MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Department of ICT

MIT, Manipal

Alcohol Detection System

ICT 2223 Embedded Systems lab mini project

IVth Sem B.Tech (IT)

230911264	**Adarsh Agrawal**	**(37)**
230911272	**Ayush Mishra**	**(38)**
230911258	**Sankalp Amogh**	**(36)**

Contents

Abstract …………………………………………………………………………… 2
List of Figures ………………………………………………………………….. 3
List of Tables …………………………………………………………………….4
Introduction ……………………………………………………………………….. 5
Methodology ……………………………………………………………………… 6
Results and Discussions …………………………………………………………… 8
References ……………………………………………………………………… 10
C Code ……………………………………………………………………………. 11

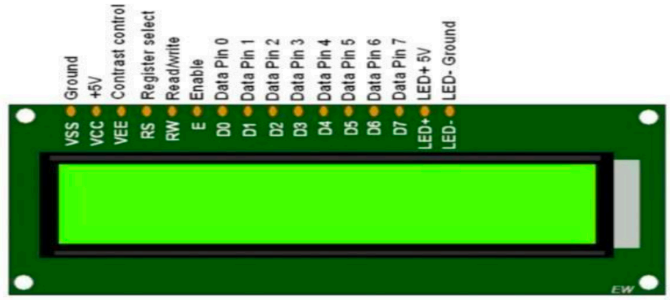
Abstract

The pervasive issue of alcohol abuse poses significant threats to individual and public safety, underscoring the need for effective detection and prevention measures. This project focuses on developing an Alcohol Detection System to address this pressing concern, aiming to provide a reliable and accessible means of identifying the presence of alcohol. This system has various applications, such as preventing drunk driving and monitoring individuals in rehabilitation settings.

The system utilises the NXP LPC 1768 microcontroller, a 32-bit ARM Cortex-M3 microcontroller known for its performance and power efficiency. The MQ3 alcohol sensor, which operates by changing its resistance in response to alcohol levels, detects alcohol vapour in the air and sends an analog signal to the microcontroller. The microcontroller then uses its built-in ADC (Analog to Digital Converter) to convert this signal into a digital value. The processed data triggers an alert if the detected alcohol level exceeds a predefined threshold. The system provides real-time feedback through visual and auditory indicators: a buzzer buzzes, signalling the presence of alcohol, and a 16x2 LCD display shows the percentage of alcohol present. The buzzer sounds when the alcohol % is high, providing an immediate and attention-grabbing alert. If no alcohol is detected, the buzzer remains off and the LCD displays low alcohol, indicating normal operation.

The Alcohol Detection System effectively detects alcohol vapour and provides clear visual and auditory alerts through an LCD display, and a buzzer. The project successfully demonstrates the feasibility of using a microcontroller-based system for alcohol detection, offering a practical solution for various applications. This system can contribute to enhancing road safety by preventing drunk driving, aiding in the rehabilitation of individuals struggling with alcohol addiction, and providing a reliable tool for law enforcement in breathalyser devices. By providing a means for early detection and intervention, this project aims to mitigate the risks associated with alcohol abuse and promote a safer environment for individuals and communities.

List of Figures



LCD *F*

DISPLAY

Fig 1. LCD Screen

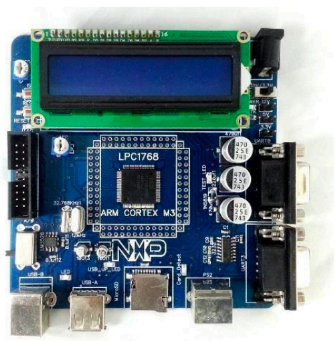


Fig2. LPC1768 microcontroller

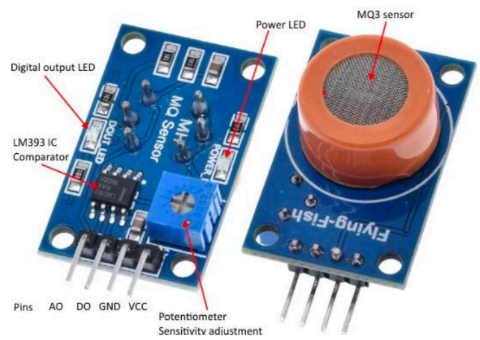


Fig 3. MQ3 Alcohol Sensor

List of Tables

Functions	Return Type	Parameters	Description
lcd_init	void	void	Initializes the LCD
lcd_write	void	void	Communicates with LCD by sending data or commands in 4-bit.
port_write	void	void	Intended to interface with an LCD using GPIO on LPC.
lcd_print_msg	void	void	Print a message stored in an array 'msg'
lcd_print_msg2	void	void	Print a message stored in an array 'msg2'
main	int	void	To detect alcohol and obtain the alcohol % using ADC and write on LCD and other initializations.
delay	void	Unsigned int	Creates a simple, blocking delay to perform empty iterations
buzzer_pattern_high	void	void	Activate a buzzer in a rapid on-off pattern, triggering an alert.

Introduction

Scope : The project focuses on developing an alcohol detection system using the NXP LPC 1768 microcontroller and the MQ3 alcohol sensor to detect alcohol vapour in the surrounding air. The system will provide real-time feedback through an LED, a 16x2 LCD display, and a buzzer. The main goal is to create a reliable and user-friendly system capable of identifying the presence of alcohol and triggering alerts when certain thresholds are met.

Project Description : The "Alcohol Detection System" uses the MQ3 alcohol sensor to detect alcohol vapours and transmit an analog signal to the NXP LPC 1768 microcontroller. This microcontroller, a 32-bit ARM Cortex-M3 known for its performance and power efficiency, converts the analog signal to a digital value using its ADC. The system is designed to activate alerts when the detected alcohol level exceeds a predefined threshold. These alerts include:

- An LED that lights up
- A message on the LCD display indicating "Alcohol Detected"
- A buzzer that produces an auditory signal.

If alcohol is not detected, the LED will remain off, and the LCD display will show "No Alcohol Detected".

Problem Statement : The project aims to address this problem by developing an effective **"Alcohol Detection System"** that can accurately identify the presence of alcohol vapour and provide clear alerts.

Objective : The primary objective of the "Alcohol Detection System" project is to create a reliable, user-friendly, and cost-effective system capable of detecting alcohol vapour and providing real-time feedback through visual and auditory alerts. This system aims to:

- Enhance road safety by deterring drunk driving.
- Support the rehabilitation process of individuals with alcohol addiction.
- Provide a reliable tool for breathalyser devices used by law enforcement.

The project intends to contribute to mitigating the risks associated with alcohol abuse and promoting a safer environment for individuals and communities.

Methodology

a) Components Required

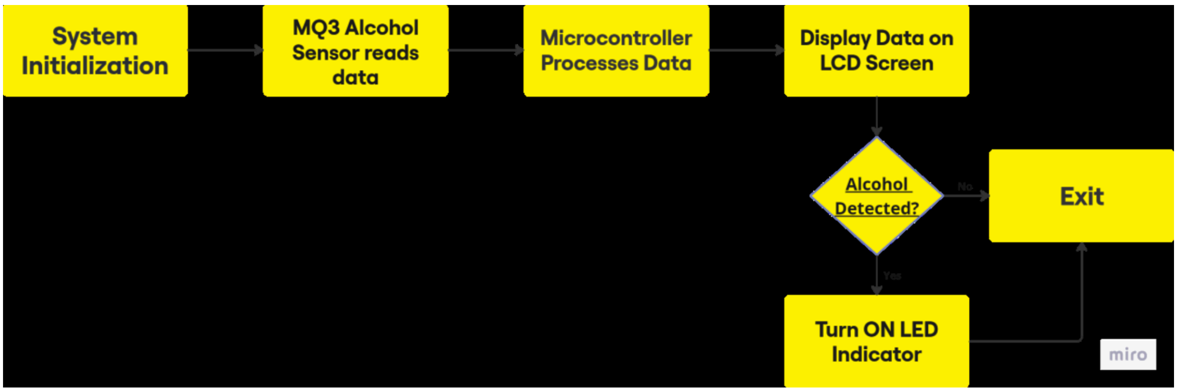
Hardware

- NXP LPC 1768 Microcontroller: A 32-bit ARM Cortex-M3 microcontroller. It provides excellent performance and power efficiency.
- MQ3 Alcohol Sensor: Used for detecting alcohol vapours in the air. It operates by changing its resistance in response to alcohol levels, which can be measured using the ADC of a microcontroller.
- Power Supply: To ensure continuous operation of the system.
- Buzzer: Used to indicate the system status, for example, ringing to signal the detection of alcohol.
- LCD: Used to display real-time messages and alerts.
- USB to B type cable: Enables connectivity between devices and facilitates data transfer, particularly useful for linking the ARM Cortex-M3 board with other peripherals or the host computer system.
- 10 core FRC cables of 8-inch length: Used for internal connections and wiring within the kit.

Software

- Language: Embedded C: Used to program the microcontroller.
- IDE: Keil Micro Vision: Used to develop and debug the embedded C code.
- Application: Flash Magic: Used to program the microcontroller.

b) Block Diagram



c) Description about the connection

LCD Connection: The LCD is connected to the microcontroller through specific pins defined in the code:

RS_CTRL (P0.27): Controls the register select of the LCD, determining whether data or command is being sent.
EN_CTRL (P0.28): The enable pin, used to latch data or commands into the LCD.
DT_CTRL (P0.23 to P0.26): These pins carry the data bits to the LCD.

Buzzer Connection: The buzzer is connected to P2.13 of the microcontroller. The code configures this pin as an output pin to control the buzzer's activation.

MQ3 Sensor Connection: The MQ3 alcohol sensor is connected to P0.21 of the microcontroller. This pin is configured as an input pin to read the digital output from the sensor. The MQ3 sensor outputs a logic LOW signal when alcohol is detected.

d) Method

The method of implementation of an Alcohol Detection System requires hardware setup, software development, and system integration.

Hardware Setup : The NXP LPC 1768 microcontroller, featuring a 32-bit ARM Cortex-M3 core, is used to control the system's functions. An MQ3 alcohol sensor detects alcohol presence, changing resistance to send an analog signal to the microcontroller. Alerts are provided via an LED, LCD, and buzzer, each activated when alcohol is detected. The system requires a stable power supply, with correct component connections: LCD pins P0.27, P0.28, and P0.23 to P0.26; buzzer to pin P2.13; and MQ3 sensor to pin P0.21.

Software Development : Embedded C code manages sensor data reading, threshold comparison, and output control. Functions like `initMQ3Sensor` and `readMQ3Sensor` read alcohol data and compare it against a defined threshold. If alcohol presence exceeds this threshold, the system activates the LED, LCD (displaying "Alcohol Detected"), and buzzer. The LCD is managed with functions like `lcd_init`, `wr_cn`, and `lcd_puts`, which facilitate command writing and display messages.

System Integration : Following a circuit diagram, components are physically connected and embedded C code is loaded onto the microcontroller. Testing and calibration fine-tuning the MQ3 sensor’s sensitivity and threshold values to ensure accurate detection. This project combines hardware and embedded programming, creating a responsive alcohol detection system.

Results and Discussion

The Alcohol Detection System integrates hardware and software to detect and alert alcohol presence effectively. Using the NXP LPC 1768 Microcontroller and an MQ3 alcohol sensor, the system accurately senses alcohol vapour, with the sensor's resistance changing upon alcohol detection. Testing confirmed the system's reliability, with the LCD showing "Alcohol Detected" or "No Alcohol Detected," the LED lighting up, and the buzzer sounding as alerts.

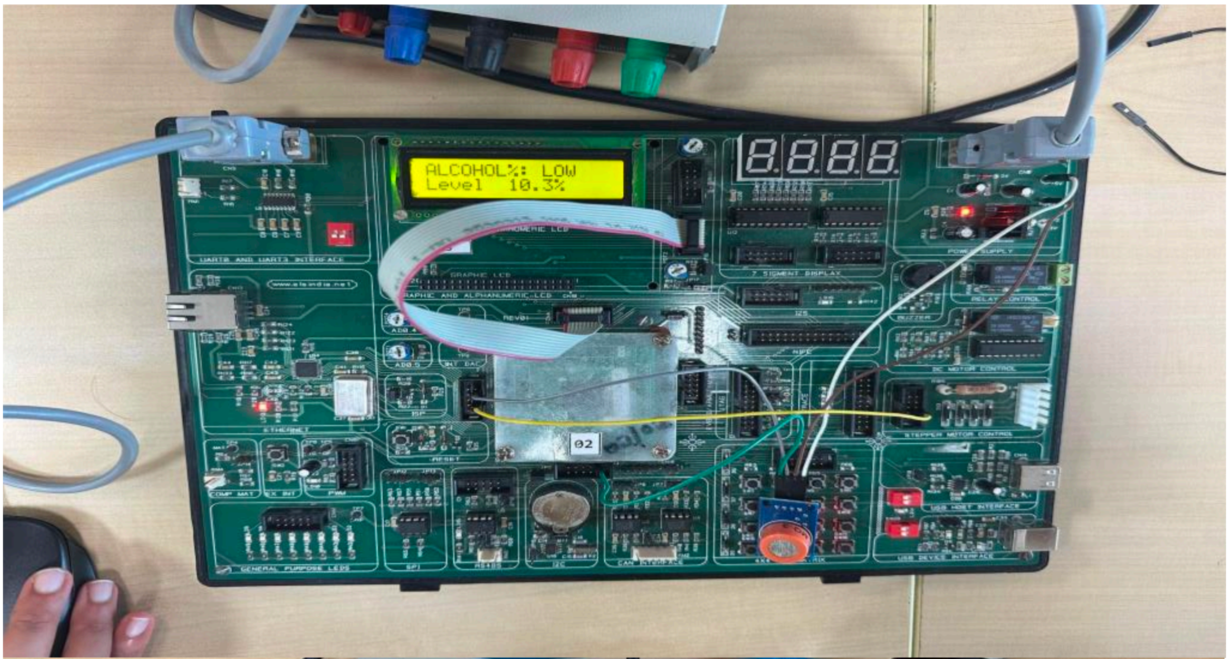
Key findings :

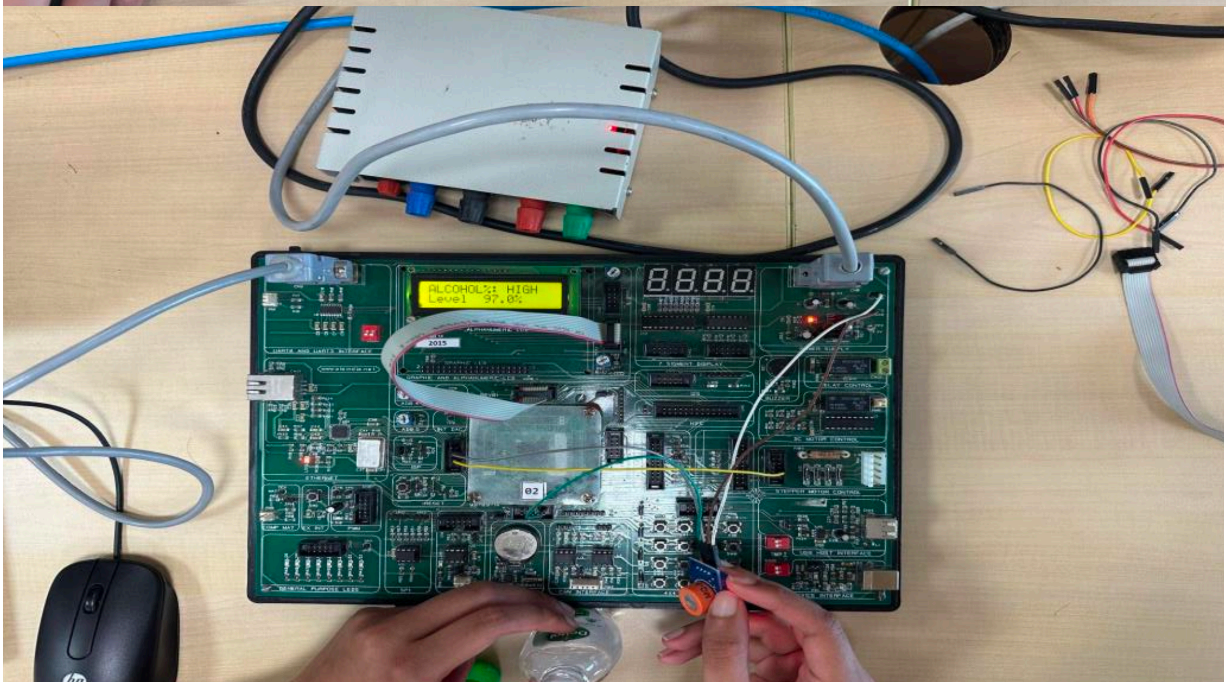
- Sensitivity: The system detects alcohol reliably, even at lower concentrations; further calibration could enhance accuracy for specific applications.
- Response Time: The system responds swiftly, making it suitable for applications like breathalysers.
- Reliability: Consistent detection and alert activation confirmed system reliability.

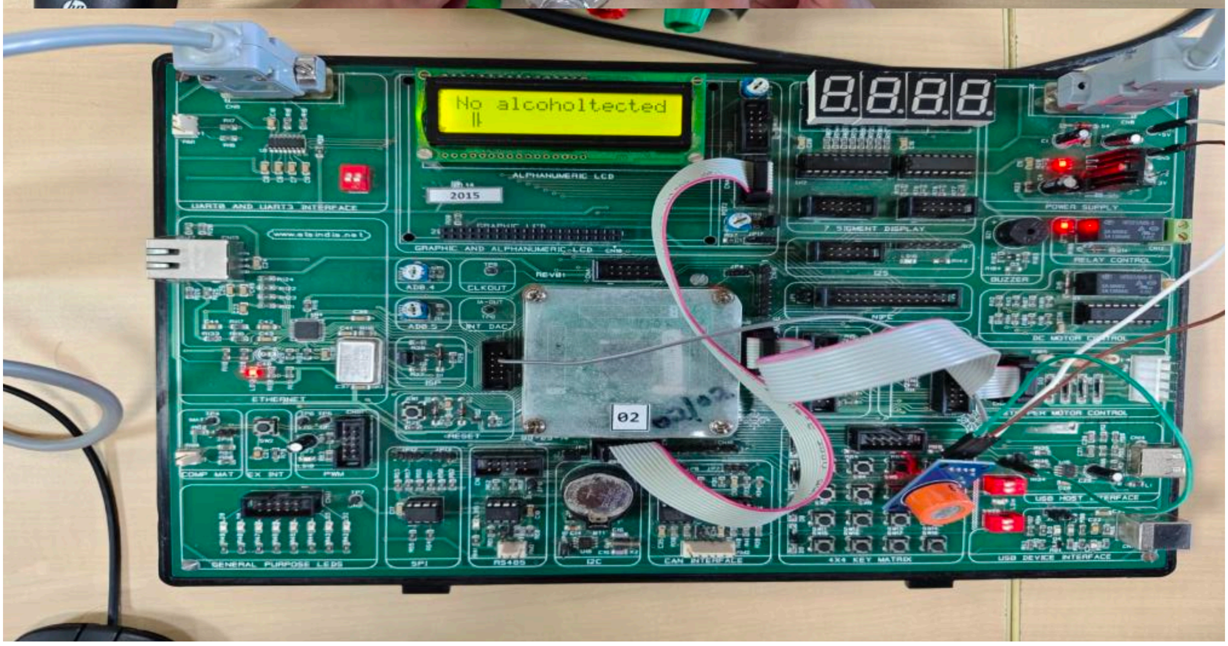
Future Improvements :

- Quantitative Measurement: Adding analog-to-digital conversion could provide specific concentration readings.
- Wireless Connectivity: Adding Bluetooth or Wi-Fi would enable remote monitoring.

This project demonstrates effective sensor-microcontroller integration, with applications in safety and healthcare. Further refinement could expand its utility across various fields.


ALCOHOL%: LOW
Level 10.3%


ALCOHOL%: HIGH
Level 97.0%


No alcoholtected

References

https://nevonprojects.com/alcohol-sensing-display-with-alarm-project/
https://ajast.net/data/uploads/67541.pdf
https://smartxprokits.in/design-of-alcohol-detection-system-for-car-diy-experiments-school-science-stem-kit-assembled-100-tested/
https://www.pantechsolutions.net/alcohol-detector-using-arduino
https://community.nxp.com/t5/8-bit-Microcontrollers/ESD/m-p/125594
https://www.tcetmumbai.in/Multicon/2019/ICTELCON%202019.pdf
https://www.researchgate.net/publication/350409199_Alcohol_Detection_Sensor-An_Apprise
https://www.pnrjournal.com/index.php/home/article/view/6911

ESD_Project_Final.pdf

ORIGINALITY REPORT

1%	1%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	1%
2	Steven F. Barrett. "Arduino II", Springer Science and Business Media LLC, 2020 Publication	1%

Exclude quotes On
Exclude bibliography On
Exclude matches < 3 words

C Code with comments

```
#include <lpc17xx.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define RS_CTRL 0x00000100 //P0.8
#define EN_CTRL 0x00000200 //PO.9
#define DT_CTRL 0x000000F0 //PO.4 TO P0.7
#define BUZZER_PIN (1<<17) // P2.13 for buzzer

unsigned long int init_command[] = {0x30,0x30,0x30,0x20,0x28,0x0c,0x06,0x01,0x80};
unsigned long int temp1 = 0, temp2 = 0, i, j, var1, var2;
unsigned char flag1 = 0, flag2 = 0;
unsigned char msg[] = {"ALCOHOL%:"};
unsigned char msg2[] = {"Level: "};
unsigned char buzzer_state = 0;

void delay(unsigned int r1) {
   unsigned int r;
   for(r = 0; r < r1; r++);
}

// Buzzer patterns for different alcohol levels
void buzzer_pattern_high(void) {
   int z;
   for(z = 0; z < 20; z++) { // Rapid beeps for high alcohol
      LPC_GPIO0->FIOSET = BUZZER_PIN;
      delay(1000000);
      LPC_GPIO0->FIOCLR = BUZZER_PIN;
      delay(1000000);
   }
}

/*void buzzer_pattern_medium(void) {
   int x;
   for(x = 0; x < 10; x++) { // Medium paced beeps
      LPC_GPIO0->FIOSET = BUZZER_PIN;
      delay(2000000);
      LPC_GPIO0->FIOCLR = BUZZER_PIN;
      delay(2000000);
   }
}*/

/*void buzzer_pattern_low(void) {
```

```
    int y;
    for(y = 0; y < 5; y++) { // Slow beeps for low alcohol
        LPC_GPIO2->FIOSET = BUZZER_PIN;
        delay(3000000);
        LPC_GPIO2->FIOCLR = BUZZER_PIN;
        delay(3000000);
    }
}*/

void lcd_init(void);
void lcd_write(void);
void port_write(void);
void delay(unsigned int);
void lcd_print_msg(void);
void lcd_print_msg2(void);

int main(void) {
    unsigned int i;
    unsigned int sensorReading;
    char disp_level[10];
    char percentageStr[14];
    float alcohol_percentage;

    SystemInit();
    SystemCoreClockUpdate();
    LPC_PINCON->PINSEL1 |= 1<<14; //function 1 on P0.23 for AD0
    LPC_SC->PCONP |= (1<<12); //peripheral power supply for ADC
    LPC_GPIO0->FIODIR = DT_CTRL | RS_CTRL | EN_CTRL;
    LPC_GPIO0->FIODIR |= BUZZER_PIN;

    lcd_init();
    lcd_print_msg();
    lcd_print_msg2();

    while(1) {
        // ADC Reading
        LPC_ADC->ADCR = (1<<0) | (1<<21) | (1<<24);
        while(((sensorReading = LPC_ADC->ADGDR) & 0X80000000) == 0);
        sensorReading = LPC_ADC->ADGDR;
        sensorReading >>= 4;
        sensorReading &= 0x00000FFF;

        // Convert ADC reading to alcohol percentage (0-100%)
        // Assuming sensor gives linear output and max ADC value (4095) = 100%
        alcohol_percentage = (float)sensorReading * 100.0 / 4095.0;
        sprintf(percentageStr, " %.1f%% ", alcohol_percentage);

        // Alcohol Level Classification and Buzzer Control
```

```
        if(alcohol_percentage >= 50.0) {
            strcpy(disp_level, " HIGH ");
            buzzer_pattern_high();
        }
        else if(alcohol_percentage >= 20.0) {
            strcpy(disp_level, " MEDIUM ");
            //buzzer_pattern_medium();
        }
        else {
            strcpy(disp_level, " LOW ");
            //buzzer_pattern_low();
        }

        // LCD Display Update
        temp1 = 0x89;
        flag1 = 0;
        lcd_write();
        delay(800);
        i = 0;
        flag1 = 1;
        while(disp_level[i] != '\0') {
            temp1 = disp_level[i];
            lcd_write();
            i += 1;
        }

        temp1 = 0xC5;
        flag1 = 0;
        lcd_write();
        delay(800);
        i = 0;
        flag1 = 1;
        while(percentageStr[i] != '\0') {
            temp1 = percentageStr[i];
            lcd_write();
            i += 1;
        }
        delay(1000000); // Delay between readings
    }
}

void lcd_init(void) {
    unsigned int x;
    flag1 = 0;
    for(x = 0; x < 9; x++) {
        temp1 = init_command[x];
        lcd_write();
    }
```

```
    flag1 = 1;
}

void lcd_write(void) {
    flag2 = (flag1 == 1) ? 0 : ((temp1 == 0x30) || (temp1 == 0x20)) ? 1 : 0;
    temp2 = temp1 & 0xf0;
    port_write();
    if (flag2 == 0) {
        temp2 = temp1 & 0x0f;
        temp2 = temp2 << 4;
        port_write();
    }
}

void port_write(void) {
    LPC_GPIO0->FIOPIN = temp2;
    if (flag1 == 0)
        LPC_GPIO0->FIOCLR = RS_CTRL;
    else
        LPC_GPIO0->FIOSET = RS_CTRL;
    LPC_GPIO0->FIOSET = EN_CTRL;
    delay(25);
    LPC_GPIO0->FIOCLR = EN_CTRL;
    delay(3000000);
}

void lcd_print_msg(void) {
    unsigned int a;
    for(a = 0; msg[a] != '\0'; a++) {
        temp1 = msg[a];
        lcd_write();
    }
}

void lcd_print_msg2(void) {
    temp1 = 0xC0;
    flag1 = 0;
    lcd_write();
    delay(800);
    i = 0;
    flag1 = 1;
    while(msg2[i] != '\0') {
        temp1 = msg2[i];
        lcd_write();
        i++;
    }
}
```