#include <lpc214x.h>

#include <stdio.h>

#define PLOCK 0x00000400

#define LED\_OFF (IO0SET = 1U << 31)

#define LED\_ON (IO0CLR = 1U << 31)

#define RS\_ON (IO0SET = 1U << 20)

#define RS\_OFF (IO0CLR = 1U << 20)

#define EN\_ON (IO1SET = 1U << 25)

#define EN\_OFF (IO1CLR = 1U << 25)

void SystemInit(void);

static void LCD\_SendCmdSignals(void);

static void LCD\_SendDataSignals(void);

static void LCD\_SendHigherNibble(unsigned char dataByte);

static void LCD\_CmdWrite( unsigned char cmdByte);

static void LCD\_DataWrite( unsigned char dataByte);

static void LCD\_Reset(void);

static void LCD\_Init(void);

void LCD\_DisplayString(const char \*ptr\_stringPointer\_u8);

unsigned int adc(int no,int ch);

void delay\_ms (int c);

int readSensor(int sen\_no);

static void delay\_us(unsigned int count);//microsecond delay

void runStepper(int direction);

int main(){

char buf[15];

unsigned int t,p;

SystemInit();

IO0DIR |= 1U << 31 | 0x00FF0000 ; // to set P0.16 to P0.23 as o/ps

IO1DIR |= 1U << 25; // to set P1.25 as o/p used for EN

//make D7 Led on off for testing

LED\_ON; delay\_ms(500);LED\_OFF;delay\_ms(500);

LCD\_Reset();

LCD\_Init();

delay\_ms(100);

while(1){

t = adc(1,4)/2;

p = readSensor(1);

if(p){

sprintf(buf,"T=%d",t);

if(t>37){

LCD\_CmdWrite(0x80); LCD\_DisplayString(buf);

LCD\_CmdWrite(0xc0); LCD\_DisplayString(" Access Denied");

delay\_ms(500);

}

else{

runStepper(0); //entry

delay\_ms(500);

runStepper(1);

LCD\_CmdWrite(0x80); LCD\_DisplayString(buf);

LCD\_CmdWrite(0xc0); LCD\_DisplayString(" Welcome");

delay\_ms(5000);

}

}

else

sprintf(buf," ");

//LCD\_CmdWrite(0x80); LCD\_DisplayString(buf);

delay\_ms(500);

LCD\_CmdWrite(0x80); LCD\_DisplayString(" ");

LCD\_CmdWrite(0xc0); LCD\_DisplayString(" ");

delay\_ms(1000);

}

}

unsigned int adc(int no,int ch)

{

// adc(1,4) for temp sensor LM34, digital value will increase as temp increases

// adc(1,3) for LDR - digival value will reduce as the light increases

// adc(1,2) for trimpot - digital value changes as the pot rotation

unsigned int val;

PINSEL0|= 0x0F300000; /\* Select the P0\_13 AD1.4 for ADC function \*/

/\* Select the P0\_12 AD1.3 for ADC function \*/

/\* Select the P0\_10 AD1.2 for ADC function \*/

switch(no) //select adc

{

case 0: AD0CR=0x00200600|(1<<ch); //select channel

AD0CR|=(1<<24); //start conversion

while((AD0GDR& (1U<<31))==0);

val=AD0GDR;

val=(val >> 6) & 0x03FF;

return val;

case 1: AD1CR=0x00200600|(1<<ch); //select channel

AD1CR|=(1<<24); //start conversion

while((AD1GDR&(1U<<31))==0);

val=AD1GDR;

val=(val >> 6) & 0x03FF;

return val;

}

// bit 6:15 is 10 bit AD value

return val;

}

void delay\_ms(int j){

unsigned int x,i;

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

{

for(x=0; x<10000; x++);

}

}

static void LCD\_CmdWrite( unsigned char cmdByte)

{

LCD\_SendHigherNibble(cmdByte);

LCD\_SendCmdSignals();

cmdByte = cmdByte << 4;

LCD\_SendHigherNibble(cmdByte);

LCD\_SendCmdSignals();

}

static void LCD\_DataWrite( unsigned char dataByte)

{

LCD\_SendHigherNibble(dataByte);

LCD\_SendDataSignals();

dataByte = dataByte << 4;

LCD\_SendHigherNibble(dataByte);

LCD\_SendDataSignals();

}

static void LCD\_Reset(void)

{

/\* LCD reset sequence for 4-bit mode\*/

LCD\_SendHigherNibble(0x30);

LCD\_SendCmdSignals();

delay\_ms(100);

LCD\_SendHigherNibble(0x30);

LCD\_SendCmdSignals();

delay\_us(200);

LCD\_SendHigherNibble(0x30);

LCD\_SendCmdSignals();

delay\_us(200);

LCD\_SendHigherNibble(0x20);

LCD\_SendCmdSignals();

delay\_us(200);

}

static void LCD\_SendHigherNibble(unsigned char dataByte)

{

//send the D7,6,5,D4(uppernibble) to P0.16 to P0.19

IO0CLR = 0X000F0000;IO0SET = ((dataByte >>4) & 0x0f) << 16;

}

static void LCD\_SendCmdSignals(void)

{

RS\_OFF;// RS - 1

EN\_ON;delay\_us(100);EN\_OFF;

delay\_us(100);// EN - 1 then 0

}

static void LCD\_SendDataSignals(void)

{

RS\_ON;// RS - 1

EN\_ON;delay\_us(100);EN\_OFF;

delay\_us(100);// EN - 1 then 0

}

static void LCD\_Init(void)

{

delay\_ms(100);

LCD\_Reset();

LCD\_CmdWrite(0x28u);//Initialize the LCD for 4-bit 5x7 matrix type

LCD\_CmdWrite(0x0Eu);// Display ON cursor ON

LCD\_CmdWrite(0x01u);//Clear the LCD

LCD\_CmdWrite(0x80u);//go to First line First Position

}

void LCD\_DisplayString(const char \*ptr\_string)

{

// Loop through the string and display char by char

while((\*ptr\_string)!=0)

LCD\_DataWrite(\*ptr\_string++);

}

void SystemInit(void)

{

PLL0CON = 0x01;

PLL0CFG = 0x24;

PLL0FEED = 0xAA;

PLL0FEED = 0x55;

while( !( PLL0STAT & PLOCK ))

{

;

}

PLL0CON = 0x03;

PLL0FEED = 0xAA;

PLL0FEED = 0x55;

}

int readSensor(int sen\_no) {

int result=0;

IO1DIR |= 1 << 24; IO1CLR = 1<< 24; // enable sensor logic: P1.24 - 0

switch (sen\_no) {

case 1: result = IO1PIN & (1<<22); //P1.22 connected to sensor1

break;

case 2: result = IO1PIN & (1<<23); //P1.23 connected to sensor2

break;

default: result = 0; }; IO1SET = 1<< 24; // disable sensor logic: P1.24

return result;

}

static void delay\_us(unsigned int count)

{

unsigned int j=0,i=0;

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

{

for(i=0;i<10;i++);

}

}

void runStepper(int direction){

int no\_of\_steps\_clk = 100;

if(direction ==1) //open

{

do{

IO0CLR = 0X000F0000;IO0SET = 0X00010000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00020000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00040000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00080000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

}while(1);

}

else

{

do{

IO0CLR = 0X000F0000;IO0SET = 0X00080000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00040000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00020000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

IO0CLR = 0X000F0000;IO0SET = 0X00010000;delay\_ms(10);if(--no\_of\_steps\_clk == 0) break;

}while(1);

}

}