Smart Home Energy Monitoring System

Alejandro Dirksen, Zaida Gonzalez, Wayne Rodenburg, Marisa Vega

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Our projects objective is to monitor and report back to a user the amount of power being used by items in their home in both on and off states, and to inform them about the amount of parasitic power being used by most items in their home. Our group felt this was an ideal choice of senior design projects as it had both electrical and programming components and dealt with the hot topic of saving money and reducing ones carbon footprint by using less energy.**

***Index Terms* — Bluetooth, Application Software, Embedded Software, Low-Power Electronics, Power Generation Economics, Power Measurement.**

I. Introduction

We chose to do our project after we formed our group and we found out that we had two EE and two CpE majors. We know that energy consumption is a problem in today's modern world, and that a fair bit of the energy consumed is wasted through standby power consumption. Standby power is power used by electronics when they are turned off or in standby mode. Since those electronics don't actually turn off, but instead they wait in standby mode for a user input and while they are waiting for an input, they are constantly drawing power while not even being used by the consumer. Prime examples of this are televisions, Blu-ray players, computers and any other electrical device in your home that is plugged into an outlet.

Our project looks to monitor, record, and report back to the user about their power usage on devices that are plugged into the SHEMS units. Device to device communication will be done via Bluetooth and that information will be stored on the central hub and then sent to an Android application on a phone or tablet. From there the user can decide what they want to see as well as turn the devices on or off remotely.

II. Project Goals and Objectives

Our projects main goal was to reduce the standby power used and in turn reduce the consumers total power bill. We will do this by informing the consumer about standby powers existence and how to combat it. This will be done by letting the consumer see how much power a device they have plugged into the SHEMS is using at any given time, when they are using it as well as when they are not using it. By doing this we feel we will be able to let the consumer see just how much power is being wasted, and in turn, how much money they are just throwing away by leaving these electronics plugged in and drawing power while not in use.

For our project we chose objectives that we felt would be useful and achievable. We first wanted to be able to display to the user in an easy and convenient way how much power they were using and how much that power was costing them in dollars and cents. We want to make the device easy to use for the consumer, requiring no technical expertise. We also wanted to make sure it was small enough to not get in the way. We also wanted to allow the consumer to be able to turn the device on and off remotely so as to disconnect it from the wall outlets power supply without having to actually unplug it, but instead by telling the central hub or Android application through the touch screen to turn that specific device on or off. This can be done either by using the devices number or by using a name that the consumer has chosen to represent that device as so that they don't have to remember what goes where, but instead can turn off the television or turn off device one.

For our projects we wanted to make sure that our project was able to accurately measure and record the current and voltage at a given time as well as transmit that data via a Bluetooth device to our central hub and from there to our Android application. We also wanted to make sure that our project actually saved the consumer money, so we had to make sure we used low power parts to keep our devices power draw to a minimum. Part of our wish list was that we wanted to ensure that we didn't take up to much space on the embedded microcontrollers limited storage space by saving all of this information, so we decided to refactor the data we had stored over time to only keep the narrow second to second focus for a short while, and ultimately only keep the daily minimum, maximum, and average power usage and at what times those occurred. The refactoring information can be found in list below.

- After a day we plan to take the average, minimum and maximum of the data for each minute and then delete the data for each second of that minute.
- 60 samples per minute combine to a minimum, maximum and average for that minute
- After a week we will go back and take the average, minimum and maximum of the data for each hour and again delete the data for each minute of that hour.
- 60 samples per hour combine to a minimum, maximum and average for that hour
- After one month we will go back and take the average, minimum and maximum of that data day and again delete the data for each hour of that day.
- 24 samples per day combine to a minimum, maximum and average for that day
- Ultimately we will only keep our daily data as an average, minimum and maximum for each day and delete all of the other data points for that day. At the end of each month we will find the average, minimum and maximum for that month into our database of data. This will allow the user to view their past months average, minimum and maximum as well as their daily average, minimum and maximum for each day as well as view their monthly and daily averages for the past years in which data has been collected.
- 28 – 31 samples per month combine to a minimum, maximum and average for that month

III. PARTS CHOSEN & SPECIFICATIONS

A. Microcontroller: MSP430G2553IN20

Originally we wanted to use the System on Chip cc2540 which included BLE and MCU application; however the chip libraries were very difficult to use. We decided to utilize one of MSP430 family of MCU because we were familiar with its libraries and coding. Among the MSP430 we chose the MSP430G2553IN20 whose typical applications includes encoding of low cost sensor, ADC that process the data ready for display or for transmission to a host system.

The following TABLE I display some of the important features that help in our decision for the MSP430G2553IN20.

TABLE I
SUMMARY OF ABSOLUTE MAXIMUM RATINGS FOR MSP430G2553IN20

Supply Voltage	1.8V TO 3.6V	Temperature cond.	-40 to 85˚C
Voltage in any pin	-0.3 <=3.9V	ESD	2kV
Internal Frequencies max	16 MHz	Architecture	16 bit RISC
Ultra low power consumption	Active mode: 230uA at 1 MHz Standby Mode: 0.5uA Off Mode: 0.1uA	Instruction Cycle Time	62.5 ns

B. Relay SSR 120VAC 8A Triac (Sharp Microelectronics)

The integration of an infrared emitting diode, a phototriac detector and a main output triac makes this device suitable for high voltage AC loads. It provides 3.0kV isolation (Vrms) from input to output highly recommended when handling high voltage. The output current It (rms) is less than 8A and it features a non zero crossing functionality. The triggering current is 0.1mA, therefore when the current falls below this value the relay will turn off, the output Triac will be in an open circuit mode. If the voltage across the Triac runs faster than the rated dV/dt the relay might not turn on. This could be an unlikely situation, in order to avoid it a snubber circuit is connected to the relay. We used the manufacturer recommended values for Cs=0.022µF and R=47Ω.

C. Fully Intergraded Hall Effect Current Sensor ACS 712

After much research and looking at a number of different ways to sensing current, including a shunt monitor, the most simple, direct and efficient way to sense the voltage, without having much error, overheating and circuit alterations is by using the Hall Effect principle. There are different types of parts in the market that uses this principle with many packaging styles including many bulky and inconvenient. Allegro Microsystems LLC is a leader in high performance semiconductors that offer a complete line of this type of sensors conveniently package in practical IC's and able to sense from low amperes to 200A. Unlike many other popular companies that have different alternative for their sensing devices and with extra features, we were mainly looking for low current values. For our project it is necessary to read from small current values .2592A (video game console) to an average value of 5A (microwave or small appliance). Also, the maximum possible current available in a circuit breaker is 15A at the time and most 15 NEMA outlets in a home are connected to this sized breaker. In addition for the purpose of this project we are focusing in everyday life current loads around a household, and most of these loads are between 1/2 amp to 5A. This information led us to use the ACS712 sensor, a current hall with a maximum load of 8A. The mechanism that this IC uses is a precise offset linear current circuit with copper conduction path where the current flows. When the current flows through this path it generates a magnetic field that the ACS712 converts into a proportional output voltage, also the thick copper conduction path allows this device to survive up to 5 times over current conditions. The terminal pins for the conductive path are electrically isolated from the signal leads (5-8) eliminating the need for opt isolation.

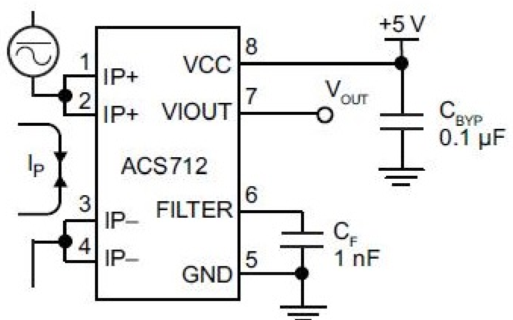

Figure 1. Characteristic performance of ACS712, output voltage vs current sensed.

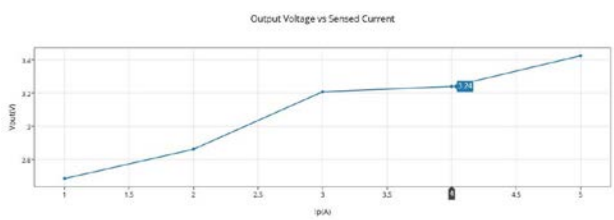

Figure 2. ACS712 Vout is an analog signal that varies linearly with a bidirectional input Ip

Important formula when using the ACS712:
Smallest current the sensor is able to resolve with noise: In
In=noise (mV)/sensitivity (mV/A)

According to the datasheet performance characteristic,
Vnoise pp=21mV Sensitivity=185mV/A
Therefore the smallest current in=113mA

Also the sensitivity value is applied when performing the current conversion from the voltage sense by ACS712. For example, a voltage is sense at 3.5V, then 2.5V must be subtracted because that is it's value at 0A, giving 1.0V. This value is then divided by the sensitivity 185mV giving us 5.40 A.

D. Circuitry for Sensing and Turning Off and On the Load

Circuitry for Sensing and turning off and on the load

The ACS sensor and the relay must be connected in series with the hot and neutral from the outlet like is shown in figure 3.

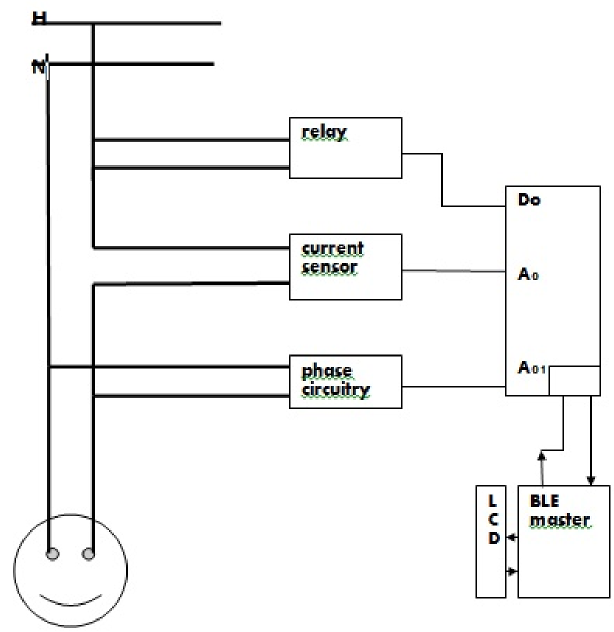

Figure 3. Schematics of the hardware of the project

The sensor provides information about the current of the corresponding load. The way it accomplishes this is by providing an analog voltage signal to the microcontroller where by using the sensitivity values (ACS712 datasheet) the current is calculated. The relay is connected in series with the sensor to be able to turn on and off the device connected to the monitoring node. The voltage circuitry shown in Figure 4 is what we use to sense the voltage use by the load, the output of the circuitry is proportional to the input, and this proportionality is used by the microcontroller to display the voltage value. We tried to use the full wave but the MSP430 voltage range in any pin is between -0.3V to 4.4V, but introducing a diode we rectified the Vout to a half wave adjusting the calculations in the code to get the final voltage value used. Phase compensation is used to calculate power. The frequency in mains varies but slightly we assume 59.9 Hz for our calculations then the voltage samples are shifted accordingly. The phase shift is done by providing N samples and the delay in this information will be used to calculate the phase angle and consequently the real and reactive power. The formulas used are P=VIcos (α-β), for real power and Q=VIx=VISin (α-β) for reactive Power.

E. PCB Board

After deciding on the general layout of the design, a schematic was drawn using Eagle cad as our graphical layout editor and later the board and Gerber files were generated. Figure 4 is the resulting final eagle board before Gerber files generation.

This work was later submitted to Advanced Circuit which provides a FreeDFM, a graphical PCB board file check that rapidly provides a report of potential errors in the design. Also, this company provides a fast shipping compare with other PCB manufacturers.

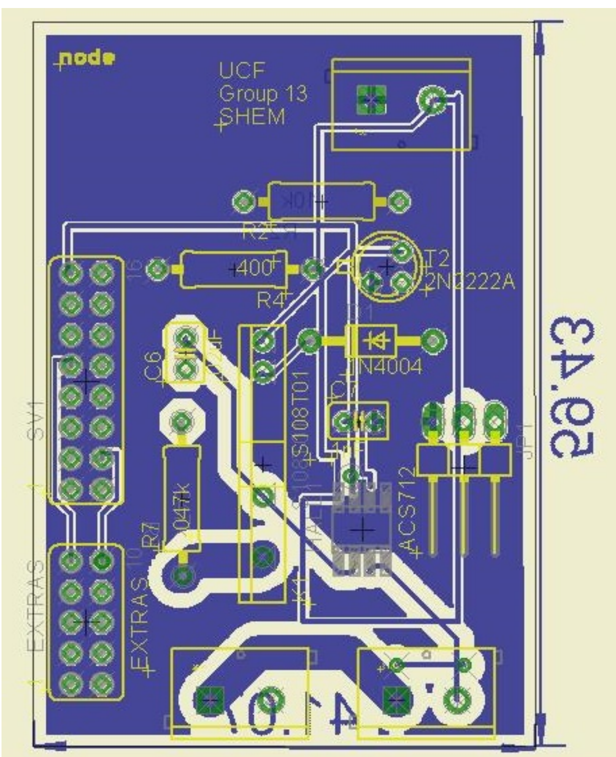

Figure 4. Eaglecad board Design for the monitoring node

IV. Software

A. Application Platform Chosen

We chose to go with an Android application over and Apple application for several reasons. One of the most important ones being that to publish and Apple application to iTunes you have to pay $99 a year for that license, where are to publish and Android application to the Google Play store it is free. Another factor that played into this decision was that Android applications can be done using a Java codebase which can be coded on any of our computers and that we have had some experience with coding in java from past classes. Additionally Android is widely used worldwide unlike Apple which has its primary customer base in the US.

B. Android Application Design

We are using the build in Bluetooth libraries, time and date, SQLite database, tabs and touch interface to build much of the app. We want to allow the consumer to easily change between views to turn devices on or off, rename devices, monitor devices and chose the time scale they want to see the information in. We want to make the interface for all of this very simple and seamless for the consumer so they will not have to learn anything new. They can simply pick the tab they want to go to and then do what they want to within that tab. They can select the device tab and from there rename, turn on or turn off any devices they desire. Then they can select the display tab and chose to see the data for the past year, past month, past week, day or hour. Then they can return to the main tab and see the data for the given time interval they have selected.

C. Satellite Node Microcontroller Code Design

We have the satellite nodes set up to sample the current and voltage and then record that every second as a minimum, maximum and average along with the time. This will then be stored on the satellite nodes microcontroller till it has saved up a minute of data. It will then contact and transmit the data for that minute to the central hub for processing.

D. Central Hub Microcontroller Code Design

We have the central hub set up to remain in a low power listening state most of the time. When it gets a contact from one of the satellite nodes it will respond and receive the data they are sending. It will then store this information into that devices text file. The user will be able to see the data there on the central hub's LCD, turn devices on and off and rename them much like they can with an Android device.

V BLUETOOTH

Because SHEMS is intend to be an ultralow power consumption device (the main purpose is for users to know how much power they are consuming but also to save power with this device), we needed the Bluetooth to be low in power consumption as well. A lot of Bluetooth devices were considered for this project, as for example the Mate Silver -RN-42 ($39.95), the TI CC2540F256 ($100 for the kit and/or free samples), and the TI CC2564 (free samples).

We decided to go with the Bluetooth Mate Silver because it is very easy to use, cost effective, and compatible with the microcontroller the project is using, which is the MSP430. In addition, this Bluetooth offers a lot of other benefits that are suitable for the project. These include:

- Low Power Consumption
- Low Voltage Operation
- The range is ideal for the project, because it will cover an average house measurements
- Bluetooth is separate from the microcontroller - there is no need to make any code in order to have the Bluetooth working
- All processes are encapsulated inside the Bluetooth, no further code needs to be done
- It is easy to connect with the circuit - RX/TX pins

TABLE II
SUMMARY OF IMPORTANT FEATURES FOR THE BLUETOOTH MATE SILVER

Specification	Value
Receiver/Transceiver	Yes
Low Energy	Yes
Built in antenna	Yes
Low Power	25 mA
Operating Voltage	3.3 V

This particular device doesn't require a program or code to make it work, like with others Bluetooth devices we researched. This Bluetooth is very simple and keep all the necessary code and processes encapsulated, so we don't need to do anything with it. The only thing that we needed to do is to connect the Receiver and Transmitter pins directly to the circuit (central hum or satellite stations) and it will automatically send and receive the data that we ordered it to send when we program the microcontroller. Of course the microcontroller needs to send and receive information through the whole circuit, and that is where we sent the information necessary for the Bluetooth. This is the easiest way to make a connection thorough the Bluetooth and the circuit.

The main idea for the interface of the Bluetooth with the microcontroller and the device itself is to transmit the data through the digital I/O pins of the microcontroller from the different satellites nodes to the central hub and from the central hub to an Android device via Bluetooth. This is a very clean and simple procedure that will not cause to much extra work, but will provide a great deal of extra interface options and data to the consumer.

VI LCD

The LCD will only be used in the central hub station. In the central hub, the information about the different satellites nodes, (which are the smaller devices whose power consumption is being measured), is received via Bluetooth and then processed. The user will be able to see in this LCD all of the necessary and important information about the devices that are plugged to the satellite nodes. This information include: device ID, how much power is consuming in terms of either watts or dollars and cents (because it is easier to understand), and how much they are saving. It would be ideal to have data from the device and store it in the memory. In this way the user will be able to look at previous power consumptions values and make their own conclusion on how efficient or inefficient the devices they are monitoring are and when or if they should use the SHEMS to turn those devices completely off by disconnecting them from the power grid though the use of a build in relay to do just this. But, this is a wish list that we can incorporate in the future.

The LCD chose is the Serial TFT 3.2", which is a resistive LCD touch screen and easier for consumers to use. Nowadays, almost everything is touch screen, so users won't have any major problems dealing with it as it is a technology they are already familiar with. As show in table III below, the features offered by this LCD are convenient for the project because we didn't want a complex, fancy, hardly to understand LCD. And also, it is low in power consumption and this was basically the breaking point between this LCD and many other considered.

TABLE III
SUMMARY OF IMPORTANT FEATURES FOR THE LCD

Specification	Value
I/O Pins	30 with expansions
Colors	65K true colors
Programmable Flash Memory	14 kb
SRAM	14 kb
Resolution	320 x 240

Operating Voltage	4.4 – 5.5 V

The LCD will be interfacing with the Central Hub by a programming code that specifies what the users will see on the screen. Through the different I/O pins from the LCD, the data will be transmitted back and forth through the microcontroller to the different satellites. Internally, the LCD is connected to batteries that will supply the necessary power for it to work effectively in a low power consumption mode

The programming of the LCD is something that goes hand to hand with the Android application because basically everything that appears in the Central Hub will appear in the Android application. And, of course, from the Central Hub or Android app, the users will be able to turn on and off their devices, label them, and see their consumption along with the history of the device, so this is something that is done in a very user friendly way.

VII Hardware Design

A. Central Hub Design

The Central Hub design is one of two hardware designs in this project, the other being the satellite nodes. The Central Hub will contain the LCD touch screen, the CC2540 microcontroller with the Bluetooth in it, and a battery for power. The first problem we faced was powering the LCD screen and the microcontroller with the battery as the battery provides 5 volts while the LCD uses 5 volts and the Microcontroller uses 3.3 volts. To accomplish the 3.3 volts from the 5 volts battery we used a 5 volts to 3.3 volts voltage regulator which is the LM3940 from Texas Instrument. This is connected to the 5 volts battery with a 0.47 μF capacitor at the input pin, the ground pin is connected to the ground and the output pin is connected to the microcontroller and a 33 μF capacitor in parallel. This way we can power up the LCD screen and the Microcontroller from the same power supply. Interfacing the LCD is as simple as connecting the RX and TX pins from the Microcontroller to the RX and TX pins on the LCD screen. To avoid confusion with the PCB manufacturing process and the possibility of getting problems with transmitting the wireless signal via the antenna, we opted to use the antenna circuit that was built onto the board already and simply use the RX and the TX pins to interface with the LCD. Since the LCD has its own Micro SD storage capabilities as well as its own Microcontroller, which is the Picaso processor, the CC2540 will be used for its Bluetooth capabilities and as possible data conversion before sending the information to the Picaso processor. The Picaso processor will display the information on the LCD screen and allow the user to interact with the software. User inputs will be sent back to the Picaso processor and then sent to the CC2540 Microcontroller and this will transmit the data to the satellite nodes' Microcontroller. The housing for all of these components is a white plastic cover that is small enough for portability and large enough to fit everything inside. The batteries will also be placed inside the plastic box and will connect to the PCB through a short wire.

B. Satellite Node Design.

The focus of our design was on this satellite monitoring station as it will be the heart of our measurements and control of the appliances and will monitor how much power is being consumed. We have looked over each individual component being used and now we will look at how we designed this satellite node. The primary part of the satellite node is the current sensor and this sensor will be connected in series with the load and will get a 5 volt supply and a ground and will output a voltage to the microcontroller that will be used to calculate the current. The next crucial component in the satellite node is the relay. The relay will need to switch on or switch off the device connected to the wall by receiving a signal from the microcontroller. In order to accomplish this, we used the Sharp S108T01. This will get connected to the load and the wall voltage source. To properly connect it the first pin of the relay will be connected to the hot line from the wall and we will add a snubber circuit as well. From the relay we will connect a wire to the ACS712 that will measure the current, and between the negative pins of the ACS712 and the neutral from the wall, we will install an outlet so that any appliance can be connected as a load. At the other end of the relay, we will connect the 5 volt source with a 400 ohm resistor in series and at the negative pin of the relay we will use a small signal switching transistor 2N2222A that connects at the collector. The ground will be connected at the emitter of the 2N2222A and the base will be connected to the digital out of the microcontroller. This ensures that the microcontroller can control when the 5 volts will flow through the relay and when it shouldn't. There will be a 1N4004 diode connected with the cathode at the 5 volts and the anode at the collector as well. This design will ensure that the microcontroller can control the relay to turn the load on or off while at the same time the current sensor can sense how much current the load is taking. To be able to calculate power, we need to be able to measure the voltage from the wall. In order to decrease the voltage so that it is safe for the microcontroller, we made use of a voltage divider circuit. This voltage divider will output its divided voltage to the microcontroller and this value will be used by the microcontroller to calculate the voltage from the wall. The final step is to power up the microcontroller and since the power supply is a 5 volts voltage source, we will use the same voltage regulator circuit that was used in the Central Hub, which is the Texas Instruments' LM3940, to convert the 5 volts to a 3.3 volt to power up the microcontroller. The casing of the Satellite node also contains the 5 volts power supply that converts the 120 VAC into the 5 VDC and it will hold our PCB with our design in it. The back end of the outside part of the casing will have the plug coming out that can connect to any wall outlet. The front end of the casing will have an outlet so that any appliance that the user wants to monitor and control its power consumption can be connected to our device.

VIII HARDWARE TESTING

A. LM3940 Voltage Regulator Testing

To make sure all of the devices are working as intended we had to test them out. We started off with the LM3940 voltage regulator. To test this part we connected it with a voltage source that allows for up to 6 volts DC output that will be used as the input. This input will connect to the input pin of the voltage regulator with a capacitor of 0.44μF. The second pin which is the ground pin was connected to the ground and the output pin will connect to the oscilloscope with a capacitor of 33μF in parallel. We set the input voltage to 5 volts and measure the value of the output, which came out to be 3.334 volts. Since the microcontroller can work between 2volts and 3.6 volts, this value is fine to run the microcontroller. To test for inaccurate output voltages, we varied the input voltage up to 6 volts and the output voltage was still inside the range needed to power the microcontroller. Even lowering the input voltage down to 4 volts we will still be in the operating voltage for the microcontroller.

B. S108T01 Relay Testing

The relay took time to figure out how to connect the pins. During the testing with the relay, we connected a modified plug into the wall outlet and connected it to the circuit. The hot was connected to the pin 1 of the relay, the pin 2 of the relay was connected to a load, which was initially a 1k ohm resistor, and the neutral was connected to the load. At the pin 3 of the relay, we connected the positive side of the 5 volts supply with a 400 ohm resistor in series and at the pin 4 of the relay we connected the ground. The initial test resulted in multiple failures as some wires were not connected properly and the 1k ohm resistor was also not capable of handling the vast amount of current through it, which was almost immediately burned. Upon rigorously investigating and replacing the relay, which was burned in the process, as well as the load by using a laptop charger, we got the relay to work as intended. When a 5 volt is applied, the relay will allow current to flow through and when this is turned off, the relay stops the flow of current through the load.

C. ACS712 Current Sensor Testing

In order to properly test the current sensor we need to be able to measure the Amps that the load is consuming. This was done with a product that measures the amount of Amps a device draws from the wall outlet. With this device connected we can compare the sensed current with the actual current being measured. The two positive pins were connected to the hot and the two negative pins were connected to one end of the load while the other end of the load was connected to the neutral. The loads used were a laptop charger, a computer, and a computer flat screen display. Using the oscilloscope we measured the voltage coming out of the acs712 and calculated the Amps. The results were very accurate and we were pleased with the consistency of the measurements.

D. Bluetooth Testing

The Bluetooth is inside the microcontroller which was located on a Key fob and also on a USB dongle. A battery powers the Key fob device and the USB gets powered by the desktop computer. The first step is to install the BTool software from Texas Instrument. Once this was installed we will connect the USB dongle to the computer and use the device manager to manually install the driver for the USB dongle that comes with the BTool software. When the installation is complete we will look in device manager to locate the port in which the USB dongle is located as we need to specify this when using BTool. Once this is set up in BTool, press the button on the Key fob and at the same time we use BTool software to scan for devices. Once found we will choose to connect to the Key fob. In order to test if the Key fob connected we write a hex value to notify the software that a button was pressed. Once this value was written we can press the button on the Key fob and as expected, the software displays this on the computer. The Key fob circuit also has an accelerometer on it which can also be turned on to further make sure that the Bluetooth connection is working properly as well as a buzzer and a proximity alert. All of these were tested and all worked seamlessly. The proximity sensor will cause the Key fob to start beeping when the Key fob loses signal with the USB dongle. This feature can be incorporated into the software to notify the user that signal is being lost.

E. Microcontroller Testing

To test the Microcontroller, we used the CC debugger. The CC debugger connects to the debug terminal of the key fob or the dongle and then connects to the computer through the USB port. To program the microcontroller we used the IAR Embedded Workbench, which is available online. To test this we tried out a code that would make the buzzer beep. After successfully testing the microcontroller and becoming familiar with the workbench we started implementing the code.

ACKNOWLEDGEMENT

We wish to acknowledge and thank our faculty advisor Dr. Samuel Richie for his help and guidance with our project. We would like to thank Duke Energy and Texas Instruments for their funding of our project. We would like to thank Mr. David Douglas for helping us procure some of the testing equipment we needed. We would also like to thank Matt our lab TA for his help with some of the problems we ran into with the different programs we had to learn to use during the course of this project.

BIOGRAPHIES

Alejandro Dirksen is an Electrical Engineering major senior who is graduating with his Bachelor's of Science in Electrical Engineering in May of 2014. Alejandro is hoping to pursue a career in research and design specializing in microwave engineering, VHDL, and Microelectromechanical systems.

Zaida Gonzalez is a computer engineering major senior who start at UCF back in 2012 after a transfer from the University of Puerto Rico in Mayaguez. After graduating in 2014, Zaida is interested to work in a local computer field company, being microcontrollers and Bluetooth what she likes the most. After the great experience of the Senior Design project, she is hopeful on finding a prestigious company in the Orlando area, which will provide with the knowledge and experience she desires.

Marisa R. Vega is a senior electrical engineering student at the University of Central Florida and will be graduating on spring 2014. After graduation, Marisa hopes to pursue her career in the Design of power distribution or power utilities.

Wayne Rodenburg is a computer engineering major senior who plan to start his career in the computer engineering field after graduation in May 2014. He has many interests in the field of computer engineering and is still unsure of which of those to pursue as a long term career.