Timers and working modes

Understand about timers and working modes to configure the periodic tasks, Pulse-width modulation, and Capture mode.

#arm #stm32 #timer #pwm #capture

Last update: May 3, 2021

Table of Content

1. Timer overview

1. Timer overview

A timer is a free-running counter with a counting frequency that is a fraction of its source clock. The counting speed can be reduced using a dedicated pre-scaler for each timer. Depending on the timer type, it can be clocked by the internal clock (which is derived from the bus where it is connected), by an external clock source or by another timer used as "master".

The F_{sys} is not the frequency that is incrementing the timer module, but it gets divided by the Pre-scaler, then it gets fed to the timer. Every clock cycle, the value of the timer is incremented by 1.

Let's see an example to calculate the timer period.

- F_{sys} = *80 MHz*
- Pre-scaler = *1:1024*
- Timer gets incremented by 1 every *1024* 1/80000000 s = 12.8 us*
- If set overflow at full 16-bit (at 65535), and start counting from 0, it will generate a signal every *12.8 us * 65535 = 838848 us = 838.848 ms*

A timer can have additional pre-load register, therefore, timer will count from 0 to the pre-load value, or vice-versa.

When driven by an external input, such as external pulse or button, timer can be used to count the number of events happened, and measure the frequency.

update soon