

```
/*  
Lilypad Arduino Rainbow Fish Twinkle Bag  
Created by Jessica Streck Ortolano  
February 2014, Code edited from: https://www.sparkfun.com/tutorials/312  
*/
```

```
#define LED1 A3  
#define LED2 6  
#define LED3 A5  
#define LED4 10  
#define LED5 11  
#define LED6 A2  
#define LED7 5  
#define LED8 A4  
#define LED9 9
```

```
#define interval 11000  
#define dead_time 1000
```

```
unsigned long PWM_counter = 0;
```

```
int offset = 0;  
int step_size = 200;  
unsigned long on_time = 0;  
unsigned long cycle_start = 0;  
char dir = 1;
```

```
int offset2 = 0;  
int step_size2 = 200;  
unsigned long on_time2 = 0;  
unsigned long cycle_start2 = 0;  
char dir2 = 1;
```

```
byte LED_tracker1 = 1;  
byte LED_tracker2 = 1;
```

```
void setup() {  
  Serial.begin(57600);  
  pinMode(LED1, OUTPUT);  
  pinMode(LED2, OUTPUT);  
  pinMode(LED3, OUTPUT);  
  pinMode(LED4, OUTPUT);  
  pinMode(LED5, OUTPUT);  
  pinMode(LED6, OUTPUT);  
  pinMode(LED7, OUTPUT);  
  pinMode(LED8, OUTPUT);  
}
```

```

pinMode(LED9, OUTPUT);
offset = random(2500,7500);
offset2 = random(2500,7500);
}

void loop()
{
  PWM_counter = micros();

  if( (PWM_counter + offset - cycle_start) >= interval) { // completed cycle, start
over
    on_time += step_size*dir;
    cycle_start = PWM_counter;
    if (on_time >= (interval - dead_time) ) {
      dir *= -1;
      on_time = interval - dead_time;
    }
    else if (on_time <= 0) {
      on_time = 0;
      dir *= -1;
      LED_tracker1 = LED_tracker2;
      while (LED_tracker1 == LED_tracker2) LED_tracker1 = random(1,10);
      offset2 = random(0,4000);
    }
  }
  else if( (PWM_counter + offset - cycle_start) >= (dead_time + on_time) ) { //
time to switch LED off
    //digitalWrite(LED1, LOW);
    LED_off(LED_tracker1);
  }
  else if( (PWM_counter + offset - cycle_start) >= dead_time) { // time to switch
LED on
    //digitalWrite(LED1, HIGH);
    LED_on(LED_tracker1);
  }

  if( (PWM_counter + offset2 - cycle_start2) >= interval) { // completed cycle, start
over
    on_time2 += step_size2*dir2;
    cycle_start2 = PWM_counter;
    if (on_time2 >= (interval - dead_time) ) {
      dir2 *= -1;
      on_time2 = interval - dead_time;
    }
    else if (on_time2 <= 0) {
      on_time2 = 0;
      dir2 *= -1;

```

```

    LED_tracker2 = LED_tracker1;
    while (LED_tracker2 == LED_tracker1) LED_tracker2 = random(1,10);
    offset2 = random(0,4000);
}
}
else if( (PWM_counter + offset2 - cycle_start2) >= (dead_time + on_time2) ) { //
time to switch LED off
    //digitalWrite(LED1, LOW);
    LED_off(LED_tracker2);
}
else if( (PWM_counter + offset2 - cycle_start2) >= dead_time) { // time to switch
LED on
    //digitalWrite(LED1, HIGH);
    LED_on(LED_tracker2);
}
}

```

```

void LED_on(byte LED)
{
    if (LED == 1) digitalWrite(LED1, HIGH);
    if (LED == 2) digitalWrite(LED2, HIGH);
    if (LED == 3) digitalWrite(LED3, HIGH);
    if (LED == 4) digitalWrite(LED4, HIGH);
    if (LED == 5) digitalWrite(LED5, HIGH);
    if (LED == 6) digitalWrite(LED6, HIGH);
    if (LED == 7) digitalWrite(LED7, HIGH);
    if (LED == 8) digitalWrite(LED8, HIGH);
    if (LED == 9) digitalWrite(LED9, HIGH);
}

```

```

void LED_off(byte LED)
{
    if (LED == 1) digitalWrite(LED1, LOW);
    if (LED == 2) digitalWrite(LED2, LOW);
    if (LED == 3) digitalWrite(LED3, LOW);
    if (LED == 4) digitalWrite(LED4, LOW);
    if (LED == 5) digitalWrite(LED5, LOW);
    if (LED == 6) digitalWrite(LED6, LOW);
    if (LED == 7) digitalWrite(LED7, LOW);
    if (LED == 8) digitalWrite(LED8, LOW);
    if (LED == 9) digitalWrite(LED9, LOW);
}

```