

课 程 设 计 报 告

课程名称： 微机原理

题 目： 炫动音乐播放器

班 级： 计算机科学与技术183

学 号：201805120513、201805120533

姓 名： 黄彤凯、朱泽华

**2020年1月**

目录

[一、程序源代码 4](#_Toc61974958)

[1.1 nasm代码源文件 4](#_Toc61974959)

[1.2 py源代码文件 41](#_Toc61974960)

[二、使用说明 44](#_Toc61974961)

[三、测试截图 45](#_Toc61974962)

[3.1 欢迎界面 45](#_Toc61974963)

[3.2 初始化完成 46](#_Toc61974964)

[3.3 两只老虎 47](#_Toc61974965)

[3.4 生日快乐 48](#_Toc61974966)

[3.5小星星 49](#_Toc61974967)

[四、思路分析 50](#_Toc61974968)

[4.1 丰富的颜色 50](#_Toc61974969)

[4.2 载入欢迎界面 50](#_Toc61974970)

[4.3 基本框线函数封装：music+外框 50](#_Toc61974971)

[4.4 闪动框线+定时器 50](#_Toc61974972)

[4.5 歌曲选择+键盘中断 51](#_Toc61974973)

[4.6 播放歌曲+扬声器 51](#_Toc61974974)

[4.7 延时的巧妙运用 51](#_Toc61974975)

[4.8 文字显示+py函数 51](#_Toc61974976)

[4.9 伪清屏技术 51](#_Toc61974977)

[五、代码截取片段分析 52](#_Toc61974978)

[5.1 封装及计算点函数 52](#_Toc61974979)

[5.2 写字的py函数以及调用 52](#_Toc61974980)

[5.3 横线以及竖线及画框函数 53](#_Toc61974981)

[5.4 扬声器以及歌曲 54](#_Toc61974982)

[5.5 键盘中断 55](#_Toc61974983)

[5.6 计时器中断（闪动效果） 56](#_Toc61974984)

[六、心得体会 57](#_Toc61974985)

[6.1 设计中比较完善的 57](#_Toc61974986)

[6.2 还希望改进的 58](#_Toc61974987)

图表录

[图表 1 欢迎界面 45](#_Toc61552147)

[图表 2 欢迎界面音效 45](#_Toc61552148)

[图表 3 初始化完成，双横线闪动（分别为赤橙黄绿青蓝紫） 46](#_Toc61552149)

[图表 4 初始化完成，双横线闪动（分别为赤橙黄绿青蓝紫）\_2 46](#_Toc61552150)

[图表 5 两只老虎界面 47](#_Toc61552151)

[图表 6 两只老虎部分音频 47](#_Toc61552152)

[图表 7 生日快乐界面 48](#_Toc61552153)

[图表 8 生日快乐部分音频 48](#_Toc61552154)

[图表 9 小星星界面 49](#_Toc61552155)

[图表 10 小星星部分音频 49](#_Toc61552156)

# 一、程序源代码

## 1.1 nasm代码源文件

; bx bp si di 放偏移地址 段基址：cs ds es ss

org 0x8400

jmp *start*

;======================设置声音======================

*cisound* dw 262;1

    dw 294;2

    dw 330;3

    dw 349;4

    dw 392;5

    dw 440;6

    dw 494;7

    ;上面为C大调的do re me fa sol la xi

;c大调 1231 1231 345 345 565431 565431 121 121

*liangzhilaohu* dw 262,294,330,262,262,294,330,262

              dw 330,349,392,330,349,392

              dw 392,440,392,349,330,262,392,440,392,349,330,262

              dw 262,294,262,262,294,262

;c大调 556517 556521 5553176 443121

*shengrikuaile* dw 392,392,440,392,262,494

              dw 392,392,440,392,294,262

              dw 392,392,392,330,262,494,440

              dw 349,349,330,262,294,262

;C大调 1155665 4433221 5544332 5544332 1155665 4433221

*xiaoxingxing*  dw 262,262,392,392,440,440,392

              dw 349,349,330,330,294,294,262

              dw 392,392,349,349,330,330,294

              dw 392,392,349,349,330,330,294

              dw 262,262,392,392,440,440,392

              dw 349,349,330,330,294,294,262

*huanyin* dw 440,494,262,494,262,330

*pl\_1* dw 0;设置频率的两个变量

*pl\_2* dw 0

;======================设置声音======================

;=========================设置画笔===================

;画笔颜色调整，调用col函数，调用前付给 si

*ts* db 0,255,255,255;背景色白

    db 1,220,140,160;方框调色紫色

    db 2,255,0,0;红色

    db 3,255,125,0;橙色

    db 4,255,255,0;黄色

    db 5,0,255,0;绿色

    db 6,0,255,255;青色

    db 7,0,0,255;蓝色

    db 8,0,0,0;黑色

;=========================设置画笔===================

*purplepen* db 1

*redpen* db 2

*bluepen* db 3

*finpen* db 0

*rdpen* db 2

      db 3

      db 4

      db 5

      db 6

      db 7

      db 1

;========================设置画笔====================

;================用于数据接入和传出===================

*x\_place* dw 0

*y\_place* dw 0

*xy\_place* dw 0

*xystart\_place* dw 0;线条封装

*xyend* dw 0;线条封装

*r1\_place* dw 0;方框封装左上

*r2\_place* dw 0;方框封装右上

*r3\_place* dw 0;方框封装左下

*r4\_place* dw 0;方框封装右下

;===============用于数据接入和传出====================

;===================数据开始=========================

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*框和横线数据

*x1\_place* dw 0

*y1\_place* dw 0

*xy1\_place* dw 0;待写入的坐标

*x2\_place* dw 319

*y2\_place* dw 0

*xy2\_place* dw 0

*x3\_place* dw 0

*y3\_place* dw 199

*xy3\_place* dw 0

*x4\_place* dw 319

*y4\_place* dw 199

*xy4\_place* dw 0

*x5\_place* dw 0

*y5\_place* dw 9

*xy5\_place* dw 0

*x6\_place* dw 319

*y6\_place* dw 9

*xy6\_place* dw 0

*x7\_place* dw 0

*y7\_place* dw 80

*xy7\_place* dw 0

*x8\_place* dw 319

*y8\_place* dw 80

*xy8\_place* dw 0

*x9\_place* dw 0

*y9\_place* dw 117

*xy9\_place* dw 0

*x10\_place* dw 319

*y10\_place* dw 117

*xy10\_place* dw 0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*music数据

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*m

*x11\_place* dw 50

*y11\_place* dw 30

*xy11\_place* dw 0

*x12\_place* dw 70

*y12\_place* dw 30

*xy12\_place* dw 0

*x13\_place* dw 90

*y13\_place* dw 30

*xy13\_place* dw 0

*x14\_place* dw 50

*y14\_place* dw 60

*xy14\_place* dw 0

*x15\_place* dw 70

*y15\_place* dw 60

*xy15\_place* dw 0

*x16\_place* dw 90

*y16\_place* dw 60

*xy16\_place* dw 0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*u

*x17\_place* dw 102

*y17\_place* dw 30

*xy17\_place* dw 0

*x18\_place* dw 132

*y18\_place* dw 30

*xy18\_place* dw 0

*x19\_place* dw 102

*y19\_place* dw 60

*xy19\_place* dw 0

*x20\_place* dw 132

*y20\_place* dw 60

*xy20\_place* dw 0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*s

*x21\_place* dw 152

*y21\_place* dw 30

*xy21\_place* dw 0

*x22\_place* dw 172

*y22\_place* dw 30

*xy22\_place* dw 0

*x23\_place* dw 152

*y23\_place* dw 45

*xy23\_place* dw 0

*x24\_place* dw 172

*y24\_place* dw 45

*xy24\_place* dw 0

*x25\_place* dw 152

*y25\_place* dw 60

*xy25\_place* dw 0

*x26\_place* dw 172

*y26\_place* dw 60

*xy26\_place* dw 0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*i

*x27\_place* dw 199

*y27\_place* dw 30

*xy27\_place* dw 0

*x28\_place* dw 202

*y28\_place* dw 30

*xy28\_place* dw 0

*x29\_place* dw 205

*y29\_place* dw 30

*xy29\_place* dw 0

*x30\_place* dw 196

*y30\_place* dw 60

*xy30\_place* dw 0

*x31\_place* dw 202

*y31\_place* dw 60

*xy31\_place* dw 0

*x32\_place* dw 208

*y32\_place* dw 60

*xy32\_place* dw 0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*c

*x33\_place* dw 231

*y33\_place* dw 30

*xy33\_place* dw 0

*x34\_place* dw 257

*y34\_place* dw 30

*xy34\_place* dw 0

*x35\_place* dw 231

*y35\_place* dw 60

*xy35\_place* dw 0

*x36\_place* dw 257

*y36\_place* dw 60

*xy36\_place* dw 0

;===================暂停键=========================

*x37\_place* dw 155

*y37\_place* dw 150

*xy37\_place* dw 0

*x38\_place* dw 155

*y38\_place* dw 166

*xy38\_place* dw 0

*x39\_place* dw 165

*y39\_place* dw 150

*xy39\_place* dw 0

*x40\_place* dw 165

*y40\_place* dw 166

*xy40\_place* dw 0

;===================暂停键=========================

;===================数据结束=========================

;=====================设置写字的点位=======================

*nameplace* dw 0

*namelength* dw 0

*name0* dw 137, 90, 154, 90, 185, 90, 137, 91, 154, 91, 163, 91, 169, 91, 170, 91, 175, 91

dw 185, 91, 137, 92, 146, 92, 147, 92, 148, 92, 149, 92, 150, 92, 151, 92, 154, 92, 164

dw 92, 167, 92, 168, 92, 171, 92, 172, 92, 173, 92, 174, 92, 175, 92, 176, 92, 185, 92

dw 137, 93, 151, 93, 154, 93, 155, 93, 156, 93, 157, 93, 158, 93, 159, 93, 165, 93, 167

dw 93, 171, 93, 175, 93, 185, 93, 137, 94, 147, 94, 151, 94, 153, 94, 159, 94, 167, 94

dw 171, 94, 175, 94, 185, 94, 137, 95, 147, 95, 151, 95, 152, 95, 155, 95, 158, 95, 167

dw 95, 171, 95, 175, 95, 185, 95, 137, 96, 148, 96, 150, 96, 155, 96, 162, 96, 163, 96

dw 164, 96, 165, 96, 167, 96, 171, 96, 175, 96, 185, 96, 137, 97, 148, 97, 150, 97, 155

dw 97, 165, 97, 167, 97, 170, 97, 171, 97, 175, 97, 185, 97, 137, 98, 149, 98, 155, 98

dw 165, 98, 167, 98, 169, 98, 171, 98, 175, 98, 185, 98, 137, 99, 148, 99, 150, 99, 155

dw 99, 165, 99, 167, 99, 168, 99, 171, 99, 173, 99, 175, 99, 185, 99, 137, 100, 148, 100

dw 151, 100, 154, 100, 156, 100, 165, 100, 167, 100, 171, 100, 174, 100, 185, 100, 137

dw 101, 147, 101, 151, 101, 154, 101, 156, 101, 165, 101, 171, 101, 185, 101, 137, 102

dw 146, 102, 153, 102, 157, 102, 165, 102, 171, 102, 185, 102, 137, 103, 153, 103, 158

dw 103, 164, 103, 166, 103, 171, 103, 175, 103, 176, 103, 185, 103, 137, 104, 152, 104

dw 158, 104, 159, 104, 160, 104, 163, 104, 167, 104, 168, 104, 169, 104, 170, 104, 171

dw 104, 172, 104, 173, 104, 174, 104, 175, 104, 185, 104, 137, 105, 150, 105, 151, 105

dw 159, 105, 185, 105

*name0l* dw 155

*name1* dw 117, 90, 139, 90, 164, 90, 181, 90, 184, 90, 197, 90, 117, 91, 126, 91, 127, 91

dw 128, 91, 129, 91, 130, 91, 131, 91, 132, 91, 133, 91, 134, 91, 135, 91, 136, 91, 137

dw 91, 138, 91, 139, 91, 140, 91, 153, 91, 164, 91, 181, 91, 182, 91, 183, 91, 184, 91

dw 185, 91, 197, 91, 117, 92, 131, 92, 135, 92, 145, 92, 146, 92, 147, 92, 148, 92, 149

dw 92, 150, 92, 151, 92, 152, 92, 153, 92, 154, 92, 164, 92, 166, 92, 170, 92, 181, 92

dw 197, 92, 117, 93, 131, 93, 135, 93, 139, 93, 145, 93, 153, 93, 160, 93, 161, 93, 162

dw 93, 163, 93, 164, 93, 165, 93, 166, 93, 167, 93, 169, 93, 170, 93, 176, 93, 177, 93

dw 178, 93, 179, 93, 180, 93, 181, 93, 182, 93, 183, 93, 184, 93, 185, 93, 186, 93, 187

dw 93, 197, 93, 117, 94, 127, 94, 128, 94, 129, 94, 130, 94, 131, 94, 132, 94, 133, 94

dw 134, 94, 135, 94, 136, 94, 137, 94, 138, 94, 139, 94, 140, 94, 145, 94, 153, 94, 164

dw 94, 168, 94, 176, 94, 180, 94, 187, 94, 197, 94, 117, 95, 127, 95, 131, 95, 135, 95

dw 139, 95, 145, 95, 153, 95, 164, 95, 167, 95, 171, 95, 176, 95, 180, 95, 183, 95, 184

dw 95, 186, 95, 197, 95, 117, 96, 127, 96, 131, 96, 135, 96, 139, 96, 145, 96, 153, 96

dw 158, 96, 159, 96, 160, 96, 161, 96, 162, 96, 163, 96, 164, 96, 165, 96, 166, 96, 167

dw 96, 168, 96, 169, 96, 170, 96, 171, 96, 172, 96, 176, 96, 177, 96, 178, 96, 179, 96

dw 180, 96, 181, 96, 182, 96, 197, 96, 117, 97, 127, 97, 131, 97, 135, 97, 139, 97, 145

dw 97, 153, 97, 165, 97, 176, 97, 180, 97, 186, 97, 197, 97, 117, 98, 127, 98, 131, 98

dw 135, 98, 139, 98, 145, 98, 146, 98, 147, 98, 148, 98, 149, 98, 150, 98, 151, 98, 152

dw 98, 153, 98, 164, 98, 176, 98, 181, 98, 182, 98, 183, 98, 184, 98, 185, 98, 186, 98

dw 197, 98, 117, 99, 127, 99, 131, 99, 135, 99, 139, 99, 145, 99, 153, 99, 163, 99, 169

dw 99, 176, 99, 183, 99, 197, 99, 117, 100, 127, 100, 130, 100, 132, 100, 134, 100, 136

dw 100, 139, 100, 162, 100, 163, 100, 168, 100, 169, 100, 176, 100, 179, 100, 180, 100

dw 181, 100, 182, 100, 183, 100, 184, 100, 197, 100, 117, 101, 127, 101, 129, 101, 133

dw 101, 137, 101, 138, 101, 139, 101, 147, 101, 151, 101, 161, 101, 163, 101, 166, 101

dw 167, 101, 176, 101, 179, 101, 183, 101, 197, 101, 117, 102, 127, 102, 128, 102, 132

dw 102, 138, 102, 139, 102, 146, 102, 147, 102, 152, 102, 160, 102, 163, 102, 164, 102

dw 165, 102, 176, 102, 179, 102, 183, 102, 197, 102, 117, 103, 127, 103, 139, 103, 145

dw 103, 153, 103, 154, 103, 159, 103, 163, 103, 171, 103, 176, 103, 179, 103, 183, 103

dw 187, 103, 197, 103, 117, 104, 127, 104, 137, 104, 139, 104, 144, 104, 154, 104, 155

dw 104, 158, 104, 163, 104, 171, 104, 175, 104, 178, 104, 183, 104, 187, 104, 197, 104

dw 117, 105, 127, 105, 138, 105, 143, 105, 155, 105, 164, 105, 165, 105, 166, 105, 167

dw 105, 168, 105, 169, 105, 170, 105, 171, 105, 174, 105, 177, 105, 184, 105, 185, 105

dw 186, 105, 187, 105, 197, 105

*name1l* dw 293

*name2* dw 117, 90, 133, 90, 153, 90, 161, 90, 166, 90, 184, 90, 197, 90, 117, 91, 129, 91

dw 133, 91, 145, 91, 146, 91, 147, 91, 148, 91, 149, 91, 150, 91, 151, 91, 152, 91, 153

dw 91, 154, 91, 161, 91, 166, 91, 182, 91, 183, 91, 184, 91, 185, 91, 197, 91, 117, 92

dw 129, 92, 133, 92, 145, 92, 153, 92, 161, 92, 166, 92, 177, 92, 178, 92, 179, 92, 180

dw 92, 181, 92, 197, 92, 117, 93, 129, 93, 133, 93, 138, 93, 145, 93, 153, 93, 161, 93

dw 166, 93, 170, 93, 177, 93, 197, 93, 117, 94, 128, 94, 129, 94, 130, 94, 131, 94, 132

dw 94, 133, 94, 134, 94, 135, 94, 136, 94, 137, 94, 138, 94, 139, 94, 145, 94, 153, 94

dw 159, 94, 161, 94, 162, 94, 164, 94, 165, 94, 166, 94, 167, 94, 168, 94, 169, 94, 170

dw 94, 171, 94, 177, 94, 181, 94, 197, 94, 117, 95, 128, 95, 133, 95, 145, 95, 153, 95

dw 159, 95, 161, 95, 163, 95, 166, 95, 170, 95, 177, 95, 181, 95, 197, 95, 117, 96, 127

dw 96, 133, 96, 145, 96, 153, 96, 159, 96, 161, 96, 166, 96, 170, 96, 176, 96, 181, 96

dw 187, 96, 197, 96, 117, 97, 127, 97, 133, 97, 145, 97, 146, 97, 147, 97, 148, 97, 149

dw 97, 150, 97, 151, 97, 152, 97, 153, 97, 158, 97, 161, 97, 166, 97, 170, 97, 175, 97

dw 176, 97, 177, 97, 178, 97, 179, 97, 180, 97, 181, 97, 182, 97, 183, 97, 184, 97, 185

dw 97, 186, 97, 187, 97, 188, 97, 197, 97, 117, 98, 126, 98, 133, 98, 137, 98, 145, 98

dw 153, 98, 161, 98, 166, 98, 170, 98, 181, 98, 197, 98, 117, 99, 128, 99, 129, 99, 130

dw 99, 131, 99, 132, 99, 133, 99, 134, 99, 135, 99, 136, 99, 137, 99, 138, 99, 145, 99

dw 153, 99, 161, 99, 162, 99, 163, 99, 164, 99, 165, 99, 166, 99, 167, 99, 168, 99, 169

dw 99, 170, 99, 171, 99, 172, 99, 181, 99, 197, 99, 117, 100, 133, 100, 145, 100, 153

dw 100, 161, 100, 166, 100, 178, 100, 181, 100, 184, 100, 197, 100, 117, 101, 133, 101

dw 145, 101, 153, 101, 161, 101, 165, 101, 167, 101, 178, 101, 181, 101, 185, 101, 197

dw 101, 117, 102, 133, 102, 145, 102, 153, 102, 161, 102, 165, 102, 168, 102, 177, 102

dw 181, 102, 186, 102, 197, 102, 117, 103, 133, 103, 139, 103, 145, 103, 146, 103, 147

dw 103, 148, 103, 149, 103, 150, 103, 151, 103, 152, 103, 153, 103, 161, 103, 164, 103

dw 169, 103, 176, 103, 181, 103, 186, 103, 187, 103, 197, 103, 117, 104, 126, 104, 127

dw 104, 128, 104, 129, 104, 130, 104, 131, 104, 132, 104, 133, 104, 134, 104, 135, 104

dw 136, 104, 137, 104, 138, 104, 139, 104, 140, 104, 145, 104, 153, 104, 161, 104, 163

dw 104, 170, 104, 171, 104, 172, 104, 175, 104, 179, 104, 181, 104, 187, 104, 197, 104

dw 117, 105, 161, 105, 162, 105, 171, 105, 180, 105, 197, 105

*name2l* dw 260

;262

*name3* dw 127, 90, 143, 90, 164, 90, 180, 90, 191, 90, 127, 91, 143, 91, 154, 91, 155, 91

dw 156, 91, 157, 91, 158, 91, 159, 91, 160, 91, 161, 91, 162, 91, 163, 91, 164, 91, 165

dw 91, 170, 91, 171, 91, 172, 91, 173, 91, 174, 91, 175, 91, 176, 91, 177, 91, 178, 91

dw 179, 91, 180, 91, 181, 91, 191, 91, 127, 92, 143, 92, 154, 92, 164, 92, 170, 92, 180

dw 92, 191, 92, 127, 93, 143, 93, 154, 93, 155, 93, 156, 93, 157, 93, 158, 93, 159, 93

dw 160, 93, 161, 93, 162, 93, 163, 93, 164, 93, 170, 93, 171, 93, 172, 93, 173, 93, 174

dw 93, 175, 93, 176, 93, 177, 93, 178, 93, 179, 93, 180, 93, 191, 93, 127, 94, 143, 94

dw 154, 94, 164, 94, 170, 94, 180, 94, 191, 94, 127, 95, 141, 95, 143, 95, 145, 95, 154

dw 95, 155, 95, 156, 95, 157, 95, 158, 95, 159, 95, 160, 95, 161, 95, 162, 95, 163, 95

dw 164, 95, 170, 95, 171, 95, 172, 95, 173, 95, 174, 95, 175, 95, 176, 95, 177, 95, 178

dw 95, 179, 95, 180, 95, 191, 95, 127, 96, 141, 96, 143, 96, 146, 96, 159, 96, 175, 96

dw 191, 96, 127, 97, 140, 97, 143, 97, 147, 97, 154, 97, 159, 97, 164, 97, 170, 97, 175

dw 97, 180, 97, 191, 97, 127, 98, 140, 98, 143, 98, 148, 98, 154, 98, 155, 98, 156, 98

dw 157, 98, 158, 98, 159, 98, 160, 98, 161, 98, 162, 98, 163, 98, 164, 98, 165, 98, 170

dw 98, 171, 98, 172, 98, 173, 98, 174, 98, 175, 98, 176, 98, 177, 98, 178, 98, 179, 98

dw 180, 98, 181, 98, 191, 98, 127, 99, 139, 99, 143, 99, 149, 99, 154, 99, 159, 99, 170

dw 99, 175, 99, 191, 99, 127, 100, 138, 100, 143, 100, 149, 100, 153, 100, 159, 100, 163

dw 100, 169, 100, 175, 100, 179, 100, 191, 100, 127, 101, 137, 101, 143, 101, 152, 101

dw 154, 101, 155, 101, 156, 101, 157, 101, 158, 101, 159, 101, 160, 101, 161, 101, 162

dw 101, 163, 101, 164, 101, 168, 101, 170, 101, 171, 101, 172, 101, 173, 101, 174, 101

dw 175, 101, 176, 101, 177, 101, 178, 101, 179, 101, 180, 101, 191, 101, 127, 102, 143

dw 102, 159, 102, 175, 102, 191, 102, 127, 103, 143, 103, 159, 103, 165, 103, 175, 103

dw 181, 103, 191, 103, 127, 104, 141, 104, 143, 104, 152, 104, 153, 104, 154, 104, 155

dw 104, 156, 104, 157, 104, 158, 104, 159, 104, 160, 104, 161, 104, 162, 104, 163, 104

dw 164, 104, 165, 104, 166, 104, 168, 104, 169, 104, 170, 104, 171, 104, 172, 104, 173

dw 104, 174, 104, 175, 104, 176, 104, 177, 104, 178, 104, 179, 104, 180, 104, 181, 104

dw 182, 104, 191, 104, 127, 105, 142, 105, 191, 105

;242

*name3l* dw 240

*name4* dw 76, 153, 79, 153, 76, 154, 79, 154, 75, 155, 78, 155, 75, 156, 78, 156, 74, 157

      dw 77, 157, 74, 158, 77, 158, 75, 159, 78, 159, 75, 160, 78, 160, 76, 161, 79, 161

      dw 76, 162, 79, 162

;20

*name4l* dw 20

*name5* dw 161, 150, 177, 150, 161, 151, 177, 151, 161, 152, 177, 152, 161, 153, 177, 153

      dw 161, 154, 177, 154, 161, 155, 177, 155, 161, 156, 177, 156, 161, 157, 177, 157

      dw 161, 158, 177, 158, 161, 159, 177, 159, 161, 160, 177, 160, 161, 161, 177, 161

      dw 161, 162, 177, 162, 161, 163, 177, 163, 161, 164, 177, 164, 161, 165, 177, 165

*name5l* dw 30

*name6* dw 244, 153, 247, 153, 244, 154, 247, 154, 245, 155, 248, 155, 245, 156, 248, 156

      dw 246, 157, 249, 157, 246, 158, 249, 158, 245, 159, 248, 159, 245, 160, 248, 160

      dw 244, 161, 247, 161, 244, 162, 247, 162

*name6l* dw 20

;=====================设置写字的点位=======================

*start:*

;===============主函数图形模式开始=======================

;进入13h号320\*200 256色的图形模式

    mov ah,00h

    mov al,13h

    int 10h

    mov ax,0xa000 ;往显存中放入点数据

    mov es,ax

;=================主函数开始=========================

    mov si,*ts*

    ;设置白色背景。

    call *col*

    ;其他调色

    call *col*

    call *col*

    call *col*

    call *col*

    call *col*

    call *col*

    call *col*

    call *col*

;=================写字==============================

    push bx

    mov bx,2

    mov [*finpen*],bx

    pop bx

    mov si,*name0*

    push bx

    mov bx,[*name0l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

;=================写字==============================

;=================算xy点位==============================

    ;第一个点计算

    mov bx,[*x1\_place*]

    mov [*x\_place*],bx

    mov bx,[*y1\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy1\_place*],bx

    ;第二个点计算

    mov bx,[*x2\_place*]

    mov [*x\_place*],bx

    mov bx,[*y2\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy2\_place*],bx

    ;第三个点计算

    mov bx,[*x3\_place*]

    mov [*x\_place*],bx

    mov bx,[*y3\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy3\_place*],bx

    ;第四个点计算

    mov bx,[*x4\_place*]

    mov [*x\_place*],bx

    mov bx,[*y4\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy4\_place*],bx

    ;第五个点计算

    mov bx,[*x5\_place*]

    mov [*x\_place*],bx

    mov bx,[*y5\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy5\_place*],bx

    ;第六个点计算

    mov bx,[*x6\_place*]

    mov [*x\_place*],bx

    mov bx,[*y6\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy6\_place*],bx

    ;第七个点计算

    mov bx,[*x7\_place*]

    mov [*x\_place*],bx

    mov bx,[*y7\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy7\_place*],bx

    ;第八个点计算

    mov bx,[*x8\_place*]

    mov [*x\_place*],bx

    mov bx,[*y8\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy8\_place*],bx

    ;第九个点计算

    mov bx,[*x9\_place*]

    mov [*x\_place*],bx

    mov bx,[*y9\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy9\_place*],bx

    ;第十个点计算

    mov bx,[*x10\_place*]

    mov [*x\_place*],bx

    mov bx,[*y10\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy10\_place*],bx

    ;第十一个点计算

    mov bx,[*x11\_place*]

    mov [*x\_place*],bx

    mov bx,[*y11\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy11\_place*],bx

    ;第十二个点计算

    mov bx,[*x12\_place*]

    mov [*x\_place*],bx

    mov bx,[*y12\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy12\_place*],bx

    ;第十三个点计算

    mov bx,[*x13\_place*]

    mov [*x\_place*],bx

    mov bx,[*y13\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy13\_place*],bx

    ;第十四个点计算

    mov bx,[*x14\_place*]

    mov [*x\_place*],bx

    mov bx,[*y14\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy14\_place*],bx

    ;第十五个点计算

    mov bx,[*x15\_place*]

    mov [*x\_place*],bx

    mov bx,[*y15\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy15\_place*],bx

    ;第十六个点计算

    mov bx,[*x16\_place*]

    mov [*x\_place*],bx

    mov bx,[*y16\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy16\_place*],bx

        ;第十7个点计算

    mov bx,[*x17\_place*]

    mov [*x\_place*],bx

    mov bx,[*y17\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy17\_place*],bx

        ;第十8个点计算

    mov bx,[*x18\_place*]

    mov [*x\_place*],bx

    mov bx,[*y18\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy18\_place*],bx

        ;第十9个点计算

    mov bx,[*x19\_place*]

    mov [*x\_place*],bx

    mov bx,[*y19\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy19\_place*],bx

        ;第20个点计算

    mov bx,[*x20\_place*]

    mov [*x\_place*],bx

    mov bx,[*y20\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy20\_place*],bx

            ;第21个点计算

    mov bx,[*x21\_place*]

    mov [*x\_place*],bx

    mov bx,[*y21\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy21\_place*],bx

            ;第22个点计算

    mov bx,[*x22\_place*]

    mov [*x\_place*],bx

    mov bx,[*y22\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy22\_place*],bx

            ;第23个点计算

    mov bx,[*x23\_place*]

    mov [*x\_place*],bx

    mov bx,[*y23\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy23\_place*],bx

            ;第24个点计算

    mov bx,[*x24\_place*]

    mov [*x\_place*],bx

    mov bx,[*y24\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy24\_place*],bx

            ;第25个点计算

    mov bx,[*x25\_place*]

    mov [*x\_place*],bx

    mov bx,[*y25\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy25\_place*],bx

            ;第26个点计算

    mov bx,[*x26\_place*]

    mov [*x\_place*],bx

    mov bx,[*y26\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy26\_place*],bx

                ;第27个点计算

    mov bx,[*x27\_place*]

    mov [*x\_place*],bx

    mov bx,[*y27\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy27\_place*],bx

                ;第28个点计算

    mov bx,[*x28\_place*]

    mov [*x\_place*],bx

    mov bx,[*y28\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy28\_place*],bx

                ;第29个点计算

    mov bx,[*x29\_place*]

    mov [*x\_place*],bx

    mov bx,[*y29\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy29\_place*],bx

                ;第30个点计算

    mov bx,[*x30\_place*]

    mov [*x\_place*],bx

    mov bx,[*y30\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy30\_place*],bx

                ;第31个点计算

    mov bx,[*x31\_place*]

    mov [*x\_place*],bx

    mov bx,[*y31\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy31\_place*],bx

                ;第32个点计算

    mov bx,[*x32\_place*]

    mov [*x\_place*],bx

    mov bx,[*y32\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy32\_place*],bx

                    ;第33个点计算

    mov bx,[*x33\_place*]

    mov [*x\_place*],bx

    mov bx,[*y33\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy33\_place*],bx

                ;第34个点计算

    mov bx,[*x34\_place*]

    mov [*x\_place*],bx

    mov bx,[*y34\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy34\_place*],bx

                ;第35个点计算

    mov bx,[*x35\_place*]

    mov [*x\_place*],bx

    mov bx,[*y35\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy35\_place*],bx

                ;第36个点计算

    mov bx,[*x36\_place*]

    mov [*x\_place*],bx

    mov bx,[*y36\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy36\_place*],bx

                ;第37个点计算

    mov bx,[*x37\_place*]

    mov [*x\_place*],bx

    mov bx,[*y37\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy37\_place*],bx

                ;第38个点计算

    mov bx,[*x38\_place*]

    mov [*x\_place*],bx

    mov bx,[*y38\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy38\_place*],bx

                    ;第39个点计算

    mov bx,[*x39\_place*]

    mov [*x\_place*],bx

    mov bx,[*y39\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy39\_place*],bx

                ;第40个点计算

    mov bx,[*x40\_place*]

    mov [*x\_place*],bx

    mov bx,[*y40\_place*]

    mov [*y\_place*],bx

    call *countxy*

    mov bx,[*xy\_place*]

    mov [*xy40\_place*],bx

;=================算xy点位==============================

;=================画框==============================

    ;选择颜色

    mov bx,[*redpen*]

    mov [*finpen*],bx

    ;给方框赋值点位

    mov bx,[*xy1\_place*]

    mov [*r1\_place*],bx

    mov bx,[*xy2\_place*]

    mov [*r2\_place*],bx

    mov bx,[*xy3\_place*]

    mov [*r3\_place*],bx

    mov bx,[*xy4\_place*]

    mov [*r4\_place*],bx

    call *rectangle*

    ;画横线

    mov bx,[*redpen*]

    mov [*finpen*],bx

    mov bx,[*xy5\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy6\_place*]

    mov [*xyend*],bx

    call *line*

;=================画框=================================

;=================music字==============================

    ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy11\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy13\_place*]

    mov [*xyend*],bx

    call *line*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy11\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy14\_place*]

    mov [*xyend*],bx

    call *row*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy11\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy14\_place*]

    mov [*xyend*],bx

    call *row*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy12\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy15\_place*]

    mov [*xyend*],bx

    call *row*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy13\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy16\_place*]

    mov [*xyend*],bx

    call *row*

     ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy19\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy20\_place*]

    mov [*xyend*],bx

    call *line*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy17\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy19\_place*]

    mov [*xyend*],bx

    call *row*

    ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy18\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy20\_place*]

    mov [*xyend*],bx

    call *row*

         ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy21\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy22\_place*]

    mov [*xyend*],bx

    call *line*

         ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy23\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy24\_place*]

    mov [*xyend*],bx

    call *line*

         ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy25\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy26\_place*]

    mov [*xyend*],bx

    call *line*

        ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy21\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy23\_place*]

    mov [*xyend*],bx

    call *row*

        ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy24\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy25\_place*]

    mov [*xyend*],bx

    call *row*

             ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy27\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy29\_place*]

    mov [*xyend*],bx

    call *line*

             ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy30\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy32\_place*]

    mov [*xyend*],bx

    call *line*

            ;画竖线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy28\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy31\_place*]

    mov [*xyend*],bx

    call *row*

                 ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy33\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy34\_place*]

    mov [*xyend*],bx

    call *line*

             ;画横线

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy35\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy36\_place*]

    mov [*xyend*],bx

    call *line*

            ;画竖线

    mov bx,[*rdpen*+5 ]

    mov [*finpen*],bx

    mov bx,[*xy33\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy35\_place*]

    mov [*xyend*],bx

    call *row*

;=================开机音效==============================

    call *outhuanyin*

    call *sleeploop3times*

    call *sleeploop3times*

;=================开机音效==============================

    sti

;=================定时器==============================

    mov bp,1

    mov word [ds:0x20],*int\_time0*

    mov word [ds:0x22],0

;=================定时器==============================

;=================声音==============================

    mov word [ds:0x24],*int\_key* ;中断地址9

    mov word [ds:0x26],0

;=================声音==============================

;=================暂停线==============================

            ;画竖线

    mov bx,8

    mov [*finpen*],bx

    mov bx,[*xy37\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy38\_place*]

    mov [*xyend*],bx

    call *row*

                ;画竖线

    mov bx,8

    mov [*finpen*],bx

    mov bx,[*xy39\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy40\_place*]

    mov [*xyend*],bx

    call *row*

;=================暂停线==============================

;=================播放按键==============================

    push bx

    mov bx,8

    mov [*finpen*],bx

    pop bx

    mov si,*name4*

    push bx

    mov bx,[*name4l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    ; push bx

    ; mov bx,2

    ; mov [finpen],bx

    ; pop bx

    ; mov si,name5

    ; push bx

    ; mov bx,[name5l]

    ; mov [namelength],bx

    ; pop bx

    ; call outpoint

    push bx

    mov bx,8

    mov [*finpen*],bx

    pop bx

    mov si,*name6*

    push bx

    mov bx,[*name6l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

;=================播放按键==============================

    jmp $

;=================主函数结束=========================

;画横线==============================================

;bp 起始位置 di 终止位置 cl色号

*line:*

    push bp

    push di

    mov bp,[*xystart\_place*]

    mov di,[*xyend*]

    mov cl,[*finpen*]

*linepart:*

    mov byte [es:bp],cl

    add bp,1

    cmp bp,di

    jbe *linepart*

    pop di

    pop bp

    ret

;画横线===============================================

;画竖线===============================================

;bp 起始位置 di 终止位置 cl色号

*row:*

    push bp

    push di

    mov bp,[*xystart\_place*]

    mov di,[*xyend*]

    mov cl,[*finpen*]

*rowpart:*

    mov byte [es:bp],cl

    add bp,320

    cmp bp,di

    jbe *rowpart*

    pop di

    pop bp

    ret

;画竖线===============================================

;画斜线，bl色号，bp起始位置，cx终点位置,坐标算死在调用,left 多加一个320

*rightslash:*

    mov byte [es:bp],bl

    add bp,321

    cmp bp,cx

    jbe *rightslash*

    ret

*leftslash:*

    mov byte [es:bp],bl

    add bp,319

    cmp bp,cx

    jbe *leftslash*

    ret

;=================计算坐标函数=========================

;计算坐标，输入值是x\_place,y\_place,写回xy\_place

;使用： mov bx,[数据一] mov [x\_place],bx  mov bx,[数据二] mov [y\_place],bx call即可

;使用完将 [xy\_place]写到bx 里面，再将bx 写回到数据三中，调用数据三即可调用坐标

*countxy:*

    push ax

    push bx

    mov ax,[*y\_place*]

    mov bx,320

    mul bx

    mov [*xy\_place*],ax

    mov bx,[*x\_place*]

    add [*xy\_place*],bx

    pop bx

    pop ax

    ret

;=================计算坐标函数=========================

;设置颜色，先mov si,ts直接重复调用即可，色号按照如下方式给出

; ts  db 0,255,255,255;背景色白

;     db 1,220,140,160;方框调色

;     db 2,255,0,0;

;     db 3,0,0,255

;=================颜色函数=========================

*col:*

    push dx

    mov dx,0x3c8

    mov al,[si]

    out dx,al

    mov dx,0x3c9

    mov al,[si+1]

    out dx,al

    mov al,[si+2]

    out dx,al

    mov al,[si+3]

    out dx,al

    add si,4

    pop dx

    ret

;=================颜色函数=========================

;=================画方框函数========================

;传入 r1 r2 r3 r4 四个点

*rectangle:*

    ;上，调用横线函数

    push bp

    push di

    mov bp,[*r1\_place*]

    mov di,[*r2\_place*]

    mov cl,[*finpen*]

*lp1:*

    mov byte [es:bp],cl

    add bp,1

    cmp bp,di

    jbe *lp1*

    ;下，调用横线函数

    mov bp,[*r3\_place*]

    mov di,[*r4\_place*]

    mov cl,[*finpen*]

*lp2:*

    mov byte [es:bp],cl

    add bp,1

    cmp bp,di

    jbe *lp2*

    ;左，调用竖线函数

    mov bp,[*r1\_place*]

    mov di,[*r3\_place*]

    mov cl,[*finpen*]

*lp3:*

    mov byte [es:bp],cl

    add bp,320

    cmp bp,di

    jbe *lp3*

    ;右，调用竖线函数

    mov bp,[*r2\_place*]

    mov di,[*r4\_place*]

    mov cl,[*finpen*]

*lp4:*

    mov byte [es:bp],cl

    add bp,320

    cmp bp,di

    jbe *lp4*

    pop di

    pop bp

    ret

;=================画方框函数========================

;===================写字函数=======================

*outpoint:*

    push bx

    push cx

    push ax

    mov bx,0

*partout:*

    mov ax,[si+2]

    mov cx,320

    mul cx

    mov [*nameplace*],ax

    mov cx,[si]

    add [*nameplace*],cx

    mov bp,[*nameplace*]

    push ax

    mov al,[*finpen*]

    mov byte [es:bp],al

    pop ax

    add bx,1

    add si,4

    cmp bx,[*namelength*];点位大小

    jb *partout*

    pop ax

    pop cx

    pop bx

    ret

;===================写字函数=======================

;===================声音的函数=====================

*int\_key:*

    mov dx,0x60

    in al,dx

    ;判断输入的是哪个按键，跳转

    ;开始判断下一个按键

    ;a

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    cmp al,0x1e

    je *outa*

    ;b

    cmp al,0x30

    je *outb*

    ;c

    cmp al,0x2e

    je *outc*

    cmp al,0x20

    je *outd*

    jmp *k1*

    ;因为代码都一样这里就用a b c d

*outa:*

;=================写字==============================

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name0*

    push bx

    mov bx,[*name0l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name2*

    push bx

    mov bx,[*name2l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name3*

    push bx

    mov bx,[*name3l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,5

    mov [*finpen*],bx

    pop bx

    mov si,*name1*

    push bx

    mov bx,[*name1l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

;=================写字==============================

    call *closes*

    call *outliangzhilaohu*

    jmp *k1*

*outb:*

;=================写字==============================

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name0*

    push bx

    mov bx,[*name0l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name1*

    push bx

    mov bx,[*name1l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name3*

    push bx

    mov bx,[*name3l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,6

    mov [*finpen*],bx

    pop bx

    mov si,*name2*

    push bx

    mov bx,[*name2l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

;=================写字==============================

    call *closes*

    call *outshengrikuaile*

    jmp *k1*

*outc:*

;=================写字==============================

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name0*

    push bx

    mov bx,[*name0l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name2*

    push bx

    mov bx,[*name2l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name1*

    push bx

    mov bx,[*name1l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

    push bx

    mov bx,1

    mov [*finpen*],bx

    pop bx

    mov si,*name3*

    push bx

    mov bx,[*name3l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

;=================写字==============================

    call *closes*

    call *outxiaoxingxing*

    jmp *k1*

*outd:*

    call *closes*

    jmp *k1*

*k1:*

    mov dx,0x20 ;键盘中断结束

    mov al,0x61

    out dx,al

    mov word [ds:0x20],*int\_time0*

    mov word [ds:0x22],0

iret

*outt:*;初始化扬声器

    mov [*pl\_1*],bx ;把do re。。。的频率先给

    mov al,*10110110B*

    out 43h,al

    push ax

    push dx

    push di

    mov dx,12h

    mov ax,32edh

    mov di,[*pl\_1*]

    div di

    mov [*pl\_2*],ax

    pop di

    pop dx

    pop ax

    mov ax, [*pl\_2*] ;计数初值=1.19 MHz, 600 Hz=1983

    out 42H, al ;送计数初值低位字节，8253的计数器2的地址42H

    mov al, ah

    out 42H, al ;送计数初值高位字节

    call *opens*

    call *sleep\_0point4sec*

    call *closes*

    ret

;延迟1s

*sleep\_1point0sec:*

    push cx

    push ax

    push dx

    mov  cx, 0FH ;延时高16位

    mov  dx, 4240H ;延时低16位 1s=1000 000us=这个

    mov  ah, 86H ;调用功能号，格式化驱动器定期中断

    int  15H

    pop dx

    pop ax

    pop cx

ret

*sleep\_0point4sec:*

    push cx

    push ax

    push dx

    mov  cx, 06H ;延时高16位

    mov  dx, 4240H ;延时低16位 1s=1000 000us=这个

    mov  ah, 86H ;调用功能号，格式化驱动器定期中断

    int  15H

    pop dx

    pop ax

    pop cx

ret

*sleeploop3times:*

    call *sleep\_1point0sec*

    call *sleep\_1point0sec*

    call *sleep\_1point0sec*

ret

*sleep\_wl:*

    push cx

    push ax

    mov ax,0fffh

*x11:*    mov cx,100

*x22:*    loop *x22*

    dec ax

    jnz *x11*

    pop ax

    pop cx

    ret

*opens:*

    ;打开扬声器

    push ax

    in al,61h

    or al,*00000011B*

    out 61h,al

    pop ax

    ret

*closes:*

    ;关闭扬声器

    push ax

    in al,61h

    and al,*11111100B*

    out 61h,al

    pop ax

    ret

*outliangzhilaohu:*

    push bx

    push di

    mov di,0

*pratoutliangzhilaohu:*

    mov bx,[*liangzhilaohu*+di]

    call *sleeploop3times*

    call *outt*

    add di,2

    cmp di,62

    jbe *pratoutliangzhilaohu*

    pop di

    pop bx

    ret

*outhuanyin:*

    push bx

    push di

    mov di,0

*pratouthuanyin:*

    mov bx,[*huanyin*+di]

    call *sleep\_1point0sec*

    call *outt*

    add di,2

    cmp di,10

    jbe *pratouthuanyin*

    pop di

    pop bx

    ret

*outshengrikuaile:*

    push bx

    push di

    mov di,0

*pratoutshengrikuaile:*

    mov bx,[*shengrikuaile*+di]

    call *sleeploop3times*

    call *outt*

    add di,2

    cmp di,48

    jbe *pratoutshengrikuaile*

    pop di

    pop bx

    ret

*outxiaoxingxing:*

    push bx

    push di

    mov di,0

*pratoutxiaoxingxing:*

    mov bx,[*xiaoxingxing*+di]

    call *sleeploop3times*

    call *outt*

    add di,2

    cmp di,82

    jbe *pratoutxiaoxingxing*

    pop di

    pop bx

    ret

;===================声音的函数=====================

;===================计时器=====================

*int\_time0:*

    push bx

    ;画横线

    inc bp

*xxx:*    cmp bp,7

    ja *decbp*

    mov bx,[*rdpen*+bp]

    mov [*finpen*],bx

     mov bx,[*xy7\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy8\_place*]

    mov [*xyend*],bx

    call *line*

     mov bx,[*xy9\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy10\_place*]

    mov [*xyend*],bx

    call *line*

    pop bx

    mov al,0x20

    mov dx,0x20

    out dx,al   ;EOI=0,中断结束

    iret

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

*decbp:*

    mov bp,1

    jmp *xxx*

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

;===================注意这里有无条件跳转=========

*int\_time1:*

    mov al,0x20

    mov dx,0x20

    out dx,al   ;EOI=0,中断结束

    iret

*sleep\_1sec:*

    push cx

    push ax

    push dx

    mov  cx, 98H ;延时高16位

    mov  dx, 9680H ;延时低16位 1s=1000 000us=这个

    mov  ah, 86H ;调用功能号，格式化驱动器定期中断

    int  15H

    pop dx

    pop ax

    pop cx

ret

;===================计时器=====================

## 1.2 py源代码文件

# This is a sample Python script.  
  
# Press Shift+F10 to execute it or replace it with your code.  
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.  
  
  
def print\_hi(name):  
 # Use a breakpoint in the code line below to debug your script.  
 print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.  
  
  
# Press the green button in the gutter to run the script.  
if \_\_name\_\_ == '\_\_main\_\_':  
 print\_hi('PyCharm')  
  
# See PyCharm help at https://www.jetbrains.com/help/pycharm/  
import binascii  
  
KEYS = [0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01]  
pointGroup = []  
  
def printPlay(textStr, line, background):  
 # 初始化16\*16的点阵位置，每个汉字需要16\*16=256个点来表示，需要32个字节才能显示一个汉字  
 # 之所以32字节：256个点每个点是0或1，那么总共就是2的256次方，一个字节是2的8次方  
 rect\_list = [] \* 16  
  
  
 for i in range(16):  
 rect\_list.append([] \* 16)  
  
 for text in textStr:  
 # 获取中文的gb2312编码，一个汉字是由2个字节编码组成  
 gb2312 = text.encode('gb2312')  
 # 将二进制编码数据转化为十六进制数据  
 hex\_str = binascii.b2a\_hex(gb2312)  
 # 将数据按unicode转化为字符串  
 result = str(hex\_str, encoding='utf-8')  
  
 # 前两位对应汉字的第一个字节：区码，每一区记录94个字符  
 area = eval('0x' + result[:2]) - 0xA0  
 # 后两位对应汉字的第二个字节：位码，是汉字在其区的位置  
 index = eval('0x' + result[2:]) - 0xA0  
 # 汉字在HZK16中的绝对偏移位置，最后乘32是因为字库中的每个汉字字模都需要32字节  
 offset = (94 \* (area - 1) + (index - 1)) \* 32  
  
 font\_rect = None  
  
 # 读取HZK16汉字库文件  
 with open("HZK16", "rb") as f:  
 # 找到目标汉字的偏移位置  
 f.seek(offset)  
 # 从该字模数据中读取32字节数据  
 font\_rect = f.read(32)  
  
 # font\_rect的长度是32，此处相当于for k in range(16)  
 for k in range(len(font\_rect) // 2):  
 # 每行数据  
 row\_list = rect\_list[k]  
 for j in range(2):  
 for i in range(8):  
 asc = font\_rect[k \* 2 + j]  
 # 此处&为Python中的按位与运算符  
 flag = asc & KEYS[i]  
 # 数据规则获取字模中数据添加到16行每行中16个位置处每个位置  
 row\_list.append(flag)  
 point\_y = 0  
 pointCont = 0  
 # 根据获取到的16\*16点阵信息，打印到控制台  
 for row in rect\_list:  
 point\_x = 0  
 for i in row:  
 if i:  
 # 前景字符（即用来表示汉字笔画的输出字符）  
 print(line, end=' ')  
 pointGroup.append(point\_x+238)  
 pointGroup.append(point\_y+150)  
 pointCont = pointCont + 1  
 else:  
 # 背景字符（即用来表示背景的输出字符）  
 print(background, end=' ')  
 point\_x = point\_x + 1  
 point\_y = point\_y + 1  
 print()  
  
 print("点的数目：")  
 print(pointCont)  
 print("点集：")  
 print(pointGroup)  
  
  
# ----------------------------以上是库的引用和函数定义，下面是代码正文----------------------  
# 允许用户自定义输入汉字短语  
inpt = input("写你所想：")  
  
# 自定义点阵字中笔画的符号  
lineSign = '\*'  
# 备选方案  
# lineSign = "0"  
  
# 自定义点阵字的背景符号  
backgroundSign = ' '  
# 备选方案  
# backgroundSign = "\*"  
  
# 调用之前定义好的函数，打印最终成果  
printPlay(inpt, lineSign, backgroundSign)

# 二、使用说明

打开本系统以后，请先等待加载完成，加载界面有欢迎两字，并且有3秒的载入音效，载入完成，双横线开始闪动，进入选择歌曲界面。

通过调用键盘中断实现换歌，具体实现请见倒数第二章。

请按键，a：两只老虎；

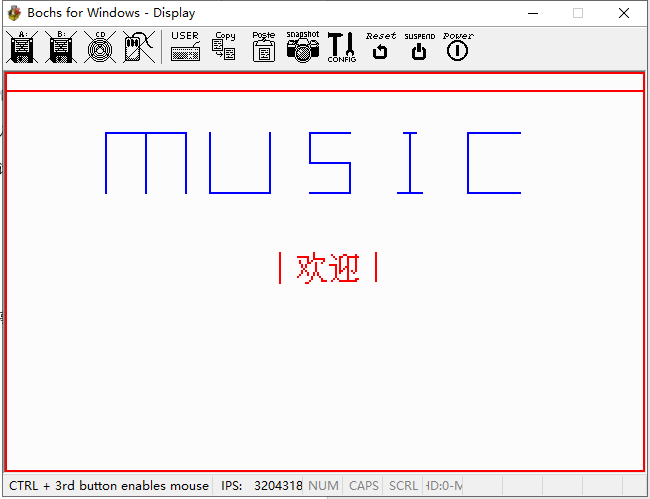
b：生日快乐

c: 小星星

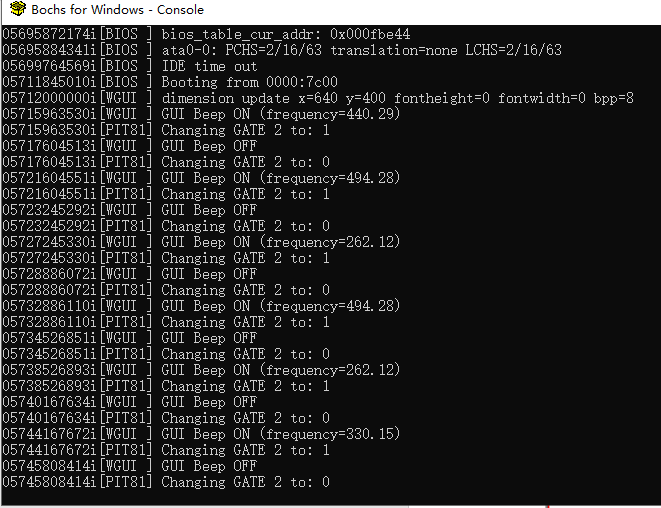
选择对应的歌曲，在播放歌曲时，闪动停止，播放完成后，闪动继续，此时方可选择下一首歌曲。

# 三、测试截图

## 3.1 欢迎界面

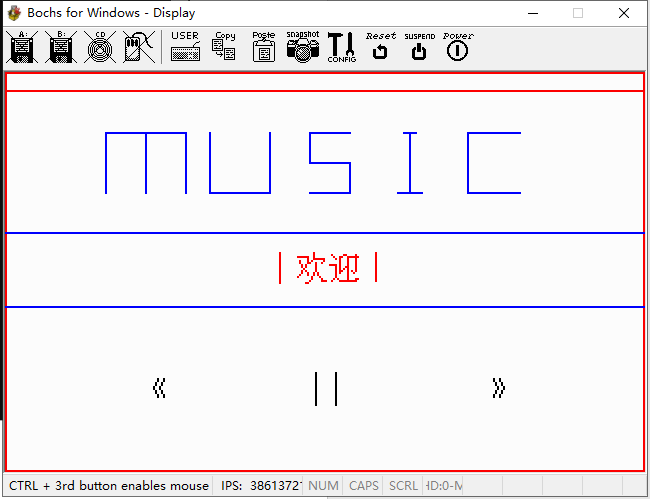


图表 1 欢迎界面



图表 2 欢迎界面音效

## 3.2 初始化完成



图表 3 初始化完成，双横线闪动（分别为赤橙黄绿青蓝紫）

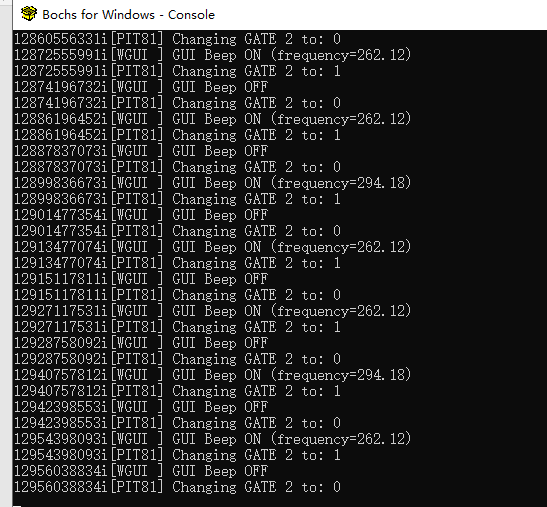


图表 4 初始化完成，双横线闪动（分别为赤橙黄绿青蓝紫）\_2

## 3.3 两只老虎

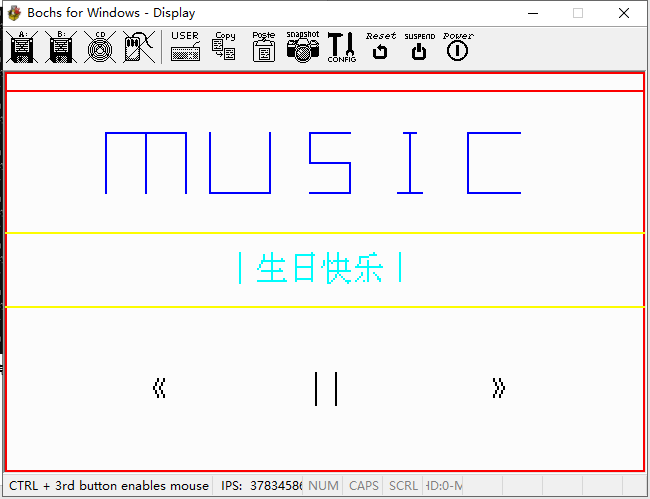


图表 5 两只老虎界面

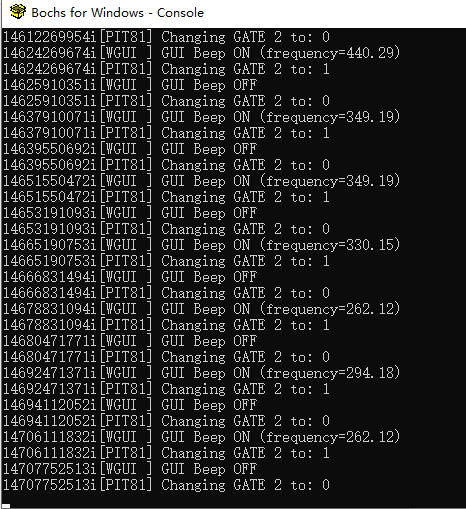


图表 6 两只老虎部分音频

## 3.4 生日快乐

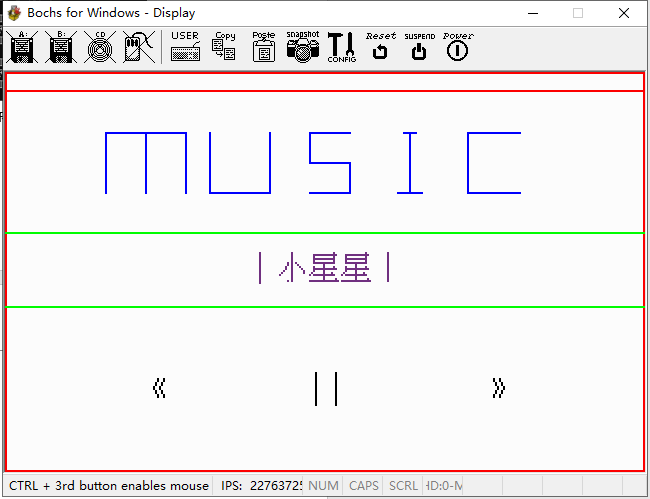


图表 7 生日快乐界面

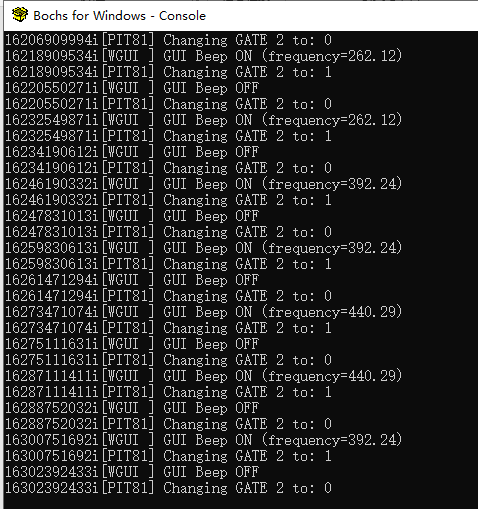


图表 8 生日快乐部分音频

## 3.5小星星



图表 9 小星星界面



图表 10 小星星部分音频

# 四、思路分析

本次课程设计，功能模块总共有如下大块：

## 4.1 丰富的颜色

本次设计了尽十种显示的颜色。来实现炫动的闪动效果。

## 4.2 载入欢迎界面

本次设计了类似于酷狗音乐的载入动画，酷狗音乐打开时会播放hello 酷狗，我们的播放器会播放 天空之城的第一句 播放完成后，才开始真正进入播放器。

## 4.3 基本框线函数封装：music+外框

本次的画线函数封装有如下几个要点：

首先是x，y，xy三个重要的点，x，y是用户传入的坐标，xy是通过count函数计算以后的点位，通过数据传回到xyplace中，保存点位。

这里count函数是通用的，只需要传入 x和y的坐标，就可以传回算好的点位。

这里画竖线和横线主要用在music上面，封装的画框的函数主要用在外框上面。

## 4.4 闪动框线+定时器

这里程序初始化好以后中间有两条横线是可以闪动的，闪动时可以选择歌曲，选择歌曲以后，闪动停止，尽情享受歌曲，歌曲播放完以后，可以继续选择歌曲。

这里的实现时通过定时器实现的，通过按键进入键盘中断时，对定时器中断替换新的中断（新的定时器中断time1为空，这样横线就不会闪动），在扬声器结束后，重新给到time0中断，实现重新闪动。

## 4.5 歌曲选择+键盘中断

通过键盘中断，判断输入的a，b，c三个字母，然后输出不同的歌曲。

## 4.6 播放歌曲+扬声器

这里按键以后，通过比较输入的键来进行跳转不同的播放函数，这里歌曲全部采用C大调的频率来播放，可能会有个别音准不准。

## 4.7 延时的巧妙运用

这里因为放歌和按键播放do re mi 不同，通过咨询老师，并且不断地调试，发现了延时的三比一的关系，这样可以极大清晰度的还原歌曲。具体的延时使用在代码里可以看到，写了很多延时的函数。

## 4.8 文字显示+py函数

这里的文本显示，涉及到文字的大小，采用16\*16的中文字库，由py生成，并且可以在py中调整位置，生成的x，y的坐标通过调用函数可以直接输出文字。

## 4.9 伪清屏技术

因为按键以后，屏幕要实时的显示当前正在播放的歌曲，清屏以后在重新绘制不仅浪费时间，还增加代码量，所以这里直接用白色的画笔将以前的字画成背景的白色，然后重新给歌曲的名字上色写出来，实现伪清屏。

# 五、代码截取片段分析

## 5.1 封装及计算点函数

*x1\_place* dw 0

*y1\_place* dw 0

*xy1\_place* dw 0;待写入的坐标

其中x,y是用户传入的坐标，xy1\_place是通过如下函数计算以后写回的点的真正位置：

*countxy:*

    push ax

    push bx

    mov ax,[*y\_place*]

    mov bx,320

    mul bx

    mov [*xy\_place*],ax

    mov bx,[*x\_place*]

    add [*xy\_place*],bx

    pop bx

    pop ax

    ret

## 5.2 写字的py函数以及调用

pointGroup.append(point\_x+238)  
pointGroup.append(point\_y+150)

字的位置的初始化。通过此可以输出16\*16的中文字的坐标，调整+号后面的位置即可调整字的位置。如下是调用写字的函数：

    push bx

    mov bx,0

    mov [*finpen*],bx

    pop bx

    mov si,*name0*

    push bx

    mov bx,[*name0l*]

    mov [*namelength*],bx

    pop bx

    call *outpoint*

需要有画笔颜色，字的数据段，以及字的点位长度，在上方代码均有体现。

这里有对字进行清除的效果就是直接把画笔颜色设置成背景颜色重画一遍就好，代码和上方一样，就不举出。

## 5.3 横线以及竖线及画框函数

;画横线==============================================

;bp 起始位置 di 终止位置 cl色号

*line:*

    push bp

    push di

    mov bp,[*xystart\_place*]

    mov di,[*xyend*]

    mov cl,[*finpen*]

*linepart:*

    mov byte [es:bp],cl

    add bp,1

    cmp bp,di

    jbe *linepart*

    pop di

    pop bp

ret

这里横线和竖线都已经封装好了，框就是对两个横线竖线进行封装就好，就不贴代码了，然后这里调用需要有画笔的颜色和传入对应的点：具体实现如下：

    mov bx,[*rdpen*+5]

    mov [*finpen*],bx

    mov bx,[*xy11\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy13\_place*]

    mov [*xyend*],bx

call *line*

## 5.4 延时处理：扬声器以及歌曲

*sleep\_1point0sec:*

    push cx

    push ax

    push dx

    mov  cx, 0FH ;延时高16位

    mov  dx, 4240H ;延时低16位 1s=1000 000us=这个

    mov  ah, 86H ;调用功能号，格式化驱动器定期中断

    int  15H

    pop dx

    pop ax

    pop cx

ret

这里用到了延时，打开扬声器和播放歌曲每一个音都要有延时，而且延时时长需要很好的控制，这里采用1：3的关系，就可以很好地播放歌曲啦！

播放的歌曲数据如下：需要有对应的C大调即可，然后通过运算播放！

*cisound* dw 262;1

    dw 294;2

    dw 330;3

    dw 349;4

    dw 392;5

    dw 440;6

    dw 494;7

    ;上面为C大调的do re me fa sol la xi

;c大调 1231 1231 345 345 565431 565431 121 121

*liangzhilaohu* dw 262,294,330,262,262,294,330,262

              dw 330,349,392,330,349,392

              dw 392,440,392,349,330,262,392,440,392,349,330,262

              dw 262,294,262,262,294,262

播放器的代码如下：

*outt:*;初始化扬声器

    mov [*pl\_1*],bx ;把do re。。。的频率先给

    mov al,*10110110B*

    out 43h,al

    push ax

    push dx

    push di

    mov dx,12h

    mov ax,32edh

    mov di,[*pl\_1*]

    div di

    mov [*pl\_2*],ax

    pop di

    pop dx

    pop ax

    mov ax, [*pl\_2*] ;计数初值=1.19 MHz, 600 Hz=1983

    out 42H, al ;送计数初值低位字节，8253的计数器2的地址42H

    mov al, ah

    out 42H, al ;送计数初值高位字节

    call *opens*

    call *sleep\_0point4sec*

    call *closes*

    ret

分为两部分，一个是计算部分，一个是播放部分：

*outliangzhilaohu:*

    push bx

    push di

    mov di,0

*pratoutliangzhilaohu:*

    mov bx,[*liangzhilaohu*+di]

    call *sleeploop3times*

    call *outt*

    add di,2

    cmp di,62

    jbe *pratoutliangzhilaohu*

    pop di

    pop bx

    ret

## 5.5 键盘中断

键盘中断用于处理播放不同歌曲，并且在播放的时候，对计时器中断要做停止操作，播放完以后重新打开计时器。如下是部分代码：

*int\_key:*

    mov dx,0x60

    in al,dx

    ;判断输入的是哪个按键，跳转

    ;开始判断下一个按键

    ;a

    mov word [ds:0x20],*int\_time1*

    mov word [ds:0x22],0

    cmp al,0x1e

    je *outa*

time函数在下一节介绍

## 5.6 计时器中断（闪动效果）

计时器一开始在初始化完成以后就开始闪动，闪动的代码如下:

*int\_time0:*

    push bx

    ;画横线

    inc bp

*xxx:*    cmp bp,7

    ja *decbp*

    mov bx,[*rdpen*+bp]

    mov [*finpen*],bx

     mov bx,[*xy7\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy8\_place*]

    mov [*xyend*],bx

    call *line*

     mov bx,[*xy9\_place*]

    mov [*xystart\_place*],bx

    mov bx,[*xy10\_place*]

    mov [*xyend*],bx

    call *line*

    pop bx

    mov al,0x20

    mov dx,0x20

    out dx,al   ;EOI=0,中断结束

    iret

*decbp:*

    mov bp,1

    jmp *xxx*

当键盘中断触发的时候，调用第二个空的计时器中断：

*int\_time1:*

    mov al,0x20

    mov dx,0x20

    out dx,al   ;EOI=0,中断结束

    iret

# 六、心得体会

## 6.1 设计中比较完善的

本次课程设计，用到的画框函数是以前的模板，所以在画框和数据段，我们小组用到的时间比较少。对于重复的算点的工作，就交给函数自己去算，我们输入x，y就好，但是虽然说着轻松，就比如music这几个字，直接用横竖画出来，就算是算x，y坐标，也花了我们两个不少的时间。

对于我们用到的颜色，我们因为要有闪动的变换，所以用了很多颜色，这个上课也有讲过，所以还是比较轻松的。

对于本次的设计，我们还专门设计了一个载入界面的音效，通过这个音效，我们可以更好地模拟酷狗音乐的使用，我们有一段天空之城的欢迎声音，代表我们的播放器在初始化了，这里还是用到了扬声器，并且适当的加速了，通过延时函数配合就好。

对于本次设计的计时器，我们调用了计时器，实现闪动的画面，感觉挺有节奏感的，很适合用来当音乐播放器使用。但是实现起来比较困难，因为如果在播放声音的时候闪动，那么系统就会崩溃，不知道什么原因，可能是计时器和扬声器冲突了。

然后这次的设计中，我们的显示文字的部分我觉得做的很不错，相比较之下的清屏重绘，我们采用的描成背景色来清屏，更加简单易懂，但是如果对于以后写的更加复杂的，以后还是使用清屏还一点。

## 6.2 还希望改进的

本次设计本来想还有实现暂停键的，但是发现，在扬声器运行的时候，按键需要按好多次才会有反应，但是有反应却不会响应，如果直接停止键盘中断，然后再停止扬声器，也是不行，bochs会报错，所以这里没有做暂停，也是尝试了很久很久，都没有成功实现暂停，可能是美中不足的一点把。

想要实时的显示歌词，但是因为字库的点太多了，会导致代码量非常庞大，就没有显示歌词了，所以直接就放弃了歌词，如果有更好的显示文字的方法，那么以后可以加上。

最后，想要在放歌的时候闪动，但是因为计时器中断会和扬声器冲突，导致系统崩溃，所以还是放弃了，只有在放歌的时候替换空的计时器中断，才可以正常运行。

总之，在很多地方，我们都还是想在重新改进的，但是尝试了很多次以后，发现微机这门课学问还是很深，我们还得在努力才可以，最后感谢老师的耐心教导！