is 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 require 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 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 sacrifices some of its transmission speed in trade 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

would be the ideal node type for implementation in the outlet monitoring and control assemblies.

There are only two disadvantages to using ZigBee as opposed to other wireless protocols. The first disadvantage is the reduced data rate. This, however, will be 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, is the lack of debugging capabilities with personal computers and other currently obtained equipment that is Bluetooth and Wi-Fi ready. ZigBee is a well known protocol, however, so there should be sufficient development resources available online. Like many technologies though, ZigBee does have proprietary characteristics, meaning that it would be ideal to begin development as soon as possible 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 will be 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, we do not want to interfere significantly with these existing networks, and second is that the option now exists to bypass the central hub and send raw data directly to a remote server to be processed. Whether this option is chosen or not, it will be convenient to have in the event of difficulties with the central hub design.

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 be 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, one could expect at least a range of 50 to 65 feet 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 will try again, until the packet goes through, or there is a timeout on the packet. This error detection algorithm guarantees that no corrupted information will 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 were to happen, it could be handled sufficiently at the application layer in a manner which is much better and easier to do than dealing with potentially corrupted information. The User Datagram Protocol (UDP) is not necessary, because there won't be large amounts of information being transferred in short amounts of time.

One aspect about using Wi-Fi devices for communication to and from sensors is that the costs for these parts are relatively high. Besides the fact that they are about twice as expensive as their ZigBee counterparts, they consume a lot more power as well. This makes 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 are many options to consider when designing the source of power for each network node in the project. Each node must have a supply voltage of 5 volts for the ATmega328, and 3.3 volts for the XBee module. Once a 5 volt power source is obtained for a given node, that voltage may simply be 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 will be needed in the final design, and will be selected on a basis of lowest price when it comes time to spin the board.

To obtain a 5 volt power source in the first place, each node has several different options; for instance, the Central Hub has 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 potentially be powered directly by the hot lines running from the outlet. At the same time, however, any network node can be powered either by a power adapter that plugs into a wall, or by a battery.

The first option which is available for the Central Hub is the Power over Ethernet module that will be used in the development environment. This provides a simple design solution, which once it is configured and installed, does not need to be maintained further. However, there are a few disconcerting aspects of

obtaining power through the Ethernet cable. Primarily, this would make the Central Hub dependent on the router's power capabilities. Routers are not traditionally intended to serve as power devices, so this could present issues. Secondly, the Power over Ethernet module used in development is not open sourced hardware, and the reference design used in development is incomplete for the Power over Ethernet module. Obviously, it is still capable of being accomplished, but it would present design problems that must be solved, which would potentially take time away from designing other more interesting features.

Another option to provide power is by tapping directly into the 120V power lines running in the electrical outlets. This is convenient for the outlet monitor and controller, being that it already has these power lines running into it; it could also be replicated easily by any other node by adding a power cable to it. A complicated design process quickly presents 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. Laptop chargers also get hot though, as they dissipate energy, meaning that this might not be the most efficient route to go.

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

The last, but not least, solution considered would be 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 is that they will have to be replaced every so often (probably every couple of months, like a typical smoke detector). If the battery dies, however, the application layer will be aware of it, because it will stop receiving a signal from whichever device ran out of battery charge, and will be able to alert the user. There are many benefits for using a battery. The first benefit is that the design will be significantly simpler than the previously discussed methods. All that is needed is the battery source, and a voltage regulator to output a constant 5 volt signal for the board. This is very similar to the method used in the development environment, and can be implemented in the same manner on each network node. Another advantage is 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 is much lower.

2.9 Development Environment

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

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

The second most important issue for this project with regard to the development environment is the amount of support that is available. If a development board is more obscure, less user friendly, has capabilities locked or not released for development, it could overcomplicate the simplest of problems. If a popular board with a lot of support is to be used, it is much more likely that the design can be much more easily implemented. Not only will the required design be much more likely done on schedule, but new functionalities can be explored with ease. Since a major goal for this project is scalability, this will improve the chances of a more complex design with extra functionalities to be feasible.

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

The fourth most important issue is flexibility. This project needs to be a scalable as possible so that various functionalities can be added or dropped at any time. It would be advantageous for a board to have two analog to digital converters (ADC's) so that both current and voltage could be monitored seamlessly (although an external multiplexor could make one analog to digital converter suffice). Any wireless communication capabilities would be beneficial; whether they be 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 is an additional capability that would allow for calculation of power factor, which would be 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 serves the purpose of providing numerous scalability options. Also, being that it is a popular high-end development board, there appears 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 be capable of processing images, and 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 are excessive for the basic design, they could be useful if a more complex design was implemented with additional features.

The price for the basic PandaBoard is \$174, and the upgraded version, The PandaBoard ES, is \$182; a modest increase in cost for a modest increase in capabilities. The processor on the PandaBoard ES has a slightly faster processor, and can encode/decode 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 does provide a lot of scalability possibilities, many of the basic features are unnecessary. The price is relatively high for this project, but allows 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) can not be bought in quantities less than one thousand, and therefore can not be used in a prototype PCB layout. Also, Texas Instruments does 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 be possible to accomplish with ease.

2.9.2 BeagleBoard

The BeagleBoard is another development environment created by Texas Instruments. It is under consideration for use as the central hub as well. Like the PandaBoard, the BeagleBoard claims to be open source hardware; however the same fine print issues exist like those of the PandaBoard. Hence, the central hub would essentially have 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 does not come with Wi-Fi or Bluetooth; an essential capability for the central hub. At this point, purchasing the PandaBoard instead would be cheaper, less time consuming, and less risky by not having to worry about compatibility issues. The original BeagleBoard is priced at \$125, and the BeagleBone is priced at \$89. However, without the peripherals for the additional scalability, purchasing an upper-mid range board seems pointless when a much cheaper board can be bought that will do 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 reduce interference with any Wi-Fi signals that may be present.

The eZ430-RF2500 appears to be the simplest and most practical solution for the project. The \$49 development kit is easy to set up, comes with a USB programming interface and emulator, providing quick and easy implementation. The kit includes 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 seems comprehensive, there seems 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 may be hard to add functionalities using any products other than ones provided by Texas Instruments.

2.9.4 Arduino

The most obvious advantage for choosing an Arduino development board for this project is 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 researching the countless other projects that have used this board, there are many other advantages as well. Everything on this board is open sourced. Software is 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 are readily available for every model of Arduino that has ever been released. This will allow us to layout a Printed Circuit Board (PCB) for the prototype based confidently on the reference design that will have been rigorously tested at 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. Whichever route is preferred to interface to an XBee module with, there are plenty of choices, which helps keep design simple to accomplish.

There are different classifications of XBee devices to consider. The first classification is whether the device is XBee Series 1 or XBee Series 2. Series 1 devices are basically easier to use. They do not have to be configured before communication to another node. This makes Series 1 devices ideal for simplifying the development process. Series 2 devices are much more advanced products that require setup, and can allow for optimization of things like power consumption. This makes 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 is 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 our home automation we have 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. We will allow a user to lock or unlock the door latch in the house at any given time. For this lock we have decided to use an electric door strike. We 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. We will only need to supply a voltage for a small period of time in order for the lock to work. The lock will communicate wirelessly with an android app or website. We 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.



3 Design

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

There are many different possible arrangements of the components listed above, and many different ways to communicate between all of them. There are different network topologies to consider, each of which having their own advantages and disadvantages. Each network can be arranged in a linear fashion, star configuration, or in a daisy chain. A daisy chain is only 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 requires each node to act as a repeater if necessary, which involves more advanced hardware and software. This approach will be avoided.

The most intuitive approach for the network implementation of this project is 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 will be 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 will be useful when communicating with nodes of the same type, such as multiple outlet monitoring and control units.

The original block diagram can be seen in Figure 1 below. 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 the 0, Universal Serial Bus (USB) communication technology would not be sufficient for this task. Also, these types of processing capabilities would require an advanced processor, and video decoding software libraries that may be proprietary or difficult to obtain. In theory, it is a simple concept, and it is most likely a completely possible design to implement, however if more convenient, intuitive ways to complete the exact same functionality exist, those

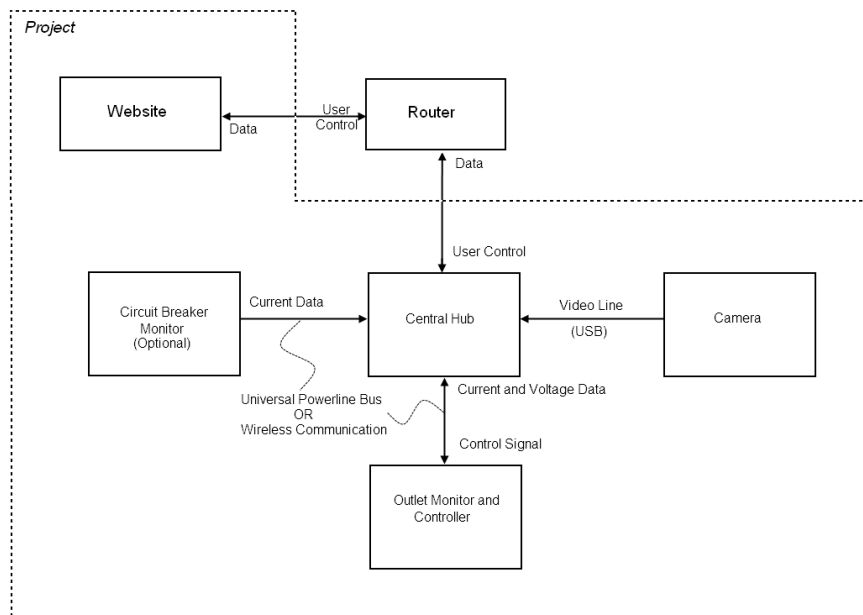


Figure 2: Original Block Diagram

will be explored so that the optional features can be investigated and added to make the design more well rounded and impressive.

While researching about how to solve the image obtaining and processing conundrum encountered in the initial design, Internet Protocol (IP) camera's were soon discovered. These alleviate issues associated with wired communication protocols such as Universal Serial Bus (USB). They provide 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 can be seen as not necessarily removing the central hub; it only moves it to a remote server with the help of the internet. Figure 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 allows 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.

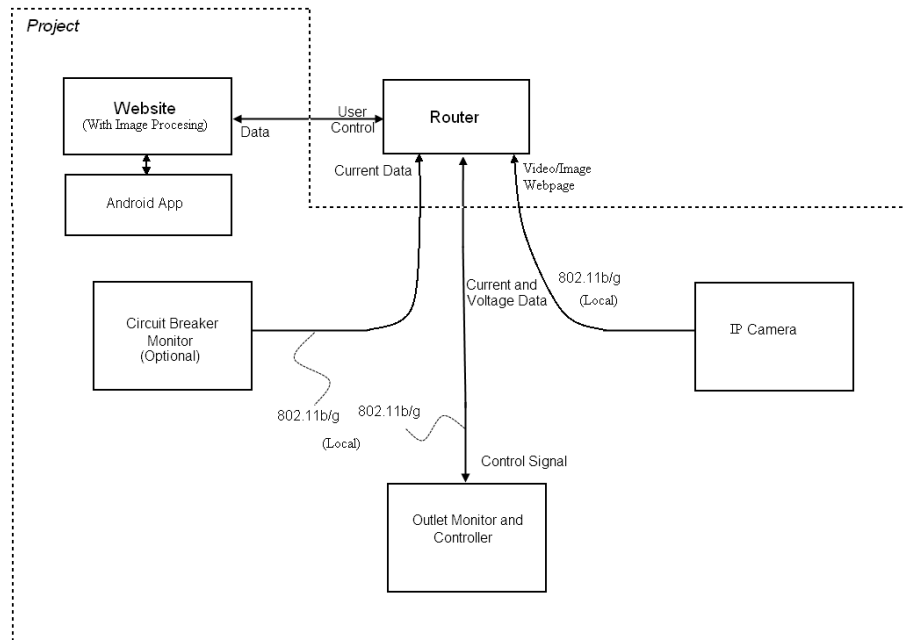


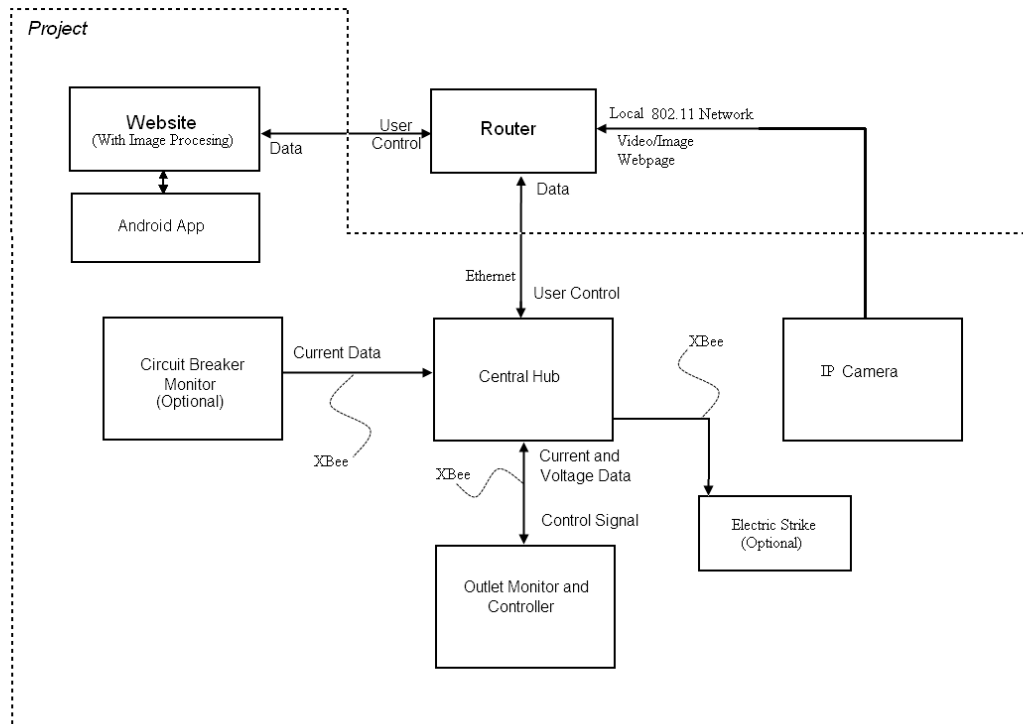
Figure 3: No Central Hub

However, a few major issues remain with this design. One of these issues is that all flexibility vanishes. It is no longer so easy to simply add the remote door locking mechanism, unless that too has an IP address. It also relies 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 be 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 are 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 necessitates 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 makes much more sense to have the IP camera send signals directly to the router, which is designed to forward information wherever it is needed. With this implementation, the image can be processed online, or on the central hub. However, it would still be ideal to process the incoming images on the same remote web server which is hosting the site where the information is accessed and edited for the system so

Figure 4: Final Block Diagram

that additional storage and processing capabilities are not required on the central



hub. This design implementation can be seen below in Figure 3.

Another feature that can be seen in this design is the addition of the Electric Strike, a wirelessly controlled deadbolt assembly. This component will be convenient to access through the internet or a cell phone application in case the physical keys were lost. In order to implement this, however, some kind of encryption scheme should be 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 is 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. It is likely that only one of these will be implemented to avoid excessive expenditures. Both of these components only need information to flow in one direction. The Circuit Breaker Monitor only needs to send current data to the central hub from the site under surveillance, and the Electric Strike only needs 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 will be used to provide a seamless interface with the hardware located at the house in which the system is installed. Essentially, the web site will serve as a link in a daisy-chain topology between the Android Application, and the hardware it is supposed to control. This will 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 Figure 3. In this design the three essential project components are implemented; the image is obtained with an IP camera, the Outlet Monitoring and Control Unit communicates with the Central Hub through an XBee network, and the website is accessed through an existing router, which is extraneous to the project. Additionally, there will be an android application communicating only with the website, and either a Circuit Breaker Current Monitor, or an Electric Strike communicating directly to the Central Hub. Incidentally, with all of these network connections, the three major different types of network topologies will be implemented. A linear network will be 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 can also be 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 can be seen connecting the Android Application to the rest of the system through the website.

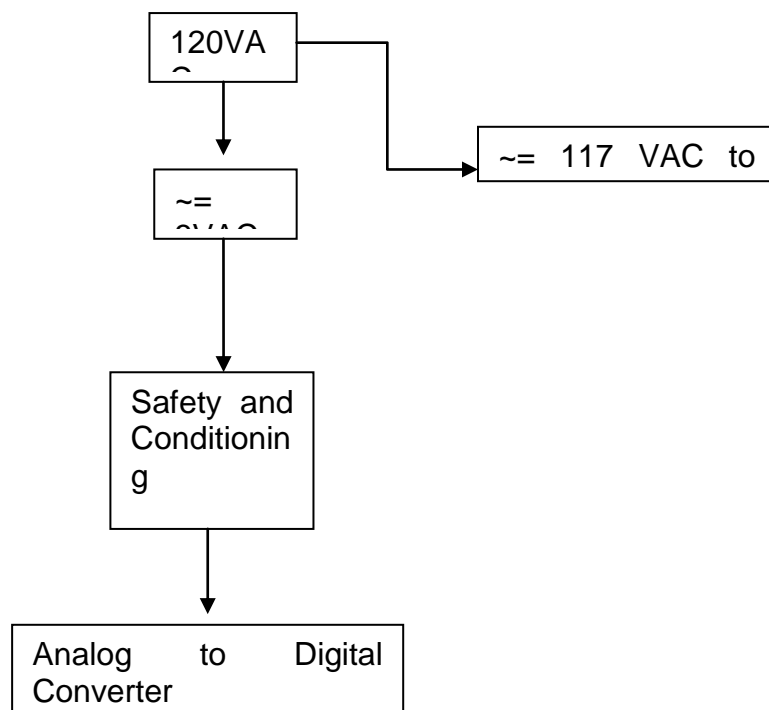
3.1 Outlet Subsystem

3.1.1 Voltage Monitor Circuit Design

Now that the group has decided on directly measuring the voltage seen by the outlet monitoring device 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

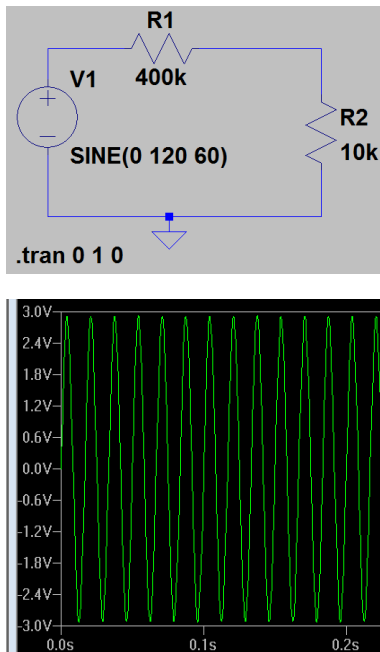
level to be seen by the ATD's (while still allowing the device plugged into the outlet to have as close to a true value of the actual voltage transmitted). Now in the range of the ATD's, the group will want to place a potentiometer before the ATD to allow for calibration of the signal (based on a known 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 3-1), so that they end with a desired output to pass to the ATD.

Figure 3-1: Block Diagram of Signal Conditioning



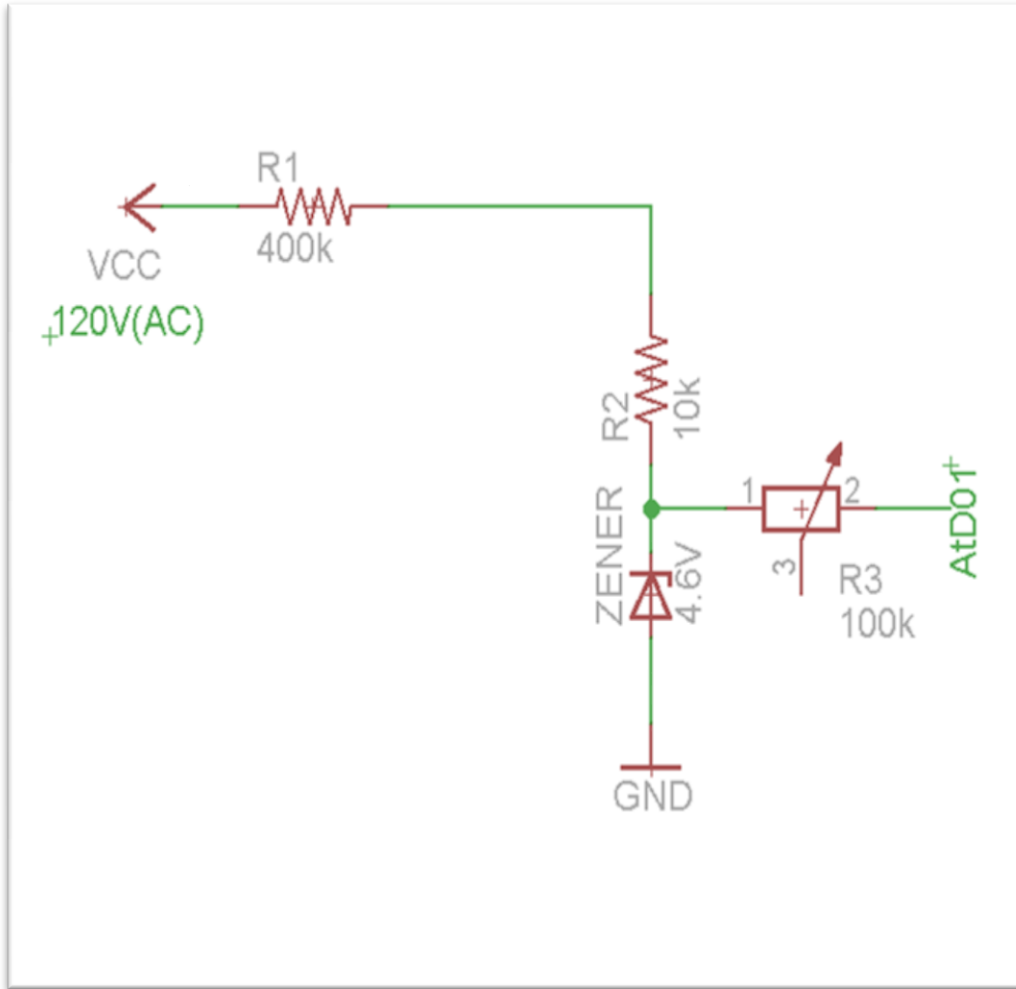
The next step in the group's quest to condition the high voltage signal is to use a simple voltage divider to send roughly 117V to the device, while sending an average of 3V to the input of the ATD. This voltage was chosen based on the input range of the ATD they will be using to perform the measurement. 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 3-2, 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 3-2: 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 breakdown 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 3-3). 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 3-3: Final Voltage Calculation Circuit



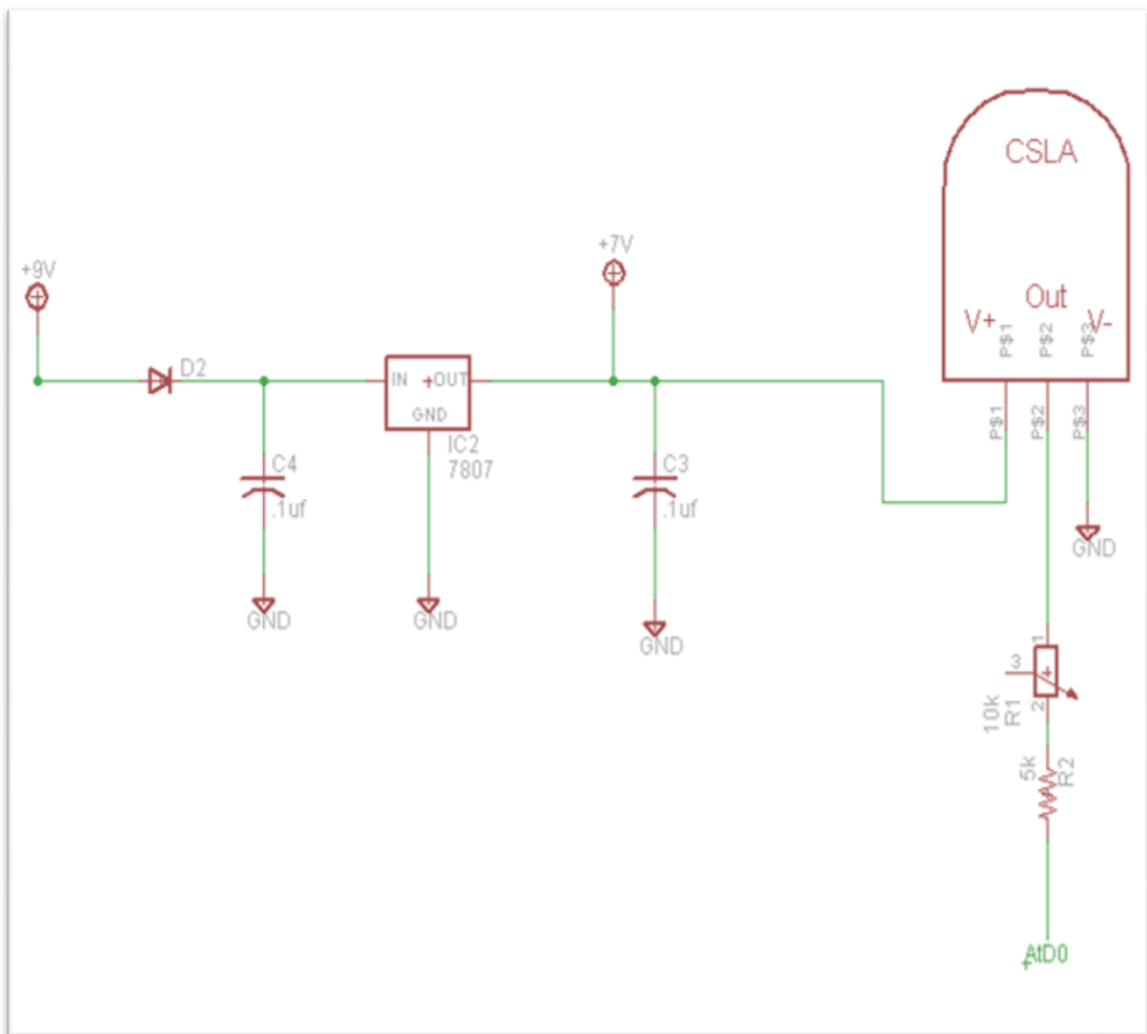
Once the signal has been conditioned for safety, the group is now faced with two options as to how they will be calculating an RMS value of the voltage. One way is by use of an envelope detector. Another way is to use various software techniques to sample the voltage at a certain rate and use an algorithm 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 3-4). 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.

Figure 3-4: Current Sensor Power Supply and Output Schematic



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

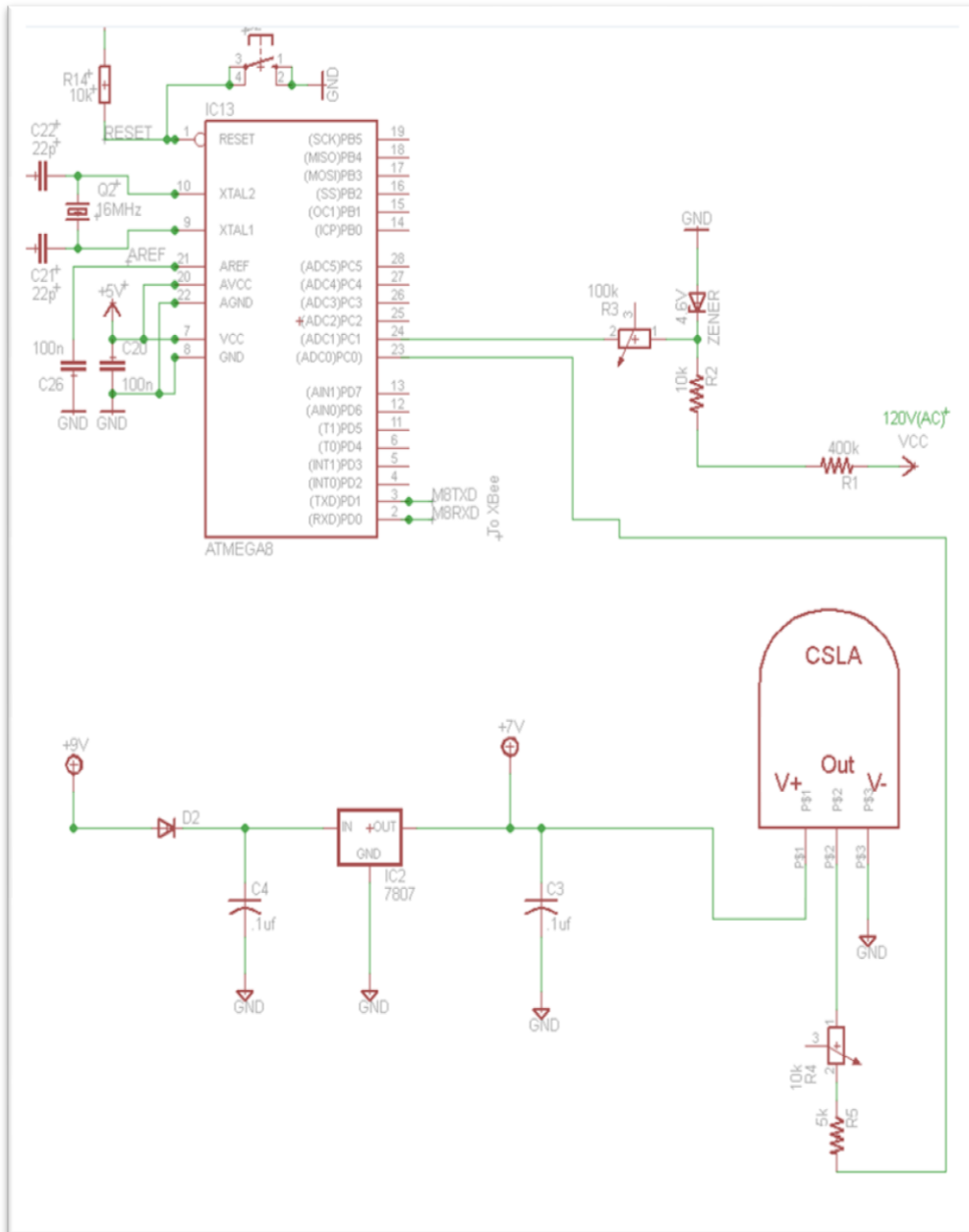
note that the price list for this schematic is around what the group has budgeted for the outlet monitoring circuit (shown in Figure 3-5 with development cost factored in). For the individual schematics, and the total schematic, they have chosen to use Eagle Cad so as to grant them the opportunity to 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 3-6).

Figure 3-5: Cost of Outlet Monitor

<u>Component</u>	<u>Quantity</u>	<u>Cost per unit(\$)</u>	<u>Total(\$)</u>
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 3-6: Completed connection of Outlet monitoring Sensors



3.1.3 Outlet Software

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 the following lines of code:

```
int analogPin = 3;    //specific pin  
  
int val = 0;         // variable to store the value read
```



```

void setup()
{
  Serial.begin(9600);    // setup serial (print screen)
}

void loop()
{
  val = analogRead(analogPin); // read the input pin
  Serial.println(val);        // debug value
}

```

Now that the group has a loop with which they are able to sample the analog input, they can simply just have the loop run 167 times (giving a total sample time of 17 ms). Inside of this loop, they can modify it as follows to tell what the peak RMS value is:

```

int analogPin = 3;    //specific pin
int valNew = 0;      // variable to store the value read
int valOld = 0; //Initialize holder
int counter = 0;
int RMSVoltage= 0;

while(counter<167){
  valNew = analogRead(analogPin); // read the input pin
  if(valOld<valNew){
    valOld = valNew
    counter++;
  }
}

```

```
    If(valOld>valNew){  
        valOld = RMSVoltage; //Last Value must be max  
        counter = 167: //Breaks while loop  
    }  
} //end
```

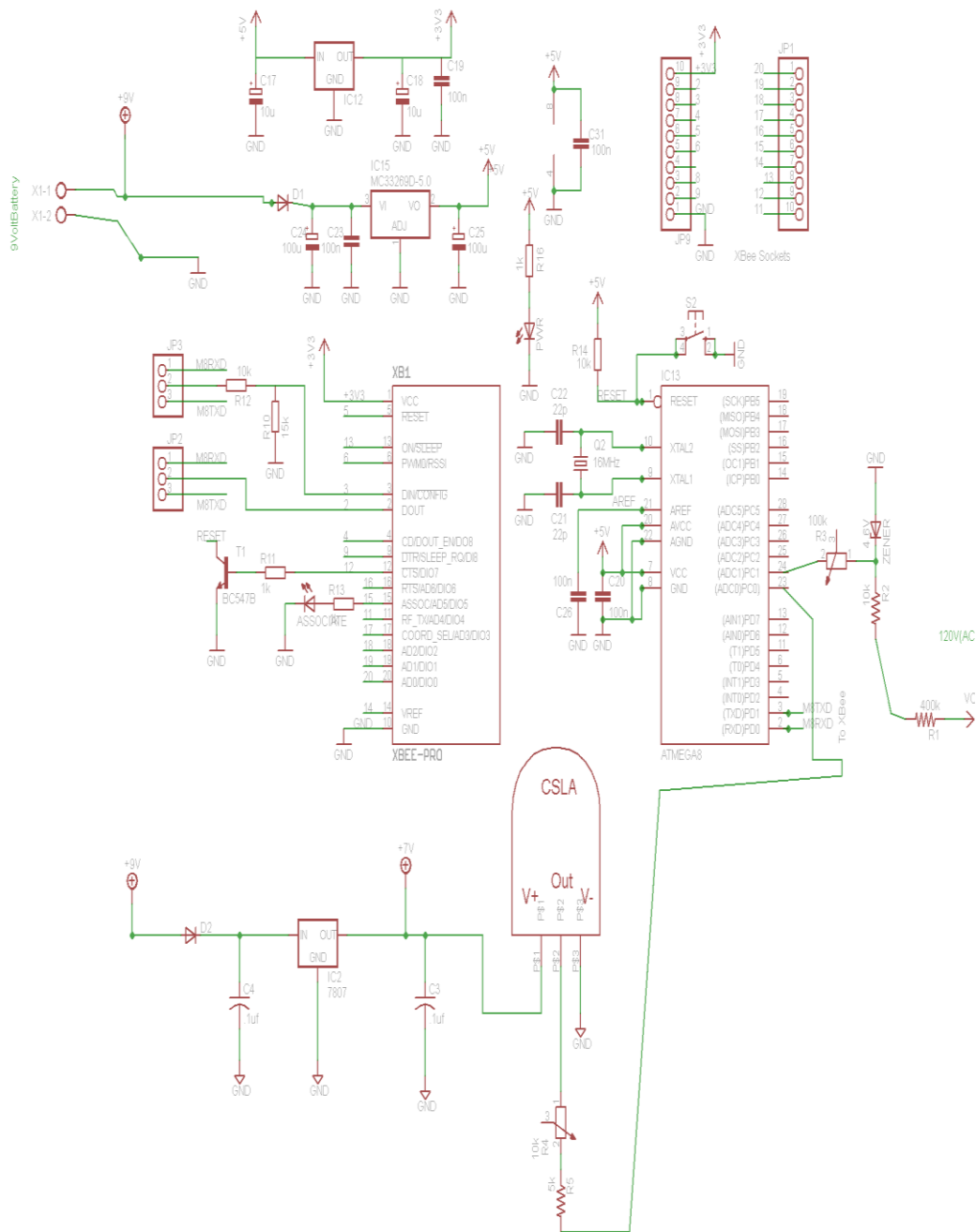
As you can see, the group will be attempting to run a basic bit search in which the software will keep running (up to 167 times) and will be placing the maximum value in the integer RMSVoltage for storage. Now that they have an accurate RMS voltage (between 0-5 volts), they must do some multiplication 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: $P = V \cdot I$. 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

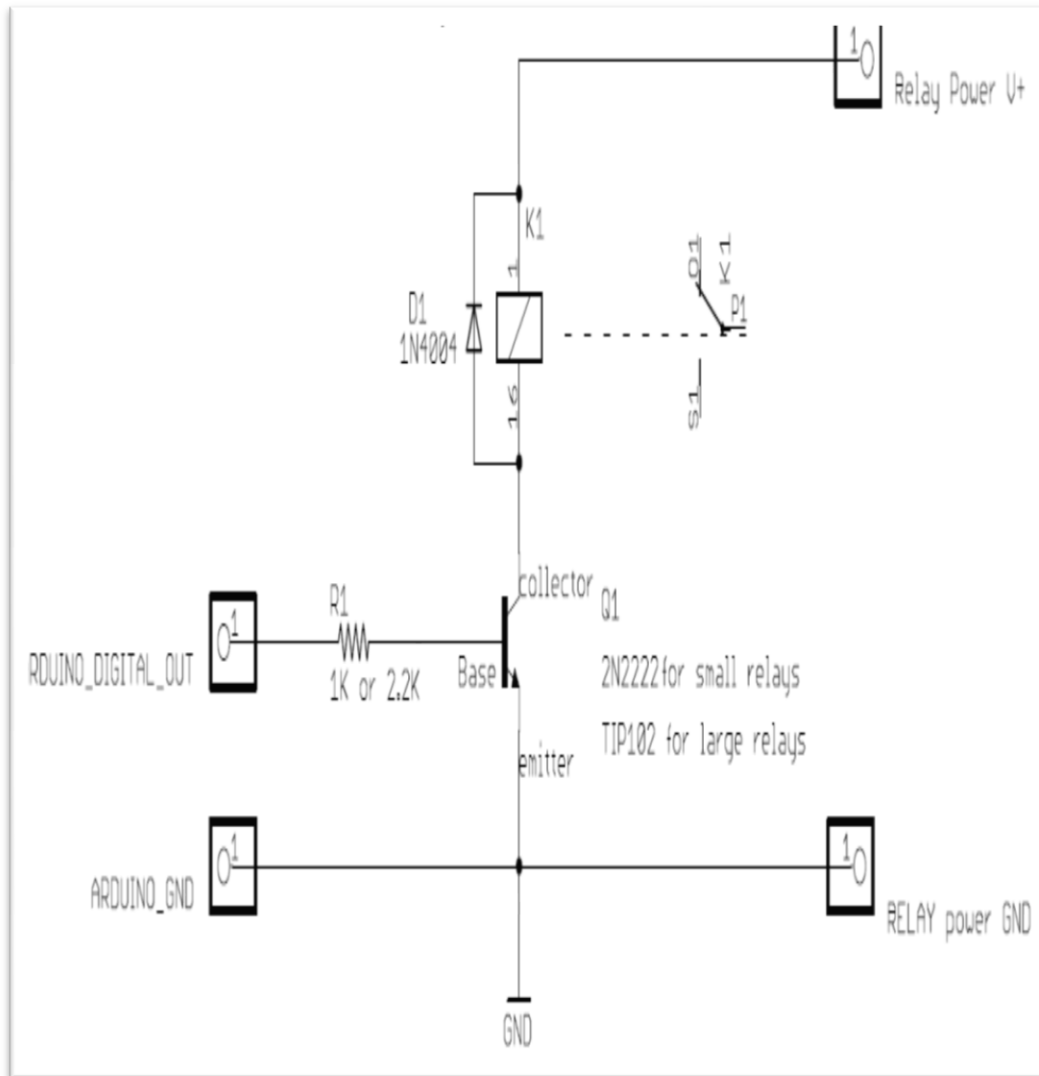
Now that the monitoring circuit has been designed (final schematic including communication shield but without power supply shown in figure 1-0) and the relay has been picked, the group must now research the specifics on how they will be gaining a control over the outlet. For all intensive purposes of 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 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 1-1. 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 1-1: Interfacing Relay to Atmega (taken from <http://www.arduino.cc/playground/uploads/Learning/relays.pdf>)



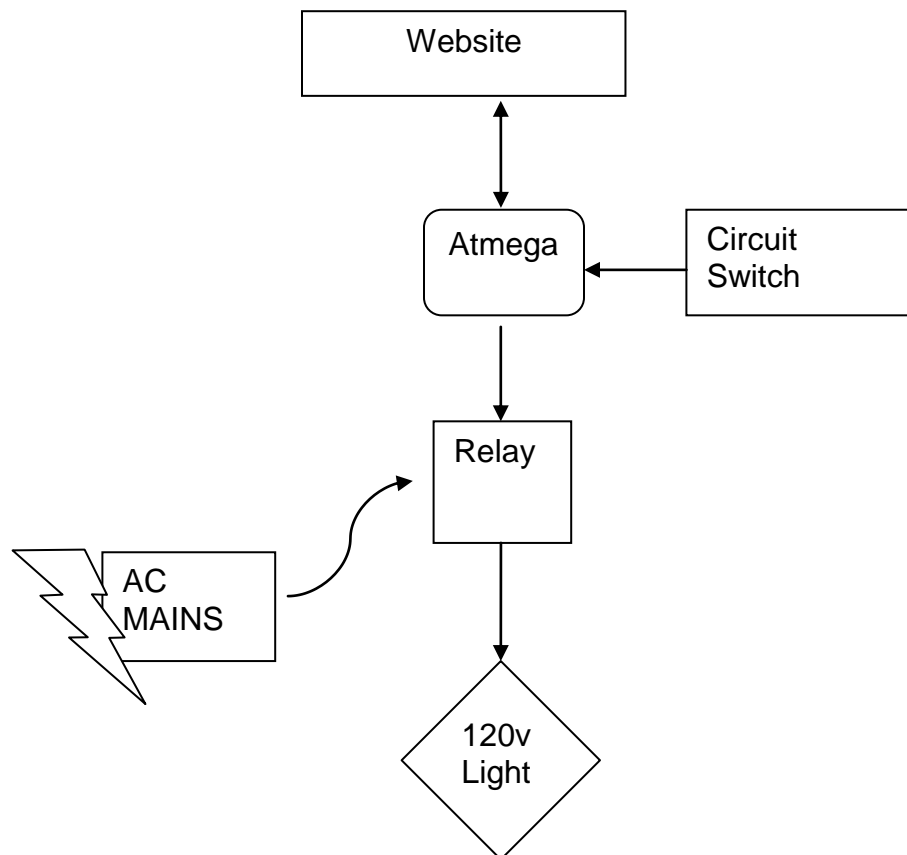
3.2 Light Control

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 1-1). 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 1-1: 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 1-2). This will allow the group to use software (example code shown in figure 1-3) 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). It's important to note that they would modify the example code slightly, in that instead of using the state change of the button to toggle an LED, they will be sending it to the transistor that will be wired directly to the relay (thus toggling the 120V light socket). 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 1-2: Example Schematic of interfacing Button with Atmega

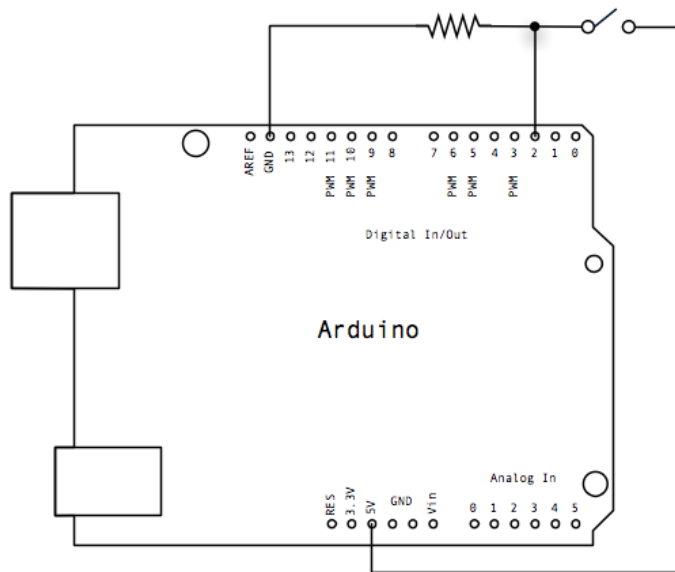


Figure 1-3: Example Code to monitor switch

```
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;     // the number of the LED pin

// variables will change:
int buttonState = 0;       // variable for reading the pushbutton status

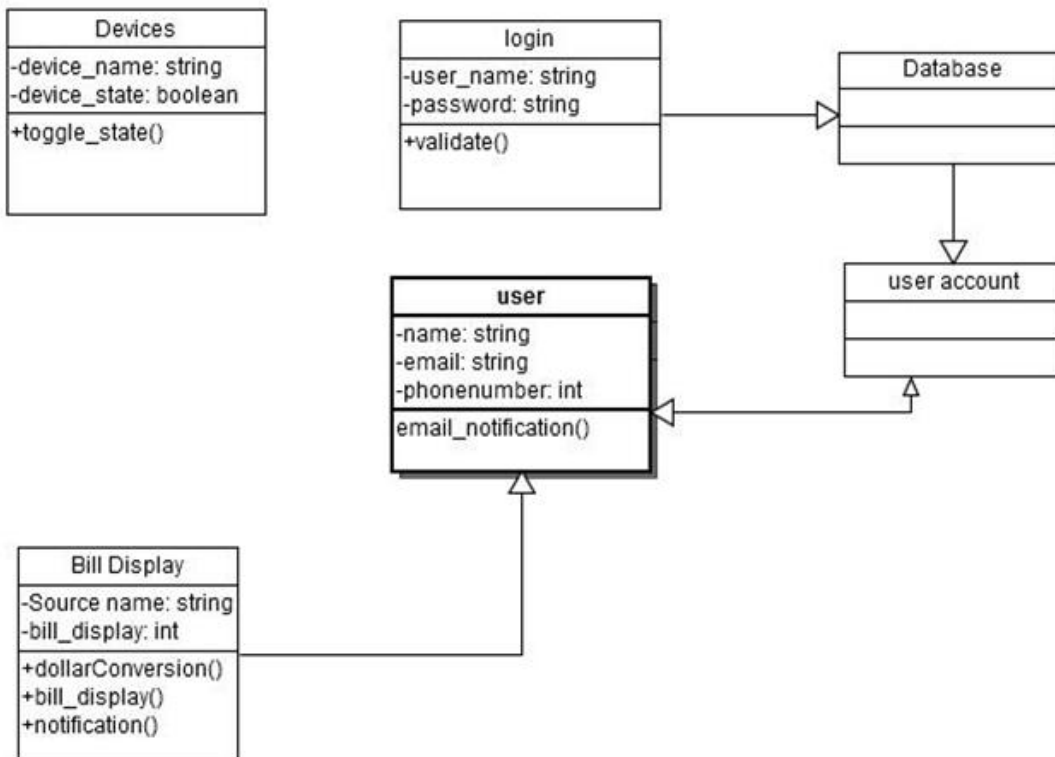
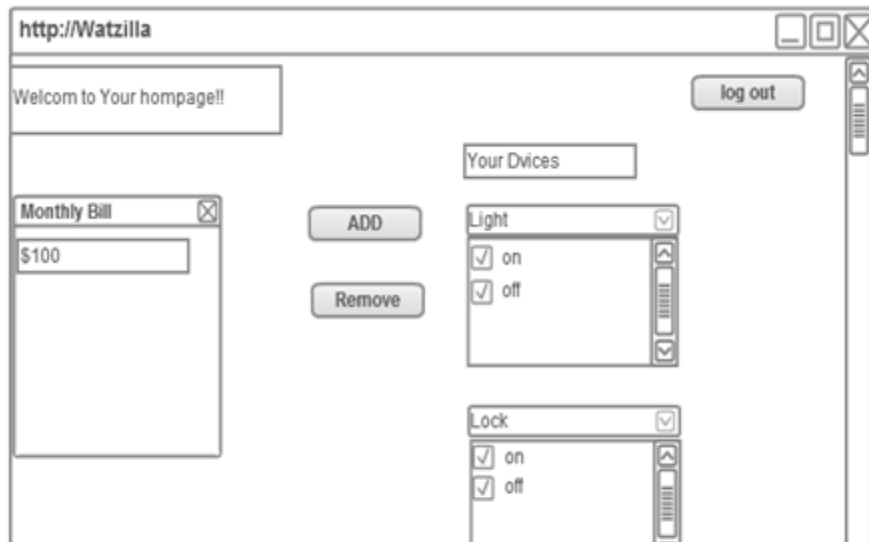
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

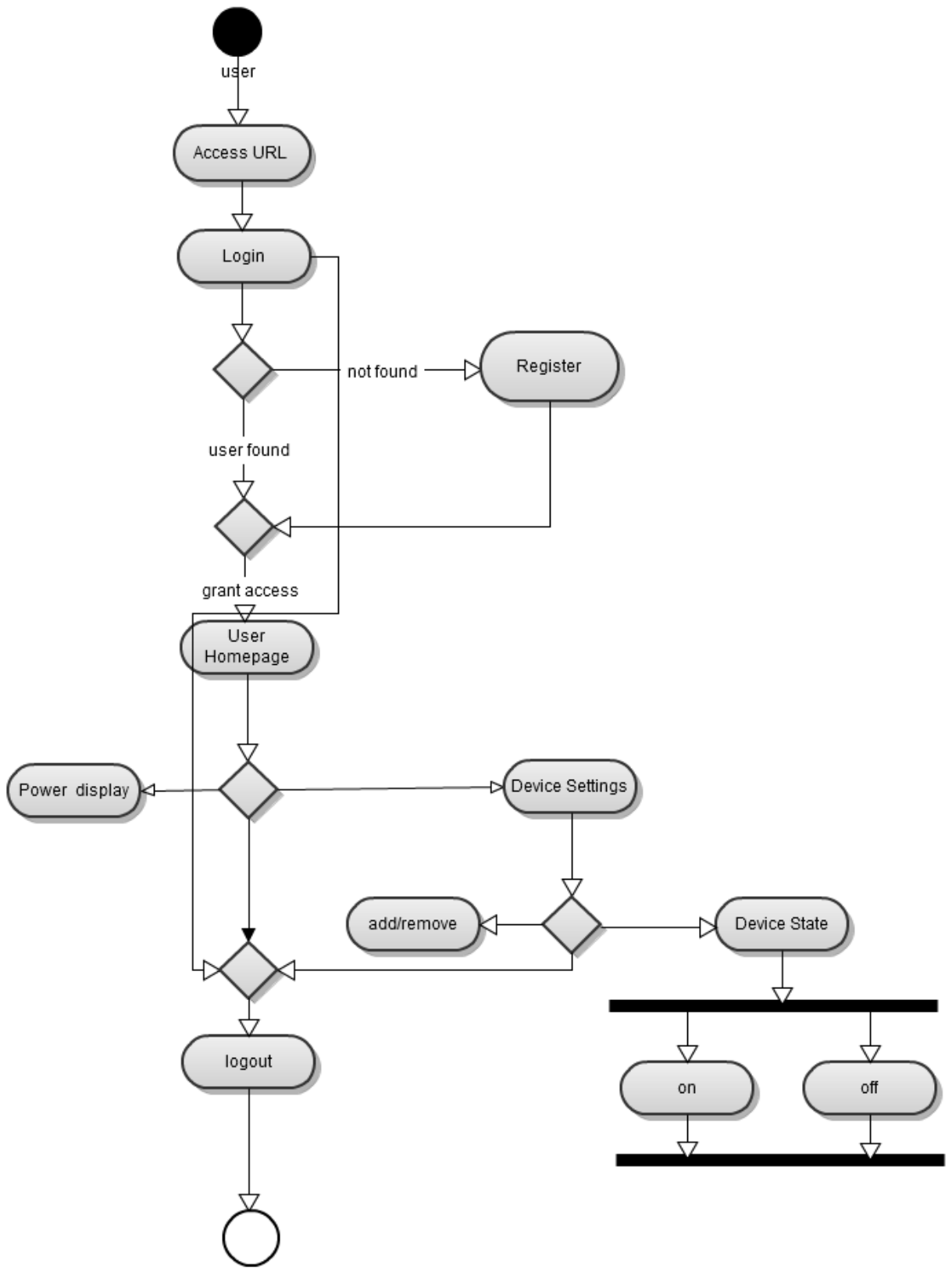
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```


3.3 Website Design

The website will have a login. A user will login using a user name and password. This will allow exclusive access to one person at a time to the home system. This will also keep anyone from accessing control over a persons' home system. Once logged in, the user will have a display of all the devices set up on their network. There will also be a display showing current power usage to date in a dollar amount. Each device will have its own drop down tab. The device will display its current state, identifying either on or off. There will also be an option to change the state of the device within a drop down box. Once and option is selected, the site will then signal the central hub and communicate with the microprocessor on the hub. The projected user interface is shown below. Figure X





3.3.1 User Accounts

One of the issues with creating a network is granting access to the network. We need to make sure only one person can control a network at a time. We want to avoid multiple requests from different users at the same time. To do this we plan on creating user accounts with password protection. PHP is particularly good for these. The advantage php has is its ability to automate server tasks. PHP can create new user details automatically, as opposed to mysql where obtaining user details need to be entered into the mysql database by the system administrator. The first objective of the code will be a choice of display. The code will deciding to display the results of user input or a blank form which requests user details.

```
<?php
$username = $_REQUEST["username"];
if (!isset($username)) {
echo "<form method=post>
Enter New Username<input name=username><br>
<input type=submit></form>";
} else {
```

Once the submit button is entered the code will access mysql tables to process the user input. The mysql can be accessed by embedding the mysql into the php code with an include statement.

```
$SQL = "select * from USERS where username='$username'";
```

And then running this query on the database:

```
$result = mysql_query($SQL) or die (mysql_error());
```

The programmer can now use the PHP mysql_numrows function to obtain the number of rows returned in the recordset:

```
$num = mysql_numrows($result);
```

The code will then check if the user already exists. If the user does not exist further code is necessary to create a new user name.

```
if ($num > 0) {
echo "Username already exists<br>";
} else {
$SQL = "insert into USERS (username) values ('$username')";
mysql_query($SQL) or die (mysql_error());
}
```

And then then connection to the database can be closed:

```
mysql_close();
```

```
}
```

```
?>
```

This process allows users to create their own accounts.

3.4 Phone Design

3.4.1 Android App

Class diagram and words

3.4.2 SMS

```
//this is the url of the gateway's interface
$url = "http://www.tm4b.com/client/api/send.php";
//initialize curl handle
$ch = curl_init;
curl_setopt($ch, CURLOPT_URL, $url); //set the url
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1); //return as a variable
curl_setopt($ch, CURLOPT_POST, 1); //set POST method
curl_setopt($ch, CURLOPT_POSTFIELDS, $request); //set the POST variables
$response = curl_exec($ch); //run the whole process and return the response
curl_close($ch); //close the curl handle
//show the result onscreen for debugging
//print $response;
```

Sockets

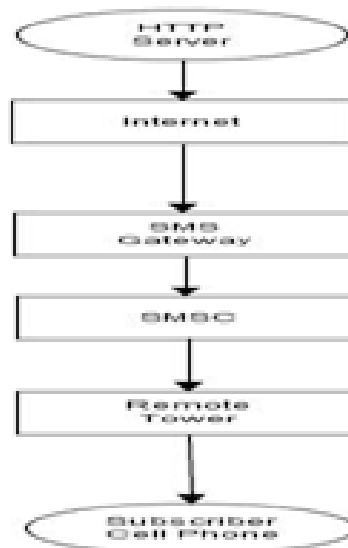
```
//First prepare the info that relates to the connection
$host = "tm4b.com";
$script = "/client/api/send.php";
$request_length = strlen($request);
$method = "POST"; // must be POST if sending multiple messages
if ($method == "GET")
{
    $script .= "?$request";
}
//Now comes the header which we are going to post.
$header = "$method $script HTTP/1.1\r\n";
$header .= "Host: $host\r\n";
$header .= "Content-Type: application/x-www-form-urlencoded\r\n";
$header .= "Content-Length: $request_length\r\n";
$header .= "Connection: close\r\n\r\n";
$header .= "$request\r\n";

//Now we open up the connection
```

```

$socket = @fsockopen($host, 80, $errno, $errstr);
if ($socket) //if its open, then...
{
    fputs($socket, $header); // send the details over
    while(!feof($socket))
    {
        $output[] = fgets($socket); //get the results
    }
    fclose($socket);
}
/*
    the message id's will be kept in one of the $output values
    print "";
    print_r($output);
    print "";
*/

```



3.5 Server/Hosting Design

After considering the website hosting options available to us, as well as the project as a whole, the best hosting option available to us seems to be hosting out own server. Hosting our own server is cost effective, which is important,

because the rest of the project is going to be costly, so minimizing cost wherever possible is a concern. In addition, because one of our group members has experience successfully hosting a website on a personal web server running on his own computer at home, we know that it will work. This will provide us with plenty of freedom as far as how we structure the website and how much space we need for it, since the only limit is the hard drive space on the hosting system. This will also prove useful for development, as all four of us could potentially have the server software set up on our machines, so we would be able to develop parts of the hosted software individually without interfering with anyone else.

Now that we have decided to host our own server, there are 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 has been the deciding factor. We will be using Abyss Web Server for our hosting software, and Dyn for our 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 our project. Abyss beats out Windows Server by being completely free, not just a free trial. If we have any interest in keeping our system running after the senior design period is over, Windows Server is not a good idea unless we plan to pay for it when the trial ends. Dyn has been 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 seems 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 fit into our project. Both solutions offer free versions and the same simple database functionality that we need. Ultimately, the choice comes down to previous experience, and the group member who will be working with SQL the most has more experience with Microsoft SQL Server, so that will be our choice for the SQL software that we use. Now that we know which software we will be using, we need to determine what data we need to be storing and how we will store it.

Data that needs 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 can have their own table and relate to the account table. The IP address of the

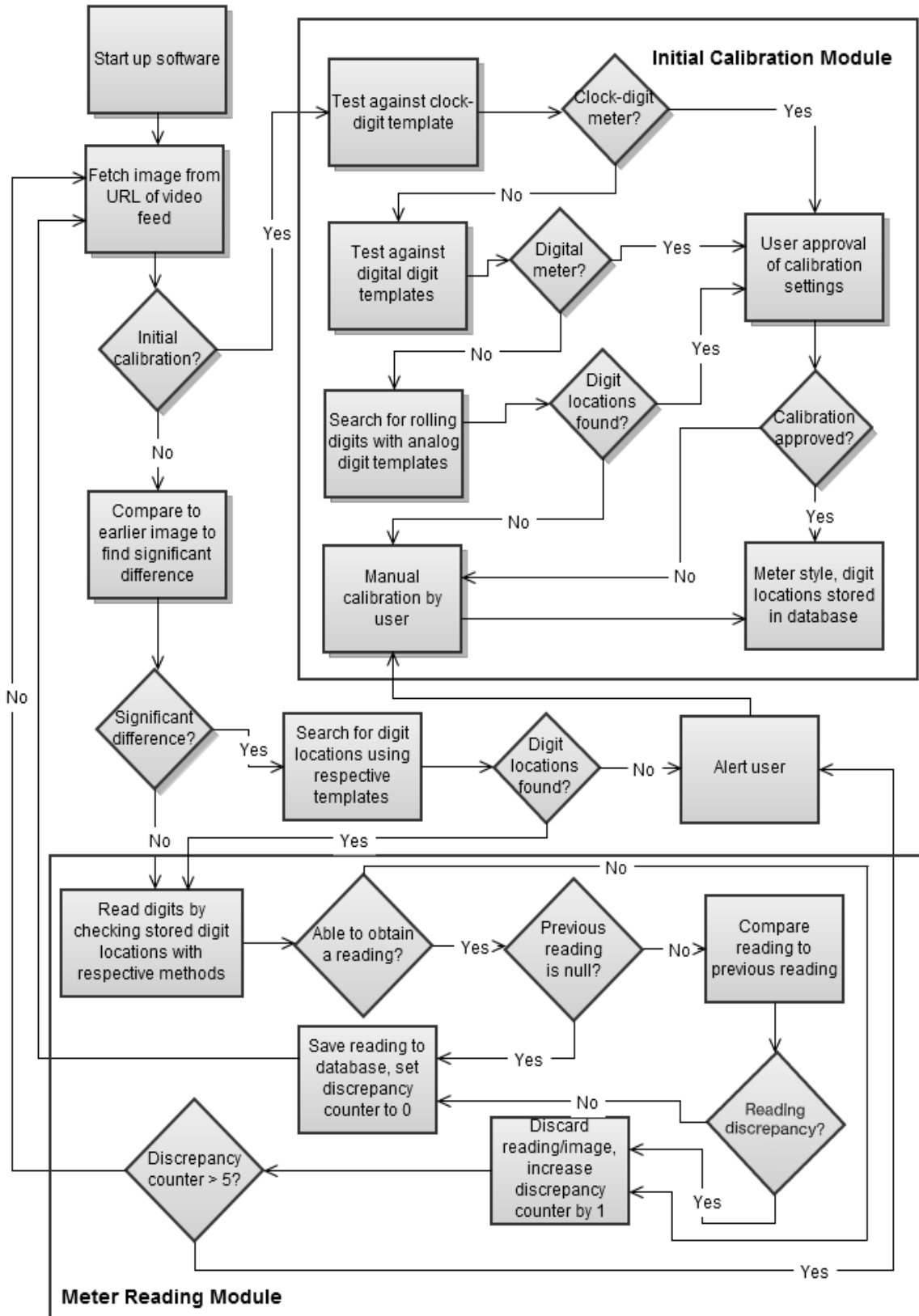
network that the user's central hub is on should be saved in order to facilitate sending signals to the system from the website, and it should have its own table. The monthly bill should be saved for every month that the user is using the system, and we could even allow for them to manually enter in previous bills if they wished to have a greater bill history shown right away on the site.

Other information that must be saved is 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). We will allow 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 will be stored in a separate table in order to allow for the addition at any time of any devices we incorporate into the project. The monthly use for every month should also be saved, just like the power bill, in order to provide a usage history to the user, and it will be saved in a separate table. Door locks will be included as a device, but will not have any monthly power use records associated with them, and their power use so far column will be null.

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, latest meter reading and the discrepancy counter. Style, reading and discrepancy counter can be stored in one table, while digit locations should be stored in a separate table, since there will be multiple locations per meter.

All of these tables will be related using primary and foreign keys. The database will be accessed from the website and from the image processing software, both of which are going to be written in languages that support SQL queries and updates. Integrating SQL into the rest of our project will not be any problem.

Figure [K8] – Flowchart of software operation



3.7 Image Processing

The image processing software needs to be able to take in a picture of a residential power meter and produce a number representing the reading of the meter. To this end, reading the meter will be divided into two main parts. The first part of the process will be determining which of the three types of displays discussed in section 2.3 the meter has, since each type of display has to be processed differently. This first part also includes finding the digits to read on the meter, because of the way the meter determination works, and will even read the meter for the digital and analog rolling meters. This first part will be considered the calibration step, and the user will have to approve the software's determinations before it uses them for regular monitoring. The second part of the process is the regular reading of the meter, which depends on the type of meter. There is more to the software than just its functionality, however, and these other considerations will be discussed first.

Our group has decided that having our software run on our web server would be of greater benefit to our project than running it on a microcontroller. Our project needs a website whether the image processing software is on it or not, and there is no limiting factor to why we shouldn't run the software on the server. Running the software on a web server 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 will be developing the majority of the image processing software has lots of experience with C# and ASP.NET, which are supported natively by Abyss, our chosen web server (see Section 3.2.2), and will be the language and platform that the software is developed in.

After considering the computer vision software options discussed in section 2.3.2, it has been decided that writing our own custom software will provide us with the best results. We will borrow 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 will be custom-made, using the existing libraries as reference material only. Creating our own software allows greater flexibility and more control over the way our software performs. Rather than trying to adapt the broader functionality of OpenCV, CVIPtools or AForge.NET's entire libraries to our needs, writing the software ourselves with the express purpose of reading a residential power meter will allow our code to be more concise and to the point, with no unnecessary extras to worry about taking up time or system resources. Theoretically, our custom software should be able to read a power meter faster than if we were to adapt the existing libraries, because there will be 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 may be 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 has been determined that for our purposes color information is unnecessary, so upon obtaining an image for processing it will immediately be converted from color to grayscale using the RGB-to-grayscale function borrowed from AForge.NET.

Once an image is ready for processing, the first part of the actual functionality of the software is to identify the meter type. It should be relatively easy to distinguish a meter of the analog clock-digit variety. Distinguishing between a digital meter and a rolling-digit style meter will be more difficult since they both use actual digits. The style of the digits will be important in distinguishing between those two types. Even though they both use digits, the main reason it matters to be able to tell the difference is because while digital digit only have 10 discrete states it can be in, while the rolling digits roll in real time, which will need to be accounted for.

It is important to note that a user's meter style should not change unless that user moves and brings the system with them. As such, the meter style only needs to be determined once, the first time the system is used with a particular meter. As such, determining the meter style, while crucial to being able to read the meter, would be a waste of time if done on every image that is processed. The solution is to make this part of the software a separate piece that is only run on initial user system setup, or if the user tells the system to run it again for whatever reason, such as they've moved and have a new meter, or they just want to recalibrate the system. The meter style could then be saved for the user, and would determine which method should be used for actually reading the meter.

We have decided that the simplest way to determine the type of meter is by first checking if the meter is an analog clock-digit meter. We decided this because the analog clock-digit meter is both the easiest to distinguish and the most common meter currently in use. If the meter is found to be clock-digit style then the meter style will be saved and the software will move on to the next step. If not, then the software has determined that the meter is one of the other two styles, and it must determine which one of those it is. The software will check to see if the meter is a digital meter and if so, that will be saved, otherwise the style will be saved as analog rolling style.

In order to check if a meter is analog clock-digit style, the software will search the image for matches to a template of a single clock-style digit, seen in Figure [K4].

The template will be resized as necessary if no matches are found with a particular template size. The template will be represented as a 2D array of pixel values, just like the image being processed. In order to find a template match, the template will first be matched with the top left corner of the image, then translated across to the right side of the image, then moved down one pixel and translated across from left to right again, until all possible locations in the image have been checked against the template. Checking against the template will involve comparing the grayscale values of corresponding pixels of the template and the image for each location, and finding the total difference between them. If the total difference remains below a given threshold, then the area of the image is considered a match with the template.

In order to account for different possible lighting conditions throughout the day, the sections of the image to be tested will have their pixel values normalized to a comparable range to the template. This will allow template matches that are lighter or darker than the actual template to be accurately identified. The normalization will be done by adjusting the lightest and darkest pixels of the search image area to match the lightest and darkest pixels of the template, respectively, adjust all of the other pixels' values proportionally. This normalization will happen individually for every instance of the template comparison, in order to ensure proper lighting conditions. A template match in the image may not register because it was a different size than the template. In order to account for this possibility, the template will run across the image several times across a range of template sizes.

If at least three but no more than seven matches of the same size are found, then the meter is considered an analog clock-digit meter. This method eliminates the need to separately search for the digit locations, because the match locations are the locations of the digits, so another search would be redundant. The match locations are saved in the database and the software moves on to regular monitoring, after user approval. If less than three or more than seven same-sized matches are found, then the software considers the meter to not be analog clock-digit style. One thing to note for clock-digit style meters is that while the number on the top point of the digit circle is most likely zero, it could be possible that this is not the case. After digit locations are determined, the software will run a check on the top area of the leftmost clock-digit, checking for a zero with the zero template for analog rolling-digit style meters.

If the meter is not considered clock-digit style, then the software will check for seven-segment digits that would be found on a digital meter. This checking will be done in a similar manner to the clock-digit check, by running a template across the entire image looking for matches. However, instead of a single template, thirty templates will be used, three for each digit, because it could be possible that at the time the image was recorded, the meter was showing all the same digit, and there is no way to know which one other than checking for all of them. These templates are displayed in Figure [K5], along with an explanation of why there are three versions of each digit. There may be some digits on the

display that are not part of the meter reading, for example like the image back in Figure [K3], but the meter reading digits are assumed to be the largest ones displayed. If this is not the case, our software will provide a manual option for doing all of these classifications which will take care of most problems. If the search finds a set of three to seven same-sized digits that are lined up appropriately, the meter will be considered a digital meter. If more than one set of digits is found, the largest set will be considered the set to read from. The location of the digits will be saved just like before, but because of the templates used for finding digits, the meter can actually be read at this point. The digit template that found the match also tells the software what that digit's value is. So, after user approval, the software saves the meter style, digit locations and first reading, and moves on to regular monitoring.

If the meter style is determined to not be digital, then the style is defaulted to analog rolling, by process of elimination. Again, if this is not correct, the style can be set manually if desired. While the style of meter is determined by default, the digits still need to be found. This meter style is the most difficult on which to find the reading digits, because there are likely many other digits on the meter for other purposes, such as serial or other identification numbers, etc. Unlike the digits of the digital meter, the rolling digits will be very similar if not the same in style and even size as any extra digits. The one key thing that will almost certainly distinguish the power reading digits from any extra digits is how lined up the digits are (or in this case, are not). Because the digits on a rolling meter roll in real time, it is very rare that they all perfectly line up horizontally. All other sets of digits should line up with each other perfectly.

Figure [K4] – Analog clock-digit template for finding meter style and digit location. It is not possible to know which direction the pointer will be pointed in, so only the center area of the pointer is included in the template. Created by Kyle Ferguson.

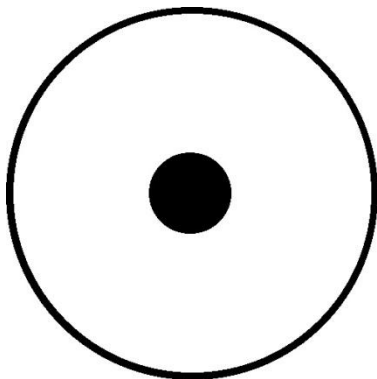
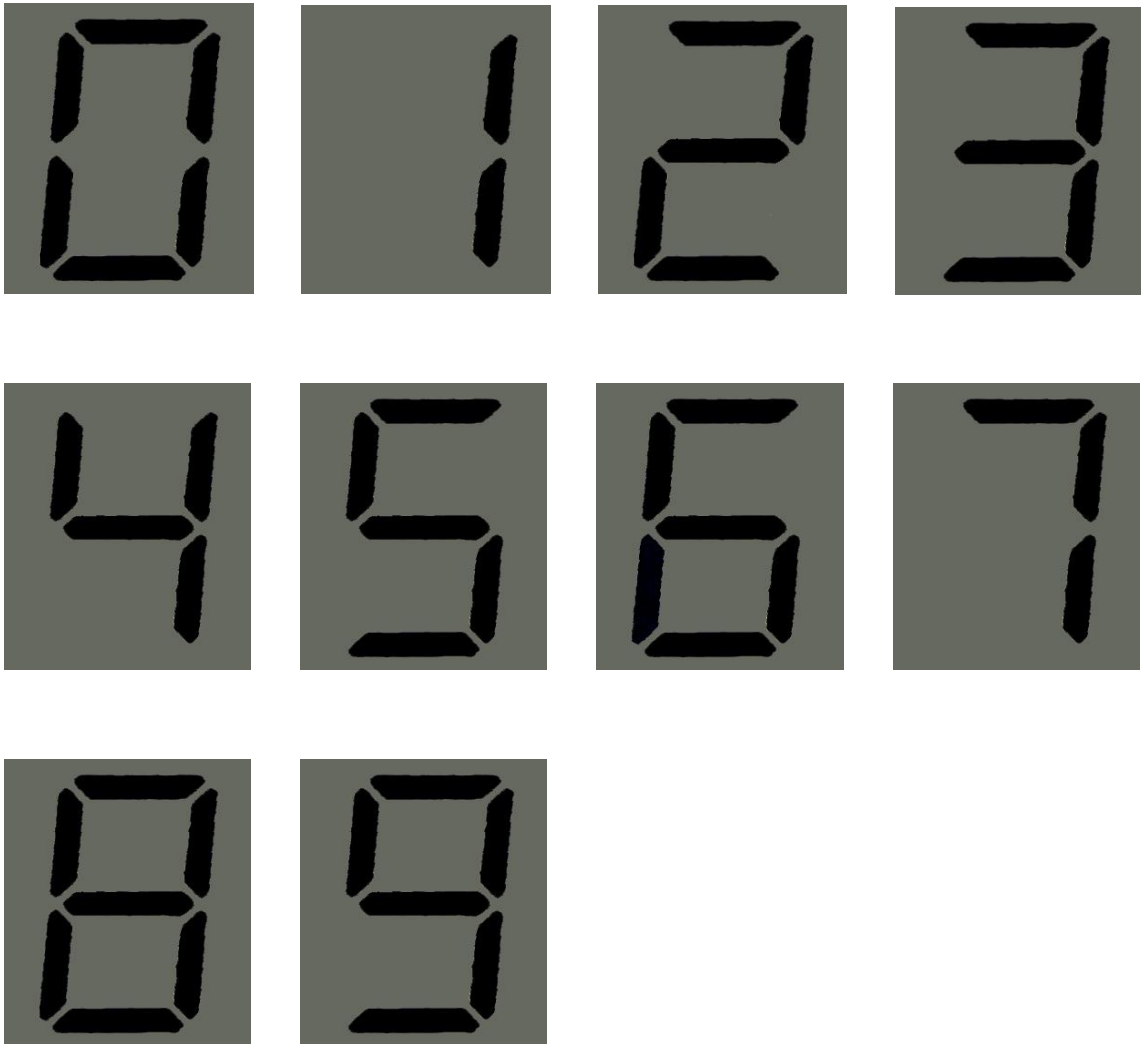


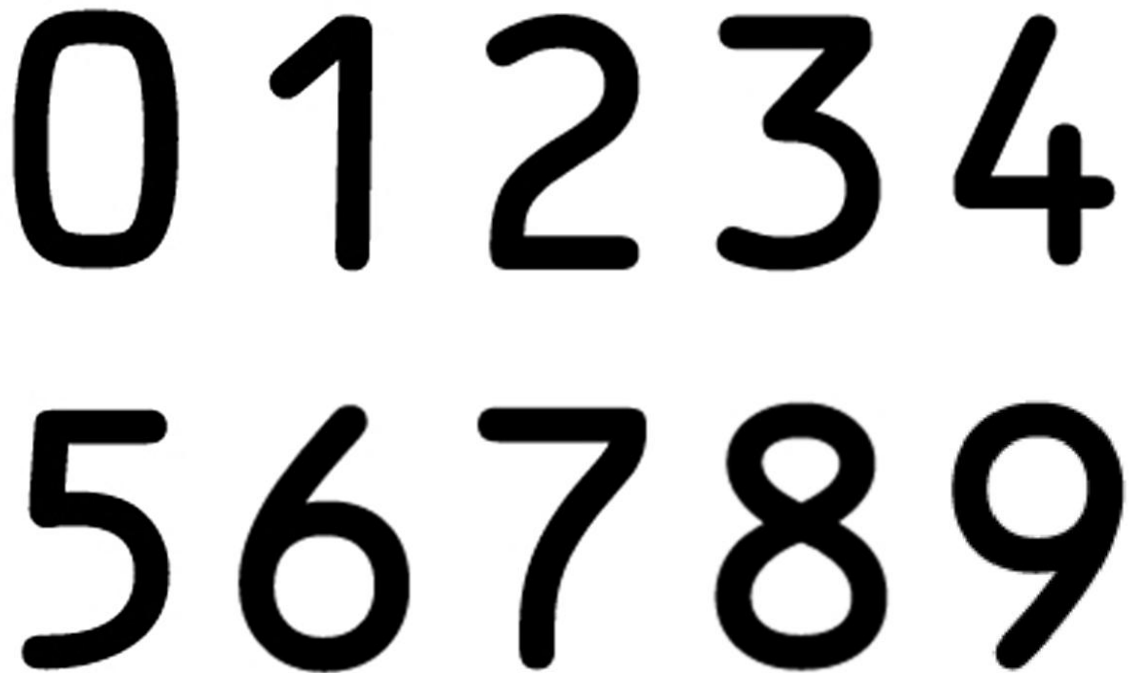
Figure [K5] – Digital seven-segment digits for finding meter style, digit location, and reading a digital meter. There are two more sets of digits that are not show. One set is slightly more skewed, and the other set is not skewed at all. The

backgrounds are gray because the backgrounds for the displays are usually some shade of gray, not white. Created by Kyle Ferguson.



So while searching the image for digits will use the same template matching methods as the digital meter, the templates for which are shown in Figure [K6], there will be a few extra features to the search. Sets of digits will be grouped together if they are close together and line up horizontally within a certain threshold. Most sets will line up almost exactly, but because the digits we are actually looking for will probably not line up, we need to allow some flexibility in what is considered “lined up”. If there is a single set of digits that don’t line up perfectly, then that set will be considered the power reading set. Additionally, because any number of fonts could be used for these numbers, the matching threshold will be slightly lower than for the digital meter to account for font differences.

Figure [K6] – Analog rolling-meter digits for finding digit location and reading the meter. Another set of digits will also be used to account for some of the more significant variations of the digits for three, four, six and nine, for example. These other templates are not shown. Created by Kyle Ferguson.



Another complication with finding the digits (and later with reading them) on this type of meter is that digit rollers will often be between digits, displaying only part of the two digits they are between. In order to consider this, while searching the image with the templates, the top and bottom halves of each comparison location will be considered separately, and their differences combined. If one half of a template comparison has a high match probability but the other half doesn't, then this will be considered a special type of match and will still be considered when grouping horizontal digit sets, as it is likely that this match is part of the power reading digit set.

If only one digit set of three to seven digits is found, perfectly lined up or not, then those digit locations will be considered the reading digits. If more than one digit set is found but none of them (or more than one) are crooked, then the software will wait 15 seconds and then try again, with the assumption that this will give the digits some time to become crooked. This check will run for up to five minutes from when it first starts trying to find a crooked digit set to be the reading digits. If at this point no appropriate digit set has been found, then the user will be prompted to manually tell the software about their meter. Otherwise, if an appropriate digit set is found, the software will take that set and adjust the digit

locations so that they do line up but are expanded both higher and lower than the actual digit locations to allow the template to work correctly. It will then ask for user approval and save meter style and digit locations, then move on to regular monitoring. Similarly to the digital meter, because of the method of finding digits, a first reading is possible at this step, so that will be saved as well.

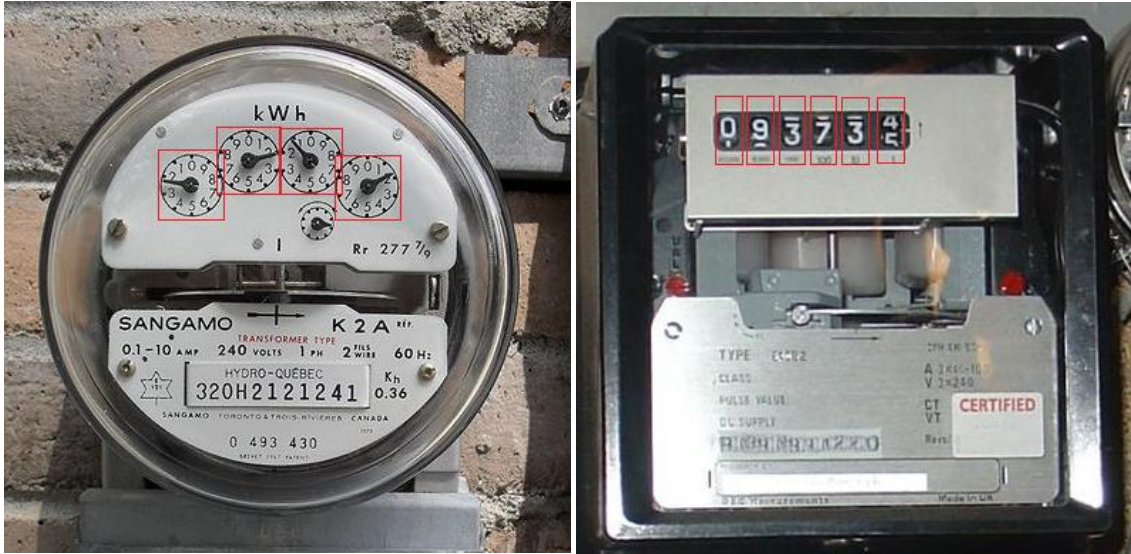
If this calibration program is unable to find suitable digit locations in its analog rolling meter search, then it will report that it was unable to determine the meter style or digit locations, and will ask the user to set these manually. Additionally, if the program does decide on a meter style and digit locations, but the user does not approve them, then the user will be asked to set them manually. This will all be done on a graphical user interface on the website. The user will select between the three meter style and specify the number of digits their meter displays for reading. The user will be shown the image that was used for calibration, and will need to drag a box around each individual digit, trying to make them all about the same size, to tell the software where each digit is. If the user is satisfied with their selections, they can save them as their meter's style and digit location settings, and the software will begin regular monitoring.

When the user is approving the software's determined digit locations, he or she will be presented with an image such as the one in Figure [K7], where the locations are marked by rectangles. Similarly, if the user must manually provide the image locations, he or she will be presented with the original image that was used to attempt to find the locations, and will provide the software with the digit locations by placing rectangles around them in a similar fashion to Figure [K7].

Like the meter type, the location of the digits on the meter will not change unless the user gets a new meter by moving or replacing their existing meter. It is possible however that the digits' locations in the image may change if the camera is bumped or otherwise changes position in any way. While it is all but guaranteed that the user will know if the meter type needs to be recalibrated, this may not be the case for the digit positions, as the user cannot be expected to know every time the camera is accidentally bumped. So, while the digit positions in the image is not likely to change and should be stored with the meter type, manual recalibration in the event of a change is not a good requirement. The digit positions could just be found again in every new image, but that is a lengthy process, so it should be avoided when possible.

The best solution would be to implement some sort of detection of a change in the position of the digits. This will be done by comparing a given image to a previous image that was processed recently, but not the immediately previous

Figure [K7] – Meter image with digit locations marked by red rectangles. Original images used under Wikimedia Commons license



image, to make sure there is enough time difference between images so that a view change will be obvious. The images will be compared by finding the total difference between the grayscale values of each corresponding pixel, not including the pixels in the areas determined to be where the digits are, since those should be allowed to change. If the total difference between the two images exceeds a given threshold, then the digit finding portion of the software will be run again to recalibrate the digit locations, and the new digit locations will replace the old ones. Even if there is a false triggering of this recalibration by, for example, a person walking past the meter and briefly casting a shadow on the meter, there are no negative side effects aside from a possible brief delay in reading. The digits positions will simply be found again in the same places as before, and the program will continue to operate normally.

If the software determines that the digit positions must be found, then it will begin one of the three methods for finding those digits, depending on the meter style. In either case, the digits will be found by using the same technique as was used for initial calibration, but instead of checking for each type, the software will already know which meter type to use, so it will just use the appropriate search technique. If it has problems finding the digits, it will alert the user somehow and ask for manual input of the digit positions from the GUI.

When the software knows where the digits are in the image, it will attempt to read them for each image it processes. The template comparisons for this part of the program will be much shorter than in calibration and digit finding, because the templates will only be compared to the digit location areas of the image. Each meter style required a different method to read, though the digital and analog rolling meters will be more similar.

For the clock-digits, the meter must be read by determining what direction the pointer is pointing in. This will be found by looking at the area directly surrounding the middle circle of the digit to try and find where the highest concentration of very dark pixels is. That position is where the pointer is pointing,

which is why it has much darker pixels. The number that the pointer is pointing to will be determined by where this dark area is in relation to the positions of the numbers on the digit circle, which by default has zero at the top and goes up to nine going clockwise, but can be manually set by the user if their meter is different. This reading method will be done on each digit, which will give the meter reading.

Reading the digital meter is the most straightforward of the meter reading methods. Each digit location will be searched with the digital digit templates, using the templates for the last reading's value of each digit and going up from there, since it is most likely that most of the digits shouldn't be changing anyway. If the expected last digit comparison produces a high enough match value, the program doesn't bother to check the rest of the values. This is done for each digit, and the results are combined to get the meter reading.

The analog rolling-digit meter will be the most difficult to read, due to the possibility of digit rollers being between digits. Using the last meter reading as a base for what digit to look for in each digit location, similar to the digital meter method, will provide some help with readings. The area for each digit location will be taller than the digital meter locations, in order to allow for comparing templates to much higher and lower parts than just the middle of the area. The template matching difference will be considered for the top and bottom halves of each template separately, as well as both halves together as a whole. When the template is checking higher up in the area, it will focus more on the bottom half, and vice versa for the lower area. This should allow for matching templates even with part of a digit obscured. Once each digit has been template-matched and its value found, they will be combined into the meter reading.

Once a reading has been obtained, it is compared to the previous reading for verification. The latest reading should not be very different from or less than the previous reading unless the meter has rolled over from all nines back to zero. If the reading is less than the previous reading or is more than a determined amount greater than the last reading, or a reading was unable to be obtained, the image and reading will be discarded and the software will move on to the next image. A discrepancy counter will exist to keep track of the number of times this happens in a row, increasing by one every time it happens. If the counter reaches five, the next failed reading will result in the software alerting the user that something is wrong so that they can attempt to resolve the unknown issue.

To summarize the process flow, the software will obtain an image from a video stream to use for processing. If this image is to be used for calibration, the software will attempt to determine and save the meter type as well as digit locations. If not calibrating, it will compare the image to a previous one to check for image location changes and adjust accordingly. The software will then read the meter and compare its reading to previous readings to ensure the reading makes sense and will discard readings that don't make sense. If at any point in this process the program is having problems figuring something out, it can alert

the user so they can attempt to alleviate the problem. A flow chart of how the software works is displayed in Figure [K8].

3.8 Central Hub

3.8.1 Central Hub Development

Given that the ATmega328 is sufficient for every aspect of the project, not to mention that it allows for many useful scalability options, it has been selected as the microprocessor for the project. One of the most appealing aspects for choosing the ATmega328 as the main processor for the project is the Arduino development environment. The time that can be saved not getting stuck on little details can be spent scaling the project, which it is looking like what will end up happening. 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 is more than enough for the given applications under development.

An entire development environment for the ATmega328 can be bought for less than \$20. Also, given that the ATmega328 can perform each of the tasks required by each node in the sensor and control network, it can easily serve as the microcontroller for each of these circuits. This will simplify the design process greatly, being that the processor will be programmed in the same fashion for each circuit, and will work similarly for each task it has been assigned to fulfill.

Another exciting aspect for development using the ATmega328 development environment discussed above is that it is highly compatible with the Arduino Ethernet Shield, which provides a reference design for Ethernet communication. This will be extremely handy, being that nobody in the group has directly had experience setting up this kind of circuitry. Furthermore, this component can be stacked simultaneously with an Arduino XBee shield. Looking at the circuitry, most of the connections would conduct through each layer of silicon. The Ethernet would be a priority to stack directly to the Arduino, so that the Serial Peripheral Interface (SPI) Bus can 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 is 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 are two more signals that are needed:

the first is a grounding signal. This will be important to make sure that the XBee Shield is not floating. The second signal that will be required by the XBee Shield, but not supplied by the Ethernet Shield, is a 5 volt power source. This can be found in multiple places by observing the trace carrying 5 volts on the Ethernet board. The first location where this can be done is by running a wire from the pin header location shown in FIGURE1.



Figure 5: 5 Volts in Header

In the event that a wire can not 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 FIGURE2.

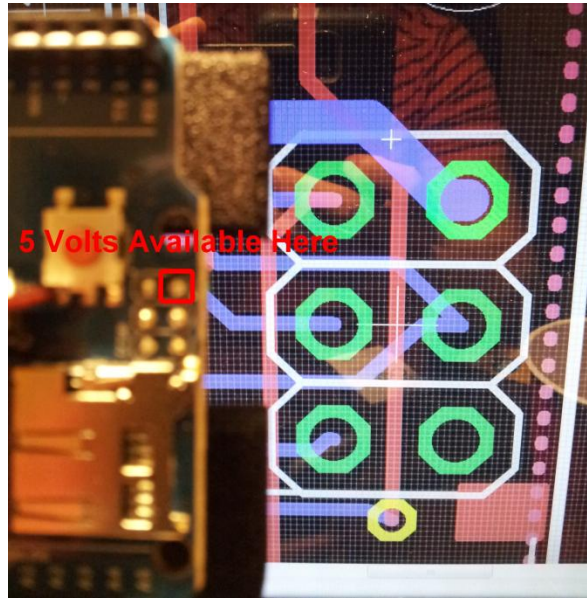


Figure 6: 5 Volts in ICSP Header

3.8.2 Final Hardware Design

The primary and most notable component on the central hub is the ATmega328, which serves as the main processor. This part is capable managing and controlling all the communication that will be fed through the central hub. The ATmega328 is powered by a 5 volt signal, provided by the MC33269D-5.0 voltage regulator, and must be grounded so that no charge will accumulate on it. It has 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 has up to 6 Analog to Digital Converters (ADC's), and at least 6 output pins, which make it a suitable candidate for the Outlet Monitor and Controller circuit, Lock Controller circuit, and the Circuit Breaker Monitor circuit as well.

There are three different package configurations available for the ATmega328 microcontroller. The MEGA8-AI package corresponds to the Thin Plastic Quad Flat Package (TQFP), and must be soldered directly to the board. The MEGA8-MI package corresponds to the Micro Lead Frame Package (MLF), is a little bit smaller than the Thin Plastic Quad Flat Package, must still be soldered to the board (but not manually, since there are basically no leads), and also has 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 is the same package used in development, and allows the implementation of a socket, from which the ATmega328 will be removable. For this reason, the Plastic Dual In-line Package, while the Thin Plastic Quad Flat Package and Micro Lead Frame Package both have their advantages, is 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 can now be programmed in the development environment, and transferred easily to the end prototype without any difficulty.

The ATmega328 has a 16 Megahertz (MHz) external clock wired between pins XTAL1 and XTAL2 to supply clock signal. The 16 Megahertz (MHz) crystal is the fastest clock speed that is specified in the ATmega328 datasheet. This should be sufficient for communicating to the XBee module communicating to the W5100 Ethernet chip, and running any other peripheral functions that might 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 has 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 occurs, the software is 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 will simply reboot. This can 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 can be pulled low is by the CAT811TTBI-CT3 voltage supervisory circuit. This component monitors the 5 volt power line, and will 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 become present.

The ATmega328 handles communication with the W5100 Ethernet chip through a Serial Peripheral Interface (SPI) Bus. It performs this by running in SPI master operation mode. The ports used for this of a Serial Clock (SCK – PB5) line which will be controlled by the ATmega328, a Master Input Slave Output (MISO – PB4) line which will transmit information from the W5100 Ethernet chip to the ATmega328, a Master Out Slave Input (MOSI – PB3) line which will transmit information from the ATmega328 to the W5100 Ethernet chip, and a Slave Select (SS – PB2) line which can just be set low throughout the communication, since there will only be one slave device: the W5100 Ethernet chip.

The 74LVC1G125 Bus buffer/line drivers are connected to each of the Serial Peripheral lines Master Output Slave Input (MOSI), Set Clock (SCK) and chip select lines (SCS). There is 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 is because the W5100 Ethernet Chip is powered by 3.3 volts, and the logic level of these signals must be reduced from the 5 volts logic signals that the ATmega328 produces.

There are a series of Light Emitting Diodes (LED's) connected to various networks to show statuses, or signals. Many of these LED's are 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 will be useful in letting the user know which nets have power at a given moment. Also, the LED's connecting the communications lines indicate these connections, and can verify whether or not there even is a connection.

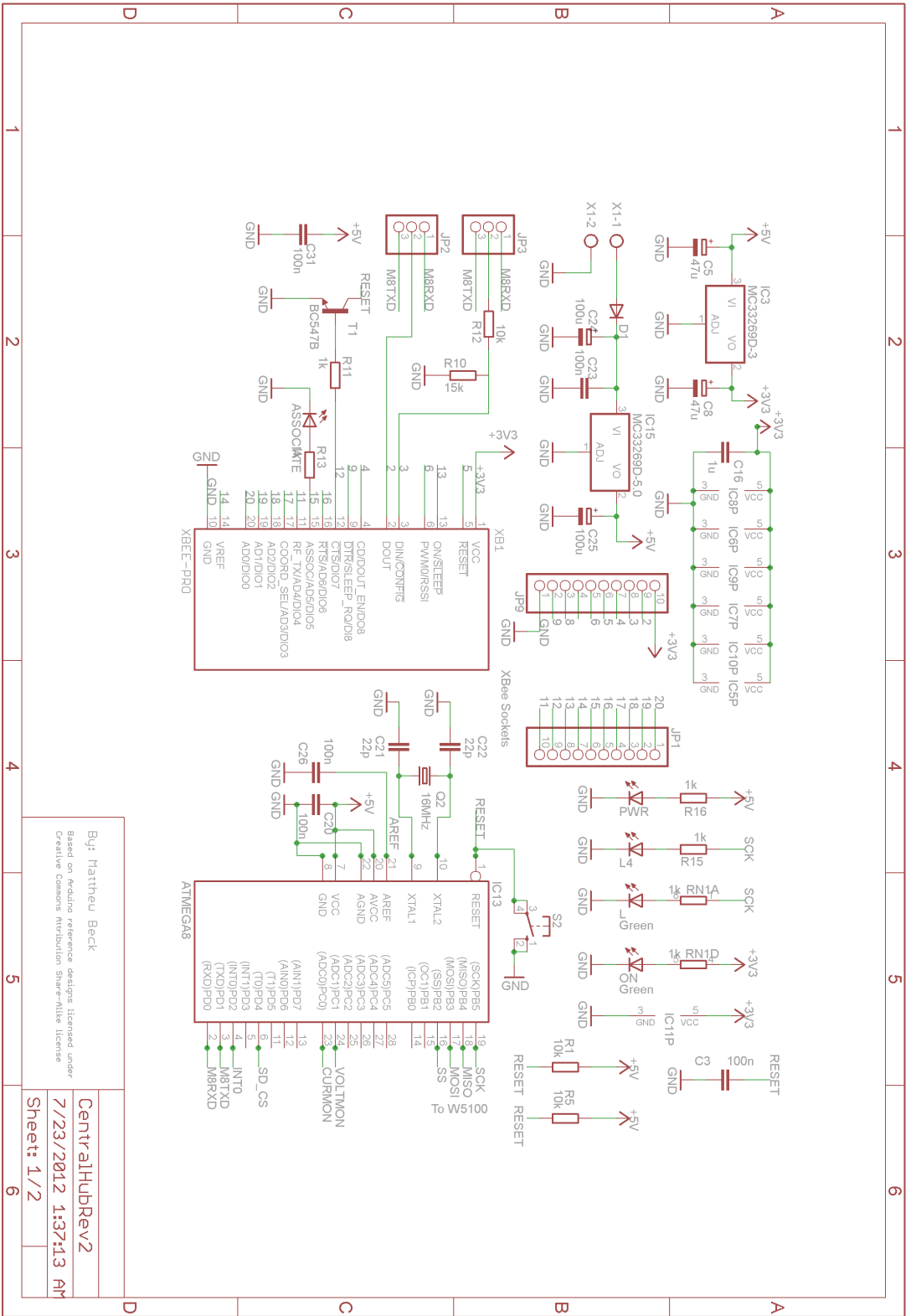
There are two major ports on the board. First is the POE-RJ45, which is the Ethernet cable connector. This will have to be placed on the board with the dimensions of the project enclosure known. It will have to hang off the printed circuit board slightly, so that it can be flush with the enclosure. The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

There is 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 will provide 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 can be handled. The terminals of the solder jumper can be shorted together, or it might have 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 allow the user to change the configuration of which lines are going to be driven as a transmit line or a receive line.

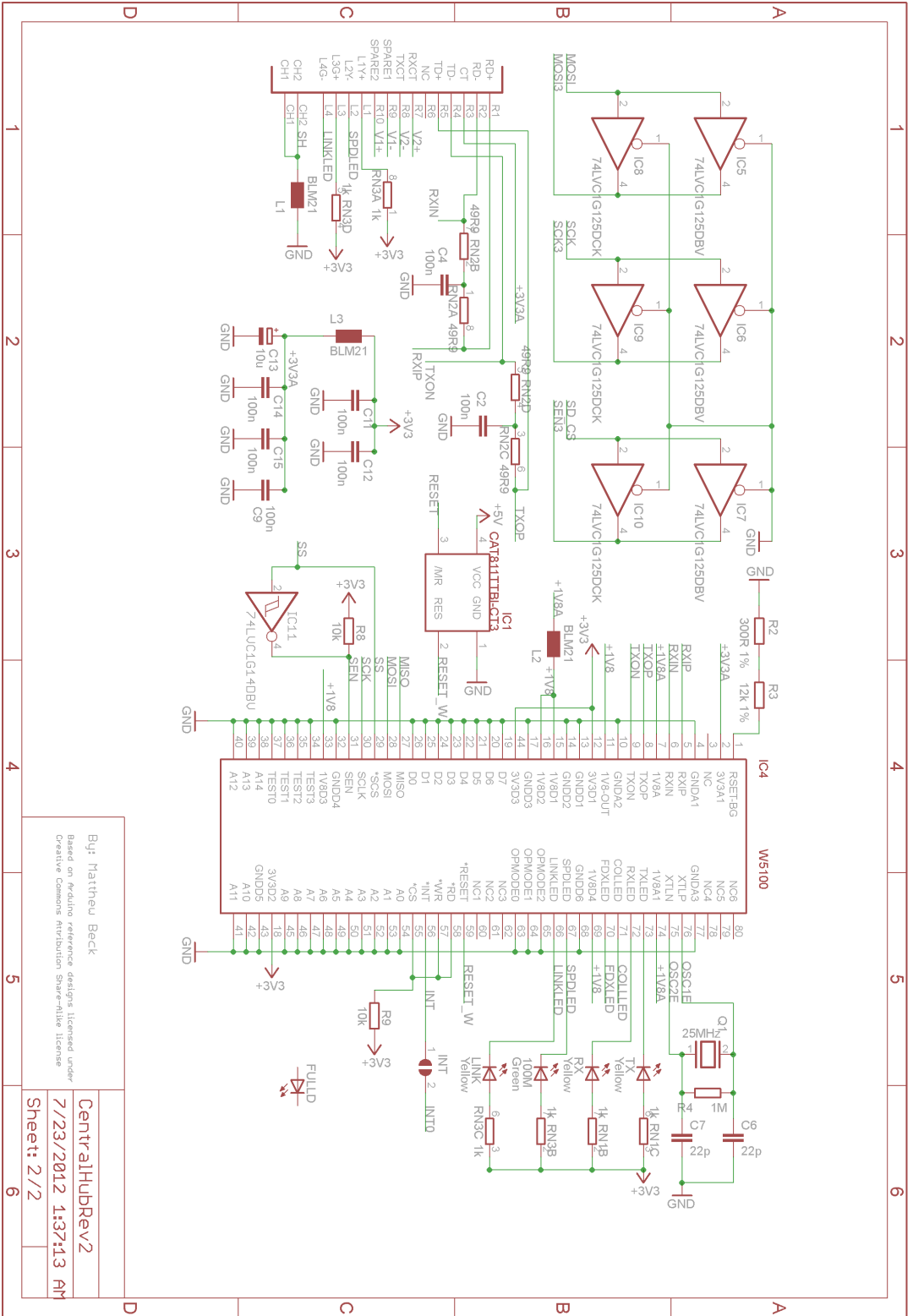
The schematic for the central hub can be seen over the next two pages. Sheet one has the main processor ATmega328, and XBee controller, and voltage regulators. Sheet two on the following page has the W5100 Ethernet Chip, the line drivers for it, the POE-RJ45 Ethernet port



By: Matthew Beck
 Based on Arduino reference designs licensed under
 Creative Commons Attribution Share-Alike License

CentralHubRev2
 7/23/2012 1:37:13 AM
 Sheet: 1/2

Figure 7: Central Hub Schematic – Sheet 1



Buj: Matthew Beck
 Based on Arduino reference designs Licensed under
 Creative Commons Attribution Share-Alike license
 CentralHubRev2
 7/23/2012 1:37:13 AM
 Sheet: 2/2

Figure 8: Central Hub Schematic – Sheet 2

3.8.3 Ethernet Communication

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

In order to use the Atmega328 we 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 we can code the Atmega328 in C it eliminates a huge learning curve for our group and allows us to simply focus on the libraries supplied by Arduino.

For this project we will need to configure the processor to send and receive data from the internet as well as other devices within the Xbee network. Our 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 we will need a long enough cord to support a connection.

To access a specific website we will code in c. We will need to include the libraries SPI.h and Ethernet.h. Enter mac address of the controller and ip address of the server for the website.

```
Ethernet.begin(mac) // obtain an ip address
```

Returns an integer value of 1 for successful DHCP connection and 0 if connection fails

Declarations:

We declare an array of type byte for the mac address and ip address

```
byte mac[ ] = {x,x,x,x,x,x };
```

```
byte ip[ ] = {x, x, x, xxx};
```

```
void setup()
```

```
{
```

```

        Ethernet.begin(mac, ip);
    }
    void loop ( ) {

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address for your controller below.
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress server( );

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;
void setup() {
// Open serial communications and wait for port to open:
Serial.begin(9600);
while (!Serial) {
; // wait for serial port to connect. Needed for Leonardo only
}

// start the Ethernet connection:
if (Ethernet.begin(mac) == 0) {
Serial.println("Failed to configure Ethernet using DHCP");
// no point in carrying on, so do nothing forevermore:
for(;;)
;
}
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");

// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
Serial.println("connected");
// Make a HTTP request:
client.println("GET /search?q=arduino HTTP/1.0");
client.println();
}
else {
// kf you didn't get a connection to the server:
Serial.println("connection failed");
}
}
}

```

```

    }
}

void loop()
{
  // if there are incoming bytes available
  // from the server, read them and print them:
  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }

  // if the server's disconnected, stop the client:
  if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();

    // do nothing forevermore:
    for(;;)
      ;
  }
}

```

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(0xB-0x1A). 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 < 0xFFFF then the data transmitted will be received by any module whose address matches the DL. If DH = 0 and DL = 0xFFFF then all devices may receive data being transmitted. If DH !=0 and DL > 0xFFFF 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 wireless communication between devices and the internet. One function in particular that will be very useful to us is the IPAddress() function. We can define an IP address(camera, hub, server) for communication purposes. Below is a basic example of its use in the arduino1.0.1 environment.

```
#include <Ethernet.h>

// network configuration. gateway and subnet are optional.

// the media access control (ethernet hardware) address for the shield:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// the router's gateway address:
byte gateway[] = { 10, 0, 0, 1 };
// the subnet:
byte subnet[] = { 255, 255, 0, 0 };

EthernetServer server = EthernetServer(23);

//the IP address is dependent on your network
IPAddress ip(192,168,1,1);

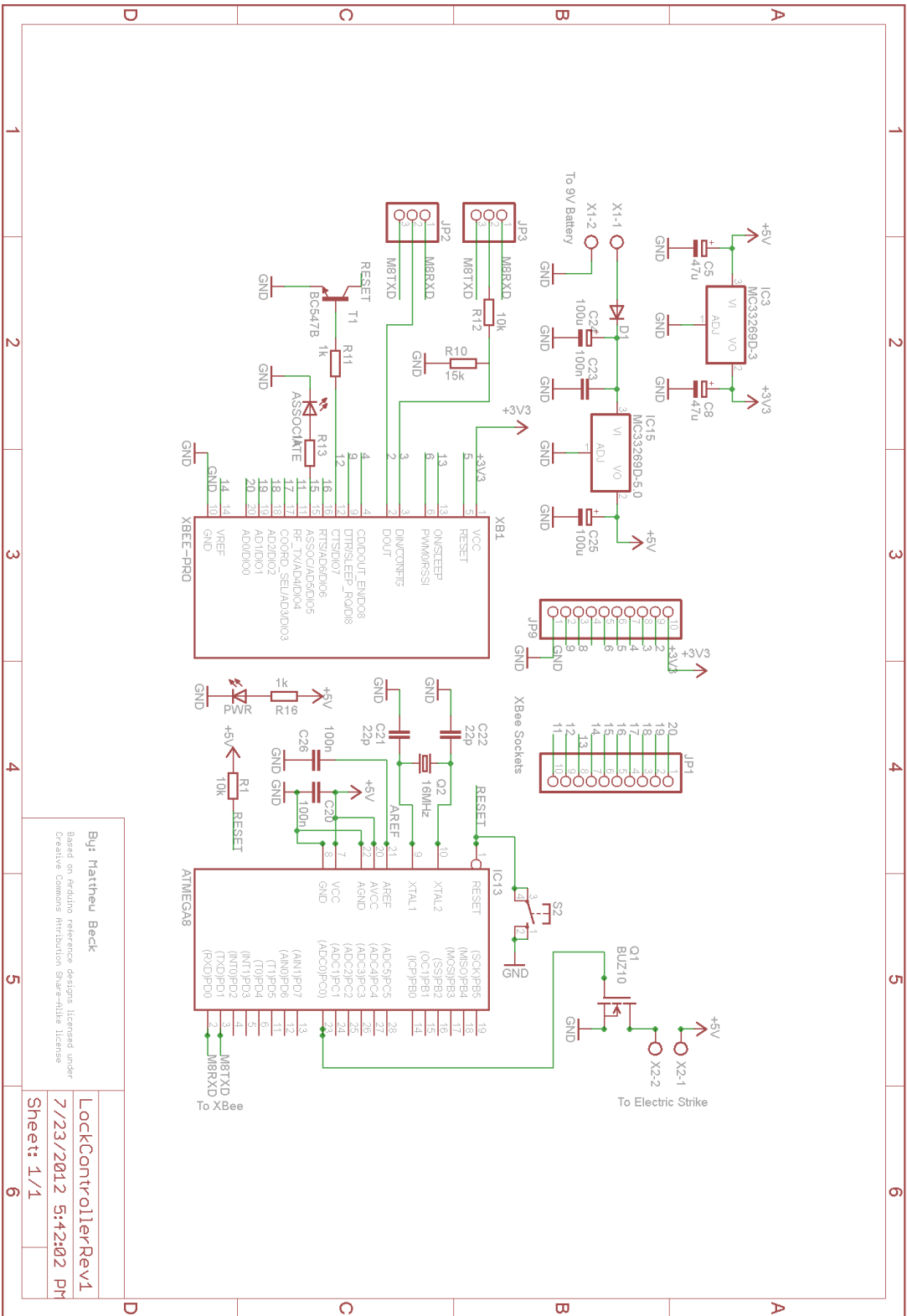
void setup()
{
  // initialize the ethernet device
  Ethernet.begin(mac, ip, gateway, subnet);

  // start listening for clients
  server.begin();
}

void loop()
{
  //print out the IP address
  Serial.println(myIPAddress);
}
```

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.



By: Matthew Beck
 Based on Arduino reference designs licensed under
 Creative Commons Attribution Share-Alike License

LockControllerRev1
 7/23/2012 5:42:02 PM
 Sheet: 1/1

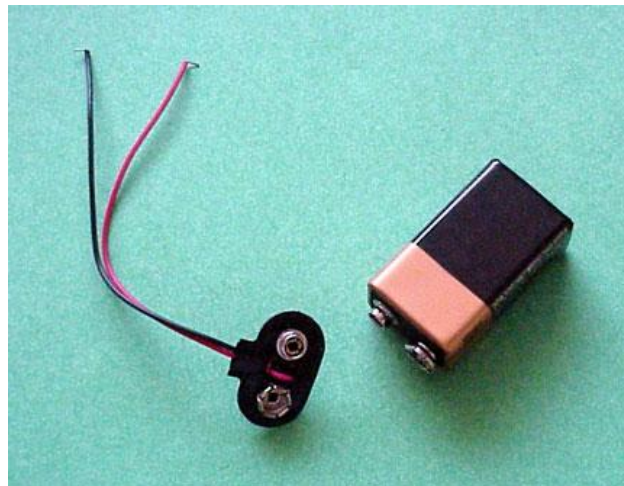
Figure 9: Lock Controller Schematic

3.10 Power Supply

Since a steady 5 volt direct current signal is required by the circuit, a voltage regulator must be chosen that can provide this consistently. The MC32269 voltage regulator series provides low dropout, positive fixed and adjustable voltage regulators. There is 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 has 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 requires a minimum of 6.0 volts at the input can be powered easily by a 9 volt battery. This allows 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 is no need for a battery holder, since the tops of these batteries have terminals that can snap to specialized connectors, and the battery itself can 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 be an ideal option for most end users. This connector can be purchased separately for very cheap, and connected to the board with a two screw terminal connector 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 is that the 3.3 volt voltage regulator can be powered by the 5.0 volt output from the other regulator, which is 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.



4 Prototyping

4.1 Hardware

There are several different components that make up this project. Along with the central hub, there are four potential network nodes at the moment: these include, but are 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 has their own dimensions, which are relevant considerations as they relate to packaging considerations and logistics. Minimization of these dimensions is a design goal, but is secondary to the actual functionality of these devices.

Along with the electronic devices themselves, there are batteries that must 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, reduce the need for hard wiring, and minimize the final packaging dimensions of the final product under development. The outlet monitor and controller will necessitate a three prong AC power socket for which to plug in the end devices that will serve as the target for monitoring and control.

Once the final board dimensions are compiled and laid out, four drill holes will be added to the board in each corner in order to mount it inside of a physical apparatus of some sort. This will take up valuable board real-estate, and should be taken into consideration when designing the final Printed Circuit Boards (PCB's). The size of the drill holes will be dependent upon the selected project enclosure, and the size of the project enclosure will be dependent on the size of the final printed circuit boards; thus, development for these factors should be taken into consideration simultaneously.

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

The project enclosures should have dimensions which are 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 should be screw mounts with the exact dimensions as those on the printed circuit board; these mounts should be positioned so that the printed circuit board is not flat against any surface. This can be done with spacers, or properly sized nuts.

The final prototype of the Central Hub will need to be directly connected to a router or modem. There are two ways to make this connection with an Ethernet cable. The easy way is 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 will not be easily replaceable, which is a significant disadvantage. The second, less significant disadvantage is that the device can not be stored as easily as it would if the Ethernet cable could be detached. Another, more desired approach would be 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 will have to hang off the board slightly. These measurements require precise measuring, and calculations of offsets. To be sure the dimensions are correct, the project enclosure should be purchased first, measured with calipers, and finally the final printed circuit board can be laid out accordingly.

The final prototype for the Outlet Monitor and Controller must have 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 should be able to plug directly into the wall, and any electric appliance which would have plugged directly into that same wall, can now plug directly into the prototype.

The final prototype of the Light Controller will be very similar to the Outlet monitor and Controller. It, too, will have a 3 prong AC power plug, but will not have an output 3 prong AC power socket. Instead, the Light Controller will have a switch and a light fixture. The light fixture will be made with a PVC frame, which will fasten into a wall with angle brackets. In this scenario, the wires bringing power to the light socket will simply run through the PVC piping. At the end of the pipe, the light socket should fit snugly into the PVC pipe. This obviously means that the light socket must be wired before inserting it into the PVC pipe. No lampshades or opaque glass will be implemented as part of the project; the main focus of the project is for electrical design, not interior design.

The final prototype for the Electric Door Strike will be so that the controller box can be accessed easily from the inside of the door. This will 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 that will be 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 must be fitted to properly house the Electric Door Strike. These dimensions will be carefully measured with calipers during development, and documented in the final report.

Additional wiring will be necessary for the outlet monitor and controller, the circuit breaker monitor, the electric door strike, and the light switch controller. It is, 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 can be loosely packaged in the final assembly, and connected to the printed circuit board with wires, or it can be physically mounted to the board. The second option is clearly the optimum solution, and will be desired for the final prototype. The dimensions of the Honeywell CSLA2CD Hall Effect current sensor can be seen in the datasheet, downloaded in the Drawing Exchange Format (DXF) data file, or measured physically with calipers upon product arrival. A combination of all of these methods would be the most promising, 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 are known, corresponding solder pads may be added to the board to provide the mounting capabilities desired for the final prototype.

The Outlet Monitor and Controller will need wiring between the various components that make it up. These components consist of the main circuit board, the 120 volt AC power plug, the 120 volt AC power socket, battery, the relay and potentially even the current monitor if it can not be mounted on the board. Each of the wires for these has specific current specifications. The wires running to between the 120 volt AC power lines must also be 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 will be used for the 120 volt AC power lines. The wire diameter can be 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 20 milliamperes under max current conditions. These lines can be wired with 24 AWG wires.

A grounding plane will be 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 will be directly connected to the ground prong that plugs in with the 120 volt AC power lines. There will also be 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 is the best approach to reduce the possibility of ground loops, which could have negative affects, compromising the system.

During the development phases of the project, it will be 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 figure 1-1) 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.

Figure 1-1: Summary of PCB 123

<u>Criteria</u>	<u>Met?</u>
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 than 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 Figure 1-2) 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.

Figure 1-2: Prototyping with Custom PCB

<u>Criteria</u>	<u>Met?</u>
Quick Turnaround	Yes (2 weeks)
Low Price	Yes (85\$)
Simplicity of Design	Yes
Silk Screen	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 Figure 1-3) 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.

Figure 1-3: Prototyping with Express PCB

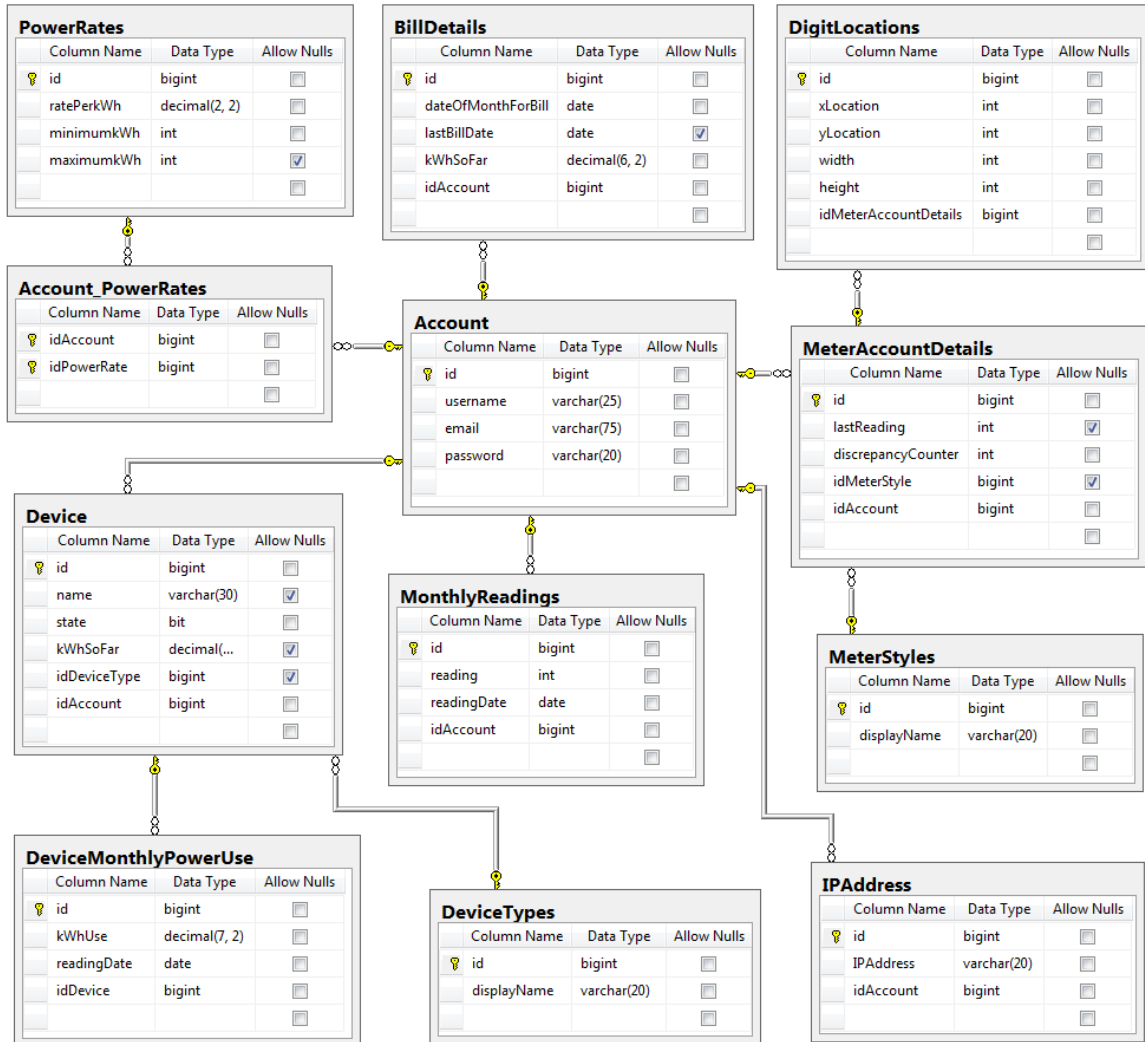
<u>Criteria</u>	<u>Met?</u>
Quick Turnaround	Yes (<1 week)
Low Price	Yes (85\$)
Simplicity of Design	Yes
Silk Screen	Yes

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 our 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, we are providing 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 [K10] below.

Figure [K10] – A diagram of the SQL Server database schema for our project. This database will hold data for our website and our image processing software.



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 our project are the server name, user ID, password, and connection timeout. The timeout can be used

For PHP, a script is used to make a call to SQL server. At the beginning of the script, a connection is made with the command `mssql_connect` using a server name, username and password. The connection is closed once the script finishes, and a connection must be opened back up each time a SQL script is run. Alternatively, a PHP extension library is available from Microsoft for making and using SQL Server connections, allowing more parameters like the ASP.NET

SqlConnection class uses, with connection timeout being of specific interest. This extension looks to be more useful than the pre-existing SQL connection method, so we will use it instead.

4.4 Image Processing Software Prototyping

The image processing software will function as described in the design section, 3.3.2. For the prototyping, some of the functionality will be described in detail using pseudo-code to show exactly how and why it should work. The main functions to be covered are the template matching functions, which are the main components of the meter detection, digit finding, and digit reading functions. The specifics of the process differ slightly for each implementation, but the basics are the same.

The software will be written with a main class for running the processing loop. This main loop will call static methods from five other classes in order to do what it needs to do. Those five other classes are a meter identification class, a digit location search class, a user alert class, a user calibration class, and a meter reading class. Those five classes will not be instantiated, but will instead just exist to hold static methods for calling within the main loop, separating the methods into logical categories. A class diagram of the software is in Figure [K9].

The class Main will be the primary class that contains the processing loop for images. It will depend the JPEGStream class of AForge.NET to obtain an image for processing. The first time this loop is run for a particular user (or if the user specifies that recalibration is needed), Main will run through the initial calibration section of its programming, which will go like this:

```
    If (MeterIdentification.ClockDigitTest(meterImage, clockDigitTemplate))
        Send identification and digit location data to web site user calibration interface
        for approval
    else if (MeterIdentification.DigitalTest(meterImage, digitalTemplates))
        Send data to web site user calibration interface for approval
    else if (MeterIdentification.DigitalTest(meterImage, analogTemplates))
        Send data to web site user calibration interface for approval
    else
        UserAlert.SendAccountMessage("Message")
    wait for user approval/calibration
```

For each of those methods that is called in the if/else statements, template matching will be performed on the entire image. The MeterIdentification methods will actually call on the corresponding DigitLocationSearch methods for the actual template matching, since both of these functions use the same template matching algorithm. For all three methods, the template(s) will be resized a few times since the exact size of any potential matches in the image is unknown. For the clock digit template specifically, the area outside of the circle in the template is less important than the inside, so more weight will be put on any differences

inside the circle than outside. The pseudo-code for the template search will resemble this in all three cases:

```
int diff, numMatches = 0
Rectangle[] digitLocations
for (however many times resizing will occur)
    apply AForge.NET bilinear resizing method
    for (i from the width of the template/2 to the width of the meterImage -
        (width of the template/2))
        for (j from the height of the template/2 to the height of the meterImage
            - (height of the template/2))
            int diff = 0
            for (k from 0 to width of the template)
                for (m from 0 to height of the template)
                    diff += absolute value of (meterImage pixel[i-(width of the
                        template/2)][j-(height of the template/2)] value - template
                        pixel[k][m] value)
            if(diff < threshold)
                numMatches++
                digitLocations.Add(matching area)
return digitLocations
```

For the clock digit template, there will be a weight and an if statement just before the line where diff is increased. The change will resemble:

```
double weight = 1
if (template pixel[k][m] falls outside main circle)
    weight = <1
diff += weight * (absolute value of...)
```

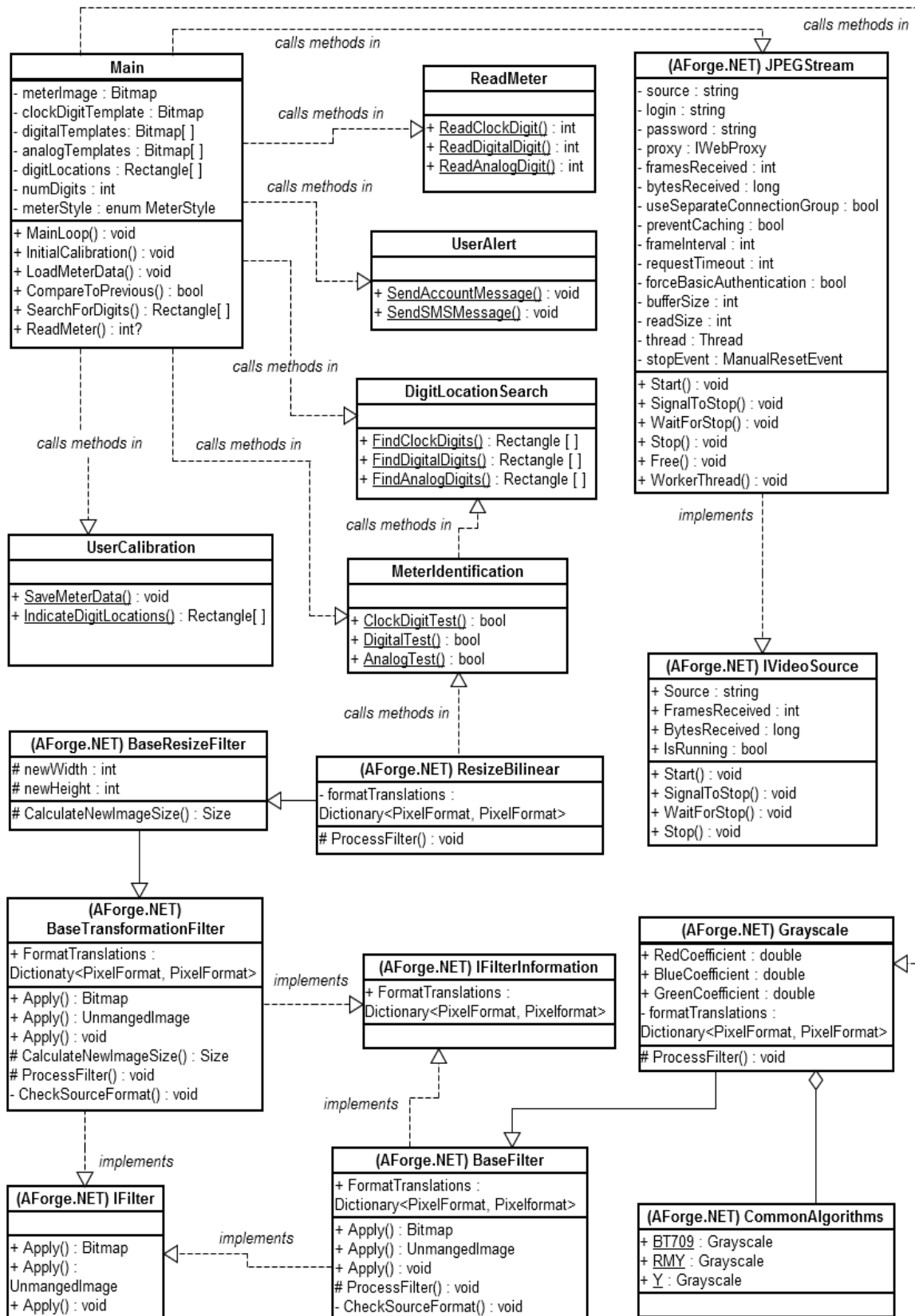
For the digital templates, this block of code will run for each template. It will be time consuming, but that is why this digit location search process will be avoided whenever possible. There will also be an added section after running through every template comparison but before returning the matching locations. This extra block of code will be for trying to ensure that the correct set of digits is chosen as the reading ones, in case there is more than one set on the meter display. Digit locations will be considered a part of a set if they are aligned horizontally and are of very similar size. The code will be as follows:

```
Rectangle[][] finalDigitLocations
foreach (location in digitLocations)
    if (finalDigitLocations is empty)
        finalDigitLocations[0].Add(location)
    else
```



```
foreach (location2[] in finalDigitLocations)
    if(if location is similar in size to location2[0] and its Y component is close
        to location2[0]'s Y component)
        location2.Add(location)
        break
```

Figure [K9] – Class diagram for the image processing software



For the analog digit templates, the method of comparing top and bottom halves of the template separately as mentioned in the design section makes the code even more complicated than the digital templates. The pseudo-code presented below goes in the section where the differences are added up:

```
diff += absolute value of...
if (m <= height of the template/2)
    diffTop += absolute value of...
else
    diffBottom += absolute value of...
```

Then after doing the whole template for a particular position:

```
if (diff < threshold)
    it's a match, add the area
else if (diffTop < thresholdTop and diffBottom >= thresholdBottom)
    top half is a match, meaning this is the bottom of a digit so only the top shows
    add the area as a digit location, extending it up above the top of the template
else if (diffBottom < thresholdBottom and diffTop >= thresholdTop)
    bottom half is a match, add the area, extending lower than the template
else
    not a match
```

After the whole image has been checked, a method similar to the digit set matching shown for the digital templates will be implemented, but rather than focusing on finding the biggest set once it compiles sets, it will look for the least aligned set. Digits will need to be somewhat aligned horizontally to be put into a set, but the extra digits that aren't part of the reading should be more lined up than the reading digits.

For the digital and analog rolling meters, the same kind of template matching algorithm will be used for reading the digits on the meter, but only applied to the saved digit locations rather than the entire image. The digit that gives the best match within a digit location, including half-template matches for the analog meter, will be the reading for that digit, as long as the match difference exceeds a threshold. For the clock-digit readings, the program will consider 10 sections of pixels around the center of a digit location. It will check for the darkest of these locations, and that one will represent the digit that the pointer is pointing to, the reading for that digit. The pseudo code for this is as follows:

```
foreach (location in digitLocations)
    int[10] sectionValues
    for (i from 0 to 10, the number of sections to check)
        int[i] += total pixel value of section[i]
    find the section with the lowest value, that is the darkest section
    the digit that this darkest section represents is the digit reading
```

Those few snippets of pseudo-code 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.

5 Testing

5.1 Outlet Monitor Testing

5.2 Light Control Testing

5.3 Door Lock Testing

5.4 Communication Testing

Testing procedure starts with the internet. Our first goal is to design the website. The website is at the core of most of the functionality of the system. It is also the least cost effective and time consuming part of the project. Implementation of the website will be simple and crucial and we must set priority to getting the It running. With the internet we want to test the server first for the webpage. Then we will begin building the user interface. Once we acquire all the necessary parts, we will test the ethernet shields connection to the internet. The arduino software will then be installed. The first test run will include the functionality of the ethernet.begin function from the arduino software library. After the buttons are set we will test the functionality between the website and the ethernet shield. We want to be able to verify a signal (on/off or 0/1) directly from the website to the microcontroller. Once we can verify a correct signal between the on and off button from the website to the hub, we will move on to Xbee communication. A test connection will be set up between the lock or outlet xbee shield and the xbee shield of the hub. We want to test the addressing of the xbee shield. Sending information from the xbee lock to the xbee hub and vise versa. This will allow us to test the data destination addressing paramaters(DH and DL) of the xbee shield. After we verify communication between the devices and the hub, we will test the digital signal between the outlet microcontroller to the hub and then the hub to the webpage.

This will then allow us to test the websites display of the outlet power output. Then we will test the opposite direction of communication, by sending a signal to the hub and then to the microcontroller on the lock.

Testing of the ip camera to the internet will also take place. Then the image processing. Then we will be testing how to display the value obtained from the image to the webpage. Once we obtain an integer value from the image

processing we can begin testing the user profile. After the user profiles are set we can test the sms messaging on the website.

With the completion of the websites functionality we can then move on to the android app.

5.5 Server/Hosting Testing

The Abyss Web Server software test will consist of installing the software on a group member's computer and making sure that it can be accessed from the web. The server connection can be tested as soon as the software is installed and the server is 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 has been tested for accessibility, we can add our own web page files in order to test that our ASP.NET and PHP code is working correctly. In order for PHP to work, a PHP add-on package must be installed, but this is provided right on the Abyss website and is only separate because some people just don't need PHP functionality, so they made it optional. Once we are sure that ASP.NET and PHP are able to work on the server as we need them to, the Abyss Web Server will be considered functional for our project, and any further testing should be done in conjunction with testing our web site and image processing software, which will be run on the server.

Dyn's DDNS service is very simple to test. We just need to create a hostname redirection service, have the redirection point from the provided free URL to the IP address of the computer that the server is being hosted on. If typing the URL into a web browser brings up whatever is on our server, then we know the service works. This functionality is simple, and requires no further testing.

5.6 SQL Database Testing

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

The first stage of testing is more for testing the table relationships, making sure that we can access the correct data with the given input. Based on Figure [K10] back in section 4.X, every piece of data should be accessible by specifying only an account id. For devices, rates and digit locations, while we can access all of the ones associated with an account from just an account id, we should be able to access a specific record by also passing in a detail for the one we want along with the account id. In order to make sure that this all works correctly, we can manually put data into the tables with insert statements. We can use various select queries to test that the correct data is accessed.

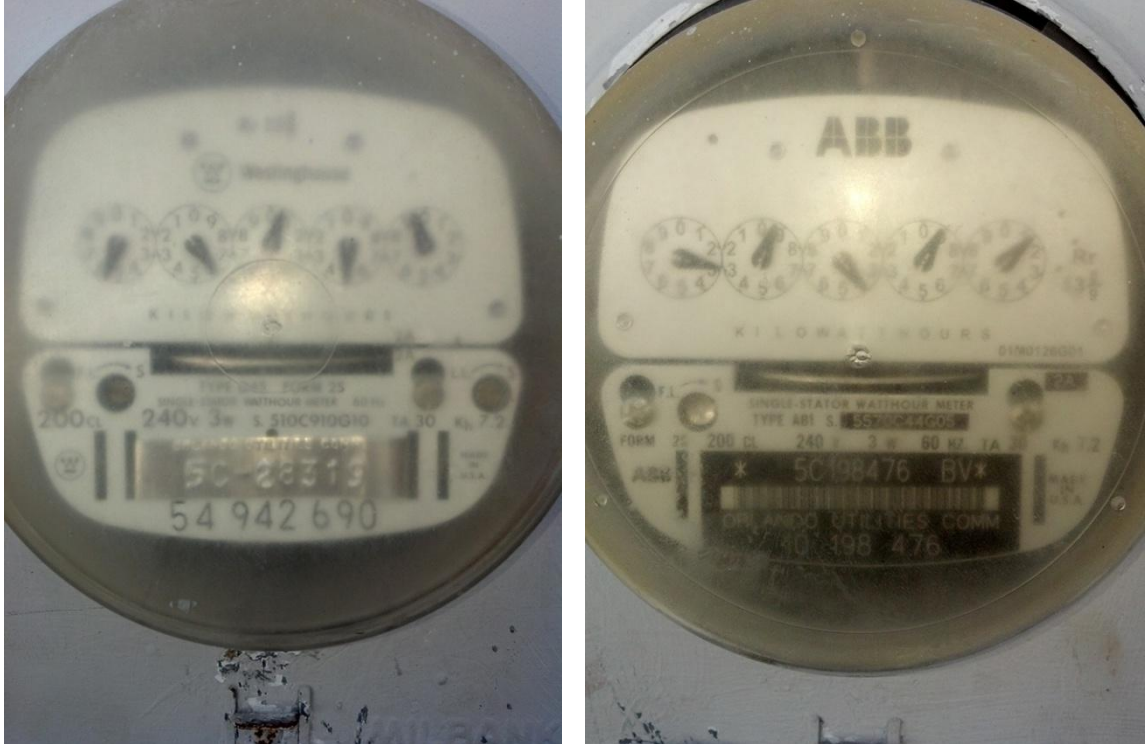
Testing the database from within the web site and the image processing software will be more focused on testing the connection with the database itself. We will need to make sure that any connections we make are correct and stable. We will test both insert statements and select queries from within the website and the image processing software. If the table reflects inserts correctly, and the website and software are receiving the proper data back, then the connections will be considered correct. If the website and software can run while using these connections for an extended period of time, then the connections will be considered stable.

5.7 Image Processing Software Testing

Testing the image processing software will involve 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, is set up. The video feed image capture functionality in particular cannot be tested without a video feed URL, although we may be able to find something online, but that functionality comes from the AForge.NET existing code, so it should not pose a problem. Similarly, the user alert and user calibration parts of the program will not be completely testable until the web site is up to provide an interface, and any data saving with SQL will need the SQL database to be available. However, the grayscale conversion, resizing, software calibration, digit location finding and digit reading functionality should be testable on their own, and the template matching 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 software calibration can be done using individual images passed in to the algorithm. We will use the images shown in Figure [K3] back in section 2.3, as well as others, some of which may offer more of a challenge for the software due to a less-than-perfectly-clear view of the meter details, such as the images shown in Figure [K11]. These same images can be used for the digit location search testing.

Figure [K11] – Some extra test images for software calibration and digit location search. These images should prove much more difficult to correctly analyze due to the opacity of the plastic casing. Digit locating will be the primary concern with images such as these



Any image locations discovered in the calibration step should be marked by red rectangles, as mentioned in section 3.3.2 and shown in Figure [K7]. This image can also be produced by the digit location function to ensure that it is working on its own, just for testing purposes at least, since it will not produce such an image during regular functioning.

The digit reading algorithms will be very straightforward for testing. We can just feed meter images with digit location indicators into the digit reading algorithms, and if they work successfully, they should give back a meter reading. Testing to make sure that comparison to earlier readings works correctly can be done by simply feeding values that represent database readings and then running the digit reading software, having it compare both to values that should produce discrepancies, and values that should not, and ensuring that it produces the correct result. Similarly, the check against a previous image to check for digit location changes can be done by simply providing images to the software that should both cause a recalibration or not cause one, and ensuring that the correct action is taken.

Once the website, camera feed and SQL database are all up and running, the components that depend on them can be tested. The website configuration interface software can be tested by accepting calibration settings on the website and then checking the database to make sure that they were stored correctly, and by manually entering calibration data on the website and again checking that it gets stored correctly. The camera feed can be tested, just to ensure that it is integrated properly with our software, by having the program grab images from the video feed and checking to make sure that the images are being grabbed

correctly. The SQL database code can be checked by having the software store some data, such as readings and digit locations, and ensuring that it makes it to the database correctly.

Once everything has been tested on its own, then the system can be 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 acceptance or asks for manual calibration. After that point, the website should begin to show real-time updates from the meter based on the data that the software is providing, which would mean that the software is working. The image location recalibration can be tested by moving the camera slightly and making sure that the software is able to continue, or asks for user input if necessary. If all of these functions are running as they should be, then that means that the software most likely works. This whole-system test will be performed on more than one running meter to determine if it really is working, for the sake of being thorough.

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 we want to minimize cost, we still need to make our project work, so the cost will still be considerable. We initially estimated that a \$100 contribution from each of us, for a total of \$400 available, would be enough to complete our project. At the very beginning, before any real research other than average prices for the parts we assumed we would need, we came up with an estimated cost of \$270 for everything. This estimate included a camera, a USB cable that we 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, we have found that we forgot to take development boards into consideration, and our estimated cost will almost certainly be exceeded by the end of the project. We may however still be able to stay under the total \$400 budget.

As of right now we have spent a total of \$245.30 on development boards, wireless adapters and an Ethernet board. The wireless adapters and Ethernet board will at least be used in our final project, so while the cost is high, it will at least go toward the final project. A summary of our budget and expenses so far is in Table [A1] below.

Table [A1] – Initial budget and spending so far

Part	Cost	Quantity	Total Cost
Arduino Duemilanove (Development board)	\$19.50	3	\$58.50

Ethernet Shield	\$59.95	1	\$59.95
Wireless Shield	\$14.95	3	\$44.85
Xbee Antenna	\$22.95	3	\$68.85
Shipping			\$13.15
Total			\$245.30
Budget			\$400

6.2 Milestones

When our group finally got started on our project, we decided to divvy up the sections of the project. While we would of course still communicate extensively to make sure that each of our parts would work together in the end, this approach allowed for each of us to focus on our 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, we initially planned out our schedule so that we should each write 5 pages per week, finishing in 6 weeks and giving us two more weeks to compile everything and possibly turn our paper in early. For Senior Design 2, we planned our schedule out by the week. This initial schedule is presented in Table [A2].

Table [A2] – Initial Weekly Schedule

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
12	8/26/2012	Receive PCB's; Begin Assembly
13	9/2/2012	Assembly
14	9/9/2012	Assembly/Troubleshooting
15	9/16/2012	Assembly/Troubleshooting
16	9/23/2012	Assembly/Troubleshooting
17	9/30/2012	Troubleshooting; Communication between Soft/Hard-ware
18	10/7/2012	Finish Up Design
19	10/14/2012	Finish Up Design
20	10/21/2012	Finish Up Design
21	10/28/2012	Finish Project
22	11/4/2012	Prepare Presentation
23	11/11/2012	Prepare Presentation
24	11/18/2012	Finish Presentation
25	11/25/2012	Practice!
26	11/30/2012	Show project/Presentation

Now that the Senior Design 1 paper is complete, we find that we were not necessarily able to meet the initial schedule, but rather finished the paper over the weekend before it was due. This still gave us the rest of the week to compile it, and has not been much of a problem.