Department of Computer Science and Engineering

Course Code: CSE461	Credits: 1.5
Course Name: Introduction to Robotics Lab	Semester: Spring 24

Hardware Project

Smart Scape: An Autonomous Smart Home

Group(2) members of Section(3)

Shihab Muhtasim(21301610)

Tasnia Jannat Ayesha(21301611)

Hasin Arman Prokriti(20201092)

A.B.M. Fahim Hasan Tanzin(21101257)

Tushar Chowdhury(21301010)

Supervising faculties

Shakir Rouf

Niloy Irtisam

Table of contents

Introduction & Objective:

Smart scape is a smart home robotics model made using Arduino and other components that merges the convenience and safety of users. The primary objective of this project is to design and implement a home automation system with the help of Arduino and sensors. The main objective of this project is to provide both user accessibility and autonomous safety measures. The project has some automatic features such as doors opening upon a person's arrival which bring convenience to the lifestyle of the users. Another feature is lighting up the room automatically when there's darkness which is a useful way to save energy as we don't have to keep the light on. We have also included safety features such as fire detection and flammable gas detection. These safety features are essential to every household/public space as they can save precious lives.

Real-world need:

An autonomous smart home solution like ours can be used in homes, offices, hotels, or in any public space. Our model is built considering the safety of fire and user convenience. The autonomous light ON/OFF system can be implemented in an important common space where the light needs to be On all the time such as a hotel lobby, office common room, etc. Moreover, the autonomous door opening solution is helpful for public spaces where the door needs to be opened every time someone comes in like the entrance of a restaurant/super shop or a room inside a house. On the other hand, we have a smoke detector alarm system that smells of any gas leak and alerts the people for their safety. Additionally, the fire detection system triggers an alarm as well as keeps the door open for people to exit immediately. Hence, this project has great potential to be implemented in many places in the world.

Demonstration Video:

https://youtu.be/rzl3d3x_Jic?feature=shared

Github Link:

https://github.com/shihabmuhtasim/Smart-Scape

Components:

- Arduino Uno
- Breadboard
- Ultrasonic Sensor
- Mini Servo Motor SG-90
- LDR Sensor
- LEDs
- MQ2 Gas Sensor
- Buzzer
- Thermistor
- Resistors
- Jumper wires

Functionality:

The are 4 major functionalities in this project:

1. **Automatic door**: we have installed an ultrasonic sensor in front of the door and the door will be moved with the help of a servo motor attached to it. The ultrasonic sensor will detect whether someone tries to enter the room and then activate the servo. The sensor sends a high-frequency sound and detects the reflected sound while recording the time taken. This value is used to calculate the distance from the object. When no one is present in front of the sensor, the distance is a greater value. The value decreases when someone stands in front of it and when the distance is less than a certain threshold, the Arduino activates the servo to open the door.

2. **Automatic Light:** This functionality helps the lights to turn on automatically when it becomes dark outside. This is best suited for porch lights as it will automatically be turned on during the evening. The LDR sensor detects the intensity of the surrounding lights. When the intensity of light falls below the threshold, the Arduino then activates the LED.
3. **Smoke or Flammable gas detector:** this safety feature is implemented in the kitchen where there is a high risk of fire hazard. The MQ2 gas sensor can measure the concentration of the smoke or flammable gasses present in the air. The readings increase if the concentration of these substances increases. After it crosses the threshold, the Arduino then sends a signal to the buzzer which starts to make a loud noise to warn the residents. The buzzer only stops when the concentration drops below that specified threshold.
4. **Fire Detection:** another safety feature of our system is fire detection. This will be implemented in places where there is a risk of fire. We have used a thermistor to detect fires. The thermistor is one type of resistor whose value changes depending on the temperature. As the temperature increases, the resistance decreases. We have connected a wire that feeds the voltage value to the A05 pin of the Arduino. The Arduino then processes the voltage value and calculates the temperature. When the temperature reaches a threshold value, the Arduino then activates the buzzer to warn people about the fire. Additionally, the Arduino also activates the servo of the door so that the door opens for the people to escape to safety. As additional safety measures, the Arduino also disables all the other connected electrical appliances.

Code:

```
#include <Servo.h>
//initialization of variables
// Task 1: Door control with ultrasonic sensor
Servo servo1;
int trigPin = 9;
int echoPin = 8;
long distance;
long duration;

// Task 2: LDR and LED control
int ldrPin = 6; // LDR pin
int ledPin = 13; // LED pin

// Task 3: Buzzer control with analog sensor
#define BUZZER_PIN 3

// Task 4: Temp sense and actuate servo and buzzer+LED
int ThermistorPin = 5;
int Vo;
float R1 = 10000; //resistor in the circuit
float logR2, R2;
float T; // Moved T declaration here
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 =
2.019202697e-07;

//set up pins
void setup() {
  // Task 1: Door control with ultrasonic sensor
  servo1.attach(7);//attach servo to pin 7 of arduino
  pinMode(trigPin, OUTPUT); //trigger the pins in
ultrasonic
  pinMode(echoPin, INPUT);

  // Task 2: LDR and LED control
  Serial.begin(9600);
  pinMode(ldrPin, INPUT); //sensor input
  pinMode(ledPin, OUTPUT); //led output

  // Task 3: Buzzer control with analog sensor
  pinMode(BUZZER_PIN, OUTPUT); //buzz output

  // Initialize Serial for temperature sensor
  Serial.begin(9600);
}
```

1.

```
//Loop to run the model
void loop() {
  // Task 1: Door control with ultrasonic sensor
  ultra_sonic();
  servo1.write(90); // 90 is closed door
  while (distance <= 10 ) { //continues checking is d <10
    ultra_sonic(); //if yes, then again checks d
    servo1.write(0); //  0 degrees is the open position
    delay(2000);     // Delay for 2 seconds before closing
the door
     // Move back to closed position
  }

  servo1.write(90);



  // Task 2: LDR and LED control
  int ldrValue = digitalRead(ldrPin); //reads pin val
  Serial.print("LDR: ");
  Serial.println(ldrValue);
  if (ldrValue == HIGH) {
    digitalWrite(ledPin, HIGH); // Turn on the LED if it's
dark
  } else {
    digitalWrite(ledPin, LOW); // Turn off the LED if it's
bright
  }

  // Task 3: Buzzer control with analog sensor
  int sensorValue = analogRead(A0); // read analog sensor
value
  Serial.print("Sensor gas: ");

  Serial.println(sensorValue);
  if (sensorValue > 500) {  // threshold value gas sensor
    tone(BUZZER_PIN, 50);
  } else {
    noTone(BUZZER_PIN); // Stop generating the tone
when condition is not met
  }
```

2.

// Task 4: Temperature decision
temp();
if (T > 100) {//threshold thermistor
tone(BUZZER_PIN, 500);
servo1.write(0);
delay(10000);
 	
} else {
noTone(BUZZER_PIN);
servo1.write(90);
}
 	
delay(1000); // Add a delay
}
 	
// Ultrasonic sensor function
void ultra_sonic() {
digitalWrite(trigPin, LOW); //initiates sending signal process
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);//send signal
delayMicroseconds(10);
digitalWrite(trigPin, LOW); //pulse complete
duration = pulseIn(echoPin, HIGH); //when echo high- receive then measure time
distance = duration * 0.034 / 2; //sound speed .034 m.micro sec
Serial.print("Distance: ");
Serial.print(distance);
Serial.print("meter ");
 	
}
3.
	// Temperature sensor function
	void temp() {
	Vo = analogRead(ThermistorPin); //voltage level
	//voltage divider
	R2 = R1 * (1023.0 / (float)Vo - 1.0); //calculates resistance of thermistor
	logR2 = log(R2); //use log to initialize the relation of volt and temp
	T = (1.0 / (c1 + c2 * logR2 + c3 * logR2 * logR2 * logR2)); //Steinhart-Hart equation for thermistors ....temperature in Kelvin (K).
	T = T - 273.15;//c
	T = (T * 9.0) / 5.0 + 32.0; //F
	
	Serial.print("Temperature: ");
	Serial.print(T);
	Serial.println(" F");
	
	delay(500);
	}
	4.

Code Explanation:

Initialization of Variables

The code starts by including the necessary library for the servo motor, Servo.h. Then, it initializes various variables and pins used in the project. First, it initializes trigPin and echoPin for the ultrasonic sensor. Then ldrPin and ledPin for the LDR sensor and LED. After that, BUZZER_PIN for the buzzer. Moreover ,it sets ThermistorPin for the thermistor. Then it Initializes R1, c1, c2, and c3 are constants used in the Steinhart-Hart equation for thermistors.

Setup Function

The setup() function is called once at the beginning of the program. It sets up the necessary pins . This segment Attaches the servo motor to pin 7 of the Arduino . Then it Sets up the trigPin as an output and echoPin as an input for the ultrasonic sensor. After that it Initializes the serial communication at a baud rate of 9600. Additionally, Sets up the ldrPin as an input and ledPin as an output for the LDR sensor and LED. Then it Sets up the BUZZER_PIN as an output for the buzzer

Loop Function

The loop() function is called repeatedly after the setup() function. It contains four main tasks to be performed continuously.

Task 1: Door Control with Ultrasonic Sensor

This task uses the ultrasonic sensor to detect objects in front of the door. The code calls the ultra_sonic() function to read the distance from the sensor. If the distance is less than or equal to 10, it opens the door by writing a value of 0 to the servo motor. After a delay of 2 seconds, it closes the door by writing a value of 90 to the servo motor.

Task 2: LDR and LED Control

This task uses the LDR sensor to detect light intensity. The code reads the value from the LDR sensor using digitalRead(ldrPin). If the value is HIGH (i.e., it's dark), it turns on the LED using digitalWrite(ledPin, HIGH). Otherwise, it turns off the LED using digitalWrite(ledPin, LOW).

Task 3: Buzzer Control with Analog Sensor

This task uses an analog sensor to detect gas levels. The code reads the value from the sensor using analogRead(A0). If the value is greater than 500 (the threshold value), it generates a tone using tone(BUZZER_PIN, 50). Otherwise, it stops generating the tone using noTone(BUZZER_PIN).

Task 4: Temperature Decision

This task uses the thermistor to detect temperature levels. The code calls the temp() function to read the temperature. If the temperature is greater than 100 (the threshold value), it generates a tone using tone(BUZZER_PIN, 500) and opens the door using servo1.write(0). After a delay of 10 seconds, it closes the door using servo1.write(90). Otherwise, it stops generating the tone using noTone(BUZZER_PIN).

Defined 2 functions outside loop:

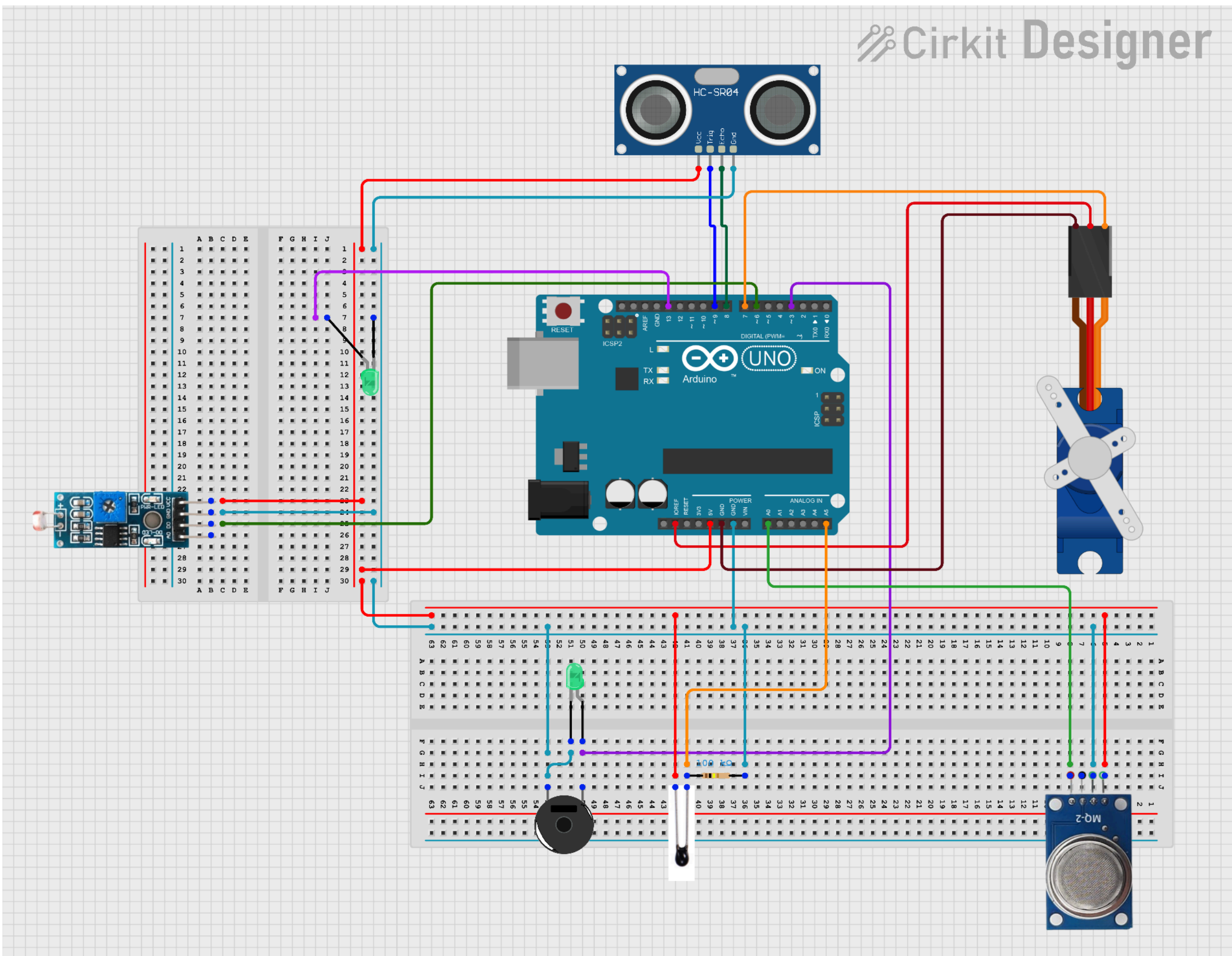
Ultrasonic Sensor Function

The ultra_sonic() function is used to read the distance from the ultrasonic sensor. It sends a high-frequency sound wave using digitalWrite(trigPin, HIGH) and measures the time taken for the wave to bounce back using pulseIn(echoPin, HIGH). The distance is then calculated using the speed of sound formula.

Temperature Sensor Function

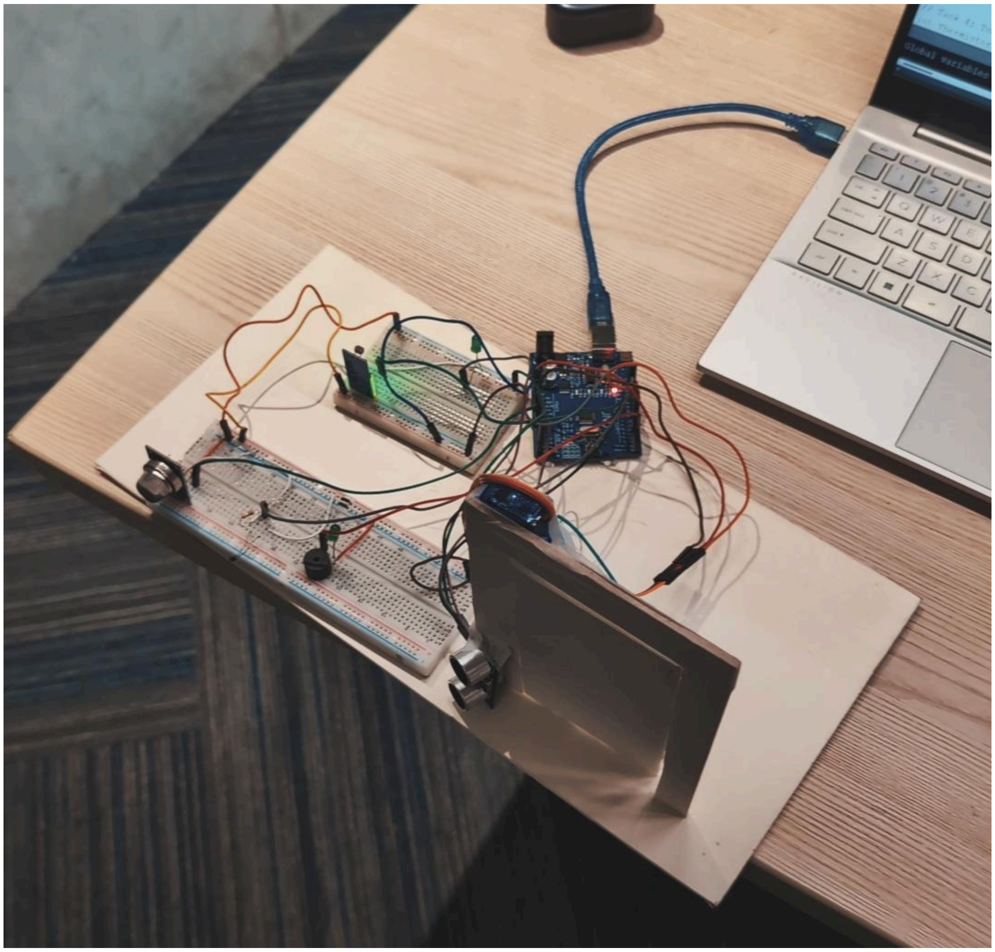
The temp() function is used to read the temperature from the thermistor. It reads the voltage level from the thermistor using analogRead(ThermistorPin) and calculates the resistance of the thermistor using Ohm's law. The temperature is then calculated using the Steinhart-Hart equation.

<u>Diagram & Pins:</u>



Ultra: Echo pin- arduino 8 Ultra: Trig pin- arduino 9 LED for thermistor: arduino 3 LDR: D0- arduino 6	LED for light sensor : arduino 13 Thermistor one end : arduino A5 MQ-2 sensor D0 : arduino A0 Servo Motor - arduino 7

Image:



Scalability:

This model has an adaptation capability to many other features that make it scalable. More features can be added in future according to the need of specific applications. For example if the user has a garden then a soil moisture sensor can be added to monitor the water content of soil and the arduino can automatically add water when needed. By implementing wireless modules like wifi or bluetooth, the model can be monitored and controlled remotely as well. Moreover, an automatic ac system can be implemented where the room temperature will always be the same.

Challenges:

We faced several challenges while building this project. Firstly, many of our components were not working or got damaged which took long hours to debug and fix. Secondly, many times the Arduino was not properly working when we tried uploading the code which caused us to reconnect all wires. Again, due to some wires not functioning it was hard to debug the bugs. Additionally, our thermistor measures voltage and we had to study to find ways to convert it to temperature. Moreover, in different environments our model has different threshold values for example where there's ceiling fan vs where there's Ac not no fan the smoke will be detected differently. So to find an ideal threshold for both smoke and heat detector was a challenge. Additionally the code segment properly merging and making everything work together required a lot of troubleshooting.

Conclusion:

The Smart Scape project is a sophisticated smart home robotics model utilizing Arduino and various sensors to enhance user convenience and safety. By automating tasks like door opening, lighting control, smoke and gas detection, and fire alarm systems, this project offers a blend of functionality and safety crucial for modern living spaces. The scalability of the design allows for future enhancements like soil moisture monitoring and remote control capabilities. Despite facing challenges during development, the project showcases the potential for innovative smart home solutions with a focus on user experience and safety.