**Section1: Introduction/Definition**

**1.1 Executive Summary**

At the Worlds Fair in 1893 Nicola Telsa captivated the United States when he lit up the streets of Chicago with the first public demonstration of alternating current. Since then the entire world had become dependent on AC power and it was without exaggeration to say we would not be a technology driven society without it. Electricity drove the industrial revolution just as it drives the economy today and if all electric power were to fail suddenly the consequences would be dramatic and life changing for all of us. If you take one minute to reflect on your daily life and everything that it consists of it was nearly impossible to find something that doesn’t use electricity or took electricity to make it. Electricity was one of our most precious and useful resources and we must do everything we could to safeguard it and make power failures as rare and isolated as possible.

It was the primary goal of this project to ensure that homeowners are unaffected by utility company power outages and that they are able to continue their day to day activities without interruption or inconvenience. With an upfront initial investment a homeowner could easily make their home immune to power failures and time spent in the dark. This could be done with the purchase of a home generator from any home improvement store and the components designed in our project. There are many sizes of generators and like anything else the bigger and more powerful the more expensive. It was up to the homeowner to decide what was in their budget and how much of their home’s electric circuitry they want to use during normal power failure. One of the best features of our project was it gave the homeowner the freedom to choose any size generator for their house while not having to change any aspect of our design.

In our project we designed a system used to monitor power during normal and emergency power operations to help save the homeowner money on their monthly power bill as well as maintaining a safely operating system. The project consisted of three key components: a power strip similar to the one you see computer equipment usually plug into, a device for measuring home generator power consumption, and also a touch screen monitor used for the user interface. The power strip contained voltage and current sensing circuits along with a microcontroller to calculate power consumption and quality. This processed information was sent wirelessly via RF communication back to an embedded touch screen microcontroller for display to the user. The main intent was for the homeowner to have multiple power strips and together they report back to the touch screen giving power consumption information for almost the entire house. The device used for measuring the generators power consumption contained essentially the same components as the power strip with minor modification, but looked a little different to fit the output configuration of the generator. The touch screen monitor would include a microcontroller and wireless RF communications for receiving and transmitting data.

In order for the user to interact with the system a touch screen was used. This touch screen had 3 inputs the user could manually enter. The first allowed the user to manually add additional power strips to the network. The second input also allowed the user to individually turn power strips on and off. Last the touch screen also allowed the user to input the KW rating of their home generator to be monitored as part of the system.

During times of receiving normal utility company power the system work as a basic power monitoring system. Any device plugged into the power strip had its power consumption and cost calculated and displayed on the touch screen. The feature aided the homeowner on monthly power bill reduction and just overall reduction in wasted power consumption. When normal utility power was lost and the home started operating on backup generator power the touch screen monitored power generation from the generator and power consumption from the home making sure of acceptable operating conditions. The touchscreen kept track of the average power usage of each device plugged into the power strip and aided the user in deciding how many appliances their home generator could handle at once. The whole point of monitoring the system in emergency power mode was to ensure generator failure due to overload was not an issue.

Many homeowners lose power for days or weeks each year, due to hurricanes and other natural disasters, which was why we believe something like this could be so useful if it were available. Not only did this design aid homeowners in time of distress, but it could also help them become more responsible with their bills and make them a little more environmentally friendly.

**1.2 Project Description**

The intentions of our project were to design a home energy management system that would be beneficial to the homeowner during normal power operating conditions and also during times of utility power loss where the home generator kicked on and took over. We designed power strip modules that resembled those that you would plug all of your computer and office equipment into at your desk. These power strip modules had voltage and current sensing circuits to calculate power consumption on individual appliances. This information was then sent back to a controller wirelessly that was then further processed to determine things such as instantaneous power consumption, daily power consumption, and total power consumption since last power bill. The controller was then communicated to a LCD touch screen monitor that gave the user access to the system. The user needed access to the system to add additional power strip modules to the system. The power strip modules also contained contactor circuits to allow the user to remotely turn on and off appliance through the LCD touch screen monitor.

The concept behind the entire project was as follows. During normal power operating conditions the homeowner may have used the LCD touch screen monitor to observe total power consumption, rate of power consumption, individual appliance power costs, and also projected monthly power bills. The user also had the ability to remotely switch on and off individual receptacles in the power strip modules to meet their monthly power consumption goals.

An affordable generator for home electrical power backup was typically unable to support the full load of the house. Thus, during emergency power operation   
(i.e. the whole house running on the store-bought generator) the user had the same options as they did during normal power operating conditions with the addition of the following features. Our project used current and potential transformers to calculate how much of the total available power the generator was producing and wirelessly sent this information back to the main controller for processing. During the initial transfer between normal utility power and generator power, the controller automatically switched off all loads connected to the power strip modules. From this point the user then used the LCD touch screen monitor to decide what appliances to power on. The controller then looked at how much power was currently being used by the homeowner compared with how much power was available from the generator and signaled the homeowner if they were about to overload the generator, which resulted in a shut down. This feature allowed the homeowner to use any appliance connected to one of our power strip modules while insuring the home generator was not overloaded. The controller was also programed with a safety feature that automatically shut off circuits during overload to maintain power in case the homeowner did not respond quickly enough.

**1.3 Motivation**

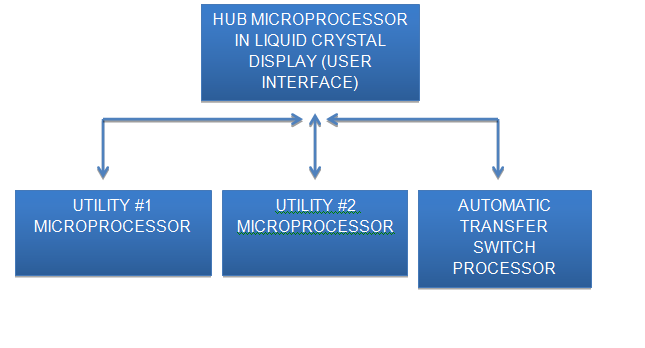
The motivation our group had for choosing this project arose from something that affects those of us living in Central Florida closely, but its applications and uses were not limited to this region. Each year, powerful tropical storms threaten the entire Eastern coast of the United States, and a large number of these hurricanes find themselves in Florida. These storms could leave hundreds of thousands of people in Central Florida alone without power, access to roads, and the ability to communicate over the internet for extended periods of time. In 2004, hurricanes Charley, Frances, and Jeanne arrived one after another in quick succession, leaving much of the region without power for weeks. Often, there was too high a demand on the limited number of workers available from the power companies to restore power in a timely manner. Thus, many homeowners decided to invest in generators to help power their homes during these times. However, most average people do not have the time to learn how exactly a generator works, and may continue with their normal routines once the alternate power source was hooked up and running. Unfortunately, this situation could lead to an overexertion and failure of the generator. For these reasons, we choose to design a home energy system which completely automated anything and everything about the generator to avoid this possible situation, while also giving the user an appropriate amount of control over the power supplied to their home. Not only did this protect the homeowner’s investment in the generator, but the system kicked into place as soon as the main source of power was lost (i.e. the incoming power from the utility company), thus potentially saving the user weeks of time without power waiting for it to be restored.

**1.4 Goals and Objectives**

The intent of our project was to design a home energy management system. This system would be versatile in that was beneficial to the homeowner during both normal power operating conditions as well as during times of utility power. Our definition of “normal power operating conditions” was when the homeowner was connected to a power source from the utility company. “Utility power” was defined by the entire house running off of a generator which alone did not provide an amount of power equal to what would be used from the power company on a daily basis. In fact, the utility power should have been an amount that would have severely limited the abilities of the homeowner to power the appliances in their home. This was an important distinction because much of our project was focused on the sometimes difficult choices that would need to be made during times of power outage. Often, the average homeowner was unaware of the many details and tremendous amount of power that go into allowing their lives to run as normal. Questions like, “Can I run my hot water heater on a generator?”, “How much power do I need to keep the refrigerator on?”, or “What are my limitations for powering my computer/television/modem during power failure?” were simplified and answered in our design, while still giving the end-user as much control as possible.

The main design of our project began with the loss of normal power, at which point the generator already connected to the user’s home turned on in response to a signal from a microprocessor. That microprocessor was constantly monitoring the current coming from the main, “normal”, power source; as soon as the current disappears, a signal was sent to the generator to turn on and take over power for the house. Our design was a power strip module that resembled the surge protector power strips which could be used at one’s computer desk to plug in your computer and its peripherals. It had two or three outlets to plug in any standard utilities. These power strip modules had voltage and current sensing circuits which would be constantly taking measurements related to power consumption. The power strip modules also contained contactor circuits to allow the user to remotely turn on and off the appliances through the LCD touch screen monitor. This information (current and voltage readings) was then processed at a microcontroller (embedded in the power strip module), and then transmitted wirelessly to the main microprocessor in the LCD.

The wireless communications in our system were done using a single protocol, which was yet to be determined. Shown below in Figure 1, each wireless device was a transceiver (i.e. had the ability to transmit and receive data); they did not communicate with each other, but only to and from the central microprocessor (in the LCD). The wireless transceivers were used over a short range in our prototype; this could have been extended to the end-user as well, assuming the main processor in the end product was located near the center of the house. This aspect would allow for lower power usage, as well as lower costs for the transceivers.



**Figure 1: Wireless communications block diagram**

The LCD contained the main “hub” microprocessor which collected data from the power strip modules and was where the bulk of the calculations and decisions were made. This display was the interface between the user and the software which controlled whether or not outlets in the power strip module were supplied with power. During normal operating conditions, the user was able to control which outlets were turned on (supplied with power) and also have the ability to turn off any or all of the outlets. There was also a section on the LCD which provided the user with pertinent data about their power consumption. In this section, an emphasis was placed on “useful” data; that is, we didn’t wish to give overwhelming amounts of graphs and numbers which may have required knowledge of math to interpret and understand. The LCD showed information in as simple a form as possible, while still providing important, useful data. The data shown was facts such as:

* Instantaneous power consumption
* Individual appliance power costs
* Daily power consumption
* Current rate of power consumption
* Total power used since the last power bill

An emphasis was placed on showing as many dollar amounts as possible, so the user could more easily relate to the information on the screen. Abstract figures, graphs, and even most other data was shown for the purposes of the demonstrations, and on a limited basis at that. This process was intended to help the user to more accurately monitor control their power usage, in order to meet their monthly power consumption goals.

**1.5 Requirements and Specifications**

The design utilized an automatic transfer switch to transfer power to a secondary power supply in case the primary source fails. The automatic transfer switch was a third party device. It only needs to be tested for functionality. For this project the use of a generator to provide secondary power in case of an outage was intended. After power had been transferred the devices were able to read the load from the generator and communicate this information via wireless signal to a main microcontroller.

Other outlets in the home also had power monitoring features. They had the ability to both transmit and receive instructions or data. Each outlet was required to read the current that was being drawn by that specific outlet. That information was then relayed to the main microcontroller which summed all of the power being drawn from the home at that time. If too much power was being drawn with respect to the generator an interrupt signal was sent to any of the microcontrollers that were in control of an appliance. These microcontrollers stopped the use of the appliance immediately upon request.

Each microcontroller measured a specific current. The accuracy of these currents was precise enough to prevent damage to any appliances or the generator. We set our initial goal to reach a precision of 1mA. The main unit received power reading information from all the other microcontrollers every second. The main unit provided absolute instructions to all other microcontrollers. It must have the ability to stop and resume power consumption of any outlet in the home. These criteria were set by the user through an LCD input screen. In case of an overload of the secondary power source appliances shut down with respect to their priority.

Outlets were sorted by priority based on user input. Each outlet priority level had a number ranging from 1-10 with 10 being the most important. In the event two outlets are given the same number the user was asked which appliance was more important. Here the user could decide if they would rather have warm water or air conditioning. All of the priority level inputs were given to the system by the user through the LCD panel. In case no priority was set the appliance draining the largest amount of electricity was shut down.

The power strips were able to achieve these specifications as part of our design:

* Send data every 5 seconds
* Enter sleep mode for 5 seconds in between packet transmissions.
* Measure currents up to 15 amps
* Measure voltages of 240V
* Up to 2 appliances could be connected to any power strip
* Have a line of sight range of at least 100 feet in an indoor environment
* Have a range of at least 50 feet between two rooms
* Power readings were measured within an error margin of 5%.
* Had an ADC that was 10 bits wide for 1024 steps of resolution

The Main control station consisted of a controller that received information from all the power strips and displays this information on the user controlled LCD. This station was able to send and receive information from the LCD. If the LCD relayed information to shut down a specific appliance an interrupt signal was sent to relay that information.

The specifications for this unit were as follows:

* Receive data from 3 different power strips 5 seconds
* Receive data from power metering device two times per second
* Communicate this data to the LCD module 3 seconds
* Had an ADC that was 10 bits wide for 1024 steps of resolution
* Receive and transmit information with an accuracy of 3 decimal places with the local microcontroller.

The LCD was driven by its own processor and was able to communicate with the microcontrollers. Through the LCD screen the user had the ability to read power readings from any appliance in his home as well as the readings from all of his appliances at once. The user was able to set priority levels for each of these outlets. In addition the user had control for turning on and off each outlet individually. If the user was trying to activate an outlet that would overload the generator then they were given a warning. The LCD and the main control station were in constant communication.

The LCD screen was required to do the following:

* Receive and transmit information with an accuracy of 3 decimal places with the main microcontroller.

While an outlet was not being used it entered a sleep mode to preserve the battery life of that specific unit. The microcontroller remained in sleep mode as long as there was no signal telling it to turn on. Once the microcontroller received the signal to turn on it returned to normal operation.

The last power metering device was the one reading the power information being delivered by the generator. This device relayed the information to the main microcontroller. All of the overloading decisions of this project were dependent upon this part. This device was able to read the power load on a generator by reading voltage and current information.

The generator power metering device was required to do the following:

* Read power supply information from the generator twice a second
* Sent the power consumption information to the main microcontroller twice a second
* Must have an accuracy of 3 decimal places
* Measure currents up to 30 amps
* Measure voltages of 240V

**1.6 Roles and Responsibilities**

Our design group consisted of four senior electrical engineering students. The members included Chris Diller, Christian Aranha, Kurt Riecken, and Arman Murat. We decided to split the research and design into four main parts where each group member was given an equal amount of work. Each of us were working on separate parts of the project, but at each group meeting we explaind to each group member what we had accomplished and how it worked. We believed it was very important that each group member was completely competent in understanding the entire design of the project.

Chris Diller had taken on the role of researching and designing the user interface of the system. Chris was responsible for researching the different kinds of touch screen monitors available today and the pros and cons each. He was also in charge of selecting the main microcontroller to power the touch screen monitor and wireless RF communications between power strip modules. Chris was designing appearance of the GUI interface and was also writing the code to implement it. Last he also wrote the code for the microcontroller receiving and transmitting data from the power strip modules and made sure that information was displayed neatly and accurately to the touch screen monitor.

Kurt Riecken had taken on the role of researching the microcontroller needed for the power measurement and calculations. Kurt was determining whether a standard microcontroller such as the Aurduino was needed or maybe something more application specific. He was also in charge of processing the signal received from the voltage and current sensor and making the appropriate calculations for power measurements. He then wrote the code for the microcontroller to receive and transmit data from the touch screen display Chris was designing.

Arman Murat took on the role of researching and designing the circuits and sensors needed for power measurement. Arman was researching the available technologies for current and voltage monitoring and determining what fit the project best. Arman and Kurt were working closely together to ensure compatibility of sensors and microcontrollers and any additional circuits needed to connect the two and make them operational.

Christian Aranha took on the role of researching and designing the wireless communications between the major components in the project. Christian was researching the latest technologies in wireless communications and determined what technology worked best for our project. Christian was also in charge of researching the analog to digital and digital to analog conversions needed for transmitting and receiving the data from the microcontrollers. We also considered creating our own wireless protocol for transmitting our data back to the touch screen and Christian also did the research on this.

Additionally we had a small amount of our project that required minimal design but was required for demonstrating purposes to the class. This part of the project included researching automatic transfer switches, circuit breakers, and receptacle circuits. Team members contributed their ideas on how the project should be demonstrated and what it took to do so.

**Section 2: Research**

**2.1 Similar Projects and Commercial Product**

After the decision was made about what our project would be as a part of our research we went through some other projects presented by other students. In our research through different colleges senior design projects, we all agreed that UCF projects were better in countless aspects and worth to review in the “similar projects” section in our paper.

The first project that was similar to ours was from Fall06-Spring07 done by Group 24 with the project name H.E.M.S. (Home Energy Monitoring System). Just like our project’s goal this group intended to save money to the user by conserving power. Their system basically had the same features just like ours such as measuring the power consumption at the current moment or overall through the whole month, transmission of this information wirelessly to the main processor and the user interface, LCD display for user interaction. Although both projects were trying to achieve the same goal, our senior design project had additional features on top of the mentioned above. One difference they had in their design was installing the power measurement components to the gate where power entered the house; the circuit panel. This made many things easier and less complicated since the whole power consumption could be read from the circuit panel. Since our projects had different features (i.e. in case of a power shortage our designed system automatically started the generator but not only started it but distributed the power produced by it according to the importance level of the outlets) this way limited generator power would be used efficiently. In order to achieve this system we chose to install the power measuring components into individual power strips which would go to every individual outlet in the sample demonstration house. In our design project we were going to have the ability to turn on and off several outlets from the LCD control panel. If we were to summarize the differences between two projects our design would be able to do the home energy monitoring and in addition to that we would be able to do emergency power monitoring and management as well.

The second project that was somewhat similar to our project was from Fall06-Spring07 done by Group 3 with the project name Airpax Smart House. This project was not about power management but it had some features that save power such as motion sensors that turn lights on and off when a person enters a room or leaves. Similar features they had were the wireless communication methods and LCD display control but basically this project was a smart house computer with a couple of power saving features but not a power management system.

The third project was another power management project Fall07-Spring08 done by Group 17. The project’s name was Occulight. The group members’ goal was to reduce power consumption by placing motion sensors into each room and adjusting the power consumption in each room by their occupancy. This project was very much like the previous project but addition to turning lights on and off this project want to take it a step further and be able to switch power outlets on and off according to the rooms occupancy and power outlets importance level. Their choice of communication was wireless the same as our projects but again our projects had different features such as in case of a power shortage our designed system would automatically kick the generator in but not only leave with that but distribute the power produced by it according to the importance level of the outlets in order not to overwhelm the generator.

Next project was again somewhat similar to our project was from Fall07-Spring08 done by Group 18 with the project name Detect All 3000. This project was not about power management, it was an emergency fire and hazardous gas detection system. The similar characteristic of this project to ours were its sensor’s interaction with the main board and its LCD control set-up.

The fifth project that caught our attention as a similar project from UCF senior design data base was from Spring08-Summer08 done by Group 1 and the project’s name was Power Monitoring System. Similar to our project’s goal this group’s project was to save money to the user by power conservation. The design project basically had the same features as ours such as measuring the power consumption at the current moment or overall through the whole month, transmission of this information wirelessly to the main processor and the user interface and LCD display for user interaction. There were many differences between the two projects as well. Our senior design project had additional features than theirs. Our project had an emergency feature: in case of a power shortage our designed system would automatically start the generator but not only started it but distributed the power produced by it according to the importance level of the outlets, this way limited generator power wouldn’t be overwhelmed. In our design project we had have the ability to turn on and off several outlets from the LCD control panel; the outlets were at the same time also power measuring units. While we had additional features, this group’s design had different features than ours as well. Their profiling system for major appliances in order to turn them off for the time the house hold was empty justified itself as a big energy saver.

The next project which was again somewhat similar to our project was from Spring10-Summer10 done by Group 2, with the project name Wireless Power Meter. This project was almost the exact same as ours again. This group did have a power strip power measurement device and a wireless protocol to transfer data to the main control device. They did have a set accuracy requirement of at least 5%. Again they lacked the emergency power management feature that our project had where smart generator power management was achieved in case of a power shortage occurring.

Another project similar to ours was from Summer07-Spring07 done by Group 6: the project’s name was Taj’s House. This project again tried to save energy and cut the power bill for the user but the energy management was achieved not automatically but remotely through a Bluetooth device. The idea was to be able to turn on and off the light switches and thermostat using a Bluetooth device this way if the user wouldn’t be home for a long time energy consumption was reduced by turning these switches off. [23]

Our last example came from Fall10-Spring11 group 13. The project name was Home Energy Management System. Their project was very similar to ours and a couple of other projects above where they had a power strip power measurement device and a wireless protocol to transfer data to the main control LCD device. Our emergency power management feature separated our project from theirs but everything else was very much alike except our choice of microcontroller and wireless communication device choice.

After the review of some similar projects we wanted to review some similar products that did the same thing as well as available for purchase in the market. The first one that drew attention was called Kill-a-Watt produced by P3 International. There were a variety of products developed and produced by the corporation and basically these components were more or less what we wanted to achieve with our project except the smart emergency power solution feature we have in addition to this power saving and monitoring system. Below is the generic Kill a Watt power strip shown and the other model that was more similar to our project since there was wireless communication to an LCD screen where output was displayed and user interaction was possible. Unlike our design, there was no switching on and off feature of the power strip available from this LCD but an impressive 0.2% accuracy was achieved. In figure 2 below the wireless module and the display screen are shown. [23]

**Figure 2: P4225 Kill A Watt®**

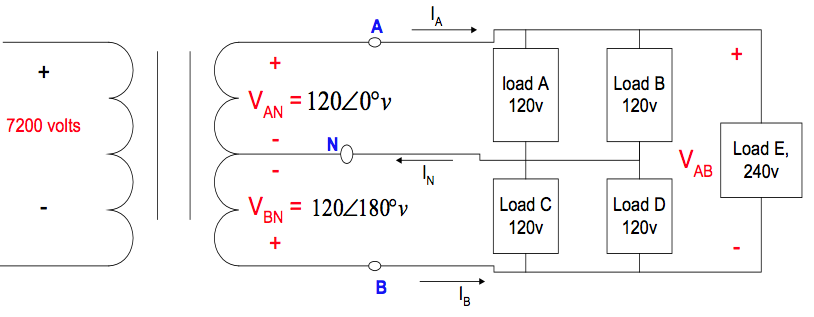
**Permission requested from www.p3international.com**

**2.2 Home Electric Service**

The two primary forms of electrical distribution here in the United States were single-phase and three-phase power. Single-phase power was used for all residential homes and very small business offices. Three-phase power was used in commercial and industrially applications and was not considered in this project, as our design would have had to be completely modified to account for it.

Single-phase power was a single alternating power source that was sent out on a conductor that measured a voltage potential between it and a ground wire. This was a two-wire system where current flowed from the source through the load and then back through the ground wire. Because it was an alternating source each 180 degrees the current reversed direction and headed in the other direction. In this type of system, only one voltage could be obtained for utilization and that voltage was dependent on the physical characteristics of the generator.

A residential home had three conductors coming to it from the power company. Two of them were “hot” conductors and the other was a “neutral” conductor. The hot conductors measured 120 volts between them and the neutral conductor and they measured 240 volts between each other. How could this be possible if the service were a single-phase system? The power company really was supplying the house with a single-phase system, but the transformer supplying the house was playing a little trick to do so. Figure 3 shows a diagram of the utility transformer and the conductors on the input and the output.



**Figure 3: Home Utility Power**

As you could see from the figure the power company’s wires came in on the left and only included 2 wires. A voltage of 7200 V between the hot and ground conductor was most commonly used on the low power poles located in residential areas. On the load side of the transfer were the “A” and “B” phases, which were by definition the outputs of the conventional transformer. In the middle of the secondary winding of the transformer was a center tap and created a point of equipotential for the “A” and “B” phases. Through simple circuit analysis it could be shown that the potential between phases “A” and “B” was 240 volts while the potential between phases “A” or “B” and the neutral conductor was 120 volts.

The reason for needing different voltages in a house was to serve the different appliances in it. Large equipment such as ranges, air conditioners, clothes driers, and hot water heaters were all 240-volt equipment because it allowed for them to remain a reasonable size and have a more efficient design. Every appliance surely could be designed to operate at 120 volts, but appliances designed at higher voltages could operate with a low current and therefore reducing the losses of the appliance.

The first prototype of this project was designed around the appliances that operated at 120 volts and if time and budget permitted, the design could have easily incorporated the appliances that operate at 240 volts with minimum modifications.

**2.3 LCD Touch screen Displays**

A touch screen device was chosen for user interface for the simplicity of use for the user. This was the only part of our project with which the user interfaced so to ensure user satisfaction we wanted to choose a user-friendly interface that anyone from a child to an elderly person could operate with no training making sure not to aggravate them. It was of the utmost importance of this project that the user would embrace the power and usefulness of the product because of their understanding and eagerness to use it.

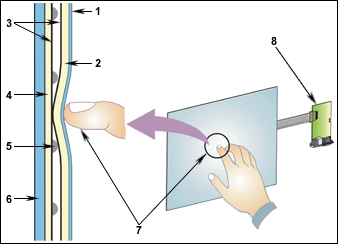
There were many types of different touch screens on the market, but the 4 most commonly encountered and used included resistive, surface wave, capacitive, and infrared. Each type of screen had its advantages and disadvantages as well as what application for which it was best suited. For this project we considered the fact that the touch screen would be located in a household and mounted on a wall near the thermostat. We also considered that this touch screen would be in use for many years and would have many dirty fingers touching the screen throughout the years. It was therefore important that during the final selection of a touch screen for this project that we selected one that would withstand the test of time and that the screen would have some kind of protectant to combat greasy smear from dirty fingers or that it would be easily cleanable.

Our touchscreen controller also incorporated an embedded microcontroller used for receiving and processing data from the various power strip modules throughout the house. When selecting the embedded microcontroller we considered the rate of transmissions the controller could output and receive, the amount of memory available for programs and data, and also the computer language the embedded controller used for programing. We planned on being able to store one month’s worth of data for each power strip module to calculate totals, power consumption, and monthly costs of individual appliances. The embedded microcontroller would only require a few analog and digital I/O because a wireless transceiver would be the only device connected to it. The rest of the design lay within the power strip modules and it was here where all power measurements would be taken and transmitted back to the touch screen controller.

In the next few sections a brief overview of each of the four types of screens were explained. Through comparison of the advantages and disadvantages of each of the types of touch screens we were able to decide which satisfied the requirements of this project the best. Even though we may have found a touch screen that fit our application perfectly, another may have been selected due to the cost constraints of this project.

**2.3.1 Resistive Touch Screens**

Resistive touch screens were some of the most cost effective and widely used touch screens on the market. They were extremely durable and were less prone to contaminates than acoustic wave touchscreens. Resistive touchscreens were also less sensitive to the effects of scratches like capacitive touchscreens were. The anatomy of a touchscreen was very simple and was one of the major factors of their durability. The monitor was made up of two sheets, which were separated by insulating dots. The top layer was flexible and was what the user pressed their finger against while the bottom layer was rigid and was what gave the display its firmness. Each of the sheets’ inside layer was coated with a transparent metal-oxide, which in turn gave the sheets their electrical characteristics. A voltage was then applied to the layers creating a gradient across them. When a user pressed anywhere on the screen the two layers made a contact at the point closing the circuit allowing the coordinates to be located. Figure 4 shown below represented the structure of the resistive touchscreen monitor as well as a list of its various components and where they were located. [3]



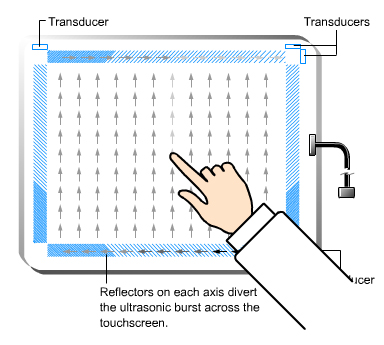
**Figure 4: Resistive Touch Screen**

**Permission requested from Fast Point Technologies**

1. Polyester Film
2. Upper Resistive Circuit Layer
3. Conductive ITO (Transparent Metal Coating)
4. Lower Resistive Circuit Layer
5. Insulating Dots
6. Glass/Acrylic Substrate
7. Touching the overlay surface causes the (2) Upper Resistive Circuit Layer to contact the (4) Lower Resistive Circuit Layer, producing a circuit switch from the activated area. The touch screen controller gets the alternating voltages between the (7) two circuit layers and converts them into the digital X and Y coordinates of the activated area.

**2.3.2 Surface Wave Touch Screens**

Surface wave touch screens were devices that utilized ultrasonic waves to process inputs from the screen. The display in figure 5 had a glass layer that had transducers on both sides used as receivers and transmitters. These transducers were placed on the X and Y axes and were used to create a coordinate system. An ultrasonic wave was generated and sent across from the transmitter to the receiver. The screen’s glass layer had reflectors on it, that when pressed, the energy of the wave created was reduced and the screen registered a hit giving its location. A negative characteristic of the surface wave touchscreen was it’s influence to bad performance from external forces such as dust, water, loud sounds, etc. Anything that could distort the acoustic wave would potentially make the device fail. [6]

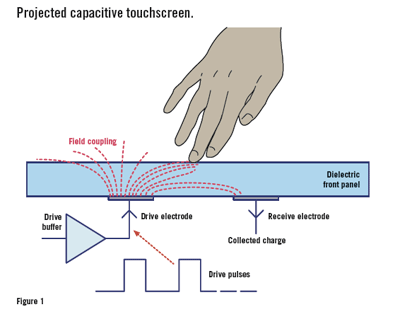


**Figure 5: Surface Wave Touch Screen**

**Permission requested from itechcompany.com**

**2.3.3 Capacitive Touch Screens**

Another type of touch screen was the capacitive touch screen. There were several different categories in which you could classify a capacitive touch screen and they included surface, projected, mutual, and self-capacitive. Each of these technologies differed from the others, but the fundamentals of how they worked remained the same for all of them. A capacitive touch screen consisted of an insulating material such as glass, which was coated with a transparent conductor like indium tin oxide (ITO). A human finger was also an electrical conductor and when that finger pressed against the screen a change in capacitance between the two charges could be calculated and when sent back to a processor this could pinpoint the location of where the finger tapped the screen. Capacitive touch screens as seen in figure 6 had a considerable disadvantage to resistive touch screens in the sense that they must have a conductor such as metal or a human finger come in direct contact with them to select a location on the screen. This was an undesirable trait for consumer electronic devices where in cold temperatures a user might be wearing gloves or a woman with long fingernails might have a hard time getting the screen to operate correctly. Capacitive screens were in general more responsive than resistive screens, but were more costly and were impractical for this project. [

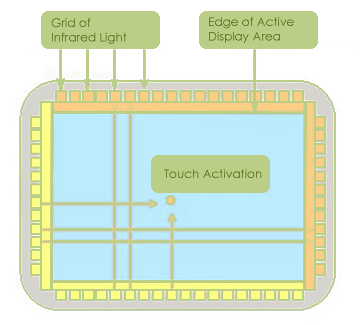


**Figure 6: Capacitive Touch Screen**

**Permission requested from eetimes.com**

**2.3.4 Infrared Touch Screens**

The infrared touchscreens used a simple design approach to deliver great accuracy and response. The screen used a grid of infrared LED’s and photo detector pairs in an X and Y coordinate system around the outer edge of the screen. These photo detectors were used to detect interruptions of the LED’s resulting from a user pressing on the screen. The accuracy of location was dependent on the number of LED’s and photo detectors along the X-axis and Y-axis. The more pairs the greater the accuracy. When the screen was pressed a photo detector triggered a hit on the two axes thus giving the coordinate location of the hit. Figure 7 below illustrated the yellow bars on the left and bottom as the transceivers and the orange bars on the top and right as the receivers. You could see how the distortion of someone’s finger pressing against the screen would allow the location to be calculated. Infrared touchscreen monitors were very durable and were used for industrial and military application so we knew they would uphold to the standards of our project. However, the infrared touchscreens were rather pricey and therefore would not be applicable to the cost constraints of this project.



**Figure 7: Infrared Touch Screen**

**Permission requested from ATouch Technologies**

**2.4 Power Monitoring Circuits/Instrumentation**

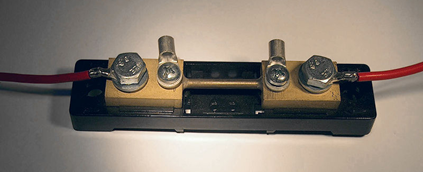
Since the main goals of the project were to measure power and, if necessary, restrict the usage of this power, the most imperative part of the project was to measure the power correctly. This correct measurement of power was going to reflect to the correct outcome and calculation of the energy cost. Appropriate energy distribution happened to be another feature of our project. In case of a power shortage the generator should kick in and limited power produced by the generator would be distributed according to the power consumption of critically important outlets without causing any overage of the generator output.

Our first question was which power we were measuring. Energy was distributed in AC form for the reason of this form traveling longer distances and it was less complicated to produce the energy in this form rather than brushless rectifier generators. After the production of this energy was transferred with high voltage lines with the least current possible to reduce the energy loss, voltage dropped down in transfer locations which were big transformers and finally were received in a regular house outlet 120 VAC which varied with 12 to 15 amps of max current drown. In a regular home there were different appliances with different uses most of which were resistive loads since they were used somehow to produce heat. Besides resistive loads we had inductive and capacitive loads as well which had outcomes on the magnetic field when connected to the current. The outcomes of these two different kinds of loads gave different types of power which were “Real Power” and “Reactive Power”. Our concern was with the Real power since that was the one that we were charged for so our calculations and measurements would be based on Real Power.

The types of power could be listed as Real power which was measured in Watts ( P = V \* I \* cosƟ ), Reactive power which was measured in VAR’s ( Q = V \* I \*sinƟ ) and Apparent power which was measured in VA ( S = P + jQ ). Since the power company charged us only on Real power, our power measuring circuit had to be measuring the real power. This brought up the importance of power factor in other words cosine of “Theta”. This was the angle between both the voltage and current RMS values and between the Apparent power and Real power. According to the previously mentioned formula “P = V \* I \* cosƟ” where Theta was the phase angle all we needed to do was multiply the values we got from the measuring circuit with cosine Theta and we found the power that we were going to be charged for which shouldn’t be a great deal to achieve. The real problem here was how we calculated Theta from the voltage and current measuring circuits. Therefore, to measure the “Theta” or the power factor which was cosine of theta all we had to do was read the time difference between zero crossings of the waveform of both voltage and current waves. This difference was proportional to the whole cycle period. If we were to accept the whole cycle as 360°; comparison of the time difference between zero crossings of the waveform of both voltage and current waves divided by the period gave the ratio of phase angle Ɵ divided by 360°. Of course all these calculations would be done by our microcontroller “MSP 430”. The important point here was the implementation of the current and voltage measuring circuits to the microcontroller since there was an allowed range of inputs to MSP 430. This was an easy problem to solve by implementing a voltage divider circuit, to both current and voltage measuring components, to reduce the voltage to make it compatible with the microcontrollers input values. The reduced factor in the voltage divider was important since later on in the representation of the power this factor was included otherwise the power consumption would have turned out wrong. [2]

After all we discussed about how we were going to measure the power it was time to cover the components we would use for the actual measuring of voltage and the current. Among numerous components that were suitable for the job we would make an educated decision and try to pick the one with the best accuracy, price and circuit compatibility. If this assignment was given in our first circuits class we would have came up with a solution that involves ohms law.

If we were to attach a resistor to the conducting wire where we wanted to measure the current and measure the voltage drop over the resistor we would have known the current. The real life application of this procedure among current measuring methods; the simplest one was the current shunt shown in figure 8.

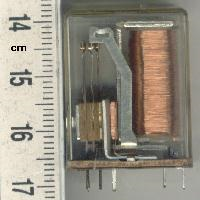


**Figure 8: Current Shunt**

Although we briefly talked about the current shunt here we would not consider it to be in our circuit. This was for several reasons. First one was the heat problem. A resistor when contacted a current in a conducting wire produced voltage and this means at the same time there was a power dissipation. The power in resistors was in the form of heat. In measurement units in watts, which could be interpreted as the more current the more heat. Heat was something we didn’t want in our circuit since it could harm everything around it starting from the shunt itself. This brought the second reason why we couldn’t use the current shunt in our circuit. As mentioned before the project’s main goal was to measure power with accuracy. Heat would tend to change the resistance so the measurements wouldn’t be as accurate with the rise of the current since the resistance would be a different value. Although shunts with a dependable resistance drifts were available these were way too pricy and still not the best choice since the size of these shunts were another inconvenience. This brought us to our real evaluation of current measuring components:

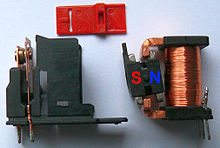
**2.4.1 Power Relays**

Our system design was all about managing power and this required for us to be able to turn on/off an outlet, which consumed too much power. In order to this we need a switch, which we could control remotely and this switch was the power relay. The relay in figure 9 was an electromagnet which moved an armature part attached to a spring. When the electromagnet was powered the magnetized metal part attached to the armature moves and either connects or disconnects the circuit according to the design or what was tried to be achieved. When the power for the electromagnet was turned off spring did the mechanical job to do the opposite of what was achieved by the electromagnet.



**Figure 9: Relay**

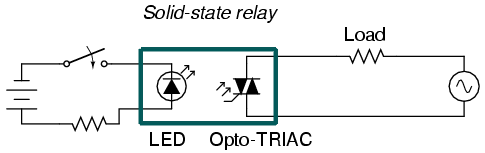
In this initial research phase we looked at many different applications of relays and saw that we could initially group them into two major groups; electromechanical and solid state. The basic principal of both applications were the same but we saw that solid state relays would be way more beneficial for our cause. There were many reasons for this selection that should be mentioned. In the electromechanical version one problem might be caused by the use of direct current and the spike problem. A voltage spike was dangerous for a semiconductor and it was a big probability for the coil to have a voltage spike when the power was switched off and this problem was eliminated with a diode where this time there might have been different issues with AC power. Alternatively, a parallel capacitor and a resistor could solve this same problem talked above. Generally speaking, an electromechanical relay such as a latching relay in figure 10 would be a perfect candidate for our project since it would serve the project’s energy saving purposes since its energy consumption was almost zero. The way a latching relay worked was a little different than the conventional simple relay described above. Here there were two coils instead of one and a permanent magnet that kept the armature arm remaining in the position it was located after the pulse, so there was no continuous current needed to keep the electromagnet charged to keep the armature in the same position. Individual pulses to each coil would give either one or zero so that the switch operation would be achieved. Another advantage of the latching relay would be its low resistance values which would mean low power consumption and less heating. Price-wise it was considerably cheaper compared to other candidates. Size was not as bad neither which was very important for the design of the strip module we were designing; though, compared to the solid state relay it was twice as big. Its power handling capabilities could be phenomenal. For the “Contactor” type of electromechanical power relays high loads such as 10 to 50 amperes could be achieved but this type again came with a time delay in order to secure the equipment. Beside the size comparison the real reason we decided not to choose latching relay was the complicated design process. Electromechanical relays needed separate wiring and a separate control algorithm.



**Figure 10: Latching Relay**

In our search for relay we didn’t even consider reed, mercury-wetted, polarized and contactor relays. The first two; reed and mercury-wetted relays were almost the same thing with the difference of the first one had its contacts in a vacuum and the second dipped in mercury for the reason to stop the corrosion. They were both too big and for small voltage usages. The polarized relay was too big as well and it was used for high precision applications, which was not what we needed for the relay part of our project. As mentioned before a contactor relay was a high voltage switch. It could handle high amperages but, since a home outlet had 12 to 15 amps and the noisy operation of this relay was a good reason for us not to consider it.

After reviewing all the pros and cons of the electromechanical relays it was time to take a look at SSR, in other words Solid State Relays. Since in solid state relays displayed had no electromagnet that part may be replaced by a LED and an Opto-TRIAC as shown in figure 11 below and everything worked flawlessly.



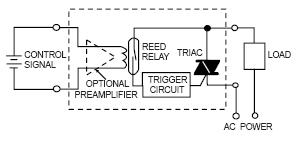
**Figure 11: Photo Coupled SSR**

**Permission requested from** [**http://www.pc-control.co.uk/relays.htm**](http://www.pc-control.co.uk/relays.htm)

Solid State Relays were semiconductor based switches and unlike electro mechanical relays, they didn’t have moving parts which meant they didn’t need maintenance unlike changing of the coil was had to be done in contactor relays. This made them very reliable with their lifetime. As mentioned above, an electromechanical relay, such as a latching relay would have been a perfect candidate for our project since it served the project’s energy saving purposes because its energy consumption was almost zero. Its low resistance values, not only meant low power consumption but fewer heating problems as well. Pricewise it was considerably cheaper compared to other candidate the SSR. So what made Solid-State relay a better choice?

An LED Opto-TRIAC was the type that came into mind when an SSR was mentioned, since it was the most popular of all. Instead of the electromagnet, a light emitting diode (LED) turned on with the application of power to the general purpose relay and shone through a gap where an Opto-TRIAC was placed on the other side. The TRIAC was a small semiconductor switch which was very similar to a diode. TRIAC would be used by itself and would be a very inexpensive choice with good switching power consumption for resistive loads, but it was not the same for reactive loads since it required additional circuitry for applications with reactive loads. An Opto-TRIAC was almost the same component described above but this time the response of it was against light flashing across the gap coming from the LED. This response opened and closed the circuit that it controlled. Solid State Relays were known for their quick responses and their vulnerability to electric noises, high price, high series resistance however way more sensitive output in terms of rating compared to the latched relay switch and their dependable long life and having no moving parts made up for their drawbacks.

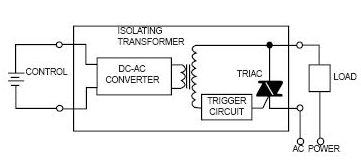
Just like the electromechanical relays, SSR’s had different types as well. The most common being the Photo-coupled SSR or as it was mentioned above as LED Opto-TRIAC SSRs, followed by the isolated Reed-Relay coupled SSR shown in figure 12 or in other words Hybrid SSR and Transformer-coupled SSR. Talked about above reed relays were vacuumed relays and in this application the signal usually came to the relay pre-amplified and through the relay trigger circuit reacted with the TRIAC trysistor switch. This application generally provided excellent isolation because of the vacuumed reed relay property.



**Figure 12: Reed-Relay-Coupled SSR**

**Permission requested from http://** **www.wiringdiagrams21.com**

The last type of SSR was Transformer-Coupled SSR in figure 13. In this application this time signal passed through a DC-AC converter to guarantee that it was AC signal to be processed which was basically a transformer and this primary excitation’s end product as the secondary excitation got to the TRIAC for the turn on and off.



**Figure 13: Transformer-Coupled SSR**

**Permission requested from http://** **www.wiringdiagrams21.com**

After all the initial research, we decided to go with the Photo-Coupled SSR. First of all this decision was made against electromechanical relays since their sizes were too big and the idea of moving parts rose questions about their reliability. They were very quick in responding to any turn on and off input and there was no internal arching that would harm any semiconductors. Another advantage we considered was its PCB compatibility both because of its small size and better technology. When powered with AC source, solid state relay had the benefit of zero crossing switching which reduced noise in the as a result the circuit could experience switching where the voltage crosses were zero.

**2.4.2 Magneto Resistive Field Sensors**

Shown below in figure 14 was a magneto resistive field sensor. It reminded us of the adjustable resistors that we used in electronics lab. The principle was: with the help of very thin ferromagnetic films, a magnetic field was created on the wire that carried the current. This magnetic field was used to adjust the resistance by tilting it in certain angles and changing the magnetic field intensity. The resistance was the largest if the magnetization and the current were acting in the same direction, and the adjustment could be minimized if the angle of magnetization and the current had 90° in between. [22]



**Figure 14: Magneto Resistive Field Sensors**

**Permission requested from** [**http://www.directindustry.com**](http://www.directindustry.com)

There were four magnetic field sensitive resistors in a sensor and these resistors were designed to form a bridge. This was called the measuring bridge and the way it worked was between the four magnetic field sensitive resistors any detected magnetic field change was measured as a potential difference along each side of the bridge. This magnetic field created might as well have been coming from a wire and the intensity of the magnetic field created by the wire could be measured this way and could be converted to another electric variable in this separate circuit and be used to measure the intensity of the current. The important thing here was the biasing of the sensor with a permanent magnet so that the measurements could be accurate for the magnetic field; read, this way, there was no need for saturation which would cause the precision of the measurement to change. Usually the output voltage was a square wave voltage which was easily convertible to digital with a Schmitt trigger. [22]

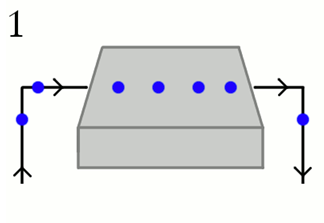
The way magnetic field sensitive resistors told the intensity of the magnetic field was actually pretty simple. The full bridge formed from 4, or the half bridge formed from 2, magneto resistors. Once put into a magnetic field they responded to it. This response since they were on the opposite sides of the bridge and placed inverted from each other was opposite from each other; in other words while the first one’s resistivity increased the other ones resistivity decreased and the potential difference created between the opposite sides of the bridge was the measurement method of this change. Just like any other magnetic component that functioned with magnetic fields the linearity of this component was not that great and compensation with an aluminum conductor had to be implemented in to the component, usually above the magnetic resistors, so that better linearity was achieved. In hindsight this linearity was not that big of a deal after installing this conductor since the current that was created in this conductor by the magnetic field did the compensation job itself. This application at the same time added to the convenience of this component since it compensated for the temperature dependency along with its other advantages such as its good accuracy and high sensitivity. At the same time it could take a lot of mechanical stress, it was durable, it had fast response rate, had a decent size, could be used in harsh conditions and the cost was reasonable compared to other types of current sensors.

It was a tough competition between Magneto Resistive Field Sensor and Hall Effect Current sensor since they were both very good options for our project’s current measuring component. Against all the advantages of Magneto resistive field sensor, the Hall effect current sensor had its own advantages such as: they were logic capable, highly linear, non-mechanical structure, resistance against heat, perfect to measure high currents but they wouldn’t break easily, were insensitive to dust, vibration, humidity, cold and hot so their characteristics and measuring sensitivity remained the same because of their well-sealed structure.

The downside of Magneto resistive sensor was its limited linear range, temperature drift and its sensitivity to interfering magnetic fields from other components compared to that of the Hall current transformer sensor. Once the temperature characteristics were compared, the Hall Effect current sensor was superior as well since magneto resistive current sensor could also have temperature drift.

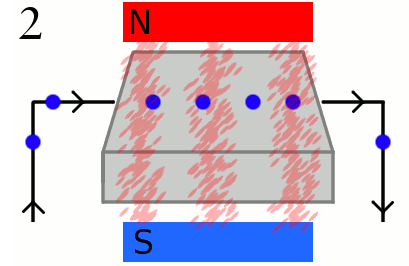
**2.4.3 Hall Effect Current Sensors**

The first discovery and where the definition of the Hall Effect came from was a rectangular thin conductor, in this particular case it was a gold sheet with current running through, got placed in a magnetic field perpendicular to the direction of the current. The magnetic field created effects such that while the current was uniform without it, with the magnetic field on, the concentration of current lost its form and the distribution of the current lost its uniformity. This was called the Hall Effect which resulted with a potential difference called the Hall voltage. The important point here was the output voltage, in other words the Hall voltage was proportional to the magnetic field and the current’s cross product and this was the basic principle of the sensing circuit. Below in figures 15, 16, and 17 the diagrams described how the Hall Effect worked. In the first representation, in figure 15, there was a conducting plate shown. In our case this was the main wire that the power strip was plugged into instead of the conducting plate that had current running through.



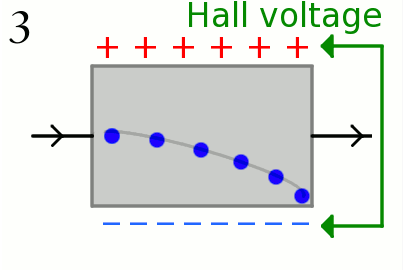
**Figure 15: Creation of Hall Effect with the Magnetic Field Representations Permission requested from** [**http://www.explainthatstuff.com**](http://www.explainthatstuff.com)

The blue dots were the electrons’ representation where our projects power measuring strip drew from the main grid. Below though, the same conducting plate was placed between the poles of a magnet. This at the same time meant that the electrons passing through the conducting plate were under the influence of the magnetic field as well. [21]



**Figure 16: Creation of Hall Effect with the Magnetic Field Representations Permission requested from** [**http://www.explainthatstuff.com**](http://www.explainthatstuff.com)

The reason for this movement of the electrons was the Lorentz Force or in other words any engineering student would call this effect as the good old “Right Hand Rule”. Since the regular path of the electrons was interrupted and a deviation occurred towards the bottom of the figure representation, now we could say that the opposite sides of the conducting plate carried opposite charges (as represented in the above figure). This created a potential difference and was called the “Hall Voltage”. [21]



**Figure 17: Creation of Hall Effect with the Magnetic Field Representations Permission requested from** [**http://www.explainthatstuff.com**](http://www.explainthatstuff.com)

Now, if we were to assume that we had more of those blue dots above in the figure which represented electrons. This would mean there would be more electrons gathered at the bottom of the figure and this would mean there was the same amount of holes on the top of the representation and this would mean there was more potential difference; in other words more “Hall Voltage” created proportional to the intensity of the current passing through the conductor. [21]

So in this sense, developing a component where the magnetic field was facing the wire from all directions or sometimes four directions with constant magnetic flux would get the current exactly proportional with the Hall voltage. The more the voltage the more the current principle with almost perfect measuring accuracy would be achieved since magnetic field supplied from all directions would take the stress factor out of the equation.

So what were the advantages of using a Hall Effect current sensor? There were countless advantages, which varied from the safety to precision. Now let’s go through these advantages, which led The Hall Effect Sensor to be our pick for our project.

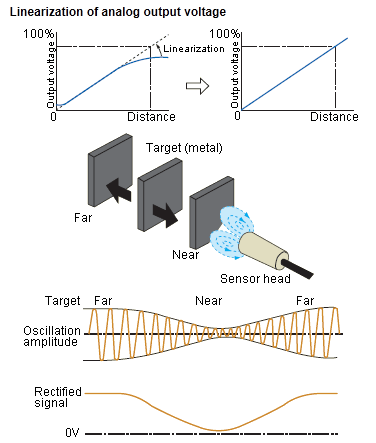
* Hall Effect current sensor shown below in figure 18 could be isolated from another high voltage in the same system. The real life applications of this fact were usually the safety applications to measure high current and voltage of systems that were even dangerous to get close to.
* Hall Effect current sensors were insensitive to dust, vibration, humidity, cold and hot so their characteristics and measuring sensitivity remained the same thanks to their very well-sealed, closed structure.
* Hall Effect current sensor’s other advantage would be considered to be internal temperature absence since they didn’t get hot because of their inductive structure but not resistive.
* Hall Effect current sensors were convenient to use because of their non-mechanical structure same as their immunity against heat. So not only they were perfect to measure high currents but they wouldn’t break easily since there were no mechanical moving parts.
* They did have a very long life because of the above reasons.
* Hall Effect current sensor were more robust than others, others would wear out by time because of arching
* They were logic capable.
* High speed and repeatability.
* Highly linear.



**Figure 18: Hall Effect Current Sensors**

**2.4.4 Eddy Current Sensors**

Eddy Current sensors like in figure 19 were another magnetic field based measurement component. Although it was called Eddy Current Sensor most of its use was done for measuring distances. Because of its name we kept it among the titles of our research material for the current sensors but despite its name the real use of it were to measure distance, position, amplitude, alignment, roundness, pitch and permeability of an object. Though, if the distance was kept constant, the application could be used to measure the current as well. The basic structure of the component was just a probe attached with a sensor that created an alternating current at the tip of it. The alternating current created a magnetic field naturally and this was the method Eddy current Sensor functioned, with these magnetic fields. If we were to assume this sensor attached to the tip of the probe as a coil with AC running through it, we would expect magnetic field shooting out of it. If we were to get the probe close to a conducting material these magnetic fields would have had an effect on the conducting material. One of these effects was usually unwanted but in this case worked to the users’ advantage, Eddy Currents.



**Figure 19: Eddy Current Sensors**

**Permission requested from /www.sensorcentral.com**

The magnetic fields created by the eddy currents opposed the magnetic field directed to the target conducting material. The sensor measured the interaction of these magnetic fields. As mentioned above the current measuring application would be achieved; the sensor could measure the change in magnetic in the same way the electric fields and produced a voltage that was proportional to the change in distance but if the distance was kept the same, this time the change in the magnetic field intensity would give the change in current, since the more the current more the magnetic field surrounding it proportionally existed.

The advantage of using an Eddy Current Sensor would be its tolerance to dirty environments, having another material between the gap of measurement not being a problem but since our power measuring strip was going to be closed and would be for inside the house usage, environmental dirt nor having another material in between the measurement gap would not be an issue for our application. Despite the fact, using Eddy Current Sensor would actually cause some problems. Since the main goals of the project was to measure power and then if it was necessary restrict the usage of this power, the most imperative part of the project was to measure the power correctly. This correct measurement of power was going to reflect to the correct outcome and calculation of the energy cost. A wrongful measurement would jeopardize the whole project and using an Eddy Current Sensor would cause that. In the power measuring strip there would be many components wired all together in a narrow space. More or less each of these components would have a magnetic field or Eddy current that would interrupt the measurement precision of the Eddy Current Sensor. Besides that our other option Hall Effect Current sensor provided way more advantages for our measuring circuit such as High speed and repeatability highly linearity, a very long life and isolation from another high voltage in the same system so it wouldn’t be affected by the other components just like Eddy Current sensor would. On top of all these advantages against Eddy current sensor, Hall Effect current sensors were insensitive to dust, vibration, humidity, cold and hot so their characteristics and measuring sensitivity remained the same just like Eddy Current sensors.

**2.4.5 Current Transformer Sensors**

Current transformer Sensors were another application of power and current measuring systems. The way the component worked was very simple and very much like the basics of electric power production. We all knew that a conducting wire that had current running through created a magnetic field surrounding the wire. With the right hand rule we could find the direction of this magnetic field as long as we know the direction of the current flowing. Current transformer sensors were based on the principle to capture this magnetic field in another conductor and use the magnetic field in it to measure the current running in the wire it surrounds. The way to do this was to wrap another conducting wire around this conducting material that surrounded the initial cable and this time created current in this wire induced by the magnetic field running through the conducting material. In figure 20 a current transformer was shown.



**Figure 20: Current Transformer**

**Permission requested from electro-tech-online.com**

This current was proportional to the original current which we wanted to measure at first and the ratio was the number of windings that were wrapped around the conducting material that had the magnetic field going through it.

There were three major types of Current transformers and the description above may have gone for either Window, Bar or wound current transformers. Window type was usually used for measuring circuits with less than 600 Volts along with Bar type; on the other hand the wound type current transformers were used for measuring over 600 V. The ratio dealt with window type was a little different than the other transformers since the conductor for this type that carried the magnetic field was considered the primary winding and there might be additional turns that might be used according to the demand. Under these circumstances when used in a circuit to measure current the transformer ratio had to be divided to the turns to find the actual current value. Bar type current transformer, which was also used to measure low voltage circuits, was almost the same as window current transformer but with a bar installed so that voltage readings could be obtained conveniently as well as current readings. Finally the window type comes with fixed primary winding and as mentioned before it was used to measure high voltages.

Consideration of Current transformer sensor as our project’s power measuring component didn’t seem to be that logical since the other alternatives, both Hall Effect current sensors and magneto resistive field sensors, appeared to be better alternatives. Hall effect current sensor provided logic capability, highly linearity, non-mechanical structure, resistance against heat, perfect to measure high currents for safety concerns, were insensitive to dust, vibration, humidity, cold and hot and well-sealed structure. On the other hand magneto resistive field sensor had good accuracy and high sensitivity. It was durable, it had fast response rate, had a decent size, could be used in harsh conditions and reasonable price. Current transformer sensors were very suitable for higher voltages and their sizes were usually bigger and they don’t provide the advantages mentioned above.

**2.5 Wireless Communication**

For communications between devices it was important for the installation to be feasible. Implementation of a home monitoring system using a wired platform could not be justified in both cost and time. A wireless technology that could penetrate walls and communicate securely with a rage of around 100 feet was necessary. We chose a range of 100 ft due to problems associated with absorption and reflection of frequency waves. This product was meant to be used in-doors where there would be several line of sight issues. Choosing a large range of 100ft should have ensured that the signal loss would not affect the quality of the product.

To explore the options in wireless technologies one must have considered the characteristics of these devices. For this project the devices most important attributes were low power consumption and range. This would ensure that the individual outlets in a home were able to communicate back and forth without the need for constant maintenance. Other factors such as data rate, speed, and security were also considered. The wireless technologies researched were the EZ430-RF2500 wireless development tool from Texas Instruments, the Xbee 802.15.4RF modules, Arduino Bluetooth technology and the Xbee ZB module.

**Overview TI eZ430-RF2500**

The EZ430-RF2500 was a development tool that included both the MSP430F2274 microcontroller and the CC2500 wireless transceiver. This development tool also comes with additional debugging software and examples. All of the communication and power monitoring was realized using this microcontroller in conjunction with the wireless transceiver. By using these tools together all the microcontrollers used for power consumption readings had the ability to communicate with each other using 2.4-GHz wireless technology. [9]

Features

* CC2500 2.4 GHz, ISM band multi-channel low power transceiver
* Supports MSP430 Application UART allowing serial communication to PC
* SimplicitTI, low power network stack
* Supports development with MSP430 hardware
* 21 available development pins
* Highly integrated, ultra-low-power MSP430 MCU with 16-MHz performance
* Supports development with some 2xx Spy Bi-Wire devices
* Supports [eZ430-T2012](http://focus.ti.com/docs/toolsw/folders/print/ez430-t2012.html) and [eZ430-RF2500T](http://focus.ti.com/docs/toolsw/folders/print/ez430-rf2500t.html) target boards

Figure 21 below of the wireless development tool gave a visual representation of the hardware. The rightmost part of the picture was the target board; the left part showed the debugging USB hardware that plugs into a computer. This board included both a CC2500 2.4GHz wireless transceiver and the MSP430F2274 microcontroller. On the target board pins 3 through 7 were general-purpose digital I/O pins A0 through A4. General-purpose digital I/O ADC10 pins A12 through A15 were given by pins 8 though 11 respectively. Pins 1 and 12 were used as ground reference while pin 2 supplies the voltage. [10]

The wireless development tool used SimpliciTI which was a lower power RF network protocol developed by Texas Instruments. It was generally used in alarms, controls, meters, or sensors. It utilized up to 8K, 1K RAM depending on the configuration. It also supported low power features for sleeping devices. This was an important feature in our device as small portable RF networks generally had onboard battery power. To minimize the power consumption it was necessary for devices to enter a sleep. Similar to the XBee mesh network the peer-to-peer network topology had the ability to extend the range of the device using repeaters. This feature was designed for use in simple small RF networks with fewer than 100 nodes. Although this technology was not as robust as the XBee mesh network it was still enough to cover the requirements of our design. The network protocol supported different communication topologies such as peer-2-peer and direct p2p through RE. For our design a basic direct peer-2-peer connection was sufficient. The feature for peer-to-peer messaging could be accessed using the following syntax.

• smplStatus\_t SMPL\_Send(lid, \*msg, len);

• smplStatus\_t SMPL\_Receive(lid, \*msg, \*len);

The MSP430 was especially useful for a project in which sensors were required for power consumption measurements. To do this we used pinput pins 8 to 11 described above. These pins were important due to their analog to digital(ADC) conversion characteristics. To measure a sensor using the MSP430 one could use ADC10 sampling. The ADC10 was a 10 bit ADC which had a total of 2^10 or 1024 values. [11]

The ADC10 was important for sampling analog data from external sensors. It could convert a voltage or temperature reading into useful information for the microcontroller. Using this information one could alter the behaviors of other appliances in the home. Several appliances were monitored simultaneously with data being shared with a main unit. This unit would receive all the data and sum up the total current being drawn. This load would be measured against the current delivered by the generator. If the load drain exceeded the load supply, then appliances of least priority needed to be interrupted so the generator did not overload. Overloading the generator may cause appliances in the home to fail.

Enabling the ADC10 was done simply by enabling the ADC10ON bit of register ADC10CTL0.

The MSP430 allowed for asynchronous communication using UART. By using UART the microcontroller could communicate with a computer over a serial port. For the purposes of our project all of the software may be tested with this wireless development tool. The wireless target boards had an integrated MSP430F2274 and CC2500 multi-channel RF transceivers. These two parts could be purchased separately and embedded into their own circuit.

The debugging software included in this package used IAR Embedded Workbench Integrated Development Environment (IDE) or Code Composer Essentials (CCE) for writing and debugging software. Both of these packages included the ability to code in both C and C++. Data was transmitted using UART on the MSP430. UART on this microcontroller was controlled by seven control registers and one read-only register, each register consisted of 8-bits. [12]

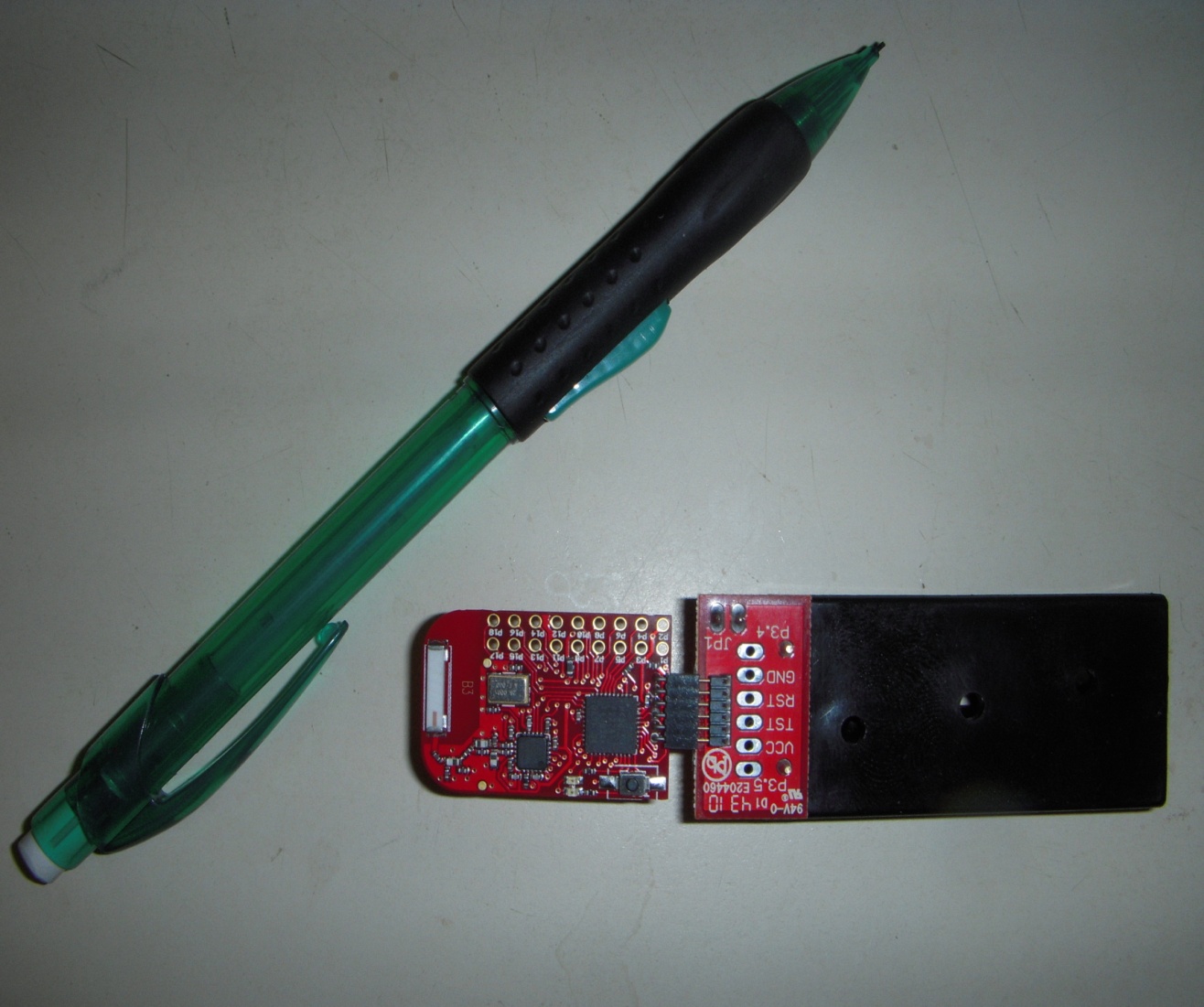


Figure 21: EZ430-RF2500 Development Tool

The CC2500 was a RF transceiver available in several MSP430 microcontrollers. The advantages of this part lay in its small size and lower power consumption. It only consumed 13.3mA while in operation. When it went into sleep mode it consumed as little as 400nA. To exit from the sleep mode it took 240micro seconds. It could be programmed to transfer data at a range between 1.2 to 500 kBaud. Operating voltages for the CC2500 should have been restricted between 1.8 and 3.6 volts. Due to the transceiver being embedded on the microcontroller these voltages were supplied by the MSP430. However, the voltages could also be supplied from a regulated power supply.

Figure 22 showed the input pins of the CC2500 transceiver. As shown the chip contained 20 pins. Some of these pins were used for ground, it was important to note that if this transceiver needed to be mounted on a ground plane. This was the primary chip ground connection. A 26 Mhz crystal was used to drive the synthesizer through the XOSC\_Q1 and XOSC\_Q2 pins. It also provided a clock for the ADC. For implementing the CC2500 transceiver on a board a number of extra components would be needed. These components and their configuration were also displayed above. The figure above did not include decoupling capacitors needed to supply voltage to the transceiver. The decoupling capacitors needed were two 220pF at 10% tolerance rated for 50V on pins 9 and 15 while pins 11 and 14(AVDD) required capacitors of 100nF at 10% tolerance with a rating of 10V. The values for the remaining components for the CC2500 Transceiver were given in figure 23 below.

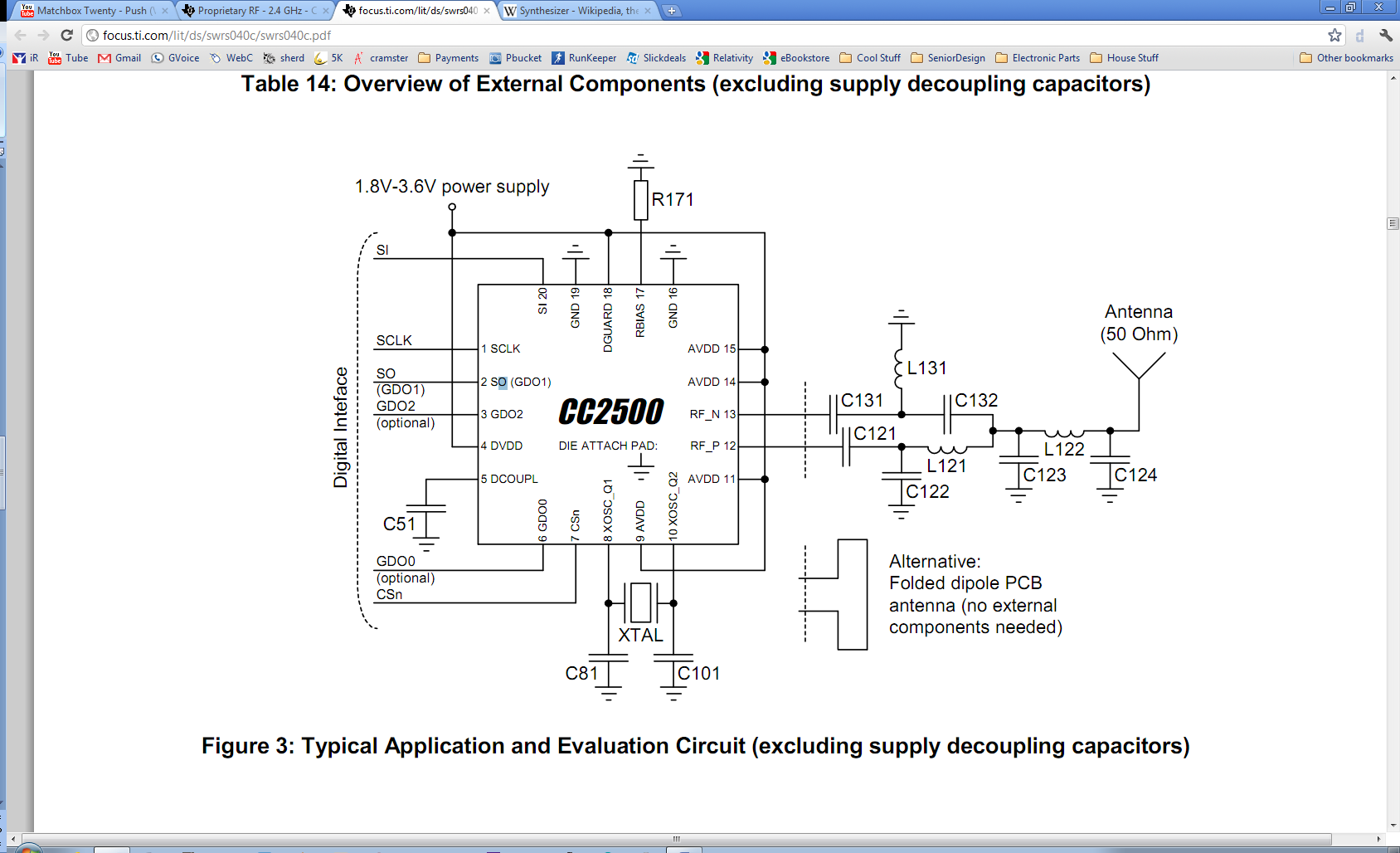


Figure 22: CC2500 transceiver and components

**Permission requested from Texas Instruments**

Communication between a microcontroller and the CC2500 was done through four input and output pins. The pins used were 20, 2, 1, and 7. Pins 6 and 3 were used to generate interrupts for the microcontroller. [13]

|  |  |  |
| --- | --- | --- |
| **Component** | **Value** | **Manufacturer** |
| C51 | 100 nF ±10%, 0402 X5R | Murata GRM15 series |
| C81 | 27 pF ±5%, 0402 NP0 | Murata GRM15 series |
| C101 | 27 pF ±5%, 0402 NP0 | Murata GRM15 series |
| C121 | 100 pF ±5%, 0402 NP0 | Murata GRM15 series |
| C122 | 1.0 pF ±0.25 pF, 0402 NP0 | Murata GRM15 series |
| C123 | 1.8 pF ±0.25 pF, 0402 NP0 | Murata GRM15 series |
| C124 | 1.5 pF ±0.25 pF, 0402 NP0 | Murata GRM15 series |
| C131 | 100 pF ±5%, 0402 NP | Murata GRM15 series |
| C132 | 1.0 pF ±0.25 pF, 0402 NP0 | Murata GRM15 series |
| L121 | 1.2 nH ±0.3 nH, 0402 monolithic | Murata LQG15HS series |
| L122 | 1.2 nH ±0.3 nH, 0402 monolithic | Murata LQG15HS series |
| L131 | 1.2 nH ±0.3 nH, 0402 monolithic | Murata LQG15HS series |
| R171 | 56 kΩ ±1%, 0402 | Koa RK73 series |
| XTAL | 26.0 MHz surface mount crystal | NDK, AT-41CD2 |

**Figure 23: Component values CC2500**

**Overview XBee ZB and 802.15.4 RF Modules**

The Xbee 802.15.4 RF considered were the regular and pro models. The advantage of the pro model being that it had a much longer range than the standard as seen in figure 24. XBee RF modules were interfaced with a microcontroller through the Data in and Data out ports. These were denoted as pins 2 and 3 on the Xbee RF module. Using the DOUT and DIN pins serial data could be sent or received by the microcontroller. The default rate for this asynchronous communication was 9600 baud or bits per second. To prevent buffering and framing errors the pins CTS RTS and DTR were used for handshaking between the Xbee and the microcontroller. This prevented situations in which the microcontroller could send serial information faster than the wireless module could process it. The digital input and output ports used 3.3v of regulated dc voltage.

The disadvantage of the Xbee 802.15.4 modules was in large due to the fact they did not support extended network range through routing. This was what gave the Xbee ZB a significant advantage for use in our design. Through using network through routing the ZB modules could extend the range of a network. On the other hand the 802.15.4 modules only supported point to point communications.

Similar to the EZ430-RF2500; this wireless solution also included a number of methods for debugging software through using a PC. This was accomplished by interfacing the Xbee module to a PC using a USB adapter. This adapter was pre-assembled and the RF module was simply connected through soldering or by using a breadboard. The USB adapter would allow communication between the PC and the Xbee. This device could be powered through a 5 V DC supply. [14]

|  |  |  |
| --- | --- | --- |
| **Xbee** | **802.15.4** | **Pro – 802.15.4** |
| RF Data Rate | 250 kbps | 250 kbps |
| Indor/Urban Range | 100 ft (30 m) | 300 ft (100 m) |
| Outdoor/RF Line-of-Sight Range | 300 ft (100 m) | 1 Mile |
| Serial Data Interface | 3.3V CMOS UART | 3.3V CMOS UART |
| Frequency Band | 2.4 GHz | 2.4 GHz |
| Serial Data Rate | 1200 bps – 250 kbps | 1200 bps – 250 kbps |
| ADC Inputs | (6) 10-bit ADC inputs | (6) 10-bit ADC inputs |
| Digital I/O | 8 | 8 |
| Antenna Options | Chip, Wire Whip,  U.FL, & RPSM | Chip, Wire Whip,  U.FL, & RPSMA |
| Encryption | 128-bit AES | 128-bit AES |

**Figure 23: Component values CC2500**

**Xbee ZB**

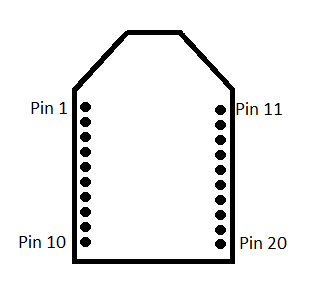
The Xbee ZB was used in wireless home applications. It was used for controlling wireless light switches and to display meter information to displays. The key features of the Zigbee modules were its low data rate, long battery life, and 128-bit AES encryption for data. Another important feature of the zigbee module was that it could communicate with a total of 65,000 devices. This was more than enough outlets for any home.

Features:

|  |  |
| --- | --- |
| * Low-cost, low-power mesh networking * No configuration needed for out-of-the-box RF communications * Support for larger, more dense mesh networks * 128-bit AES encryption * Frequency agility * Over-the-air firmware updates (change firmware remotely) * ZigBee mesh networking protocol       - Improved data traffic management       - Remote firmware updates       - Self-healing and discovery for network stability * Low-power sleep modes * TX current: * RX current: * Power-Down current: Industrial temperature rating (-40C to 85C) * 1Mbps Max data rate * 2mW output (+3dBm) * (4) 10-bit ADC inputs * (10) digital IO pins * 128-bit AES encryption * Local or over-air configuration * AT or API command set * Fully FCC certified | 45mA @ 3.3V  40mA @ 3.3V  <1uA @25 degree C |

Mesh networking was also supported by the ZigBee protocol. This essentially increased the range and reliability of our whole network. Instead of each power strip having to communicate back to a central location, each power strip microcontroller could relay their information via other power strips. This gave multiple paths for the data to flow. All of these connections between nodes were not done through our software. They were built in capabilities of the ZigBee, allowing for nodes to be dynamically updated and optimized during operation. If one of the power strips was powered down, then the flow of information would not be interrupted as it could still be relayed through a different node in the home. This would allow us to set enable a devices sleep mode without compromising communication of the devices. Using this technology the range between devices ceased to be an issue. All homes had several outlets in each room. Using this mesh network the size of a home would not add any difficulties to the network. In some cases it could even improve the network as more data paths were introduced into the system.

UART PIN communication with the XBee ZB as referenced by figure 24 was done through four of the pins. Pin 3 was used to transfer data from the microcontroller into the wireless module as an asynchronous serial signal. The remaining pins were referenced in figure 25. This signal would remain high until data was transmitted. Data was finally then transferred to another wireless module through the pin number 2 which was the data output pin. Pins 12(CTS) and 16(RTS) were used as output and input digital ports. The XBee RF modules could communicate with any other device that was also compatible with UART. [15]



**Figure 24: XBee Module**

**Mounting**

Xbee modules were designed to mount into a receptacle. These receptacles could be mounted onto a board using through-hole and surface mount technologies. By using receptacles there was a lesser chance of damaging the module. All soldering could be done prior to the installation. The modules were not sensitive to nearby processors or components. Care should have been taken for installing the antenna in a proper location. Metal objects close to the antenna would cause reflections that would reduce the efficiency of the antenna. Due to the nature of this project microcontrollers would be encased in a small enclosure. An external antenna must have been considered as the on chip antenna would most likely be insufficient to meet our needs. [16]

|  |  |  |  |
| --- | --- | --- | --- |
| **PIN** | **NAME** | **PIN** | **NAME** |
| 1 | VCC | 10 | GND |
| 2 | DOUT | 11 | AD4 DIO4 |
| 3 | DIN | 12 | CTS DIO7 |
| 4 | DO8 | 13 | ON/SLEEP |
| 5 | RESET | 14 | VREF |
| 6 | PWM0/RSSI | 15 | ASSOC AD5 DIO5 |
| 7 | PWM1 | 16 | RTS AD6 IO6 |
| 8 | RESERVED | 17-20 | AD3-AD0 DIO3-DIO0 |
| 9 | DTR SLEEP\_RQ DI8 |  |  |

**Figure 25: XBee Module Pins**

**Arduino Bluetooth Technology**

Features:

|  |  |
| --- | --- |
| * Microcontroller | ATmega168 |
| * Operating Voltage | 5V |
| * Input Voltage | 1.2-5.5 V |
| * Digital I/O Pins | 14 (of which 6 provide PWM output) |
| * Analog Input Pins | 6 |
| * DC Current per I/O Pin | 40 mA |
| * DC Current for 3.3V Pin | 50 mA |
| * Flash Memory | 16 KB (of which 2 KB used by bootloader) |
| * SRAM | 1 KB |
| * EEPROM | 512 bytes |
| * Clock Speed | 16 MHz |
|  |  |

The Arduino Bluetooth was a wireless technology using the WT11 class 1 Bluetooth 2.1 module mounted alongside an ATmega168 microcontroller. The Bluetooth modem could also be purchased separately and then connected to an Arduino board shown in figure 26. This was not an issue related to our project since all of our components would be connected using a professional PCB layout. The board components could be powered with a voltage range between 1.2V and 5.5V. This pin should be supplied with a voltage of around 3V. Voltages outside of this range would damage the board. The voltage was supplied to the board via the 9V input pin. The 9V pin cannot exceed 5.5V. This voltage could be supplied from the V+ on board DC-DC converter or it could also come from a regulated 5 volt power supply.

The device comes with 14 digital pins for both input and output. These pins could be accessed using functions defined as pinMode(), digitalWrite(), and digitalRead(). No pin could receive more than 40mA of current. These would be the pins used in interfacing the Arduino board to the Blueooth modem.

Developing the wireless interface for the Bluetooth modem required a Bluetooth dongle for the computer and a Blueooth modem for the Arduino board. These two devices could be used in the debugging phase to communicate with each other. As with the TI eZ430-RF2500, Bluetooth technology could also be used as part of a development tool for the wireless software. Once the software was tested and fully functional then each board would be loaded with the software so devices could communicate with each other autonomously. The final design would not require a computer.

Communication for the Bluetooth Arduino device was accomplished via the Serial port. Serial port 0 (RX) was used to receive TTL serial data while Serial 1 (TX) was used to transmit data to other devices. In addition to the RX and TX inputs the Bluetooth modem needs an additional PWR and GND pins to be connected. To initiate wireless communication we would need to access the serial communications port using the code.

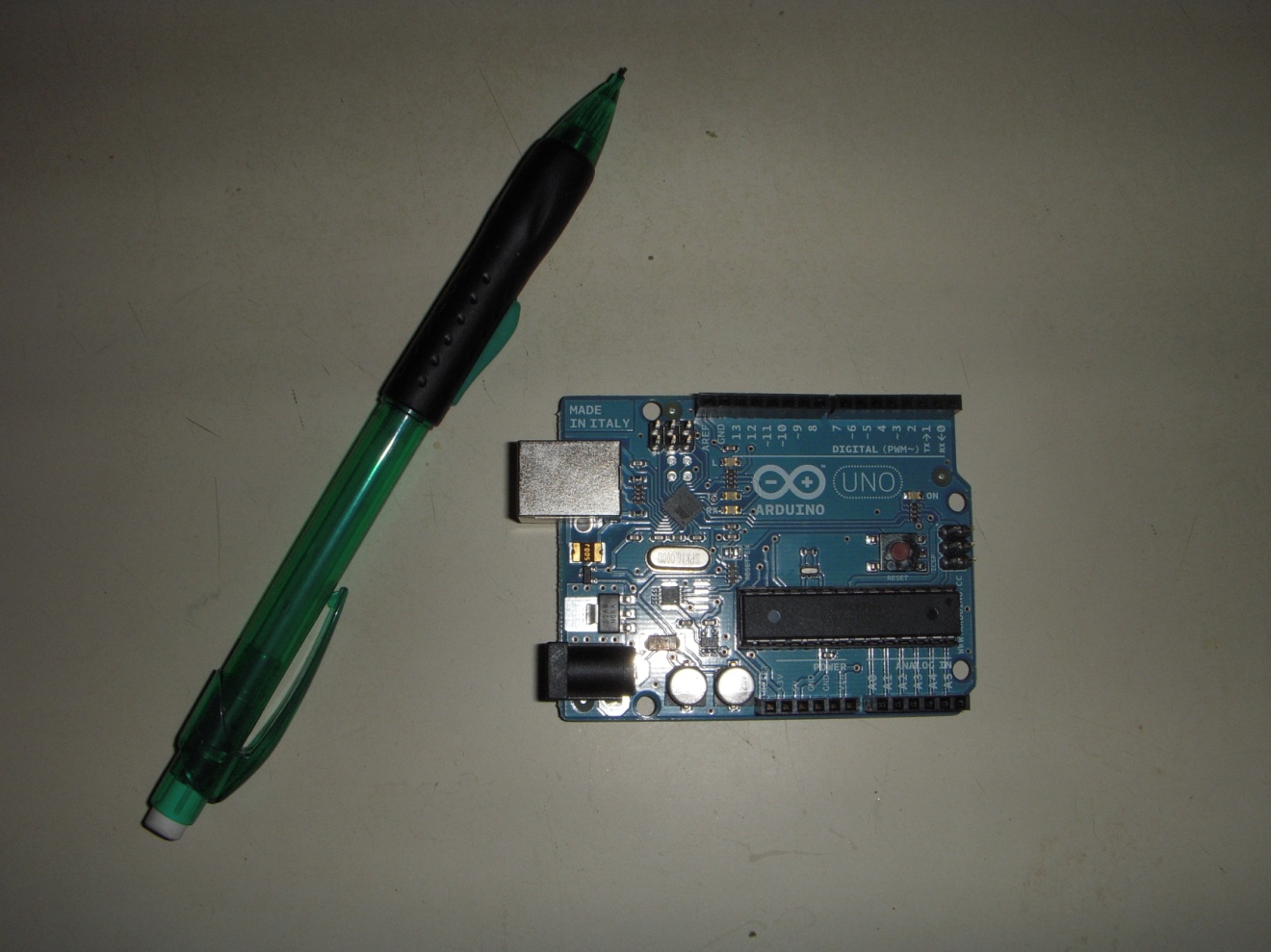
void setup() {

pinMode(ledpin, OUTPUT);

Serial.begin(9600);

}

Implementing Bluetooth communications using the Arduino board was extremely easy given the amount of information that was shared. Range with the Bluetooth WT11 could be an issue. Shown below the size of the Arduino was also an issue. Comparing this picture to the development tool of the TI eZ430-RF2500 one could see that there was a significant size difference between the device components. To take this design to the next step all of the components used for the power metering and control applications would need to fit inside a conventional power outlet.



**Figure 26: Arduino Uno**

For this project all of the discussed wireless technologies were possible. The Xbee ZB platform was widely used in home power metering applications. The implementation of this method may not be as challenging. In addition the node jumping characteristics of the Xbee ZB increase the range and versatility of that platform. Texas Instruments solutions to home metering applications were harder to implement but they were also cheaper.

The easiest to implement of all the technologies researched was the Bluetooth modem using the Arduino board. Since the Arduino board was an avid development platform among hobbyist there were many examples showing how to interface the two components. The biggest downside to this option was the cost. Arduino boards cost around $30 dollars. For the testing phase part of our project we would need a total of 5 boards. There was an advantage of not using the Arduino Blutooth. $115 could be saved when you compare the cost of the boards to the cost of parts we could sample for free from Texas Instruments.

Our final design choice for the wireless development of our project would be using the MSP430 with the CC2500 transceiver. This was due to its embedded nature of the C2500 transceiver and low power consumption. The cost and size of the MSP430 were also important features we considered. Given the amount of room for a conventional power outlet the production design needs to be small enough to fit inside of it. The implementation of these two components would be much more difficult than the technologies discussed earlier. Due to the lack of information regarding these components we feel there was a greater opportunity to learn. [17]

**2.5.6 Antenna Options**

All of the distance specifications shown on each part were given by using controlled experiments. It does not take into account the loss of signal due to absorption, reflection, and scattering of the waves. Metal surfaces could reflect signals. Other enclosures around the wireless module would absorb frequency signal. Due to these natural factors different antenna styles should be considered.

There were many antenna options for each of our wireless solutions. These antenna options give us the ability to scale our design. The considered wireless transceivers come with an on-chip antenna which was unlikely to meet our project specifications. While these antennas offer high portability they would not provide a strong enough signal for the design. Given the design specifications the transceivers would be contained within an enclosure. This enclosure would reduce the effectiveness of the on-chip antenna. A larger antenna inside the unit may be needed.

Wire whip antennas similar to those seen on older cars offer a longer range of communication but must be visible for high performance. This antenna option had a clear advantage over the on-chip antenna due to its longer range. However it does not provide much portability. Aesthetics would also be an issue. A consumer would prefer a hidden antenna over an unattractive whip antenna. Given the time constraints the project would be developed allowing form to follow function. Other antennas such as those used in laptops were more attractive but harder to implement.

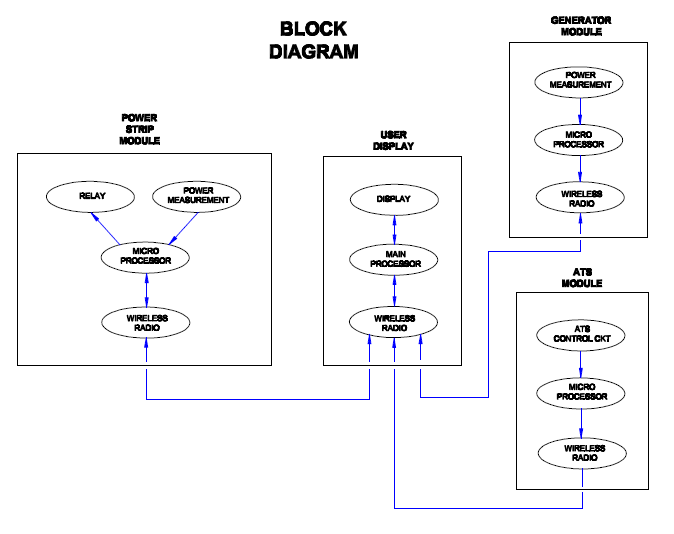
Wireless laptop technology uses a U.FL, RPSMA antenna. This solution offers a small footprint for PCB mounting. When attached to this Male connection a female coaxial RF antenna could be used to boost Wi-Fi signal. This antenna does not have to be exposed outside of the unit.

For the initial prototype of this project aesthetics would not be an issue if the on-chip antenna was not powerful enough to meet the design specifications we would mount a whip antenna outside the unit to meet the range required.

**2.6 Microcontrollers**

Each block of our design would incorporate a microcontroller, and each of these would communicate with a central, “hub”, microcontroller which would be physically connected to the LCD screen output. Each microcontroller would need to take in data from its source, analyze that information, and wirelessly transmit the necessary data to the hub microcontroller. The microcontroller associated with the Automatic Transfer Switch (ATS) module would need to communicate which power source was currently being used for the house. The microcontroller connected to the Generator module would take the information from the current/voltage sensor and tell the hub the maximum supply voltage available from the generator. The microcontroller connected to the Power Strip module would analyze the power measurement data from each of the three outlets on our power strip, and submit this information to the hub microprocessor. This microcontroller would also need the ability to turn on or off any and all of the outlets, based on the supply needs of the user and the power supply capabilities of the generator (when in use). Finally, we would need a centralized microprocessor at the LCD module which would take in all the pertinent data from all other modules as well as inputs from the LCD user display, and transmit orders to the power strip microcontroller accordingly. In addition, it would need to relay information back to the LCD user display. In total, this give four separate microcontrollers, each with its own wireless transceiver, capable of communicating to and from the User Display module. These concepts as they relate to the overall project, in regards to the individual microprocessors, were portrayed below in Figure 27. Ideally, all of these microcontrollers should use minimal power, and have the ability to remain in a standby mode for long periods of time, as well as wake up quickly.

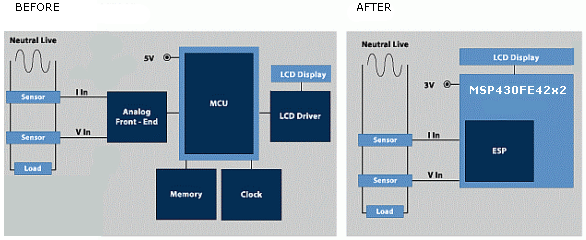
Choosing a microcontroller among the many options available to us was a difficult task, as there were many viable options. As the technology expands and develops, more complex microcontrollers that have many more functions were available for minimal costs. We thus wanted to simplify and break down our specific needs as concise a form as possible, to make the best choice clear. Our microcontrollers must be low power consuming, compatible with our chosen wireless interface as well as each module in our main block diagram, and with a fast response and relay time to minimize the time that our end user spends without power to his home.



**Figure 27: Microcontroller block diagram**

**2.6.1 MSP 430**

One of the microcontrollers considered was TI’s MSP430 family of 16-bit RISC mixed-signal processors, shown below in Figure 1. In addition to the great support from TI for the engineering students at the University of Central Florida, the MSP430 offered tremendous power and a wide array of applications and capabilities for very low power consumption. The MSP430 offered many built-in features for the monitoring and metering of all utilities, including one-to-three phase electricity metering. In addition, TI boasted having the “world’s lowest power microcontrollers” in its MSP430 line, essentially a high performance-to-power ratio. A low power demand from the microprocessor was ideal for our project, which would allow the generator to support as many electronic devices as possible when in use. The MSP430 also contained several built-in low power modes, where the processor drained virtually no power, and could be woken up with just a 1 micro-second delay. These modes varied slightly from each other, and could be easily chosen between based on the needs of each of the microcontrollers in our project. As shown below in Figure 28, the MSP430 microprocessor available today was an improvement from previous versions and other products for single phase metering applications, due to the consolidation of many parts into one, and also very easily connected and communicated with an LCD display, as would occur in our project. This would work well for our group, as dual and three phase metering was not necessary in a residential home.



**Figure 28: Microcontroller**

**Permission requested from www.ti.com**

The seven (7) LPMs (Low Power Modes) of the MSP430 allowed for increased optimization of coding. With the sub-1 µs wakeup time and several different interrupt sources in combination with the LPMs, only the corresponding clocks and peripherals that were needed would be used. The low power operating modes were as follows:

* Ultra-Low Power Active Mode: Down to 120 µA/MHz @ 2.2V
* Standby Mode with self-wakeup & RAM retention (LPM3): Down to 0.7 µA @ 2.2V
* Standby Mode with self-wakeup (LPM4): Down to below 100nA @ 2.2V
* Shutdown Mode with RAM retention (LPM3.5): Down to below 100nA @ 2.2V
* Additionally, the MSP430 had instant (less than 1 µs) wakeup time from Low Power Modes, as well as an always-on zero power brown-out rest.

Texas Instruments had available a line of MSP430 microcontrollers built specifically for the purposes of utility metering. This series, the MSP430AFE2xx series, includes low-power 16-bit analog-front end (AFE, which contains a tuner and an analog-to-digital converter, and which here was specifically used for measuring) MCUs.

**2.6.2 Arduino Uno**

A second microcontroller brand that was considered was the Arduino line of microprocessors. This was an open-source platform, which was designed to be easy to understand and provided both amateur and professional engineers a common platform to design and create projects to their needs. Due to its open-source nature, support for this microcontroller could be hit or miss. Typically, when you ran into an issue, someone else had run into that same issue before, and it was only a matter of time spent scouring the forums for the correct answer to suit your needs. However, this could be a very difficult, tedious process, and it may also turn out that the answer did not exist. Thus, reliable support would be a concern with this choice. The Arduino also tended to be a cheaper alternative than other microcontroller platforms available, which would definitely be a benefit for our project. Additionally, the programming and both the software and hardware that would be developed for this board were open source, allowing for optimal personalization and cost-saving measures. The software ran across all major platforms (Windows, Macintosh OSX, as well as Linux), which was a good feature to have to allow work at nearly any computer. However, our group had access to Windows computers on a regular basis, so this benefit was limited. [17]

Our group considered microcontrollers in several I/O board layouts from the Arduino line. First, the Arduino Uno was the standard layout, with a USB port, 14 I/O pins, and 32KB of flash memory. This board could be powered through the USB connection, and the power source was selected automatically. Additionally, the board could be powered by an external battery supply, either into an AC adapter or directly into a battery. Its operational voltage was between 6 and 20 V, though it was recommended to operate at >7V; below this value, the board may become unstable. Conversely, operating past 12V may cause the board to overheat. Therefore the recommended range was between 7 and 12 V. This information was summarized below, along with other pertinent data, in Figure 29.

|  |  |
| --- | --- |
| **Factor** | **Data** |
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Recommended Vin | 7V-12V |
| Limit of Vin | 6V-20V |
| Digital I/O Pins | 14 |
| Analog Input Pins | 6 |
| Flash Memory | 32 KB (31.5 available) |
| Clock Speed | 16 MHz |

**Figure 29: Arduino Uno Specifications**

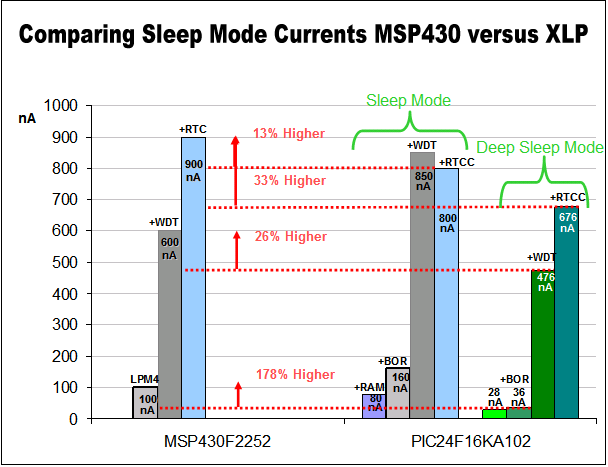
**Permission requested from www.microchip.com**

There were four power pins: VIN, 5V, 3V3, and GND. The 5V pin was used to power the components on the board, and the 3V3 pin provides a 3.3 V power supply. The Arduino Uno had 14 digital I/O pins which operate at 5 volts, and able to receive a maximum of 40 mA current, and each with an initially disconnected internal pull-up resistor (valued between 20 and 50 kΩ.

A more powerful board than the Arduino Uno, the ATMega 2560 was a board layout and processor that would be one step up in terms of raw computing power. This board had 54 digital I/O pins, 16 analog inputs, 4 hardware serial ports (UARTs), and a 16 MHz crystal oscillator. The board had 256 KB of flash memory, which was a significant increase from the Uno (32 KB). More memory would allow our group to accomplish more tasks as well as store more information for processing before having to clear space. This could allow for more flexibility in the programming of each of the MCUs, as well as in the data retrieval by the main processor in the LCD. Additionally, the Mega 2560 (which was a 2nd generation controller in this line) could be powered either through the USB port or an external power supply. This extra feature may come in handy in allowing more flexibility but would more than likely not be used, as this microprocessor would be powered independently of our LCD (which may use a USB connection).

**2.6.3 Microchip PIC**

A third microcontroller considered was the PIC (short for Peripheral Interface Controller) manufactured by Microchip. The PIC microcontroller was available in a variety of configurations and specifications in 8-, 16-, and 32-bit sizes. There were several draws for using the PIC microcontroller. First, there were 9 programmable sleep modes, during which the PIC draws less than 1 µA. Five of these modes were a “deep sleep”, which was a preferable choice for programming applications with long periods of inactivity, and which includes 7 possible wake-up sources. These sources included: Brown-out Reset, Real-time Clock, Reset, Timer, Interrupts, and Power-on Reset. Additionally, in direct comparison with the MSP430 series of MCUs from TI, the technology in the PIC microcontroller required 13% less current in its sleep mode for a real time clock, and between 33% and 178% less current in its deep sleep modes. Only in the Time mode was the current less in the MSP430. This information was shown below in Figure 30.



**Figure 30: MSP430 vs. XLP**

**Permission requested from www.microchip.com**

Second, the PIC MCU ran on very low power, which would be useful for our project. Sleep currents for the PIC run as low as 20 nA, and active mode currents could be as low as 50 µA/MHz. Also, more than 80% of the basic instructions could execute in only a single clock cycle. This could help to reduce the time needed when multi-tasking; for our project this means that nearly as soon as the user specifies which utilities and outlets were to be turned on (when running on generator power), this information could be relayed to the other microprocessor and, within 2 clock cycles, the outlets could be active. [7]

The mid-range architecture PIC which our group would be interested in had between 8 and 64 pins, a single interrupt ability (which may limit the capabilities of our project), up to 368 bytes of data memory, 14 KB of program memory. The MCP3905 was an energy metering integrated circuit available from Microchip and was designed specifically for use with the Microchip PIC. It was capable of supplying active power measurement for single-phase, residential energy metering, and would thus be very useful for our project. The IC features an easy-to-use capability to supply a frequency which was directly proportional to the Pave, or average active real power, as well as less than 0.2% error on most applications.

As none of the group’s members were sufficiently fluent in other programming languages, another concern for choosing a microcontroller was the capability to program the MCU in C, C++, or (more likely) a combination of the two. Having a microcontroller with this capability would greatly reduce the amount of time required for the group’s members to learn to program the necessary functions and code for the signal processing and data manipulation portion of our project. The MSP430 uses a combination C and C++ languages in all its programming, and would thus be a valid choice. The Arduino uses its own programming language, whose syntax was very similar to C, and thus would pose only a small challenge to learn. Finally, the Microchip PIC used an assembly programming language, similar to that used in the MC68HC11 microprocessor. However, there was a programming package available from the manufacturer (Microchip) that allows design-users to program in a friendlier, C environment.

**2.7 PCB Fabrication**

The final printed circuit board needed to be durable enough for our presentation. There were many hobbyist options when it comes to creating your own PCB boards. We would consider some of these options as other methods were not as cost effective and take longer to process. After concluding the test procedure a whole mockup of the design would be implemented using one of the following design options.

The first option included creation of the PCB using a permanent waterproof marker and an acidic FeCl3 solution. The advantage of this option was that it used materials which could be found locally. Implementation of this design could take hours as opposed to the days or weeks it could take to receive a manufactured PCB. Using this method a layout could be created in Photoshop or Microsoft Paint. After your circuit layout was printed you simply place it on top of a copper sheet and mark where the holes should be. Now remove the template and simply trace the connections on the copper sheet with your marker. Applying a coat of ink right before the copper board goes into the FeCl3 solution adds a layer of permeability. The solution would not eat through the ink; instead it would corrode all of the copper around your connections. Wash off the PCB thoroughly and then mount your parts onto the board by using a solder. Small parts should be mounted first. [18]

The problem with this hobbyist option was that drilling the holes for such a small component as an MSP430 would not be possible. The pin connections were too close and the risk of damaging the board was significant. This method seemed to be most effective when you were building a circuit which does not implement components that were small and fragile. This option could be used in our design to connect the larger components. Smaller circuits for components such as the transceiver and the MSP430 would be created using professionally manufactured boards. These companies accept designs through their website and have a turnaround time of around one to two weeks.

Online vendors such as pcb.com or expresspcb.com offered a professional PCB product. By using a third party vendor we could be certain that all the components were mounted correctly and that there were no erroneous connections made between the components. These companies offer high quality fabrication at a higher cost. Although processing time and cost were higher we were benefitted with some advantages in reliability and accuracy. 4pcb.com currently offers a free PCB with the order of four or more PCB at a price of $33 each. This area would be enough to meet the design for all of our circuit components and different units. By printing the same design on each board we could save in fabrication costs. In total we would print five boards which could create 10 circuits. Three in total for each power strip along with one for the main circuit and an additional current sensing circuit near the generator. Extra boards would be included in the order in case we run into any problems.

Due to time constraints we would order an initial test board as soon as a design was finalized. This would allow us to test the circuit for a single power strip. If the circuit passes the testing phase we would order other identical circuits to build the remaining circuits.

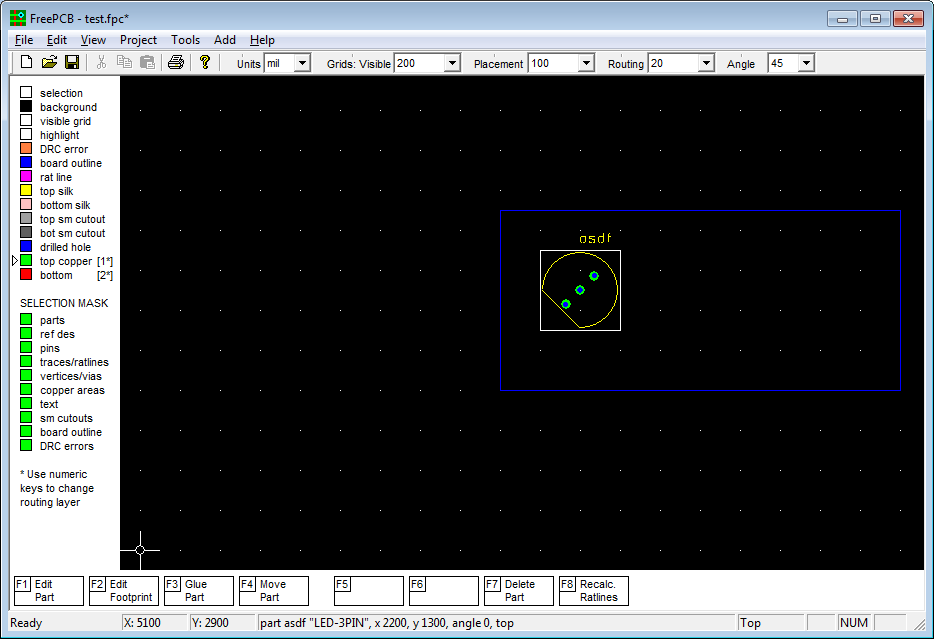
To design these circuits companies required designs which were created using specific file formats. In the case of pcb.com they accept gerber, drill, and drill tool list files. Other formats such as ODB++ and AutoCAD were also accepted. There were several programs that offer the ability to create designs in these file formats. The different options for creating the PCB layouts were explored below. [19]

**2.8 PCB Software**

To create our PCB we would need a schematic diagram. We would consider two different options for PCB software. The first option for our PCB design would be using EagleCAD. It contained a large library of parts from various manufacturers. Limitations with EagleCAD include size and layers. These limitations were lifted if you purchase the professional version of the software. At around $1500 the processional version was out of our budget range. The light version of the software was only $50 but it imposes limitations on the amount of schematic sheets, layers, and the area size. The area allowed using the light version of the software was only 100x80mm. These restrictions should not be a problem because our design was small enough to fit inside the size requirements. In addition the two layers provided would be enough. In choosing between the two software packages the only decision we need to make was in the cost and feasibility. The most important feature with the paid version of EagleCAD was the parts ordering process. After the design was finished this feature allows us to order all of the parts in the project at once. This feature was very attractive since we cannot afford any delay during this part of the project.

The second option would be using FreePCB; an open-source PCB editor. As seen below in figure 31 the FreePCB software was intuitive and very easy to use. It provided up to 16 copper layers and we could design a board of up to 60 x 60 inches. Users have the ability to add pins traces and stub traces visually. The menu on the left gave quick access to select different masks. By selecting different masks users could edit pins or traces without altering other components of the PCB. It turns out that this software was very similar to EagleCAD but with the limitations removed. The downside to the FreePCB software was that it comes with a limited library of parts. It also lacks the ability for a user to purchase all of the parts needed for the design through the software. [20]

Upon considering the limitations of each PCB software we felt confident that the better choice would be EagleCAD. Due to the requirements of our microcontroller design the hobbyist version of EagleCAD costing $50 would be enough to develop the design. This version of the software imposes heavy limitations on the standard features. These limitations would not cause any issues with our design since the design could be developed within these constrictions.

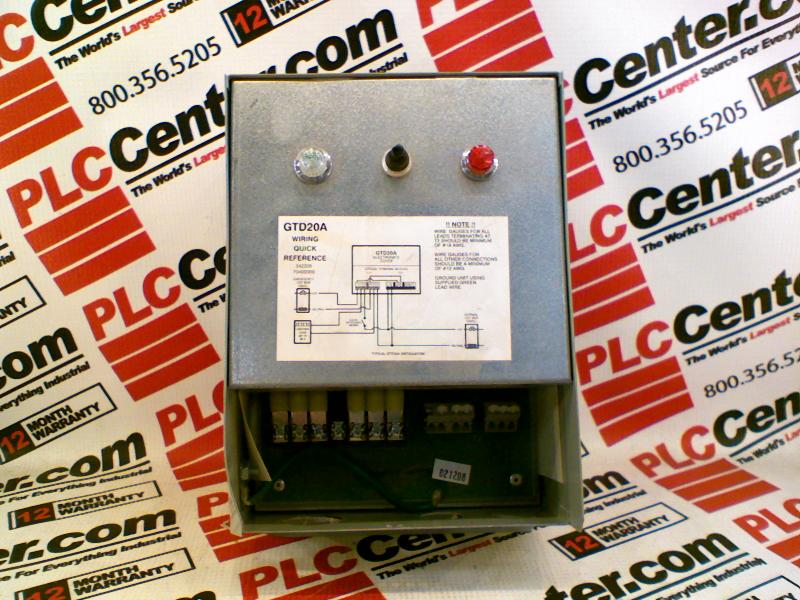


**Figure 31: Free PCB Software**

**2.9 Automatic Transfer Switch**

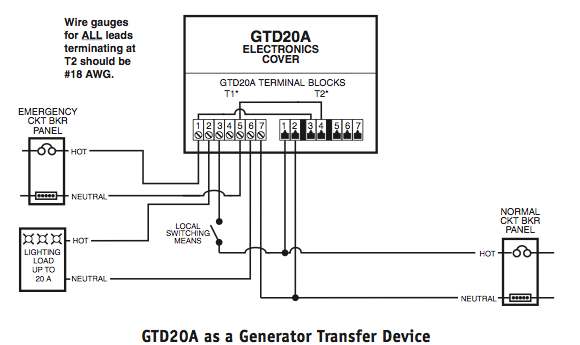
An automatic transfer switch was a device commonly used to switch an electrical loads power source from normal to emergency power during a time of normal power failure. This ensured a minimal disruption in available power and allows the consumer to keep on utilizing whatever electrical appliance they were using before power loss. As soon as normal utility power was restored the automatic transfer switch senses this change an then switches the load back to normal power.

Our project would demonstrate the concept of using our power strip module to automate and monitor an entire house. However due to cost, time, and location this was impossible so we were demonstrating the overall concept using a small-scale model. To replicate normal and emergency power we would utilize the common receptacle outlet seen in any classroom. These receptacles were rated for 120 volts, 20 amps and supply more than enough power for the purpose of demonstration. With the availability of 20 amps coming from the receptacle we had to ensure the transfer switch we selected was rated for it. This was why we chose to use the Bodine GT20A transfer switch in our project. This transfer switch seen below in figure 32 was normally used for emergency lighting in commercial buildings, but it was inexpensive and was perfect for the requirements of this project.



**Figure 32: Automatic Transfer Switch**

The overall design of the transfer switch was out of the scope of this project. We were simply adding it in for demonstrating purposes so the audience could visualize the switch between normal and emergency power and the effect it had on the performance of the system. Shown below was figure 33 and it was the typical wiring diagram for normal transfer switch operation.



**Figure 33: ATS wiring schematic**

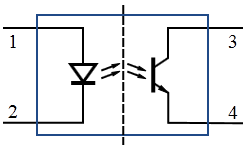
**Permission requested from Bodine.com**

The block that says “Emergency Ckt Brk Panel” would be our emergency power that we bring to the transfer switch from the wall receptacle. A normal extension cord with (3) 12 gauge wires would be used which meet NEC standards. The block that says “Normal Ckt Brk Panel” would be our normal power that also comes from a receptacle using the same type of extension cord. The last block that says “Lighting Load” would serve as the feed to our “house panel”. If this were an actual scale model this would be the connection to your houses main electrical panel. We would use a multiple outlet power strip to serve as our “house panel” and our power strip modules would plug directly into the multiple power outlet strip. This would give the user the overall concept of the home power system using a small-scale model.

Referring back to Figure --- you could three terminal located on the right side of the automatic transfer switch and they were labeled 5, 6, and 7. These terminals were used for low voltage controlling of the automatic transfer switch. For example if this transfer switch were used in a commercial application where low voltage switching was utilized the signal from the switch would connect to terminals 5,6, and 7 on the transfer switch. The actual wiring configuration depends on what you were trying to achieve with the switching. We would be utilizing these terminals to connect a wireless transceiver to them. After reading through the user’s manual of the transfer switch we found that during normal power operation terminals 5 and 6 would measure 12 volts between the two. When normal power was lost and the transfer switch switches over to emergency power the 12 volts between terminals 5 and 6 was lost. This feature would serve as the interrupt to our touch screen controller to inform the program whether it was operating in normal or emergency conditions.

**2.10 Optical isolator**

An optical isolator was basically a transfer device. It was generally used in electronic circuits to secure the more expensive equipment like micro controllers in the expense of its own life. This usually was not a big deal since an optical isolator was a cheap component. The protection it provided was against voltage and current spikes which would harm expensive equipment. As seen in the schematic representation below it was formed of two parts with a gap in between. In the first part there was a light emitting diode and in the second part there was a photo sensor separated from each other with a gap. Basically this gap was what secured the expensive equipment from the surges since there was no conducting element between the LED and the photo sensing diode. The way it functions was once there was electrical input signal LED lights up. This light can’t be seen from outside because it happened inside the component in a dark environment. This light was received by the photo sensor and was converted to a proportional electrical energy. If there were a voltage spike only LED would be harmed and the other side of the optical isolator would remain unharmed. Usually opto-isolators shown in figure 34 could withstand 10kV though without breaking down and secure the circuit. Another characteristic property of optical isolator would be its being a half way rectifier.



**Figure 34: Schematic of an Optical Isolator**

**Public domain no permission needed.**

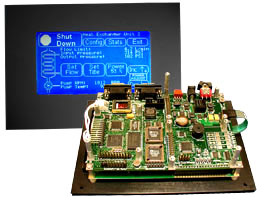
While designing the voltage sensing circuit, it was the first time us as a group got introduced to this component. In every forum we read about wiring microcontroller in an electronics circuit experienced electronics designer engineers were advising the usage of an optical isolator to protect the heart of the circuit; the micro-controller. After the voltage divider we decided that it would be wise to put an optical isolator since this circuit was not limited to the Hall voltage as it was in the current measuring circuit but it was connected to the main grid where it was vulnerable to rapid voltage spikes.

**Section 3: Design**

**3.1 Touch Screen Monitor**

After careful considerations of the needs and goals of this project we found that the QScreen Controller tm from Mosaic Industries, Inc. best suited those requirements. The QScreen Controller tm was a combination of a C-programmable single-board computer with a touch screen operated graphical user interface. This touch screen monitor was powered by the Motorola 68hc11 microcontroller, which we were very familiar with from the laboratory section of Embedded Systems class. The fact the each of the group members was familiar with the C-programming language as well as the assembly language of the Motorola 68hc11 was what lead us to selecting this product for our project. [1]

Because this project was a proto-type of product that would be utilized in consumer household, appearance and size were of great importance and concern. We envisioned this touch screen controller to be mounted near the houses existing thermostat so for cosmetic purposes we want our display to be about the same size as a standard thermostat. After taking measurements of a few thermostats we found the average to be 4” tall and 6” wide. The QScreen controllers overall dimensions were 4.125” tall and 6” wide making it the ideal candidate for size requirements. A picture of the overall appearance and design of the QScreen controller could be seen below in Figure 35. As you could tell from the figure below the QScreen controller would surface mount perfect in a household wall with a cutout not much bigger then needed for a typical switch or receptacle outlet box.

****

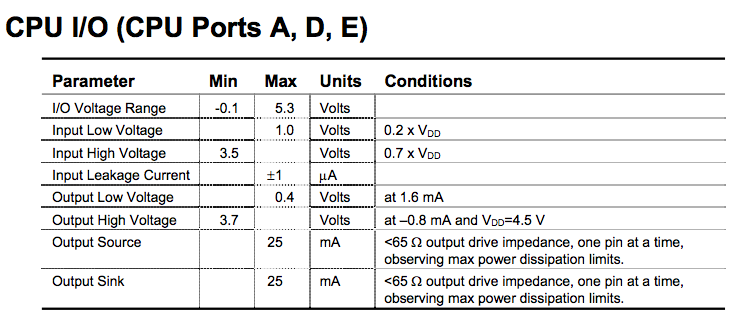
**Figure 35: QScreen Controller**

**Permission requested from Moasic Industries**

The QScreen controller was equipped with a touchscreen operated graphical user interface on a high-contrast 128X240 pixel display with a 5X4 touchscreen overlay. The screen itself measures 4.8” diagonally and was a high contrast CCFL white-on-blue monochrome LCD. The user did have the ability to control backlight and contrast of the screen using software provided with the product. Because of the ability to navigate through different menus this size screen was more then enough to satisfy the requirements of this project.

As noted earlier the main reason for using the QScreen controller was having previous knowledge of the Motorola 68hc11 as well as the C programming language. The QScreen controller gives the programmer the option of programming in either Control C or QED-Forth. We chose to use Control C as our programming language for the programming of this project. Control C tm cross-compiler comes with the purchase of the QScreen controller was written by Fabius Software Systems and customized by Mosaic Industries to facilitate programming the QScreen controller in C. It was a full ANSI C compiler and macro pre-processor. The compiler supports floating-point math, structures and unions, and allows you to program familiar C syntax. Programming was easy done using an interactive debugger and multitasking executive. The controller was pre-loaded with hardware control routines including drawing functions for the display. After programming in ANSI C by compiling our code on our PC and downloading the code to the QScreen Controller it would then be automatically executed. The real time operating system and onboard flash memory manages all required initializations and auto-starts our application code.

The Qscreen controller was not only our graphical interface to the home user, but it must also have been able to receive signals from multiple power strip modules through wireless technologies. The controller must then take the received data and process the information so it could be accurately displayed on the screen to the home user. This task was easy done using the dozens of real time analog and digital I/O lines on the QScreen’s Motorola 68hc11 microcontroller. The 68hc11 controls eight 8-bit A/D lines, 8 digital I/O lines including timer-controlled and PWM channels, and two RS232/485 ports. Two of these analog inputs and outputs would be utilized for the wireless RF transmission of the data being sent and received. It was important to verify compatible voltages on the input and output pins of each device in figure 36 show below.



**Figure 36: CPU I/O**

**Permission requested from Moasic Industries**

One of the most impressive features of the Qscreen was the pre-coded I/O drivers provided for all I/O, and this makes it easy to do data acquisition, pulse width modulation, motor control, frequency measurement, data analysis, analog control, PID control, and communications. Most likely we would utilize some of the analog inputs to receive wireless transmissions from a TI C2500 transceiver/receiver. The processor was more than powerful enough for the application of our project and was a good selection because the project could easily have options added on in the future.

**3.1.1 Display Layout and Design**

Now that we had selected the touch screen monitor for the project we need to design the functionality and appear of the interface to the user. We wanted to avoid a busy screen and keep it simple and easy for the user to read. Multiple pages would be avoided so as not to lose the user in navigation throughout the program. A preliminary touch screen output could be seen below in figure 37. As you could see the legend indicated that a boxed area with the dotted background would be the only input area for the user. These areas included adding a module, turning modules on and off, and also the ability to input the power rating of their home generator.

**Figure 37: LCD**

ADD MODULE

MODULE #

1

2

ON/OFF

CURRENT KW

AVERAGEKW

DAILY COST

MONTHLY COST

GENERATOR KW RATING

KW

CURRENT GENERATOR CONSUMPTION

REMAINING GENERATOR POWER

KW

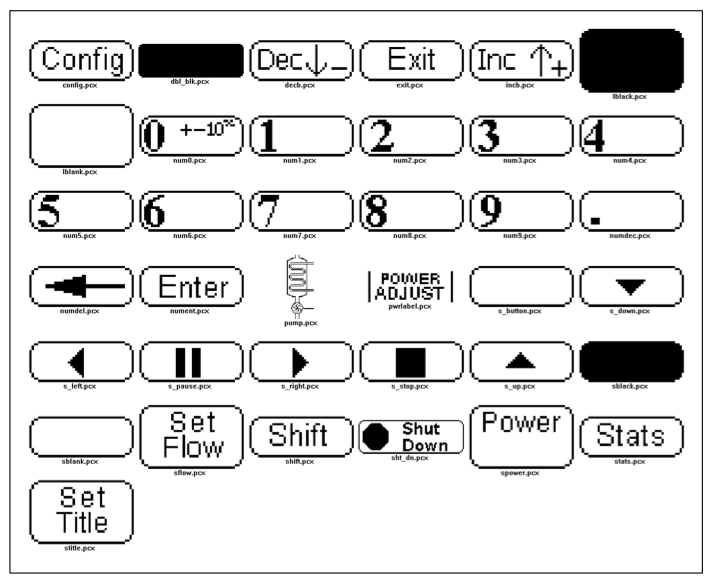
KW

We want to make sure the product was expandable to the user, which was why we included the “Add Module” feature. This allows the user to make the initial investment of the touch screen controller and maybe one or two power strip modules to test out and use in their house. At any time in the future they could always purchase another power strip module and connect it to their existing network. All the user needed to do was plug in the power strip module wherever they would like to use it and then simply walk to the touchscreen monitor and click “Add Module”. This would automatically recognize the power strip module just plugged in and would assign to it the next number in the module list. It was important to note that if the user wished to add multiple power strip modules at the same time that they would have to plug each one in separately and click the “Add Module” button before plugging in the next. The reason for this was when the user clicked “Add Module” the program was going to be looking for new devices and if more than one was available an error might occur.

The next feature the user would be able to control was turning the power strip modules on and off. This option could be used during normal power operations, but was mainly intended to be used during times of emergency generator power. During normal power operation the user should have all the modules powered on and simply turn off their appliance how they normally do. The only advantage to using the on and off feature during this time was to conserve power on appliance that draw power even when they were off such as televisions, computer, etc. This way you could shut the power off to them via the touch screen without having to unplug each power cord from the wall. The main purpose of the on and off feature was used when the system was in emergency mode. As soon as normal power was lost and the transfer switch switches over to emergency mode the QScreen controller would automatically shut off all power strip modules and wait for the user to decide what modules they want on. By looking at the “average power consumption” column the user could identify normal instantaneous power draw from individual appliances measured in KW. The QScreen knows the KW rating of the generator since the user initially input its value. As the user turns on individual power strip modules the program would calculate the remaining available generator power and report it back to the screen. This way the user would be able realize when they were approaching the limitations of the generator. Safety features would be added to insure the user does not accidentally turn on an appliance that would overload the generator causing failure.

The user would also have the ability to enter the KW rating of their home generator. This was important for the “emergency mode” operation of project. The software would monitor the usage of power from the generator as well as all the power strip modules and would compare to make sure the power consumption of the all the power strip modules together does not exceed the rating of the generator. If this were to happen the generator would fail and the project would have not satisfied its goal.

The Qscreen was pre-loaded with GUI development software to help aid in the design of the Qscreen output. Most programmers began with sketches of the screen and eventually worked their way into programming once the design look was achieved. The LCD device drivers provided 2 methods for writing graphics objects to the screen. Graphics may either be placed in the graphics array stored in RAM or sent directly to the LCD’s RAM. The designer had the ability to create their own symbols and graphics, but they may have also used the generic symbols pre-loaded with the GUI software. Figure 38 below showed these available pre-loaded symbols.

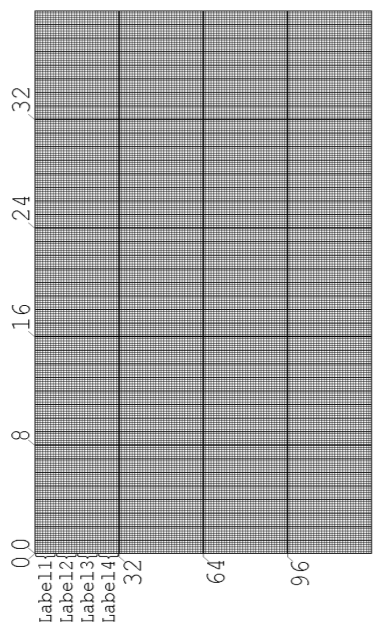


**Figure 38: Preloaded controller graphics**

**Permission Requested from Moasic Industries**

When graphics data was placed in the graphics array the Update\_Graphics function must be called to make the modified graphics array appear on the LCD. This technique was faster than drawing many different graphics and buttons individually to the screen directly. Menus and static graphics were displayed this way. This was the standard method of placing graphics on the screen.

Direct drawing to the LCD was useful when you only want tochange a small portion of the screen. When drawing only a small number of objects covering a small part of the screen, direct drawing was faster than using the graphics array and the Update\_Graphics function since only the affected portion was updated. Button presses use direct drawing since the buttons were single objects and occupy a small part of the screen. This provides a much faster response than having to draw to the graphics array and then update the entire display. Figure 39 shown below was supplied with the purchase of the Qscreen and was used as in aid in the design of the GUI interface.



**Figure 39: LCD geometry**

**Permission requested from Moasic Industries**

**3.1.2 Programming**

The Qscreen controller could be programmed in C or FORTRAN programming languages. All of our group members were familiar with the C programming language so we would utilize it in the software design of this project.

To begin the first piece of software that needs to be written deals with recognizing power strip modules throughout the home network. When a user plugs in a module and goes back to the touchscreen to hit the “Add Module” function the software would poll the network to search for new devices. This could be achieved by programming the wireless transceiver in the microcontroller to send out a pre-defined signal indicating it was new to the network. Upon finding the new module the software in the touch screen controller would assign it an address and then proceed with normal operations. To realize this feature a more in-depth study of the communications protocol of the TI C2500 must be done as well as the A/D conversions in the main microcontroller. These studies were currently underway and involve a cumbersome trial and error approach, as none of us were computer engineering or computer science majors. We were confident however that we would be able to implement the programming in this project over the next few months.

After power strip modules had been added to home network the QScreen controller was now ready to take samples from all of them and analyze, store, and display the result to the touch screen. In order to accomplish this we were going to have to receive packets of information in sequence from each of the power strip modules. At this time we were unsure of the bit size of each of the packets sent individually, but regardless of the size of the packet the concept remained the same. The QScreen would mostly be receiving information from the power strip modules, but it also must be able to send a signal to either turn on or off the relay controlling the receptacles in each individual power string monitor. To achieve this every time information was received from a power strip module the program would determine whether or not the on/off button was pressed. If the button was pressed the program would tell the controller to send a signal to the module turning on or off the relay and then would proceed to receiving information from the next power strip module. If the on/off button wasn’t pressed the program would simply move on to receiving information from the next power strip module. Below in figure 40 was a flow chart showing how the software would execute receiving/transmitting information from all of the power string modules.

STORE DATA IN MEMORY

QSCREEN

RECEIVE MEASUREMENT

R

CALCULATIONS/UPDATE SCREEN

CHECK TOUCHSCREEN FOR INTERUPTS

NO

MOVE TO NEXT MODULE

YES

EXECUTE ROUTINE

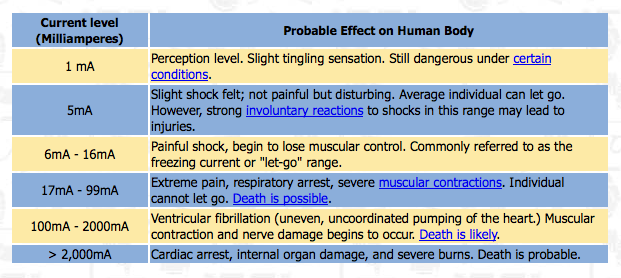
**Figure 40: Program flow chart**

It was very important that the design of the touchscreen controller includes a battery backup because all of the information from each individual power strip was store in its memory. A sudden lose in power could potentially wipe an entire month’s worth of data.

**3.2 Power Measuring and Circuits**

Measuring the electrical power of a given circuit was the most critical and fundamental concept of this project. Not only were we concerned with allowing the user to view an accurate power consumption and cost analysis, but we were also using power consumption analysis to determine the safe operating loads of a home backup generator.

There were many different methods to measure the electrical power of a given circuit. For instance the Hall Effect sensor used a current perpendicular to a magnetic field to measure voltage, current, and power levels. Another way was to use voltage response measurements where the electrical characteristics of a circuit were determined from the amplitude and phase of a test current flowing through the circuit. To measure power as stated previously we needed a sensor to measure the phase difference between the voltage and the current. It was important for safety purposes for us to consider the voltage and current magnitudes of the circuits to be measured. As shown below in figure 41 a very small amount of current could be fatal if short-circuited across the body. This problem demonstrates the advantages of using the Hall Effect sensor where the output voltage of the sensor was much smaller in proportion to the input signal therefore protecting the user from a potentially deadly shock.



**Figure 41: Shock hazard safety chart**

**Permission requested from** [**www.osha.gov**](http://www.osha.gov)

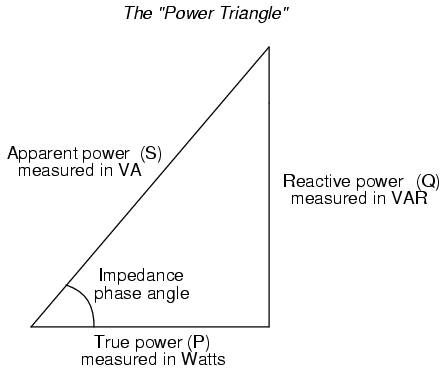
Our power strip modules as well as the module that monitors the generators power output would utilize the same power measuring circuit. For us to measure and calculate the output power of a given device we need to measure and record the voltage and current waveforms of that device. The voltage and current waveforms coming from the utility grid were analog signals, but our microcontroller used for data management and calculations only deals with digital signals. To overcome this problem would be using analog to digital convertors.

Our application requires one measurement device to capture the voltage across the terminals of a load, and the second device to capture the current going through the load. However, the actual power calculation depends on the resistive and reactive components in the circuit. In order to have an accurate power measurement, we find it necessary to study the electric power a little bit more and define its different components. In an alternating current power system, energy storage elements such as inductance and capacitance introduce periodic reversals of the direction of energy flow. The portion of power flow that averaged over a complete cycle of the AC waveform, results in net transfer of energy in one direction was known as real power. On the other hand, the portion of power flow due to stored energy, which returns to the source in each cycle, was known as reactive power. The utility company does not charge residential customers for the reactive power drawn from the circuit because this energy was conserved and was returned to the source. As shown below in figure 42 alternating current power systems consist of three types of power.

|  |  |  |
| --- | --- | --- |
| NAME | UNITS | EQUATION |
| Real Power (P) | Watts (W) |  |
| Reactive Power (Q) | Volt-Amp-Reactance (VAR) |  |
| Apparent Power (S) | Volt-Amp (VA) |  |

**Figure 42: Power types table**

Real power was the measurement of a circuit’s dissipative elements such as incandescent lights, ovens, hot water heaters, etc. The units of measurement for real power were watts and were represented by the symbol (P). Since the power company only charges residential customers for the usage of real power this would be the main focus of our power measurements. As noted above in table 42 the equation that governs real power is: . We know that (V) was for volts and unless otherwise noted should always be considered to be the RMS voltage of the given input. The symbol (I) was for current and also should be considered to be the RMS value unless otherwise noted. The symbol was known as the impedance angle and was the representation of the amount of reactive and real power present in a circuit. Figure 43 shown below was a representation of the power triangle and the relationships between the different units of power. Once the impedance angle was known the amount of real power in a circuit could easily be derived with basic trigonometric functions and this was where the cosine function comes into the equation. [4]



**Figure 43: Power triangle**

The cosine of the impedance angle was directly proportional to the amount of real power in a circuit, which was called power factor PF. The power factor was the cosine of the phase angle between voltage and current. A power factor of 1 represents a purely resistive load and power factor less than 1 represents a reactive circuit. Power factor was also defined as the ratio of true or real power to apparent power. Power factor takes into account both the phase and wave-shape contributions to the difference between true and apparent power. In our case, both the voltage and the current waveforms were sinusoidal. Therefore, to measure the power factor all we have to do was read the time difference between zero crossings of the waveform.

The power company wants a unity power factor because if the power factor was less then unity then the power company had to supply more current for the same load. Low power factor was caused by inductive loads such as transformers, electric motors, and high-intensity discharge lighting, which were a major portion of the power consumed in household. Unlike resistive loads that create heat by consuming kilowatts, inductive loads require the current to create a magnetic field, and the magnetic field produces the desired work. The total or apparent power required by an inductive device was a composite of the real power and the reactive power discussed above. In order for our power measurement to be within tolerance, we have to make sure that the error caused by the power factor was acceptable.

**3.2.1 Current Measuring Circuit**

Our group decision for the current measuring component for our design was **Hall Effect Current Sensor.** To reach this decision we evaluated many options. Among these options were the Current Shunt, Magneto Resistive Field Sensor, Hall Effect Current Sensor, Eddy Current Sensor and the Current Transformer. The Current Shunt was not a good option for several reasons. One was the heat problem since current going through a resistor in conducting wire produces voltage and this means at the same time there was a power and there was heat dissipation and this could be interpreted as the more the current the more the heat. Heat was something we don’t want in our circuit since it could harm everything around it starting from the shunt itself. Another reason why we cannot use current shunt in our circuit was that heat would change the resistance so the measurements wouldn’t be as accurate with the rise of the current since the resistance would be a different value and this wouldn’t serve project’s main goal was to measure power with accuracy. This accuracy may not be achieved if Eddy Current Sensor was our choice current measuring component neither. In our power measuring strip design, out of many components more or less all of them would have magnetic fields since they were all electrical instruments. These components would interrupt the precision of the Eddy Current Sensor. On the other hand; Hall Effect Current sensor looks way more superior in measuring the circuit task because of its advantages such as: High speed and repeatability highly linearity, a very long life and isolation from another high voltage in the same system, so it wouldn’t be affected by the other components just like Eddy Current sensor would. Current transformers were a no go as well since they were really good for higher voltage measurements but their sizes were usually larger than the our design would allow and they wouldn’t be as beneficial as neither magneto resistive field sensor nor hall effect current sensor. The superiority of two sensors above compared to current transformer would be summarized as: for Hall effect current sensor; logic capability, highly linearity, non mechanical structure, resistance against heat, perfect to measure high currents for safety concerns, were insensitive to dust, vibration, humidity, cold and hot and well sealed structure. On the other hand for magneto resistive field sensor advantages over current transformer would be: good accuracy and high sensitivity, durability, fast response rate, decent size, could be used in harsh conditions and reasonable price.

The two alternatives left, Magneto Resistive Field Sensor and Hall Effect current sensor were both good choices for our power measuring circuit design but our choice was towards Hall Effect Current Sensor. Besides the advantages they would bring to our power measuring circuit design reviewed in the above paragraph we had to go into detail to compare the two in order to make the best decision for our design. When two compared, we sow that Magneto resistive sensor had limited linear range, temperature drift and was sensitive to interfering magnetic fields from other components. Under these circumstances, our choice of going with Hall Effect current sensor was justified.

Below in figure 44 was HONEYWELL S&C CSLA2CD**;** our choice of Hall Effect Current Sensor as a part of the Power Measuring Circuit Blog Diagram:

POWER MEASURING CIRCUIT

CIRCUIT

OUTLET

SOURCE

SOLID STATE RELAY

MSP430

MICRO

CONTROLLER

**Figure 44: Power measuring circuit block diagram**

Earlier we were discussing that; in the design part for the solid state power relay, we decided that SSR should be the first component to be place closest to the outlet in our circuit design in order to be able to shut the power off whenever we want from the closest point to the source. The next component in the design was the Hall Effect current measuring sensor. Since the power measuring strip would have bunch of electrical devices plugged in to it each drawing different currents, our Hall Effect current sensor would be measuring the sum of these currents. Among many products our pick was HONEYWELL S&C CSLA2CD.

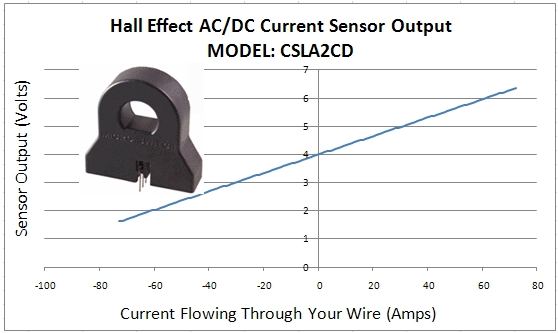
As mentioned earlier and could be seen in the blog diagram above Honeywell CSLA2CD Hall Effect current sensor was the first component of the power measuring circuit other than the Voltage measuring circuit. The specifications of the component were as follows:

* **Current Measuring Range AC & Dc** Up to 72A
* **Sensor Type**   Open Loop Linear
* **Sensed Current Type**   ac or dc
* **Sensed Current Range**   0 A to 72 A
* **Package Style**   PCB Bottom Mount
* **Output Type**   Voltage
* **Sensitivity**   32.7 mV N\* ± 3.0 @ 8 Vdc
* **Supply Current**   20 mA max.
* **Offset Voltage**   Vcc/2 ± 2 %
* **Supply Voltage**   5.4 Vdc to 13.2 Vdc
* **Offset Shift (%/ °C)**   ± 0.02
* **Response Time**   3 µs
* **Operating Temperature Range**   -25 °C to 85 °C [-13 °F to 185 °F]
* **Storage Temperature Range**   -40 °C to 100°C [-40 °F to 212 °F]
* **Housing Material**   PBT Polyester
* **Mounting**   PCB on 3 pins
* **Pinout Style**   3 pin

In order to power Hall Effect current sensor, we would need an additional circuit. This circuit had to be a DC circuit and be able to provide 5.4 to 13.2 VDC supply voltage to power the component. Estimated voltage consumption of the component should be around 8 VDC at 20 mA.

Right after measuring the current; Hall Effect current sensor would produce Hall Voltage proportional to the amount of current we were measuring. This voltage amount would be in AC form and be directed to the analog pin of MSP 430 in order to be a part of our power consumption calculation. But, of course compatibility with MSP 430 was imperative since the analog pin input had to be between 0 to 5 Volts. This means that our Hall voltage had to between 0 to 5 Volts so that MSP 430 could start the power calculation. A unique property of Hall Effect current sensors was the output voltage was very much related with the input supply voltage to power the component. This means that, whatever input voltage we chose between the allowed specifications, which were 5.4 to 13.2 VDC the output voltage should be proportional to it. This means that, according to how we design the supply voltage circuitry of the current measuring circuit we have to come up with a way to make it be compatible with MSP 430. An important point to pay attention here was we have to keep in mind that there would be other electrical components that would need DC supply voltage as well such as Solid state relay and op-amps, so it would be wise to have a 12 VDC and distribute it accordingly to necessary components by parallel circuitry with suitable voltage dividers.

Let’s look at Honeywell CSLA2CD linear current sensors output voltage characteristics shown in figure 45 according to its supply voltage and then discuss possible design scenarios. The measured current to voltage output relation of Honeywell CSLA2CD linear current sensor was shown below (Assumed supply voltage was 8VDC):



**Figure 45: Hall Effect sensor output**

**Permission requested from** [**http://scienceshareware.com**](http://scienceshareware.com)

As seen in the figure above when current was “0” sensor outputs 4V. This was exactly half of the assumed 8 VDC supply voltage. Another important thing to know about the component was every 1 A increase in current would cause 0.033 Volts increase in Hall Voltage. This means that in order to calculate the current the function we have to program into MSP 430 is:

**MEASURED CURRENT = [VOUT  - 4.0] / 0.033**

Since we started assuming the supply voltage as 8VDC lets base the first design scenario to this set up. From the outlet we should expect max delivery of 15 Amps but to be safe lets do our calculations based on 20 A. If we were to draw 20 amps, then according to the formula above the Vout which would enter the MSP 430 analog pin should be [( 20 \* 0.033) + 4.0] = 4.66 V which was within 5 V rate that MSP 430 has. In this scenario if the supply voltage was 9 VDC then final Vout would be 5.16 V and this would be above so it wouldn’t be compatible with MSP 430. The other optional scenario would be a higher supply voltage let say 12 VDC. This time Vout turns out to be 6.66 V for the highest possible current measurement for 20 A even if it was 12 A it would be 6.4 V still too high for MSP 430 to read. In this case we would have to build a op-amp circuit with a gain less than 1 so the voltage would be regulated. An inverting op-amp would be the right decision so the output would be a negative voltage and we could add some offset to it to bring it above 0 volts level. To give an example if the gain of the op-amp circuit was -0.5, an input Hall voltage of 6.66 V with maximum current possible would be -3.33 V and min current of 0 would give 6 V which would end up as -3.0 V in the output of the op-amp. If we were to give 5 V offset to this result it would be between 2 and 1.67. Since the op-amp choice was an inverting one the higher the current the lower the Vout. In this case 0 A current would give 2 V and 20 A would give 1.67 V both within the MSP 430 limits.

Out of the two possible scenarios of design the first one seems to be easier to do, so we decided to go with the first option.

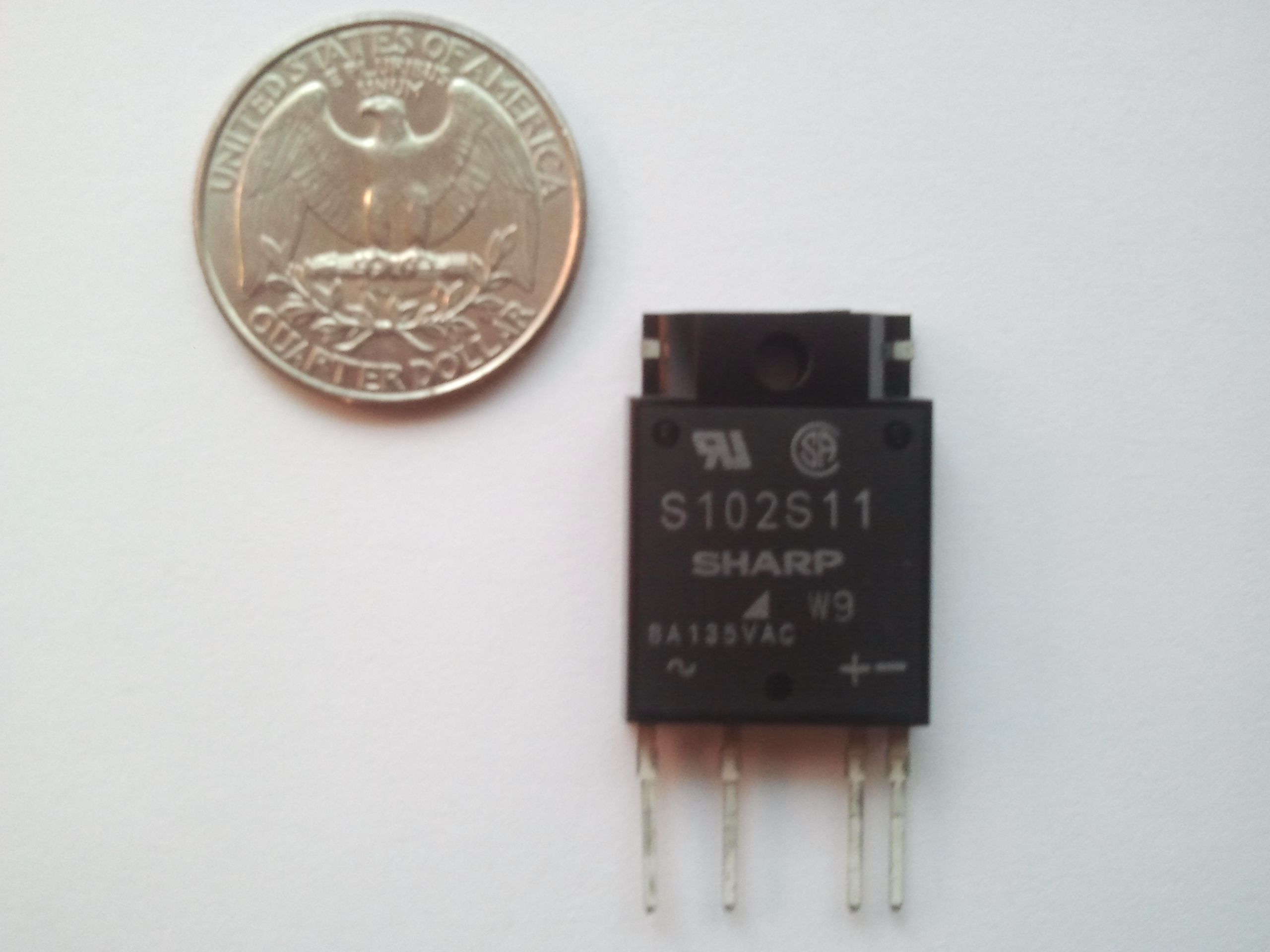
**3.2.2 Voltage Measuring Circuit**

This part was way easier to design compared to the other parts. Although it was called the voltage measuring circuit, we already know that it was 120 VAC (RMS). The point of this circuit design was to represent the voltage sine wave to MSP 430 through its analog pin, so that we could have a power calculation. As we could recall from the previous design this analog pin was rated 5 V so this means some work had to be done to get 120 VAC to 5V. As a matter of the fact we just have to know about the sine wave of the voltage since we were interested in the “0” crossings of the sine wave. The reason for this as explained before was our power formula; P = V \* I \* cosƟ, and if we recall in order to find “Ɵ”, we need to know the time difference between the current and voltage waves, so we have to look at the “0” crossings of both waves. While designing the voltage sensing circuit, after the voltage divider it would be wise to put an optical isolator for two good reasons. The first reason is; since this circuit was not limited to the Hall voltage as in the current measuring circuit, but connected directly to the grid there might be some voltage spikes and these would harm the micro controller and opto-isolator would prevent that. Second of all it was easier to be able to tell the zero crossing point of the voltage wave with this component. Once we know the voltage signal’s zero crossing time and from the previous current sensing circuit we know the current sine-wave’s zero crossing time, MSP 430 could calculate “Ɵ” and could apply it to; P = V \* I \* cosƟ formula to find the real power.

**3.3 Power Relay Design**

Since our system design was all about managing power and for this reason for us to manage the power we have to have control over it and this means for us to be able to turn an outlet on and off which consumes too much power. In order to this we need a switch which we could control remotely and the electrical switch component in circuits was called the power relay. There were two main types mechanical and Solid State Relays as mentioned before in the research part. Our research resulted with the group decision of going with the Solid State Relays.

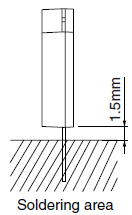
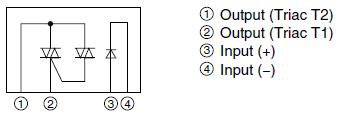
Solid State Relays were semiconductor based switches and unlike electro mechanical relays, they don’t have moving parts which means they don’t need maintenance unlike changing of the coil was had to be done in contactor relays. This makes them very reliable with their life time. Solid State Relays were known for their quick responses and dependable long life and there was no internal arching that would harm any semi-conductors. Another advantage we would consider was its PCB compatibility both because of its small size and better technology. Again solid state relay have the benefit of zero crossing switching which reduces noise in the as a result the circuit could experience switching where the voltage crosses were zero. Our group decision was going with the Photo-Coupled Solid State Relay and our pick**, Sharp S0116S01 Solid State Relay** was shown in figure 46 below.



**Figure 46: SHARP S116S01 SERIES**

Here were some specs about our choice Sharp S0116S01 Solid State Relay shown in figure 47:

* Input Forward Current (IF) = 50Ma
* Input Reverse Voltage (VR) = 6 V
* Output RMS ON-state current (IT(rms)) = 16 A
* Output Peak one cycle surge current (Isurge) = 160 A
* Output Repetitive peak OFF-state voltage (VDRM) = 400 V
* Output Non-Repetitiv peak OFF-state voltage (VDSM) = 600 V
* Output Critical rate of rise of ON-state current (dI/dt) = 50 A/µs
* Output Operating frequency (f) = 45 ~ 65 Hz
* Isolation voltage (VISO(RMS)) = 4.0 kV
* Operating temperature (TOPR) = -25 `~ +100 °C
* Storage temperature (TSTG) = -30 ~ +100 °C
* Soldering temperature (TSOL) = 260 °C



**Figure 47: Internal connection diagram**

**Permission requested from** [**WWW.media.digikey.com**](http://WWW.media.digikey.com)

As later would be shown in the main schematics of the whole power measuring circuit design, we were going to place the solid state relay at the entrance of the circuit where the current was received from the outlet. The reason for this is; since this was the entry point and in case of over usage of power we have to cut the power off from the closest point to the source which was the outlet. This needs a control unit to monitor the power usage and display it to the user and let the user to decide to shut the power off. In order to do this, micro-controller wirelessly connected to a LCD display was planned to be used. Once the read form the controller was received by and if the user makes the decision to turn the power strip off, the user interaction from the LCD screen wirelessly transferred to the microcontroller and from the microcontroller signal goes to the solid state relay which was connected to one of microcontrollers I/O pins. The compatibility of Sharp S0116S01 Solid State Relay with MSP 430 through I/O pin works out perfectly since both MSP 430 I/O pin output and Sharp S0116S01 Solid State Relay’s supply current were the same: 50 mA DC current.

The most important characteristic of the component after the compatibility with MSP 430, was the output current ceiling As we have discussed in the power analysis of the power measuring circuit the outlet had a 20A current rating and an outlet at a North American house hold should be expected to have 12-15 Amps. This was usually 12 for houses and 15 A for the commercial buildings ready to be drawn. Since our choice; Sharp S0116S01 Solid State Relay had a rating of 16 Amps we ensure that there wouldn’t be any problems with the drawn current because 12 to 15 Amps were below 16 which was within the limit.

Another important point we paid attention was the normally open state and the number of contacts of the solid state relay. Sharp S0116S01 Solid State Relay was normally open and had four contacts which in other words could be described as poles. Normally open state would let the current be drawn from the main grid under normal conditions unless an input signal was received from the microcontroller to shut the current off in case off an power overage which was done manually from the LCD screen by the user.

Block diagram for centering solid state power relay was shown below in figure 48, thick arrows represent the current being measured while thin black ones were the data signals:

SOLID STATE RELAY

Sharp S0116S01

MSP-430

MICRO

CONTROLLER

OUTLET

SOURCE

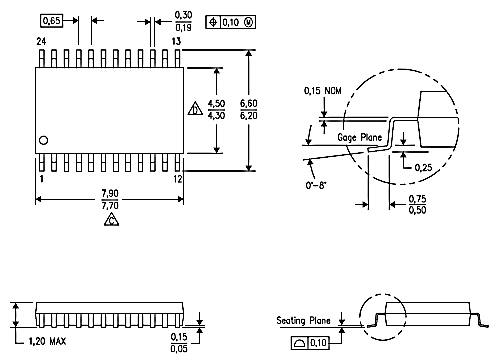
**Figure 48: Blog Diagram For SSR**

The four poles were designed for perfect adaptation for a printed circuit board, it was very easy to solder and the soldering instructions were shown in figure --. As mentioned before there were four poles two control input ports and two output ports. One of the input ports would receive the connection coming from the MSP 430 and the other input port should be grounded. The other left 2 ports which were the output ports would be used to connect the component in series to the coming circuit just at the entrance from the outlet, in other words current that would go to the power measuring components should enter the component from one output and leave from the other one. This way if the user wants to disconnect the power going into the power strip this would be achieved.

**3.4 MSP430**

After careful consideration of all factors with each of the microprocessors above, our group decided to use the MSP430AFE series from Texas Instruments as our microprocessor in the power strip module. This particular series of microcontrollers was specifically designed for the purposes of utility metering and uses a reduced-instruction set computing (RISC) architecture, meaning that there was less time between an input and the implementation of that command, taking into account computations calculated in between. Other benefits for using the MSP430 were its extreme low power requirements, especially when in sleep modes, and the wide range of peripherals from TI for electricity metering.

The MSP430AFE253 was the exact microprocessor that would be used in the power strip portion of our project. “MSP” stands for “Mixed-Signal Processor”; 430 was the MCU platform; F stands for Flash-based memory; E means it was optimized for e-meter applications and end-equipment; 2 was the 2nd generation, and 53 was the series in the 2nd generation. The MSP430AFE253 was 7.9mm long, the body was 4.5mm wide (with the length of the pins added, it was 6.6mm wide). Each pin was 0.3mm wide, and with 0.65mm spacing in between each pin. The height of the MSP430 was a maximum of 1.2mm. The dimensions were shown below pictorially in Figure 49. The top left drawing was a top view; the top-right drawing was a magnified view of a single pin; the bottom-left drawing was a side view of the entire MCU; and the bottom-right drawing was a side view rotated 90o.



**Figure 49: MSP430AFE253 dimensions**

**Permission requested from www.ti.com**

Available with this microcontroller was a software suite with a built-in library of functions to more easily and quickly manipulate data for a variety of purposes. This library contains a large amount of code support to perform many of the basic calculations related to and required for energy metering. These values include those found in figure 50, shown below.

|  |  |
| --- | --- |
| **Base** | **Specific** |
| Voltage | RMS  Peak |
| Current | RMS  Peak |
| Energy | Reactive  Active  Apparent |
| Power Factor |  |
| Temperature |  |
| Frequency |  |
| Tamper detection |  |

**Figure 50: Quantities, MSP430**

**Information from** [**www.ti.com**](http://www.ti.com)

Another library available from TI was the Capacitive Touch Library, which would help in enabling the capacitive touch capabilities for our MSP430. In the case that our group chooses to use a touch screen LCD hooked up directly to an MSP430, this library of code would be very helpful in helping us to learn the environment as well as write the necessary code for our project. Several references and guide manuals were also on TI’s website for free download which help guide first-time users through coding a touch screen interface with the MSP430. The code for the MSP430 could be written and developed within the free environment of a program called Code Composer Studio. This program was free for students, and provides all the support and features needed to code, test, debug, and run our project using the MSP430.

Another important aspect of the microcontroller was its limitations and recommended operating conditions. Our group would need to be aware of these values especially from the PCB implementation stage and through the final testing, but also when initially designing the circuit diagrams and board layout. First, the range of temperatures allowed for the device was hardly restrictive, and should not be a cause for concern for our group; once programmed, the device cannot be at a temperature lower than -40oC or higher than 85oC. Of greater concern were the maximum and minimum voltage and current values for each part of the MCU, as well as the processor frequency range. Values below were recommended values, not absolute extremes.

Supply voltage, Vcc: 1.8V – 3.6V

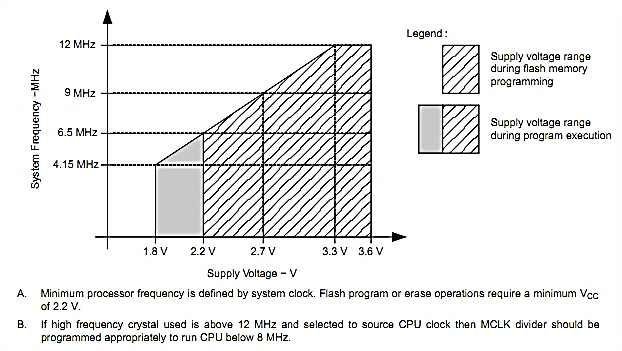
Supply voltage, Vss: 2.2V – 3.6V

fsystem: Vcc=1.8V 4.15 MHz

Vcc=2.7V 9 MHz

Vcc >=3.3V 12 MHz

Additionally, the recommended range of system frequency (fsystem) changes throughout the programming process, as well as during normal running operations. If possible, the supply voltage should not exceed 3.3 V during execution of the program, as shown below in Figure 51.



1. **Minimum processor frequency was defined by system clock. Flash program or erase operations require a minimum Vcc of 2.2 V.**
2. **If high frequency crystal used was above 12 Mhz and selected to source CPU clock then MCLK divider should be programmed appropriately to run CPU below 8 MHz.**

**Figure 51: Frequency range for MSP430**

**Permission requested from** [**www.ti.com**](http://www.ti.com)

The clock system of the MSP430 was designed to allow for some degree of flexibility. By choosing one of the three clocks and setting the clock speed as low as possible, the power consumption could be reduced. The first clock was the *LFXTCLK* (Low Frequency Crystal Clock), which connects to the XIN and XOUT pins and had an oscillation of 32kHz. The second clock, the *XT2CLK* (Crystal 2 Clock), connected to the XT2IN and XT2OUT pins, was intended to be the high-speed clock source. Finally, the third clock, the *DCOCLK* (Digitally Controlled Oscillator Clock), was an internally generated clock source that runs at 900kHz (which could be reduced internal using clock dividers). Essentially, when an ultralow-power stand-by mode was desired, the low-frequency auxiliary clock (LFXTCLK) should be used; for high performance signal processing, the high-speed master clock (XT2CLK) should be used.

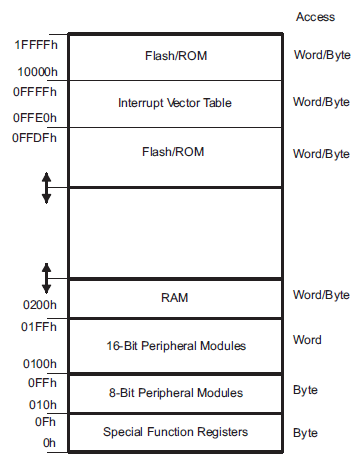
The interrupt vectors, which each contain the corresponding 16-bit address of the code for the interrupt instruction, were located from address 0FFE016 to 0FFFF16. In this range was also the power-up starting addresses. These addresses should be known to our group and were shown below in Figure 52.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INTERRUPT SOURCE** | **INTERRUPT FLAG** | **SYSTEM INTERRUPT** | **WORD ADDRESS** | **PRIORITY** |
| Power-up  External reset  Watchdog  Flash memory | WDTIFG  KEYV | Reset | 0FFFE16 | 15 (highest) |
| NMI  Oscillator fault  Flash memory access violation | NMIFG  OFIFG  ACCVIFG | Non-maskable | 0FFFE16 | 14 |
| Watchdog timer | WDTIFG | Maskable | 0FFF416 | 10 |
| Timer\_A3 | TACCR0 CCIFG | Maskable | 0FFF216 | 9 |
| Timer\_A3 | TACCR1 and TACCR2  CCIFGs, TAIFG | Maskable | 0FFF016 | 8 |
| USART0 receive | URXIFG0 | Maskable | 0FFEE16 | 7 |
| USART0 transit | UTXIFG0 | Maskable | 0FFEC16 | 6 |
| ADC10 | ADC10IFG | Maskable | 0FFEA16 | 5 |
| I/O Port P2 | P2IFG.0 to P2IFG.7 | Maskable | 0FFE616 | 3 |
| I/O Port P1 | P1IFG.0 to P1IFG.7 | Maskable | 0FFE416 | 2 |

**Figure 52: Interrupt address locations**

**Reproduced with permission requested from www.ti.com**

The flash memory in the MSP430 could be programmed in one of two ways: using the in-system process by the CPU, or via the Spy-Bi-Wire/JTAG port. Most locations could be erased one at a time, or the entire 0 to n locations (512 bytes each) could be erased in one step. Words were organized with the low byte placed at an even address and the high byte placed at the next highest odd address (e.g. LB=000416 and HB=000516). As shown below in Figure 53, there were separate address spaces for different types of memory and registers. Important to us were the Flash/ROM memory spaces (1000016-1FFFF16), the Interrupt Vector Table (0FFE016-0FFFF16), and the Special Function Registers (016-0F16). Also, any 16-bit peripheral modules were mapped into the address space 01016 to 0FF16. Code could only be accessed on even addresses, and data may be accessed as words or bytes. [8]

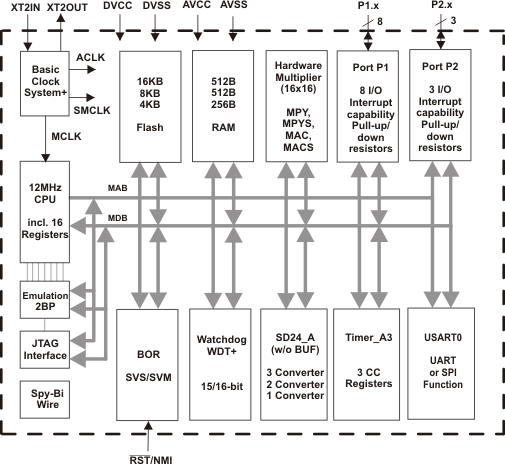


**Figure 53: Address space of MSP430**

**Permission requested from www.ti.com**

**3.4.1 Programming the MSP430**

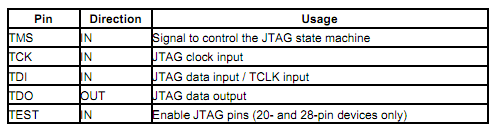
Before writing code for the microcontroller, it was helpful to try and understand the physical architecture of the chip as well as the interconnections between each of the pins and internal devices. The entire functional block diagram was shown below in Figure 54. First, the clock system had an input port (XT2IN) and an output port (XT2OUT), both of which double as digital I/O pins. From here, the clock was of course communicating to the CPU. Next, the 8-wide digital I/O port P1.x communicates with the Timer\_A3 as well as the main CPU, via MDB.



**Figure 54: MSP430AFE253 Function Block Diagram**

**Permission requested from** [**www.ti.com**](http://www.ti.com)

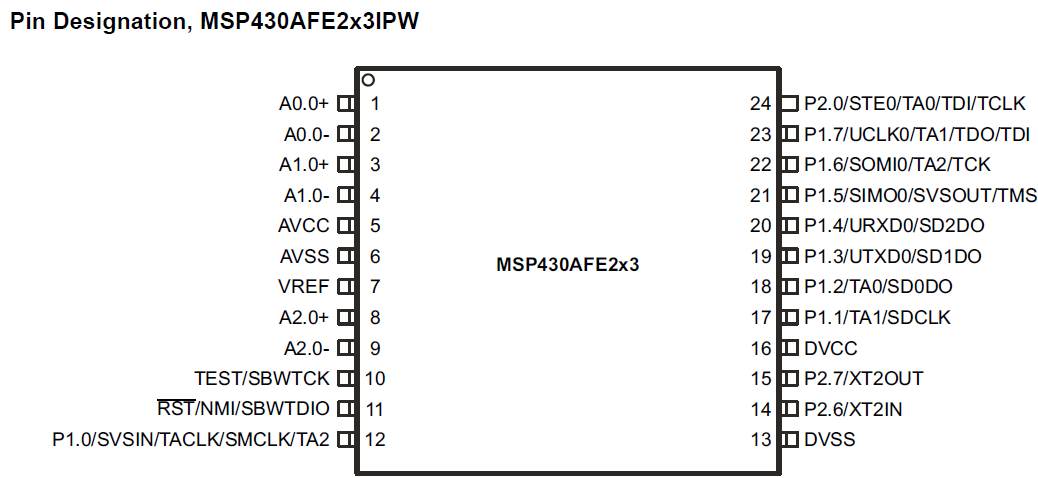
The MSP430 could be programmed in one of two ways: using the in-system CPU process, or externally using the JTAG/Spy-Bi-Wire Port, which uses flash memory. Programming through the JTAG interface requires four input signals and one output signal, in addition to ground and Vcc, if powered externally. The input pins were TMS, TCK, TDI, and TEST, and the output was TDO; the pins were summarized below in Figure 55, along with their purposes.



**Figure 55: JTAG Interface Signals**

**Permission requested from** [**www.ti.com**](http://www.ti.com)

The terminals and I/O pins and their locations on the microcontroller were shown below in Figure 56. As seen below, the two analog input pins (A0 and A1) were located in the upper left corner of the microcontroller, with the analog supply voltage located directly below them, and a third analog input pin located beneath that. The higher pin in each pair was the positive terminal (pins 1, 3, 5, and 8), and the lower pin was the negative terminal (pins 2, 4, 6, and 9). The last three pins on the left hand side of the microcontroller were TEST, (RST)’, and the first general purpose I/O pin (pins 10, 11, and 12). The right-hand side of the microcontroller consists primarily of general-purpose digital I/O pins (pins 14, 15, and 17-24), in addition to a digital supply voltage (pins 13 and 16).



**Figure 56: MSP430AFE23 Pin Designation**

**Permission requested from** [**www.ti.com**](http://www.ti.com)

In addition to a detailed diagram of the exact locations of each pin on the MPS430, a useful tool would be a chart detailing the purpose, functionality, and I/O status of each pin. This diagram was shown below, in Figure 57.

|  |  |  |  |
| --- | --- | --- | --- |
| **Terminal name** | **Pin number** | **I/O capability** | **Description** |
| A0.0+ | 1 | I | Positive analog input (A0.0) |
| A0.0- | 2 | I | Negative analog input (A0.0) |
| A1.0+ | 3 | I | Positive analog input (A1.0) |
| A1.0- | 4 | I | Negative analog input (A1.0) |
| AVCC | 5 | N/A | Analog supply voltage, positive terminal. **NOTE: Must not power up prior to DVCC.** |
| AVSS | 6 | N/A | Analog supply voltage, negative terminal. |
| VREF | 7 | I/O | Input for an external reference voltage/ output for internal reference voltage. |
| A2.0+ | 8 | I | Positive analog input (A2.0) |
| A2.0- | 9 | I | Negative analog input (A2.0) |
| TEST/ SBWTCK | 10 | I | Selects test mode for JTAG pins on P1.5 to P1.7 and P2.0. |
| (RST)’/NMI/ SBWTDIO | 11 | I | Reset or nonmaskable interrupt input. |
| P1.0/SVSIN/ TACLK/ SMCLK/TA2 | 12 | I/O | General-purpose digital I/O pin. |
| DVSS | 13 | N/A | Digital supply voltage, negative terminal. |
| P2.6/XT2IN | 14 | I/O | Input terminal of crystal oscillator. General-purpose digital I/O pin. |
| P2.7/XT2OUT | 15 | I/O | Output terminal of crystal oscillator. General-purpose digital I/O pin. |
| DVCC | 16 | N/A | Digital supply voltage, positive terminal. |
| P1.1-1.7, 2.0 | 17-24 | I/O | General-purpose digital I/O pin. |
|  |  |  |  |

**Figure 57: Terminal functions of MSP430**

**Reproduced from** [**www.ti.com**](http://www.ti.com)**, permission requested.**

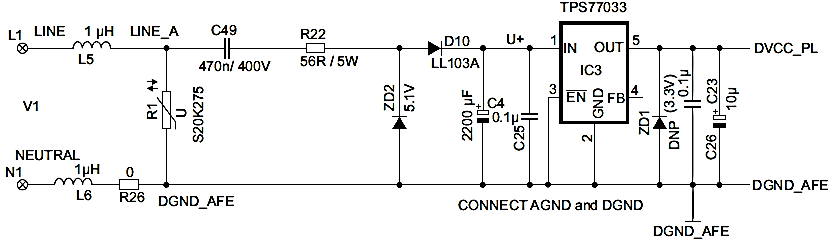
One of the benefits of using this microprocessor was the existence of two all-purpose digital I/O ports. Port P1 includes pins 12 and 17-23, and was 8 bits; port P2 was 3 bits, and encompasses pins 24, 14, and 15 (both in that order). Properties of these include:

* Each of the 11 I/O pins was independently programmable, allowing for any combination of input, output, and interrupt conditions.
* Edge-selectable interrupt was possible on all 11 bits.
* Each pin had an individually programmable pull-up/pull-down resistor.
* Finally, the unused bits in port P2 (i.e. bits [5:1]) were read as 0, and any write instructions to these bits were ignored.

Several other features of the MSP430AFE253 were worth mentioning at this time. First, the MSP430 had a watchdog timer (WDT+) which was intended to perform a controlled system restart after a software problem occurs. The time interval chosen for this function could be chosen by the user. In addition, if the WDT+ was not needed in a program, it may be used as an “interval timer”, which could provide a system interrupt at predetermined periods in time. This function may be used by our group to periodically (though at regular intervals) provide an interrupt to the current and voltage measuring system in the power strip to force the system to send the data to the microcontroller in the LCD.

A second feature was the brownout circuit/supply voltage supervisor. The brownout circuit was used to provide a reset signal during power on and power off periods. The SVS (supply voltage supervisor) circuit was used as a voltage detection system. If the supply voltage to the microcontroller drops below a predetermined level (set by the user), this was detected by the SVS. In addition, the circuitry here was capable of performing both automatic resets and non-automatic resets of the system. Only after the brownout circuit releases the device would the CPU begin code execution.

Available on the website for the MSP430 was a PDF file which details the implementation of a single-phase electronic watt-hour meter using the MSP430AFE2xx series of microcontrollers. This documentation was a valuable resource for our group and would be used as a cross-checking for our group at various stages throughout the months to ensure that our plan for implementation was viable. Part of this report includes details on the hardware implementation used in the watt-hour meter, including the two power supplies used. The first was a resistor capacitor power supply, shown below in Figure 58. The circuit uses a series of devices at the input in an effort to reduce the dc voltage provided to the MSP430 to the appropriate level of 3.3V for full-speed operation. Additional help and support, as well as the design equations for the power supply, were shown in the ‘Capacitor Power Supplies’ section of *MSP430 Family Mixed-Signal Microcontroller Application Reports*. The left-hand side of the diagram comes from a 110/220 VRMS alternating current supply source, and the right-hand side goes directly into the digital supply port and the digital ground port of the MSP430AFE.

**Figure 58: MSP430 Energy Meter Capacitive Power Supply**

**Permission requested from** [**www.ti.com**](http://www.ti.com)

A second power supply circuit was constructed to ensure that enough power and current would be available to power RF transceivers, which our project would also use. Although the microcontroller used in the paper from TI was different than that which our group would be using, we would anticipate similar issues with providing sufficient power to the RF transceiver. This power supply would provide 3.3V (DC) from the 110 or 220 V (AC) in order to power the transceiver properly. [8]

**3.4.2 Code Composer Studio**

This section would detail the process for programming the MSP430, and would include all pertinent information related to programming. The programming of the MSP430 could be done entirely in the C language, and was done in the Code Composer Studio (CCSv4) environment. CCS was a whole suite of tools including compilers for each of the MCUs from TI, a debugger, source code editor, project build environment, and simulators, and had been made available for free to the engineering students at the University of Central Florida. To begin the coding process, our group needed to become familiar with the programming process for the MSP430G2231 microprocessor included in with the Launchpad development kit. These step-by-step instructions would be used every time that our group begins to code a new section of our project; the only difference would be the microprocessor that would be chosen (see step 2 below), which would be the MSP430AFE253. The process was detailed below:

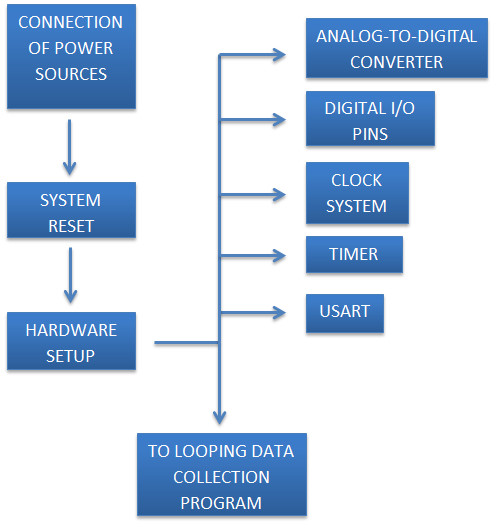
1. Open Code Composer Studio and select OK to choose the default workspace.
2. Choose “MSP430” as the Project Type, then after two pages, select MSP430G2231 (MSP430AFE253 later on) as the Device Variant.
3. At this point, you could open a new project by selecting File->New->New Project.
4. The window main.c was where the code could be written. You could also link an existing c file to your program by navigating Project->Link Files to Active Project, and then navigating to the file(s) needed. This was useful in linking files which contain the regimen for calculating voltage values, multiplying two numbers, or a number of other small operations which could be referenced.

The coding process involves using many of the functions provided with the software to manipulate the data flowing into the MSP430. The following functions, shown in Figure 59 below, would likely be used in one way or another; this list would be used as a quick reference over the next several months while our group edits and debugs our code.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function name** | **Action** | **Bits of answer?** | **Use in project** |
| dc\_filter | Filters the DC content from an AC mains waveform signal. | Same as input | Filters out DC content of Voltage and Current for Power calculations, leaving only AC. |
| debounce | Used to debounce a push button switch, i.e. eliminates errors associated with pushing a button to activate a clock edge. | N/A | Used for testing and debugging the circuit (i.e. a manual clock edge). |
| div\_sh48 | Shifts a 48 bit integer upwards by a specified amount, and then divides a 16 bit integer into the 48 bit integer. | 32 | Combines two operations into one to increase efficiency and decrease time spent. Useful for dot product calculations. |
| div48 | Divides a 16 bit integer into a 48 bit integer. | 32 | For dot product calculations. |
| imul16 | Completes a 16x16 2’s complement multiplication. | 32 2’s complement | Multiplication of Current and Voltage. |
| isqrt32 | Computes the square root of a 32 bit number, where the last 16 bits were after the decimal point. NOTE: be careful not to have a negative number as input. | 16 |  |
| mul48 | Computes a 32x16 multiplication function. | 32 |  |
| q1\_15\_mul | Computes a 16x16 multiplication function. | 32 |  |

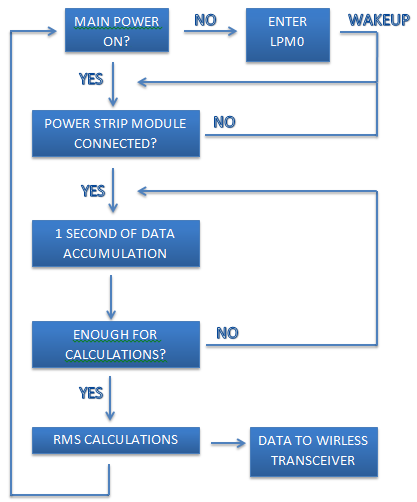
**Figure 59: Table of functions for MSP430 programming**

The process for our data collection code was detailed below in the flow charts in Figures 001 and 002 below. Our code begins with the connection of the power sources to our microprocessor, then an initial system reset. The next step was the hardware setup, including: analog-to-digital converter, digital general-use I/O pins, Clock system, timer, and Universal serial asynchronous receiver/transmitter (USART). This initial process was shown in Figure 60.



**Figure 60: Initial system setup flowchart**

Next, the system checks if the main power was on. If not, the device would go into LPM0, and then wakeup. With the main power on, the system would check for a signal from the power strip module to ensure proper connectivity. Once this was passed, the energy accumulation phase could begin as seen in figure 61. First, a small amount of time (perhaps 1s) of data was accumulated. Then, a check in the system was made to see if enough data had been gathered for calculations of current, voltage, power, etc. If not, the system returns to the previous step and gathers another second’s worth of data. When a sufficient amount of information was stored, the system would move to the next step and calculate the RMS values of the parameters needed. These would be stored and then sent to the wireless transceiver through the SPI/UART (Serial Peripheral Interface/Universal Asynchronous Receiver/Transmitter). After transmission of the data, the system would return to the beginning of the program (but after the initial setup of the clock, port pins, etc) and perform a system check for the main power. The process would repeat indefinitely until the power was disconnected or an interrupt signal was received from the hub microprocessor in the LCD.



**Figure 61: Looping data collection/transmission process**

**3.5 Wireless Design**

The wireless design that follows shows a visual representation of the data flow between devices. The CC2500 retrieves input from the main MSP430 controller and transfers it to transceiver on the power strip. This power strip only had to send data out; it does not take in any data. The only inputs were the awake and control signals. The awake signal takes the MSP430 out of sleep mode, when it receives this signal was also transfers all of its data measurements to the CC2500 and then returns to sleep mode. The control lines shown in figure 62 send a word of information containing the information to turn devices on or off. The control signal also takes the MSP430 out of its hibernation state. It was important to note that the transceiver does not send any data regarding the sleep mode. This was done locally through the MCU software.

**Figure 61: Wireless Data Flow**

CC2500

CC2500

MAIN

MSP430

Control

LOCAL

MSP430

Power Strip

Data Out

Data Out

Data Out

Awake

Awake

Awake

Control

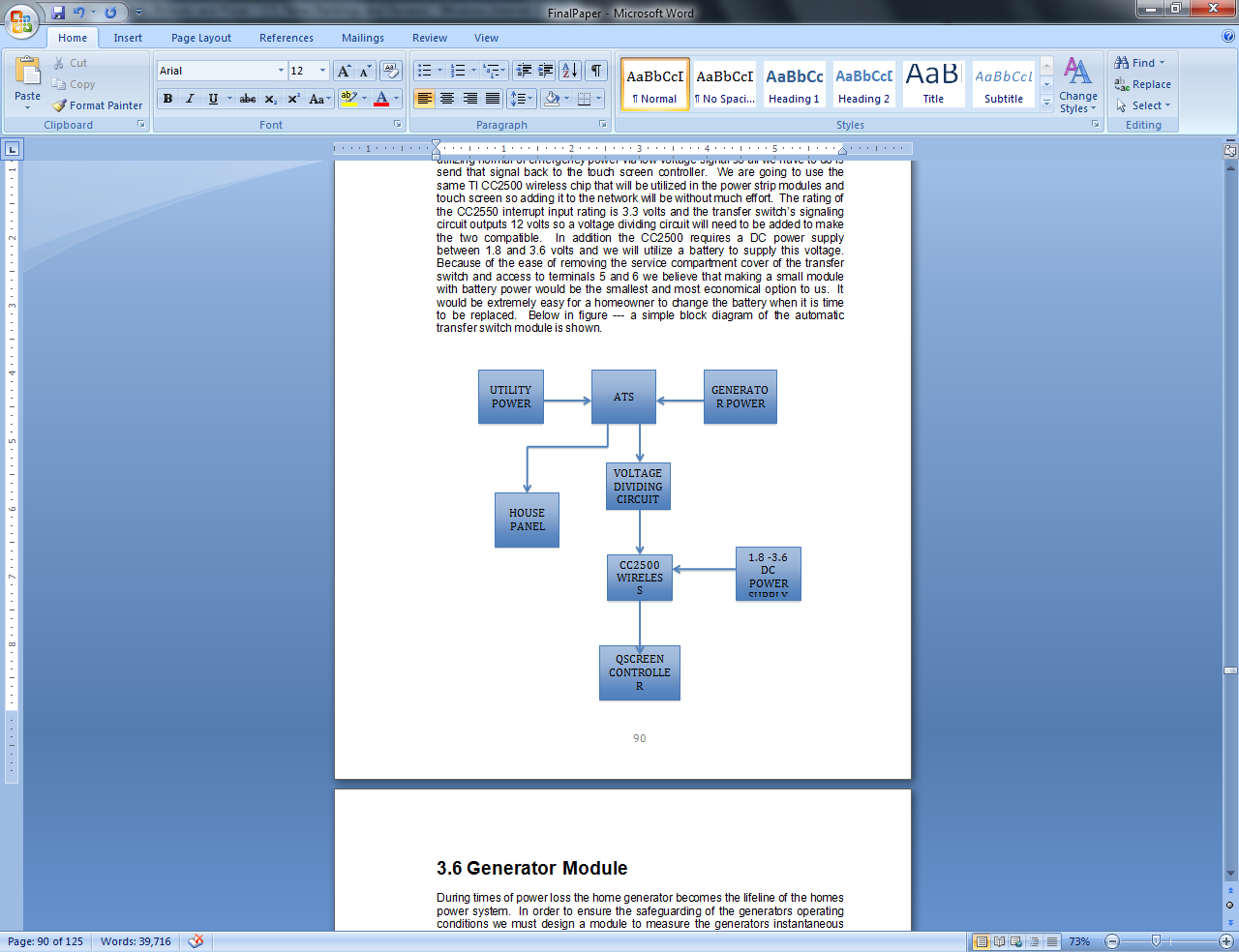
Control

Control

**3.5.1 ATS Wireless Module**

The automatic transfer switch must have the capability to wirelessly transmit whether it was in normal or emergency mode back to the touch screen controller. The automatic transfer switch does not come standard with this module so we would have to design it ourselves. The design for this module was rather simple and was not considered to be part of the “main design” of the project. The reason for the adding the automatic transfer switch to the project was for the purpose of demonstration. Without being able to simulate the transfer between normal and emergency power the project would not be able to show its usefulness. The project would simply look like any other home power monitoring system, which was not the intent of our groups design.

The automatic transfer switch already provides a means of signaling whether it was utilizing normal or emergency power via low voltage signal so all we have to do was send that signal back to the touch screen controller. We were going to use the same TI CC2500 wireless chip that would be utilized in the power strip modules and touch screen so adding it to the network would be without much effort. The rating of the CC2550 interrupt input rating was 3.3 volts and the transfer switch’s signaling circuit outputs 12 volts so a voltage dividing circuit would need to be added to make the two compatible. In addition the CC2500 requires a DC power supply between 1.8 and 3.6 volts and we would utilize a battery to supply this voltage. Because of the ease of removing the service compartment cover of the transfer switch and access to terminals 5 and 6 we believe that making a small module with battery power would be the smallest and most economical option to us. It would be extremely easy for a homeowner to change the battery when it was time to be replaced. Below in figure 62 a simple block diagram of the automatic transfer switch module was shown.



**Figure 62: ATS Block Diagram**

**3.6 Generator Module**

During times of power loss the home generator becomes the lifeline of the homes power system. In order to ensure the safeguarding of the generators operating conditions we must design a module to measure the generators instantaneous power consumption and have that information wirelessly transmitted back to the touch screen controller for program manipulations. The output from a typical home generator was 240 volts with a neutral. The ground was picked up from the houses existing grounding system.

The power measurement module for the generator would be slightly different than that of the power strip modules because we have two phases to monitor instead of one. This was not a problem though as we only need one additional power measuring circuit to accommodate for the extra phase. The total power of the generator was simply the addition of the two individual phases. The following equations govern the total power consumption of the generator.

Except with the addition of the extra power measuring circuit the main components remain the same as the power strip modules. First the power would be measured via current and voltage sensors and then power calculations would be done using the MSP 430. This information would then be wirelessly transmitted back to the touch screen controller for further processing and display to the user. A block diagram of the generator module could be seen below in figure 63.

GENERATOR

PHASE A

PHASE B

POWER MEASURING CICUIT

POWER MEASURING CICUIT

BATTERY POWER

MSP 430

CC2500 WIRELESS

QSREEN CONTROLLER

**Figure 63: Generator Module Diagram**

**3.6.1 Generator Power Measurements**

In case of a power shortage our ATS module was going to replace the power grid with the generator output. There was couple of slight differences between the regular power measurements and the generator power measurements. The generator power measuring circuit was going to be limited power output and there was need to shut it off because of over usage. Because we have to do a presentation of our project indoors and it would be loud to bring a generator indoors we would represent generator output with another outlet which we would limit its output on purpose.

Energy in a home type generator was produced in AC form since these types of generators were just a smaller version of the power plant generators. Usually the generator was a stand by generator which was connected to the house wiring with an ATS or it had a plug on it if it was a portable generator. Doesn’t matter which type it was sometimes it would be 240 VAC but most of the time we’ll receive in a regular house outlet, 120 with 12 amps of max current could be drawn. In a regular home there were different appliances with different uses most of which were resistive loads since they’re used somehow to produce heat like oven or water heater, besides resistive loads we have Inductive and capacitive loads as well which have magnetic field outcomes when connected to the current. . The same way as it was with the power grid, with generator the outcomes of these two different kinds of loads give different types of power which were “Real Power” and “Reactive Power”. Our concern was with the Real power the same way when it was with measuring the power from the main grid because it was the generic scale of power measurements, so our measurements would be based on Real Power again.

Since our generator would be producing Apparent power which was measured in VA ( S = P + jQ ) and as we mentioned before the types of power could be listed as Real power which was measured in Watts, Reactive power which was measured in VAR’s. Since the generic power measurement was Real power, then our power measuring circuit had to be measuring the real power for the generator as well. This brings us back to the power factor calculation or cosine of “Theta”. This was the angle between both the voltage and current values or the same way the same angle between apparent power and real power. According to the previously mentioned formula “P = V \* I \* cosƟ” where Theta was the phase angle all we need to do was multiply the values we get from measuring the circuit with cosine Theta and we find the power. Calculation of Theta from the voltage and current measuring circuits would be as follows: All we have to do was read the time difference between zero crossings of the waveform of both voltage and current waves. This difference was proportional to the whole cycle period. Since all these calculations would be done by our microcontroller “MSP 430”. and there was an allowed range of inputs to MSP 430 by implementing voltage divider circuit, to both current and voltage measuring we could make these values compatible with the microcontrollers pin input values.

This limited power output of generator’s would be measured with the exact same methods that were described in the previous sections, as a matter of the fact it would be the same power measuring strip that we designed to measure the power from the regular power grid.

Previously we discussed about how we would design power strips that we would plug into outlets and these power strips would measure the power consumption of appliances which were plugged into it and would report wirelessly to LCD display. Now for the generator output nothing had changed since ATS would switch to generator power and the power strip would measure the power consumption of the same appliances plugged into it and would report wirelessly to LCD display. Since these power measuring strips were plugged into outlets they don’t care which source was the provider of the power, they just measure it, doesn’t matter if it was the power from the main grid or the generator.

Once the measured power was sent from MSP 430 to LCD through the wireless module, this was where things change a little bit. Now we do have a read reported wirelessly from the power strips but this time the source was not the grid but the generator. The only way for our main processor to be able to tell the difference of this change in the power sources was if it receives a message from the ATS module. As mentioned before ATS was the switch which would detect the shortage and switch to generator power. In this case either wirelessly or a hard wire from the ATS module to the main processor would solve this problem. This way our system would adjust to the emergency protocol and turn on only the imperative outlets, and turn off all the others in order not to overwhelm the generator. Another solution would be instead of having ATS and the main processor talk to each other we could solve the problem with software. This solution would be if the system overall power consumption falls under a certain limit, which would be usually the generators power rating we could get the system override the current status and get into the emergency status. Only problem with this solution would be the transition period would be overwhelming on the generator although it was for a short time, but it would save us plenty time from the design part since it only appears to be an if statement that we have to add into the code.

**3.7 Parts List**

The parts list below contains a list of all the components required for our project. Each appliance controller would consist of one CC2500 Transceiver, one MSPR430, an outlet, and a circuit board. These components would be housed separately. The LCD display and the automatic transfer switch were standalone devices and would be purchased individually. These two devices would not be designed; they were simply to be interfaced with the rest of the parts of the Power Safe system. The power metering circuit would also contain the same components as the appliance controller. In addition it would contain a relay that could support 30A of current delivered by the generator.

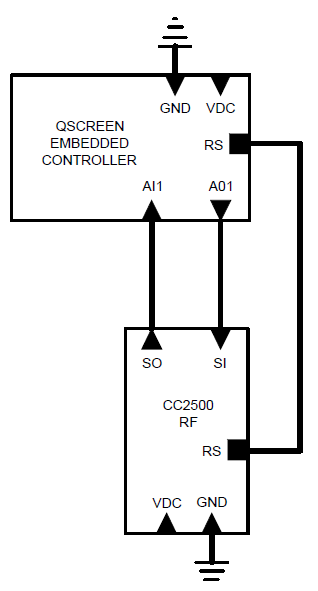
* 4 CC2500 Transceivers
* 1 LCD Display
* 1 Automatic Transfer Switch
* 2 Housings for outlet/controller
* 3 current transformers
* 4 MSP430F2274
* 5 circuit boards
* EagleCAD Software

|  |  |  |  |
| --- | --- | --- | --- |
| **Part Name** | **Quantity** | **Cost** | **Total** |
| Microcontroller Components | ~ | $10 | $20.00 |
| LCD Display | 1 | $400.00 | $400.00 |
| Automatic Transfer Switch | 1 | Free | Free |
| Outlet Housing | 1 | $10.00 | $10.00 |
| Current Transformer | 4 | $8.00 | $32.00 |
| MSP430F2274 | 4 | $2.75 | $10.00 |
| Initial PCB design | 1 | $33.00 | $33.00 |
| Circuit Board | 4 | $33.00 | $132.00 |
| Solid State Relay | 4 | $10.00 | $40.00 |
| EagleCAD Software | 1 | $49 | $49 |
| Power Outlet | 4 | $10 | $40 |
| Total |  |  | $766.00 |

**Figure 64: Parts Prices**

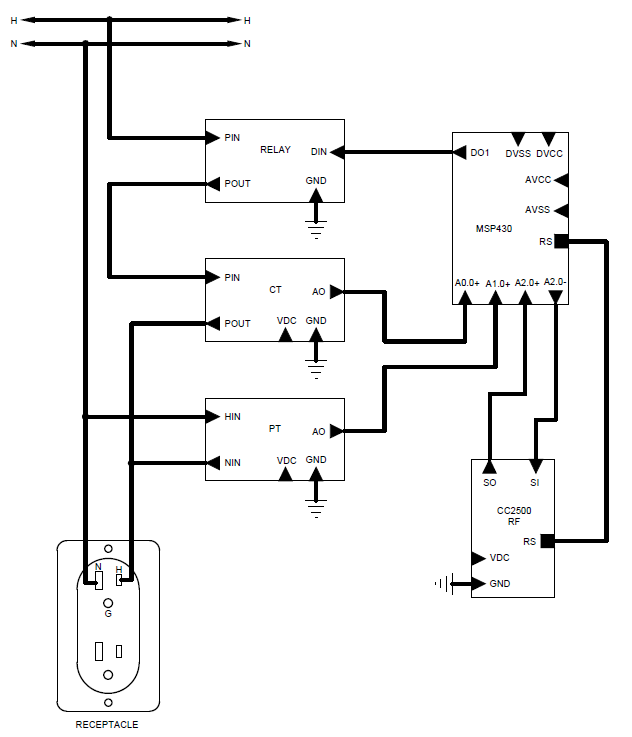
**3.8 Schematics**

The following three diagrams detail the overall schematic for our project. The first diagram, figure 65 shown below, was the connection of the TI CC2500 wireless transceiver to the QScreen controller utilizing pins 1 and 2 on Port E.

****

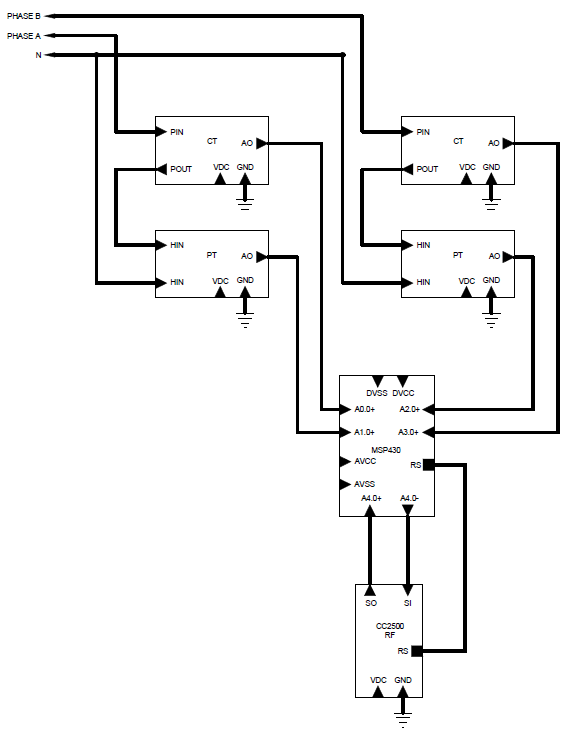
**Figure 65: QScreen Wireless Transceiver**

The next diagram, figure 66 below, was the complete schematic of: the current transformer, potential transformer, power relay, microcontroller and wireless transceiver. These items wired together control the duplex receptacle.



**Figure 66: Power strip module schematic**

The last schematic, below in figure 67, was the current and potential transformers measuring the current power consumption of the generator.



**Figure 67: Generator Module Schematic**

**Section 4: Testing**

**4.1 Procedures**

The software for the wireless transmission of data would be developed using the EZ430-RF2500 Development Tool. Through using the debugging software provided with this tool we would test to make sure the data being received by the transceiver was correct. The LCD panel would be tested along with one development board. Once we have proper communication between these two devices then the rest of the power strips could be added to test functionality between multiple devices. All of the initial testing would be done using breadboards and pre-assembled microcontrollers. We cannot proceed with the final testing design until all components were working as described.

Once this initial testing had been completed and was working to the correct specifications we would develop the first PCB. Once again we must test the PCB with our original design to confirm that no abnormalities have occurred. Once the first PCB was working we would develop the remaining boards to complete the project.

Following was the testing procedure we would apply for the project. This test procedure was developed to validate all of the steps covered under the requirements and specifications. Following successful completion of the testing procedure all of the individual parts of this project would be functional. The final testing procedure would test the final case in which all of the components were working together. Once this step was complete all of the components would be individually mounted on their own circuit boards. By isolating the initial testing procedure with the actual implementation of the printed circuit boards we would be able to verify that any abnormalities were not a design problem. Actual results for all testing procedures were yet to be determined.

**4.1.1 Power Sensing Circuit**

**Case Name: Case one – Power Sensing Circuit**



Unit tested: Current & Voltage Sensing Circuit

Assumptions: Test lamp had an expected power usage of 60 watts.

Test Data: Current, Voltage, and Power

Steps to be executed:

1. Plug in a regular 120V AC 60Watt desk lamp with an on/off switch into an outlet
2. Measure the current being drawn by the lamp with a multimeter
3. Calculate the total power consumption and compare the expected values
4. Plug in the same 60 Watt desk lamp into our circuit
5. Measure the output current and voltage from our circuit
6. Compare these values to the actual values from earlier. The values must be accurate to .1A and 1V.

Expected Result: The actual values measured from the unit should be close to the advertised power consumption of our lamp. The measured values would be calculated using the power formulas. The expected power consumed by the lamp was 60 Watts.

Summary: The power sensing circuit was simply a step down in voltage and current output. These outputs should be in the range of values that could be read by the MSP430 ADC10 pins. Given a voltage input of 8 volts and a resolution of 1024 each step would have an accuracy of around .008 volts. This means that with each step up or down of .008 volts we could output a new voltage reading from the power strip to the main control of the home.

**4.1.2 MSP430**

**Case Name: Case two – MCU**



Unit tested: MSP430 Microcontroller

Assumptions: Test lamp had an expected power usage of 60 watts.

Test Data: Current, Voltage, and Power

Steps to be executed:

1. Take the results of the measured values from test case 1. Calculate the actual power usage through the power triangle equations.
2. Physically wire the output of the power sensing circuit into the input of the MSP430 ADC10 pins.
3. Through the software read and display the individual current and voltage values of the power sensing circuit.
4. Calculate power measurement using MSP430 functions.
5. Verify that the values calculated from case 1 were equal to the function calculated values on the MSP430.
6. Enter sleeping mode
7. Leave sleeping mode when an interrupt was received

Expected Result: The results of test case two should be identical to test case 1. A power consumption reading approximately 60 Watts was expected. After the microcontroller sends an information packet it should enter sleeping mode until it receives a turn-on signal.

Summary: Each microcontroller would be tested individually before they were interfaced wirelessly. To meet our specifications each microcontroller would measure current from a controlled target device. Readings from each device would be measured to make sure they were accurate. Once all of the devices were individually tested then their data would be sent wirelessly to the main microcontroller. This information would be aggregated and then transmitted to be displayed on the LCD panel.

If all the devices were able to measure current and display these measurements to a screen then the next step would be to transmit a signal from the main microcontroller back to each power sensing microcontroller. The transmitted signal would need to accomplish three tasks. Each power strip containing an MCU must restrict power consumption which would be represented by a light. Another signal should resume the power consumption. If both of the tasks were complete then the MCU must also go to sleeping mode while there was no power being consumed by the power strip. The main microcontroller must have constant communication with all power strips in the home as well as the MCU measuring current on the secondary power supply.

**4.1.3 Wireless**

**Case Name: Case three – Wireless**

Unit tested: CC2500 Transceiver

Assumptions: The MSP430 MCU was able to read and output power consumption information to the transceiver from a given outlet.

Test Data: Hard coded data from software and power measurements given by the MSP430.

Steps to be executed:

1. Hard code data into the EZ430-RF2500 wireless Development Tool
2. Using this code send a data packet to a different transceiver
3. Read data from the transceiver and compare it to the original packet
4. Autonomously send measured data from one power strip to another. Each power strip contains one microcontroller and one transceiver.
5. Test low power consumption using wireless information. When a sleep mode signal was received the MSP430 should enter sleep mode. If a turn-on signal was received the MSP430 should send a second packet of information. The MSP430 should send a packet of information every 5 seconds.
6. Repeat steps 1 through 4 indoors with a line of sight range from 10ft to 100ft. Again test for this range between 2 to 3 rooms on the same floor and then between rooms from adjacent floors.

Expected Result: The transceiver should be able to send and receive accurate packet information between rooms.

Summary: The wireless modules would need to be checked first for functionality and then for range and reliability. Communication between two to three rooms at 50 feet was necessary. If adequate range was not met then considering different antenna options could increase the range. Each unit should be able to consistently receive and transfer data bytes. Once all power strips were functional then the design would be tested to find the limitations in range of the design. The open air tests would be used to determine to maximum range of our design. By adding walls in between the wireless communication we could see how much of an effect different materials have on the range of the system.

**Case Name: Case four – LCD**

Unit tested: QSscreen Controller

Assumptions: MSP430 was functional and could correctly send and receive data to other microcontrollers.

Test Data: Numbers generated by MSP430

Steps to be executed:

1. Cycle through different menu options by touching the LCD screen
2. Output total power consumption reading information from the MSP430 onto the screen
3. Send an interrupt control to the microcontroller for powering outlets on and off
4. Allow a user to input the current cost per kilowatt hour given by their local power company
5. Display current date and time
6. Cycle through each power strip to read individual power measurements
7. LCD should display an individual power strips current power consumption as well as total power consumption
8. Display total monthly cost for all power strips individually and summed together
9. Ability to store and compare the cost of up to 12 consecutive months of information
10. Calculate kilowatt hour conversions using computer software and compare the results to the displayed information on the LCD

Expected Result: LCD screen touch screen function would properly cycle through menus. All of the power reading information would be calculated locally by the main microcontroller. The LCD process should take this information as input and apply its own calculations to determine the monthly cost for the appliances. Power to kilowatt conversions should be determined by the LCD processor.

Summary: The first test for the LCD would be a basic touch test. This test consists of touching the screen to see if it responds correctly. Touch response should be speedy and accurate. While touching the screen all the different menus would need to be tested. This would ensure the boundaries of the buttons were correct and that the LCD screen does not have any hardware issues. The LCD unit also needs to be tested for numeric accuracy. Data received by the microcontroller needs to be displayed on the LCD correctly. All numbers would be displayed in real time. The LCD unit contains a local processor that should receive inputted cost per kilowatt hour information and output total cost for the household.

**4.1.4 Software**

**Case Name: Case five – Software**



Unit tested: Software

Assumptions: Microcontrollers could read information from the power sensing circuit.

Test Data: All data transmitted and received by the microcontrollers. This includes interrupts as well as voltage and current readings.

Steps to be executed:

1. Test the software with a single power strip module
2. Verify that interrupts were working as described
3. Output all numeric data to a screen and verify the numbers were correct
4. Plug in a second power strip and test the software for module priority

Expected Result: The code should display all data correctly and the priority routines should work as described. If a priority power strip exceeds the power distribution of the backup generator then it should turn off immediately.

Summary: Software would be tested throughout the testing process. Component specific tests would reveal any abnormalities with the software. After the home modeling system was complete a full function test of the code could be completed. The software must be tested to see if it was displaying the correct information. All of the math must be verified within the code so no measured values were incorrectly compared. Non essential features in the coding of the LCD would be tested last. The LCD should display all of the necessary information in the correct location. The code must not present any abnormalities when a user requests information from the LCD. The code should also be tested for speed and efficiency.

**4.1.5 Automatic Transfer Switch**

**Case Name: Case six – Automatic Transfer Switch**



Unit tested: ATS

Assumptions: The transfer switch was simply transferring power from one power supply to another.

Test Data: Current

Steps to be executed:

1. Interface the ATS with one primary 120V outlet
2. Read the output current from the ATS
3. Interface a second 120V outlet
4. Remove the primary power source from the outlet
5. Measure the current coming out of the ATS
6. If there was current after step 5 then the ATS was functional

Expected Result: The automatic transfer switch would autonomously switch power output from the primary to the secondary power supply once the primary power was lost. Power delivery from the ATS should not be interrupted.

Summary: The primary focus of the project was not on the switch. The automatic transfer switch was a self contained and individually manufactured part of our project. The only test necessary was to ensure that the device was functional. If the device was functional it could be implemented into the project.

**4.1.6 Scaled Test**

**Case Name: Case seven – Scaled Test**



Unit tested: All Units Simultaneously

Assumptions: All units have been tested individually and were working according to the specifications and requirements section of the project. Before this test was conducted steps 1 through 5 should receive a passing grade.

Test Data: Voltage, Current, Power from different appliances and different outlets simultaneously, digital interrupts would also be tested.

Steps to be executed:

1. Wire up all circuits and plug in the power source for the ATS
2. Turn on the generator and remove the primary source of power from the outlet.
3. Observe circuit for any abnormalities
4. Send interrupt and control inputs to all of the MS430 boards
5. Read information on the LCD
6. Cycle through all menus on the LCD and test all menu functions including the ability to set priority devices.
7. Plug appliances into each outlet to verify the power strips were functional and that the priority power strips remain on if power drain exceeds power delivery.

Expected Result: Circuit should behave normally and all components should work as they were individually tested before.

Summary: The scaled modeling of the home monitoring system consists of three power strips, one controlling station with an LCD, and a monitoring device on the secondary power supply. This was our final design implemented using breadboards and microcontrollers. After it passes this testing phase then the circuit would be implemented using PCB technology. The project would be demonstrated using two sources of power. One source would be directly connected to a conventional outlet supplying 120V of AC current. Another source would provide the system with a limited and varying amount of current. The secondary source would be represented by a small generator. By altering the current supplied different outlets would be shut off according to their priority. Outlets which have higher priority would remain turned on if there was sufficient power to supply them. All of the information regarding each outlet and the secondary power supply load would be displayed on the LCD. Users would have the ability to interact with the LCD to see the intended information displayed about each power strip. All functionalities would be tested with respect to the specifications and requirements. A visual representation of the final scaled test was given below. Communication data flow between all of the devices was show by the arrows.

**Section 5: Administrative Content**

**5.1 Timeline**

June 8, 2011

* Decide on the final version of the specifications of the project
* Finish and submit initial project and group identification document

June 15, 2011

* Research the parts needed for the hardware parts and microcontroller of the project
* Discuss and research the compatibility of these parts with each other along with their price, availability and delivery timing

June 22, 2011

* Organize parts list according to parts’ availability and vulnerability and add back up parts to this list and finalize the list
* Finalize parts list
* Order supplies

June 29, 2011

* Double check and confirm the delivery of the parts
* Start mastering the coding necessary for the micro controller
* Start the Senior design paper

July 6, 2011

* Start code draft implementation
* Discuss each group member’s progress on final paper on weekly meetings

July 13, 2011

* Discuss each group member’s progress on final paper on weekly meetings
* Half-way point where each member was expected to have written 15 pages
* Proofread, discuss, and edit each other’s work so far completed
* Prepare for the Team Presentation

July 20, 2011

* Prepare for the Team Presentation
* Discuss each group member’s progress on final paper on weekly meetings

July 27, 2011

* Team ready for Presentation
* Complete Senior Design paper
* Last editing of the senior design paper

August 3, 2011

* Submit senior Design paper
* Present the project to class

August 10, 2011

* Continued research over semester break
* PCB Eagle file reviewed and edited

August 17, 2011

* Possibly consider potential minor changes of parts for various reasons found in research over break
* Small segments of code should be tested on the MSP430 and LCD microprocessor
* Group growing to understand exact functionality of each part, as well as the parts as a whole

August 22, 2011

* Start the implementation of the first prototype
* Start debugging the code
* Check if the code was compatible with the actual microprocessor and the other parts
* Ask two professors this week and next to be on our group’s committee

August 29, 2011

* Continue writing/debugging code
* About half of the code should be written at this point
* Working on implementing pre-built functions into the code

September 12, 2011

* All parts should be in at this point
* Additionally, 2-3 back-up parts for each of the smaller, inexpensive units should be present, or ordered

September 19, 2011

* Subsections of project work independently (i.e. current/voltage sensing circuits, communication between microcontrollers, LCD showing functionality)
* Code experiencing minimal issues

September 26, 2011

* The halfway point for the project, most of the parts should be together
* Group members brainstorm about best way to present the project
* Code to be perfected
* Full prototype works as per initial specifications

October 14, 2011

* Project continued to be tested

October 31, 2011

* All parts connected and talking to each other
* Necessary minor adjustments for the code and hardware

November 7, 2011

* Final project was ready
* Testing phase
* Final paper and schematics review

November 14, 2011

* Statistical analysis of data
* Get ready for the presentation

November 21, 2011

* Final report typed and ready to present

December 1, 2011

* Group presentations this week

**5.2 Future Improvements**

The current project was focused heavily in the ability to give home owners a reliable source of electricity if there was a power failure. Once this part of the project was complete a number of other functions could be added. These functions should be geared toward helping a user save money on their energy bill. Since all of the applications in your home were being monitored then users could be given a status of their current consumption.

By using this information home owners would have the ability to control their own power usage. They would be able to set constraints on how much power they want to use per month. Users could be alerted to see if they were on track with their energy consumption for the month. An automated setting could even shut down low-priority appliances in order for the bill not to exceed a certain amount for that billing period. Meters usually sit outside of a home and home owners were not aware of their total consumption. They may even be billed incorrectly if they were not aware of how much power their home was consuming. The consumption statistics for different months or years may be compared with one another to see the history of power consumption for a home.

Currently when information about a month goes by it was lost. If all of the data for these months could be saved than a detailed report could be generated. This information could be available by subscription on a website. By introducing internet communication a user would have the ability to save statistics information and to control appliances from any remote location. Users would be able to monitor their power usage and set schedules for operating their appliances. This information would also be available from any mobile device. An application for your favorite smart phone would also give the ability for a user to have an in-home remote for their appliances.

Scheduling would also be an important energy saving application. If a user had a routine schedule he or she could set their air conditioning to turn off as they leave for work in the morning and resume functionality 1 hour before they return home.

The concept of our project was to have a house with almost every electricity-consuming appliance connected to one of our power strips. To make this a reality though we would have to design additional power strip outlets to accommodate many of the common receptacle configurations found in your household. The typical wall receptacle was a type 5-15R, but other receptacle such as the 6-15R, 10-30R, and 10-50R were commonly used for ranges, driers, and hot water heaters. This project was design to accommodate the 5-15R configurations and it would not work for the rest. The reason for the different configurations was some appliances need 120 volts, 240 volts with ground and no neutral, and also 240 volts with ground and a neutral. Each one of these situations alters the power equations needed to calculate power consumption and quality. This leaves the project at a great disadvantage because the biggest energy consuming appliances use these receptacle configurations that we have not designed to. A complete system would require more time and money and was unfortunately not in the scope of this project.

Another improvement that could be made lies within the power strip module. Our prototype design powers the microcontroller, sensors, and wireless communications via battery, which we would ideally like to have an AC to DC convertor in its place. Having a battery as the power source was inefficient and requires more maintenance then there needs to be. When the battery fails measurements and transmissions were stopped and the touch screen stops recording for that power strip module until a battery was replaced. In the mean time the user was no longer receiving an accurate monthly power cost analysis and the appliance must be unplugged from the power strip module and plugged back into the normal wall receptacle for it to work. This was a big inconvenience and was probably the biggest downside currently to the project.

The programming could also be improved for more efficiency and a faster executing time. Because all of the group members were electrical not computer engineering majors our programming skills were probably not up to par with the more experienced programmers out there. The more efficient your code is, the faster the program could execute and if there was less code to process you were also saving power consumption of the processor. Even though our code works and it executes as it should, there would always be areas that could be improved.

**Section 6: Design Summary**

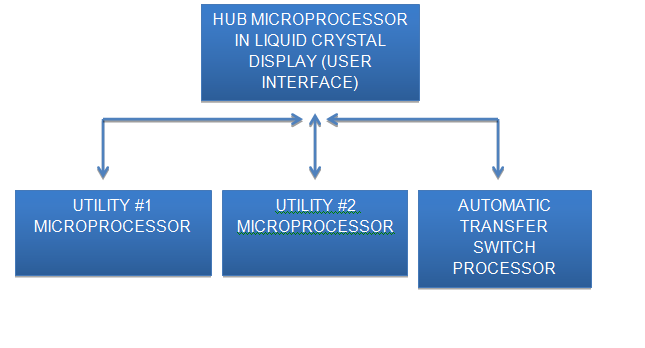
The last few pages of this document include a brief overview of the entire project. The projects intentions, goals, specifications, and design were reviewed and were presented in a compact format to give the reader the overall concept of what the project was about. First the inspiration for the project and reasons we believe it was worthy of taking up was discussed. Second the goals of the project and what we intend to accomplish would be looked at. Third the specifications and requirements would be reviewed. Last but not least we would present the overall design using schematics and narratives to further clarify any questions the ready might have.

**6.1 Intentions and Goals**

The intent of our project was to design a home energy management system. This system would be versatile in that it would be beneficial to the homeowner during both normal power operating conditions as well as during times of utility power. Our definition of “normal power operating conditions” would be when the homeowner was connected to a power source from the utility company. “Utility power” was defined by the entire house running off of a generator which alone does not provide an amount of power equal to what would be used from the power company on a daily basis. In fact, the utility power should be an amount that would severely limit the abilities of the homeowner to power the appliances in their home. This was an important distinction because much of our project was focused on the sometimes difficult choices that would need to be made during times of power outage. Often, the average homeowner was unaware of the many details and tremendous amount of power that go into allowing their lives to run as normal. Questions like, “Can I run my hot water heater on a generator?”, “How much power do I need to keep the refrigerator on?”, or “What were my limitations for powering my computer/television/modem during power failure?” would be simplified and answered in our design, while still giving the end-user as much control as possible.

The main design of our project begins with the loss of normal power, at which point the generator already connected to the user’s home would turn on in response to a signal from a microprocessor. That microprocessor would be constantly monitoring the current coming from the main, “normal”, power source; as soon as the current disappears, a signal was sent to the generator to turn on and take over power for the house. Our design would be a power strip module that resembles the surge protector power strips which could be used at one’s computer desk to plug in your computer and its peripherals. It would have two or three outlets to plug in any standard utilities. These power strip modules would have voltage and current sensing circuits which would be constantly taking measurements related to power consumption. The power strip modules would also contain contactor circuits to allow the user to remotely turn on and off the appliances through the LCD touch screen monitor. This information (current and voltage readings) would then be processed at a microcontroller (embedded in the power strip module), and then transmitted wirelessly to the main microprocessor in the LCD.

The wireless communications in our system would be done using a single protocol, which was yet to be determined. Shown below in Figure 68, each wireless device would be a transceiver (i.e. had the ability to transmit and receive data); they would not communicate with each other, but only to and from the central microprocessor (in the LCD). The wireless transceivers would be used over a short range in our prototype; this could be extended to the end-user as well, assuming the main processor in the end product was located near the center of the house. This aspect would allow for lower power usage, as well as lower costs for the transceivers.



**Figure 68: Wireless communications block diagram**

The LCD would contain the main “hub” microprocessor which would collect data from the power strip modules and was where the bulk of the calculations and decisions would be made. This display would be the interface between the user and the software which would control whether or not outlets in the power strip module were supplied with power. During normal operating conditions, the user would be able to control which outlets would be turned on (supplied with power) and also have the ability to turn off any or all of the outlets. There would also be a section on the LCD which would provide the user with pertinent data about their power consumption. In this section, an emphasis would be placed on “useful” data; that is, we don’t wish to give overwhelming amounts of graphs and numbers which may require knowledge of math to interpret and understand. The LCD would show information in as simple a form as possible, while still providing important, useful data. The data shown would be facts such as:

* Instantaneous power consumption
* Individual appliance power costs
* Daily power consumption
* Current rate of power consumption
* Total power used since the last power bill

An emphasis would be placed on showing as many dollar amounts as possible, so the user could more easily relate to the information on the screen. Abstract figures, graphs, and even most other data would be shown for the purposes of the demonstrations, and on a limited basis at that. This process was intended to help the user to more accurately monitor control their power usage, in order to meet their monthly power consumption goals.

**6.2 Requirements and Specifications**

The design would utilize an automatic transfer switch to transfer power to a secondary power supply in case the primary source fails. The automatic transfer switch was a third party device. It only needs to be tested for functionality. For this project the use of a generator to provide secondary power in case of an outage was intended. After power had been transferred the devices would be able to read load from the generator and communicate this information via wireless signal to a main microcontroller.

Other outlets in the home would also have power monitoring features. They must have the ability to both transmit and receive instructions or data. Each outlet would be required to read the current that was being drawn by that specific outlet. That information would be relayed to the main microcontroller which would sum all of the power being drawn from the home at that time. If too much power was being drawn with respect to the generator an interrupt signal could be sent to any of the microcontrollers that were in control of an appliance. These microcontrollers must stop the use of the appliance immediately upon request.

Each microcontroller would measure a specific current. The accuracy of these currents should be precise enough to prevent damage to any appliances or the generator. We would set our initial goal to reach a precision of 1mA. The main unit would receive power reading information from all the other microcontrollers every second. The main unit would provide absolute instructions to all other microcontrollers. It must have the ability to stop and resume power consumption of any outlet in the home. These criteria would be set by the user through an LCD input screen. In case of an overload of the secondary power source appliances would shut down with respect to their priority.

Outlets would be sorted by priority based on user input. Each outlet priority level would have a number ranging from 1-10 with 10 being the most important. In the event two outlets were given the same number the user would be asked which appliance was more important. Here the user could decide if they would rather have warm water or air conditioning. All of the priority level inputs would be given to the system by the user through the LCD panel. In case no priority was set the appliance draining the largest amount of electricity would be shut down.

The power strips must be able to achieve these specifications as part of our design:

* Send data every 5 seconds
* Enter sleep mode for 5 seconds in between packet transmissions.
* Measure currents up to 15 amps
* Measure voltages of 240V
* Up to 2 appliances could be connected to any power strip
* Have a line of sight range of at least 100 feet in an indoor environment
* Have a range of at least 50 feet between two rooms
* Power readings must be measured within an error margin of 5%.
* Must have an ADC that was 10 bits wide for 1024 steps of resolution

The Main control station would consist of a controller that receives information from all the power strips and displays this information on the user controlled LCD display. This station should be able to send and receive information from the LCD. If the LCD relays information to shut down a specific appliance an interrupt signal must be sent to relay that information.

The specifications for this unit were as follows:

* Receive data from 3 different power strips 5 seconds
* Receive data from power metering device two times per second
* Communicate this data to the LCD module 3 seconds
* Must have an ADC that was 10 bits wide for 1024 steps of resolution
* Receive and transmit information with an accuracy of 3 decimal places with the local microcontroller.

The LCD screen was driven by its own processor and must be able to communicate with the microcontrollers. Through the LCD screen the user should have the ability to read power readings from any appliance in his home as well as the readings from all of his appliances at once. The user should be able to set priority levels for each of these outlets. In addition the user should have control for turning on and off each outlet individually. If the user was trying to activate an outlet that would overload the generator then they would be given a warning. The LCD and the main control station were in constant communication.

The LCD screen would be required to do the following:

* Receive and transmit information with an accuracy of 3 decimal places with the main microcontroller.

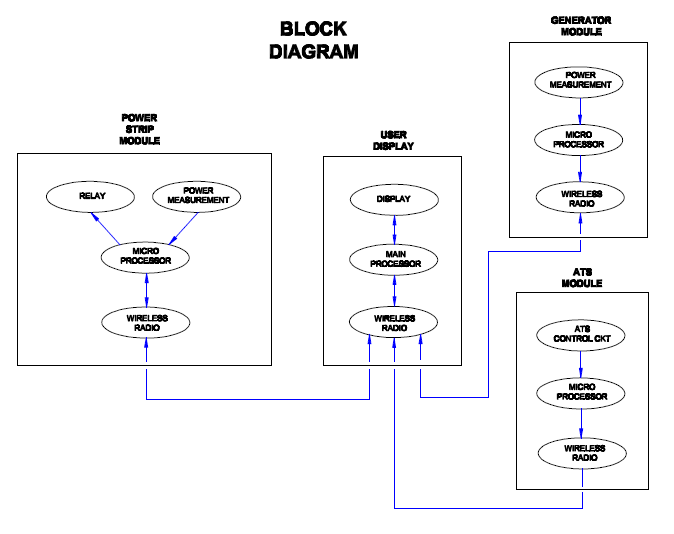
While an outlet was not being used it should enter a sleep mode to preserve the battery life of that specific unit. The microcontroller would remain in sleep mode as long as there was no signal telling it to turn on. Once the microcontroller receives the signal to turn on it would return to normal operation.

The last power metering device was the one reading the power information being delivered by the generator. This device relays the information to the main microcontroller. All of the overloading decisions of this project were dependent upon this part. This device must be able to read the power load on a generator by reading voltage and current information.

The generator power metering device was required to do the following:

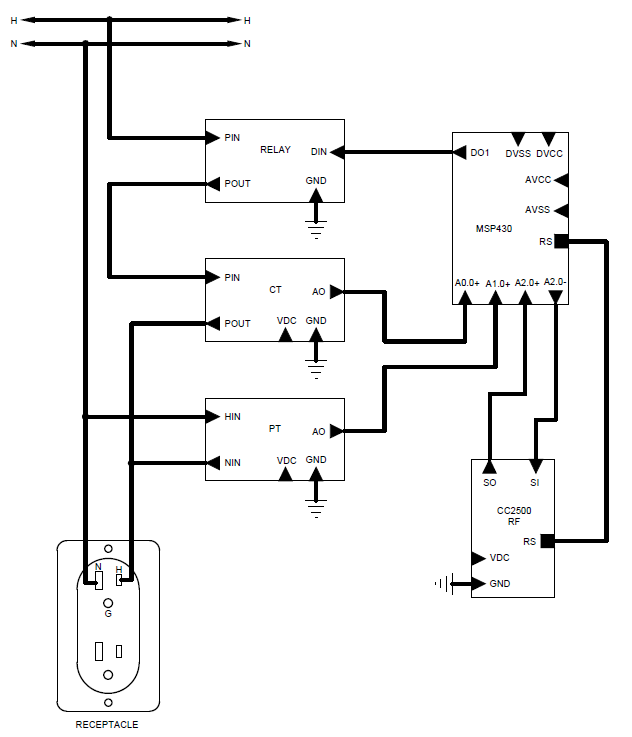
* Read power supply information from the generator twice a second
* Send the power consumption information to the main microcontroller twice a second
* Must have an accuracy of 3 decimal places
* Measure currents up to 30 amps
* Measure voltages of 240V

To begin we start off with Figure 69 shown below and this was the overall block diagram view of the entire project. As you could see the project consists of four main components including the ATS module, Power strip module, Generator module, and the user display. Each module would be briefly discussed in the proceeding paragraphs.

**Figure 69: Project Block Diagram**

**6.3 Power Strip Module**

The power strip module would contain all of the components we researched and documented in our design. The main components of the power strip module include a duplex receptacle, power relay for switching the receptacle on and off, a current transformer for measuring current, a potential transformer for measuring voltage, a MSP430 microcontroller for power calculation and last but not least a RF wireless transceiver for sending a receiving data from the main touch screen monitor. Figure 70 below was the wiring schematic of the power strip module and give a clear indication of the functionality of the circuit. Please refer back to the design section for a breakout of all the components and their functions.



**Figure 70: Power Strip Module Schematic**

**6.4 ATS Module**

The automatic transfer switch must have the capability to wirelessly transmit whether it was in normal or emergency mode back to the touch screen controller.

The transfer switch already provides a means of signaling whether it was utilizing normal or emergency power via low voltage signal so all we have to do was send that signal back to the touch screen controller. We were going to use the same TI CC2500 wireless chip that would be utilized in the power strip modules and touch screen so adding it to the network would be without much effort. The rating of the CC2550 interrupt input rating was 3.3 volts and the transfer switch’s signaling circuit outputs 12 volts so a voltage dividing circuit would need to be added to make the two compatible. In addition the CC2500 requires a DC power supply between 1.8 and 3.6 volts and we would utilize a battery to supply this voltage. Because of the ease of removing the service compartment cover of the transfer switch and access to terminals 5 and 6 we believe that making a small module with battery power would be the smallest and most economical option to us. It would be extremely easy for a homeowner to change the battery when it was time to be replaced. Below in figure 71 a simple block diagram of the automatic transfer switch module was shown.

ATS

VOLTAGE DIVIDING CIRCUIT

CC2500 WIRELESS

1.8 -3.6 DC POWER SUPPLY

QSCREEN CONTROLLER

GENERATOR POWER

UTILITY POWER

HOUSE PANEL

**Figure 71: ATS Module Block Diagram**

**6.5 Generator Module**

During times of power loss the home generator becomes the lifeline of the homes power system. In order to ensure the safeguarding of the generators operating conditions we must design a module to measure the generators instantaneous power consumption and have that information wirelessly transmitted back to the touch screen controller for program manipulations. The output from a typical home generator was 240 volts with a neutral. The ground was picked up from the houses existing grounding system. The power measurement module for the generator would be slightly different than that of the power strip modules because we have two phases to monitor instead of one. This was not a problem though as we only need one additional power measuring circuit to accommodate for the extra phase. The total power of the generator was simply the addition of the two individual phases. Except with the addition of the extra power measuring circuit the main components remain the same as the power strip modules. First the power would be measured via current and voltage sensors and then power calculations would be done using the MSP 430. This information would then be wirelessly transmitted back to the touch screen controller for further processing and display to the user. A block diagram of the generator module could be seen below in figure 72.

GENERATOR

PHASE A

PHASE B

POWER MEASURING CICUIT

POWER MEASURING CICUIT

BATTERY POWER

MSP 430

CC2500 WIRELESS

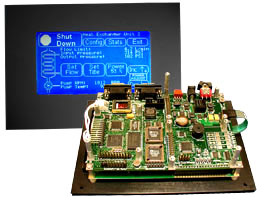
QSREEN CONTROLLER

**Figure 72: Generator Module Block Diagram**

**6.7 User interface**

After careful considerations of the needs and goals of this project we found that the QScreen Controller tm from Mosaic Industries, Inc. best suited those requirements. The QScreen Controller tm was a combination of a C-programmable single-board computer with a touch screen operated graphical user interface. This touch screen monitor was powered by the Motorola 68hc11 microcontroller, which we were very familiar with from the laboratory section of Embedded Systems class. The fact the each of the group members was familiar with the C-programming language as well as the assembly language of the Motorola 68hc11 was what lead us to selecting this product for our project.

Because this project was a proto-type of product that would be utilized in consumer household, appearance and size were of great importance and concern. We envision this touch screen controller to be mounted near the houses existing thermostat so for cosmetic purposes we want our display to be about the same size as a standard thermostat. After taking measurements of a few thermostats we found the average to be 4” tall and 6” wide. The QScreen controllers overall dimensions were 4.125” tall and 6” wide making it the ideal candidate for size requirements. A picture of the overall appearance and design of the QScreen controller could be seen below in Figure 73. As you could tell from the figure below the QScreen controller would surface mount perfect in a household wall with a cutout not much bigger then needed for a typical switch or receptacle outlet box.

****

**Figure 73: QScreen Controller**

**Permission requested from Moasic Industries**

The QScreen controller was equipped with a touchscreen operated graphical user interface on a high-contrast 128X240 pixel display with a 5X4 touchscreen overlay. The screen itself measures 4.8” diagonally and was a high contrast CCFL white-on-blue monochrome LCD. The user does have the ability to control backlight and contrast of the screen using software provided with the product. Because of the ability to navigate through different menus this size screen was more than enough to satisfy the requirements of this project.

As noted earlier the main reason for using the QScreen controller was having previous knowledge of the Motorola 68hc11 as well as the C programming language. The QScreen controller gives the programmer the option of programming in either Control C or QED-Forth. We chose to use Control C as our programming language for the programming of this project. Control C tm cross-compiler comes with the purchase of the QScreen controller was written by Fabius Software Systems and customized by Mosaic Industries to facilitate programming the QScreen controller in C. It was a full ANSI C compiler and macro pre-processor. The compiler supports floating-point math, structures and unions, and allows you to program familiar C syntax. Programming was easy done using an interactive debugger and multitasking executive. The controller was pre-loaded with hardware control routines including drawing functions for the display. After programming in ANSI C by compiling our code on our PC and downloading the code to the QScreen Controller it would then be automatically executed. The real time operating system and onboard flash memory manages all required initializations and auto-starts our application code.

Now that we have selected the touch screen monitor for the project we need to design the functionality and appear of the interface to the user. We want to avoid a busy screen and keep it simple and easy for the user to read. Multiple pages would be avoided so as not to lose the user in navigation throughout the program. A preliminary touch screen output could be seen below in figure 74. As you could see the legend indicated that a boxed area with the dotted background would be the only input area for the user. These areas include the adding a module, turning modules on and off, and also the ability to input the power rating of their home generator.

**Figure 74: LCD Graphical Interface**

ADD MODULE

MODULE #

1

2

ON/OFF

CURRENT KW

AVERAGEKW

DAILY COST

MONTHLY COST

GENERATOR KW RATING

KW

CURRENT GENERATOR CONSUMPTION

REMAINING GENERATOR POWER

KW

KW

We want to make sure the product was expandable to the user, which was why we included the “Add Module” feature. This allows the user to make the initial investment of the touch screen controller and maybe one or two power strip modules to test out and use in their house. At any time in the future they could always purchase another power strip module and connect it to their existing network. All the user need to do was plug in the power strip module wherever they would like to use it and then simply walk to the touch screen monitor and click “Add Module”. This would automatically recognize the power strip module just plugged in and would assign to it the next number in the module list. It was important to note that if the user wished to add multiple power strip modules at the same time that they would have to plug each one in separately and click the “Add Module” button before plugging in the next. The reason for this was when the user clicks “Add Module” the program was going to be looking for new devices and if more than one was available an error might occur.

The next feature the user would be able to control was turning the power strip modules on and off. This option could be used during normal power operations, but was mainly intended to be used during times of emergency generator power. During normal power operation the user should have all the modules powered on and simply turn off their appliance how they normally do. The only advantage to using the on and off feature during this time was to conserve power on appliance that draw power even when they were off such as televisions, computer, etc. This way you could shut the power off to them via the touch screen without having to unplug each power cord from the wall. The main purpose of the on and off feature was used when the system was in emergency mode. As soon as normal power was lost and the transfer switch switches over to emergency mode the QScreen controller would automatically shut off all power strip modules and wait for the user to decide what modules they want on. By looking at the “average power consumption” column the user could identify normal instantaneous power draw from individual appliances measured in KW. The QScreen knows the KW rating of the generator since the user initially input its value. As the user turns on individual power strip modules the program would calculate the remaining available generator power and report it back to the screen. This way the user would be able realize when they were approaching the limitations of the generator. Safety features would be added to insure the user does not accidentally turn on an appliance that would overload the generator causing failure.

The user would also have the ability to enter the KW rating of their home generator. This was important for the “emergency mode” operation of project. The software would monitor the usage of power from the generator as well as all the power strip modules and would compare to make sure the power consumption of the all the power strip modules together does not exceed the rating of the generator. If this were to happen the generator would fail and the project would have not satisfied its goal.

**Section 7: Conclusion**

The engineering industry had been built around the needs of the consumer, be it those of individuals, companies, or military corporations. Necessity was the mother of invention (perhaps in our field more than any other), and everything that goes into our jobs and products (what we do for the products we sell, how we do it, and what features go into it) was almost always derived from the end-user’s needs and means.

Through the last decade, our country had seen enough financial hardship and stress to suffice for a lifetime, and as a result, more and more people were concerned over every penny that comes and goes through their households. More recently, this had encouraged the practice of trimming the use and overuse of household utilities to save money on everything from gas to water and electricity. While their efforts were certainly applauded, most homeowners were not aware of the numerous appliances that they take for granted which may drain a large portion of their monthly energy budgets. Turning off the A/C in the summer or the heat in the winter was one option that many have embraced, but having the ability to monitor all of the smaller appliances in the home may also help lead to more energy-aware consumers. Thus, one of the goals of our project was to aid the homeowner in his goal of lowering his monthly electric bill by showing the power usage as well as the costs of running appliances.

Another facet of our project for a home energy monitoring system was an automatic transfer switch (ATS) from normal (electric company) to utility (generator) power, a feature which had been neglected by many of the competing models. Not only was this interesting on its own, but was an extremely helpful tool to those affected by hurricanes and other strong thunderstorms each year in Central Florida. The ATS in our design would ensure that not a minute would go by where those caught in storms would not have access to the many things they may need, including: access to the media (television, radio, internet, etc) with potentially life-saving information about the surrounding weather; the ability to charge wireless electronic devices (cell/smart phones, tablets) with satellite access; a running refrigerator to keep food fresh while the power lines were down; and a hot water heater to take showers, clean clothes, and wash dishes. In addition, our design does not allow for an overload of the generator, helping to save the user the cost of replacing it.

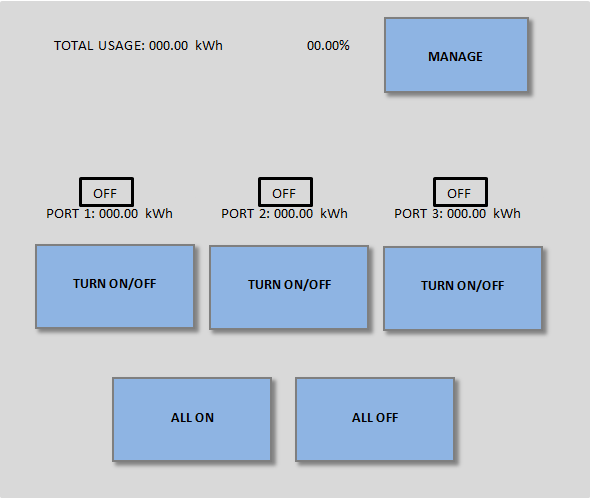
Technology to conserve energy was certainly not new, but it was at an early stage in its development; it had a long way to go, and many innovations along the way. Nor was the concept to monitor utility usage new or novel. In fact, most of the large companies that design and manufacture microprocessors have had whole series of products specifically targeted toward monitoring and measuring utility usage, and peripherals associated with it. However, our project combines these ideas, along with an automatic transfer switch, into a home energy system that had the potential to stand out among and even rise above its competitors.

**Section 8: Product Manual**

The user manual for our product would be as simple as possible, and ideally should only consist of two or three different commands. The process for running and using our system was detailed below and was in two parts: the first part was during normal power operations, and the second part was during a power failure.

**Normal operating conditions (main power)**

1. To begin, wake the system up by touching anywhere on the screen, or by pressing any of the buttons on the side.
2. You would be greeted with a home screen similar to the one shown below in Figure 75. Each port shown displays power currently consumed as well as its current state (on/off).



**Figure 75: LCD home screen**

1. To switch the state of an outlet/utility (i.e. to turn on outlet on that was off, or to turn an outlet off that was on), simply press the button beneath that port’s name, labeled “TURN ON/OFF”.
2. To turn all ports on, press the button “ALL ON”, located near the bottom of your screen.
3. Conversely, to turn all ports off, press the button “ALL OFF”, also located near the bottom of your screen.
4. At the top of the screen would be a running total of KWh for the month, as well as a percentage, which was the percentage of your monthly goal you have used thus far. By clicking the button labeled “MANAGE” beneath this total, a separate screen would appear which would allow you to change this goal.
5. On the new screen, the user would see their current monthly goal for power consumption. By pressing the button next to this screen labeled “EDIT”, the user could input any goal he/she wishes, and the program would automatically update with this information.
6. Next, the user could select “EXIT” on this screen, and then “MAIN SCREEN” on the next screen, and would be returned to the first screen.

**Utility (generator) operating conditions:**

1. During utility operating conditions, the user would be greeted with the same screen as before (see Figure 400). Each port shown displays power currently consumed as well as its current state (on/off). Each port would also display a percentage number which corresponds to the percentage of power that unit consumes with respect to the amount of power available from the generator.
2. To switch the state of an outlet/utility (i.e. to turn on outlet on that was off, or to turn an outlet off that was on), simply press the button beneath that port’s name, labeled “TURN ON/OFF”.
3. To turn all ports on, press the button “ALL ON”, located near the bottom of your screen. However, if doing so would overload the generator, the screen will read “WARNING: GENERATOR CAPACITY REACHED”, and would not change the state of any of the connected ports.
4. Conversely, to turn all ports off, press the button “ALL OFF”, also located near the bottom of your screen.
5. When normal power was restored, the main home screen would return to how it looked in the first part of the manual, and the user would no longer be limited to using a certain number of ports.