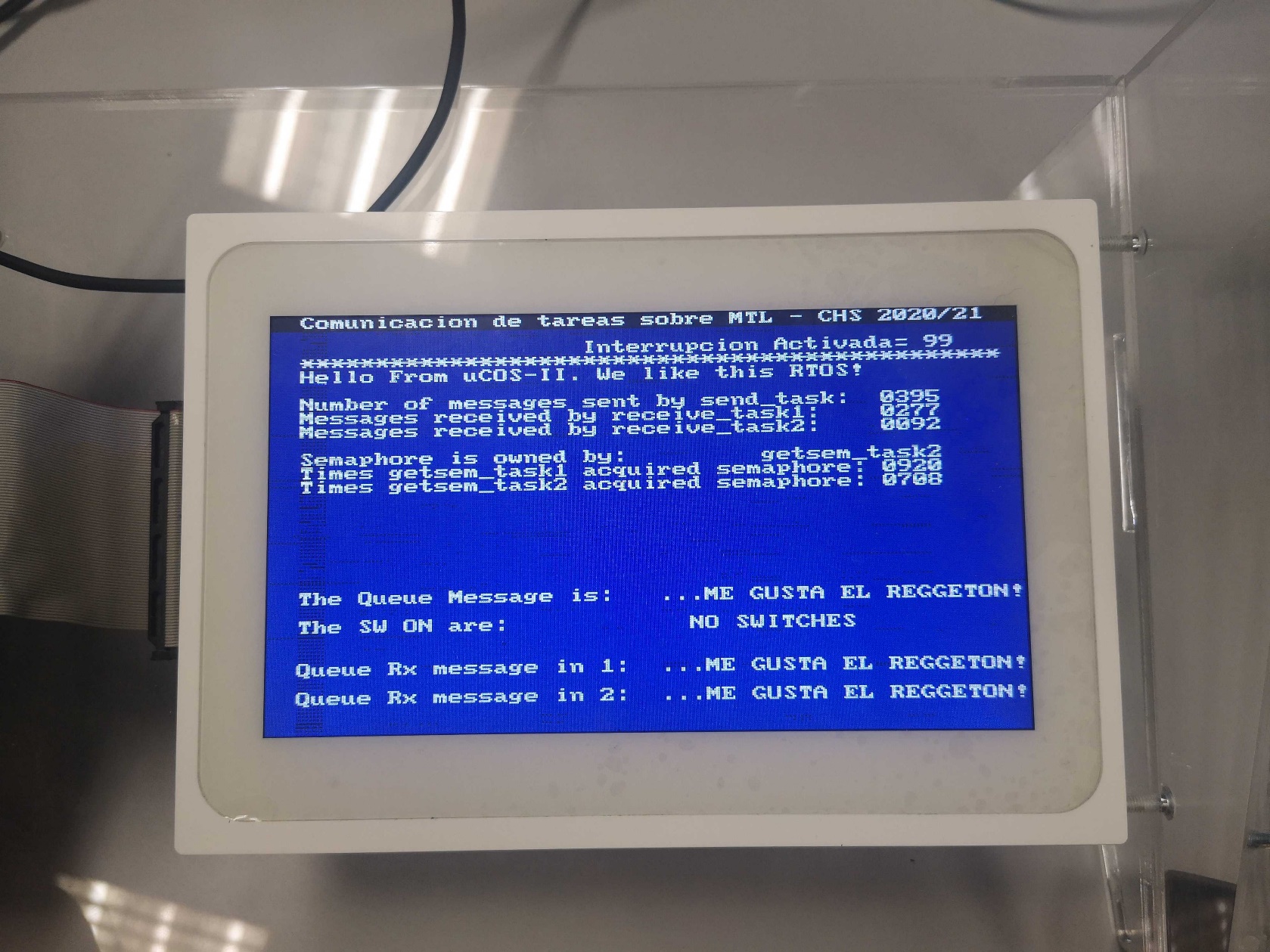
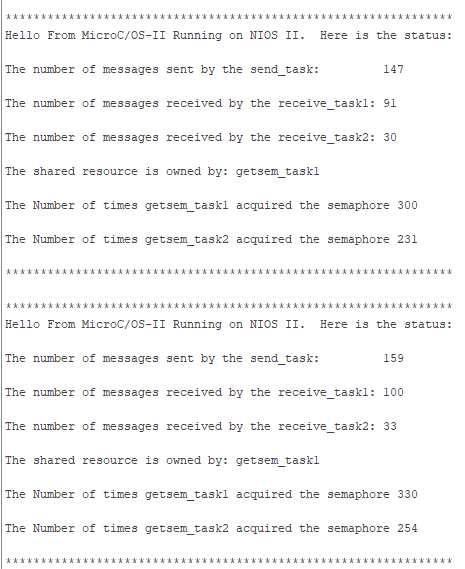
Practica 2 RTOS

Jose Luis Rocabado Rocha

1. Ejercicio 1

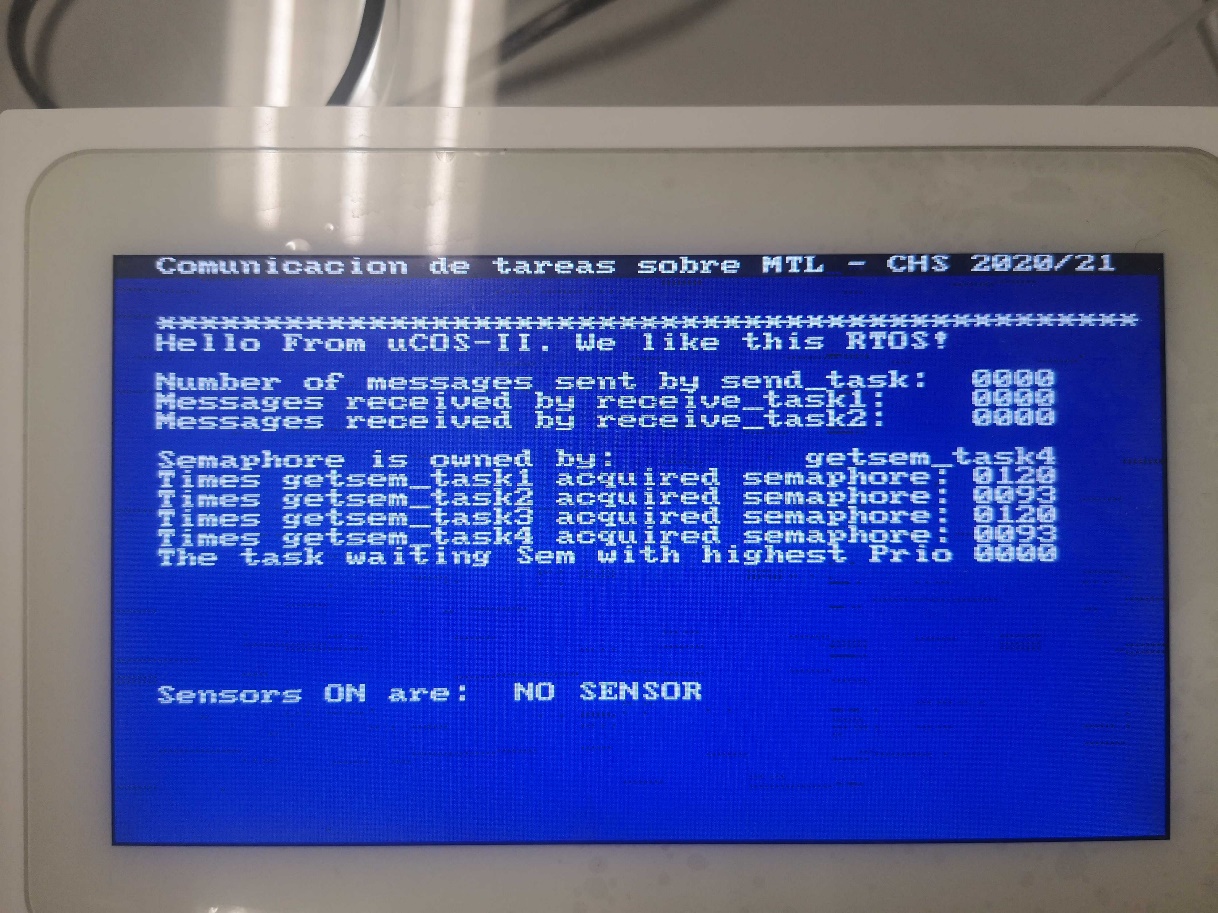
Se comprueba el flujo de ejecución del código así como la correcta ejecución del *status* de las tareas.



1. Ejercicio 2

Se comprueba que se representa el estatus de las tareas en la pantalla MTL.

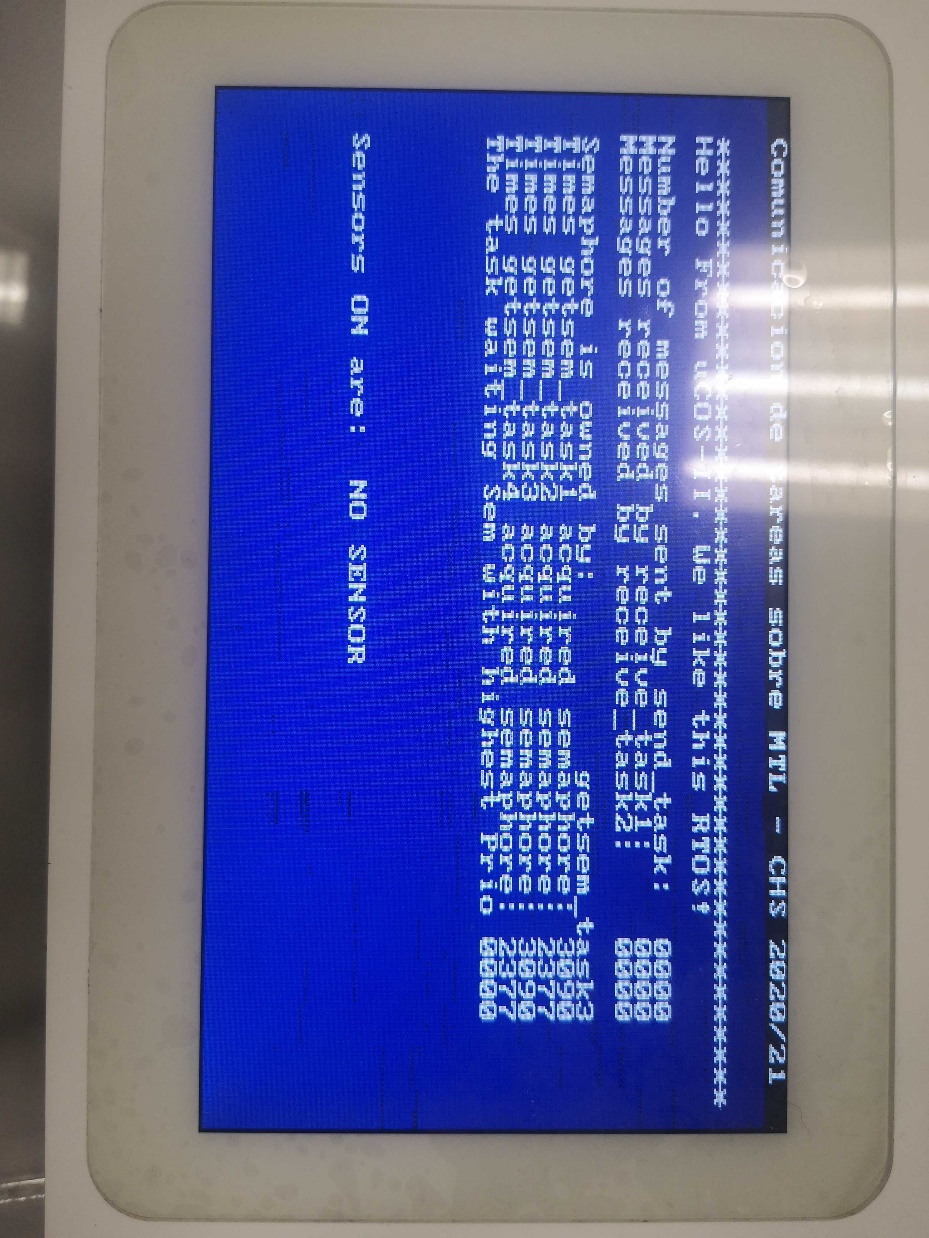
a-c) Funciona todo correctamente, no hay ninguna tarea que esté esperando al semáforo



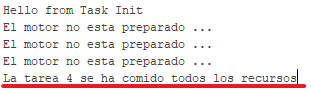
d) Al evitar que la task4 postee el semáforo, los recursos se quedan bloqueados y podemos observar que la tarea de mayor prioridad que está esperando a los recursos es la tarea task1 con prioridad 8. Se podría considerar un problema de inversión de prioridades por que la tarea de menos prioridad está utilizando completamente los recursos.



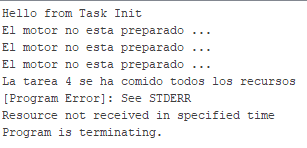
e) Funciona correctamente, no se retienen recursos



f) Se ejecutan las tareas hasta que la tarea 4 consume los 3 recursos obteniendo el mensaje siguiente por consola y haciendo que se pare la ejecución del programa

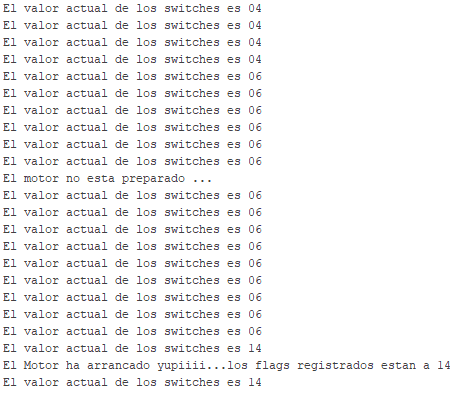


g) Por último se ejecuta todo correctamente hasta que la tarea 4 ocupa todos los recursos, por lo que genera que la tarea 3 genere un error de *timeout* ya que a los 100 *ticks* no ha recibido los recursos.



1. Ejercicio 4

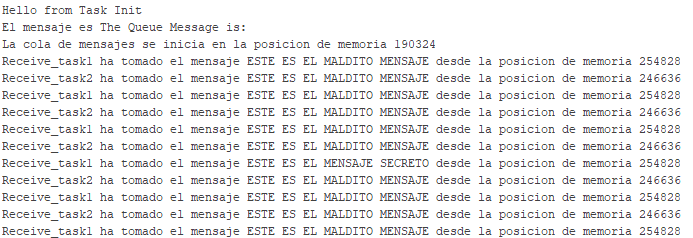
Se observa como las flags van cambiando y en el momento que se activa el swicth (Solo se capturan los sets, no los clears) y cuando se cumple el pend se enciende el motor.



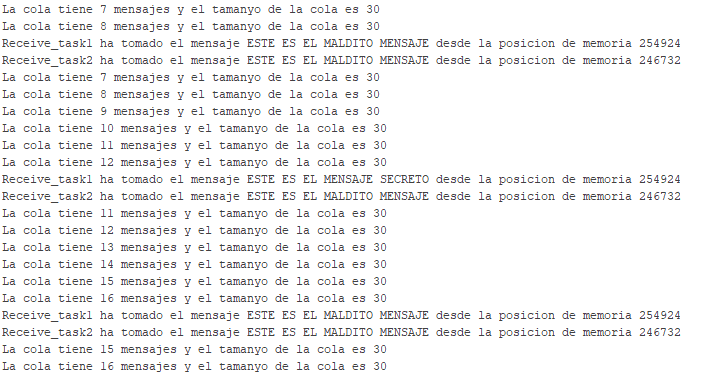


1. Ejercicio 5

Se obtiene el contenido de la cola y su tamaño dentro de las tareas consumidoras tal y como se muestra en la siguiente figura



Si añadimos el análisis de la cola, obtenemos la siguiente figura, donde se observa como los consumidores reciben los mensajes generados por el productor. También se aprecia cómo se generan más mensajes que los que se reciben.



Por último en el apartado extra, se comprueba el correcto funcionamiento de la cola en estilo LIFO y del *flush* de la cola en caso de que se detecten fallos como un *overflow* (cola con más de 9 mensajes genera un error que se pone delante de la cola para que pueda ser leído por la primera tarea que pueda procesar el error y continúe con la ejecución).

