**实验报告八 创新实验**

一、实验目的：

巩固本学期所学到的内容，尽量用上更多所学过的硬件。并且进一步探索某些硬件的可能性，例如用蜂鸣器实现不同频率的声音形成音乐，以及让LCD图案变化有类似动画的效果。

二、预期效果：

本次实验我们设计做一个，通过按下键盘不同按键后让蜂鸣器发出不同音调的声音，实现乐器演奏功能，以及按下部分键后会播放已设定的音乐，实现音乐播放功能。同时搭配上一个电子乐谱，可以实时显示当前发声的音符。

三、使用的硬件设备及连接方法

使用的设备有：

LCD液晶显示屏，CH451键盘及数码管，蜂鸣器，开关

连线为：

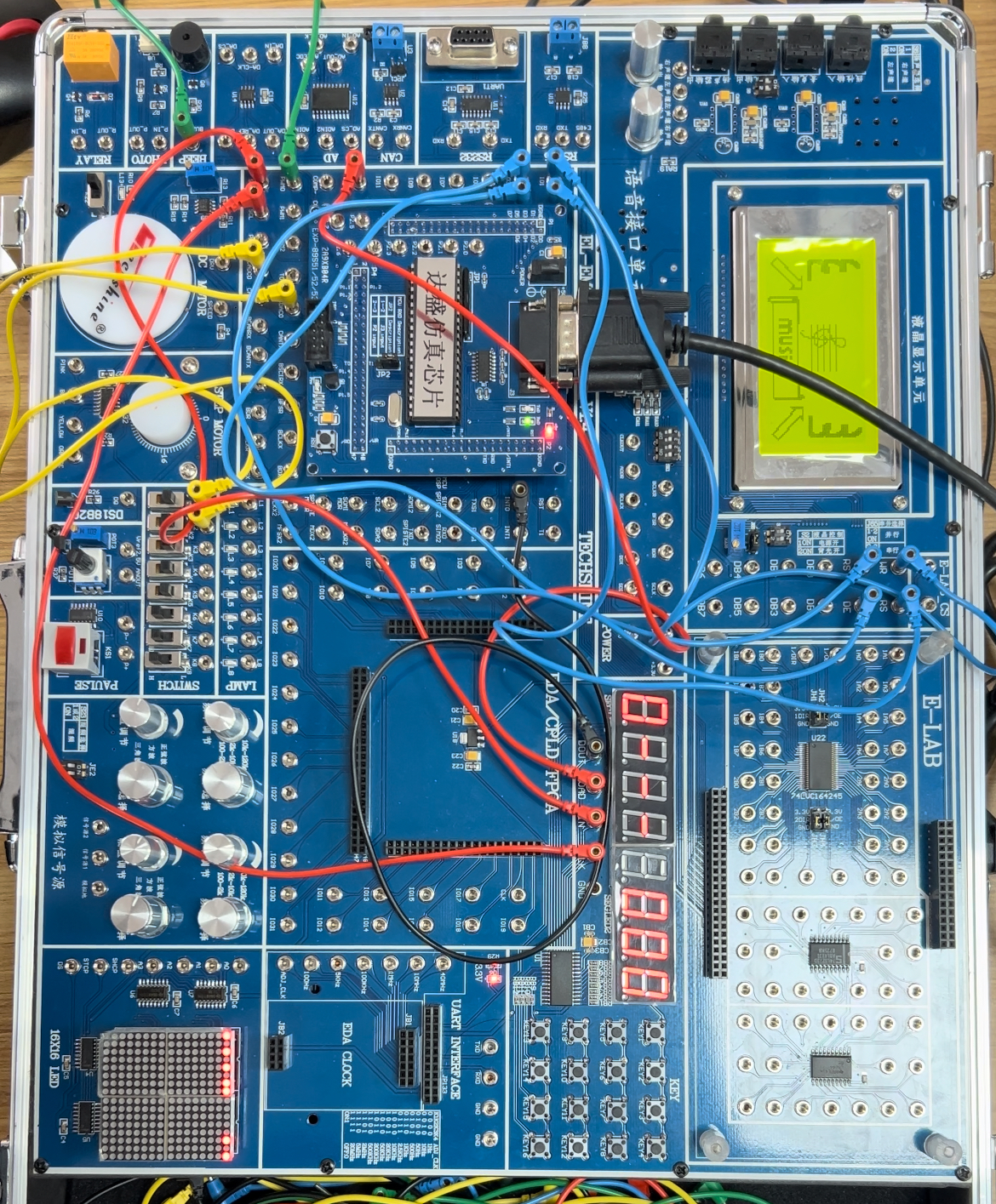
IO1、IO2、IO3、IO4分别连接WR、RS、RST、DE

PWM0连接BEEP

CAP0、SDA、SCL分别连接LOAD、DIN、SLCK

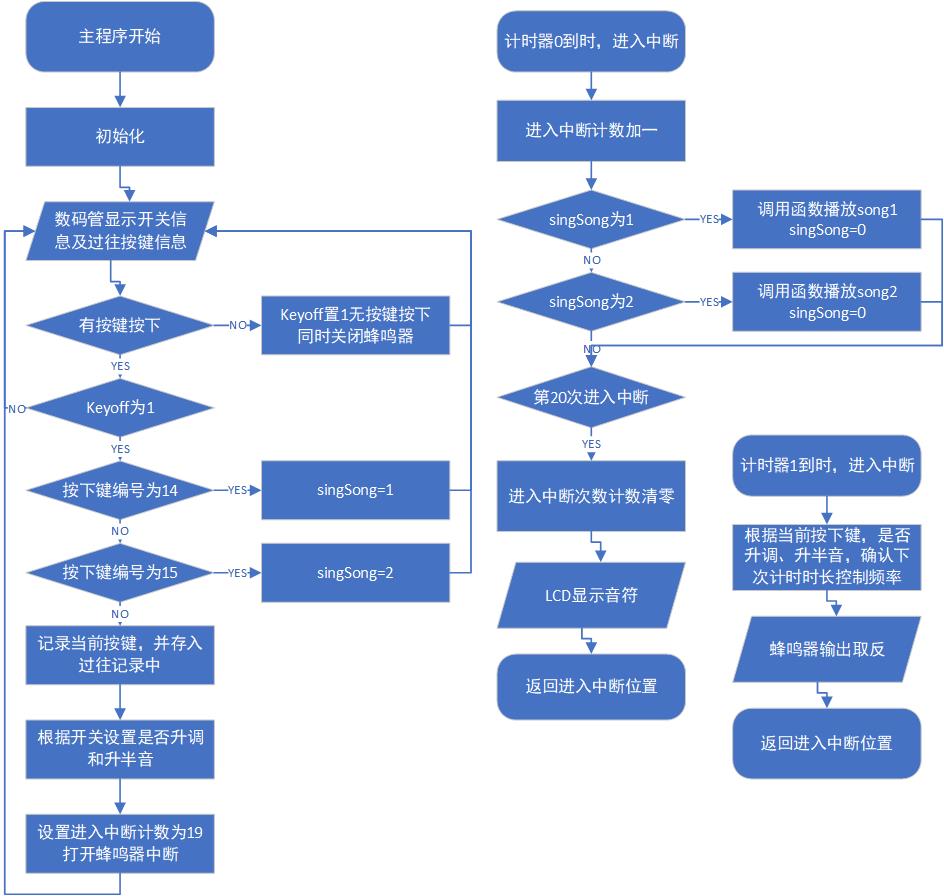
DOUT连接INT0

CADC0、CDAC0分别连接k1、k2



四、程序分析

1、流程图的形式给出程序设计的思路。



2、代码及必要注释：

由于全部代码较长，请移步至附件“code8.c”进行查阅完整代码。

以下是部分主要函数的展示：

主函数及中断：



五、问题分析

1、实验过程中遇到的问题及解决方法。

（1）程序中常量太多，数据区太大，TARGRT NOT CREATE

由于本次试验我们用的lcd显示有动画效果，需要保存很多画面，以及有几首歌需要内置保存，因此需要很多数据。

解决方法：首先是把这些数据改为常量，并且在声明时加上code，这样子可以让这些常量存在程序存储器中而不是数据存储器，这样可以存更多数据。其次我们在编译器中将程序大小改为大，这样就可以正常运行了。

（2）画好的图案显示出来却是乱码

解决方法：首先要弄明白是按行输出还是按列输出，并且要设置正确的控制字，让lcd进入图像显示模式，这样lcd才可以正常输出。

（3）无法将图案输出到指定位置，有些图案无法对齐

解决方法：将显示屏区域进行划分，并且将显示到指定区域集成到函数里面。对齐则需要弄清楚显示原理，手动对齐。

1. 蜂鸣器会因进入定时器0的中断而被打断发声（定时器0中显示lcd需要至少50ms时间）

解决方法：设置中断优先级，使定时器1即蜂鸣器所用定时器优先级最高，从而使得不会被打断，稳定发声。

1. ：（在没有设置程序的size前遇到的问题）将lcd显示程序放置在主函数中会导致占用过大无法编译。

解决方法：将显示程序放置于定时器中断中。同时，因为显示程序需要占用一定时间，故每1s调用一次，防止不断调用显示而导致主程序被阻塞。

而在按下按键后需立即刷新显示，使用改变T0count（进入中断次数计数，达到20为过了1s）为19的方式，使得下一次进入显示中断立刻调用一次显示。

1. ：原先将显示程序放于定时器中断0中，而唱歌的程序放置于主程序中，因每1s会调用一次lcd显示会占用些许时间，使得主程序中的演唱可能会产生问题，某个音被延长些许。

解决方法：由于播放时恰不允许进行其他操作，故将演唱程序放置于定时器0中断中，与显示不并行。同时，为在演唱时即时显示所输出字符，演奏程序中在每个音弹奏时进行lcd显示，而每个音音长函数的设置，还要根据显示时间进行调整。

2、实验的收获或感想。（可选）

本次试验自由发挥，一开始我们还在纠结到底做什么，本来想的是利用16X16LED来做一个贪吃蛇小游戏，原本我们都开始编写了，后来由于感觉做得相似的人比较多，就决定换一个项目。我们也是由于之前是用蜂鸣器那个实验，我们蜂鸣器必须通过一定频率的高电平信号才能发声给我们带来的灵感。正好我和我的搭档这学期还报名了钢琴的选修课，对音乐有一定的了解和基础，于是我们打算制作一个迷你版的钢琴，我们有这个想法之后一拍即合。我继续负责我之前只做过的lcd显示部分，不过这次要加点难度了。我不仅要实现图案的显示还需要对键盘按下的键作出反应，有一定的动画效果。

起初把我们难倒许久的就是内存储存的问题，我储存多个画面的图片，他储存音乐都需要不少的空间，我们也是通过上网查询，和向同学询问，找到了解决方法。这次的函数其实和之前文字显示的有不少不同，图案输出和文字输出不一样，有不少函数都是直接重写了的。也不是每次写完都很成功直接能跑起来，也是经历了很多轮的调试，经历的问题有爆内存，如果数据缺失有的显示位置没有给出显示内容可能产生闪烁花屏之类的问题。

这次实验我们在最后一节课就开始做了，节课之后又做了两个下午，最后完成了，总体来说效果达到了预期，看着这个迷你版的小电子钢琴包括内置音乐和显示的乐谱，看起来还蛮精致的，真的感觉这学期单片机学有所成，非常有成就感！

之前的实验六中，蜂鸣器不讲武德卡了我们好一段时间，这一次为报这一箭之仇，同时也因为我们两个确实对于音乐比较感兴趣，我们选择了这个项目。

不同的音调是对应有确切的声音频率的，根据网络搜索得知各个声音的频率，又知道了我们设备的时钟频率，我就根据公式计算出了计时器所要设置的计数初值。

（公式为：65536-(1/2f)/(11/11.0592MHZ) 注意后面为MHZ）

为了满足更复杂的演奏需求，加入了一个开关控制将所有音升半音，这个则需要一些乐理知识（知道对应的哪个就行）。

开始的时候我们选择的调似乎太低了一些，而且仅有14个音，即两个八度，所以后续又加了个开关可以控制将所有音升高两个八度，所以共计4个8度及相应的半音。

当蜂鸣器确乎可以根据我们按下的声音发出不同声调的声音时，还是挺令人兴奋的，蜂鸣器这略带沙哑模糊的电子音让我想到了小时候玩计算器，用计算器弹东西的那种感觉。

16个按键，14个是音，剩下两个内置了两首歌。我编写了个sing函数用于唱出音的音高和音长，简化了后续填入歌曲时的代码，这个让导入一首歌变得其实还是蛮容易的，只需在开头设置默认音高，已经歌曲的速度，中间遇到特殊的音，特殊段落也可以更改，然后按着谱子sing（）函数sing出来即可。原本内置的两首歌为《卡农》和《小星星》，后续将《小星星》改为了室友宋羽豪爱的《漠河舞厅》。

整合我和冯哲熙的代码倒是花了不少时间，因为一开始我们并未设置程序size为large，所以在将他的显示函数扔到程序中时时不时有各种过大TARGET NOT CREATE报错，这个就需要一点点去磨合程序，修改细节，部分遇到的问题在问题分析中有阐述。修修改改，最终让所有函数归于正轨，程序正常运行还是挺有成就感的。