WT EXPERIMENT - 4

Name	Class	Roll no.
Ansh Sarfare	D15A	49
Vedant Sanap	D15A	47
Jai Talreja	D15A	59

Aim: Component and Arduino Introduction

Introduction:

The Arduino Uno is a versatile and widely used microcontroller board that serves as the foundation for many electronics projects. It is an open-source platform that allows users to design and develop interactive electronic devices. The board consists of an Atmega328 microcontroller that can be programmed to control various hardware components, making it ideal for applications in automation, robotics, IoT, and more.

In this experiment, we will use an Arduino Uno to create an object segregator that differentiates between wet and dry objects using an ultrasonic sensor and a soil moisture sensor. The main goal is to measure the distance of an approaching object, analyze its moisture content, and trigger a servo motor to sort the object accordingly.

Arduino Uno Overview

The Arduino Uno is one of the most popular boards in the Arduino family due to its simplicity, low cost, and ease of use. It features:

- **Microcontroller**: Atmega328P
- **Operating Voltage**: 5V
- **Digital I/O Pins**: 14 (6 can be used as PWM outputs)
- **Analog Input Pins**: 6
- **Flash Memory**: 32 KB
- **Clock Speed**: 16 MHz
- **USB Connection**: For programming and communication
- **Power Supply**: Can be powered via USB or an external power supply

The Arduino Uno can be programmed using the Arduino IDE, which allows users to write and upload code that controls various electronic components connected to the board.

Installation and Setup

Installing the Arduino IDE

1. Download and install the Arduino IDE from Arduino's official website.
2. Connect the Arduino Uno to your computer via USB.
3. Open the Arduino IDE and select the correct board under **Tools > Board > Arduino Uno**.
4. Select the correct port under **Tools > Port**.
5. Upload a sample program to verify the setup.

Running a Test Blink Program

```
void setup() {

  pinMode(LED_BUILTIN, OUTPUT);

}

void loop() {

  digitalWrite(LED_BUILTIN, HIGH);

  delay(1000);

  digitalWrite(LED_BUILTIN, LOW);

  delay(1000);

}
```

If the built-in LED blinks, your setup is working fine.

Components Used

1. **Arduino Uno**: Acts as the main controller for reading sensor data and controlling the servo motor.
2. **Ultrasonic Sensor (HC-SR04)**: Measures the distance of an object.
3. **Soil Moisture Sensor**: Determines whether the detected object is wet or dry.
4. **Servo Motor (SG90 or similar)**: Moves objects based on classification.
5. **Jumper Wires**: Connects the components.
6. **Breadboard**: Provides a non-permanent circuit connection.

Code:

```
#include <Servo.h>

// Pin definitions
const int trigPin = 7;
const int echoPin = 8;
const int servoPin = 9;
const int moisturePin = 2;
const int moistureThreshold = 500;

Servo myServo;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  myServo.attach(servoPin);
  myServo.write(100);
  delay(1000);
}

void loop() {
  long duration = measureDistance();
  long distance = duration * 0.0344 / 2;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print(" cm");

  if (distance < 20) {
    delay(2000);
    bool isWet = digitalRead(moisturePin);

    Serial.print(", Moisture: ");
    Serial.print(isWet);
    Serial.print(" -> ");
    Serial.println(isWet ? "Wet" : "Dry");

    if (isWet) {
```

```
      Serial.println("Wet object detected! Moving clockwise.");
      myServo.write(180);
    } else {
      Serial.println("Dry object detected! Moving counterclockwise.");
      myServo.write(20);
    }
    delay(1500);
    Serial.println("Returning to 100 degrees.");
    myServo.write(100);
    delay(2000);
  } else {
    Serial.println(" - No object detected. Servo remains at 100 degrees.");
  }
  delay(2000);
}

long measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  return pulseIn(echoPin, HIGH);
}
```

Code Breakdown

Libraries

- `#include <Servo.h>`: Includes the Servo library to control the servo motor.

Pin Setup

- **Ultrasonic Sensor**: `trigPin (7)`, `echoPin (8)`
- **Servo Motor**: `servoPin (9)`
- **Moisture Sensor**: `moisturePin (2)`

Setup Function

- Initializes the serial monitor, ultrasonic sensor, and servo motor.
- Moves the servo to the neutral position (100 degrees).

Loop Function

1. Measures the object's distance using `measureDistance()`.
2. If an object is detected within **20 cm**:
    - Reads the moisture sensor.
    - If wet, moves the servo clockwise (180 degrees).
    - If dry, moves the servo counterclockwise (20 degrees).
    - Returns the servo to its initial position.
3. If no object is detected, the servo remains at **100 degrees**.

Distance Measurement Function

- Sends a pulse from the ultrasonic sensor and calculates the time taken for the echo.
- Converts this time into a distance value.

Methodology:

1. **Ultrasonic Sensor Detection:** The ultrasonic sensor continuously monitors for the presence of an object. If an object is detected within a predefined range, the system proceeds to the next step.

2. **Moisture Sensor Analysis:** Once an object is detected, the moisture sensor determines whether it is wet or dry based on its moisture content.

3. **Servo Motor Action:**

    - If the object is wet, the servo motor moves in a specific direction (e.g., clockwise) to place the object in the "wet" section.

    - If the object is dry, the servo moves in the opposite direction (e.g., counterclockwise) to place the object in the "dry" section.

4. **Arduino Processing:** The Arduino collects sensor data, processes it, and controls the servo motor accordingly. It also provides real-time feedback via the Serial Monitor for debugging and monitoring.

Conclusion

This project demonstrates how Arduino can be used to classify objects based on their moisture content. By integrating multiple sensors and actuators, this system provides a simple yet effective way to automate object segregation. This fundamental concept can be extended to industrial sorting applications and environmental waste management systems.