Timers in STM32F407

Timers

Timer peripherals are used to maintain accurate timing information, generate accurate delays, measure delay. Timers in modern microcontroller are further enhanced to link these delays and measurement to events, then map these events to GPIOs or other peripherals. For examples,

1. Timers can be used to keep time in microcontroller that can be mapped with real time clock. Maintaining hours, minutes and seconds is one example
2. Timer can be used to generate accurate delays before or after performing specific operation. Toggling LED after each second is one example
3. Timers can be used to measure delays that we faced during execution of a specific process. Measuring how long a button was pressed is an example
4. Modern timers can be linked with GPIO to measure delay between two positive edges. This is called input capture
5. Timer can be linked to GPIO based on comparison with input count. If count value is higher than compare value, output is high. If count is lower, output is low. This functionality is called compare
6. Modern timers can generate Pulse Width Modulated (PWM) signals with dead time and complementary outputs
7. Timer events can be linked with ADCs or DMAs to schedule input reading from sensors or can be linked with DAC for output generation rate control

Cortex-M4 System Timer Systick

In every cortex-M4 core system timer SysTick is present to support basic timing functionality required. SysTick is a 24 bit down counter. This means that on every clock edge/tick, it’s value will be decremented by 1. When the value is deceremented from 0, an event is generated. This event is called underflow event. The value of SysTick is reloaded with higher vale and down count continues.