

北京交通大学

电工电子实验教学中心

嵌入式系统课程设计

设计报告

设计题目：音节跳动-音乐频谱显示器

学生 1	姓 名:	刘骐铭
	学 号:	18211382
	专 业:	轨道交通信号与控制
学生 2	姓 名:	魏中正
	学 号:	18211388
	专 业:	轨道交通信号与控制
选课教师:	赵翔	

2021 年 11 月 4 日

目 录

目录

1	设计任务要求.....	1
1.1	任务背景.....	1
1.2	设计任务.....	2
2	系统设计方案.....	3
2.1	系统顶层设计 (15 分)	3
2.2	小组人员分工 (5 分)	3
2.3	分工接口定义 (5 分)	3
3	功能模块设计方案 (30 分)	4
3.1	模块 1 时间显示功能.....	4
3.2	模块 2 时间显示功能.....	6
3.3	模块 3: 波形显示功能.....	8
3.3.3	CubeMX 设计方案.....	9
3.4	模块 4 频谱显示功能.....	12
3.5	模块 5 多种频谱显示功能.....	13
3.6	模块 6 主音调测量.....	15
3.7	模块 7 音乐播放功能.....	17
4	系统设计中问题及解决 (10 分)	19
5	系统使用说明.....	20
5.1	系统外观及接口说明 (5 分)	20
5.2	系统操作使用说明 (15 分)	22
5.3	技术指标 (5 分)	24
6	总结.....	24
6.1	设计结果.....	24
6.2	心得体会.....	25
6.3	意见建议.....	25

1 设计任务要求

1.1 任务背景

当我们使用音乐播放器播放音乐时，在界面上会看到一个随着音乐节奏跳跃的柱形图，跟随着音乐的节奏像波浪般跳跃、舞动，让我们在聆听音乐的同时还能观察到每一个音符的细节变化，配合多种动感的显示效果，让播放器界面变得生动了起来；在录音棚等专业的音乐制作设备上，我们也经常能看到屏幕上有一排高高低低的光柱在不断地上下跳动，与音乐的高低和强度同步，这就是音乐的频谱显示器；如图1.1.1所示：



图 1.1.1

音乐频谱显示器是一种能够采集音频信号并显示音频信号频谱的设备。

本题目要求设计并制作一个多功能音乐频谱显示器，具有采集声音信号、实时显示音频信号时域波形、功率谱的功能；具有时钟和闹钟功能，可以显示时间、万年历；具有闹钟功能可以开启或关闭闹钟，设置闹钟时间；具有音乐播放功能，可以使用蜂鸣器播放MIDI格式（或简化格式）的电子音乐。

题目还要求在频谱显示功能的基础上扩展乐曲主音调提取功能，可以检测并显示音高最大的音调及对应的音阶、音高；可以提取、存储乐曲的主音调并回放。



图 1.1.2

1.2 设计任务

使用本课程提供的 STM32F746 开发平台和配套电路模块设计并制作一个音乐频谱显示及音调提取系统。具体要求如下：

(1) 时间显示功能 (15 分)

在屏幕上显示当前日期、时间、星期。可以通过按键或触摸屏修改当前日期、时间。

(2) 闹钟功能 (10 分)

可通过按键或触摸屏开启或关闭闹钟、设定闹钟时间，在设定的闹钟时间到来时可进行声光提示。

(3) 波形显示功能 (10 分)

系统可以通过麦克风模块采集声音信号，在液晶屏上实时显示声音信号的时域波形。

(4) 频谱显示功能 (15 分)

系统可以通过麦克风模块采集声音信号，在液晶屏上实时显示声音信号的功率谱。要求可显示的频率范围不低于 100Hz~18kHz，功率谱不少于 32 点 (段)。

(5) 频谱显示类型切换功能 (10 分)

具有多种频谱显示模式，可手动或自动循环切换显示样式。

(6) 主音调测量功能 (10 分)

系统可自动测量并显示当前声音信号功率谱最大分量（主音调）的频率值，频率分辨率 $\leq 100\text{Hz}$ ，误差 $\leq 5\%$ 。

(7) MIDI 音乐播放器功能 (15 分)

系统的 USB 接口可以设置为大容量存储设备模式 (Mass Storage Device)，可以在 PC 机上通过 USB 线将 MIDI 音乐文件 (文件格式可用标准 MIDI 格式也可自定) 保存到

播放器中。通过触摸屏或按键选择文件后，系统可通过无源蜂鸣器播放音乐。

(8) 音调提取、回放功能 (15 分)

播放器可提取功率谱中幅度最高的谱线并保存、回放。可通过按键或触摸屏控制保存一段长度为 20 秒的音频，系统可自动提取并保存音频的关键信息（主音调）并通过无源蜂鸣器进行回放。

(9) 附加功能 (附加分，上限 10 分)

扩展其他对本系统有意义的功能，如电子琴、节拍器或界面美化、中文界面、屏保等功能，视难度酌情加分

2 系统设计方案

2.1 系统顶层设计 (15 分)

2.1.1 设计方案

本次设计了一个有音乐频谱显示及提取功能的系统，共分为了八项小功能，通过讨论，我们决定将一、二功能放在一个 emwin 的 window 界面上，将波形显示功能放在第二个 window 界面上同时系统可以实现主因调的测量功能，将频谱显示功能放在第三个 window 界面上，并通过触摸屏修改的方式来切换 window 界面，实现频谱显示类型切换功能。将 USB 接口设置为 MSD，将文件存储到播放器中，使用触摸屏进行播放音乐。并在第二个 window 界面上，可以通过触摸屏保存一段音频，并可以回放。

2.1.2 系统框图

如图 2.1.1 所示：

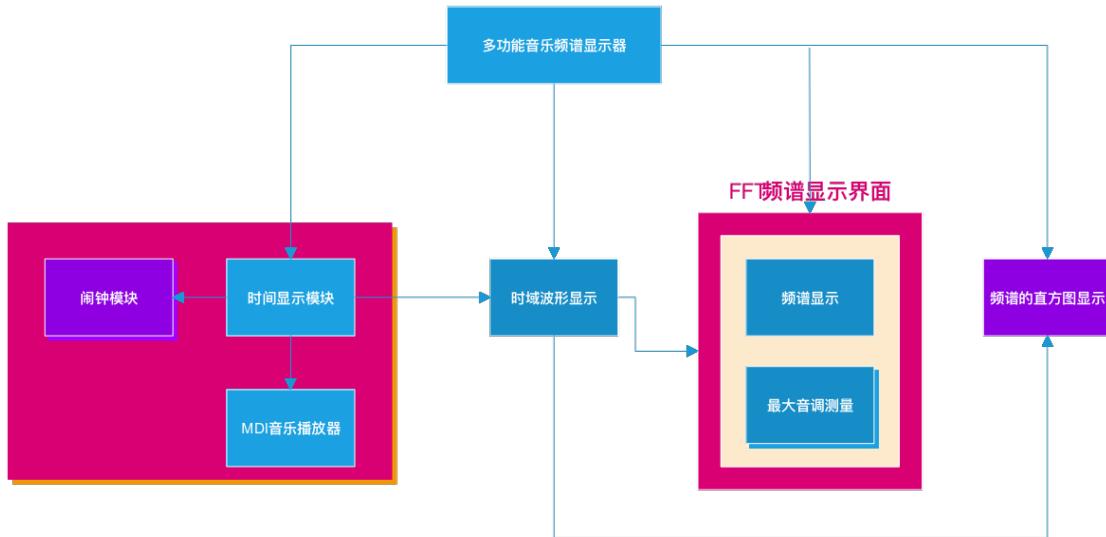


图 2.1.1

2.2 小组人员分工 (5 分)

刘祺铭：波形显示功能，频谱显示功能，频谱显示类型切换功能，主音调测量功能；

魏中正：时间显示功能，闹钟功能，MIDI 音乐播放器功能；

2.3 分工接口定义 (5 分)

采用窗口界面切换的方式进行各个模块间的转换；采用按钮作为各个模块的接口。按钮按下时，隐藏当前界面，显示目标界面，以此类推。由于采用显示、隐藏窗口的方式，多个

窗口共同运行，保证功能间的无缝切换。

例如如图 2.3.1 所示，通过“ClktoTime”按钮即可转换到时域模块，如图 2.3.2 所示。



图 2.3.1



图 2.3.2

3 功能模块设计方案（30 分）

3.1 模块 1 时间显示功能

3.1.1 模块功能要求

在屏幕上显示当前日期、时间、星期。可以通过按键或触摸屏修改当前日期、时间。

3.1.2 模块设计思路

利用 TIM2，设置一个 1s 触发一次的事件，从而做到 1s1 次的计时，将 second 与 59 比较，大于则 minute 加 1，同理，分钟，小时，年月日都这样进行比较，从而显示时间。

闹钟利用 1s 一次的事件，进行倒计时，可以通过按键关闭或者开启定时器从而控制闹钟的开断，在闹钟倒计时到零后，开启 TIM3 的 Channel2 通道，输出一个 PWM 波，从而让蜂鸣器发出声音，并且改变 PI 的状态，让灯发光。

3.1.3 CubeMX 设计方案

开启 TIM2 和 TIM3 的 Channel2 通道，并设置 P1 和 PC7 为 GPIO_OUT 模式，PI11 和 PG7 为 EXIT 中断模式，并将 TIM2 的时钟频率调为 1s，TIM3 的时钟频率调为 1ms，输出 PWM 波。

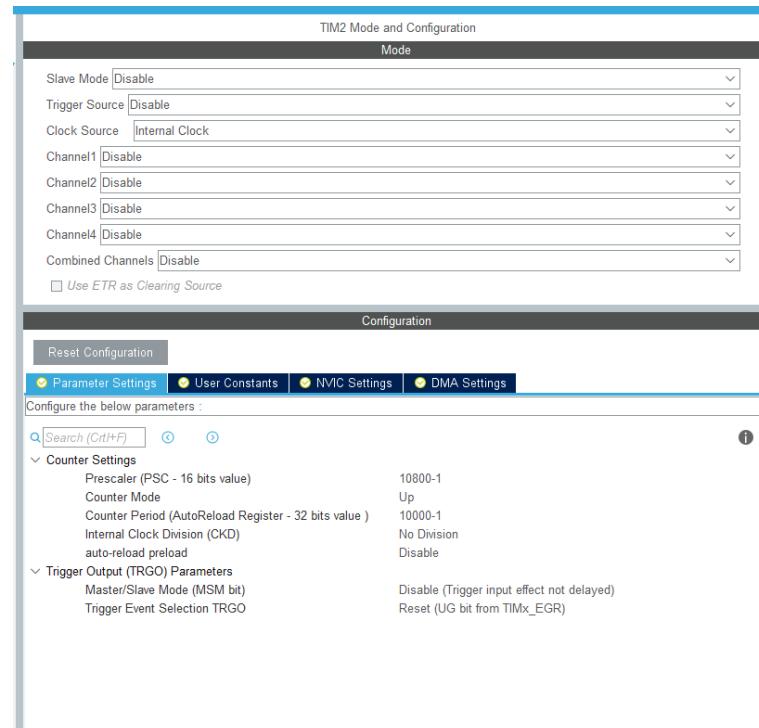


图 3.1.1

3.1.4 模块内部结构及软件流程图

模块内部结构如图 3.1.2 所示，软件流程图如图 3.1.3 所示；



图 3.1.2

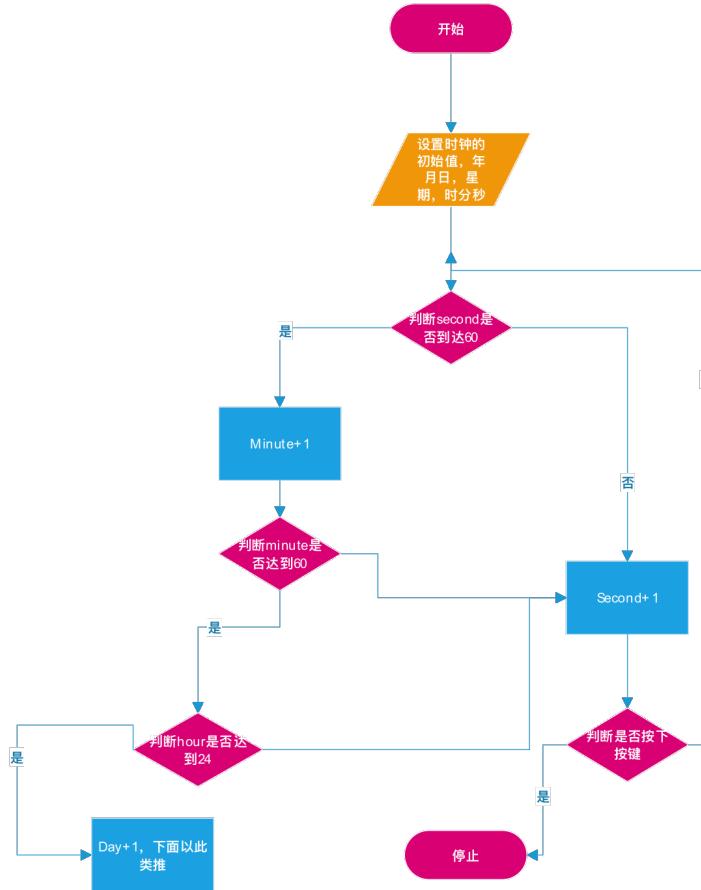


图 3.1.3

3.2 模块 2 时间显示功能

3.2.1 模块功能要求

可通过按键或触摸屏开启或关闭闹钟、设定闹钟时间，在设定的脑中时间到来时可进入声光提示；

3.2.2 模块设计思路

利用 TIM2，设置一个 1s 触发一次的事件，从而做到 1s1 次的计时，将 asecond 与 59 比较，大于则 minute 加 1，同理，分钟，小时都这样进行比较，从而显示闹钟。

闹钟利用 1s 一次的事件，进行倒计时，可以通过按键关闭或者开启定时器 2 从而控制闹钟的开断，在闹钟倒计时到零后，开启 TIM3 的 Channel2 通道，输出一个 PWM 波，从而让蜂鸣器发出声音，并且改变 PI 的状态，让灯发光。

3.2.3 CubeMX 设计方案

开启 TIM2 和 TIM3 的 Channel2 通道，并设置 P1 和 PC7 为 GPIO_OUT 模式，PI11 和 PG7 为 EXIT 中断模式，并将 TIM2 的时钟频率调为 1s，TIM3 的时钟频率调为 1ms，输出 PWM 波；如图 3.2.1、图 3.2.2 所示；

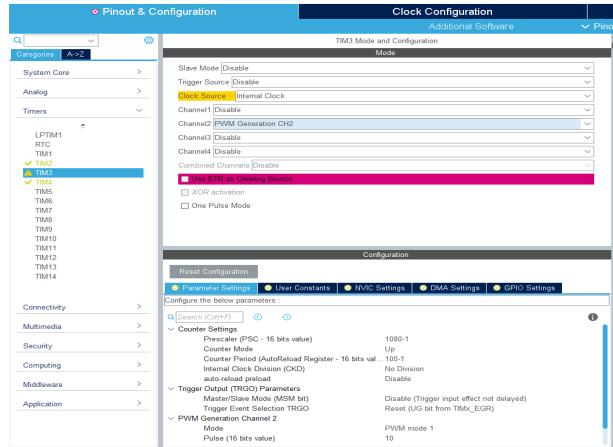


图 3.2.1

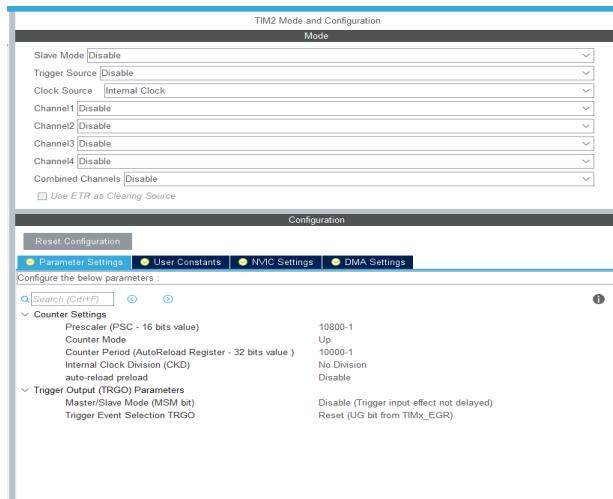


图 3.2.2

3.2.4 模块内部结构模块软件流程图

模块内部结构如图 3.2.3 所示，软件流程图如图 3.2.4 所示；



图 3.2.3

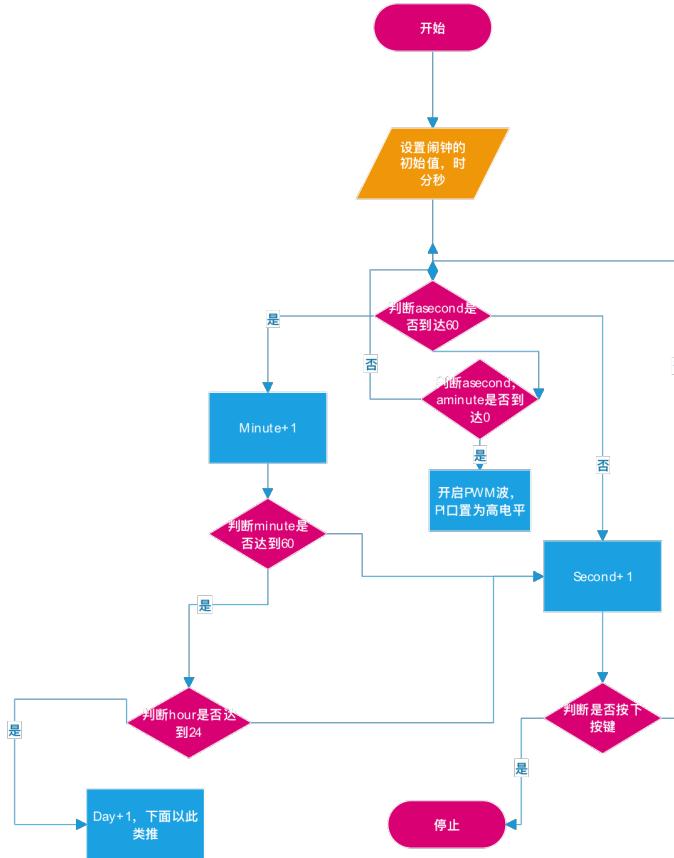


图 3.2.4

3.3 模块 3：波形显示功能

3.3.1 模块功能要求

系统可以通过麦克风模块采集声音信号，在液晶屏上实时显示声音信号的时域波形。

3.3.2 模块设计思路

首先实现麦克风和 STM32F746NG 板的连接，采用的是 ADC3 的 IN0 接口，并开启转换完成中断、DMA 采样，在 IAR 中的“main.c”文件的 main 函数中开启 ADC、定时器、DMA 采样，其中 DMA 设置为每秒采集 512 个点；并宏定义一个数组“buffer”作为接收 DMA 采样的缓存数组，大小为 512。在中断函数中找到 ADC 转换完成中断，找到其中断回调函数，并在“main.c”文件中重新定义，在里面设置关闭定时器。控制定时器即控制 DMA，因为 DMA 采样是由定时器作为触发源的。

在 GUI_app 文件中的 while (1) 主循环中利用“GRAPH_DATA_YT_AddValue ()”函数将 buffer 的数值，采用 for 循环，赋给 hData1 中，hData 即用来在 LTDC 屏上显示波形；在每一次显示完毕后，打开定时器。

关于定时器的打开与关闭，是因为 DMA 采样一次后屏幕直接刷新显示，但由于采样速率大于屏幕刷新速率，因此直接关闭和打开定时器来避免显示问题。

3.3.3 CubeMX 设计方案

除了基本的“system core”、“时钟树”的设置外，还需分别对定时器、ADC、LTDC 显示屏进行设置；

定时器：如图 3.3.1 所示，采用 TIM2、Channel2、“PWM 无输出模式”，采样速率设为 40kHz；由于是从 0 开始计数，预分频设为 270-1，计数周期设为 10-1，因此算下来： $108000000/270/10=40k$ 。

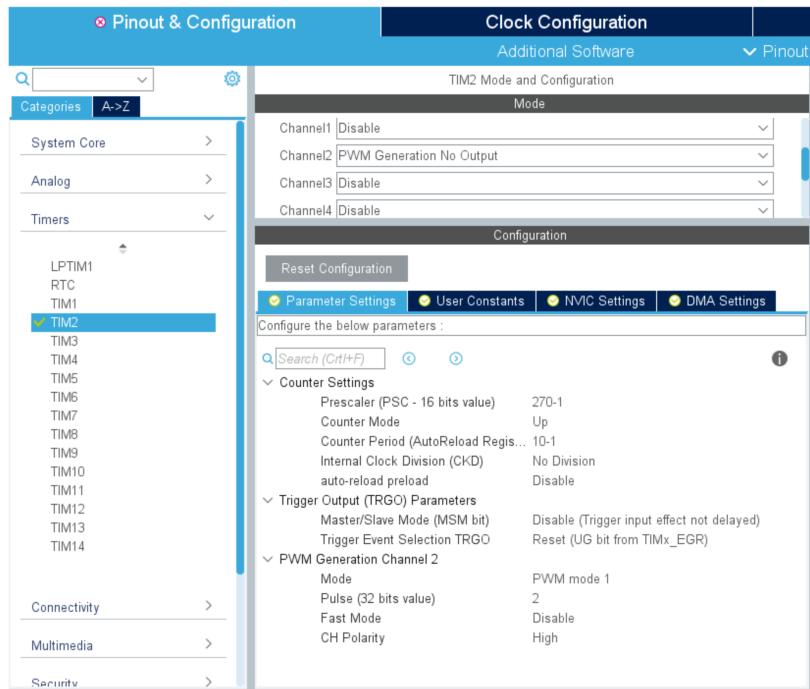


图 3.3.1

ADC 设置：

如图 3.3.2 所示；

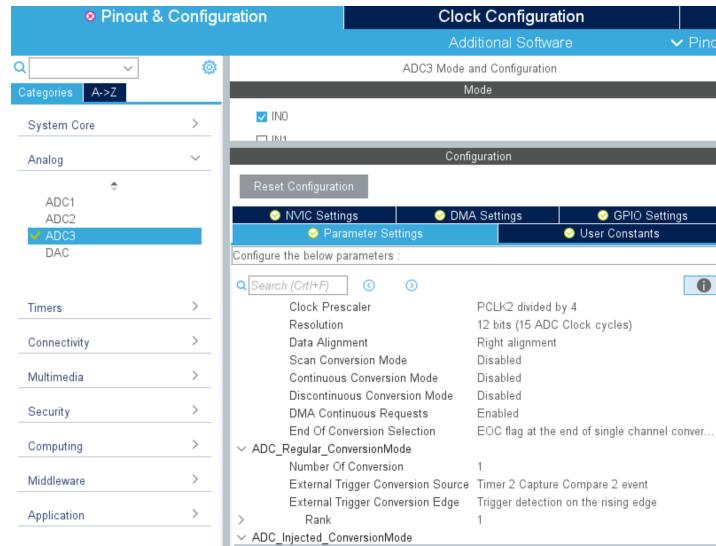


图 3.3.2

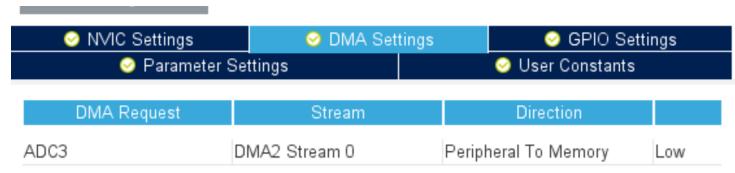


图 3.3.3

Parameter Settings		User Constants	
NVIC Settings		DMA Settings	
GPIO Settings		GPIO Settings	
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
ADC1, ADC2 and ADC3 global interrupts	<input checked="" type="checkbox"/>	0	0
DMA2 stream0 global interrupt	<input checked="" type="checkbox"/>	0	0

图 3.3.4

LTDC 显示屏设置：修改对应的管脚

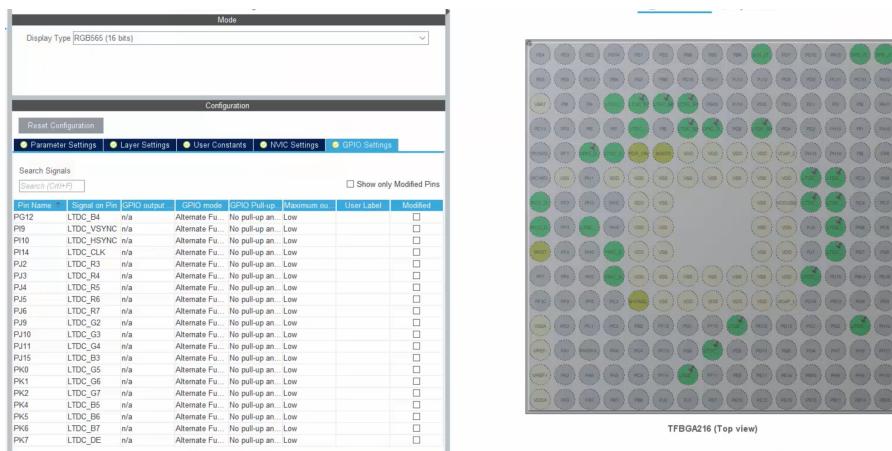


图 3.3.5

3.3.4 模块内部结构模块软件流程图

模块内部结构如图 3.3.6 所示，软件流程图如图 3.3.7 所示；



图 3.3.6

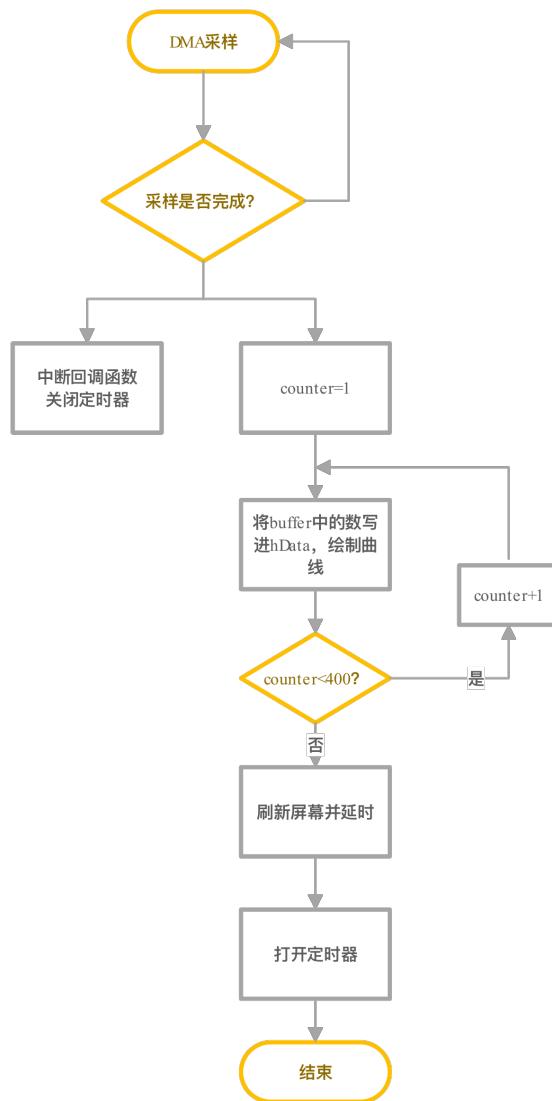


图 3.3.7

3.4 模块 4 频谱显示功能

3.4.1 模块功能要求

系统可以通过麦克风模块采集声音信号，在液晶屏上实时显示声音信号的功率谱。要求可显示的频谱范围不低于 100Hz-18kHz，功率谱不少于 32 点（段）。

3.4.2 模块设计思路

如图 3.4.1 所示，在 DMA 采样完成后在中断回调函数中进行引入 FFT 变换函数，“arm_rfft_fast_f32(&S, testInput_f32, testOutput_f32, 0)”函数对 buffer 中的数据全部进行 FFT 变换，变换完成后的数值在存在“testOutput_f32”；再将“testOutput_f32”中的 FFT 变换结果利用“arm_cmplx_mag_f32(testOutput_f32, testOutputMag_f32, 512)”变换为功率谱，并在 GUI_app 中输出到 LTDC 屏幕上。

```
/* 1024 点实序列快速 FFT, ifftFlag = 0:正变换 */
arm_rfft_fast_f32(&S, testInput_f32, testOutput_f32, 0);
/* 解出了 512 点复数, 共1024个float32值, 计算512点功率谱, 存放在testOutputMag_f32中*/
arm_cmplx_mag_f32(testOutput_f32, testOutputMag_f32, 512);
```

图 3.4.1

3.4.3 CubeMX 设计方案

在 StemWin 中额外设置一个窗口命名为“Fre”，添加按钮和文本；显示相应界面信息和进行切换功能；

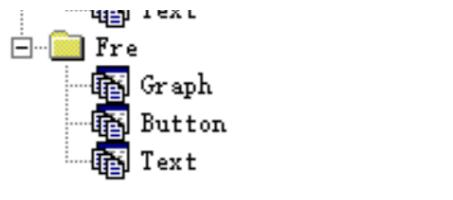


图 3.4.2



图 3.4.3

3.4.4 模块内部结构模块软件流程图

模块内部结构如图 3.3.6 所示，软件流程图如图 3.4.4 所示；

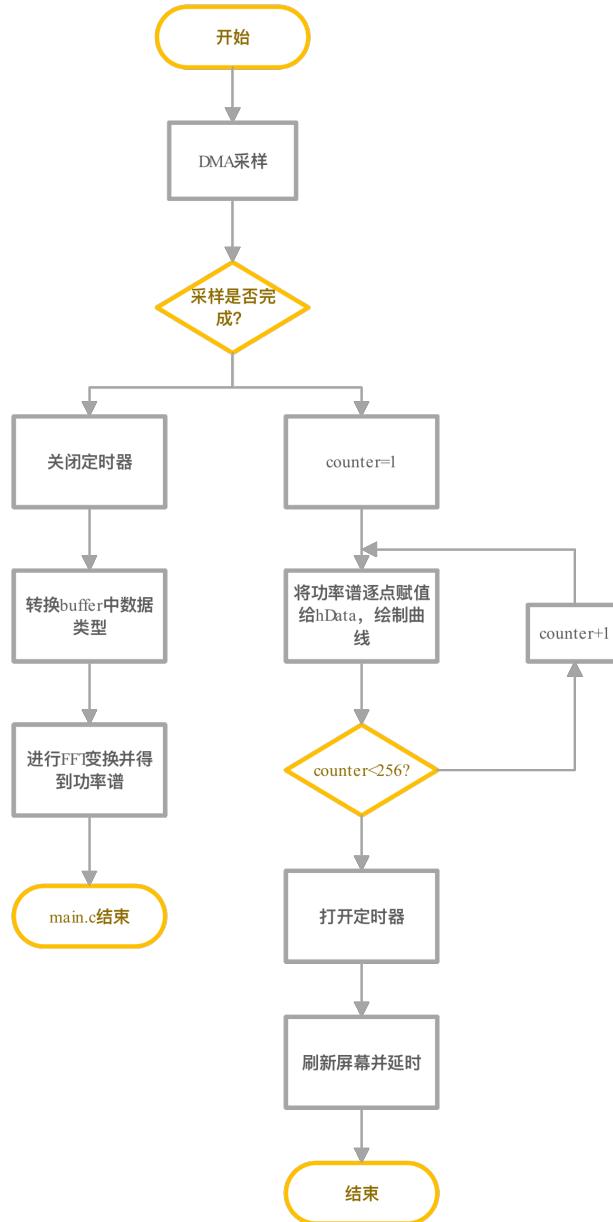


图 3.4.4

3.5 模块 5 多种频谱显示功能

3.5.1 模块功能要求

具有多种频谱显示样式，可手动或自动循环切换显示样式；

3.5.2 模块设计思路

采用直方图来显示频谱；利用 DrawGradientV () 函数，确定矩形左上角及右下角的坐标，确定一个矩形包含的点数，并将这些点在一个取最大值的函数中取得最大值，作为矩形的高度，由此得以确定；采用 for 循环，依次进行点数的遍历，直至 256 点功率谱遍历完成。

3.5.3 CubeMX 设计方案

如图 3.5.1 所示，新建一个窗口命名为“Fre2”，并在其中添加按钮和文本，显示相关信息和进行页面切换；在这里由于“GradientV”函数的特殊性——只能在最底层窗口上画矩形，该页面的设置没有让直方图在 window 上画，而是在底板上画，如图 3.5.2 所示；

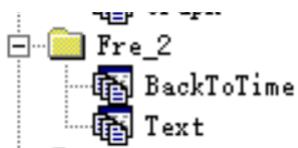


图 3.5.1



图 3.5.2

3.5.4 模块内部结构模块软件流程图

内部结构如图 3.3.6 所示，软件流程图如图 3.5.3 所示；

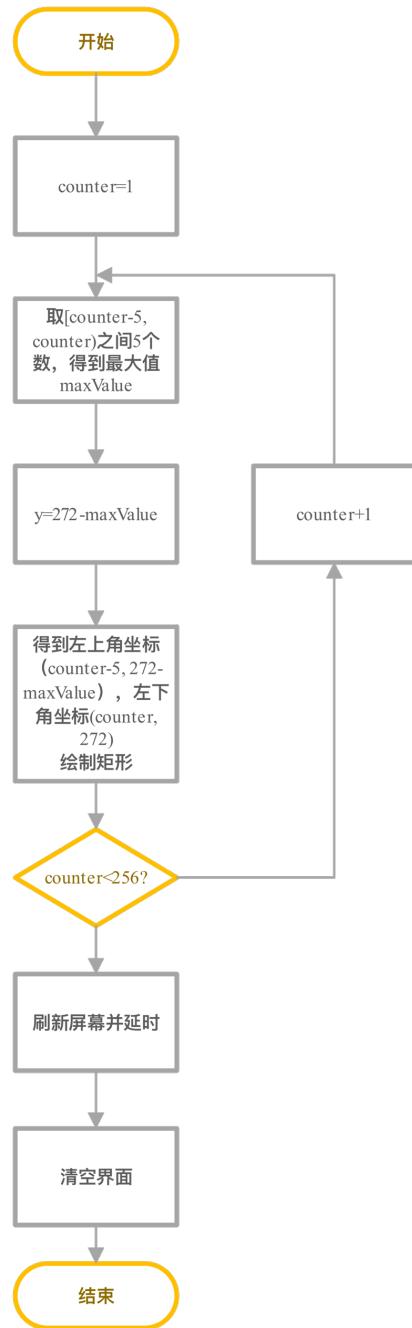


图 3.5.3

3.6 模块 6 主音调测量

3.6.1 模块功能要求

系统可自动测量并显示当前声音信号功率谱的最大分量（或主音调）的频率值，频率分辨率大于 100Hz，误差不大于 5%；

3.6.2 模块设计思路

由于 DMA 采样的点数为 512 点，变成功率谱后为 256 点；利用 for 循环依次比较功率谱

的大小，找到最大的功率谱后将下标赋给参数“max”，利用公式“最大频率=最大功率对应下标/FFT 采样点数*采样点数”计算出最大频率并利用“sprintf”函数进行显示；

3.6.3 CubeMX 设计方案

如图 3.6.1 所示，在 StemWin 界面中额外添加一个文本，最后的最大值将显示在这个文本上；

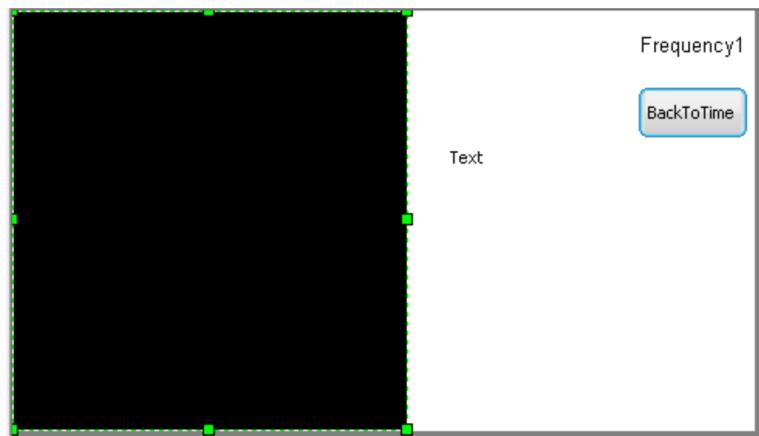


图 3.6.1

3.6.4 模块内部结构及模块软件流程图

模块内部结构如图 3.3.6 所示，软件流程图如图 3.6.2 所示；

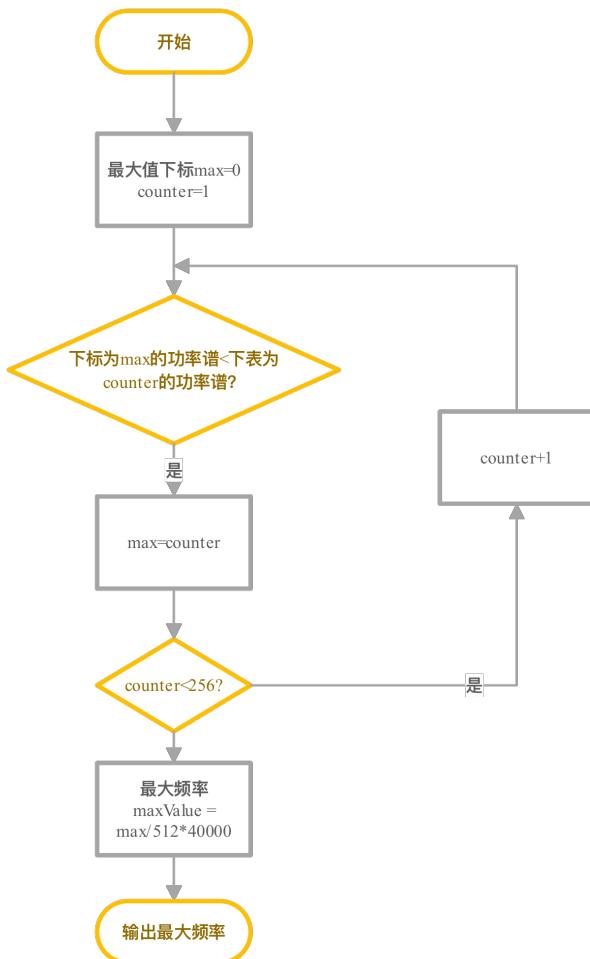


图 3.6.2

3.7 模块 7 音乐播放功能

3.7.1 模块功能要求

可通过按键或触摸屏选择音乐文件，并播放，并可以通过 USB 线，在 PC 端修改音乐文件，STM32 可以播放修改后的音乐。

3.7.2 模块设计思路

开启 SD 卡，USB 口，USBdevice，USB_OTG_FS 等相关硬件设置，首先设置 SD 卡，挂载系统，再将音乐文件写入 SD，在本次开发中，我使用 TXT 格式储存音乐，一个文件储存音调，一个文件储存节拍。

在写入文件后，编写 audioplay 音乐函数，读取储存到 SD 中的音乐文件，然后保存到数组中，再读取数组储存的音调和相应的节拍，通过 PWM 波控制无源蜂鸣器发出不同频率的声音和延迟不同的事件，进而演奏音乐。

3.7.3 CubeMX 设计方案

开启 SDMMC1, USB_OTG_FS,USBdevice, PC13 端口设置为 GPIO_Inout 模式, 开启 USB 的 MSB 大容量存储模式。

文件系统的 FATFS 模式, 以便挂载 SD 卡的文件系统。

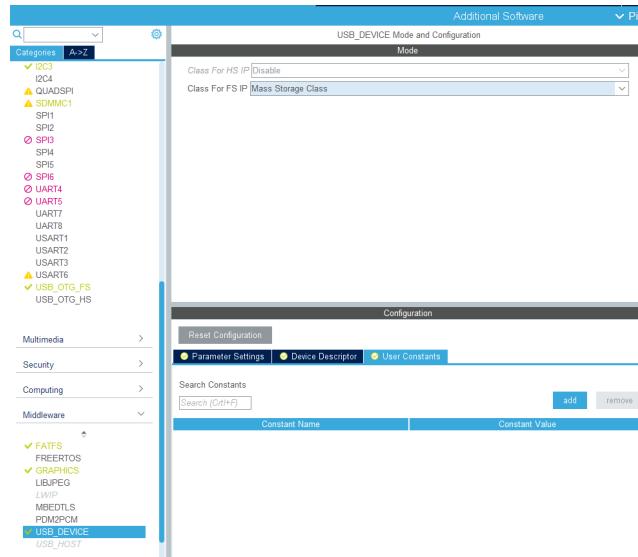


图 3.7.1

3.4 模块内部结构模块软件流程图

模块内部结构如图 3.3.6 所示, 软件流程图如图 3.4.4 所示;

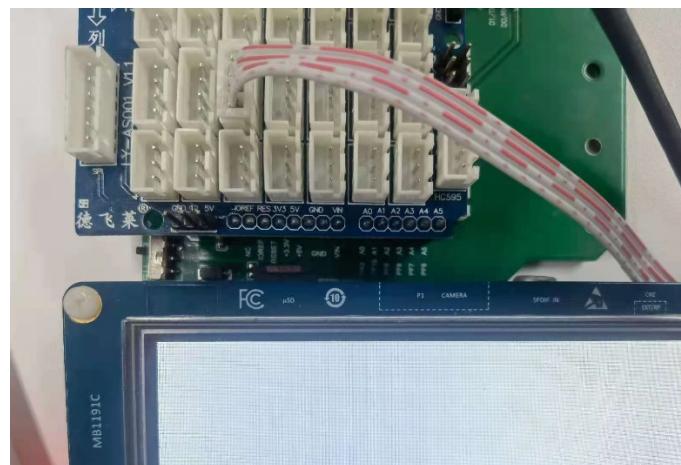


图 3.7.2

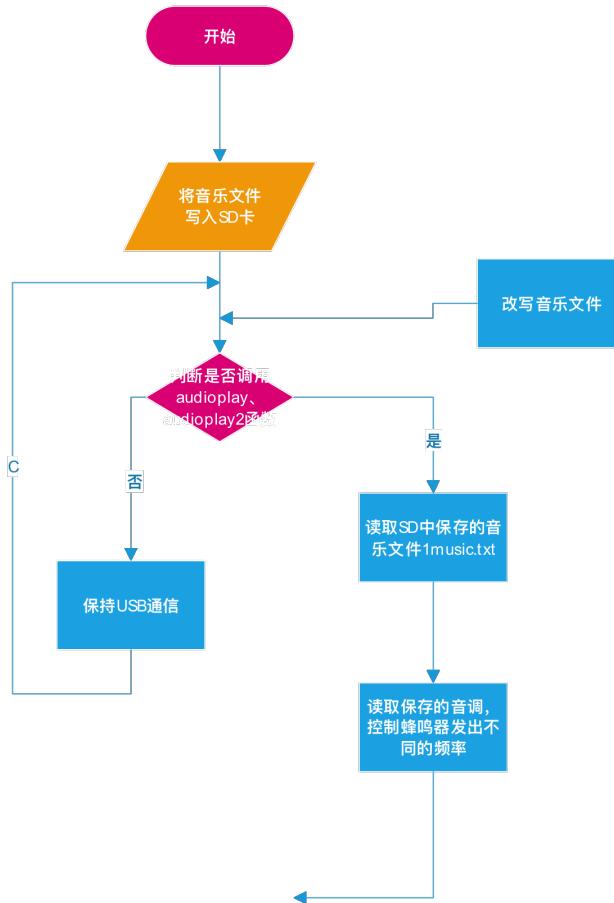


图 3.7.3

4 系统设计中问题及解决 (10 分)

4.1 问题 1

问题：在写入 SD 的过程中，发现可以写入 SD 卡，但是当开启 FATFS 文件系统时，无法写入。

产生原因：SD 卡没有进行格式化，使用的 USB 线为充电线，PC 端无法识别。

解决方法：先利用 SD 卡读取工具，将 SD 卡进行格式化，然后选择 USB 数据线插入，发现可以写入数据。

4.2 问题 2

问题：怎么让蜂鸣器发出声音，怎么让蜂鸣器发出不同的频率

产生原因：蜂鸣器发出声音有两种方法，一是利用 PWM 波，另一种是利用 GPIO 端口高低电平的快速切换。但是两种发出的声音都不清晰。

解决方法：使用 PWM 波控制蜂鸣器发出声音，GPIO 端口的频率不确定，而且 PWM

波发出的声音更加清晰, 先确定溢出计数, 在通过修改预分频器的方法来改变 TIM3 的频率, 从而让 PWM 波产生不同到达频率, 无源蜂鸣器发出不同的声音。

4.3 问题 3

问题: 在显示时域波形时显示异常, 出现波形重叠、停顿等现象;

产生原因: 开启了 DMA 采样却仍然使用 ADC 的“getvalue”函数, 从 CPU 中读取数值而不是从 DMA 的缓存数组中读取, 因此造成了波形卡顿; DMA 应设置为了双通道, 因此采集了两个通道的波形同时存在了一个 buffer 缓冲区中, 造成了波形的重叠;

解决方案: 采用 buffer 数据在 GUI_app 文件中利用 for 循环直接输出; 关于仅仅输出波形一事, 无需再在中断回调函数中进行设置;

4.4 问题 4

问题: 显示与采样不同步

产生原因: 采样速率由采样率-40kHz 所决定, 显示速率与屏幕刷新率有关, 但无需求出; DMA 采样了 512 个点, 将在 GUI_app 中显示 i=400 个点, 但是在 for 循环显示 10 个, 比如说, 点后, DMA 继续采样又接着显示了下一个 400 个点, 屏幕在没显示完 400 个点时又紧接着显示了下一组 400 个点;

解决方案: 适时关闭 DMA 采样; 显示完后再将 DMA 打开中断回调: 采样完成后将执行这个中断回调函数, 因此在该函数中关闭 DMA 采样, 再循环完成——显示完成后, 再将 DMA 采样打开;

5 系统使用说明

5.1 系统外观及接口说明 (5 分)

系统外观, 如图 5.1.1、图 5.1.2 所示,



图 5.1.1



图 5.1.2

如图 5.1.3 所示图中的第一部分，为按键，此按键可以控制闹钟及时钟的中断，按下去，闹钟停止。

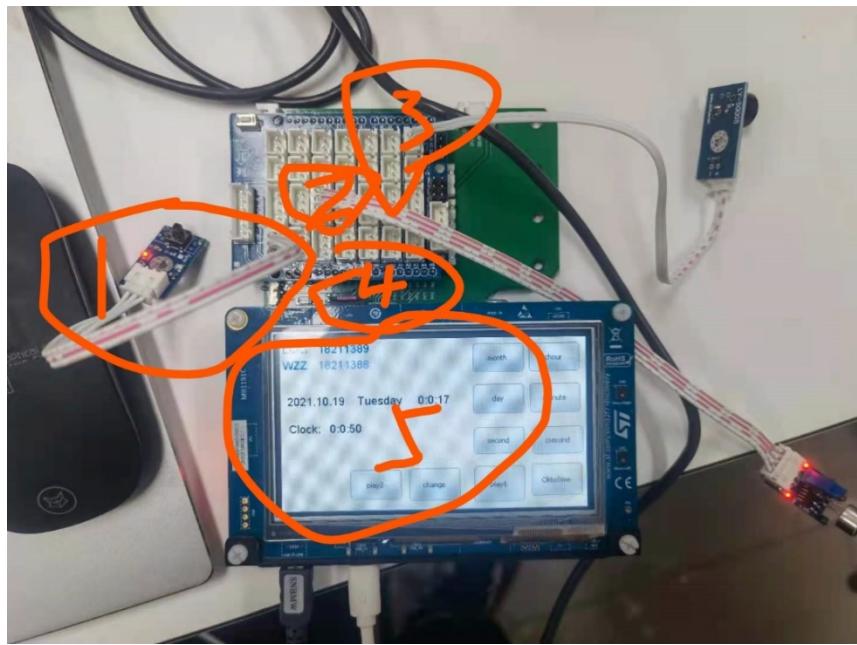


图 5.1.3

板子下面的蓝色按键，可以控制闹钟及时钟的开始，按下去，时钟及闹钟开始计时。

第二部分为 DAC 转换，接入的为蜂鸣器，此部分可以接收外部的声音信号，将模拟量转化成数字量，并绘制成时域图形。

第三部分为蜂鸣器，此部分可以发出音乐，通过 PWM 波控制蜂鸣器发出不同频率的声音，加上节拍，共同构成了音乐。

第四部分为 SD 卡，此部分可以通过 USB 数据线与 PC 端连接，通过 PC 端修改保存在 SD 卡中的音乐文件，从而修改蜂鸣器发出的声音。

第五部分为显示屏幕，此部分将所有功能完整的显示在屏幕上，通过屏幕上的按钮，可以完成时钟修改，闹钟修改，播放音乐，切换页面等功能。

5.2 系统操作使用说明 (15 分)



图 5.1.4

开机即可进入日期界面，如图 5.1.4 所示。在本界面中，可以通过右上角的六个按钮进行，时钟的月，日，时，分，秒加一或减一，以及可以完成闹钟的时分秒加一或减一，change 按钮可以控制加一或者减一，play1 表示播放第一个音乐，play2 表示播放第二个音乐。Clktotime 按钮的功能是从日期界面跳转到时域界面。



图 5.1.5

如图 5.1.5 所示，本界面为时域界面，在本界面中，button 按钮可以跳转到日期界面，gotofre1 按钮可以跳转到频域的第一种样式界面，gotofre2 按钮可以跳转到频域的第二种样式界面。

如图 5.1.6 所示，为频域的第一种样式界面，此界面只有一种按钮，本按钮可以跳转到时域界面。

如图 5.1.7 所示，本界面为频域的第二种样式界面，此界面只有一种按钮，本按钮可以跳转到时域界面。



图 5.1.6



图 5.1.7

5.3 技术指标 (5 分)

能够准确显示时间、日期，精确到秒，并能对时间、日期进行数值上的任意调整。对于波形显示方面，定时器采用了 40kHz 的采样频率，通过 ADC3 采用 DMA 采样，采样点数设为了 512，同时这也是 FFT 的点数，显示出来的频谱是基于 FFT 的 256 点功率谱。在显示时域波形方面，78Hz~20kHz，在显示最大频谱时能够精确到个位，误差不超过 5%。

6 总结

6.1 设计结果 (5 分)

本作品实现了基于 STM32F407ZG 板上的嵌入式频谱显示系统的设计，能够进行正常的时间、日期显示，并进行闹钟设置，同时能够从 SD 卡读取音乐文件并进行播放；通过按钮切换后，系统可以进入波形显示界面，关于波形，一共可以显示由麦克风输入的声音信号的

时域波形和两种频域波形，并能够进行最大频率的测量。

本系统功能完善，且能够精确的实现各个指标，体现着设计者严谨科学的态度与高超的水平。

6.2 心得体会（5分）

6.2.1 刘骐铭 18211382:

当时选课时确实是抱着嵌入式与软件方面相比于 EDA、DSP 更大的想法而选的，同时也有着些进入相关职业的一些热身的心态，这八周下来，从听 mooc 到线下上课到着手实践设计是进度紧密的；我庆幸自己能够按照大致的安排严格的执行着进度，虽然仍然拖到了最后一刻才提交，但是总算有一个较为完整、成功的结果，这是以前少有的。这门课的完成不仅对我的职业规划有着意义，同时也是我对于这工程类实验、项目与考核的一个里程碑，从四处询问到最后时刻帮助他人解答问题，看到自己对于相关操作熟练的讲授，心中多少是有成就感的，但是从现实来说，这或许就是这一类职业的体验了——不断踩坑、不断调试、不断积累经验，过程很艰难，但当稍稍入门后，一切都变得清晰起来。

这八周里，我在收获专业知识的同时，也为我将来从事相关职业、继续进行深造树立了信心，愿在以后的学业、职业道路上能取得更大的进展。

6.2.2 魏中正 18211388:

在选择这门嵌入式系统课程之前，我对嵌入式系统没有太多理解，但是在经历过八周的学习后，我明白了嵌入式系统的整个大概流程，如何在 STM32 进行开发，开发最后形成的产品是什么。

在之前的实验中，我做实验都是没有规律的，但是这门课要求我们必须要有整体规划和部分功能的分工，在这个过程中，我学习到了如何利用定时器制作 1s 的定时和输出 PWM 波，以及如何使用函数输出不同的频率，从而让蜂鸣器发出不同的声音，组成音乐。

在学习这门课的过程中，我付出了许多时间，但是也收获了许多知识，收获匪浅。

6.3 意见建议

意见：课程进度设置合理，先从 mooc 进行入门，再结合课上的讲解答疑，能够以相对高效的方式让学习者快速入门嵌入式的设计；同时题目在第一周就进行发布，有助于学习者进行尽早的准备，合理规划时间；

建议：mooc 可以加大宣传力度，并提早释放，比如在暑假时候就可以开放并让选课学生得知有 mooc；八周的时间再加上大四培养计划内的一些课程，再加上考研、出国申请、

找工作等事务，时间极其紧张。另外建议指导者能够更多调研学生在第八周的情况，适当调整答辩时间，以减轻学习者考试等心理压力，有助于保持其身心健康。