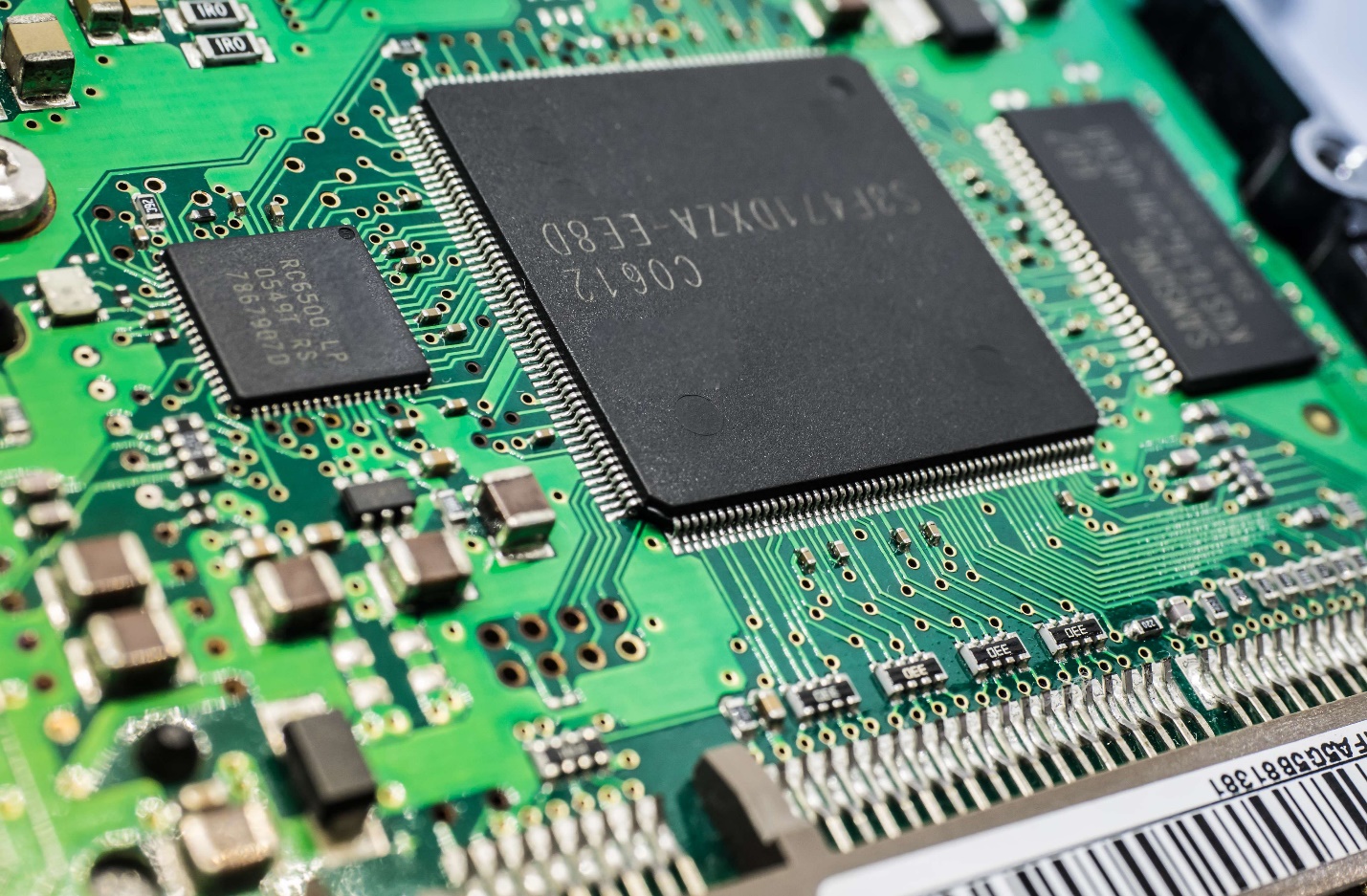
POC for Timers, WatchDog, FreeRTOS

July Trendafilov(4674693)



Contents

[Timers 3](#_Toc119368991)

[Watchdog 4](#_Toc119368992)

[FreeRTOS 6](#_Toc119368993)

# Timers

To setup a timer is STM32, some registers are required to be set. In the following example function SetTim2 is used to setup TIM2 with interrupt.

Text

Description automatically generated

The following code firstly sets up the prescaler divider to match the specified time frame. For example here the Timer counts to 1s. To do so the PSC register must be set to 7200 is order for the Clock rate to be prescaled:

72MHZ : 7200 = 10kHz

By that 1 : 10000 = 0,0001 and one 0,1 ms will pass as the counter ticks. When the counter reaches 10000 that’s when a second has passed and an interrupt is generated. When the Timer reaches the ARR value the interrupt is generated and the flag is set to 1.

Text

Description automatically generated

In the IRQ handler, firstly the interrupt bit must be toggled and then it is printed to the screen that TIM2 interrupt appeared at a frequency of 1 second.

Graphical user interface, text, application

Description automatically generated

# Watchdog

The watchdog is used to reset a system when a task takes too long time. The following function sets up the watchdog by writing a prescaler and a reload value. If the reload value is passes, a reset on the system is done.

Text

Description automatically generated

The watchdog timer has a 40kHz clock rate. To set a specific threshold a prescaler can be used like this:

The IWDG->PR register has the value 5 which means that the clock rate will be divided by 128:

Graphical user interface, text, application, chat or text message

Description automatically generated

40000 : 128 = 312.5 -> 1 : 312.5 = 0.0032 so each tick 3,2 ms pass.

If the value of the RLR register is reached which 1535 ticks, a reset is triggered.

This is printed to the serial monitor and it is outside the infinite loop and it is executed only once on setup.

A screenshot of a computer

Description automatically generated with medium confidence

The time when the watchdog will trigger is 3.2ms \* 1535 ticks = 4.912s if not reloaded. If the watchdog is reloaded in this time frame everything will be fine.

Graphical user interface, text

Description automatically generated

In the main writing 0xAAAA to KR reloads the timer.

Text

Description automatically generated

But if we comment the reload timer register, the watchdog will surpass the threshold. The watchdog will trigger a reset.

Graphical user interface, application

Description automatically generated

This is the serial communication. Every second the TIM2 generates an interrupt and on the 5th second IWDG is triggered.

Graphical user interface, text, application

Description automatically generated

# FreeRTOS

To use freeRTOS in a project one must start the scheduler and initialize the Kernel.

Graphical user interface, text

Description automatically generated

In this example I will use thread flags and mutex:

Text

Description automatically generated

Then is function MX\_FREERTOS\_Init I initialize the mutex and start the threads:

Text

Description automatically generated

Here the mutex has its default attributes so we must manually lock and unlock it and three function are created – one dummy and two for the workshop.

In function AddToCounter, sharedVariable is always incremented until it passes 1000 and then signals to PrintToSerial to take control. AddToCounter is guarded by a mutex and if the value of sharedVariable passes 1000, a flag is set and the mutex is released and then function PrintToScreen acquires the mutex, changes the value of sharedVariable and then releases it and returns the context to AddToCounter and this process is repeated forever.

Text

Description automatically generated

osDelay is used to switch the context.

A screenshot of a computer

Description automatically generated with medium confidence

And the output of the program is:

Graphical user interface, application

Description automatically generated

As observed when the value passes 1000, 900 is subtracted and the counting start again.