

51单片机-点亮第一个LED

仿真配置

根据视频：[51单片机Proteus仿真教程](#)，在 Proteus 上配置好51单片机最小系统

LED点亮逻辑

根据电路原理，LED左右两端需要分别接入高电平和低电平，使得电路串通，LED才会亮。

一般来说，要点亮LED，需要一个电源 POWER 为其提供一个高电平，然后用代码控制寄存器低/高电平，以实现LED的亮/灭

如何用代码操控寄存器的高低电平？

1. **操控整个8位寄存器**：用8位二进制来代表8个接口的高低电平，1为高电平，0为低电平，如二进制 1111 1110 代表最低位接口（Px.0接口）为低电平，其余为高电平。但是在C语言中直接写 1111 1110 会被认作十进制数而非二进制数，因此需要转化为十六进制数 1111 1110 -> 0xFE，代码如下：

```
1  /*令P1寄存器中第一个接口为低电平，其余为高电平*/
2  P1 = 0xFE //1111 1110
3
4  /*令P2寄存器中高低电平交替*/
5  P2 = 0x55 //0101 0101
```

2. **单独操控位**：在某些头文件中有对位寄存器的定义，但是本文使用的 "REG51.h" 没有，因此要用 sbit 手动定义：

```
1  sbit P2_0 = 0xA0; //具体哪个寄存器可以open头文件查看寄存器地址
2  P2_0 = 0; //0为低电平 1为高电平
3
4  sbit P2_1 = 0xA1; //定义P2.1接口
5  P2_1 = 0; //0为低电平 1为高电平
```

硬件设置

在 P2.0/A8 接口上接上一个 LED-YELLOW，由于 LED-YELLOW 所需电流小，为了避免 LED-YELLOW 烧坏，需在其右端连上一个限流电阻 RES，为 LED-YELLOW 右端配置一个高电平，需要添加电源 POWER。

代码实现

为了让 LED-YELLOW 成功点亮，需要在其左端（即接口位）配置一个低电平，由于其左端接入的是接口 P2.0/A8，因此需要寄存器为 P2.0/A8 接口输出一个低电平以完成电路的串通，完整代码如下：

```
1  #include "REG51.h"
2  sbit P2_0 = 0xA0; //定义位寄存器
3
4  void main(void){
5      while(1){
6          P2_0 = 0;
7      }
8  }
```