

第八十一节： 时间和速度的起源（指令周期和晶振频率）。

【81.1 节拍。】

单片机的 C 语言经过 C 编译器后，翻译成很多条机器指令，单片机逐条执行这些指令，每执行一条指令都是按照固定的节奏进行的，两条指令之间是存在几乎固定的时间间隔（实际上不是所有指令的间隔时间都绝对一致，这里方便理解暂时看作是一致），这就是节拍，每个节拍之间的时间间隔其实就是指令周期，因此，指令周期越短，节拍就越短，单片机的运算速度就越快。指令周期是由什么决定的呢？指令周期是由“心跳速度”和“心跳个数”决定的。指令周期都是由固定的 N 个“心跳个数”组成的，指令周期到底由多少个“心跳个数”组成？每种单片机每类指令各不一样。我们用的 51 系列单片机，最短的单周期指令是由 12 个“心跳个数”组成，依次类推，双周期指令由 24 个“心跳个数”组成，4 周期指令由 48 个“心跳个数”组成。但是光有“心跳个数”还不够，还必须搭配知道“心跳速度”才能最终计算出指令周期。这里的“心跳速度”就是晶振的频率，“心跳个数”就是累计晶振的起振次数。比如，假设我们用的 51 单片机是 12MHz（本教程实际用的是 11.0592MHz），那么每个单周期的指令执行的时间是： $12 \times (1/12000000)$ 秒=1 微秒。这个公式左边的“12”代表“12 个晶振起振的次数”，这个公式右边的“ $(1/12000000)$ ”代表晶振每起振 1 次所需要的单位时间。二者结合，刚好就是“心跳个数”乘以“单个心跳周期”等于指令周期，而指令周期就是节拍的时间。

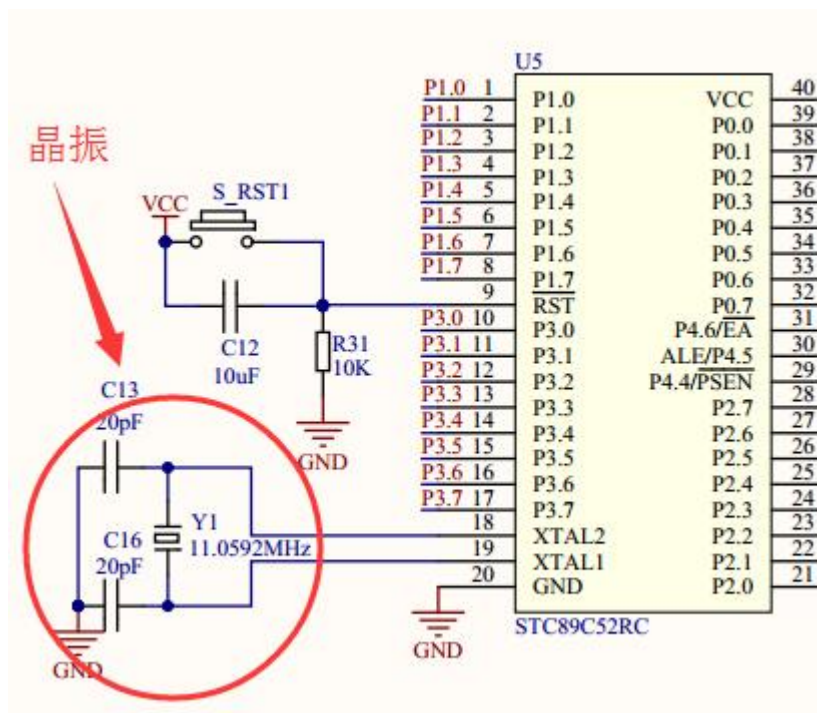


图 81.1.1 单片机的晶振

【81.2 累计节拍次数产生延时时间。】

有了这个最原始的“节拍”概念，现在开始编写一个练习程序，让一个 LED 灯闪烁，闪烁的本质，就是让一个 LED 灯先亮一会（“一会”就是延时），然后紧接着让 LED 灯熄灭一会（“一会”就是延时），依次循环，在视觉上看到的连贯动作就是 LED 闪烁。这里的关键是如何产生这个“一会”的延时，本节教程所用的就是一个 for 循环来执行 N 条空指令，每执行一条空指令就需要消耗掉 1 个左右的指令周期的时间（大概 1 微秒左右），空指令执行的循环次数越多，产生的延时时间就越长。例子如下：

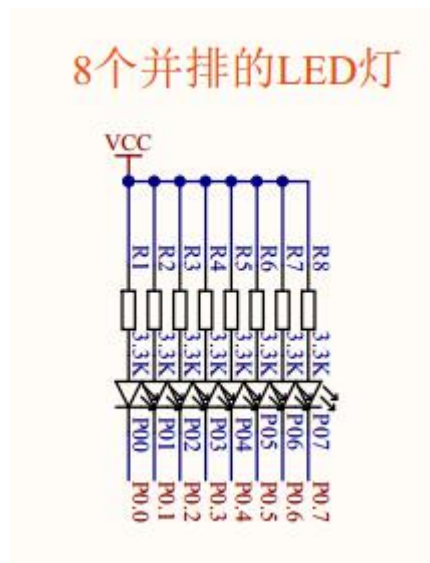


图 81.2.1 灌入式驱动 8 个 LED

```
#include "REG52.H"
sbit P0_0=P0^0; //利用 sbit 和符号“^”的组合，把变量名字 P0_0 与 P0.0 引脚关联起来
unsigned long i; //for 循环用的累计变量
//unsigned int i; //如果把 for 循环的变量 i 改成 unsigned int 类型，闪烁的频率会加快。
void main()
{
    while(1)
    {
        //第（1）步
        P0_0=0; //LED 灯亮。

        //第（2）步
        for(i=0;i<5000;i++) //累计的循环次数越大，这里的延时就越长，“亮”持续的时间就越长。
        {
            ; //分号代表一条空指令
        }

        //第（3）步
        P0_0=1; //LED 灯灭。

        //第（4）步
        for(i=0;i<5000;i++) //累计的循环次数越大，这里的延时就越长，“灭”持续的时间就越长。
        {
            ; //分号代表一条空指令
        }

        //第（5）步：这里已经触碰到主循环 while(1)的“底线”，所以接着跳转到第（1）步继续循环
    }
}
```

```
}  
}
```

现象分析：

理论上，每执行 1 条指令大概 1 微秒左右，但是实际上，我们看到的实验现象，发现累计循环才 5000 次，按理论计算，应该产生 0.005 秒左右的延时才合理，但是实际上居然能产生类似 0.5 秒的闪烁效果，中间相差 100 倍！为什么？C 语言跟机器指令之间是存在翻译的“中间商”环节，一条 C 指令并不代表一条机器指令，往往一条 C 指令翻译后产生 N 条机器指令，比如上面的代码，用到 for 循环变量 i，用的是 unsigned long 变量，意味 4 个字节，即使一条 C 语言赋值指令估计可能也要消耗 4 条单周期指令，在加上 for 循环的判断指令，和累加指令，以及跳转指令，所以我们看到的 for(i=0;i<5000;i++) 并不代表是真正仅仅执行了 5000 个指令周期，而是有可能执行了 500000 条指令周期！假如我们把上述代码中的 i 改成 unsigned int 变量（2 字节），是会看到闪烁的速度明显加快的，其中原因就是 C 编译器与机器指令之间存在翻译后的“1 对 N”的关系。