

Control and Automation in Solar-Powered Homes

Joshua de la Rosa, Department of Electrical Engineering
Arizona State University, Tempe, AZ

Prof. Bruce Land, Department of Electrical and Computer Engineering
Cornell University, Ithaca, NY

Introduction

The 2009 Cornell University Solar Decathlon (CUSD) consists of a number of students that are divided into different teams that work together to design and build a functioning house that is completely solar-powered. The Engineering Team is responsible for designing and creating a control system that address issues such as power management, electrical installation, and home automation and control. The 450-square-foot house is still in the design phase and has not undergone any architectural completion. Thus, the control/automation system will not have a physical house to manifest itself in for at least eight months. However, because the deadline of the house's total completion puts restraints on time, the task of the Engineering Team is two-fold: to design and build a bench-top model of a control system that will function properly and also to develop a standard, easy-to-follow protocol that allows for the flexibility to easily install electrical components to the control system of the solar house.

Background

The solar-powered house consists of several subsystems. This summer, our team focused primarily on the lighting, which controls the light fixtures as well as switched outlets for lamps. This was the first year that CUSD's Engineering Team started to design the control systems for the house even before the house has been built. We received various donated equipment for the controls from two companies: ELK Products and Smart Homes Insteon™. These companies specialize in home control and automation and although they are separate companies, their technology is compatible with each other, making integration less of a problem. Using these equipment, the team hoped to incorporate the lighting system so they can be remotely and automatically controlled.

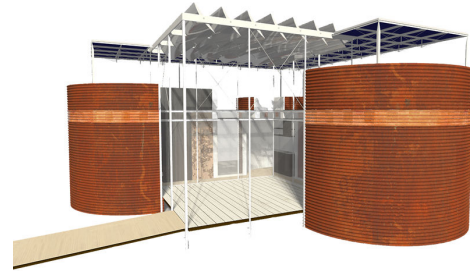


Figure 1: Design Rendition of 2009 CUSD House

Hardware Methodology - "How to Put it Together"

In order to establish a bench-top model lighting system, the design was broken into two primary tasks. The first task was implementing the proper hardware connections from the main control board to other sub-circuit components and configuring the software to program them. This included wiring and cable connections made to AC transformers, Ethernet switches and computers, and the lighting network interfaces (see Figure 2 below).

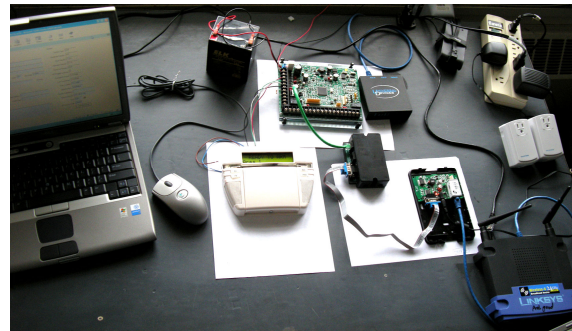


Figure 2a: Photo of Control "Backbone"

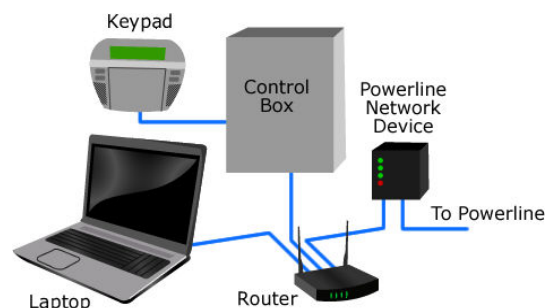


Figure 2b: Diagram of Control "Backbone"

As seen in Figure 2a, the main control board processes commands, receives information from thermostats and sensors, and stores programs. It is connected via RJ45 cable to a main serial port interface (MSI) to allow serial communication and to upload/download programs between the control board and any computer device. The MSI is then connected to via RS232 cable to the MIXEP in order to establish an Ethernet network using TCP-IP protocol. These three components (the control board, the MSI, and the MIXEP) make up the control box seen in Figure 2b. This control box was designed to communicate with various user interfaces. It is controlled by either a keypad or remote software. Because the keypad is directly linked to the main board, it can communicate directly with the main board. The software on the computer communicates with the board via switch or router and the Ethernet component of the control box. Once the control board receives instructions from the keypad or software, it sends information to the powerline network device.

The second task was to actually construct a bench-top model of the different lighting configurations that will be implemented. The purpose of this model was to test the software program to remotely and automatically control the lights and also to showcase it as a demonstration to those who will actually be integrating this system into the real house. For simplicity's sake, we decided to use a wooden board to place electrical switches, outlets, and bulbs to represent different configurations. We then hardwired them to a power supply that easily plugs into the wall to give the required 110VAC. There were many ways to control the lights in a system, but because this is a small house, only three simple configurations were used:



Figure 3a: Photo of Lighting Bench-top Model

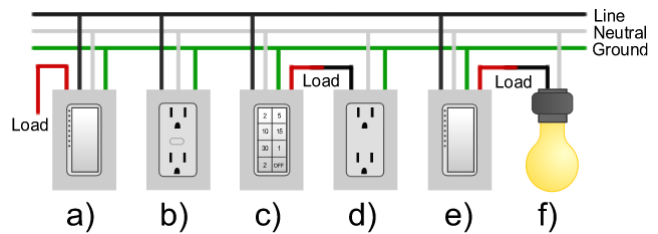


Figure 3b: Diagram of Lighting Configuration

1. The first is hardwiring an Insteon device directly to a light source (see Figure 3b – Switch E controls Light F). This is *exactly* like hardwiring a regular switch to, say, a ceiling light. The only difference is, you can control that switch remotely if you want.
2. The second one is an Insteon switch controlling a regular outlet (see Figure 3b – Switch C controls Outlet D). This is neat because you don't need to replace every outlet with an Insteon outlet. You can *indirectly* control the outlet by directly controlling the switch (i.e. Turn the switch on from your computer and the outlet gets turned on).
3. The last one is an Insteon switch controlling an Insteon outlet (see Figure 3b – Switch A controls Outlet B). With this setup you can remotely control either the switch or the outlet *directly* (So if you wanted to turn on a light from your computer without having to turn it on from the switch, you could do that).

After we were able to configure the devices to function, we had to link them to the control board to control them remotely from the keypad, computer, or programmed to be automatic. The software is designed in such a way as to allow for easy installation of multiple devices. Once in place, a device's address (which is unique to each device) only needed to be entered into the software's database in order for it to function.

An equally important aspect of the project was documentation. Included with each of the parts donated by ELK and Smart Homes was an instruction/installation manual. Thus our goal was to compress these multiple manuals, which altogether comprised of hundreds of pages; into one simplified document for step-by-step installation of the control board as well as setup of the hardware and software. Because our contribution was focused primarily on a few

switches and outlets, we had to ensure that the protocol we created was capable of being reproduced on a large-scale for the actual household installation. The format for the installation manual is a step-by-step procedure that includes in each step the instruction, a diagram or photo of the step, and the reference(s) in the actual manufacturer’s manual (see Figure 4 below).

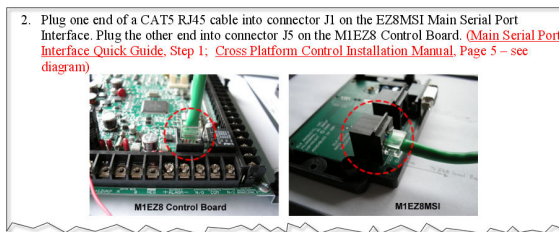


Figure 4: Excerpt from Instruction Manual

The entire documentation process required constant revision due to malfunctioning equipment and parts. However we were finally able to complete a 10-page installation manual.

Engineering Design - “How It Works”

Because the lighting system was connected directly to the main board, communication and programming was all centralized into the user interfaces (i.e. the keypad and computer). In fact, the board was designed to connect to up to eight different control zones including temperature sensing, motor functions, etc. The software on the computer is a user-friendly, GUI interface that allows a resident to either remotely control the lights and thermostats or create rules to allow the control board to automate tasks. The screen shot below in Figure 5 shows an example program that automatically turns off the garage lights at 8:00 A.M. Once the program is downloaded onto the board, it sends data to the Insteon devices.

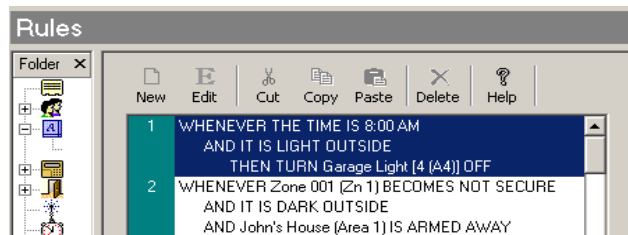


Figure 5: Screen Shot of Software Program

At the engineering level, the Insteon devices communicate on the powerline by adding

a carrier frequency of 131.65 kHz to the powerline voltage, which in the United States is nominally 110 VAC RMS. The Insteon data is modulated onto the carrier frequency using a binary phase-shift keying (BPSK), ideal for performance in the presence of noise.

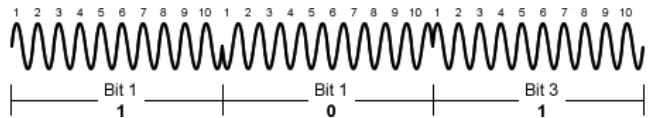


Figure 6: Insteon Powerline Carrier Signal with BPSK modulation

As shown in Figure 6 above, Insteon uses ten cycles of carrier for each bit of data. All Insteon devices are peers, meaning that they can act as either controllers (send messages), responders (receive messages), or repeaters (relays messages). This allows the network between devices to be flexible: one controller can send messages to multiple responders and similarly, many controllers can send messages to one responder.

Model Test Results – “Did It Work?”

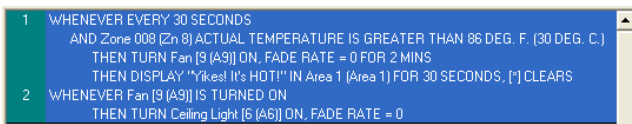
After making all the proper hardware connections, programming and installation was straightforward. However, we were set back because the software was unable to communicate with the main board. Initially, we believed the problem was faulty wiring, but after debugging, we discovered that the software was able to *connect* to the board; it just wasn’t able to *communicate* with it. We concluded that the main board itself was malfunctioning, and had to order a new one from ELK Products. Once replaced, the software successfully connected and communicated with the board, allowing us to download programs as well as the database of Insteon devices onto the control board.

However, in those few weeks that we were unable to get the control to function properly, we shifted the team’s energy into the second task, which was building the bench-top model. Upon completion of the model and the arrival of the new parts, we were able to link the lighting system to the control system.

However, upon enrolling each device into the software database, there seemed to be no communication between the remote software, the

keypad, and the Insteon switches and outlets. Information was able to be sent from the control system, but not received by the devices. The problem was later found to be a malfunctioning Powerline modem, which is the crucial bridge component that allows for communication between the control and the devices. After quickly replacing this part, we were finally able to get a working control system that we could test.

The first test we made was a manual test: ensuring that the physical switches were functioning properly. Our tests were favorable, as the switches were able to successfully activate their corresponding outlets/bulb. The second test we made was a remote access test: being able to turn off/on the switches remotely using the keypad software. The results of this test were also favorable as they functioned as they would in the first test. The third test we made was an automation test: downloading a simple program into the control system to toggle the light switches every two minutes. Upon downloading and activating the program on the keypad, we were able to see each light toggle off and on after timing each two minutes. After successfully passing these three simple tests, we decided to run another automated test using a series of outlets, switches and a temperature sensor that we connected to the control board to take temperature readings. Figure 7 below shows a snapshot of the downloaded program:



```
1 WHENEVER EVERY 30 SECONDS
  AND Zone 008 (Zn 8) ACTUAL TEMPERATURE IS GREATER THAN 86 DEG. F. (30 DEG. C.)
  THEN TURN Fan [9 (A9)] ON, FADE RATE = 0 FOR 2 MINS
  THEN DISPLAY "Yikes! It's HOT!" IN Area 1 (Area 1) FOR 30 SECONDS. [1] CLEARS
2 WHENEVER Fan [9 (A9)] IS TURNED ON
  THEN TURN Ceiling Light [6 (A6)] ON, FADE RATE = 0
```

Figure 7: "Thermostat" Program

After downloading this program, and enabling the tasks on the keypad, we placed the temperature sensor (which was connected to Zone 008) next to the bulb controlled by a switch (here labeled *ceiling light*) to increase the temperature. Once the temperature reached 86°F, the keypad beeped and displayed, "Yikes! It's HOT!" on the LCD screen. Then the fan plugged into a controlled outlet was turned on and started to cool off the temperature sensor. Once that was completed, the ceiling light turned off. This program illustrated a crude form of a thermostat.

Although not necessarily an effective, practical application, it demonstrates the power and flexibility that the control system has.

There is one important issue that needed to be addressed with the control system: security. Although a physical breach of the control system is serious, electronic breach is just as jeopardizing and therefore equally important. One suggestion we made for the house was to not install a wireless router that connects to the Ethernet interface of the control board. This is because an outside computer may be able to connect into the network and hack into the control system, giving the intruder complete access to the lighting system. Another possibility is whether or not the main board can be breached from the powerline (e.g. an outlet outside the house). This issue may warrant further investigation in the future. What we know for sure is that the control system is secured upon losing power. In the event of a power failure, when power is restored, the settings on the control system are retained so the resident does not need to reconfigure it.

Conclusion

After designing and creating a bench-top model of a lighting system, we were able to control and automate it using remote software and keypad capabilities. From it, we derived a more complete understanding of how this cutting-edge technology which is intended for residential/commercial use can be implemented into our solar-powered house. As a result, we were able to accurately document these procedures for the actual, future installation.

References

INSTEON: The Details; SmartHome Technology; 2005; Pg 33.

Acknowledgements

This research was supported by the Cornell University Louis Stokes Alliance for Minority Participation. Major equipment donations were provided by ELK Products and Smart Homes Insteon. Funding and participation was also provided by the Cornell University Solar Decathlon Faculty Advisors. Thank you.