OSES LAB #2 – messages

Please read the assignment description carefully

Purpose of the lab

This lab is intended to assess your knowledge of message-based communication and play with a basic MVC system.

Deliverables and deadlines

You shall provide a report (pdf file only) where you describe the oil file and the code you implemented for the exercise we proposed. The output produced by your solutions shall also be included. Insert in the report at least one SimulIDE screenshot that clearly shows the input/output signals and data discussed in the following.

The report shall be uploaded to the portale della didattica using the Elaborati section. The report shall be provided by October 31st, 2025, 18:00.

Exercise #1 (Arduino Uno on SimulIDE)

A system consists of three tasks:

A periodic task V drives an LED connected to GPIO pin 13 of the Arduino Uno. It makes the LED blink in 4 different ways depending on an `int` message it receives from task M. The effect of a message lasts until V receives a new message. More specifically:

- Message 0 turns the LED off.
- Message 1 makes the LED blink “slowly” (1Hz).
- Message 2 makes the LED blink “fast” (4Hz).
- Message 3 turns the LED on continuously.

The LED must be turned off initially.

A periodic task C samples a switch S connected to GPIO pin 12 (see the accompanying schematic `2023_lab02_simulide.simu`) and analog input voltage A0 every 100ms. It then sends an `int` messages to task M, encoded as follows:

- Bits 0… 9 represent the value of A0 expressed as an integer in the range [0, 1024[.
- Bit 12, if set, indicates the switch S has been pressed continuously for *at least* 1s.

A periodic task M runs every 500ms and, depending on the messages it receives from task C, performs the following actions, by sending appropriate messages to task V:

- If the switch S has been pressed continuously for at least 1s, the current value of A0 becomes the new *reference value* R.
- If no reference value has been set yet, the LED must be on continuously.
- If a reference value R has been set and the absolute value X of the difference between the current value of A0 and R is less than 100, the LED must be off.
- If X is at least 100, but less than 200, the LED must blink slowly.

- If X is 200 or more, the LED must blink fast.

Additional questions:

- Calculate the worst-case latency between a variation of A0 and the corresponding change of state of the LED.

Main design points and hints:

- Assign appropriate priorities to tasks M, V, and C, justifying your choices.
- Design an appropriate message-passing scheme among tasks M, V, and C. Explain and justify your design choices.
- Set the period of task V in a way that fulfills requirements and minimizes overheads.
- Design a set of test cases that exercises all possible actions task M may perform. Perform those tests and include the results in the report.
- Make sure you initialize analog and digital I/O lines properly before use.
- Make sure you correctly handle the case in which a "faster" task (smaller period) must send a message to a "slower" task (larger period).
- In case of doubts about the expected behavior of the code, run the example solution you can download from the course web site (.hex file for use in SimulIDE).

Exercise #2 (Arduino Uno on SimulIDE)

Make task M event-driven: Instead of running periodically, task M must be activated only when there is a message for it to process. Calculate the worst-case latency between a variation of A0 and the corresponding change of state of the LED again and compare it with the value you obtained in Exercise #1.