

一、任务要求

设计一个模拟纸牌 24 点玩法的单片机模拟器，由单片机随机发出四个牌点，测试者在实验板按键上输入计算过程，单片机判断计算结果是否正确，并评定测试者的成绩。

- 【基本要求】：
- 1. 测试多组数据
 - 2. 单片机在数码管上随机给出 4 张牌点，牌点范围为 1-13，10 到 13 由字符 a, b, c, d 表示
 - 3. 在单片机实验板按键中设计 4 个数学运算键 + - × ÷
 - 4. 4 个数字中每使用完一个后该字符应有明显标识
 - 5. 只允许测试者使用四则运算(可加括号)
 - 6. 无论测试结果是否正确，均有声光效果提示；
 - 7. 测试完毕，由单片机给出测试者的最终成绩

二、功能特点与使用说明

- 功能特点：
- 1. 使用最低的系统时钟频率， 降低功耗
 - 2. 键盘扫描使用了软件消抖，防止按键被重复判别为弹起或按下
 - 3. 键值判断使用的是键值跳变判断，即使你一直将键盘按下，也不会重复执行命令
 - 4. 使用后缀表达式计算字符串算式，代码简洁易懂

使用说明
各键盘功能如表所示：

按键	功能
F1	开始游戏，随机生成四个数并显示在屏幕上
F2	重新开始游戏，之前生成的四个数不变
1	使用第一个数
2	使用第二个数
3	使用第三个数
4	使用第四个数
F5	加
F6	减
F7	乘
F8	除
0	括号
点 (K10)	反括号
等于	计算结果，如果为24，则黄灯亮，不为24，绿灯亮

三、方案设计

1. 资源分配：
- (1) 使用外部晶振 22.1184MHz， 使用二分频作为系统时钟；
 - (2) 定时器 T2：
开启中断，系统时钟 12 分频，16 位自动重装模式；
初始值和自动重装值均为 FB80，计算 $(65536 - 0xFB80) * (12 / 11059200) = 0.00125s = 1.25ms$ ，即中断 1.25ms 发生一次；该定时器用来产生 1.25ms 定时，用来扫描键盘，判断其是否被按下，扫描按键，完成一次轮询扫描时间为 $5 * 1.25ms = 6.25ms$
 - (3) 定时器 T4：
开启中断，系统时钟 12 分频， 16 位自动重装模式；
初始值和自动重装值均为 FB80，计算 $(65536 - 0xFB80) * (12 / 11059200) = 0.00125s = 1.25ms$ ，即中断 1.25ms 发生一次；该定时器用来产生 1.25ms 定时，用来扫描键值，并执行相应的键盘操作
 - (4) 使用了比较多的全局变量， 将不需要修改的常量数据存入外部 RAM 中

2. 设计思路:

键盘设计思路: 首先通过 T2 扫描键盘, 判断键盘是否被按下, 这里运用了软件消抖的方法, 连续检测到四次键盘被按下 (或松开) 才会把键值数组里相应的地方置为 1 (0)。然后通过 T4 扫描键值数组, 如果检测到跳变 (防止一直按着某个键, 其键值一直为 1, 导致其功能一直被反复执行), 且跳变后键值为 1, 则执行此键位的命令。

LCD 屏设计思路: 通过字模软件生成数字字母及运算符号的 32 位字模数组, 通过选择左右屏以及生成的 X, Y 坐标可以将数字显示在自己想要的地方

主程序设计思路: 随机生成四个数并将其存在一个数组之中, 然后显示在屏幕的上方。按键选择数或符号, 每选择了一个数, 就会将其原数所在的区域清除, 表示此数以及被选择了, 然后将此数或此符号用字符的形式存在一个字符数组里面, 最后按下等于键, 就会运行计算此字符数组对应的表达式的值的程序, 最后结果算出来如果等于 24, 则会亮起黄灯, 如果错误则会亮绿灯。

计算字符串表达式的设计思路:

1. 中缀表达式转换为后缀表达式:

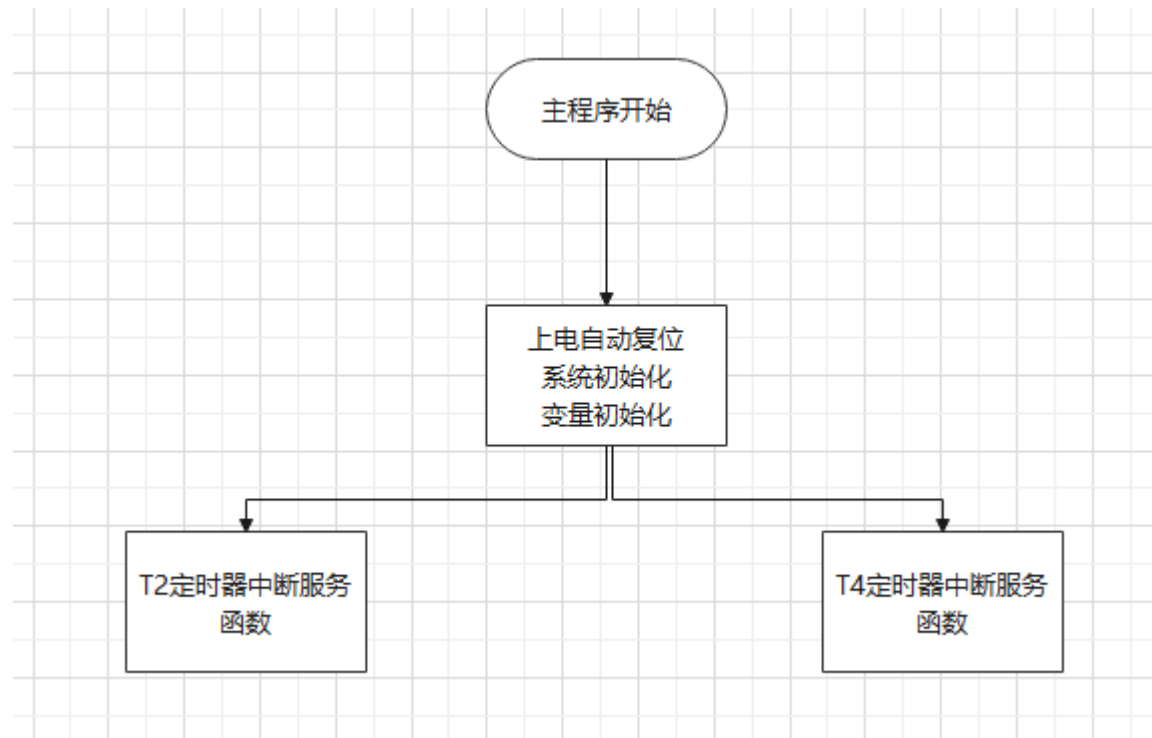
- a). 数字直接入队列
- b). 运算符要与栈顶元素比较
 - ① 栈为空直接入栈
 - ② 运算符优先级大于栈顶元素优先级则直接入栈
 - ③ 小于或等于则出栈入列, 再与栈顶元素进行比较, 直到运算符优先级小于栈顶元素优先级后, 操作符再入栈
- c). 操作符是 (则无条件入栈
- d). 操作符为), 则依次出栈入列, 直到匹配到第一个 (为止, 此操作符直接舍弃, (直接出栈舍弃

2. 通过逆波兰表达式计算结果:

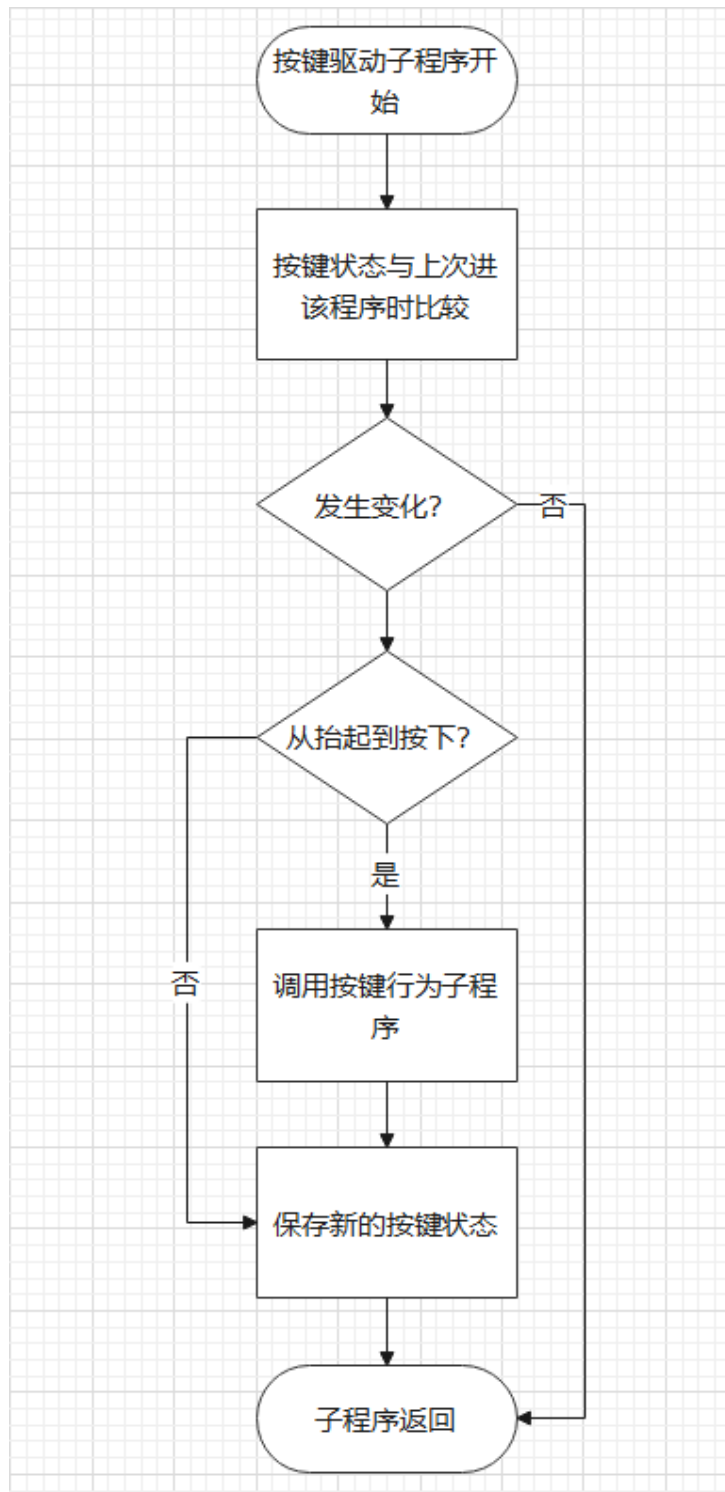
获取字符串表达式的每一项, 判断这一项是数据还是操作符, 如果是数据, 则让它入栈, 如果是操作符, 让数据出栈, 和操作符进行运算

四、流程图

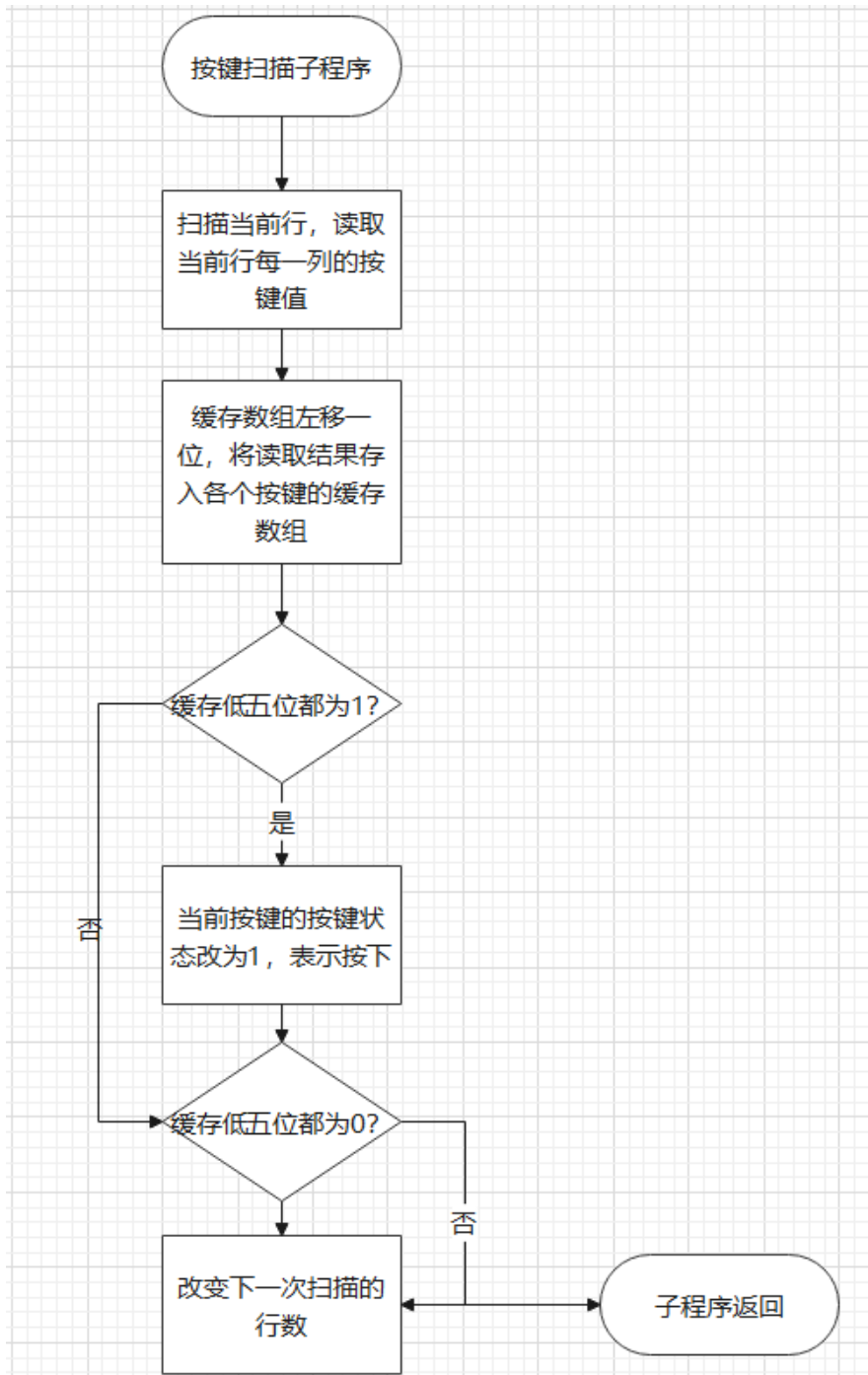
1.主程序 int main()



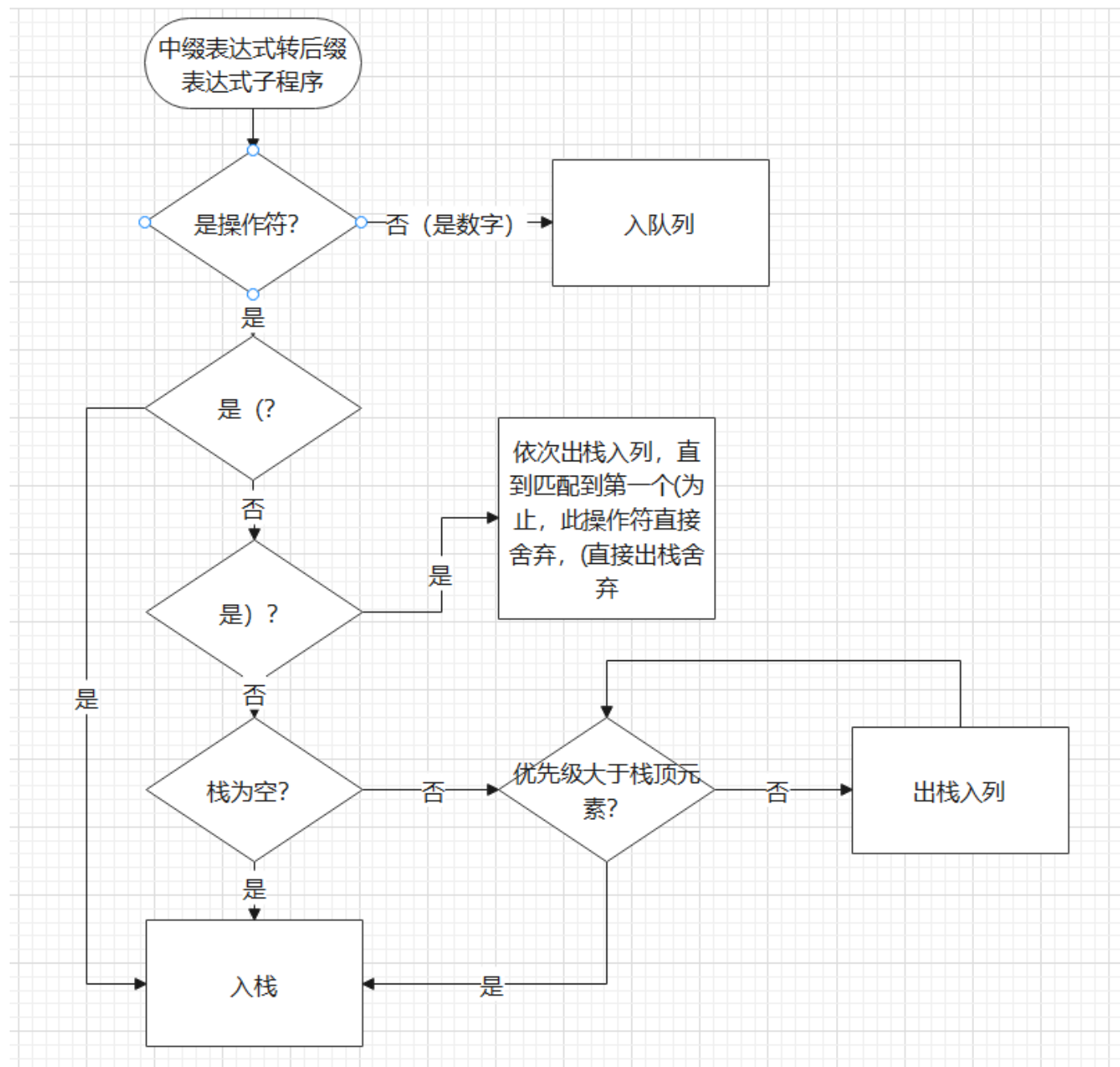
2. 按键检测子程序 void KeyboardDriver ()



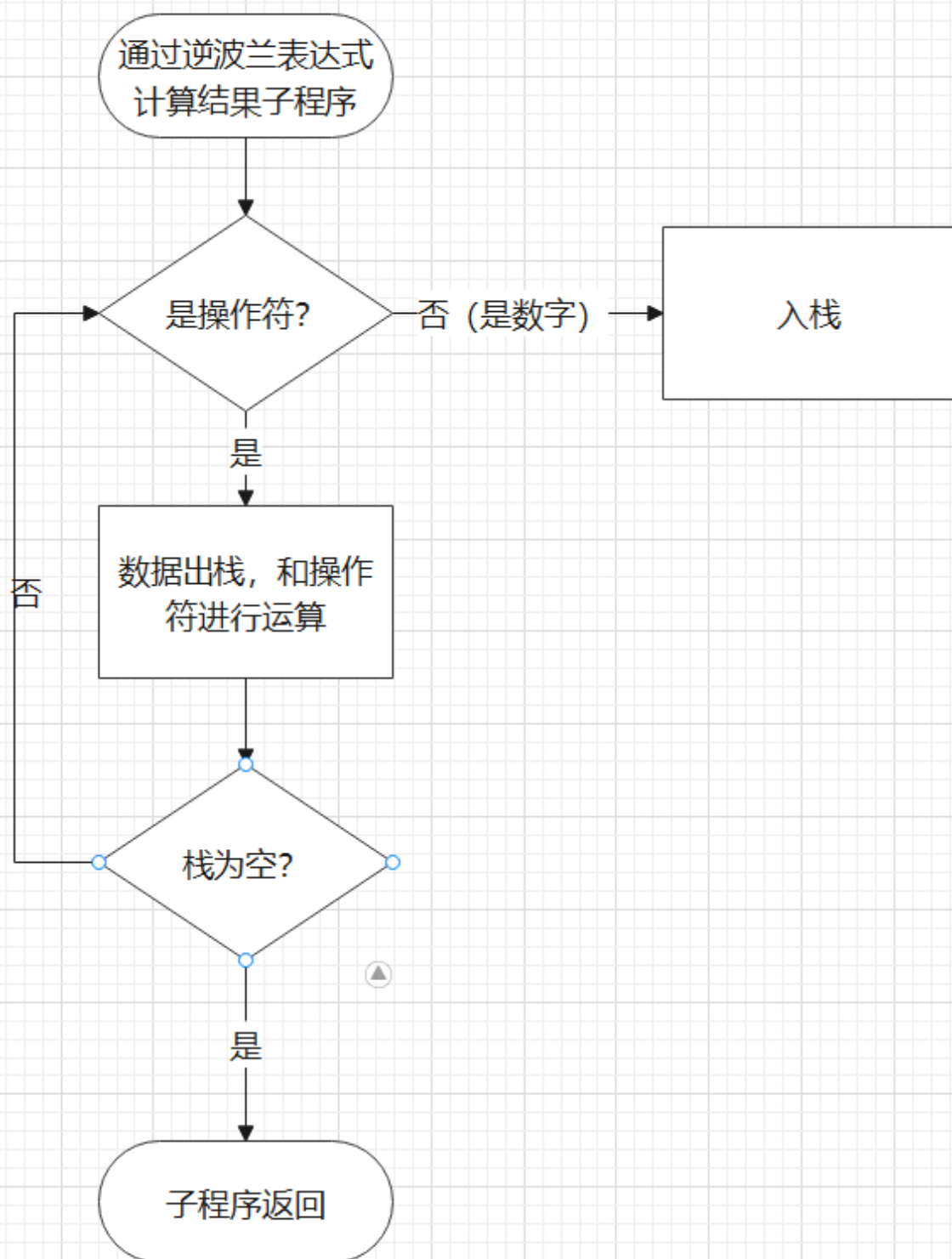
3. 按键扫描子程序 void Keyscan(unsigned char scannum)



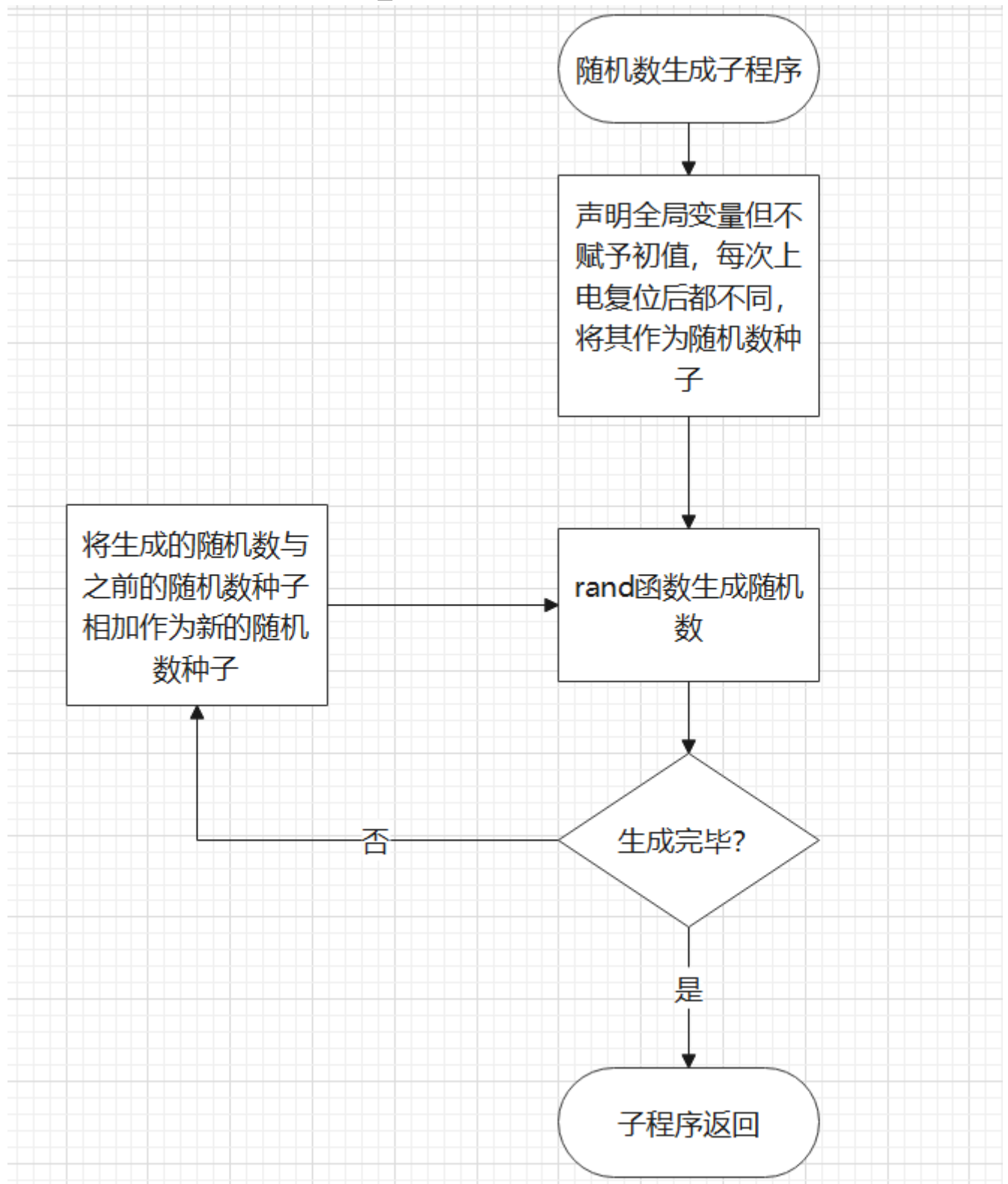
4. 中缀表达式转后缀表达式子程序



5. 逆波兰表达式计算字符算式



5. 随机数生成子程序 void set_24()

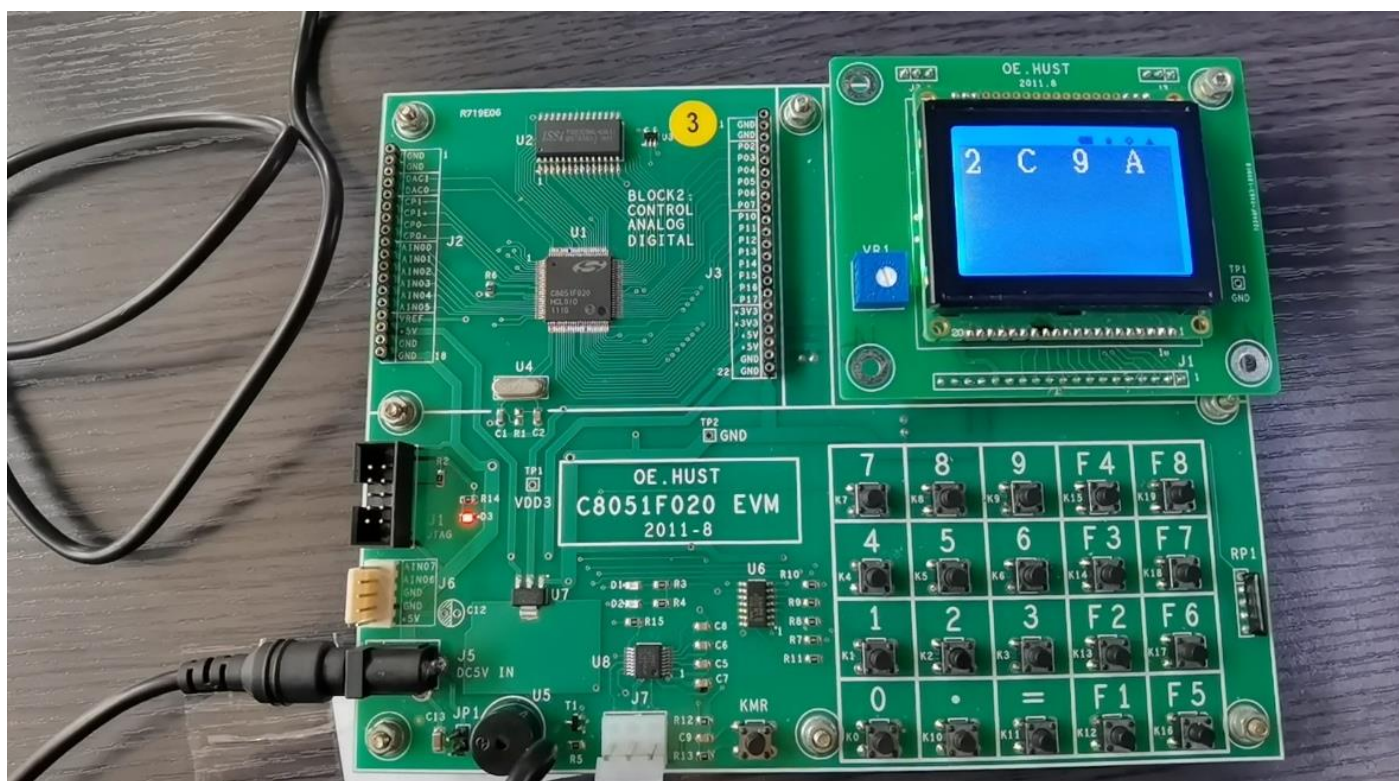


六、程序测试方法与结果

按下 F1，随机生成四个数：



再按一次生成四个随机数



多次按下可见四个数完全随机

测试按键，选择第一个数



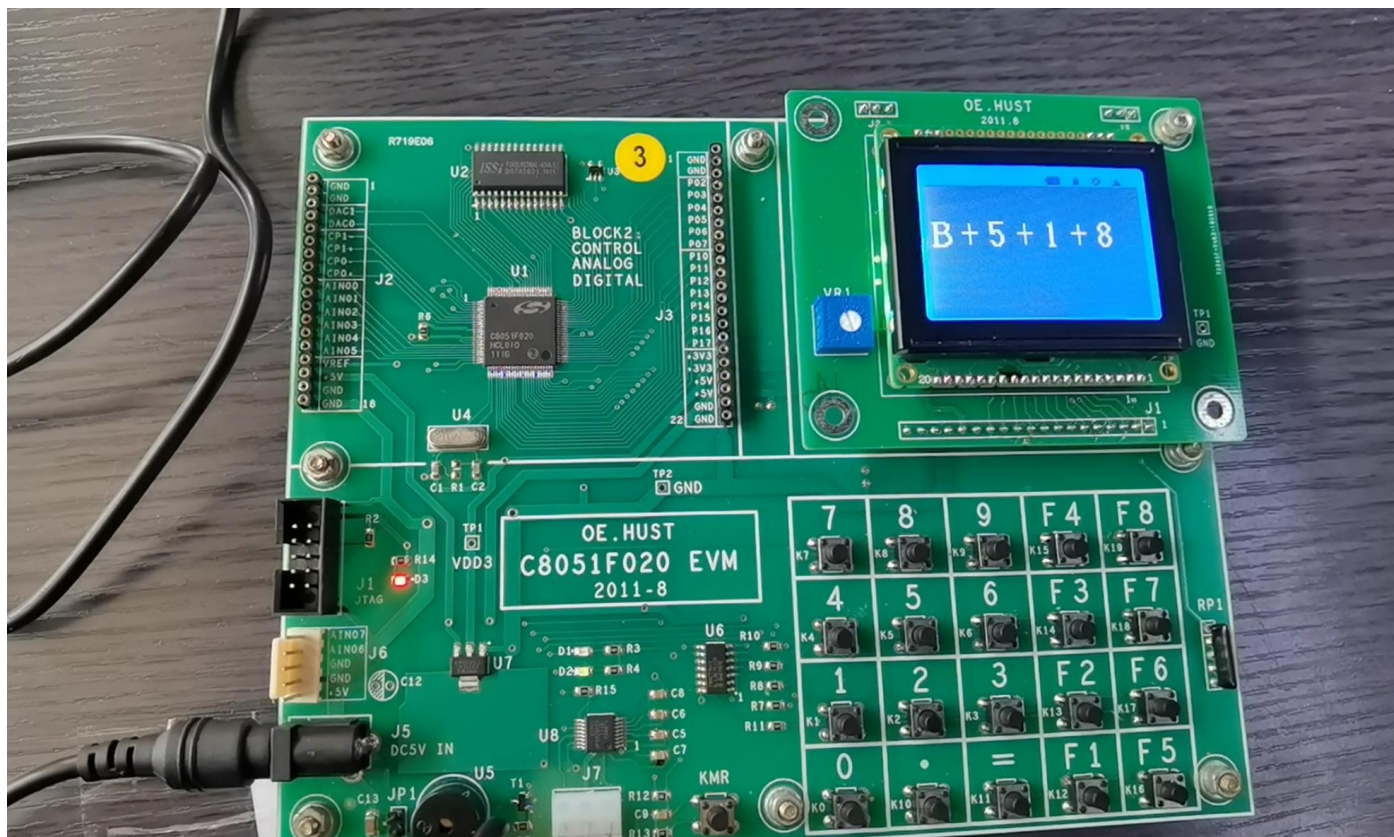
可见第一个数被挪下，且原位置数消失

接着测试+*/<>



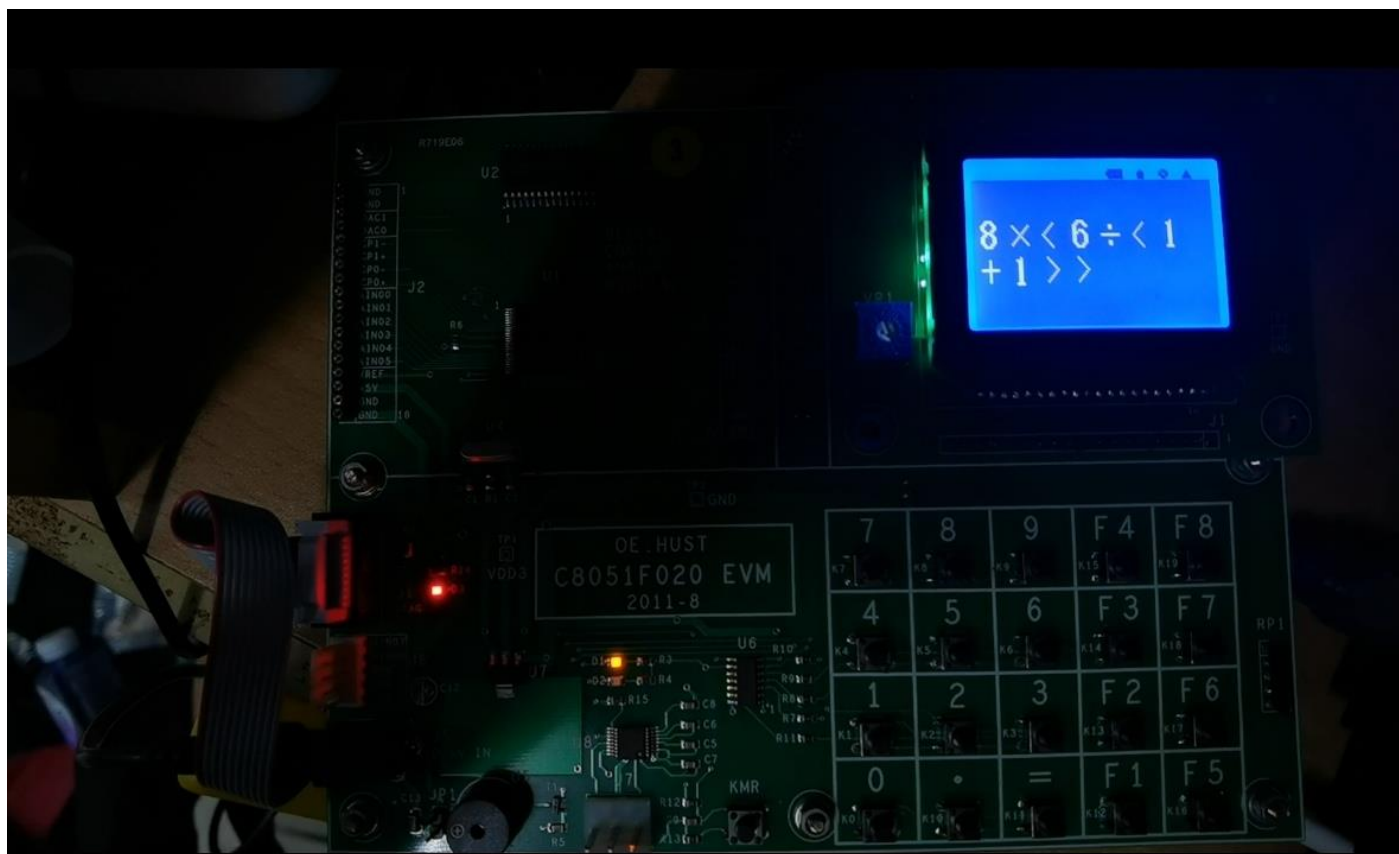
可见功能正常

接着测试计算，首先测试算出来的数不是 24



可见 D2（绿灯）亮起，说明计算错误

接着测试计算，测试算出来的数是 24



可见经过多重括号计算出的值为 24，D1（黄灯亮），说明整个程序是完整可行符合要求的

七、心得与体会

在本次专题实验中我再次深刻体会到单片机是将软件与硬件紧密结合的，软件的编程无法脱离实际的硬件电路而存在，软件的编程是为硬件服务的，硬件工程师需要根据实际电路编写程序，而不是纯软件编程。有时候软件上编写正确，硬件上不一定可以实现所想的功能，操纵硬件，是单片机课程的魅力所在。

同时，通过这次专题实验，我强化了我的编程能力，思维，学到了一些编程上的技巧，也学会如何管理自己的代码等。

本次实验中我遇到的最大困难是全局变量和函数内静态变量的使用，不能滥用全局变量和静态变量，对于小容量的单片机来说，内存资源很宝贵，如果占用太多空间，内存不足会导致程序运行异常，我也在这个问题上卡了很久，以后会更加注意单片机内存的利用。

另外本次实验还锻炼了我的动手能力，满足我动手实践的需求，让我这段时间过得相当充实，谢谢老师给我这次机会。