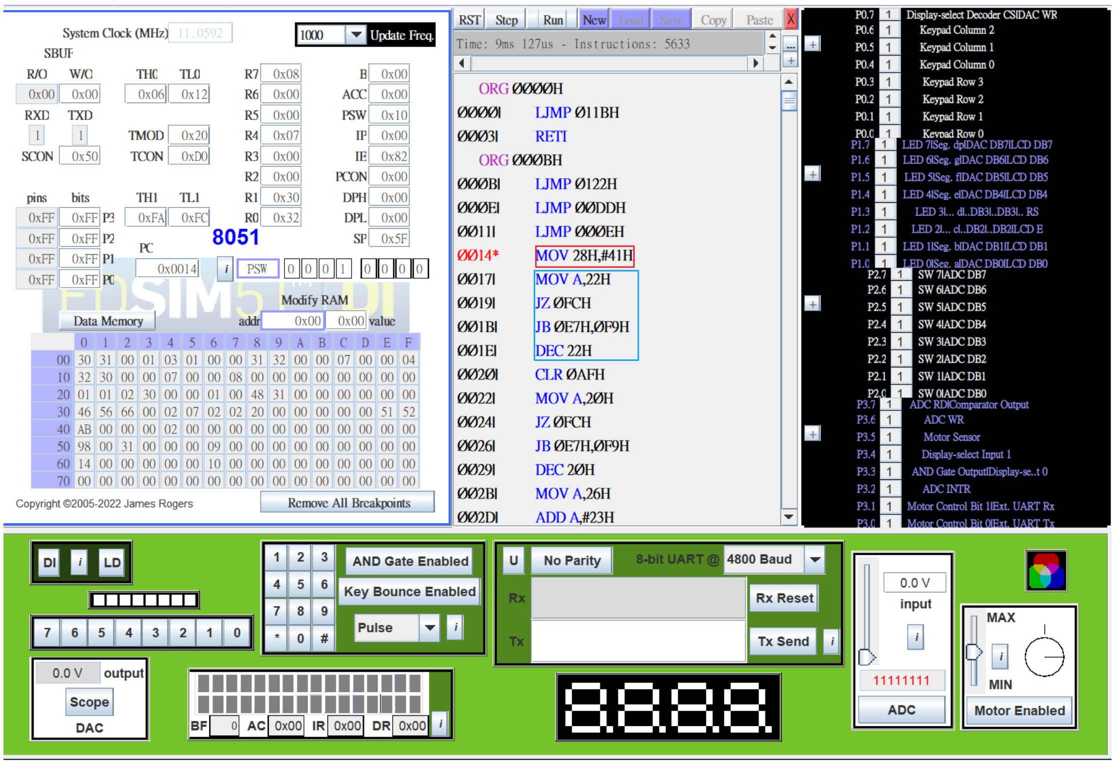
109062318 簡弘哲

1. Producer1 running & semaphore changes

   According to .map file, the address of _Producer1 is 0014, so I set a breakpoint there. Red block corresponds to the variable initialization ch='A', and blue block is the assembly code of SemaphoreWaitBody and decrease Semaphore 'empty' (addr = 22H) by 1

/ersion 2.1.32 & Dynamic Interface x | test3threads.hex



```
System Clock (MHz) 11.0592        1000  Update Freq.

R/O 0x00   W/O 0x00   TH0 0x06   TL0 0x12
RXD 1      TXD 1      TMOD 0x20
SCON 0x50             TCON 0xD0
pins/bits: P3 0xFF 0xFF   P2 0xFF 0xFF   P1 0xFF 0xFF   P0 0xFF 0xFF
TH1 0xFA   TL1 0xFC
R7 0x08  R6 0x00  R5 0x00  R4 0x07  R3 0x00  R2 0x00  R1 0x30  R0 0x32
B 0x00  ACC 0x00  PSW 0x10  IP 0x00  IE 0x82  PCON 0x00  DPH 0x00  DPL 0x00  SP 0x5F
PC 0x0014     PSW 0 0 0 1  0 0 0 0
Modify RAM  addr 0x00  value 0x00
```

Data Memory

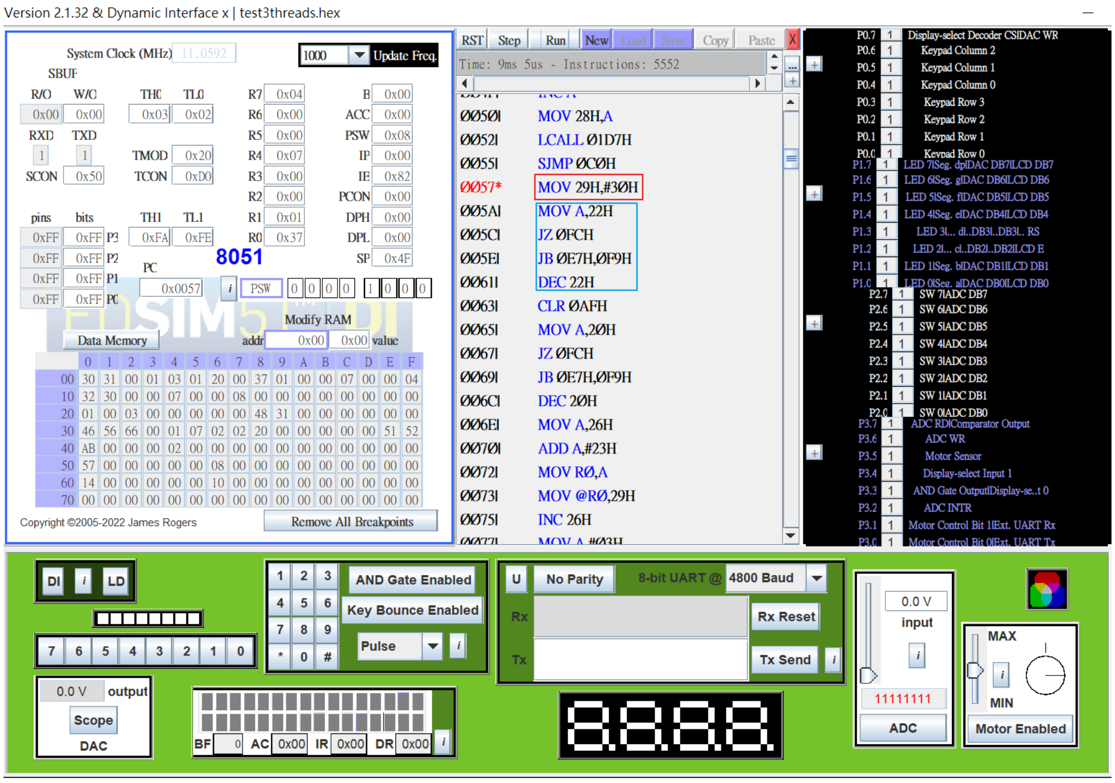
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	30	31	00	01	03	01	00	00	31	32	00	00	07	00	00	04
10	32	30	00	00	07	00	00	08	00	00	00	00	00	00	00	00
20	01	01	02	30	00	00	01	00	48	31	00	00	00	00	00	00
30	46	56	66	00	02	07	02	02	20	00	00	00	00	00	51	52
40	AB	00	00	00	02	00	00	00	00	00	00	00	00	00	00	00
50	98	00	31	00	00	00	09	00	00	00	00	00	00	00	00	00
60	14	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2022 James Rogers

```
Time: 9ms 127us - Instructions: 5633

      ORG ØØØØH
ØØØØ|      LJMP Ø11BH
ØØØ3|      RETI
      ORG ØØØBH
ØØØB|      LJMP Ø122H
ØØØE|      LJMP ØØDDH
ØØ11|      LJMP ØØØEH
ØØ14*      MOV 28H,#41H
ØØ17|      MOV A,22H
ØØ19|      JZ ØFCH
ØØ1B|      JB ØE7H,ØF9H
ØØ1E|      DEC 22H
ØØ2Ø|      CLR ØAFH
ØØ22|      MOV A,2ØH
ØØ24|      JZ ØFCH
ØØ26|      JB ØE7H,ØF9H
ØØ29|      DEC 2ØH
ØØ2B|      MOV A,26H
ØØ2D|      ADD A,#23H
```

2. Producer2 running & semaphore changes

   According to .map file, the address of _Producer2 is 0057, so I set a breakpoint there. Red block corresponds to the variable initialization num='0', and blue block is the assembly code of SemaphoreWaitBody and decrease Semaphore 'empty' (addr = 22H) by 1



Version 2.1.32 & Dynamic Interface x | test3threads.hex

System Clock (MHz) 11.0592 — 1000 Update Freq.

SBUF R/O	0x00	TH0	0x03	R7	0x04	B	0x00
SBUF W/O	0x00	TL0	0x02	R6	0x00	ACC	0x00
RXD	1	TMOD	0x20	R5	0x00	PSW	0x08
TXD	1	TCON	0xD0	R4	0x07	IP	0x00
SCON	0x50	TH1	0xFA	R3	0x00	IE	0x82
P3	0xFF 0xFF	TL1	0xFE	R2	0x00	PCON	0x00
P2	0xFF 0xFF	PC	0x0057	R1	0x01	DPH	0x00
P1	0xFF 0xFF	PSW	0 0 0 0 1 0 0 0	R0	0x37	DPL	0x00
P0	0xFF 0xFF					SP	0x4F

Modify RAM addr 0x00 value 0x00

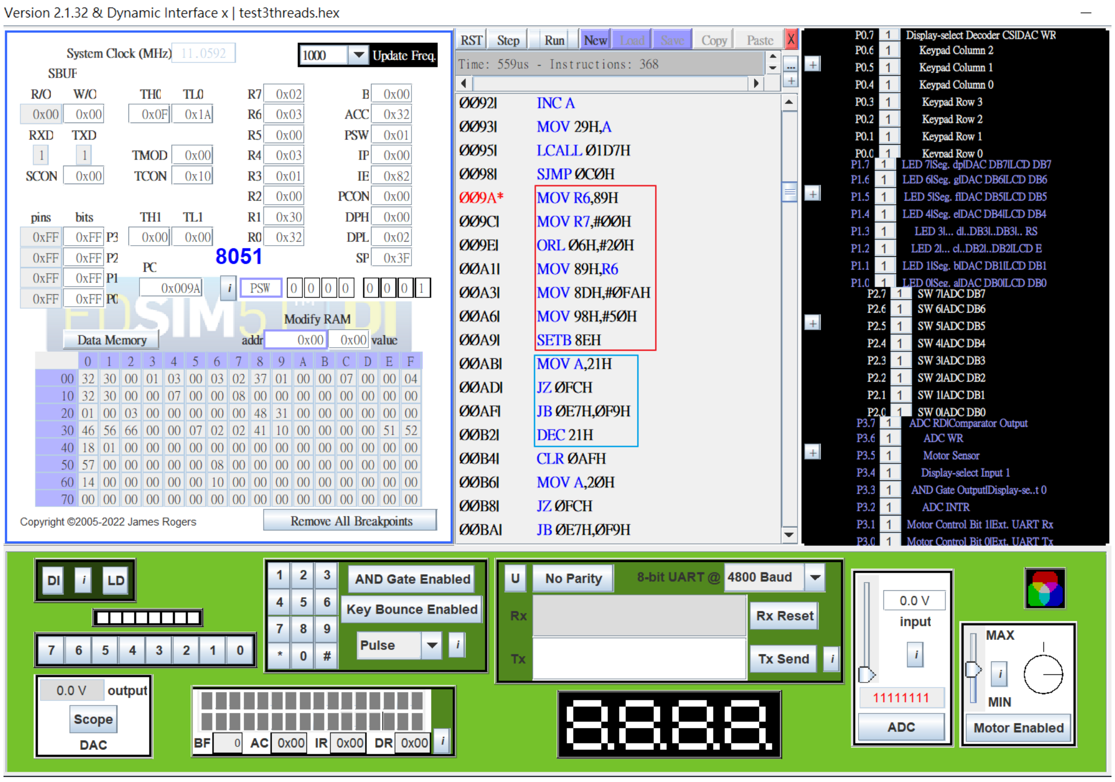
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	30	31	00	01	03	01	20	00	37	01	00	00	07	00	00	04
10	32	30	00	00	07	00	00	08	00	00	00	00	00	00	00	00
20	01	00	03	00	00	00	00	00	48	31	00	00	00	00	00	00
30	46	56	66	00	01	07	02	02	20	00	00	00	00	00	51	52
40	AB	00	00	00	02	00	00	00	00	00	00	00	00	00	00	00
50	57	00	00	00	00	00	08	00	00	00	00	00	00	00	00	00
60	14	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2022 James Rogers

```
Time: 9ms 5us - Instructions: 5552

ØØ5Ø   MOV 28H,A
ØØ52   LCALL Ø1D7H
ØØ55   SJMP ØCØH
ØØ57*  MOV 29H,#3ØH
ØØ5A   MOV A,22H
ØØ5C   JZ ØFCH
ØØ5E   JB ØE7H,ØF9H
ØØ61   DEC 22H
ØØ63   CLR ØAFH
ØØ65   MOV A,2ØH
ØØ67   JZ ØFCH
ØØ69   JB ØE7H,ØF9H
ØØ6C   DEC 2ØH
ØØ6E   MOV A,26H
ØØ7Ø   ADD A,#23H
ØØ72   MOV RØ,A
ØØ73   MOV @RØ,29H
ØØ75   INC 26H
```

3. Consumer running & semaphore changes

According to .map file, the address of _Consumer is 009A, so I set a breakpoint there. Red block corresponds to the initialization of TMOD, TH1…, and blue block is the assembly code of SemaphoreWaitBody and decrease Semaphore 'full'(addr = 21H) by 1



Version 2.1.32 & Dynamic Interface x | test3threads.hex

System Clock (MHz) 11.0592 — 1000 Update Freq.

SBUF R/O	0x00	TH0	0x0F	R7	0x02	B	0x00
SBUF W/O	0x00	TL0	0x1A	R6	0x03	ACC	0x32
RXD	1	TMOD	0x00	R5	0x00	PSW	0x01
TXD	1	TCON	0x10	R4	0x03	IP	0x00
SCON	0x00	TH1	0x00	R3	0x01	IE	0x82
P3	0xFF 0xFF	TL1	0x00	R2	0x00	PCON	0x00
P2	0xFF 0xFF	PC	0x009A	R1	0x30	DPH	0x00
P1	0xFF 0xFF	PSW	0 0 0 0 0 0 0 1	R0	0x32	DPL	0x02
P0	0xFF 0xFF					SP	0x3F

Modify RAM addr 0x00 value 0x00

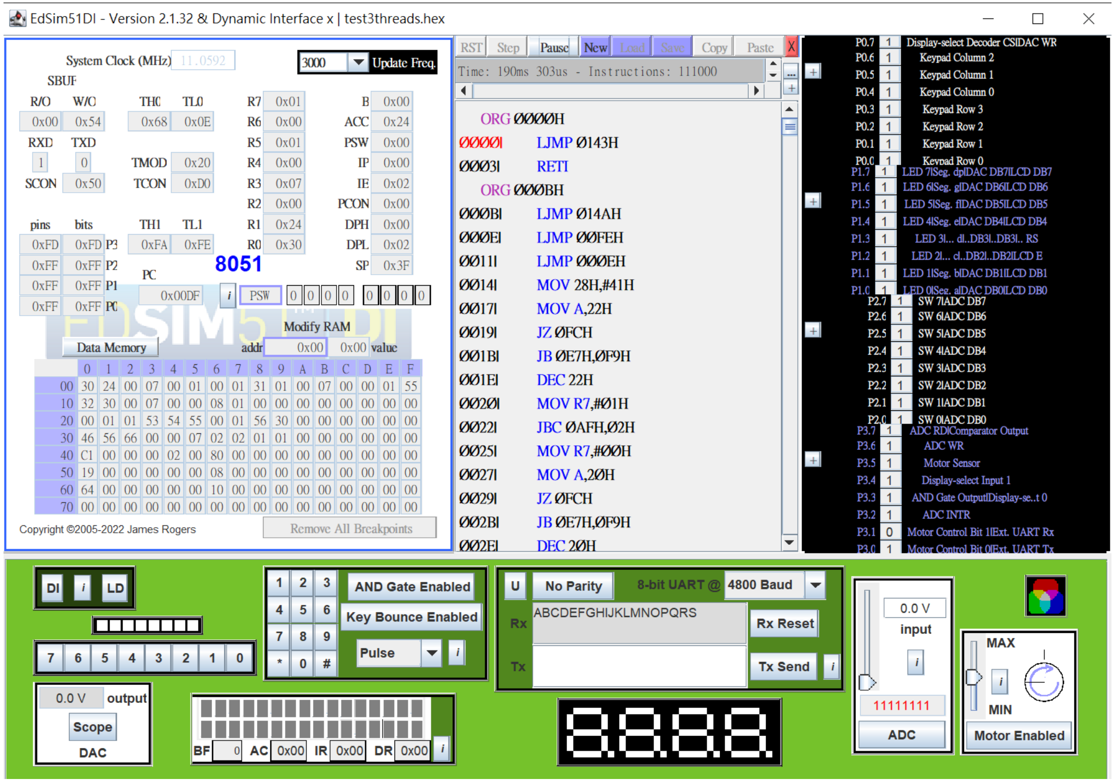
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	32	30	00	01	03	00	03	02	37	01	00	00	07	00	00	04
10	32	30	00	00	07	00	00	08	00	00	00	00	00	00	00	00
20	01	00	03	00	00	00	00	00	48	31	00	00	00	00	00	00
30	46	56	66	00	00	07	02	02	41	10	00	00	00	00	51	52
40	18	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	57	00	00	00	00	00	08	00	00	00	00	00	00	00	00	00
60	14	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2022 James Rogers

```
Time: 559us - Instructions: 368

ØØ92   INC A
ØØ93   MOV 29H,A
ØØ95   LCALL Ø1D7H
ØØ98   SJMP ØCØH
ØØ9A*  MOV R6,89H
ØØ9C   MOV R7,#ØØH
ØØ9E   ORL Ø6H,#2ØH
ØØA1   MOV 89H,R6
ØØA3   MOV 8DH,#ØFAH
ØØA6   MOV 98H,#5ØH
ØØA9   SETB 8EH
ØØAB   MOV A,21H
ØØAD   JZ ØFCH
ØØAF   JB ØE7H,ØF9H
ØØB2   DEC 21H
ØØB4   CLR ØAFH
ØØB6   MOV A,2ØH
ØØB8   JZ ØFCH
ØØBA   JB ØE7H,ØF9H
```

4. Before: Unfair version

If Producer1 is spawned before Producer2, then UART would only output characters(as shown below)



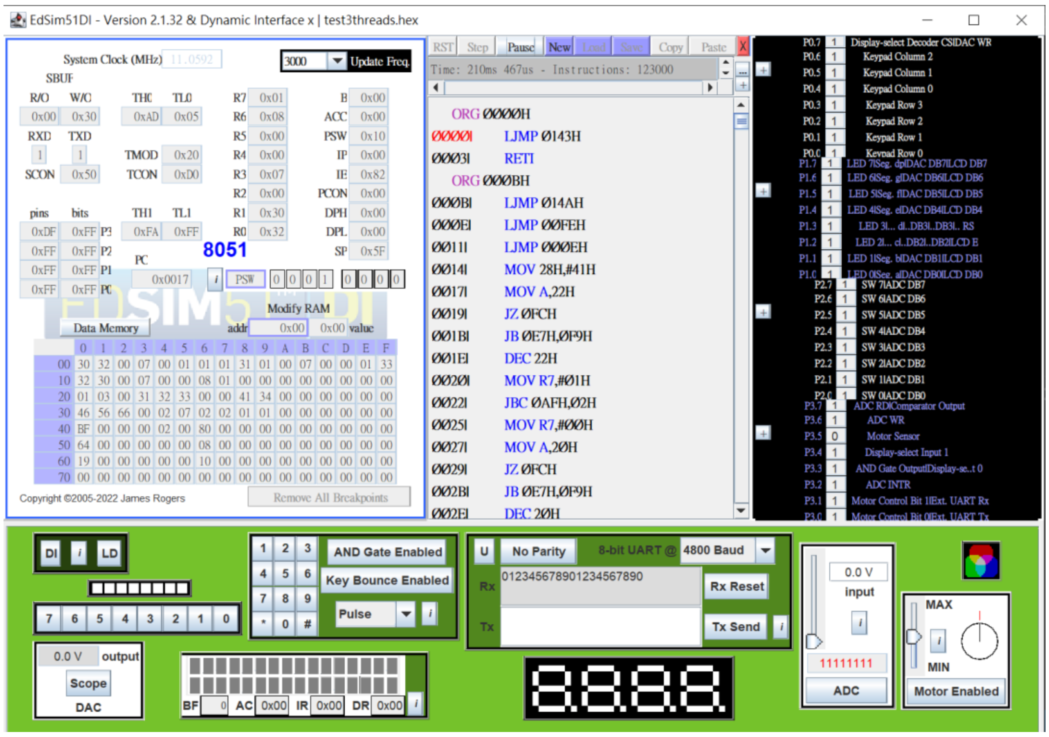
EdSim51DI - Version 2.1.32 & Dynamic Interface x | test3threads.hex

```
Time: 190ms 303us - Instructions: 111000

    ORG ØØØØH
ØØØØ|       LJMP Ø143H
ØØØ3|       RETI
    ORG ØØØBH
ØØØB|       LJMP Ø14AH
ØØØE|       LJMP ØØFEH
ØØ11|       LJMP ØØØEH
ØØ14|       MOV 28H,#41H
ØØ17|       MOV A,22H
ØØ19|       JZ ØFCH
ØØ1B|       JB ØE7H,ØF9H
ØØ1E|       DEC 22H
ØØ2Ø|       MOV R7,#Ø1H
ØØ22|       JBC ØAFH,Ø2H
ØØ25|       MOV R7,#ØØH
ØØ27|       MOV A,2ØH
ØØ29|       JZ ØFCH
ØØ2B|       JB ØE7H,ØF9H
ØØ2E|       DEC 2ØH
```

8-bit UART @ 4800 Baud, No Parity

Rx: ABCDEFGHIJKLMNOPQRS

If Producer2 is spawned before Producer1, then UART would only output numbers(as shown below)



EdSim51DI - Version 2.1.32 & Dynamic Interface x | test3threads.hex

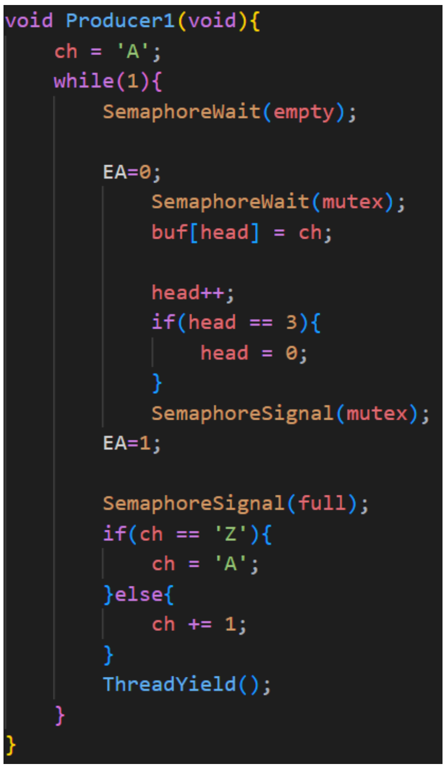
```
Time: 210ms 467us - Instructions: 123000

    ORG ØØØØH
ØØØØ|       LJMP Ø143H
ØØØ3|       RETI
    ORG ØØØBH
ØØØB|       LJMP Ø14AH
ØØØE|       LJMP ØØFEH
ØØ11|       LJMP ØØØEH
ØØ14|       MOV 28H,#41H
ØØ17|       MOV A,22H
ØØ19|       JZ ØFCH
ØØ1B|       JB ØE7H,ØF9H
ØØ1E|       DEC 22H
ØØ2Ø|       MOV R7,#Ø1H
ØØ22|       JBC ØAFH,Ø2H
ØØ25|       MOV R7,#ØØH
ØØ27|       MOV A,2ØH
ØØ29|       JZ ØFCH
ØØ2B|       JB ØE7H,ØF9H
ØØ2E|       DEC 2ØH
```

8-bit UART @ 4800 Baud, No Parity

Rx: 01234567890123456790

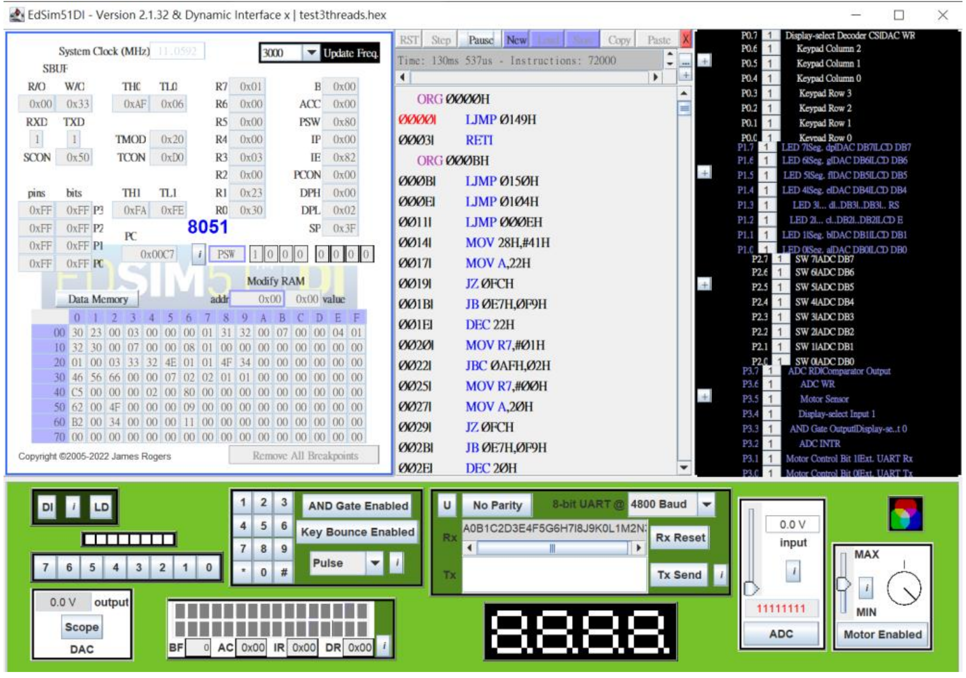
5. After: fair version & Fairness

   I add an explicit ThreadYield() at the end of Producer1 and 2 to force real round-robin, so that after each producer went through 1 iteration, it will switch to the other thread by calling ThreadYield() explicitly. The screenshot below take Producer1 as example(Producer2 the same).



```
void Producer1(void){
    ch = 'A';
    while(1){
        SemaphoreWait(empty);

        EA=0;
            SemaphoreWait(mutex);
            buf[head] = ch;

            head++;
            if(head == 3){
                head = 0;
            }
            SemaphoreSignal(mutex);
        EA=1;

        SemaphoreSignal(full);
        if(ch == 'Z'){
            ch = 'A';
        }else{
            ch += 1;
        }
        ThreadYield();
    }
}
```

   If Producer1 is spawned first in main, then the first character of UART output will be alphabet a, then number 0, b, 1,…etc(as shown below)



EdSim51DI - Version 2.1.32 & Dynamic Interface x | test3threads.hex

System Clock (MHz) 11.0592 — Update Freq. 3000

SBUF R/O 0x00, W/O 0x33; RXD 1, TXD 1; SCON 0x50

TH0 0xAF, TL0 0x06; TMOD 0x20; TCON 0xD0; TH1 0xFA, TL1 0xFE

R7 0x01, R6 0x00, R5 0x00, R4 0x00, R3 0x03, R2 0x00, R1 0x23, R0 0x30

B 0x00, ACC 0x00, PSW 0x80, IP 0x00, IE 0x82, PCON 0x00, DPH 0x00, DPL 0x02, SP 0x3F

P3 pins 0xFF bits 0xFF; P2 0xFF 0xFF; P1 0xFF 0xFF; P0 0xFF 0xFF

PC 0x00C7; PSW 1 0 0 0 0 0 0 0

Data Memory — Modify RAM addr 0x00 value 0x00

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	30	23	00	03	00	00	00	01	31	32	00	07	00	00	04	01
10	32	30	00	07	00	00	08	01	00	00	00	00	00	00	00	00
20	01	00	03	33	32	4E	01	01	4F	34	00	00	00	00	00	00
30	46	56	66	00	00	07	02	02	01	01	00	00	00	00	00	00
40	C5	00	00	00	02	00	80	00	00	00	00	00	00	00	00	00
50	62	00	4F	00	00	00	09	00	00	00	00	00	00	00	00	00
60	B2	00	34	00	00	00	11	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

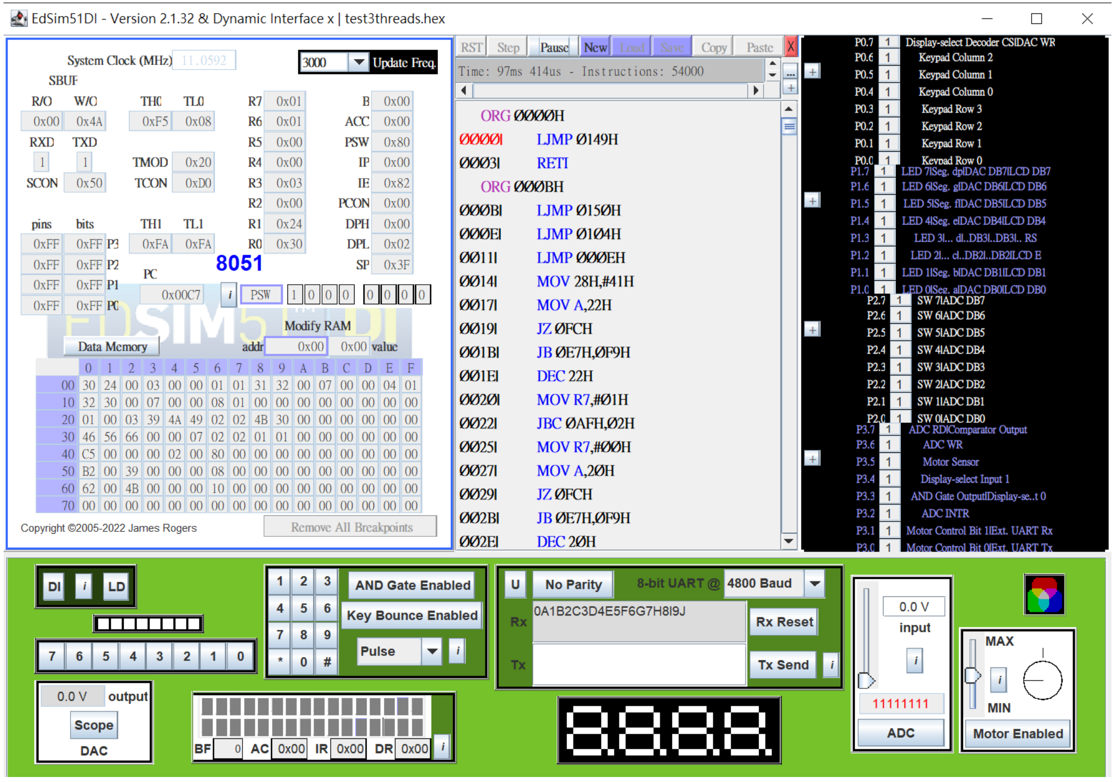
Copyright ©2005-2022 James Rogers

```
Time: 130ms 537us - Instructions: 72000

      ORG 0000H
0000|     LJMP 0149H
0003|     RETI
      ORG 000BH
000B|     LJMP 0150H
000E|     LJMP 0104H
0011|     LJMP 000EH
0014|     MOV 28H,#41H
0017|     MOV A,22H
0019|     JZ 0FCH
001B|     JB 0E7H,0F9H
001E|     DEC 22H
0020|     MOV R7,#01H
0022|     JBC 0AFH,02H
0025|     MOV R7,#00H
0027|     MOV A,20H
0029|     JZ 0FCH
002B|     JB 0E7H,0F9H
002E|     DEC 20H
```

No Parity — 8-bit UART @ 4800 Baud

```
Rx: A0B1C2D3E4F5G6H7I8J9K0L1M2N3
```

If Producer2 is spawned first in main, then the first character of UART output will be number 0, then alphabet a, 1, b,…etc(as shown below)


EdSim51DI - Version 2.1.32 & Dynamic Interface x | test3threads.hex

System Clock (MHz) 11.0592 | 3000 Update Freq.

R/O	W/O	TH0	TL0	R7	0x01	B	0x00
0x00	0x4A	0xF5	0x08	R6	0x01	ACC	0x00
RXD	TXD			R5	0x00	PSW	0x80
1	1	TMOD	0x20	R4	0x00	IP	0x00
SCON	0x50	TCON	0xD0	R3	0x03	IE	0x82
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x24	DPH	0x00
0xFF	0xFF P3	0xFA	0xFA	R0	0x30	DPL	0x02
0xFF	0xFF P2					SP	0x3F
0xFF	0xFF P1	PC	0x00C7				
0xFF	0xFF P0						

PSW 1 0 0 0 0 0 0 0

Data Memory — Modify RAM addr 0x00 0x00 value

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	30	24	00	03	00	00	01	01	31	32	00	07	00	00	04	01
10	32	30	00	07	00	00	08	01	00	00	00	00	00	00	00	00
20	01	00	03	39	4A	49	02	02	4B	30	00	00	00	00	00	00
30	46	56	66	00	00	07	02	02	01	01	00	00	00	00	00	00
40	C5	00	00	00	02	00	80	00	00	00	00	00	00	00	00	00
50	B2	00	39	00	00	00	08	00	00	00	00	00	00	00	00	00
60	62	00	4B	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

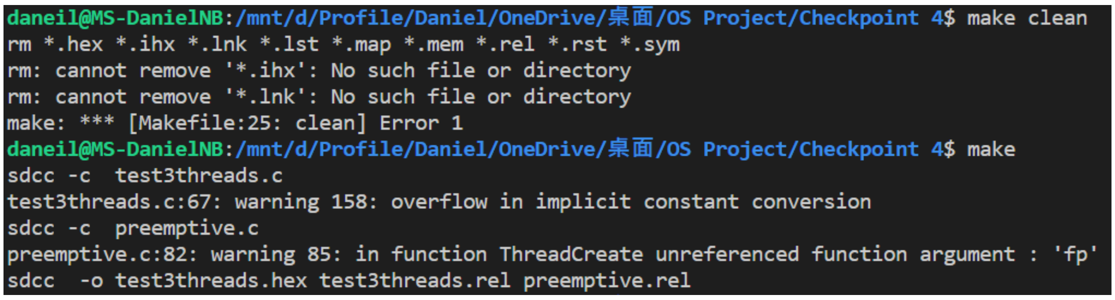
Copyright ©2005-2022 James Rogers

```
Time: 97ms 414us - Instructions: 54000

      ORG 0000H
0000|     LJMP 0149H
0003|     RETI
      ORG 000BH
000B|     LJMP 0150H
000E|     LJMP 0104H
0011|     LJMP 000EH
0014|     MOV 28H,#41H
0017|     MOV A,22H
0019|     JZ 0FCH
001B|     JB 0E7H,0F9H
001E|     DEC 22H
0020|     MOV R7,#01H
0022|     JBC 0AFH,02H
0025|     MOV R7,#00H
0027|     MOV A,20H
0029|     JZ 0FCH
002B|     JB 0E7H,0F9H
002E|     DEC 20H
```

No Parity — 8-bit UART @ 4800 Baud

Rx: 0A1B2C3D4E5F6G7H8I9J

6. Typescript and screenshots


```
daneil@MS-DanielNB:/mnt/d/Profile/Daniel/OneDrive/桌面/OS Project/Checkpoint 4$ make clean
rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym
rm: cannot remove '*.ihx': No such file or directory
rm: cannot remove '*.lnk': No such file or directory
make: *** [Makefile:25: clean] Error 1
daneil@MS-DanielNB:/mnt/d/Profile/Daniel/OneDrive/桌面/OS Project/Checkpoint 4$ make
sdcc -c  test3threads.c
test3threads.c:67: warning 158: overflow in implicit constant conversion
sdcc -c  preemptive.c
preemptive.c:82: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o test3threads.hex test3threads.rel preemptive.rel
```