```c
/*
 * GccApplication28.c
 *
 * Created: 13/06/2024 14:50:49
 * Author : mattia
 */

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

#define B2 (1<<PIND2)
#define B3 (1<<PIND3)
#define B7 (1<<PIND7)

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

#define Leds (L0 | L2 | L3 | L4 | L5)
#define Buttons (B2 | B3 | B7)

typedef enum{
	WAIT,
	MIN,
	MED,
	MAX,
	RESET
	}States;
States currentState = WAIT;

volatile uint8_t press;
volatile uint8_t oldval = 0xff;
volatile int tick = 0;
volatile int seconds = 0;
volatile int presstimer = 0;
volatile int pressArray[] = {0, 0};
volatile bool buttonRegister[] = {false, false};
volatile bool pressedb2 = false;
volatile bool pressedb3 = false;

void pressFun();
void pressionStateChange();
void blink(int period);
void stop();

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 = 79; //5ms

        TCCR0B |= ((1<<CS00) | (1<<CS02));
        TCCR1B |= ((1<<CS10) | (1<<CS12));
        TCCR2B |= ((1<<CS20) | (1<<CS21) | (1<<CS22));

        sei();

    while (1)
    {
            switch(currentState){
                case WAIT:
                    tick = 0;
                    seconds = 0;
                    pressFun();
                    pressionStateChange();
                    break;

                case MIN:
                    blink(200);                    1000 / 5   -> periodo
                    pressFun();
                    pressionStateChange();
                    stop();
                    break;

                case MED:
                    blink(150);                    750 / 5
                    pressFun();
                    pressionStateChange();
                    stop();
                    break;

                case MAX:
                    blink(100);                    500 / 5
                    pressFun();
                    pressionStateChange();
                    stop();
                    break;

                case RESET:
                    PORTC &=~ Leds;
                    for(int i = 0; i<2; i++){
                        buttonRegister[i]=false;
                        pressArray[i]=0;
                    }
                    presstimer = 0;
                    seconds = 0;
                    tick = 0;
                    press = 0;
                    currentState = WAIT;
                    pressedb2 = false;
                    pressedb3 = false;
                    break;
            }
    }
}

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){
	seconds++;
}

ISR(TIMER2_COMPA_vect){
	presstimer++;
}

void pressFun(){
	switch(press){
		case B2:
			if(!(buttonRegister[0])){
				pressedb2 = true;
				pressArray[0] = presstimer;
				PORTC |= L0;
			}
			break;

		case B3:
			if(!(buttonRegister[1])){
				pressedb3 = true;
				pressArray[1] = presstimer;
				PORTC |= L2;
			}
			break;

		default:
			if(pressedb2){
				pressedb2 = false;
				buttonRegister[0] = true;
			}
			if(pressedb3){
				pressedb3 = false;
				buttonRegister[1] = true;
			}
			presstimer = 0;
			break;
	}
}

void pressionStateChange(){
	States nextState = RESET;

	if(buttonRegister[0] && buttonRegister[1]){
		if(pressArray[0] < 400){
			if(pressArray[1] >= 400){
				nextState = MED;
			}
		}
		else if(pressArray[1] < 400){
```

se b2 minore di 2 e b3 maggiore di 2 -> medio

if(pressArray[0] > 400){ MANCA QUESTA RIGA

```
                nextState = MIN;
            }
            else{
                nextState = MAX;
            }

            for(int i = 0; i<2; i++){
                buttonRegister[i]=false;
                pressArray[i]=0;
                PORTC &=~ (L0 | L2);
            }
            tick = 0;
            seconds = 0;
            currentState = nextState;
        }
}

void blink(int period){
        int cycle = period/5;
        if(tick <= cycle){
            PORTC |= L3;
            PORTC &=~ (L4 | L5);
        }
        else if(tick <= 2*cycle){
            PORTC &=~ (L3 | L4 | L5);
        }
        else if(tick <= 3*cycle){
            PORTC |= L4;
            PORTC &=~ (L3 | L5);
        }
        else if(tick <= 4*cycle){
            PORTC &=~ (L3 | L4 | L5);
        }
        else{
            PORTC |= L5;
            PORTC &=~ (L3 | L4);
        }

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

void stop(){
        if((seconds == 8) || (press == B7)){
            currentState = RESET;
        }
}
```