```c
/*
 * GccApplication27.c
 *
 * Created: 13/06/2024 10:19:10
 * Author : mattia
 */

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

typedef enum{
	SELECTION,
	PAYMENT,
	START,
	RESET
	}States;
States currentState = SELECTION;

typedef enum{
	WATER,
	WASHING,
	CENTRIFUGATION,
	_
	}WashingStates;
WashingStates currentWashing = WATER;

typedef enum{
	HALF,
	FULL
	}Load;
Load currentLoad;

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

void paying();
void showAddedCash();
void blink(int period, int cycle, uint8_t led);

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

    sei();

while (1)
{
          switch(currentState){
                case SELECTION:
                      switch(press){
                            case B2:
                                  press = 0;
                                  currentLoad = HALF;
                                  currentState = PAYMENT;
                                  break;

                            case B3:
                                  press = 0;
                                  currentLoad = FULL;
                                  currentState = PAYMENT;
                                  break;
                      }
                      break;

                case PAYMENT:
                      paying();
                      showAddedCash();
                      break;

                case START:
                      switch(currentWashing){
                            case WATER:
                                  blink(40, 20, L4);
                                  break;

                            case WASHING:
                                  blink(100, 25, L5);
                                  break;

                            case CENTRIFUGATION:
                                  blink(100, 75, L5);
                                  break;
                      }
                      if(press == B4){
                            currentState = RESET;
                      }
                      break;

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

200 / 5 = 40
40 / 2 = 20

500 / 5 = 100
25% di 100 = 25

```
                        cashadded = 0;
                        currentWashing = WATER;
                        currentState = SELECTION;
                        break;
                }
        }
}

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 paying(){
        int total;

        switch(currentLoad){
                case HALF:
                        total = 5;
                        break;

                case FULL:
                        total = 7;
                        break;
        }

        switch(press){
                case B5:
                        cashadded += 4;
                        if(cashadded > total){
                                cashadded -= 4;
                        }
                        press = 0;
                        break;

                case B6:
                        cashadded += 2;
                        if(cashadded > total){
                                cashadded -= 2;
                        }
                        press = 0;
                        break;

                case B7:
                        cashadded += 1;
                        if(cashadded > total){
                                cashadded -= 1;
                        }
                        press = 0;
                        break;
        }
```

5 monete da 50 centesimi per arrivare a 2.50

```
        if(cashadded == total){
                PORTC |= L3;
                switch(currentLoad){
                        case HALF:
                                if(press == B2){
                                        press = 0;
                                        currentWashing = WATER;
                                        TCCR0B |= ((1<<CS00) | (1<<CS02));
                                        TCCR1B |= ((1<<CS10) | (1<<CS12));
                                        currentState = START;
                                }
                                break;

                        case FULL:
                                if(press == B3){
                                        press = 0;
                                        currentWashing = WATER;
                                        TCCR0B |= ((1<<CS00) | (1<<CS02));
                                        TCCR1B |= ((1<<CS10) | (1<<CS12));
                                        currentState = START;
                                }
                                break;
                }
        }
}

void showAddedCash(){
        switch(cashadded){
                case 7:
                        PORTC |= (L2 | L1 | L0);
                        break;

                case 6:
                        PORTC &=~ L0;
                        PORTC |= (L2 | L1);
                        break;

                case 5:
                        PORTC &=~ L1;
                        PORTC |= (L2 | L0);
                        break;

                case 4:
                        PORTC &=~ (L1 | L0);
                        PORTC |= L2;
                        break;

                case 3:
                        PORTC &=~ L2;
                        PORTC |= (L0 | L1);
                        break;

                case 2:
                        PORTC &=~ (L0 | L2);
                        PORTC |= L1;
                        break;

                case 1:
                        PORTC &=~ (L1 | L2);
                        PORTC |= L0;
                        break;

                default:
                        PORTC &=~ (L0 | L1 | L2);
                        break;
```

Ogni volta che viene aggiunta una moneta da 50 si accende un led

```
	}
}

void blink(int period, int cycle, uint8_t led){
	WashingStates nextWashingState;
	int maxseconds;

	switch(currentWashing){
		case WATER:
			nextWashingState = WASHING;
			break;

		case WASHING:
			nextWashingState = CENTRIFUGATION;
			break;

		case CENTRIFUGATION:
			nextWashingState = _;
			break;
	}

	switch(currentLoad){
		case HALF:
			switch(currentWashing){
				case WATER:
					maxseconds = 2;
					break;

				case WASHING:
					maxseconds = 4;
					break;

				case CENTRIFUGATION:
					maxseconds = 2;
					break;
			}
			break;

		case FULL:
			switch(currentWashing){
				case WATER:
					maxseconds = 3;
					break;

				case WASHING:
					maxseconds = 6;
					break;

				case CENTRIFUGATION:
					maxseconds = 3;
					break;
			}
			break;
	}



	if(tick <= cycle){
		PORTC |= led;
	}
	else{
		PORTC &=~ led;
	}
	if(tick >= period){
		tick = 0;
```

```
        }

        if(seconds == maxseconds){
                if(nextWashingState != _){
                        PORTC &=~ led;
                        seconds = 0;
                        currentWashing = nextWashingState;
                }
                else{
                        PORTC &=~ led;
                        currentState = RESET;
                }
        }
}
```