

Report of Project Checkpoint 4

112062630 JING-YONG, SU 蘇敬詠

1. Screenshots for compilation

```
misubrian@misubrian-Katana-15-B13VFK:~/OS/project/CP4/misu$ make clean
rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
rm: 無法刪除 '*.ihx': 沒有此一檔案或目錄
rm: 無法刪除 '*.lnk': 沒有此一檔案或目錄
make: *** [Makefile:25: clean] 錯誤 1
misubrian@misubrian-Katana-15-B13VFK:~/OS/project/CP4/misu$ make
sdcc -c test3threads.c
sdcc -c preemptive.c
preemptive.c:208: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc -o test3threads.hex test3threads.rel preemptive.rel
misubrian@misubrian-Katana-15-B13VFK:~/OS/project/CP4/misu$
```

2.

Addresses:

Producer1: 0x001D
Producer2: 0x0092
Consumer: 0x0107
myTimer0Handler: 0x02A9
Semaphore full: 0x3C
Semaphore empty: 0x3D
Semaphore mutex: 0x3E
turn: 0x77 (1 for turn of producer1, 2 for producer2)

(a)

(i) Producer1 starts running. Initially, full = 0, empty = 3, mutex = 1.

(iii) The mutex has been unlocked. After signal(full), full = 1.

System Clock (MHz): 11.0592 | 100 | Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0xFE
0x00	0x00	0x07	0x03	R6	0x01	ACC	0x01
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0xC9
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
				R1	0x00	DPH	0x00
				R0	0x01	DPL	0x01
						SP	0x4F

pins bits

0xFF	0xFF	P3	0xFA	0xFA
0xFF	0xFF	P2		
0xFF	0xFF	P1		
0xFF	0xFF	P0		

PC: 8051 | PSW: 11001001

Data Memory

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	38	03	00	00	04	02	02	01	00	00	00	00	01	00	
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
20	09	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
30	07	01	00	02	41	00	00	46	56	66	00	41	01	02	01	
40	1B	01	00	00	02	00	80	37	37	03	00	00	04	02	02	
50	56	00	BD	03	00	00	08	00	00	00	00	00	00	00	00	
60	92	00	00	00	00	00	10	00	00	00	00	00	00	00	00	
70	00	42	10	00	00	00	01	01	00	00	00	00	00	00	00	

Remove All Breakpoints

Executed 0x0062: INC 3CH | Time: 9ms 145us

```

0056| MOV R6, 82H
0058| MOV R7, 83H
005A| MOV 76H, R6
005C| MOV C, 00H
005E| MOV 0AFH, C
0060| INC 3EH
0062| INC 3CH
0064| MOV A, #5AH
0066| CJNE A, 3BH, 03H
0069| SETB C
006A| SJMP 01H
006C| CLR C
006D| MOV 00H, C
006F| JC 0BH
0071| MOV R7, 3BH
0073| INC R7
0074| MOV A, R7
0075| MOV R6, A
0076| RLC A
0077| SUBB A, 0E0H
0079| MOV R7, A
007A| SJMP 04H
007C| MOV R6, #41H
007E| MOV R7, #00H
0080| MOV 3BH, R6
  
```

(iv) Producer1 is running now, but it get stuck because it's the turn of producer2 now. (turn = 2)

System Clock (MHz): 11.0592 | 100 | Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0xFE
0x00	0x00	0x81	0x1E	R6	0x42	ACC	0x01
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0x89
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
				R1	0x00	DPH	0x00
				R0	0x01	DPL	0x01
						SP	0x4F

pins bits

0xFF	0xFF	P3	0xFA	0xFB
0xFF	0xFF	P2		
0xFF	0xFF	P1		
0xFF	0xFF	P0		

PC: 8051 | PSW: 10001001

Data Memory

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	38	03	00	00	04	02	02	01	00	00	00	00	00	42	
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
20	09	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
30	07	01	00	02	41	00	00	46	56	66	00	42	01	02	01	
40	1B	01	00	00	02	00	80	37	37	03	00	00	04	02	02	
50	56	00	BD	03	00	00	08	00	00	00	00	00	00	00	00	
60	92	00	00	00	00	00	10	00	00	00	00	00	00	00	00	
70	00	42	10	00	00	00	02	00	00	00	00	00	00	00	00	

Remove All Breakpoints

Time: 13ms 410us - Instructions: 7738

```

001D* MOV 3BH, #41H
0020| MOV A, #01H
0022| CJNE A, 77H, 0FBH
0025| MOV A, 3DH
0027| JB 0E7H, 0FBH
002A| JZ 0F9H
002C| DEC 3DH
002E| MOV A, 3EH
0030| JB 0E7H, 0FBH
0033| JZ 0F9H
0035| DEC 3EH
0037| SETB 00H
0039| JBC 0AFH, 02H
003C| CLR 00H
003E| MOV A, 76H
0040| ADD A, #34H
0042| MOV R0, A
0043| MOV @R0, 3BH
0045| MOV R6, 76H
0047| MOV R7, #00H
0049| MOV 82H, R6
004B| MOV 83H, R7
004D| INC DPTR
004E| MOV 08H, #03H
0051| MOV 00H, R7
  
```

2. (b)

(I) Producer2 starts running. Initially, full = 1, empty = 2, mutex = 1.

The screenshot displays the Proteus ISIS simulation environment for an 8051 microcontroller. The top status bar shows the system clock at 11.0592 MHz and the update frequency at 100. The left panel shows the internal registers and pins. The right panel shows the assembly code being executed.

Internal State:

- Registers:** R0-R7, ACC, PSW, IP, IE, PCON, DPH, DPL, SP.
- Timers:** TH0, TL0, TH1, TL1.
- Serial Port:** SBUF, RXD, TXD, SCON.
- Pins:** P0, P1, P2, P3.
- PC (Program Counter):** 0x0092.
- PSW (Program Status Word):** 00010000 (highlighted with a red box).
- SP (Stack Pointer):** 0x5F.

Assembly Code:

```

0092* MOV 3FH, #30H
0095| MOV A, #02H
0097| CJNE A, 77H, 0FBH
009A| MOV A, 3DH
009C| JB 0E7H, 0FBH
009F| JZ 0F9H
00A1| DEC 3DH
00A3| MOV A, 3EH
00A5| JB 0E7H, 0FBH
00A8| JZ 0F9H
00AA| DEC 3EH
00AC| SETB 01H
00AE| JBC 0AFH, 02H
00B1| CLR 01H
00B3| MOV A, 76H
00B5| ADD A, #34H
00B7| MOV R0, A
00B8| MOV @R0, 3FH
00BA| MOV R6, 76H
00BC| MOV R7, #00H
00BE| MOV 82H, R6
00C0| MOV 83H, R7
00C2| INC DPTR
00C3| MOV 08H, #03H
00C6| MOV 00H, R7
  
```

Data Memory:

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	38	03	00	00	04	02	02	38	39	00	00	00	00	42	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	09	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	07	02	00	02	41	00	00	46	56	66	00	42	01	02	01	00
40	1B	01	00	00	02	00	80	37	37	03	00	00	04	02	02	00
50	20	00	01	FE	01	00	89	38	00	00	00	00	00	42	00	00
60	92	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	42	10	00	00	00	01	02	00	00	00	00	00	00	00	00

(ii) Producer2 is running. It's in its critical section because mutex = 0.

It has gotten out of "wait(empty)", therefore, "empty" has been minus one. Now full = 1, and empty = 1.

System Clock (MHz) 11.0592 100 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0x00
0x00	0x00	0x04	0x0B	R6	0x00	ACC	0x01
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0x11
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFF	0xFF	P3	0xFA 0xFE	R0	0x00	DPL	0x00
0xFF	0xFF	P2				SP	0x5F
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 8051 0x00AC i PSW 0 0 0 1 0 0 0 1

Modify RAM

Data Memory	addr	0x00	0x00	value
	0	1	2	3
00	37	38	03	00
10	00	00	00	00
20	09	00	00	00
30	07	02	00	02
40	1B	01	00	02
50	20	00	01	FE
60	92	00	00	00
70	00	42	10	00

Copyright ©2005-2024 James Rogers Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Executed 0x00AA: DEC 3EH | Time: 17ms 938us

```

0092* MOV 3FH, #30H
0095| MOV A, #02H
0097| CJNE A, 77H, 0FBH
009A| MOV A, 3DH
009C| JB 0E7H, 0FBH
009F| JZ 0F9H
00A1| DEC 3DH
00A3| MOV A, 3EH
00A5| JB 0E7H, 0FBH
00A8| JZ 0F9H
00AA| DEC 3EH
00AC| SETB 01H
00AE| JBC 0AFH, 02H
00B1| CLR 01H
00B3| MOV A, 76H
00B5| ADD A, #34H
00B7| MOV R0, A
00B8| MOV @R0, 3FH
00BA| MOV R6, 76H
00BC| MOV R7, #00H
00BE| MOV 82H, R6
00C0| MOV 83H, R7
00C2| INC DPTR
00C3| MOV 08H, #03H
00C6| MOV 00H, 07

```

(iii) The mutex has been unlocked.

System Clock (MHz) 11.0592 100 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0xFF
0x00	0x00	0x07	0x05	R6	0x02	ACC	0x01
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0xD1
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFF	0xFF	P3	0xFA 0xFE	R0	0x35	DPL	0x02
0xFF	0xFF	P2				SP	0x5F
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 8051 0x00D7 i PSW 1 1 0 1 0 0 0 1

Modify RAM

Data Memory	addr	0x00	0x00	value
	0	1	2	3
00	37	38	03	00
10	35	00	00	00
20	0B	00	00	00
30	07	02	00	02
40	1B	01	00	02
50	20	00	01	FE
60	CB	00	BD	03
70	00	42	10	00

Copyright ©2005-2024 James Rogers Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Executed 0x00D5: INC 3EH | Time: 18ms 36us

```

00CB| MOV R6, 82H
00CD| MOV R7, 83H
00CF| MOV 76H, R6
00D1| MOV C, 01H
00D3| MOV 0AFH, C
00D5| INC 3EH
00D7| INC 3CH
00D9| MOV A, #39H
00DB| CJNE A, 3FH, 03H
00DE| SETB C
00DF| SJMP 01H
00E1| CLR C
00E2| MOV 01H, C
00E4| JC 0BH
00E6| MOV R7, 3FH
00E8| INC R7
00E9| MOV A, R7
00EA| MOV R6, A
00EB| RLC A
00EC| SUBB A, 0E0H
00EE| MOV R7, A
00EF| SJMP 04H
00F1| MOV R6, #30H
00F3| MOV R7, #00H
00F5| MOV 3EH, 06

```

(iv) After exiting critical section i.e. signal(full), full = 2.

System Clock (MHz) 11.0592 100 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0xFF
0x00	0x00	0x07	0x06	R6	0x02	ACC	0x01
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0xD1
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFF	0xFF	P3	0xFA	0xFF		DPL	0x02
0xFF	0xFF	P2				SP	0x5F
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 8051

Modify RAM

Data Memory

addr	0x00	0x00	value
0	0	1	2
1	3	4	5
2	6	7	8
3	9	A	B
4	C	D	E
5	F		
6	00	37	38
7	03	00	00
8	04	02	02
9	01	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00
10	35	00	00
11	00	00	00
12	00	00	00
13	02	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	0B	00	00
19	00	00	00
20	00	00	00
21	00	00	00
22	00	00	00
23	00	00	00
24	00	00	00
25	00	00	00
26	00	00	00
27	00	00	00
28	00	00	00
29	00	00	00
30	07	02	00
31	02	41	30
32	00	46	56
33	66	00	42
34	02	01	01
35	30		
36	1B	01	00
37	00	02	00
38	80	37	37
39	03	00	00
40	00	04	02
41	02	00	00
42	00	00	00
43	00	00	00
44	00	00	00
45	00	00	00
46	00	00	00
47	00	00	00
48	00	00	00
49	00	00	00
50	20	00	01
51	FE	01	00
52	89	38	00
53	00	00	00
54	00	00	00
55	00	00	00
56	00	00	00
57	00	00	00
58	00	00	00
59	00	00	00
60	CB	00	BD
61	03	00	00
62	10	00	00
63	00	00	00
64	00	00	00
65	00	00	00
66	00	00	00
67	00	00	00
68	00	00	00
69	00	00	00
70	00	42	10
71	00	00	00
72	02	02	00
73	00	00	00
74	00	00	00
75	00	00	00
76	00	00	00
77	00	00	00
78	00	00	00
79	00	00	00

Copyright ©2005-2024 James Rogers

Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Executed 0x00D7: INC 3CH | Time: 18ms 37us

```

00CB| MOV R6,82H
00CD| MOV R7,83H
00CF| MOV 76H,R6
00D1| MOV C,01H
00D3| MOV 0AFH,C
00D5| INC 3EH
00D7| INC 3CH
00D9| MOV A,#39H
00DB| CJNE A,3FH,03H
00DE| SETB C
00DF| SJMP 01H
00E1| CLR C
00E2| MOV 01H,C
00E4| JC 0BH
00E6| MOV R7,3FH
00E8| INC R7
00E9| MOV A,R7
00EA| MOV R6,A
00EB| RLC A
00EC| SUBB A,0E0H
00EE| MOV R7,A
00EF| SJMP 04H
00F1| MOV R6,#30H
00F3| MOV R7,#00H
00F5| MOV 3EH,R6
  
```

(v) Producer2 is running now, but it get stuck because it's the turn of producer1 now. (turn = 1)

System Clock (MHz) 11.0592 100 Update Freq.

SBUF

R/O	W/O	TH0	TL0	R7	0x00	B	0xFF
0x00	0x00	0x56	0x1C	R6	0x31	ACC	0x02
RXD	TXD	TH0D	0x20	R5	0x00	PSW	0x11
1	1	TCON	0xD0	R4	0x00	IP	0x00
SCON	0x50			R3	0x00	IE	0x82
				R2	0x00	PCON	0x00
pins	bits	TH1	TL1	R1	0x00	DPH	0x00
0xFF	0xFF	P3	0xFA	0xFF		DPL	0x02
0xFF	0xFF	P2				SP	0x5F
0xFF	0xFF	P1					
0xFF	0xFF	P0					

PC 8051

Modify RAM

Data Memory

addr	0x00	0x00	value
0	0	1	2
1	3	4	5
2	6	7	8
3	9	A	B
4	C	D	E
5	F		
6	00	37	38
7	03	00	00
8	04	02	02
9	01	00	00
A	00	00	00
B	00	00	00
C	00	00	00
D	00	00	00
E	00	00	00
F	00	00	00
10	35	00	00
11	00	00	00
12	00	00	00
13	00	00	00
14	00	00	00
15	00	00	00
16	00	00	00
17	00	00	00
18	0B	00	00
19	00	00	00
20	00	00	00
21	00	00	00
22	00	00	00
23	00	00	00
24	00	00	00
25	00	00	00
26	00	00	00
27	00	00	00
28	00	00	00
29	00	00	00
30	07	02	00
31	02	41	30
32	00	46	56
33	66	00	42
34	02	01	01
35	30		
36	1B	01	00
37	00	02	00
38	80	37	37
39	03	00	00
40	00	04	02
41	02	00	00
42	00	00	00
43	00	00	00
44	00	00	00
45	00	00	00
46	00	00	00
47	00	00	00
48	00	00	00
49	00	00	00
50	20	00	01
51	FE	01	00
52	89	38	00
53	00	00	00
54	00	00	00
55	00	00	00
56	00	00	00
57	00	00	00
58	00	00	00
59	00	00	00
60	CB	00	BD
61	03	00	00
62	10	00	00
63	00	00	00
64	00	00	00
65	00	00	00
66	00	00	00
67	00	00	00
68	00	00	00
69	00	00	00
70	00	42	10
71	00	00	00
72	02	02	00
73	00	00	00
74	00	00	00
75	00	00	00
76	00	00	00
77	00	00	00
78	00	00	00
79	00	00	00

Copyright ©2005-2024 James Rogers

Remove All Breakpoints

RST Step Run New Load Save CPY Paste BP

Time: 20ms 804us - Instructions: 12286

```

0095| MOV A,#02H
0097| CJNE A,77H,0FBH
009A| MOV A,3DH
009C| JB 0E7H,0FBH
009F| JZ 0F9H
00A1| DEC 3DH
00A3| MOV A,3EH
00A5| JB 0E7H,0FBH
00A8| JZ 0F9H
00AA| DEC 3EH
00AC| SETB 01H
00AE| JBC 0AFH,02H
00B1| CLR 01H
00B3| MOV A,76H
00B5| ADD A,#34H
00B7| MOV R0,A
00B8| MOV @R0,3FH
00BA| MOV R6,76H
00BC| MOV R7,#00H
00BE| MOV 82H,R6
00C0| MOV 83H,R7
00C2| INC DPTR
00C3| MOV 08H,#03H
00C6| MOV 09H,R7
00C8| LCALL 0306H
  
```

2. (c)

(I) Consumer starts running. Initially, full = 0, empty = 3, and mutex = 1.

System Clock (MHz) 11.0592 | 100 | Update Freq.

8051

PSW 1 0 0 0 0 0 0 1

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	39	37	03	00	00	04	02	02	99	01	80	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	07	00	00	02	00	00	00	46	56	66	00	00	00	03	01	00
40	7D	01	81	00	00	00	00	00	00	00	00	00	00	00	00	00
50	1D	00	00	00	00	00	08	00	00	00	00	00	00	00	00	00
60	92	00	00	00	00	00	10	00	00	00	00	00	00	00	00	00
70	00	42	10	00	00	00	00	01	00	00	00	00	00	00	00	00

Remove All Breakpoints

Executed 0x0107: CLR 0AFH | Time: 494us - I

```

0107* CLR 0AFH
0109| ORL 89H, #20H
010C| MOV 8DH, #0FAH
010F| MOV 98H, #50H
0112| SETB 8EH
0114| SETB 0AFH
0116| MOV A, 3CH
0118| JB 0E7H, 0FBH
011B| JZ 0F9H
011D| DEC 3CH
011F| MOV A, 3EH
0121| JB 0E7H, 0FBH
0124| JZ 0F9H
0126| DEC 3EH
0128| SETB 02H
012A| JBC 0AFH, 02H
012D| CLR 02H
012F| MOV A, 75H
0131| ADD A, #34H
0133| MOV R1, A
0134| MOV 99H, @R1
0136| MOV R6, 75H
0138| MOV R7, #00H
013A| MOV 82H, R6
013C| MOV 83H, R7
  
```

(ii) Consumer is in its critical section because mutex = 0.

It has gotten out of “wait(full)”, therefore, full has been minus one. Now full = 1, and empty = 1.

System Clock (MHz) 11.0592 | 1 | Update Freq.

8051

PSW 1 1 0 0 0 0 0 1

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	37	38	00	00	00	04	01	00	01	00	00	00	00	00	44	00
10	39	37	00	00	00	00	33	00	00	00	00	00	00	00	00	00
20	0F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	07	00	00	02	31	43	32	46	56	66	00	4	01	01	00	33
40	16	01	00	FE	01	00	C0	37	34	00	00	0	04	01	00	00
50	22	00	01	FF	02	00	89	38	00	00	00	00	00	44	00	00
60	95	00	02	00	00	00	11	39	37	00	00	00	00	33	00	00
70	00	42	10	00	00	01	00	01	00	00	00	00	00	00	00	00

Remove All Breakpoints

Time: 80ms 182us - Instructions: 47081

```

0128| SETB 02H
012A| JBC 0AFH, 02H
012D| CLR 02H
012F| MOV A, 75H
0131| ADD A, #34H
0133| MOV R1, A
0134| MOV 99H, @R1
0136| MOV R6, 75H
0138| MOV R7, #00H
013A| MOV 82H, R6
013C| MOV 83H, R7
013E| INC DPTR
013F| MOV 08H, #03H
0142| MOV 09H, R7
0144| LCALL 0396H
0147| MOV R6, 82H
0149| MOV R7, 83H
014B| MOV 75H, R6
014D| MOV C, 02H
014F| MOV 0AFH, C
0151| INC 3EH
0153| INC 3DH
0155| JBC 99H, 0BEH
0158| SJMP 0FBH
015A| MOV 77H, #01H
  
```


(iii) After exiting critical section, empty becomes 2 and mutex becomes 1.

System Clock (MHz): 11.0592

Update Freq. 1

Executed 0x0153: INC 3DH | Time: 80ms 281us

0147| MOV R6, 82H
 0149| MOV R7, 83H
 014B| MOV 75H, R6
 014D| MOV C, 02H
 014F| MOV 0AFH, C
 0151| INC 3EH
 0153| INC 3DH
 0155| JBC 99H, 0BEH
 0158| SJMP 0FBH
 015A| MOV 77H, #01H
 015D| SETB 03H
 015F| JBC 0AFH, 02H
 0162| CLR 03H
 0164| MOV 3DH, #03H
 0167| MOV 3CH, #00H
 016A| MOV 3EH, #01H
 016D| MOV C, 03H
 016F| MOV 0AFH, C
 0171| MOV DPTR, #001DH
 0174| LCALL 01AEH
 0177| MOV DPTR, #0092H
 017A| LCALL 01AEH
 017D| LJMP 0107H
 0180| LJMP 018BH
 0183| RET

(iv) After a while, consumer is running but gets stuck, because the buffer is empty now. (empty = 3)

System Clock (MHz): 11.0592

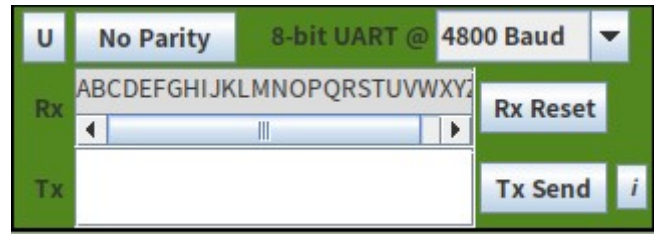
Update Freq. 100

Executed 0x0155: JBC 99H, 0BEH | Time: 82ms

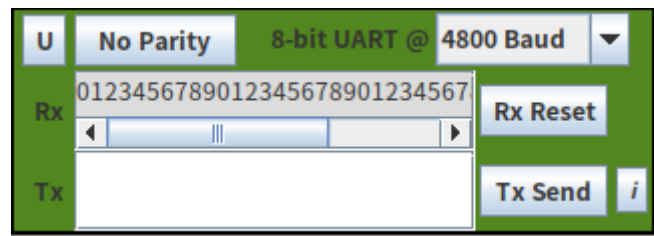
0147| MOV R6, 82H
 0149| MOV R7, 83H
 014B| MOV 75H, R6
 014D| MOV C, 02H
 014F| MOV 0AFH, C
 0151| INC 3EH
 0153| INC 3DH
 0155| JBC 99H, 0BEH
 0158| SJMP 0FBH
 015A| MOV 77H, #01H
 015D| SETB 03H
 015F| JBC 0AFH, 02H
 0162| CLR 03H
 0164| MOV 3DH, #03H
 0167| MOV 3CH, #00H
 016A| MOV 3EH, #01H
 016D| MOV C, 03H
 016F| MOV 0AFH, C
 0171| MOV DPTR, #001DH
 0174| LCALL 01AEH
 0177| MOV DPTR, #0092H
 017A| LCALL 01AEH
 017D| LJMP 0107H
 0180| LJMP 018BH
 0183| RET

3. (a) Unfair version:

If we create producer1 earlier than producer2, then the buffer always be filled with items (A~Z) from producer1. The reason is that the period of timer is long enough for producer1 to full the buffer. There's no chance for producer2 to get out from wait(empty).



If we create producer2 first, then we get all output with 0~9.



3. (b) Fair version:

Use a shared variable called "turn" to control. Initially, turn is set to 1.

If turn = 1, producer1 can execute the producing code, otherwise, producer2 can execute the producing code. Whenever producer1 finishes producing, it gives the chance for producer2 by setting turn = 2, and vice versa.

```

void Producer1(void)
{
    /*
     * [TODO]
     * initialize producer data structure, and then enter
     * an infinite loop (does not return)
     */
    item1 = 'A';
    while (1)
    {
        while(turn != 1){}
        /* [TODO]
         * wait for the buffer to be available,
         * and then write the new data into the buffer */
        SemaphoreWait(empty);
        SemaphoreWait(mutex);
        critical{
            buff[tail] = item1;
            tail = (tail + 1) % 3;
        }
        SemaphoreSignal(mutex);
        SemaphoreSignal(full);
        item1 = (item1 != 'Z')? (item1 + 1) : 'A';

        critical{
            turn = 2;
        }

        //printf("producing %c\n", item);
    }
}

```

The UART will look like this:



Clearly, it's completely fair because producer 1 and 2 take turns to execute.