GNG1103

Final Design Report

Automation 3 Final Report

Submitted by

Automation 3

Noah Fisherman, 300078707
Tal Marianovski, 300069572
Sinthu Jeyanathan, 300071893
Jay Jiuk Chang, 300066454
Brydon Jewett, 300074338

April 15th 2019

University of Ottawa

Abstract

The goal of this project was to provide an a low-cost, modular home automation system for a modular small-home. The priorities of the product were to reduce energy consumption and maximize modularity. The team decided to integrate three subsystems, opting to automate the lighting, and temperature of the house, as well as create a door-monitoring system, with an audio and visual alert inside the house when the door was being approached. The team was able to successfully create this system with a final cost of $106.75, and was able to learn vital skills such as project management and arduino coding in the process. Moving forward with the project would take the form of optimizing the wiring, and improving user access and modularity, as well as increasing user control over the temperature system, although the project was very successful despite this.

Table of Contents

List of Figures

List of Tables

List of Graphs

Introduction

Approximately every 6 days, a woman is killed by an intimate partner in Canada. There are Over 500 shelters for women and children fleeing violence in Canada, and on any given night, almost 4000 women and 3000 children sleep in shelters because it is not safe at home. On any given night, about 300 women and children are turned away because shelters are full. Our client-The Ottawa Violence Against Women Shelters- require small modular housing due to lack of space in their facilities. Our goal was to create an automated system to implement in the modular house that would allow the residents to live comfortably while also saving energy. Our modular solution will contain motion activated lights, temperature sensitive cooling system, and range detection door alerts.

For our motion activated lights we decided to use LED lights. LED lights are more energy efficient and use up to 90% less energy than traditional light bulbs. Using strip lights allowed us to make the lighting more modular. The lights are attached to PVC pipes which can be hung up along the walls of the house.

For the temperature sensitive cooling system, we utilized only a fan. This allowed us to have a lower cost than if we also used a heater. Only using a fan also utilizes less energy and can be easily adapted to using a heater instead, or to using both. We used an embedded temperature sensor chip that allowed the use of OneWire and DallasTemperature arduino libraries and also provided simplified temperature calculation.

For the range detection door alerts we used an ultrasonic sensor placed next to the door to alert residents of approaching people. Residents of the home will be alerted by both a LED light ring and a sound alert to account for possible disabled residents. We used and Ultrasonic Rangefinder Easy to implement into house. This sensor reliably monitors the area surrounding door. We utilized both an LED Ring and Speaker Alarms as alerts which Increased accessibility and allowed even those who may be blind or deaf to utilize the door monitoring system. Both the LED ring and speaker fit within the box to Increase modularity, as the full box can be moved as a unit

Need Identification and Product Specification process

Customer/User Need	**Importance** **(5 most important)**
Design reduces the overall energy consumption of the house	3
Design is safe for children, without exposed wires	5
Solution remains low-cost	3
Design provides adequate lighting for the house	5
Solution is modular, easy to assemble, and can be added on to at a later date	4
Solution increases accessibility of the home, allowing disabled users to easily utilize the home’s lights and other automation capabilities	4
Lights can be triggered remotely, and do not need to be manually turned on	3
Automation maintains aesthetic of the house, minimizing exposed components	3
House automatically adjusts temperature to the desired setting, that can be adjusted by the user	4

Table 1- Customer Needs Categorization

Product Specifications

All subsystems were connected to an arduino and all elements of each subsystem interact through the arduino.

Motion Activated Lighting

Customer need satisfied: *Provides adequate lighting, Lights are remotely triggered, Reduces overall energy consumption, Solution is modular, Low cost.*

Lighting is a very important design aspect that is essential to the project. Using LED light strips was an obvious choice as they are energy efficient, low cost, and easy to implement. Our original design had us attaching the light strips directly to the wall but when implementing the system we decided to attach the lights to pvc pipes to ensure modularity and ensure the longevity of the adhesive.

We chose to place the lights on a motion sensor in order to increase energy efficiency because lights cannot be left on for excessive periods of time if someone is home. The motion sensor also makes the house more accessible for persons with disabilities as the lights can be turned on by those without fine motor skills. During prototype development we tested different delay periods to determine what was optimal so that lights wouldn't turn off to suddenly but would stay off during sleep.

Ultrasonic sensor + Dual door alert

Customer need satisfied: *Increases accessibility to disabled users*

By using an ultrasonic sensor in place of a door bell, the resident can be made aware of people outside the home for long periods of time. We chose a ultrasonic sensor since it uses sound not light and is therefore not hindered by over or under exposure, making it ideal for outdoor use. Since the sensor is placed on a delay people simply passing by the house won't trigger it. We opted to use 2 different alerts, an LED ring and a speaker. This ensures accessibility to persons who may be deaf or blind as they would only be able to use one of the 2 alerts.

Cooling fan + Temperature Sensor

Customer need satisfied: *Automatically adjusts temperature to the desired setting, Low cost*

We knew from early on that implementing both a heating and cooling system would be outside our budget. We chose to implement only a cooling system for this reason. We used a simple temperature sensor and set an appropriate temperature range in the arduino code. We chose to use a standard standing fan for cooling. When the temperature of the house is too high the fan turns on, the fan turns off if the temperature is too cold. The fan’s power cable is stripped at the end and connected to the relay allowing the system to turn on and off by controlling the power. Similar to the lighting system automating the cooling system conserves energy as the fan won’t be on unnecessarily. Using a temperature sensor and preset range also increases accessibility to those with limited mobility.

Conceptual Designs

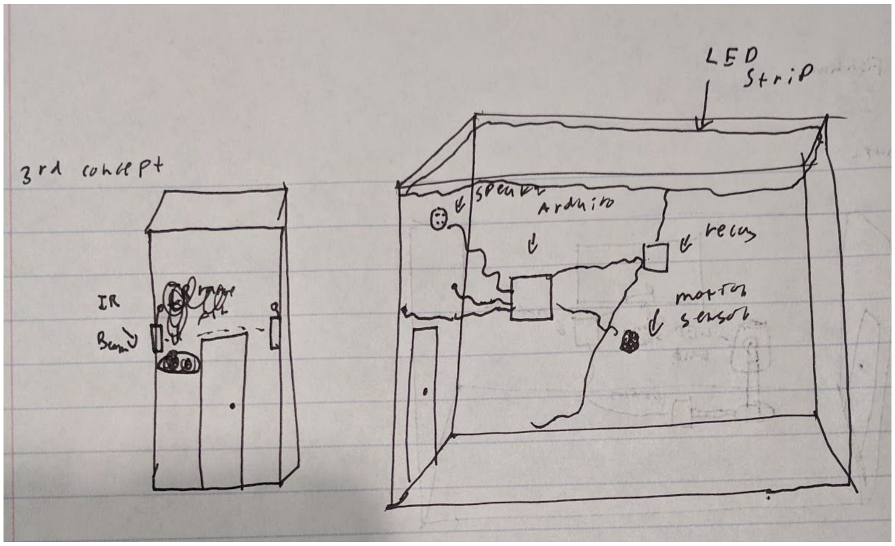


Fig 1 - Design Idea 1

LED Light strips provide adequate lighting while minimizing current draw, and can be easily controlled with an arduino system. We are also considering using a basic lamp with a relay system that could also be easily controlled with an arduino. The lights would likely be controlled with a light or motion sensor, or some combination of the two. Additionally, an IR range finder or motion sensor could be installed to the front of the house. This would communicate to the resident if the door was being approached through either a light or speaker within the house, allowing the resident to better understand their surroundings, and be safer.

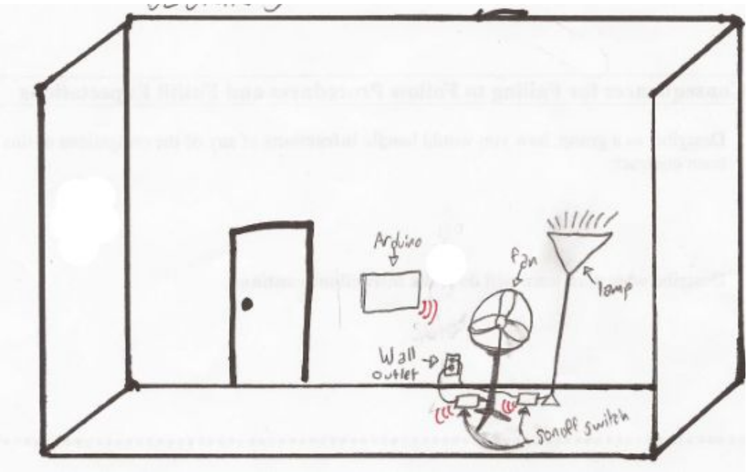


Fig 2- Design Idea 2

This design has a regular standing fan which has a Sonoff switch attached to it. The Sonoff switch makes it possible to connect any regular standing fan to the arduino board. This also means that the fan can be controlled remotely from the homeowners app and the app can detect if the fan has been on for too long and alert the homeowner. The same can be done with any regular standing or table lamp.

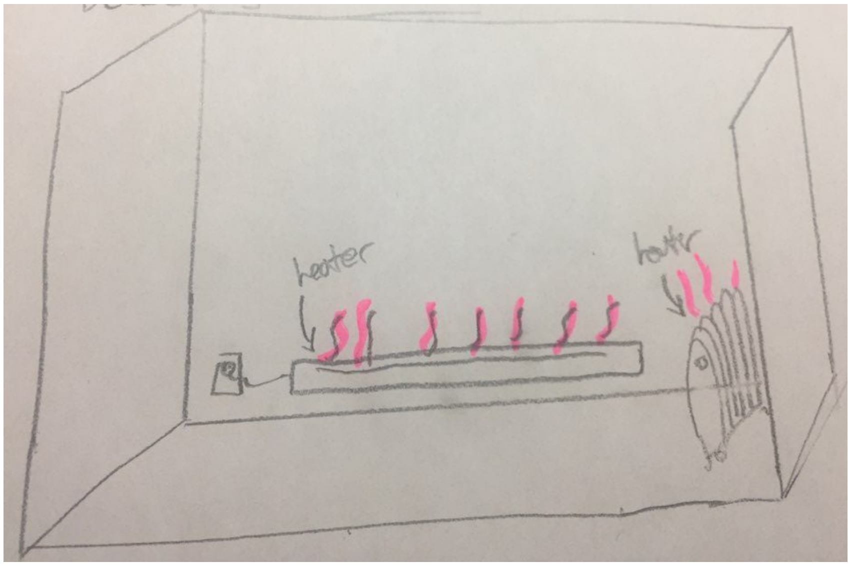


Fig 3 - Design Idea 3:Heater
Since the project take place in Ottawa it has to endure cold weather so there should be a heater that can warm up the place.

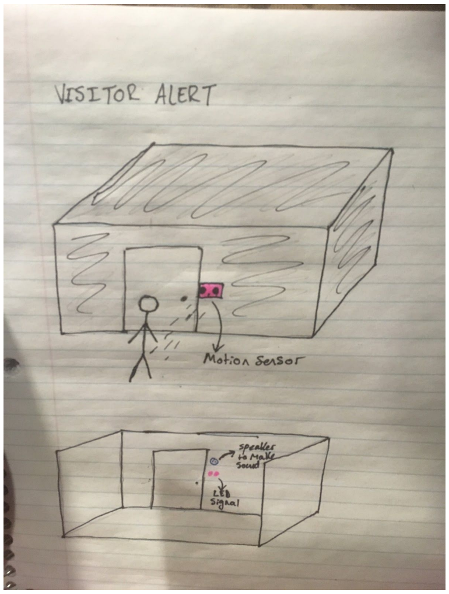
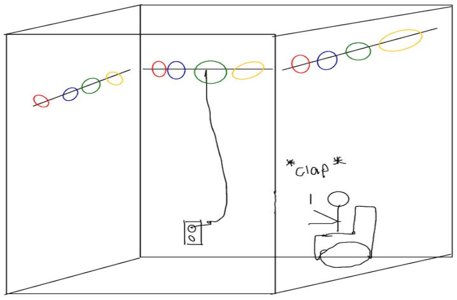
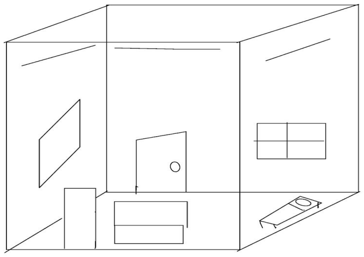


Fig 4 - Design Idea 4:Visitor Alert
The Alert uses a motion sensor to alert the tenant when someone is approaching the home. To alert the tenant two LEDS will light up and a noise will sound coming from a speaker next to the door.



clap activated lights

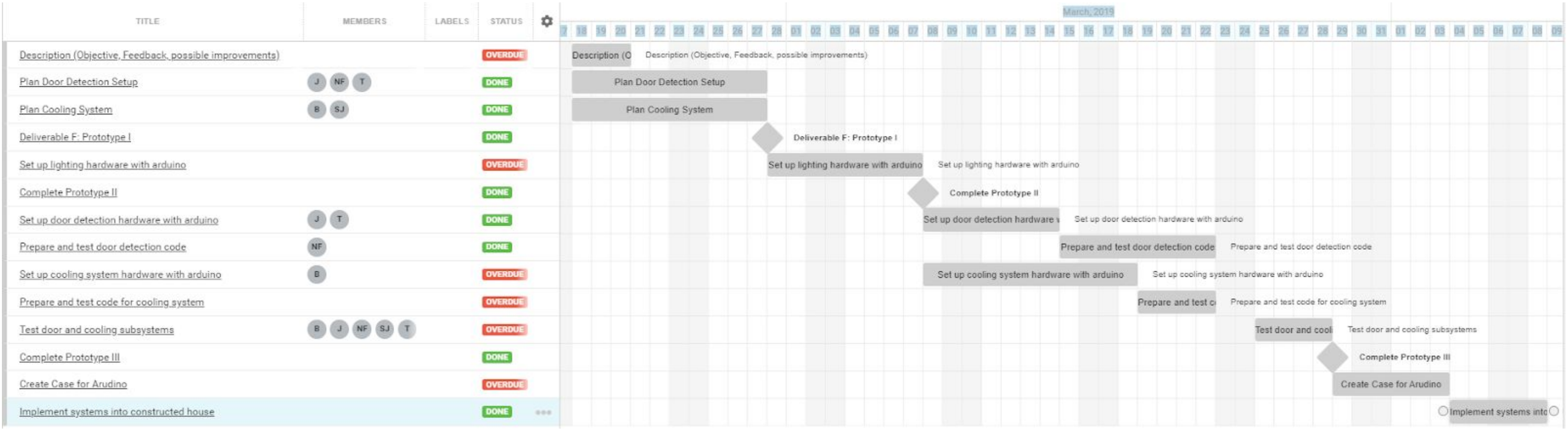
Fig 5- Design Idea 5:
This design has LEDs in home and to ensure that any resident of the home can easily access these lights, we got a idea that outfit them with “The clapper”. This device allows the resident to turn on and off the lights by either clapping their hands or simply making two brief consecutive noises.



No exposed wiring to ensure safety for children

Fig 6 - Design Idea 6:
All wiring involved in the automation of this house will be kept to the minimal and securely hidden so in the case of a child living in the home, they will be in no danger of electrocution. This also makes the house more aesthetically pleasing.

Project Plan, Execution, Tracking & Bill of Materials



TITLE	MEMBERS	LABELS	STATUS
Description (Objective, Feedback, possible improvements)			OVERDUE
Plan Door Detection Setup	J NF T		DONE
Plan Cooling System	B SJ		DONE
Deliverable F: Prototype I			DONE
Set up lighting hardware with arduino			OVERDUE
Complete Prototype II			DONE
Set up door detection hardware with arduino	J T		DONE
Prepare and test door detection code	NF		DONE
Set up cooling system hardware with arduino	B		OVERDUE
Prepare and test code for cooling system			OVERDUE
Test door and cooling subsystems	B J NF SJ T		OVERDUE
Complete Prototype III			DONE
Create Case for Arudino			OVERDUE
Implement systems into constructed house			DONE

March, 2019
17 18 19 20 21 22 23 24 25 26 27 28 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03 04 05 06 07 08 09
Description (O
Description (Objective, Feedback, possible improvements)
Plan Door Detection Setup
Plan Cooling System
Deliverable F: Prototype I
Set up lighting hardware with arduino
Set up lighting hardware with arduino
Complete Prototype II
Set up door detection hardware v
Set up door detection hardware with arduino
Prepare and test door detection code
Prepare and test door detection code
Set up cooling system hardware with arduino
Set up cooling system hardware with arduino
Prepare and test c
Prepare and test code for cooling system
Test door and cooli
Test door and cooling subsystems
Complete Prototype III
Create Case for Arudino
Implement systems into

Fig 7 - Project Planning

We utilized a Gantt chart to plan our project, breaking up each milestone into applicable tasks, and maximizing overlap between members wherever possible. We utilized a trello board to track our process, and were generally able to follow our gantt chart by meeting 1-2 times out of class and labs per week. This allowed us to properly execute our plan and complete the system properly by the design day date, and stay up to date on all deliverables. We prioritized finishing the subsystems in order of their importance, beginning with the lighting system after planning all subsystems. In executing our project plan, we created and tested each subsystem individually, integrating them together as the last step in our process. This allowed us to develop a deep understanding of each subsystem before moving onto the next, as well as enabling us to apply the lessons learned in a previous subsystem to the next. This proved to be a good system that enabled us to finish our overall subsystem and integrate it before design day.

Unit Name	Quantity	Unit Cost	Extended Cost
Arduino Uno	1	$25	$25
LED Light Strip	1	$11	$11
PIR Motion Detector	1	$8.99	$8.99

Standing Fan	1	$19.99	$19.99
Temperature Sensor	1	$4.95	$4.95
4-Channel Relay	1	$10.86	$10.86
Solderable Breadboard (6 Pcs.)	.16667	$16.88	$2.81
Ultrasonic Sensor	1	$4.95	$4.95
PCB Mount Mini-Speaker	1	$3.45	$3.45
12-Neopixel Ring Light	1	$9.95	$9.95
22-AWG Hook-Up Wire (5Ft)	3	$1.60	$4.80
Total:			$106.75

Table 2: Bill of Materials

The team went slightly over our desired budget of $100, however, doing so was vital to ensure all aspects of the desired design were properly integrated. In order to cut costs slightly, the team could have used a simple LED in place of the NeoPixel ring to illuminate when the door alert was triggered, however, doing so would be a less obvious visual alert than the spinning lights on the neopixel ring.

Analysis

Analysis of Lightbulbs Vs. LED strip lighting:

Power Draw of LED Strip Lighting:[1]

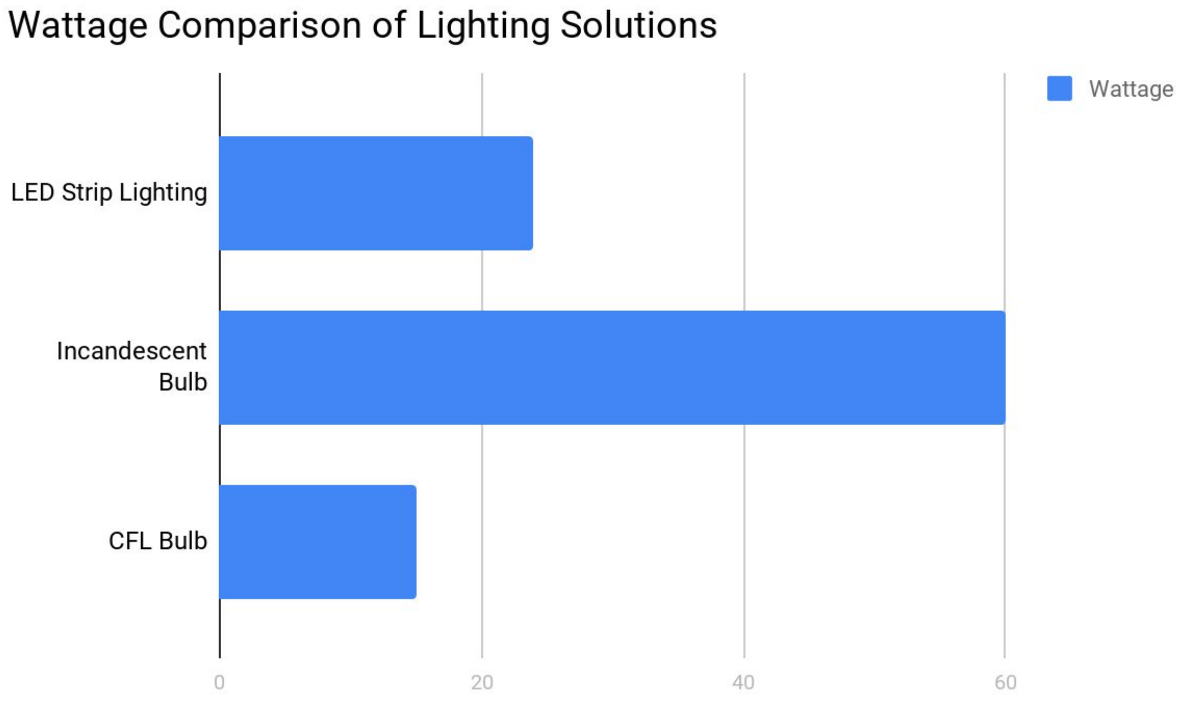
$$4.8\ Watts/m\ [1]\ *\ 5\ m\ = 24\ Watts$$

Power Draw of Standard Light Bulbs producing 800 Lumens:[2]

Incandescent: 60 Watts

CFL Bulb: 15 Watts

Graph of Different Lighting options:



Graph 1: Wattage Comparison

With the wattage of lighting considered, we decided to utilize LED Strip Lighting. Although CFL lights would have provided similar brightness of lighting whilst pulling less wattage, we decided that utilizing LED strip lights would be easier for us to control with the arduino we had planned to use. Additionally, it was more economical to utilize LED strips, as these strips were far cheaper than purchasing both a CFL lightbulb, as well as a lamp to utilize it with. Finally, using LED strips allowed us to easily raise the lighting into the ceiling, taking up

---

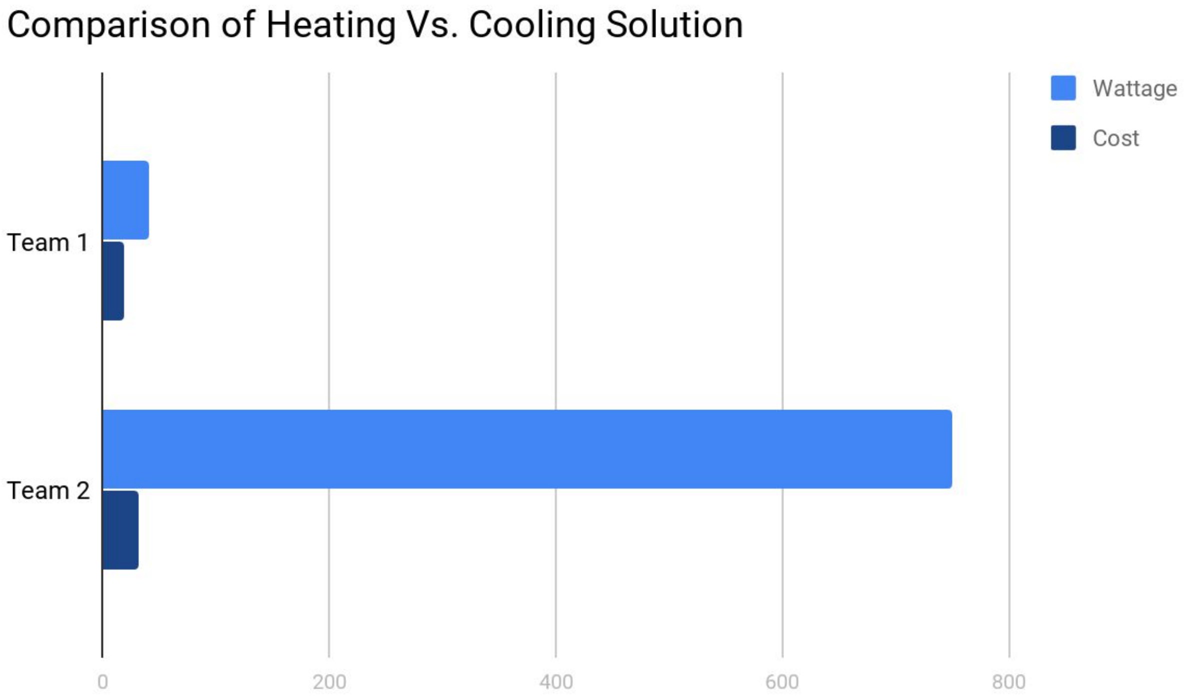
[1] Susay Inc. “Susay LED Strip Product Details.”

[2] “Shopping for Light Bulbs.”

less space in the house. This was important to us, because the space in the house was very limited, and we wanted to take up as little space as possible with our lighting solution. In the same vein, using LED strips attached to PVC piping increases modularity, allowing all the lighting in the house to be easily taken down when the house needed to move.

Analysis of Heating Vs. Cooling System in the House

Power Draw and Cost of Fan selected:
  Cost: $20 CAD
  Power Draw: 42 Watts[3]
Power Draw and Cost of Considered Heater[4]:
  Cost: $32 CAD
  Power Draw: 750-1000 Watts



Graph 2: Temperature Control Solution

After Considering both cost and current draw, as well as safety, our team decided that using a fan would be far more optimal than using a stand heater. The first determining factor for utilizing the fan system, rather than using a heater to regulate the household temperature, was the cost. Although the difference between the cost of the fan and heater was low, we did consider it in making our selection of the fan. Another determining factor was the safety of the

---

[3] Walmart Corp. "Mainstays 16" Fan Product Details."
[4] Walmart Corp. "Royal Sovereign Oscillating Fan Heater Product Details."

system. For our selection, this was twofold. Firstly, the heating system can get extremely hot, even on the exterior. As the housing system would be utilized for women with young children, we wanted to ensure the safety of these children as much as possible, and the fan allowed us to better do this. The heat generated by the system could also prove to be an issue due to the bare wood construction of the house. The other major safety consideration was the amount of power consumed by the heater. Because of the use of fairly thin, 22-gauge wire, we wanted to minimize power draw as much as possible. Although safety and cost were important considerations for selecting the fan system over a heater system was the power draw. In trying to optimize an energy-efficient house, we worked to minimize power draw wherever possible. Even on its lowest power draw settings, the heater pulls nearly 20 times the power of the selected fan. Even though we utilized a temperature sensor to control the temperature system, and the heating of the heater would shut off the system on the high-temperature trigger before the cooling system would on the low-temperature trigger, it would still utilize much more power than the fan, leading us to decide on using the fan.

Prototyping, Testing and Customer Validation

Prototype I

The initial prototype was designed to receive an improved understanding of how to construct our automation ideas into a household. It also provides information on how to interface with the arduino, relay and physical components and how they will work together to create our ideal system. The prototype was low fidelity which included a financial goal that consisted of not spending any money which meant that it could not be physical. The prototype consisted of a circuit board schematic, Figure 1, which told us that the arduino contained enough pins for all the components as well as organizational ideas so that the circuit will look simple and clean. The second part of the prototype consisted of coding for all three components. The coding, Figure 2, compiled without errors which showed us that the program created would work and could be run when the circuit was built. The stopping point to the prototype was when the schematic showed a good sense of what our system will look like, and the code runs without errors.

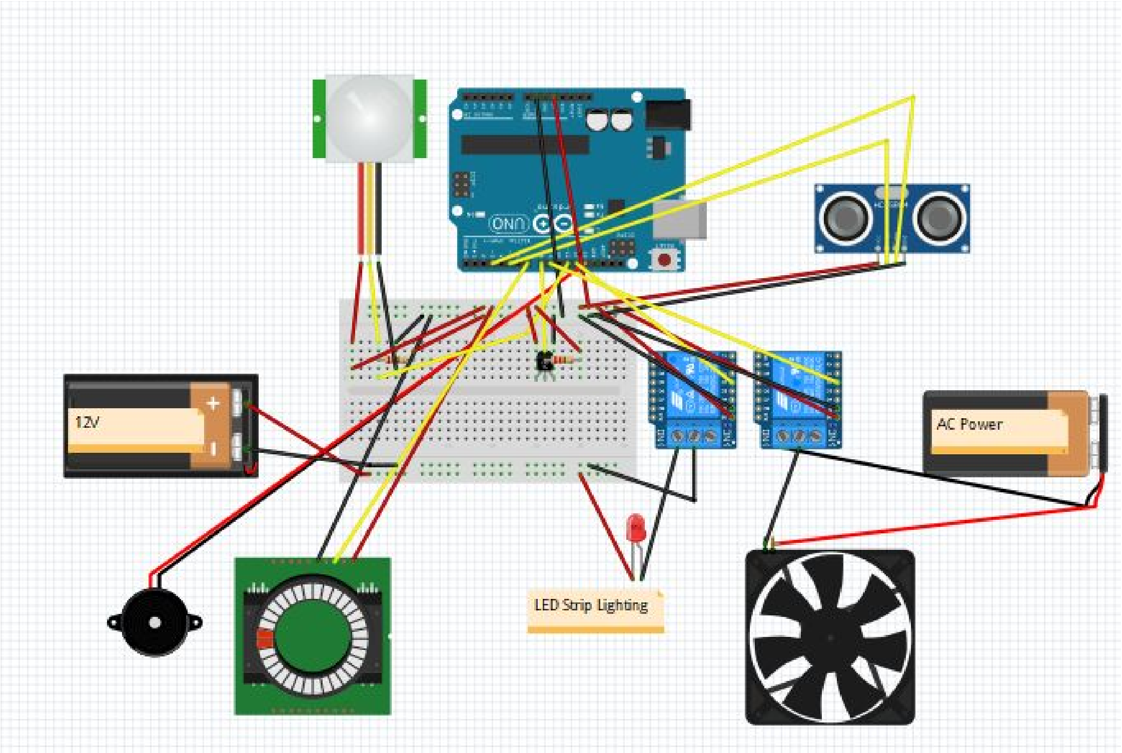
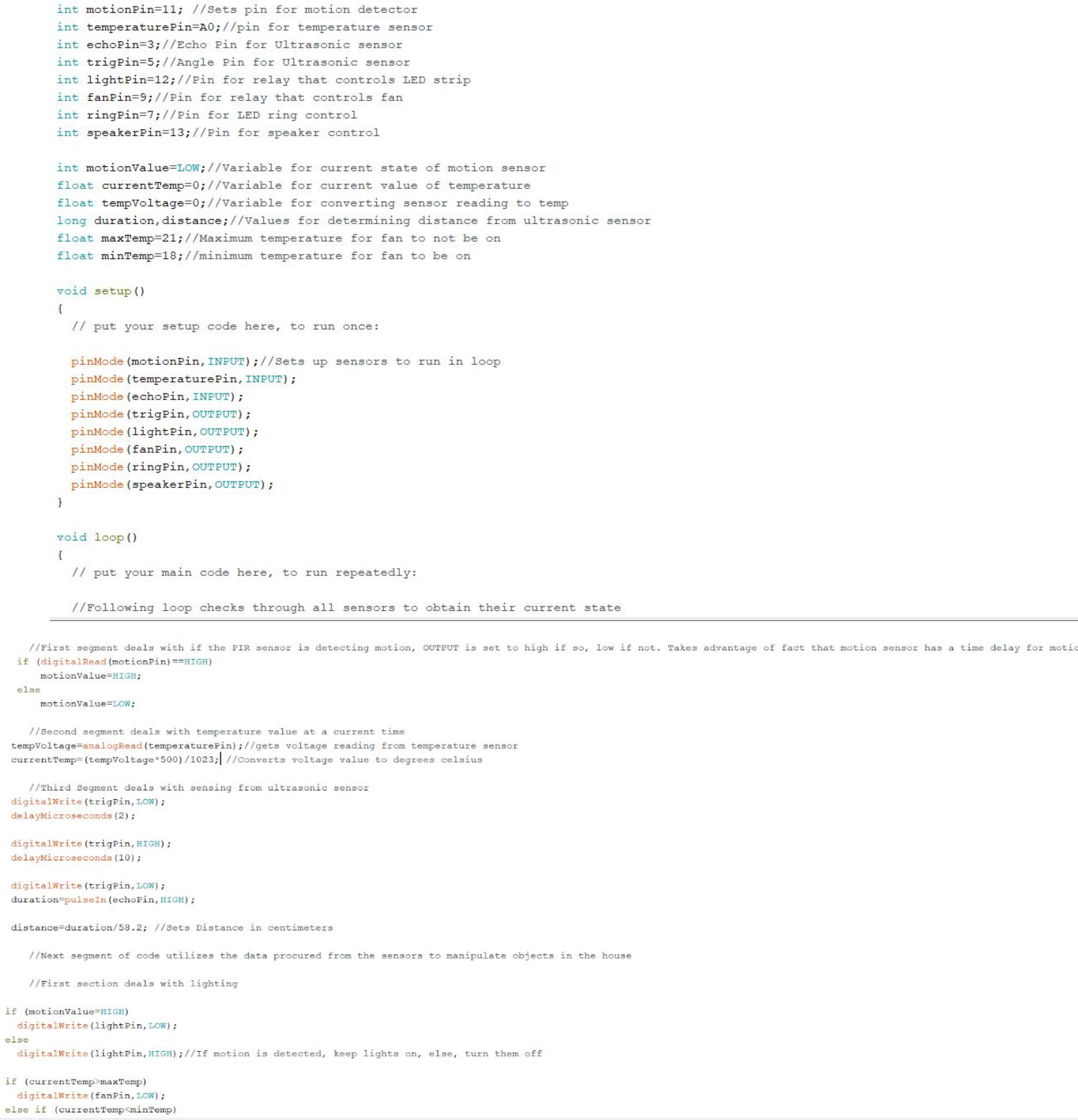


Fig 8: Arduino Schematic



```
int motionPin=11; //Sets pin for motion detector
int temperaturePin=A0;//pin for temperature sensor
int echoPin=3;//Echo Pin for Ultrasonic sensor
int trigPin=5;//Angle Pin for Ultrasonic sensor
int lightPin=12;//Pin for relay that controls LED strip
int fanPin=9;//Pin for relay that controls fan
int ringPin=7;//Pin for LED ring control
int speakerPin=13;//Pin for speaker control

int motionValue=LOW;//Variable for current state of motion sensor
float currentTemp=0;//Variable for current value of temperature
float tempVoltage=0;//Variable for converting sensor reading to temp
long duration,distance;//Values for determining distance from ultrasonic sensor
float maxTemp=21;//Maximum temperature for fan to not be on
float minTemp=18;//minimum temperature for fan to be on

void setup()
{
  // put your setup code here, to run once:

  pinMode(motionPin,INPUT);//Sets up sensors to run in loop
  pinMode(temperaturePin,INPUT);
  pinMode(echoPin,INPUT);
  pinMode(trigPin,OUTPUT);
  pinMode(lightPin,OUTPUT);
  pinMode(fanPin,OUTPUT);
  pinMode(ringPin,OUTPUT);
  pinMode(speakerPin,OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:

  //Following loop checks through all sensors to obtain their current state
```

```
   //First segment deals with if the PIR sensor is detecting motion, OUTPUT is set to high if so, low if not. Takes advantage of fact that motion sensor has a time delay for motio
  if (digitalRead(motionPin)==HIGH)
      motionValue=HIGH;
  else
      motionValue=LOW;

   //Second segment deals with temperature value at a current time
 tempVoltage=analogRead(temperaturePin);//gets voltage reading from temperature sensor
 currentTemp=(tempVoltage*500)/1023; //Converts voltage value to degrees celsius

   //Third Segment deals with sensing from ultrasonic sensor
 digitalWrite(trigPin,LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin,HIGH);
 delayMicroseconds(10);

 digitalWrite(trigPin,LOW);
 duration=pulseIn(echoPin,HIGH);

 distance=duration/58.2; //Sets Distance in centimeters

   //Next segment of code utilizes the data procured from the sensors to manipulate objects in the house

   //First section deals with lighting

if (motionValue=HIGH)
  digitalWrite(lightPin,LOW);
else
  digitalWrite(lightPin,HIGH);//If motion is detected, keep lights on, else, turn them off

if (currentTemp>maxTemp)
  digitalWrite(fanPin,LOW);
else if (currentTemp<minTemp)
```

Fig 9: Initial Arduino Code

Prototype Ⅱ

The goal of the second prototype was to produce a physical system so that the user can get a visual confirmation of our designs outcome. This prototype is a medium fidelity which provides the lighting component of our design. We chose to present lighting because it is the most important part of our design and allowed us to purchase parts that would be used in our final product which saved us money. The product was constructed from an arduino, PIR sensor, LED light stripes and a relay. It gave us an insight on how to encase our product and how it will be module while providing the client with an application that they can see that will be implemented into the final project. The prototype also brought us positive results that the coding works as well as showing us that the LED light srip will provide effiecient lighting, Figure 4. The stopping criteria was when the prototype successfully turned on when the motion detector suspects motion and turns off at the set delay.

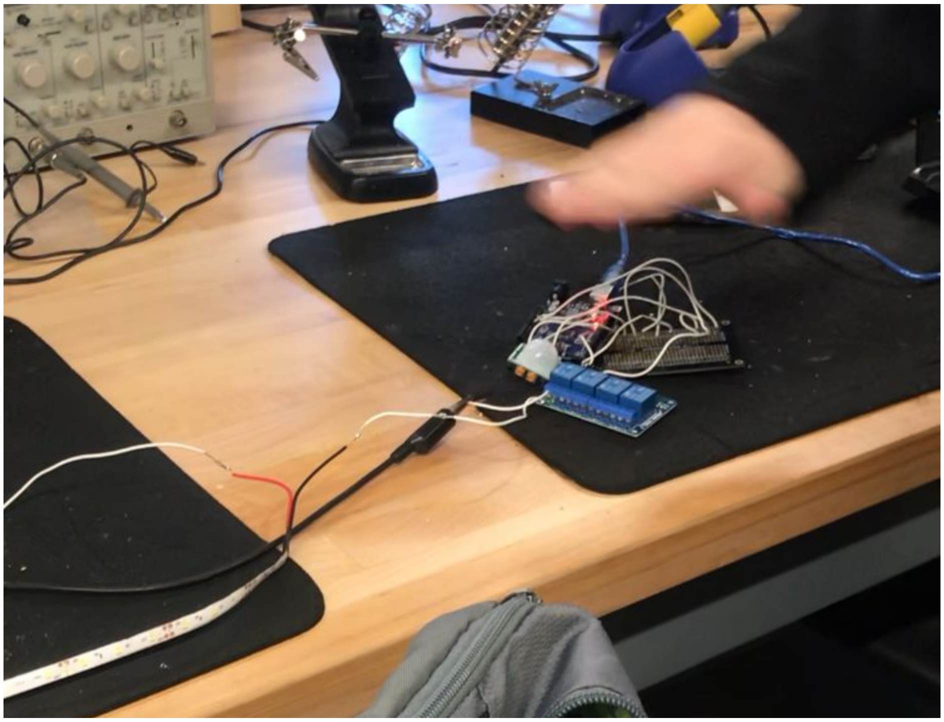


Fig 10: Lights Off

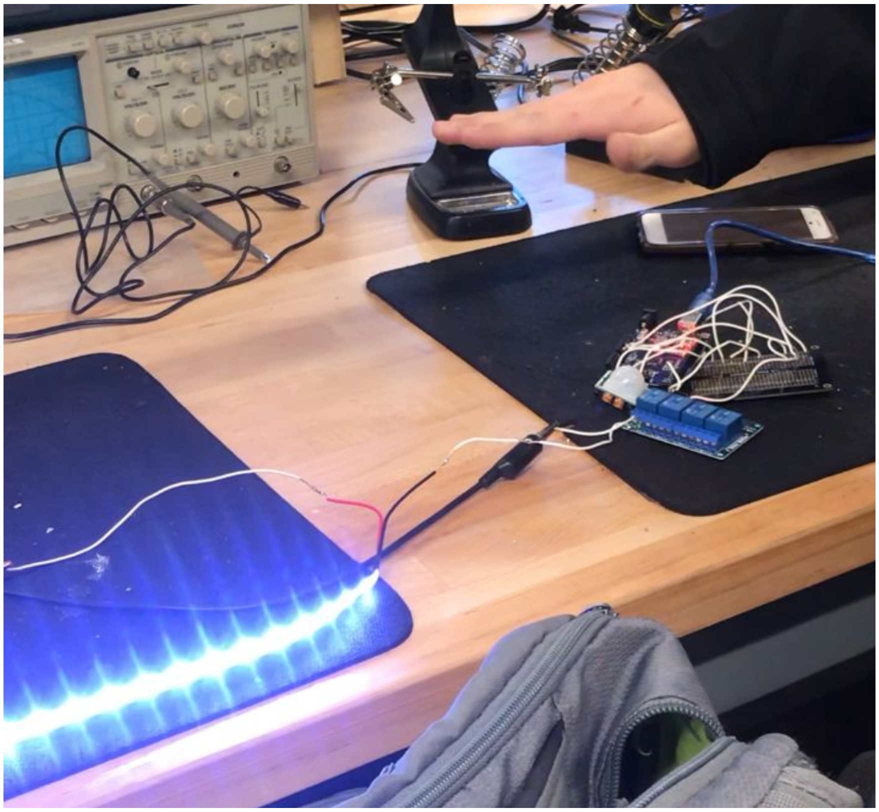


Fig 11: Lights On

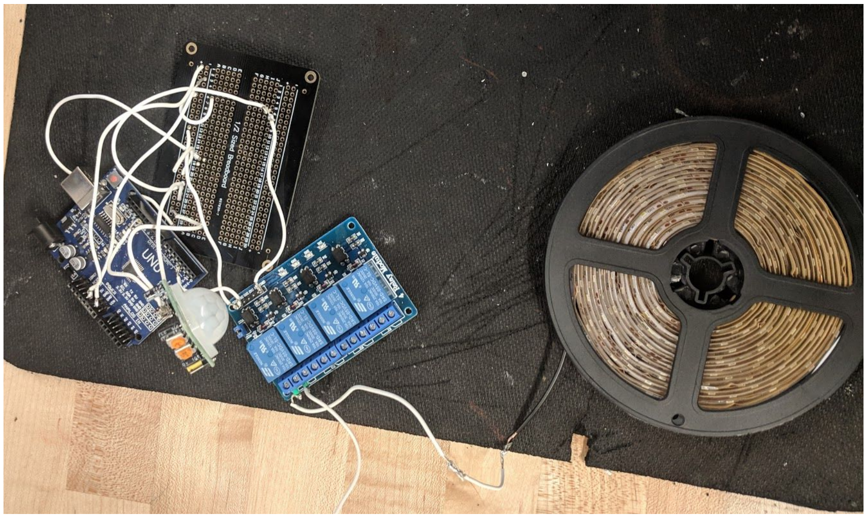


Fig 12: Circuit and Light Strips

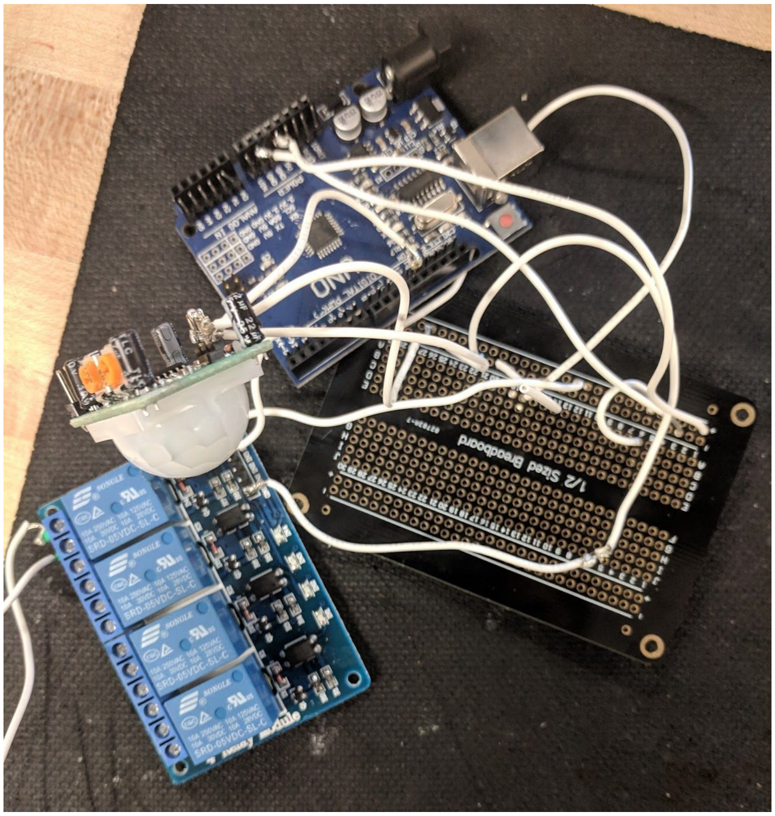


Fig 13: Circuit

Prototype Ⅲ

For the third prototype we tested the concluding subsystems and created a compact modular design that would contain our system, Figure 9. The box contained the arduino, relay, PIR sensor, LED light strip, speaker, temperature sensor and had quick connection pins that came outside of the box so that the ultrasonic sensor, fan and lights can be connected and disconnected with ease, Figure 7. The prototype had to run all subsystems with no errors and worked to our ideal liking. That is why our product was high fidelity and the stopping criteria was when each and every part of our system ran smoothly so that the client can see how a final product will interact in a household. Not only did the system need to prove its worth but also be modular so that it could be implemented into the house. The lights were attached to PVC cuts so they could be hung in the build in under a minute while the box only needed a small ledge and an outlet to be fully functional.

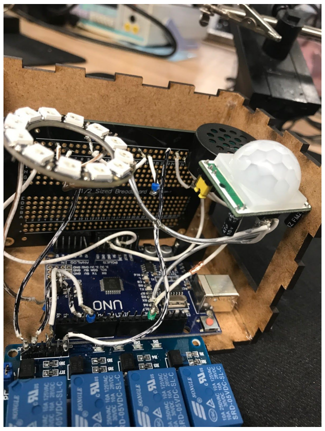


Fig 14: Inside look

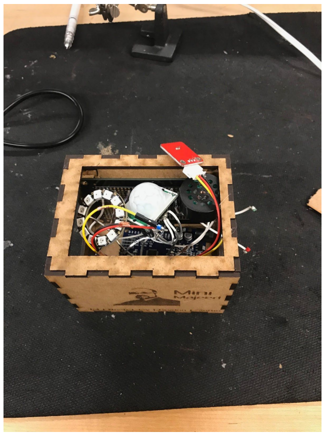


Fig 15: Birds Eye

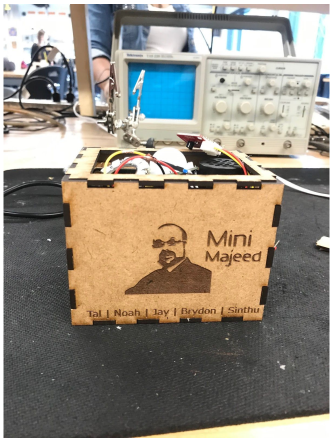


Fig 16: Final product

Final Solution

Our final solution employed the three subsystems we determined to be vital; cooling, lighting, and door monitoring, integrating them into a modular solution, with all components in one box.

The first major feature of our system is the lighting. Our final product employs the use of LED strip lighting, attached to PVC piping sections. These PVC sections rest on screws built into the modular sections of the house to allow for easy removal before the house is disassembled and moved. The LED strip used is controlled via a 4-channel relay attached to an arduino to allow for the automated control, and are powered directly from the 12-Volt battery installed by the solar team. The LED strips are controlled via a PIR motion sensor. This sensor monitors any motion within the house and toggles the lights on when motion is detected, then after an effective 10-minute delay in motion, the lights will turn off. This delay is achieved through a combination of both hardware and code. We utilized the built-in delay potentiometers of the PIR sensor to ensure the sensor continues to send a voltage signal to the arduino for 5 minutes after the last motion was detected, and enabled retriggering to allow the PIR sensor to reset this timer if more motion is detected within the duration of this delay. This allowed us to successfully automate the home lighting. We were able to conduct final testing on the house, and demonstrate that as we desired, the lights would turn off after a period of detecting no motion, and would turn back on following this period if more motion was detected.

The second feature of our final solution was the temperature control. To achieve this temperature control, we utilized a temperature sensor and a standing fan attached to a relay. In order to process the data from the temperature sensor easily, the OneWire and DallasTemperature arduino libraries were used. When the temperature sensor detected a temperature outside of the desired range (set, and adjustable in software only), the fan would either turn off or on, if the value was below the minimum or above the maximum desired value respectively. The fan attached to the same 4-channel relay as the lighting system, using a different channel, and was powered through an inverter attached to the battery provided by solar. We were able to observe the fan successfully turn off and on when desired in our final testing, and were able to easily adjust the temperature this occurs at for design day demonstration purposes.

The final feature of our design was the ultrasonic sensor-based door alert. We utilized an ultrasonic rangefinder that we embedded into the side of the house in order to monitor the distance of objects to the door. When the object is closer than a specified range (indicating something approaching the house), an auditory and visual alert are triggered within the house. In this solution, we used a small speaker embedded in our modular box that would trigger with a doorbell-like sound, as well as using a ring of RGB neopixel LEDs that perform a circular-chase sequence. We were able to successfully test this subsystem on design day by standing a certain distance away from the sensor and monitoring the inside of the house to see our alerts trigger properly.

Conclusions and Recommendations for Future Work

Through the design process applied to this process, our group learned many important lessons, both technical and non-technical. The most important non-technical skills learned from working on this project were the importance of planning and team coordination. Project planning was very important to our success as all subsystems were interconnected, therefore, planning both the subsystems individually as well as how they work together was very vital. We were able to successfully plan our project to account for this, and this was a very vital skill to learn. Coordination of the team was also a very important skill to learn. Organizing the work for each team member to take on was very vital, and learning how to coordinate the team to use the least time and achieve the greatest success was incredibly important to learn. The most important technical skills this project instilled were modular wiring and arduino code. Through our failures following the first prototype, we found the importance of modular, clean wiring to success. When we applied this newfound knowledge to the future prototypes, we were able to create a more successful project. Additionally, the arduino code skills learned were also vital. Although team members had experience with software preparation before, this project allowed us to hone our skills by utilizing several different sensor inputs at the same time. The way arduino processes delays also proved to be important to learn, and understanding this allowed us to better succeed.

In order to move forward with this process, the most important thing to change is the modularity of the system. By re-cutting our component box in a larger size and re-doing the wiring in a slightly more organized way, the product could be easier to use, more visually appealing, and more modular. Additionally, we could use a real-time operating system library with the arduino code to better process the sensor inputs with delays, improving usability. Finally, we would add a way for the user to manually change the temperature system’s trigger conditions, allowing them to adjust the temperature range in hardware rather than relying on code to do so.

Bibliography

1. Susay Inc. “Susay LED Strip Product Details.” *Amazon: Susay Waterproof LED Light Strip 12V with 300 SMD LED Cool White*, Amazon, www.amazon.ca/gp/product/B005EHHLD8/ref=ppx_yo_dt_b_asin_title_o06_s00?ie=UTF8&psc=1.

List of data specifications for the utilized LED lighting strip

2. “Shopping for Light Bulbs.” *Consumer Information*, United States Federal Trade Commission, 13 Mar. 2018, www.consumer.ftc.gov/articles/0164-shopping-light-bulbs.

3. Walmart Corp. “Mainstays 16” Fan Product Details.” *Mainstays 16" 3-Speed Oscillating Pedestal Fan, Black, FS40-8MB*, Walmart, www.walmart.com/ip/Mainstays-16-3-Speed-Oscillating-Pedestal-Fan-Black-FS40-8MB/55347312.

Cost of fan purchased was different to price displayed on site. Identical fan as Canadian Walmart site did not list power draw.

4. Walmart Corp. “Royal Sovereign Oscillating Fan Heater Product Details.” *Royal Sovereign Oscillating Fan Heater*, Walmart, www.walmart.ca/en/ip/royal-sovereign-oscillating-fan-heater/6000195409149.

Product Specs for considered fan

APPENDICES

APPENDIX Ⅰ: User Manual

Product Features:

The product consists of three connected subsystems. Each subsystem performs a different task, combined, they control the overall automation of the house.

Lighting: The product features a lighting system that allows the user to save energy by employing motion sensing lights. The system utilizes a PIR motion sensor to detect motion in the house and only turns on the attached LED strips when this motion is detected, with a built-in delay.

Temperature: The product's temperature control system automates an attached household fan, allowing users to save energy by automatically turning the fan on or off when a certain temperature is reached.

Door Alert: The product's door alert subsystem monitors the door of the house the system is installed in, allowing the user to be alerted by both a sound and light indication when the door is being approached.

Maintenance:

Maintenance of this product is mild, and only needs to occur once the house is to be moved. To maintain the system in this case, detach all wires from the outside of the box, and remove the PVC piping from the house, starting with the piping placed to the left of the door. Once the lighting has been taken down, unplug the lights from the wires sticking out of the box, and unplug the battery from these as well. In order to ensure the system is properly maintained, visually examine the wires within the box and ensure that no contact between pins is occuring. Finally, ensure the lighting is properly attached to the PVC piping. Perform these checks both before and moving the house to its new location to ensure proper operation. No other maintenance is necessary in daily operation.

Product Functions:

The product serves to automate a small, modular home. The primary functions of the product are to control the lighting and temperature within the home, as well as to alert the occupants when the door is being approached. The product automatically turns on and off the lights when motion is detected or when this motion ceases, and the product regulates the temperature inside the house to a desired setting. FInally, the product has an audible ring and light circle effect when the door is being approached from the outside to alert the occupant.

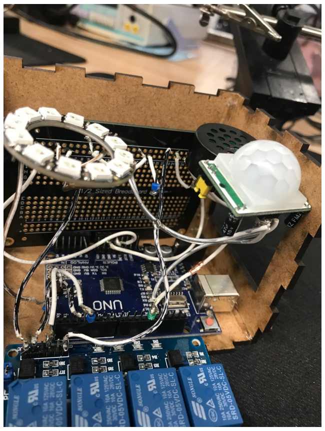


Fig 17: Exploded Box View

Parts List

Part Name	**Quantity**	**Function**
Arduino UNO	1	Controls automation of the house
4-channel, 5V Relay	1	Acts as a switch for the lights and fan
PIR Sensor	1	Detects motion in the house
NeoPixel 12-Pixel Ring	1	Visual alert for door alert
PCB-Mount Speaker	1	Auditory alert for door alert
Solderable Breadboard	1	Allows powering of all components
Ultrasonic Sensor	1	Monitors distance of objects from the door
Temperature Sensor	1	Detects Temperature for automatic Control
MDF Housing	1	Places all components in a modular box
Wires	Assorted	Allows for easy connection of all components
LED Strip Lighting: 5m	1	Provides lighting in the house
16” Standing Fan	1	Provides cooling for house
PVC Pipe Pieces	4	Modular mounting for LED strips

Table 3: Parts List

Installation Instructions

1. Ensuring LED’s stay adhered to their PVC piping, place them on the screws attached to the house, beginning from the right of the door, and beginning with the LED section with wires attached
2. Place component box on the shelf below the window of the house
3. Attach long black wire from the right side of the box to the negative side of the battery, attach red wire from the LED strip.
4. Attach black wire from the LED strip to the labeled wire out of the left side of the box.
5. Attach plug-side of the fan into the inverter
6. Attach labeled wire from the fan plug to the labeled wire on the right side of the box.
7. Attach the un-labeled wire from the fan plug to the red-wire attached to the fan body
8. Attach the black wire attached to the fan body to the green wire on the left side of the box
9. Place Ultrasonic rangefinder into the slot cut in the house wall to the left of the door
10. Attach colored wires on the rangefinder to their respective colored pin headers on the right side of the box
11. Plug a smartphone charge adapter (not included) into the inverter, and using the USB-B to USB-A cable included, plug the arduino into this adapter
12. Turn on the power to the inverter.

Safety Guidelines:

Although the system does automate the home, it should not be used when the user will be leaving for an extended time. If the user plans to be away for longer than one day, the inverter power should be shut off and the Leads to the LED strip lighting attached to the battery should be unscrewed and removed. This ensures that the system will not short out while the user is away, mitigating an electrical fire

Troubleshooting:

Lights are not staying on long enough/are staying on too long:

Open the box containing the system and turn adjust the left dial on the motion sensor (domed sensor) until the delay is set to a desired point. If more adjustment is desired, modification to the delay at the end of the final loop in the included arduino code can be adjusted.

Lights are not triggering enough/are triggering to often:

To adjust the sensitivity of the motion sensor, open the box, and turn the rightmost dial on the motion sensor (domed sensor) to the desired point.

Lights are not turning on:

Ensure lights are plugged into the correct wires on the left-side of the box, and that the power and ground wires are properly assigned. Additionally, ensure the connection to the battery is properly established and that the battery still has power left to power the system. Finally, examine the third channel of the relay and ensure the connections in the relay are entact.

Fan is not turning on:

Ensure fan is plugged into the correct wires on the left-side of the box, and that the power and ground wires are properly assigned. Additionally, ensure that the fan plug is properly plugged into the inverter and that the battery still has power left to power the system. Finally, examine the first channel of the relay and ensure the connections in the relay are entact.

Ultrasonic Rangefinder is not working:

Ensure all wires are properly connected to both the arduino pins and the rangefinder. Additionally, ensure the wire colors properly match the pin headers on the right side of the box properly match the wire colors on the rangefinder.

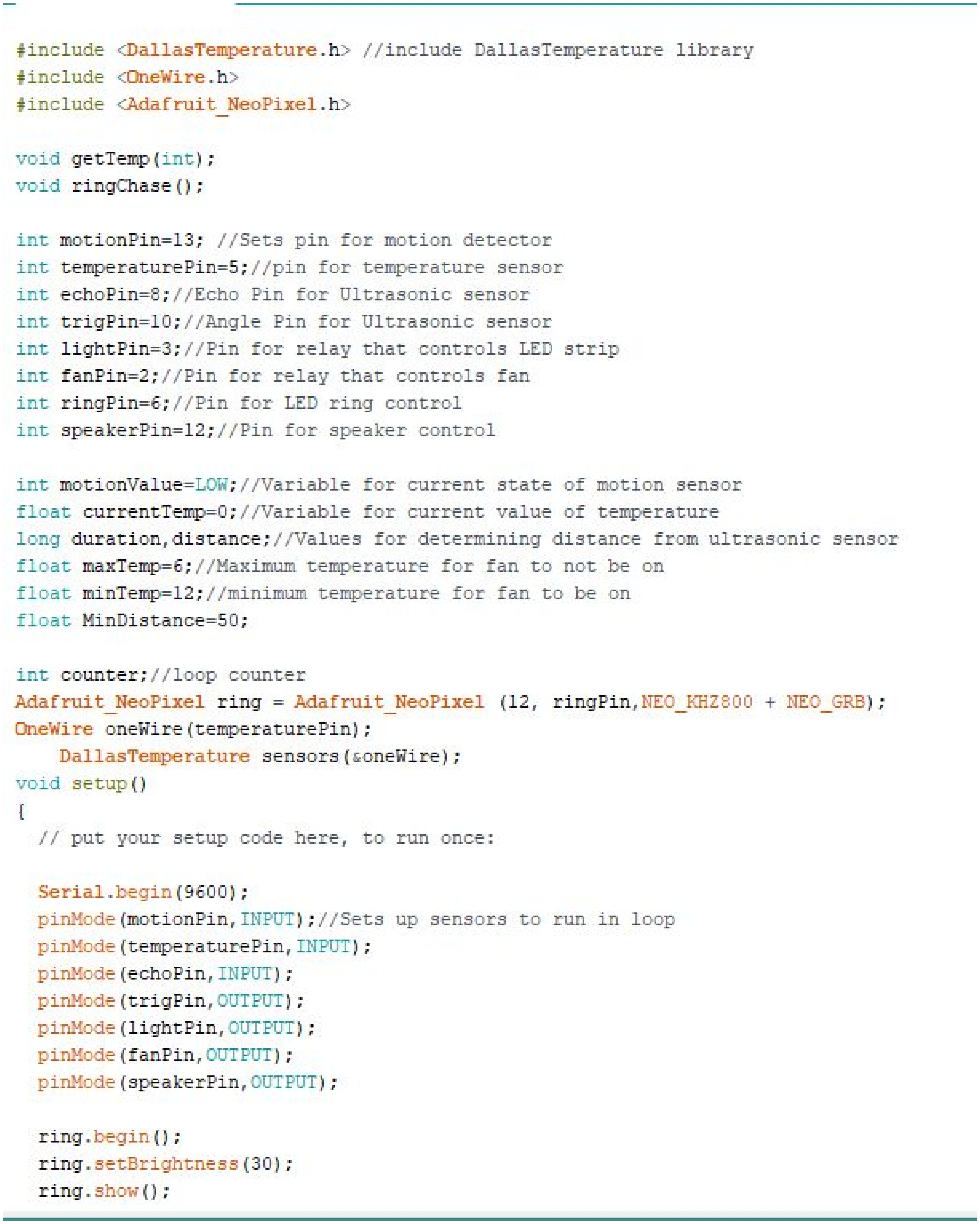
Ultrasonic Rangefinder does not trigger at a far enough range:

Using the included arduino code, modify the MinDistance variable to the desired setting in centimetres, then upload this code to the arduino.

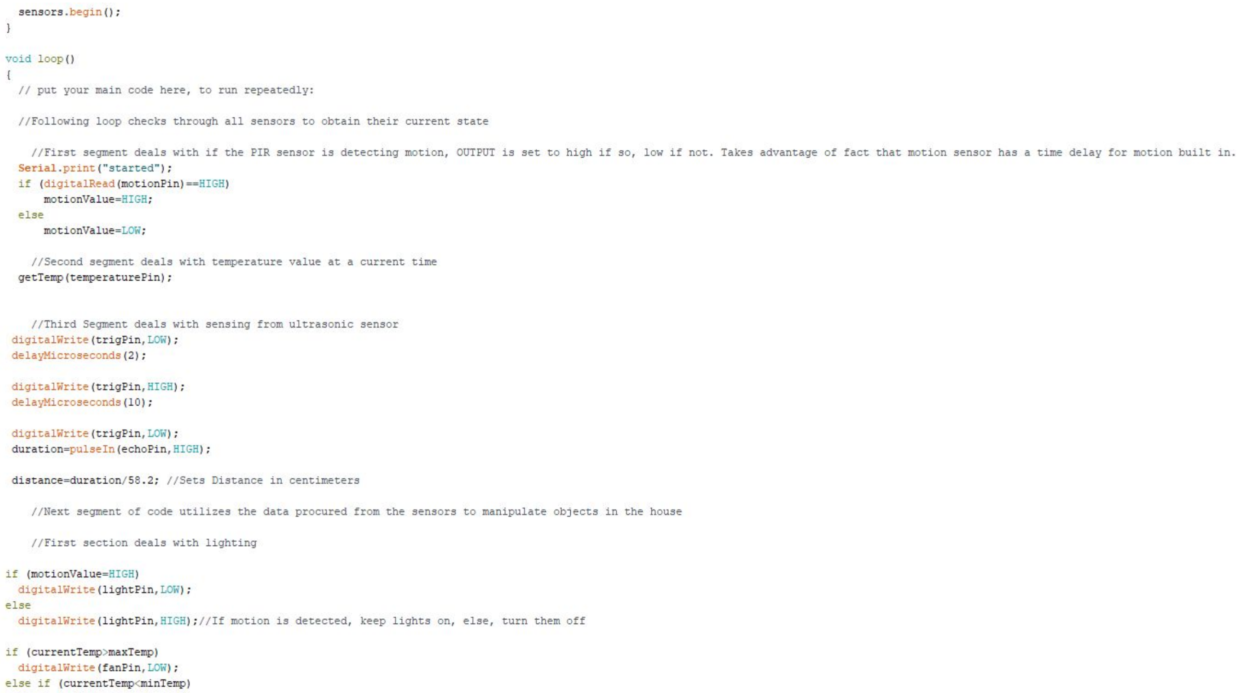
Fan does not keep house at a comfortable temperature:

Using the included arduino code, modify the MinTemp and MaxTemp values to set them to the desired temperature (in degrees celsius), then upload this code to the arduino.

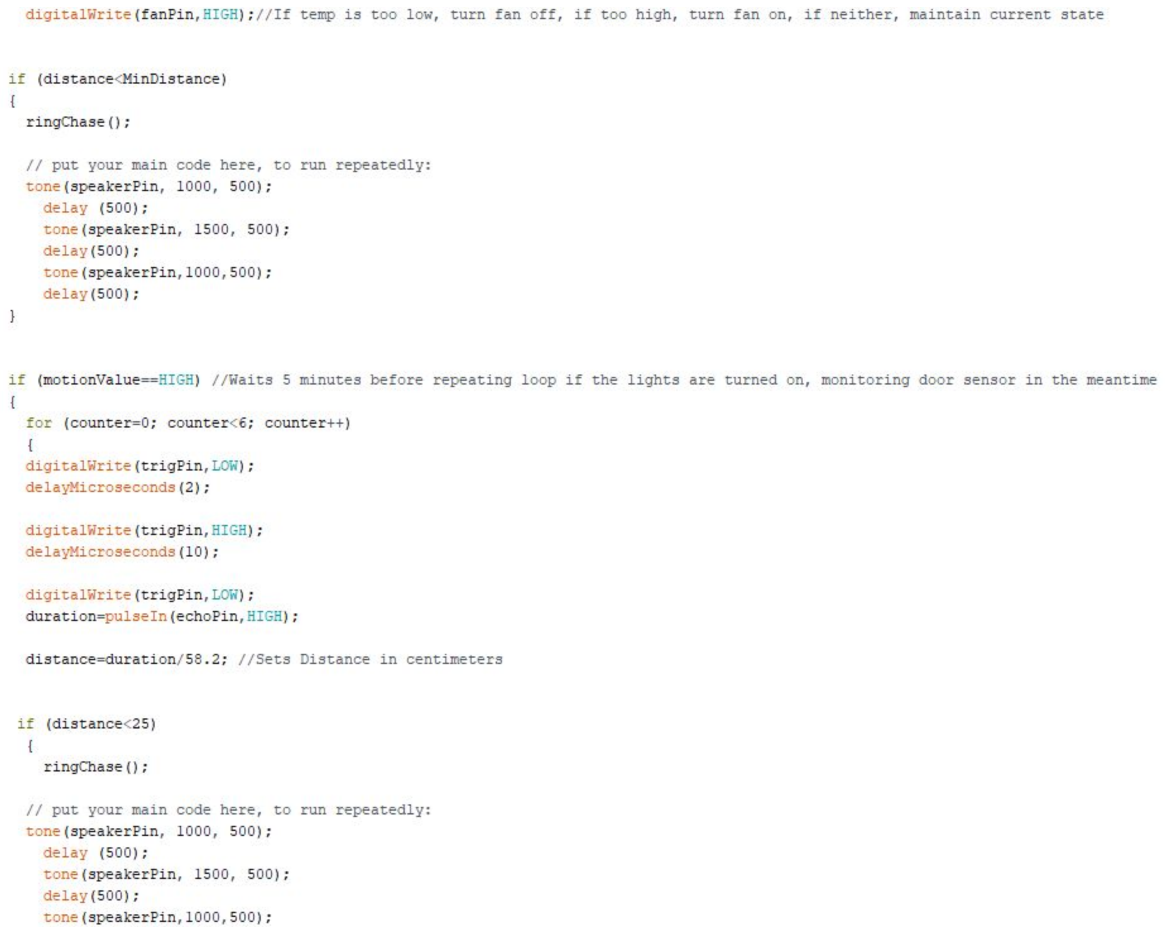
APPENDIX Ⅱ: Design Files



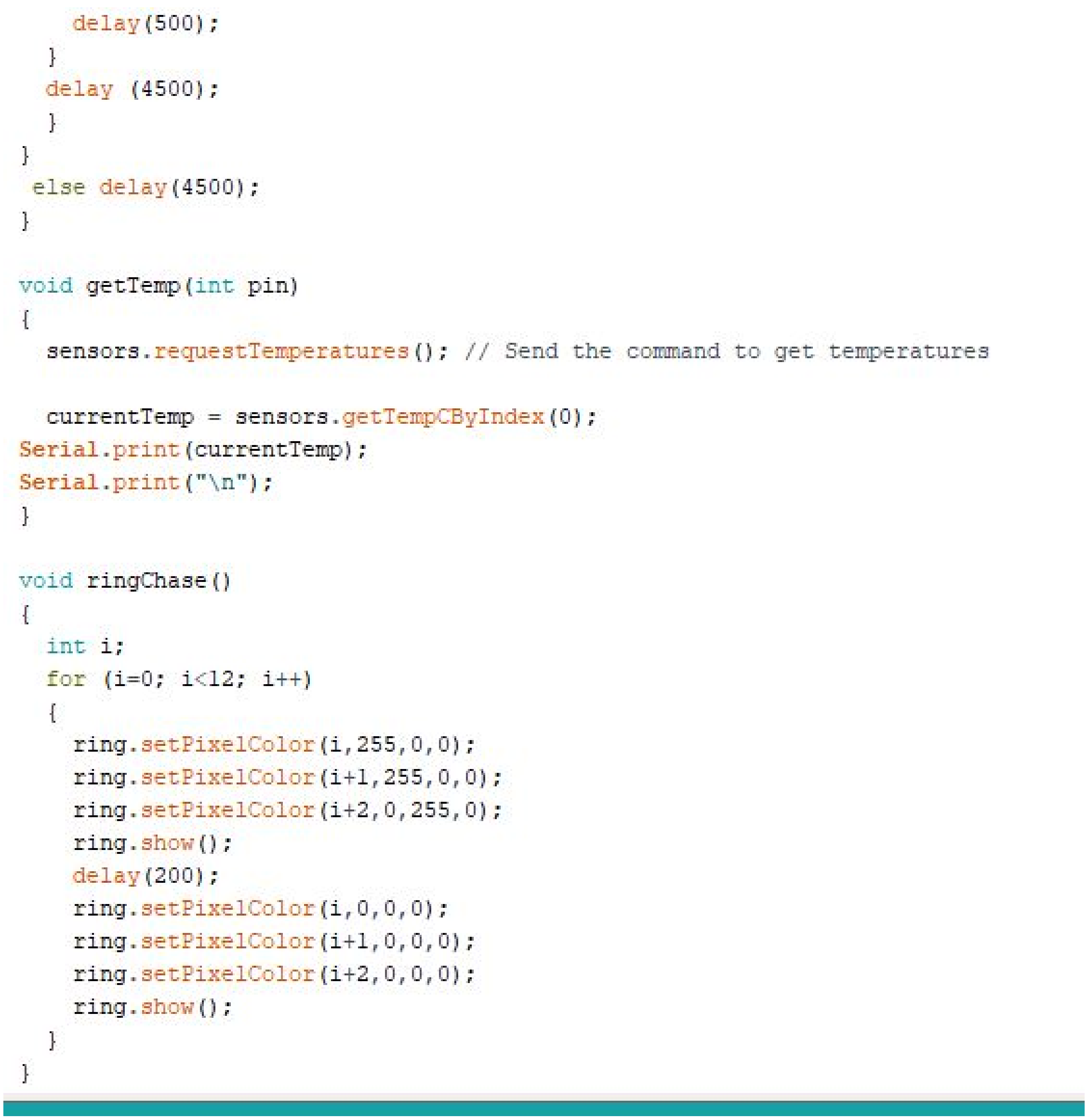
```
#include <DallasTemperature.h> //include DallasTemperature library
#include <OneWire.h>
#include <Adafruit_NeoPixel.h>

void getTemp(int);
void ringChase();

int motionPin=13; //Sets pin for motion detector
int temperaturePin=5;//pin for temperature sensor
int echoPin=8;//Echo Pin for Ultrasonic sensor
int trigPin=10;//Angle Pin for Ultrasonic sensor
int lightPin=3;//Pin for relay that controls LED strip
int fanPin=2;//Pin for relay that controls fan
int ringPin=6;//Pin for LED ring control
int speakerPin=12;//Pin for speaker control

int motionValue=LOW;//Variable for current state of motion sensor
float currentTemp=0;//Variable for current value of temperature
long duration,distance;//Values for determining distance from ultrasonic sensor
float maxTemp=6;//Maximum temperature for fan to not be on
float minTemp=12;//minimum temperature for fan to be on
float MinDistance=50;

int counter;//loop counter
Adafruit_NeoPixel ring = Adafruit_NeoPixel (12, ringPin,NEO_KHZ800 + NEO_GRB);
OneWire oneWire(temperaturePin);
    DallasTemperature sensors(&oneWire);
void setup()
{
  // put your setup code here, to run once:

  Serial.begin(9600);
  pinMode(motionPin,INPUT);//Sets up sensors to run in loop
  pinMode(temperaturePin,INPUT);
  pinMode(echoPin,INPUT);
  pinMode(trigPin,OUTPUT);
  pinMode(lightPin,OUTPUT);
  pinMode(fanPin,OUTPUT);
  pinMode(speakerPin,OUTPUT);

  ring.begin();
  ring.setBrightness(30);
  ring.show();
```



```
  sensors.begin();
}

void loop()
{
  // put your main code here, to run repeatedly:

  //Following loop checks through all sensors to obtain their current state

    //First segment deals with if the PIR sensor is detecting motion, OUTPUT is set to high if so, low if not. Takes advantage of fact that motion sensor has a time delay for motion built in.
  Serial.print("started");
  if (digitalRead(motionPin)==HIGH)
      motionValue=HIGH;
  else
      motionValue=LOW;

    //Second segment deals with temperature value at a current time
  getTemp(temperaturePin);


    //Third Segment deals with sensing from ultrasonic sensor
 digitalWrite(trigPin,LOW);
 delayMicroseconds(2);

 digitalWrite(trigPin,HIGH);
 delayMicroseconds(10);

 digitalWrite(trigPin,LOW);
 duration=pulseIn(echoPin,HIGH);

 distance=duration/58.2; //Sets Distance in centimeters

    //Next segment of code utilizes the data procured from the sensors to manipulate objects in the house

    //First section deals with lighting

if (motionValue=HIGH)
  digitalWrite(lightPin,LOW);
else
  digitalWrite(lightPin,HIGH);//If motion is detected, keep lights on, else, turn them off

if (currentTemp>maxTemp)
  digitalWrite(fanPin,LOW);
else if (currentTemp<minTemp)
```



```
  digitalWrite(fanPin,HIGH);//If temp is too low, turn fan off, if too high, turn fan on, if neither, maintain current state


if (distance<MinDistance)
{
  ringChase();

  // put your main code here, to run repeatedly:
  tone(speakerPin, 1000, 500);
    delay (500);
    tone(speakerPin, 1500, 500);
    delay(500);
    tone(speakerPin,1000,500);
    delay(500);
}


if (motionValue==HIGH) //Waits 5 minutes before repeating loop if the lights are turned on, monitoring door sensor in the meantime
{
  for (counter=0; counter<6; counter++)
  {
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);

  digitalWrite(trigPin,LOW);
  duration=pulseIn(echoPin,HIGH);

  distance=duration/58.2; //Sets Distance in centimeters


 if (distance<25)
  {
    ringChase();

  // put your main code here, to run repeatedly:
  tone(speakerPin, 1000, 500);
    delay (500);
    tone(speakerPin, 1500, 500);
    delay(500);
    tone(speakerPin,1000,500);
```



```
    delay(500);
  }
  delay (4500);
  }
}
 else delay(4500);
}

void getTemp(int pin)
{
  sensors.requestTemperatures(); // Send the command to get temperatures

  currentTemp = sensors.getTempCByIndex(0);
Serial.print(currentTemp);
Serial.print("\n");
}

void ringChase()
{
  int i;
  for (i=0; i<12; i++)
  {
    ring.setPixelColor(i,255,0,0);
    ring.setPixelColor(i+1,255,0,0);
    ring.setPixelColor(i+2,0,255,0);
    ring.show();
    delay(200);
    ring.setPixelColor(i,0,0,0);
    ring.setPixelColor(i+1,0,0,0);
    ring.setPixelColor(i+2,0,0,0);
    ring.show();
  }
}
```

Fig. 18 Final Arduino Code

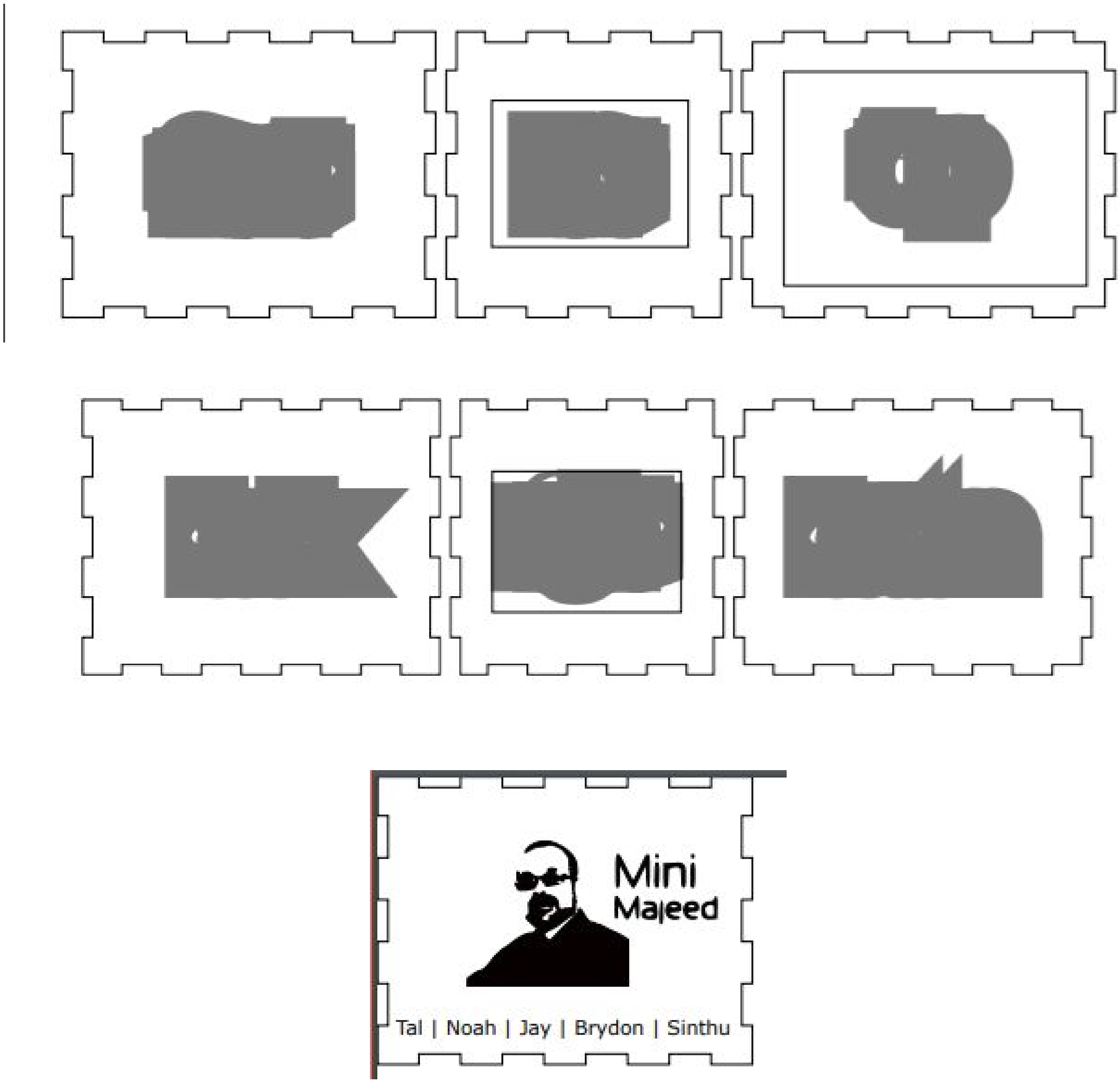


Fig. 19 - File used for Box Cutting