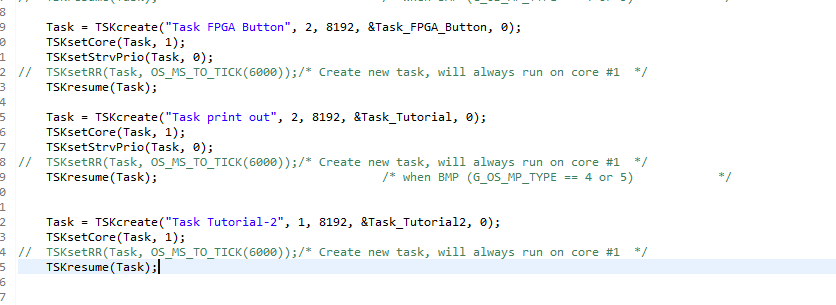
**LINGI 2315 - Evaluation 2 - mAbassi Project P1**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Delcoigne Ben | Noma | 38771700 |

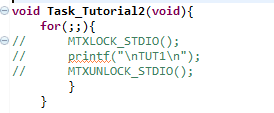
Definition of the tasks, the content of the new task and the implementation and explanation of the Starvation mechanism with relevant screenshots of your code

I first create all tasks, all running on the same core. I already added the starvation mecanism.

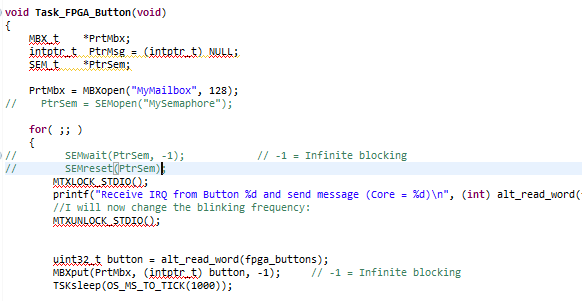
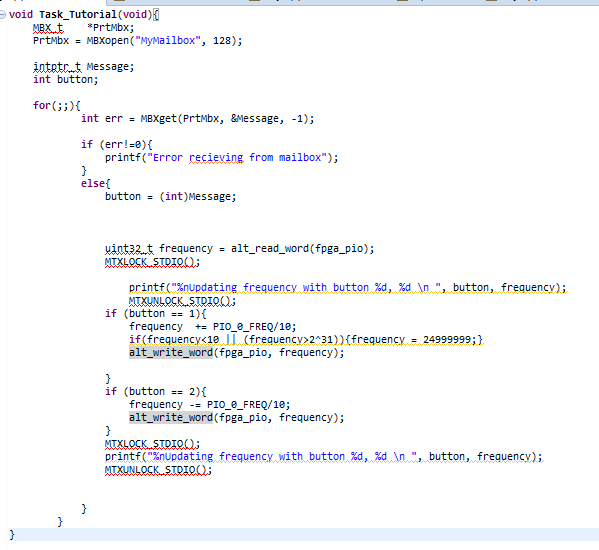
Note that task\_tutorial\_2, an infinite loop, has a higher priority than the two others.



Task tutorial 2 is just a simple while loop.



Task tutorail is in charge of opening the mailbox mymailbox and reading the messages from it. From the message recieved (which is the value of a button), it changes the frequency of the led (in the pio)



The last task, fpga button, is in charge of reading the buttons (every 1 second) and to updates the mailbox.

Since starvation is set to 1 for both tasks of priority 2, as they never run, they will increase their priority to priority 1 and become equals with the infinite loop. They thus now act all three in a round-robin manner. They do however have the autorisation to go to priority 0 but they do not need to do so.

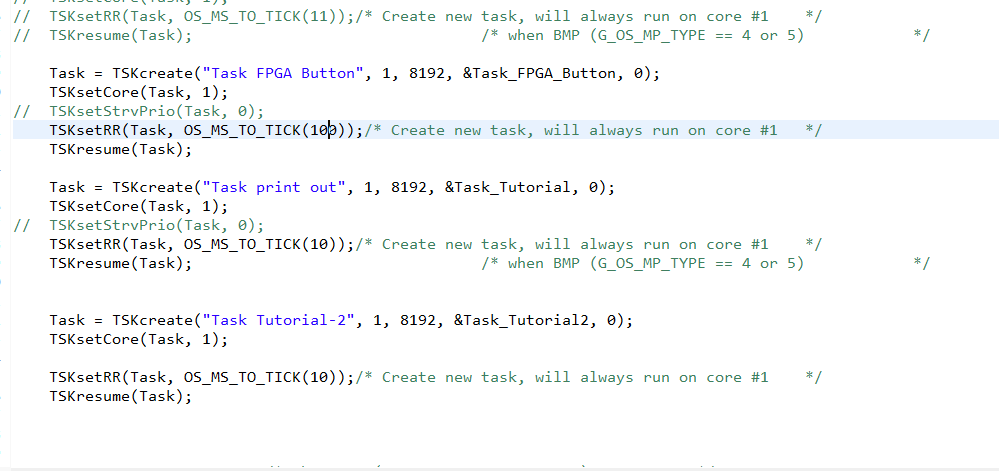
**LINGI 2315 - Evaluation 2 - mAbassi Project P2**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Delcoigne Ben | Noma | 38771700 |

Definition of the tasks, the implementation and explanation of the Round-Robin mechanism with relevant screenshots of your code

The only operated change was to remove the starvation and to set round robin timings:

I set all priorities to 1;



Task FPGA button runs for 100ms, reading the button and when an entry is new (a new input), it sends it to the mailbox.

Taskprint out runs for 10ms and opens the mailbox.Either it blocks on the call if it is empty, or it runs and updates the blinking frequency (no code change since page 1)

Task\_Tutorial\_2 is an infinite loop running for 10ms each cycle.

Total cycle time is thus 120ms.