eHome

Matthew Beck, Kyle Ferguson, Travis Helms, Dejan Lee-Hue

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

***Abstract* — The goal of this project is to develop a system that will allow users to take a more active role in reducing their monthly power consumption. The project monitors power use for the whole house as well as specially developed outlets, and converts this information into a monetary cost that is shown on a website and allows the user to track their monthly power bill in real time, helping the user to be more conscious of their power consumption so that they could adjust accordingly. A secondary feature of the project is allowing control of lights, outlets and even door locks from the website. This allows the user to be even more in control of their home power user.**

***Index Terms* — eHome, Remote Home Control, Smart House, Energy Conservation, Microcontroller, XBEE, Power Monitoring.**

I. Introduction

Energy conservation is a very hot topic right now. Household devices are becoming more and more energy-efficient. However, many people are still often shocked at the end of the month when they get their power bill. How did we use so much power? What is using so much? These are some of the questions that this project aims to help answer.

The project contains two types of power monitoring methods. First, it monitors the power use for the entire house. This monitoring will be used to estimate a bill for the current billing cycle, and will be stored to provide a history to the user after the system has been in place for a while. This monitoring is performed by mounting a camera aimed at the residential power meter located outside the user's home. The camera is a wireless IP camera connected to the user's wireless network, and its video stream can be accessed by going to the camera's IP address. The project has a web service that accesses this stream frame-by-frame, and uses an image processing algorithm designed for this project that produces a meter reading from the image. This reading has a reading obtained at the beginning of the billing cycle subtracted from it to get the power consumption for the cycle so far. If the user enters the rates that they are charged per kWh, the rate can be used to find an estimated bill so far for the cycle as well as a projected estimate of the final bill, based on average daily use so far in the cycle. Hopefully being able to see an estimated bill amount instantly on demand will help the system's users to be more thoughtful and active about lowering their power use.

The second type of power monitoring is done on individual outlets specially designed as part of this project. The outlets have current and voltage monitoring equipment built into them that relays back to the system how much power is being used by anything plugged into the outlet. The outlet designed for this project is a standard USA 3-prong wall outlet, so something like a dryer would not be able to monitored in the current state of the project, but it would not be hard at all to scale up to different and more powerful types of outlets. Like the whole-house power monitoring system, the monitoring done on an individual outlet will allow the system to show the user how much power that particular outlet (and therefore, whatever is plugged into that outlet, such as a refrigerator) is using, how much it has cost so far this billing cycle, and a history of its power use. This would allow the user to even more efficiently improve their power consumption by targeting specific appliances or devices.

Power monitoring is only half of the project; the other half is concerned with controlling certain elements of the home remotely. Our system allows the user to turn on or off lights, outlets, and even lock or unlock a door, remotely using the system's website. The website will show the user what state each device is in, and the user can alter that state whenever they wish. This allows a user to save energy by being able to turn off lights or plugged in devices that they may have accidentally left on while they are not home. It even provides some extra security by allowing the user to turn lights on and off while they are away on vacation, to help deter burglaries. While outlets can only be controlled from the website, lights can still be controlled by a physical light switch, and doors can still be locked or unlocked with a key.

Because our project is about conserving power, we wanted to build our components so that they will not add a significant amount of power use to the home. To that end, each of the devices in our project, as well as the central hub, incorporates low-powered parts that run on a 9 volt battery. The system will be able to notify the user if one of the devices does not respond to a command, so if a battery needs to be replaced, the user will know. The devices (outlet, light and door lock), plus the central hub, are each controlled by a microcontroller connected to an Xbee antenna for wireless communication, neither of which uses much power at all, satisfying that goal.

II. SYSTEM DEFINITION

A. Definition

The main component of our system is what we call the central hub. The central hub consists of a microcontroller attached to an Ethernet shield and an Xbee antenna. The central hub acts as a router for our system, connecting to the internet over Ethernet, and transmitting commands and receiving responses from the various devices in the system. The microcontroller we chose to use for all of the devices in our system is the Atmega328, as it meets all of our requirements for each device, and it is fairly cheap. Having all of the devices use the same chip, combined with the relative similarity of each device (excluding the central hub) means that programming for each device will be much easier, since each device will be programmed in nearly the same way.

The devices all communicate with the central hub using IEEE 802.15.4 protocol, commonly known as ZigBee. The ZigBee functionality for each device is provided by an Xbee brand ZigBee antenna incorporated into each device. Individual devices will not be communicating with each other, but rather each device will only communicate with the central hub. Since all of our ZigBee antennas will pick up all other ZigBee transmissions that are within range, a communication protocol has been developed as part of our project that ensures that devices only process transmissions sent from the central hub. The protocol also includes device type and unique addressing elements, ensuring that the central hub knows which device it is sending commands to or receiving responses from.

The commands that the central hub would send are turn on or off and check power use for outlets, toggle for lights, lock or unlock for a door lock, and check state for any device. In all of these cases, a response will be send back to the hub from the device. In some cases, such as check state or check power use, the response will include data, while in other cases it will be merely an acknowledgement, but in either case it allows the central hub to know if the device is still active. If the hub does not receive a response, it will alert the system's web service to send a text alert to the user that the particular device is not responding; this way devices will not malfunction without the user being aware. The central hub communicates with the website and the web service for the system, so if they do not get a response from the central hub, they will alert the user, so the hub is covered for malfunctions as well.

The central hub communicates with the website and the web service over Ethernet. When a user chooses a function on the website, a command is sent to the central hub using its IP address. The hub interprets this command and then sends the appropriate command to the specified device in the system.

The website is where all of the user interaction with our system takes place. On initial system setup, the user will have to go to our website and create an account, then set up their particular set of equipment to be associated with their account. There will be some settings that must also be set, such as determining digit location in a sample image from their meter camera, and entering power use charging rates and billing cycle dates from their power company. All of this, as well as controlling devices, will be done from the website. The site is hosted on a personal web server set up on a group member's computer. Also hosted on that web server are a web service that does the image processing from the meter camera, and a database to keep track of accounts, usage data and more. In a real world implemented situation, these would all of course be hosted on a dedicated server machine.

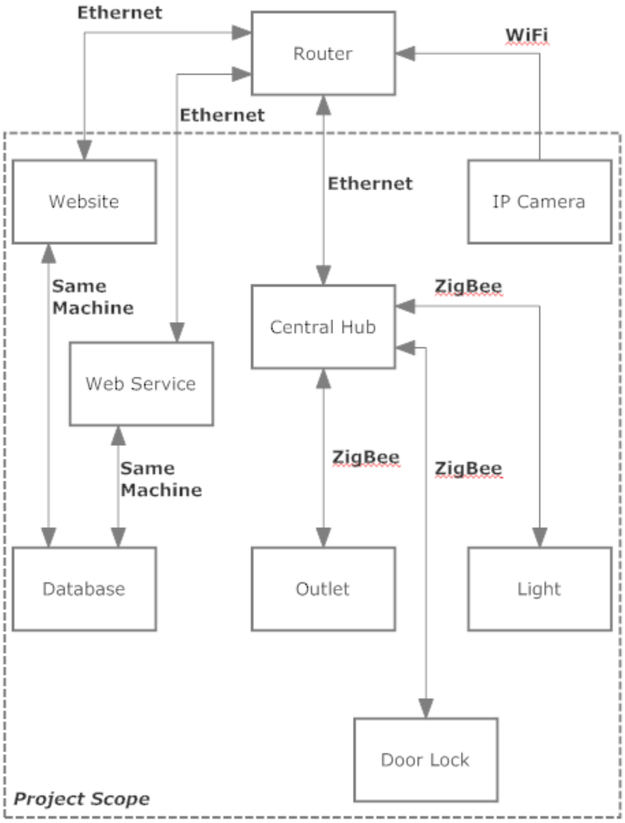


Fig. 1. Block diagram for the final project

B. Objectives

The following are objectives that we wanted to make sure our system met in order to be useful.

1) After initial setup, the meter reading module should run automatically, but if there is an error, the system should be able to alert the user immediately.
2) The same applies to devices; they should work automatically, but the user should be alerted if there is some problem.
3) Because the power meter is outside, the camera for the system should be meant for outdoor use, and have some night-vision capabilities
4) Devices should communicate wirelessly with the central hub, to eliminate the need to run wires all around a home.
5) Meter reading module must be able to check that readings make sense before accepting them. Must also be able to account for rollover if it should happen.
6) Lights should be able to be switched on manually from a switch on the wall as well as from the website.
7) Door locks should still be lockable and unlock able manually using a key, as well as from the website.
8) All devices should default to working like normal versions of themselves if their battery should die, i.e. the outlet should default to being "on" if its battery dies while it was set to "off".
9) The website should provide the user with all interfacing they would need to do with the system, except initial setup and any battery changing.
10) The website should show the estimated power bill in real time.

C. Demonstration Setup

For our demonstration, we have constructed a central hub and one of each device type that will be used in the system: an outlet, a light and a door lock. We are using an IP camera that we purchased and have not modified in any way to serve as our system's meter camera. In order to demonstrate the visual meter reading live, we will have a recorded video of a residential power meter that we will play on a laptop during the demonstration, and we will have the camera pointed at that video.

Because the central hub connects to the internet via Ethernet, and we will be demonstrating the system on campus, we will be using Internet Connection Sharing on a group member's laptop, connecting the laptop to the school's Wi-Fi, then connecting the IP camera to the laptop with an Ethernet cable. While the camera is meant to be wireless in our system, it will have to be used over Ethernet due to the demonstration constraints. We have asked the UCF network administrators, and they have added the MAC address of the central hub to the allowed devices for the demonstration room, so that we can plug it in to an Ethernet port in one of the walls. As long as we communicate with the central hub with another device that is connected to UCF's WPA2 network, it will work. Due to this constraint however, the website and web service will need to be hosted on a laptop that is on campus for communication to work.

For the light device, there is a light bulb; for the outlet, there is an outlet that anything can be plugged in to. For the door lock however, we decided that we didn't want to just show that the latch was locked closed or not, so we built a miniature door, to demonstrate exactly how we envisioned the door lock functioning.

III. Design And Implementation

A. Hardware

For the central hub, we needed three essential things. We needed to be able to communicate over the internet, we needed to communicate wirelessly with our other devices, and we needed a microprocessor to coordinate this communication.

After researching our options for wireless communication, we settled on ZigBee over Wi-Fi or Bluetooth because ZigBee antennas are fairly cheap, ZigBee is compatible with the Arduino development environment we chose, and it satisfies all of our requirements for range and number of supported devices.

For communicating with the internet, Ethernet was chosen as the best choice to connect the central hub to the user's home router. Ethernet does have its disadvantages, such as the equipment for implementing it being fairly expensive and the fact that it's wired means that the user would be required to place the hub near their router. However, using Ethernet eliminates any hassle of configuring the hub to connect to the user's wireless network, which would require the hub to have some sort of direct user interface. The central hub should never need to be handled unless a battery change is needed, so the need to place it near the router should not cause any inconvenience.

When looking at development environments, we settled on the Arduino Uno development board. We liked this environment because it provided all necessary inputs and outputs, and has a simple computer interface for programming and debugging. The Arduino Uno environment is based on the Atmega328 microprocessor, and since the Arduino environment is so suited to our project, it made sense to use the Atmega328 for our project. The Atmega328 provides enough programmable memory to store the non-intensive functionality we would need of a microprocessor, and they are less than three dollars apiece, making them an excellent choice for our project. There are three different physical versions of the Atmega328, two of which must be mounted directly onto a circuit-board to be used. The other version, which we chose, can be inserted into pin headers that are soldered to the board, allowing the microprocessor to be inserted and removed for easy programming via the Arduino Uno environment before moving over to the final boards we will be using when we present.

For the design of the outlet monitor and controller, we had to research various ways to obtain a power reading, while maintaining control of the outlet. Since we knew that the most dangerous part of power calculations is in the current reading, we decided to use a Hall Effect sensor to indirectly measure this for us. It is important to note that since the sensor we chose gave us a range of -75A to 75A of current that can be detected (with a 32 mV output per amp), we decided to wrap the wire being measured around the sensor 4 times, thus effectively quadrupling our sensitivity and raising the precision for the range of 0 to 20 Amps.

Next we needed to determine the voltage seen by the outlet. To do this we decided to use a small transformer to knock down the voltage to a level we felt comfortable measuring. Once the voltage was lowered to around 13 VAC, we connected the output to a voltage divider to further lower the voltage to a level the Atmega328 could handle. Once in range of the Analog to Digital inputs, we connected the output of the voltage divider across a Zener diode (to guarantee that a voltage spike will not burn out our input pin) and then finally through an Envelope Detector to simplify the measurement of the voltage (final circuit shown in figure 4). Once we obtained the voltage and the current seen by the outlet, we must now convert the raw data to a digital power usage.

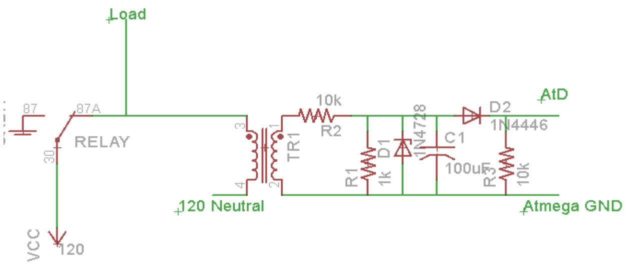


Fig. 2. Voltage measurement final circuit

This conversion required first calibrating the voltage seen by our device in terms of an A/D output. With a meter attached to our outlet, we did a voltage read, and saw that 120V on the meter corresponded to an output of 300 from our converter. We then simply had to multiply this value by .4 to obtain a 120v reading.

Once the voltage was calibrated, we had to calibrate the current reading. To do this we used a 100W light bulb and plugged it into our circuit, while still reading that the outlet was at 120V. We first noted that with 0 current running through the outlet we got an A/D reading (after subtracting the DC offset of 3.5V) of around 14. To handle this “noise” we simply subtracted the noise and continued with the calculation. Since the current of a 100W light bulb is approximately .8 amps, we then mapped this value to the value outputted by the attached A/D converter to .8 amps and came up with a multiplier of .103. This allowed us to convert the voltage seen by the A/D to a current seen by the device! Once both conversions took place, we could finally multiply them together and got around 100W.

The final step to the outlet monitor and controller was that of granting control to the user. To do this we simply connected a relay to the collector of a BJT transistor. Then we connected a resistor (in series with the output of a Digital I/O pin on the microcontroller) to the base of the BJT (with the emitter connected to ground). This effectively grants our circuit the power of digital control.

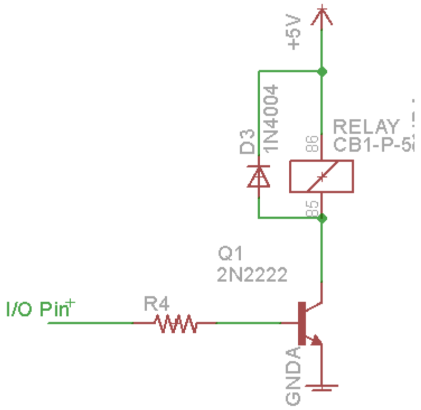


Fig. 3. Outlet Control schematic

Once we decided on a method to control the power to our outlet, we applied the same principle to the design of the light controller. We simply used the same circuit shown in figure 5. The final step was to grant the user physical control via a pushbutton switch. This was done rather easily with the use of a switch and a resistor. One side of the switch was connected across a pull down resistor and ultimately to a digital pin on the microcontroller. The other side was then connected to the 5 volt line on the board. This was an effective way to allow the microcontroller to view a change of state caused by simply pressing the push button switch.

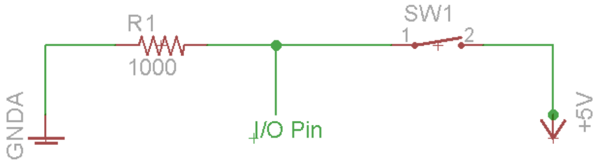


Fig. 4. Light Controlling Switch

The final device our group design was that of an electric door strike controller. We chose to use an electric door strike because we wanted to the user to always have access to their front door via a physical key. An electric door strike simply works by placing a voltage across its coil, forcing the strike to retract, allowing a door to be pulled open. Since the microcontroller was not capable of pushing the required current of 1.5 Amps, we decided to use a MOSFET as a digitally controlled switch as well as an amplifier. We then connected the coil of the strike to the drain of the FET, along with the gate to the digital output of the microcontroller, and finally connected the source of the FET to ground. This circuit allowed for a simple solution to the problem of integrating an inductive element into our system.

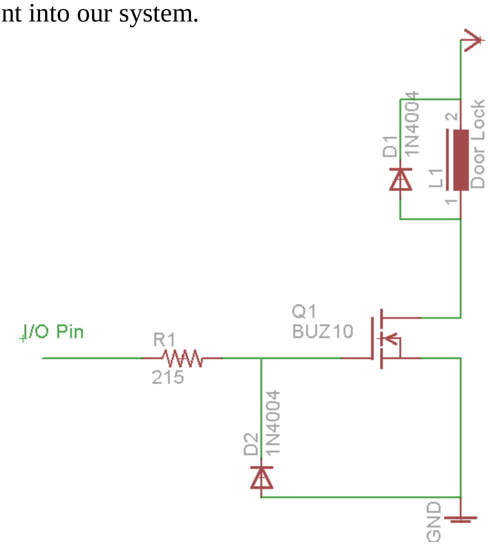


Fig. 5. Door Lock Schematic

B. Microprocessor Programming

Programming the Atmega328 microprocessors on each device was fairly simple. Each device is constantly running in a loop where it checks if it has received any messaged over its Xbee adapter, which is sent to the Atmega328 as Serial communication. If something has been received, it checks if the received byte is the byte designated as the header for all Xbee communication in our system, according to the protocol we have developed. If the byte is our header, then it will check the next byte to see if the communication was sent from the central hub. If it was, then the next byte will be checked to see if the device's address is being specified, meaning this command is meant for this device. If the byte specifies this device, it will check the next byte which specified the action that the device should perform. The Atmega328 will then set or check the appropriate pins to perform the specified action. Afterward, it will send a response back to the central hub via the Xbee as either just an acknowledgement that it successfully performed the requested action, or the response will contain any requested information, such as the state or power use of the device. The device will then go back into the loop of waiting for a Serial input from the Xbee adapter. If the received byte is not the header, or the next two bytes do not specify this device's address, then it will just ignore the bytes that follow and will start back into the loop of checking for the header.

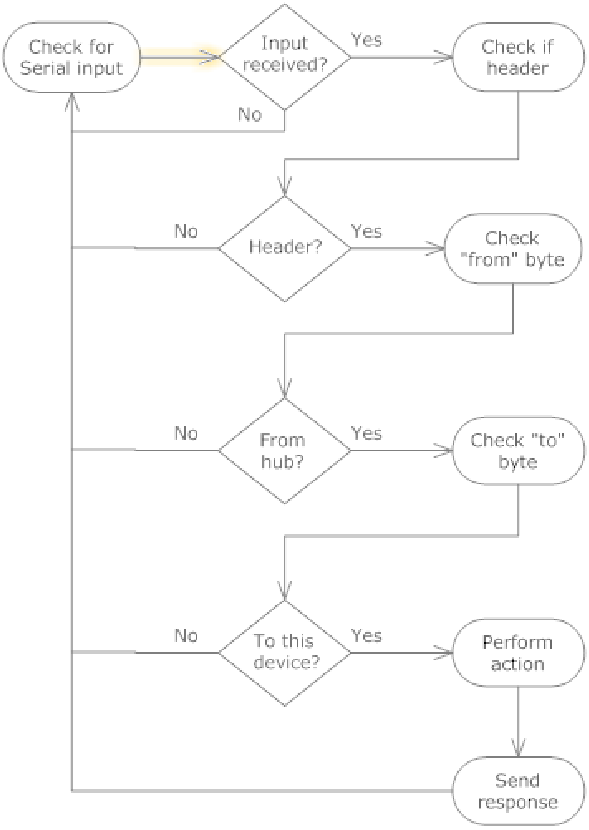


Fig. 6. Generalized programming loop for Atmega328 microprocessor on devices other than the central hub

The central hub's programming is slightly more complicated, as it must interact with both the devices and with the internet. It runs through a loop where it checks for Serial input responses from the devices, using the same protocol of header, from address, to address and command, and it processes these responses accordingly. Some of those responses require the central hub to send information out to the web service, such as the results from a state check or a power use check.

The second part of the loop listens for communication from the internet over the Ethernet adapter. Commands are sent to the central hub by sending an http request to the hub's IP address followed by a slash and then some characters that tell the hub what to do, according to our protocol. This IP address command structure will also have a header byte that the hub will be listening for. Once it receives the header, it will pay attention to and process the command that follows. These commands will come from the website, and will be triggered when the user clicks on an action button representing the task they want to the system to perform. The command that is sent to the central hub will include the function to perform, the address of the device that should perform the action, and a return IP address that the hub can send a response to, which will point to the web service.

C. Web Service

The web service serves primarily as the image processing module that will read the meter from the images obtained from the IP camera, although it does have some interaction with the website.

Originally the image processing was planned to be almost fully automated, with user input required only if some error occurred in the automation. This included automatically determining the style of meter that the user has, as there are three different styles, one where each digit looks like a little clock, a digital meter with seven-segment digits, and an analog rolling-odometer style of meter. That functionality was dropped in favor of having the user just select their meter type on the website, as automating the process proved extremely difficult within time constraints. Additionally, the system was to locate the digits in the image automatically as well, but because that requires very similar techniques to what is required to determine the meter type, it was also dropped due to difficulty and time constraints. Instead, an interface was developed for the website where the user would tell the system where each digit is located. This interface will be discussed further in the website section.

The major core functionality of the image processing module, reading the meter, is still intact. We are currently able to read digital meters and clock-digit meters. Analog meters have proven to be more difficult than would seem at first, and due to that and a lack of availability of that type of meter to test on anyway, that functionality was dropped.

For the a digital digit, the web service runs through a process of converting the image section to grayscale, then isolating the darker segments of the seven-segment digit from the lighter background into a black-and-white image, and finally checks each segment to see if it is shown by detecting an "edge", where the pixel value changes from one pixel to the next in the area where the segment would be shown. This does require that the user be fairly accurate when telling the system where each digit is, as having the digit be heavily off-center would cause problems.

For a clock-digit, the service goes through a similar process of converting the image section to grayscale, then into a black-and-white image. It then calculates the total value of each pixel within each of ten "slices" of the image, each representing where the pointer would be for a certain value. The "slice" with the smallest value (meaning it was the darkest "slice") is considered to be where the pointer is, though cases where two "slices" are adjacent, have nearly the same darkness value, and are the two darkest "slices" are also handled appropriately. The darkest "slice(s)" will determine the value for that digit. This method also requires that the user be fairly accurate at centering each digit in its area.

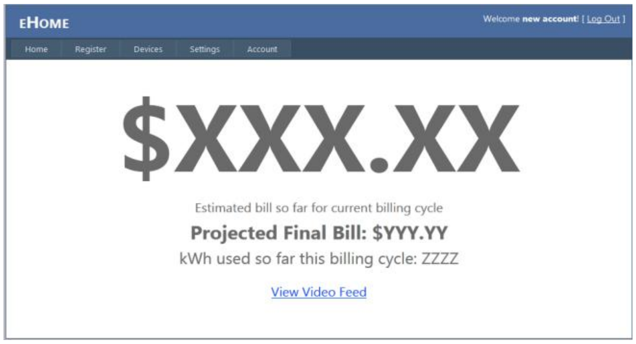
The web service incorporates some error-checking functionality when reading the meter. If it is able to get a successful reading, it will compare this reading to the previous reading observed. If the reading is less than the previous reading or is a large jump up from the last reading, then the meter will try a few more times to get proper reading, before alerting the user if it can't get one. The exception to this rule is if the meter rolls over from the maximum reading, nines in all digits, back to zero. In this case, it is acceptable for the new reading to be less than the last reading, as long as the reading is not too large.

The web service also contains methods for sending messages to the user via text message. The user must enter their mobile phone number and their carrier if they wish to receive text alerts. We have a simple mail server set up for this purpose, and sending a text message works like sending an email. The provider is needed because each provider has a specific domain that text messages must be sent to. A text message is sent to the user by sending an email to the user's ten-digit phone number at the provider's domain. We just have to tell the web service the message to send, who to send to, and it takes care of the rest.

D. Website

The website is the only part of the system that the user will actively interact with, so we wanted to make sure it was easy to use and made sense. The website is where the user will set up their account, register their system with our service and get everything up and running. After everything is set up, the website is where the user will go to get their power use information, to control their devices, and to change any settings on their account, including adding new devices.

The website has several pages to it. First, there is the home page. If the user is logged in, the home page will feature the estimated bill so far for the current billing cycle front and center in fairly large font. Below that, not quite as large, will be the kWh used so far this billing cycle. Both of these controls update dynamically in real time, without the need to refresh the page. This page also offers a link to the user's video feed of their meter, in case they ever want to check it.



$XXX.XX

Estimated bill so far for current billing cycle

Projected Final Bill: $YYY.YY

kWh used so far this billing cycle: ZZZZ

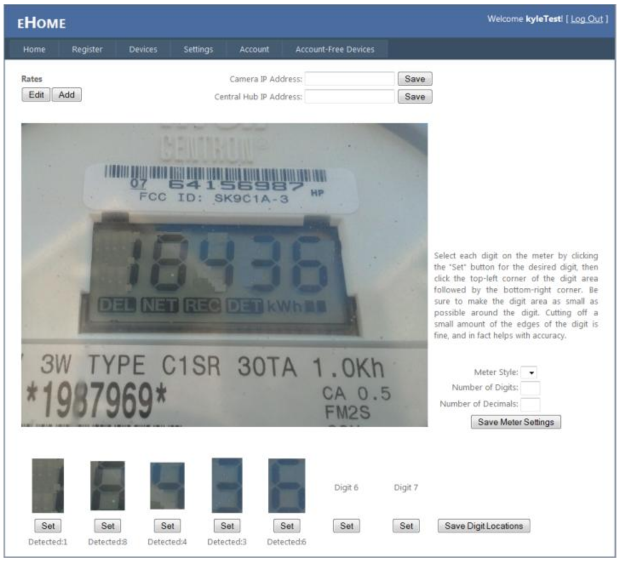
Fig. 7. Web site home page

There is a page dedicated to usage history that will be more useful as time passes while using the system. This page will eventually present the user with a graph that shows their user over a specified length of time. The graph can show power use/monthly bill for the entire home or for each individual outlet in their system.

The website has page dedicated to devices. This page shows all devices on the user's system, and the state of each, along with buttons for controlling these devices. This is also the page where new devices can be added. Each outlet will show the power it has used so far in the billing cycle along with the cost of that power, and offer a link to the history page pre-configured to show the history of the particular outlet.

Another page on the website is the system settings page. This is where the user will configure their system when first set it up, as well as if they need to fix an error related to the meter view from the camera. This page is mostly concerned with setting up the meter reading component, as this is where they will select a meter type, tell the system how many digits their meter has and how many of those, if any, are decimal places. If the meter is clock-digit, they will also need to tell the system which orientation (clockwise or counter-clockwise) each digit is, as the digits alternate between the two from one digit to the adjacent digits due to the mechanics of the meter. The other settings on this page are the IP addresses of the camera and the central hub, so which will need to be set so that the website and web service know where to obtain the video feed from, and where to send hub communications to. The final setting on this page is for the user's power rates, as determined by their power company. The user will input the rate that they are charged per kWh, as well as the maximum and minimum kWh at which the rate is charged. More than one rate can be set, as some companies charge more after passing a certain threshold.

The GUI for showing the system where the digits are is on this page as well. The GUI consists of a sample image taken from the user's camera feed. There is a button for each digit, and the user must press it to set the digit. They will set the digit by first clicking the top-left corner of the digit area, and then clicking the bottom-right corner of the area. At this point, the area that they pointed out will be cropped out and shown by the button, as well as a reading that the web service obtained for the digit, if it was able to get a successful reading at all. If the reading is incorrect, the user will want to re-select the digit area. The process will be repeated until all digit locations are determined. It is up to the user to specify the digits in the correct order to get correct readings. Once all settings have been configured, the user will hit a button on this page that will start the ongoing meter reading loop on the web service.



Camera IP Address:

Central Hub IP Address:

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

Meter Style:

Number of Digits:

Number of Decimals:

Detected:1 Detected:8 Detected:4 Detected:3 Detected:6

Fig. 8. Settings page with digit locations specified.

The final page that is used for system functionality is the account settings page. On this the user will be able to change their password and their mobile phone information. For updating phone information, there is a dropdown with the phone carriers that the user can choose from. We do not currently have a method for the user to manually enter their carrier's text domain, so we can only send texts to phone numbers on the pre-defined carriers. When the user wants to update their phone number information, they must first verify their phone number by entering a verification code that will be randomly generated and then sent to the phone number that they entered. This is to prevent people from being able to cause unwanted text alerts to random phone numbers, and to prevent accidentally saving the wrong number.

There are also two other pages, a registration page and a login page. If a user is not logged in, then the only page that they will have access to is an information-free version of the homepage, the registration page, and the login page. Clicking the links to any other pages will redirect the user to the login page. The login page offers an option to keep the user logged in, which would prevent the user from having to login every time they visited the page. This option is ideal should only be used on a non-publicly shared computer.

E. Database

All data that needs to be stored for any amount of time beyond its initial creation is stored in our database. The database is a Microsoft SQL Server Express database designed specifically for this project. The database contains all of the tables that the website uses for user accounts, and there are tables that link up to this table that tracks all other information about that particular user's system. This data includes the IP addresses to use to communicate with the central hub and to access the camera feed, the locations of each digit in a meter image, the meter style, the latest reading, first reading of the current billing cycle, a history of monthly power use, billing rates and much more.

IV. Conclusion

Our project consists of several components working together to help reduce energy waste one household at a time. We have hardware for turning things on and off when needed, and software that tells the hardware what to do. We have software that reads a power meter visually to track the household's power user, and we have a website that ties everything together and provides the user with an interface for the system. The idea of this project is not new, and there are even forms of this type of system already available to consumers, but we hope that we have found an interesting way to do it.

Biography

Matthew Beck will be graduating with a bachelor's degree in Electrical Engineering from the University of Central Florida. He is currently employed by UCF as a Supplemental Instruction Leader for Computer Science. Matt is currently developing a number of communication technologies, and is actively seeking capital to continue the development and marketing for these ongoing projects.

Kyle Ferguson will be graduating with a bachelor's degree in Computer Engineering from the University of Central Florida. He is currently involved in the College Work Experience Program at Lockheed Martin MFC, and plans to continue working there after he graduates. He also plans to get back into activities that he has not had much time for during his final few semesters, such as spending time with friends, video games and sleeping.

Travis Helms will be graduating with a bachelor's degree in Electrical Engineering from the University of Central Florida. He has accepted a Digital Design Engineer Job with Harris Corporation in Palm Bay, Florida. He plans to move as close to the beach as possible and continue surfing in his free time!

Dejan Lee Hue will be graduating with a bachelor's degree in Computer Engineering from the University of Central Florida. He is currently seeking employment as a programmer and hopes to remain in the Orlando area after Graduation.

Acknowledgements

We would like to thank Dr. Samuel Richie for his guidance throughout the project, and our friends and family members for putting up with and even supporting us on our quests to earn our degrees.