**Main c Program.**

/\*

 \*  Project Owner   : TU Chemnitz

 \*  Project Name    : ASE Tutorial Unit-1

 \*  File Name       : main.c

 \*  Author          : ASE Admin

 \*  Created on      : 25 Feb 2024

 \*/

/\* Includes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "init.h"

#include "xpc56el.h"

#include "siu.h"

/\* Global variables \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int counter\_value = 0;

int counter\_mode = 1; // 1 = count up, 0 = count down

/\* Task-1 implementation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void task\_pot(int x){

    int thresholds[] = {682, 1364, 2046, 2728, 3410};

    int leds[] = {11, 6, 10, 9, 8, 7}; // U1, U2, Tx, Rx, P

    for (int i = 0; i < 6; i++) {

        SIU.GPDO[leds[i]].R = (x >= thresholds[i]) ? 1 : 0;

    }

}

/\* Task-2 implementation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void task\_ldr(int x){

    task\_pot(x); // Reuse same LED logic for light sensor values

}

/\* Task-3 implementation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void task\_counter(void){

    int u1 = (counter\_value >> 2) & 1;

    int u2 = (counter\_value >> 1) & 1;

    int u3 = counter\_value & 1;

    SIU.GPDO[11].R = u1; // U1 (D[11])

    SIU.GPDO[6].R = u2;  // U2 (C[11])

    SIU.GPDO[10].R = u3; // U3 (A[6])

}

int main(void) {

    peripheralsInit();

    PIT\_ConfigureTimer(1, 1000);

    PIT\_StartTimer(1);

    while(1){

        systemFunction();

        int sw1 = SIU.GPDI[4].R;

        int sw2 = SIU.GPDI[5].R;

        int sw3 = SIU.GPDI[6].R;

        int sw4 = SIU.GPDI[7].R;

        int bt1 = SIU.GPDI[12].R;

        int bt2 = SIU.GPDI[14].R;

        if (bt1) counter\_mode = 1;

        if (bt2) counter\_mode = 0;

        if (sw1) {

            int pot\_value = (ADC0.CDR[5].R & 0x00000FFF);

            task\_pot(pot\_value);

        } else if (!sw1 && sw2) {

            int ldr\_value = (ADC1.CDR[3].R & 0x00000FFF);

            task\_ldr(ldr\_value);

        } else if (!sw1 && !sw2 && sw3 && sw4) {

            task\_counter();

        } else {

            for (int i = 0; i < 6; i++) SIU.GPDO[7 + i].R = 0; // Turn off LEDs

        }

        osalThreadDelayMilliseconds(10UL);

    }

}

void PIT\_Channel\_1(void){

    if (counter\_mode) {

        counter\_value = (counter\_value + 1) % 8;

    } else {

        counter\_value = (counter\_value - 1 + 8) % 8;

    }

}

void PIT\_Channel\_2(void){}

void PIT\_Channel\_3(void){}

SIU.c program

/\*

Project Owner   : TU Chemnitz

Project Name    : ASE Tutorial Unit-2

File Name       : siu.c

Author          : ASE Admin

Created on      : 25 Feb 2024

\*/

/\* Includes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "siu.h"

#include "xpc56el.h"

/\*

@brief  SIU pin configuration function

@param  void

@retval void

\*/

void SIU\_Init(void)

{

// Configure all pins which are required for your task here

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/ Input pin configurations \*/

// Switches (SW1 - SW4 on pins D[4] to D[7])

SIU.PCR[4].R = 0x0100; // Input, no pull-up/down

SIU.PCR[5].R = 0x0100;

SIU.PCR[6].R = 0x0100;

SIU.PCR[7].R = 0x0100;

// Buttons (BT1 on pin D[12], BT2 on D[14])

SIU.PCR[12].R = 0x0100;

SIU.PCR[14].R = 0x0100;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/ Output pin configurations \*/

// LEDs

SIU.PCR[11].R = 0x0200; // U1

SIU.PCR[75].R = 0x0200; // U2

SIU.PCR[38].R = 0x0200; // U3

SIU.PCR[10].R = 0x0200; // Tx

SIU.PCR[9].R = 0x0200;  // Rx

SIU.PCR[8].R = 0x0200;  // P

SIU.PCR[7].R = 0x0200;  // Extra LED if needed

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/ Analog pin configurations \*/

// Potentiometer on ANP5 (ADC0 channel 5)

SIU.PCR[55].R = 0x2500; // Analog input

// Light sensor on ANP3 (ADC1 channel 3)

SIU.PCR[53].R = 0x2500; // Analog input

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

SIU.h program

/\*

 \*  Project Owner   : TU Chemnitz

 \*  Project Name    : ASE Tutorial Unit-2

 \*  File Name       : siu.h

 \*  Author          : ASE Admin

 \*  Created on      : 25 Feb 2024

 \*/

#ifndef SIU\_H\_

#define SIU\_H\_

/\* Function prototypes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void SIU\_Init(void);

#endif /\* SIU\_H\_ \*/