第十四节：二进制与字节单位，以及常用三种变量的取值范围。

【14.1　为什么要二进制？】

为什么要二进制？我们日常生活明明是十进制的，为何数字电子领域偏要选择二进制？这是由数字硬件电路决定的。人有十个手指头，人可以直接发出十种不同声音来命名 0,1,2,3...9 这些数字，人可以直接用眼睛识别出十种不同状态的信息，但是数字底层基础硬件电路要直接处理和识别十种状态却很难，相对来说，处理和识别两种状态就轻松多了，所以选择二进制。比如，一颗 LED 灯的亮或灭，一个 IO 口的输出高电平或低电平，识别某一个点的电压是高电平或低电平，只需要三极管等基础元器件就可把硬件处理电路搭建起来，二进制广泛应用在数字电路的存储，通讯和运算等领域，想学好单片机就必须掌握它。

【14.2　二进制如何表示成千上万的大数值？】

二进制如何表示成千上万的数值？现在用 LED 灯的亮和灭来跟大家讲解。

（1）1 个 LED 灯：

灭　　第 0 种状态
亮　　第 1 种状态

合计：共 2 种状态。

（2）2 个 LED 灯挨着：

灭灭　　第 0 种状态
灭亮　　第 1 种状态
亮灭　　第 2 种状态
亮亮　　第 3 种状态

合计：共 4 种状态。

（3）3 个 LED 灯挨着：

灭灭灭　　第 0 种状态
灭灭亮　　第 1 种状态
灭亮灭　　第 2 种状态
灭亮亮　　第 3 种状态
亮灭灭　　第 4 种状态
亮灭亮　　第 5 种状态
亮亮灭　　第 6 种状态
亮亮亮　　第 7 种状态

合计：共 8 种状态。

（4）8 个 LED 灯挨着：

灭灭灭灭灭灭灭灭　　第 0 种状态
灭灭灭灭灭灭灭亮　　第 1 种状态
.....　　第 N 种状态
亮亮亮亮亮亮亮灭　　第 254 种状态
亮亮亮亮亮亮亮亮　　第 255 种状态

合计：共 256 种状态。

（5）16 个 LED 灯挨着：

灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭	第 0 种状态
灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭亮	第 1 种状态
.....	第 N 种状态
亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮灭	第 65534 种状态
亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮	第 65535 种状态

合计：共 65536 种状态。

（6）32 个 LED 灯挨着：

灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭	第 0 种状态
灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭灭亮	第 1 种状态
.....	第 N 种状态
亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮灭	第 4294967294 种状态
亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮亮	第 4294967295 种状态

合计：共 4294967296 种状态。

结论:

连续挨着的 LED 灯越多，能表达的数值范围就越大。

【14.3　什么是位？】

什么是位？以上一个 LED 灯就代表 1 位，8 个 LED 灯就代表 8 位。位的英文名是用 bit 来表示。一个变量的位数越大就意味着这个变量的取值范围越大。一个单片机的位数越大，就说明这个单片机一次处理的数据范围就越大，意味着运算和处理速度就越快。我们日常所说的 8 位单片机，32 位单片机，就是这个位的概念。为什么 32 位的单片机比 8 位单片机的处理和运算能力强，就是这个原因。

【14.4　什么是字节？】

什么是字节？字节是计算机很重要的一个基本单位，一个字节有 8 位。8 个 LED 灯挨着能代表多少种状态，就意味着一个字节的数据范围有多大。从上面举的例子中，我们知道 8 个 LED 灯挨着，能表示从 0 到 255 种状态，所以一个字节的取值范围就是从 0 到 255。

【14.5　三种常用变量的取值范围是什么？】

前面章节曾提到三种常用的变量：unsigned char，unsigned int ,unsigned long。现在有了二进制和字节的基础知识，就可以跟大家讲讲这三种变量的取值范围，而且很重要，这是我们写单片机程序必备的概念。

unsigned char 的变量占用 1 个字节 RAM，共 8 位，根据前面 LED 灯的例子，取值范围是从 0 到 255。

unsigned int 的变量占用 2 个字节 RAM，共 16 位，根据前面 LED 灯的例子，取值范围是从 0 到 65535。多说一句，对于 51 内核的单片机，unsigned int 的变量是占用 2 个字节。如果是在 32 位的 stm32 单片机，unsigned int 的变量是占用 4 个字节的，所以不同的单片机不同的编译器是会有一些差异的。

unsigned long 的变量占用 4 个字节 RAM，共 32 位，根据前面 LED 灯的例子，取值范围是从 0 到 4294967295。

【14.6　例程练习和分析。】

现在我们编写一个程序来验证 unsigned char，unsigned int，unsigned long 的取值范围。

定义两个 unsigned char 变量 a 和 b，a 赋值 255，b 赋值 256，255 和 256 恰好处于 unsigned char 的取值边界。

再定义两个 unsigned int 变量 c 和 d，c 赋值 65535，d 赋值 65536，65535 和 65536 恰好处于 unsigned int 的取值边界。

最后定义两个 unsigned long 变量 e 和 f，e 赋值 4294967295，f 赋值 4294967296，4294967295 和 4294967296 恰好处于 unsigned long 的取值边界。

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 int c;    //定义一个变量 c，并且分配了 2 个字节的 RAM 空间。
    unsigned int d;    //定义一个变量 d，并且分配了 2 个字节的 RAM 空间。
    unsigned long e;   //定义一个变量 e，并且分配了 4 个字节的 RAM 空间。
    unsigned long f;   //定义一个变量 f，并且分配了 4 个字节的 RAM 空间。

    a=255;          //把 255 赋值给变量 a，a 此时会是什么数？会超范围溢出吗？
    b=256;          //把 256 赋值给变量 b，b 此时会是什么数？会超范围溢出吗？
    c=65535;        //把 65535 赋值给变量 c，c 此时会是什么数？会超范围溢出吗？
    d=65536;        //把 65536 赋值给变量 d，d 此时会是什么数？会超范围溢出吗？
    e=4294967295;  //把 4294967295 赋值给变量 e，e 此时会是什么数？会超范围溢出吗？
    f=4294967296;  //把 4294967296 赋值给变量 f，f 此时会是什么数？会超范围溢出吗？

    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。
    View(e);   //把第 5 个数 e 发送到电脑端的串口助手软件上观察。
    View(f);   //把第 6 个数 f 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:255
十六进制:FF
二进制:11111111

第 2 个数
十进制:0
十六进制:0
二进制:0

第 3 个数
十进制:65535
十六进制:FFFF
二进制:1111111111111111

第 4 个数
十进制:0
十六进制:0
二进制:0

第 5 个数
十进制:4294967295
十六进制:FFFFFFFF
二进制:11111111111111111111111111111111

第 6 个数
十进制:0
十六进制:0
二进制:0
```

分析：

通过实验结果，我们知道 unsigned char 变量最大能取值到 255，如果非要赋值 256 就会超出范围溢出后变成了 0。unsigned int 变量最大能取值到 65535，如果非要赋值 65536 就会超出范围溢出后变成了 0。unsigned long 变量最大能取值到 4294967295，如果非要赋值 4294967296 就会超出范围溢出后变成了 0。

【14.7　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。