# 实验3实验报告

## 实验流程图

见diagram.pdf

## 程序代码

### user\_program.asm

.ORIG x3000

LD R6,SSP;initialize the stack pointer

LD R4,CON

STI R4,ENTRY;set up the keyboard interrupt vector table entry

LDI R2,KBSR

NOT R2,R2

LD R3,NUM1

NOT R3,R3

AND R2,R2,R3

NOT R2,R2

STI R2,KBSR;enable keyboard interrupts

AND R2,R2,#0;whether it is star or well 0:STAR -1:WELL

LEA R3,STAR1

LEA R4,WELL1

PSTAR AND R5,R5,#0

LEA R0,STAR1

AND R2,R2,R2

BRn PWELL

JSR OFF

PUTS

JSR ON

JSR DELAY

LEA R0,STAR2

AND R2,R2,R2

BRn PWELL

JSR OFF

PUTS

JSR ON

JSR DELAY

BR PSTAR

PWELL AND R5,R5,#0

LEA R0,WELL1

AND R2,R2,R2

BRz PSTAR

JSR OFF

PUTS

JSR ON

JSR DELAY

LEA R0,WELL2

AND R2,R2,R2

BRz PSTAR

JSR OFF

PUTS

JSR ON

JSR DELAY

BACK BR PWELL

DELAY ST R1, SaveR1

LD R1,COUNT

REP ADD R1, R1, #-1

BRp REP

LD R1, SaveR1

RET

OFF ST R1, SaveR1

ST R3, SaveR3

LD R3,NUM2

LDI R1,KBSR

AND R1,R1,R3

STI R1,KBSR

LD R1, SaveR1

LD R3, SaveR3

RET

ON ST R1, SaveR1

ST R3, SaveR3

LDI R1,KBSR

NOT R1,R1

LD R3,NUM1

NOT R3,R3

AND R1,R1,R3

NOT R1,R1

STI R1,KBSR;enable keyboard interrupts

LD R1, SaveR1

LD R3, SaveR3

RET

COUNT .FILL x7FFF

SaveR1 .BLKW 1

SaveR2 .BLKW 2

SaveR3 .BLKW 3

SSP .FILL x3000

ENTRY .FILL x0180

CON .FILL x2000;starting address of the interrupt\_service\_routine

KBSR .FILL xFE00

NUM1 .FILL x4000;for changing KBSR OR

NUM2 .FILL xBFFF;for changing KBSR AND

STAR1 .STRINGZ "\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \n"

STAR2 .STRINGZ " \*\* \*\* \*\* \*\* \*\* \*\* \*\* \n"

WELL1 .STRINGZ "## ## ## ## ## ## ## ## \n"

WELL2 .STRINGZ " ## ## ## ## ## ## ## \n"

.END

### interrupt\_service\_routine.asm

.ORIG x2000

START ST R1,SaveR1

ST R3,SaveR3;The address of \*

ST R4,SaveR4;The address of #

ADD R5,R5,#1

INPUT LDI R1,KBSR

BRzp INPUT

LDI R3,KBDR

AND R4,R4,#0

ADD R4,R4,#10;R4<-counter

OUTPUT LDI R1,DSR

BRzp OUTPUT

STI R3,DDR

ADD R4,R4,#-1

BRp OUTPUT

LD R3,NEWLINE

STI R3,DDR

;change pattern

AND R2,R2,R2

BRz GOWELL

BRn GOSTAR

NO NOT R2,R2

BR STOP

GOSTAR LD R0,SaveR3

BR NO

GOWELL LD R0,SaveR4

BR NO

STOP LD R1,SaveR1

LD R3,SaveR3

LD R4,SaveR4

RTI

KBSR .FILL xFE00

KBDR .FILL xFE02

DSR .FILL xFE04

DDR .FILL xFE06

NEWLINE .FILL x0A

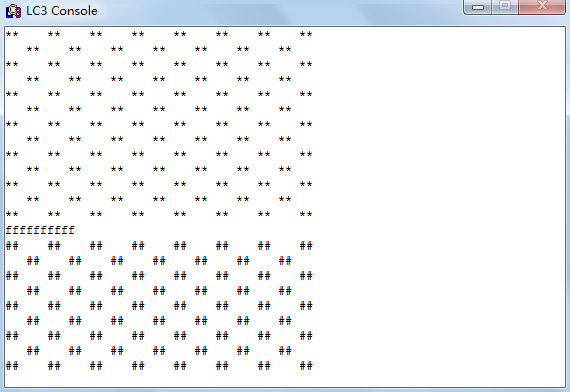
SaveR1 .BLKW 1

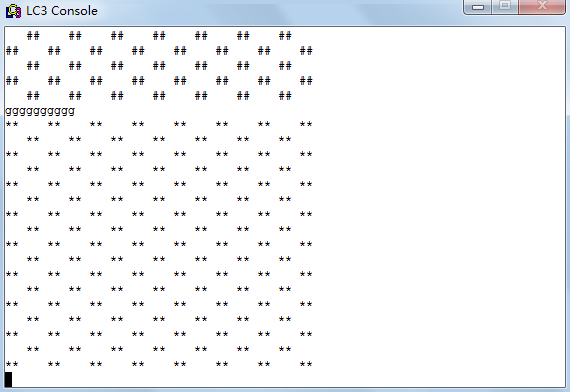
SaveR3 .BLKW 1

SaveR4 .BLKW 1

.END

## 程序运行结果





## 算法分析

本程序通过R2记录当前需要输出的字符，每次进入中断程序后需要改变R2的值以改变需要输出的字符，在每次输出之前判断当前的R2，以确定程序是否进入正确的输入模块，若不是，则通过BR操作转到与当前R2相符的输入模块。

考虑到中断信号有可能在系统调用中出现，当TRAP之前，程序关闭了中断，以保证一行输出的完整性，在TRAP之后，再开启中断。