K Lokeshwar Reddy
BU21EECE0100334

8051(MICRO-CONTROLLER) (Trainer: Dr. Jeevan K M)

Learning Objective:

- Understanding how to create patterns of LED blinking using the 8051 microcontroller.

Inputs and Outputs:

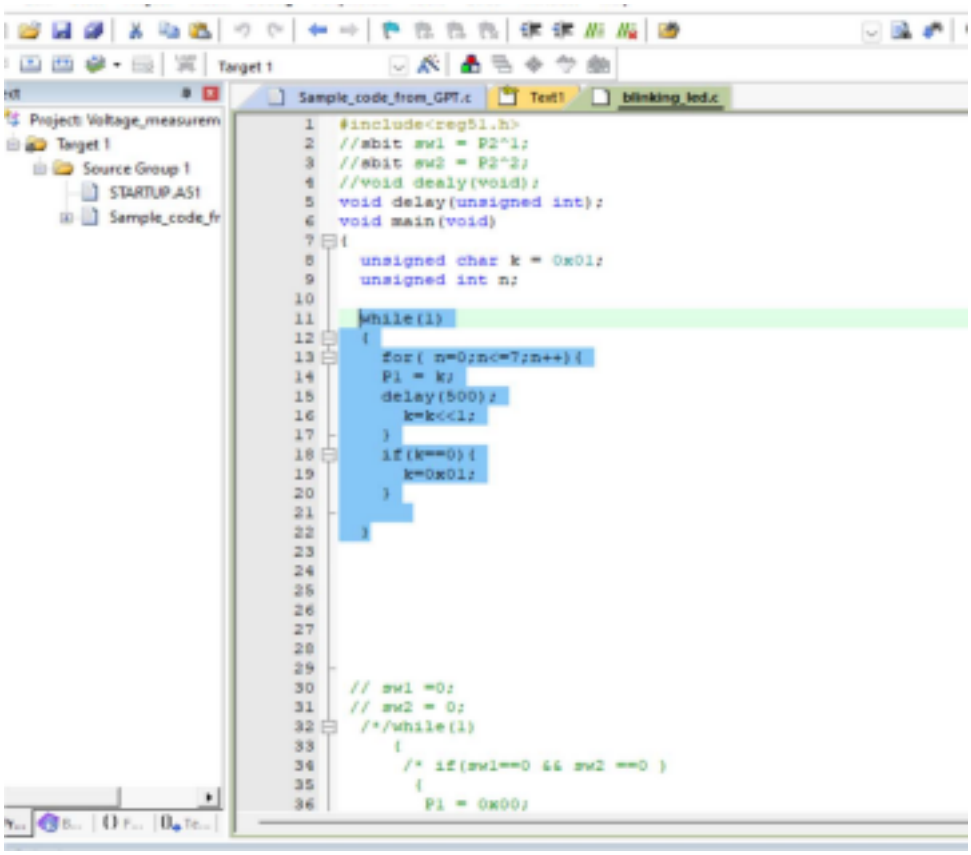
- Inputs: None
- Outputs: LED states (ON/OFF)

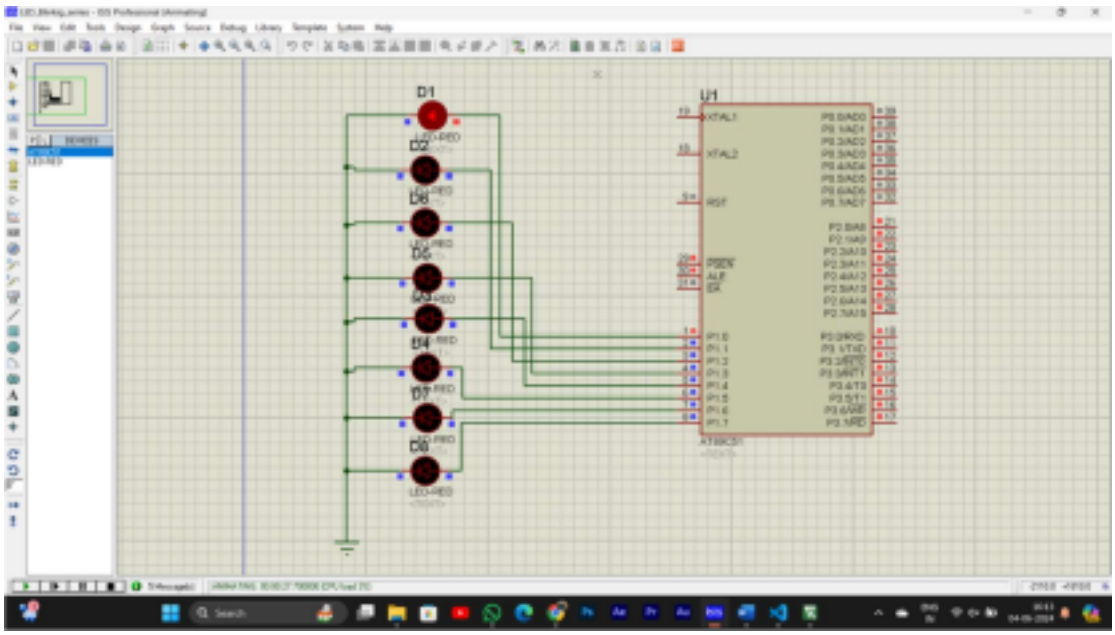
Logic:

- LEDs are connected to the microcontroller. By assigning a hexadecimal value to the port, the microcontroller converts this value to binary. Each bit in the binary representation controls an individual LED: a bit value of '1' turns the LED on, and '0' turns it off. For example, if `P1 = 0x01`, only the first LED (connected to the least significant bit) will be on, and the rest will be off.
- The other terminals of the LEDs are connected to the ground, creating a simple circuit for controlling the LEDs based on the port value.
- Various LED blinking patterns can be implemented using this basic principle. Two main approaches can be utilized:
    1. Direct assignment of hexadecimal values to the port.
    2. Using bitwise left shift operations within a loop to create sequential blinking patterns.

Common Mistakes:

- Syntax Errors: Common when writing embedded C code.
- Indentation Errors: Proper code formatting is crucial to avoid logical errors.
- Port Mismatch: Ensuring the correct port is being used for LED control is essential to achieve the desired output.





ADC Using 8051

Learning Objective:

- Understanding the process of converting an analog signal to a digital signal using the ADC 0808.
- Learning how to interface the ADC 0808 with the 8051

microcontroller. Inputs and Outputs:

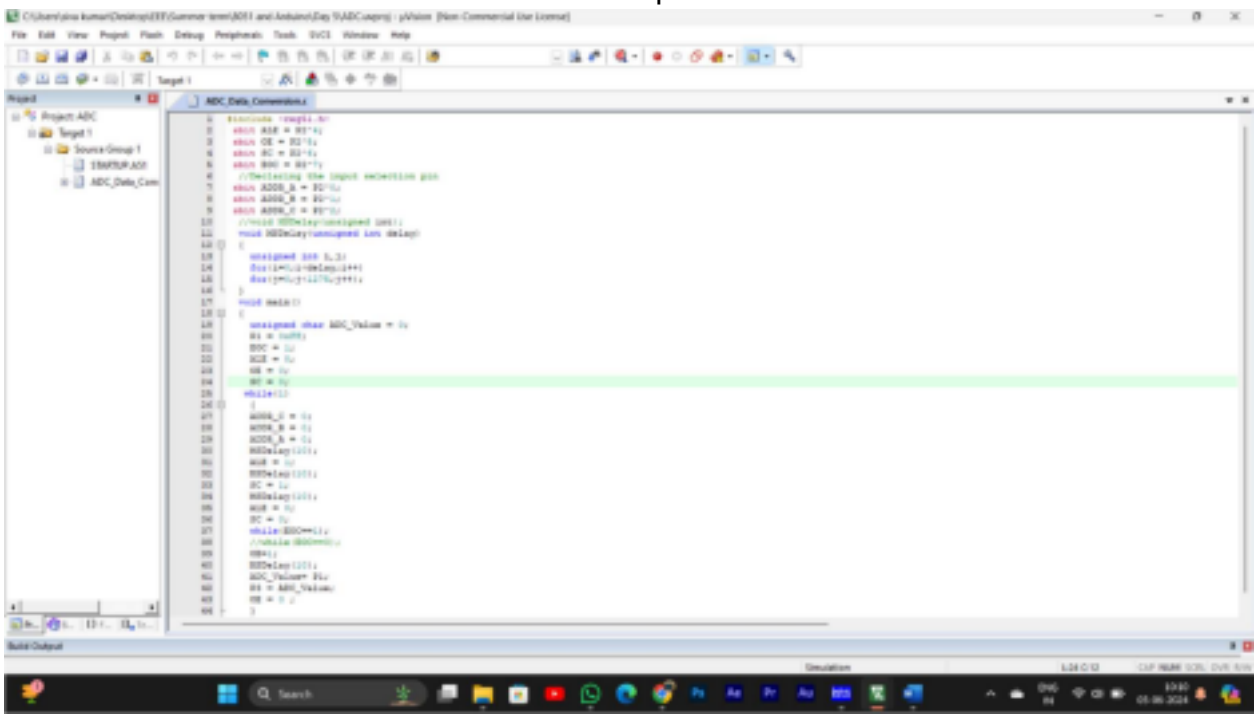
- Input: A potentiometer (1k or 10k ohms).
- Outputs: LEDs.

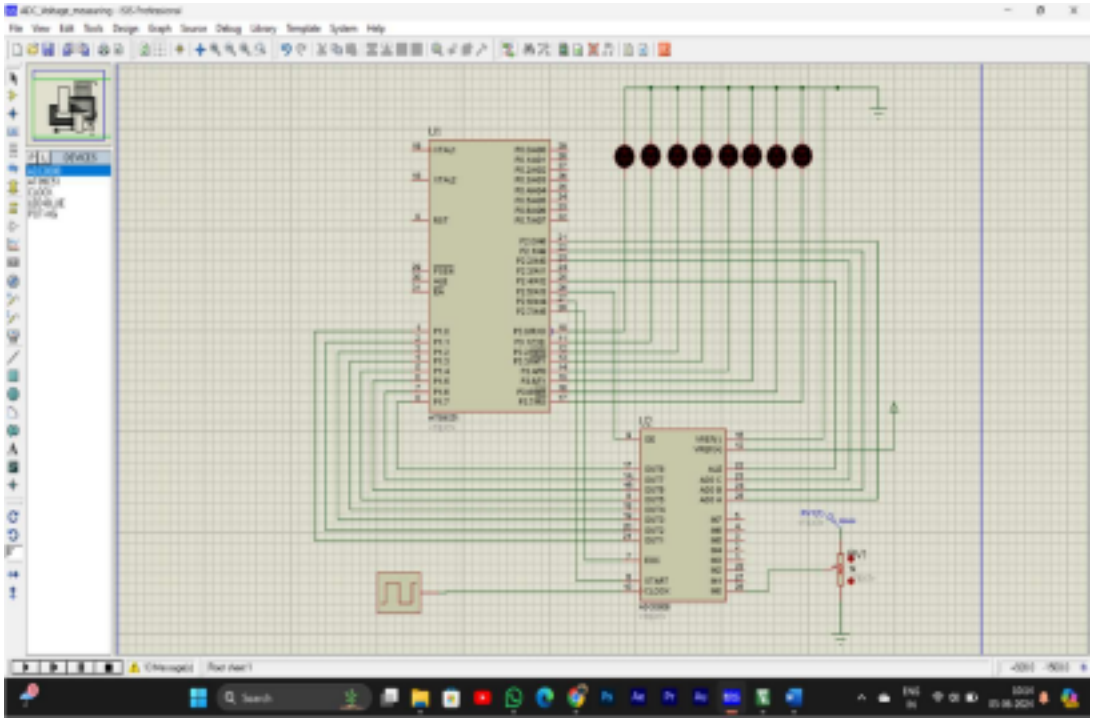
Logic:

- The analog output from the potentiometer is fed into one of the input channels of the ADC 0808.
- The digital output pins (D0-D7) of the ADC 0808 are connected to Port 1 of the 8051 microcontroller.
- The selection lines (A, B, C) of the ADC are connected to Port 2 of the 8051 to select the input channel.
- Additional control lines such as End of Conversion (EOC), Start, Address Latch Enable (ALE), and Clock Pulse are also connected to Port 2.
- The LEDs, which display the digital value of the potentiometer reading, are connected to Port 3 of the 8051 microcontroller.
- Based on the variation in resistance of the potentiometer, the digital value output from the ADC will change, which in turn will change the state of the LEDs connected to Port 3.

Common Mistakes:

- Selection Line Configuration: Double-check the configuration and connections of the selection lines (A, B, C) to ensure the correct input channel is selected on the ADC 0808.
- EOC Pin Connection: Ensure the End of Conversion (EOC) pin of the ADC 0808 is properly connected to the 8051 to correctly detect when a conversion is complete.





8051

Learning Objective:

- Learning how to control LED blinking using the 8051 microcontroller.

Inputs and Outputs:

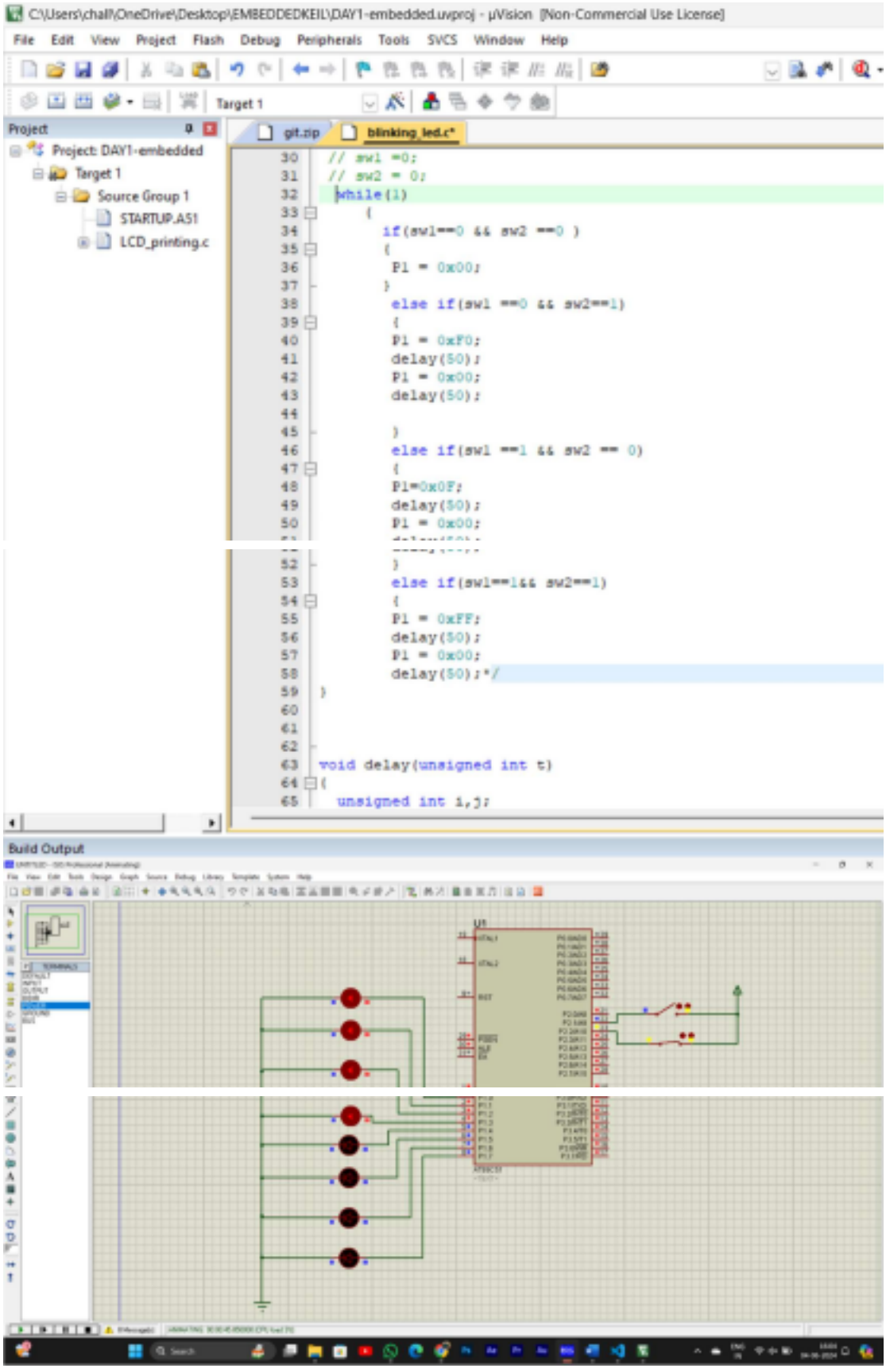
- Inputs: Switches
- Outputs: LEDs

Logic:

- LEDs are connected to Port 1 (P1) of the 8051 microcontroller.
- The code involves passing hexadecimal values to Port 1, which determines the state of each LED.
- A delay function is used to create the blinking effect by toggling the LEDs on and off.

Common Mistakes:

- Project Creation in Keil Software: Errors in creating a project or missing the step of creating a target before running the code.
- Syntax and Indentation Errors: Mistakes in code syntax and improper indentation can lead to compilation errors.
- Port Confusion: Errors in connecting the LEDs to the correct ports on the microcontroller.



Voltage Measurement and LCD displaying(8051)

Learning Objective:

- Understanding how to measure voltage variations using an ADC.
- Displaying the voltage variations on LEDs.
- Displaying the voltage measurement status on an LCD display.

Inputs and Outputs:

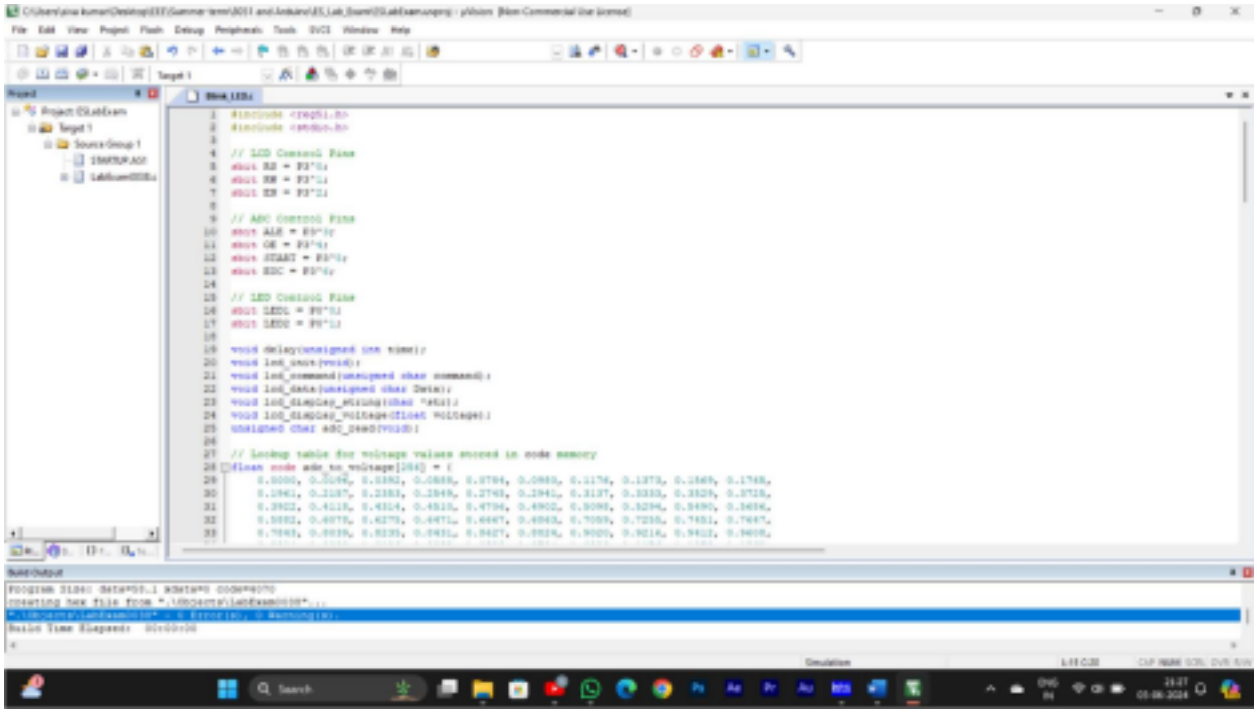
- Input: Potentiometer.
- Outputs: LEDs and LCD display.

Logic:

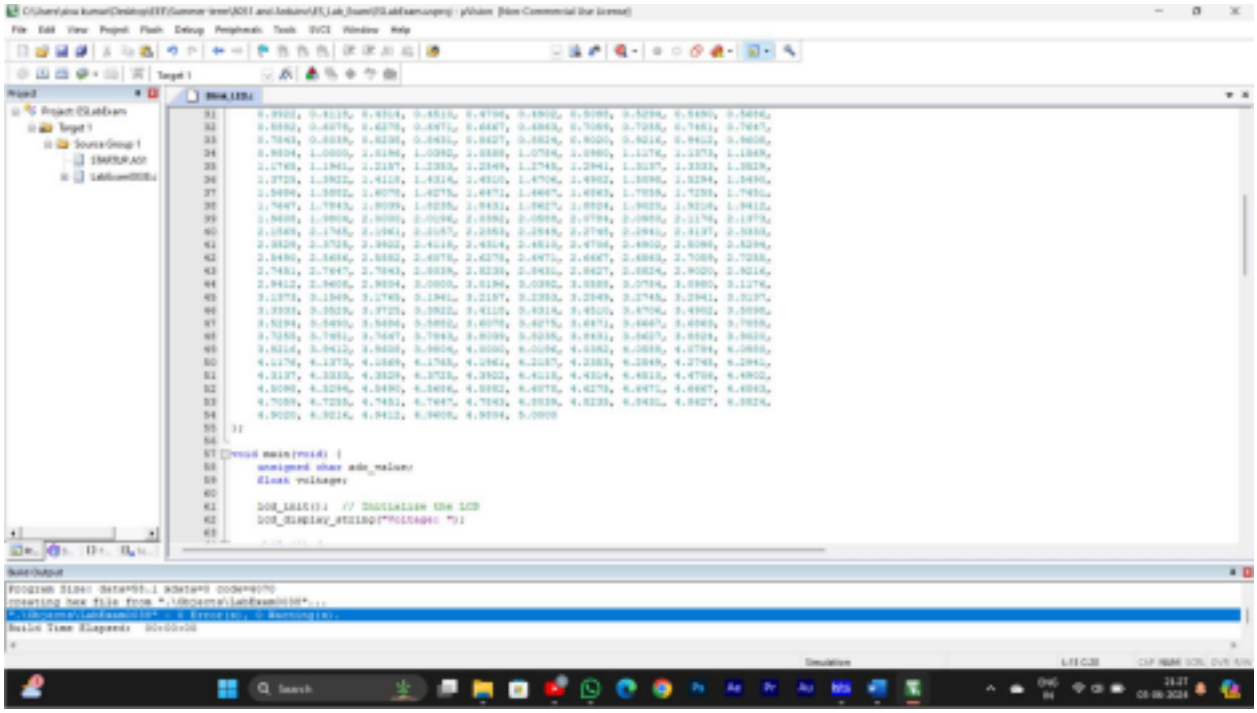
1. Connections:
    - Connect the potentiometer output to an input channel of the ADC 0808.
    - Provide a 5V power supply to the ADC 0808.
    - Connect the digital output pins (D0-D7) of the ADC 0808 to a port on the 8051 microcontroller.
    - Connect the LCD display and LEDs to the 8051 microcontroller.
2. Operation:
    - The ADC converts the analog signal from the potentiometer to a digital signal.
    - The 8051 microcontroller reads the digital signal from the ADC.
    - Based on the digital value, control the LEDs to indicate voltage levels.
    - Convert the digital value to a corresponding voltage value and display it on the LCD.

Common Mistakes:

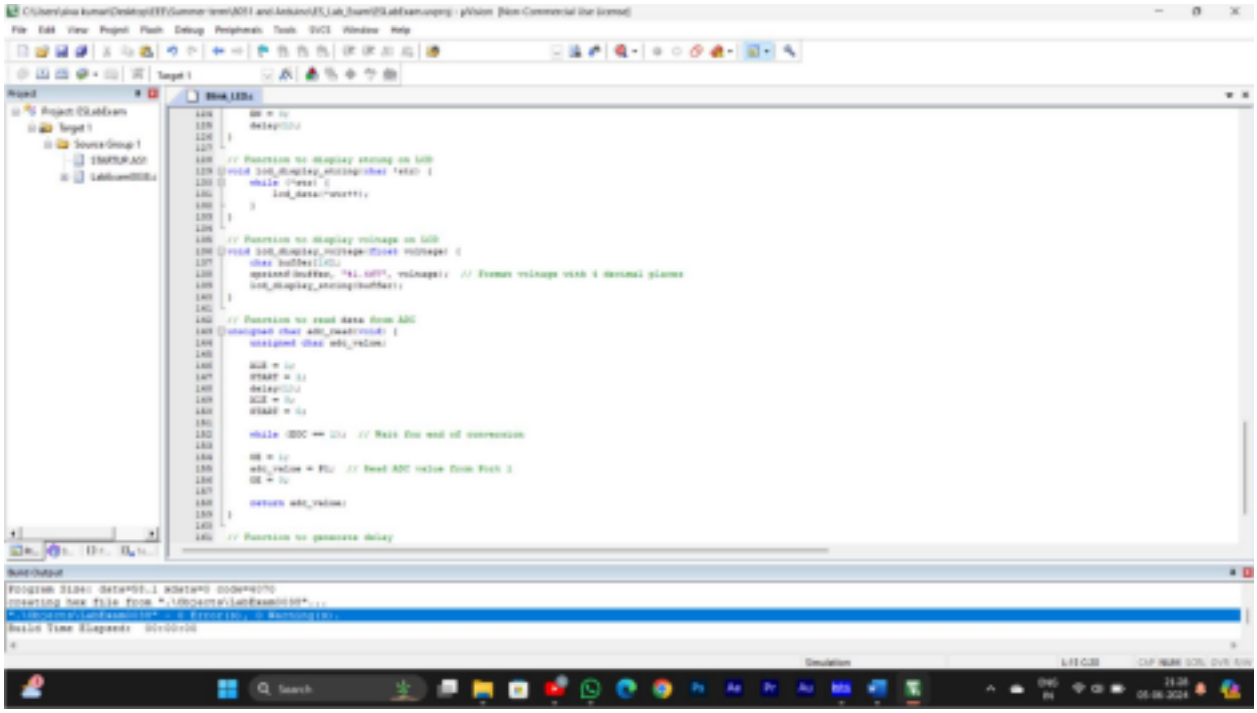
- Port Mismatch: Ensure that the port names used in the code match the actual circuit connections.
- Proper Connections: Verify that all connections are secure. A single loose connection can lead to data loss and no output.
- Input Levels: Ensure that input levels to the microcontroller and ADC are correct (either high or low as required).

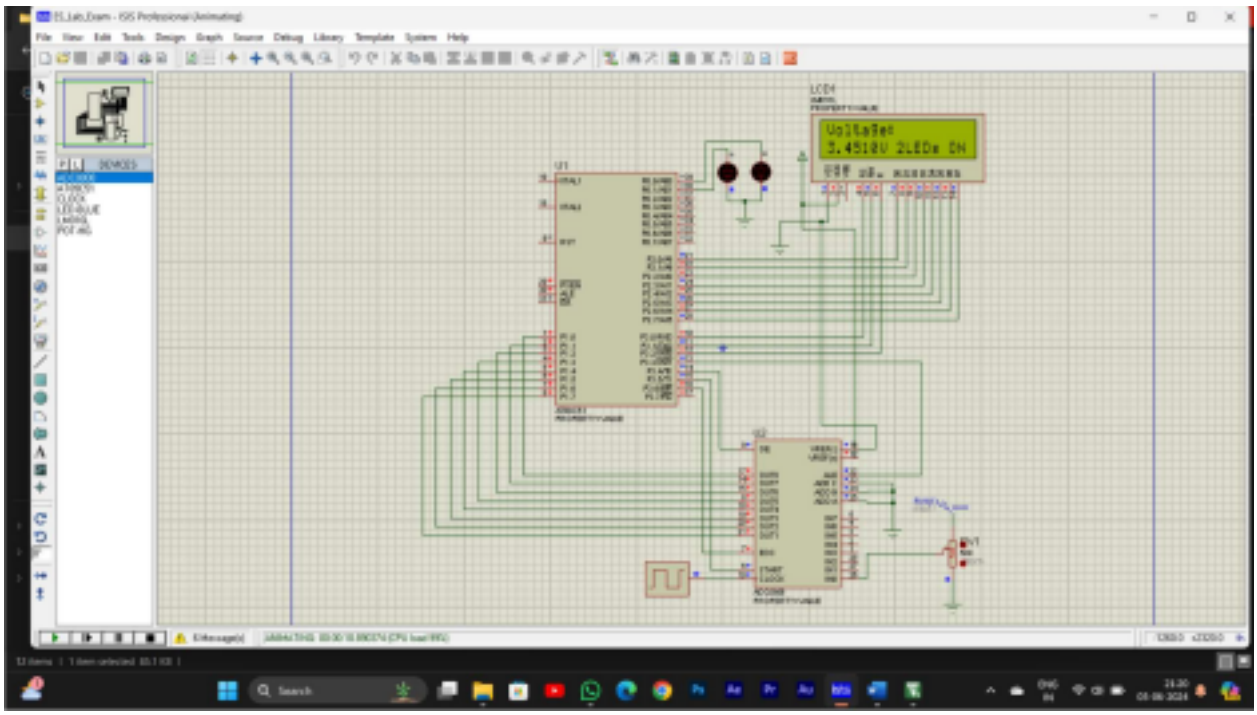


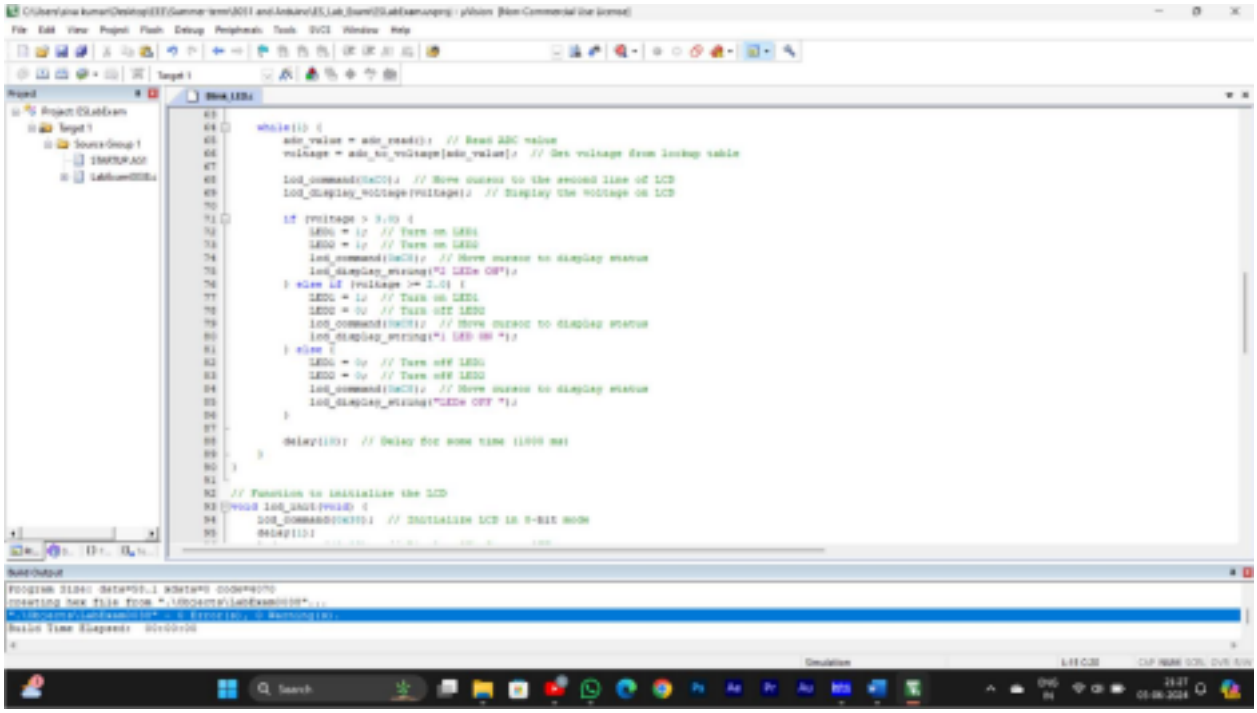
```
#include <reg51.h>
#include <stdio.h>

// LCD Control Pins
sbit RS = P3^5;
sbit RW = P3^6;
sbit EN = P3^7;

// ADC Control Pins
sbit ALE = P0^3;
sbit OE = P3^4;
sbit START = P3^0;
sbit EOC = P0^4;

// 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, 0.0039, 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: ");
```

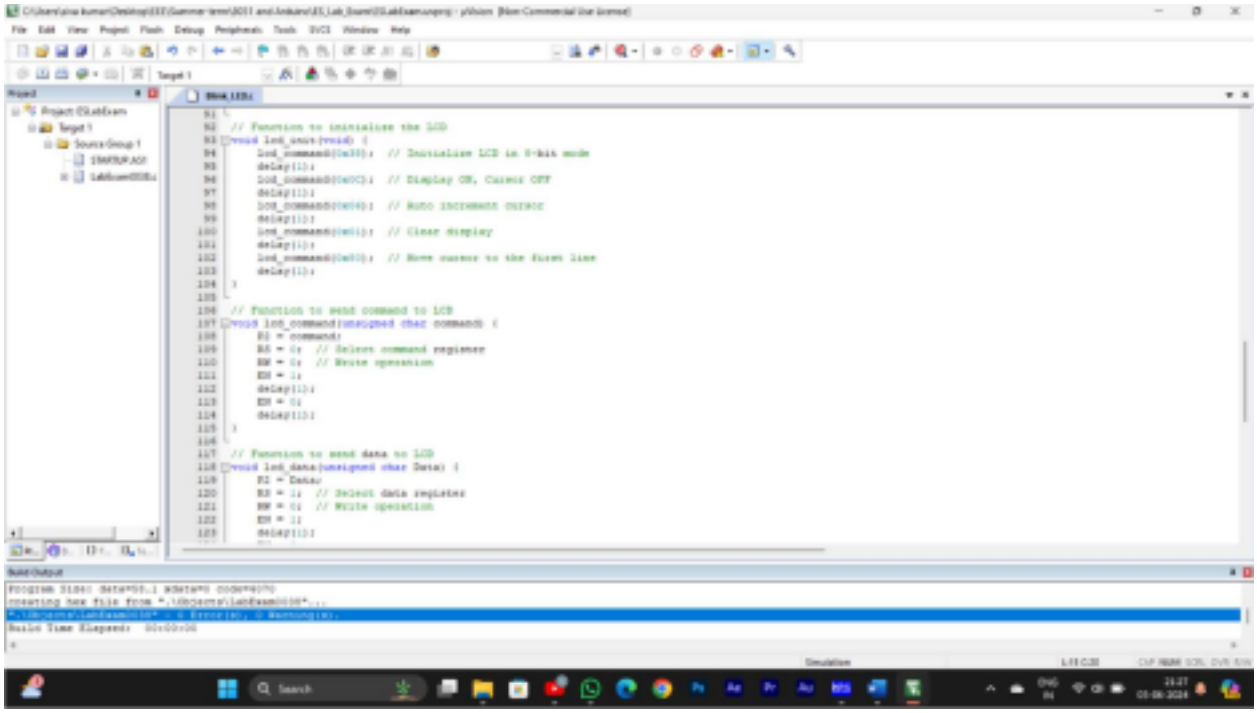



```
    0.9020, 0.9216, 0.9412, 0.9608, 0.9804, 0.0000
};

void main(void) {
    unsigned char adc_value;
    float voltage;

    lcd_init();  // Initialize the LCD
    lcd_display_string("Voltage: ");
```









8051(micro-controller) using LCD Display

Learning Objective:

- Learning how to use the 8051 microcontroller to display text on an LCD.

Inputs and Outputs:

- Inputs: Commands for the LCD display.
- Output: Displayed text on the LCD.

Logic:

- Implement functions to send commands and data to the LCD.
- Use various LCD commands such as clear display, cursor positioning, and line shifting.
- Convert numerical values to ASCII before displaying them on the LCD.
- Make basic connections to power supply, ground, and enable pins.

Common Mistakes:

- Command Mismatch: Ensure the correct commands are sent to the LCD.
- Command Positioning: Correct positioning of commands is crucial for proper display.
- Clear Display and New Line Commands: Use appropriate commands to clear the display or move to a new line when needed.