Clock Speed:

The clock speed measures the number of cycles CPU executes per second, measured in MHz (megahertz) or GHz (gigahertz). A “cycle” is technically a pulse synchronized by an internal oscillator.

STM32F4 Discovery Board Reset And Clock Control (RCC):

Three different clock sources can be used to drive the system clock (SYSCLK):

- HSI oscillator clock – High Speed Internal
- HSE oscillator clock – High Speed External
- Main PLL (PLL) clock

The devices have the two following secondary clock sources:

- 32 kHz low-speed internal RC (LSI RC)
- 32.768 kHz low-speed external crystal (LSE crystal)

The first three clocks are used to drive the system clock for the microcontroller. The final two are low speed clocks and are primarily used to drive the watchdogs. Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

HSI:
After reset, the STM32 enables the HSI oscillator. This has a relatively low accuracy, only 1%, but is suitable for most applications. Using the HSI oscillator eliminates the need for an external clock in the final circuit design. On the STM32F4, the HSI oscillator has a clock speed of 16 MHz.

HSE:
The STM32 can operate using an external clock circuit. It is possible to design an external clock to run with a greater accuracy than the internal HSI clock enabling finer control of the operating parameters of the final circuit. The exact specification of the external clock frequency varies but is typically 4-16 MHz. The STM32F4 Discovery Board has a built in external oscillator circuit fitted with a 8 MHz crystal.

PLL:
The PLL is used to multiply it’s input clock source by a factor varying between 2 to 16. The input of the PLL is one of HSI, HSE or HSE/2. It is important to note that the configuration of the PLL cannot be changed once it has been enabled.

LSI:
The LSI is a low power clock used for the watchdog timers.

LSE:
The LSE is powered by an external 32.768 KHz clock. This provides a method of providing a low speed accurate clock for the real time clock.

Peripheral clocks
Lowering current draw has been a goal for most microcontroller manufacturers. One of the techniques used to achieve this is to switch off on-chip peripherals by removing access to their master clocks. On the STM32 devices, these clocks are known as the hardware and peripheral clocks and are controlled by the RCC (Reset and Clock Control) group of registers. Since there are more than 32 on chip peripherals, there are actually two registers used to switch on a clock: RCC_AHB1ENR and RCC_AHB2ENR for the Hardware clock, APB for the Peripheral clock. The clock is controlled by set/reset registers, so to turn a system on you set a bit in the ENR register, and to turn that same peripheral off you set the bit in the corresponding RCC_AHBxRSTR register.

Clock Distribution
Once the clock source has been selected it is necessary to configure the internal system and peripheral clocks. The internal clocks are:

- System Clock
- Advanced High Performance Bus (AHB)
- Low speed Advanced Peripheral Bus (APB1)
- High speed Advanced Peripheral Bus (APB2)

Each of these clocks can be scaled using prescalers.

System Clock
The system clock is used to determine the speed at which instructions are executed and has a maximum speed of 168MHz.

Advanced High Performance Bus (AHB)
Derived from the system clock, this bus has a maximum speed of 168MHz.

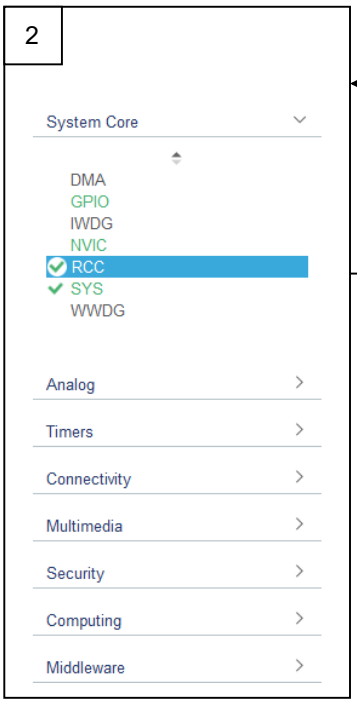
Low speed Advanced Peripheral Bus (APB1)
Derived from AHB, this bus has a maximum speed of 42MHz.

High speed Advanced Peripheral Bus (APB2)
Derived from AHB, this clock has a maximum frequency of 84MHz.

Non-System Clock Peripherals
A number of the peripheral clocks are not derived from the system clock but have their own independent source.

STM32F4 Discovery Clock Speed Config:

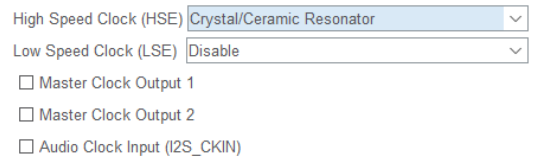
2



1

First start the project after that you need to go the Pinout & Configuration tab. After going to this tab, select the to the system core section and after that choose RCC sector.

3



If you are going to use HSI in your project, you do not need to make any changes in this part. If working with HSE oscillator, Crystal/Ceramic Resonator should be selected from HSE field.
In this case instruction explain how to work with HSE.

Configuring clock speed using HSE&HSI and PLL :

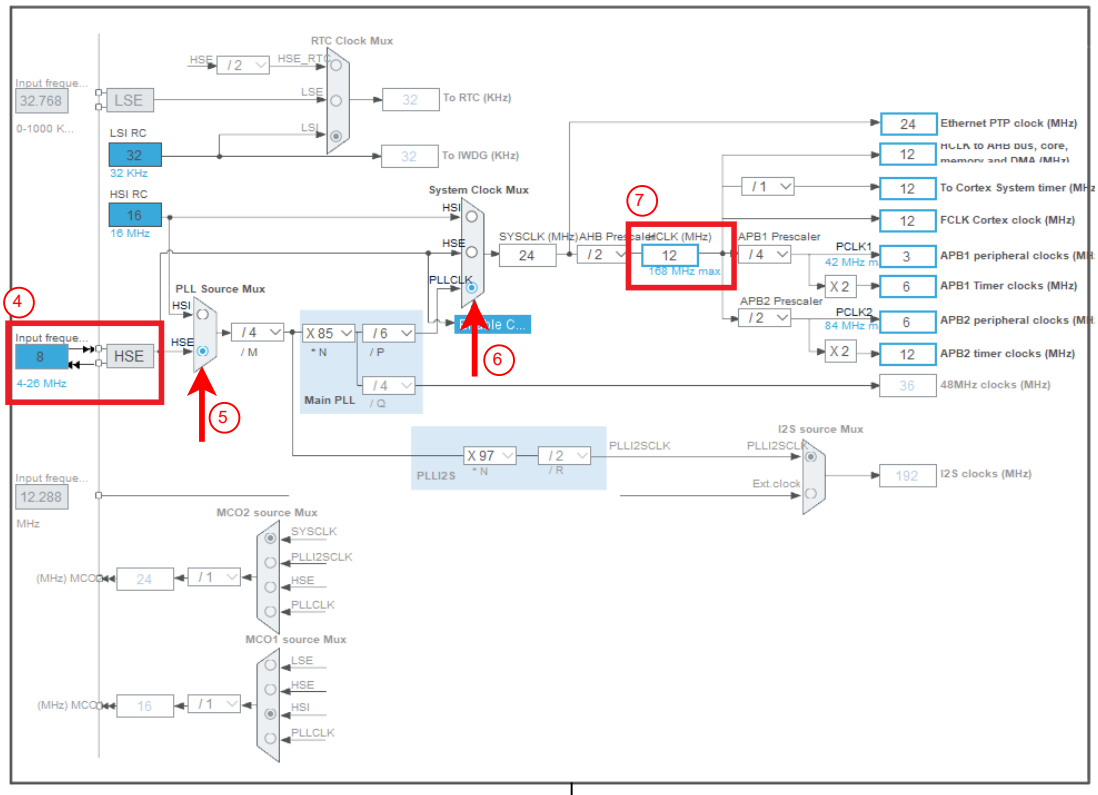
4.1 - If you are going to use HSE. Set the frequency of the HSE osliator (described on page 1) (8MHz).

5.1 - **If you are going to use <u>HSE</u>**: PLL circuit will be used so need to specify the oscillator source for the PLL circuit, here we choose the HSE oscillator.

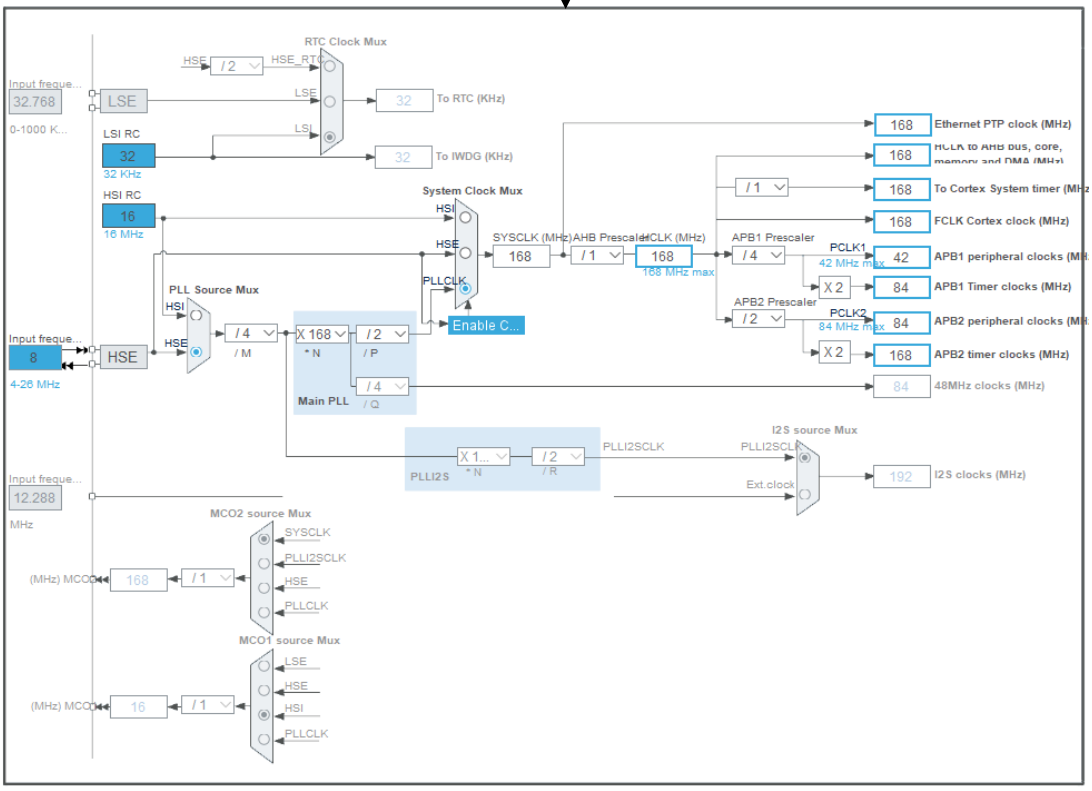
5.2 - **If you are going to use <u>HSI</u>:** PLL circuit will be used so need to specify the oscillator source for the PLL circuit, here we choose the HSI oscillator.

6 - Choose PLLCLK because circuit produce the clock speed by processing it in the PPL circuit.

7 - At this stage, enter the clock speed value which you want in the marked constructor and press "enter". The CubeIDE automatically calculates the clock speed for us using PLL circuit prescallers and other prescallers. (In this instruction value of 168 entered.)

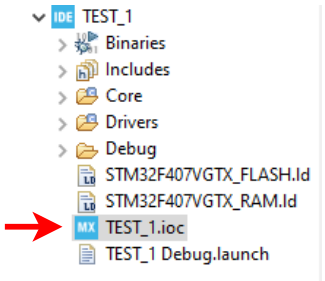


CubeIDE Calculates Clock Speed



Result :

END - The CubeIDE did the calculation automatically and entered the values. These values and prescallers will affect the circuit elements in the future like timers etc. You may need to go back and look at this table again. As shown in the file below, you can come to the project files section and open the .ioc file and look at this tab.



https://www.intel.com/content/www/us/en/gaming/resources/cpu-clock-speed.html

https://blog.mark-stevens.co.uk/

https://bluetechs.wordpress.com/