EE175AB Final Report

Autonomous Home System/Garage Sensor

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

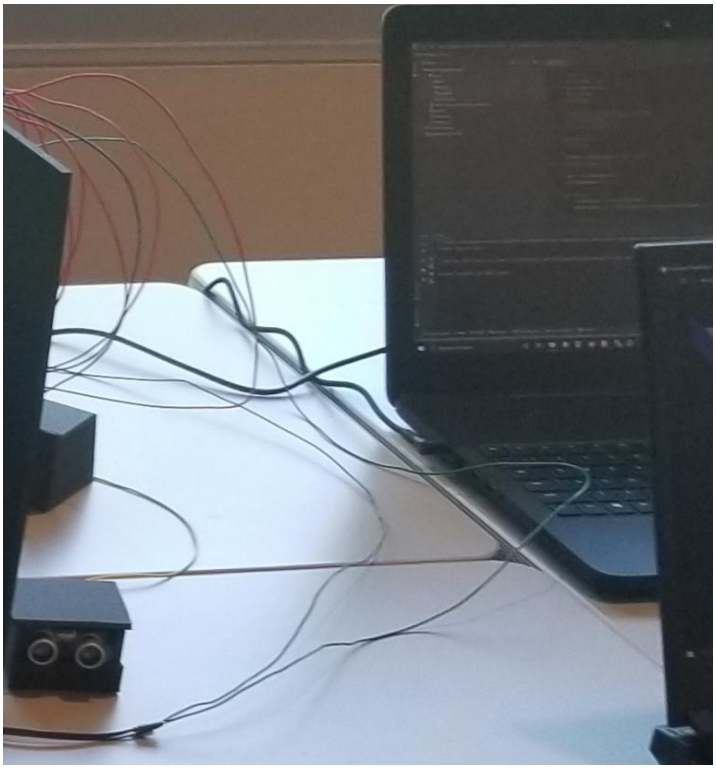
Project Team Member(s)	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan
Date Submitted	03/11/2022
Section Professor	Roman Chomko (Professor), Maliha Tasnim (TA)
Revision	Revision 6.0
URL of Project YouTube Videos, Wiki/Webpage	Code Link: https://drive.google.com/drive/folders/1D9LtfbukFU6IwMimplIv2GoSg-CRbf3M?usp=sharing Video Link: https://youtu.be/j1qS2g2dSCA
Permanent Emails of all team members	carel009@ucr.edu, gjabe001@ucr.edu, rbhui001@ucr.edu

Summary

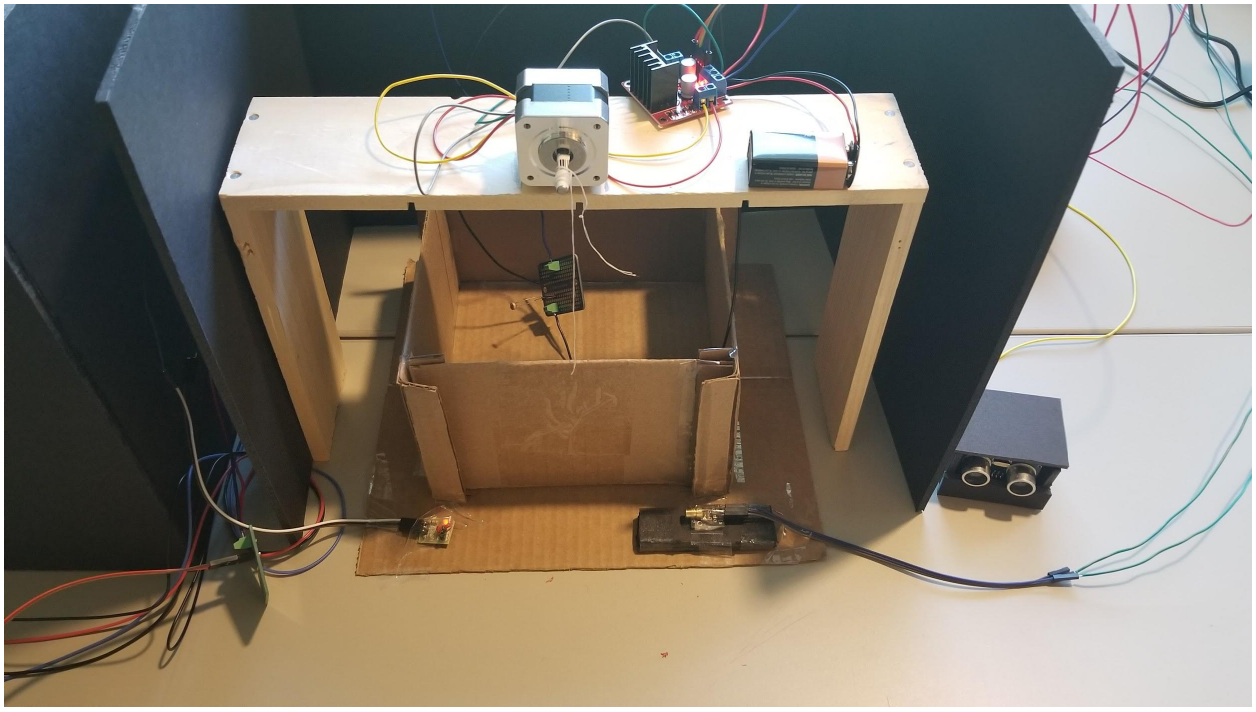
This report presents the autonomous home system/garage sensor. The project is composed of multiple subsystems for smart homes/security systems.

Autonomous Home System/Garage Sensor

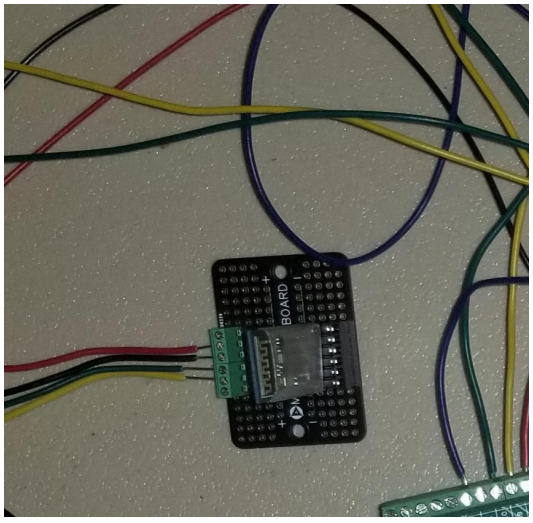
Ai License Plate Reader



Garage Door Control

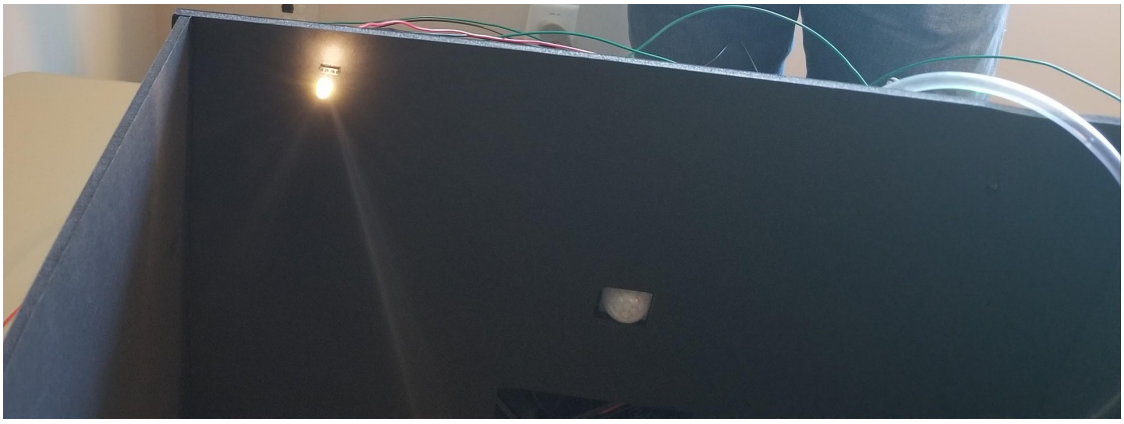


Bluetooth Keypad Entry

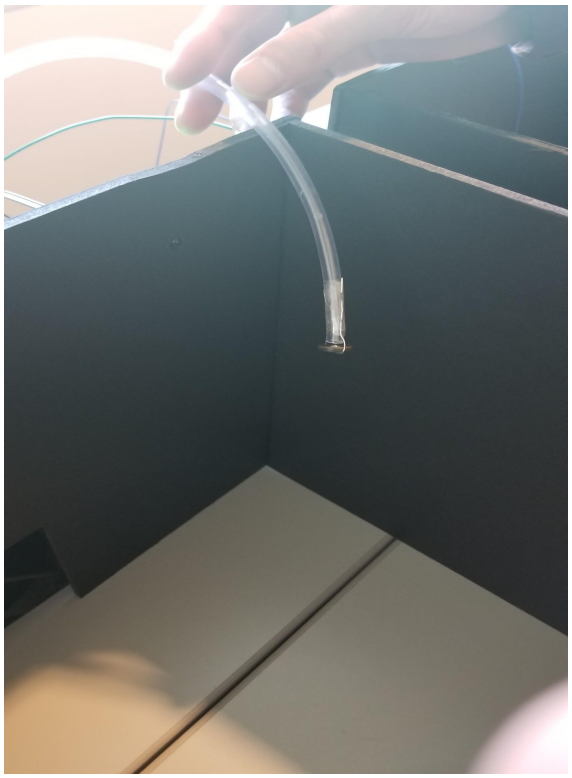


Smart Temperature Control

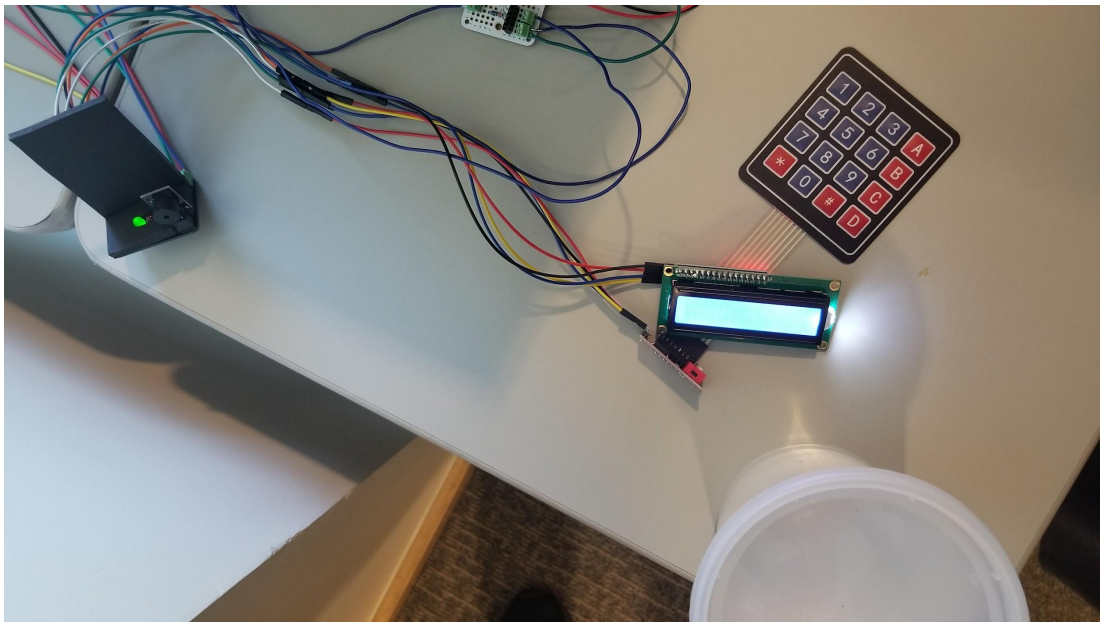
Ambient Lighting Control



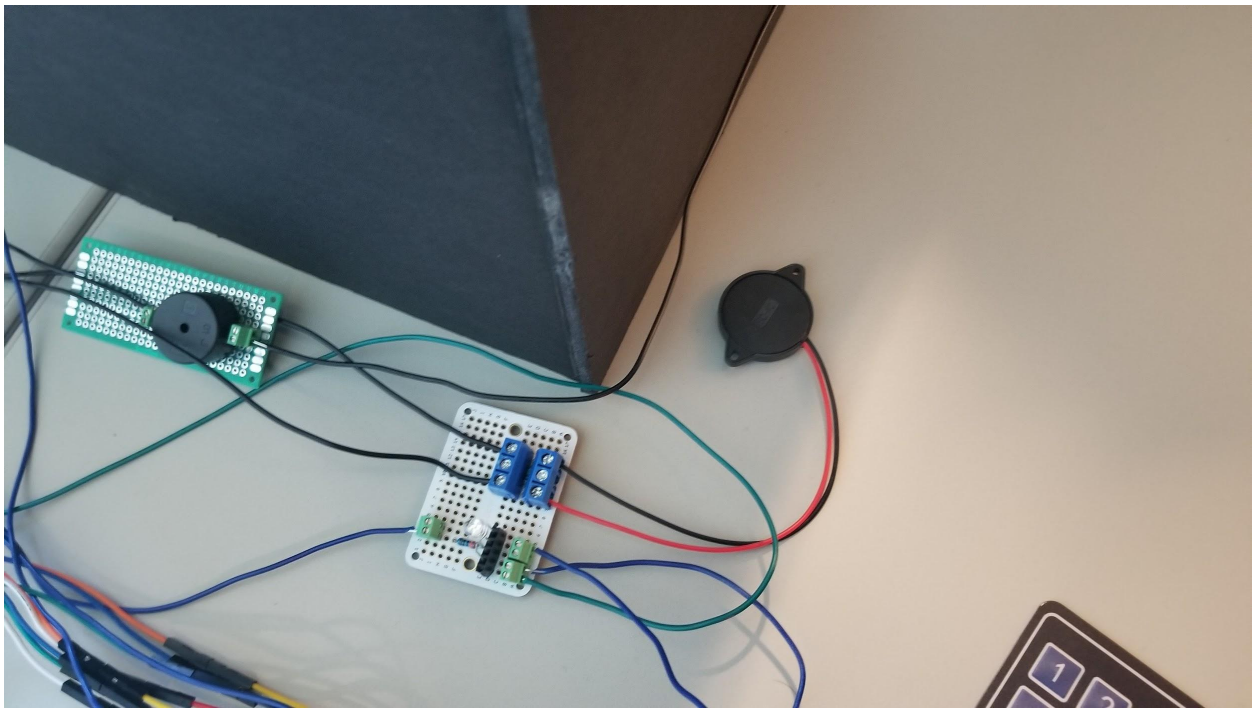
Fire Detection System



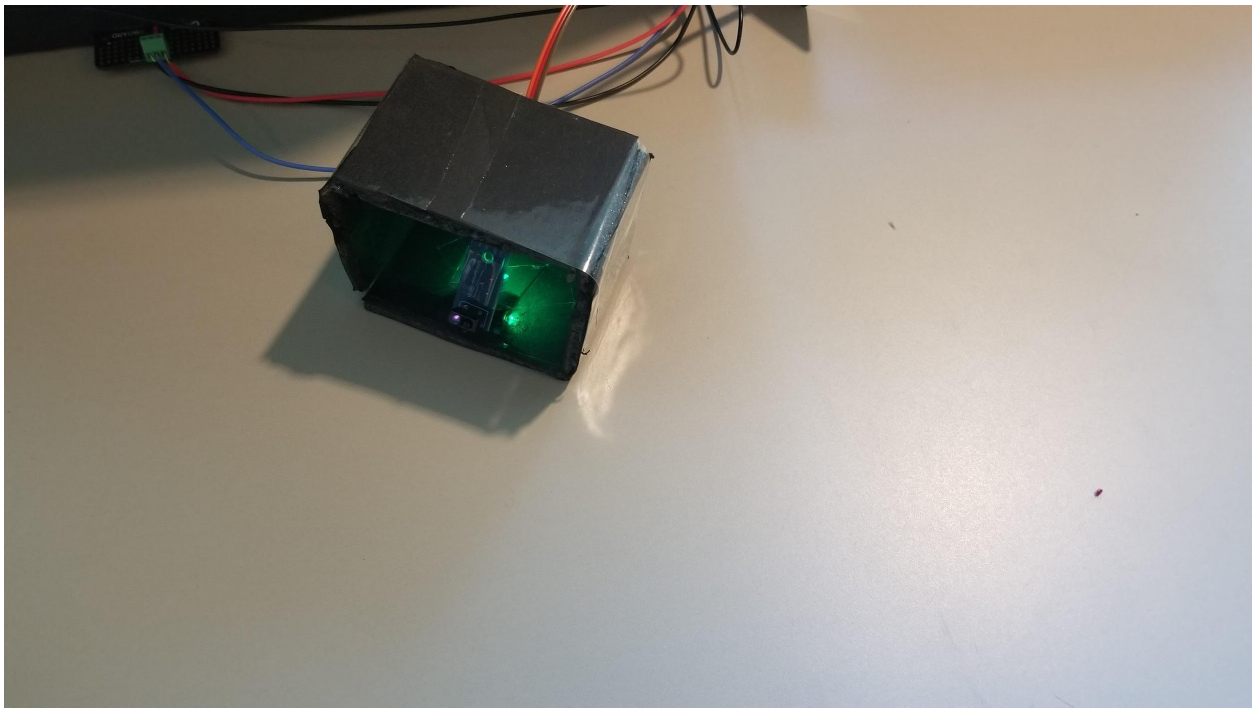
Home Security System



Door Knock Recognition



Package Detection System



Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
v6	Minor changes to v5	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan	03/08/2022	
v5	All subsystems merged and working together on PCBs	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan	03/04/2022	
v4, v3, v2	Merge fixes	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan	02/11/2022 to 02/25/2022	
v1	Initial merge	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan	02/04/2022	

Table of Contents

1 * Executive Summary

The motivation of the project is to provide customers with the option of a cheap and reliable smart home/security system that is non intrusive.

The goal of the project is to use basic hardware and software in a way that is user friendly and that allows for both modularity and users to build their own smart home/security system.

The design objective is to use both hardware and software that is readily available to all and use everyday devices to reduce cost.

Key features of the project are: (The names after each feature signifies who is responsible)
Ai License Plate Reader (All)
Garage Door Control (Graham & Raunac)
Bluetooth Keypad Entry (Christopher)
Smart Temperature Control (Christopher)
Ambient Lighting Control (Graham)
Fire Detection System (Graham)
Home Security System (Christopher)
Door Knock Recognition (Raunac)
Package Detection Sensing (Raunac)

Testing results:
Christopher Arellano:
System Testing - conducted by trying multiple input combinations, triggering sensors with varying ranges, motions and temperatures.
Design Specification accuracy - 80% - 90%

Raunac Bhuiyan:
The garage door + License plate reader - tested using two conditions: 1. With license plate recognition 2. Opening with only light. Specification accuracy - 95%
The door knock system - conducted with varying “knocks” i.e. how loud does the knock need to be for the sensor to pick up. Specification accuracy - 90%
Package system - conducted with how far away can the sensor detect an object. Specification accuracy - 100%

Graham Jabeguero:
System Testing - tested the Fire Detection System and Ambient Light System individually, together, then with all other parts.
Design Specification accuracy - 90%

Important Achievements:

Christopher Arellano:
All subsystems functioning as intended, Home Security System is activated/deactivated upon correct pin entry(1234*), Home Security System is triggered with motion if alarm is active, Home Security System displays system messages, Bluetooth Keypad Entry app made to function as physical Keypad, Smart Temperature Control is triggered with motion when in auto temp(A*), Smart Temperature Control is always on when in always active(AA*), Smart Temperature Control is always off when in temp off(AAA*), Smart Temperature Control displays current temperature when in display(D*), sonar sensor triggers arduino to send a message to python to take a picture for Ai License Plate Reader.

Raunac Bhuiyan:
All subsystems function 100% as intended. The Package Detection system uses an Infrared sensor to detect any object within 2.5 centimeters. The doorknock system is activated with a piezo sensor which senses a tap on the surface (i.e. knocks on the table), to alarm the buzzer and LEDs. The garage door controller can be opened by exposing the photoresistor to light or reading a license plate from python. To close the garage, assuming a car has entered the garage, and the headlights turn off, the photoresistor no longer detects light, and the garage begins to transition to its closed state.

Graham Jabeguero:
All subsystems operated as expected. The Fire Detection System activates the sprinkler when the infrared flame sensor senses fire and deactivates the sprinkler when fire is no longer detected. Also when the system detects fire, it will print the phrase “FIRE DETECTED” on the LCD screen and when it stops detecting fire, it will print “Sprinklers stop.” As for the Ambient Lighting Control, a photoresistor detects the outside light levels to output the intended brightness for the LED inside the home. When the photoresistor detects full brightness outside, the LED will turn off. When the photoresistor detects no light outside, the LED will turn on with its brightest settings. There are three other brightness levels in between these two making a total of five settings.

2 * Introduction

This space may be used to provide an introduction for the design and ties to other project materials.

2.1 * Design Objectives and System Overview

What was designed?: Autonomous Home System/Garage Sensor

Technical principles: Smart Home/Security System

Why is this a meaningful project?: The project contains features that current smart home/security systems lack and would like to see implemented in the near future.

What are the intended applications?: The intended applications are to introduce new smart home/security system features that are non intrusive, effective, modular and affordable.

How is it related to subjects of electrical engineering?: The project deals with circuit design, logic design, embedded systems design, pcb design, pwm, i2c, analog signals, digital signals, bluetooth communication, serial communication, analog to digital conversion, etc.

Bluetooth Keypad Entry: takes ~3 seconds for signal to be received by Arduino
Smart Temperature Control: takes ~5 seconds to trigger temp. control upon movement
Home Security System: takes ~2 seconds to trigger alarm upon movement, when armed
Ai License Plate Reader(sonar): takes ~2 seconds to detect an object within range and send signal to python to take a picture
Ai License Plate Recognition (reading the plate and activating the stepper motor): After a picture is taken, the code reads the plate and triggers the stepper motor within ~2 seconds.
Package Detection System: Takes ~2 seconds for the sensor to detect an object within 2.5 centimeters.
Door Knock Recognition: After sensing the “knock” it takes ~1 second to trigger the buzzer and LED lights.
Fire Detection System: Turns on sprinklers when the system senses fire for ~ 3 seconds then turns off after it stops sensing fire for ~ 2 seconds.
Ambient Light: If the system senses a light level of more than 600, turn off lights; Less than 600 will gradually emit a brighter light depending on how low the light level is.

Initial Approximate Cost: ~$169
Final Approximate Cost: ~$371

Christopher Arellano:
Home Security System
Bluetooth Keypad Entry
Smart Temperature Control
AI License Plate Reader (Sonar sensor, Arduino & python serial communication, Camera control)

Raunac Bhuiyan:
AI License Plate Reader (Reading the license plate, activate Stepper motor, Arduino & python serial communication)
Garage door controller

Door Knock recognition
Package Detection system

Graham Jabeguero:
Fire Detection System
Ambient Light System
Garage Door Control
AI License Plate Reader (Image Processing)

2.2 * Backgrounds and Prior Art

This LPR/ANPR camera is able to read license plates. Furthermore, this camera is able to read license plates from a certain distance. Similarly, for our project we decided to implement a sonar sensor which is used to detect when a car is in close proximity. At this point, the camera will know when to take a picture of the license plate. In other words, this product is somewhat similar to our ai license plate recognition system. Of course, it is important to note that this product is mostly used for reading non U.S. license plates. In other words, it is most likely able to read only Indian license plates. This is likely due to the formatting of the license plates compared to countries outside of the U.S. Similarly, our ai license plate is only able to read non U.S. license plates [4].

2.3 * Development Environment and Tools

AI license plate reader:

Arduino

Camera

PC

Sonar sensor

Arduino IDE

Pycharm IDE

Garage Door control:

Arduino

Photoresistor

Stepper motor

L298N motor driver

9V Duracell battery

Laser trip sensor

Arduino IDE

Bluetooth keypad entry:

Arduino

Bluetooth receiver

Smartphone

MIT App Inventor

Arduino IDE

Smart temperature control:

Arduino

Temperature sensor

PIR sensor

12v fan x2

12v peltier pad x2

Arduino IDE

Ambient lighting control:

Arduino

Photoresistors

LEDs

Arduino IDE

Fire detection system:

Arduino

Flame sensor

Sprinkler water pump

Arduino IDE

Home security system:

Arduino

LEDs

Keypad

LCD display

PIR sensor

Buzzer

Arduino IDE

Door knock recognition:

Arduino

Piezo sensor

Passive Buzzer

RGB (Red, Green, Blue) LED

Arduino IDE

Package Detection sensing:

Arduino

Infrared sensor

LED

Arduino IDE

Hardware used:

Arduino

PC

Photoresistors

LEDs

12v fan x2

12v peltier pad x2

Temperature sensor

Bluetooth receiver

Smart phone

Camera

Sonar sensor

Laser trip sensor

Keypad

LCD display

Infrared sensor

PIR sensor

Sprinkler water pump

Buzzers

Flame sensor

Piezo sensor

Stepper motor

Software Used:

MIT App Inventor

Arduino IDE

Pycharm IDE

2.4 * Related Documents and Supporting Materials

N/A

2.5 * Definitions and Acronyms

N/A

3 * Design Considerations

This section describes issues that need to be addressed or resolved prior to or while completing the design as well as issues that may influence the design process.

3.1 * Realistic Constraints

Realistic Constraints: Time, Skill, Size, Cost

Time: not enough time to implement everything intended in a more presentable cleaner manner. Took longer than anticipated to merge all project components. Parts got damaged and needed to be replaced. Meet up time conflicts.

Skill: most skills needed to complete the project needed to be picked up along the way.

Size: project size is big and hard to carry and requires everything to be assembled and disassembled for demos/presentations.

Cost: project required more than the $200 budget and many materials used were not counted in the project cost. Parts got damaged and needed to be replaced.

3.2 * Industry Standards

Industry Standards Involved in the Project: I2C, Bluetooth communication, Serial communication via USB

Industry Standards effects on Project: Arduino pin usage reduction by using I2C, wireless communication from smartphone to Arduino via Bluetooth, serial communication between Arduino and python via USB.

3.3 * Knowledge and Skills

Christopher Arellano:

- How to make an app
- How to appropriately use sensors
- How to control temperature using fans and peltier pads
- How to use serial communication to communicate from arduino to python
- How to use I2C for both LCD display and Keypad
- How to use Bluetooth to receive signals from smartphone to arduino

Courses:

- EE 001A
- EE 001B
- EE 100A
- CS 010A
- CS 010B
- CS 010C
- CS 061
- EE/CS 120A
- EE/CS 120B
- EE/CS 122A

- EE/CS 128

New Knowledge/Skills:

- How to make an app using MIT App Inventor
- How to communicate from arduino to python
- How to code in python
- How to code arduino
- How to use sensors
- How to use I2C
- How to use Bluetooth
- How to control fans & peltier pads
- How to solder
- How to design PCBs
- How to work in a team
- How to debug code effectively
- How to debug circuitry
- How to merge code successfully

Raunac Bhuiyan:

- How to appropriately use sensors
- Having a good foundation of the C/C++ language to write most of the code for arduino
- To avoid any memory leaks in the code, no pointers or linked lists were utilized. In fact, every system except the door knock, does not utilize arrays or vectors.

Courses:

- EE 100A and EE100B
- CS 010A/010
- CS 010B/012
- EE/CS 120B
- EE 128

New Knowledge/Skills:

- How to use Serial Communication to make the Arduino IDE and Pycharm (python) IDE work together (in our case, the license plate recognition system works with the garage door)
- Write code in python
- Using an openCV code and adjusting it to work with our final project
- Write code in arduino
- How Sensors work with Input and Output + the difference between analog and digital pins
- How to operate a stepper motor with a L298N motor driver and 9V battery
- How to use an Infrared sensor (both digital and analog input)
- Learn the difference between active and passive buzzer (chose the passive buzzer for final project)
- How to control the piezo sensor
- How to use the K64F (kinetis) microcontroller

Graham Jabeguero:

- How to appropriately operate the water pump using a relay module
- How to correctly setup sensors
- How to work with Arduino IDE
- How to work with Python

Courses:

- CS 010
- CS 012
- CS 061
- EE 100A
- EE 100B
- EE/CS 120A
- EE/CS 120B
- EE 128

New Knowledge/Skills:

- How to effectively work as a group
- How to design and solder PCBs
- How to code in Arduino
- How to code in Python
- How to correctly use sensors
- How to get familiar with Arduino Uno and Arduino Mega
- How to properly cut electrical wires and be used for PCBs
- How to debug and integrate subsystems efficiently

3.4 * Budget and Cost Analysis

Cost Analysis:
Arduino Mega (~$40)
Photoresistors 2 pcs (~$2)
LEDs (~$5)
12v fan x2 (~$16)
12v peltier pad x10 (~$27)
Temperature sensor (~$2)
Bluetooth module (~$7)
Sonar sensor (~$2)
Acxico Laser Sensor 2 pcs (~$7)
Keypad (~$5)
I2C LCD display (~$5)
HiLetgo Infrared sensor 10pcs (~$9)
PIR sensor (~$2)
Sprinkler water pump (~$4)
Passive Buzzer 2 pcs (~$3)
Flame sensor (~$2)
Piezo sensor (~$2)
Stepper motor (~$14)
Duracell Battery 4 pcs (~$16)
L298N motor driver (~$7)
Tuofeng solid wires (~$17)

Tuofeng silicon wires 2 sets (~$35)
Mini PCB boards 3 sets (~$45)
Screw Terminal headers 2 sets (~$30)
Arduino Mega screw terminal shield (~$17)
Screw terminal power rail set (~$11)
12v power supply (~$13)
Relays (~$20)
I2C keypad adapter (~$6)

Final Approximate Cost: ~$371

3.5 * Safety

Safety Considerations: The safety of the user was taken into consideration when designing the project. Any design that was considered a hazard was fixed by making changes to the designs. One major safety consideration was made for the original Package Detection Sensing design. The original design was to use a laser and mini mirrors to create a laser grid to detect packages. The change made with the consideration of user safety was to scrap the original design and use an infrared sensor instead to detect packages. The safety considerations for the project were to check circuitry before connecting to the rest of the subsystems, check for correct wiring, disconnect from power when working on circuits, and check code for errors that would prevent the system to function as intended.

Safety Objectives: The safety objectives of the project are to maintain the user safe from hazardous designs and electrical malfunctions. The safety objectives for the project were to avoid electrocution when working on circuits by disconnecting from power, avoid improper circuit building by checking circuit design, and avoid improper wiring by checking wiring before powering on.

3.6 * Documentation

Project documentation and changes to designs were made on a shared google doc.

Change notices and project progress was communicated via discord.

Version control was achieved by creating new code files every time a major change was made.

4 * Experiment Design and Feasibility Study

Include this section if you need to conduct experiments to evaluate the feasibility of your project ideas, alternatives, trade-offs and realistic engineering constraints, and to answer key design questions, such as what parts to use, how to collect data, whether accuracy of sensor is sufficient for the design objective, whether battery provides enough power, what hardware or software interface methods to use to connect the different modules, etc.

4.1 * Experiment Design

Christopher Arellano:

Keypad Input Test: Tested by trying multiple input combinations.

Bluetooth Keypad Input Test: Tested by trying multiple input combinations.

Temperature Sensor Test: Tested by varying the temperature using fans and peltier pads.

PIR Sensor Test: Tested by triggering sensors with varying motions.

Sonar Sensor Test: Tested by triggering sensors with varying ranges.

Raunac Bhuiyan:

Garage Door Control and Ai License Plate reader (w/ Arduino and Python ONLY): A stepper motor rotates clockwise and counterclockwise. In this case, it is used to open and close the garage. A laser and laser sensor is used as a safety feature to prevent anyone or an object from entering the garage while the stepper motor is running. In other words, when the garage door is opening or closing, no one can pass through. Also, inside the garage, there is a photoresistor which can detect light. Moreover, if a car were to turn on its headlights from inside the garage, the door would open. On the other hand, if a car has entered the garage and the photoresistor still detects the headlights, then the garage will remain in an open state. As for the license plate reader, it is written in python. It can take a photo of a license plate, and if the characters are recognized, then it will send a signal to the arduino via serial communication and activate the stepper motor. To clarify, the sonar sensor detects when a car is in range, at that point the webcam/camera will take a photo of the license plate.

The Package Detection System: An HiLetgo Infrared sensor has both digital and analog capabilities. In this case, the analog port was utilized. This sensor is able to detect an object within a range of 2.5 centimeters. Therefore, if the sensor is able to detect an object within that range, then the LCD display screen will output “Object detected.” If the object/package is removed from that specific area, then the screen will display “Object removed.”

The Door Knock Recognition System: A piezo sensor uses analog input and output. In this case, the sensor is placed on the table. Furthermore, if a user were to knock or hit the table with a certain amount of force, the sensor would detect the “knock.” Moreover, after the sensor notices a “knock,” the passive buzzer will play a tune which consists of three different frequencies. For clarification, the three frequencies alternate and play in order once forward then backward. Furthermore, as the sound plays, an RGB LED (i.e. one LED that can alternate between three colors at once), will change the color depending on the frequencies. Of course, the buzzer and LED is only used to notify the user that someone has knocked on the door.

Graham Jabeguero:

Flame Sensor Test:

Objective is to correctly sense if there was a flame nearby using a lighter.

Tested by using a lighter and also individually testing multiple flame sensors of the same model for more accuracy.

Water Pump Test:

Objective is to correctly turn on a relay module, which in turn will turn on the water pump, once it receives a signal from the flame sensor.

Tested by altering the settings of the relay module, such as setting the level trigger to high or low, and debugging the code based on the level trigger. Also using a lighter to make the flame sensor give a signal to the relay turn on.

Ambient Light's Sensor Test:

Objective is to correctly configure the photoresistor to sense the correct analog to digital values.

Tested by setting up the photoresistor whether the wires were correct and using a flashlight on the photoresistor to give varying values.

Garage Door Control Test:

Objective is to open and close the garage door without the need for the user to manually do this task.

Tested by using a flashlight as the car for the garage and using an infrared sensor, photoresistor, and laser to correctly go through the phases of the automated garage system.

AI License Plate Reader Test:

Objective is to correctly read the letters of a license plate by using a webcam.

Tested by using a webcam from a laptop and displaying a picture of a car (with a readable, clear license plate) in front of the camera. Used two different pictures of a car and tested each picture ten times and with different lighting.

4.2 * Experiment Results, Data Analysis and Feasibility

Christopher Arellano:

Keypad Input Test Result: Correct input reads if Ai License Plate Reader is not used (Accuracy: 90%).

Bluetooth Keypad Input Test Result: Correct input reads if Ai License Plate Reader is not used (Accuracy: 90%).

Temperature Sensor Test Result: Temperature reads are within + or - 2 degrees Farenheit from actual temperature (Accuracy: 90%).

PIR Sensor Test Result: Triggered with any slight motion that is visible by the sensor (Accuracy: 85%).

Sonar Sensor Test Result: Triggered multiple times if object is detected moving (Accuracy: 80%).

Raunac Bhuiyan:

Ai License Plate recognition result: The code is an openCV. It is able to detect a license plate from a picture; however, it is only able to read characters from a non U.S. license plate (i.e. Indian license plate). Furthermore, sometimes it does not read the characters properly. For example, if the characters on the plate are “KA05ME”, it will be read as “KAosTL” or something equivalent. Therefore, to account for these cases, the code was adjusted to grant access to the garage door, as long as it could read some of the characters or a blank string. In other words, it will not read any characters on the license plate and will grant access. Furthermore, it has interference from the Bluetooth Keypad as described above. Total system accuracy is 75%.

The garage door result: The stepper motor will only activate if either the photoresistor senses a certain amount of light or a license plate has been read from the code as described in 4.1. Total system accuracy is 95%.

The Package Detection system: The infrared sensor detects objects within 2.5 centimeters. Total accuracy 99%.

The door knock system: The piezo sensor can detect a knock on the table, and triggers the passive buzzer. However, the threshold had to be changed on some occasions because sometimes the sensor reads a “knock” even with a slight tap. Moreover, this causes the buzzer and LED’s to activate when not intended. Total system accuracy is 80%

Graham Jabeguero:

Flame Sensor Test Result: Reads the correct analog-to-digital values when a flame is nearby. (Accuracy: 99%)

Water Pump Test Result: Correctly turns on relay module which then turns on/off the water pump. There is some delay due to interference with other subsystems. (Accuracy: 90%)

Ambient Light Sensor Test Result: Reads the correct analog-to-digital values based on the light it receives. Slight flickers of the LED due to the alternating shifts of light the sensor is picking up. (Accuracy: 80%)

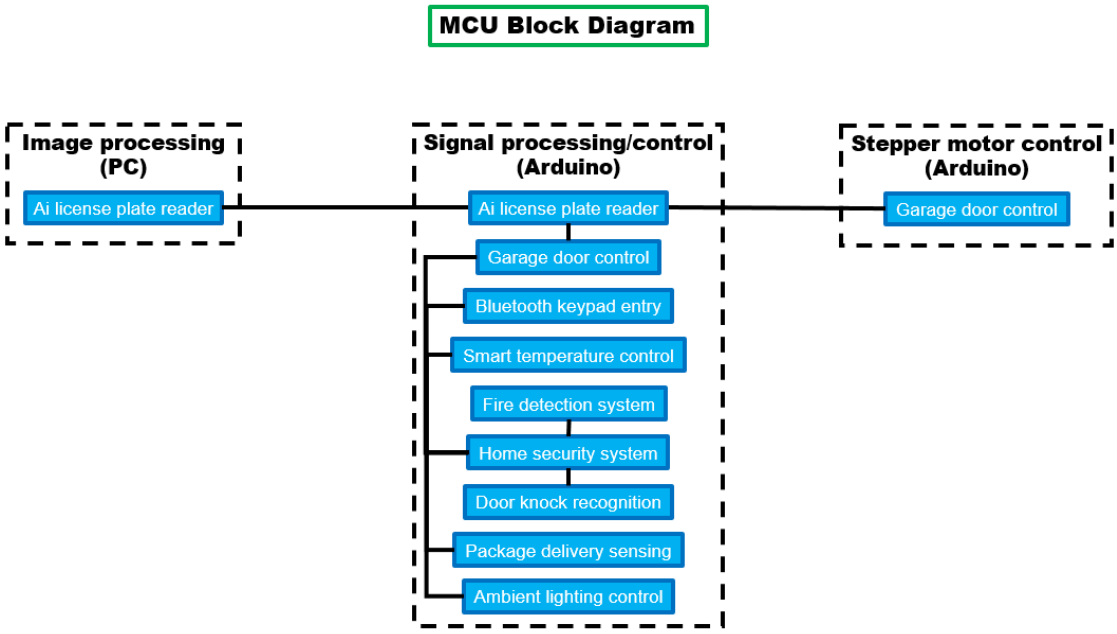
Garage Door Control Test Result: Correctly goes through each phase seamlessly as stepper motor and each sensor works properly. (Accuracy: 95%)

AI License Plate Reader Test Result: Slight reading errors when reading the first picture. Second picture would almost read as well but needed sufficient light to do so. (Accuracy: 75%)

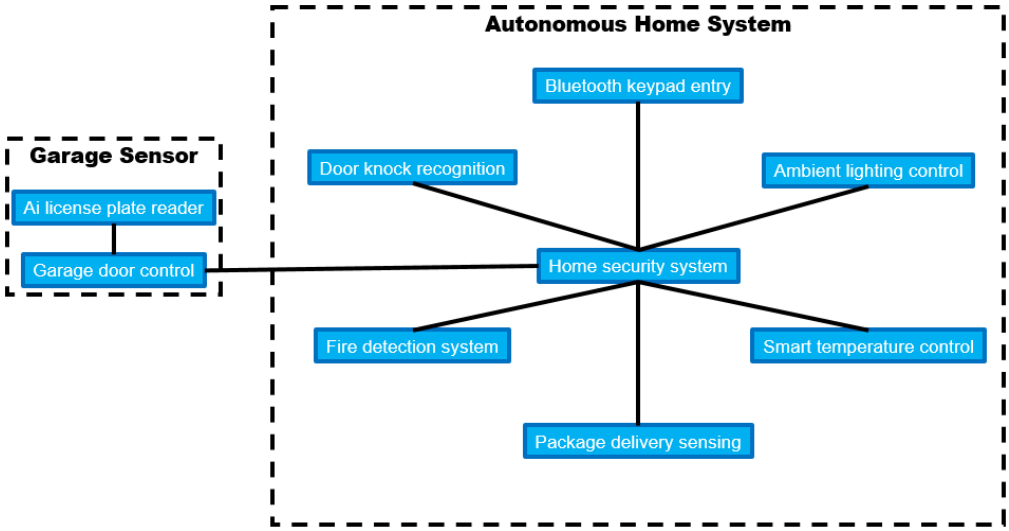
5 * Architecture and High Level Design

The architecture provides the top level design view of a system and provides a basis for more detailed design work. These are the top level components of the system you are building and their relationships.

5.1 * System Architecture and Design

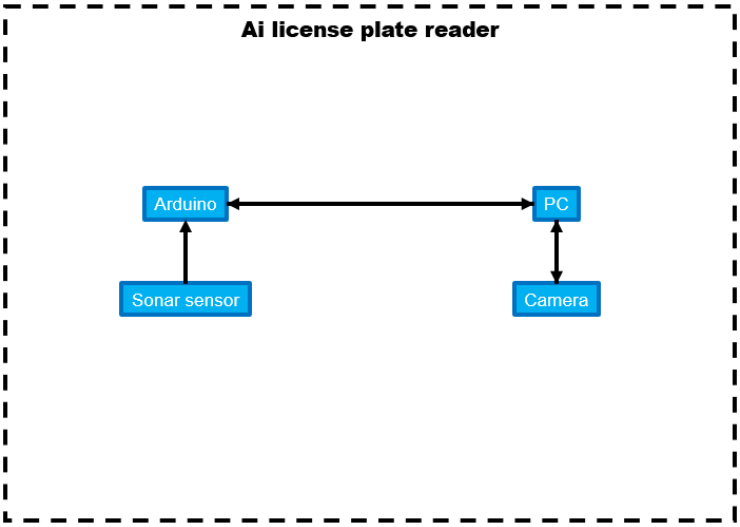


Simplified System Block Diagram



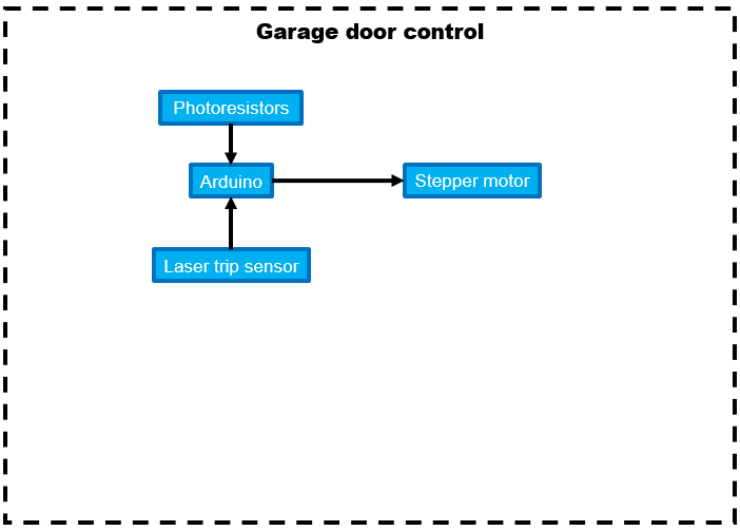
(1)
- Arduino
- PC
- Camera
- Sonar sensor

Sub-System Block Diagram



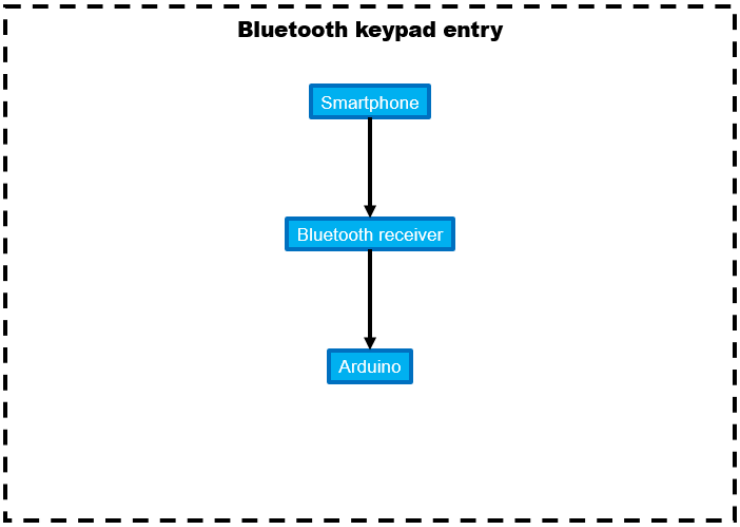
(2)
- Arduino
- Photoresistors
- Stepper motor
- Laser trip sensor

Sub-System Block Diagram



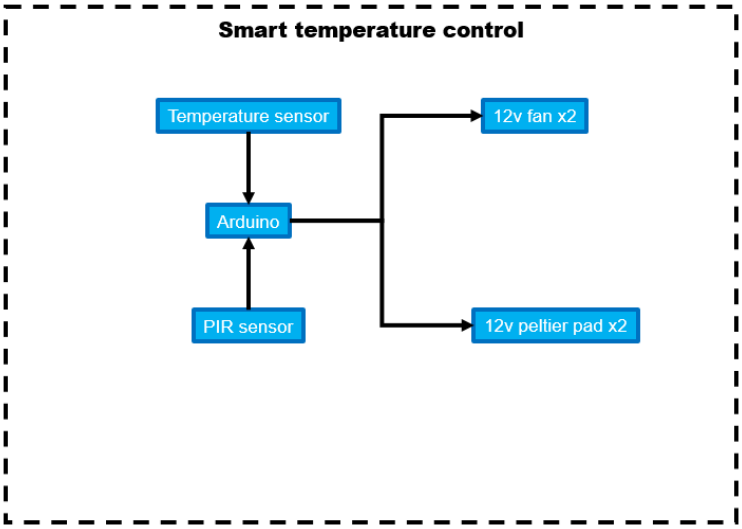
(3)
- Arduino
- Bluetooth receiver
- Smartphone

Sub-System Block Diagram



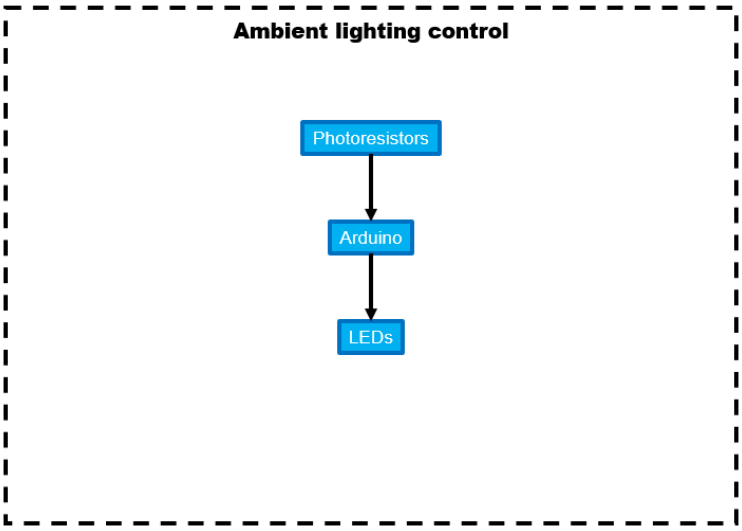
(4)
- Arduino
- Temperature sensor
- PIR sensor
- 12v fan x2
- 12v peltier pad x2

Sub-System Block Diagram



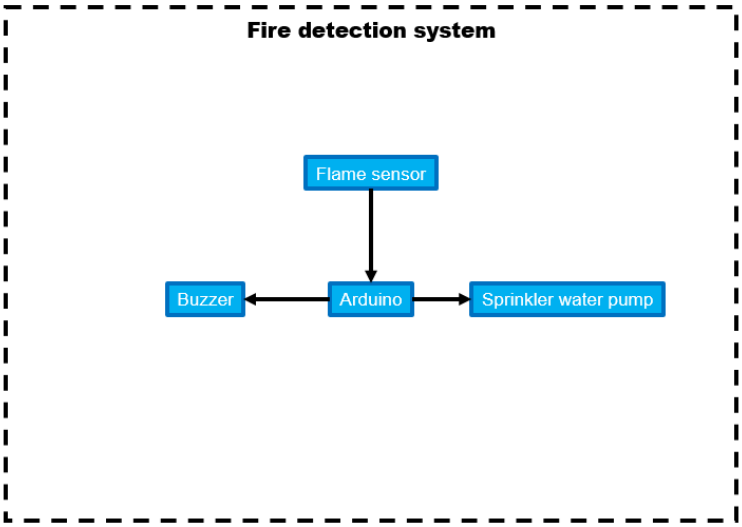
(5)
- Arduino
- Photoresistors
- LEDs

Sub-System Block Diagram



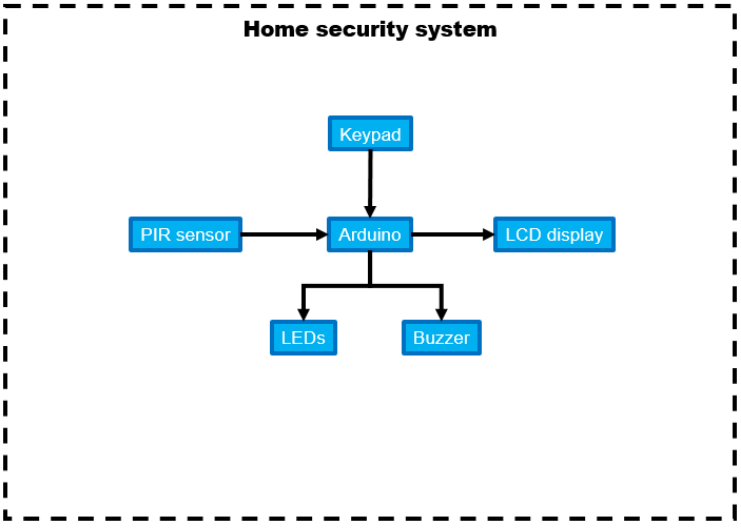
(6)
- Arduino
- Flame sensor
- Buzzer
- Sprinkler water pump

Sub-System Block Diagram



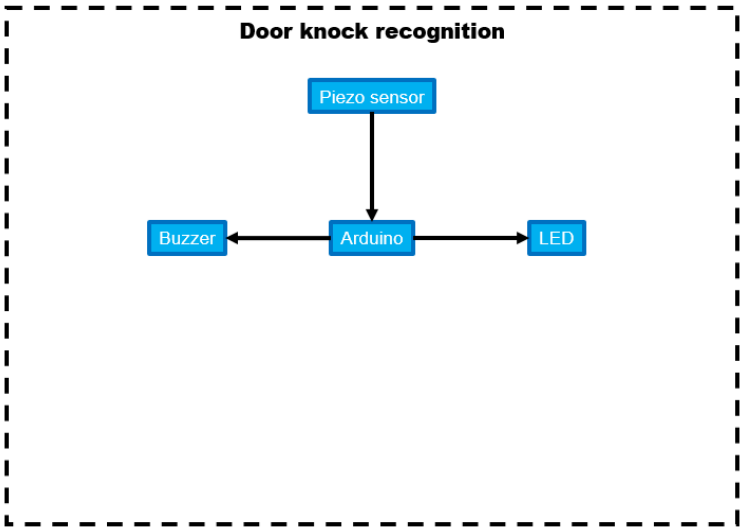
(7)
- Arduino
- LEDs
- Keypad
- LCD display
- PIR sensor
- Buzzer

Sub-System Block Diagram



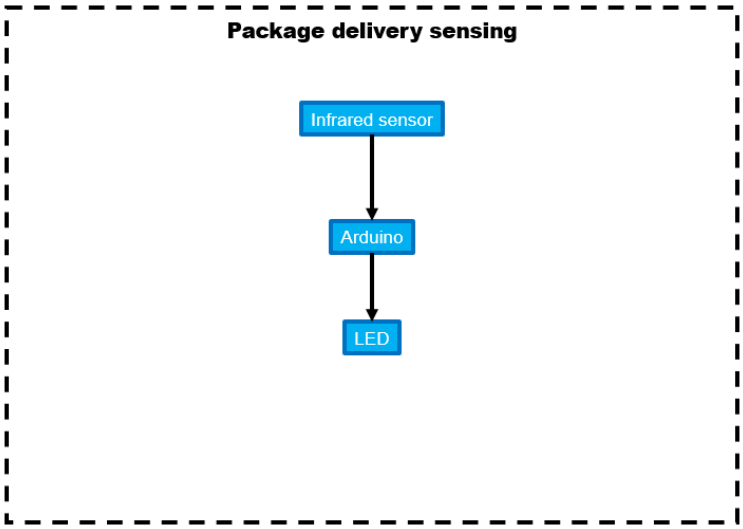
(8)
- Arduino
- Piezo sensor
- Buzzer
- LED

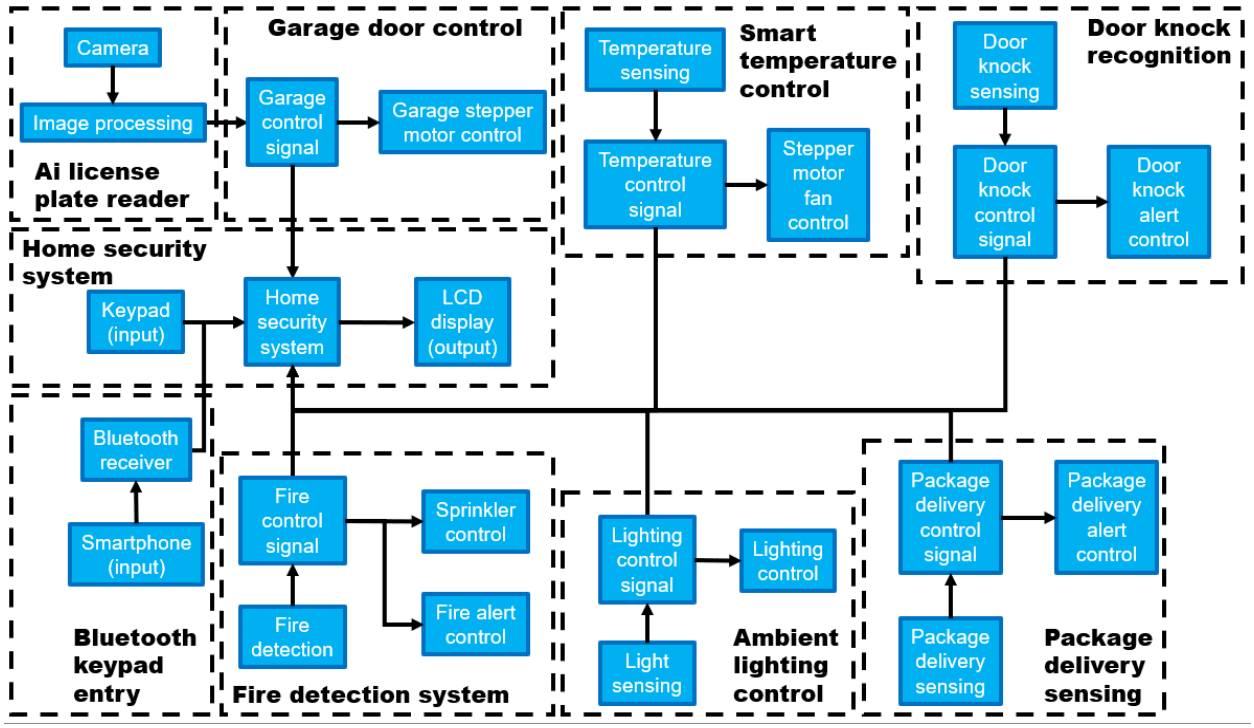
Sub-System Block Diagram



(9)
- Arduino
- Infrared sensor
- LED

Sub-System Block Diagram





Project Block Diagram: https://1drv.ms/p/s!AlgS2t5hwaFUz3Vj4o0SLlhHCh8e?e=ljzpHF

Christopher Arellano:
Home Security System
Bluetooth Keypad Entry
Smart Temperature Control
AI License Plate Reader (Sonar sensor, Arduino & python serial communication, Camera control)

Raunac Bhuiyan:
AI License Plate Reader (Reading the license plate, activate Stepper motor, Arduino & python serial communication)
Garage door controller
Door Knock recognition
Package Detection system

Graham Jabeguero:
Fire Detection System
Ambient Light System
Garage Door Control
AI License Plate Reader (Image Processing)

5.2 * Hardware Architecture

The Project is composed of multiple subsystems that communicate with one another and make up an entire smart home/security system. The central subsystem of the project is the Home Security System

because the LCD display is used to display status messages from the other subsystems. Each subsystem is made onto individual PCBs that connect to a single Arduino Mega to handle the functionality of each subsystem.

Christopher Arellano:

Home Security System

Bluetooth Keypad Entry

Smart Temperature Control

AI License Plate Reader (Sonar sensor, Arduino & python serial communication, Camera control)

Raunac Bhuiyan:

AI License Plate Reader (Reading the license plate, activate Stepper motor, Arduino & python serial communication)

Garage door controller

Door Knock recognition

Package Detection system

Graham Jabeguero:

Fire Detection System

Ambient Light System

Garage Door Control

AI License Plate Reader (Image Processing)

5.3 * Software Architecture (only required if your design includes software)

Each subsystem of the project is made into individual functions to better manage the functionality of the entire system. The individual functions of each subsystem are called on the main function to permit the functionality of multiple subsystems at the same time. The only way that multiple subsystems cannot function at the same time is if they are similar in functionality. For example the Door Knock Recognition system and the Fire Detection system cannot function at the same time because they are similar in functionality.

Christopher Arellano:

Home Security System

Bluetooth Keypad Entry

Smart Temperature Control

AI License Plate Reader (Sonar sensor, Arduino & python serial communication, Camera control)

Raunac Bhuiyan:

AI License Plate Reader (Reading the license plate, activate Stepper motor, Arduino & python serial communication)

Garage door controller

Door Knock recognition

Package Detection system

Graham Jabeguero:

Fire Detection System

Ambient Light System

Garage Door Control

AI License Plate Reader (Image Processing)

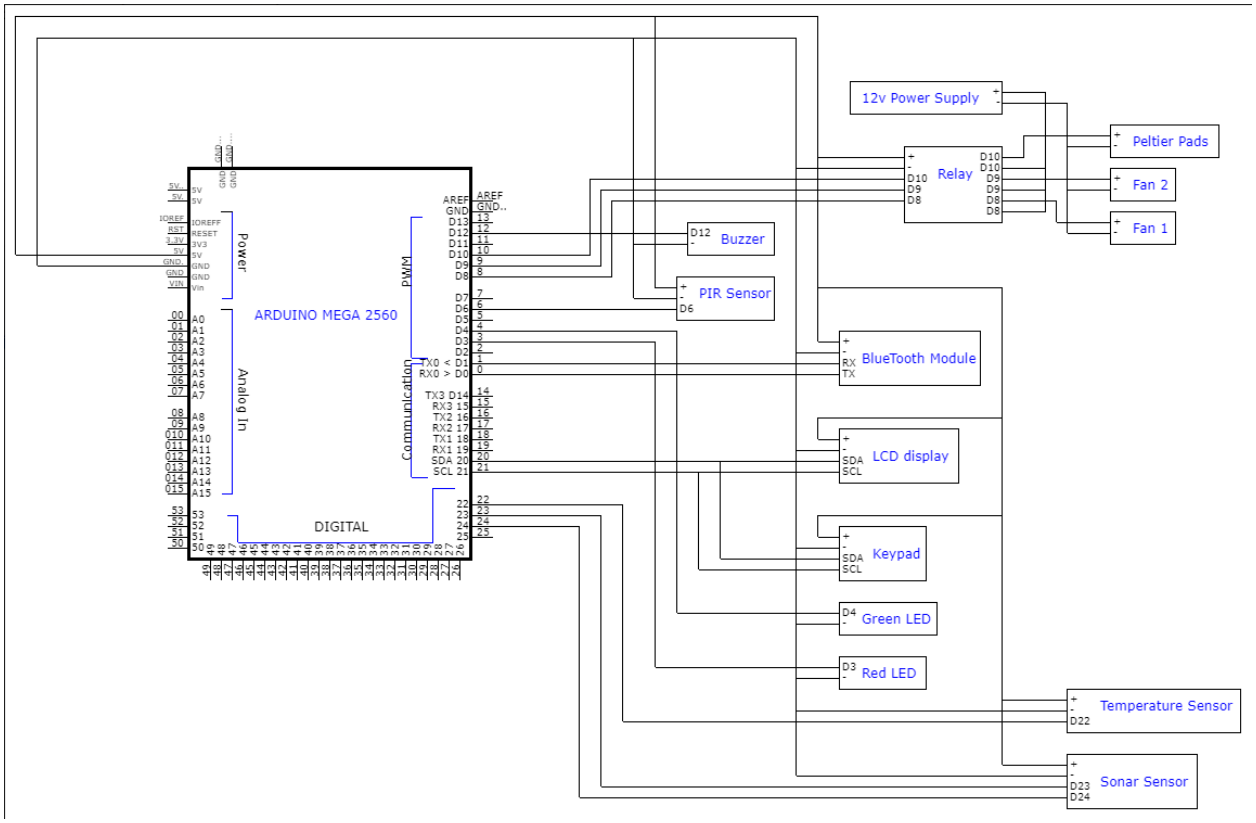
5.4 * Rationale and Alternatives

The project was approached in the way it is because the project is made up of multiple subsystems that make up an entire smart home/security system. In order to evenly break up the workload each group member was assigned certain subsystems or parts of subsystems to work on. The subsystems were assigned to each group member based on what they felt capable of completing within the project's allotted time. Though most if not all subsystems made for the project required all group members to learn new things and pick up skills along the way. Initially the project was designed to include an Atmega chip and the K64f board along with the Arduino in order to allow project group members to work with environments they were familiar with. Yet, when the initial project idea was proposed to the professor we were told that the use of the Atmega chip and K64f board were not a necessary expense and that the project could be done using a single Arduino to handle all functionalities of the project and reduce cost. Additionally the professor advised us to remove the raspberry pi from our original design because it also was not a necessary expense and could be replaced with a pc in order to reduce cost.

6 * Low Level Design

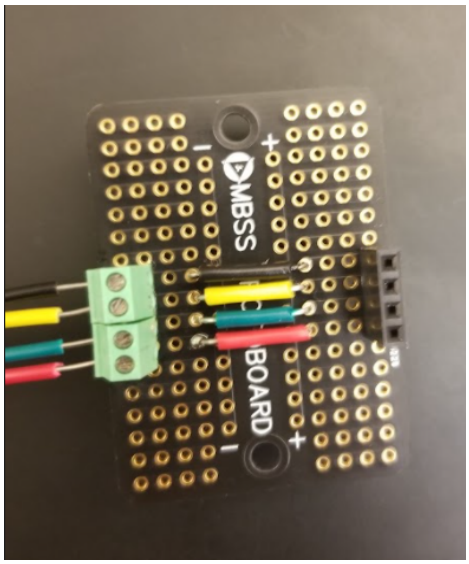
Christopher Arellano:

Project Components' Schematic:

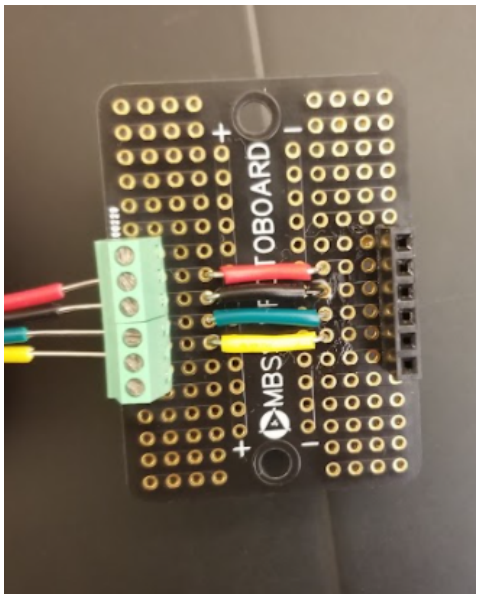


Project Components' PCBs:

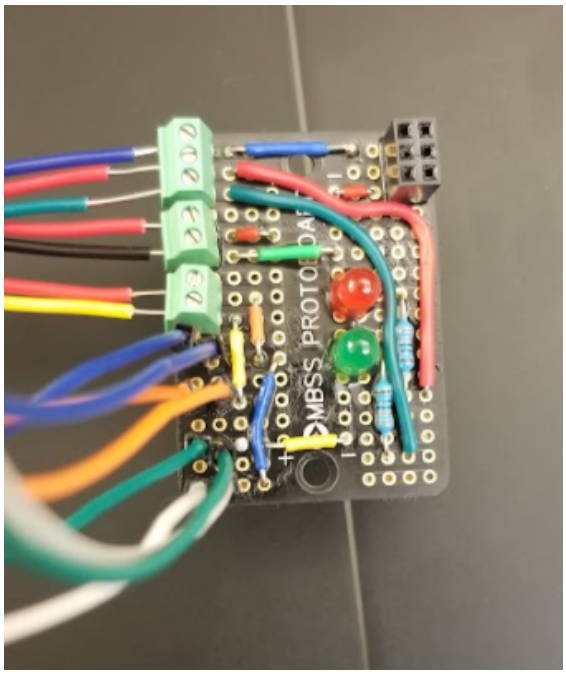
Sonar Sensor PCB:



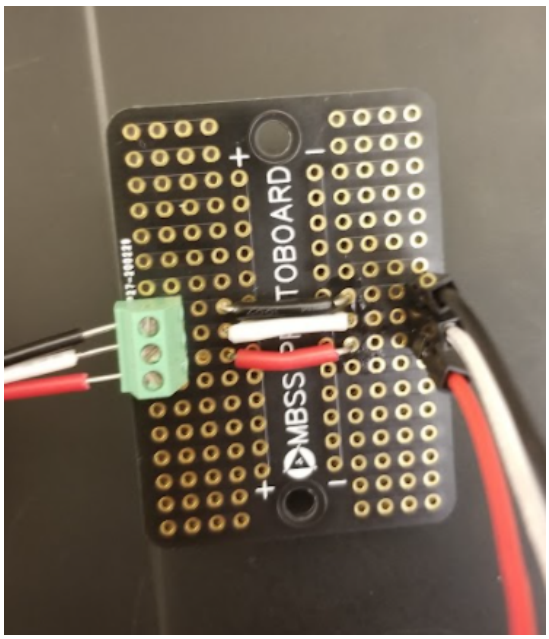
Bluetooth Module PCB:



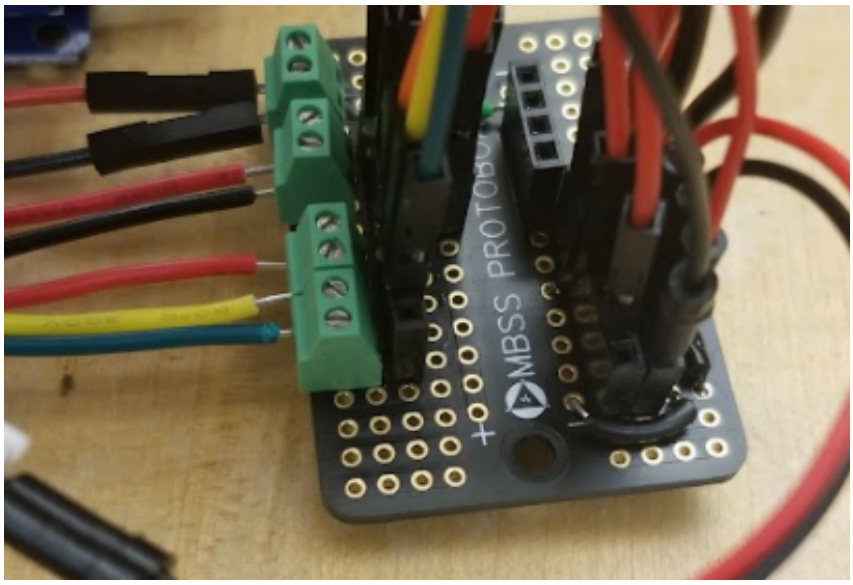
Home Security System PCB:



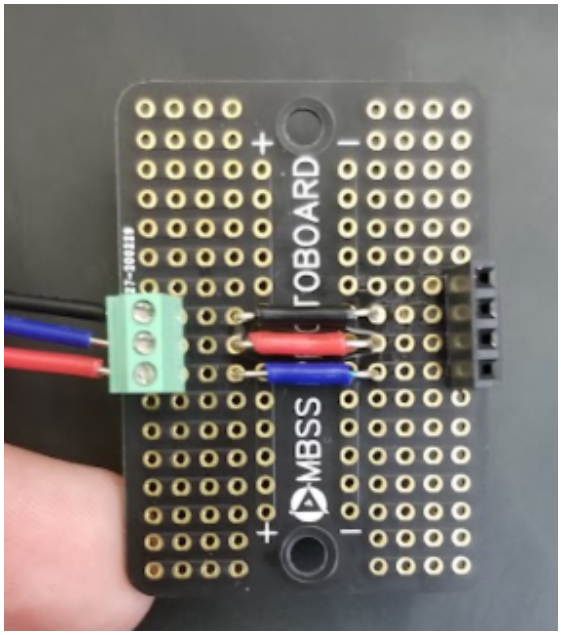
PIR Sensor PCB:



Fan & Peltier Pad PCB:



Temperature Sensor PCB:

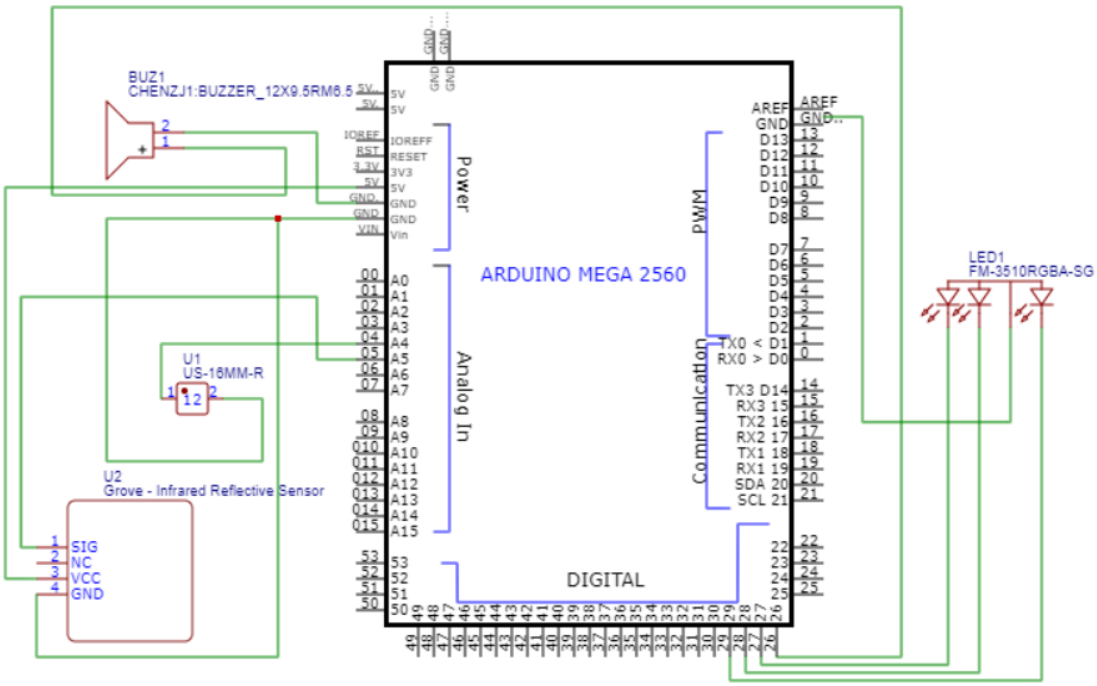


Low level design (Christopher Arellano):

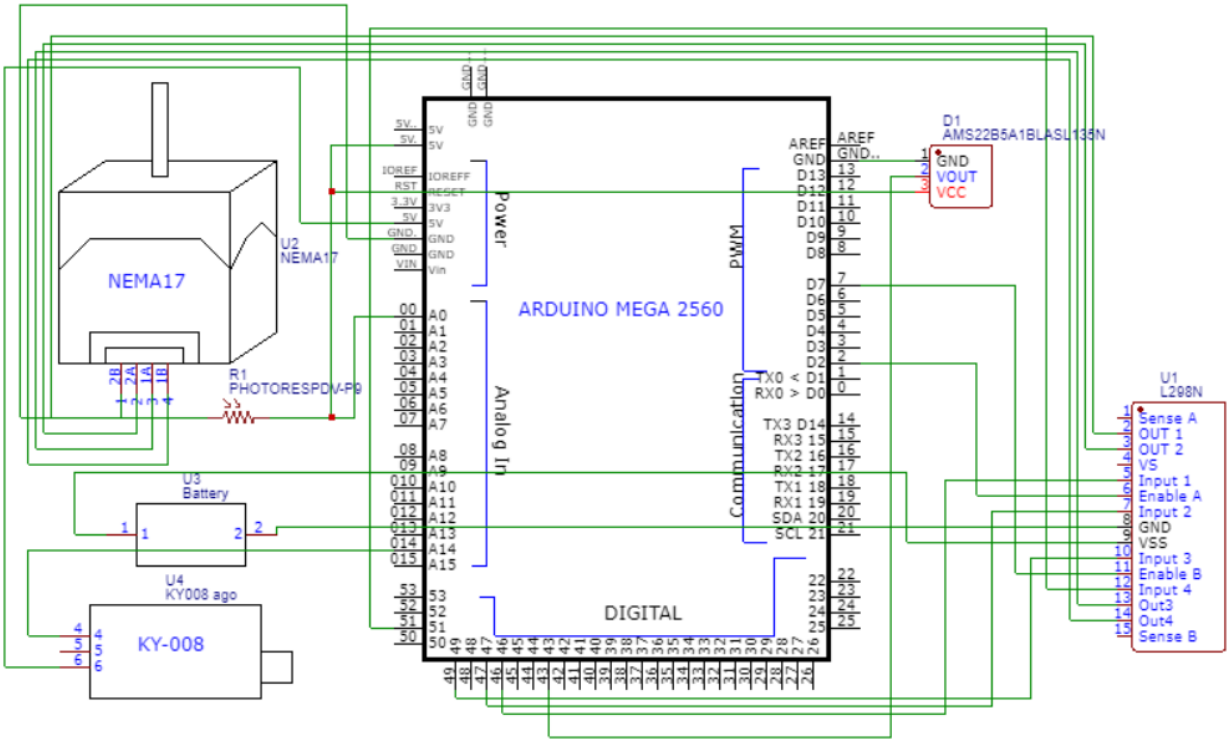
https://drive.google.com/drive/folders/1PmJtPYYOCeEAiPhMb7bRVCeLLxbA352g?usp=sharing

Raunac Bhuiyan:

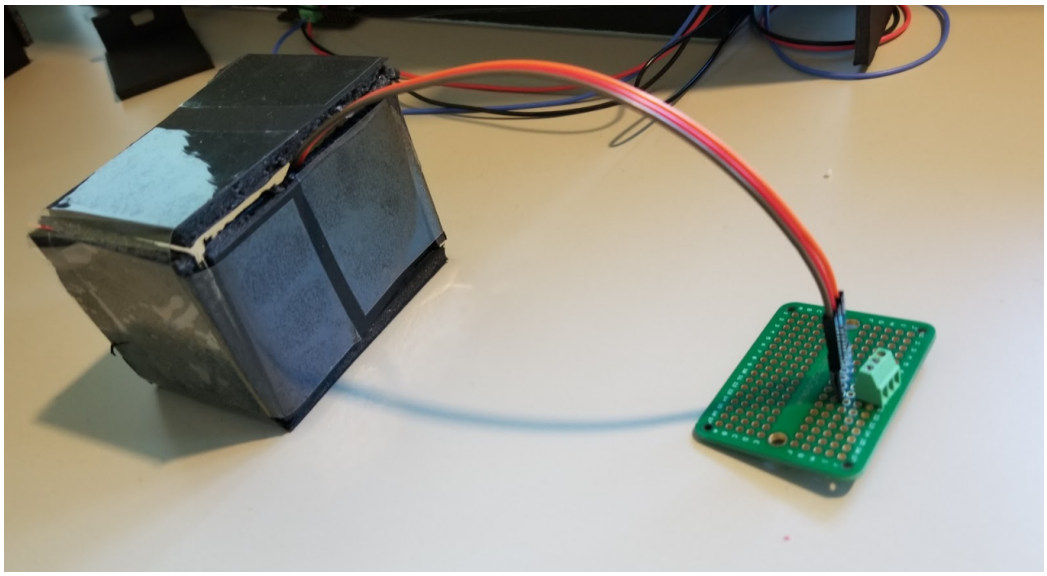
Door Knock System and Package Detection System SCHEMATICS:



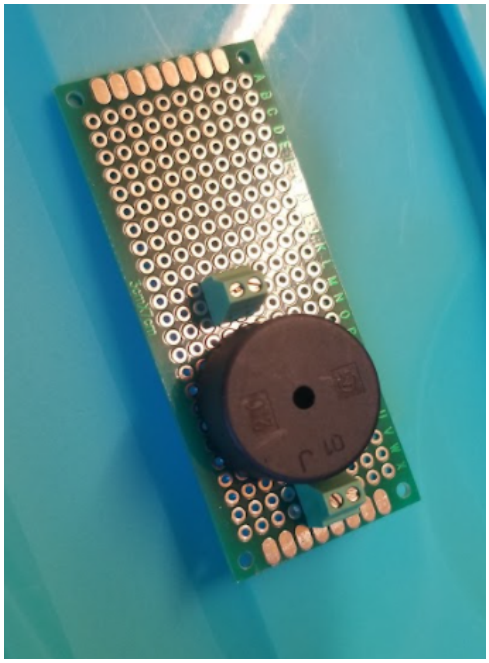
The Garage Door Control System SCHEMATIC:

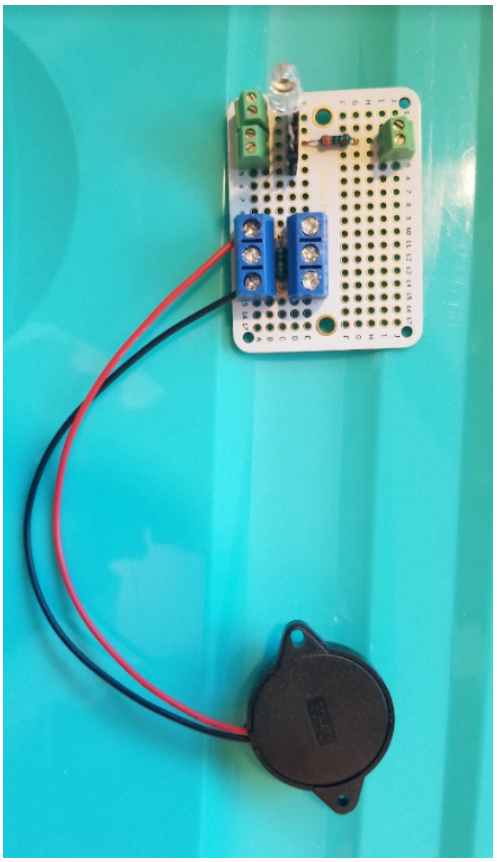


PCB design of Package Delivery System:

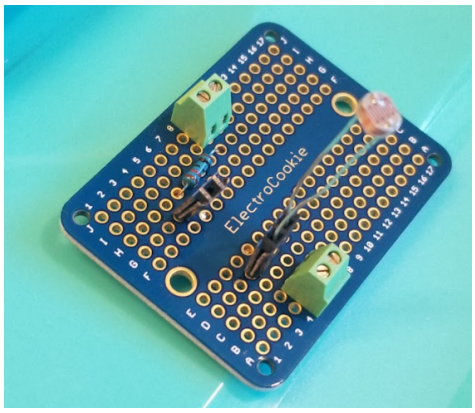


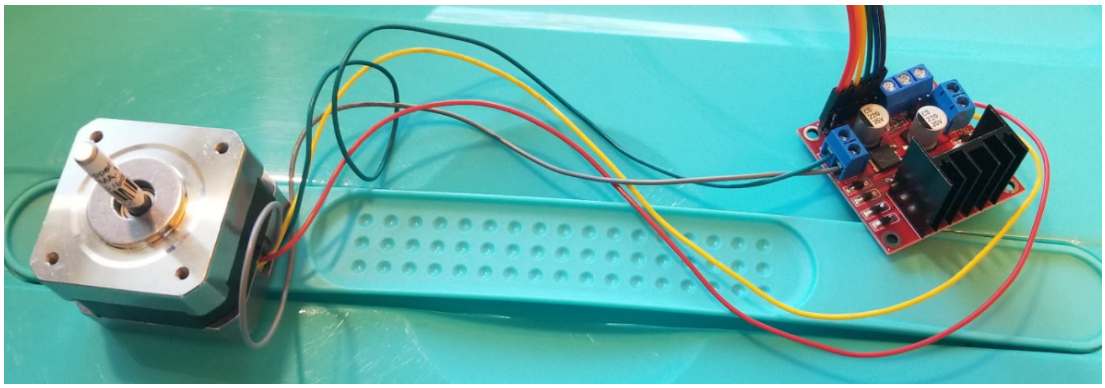
PCB design of Package Delivery System + Passive Buzzer:





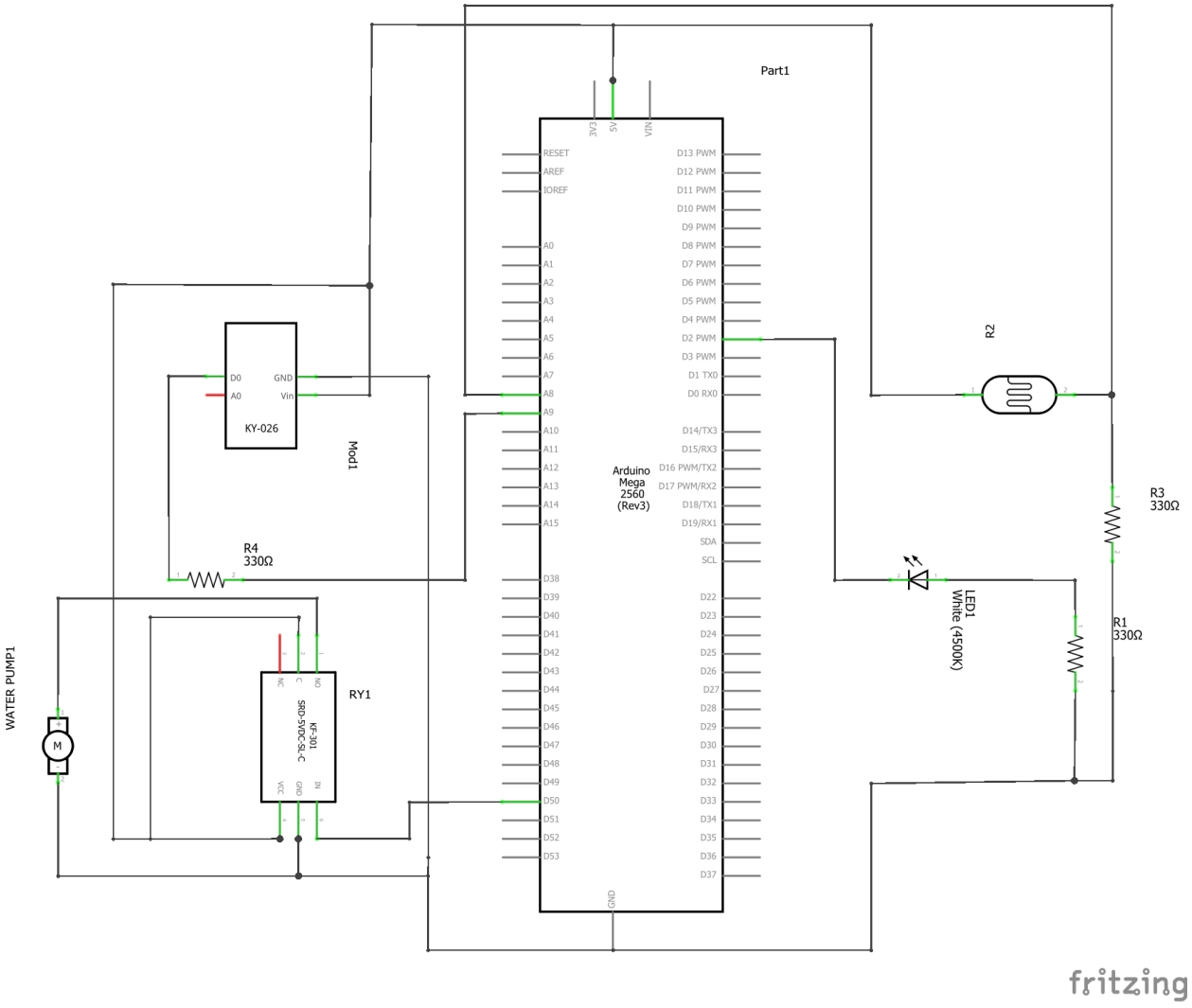
PCB Design of the Garage Door Controller System (photoresistor + stepper motor w/motor driver):





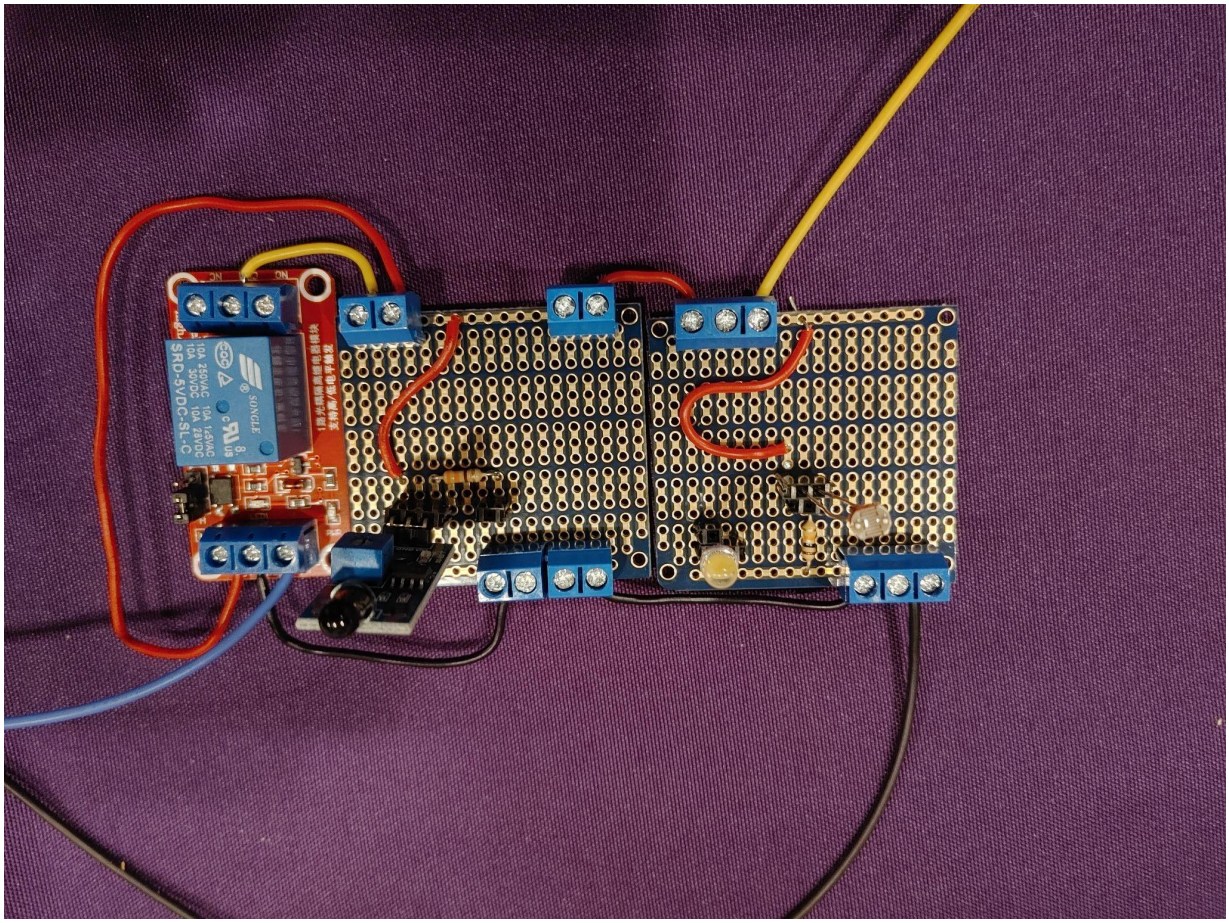
Graham Jabeguero:

Fire Detection System (Left) AND Ambient Lighting Control (Right):



Project Components' PCB:

Fire Detection System (Left) AND Ambient Lighting Control (Right)



7 * Technical Problem Solving

Document all the technical difficulties you encountered

Christopher Arellano:

Arduino pin usage - reduced by using I2C for keypad and LCD display

Arduino & python communication issues - resolved by using serial communication

Keypad and Bluetooth Keypad issues - resolved by making separate input arrays

PIR sensor issues - resolved by changing PIR sensor mode to only trigger with motion

Sonar sensor issues - resolved by decreasing range

Temperature control issues - resolved by using larger fans and peltier pads

Raunac Bhuiyan:

Stepper motor “lags” due to a low power supply. In other words, the stepper motor appears to have a weaker torque then intended - somewhat resolved with a 9V battery

The webcam/camera does not read every character of the license plate - resolved by adding several conditions. For example, if the plate is “KAME05”, and the code reads it as “KAMOS” it will still be valid.

Door knock sensor “interrupted” the other systems - resolved by lowering threshold + making code changes (i.e. no ‘while loops’)

Package Detection System issue - resolved by making the input to analog instead of digital + code changes

Graham Jabeguero:

Fire Detection System issues - exact time to turn on sprinkler was inconsistent when merging all subsystems. Slightly resolved by drastically lowering the counter.

All:

Individual project component merge issues - resolved by making code changes

Total project component merge issues - resolved by making code changes

8 * Test Plan

8.1 * Test Design

Christopher Arellano:

Keypad & Bluetooth Keypad Input Test:

Objective: Determine if input is received correctly.

Setup: Use Keypad and Bluetooth Keypad to test input.

Procedure: Try multiple input combinations to test if input is received correctly.

Expected Results: Receive the same input from Keypad and Bluetooth Keypad.

Temperature Sensor Test:

Objective: Determine if temperature read is correct.

Setup: Use fans and peltier pads to vary temperature.

Procedure: Change temperature using fans and peltier pads to test if temperature is read correctly.

Expected Results: Temperature reads to be high when warm and low when cool.

PIR Sensor Test:

Objective: Determine if correctly triggered with motion.

Setup: Use PIR Sensor to detect motion.

Procedure: Vary motion to test for correct triggers.

Expected Results: PIR Sensor to trigger only with detected motion.

Sonar Sensor Test:

Objective: Determine if correctly triggered by object detection within range.

Setup: Use Sonar Sensor to detect object within range.

Procedure: Vary ranges of object to test for correct triggers.

Expected Results: Sonar Sensor to trigger only with objects detected within range.

Raunac Bhuiyan:

Ai License Plate Recognition:

Test case 1: Read characters from a California license plate after taking a picture. Did not recognize/read a single character, instead just read a blank string (i.e. a license plate with no characters).

Test case 2: Read characters from an Indian license plate after taking a picture. The license plate was recognized; that said, sometimes it incorrectly misreads a character. For example, it will read “5” as “S.”

Garage Door Control:

Test case 1: Activate stepper motor by exposing the photoresistor to a certain amount of light. The threshold (i.e. the amount needed to activate the stepper motor) was adjusted because of the environment. In other words, the lighting in the lab rooms accidentally triggers the stepper motor when not intended; therefore, it was adjusted.

Test case 2: Activate stepper motor by having the python code send a signal to the arduino. In this case, a license plate must be read first. Functions the same way as test case 1, except, it is no longer dependent on the amount of light.

Test case 3: Activate stepper motor and laser sensor. While the stepper motor is running, the laser will activate as well. Furthermore, if the laser sensor detects an object, the motor will pause, and will only continue if the laser sensor no longer detects anything.

Package Delivery Detection:

Test case 1: As long as the Infrared sensor can detect an object within 2.5 centimeters, the LCD screen will display “Object detected.” However, to prevent the LCD screen from overloading (i.e. printing the same message while the IR sensor detects something), the code was adjusted and rewritten to ensure it will only print the message once after detecting an object. Afterwards, if the object is removed from the range, it will display “Object removed.”

Door Knock Recognition:

Test case 1: Knock gently on the table, so the piezo sensor can detect it. It activates the passive buzzer and RGB (red, green, blue) LED.

Test case 2: Threshold is increased, so now it requires a little bit more force (i.e. harder knock on the table) to activate the buzzer and RGB LED.

Graham Jabeguero:

Fire Detection System Test:

Objective:

Correctly sense nearby flames and extinguish them using sprinklers. Also to display a message using the serial monitor in Arduino IDE ONLY when the sprinkler turned on and off. This tests the subsystem for the Fire Detection System. The technical design objective measures the flame sensor and the water pump.

Setup:

Used a small cardboard box and oriented the flame sensor inside at the right wall and placed the sprinkler end in the ceiling. All other components (water pump, resistor, breadboard, an Arduino) were placed on the back of the box.

Procedure:

Light a fire using a lighter inside of the box and wait until the flame sensor detects it and turns on the sprinkler to extinguish it. Also looked at the serial monitor to see if it displayed the correct phrases. Did this 5 times to gain more accuracy.

Expected Results:

Turn on the sprinkler when the system senses fire for about 3 seconds then turns off after it stops sensing fire for about 2 seconds. Displays a message in the serial monitor when the sprinkler turns on/off.

Ambient Lighting Control Test:

Objective:

Output the correct LED brightness based on the value received from the photoresistor. This tests the subsystem for the Ambient Lighting Control. The technical design objective measures the photoresistor and the LED.

Setup:

Used the same small cardboard box from the Fire Detection System Test. Placed the LED on the ceiling and placed the photoresistor outside of the side wall. All other components (breadboard and Arduino) were placed behind the box.

Procedure:

Tested with varying ranges of light using a flashlight to output the different levels of brightness of the LED inside.

Expected Results:

Turn off the LED when shining light on the photoresistor and gradually turn on the LED when dimming the light until very dark.

Garage Door Control Test (w/ K64F and Arduino):

Objective:

Determine if sensors are triggered correctly and if Arduino can communicate with K64F to turn on the stepper motor for the garage door.

Setup:

Built a mini garage out of cardboard and placed the stepper motor above the door. The small door is connected to the stepper motor with a string so that it could be pulled up and down depending on what phase the system is in. A laser is placed in parallel with the door in front. The infrared sensor is placed perpendicular to the door at the front left side. The photoresistor is placed inside the garage. Communication between Arduino and K64F via I2C was established.

Procedure:

Use a flashlight as an interpretation of a car going into and out of the garage. Testing the sensors to go through each of the garage phases while also testing out the laser sensor to test if the garage is stopped by this.

Expected Results:

The garage can successfully open the garage when the car is in close proximity and close the garage when the car is inside (turning on it's headlights and then turning off). After this, the garage should

open when it detects the car's headlights and close when the car is no longer in close proximity. Also, the laser should also stop the garage from functioning if there is anything blocking the laser.

8.2 * Bug Tracking

Bugs are to be handled as soon as they are discovered. Bugs are to be handled by the project group member who is responsible for the component(s) containing the bug(s).

8.3 * Quality Control

Every time a major change is made to the code of the merged project components, a new code file is created. Every code file created is considered a new version of the project in order to be able to revert changes if needed. If only small changes are made within the project code, then no new code file is created. Code files are used to keep track of major changes and to keep track of what currently works and what needs to be fixed.

8.4 * Identification of critical components

Christopher Arellano:

I2C: To correctly display on LCD and read from keypad.

Bluetooth: To be able to use bluetooth keypad.

Temperature Sensor: To be able to correctly adjust temperature.

PIR Sensor: To be able to trigger auto temp and alarm with motion.

Sonar Sensor: To be able to trigger message from arduino to python.

Raunac Bhuiyan:

Stepper motor: requires 9V battery and L298N motor driver to run the motor.

Laser module: requires the arduino 5V and GND port

HiLetgo IR sensor: requires arduino 5V, GND, and an analog port

Photoresitor: requires the arduino 5V, and an analog port

Piezo sensor: requires an arduino analog port

Passive Buzzer: requires arduino digital port

RGB (red, green, blue) LED: requires one resistor and three arduino digital ports

Graham Jabeguero:

Photoresistor (Ambient Lighting Control): To correctly give the signal to output the corresponding brightness to LED.

Flame Sensor: To be able to give a signal to the relay to turn on.

Relay Module: To be able to turn on the water pump for extinguishing any fire.

8.5 * Items Not Tested by the Experiments

N/A

9 * Test Report

Christopher Arellano:

Keypad & Bluetooth Keypad Input Test Report:

Test Results: Correct input read if Ai License Plate Reader is not used.

Comparison with Expected Results: Received the same input from Keypad and Bluetooth Keypad if Ai License Plate Reader is not used.

Analysis of Test results: character ‘k’ is input into the input array when the correct license plate is read by Ai License Plate Reader.

Corrective Actions Taken: press ‘*’ on either keypad or bluetooth keypad to clear the input array before trying to input after the Ai License Plate Reader is used.

Temperature Sensor Test Report:

Test Results: Temperature reads are within + or - 2 degrees fahrenheit from the actual temperature.

Comparison with Expected Results: Temperature reads are high when warm and low when cool.

Analysis of Test results: Temperature can only be adjusted to approximately + or - 5 degrees fahrenheit from 75 degrees fahrenheit.

Corrective Actions Taken: Set user temperature selection from only 70 degrees fahrenheit to 80 degrees fahrenheit.

PIR Sensor Test Report:

Test Results: PIR Sensor is triggered by any motion detected.

Comparison with Expected Results: PIR is only triggered by detected motion.

Analysis of Test results: Motion within the PIR’s view triggers auto temp and/or alarm.

Corrective Actions Taken: Allowed user to select always active for the temperature control, gave the user 7 seconds before PIR detects motion after setting the alarm.

Sonar Sensor Test Report:

Test Results: Sonar Sensor is triggered by any object within range.

Comparison with Expected Results: Sonar Sensor is only triggered by objects within range.

Analysis of Test results: Any slight back and forth movement of an object within range may trigger the Sonar Sensor.

Corrective Actions Taken: Shortened the range to detect an object.

Raunac Bhuiyan:

Ai License Plate Recognition:

Test case 1: Read characters from a California license plate after taking a picture. Did not recognize/read a single character, instead just read a blank string (i.e. a license plate with no characters).

Test case 2: Read characters from an Indian license plate after taking a picture. The license plate was recognized; that said, sometimes it incorrectly misreads a character. For example, it will read "5" as "S."

Garage Door Control:

Test case 1: Activate stepper motor by exposing the photoresistor to a certain amount of light. The threshold (i.e. the amount needed to activate the stepper motor) was adjusted because of the environment. In other words, the lighting in the lab rooms accidentally triggers the stepper motor when not intended; therefore, it was adjusted.

Test case 2: Activate stepper motor by having the python code send a signal to the arduino. In this case, a license plate must be read first. Functions the same way as test case 1, except, it is no longer dependent on the amount of light.

Test case 3: Activate stepper motor and laser sensor. While the stepper motor is running, the laser will activate as well. Furthermore, if the laser sensor detects an object, the motor will pause, and will only continue if the laser sensor no longer detects anything.

Package Delivery Detection:

Test case 1: As long as the Infrared sensor can detect an object within 2.5 centimeters, the LCD screen will display "Object detected." However, to prevent the LCD screen from overloading (i.e. printing the same message while the IR sensor detects something), the code was adjusted and rewritten to ensure it will only print the message once after detecting an object. Afterwards, if the object is removed from the range, it will display "Object removed."

Door Knock Recognition:

Test case 1: Knock gently on the table, so the piezo sensor can detect it. It activates the passive buzzer and RGB (red, green, blue) LED.

Test case 2: Threshold is increased, so now it requires a little bit more force (i.e. harder knock on the table) to activate the buzzer and RGB LED.

Graham Jabeguero:

Fire Detection System Test Report:

Test Results: Flame sensor triggers correctly when a flame is nearby, activating the sprinkler and correctly displaying what has happened in the serial monitor.

Comparison with Expected Results: There seems to be a substantial delay (varying between 5 - 10 seconds) when activating the sprinklers. Also could not use the serial monitor due to interference with our Python code.

Analysis of Test Results: This is due to the fact that there are many other subsystems also running as well which causes the delay.

Corrective Actions Taken: Reduced the counter so that the delay is reduced substantially. Also used the LCD screen instead to display the messages of this system

Ambient Lighting Control Test Report:

Test Results: Correctly outputs the correct brightness based on the photoresistor. When integrating however, I could only get 2 settings to be outputted.

Comparison with Expected Results: Should output 5 different brightness to the LEDs.

Analysis of Test Results: Due to not using an Arduino Mega pin that did not have pulse width modulation compatible.

Corrective Actions Taken: Had to alter the configuration of Arduino Pins so that the LED pin could use pulse width modulation.

Garage Door Control (w/ K64F and Arduino) Test Report:

Test Results: Communication between Arduino and K64F worked as expected. Sensors correctly triggered the different phases the systems went through: opening the garage to let the car in, making sure the car was in the garage to close it, opening the garage when the user wanted to drive out of the garage, closing the garage when the car fully left the garage. Laser also stopped the garage functions when it was blocked.

Comparison with Expected Results: No difference between expected and actual result.

Analysis of Test Results: It was deemed very insufficient if we used both the Arduino and K64F. The K64F was used to only control the stepper motor and we, as a group, did not know that the stepper motor could be controlled by the Arduino.

Corrective Actions Taken: Our group decided to take out the K64F and only use the Arduino to not only control the sensors but also the stepper motor.

10 * Conclusion and Future Work

10.1 * Conclusion

Christopher Arellano:
Project Components:
Home Security System
Bluetooth Keypad Entry
Smart Temperature Control
AI License Plate Reader (Sonar sensor, Arduino & python serial communication, Camera control)

Conclusion: The finished work I have contributed to the project has met the overall project goals and the quantitative technical design objectives. All of my project components work as initially intended and function properly with my partner's project components. By doing this project I have gained a lot of technical knowledge and skill and have gone through a lot of professional and personal growth. This project has taught me that in order to be an effective engineer one must be able to adjust to many changes and learn how to pick up new skills along the way. Additionally, one must be open to learning new things in order to complete one's tasks. This project has made me pick up skills on how to use sensors, how to code an arduino, how to code in python, how to make an app, how to use bluetooth for an arduino, how to use I2C to reduce pin connections on an arduino for both Keypad and LCD display, etc. The project has made me gain technical knowledge on how to better debug code and circuitry, how to test the logic of my components, how to check for faulty circuit design, how to check for faulty hardware, etc. The project has made me grow professionally by making me do things that I have not learned from school and that I had to learn on my own. One example of something I had to learn on my own was how to design my PCBs and solder my PCBs. Previously I had never soldered or used PCBs for any class, all of my previous classes used only breadboards. The project has made me grow personally by making me work in a group on the project and learn how to work together with my partners even though we each think differently, have different schedules, have personal things to deal with, have other classes to deal with, etc.

Future Work: One thing that I would consider to better the project is to make every project component wireless and have the components communicate through a network instead, in order to remove the use of wires from the project components to the arduino. Another thing that I would consider to better the project is to build a solid house that has everything connected in order to avoid dealing with the hassle of assembling the project before a demo and disassembling the project after a demo. Additionally I would consider making housings for each of the project components to make the circuits not visible to the user and to avoid damaging the circuits. Lastly, I would consider getting a newer bluetooth module to permit the arduino to send and receive signals from both a pc and smartphone at the same time. By using a newer bluetooth module serial communication via usb from arduino to python could be replaced with serial communication via bluetooth from arduino to python.

Raunac Bhuiyan:
Garage Door Control system (w/ Arduino ONLY)
Ai License Plate Recognition (reading the characters from image, send signal via serial communication to arduino IDE)
Package Delivery Detection system
Door Knock Recognition system

Conclusion/Future Work:
All of the systems I have contributed to (listed above) meet the project's requirements. The garage door system utilizes both the openCV code and arduino. Therefore, I had to learn how to use serial communication in order to make the two codes work together. This was critical since otherwise, the garage door would not open. The Door Knock recognition utilizes the piezo sensor and passive buzzer. Moreover, after sensing a knock, it will trigger the passive buzzer and LED to activate. The package delivery system utilizes the analog ports to ensure the IR sensor is able to detect objects within a certain range. As for future implementations, I would use an infrared sensor that has a wider range. In other words, the sensor would be able to detect objects not limited to 2.5 centimeters. As for the garage door system, I would consider using a 12V DC power supply because a 9V battery is very limited. In other words, batteries have a limited supply of energy, while power supplies do not.

Graham Jabeguero:
Project Components:
Fire Detection System
Ambient Lighting Control
Garage Door Control (w/ K64F AND Arduino)
AI License Plate Reader (Image Processing)

The achieved work I contributed was met by the overall project goals and the technical objectives. All project components worked individually and together in the entire system to an expected degree. This project forced me to learn many technical skills such as learning how to solder, designing one’s own PCB and making it by yourself by soldering. Another technical skill would be to effectively debug one’s and other peoples systems for integration. I’ve never fully coded in Arduino and in Python and this project showed me how to do this. I found a way to activate modules that needed higher voltage than what the microchip can output, like a water pump that needed 12V, by using a relay module. The need to communicate between microchips using I2C (although this knowledge was not implemented in the final design due to being insufficient). As for professional and personal growth, I learned how to get along with others as a group in order to effectively reach the project goals. I also learned to respect other people’s time and thoughts which I did not know would be an essential skill I needed as an engineer.

Future Work: If we had more time in this project, I would consider using the buzzer in the Security System for the Fire Detection System so that the user knows when there is a fire without looking at the LCD screen. I would also consider using multiple LEDs instead of one for the Ambient Lighting Control by the use of transistors or shift registers.

10.2 * Acknowledgement

(Professor) Roman Chomko:

Advising us to remove unnecessary MCUs in order to avoid unnecessary expenses and reduce cost and advising us on how to better our project.

(TA) Maliha Tasnim:

Advising us on how to make alternative designs, helping us debug, helping us pinpoint issues with our project, advising us how to improve our project and guiding us on what materials to use for our project.

“Techno Kidzo” (YouTuber):

Demonstrated how to set up and use the openCV license plate reader code.

https://youtu.be/8tQULVb06Zo

This is also his code from gitHub: https://github.com/creativekids11/License-Plate-Recognition

Justin also known as “Tabletop Robotics” (YouTuber):
This person provided the instructions on how to use serial communication between python IDE and arduino IDE.
https://sites.google.com/view/tabletop-robotics/coding-and-how-to/pythonarduino-communication?authuser=0

11 * References

[1] Justin, "Python/Arduino Communication," 2019. Available: Tabletop Robotics, https://sites.google.com/view/tabletop-robotics/coding-and-how-to/pythonarduino-communication?authuser=0. [Accessed: 7-Feb-2022].

[2] T. Kidzo, "License plate recognition with OCR using OpenCV Python | Full Tutorial | OpenCv Python," *YouTube,* 13-Dec-2021. [Online]. Available: https://www.youtube.com/watch?v=8tQULVb06Zo. [Accessed: 14-Jan-2022].

[3] creativekids11, "Creativekids11/license-plate-recognition," *GitHub*, 1AD. [Online]. Available: https://github.com/creativekids11/License-Plate-Recognition. [Accessed: 11-Mar-2022].

[4] HSmart HD, *5.0MP Vechile License Plate Recognition LPR ANPR IPC 5MP SONY 335 Camera ONVIF Outdoor Waterproof HD 6-22mm Lens For parking lot*. [Online]. Available: https://www.aliexpress.com/item/4000921232021.html?_randl_currency=USD&_randl_shipto=US&src=google&aff_fcid=6a3f5e5479b34678ab3811f6203b7a04-1646790848109-01346-UneMJZVf&aff_fsk=UneMJZVf&aff_platform=aaf&sk=UneMJZVf&aff_trace_key=6a3f5e5479b34678ab3811f6203b7a04-1646790848109-01346-UneMJZVf&terminal_id=a024e7361bd0468985ec67cb57f884c3&afSmartRedirect=y. [Accessed: 17-Feb-2022].

12 * Appendices

* **Appendix A**: Parts List
Hardware used:
Arduino Mega
Laptop
2 Photoresistors
LEDs
12v fan x2
12v peltier pad x2
Temperature sensor
Bluetooth receiver
Smart phone
Camera/laptop webcam
Sonar sensor
Ky008 Laser/Sensor module
Keypad
LCD display
HiLetgo Infrared sensor
PIR sensor
Sprinkler water pump
Passive and Active Buzzers
Flame sensor
Piezo sensor
Stepper motor
L298N motor driver
9V battery

* **Appendix B**: Equipment List
Hardware used:
Arduino
PC

Software Used:
MIT App Inventor
Arduino IDE
Pycharm IDE

* **Appendix C**: Software List
Software Used:
MIT App Inventor
Arduino IDE
Pycharm IDE

Code Link:
https://drive.google.com/drive/folders/1D9LtfbukFU6IwMimplIv2GoSg-CRbf3M?usp=sharing