

Hartstone Hard Real Time Benchmark

Compatibility Issues

Daniel Casini, Emiliano Palermi, Matteo Pampana

Abstract This document presents the set of possible compatibility issues that has been analyzed and their relative solutions. Considering the project specification, in this test two boards are used under two different operating systems. In particular, the STM32F4 board and the Flex board are used, using the two most important operating systems in the real time application field, that are FreeRTOS and Erika.

1 General Consideration

The system that is going to be developed is a benchmark, this fact impact to the implementation of the additional libraries, that eventually, needs to be implemented: any kind of interference with the benchmark is acceptable from this point on. The basic functionalities that need to be tested are the USART interface, for each possible configuration and the possibility to implement a task set composed by periodic tasks, with implicit deadlines; the task sets implementation must provide some additional features in order to check, record and eventually skip the task deadline.

2 USART Interface

Starting from the basic considerations, both the STM32F4 and the Flex board are capable to be configured in order to communicate via the USART interface. This fact can be easily checked from the datasheets. To simplify the next steps, it was necessary to check if some libraries are already available in order to avoid additional work, that can cause a meaningless waste of time.

2.1 STM32F4 Board

The ST Microelectronics provides an entire set of low level instructions to configure, receive and send data on the USART interface. In addition, it provides a lots of demos that can be easily integrated both under Erika and FreeRTOS. Taking into account the precondition of no-interference, it has been decided to interact with the interface using the polling technique (hence, avoiding interrupts).

2.2 Flex Board

This board is equipped with a dsPIC, provided by Microchip. The manufacturer gives to the customers a set low level instructions to interact with the USART interface. In addition, it provides a demo to supports the implementation of such kind of interaction, both under Erika and FreeRTOS. It can be easily checked, that the two configurations in which the Flex board is involved can be successfully integrated with the operating systems chosen both using the polling technique and the interrupts, but considering the no-interference assumption, the first one was chosen.

2.3 Conclusion

Recapping the information from the two boards, both the STM32F4 and the FLEX board can easily supports the USART interface without any kind of modification, but a different choice has been taken: in order to simplify the next steps, it was decided to implement a library for each board, in order to provide a uniform interface towards the serial port, on top of which the two benchmark implementations, that are strongly operating system dependent, can be built.

3 Periodic Task Set

In order to enable the design of the two benchmark implementations, it is necessary to check whether the operating systems already support the implementation of periodic tasks. Moreover, the benchmark requires to check, record and eventually skip the task deadline, but since this operations are strictly related with the implementation they cannot be treated at this first level, so they will be analyzed in the next workpackage. The aim of this point is only to check the basic functionalities.

3.1 FreeRTOS

This operating system does not provide a direct support to the period task but anyway, it provides some basic functions that can be merged in order to produce tasks with a periodic pattern. Therefore, it is possible to implement a basic task set composed by periodic tasks with implicit deadlines using the two boards under FreeRTOS, since the porting for this two boards are already available.

3.2 Erika

This operating system directly supports the implementation of periodic task with their alarm structures. Therefore any kind of problem can arise from this operating system, in both board configurations, considering only a task set composed by periodic tasks with implicit deadlines, since these two operating system porting are already available.

3.3 Conclusion

Analyzing the results of the four configurations, as it was expected using real-time operating systems, both are compliant with the design of periodic tasks. Concluding, any kind of problem can arise from this first basic level of periodic task implementation.