UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

Máster Universitario en Ingeniería de Telecomunicación

Tarea 4 - II
Comunicaciones entre tareas en uCOS-II

CHS

Autor:
Andrés Ruz Nieto
Gerardo Arias Martínez

VALENCIA, 2022

TELECOM UPV VLC

Ejercicio 1

```
*********************************************************************
Hello From MicroC/OS-II Running on NIOS II.  Here is the status:

The number of messages sent by the send_task:         615

The number of messages received by the receive_task1: 442

The number of messages received by the receive_task2: 147

The shared resource is owned by: getsem_task2

The Number of times getsem_task1 acquired the semaphore 1470

The Number of times getsem_task2 acquired the semaphore 1132

*********************************************************************
```

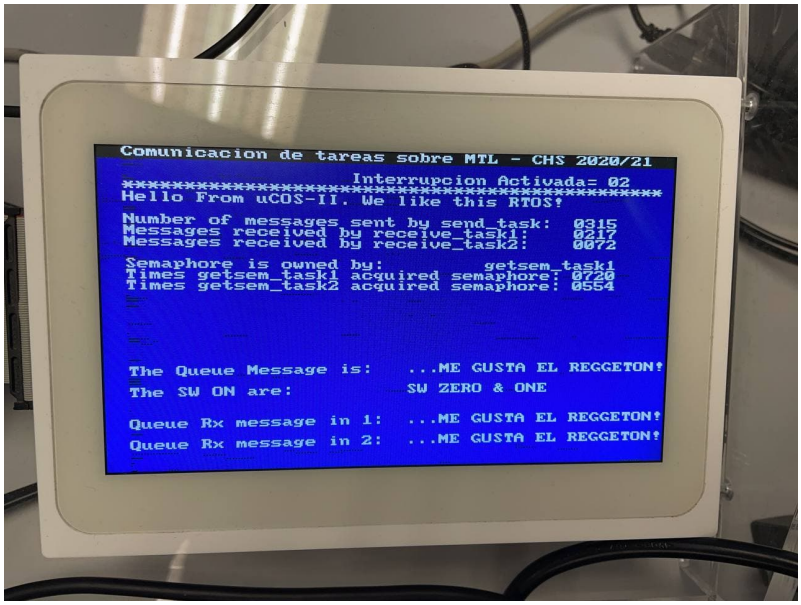


Figura 1: Con switches 0 y 1 activados

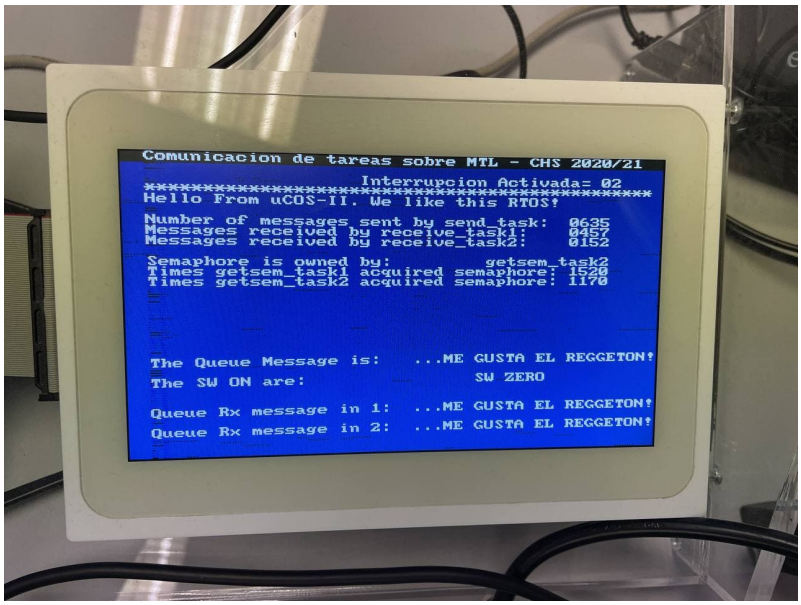


Figura 2: Con switch 0 activado

Figura 3: Sin switches activados

Ejercicio 2

Apartados a,b,c

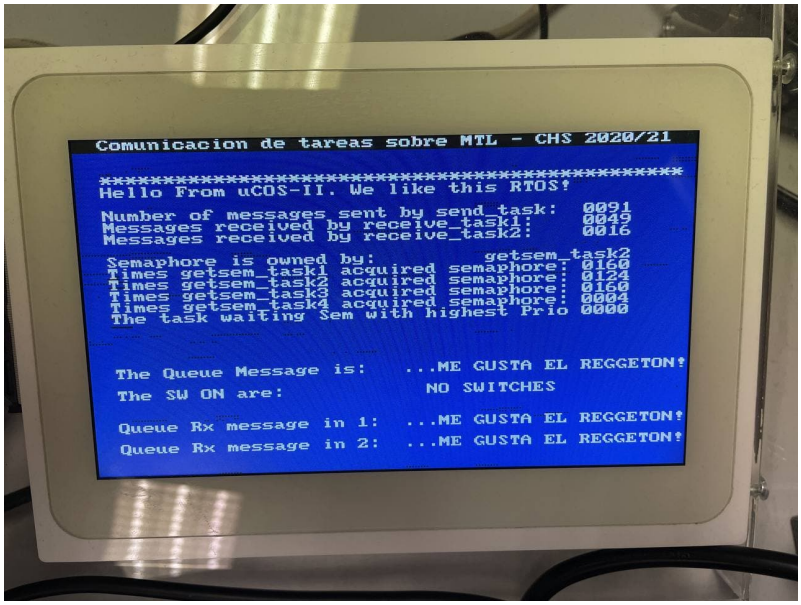


Figura 4: MTL con las tareas 3 y 4 programadas

La tarea que espera al semáforo es 0, esto es debido a la gran velocidad de las tareas y a que el query pregunta cada x tiempo, por lo que a este no le ha dado tiempo a ”pillar” a una tarea con el semáforo.

Apartado d

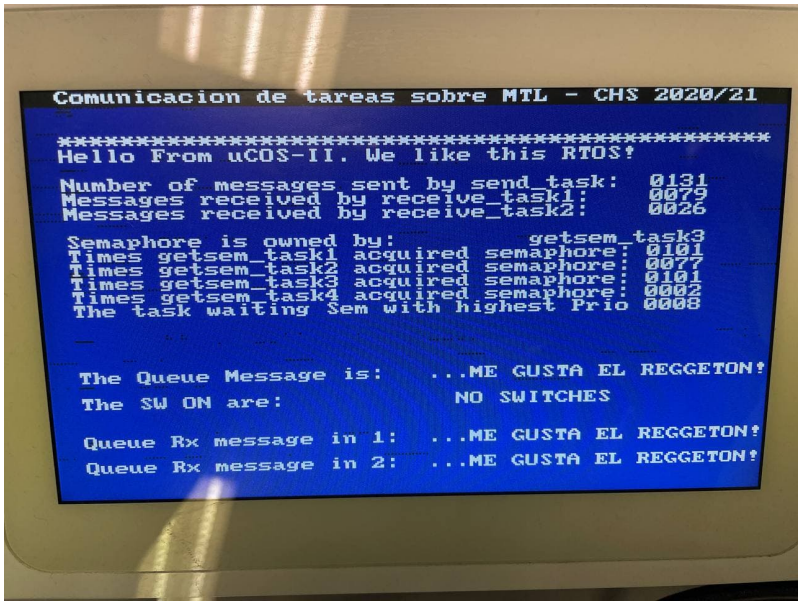


Figura 5: MTL con OSSemPost() comentado

```
Hello from Task Init
La tarea 4 se ha comido todos los recursos
```

Figura 6: Resultado por pantalla

La tarea con prioridad 8 (getsem_task1) se queda pendiente, ya que la tarea getsem_task4 no devuelve el semáforo

Apartados e,f

A la tercera vez que la tarea 4 coge el semáforo, al no devolver el semáforo se queda con todos los recursos y el resto de tareas no pueden acceder a él. Debido a ello, se genera una inversión de prioridades.

Ejercicio 3

Apartados a,b,c,d

```
El numero de veces que Task2_Mutex adquiere el mutex es: 02146
El numero de veces que Task1_Mutex adquiere el mutex es: 02147
La tarea que tiene el mutex es: 255
Y ha heredado la prioridad 04
El numero de veces que Task2_Mutex adquiere el mutex es: 02147
El numero de veces que Task1_Mutex adquiere el mutex es: 02148
La tarea que tiene el mutex es: 255
Y ha heredado la prioridad 04
El numero de veces que Task2_Mutex adquiere el mutex es: 02148
El numero de veces que Task1_Mutex adquiere el mutex es: 02149
La tarea que tiene el mutex es: 255
Y ha heredado la prioridad 04
El numero de veces que Task2_Mutex adquiere el mutex es: 02149
El numero de veces que Task1_Mutex adquiere el mutex es: 02150
La tarea que tiene el mutex es: 255
Y ha heredado la prioridad 04
El numero de veces que Task2_Mutex adquiere el mutex es: 02150
El numero de veces que Task1_Mutex adquiere el mutex es: 02151
La tarea que tiene el mutex es: 255
Y ha heredado la prioridad 04
```

Figura 7: MUTEX

Cuando la tarea que tiene el mutex es 255 (0xFF) es porque ninguna tarea lo tiene. La prioridad que se hereda al obtener el mutex es 4.

Al comentar OSMutexPost de getmutex_tast2, esta tarea se queda con el mutex

```
Hello from Task Init
El numero de veces que Task1_Mutex adquiere el mutex es: 021
El numero de veces que Task2_Mutex adquiere el mutex es: 021
La tarea que tiene el mutex es: 17
Y ha heredado la prioridad 04
```

Ejercicio 4

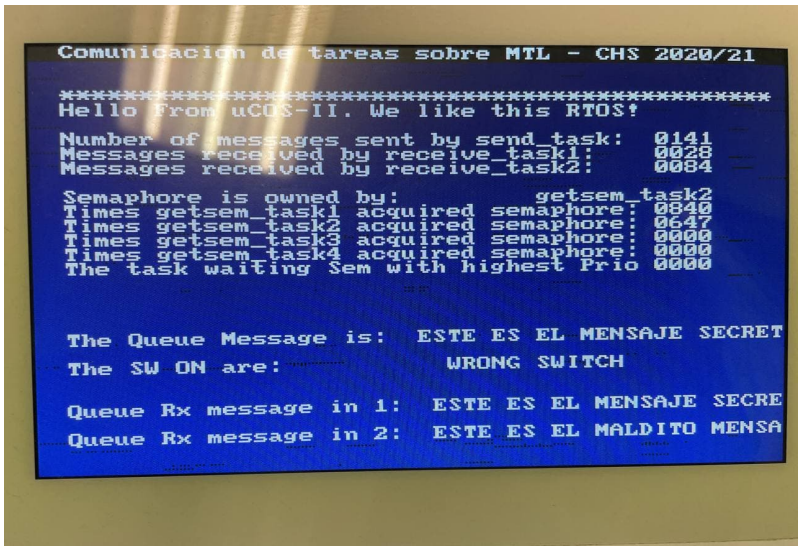
```
El Motor ha arrancado yupiiii...los flags registrados estan a 14
```

Tenemos una tarea del estado de switch que va posteando los flags de los ”sensores”. Después tenemos otra tarea que está mirando los sensores cada 2000 ticks de reloj, si pasa este tiempo y los flags de los sensores no están activos no arrancará el motor, en cambio, si todos los flags están a 1 sí arrancará. Para ello también se ha tenido que crear un FLAG GROUP

Ejercicio 5

Los receptores se quedan pendientes de la cola y los mensajes que recogen por el puntero lo emiten a través de la MTL.

```
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256220
La cola tiene 0 mensajes y el tamaño de la cola es 30
La cola tiene 1 mensajes y el tamaño de la cola es 30
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256220
La cola tiene 1 mensajes y el tamaño de la cola es 30
La cola tiene 2 mensajes y el tamaño de la cola es 30
Receive_task1 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 264412
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256220
La cola tiene 1 mensajes y el tamaño de la cola es 30
La cola tiene 2 mensajes y el tamaño de la cola es 30
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256220
La cola tiene 2 mensajes y el tamaño de la cola es 30
```



La cola se empieza a llenar hasta saturarse.

En el apartado d, si el mensaje es .Este es el mensaje secreto”la cola se vaciará

```
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256296
La cola tiene 28 mensajes y el tamaño de la cola es 30
La cola tiene 29 mensajes y el tamaño de la cola es 30
Receive_task2 ha tomado el mensaje ESTE ES EL MENSAJE SECRETO desde la posicion de memoria 256296
La cola tiene 29 mensajes y el tamaño de la cola es 30
Receive_task1 ha tomado el mensaje ESTE ES EL MENSAJE SECRETO desde la posicion de memoria 264488
La cola se ha vaciado
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256296
La cola tiene 0 mensajes y el tamaño de la cola es 30
La cola tiene 0 mensajes y el tamaño de la cola es 30
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256296
La cola tiene 0 mensajes y el tamaño de la cola es 30
La cola tiene 1 mensajes y el tamaño de la cola es 30
Receive_task2 ha tomado el mensaje ESTE ES EL MALDITO MENSAJE desde la posicion de memoria 256296
La cola tiene 1 mensajes y el tamaño de la cola es 30
La cola tiene 2 mensajes y el tamaño de la cola es 30
```