

```

/*
 * impastatrice.c
 *
 * Created: 11/06/2024 15:53:37
 * Author : Christian
 */

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

#define l1 (1 << PINC1)
#define l2 (1 << PINC2)
#define l3 (1 << PINC3)
#define l5 (1 << PINC5)

#define b2 (1 << PIND2)
#define b4 (1 << PIND4)
#define b5 (1 << PIND5)
#define b7 (1 << PIND7)

typedef enum {
    riposo,
    rotazione,
    reset
} stato;
stato currentstate = riposo;

volatile uint8_t pulsantepremuto = 0;
volatile uint8_t oldvalue = 0xff;
volatile int velocita[] = {900, 750, 600, 300};
volatile int velocitacorrente = 0;
volatile int timer = 0;
volatile bool direzione = false;
volatile int tick = 0;

void inverti();
void led();

int main(void)
{
    DDRC |= ( l1 | l2 | l3 | l5);
    PORTC &=~ ( l1 | l2 | l3 | l5);

    DDRD &=~ (b2 | b4 | b5 | b7);
    PORTD |= (b2 | b4 | b5 | b7);

    PCICR |= (1 << PCIE2);
    PCMSK2 |= (b2 | b4 | b5 | b7);

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

    TCCR1B |= (1 << WGM12);
    TIMSK1 |= (1 << OCIE1A);
    OCR1A = 15626; //5ms

    sei();

    while (1)
    {
        switch (currentstate){

            case riposo:

```

```

        PORTC |= 15;
        if((pulsantepremuto == b4) || (pulsantepremuto == b5)){
            currentstate = rotazione;
        }
        break;

    case rotazione:
        TCCR0B |= ((1 << CS00) | (1 << CS02));
        TCCR1B |= ((1 << CS10) | (1 << CS12));
        led();
        if (pulsantepremuto == b4){
            if (timer >= 1){
                if (velocitacorrente < 3){
                    velocitacorrente++;
                }
                timer = 0; // Reset debounce timer
            }
            pulsanteremuto = 0;
        }

        else if (pulsantepremuto == b5)
        {
            if (timer >= 1){
                if (velocitacorrente > 0) // 0 is the min index for
                {
                    velocitacorrente--;
                }
                timer = 0; // Reset debounce timer
            }
            pulsanteremuto = 0;
        }

        if (pulsantepremuto == b7)
        {
            if (timer >= 1)
            {
                inverti();
                timer = 0;
            }
            pulsanteremuto = 0;
        }

        if (pulsantepremuto == b2)
        {
            currentstate = reset;
        }
        break;

    case reset:
        TCCR0B &=~ ((1 << CS00) | (1 << CS02));
        TCCR1B &=~ ((1 << CS10) | (1 << CS12));
        PORTC &=~ ( 11 | 12 | 13 | 15);
        pulsanteremuto = 0;
        tick = 0;
        timer = 0;
        currentstate = riposo;
        break;

}
}
}

```

```

void inverti(){
    direzione = !direzione;
}

void led(){
    int period = velocita[velocitacorrente];
    int cycle = period/3;
    if(direzione){
        if(tick <= cycle){
            PORTC |= 13;
            PORTC &=~ (12 | 11);
        }
        else if(tick <= 2*cycle){
            PORTC |= 12;
            PORTC &=~ (11 | 13);
        }
        else{
            PORTC |= 11;
            PORTC &=~ (12 | 13);
        }
        if(tick >= period){
            tick = 0;
        }
    }
    else{
        if(tick <= cycle){
            PORTC |= 11;
            PORTC &=~ (12 | 13);
        }
        else if(tick <= 2*cycle){
            PORTC |= 12;
            PORTC &=~ (11 | 13);
        }
        else{
            PORTC |= 13;
            PORTC &=~ (12 | 11);
        }
        if(tick >= period){
            tick = 0;
        }
    }
}

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

ISR (TIMER0_COMPA_vect){
    tick++;
}

ISR (TIMER1_COMPA_vect){
    timer++;
    secondi = timer
}

```