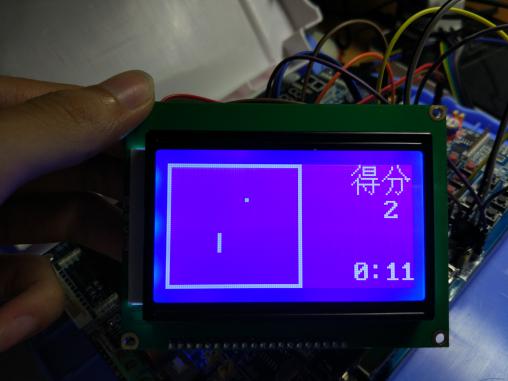
**实战报告**

1. **功能/操作说明**

做的是一个贪吃蛇，用LCD12864，可长按加速键加速，有暂停键可以中途暂停，可以看到已得分数，可以记录已玩时间，满10分可过关，全部只有3关，然而我自己还没通关过，到第三关后来都得无奈自己去撞墙。先看下效果图吧，还没焊接起来，是在开发板上的，报告写完就开始焊，如果交报告前来得及焊完就拍照上来。

开始界面是这样的，按确定键即可开始。然后就是第一关，很普通的地图，

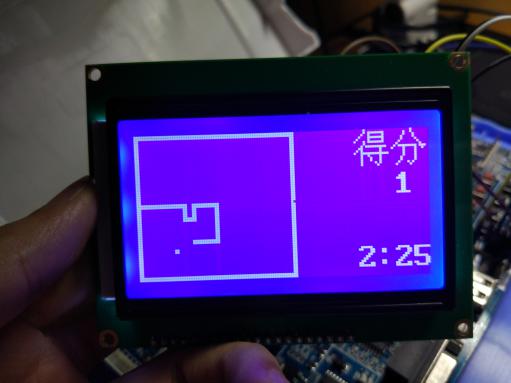


右上角是已得分数，达到10分即可过关，右下角是整个游戏进行中的时间，按加速键时间也不会变快，蛇走得快而已，暂停的时候会停止计时。

第一关过后，会有这个页面。按确认键才会继续，没按确认键的时候计时是停止的。然后是第二个，加了个笑脸的障碍物。



第二关过后，是这个页面，一样，按确认键继续。

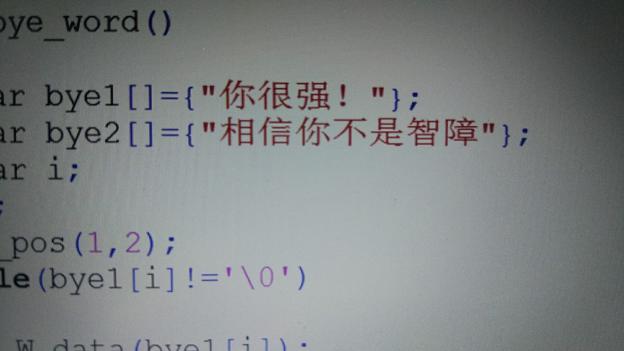


第三关是自己画障碍物的，同样也是得到10分就可以过关，蛇走过的地方就是障碍物。

中途输了的话，会显示这个页面，会闪烁，按确认键则可以回到一开始的页面重新开始。



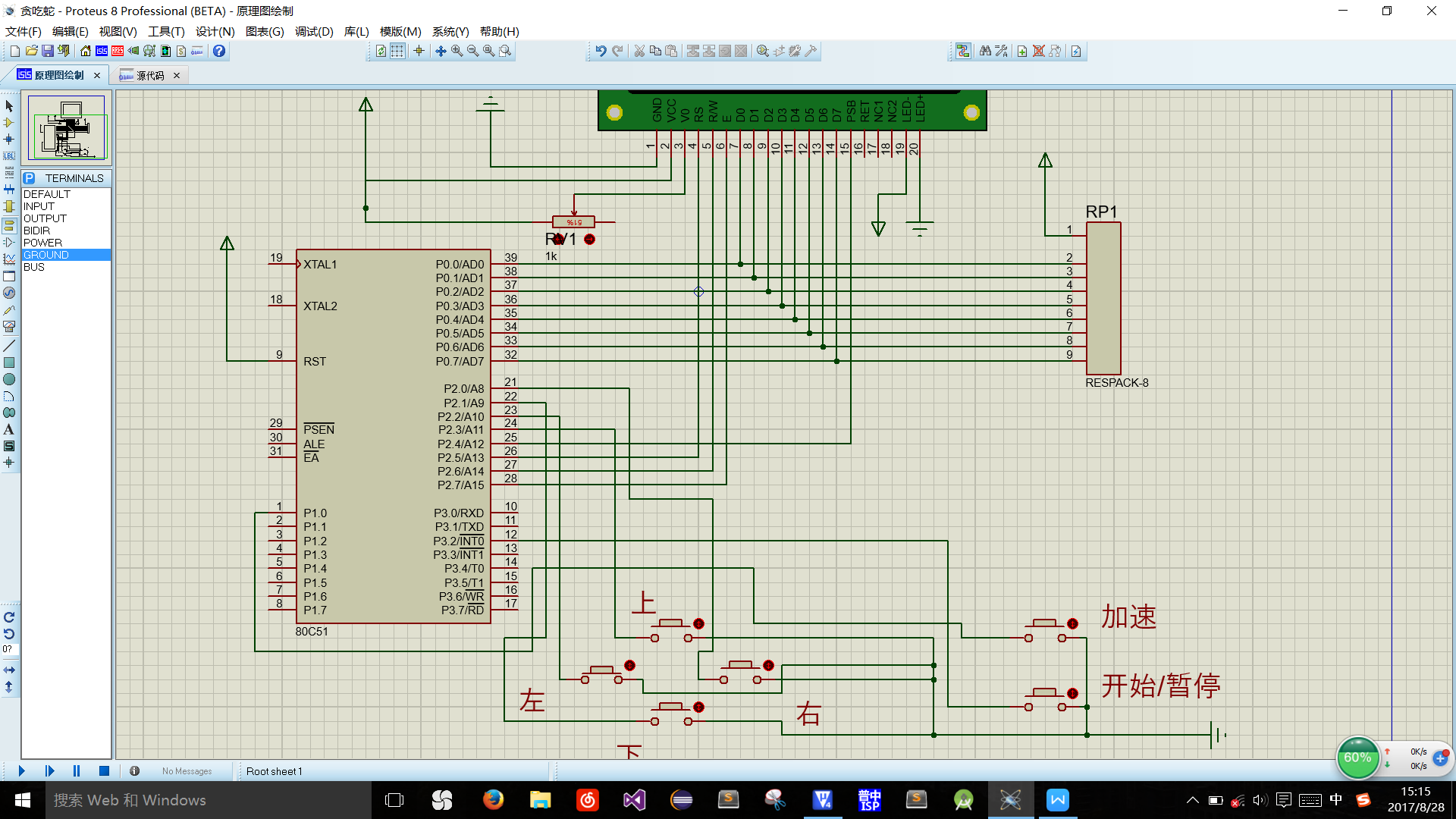
至于全部通关后的页面，我还没通关过，虽然有个很无赖的玩法可以通关，不过我不想那样玩，那样很不好玩，通关后就是这样了。



整个流程就是这样。

1. **硬件原理**

原理图如下：

****

按键是独立按键，开始跟暂停是用外部中断0，计时的部分是用定时器中断0，用的晶振是12.0592MHZ的，LCD12864是5V带字库，ST7920控制的。

主要是对LCD12864的各种操作，RS，RW，EN，PSB等，RST空脚，复位用单片机的复位就行。PSB始终高电平，采用并口方式。RS为数据/命令选择端，高电平表示D0到D7为显示数据，低电平表示D0到D7为显示指令。RW为读/写选择端，高电平表示读数据，为低电平表示写数据。EN为使能信号。V0接电位器用于调背光。

1. **程序算法**

源代码会另外放一个文件。

主要有对一个像素点的点亮及熄灭，用到了按位或跟按位异或，刚好暑假有从图书馆借了本C++的书回去再重温一遍。点亮有参考了网上一个做法，然后对位的判断和熄灭全是自己写的。

点亮一个点结构体即四个像素点这个，我是用左上角的点为基准。

蛇的数据我放在一个结构体里，本来蛇身是打算用动态链表来写的，奈何C跟C++的语法差得有点多，用了一天的时间，还是不习惯也不会用C写链表，所以干脆换成数组写，然后觉得前进一步就得在数组里移动一次太麻烦了，而且数组长度有限，移动的话会有循环，这样就很麻烦，而且操作也困难，就直接数组内位置不变，数值改变，当吃到食物后蛇头的数组下标再加1。后来写后面的部分才知道C得在函数一开始就定义跟声明，跟C++需要就可以定义不同，才知道一开始链表没写错，一直报错是因为定义的问题。

打印食物的函数，用伪随机数的话每次重启食物都是在一样的地方，这样不好玩，我不想这样做，原本我打算用定时器的数值作为种子，就跟用电脑系统时间做种子那样，可是并不是很会，就换成了蛇头的下一节的坐标做种子，因为蛇头的坐标就是食物的坐标，这样其实还是在固定的地方而已，而在第二节的地方就有多个选择了，然而后来调试的时候发现这样在第三关的时候有可能陷入死循环或者严重卡顿，因为在第三关打点的地方比较多，而打印食物的函数每次判断的时候都发现点都是点亮的，这样就一直在重新找点，停在这里导致蛇不会前进，这样蛇的第二节一直没变，srand()的种子也就一直没变，所以就死循环了。所以我在函数开头、循环的外面加了两个变量，用于存储两个种子，而每循环一次，他们就会自增一次，这样就算找到的点是点亮的，它也会自己换个种子继续找，就不会陷在循环里出不来，而且每次都会有不同地方的食物。

按键的判断挺简单的，方向键只要不是跟已存的方向相反，就可以修改，开始/暂停键直接取反就行。

计时函数是用定时器中断0写的，只要每过一秒就调用函数，然后分跟秒是全局变量，按正常电子钟那样判断就行。

在判断是否撞墙那里，我看到网上基本说的是判断蛇头是否离开了地图的坐标范围以及跟蛇身的坐标进行对比判断是否吃到自己，我觉得好麻烦，而且我地图是有障碍物的，这样明显行不太通，然后我在家的时候有写了一个判断像素点是否点亮的函数，点亮则返回1，熄灭则返回0，就想蛇前进的下一个点无非两种情况，点亮跟熄灭，熄灭简单，直接走就行，点亮的话又只有两种情况，食物跟墙壁、障碍物、自身，后三者其实只是一种情况而已，用一个else就搞定了，直接将全局变量snake.life蛇是否活着的变量变成0退出循环就行，食物的话就将数组存多一个，然后再重新生成一个食物出来就行，思路就很简单了。