

弟子教你学51单片机技术



# 从零到项目开发学51单片机 与C语言技术

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程

弟子教你学51单片机技术



# 第三讲 点亮一个LED发光管

基于DZ80C52开发板

QQ:2539606882

从零到项目开发学51单片机与C语言技术视频教程



## 如何点亮一个发光管

- 本讲主要学习以下内容：
- 1、单片机最小系统和典型系统
- 2、如何建立工程、完成“点亮一个发光管”程序的编写、编译
- 3、如何进行程序烧写

学习目标：学会用C语言控制单片机，点亮开发板上的第1个LED发光管。复习单片机的数电、模电、C语言基础知识。



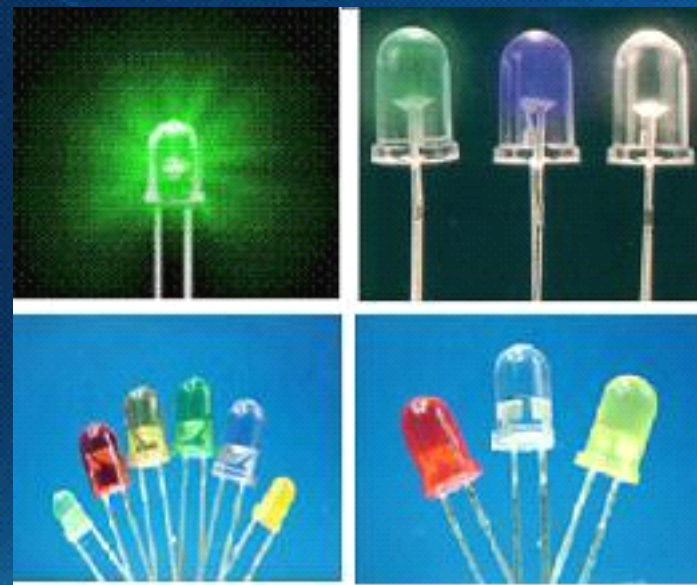
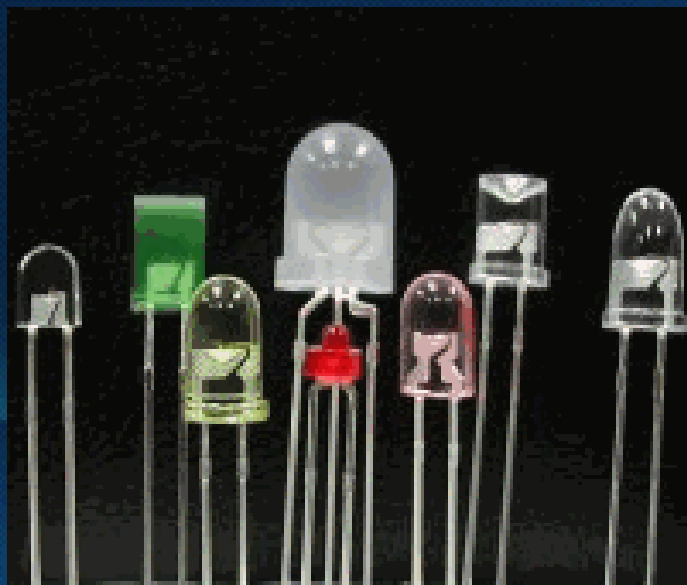
# 需要用到的模拟电子技术知识

- 1、LED发光二极管。
- 它是半导体二极管的一种，可以把电能转化成光能；常简称为LED。发光二极管与普通二极管一样是由一个PN结组成，也具有单向导电性。
- 当给发光二极管加上正向电压后，从P区注入到N区的空穴和由N区注入到P区的电子，在PN结附近数微米内分别与N区的电子和P区的空穴复合，产生自发辐射的荧光。不同的半导体材料中电子和空穴所处的能量状态不同。当电子和空穴复合时释放出的能量多少不同，释放出的能量越多，则发出的光的波长越短。常用的是发红光、绿光或黄光的二极管。

弟子教你学51单片机技术



# 认识一下各种LED发光管



从零到项目开发学51单片机与C语言技术视频教程

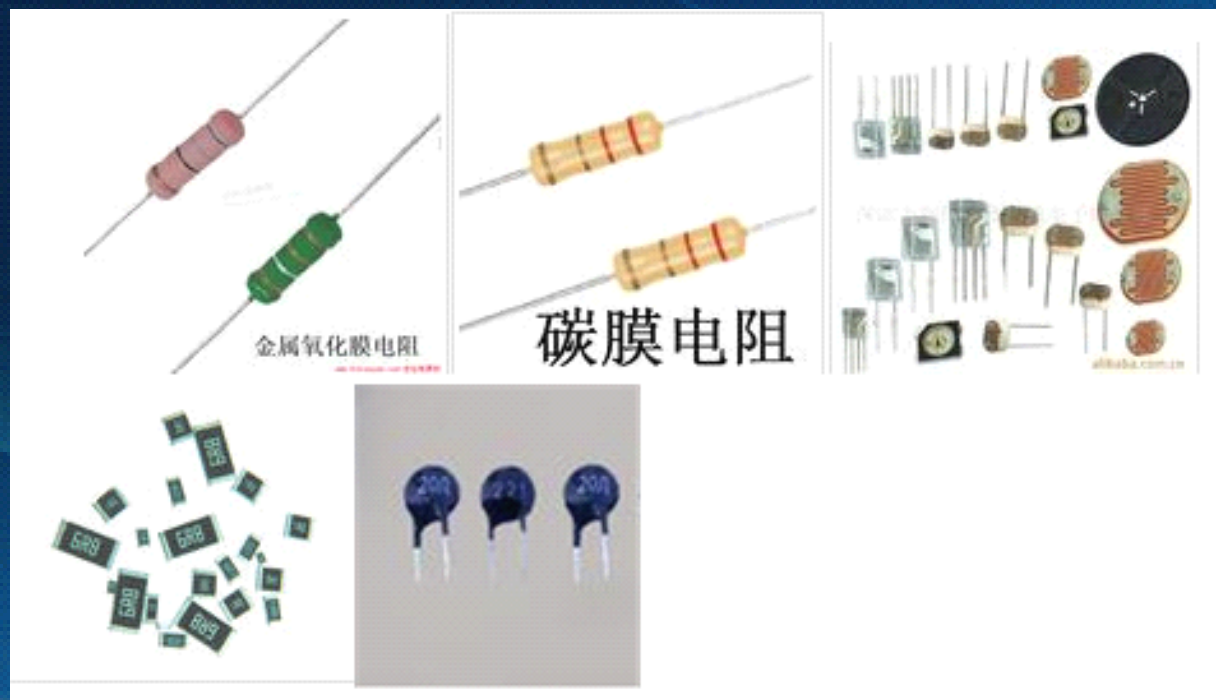


# 电阻

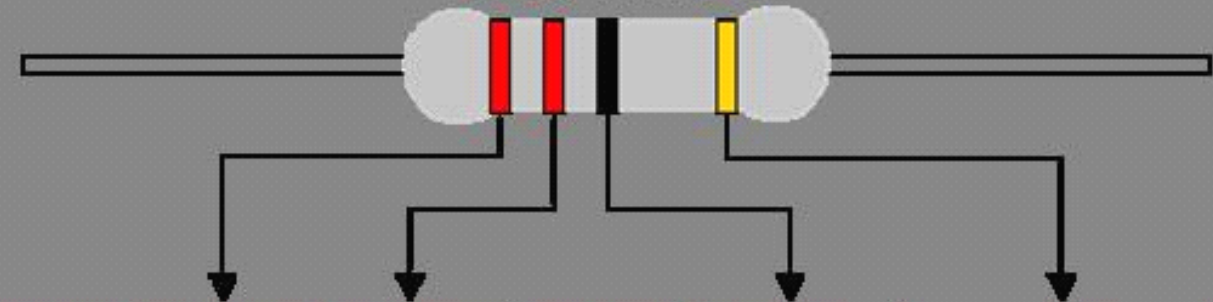
- 电阻，因为物质对电流产生的阻碍作用，所以称其该作用下的电阻物质。电阻将会导致电子流通量的变化，电阻越小，电子流通量越大，反之亦然。没有电阻或电阻很小的物质称其为电导体，简称导体。不能形成电流传输的物质称为电绝缘体，简称绝缘体。



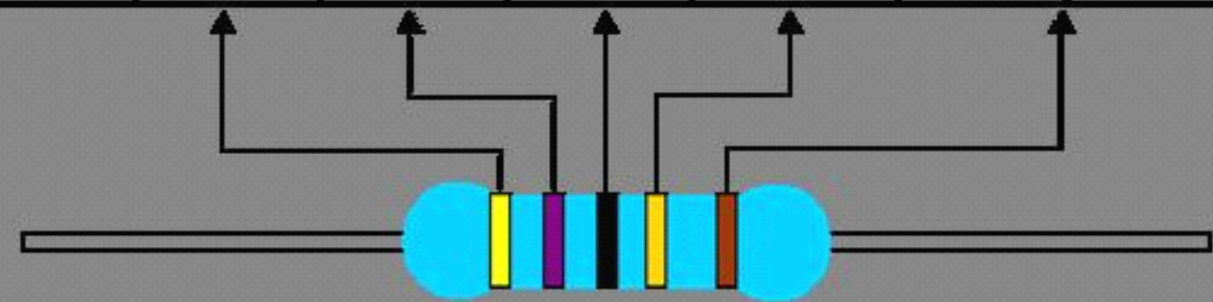
# 认识一下各种电阻



### 数值的读取方法



颜色	每一段	第二段	第三段	乘数	误差
黑色	0	0	0	1	
棕色	1	1	1	10	± 1% F
红色	2	2	2	100	± 2% G
橙色	3	3	3	1K	
黄色	4	4	4	10K	
绿色	5	5	5	100K	± 0.5% D
蓝色	6	6	6	1M	± 0.25% C
紫色	7	7	7	10M	± 0.10% B
灰色	8	8	8		± 0.05% A
白色	9	9	9		
金色				0.1	± 5% J
银色				0.01	± 10% K
无					± 20% M

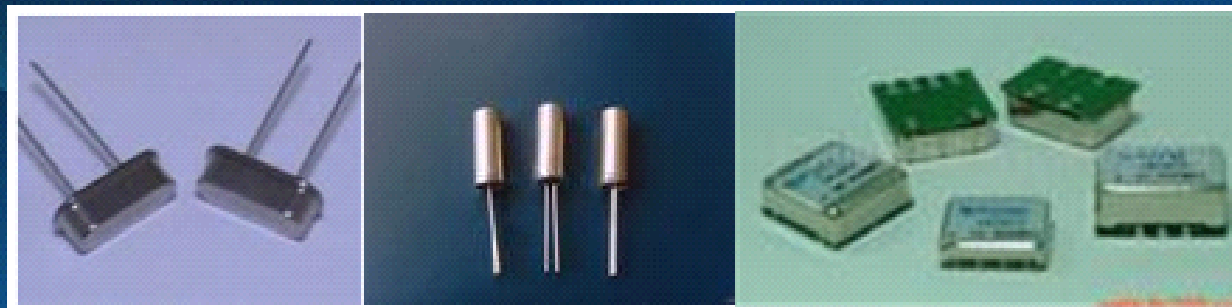


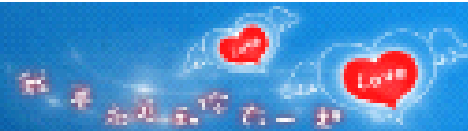




# 晶振

- 晶振全称为晶体振荡器，其作用在于产生原始的时钟频率，这个频率晶振经过频率发生器的放大或缩小后就成了电脑中各种不同的总线频率。





# 电源

- 发电机能把机械能转换成电能，干电池能把化学能转换成电能。发电机、电池本身并不带电，它的两极分别有正负电荷，由正负电荷产生电压(电流是电荷在电压的作用下定向移动而形成的)，电荷导体里本来就有，要产生电流只需要加上电压即可，当电池两极接上导体时为了产生电流而把正负电荷释放出去，当电荷散尽时，也就荷尽流(压)消了。干电池等叫做电源。通过变压器和整流器，把交流电变成直流电的装置叫做整流电源。能提供信号的电子设备叫做信号源。晶体三极管能把前面送来的信号加以放大，又把放大的信号传送到后面的电路中去。晶体三极管对后面的电路来说，也可以看做是信号源。整流电源、信号源有时也叫做电源，电压的单位是V。单片机供电电压一般为5V 3.3V。



# 电流

- 电流，是指电荷的定向移动。电源的电动势形成了电压，继而产生了电场力，在电场力的作用下，处于电场内的电荷发生定向移动，形成了电流。电流的大小称为电流强度（简称电流，符号为 $I$ ），是指单位时间内通过导线某一截面的电荷量，每秒通过1库仑的电量称为1「安培」（A）。安培是国际单位制中所有电性的基本单位。除了A，常用的单位有毫安（mA）、微安（ $\mu A$ ）。



# 需要用到的数字电子技术知识

- 二进制
- 二进制是计算技术中广泛采用的一种数制。二进制数据是用0和1两个数码来表示的数。它的基数为2，进位规则是“逢二进一”，借位规则是“借一当二”，由18世纪德国数理哲学大师莱布尼兹发现。当前的计算机系统使用的基本上是二进制系统



# 十六进制

- 是计算机中数据的一种表示方法.同我们日常中的十进制表示法不一样.它由0-9,A-F,组成.与10进制的对应关系是:0-9对应0-9;A-F对应10-15;N进制的数可以用0---(N-1)的数表示超过9的用字母A-F。



# 字节

- 字节(Byte)是计算机信息技术用于计量存储容量和传输容量的一种计量单位，1个字节等于8位二进制。



## 位的定义

- 数据存储的最小单位。在计算机中的二进制数系统中，位，简记为b,也称为比特，每个0或1就是一个位(bit)。计算机中的CPU位数指的是CPU一次能处理的最大位数。



# 需要用到的C语言基础知识

- C语言是一种计算机程序设计语言。它既具有高级语言的特点，又具有汇编语言的特点。它由美国贝尔研究所的D.M.Ritchie于1972年推出。1978后，C语言已先后被移植到大、中、小及微型机上。它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到C语言，适于编写系统软件，三维，二维图形和动画。具体应用比如单片机以及嵌入式系统开发。





# 头文件

- 在C语言家族程序中，头文件被大量使用。一般而言，每个C++/C程序通常由头文件(header files)和定义文件(definition files)组成。头文件作为一种包含功能函数、数据接口声明的载体文件，用于保存程序的声明(declaration)，而定义文件用于保存程序的实现(implementation)。而且.c就是你写的程序文件。



# 单片机最小系统和典型系统

- 最小系统：指能维持单片机运行的最简单配置的系统。
- 组成：单片机、电源、时钟电路（最常用的是外部晶振）、复位电路（最常用的是外部复位）。
- 复位：上电复位、手工复位、看门狗复位。
- 时钟：决定了程序运行的快慢
- 时钟周期、时钟频率，时钟的作用
- RC振荡器的优点：成本低，可以集成在单片机内部。
- 采用晶体振荡器的优点：精度高
- 内外时钟选择方法：在ISP下载用户程序时，在STC-IAP中设置
- 各种实用的单片机应用系统：在最小系统基础上，通过I/O接口连接各种外部输入输出设备。
- 1、发光管和单片机的什么脚连？
- 2、可以直接连吗？

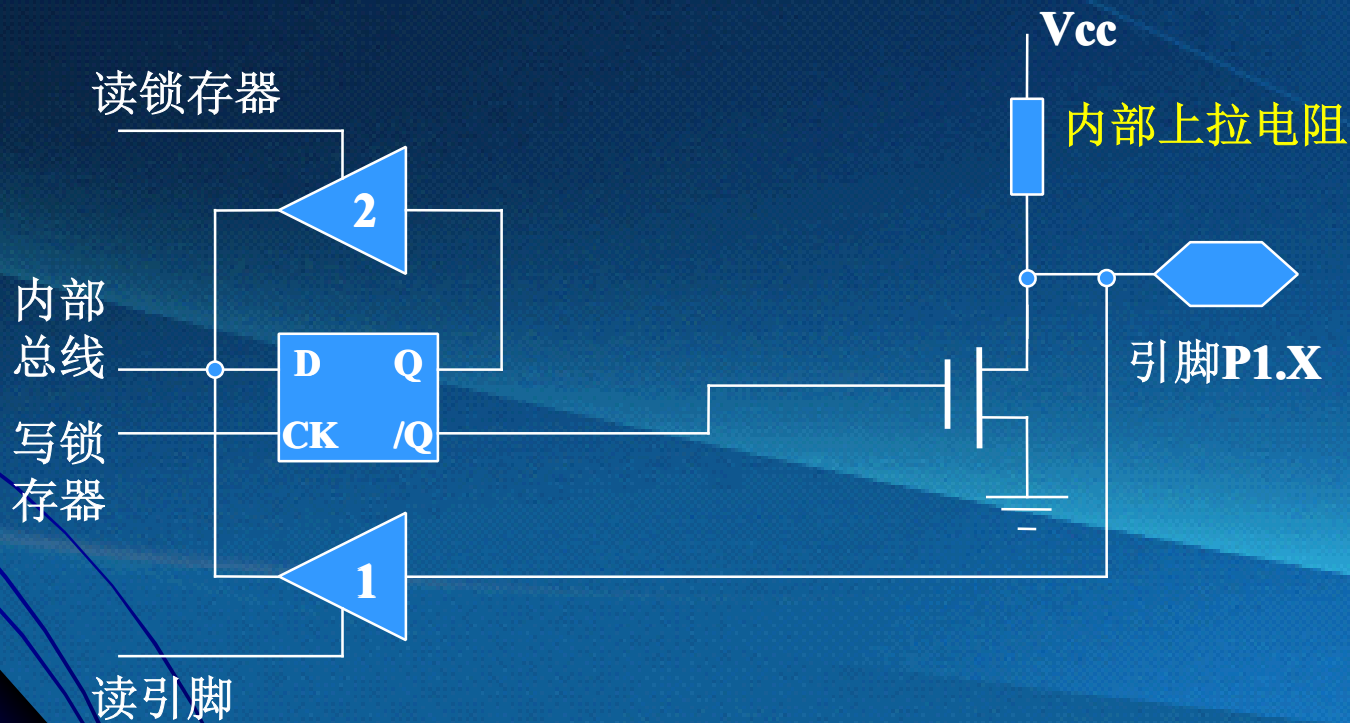


## I/O端口驱动能力

- 每个I/O口允许最大20mA的灌电流，可以直接驱动LED和继电器。
- 高电平输出时，一般是对负载提供电流，其提供的电流叫“拉电流”；
- 低电平输出时，一般是要吸收负载的电流，其吸收的电流叫“灌电流”。

# 单片机的引脚（P1口）

- P1.0—P1.7: 准双向I/O口（内置了上拉电阻）





## 51开发板介绍

- 51开发板介绍、LED发光管简介、如何用开发板快速完成本次实验所需硬件电路的搭建。
- 现在市面上单片机实验板种类繁多，有的非常简陋，有的相对高级，价格与外围器件扩展的程度正相关。弟子设计的这一款DZ80C51实验板个人认为不算简陋但也称不上高级，和网上的各种普及型实验板都差不多，市售价格在200元左右，非常适合初学者使用。能够把这款实验板的所有功能都开发出来的话，你的单片机开发水平就相当可以了。由于电路原理非常简单，电路原理图与PCB图我将免费公布，以利于广大初学者自制，需要者请联系向弟子索要。你也可以参考弟子板子的硬件结构，根据自己的要求自行设计。

## 实战环节二：UV2软件、STC-ISP、 USB驱动的安装

- 具体见我们编写的《51单片机C语言实践教程》由于网上流传的KEIL UV2汉化版，许多术语翻译很不准确，给初学者学习带来很大的困惑，因此我们对KEIL UV2汉化版进行了术语翻译的修正，本视频教材采用KEIL UV2汉化修正版。

弟子教你学51单片机技术

## 实战环节三：如何建立工程、完成“点亮一个发光管” 程序的编写、编译

- 常见51的C编译器简介
- 最简单的C语言程序结构
- REG51.H头文件
- 如何通过I/O口输出
- 位变量定义
- C51中端口及其位的表示
- While语句

从零到项目开发学51单片机与C语言技术视频教程



## 常见51的C编译器

- 51单片机的C编译器也简称C51编译器，有很多种，一般用的是KEIL的C51编译器，它包含在UV2集成开发环境中，不需要单独安装。





## 你的第一个单片机程序

- `#include "reg52.h" /*必需加的头文件*/`
- `void main(void) /*主函数声明*/`
- `{`
- 这里写你的程序
- `}`
  
- C程序有起点，却没有终点。
- 上面的程序就这么简单？！是不是真的呀？当然是真的，虽然简单，但这就是一个完整的单片机程序。



## 程序讲解

- 加入主函数main
- 加入大括号 { }
- 把P1放入大括号中，再给P1赋值,这时可以给P1赋一个8位的二进制作数值 P1口有8位，现在只用到它的第1位。P1.0口，看电路原理图知，只要P1.0口输出一个低电平0就可以点亮发光管。别的7个输出高电平1，用二进制的写法是 B1111 1110,用十六进制的写法是0xfe,开如不熟悉数字电路中的进制，你可以用查表法来学习。现在就把 P1=0xfe;放入主函数中。
- 现在暂时不管C语言的语法，我们把第一次写好的源码编译一次看看。
- 编译成功后，把生成的HEX文件烧写到单片机芯片上，看看实际的控制结果。第一个LED灯有没有点亮？



## 一只勤劳的小蜜蜂

- 单片机象一只勤劳的小蜜蜂，不到死的一刻都会工作的，单片机也一样，只要给它通电了，它会一直工作，这也是我们编写的程序会一直执行下去。程序是由主函数开始的，也是由主函数结束，但程序要一直运行的，在上面的程序中，执行的源码就停在**P1=0xfe**上了，也就是说，点亮了一个发光管后，它就不干活了。因此，我们要想单片机一直为你干活，就要想办法了，C语言的发明者们给我们发明了一条循环语句。只要把这个循环语句加入到我们的主函数中，单片机就能听我们的话，好好干活。
- 重写一个点亮一个LED灯的项目源码。
- 再编译，烧写新的HEX文件到单片机芯片中，看看结果。



# 一个完整的点亮LED源码程序

```
• #include<reg52.h> //包含头文件，一般情况不需要改动，
• //头文件包含特殊功能寄存器的定义

• sbit LED=P1^0;//用sbit关键字定义LED到P1.0端口，
• //LED是自己任意定义且容易记忆的符号
• /*-----
• 主函数
• -----*/
• void main (void)
• {
• //此方法使用bit位对单个端口赋值
• LED=1; //将P1.0口赋值1，对外输出高电平
• LED=0; //将P1.0口赋值0，对外输出低电平

• while (1) //主循环
• {
• //主循环中添加其他需要一直工作的程序
• }
• }
```



# 烧写第一个程序看结果

- 对于一点成就感都没有的初学者来说，激动人心的时候就要到了：我们要把这个程序固化到单片机的ROM里了。连接好串口线，烧写程序，看看开发板上的程序运行了吗？第一个LED发光管点亮了吗？如果是的话，恭喜你啊！你已经成为一名单片机的开发人员啦！呵呵，虽然这个程序没有什么实用价值，你也不明白语句的意义，但是它给你带来成就感和自信心，这才是作为初学者最需要的，有了信心与兴趣，一切都不再成为负担和难题。



# 包含语句

# include <系统文件名>

# include “用户文件名”

系统自带文件，文件名必须由<>括起来，程序员无需人工添加该文件到工程中。编译时，会自动在系统的目录中寻找找到它。

用户文件，文件名必须由“ ”括起来，程序员必须人工添加该文件到项目中。编译时，会在当前项目的目录中寻找找到它。

预处理指令，相当于汇编中的伪指令。

仅在编译过程中起作用。

用#开头

# include是文件包含指令  
这条指令会导致预处理器把#include后面跟的文件的内容替换到这条指令的位置。



## 头文件知识

- REG51.H、REG52.H、STC.H
- STC.H适用于所有STC出的 51单片机(包括1T和传统单片机)
- 相应头文件内包含: REG51.H
- 头文件包含特殊寄存器SFR定义, 包含特殊功能位定义等
- 还可以自己写自己用的专用头文件, 定义你需要定义的IO输出输入口, 更能方便自己写程序。
- 头文件在K软件中的位置

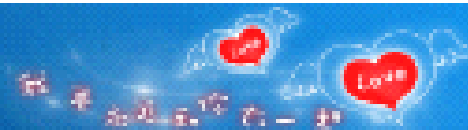




## 特殊功能位定义

- 也可以说是特殊功能位声明。
- Sbit: 一般用来定义一个特殊功能寄存器 (SFR) 的可寻址的某一位。利用它, 可以访问8051单片机
- 特殊功能寄存器 ( SFR ) 中的可寻址位。
- 一般格式:
- Sbit 位变量名 = 特殊功能寄存器名 ^ 位位置
- Sbit LED1 = P0 ^ 1
- 注意: SFR可寻址位只能通过特殊功能位定义的方式来使用它, 大部分在STC.H头文件中已经定义了, 用户不需要再自己定义, 但端口寄存器的某一位需要用户自己定义。





## C51中端口及其位的表示

端口**P1**的数据寄存器用**P1**表示，某一位用**P1 . 0**表示。

**Pn**      **Pn.7** **Pn.6** **Pn.5** **Pn.4** **Pn.3** **Pn.2** **Pn.1** **Pn.0**

端口控制寄存器的相应位设置为1表示输出高电平，设置为0表示输出低电平。

输出有锁存，一直保持到被新的值取代为止。



## While语句

- While (表达式)
- {语句 (内部也可为空) }
- 特点: 先判断表达式, 后执行语句。
- 原则: 若表达式不是0, 即为真, 那么执行语句。
- 否则跳出while 语句。



## 编译

- 程序编写好后或者是修改后，必须进行编译。
- 如何观察存储器已用量。



## 实战环节四：如何进行程序烧写

- 烧写程序的几种方法：
  - 1、用传统的并行烧写器，不推荐。
  - 2、采用目前流行的IAP在线下载程序，STC的51单片机可以不要编程器，通过USB或串口下载程序。
- STC提供的下载编程工具采用串口下载；



# Proteus仿真软件

- Proteus仿真软件是收费软件，听说要1000多元一个软件的，弟子不提供本软件给你，如需要本软件的请自己联系软件商购买正版软件。
- 另外也可以自己在网上下载试用版，试用后24小时内删除掉，谢谢理解。
- 弟子不提供Proteus仿真软件的技术支持。
- 本视频中部分实验用试用版Proteus进行讲解，有心想学好单片机的朋友，请购买DZ-80C51或DZ-80C52开发板



## 结合[例程]讲解如何点亮多个发光管

```
• #include "reg52.h" /*包含STC单片机头文件*/
• sbit LED0 = P1^0;
• //sbit LED1 = P1^1;
• //sbit LED2 = P1^2;
• //sbit LED3 = P1^3;
• //sbit LED4 = P1^4;
• //sbit LED5 = P1^5;
• //sbit LED6 = P1^6;
• //sbit LED7 = P1^7;
• void main(void)
• {
•     LED0=0; /*相关引脚为输出底电平*/
•     // LED1=0;
•     // LED2=0;
•     // LED3=0;
•     // LED4=0;
•     // LED5=0;
•     // LED6=0;
•     // LED7=0;
•     while(1);
• }
```



# STC单片机不断电下载

- 前面我们介绍是STC单片机的标准下载方式，操作过程中需要按电源开关通断一次，也就是要先断电、然后再上电，下面我们介绍一种不断电下载方法（也称自定义下载），软件编写时稍微复杂些，但操作方便了不少，特别是反复烧写同一例程时，比较方便。
- 不断电下载操作步骤：
  - 1、将uart.c、uart.h这对文件拷贝到项目文件夹中，并添加到项目中。
  - 2、在主文件开头加入#include"uart.h"
  - 3、在主函数开头加入uart1\_ini();
  - 4、编译后，先用标准方法下载一次。但如果上一次采用的是不断电下载，则本步骤可以跳过。
  - 5、对“自定义页”进行设置，波特率9600，预设自定义下载命令字符为12（不修改，使用默认值也可以），然后点“发送”按钮，程序就开始下载了。
  - 6、建议选中“当目标代码发生变化后自动调入文件，并立即发送自定义下载命令”，这样“发送”按钮也不用点了，程序编译后，自动完成下载。



## 自动下载烧写实验

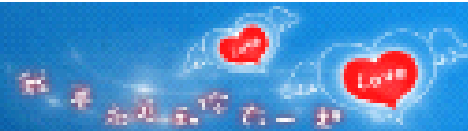
- `#include"reg52.h"/*包含STC单片机头文件*/`
- `/*自定义下载必须加下面这条语句*/`
- `#include"uart.h"/*包含串口相关函数头文件-----自动下载程序源码*/`
- `void main(void)`
- `{`
- `/*自定义下载必须加下面这条语句，预设自定义下载命令字符为12，可以在stc.h中修改*/`
- `uart1_ini();/*串口初始化-----自动下载程序源码*/`
- 
- `P1=0xfe;/*输出底电平点亮1个发光管*/`
- `while(1);`
- 
- `}`





## 学习资料

- STC89C52XX数据手册.pdf
- 《51单片机C语言实践教程》
- 在上一讲“可参考”文件夹中有大量网上收集的51
- 单片机教程，可供阅读。



## 总结

- 这一讲到这里也讲得差不多了，最好总结一下：
- 用事实来说话，学单片机难不难呀？
- 单片机主要是用来实现自动控制用的。生活中，每天都同它有关系发生。它只是装在产品的内部，我们人眼没有直接看到它。举个例子吧，你手上的MP3、MP4、MP5、现代大学生每人都有一个手机，它里面的都是由单片机控制的，现在的叫法做(RMA芯片)。要学会单片机，只要你学会了如何控制单片机的IO口，你就进入了单片机世界的大门。



## 课后练习

1. 第一个发光管以间隔200ms闪烁
2. 8个发光管由上至下间隔1s流动，其中每个管亮500ms, 灭500ms, 亮时蜂鸣器响，灭时关闭蜂鸣器，一直重复下去。
3. 8个发光管来回流动，第个管亮100ms, 流动时让蜂鸣器发出“滴滴”声。
4. 用8个发光管演示出8位二进制数累加过程。
5. 8个发光管间隔200ms由上至下，再由下至上，再重复一次，然后全部熄灭再以300ms间隔全部闪烁5次。重复此过程。
6. 间隔300ms第一次一个管亮流动一次，第二次两个管亮流动，依次到8个管亮，然后重复整个过程。
7. 间隔300ms先奇数亮再偶数亮，循环三次；一个灯上下循环三次；两个分别从两边往中间流动三次；再从中间往两边流动三次；8个全部闪烁3次；关闭发光管，程序停止。