

# LAB 3

## Introduction to FreeRTOS

### Objective:

- The main objective for this lab is to understand how freeRTOS work with ESPRESSIF. The lab consists in creating two tasks.
  - Task 1 - Toggle an LED every 500 ms
  - Task 2 – Print “Hello World” every second

### Bonus:

- For EE4178 is just a bonus and for EE5190 is mandatory
  - Create another task which toggles another LED every 2 seconds and it should display in the terminal its current state; for example, LED is ON.
- Bonus for EE5190
  - Create a fourth task that runs the previous lab lightshow sequences and make it run every 5 seconds.

### Pre-Lab:

- What type of programming is RTOS based of?
- What is the definition of the acronym RTOS?
- What is the function to create a task?
- When creating a task what parameter should be considered?

## C helpful functions

For this lab, the most important function call is `xTaskCreate` which is a function in `task.h` file. This function has a few parameters that must be pass on when calling the function. For instance we want to create a simple that call `example_task` which has a stack of 2048 and it will have no arguments or handle; it will be the following: `xTaskCreate(&example_task,"example task", 2048, NULL, NULL );\`

Data type	Variable name	Description
TaskFunction_t	<i>pvTaskCode</i>	task function name
const char *	<i>pcName</i>	name to associate the task
configSTACK_DEPTH_TYPE	<i>usStackDepth</i>	stack size
void *	<i>pvParameters</i>	arguments
TaskHandle_t *	<i>pxCreatedTask</i>	handle to store task

```
BaseType_t xTaskCreate(  
    TaskFunction_t pvTaskCode,  
    const char * const pcName,  
    configSTACK_DEPTH_TYPE usStackDepth,  
    void *pvParameters,  
    UBaseType_t uxPriority,  
    TaskHandle_t *pxCreatedTask  
);
```