**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

In today's technologically advanced world, electronic devices have become an integral part of our daily lives. From smartphones and laptops to smart home devices and wearable technology, the proliferation of electronic gadgets has revolutionized the way we live, work, and communicate. However, with the increasing number of these devices, the need for effective detection and management of electronic devices has become more critical than ever.

This mini project focuses on the development of a system for electronic device detection. The primary objective is to design and implement a solution that can accurately identify and monitor the presence of various electronic devices within a specified area. This system has a wide range of potential applications, including enhancing security in restricted zones, managing device usage in educational or corporate environments, and optimizing network resources in smart homes or offices.

In this project, we will explore the various components and methodologies involved in electronic device detection. We will discuss the challenges faced in the detection process, the technologies employed to overcome these challenges, and the potential benefits of implementing such a system. Through this mini project, we aim to contribute to the growing field of electronic device management and provide a foundation for future research and development in this area.

**1.2 Literature Survey**

A literature survey is a comprehensive summary of previous research on a particular topic. The literature review surveys articles, books, and other sources relevant to a particular area of research. The review should enumerate, describe, summarise, objectively evaluate and clarify the previous research.

This particular project is used to find what and which is a better way to opt and how to understand what are the techniques to use. Electronic device detection has garnered significant attention due to the increasing ubiquity of electronic gadgets in various environments. This literature survey reviews the existing research and technologies in electronic device detection, focusing on methodologies, applications, and the challenges faced in the field.

Electronic device detection has significant implications for security and surveillance focused on using device detection in restricted areas to prevent unauthorized device usage. Their system utilizes signal detection to monitor and control device presence, enhancing security measures. Highlighted the use of electronic device detection in preventing data breaches and maintaining network integrity in corporate environments.

Despite the advancements, several challenges remain in the field of electronic device detection. Signal interference and environmental factors can significantly impact detection accuracy. Their research emphasized the need for robust algorithms that can adapt to dynamic environments and minimize false positives.

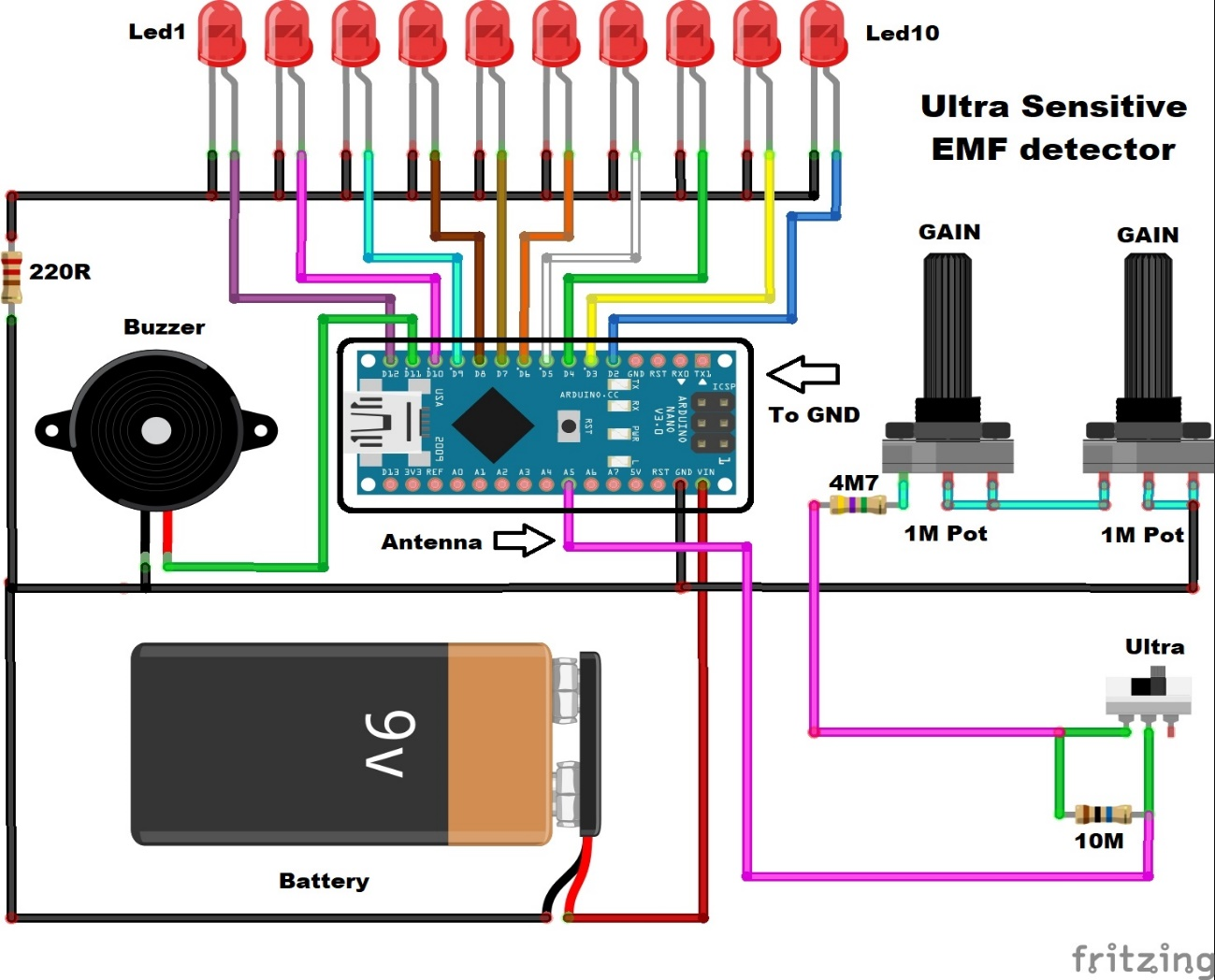
Emerging technologies, such as the Internet of Things (IoT) and 5G networks, present new opportunities and challenges for electronic device detection. The integration of IoT devices with detection systems to enable real-time monitoring and management of electronic devices. Their study demonstrated the potential of leveraging IoT infrastructure to enhance detection capabilities.

To implement this, we’ll deal with ONE types of software namely, Arduino IDE and which are used for simulation purposes. In Arduino IDE, code is written in a normal way using python.

**CHAPTER 2**

**PROJECT OVERVIEW**

**2.1 Schematic Diagram and its Description**



**Components**

1. Arduino Nano: The central microcontroller used to process the signals.
2. LEDs (Led1 to Led10): Indicate the strength of the detected EMF.
3. Buzzer: Provides an audible alert when EMF is detected.
4. 9V Battery: Powers the entire circuit.
5. Antenna: Detects the EMF signals.
6. Potentiometers (1M Pot): Used to adjust the gain of the signal.

vii) Resistors:

i)220Ω (connected to the buzzer)

ii)4.7MΩ (connected to the potentiometers)

iii)10MΩ (connected to the switch)

viii) Switch (Ultra): Turns the circuit on and off.

Connections

i) Power Supply

i) The 9V battery is connected to the Arduino Nano's VIN and GND pins.

ii) The positive terminal of the battery is connected to the VIN pin of the Arduino.

iii) The negative terminal of the battery is connected to the GND pin of the Arduino.

ii) LEDs

i) Each LED is connected to a digital pin on the Arduino (D2 to D11).

ii) The cathode of each LED is connected to the GND rail.

iii) Buzzer

i) One terminal of the buzzer is connected to a digital pin on the Arduino (D12).

ii) The other terminal is connected to the GND rail through a 220Ω resistor.

iv) Antenna

i) The antenna is connected to an analog pin on the Arduino (A0).

v) Potentiometers

i) Two 1MΩ potentiometers are used to adjust the gain.

ii) The wiper (middle pin) of each potentiometer is connected to analog pins on the

Arduino (A1 and A2).

iii) One end of each potentiometer is connected to the GND rail.

iv) The other end is connected to the 4.7MΩ resistor

vi) Switch

i) The switch (Ultra) is connected to the 10MΩ resistor.

ii) One terminal of the switch is connected to the GND rail.

iii) The other terminal is connected to the antenna line.

**2.2 Operation Of Project**

i) When the switch is turned on, the circuit is powered by the 9V battery.

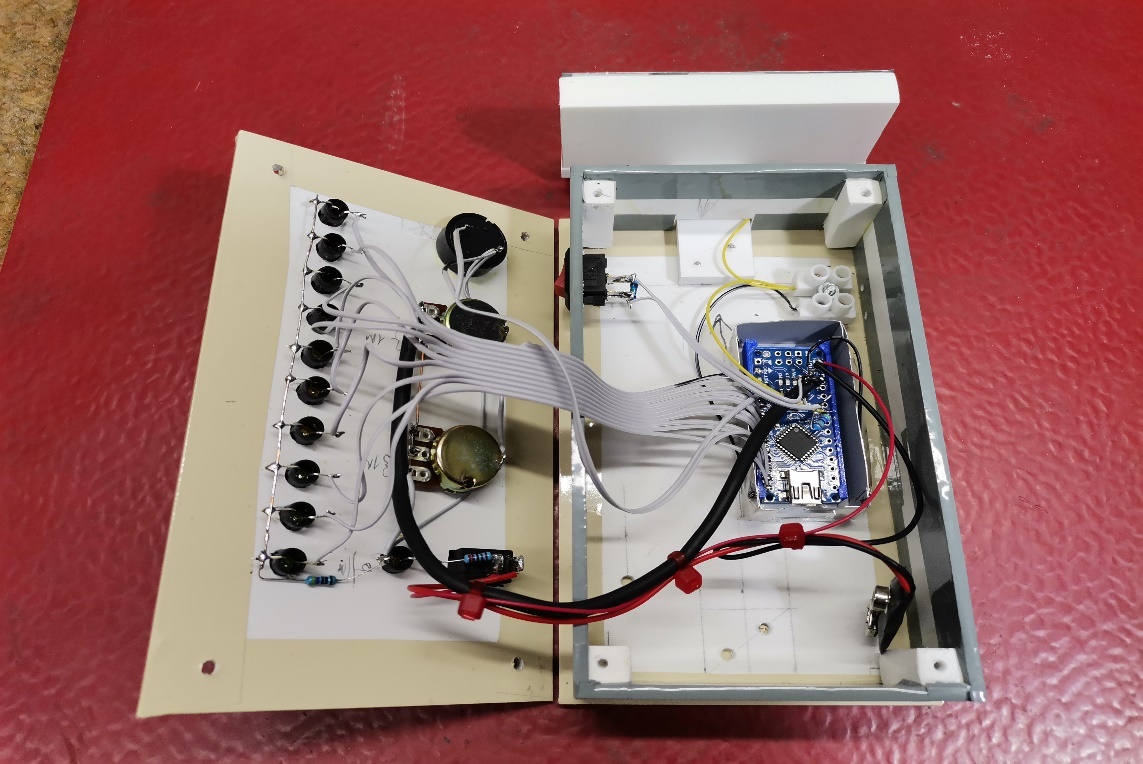
ii) The antenna detects EMF signals and sends them to the Arduino.

iii) The Arduino processes the signals and lights up the LEDs based on the strength of the detected EMF.

iv) The potentiometers allow for adjusting the sensitivity of the detection.

v) The buzzer provides an audible alert when EMF is detected.

**2.3 Images Of The Device**





**CHAPTER 3**

**SOFTWARE DESCRIPTION**

**3.1 Arduino IDE**

The Arduino Integrated Development Environment (IDE) is an open-source platform used for writing and uploading programs to Arduino-compatible boards. It provides a simple and accessible interface for coding and hardware interaction. Here’s a detailed overview of the Arduino IDE:

Features of Arduino IDE

i) Code Edito

i) Syntax Highlighting: Helps distinguish different elements of code, making it easier to read and debug.

ii) Auto-Completion: Suggests functions and variables as you type, speeding up the coding process.

iii) Bracket Matching: Helps in maintaining the correct structure of the code by matching opening and closing brackets.

ii) Libraries

i) Built-in Libraries: Includes libraries for basic tasks like controlling LEDs, sensors, motors, etc.

ii) Third-Party Libraries: Users can download additional libraries from the Arduino Library Manager or from other sources to extend functionality.

iii) Board Manager

i) Allows you to add support for new Arduino boards by installing the necessary packages.

iv) Serial Monitor

i) Used to communicate with the Arduino board. It can display data sent from the board and send data to the board.

v) Sketch Management

i) Sketch: The name for a program written on the Arduino IDE. Each sketch is saved as a file with a ‘.ino’ extension.

ii) Multiple Sketches: You can open multiple sketches in different windows for easy comparison and reference.

vi) Programming Languages

i) The primary language used is a simplified version of C/C++.

ii) Supports standard C/C++ functions.

vii) Cross-Platform

i)Available for Windows, macOS, and Linux.



Fig: Logo of Arduino IDE

**3.2 Arduino IDE code**

// EMF Detector for LED Bargraph v1.0

// 5.12.2009

// original code/project by Aaron ALAI - aaronalai1@gmail.com

// modified for use w/ LED bargraph by Collin Cunningham - collin@makezine.com

// modified by Mirko Pavleski: Dot Led mode + Sound

#define NUMREADINGS 25

// raise this number to increase data smoothing

int senseLimit = 2;

// raise this number to decrease sensitivity (up to 1023 max)

int probePin = 5;

// analog 5

int val = 0;

// reading from probePin

int LED1 = 12;

// connections

int LED2 = 10;

// to

int LED3 = 9;

// LED

int LED4 = 8;

// bargraph

int LED5 = 7;

// anodes

int LED6 = 6;

// with

int LED7 = 5;

// resistors

int LED8 = 4;

// in

int LED9 = 3;

// series

int LED10 = 2;

// variables for smoothing

int readings[NUMREADINGS];

// the readings from the analog input

int index = 0;

// the index of the current reading

int total = 0;

// the running total

int average = 0;

// final average of the probe reading

void setup()

{

pinMode(2, OUTPUT);

// specify LED bargraph outputs

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

pinMode(12, OUTPUT);

Serial.begin(9600);

// initiate serial connection for debugging/etc

for (int i = 0; i < NUMREADINGS; i++)

readings[i] = 0;

// initialize all the readings to 0

}

void loop()

{

val = analogRead(probePin);

// take a reading from the probe

if(val >= 1)

{

// if the reading isn't zero, proceed

val = constrain(val, 1, senseLimit);

// turn any reading higher than the senseLimit value into the senseLimit value

val = map(val, 1, senseLimit, 1, 1023);

// remap the constrained value within a 1 to 1023 range

total -= readings[index];

// subtract the last reading

readings[index] = val;

// read from the sensor

total += readings[index];

// add the reading to the total

index = (index + 1);

// advance to the next index

if (index >= NUMREADINGS)

// if we're at the end of the array...

index = 0;

// ...wrap around to the beginning

average = total / NUMREADINGS;

// calculate the average

int thisPitch = map (average, 50, 950, 100, 1500);

tone(11, thisPitch,120);

if (average > 50 && average < 150)

{

// if the average is over 50 ...

digitalWrite(LED1, HIGH);

// light the first LED

}

Else

{

// and if it's not ...

digitalWrite(LED1, LOW);

// turn that LED off

}

if (average >= 150 && average < 250)

{

// and so on ...

digitalWrite(LED2, HIGH);

}

Else

{

digitalWrite(LED2, LOW);

}

if (average >= 250 && average < 350)

{

digitalWrite(LED3, HIGH);

}

Else

{

digitalWrite(LED3, LOW);

}

if (average >= 350 && average < 450)

{

digitalWrite(LED4, HIGH);

}

Else

{

digitalWrite(LED4, LOW);

}

if (average >= 450 && average < 550)

{

digitalWrite(LED5, HIGH);

}

Else

{

digitalWrite(LED5, LOW);

}

if (average >= 550 && average < 650 )

{

digitalWrite(LED6, HIGH);

}

Else

{

digitalWrite(LED6, LOW);

}

if (average >= 650 &&average < 750)

{

digitalWrite(LED7, HIGH);

}

Else

{

digitalWrite(LED7, LOW);

}

if (average >= 750 && average <850)

{

digitalWrite(LED8, HIGH);

}

Else

{

digitalWrite(LED8, LOW);

}

if (average >= 825 && average <900)

{

digitalWrite(LED9, HIGH);

}

Else

{

digitalWrite(LED9, LOW);

}

if (average > 900)

{

digitalWrite(LED10, HIGH);

}

Else

{

digitalWrite(LED10, LOW);

}

Serial.println(val);

// use output to aid in calibrating

}

}

**CHAPTER 4**

**OUTPUT IMAGES**

OUTPUT image of the device when it detect’s the device







**CHAPTER 5**

**APPLICATIONS**

**5.1 APPLICATIONS**

Security and Law Enforcement

1. Airport Security: EDDS are used to detect unauthorized electronic devices in luggage and on passengers to prevent smuggling and ensure safety.
2. Prison Security: Detecting contraband phones and electronic devices in prisons to prevent unauthorized communication and contraband activities.
3. Building Security: Preventing unauthorized recording devices in sensitive areas such as government buildings, corporate offices, and research facilities.

Educational Institutions:

1. Examination Halls: Ensuring that students do not bring in unauthorized electronic devices that could be used for cheating.
2. Classrooms: Maintaining academic integrity by preventing the use of hidden electronic devices during tests and lectures.

Corporate and Business Environments:

1. Confidential Meetings: Preventing corporate espionage by detecting hidden recording devices during confidential meetings and discussions.
2. Data Centers: Securing data centers from unauthorized electronic devices that could be used for data breaches.

Event and Conference Security:

1. Conferences and Seminars: Ensuring the security of intellectual property and confidential discussions by detecting hidden recording devices.
2. Public Events: Maintaining security by detecting and preventing unauthorized electronic devices in large gatherings.

Healthcare Facilities:

1. Sensitive Areas: Preventing unauthorized devices in sensitive areas such as operating rooms and intensive care units to maintain privacy and security.
2. Data Protection: Ensuring the protection of patient data by detecting unauthorized electronic devices.

Transportation Security:

1. Public Transport: Ensuring the safety of passengers by detecting hidden electronic devices in buses, trains, and other public transport systems.
2. Cargo Security: Detecting hidden electronic devices in cargo to prevent smuggling and ensure the integrity of transported goods.

Military and Defense:

1. Base Security: Detecting unauthorized electronic devices to prevent espionage and maintain the security of military installations
2. Field Operations: Ensuring operational security by detecting hidden electronic devices that could be used for surveillance or tracking.

**CHAPTER 6**

**ADVANTAGES & DISADVANTAGES**

6.1 ADVANTAGES:

1. Acts as a deterrent for individuals considering bringing unauthorized devices into restricted areas, thereby promoting compliance with policies.
2. Enables monitoring of electronic device usage, which can be essential in environments where device usage needs to be regulated or controlled.
3. Helps in protecting sensitive information by detecting unauthorized recording or transmission devices, thus reducing the risk of data breaches.
4. Automates the process of detecting electronic devices, reducing the need for manual checks and increasing operational efficiency.
5. Acts as a deterrent for individuals considering bringing unauthorized devices into restricted areas, thereby promoting compliance with policies.
6. By limiting unauthorized device usage, it can help maintain productivity in workplaces or educational institutions.

6.2 DISADVANTAGES:

1. May generate false alarms by detecting non-threatening devices, leading to unnecessary interruptions and reduced system credibility.
2. Raises privacy issues, as continuous monitoring of personal electronic devices can be seen as intrusive by employees or users.
3. Installation and maintenance of electronic device detection systems can be expensive, making it a significant investment for some organizations.
4. Can be complex to implement and manage, requiring technical expertise and ongoing maintenance to ensure effective operation.
5. Might not detect all types of devices, especially those that are specifically designed to evade detection, thus limiting the system’s effectiveness.
6. Could be misused by organizations to excessively monitor employees or individuals, leading to potential ethical and legal issues.

**CHAPTER 7**

**CONCLUSION**

**REFERENCES**

● Code is from GITHUB

● Reference from IEEE journals.

● Guidance from College Facult