版权声明 本文内的全部内容均由本人编写，可以用来参考学习，但不可用来进行商业活动。一切代码解释权均归作者扎楠所有。所有题目以实验指导书为准。 之所以制作这个word只是为了给将来和即将面对席大林老师以及他的课程的同学一些程序上的参考，避免走弯路，花更多的时间在缤纷多彩的大学生活中，不要纠结于这些琐事。

双音多频(DTMF)信号的检测

姓名：

学号：

指导教师：

完成时间：

**一、实验目的**

1. 理解DTMF信号的产生原理及其检测方法
2. 提高分析和解决问题的能力
3. 提高数字信号处理的实际能力

**二、实验内容**

1. 设置参数，读入电话号码
2. 根据输入的电话号码产生包含两个频率分量的205点时域离散DTMF信号，
3. 对时域离散DTMF信号利用Goertzel算法(参见附录)进行频率检测，画出幅度谱
4. 根据幅度谱的两个峰值，分别查找并确定所输入的电话号码

**三、各部分工作原理**

**1.双音多频信号的工作原理及产生**

现在我们所用的电话中，每一个数字按键(包括#和\*键) 都是由两种频率的单音信号组成的，这两种单音信号被分为高频带和低频带。低频带有四个频率，即697Hz，770Hz，852Hz 和941Hz；高频带也有四个频率，即1209Hz，1336Hz，1447Hz 和1633Hz。每一位号码均由一个低频带频率和一个高频带频率叠加形成。图1所示是国际标准(ITU84)认可的数字和符号键的频率分配情况。最右一排目前尚未应用于开发。

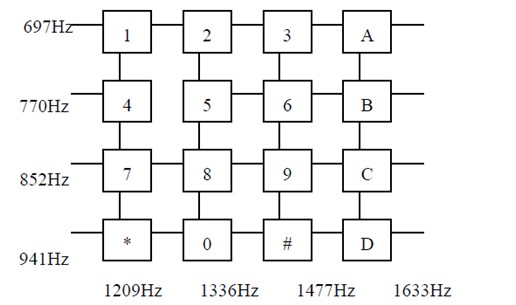


图1 按键电话频率配置

每个按键对应的DTMF信号为，其中f1为低频单音，f2为高频单音。电话中的双音多频信号有两个作用：用拨号信号去控制交换机接通被叫的用户电话机；控制电话机的各种动作，如播放留言、语音信箱等。

用8kHz对DTMF信号进行采样，采样后得到时域离散信号为：



由于采样率是8kHz，因此要求每125ms输出一个样本，得到的序列再送到D/A变换器，它的输出经过平滑滤波便是连续时间的DTMF信号。DTMF信号通过电话线路再送到交换机。

**2.双音多频信号检测原理及方法**

在接收端，要对收到的双音多频信号进行检测，即检测两个正弦波的频率，以判断其对应的十进制数字或符号。可以利用DFT或者FFT对双音多频信号进行频谱分析，由信号的幅度谱大致判断出信号的两个频率，最后通过查表确定出对应的数字或符号。FFT是DFT的快速算法，但当计算的频率点数远小于DFT的变换区间长度时，用FFT快速算法的效果不明显，而且还要占用很多内存，因此不如直接用DFT合适。为了提高系统的检测速度并降低成本，已开发出一种特殊的DFT算法，称为戈泽尔算法(Goertzel)。

置于参数的选择，由于要检测信号的频率范围为697~1633Hz，但考虑到存在语言干扰，除了检测这8个频率外，还要检测他们的二次倍频的幅度大小。波形正常且干扰小的正弦波的二次倍频是很小的。如果发现二次谐波很大，则认为不是DTMF信号。这样频谱分析的频率范围为697~3266Hz(倍频,1633x2)。按照采样定理，信号的最高频率不能超过折叠频率，即0.5Fs≥3266Hz，由此要求最小采样频率为6.53kHz。这里已经规定Fs=8kHz，因此一定满足对频谱分析范围的要求。按照Tpmin=13.7ms，Fs=8kHz，算出对信号最少的采样点数为。为了使结果更加准确，我们将N设为512.

DFT的频率采样点频率为，相应地在模拟域的采样点频率为，希望选择一个合适的N，用该式算出的能接近要检测的频率，或者用8个频率中的任一个频率代人中时，得到的k值最接近整数值。这样根据最大幅度检测的频率虽然有误差，但由此可以正确判断DTMF信号所表示的数值。

**3. 戈泽尔(Goertzel) 算法原理**

Goertzel算法利用相位因子{}的周期性，并使我们能将DFT运算表示为线性滤波运算，由于，我们可以利用该因子去乘DFT，则有



现在能够用Goertzel算法实现DTMF解码器了，由于有8种可能的音频要检测，所以需要8个(7)式所给出的滤波器，每个滤波器调谐到这8个频率之一上。在DTMF解码器中，不需要计算复数值X(k)，只要求出幅度值，或幅度平方值就可以了。所以，计算涉及分子项（滤波器计算的前向部分）的DFT值的最后一步可以简化，我们有



于是，DTMF解码器中就完全避免了复数运算。

**四、总系统图**



**五、程序主体部分**

本程序的主体部分是戈泽尔算法的实现和图像的生成，这是前半个学期席大林老师信号工程处理课的核心部分与毋立芳老师数字图像处理的精华部分的整合。

首先根据DFT的变换公式：

经欧拉公式变换，得 

就有实部和虚部分别为：

Re[k]=X[n]\*cos();Im[k]=X[n]\*cos();

求实部和虚部的平方和，再开根号就得到幅度最大值。

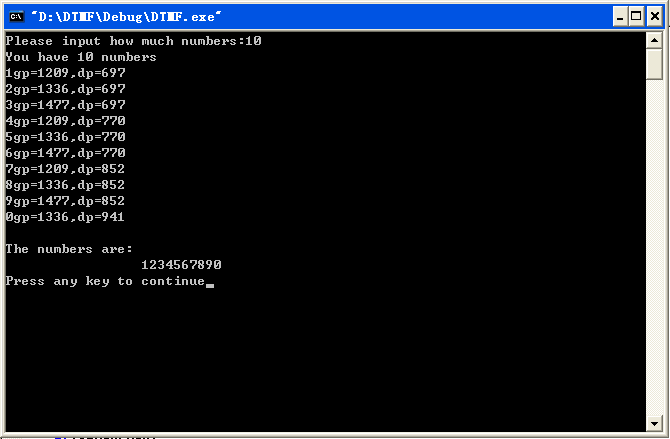
画数字的频谱图时，具体方法如下：将横轴上256个像素点作为0~255个频率，求一个频率值时候，利用该频率对应的点数，求出归一化后的数，从横轴向上画相应个点数，循环下去，将所有256个频率都画好，就得到图。

**六、程序框图**



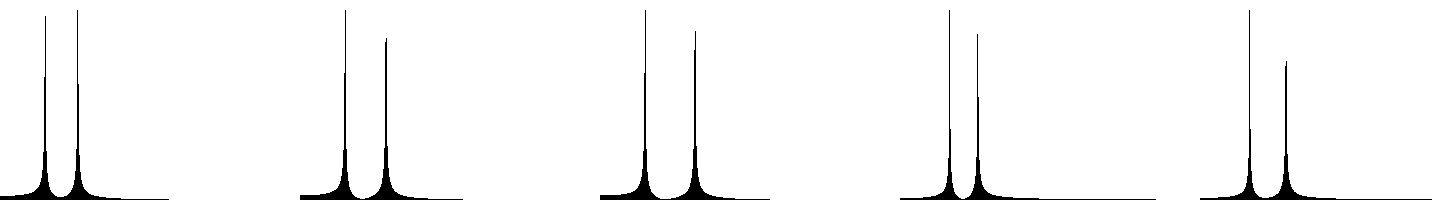
**七、程序结果显示**

试以10作为输入的数字个数，分别输入1、2……9、0十个数，在屏幕上显示了输入数字的高频与低频，以及输出的结果。

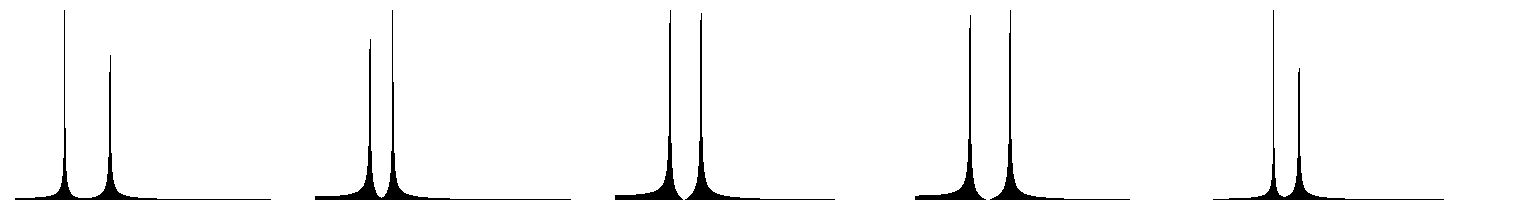


在程序运行结束之后，在d:\\DTMF\\pp.bmp上画出频谱图，在此文件夹中生成pp.bmp覆盖原图像，并把变换后的幅值存在output.txt中。

得到上述10个数的频谱图：由于空间有限，截两个图。



1 2 3 4 5



6 7 8 9 0

也可以利用Matlab对output文件进行绘图：

其主体程序为：

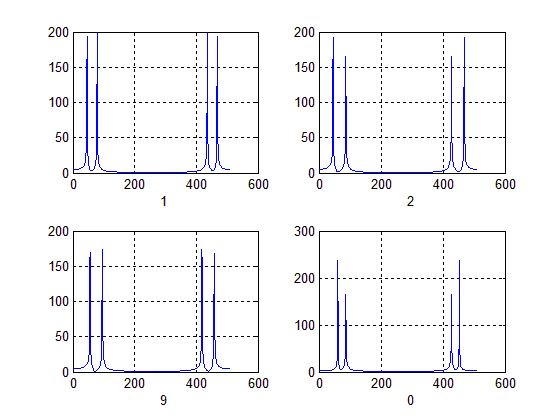
subplot(2,2,1); //在2x2的空间上第一个位置画output内容的图

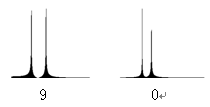
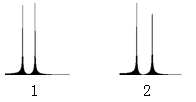
fid=fopen('d:\\DTMF\\output1.txt','r');

myData=fread(fid,512,'double');

plot(myData);grid;xlabel('1');

得到1、2、9、0的图像：





和原图比较可以知道，通过程序绘制的图像正确。

**八、参考文献**

1.谭浩强 《C程序设计》 清华大学出版社2005

2.胡广书 《数字信号处理导论》 清华大学出版社2005

3.程佩青 《数字信号处理》 清华大学出版社 2012

4.唐向宏等 《MATLAB及在电子信息类课程中的应用》 电子工业出版社 2009

5.鲁沐浴 《C语言最新编程技巧200例》 电子工业出版社 1997

6.刘维 《精通matlab与C\C++混合程序设计》 北航出版社 2012

7.冈萨雷斯 《数字图像处理》 国外教材

**附录、程序源代码**

#include "stdio.h"

#include "math.h"

#include "conio.h"

#include "string.h"

#include "windows.h"

#define PI 3.1415926

#define N 512 //采样点数

//将0--9，\*#的高频和低频分量存成数组，以便画图和显示时调用

int gaopin[12]={1336,1209,1336,1477,1209,1336,1477,1209,1336,1477,1209,1477};

int dipin[12]={941,697,697,697,770,770,770,852,852,852,941,941};

int Q=0; //循环因子，目的是在一张图上画多个频谱

int Qm=0; //输入多少个数就画几个频谱

double input[N];

char jianru(void); //键入键值

char Goertzel(void); //利用戈泽尔算法确定输入数字，并画频谱

void DTMF(char number); //构造DTMF信号

int main()

{char jian=0;

int i;

int en=0;

int time=0; //输入号码位数

int number=0; //输入的号码

char key[50]={0},ok[50]={0};

printf("Please input how much numbers:");

for(i=0;en==0;i++) //要输入数字的位数

{jian=getch();

printf("%c",jian);

if(jian>47&&jian<58)time=time\*10+jian-48; //ASCII转十进制

else if(jian==13){en=1;printf("\n");

printf("You have %d numbers\n",time);}

}

Qm=time; //确定画图循环次数

for(i=0;i<time;i++)

{key[i]=jianru(); //读取键值的ASCII码，并转换成对应的数字和符号

DTMF(key[i]); //根据相应高低频生成相应DTMF波

ok[i]=Goertzel(); //戈泽尔运算，求高低频率，确定对应的输入，并画图

Q++;

}

printf("\nThe numbers are:\n\t\t ");

for(i=0;i<time;i++)printf("%d",key[i]);

printf("\n");

return 0;

}

//Goertzel算法

char Goertzel(void)

{

int n,k,k1=0,k2=0,loop; //k1、k2是两个幅度最大的频率。

double Re[N]={0},In[N]={0},output[N]={0},max=0,cimax=0,f1=0,f2=0,temp; //max和cimax为幅值最大和次之频率的值

char ok=0,tu[30];

FILE \*fp;

unsigned char header[1078]; //图像头文件和像素点

unsigned char pixel[200][3000];

int i,j; //像素点的高、宽

int m=0; //归一化的频率，即k的值

double l; //归一化因子，作图时能在有限的纵向空间显示图

strcpy(tu,"d:\\DTMF\\outputx.txt");

for(k=0;k<N;k++)

{for(n=0;n<N;n++) //使用欧拉公式将复指数化为三角函数，再分别计算实部和虚部。

{Re[k]+=input[n]\*cos(2\*PI\*n\*k/N);

In[k]+=input[n]\*sin(2\*PI\*n\*k/N);}

output[k]=sqrt(Re[k]\*Re[k]+In[k]\*In[k]);

//计算幅度谱，由幅度谱确定功率最大的频点

if(k<(N/2)) //只计算前半部分，找第一个最大值频点

if(output[k]>max)

{k1=k;

max=output[k];

}

}

for(k=0;k<(N/2);k++) //找第二个最大值频点

{if(output[k]!=max)

if(output[k]>cimax)

{ k2=k;

cimax=output[k];}

}

//将采样频点转化为实际频率

f1=((double)k1)\*8000.0/((double)N);

f2=((double)k2)\*8000.0/((double)N);

if(f1>f2)

{ temp=f1;

f1=f2;

f2=temp;}

//printf("\n%f\t%f\t%d\t%d\t%f\t%f\n",max,nmax,k1,k2,f1,f2);

for(k=0;k<12;k++) //最终分析频谱结果

{if(fabs(f1-(double)dipin[k])<20.0&&fabs(f2-(double)gaopin[k])<20.0)

{if(k<10) ok=k+48;

else if(k==10) ok=42;

else if(k==11) ok=35;

break;

}

}

//将计算后需要画图的文件先存储，画图时再读取。

tu[14]=ok;

if((fp=fopen(tu,"wb"))==NULL)printf("\nsorry,can not open output.dat\n");

fwrite(&output,sizeof(double),N,fp);

fwrite(&max,sizeof(double),1,fp);

fwrite(&cimax,sizeof(double),1,fp);

fclose(fp);

if((fp=fopen("d:\\DTMF\\pp.bmp","rb"))==NULL)

{printf("\nsorry,can not open 123.bmp\n");}

fread(&header,1,1078,fp);

fread(&pixel,1,600000,fp);

fclose(fp);

//根据输入的键值的ASCII码读取相应的文件进行画图

if(k==48)fp=fopen("d:\\DTMF\\output0.txt","rb");

if(k==49)fp=fopen("d:\\DTMF\\output1.txt","rb");

if(k==50)fp=fopen("d:\\DTMF\\output2.txt","rb");

if(k==51)fp=fopen("d:\\DTMF\\output3.txt","rb");

if(k==52)fp=fopen("d:\\DTMF\\output4.txt","rb");

if(k==53)fp=fopen("d:\\DTMF\\output5.txt","rb");

if(k==54)fp=fopen("d:\\DTMF\\output6.txt","rb");

if(k==55)fp=fopen("d:\\DTMF\\output7.txt","rb");

if(k==56)fp=fopen("d:\\DTMF\\output8.txt","rb");

if(k==57)fp=fopen("d:\\DTMF\\output9.txt","rb");

if(k==42)fp=fopen("d:\\DTMF\\output#.txt","rb");

if(k==35)fp=fopen("d:\\DTMF\\output\*.txt","rb");

if((fp)==NULL)printf("\nsorry,can not open output.txt\n");

fread(&output,sizeof(double),N,fp);

fread(&max,sizeof(double),1,fp);

fread(&cimax,sizeof(double),1,fp);

fclose(fp);

if(cimax>max)

max=cimax;

l=190.000/max;

for(i=0;i<N;i++)

{output[i]=(int)(output[i]\*l);} //归一化显示

//在图上画出多个数字的频谱

for(i=Q\*300+0,m=0;i<Q\*300+256&&m<256&&Q<Qm;i++,m++)

{for(j=5;j<5+output[m];j++) pixel[j][i]=0; }

if((fp=fopen("d:\\DTMF\\pp.bmp","wb"))==NULL){

printf("\nsorry,can not open 123.bmp\n");}

fwrite(&header,1,1078,fp);

fwrite(&pixel,1,600000,fp);

fclose(fp);

return ok;

}

//读每次输入的号码

char jianru()

{ long w;

char key,en=1;

int number;

do{ key=getch();

printf("%c",key);

if(key>47&&key<58)

{ en=0;number=key-48;}

else if(key==42)

{ en=0; number=10;}

else if(key==35)

{ en=0; number=11;}

else printf("input error!\n");

Beep(dipin[key-48],300);//输入每个数字时候由系统发高频与低频声音

//Beep函数的主要功能是输出前面的变量频率的声音，后面的变量是输出的时间由于低频信号声音比较低沉，故声音发出时间较短即可

for(w=0;w<4125000;w++){}//此处作为延时，低音输出后，延迟过后可以输出下一个声音

Beep(gaopin[key-48],400);//由于高频声音较为尖锐，为了区分声音信息，需要把时间设长一点。

}

while(en);

return number;

}

//构造DTMF信号

void DTMF(char number)

{ float f1,f2;

int i;

f1=(float)dipin[number];

f2=(float)gaopin[number];

for(i=0;i<N;i++)

input[i]=(float)(sin(2.0\*PI\*f1\*i/8000)+sin(2.0\*PI\*f2\*i/8000));

printf("gp=%d,dp=%d\n",gaopin[number],dipin[number]);

//输入每个数字时候输出他的低频与高频

return;

}