Constructor University Bremen

Natural Science Laboratory

Introduction to Robotics and Intelligent Systems

Spring Semester 2024

Project: Smart Security System.

Instructor: Hu Fangning

Author: Wanzia Nambule, Cherub Ngirazi, Ahmed Abasimel, Gerard Nahimana

Place of execution: Robotics Lab
Rotation I, Group A

INTRODUCTION

In an era of increasing concerns about safety and security, the need for robust and proactive security measures has never been more critical. Our project addresses this need by introducing a Smart Security System designed to alert individuals to potential dangers in their environment. Through the integration of sensors, microcontrollers, and audio feedback mechanisms, our system serves as an early warning system, capable of promptly notifying individuals of impending threats.

Setup and Execution

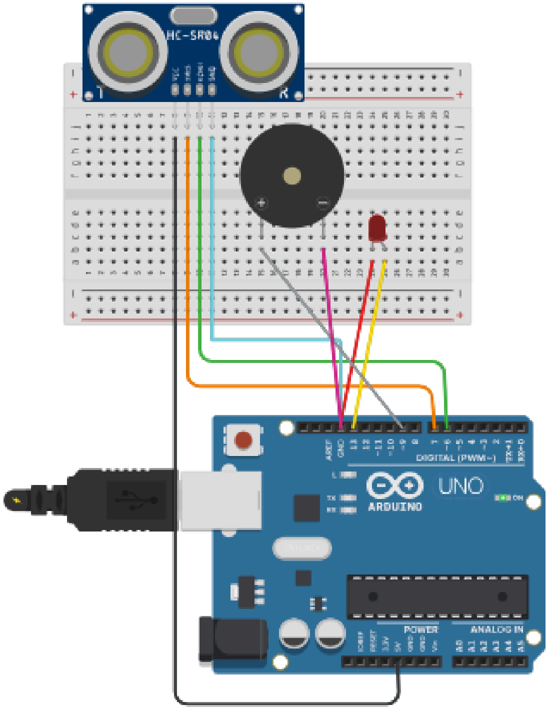


Figure 1: Image setup of the project.

1. Connect the HC-SR04 ultrasonic sensor's VCC and GND pins to the Arduino's 5V and GND pins, respectively.
2. Connect the sensor's TRIG pin to digital pin 7 on the Arduino and the ECHO pin to digital pin 6.
3. Connect the LED's anode (longer leg) to digital pin 8 on the Arduino and the cathode (shorter leg) to GND.

4. Connect the buzzer’s positive pin to digital pin 9 on the Arduino and the negative pin to GND.
5. Upload the provided code below to the Arduino.

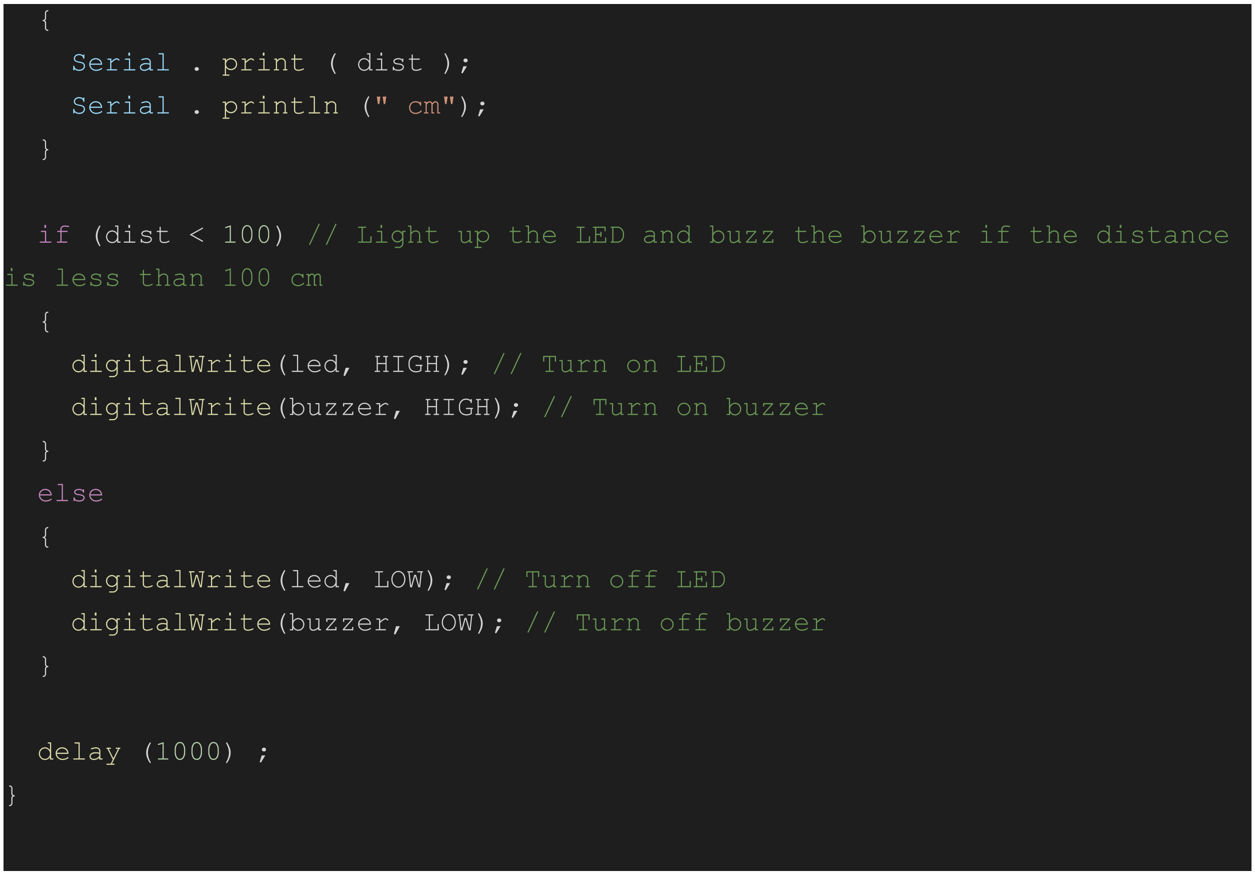
```
const int echo = 6;
const int trig = 7;
const int led = 8; // Define the LED pin
const int buzzer = 9; // Define the buzzer pin

void setup ()
{
  pinMode (trig , OUTPUT );
  pinMode (echo , INPUT );
  pinMode (led , OUTPUT ); // Set the LED pin as output
  pinMode (buzzer , OUTPUT ); // Set the buzzer pin as output
  digitalWrite (trig , LOW);
  digitalWrite (led , LOW); // Initially turn off the LED
  digitalWrite (buzzer , LOW); // Initially turn off the buzzer
  Serial . begin (9600) ;
}

void loop ()
{
  // send an impulse to trigger the sensor start the measurement
  digitalWrite (trig , HIGH );
  delayMicroseconds (15) ; // minimum impulse width required by HC -SR4
sensor
  digitalWrite (trig , LOW);
  long duration = pulseIn (echo , HIGH ); // this function waits until the
sensor

  const long vsound = 340; // [m/s]
  long dist = ( duration / 2L) * vsound / 10000L; // 10000 is just the
scaling

  if ( dist > 500L || dist < 2L)
  {
    Serial . println (" Invalid range !");
  }
  else
```



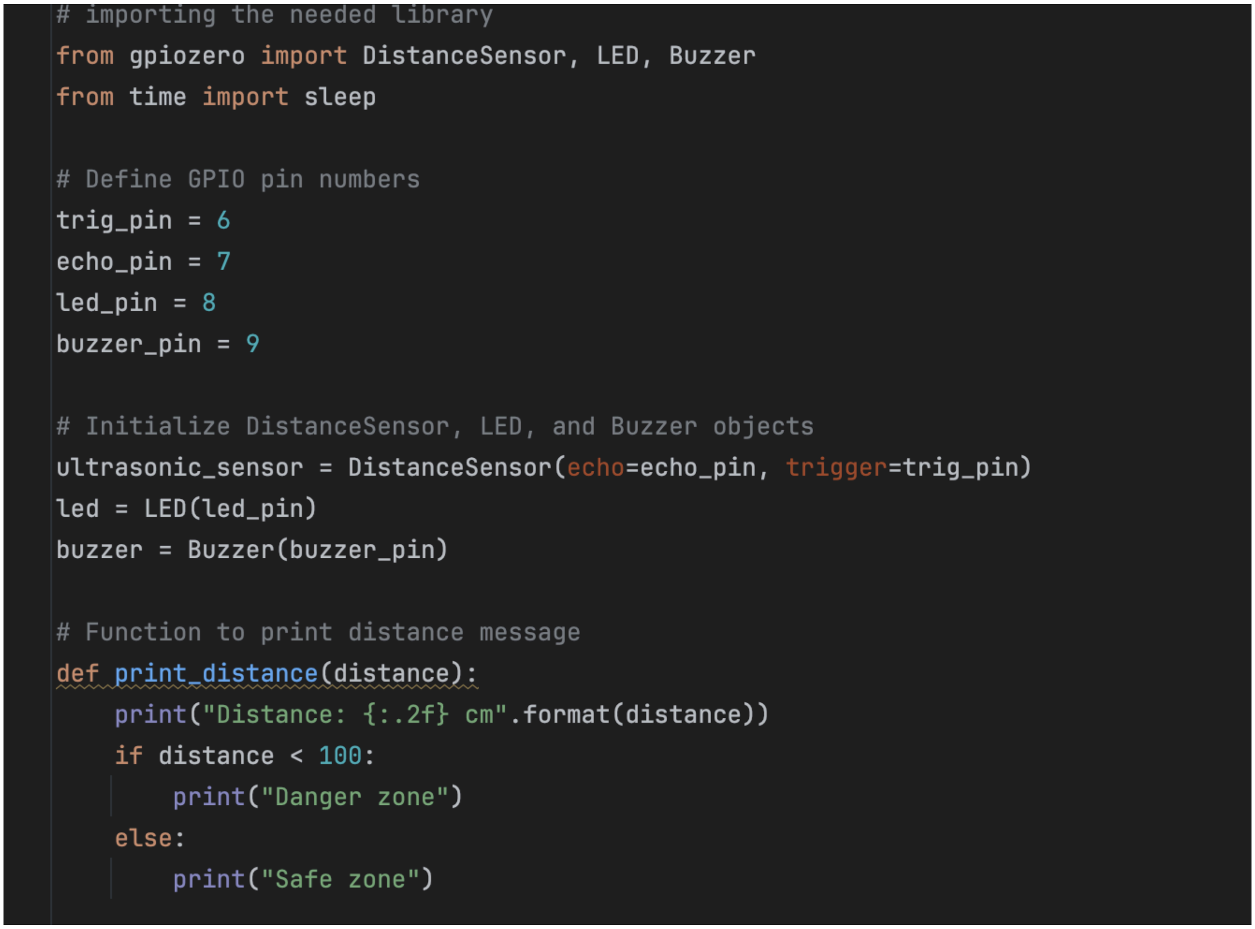
```
  {
    Serial . print ( dist );
    Serial . println (" cm");
  }

  if (dist < 100) // Light up the LED and buzz the buzzer if the distance is less than 100 cm
  {
    digitalWrite(led, HIGH); // Turn on LED
    digitalWrite(buzzer, HIGH); // Turn on buzzer
  }
  else
  {
    digitalWrite(led, LOW); // Turn off LED
    digitalWrite(buzzer, LOW); // Turn off buzzer
  }

  delay (1000) ;
}
```

The circuit has a buzzer, LED, motion sensor, and an Arduino board all connected to a breadboard. When the ultrasonic sensor detects a person or an object within less than 100cm of “the danger zone”, the LED lights up and a buzzer which is also integrated into the system provides an audible warning about the potential danger ahead.

Python code of the program

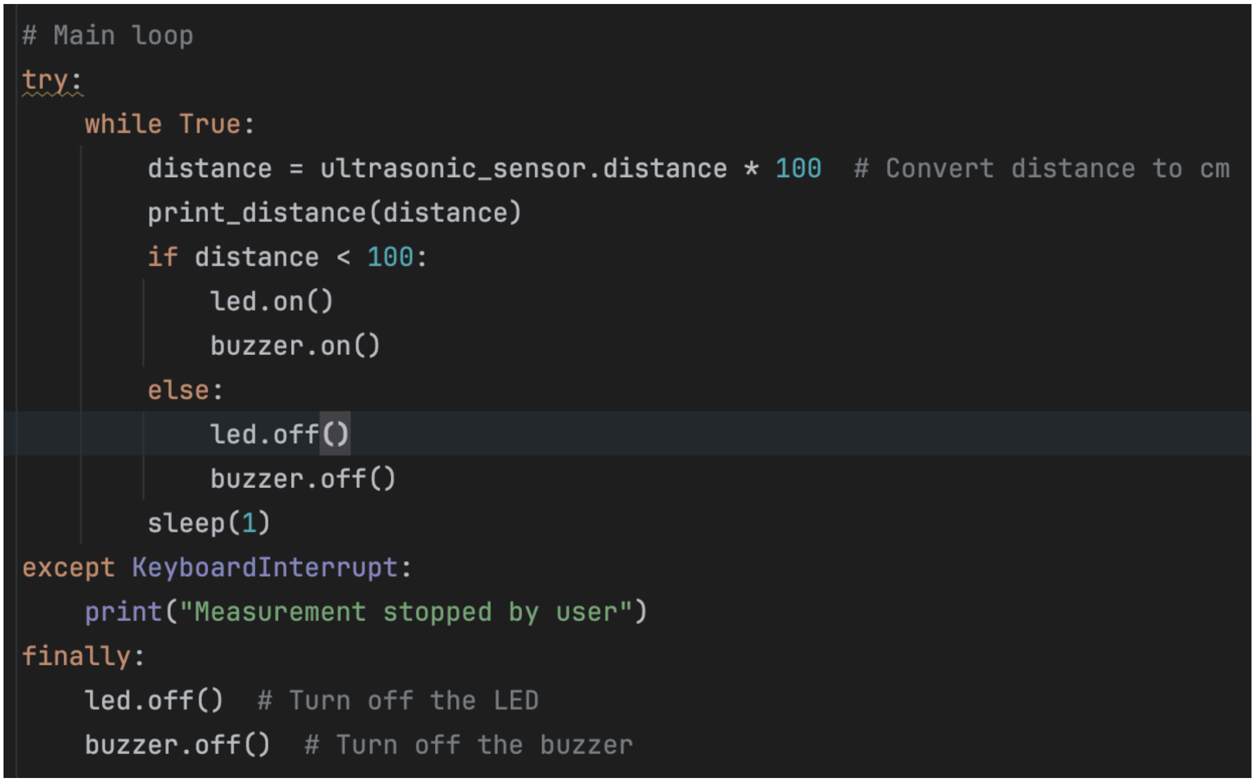


```
# importing the needed library
from gpiozero import DistanceSensor, LED, Buzzer
from time import sleep

# Define GPIO pin numbers
trig_pin = 6
echo_pin = 7
led_pin = 8
buzzer_pin = 9

# Initialize DistanceSensor, LED, and Buzzer objects
ultrasonic_sensor = DistanceSensor(echo=echo_pin, trigger=trig_pin)
led = LED(led_pin)
buzzer = Buzzer(buzzer_pin)

# Function to print distance message
def print_distance(distance):
    print("Distance: {:.2f} cm".format(distance))
    if distance < 100:
        print("Danger zone")
    else:
        print("Safe zone")
```



```python
# Main loop
try:
    while True:
        distance = ultrasonic_sensor.distance * 100  # Convert distance to cm
        print_distance(distance)
        if distance < 100:
            led.on()
            buzzer.on()
        else:
            led.off()
            buzzer.off()
        sleep(1)
except KeyboardInterrupt:
    print("Measurement stopped by user")
finally:
    led.off()  # Turn off the LED
    buzzer.off()  # Turn off the buzzer
```

Group Contribution:

1. Cherub:
- Led the conceptualization and initial design phase of the Smart Security System.
- Wrote and debugged the code for controlling the system's behavior based on sensor inputs.
- Assisted in documenting the project's progress and findings, including writing sections of the project report.

2. Ahmed:
- Conducted research on sensor technologies and their applications in security systems.
- Assisted in documenting the project's progress and findings, including writing sections of the project report.

3. Gerard:
- Worked on the python code ensuring correct functionality.
- Assisted in documenting the project's progress and findings, including writing sections of the project report.

4. Wanzia:
- Contributed to the preparation of the project documentation, ensuring completeness and accuracy of information.

- Conducted extensive testing and troubleshooting to ensure the functionality and reliability of the system.

Each member of our group played a crucial role in the development and implementation of the Smart Security System, leveraging their individual strengths and expertise to achieve our collective goals. Through effective collaboration and dedication, we have successfully created a functional and reliable system that addresses the need for enhanced safety and security in diverse environments.