ABSTRACT

The increasing prevalence of electronic devices and the widespread use of wireless communication technologies have raised concerns about the potential health effects of electromagnetic fields (EMFs) on humans. To address these concerns, this project aims to develop an EMF detector using Arduino, a popular open-source microcontroller platform, combined with appropriate components.

The EMF detector system utilizes the Arduino microcontroller board as the core, which provides the necessary processing power and interface capabilities. The system incorporates a copper wire used a Antenna, capable of measuring the strength and intensity of EMFs in the vicinity.

The hardware components, including the Arduino board and the Antenna, are carefully connected and calibrated to ensure accurate and reliable measurements. The software aspect involves programming the Arduino board to read the sensor data and convert it into meaningful units, such as milligauss or volts per meter.

To visualize the detected EMF levels, an output display is integrated into the system. This can be a simple LCD screen on a connected computer or smartphone. The display provides real-time feedback on the strength of the EMF, enabling users to identify and assess potentially high EMF areas.

Furthermore, the project can be extended to include additional features, such as data logging and analysis. By incorporating a data logging mechanism, the system can record EMF measurements over time, enabling users to track and analyze trends in EMF exposure. This information can be valuable in assessing potential risks and implementing necessary precautions.

Overall, this EMF detector using Arduino offers an accessible and affordable solution for individuals concerned about the potential health impacts of EMFs. It provides a tangible means of measuring and monitoring EMF levels, empowering users to make informed decisions regarding their exposure to electromagnetic fields.

ACKNOWLEDGMENT

We would like to place on record our sincere thanks and gratitude to the concerned people, whose suggestions and words of encouragement has been valuable.

We express our heartfelt gratitude to **BNM Institute of Technology**, for giving us the opportunity to pursue Degree of Electronics and Communication Engineering and helping us to shape our career. We take this opportunity to thank **Prof. T. J. Rama Murthy**, Director, **Dr. S.Y. Kulkarni,** Additional Director**, Prof. Eishwar N Maanay,** Dean and **Dr. Krishnamurthy G.N.,** Principal for their support and encouragement to pursue this mini-project. We would like to thank **Dr. P. A. Vijaya,** Professor and Head, Dept. of Electronics and Communication Engineering, for her support and encouragement.

We would like to thank our guide Smt. Anuradha J.P**,** Dept. of Electronics and Communication Engineering**,** who has been the source of inspiration throughout our mini-project work and has provided us with useful information at every stage of our mini-project.

We would also like to thank our Mini-Project coordinator Smt. Anuradha J.P Assistant Professor**,** Dept. of Electronics and Communication Engineering, who is kind enough to extend her help for our mini-project whenever the need arose.

Finally, we are thankful to all the teaching and non-teaching staff of Department of Electronics and Communication Engineering for their help in the successful completion of our mini-project. Last but not the least we would like to extend our sincere gratitude to our parents and all our friends who were a constant source of inspiration.

B A MD Mannan

M Nandeesh

Deepak S S

TABLE OF CONTENTS

Abstract…………………………………………………………………...i

Acknowledgement ..……………………………………………………...ii

List of Figures……………………………………...............................iv

List of Abbreviations……………………………………………………..v

Chapter 1- Introduction 1

1.1 Introduction……………………………………………………………..2

1.2 Motivation……………………………………………………………....3

1.3 Objective and scope…………………………………...………..4

1.4 Organisation of the Report……………………………………………...5

Chapter 2-Literature Survey 6

Chapter 3-Hardware and Software Requirements 8

3.1 Hardware requirements……………………………..……...............9

3.2 Software requirements…………………...…………………………17

Chapter 4-Methodology and Implementation 18

4.1 Block diagram…………………….…………………………………...19

4.2 Code…………………………………………………………………...21

4.3 Images…………………………………………………………………28

Chapter 5-Results 30

Prototype of EMF Detector……….……………………………………31

Chapter 6-Conclusion 32

Conclusion………………..………………………………………………33

References…………………………………………………………………...34

LIST OF FIGURES

Figure No.	Description	Page No
Fig.3.1	Arduino Nano	**15**
Fig.3.2	LCD Display	**16**
Fig.3.3	100k Variable Resistor	**18**
Fig.3.4	Buzzer	**20**
Fig.4.1	Block Diagram	**23**
Fig.4.2	Pictures of the Project	**33**

LIST OF ABBREVIATIONS

1. EMF- Electromagnetic field
2. LCD- Liquid crystal display
3. LED- Light emitting diode
4. PCB- Printed circuit board
5. RF- Radio frequency signals
6. EMW- Electromagnetic waves
7. EMC- Electromagnetic Compatibility
8. Vin- Input voltage pin
9. TxD- Transmission Data
10. RxD- Receive Data
11. RST- Reset
12. RW- Read/Write

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Introduction

An electromagnetic field (EMF) detector is a device that measures the strength and presence of electromagnetic radiation in its surroundings. It is often used to detect and measure the levels of electromagnetic fields emitted by various electronic devices, power lines, or other sources.

In this guide, we will explore how to build an EMF detector using an Arduino microcontroller. Arduino is an open-source electronics platform that provides a simple and cost-effective way to create interactive projects. By combining Arduino with a few additional components, we can construct a basic EMF detector that can detect and display the intensity of electromagnetic fields.

Before we proceed, it's important to note that this project aims to create a basic EMF detector for educational purposes. The accuracy and sensitivity of the detector may vary depending on the components used and the environment in which it is deployed. For professional applications requiring precise measurements, specialized equipment is recommended.

Now, let's delve into the components and steps required to build an EMF detector using Arduino.

1.2 Motivation

There can be several motivations behind creating an EMF (Electromagnetic Field) detector using Arduino:

1. **Curiosity and Learning:** Arduino is a popular platform for DIY electronics projects, and building an EMF detector can be a fascinating way to explore the world of electromagnetic fields, electronics, and sensors. It allows you to gain a deeper understanding of how electromagnetic fields work and how they can be measured.
2. **Personal Safety:** EMF detectors are commonly used to measure the intensity of electromagnetic fields in the environment. Some people are concerned about potential health risks associated with prolonged exposure to high levels of electromagnetic radiation. By building your own EMF detector, you can monitor the electromagnetic field levels around you and make informed decisions regarding your exposure.
3. **Electromagnetic Compatibility (EMC):** EMF detectors are valuable tools for assessing the electromagnetic compatibility of electronic devices. They can help identify sources of electromagnetic interference (EMI) and evaluate the effectiveness of shielding measures. By building an EMF detector, you can troubleshoot and optimize the performance of your electronic circuits or devices.
4. **Paranormal Investigations:** Some individuals are interested in paranormal activities and believe that fluctuations in electromagnetic fields can be an indicator of the presence of ghosts or other supernatural phenomena. Building an EMF detector can be a fun project for those interested in exploring the paranormal.
5. **Environmental Monitoring:** EMF detectors can be used to measure electromagnetic pollution in the environment, such as near power lines, electrical transformers, or high-voltage equipment. By creating your own EMF detector, you can contribute to monitoring and raising awareness about potential sources of electromagnetic pollution.

Overall, building an EMF detector using Arduino provides an opportunity to combine technical skills, scientific curiosity, and practical applications in various fields.

1.3 Objective and Scope

Objectives:

- To Design a EMF detector which would be sensitive enough to detect low levels of electromagnetic radiation.

  The EMF detector should provide accurate and precise measurements of the electromagnetic fields it detects, with minimal error or variability.
- The EMF detector should be able to detect and measure electromagnetic fields across a wide range of frequencies, from radio waves to microwaves and beyond.
- The EMF detector should be portable and easy to use, allowing users to move it around and take measurements in different locations.
- The EMF detector should have a display that shows the measurements in realtime, and it should be able to log and store data for further analysis and interpretation.
- The EMF detector should be affordable and accessible to a wide range of users, without compromising its performance and accuracy.

Scope:

- It involves the design and prototyping of the device, including the selection of components, circuit design, and testing.
- To evaluate the device based on its performance, including sensitivity, range, accuracy, and precision.
- The device may need to comply with safety and regulatory requirements.
- The data collected by the device with reference to electromagnetic field measurements will be used for analysis and identification of patterns and trends in the data.

1.4 Organisation of the Report

Chapter 1: It provides an overview of EMF detector using Arduino Nano, motivation and objective and scope of EMF detector.

Chapter 2: Explains about the research works conducted in the area of EMF detector and the methodology adopted.

Chapter 3: Describes the various circuits employed towards the implementation of the proposed work and the software requirements.

Chapter 4: Gives an insight about the methodology and implementation of the project.

Chapter 5: Result obtained upon working of the prototype.

Chapter 6: Provides conclusion and summarises the entire work.

CHAPTER 2
LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

The first EMF detector was invented by Heinrich Hertz in 1887. Hertz was a German physicist who was studying the behavior of electromagnetic waves. He built a device that could detect the presence of these waves, and he used it to prove the existence of radio waves. Hertz's EMF detector was a simple device. It consisted of a spark gap and a loop of wire. When the spark gap was fired, it produced a short pulse of electromagnetic radiation. This radiation would induce a current in the loop of wire, which could be detected by a galvanometer.Modern EMF detectors are much more sophisticated than Hertz's original device. They use a variety of different sensors to measure the strength of electromagnetic fields. These sensors can measure fields over a wide range of frequencies, and they can be used to detect fields from a variety of sources.One of the main differences between modern EMF detectors and Hertz's original device is that modern detectors are much more sensitive. They can detect fields that are much weaker than the fields that Hertz could detect. Another difference between modern EMF detectors and Hertz's original device is that modern detectors can measure the direction of electromagnetic fields.

To build an EMF detector we have referred Mirko pavleski's paper published on hackster.io an AVNET COMMUNITY, So in our project we are using an antenna that is a copper wire of 1.5mm diameter which will be used as a sensor for detecting EMF radiations. we are using an arduino nano microcontroller to capture the data from antenna using its analog pins and display the EMF intensity on 16x2 LCD display using its digital pins. We connect the antenna to a sensitivity switch whose either ends are connected to two resistors in series with sensor, the middle pin of the switch is given to common ground.

The two resistors value is selected in such a way that detector will be able to detect weak and strong EMF Radiations. Greater the value of resistor, Lesser is the sensitivity of the detector, Lesser the value of resistor, Greater is the sensitivity of EMF and it can detect weak EMF radiations or small change in EMF radiations.

CHAPTER 3

HARDWARE AND SOFTWARE REQUIREMENTS

CHAPTER 3

HARDWARE AND SOFTWARE REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

- **ARDUINO NANO:**

  The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P microcontroller (MCU) and developed by Arduino.cc and initially released in 2008. It offers the same connectivity and specs of the Arduino Uno board in a smaller form factor.

  - **TX and RX pins:** are used for serial communication. The RST pin is used to reset the Arduino Nano. The GND pin is the ground pin.
  - **Digital pins (D2-D13) :** can be used as either input or output pins. When used as an input pin, the pin can be used to read a digital signal from a sensor or switch. When used as an output pin, the pin can be used to drive a LED or other digital device.
  - **Analog pins (A0-A7) :** can be used to read analog signals from sensors. The analog signals are represented as a voltage between 0 and 5 volts. The Arduino Nano has a built-in analog-to-digital converter (ADC) that can be used to read the analog signals from the analog pins.
  - **Vin pin:** It is the power input pin.
  - **5V pin:** It is the regulated 5 volt power output pin.
  - **3V3 pin:** It is the 3.3 volt power output pin.
  - **IOREF pin:** It is the reference voltage pin.
  - **ISP pin:** It is the in-system programming (ISP) header.
  - **16MHz Crystal:** A 16MHz crystal is a type of resonator that is used to create an oscillating signal at a frequency of 16 million hertz. It is a small, square, two-lead component that is typically made of quartz.
  - **Atmega328P Microcontroller:** The ATmega328P is an 8-bit microcontroller based on the AVR enhanced RISC architecture. It has 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM.
  - **Mini USB Jack:** it is a cable used for programming the Arduino via computer.

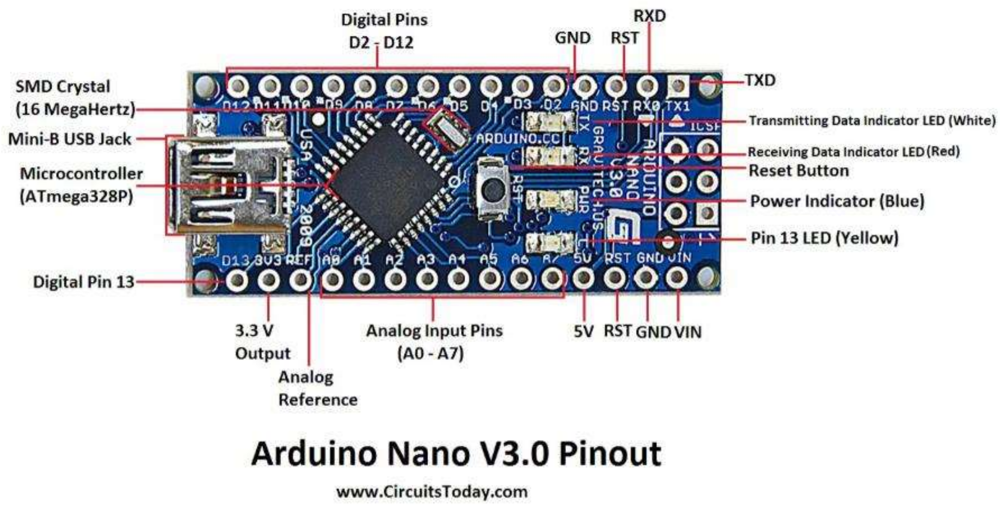


Fig. 3.1: Arduino Nano

- **LCD(Liquid crystal display):**

The LCD (Liquid Crystal Display) module used in the EMF detector is typically a character-based LCD that can display alphanumeric characters. The most common types used are 16x2 and 20x4 LCD modules, which refer to the number of characters each line can display.

❖ **Display Area**: The display area consists of rows and columns where characters can be shown. For a 16x2 LCD, there are two rows with 16 columns each, while a 20x4 LCD has four rows with 20 columns each.

❖ **Backlight**: The backlight provides illumination for the LCD display, allowing the characters to be visible in low-light conditions. It usually consists of an LED that can be controlled to turn the backlight on or off.

❖ **Contrast Adjustment**: LCD modules often have a contrast adjustment feature to control the visibility of the characters. It is usually done using a potentiometer connected to the VO (Voltage Out) pin of the LCD module.

❖ **Power Pins**:

- **VCC** (Power): The VCC pin is connected to the positive power supply (typically 5V) to power the LCD module.
- **GND** (Ground): The GND pin is connected to the ground (0V) reference of the power supply.

❖ **Control Pins**:

- **RS** (Register Select): The RS pin is used to select between data and command modes. When RS is low (0), the data sent to the LCD is interpreted as a command. When RS is high (1), the data sent to the LCD is interpreted as character or data.
- **RW** (Read/Write): The RW pin is used to control the direction of data transfer. When RW is low (0), the LCD is in write mode, allowing data to be written to the module. When RW is high (1), the LCD is in read mode.
- **E** (Enable): The E pin is used to enable or disable the data transfer to the LCD module.
- **Data Pins**: The data pins (usually D0-D7) are used to send 4-bit or 8-bit data to the LCD module, depending on the mode of operation. These pins are used to transfer character or command data.
- **Contrast Adjustment**: The contrast of the LCD display can be adjusted using a potentiometer. It allows you to control the visibility and readability of the characters on the screen.

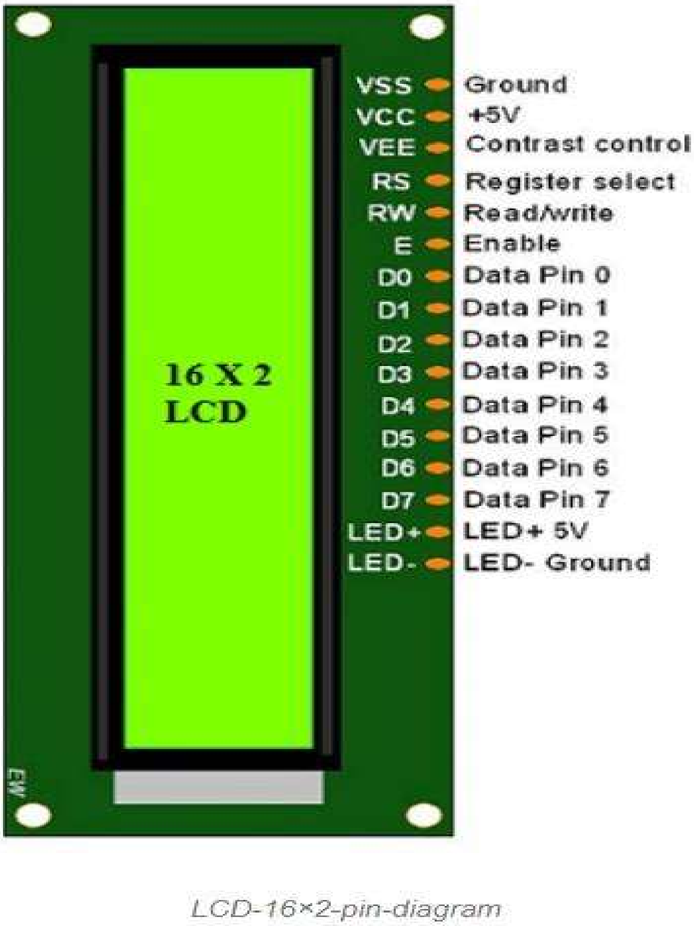


Fig.3.2: 16x2 LCD Display

- **CONNECTING CABLE:**
  - ❖ To connect the components of a EMF detector you will need the following cables:
  - ❖ Jumper wires (male-to-male, female-to-male, and female-to-female).
  - ❖ Power cables (e.g. USB cable for powering the microcontroller).

- **100k VARIABLE RESISTOR:**

  A 100k variable resistor, also known as a potentiometer, is a type of electrical component that allows you to adjust the resistance along its track. It consists of three terminals: two outer terminals and one middle terminal.

❖ **Terminal Configuration:**

- The two outer terminals of the potentiometer are typically labeled as "1" and "3".
- The middle terminal, often referred to as the wiper or slider, is labeled as "2".

❖ **Resistance Range:**

- The 100k notation indicates that the resistance value of the potentiometer can be adjusted between 0 ohms and 100,000 ohms (or 100 kilohms).
- By turning the knob or shaft of the potentiometer, you can vary the resistance anywhere within this range.

❖ **Voltage Divider:**

- In an EMF detector circuit, the 100k variable resistor can be used as part of a voltage divider.
- A voltage divider is a simple circuit that divides the input voltage based on the resistance values.
- By connecting one end of the potentiometer to the input voltage and the other end to the ground (GND), you create a fixed resistance path.
- The middle terminal of the potentiometer is then connected to the input of the EMF sensor module or any other component that requires a variable voltage.

❖ **Adjusting Sensitivity:**

- In an EMF detector, the variable resistor allows you to adjust the sensitivity of the sensor.
- By varying the resistance, you can change the voltage level being supplied to the sensor module, which affects the detection threshold of the EMF signals.
- Increasing the resistance reduces the voltage supplied to the sensor, making it less sensitive to weak electromagnetic fields. Decreasing the resistance increases sensitivity.

- **Calibration and Fine-tuning:**
  - The 100k variable resistor enables calibration and fine-tuning of the EMF detector.
  - During the calibration process, you can adjust the potentiometer to a specific resistance value that corresponds to a known EMF reference point.
  - This calibration ensures accurate measurement and interpretation of the EMF values detected by the sensor.

Fig. 3.3: Variable Resistor

- **<u>ANTENNA</u>:**

When building an EMF detector, a copper wire antenna is often used to capture and detect electromagnetic fields. Here is a detailed description of the copper wire antenna and its role in the EMF detector:

- **Antenna Purpose:** The antenna serves as a receiving element in the EMF detector. It captures the electromagnetic waves present in the environment and converts them into electrical signals that can be processed by the detector circuit.
- **Copper Wire Selection:** Copper wire is commonly used for antennas due to its excellent electrical conductivity. It allows efficient reception of electromagnetic signals. The gauge or thickness of the copper wire depends on the specific requirements of the detector circuit and the desired sensitivity.
- **Length of the Antenna:** The length of the copper wire antenna is an important factor in determining its reception capabilities. The ideal length of the antenna is often a quarter wavelength or a multiple of the wavelength of the target frequency. However,

for general-purpose EMF detectors, a length of approximately one meter (3 feet) is commonly used. This length provides reasonable sensitivity across a range of frequencies.

- **Antenna Construction:** To construct the antenna, strip both ends of the copper wire to expose the conductive copper. Attach one end of the wire to the designated input or antenna pin on the EMF detector circuit. Ensure a secure connection, such as soldering or using a connector, to maintain a good electrical connection.
- **Antenna Orientation:** The orientation of the copper wire antenna can influence its sensitivity to different types of electromagnetic fields. Experimentation with different orientations can help identify the most optimal position for detecting specific types of EMF sources.
- **Grounding**: In some cases, it may be beneficial to ground the EMF detector circuit to reduce noise and interference. This can involve connecting a separate wire from the ground pin of the circuit to a grounding point, such as a metal rod or the ground terminal of a power outlet.

• BUZZER:

In an EMF detector, a buzzer is typically used to provide an audible indication when the detected electromagnetic field exceeds a certain threshold. The buzzer emits a sound or tone to alert the user when there is a significant presence of electromagnetic radiation.

- **Buzzer Type:** The buzzer used in an EMF detector is usually an active buzzer, also known as a self-oscillating or self-driven buzzer. It is different from a passive buzzer, which requires an external oscillating circuit to produce sound. An active buzzer has a built-in oscillator that generates the desired tone when a voltage is applied.
- **Operating Voltage:** Buzzer modules for Arduino and other microcontroller-based projects typically operate at a voltage between 3V and 5V. Make sure to check the specifications of your specific buzzer to determine its operating voltage.
- **Pins:** A buzzer usually has two pins, labeled as positive (+) and negative (-) or signal (+) and ground (-). The positive/signal pin is connected to a digital pin on the Arduino or a microcontroller, while the negative/ground pin is connected to the ground (GND) pin on the Arduino.
- **Sound Generation:** When a voltage is applied to the positive/signal pin of the buzzer, the internal oscillator circuit generates an oscillating signal, which causes the buzzer

to vibrate and produce sound. The frequency of the sound produced is determined by the characteristics of the internal oscillator circuit.

- **Sound Intensity:** The intensity or volume of the sound produced by the buzzer can vary depending on the design and specifications of the specific buzzer module. Some modules may have a built-in potentiometer that allows you to adjust the volume by varying the voltage applied to the buzzer.
- **Control and Timing:** To control the buzzer in an EMF detector, you can use a digital pin on the Arduino or a microcontroller. By setting the digital pin to HIGH, you activate the buzzer and it starts producing sound. Setting the pin to LOW turns off the buzzer. You can control the timing and duration of the buzzer sound by using appropriate programming techniques.

Fig 3.4: Buzzer

Other hardware components like Battery, Sensitivity switch, LED, Resistors, Toggle switch and wires are used in EMF detector .

3.1 SOFTWARE REQUIREMENTS:

- The Arduino IDE (Integrated Development Environment) is a software application used for writing, compiling, and uploading code to Arduino boards. It provides an easy-to-use interface for beginners and advanced users to program their Arduino microcontrollers.
- To use the Arduino IDE, you can follow these steps:
- Connect Arduino Board: Connect your Arduino board to your computer using a USB cable. Make sure your board is properly recognized by your computer (drivers might need to be installed).
- Launch Arduino IDE: Open the Arduino IDE that you installed in the first step.
- Select Board and Port: From the "Tools" menu, select the appropriate Arduino board you are using (e.g., Arduino Uno, Arduino Mega). Also, choose the correct port to which your Arduino board is connected.
- Write Code: In the Arduino IDE, you will see a text editor where you can write your Arduino code. You can start with the basic "Blink" example, which is often used to test the connection. Write or paste your code into the editor.
- Verify and Compile: Click on the "Verify" button (checkmark icon) to compile your code. The IDE will check for any syntax errors in your code. If there are errors, you will need to fix them before proceeding.
- Upload to Arduino: Once your code is successfully compiled, click on the "Upload" button (right arrow icon) to upload the code to your Arduino board. The IDE will compile the code again, and if everything goes well, it will upload the compiled binary to the Arduino.
- Monitor Serial Output: You can monitor the output of your Arduino board by opening the Serial Monitor. From the "Tools" menu, select "Serial Monitor." You can use the Serial Monitor to send and receive data between your Arduino and the computer.
- That's a basic overview of using the Arduino IDE. It offers many more features and libraries to help you develop complex Arduino projects. You can explore the official Arduino website, online tutorials, and the vast Arduino community for more information and examples to get started with your Arduino programming journey.

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

CHAPTER 4

METHODOLOGY AND IMPLEMENTATION

4.1 Block Diagram:

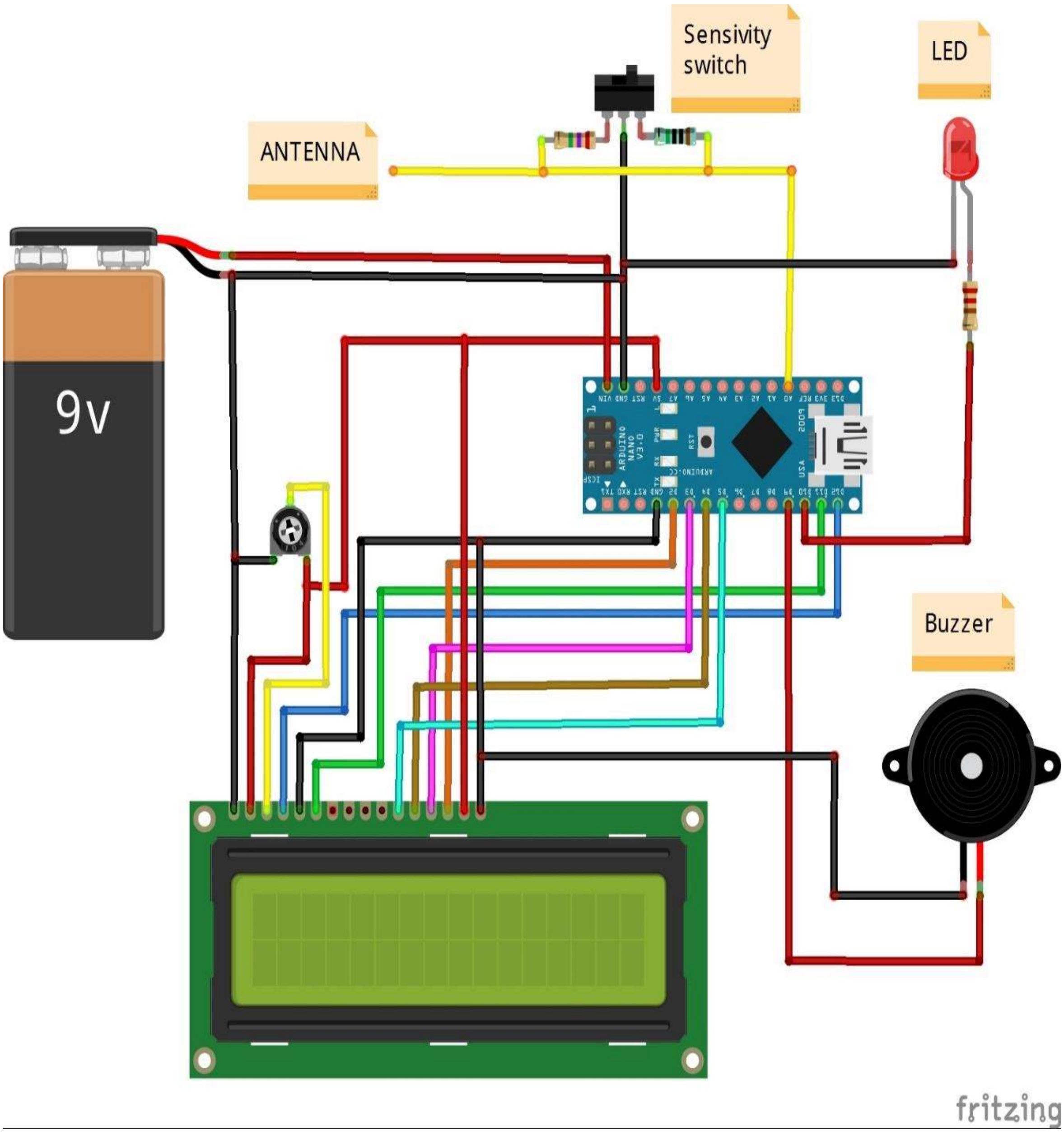


Fig. 3.5: Block diagram

This is a simple device capable of detecting very weak electromagnetic fields. The relative field intensity is displayed on the LCD display and at the same time are given a buzzer sound signalization and LED light signalization. In this case the sensor is a plain copper wire, with a 1.5mm diameter, but you can use any piece of wire or metal tile. Sensitivity can be adjusted via code, and also by changing the value of the resistor connected between A0 and grounding. With the help of a switch, one of the two values of the resistor is selected, and thus the degree of sensitivity of the device. So we can easily calibrate it, by comparing it to a recalibrated industrial device.

As can be seen below the circuit is very simple and consists of Arduino Nano microcontroller and several external components.

The code is a combination of two parts (Arduino based VU meter by KT Audio for LCD display part, and Aaron ALAI EMF Detector for sensor part) and also modifications to certain parts of it for the purpose of greater stability of the whole device. You can download it on link given below.

As you can see in the video, this device can easily detect electromagnetic fields generated by power cables that are only under voltage and not connected to a consumer. For example, an electromagnetic field from an old CRT monitor can be detected at a distance of 3m and more.

The detector is mounted in a convenient housing and is powered by a 9V battery.

4.2 THE CODE:

```
#include <LiquidCrystal.h>

byte Bar[8] = {B11111,B00000,B11111,B11111,B11111,B11111,B00000,B11111};

byte L[8] = {B00111,B01000,B10100,B10100,B10100,B10111,B01000,B00111};

byte R[8] = {B00111,B01000,B10110,B10101,B10110,B10101,B01000,B00111};

byte EndMark[8] = {B10000,B01000,B00100,B00100,B00100,B00100,B01000,B10000};

byte EmptyBar[8] =
{B11111,B00000,B00000,B00000,B00000,B00000,B00000,B11111};

byte peakHoldChar[8] =

{B11111, B00000, B01110, B01110, B01110, B01110, B00000, B11111};

String main_version = "1.0";

int  right;                    //Variables to store and calculate the channel levels

const int numReadings = 5;          //Refresh rate. Lower value = higher rate.

int indexR = 0;

int totalR = 0;

int maxR = 0;

int inputPinR = A0;                 //Input pin Analog 0 for RIGHT channel

int volR = 0;

int rightAvg = 0;

long peakHoldTime = 100;           //peak hold time in miliseconds

long peakHold = 0;

int rightPeak = 0;

long decayTime = 0;

long actualMillis = 0;

int pin10 = 10;        // output of red led
```

```
int val = 0;

int pin9 = 9;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);          //lcd configuration

void setup()

{

 lcd.begin(40, 2);               //Setting up LCD. 16 chars and 2 rows

 lcd.createChar(1, Bar);

 lcd.createChar(3, R);

 lcd.createChar(4, EmptyBar);

 lcd.createChar(5, EndMark);

 lcd.createChar(6, peakHoldChar);

 //Showing loading  message and loading bar

 String KTAudio = "    EMF";

 for (int i = 0; i <= 16; i++)

 {

  lcd.setCursor(0, 0);

   lcd.print(KTAudio.substring(0, i));

  delay(50);

 }

 KTAudio = "  EMF-detector " + main_version;

 for (int i = 0; i <= KTAudio.length(); i++)

 {

  lcd.setCursor(0, 1);

  lcd.print(KTAudio.substring(0, i));

  delay(50);
```

```
  }
  delay(500);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Loading...");
  for (int i = 0; i < 16; i++)
  {
    lcd.setCursor(i, 1);
    lcd.write(4);
  }
  for (int i = 0; i < 16; i++)
  {
    lcd.setCursor(i, 1);
    lcd.write(1);
    delay(50);
  }
  delay(500);
  lcd.clear();
  decayTime = millis();
}
void loop()
{
   lcd.setCursor(0, 0);
   lcd.write(" EMF-intensity");
  actualMillis = millis();
```

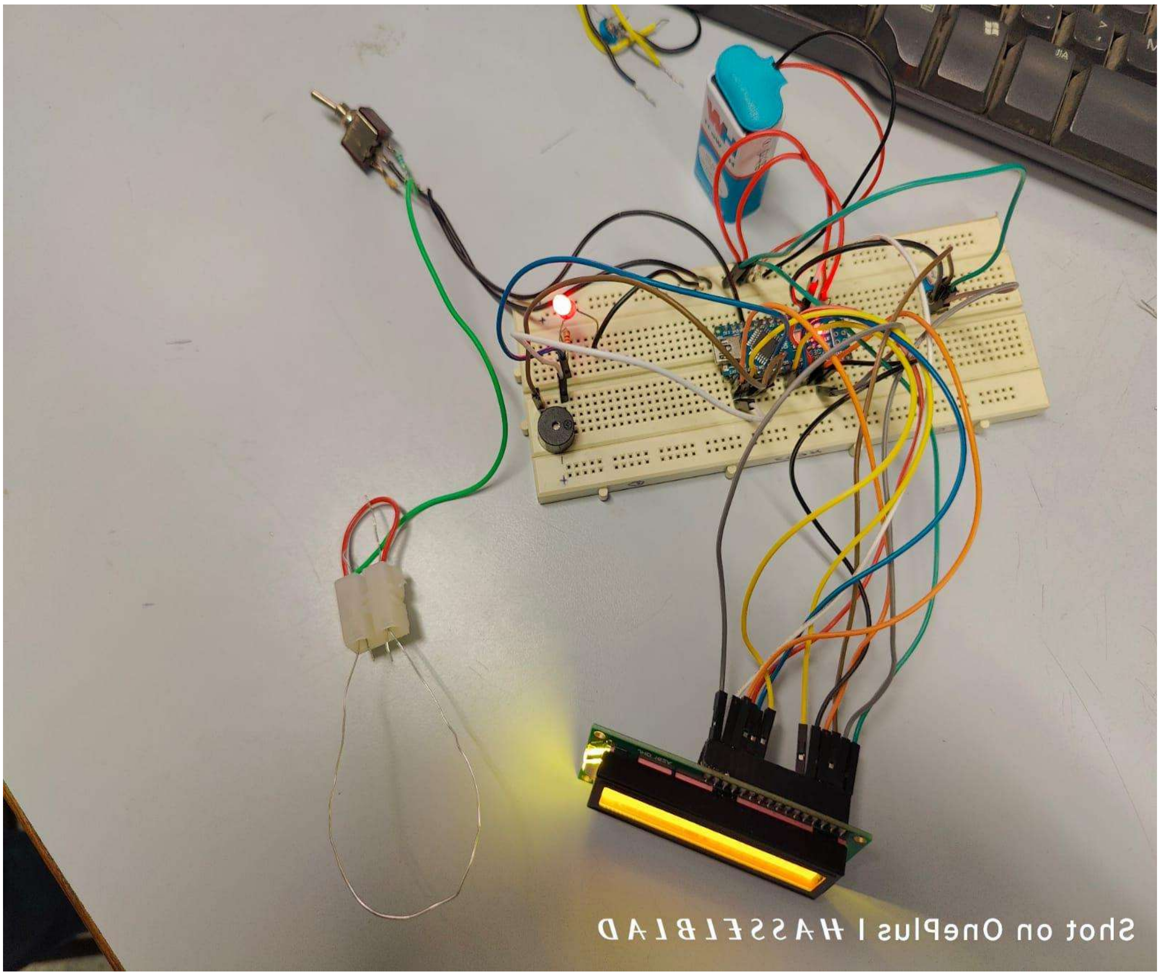
```
 lcd.setCursor(0, 1);       //R channel index
 lcd.write(3);              //R symbol
 lcd.setCursor(15, 1);      //closing tag / end mark index 2
 lcd.write(5);              //closing tag / end mark
 totalR = analogRead(inputPinR) ;
  if(totalR >= 1){
   totalR = constrain(totalR, 0, 100);            // mess with these values
   totalR = map(totalR, 0, 100, 1, 255);     // to change the response distance of the device
   analogWrite(pin10, totalR);          // *note also messing with the resistor should change
   analogWrite(pin9, totalR);                              // the sensitivity
  }
else
{
 // analogWrite(pin10, val); just tuns on the led with                                   //
the intensity of the variable val
 analogWrite(pin10, 0);                  // the else statement is just telling the microcontroller
 analogWrite(pin9, 0);             // to turn off the light if there is no EMF detected
  }
 if(totalR > maxR)
 {
   maxR = totalR;
 }
 indexR++;
 if (indexR >= numReadings)
 {
   indexR = 0;
```

```
  right = maxR;
  maxR = 0;
}
volR = right / 3;
if(volR > 14)
{
  volR = 14;
}
if (volR < (rightAvg - 2))
{
  if (decayTime < actualMillis)
    rightAvg--;
  volR = rightAvg;
}
else if (volR > (rightAvg + 2))
{
  volR = (rightAvg + 2);
  rightAvg = volR;
}
else
{
  rightAvg = volR;
}
if (volR > rightPeak)
{
```

```
    rightPeak = volR;
  }
  drawBar(volR, rightPeak, 1);
  if (decayTime < actualMillis)
    decayTime = (millis() + 50);
  if (peakHold < actualMillis)
  {
    peakHold = (millis() + peakHoldTime);
    rightPeak = -1;
  }
}
void drawBar(int data, int peakData, int row)
{
  //If the previous peak data is 1 or 0, then not taking care of the value.
  if (peakData < 2)
  {
        peakData = -1;
  }
  //First char (idx 0) = R or L
  //Last (16th) char (idx 15) is the closing mark of the bar.
  //We have 14 chars to write.
  for (int col = 1; col < 15; col++)
  {
    lcd.setCursor(col, row);
          if (col < data)
```

```
    {
        lcd.write(1); //write bar element
    }
    else if (peakData == col)
    {
    lcd.write(6); //write the peak marker
    }
    else
    {
    lcd.write(4); //write "empty"
    }
  }
}
```

4.3 IMAGES:



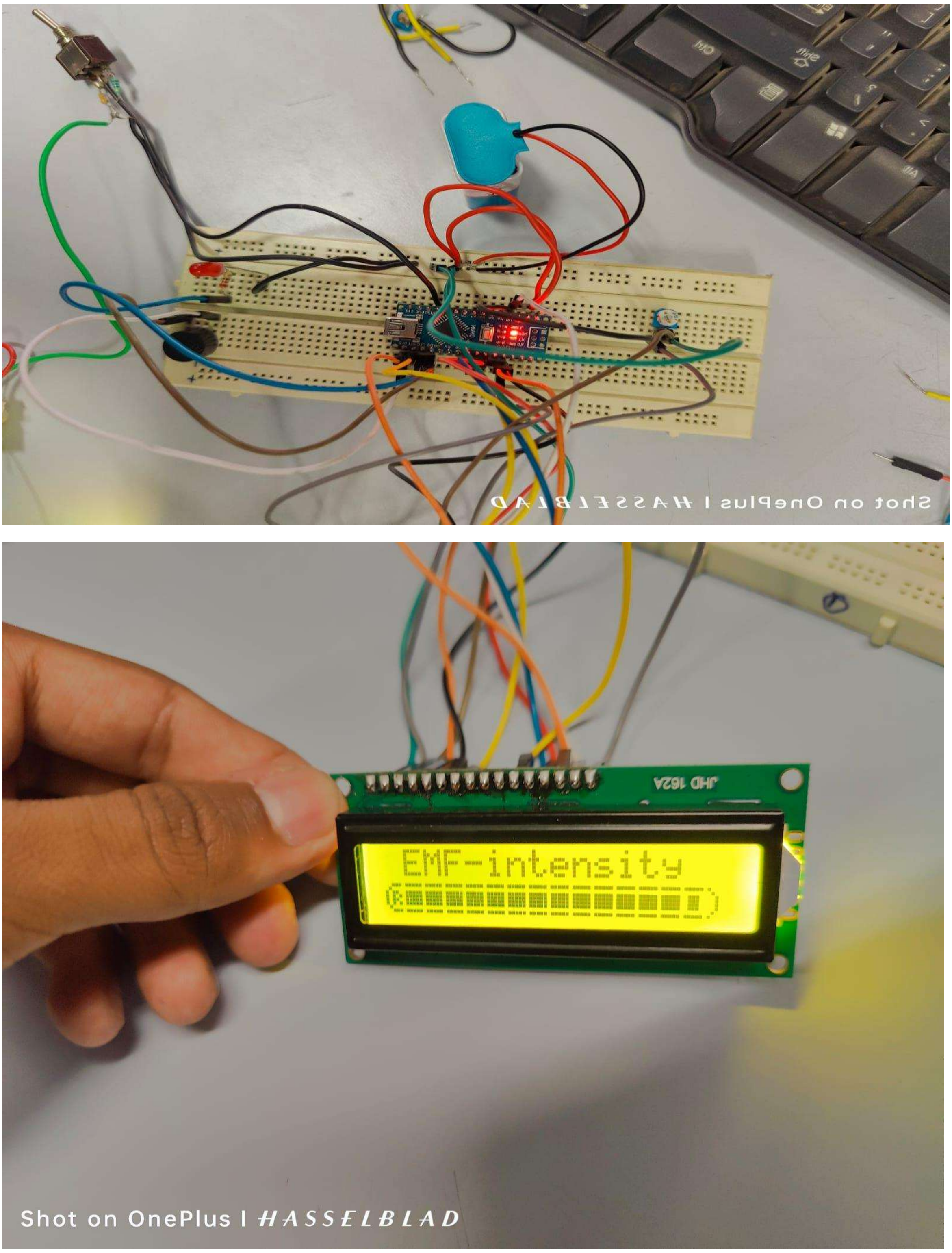


fig.4.2- pictures of the Project

CHAPTER 5
RESULTS

CHAPTER 5

RESULTS

The result of an EMF detector using an Arduino Nano gives the EMF detector will provide a measurement or indication of the strength of the electromagnetic field being detected.

Sound Output: In addition to a visual display, you can incorporate a buzzer or speaker into the EMF detector circuit. The Arduino Nano can be programmed to activate the buzzer or produce different tones based on the strength of the electromagnetic field. For example, the buzzer may emit a continuous tone or beep faster as the field strength increases.

EMF Intensity: using Arduino Nano and LCD display we are able to detect the intensity of the EMF in the surrounding environment and display the range of EMF field i.e., the EMF intensity is displayed on the LCD display. The EMF intensity varies according to EMF field and the corresponding EMF intensity is displayed on the LCD.

CHAPTER 6

CONCLUSION

CHAPTER 6

CONCLUSION

An EMF detector built using an Arduino Nano can be a useful tool for measuring and detecting electromagnetic fields in its vicinity. By combining an Arduino Nano with an EMF sensor module, LCD module, and potentially a buzzer, you can create a device that provides real-time readings of EMF levels and alerts the user when certain thresholds are exceeded.

The Arduino Nano serves as the brain of the EMF detector, processing the input from the EMF sensor module and controlling the output to the LCD module and buzzer. The EMF sensor module detects electromagnetic fields and converts them into electrical signals that can be read by the Arduino. The LCD module allows for the display of EMF readings, providing a visual representation of the detected electromagnetic field intensity. The buzzer, if included, adds an audible alert to notify the user when the EMF levels surpass a predefined threshold.

By combining these components and programming the Arduino Nano accordingly, you can create an EMF detector that provides both visual and auditory feedback, enabling users to assess and monitor electromagnetic fields in their surroundings. This can be useful in various applications, such as evaluating the EMF emissions of electronic devices, assessing the safety of certain environments, or conducting research in the field of electromagnetic radiation.

It's important to note that while an Arduino-based EMF detector can provide indications of EMF levels, it may not offer the same level of accuracy and precision as professional-grade equipment. Additionally, interpreting EMF readings requires understanding the context and considering factors such as the type of electromagnetic field, background levels, and any potential sources of interference. Therefore, it's essential to use an Arduino-based EMF detector as a tool for general awareness and preliminary assessment, rather than relying solely on its readings for conclusive evaluations or safety determinations.

REFERENCES

[1]. From IIT Bombay EMF Detector Project Report.

[2]. “Arduino EMF detector”, Instructables, [Online]. Available: https://www.instructables.com/Arduino-EMFDetector/

[3]. Youtube video: https://youtu.be/1pRq9hw-6rk

[4]. Hackster.io : https://www.hackster.io/mircemk/diy-ultra-sensitive-emf-detector-4be895#schematics