3. Typescript and screenshots

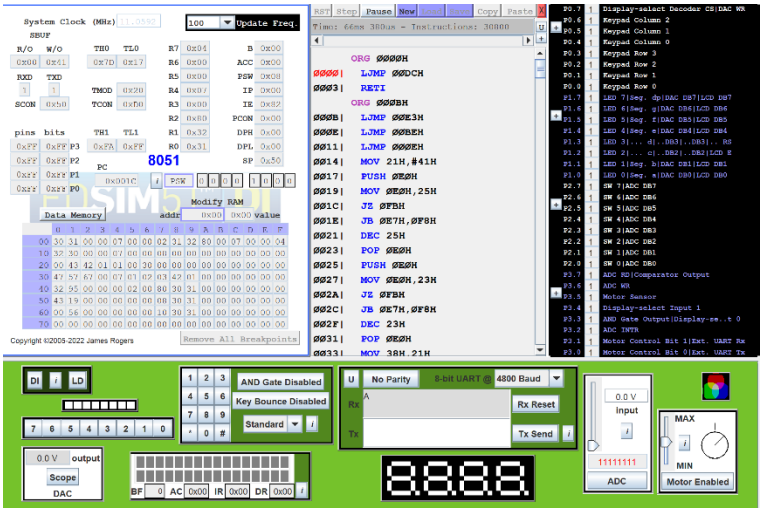
3.1 Typescript for compilation

```
PS C:\Users\peggy\OS Checkpoint\OS-checkpoint-4> make clean
del *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
PS C:\Users\peggy\OS Checkpoint\OS-checkpoint-4> make
sdcc -c  testpreempt.c
testpreempt.c:91: warning 158: overflow in implicit constant conversion
sdcc -c  preemptive.c
sdcc  -o testpreempt.hex testpreempt.rel preemptive.rel
```

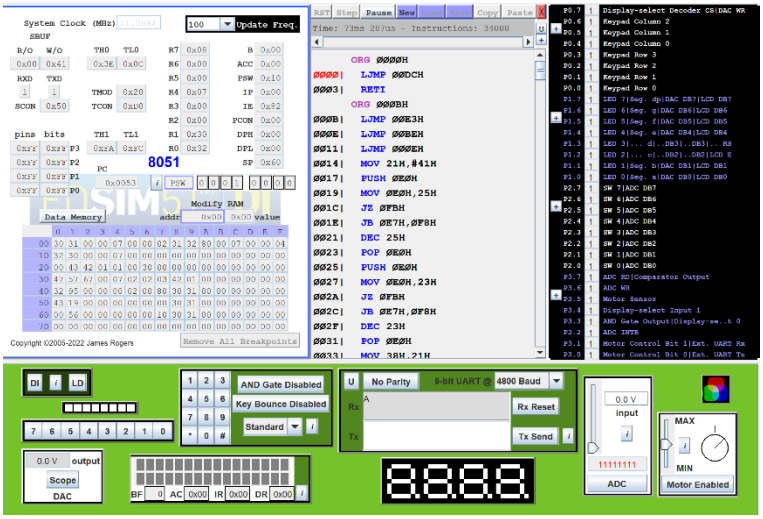
3.2 Screenshots and explanation

- **Screenshot when the Producer is running and show semaphore changes.**

Current thread ID (stored at 0x35) is 1, which is the thread ID of Producer1. The semaphore `empty` (stored at 0x25) changes into 0, which indicates there's no empty buffer. The semaphore `full` (stored at 0x24) changes into 1, which indicates there is 1 full buffer.

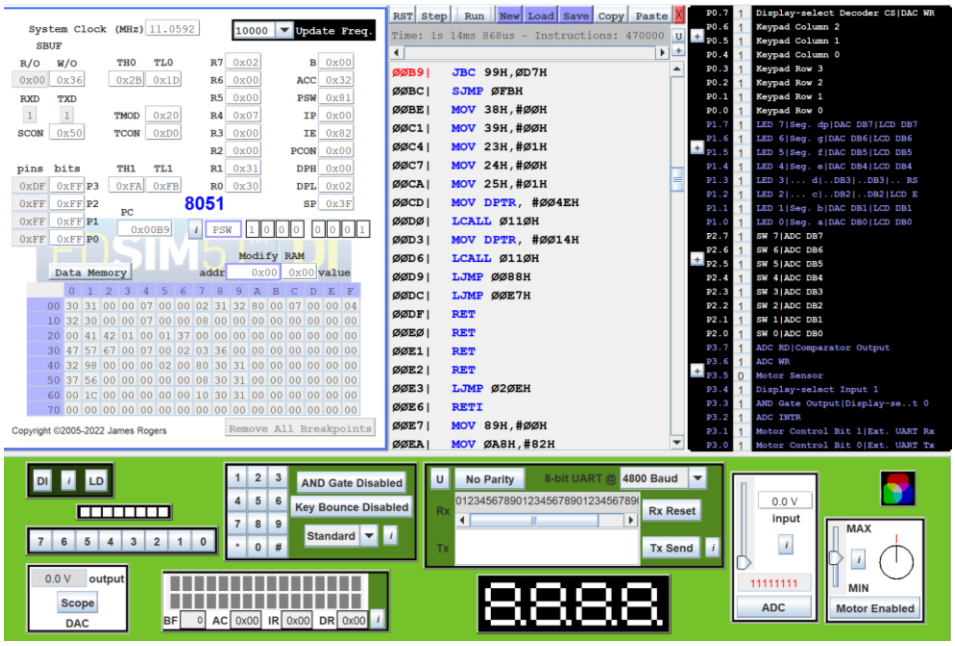


Current thread ID (stored at 0x35) is 1, which is the thread ID of Producer1. The semaphore `empty` (stored at 0x25) changes into 0, which indicates there's no empty buffer. The semaphore `full` (stored at 0x24) changes into 1, which indicates there is 1 full buffer.



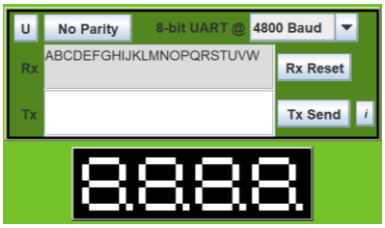
· **Screenshot when the Consumer is running and show semaphore changes.**

Current thread ID (stored at 0x35) is 0, which is the thread of Consumer. The semaphore `full` (stored at 0x25) changes into 0, which indicates there's no full buffer. The semaphore `empty` (stored at 0x24) changes into 1, which indicates there is 1 empty buffer.



2. Fairness

Here I use round-robin scheduling policy. Hence, one of the producer will suffer from starvation.

	data memory	output
Producer1 is created first		
Producer2 is created first