

Labs/Lab02/LED_Matrix/LEDS.ino

```
#include <Arduino.h>

#define OP_DECODEMODE 8
#define OP_SCANLIMIT 10
#define OP_SHUTDOWN 11
#define OP_DISPLAYTEST 14
#define OP_INTENSITY 10

int DIN = 47;
int CS = 49;
int CLK = 51;
int THUMBSTICK_X = A0;
int THUMBSTICK_Y = A1;

byte spidata[2];

// Function prototypes
void spiTransfer(volatile byte opcode, volatile byte data);
int readThumbstickValue(int pin);
int convertToIndex(int value, bool invert = false);

// Setup function
void setup(){
    // Configure pins for the LED matrix
    pinMode(DIN, OUTPUT);
    pinMode(CS, OUTPUT);
    pinMode(CLK, OUTPUT);
    digitalWrite(CS, HIGH);

    // Initialize the LED matrix
    spiTransfer(OP_DISPLAYTEST, 0);
    spiTransfer(OP_SCANLIMIT, 7);
    spiTransfer(OP_DECODEMODE, 0);
    spiTransfer(OP_SHUTDOWN, 1);

    // Clear the display
    for (int i = 0; i < 8; i++) {
        spiTransfer(i, 0);
    }

    // Initialize serial communication
    Serial.begin(9600);
}

// Main loop function
void loop(){
    // Read the thumbstick values and convert them to row and column indices
    int row = convertToIndex(readThumbstickValue(THUMBSTICK_Y));
    int col = convertToIndex(readThumbstickValue(THUMBSTICK_X), true);

    // Print the row and column values to the serial monitor
    Serial.print("Row: ");
```

```
Serial.print(row);
Serial.print(", Col: ");
Serial.println(col);

// Light up the LED at the specified row and column
spiTransfer(row, 1 << col);
delay(50);

// Turn off the LED at the specified row and column
spiTransfer(row, 0);
}

// Function to transfer data to the LED matrix
void spiTransfer(volatile byte opcode, volatile byte data){
    int offset = 0;
    int maxbytes = 2;

    // Clear the SPI data buffer
    for(int i = 0; i < maxbytes; i++) {
        spidata[i] = (byte)0;
    }

    // Load SPI data
    spidata[offset+1] = opcode+1;
    spidata[offset] = data;

    // Send SPI data
    digitalWrite(CS, LOW);
    for(int i=maxbytes;i>0;i--)
        shiftOut(DIN,CLK,MSBFIRST,spidata[i-1]);
    digitalWrite(CS,HIGH);
}

// Function to read the thumbstick value
int readThumbstickValue(int pin) {
    return analogRead(pin);
}

// Function to convert the thumbstick value to a row or column index
int convertToIndex(int value, bool invert) {
    if (invert) {
        value = 1023 - value;
    }
    return (int)((value / 1023.0) * 8);
}
```