Michael Lodder
David Chu
James Bradley

# Cellular Home Automation

# Table of Contents

To: Professor Al Davis

From: Michael Lodder, David Chu, James Bradley

Date: April 24, 2006
Subject: Proposal for Senior Project on Plug and Play Home Automation

---

## 1. Introduction and Motivation:

Home Automation is becoming increasingly popular. By installing a network of microcontrollers into home appliances a person is given an increased amount of control over their home. Home automation allows devices such as lights, doors, ovens, thermostats, bread makers, microwaves, and garage doors to be conveniently monitored and controlled remotely through devices such as cell phones and the internet. However, there are some drawbacks present in the current process of installing home automation systems. The task of implementing an automated system is often very expensive with prices usually falling in the range of thousands of dollars. Most people spend over 25,000 dollars to have a home automation system wired into their house. In addition, if a person wishes to later add or automate anything else in their home, they must call a construction contractor who then comes, installs, and revamps the controls to include the new part. This system is very inconvenient and can detract from the benefits of home automation. Our project is to make a version of home automation that is less expensive, can be controlled via a cell phone or the internet, and is easy to install like a plug-and-play device, so a person could do it themselves. We are motivated to undertake this particular project because we feel that it is diverse and encompasses many of the areas of computer engineering while also being affordable, practical, useful, and desirable.

## 2. Overall Design:

The general idea of our project is to modify household appliances such as door locks, lights, ovens, grills, and thermostats so that they can be remotely accessed and controlled by a cell phone and/or a computer in addition to hands on control by a user. A server controlling all of the home appliances can be accessed via cell phone or webpage. The cell phone will interface with the server by pressing buttons on the phone to pre-recorded messages. The server will then authenticate the person by accessing a database of valid users and their passwords. This way, not just anyone will have control over the household appliances. Once authenticated by the server, the user will be able to access information about home appliances such as which lights are on, what doors are unlocked, what the current temperature of the house is, and so forth. If the user then wishes to change the status of one of these appliances he or she can give a command through the phone or webpage which then sends the given command to the appropriate home device. The interfaces that the user will interact with to communicate with their home automation system will depend on the medium of communication.

Our baseline goal for this project is to be able to access and control door locks and light switches remotely using the webpage and a cell phone. The baseline cell-phone interface will be as follows: A person calls the designated phone number, and once the computer has authenticated them, the user can check or change the state of the lock or the light. The web page interface will accomplish the same results. The baseline web interface will have a graphical interface but nothing fancy like a Macromedia Flash website.

If we have extra time on our project we will add more home appliances with which the user can interact. For instance, we would also like to be able to control the thermostat, turn on and off the oven, and even have a remote tracking system so you can see where people are in the house. Furthermore, we could also upgrade both the internet and cell phone interface. For the internet interface we will add graphical layouts of the house showing what the status of each of the appliances is, and also allow the user to interact with these appliances by clicking on fun and intuitive graphics. We could also implement timers for the devices within the computer to automatically turn them on or off, or manipulate them in some other manner with a signal through the network at a certain pre-defined time. Also, our project could be able to make the system easily expandable. In other words, the computer can detect a new appliance automatically by sensing a new signal coming in on the network, and add it to the control menu, and allow the user to assign a name to the device.

## 3. Implementation:

- *Control Program-*

An application programmed in C# will be made to manage the cell-phone interface and the web user interface. It will be able to spawn threads and enable or disable access to the devices remotely. It will look almost identical to the web-page interface and have some buttons and links for the cell-phone interface so someone can check and see if the computer is talking to someone. It will also give access to each device to a user on the server and will also be password protected like the cell phone interface and the webpage.

- *Cell-Phone Interface-*

To allow a person to dial into the server a program called TAPIex will be used. TAPIex is a program that allows the computer to answer the telephone line that it is connected to through a dial-up modem. After the computer answers, it will begin to go through a sequence of menus created by a programmer, playing pre-recorded messages. The menus will be in the following order: 1- will prompt user to enter ID# and password; 2-For door control press 1, for light control press 2; 3-enter the door/light number to control; 4-it will state the current status, and press 1 to change the state or press 2 to go back, afterwards all of these return to the main menu. Light control will be identical but will say to turn on/off instead of lock/unlock. When the person touches a button on their phone to navigate through the menu, TAPIex will read which button was pushed and allow a programmer to design a program to manipulate the data accordingly.

ASP.NET and C# are the languages that will be used to make the control interface. All programming will be done in Microsoft Visual Studio 2005.

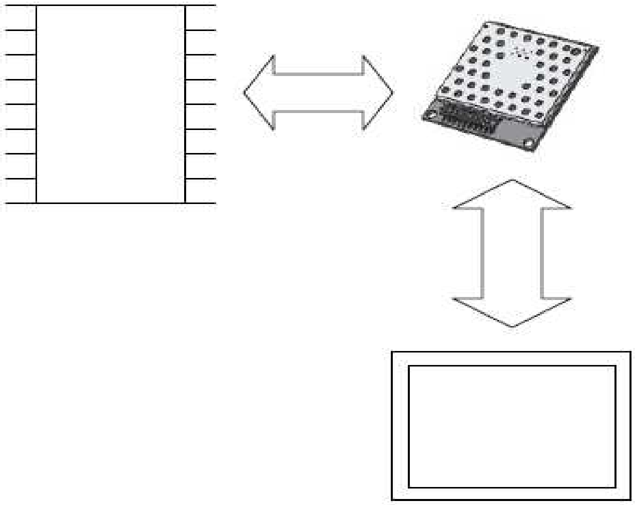
- *Web User Interface-*

To allow a person to control applications from the web, a webpage with links to the various appliances and houses will be designed with C# and ASP.NET. When a user initially tries to access the program, a page will come up and ask for their ID# and password. The server will then access a Microsoft Access database for their information and retrieve it. After they are authenticated in the system, all of the applications in their house will be available to be controlled through the use of clicking on buttons. The current status will be displayed next to an icon and label of the appliance on the page as well.

- *Communication from Computer to Appliance-*

The computer will connect through a serial port to a wireless transceiver. The wireless transceivers are Aerocomm AC4490-200M. They transmit on a frequency of 900MHz and will enable it to reach up to four miles in direct line of sight. This is the same frequencies that modern day portable telephones use which is able to broadcast throughout the entire house with no problems, unless a person's home is bigger than four miles long. This means our transceivers will be able to achieve the same quality. The transceivers communicate using the TTL format. Since the computer will be sending data using the RS-232 format, a converter will be needed. These are available at any computer store for $20. At the receiving end, another Aerocomm transceiver will communicate with a microcontroller which uses the TTL format as well, so no converter will be needed. They transmit up to 115Kilobits per second, which is more than fast enough for our design since we won't be transmitting more than a few bytes.

A backup plan is to implement power line networking. For power line networking from the computer to the appliance, Intellon, a company based in Florida, offers network cards and chips that are designed to send data over home power lines called the Power Packet. These can be used with old and new house wirings making it easy to setup, they don't use any additional power so power bills stay the same as before. Each of these chips runs on a network card or through USB. It would be the same as a computer connected to another computer through ethernet cards. They enable transmission speeds up to 15Megabits per second, which is more than enough for our application. The downside to using this method is we will have to buy a data converter from the ethernet protocol to TTL which costs hundreds of dollars and defeats the whole purpose of implementing a cheap home automation design. We will use this method though if wireless proves to be a problem.



- *Signal Processing -*

The microcontroller will read in the data sent from the computer through the transceivers and will determine what to do with the appliance that it is connected to. The microcontrollers are Motorola 68HCS08AW16s. They can be programmed in C or assembly using tools from www.freescale.com for free. The only thing they will be doing is telling the computer the current state of the appliance, on/off or locked/unlocked, and changing the state of the device and sending the information back to the computer.

- *Controlling devices -*

a) **Doors:**

To unlock/lock a door, the microcontroller will send a signal to a solenoid that is around the door lock. The solenoid door lock is made by www.keylesspro.com which comes with a radio transmitter to tell the door to lock/unlock. The solenoid creates a magnetic field depending on voltage levels and when it is off, does not make the door harder to open which would be the case if a stepper motor were to be used. If a stepper motor were implemented, a way to disconnect it from the door lock for manual overrides with a key would need to be made so a person is not locked out when the power is down. Using a solenoid does not create this problem which means a person is not locked out of their house when the power is out. The solenoid will pull or push the lock depending on which command was given. A radio transmitter connected to the microcontroller will send the signal to the solenoid telling it what to do. This way no wires will be needed to be run to the door. A stepper motor will be used as a backup in case we can't get the solenoid working. If a person uses their key to unlock/lock a door, the microcontroller will sense this and update it on the computer. A sensor to the microcontroller used to determine if the door is locked or not will consist of the deadbolt closing a circuit when it's locked. When the deadbolt is in the locked position, a pin to the microcontroller will be receiving electrical current, so the microcontroller will know which state the door is in.

b) **Lights:**

To turn on/off a light bulb, the microcontroller will be connected to a touch sensitive light switch. It will send a signal to a switch capacitor circuit which is equivalent to having a person's finger touch the switch. In this way, if a person wants to turn the light on/off without the computer, it’s easy to do. If they touch the switch and the light changes states, the microcontroller will tell the computer this change and update it. It will be able to sense this by checking to see if it is receiving power on a certain pin or not.

- *Power Supplies-*

The wireless transceivers and microcontrollers both need five volts to power them. We will need to design a bandpass noise filter circuit and a half wave signal rectifier to produce the five volts. The noise filter is necessary because microcontrollers and transceivers require as clean of a signal from the power supply as possible. The rectifier will be needed to lower the voltage from a wall socket, which produces 120 VAC, to produce a 5 VDC. To make these, standard electronic components such as a diode or op-amp will be used, and can be purchased from the University of Utah engineering department for less than $1. The door solenoid controller will be run by 2 AA batteries.

## 4. Risks:

We feel that the risks associated with our project are minimal. Our baseline system is not overly complex and we feel that if we put enough time into the project, we will be able to get it done by the end of Fall Semester. Some safety concerns we have are that we will be working with 120 volts in the wall sockets, so we will need to take extra precautions not to hurt ourselves or our circuitry while hooking up our systems. We will also need to make sure that our power supply circuit doesn't fry the microcontroller/transceiver by supplying too much power which can happen before or after a power outage, which sometimes generate spikes in the power line.

We will also need some kind of manual override system for cases where the power goes out, or the server fails. For example, we would not want a system that locked someone out of his or her house just because he or she could not unlock the door remotely. We will not have this same problem with the light switches, because we will use capacitive switches which do not appear as being off or on. Also, when the power is out, the light cannot be expected to be on.

Another important item to consider is how we are going to check the status in the event of a manual override. We would need some kind of a system in place so that the microcontrollers can detect a manual override in the event that someone has locked the door by hand, flipped the light switch, set the oven, or changed the thermostat. For devices already governed by a microcontroller, such as the oven, or the thermostat, once we figure out the way the internal computer workings, manual override status checks would be simple. However, for simpler devices that have no embedded computers, it will be a much more difficult task to detect these overrides. Each system that we connect to will need a different sensor so the microcontroller knows which position or state the appliance is in. More than likely we will need to design a custom sensor circuit ourselves for each appliance.

Another risk exists if we cannot get a current touch light switch circuit interfacing with our microcontroller. If this happens then we will need to design our own circuit. Many web sites are available that describe how this can be done, but our circuit may not be as efficient as one already made.

## 5. Decisions and Learning:

Things that we will need to learn include figuring out how to interact with each of the interfaces of the appliances that we want to incorporate into our system. Interfaces such as light switches and door locks would be relatively simple, while interfaces such as the oven or the thermostat will take quite a bit more time to figure out.

Initially we were going to use power-line networking because it was simpler, but then learned that we would need a lot of voltage format converters such as ethernet to RS-232 then RS-232 to TTL for each microcontroller which costs a lot of money and would have made our circuit quite bulky. For these reasons we chose to go with the wireless solution, which reduced costs and size.

Next, we will have to learn how to read network packets so as to interpret the correct signal for the microcontroller, and to send the data back to the computer. We will need to learn to network program since none of us has had much experience with this. To program our webpage, we will need to learn how to program C# on a webpage.

## 6. Qualifications:

All three of us have been computer engineering majors at the University of Utah for at least three years and have taken classes that consist of advanced programming and hardware circuit design. Mike and David have taken advanced circuit design classes while James has taken more intensive software programming classes. Additionally, we all work for companies where we are exposed to a lot of programming and circuit design.

## 7. Bill of Materials:

| Product | Where | Price | Quantity | Contact |
|---|---|---|---|---|
| RD5200-ETH-U Wall Adapter | **www.intellon.com** | $12 | 6, if needed | Pam Garris,<br>**pam@intellon.com** |
| TAPIex license | **www.plimus.com** | $65 | 1 | **techsupport@tapiex.com**<br>**sales@tapiex.com** |
| MC68HCS08AW16 | **www.freescale.com** | Free | 5 | 800-521-6274 |
| Metroworks CodeWarrior | **www.freescale.com** | Free | 1 | 800-521-6274 |
| Touch Light Switch | **Lighting Design** | $15 | 2 | |
| Stepper Motor | **www.grainger.com** | $25 | 2, if needed | |
| Door lock w/radio transmitter | **www.keylesspro.com** | $119 | 1 | |
| Electronic components | **University of Utah Engineering stock room** | 0.10¢ | Buy as needed | |
| AeroComm AC4490-200M | **www.mouser.com** | $62.50 | 5 | **support@mouser.com** |
| RS-232 to TTL converter | **www.digikey.com** | $20 | 1 | **support@digikey.com** |
| Microsoft Visual Studio 2005 | **U of U Computer Science Department** | Free | 1 | |
| Ethernet to RS-232 converter | **www.netburnerstore.com** | $99 | 5, if needed | |

## 8. Work Schedule:

We intend to start working on this project over the summer due to the fact that we will have more free time and no school. Hopefully, over the summer we can accomplish most of our baseline goal so that when we get back into school we can focus on creating the extra bells and whistles for our project. In order to be able to start our project over the summer we will need to complete our ordering of the parts and have them arrive as soon as possible. If our project takes longer than we anticipate, we will work hard to finish the baseline one month before the end of the semester to give time for proper testing. Here is the timeline for our project:

May 1st: Finish planning the details of project, get parts ordered.
June 1st: Complete microcontroller software and power supply circuit
July 1st: Interfacing with door and microcontroller completed
August 1st: Communication with computer and microcontroller completed
September 1st: Web page completed with user database
October 1st: Interfacing with light switch complete, telephone program implemented
November 1st: Testing phase and finishing baseline project
December 8th: Finishing touches and demonstration, Added extra features.

## 9. Division of Labor:

The general work layout that we have derived so far is that James Bradley will be mostly responsible for software design of the webpage and user interface program with Mike Lodder. Mike Lodder will mainly work on programming the embedded microcontrollers and with David Chu interfacing them with the different appliances while David Chu will be responsible for designing and testing the analog circuits. James and Mike will also be responsible for handling the communication between the computer and the microcontroller. James and Dave will also be responsible for the wireless communication, signal processing, and implementing the telephone program. We will all work together to test each other's contributions to make sure that they are as robust and bug-free as we can make them. We will also all document our own projects as we build them. If anyone is having trouble in a particular, all will help with that part until it is completed.

## 10. Conclusion:

Upon completion of our project, one should be able to control and check the status of a door lock and a light switch via their cell phone and/or computer. Also, if time permits our system will also be able to interact with other appliances and be easy to expand, with software to support plug-and-play appliances. By endeavoring on this project we hope to improve upon the industry standard of home automation by making it more affordable, practical, and maintainable.