

Escuela de Ingeniería Informática

Karl Deilmann  
Dani Quintero

Computación en la Nube

# Report

21 of Sep - 29 of Dec

# Contents

A	2
References	2

# Exercise 1b

## Thread load-balancing

The code we present to this assignment is based on the 3rd example code for RTOS/ChibiOS. When uploaded to the Arduino  $\mu$ -Controller, the startup procedure is the same as in the example code. To enable dynamic load balancing, we added a call of `top_load_balance()` in the `static THD_FUNCTION(top, arg)` function. We implemented our logic in `top_load_balance()`.

We read the thread-load of the last cycle and measure how close we are to the goal of 85% CPU utilisation. We use the relative distance between *current load* to *target load* and compute a dynamic adjustment value. This value is added (or subtracted) to the value stored in `threadLoad[tid]`, where `tid` is the thread to adjust (this structure is provided by the base code), and thus will enable to run said thread longer (or shorter) on the next cycle. With this technique we approach the optimal CPU utilization of 85% and reach it after a few cycles.

## Bonus Task: Handle random spikes in thread load

For the bonus task, we just added a simple log in the idle thread (`loop()`) to periodically disrupt the balance by changing the value in `threadLoad[tid]` and thus changing the load behaviour of one thread. Our load-balancing logic takes care on the next cycle to adjust the values (across all threads) and restore the balance to the target load.

# Appendix A

Git repository: [https://github.com/Puvogel/Internt\\_de\\_las\\_Cosas](https://github.com/Puvogel/Internt_de_las_Cosas)