

设计题目（五）

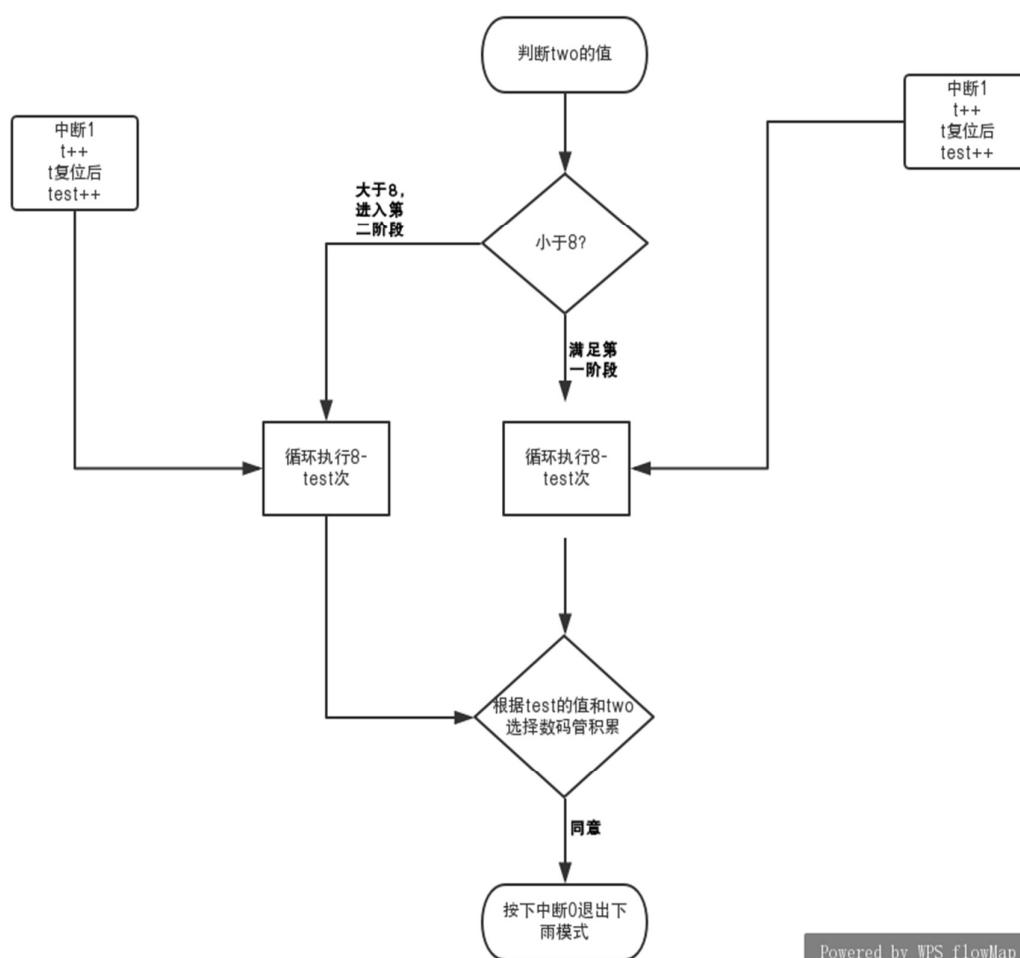
1（100 分）

1) 使用 STC 单片机上的七段数码管，在竖向方向上实现，从上向下“下雨”的效果，填满所有 7 段数码管，然后清空，再重新填充七段数码管。（80 分）

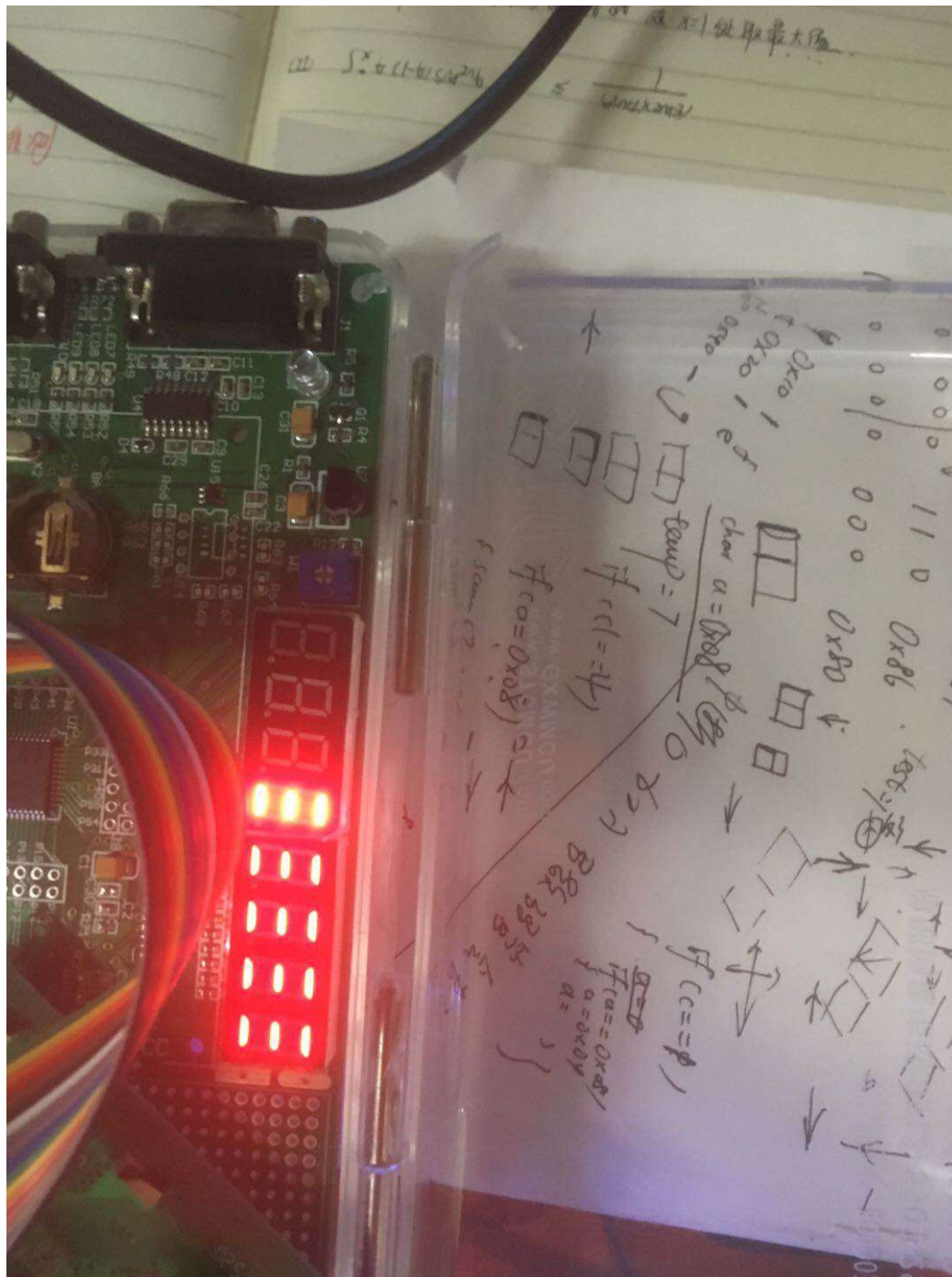
2) 在 7 段数码管上实现”贪吃蛇“演示效果（20 分）

设计思路（以图描述为主）

1.（下雨效果流程图）



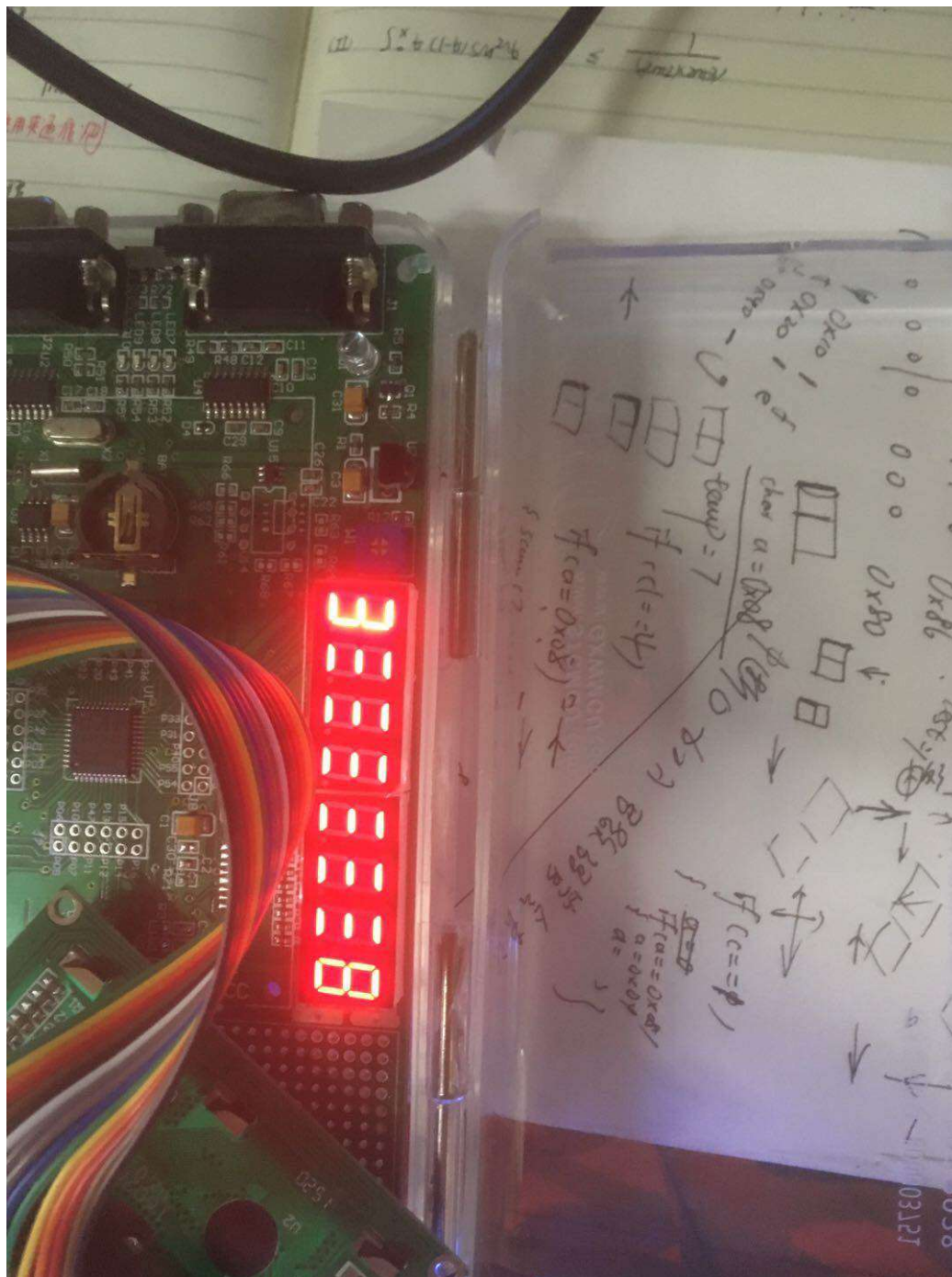
具体解释第一阶段以及积累：



如图，“|||”处于逐渐积累的过程

第二阶段:

两横杠逐渐下来.....，下到最后就积累了一个 8，像俄罗斯方块那种



2.贪吃蛇模式（按下中断 0 触发）

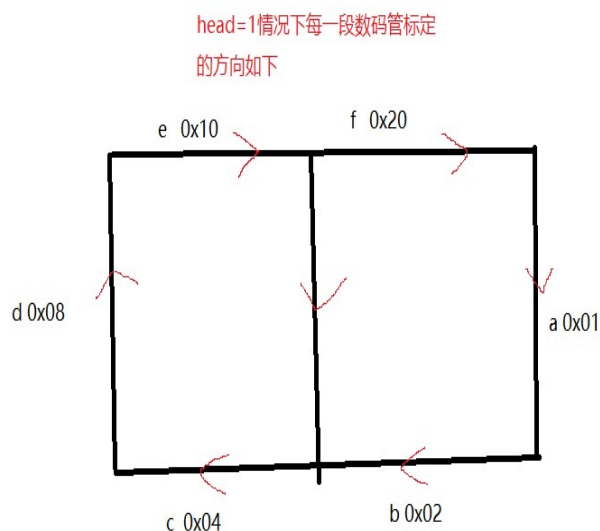
利用键盘上的 4（向前），1（向右），9（向左），6（向后）

(1)首先如图所示，写出每一段亮的时候所对应的段码是什么。

(2)head=1 是代表蛇头的正方向，

例如图下我处于 f 段 0x20，如果此时 head=1，就是蛇头朝向右边，当你按下键盘中的 4（向前）它就会跑到前一个数码管的 a 段，并且蛇头方向与图中标定的正方形相反，所以 head 要置为 0。

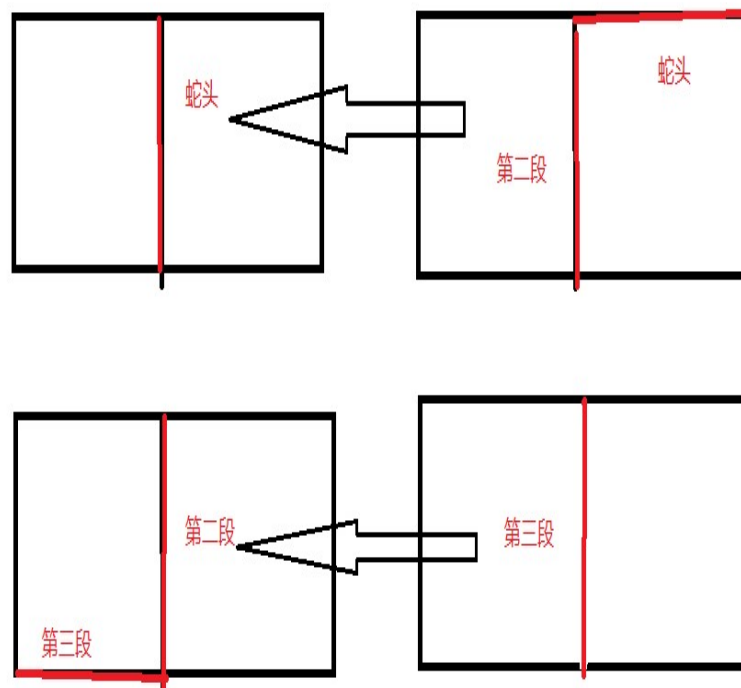
例如图下我处于 f 段 0x20，如果此时 head=0，就是蛇头朝向左边，当你按下键盘中的 4（向前）它就会跑到前一个数码管的 g 段，并且蛇头方向与图中标定的正方形相反，所以 head 要置为 0。



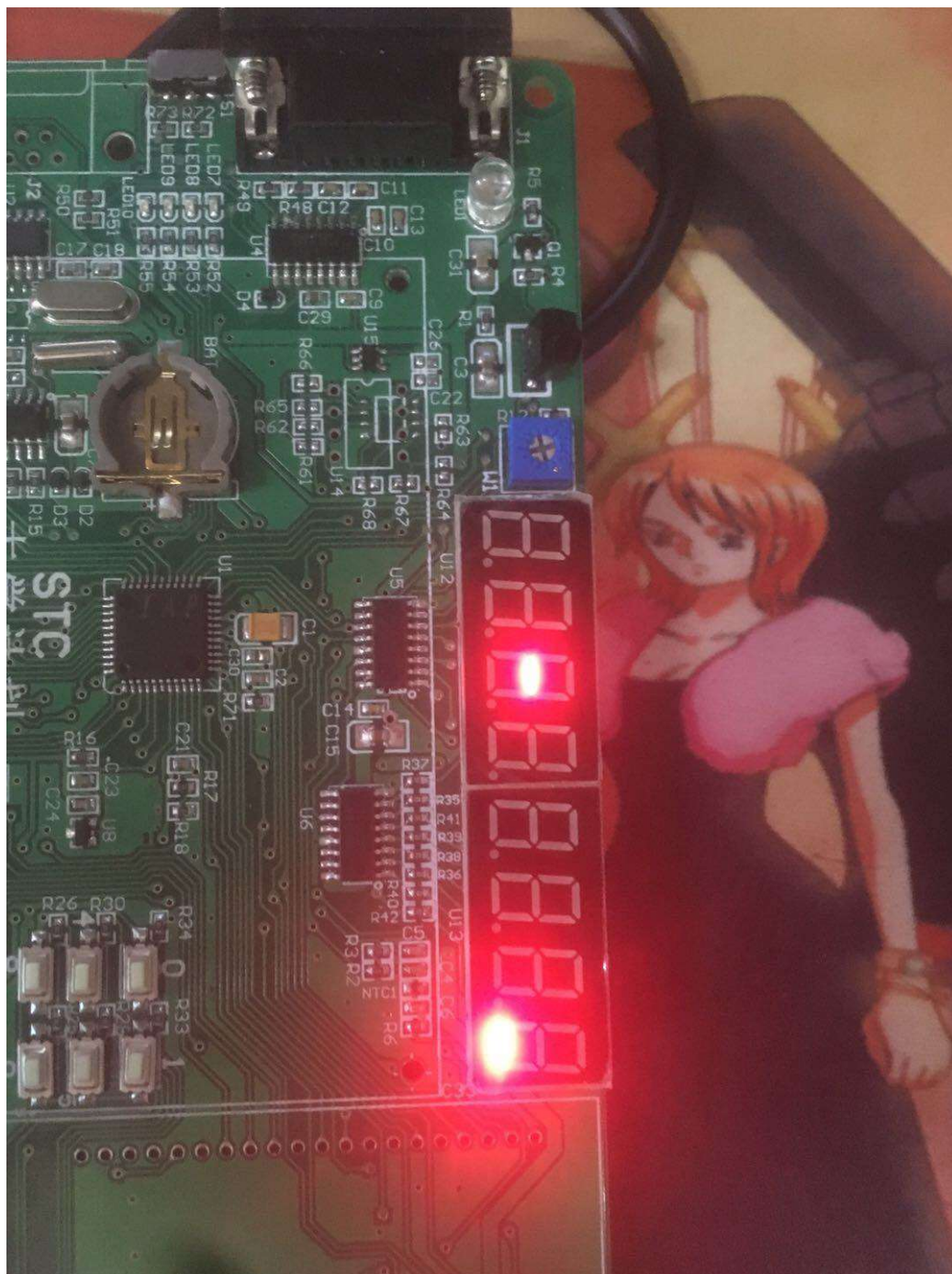
(3)穷举法，利用多个 switch 语句，把蛇头所有能走的方向都写好，那么几乎完成了贪吃蛇的功能了。因为当你蛇长度变长之后，以三段为例，当你按下方向键，蛇开始走，你蛇头前一刻所在的地方就是你蛇第二段要去的地方，蛇第二段前一刻所在的位置就是你蛇第三段要去的位置，所以你只需要写你个 oldtemp 这样的变量保存前一刻的值就能够实现到蛇变成，以及自由行走（代码只写了 3 段长度自走行走）

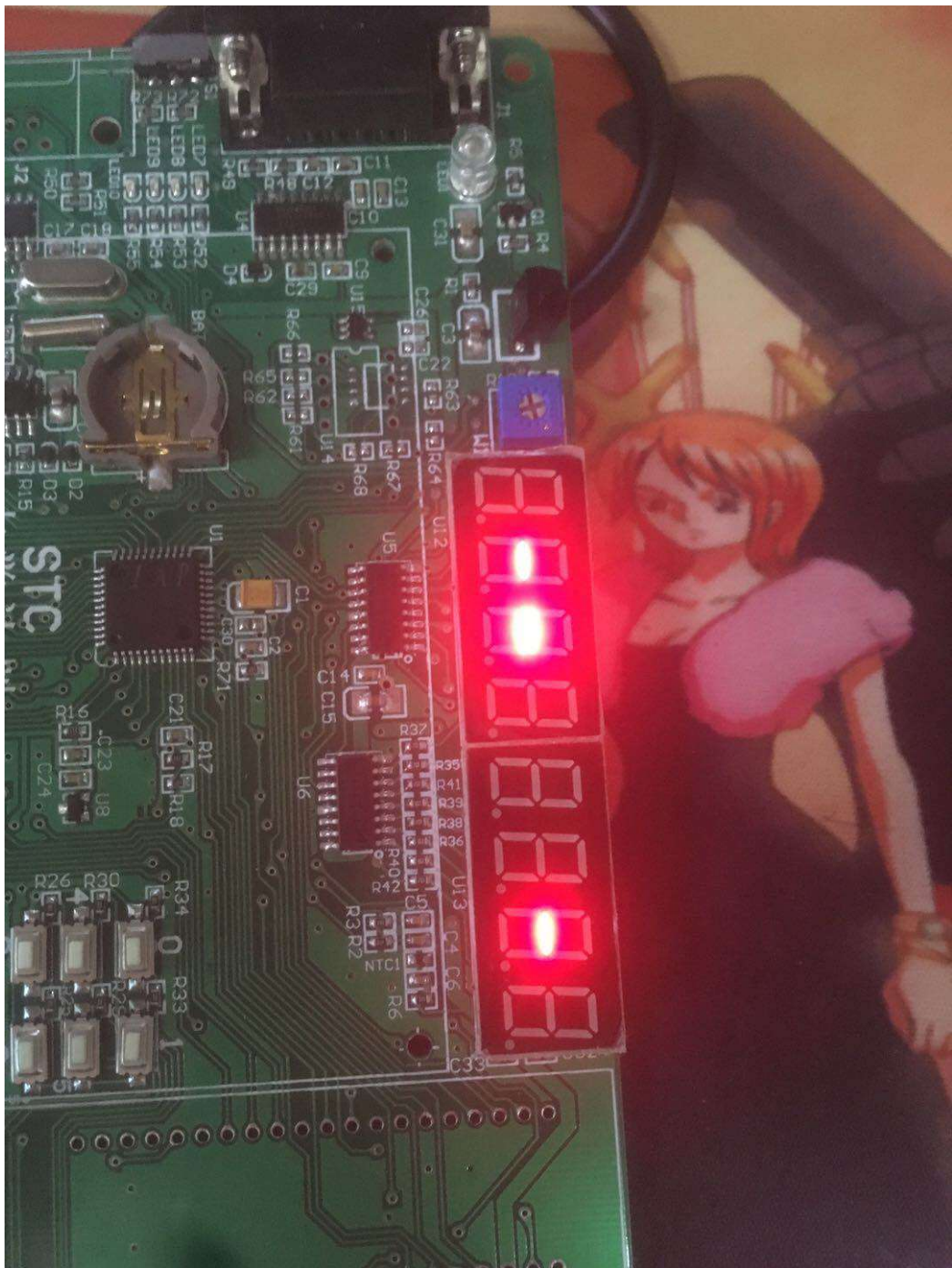
(4)怎么吃：在我代码里头，temp 变量是表示当前蛇头所在的哪个数码管，flag 变量是表示当前是哪个段码，当你走到食物的时候，会判断（if(flag==.. && temp==...)），如果满足，代表你蛇头已经走到那个地方去了，长度会加 1（代码里写的是 b，b=1 代表长度为 1...）

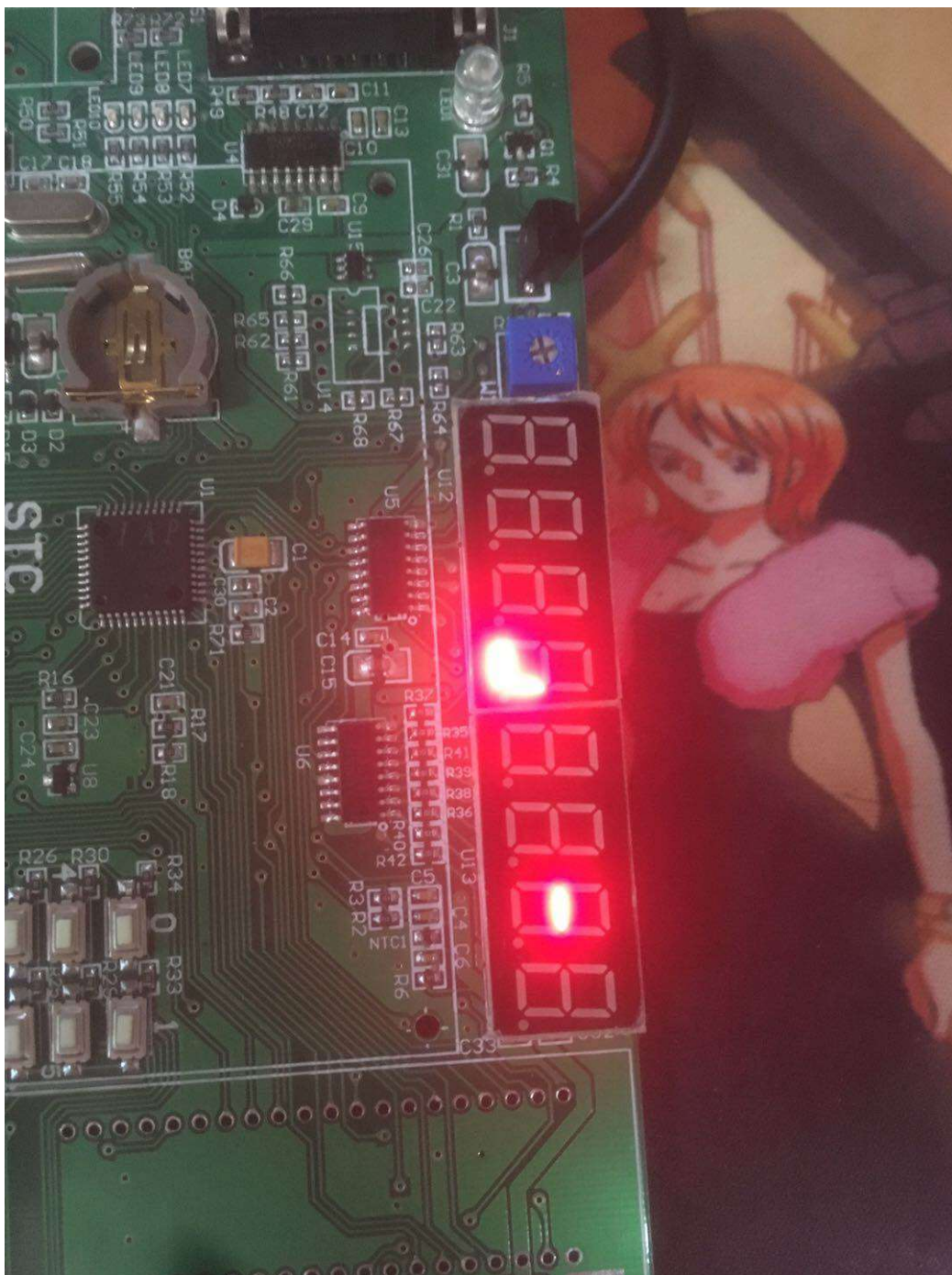
(5)如果让数码管轮流导通：因为进入贪吃蛇之后我把定时器中断关闭了，所以我数码管的导通是依靠单片机执行代码的速度，放在 `while`（循环里面），单片机执行代码速度是微秒级别的，所以数码管显示也不会有问题，成功欺骗自己的眼睛。

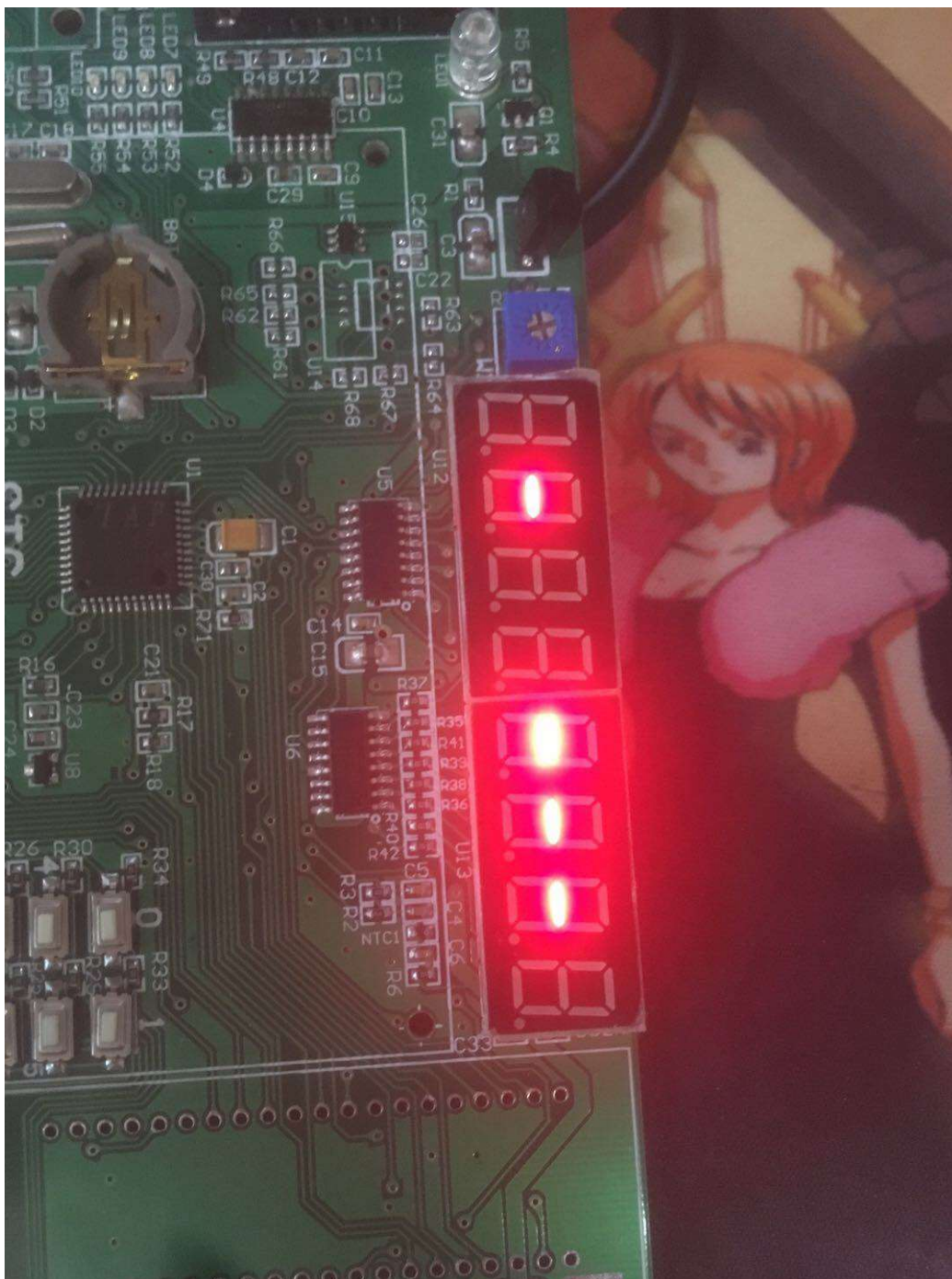


效果展示：

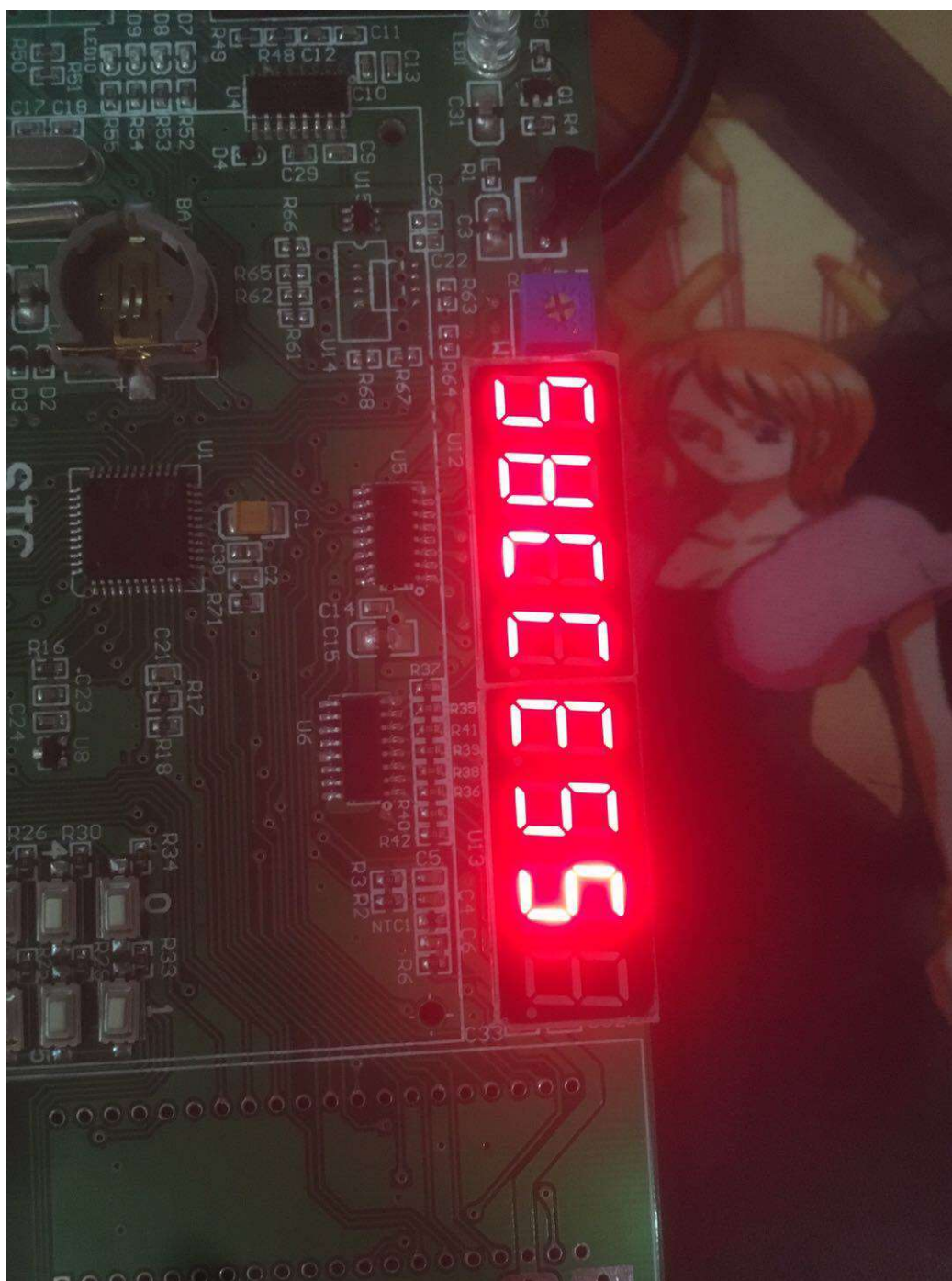








吃完第三次游戏结束.....



实验遇到的困难：

1.开始的时候写下雨的模式写了一大堆逻辑，使得代码很长很长，但是功能是可以实现，没有什么问题，后来因为加入了贪吃蛇模式，代码长度到了 1600 行，觉得太长了，于是我优化了代码，减少了 700 行（700 行都是粘贴复制的穷举法，实在不行）

2.贪吃蛇模式一开始写 switch 语句的时候没有给每个 case 后面放一个 break，使得逻辑都出错了，排查了好一会才发现了这个问题。