

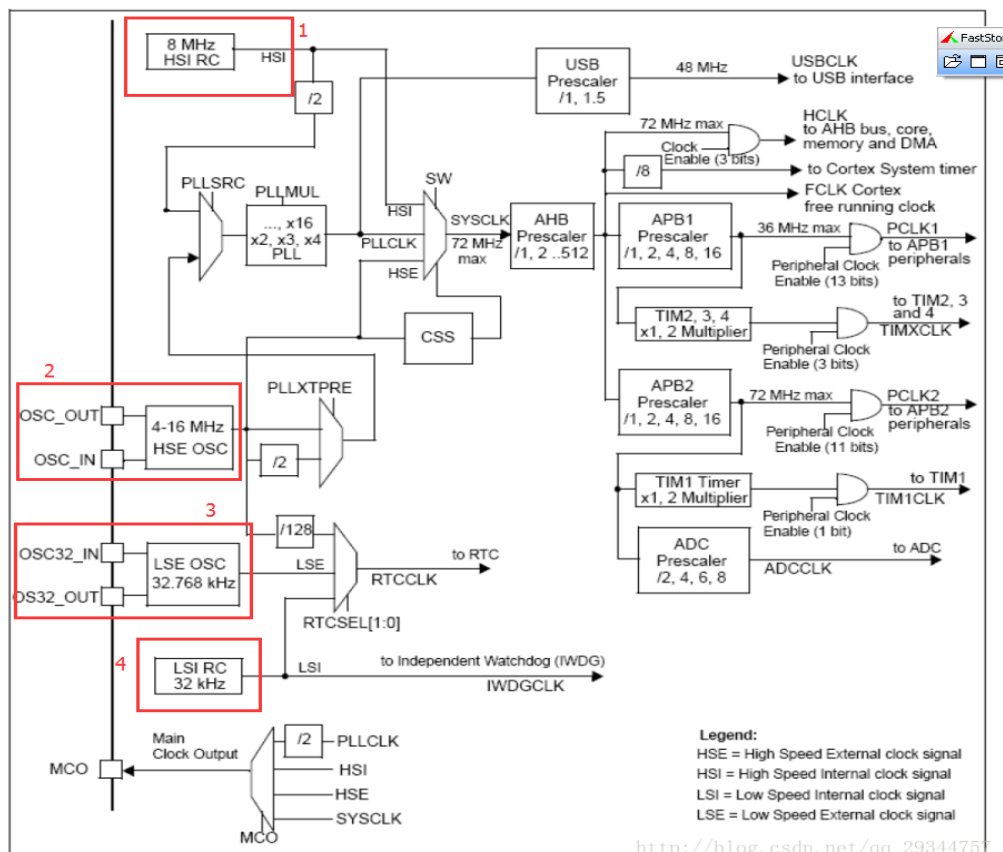
## • UART 串口通信 •

在之前 51 的学习中，我们也曾接触过有关 UART 的相关知识。UART 应该是我们接触到的第一种通讯方式。为了减少讲解的重复性，相关的代码与学习方法，我们就放在之后的资料中，想要了解的同学选择进行下载即可。关于串口助手的软件，我们也会在后续的资料总提供。

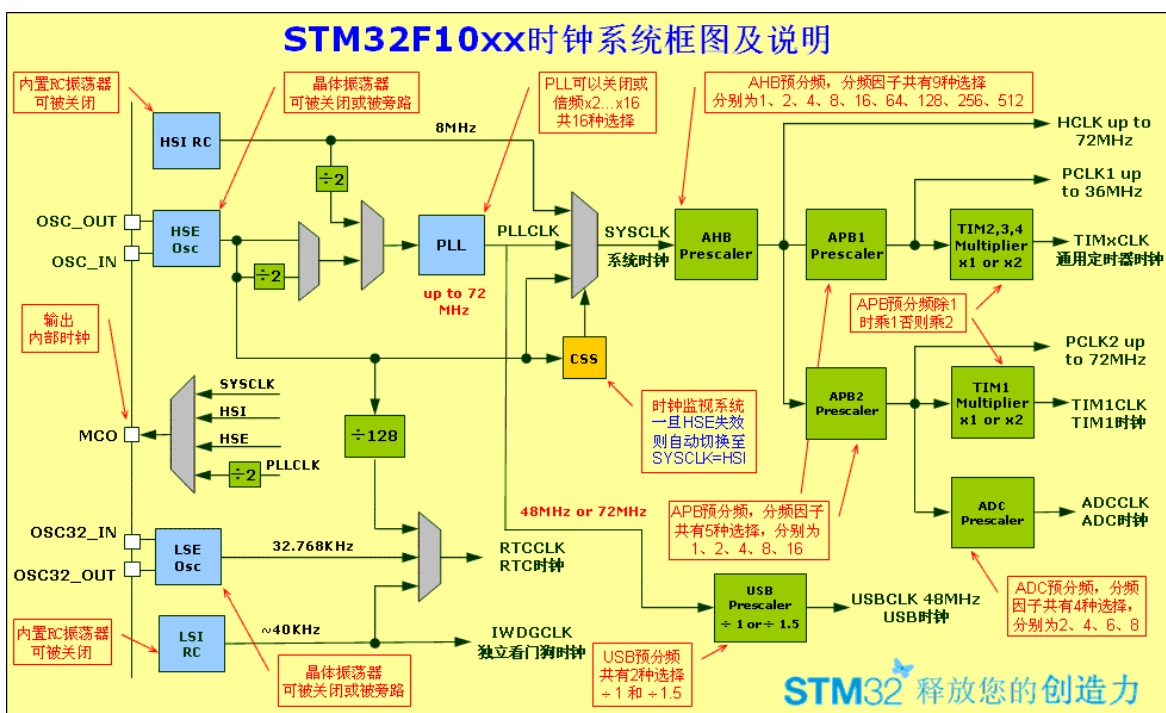
## • 时钟与定时器中断 •

对于单片机而言，最重要的部分莫过于时钟。好比人的心脏需要规律的跳动人才能正常的生活，单片机需要时钟提供精准的工作频率来读取代码与执行各种操作以完成正常的工作。在之前的讲解中，我们提到过 32 可以通过开启时钟来赋予各项外设工作状态，也通过此方式节省不必要消耗的能量。对于 32 而言，有关时钟部分的理解对于初学者而言可以说是比较复杂。我自己在之前的学习中也未曾深入了解过，不过作为入门指南，希望我的讲解能带给大家一些启示。

首先，我们先来看看 stm32 手册里给出的“时钟树”。图中表明了四处地方，可以理解为我们 32 内部的时钟源，即产生时钟的源头。在 51 单片机中，我们通常外接一个 11.0592MHz 的晶振，再利用 RC 时钟电路来产生时钟。同理，在 32 中也是由 RC 时钟电路产生时钟源，不过根据 RC 电路的位置，可将 32 的时钟源分为内部时钟电路，外部时钟电路，以及内外部时钟电路。



(P 2-4-81) 时钟树



(P 2-4-82) 带有部分讲解的时钟树

#### (1) 内部时钟电路：

晶体振荡器和 RC 时钟电路都在 STM32 芯片内部，如图中标注 1、标注 4。

标注 1 处是产生 8MHz 的时钟源，称为 HSI，高速内部时钟源（H 意为高速，S 意为源，I 意为内部）；

标注 4 处是产生 32KHz 的时钟源，称为 LSI，低速内部时钟源；

#### (2) 内外部时钟电路：

晶体振荡器在 STM32 芯片外部，RC 时钟电路在 STM32 芯片内部，如图中标注 2、标注 3。

标注 2 处是产生 4-16MHz 的时钟源，称为 HSE，高速外部时钟源；

标注 3 是产生 32.768KHz 的时钟源，称为 LSE，低速外部时钟源；

OSC\_OUT 和 OSC\_IN、OSC32\_OUT 和 OSC32\_IN 分别接晶振的两个引脚。前者一般接 8MHz 晶振；后者一定接 32.768KHz，因为这个时钟源是供给 RTC 实时时钟使用的。在 51 单片机中没有集成 RTC 模块，在做电子时钟时用到的 DS1302 集成芯片时，也是为其提供的也是 32.768KHz 的晶振。

#### (3) 外部时钟电路

晶体振荡电路和 RC 时钟电路都在 STM32 芯

片外部。如图中标注 2、标注 3。

OSC\_OUT 和 OSC\_IN、OSC32\_OUT 和 OSC32\_IN 除了分别接晶振的两个引脚，对于 OSC\_IN 和 OSC32\_IN 引脚，还可以接入外部的 RC 时钟电路，其时钟源直接由外部供给，不过这种方案少见。

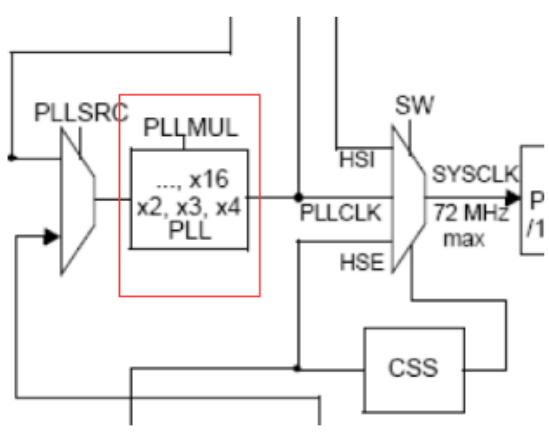
综上所述，STM32 的时钟源有 4 个：HSI、HSE 和 LSI、LSE。

HSI 时钟源，它是在由 STM32 在内部用 RC 振荡电路实现的高速内部时钟源。HSI RC 振荡器能够在不需要任何外部期间的条件下提供系统时钟，它的启动时间比 HSE 晶体振荡器短，但是不精准，即使在校准之后它的时钟频率精度仍较差。在手册中还明确说，当 HSI 被用作 PLL 时钟输入时，系统时钟能得到的最大频率是 61MHz，这显然不能发挥 STM32 最极致的性能。虽然 HSI 不精准，但是鉴于启动速度原因考虑，STM32 上电复位，默认是采用 HSI 时钟源的，当然开发者可以不修改这个时钟源，那么系统将一直工作在一个时钟源不稳定不精准的环境下。

然而一般做法是改变时钟源，将时钟源改为 HSE。改变时钟源的通道是在相关寄存器设置的，在图中的 PLLSRC 可以实现对这两个频率的切换。

接下来我们要了解一下“锁相环倍频器 PLL/ 预分频器 Prescaler”。

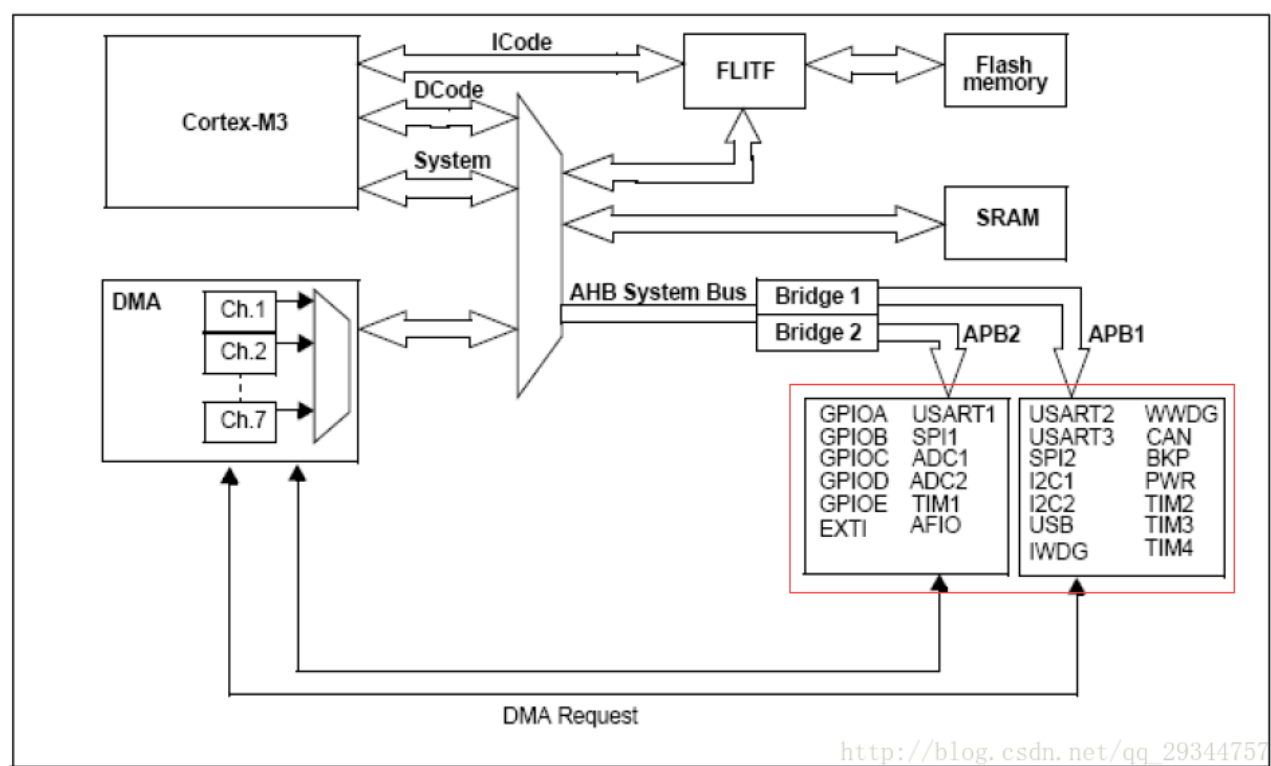
STM32 的 cpu 的工作常规频率是 72MHZ（超过 72MHz 工作称为超频工作，CPU 耗电加剧，且会发烫），但是我们接入的晶振是 8MHz，这就需要一个对频率加倍的操作，即倍频。如图中的 PLLMUL，PLLMULL 实现对接入时钟源的倍频，如 x2、x3、x4...倍频后的时钟源为 PLLCLK。



(P 2-4-83)PLLMULL

预分频器是实现频率削减作用的。将 HSE 倍频之后提供给 cpu，但是除了 cpu 之外，其他片内外设，如 SPI 控制模块、IIC 控制模块等的工作同样需要时钟源，这些外设的时钟源肯定是低于 cpu 运行时钟的，例如 USB 通讯才需要 48MHz，所以需要对比倍频后的时钟源进行分频。一般芯片的分频做法都是对一个时钟源倍频后供给某些部件，其他低于此倍频后的时钟都是基于此时钟源来分频的。用户可通过多个预分频器配置 AHB，高速 APB (APB2) 和低速 APB (APB1) 域的频率。AHB 和 APB2 域的最大频率是 72MHz。APB1 域的最大允许频率是 36MHz。SDIO 接口的时钟频率固定在 HCLK / 2。经过时钟源的选择、分频 / 倍频，就可以到 HCLK (高性能总线 AHB 用)、FCLK (供给 cpu 内核的用，常说的 cpu 主频)、PCLK (高性能外设总线 APB)、USBCLK、TIMXCLK、TIM1CLK、RTCCLK 等，外设是挂载 STM32 的总线上的，具体哪个外设挂载哪个总线，看下图：

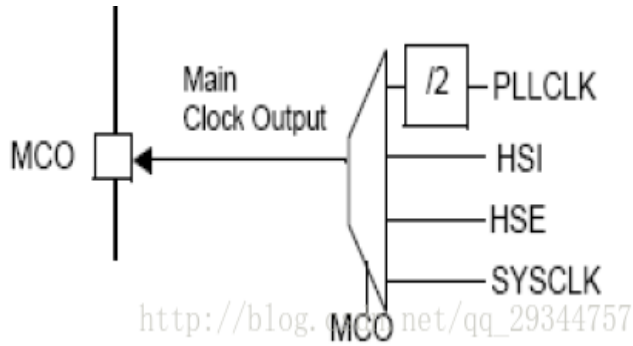
在软件开发中，我们要做的也无非设置门电路以选择时钟源输入、倍频 / 分频系数和打开 / 关闭对应外设所在总线的时钟。



[http://blog.csdn.net/qq\\_29344757](http://blog.csdn.net/qq_29344757)

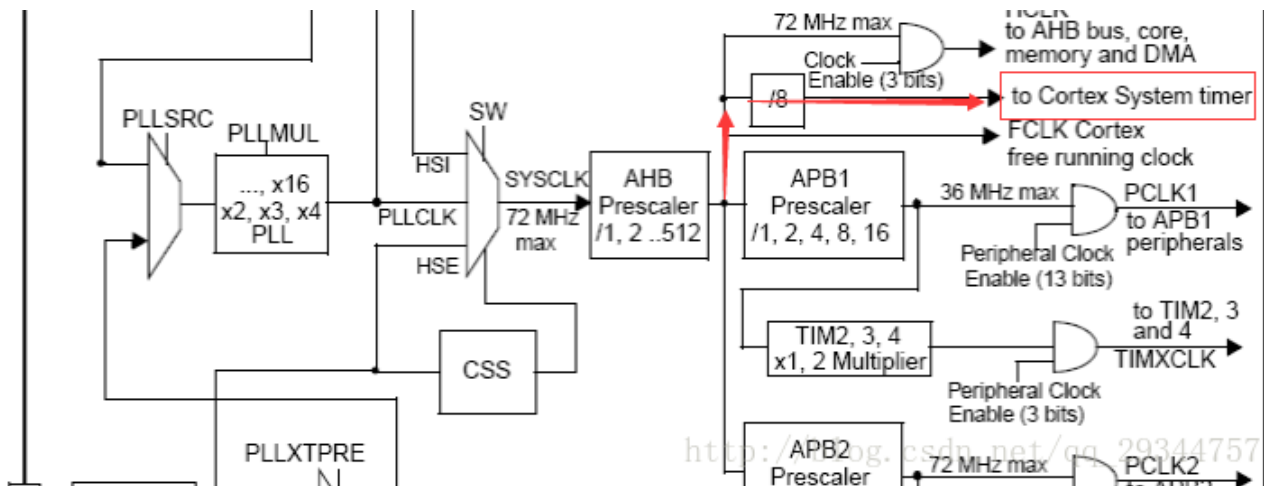
(P 2-4-84) 外载与总线的对应

另外，STM32 中也自带外部输出时钟源，即下图的“MCO”功能模块，可以将 PLLCLK / 2、HSI、HSE、SYSCLK 输出，供给其他系统作为输入时钟源，对这一功能模块也是又相应的寄存器，图中以 MCO 标注。



(P 2-4-85) “MCO” 功能模块

还有很重要的一点，即系统滴答时钟“Systick”。Systick 就是一个定时器而已，只是它放在了 NVIC 中断服务函数中，主要的目的是为了给操作系统提供一个硬件上的中断，称之为滴答中断操作系统进行运转的时候，也会有时间节拍。它会根据节拍来工作，把整个时间段分成很多小小的时间片，而每个任务每次只能运行一个时间片的时间长度，超时就退出给别的任务运行，这样可以确保任何一个任务都不会霸占操作系统提供的各种定时功能，都与这个滴答定时器有关。因此，需要一个定时器来产生周期性的中断，而且最好还让用户程序不能随意访问它的寄存器，以维持操作系统的节拍。只要不把它在 SysTick 控制及状态寄存器中的使能位清除，就一直执行。



(P 2-4-86) 时钟

RCC(复位与时钟控制器)通过 AHB 时钟 (HCLK)8 分频后作为 Cortex 系统定时器 (SysTick) 的外部时钟。通过对 SysTick 控制与状态寄存器的设置，可选择上述时钟或 Cortex (HCLK) 时钟作为 SysTick 时钟 (后者图中没画出)。另外，还有其他时钟，如 USB 时钟，ADC 时钟、独立看门狗时钟等，它们各自的时钟源通过前面学习，也可以轻易分析出来，这里不再赘述。(转自 CSDN “bright261” 原创文章《STM32 时钟体系结构》。https://blog.csdn.net/qq\_29344757/article/details/73479924)

通过以上系统的讲解，我们对时钟有了一个总体的印象。至于细节方面，出于篇幅我们便不再深入讲解。简而言之，时钟不仅可以带

给 stm32 稳定的工作状态，而且还将影响多种与时序关联紧密的外设，可以说是 32 的核心。时钟还能带来的另一项功能即是定时器，即设置我们所需的时间，在时间达到时进行另外的操作等。

有关定时器的讲解，我们以 STM32F103 系列为例。STM32F103 系列的单片机一共有 11 个定时器，其中：

- 2 个高级定时器；
- 4 个普通定时器；
- 2 个基本定时器；
- 2 个看门狗定时器；

1 个系统滴答定时器即“Systick”定时器。除去看门狗定时器以及滴答定时器，其余八个