E-BIN

a smart Garbage system

Term Project for the course of 19ECE304 Microcontrollers and Interfacing

Credits:

C H Keerthi Vardhan - CB.EN.U4ECE21206
D Harsha Vardhan - CB.EN.U4ECE21207
L Likhitha Ram - CB.EN.U4ECE21228
N Maneeswar Reddy - CB.EN.U4ECE21230

Faculty incharge: C B Rajesh

Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India.

Abstract—**This paper presents the design and implementation of a smart and ecofriendly waste monitoring and control system design called the "E-Bin". As urbanization accelerates and populations grow, efficient waste management became a critical challenge. The traditional waste disposal methods are proving inadequate in handling the increasing volumes of waste, leading to environmental pollution and health hazards. This paper introduces the concept of the "E-Bin," a smart dustbin designed using the microcontroller applications, along with the convergence of Keil µVision4 for coding and Proteus simulator for simulation of the design, through the integration of advanced technologies.**

Keywords: **E-Bin; microcontroller; garbage; environment; ecofriendly .**

CAD Tools used : **Keil µVision4, Proteus.**

I. INTRODUCTION

At the present time, our environment is being polluted by massive deposits of global waste. This will have a catastrophic effect on human life and the surroundings. It was anticipated that the global waste will grow by 70% by 2050 unless immediate necessary measures of monitoring and control are put in place Environmental sanitation is very essential for healthy living. In our daily livelihood, garbage bins are usually kept without proper monitoring until they are filled to the point of overflowing onto the surroundings and spilling out, resulting in environmental pollution, which has serious health-related issues to human beings and the environment To address this challenge, our project introduces the E-Bin—a pioneering smart waste management system leveraging the power of the microprocessors and interfacing This project aims to redefine the landscape of waste management, promoting environmental consciousness and paving the way for smarter, more resilient cities.

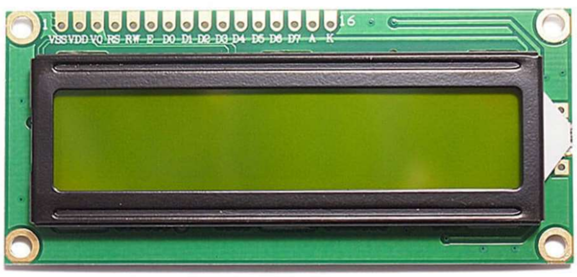
II. DESCRIPTION

The combination of Keil µVision for coding and Proteus for simulation in the field of embedded systems and microcontroller applications had made it possible for the development of the project's resilience and effectiveness. We have utilized various elements of the project, each of which is briefly described below.

LPC2138:

The LPC2138 is a microcontroller belonging to the ARM7TDMI-S family, manufactured by NXP Semiconductors that has a 32-bit RISC processor architecture. It operates at a clock frequency of up to 60 MHz having multiple serial communication interfaces, including UART, SPI, and I2C.USB 2.0 Full-Speed interface for connectivity. With on-chip peripherals such as USB, timers, and ADC, it is commonly used in industrial automation and embedded systems.

LM016L LCD:



:

The LM016L is a popular alphanumeric LCD (Liquid Crystal Display) module widely used in embedded systems for displaying information. It is a 16x2 character LCD module, meaning it can display 16 characters in each of its two rows. It typically supports a standard set of ASCII characters, numbers (0-9), and some special characters. A Liquid Crystal Display (LCD) is a flat-panel display technology that uses a liquid crystal solution sandwiched between two layers of glass or plastic. The LCD technology has become widespread and is commonly used in various electronic devices such as televisions, computer monitors, digital watches, and embedded systems. The LM016L typically has 16 pins, including power supply (VCC and GND), data lines (D0-D7), control lines (RS, RW, and EN), and optional backlight connections.

UART:

UART is a common communication interface used for serial communication between devices. It allows for the transmission of data between the microprocessor and other peripherals, such as sensors, displays, or communication modules.

SPST PUSH BUTTON:

A Single Pole Single Throw (SPST) push button is a simple type of switch with two terminals. When the button is pressed, the circuit is closed, allowing current to flow; when released, the circuit opens, interrupting the current. SPST buttons are commonly used for basic on/off applications and have a straightforward design, making them suitable for various electronic and electrical devices.

BUZZER:

A buzzer is an electronic signaling device that produces a buzzing or beeping sound when an electric current is applied. It typically consists of a piezoelectric element or an electromagnetic coil attached to a diaphragm. When energized, the piezoelectric element vibrates, or the electromagnetic coil induces movement in the diaphragm, creating sound waves

LED :

A Light-Emitting Diode (LED) is a semiconductor device that emits light when an electric current passes through it. Composed of a semiconductor chip housed in a plastic shell, the color of the emitted light depends on the semiconductor material used**.** LEDs operate on low voltage, making them suitable for battery-powered devices, and they have become integral in energy-efficient lighting solutions and electronic displays.

III. PROPOSED METHODOLOGY

The E-Bin project envisions the development of a sophisticated smart dustbin system designed to revolutionize waste management through the integration of cutting-edge technologies. Leveraging the LPC2138 microcontroller as the brain of the system, this innovative solution incorporates LEDs, a buzzer, and an alphanumeric LCD (LM016L) to provide real-time monitoring and insightful feedback on the status of individual dustbins. In this project the microcontroller is configured to the SPST push buttons that acts as the levels of space left out in each corresponding dustbin. If the brim of the dustbin is accumulated with waste, then the buzzer is activated and leaving the LED to glow RED. Accordingly, the space left out can be monitored using the LEDs–glows GREEN if empty and if almost full the LED glows ORANGE. By this the level of the dustbins are monitored and pollution can be controlled.

IV. ALGORITHM

1**. Initialize Hardware:**
- Configure the LPC2138 microcontroller's pins and peripherals using the necessary register settings.
- Set up I/O pins for LEDs, buzzer, and LCD.

2. Initialize LCD
- Call the LCD_init() function to initialize the LCD module for communication.

3. Main Loop
- Enter an infinite loop (while (1)) for continuous monitoring of the dustbin status.

4. Clear LCD and Set Cursor
- Clear the LCD screen using LCD_command(0x01).
- Set the cursor to the beginning of the first line using LCD_command(0x80).

5. Display "BIN1:" on LCD
- Write the string "BIN1:" to the LCD using LCD_write_string().

6. Check Dustbin Level (Switch P0.02)
- If the dustbin level switch (P0.02) is ON
  - Turn on the Red LED (P1.20) and the Buzzer (P1.29).
  - Display "FULL" on the LCD.
- Else:
  - Turn off the Red LED and the Buzzer.

7. Check Near Full Level (Switch P0.03)
- If the near-full level switch (P0.03) is ON:
  - Turn on the LED at P1.21.
  - Display "NEARLY FULL" on the LCD.
- Else:
  - Turn off the LED at P1.21.

8. Check Empty Level (Switch P0.04)
- If the empty level switch (P0.04) is ON:
  - Turn on the LED at P1.22.
  - Display "EMPTY" on the LCD.
- Else:
  - Turn off the LED at P1.22.

9. Repeat Steps 5-8 for BIN2

- Move the cursor to the beginning of the second line using LCD_command(0xC0).
- Display "BIN2:" on the LCD.
- Repeat the dustbin level checks for BIN2 using switches P0.06, P0.07, and P0.08.

10. Delay and Loop

- Introduce a delay of 1.5 seconds using ms_delay(1500) before the next iteration of the loop.

11. End of Main Loop

- Continue looping to monitor the dustbin status indefinitely.

12. Subroutine Definitions

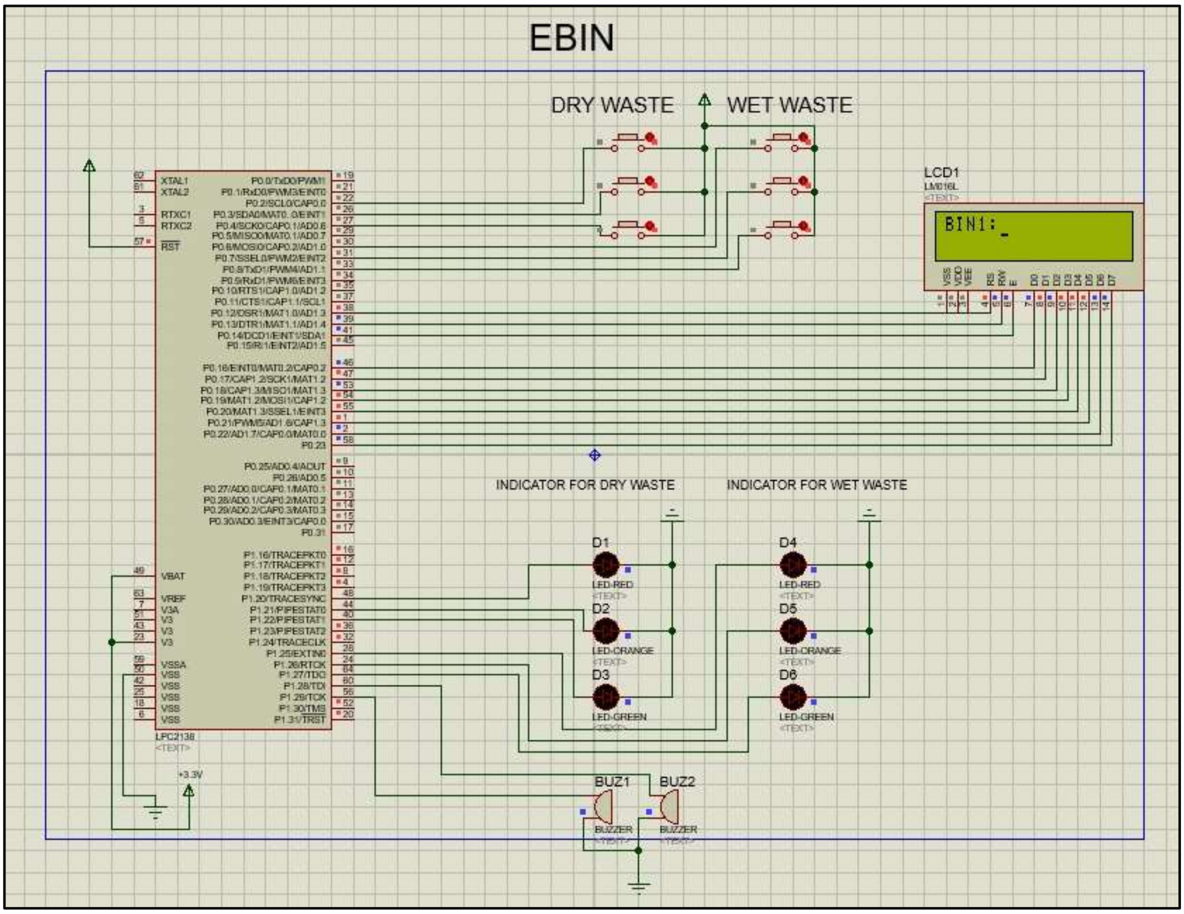
- Include definitions for ms_delay(), LCD_init(), LCD_command(), LCD_data(), and LCD_write_string() functions.

13. End

This algorithm outlines the steps of the provided C code for the project, describing the initialization, LCD display, dustbin level monitoring, and LED/buzzer controls.

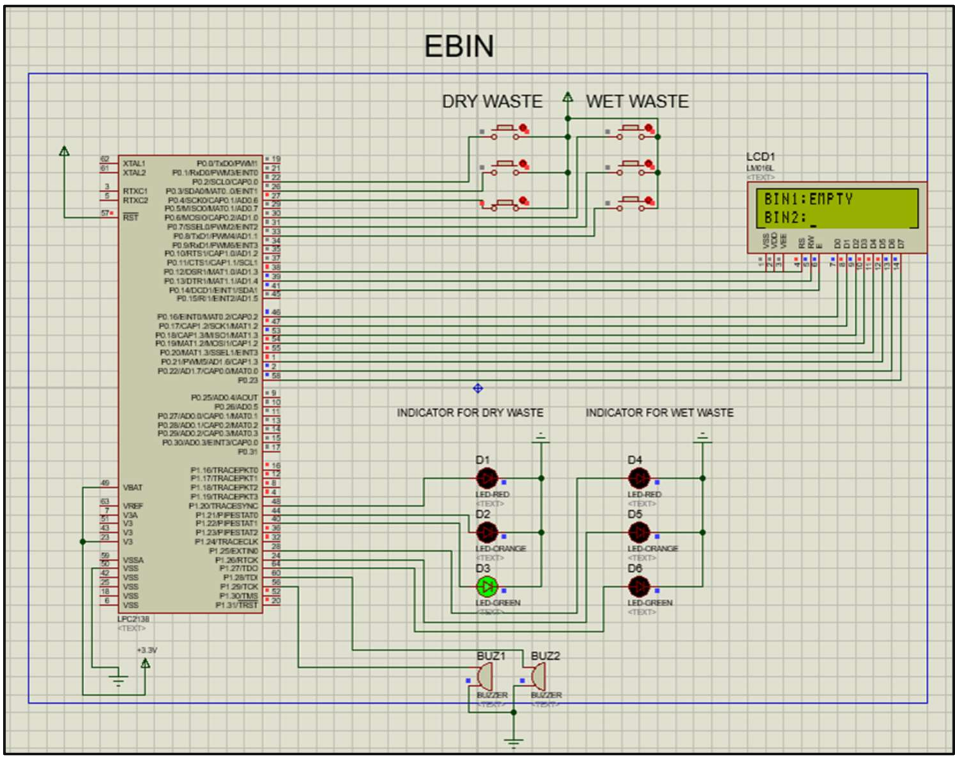
V. SIMULATION AND ANALYSIS

Design: Implemented the design using the proteus simulator

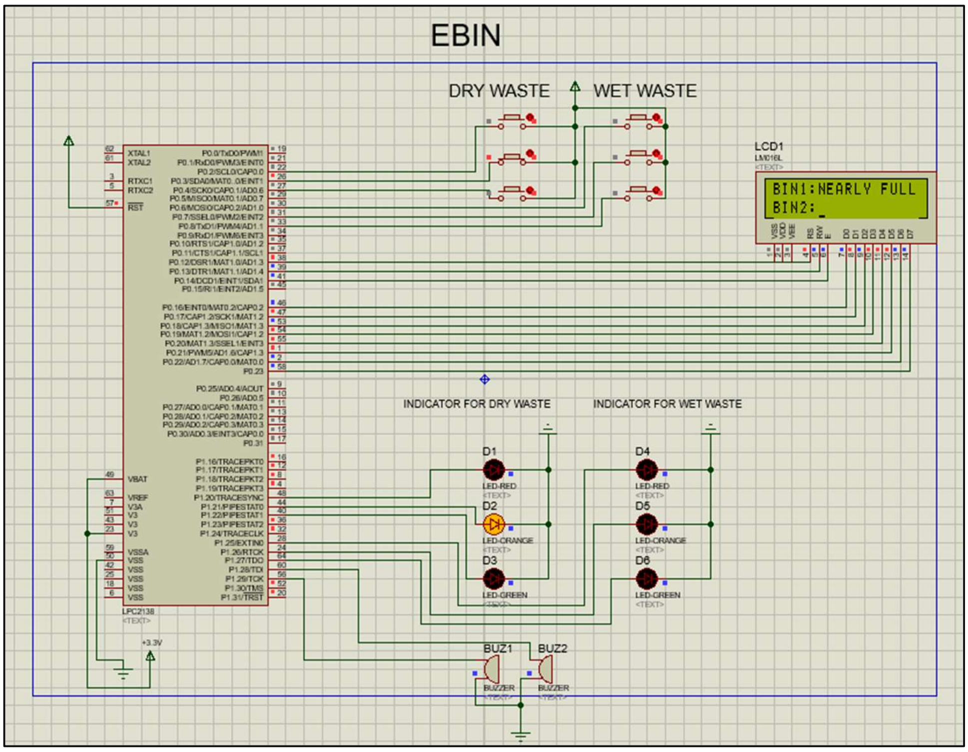


Analysis:

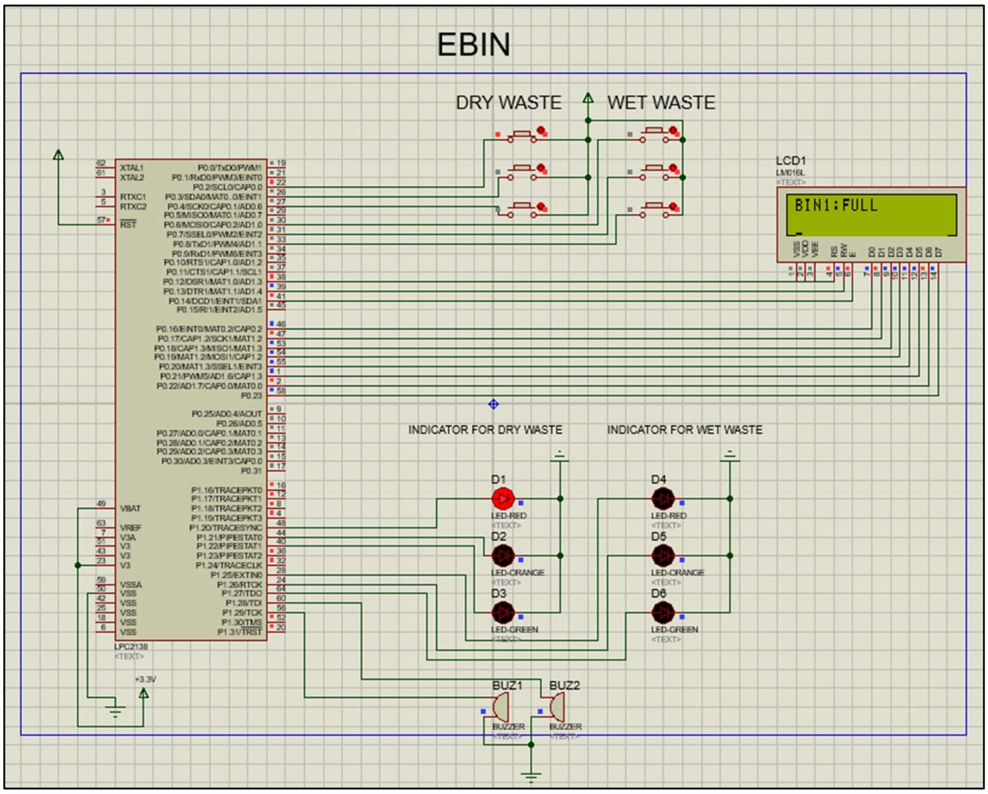
a) when the bin is empty the LCD displays “EMPTY” and is monitored with the blinking of Green LED.



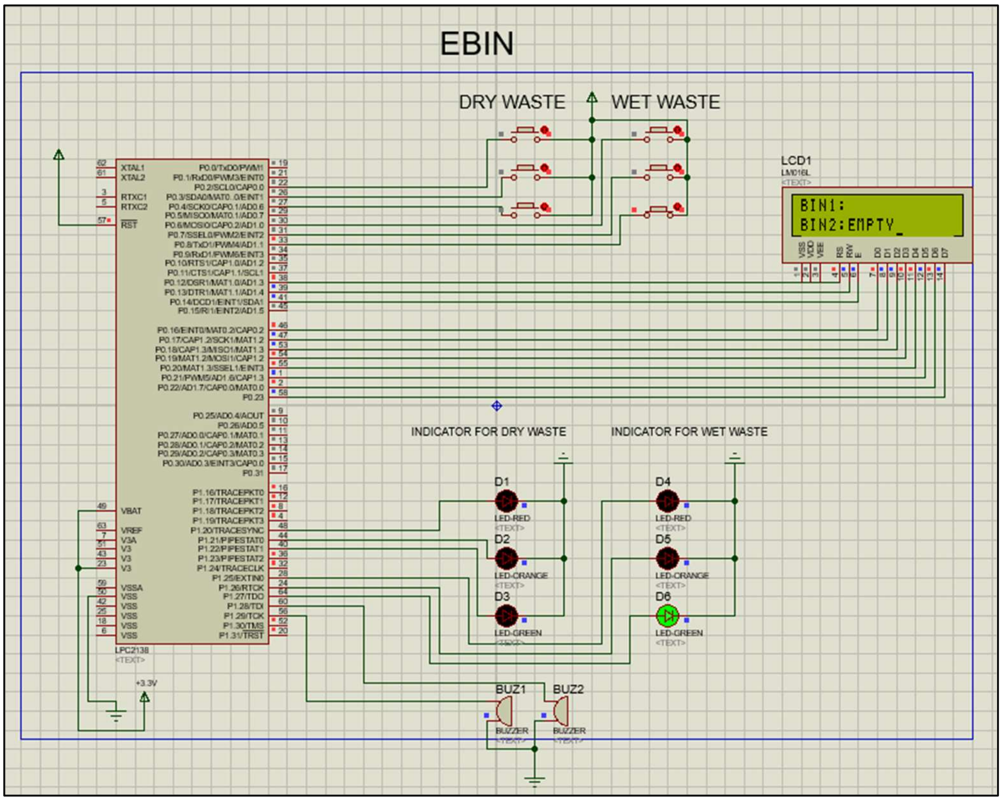
b) when the bin is 75% filled then the LCD displays “ALMOST FULL” and is monitored with the blinking of Orange LED.



c) when the bin is completely filled,then the LCD displays “FULL” and is monitored with the blinking of Red LED and is notified with a buzz through the Buzzer.



This procedure is continued with the BIN2 too indicating the levels of the dustbin2.



VI. SOURCE CODE

```
#include <lpc214x.h>
void LCD_init();
void LCD_command(unsigned char cmd);
void LCD_data(unsigned char data);
void LCD_write_string(unsigned char *str);
void ms_delay(int k);
int main()
{
   PINSEL1 = 0x00000000;  // select p0.16-
p0.23
   PINSEL2 = 0X00000000;  // p1.16-p1.18
   IODIR0 = 0x0FF << 16;
   IODIR0 |= (1 << 12) | (1 << 13) | (1 << 14);
// Set P0.12 (RS), P0.13 (RW), P0.14 (E) as
outputs

IODIR1 |= (1 << 20);  // Set P1.20 as output
(LED)
IODIR1 |= (1 << 21);  // Set P1.21 as output
(LED)
IODIR1 |= (1 << 22);  // Set P1.22 as output
(LED)
IODIR1 |= (1 << 25);  // Set P1.25 as output
(LED)
IODIR1 |= (1 << 26);  // Set P1.26 as output
(LED)
IODIR1 |= (1 << 27);  // Set P1.27 as output
(LED)
IODIR1 |= (1 << 28);  // Set P1.28 as Buzzer
IODIR1 |= (1 << 29);  // Set P1.29 as Buzzer
 LCD_init();

   while (1)
   {
      LCD_command(0x01);  //clear
      LCD_command(0x80);  // Set cursor to
the beginning of the first line
      LCD_write_string("BIN1:");  // Write to
the first line

      if ((IO0PIN & (1 << 2)))//if P0.02 switch is
on then ....
{
      IOSET1 = 1 << 20; //Turn the Red Led ON
      IOSET1 = 1 << 29;    //Turn the Buzzer ON

      LCD_write_string("FULL");//Will diplay on
the LCD
}
else
{

IOCLR1 = 1 << 20;//Turn the Red Led OFF
IOCLR1 = 1 << 29;////Turn the Buzzer OFF
}
if ((IO0PIN & (1 << 3))) //if P0.03 switch is on
{
IOSET1 = 1 << 21;// Turn on LED at P1.21
LCD_write_string("NEARLY FULL");//Will diplay
on the LCD
}
else
{

IOCLR1 = 1 << 21;// Turn off LED at P1.21
}
if ((IO0PIN & (1 << 4)))//if P0.03 switch is on
then ....
{
 IOSET1 = 1 << 22;// Turn on LED at P1.22
LCD_write_string("EMPTY");//Will diplay on
the LCD
}
else
                                    {
IOCLR1 = 1 << 22;// Turn OFF LED at P1.22
                                    }
LCD_command(0xC0);// Set cursor to the
beginning of the second line
LCD_write_string("BIN2:");
if ((IO0PIN & (1 << 6)))
                                    {
IOSET1 = 1 << 25;
IOSET1 = 1 << 28;
LCD_write_string("FULL");
                                    }
else
                                    {
  IOCLR1 = 1 << 28;
  IOCLR1 = 1 << 25;
                                    }
  if ((IO0PIN & (1 << 7)))
                                    {
IOSET1 = 1 << 26;
```

```
LCD_write_string("NEARLY FULL");
}
Else
{
IOCLR1 = 1 << 26;
}
if ((IO0PIN & (1 << 8)))
 IOSET1 = 1 << 27;
LCD_write_string("EMPTY");
}
else
{
 IOCLR1 = 1 << 27;
}
  ms_delay(1500);

  }

   return 0;
}

void ms_delay(int k)
{
   int i, j;
   for (i = 0; i < k; i++)
  {
      for (j = 0; j < 1008; j++)
      {
      }
   }
}

void LCD_init()
IOSET0 |= 1 << 14;      // E=1
   ms_delay(10);
   IOCLR0 |= 1 << 14;  // E=0
}

void LCD_write_string(unsigned char *str)
{
   int i = 0;
   while (str[i] != '\0')
   {
      LCD_data(str[i]);
      ms_delay(100);
      i++;
   }
}
```

```
{
   LCD_command(0x38);  // select 5x8 cell in
16x2
   ms_delay(100);
   LCD_command(0x01);  // clear lcd
   ms_delay(100);
   LCD_command(0x0E);  // set cursor
   ms_delay(100);
   LCD_command(0x06);  // Auto increment
   ms_delay(100);
   LCD_command(0x80);  // set cursor at
address location
   ms_delay(100);      // first cell row 0
}

void LCD_command(unsigned char cmd)
{
   IOCLR0 = 0xFF << 16;  // clear content in
data lines
   IOCLR0 |= (1 << 12) | (1 << 13); // RS=0,
RW=0
   IOSET0 |= cmd << 16;   // data=cmd
   IOSET0 |= 1 << 14;     // E=1
   ms_delay(10);
   IOCLR0 |= 1 << 14;  // E=0
}

void LCD_data(unsigned char data)
{
IOCLR0 = 0xFF << 16;  // clr data lines
   IOSET0 |= 1 << 12;     // RS=1
   IOCLR0 |= 1 << 13;     // RW=0
   IOSET0 |= data << 16;   // data =DATA
```

VII. SCOPE FOR IMPROVEMENT

The proposed E-Bin can be improved in future by incorporated the dustbin equipped with this system with the help of solar panels to generate energy for the sensors and communication devices. This sustainable approach reduces the reliance on external power sources.

VIII. INFERENCE

The E-Bin project employs LPC2138 microcontroller, LEDs, buzzer, and an LCD to revolutionize waste management. Through real-time monitoring, the system dynamically adapts LED indicators, a buzzer, and LCD displays to convey dustbin status. Utilizing an algorithmic approach, the project optimizes waste collection efficiency. Rigorous simulation and testing in Keil uVision and Proteus ensure robust functionality. Thorough documentation enhances project clarity, fostering adaptability and scalability. The user-friendly interface and environmental sustainability focus position E-Bin as a technological solution for efficient, adaptive, and cleaner urban waste management.

IX. CONCLUSION

In conclusion, the E-Bin project introduces a smart waste management solution that leverages the LPC2138 microcontroller, LEDs, a buzzer, and an LCD for real-time dustbin monitoring. The system's adaptability and efficiency are evident in its dynamic response to varying waste levels, communicated through intuitive LED indicators and LCD displays. Rigorous testing and documentation ensure reliability and facilitate future modifications. E-Bin aligns with environmental sustainability goals by promoting optimal waste collection practices, contributing to cleaner urban environments. This project showcases the potential of technology to enhance waste management processes, emphasizing user-friendliness, adaptability, and a commitment to a more sustainable future.

X. REFERENCES

- Steve Furber, “ARM system On Chip Architecture”, Addison Wesley, 2000.
- https://embeddedcenter.wordpress.com/ece-study-centre/display-module/lcd-16x2-lm016l/
- https://www.nxp.com/docs/en/data-sheet/LPC2131_32_34_36_38.pdf