```c
/*Simulazione tastierino numerico in modo che soltanto all'inserimento della password B3 B5
B4 B5 si abbia l'apertura della porta,
simulata dall'accensione del led L5. l'inserimento delle cifre deve essere segnalato
dall'accensione dei led L0 L1 L2 L3 in sequenza.
Alla fine dell'inserimento delle cifre se la password è corretta dovranno lampeggiare 3
volte (500 ms acceso, 200ms spento, 500ms acceso, 200ms spento e 500ms acceso).
Premendo B7 due volte si richiuderà la porta senza inserimento della password.
*/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdbool.h>

#define l0 (1<<PINC0)
#define l1 (1<<PINC1)
#define l2 (1<<PINC2)
#define l3 (1<<PINC3)
#define l5 (1<<PINC5)

#define b3 (1<<PIND3)
#define b4 (1<<PIND4)
#define b5 (1<<PIND5)
#define b7 (1<<PIND7)

volatile uint8_t pulsantepremuto = 0;
volatile uint8_t oldval = 0xff;
volatile uint8_t password[] = {b3, b5, b4, b5};
volatile uint8_t guess[] = {0,0,0,0};
volatile uint8_t b7presscount = 0;

volatile int currentpos = 0;
volatile int tick=0;
volatile int timer=0;
volatile int loopcount=0;

typedef enum {
	inserimentopassword,
	avvio, // controllo password  e lampeggio
	reset
} states;
states currentstate = inserimentopassword;

void flash_leds();
void check_password();

int main(void)
{
	DDRC |= (l1 | l2 | l3 | l0 | l5);
	PORTC &=~ (l1 | l2 | l3 | l0 | l5);

	DDRD &=~ (b3 | b4 | b5 | b7);
	PORTD |=(b3 | b4 | b5 | b7);

	PCICR |= (1 << PCIE2);
	PCMSK2 |= (b3 | b4 | b5 | b7);

	TCCR1B |= (1 << WGM12);
	TIMSK1 |= (1 << OCIE1A);
	OCR1A = 1564; //100 ms

	sei();

	while (1)
	{
```

```
                switch (currentstate){
                        case inserimentopassword:
                        if (pulsantepremuto !=0)
                        {
                                guess[currentpos]= pulsantepremuto;
                                switch(currentpos){
                                        default:
                                                PORTC |= l0;
                                                break;
                                        case 3:
                                                PORTC|=l3;
                                                break;
                                        case 2:
                                                PORTC |=l2;
                                                break;
                                        case 1:
                                                PORTC|=l1;
                                                break;
                                       }
                              currentpos ++;
                              pulsantepremuto = 0;
                        }
                        if (currentpos == 4) {
                                check_password();
                        }
                        break;

                        case avvio:
                        TCCR1B|= (1<< CS10) | (1 << CS12);
                        flash_leds();
                        PORTC |= l5; // Open the door
                        if (pulsantepremuto == b7)
                        {
                                b7presscount++;
                                if (b7presscount == 2) {
                                        currentstate = reset;
                                }
                                pulsantepremuto = 0;
                        }
                        break;

                        case reset:
                        TCCR1B &=~ ((1<<CS10) | (1<<CS12));
                        PORTC &=~ (l0|l1|l2|l3|l5);
                        currentpos = 0;
                        tick = 0;
                        b7presscount = 0;
                        loopcount = 0;
                        pulsantepremuto = 0;
                        currentstate = inserimentopassword;
                        break;

                }
        }
 }

void check_password() {
        bool errorcheck = false;
         for (uint8_t i = 0; i < 4; i++) {
                if (guess[i] != password[i]) {
                        currentstate = reset;
                        errorcheck = true;
                }
         }
         if(!(errorcheck)){
```

```
                currentstate = avvio;
                tick = 0;
        }
}

void flash_leds() {
        if(tick < 5 && loopcount < 3){
                PORTC |= (l0 | l1 | l2 | l3);
        }
        else{
                PORTC &=~ (l0 | l1 | l2 | l3);
        }

        if(tick == 7){
                tick = 0;
                loopcount++;
        }
}

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))){
                        pulsantepremuto = (1<<i);
                }
        }
}

ISR(TIMER1_COMPA_vect) {
        tick++;
}
```

&& -> AND
|| -> OR

500 / 100 = 5

7 = periodo -> 5 acceso + 2 spento (quanto tempo ci impiega in totale