ANALYSIS OF SELECTED RTOS CHARACTERISTICS

Lukáš Otava
Master Degree Programme (2), FEEC BUT
E-mail: xotava01@stud.feec.vutbr.cz

Supervised by: Pavel Kučera
E-mail: kucera@feec.vutbr.cz

Abstract: This paper is focused on a comparison of the basic characteristics of the selected real-time operating systems (RTOS). Three different RTOS (CoOS, FreeRTOS, uC/OSIII), running on the same Cortex-M3 HW configuration, were selected. The results of such experiments are rarely presented in the technical articles or specifications. The comparison is based on measurements of the basic time parameters including an overhead of the RTOS core and an implementation of the timer functionality. The best suited RTOS with the most reliable results will be utilized in the robot's control system.

Keywords: Cortex-M3, RTOS performance, RTOS measurement, context switching time, timer

1 INTRODUCTION

With an increasing power on the one hand, and decreasing price of nowadays microcontrollers on the other hand, more attention is focused onto the software part of the control systems (called firmware in embedded solution). There is a trend of integration many undemanding subsystems into the one high power microcontroller in the recent years. Therefore a problem of uniform utilization of the shared resources in the microcontroller has to be maintained. Program memory and data memory together with a computing power are typical examples of the most important shared resources of the microcontrollers. In complex control systems, there are many tasks that must be executed apparently concurrently with the time constraints. There are also requirements on time-restricted reactions on external the events (called interrupts). System that performs these requirements in time-deterministic way are known as real time operating systems (RTOS). RTOS help to keep even complex computational problems relatively clear.

This paper is focused on comparison of the basic characteristics of three different RTOS running on the same hardware. Such results are not usually published in the RTOS specifications or in the technical articles. Different results are expected according to the different implementation of the RTOS kernels. The best suited RTOS with the most reliable results will be implemented as a firmware of a robotic vehicle with a differential drive. Control system block diagram is shown in Figure 1. This control system was constructed at the Department of Control and Instrumentation, BUT. The photo of this system is shown in Figure 2.

2 RTOS CHARACTERISTIC MEASUREMENT

There are many available solutions of RTOS systems suited for ARM Cortex-M3 architecture[1] on the market. CooCox CoOS, Real Time Engineers FreeRTOS and Micrium uC/OSIII were selected for measurements as comparable systems (similar range of features). CoOS represents small RTOS available only for Cortex-M3 architecture. FreeRTOS represents open-source solution available for 33 architectures of microcontroller based systems[2]. uC/OSIII represents commercial product. Overhead of different RTOS kernels is comprised in the remaining tests. The overhead is determined mostly by a context switching time and a kernel operations time (tasks scheduling, objects synchronization). All tests are performed with $40MHz$ microcontroller clock frequency. Microcontroller code is compiled

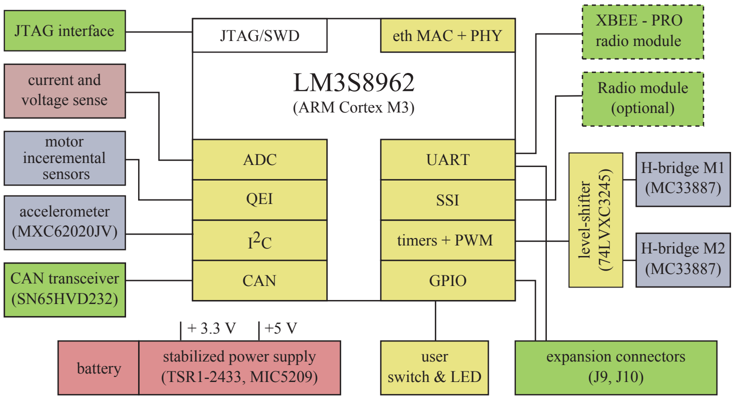

Figure 1: Block diagram of control system

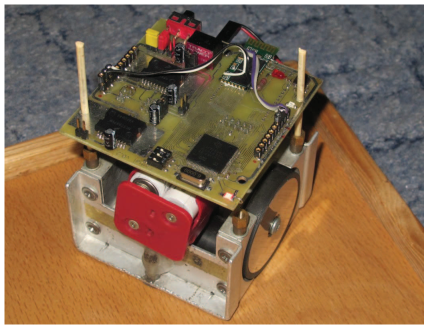

Figure 2: Differential drive robot

with GCC 4.6.2 in CoIDE 1.6 development tool with Thumb-2 instruction set. All test are performed with RTOS scaled to a minimal configuration.

2.1 Measurement of the context switching time overhead

A context switch is an operation of the kernel that switches the CPU core from one task to another. The context switch together with object synchronization are the most often executed operations that are reflected in RTOS kernel overhead.

For the measurement purposes, there are two tasks running in the, one is with high and one is with the low priority. The time, when the high priority task is waiting for the low priority task (using mutex or queue) is measured by a HW timer in the microcontroller. Only two synchronization objects are tested from many others that are provided by selected RTOS. Resulting time incorporates kernel overhead. Timer is clocked by an external reference generator Agilent 33120. Histogram of 1000 measurements is shown in Figure 3. Expanded uncertainty of the measurement is 0.0642 μs with 95% confidence.

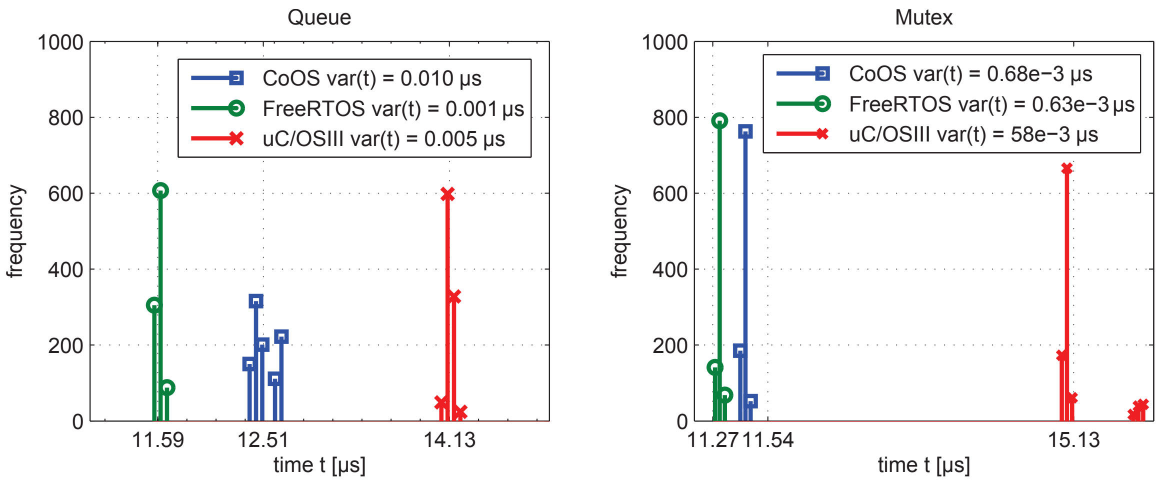

Figure 3: Frequency histograms of context switching times with synchronization waiting

2.2 SW timer measurement

Software timer implementation is comprised by a deviation of individual period of the SW timer. This deviation is commonly called jitter. This period is measured by a HW timer in the microcontroller clocked by a reference generator Agilent 33120. There is an additional task with a dummy computation load in this test. Frequency histogram for 7000 measurements is shown in Figure 4. Expanded uncertainty of the measurement is 0.192 μs with 95% confidence.

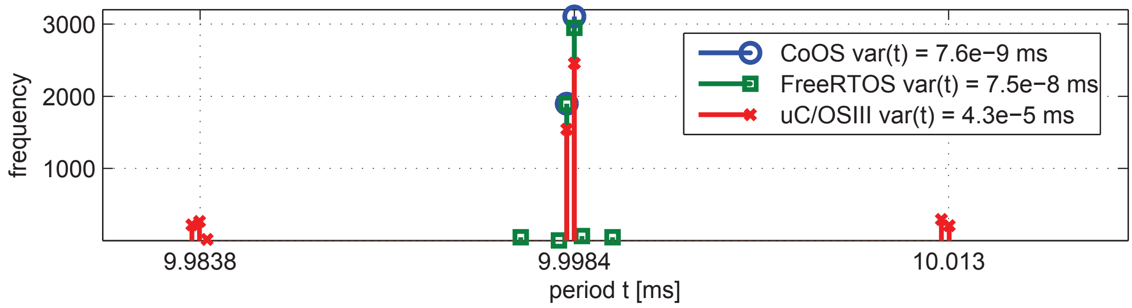

Figure 4: SW timer period histogram

Absolute timing comparison among selected RTOS was performed too. An aliasing effect based measurement was performed with a sampled triangle reference signal. The performed tests verifies that the timer functionality, from the absolute timing point of view, is implemented identically and the deviation of the timer period is caused mainly by the used crystal oscillator.

The amount of code memory needed by the tests is summarized in the Table 1.

Table 1: Code size comparison

code size [B]	CoOS	FreeRTOS	μC/OSIII
queue waiting	6860	7988	10388
mutex waiting	7236	6928	9864
SW timer abs. timing	15144	16776	12908
SW timer rel. timing	14936	16565	12708

3 CONCLUSION

In this paper a description and test results of the selected RTOS characteristics are presented. The context switching time, acquired by waiting for the synchronisation object of mutex and queue, was measured for three different RTOS running on the same hardware. For each experiment the measurement uncertainity was determined. There are some visible differences in times and their dispersion in 1000 measurements among RTOS. The shortest time and dispersion was measured for FreeRTOS. Context switch time of uC/OSIII is longer but with lower dispersion than CoOS.

Implementation of SW timer is compared by timer period dispersion from 7000 measurements. SW timer period jitter is characterized by these measurements. Utilization Agilent 33120 as a reference time base resulted in a reduction of the measurement uncertainty to the minimum. CoOS and FreeRTOS showed the best results with the lowest period dispersion.

To sum up, FreeRTOS can be pronounced as the best performed from the presented comparison. Therefore it was used as basis for a control system of a robotic vehicle. However general recommendation could not be declared as the best suited RTOS is strictly depended on the chosen HW platform and running tasks.

REFERENCE

[1] Yiu Joseph. *The Definitive Guide to the ARM Cortex-M3, Second Edition*. Burlington: Newnes, 2009.

[2] Barry R. *FreeRTOS API Reference* [online]. 2010, [cit. 24.4.2012]. Available at URL: <http://www.freertos.org/a00106.html>.