

第十五节：二进制与十六进制。

【15.1 十六进制是二进制的缩写。】

在我的印象中，C51 编译器好像并不支持二进制的书写格式，即使它能支持二进制的书写格式，二进制的书写还是有个弊端，就是数字太多太长了，写起来非常费劲不方便，怎么办？解决办法就是用十六进制。十六进制是二进制的缩写，之所以称它为二进制的缩写，是因为它们的转换关系非常简单直观，不需要借助计算器即可相互转换。

【15.2 何谓十六进制？】

何谓十六进制？欲搞清楚这个问题，还得先从十进制说起。所谓十进制，就是用一位字符可以表示从 0 到 9 这十个数字。所谓二进制，就是用一位字符可以表示从 0 到 1 这二个数字。所谓十六进制，当然也就是用一位字符可以表示从 0 到 15 这十六个数字。但是十六进制马上就会面临一个问题，十六进制的 10 到 15 这 6 个数其实是有两位字符组成的，并不是一位呀？于是 C 语言用这些字符 A, B, C, D, E, F 分别替代 10, 11, 12, 13, 14, 15 这 6 个数，10 前面的 0 到 9 还是跟十进制的字符一致。A, B, C, D, E, F 也可以用小写 a, b, c, d, e, f 来替代，在数值上不区分大小写，比如十六进制的 a 与 A 都是表示十进制的 10。

【15.3 二进制与十六进制是如何转换的？】

前面提到了十六进制是二进制的缩写，它们的转换关系非常简单直观，每 1 位十六进制的字符，对应 4 位二进制的字符。关系如下：

| 十进制 | 二进制 | 十六进制 |
|-----|------|------|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

二进制转换成十六进制的时候，如果不是 4 位的倍数，则最左边高位默认补上 0 凑合成 4 位的倍数。比如一个二进制的数 101001，可以在左边补上 2 个 0 变成 00101001，然后把每 4 位字符转成 1 个十六进制的字符。左边高 4 位 0010 对应十六进制的 2，右边低 4 位 1001 对应十六进制的 9，所以二进制的 101001 合起

来最终转换成十六进制的数是 29（实际上正确的写法是 0x29，为什么？请继续往下看。）。

【15.4 十六进制数的标准书写格式是什么样子的？】

十六进制的标准书写格式是什么样子的？实际上，十六进制 29 并不能直接写成 29，否则就跟十进制的写法混淆了。为了把十六进制和十进制的书写格式进行区分，C 语言规定凡是十六进制必须加一个数字 0 和一个字母 x 作为前缀，也就是十六进制必须以 0x 作为前缀，所以刚才的十六进制 29 就应该写成 0x29，否则，如果直接写 29 编译器会认为是十进制的 29，而十进制的 29 转换成十六进制是 0x1D（十进制与十六进制之间如何转换在后面章节会讲到），0x29 与 0x1D 可见差别很大的，凡是不加前缀的都会被默认为十进制。多说一句，在 C 语言程序里，对于同样一个数值，既可以用十六进制，也可以用十进制，比如：d=0x2C 与 d=44 的含义是一样的，因为十六进制的 0x2C 和十进制的 44 最终都会被 C51 编译器翻译成二进制 00101100，是表示同样大小的数值。

【15.5 例程练习和分析。】

现在我们编写一个程序来观察十六进制和二进制的关系。

程序代码如下：

```
/*---C 语言学习区域的开始。-----*/

void main() //主函数
{
    unsigned char a;    //定义一个变量 a，并且分配了 1 个字节的 RAM 空间。
    unsigned char b;    //定义一个变量 b，并且分配了 1 个字节的 RAM 空间。
    unsigned char c;    //定义一个变量 c，并且分配了 1 个字节的 RAM 空间。
    unsigned char d;    //定义一个变量 d，并且分配了 1 个字节的 RAM 空间。

    a=0x06;    //十六进制前记得加 0x 前缀，超过 9 部分的字母不分大小写。
    b=0x0A;    //十六进制前记得加 0x 前缀，超过 9 部分的字母不分大小写。
    c=0x0e;    //十六进制前记得加 0x 前缀，超过 9 部分的字母不分大小写。
    d=0x2C;    //十六进制前记得加 0x 前缀，超过 9 部分的字母不分大小写。

    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。
    View(c);    //把第 3 个数 c 发送到电脑端的串口助手软件上观察。
    View(d);    //把第 4 个数 d 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:6
十六进制:6
二进制:110

第 2 个数
十进制:10
十六进制:A
二进制:1010

第 3 个数
十进制:14
十六进制:E
二进制:1110

第 4 个数
十进制:44
十六进制:2C
二进制:101100
```

分析：

通过实验结果，我们知道二进制与十六进制的转换关系确实非常清晰简单，所以十六进制也可以看作是二进制的缩写。

【15.6 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。