

# Embedded and Real-time Systems

## Real-time tasks

### Exercise 1 – Timeline scheduling

Use the timeline scheduling to schedule four tasks. Each task should periodically switch on and switch off its LED.

- Task 1 (green LED) – 100 ms on / 100 ms off
- Task 2 (orange LED) – 150 ms on / 150 ms off
- Task 3 (red LED) – 250 ms on / 250 ms off
- Task 4 (blue LED) – 400 ms on / 400 ms off

Build this exercise on top of Exercise 11 from Assignment 1.

#### Hints:

- Determine the lcm and gcm of the tasks. Set the systick to the gcm.

### Exercise 2 – Scheduling by HW timers

The board comes with 14 HW timers. Each can be configured to periodically invoke its interrupt handler. Interrupt handlers are preemptible and can be associated with priorities. This allows for relatively simple preemptive real-time scheduling.

Implement a timer driver and setup 4 timers – each for one task. Re-implement Exercise 1 with the timers. Use Rate-Monotonic rule to determine the priorities and assign the respective priorities to the timer interrupts.

#### Hints:

- You can use any of the 14 timers. However, the general purpose ones TIM2-TIM5 should be fine for the task.
- First make the timer to count
  - Enable timer clock: `__HAL_RCC_TIM{XXX}_CLK_ENABLE [1]`
  - Initialize timer: `HAL_TIM_Base_Init [1]`
  - Start Timer: `HAL_TIM_Base_Start [1]`
  - Read current timer value: `__HAL_TIM_GetCounter [1]`
- Then configure interrupt on timer update:
  - Enable timer update interrupt: `__HAL_TIM_ENABLE_IT [1]`
  - Enable interrupt on interrupt controller: `HAL_NVIC_EnableIRQ [1]`
  - If necessary adjust interrupt priority: `HAL_NVIC_SetPriority [1]`
  - Define matching IRQ handler (see assembly initialization code in your project for list of interrupt handlers)
  - Once interrupt occurs clear timer update flag in order to service the interrupt: `__HAL_TIM_CLEAR_FLAG [1]`

- TIM2-5 are connected to APB1 bus, which has on F407 device 42MHz clock (base clock of 168MHz divided by 4). But, timer has internal PLL, which double this frequency for timer, up to 84MHz. Not each timer is connected to APB1, there are also timers connected on APB2, which works at 84MHz by default, and internal PLL increases this to up to 168MHz.

Timer count frequency for TIM2-5 is computed as follows:

`timer_tick_frequency = Timer_default_frequency / (prescaler_set + 1)`

### Exercise 3 – FreeRTOS

FreeRTOS [2] is an open-source real-time operating system with very low footprint. It is essentially just a scheduler and a few synchronization primitives. A sample project with FreeRTOS can be found at GitHub <https://github.com/d3scomp/stm32f4-rtos>.

Take the sample project and use it to re-implement Exercise 2 with the help of FreeRTOS tasks.

### Exercise 4 – Synchronization

Modify your tasks from Exercise 3 as follows:

Each task should print:

```
"<color> [\r\n"
"<color> led <state>\r\n"
"] <color>\r\n"
```

<color> corresponds to "green", "orange", "red", "blue"

<state> corresponds to "on", "off"

Lower the UART speed to somewhere around 4800 or 9600 bps. You should get to the situation when strings are corrupted because printing of one strings overlaps with another.

Protect the printing of the string by a mutex. A task should enter/exit a mutex 3 times (once for each line printed).

Check that the tasks correctly preempt one another (i.e. green one gets never preempted, orange one can be preempted only by the green one, etc.).

#### Hints:

- Most likely you will need a bigger stack per task. The stack size is counted in 32-bit words. A stack of 256-512 words (i.e. 1-2KB) should be possibly enough for you. A symptom of having too small stack is that your code starts behaving absolutely unexpectedly. See **FreeRTOS/include/task.h**.
- The right function to create the mutex is `xSemaphoreCreateMutex`. You lock it by `xSemaphoreTake` and unlock it by `xSemaphoreGive`. See **FreeRTOS/include/semphr.h**.

## References

1. Description of STM32F4xx HAL drivers:  
[http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user\\_manual/DM00105879.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00105879.pdf)
2. <https://www.freertos.org/>