# Smart Home Systems

Daniel Moody, Vu Ha, Joel LeGros, Thien Luu

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper discusses the design and implementation of a system that monitors and controls power outlets while also supporting sensors to gather information of the current state of a house. The primary goal of the system is to give the user convenience through control and information gathering. In order to do this, the system integrates a broad range of technologies from computer and electrical engineering such as High-level programming, internet communication, RF transceivers, and microcontroller systems. The design includes utilization of an android phone application for user interface, a small inexpensive web server for internet communication, and power outlet modules.**

***Index Terms* — Application Software, Internet, Master-Slave, Open Source Software, Wireless Sensor Networks, ZigBee**

## I. INTRODUCTION

The motivation for developing smart home systems comes from many reasons, but most prominent are convenience, security, energy management, connectivity and luxury. Smart Home systems are one of the newer areas of research that have not been fully integrated into our society. This is because the research requires many other disciplines of research and engineering to produce a functional smart home. The cost of installing a smart home is also a large hindrance to the emergence of smart home systems into the market. The extra cost of the install is from the fact that even though a majority of homes were built in the near past, technology has been growing exponentially. This means that most homes were built before this technology was available, and this creates a barrier for the development and sales of smart home systems. However the technology is becoming better and cheaper, and this will help to make smart home systems an expense worth having when new homes are being built.

The biggest motivation behind smart home systems is the convenience. Convenience is really another way of saying "time saver", and into day's world where everything is moving faster, every second has value. Most of the technology we use today is based of convenience, for example cars get us where we need to go faster, phones get us information from other people faster, and computer's get work done faster. Smaller conveniences in the home will be desirable because they allow the home to save the user time as well. There are already many convenient technologies in the home like the dishwasher, washing machine, and microwave ovens. These technologies are more mechanical in nature and often there are much less computerized conveniences in the home. A Smart home systems goal is to introduce the benefits of computerized technology. For example, when using the smart home system, the user will not need to walk around turning off lights, they can save that little bit of extra time by just pressing a button on their phone, or even have the lights programmed to shut off after a certain amount of time. Maybe there is some music on the user's computer they would like to play on a sound system, smart home systems will allow the user to play

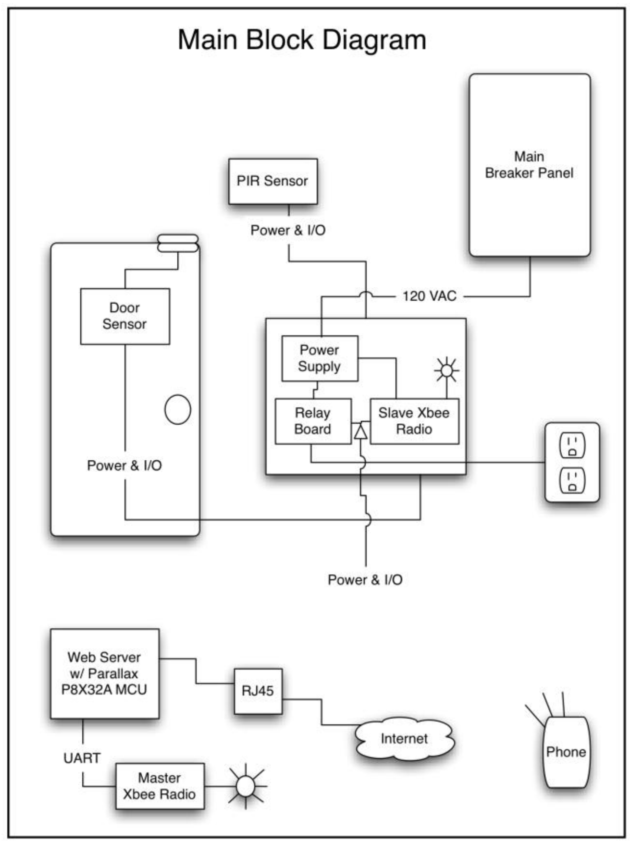


Fig. 1. This block diagram shows the general layout and intended use of the project. Antennas from Xbee Radios show wireless communication. Some cables will carry multiple wires.

the music from where ever they are without needing to go to their computer, find the song, and make sure the song is in a playable format for their sound system. There are many other small conveniences provided from the smart home.

Security is also a big factor in the emergence of smart home systems. With a sophisticated enough system, home security becomes a powerful tool that gives piece of mind and power to the user. Security systems are also a large deterrent for crime. The mere presence of a camera will put doubt in any criminals mind about committing a crime. There are already many security systems on the market available and in use today, however integrating the security system into the smart home gives the user a one stop access to everything in their home. If the smart home system integrates the smart phone into the system then this means that the user will always know the status of the security of their home.

## II. OVERALL SYSTEM LAYOUT AND COMPONENTS

This project has several key components that are utilized. See Fig for an overview of the system. These components are a Smartphone, Spinneret Web server, Xbee MCU/wireless communication, and the Smart Home Systems unique PCB. Each of these components has a general role and intended use for the project. By adhering to this intended use, the user can realize the motivations and goals of the project. The overall system can viewed as a block diagram in Fig. 1.

### *A. Smartphone Control*

The Smartphone is the user's way of interfacing with the smart home system. The user will access an app that is written specifically for an android phone, which will allow the user to login to our server and have functionality of a few things. The user will be able to have access to relayed power over outlets or anything the relay is in line with like lighting. Other areas of control will be in the form of lighting, accessing power management info, and sensor information. An example of the Smartphone used in this project is seen in Fig 2.

One of many uses of the Smartphone will be to turn on and off lighting. Lighting is one of the main features that the group wants to include in the wireless control modules of the smart home project. It is important not only because it is convenient, but because it also will play an important part in reducing power consumption throughout the home by reducing light levels when not necessary. It can also come into play as in home security system by illuminating the home when the residents are away on vacation or when approaching the home at night. Thefts may think the owners at home if the when to turn on/off. The on/off controller can also extend to other devices and can be used with a the relay control to shut an appliance off when not being used.

HTC Thunderbolt client Specifications

- Platform: 1GHz Scorpion CPU, Android OS 2.2
- Memory: 768 MB RAM, 8 GB Internal Storage, microSD expansion
- Display: capacitive touchscreen, 480 x 800 pixels, 4.3 inches
- Network: GSM 850 / 900 / 1800 / 1900, HSDPA 900 / 2100, HSDPA 850 / 1900 - North America
- Connectivity: GPRS Class 10 (4+1/3+2 slots), 32- 48 kbps, EDGE Class 10, 236.8 kbps, Speed HSDPA, 7.2 Mbps; HSUPA, 2 Mbps


Fig. 2. The Android Smartphone, HTC Thunderbolt, used in testing and development of the Smart Home Systems project.

These features will all be controlled with a relay and therefore allow the user to place anywhere in the home. It can be placed near an outlet, or right before a lighting fixture. Because of its small design, it will be able to fit into tight spaces either behind walls or above the ceiling. This important feature will be able to be controlled via the android app from the Smartphone.

### *B. Spinneret Web Server*

The Web server that we've used is called the Spinneret by Parallax Inc. The spinneret is made up of several key components.

It uses parallax P8X32A propeller chip. This chip is custom designed by parallax and has multi-processing. The chip itself has a small footprint of 12.3 x 51 x 3.41 mm. It has a built in interpreter that uses a high level language called "Spin" as well as low-level assembly language. I has an architecture of 32-bits as well global RAM of 64 K bytes. The chip has I2C serial communication and requires the range of 2.7 to 3.3 VDC to operate.

By using an adapter called the "Prop Plug," (USB to Serial adapter) we're able to program the chip. The adapter has a four pin header that allows for transfer, receive, reset (active low), and ground (VSS). It's capable of a 3 M baud for the asynchronous communication rate.

The server also uses the WIZnet W5100 single chip. This allows for an internet communication at through 10/100 Ethernet control. This is all done without an OS to keeps things simple and cost effective. The W5100 has an integrated TCP/IP stack as well as Ethernet MAC and PHY. It takes in a voltage of 2 to 5.5 VDC.

### *C. Xbee Radios*

The Xbee radios that we are using provide wireless communication between the web server and the modules controlling the outlets and receiving information from the sensors. The smart home system is using the Xbee series 1 format radios. They use the 802.15.4 stack commonly known as Zigbee. The radios operate 3.3 VDC at 50mA. The 1 mW output can reach to about 300 ft (100m) range. This allows for simple communication for our system. The Xbee radios allow for either point to point or multipoint topologies.

XBee modules provide two friendly modes of communication – a simple serial method of transmit/receive or a framed mode providing advanced features. XBees are ready to use out of the package, or they can be configured through the X-CTU utility or from your microcontroller. These modules can communicate point to point, from one point to a PC, or in a mesh network.

Features:

- Parallax support
- Wire antenna
- Cross-compatibility with other 802.15.4 XBee modules
- Low-power sleep modes
- 100 ft (30 m) indoor/urban range and 300 ft (100 m) outdoor line-of-sight range
- Configured with API or AT commands, local or over the air
- 8 digital I/O and (7) 10-bit ADC inputs
- 802.15.4 network topology
- Multiple antenna options

Specifications:

- Up to 115.2 kbps interface data rate
- 2.4 GHz frequency band (accepted world-wide)
- Industrial temperature rating (-40C to 85C)
- Transmit power 1 mW (+0 dBm)
- Supply voltage 2.8-3.4 VDC; transmit current 45 mA; receive current 50 mA
- Power-down current <10 uA
- 3.3V CMOS UART interface level

### *D. PCB Modules*

The PCB was designed using free software called Express PCB. The company also manufactured the boards as well. Other companies like Eagle Cadsoft were considered, but ultimately Express PCB was chosen based on its ease of use and intuitive nature for our group. Price was also a consideration, and Express PCB allowed us to make a quantity of three two-layer boards for $85. Production of the board took only 10 days to make as well. The company also allowed for options of one and two day turnaround of production as well. This was appealing for any last minute revisions we might have had with the design. The two-layer PCB boards helped with our helped with our wire routing for the design as well as well as a general saving on space. The $85 deal from Express PCB also had a stipulation on board size. The PCB could not be more than a dimension of 3.8 x 2.5 inches.

## III. ANDROID APPLICATION

The user will interface with the system through an Android smart phone. Smart phones are becoming widely used and setting up the UI on a smart phone will increase its accessibility while allowing it to be portable. The Android platform was chosen because it is more developer friendly with a free application market and Java programming environment.

### *A. User Interface Layout*

The goal of the UI is to be lightweight, quick and easy. When first running the application it will start into a login screen. The user will need to enter a username and password which will be stored on the web server. When the user presses the login button, the application will then ask a DDNS service for the web server's current IP address. Then after obtaining the IP address, the application will request the User and Module database text file from the web server. If the database does not respond the application will inform the user and not allow login. If communication is successful then the text file will be parsed and all the current users will be entered into a SQL database. The application will then look for the entered username and password.

If the user database is empty, such as in the case that the system is being used for the first time, the application will prompt the user as soon as they enter the login screen and ask for the master user to be created. The system will not function without a master user and the master user will always be the first username and password entered into the system. The master user will have the ability to add, edit and delete other users as well as have access to all features of the system.

After successful login, the home screen will be entered and show a list of all modules currently active and display the latest status. The status can be updated with an update button and will also be updated anytime the user decides to turn an outlet on or off. The user can change the state of any module from the list by simply clicking on it. This will bring the user to a module detail screen showing the current settings of that module. Also available from the home screen will be the ability to add modules, log out, search logs. If the user is the master user they will be able to add, edit and delete users through the user administration button.

### *B. Database Updating*

The system will need to constantly be receiving updates in order to stay current with the status of house. This will be achieved by requesting updates from the server every time the user interacts with the application. The application will also request updates after a certain period of inactivity. This can be set by the master user depending on how often the user wants data to be downloaded from the web server. The User and Module database will need to be rather small in order for updates to function in a timely manner and will need a set limit on the maximum number of users and modules able to be created within a given system.

The Log function will update only the most recent events that have occurred in the system. This will be done by organizing each day as an individual text file on the web server. The application will only update the log file by adding the days in a chronological order. For example the application cannot just grab one day from a month ago, it will need to grab all the days after that day as well. The user can select to update the log files for any date in the past and all logs from within the time period will be downloaded and entered into the database. Because this can take a long time depending on the size and history of the log, the user will be warned and asked if they would like to proceed anyways. Once the log files have been downloaded they will not need to be updated as the log files will not be changeable.

### *C. Server Communication*

There will be two main modes of communication between the web server and the phone application, commands and database updates. All communication with the web server will be through http GET request. This is because the web server is run on a microcontroller and is limited in the kinds of internet protocols it can receive and understand. Fortunately everything needed by the system can be sent through simple get requests.

The first thing the application does on startup is get the IP address of the web server. This is done by sending a get request to the domain name for the web server. The DDNS service will update a DNS to a dynamic IP address. The application gets a response from the DDNS service and then parses the IP address from the response. The most recent IP address is saved and used for construction URLs.

The get requests are sent by constructing the URL from within the application dynamically depending on the user input. Apache libraries are used in sending and receiving data in HTTP. A command request to turn an outlet on or off is sent through the URL and is parsed at the web server. When requesting a database update, the file location is used in the URL. All requests must be answered with a response from the web server acknowledging that the request was received and handled correctly. If not then the user is notified that the server was unable to communicate.

## IV. SPINNERET WEB SERVER

The spinneret web server is a compact efficient design for running a web server created by Parallax, a company that sells various electronic parts and kits tailored to hobbyist and amateurs. The Spinneret Web Server includes the parallax P8X32A propeller chip, WizNet W5100 Ethernet controller chip, 32 kilobyte EEPROM and 32 kilobyte RAM, a real-time clock from a crystal oscillator and an SD card slot. It also includes a voltage regulator that can take in 5 to 12 volts and outputs 3.3 volts. The Spinneret Web Server is a completely open source project with all hardware and software available from the parallax website.

*A. Spin Objects*

The web server can be programmed in several languages, with open source compilers available on the parallax website; however parallax created the high level spin language especially for the propeller microcontroller. SPIN language is similar to python and is object orientated. The language itself has a low level feel to it, using assembly size declarations and accepting assembly language as correct syntax.

Fostering the open source work on the propeller and SPIN language, parallax has an object exchange section on their website, allowing users to upload SPIN object files. Parallax also held a contest when they released the Spinneret Web Server, which was one of their first kits to include their propeller microcontroller. This spawned a large amount of open source software for the propeller. This system utilizes many of the open source objects available from the parallax website.

The multi socket HTTP web server was created by Mike G. it works closely with the W5100 Ethernet controller's drivers to continually monitor and update sockets, handle HTTP request, build response headers, and use FTP protocols to serve up larger files. The code is simple and well documented allowing for manipulation to done to handle the type of request required by the Smart Home System. The other SPIN objects used by the project include the SD card drivers, Xbee UART communication drivers, a string manipulation object, and real-time clock drivers.

*B. Database Architecture*

The web server is run by the small propeller microcontroller, so one of the biggest challenges was creating the database that would store all the information about the user, current system status and logs. The main challenge in its implementation was that the limited amount of RAM on web server, so the code had to be designed in an efficient specific format allowing it to only be able to specifically complete the task required by this project.

There are three entities in the database, which include Users, Modules, and System Events. Each of these can be represented as a table and each will be stored in a text file on the SD card. The propeller uses string manipulation and parsing to search and edit the database. This method is rather slow but will only be imparted on the User and Module entities which will have a set limit to keep processing time low. The System Event log will not be edited or searched on the web server, only insertions will be made to the end of file. The entities will be organized using space characters as column separators and carriage return characters to indicate new rows.

The User file will contain a Master user as the first user in the database, and then nine other users for a total of 10 users allowed in the database. A User will contain a unique username, password, name and security level attribute. The Module file will contain up to 25 modules with each module containing a unique panID, outlet status, motion status, and security sensor status, location, and security level attributes.

For the System Event log file, the attributes will be date, time, type, description, user, module, and security level. The description attribute will be a short sentence describing the event. For this attribute periods will be used to indicate spaces within the text files on the microcontroller. The System Event log will also be created by separating the log information into separate days, containing each days with of events into one text file. This is for increased efficiency when updating the database to the phone applications and also so that the user can decide how much data they would like to download from the database.

*C. DDNS and IP updating*

For most residential connections, the ISP will assign a dynamic IP address. This posed a problem because the phone application will communicate with the web server by using an IP address in the construction of the URL. The solution was to use a Dynamic DNS service which will automatically update a DNS to resolve to the changing IP address. There are several companies that provide this service, and for this project we decided to use No-IP.com. They provide 3 DDNS for free.

In order to update the DDNS to always be pointing to a dynamic IP address such as at a residential gateway, the web server must send a get request to the No-IP server. When the No-IP server gets the request, they will get the IP address that is sending the request and update your DDNS. The web server will need to only update the DDNS in cases where the residential IP address has changed. This can only occur when the modem at the residential address needs to reconnect to the ISP. This can happen for in a loss of connection or loss of power. For this reason the web server will only update the IP address on start up.

Problems with this are that if connection is lost without a power outage, then the web server will need to be restarted manually if the gateway IP address has changed. Another problem with using the DDNS service is that it takes a short period of time for the service to update your DDNS, on the order of 5 to 10 minutes, during which the user will be unable to use the system.

*D. Xbee Control*

The system uses Xbee Series 1 transceivers, which are small RF microcontrollers, to exchange data between the modules and the web server. The Xbee radios will be set up in a network with a master radio at the web server and slave radios at the power outlet modules. The slave radios will not transmit data unless they receive a request from the master to do so. This network mode is a simple communication protocol and ensures that the master radio will always only be receiving a single transmission at a time. If two or more slave radios were to try to transmit to the master radio at once, the master radio would not be able to understand the transmission.

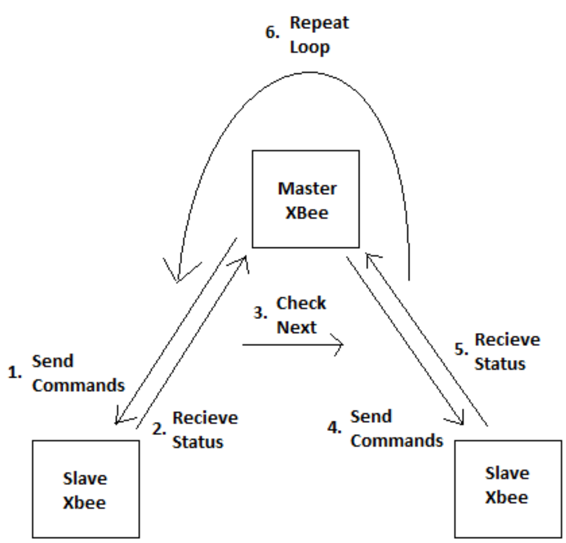


Fig. 3. Xbee network setup for sending and receiving data. Master Xbee represents the Xbee at the web server and Slave Xbee represents Xbee at the power outlet modules.

In order to send out commands and get information on current states of the power outlet modules, the master radio will need to loop through all the slave radios, sending out queued commands and checking the status of each one as it moves through all the slave radios in the network. The master radio will be able to send commands to a slave radio in the order of twenty to fifty milliseconds. In the extreme case that there were twenty-five power outlet modules in the network and they all required fifty milliseconds to communicate with the master radio, then the system would take a little over a second to communicate with all of the radios. This is about the maximum time for detection from the motion sensors and security sensors. A flow diagram in Fig 3 demonstrates how this process will work.

The Xbee transceivers will only be sending digital I/O signals between each other. An I/O signal will be used for each of the power outlet's sensors and relay switch control. There will be a total of 4 digital I/O pins used on the Xbee radios, one for each of the following: relay switch control, motion sensor output, manual switch output, and security sensor output.

When the master connects to a slave, it will send its digital I/O for the relay switch turning it to on state or off state, and the slave will send 3 digital I/Os for the for motion sensor, security sensor, and manual switch status.

The master XBee will use UART communication protocols to talk to the web server and tell the current state of the I/O pins and which slave it is currently talking to. The web server will record this data into its Module database file. Then the web server will respond by programming the Xbee to check the next slave in the network depending on how many slaves are in the database. This process will be repeated and all power outlets in the network will be checked.

## V. PCB Design and Layout

The power outlet control modules where designed to draw power from the house power by using a AC to DC power conversion system. The PCB was uniquely designed specifically for the power out modules to support several digital I/Os, such as relay control switches, sensors and anything else capable of using 3.3V digital I/O signals. The schematic used for the PCB design is seen in Fig. 5. The completed product of the schematic is viewable in Fig. 4.

*A. Transformer & Rectifier Bridge*

The step down transformer takes in 120 VAC from a line of the breaker panel, and outputs 9 VDC at an operating current draw of 500mA. After passing through the transformer the power is sent through a diode rectifying bridge using 1N4007 diodes to rectify the alternating current to direct current. A 1000uF parallel capacitor is used to minimize the ripple voltage for the input to the regulators.

*B. Voltage Regulators*

The power gets regulated from after the bridging diodes by two specific regulators. These regulators are the LM317 and LM7805.

The LM317 is a positive adjustable regulator, considered one of the most popular variable voltage regulators in its class. The regulator has three terminals and is capable of supplying more than 1.5A over the voltage range of 1.25 V – 37 V. It basically requires two external resistors to set the output voltage. The regulator has been adjusted to output 3.2 V using a 100 ohm resistor for R1 & 150 ohm resistor for R2. See equation (1) below for this calculation.

$$Vout = 1.25\left(1 + \frac{R2}{R1}\right) \tag{1}$$

The output is short-circuiting protected and it has about 80 dB of ripple rejection. This regulator in particular is powering the slave Xbee. We've used a 47uF capacitor minimize noise from the output of the regulator.

The LM7805 is being used to output 5 V. A 47uF capacitor was also used to minimize noise from the regulator.

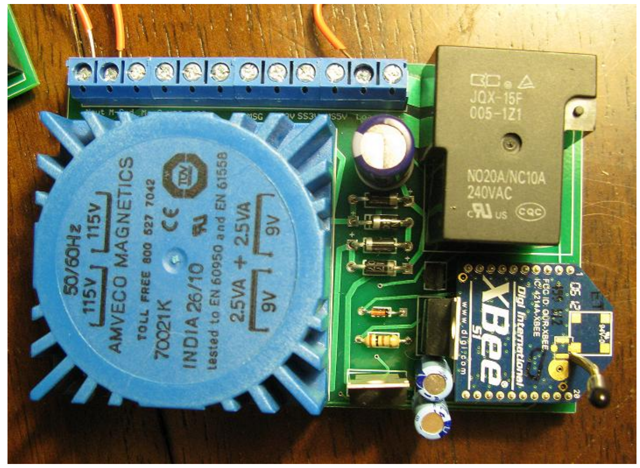


Fig. 4. Top side of final completed PCB design with Xbee module attached.

*C. Xbee Modules*

The Xbee series 1 is receiving 3.2 V from the LM317 regulator for power. Digital I/Os of 0-3 were used to deliver signals to the components of our project. Digital I/O 0 (pin 20) is used to control the signal for the relay switch. The output of the Xbee controls the transistor which directly controls the relay. The relay uses an

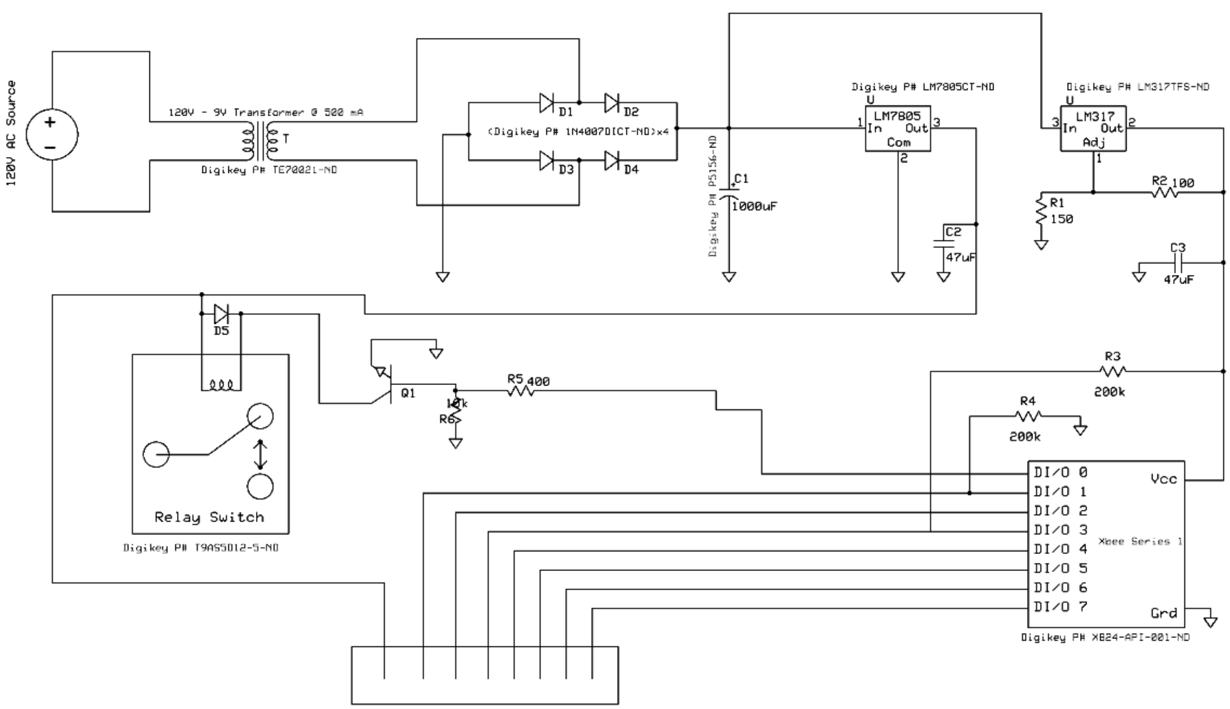


Fig. 5. A custom designed schematic of the PCB.

inductor to create a magnetic field which allows the switch to be closed upon seeing an induced current. Digital I/O 1 is the manual switch which allows overridden control of the outlet/load that is attached to the system. It uses a 200 kOhm pull down resistor. Digital I/O 2 is used for the motion sensor output. This takes voltage changes from the sensor and feeds it back to the master Xbee for data. Digital I/O 3 is used for the Door sensor output. This uses a 200 kOhm pull up resistor.


## ACKNOWLEDGEMENTS

We would like to thank Dr. Richie for taking the time to advice, helped us with our project, and motivated us to use higher technology for the best fit in designing our project.

We also would like to thank Mike Gebhard from Agave Robotics LLC, for help in setting up and programming the Spinneret Web Server.


## REFERENCES


[1] Android Open Source Project
Website: http://developer.android.com

[2] Digi International® Inc.
Website: http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#overview

[3] Ladyada, "PIR motion sensors", AdaWiki, Website: http://www.ladyada.net/learn/sensors/pir.html

[4] Parallax Inc.
Website: http://www.parallax.com/

[5] ExpressPCB Company
Website: http://www.expresspcb.com/


## ABOUT THE ENGINEERS

Joel LeGros is currently an undergraduate senior at University of Central Florida, graduating in electrical engineering in summer 2013. Joel is working for Summit Church as a lead audio engineer in Orlando. He plans to continue to work in audio engineering field upon graduation.

Thien Luu is a senior undergraduate student in Electrical Engineering at University of Central Florida. Thien is currently working for Florida Air National Guard as an emergency manager in logistics section. He plans to pursue an engineering career in Department of Defense and to acquire a graduate degree after graduation in fall 2012.

Daniel Moody is a senior undergraduate Computer Engineer student, graduating in fall 2012. He is currently working on software development for a UCF research grant funded by the National Science Foundation. He is most interested in developing software for mobile platforms. He is planning on working in the Computer Engineering field after graduation and eventually continue his education at a later date.

Vu Ha is an Electrical Engineer senior, graduating fall 2012. Vu is currently a private tutor in math and science. He is looking position in or related to Electrical Engineering and also planning to obtain a graduate degree.