|  |  |  |  |
| --- | --- | --- | --- |
| BN | Section | Student ID | اسم الطالب |
| **33** | **4** | **9220965** | **يسن عصام يسن احمد محمد** |
| **42** | **4** | **9220988** | **يوسف سمير عيد محمود زيدان** |

# System Design

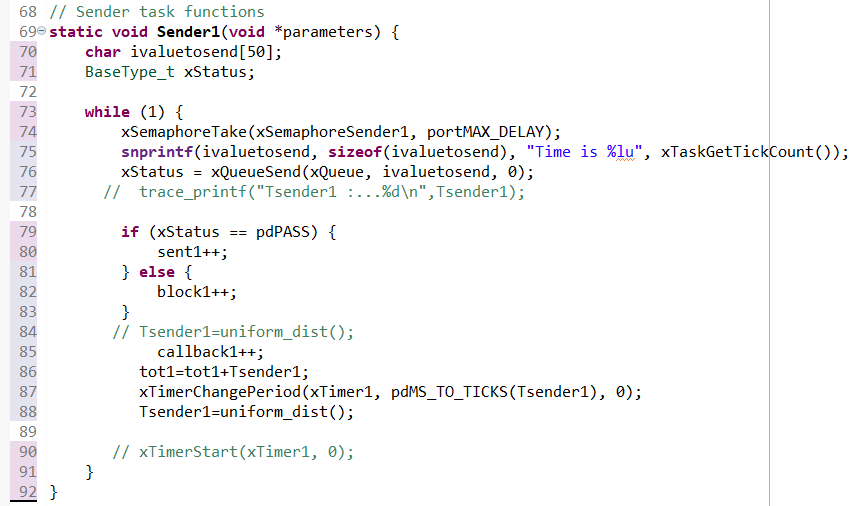
Project aims to generate random period for senders task between two boundaries using uniform distribution.There are one shared queue with fixed size between three senders tasks and receiver task. Each task have timer, callback function and semaphore. The sender or receiver task is trying to take its semaphore to send or receive, the role of callback function is to realise semaphore to each task. Making task to be blocked until timer expire and callback function release semaphore.

The highest priority of all tasks is receiver task , Then from the three senders one has higher priority than the last two, which are same priority. The timer of receiver is fixed with period 100ms, timer of the three senders task has a random period from lower and upper boundaries, if task has sent or received, task sleeps for the period of the its own timer. boundaries changes every successful 1000 received message using the reset function , the random period of senders tasks changes every time it expires in the same boundaries. the fixed size of queue is 3 for all boundaries then changed to be 10.

If the program is initially started, The reset function have to reset the number of sent, received and blocked messages, clear the queue and Updates the boundaries. Using new boundaries, if the total received message is 1000 then the reset function have to prints the number of total sent and total blocked messages for all tasks and for each task and reset the number of them and the received and check for the boundaries limit , if it not obtained it updates boundaries with new one , if limit has reached it destroy timers and end program and close the kit.

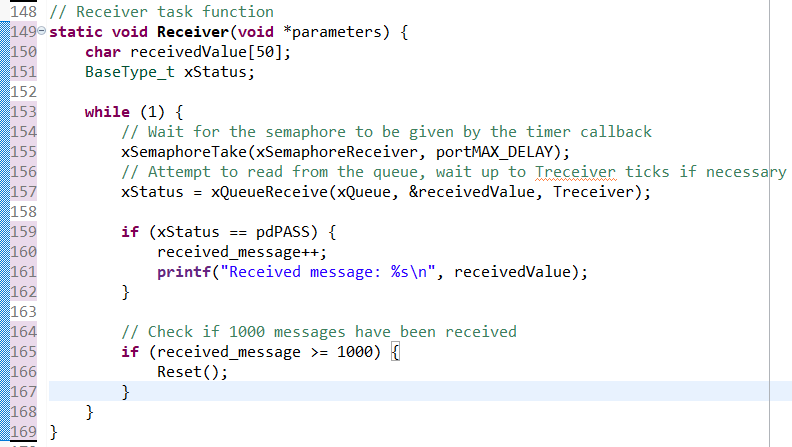
The sender task: example on first sender

Each sender task waits for the timer callback function to release the its semaphore, when it get unblocked it tries to send a message to the shared queue with the time , if it has sent it increments sent message of task , if it blocked it increments blocked message, then the task sleeps for the random period of its own timer , later on the average time is calculated.



The receiver task:

Receiver task waits for fixer timer callback function to release its semaphore, when it get unblocked it tries to receive from the queue if received it increments the total received message, check if number have reached 1000,if it reached then it reset function.



Start

Creating the queue and semaphores

Creating task with semaphores

RESET

Creating timers,random period for sender,fixed period for receiver

Game over

Increment blocked message

Sent?

yes yeye no

Reached limit of boundaries?

no

yes

yes

1000 received message?

Increment sent message

no

Increment received message

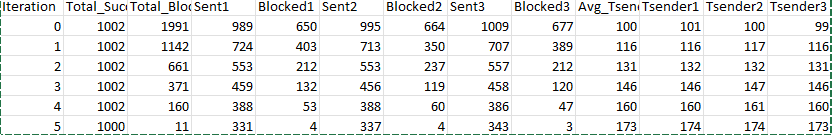
Figure 1: Detailed System Design Flow

yes

Received ?

# Results and Discussion

## For Queue Size 3



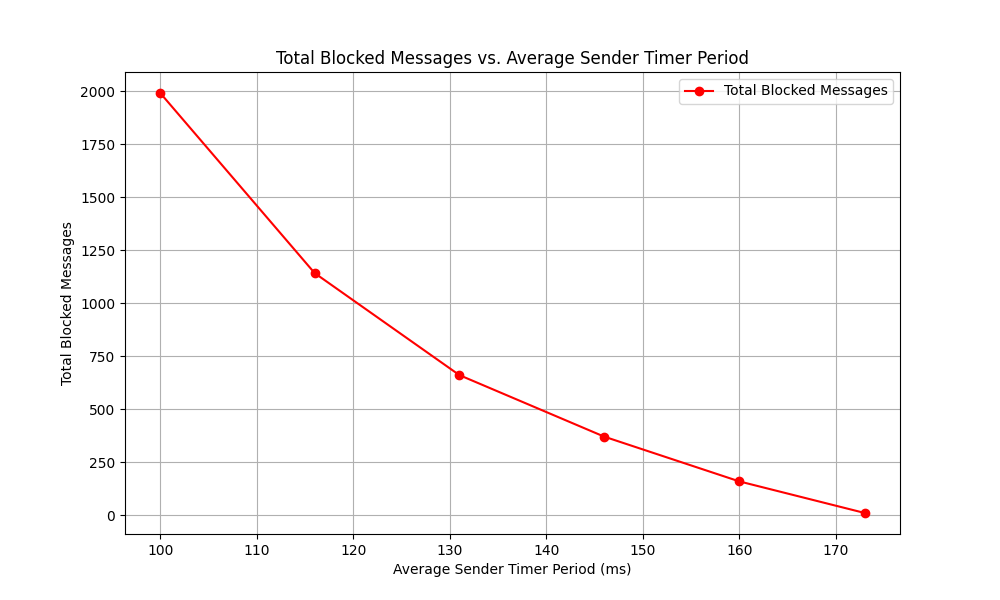


Figure 2: total blocked messages as function of

average sender timer period

as expected as when the period increase the chance of being blocked on the queue decreases and with the limited queue size in higher avg period it approaches 0 blocked massages here for the Sender3(higher priority) it reached 3 blocked massages in the last iteration

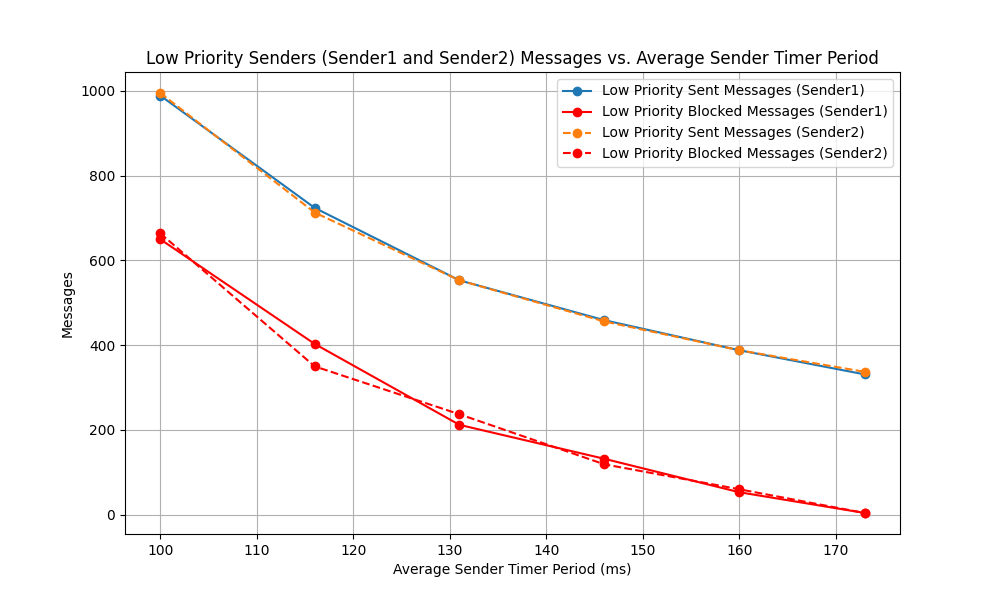


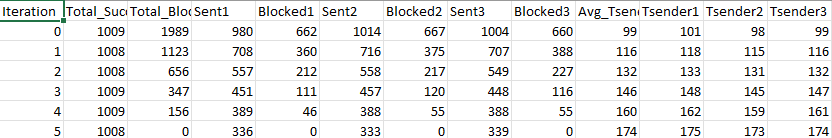
Figure 3: sent and blocked messages for lower priority sender tasks (Sender1 and Sender2)

We see Here the same trend of decreasing number of blocked messages with increasing the average period time but here for the 2 lower priority here send messages are blocked + successfully sent

sent+=block;

* Blocked Messages: High number of blocked messages at lower average sender timer periods.
* Sent Messages: The number of successfully sent messages increases with higher average sender timer periods.

## For Queue Size 10



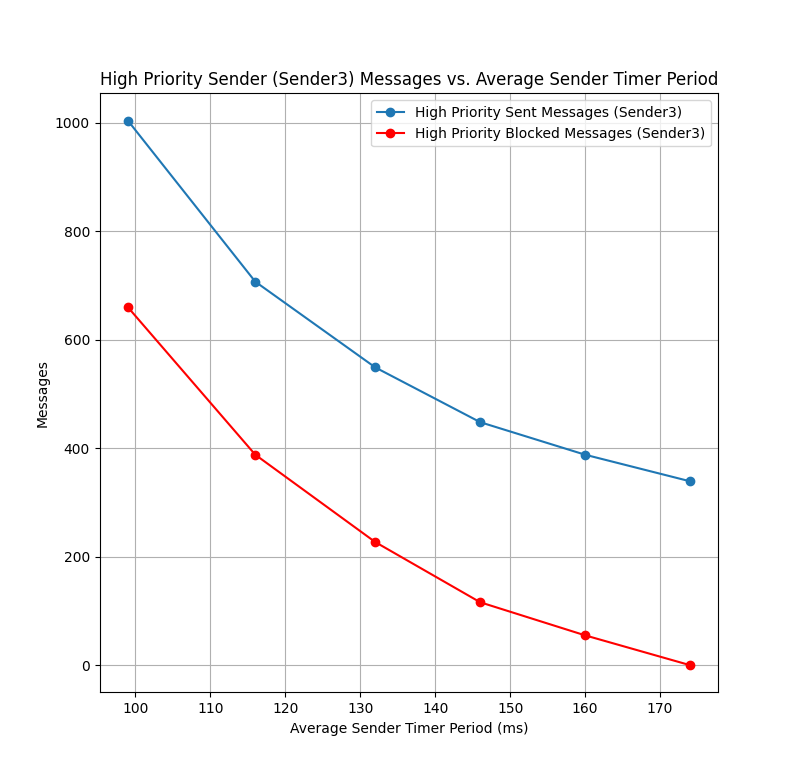
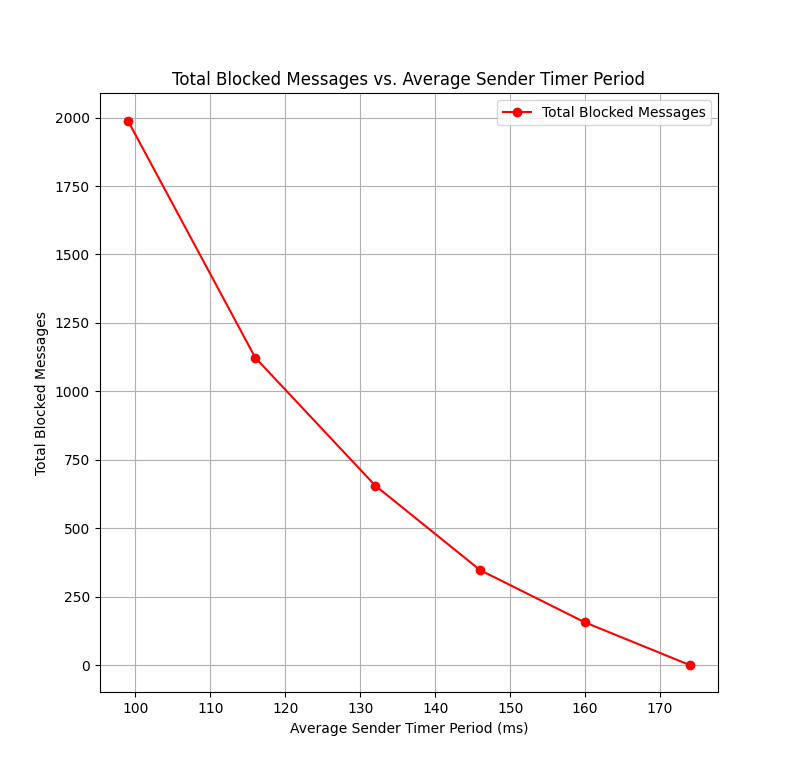


Figure 4: total blocked messages as function Figure 5: sent and blocked messages for high priority sender of average sender timer period task (Sender3)

Here the number of blocked reaches 0 in the last iteration because of the bigger queue size and higher Sender period

* Blocked Messages: Significantly reduced number of blocked messages due to increased queue capacity.
* Sent Messages: Nearly all messages are successfully sent even at lower average sender timer periods.

# Console output and Random Number generation

This is the output of the program it says Time is XYZ

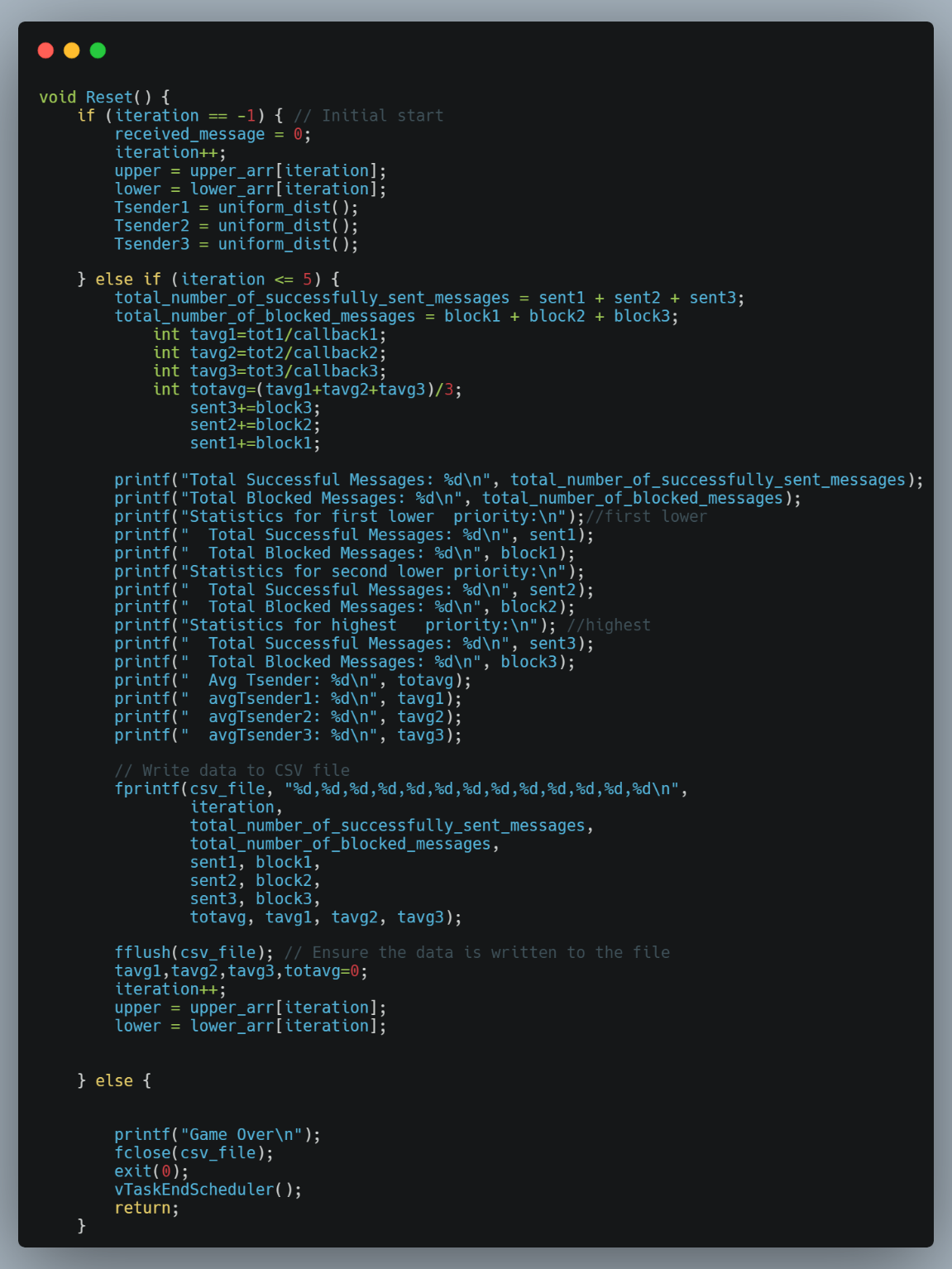
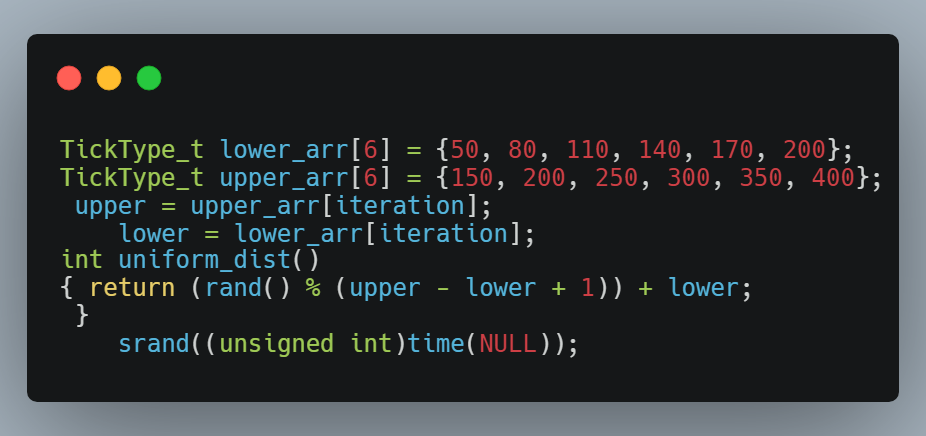
XYZ is the time in system ticks

Figure 6: Output in the console

References

1. “FreeRTOS\_Reference\_Manual”
2. David E. Simon, "An Embedded Software Primer".

## Code Snippets



* This is the Reset function which is responsible for printing and saving the statistics of and updating the values of Tavg and total Avg and total successfully send and blocked messages in each iteration and updating the values of Tsender for the first time also it handles the end of the program to ensure propre closure of csv file and closing exiting with exit(0);

* This a code snippet of how we generate a Random Number in uniform distribution with upper and lower bound