

第四十八节： while 循环语句。

【48.1 程序的“跑道”。】

经常听到这句话“程序跑起来了吗？”，程序在哪里跑？有跑道吗？有的。循环语句就像一条椭圆的跑道，程序在跑道上不停的跑，不知疲倦的跑，永无止境，一秒钟几百万圈的速度。单片机的 main 主函数内往往有一条 while(1) 语句，这就是单片机的“循环跑道”，称之为主循环，主循环内还可以继续嵌套多层 while 循环语句。

【48.2 while 循环的常见格式。】

常见格式如下：

```
while(条件)
{
    语句 1;
    语句 2;
    .....
    语句 N;
}
语句 N+1;
```

上面的“花括号内”称为“循环体内”，“花括号外”称为“循环体外”，现在来分析一下上述代码的执行规律，如下：

(1) 像 if 语句一样，先判断 while 的(条件)是否为真。如果为“假”，就不执行循环体“内”的“语句 1”至“语句 N”，直接跳到循环体“外”的“语句 N+1”处开始往下执行。如果为“真”，才执行循环体“内”的“语句 1”至“语句 N”，当执行完循环体“内”最后的“语句 N”时，单片机会突然返回到第一行代码“while(条件)”处，继续判断循环的(条件)是否为真，如果为假就跳到循环体“外”的“语句 N+1”，表示结束了当前循环。如果为真就继续从“语句 1”执行到“语句 N”，然后再返回 while(条件)处，依次循环下去，直到条件为假时才罢休，否则一直循环下去。

(2) while(条件)语句中，条件判断真假的规则跟 if 语句一模一样，有两种类型：一种是纯常量或者变量类型的，只要此数值不等于 0 就认为是真，所以 while(1)也称死循环语句，因为里面的条件永远不为 0。对于死循环这样的语句，如果不遇到 break, return, goto 这些语句，那么就永远也别想跳出这个循环；另外一种关系判断，以及关系语句之间的像“与或”关系这类的判断。这些条件判断的真假，跟 if 语句的规则是一样的，这里不再多讲。break, return, goto 这些语句后面章节会讲到。

【48.3 while 省略花括号，没带分号。】

```
while(条件)
    语句 1;
    语句 2;
    .....
    语句 N;
    语句 N+1;
```

上面的代码，居然没有了花括号，问题来了，此循环语句的“有效射程”究竟是多远，或者说，此循环语句的循环区域在哪里。现在跟大家解开这个谜团。第一行代码，while(条件)后面“没有分号”，接着第二行就是“语句 1”，所以，这种情况跟 if 语句省略花括号的写法是一样的，此时循环体默认只包含离它最近的一条且仅仅一条的“语句 1”，因此，上述的语句，等效于下面这种添加花括号的写法：

```
while(条件)
{
    语句 1;
}
语句 2;
.....
语句 N;
语句 N+1;
```

【48.4 while 省略花括号，带分号。】

```
while(条件);
    语句 1;
    语句 2;
    .....
    语句 N;
    语句 N+1;
```

这次的代码跟刚才“48.3”的代码唯一的差别是，第一行代码，while(条件)后面“有分号”。所以它循环的有效范围就在第一行就结束了，不涉及“语句 1”。此时，等效于下面这种添加花括号的写法：

```
while(条件)
{
    ;    //这里的分号代表一条空语句
}
语句 1;
语句 2;
.....
语句 N;
语句 N+1;
```

如果 while 的（条件）一直为“真”，单片机就一直在循环体内执行一条“无意义”的空语句，相当于“耗着”的状态，执行不到后面“语句 1”的语句，除非，条件为“假”才罢休才会跳出循环体。

循环体内什么都没有，只写一条“空语句”，这种写法在实际项目中也是有用武之地的，比如，等待某件事是否满足条件，如果不满足，就一直死等死磕在这里，其它事情都干不了，这种“死等死磕”的做法，专业术语叫“阻塞”，与之反面相对应的是另外一个词叫“非阻塞”。对于循环的“阻塞”用法，老练的工程师通常会多加一个超时的判断，这些内容大家暂时不用深入了解，后续章节我会讲到。

【48.5 例程练习和分析。】

现在编写一个程序来熟悉一下 while 语句的书写和使用格式。

程序代码如下：

```

/*---C 语言学习区域的开始。-----*/

unsigned char a=0; //观察这个数最后的变化
unsigned char b=0; //观察这个数最后的变化

unsigned char i;    //控制循环体的条件判断变量

void main() //主函数
{
    i=3;
    while(i) //i 不断减小，直到变为 0 时才跳出此循环体
    {
        a=a+1; //当 i 从 3 减少到 0 的时候，这条语句被循环执行了 3 次。
        i=i-1; //循环的条件不断发生变化，不断减小
    }

    i=0;
    while(i<3) //i 不断增大，当 i 大于或者等于 3 时才跳出此循环体
    {
        b=b+2; //当 i 从 0 增加到 3 的时候，这条语句被循环执行了 3 次。
        i=i+1; //循环的条件不断发生变化，不断增加
    }

    View(a); //把第 1 个数 a 发送到电脑端的串口助手软件上观察。
    View(b); //把第 2 个数 b 发送到电脑端的串口助手软件上观察。

    while(1)
    {
    }
}

/*---C 语言学习区域的结束。-----*/

```

在电脑串口助手软件上观察到的程序执行现象如下：

开始...

第 1 个数

十进制:3

十六进制:3

二进制:11

第 2 个数

十进制:6

十六进制:6

分析:

变量 a 为 3。a 初始化为 0，进入循环体内后，a 每次加 1，循环加 3 次，因此从 0 变成了 3。

变量 b 为 6。b 初始化为 0，进入循环体内后，b 每次加 2，循环加 3 次，因此从 0 变成了 6。

【48.6 如何在单片机上练习本章节 C 语言程序？】

直接复制前面章节中第十一节的模板程序，练习代码时只需要更改“C 语言学习区域”的代码就可以了，其它部分的代码不要动。编译后，把程序下载进带串口的 51 学习板，通过电脑端的串口助手软件就可以观察到不同的变量数值，详细方法请看第十一节内容。