Clock source and Timer module

Select the clock source and the frequency, enable and setup timers

#arm #stm32 #clock #timer

Last update: April 20, 2021

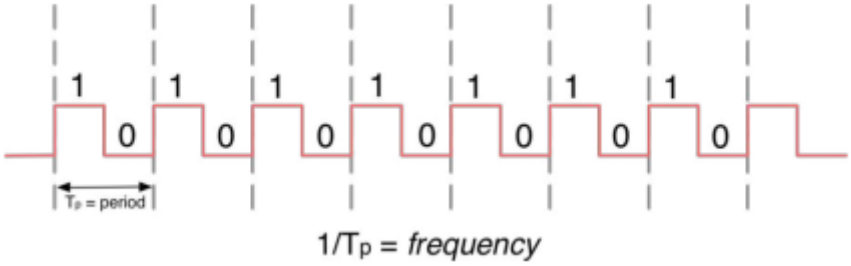
Table of Content

1. Clock source

Almost every digital circuit needs a way to synchronize its internal circuitry or to synchronize itself with other circuits.

A *clock* is a device that usually generates a periodical square wave signal, with a 50% duty cycle. A clock signal oscillates between V_L and V_H voltage levels, which for STM32 microcontrollers are a fraction of the V_{DD} supply voltage.

The most fundamental parameter of a clock is the frequency, which indicates how many times it switches from V_L to V_H in a second. The frequency is expressed in *Hertz*.



A typical clock signal

The majority of STM32 MCUs can be clocked by two distinct clock sources alternatively:

- an *internal RC oscillator* (**High Speed Internal** (HSI)) or
- an *external dedicated crystal oscillator* (**High Speed External** (HSE)).

There are several reasons to prefer an external crystal to the internal RC oscillator:

- An external crystal offers a higher precision compared to the internal RC network, which is rated of a 1% accuracy, especially when PCB operative temperatures are far from the ambient temperature of 25°C
- Some peripherals, especially high speed ones, can be clocked only by a dedicated external crystal running at a given frequency

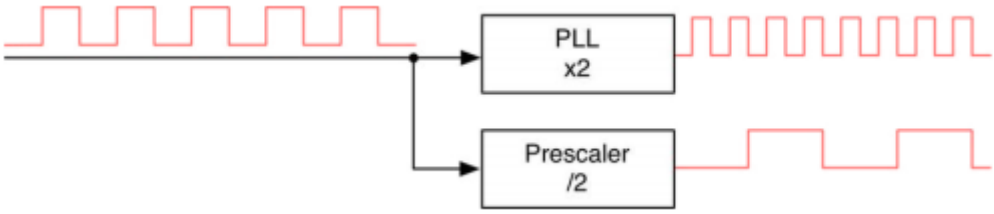
Together with the high-speed oscillator, another clock source can be used to bias the low-speed oscillator, which in turn can be clocked by:

- an external crystal (**Low Speed External** (LSE)) or
- the internal dedicated RC oscillator (**Low Speed Internal** (LSI)).

The low-speed oscillator is used to drive the *Real Time Clock* (RTC) and the *Independent Watchdog* (IWDT) peripheral.

Using several **Programmable Phase-Locked Loops** (PLL) and pre-scalers, it is possible to increase/decrease the source frequency at needs, depending on the requested performances, the

maximum speed for a given peripheral or bus and the overall global power consumption.

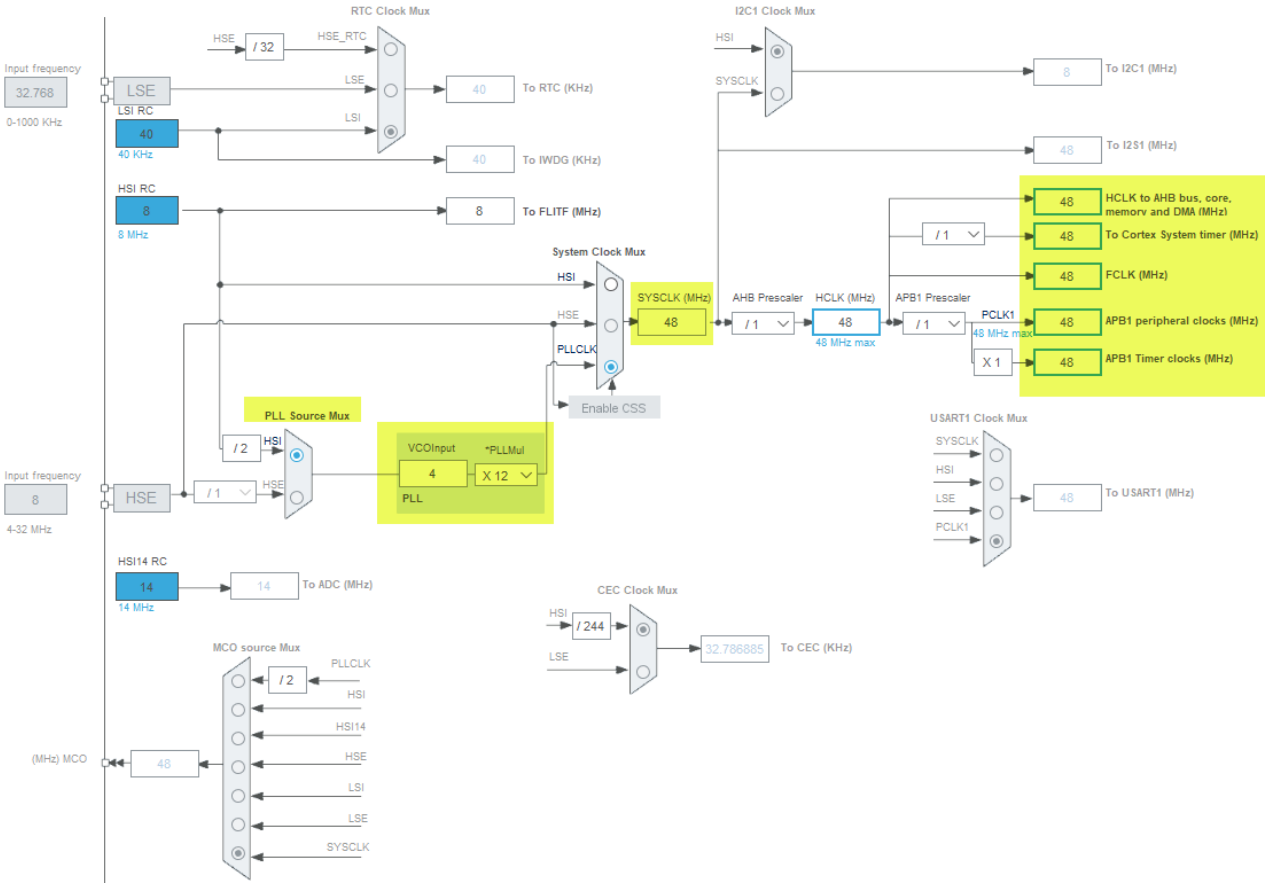


PLL is used to increased/ decrease clock frequency

2. Clock tree

The clock tree configuration is performed through a dedicated peripheral named **Reset and Clock Control** (RCC), and it is a process essentially composed by three steps:

1. The high-speed oscillator source is selected (HSI or HSE) and properly configured, if the HSE is used.
2. If need to feed the `SYSCLK` with a frequency higher than the one provided by the high-speed oscillator, then configure the main PLL (which provides the `PLLCLK` signal).
3. The System Clock Switch (SW) is configured to choose the system clock source from HSI, HSE, or `PLLCLK`. Then select the AHB, APB1 and APB2 (if available) pre-scaler settings to reach the wanted frequency of the High-speed clock (`HCLK` - that is the one that feeds the core, DMAs and AHB bus),and the frequencies of Advanced Peripheral Bus 1 (APB1) and APB2 (if available) buses.



Clock tree of STM32F0

SysTick

The System Tick Time `SysTick` generates interrupt requests on a regular basis. This allows an OS to carry out context switching to support multiple tasking. For applications that do not require an OS, the SysTick can be used for time keeping, time measurement, or as an interrupt source for tasks that need to be executed regularly.

3. Timers

Comments