universidade de aveiro

2021-22

Assignment 4

Implementing cooperative tasks in Zephyr

Autores: Bruno Feitais 93384

Afonso Lavrador

Turma: P2

Data: 31/05/2022

Docente: Pedro Pedreiras

Introduction:

The aim of this assignment is to learn how to implement a set of cooperative real-time tasks in Zephyr. Replicating the typical structure of embedded software, a mix of periodic and sporadic tasks will be considered. Two distinct IPC mechanism shall be used.

Specification:

The system to implement does a basic processing of an analog signal. It reads the input voltage from an analog sensor, digitally filters the signal and outputs it.

• Input sensor: Emulated by a 10 kΩ potentiometer, supplied by the DevKit 3 V supply (VDD).

• Digital filter: moving average filter, with a window size of 10 samples. Removes the outliers (10% or high deviation from average) and computes the average of the remaining samples.

• Output: pwm signal applied to one of the DevKit leds.

The system was structured with three tasks, namely one task for acquiring the sample, one for filtering and the other to output the signal. The sampling task is periodic, while the other two are sporadic, being activated when new data is available. The sampling period, in milliseconds, is specified in a macro called "SAMP_PERIOD_MS".

The application was developed using two different approaches for IPC:

• Shared memory + Semaphores

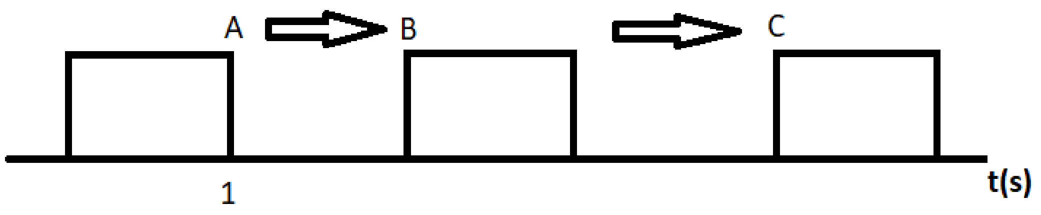
• FIFO Queue

Shared memory + Smaphores:

Thread A – Reads 10 ADC values and saves it on the shared memory.

Thread B – Gets the 10 ADC values and does the average, then it saves the average on the shared memory.

Thread C – Gets the average and sends it to the LED 1.



A – Ends after 1000ms, Thread A period is 1000ms.

B – Starts when Thread A finishes, and when its done sends a signal so task C knows when it can start.

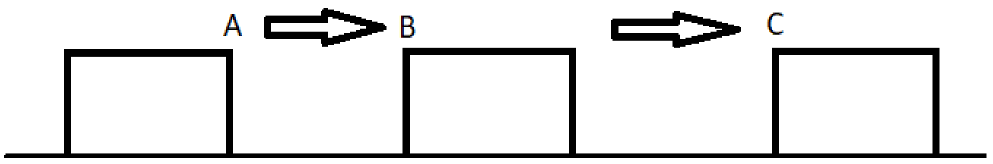
C – After doing everything it should do it sends a signal so task A knows that can start doing everything again.

FIFO:

Thread A – Reads 1 ADC value and sends it to the FIFO queu.

Thread B – Gets the ADC value and waits until it receives 10 values from task A, after that it does the average, then it sends the average to the FIFO queu.

Thread C – Gets the average and sends it to the LED 1.



A – Ends after 200ms, Thread A period is 200ms.

B – Starts when Thread A finishes, and when its done sends the value to FIFO queu so task C knows when it can start.

C – Reads the value of the FIFO queu and does everything that it should do, thread A starts after task C uses the last value on the FIFO queu.

GitHub:

On this link is possible to find the project repository containg the project code.

https://github.com/BrunoFeitais/assignment4SETR.git