

```

/*
 * GccApplication21.c
 *
 * Created: 08/06/2024 18:39:57
 * Author : mattia
 */

#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdbool.h>

#define B2 (1<<PIND2)
#define B3 (1<<PIND3)
#define B4 (1<<PIND4)
#define B5 (1<<PIND5)
#define B6 (1<<PIND6)
#define B7 (1<<PIND7)

#define L0 (1<<PINC0)
#define L1 (1<<PINC1)
#define L2 (1<<PINC2)
#define L3 (1<<PINC3)
#define L4 (1<<PINC4)
#define L5 (1<<PINC5)

#define Leds (L0 | L1 | L2 | L3 | L4 | L5)
#define Buttons (B2 | B3 | B4 | B5 | B6 | B7)

volatile uint8_t press;
volatile uint8_t oldval = 0xff;
volatile uint8_t pw[] = {B2, B7, B7, B3, B2};
volatile uint8_t guess[] = {0, 0, 0, 0, 0};
volatile int tick = 0;
volatile int nopress = 0;
volatile int tick2 = 0;
volatile int currentPos = 0;

typedef enum{
    PASSWORD,
    OPENING,
    OPENED,
    CLOSING,
    RESET
}States;
States currentState = PASSWORD;

void displayLed();
void blink();
void stop();
bool passCheck();

int main(void)
{
    DDRC |= Leds;
    PORTC &=~ Leds;

    DDRD &=~ Buttons;
    PORTD |= Buttons;

    PCICR |= (1<<PCIE2);
    PCMSK2 |= Buttons;

    TCCR0A |= (1<<WGM01);
    TIMSK0 |= (1<<OCIE0A);
    OCR0A = 79; //5ms

```

```

TCCR1B |= (1<<WGM12);
TIMSK1 |= (1<<OCIE1A);
OCR1A = 15626; //1s

TCCR2A |= (1<<WGM21);
TIMSK2 |= (1<<OCIE2A);
OCR2A = 157; //10ms

TCCR1B |= ((1<<CS10) | (1<<CS12));

sei();

while (1)
{
    switch(currentState){
        case PASSWORD:
            if(press){
                guess[currentPos] = press;
                displayLed();
                currentPos++;
                press = 0;
                nopress = 0;
            }

            if(currentPos == 5){
                if(passCheck()){
                    currentState = OPENING;
                    TCCR0B |= ((1<<CS00) | (1<<CS02));
                    TCCR2B |= ((1<<CS20) | (1<<CS21) | (1<<CS22));
                }
                else{
                    currentState = RESET;
                }
            }

            if(nopress == 4){
                currentState = RESET;
            }
            break;

        case OPENING:
            blink();
            stop();
            break;

        case OPENED:
            blink();
            stop();
            break;

        case CLOSING:
            blink();
            break;

        case RESET:
            TCCR0B &=~ ((1<<CS00) | (1<<CS02));
            TCCR2B &=~ ((1<<CS20) | (1<<CS21) | (1<<CS22));
            PORTC &=~ Leds;
            press = 0;
            currentPos = 0;
            nopress = 0;
            tick = 0;
            tick2 = 0;
            currentState = PASSWORD;
    }
}

```

```

    }
}

ISR(PCINT2_vect){
    uint8_t change = oldval ^ PIND;
    oldval = PIND;
    bool hold = false;
    for(uint8_t i = PIND2; i <= PIND7; i++){
        if((change & (1<<i)) && !(PIND & (1<<i))){
            press = (1<<i);
            hold = true;
        }
    }
    if(!(hold)){
        press = 0;
    }
}

ISR(TIMER0_COMPA_vect){
    tick++;
}

ISR(TIMER1_COMPA_vect){
    nopress++;
}

ISR(TIMER2_COMPA_vect){
    tick2++;
}

void displayLed(){
    switch(currentPos){
        case 4:
            PORTC |= L4;
            break;

        case 3:
            PORTC |= L3;
            break;

        case 2:
            PORTC |= L2;
            break;

        case 1:
            PORTC |= L1;
            break;

        default:
            PORTC |= L0;
            break;
    }
}

void stop(){
    if(press == B5){
        currentState = CLOSING;
        tick = 0;
        tick2 = 0;
        press = 0;
    }
}

void blink(){

```

```

int period = 100;
int cycle;
int maxtime;
States nextState;
bool blinkstate = true;

switch(currentState){
    case OPENING:
        cycle = 25;
        maxtime = 400;
        nextState = OPENED;
        break;

    case OPENED:
        blinkstate = false;
        maxtime = 300;
        PORTC |= L5;
        nextState = CLOSING;
        break;

    case CLOSING:
        cycle = 75;
        maxtime = 350;
        nextState = RESET;
        break;
}

if(blinkstate){
    if(tick <= cycle){
        PORTC |= L5;
    }
    else{
        PORTC &=~ L5;
    }

    if(tick >= period){
        tick = 0;
    }
}

if(tick2 == maxtime){
    currentState = nextState;
    tick2 = 0;
    tick = 0;
}
}

bool passCheck(){
    for(int i = 0; i<5; i++){
        if(guess[i] != pw[i]){
            return false;
        }
    }
    return true;
}

```

lampeggia l5 solo nel caso di apertura e chiusura

quando arriva alla fine del tempo passa allo stato successivo