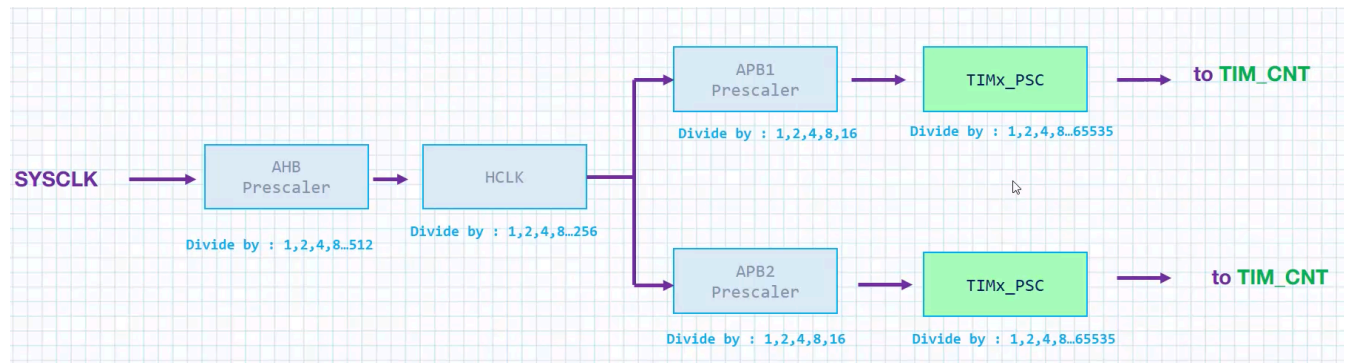


STM32 Timers

Operation modes:

1. It can be used as a time-base generator.
2. Input capture mode: Capture the frequency of an external event.
3. Output compare mode: Control an output waveform or indicate when a period of time has elapsed.
4. One pulse mode:

Timer pre-scaling



Common terminology used when working with timers:

1. Update event: When a timeout occurs or how long it takes for a flag to be raised.
2. Period: Value loaded into the auto-reload register (TIM_ARR).
3. Up counter: Counts from zero to set value.
4. Down counter

Computing update event

$$\text{Update event} = \text{Timer clock} / (\text{Prescaler} + 1) \times (\text{Period} + 1)$$

The prescaler is used to divide the input clock frequency of the timer.

Period refers to the value that the timer counts up to (or down to) before it resets and generates an update event or interrupt.

Developing a timer output compare driver

Objective

The objective is to configure the timer to directly control the GPIO pin, toggling an LED connected to PA5 each time the timer reaches its timeout period.

Hardware and Peripherals

- **Timer Module:** TIM2
- **Clock Bus for Timer Module:** APB1
- **GPIO Pin:** PA5 (connected to the user LED)
- **Alternate Function for PA5:** AF01 (TIM2 channel 1)

GPIO Pin Initialization

1. **Enable Clock Access to GPIOA:**
 - Enable the clock for the GPIOA peripheral.
2. **Configure PA5 as an Alternate Function:**
 - Set PA5 to alternate function mode.
 - Select the appropriate alternate function (AF01) to connect PA5 to TIM2 channel

Timer Initialization

1. **Enable Clock on APB1 Bus:**
 - Enable the clock for the TIM2 timer module.
2. **Set Prescaler and Period:**
 - Configure the prescaler to divide the timer clock frequency.
 - Set the auto-reload register (period) to determine the timer's timeout period.
3. **Configure Output Compare Mode:**
 - Set TIM2 to operate in output compare mode.
 - Configure the output compare mode to toggle the output each time the timer reaches the compare value.
4. **Clear Timer Counter:**
 - Reset the timer counter to start counting from zero.
5. **Enable Timer:**
 - Enable the timer to start counting.

Operational Details

- **Prescaler:** Reduces the input clock frequency to a manageable value for the timer. This is done by dividing the input clock by (prescaler + 1).

- **Auto-Reload Register (ARR):** Defines the value at which the timer counter resets to zero and generates an update event. The timer counts from 0 to ARR, which determines the timer period.
- **Output Compare Toggle Mode:** Configured to toggle the state of PA5 each time the timer reaches the compare value. This means the LED connected to PA5 will change state (on/off) every time the timer completes a count cycle up to the ARR value.

Configuration in code

- **System Clock Frequency:** 16 MHz
- **Desired Timer Frequency:** 1 Hz (for LED blinking every 1 second)
- **Prescaler Calculation:**
 - To achieve a timer clock of 10 kHz, the prescaler should be set to 1599.
 - $\text{Timer Clock} = \text{System Clock} / (\text{Prescaler} + 1)$
- **Auto-Reload Value:**
 - To achieve a 1-second period with a 10 kHz timer clock, set ARR to 9999.
 - $\text{Timer Period} = (\text{ARR} + 1) / \text{Timer Clock}$