```
/*
 * GccApplication1.c
 *
 * Created: 04/12/2017 12:49:31
 * Author : g
 */

//20MHz clock
#define F_CPU 2000000UL

// Calculate the value needed for
// the CTC match value in OCR1A.
#define CTC_MATCH_OVERFLOW ((F_CPU / 1000) / 8)

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

volatile unsigned long timer1_millis;
long milliseconds_since_dir;
long milliseconds_since_step;

uint8_t dir;
#define dir_delay_ms 3000

uint8_t step;
#define step_delay_ms 2

ISR (TIMER1_COMPA_vect)
{
    timer1_millis++;
}

unsigned long millis ()
{
    unsigned long millis_return;

    ATOMIC_BLOCK(ATOMIC_FORCEON) {
        millis_return = timer1_millis;
    }

    return millis_return;
}

void _step() {
    switch(step) {
        case 0:
            PORTB = 0b00010000;
            break;
        case 1:
            PORTB = 0b00001000;
            break;
        case 2:
```

```
            PORTB = 0b00000100;
            break;
        case 3:
            PORTB = 0b00000010;
            break;

    }
}
void run() {
    unsigned long milliseconds_current = millis();
    if(milliseconds_current - milliseconds_since_dir > dir_delay_ms) {
        dir != dir;
        milliseconds_since_dir = milliseconds_current;
    }

    if(milliseconds_current - milliseconds_since_step > step_delay_ms) {
        step++;
        if(step == 4) step = 0;
        _step();
        milliseconds_since_step = milliseconds_current;
    }
}

int main(void)
{
    DDRB |= 0b00011110;
    // CTC mode, Clock/8
    TCCR1B |= (1 << WGM12) | (1 << CS11);

    // Load the high byte, then the low byte
    // into the output compare
    OCR1AH = (CTC_MATCH_OVERFLOW >> 8);
    OCR1AL = CTC_MATCH_OVERFLOW;

    // Enable the compare match interrupt
    TIMSK1 |= (1 << OCIE1A);

    // Now enable global interrupts
    sei();

    while (1)
    {
        run();
    }
}
```