Exercise 7 Virtual timer

In this exercise we will configure a number of virtual timers to trigger a callback function at various frequencies.

In the Pack Installer select "Ex 7 Virtual Timers" and copy it to your tutorial directory.

This is our original led flasher program and code has been added to create four virtual timers to trigger a callback function. Depending on which timer has expired, this function will toggle an additional LED.

The timers are defined at the start of the code

```
osTimerId_t timer0,timer1,timer2,timer3;

static const  osTimerAttr_t timerAttr_timer0 = {

        .name = "timer_0",

};

static const  osTimerAttr_t timerAttr_timer1 = {

        .name = "timer_1",

};

static const  osTimerAttr_t timerAttr_timer2 = {

        .name = "timer_2",

};

static const  osTimerAttr_t timerAttr_timer3 = {

        .name = "timer_3",

};
```

They are then initialized in the main function;

```
timer0 = osTimerNew(&callback, osTimerPeriodic,(void *)0, &timerAttr_timer0);

timer1 = osTimerNew(&callback, osTimerPeriodic,(void *)1, &timerAttr_timer1);

timer2 = osTimerNew(&callback2, osTimerPeriodic,(void *)2, &timerAttr_timer2);

timer3 = osTimerNew(&callback2, osTimerPeriodic,(void *)3, &timerAttr_timer3);
```

Each timer has a different handle and ID and passed a different parameter to the common callback function;

```
void callback(void const *param){

switch( (uint32_t) param){
```

```
            case 0:
                    GPIOB->ODR ^= 0x8;
            break;
            case 1:
                    GPIOB->ODR ^= 0x4;
            break;
            case 2:
                    GPIOB->ODR ^= 0x2;
            break;
    }}
```

When triggered, the callback function uses the passed parameter as an index to toggle the desired LED.

In addition to the configuring the virtual timers in the source code, the timer thread must be enabled in the RTX5 configuration file.

Open the RTX_Config.h file and press the configuration wizard tab

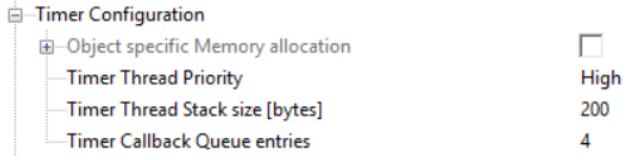

Fig 29 configuring the virtual timers

In the system configuration section make sure the User Timers box is ticked. If this thread is not created the timers will not work.

Build the project and start the debugger

Run the code and observe the activity of the GPIOB pins in the peripheral window

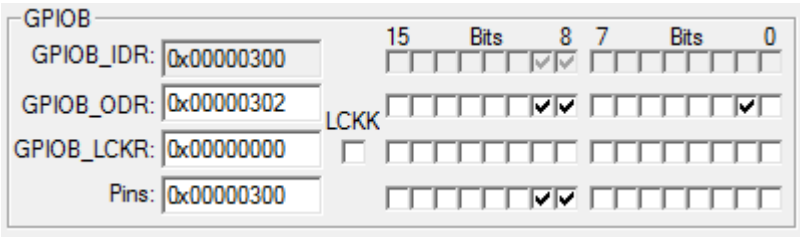

Fig 30 The user timers toggle additional LED pins

There will also be an additional thread running in the System and Thread Viewer window

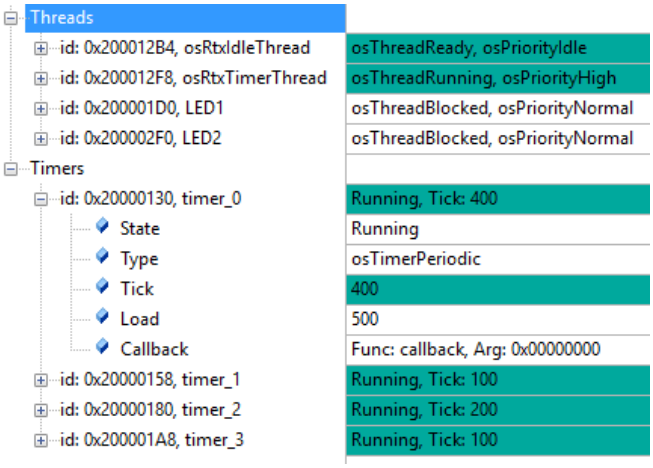

Fig 31 The user timers create an additional osTimerThread

The osDelay() function provides a relative delay from the point at which the delay is started. The virtual timers provide an absolute delay which allows you to schedule code to run at fixed intervals.