



Introduction to the use of FreeRTOS

Objectives:

Learn how to use the basic functionalities of the FreeRTOS real-time kernel.

1. Study the the FreeRTOS API, the tasks' structure, the initialization an termination functions and some of the IPC mechanisms.
2. Develop simple applications using the FreeRTOS real-time kernel.

Procedure:

1. Analysis of the supplied source-code:

Download the supplied software “freertos-class-sources.zip”, from the course website.

Unzip the file and open the project “./FreeRTOS-STR/FreeRTOS/Demo/DETPIC32_MPLAB” using Microchip's MPLABX IDE.

1. Study the application code (“main.c”). Pay special attention to the system calls used to define, create and activate the tasks. Study in detail the structure of the tasks.
2. Compile and execute the supplied example. Observe the application behavior and explain it. Detailed instructions about the process of compiling and executing applications on the DETPIC32 board are supplied on the course site.

2. Tasks to carry out:

Complete the tasks specified below. Annotate all the changes made to the code. You must deliver a short report (max 2 pages) containing your observations and summarizing the required code changes.

1. Invert the tasks' priorities. Observe the activation pattern of the LED. Explain why its behavior changes.
2. Writing to the terminal (via the UART, using the “printf()” function) is not reentrant, so the information may get corrupted. Create a program with two tasks that send information to the terminal. Create a situation in which the information corruption becomes evident. Modify the program to implement a “Gatekeeper” mechanism, in which the actual writing is made by a single task. The used IPC should be a queue.
3. **(Optional)** Implement an admission control mechanism. This mechanism may consist in a function which receives all the arguments required to crate a task (i.e., the same arguments as the xTaskCreate() function), as well as additional parameters



that you find necessary.

- Depending on the schedulability test, this function should create the task, via the `xTaskCreate()` function, or return an error..
- Assume that task priorities are assigned according with the “Rate-Monotonic” criteria

The FreeRTOS API is documented on <http://www.freertos.org/a00106.html>