```c
/*
 * GccApplication26.c
 *
 * Created: 13/06/2024 09:19:08
 * Author : mattia
 */

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

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

#define L1 (1<<PINC1)
#define L2 (1<<PINC2)
#define L3 (1<<PINC3)
#define L4 (1<<PINC4)

#define Leds (L1 | L2 | L3 | L4)
#define Buttons (B2 | B3 | B7)

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

volatile uint8_t press;
volatile uint8_t oldval = 0xff;
volatile int tick = 0;
volatile int seconds = 0;
volatile int modes = 0;

void lightLeds(int period, int cycle, uint8_t led1, uint8_t led2, uint8_t led3);
void changeMode();
void stop();
void modeState();

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

	DDRD &=~ Buttons;
	PORTD |= Buttons;

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

	TCCR0A |= (1<<WGM01);
	TIMSK0 |= (1<<OCIE0A);
	OCR0A = 17; //1ms

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

	sei();

    while (1)
```

```
    {
            switch(currentState){
                  case OFF:
                        changeMode();
                        break;

                  case MIN:
                        lightLeds(20, 5, L1, 0, 0);
                        changeMode();
                        stop();
                        break;

                  case MED:
                        lightLeds(20, 5, L1, L2, 0);
                        changeMode();
                        stop();
                        break;

                  case HIGH:
                        lightLeds(20, 5, L1, L2, L3);
                        changeMode();
                        stop();
                        break;

                  case MAX:
                        PORTC |= L4;
                        lightLeds(20, 5, L1, L2, L3);
                        changeMode();
                        stop();
                        break;

                  case RESET:
                        TCCR0B &=~ (1<<CS00) | (1<<CS02);
                        TCCR1B &=~ (1<<CS10) | (1<<CS12);
                        PORTC &=~ Leds;
                        tick = 0;
                        seconds = 0;
                        press = 0;
                        modes = 0;
                        currentState = OFF;
            }
    }
}

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

ISR(TIMER0_COMPA_vect){
      tick++;
}

ISR(TIMER1_COMPA_vect){
      seconds++;
}

void lightLeds(int period, int cycle, uint8_t led1, uint8_t led2, uint8_t led3){
      if(tick <= cycle){
            PORTC |= led1;
```

```
        }
        else{
                PORTC &=~ led1;
        }
        if(tick == period){
                tick = 0;
        }

        if(tick <= 2*cycle){
                PORTC |= led2;
        }
        else{
                PORTC &=~ led2;
        }
        if(tick == period){
                tick = 0;
        }

        if(tick <= 3*cycle){
                PORTC |= led3;
        }
        else{
                PORTC &=~ led3;
        }
        if(tick == period){
                tick = 0;
        }
}

void changeMode(){
        switch(press){
                case B2:
                        if(modes > 0){
                                modes --;
                                modeState();
                        }
                        press = 0;
                        break;

                case B3:
                        if(modes < 4){
                                modes ++;
                                modeState();
                        }
                        press = 0;
                        break;
        }
}

void modeState(){
        switch(modes){
                case 4:
                        TCCR0B |= (1<<CS00) | (1<<CS02);
                        TCCR1B |= (1<<CS10) | (1<<CS12);
                        PORTC &=~ Leds;
                        tick = 0;
                        seconds = 0;
                        currentState = MAX;
                        break;

                case 3:
                        TCCR0B |= (1<<CS00) | (1<<CS02);
                        TCCR1B |= (1<<CS10) | (1<<CS12);
                        PORTC &=~ Leds;
                        tick = 0;
```

```
                seconds = 0;
                currentState = HIGH;
                break;

        case 2:
                TCCR0B |= (1<<CS00) | (1<<CS02);
                TCCR1B |= (1<<CS10) | (1<<CS12);
                PORTC &=~ Leds;
                tick = 0;
                seconds = 0;
                currentState = MED;
                break;

        case 1:
                TCCR0B |= (1<<CS00) | (1<<CS02);
                TCCR1B |= (1<<CS10) | (1<<CS12);
                PORTC &=~ Leds;
                tick = 0;
                seconds = 0;
                currentState = MIN;
                break;

        default:
                TCCR0B &=~ (1<<CS00) | (1<<CS02);
                TCCR1B &=~ (1<<CS10) | (1<<CS12);
                PORTC &=~ Leds;
                tick = 0;
                seconds = 0;
                currentState = OFF;
                break;
    }
}

void stop(){
    if((press == B7) || (seconds == 10)){
        currentState = RESET;
    }
}
```