how works SysTick on MCUs? Is it regulary at 1ms ?

SysTick, short for System Tick Timer, is a basic timer found in many microcontroller units (MCUs), particularly those based on ARM Cortex-M cores. It provides a simple way to generate periodic interrupts and is often used for tasks like creating time delays, measuring time intervals, or implementing real-time operating systems (RTOS).

SysTick is not necessarily configured to generate interrupts every 1ms by default. The actual rate at which SysTick interrupts occur is determined by two factors: the MCU's core clock frequency and the value programmed into the SysTick Reload Value Register.

To set up SysTick to generate interrupts at a specific rate, you can follow these steps:

1. Determine the desired interrupt frequency (e.g., 1ms or 1000Hz).
2. Calculate the required SysTick Reload Value: Reload_Value = (Core_Clock_Frequency / Desired_Interrupt_Frequency) - 1.
3. Program the calculated Reload Value into the SysTick Reload Value Register.
4. Enable SysTick and its interrupt in the SysTick Control and Status Register.

For example, if your MCU has a core clock frequency of 48MHz and you want to generate SysTick interrupts every 1ms (1000Hz), you would calculate the Reload Value as follows:

Reload_Value = (48,000,000 / 1,000) - 1 = 47,999

Once you've set up SysTick, it will generate interrupts at the desired rate, and you can use an interrupt handler function to execute specific tasks when the interrupt occurs.

Keep in mind that the maximum period for SysTick interrupts is limited by the size of the Reload Value Register, which is 24 bits wide. This means the maximum Reload Value is 0xFFFFFF (16,777,215), which limits the minimum interrupt frequency for a given core clock.