EE175AB Final Report

Autonomous Home System/Garage Sensor

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

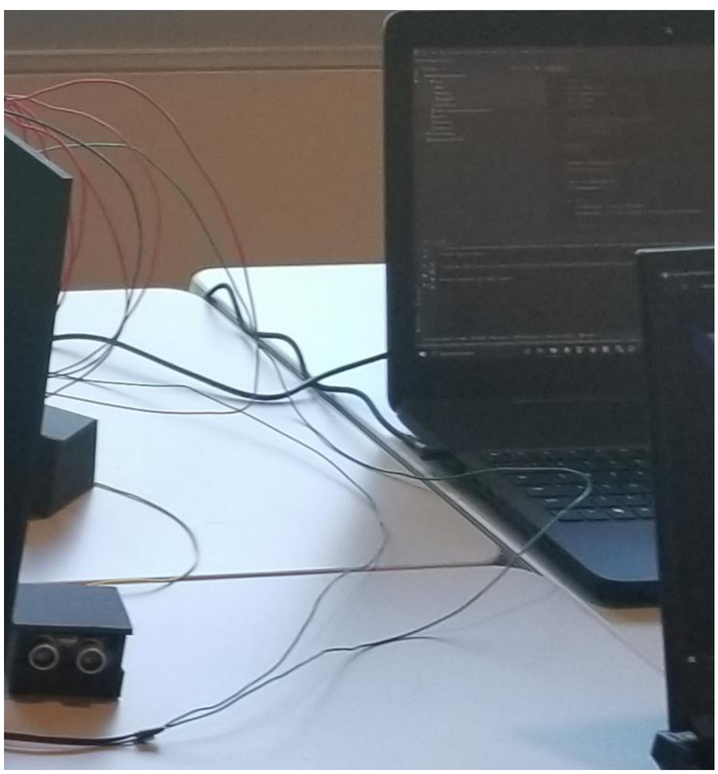
Project Team Member(s)	Christopher Arellano, Graham Jabeguero, Raunac Bhuiyan
Date Submitted	06/9/2022
Section Professor	Roman Chomko (Professor), Maliha Tasnim (TA)
Revision	Revision 7.0
URL of Project YouTube Videos, Wiki/Webpage	Code Link: https://drive.google.com/drive/folders/1D9LtfbukFU6IwMimplIv2GoSg-CRbf3M?usp=sharing Video Link: https://youtu.be/j1qS2g2dSCA
Permanent Emails of all team members	carel009@ucr.edu, gjabe001@ucr.edu, rbhui001@ucr.edu

Summary

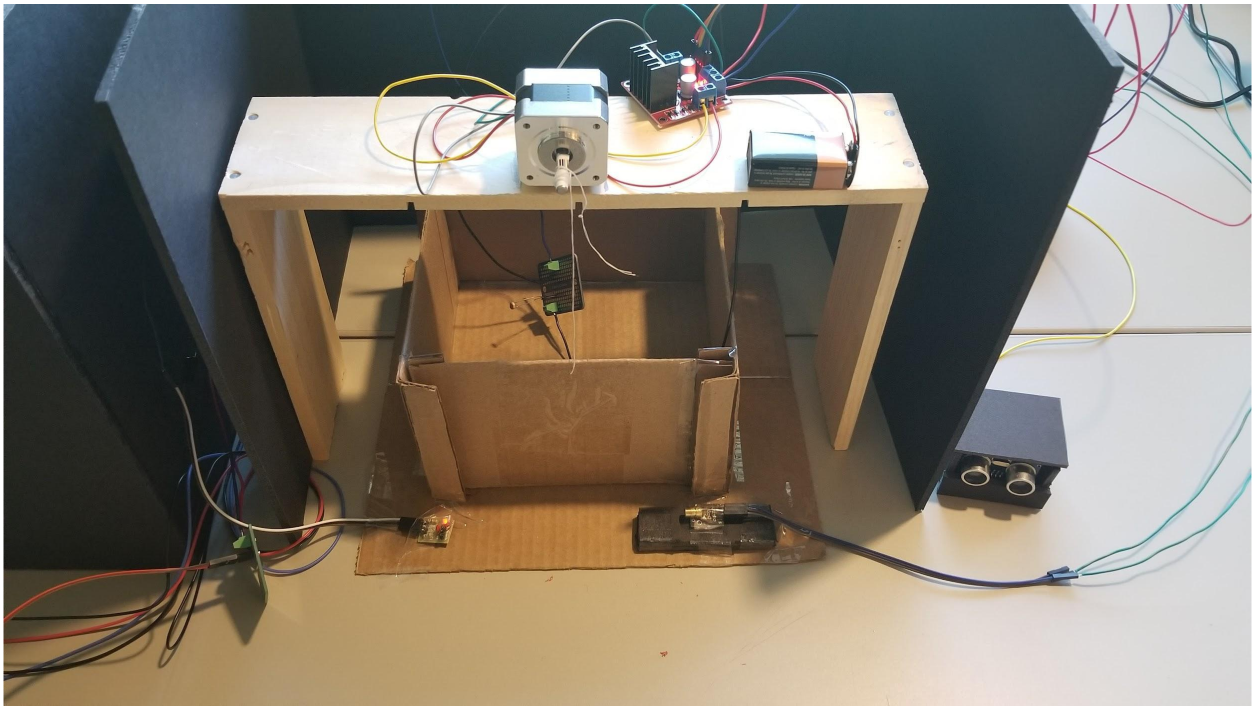
This report presents the Autonomous Home System/Garage Sensor (AHS). The project is composed of nine subsystems for smart homes/security systems. The subsystems are shown from page two to six.

Autonomous Home System/Garage Sensor

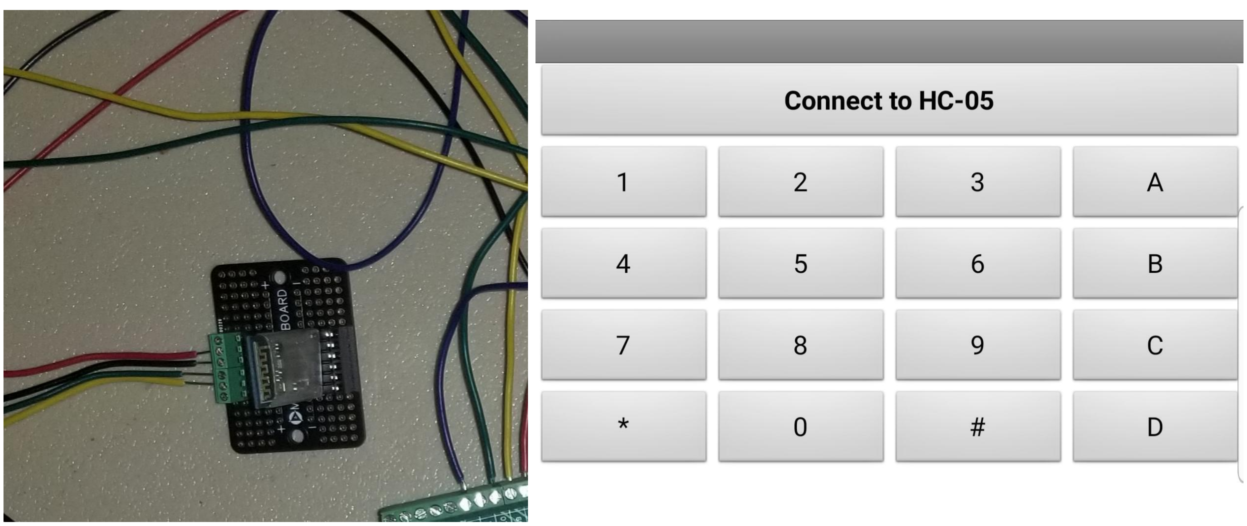
AI License Plate Reader



Garage Door Control



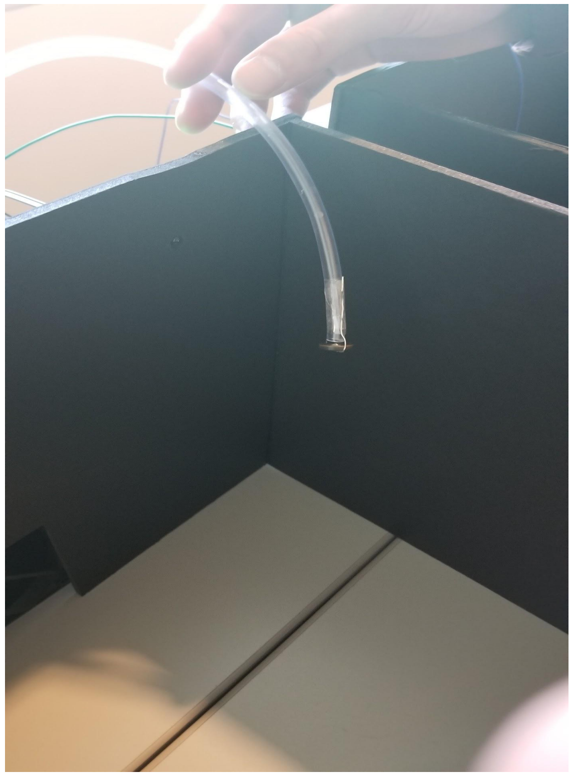
Bluetooth Keypad Entry



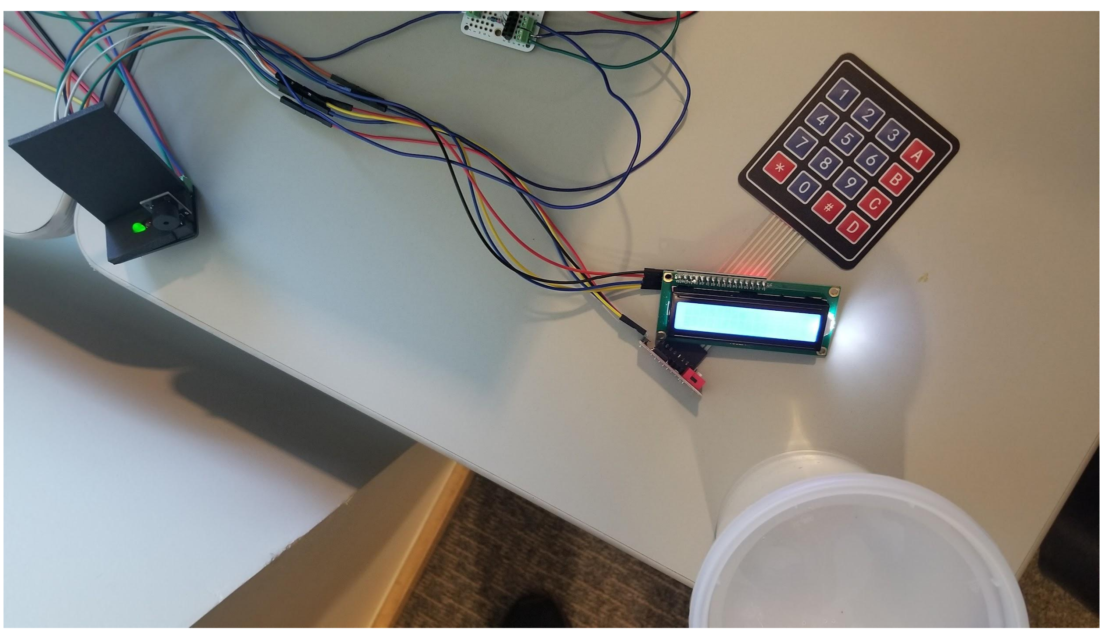
Smart Temperature Control

Ambient Light System

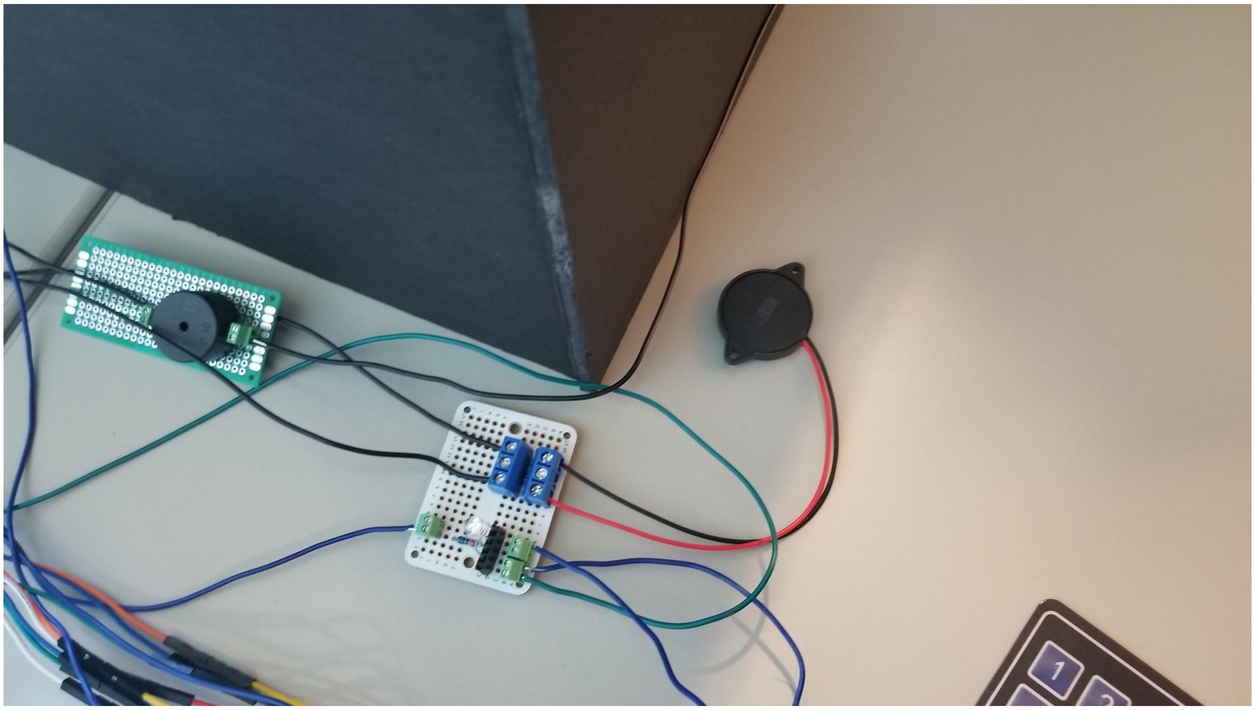
Fire Detection System



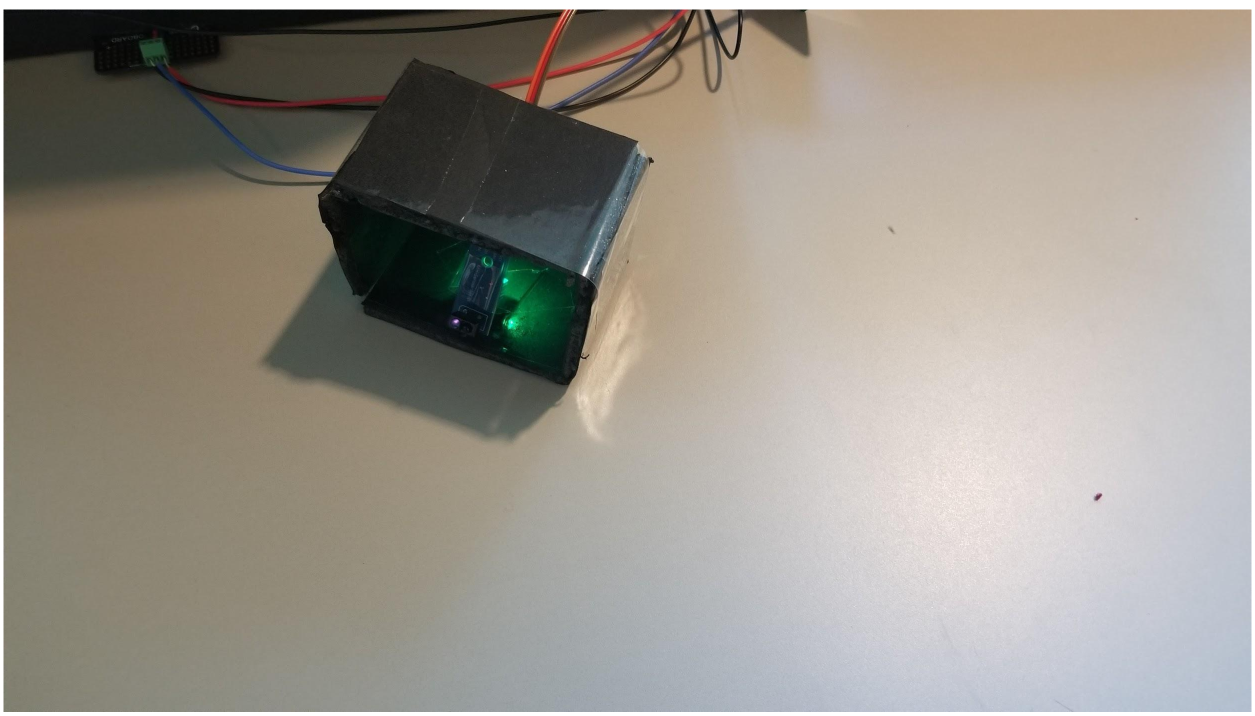
Home Security System



Door Knock Recognition System



Package Detection System



Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
v7	Revised Project Report (Final Version)	Graham Jabeguero, Raunac Bhuiyan	06/09/2022	
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 Autonomous Home System & Garage Sensor (or AHS) is a product for home owners to implement smart home features and a security system to their homes in a convenient manner. The motivation of this project is that most homes have not developed technologically as other products have such as smartphones, so this product could change this dilemma. The design objective is to provide customers with the option of a reliable smart home and security system that is non intrusive.

The AHS has nine subsystems. The Package Detection System utilizes an infrared sensor to detect packages within 2.5 centimeters of one's front door. The Door Knock Recognition System uses a Piezo sensor that senses a knock on the front door in order to ring the doorbell and blink a RGB LED to visually tell the homeowner that someone is at their door. The AI License Plate system utilizes a sonar sensor and a web camera. The sonar sensor detects a car in front of one's garage door. The web camera will activate and take a photo of the license plate. Moreover, if the character of the plate is valid, the Garage Door System will activate. In fact, the Garage Door System can be activated by exposing the photoresistor to light or reading a license plate using 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. The Fire Detection System activates the sprinkler when the infrared flame sensor senses fire and deactivates the sprinkler when a fire is no longer detected. It communicates to the homeowner via LCD screen. The Ambient Lighting System uses a photoresistor that detects the outside light levels to output the intended brightness for the indoor lights inside the home. When the photoresistor detects full brightness outside, the lights will turn off. Similarly, when the photoresistor detects no light outside, the lights will turn on with its brightest settings. The Home Security System is triggered when the motion sensor detects motion. Furthermore, an active buzzer will play a sound. The Smart Temperature System utilizes a temperature sensor and fan. For example, if the temperature is above 76 degrees, the fans will activate. The Keypad System controls the Home Security System and Smart Temperature System. For example, the Home Security System is activated or deactivated upon correct pin entry "1234*." The Keypad System can also take inputs using the homeowner's smartphone via Bluetooth.

As for testing results, 95% of the entire product worked as planned.

The most important achievement of this project is integrating all subsystems together to work correctly. Using serial communication between Arduino and Python for the AI License Plate Reader and the Garage Door System also was an important achievement. Another achievement is transitioning all subsystems from breadboard to printed circuit board (PCB). Also, every subsystem except the Ai License Plate Reader utilizes one Arduino Mega.

2 * Introduction

2.1 * Design Objectives and System Overview

This project, Autonomous Home System & Garage Sensor (AHS), was designed as a product to implement smart home features to existing electrical components and add new electrical components of one's home such as automating the control of lighting and temperature. The project's technical principles consist of providing nine features all into one microchip (an Arduino Mega using ATmega2560) and implementing one of the subsystems, the AI License Plate Reader, that uses Python to communicate with Arduino Mega via serial communication. The motivation for this project was addressing the problem of the lack of convenient control and functionalities for homeowners in contemporary homes. This product is unique in that the scale of the product affects the homeowner's entire home unlike other smart home products that feature one specific functionality. The intended use for this product is aimed towards homeowners who want smart home features that are affordable and effective.

High Level Description:

- The Fire Detection System activates the sprinkler when the infrared flame sensor senses fire and deactivates the sprinkler when a fire is no longer detected.
- The Ambient Light System uses a photoresistor that detects the outside light levels to output the intended brightness for the indoor lights inside the home.
- The Home Security System activates an alarm if the motion sensor detects any movements inside the house.
- The Smart Temperature System activates the fan if the sensor detects a temperature that is 76 degrees or higher.
- The Keypad Entry System controls the Home Security System and Smart Temperature System. For example, the Home Security System is activated or deactivated upon correct pin entry "1234*." The Keypad System can also take inputs using the homeowner's smartphone via Bluetooth.
- The Package Detection System utilizes an infrared sensor to detect if an object is within a certain distance. If so, then the LCD screen will display a message.
- The Door Knock System uses a Piezo sensor that senses a knock on the front door in order to ring the doorbell and blink a RGB LED to visually tell the homeowner that someone is at their door.
- The AI License Plate Recognition System detects a car and takes a picture of the license plate. Then, the image is processed and the characters are read.
- The Garage Door Control System opens and closes the garage door.

Quantitative Technical Design Objectives:

- The Fire Detection System utilizes the infrared flame sensor to detect fire. It takes approximately 4 seconds for the sensor to detect the fire. Then, the sprinklers activate and output water. The sprinklers will turn off after 5 seconds. The accuracy of this subsystem is 80%.
- The Ambient Light System senses a light level of more than 600, turns off lights; Less than 600 will gradually emit a brighter light depending on how low the light level is. The accuracy of this subsystem is 85%.
- The Home Security System, when activated, takes 2 seconds to trigger an alarm. The accuracy of this subsystem is 90%.

- The Smart Temperature System takes 5 seconds to trigger temperature control upon movement. When the temperature is read, it is within + or - 2 degrees Farenheit from actual temperature. The accuracy of this subsystem is 90%.
- The Keypad Entry System takes 3 seconds for the signal to be received by the Arduino. The accuracy of this subsystem is 90%.
- The Garage Door System utilizes a stepper motor which will only activate if either the photoresistor senses light level above 800 or a license plate has been read. The accuracy of this subsystem is 95%.
- The Package Detection System utilizes an infrared sensor that detects objects within 2.5 centimeters. The accuracy of this subsystem is 99%.
- The Door Knock Recognition System utilizes a Piezo Sensor that 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. Therefore, the accuracy of this subsystem is 80%.
- The AI License Plate Recognition System is an OpenCV source code. 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. 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. It will not read any characters on the license plate and will grant access. It has interference from the Bluetooth Keypad as described above. Total system accuracy is 75%.

Christopher Arellano was responsible for the following subsystems:
Home Security System
Smart Temperature System
Bluetooth Keypad Entry System

Raunac Bhuiyan was responsible for the following subsystems:
Package Detection System
Door Knock Recognition System
Garage Door Control System (using only Arduino)

Graham Jabeguero was responsible for the following subsystems:
Ambient Light System
Fire Detection System
Garage Door Control System (using FRDM-K64F and Arduino)

Christopher Arellano, Raunac Bhuiyan, and Graham Jabeguero were responsible for the AI License Plate Reader:

- Raunac Bhuiyan was responsible for making the Arduino Mega and Python communicate via serial communication.
- Christopher Arellano was responsible for making sure the webcam activates and takes a picture of a car's license plate when it is in a 5 cm distance.

- Raunac Bhuiyan and Graham Jabeguero were responsible for making sure that the picture that the webcam takes is processed to make the picture be read correctly.

2.2 * Backgrounds and Prior Art

Figure 1 shows a LPR/ANPR camera that is able to read license plates. This camera is able to read license plates from a certain distance. For our project we decided to implement a sonar sensor which is used to detect when a car is in close proximity. The camera will know when to take a picture of the license plate. This product is similar to the AI License Plate Reader of the AHS. It is important to note that this product is used for reading non U.S. license plates. It is able to read only European or Indian license plates. This is due to the formatting of the license plates compared to countries outside of the U.S. Similarly, the AI License Plate Reader of the AHS can only read non U.S. license plates [4].

Figure 1: LPR/ANPR Camera

2.3 * Development Environment and Tools

This product's hardware development utilizes an Arduino Mega 2560 Rev3 (that is based on Atmega2560) with Windows. The software development utilizes Arduino IDE for all subsystems except for the AI License Plate Reader and Keypad Entry System's Bluetooth component. The AI License Plate Reader was developed using Pycharm and the Keypad Entry System's Bluetooth component was developed using MIT App Inventor in order to connect an Android to the Keypad Entry System.

2.4 * Related Documents and Supporting Materials

ATmega2560 Datasheet (Used for the Arduino Mega 2560 Rev3):

https://content.arduino.cc/assets/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf

Arduino IDE (Used for all subsystems except for the AI License Plate Reader):

https://www.arduino.cc/reference/en/

Pycharm IDE Documentation (Used for the AI License Plate Reader):

https://www.jetbrains.com/help/pycharm/quick-start-guide.html

2.5 * Definitions and Acronyms

AHS - Autonomous Home System & Garage Sensor

3 * Design Considerations

3.1 * Realistic Constraints

Constraints that affected the project were: time, skill, size, and voltage supply.

To present the AHS, a home model was needed to present how all subsystems functioned. Time affected the presentability of the product as the home model was initially planned to be 3D printed but later modeled using medium weight chipboard. Integration of all 9 subsystems and the replacement of damaged components, like resoldering PCBs or replacing faulty sensors that did not function correctly, took most of the product's production. If more time was allotted, the home model would have been 3D printed and therefore be presentable in a professional manner.

These were the applicable skills that needed to be learned in order to complete this project: Python, OpenCV, SPI and I2C communication, soldering PCBs, controlling unfamiliar sensors such as motion or sonar sensors, Arduino. Python, OpenCV, and SPI communication were used to complete the AI License Plate Reader subsystem. Assembling PCBs by soldering was used to physically present the project in a professional manner instead of using breadboards. Learning how to implement new and unfamiliar sensors and Arduino was needed to make all subsystems work correctly as planned except for the AI License Plate Reader. I2C communication was needed to reduce the physical electrical wires of the Bluetooth Keypad System and Home Security System.

A physical constraint that impacted the project was the size of the entire project which took 30 in x 20 in x 12 in. It was impossible to carry the entire project as one so it had to be disassembled into 3 parts because of this problem. It took an hour and a half to completely assemble the project which was inconvenient when demoing and presenting in a timely manner.

A different voltage supply of 12V was needed for the L298N motor driver of the Garage Door Control System. Initially, a relay module was to be used for the voltage supply of the motor driver since it was being used already by the Smart Temperature System. This failed unexpectedly due to the motor driver needing a specific ground than the relay module. An external 9V battery was used as a replacement for it.

3.2 * Industry Standards

The Industry Standards involved in this project were: SPI, I2C, Bluetooth, USB.

Serial Peripheral Interface (SPI) initially operated the Garage Door Control System. More specifically, the Garage Door Control System utilized the Kinetis (FRDM-K64F) and Arduino Uno. In this case, the stepper motor which controls the garage door, was programmed via FRDM-K64F. The laser module, HiLetgo Infrared sensor, and the photoresistor were programmed via Arduino Uno. This method of utilizing two different microcontrollers was expensive and unnecessary. This had the potential to slow down or interfere with the other subsystems. Therefore, the final product only utilized one Arduino Mega for all subsystems except for the AI License Plate Reader.

I2C (Inter-Integrated Circuit) controlled the communication between the Home Security System's keypad component and the Arduino Mega to reduce the number of electrical wires connected to them to four.

Bluetooth communication controlled the Bluetooth Keypad System to the Home Security System. It acted as another method for the homeowner to input keypad inputs via smartphone without the need to use the Home Security System's physical keypad.

Serial communication via USB controlled the communication between the AI License Plate Reader and Garage Door Control System since the former was written in Arduino and the latter was written in Python. Certain test cases were written in the Python code, and if one of them were valid, the Python would send a 1 (true) to the Arduino. Therefore, this would indicate that a valid license plate was read and the arduino can open the garage door.

3.3 * Knowledge and Skills

Christopher Arellano

Prior Knowledge:

Engineering Circuit Analysis I and II (EE 001A/EE 001B), Electronic Circuits I (EE 100A), Introduction to Computer Science for Science, Mathematics, and Engineering I and II (CS 010/CS 012), Introduction To Data Structures and Algorithms (CS 014), Machine Organization and Assembly Language Programming (CS 061), Logic Design (EE/CS 120A), Introduction to Embedded Systems (EE/CS 120B), Intermediate Embedded and Real-Time Systems (CS 122A), Sensing and Actuation for Embedded Systems (EE 128)

New Knowledge:

Designing an app using the MIT App Inventor software, Programming in Python and Arduino, Serial Communication between Arduino and Python, Utilizing different types of sensors, Camera, Serial Communication via I2C chip, Bluetooth, Solder on printed circuit boards (PCB), Control fans and peltier pads.

Raunac Bhuiyan

Prior Knowledge:

Electronic Circuits I and II (EE 100A/EE 100B), Introduction to Computer Science for Science, Mathematics, and Engineering I and II (CS 010/CS 012), Logic Design (EE/CS 120A), Introduction to Embedded Systems (EE/CS 120B), Sensing and Actuation for Embedded Systems (EE 128)

New Knowledge:

Program in Arduino and Python, Serial Communication between Arduino IDE and Python IDE, OpenCV, Utilize different types of sensors such as Infrared and Piezo, Make use of motor drivers, Camera and Image Processing, K64F Microcontroller, Soldering on PCBs.

Graham Jabeguero

Prior Knowledge:

Introduction to Computer Science for Science, Mathematics, and Engineering I and II (CS 010/CS 012), Machine Organization and Assembly Language Programming (CS 061), Electronic Circuits I and II (EE

100A/EE 100B), Logic Design (EE/CS 120A), Introduction to Embedded Systems (EE/CS 120B), Sensing and Actuation for Embedded Systems (EE 128)

New Knowledge:

Soldering on PCBs, Cutting electrical wires, OpenCV, Program in Arduino and Python, Utilizing different types of sensors such as the Flame sensor, Debug and integrate subsystems efficiently.

3.4 * Budget and Cost Analysis

This project did not use parts from any Department. All hardware components, such as microcontrollers, solid electrical wires, or PCBs, were purchased with personal expenses. This required the purchase of a soldering kit and learning how to use it. The assistance of designing and printing the product's PCB by the Department was suggested by TA Maliha Tasnim but was later refused due to time constraint. Taking the help of the Department, the total cost would have been lower.

The total cost without accounting for tax:

Cost Analysis	
Part	Price $
Arduino Mega 2560 Rev3	$40
Photoresistors x2	$2
LEDs	$5
12 V Fan x2	$16
12 V Peltier Pad x10	$27
Temperature sensor	$2
Bluetooth module	$7
Sonar sensor	$2
Acxico Laser sensor x2	$7
Keypad	$5
I2C LCD display	$5
HiLetgo Infrared sensor x10	$9
PIR sensor	$2
Sprinkler water pump	$4
Passive Buzzer x2	$3

Cost Analysis	
Flame sensor	$2
Piezo sensor	$2
NEMA 17 Stepper motor	$14
Duracell Battery x4	$16
Quni L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper For Arduino	$7
Tuofeng Solid Wires	$17
Tuofeng Silicon Wires	$35
ElectroCookie Mini PCB Prototype Board	$45
Screw Terminal Block	$30
Screw/Terminal Block Shield Board Kit for Arduino Mega-2560 REV3	$17
Screw Mount Terminal Block Module	$11
12 V Power Supply	$13
DZS Elec 5V 4 Channel Relay Optocoupler Isolation Module	$20
HiLetgo I2C Adapter	$6
NEWACALOX Soldering Gun	$26
Total Cost	$397

3.5 * Safety

One of the safety issues in this project was the risk of burning certain hardware components. This is because our subsystems require the use of sensors. Furthermore, these parts rely on the Arduino’s 5V power supply port. For instance, the Garage Door System utilizes a laser sensor module, a photoresistor, and a stepper motor. These three components rely on the Arduino’s 5V power supply. Therefore, to ensure the parts do not burn, a screw mount terminal block was purchased. These terminal blocks supply the 5V power supply necessary to make the sensors function.

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 achieved by creating new code files every time a major change was made. The last version of the project's code is called "Merging_final_project_code_v6.ino," making it a total of six iterations during the product's development.

4 * Experiment Design and Feasibility Study

4.1 * Experiment Design

Objectives:

- Garage Door Control System's stepper motor, for the duration of ten seconds each, rotates clockwise and counterclockwise when an object is detected by the HiLetgo infrared sensor within 3.0 cm or when a light value of 500 or more is detected by the photoresistor. Blocking the KY-008 laser transmitter halts the motor's functionality and resumes it if the laser is not being blocked.

- AI License Plate Reader, when pressing the spacebar key on the keyboard, takes a picture of a car's license plate through a webcam, processes the image to be black and white for readability, reads the correct letters of the car's license plate, and outputs these letters through the Pycharm's terminal.

- Package Detection System's HiLetgo infrared sensor detects an object within 2.5 cm and sends a message, "Objected Detected," in Arduino IDE's serial monitor. It sends another message, "Object Removed," if the detected object has been removed from the infrared sensor's 2.5 cm range.

- Door Knock Recognition System's Adafruit Piezo sensor detects a vibration value of 25 (equivalent to an average vibration of a door knock). If the Piezo sensor detects this value or more, the Passive Buzzer plays a short melody and the RGB LED emits an order of blue, red, and green and dims after.

- Fire Detection System's Teyleten infrared flame detection sensor detects flames within 7.5 cm. After three seconds of detection, it sends a message, "Fire Detected," through Arduino IDE's serial monitor and the WayinTop water pump activates until no flame is detected by the infrared flame detection sensor. When a flame is no longer detected, the water pump deactivates after a delay of two seconds and the serial monitor displays a message, "Sprinklers stopped."

- Ambient Light System's warm white LED emits five different brightness levels depending on the light being received by the photoresistor. If the photoresistor receives a light value of 600 or more, the LED emits no brightness. If the photoresistor receives a light value of 300 or less, the LED emits the highest brightness of 255. Any light received between 300 or 600 by the photoresistor will emit a LED brightness of 10, 50, or 128.

Setup:

- The Garage Door Control System needed a power supply of 12 V to operate the NEMA 17 stepper motor. The stepper motor needs to be connected to the L298N motor driver. Since this is a bipolar stepper motor, four wires are connected to the motor driver. This motor driver needs to be connected to the Arduino's 5 V power supply and ground. A photoresistor is utilized to signal the door to close. The KY-008 laser transmitter and sensor are connected to the Arduino Uno. For the FRDM-K64F and Arduino microcontroller set up, an HiLetgo infrared sensor is also utilized to activate the stepper motor.

- For the AI License Plate Reader, an openCV program is obtained from an open-source GitHub repository. To test this program, various pictures of car license plates from Google were acquired. At least 15 pictures were used for testing.
- The Package Detection System utilizes an HiLetgo infrared sensor. The Arduino's software serial monitor is utilized for displaying results. This infrared sensor has both digital and analog capabilities; therefore, both were used for testing. The hardware components are placed on a breadboard and are connected to the Arduino microcontroller via USB.
- The Door Recognition Knock System utilizes one Piezo sensor. A one million ohm resistor is used to operate the sensor. An RGB LED and a Passive Buzzer are also utilized. The hardware components are placed on a breadboard and are connected to the Arduino Uno via USB.
- The Fire Detection System is set up with the Teyleten infrared flame detection sensor and a WayinTop water pump. The hardware components are placed on a breadboard and are connected to the Arduino Uno via USB.
- The Ambient Light System is set up with an white LED and a photoresistor. The hardware components are placed on a breadboard and are connected to the Arduino Uno via USB.

Procedure:

- The Garage Door Control System has one test case. The garage door should open (i.e. the stepper motor should rotate clockwise) if the photoresistor detects a light value of 600 or greater. When it uses the FRDM-K64F board, the stepper motor is activated via infrared sensor.
- The AI License Plate Reader has one test case. After the webcam takes a picture of the license plate by pressing the spacebar, the image is processed, and the characters will be read. If it is valid, then the program will print "Plate Recognized." Recorded what the program read and did this for ten trials for one of the pictures of car license plates.
- The Package Detection System has two test cases. If the HiLetgo infrared sensor detects an object within 2.5 cm, then the message "Object Detected" will be printed on the Arduino serial monitor. If the object is no longer within 2.5 cm, then it will print "Object Removed."
- The Door Knock Recognition System has one test case. If the Piezo sensor detects a vibration value of 25 or higher, then the RGB LED and Passive Buzzer will activate.
- The Fire Detection System is 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. Used a lighter to make the flame sensor give a signal to activate the relay which will activate the sprinklers.
- The Ambient Light System is tested by setting up the photoresistor by checking the wiring configuration and using a flashlight on the photoresistor to output all five levels of brightness to the LED.

Result:

- Garage Door Control System's stepper motor operated as planned based on the objective described above. Blocking the laser transmitter successfully halts the stepper motor's functionality and the infrared sensor and photoresistor give the correct signals to activate the stepper motor.

- AI License Plate Reader's webcam correctly takes the picture when pressing the spacebar on the keyboard. It successfully processes the image to be black and white for better readability. The program read the license plate correctly seven out of ten tries. The accuracy of the program is 70%.

- The Package Detection System correctly detects an object within the 2.5 cm range and sends the correct message to the Arduino IDE's serial monitor. Also, sends the correct message if the object gets taken out of the 2.5 cm range.

- Door Knock Recognition System correctly plays the buzzer and the RGB LED when detecting a vibration value of 25 and higher.

- The Fire Detection System successfully turns on the sprinklers when fire is detected within a 7.5 cm range. The delays for turning on and off the sprinklers and the messages being displayed were both correct.

- The Ambient Light System's five levels of brightness were correctly outputted depending on how much light the photoresistor was being received by the flashlight.

Responsibilities:

- Raunac Bhuiyan was 100% responsible for the Garage Door Control System, Package Detection System, and the Door Knock Recognition System.

- Christopher Arellano was 100% responsible for the Keypad Entry System, Smart Temperature System, and the Home Security System.

- Graham Jabeguero was 100% responsible for the Garage Door Control System, Fire Detection System and the Ambient Light System.

- All three team members were responsible for the AI License Plate Recognition System.

*Note: Team member Christoper Arellano is not enrolled in this course. These are his original objectives/procedures.

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.

4.2 * Experiment Results, Data Analysis and Feasibility

- AI License Plate Reader 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.” Therefore, to account for these cases, the code was adjusted to grant access to the garage door, as long as it could read the first few characters or a blank string. In other words, it will not read any characters on the license plate and will grant access. Total system accuracy is 70%.
- The Garage Door Control System’s stepper motor will only activate if either the photoresistor senses a light level of 600 or a license plate has been read from the program 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 Recognition System’s Piezo sensor can detect a vibration on the table, and triggers the Passive Buzzer. However, the threshold had to be changed from 25 to 50 on some occasions because the sensor reads the vibration even with a slight tap. Moreover, this causes the buzzer and LED’s to activate when not intended. Total system accuracy is 80%
- The Fire Detection System’s flame sensor reads the correct analog-to-digital values when a flame is nearby. The Water Pump correctly turns on the relay module which then turns on/off the water pump. Total system accuracy is 90%.
- The Ambient Light System 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. Total system accuracy is 80%.

*Note: Team member Christoper Arellano is not enrolled in this course. These are his original objectives/procedures.

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%).

5 * Architecture and High Level Design

5.1 * System Architecture and Design

Figure 2 is the block diagram of the entire system. The AI License Plate Reader subsystem takes a picture of the license plate of a car. The image is then processed on the Python software. The Garage Door Control System operates after receiving the signal from Python. This is essential because the garage door will open if the license plate is recognized. The Garage Door Control System can also operate as an independent system. This means that it is possible to open or close the garage door without reading a car's license plate. This is because the Garage Door Control System can operate if the interior of the garage senses a certain amount of light. It is important to note that the Garage Door Control System has a laser security feature that prevents anyone from entering or exiting the garage while the door is operating.

The Smart Temperature System senses temperature in degrees Fahrenheit. Depending on the temperature, the fans will operate. The Keypad Entry System can allow the user to enter a key that activates the fans regardless of the temperature. The fans can operate without sensing the temperature. The Door Knock Recognition System senses a knock and alerts the user by playing a sound.

The Home Security System is responsible for keeping the home secure by setting up an alarm. The Keypad Entry System allows the user to arm or disarm the Home Security System. While the Home Security system is active, any motion detected inside the home triggers the alarm. This alerts the user that someone has broken into their house. The Keypad Entry System can be accessed via Bluetooth communication. The user can connect to the Keypad Entry System using their smartphone.

The Fire Detection System senses fire and sends a signal to the sprinklers. The sprinklers activate and pump out water to extinguish any fire within the user's house. The sprinklers will stay active as long as fire is detected inside the house. A message will be displayed on the LCD monitor notifying the user of a fire.

The Ambient Light System senses light from the outside and activates the lights inside the home. The brightness of the light brightens or dims depending on the brightness of the light outdoors. The Package Detection System senses a package in front of the user's porch and notifies the user. This will be done by the LCD monitor displaying a message. However, the package needs to be placed within a certain distance; otherwise, the user will not be alerted. It is important to note that Wi-Fi is not necessary to operate the entire system.

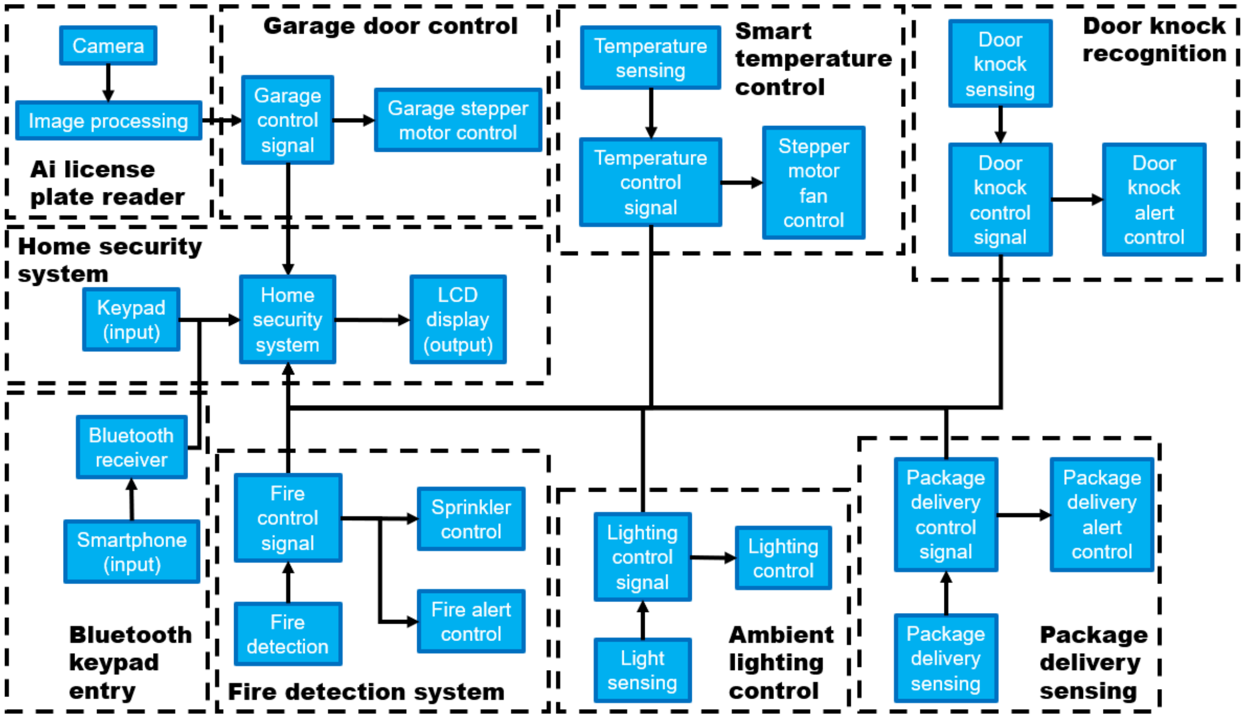


Figure 2: All System Block Diagram

Component tests and assembly of the following items done by Christopher Arellano:
Home Security System
Bluetooth Keypad System
Smart Temperature System
AI License Plate Reader (Sonar sensor, Arduino & Python serial communication, Camera control)

Component tests and assembly of the following items done by Raunac Bhuiyan:
Door Knock Recognition System
Package Detection System
Garage Door Control System
AI License Plate Reader (Character recognition, Arduino & Python serial communication)

Component tests and assembly of the following items done by Graham Jabeguero:
Fire Detection System
Ambient Light System
Garage Door Control System
AI License Plate Reader (Image Processing)

5.2 * Hardware Architecture

Figure 3 is composed of multiple subsystems that communicate with one another and make up the AHS. The central subsystem of the project is the Home Security System because the LCD 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 except the AI License Plate Reader. The AI License Plate Readers hardware is the PC. A USB A-Male to B-Male cable was utilized to connect the PC and the Arduino Mega.

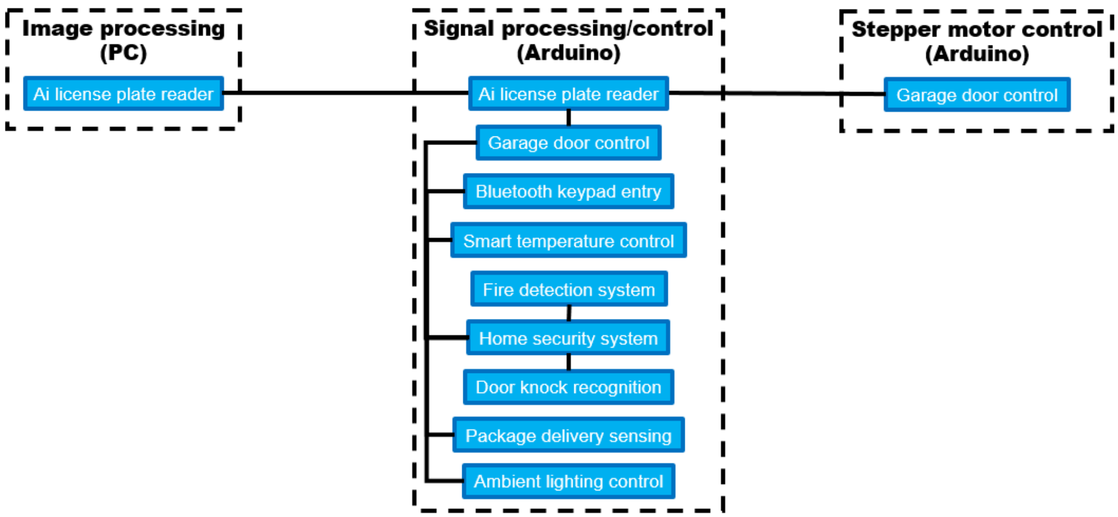


Figure 3: MCU Block Diagram

Component tests and assembly of the following hardware done by Christopher Arellano:
Home Security System
Bluetooth Keypad System
Smart Temperature System
AI License Plate Reader (Sonar sensor, Arduino & Python serial communication, Camera control)

Component tests and assembly of the following items done by Raunac Bhuiyan:
Door Knock Recognition System
Package Detection System
Garage Door Control System
AI License Plate Reader (Character recognition, Arduino & Python serial communication)

Component tests and assembly of the following items done by Graham Jabeguero:
Fire Detection System
Ambient Light System
Garage Door Control System
AI License Plate Reader (Image Processing)

5.3 * Software Architecture

Figure 4 shows the entire system is programmed using a single Arduino Mega except the AI License Plate Reader. The AI License Plate Reader was programmed in Python and communicated with Arduino using serial communication. Each subsystem of the project is divided into individual functions to facilitate the management 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.

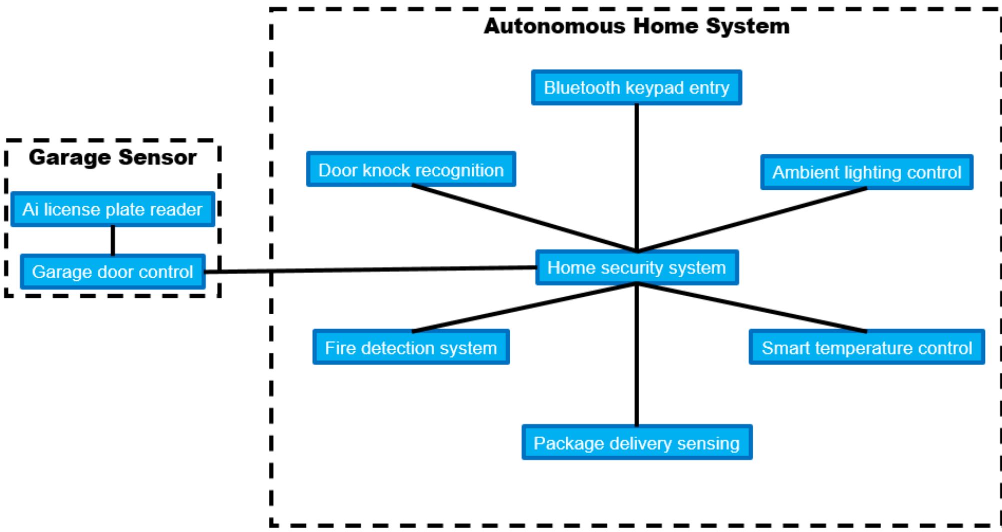


Figure 4: Simplified Block Diagram

Component tests and programming of the following software done by Christopher Arellano:
Home Security System
Bluetooth Keypad System
Smart Temperature System
AI License Plate Reader (Sonar sensor, Arduino & Python serial communication, Camera control)

Component tests and programming of the following software done by Raunac Bhuiyan:
Door Knock Recognition System
Package Detection System
Garage Door Control System
AI License Plate Reader (Character recognition, Arduino & Python serial communication)

Component tests and programming of the following software done by Graham Jabeguero:
Fire Detection System

Ambient Light System
Garage Door Control System
AI License Plate Reader (Image Processing)

5.4 * Rationale and Alternatives

There were possible alternatives to the development of this project using multiple microcontrollers, such as adding a FRDM-K64F, a Raspberry Pi, and an ATmega1284p with the Arduino Mega. This was decided as the project's initial development because it was believed to make the product more unique. But, consulting with the professor, Dr. Roman Chomko, it was more reasonable to put all subsystems into a single Arduino Mega because using multiple microcontrollers and attempting to connect them to each other would be an unnecessary expense to the overall product budget. Other than budget cost, we realized it was possible to develop the product with one Arduino Mega.

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. Most of the subsystems made for this project required all group members to learn new applicable skills within this allotted time.

7 * Low Level Design

The product's circuit schematic is broken up into three parts based on each group member's responsibilities. The schematics for 6.1, 6.2, and 6.3 were each produced individually by the following order: Christopher Arellano, Raunac Bhuiyan, Graham Jabeguero.

7.1 *Home Security, Smart Temperature, Bluetooth Keypad Entry

The **Figure 5** schematic consists of three subsystems: Home Security System, Smart Temperature System, and Bluetooth Keypad System.

The Home Security System's main components consist of an I2C LCD and a 4x4 matrix array keypad that are powered by connecting them to the Arduino Mega's 5V power source. Both components use I2C to decrease the electrical wires needed to connect them to the Arduino Mega by utilizing the SDA and SCL pin. Other components include a buzzer and a red and blue LED for the alarm.

The Smart Temperature System's components consist of two 12V fans and a peltier pad that are powered by a 12V 4 channel relay module. The sensors include a temperature sensor and a sonar sensor. The relay is powered by an external 12V battery while the sensors are powered by connecting them to the Arduino Mega 5V power source pin. The system uses the components from the Home Security System to control the fans mode via keypad and displays the mode and temperature using the LCD.

The Bluetooth Keypad Entry System shows only a Bluetooth module powered by the 5V pin in the Arduino Mega. The other component, the product owner's phone, is not shown here since it does not connect physically to the Arduino Mega. The product owner's phone will use an app developed using MIT App Inventor to send signals to the bluetooth module. The bluetooth module is connected to the communication pins, RX and TX of the Arduino Mega, to receive signals from the phone.

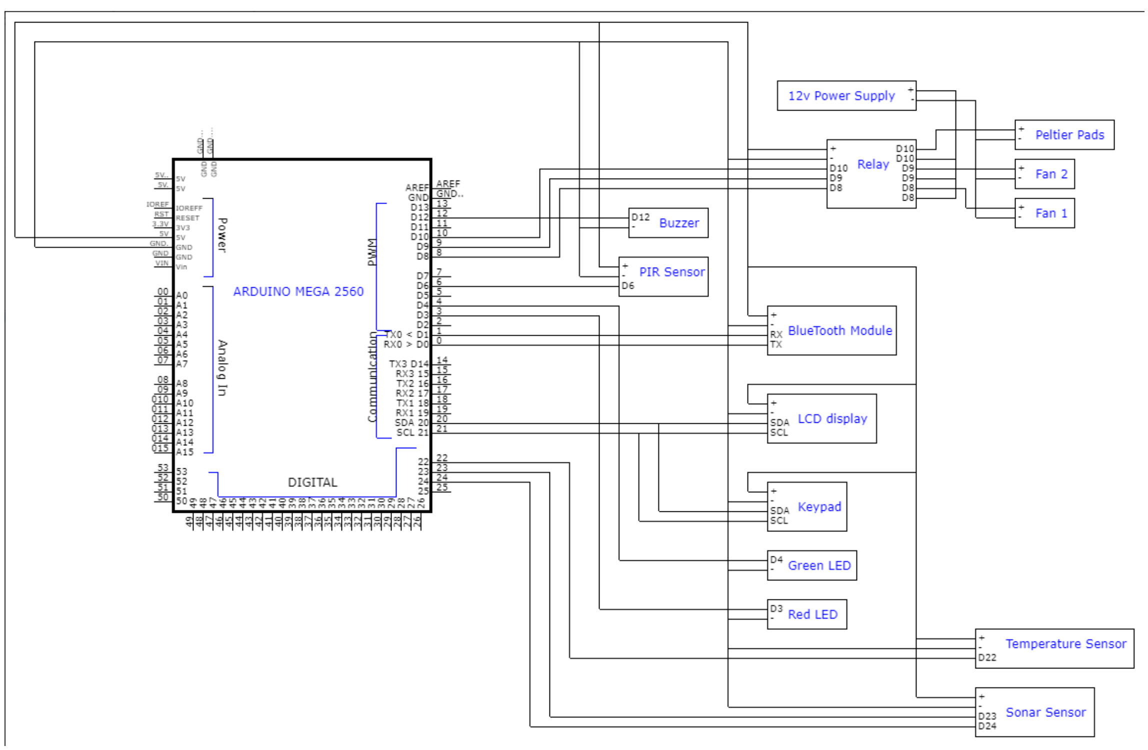


Figure 5: Home Security/Smart Temperature/Bluetooth Schematic

7.2 *Door Knock Recognition, Package Detection

This **Figure 6** schematic shows the Door Knock Recognition and Package Detection system. The Door Knock Recognition system uses a Passive Buzzer. This component is connected to an Arduino digital and ground pin. Moreover, this part is responsible for playing sound with random frequencies. The Door Knock Recognition system also uses a RGB LED. This component is one LED that alternates between the colors red, green and blue. Furthermore, it is connected to three Arduino digital pins. The Package Detection system utilizes the HiLetgo infrared sensor. This sensor is connected to the Arduino 5V and analog pin. This component will detect any object within 2.5 cm by bouncing the emitted IR light to itself.

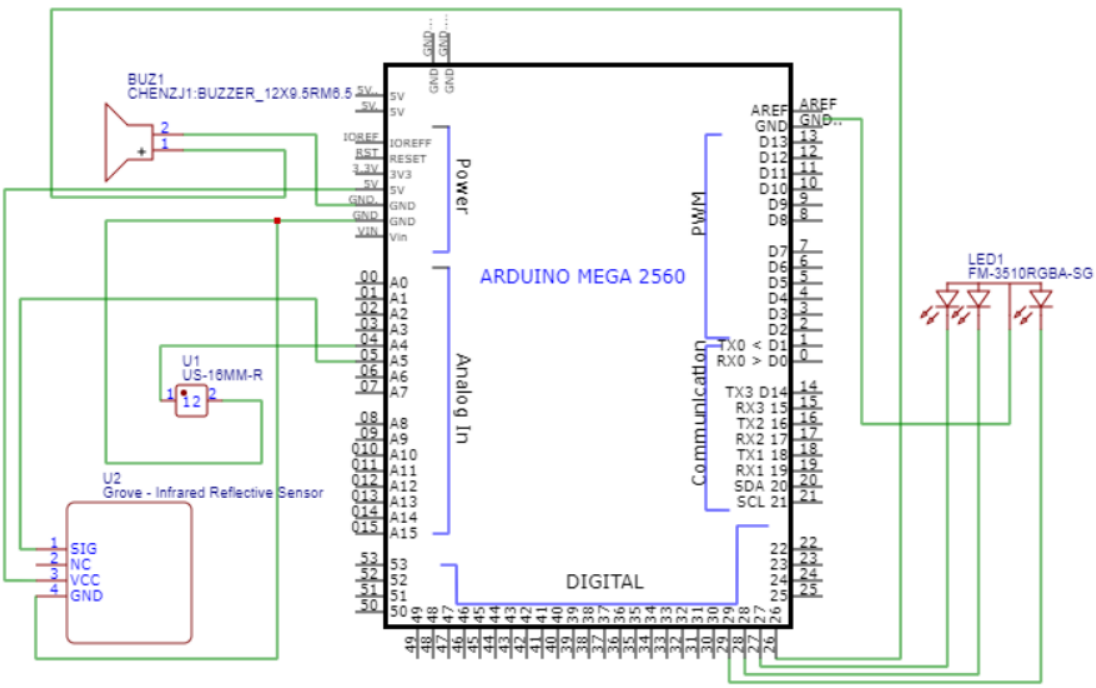


Figure 6: Door Knock Recognition/Package Detection Schematic

7.3 *Garage Door

This **Figure 7** schematic shows the Garage Door Control system. The L298N motor driver is used to activate the NEMA17 stepper motor. This component is connected to the Arduino 5V and ground pin. Moreover, this motor driver is also responsible for supplying the 9V power supply that is received from the Duracell battery. Also, the 9V Duracell battery is connected to a battery clipper which is not shown in the schematic. The KY-008 Laser module is connected to the Arduino 5V and ground pin. Furthermore, this component only activates while the stepper motor is in motion. The photoresistor is connected to an Arduino analog pin as well as the 5V and ground pin. This component can be used to activate the NEMA 17 stepper motor.

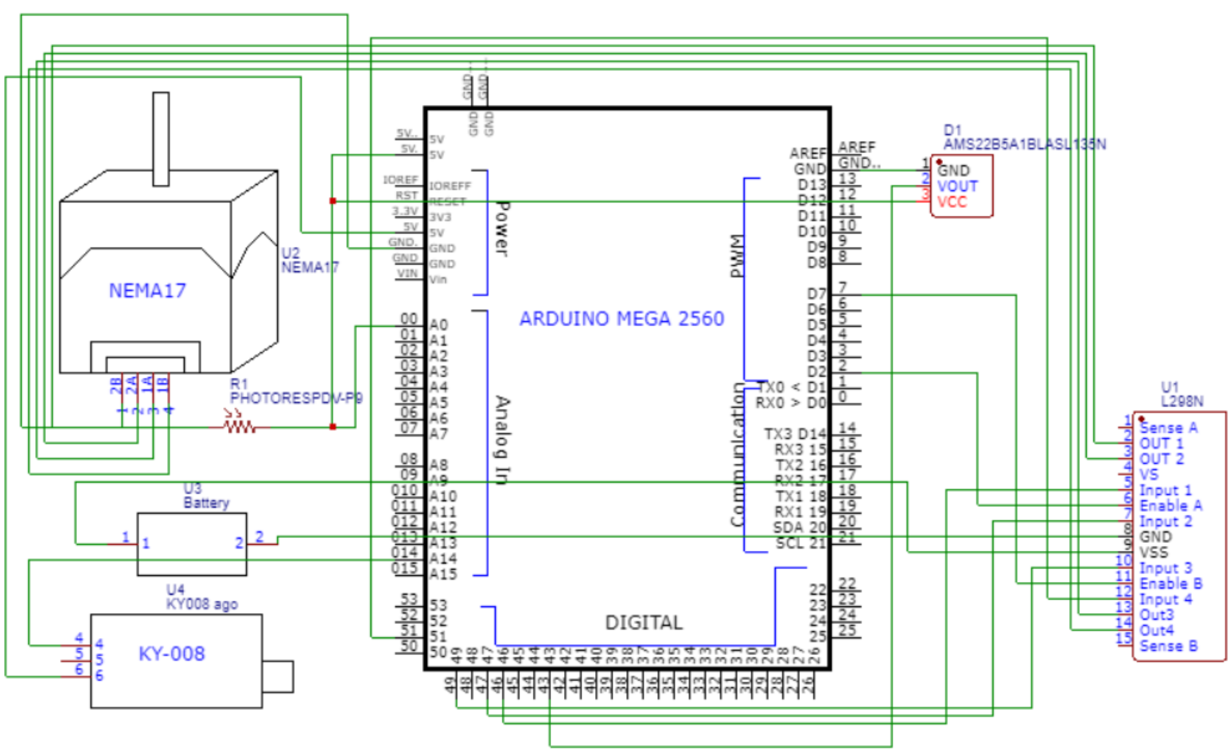


Figure 7: Garage Door Schematic

7.4 Fire Detection, Ambient Lighting

This **Figure 8** schematic shows the Fire Detection System on the left hand side and the Ambient Light System on the right hand side. The Fire Detection System consists of a KY-026 IR flame sensor powered by the Arduino Mega’s 5V pin which detects the fire within one’s home. The output of the system is the water pump sprinkler powered by a relay module to make the sprinkler’s function properly. It also uses the Home Security System’s LCD to display statuses of the system.

The Ambient Light System is powered entirely by the Arduino Mega 5V pin. It consists of the one photoresistor to detect outdoor environmental light and a LED to output the indoor lights of one’s home. Both subsystems are also connected to common ground.

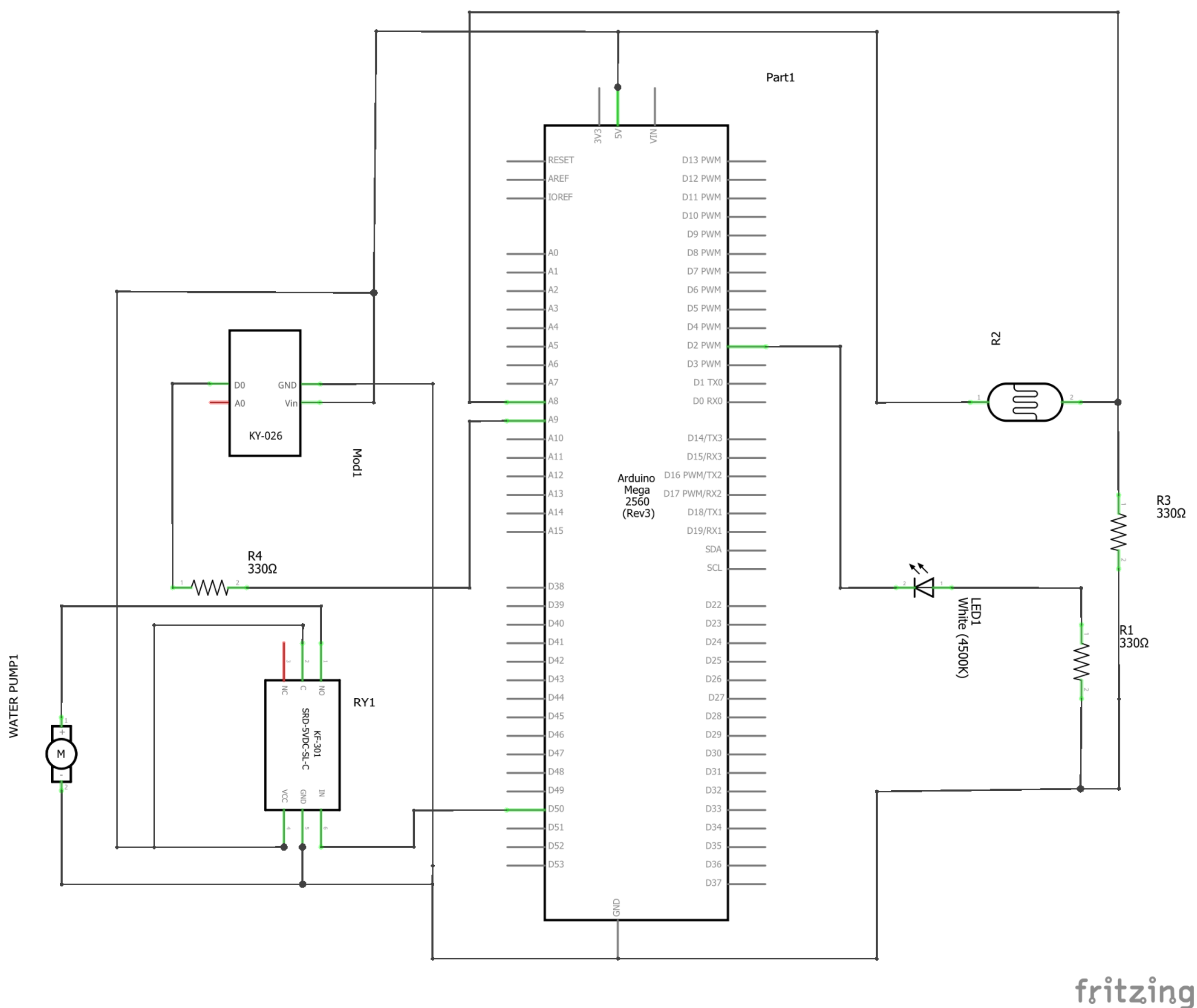


Figure 8: Fire Detection/Ambient Lighting Schematic

8 *Technical Problem Solving

Technical problems related to our project involved hardware and software problems. Subsection 8.1 and 8.2 will discuss problems on hardware and how they were solved. Subsection 8.3 and 8.4 will discuss problems on software and how they were solved.

8.1 Hardware Problems

- The Garage Door Control System uses a stepper motor to operate the garage door. However, when the stepper motor ran, the torque was slower than expected. In other words, it was not running at normal speed. This problem was identified by Raunac Bhuiyan.
- The Door Knock Recognition System interrupted the other subsystems. If a user were to knock on the door, the other subsystems would be halted. In some cases, after knocking on the door, the passive buzzer would continuously play sound. This problem was identified by Raunac Bhuiyan.
- The Package Detection System used a HiLetgo infrared (IR) sensor. The range was limited to 2.5 cm. However, there will be times when the system does not recognize an object. This problem was identified by Raunac Bhuiyan.
- The Ambient Light System used a photoresistor and white LED. After integration of all subsystems, the LED did not correctly emit the correct brightness no matter what the photoresistor received . This problem was identified by Graham Jabeguero.

8.2 Solving the Hardware Problems

- The Garage Door System was solved by replacing the external battery for the, it appeared that the stepper motor required a 12 V power supply. However, this was not possible. The problem was solved with a 9 V Duracell battery. Nonetheless, the torque was still slow, but there was an overall improvement to the system's stepper motor speed. This issue was solved by Raunac Bhuiyan.
- To solve the problem of the Door Knock Recognition System, certain portions of the code were changed. More specifically, while loops were not used in the program. Moreover, the while loops were replaced with if condition statements. This issue was solved by Raunac Bhuiyan.
- To solve the Door Knock Detection System's problem, certain portions of the code were changed. Similar to the Door Knock Recognition System, while loops were removed and replaced with if condition statements. Furthermore, since the HiLetgo infrared sensor had both digital and analog capabilities, analog was the preferred option. This issue was solved by Raunac Bhuiyan.
- The Ambient Light System was solved by changing the pins of the Arduino Mega. Moreover, the system required the use of PWM (pulse width modulation). Therefore, pins that supported PWM were used to operate this subsystem. This issue was solved by Graham Jabeguero.

8.3 Software Problems

- The AI License Plate Reader System uses image processing to read the characters of a car's license plate. However, the webcam or camera did not read every character of the license plate correctly. For example, if the plate is "KAME05," sometimes the system recognizes it as "KATSOs." This problem was identified by Raunac Bhuiyan and Graham Jabeguero.

- The Fire Detection System used an IR flame sensor and water pump sprinkler. The desired time to turn on the sprinkler was inconsistent after integration of all other subsystems. If the system detected fire, the sprinkler did not activate when intended. This problem was identified by Graham Jabeguero.

8.4 Solving the Software Problems

- The AI License Plate Reader System system was solved by changing what the camera takes a picture of. Printed images of car license plate readers were used instead. Initially, the license plate was on a smart device, and the computer's webcam would take a picture. However, with physical images, the program was able to better recognize the characters of the license plate. This issue was solved by Graham Jabeguero.

- The Fire Detection System was solved by editing the program. Instead of using the Arduino's "delay()" function as a time module, counters were used. The counters number had to be reduced several times to account for other subsystems. This issue was solved by Graham Jabeguero.

Note: The Following was written by team member Christoper Arellano, who is not a student in this course.

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

10 *Test Plan

10.1 *Test Design

Test Case 1:

1. The objective is to test the integration of the Home Security System, Smart Temperature System, and Bluetooth Keypad System. It will test the Home Security's alarm, the keypad, and the LCD. It will test the Smart Temperature System's fans and the motion sensor. It will test the Bluetooth Keypad System's bluetooth receiver. This measures the accuracy of all components when integrated together.

2. The Home Security System, Smart Temperature System, and Bluetooth Keypad System need to be correctly working individually using an Arduino Mega. An android phone needs the created app from MIT App Inventor installed.

3. Connect the Home Security System to the Arduino Mega. Test the Home Security's alarm by inputting the sequence "*1234" on the keypad. Connect the Smart Temperature System to the Arduino Mega. Test the Home Security's alarm again and the test the Smart Temperature System's fan's by inputting the sequence "A*", "AA*", "AAA*." Connect the Bluetooth Keypad System to the Arduino Mega. Test the Home Security System, Smart Temperature, and Bluetooth Keypad by using the android app and inputting all the sequences above.

4. The integration of the Home Security System, Smart Temperature System, and Bluetooth Keypad System worked just like it was before integration.

   Christopher Arellano was responsible for this test.

Test Case 2:

1. The objective is to test the integration of the Door Knock Recognition System and Package Detection System. It will test the Door Knock Recognition System's piezo sensor, buzzer, and RGB LED. It will test the Package Detection System's infrared sensor. This measures the accuracy of all components when integrated together.

2. The Door Knock Recognition System and Package Detection System need to be correctly working individually using an Arduino Uno.

3. Connect the Door Knock Recognition System to the Arduino Uno. Test the Door Knock Recognition System's piezo sensor by placing the piezo sensor on a table and knocking on the table itself. Connect the Package Detection System to the Arduino Uno. Test the Door Knock Recognition System again and test the Package Detection System by placing one's hand towards the infrared sensor.

4. The integration of the Door Knock Recognition System and Package Detection System worked just like it was before integration.

Raunac Bhuiyan was responsible for this test.

Test Case 3:

1. The objective is to test the integration of the Ambient Lighting System and Fire Detection System. It will test the Ambient Lighting System's photoresistor and LED. It will test the Fire Detection System's flame sensor and water pump. This measures the accuracy of all components when integrated together.

2. The Ambient Lighting System and Fire Detection System need to be correctly working individually using an Arduino Uno. A lighter is needed to test the Fire Detection System.

3. Connect the Ambient Lighting System to the Arduino Uno. Test the Ambient Lighting System by shining a light towards the photoresistor using a flashlight or phone. Observe the different brightness by dimming or letting more light into the photoresistor. Connect the Fire Detection System to the Arduino Uno. Test the Ambient Lighting System again and then test the Fire Detection System by lighting a fire near the flame sensor using the lighter. Hold the fire for a few seconds to turn on the water pump.

4. The integration of the Ambient Light System and Fire Detection System worked just like it was before integration.

   Graham Jabeguero was responsible for this test.

Test Case 4:

1. The objective is to test the integration of the AI License Plate Reader and Garage Door Control System. It will test the AI License Plate Reader's camera with the Garage Door Control's stepper motor. This measures the two's functionality together.

2. The Garage Door Control System needs to be correctly working as an individual system using an Arduino Uno. The AI License Plate Reader needs to be correctly working as an individual system using Pycharm. A webcam and an A4 size picture of a car with a readable license plate is needed.

3. Connect the Garage Door Control System to the Arduino Uno. Test the system by inputting serial commands "1" or "2" in the Arduino Serial Monitor to see if the stepper motor runs correctly. Open up Pycharm and run the AI License Plate Reader code. Test this system by placing the picture of the car in front of the webcam and then press space bar to capture the picture. Observe if it correctly reads the characters of the license plate and see if the stepper motor runs as well.

4. The integration of the AI License Plate Reader and Garage Door Control System worked just like it was before integration.

   Christopher Arellano, Raunac Bhuiyan, and Graham Jabeguero were responsible for this test.

Test Case 5:

1. The objective is to test the integration of all nine subsystems. It will test the functionality of all subsystems. This measures the accuracy of all components.

2. There needs to be four integrations of subsystems that work correctly. The first integration is the Home Security System, Smart Temperature System, and Bluetooth Keypad System. The second integration is the Door Knock Recognition System and Package Detection System. The third integration is the Ambient Lighting System and the Fire Detection System. The fourth integration is the AI License Plate Reader and the Garage Door Control. The final integration will use an Arduino Mega.

3. Connect the first integration, Home Security System, Smart Temperature System, and Bluetooth Keypad Entry System, to the Arduino Mega. Test all systems to see if it works correctly. Connect the second integration, Door Knock Recognition and Package Detection System, to the Arduino Mega. Test all systems including the first integration to see if it works correctly. Connect the third integration, Ambient Lighting System and the Fire Detection System, to the Arduino Mega. Test all 3 integrations together. Connect the fourth integration, AI License Plate Reader and the Garage Door Control System, to the Arduino Mega. Test all nine subsystems to see if everything works correctly.

4. The integration of all nine subsystems worked like it was before integration.

   Christopher Arellano, Raunac Bhuiyan, and Graham Jabeguero were responsible for this test.

10.2 * 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).

10.3 * 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.

10.4 * Identification of critical components

Christopher Arellano was responsible for the following task:

- I2C: Used to display characters in the LCD of the Home Security System and reads from the keypad.

- Bluetooth: Used to connect the user's phone to the AHS as a Bluetooth keypad.

- Temperature Sensor: Used to measure the temperature of the house. Needed for the Smart Temperature System.

- PIR Sensor: Used to trigger “Auto Temp” mode of the Smart Temperature System when motion is detected. Used to trigger the alarm of the Home Security System when motion is detected.

- Sonar Sensor: Used to trigger messages from Arduino to Python for the communication of the AI License Plate Reader and the Garage Door Control System.

Raunac Bhuiyan was responsible for the following task:

- L298N motor driver: Used to power on the stepper motor for the Garage Door Control System.

- Laser module: Used as a safety precaution when the stepper motor is operating for the Garage Door Control System.

- HiLetgo IR sensor: Used to detect objects within the user’s front door for the Package Detection System.

- Photoresistor (Garage Door Control): Used to trigger a signal to open the garage when a car is already inside for the Garage Door Control System.

- Piezo sensor: Used to detect vibrations of the user’s front door to send a signal to the Passive Buzzer for the Door Knock Recognition System.

Graham Jabeguero was responsible for the following task:

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

- Relay Module: Used to power on the water pump for the Fire Detection System.

- Flame Sensor: Used to give the signal to power on the relay module for the Fire Detection System.

10.5 * Items Not Tested by the Experiments

N/A

11 * Test Report

11.1 *Test 1

Christopher Arellano performed this experiment:

1. The integration of the Home Security System, Smart Temperature System, and the Bluetooth Keypad Entry System worked like it was before integration. Integration was a 100% success.
2. The expected results and the actual results were the same.
3. Successful integration meant that the team can progress to the fifth integration.
4. No corrective actions were taken because the experiment was 100% successful.

11.2 *Test 2

Raunac Bhuiyan performed this experiment:

1. The detection of the object by the Package Detection System halted the functionality of the Door Knock Recognition System. Integration was a 60% success.
2. It was unexpected that the detection of an object by the Package Detection System halted the functionality of the Door Knock Recognition System. In a 100% successful integration, when identifying a nearby object by the Package Detection System, the Door Knock Recognition System functions as if it were a standalone system.
3. Identified where the problem was as it was due to the Package Detection System's code being stuck in a loop when it identified an object.
4. Replaced the loop with if-statements, effectively making the integration a 100% success.

11.3 *Test 3

Graham Jabeguero performed this experiment:

1. The integration of the Ambient Lighting System and the Fire Detection System was an 80% success.
2. There was a slight delay in turning on the sprinklers of the Fire Detection System that was not expected.
3. This is due to having another system, the Ambient Light System, integrated with the Fire Detection System as this makes the overall code run slower than usual.
4. Fixed this problem by reducing the delay count that the Fire Detection System had from 3000 counts to 1000 counts. This change fixed the exact time the sprinklers turned on.

11.4 *Test 4

Christopher Arellano, Raunac Bhuiyan, and Graham Jabeguero performed this experiment:

1. The integration of the AI License Plate Reader and Garage Door Control System was a 90% success.
2. There was a problem trying to line up the picture of the car to the webcam and taking a picture of it by signaling the AI License Plate Reader with a sonar sensor.
3. Aligning the picture to the webcam and signaling the sonar sensor was hard for one person to do at the same time since the sonar sensor and webcam was far 12 inches away.
4. When testing these two subsystems, there always needed to be at least two team members to test it as aligns the picture of the car to the webcam and the other holds their hand in front of the sonar sensor to signal the AI License Plate Reader to take the picture and process it.

11.5 *Test 5

Christopher Arellano, Raunac Bhuiyan, and Graham Jabeguero performed this experiment:

1. The integration of the entire project was a 86.1% success.
2. The expected results were a 100% success and the actual result was 86.1%. This was due to interference with other subsystems
3. Some subsystems such as the Fire Detection System and Door Knock Recognition System interfered with each other. An example of this occurring is when fire was detected by the Fire Detection System and it signals the doorbell of the Door Knock Recognition to ring.
4. There were no corrective actions taken as the team members did not have time to fix this.

12 * Conclusion and Future Work

12.1 * Conclusion

In conclusion, the finished work meets the overall project goals and the quantitative technical design objectives. The AHS has nine sub-systems that work together synchronously. For example, the AI License Plate System and the Garage Door Control System work synchronously. First the sensor will detect a car approaching, and the camera will take a snapshot of the license plate. Then, this will send a signal to the Arduino to open the garage door. The Keypad Entry System is responsible for activating the Home Security, Smart Temperature, and Fire Detection System. In other words, if a user enters the correct key or password onto the Keypad Entry System, it can be manually turned on or off the three subsystems. In addition, the Keypad Entry System is usable on a smartphone or any device with Bluetooth capabilities. As for the Door Knock Recognition, Package Detection System, and Ambient Lighting System, these work independently. For instance, if the Package Detection System alerts the user that a package is delivered, this will not interfere with the other subsystems.

In completing this project, Graham Jabeguero learned technical skills such as soldering a PCB to finalize a product and coding in Arduino and Python. In terms of professional growth, he learned how to work in a group and communicate effectively to progress the project forward.

Raunac Bhuiyan learned technical skills such as coding in Arduino and Python and how to communicate between two different IDEs through serial communication. In terms of professional growth, he learned how to work in a group and ask for feedback through communicating with his team.

A potential future design of the AHS would be to make every project component wireless and have the components communicate through a network instead. As a result, it will remove the use of wires from the project components to the Arduino. Moreover, assembling the entire project is costly because it takes a long time. Additionally, to avoid damaging the circuits, they should not be visible to the user. Another consideration would be to replace the current Bluetooth module. An updated version of this Bluetooth module will permit the Arduino to send and receive signals from both a PC and smartphone synchronously. Therefore, reliance on the USB will no longer be necessary. Thus, instead of serial communication via USB, it will be serial communication via Bluetooth from Arduino to python. In regards to the marketability of the AHS, it is important to acknowledge that the primary demographic who will use this product are people 60 and older. The reason is that, in the U.S., most homeowners are people 60 and older. Moreover, older people have less experience with newer technology. Therefore, a future design of this project will be more user-friendly. Another future design of this product would be to make it usable for people who live in apartment complexes. For example, it is possible to implement the AI License Plate System at the front gate of an apartment building; as a result, only verifiable users or residents will be able to enter the complex. Thus, it will also improve the security of the building.

During the development of the AHS, each team member was responsible for completing their systems. Otherwise, it would not be possible to merge the entire system. Graham Jabeguero was responsible for the Ambient Lighting System, and Fire Detection System. Raunac Bhuiyan was responsible for the Garage Door Control, Package Delivery, and Door Knock Recognition System. Christopher Arellano was responsible for the Keypad Entry System, Home Security System, and Smart Temperature System. As for the AI License Plate System, every team member had a responsibility to ensure the system was functional. For example, Christopher made sure the camera would activate if a car approaches. Raunac Bhuiyan made it possible for the Arduino and Python code to work synchronously via serial communication. Graham tested numerous license plates to make sure that image processing was correctly reading the characters of the license plate.

12.2 * 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

13 * 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].

14 * Appendices

* **Appendix A**: Parts List
Hardware used:
Arduino Mega
Laptop
2 Photoresistors
LEDs
12 V fan x2
12 V peltier pad x2
Temperature sensor
Bluetooth receiver
Smart phone
Camera/laptop webcam
Sonar sensor
KY008 Laser/Sensor module
Keypad
LCD
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