**西南交通大学**

**嵌入式系统实验日志**



**实验名称：**MP3播放器设计

**实验序号：**12

**实验组号：**3组-6队

**使用语言：**Python

**姓名学号：**杨志成 2023211042

**实验时间：**2023年12月22日

**指导老师：**徐图

**一、实验目的**

1.了解树莓派音频模块的设置

2.熟悉和掌握 Tkinter 基础，并学会设计 Tkinter 界面

3.熟悉和掌握 Pygame 音频处理相关知识，并设计音频播放程序

4.结合Tkinter 和 Pygame，设计一个 MP3 播放器，提高综合能力

**二、实验步骤：**

1. 对树莓派进行相应的音频设置，检测是正常。

2.新建MusicPlay.py、MusicList.py、MusicButtonControl.py 三个文件，以及 music 文件夹 （用来存放音频文件）

3.编码实现MP3音乐播放器。

**三、硬件连接图，软件流程图：**

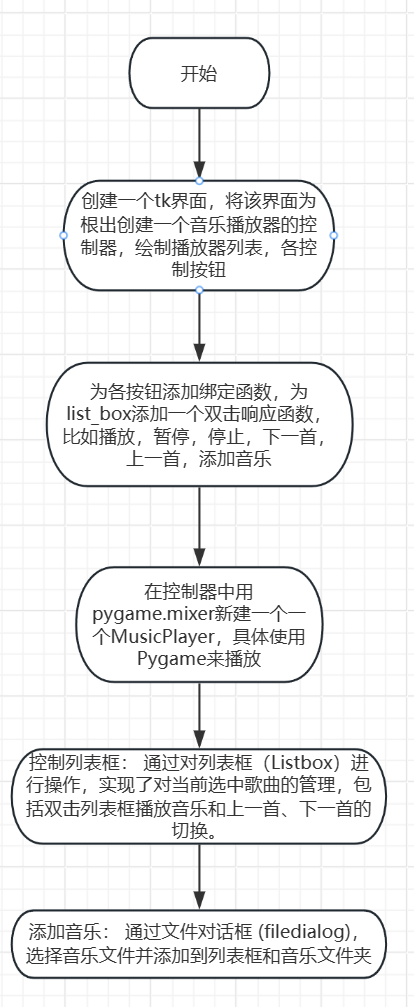


图1 软件流程图

**四、关键代码分析：**

MusicButtonControl.py

import tkinter as tk

from MusicPlay import MusicPlayer

from MusicList import MusicList

import os

from tkinter import filedialog

from shutil import copyfile

class MusicButtonControl:

def \_\_init\_\_(self, master, music\_folder):

#初始化 设置标题等

self.master = master

self.master.title("MP3 Music Player")

self.music\_list = MusicList(music\_folder)

self.music\_player = MusicPlayer()

self.label = tk.Label(master, text="播放中: ")

self.label.pack()

#设置播放列表

self.list\_box=tk.Listbox(master,width=30,height=30, bg="#FFFACD")

for item in self.music\_list.music\_files:

self.list\_box.insert(tk.END,item)

self.list\_box.bind("<Double-Button-1>", self.playMusic)

self.list\_box.pack(side=tk.LEFT,fill=tk.X)

#设置各个按钮，播放，下一首，上一首等

self.play\_button = tk.Button(master, text="播放", command=self.play\_music)

self.play\_button.pack(side=tk.LEFT)

#暂停按钮

self.pause\_button = tk.Button(master, text="暂停", command=self.pause\_music)

self.pause\_button.pack(side=tk.LEFT)

self.stop\_button = tk.Button(master, text="停止", command=self.stop\_music)

self.stop\_button.pack(side=tk.LEFT)

self.next\_button = tk.Button(master, text="下一首", command=self.next\_music)

self.next\_button.pack(side=tk.LEFT)

self.prev\_button = tk.Button(master, text="上一首", command=self.prev\_music)

self.prev\_button.pack(side=tk.LEFT)

self.add\_music =tk.Button(master,text="添加音乐",command=self.add\_music\_func)

self.add\_music.pack(side=tk.LEFT)

# 播放函数

def play\_music(self):

if self.music\_player.playing:#如果已经在播放，则点击按钮就是继续播放

self.music\_player.unpause\_music()

else:

file\_path = self.music\_list.get\_current\_music()#获取当前列表路径

self.music\_player.load\_music(file\_path)

self.music\_player.play\_music()#播放音乐

self.label["text"] = f"播放中: {os.path.basename(file\_path)}"

def pause\_music(self):#利用mixer暂停正在播放的音乐

self.music\_player.pause\_music()

def stop\_music(self):#停止播放的音乐

self.music\_player.stop\_music()

def next\_music(self):#下一首音乐，先stop音乐

self.music\_player.stop\_music()

file\_path = self.music\_list.next\_music()

self.music\_player.load\_music(file\_path)#在列表中获取下一首歌的地址

self.music\_player.play\_music()

self.label["text"] = f"播放中: {os.path.basename(file\_path)}"

cur=self.list\_box.curselection()

if cur:

selected\_index=cur[0]#更新listbox里面的选项

next\_index=(selected\_index+1)%self.list\_box.size()

self.list\_box.selection\_clear(selected\_index)

self.list\_box.selection\_set(next\_index)

self.list\_box.activate(next\_index)

def prev\_music(self):#上一首音乐

self.music\_player.stop\_music()

file\_path = self.music\_list.prev\_music()

self.music\_player.load\_music(file\_path)

self.music\_player.play\_music()#在列表中获取上一首歌的地址

self.label["text"] = f"播放中: {os.path.basename(file\_path)}"

cur = self.list\_box.curselection()

if cur:

selected\_index = cur[0]#更新listbox中选中的索引减一

previous\_index = (selected\_index - 1) % self.list\_box.size()

self.list\_box.selection\_clear(selected\_index)

self.list\_box.selection\_set( previous\_index)

self.list\_box.activate( previous\_index)

def playMusic(self,event):#播放音乐

self.music\_player.stop\_music()

index=self.list\_box.curselection()

file\_path = self.music\_list.get\_music(index[0])

self.music\_player.load\_music(file\_path)

self.music\_player.play\_music()

print(file\_path)

self.label["text"] = f"播放中: {os.path.basename(file\_path)}"

def add\_music\_func(self):#用filedialog选择文件名

file = filedialog.askopenfilename()

self.list\_box.insert(tk.END,os.path.basename(file))#在list\_box最后插入文件名

copyfile(file,'music/'+os.path.basename(file))

self.music\_list.music\_files.append(os.path.basename(file))

pass

**五、实验结果：**



图2 音乐播放器界面

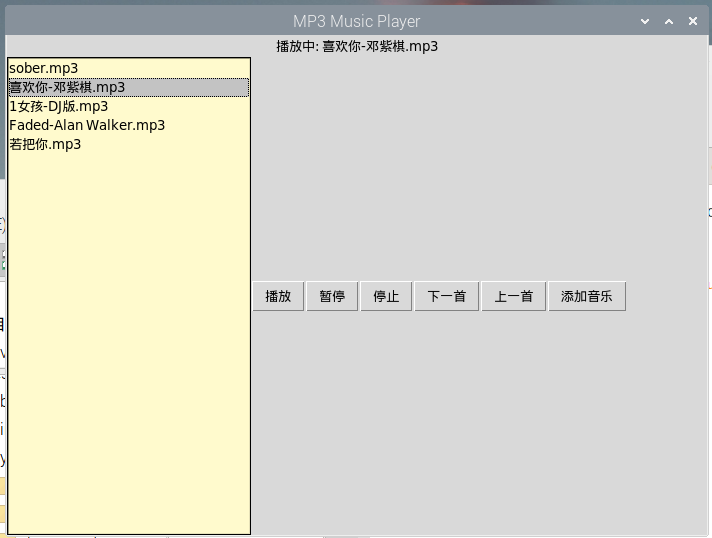


图3 双击列表中点击播放

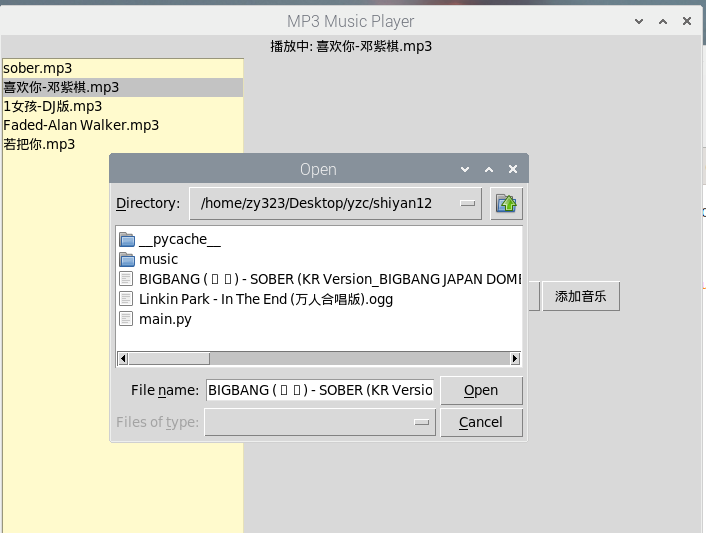


图4 播放音乐

如上图所示，左侧是音乐列表，右边的视图是多个按钮，各个按钮有着不同的功能。

**六、实验中遇到的问题及处理方法：**

碰到按下上一首和下一首按钮不能播放音乐的问题，仔细阅读代码后发现是逻辑错误，播放音乐后，self.playing是true，但是切换下一首后需要先停止播放先让该bool值变为false，才可以继续播放切换上一首或者下一首。

**七、实验中，自己的创意部分：**

图形用户界面 (GUI): 使用了 Tkinter 库创建了一个简单的图形用户界面，使用户可以通过按钮进行交互，而不是在终端使用命令行。

音乐播放功能: 整合了 Pygame 库，实现了基本的音乐播放功能，包括播放、暂停、停止、上一首、下一首。

音乐列表展示: 使用了 Tkinter 的 Listbox 组件，展示了当前播放列表的歌曲。

添加音乐功能: 提供了添加音乐的按钮，通过文件对话框选择音乐文件并添加到播放列表。

双击播放功能: 在 Listbox 上实现了双击歌曲名称即可播放的功能，提升了用户的操作便捷性。

界面布局: 界面简洁、清晰，使用户可以直观地理解和使用。

虽然这个音乐播放器相对简单，但是在教学和学习阶段，它提供了一个很好的例子，展示了如何使用 Python 的 Tkinter 和 Pygame 库构建一个基本的图形应用程序。

**八、实验中的收获：**

1.了解了树莓派音频模块的设置

2.熟悉了和掌握了 Tkinter 基础，并学会设计 Tkinter 界面

3.熟悉了和掌握了 Pygame 音频处理相关知识，并设计音频播放程序

4.结合了Tkinter 和 Pygame，设计了一个 MP3 播放器，提高综合能力