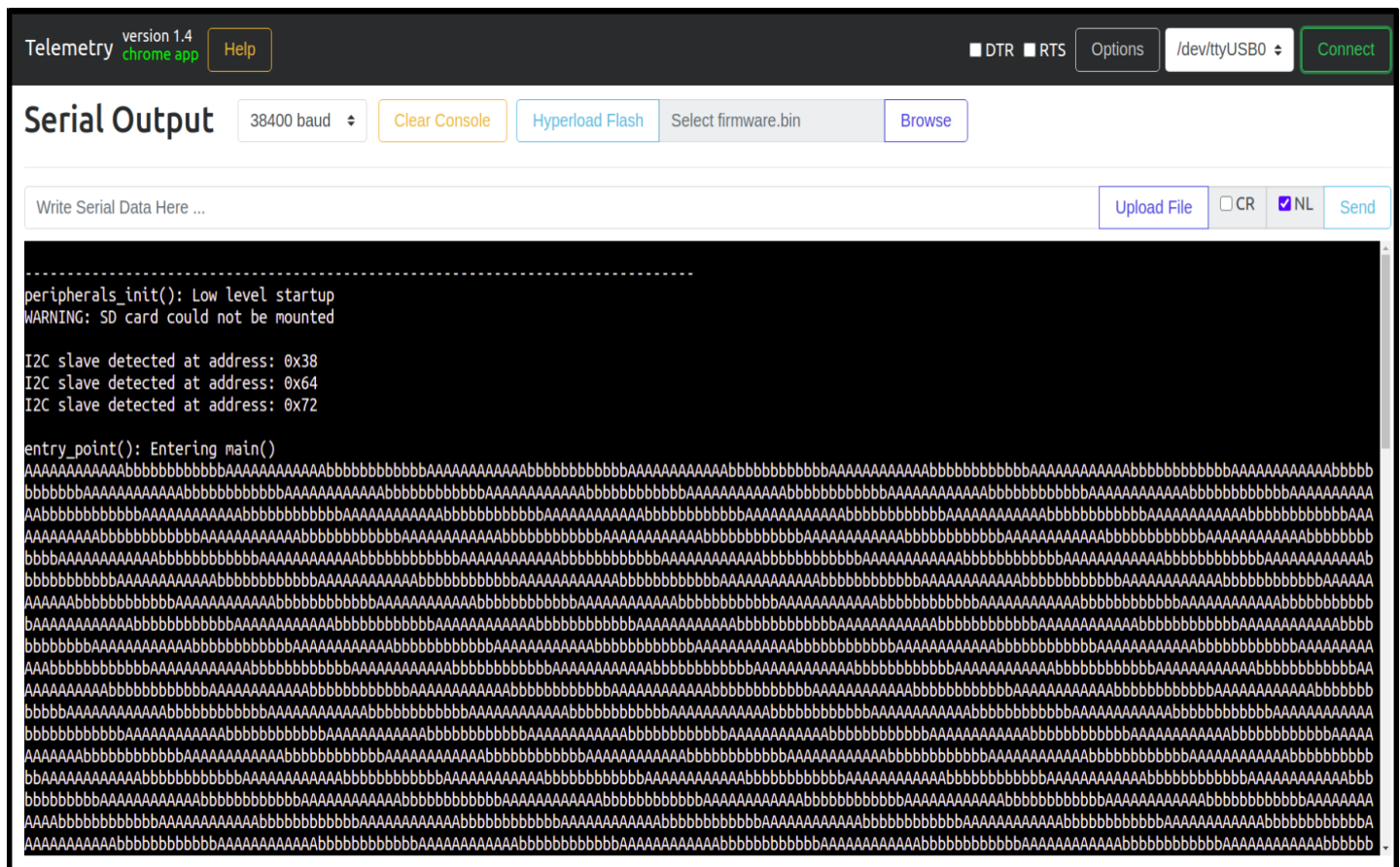


Assignment Title : Multiple Tasks
Date : 09/08/20
Submitted by : Tirth Pandya

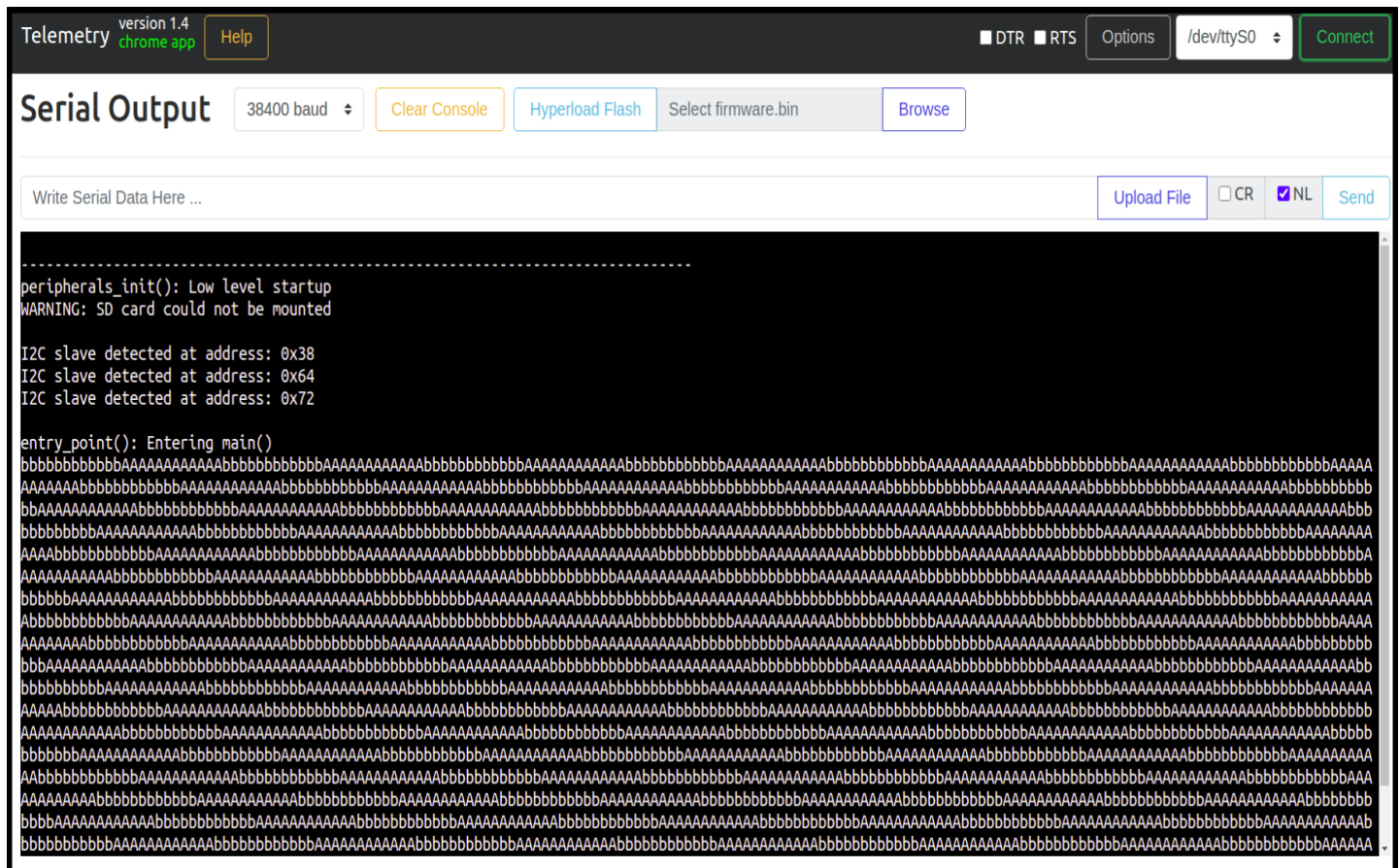
- Tasks:
The two tasks - task_one and task_two - print out “AAAAAAAAAAAA” and “bbbbbbbbbbbbbb” respectively.

Case 1: task_one high priority, task_two low priority



As we can see here, the content inside the task one that has higher priority gets printed first and then it goes to sleep for 100 ticks. When the task one sleeps, the task two that has lower priority takes over the cpu and prints its contents.

Case 2: task_one low priority, task_two high priority



Here, the task two that has higher priority over the task two gets executed first and when the task two goes to sleep, the task one takes over the cpu and prints its contents and then goes to sleep. It keeps repeating forever and.

Case 3: task one and task two having equal priority

As per the basic rule of the RTOS, whenever the priorities are equal for more than one tasks, they get time slices and the round-robin scheduler takes care of allocation.

We have configured the baud rate at 38400. Therefore, per one RTOS tick, 3.8 characters are printed and then task 2 gets its time slice. Therefore when the two tasks are assigned same priorities, we can see 3 or sometimes 4 characters being printed and then the other task takes over the cpu.

I have observed some irregularities though. Sometimes it prints 5 characters instead of 3 or 4. The number of characters in the string as well as the order of the 'xTaskCreate()' call also affects the output when the two priorities are equal.

Serial Output

38400 baud

Clear Console

Hyperload Flash

Select firmware.bin

[Browse](#)

Write Serial Data Here ...

Upload File

☐ CR

☒ NL

Send

```
peripherals_init(): Low level startup
WARNING: SD card could not be mounted
```

```
I2C slave detected at address: 0x38
I2C slave detected at address: 0x64
I2C slave detected at address: 0x72
```

```
entry_point(): Entering main()
```

[illegible]