****

****

**QG最终考核详细报告书**

**题 目\_\_\_\_\_放置类塔防游戏\_\_\_\_\_\_\_\_**

**学 院\_\_\_\_\_信息工程学院\_\_\_\_\_\_\_\_\_\_**

**专 业\_\_\_\_\_电子信息类\_\_\_\_\_\_\_\_\_\_**

**年级班别\_\_\_**\_\_**2020级10班**\_\_\_\_\_\_\_\_\_\_\_

**学 号\_\_\_\_\_3120002524\_\_\_\_\_\_\_\_\_\_\_\_**

**学生姓名\_\_\_\_\_刘骏帆\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2021 年 5 月 15 日**

目录

[塔防类放置游戏 3](#_Toc19913)

[1. 项目简介 3](#_Toc32735)

[2. 总体设计 3](#_Toc26639)

[3. 详细设计 4](#_Toc19737)

[3.1软件设计 4](#_Toc14938)

[3.1.1 游戏界面显示模块 4](#_Toc32238)

[3.1.2 双机通信功能模块 7](#_Toc12250)

[3.1.3 防御塔的放置 7](#_Toc2062)

[3.1.4 防御塔的攻击 8](#_Toc18875)

[3.1.5 关卡结束 9](#_Toc10566)

[3.1.6 断电存储功能 12](#_Toc21151)

[3.2硬件设计 14](#_Toc7572)

[3.2.1 LCD显示模块 14](#_Toc19176)

[3.2.2串口通信模块 14](#_Toc11971)

[3.2.3 按键模块 14](#_Toc15453)

[3.2.4 数码管显示模块 14](#_Toc12800)

[3.2.5 EEPROM断电存储器 14](#_Toc22510)

[3.2.6 蜂鸣器鸣响模块 15](#_Toc27406)

[4. 数据结构和数据处理 15](#_Toc5796)

[4.1 数据结构 15](#_Toc18931)

[4.2 数据处理 15](#_Toc7934)

[4.2.1 游戏内金币的变化 15](#_Toc24421)

[4.2.2 坐标转化 15](#_Toc31306)

[4.2.3 更新排行榜 16](#_Toc1338)

[5. 亮点设计 16](#_Toc16832)

[5.1 较好的程序衔接性 16](#_Toc25503)

[5.2 敌人的成堆分批出现 16](#_Toc26976)

[5.3 LM4229屏幕的使用 17](#_Toc242)

[5.4 断电存储器的使用 17](#_Toc16105)

[6. 系统兼容性说明 18](#_Toc14698)

[7. 总结体会 18](#_Toc9572)

[7.1 项目总结 18](#_Toc9456)

[7.2 项目体会 18](#_Toc6672)

# 塔防类放置游戏

## 项目简介

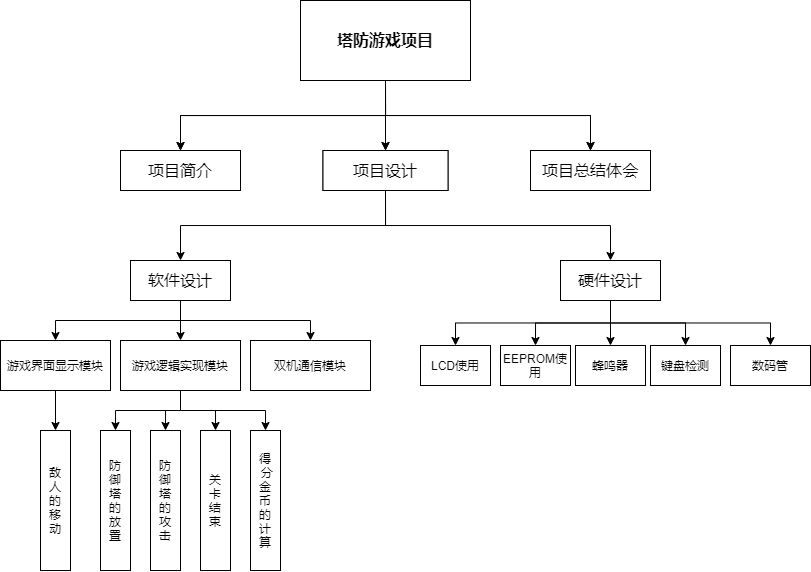
伴随着学习51单片机，我学会了LED灯，数码管，蜂鸣器，键盘检测，LCD液晶屏的显示原理，还有两个单片机的通信。为了将知识输出为作品，巩固自身学习的知识，并且促进自己的学习，所以有了这个项目的产生。

塔防类游戏项目，既能给我们嵌入式的学习带来乐趣，并且还能综合这段时间所学习的，以及未学习然后朝着这个目标去学习的知识得到良好的提升。

在该项目中，用户能够在游戏机上操控光标，在敌人途径两边设置防御塔，以此来消灭敌人，保护自己的基地，从而通关到达更高难度的关卡。

## 总体设计

项目通过使用LCD显示模块，串口通信模块，数码管动态扫描模块，蜂鸣器高低音调转化发出游戏开始与结束提示音的模块，以及键盘扫描模块和EEPROM断电存储器模块实现了，游戏的显示，遥控器单片机，显示单片机的通信，游戏计分，游戏关卡开始与结束的提示，按键控制防御塔放置位置，菜单的选择，还有断电之后的数据保存。

**图 １项目架构图**

## 详细设计

### 3.1软件设计

#### 游戏界面显示模块

##### A. LCD基本显示模块

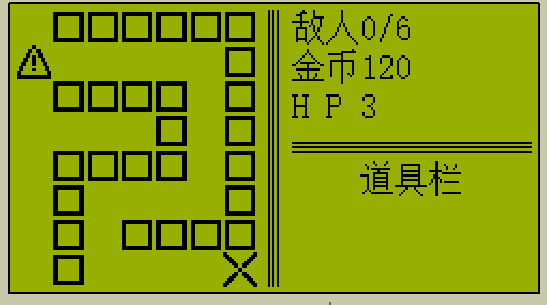
根据网上搜寻到的LM4229指令集，初始化液晶显示屏的时候，先设置文本显示区的首地址，设置文本显示区宽度，以及图形显示区和图形显示区宽度，设置光标为8\*8的方块，显示开关设置文本开，图形开。

液晶显示屏的清屏操作，是通过把整个屏幕都用0x00的熄灭LCD点阵进行显示屏的清屏。

显示函数，显示函数由于字的大小占据了16\*16的大小，而字符的显示只有半个字的大小。所以要根据这个特点取设置两个函数一个是显示汉字等16\*16大小的元素图案和显示半个字大小的字符的函数，否则的话由于存储字符信息的数组再显示完左半边后右半边根据取模的不同所以不能当作16\*16的来显示。字库信息是通过取字模工具，取模方式选择逐行式，取模走向设置为顺向高位在前。同时显示函数还将屏幕划分为了0<=x<=28,0<=y<=112的坐标来给显示精确的控制。

##### B. 静态游戏界面

静态游戏界面显示包括：用户的交互菜单界面，游戏的关卡图，以及游戏内一些相关信息例如：剩余敌人，金币，以及我方基地HP等元素此处的静态显示，是通过for循环以及封装好的LM4229显示函数的16\*16方块显示函数，对静态的一些游戏界面进行函数封装。



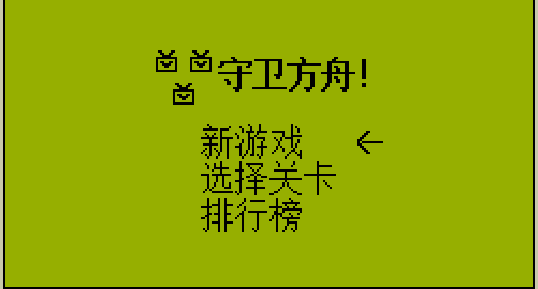
**图 2第二关静态界面图**

##### C. 动态游戏界面

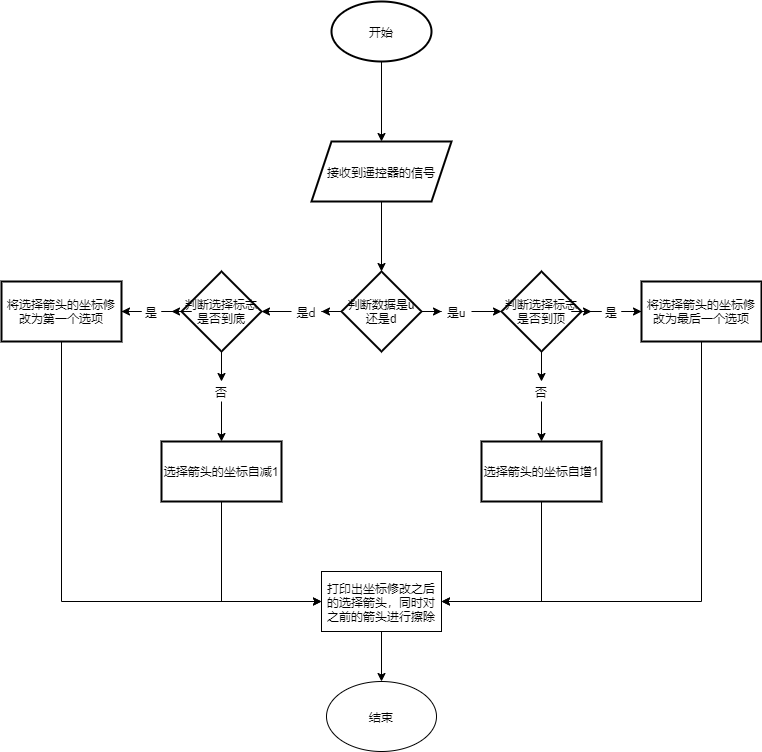
动态游戏界面显示，指的是给予用户提示的选择标志，以及游戏内敌人的移动，还有金币，剩余敌人，我方基地HP的剩余数值的变化，防御塔的放置。我是通过给显示函数的坐标予以（x,y）坐标的变量，通过此变量来改变相关元素的显示，但是此时会出现全屏重复的情况，所以我还设置了一个“擦除方块”，即16进制数均为0x00的元素，对移动方块的“影子”进行擦除。

###### 菜单的选择

在菜单界面的箭头移动，我是通过串口通信接收到的信号对于箭头的坐标进行处理，同时判断，如果坐标超出了一定的范围那么并不是让用户的操作进行不处理，而是人性化的使用户的操作得到合理的反馈。例如：在主菜单页面的时候，如果用户在第一个选项中还选择了向上的信号，那么程序会经过判断把箭头移向最下面那个选项，反之箭头在最后一个选项的时候同理。



**图 3游戏主菜单图**



**图 4菜单选择流程图**

###### b. 游戏中敌人的移动

在游戏内敌人的移动中，我选择了用全黑的方块对敌人进行表示，敌人的移动是通过定时器中断，设定好一定的时间，对定时器中断之后，给予敌人的坐标发生变化，同时“擦除”敌人的影子。因为每一个关卡中敌人移动的路径是不同的，所以我封装了一个函数对敌人的移动路径也进行了规定，当敌人横向移动到某个值的时候就会竖向移动等。

由于每张地图中的敌人移动路径是不一样的，所以我定义了一个全局变量标记关卡的选择，从而在定时器0进入中断服务程序的时候，通过switch函数来对不同关卡的敌人移动路径进行不同的选择。

并且为了实现敌人分批分波次的出现，或者说实现敌人的等距出现，我对后续出现的坐标进行了定义变量，当前面一个方块走到指定的位置的时候，后续的方块跟着一起通过之前的方法也跟着实现。

同时在第二关时，由于地图比较复杂，所以在敌人路径移动的时候可能会产生，非一步一步的移动，而是在一次定时器中断中坐标移动了两次，后发现是因为if语句的使用使得前一个if让坐标增加之后恰好满足后一个if的条件，使得坐标移动两次。为了避免这样的发生，我想要让程序执行完第一步直接就结束中断服务，我选用了else if的嵌套，如果坐标满足了前面的if条件，那么就不会执行后面的else if。

#### 3.1.2 双机通信功能模块

发送方初始化：对SCON串口控制寄存器，设置串口工作方式为1，10位异步收发16进制数据的传输，开启总中断和串口中断，然后通过键盘检测，检测出不同的操作给变量send\_data予以不同的值，之后通过SBUF发送到一号单片机，由TI发送中断标志位来判断是否发送完全。

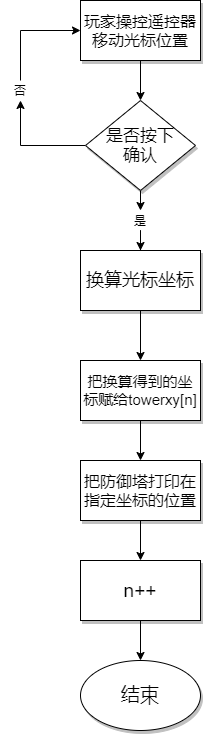
接收方初始化：先对TMOD定时器寄存器进行选择定时器1为工作方式2，自动重装初值；根据单片机的波特率9600设定好单片机的初值为0xfd，同样的和发送方一样，对SCON串口控制寄存器，进行串口工作方式设置为工作方式1，打开REN允许接收串口信息，打开中断。与接收方不同的是，这里需要设置一个定时器来不断的接收SBUF里传送过来的数据，同时接收到之后要软件清零接收中断标志。

发送方的键盘扫描，把信号发送之后以及接收方根据接收信号执行相应功能后，都需要及时的将发送或者接收的信号置0，否则会造成，发送方不断发送，接收方不断执行的错误。

#### 3.1.3 防御塔的放置

进入游戏界面之后，LCD屏幕会开启光标闪烁功能，根据用户遥控器经过双机通信之后，根据收到的信号，对光标的坐标进行改变，使得用户可以操纵光标的位置，从而确定用户即将要放置防御塔的地方。

当用户按下确认键之后，设置好的存储防御塔坐标的数组，会保留当前光标x,y的坐标，之后在液晶屏上打印出防御塔，从而游戏内的金币也得到相应的扣除，也就是记录游戏金币的变量会得到相应的变化，再打印一边游戏信息，随后标记第几座防御塔（数组下标），的变量也会相应的自增1，表示已经放置了一座防御塔了。



**图 5防御塔放置流程图**

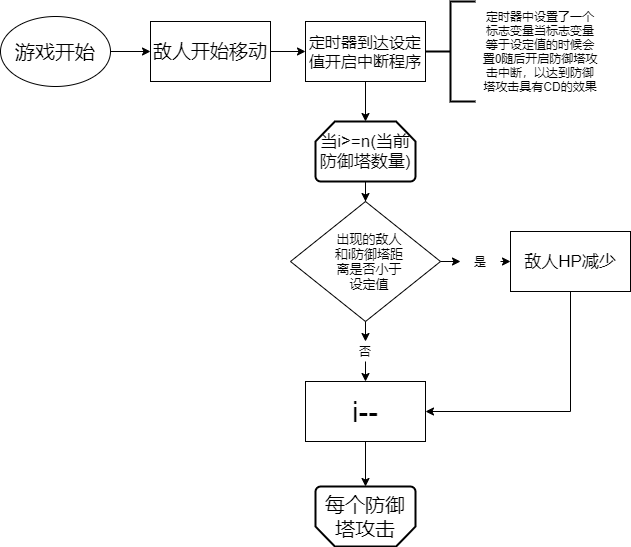
#### 3.1.4 防御塔的攻击

根据存储防御塔坐标的(x,y)与敌人的坐标进行距离计算，当小于设定的值的时候，敌人就受到防御塔的攻击

同时，为了使得敌人具有一定的血量不会直接被防御塔攻击到就死，所以会对应的设有，每个敌人的血量。敌人会在受到攻击之后，进入损伤状态，程序会根据HP是否为满的来判断该显示哪种状态的敌人。

当敌人进入到防御塔攻击距离的时候，敌人的血量会得到相应的扣除，但敌人血量扣除为0的时候，再把敌人擦除之后，敌人数减1，随后重置敌人的坐标；并且屏幕的左上角会出现相应的被击败的敌人，表示已经有敌人被击败了。

并且为了设计出防御塔的攻击是具有冷却时间的，在定时器中断中，只有当标志变量到达设定的值的时候才会除法防御塔攻击中断，从而达到了防御塔攻击具有冷却时间的功能。



**图 6防御塔攻击流程图**

#### 3.1.5 关卡结束

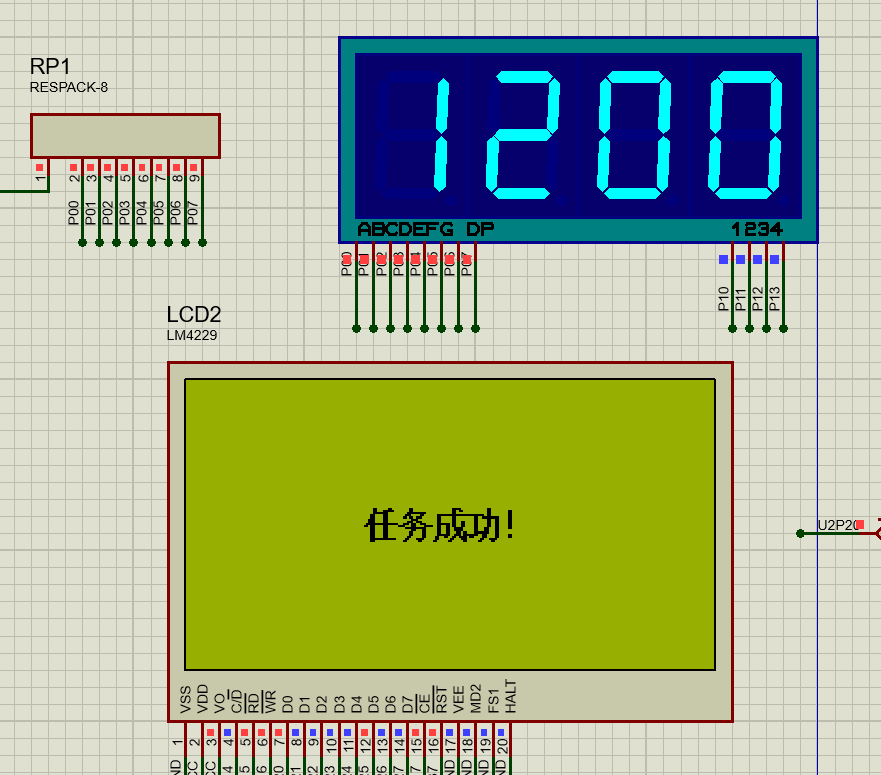
在关卡结束时候，也就是敌人均被打败或者我方基地HP被减为0的时候，即判定本关卡结束。

程序会在if判断敌人数量是否为0的时候，如果为0紧接着会判断我方基地的HP是否为0来代表游戏的成功或者失败，并且都会把已经设定好的关卡结束标志变量进行置1，随后才能跳出遥控器遥控光标的死循环。若游戏成功还会计算玩家本关卡的得分并且加上之前的得分在数码管中显示出来，具体计算方法是：剩余金币数量+击败敌人的数量\*10+基地HP\*100。

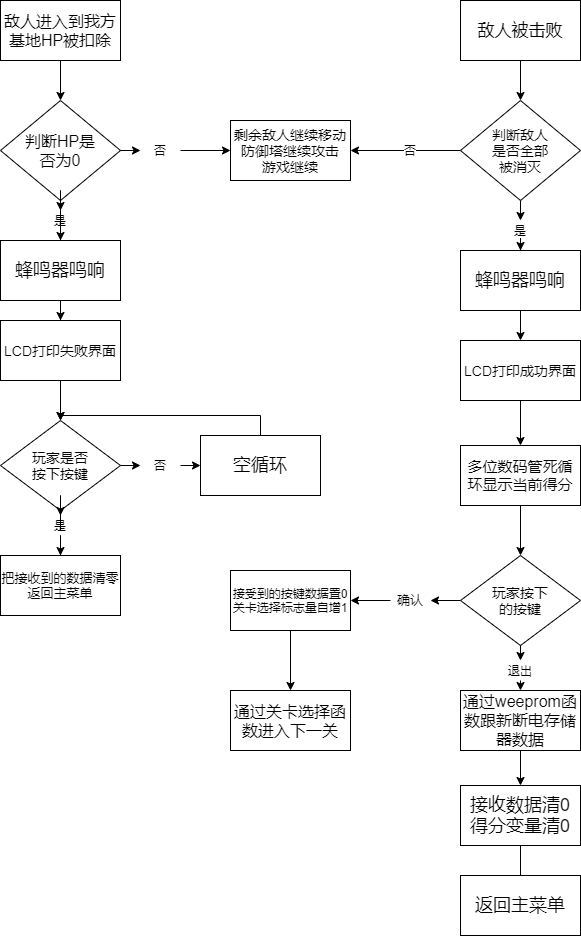
随后根据玩家选择的按钮是确定还是取消来判断是否进入下一关卡，同时，因为多位数码管的显示需要用到数码管的动态显示，在显示得分的时候需要一个死循环不断的去一次显示数码管的每个位数，所以为了跳出这个死循环，并且节省一下单片机内部RAM的使用会通过判断关卡结束标志来判断是否跳出死循环。

如果玩家选择了确定的时候，关卡标志变量会变为0，关卡选择变量自增1，然后跳出已通过关卡的死循环，回去上一级的switch(关卡选择)进行选择，并且因为死循环的跳出，数码管计分器会停止工作。

若玩家选择了取消，那么会记录当前的得分，先存入长度为5的数组中，进行排序，之后再通过循环依次存入EEPROM断电存储器中。



**图 7关卡成功图**



**图 8游戏结束流程图**

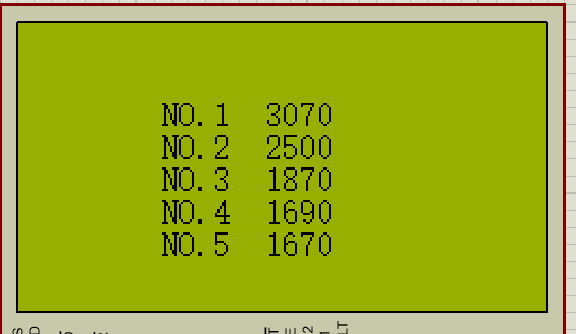
#### 3.1.6 断电存储功能

根据eeporm使用的时序图来对eeprom的一些初始化、开启、等待应答、停止、写入字节、读取字节等基本函数先写好。

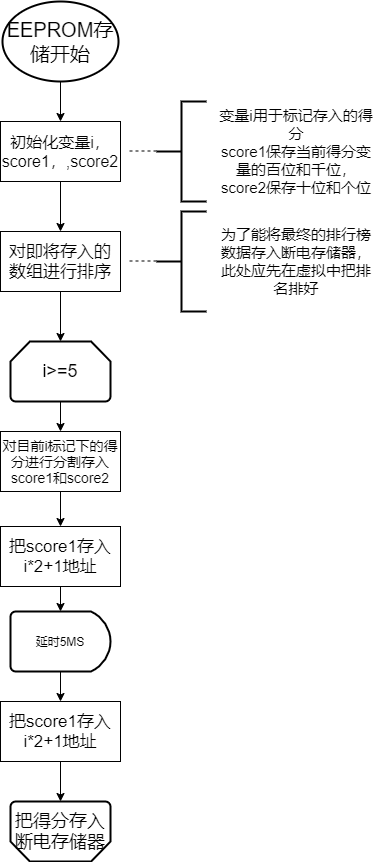
随后编写，写入数据，读取数据函数。写入数据先是启动、写入指令告诉EEPROM是操作、等待响应、写要存入数据的字节地址、等待响应、写入数据、等待响应、停止。读取数据操作和写入数据操作类似，先启动EEPROM随后这里要写入先像写数据的写指令部分一样，先对EEPROM写入写指令，之后写入要访问的地址，等待响应之后，在写入读指令，之后读取出指定地址的数据。

在此项目的断电存储功能，是用于存储玩此游戏每次成功关卡的累计值，所以它是需要存储一个四位数的得分的，但是如果只是单纯的使用写入数据函数的话是显然不行的，因为写入数据函数是指定一个字节地址进行数据存入的，所以一个字节不能超过255大小的数据，这时候就需要将得分分为两部分进行存储了。

并且，在存入数据以及读取数据的时候，经过多次的实验发现，在写入或者读取完一个数据之后，需要人为的增加一个延时函数，而这是我在看书没有学到的，因为书上的内容是把60S计时器的秒数存进去，而这时候它是已经隔一秒才记录一个数据了的，所以不需要延时。这次我是通过自己实践总结发现的，否则数据将会错乱。



**图 9游戏得分排行榜图**



**图 10得分存储流程图**

### 3.2硬件设计

#### 3.2.1 LCD显示模块

LCD显示模块相当于是这次项目中的核心，所有的游戏画面，以及和玩家的交互都是需要在LCD上显示出来的，为了便于玩家的游戏体验，这次的项目选用了240\*128大小，驱动器为T6963C的LM4229显示屏。

这块显示屏是通过R/D端和W/R端来控制读和写的与平时的R/W端口的指令不一样，并且它是通过C/D端来实现对数据还是指令的控制，同时与平时RS端口控制的电位是相反的。

#### 3.2.2串口通信模块

串口通信中的串行通信，将两个单片机的TEX和REX进行交叉串连，由于是在proteus内的仿真单片机，所以默认两个单片机共地了，就不需要我们人为的将单片机的接地引脚进行共地。

#### 3.2.3 按键模块

游戏的遥控器是通过二号单片机执行相应的功能的，键盘扫描模块中，设置有上下左右、确认、取消、重置按键来成为玩家操控游戏的一个接口。

在按键模块中，按键连接二号单片机的一组I/O口，随后不断的对I/O口的电位进行检测，不是初始状态的时候进入while循环语句，等待用户把按键松开的时候执行相应的功能。

#### 3.2.4 数码管显示模块

玩家通过一关之后就会由四位数的数码管来充当游戏当前分数的计分器，通过数码管的动态扫描，来显示玩家的得分。

同时，由于数码管接的是单片机的P0口所以需要上拉电阻来使得数码管能够显示。

#### 3.2.5 EEPROM断电存储器

EEPROM断电存储器，采用的是24C02C型号的存储器。通过使用IIC总线通信，使用SDA数据线和SCL时钟线，进行数据的传输，从而来实现游戏程序断电存储的效果，使得游戏能够拥有排行榜的效果，用于存储不同名次玩家的得分。

#### 3.2.6 蜂鸣器鸣响模块

在单片机的一个串口中，连接了蜂鸣器，当玩家进入游戏以及游戏结束之后，蜂鸣器通过高低电平的转化会鸣响表示游戏开始以及结束。

## 数据结构和数据处理

### 数据结构

在此次的项目中，运用到的数据结构只有结构体数组，用于防御塔，敌人坐标以及信息的存储，以及得分排行榜的存储。

### 4.2 数据处理

#### 4.2.1 游戏内金币的变化

最初我的设计是分别定义金币的个位数，十位数和百位数，然后通过字库里面的数字字模，因为是16个数据作为一个字符显示的数据，并且LCD会根据写一个数据之后把数组继续往后面读16位数（显示字符）或者32位数（显示汉字）。所以只要改变存放0~9字符数组的首地址就可以达到相应的显示，封装成函数之后，就是在参数位置上输入金币数量的各位的数值就好。所以最初认为这样比较清晰简单，明了。

可是到了放置防御塔然后金币得到相应的扣除的时候，这样的处理会很麻烦计算，我不能清楚的知道该扣除哪位上的数据，数字该怎么变如果0之后再减那又是另外一种情况，所以这样的思路是不可取的。

同时也为了节省单片机中内部的RAM空间，我采用了金币作为一个变量，利用整型数据进行除法运算数据丢失的特点，分别求出了它的各位的数字，之后通过函数显示。

#### 4.2.2 坐标转化

在程序中，光标的大小是8\*8的大小，而光标的坐标是按照（2，2）移动的，而lcd里显示文字和字符的单位是按照（2，14）来进行移动的，所以在记录光标位置给防御塔坐标的时候，需要通过一个换算来得到合适的防御塔坐标。

#### 4.2.3 更新排行榜

为了实现玩家结束游戏之后对得分进行保存，并且更新排行榜，而且在过完一关之后不会马上把他的得分进行排行榜更新，而是在他失败，或者退出游戏之后才对此次游戏的得分进行排行榜更新。

所以要对玩家的得分进行数据的处理，我先将玩家的得分不断存储在一个无符号整型变量中，当玩家游戏失败之后，或者选择退出游戏之后，会把该变量的值存入一个记录排行榜的数组中。

但是这里又会有一个问题，我该如何确定好此次得分应该在排行榜的位置。

所以在这里，先把得分存在了数组的最后面之后再使用了插入排序对数组进行排序，排序完之后就相当于在数组中创建了一个虚拟的排行榜，排序之后被放到第六位的就相当于淘汰了，由此来形成排行榜的效果。

之后会使用写入eeprom函数把数组的前五项数据写入进存储器中，在此处也有一个数据处理的小方法，因为存储器的一个地址只能存储一个字节的数据，所以存储的数据不能超过255，但是得分是四位数的数据，所以需要把数据拆分进行存储，同时为了减少程序的data，这里只需要把数据按千位百位为一个数据，十位个位为一个数据进行存储就好了。

所以相应的，从eeprom存储器读取排行榜数据的时候也是相同的，两个地址构成一个分数。

## 亮点设计

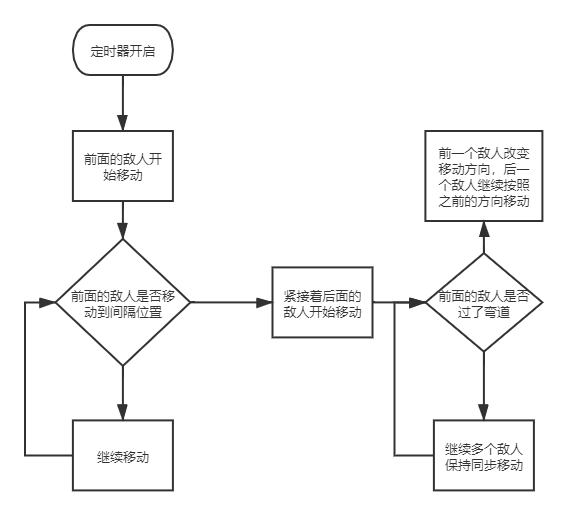
### 较好的程序衔接性

在此次项目中有很多需要死循环才能操作的地方，例如：根据遥控器接收到的信号对游戏进行操作，游戏结束之后数码管的显示等；

而为了实现进入之后能跳出来，就有点像定时器的中断，所以我们也应该像定时器的中断标志一样设置一些标志变量用来判断是否该跳出死循环。

### 敌人的成堆分批出现

在此次项目中，一些关卡图中，敌人是需要曲折的前进的，并且一批出来的敌人并不是一个，所以为了实现这样的移动，我在程序中通过判断前面敌人是否到达指定的位置，然后再根据敌人之间的空隙进行条件判断，跟在第一个敌人后面那个敌人是否应该继续按照前一条的语句进行移动，从而达到了成堆，曲折前进。



**图 11 敌人移动流程图**

### 5.3 LM4229屏幕的使用

在这次的项目中，我认为我LM4229这个硬件的使用也能算的上一个亮点。

作为这次项目中的核心，屏幕的选取是十分重要的，应该选取合适大小的屏幕给予用户良好的游戏体验，这次的项目是使用proteus仿真软件进行的，LM4229也是proteus元件库里最大的LCD显示屏，同时它240\*128的大小尺寸，也是符合横屏游戏机的。

对于这块LCD液晶屏，在详细设计中也有说过，和我们平常学习的LCD屏幕的驱动芯片是不一样的，并且我上网查找资料，搜索其他人使用的例子的时候，也少有见到，为数不多的也只有十多年前一些其他搞笑学生的课程设计而已，最后我是通过去查阅该款芯片的数据手册来学习使用这块屏幕的，所以项目前期的时候在这里也用了较多的时间。

### 5.4 断电存储器的使用

项目中使用了断电存储器，从而能够实现游戏的排行榜功能可供玩家不断的挑战自我

## 系统兼容性说明

软件程序系统兼容性良好，可以在不同版本的计算机中运行，具有向上兼容性和向下兼容性。

同时软件的编写也可以通过许多不同的编译器对其进行修改。

## 总结体会

### 项目总结

由于对于单片机知识学习的时间较少，所以在本次项目的前期学习阶段花费了较多的时间去学习掌握单片机硬件使用，以及各个核心模块的功能实现，以至于不能像之前的项目一样，去给自己的程序添加更多的拓展功能，这是十分可惜的。但是，我认为每个人的学习始终是要有这样的摸索过程的，就像这次的项目中，12天才完成了第一关，后面的关卡只能使用非常短的时间去完成，不过，没有之前的12天的话可能我也不能有后面的发展吧，希望这次的摸索过程也能成为我以后路上的垫脚石。

### 项目体会

这次的项目是QG考核最后一次的考核题目了，进入QG的训练营加考核以来历时两个月，从最开始招新面试前的两个星期的打印“hello world”，到做出笔记管理系统，从点亮第一颗LED，到做出塔防游戏。这无疑是我的成长。

每次的考核，无论是图书管理系统，笔记管理系统，塔防游戏，对我来说好像都是不可能完成的项目，每次似乎都是涉及了不同的知识。就像图书管理系统考察了对文件的读取，对基本数据结构的运用，笔记管理系统考察了对文件的操作，对复杂数据结构的处理。之前的项目会有大部分自己没接触过的知识点，同样的这次的项目也是从0开始的小白开始的一个项目，项目开始之前我不知道说明叫做LCD，不知道什么叫做双机通信，也不知道单片机原来还可以断电存储。

但是，每天晚上看着书，跟着书上敲上代码，去找其他人的实验，理解里面运用的方法，从中学习，运用到自己的项目中。遇到不同的知识点，查找硬件的数据手册，去论坛，去问淘宝客服，去问素不相识的路人，似乎只要看到了我需要的知识，就会去询问，了解之后再自己创造。一点一点的积累，我也勉强在这十四天中做出了这次的项目，虽然还有很多不成熟的地方，不够完善的机制，没有拓展的区域，可是我也认为这比起十四天前的我也算好了，毕竟我学会了很多了。

这次项目中是历时最长的，所需学习知识最多的一次项目，我有体会过不知从何下手的迷茫，不知道哪里错了的焦虑。我也想过是不是应该放弃，但是我还是走下去了，因为看着每次一个模块程序运行起来的喜悦，修改了一个BUG的快乐，看着自己的作品慢慢的能运作起来，真的每一步都是对我极大的鼓舞。

最后，考核期就要结束了，无论结果如何，我相信这一次会是我一次难忘的经历，人生中再一次全力以赴，抛开游戏，抛开娱乐，挤出更多的时间去学习考核知识，这一切的一切都培养了我学习的习惯，同时也在考核中收获了一些优秀的竞争对手，收获了可以一起交流的小伙伴！