Sistemas Embutidos e de Tempo Real
Implementing cooperative tasks in Zephyr

João Carrasco 80305, Daniel Oliveira 89208, and João Carvalho 106310

Universidade de Aveiro, Portugal

Abstract. Implementation of a set of cooperative real-time tasks in Zephyr. Basic processing of an analog signal.

1 Task scheduling

In this project we implemented two different systems with three task each (A, B and C). Thread A is responsible to read the value from the ADC and store it in a buffer. Thread B should use the values stored in the buffer to filter the average value. Thread C should use the value computed by thread B to output the correspondent intensity on the LED.

We use 2 different approaches:

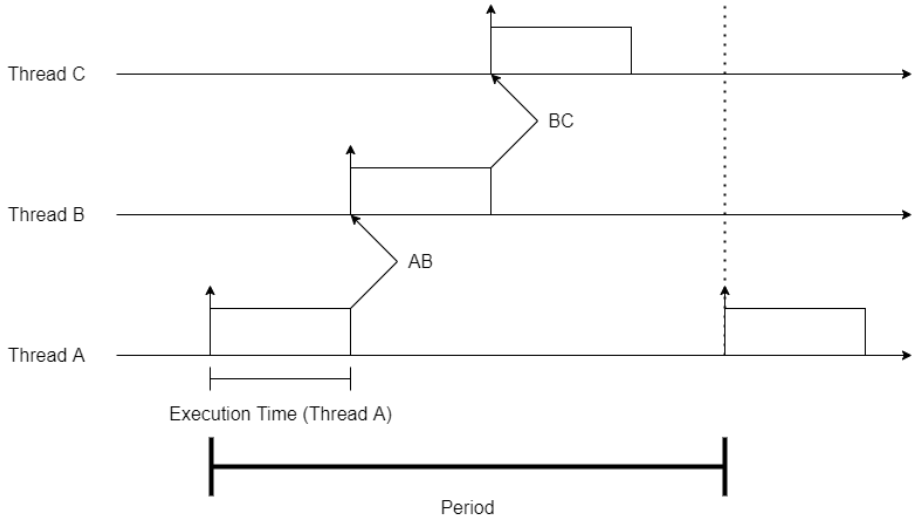
- **Shared memory + Semaphores** The goal behind the use of semaphores is for threads to wait for permission to proceed their task which can be given by other threads through signals modifying the value of the semaphore. Semaphores are thread safe, which means that conceding and waiting for permissions are atomic operations. In this project, there was a need for 2 semaphores, one between A and B and another between B and C.
- **FIFO Queue** The goal behind the use of fifos is for threads to either wait for at least one value to be inserted into the given queue, which means it will be on stand by while the queue is empty. Any task can read or write into a fifo. These fifos are thread safe, which means that inserting and waiting for values to be inputed are atomic operations. In this project, there was a need for 2 fifos, one between A and B and another between B and C.

2 Task Execution

June 1st, 2022

References

1. https://github.com/Joaoprcf/setrp4.git


Thread C
BC
Thread B
AB
Thread A
Execution Time (Thread A)
Period