

Arduino 基础入门教程：LED 闪烁程序解析

本实验目标：完成一个LED灯持续闪烁的程序

目标分析：为了完成这个实验，我们首先对于它进行系统化的分析

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，能够做到每隔一秒亮一次灭一次

明确目标系统，开始正式学习---

代码全文

```
void setup() {                //初始化函数
    pinMode(13, OUTPUT);      // 设置13号引脚为输出模式
}

void loop() {                 //重复执行函数
    digitalWrite(13, HIGH);   // 输出高电平
    delay(1000);              // 等待1000毫秒（1秒）
    digitalWrite(13, LOW);    // 输出低电平
    delay(1000);              // 再次等待1秒
}
```

代码逐行解析

1. void setup() { ... }

- **作用：**Arduino开发板通电后，此函数**仅执行一次**，用于初始化设置。
- **详细解释：**
 - `void`：表示此函数没有返回值（暂时无需深究，固定语法）。
 - `setup()`：函数名称，必须按照此格式定义。
 - `{ ... }`：大括号内包含初始化代码。

关键代码： `pinMode(13, OUTPUT);`

- `pinMode`：Arduino内置函数，用于设置引脚的工作模式。
- `13`：目标引脚的编号（Arduino开发板上的物理接口）。
- `OUTPUT`：模式参数，表示将引脚设置为**输出模式**（可控制外部设备）。
- `;`：语句结束符，表示一行代码结束。

2. `void loop() { ... }`

- **作用：**在 `setup()` 执行后，此函数会**无限循环运行**，包含程序的主要逻辑。
- **详细解释：**
 - `loop()`：函数名称，固定语法。
 - `{ ... }`：大括号内包含循环执行的代码。

关键代码解析：

行1: `digitalWrite(13, HIGH);`

- `digitalWrite`：向数字引脚输出高电平（HIGH）或低电平（LOW）。
- `13`：目标引脚编号（与 `setup()` 中设置的引脚一致）。
- `HIGH`：参数值，表示输出5V电压。
- **作用：**使13号引脚通电，若连接LED则会发光。

行2: `delay(1000);`

- `delay`：暂停程序执行一段时间。
- `1000`：时间参数，单位为毫秒（ms）， $1000\text{ms} = 1\text{秒}$ 。
- **作用：**保持当前状态（LED亮起）1秒钟。

行3: `digitalWrite(13, LOW);`

- `LOW`：参数值，表示输出0V电压。
- **作用：**断开13号引脚的电压，LED熄灭。

行4: `delay(1000);`

- **作用：**保持LED熄灭状态1秒，随后循环重新开始。

核心概念详解

1. 电平 (Voltage Level)

- **定义：**电路中某一点的电压状态。
- **在Arduino中的实现：**
 - **高电平 (HIGH)**：5V电压（部分开发板为3.3V），表示“开启”状态。
 - **低电平 (LOW)**：0V电压，表示“关闭”状态。
- **应用：**通过控制电平变化，可驱动LED、继电器等设备。

2. 数字信号 (Digital Signal)

- **定义：**仅有两种离散状态的信号（HIGH/LOW）。
- **特点：**
 - 简单可靠，抗干扰能力强。
 - 适用于开关控制（如LED亮灭、电机启停）。

3. 引脚模式 (Pin Mode)

- 输入模式 (INPUT)：**
 - 引脚用于接收外部信号（如读取按钮状态、传感器数据）。
 - 示例：`pinMode(2, INPUT);` // 设置2号引脚为输入模式。
- 输出模式 (OUTPUT)：**
 - 引脚用于控制外部设备（如LED、蜂鸣器）。
 - 示例：`pinMode(13, OUTPUT);` // 设置13号引脚为输出模式。

4. 时间控制 (delay() 函数)

- 原理：**暂停程序执行，期间Arduino不执行任何操作。
- 注意事项：**
 - 单位是毫秒（1秒 = 1000毫秒）。
 - 过长延时可能导致程序响应迟缓，需根据场景调整。

硬件连接说明

1. 板载LED（无需接线）

- Arduino UNO开发板的13号引脚已连接一个黄色LED和限流电阻。
- 直接上传代码即可观察到板载LED的闪烁效果。

2. 外接LED电路（可选）

13号引脚 → 220Ω电阻 → LED正极（长脚）
LED负极（短脚） → GND（接地）

- 电阻作用：**限制电流，防止LED过流损坏（推荐220Ω~1kΩ）。
- 极性注意：**LED长脚为正极，短脚为负极，接反则不发光。

常见问题解答

Q1：上传代码后LED不亮

- 检查步骤：**
 - 确认USB线连接正常，开发板通电（电源指示灯亮起）。
 - 检查代码是否上传成功（IDE提示“上传完成”）。
 - 外接LED时，确认正负极连接正确。

Q2: `delay(1000)` 能否修改?

- 可以自由调整:
 - `delay(500)`: LED亮灭各0.5秒, 闪烁频率加快。
 - `delay(2000)`: LED亮灭各2秒, 闪烁频率减慢。

Q3: 为什么需要两个 `digitalwrite` 和 `delay`?

- 逻辑解析:
 1. `digitalwrite(13, HIGH)` → LED亮。
 2. `delay(1000)` → 保持亮1秒。
 3. `digitalwrite(13, LOW)` → LED灭。
 4. `delay(1000)` → 保持灭1秒。
- 缺少任一语句将导致闪烁异常 (如常亮或快速闪烁)。

扩展实验

实验1: 不对称闪烁

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(200);    // 亮0.2秒  
    digitalWrite(13, LOW);  
    delay(800);    // 灭0.8秒  
}
```

实验2: 多LED控制

```
void setup() {  
    pinMode(12, OUTPUT); // 新增12号引脚控制第二个LED  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(12, HIGH);  
    digitalWrite(13, LOW);  
    delay(500);  
    digitalWrite(12, LOW);  
    digitalWrite(13, HIGH);  
    delay(500);  
}
```

总结

本代码通过控制引脚电平变化实现LED周期性闪烁, 涉及数字信号输出、引脚模式设置和时间延迟等基础概念。理解此程序后, 可进一步学习传感器输入、PWM调光等进阶内容。

有问题找柯萌(ง•̀•́)ง
mingyoufhh@outlook.com