

Name: 吳嘉濬 Student ID: 109021115

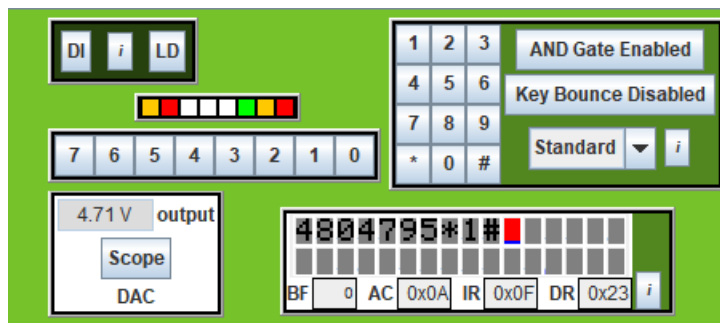
```
User@MSI MINGW64 ~/Desktop/OS/checkpoint5
$ ls
buttonlib.c buttonlib.h keylib.c keylib.h lcdlib.c lcdlib.h Makefile preemptive.c preemptive.h testlcd.c

User@MSI MINGW64 ~/Desktop/OS/checkpoint5
$ make
sdcc -c --model-small testlcd.c
sdcc -c --model-small preemptive.c
preemptive.c:176: warning 59: function 'now' must return value
preemptive.c:278: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc -c --model-small lcdlib.c
lcdlib.c:81: warning 85: in function delay unreferenced function argument : 'n'
sdcc -c --model-small buttonlib.c
sdcc -c --model-small keylib.c
sdcc -o testlcd.hex testlcd.rel preemptive.rel lcdlib.rel buttonlib.rel keylib.rel
Multiple definition of _SemaphoreCreate_PARM_2
Multiple definition of _SemaphoreCreate
make: *** [Makefile:12: testlcd.hex] Error 1

User@MSI MINGW64 ~/Desktop/OS/checkpoint5
$ ls
buttonlib.asm buttonlib.rst keylib.lst lcdlib.c lcdlib.sym preemptive.lst testlcd.c testlcd.mem
buttonlib.c buttonlib.sym keylib.rel lcdlib.h Makefile preemptive.rel testlcd.hex testlcd.rel
buttonlib.h keylib.asm keylib.rst lcdlib.lst preemptive.asm preemptive.rst testlcd.lk testlcd.rst
buttonlib.lst keylib.c keylib.sym lcdlib.rel preemptive.c preemptive.sym testlcd.lst testlcd.sym
buttonlib.rel keylib.h lcdlib.asm lcdlib.rst preemptive.h testlcd.asm testlcd.map
```

I only finished the part1 of checkpoint5.

The following is the screenshot of LCD display.



The mechanism is similar to checkpoint4, Button is Producer1, Keypad is Producer2, and LCD is Consumer.

To maintain fair version as I did in checkpoint4, Button(Producer1) and Keypad(Producer2) have their own additional mutex ok1 and ok2, which ok1 is initialized to 1 and ok2 is initialized to 0. So after Button has produced a number, it can't produce the following one until Keypad increases ok1, so this ensures that two producers will produce numbers(or '*', '#') in turns one by one, without anyone having starvation.