#include <avr/io.h>

#include <util/delay.h>

#include <u8g2.h>

#include <u8x8\_avr.h>

#define SSD1306\_ADDR 0x78

u8g2\_t u8g2;

#define \_\_DELAY\_BACKWARD\_COMPATIBLE\_\_

#include <stdio.h>

#include <string.h>

int type = 8\*60;

*uint8\_t* minute ;

*uint8\_t* second ;

void updateMenu();

void UV\_light();

char res[100];

int main (void)

{

//Input

DDRC = 0x00; // PortC input (Place where 0 is input and 1 is output)

//pin 0 - select

//pin 1 - up (increase the working time)

//pin 2 - stop (force stop working)

//pin 3 - down (decrease the working time)

DDRD = 0xfb; // PortD pin1 input others are output 1111 1011

//pin 1 we giving the power (V cc)

//pin 3 UV lights are connected

//pin5 Green Led (Working indicator)

//pin6 Red Led (Off mode indicator)

//Output

DDRB = 0x05; // DDRC |= 0B00000101; pin0,pin2 are output and other input

//This is for buzzer

PORTC = 0xff; //all the input value of port c assign to 1

PORTD = 0x02; // pin0 assign to 1

u8g2\_Setup\_ssd1306\_i2c\_128x64\_noname\_f(&u8g2, U8G2\_R0, u8x8\_byte\_avr\_hw\_i2c, u8x8\_avr\_delay);

u8g2\_SetI2CAddress(&u8g2, SSD1306\_ADDR);

u8g2\_InitDisplay(&u8g2);

u8g2\_SetPowerSave(&u8g2, 0);

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_calibration\_gothic\_nbp\_tr);

u8g2\_SetFontRefHeightText(&u8g2);

u8g2\_SetFontPosTop(&u8g2);

u8g2\_DrawStr(&u8g2, 40, 20, "Sensitizing");

u8g2\_DrawStr(&u8g2, 20, 40, "Using UV Rays");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(2000);

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_SetFontRefHeightText(&u8g2);

u8g2\_SetFontPosTop(&u8g2);

u8g2\_DrawStr(&u8g2, 10, 15, "Designed by");

u8g2\_DrawStr(&u8g2, 10, 40, "Group 14");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(2000);

while(1){

if((PIND & (1<<PIND2)))

{

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_SetFontRefHeightText(&u8g2);

u8g2\_SetFontPosTop(&u8g2);

u8g2\_DrawStr(&u8g2, 0, 0, "Press Select to");

u8g2\_DrawStr(&u8g2, 0, 30, "Sensitizing");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(2000);

}

else

{

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_SetFontRefHeightText(&u8g2);

u8g2\_SetFontPosTop(&u8g2);

u8g2\_DrawStr(&u8g2, 0, 0, "Press On");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(2000);

}

// If the power button is on red led should blink all others off

if((PIND & (1<<PIND2)))

{

PORTD |= (1<<PORTD6); // make portD pin6 high

}

//If power is not on all the system is off

else

{

PORTD &= ~(1<<PORTD6);

}

//If power button on or off Green led and UV are initially of

PORTD &= ~(1<<PORTD3); // make portD pin3 low

PORTD &= ~(1<<PORTD5); // make portD pin5 low

//If Select button is pressed then Red Led should off and UV should on

if(~PINC & 0x01) //if portC pin0 is low

{

PORTD &= ~(1<<PORTD6); //portD pin6 is low

UV\_light();

}

//If up button pressed increase the working time by 120 units

if(~PINC& 0x02) //if portC pin1 is low

{

*\_delay\_ms*(500);

type = type + 120;

if(type > 60\*10){type=60\*10;}

updateMenu();

}

//If down button is pressed decreasing the working time by 120 units

if(~PINC& 0x08) //if portC pin4 is low

{

*\_delay\_ms*(500);

type = type - 120;

if(type < 60\*6){type=60\*6;}

updateMenu();

}

updateMenu();

}

}

// when we want to turn on led we want to call this function

void UV\_light(){

int temp = 0;

while(temp <= type)

{

int i = 0;

while (i<10){

//If power is not supply then UV and Green Light can not be light up

if ((~PIND & (1<<PIND2)))

{

PORTD &= ~(1<<PORTD3);

PORTD &= ~(1<<PORTD5);

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_calibration\_gothic\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 30, 30, "Caution..!");

u8g2\_SendBuffer(&u8g2);

// LCD ............

}

//If stop button push then forced stop

else if(~PINC& (1<<PINC2))

{

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_calibration\_gothic\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 40, 20, "Force ");

u8g2\_DrawStr(&u8g2, 30, 35, "Stopped");

u8g2\_SendBuffer(&u8g2);

// LCD ............

*\_delay\_ms*(100);

return;

}

//If there no any other problem work UV and Green Led

else

{

PORTD |= (1<<PORTD3);

PORTD |= (1<<PORTD5);

*\_delay\_ms*(100);

i=i+1;

}

}

temp = temp+1;

minute = (type-temp)/60;

second = (type-temp)%60;

if (second<10){*sprintf*(res, "0%d : 0%d", minute,second);}

else{*sprintf*(res, "0%d : %d", minute,second);}

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 30, 5, "\*\*\*\*\*\*\*\*\*\*");

u8g2\_DrawStr(&u8g2, 30, 20, "Time Left");

u8g2\_DrawStr(&u8g2, 40, 40, res);

u8g2\_SendBuffer(&u8g2);

// LCD ............

}

PORTD &= ~(1<<PORTD3);

PORTD &= ~(1<<PORTD5);

if (type+1==temp){

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 50, 20, "O");

u8g2\_DrawStr(&u8g2, 40, 35, "/ | )");

u8g2\_DrawStr(&u8g2, 41, 45, "/ |..........");

u8g2\_SendBuffer(&u8g2);

// LCD ............

}

return;

}

void updateMenu(){

switch(type){

case 6\*60:

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 30, 20, "Small Items");

u8g2\_DrawStr(&u8g2, 30, 40, "6 Minutes");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(200);

// LCD ............

break;

case 8\*60: //Third display state

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 20, 20, "Medium Items");

u8g2\_DrawStr(&u8g2, 30, 40, "8 Minutes");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(200);

// LCD ............

break;

case 10\*60:

// Sending information to LCD to print

u8g2\_ClearBuffer(&u8g2);

u8g2\_SetFont(&u8g2, u8g2\_font\_smart\_patrol\_nbp\_tr);

u8g2\_DrawStr(&u8g2, 20, 20, "Large Items");

u8g2\_DrawStr(&u8g2, 30, 40, "10 Minutes");

u8g2\_SendBuffer(&u8g2);

*\_delay\_ms*(200);

// LCD ............;

break;

}

return;

}