

51 单片机汇编语言教程-由慧净助学会员收集整理（全部 28 课）

51 单片机汇编语言教程：第 4 课-第一个单片机小程序

（基于 HJ-1G、HJ-3G 实验板）

上一次我们的程序实在是没什么用，要灯亮还要重写一下片子，下面我们要让灯持续地闪烁，这就有一定的实用价值了，比如能把它当成汽车上的一个信号灯用了。怎样才能让灯持续地闪烁呢？实际上就是要灯亮一段时间，再灭一段时间，也就是说要 P10 持续地输出高和低电平。怎样实现这个要求呢？请考虑用下面的指令是否可行：

```
SETB P10
```

```
CLR P10 .....
```

这是不行的，有两个问题，第一，计算机执行指令的时间很快，执行完 SETB P10 后，灯是灭了，但在极短时间（微秒级）后，计算机又执行了 CLR P10 指令，灯又亮了，所以根本分辨不出灯曾灭过。第二，在执行完 CLR P10 后，不会再去执行 SETB P10 指令，所以以后再也没有机会让灭了。

为了解决这两个问题，我们能做如下设想，第一，在执行完 SETB P10 后，延时一段时间（几秒或零点几秒）再执行第二条指令，就能分辨出灯曾灭过了。第二在执行完第二条指令后，让计算机再去执行第一条指令，持续地在原地兜圈，我们称之为循环，这样就能完成任务了。

以下先给出程序（后面括号中的数字是为了便于讲解而写的，实际不用输入）：

；主程序：

```
LOOP: SETB P10          ; ( 1 )
      LCALL DELAY       ; ( 2 )
      CLR P10           ; ( 3 )
      LCALL DELAY       ; ( 4 )
      AJMP LOOP         ; ( 5 )
```

；以下子程序

```
DELAY: MOV R7, #250     ; ( 6 )
D1:    MOV R6, #250     ; ( 7 )
D2:    DJNZ R6, D2      ; ( 8 )
      DJNZ R7, D1       ; ( 9 )
      RET               ; ( 1 0 )
      END               ; ( 1 1 )
```

按上面的设想分析一下前面的五条指令。

第一条是让灯灭，第二条应当是延时，第三条是让灯亮，第四条和第二条一模一样，也是延时，第五条应当是转去执行第一条指令。第二和第四条实现的原理稍后谈，先看第五条，LJMP 是一条指令，意思是转移，往什么地方转移呢？后面跟的是 LOOP，看一下，什么地方还有 LOOP，对了，在第一条指令的前面有一个 LOOP，所以很直观地，我们能认识到，它要转到第一条指令处。这个第一条指令前面的 LOOP 被称之为标号，它的用途就是给这一行起一个名字，便于使用。是否一定要给它起名叫 LOOP 呢？当然不是，起什么名字，完全由编程序的人决定，能称它为 A，X 等等，当然，这个时候，第五条指令 LJMP 后面的名字也得跟着改了。

第二条和第四条指令的用途是延时，它是怎样实现的呢？指令的形式是 LCALL，这条指令称为调用子程序指令，看一下指令后面跟的是什么，DELAY，找一下 DELAY，在第六条指令的前面，显然，这也是一个标号。这条指令的作用是这样的：当执行 LCALL 指令时，程序就转到 LCALL 后面的标号所标定的程序处执行，如果在执行指令的过程中遇到 RET 指令，则

51 单片机汇编语言教程-由慧净助学会员收集整理（全部 28 课）

程序就返回到 LCALL 指令的下面的一条指令继续执行，从第六行开始的指令中，能看到确实有 RET 指令。在执行第二条指令后，将转去执行第 6 条指令，而在执行完 6，7，8，9 条指令后将遇到第 10 条指令：RET，执行该条指令后，程序将回来执行第三条指令，即将 P10 清零，使灯亮，然后又是第四条指令，执行第四条指令就是转去执行第 6，7，8，9，10 条指令，然后回来执行第 5 条指令，第 5 条指令就是让程序回到第 1 条开始执行，如此周而复始，灯就在持续地亮、灭了。

在标号 DELAY 标志的这一行到 RET 这一行中的所有程序，这是一段延时程序，大概延时零点几秒，至于具体的时间，以后我们再学习如何计算。程序的最后一行是 END，这不是一条指令，它只是告诉我们程序到此结束，它被称为伪指令。

单片机内部结构分析：为了知道延时程序是如何工作的，我们必需首先了解延时程序中的一些符号，就从 R1 开始，R1 被称之为工作寄存器。什么是工作寄存器呢？让我们从现实生活中来找找答案。如果出一道数学题：123+567，让你回答结果是多少，你会马上答出是 690，再看下面一道题：123+567+562，要让你要上回答，就不这么不难了吧？我们会怎样做呢？如果有张纸，就不难了，我们先算出 123+567=690，把 690 写在纸上，然后再算 690+562 得到结果是 1552。这其中 1552 是我们想要的结果，而 690 并非我们所要的结果，但是为了得到最终结果，我们又不得不先算出 690，并记下来，这其实是一个中间结果，计算机中做运算和这个类似，为了要得到最终结果，一般要做很多步的中间结果，这些中间结果要有个地方放才行，把它们放哪呢？放在前面提到过的 ROM 中能吗？显然不行，因为计算机要将结果写进去，而 ROM 是不能写的，所以在单片机中另有一个区域称为 RAM 区（RAM 是随机存取存储器的英文缩写），它能将数据写进去。特别地，在 MCS-51 单片机中，将 RAM 中分出一块区域，称为工作寄存器区。

51 实验板推荐(点击下面的图片可以进入下载资料链接)

