Embedded - Intermediate Report

Vasileios Konstantaras, Alena Tesarova, Stylianos Tsagkarakis

25 April 2020

1 RTOS Selection

We have selected FreeRTOS as the RTOS we are going to install and evaluate for this assignment. Since there is a lack of hardware we decided to go with this because there is a simulation of FreeRTOS through a Windows port (for more info see here), thus it will be easier for us.

2 Metrics for Benchmark

The metrics we are going to evaluate separate in three sets:

1. Latency
    - System calls latency
    - Latency of peripheral functions
    - Context switch overhead
    - Jitter causes by OS overhead
2. Performance / Throughput
    - Memory usage
    - Program memory usage
    - Reliability and determinism
3. OS Features
    - Supported algorithms
    - Real-time tracing features
    - Support of static memory allocation
    - Security
    - OS Support / Community support

3 Implementation

While benchmarking FreeRTOS we should follow the requirements mentioned below:

- The measurements should be as accurate as possible.
- The solutions developed should be as non-intrusive as possible.
- The measurement should give average results.

Example and principles [1]

- The latency of functions can be examined by measuring the execution time. The on- chip timer based software tool gives timestamps at both the beginning and the end of a given function. It is suggested to use two methods of measuring time.
- The memory usage of a given data structure can be measured by calculating the amount of heap used in the *malloc()* function when the concerned structure is created.
- The program code size is shown by a map file created by the compiler.

[1] All of the above will be tested in custom scripts written by us to test the system both normal execution load and under extreme load.