```c
/*
所属:3EP1-07
学籍番号:1614266
名前:大谷直也
作成日:2018/11/17
*/

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

#define user_CTOP 50000UL
#define CTOP 10000UL

void update_led();
/*回路全体図*/
volatile unsigned char map[8] =
{
 //    0b10111111,
      0b10010000,
         0b10011111,
      0b11000001,
      0b11111101,
      0b10000001,
      0b10111111,
      0b10000001,
      0b11111101
};
/*swicth操作用*/
volatile unsigned char stat;
volatile unsigned char sw;
volatile unsigned char sw_flag;

/*ブザーの時間*/
volatile unsigned int period;

/*位置などの記録*/
static unsigned char scan = 0;
unsigned char my_state = 0;

/*プレイヤー操作変数*/
unsigned char x = 0x40;
/*プレイヤー点滅用*/
unsigned char x_sub = 0;
/*迷路の見える範囲*/
unsigned char smog_b = 0xE0;
 /*switch用*/
bool sw_flag2 = false;
/*点滅中の値の変更防止*/
bool x_flag = false;

ISR(PCINT1_vect) /*スイッチ*/
{
      stat = 1;
}

ISR(TIMER0_COMPA_vect) /*ledアップデート*/
{
      update_led();
}

ISR(TIMER2_COMPA_vect) /*ブザー*/
{
      PORTD  ^= _BV(PORTD3);
      if(sw_flag2 == true){
            if(period != 0){
                  period --;
                  if(period == 0){
                        sw_flag2 = false;
                        DDRD = 0xF6;
                  }
            }
      }
      else{
            OCR2A = 0;
      }

}

void update_sw() /*switchフラグ管理*/
{
      static unsigned long cnt;
      switch (stat) {
      case 0:
            return;
      case 1:
            cnt = CTOP;
            stat = 2;
            return;
      case 2:
            cnt--;
            if (cnt == 0) {
                  sw = ~(PINC >> 4) & 3;
                  sw_flag = 1;
                  stat = 0;
            }
            return;
      }
}

void update_led() /*マトリクスLEDのアップデート*/
{
      static unsigned char sc = 0xFE;

      PORTB = 0;
      sc = (sc << 1) | (sc >> 7);
      PORTD = (PORTD & 0x0F) | (sc & 0xF0);
      PORTC = (PORTC & 0xF0) | (sc & 0x0F);
      scan = (scan + 1) & 7;

/*霧の発生*/
      if(my_state != 0){
            if(scan == my_state ){
                  PORTB = map[scan] & smog_b;
                  PORTB |= x;
            }else if(scan == (my_state + 1)){
                        PORTB = map[scan] & smog_b;
            }else if(scan == (my_state - 1)){
                  PORTB = map[scan] & smog_b;
            }

      }
      else{
            if(scan == my_state ){
                  PORTB = map[scan] & smog_b;
                  PORTB |= x;
            }else if(scan == (my_state + 1)){
                  PORTB = map[scan] & smog_b;
            }
      }
/*霧の発生終了*/
}

int main()
```

```
{
    unsigned long user_cnt = 0;
    bool user_b = false;

    DDRB = 0xFF;
    DDRC = 0x0F;
    DDRD = 0xF6;

    PORTB = 0x00;
    PORTC = 0x30;
    PORTD = 0x00;
  PCICR =  _BV(PCIE1);//ピン変化割り込み
  PCMSK1 = 0x30;

    TCNT0 = 0;
    OCR0A = 249;
    TCCR0A = 2;
    TCCR0B = 3;
    TIMSK0 |= _BV(OCIE0A);//2ミリ秒ごとに点灯行の切り替え

    TCCR2A = 02;
    TCCR2B = 0x04;
    TIMSK2 =  _BV(OCIE2A);//ブザー用割り込み
    OCR2A = 0; //割り込み時間は音階によって変化
    OCR2B = 0;


    sei();
    for (;;) {
        wdt_reset();
     update_sw();
        user_cnt ++;

        /*プレイヤーを点滅*/
        if(user_cnt >= user_CTOP){
            user_cnt = 0;
            if(user_b == false){
                user_b = true;
                x_sub = x;
                x = x & 0;
                x_flag = false;
            }else{
                x_flag = true;
                user_b = false;
                x = x_sub;
            }

        }
        /*switchが押されたときの処理*/
     if (sw_flag) {
            sw_flag = 0;
            switch (sw) {
                case 0:
                    break;
                case 1:/*左*/
                    if(x_flag ==  true ){
                        sw_flag2 = true;
                        DDRD = 0xFE;
                        x = (x >> 7) | (x << 1);
                        smog_b = (smog_b >> 7) | (smog_b << 1);
                        if((map[my_state] & x) == 0){
                            period = 1000;
                            OCR2A = 62;
                        }
                        else{
                            x = (x << 7) | (x >> 1);
                            smog_b = (smog_b << 7) | (smog_b >> 1);
                            period = 1000;
                            OCR2A = 238;
                        }
                    }
                    break;

                case 2:/*右*/
                    if(x_flag ==  true ){
                        sw_flag2 = true;
                        DDRD = 0xFE;
                        x = (x << 7) | (x >> 1);
                        smog_b = (smog_b << 7) | (smog_b >> 1);
                        if((map[my_state] & x) == 0)
                        {
                            period = 1000;
                            OCR2A = 62;

                        }else{
                            x = (x >> 7) | (x << 1);
                            smog_b = (smog_b >> 7) | (smog_b << 1);
                            period = 1000;
                            OCR2A = 238;

                        }
                    }
                    break;
                case 3:/*下*/
                    if(x_flag ==  true ){
                        sw_flag2 = true;
                        DDRD = 0xFE;
                        my_state = (my_state + 1) & 7;
                        if((map[my_state] & x) == 0){
                            period = 1000;
                            OCR2A = 62;
                        }
                        else{
                            my_state = (my_state - 1) & 7;
                            period = 1000;
                            OCR2A = 238;
                        }
                    }
                    break;
            }
        }
    }
    return 0;
}
```