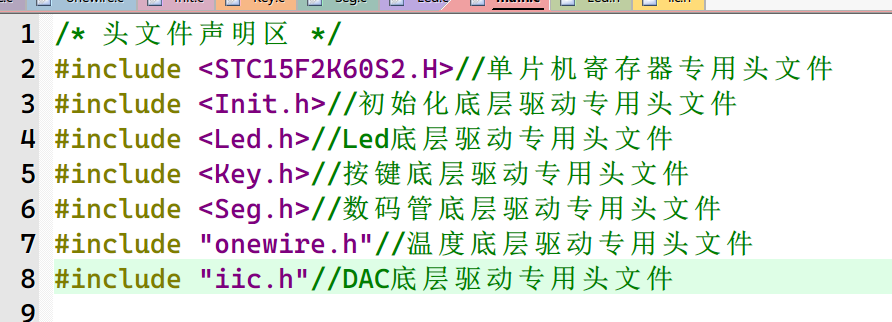
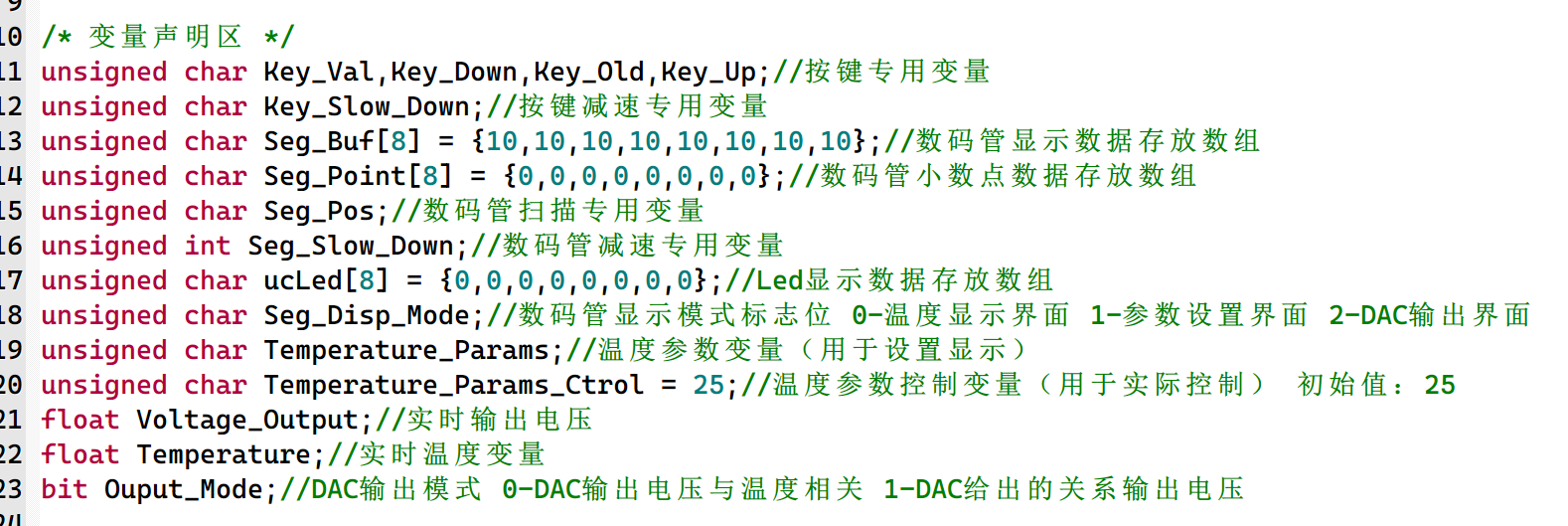
一、整体框架

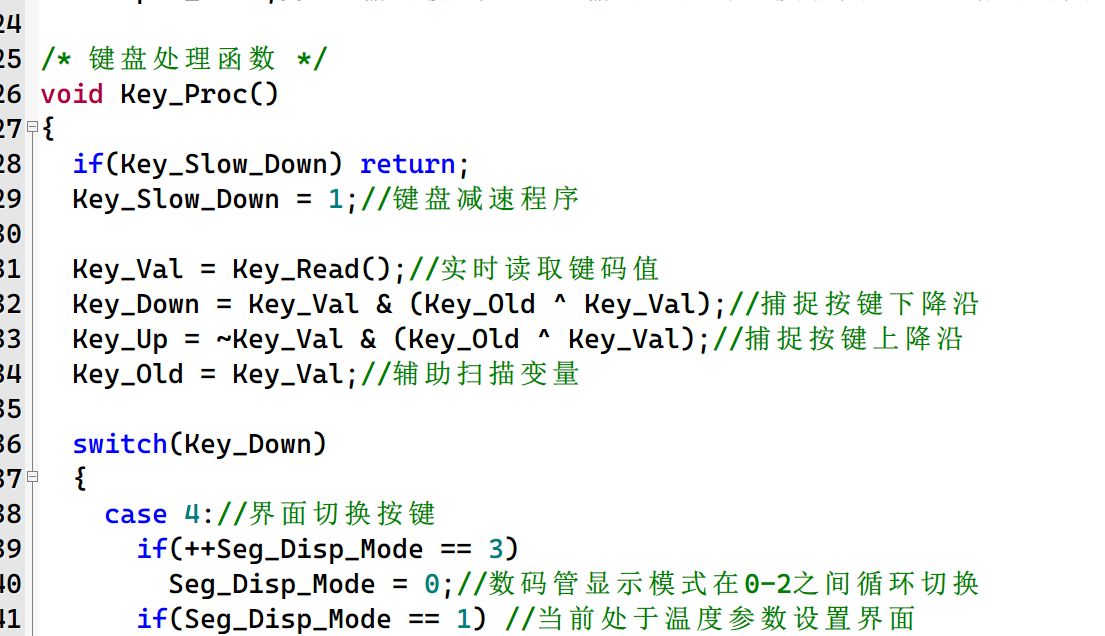
1.头文件声明区域：所有底层驱动声明的一个地方



2.变量声明区域：所有用户变量声明的一个地方



3.按键处理区域：所有按键相关的操作都在这个函数内编写



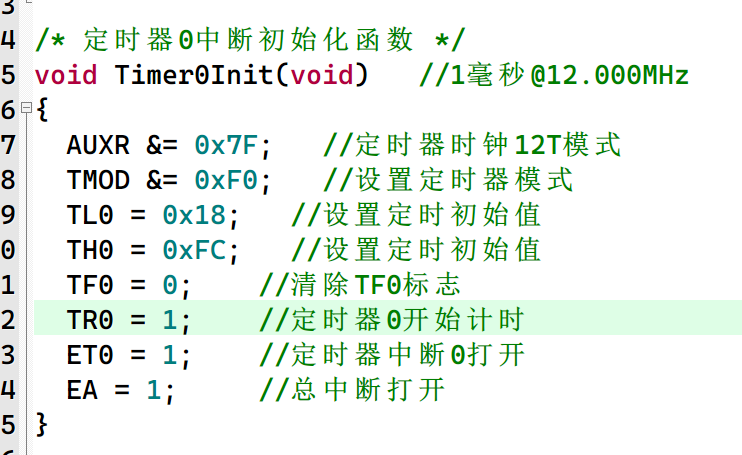
4.信息处理区域：分为两大部分 信息采集和数据显示



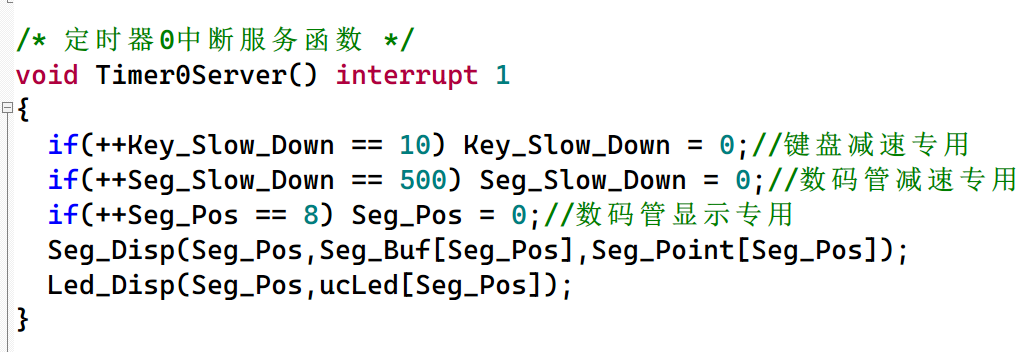
5.其他显示区域：Led、蜂鸣器、继电器等其他外设存放的地方



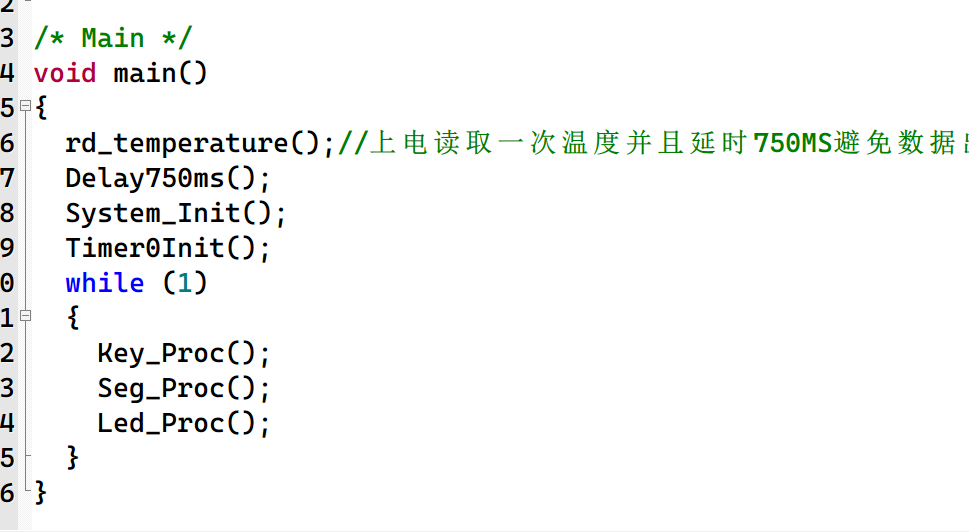
6.定时器初始化函数：一般无需做任何更改



7.中断服务函数：一般是用于计时和计数用 一毫秒执行一次中断服务



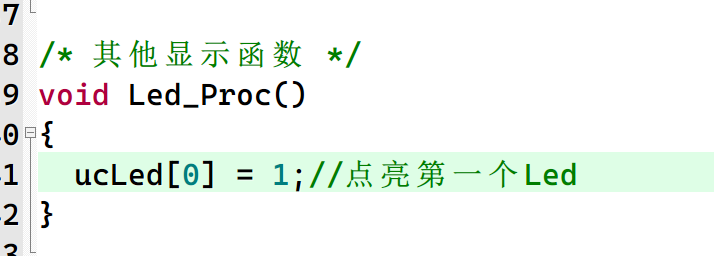
8.主函数：一般固定不变



二、Led模块的基本使用

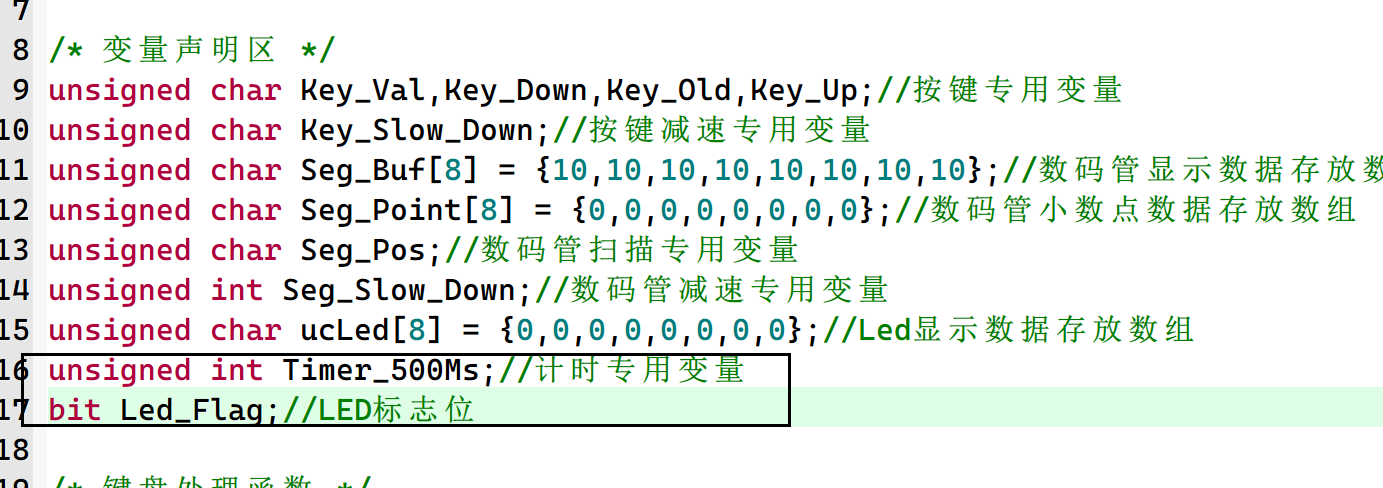
1.点亮一个Led

ucLed[需要点亮的位（0-7）] = 0（灭）/ 1（亮）



2.闪烁一个Led

第一步：定义两个变量 一个用于计时 一个充当标志位



常见定义类型：

Unsigned char:0-255

Unsigned int:0-65535

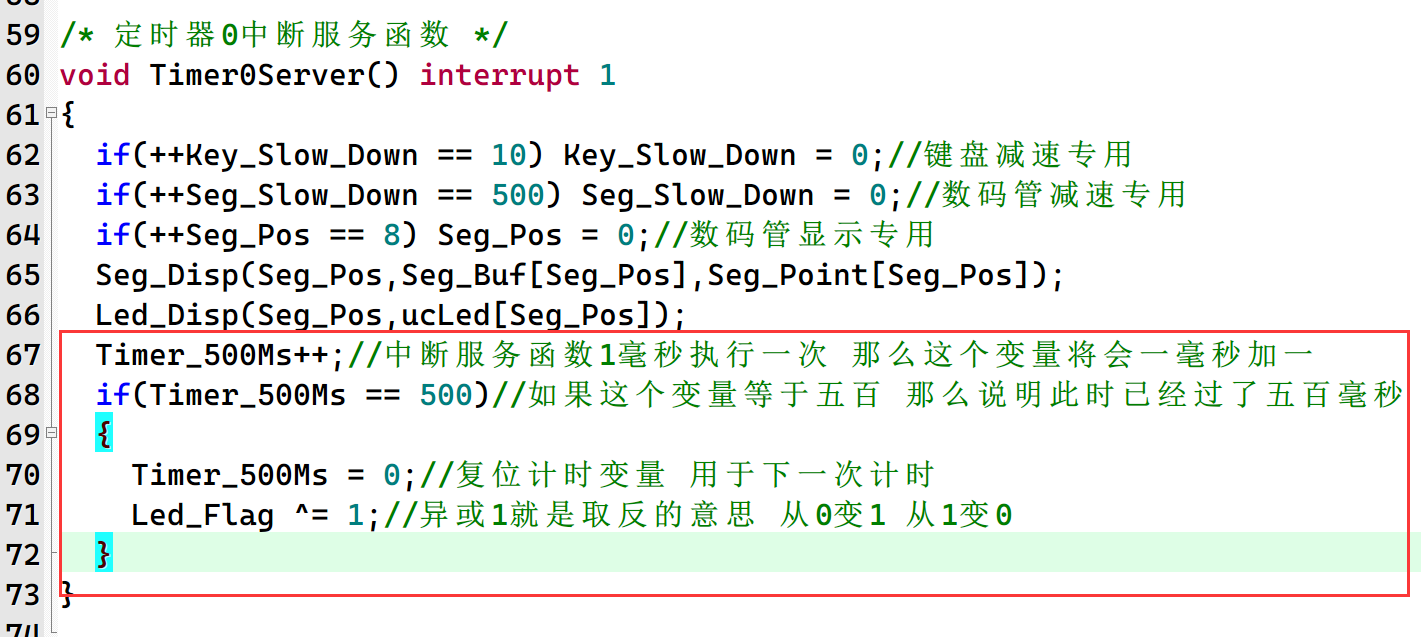
Float:小数

Bit:不是0就是1

定义原则：尽量用占用小的变量类型 不要太浪费资源

定义方式：[变量类型]【空格】[变量名称（自己自定义）] （【=】【初始值】）【;】（如果没有给初值 那么默认为0）

第二步：在中断服务函数中编写代码



If语句：

If（判断条件）

{

//条件成立执行这个括号的内容

}

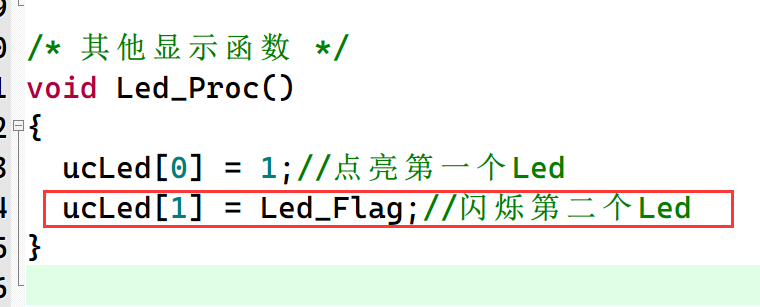
Else

{

//条件不成立执行这个括号的内容

}

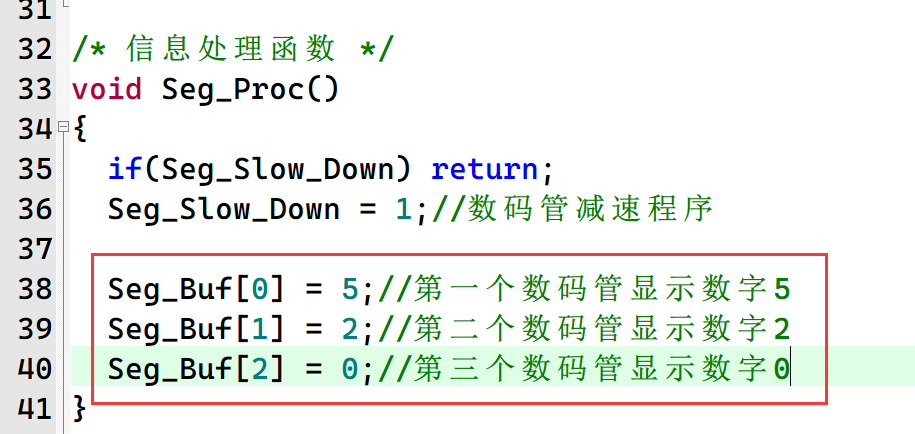
第三步：在Led\_Proc内编写代码



三、Seg模块的基本使用

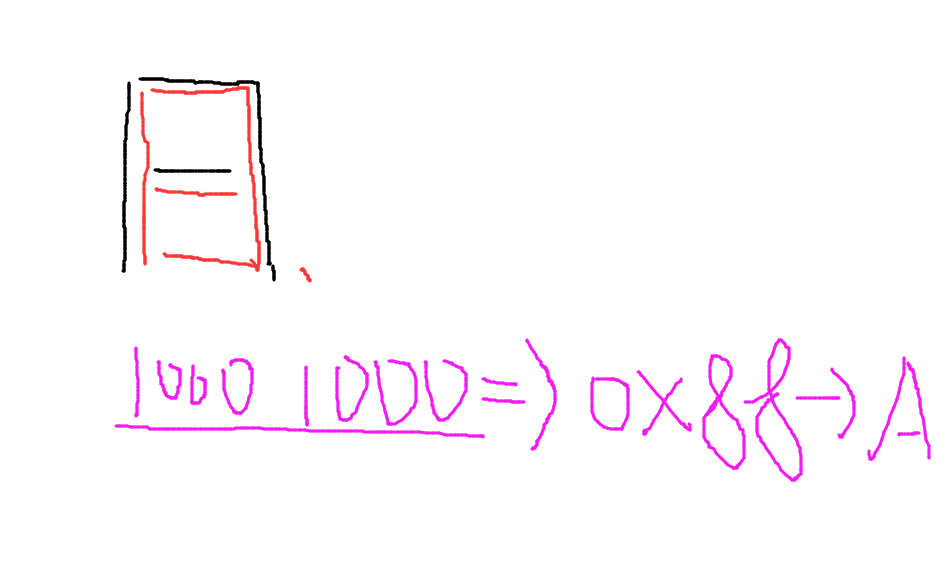
1.显示固定数字

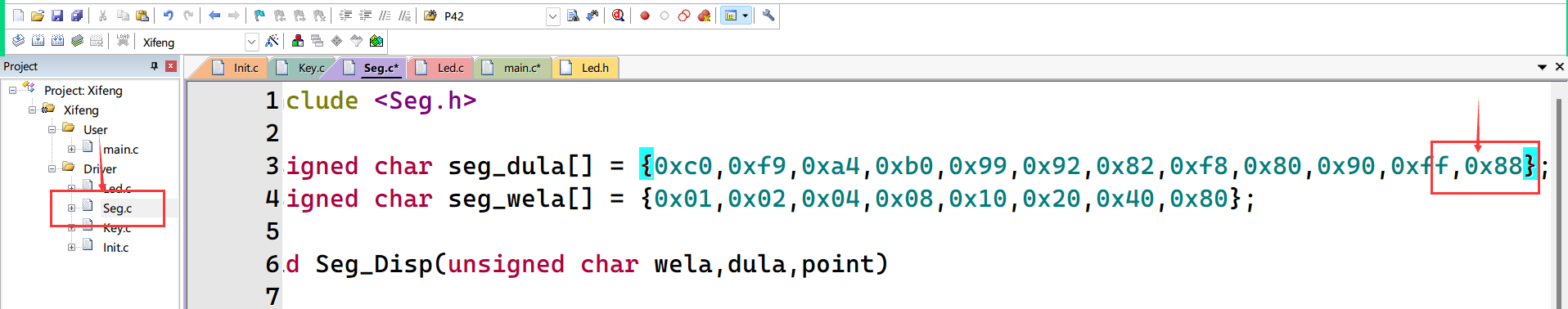
Seg\_Buf[需要显示的数码管（0-7）] = 0-9（数字）



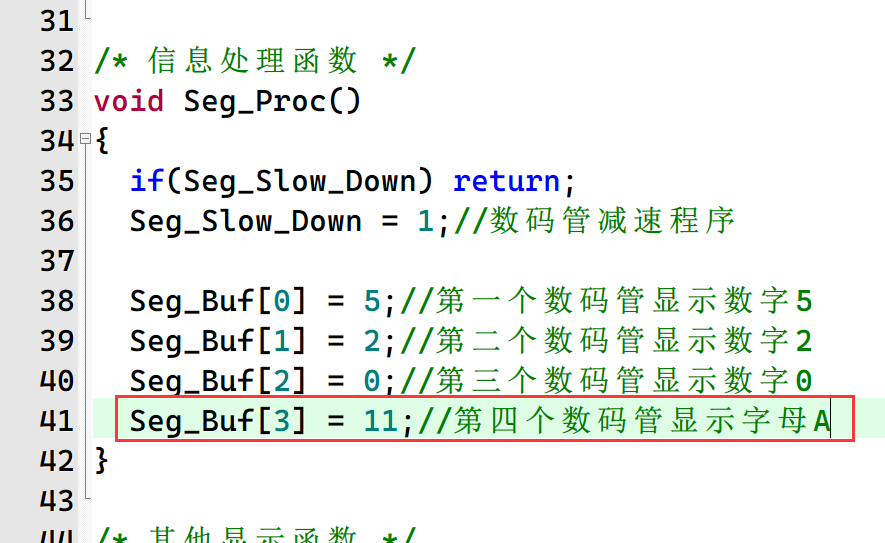
2.显示固定字母

第一步：推断字母段码后在底层内添加



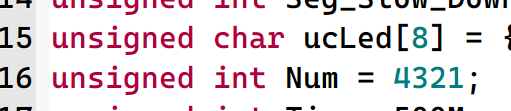


第二步：显示字母

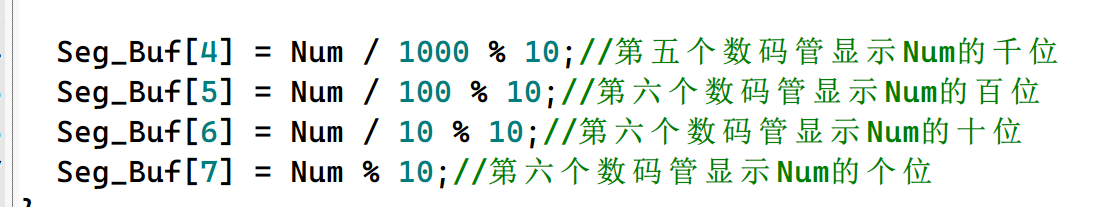


3.显示变量

第一步：定义变量



第二步：显示变量

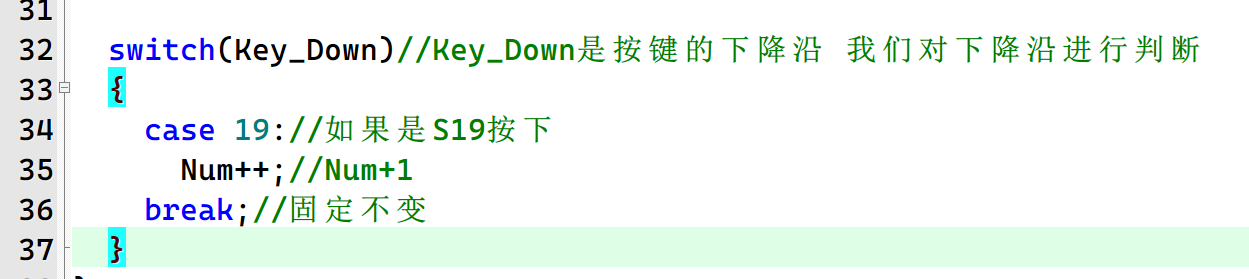


Num的第X位 = Num / 1（后面X个0） % 10

个位是 / 1 故可以省略不写

四、Key模块的基本使用

1.按下某个按键执行某个功能



Switch（Key\_Down(下降沿-按下) / Key\_Up(上升沿-松手)）

{

Case 某一个按键数字：

//执行语句

Break;

}