LSU Remote Outlet Management System

RISC Takers: Paul Banks, Yorick Robinson, Evan Tu, Hans Weggeman
Faculty Advisor: Ashok Srivastava Instructor: John Scalzo

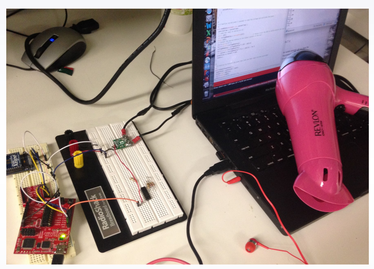
The Problem: As the global population continues to soar, our search for meeting the energy demands of tomorrow are becoming more and more challenging. Throughout the day, we flip, press, or touch our way to use energy in anything from iPods to cars. Unfortunately, we often give little thought about where it all comes from and how wasteful many of our habits are. According to the Department of Energy, roughly $ 4 billion dollars a year are wasted on average in home's annual power usage. For the typical family, this equates to a minimum of $130 a year, which is "…--more than some people spend on a typical month's electric bill" (Raphael). As consumers, people should have more control on how much they use and when they actually decide to use it.

Our Solution: Our system, The Remote Outlet Management System (ROM System), is a powerful and convenient solution to this problem. A user in our system will be able to remotely turn their electronic devices on and off, monitor device power usage and active times. Expanding on the ability to turn devices on and off, a user can set a schedule for devices in the form of a date and time, and turn a device on for an arbitrary amount of time. All of these functions can be conveniently accessed through any web-connected device, and we have also developed an iOS application mirroring the website.

Plug-In and Main Hub

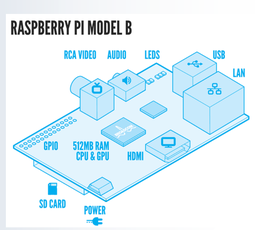
The Plug-In device of the Remote Outlet Management System uses multiple technologies to enable users to have wireless command of any 120Vac outlet they desire. To maintain the most compact possible design, the various modules and supporting circuitry within the plug-in are powered by means of a transformerless power supply. This type of power supply is capable of using the 120Vac source to produce a steady DC output voltage, while maintaining cost and size efficiency.

The microcontroller housed in the Plug-In is the TI MSP430G2553. There are three different PCB boards within the plug-in, two of which were designed and the other purchased for cost minimization. The plug-in has power management capability by enabling the users to see which devices are active and calculating the power consumed by the monitored device. To maintain isolation from the line to the device and the peripheral devices, an ACS711EX Hall Effect current sensing transducer uses the magnetic field produced by the current to the device to produce a desired analog output used to compute power consumption values. This output is sent to an Analog-to-Digital Converter pin on the MSP430 for sampling.



The switching of a device is done by the implementation of a power triac driven by an opto-coupler (MOC3043). The zero crossing characteristic of the opto-coupler ensures isolation of peripheral devices from the mains connected to the device. The signal to drive this device is sent from a GPIO pin on the MSP430 after receiving a signal from the user. The way the Plug-In communicates with the Main Hub is through an X-Bee RF module. They are connected through serial and the X-Bee acts as a wireless serial connection from the Plug-In to the Main Hub. Finally, the casing of the design was designed using freeCAD with dimensions 113mm X 63mm X 40mm.

For our Main Hub the primary component is the Raspberry Pi, a very compact and powerful microprocessor. We have installed a Linux distribution on the Pi as this will be where the database is stored. The Pi has an integrated Ethernet port and we have installed a USB Wi-Fi dongle to allow users either a wireless or wired connection for their Main Hub. Connected along the serial lines will be an X-Bee RF module. This is what will allow the Main Hub to communicate with all of the ROM Plug-Ins. This X-Bee will be configured to work as a coordinator, meaning it manages the network of X-Bees.



Conclusion

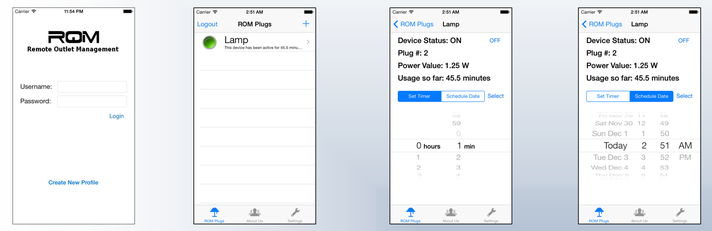
The current sensing circuit of our project works as intended; frequently sampling the output of the Hall Effect Sensor and sending the average data over serial lines. Testing on the power supply, both physical and virtual, has shown full functionality, however the lack of PCB fabrication supplies has forced us to forgo this part of the project, and we will be using an external power supply for demonstration purposes. Because the switching device contains surface mount components, it was not possible to physically test the functionality, however virtual simulations depicted proven concept. X-Bees have successfully communicated to each other in simulated environments with correct data and readings being sent. The ROM application is functional, excluding an interface with the database. The application successfully serializes and parses JSON files to distribute user and ROM plug data throughout the application. Home automation systems with basic functionality of scheduling and remotely turning on and off devices is only the beginning to what can be achieved. Conditional statements, event based driven logic, and even PID tuned automation are technologies that will be implemented in the future.

Database

Mongo DB, Express, Angular JS, and Node enables a complete single web-app with easy and flexible data validation. MongooseJS allows for code defined schemas utilized to create a home page, log-in creation, user validation, and a user page. The database was designed to store usernames, passwords, load readings, and plug-in connections associated with each account. The user page enables user to add, modify, and schedule plug in devices. The database then sends the appropriated signals via X-bee wireless communication modules in order to receive current load read outs from your plug given a web connected device. These changes are then reflected server-side in order to update all of your devices simultaneously. Security was an important issue given the nature of the project. In the design, bcrypt encryption method was utilized in order to safely store username and passwords.

App

The ROM application is the interface between users and the ROM hub, which stores data related to the individual plug-ins. Users will be able to create user profiles with which they can monitor and control household devices connected to each plug-in. Once logged in, users can also add new plugs through the application with a unique identifier. Each plug-in that is associated with a user can be switched ON and OFF through the app either immediately or at a future date and time.



This is accomplished with the exchange of JSON files between the application and the server housing the database. Using native iOS libraries, a connection to the server is established, the JSON files are periodically read in and the persistent store within the iOS devices memory is updated with any changes to power values. Likewise, as a user makes changes to any of the properties associated with each plug-in, the files are overwritten on the server. This is how data synchronization is maintained between the database and an iOS device.