ELECTROSTATIC DETECTER USING ARDUINO

PROJECT REPORT

Submitted by

LAKSHMANAN S - 2022504310

BACHELOR OF ENGINEERING IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY

CHENNAI 600 044

ABSTRACT

The Electrostatic Detector with Arduino project aims to develop a cost-effective and efficient system for detecting static electricity in various environments. Static electricity, which can lead to equipment malfunctions, damage to sensitive electronics, and even safety hazards, is often challenging to monitor without specialized instruments. This project leverages an Arduino microcontroller and a capacitive sensor to detect electrostatic charges in real-time.

The primary objective of this project is to design a simple, user-friendly, and reliable electrostatic detection system that can alert users to the presence of static charge buildup. The system uses a capacitive sensor to measure the electric field generated by static charges on objects or surfaces. This signal is then processed by the Arduino to determine the charge intensity. An LED indicator or buzzer provides visual or auditory feedback to notify users when static electricity reaches a threshold level.

The design utilizes minimal components, making it accessible for both educational purposes and practical applications in industries such as electronics, manufacturing, and laboratory settings. The Arduino platform allows for easy customization of the detection threshold, sensitivity, and alert mechanism, ensuring the system can be tailored to different environments.

KEY WORDS:

Electrostatic detection, Arduino microcontroller, capacitive sensor, static charge monitoring, real-time feedback, safety alert system, electronic component protection, environmental sensing, low-cost solution, user-friendly design.

TABLE OF CONTENTS

INTRODUCTION:

Static electricity is an ever-present phenomenon that can cause significant problems in various environments, particularly in industries dealing with sensitive electronic components. The accumulation of electric charge on surfaces can result in static discharges, which may damage delicate circuits, cause equipment malfunctions, or even pose safety hazards. Detecting static electricity and monitoring its buildup is crucial to minimizing these risks.

To address this challenge, the Electrostatic Detector with Arduino project aims to create a cost-effective, user-friendly solution for real-time static electricity detection. The Arduino platform provides an open-source, customizable solution that allows users to fine-tune the sensitivity and detection thresholds according to specific needs.

Once a static charge is detected, the system alerts users via an LED indicator or a buzzer, providing immediate feedback about the presence of potentially harmful electrostatic buildup. This real-time alert system is essential for preventing damage to sensitive electronic equipment and for ensuring the safety of both personnel and devices in environments where static electricity is a concern, such as electronics manufacturing, laboratories, and cleanrooms.

In conclusion, the Electrostatic Detector with Arduino offers an accessible, practical, and scalable solution for monitoring static electricity. By leveraging Arduino's versatility, this project makes static electricity detection more accessible to both hobbyists and professionals, enhancing safety and protecting electronic equipment from the damaging effects of electrostatic discharge.

OBJECTIVE:

The objective of this project is to design and implement an electrostatic detector system using an Arduino microcontroller. The system will be able to detect and measure the presence of electrostatic charges on various objects. Electrostatic fields are commonly present in various environments, but they are often invisible and difficult to measure without specialized equipment

The project will employ a capacitive sensor or a custom electrostatic field sensor to detect the variations in the electric field caused by charged objects. These sensors will be interfaced with the Arduino, which will process the data and display the results using LEDs or a small LCD screen. The sensor will detect the strength of the electrostatic field, allowing users to identify the presence of static charges on objects without needing direct contact.

In addition to detection, the system will also provide real-time feedback to the user, giving information about the relative strength of the electrostatic charge detected. This feedback will help users understand how charged objects influence their surroundings and allow for safer handling of static-sensitive equipment.

The project will serve as an educational tool for learning about electrostatics, sensors, and microcontroller programming. It will also provide insights into how electrostatic phenomena affect everyday life, from handling electronics to studying natural static electricity. By integrating a sensor with Arduino, this project can be expanded or adapted for various applications, from scientific experiments to practical uses in static-sensitive environments like electronics workshops.

PRINCIPLES OF ELECTRO STATIC DETECTER:

1. Electric Charge:

Electrostatic detection relies on the principle that charged objects create an electric field around them. Static electricity results from an imbalance of positive and negative charges on an object's surface.

2. Electric Field:

A charged object generates an electric field that extends outward from the object. This field can influence nearby conductive materials or sensors, and its intensity is proportional to the amount of charge.

3. Capacitance Changes:

The presence of an electrostatic charge (or the movement of charged particles) can change the capacitance of a system

4. Field-Induced Voltage:

When a charged object approaches a sensor, the electric field induces a voltage difference on the sensor's plates. This voltage change is used to detect the presence and intensity of the charge.

5. Induced Voltage or Current: The change in capacitance due to the electrostatic influence can induce a voltage or a small current in an associated circuit. Even without a direct sensor, this change can be monitored by using the properties of the system's reactive components (like inductors, capacitors, or resistive elements)..

EXISTING SYSTEM:

One existing system for electrostatic detection is the Simco-Ion 5920 Electrostatic Fieldmeter , a widely used device in industrial and laboratory environments to monitor electrostatic fields. This handheld system detects the electric field generated by static charges on surfaces, offering precise measurements of electrostatic field strength in volts per meter (V/m).

The device utilizes a non-contact sensor, which allows for safe detection of electrostatic fields without the need for direct contact with charged objects.

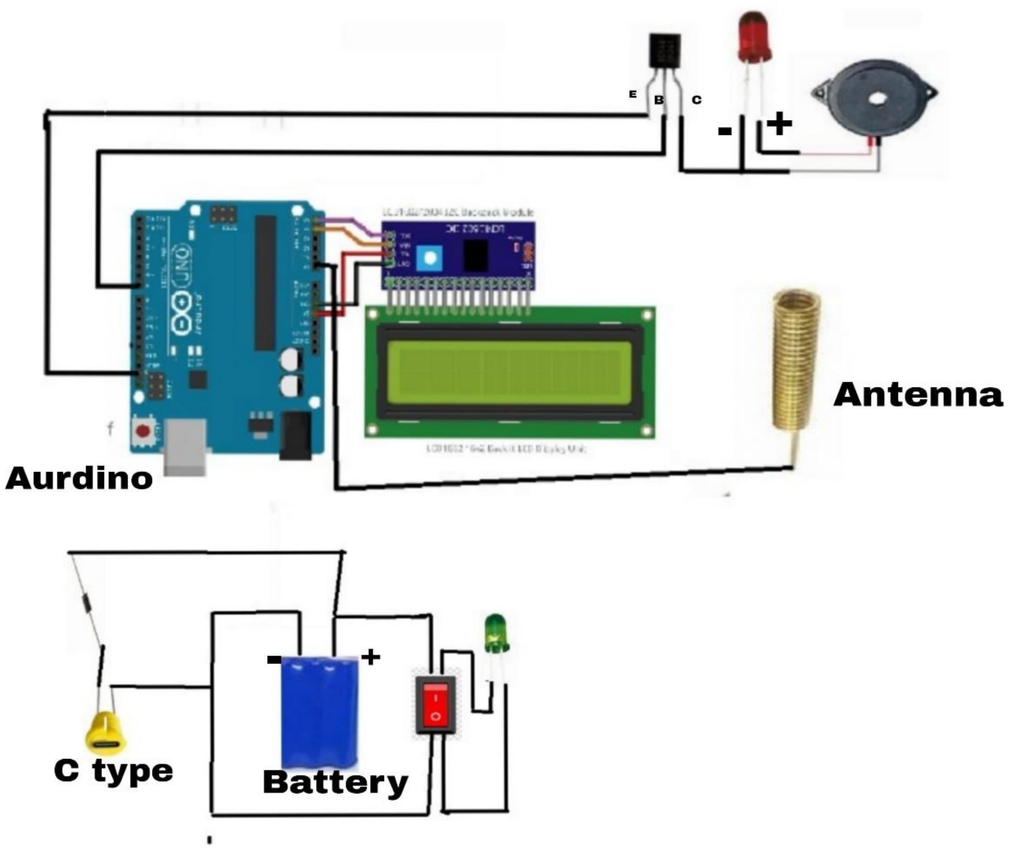
The Simco-Ion 5920 is designed to be highly sensitive and accurate, enabling the detection of static charges over a broad range of voltages. It is capable of measuring both low and high levels of static charge, making it versatile for a variety of applications. The real-time measurement is displayed on a digital screen, providing immediate feedback to the user about the presence and intensity of static electricity in the environment.

This electrostatic fieldmeter is particularly useful in environments where electrostatic discharge (ESD) poses a significant risk, such as in electronics manufacturing, cleanrooms, or chemical processing plants. By detecting and measuring electrostatic fields, it helps identify areas where static buildup could potentially cause damage to sensitive equipment or create safety hazards.

SOURCE CODE :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// LCD configuration
#define LCD_ADDR 0x27  // Default I2C address for most LCD
modules (can be 0x3F for others)
#define LCD_COLS 16    // Number of columns on the LCD
#define LCD_ROWS 2     // Number of rows on the LCD

LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

// Pin definitions
#define LED_PIN 7      // LED connected to Pin 7
#define INPUT_PIN A0   // Analog input at A0

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize LCD
  lcd.init();          // Use lcd.init() instead of lcd.begin()
  lcd.backlight();     // Turn on the LCD backlight
  // Display initialization message
  lcd.setCursor(0, 0);  // Set cursor to top-left corner
  lcd.print("ELECTROSTATIC DETECTOR");
  delay(2000);          // Show the message for 2 seconds

  // Clear LCD for new data
  lcd.clear();

  // Initialize pin modes
  pinMode(LED_PIN, OUTPUT);
  pinMode(INPUT_PIN, INPUT);
}

void loop() {
  // Read analog input value from A0
  int inputValue = analogRead(INPUT_PIN);

  // Print input value to Serial Monitor (optional for debugging)
  Serial.print("Analog Input: ");
  Serial.println(inputValue);

  // Display the input value on the LCD
  lcd.clear();                // Clear the LCD for new data
  lcd.setCursor(0, 0);        // Set cursor to the top-left corner
  lcd.print("Input Value: ");
  lcd.print(inputValue);

  // LED control based on input value
  if (inputValue < 950) {
    digitalWrite(LED_PIN, LOW);  // Turn LED OFF
    lcd.setCursor(0, 1);         // Set cursor to the second line
    lcd.print("OFF");
  } else {
    digitalWrite(LED_PIN, HIGH); // Turn LED ON
    lcd.setCursor(0, 1);         // Set cursor to the second line
    lcd.print("DETECTING");
  }

  // Small delay for stability
  delay(500);
}
```

CIRCUIT DIAGRAM :



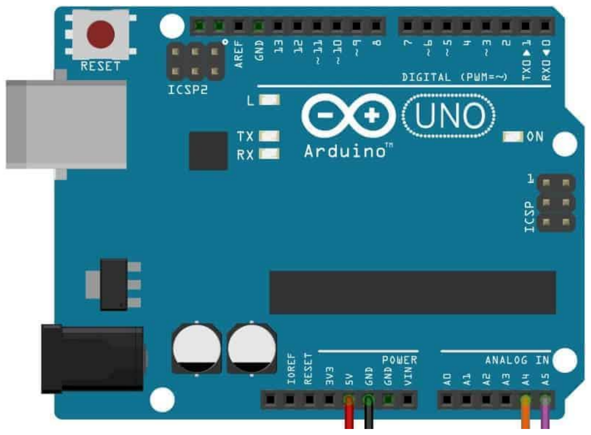
Components :

1. Arduino (Uno)
2. Conductive antenna (copper wire 1dbi)
3. Resistor
4. Pcb board
5. Lcd display
6. Rechargeable battery
7. Diode (IN 40007)
8. Connecting wires (jumpers)
9. Switch
10. LED and Buzzer (for output)

These are the hardware components used in electro static detector with arduino uno.

IN these 10 components each one has separate configuration and each can be doing different operations

1. Arduino (uno):



To build an electrostatic detector with Arduino Uno, use a capacitor and resistor to create an RC circuit that detects capacitance changes from nearby electrostatic fields. The Arduino reads these changes via a digital pin and can trigger an LED or buzzer for feedback.

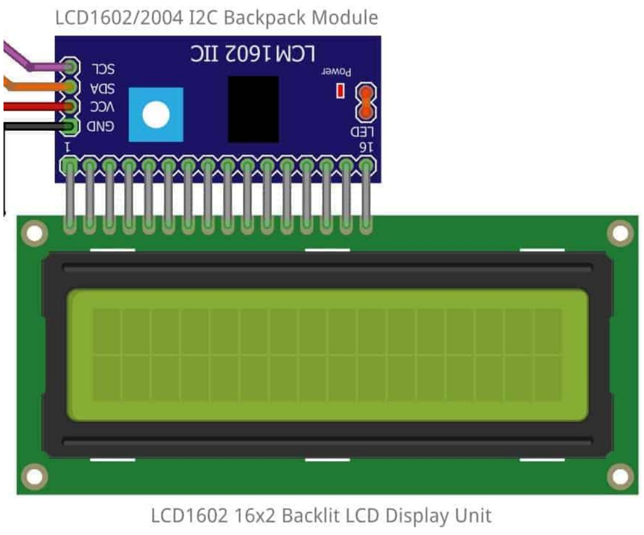
2.Antenna (Copper Wire) :

Copper wire is often used in electrostatic detectors as electrostatic probes or sensor leads. When a charged object is nearby, the copper wire can pick up changes in the electrostatic field, causing a shift in capacitance.

3. Resisters :

Resistors are used in electrostatic detectors to create RC circuits, which help measure changes in capacitance caused by nearby electric fields. These circuits allow the Arduino to detect the presence of electrostatic charges by monitoring charging times. Resistors can also be used in voltage dividers to scale down the signal for the Arduino to read.

4, LCD display :

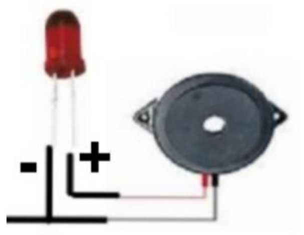

LCD1602/2004 I2C Backpack Module
LCM 1602 IIC
SCL
SDA
VCC
GND
Power
LED
1
16
LCD1602 16x2 Backlit LCD Display Unit

An LCD display in an electrostatic detector shows real-time data, such as the detected capacitance or the presence of an electrostatic field. It can display information like "Charge Detected" or numerical values related to capacitance changes. The Arduino processes the sensor readings and sends the data to the LCD for easy viewing. This helps users monitor electrostatic activity without needing a computer.

5. Diode IN 40007:

A 1N4007 diode in an electrostatic detector can be used for protection and signal conditioning. It protects the circuit from voltage spikes or reverse polarity that could damage sensitive components. It can also be used to rectify AC signals or smooth out irregular voltages, ensuring stable readings. The diode helps the Arduino safely process the signal from the electrostatic sensor.

6. LED and BUZZER :



An LED and buzzer in an electrostatic detector act as indicators for detecting electrostatic charges. The LED can light up when a charge is detected, providing a visual alert. The buzzer can sound when electrostatic fields are present, giving an audible warning. Both components make it easy to know when electrostatic activity is happening without needing to look at a screen.

7. Power Supply :

The Arduino Uno will need a power source, such as a USB connection or a 3.7V battery, to power the system.

8. Wires (for Probing):

Wires or metal plates can be used as probes to detect the electrostatic field. When a charged object comes close, it will induce a small change in capacitance, which the Arduino can measure.

CODE EXPLANATION:

1. Include Necessary Libraries

cpp

Copy code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

- **Wire.h: This library is used for I2C communication, which allows communication with I2C devices like an LCD using only two data lines: SDA (data) and SCL (clock).**
- **LiquidCrystal_I2C.h: This library is used to control LCD screens over I2C communication. It simplifies the process of interfacing with an LCD screen by using fewer wires compared to parallel connections.**

---

2. LCD Configuration

cpp

Copy code

#define LCD_ADDR 0x27 // Default I2C address for most LCD modules (can be 0x3F for others)

#define LCD_COLS 16   // Number of columns on the LCD

#define LCD_ROWS 2   // Number of rows on the LCD

- **LCD_ADDR 0x27: The I2C address of the LCD. Most I2C LCDs use 0x27, but some modules use 0x3F. This address is critical for the communication between Arduino and the LCD.**
- **LCD_COLS 16: Specifies the number of columns on the LCD screen. In this case, the LCD can display 16 characters per row.**
- **LCD_ROWS 2: Specifies that the LCD has 2 rows for displaying information.**

---

3. Create the LCD Object

cpp

Copy code

LiquidCrystal_I2C lcd(LCD_ADDR, LCD_COLS, LCD_ROWS);

- **LiquidCrystal_I2C lcd(...): This creates an instance of the LiquidCrystal_I2C class and assigns it to the variable lcd. The constructor takes the LCD's I2C address (0x27), the number of columns (16), and rows (2). Now, the lcd object can be used to control the LCD screen.**

---

4. Pin Definitions

cpp

Copy code

#define LED_PIN 7   // LED connected to Pin 7

#define INPUT_PIN A0   // Analog input at A0

- **LED_PIN 7: The LED is connected to digital pin 7 of the Arduino. This pin will be used to turn the LED on or off based on the sensor input.**
- **INPUT_PIN A0: The analog sensor is connected to pin A0. This pin will read the analog signal from the sensor, which measures the electrostatic field or capacitance.**

---

5. Setup Function

The setup() function runs only once when the Arduino starts. It is used to initialize various components and set up the system.

cpp

Copy code

```
void setup() {
  // Initialize serial communication
  Serial.begin(9600);
```

- **Serial.begin(9600): Initializes the serial communication at a baud rate of 9600. This allows the Arduino to send data (like sensor readings) to the Serial Monitor for debugging or observation.**

cpp

Copy code

```
  // Initialize LCD
  lcd.init();         // Use lcd.init() instead of lcd.begin()
  lcd.backlight();    // Turn on the LCD backlight
```

- **lcd.init(): Initializes the LCD screen.**

- **lcd.backlight(): Turns on the backlight of the LCD, making the display visible.**

cpp

Copy code

```
  // Display initialization message
  lcd.setCursor(0, 0);  // Set cursor to top-left corner
  lcd.print("System Init");
  delay(2000);          // Show the message for 2 seconds
```

- **lcd.setCursor(0, 0): Moves the cursor to the top-left corner (row 0, column 0) of the LCD to start displaying text.**
- **lcd.print("System Init"): Displays the message "System Init" on the LCD to show that the system is initializing.**
- **delay(2000): Pauses the program for 2 seconds (2000 milliseconds) to give the user time to see the initialization message.**

cpp

Copy code

```
  // Clear LCD for new data
  lcd.clear();
```

- **lcd.clear(): Clears the LCD screen to prepare it for the next set of data (e.g., the sensor input).**

cpp

Copy code

```
  // Initialize pin modes
  pinMode(LED_PIN, OUTPUT);
  pinMode(INPUT_PIN, INPUT);
}
```

- **pinMode(LED_PIN, OUTPUT): Configures pin 7 as an output so that it can control the LED (turn it on or off).**
- **pinMode(INPUT_PIN, INPUT): Configures pin A0 as an input to read data from the sensor.**

---

6. Main Loop

The loop() function runs continuously after the setup() function has completed. This is where the core functionality of the electrostatic detector happens.

cpp

Copy code

```
void loop() {
  // Read analog input value from A0
  int inputValue = analogRead(INPUT_PIN);
```

- **analogRead(INPUT_PIN): Reads the value from the analog sensor connected to pin A0. The value is between 0 and 1023 (10-bit ADC), where higher values correspond to stronger electrical fields (more capacitance).**

cpp

Copy code

```
  // Print input value to Serial Monitor (optional for debugging)
  Serial.print("Analog Input: ");
  Serial.println(inputValue);
```

- **Serial.print("Analog Input: ") and Serial.println(inputValue): Send the sensor's analog reading to the Serial Monitor for debugging purposes. This helps to monitor how the sensor is responding.**

cpp

Copy code

```
// Display the input value on the LCD
lcd.clear();                // Clear the LCD for new data
lcd.setCursor(0, 0);        // Set cursor to the top-left corner
lcd.print("Input Value: ");
lcd.print(inputValue);
```

- **lcd.clear(): Clears the LCD to prepare for the new data to be displayed.**
- **lcd.setCursor(0, 0): Moves the cursor back to the top-left corner.**
- **lcd.print("Input Value: "): Displays the text "Input Value: ".**
- **lcd.print(inputValue): Displays the actual value read from the analog sensor.**

cpp

Copy code

```
// LED control based on input value
if (inputValue < 10) {
  digitalWrite(LED_PIN, LOW);  // Turn LED OFF
  lcd.setCursor(0, 1);        // Set cursor to the second line
  lcd.print("OFF");
} else {
  digitalWrite(LED_PIN, HIGH); // Turn LED ON
  lcd.setCursor(0, 1);        // Set cursor to the second line
  lcd.print("DETECTING");
}
```

- **if (inputValue < 10): Checks if the analog sensor value is less than 10, which might indicate no significant electrostatic field.**
  - **digitalWrite(LED_PIN, LOW): Turns the LED OFF if no field is detected.**
  - **lcd.setCursor(0, 1): Moves the cursor to the second row of the LCD.**
  - **lcd.print("OFF"): Displays the message "OFF" to indicate no detection.**
- **else: If the input value is 10 or higher, it suggests that an electrostatic field has been detected.**
  - **digitalWrite(LED_PIN, HIGH): Turns the LED ON to indicate detection.**
  - **lcd.setCursor(0, 1): Moves the cursor to the second row again.**
  - **lcd.print("DETECTING"): Displays the message "DETECTING" on the LCD.**

cpp

Copy code

```
  // Small delay for stability
  delay(500);
}
```

- **delay(500): Adds a 500-millisecond delay to the loop to give the system some time to stabilize before reading the sensor again. This helps avoid excessive flickering on the display and ensures smoother operation.**

Summary of How the System Works:

- **The analog sensor detects changes in the electrostatic field and provides an analog value to the Arduino.**
- **The Arduino reads this value and displays it on the LCD screen.**
- **If the value is below a threshold (in this case, 10), the LED turns OFF, and the display shows "OFF."**
- **If the value exceeds the threshold, the LED turns ON, and the display shows "DETECTING" to indicate that the system has detected an electrostatic charge.**

This system can be used for detecting nearby electrostatic charges or objects that influence capacitance, and provides both visual feedback on the LCD and audible/visual feedback with the LED.

ADVANTAGES :

1. Non-contact
2. Real-time
3. Cost-effective
4. Simple
5. Versatile

DISADVANTAGES :

1. Insensitive
2. Interference
3. Calibration
4. Limited-range
5. Single-purpose

APPLICATION:

1. Electrostatic Discharge (ESD) Prevention
2. Static Electricity Experiments
3. Dust and Particle Detection
4. Lightning Prediction
5. Quality Control in Manufacturing
6. Environmental Monitoring
7. Safety in Explosive Environments
8. DIY and Hobbyist Projects

Conclusion :

The electrostatic detector with Arduino is a simple, cost-effective project that demonstrates static electricity detection. It is practical for ESD prevention, educational experiments, and DIY applications, with scope for future improvements.

Reference :

1."Electrostatic Discharge Protection and Detection" - A book or article on ESD (Electrostatic Discharge) protection systems, often detailing detection methods.

2. "Arduino-based Electrostatic Field Detector" - Research papers or project tutorials using Arduino platforms for building electrostatic detectors.

3."Principles of Electrostatics in Engineering and Science" by R. L. Stoller - A book explaining the fundamentals of electrostatics, including detection techniques.

4. "Electrostatic Field Detection for Industrial Applications" - Industry guides on the use of electrostatic field detectors in various sectors like electronics manufacturing and cleanrooms.

THANK YOU