Project Title: Automatic Anti-theft Alarm and Alert System using Arduino

Abstract:

The presented project showcases the fusion of a Microwave motion sensor, a LED, a buzzer, and a 16x2 LCD display within an integrated system. This amalgamation culminates in the creation of an anti-theft solution that serves to detect motion via the Microwave sensor, initiate visual and auditory alerts through the LED and buzzer, and deliver pertinent notifications on the LCD display. The outcome is an innovative security mechanism that efficiently alerts users to motion occurrences and furnishes them with relevant event data.

Introduction:

The project's essence lies in the amalgamation of a Microwave motion sensor, LED indicator, buzzer for auditory feedback, and a 16x2 LCD display. This integration manifest an anti-theft apparatus capable of identifying motion, instigating immediate alerts, and disseminating event specifics through a user-friendly interface.

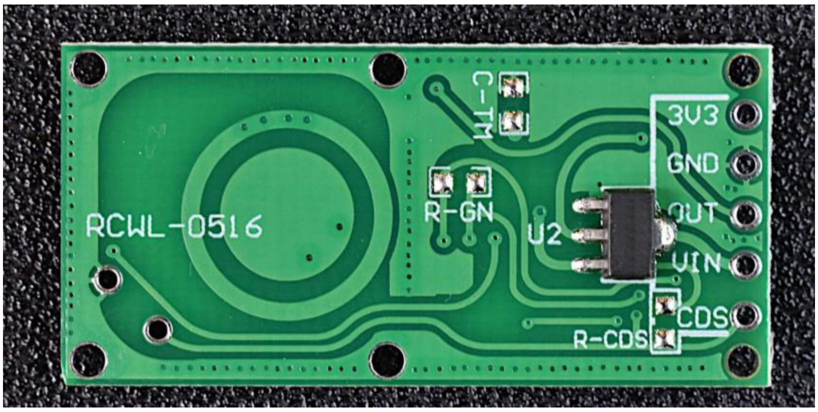
Problem Statement:

This endeavor is fueled by the challenge of conceptualizing and implementing a comprehensive anti-theft system that capitalizes on motion detection, communicates real-time notifications, and conveys crucial event information via the synergistic deployment of a Microwave motion sensor, LED status indicator, buzzer for auditory feedback, and a 16x2 LCD display.

Components Used:

1. **Microwave Sensor**
   A microwave sensor, also known as a microwave motion sensor or radar sensor, is an electronic device used to detect motion and movement within a certain area by emitting

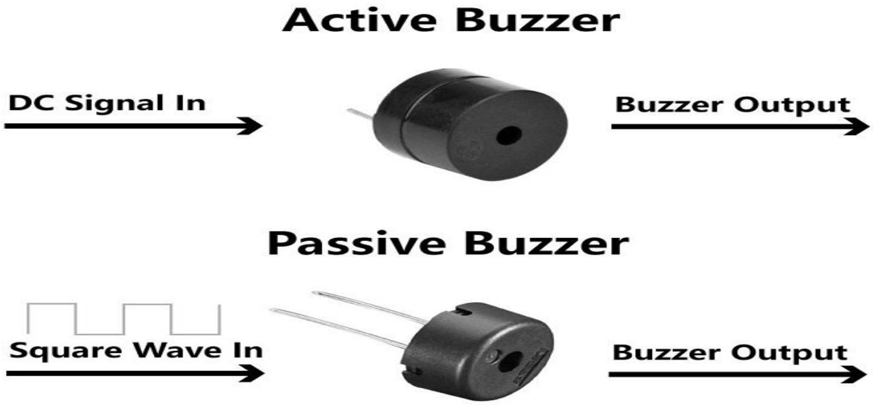
microwave signals and analyzing their reflections. This technology is commonly employed in various applications, including security systems, automatic doors, occupancy detection, and smart lighting control.



2. **Buzzer**

A buzzer is a device used to produce audible sound signals or alerts. It consists of a piezoelectric element or a magnetic coil that converts electrical energy into mechanical vibrations, which, in turn, generate sound waves in the air. Buzzer devices are commonly found in a wide range of applications, including electronics, alarms, notifications, and interactive systems.

Difference between Active and Passive Buzzer:

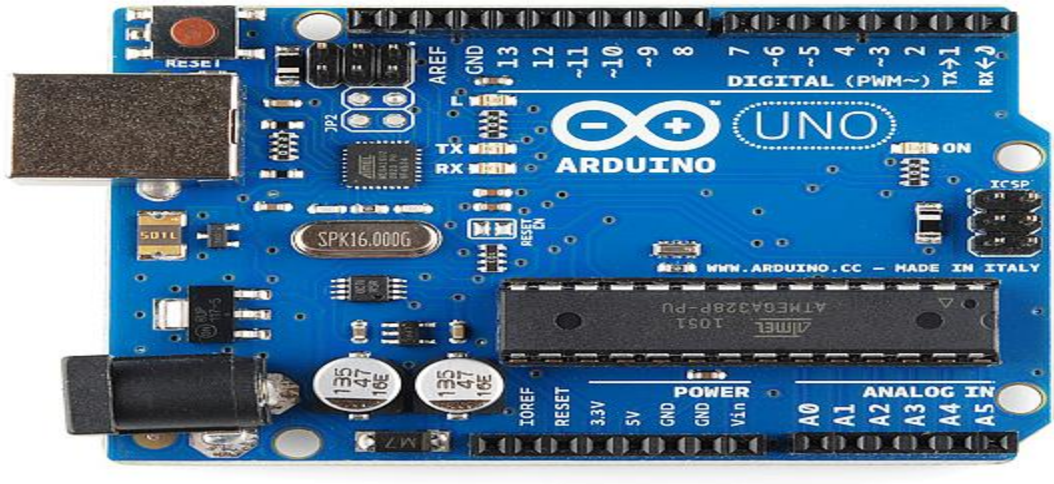


3. **LCD (16 x 2):**

   A LCD (Liquid Crystal Display) is a type of flat panel display technology that uses liquid crystals to produce images or text. The "16 x 2" notation refers to the dimensions of the display, indicating the number of rows and columns of characters it can display. In this case, a 16 x 2 LCD display can show up to 16 characters in each of its two rows. LCD displays are commonly used in a wide range of devices, from calculators and digital watches to more complex applications like information panels and embedded systems.

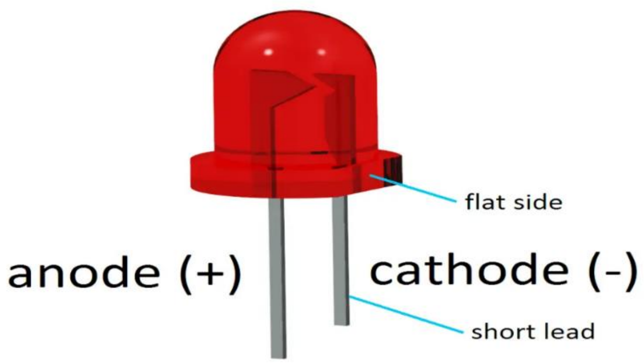
4. **Arduino:**

Arduino Uno is a popular microcontroller board based on the Atmega328P microcontroller. It is a fundamental and widely used member of the Arduino family, known for its simplicity, versatility, and accessibility. Arduino Uno serves as an excellent starting point for beginners and enthusiasts who want to explore electronics, programming, and physical computing.



5. **LED:**

An LED, or Light Emitting Diode, is a semiconductor device that emits light when an electric current passes through it. LEDs are widely used in various applications for their efficiency, durability, and versatility. They come in a range of colors and sizes, making them valuable components in electronics, lighting, displays, indicators, and more.

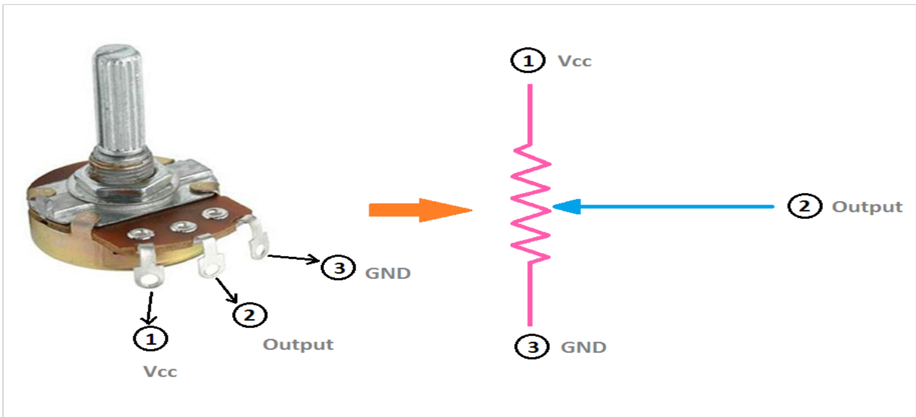


6. **220 Ohm Resistor:**

A 220-ohm resistor is an electrical component used in electronic circuits to limit the flow of electric current. It is commonly referred to as a "220-ohm resistor" due to its resistance value of 220 ohms. Resistors are fundamental components in electronics, serving various purposes such as current limiting, voltage division, and signal conditioning.

7. **10k Ohm Potentiometer:**

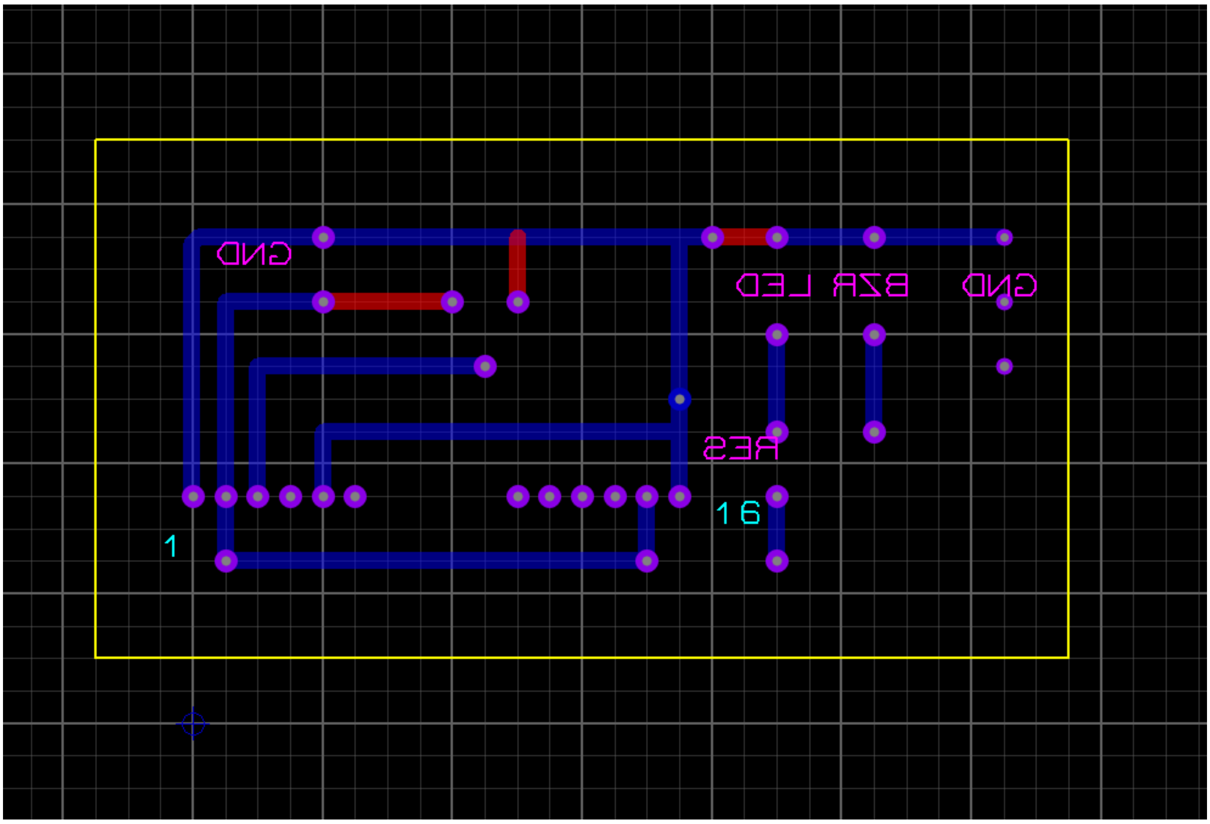
A 10k Ohm Potentiometer, often referred to simply as a "10k pot," is an electronic component that is used to vary resistance in a circuit. It's a type of variable resistor with three terminals that allows you to adjust the resistance value by turning a knob or shaft.



Circuit Diagram and Implementation

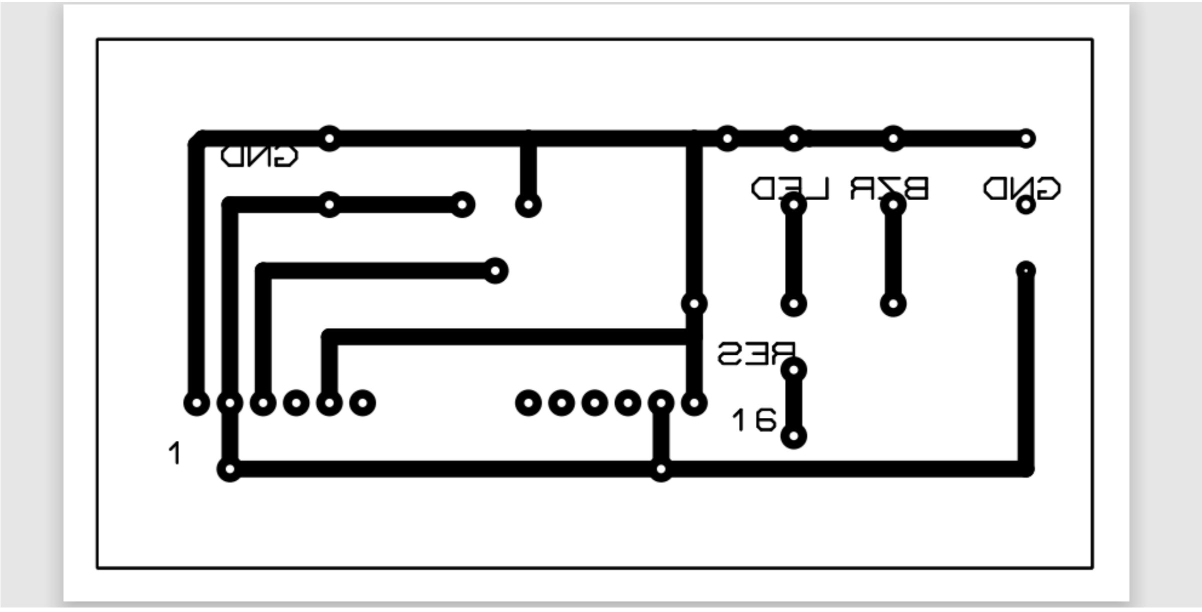
PCB of the Circuit:

To make PCB first of all we will arrange all the components in a the PCB screen and then we can select the size of the wire. We must use t40 or t50 wire so that when we apply FeCl3 Solution on it it will retain. We can use both auto-routing and manual routing as we know that we should have to select only buttom copper so if we apply auto routing then some wires might be in top copper format , so we will prefer manual routing ans the circuit that we get after making PCB is given as.



PCB For Printing:

Once our PCB get ready then we will save it a bitmap file or PDF file with x mirror, x mirror is very necessary because if we mirrored the circuit before patching it will work in a fine manner otherwise it will be mirrored on the copper board.



Steps for making a PCB Board:

- We should keep in mind while making PCB a few steps.
- Stick the paper on copper board.
- Start ironing,
- Use permanent marker for proper lining.
- Add it in $FeCl3$ solution.
- Drilling holes.
- Putting components.
- Soldering.
- Verifying by using voltmeter.

Implementation:

Here's the step-by-step implementation of the anti-theft system using Arduino Uno, Microwave motion sensor, 16x2 LCD display, Buzzer, 220-ohm resistor, breadboard, jumper wires, and a power source (USB cable):

Step 1: Gather Materials

Collect all the required components mentioned above.

Step 2: Setup Components

- Connect the Arduino Uno to the breadboard.
- Connect the microwave motion sensor to the breadboard.
- Connect the 16x2 LCD display to the breadboard.
- Connect the buzzer and the 220-ohm resistor to the breadboard.

Step 3: Wiring Connections

Here's how to connect the components:

- Connect the power and ground pins of the microwave motion sensor to the appropriate rails on the breadboard.
- Connect the output pin of the microwave motion sensor to a digital input pin on the Arduino Uno (e.g., Pin 7).
- Connect the data pins of the 16x2 LCD display to digital pins on the Arduino Uno (e.g., RS to Pin 12, EN to Pin 11, D4 to Pin 5, D5 to Pin 4, D6 to Pin 3, D7 to Pin 2).
- Connect the anode of the buzzer to a digital pin on the Arduino Uno (e.g., Pin 6) through the 220-ohm resistor. Connect the cathode of the buzzer to the ground rail.

Step 4: Write and Upload the Code

Write the Arduino code that reads the motion sensor's output, controls the LCD display, and activates the buzzer. Modify and adapt the code based on the components' specifications and your specific requirements.

Step 5: Arduino Code:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
//the time we give the sensor to calibrate (10-60 secs according to the
datasheet)
int calibrationTime = 30;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int microPin = 7;    //the digital pin connected to the Microwave sensor's
output
int ledPin = 5;    //the digital pin connected to the LED output
int Buzzer = 6;    //the digital pin connected to the BUZZER output
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("WELCOME!");
  Serial.begin(9600);
  pinMode(microPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  digitalWrite(microPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
  lcd.clear();
  lcd.print("calibrating sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      lcd.print(".");
```

```
      delay(1000);
      }
    Serial.println(" done");
    lcd.clear();
    lcd.println(" done");
    lcd.clear();
    lcd.println("SENSOR ACTIVE");
    Serial.println("SENSOR ACTIVE");
    delay(50);
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
 if(digitalRead(microPin) == HIGH){
       digitalWrite(ledPin, HIGH);   //the led visualizes the sensors output
pin state
       tone(Buzzer,500);
       if(lockLow){
         //makes sure we wait for a transition to LOW before any further
output is made:
         lockLow = false;
         Serial.println("---");
          lcd.println("---");
         Serial.print("motion detected at ");
         lcd.clear();
         lcd.print("motion detected at ");
         Serial.print(millis()/1000);
          lcd.print(millis()/1000);
         Serial.println(" sec");
         lcd.println(" sec");
         delay(50);
         }
         takeLowTime = true;
       }

     if(digitalRead(microPin) == LOW){
       digitalWrite(ledPin, LOW);  //the led visualizes the sensors output pin
state
       noTone(Buzzer);
       if(takeLowTime){
        lowIn = millis();          //save the time of the transition from high
to LOW
        takeLowTime = false;       //make sure this is only done at the start
of a LOW phase
        }
       //if the sensor is low for more than the given pause,
```

```
        //we assume that no more motion is going to happen
        if(!lockLow && millis() - lowIn > pause){
            //makes sure this block of code is only executed again after
            //a new motion sequence has been detected
            lockLow = true;
            Serial.print("motion ended at ");
            lcd.clear();

            lcd.print("motion ended at");
            Serial.print((millis() - pause)/1000);
            lcd.print((millis() - pause)/1000);
            Serial.println(" sec");
            lcd.println(" sec");
            delay(50);
            }
        }
}
```

Step 6: Test the System

Upload the code to the Arduino Uno using the Arduino IDE. Make sure all connections are secure. Power up the system using the USB cable.

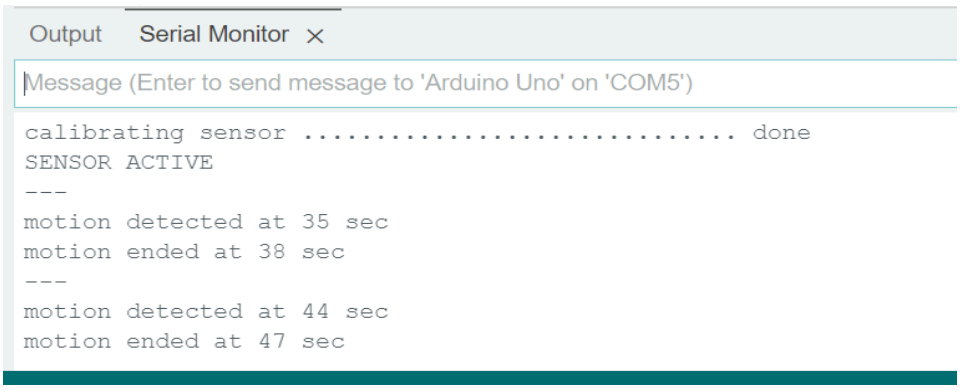
Step 7: Calibration

Adjust the sensitivity of the microwave motion sensor as needed by referring to its datasheet or user manual.

Step 8: Deploy and Monitor

Place the microwave motion sensor in the area you want to monitor for motion. The LCD will display "Motion Detected!" and the buzzer will sound when motion is detected.

Arduino Output:



Project Working Video Links:

https://www.kapwing.com/videos/64e0a787bf83f70063bdf5e9

https://www.kapwing.com/videos/64e0a6e52682aa0025826746

Applications:

An anti-theft system using Arduino, LCD, and a buzzer has a range of applications where security, monitoring, and alerting are essential.

Here are some potential applications for such a system:

Home Security System: The system can be used to secure homes by detecting unauthorized entry or motion in sensitive areas. When motion is detected, the LCD can display a warning message, and the buzzer can sound an alarm, alerting homeowners and deterring intruders.

Vehicle Security: The anti-theft system can protect vehicles from theft or unauthorized access. If someone tries to break into a vehicle, the system can trigger the buzzer and display a message on the LCD to alert the owner.

Office Security: The system can be employed to safeguard office spaces during non-working hours. Any movement in restricted areas can trigger the alarm and notify security personnel.

Retail Stores: Retailers can use the system to prevent shoplifting. When a person moves near a valuable display, the buzzer can sound, and the LCD can display a warning, deterring potential thieves.

Personal Belongings: Attach the system to personal belongings such as bags or luggage to prevent theft. If someone tries to move or tamper with the belongings, the system can raise an alert.

Future Scope:

The anti-theft system using Arduino, LCD, and a buzzer has a promising future scope due to the continuous advancements in technology and the increasing demand for security solutions.

Here are some potential future developments and applications for this system:

The GSM module:

The GSM module can send text messages or make calls to alert you when the system detects unauthorized activity. This ensures that you receive timely notifications even if you're not physically present at the location.

Integration with Smart Home Systems: The system could be integrated with smart home platforms, allowing users to monitor and control security alerts remotely through their smartphones or other connected devices.

Wireless Connectivity: Integrate wireless communication modules (such as Wi-Fi or Bluetooth) to enable remote monitoring, control, and notifications via smartphone apps or cloud platforms.

Camera Integration: Incorporate a camera module to capture images or videos when motion is detected, providing visual evidence of security events.

Emergency Response Integration: Integrate with emergency response services, such as police or private security companies, for rapid intervention in case of a security breach.

Conclusions:

The seamless integration of a Microwave motion sensor, LED, buzzer, and 16x2 LCD display materializes a formidable anti-theft mechanism. This model is cheaper than any other security system available in the market. This apparatus adeptly detects motion, elicits prompt alerts, and renders comprehensive event details accessible through an intuitive interface, signifying its potential to bolster security measures effectively.

References:

https://www.instructables.com/Automatic-Anti-Theft-Alarm-and-Alert-System-Using-/

https://www.slideshare.net/AdityaNag6/home-security-system-using-arduino-gsm