“Project Lion” - A SMART HOME SENSOR NETWORK

Author: Watipatsa Nsunza, Salum Issa, and Faniry Sandamahery
Supervisor: Dr. Zeng Yujiang

May 25, 2015
Abstract

Project Lion is a smart home sensor network. The goal of the project is to implement an autonomous (time controlled) home network that can provide necessary home functions at a specific time. The home functions appear as variables of time, weather, or interference/intrusion on the home network. With these tools, the home can maintain a very comfortable and livable environment. “Project Lion” is a growing project that we imagine to be referred to as “A VERY SMART HOME” in the near future. The engineers in this project are working hard to achieve all the goals we have set for this project before graduating from HUST in 2016. The Project we are introducing now is known as “Project Lion” A Smart Home Network that is supposed to grow into “Project Lion” A VERY SMART HOME.

Key words: Project Lion, Smart home, security system, light control system temperature control system, sensor network
Abstract

List of Figures

0.1 MODEL DESIGN ..................................................... 1
0.2 AIMS ................................................................. 1

1 Light Control System ........................................... 3

1.1 Overview ............................................................. 3
1.2 Light sensor .......................................................... 3
1.3 How did I get the idea? Inspiration .............................. 4
1.3.1 Circuit drawing from Fritzing .................................. 5
1.4 Autonomous Lights (First Feature) .............................. 6
1.4.1 Principle and code ............................................... 6
1.5 Fading lights (Second Feature) ................................. 7
1.5.1 Principle and code ............................................... 8
1.6 Light &Temperature Code LLT v.1 ............................. 9

2 Temperature Control System ......................................... 14

2.1 Overview ............................................................. 14
2.2 Beginning Steps .................................................... 14
2.2.1 Required components ........................................ 14
2.3 Seasonal Variance .................................................. 14
2.3.1 Summer .......................................................... 14
2.3.2 Spring ........................................................... 15
2.3.3 Winter .......................................................... 15
2.4 conclusion ............................................................ 15
2.4.1 challenges ......................................................... 15
2.4.2 final peice ......................................................... 16

3 LION SECURITY SYSTEM .............................................. 18

3.1 intro ................................................................. 18
3.2 Security/Automated door system ................................. 18
3.3 Process ............................................................... 18
3.3.1 component list ................................................... 18
# List of Figures

<table>
<thead>
<tr>
<th>Figure 0-1</th>
<th>lion model design</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 0-2</td>
<td>First Draft of &quot;Project Lion&quot;</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1-2</td>
<td>LED Simulation</td>
<td>5</td>
</tr>
<tr>
<td>Figure 1-3</td>
<td>light controller Circuit drawing from Fritzing</td>
<td>5</td>
</tr>
<tr>
<td>Figure 1-4</td>
<td>Fading Light Feature</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2-1</td>
<td>Temperature Control System v.1</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2-3</td>
<td>Light-Temperature Control System</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3-1</td>
<td>Security System fetched on instructables</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3-2</td>
<td>Lion Security System Ver.2</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3-3</td>
<td>5v power supply unit</td>
<td>28</td>
</tr>
<tr>
<td>Figure 3-4</td>
<td>Team Lion</td>
<td>31</td>
</tr>
</tbody>
</table>

0.1 MODEL DESIGN

The Design of the project will be a model design that we’ll use to demonstrate the smart home functions known as “Project Lion”. We have a few other ideas that we would like to include in our project but due to time we might not be able to introduce them in the first version of this project.

![Figure 0-1 lion model design](image)

0.2 AIMS

Project Lion “A Smart Home Network” The following are the goals we have set to implement in the first version of the project:
Figure 0-2  First Draft of "Project Lion"
Chapter 1  Light Control System

1.1  Overview

Why I chose to implement a light sensor? First of all, I think light is needed every second by human beings; from literature or love stories, many people usually use that word “light” to express their ideas or their deep emotions. Scientifically or naturally speaking, we know that the sun provides light for any kind of living creature to survive; including us human beings. These two aspects are to me the principal motivation to go forward and make me think that light can do a lot more than we can imagine. Therefore, I was decided to take charge of the light feature of our project in this smart home sensor network as a telecom engineer.

1.2  Light sensor

Beforehand, in this project as a model, we are using Arduino, which is more affordable and easy to use, this can help me achieve the idea I’m thinking of. We can program an ATmega328 microchip on it using a language very similar to the “C Programming language” that we’ve learn in previous years. As I said before, I’m very interested in light that I tried to find something to improve the daily light capabilities we are using to make it more efficient to our needs especially as home owners. There are two main features of the light that I worked on, they are each related to the location of obstacle. 1. Turn on and off the lights automatically 2. Fade the lights depending on position In the first feature, I plan to help the owners of the house to not think about turning the home lights on or off by hand when they want to enter any room in the house or leave the building. By automatically turning on and off the lights, time can be saved, money can be saved. The main idea is making the lights in some places of the house fading on or off as he/she is moving from one place to another. We automatically turn on the lights when he or she is entering a room or approaching the house and turn it off once he or she is leaving the room. So how does it work? I’ll describe how it works now along with some pictures. The device below is the key thing that can perform both features of the lights. It is a distance sensor HC-SR04 from Arduino. It has 4 pins, GND, TRIG, ECHO and VCC. In principle, it calculates the distance between an obstacle and the sensor and sends that distance to the Arduino.

How the Ultrasonic Sensor operates? The picture on the right represents top view of the device which has the Source and Detector. The red rectangle is the obstacle. The sensor sends ultrasound from the source and when it meets an obstacle, it sends a signal back to the detector, the sensor now calculates the time taken by the ultrasound to reach the obstacle and return back to the detector which now converts that calculation into distance for more simplicity and other applications. The Atmega328p can take the time and convert it into distance. The appropriate
Chapter 1  Light Control System

(a) Ultrasonic Distance Sensor

(b) distance

formula is:

*Distance = TIME/2\times 2

*TIME: 2 way travelling time of the ultrasound We should notice in the description that the wave travels twice, but we only need the one way path, that's why we are dividing 2 in the formula.

1.3 How did I get the idea? Inspiration

After checking the Web, I found a helpful topic to guide me in achieving the goal. This original picture is extracted from http://www.instructables.com/id/How-to-make-A-light-up-distance-sensor/

In the original version found at the webpage, the system can perform only the first feature described above. There is 3 green LEDs, 2 yellow LEDs and 1 red LED that light up one by one as we approach the ultrasound distance sensor at the front.

For example, when the obstacle is at 100cm (now programmed as the max range), the first green LED is on. When it is at 75cm the second green LED is on, and further… until the obstacle is very close to the distance sensor, when the last red LED is on it indicates that the obstacle is very close. The pull-up resistors on each LED are used to protect the LED from high voltage directly from the power supply.
The operating range of the distance sensor HC SR-04 is between 2 cm to 400 cm max. It can be powered by the 5V digital pin available on Arduino Uno. In our project, we’ve used a small distance range between 2cm to 100 cm for convenience. Particularly, I used 3 red LEDs to indicate 3 different positions at the front of the house as I will demonstrate here with a picture. Before I start, it is very necessary I think to explain a part of the Arduino Code is dealing with distance.

Simply speaking, pingPin is the pin from the 2 middle pins (TRIG and ECHO) of the HC SR04 distance sensor to one digital Arduino Pin. It is set as INPUT by the pinMode statement. TRIG and ECHO are attached together to send and receive ultrasound. Then, that INPUT is sent in the “air” by setting its state to HIGH by the function pulseIn. When it touches any obstacle,
Chapter 1  Light Control System

Chapter 1  Light Control System

```c
pinMode(pingPin, INPUT);
long duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
float cm = microsecondsToCentimeters(duration);
```

it signals back distance (2 way travelling distance) picked-up is held in duration. Duration is a
time, we convert it into distance calling the function microsecondsToCentimeters which will
return the formula in the above picture distance = time/29/2; Cm is hold in currentValue later
used to know the current position of the obstacle. It keeps track of the obstacle displacement.

1.4  Autonomous Lights (First Feature)

As we can see on the right part of the above breadboard, we have 3 red LEDs which con-
nected to digital PINs of the Arduino and protected by resistors. These 3 LEDs represent the 3
positions according to the obstacle’s location. Then the distance sensor on the left is supposed
detect the position and then trigger the 3 LEDs on the right “ON” one by one. The idea behind
that is, it can show us for example that somebody is entering a room, first LED on, then if he/she
moves to kitchen; the second LED is on, and the third LED is on if the person moves to the living
room. As we can see, it is more convenient and efficient an answer to our natural needs to keep
us control darkness.

1.4.1  Principle and code

I think it’ll be more convenient to give the code first for more clarity about the principle.

```c
if (currentValue<100)
    digitalWrite(corridor3,HIGH);
if (currentValue>100)
    digitalWrite(corridor3,LOW);
if (currentValue<75)
    digitalWrite(corridor2,HIGH);
if (currentValue>75)
    digitalWrite(corridor2,LOW);
if (currentValue<50)
    digitalWrite(corridor1,HIGH);
if (currentValue>50)
    digitalWrite(corridor1,LOW);
```
- corridor\(x\) represents the 3 red LEDs \((x=1,2,3)\)

- \(\text{currentValue}(\text{cm})\) indicates the actual position of the obstacle. As we can see, each LED has on or off state depending on the \(\text{currentValue}\) i.e when \(\text{currentValue}>100\), then LED3 is off triggered by the signal sent by the Arduino to the corresponding digital pin of corridor3. When \(75<\text{currentValue}<100\), LED3 is on, otherwise LED2 is on and LED1 is on if the obstacle is less than 50 cm away from the sensor.

* The triggering of the LEDs is performed by the function \(\text{digitalWrite}(x,y)\) where \(y\) is either HIGH or LOW to send the signal to \(x\).

* The \(\text{currentValue}\) reference can be changed as we wish as long as it is less than the max range of the HC SR04 sensor (200cm).

* Those LEDs are connected by preference to NOT pwm digital pin of the Arduino even though it can be so for reasons that I’ll explain in the next feature.

### 1.5 Fading lights (Second Feature)

The second thing is that, we can now feel that the distance sensor works by calculating the distance, and sending the signal to the LEDs to turn them on at a certain distance. So now comes in my mind, another feature to improve the home light and something more convenient in some situations. What happens if somebody wants to have it lighter at one place and darker in another? Or what if we give more light to him or her as long as they are approaching a single area and make the other areas darker as they are leaving? Because previously, the LEDs are on and off, but now what if we’re only fading them according to position? That’s the new feature we’ll describe now as a second improvement I highlighted before and we will present it now in the following session!

I think these 2 pictures can help for description. I’ll try my best to illustrate as clear as possible. We have 2 top down pictures of the same circuit but bright light in two different positions. I took them to illustrate the change in the light brightness as long as the obstacle moves. Even though it’s not a video, as we can see the only difference in the two picture’s is the light brightness. I used my phone as an obstacle to have picture at the same time. When my phone is near the sensor, it corresponds to the top picture where the left red LED is brighter than the right one. In real time video, we can observe the brightness change more as long as I move but in this picture, it is at specific times. In the second time, I move away from the sensor, and as obvious, the brightness should change in an opposite way, i.e the left red LED will fade away and the right one will fade up. That’s why in the bottom picture, right LED is brighter than the left LED.
1.5.1 Principle and code

I’m going to show now how these fading lights work. Here I need a little bit of mathematic.

```plaintext
currentTime=cm;
brightness = ceil(range/currentValue)*5;
brightness2 = currentVALUE*5;
```

I saved the cm measurement first as I said in explanation earlier when describing “currentTime” for more clarity.

Let’s say y1= the brightness for LED1 and y2= the brightness2 for LED2

What I want is more brightness for the LED when approaching and less otherwise. So, I achieved these two ideas by:

*Using y1=1/x, the way it works as we know, the greater x is (when moving far away), the less y will be and vise versa; the lower x is (approaching), the greater y will be. It corresponds to the LED1 fading principle.

*Conversely y2=x idea is the opposite principle of the above. It corresponds to the LED2 fading principle.

*Notice that y1 and y2 act in opposite way as long as x becomes higher or lower.

*The other values such as 5 and range are there only for an easy observation of the change.
Chapter 1 Light Control System

To apply brightness and brightness2 to LED1 and LED2 respectively, we achieve this by using analogWrite(corresponding brightness, correspondingLed) to PINs corresponding to LED1 and LED2. Here the explanation why we use not pwm pin of the Arduino or pulse width modulation. The reason is that, pwm allows the pin connected to it to receive a controllable voltage (value ranging between 0 and 255) NOT a constant voltage. The Lower voltage fed to a pin, the low the brightness LED has and inversely. On a normal pin, the voltage fed to a pin is constant (255 =5V) we needed this in the first (autonomous light) feature of this system to the turn the lights only on and off.

The two LEDs will have brightness in an opposite direction depending on whether we are approach or we moving away from the sensor. Comparatively, in the smart home, as stated before; the two fading LEDs are used to help the owner have light conveniently and enlighten his or her surrounding conveniently.

1.6 Light &Temperature Code LLT v.1

```c
#include "DHT.h"
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int pingPin = 13; // setup pingpin as 13
const int ledPin = 5;
const int ledPin2 = 6;
const int fan =11;

const int winter = 9;
const int spring =10;
const int summer = 7;

const int corridor1=4;
const int corridor2=3;
const int corridor3=2;

int i=0;
int h; // temperature
int tempMin=5;
int tempMax=35;
int fanSpeed;

int brightness; // initial brightness
int brightness2;
int deltaFade = 1; // change in brightness
```
Chapter 1  Light Control System

```c
int deltaFade2 = 1;
float currentValue;
float range = 30;
int reference = range;

void setup() {
    Serial.begin(9600); // this is not necessary
    dht.begin();

    pinMode(ledPin, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(fan, OUTPUT);
    pinMode(corridor1, OUTPUT);
    pinMode(corridor2, OUTPUT);
    pinMode(corridor3, OUTPUT);
    pinMode(winter, OUTPUT);
    pinMode(spring, OUTPUT);
    pinMode(summer, OUTPUT);

    digitalWrite(ledPin, HIGH);
    digitalWrite(ledPin2, HIGH);
    delay(500);
    digitalWrite(ledPin, LOW);
    digitalWrite(ledPin2, LOW);

    digitalWrite(corridor1, HIGH);
    digitalWrite(corridor2, HIGH);
    digitalWrite(corridor3, HIGH);
    delay(500);
    digitalWrite(corridor1, LOW);
    digitalWrite(corridor2, LOW);
    digitalWrite(corridor3, LOW);

    digitalWrite(winter, HIGH);
    digitalWrite(spring, HIGH);
    digitalWrite(summer, HIGH);
    delay(500);
    digitalWrite(winter, LOW);
    digitalWrite(spring, LOW);
    digitalWrite(summer, LOW);
}

void loop() {

    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(3);
```
digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING)); a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  long duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
float cm = microsecondsToCentimeters(duration);

  currentValue = cm;
  brightness = ceil(range/currentValue)*5;
  brightness2 = currentValue*5;

if(currentValue < range){
  if(currentValue < reference){
    if(brightness <= 255){
      // change brightness for next loop execution
      brightness += deltaFade;
      brightness2 -= deltaFade2;
      // set initial LED brightness
      analogWrite(ledPin, brightness);
      analogWrite(ledPin2, brightness2);
    }
    else{
      analogWrite(ledPin, 255);
    }
  }
  else{
    // if greater than reference
    brightness -= deltaFade;
    brightness2 += deltaFade2;
    analogWrite(ledPin, brightness);
    analogWrite(ledPin2, brightness2);
  }
  reference = currentValue;
}
else{
  reference = range;
  brightness = 0;
  analogWrite(ledPin, brightness);
  analogWrite(ledPin2, brightness);
  digitalWrite(ledPin, LOW);
  digitalWrite(ledPin2, LOW);
}
float h = dht.readTemperature();
Serial.print("TEMPERATURE = ");
Serial.print(h);
Serial.println();
delay(1000);

if (currentValue < 20) {
digitalWrite(corridor3, HIGH);
}
if (currentValue < 15) {
digitalWrite(corridor2, HIGH);
}
if (currentValue < 10) {
digitalWrite(corridor1, HIGH);
}
if (currentValue <= 17) {
    if (h <= 20.00) {// WINTER
        digitalWrite(summer, LOW);
        digitalWrite(spring, LOW);
        digitalWrite(winter, HIGH);
        analogWrite(fan, 0);
        /* for(i=0;i<80; i++){ // show winter
            analogWrite(winter, i);
        }
        delay(10);
        for(i=80;i>0;i--){
            analogWrite(winter, i);
        }*/
    }
    else if (20.00 < h <= 25.00) {// SPRING
        digitalWrite(winter, LOW);
        digitalWrite(summer, LOW);
        digitalWrite(spring, HIGH);
        analogWrite(fan, 80);
        /*for(i=0;i<80; i++){ // show spring
            analogWrite(spring, i);
        }*/
    }
}
Chapter 1  Light Control System

```c
    delay(10);
    for(i=80;i>0;i--){
        analogWrite(spring,i); /*
    }
    
    if(h>25) { // SUMMER
        digitalWrite(fan,HIGH);// turn on fan
        digitalWrite(summer,HIGH);// show summer
        digitalWrite(winter,LOW);
        digitalWrite(spring,LOW);
    }
    
    // out of the room
    else{
        digitalWrite(winter,LOW);
        digitalWrite(summer,LOW);
        digitalWrite(spring,LOW);
        digitalWrite(fan,LOW);
    }
    delay(10);
    }

    float microsecondsToCentimeters(long microseconds) {
        // According to Parallax's datasheet for the PING)), there are
        // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
        // second). This gives the distance travelled by the ping, outbound
        // and return, so we divide by 2 to get the distance of the obstacle.
        return microseconds / 29 / 2;
    }
```

In conclusion, the smart home light feature is achieved in this project by the above key codes that I’ve tried to describe in this report. All knowledge from the previous years was fundamental for any engineer either in a hardware course or software course as well. It comes along needed; some knowledge about mathematical theory to compensate the work. As I said earlier about my interest on light, I think and I believe that no one can bypass the needs of light and that’s one of the main motivations for me to try to deal with and go too deep in it. I know, what I’ve done in this project is not yet to all extremities of what I imagine the future version of the light control system to be, but I recognized that it helps me a lot to achieve more interesting goals in the future.
Chapter 2  Temperature Control System

2.1  Overview

I choose temperature control because I think for a “smart home” or even a “normal home”, an air-conditioning system is always a useful thing in our daily life; so I came up with this idea of making a fan controlled by weather changes or can be referred to as an automated air-con system. I did my research in a couple days and but most the content online wasn’t exact to what I want to do, most didn’t use software but I end up on a website which gave me a view of what I wanted to do and even some direction in order to achieve it, in which I could to remove some things not needed or even modify it; it wasn’t easy to find but eventually I came across this website. www.electr9schematics.com/9540/arduino-tan-speed-controlled-temperature/

2.2  Beginning Steps

2.2.1  Required components

I started working on the temperature system, I found the components I needed and then I started putting everything together. The following were the components needed for the temperature controlling system (TCS): a. Arduino (with atmega328p)
  b. 1k resistor
  c. 100uF Capacitor
  d. BD139
  e. 5V FAN
  f. 1N4007
  g. Dht11

2.3  Seasonal Variance

The air-conditioning system only show us three type of states varied by season specifically winter, spring, and summer. As the introduction states, “this is the first version of our project.. there are a lot of functions we could introduce if we had enough time and enough money!! Lets get back to the topic, the three LED’s on the TCS show different states in each of the three seasons.

2.3.1  Summer

Let’s begin with summer which usually very hot and people are sweating so much, so during summer the red LED will be on to tell us it’s summer and our fan speed will be fast in order to
cool down the temperature so that the people who re there can be relieved; the temperature will also be displayed on the LCD. The temperature range during summer is 25 degrees; so for this system 25 degrees and above is summer.

2.3.2 Spring

For spring only the green LED will be on to tell us that it is spring time now and our fan will spin at a normal speed and not as fast as summer. We all know during spring the fan is not needed much so for this system the fan speed will be low, and the temperature range for spring is between 20 and 24 degrees; and also the temperature and humidity can be displayed on our LCD screen.

2.3.3 Winter

We’re now down to our last season which is winter, during the winter time no one will use fan because it’s cold already, so at this time our yellow LED will be on to tell us it is winter season now and our fan will be completely off because it’s winter; the range for winter is below 20 degrees, that is where winter starts according to our system.

2.4 conclusion

So this is how the part of the temperature control system is working; the system is automatic and doesn’t require an external effort from the people it is serving for it to change the temperature in the home environment, just electric power.

2.4.1 challenges

During my working time I encountering a few problems and hard times, like finding the components even loosing some of them during the early stages due to improper care!!! When
Chapter 2  Temperature Control System

2.4 conclusion

everything was working ok on the breadboard, then I start working on putting the equipment on the circuit board and soldering them along with my partners.

(a) Temperature Control System ver.2

(b) Light-Temperature Control System ver.2

2.4.2 final peice

This is how the circuit looks after soldering, it took us at least 4 days to complete the final stages, but thank God we did it even at a high difficulty level, we manage to finish it. and then after that we started placing it in to the puzzle house together which was the most difficult part for us, because it was our first project; actually we were encountering problems each and every day some pertaining to hardware and other software issues; I think me and my group learned so much during all the whole period of the project.
Figure 2-3  Light-Temperature Control System

This was almost the final process.
Chapter 3  LION SECURITY SYSTEM

3.1 intro

…why I choose a Security System It’s always been my dream, this is why I came to China to Study. My family moved to the United states from Malawi when I was turning 9 years old after my parents were held at gun point by a large number of robbers in our home shortly after one of their business trips. I know our Lord Jesus protected them and after I finished High School and started college in the US, by the end of my first semester I discovered I wanted to develop a home security company that would help protect the lives of so many people as get killed in my home country because the country still has a low security level; it took 4 days for the police to come to our home after we were robbed and my parents lost their hard work but we had our lives spared .. our old posessions were never located again after ...

NOTE: In this project, we are programming 2 seperate ATMEL atmega328p-pu microchips for two internetworked systems.

3.2 Security/Automated door system

I was working on the security of the home network which will be triggered by interference from inside the network or from an outside intrusion on the home network. The parts of the security will include sensors and surveillance to monitor any interferance on the home network to prevent any intrusion. I will also be working on making the home network a “time driven” network that will turn on and turn off some home functions based on time.

After our first meeting, I went on to research some simple security systems based on arduino and that I can improve to meet the needs of this smart home network; then I stumbled upon the instructable at the following website: http://www.instructables.com/id/Arduino-Home-Monitor-System/?ALLSTEPS It included all the basic requirements that I could use or remove from the system I was looking to develop.

3.3 Process

3.3.1 component list

The first step was to follow every step in the instructable and make sure the system and code was functional and can actually be used. I organized the following list of components that were required for this project:

-Arduino IDE (Integrated Development Environment)
-Arduino Uno
- 16x2 LCD Display
- 10k Potentiometer
- Magnetic Reed Switch (replaced with a normal switch)
- PIR Motion Sensor
- DS1307 Real Time Clock
- 32.768kHz Crystal
- 3V Coin Cell Battery
- 5v Regulator (LM7805)
- 10uF Electrolytic Capacitor x2
- 10k Resistor x3
- Piezo Speaker
- LED
- Iron Ferrite Choke (not needed in this model setup)
- A generous length of CAT5 cable
- breadboard
-circuit board

-Soldering Iron

-Solder (of course)

-wire

I acquired most of the required components that I will be using and eliminated the ones I don’t need; (i.e. the magnetic reed switch because it was expensive and would not be required for this model demonstration). I replaced it with an emergency switch that could also activate the security system and was much more affordable by about 80

I started developing the security system. After following the guide and steps I was able to develop the first version of the security system.
3.4 Security Features/ Functions

3.4.1 system modifications

The first version of the security system was identical to the one specified in the original instructable which included only 2 security features: motion sensing using a passive infrared sensor (PIR), and a door breach alert using a magnetic reed switch. I ended up removing the magnetic reed switch feature attached to one PIN because I did not see necessary for this model home but later on in the 2nd version of the Lion Security System, I replaced that security feature with an emergency security switch attached to the same “door pin” on the “atmega328p” microchip to be triggered by the home occupants during the course of an emergency. I also implemented the automated door system into the security features and I was supposed to work together with one of our teammates but due to other reasons he left the group and I ended up doing the automated door system for the home too because I was very interested in developing it. I also updated the code for the system; first to display both date, time, temperature, and humidity on the screen, and also control the automated door system, and sync the alarm sound with the LCD screen to display important system notifications on the screen to show the system status (i.e. “Intruder Alert” when the emergency switch is active).

3.4.2 security center

I then thought of adding more security features to make the building more secure during an emergency situation. I thought of it in a practical way; what I would want to happen in my home if there was an emergency, so I developed a system that required a “Security Center” with a security guard to keep watch of the premises and clear the security system after the home is secure. This system also locks the building down if there’s an emergency (“Code Red Alert”) and seals all the doors until the police and Security Center have been notified and have cleared the system; the system status then displays (“LION IS SAFE”).

3.4.3 time-controlled system

Since one of our goals was to create a time controlled system, I used the clock already available in the research security system which originally included the time function and library using a “DS1307” clock. I later improved the code to control the PIR (Passive Infrared Sensor) by time; I imagined that during day time, a smart home would have a lot of activity/motion from people and/or pets moving around the house and that would constantly activate the alarm system of the building, so then I decided to only activate the PIR automatically at 6:00pm just for this demonstration, in a real smart home setting it would be set to activate at an hour when all the home occupants and pets have all left for bed; which could also be determined by the PIR sensor if there was no motion for a certain period of time. The system also activates the external
lights of home at 6:00pm, I connected 5 LEDs in series to a single LED pin on the atmega328p microprocessor.

3.4.4 automated door access

The last feature implemented to the security system is the automated door access control. This feature is very important to this security part because there would be no need for security if the door was always open. The automated door control system is the key feature of our security system. I originally purchased two "SG90 Micro Servos" from "taobao.com" along with my Arduino uno before deciding to make automated doors for this house, I wanted apply them to a wifi car project I am working on. After learning how to control a servo motor online, I began to program the automated door feature into the security system. It was challenging but I thank God I was able to achieve this also. I first calculated the initial angle that I wanted the door to be opened at. The servo motor was designed to open only 180°s so after calculating my initial point to be at the 120th degree and to open at the 0th degree, I programmed that into the atmega328p microchip. Then I applied the door access granted/denied feature into this system and also auto door locking before access is granted a during a “Building Lock Down” emergency.

3.5 Programming an Atmega328p Chip

3.5.1 burning the bootloader

Burning the boot loader/ soldering process For a standalone system, the basic requirement was to burn a “bootloader” for the Atmega328p to operate outside the Arduino environment. This was necessary for us because this was a stage we wanted to reach from the beginning. After researching, I came across a link that guided me step by step on reaching this stage using an Arduino Uno and 2 atmega chips that inter-programmed each other and burned the “bootloader” on both chips.

http://www.arduino.cc/en/Tutorial/ArduinoToBreadboard

3.6 Soldering (Final Process)

After multiple tests on the breadboard, I then moved this system and soldered it to the circuit board sketching an idea of where I imagined everything to be. After overcoming a few challenged and minor mistakes (mainly during the connection stage after soldering) every part of the security system began to function properly. This marked a peak moment in our project when both systems were functional even with minor errors that we eventually resolved.

3.6.1 Lion Security System Code LSS v.1
LION SECURITY SYSTEM (LSS)
Kenyon Nsunza
NOTE: DS 1307 LIBRARY REQUIRED

#include <LiquidCrystal.h>
#include <Wire.h>
#include "RTClib.h"
#include <Servo.h>
#include "DHT.h"
define DHTPIN A1
define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int pir = 2; // Passive Inferred Sensor
const int emergency_seal_exit = A3; // emergency switch to seal the building during day time.
const int led = 4; // Led Alert Visual Signal
const int alarm = 5; // Alarm Sound Signal Signal
const int dooropen = 3; // Button to open door

// Liquid Crystal LCD with pin assignments on arduino uno
LiquidCrystal lcd(7,8,9,10,11,12);
RTC_DS1307 RTC;
int servoPin = 13;
Servo servo;
int angle = 120;
int building_lockdown = 0;
int emergency_exit; // in one central room single press, autolock after external button is pressed to seal building until remote reset from security_center.
int security_center; // unlocks front door (using passcode) for system reset
int pir_on;
int pir_off;
//int tempPin=A1;

byte thermometer[8] = //icon for thermometer
{  
  B00100,
  B01010,
  B01010,
  B01110,
  B01110,
  B11111,
  B11111,
  B01110
};

byte humidity_droplet[8] = //icon for water droplet
{  
  B00100,
Chapter 3  LION SECURITY SYSTEM

void setup(){
  // LCD row and column and dimensions 16X2
  servo.attach(servoPin);
  lcd.begin(16,2);
  Wire.begin();
  RTC.begin();
  // servo.attach(servoPin);
  lcd.createChar(1, thermometer);
  lcd.createChar(2, humidity_droplet);
  dht.begin();

  if (!RTC.isrunning()) {
    // Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  pinMode(dooropen, INPUT);
  pinMode(pir, INPUT);
  pinMode(emergency_seal_exit, INPUT);
  pinMode(led, OUTPUT);
  pinMode(alarm, OUTPUT);
  pinMode(security_center, INPUT);
  pinMode(tempPin, INPUT);

  //display message on the LCD
  lcd.setCursor(0,0);
  lcd.print("Initializing LSS");
  lcd.setCursor(0,1);
  lcd.print(" Please Wait.. ");
  delay(5000); //PIR Calibration time
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" LION IS ONLINE ");
  delay(2000);
}

void signal_pir(){
  digitalWrite(led, HIGH);
  tone(alarm, 300);
  delay(100);
  digitalWrite(led, LOW);
  delay(400);
  digitalWrite(led, HIGH);
Chapter 3  LION SECURITY SYSTEM

tone(alarm, 200);
delay(100);
digitalWrite(led, LOW);
delay(400);
digitalWrite(led, HIGH);
tone(alarm, 300);
delay(100);
digitalWrite(led, LOW);
delay(400);
digitalWrite(led, HIGH);
tone(alarm, 200);
delay(100);
digitalWrite(led, LOW);
delay(400);
}

void signal_emergency_seal_exit(){
digitalWrite(led, HIGH);
tone(alarm, 300);
delay(100);
digitalWrite(led, LOW);
delay(400);
digitalWrite(led, HIGH);
tone(alarm, 200);
delay(100);
digitalWrite(led, LOW);
delay(400);
digitalWrite(led, HIGH);
tone(alarm, 300);
delay(100);
digitalWrite(led, LOW);
delay(400);
digitalWrite(led, HIGH);
tone(alarm, 200);
delay(100);
digitalWrite(led, LOW);
delay(400);
}

void trigger_pir(){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" LION IN DANGER ");
signal_pir();
}

void trigger_emergency_seal_exit(){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" INTRUDER ALERT!");
signal_emergency_seal_exit();
}
void loop(){

  DateTime now = RTC.now();
  //pir_on = (now.hour() >= 0 & now.minute() >= 59 & now.second() >= 59);
  if(now.hour() >= 18){
    digitalWrite(led,HIGH);
  }
  if(now.hour() <= 7){
    digitalWrite(led,HIGH);
  }

  int emergency_seal_exit = digitalRead(A3);
  int pir = digitalRead(2);

  if(emergency_seal_exit == HIGH){
    building_lockdown = 1;
    trigger_emergency_seal_exit();
  }
  else if(pir == HIGH & now.hour() >= 18){
    building_lockdown = 1;  // lock door with all the force inside the motor + screws
    trigger_pir();
  }
  else if(pir == HIGH & now.hour() <= 6){
    building_lockdown = 1;  // lock door with all the force inside the motor + screws
    trigger_pir();
  }

  else{
    noTone(alarm);
    lcd.clear();
    DateTime now = RTC.now();
    lcd.setCursor(7,0);
    lcd.print(now.month(), DEC);
    lcd.print(\'\.'\);  
    lcd.print(now.day(), DEC);
    lcd.print(\'\.'\);  
    lcd.print(now.year(), DEC);
    lcd.setCursor(0,0);
    lcd.print(now.hour(), DEC);
    lcd.print(\'\.'\);  
    lcd.print(now.minute(), DEC);

    //float degree = digitalRead(tempPin);  // to get the temperature value
    //degree = (5.0 * degree * 100.0)/1024.0;
    float h=dht.readTemperature();

    lcd.setCursor(1, 1);
    lcd.write(1);
    lcd.setCursor(3, 1);

int dooropen = digitalRead(3); // door read not exact because of loop
if (dooropen == HIGH && building_lockdown == 0) {
  servo.attach(servoPin);
  // open the front door to 120 degrees
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" access granted ");
  delay(1500);
  for (angle = 120; angle > 0; angle --) {
    servo.write(angle);
    delay(5);
  }
  servo.detach();
  delay(3000);
  servo.attach(servoPin);
  // now close the door back from 120 to 0 degrees
  for (angle = 0; angle < 120; angle ++) {
    servo.write(angle);
    delay(15);
  }
  dooropen = digitalRead(3);
}
else {
  angle = 120;
}
int security_center = digitalRead(6);
if (security_center == HIGH) {
  building_lockdown = 0;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" LION IS SAFE ");
  delay(2000);
}
3.7 5V Power Supply System

This last part was a major part of our project because none of the smart home functions we are introducing are real if we cannot power them. After searching online for a way to power both systems I came across a circuit in a video I watched that had a 5V regulated system using an “LM7805” transistor.

![5V power supply diagram](image)

Figure 3-3 5v power supply unit

After testing the system on a breadboard I moved it to a circuit board. This is when we met the biggest problem in the course of this project that for a moment seemed to have completely killed our project.

3.8 Disastrous error

3.8.1 Problem encountered

The project was finished, the only part we were left with was how to place the door of the house into the design; it required creative planning to fit everything in place. After the power supply was tested and working on the breadboard I decided to test it on the system to see if both the security system (LSS) and the light&temperature system (LLT) could be powered by
a single power supply or if we needed 2 power supplies as it already was; using 2 Arduino Uno 5v boards. There was no information we found online about this and how much exact voltage was enough for an atmega328p chip; not even the data sheet had information on the “Current” limit for the chip, only that the system is powered by a minimum 5volts. A 12v power supply was ok for the Arduino Uno which converts it to 5V, other sources also showed that up to 35 Volts was ok for the “LM7805 5v regulator” that I was already testing. After a long struggle of research with no true answer, except that the above schematic that showed a 9v power supply regulated to 5v by the “LM7805” and the standard 9v battery did not have enough current for both systems, and I could not find a 250volt AC to 9volt DC power supply nearby, and also with the limited understanding of what a “switching power supply” (required for an atmega chip) is and the 5v “nonswitching power supply I had available wasn’t working”; I discussed with my teammates and after seeing the current state of the project and that none of us were satisfied to power the system with simply an Arduino Uno and that the deadline for this project could be due any day, they trusted me to run some tests. First I tested only the breadboard power supply circuit I designed from the schematic using the LM7805 transistor by supplying it with 12V (1.25mA) power supply instead of the 9V showed in the circuit (from my view this was how we could power both atmega chips because other sources showed it can take in up to 35Volts). After trying this test for the first time, the voltage regulator still maintained the 5volt output. After sharing this with my teammates we saw it ok for me to build the circuit. Eventually after sometime passed we just saw a “power surge” that seemed to have turned both systems off to a “0” (low) working state, with an absolute non-functional state for the system . After all our hard work after the hard work the guys put it in, I could help but take the blame for letting my team down as a leader. They did not blame me at all they continued to trust me and to trust that everything will work out, all that could flash in my head were the words from my Electrical &Electronics teacher 5 years ago in my country; in Electrical engineering 1 mistake has 200 times more consequences. I had Ideas on what might have happened but not to what extent, until I researched more on the atmega328p, mainly a “bricked chip”. The manual and sources showed that the chip has digital fuses (fusebits) and that they autonomously switch off the output pins when the chip is supplied with a high voltage above the normal limit that can be set. This was a relief after a week of seeing the smart home in a dead state. I knew all we had to do next was to purchase at least 1 new Arduino Uno with an operating atmega328p just like in the beginning of the project and extra chips that we can program and keep available. This recovery procedure would cost us an extra 60 rmb, but we were all willing because it was worth it to us, we had already put in enough time and money to leave this project. This resolved our problem and restored the system back we all joyed in seeing “LION IS ONLINE” more than before.
3.8.2 other problems encountered

The other problem I encountered was the servo motor easily breaking in the last stages of the project, because to tighten the screw form the door to the motor required a lot of extra force that this chip (15rmb motor) could not handle, so about 3x 15rmb motors broke, then I decided to order 2 (25rmb) metal gear servos from taobao, but they were not applicable into this system because they only rotate 360° in one direction; I broke 1 of the two also while trying to configure it with the 180° system form the broken servos, I also failed at this method. The last solution was to purchase 1 more plastic “SG90 Micro Servos” and just be careful and find a new way to mount the door to this chip servo now that I know how fragile they are (hence the ugly door on the project lion model). In total this recovery process cost me an extra 110rmb.

*This project could have cost us only 284rmb without these mistakes.

3.9 Conclusion

Lion Security
Why Lion? (LION PRIDE)
Q. What’s safer than a lion?
A. Another Lion that defeats it.

Lions are the only cat species that form social groups. All other cats are lone hunters. The social groups lions form are called prides. A pride of lions typically includes about five females and two males and their young.

Males must win their way into a pride of females and once they do they must fend off challenges from males outside the pride who try to take their place.

http://animals.about.com/od/l/g/lion-pride.htm
Chapter 3  LION SECURITY SYSTEM

3.9  Conclusion

Figure 3-4  Team Lion
Acknowledgement

Team lion consists of three Batchelor Degree students in the Field of Telecommunications Engineering. As engineering students each one of us has developed skills in specific aspects of our major that we applied in this project. There was no criteria when we made this Team only that we were looking for individuals interested in our major, willing to learn and develop new ideas, willing to stay up day and night, and sacrifice our winter holiday to achieve all our goals. We managed to organize a hardworking team that has put in all efforts toward innovating smart homes. Each one of the team members has contributed outstanding ideas to this project that come together to make up "Project Lion”.

We would like to thank our Teacher Dr. Zeng Yujiang for the opportunity for presenting us the opportunity to work on this project and the guidelines he gave us to accomplish what we didn’t know we were capable of; the inspiration video during the presentation of what other students in china have been doing really motivated us to push this project as far it did.

We also want to take this time to thank each member of Team Lion for the skills they have brought into this project and also learned and developed during the course of the project. Before that though we want to thank Mr. Kenyon’s girlfriend “Tonthozo” who took her time to put the pieces of the puzzle (Model House) together, we would not have a house to put all our skills inside if it were not for her; maybe a shoe box. Next we want to thank Our Leader Kenyon Nsunza (from Malawi) who helped us to think ”outside of the box” to actually design the model home and contributing his hardware, video editing, and writing skills to our project and improving each part. Faniry Sandamahery (from Madagascar) who contributed his skills in the C programming language to each part to resolve errors and also his ideas and strong work ethic has made our project better than it would be. Our team also includes a skillful member Mr. Salum Issa (from Tanzania ”happo vippi”) who brought in the best ideas to our project to make the home a livable environment and also contributed his soldering skills in our project, our project be as strong without his skills and strong commitment.

This was a very great learning experience for Team Lion and the memories, skills developed, success, failure will not be forgotten and will contribute to make our next projects become more real and more innovative to our various countries. We are a growing team and we enjoyed working together and we would like to keep this Team working for the rest of our engineering experience and bring our developed skills together from whatever location we end up.