Electric Power Savings in a Smart House

Patrick Cauley
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
Pec1622@jagmail.southalabama.edu

Kaan Oezpenpe
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
ko1825@jagmail.southalabama.edu

John Bishop
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
jcb1102@jagmail.southalabama.edu

Donald Manning
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
Dtm1921@jagmail.southalabama.edu

Kiara Risher
Electrical and Computer Engineering
University of South Alabama
Mobile, USA
kar1922@jagmail.southalabama.edu

***Abstract*— Home automation is quickly becoming a major direction of the future. It increases the quality of life of the user, and it provides the added benefit of reducing energy usage. Reducing energy usage is one of a consumer's best options for avoiding the rising cost of energy. The two areas most associated with energy use inside a home are lighting and temperature control. However, lighting and temperature are two areas most often associated with user comfort. The project sought to create a smart home small-scale living environment capable of sustaining the user-specified comfort levels while striving to decrease the associated energy use of lighting and temperature control. The model home environment will be implemented using a MATLAB simulation augmented by a physical hardware platform in what's known as a Hardware-in-the-Loop (HIL) configuration. A LabVIEW module will be running as a control system for the lighting, temperature, and comfort parameters. While the proposed platform will be implemented in a simulated, scale-living environment, it is the group's recommendation that further iterations of the project concentrate on a life-size, physical home.**

Keywords—> Smart Home, Controller, Small Prototype Model Living Environment (SMLE), Hardware-in-the-Loop (HIL), Simulink, LabVIEW

I. Introduction

Up until recently, the world has been relying solely on non-renewable energy sources like fossil fuels and natural gas to meet its energy demands. However, these are finite resources, and the demand continues to go up while the supply continues to dwindle down. This unbalanced supply and demand cycle signifies an increasingly dire global situation. It also has the added consequence of higher energy prices. A major way to combat this issue is by reducing the consumption of energy in general.

On the individual consumer level, smart home technologies can be one such way of reducing energy consumption. Reducing energy consumption of the home is one of a consumer's best options for avoiding the rising cost of energy while contributing toward future goals of sustainability. The two areas most associated with energy use inside a home are lighting and temperature control. However, lighting and temperature are two areas most often associated with user comfort. The goal of this project was to deploy a smart home small-scale living environment capable of sustaining a balance between the user-specified comfort levels of lighting and temperature and the power used to reach those user-specified comfort levels.

II. Smart Home Small-Scale Living Environment

A. Smart Home Devices

The rise of powerful but inexpensive hardware and software platforms coupled with the ubiquitous use of smart phones has led to a sharp rise in smart device deployment in homes. A smart home device is any technology that uses sensors and controllers to detect changes or automate tasks within a household. These devices range from smart lighting and smart thermostats to smart security systems and smart appliances. While the applications are becoming more diverse and widespread, smart home devices all share the similar goals of increasing convenience and quality of life of the homeowner [1].

Most smart home device configurations have at least three main modules: sensory data collection, data processing, and decision implementation [2]. For many applications, all three modules can be implemented using one platform or technology. For example, a very popular choice is the Arduino. A very basic Arduino-centered configuration involves having multiple sensors placed in the home connected back to the Arduino running the control algorithms, known as the controller. Then, these sensors report measurements in real time back to that controller. The controller processes the data and determines if the measurements warrant any correction via actuator output or notification of the user. If that is found to be the case, then controller directs the appropriate output. This bi-directional process of home data inputs and appropriate response outputs describes the basic functionality present in smart home devices [3].

B. Hardware-In-The-Loop

Hardware-in-the-loop is a technique used to develop and test control systems and complex embedded platforms. In a typical setup, the controller or system is connected to a simulation or model. The signals generated from the simulation or model are then used as inputs for the system. Those inputs are used to generate outputs, and those outputs are sent back out to the simulation or model.

A major benefit of HIL is the bypassing of real-world constraints while still evaluating close approximations of real-world performance. This allows for faster system evaluation and earlier system re-design. Due to the ability to hasten the simulation cycles and test over a variety of conditions, potential issues can be observed and addressed much faster and more precisely. It is easier to be proactive and design tests for edge cases and assumption testing, thereby allowing for more opportunity to adjust the system earlier in the design process.

C. Design Approach

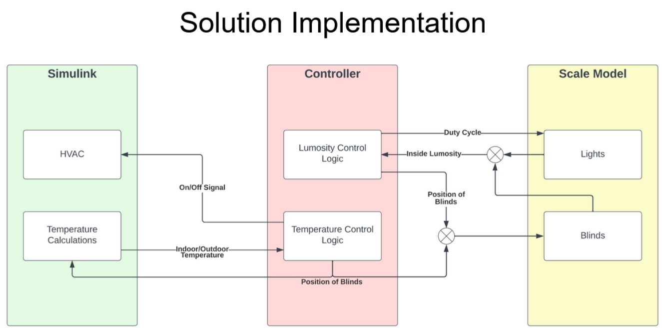

Fig. 1. Block diagram describing the basic system implementation

Overall, the Smart Home Small Scale Living Environment is a smart home technology implemented using an HIL platform. As shown in Figure 1, the project is broken into three modules: simulation, controller, and scale-model. The simulation was developed in Simulink. It models the thermodynamic processes taking place in a house over a 24-hour period. It provides the controller with an indoor temperature and an outdoor temperature. The simulation interfaces with the controller via an Arduino Due. The controller is implemented in Labview. The user-specified light and temperature values are entered into a user interface that feeds those values to the controller. The controller receives the temperature data from the simulation and the luminosity value from inside the scale model. It then uses complex logic and decision making to balance the user-specified light and temperature preferences with the current state of the home as evaluated from the simulation and scale-model data. The controller outputs the appropriate signals to the simulation and scale-model to achieve and maintain that balance. The scale-model is a physical platform constructed from foam board. It houses LEDs that serve as the model's lighting system, stepper motors that control the blinds, and a luminosity sensor. The LabVIEW controller interfaces with the model via an Arduino Mega.

The project's development was split into four parts: research and literature review, planning and design, construction and implementation, and data collection and testing. Each of the above four steps occurred for each component of the project: simulation, controller, and scale-model. Each step had its own unique challenges and difficulties, and the development of this project had a non-linear path. There were numerous periods of plateaus and negative gains. The work was divided among the members based on the individual's experience, strengths, and abilities.

D. Additional and Safety Considerations

Further iterations of this project may include interfacing the controller with a mobile app or running the HIL configuration using a TCP/IP protocol. This would mean the project would have a networking component, and therefore that would require consideration of possible cybersecurity protocols to maintain homeowner data security.

Safety is held to the highest of standards. This is a project about automation and with anything automated, there is a possibility of the machine acting unpredictably. Some safety concerns for this project are as follows; flashing lights could cause epileptic episodes, tangle hazard with manual blind control and wiring, and faulty wiring could cause a fire. To prevent an epileptic attack, the duty cycle of the lights should go straight from 45 Hz to 0 Hz. To prevent any tangle hazards, all wires were run off the ground and secured to the nearest wall or ceiling. To prevent any fires due to faulty wiring, all wiring diagrams were double-checked, and all flammable materials were and should always be kept from the project's components.

III. Methods and Procedures

A. Simulink Simulation

The simulation was implemented via a Simulink model in MATLAB. It simulated the thermodynamic processes taking place inside a home over a 24-hour period in order to generate an indoor temperature which was then sent to the LabVIEW controller. The base for the model was a pre-developed example simulation provided by MATLAB which featured a basic thermodynamic model of a house, a heat pump, and a thermostat which operated based on a relay.

For the purposes of the project, the basic-example simulation had to be heavily modified to address the following points of concern: an A/C system capable of cooling the house, a way to incorporate blinds and their effect on heat exchange, a solar heat gain provided by sunlight, a way to scale flow and exchange rates to accurately reflect real time, a way to reliably exchange inputs and outputs between the simulation and controller, and a way to run the model in discrete time without affecting accuracy of results. A brief overview of the project's completed simulation and primary concerns is described below.

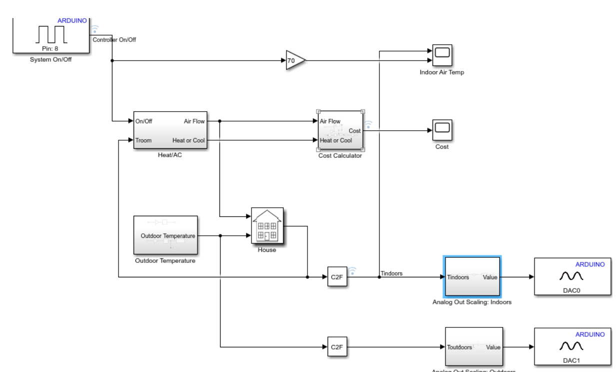

Fig. 2. The simulated house model implemented in Simulink

Overall, the simulated indoor temperature was governed by the below equation:

$$Temperature_{indoor} = Heat_{gain} - Heat_{loss} \tag{1}$$

For each time step, the heat gains and losses were calculated via three heat exchange processes: solar radiance, conduction, and convection. Solar radiance was calculated by using proportions of GHI values that corresponded to certain times of the day as shown below:

$$SolarHeatGain = [(1-x)(GHI)](WindowArea)\left(\frac{1}{HeatCapacity * AirMass}\right) \tag{2}$$

Conduction was calculated via heat loss or heat gained through the walls and windows by finding the indoor/outdoor temperature difference and multiplying it with a conductance value. Conduction calculations were altered by the current position of the blinds. Blind position altered the surface area of the windows, which affected the conduction through them. The implemented relationship between conduction and surface area is demonstrated in the below equations:

$$ThermalResistance = \left(\frac{1}{SurfaceArea}\right) \tag{3}$$

$$Conductance = \left(\frac{1}{ThermalResistance}\right) \tag{4}$$

The convection was calculated by determining if the system was heating or cooling and using a corresponding air temperature and flow rate.

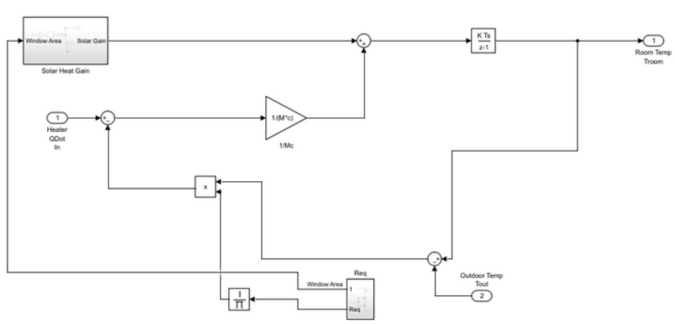

Fig. 3. The system and subsystems showing heat loss and heat gain calculations

To reliably exchange inputs and outputs between the controller and simulation, two methods were used: analog out and digital inputs in. For the temperature data, the values were scaled from 0 – 4095 and output via the DAC pins on the Arduino Due. For the blind position and heating/cooling commands, digital input combinations corresponded to values in a lookup table which then fed into critical points in the simulation.

Finally, the simulation was run via the Simulink Arduino toolbox. It allowed for the simulation executable to run real-time on an Arduino board. The simulation was run on a time basis where 1 second of real time corresponded to 1 minute of simulated time. This allowed for a 1 day run to be completed in 24 minutes.

B. *Controller*

The core of our project revolves around a LabVIEW based controller and a Simulink simulation to obtain the temperature values. This section goes into the details of the construction and configuration of the controller.

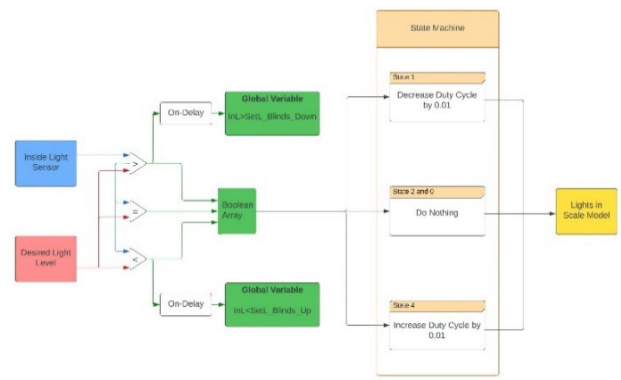

Fig 3. Shows a block diagram for the light control logic

Figure 3 shows the control logic used to determine the duty cycle of the indoor LEDs in the small-scale model. The controller takes the indoor light level and compares it to the set light level on a scale of 0 to 100. The results of the three comparisons (greater than, less than, or equal to) are then fed into a Boolean array. This Boolean array is then converted into a decimal number and then fed into a state machine as an input. This way there can be a designed reaction to any combination of inputs into the controller. As shown above a state of 1 means that the state machine needs to decrease the duty cycle, therefore making it darker. This follows the logic since for the machine to get a state of 1 the Boolean array must be [0 0 1]. Which means that the inside light level was greater than the set light level. State 2 and 0 are both states that do not change the duty cycle. The reason there are two states is that the default state is state 0; so that if a state input is a state that doesn't exist, then it will go to the default state.

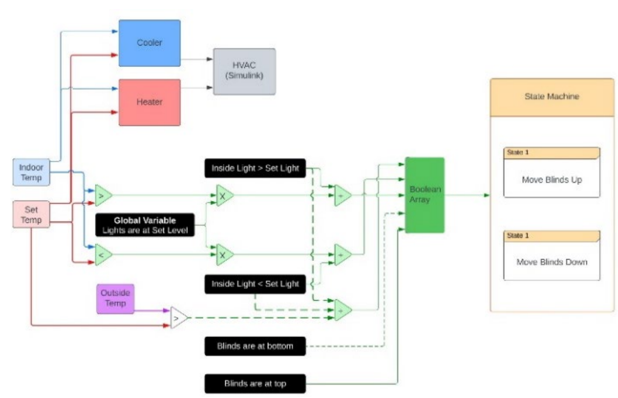

Fig 4. Shows a block diagram for the temperature control logic

Figure 4 shows a block diagram of the logic for the temperature controller. It starts with a set temperature that the user inputs. It then compares the indoor and outdoor to the set point. This creates three Boolean outputs. The Boolean outputs of the indoor and outdoor temperature comparisons are cascaded into an AND and an OR gate. They are multiplied with the global variable that holds the state of whether or not the inside light level is currently at its set point. The result of these logic gates are then added with different global variables. The global variables contain the data of whether or not the inside light has been unable to reach the set light level for at least five seconds. This is used to determine if the duty cycle is unable to get the inside light level to the set light level and the blinds must be moved. The results of all previous Boolean operations and two more global variables containing if the blinds are at the top or bottom of the window are then all fed into a Boolean array. This array is converted into a decimal number and sent to a state machine which has three states. The states are for the blinds to go up, down, or do nothing.

C. Hardware

The hardware part of this system consists of a small-scale model house, six-volt power supply, two driver boards, two motors inside the windows for the blinds, one photoresistor in the middle of the back wall pointed at the windows to measure luminosity, twenty LEDs on the ceilings connected in parallel with a digital switch (MOSFET) to dim the lights, and an Arduino. The Arduinos are connected to the other components to control the motors, photoresistor, and LEDs, allowing the user to adjust the blinds, measure luminosity, and adjust the brightness of the LED lights. The driver boards provide the power needed for the motors, allowing the blinds to be raised and lowered. All these components are integrated together to create an automated system that can be adjusted with LabVIEW. The circuit diagram for the entire small-scale house is shown in figure 4.

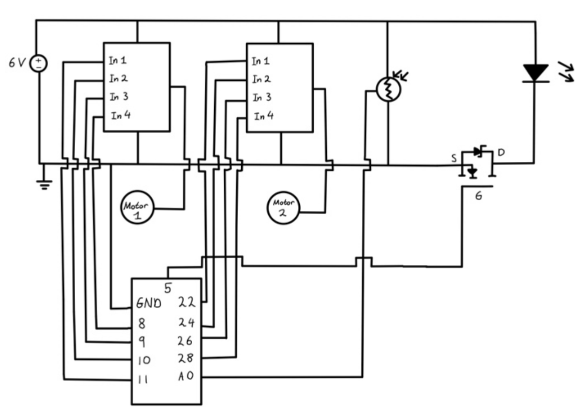

Fig. 5: Circuit schematic of all hardware parts

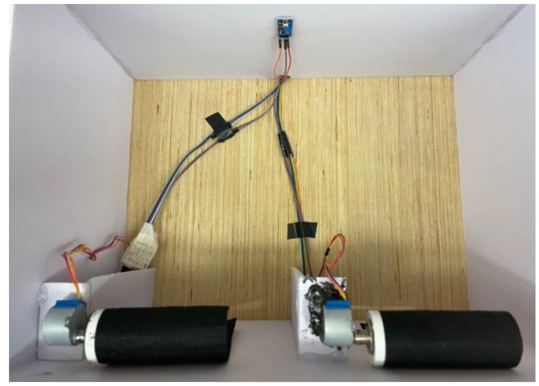

Fig.6: Hardware components inside small-scale living environment

IV. Results

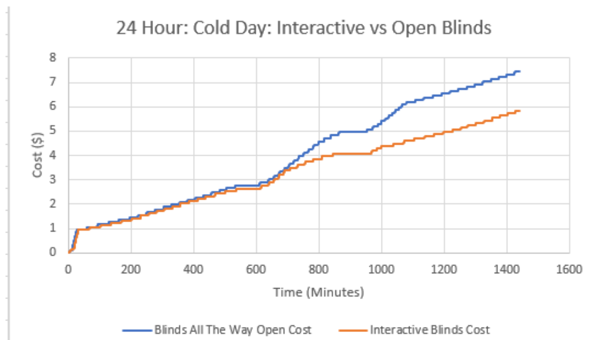

Fig. 7: Two 24-hour runs comparing interactive blinds to open blinds

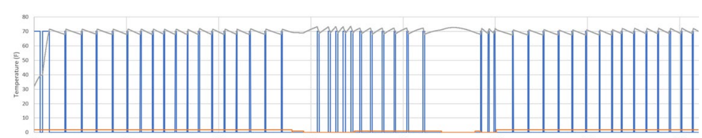

Fig. 8: 24-Hour cold day run showing interactive blinds leveraging outdoor heat gain to maintain indoor set point and light value

Table I – Simulation results

Temp Data	Window	Run Time	Cost	Set Point	Indoor Temp	Avg Outdoor Temp
Hot Day						
1-Aug	Interactive	1 Day	$ 5.50	70	69.79	82
1-Aug	Closed	1 Day	$ 5.34	70	69.55	82
1-Aug	Half Down	1 Day	$ 10.99	70	69.46	82
1-Aug	Open	1 Day	$ 16.26	70	70.07	82
Cold Day						
1-Dec	Interactive	1 Day	$ 5.84	70	69.4	51.6
1-Dec	Closed	1 Day	$ 4.95	70	69.77	51.6
1-Dec	Half Down	1 Day	$ 5.30	70	70.05	51.6
1-Dec	Open	1 Day	$ 7.43	70	70.58	51.6
Mixed Day						
Mixed	Interactive	1 Day	$ 4.10	70	70.3	78.26
Mixed	Closed	1 Day	$ 3.93	70	70.17	78.26
Mixed	Half Down	1 Day	$ 10.06	70	70.52	78.26
Mixed	Open	1 Day	$ 17.00	70	70.96	78.26

V. Discussion

Based on the results, the controller was able to turn on the HVAC system less often by opening/closing the blinds while maintaining the user's specified luminosity level, allowing the user to save money. The simulation was able to send the values for the indoor and outdoor temperature to the controller via an Arduino Due. The control system was able to take in the values from the simulation. If the control system was able to change the indoor temperature value using blind system without affecting the user's set luminosity value, then the blinds would move accordingly. If it was unable to do this, then the blinds would stay in their current position and the HVAC system would turn on. The control system was also able to communicate with the SMLE via an Arduino Mega. The SMLE was able to adjust the lights and blinds according to the input from the control system.

The project accomplished its goal of creating a smart home system that allows the user to maintain comfort while being able to save money by reducing the amount of energy used in the lighting and HVAC system in a home. This project has laid a foundation for future development. Areas that could be added to this project include cyber security, modeling appliances, multiple rooms, or creating an app to control the smart home.

VI. Conclusion

The project accomplished its goal of creating a smart home system that allows the user to maintain comfort while being able to save money by reducing the amount of energy used in the lighting and HVAC system in a home. This project has laid a foundation for future development. Areas that could be added to this project include cyber security, modeling appliances, multiple rooms, or creating an app to control the smart home.

REFERENCES

1] S. K. Vishwakarma, P. Upadhyaya, B. Kumari and A. K. Mishra, "Smart Energy Efficient Home Automation System Using IoT," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-4, doi: 10.1109/IoT-SIU.2019.8777607.

[2] C. -H. Chen, Y. -C. Wu, C. -M. Hong and J. -X. Zhang, "Applications of Environment Monitoring System Based on IoT via Wireless Sensor Network," 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2021, pp. 1-2, doi: 10.1109/ISPACS51563.2021.9651026.

[3] T. Gelu-Ovidiu. "Smart Home Automation with Arduino." Annals of the University of Petrosani Electrical Engineering, vol. 19, Jan. 2017, pp. 5–14. EBSCOhost,https://searchebscohostcom.libproxy.usouthal.10
1109/ICEARS5357
9.2022.9752265.