THOMPSON RIVERS UNIVERSITY

ENGR 2000 - Engineering Design III

Smart Home

Luka Aitken (T00663672)

Almat Bolatbekov (T00667332)

Brandon Breithaupt (T00591195)

Toma Aitken (T00663680)

April 12, 2022

Table of Contents

List of Figures

List of Tables

1 Introduction

Smart home systems are becoming more common as the world progresses into the future. There are around 170 million individuals who use smart home systems in the world, with more people installing these systems each year to help with everyday living. These systems can benefit house residents by providing security and alerts of hazards throughout the house. Some advantages of owning a smart house system are being able to perform everyday tasks without as much manual effort such as turning on lights, ventilation, and security. As more people have smart home systems, it can also reduce the risks of inside and outside hazards. Without a smart home system to keep the house safe, the house is vulnerable to break-ins, floods, and other kinds of unpleasant events that can lead to uncontrolled situations. These events can cause damage to the house and its residents depending on the scenario, which can lead to paying for those damages that can cost lots of money to repair. With this, having some sort of system to keep the house in check and safe from break-ins and floods is definitely most needed. That is where our smart home system comes in [1].

To begin with our smart home, we came up with many ideas on how to create a reliable system that requires minimal human effort, while following the project requirements and constraints. Our goal was to come up with a design that can keep the home secured and safe from dangers such as intruders and floods, but also accessible to household members for everyday tasks inside and outside the house at an affordable price. We designed the code using the program MBED to create and test each component for the best possible smart features for the house. After choosing our final design, we built a small-scale version of a house, where we then did all the necessary tests to ensure that they fit the project requirements and constraints.

Through this report, we will discuss the problems we faced when designing and building the smart features of the house and the house itself. We will also discuss the other solutions that we didn't use but took some inspiration from to come up with our final design. Additionally, the environmental, societal, safety, economic considerations as well as sustainability are some other key factors that will be discussed throughout this report.

2 Design Problem

2.1 Problem Definition

According to the S. Stasha home burglary is a fairly common crime that happens more often than many people would think, as the latest burglary statistics reveal. Statistics show that there are almost three burglaries every minute in the U.S. and only 17% of US citizens have security systems in their homes. Smart house systems provide control of your security systems, automate your lights, trigger an alarm to ring if there is an unwanted motion or entry and get immediate alerts if doors or windows open unexpectedly. It is clearly seen that smart houses can prevent burglaries and unwanted entering. One of the most obvious energy-wasting habits is leaving the lights on. Smart houses system helps to monitor your lightning from your smartphone which can save your electricity consumption and money. There are also other problems like flooding and this problem also can be fixed by smart house systems that use water level sensors [2].

2.2 Design Requirements

2.2.1 Functions

- The smart home should be able to detect movement outside the house and turn the lights on.
- The smart home should be able to read the humidity level in the garage and turn on the vents to reduce the humidity.
- The smart home should be able to read when doors are open and the security system is on to alert the homeowner of an intruder.
- The smart home should be able to alert the homeowner of any change in the system through the Bluetooth app and be able to take commands via the Bluetooth app as well.
- The smart home should be able to open the garage remotely and be time sensitive so that the contents of the garage are not at risk.

Performance	8
Safety	8
Simplicity	7
Maintenance	6
Affordability	7

Table 2.2.1: Importance Chart.

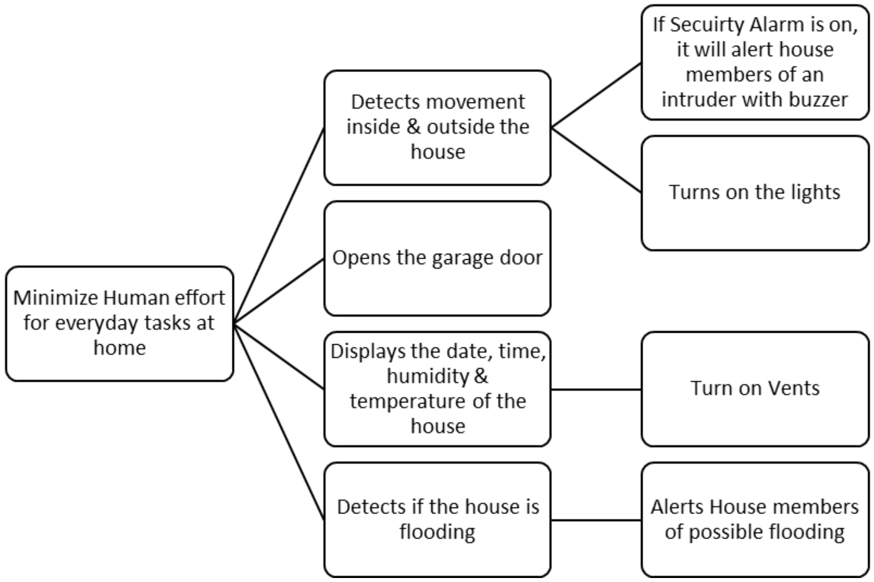


Figure 2.2.1: Function Tree.

2.2.2 Objectives

- Elimination of human manual effort.
- Cost-effective system.
- Prevent strangers from breaking inside.
- Warn household members of flooding and fire.

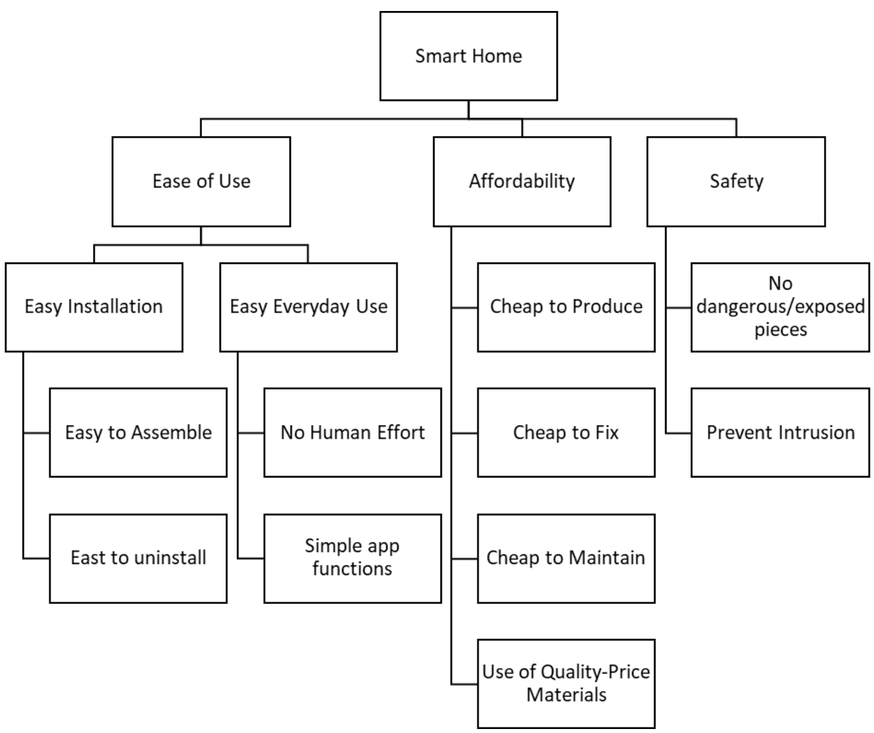


Figure 2.2.2: Objective Tree.

2.2.3 Constraints

- **Cost:** The cost should be less than **$40.**
- **Required Feature:**
    - Automatic lighting inside and outside the house.
    - Automatic ventilation
    - Security alarm
    - Smart garage
    - Automatic **power off** in the event of **flooding** inside the house.
- Environmental constraints:
    - The largest environmental constraint is the consumption of energy that is needed to be used to run our system. Here in BC our energy is mostly derived from hydro power plants which are dams built on water systems that use potential energy to drive and spin turbines to create electricity within a power grid. One way we can reduce this is through the use of solar panels which will create a net zero energy usage for our system. Another very large problem for the environment would be the material used for our system. For our prototype we will use cardboard and other recyclable items in order to reduce our impact on the environment. On a large scale our system will be designed with reusable, recyclable materials in mind in order to reduce our footprint on the environment.
- Economic constraints:
    - The largest economic constraints for our prototype is that we have a budget of 40$ to build and test our prototype. On a large scale some of the more pressing economic constraints will be the price of the system and how much it will cost to build, maintain, and run this system for the duration of its life. We can reduce some of these costs through the use of a solar panel to have a net zero system in place with the ability to run all of our systems and produce an equal amount of the energy used through this.

3 Solution

3.1 Solution 1

For our first solution, our approach was to make a design that features multiple rooms using components for each room. Each room would have an automatic light system that uses motion sensors with all the cabling underneath the house that can be accessed by a removable floor. The removable floor is used to access the electronics easily and helps with organization. For the garage, it uses a string that is attached to a servo motor and the garage door to pull the garage door inwards to open for access inside. To close the door, the servo motor will release the string, letting the door fall down to its previous position. To help the door to close, there would be small weights attached at the bottom to weigh down the door to prevent the door from only closing halfway. The format of the house is not squared to make the house more realistic and ideal. With the smart app to control each component, it can turn on and off lights inside and outside, open and close the garage, and can set the temperature by controlling the fan speed which is displayed on an LCD display alongside with the time of the day. Additionally, there can be an alarm set from the app that uses motion sensors inside the main room and garage to detect movement and alert the residents.

Advantages:

- **Organized** cabling.
- More **realistics** with **multiple** rooms.

Disadvantages:

- More **expensive** with more components.
- **Too Complex** to build.

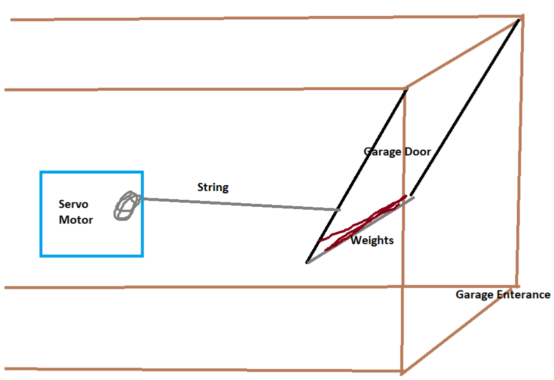


Figure 3.1.1: Garage Door Mechanic for First Solution.

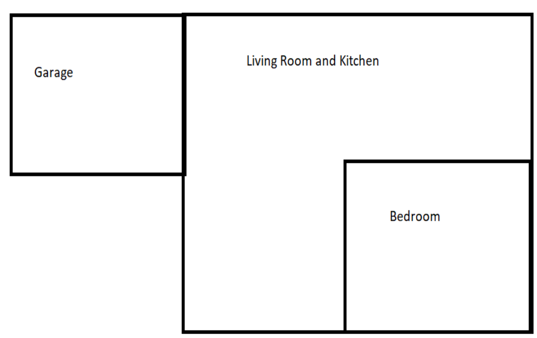


Figure 3.1.2: Smart House Layout of first Solution.

3.2 Solution 2

For our second solution, our approach was to make a simpler design than our first solution with a much simpler solution. The house only has two rooms which are the garage and the living space, allowing it to implement fewer components making the smart home system much cheaper. The wiring for the components are hidden between the walls allowing easy access to change or fix any wires if needed. The garage uses the motor itself with a small dowel attached to the spinning piece of the servo motor that will push the garage door upwards towards the outside of the house. For the smart app, it can turn the lights on and off both inside and outside, can open and close the garage door, and can set the temperature by using the fan to control its speed and be displayed on an LCD display that also displays the time of the day. Also, the app can set up an alarm during any time of the day that uses motion sensors in the house to detect any movement and alert the residents if anything is detected.

Advantages:

- **Cost Effective**.
- **Organized** Cabling.
- **Simplicity**.

Disadvantages:

- Less **space** for interior.
- **Hard** to access **wiring** as everything is hidden together.

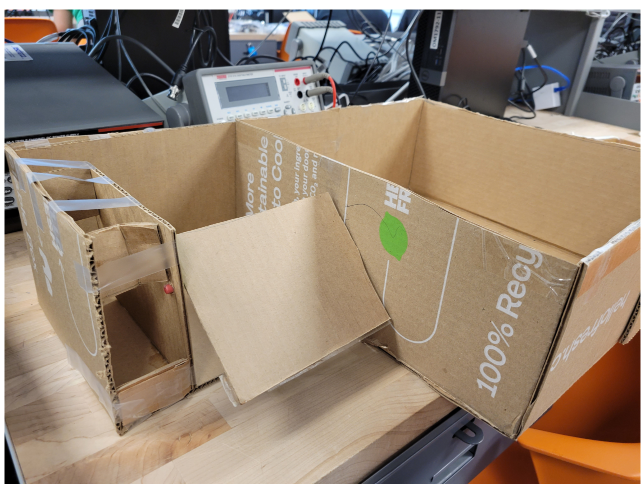


Figure 3.2.1: Prototype of Solution 2.

3.3 Final Solution

For our final solution, we came up with a design where all the wiring is throughout the house and the framing of the house uses only two rooms for the garage and the living space. The smart app uses buttons to control the lights inside and outside, garage door, security alarm, and uses a slider to control the temperature inside the house.

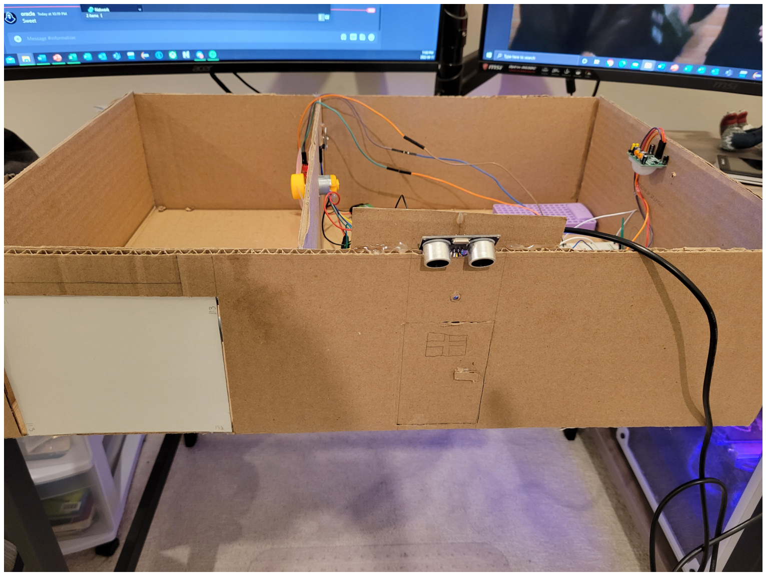


Figure 3.3.1: Front View of Final Design.

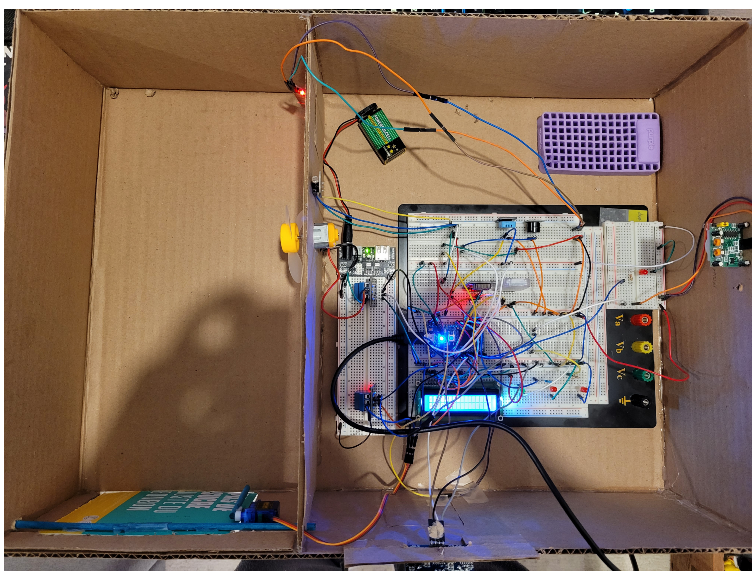


Figure 3.3.2: Top View of Final Design.

The reason why we chose this solution over the other two solutions is that this design offered a more reliable system that is easy to manage. For both of the previous designs, they used switches to control the functions of each component on the smart app, which can make it confusing for the user to determine what each switch meant as it did not tell if it was on or off.

The final design uses buttons to control each smart function for the house and makes it easier to understand what each button did, as it tells the user what each button does when pressed by changing the colour of the button. Our first solution provided a reliable system that offered a system that can have multiple components for each room inside the house. However, this design would cost lots of money due to all the components and extra microcontrollers needed to allow the smart home system to work. With the final design, it only uses one microcontroller to control all the components inside the house, making it much cheaper to afford. Also with the garage motor, it used a string to pull up the garage door inwards, but was hard to implement as the string could break and require maintenance. Our second solution followed a similar design to our final design, but hid all the wiring inside the walls to reduce the wiring across the house itself. However, by doing this, it made it much harder to manage or fix any components or wiring that needed to be changed due to the crowdedness of all the components inside the walls.

		Solutions					
		Solution 1		Solution 2		Final Solution	
Criteria	**Weight**	**Score**	**Partial Score**	**Score**	**Partial Score**	**Score**	**Partial Score**
Use of Standard Parts	0.30	6/10	0.180	6/10	0.180	7/10	0.210
Safe	0.25	6/10	0.150	7/10	0.175	7/10	0.175
Simplicity	0.20	7/10	0.140	8/10	0.160	8/10	0.160
Management	0.25	7/10	0.175	6/10	0.150	8/10	0.200
Sum	1.00		0.645		0.665		0.745

Table 3.3.1: Decision Matrix Chart for the Considered Alternatives.

3.3.1 Components

The MBED microcontroller is the main hub of the system, where it allows code to run through the board into all the other components [3]. The ultrasonics sensor located outside detects and determines the distance between the front door and an object [4]. The PIR sensor located inside the house detects any movement inside the house [5]. The servo motor rotates its dowel to the right to open and hold the garage door [6]. The fan motor is used to ventilate the house [7]. The outside LED represents the outside lights and shows that there is an object or person near the front door. The inside LED represents the inside lights and shows that there is movement inside the house. The green LED represents the AC turning on and the red LED represents the heater turning on [8]. The real-time module takes the time from the smart app system, stores it, and displays it through the LCD display [9]. One of the two relays is going to be used to build a breaker box so that we can shut off the power to everything in the case of an emergency. The other relay is going to be used as an electrical switch to control the fan for our ventilation system [10]. The buzzer is used to alert residents of a break-in [11]. The LCD display shows the time of the day and temperature inside the house [12]. The humidity and temperature sensor detects the humidity and temperature of the house [13]. The photoresistor detects light outside the house and determines if the outside sensor is on or off [14]. The water level sensor detects if there are rising water levels inside the house, and will turn off components if the water level is high enough [15]. The 220-ohm resistors are used to limit the current going through each LEDs [16]. The connecting wires connect each part to the microcontroller and breadboard by using pins [17]. The Dupont Wires help connect each part that is farther away from the microcontroller and breadboard with female to male pins [18]. The breadboard is used to connect all different components together by using the pins from wires and other components [19]. The house frame is the main component of the design, where it will house all the electronics. The smart app system controls the lights inside and outside, the garage door, the security alarm, and the temperature of the house, along with synchronizing the time from the phone to the system. The power supply module and adapter give the circuit constant power to power each component [20].

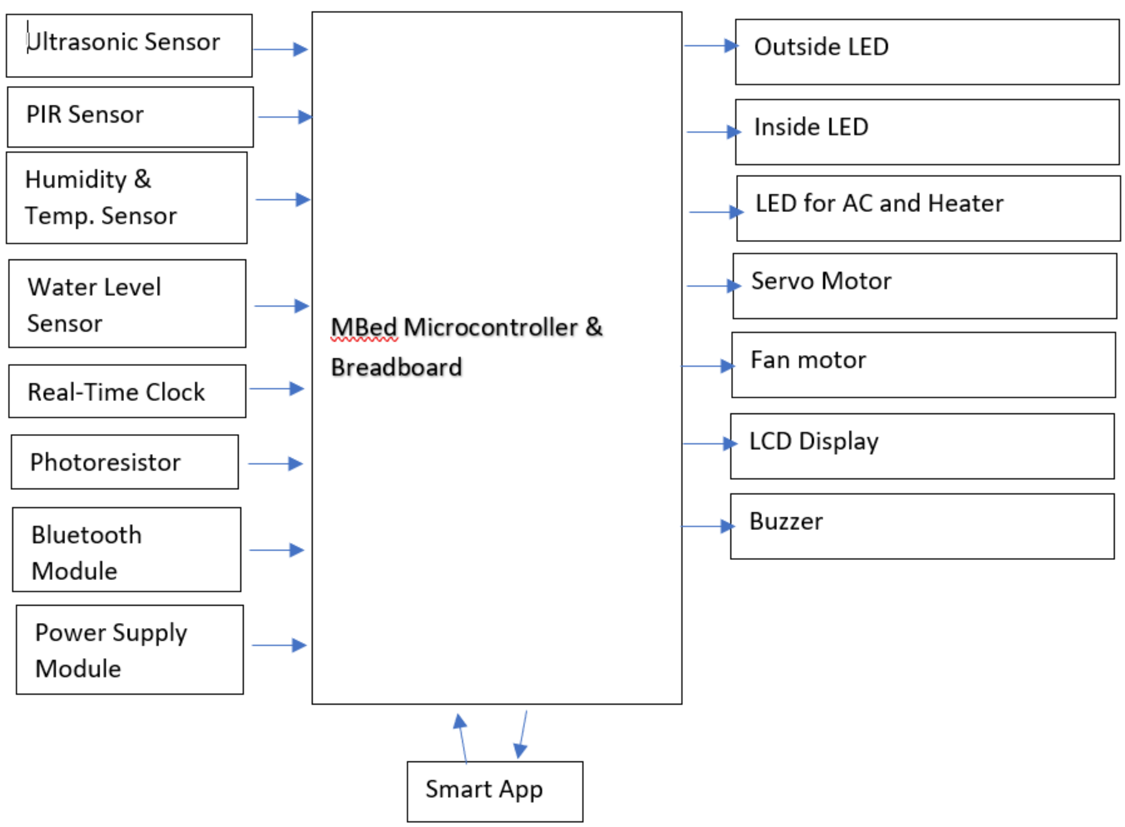


Figure 3.3.3: Block Diagram of Final Solution.

3.3.2 Features

The features of our final design is that it is compact, easy to use, and easy to manage, allowing it to be friendly towards anyone. With this design, all the wiring and components are placed around the house for accessibility. With the smart app system, it uses buttons allowing the user to manually control each component connected to the smart app. These include turning on and off the light inside and outside sensors of the house, garage door, temperature, security system, and a manual shutdown for all components. Also, pressing the buttons to turn on the components will turn green, and pressing the buttons to turn off components will turn red on the smart app. To implement the time of the day into the smart house, the system will synchronize the time from the user's phone and transfer the data of the time into the real-time module, which then will be displayed on the LCD display. For the outside of the house, there is an ultrasonic sensor above the door that detects if anything gets close to the door and will turn on the outside light. For the inside of the house, there is a PIR sensor hanging on the wall that detects movement within its range and will turn on the inside light. Also, there is a humidity and temperature sensor inside the house that detects the temperature of the room and how humid it is inside. The temperature is displayed on the LCD display for easy reading. Depending on how humid the room is, it will turn on the fan to ventilate the room if needed. Finally, the water level sensor inside the house will detect if any water levels are occurring and rising, and will turn off all components that give off power to reduce the risk for residents living inside.

Figure 3.3.4: Smart App.

3.3.3 Environmental, Societal, Safety, and Economic Considerations

When designing our smart home, we took into consideration the house and smart system to make sure that it was environmentally friendly, safe to use, and inexpensive. We also wanted to take into consideration what materials were used when designing and constructing the final design.

For environmental considerations, we constructed the entire frame all out of cardboard, which can be recycled very easily and be used for other purposes. We used a hot glue gun with 0.7cm diameter gun sticks to attach each cardboard component, making sure that the final design is stable enough to be moved around.

For safety considerations, the entire frame for our final design has no sharp edges to prevent the user from getting any kind of cuts. We also took into consideration to make sure that there were no bare or open wires where anyone could get an electrical shock.

Our economic considerations were that we used cheap and inexpensive materials to build and design our prototype. For our final solution, we would like to consider the size and scope of the house we are installing in. We are looking at the cost of the wire. By running most of the power in the house in parallel we can reduce our total wire usage and use as little wire to build the system within the house [21]. Unfortunately, as for sensors and other components, we cannot do much to reduce the cost of these parts.

For societal considerations, we are making sure that our system is not intrusive and does not collect unnecessary information from our users. Our app will not collect the data off of the user's phone other than the time for the initial startup of the system.

3.3.4 Limitations

With each design, there are always limitations. The limitation of our final design is that there weren't enough key components to make multiple rooms with smart technology inside the house. By adding smart technologies to multiple rooms, it would reduce the amount of manual effort to do certain tasks, but would raise the price of the smart home system with the addition of more components per room. Also, there was only a limited space on the microcontroller, only allowing a certain amount of features to be added to the smart home system. Additionally, the house itself is all made out of cardboard, which makes it less durable than wood, plastic, or metal. Using stronger materials, it would provide better durability for the house itself.

4 Team Work

4.1 Meeting 1

Time: March 6, 2022, 4:20 pm to 4:45 pm

Agenda: Discussing about parts/components and distributing tasks.

Team Member	Previous Task	Completion State	Next Task
Luka	N/A	N/A	Brainstorming Ideas
Brandon	N/A	N/A	Brainstorming Ideas
Toma	N/A	N/A	Brainstorming Ideas
Almat	N/A	N/A	Brainstorming Ideas

4.2 Meeting 2

Time: March 15, 2022, 2:30 pm to 4:20 pm

Agenda: Discussing about how each component will work together and writing a design problem.

Team Member	Current Task	Completion State	Next Task
Luka	Brainstorming Ideas	70%	Gantt Chart
Brandon	Brainstorming Ideas	70%	Problem Definition
Toma	Brainstorming Ideas, Writing Design Problem	70%	Function Tree
Almat	Brainstorming Ideas	70%	Objective Tree

4.3 Meeting 3

Time: March 21, 2022, 5:15 pm to 5:40 pm

Agenda: Distributing tasks to group members, and discussing the code.

Team Member	Current Task	Completion State	Next Task
Luka	Gantt Chart	10%	Building House
Brandon	Problem Definition	20%	Creating Code
Toma	Function Tree	40%	Introduction/Design Problem
Almat	Objective Tree	40%	Building House

4.4 Meeting 4

Time: March 29, 2022, 1:30 pm to 4:20 pm

Agenda: Start building the house, creating the code and writing up the report.

Team Member	Current Task	Completion State	Next Task
Luka	Gantt Chart, Building House	20%	Creating App
Brandon	Creating Code	20%	Considerations
Toma	Introduction, Design Problem	40%	Writing Solution 1 and 2
Almat	Building House	40%	Help Coding

4.5 Meeting 5

Time: April 1, 2022, 3:15 pm to 3:50 pm

Agenda: Discussing about the layout of the house

Team Member	Current Task	Completion State	Next Task
Luka	Gantt Chart, Building House	60%	Creating App
Brandon	Creating Code	27%	Considerations
Toma	Introduction, Design Problem	80%	Writing Solution 1 and 2
Almat	Building House	40%	Help Coding

4.6 Meeting 6

Time: April 5, 2022, 2:30 pm to 4:20 pm

Agenda: Building and coding

Team Member	Current Task	Completion State	Next Task
Luka	Creating App, Building House	70%	Help Coding
Brandon	Creating Code	35%	Considerations
Toma	Writing Solutions 1 and 2	60%	Final Solution, Conclusion
Almat	Creating Code	35%	Problem Definition

4.7 Meeting 7

Time: April 9, 2022, 1:30 pm to 4:20 pm

Agenda: Building and Coding

Team Member	Current Task	Completion State	Next Task
Luka	Creating App, Building House	50%	Help Coding
Brandon	Creating Code	40%	Write Considerations
Toma	Writing Solutions 1 and 2	80%	Final Solution and Conclusion
Almat	Problem Definition	90%	Help Coding

4.8 Meeting 8

Time: April 10, 2022, 1:30 pm to 4:00 pm

Agenda: Adding components and finish coding

Team Member	Current Task	Completion State	Next Task
Luka	Creating App, help coding	75%	PowerPoint
Brandon	Coding	70%	PowerPoint
Toma	Finishing Report	70%	PowerPoint
Almat	Adding Components	90%	Help Coding

4.9 Meeting 9

Time: April 11, 2022, 12:50 pm to 4:20 pm

Agenda: Finish Building and coding

Team Member	Current Task	Completion State	Next Task
Luka	Creating App, help coding	95%	PowerPoint
Brandon	Creating Code	90%	PowerPoint
Toma	Finishing Report	90%	PowerPoint
Almat	Help Coding	90%	PowerPoint

4.10 Meeting 10

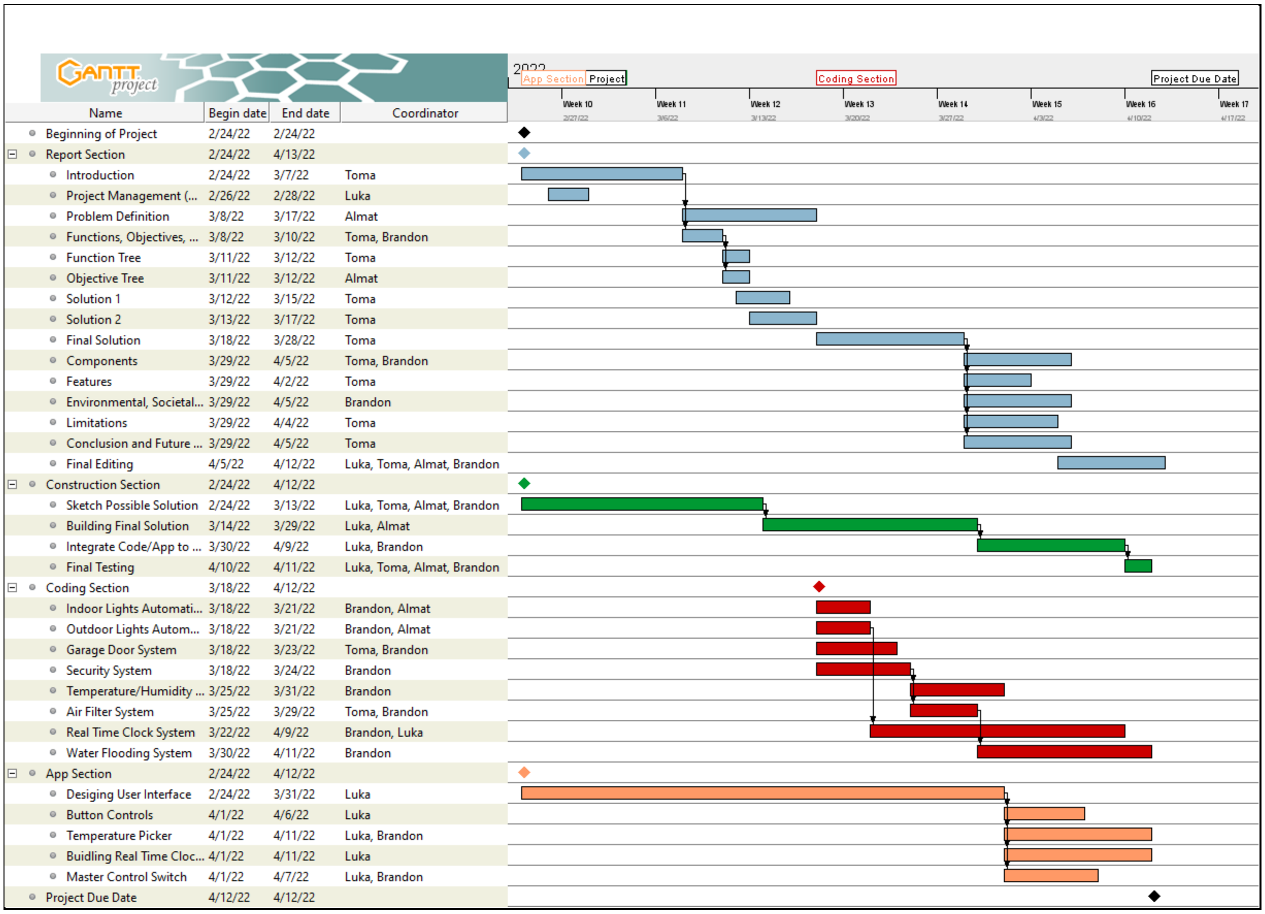
Time: April 12, 2022, 9:30 pm to 1:20 pm

Agenda: Final Testing and PowerPoint

Team Member	Current Task	Completion State	Next Task
Luka	Final Testing Code, PowerPoint	100%	N/A
Brandon	Final Testing Code, PowerPoint	100%	N/A
Toma	Final Testing Code, PowerPoint	100%	N/A
Almat	Final Testing Code, PowerPoint	100%	N/A

5 Project Management

Slack time is included with the tasks listed in gantt chart.



Name	Begin date	End date	Coordinator
Beginning of Project	2/24/22	2/24/22	
Report Section	2/24/22	4/13/22	
Introduction	2/24/22	3/7/22	Toma
Project Management (...	2/26/22	2/28/22	Luka
Problem Definition	3/8/22	3/17/22	Almat
Functions, Objectives, ...	3/8/22	3/10/22	Toma, Brandon
Function Tree	3/11/22	3/12/22	Toma
Objective Tree	3/11/22	3/12/22	Almat
Solution 1	3/12/22	3/15/22	Toma
Solution 2	3/13/22	3/17/22	Toma
Final Solution	3/18/22	3/28/22	Toma
Components	3/29/22	4/5/22	Toma, Brandon
Features	3/29/22	4/2/22	Toma
Environmental, Societal...	3/29/22	4/5/22	Brandon
Limitations	3/29/22	4/4/22	Toma
Conclusion and Future ...	3/29/22	4/5/22	Toma
Final Editing	4/5/22	4/12/22	Luka, Toma, Almat, Brandon
Construction Section	2/24/22	4/12/22	
Sketch Possible Solution	2/24/22	3/13/22	Luka, Toma, Almat, Brandon
Building Final Solution	3/14/22	3/29/22	Luka, Almat
Integrate Code/App to ...	3/30/22	4/9/22	Luka, Brandon
Final Testing	4/10/22	4/11/22	Luka, Toma, Almat, Brandon
Coding Section	3/18/22	4/12/22	
Indoor Lights Automati...	3/18/22	3/21/22	Brandon, Almat
Outdoor Lights Autom...	3/18/22	3/21/22	Brandon, Almat
Garage Door System	3/18/22	3/23/22	Toma, Brandon
Security System	3/18/22	3/24/22	Brandon
Temperature/Humidity ...	3/25/22	3/31/22	Brandon
Air Filter System	3/25/22	3/29/22	Toma, Brandon
Real Time Clock System	3/22/22	4/9/22	Brandon, Luka
Water Flooding System	3/30/22	4/11/22	Brandon
App Section	2/24/22	4/12/22	
Desiging User Interface	2/24/22	3/31/22	Luka
Button Controls	4/1/22	4/6/22	Luka
Temperature Picker	4/1/22	4/11/22	Luka, Brandon
Buidling Real Time Cloc...	4/1/22	4/11/22	Luka
Master Control Switch	4/1/22	4/7/22	Luka, Brandon
Project Due Date	4/12/22	4/12/22	

Table 5.1: Gantt Chart.

6 Conclusion and Future Work

In conclusion, we achieved plenty of considerable designs on how to create a smart home system that does not require a lot of human effort. First off, we came up with a couple of solutions that could be used to make the best possible smart home, while following all the constraints. Our objective was to build a program and working prototype of a smart home that is easy to use and does not require a lot of human effort.

From our research, the biggest challenges for homes are managing the home itself and the things that could possibly go wrong. Some such incidents could be break-ins, flooding, and system failures. Each of these events can't be predicted and waiting until the last minute could result in damages that can cost lots of money due to a lack of a system to warn residents about an unfortunate event. Furthermore, having unnecessary electronics and systems turned on could result in an increased cost from power consumption and negative effects of not being energy efficient[22][23].

Figure 6.1: Image of Smart Home System [24].

With the help of smart homes, we can reduce the number of unpleasant incidents with the use of smart home systems. Using these systems can reduce the chances of break-ins and alert residents of early floodings and system failures. Also, it can save money and increase energy efficiency by reducing the amount of power consumption with the touch of a button. As a result, unnecessary appliances and systems will be turned off to reduce power consumption [23].

For recommendations on how the process will continue in the future of improving smart homes, we would include more security throughout the house to make the house much safer from break-ins and add a password to turn off the alarm. Also, we could use another microcontroller to allow more components and features to improve house living. Additionally, we would include more digital screens around the house to display the date, day of the week, and weather.

7 References

1. L. Cvetkovska. "30 Smart Home Statistics for All High-Tech Enthusiasts." Comfy Living. https://comfyliving.net/smart-home-statistics/#:~:text=A%202019%20survey%20showed%20that,in%20the%20next%20three%20years (Accessed March 22, 2022)
2. J. Deesing. "The 10 Biggest Energy Wasting Habits At Home." Alliance to Save Energy. https://www.ase.org/blog/10-biggest-energy-wasting-habits-home (Accessed April 8, 2022)
3. "mbed LPC1768." Arm MBED. https://os.mbed.com/platforms/mbed-LPC1768/ (Accessed April 11, 2022)
4. "Arduino – Ultrasonic Sensor." Tutorialspoint. https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm#:~:text=Advertisements,or%201%E2%80%9D%20to%2013%20feet (Accessed April 11, 2022)
5. "How PIRs Work." Adafruit. https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work (Accessed April 11, 2022)
6. N. Koumaris. "Using The SG90 Servo Motor With An Arduino." Electronic-Lab. https://www.electronics-lab.com/project/using-sg90-servo-motor-arduino/ (Accessed April 11, 2022)
7. "First Test: Super StarterKit from Elegoo – Motor 3-6V DC." Project Hub. https://create.arduino.cc/projecthub/ingo-lohs/first-test-super-starterkit-from-elegoo-motor-3-6v-dc-5b199d (Accessed April 11, 2022)
8. "Learn About LED Lighting." Energy Star. https://www.energystar.gov/products/lighting_fans/light_bulbs/learn_about_led_bulbs (Accessed April 11, 2022)
9. "Interface DS1307 RTC Module with Arduino." Last Minute Engineers. https://lastminuteengineers.com/ds1307-rtc-arduino-tutorial/ (Accessed April 11, 2022)
10. P. Marian. "Single Pole Double Throw SPDT Relay." Electro Schematics. https://www.electroschematics.com/spdt-relay-switch/ (Accessed April 11, 2022)
11. "Piezo Buzzer." Arm MBED. https://os.mbed.com/components/Piezo-Buzzer/ (Accessed April 11, 2022)
12. "LCD (Liquid Crystal Display)." TechTarget. https://www.techtarget.com/whatis/definition/LCD-liquid-crystal-display#:~:text=LCD%20(Liquid%20Crystal%20Display)%20is,computer%20monitors%20and%20instrument%20panels. (Accessed April 11, 2022)
13. "DHT11 Humidity & Temperature Sensor." Mouser. https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf (Accessed April 11, 2022)
14. "How to Use a Photoresistor." Instructables circuits. https://www.instructables.com/How-to-use-a-photoresistor-or-photocell-Arduino-Tu/#:~:text=A%20photoresistor%20or%20photocell%20is,and%20dark%2Dactivated%20switching%20circuits. (Accessed April 11, 2022)

15. "How Water Level Sensor Works and Interface it with Arduino." Last Minute Engineers. https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/ (Accessed April 11, 2022)
16. "Why 220 ohm resistor is needed?." Arduino Forum. https://forum.arduino.cc/index.php?topic=519234.0 (Accessed April 11, 2022)
17. M. Hemmings. "What is a Jumper Wire?." Sparkfun Education. http://blog.sparkfuneducation.com/what-is-jumper-wire (Accessed April 11, 2022)
18. E. Erriez. "Dupont Crimp Tool Tutorial." Instructables circuits. https://www.instructables.com/Dupont-Crimp-Tool-Tutorial/#:~:text=Dupont%20is%20also%20called%20Jumper,Cheap. (Accessed April 11, 2022)
19. I. Buckley. "What Is a Breadboard and How Does It Work? A Quick Crash Course." MakeUseOf. https://www.makeuseof.com/tag/what-is-breadboard/ (Accessed April 11, 2022)
20. "Power Supply." Arduino. https://forum.arduino.cc/t/power-supply/131461 (Accessed April 11, 2022)
21. T. Thiele. "Series vs Parallel Circuits: What's the difference." The Spruce. https://www.thespruce.com/series-and-parallel-circuits-the-basics-1152850 (Accessed April 11, 2022)
22. E.V. Gravrock. "Evaluating Smart Home Challenges To Improve Future Smart Cities." Forbes. https://www.forbes.com/sites/forbesbusinesscouncil/2021/04/27/evaluating-smart-home-challenges-to-improve-future-smart-cities/?sh=7e3b06461097 (Accessed April 9, 2022)
23. "The 7 Greatest Advantages of Smart Home Automation." BlueSpeedAV. https://bluespeedav.com/blog/item/7-greatest-advantages-of-smart-home-automation (Accessed April 9, 2022)
24. C. Null. "Smart home systems: DIY vs. hiring a professional." TechHive. https://bluespeedav.com/blog/item/7-greatest-advantages-of-smart-home-automation (Accessed April 9, 2022)

8 Appendix

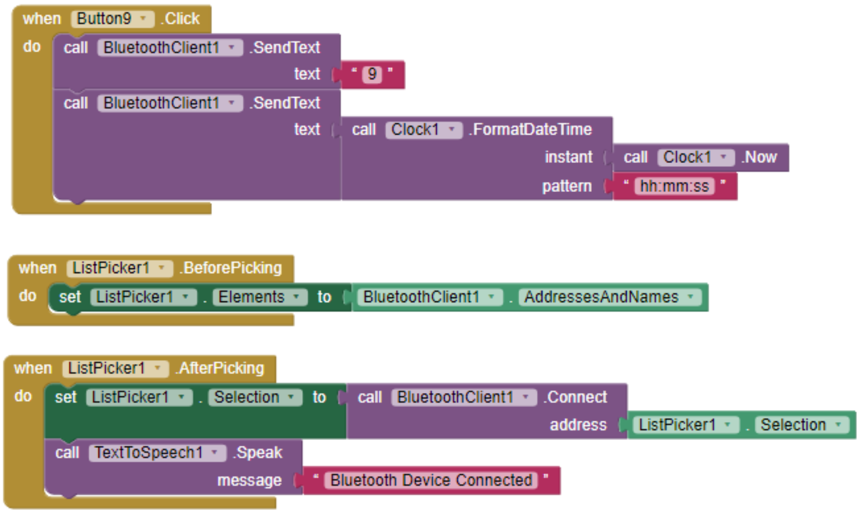


Figure A: App Blocks of Clock and Bluetooth Device

Figure B: App Blocks of Main Buttons

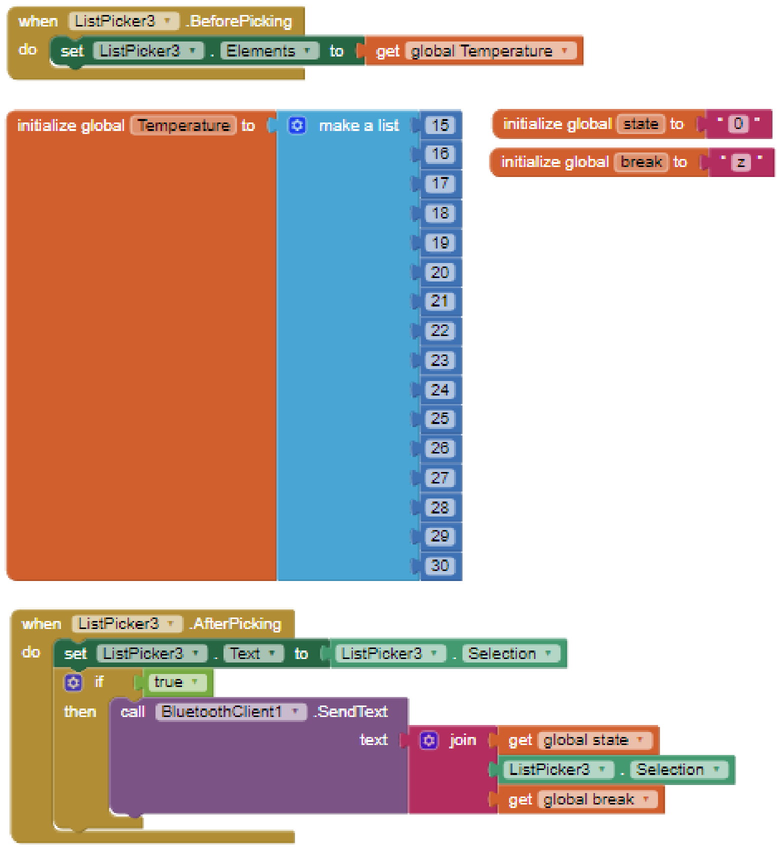


Figure C: App Blocks of Temperature Picker

8.1 Smart Home Code

```
#include "mbed.h"
#include "ds1307.h"    //RTC
#include "DHT11.h"//humidity and temp sensor
#include "hcsr04.h" // ultra sonic sensor
#include "TextLCD.h" // LCD screen
#include "Servo.h" //servo motor
AnalogIn water(p17);
DigitalOut buzzer(p6);
DigitalOut fan(p7);
DS1307 my1307(p28,p27);
Serial device(p9, p10);
DHT11 sensor(p8);
TextLCD lcd(p14, p12, p15, p16, p29, p30);
HCSR04 Distance(p11,p13); //ultra sonic sensor
DigitalIn LS(p25); //photo resistor
DigitalOut lights(p18);//outside lights
DigitalOut insideLights(p20);
DigitalIn insideSensor(p19);
DigitalIn security(p22);
DigitalOut AC(p23);
DigitalOut Heat(p24);
DigitalOut mastershutoff(p26);
Servo myservo(p21);
Serial pc(USBTX,USBRX);
DigitalOut myled(LED1);
int sec;
int minute;
int hours;
```

```
int day = 5;
int date = 8;
int month =5;
int year = 2022;
int ave = 0;
int temp;
unsigned int OSD = 0;
float waterlevel();
void alarm();
int get_temp();
void LCD(int hours, int mins, int secs, int temp);
void set_time(int sec,int min, int hours, int day, int date, int month, int year);
void get_time();
int ave_humidity(int h);
int get_avehumidity();
int get_humidity(int count);
int motion_sensor();
bool light_sensor();
void garage_door();
void fans(int on);
void ventalation(int count);
int OutSide();
void outsidelights();
void shut_off();
void shut_off()
{
  lights = 0;
  AC = 0;
```

```
    Heat = 0;
    insideLights = 0;
    fan =0;
    mastershutoff = 0;
}
float waterlevel()
{
    float waterlvl = water;
    return waterlvl;
}
void alarm()
{
    buzzer = 1;
    wait(1);
    buzzer = 0;
    wait(1);
}
int get_temp()
{
    int s = sensor.readData();
    int T = sensor.readTemperature();
    return T;
}
void LCD()
{
    temp = get_temp();
    get_time();
    lcd.printf("%.2i:%.2i:%.2i \n",hours,minute,sec);
```

```
    lcd.printf("%i C\n", temp);
}
void set_time(int sec,int min, int hours, int day, int date, int month, int year)
{
    my1307.settime( sec, min, hours, day, date, month, year);
    // wait(3);
}
void get_time()
{
    my1307.gettime( &sec, &minute, &hours, &day, &date, &month, &year);
}
int ave_humidity(int h)
{
    if (ave == 0) {
        ave = h;
    } else if(h <= ave+2) {
        ave =  (ave + h)/2;
    } else if ( h > (ave+2)|| (h == 0)) {
        ave = ave;
    }
    return ave;
}
int get_avehumidity()
{
    int a = ave_humidity(0);
    return a;
}
int get_humidity(int count)
```

```
{
   int s = sensor.readData();
   int humidity = sensor.readHumidity();
   ave_humidity(humidity);
   return humidity;
}
int motion_sensor(bool light)
{
   wait(1); //Wait for sensor to take snap shot of still room

   if (insideSensor == true) {
      return  insideLights = 1;
   } else
      return insideLights = 0;
}
bool light_sensor()
{
   pc.printf("i am in light_sensor\n");
   int  light =  LS;
   return light;
}
void garage_door_open()
{
   myservo.write(90);
}
void garage_door_close()
{
   myservo.write(-90);
```

```
}
void fans(int on)
{
   fan = on;
}
void ventalation(int count)
{
   int H = get_humidity(count);
   int h = get_avehumidity();
   if ( H > (h+2)) {
      fans(1);
   } else {
      fans(0);
   }
}
int OutSide()
{
   pc.printf("I am in outside\n");
   Distance.start();
   int D = Distance.get_dist_cm();
   return D;
}
void outsidelights()
{
   pc.printf("i am in outsidelights\n");
   if ((light_sensor()== false) || (OutSide() > 20)){
      lights = 0;
      pc.printf("I am in the first if\n");
```

```
    } else if(((light_sensor()== true )&& hours < 1) || ((OutSide() <= 20) && (hours >= 1))) {
        lights = 1;
        pc.printf("I am in the second if\n");
    }
}
int main()
{
    int count_set = 0;
    int state=0;
    int security1 = 0;
    int count = 1;
    bool light = false;
    device.baud(9600); //Set the baud rate to 9600
    char time [9];
    int h_a,h_b,m_a,m_b,s_a,s_b;
    int T=0;
    int OSL=0;
    mastershutoff = 1;
    while(1) {
        if( count == 1)
        {
         garage_door_close();
         wait(6);
        }
        LCD();
        if(OSL == 1) {
            outsidelights();
        }
```

```
int a =  ave + 2;
 pc.printf("%d\n",a);
 int b = get_humidity(count);
 pc.printf("%d\n",b);
 if ((b > a) && (mastershutoff == 1))
 {
  pc.printf("I am here 1");
  fan = 1;
 }
 else
 {
 fan = 0;
 }
 count++;
 float h2o = waterlevel();
 pc.printf("water = %f\n", h2o);
   if (h2o > 0.1) {
         alarm();
         shut_off();
         myled = 1;
    }
   else if ((water <= 0) && (state == 's'))
       {
          mastershutoff = 1;
       }
 if((security1 == 1) && (security == 0))
 {
    alarm();
```

```
}
if(device.readable()) {
    for(int i=0; i<9; i++) {
        time[i] = device.getc();
        if(time[i] == 'z') {
            break;
        }
    }
    state = time[0];

    switch(state) {
        case '1':
            pc.printf("im in state 1");
            OSL = 1;
            break;
        case '2':
            pc.printf("im in state 2");
            lights = 0;
            OSL = 0;
            break;
        case '3':
            pc.printf("im in state 3");
            light = true;
            motion_sensor(light);
            break;
        case '4':
            pc.printf("im in state 4");
            light  = false;
```

```
        motion_sensor(light);
        break;
    case '5':
        pc.printf("im in state 5");
        garage_door_open();
        break;
    case '6':
        pc.printf("im in state 6");
        garage_door_close();
        break;
    case '7':
        pc.printf("im in state 7");
        security1 = 1;
        break;
    case '8':
        pc.printf("im in state 8");
        security1 = 0;
        break;
    case '0':
        pc.printf("im in state 0");
        T = ((time[1]-48)*10) + (time[2]-48);
        pc.printf("%d",T);
        temp = 25;
        if (T > temp) {
            Heat = 1;
            AC = 0;
        } else if(T < temp) {
            AC = 1;
```

```
            Heat = 0;
        }
        break;
    case '9':
        pc.printf("im in state 9");
        if (count_set == 0) {
            h_a = ((time[1]-48)*10);
            h_b = time[2]-48;
            m_a = ((time[4] -48)*10);
            m_b =  time[5]-48;
            s_a = ((time[7]-48)*10);
            s_b =  time[8]-48;
            hours = h_a+h_b;
            minute = m_a+m_b;
            sec = s_a+s_b;
            set_time(sec,minute,hours,day,date,month,year);
            count_set = 1;
        }
        LCD();
        break;
  }
  }
 }
}
```