5-2016

Energy Demand Management

Doug W. Bouler III
dbouler@vols.utk.edu

Alex Wortham

Asanga Bandara

Alex Chaloux

Control and Monitoring of Residential Loads

11/15/15

Team SP15-01
Doug Bouler, Alex Wortham, Asanga Bandara, Alex Chaloux, & Chauncey Meade

Customer
Harshal Upadhye – EPRI

Table of Contents

Executive Summary

As modern civilization becomes increasingly saturated by electronic devices, an issue of infrastructure stability has arisen. The electrical distribution network, known as “the grid” has become burdened by demand driven by the rise in electronic usage. In East Tennessee power is often developed through the use of hydroelectric dams, coal-burning steam plants, or nuclear reactors. Our current electrical distribution system is based on demand response. If a lightbulb is turned on, a demand is created. The demand created from the lightbulb must be met through power distribution. If the power source is a steam turbine, it must speed up to accommodate for the demand.

Residential power bills in the Southeast are typically calculated by a set price per kilowatt hour. In many urban cities, not only is energy (power over time) factored into the cost of power, but also demand charges. Instantaneous power can cause major instability in the grid and therefore peak times for electric usage are more costly than off-peak times. Therefore, giving customers incentive to use less power over a short interval is beneficial to both the power companies and the consumer.

Our project focuses on alleviating potential stress on the grid by stifling perturbations in residential demand. Common appliances such as dishwashers, water heaters, and air conditioners can strain the grid if used simultaneously. The goal of our project is to control these common appliances in a way that will not cause instantaneous stress to the grid while informing consumers of the savings that they could potentially accrue as a function of their demand usage.

We believe that both of our goals have been achieved. We have successfully been able to design and implement a system that uses a feedback network in order to control the demand usage of residential loads. Due to proprietary information and the scope of this project, it was necessary for our team to design these residential loads ourselves. By using load curves acquired from a research paper about common appliances, we were able to measure and replicate the power usage of these devices within a five percent error. We also met our goal of informing the consumer of his or her demand usage. By using an internet interface, customers can start and stop machines, see their current demand usage, and track their electric bills based on energy consumption and demand charges.

Our project flowed very smoothly. After our customer, Harshal, had given us clear direction, we immediately began brainstorming of solutions. There were a few disagreements about how to implement the hardware most effectively, but in the end we always came together to make a decision. Team members stayed highly involved and effectively communicated needs to the team, which in turn allowed us to finish the project earlier than expected.

Design Process

The first step of our design process was the formation of our requirements. It was very important to our team to create our requirements such that they not only described our goals, but also laid the framework for how we would complete our solution.

We felt that each requirement on our list, especially in the software section, represented a modular piece of work that could be condensed to a binary completion status: complete or incomplete. Because our requirements were set up in this manner, the decomposition of our project into manageable pieces began first and foremost with our requirements document. For the larger requirements, we needed to break apart the more daunting pieces into several smaller portions that would be more workable. We accomplished this using frequent meetings and team communication, and by striving to finish portions of our project ahead of schedule. By finishing things as quickly as possible, we allowed ourselves more time to dissect and conquer the larger tasks.

For example, our first requirement (1.1) is a very large task. It can be determined whether or not the requirement has been met, but that particular requirement does not self-decompose without a little bit more effort. To break requirements such as this one into smaller pieces, we allowed those with knowledge in the particular area to discuss ways to assign sub-tasks in order to fulfill the requirement. For requirement 1.1, Asanga and Alex Wortham were instrumental in finding hardware that could effectively simulate devices. Since those two individuals researched and found pieces that would work together well, they were then able demonstrate how we could break the requirement apart into smaller tasks to implement each of those pieces. This example repeated itself frequently, and since our team was constantly meeting and communicating we were able to stay on the same page and effectively complete our required tasks.

Often, the tasks that we created to meet our requirements required knowledge that we did not have readily available. This was especially true for the simulation of appliances. Due to the large number of electronic devices available, it can be a daunting task to choose and implement specific pieces of hardware into a solution. Specifically, our team did a lot of research on the microcontroller that we would use, the simulation circuit, and the measuring of power from our simulation.

For our microcontroller, we considered several options, including a BeagleBone Black (which we ultimately decided on), a Raspberry Pi, and Arduino boards. Overall, the BeagleBone Black and the Raspberry Pi were more suited to our requirements than the Arduinos. The two products are very similar, but the BeagleBone Black offered many more pin connections than the Raspberry Pi, and required less initial setup. BeagleBone also packs more processing power into their controller, which was an added bonus since we are using the controller to run a web application to display information and accept input from the user.

The simulation circuit was a much different issue to solve, because it was a more open-ended problem. With the controller, it was very easy for us to come up with a list of necessities and pick the product that best met those requirements. With the simulation circuit, there were many more possibilities and fewer restrictions. Our team researched some different options, including using motors to create an electrical demand similar to the simulated demand. However, a simulation of this sort would have been difficult to implement properly, and our goal is not primarily to create an interesting simulation. Instead, we needed to create the simplest simulation possible so that we could spend more of our time on the control of the simulated demands, which is the most important piece of our project, both to our customer and the industry. We ultimately came to the conclusion to use an I^2C chip coupled with sets of resistors and LEDs to create demands proportional to the power used by each simulated device.

The measurement of our simulated devices is an extremely important part of our project. In order to accurately control our devices, we must get accurate measurements from the simulation circuitry. When trying to decide how to measure our control diagram, we chose a current sensor that was readily available to us. This sensor outputs a voltage that can be read and interpreted by the BeagleBone Black, making it eligible to use in our system. We then designed our simulation circuitry to produce currents that were within the range of the sensor. Alternatively, we could have been more selective with our current sensors, and probably produced slightly more accurate results. However, the results that we were able to obtain were accurate enough for our demonstration and to satisfy our customer. Therefore, the extra work to find, purchase, and design around a different current sensor would not have been warranted.

The core of our project is the control and monitoring of residential electrical demands. In order for our project to be a success, we needed to be able to provide residential customers with enough information and control to make informed decisions about how the appliances in their home should not be run concurrently in order to avoid demand charges and strain on the grid. In order to accomplish this, we needed a hardware device that was capable of reading measured power consumption from four different appliances to predict future power consumption, and prevent the appliances from turning on if doing so would exceed a user-set demand threshold. In other words, the user is able to determine the maximum instantaneous demand that they want to occur, and our system is responsible for ensuring that the demand threshold is not exceeded. In order to accomplish this, we would also need a user interface to provide users with the ability to input this information and to inform users of the necessary factors to allow them to decide on a suitable demand threshold.

Since our device is also responsible for ensuring that no appliance turns on when it will cause the demand threshold to be exceeded, we needed a way to prevent appliances from turning on. This needs to be implemented in hardware, because in practice our device will not be able to communicate with the appliances that it is controlling. Because these appliances cannot respond to input from our device, we would instead need to turn the power off for the appliances that are not allowed to start by using a relay.

To gather the necessary information to make decisions about which appliances are allowed to start at any given time, we also need to measure the power that the appliances use. Keeping a record of demand information for a given appliance will allow our device to predict future demand, which is necessary for our device to make an informed decision about which appliances are allowed to start running at any given time.

Lastly, we do not have access to real appliances to use for our project due to time, money, space, and project scope considerations. Therefore, we also need a way to simulate each appliance that we will attempt to control. For our project, we are simulating an air conditioner, water heater, clothes dryer, and dishwasher because these are four of the largest electrical loads in a typical residence. We would need to create a hardware simulation of the demand that these devices draw so that our controller can measure the demand and respond accordingly as it would if it was connected to real appliances.

This essentially divides our design into five components: the user interface, the control of appliances, the measurement of demand from appliances, the simulation of four appliances, and the software to control everything.

For our user interface, it is important to provide users with the necessary information and capability to make informed decisions about the demand threshold that they want to set for their household. In order to accomplish this, we created a dashboard-style web page that can be seen in Figure 1.

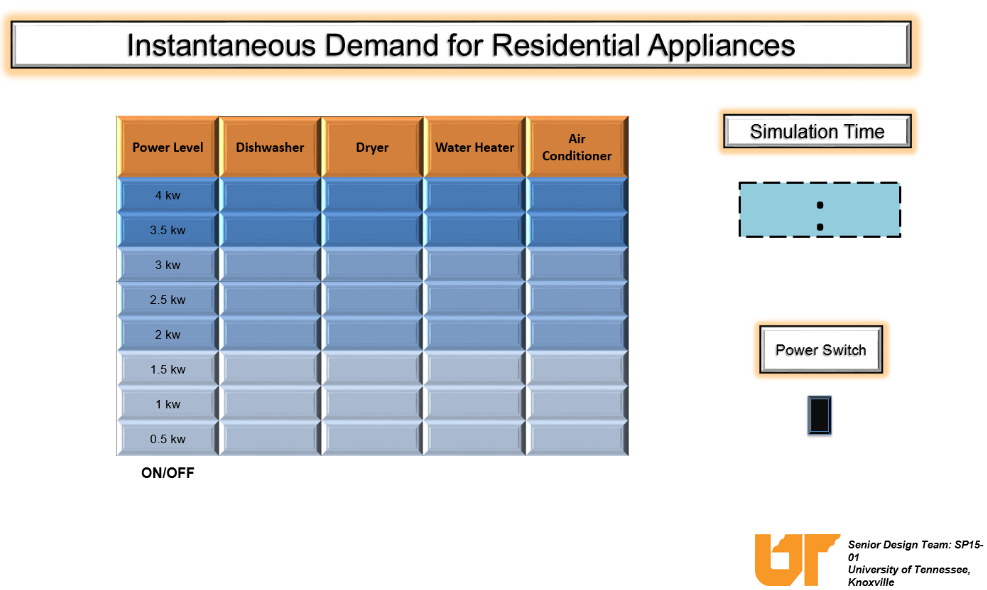

Figure 1.

In the billing breakdown, we are displaying the peak demand for the current billing cycle, as well as the total energy usage for the current billing cycle. These values are then converted to dollars using a preset charge per kilowatt-hour of energy and a charge per kilowatt of peak demand. These two dollar values are then added together to calculate the expected amount due when this user receives their bill from the utility company. The average residential customer will not have a working knowledge of electrical systems, especially the concept of peak demand. This could cause frustration for consumers, so our breakdown is intended to assist users with understanding the charges on their bill. Understanding these charges will allow users to make a more informed decision about the peak demand that they want to allow.

The next section of our interface is the demand management section. This section is where the user will actually set their desired demand threshold. The slider on the left side of that section currently allows a user to set a threshold between 2 kW and 10 kW. We used this range because 2 kW is the minimum threshold that we could use and still be able to start each device, and 10 kW is more than all of our appliances will collectively demand when running simultaneously. Next to the slider is a graphic that shows how the demand threshold will translate to dollars per month, if the demand threshold is met exactly. This allows the user to weigh the cost of the desired peak demand with the convenience of being able to run appliances simultaneously. On the right side of this section is a graphical representation of the cost associated with the actual peak demand from the controlled appliances. This allows the user to see how close the actual peak demand is to the desired demand threshold.

The final section of our web interface is a summary of the total energy consumption from the four controlled appliances. This section shows information about how much total energy each appliance has used in a table, and as a pie chart. This information is not inherently necessary to the user, but it can be helpful to understand the total energy portion of the power bill, even though users are already familiar with this style of billing. Since we have the information available already for use with the billing calculation, it was not difficult to provide the user with a little bit more information about the breakdown of their energy consumption.

Altogether, we feel that the web interface provided gives the user enough information to make an informed decision about the desired demand threshold. The user also has the ability to set the demand quickly and easily, which meets the requirements for our user interface.

(website.png) needs to be included here.

Now that we have accepted the user's desired demand threshold, our device must ensure that that threshold is not exceeded during operation of the controlled appliances. In order to accomplish this task, we use a prediction for each appliance's demand curve during a normal operation. This prediction is taken from demand data that was collected the last time the appliance ran. At any given time, our software can plot the demand curve of a given appliance aggregated with the demand curves for

appliances that are currently running. This will yield a predicted total demand curve if the given appliance were to start at the given instant. Using this information, our software can ensure that the demand threshold is not exceeded.

To ensure that the demand threshold is not exceeded, we needed to calculate the peak demand in the same way that a utility company will calculate the peak demand. A typical utility company does not measure the peak demand at every instant, but instead measures peak demand in fifteen minute windows. For example, if a single appliance runs with a 4 kW demand for 7.5 minutes, and then does not run at all for 7.5 minutes, the calculated average demand for that period of time will be 2 kW. Our software is set up to calculate the predicted peak demand in the same manner. Therefore, when our software is predicting the peak demand for an appliance, it checks the average demand for 15 minute windows, and ensures that the maximum average demand for any 15 minute window does not exceed the demand threshold. If the maximum average demand from any 15 window will exceed the specified threshold if a given appliance is started, then that appliance will not be allowed to start. In order to accomplish this, we are using relays to disconnect the appliances that we determine are not allowed to start. In the real world, we would use relays in the same manner to disconnect appliances, ensuring that they cannot start and cause the demand threshold to be exceeded.

To accurately predict the demand for appliances, our device measures the demand from each appliance when it is running, and stores that information. For our measurements, we used a current sensor that was connected to our simulation. Since the voltage of the simulation circuits was known, we did not need to measure the voltage for our simulation. In practice, our device would need to measure power using both current and voltage. This would allow our device to function properly even if the supply voltage did not remain constant, giving a more accurate reading.

In order to properly demonstrate the measuring and controlling capability, we needed to design a simulation to work in conjunction with our device. This simulation needed to produce demand patterns that mirror the demand of actual appliances, and we needed to be able to start and disable appliances using our device. To accomplish this, we designed a network of resistors that could be used to generate a variable demand. We used a microcontroller to open and close relays in order to generate a demand curve that could be measured by our device. Each appliance simulation has 8 relays, and for each relay that is closed the current in our simulator circuit increases linearly. This resistor network effectively gives us the ability to simulate a single demand curve with a step function with 8 possible values, in addition to zero. For our simulation, we scaled each step to represent 500 W of power through a real appliance. Therefore, each simulated appliance can represent a real demand of up to 4 kW. We put together four of the resistor networks, each equipped with a current sensor that fed information into our device. As our simulator starts an appliance and varies the demand that the appliance is drawing, our device will measure that information and use it to ensure that the demand threshold is not exceeded.

Collectively, the user interface, controller, and measuring equipment can assist a residential customer with the monitoring and control of large electrical demands within their home. The simulation allows us to demonstrate a functioning prototype with real information, proving that our device would work in a real-world home. We feel that our system provides residential customers with enough information to make an informed decision about the demand threshold, and our system responds correctly to user input and doesn't allow the set threshold to be exceeded by the major appliances in a home. This device, minus the simulator, could be installed in a home as is, and we believe that it would function as intended and to the satisfaction of our customer.

To evaluate whether our design properly met our requirements, we needed to go step by step through each of our requirements and verify each one. This process was simplified greatly because of the way that we organized our requirements. Our final testing process was very straightforward: for each requirement, we needed to ensure that the desired output or functionality was present, accurate, and worked as expected.

Testing our hardware was relatively simple. Once we had the simulation circuits set up, we were able to use our simulating BeagleBone Black controller to manually set relays, and ensure that the circuits responded as expected. Each individual path in our circuits has an LED that we were able to use for feedback and demonstration purposes. We began using very simple tests, and then expanded to testing our circuit with the simulated appliance demand curve to ensure that we got the expected outcome.

From there, we added the current sensors to each circuit and connected them to our "management" BeagleBone controller. We were able to read in values and determine whether or not they were correct, since we knew that the simulation circuits were working as expected. We were also able to compare output from our current sensors with measurements from an ammeter. This allowed us to verify each current sensor's accuracy.

After we knew that our current sensors were working correctly, we were able to test the software. To test the software, we began by testing very small pieces of code, and after we verified each piece of code, we were able to add to it and ensure that each step of creating our device functioned properly. Once we were deep enough into this iterative process, we were able to ensure that entire requirements were met. The key to testing our software was ensuring that each time we changed a piece of code, we tested to make sure that our changes had not corrupted anything else. By testing carefully and often, we were able to avoid major problems with our device and simulations.

Our project is successfully completed according to our requirements document. The primary result of our project is a device that can monitor and control residential appliances when connected to current sensors and relays that enable appliances to be shut off. This device also provides a user-friendly interface to allow a user to set a maximum allowable demand. In addition, we have provided the ability to simulate four appliances within a residence. This simulation assists in the testing and demonstration of our device. Collectively, the results fulfill the specified requirements.

Lessons Learned

Along the way our team made mistakes, but we always learned something from those mistakes. As an example, using a multimeter in a very tight space with several open wires will inevitably end with something destroyed. In that case, it was our microcontroller. Overall, effective communication between hardware development and software development was key. Asanga, our hardware tester and builder, was in constant communication with Alex Wortham, our software developer, making the transition from paper to reality extremely comfortable. Rarely have I ever seen a prototype work so well. One of the main things that we have learned as a group is that diversity is key to any project. One electrical engineer, three computer scientists, and a computer scientist brought this project to completion with knowledge gained from the other disciplines.

In the future, we would like to be able to learn even more from each other. Through the pressures of school and work, we weren’t always able to know exactly what each team member had created. The software team asked many questions about hardware, as did the hardware team about software, but given more time and less stress we feel like we could have taken more ownership of this project by knowing more about all of the pieces.

Presenting was a big part of this project and we feel that we prepared well and presented effectively. We learned from last year through experience who was able to communicate our ideas most effectively, but we still wanted for everyone to remain involved. In future projects we would spend more time preparing presentations as a team so that every member of the team would feel included in making the slides and knowing what they should say in order to best communicate information to our audience.

Team Member Contributions

Roles

Doug Bouler – Team Lead, Hardware Designer, Lead Presenter

Asanga Bandara – Hardware Tester, Solutions Architect, Integrations Expert

Alex Wortham – Software Development, Librarian

Alex Chaloux – Software Tester, Lead Writer

Chauncey Meade – Software Evaluator, Graphics Coordinator

Responsibilities

Doug Bouler was responsible for all administrative activities which include: MBO reports, presentations, setting meeting times, communication with the customer, and dividing responsibility among team members. He was also responsible for designing the hardware to be used in this device. Doug's responsibilities also included designing presentations and creating graphics that could be used in papers.

Asanga Bandara was responsible for testing and building the hardware used in the project. He also was responsible for integrating the interface between hardware in software. His responsibilities also included constant contact with the software team to insure that the inclusion of hardware to software would be smooth and succinct.

Alex Wortham was responsible for designing software capable of reading, writing, and manipulating data to meet the specifications of our customer. He was also responsible for maintaining contact with Asanga in order to make transitions between hardware and software as easy as possible. Alex Wortham was also responsible for documenting changes to software that would be beneficial for anyone planning to continue this project in the future.

Alex Chaloux was responsible for testing software mostly written by Alex Wortham. He communicated with Alex Wortham any and all performance issues or errors in the software development process. Alex Chaloux was also responsible for writing the majority of the team papers.

Chauncey Meade was responsible for evaluating the integration between hardware and software. He was also responsible for generating graphics that could be used in the user interface created by Alex Wortham.

Contributions

Doug Bouler

- Managed team meetings and customer meetings
- Prepared PowerPoint presentations
- Designed display board
- Managed team responsibilities
- Consistently monitored team progress
- Designed and simulated "simulated loads"

Asanga Bandara

- Created hardware for simulation loads
- Soldered simulation clock interface
- Created hardware for display board
- Produced circuits necessary for interfacing between software and hardware (I^2C, current monitor, etc.)
- Highly involved in team meetings
- Contributed highly to hardware design and production

Alex Wortham

- Wrote software for entirety of project
- Instrumental in designing final board
- Highly involved in team meetings
- Wrote significant documentation detailing software
- Formatted most documents to LaTex

Alex Chaloux

- Wrote majority of team papers including societal impact paper and final report
- Helped troubleshoot software issues during team meetings
- Proofread and edited documents and presentations
- Highly involved in team meetings and customer interaction

Chauncey Meade

- Came to a few meetings at the beginning of the semester
- Came to all customer meetings
- Wrote introduction paragraph of requirements document
- Not involved in design process (hardware or software)
- Did not fulfill expectations when asked to write parts of final report and to help design graphics for user interface

Team and Customer Approval

I, ________________________, declare that I am satisfied with the contents of this project and the contents of this final report.

Customer - *Harshal Upadhye*

____

Team Lead – Doug Bouler

____

Hardware Tester – Asanga Bandara

____

Software Developer – Alex Wortham

____

Software Tester – Alex Chaloux

____

Software Evaluator – Chauncey Meade

Appendix

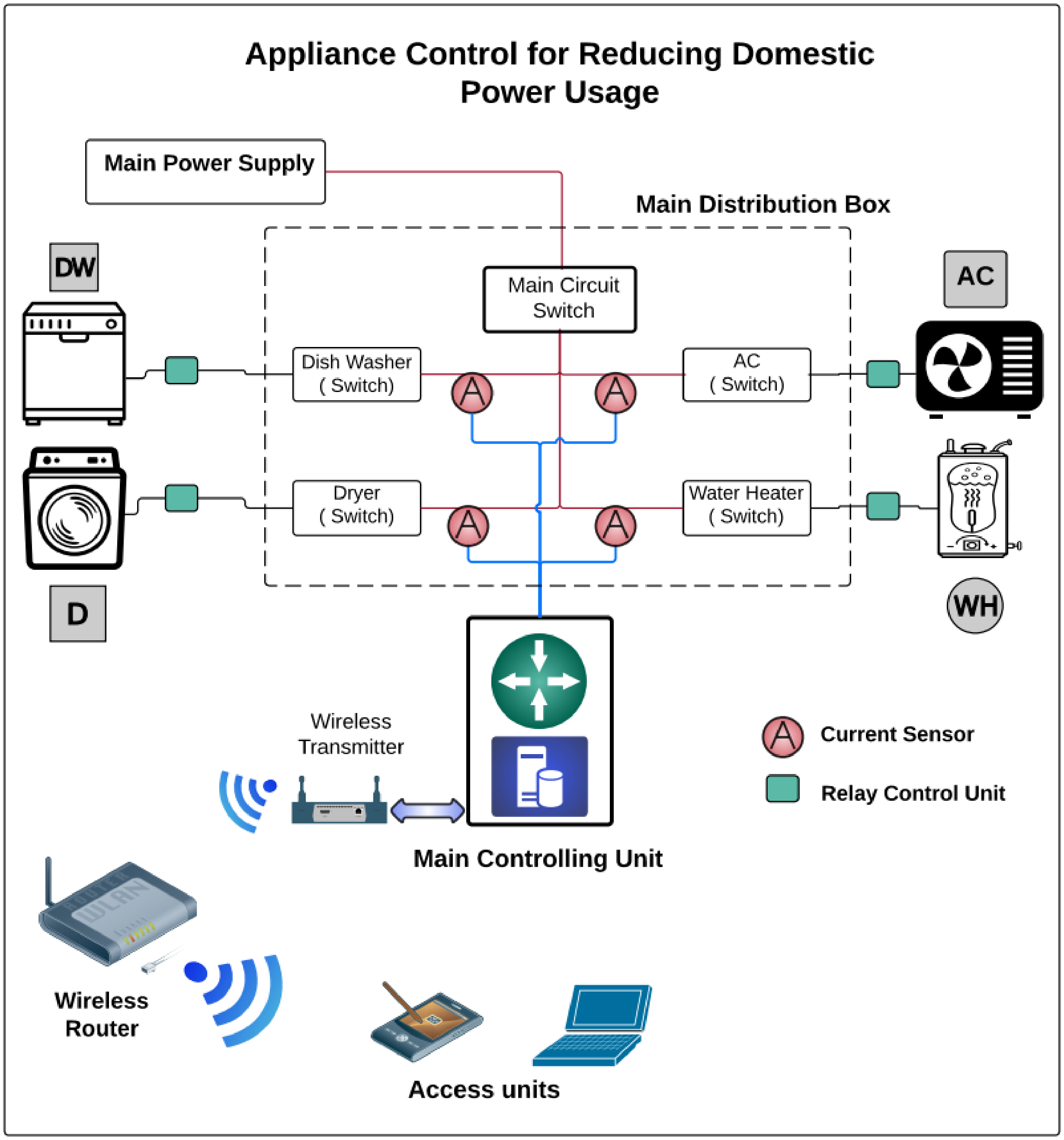

Figure 3 shows a diagram of our simulation and control systems.