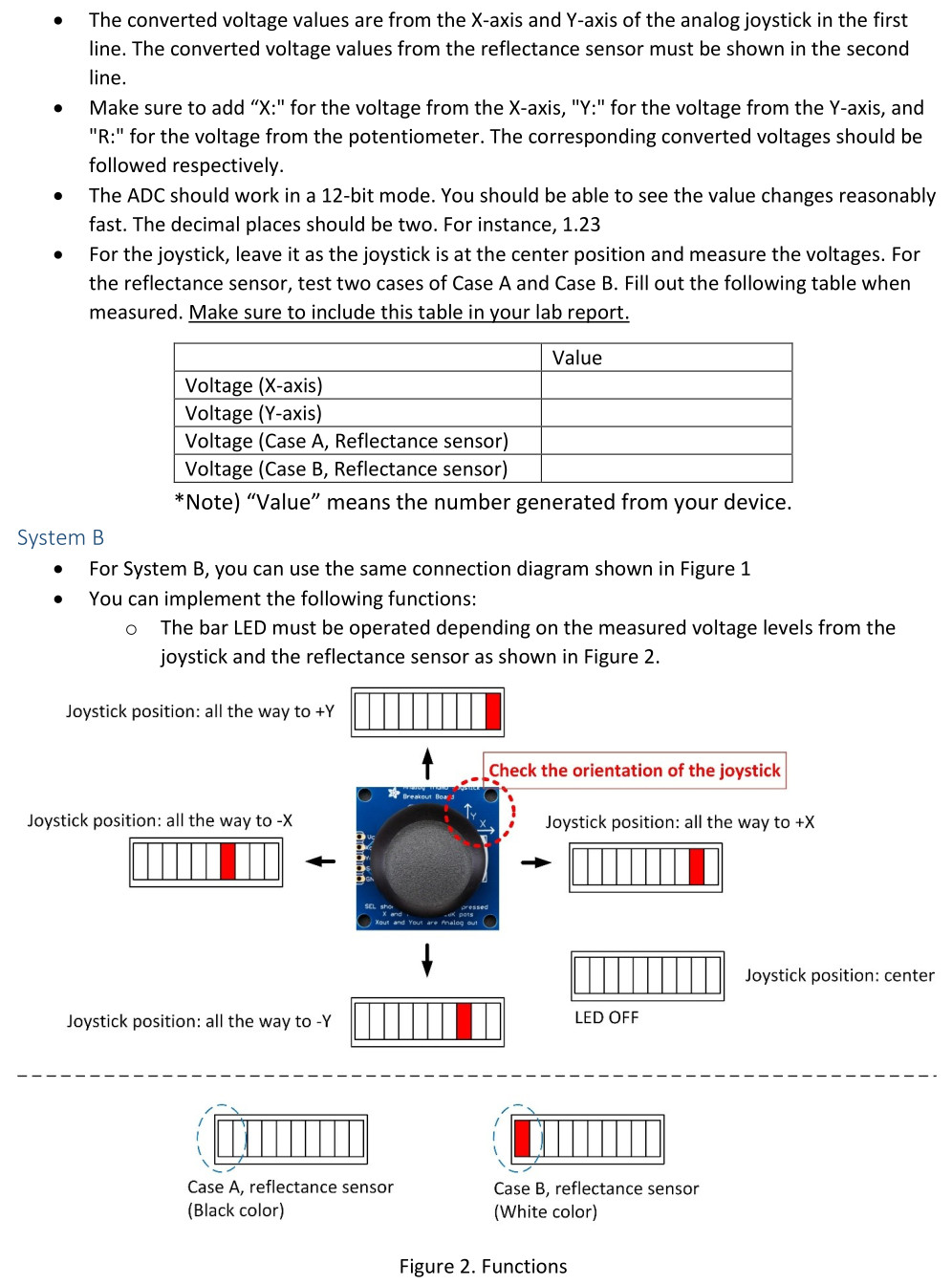
**Prompt:**



For System B, you can use the connection scheme described below.

You can implement the following functions:

The bar LED must be operated depending on the measured voltage levels from the joystick and the reflectance sensor as explains below.

Joystick Position and Bar LED Behavior:

Joystick position: all the way to +Y: LED1 turns on.

Joystick position: all the way to -Y: LED3 turns on.

Joystick position: all the way to -X: LED4 turns on.

Joystick position: all the way to +X: LED2 turns on.

Joystick position: center: LED 1, 2, 3, and 4 should all be off.

Reflectance Sensor Behavior:

Case A (Black color detected): LED5 should be off.

Case B (White color detected): LED5 turns on.

This table shows the voltage values for the sensors

|  |  |
| --- | --- |
|  | Value |
| Voltage (X-axis) (Right/Center/Left) | 3.29/1.67/0 |
| Voltage (Y-axis) (Up/Center/Down) | 3.29/1.67/0 |
| Voltage (Case A – Black, Reflectance sensor) | 2.70 |
| Voltage (Case B – White, Reflectance sensor) | 0.19 |

**Main code that I want you to add these additions too:**

**#include** <msp430.h>

**void** **LCD\_command**(**unsigned** **char**);

**void** **LCD\_write**(**unsigned** **char**);

**void** **LCD\_init**(**void**);

**void** **display\_voltage**(**int**, **int**, **int**, **int**, **int**, **int**);

**unsigned** **int** adc\_raw[3];

**void** **main**(**void**) {

WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

PM5CTL0 &= ~LOCKLPM5; // Enable GPIO

// Configure ADC

ADC12CTL0 = ADC12SHT0\_6 | ADC12MSC | ADC12ON;

ADC12CTL1 = ADC12SHP | ADC12CONSEQ\_1;

ADC12CTL2 = ADC12RES\_2;

ADC12MCTL0 = ADC12INCH\_9;

ADC12MCTL1 = ADC12INCH\_10;

ADC12MCTL2 = ADC12INCH\_11 | ADC12EOS;

// Configure LCD pins

P3DIR |= 0xFF;

P3OUT &= ~0xFF;

P8DIR |= 0x0E;

P8OUT |= 0x0E;

LCD\_init();

**while** (1) {

ADC12CTL0 |= ADC12ENC | ADC12SC; // Start conversion

**while** ((ADC12IFGR0 & BIT2) == 0);

adc\_raw[0] = ADC12MEM0;

adc\_raw[1] = ADC12MEM1;

adc\_raw[2] = ADC12MEM2;

**int** v\_int[3], v\_dec[3];

**int** i;

**for** (i = 0; i < 3; i++) {

**float** voltage = (adc\_raw[i] / 4095.0) \* 3.3;

v\_int[i] = (**int**)voltage;

v\_dec[i] = (**int**)((voltage - v\_int[i]) \* 100);

}

display\_voltage(v\_int[0], v\_dec[0], v\_int[1], v\_dec[1], v\_int[2], v\_dec[2]);

**\_\_delay\_cycles**(500000);

}

}

**void** **display\_voltage**(**int** x\_int, **int** x\_dec, **int** y\_int, **int** y\_dec, **int** r\_int, **int** r\_dec) {

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

**\_\_delay\_cycles**(3000);

LCD\_command(0x80); // Set cursor to first line

LCD\_write('X'); LCD\_write(':');

LCD\_write('0' + x\_int); LCD\_write('.');

LCD\_write('0' + (x\_dec / 10)); LCD\_write('0' + (x\_dec % 10));

LCD\_write(' ');

LCD\_write('Y'); LCD\_write(':');

LCD\_write('0' + y\_int); LCD\_write('.');

LCD\_write('0' + (y\_dec / 10)); LCD\_write('0' + (y\_dec % 10));

LCD\_command(0xC0); // Move to second line

LCD\_write('R'); LCD\_write(':');

LCD\_write('0' + r\_int); LCD\_write('.');

LCD\_write('0' + (r\_dec / 10)); LCD\_write('0' + (r\_dec % 10));

}

**void** **LCD\_command**(**unsigned** **char** in) {

P3OUT = in;

P8OUT &= ~BIT3;

P8OUT &= ~BIT2;

P8OUT |= BIT1;

**\_\_delay\_cycles**(200);

P8OUT &= ~BIT1;

}

**void** **LCD\_write**(**unsigned** **char** in) {

P3OUT = in;

P8OUT |= BIT3;

P8OUT &= ~BIT2;

P8OUT |= BIT1;

**\_\_delay\_cycles**(200);

P8OUT &= ~BIT1;

}

**void** **LCD\_init**() {

P8OUT &= ~BIT1;

**\_\_delay\_cycles**(15000);

LCD\_command(0x30);

**\_\_delay\_cycles**(300);

LCD\_command(0x30);

**\_\_delay\_cycles**(300);

LCD\_command(0x30);

**\_\_delay\_cycles**(300);

LCD\_command(0x38);

LCD\_command(0x10);

LCD\_command(0x0F);

LCD\_command(0x06);

LCD\_command(0x01);

**\_\_delay\_cycles**(3000);

}

**Example of a program from a previous lab that uses the LEDs (That might help):**

#include <msp430.h>

#include <stdio.h>

// NAME : Blake Jackson, Kyle Rex

// Section : 501//

int main(void)

{

    WDTCTL = WDTPW | WDTHOLD;   // stop watchdog timer

    PM5CTL0 &=~LOCKLPM5;

    P3DIR |= 0xF;

    P3OUT |= 0xF;

    P8DIR |= 0xF;

    P8OUT |= 0xF;

    P6DIR &= ~BIT0;

    P6REN |= BIT0;

    P6OUT |= BIT0; // PULL UP

    while(1)

    {

       if ((P6IN & BIT0) == 0)

            {

               P3OUT &=~ 0X01;

               \_\_delay\_cycles(150000);

              P3OUT |=0X01;

              P3OUT &=~0X02;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X02;

              P3OUT &=~ 0X04;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X04;

              P3OUT &=~ 0X08;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X08;

              P8OUT &=~0X01;

              \_\_delay\_cycles(150000);

              P8OUT |=0X01;

              P8OUT &=~ 0X02;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X02;

              P8OUT &=~ 0X04;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X04;

              P8OUT &=~ 0X08;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X08;

              P8OUT &=~ 0X04;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X04;

              P8OUT &=~ 0X02;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X02;

              P8OUT &=~ 0X01;

              \_\_delay\_cycles(150000);

              P8OUT |= 0X01;

              P3OUT &=~ 0X08;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X08;

              P3OUT &=~ 0X04;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X04;

              P3OUT &=~ 0X02;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X02;

              P3OUT &=~ 0X01;

              \_\_delay\_cycles(150000);

              P3OUT |= 0X01;

               }

       }

    return 0;

}

**Connections:**

MSP430FR5994 Launchpad Connections

16x2 LCD screen Connections

P8.3 connects to RS

P8.2 connects to R/W

P8.1 connects to E

P3.0 connects to DB1

P3.1 connects to DB1

P3.2 connects to DB2

P3.3 connects to DB3

P3.4 connects to DB4

P3.5 connects to DB5

P3.6 connects to DB6

P3.7 connects to DB7

LED Connections

P6.0 connects to LED1

P6.1 connects to LED2

P7.0 connects to LED3

P7.1 connects to LED4

P6.2 connects to LED5

Reflectance Sensor Connections

G port of sensor connects to the GND port

V port of sensor connects to 3.3V port

2 port of sensor connects to P4.3

2 Axis Analog Joystick Connections

GND port of joystick connects to the GND port

VCC port of joystick connects to 3.3V port

Xout port of joystick connects to P4.1

Yout port of joystick connects to P4.2