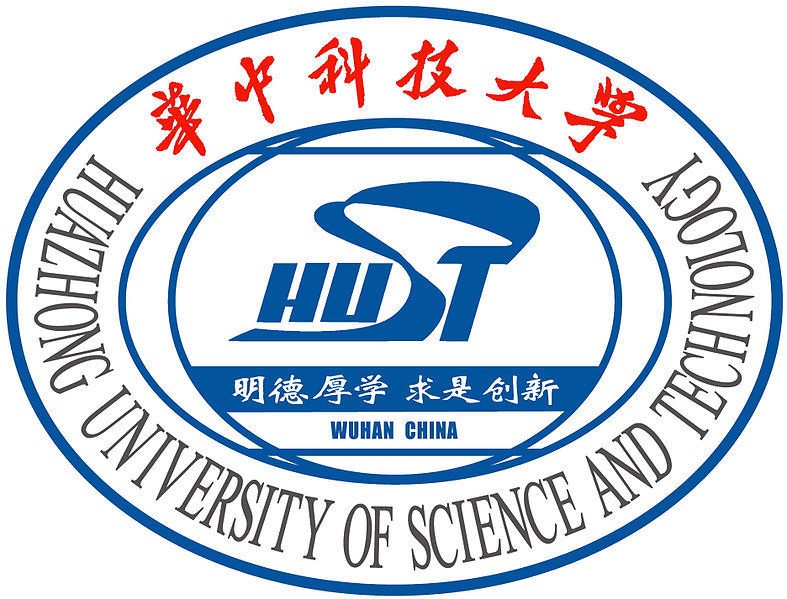
2019年华中科技大学电气与电子工程学院

第三届“启亦电子杯”创客竞赛

**作品设计报告**

**Final Report**



作品题目：MIA(弥雅)-乐器辅助学习工具

团队名称: 大碗热干面

团队成员: 谭越瑾

叶泽模

**2019年华中科技大学电气与电子工程学院**

**第三届“启亦电子杯”创客竞赛**

**参赛作品原创性声明**

本人郑重声明：所呈交的参赛作品报告，是本人和队友独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果，不侵犯任何第三方的知识产权或其他权利。本人完全意识到本声明的法律结果由本人承担。

参赛队员签名：

日期： 2019 年 9 月 5 日

MIA 结题报告

摘要：

随着人们生活水平的提高，越来越多的人更加注重精神需求的满足，这样也更有利于人们的健康发展。例如，有人就会选择学习一门乐器，来丰富业余生活，满足精神需求。由此，本小组决定设计一个比较完善的多功能的乐器辅助硬件，帮助初学者更快入门.它有完善的人机交互界面,有一个统一组织的“准系统”，能够帮助乐器初学者调动学习欲望，以免出现疲惫、倦怠、灰心等心理状态，从而为乐器初学者提供较为高效的学习状态。该作品具有以下功能：密码开机功能，菜单选择功能，游戏功能，单音练习功能，节奏练习功能，MIA-LINK功能以及文件系统功能。

关键词：乐器，辅助学习，硬件系统，创客

MIA CONCLUDING REPORT

ABSTRACT

With the improvement of people's living standards, more and more people pay more attention to the satisfaction of spiritual needs, which is more conducive to people's healthy development. For example, someone will choose to learn an instrument to enrich their spare time and meet their spiritual needs. Therefore, the team decided to design a more comprehensive and versatile instrument assisting hardware to help beginners get started faster. It has a complete human-computer interaction interface and a unified “quasi-system” that can help the beginners of the instrument to move. Learning desires, so as to avoid fatigue, burnout, discouragement and other psychological states, thus providing a more efficient learning state for beginners of musical instruments. The work has the following functions: password boot function, menu selection function, game function, mono practice function, rhythm practice function, MIA-LINK function and file system function.

Key words: musical instruments, assisted learning, hardware systems, makers

目录

# 第1章   调查/设计目标/系统框图结构/功能

## 1.1前期调查

通过一定的交流调查以及上网搜索相关资料，我们发现当今人门普遍愿意学习至少一门乐器.学习音乐有诸多好处,在《音乐学科核心素养与初中音乐教育研究》一文中提到“音乐对学生的记忆力的增强有很大的帮助，学习音乐的学生需要欣赏和记忆乐谱，在这样直接和间接的活动中，大大地增强了学生的记忆力".《高中音乐教育中器乐教学的必要性与教学方法研究》中也提到“没有器乐教学的高中音乐教育是不完整的，不但违背了音乐艺 术教育的本质，同时也未能满足高中阶段学生情感需求、艺术修养的满足与提升”.同时音乐能丰富我们的生活,提升公民文化素养,为建设社会主义文化强国做出贡献.

然而,乐器的学习并非一件简单轻松的事.

有文章指出,“在练琴的过程中不免出现疲惫、倦怠、灰心等心理状态, 这是由于在长期的习惯中, 一成不变、按部就班成为了学琴过程中的一大障碍”.乐器练习时的单调无聊劝退了一大波充满热情的热血青年.如何破局呢?

《小学音乐课堂乐器教学实践研究与思考》一文中提到“兴趣是学生最好的老师，在小学音乐课堂教学中，教师开展 乐器教学活动时，只有全面调动学生的学习欲望，才会让学生更加 积极的对乐器知识进行探索。在课堂上，小学音乐教师要引导学生 时刻保持着对乐器的好奇心，在此基础上引导进行乐器练习，可以 获得良好的效果”。

《小学音乐课堂乐器教学实践研究与思考》中提到“新时期下，随着社会经济的快速发展，乐器已经进入到众多家庭中，人们对于乐器的接受度也越来越高”。同时结合本小组调研情况，乐器辅助工具可能有相当的潜在市场前景。

因此我们小组决定以"快乐练习"为主题，设计出一款以激发兴趣为目标导向的,适合乐器初学者的乐器辅助学习工具，让他们更好地踏入乐器学习的正轨。

## 1.2设计目标

本小组旨在设计一款乐器辅助学习工具，实现知识的实用化转换，产生一定的社会价值。通过这一过程，加强并提高我们相应的知识水平，同时发挥积极的社会意义，对乐器初学者能起到积极的影响，由此呼吁更多愿意学乐器的人加入到学习乐器的旅途中。

另一方面，此产品针对乐器初学者这一特殊人群，理应达到让初学者愿意持之以恒地学习这一结果，同时也期待产品的功能丰富多彩，支持各式各样的乐器，符合我们最初的意愿.

## 1.3方案设计与功能设想

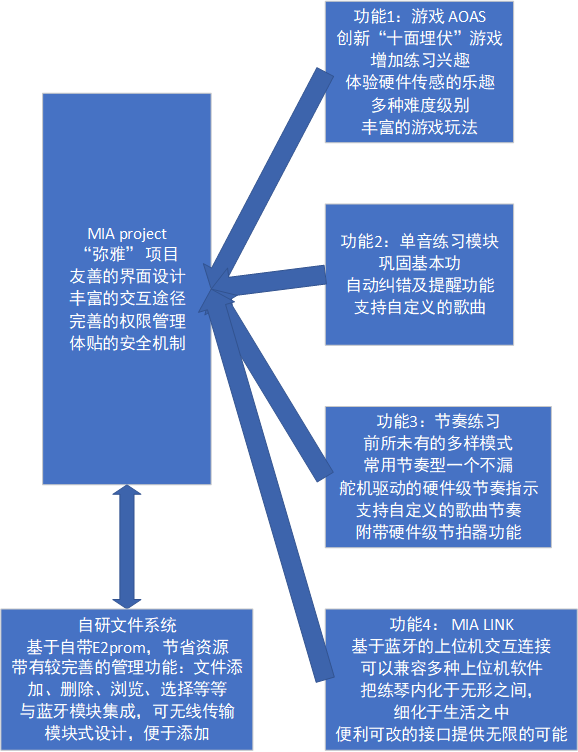


图1 方案设计与功能设想图

# 第2章    方案综述与总体思路

## 2.1总体思路:



图2 硬件结构图

根据计划,这是一个比较完善的多功能的乐器辅助硬件.它有完善的人机交互界面,有一个统一组织的"准系统",可以想见这是一个庞大的工程,需要紧密的团队合作才能在可接受的时间内完成.为了便于团队并行編程,我们将整个项目拆分为十余模块,事前不惜时间细细商讨模块功能及模块间接口,并规划出硬件的大概样貌,制定了协同编程规范(详见见附录文件).然后分工合作完成各模块的编写,每编写完成一个模块后,进行简单验证.最后确定最终硬件,进行联合软件调试,再整机包装,项目原型机基本完成.

## 2.2以文件为单位的模块说明:

### 2.2.1 MIA.inoMIA主函数

实现功能:

登陆管理:允许授权用户登录,保护数据安全

权限管理:超级用户可进行修改密码和初始化工作

菜单选择:按钮和旋钮两种方式联合进行,选择后进入相应模块

背景音乐:循环播放,改善用户体验,丰富文化内涵

程序选择性说明:

对象集中设计:

          所有对象的构造和初始化均在此文件中,便于移植与统一管理

目录的实现:

通过变量记录当前选择项,循环动态检测按钮情况,更新按键值并刷新屏幕

        程序并行性:

菜单中要同时驱动两个屏幕,一个数码管,同时要实时响应按键变化,同时还要不间断的输出背景音乐和控制舵机旋转以指示角度.并行项目很多.程序经过精心设计,构造出一个小巧的循环,高速响应各种输入,对任务执行顺序进行合理的切换,使其达到微观串行,宏观并行的效果.

### 2.2.2 func0.cpp 游戏模块

实现功能:

生成与显示：支持三种不同的敌人的生成与显示，让游戏更加富有变化

最高分纪录：两种难度独立统计最高分

准星的移动：实现准星的实时移动，更好的交互式体验

大招的累计：一局游戏中，分数每增加十，便累计一次可消灭当前所有存活敌人的 大招

程序选择性说明:

游戏流畅性的保障：由于要在游戏中保证对玩家的操作的及时响应,同时要保证各项及时的相对精确,我们对游戏主循环进行了精心的设计,使其尽量短小精悍,能在较短时间内完成一次刷新.具体措施如:减少不必要的显示刷屏--仅当屏幕发生较大变化时(如敌人状态更新或准星移动超过阈值时才进行刷屏(50ms);对FFT检测时机优化,当声音太小时跳过本轮的FFT检测(120ms);压缩按键响应时的延迟(5ms);对2004刷新时采用部分刷新的做法(2ms)...

### 2.2.3 func1.cpp 单音模块

实现功能:

单音的识别判断

错误提醒

支持自定义歌曲

支持尤克里里\钢琴等多种乐器(后续可能进行更新支持更多乐器)

练习统计功能

程序选择性说明:

精确的时间把控:由于FFT函数需要用大量时间进行计算(约120毫秒) ,因此程序采用内部定时器进行精确的中断计时,以获取精确的时间,把握音乐节奏.

### 2.2.4 func2.cpp 节奏模块

实现功能:

尤克里里专属节奏练习

四种传统通用节奏练习（还能持续添加）

自定义歌曲节奏练习

节拍器的辅助练习

程序选择性说明:

该模块将节拍器与蜂鸣器融为一体，通过内部定时器的精确时间补偿，使得节拍器与蜂鸣器同步工作，保障音乐节奏的稳定性

### 2.2.5 func3.cpp MIA-LINK模块

实现功能:

自带游戏总会让人感到无聊?不满足于2004\12864亮瞎眼的显示质量?想体验更加广大的世界?

MIA-LINK 利用内置的蓝牙模块可实现核心板与电脑,手机等带有丰富功能的上位机实现连接,以实现多种多样的丰富拓展:

如:

利用琴声与浏览器进行交互,实现以琴为板,上网冲浪.

通过对按键映射的客制化适配,多种游戏将可通过对琴的操作来参与,全新体验!

程序选择性说明:

利用成熟串口透传通信,内容是自创的编码方式,以减少开销,提高研发速度.同时软件支持后续升级,提供更多功能!

### 2.2.6 func4.cpp 文件模块

实现功能:

基于自带E2prom，节省资源.

带有较完善的管理功能：文件添加、删除、浏览、选择等等...

程序选择性说明:

与蓝牙模块集成，可实现无线传输文件.

模块式设计，便于添加到其他功能中.

**以下是程序模块的说明,只重点介绍各模块实现的功能,具体实现原理见附录源代码.**

### 2.2.7 button.cpp / button.h 按键模块

实现功能:

四个按键引脚的初始化

一次性返回四个按键的状态

程序选择性说明:

通过返回一个八位二进制数，后四位分别代表四个按钮的状态，由此可以只用一次防抖延时便获取四个按钮状态。

### 2.2.8 eepromMgr.cpp / eepromMgr.h 存储器硬件抽象接口层

实现功能:

对内置的EEprom进行稍微的简化包装,提供一系列接口便于调用.

程序选择性说明:

关于各个函数的功能请见头文件注释.

### 2.2.9 fft\_c.cpp / fft\_c.h  FFT 算法包装,简化接口

实现功能:

对开源FFT实现代码进行了二次包装,删减了部分代码,留下对本项目有用的功能.

程序选择性说明:

详见arduinoFFT文件(几乎未改),本项目的 fft\_c.cpp / fft\_c.h 只进行了简单包装.

### 2.2.10 led.cpp / led.h led 封装模块

实现功能:

0到9数字显示

节奏舞动

跑马灯模式

位控led模式

程序选择性说明:

用八位二进制数分别控制数码管上的八盏灯，通过循环、判断等方式较为简便的实现各种功能。

### 2.2.11 miaTool.cpp / miaTool.h 可复用的杂项辅助函数集中

实现功能:

部分引脚的集中定义

部分可重用数据表的集中

随机歌曲填充函数

嵌入型文件浏览器

蓝牙包装类主菜单

Bgm实现模块的包装类

超级用户模式实现函数

程序选择性说明:

bgm包装类实现费了些功夫,勉强实现了主菜单下背景音乐的并行播放,但仍然些许地受菜单切换时更新显示器的长耗时的影响.

### 2.2.12 sensor.cpp / sensor.h 声音传感器包装函数

实现功能:

判断声音的有无（一定范围）

初始化音频引脚

程序选择性说明:

十分简洁，有需求请参照附录

### 2.2.13 showu8g2.cpp / showu8g2.h

实现功能:

各式封面图的封装

各种音符的显示

敌人、准星的显示

程序选择性说明:

封面图主要是将图片切换大小端后进行二值化处理，其它图形的显示主要是合理运用u8g2中的各类函数，合理做出合适的图形。

### 2.2.14 其他文件:

为来自互联网或arduino 开发环境的开源资源内容,是各个文件模块的支持模块或基础接口(还有一些是由于编译需要拷贝而来arduino IDE 中的文件--因为本项目是在Visual Studio Code下完成的,并非官方IDE)

# 第3章    方案的迭代改进

这里只提到进行过重大变更的方案,简略记述其改进缘由和改进结果,仅供参考.

## 3.1 MIA.inoMIA主函数:

最初想法：简单开机密码，用按键控制功能选择

改进结果：新增管理员模式，可以修改密码，新增旋钮选择功能，新增背景音乐

比较：改进后安全性有保障，用户操作更方便，交互性更好

## func0.cpp 游戏模块：

最初想法：128×64的显示屏分均为四个部分，矩形代表敌人，圆形代表准星。敌人与准星只出现在四个固定区域的固定位置。每击杀10位敌人获得一次大招。

改进结果：增加敌人的种类，增加准星的可移动性（由固定的四个位置变为全屏移动），增加游戏暂停功能，增加准星初始化调整功能，增加灵敏度（控制准星移动快慢）调节功能。准星与敌人图像重新制作。

比较：改进后，准星更像准星，敌人更像敌人，游戏富有变化，交互更加通畅，难度更加科学，游戏更有体验。

## 3.3 func1.cpp 单音模块:

改进：新增统计功能;加入对钢琴的支持(原来只适用于尤克里里)

比较：能记录练习时间和练习次数,加强成就感,给予积极地正反馈;丰富了所支持的乐器种类,若有需要,日后还能对多种乐器定制界面,实现支持,真正实现宽功能的MIA!

## 3.4 func2.cpp 节奏模块：

最初想法：通过超声波传感器辅助判断正误，有两种节奏模式

改进结果：去除超声波模块，新增节拍器、尤克里里专属、传统节奏、自定义歌曲节奏练习等功能

比较：超声波传感器捕捉运动的、较小的物体的能力较差，所以就没有安排超声波模块登场。新增一系列功能，简言之就是提供多种选择，让用户有更好的节奏练习体验。

## 3.5 func3.cpp MIA-LINK模块:

最初(立项答辩时)没有此功能;中期答辩时提出能和电脑进行简单的交互,借助特殊软件(chromium)完成"弹琴上网"功能;随后提出全面兼容的设想--通过特异性的适配程序将(一门乐器+MIA)变成一个多功能外设,可以有办公,休闲,游戏等多种使用方式,将乐器练习归化为在平日生活中的一个小习惯.至此,MIA不在局限于一个小小的8位单片机,它可以作为重要的一份子参与多种事务,拥有广阔的拓展空间与无限可能,越来越多的脑洞大开的练琴方式等着你的发掘!

## 3.6 func4.cpp / EEpromMgr.cpp / EEpromMgr.h 文件模块：

最初想法：采用链表的方法进行管理,每个文件就是一个链结,维护一个总表,便于快速找到并维护每个文件.

改进：难度太大,以现有水平,虽历经很长时间的编程与找BUG,但仍有相当部分功能无法调通(也有arduino环境的调试功能太过孱弱的原因),故只得放弃该方案.后重写所有有关模块,采用最暴力的方法--从头到尾一个文件接一个文件地存储,并从头到尾一个字节一个字节地查找.虽然效率很低,但应付20首歌不成问题.(4kb的EEprom大概就这么多歌)

此项难以说成是一种改进,但毕竟是从无到有地实现了文件管理这一功能.

## 3.7 led.cpp / led.h led封装模块:

最初想法：采用10×8的大表存储0到9十个数字对应led灯的明暗，代码部分冗 长。

改进结果：将10×8后的8改为八位二进制，利用循环、判断等条件进一步完善代码。

比较：改进后节约空间，便于检验

# 第4章 方案测试与测试结果

测试方案为对所有模块功能逐一进行验证,各项测试基本通过.

## 4.1 测试模块：MIA.inoMIA主函数:

测试结果：





组图3 MIA主函数测试图

## 4.2 测试模块：func0.cpp 游戏模快

测试结果：

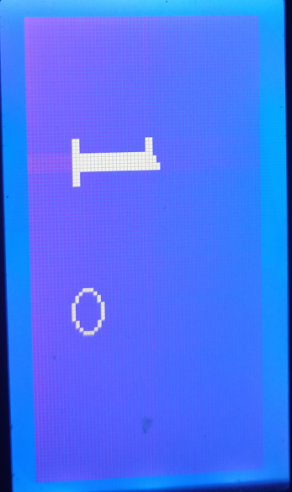


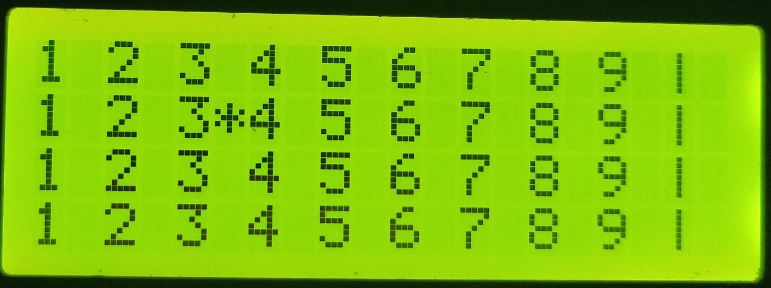


组图4 游戏模块测试图

## 4.3 测试模块：func1.cpp 单音模块

测试结果：



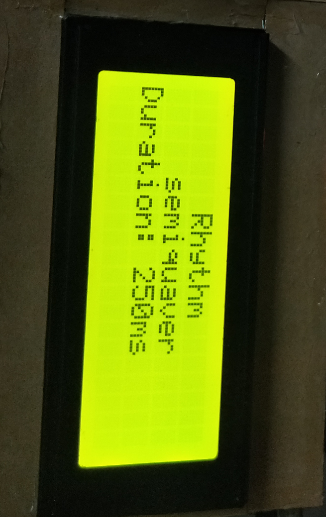


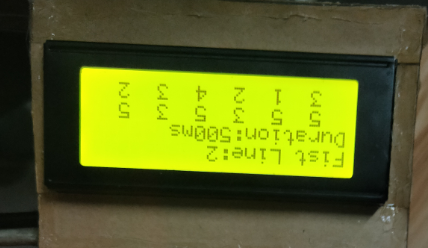
组图5 单音模块测试图

## 4.4 测试模块：func2.cpp 节奏模块：

测试结果：







组图6 节奏模块测试图

## 4.5 测试模块：func3.cpp MIA-LINK模块

测试结果：

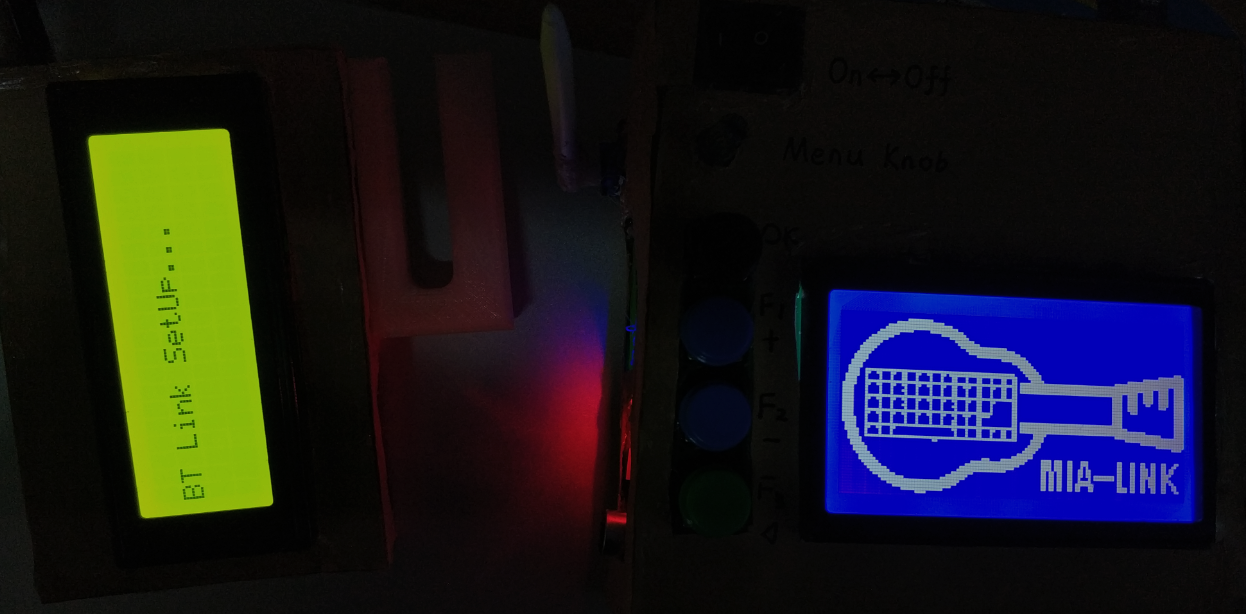
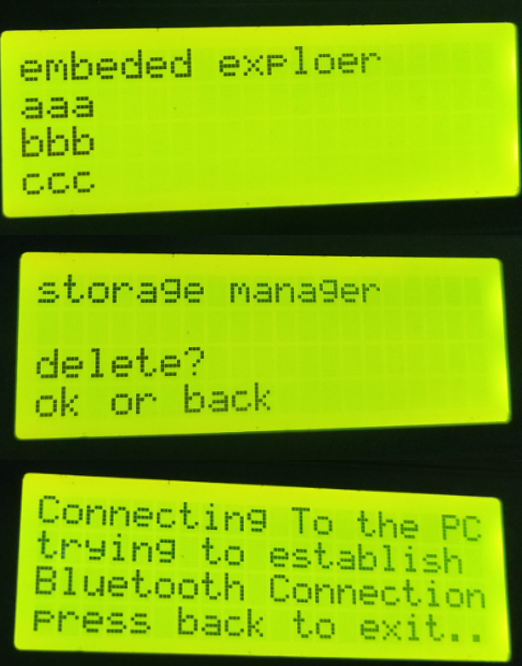


图7 MIA-LINK模块测试图

## 4.6 测试模块：func4.cpp / EEpromMgr.cpp / EEpromMgr.h 文件模块：

测试结果：



组图8 文件模块测试图

# 第5章 自我评价/未来规划

## 5.1 自评（谭越瑾）：

这个项目算是第一次正式系统地将所学知识用于实践项目，该项目总体说来还算是差强人意。

首先，我深刻地意识到课本上学习的知识还太少，就算完成我们这样一个简单的项目也不够，如何实现团队合作的编程？如何定义软件接口？如何确定软件结构？如何实现任务的并行化？课本上只交给我们基本的语法，只给我们基本的语言工具，至于如何使用好这一工具，使其发挥出自己的威力？还需要我们大量的课外学习与拓展。

其次，创新不仅仅需要灵感，更需要对生活的细腻体悟。比如说在创客项目中，要细心地观察体悟生活，才能发现生活中的不便之处和“痛点”，找到硬需求而非伪需求，并能找出代价可以接受的解决方案，这才是创客项目的基石。一个好项目不仅要看其解决的问题是不是重要且富有意义,也要考虑其解决问题的方案和代价.

当然，创新也有一定的方法可循，比如，我们组就运用了奥斯本检核表发对项目进行检核，提出了很多修改方案--进行扩大，添加新功能；进行缩小，删减不必要的鸡肋模块和传感器。可以说,前期知识储备给了我们具有相当意义的指导.

对于大一点的项目来说,团队合作就相当重要了.众人拾柴火焰高,倘若只靠一个人,面对这浩大的工程量,这实现代价也是难以想象的.当然,三个和尚没水喝,人多更要求实现良好的合作和管理,这样才能更好地发挥每个人的价值.不言而喻,这个项目对我而言具有相当大的挑战性,不知多少次陷入尝试DE一个BUG,但此BUG不消反而出现一堆新BUG的无助,疑惑与沮丧,但我在这当中学会更好地管理情绪,与人沟通,合作交流,以促进项目更好更快地完成.

当一个项目要实现诸多比较复杂的功能时,对底层的要求就会越来越高,而这一系列操作就会逐渐超出我们的知识范围--例如,可以尝试使用中断和定时器以改进程序性能吗?中断程序会影响其他部分吗?如何在有限的硬件接口上实现与多个外设的通信?TWI协议USART协议SPI协议有什么区别?如何合理使用?了解这些东西颇费功夫,希望能在课程中得到体现.毕竟,玩单片机其实就是玩外设,而与外设交互的核心就是各种通信协议.

其次,我认为课程的培训时间可以稍稍延长,增加一些内容.比如说上面的常见通讯协议的介绍和选择,还有蓝牙模块或网络模块的使用--这样可以教给我们使用大型模块的经验,还有中断与定时--中断可以说是微处理器里面一个里程碑级的设计,稍加使用可大大简化部分程序;还有一些有关AVR单片机的基本内容,比如:对其中数据类型的说明,输入输出模式的简介等等,毕竟他有一个INPUT\_PULLUP,这样接开关时可以省去不必要的上拉电阻,可以简化电路.

这种项目是没有上限的,我认为稍稍延后至大二可能可以给同学们更多的发挥空间(那时才会学完数电模电).

## 5.2 自评（叶泽模）：

刚刚接触这个项目的时候，我就感觉这个项目有些难度，也许很难实现既定功能。但通过半个多月的努力，还算是可以勉强完成，同时也感觉收获不少。

首先是关于团队合作的认识，这个项目涉及好几个方面，如果团队成员之间没有一个良好的沟通，是难以将这个项目做下去的。同样，只有团队成员都遵守团队最初定下的规章，才能更有效地将进度往前推。在这个时间紧迫、难度不小的情况下，团队合作就十分重要了。团队成员一起做项目，不仅可以起到很好的督促作用，也能让自己理解团队做项目时的运作方式。良好的项目管理当然也就是其中较为重要的一部分。

然后是关于创新的理解，创新还是源于生活，源于某个灵光一现的想法，当我们有想法将它付诸于实践时，创新可能就来了。但仅仅只有好点子还不够，一个创新性的项目往往更加困难，一定的时间精力的投入才有可能化空想为实物。因此，相对应的能力，或者一定量的知识储备在实现创新想法的过程中是不可或缺的。这些林林总总的因素加和到一起也许才能更好的发挥创新的价值。

最后是关于此次创客的想法，创客是一个很好的平台，我们可以从中学到一些有趣的知识，可以自己动手做出某个玩意，可以在实践中摸索学习、不断进步。此次创客竞赛也让我受益匪浅，我至少明白了团队合作与项目管理的重要性，知道了创新想法实施的不易，在这过程中，自己的相关知识水平当然也得到一定的提高。总之，此次确实收获颇多，同时我也建议合理根据课程难度安排课程进度，例如刚开始可以轻松一点，中期多一点，后面宽松一些。

## 5.3 未来规划:

* 重构软件结构.多读书,学习别人的代码,改进现有混乱的结构,可参"设计模式",将程序改为完全面向对象的可扩充的样子.充分利用arduino支持部分C++的特性.
* 加入新的模块.利用EEProm存歌只是实现原型机的缓兵之计,日后可加入TF卡模块实现对FAT文件系统的支持,提高硬件系统的兼容性.加入MP3硬解码模块,改善系统音质,提高用户体验.加入FTF屏幕模块,真正实现高分辨率的全彩显示,震撼影音!
* 更换平台.不同于其他项目,本项目不是单单实现一个简单的功能,正向嵌入式系统方向发展,如果可能的话,可转向树莓派平台或传说中的STM32F4平台,这才能真正实现所有功能...
* 扩充MIA-LINK功能，设计出更友好的桌面应用程序的图形界面，同时加入重力传感器的控制作用，以支持更多、更广泛的游戏，或实现更多操作。
* 改进对音乐格式的支持,支持休止,连音,重音,重奏等等高级特性,丰富音乐表达.
* 游戏功能普遍化。

# 参考文献

[1]许杨杨.音乐学科核心素养与初中音乐教育研究[J].大众文艺,2019(16):221-222.

[2]宋涛.浅析儿童乐器学习中的心智引导[J].黄河之声,2019(01):98.

[3]侯张平.高中音乐教育中器乐教学的必要性与教学方法研究[J/OL].黄河之声,2019(13):94-95[2019-08-28].

[4]徐弛.小学音乐教学中的器乐教学分析[J/OL].黄河之声,2019(13):102[2019-08-28].

[5]周鸿福.小学音乐课堂乐器教学实践研究与思考[J/OL].黄河之声,2019(13):105[2019-08-28].

[6](美)John M. Hughes著.Arduino技术指南[M]. 武传海译.北京 : 人民邮电出版社, 2017.

[7](英) Simon Monk著.Programming Arduino next steps : going further with sketches.张佳进 ... [等] 译.北京 : 人民邮电出版社, 2016.

[8](美) Massimo Banzi, (美) Michael Shiloh著.Getting started with Arduino.程晨译.北京 : 人民邮电出版社, 2016

[9](英) Alan Trevennor著.AVR单片机实战 : Arduino方法.程晨译.北京 : 机械工业出版社, 2014

# 附录

Copyright @ MIA project

## 附录1：内部协议文件

### 1.1 歌曲文件格式:

第1byte :文件类型,0b01010110 表示单音歌曲,0b01010001 表示和弦歌曲

后面每两个byte为一组

00000000

0000   0000

B1

B2.1    B2.2

对于单音歌而言:

B1:

频率/HZ

从1开始的!!!

1 NOTE\_G3 196

2 NOTE\_A3 220

3 NOTE\_B3 247

4 NOTE\_C4 262 1

5 NOTE\_D4 294 2

6 NOTE\_E4 330 3

7 NOTE\_F4 349 4

8 NOTE\_G4 392 5

9 NOTE\_A4 440 6

10NOTE\_B4 494 7

11NOTE\_C5 523 i

12NOTE\_D5 587

13NOTE\_E5 659

14NOTE\_F5 698

暂时只规定14个音,约涵盖两个八度

0000 0000代表文件截止.

B2.1:

时长:用0b00010000减去B2.1所得值,转换为二进制小数,0010000代表个全音.

B2.2:

其他描述,如强度,连音,休止etc..暂时不实现(学琴对象是初学者!!!)

在此版本中先初始化为1111

### 1.2 Eeprom 内的文件格式定义:

0 标识符 0b01010110

1-10 密码存储部分(10byte)

11-12 int 总歌曲

13-14 int 已用空间

15-… 歌曲部分

最后100字节:

应用保留的存储空间

具体的:

\*\*00-\*\*01 uint 游戏 十面埋伏的简单最高分

\*\*02-\*\*03 uint 游戏 十面埋伏的困难最高分

\*\*03-\*\*04 uint 单音练习的次数

\*\*04-\*\*08 ulong 单音练习毫秒数

\*\*09-\*\*12 ulong 单音练习的音节数

\*\*13-\*\*14 uint 节奏练习的次数

\*\*15-\*\*18 ulong 节奏练习的毫秒数

### 1.3 蓝牙透传通信协议:

虽然不是完整的可靠的协议,但勉强能用...

1.初始化连接:

建立连接后,从机发送字母a

主机响应字母a

从机响应字母 b

主机再发送8 byte模式标识符

从机响应字母 b

2.通信:

从机发送数字1-8,等待主机响应

主机回发数字1-8,完成一次通信

3.结束:

从机发送字母z

## 附录2：源码

见文件夹