

本实验目标：完成一个LED灯逐渐变亮又变暗的小实验

目标分析：为了完成这个实验，我们首先对于它进行架构的分析

传感器：无

控制器：UNO/ESP32

执行器：LED灯

接着是效果分析：

有一个小灯，它能够随着时间逐渐变亮又变暗，用亮度呈现出好像是在呼吸一样的效果

明确目标系统与效果，开始正式学习---

代码全文

```
void setup() {  
    pinMode(3, OUTPUT); // 设置3号引脚为输出模式  
}  
  
void loop() {  
    // 亮度渐增循环  
    for (int i = 0; i <= 255; i = i + 1) {  
        analogWrite(3, i);  
        delay(10);  
    }  
  
    // 亮度渐减循环  
    for (int i = 255; i >= 0; i = i + (-1)) {  
        analogWrite(3, i);  
        delay(10);  
    }  
}
```

代码逐行解析

1. void setup() { ... }

- **功能：**Arduino初始化设置（仅执行一次）
- `pinMode(3, OUTPUT);`
 - `pinMode`：引脚模式设置函数
 - `3`：目标引脚编号（物理接口位置）
 - `OUTPUT`：输出模式（控制外部设备）
 - **特殊说明：**只有支持PWM（脉宽调制）的引脚（标记为~）才能使用 `analogWrite`

2. void loop() { ... }

- **功能：**主循环体（反复执行）

第一个 for 循环（渐亮）

代码结构：

```
for (int i = 0; i <= 255; i = i + 1) {
  analogWrite(3, i);
  delay(10);
}
```

- **for 循环三要素：**
 1. **初始化：** `int i = 0` （创建计数器i，初始值为0）
 2. **条件判断：** `i <= 255` （当i≤255时继续循环）
 3. **迭代操作：** `i = i + 1` （每次循环i增加1）
- **执行流程：**

步骤	操作	当前i值	PWM输出值
1	初始化i=0	0	0（全关）
2	检查i≤255 → 成立	0	0
3	执行循环体	0	0
4	i自增 → i=1	1	5
...
256	i=255时条件成立	255	255（全开）
257	i=256时条件不成立 → 退出循环	-	-

- `analogWrite(3, i);`
 - `analogWrite`：PWM模拟输出函数
 - `3`：目标引脚（必须支持PWM）
 - `i`：占空比参数（0-255）
 - **PWM原理：**通过快速开关产生等效电压

参数值	占空比	等效电压（5V系统）	LED亮度
0	0%	0V	熄灭
127	50%	2.5V	中等亮度
255	100%	5V	最亮

- `delay(10);`
 - 保持当前亮度10毫秒

- 控制亮度变化速度 (10ms×256步=2.56秒完成渐亮)

第二个 for 循环 (渐暗)

```
for (int i = 255; i >= 0; i = i + (-1)) {  
    analogWrite(3, i);  
    delay(10);  
}
```

- 与渐亮循环的区别：
 1. 初始化: `i = 255` (从最大亮度开始)
 2. 条件判断: `i >= 0` (当i≥0时继续循环)
 3. 迭代操作: `i = i + (-1)` (等效于 `i--`)
- 改进建议 (标准写法) :

```
for (int i = 255; i >= 0; i--) {  
    // 循环内容  
}
```

核心概念详解

1. PWM (脉宽调制)

特性	说明
全称	Pulse Width Modulation (脉宽调制)
工作原理	通过调节高电平持续时间占比控制等效电压
Arduino实现	内置硬件PWM, 频率约490Hz (不同引脚可能不同)
分辨率	8位 (0-255共256级)
适用场景	LED调光、电机调速等需要模拟量控制的场合

2. for循环结构

```
for (初始化; 条件; 迭代) {  
    循环体  
}
```

- 执行顺序：
 1. 执行初始化表达式 (仅一次)
 2. 检查条件表达式：
 - 若为真 → 执行循环体 → 执行迭代表达式 → 回到步骤2
 - 若为假 → 退出循环
- 渐亮循环参数分析：

参数	值	说明
初始值	i=0	从最低亮度开始
循环条件	i≤255	包含255次有效循环（0-255）
步长	+1	每次增加1级亮度
总执行次数	256次	产生线性渐变效果

3. 亮度渐变原理

- **人眼视觉暂留**：亮度变化速度>24Hz（约40ms/次）时视为连续变化
- **本程序参数**：
 - 单次亮度变化间隔：10ms
 - 单方向渐变时间：256×10ms=2560ms=2.56秒
 - 完整呼吸周期：2.56×2=5.12秒

硬件连接说明

标准PWM调光电路

Arduino引脚3 → 220Ω电阻 → LED正极（长脚）
LED负极（短脚） → GND

- **电阻计算示例**（红色LED）：
 - 工作电压：2.0V
 - 最大电流：20mA
 - 所需电阻： $R=(5V-2V)/0.02A=150\Omega$ → 选用220Ω标准电阻（最适合，并非必须）

关键问题解析

Q1：为什么选择引脚3？

- **硬件限制**：只有支持PWM的引脚可用 `analogWrite`
- **Arduino UNO的PWM引脚**：3,5,6,9,10,11（标有~符号）

Q2：如何改变呼吸速度？

- **调整延时参数**：

```
delay(5); // 加快速度（总周期2.56秒）  
delay(20); // 减慢速度（总周期5.12秒）
```

- **修改步长**（注意参数范围）：

```
i = i + 2 // 步长改为2，循环次数减半
```

Q3：亮度变化不均匀的可能原因

- **非线性感知**：人眼对亮度变化感知呈对数特性
- **解决方案**：使用伽马校正

```
float gamma = 2.8;
int value = pow((i/255.0), gamma) * 255;
analogWrite(3, value);
```

扩展实验

实验1：非线性渐变

```
void loop() {
    // 加速渐亮
    for (int i = 0; i <= 255; i++) {
        analogWrite(3, i);
        delay(255 - i); // 延时逐渐缩短
    }

    // 减速渐暗
    for (int i = 255; i >= 0; i--) {
        analogWrite(3, i);
        delay(255 - i);
    }
}
```

实验2：多级亮度控制

```
void loop() {
    // 阶梯渐变（10级）
    for (int level=0; level<=10; level++){
        int brightness = map(level, 0,10, 0,255); //map()函数是映射用的，将0,10的值映射到0,255中
        analogWrite(3, brightness);
        delay(500);
    }
}
```

总结

本程序通过PWM技术实现了LED呼吸灯效果，核心要点包括：

1. **PWM原理**：理解占空比对等效电压的控制
2. **循环结构**：掌握 for 循环的三要素和执行流程
3. **硬件知识**：正确选择PWM引脚并连接电路
4. **参数调整**：通过修改延时和步长控制效果

掌握基础实现后，可通过非线性调整、多设备联动等方式扩展应用。

有问题找柯萌ㄟㄎㄎㄎ
mingyoufhh@outlook.com