**Report**

**(ES\_Lab\_Exam)**

Reported by: P. PRUDVI REDDY (BU21EECE0100361)

Mail Id: [ppoli@gitam.in](mailto:ppoli@gitam.in)

**Objective:**

* To develop an embedded system using an AT89C51 microcontroller, ADC0808, and LCD to monitor voltage levels and control LEDs based on predefined voltage thresholds for managing electrical equipment usage in a voltage-fluctuating environment.

Inputs:

1. Line Voltage (0-5V):

* This is the input voltage to be monitored. For study purposes, it's scaled to range from 0 to 5V.
* The voltage is fed into the ADC0808 for conversion to a digital signal.

Outputs:

1. LCD Display:

* Displays the real-time voltage level.
* Displays the current status of the LEDs (e.g., "2LEDs ON", "1LED ON", "LEDs OFF").

1. LEDs (LED1 and LED2):

* LED1 and LED2 are used to simulate the control of electrical equipment based on the voltage levels.
* If the voltage is greater than 3V, both LED1 and LED2 are turned on.
* If the voltage is between 2V and 3V, only LED1 is turned on.
* If the voltage is less than 2V, both LEDs are turned off.

1. Leds glowing:

* To know the leds are glow or not we can analyse in this statement where the anode part of the led the blink of the dot represents the OFF stage of the led where the no blink of the dot represents the ON stage of the led

**Logic:**

1. Initialization:

* Set up the LCD for display.
* Initialize the ADC0808 for voltage measurement.
* Initialize LEDs to the off state.

1. Voltage Monitoring:

* Continuously read the analog voltage (0-5V) using the ADC0808.
* Convert the ADC output (0-255) to the corresponding voltage level.

1. Display Voltage:

* Show the real-time voltage on the LCD.

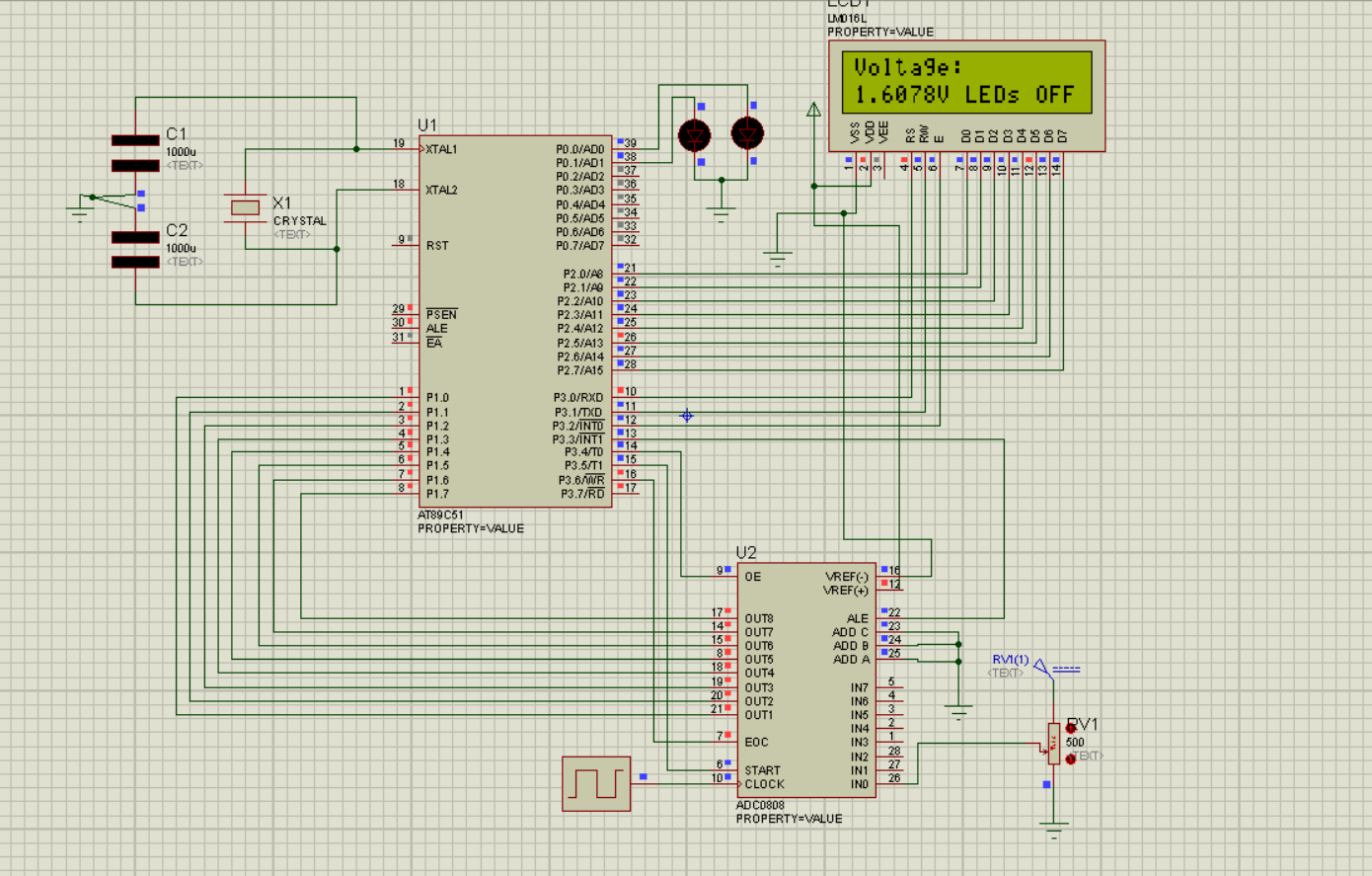
1. Control LEDs:

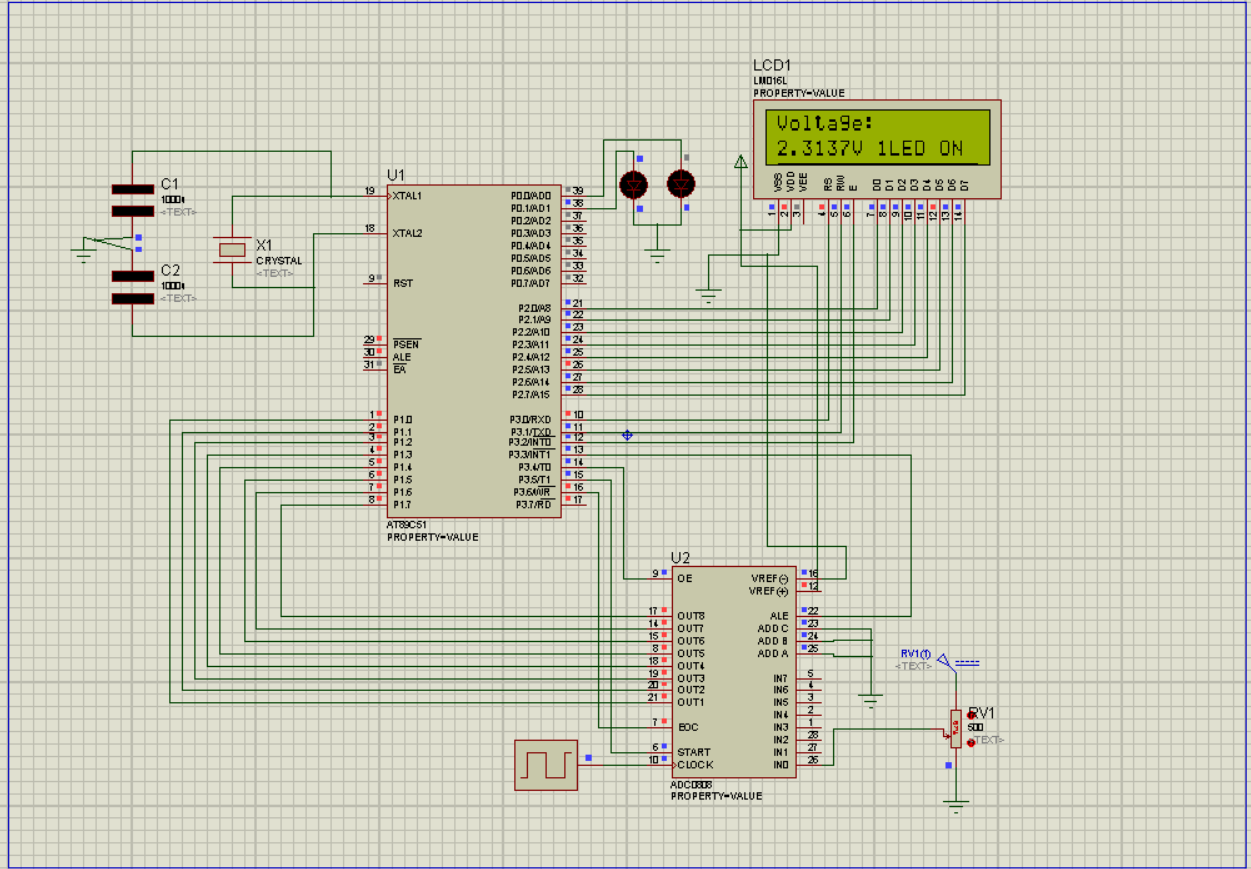
* If voltage > 3V:
* Turn on LED1 and LED2.
* Display "2LEDs ON" on the LCD.
* If 2V ≤ voltage ≤ 3V:
* Turn on LED1, turn off LED2.
* Display "1LED ON" on the LCD.
* If voltage < 2V:
* Turn off LED1 and LED2.
* Display "LEDs OFF" on the LCD.

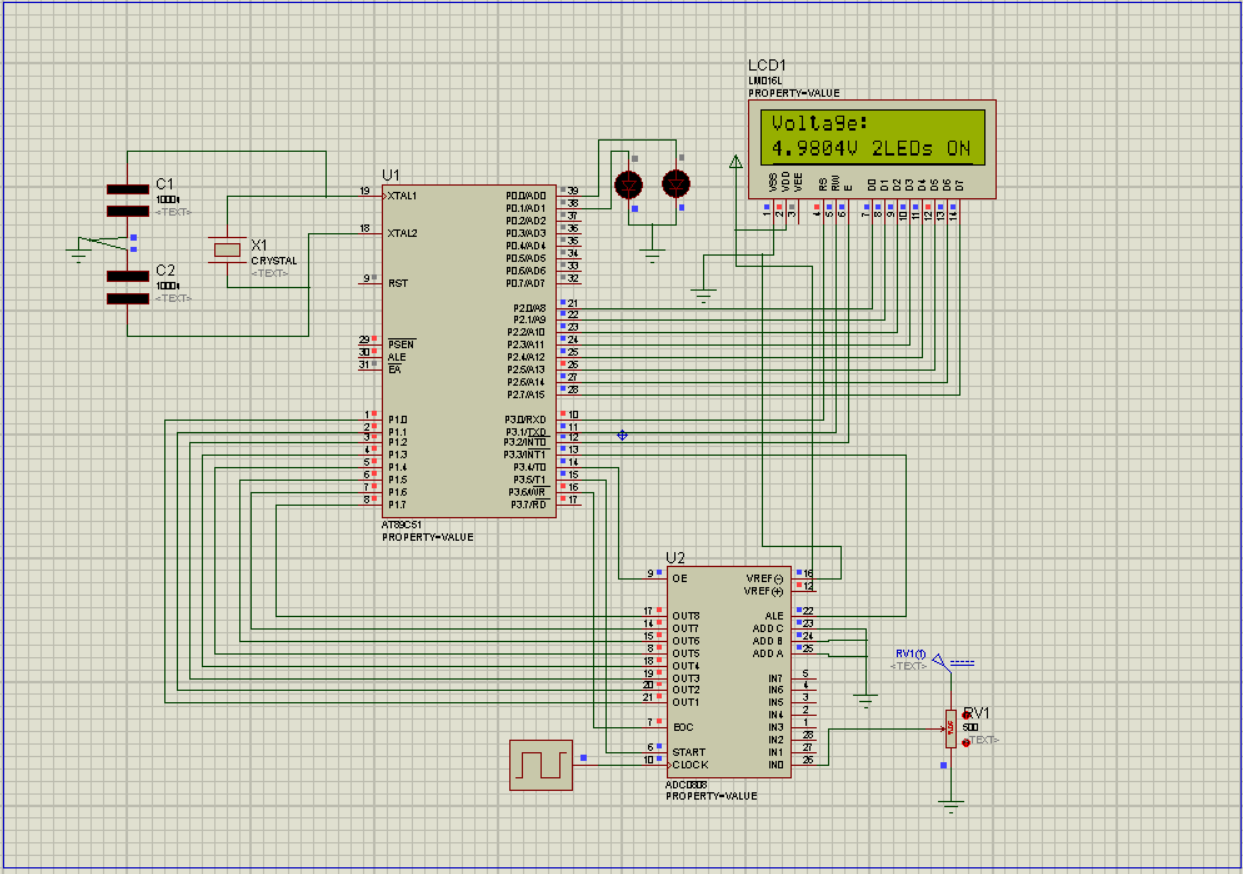
1. Continuous Loop:

* Repeat the monitoring, display, and control steps continuously for real-time voltage management.

Results:







Code:

#include <reg51.h>

#include <stdio.h>

// LCD Control Pins

sbit RS = P3^0;

sbit RW = P3^1;

sbit EN = P3^2;

// ADC Control Pins

sbit ALE = P3^3;

sbit OE = P3^4;

sbit START = P3^5;

sbit EOC = P3^6;

// LED Control Pins

sbit LED1 = P0^0;

sbit LED2 = P0^1;

void delay(unsigned int time);

void lcd\_init(void);

void lcd\_command(unsigned char command);

void lcd\_data(unsigned char Data);

void lcd\_display\_string(char \*str);

void lcd\_display\_voltage(float voltage);

unsigned char adc\_read(void);

// Lookup table for voltage values stored in code memory

float code adc\_to\_voltage[256] = {

0.0000, 0.0196, 0.0392, 0.0588, 0.0784, 0.0980, 0.1176, 0.1373, 0.1569, 0.1765,

0.1961, 0.2157, 0.2353, 0.2549, 0.2745, 0.2941, 0.3137, 0.3333, 0.3529, 0.3725,

0.3922, 0.4118, 0.4314, 0.4510, 0.4706, 0.4902, 0.5098, 0.5294, 0.5490, 0.5686,

0.5882, 0.6078, 0.6275, 0.6471, 0.6667, 0.6863, 0.7059, 0.7255, 0.7451, 0.7647,

0.7843, 0.8039, 0.8235, 0.8431, 0.8627, 0.8824, 0.9020, 0.9216, 0.9412, 0.9608,

0.9804, 1.0000, 1.0196, 1.0392, 1.0588, 1.0784, 1.0980, 1.1176, 1.1373, 1.1569,

1.1765, 1.1961, 1.2157, 1.2353, 1.2549, 1.2745, 1.2941, 1.3137, 1.3333, 1.3529,

1.3725, 1.3922, 1.4118, 1.4314, 1.4510, 1.4706, 1.4902, 1.5098, 1.5294, 1.5490,

1.5686, 1.5882, 1.6078, 1.6275, 1.6471, 1.6667, 1.6863, 1.7059, 1.7255, 1.7451,

1.7647, 1.7843, 1.8039, 1.8235, 1.8431, 1.8627, 1.8824, 1.9020, 1.9216, 1.9412,

1.9608, 1.9804, 2.0000, 2.0196, 2.0392, 2.0588, 2.0784, 2.0980, 2.1176, 2.1373,

2.1569, 2.1765, 2.1961, 2.2157, 2.2353, 2.2549, 2.2745, 2.2941, 2.3137, 2.3333,

2.3529, 2.3725, 2.3922, 2.4118, 2.4314, 2.4510, 2.4706, 2.4902, 2.5098, 2.5294,

2.5490, 2.5686, 2.5882, 2.6078, 2.6275, 2.6471, 2.6667, 2.6863, 2.7059, 2.7255,

2.7451, 2.7647, 2.7843, 2.8039, 2.8235, 2.8431, 2.8627, 2.8824, 2.9020, 2.9216,

2.9412, 2.9608, 2.9804, 3.0000, 3.0196, 3.0392, 3.0588, 3.0784, 3.0980, 3.1176,

3.1373, 3.1569, 3.1765, 3.1961, 3.2157, 3.2353, 3.2549, 3.2745, 3.2941, 3.3137,

3.3333, 3.3529, 3.3725, 3.3922, 3.4118, 3.4314, 3.4510, 3.4706, 3.4902, 3.5098,

3.5294, 3.5490, 3.5686, 3.5882, 3.6078, 3.6275, 3.6471, 3.6667, 3.6863, 3.7059,

3.7255, 3.7451, 3.7647, 3.7843, 3.8039, 3.8235, 3.8431, 3.8627, 3.8824, 3.9020,

3.9216, 3.9412, 3.9608, 3.9804, 4.0000, 4.0196, 4.0392, 4.0588, 4.0784, 4.0980,

4.1176, 4.1373, 4.1569, 4.1765, 4.1961, 4.2157, 4.2353, 4.2549, 4.2745, 4.2941,

4.3137, 4.3333, 4.3529, 4.3725, 4.3922, 4.4118, 4.4314, 4.4510, 4.4706, 4.4902,

4.5098, 4.5294, 4.5490, 4.5686, 4.5882, 4.6078, 4.6275, 4.6471, 4.6667, 4.6863,

4.7059, 4.7255, 4.7451, 4.7647, 4.7843, 4.8039, 4.8235, 4.8431, 4.8627, 4.8824,

4.9020, 4.9216, 4.9412, 4.9608, 4.9804, 5.0000

};

void main(void) {

unsigned char adc\_value;

float voltage;

lcd\_init(); // Initialize the LCD

lcd\_display\_string("Voltage: ");

while(1) {

adc\_value = adc\_read(); // Read ADC value

voltage = adc\_to\_voltage[adc\_value]; // Get voltage from lookup table

lcd\_command(0xC0); // Move cursor to the second line of LCD

lcd\_display\_voltage(voltage); // Display the voltage on LCD

// Clear previous LED status

LED1 = 0;

LED2 = 0;

if (voltage > 3.0) {

LED1 = 1; // Turn on LED1

LED2 = 1; // Turn on LED2

lcd\_command(0xC8); // Move cursor to display status

lcd\_display\_string("2LEDs ON ");

} else if (voltage >= 2.0) {

LED1 = 1; // Turn on LED1

LED2 = 0; // Turn off LED2

lcd\_command(0xC8); // Move cursor to display status

lcd\_display\_string("1LED ON ");

} else {

LED1 = 0; // Turn off LED1

LED2 = 0; // Turn off LED2

lcd\_command(0xC8); // Move cursor to display status

lcd\_display\_string("LEDs OFF ");

}

delay(500); // Delay for some time (500 ms)

}

}

// Function to initialize the LCD

void lcd\_init(void) {

lcd\_command(0x38); // Initialize LCD in 8-Bit mode

delay(1);

lcd\_command(0x0C); // Display ON, Cursor OFF

delay(1);

lcd\_command(0x06); // Auto increment cursor

delay(1);

lcd\_command(0x01); // Clear display

delay(1);

lcd\_command(0x80); // Move cursor to the first line

delay(1);

}

// Function to send command to LCD

void lcd\_command(unsigned char command) {

P2 = command;

RS = 0; // Select command Register

RW = 0; // Write operation

EN = 1;

delay(1);

EN = 0;

delay(1);

}

// Function to send data to LCD

void lcd\_data(unsigned char Data) {

P2 = Data;

RS = 1; // Select Data Register

RW = 0; // Write operation

EN = 1;

delay(1);

EN = 0;

delay(1);

}

// Function to display string on LCD

void lcd\_display\_string(char \*str) {

while (\*str) {

lcd\_data(\*str++);

}

}

// Function to display voltage on LCD

void lcd\_display\_voltage(float voltage) {

char buffer[16];

sprintf(buffer, "%1.4fV", voltage); // Format voltage with 4 decimal places

lcd\_display\_string(buffer);

}

// Function to read Data from ADC

unsigned char adc\_read(void) {

unsigned char adc\_value;

ALE = 1;

START = 1;

delay(1);

ALE = 0;

START = 0;

while (EOC == 1); // Wait for end of conversion

OE = 1;

adc\_value = P1; // Read ADC value from Port 1

OE = 0;

return adc\_value;

}

// Function to generate delay

void delay(unsigned int time) {

unsigned int i, j;

for(i = 0; i < time; i++) {

for(j = 0; j < 1275; j++);

}

}

Common mistakes How do I overcome:

* the common mistakes I faced by implementing this project that is where some of the required components are not found, and due to large network the connections are mismatched during connections, while running the program the project where the led are not glow and also i faced the problem while adjusting the potentiometer
* from the above common mistakes I overcome with this following steps they are where I surf over the internet to know the exact of the components in the software they are called , In the connection problem where I took more time and analyze the situation make rough diagram and implement successfully , in the case of the led where observe the led dots so I get respect with desired outcome , where in the potentiometer where I taken more time observe its characteristics in the internet and overcome this problem.