S.H.A.P.E.R. Smart Home Automated Power Expense Regulator

Juan A. Aleman, Maria D. Alfonso, Gregory Pierre, Francine Vassell

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

***Abstract* — The objective of this project is to create an efficient system that implements the "smart" appliance concept while also raises awareness about energy consumption. The circuit created for this project is able to control the lighting system and thermostat relying on the input obtained from motion and ambient light sensors. Moreover, the power consumed by each appliance is calculated and displayed for the user knowledge. Finally, all of the features mentioned above are accessible through a mobile application. Each module will communicate via I2C.**

***Index Terms* — Bluetooth, Home Automation, Lighting Control, Motion Detection, Power Measurement, Smart Homes.**

I. Introduction

Home automation systems have become very popular over the past several years. People have accepted and embraced this new technology-driven era where "smart" phones, cars, TVs, and even washing machines are built every day. Moreover, the energy consumption is also an issue that should concern every person. This project is aimed to create a conscious mind in the consumers when it comes to energy consumption, doing so through the use of smart applications and technology. The main focus of this system is to use automation to control lighting facilities, thermostat functions and high energy use appliances. This system called S.H.A.P.E.R. will give the home owner the ability to control their facilities via Bluetooth through an application on their mobile device and the main hub of the system using the Arduino Uno will automate these functions to save power consumption. The main hub's major functions will be to detect motion, allow for dimming capabilities, detect external lighting and calculate power used by an appliance. The hub's interface will utilize functions to prioritize the order of options when turned on and the user may edit these order where necessary.

II. System Concept

A graphical representation of the overall S.H.A.P.E.R. system is shown below

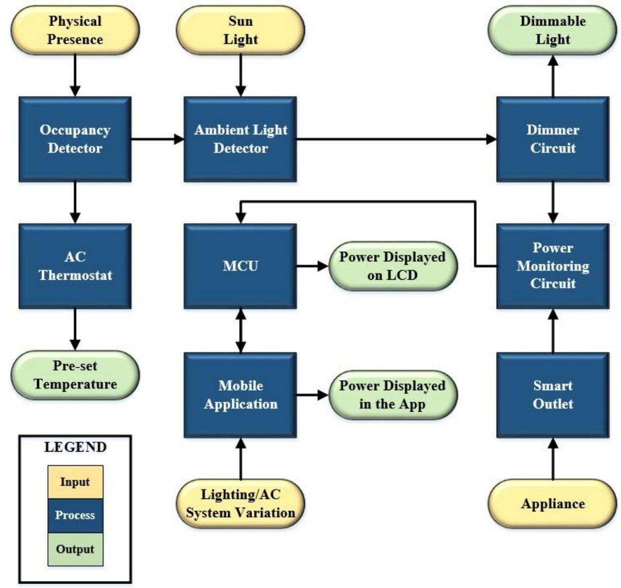


Fig. 1. Flowchart of the Overall S.H.A.P.E.R. System

The project prototype is mainly divided into three modules: Sensors Station, Power Monitoring Station, and Main Hub. The communication among these three modules is achieved via I2C serial computer bus.

The *Sensors Station* is designed to obtain and process the information received from the sensors. It includes the motion and ambient light sensors as well as the dimmable lighting circuit. Once physical presence is detected, the information is sent to the Main Hub in order to adjust the thermostat to the desired settings. The information also triggers the ambient light sensor, which compiles the desired luminance data needed to adjust dimmable lighting system. The light bulbs are dimmed according to the preset preferences.

The *Power Monitoring Station* is designed to measure the current flow from each appliances and utilize it to calculate the desired power values. It includes a current sensor that is in charge of estimating the current flowing through the desired appliance. Such value is processed in the microcontroller to obtain the power being consumed per appliance. This information is later sent to the Main Hub in order to be displayed and viewed by the user. Moreover, a smart outlet is included in this module, which allows the user to cut off the current flow to a desired appliance.

The *Main Hub* is designed to combine the information obtained from the previous mentioned stations and make the system work as a whole. It obtains the information from the motion sensor and sets the thermostat temperature accordingly. Furthermore, it gets the information from the Power Monitoring Station and displays it in an LCD.

In addition to these three modules, a mobile application was developed to perform the same functions. This app will control the lighting system, thermostat and smart outlet while also making the user aware of their power consumption habits.

III. System Components

This section provides an overall view of each individual component. Since each module has its own peripherals, a better presentation of the entire circuit can be achieved by briefly describing each part, which are later interfaced to create the final product.

A. Microcontroller

The heart and brain of our system is an Atmel AVR microcontroller, model ATmega328P. This particular family was chosen for its high performance and power saving modes which are compliant with the objectives of our project. The CPU runs at 20 MHz, which is perfect for amount of computations needed by the system. In addition, the development board featuring this microcontroller is affordable and easy to use, and the Arduino IDE makes the programming and debugging of the module simple.

B. Bluetooth

The communication of the Main Hub Station with mobile application is performed by the BlueSMiRF Bluetooth Modem designed by Sparkfun. It features an RN-42 module with a range of about 50 to 60 feet, which suits the desired range among the circuit modules. It operates in harsh RF environments like WiFi and ZigBee, which is very useful when it comes to a home environment. The module features a built-in antenna that makes it ready to use in applications. The module uses UART to communicate with the microcontroller, which makes the communication easier to implement with the ATmega328P. It is easy to program through command syntax.

C. PIR Motion Sensor

The motion sensor is responsible for letting the system know if a person enters or leaves a room in order to activate or deactivate the lighting system and thermostat. The module chosen features a probe LHI778 which has high sensitivity needed in the system to accurate detect the presence of a person. It also contains two potentiometers; one is used to adjust the distance range and the other one is to adjust the time delay. These features are very helpful when alterations are needed to obtain better precision.

D. Ambient Light Sensor

The ambient light sensor is responsible for obtaining the amount of light being perceived in a room in order to dim the light bulbs to a certain percentage. The chosen module is the Phantom YoYo Light sensor which features a photo resistor that makes it adopt a photoconductive resistance giving it the ability to detect a wide range of luminance or light. The interface to the microcontroller is done via analog pins which is compatible with the ATmega328P, and it outputs a range of values according to the intensity of the ambient light.

E. Dimmer

In order to create a dimmable lighting system, the use of a dimmer circuit is needed. The chosen module is the Krida AC Phase Control circuit that specifically works with LED dimmable light bulbs. The input range is from 110 VAC to 240 VAC with an auto detect frequency of 50Hz to 60 Hz. This is perfect for our circuit since most of the power outlets in U.S.A. work with 120VAC and 60 Hz. The module also features a zero-cross detection that makes it easier to program the dimmable stages. Since LED light bulbs consume significantly less power than fluorescent light bulbs and this module is designed to work with LED light bulbs, creating a lighting system that meets the project requirements is achievable by using the featured dimmer.

F. Current Sensor

Measuring the power consumption of each appliance can be achieved by obtaining a current sensor input and develop an algorithm that uses such input to calculate the real time power consumption. The calculations will not be one hundred percent accurate because analog input errors need to be accounted for, but overall, the obtained values can be very similar to the expected values. The module used is the Pololu ACS714 Current Sensor which features an analog interface perfect to be paired with the ATmega328P. It accepts a bidirectional current input with a magnitude up to 30A. It has high accuracy and reliability with a typical total output error of ±1.5% which is desirable when measuring real time power.

G. Smart Outlet

The smart outlet function is to cut off the current flow of certain appliances as desired by the user. There is an existing circuit that presents all the desired features for this component of the project. The module contains a relay that is responsible for stopping the signal to reach the appliance. Moreover, it can be connected to a microcontroller and be driven by it, and it does not need a separate power supply, which works perfectly with the project microcontroller. In fact, one microcontroller was used to handle the smart outlet and the power monitoring functions. The module can be connected to a single powered device or to a power strip to control multiple loads, which is very helpful in this system in the case the user wants to deactivate more than one appliance at the same time.

IV. Hardware Detail

Each of the stations and their major components outlined in the previous section will now be explained in technical detail. To avoid redundancy the microcontroller unit and the I2C bus will be explained separately since these are part of every station in the system.

A. Microcontroller

For the desired design more than one microcontroller is needed. For the prototype, three microcontrollers were used; however, in a real life situation, more parts will be required. The team decided to work with the same microcontroller in order to make the process smoother; therefore, the desired characteristics included the ability to support analog and PWM inputs, UART and SPI interfaces, low power modes, and optimal speeds

The ATmega328P provides 23 general purpose I/O; including 5 analog pins, 6 PWM pins, an RX/TX combination of pins, and an SCL/SDA combination of pins. Most of the pins can be used to emulate UART serial interfaces needed for the Bluetooth communication. It also includes a watchdog timer needed to automatically reset the system in case a software or hardware failure occurs. In this case the MCUs will be powered by a voltage regulator supply that converts 120VAC to 9VDC for the proper functionality of the microcontrollers.

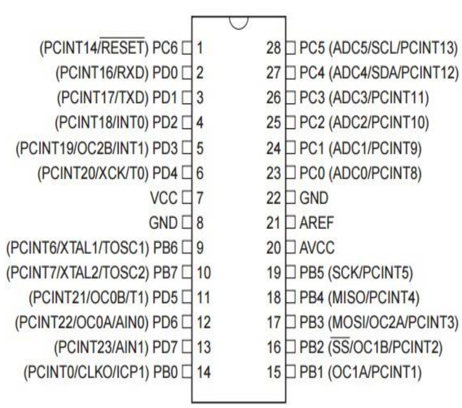


Fig. 2. ATmega328P Pinout (Image Courtesy of Atmel)

B. I2C

The method for communication between the stations is going to be via I2C. This type of bus uses the SDA data line, the SCL clock line and the supply voltage pins in the ATmega328P microcontroller to achieve the communication. One of the advantages of using this among with the Arduino Wire library is the fact that it allows to transmit and receive data from multiple addresses at the same time. Since there are three stations in the system this allows a smooth flow of data and reduces the amount of delays every time a command is sent from one board to the other.

This is useful and user friendly when sending simple information such as a character, yet when transferring something larger like a string things may get a little trickier. One of the main purposes in the program is to send the amount of power read in the Power Monitoring Station and transmit it to the main hub so it can be displayed on an LCD and in the Android App. Figure 3 shows in more detail how this process works. Once the power is read in the form of a integer for both smart outlets, the programs places both of these into a string with a "&" symbol as a separator. Then, the string is placed into an array so each character can be sent. Once the communication has been established and the character has been sent, it is received in the Main Hub station in the form of a string with the ASCII values for each character, this is then placed in an array that separates each ASCII value of the desired numbers with a "&" symbol. The program places each of those values back to their original form into a string. Finally, the separator character is recognized and each value is assigned to its own separate string; this way it can be displayed on one of the LCD screens and the mobile application.

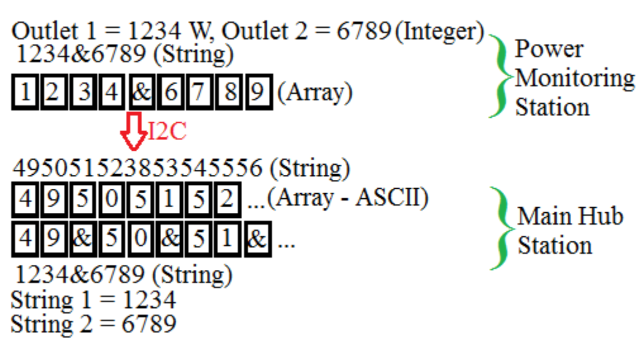


Fig. 3. Transmission of data via I2C

C. Sensors Station

The Sensors Station has as an input physical presence and also amount of light in the room. Moreover, it outputs a percentage of brightness in a lightbulb that is being controlled by the dimmer. At the same time the system is able to recognize the change in power being consumed by the lightbulb and send this to the Main Hub Station. All of the modules on this board can be powered using 5V for the input voltage, the board itself is powered with 9V. The LHI778 PIR sensor contains a dual element pyro-electric ceramic element with Fet source follower connection. Its infrared filter provides high sensitivity with accurate performance at low noise. The sensitivity of this device is modified to suit the user's needs. In this case a delay was added so if there is no movement for a certain period of time it will still send a signal for a little longer. The Phantom YoYo ambient light sensor uses a photo resistor as stated that by altering the resistance in the circuit, emits a voltage signal that varies according its input. The pinout for this particular module is fairly simple to understand as it consists of only VCC, GND and the Voltage that is being outputted into the board as shown in Figure 6. The PIR sensor modules also shares this convenient PIR structure.

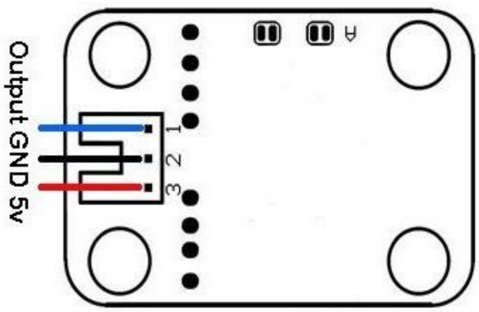


Fig. 4. Ambient Light Sensor Pinout (Courtesy of P. Yoyo)

The Krida Electronics AC Dimmer is a module that allows both PIR and ambient light sensor to control the amount of light being emitted from the lightbulb. It features auto-detection of current, so it automatically senses if the current flowing through is 120V or 240V and if it has a frequency of 60 Hz or 5 Hz respectively. It has terminals for both the AC input and the AC load, a Sync gate for the zero-cross detector output and a Gate pin for the TRIAC (triode for alternating current) gate input. At the same time the Sensors Station features a current monitor sensor that can detect the change of current between the AC load terminal and the lightbulb, hence if the lightbulb dims S.H.A.P.E.R. is smart enough to recognize this alteration in the system.

Finally, this station also connects to the Main Hub Station via I2C since the user is free to modify the brightness of the lightbulb simple by using the app. Keep in mind that once the system detects the I2C communication, it overrides the input received by the PIR and ambient light sensors. Also the changes in the current from the dimmer to the lightbulb are transmitted to the Main Hub so they can be displayed and recorded.

D. Power Monitoring Station

The Power Monitoring stations features mainly the smart outlet control and the measurement of current flow through appliances. This is the main source of power consumption detection in the system using the Pololu ACS714 current sensor.

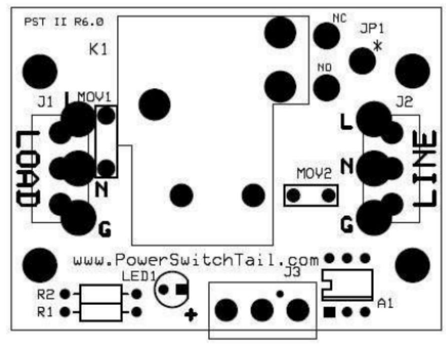


Fig. 5. Smart Outlet Layout (Courtesy of Power Switch Tail)

Beginning with the smart outlet board, as it can be seen in Figure 8 it has a similar terminal connection to the AC line and the AC load like the dimmer module. Moreover, the main component in this board is the relay that will dictate the flow of current through the board. This relay can be set up to be either normally closed (NC) or normally opened (NO) depending on the users preference by simply creating a connection between one of the terminals and the relay. This relay is controlled by an output terminal that connects its pin 1 to the LED pin in the MCU and pin 2 to ground to either open or close the circuit in a user-friendly way.

The current sensor is connected between the AC load terminal in the smart outlet and the appliance as stated in Figure 10. The pinout is very similar to the one found in the PIR and ambient light sensor. It connects to a VCC an GND and it also has a VIOut pin that connects to the MCU as it can see below in Figure 9.

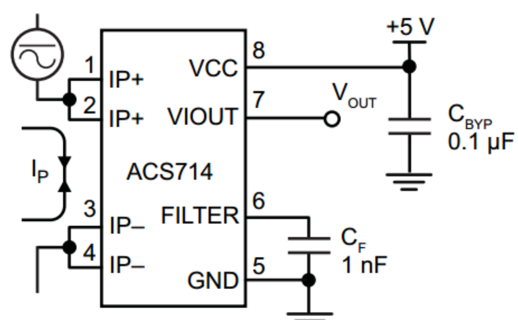


Fig. 6. ACS 714 Pinout (Courtesy of Pololu)

The microcontroller then reads the analog signal from the sensor. This sensor reads sends the values than then are read by the MCU in a scale from 0 counts to 1023 counts in digital and this need to be converted to 0V to 5V scale in analog used by the microcontroller. For the base model the sensitivity is 185 mV/A. Thus, in order to obtain the current range:

$$Min = \frac{0V}{\frac{0.185V}{A}} = 0A$$

$$Max = \frac{5V}{\frac{0.185V}{A}} = 27.03A$$

As it can be seen the current ranges from 0A to 27.03A. Which results in the following formula to be added to the program:

$$Current\ Read = \#\ of\ Counts\ Read * \frac{27.03\ A}{1023\ Counts}$$

Then, based on the power formula:

$$P = I * V$$

$$P_{Consumed\ (W)} = Current\ Read\ (A) * 120\ V$$

The power consumed is obtained by multiplying the current read times 120V, which is the constant voltage supplied by the mains electricity from the wall sockets.

Finally, in order to control the smart outlet and to send the power consumption, the Power Monitoring Station connects with the Main Hub via I2C. The user can either open or close the current flow by using the Android app and also has access to the power consumption records provided by the current sensor.

E. Main Hub Station

This station is programmed to receive the input from the other two stations and output it to either the thermostat, LCD or Android app and store it in a SD card. The first type of data received is if the PIR sensor read if there is someone in the room, if this is true then it sets the temperature to a preferred preset and if there is nobody then it sets it to 78 degrees Fahrenheit. The thermostat features a thermistor, whose resistance varies according to the temperature in the room. Also, the user can manually modify the temperature by pressing the up or down switch buttons that are available to use on the board and can also view the current temperature and the desired temperature on a 16x2 LCD display. Figure 10 exhibits a simulated representation of what the thermostat screen looks like.

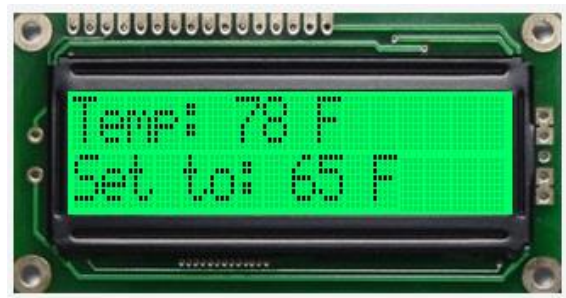


Fig. 7. Simulated Representation of Thermostat LCD

Furthermore, the Main Hub Station receives the power consumption data as stated before and this gets displayed on a 20x4 LCD screen that shows the data obtained from the different current sensors in the system and also adds them up to display the total current energy consumption in the system. In order to display this information correctly the string containing the characters with the power consumed is going to be outputted, yet it is also used to determined where the "W" character that for Watts is going to be positioned. The more figures the string has, the more spaces the "W" will have to move to the right. Thus, the string obtained is converted to a float type so it can be compared to 10, 100, 1,000 and 10,000. If the power obtained is less than 10 then W obtains a position (X,Y) on the LCD. Otherwise if it is below 100 the position (X+1,Y) and it does the same respectively for the other two cases to ensure the power being displayed is as accurate and well-presented as possible; once the data is storage in the SD card the user can easily access it via the Android application, which also allows a convenient access to the power being consumed currently.

As stated before the BlueSMiRF Bluetooth module is the right fit for the system's communication tasks. Table 1 also describes in more detail the label and function of each pin and also what it represents in the module. As expected it breaks down into inputs, outputs and voltage in and ground sections.

The RX-I and TX-I pin is used to receive and send data from another device, which is the reason why it is connected to the TX and RX of the boards respectively. VCC is the module's voltage supply that is redirected to a 3.3V regulator; hence, it can receive an input range from 3.3V to 6V and GND is the ground 0V reference voltage. RTS-O and CTS-I are used for hardware flow control for some serial interfaces, in this case they are not used as they are commonly applicable in more low level applications.

Pin Label	**Pin Function**	**Input/Output/ Power**
RTS-O	Request to send	Output
RX-I	Serial receive	Input
TX-O	Serial transmit	Output
VCC	Voltage supply	Power In
CTS-I	Clear to send	Input
GND	Ground	Power In

Table 1. BlueSMiRF Pin breakdown (Courtesy of Sparkfun)

The way the Bluetooth is going to receive information is by waiting for certain command to be activated once a particular condition is met. For example, if the user desires to activate one of the smart outlets, the app will send a character via serial communication; let it be "1". The Main Hub Station will be waiting for this character to arrive via Bluetooth, once it is obtained it closes the smart outlet circuit so the current flow can go through and the appliance can turn on. The Android app is also used as an input to the system since it can send commands to any of the two other systems; however, the command has to go through the Main Hub first before branching to its final destination.

V. Software Detail

The software component of the S.H.A.P.E.R. will execute based on a priority of tasks. In order for this project to function logically, the lighting system in which the user will use when in a certain room and in conjunction the ambient sensors will activate first. Second, when there is someone in the room, the thermostat will initiate and run. Third, the power monitoring system will initiate to calculate the amount of power a certain device is consuming. If the system does not detect the motion of the user, then there is no point in the running the whole system.

A. Lighting Control System

In every household, lighting is a central commodity within the home. So this was the first priority of the system. S.H.A.P.E.R. allows the user to modify the lighting by using the Krida Electronics dimmer module connected to the Sensor station. As expected, the system is activated using the mobile application, but also by using the PIR motion detection sensor in addition to the ambient light sensor.

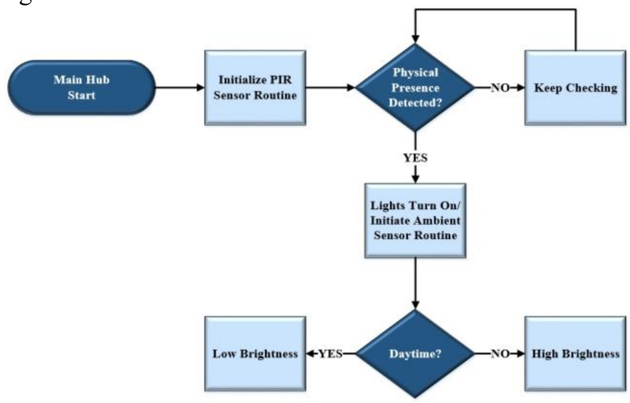


Fig. 8. Lighting System Routine

Fig 5. Demonstrates the logic of when the main hub turns on. The PIR sensor routine will run first thus detecting motion if there is any. This simple design is a lot easier to implement in C code and will set the basis for how the system should run. The PIR sensor activation is set to HIGH if motion is detected, and LOW if no motion is detected. This will send a signal in the "for" loop to set the Light to HIGH which will allow it to turn on.

B. Thermostat System Routine

The thermostat routine is initiated once the PIR sensor detects motion in the room. In the C code, a temperature of 78 Fahrenheit degrees is already pre-programmed as a default for efficiency reasons. When the user enters the room, the user's temperature preference is applied. In the code, the routine will check if the PIR sensor received a HIGH in that there is someone in the room. By checking this loop, it will authorize the thermostat routine to initiate. Again, the process works in a logical step by step method.

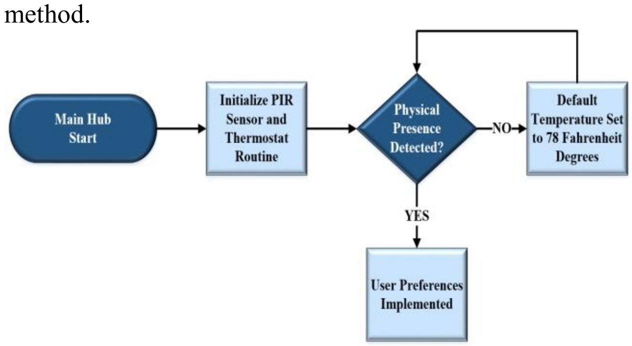


Fig. 9. Thermostat Routine Diagram

The thought process behind this routine is to aid the user in conserving energy since the air conditioning unit is a high power usage item. The code will display the temperature on a simulated thermostat, or an LCD display in this case.

VI. Mobile Application

This section provides an overall view of each individual component. This project has three different circuit designs that conform the whole system. Since each module has its own peripherals, a better presentation of the entire circuit can be achieved by briefly describing each part, which are later interfaced to create the final product.

A. Design Environment

After choosing android as the operating system for us to create our application, the choice was made to use Android Studio as our choice for developing our mobile application. In the process of creating the application, a large amount of information is created for the development of a fully functioning application. The most valuable aspect of Android Studios, is that it will further bring organization to the design of our application. We will be able store our work in packages, project files, and test. We will need to know how much computational resources the application we create is using up.

Android Studio will allow the use of memory and CPU monitoring tool to view the performance of our application. Configured "lint" and other IDE inspections will run when a program is compiled to further verify the execution of our developed code. In Android Studios, full issue explanations are displayed when hovering over problem section of code to ease the developer with correcting things that need to be fixed. The code has been made easily readable for those who may want to view the code in the future and return to the code to make changes. Android Studio will allow to easily annotate parts of the code such as return values, variables, and parameters. Android Studio utilizes a dynamic layout preview that will ease us in the conception of the application. Elements can be added to our design without mistakenly doing away with something created that was desired.

B. Xml Layout

The screens of the mobile application are separated in different activities. Each activity has an xml layout and a java file to control its functionality. The basic layout of the mobile application are developed in an xml layout. In the xml layout, is where the skeleton of the structure is created. This is where the user interface is created to interact with the users. Buttons, text, lists, color, and other features are added here to be displayed to the user.

C. Java Class

The java class is where what is created in the xml file is given functionality. Buttons are given the intent to open other activities in the mobile application. A Bluetooth connection is made allowing data to be taken and sent back to control and display information in the text view and graph.

D. Design Process

In the creation of this mobile application, it is very imperative to emphasize universal usability for this portion of the application. Universal usability means that many people would be able to use the application without having expert knowledge. Novice users would be fully able to understand what choices were available for them to make, almost intuitively.

Fig. 10. Mobile Application Home Screen

E. Application Screens

The home screen is the gateway to other sections of the application and utilizes symbols to increase the intuitiveness for the user.

Lighting control and cooling control screens are developed to give the user the ability to see and manipulate the amount that these separate systems are utilized. The brightness in the room can be adjusted and the current settings on the thermostat can be changed.

The power monitoring screen continually displays the amount of power being consumed by the system. The user can choose to have this information displayed to him in the form of charts and graphs.

One of the main graphs used is a line graph that shows the energy used per appliance in the household over time. This would show what time of day uses the most power from appliances and how the time of day affects the power consumption of each appliance.

To better aid the user in seeing how they are being affected by their energy use, an expense report is given to show them how much their power consumption will cost them.

VII. CONCLUSION

The S.H.A.P.E.R. system serves as a maintainable, affordable and easy to use system for the average day user. This project, even on small scale, has been enlightening to how simple systems such as microcontrollers, elements and software can work together to create something useful and attainable.

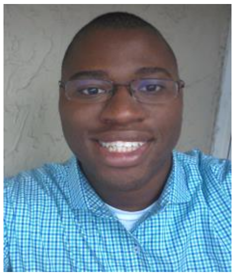
The amount of time in determining how to create a system that would be easy to implement was difficult. We had to consider parts that were adequate but also would fall into the team's budget. Throughout the beginning of the project, certain parts were considered to be useful from the research, but when applied in reality, some components were not applicable to the project.

The world continues to change and engineers such as ourselves will always need to utilize the fundamentals of engineering in order for society to continue to run and be proficient.

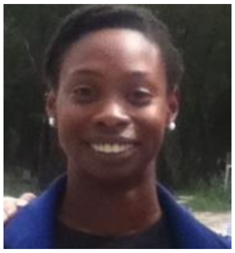
THE ENGINEERS

Juan A. Aleman is a 25-year old graduating Electrical Engineer student who accepted a full-time position at Siemens in Orlando, FL for the Wind Engineer in Training Development Program specializing in Geared and Direct Drive Wind Turbine Generators and the field service operation of windfarms across in The United States and Latin America

Maria D. Alfonso is a 31 year-old graduating Computer Engineering student who has plans on pursuing a career in the embedded systems field or as Software/Computer Engineer with the Disney Company.



Gregory Pierre is a 27 year-old graduating Computer Engineering student. Gregory has an interest and plans to pursue a career in software systems. Plans on pursuing a career working at the illustrious Duke Energy.



Francine Vassell is a 23 year-old graduating Computer Engineering student who has plans to work for CenturyLink as wither a Network Engineer or Operations Technician. She sees that networking technologies such as LANs, MPLS systems, and routing technologies are crucial to the lives of many in a very technological atmosphere.

ACKNOWLEDGMENT

The authors wish to acknowledge the assistance and support of Dr. Samuel Richie and Carmen Henriquez; University of Central Florida

In addition, the authors wish like to give a special thanks to Boeing and Leidos for kindly sponsoring our project.

REFERENCES

[1] Atmel. "ATmega328P." ATmega328P. N.p., Nov. 2015.Web.03Mar.2016. <http://www.atmel.com/devices/atmega328p.aspx>

[2] Sparkfun. "Using the BlueSMiRF." Using the BlueSMiRF. Web. 01 Apr. 2016. <https://learn.sparkfun.com/tutorials/using-the-bluesmirf>.

[3] "Automotive Grade, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 KVRMS Voltage Isolation and a Low-Resistance Current Conductor." ACS714. Allegro MicroSystems, Web. 09 Mar. 2016. <https://www.pololu.com/file/download/ACS714.pdf?file_id=0J196>.

[4] PN PSTK-120 PowerSwitch Tail 120vac Kit PN PSTK-240 PowerSwitch Tail 240vac Kit Rev 6. Web. 02 Apr. 2016 <http://www.powerswitchtail.com/Documents/PST%20Kit%20Assembly%20and%20Application%20Instructions%20R6%207-15-2013