**西安电子科技大学**

微机原理课程设计  **课程实验报告**

**实验名称**  键盘电子乐器演奏程序设计

计算机科学与技术 学院 1603011 班

成 绩

姓名 张瑞 学号 16030110032

同作者 祁青雅 沈童 李睿玉

实验日期 2019 年 4 月 9-14 日

实验地点E-Ⅱ-311实验批次第六批

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）  五、实验结果分析及回答问题（或测试环境及测试结果） |

**键盘电子乐器演奏程序设计**

1. **课程设计目的**

1. 了解利用 8254 定时器产生不同音符的原理及方法。

2. 加深对计算机键盘扫描程序和执行过程的理解。

3. 初步掌握键盘电子乐器演奏程序的设计方法。

1. **课程设计的内容**

设计一个可以通过微机键盘演奏不同音乐的控制系统，系统通过按下微机键盘模仿电子琴键驱动实验箱上的喇叭发生，实现演奏音乐的功能。

1. **系统功能与设计要求**
2. **基本功能要求**

能够演奏 C 调包含高、中、低音的不同节拍的乐曲，音调与频率的对应关系如下：

中 音：1-523Hz, 2-575Hz, 3-659Hz, 4-698Hz, 5-784Hz, 6-880Hz, 7-988Hz

低 8 度音：基本音频率/2 例如低音1为 523/2=261.5Hz

高 8 度音：基本音频率\*2，例如高音1为 523\*2=1046Hz

微机键盘与电子琴键盘的对应关系为：高音依次为 Q,W,E,R,T,Y,U，中音依次为A,S,D,F,G,H,J，低音依次为 Z,X,C,V,B,N,M。

1. **发挥部分**
2. 增加自动演奏乐曲功能。当按下 P 键后，自动播放一首乐曲，曲长大于 20 秒。
3. 增加琴键图形显示功能。屏幕上显示 21 个琴键（高中低音），当按下某键后屏幕上的对应琴键有所反映。
4. 增加一个音调 D，当从 C 调改为 D 调后，演奏音按照 D 调(比 C 高一个调)发音。

**四、设计思路**

一首乐曲由若干音符组成，一个音符对应一个频率，将与一个频率对应的计数初值写入计数器就可以产生相应的频率，计算公式如下：

计数初值 = CLK0(输入频率)/OUT0（输出频率）

当音符对应的频率确定后，还需要知道每个音符演奏的时间。我们知道，音符的演奏时间是由节拍控制的，分为一拍，半拍，1/4 拍，1/8……，如果在一首乐曲中，音符演奏的最短时间是 1/8 拍，我们就将 1/8 拍作为一个最短时间单位 1，那么 1/4 拍单位时间就是 2，1/2 拍就是 4，一拍就是 8，假定一拍的时间为 1s,那么 1/2,/1/4,1/8 拍的时间就是0.5s,0.25s,0.125s。

音符的演奏时间 = 单位时间 × N ( N 为调式参数，影响音乐的节奏)

编程首先应该对计数器初始化，然后扫描键盘，根据扫描结果选择对应的频率，输出到实验箱的音频电路输入端口，并调用延时子程序控制节拍。

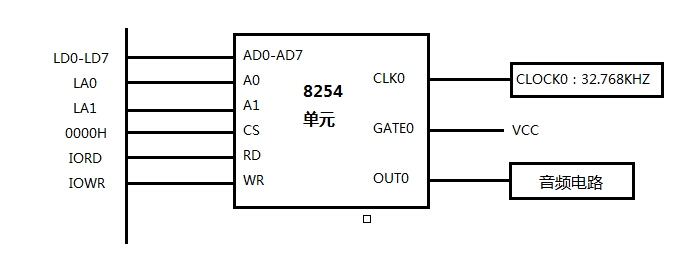
**五、操作步骤**

**1．硬件设计方案及硬件连线图**

* 1. **设计方案**

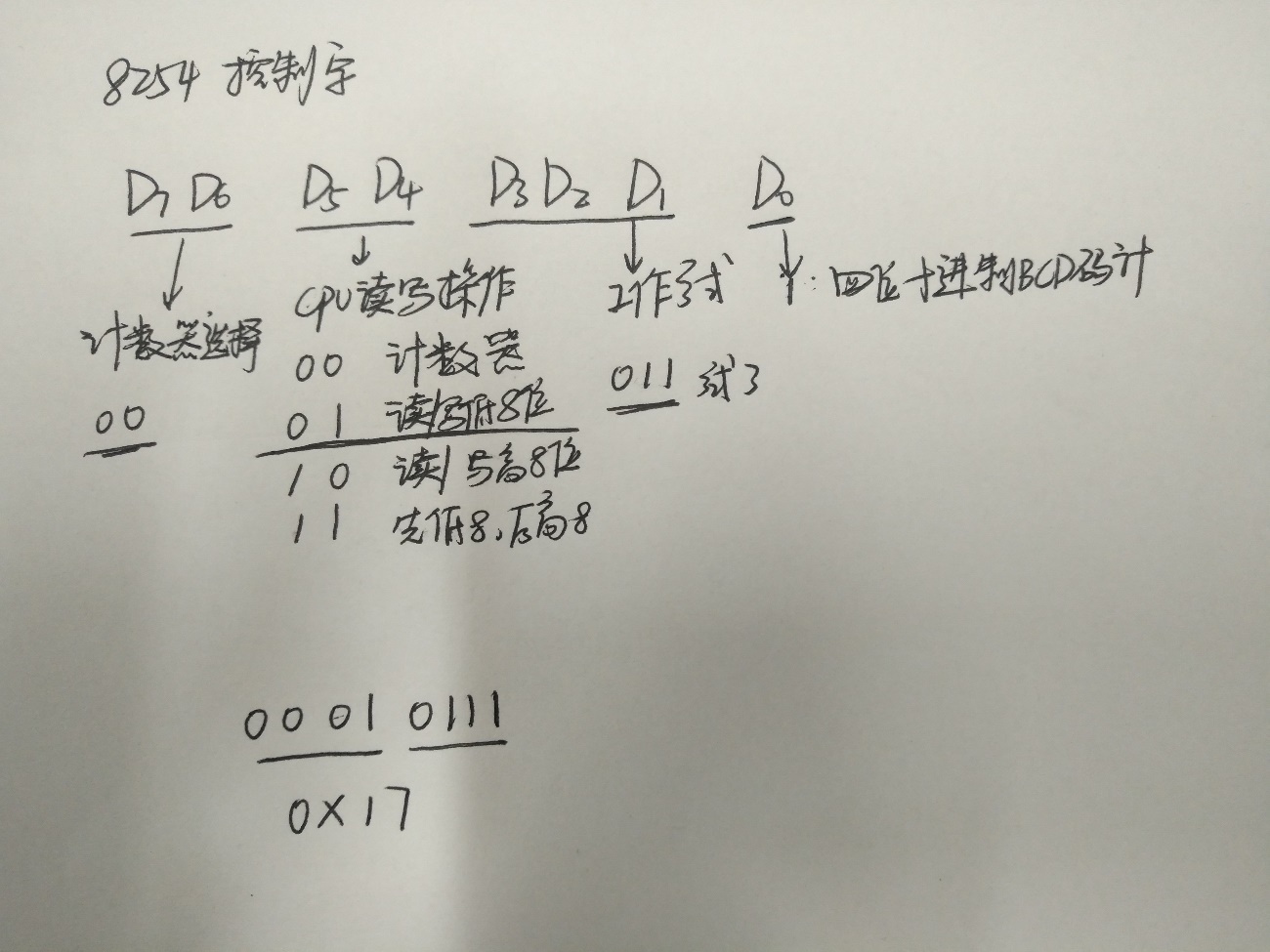
在本次实验中我们为了控制音调的高低，使用的是8254定时器，使用方式三，通过改变分频值来控制。分频值的计算是用初始的时钟频率32.768KHZ来除以想得到的频率，然后舍入成整数来作为分频值输入8254。  
 具体的连线情况大致如上图，从总线上的LD0-LD7接8254的D0-D7，LA0和LA1接8254的A0和A1，IOWR和IORD接8254的WR和RD，地址0000H接8254的CS。除此之外，8254的GATE0接VCC，CLK0接初始的时钟频率32.768KHZ，OUT0接音频电路即可。

* 1. **连线图**

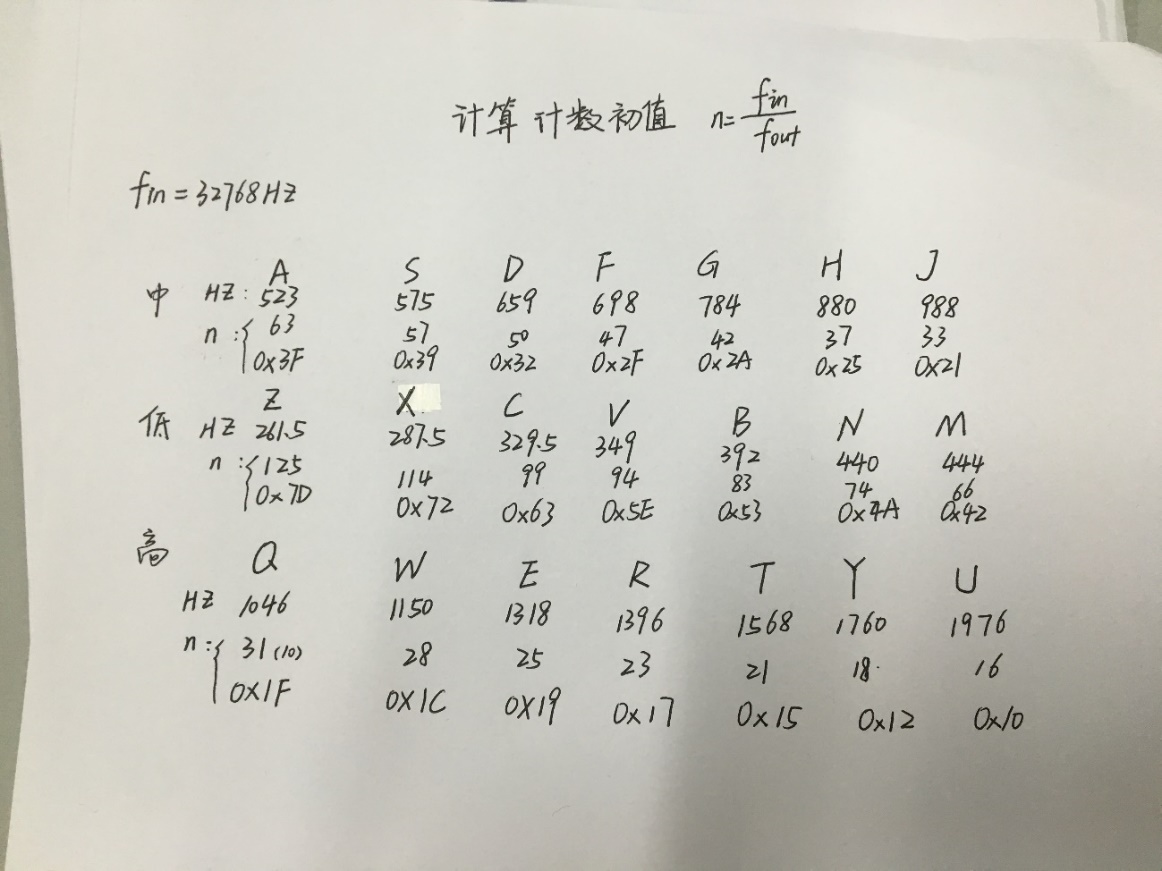


* 1. **8254控制字的计算**

我们选择使用计数器0的工作方式3进行分频，计数初值只需8位，因此读/写低八位，计数初值的形式是选择十进制BCD码计数。

****

* 1. **计数初值的计算**

****

**2.软件设计方案及流程图**

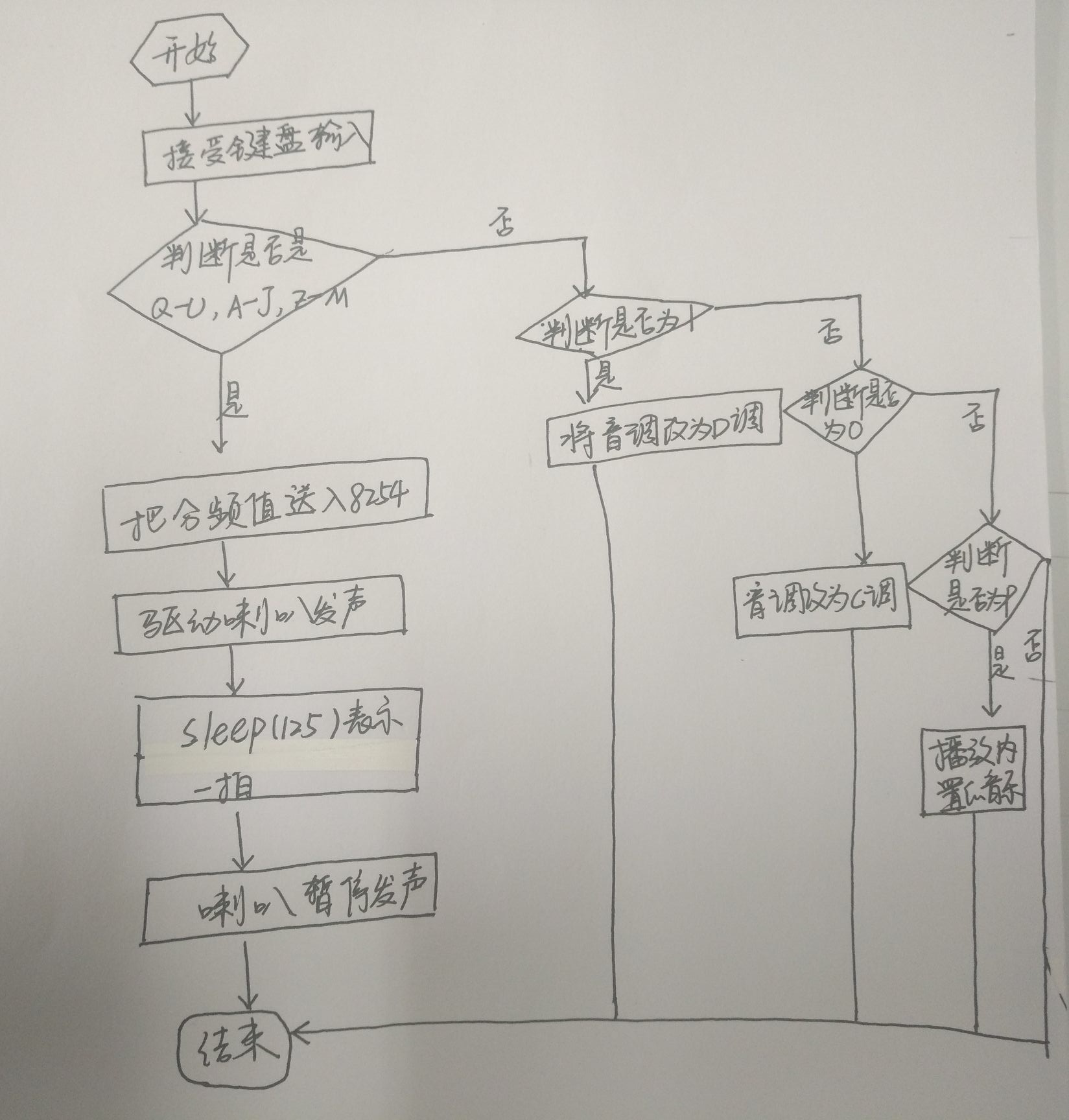
**a) 设计方案**

对于软件开发环境，我们采用的 PC机中的 VC++ 6.0集成环境，新建了一个 MFC的工程，加入硬件总线驱动相关的参数（，然后编写自己的程序逻辑。

**b) 设计思想：**

采用先进的面向对象思想，将界面中的每一个琴键按钮当作一个对象，用一个类表示，当触发该按钮时，就相当于触发该事件，将消息传入该类，执行该段程序。

**c) 软件流程图:**该系统软件流程图如下图所示：



* 1. **具体实现思路：**

**自动播放音乐的实现:**

对于一首音乐：我们将其分为了三个部分：

Music[]:音符，用1234567代替do,re,mi,fa,suo,la,xi;

Musicc[]:各音符之间的延时，可以从乐谱里面看到；

Musiccc[]:各音符的调，分别用2，1，0高中低音调；

分别依据乐谱谱写我们所需要的数组，然后通过循环遍历来调用函数来给硬件传递分频的计数值，最终能播放整首音乐。

**高低音的实现：**

初始化时，将音调全部置为中调，即为1，对于高音，只需要将musiccc[i]++即可，同理低音就musiccc[i]--;

**每个音符的初始化及键盘键入设置：**

**音符的频率是我们算出来的，然后在实际调试过程中，我们稍微改变了一点，**使我们的电子琴更接近原生的音乐，例如高音的do，用键盘上的A键触发，十六进制为0x41,do对应的频率为0x003c，使用这样一句进行初始化：

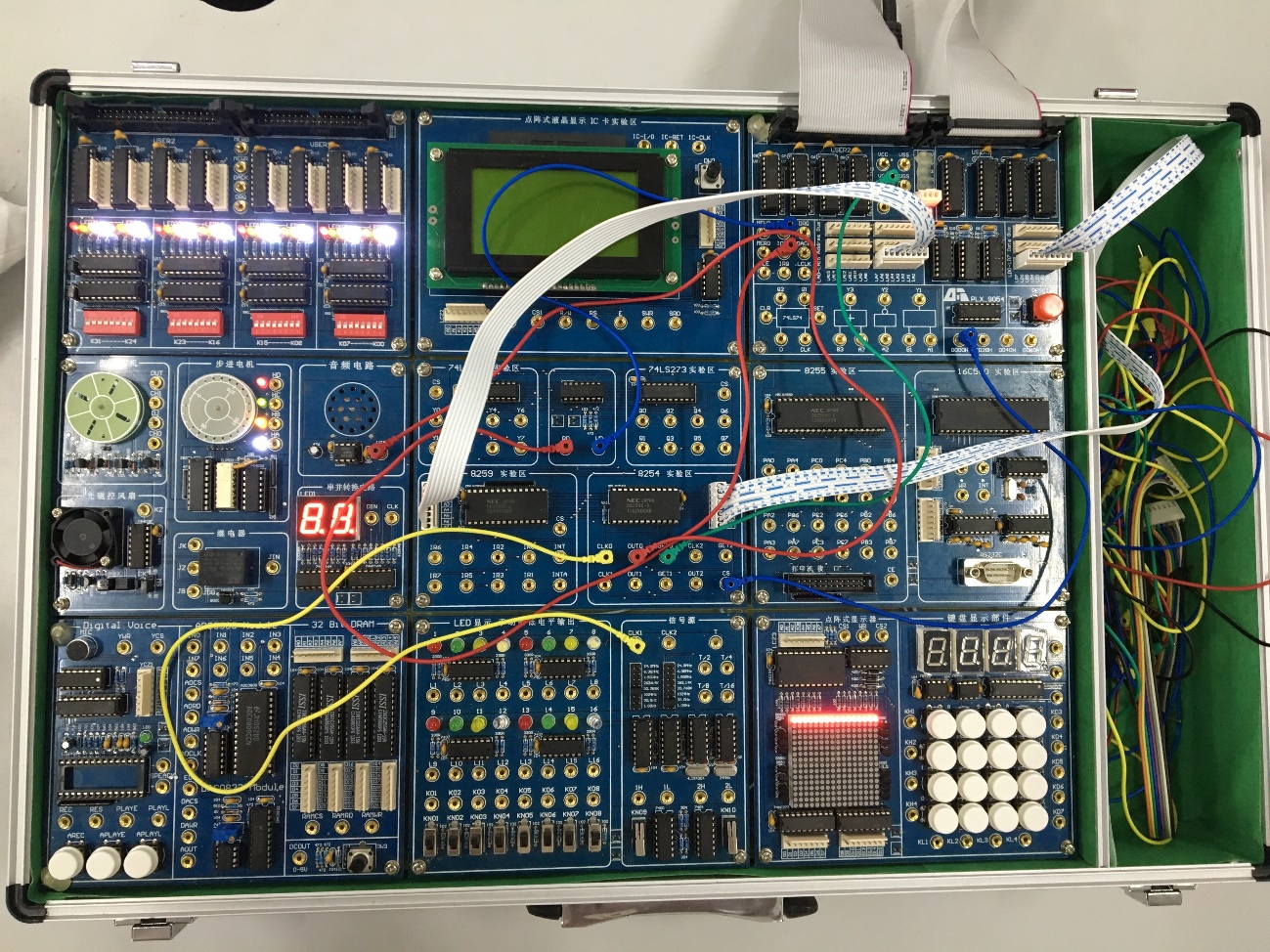
IOBbWrite(0x0000,0x0056);

键盘键入使就触发OnButton（）函数，相当于用鼠标控制软件一样。

将音乐的节奏和音调存储在数组里面，遍历数组，检测到对应字符调用相应的函数控制扬声器发出频率（音调）不同的声音。

**七、完成内容及实验结果**

将实验板上的电路连接好，如下图：



**本系统实际完成的功能：**

**1.乐器演奏功能：**

能够演奏 C 调包含高、中、低音的不同节拍的乐曲，

其中微机键盘与电子琴键盘的对应关系为：

高音依次为Q,W,E,R,T,Y,U，

中音依次为A,S,D,F,G,H,J，

低音依次为 Z,X,C,V,B,N,M。

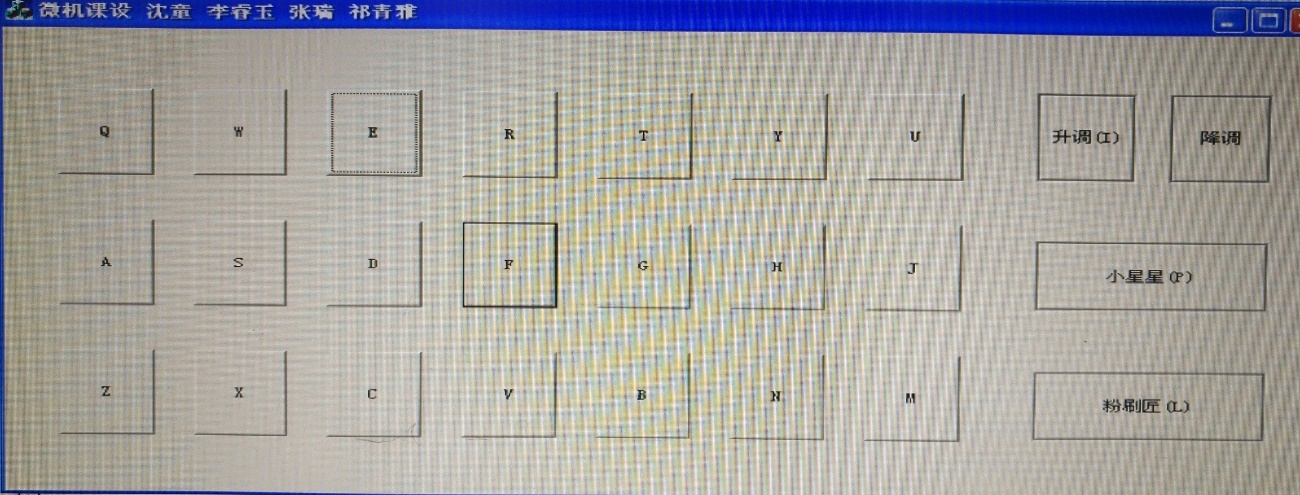
**2.自动演奏乐曲功能:**

运行程序，按下P键或L键后自动播放音乐，曲长大于20秒。

**3.琴键图形显示功能：**

运行程序，屏幕上显示虚拟琴键，有21个，包括高中低音，当按下某键后，屏幕上的对应琴键有所反映。

编译运行程序。键盘控制程序运行，界面会根据键盘按键做出相应起落，喇叭发出相应音调，当按下P键或图形化界面上的“小星星”时，或按下L键或“粉刷匠”自动播放相应的乐曲，界面如下：



**4.升调降调功能：**

点击图形化界面上面的“升调”将会按照C调演奏，点击图形化界面上面“降调”将会按照D调演奏。

**八、个人心得**

本次微机原理与系统设计课程设计中，我们小组合作完成了键盘电子乐器演奏程序设计。其中我负责软件部分，通过与前面硬件之间算好的数据对接，实现软件界面的设计及几个功能的设计。

通过微机实验，让我对微机硬件等有了进一步的了解，同时，作为主编程的人员， 我也对MFC编程有了更熟悉的掌握。在这次实验中，我觉得团队非常重要，这大概是为什么大三许多课程设计都是队伍的原因。通过组队，可以提升我们的合作能力及交流能力，在西电的工程设计实验中，大一大二独当一面，大三大四精诚合作，对我们学术可谓质的提升。我们队先对项目整体进行了解，使用8254的方式3来实现分频，通过电脑程序写入计数值，然后再加上几个附加功能，最终实现这些项目。

在本次实验过程中我们比较顺利的实现了实验目标，但也遇到许多细微的问题，让我印象深刻的就是在实验板上调试设计好的音调时音调不准确，之后我们组员几个通过微调改变计数值，使得最后的音乐更加真实。在升降调中，采用另外一个数组实现音调的升降。通过与其他同学的交流以及查阅相关资料，最终做成了此次实验。

总而言之，通过此次工程设计我将微机原理课的理论真正进行了实践，同时还提高了团队协作能力，编程方面也得到了进一步的提升，感谢学校给我们提供这样的平台来提升我们的能力！

**九、附录：**

**课程设计实验环境：**

1. 硬件配置：

微机（Pentium 4） 一台

微机接口技术实验箱 一个

ISA–PCI 转接卡 一块

连接电缆 一条

微机接口技术实验讲义 一本

连接导线 若干条

2. 软件环境： Windows XP/2000/Win 7 平台 Visual C++ 6.0 编译器

**程序代码（主要修改的代码）：**

// xxpDlg.cpp : implementation file

//

#include <string.h>

#include <windows.h>

#include "stdafx.h"

#include "xxp.h"

#include "xxpDlg.h"

#include "QTH.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog

{

public:

CAboutDlg();

// Dialog Data

//{{AFX\_DATA(CAboutDlg)

enum { IDD = IDD\_ABOUTBOX };

//}}AFX\_DATA

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CAboutDlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

// Implementation

protected:

//{{AFX\_MSG(CAboutDlg)

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

};

/\*\*

\*\*\*这些部分为MFC自动生成的部分\*\*\*

\*\*\*/

BEGIN\_MESSAGE\_MAP(CXxpDlg, CDialog)

//{{AFX\_MSG\_MAP(CXxpDlg)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTONQ, OnButtonq)

ON\_BN\_CLICKED(IDC\_BUTTONW, OnButtonw)

ON\_BN\_CLICKED(IDC\_BUTTONE, OnButtone)

ON\_BN\_CLICKED(IDC\_BUTTONR, OnButtonr)

ON\_BN\_CLICKED(IDC\_BUTTONT, OnButtont)

ON\_BN\_CLICKED(IDC\_BUTTONY, OnButtony)

ON\_BN\_CLICKED(IDC\_BUTTONU, OnButtonu)

ON\_BN\_CLICKED(IDC\_BUTTONA, OnButtona)

ON\_BN\_CLICKED(IDC\_BUTTONS, OnButtons)

ON\_BN\_CLICKED(IDC\_BUTTOND, OnButtond)

ON\_BN\_CLICKED(IDC\_BUTTONF, OnButtonf)

ON\_BN\_CLICKED(IDC\_BUTTONG, OnButtong)

ON\_BN\_CLICKED(IDC\_BUTTONH, OnButtonh)

ON\_BN\_CLICKED(IDC\_BUTTONJ, OnButtonj)

ON\_BN\_CLICKED(IDC\_BUTTONZ, OnButtonz)

ON\_BN\_CLICKED(IDC\_BUTTONX, OnButtonx)

ON\_BN\_CLICKED(IDC\_BUTTONC, OnButtonc)

ON\_BN\_CLICKED(IDC\_BUTTONV, OnButtonv)

ON\_BN\_CLICKED(IDC\_BUTTONB, OnButtonb)

ON\_BN\_CLICKED(IDC\_BUTTONN, OnButtonn)

ON\_BN\_CLICKED(IDC\_BUTTONM, OnButtonm)

ON\_BN\_CLICKED(IDC\_BUTTONP, OnButtonp)

ON\_BN\_CLICKED(IDC\_BUTTON1, OnButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, OnButton2)

ON\_BN\_CLICKED(IDC\_BUTTON3, OnButton3)

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CXxpDlg message handlers

BOOL CXxpDlg::OnInitDialog()

{

//初始化

CDialog::OnInitDialog();

Init9054();

IO8bWrite(0x0003,0x0017);

time=1;

for(int i=0;i<42;i++)

musiccc[i]=1;

// int musiccc[]={1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,0,1,1,0,1,1,1,1};

// Add "About..." menu item to system menu.

// IDM\_ABOUTBOX must be in the system command range.

ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);

ASSERT(IDM\_ABOUTBOX < 0xF000);

CMenu\* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != NULL)

{

CString strAboutMenu;

strAboutMenu.LoadString(IDS\_ABOUTBOX);

if (!strAboutMenu.IsEmpty())

{

pSysMenu->AppendMenu(MF\_SEPARATOR);

pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);

}

}

// Set the icon for this dialog. The framework does this automatically

// when the application's main window is not a dialog

SetIcon(m\_hIcon, TRUE); // Set big icon

SetIcon(m\_hIcon, FALSE); // Set small icon

// TODO: Add extra initialization here

return TRUE; // return TRUE unless you set the focus to a control

}

void CXxpDlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == IDM\_ABOUTBOX)

{

CAboutDlg dlgAbout;

dlgAbout.DoModal();

}

else

{

CDialog::OnSysCommand(nID, lParam);

}

}

// If you add a minimize button to your dialog, you will need the code below

// to draw the icon. For MFC applications using the document/view model,

// this is automatically done for you by the framework.

void CXxpDlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // device context for painting

SendMessage(WM\_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

// Center icon in client rectangle

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// Draw the icon

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

//CDialog::OnPaint();

CRect rect;

CPaintDC dc(this);

GetClientRect(rect);

dc.FillPath();

CDialog::OnPaint();

}

}

// The system calls this to obtain the cursor to display while the user drags

// the minimized window.

HCURSOR CXxpDlg::OnQueryDragIcon()

{

return (HCURSOR) m\_hIcon;

}

BOOL CXxpDlg::PreTranslateMessage(MSG\* pMsg)

{

// TODO: 在此添加专用代码和/或调用基类

//判断是否为键盘消息

if(pMsg->message == WM\_KEYUP){

IO8bWrite(0x0000,0x0000);

return true;

}

/\*

if(pMsg->message == WM\_KEYDOWN){

return true;

}\*/

if (WM\_KEYFIRST <= pMsg->message && pMsg->message <= WM\_KEYLAST)

{

if(pMsg->wParam==0x41)

{

(CButton \*)GetDlgItem(IDC\_BUTTONA)->SetFocus();

// MessageBox(NULL,\_T("a"),MB\_OK);

IO8bWrite(0x0000,0x003C);

//OnButtona();

return TRUE;

}

/\*\*\*

\*\*\*这些部分为相似的部分，绑定键盘与操作\*\*\*

\*\*\*/

}

return CDialog::PreTranslateMessage(pMsg);

}

void CXxpDlg::OnButtonq()

{

// TODO: Add your control notification handler code here

(CButton \*)GetDlgItem(IDC\_BUTTONQ)->SetFocus();

IO8bWrite(0x0000,0x001B);

Delay(time);

IO8bWrite(0x0000,0x0000);

Delay(1);

}

/\*\*\*

\*\*\*这些部分为相似的部分，按键的功能\*\*\*

\*\*\*/

void CXxpDlg::OnButtonm()

{

// TODO: Add your control notification handler code here

(CButton \*)GetDlgItem(IDC\_BUTTONM)->SetFocus();

IO8bWrite(0x0000,0x0045);

Delay(time);

IO8bWrite(0x0000,0x0000);

Delay(1);

}

void CXxpDlg::Delay(int n)

{

Sleep(200\*n);

}

void CXxpDlg::OnButtonp()

{

// TODO: Add your control notification handler code here

(CButton \*)GetDlgItem(IDC\_BUTTONP)->SetFocus();

int music[]={1,1,5,5,6,6,5,4,4,3,3,2,2,1,5,5,4,4,3,3,2,5,5,4,4,3,3,2,1,1,5,5,6,6,5,4,4,3,3,2,2,1};

int musicc[]={1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,1,2};

int len= 42;

for(int i=0;i<len;i++)

{

if(musiccc[i]==2)

{

time=musicc[i];

if(music[i]==1) OnButtonq();

if(music[i]==2) OnButtonw();

if(music[i]==3) OnButtone();

if(music[i]==4) OnButtonr();

if(music[i]==5) OnButtont();

if(music[i]==6) OnButtony();

if(music[i]==7) OnButtonu();

}

if(musiccc[i]==1)

{

time=musicc[i];

if(music[i]==1) OnButtona();

if(music[i]==2) OnButtons();

if(music[i]==3) OnButtond();

if(music[i]==4) OnButtonf();

if(music[i]==5) OnButtong();

if(music[i]==6) OnButtonh();

if(music[i]==7) OnButtonj();

}

if(musiccc[i]==0)

{

time=musicc[i];

if(music[i]==1) OnButtonz();

if(music[i]==2) OnButtonx();

if(music[i]==3) OnButtonc();

if(music[i]==4) OnButtonv();

if(music[i]==5) OnButtonb();

if(music[i]==6) OnButtonn();

if(music[i]==7) OnButtonm();

}

}

}

void CXxpDlg::OnButton1()

{

// TODO: Add your control notification handler code here

for(int i=0;i<42;i++)

musiccc[i]++;

}

void CXxpDlg::OnButton2()

{

// TODO: Add your control notification handler code here

for(int i=0;i<42;i++)

musiccc[i]--;

}

void CXxpDlg::OnButton3()

{

(CButton \*)GetDlgItem(IDC\_BUTTONP)->SetFocus();

int len= 42;

int music[]={5,3,5,3,5,3,1,2,4,3,2,5,5,3,5,3,5,3,1,2,4,3,2,1,2,2,4,4,3,1,5,2,4,3,2,5,5,3,5,3,5,3,1,2,4,3,2,1};

int musicc[]={2,2,2,2,2,2,4,2,2,2,2,8,2,2,2,2,2,2,4,2,2,2,2,8,2,2,2,2,2,2,4,2,2,2,2,8,2,2,2,2,2,2,4,2,2,2,2,8};

int musiccc[]={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

for(int i=0;i<len;i++)

{

if(musiccc[i]==2)

{

time=musicc[i];

if(music[i]==1) OnButtonq();

if(music[i]==2) OnButtonw();

if(music[i]==3) OnButtone();

if(music[i]==4) OnButtonr();

if(music[i]==5) OnButtont();

if(music[i]==6) OnButtony();

if(music[i]==7) OnButtonu();

}

if(musiccc[i]==1)

{

time=musicc[i];

if(music[i]==1) OnButtona();

if(music[i]==2) OnButtons();

if(music[i]==3) OnButtond();

if(music[i]==4) OnButtonf();

if(music[i]==5) OnButtong();

if(music[i]==6) OnButtonh();

if(music[i]==7) OnButtonj();

}

if(musiccc[i]==0)

{

time=musicc[i];

if(music[i]==1) OnButtonz();

if(music[i]==2) OnButtonx();

if(music[i]==3) OnButtonc();

if(music[i]==4) OnButtonv();

if(music[i]==5) OnButtonb();

if(music[i]==6) OnButtonn();

if(music[i]==7) OnButtonm();

}

}

}