EMBEDDED SYSTEMS AND IOT

TEAM NAME: **GADGET FREAKS**

PROJECT NUMBER: **66**

PROFESSOR: **PROF. NITHYA S.**

MEMBERS:

AKASH (22BIT0546)

SUDHARSANAN (22BIT0676)

ARNAV VADAN (22BIT0240)

DAKSH RUPANI (21BIT0191)

OM PATRO (21BIT0115)

SUDHANSHU SHUKLA (21BIT0735)

INTRODUCTION

The Internet of Things (IoT) refers to the network of physical objects— “things”—embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. These objects range from ordinary household items to sophisticated industrial tools. IoT enables devices to communicate and interact, both with each other and with centralized systems, to collect and share data.

IoT has a wide array of applications, including smart homes, where devices like thermostats, lights, and security systems can be controlled remotely. In healthcare, IoT devices can monitor patients' vital signs in real-time. In agriculture, IoT can optimize irrigation and monitor crop health. Industrial IoT (IIoT) enhances manufacturing processes through predictive maintenance and improved operational efficiency.

The benefits of IoT include increased efficiency, convenience, and the ability to make data-driven decisions. However, it also raises concerns about security, privacy, and the complexity of managing numerous connected devices. As IoT continues to evolve, it promises to transform various sectors by making them more interconnected and intelligent.

CAR AUTOMATION SYSTEM

A car automation system with a temperature sensor and an ultrasonic sensor enhances vehicle safety and efficiency. The temperature sensor monitors the car's internal and external temperatures, ensuring optimal operating conditions and preventing overheating. This sensor can trigger alerts or automatic adjustments to the car's cooling system, maintaining a safe and comfortable environment for passengers.

The ultrasonic sensor, on the other hand, measures the distance between the car and nearby obstacles. It emits ultrasonic waves that bounce back upon hitting an object, allowing the system to calculate the distance based on the time taken for the waves to return. This sensor is crucial for collision avoidance and parking assistance. When the car approaches an obstacle, the system can alert the driver or automatically adjust the car's direction to avoid a collision.

Integrating these sensors into a car automation system improves overall safety and convenience. The temperature sensor ensures the vehicle operates within safe temperature ranges, while the ultrasonic sensor aids in navigating tight spaces and avoiding accidents. Together, they contribute to a smarter, more responsive driving experience, enhancing both driver and passenger safety.

ARDUINO UNO BOARD

The Arduino Uno is a popular microcontroller board based on the ATmega328P microcontroller. It is widely used in electronics projects due to its simplicity, versatility, and ease of use. The board features 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It can be powered via a USB cable or an external power supply, making it flexible for various applications.

The Arduino Uno is part of the Arduino family, which includes a range of boards and modules designed for different purposes. It is programmed using the Arduino Integrated Development Environment (IDE), which supports C/C++ programming languages. The IDE provides a user-friendly interface for writing, compiling, and uploading code to the board.

One of the key advantages of the Arduino Uno is its extensive community support. There are numerous tutorials, forums, and resources available online, making it accessible for beginners and experienced developers alike. The board is commonly used in projects such as robotics, home automation, wearable technology, and interactive art installations. Its open-source nature allows for continuous innovation and collaboration within the maker community.

SENSORS

1. ULTRASONIC SENSOR

An ultrasonic sensor measures distance by emitting ultrasonic waves and detecting their reflection from objects. It consists of a transmitter that sends out sound waves at a frequency higher than human hearing and a receiver that listens for the echo. The sensor calculates the distance to an object based on the time it takes for the sound waves to return. Ultrasonic sensors are widely used in applications like obstacle detection, distance measurement, and level sensing in various fields, including robotics, automotive systems, and industrial automation. They are valued for their accuracy, reliability, and ability to operate in diverse environments.

2. TEMPERATURE SENSOR

A temperature sensor measures the temperature of an environment or object. It converts temperature data into an electrical signal that can be read by a device or system. There are various types of temperature sensors, including thermocouples, resistance temperature detectors (RTDs), thermistors, and semiconductor sensors. These sensors are used in a wide range of applications, from household appliances like ovens and refrigerators to industrial processes and environmental monitoring. They are valued for their accuracy, reliability, and ability to provide real-time temperature data, which is crucial for maintaining optimal operating conditions and ensuring safety.

ACTUATOR USED

BUZZER

Buzzers offer a more immediate and noticeable form of communication, which is crucial for safety and efficiency in car automation systems.

1. Audible Alerts: Buzzers provide audible alerts, which are more effective in grabbing attention, especially in noisy environments or when the driver is not looking at the dashboard.

2. Immediate Response: Sound can be perceived instantly, ensuring that critical warnings, such as proximity to an obstacle, are noticed immediately.

3. Safety: In situations like drunk driving detection, a buzzer can alert nearby vehicles and pedestrians, enhancing overall safety.

4. Versatility: Buzzers can produce different sounds for various alerts, making it easier to distinguish between different types of warnings.

MATERIAL REQUIRED

1. CARDBOARD (for base)

2. WHEELS

2. ARDUINO UNO BOARD

4. ULTRASONIC SENSOR

5. TEMPERATURE SENSOR

6. CONNECTING WIRES

7. BATTERIES

8. SWITCH

9. MOTORS

CODE FOR THE FOLLOWING

```
// Pin Definitions
#define trigPin 8
#define echoPin 9
// Motor Pins
#define motor1Pin1 2  // Left motor forward
#define motor1Pin2 3  // Left motor backward
#define motor2Pin1 4  // Right motor forward
#define motor2Pin2 5  // Right motor backward
// Constants
const int obstacleThreshold = 20;  // Distance in cm to detect obstacles
void setup() {
  Serial.begin(9600);
  // Sensor Pin Modes
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  // Motor Pin Modes
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(10,OUTPUT);
```

```
  pinMode(11,OUTPUT);
}
void loop() {
digitalWrite(10,HIGH);
digitalWrite(11,HIGH);
  long distance = measureDistance();
  if (distance < obstacleThreshold) {
    Serial.println("Obstacle detected! Stopping.");
    stopCar();
    delay(500);  // Wait a bit before deciding direction
    // Re-check obstacle direction and act accordingly
    if (checkObstacleOnRight()) {
      goBackward();
      delay(1000);  // Move backward for 1 second
      turnLeft();   // Turn left if obstacle is on the right
    } else {
      turnRight();  // Turn right if obstacle is not on the right
    }
  } else {
    moveForward();
  }
  delay(100);
}
// Function to measure distance with ultrasonic sensor
```

```
long measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  long distance = duration * 0.034 / 2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  return distance;
}
// Function to check for obstacle on the right by turning slightly and measuring
bool checkObstacleOnRight() {
  // Turn slightly right to check
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, LOW);
  delay(500);  // Small turn to the right
  long rightDistance = measureDistance();
  // Return to original direction
  stopCar();
```

```
  Serial.print("Right Distance: ");
  Serial.print(rightDistance);
  Serial.println(" cm");
  // Check if there's an obstacle on the right within threshold
  return rightDistance < obstacleThreshold;
}
// Function to move car forward
void moveForward() {
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
digitalWrite(motor2Pin1, HIGH);
digitalWrite(motor2Pin2, LOW);
Serial.println("Moving forward");
}
// Function to stop the car
void stopCar() {
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, LOW);
  Serial.println("Car stopped");
}
// Function to turn the car left
void turnLeft() {
```

```
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(500);  // Turn for half a second
  Serial.println("Turning left");
  stopCar();
}
// Function to turn the car right
void turnRight() {
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
digitalWrite(motor2Pin1, LOW);
digitalWrite(motor2Pin2, HIGH);
  delay(500);  // Turn for half a second
  Serial.println("Turning right");
  stopCar();
}

// Function to go backward
void goBackward() {
  digitalWrite(motor1Pin1,  LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
```

```
  digitalWrite(motor2Pin2, HIGH);

  Serial.println("Going backward");

}
```

OUTPUT

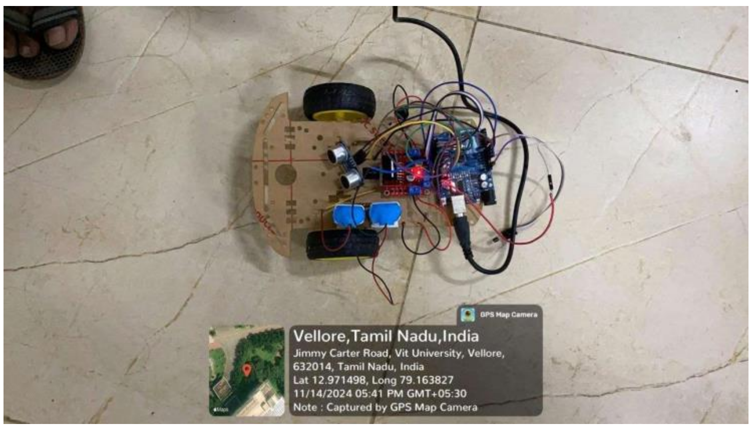
```
Distance: 35 cm
Moving forward
Distance: 34 cm
Moving forward
Distance: 33 cm
Moving forward
Distance: 36 cm
Moving forward
Distance: 32 cm
Moving forward
Distance: 38 cm
Moving forward
```

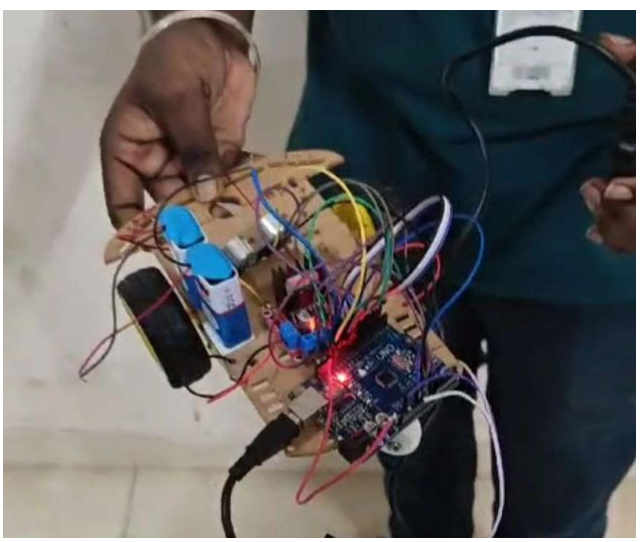
```
Distance: 19 cm
Obstacle detected! Stopping.
Car stopped
Right Distance: 22 cm
Turning right
Distance: 18 cm
Obstacle detected! Stopping.
Car stopped
Right Distance: 15 cm
Going backward
Turning left
```

```
Distance: 18 cm
Obstacle detected! Stopping.
Car stopped
Right Distance: 25 cm
Turning right
Turning right
```

```
Distance: 35 cm
Moving forward
Distance: 33 cm
Moving forward
Distance: 30 cm
Moving forward
```

PROTOTYPE IMAGES





VIDEO LINK

https://drive.google.com/drive/folders/1BByc8SFzrKn9JXHAt_OGTketpXX6-i1W

REFERENCES

1. https://builtin.com/articles/iot-in-vehicles
2. https://www.aimprosoft.com/blog/automotive-iot-use-cases-for-cars-vehicles/
3. https://store.arduino.cc/collections/boards-modules
4. https://ieeexplore.ieee.org/