程序执行所需要的时间：

[指令周期](https://baike.baidu.com/item/%E6%8C%87%E4%BB%A4%E5%91%A8%E6%9C%9F)（Instruction Cycle）：取出并执行一条指令的时间。[1] 

[总线周期](https://baike.baidu.com/item/%E6%80%BB%E7%BA%BF%E5%91%A8%E6%9C%9F)（BUS Cycle）：也就是一个访存储器或I/O端口操作所用的时间。[1]

[时钟周期](https://baike.baidu.com/item/%E6%97%B6%E9%92%9F%E5%91%A8%E6%9C%9F)（Clock Cycle）：又称震荡周期，是处理操作的最基本单位。(晶振频率的[1]  倒数）[1]

[指令周期](https://baike.baidu.com/item/%E6%8C%87%E4%BB%A4%E5%91%A8%E6%9C%9F)、[总线周期](https://baike.baidu.com/item/%E6%80%BB%E7%BA%BF%E5%91%A8%E6%9C%9F)和[时钟周期](https://baike.baidu.com/item/%E6%97%B6%E9%92%9F%E5%91%A8%E6%9C%9F)之间的关系：一个指令周期由若干个总线周期组成，而一个总线周期时间又包含有若干个时钟周期。[1]

根据正点原子的说法：STM32有三级流水线，指令周期不定，arm给出的是1.25MIPS/Mhz，一个平均执行速度。

所以使用72mhz的情况下，每秒执行72m\*1.25条单指令。

参考资料：<https://blog.csdn.net/fantasy_wxe/article/details/42838571>

根据51单片机的程序进行类比。

51的500ms延时函数：

void delay500ms(void)

{

unsigned char i,j,k;

for(i=15;i>0;i--)

for(j=202;j>0;j--)

for(k=81;k>0;k--);

}

一层循环n:R5\*2 = 81\*2 = 162周期 DJNZ 2机器周期

二层循环m:R6\*(n+3) = 202\*165 = 33330周期 DJNZ 2机器周期 + R5赋值 1周期 = 3周期

三层循环: R7\*(m+3) = 15\*33333 = 499995周期 DJNZ 2周期 + R6赋值 1周期 = 3周期

循环外: 5周期 子程序调用 2周期 + 子程序返回 2周期 + R7赋值 1周期 = 5周期

延时总时间 = 三层循环 + 循环外 = 499995+5 = 500000周期

计算公式:延时时间=[(2\*R5+3)\*R6+3]\*R7+5

涉及到对arm芯片的汇编语言的指令的指令周期，近似认为大多数指令为单周期。

单循环和多循环看不出明显的差别，同时近似认为所有指令为单指令情况。

则

void delay\_ms(time\_ms)

{

time\_count = time\_ms\*72000\*1.25-3;//每毫秒执行的指令数为72000\*1.25，for循环内计数需要减去子程序调用，返回，和赋值所占用的指令共3条

for(;time\_count>0; time\_count--);

}

void delay\_us(time\_us)

{

time\_count = time\_ms\*72\*1.25-3;//每微秒执行的指令数为72\*1.25，for循环内计数需要减去子程序调用，返回，和赋值所占用的指令共3条

for(;time\_count>0; time\_count--);

}

在f1开发板上测试，从函数开始到结束，右下角T1从0.0005到81.0005，经过约81sec，结果不太正确，需要改进。