

به نام خدا

هارمونیک سنج دیجیتال

نویسندگان

محمد نحوی، مهدی خراسانی

کلمات کلیدی

FFT، هارمونیک، VB، پورت موازی.

چکیده

در این مقاله توسط یک ADC سیگنال از پورت موازی خوانده شده و بر روی مونیتور نشان داده میشود، سپس طیف فرکانسی آن به صورت نمودار میله ای رسم می شود.



مقدمه :

اگر چه تمامی دنیای اطراف ما آنالوگ (پیوسته) است ولی دنیای دیجیتال و کامپیوتر نیز با سرعت روز افزونی رو به پیشرفت و توسعه است. کامپیوتر جامع ترین دستگاه دیجیتال می باشد که معمولا تمامی وسائل دیگر قابلیت اتصال به آن را دارا می باشند. کامپیوتر های خانگی یا همان PC در همه جا یافت می شوند، و امروزه تقریبا تمامی مردم با آن سروکار دارند. این عمومیت ما را بر آن داشت تا سعی در استفاده هر چه بیشتر از این پردازنده همه کاره داشته باشیم. اسیلوسکوپ و کامپیوتر دو ابزاری هستند که یک مهندس الکترونیک همواره با آنها سروکار دارد. این نیاز ما را بر آن داشت تا با صرف هزینه کم و با سافت سفت افزاری کوچک، ارزان و طرامی نرم افزاری با کاربری بسیار ساده یک اسیلوسکوپ دیجیتال و اسپکتوراه آنالیز را توسط کامپیوتر پیاده سازی کرده و در اختیار کاربران قرار دهیم.

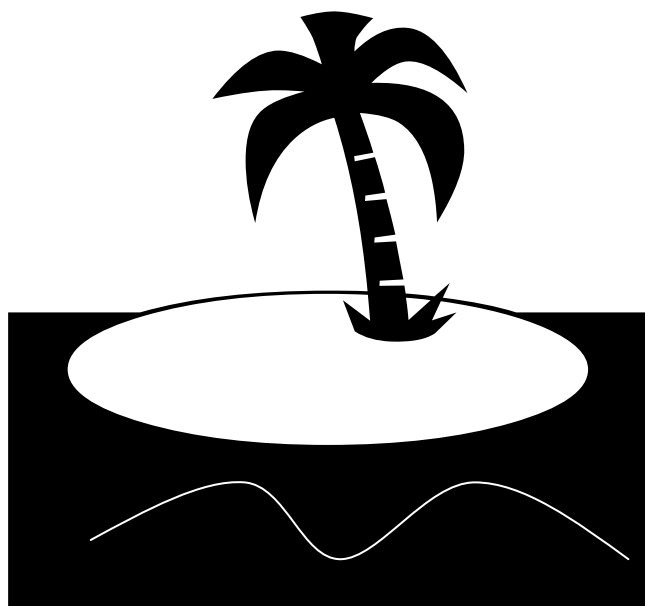
امید است توضیحات صفحات بعد دانشجویان عزیز را برای تشریح هر چه بیشتر این پروژه یاری کند.

از زحمات اساتید ممتزم جناب آقای دکتر جواد مدادینیا و جناب آقای مهندس مجید بقایی نژاد که در این پروژه ما را یاری کردند کمال تشکر را دارد. از همکاری خانمها بفتیاری و امینی برای کمک در تحلیل فرمول FFT تشکر می کند.

به امید روزی که نام ایران بر اوج قله افتخارات علمی دنیا طنین انداز شود.

عملکرد کلی اسکوپ و اسپکتورام آنالایزر دیجیتال :

این پروژه با توجه به نام آن می بایست فقط توسط میکرو پیاده سازی شود ولی به دلیل کمبود وقت و همچنین نبود برخی از قطعات قبل از عید ما ناچار به پیاده سازی آن توسط کامپیوتر از طریق پورت موازی شدیم . در زیر توضیح مختصری در رابطه با نحوه کار نرم افزار کامپیوتری آورده شده است . عملکرد کلی این پروژه به اینصورت است که ابتدا توسط یک ADC نمونه ای به مدت یک ثانیه از موج مورد نظر گرفته و در همین حین آن را نشان می دهد و سپس با زدن یک کلید طیف فرکانسی آن را رسم می کند . با نوشتن شماره هارمونیک مورد نظر می توان فرکانس و دامنه آن را بدست آورد . برای بدست آوردن هارمونیک های موج مورد نظر از سری فوریه استفاده شده است . البته به دلیل اینکه سری فوریه به وسیله انتگرال بسیار زمان بر می باشد ، از روش سری فوریه گیری سریع یا همان FFT استفاده شده است .



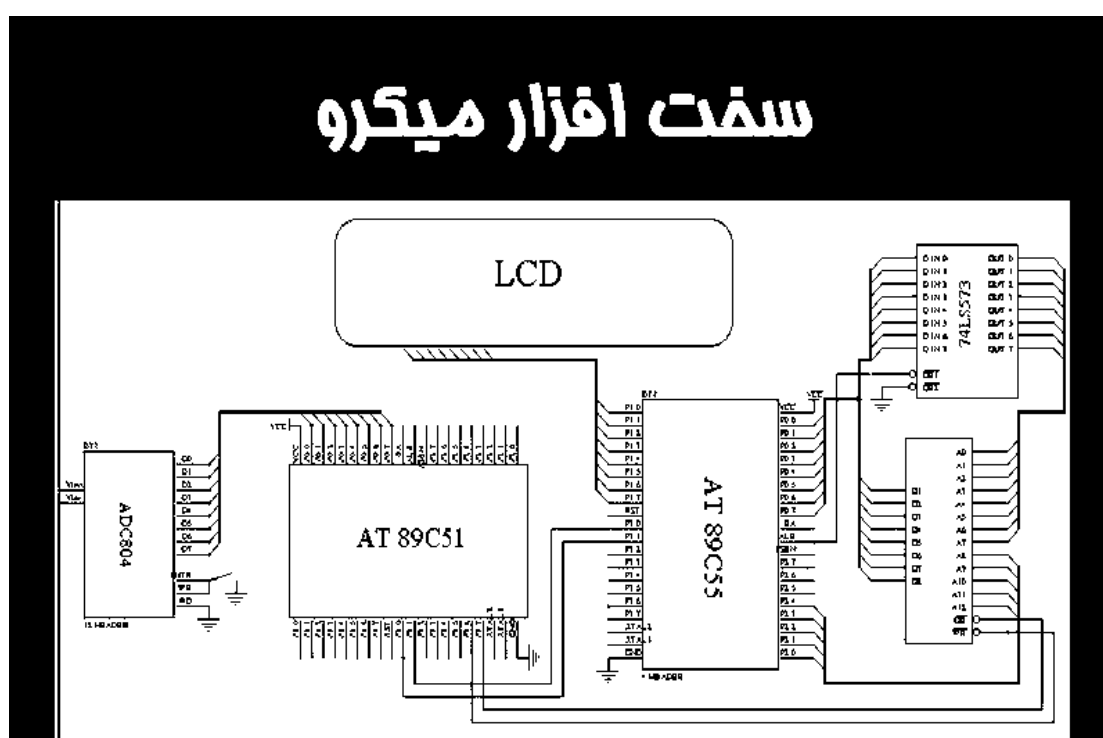
سخت افزار

سخت افزار :

سخت افزار کامپیوتری این پروژه شامل یک ADC و مدار بایاس آن و همچنین یک کانکتور پورت موازی می باشد . برای اینکه بتوان از حداکثر فرکانس پورت موازی (حدود 1MHZ) استفاده کرد ، باید از ADC استفاده کرد که بتواند با فرکانس بسیار بالا نمونه برداری کند .

به منظور استفاده از هارمونیک سنج این پروژه توسط میکرو نیز پیاده سازی شده است .

سخت افزار مربوط به میکرو در شکل زیر نشان داده شده است :



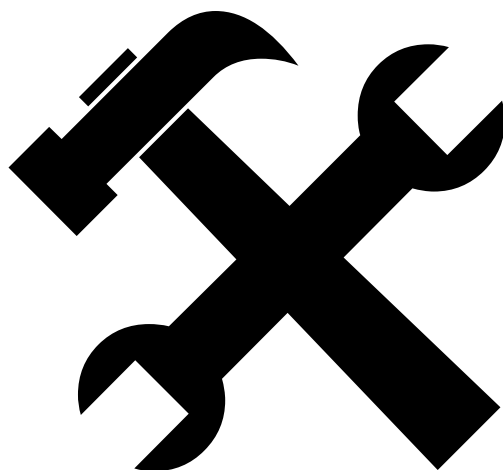
برای اینکه بتوان این پروژه را توسط میکرو پیاده سازی کرد نیاز به RAM بالا برای ذخیره نمونه های گرفته شده و همچنین یک نمایش گر LCD برای نمایش خروجی می باشد .

این سخت افزار شامل دو عدد میکرو یک RAM خارجی 8KB و مدار بایاس آن و یک LCD. برای نشان دادن خروجی می باشد .

به دلیل محاسبات اعشاری استفاده شده در نرم افزار میکرو فایل HEX آن در حدود 20KB بود و برای حل این مشکل از میکرو ۸۰۵۵ که دارای 20KB ROM داخلی می باشد استفاده شده است .

برای ذخیره نمونه های گرفته شده و همچنین انجام محاسبات روی آنها ناگزیر به استفاده از RAM خارجی 8KB شدیم و همچنین مدار بایاس آن که متشکل از لچ ۵۷۳ می باشد . برای تبدیل داده های آنالوگ به ارقام دیجیتال از ADC 804 استفاده شده است . برای نمایش خروجی از یک LCD کارکتری استفاده شده است . برای انتخاب یک هارمونیک و در خواست دیدن فرکانس و دامنه آن می توان از یک کیبرد نیز بهره برد که به علت کمبود وقت و همچنین کمبود ROM میکرو ۸۰۵۵ از این کار صرف نظر شده است .

در کنار میکرو ۸۰۵۵ از یک میکرو ۸۰۵۱ نیز برای افزایش پورت استفاده شده است . علت اینکه از PPI استفاده نشده است این است که اولاً برنامه نویسی آن مشکل است و ثانیاً قیمت آن چندان تفاوتی با میکرو ۸۰۵۱ ندارد و همچنین برای اتصال دو میکرو از ارتباط سریال استفاده شده است که هم برنامه نویسی آن راحتتر می باشد و همچنین فقط ۲ بیت از هر میکرو را اشغال می کند .



نرم افزار

نرم افزار پروژه:

با توجه به این نکته که به دست آوردن هارمونیک های تابع توسط فرمول FFT در دروس کارشناسی بررسی می شود ، ما برای به دست آوردن فرمول و تحلیل آن دچار مشکلات زیادی شدیم . در پایان پس بدست آوردن فرمول و امتحان روش های مختلف و اطمینان از روش استفاده شده از فرمول ، توانستیم برنامه FFT را به زبان C بنویسیم .

فرمول ریاضی FFT:

$$F(U) = \frac{1}{n} \sum_{x=n-1}^{x=0} \left(F(x) \times e^{\frac{-j \times 2 \times \pi \times x \times u}{n}} \right)$$

در فرمول بالا U فرکانس مورد نظر است .

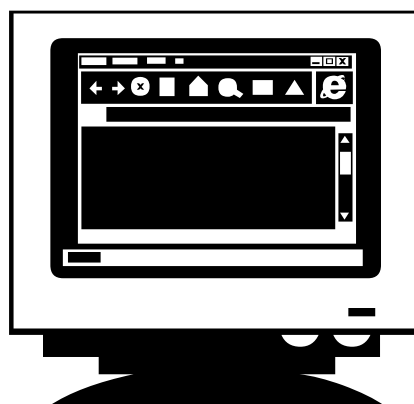
F(u) دامنه فرکانس یا هارمونیک مورد نظر می باشد .

N تعداد نمونه های گرفته شده می باشد .

X شماره نمونه ها می باشد .

F(X) دامنه هر نمونه می باشد .

البته فرمول ، تحلیل آن و مثال هایش در کتب کارشناسی آورده شده است که خارج از حوصله این مقاله است . ما در اینجا روش استفاده از فرمول و تحلیل برنامه نوشته شده را تشریح می کنیم .



نرم افزار نوشته شده به زبان C :

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<dos.h>
#include<math.h>
#include <iostream.h>
#include <complex.h>
#include <graphics.h>
#include <dos.h>
#define pi 3.141592653589793
//-----
double uf=0;
double s1[1000];
int numspec=1000;
double hurmony[1000]; // unsigned
double rhurmony[500]; // unsigned
void fory (void);
//-----
void main(void)
{
    int graphdriver = DETECT, graphmode;
    initgraph(&graphdriver, &graphmode, "c:\\tc\\bgi");
    int com;
    int j,i;
    com=0;
    //-----
    uf=0;
    for(i=0;i<1000;i++)
    {
        s1[i]=(50*sin(100*pi*uf*2));
        uf=uf+0.001;
        putpixel(i,(100+s1[i]),15);
    }
    //-----

    fory();
    for(i=0;i<500;i++)
```

```

    rhurmony[i]=hurmony[i]+hurmony[1000-i];
getch();
uf=0;
for(i=0;i<500;i++)
{
    printf(" * %lf * * %lf *",rhurmony[i],uf);
    uf=uf+1;
    printf(" \n ");
    delay(50);
}

getch();

} end of main
//-----
void fory (void)
{
    double kc=0;
    double ks=0;
    int i;
    int j;
    for(i=0;i<numspec;i++) //---- for u=0....u=n-1
    {
        j1=0;
        kc=0;
        ks=0;
        for(j=0;j<numspec;j++)//----- for x=0... x=n-1
        {
            kc=kc+cos((2*pi*j*i)/numspec)*s1[j];
            ks=ks+sin((2*pi*j*i)/numspec)*s1[j];
        }
        hurmony[i]=(sqrt(pow(kc,2)+pow(ks,2))/numspec);
    }
}
//-----

```

با توجه به برنامه بالا برای محاسبه هارمونیک ها از یک تابع با نام Fory استفاده شده است . در این تابع برای پرهیز از استفاده از روابط شی گرا در C از فرمول ریاضی:

$$e^{ju} = \cos(u) - j\sin(u)$$

برای محاسبات مختلط استفاده شده است. به این ترتیب که تمامی اعداد حقیقی در سیگما با هم جمع شده و همچنین تمامی قسمت های موهومی نیز با هم جمع شده و سپس اندازه آن محاسبه شده و به عنوان عدد نهایی این مرحله از فرمول در محاسبات بعدی استفاده شده است.

متغیر های استفاده شده در تابع Fory به ترتیب زیر می باشد :

KC : برای نگه داری مجموع اعداد حقیقی درون سیگما می باشد .

KS : برای نگه داری مجموع اعداد موهومی درون سیگما می باشد .

در این تابع از دو حلقه FOR استفاده شده است که حلقه اول برای محاسبه $F(u)$ ها می باشد (U از ۰ تا ۱۰۰۰) و حلقه داخلی برای محاسبه دامنه هر $F(u)$ (از $X=0$ تا $X=N-1$) می باشد .

تابع MAIN :

در این تابع برای اطمینان از صحت کارکرد فرمول و روش استفاده از آن بوسیله تابع SIN در C یک تابع سینوسی به فرمول داده شد و همچنین بوسیله حلقه های FOR یک تابع مربعی نیز تولید شد و سپس خروجی اولیه در آرایه Hurmony و سپس خروجی نهایی در آرایه Rhythm ذخیره شد که خروجی آن مطابق انتظار بود .

نکاتی که حتما باید در نظر گرفته شوند :

- ۱- همواره باید فرکانس نمونه گیری حتما با تعداد نمونه ها برابر باشند . مثلا اگر شما هر 1MS یک نمونه می گیرید حتما باید به تعداد ۱۰۰۰ عدد نمونه گرفته شود . در غیر این صورت خروجی آن خروجی مورد نظر نخواهد بود .
- ۲- همچنین باید در نظر گرفت که اگر شما با فرکانس 1KHZ نمونه گیری می کنید حداکثر می توانید هارمونیک های تا فرکانس 500HZ را محاسبه کنید .
- ۳- نکته آخر اینکه با توجه به قالب فرمول اگر شما ۱۰۰۰ نمونه می گیرید دامنه فرکانس ۹۹۹ مکمل دامنه فرکانس ۰ می باشد و به همین ترتیب برای فرکانس های دیگر . به عنوان مثال اگر

هارمونیکهای یک موج سینوسی یا کسینوسی با دامنه ۵۰ و فرکانس 100Hz را با شرط اینکه فرکانس نمونه گیری 1KHz باشد و همچنین تعداد ۱۰۰۰ نمونه گرفته شده باشد را بدست آوریم ، در خروجی یک هارمونیک با فرکانس ۱۰۰ و دامنه ۲۵ و یک هارمونیک با فرکانس ۹۰۰ و دامنه ۲۵ خواهیم داشت ، که با جمع کردن این دو دامنه ، دامنه هارمونیک این موج در فرکانس 100Hz به دست می آید . البته با توجه به این نکته که موجهای کسینوسی و سینوسی فقط دارای یک هارمونیک می باشند .

به منظور استفاده بهینه از پروژه در مکان های گوناگون و با توجه به این نکته که در هر زمان نمی توان به کامپیوتر دسترسی داشت این پروژه توسط میکروکنترلر و نمایش گر LCD نیز پیاده سازی شده است . به دلیل استفاده از متغیرهای اعشاری در برنامه میکرو که کار برنامه نویسی به زبان اسمبلی را بسیار دشوار می کرد از یک کامپایلر زبان C به کد HEX به نام فرانکلین استفاده شده است .



نرم افزار نوشته شده به زبان فرانکلین :

```
//          IN THE NAME OF GOD
#include<stdio.h>
#include<reg51.h>
#include<math.h>
#define pi 3.141592653589793

//-----

#define numhurmony 1000

float k;
float s1[1000];
char ok;
int rhurmony[500]={0};
char show[3];
//----- for fory

int numspec=1000;//----- tedad nemoneha
char spec[1000]={0};//-----nemoneh ha
float hurmony[1000]={0};//----- f(u)
float uf=.001;//----- u= u + uf 1/(numspec*^t)
int counter=0;

//-----
void fory (void);
//----- external int
void external(void) interrupt 0
{
    SBUF=0xff;
}

//----- serial func

void serial(void) interrupt 4
{
    if(RI)
    {
        if(counter<1000)
        {
            spec[counter]=SBUF;
            counter++;
        }
        else
        {
            counter=0;
        }
    }
}
```

```

        ok=50;
    }
    R≠0;
}
if(TI)
    TI=0;
}

//----- for LCD

    sbit clk = P3^3;
    sbit rw  = P3^4;
    sbit rs  = P3^5;
//-----

void lcd_cmd(char);
void lcd_dly(void);
void lcd_show(char);
void lcd_set(void);

//-----

void fory(void);

//-----

int i,j;
char temp;
//*****
void main(void)
{
    char i,j;
    char k,k1;
//-----
    TMOD=0x20;

    SCON=0x50;

    TH1=-3;

    IE= 0x99;

    ok=0;

    lcd_set();
//-----
l1:
    if(ok==50)

```

```

{
    fory();
    for(i=0;i<500;i++)
        r hurmony[i]=hurmony[i]+hurmony[999-i];

    for(i=0;i<500;i++)
    {
        lcd_cmd(0x8b);
        lcd_show('H');
        lcd_show(i);
        k1=i;
        for(j=2;j>=0;j--)
        {
            k=k1%10;
            show[j]=k;
            k1=k1/10;
        }
        for(j=0;j<3;j++)
            lcd_show(show[j]);

//-----
        lcd_show('F');
        k1=r hurmony[i];
        for(j=2;j>=0;j--)
        {
            k=k1%10;
            show[j]=k;
            k1=k1/10;
        }
        for(j=0;j<3;j++)
            lcd_show(show[j]);

        lcd_dly();
        lcd_dly();
        lcd_dly();
        lcd_dly();
        lcd_dly();
        lcd_dly();
        lcd_dly();

    }
    ok=0;
}
goto l1;
}
//-----
void fory (void)
{

```

```

double kc=0;
double ks=0;
int i;
float j1=0;
int j;
for(i=0;i<numspec;i++) //---- for u=0....u=n-1
{
    j1=0;
    kc=0;
    ks=0;
    for(j=0;j<numspec;j++)//----- for x=0... x=n-1
    {
        kc=kc+cos((2*pi*j*i)/numspec)*s1[j];
        ks=ks+sin((2*pi*j*i)/numspec)*s1[j];
    }
    hurmony[i]=(sqrt(pow(kc,2)+pow(ks,2))/numspec);
}
}
//-----
void lcd_cmd(char set)
{
    P1 = set;
    rs = 0;
    rw = 0;
    clk = 1;
    clk = 0;
    lcd_dly();
}
//-----
void lcd_show(char show)
{
    P1=show;
    rs=1;
    rw=0;
    clk=1;
    clk=0;
    lcd_dly();
}
//-----
void lcd_set(void)
{
    lcd_cmd(0x38);
    lcd_cmd(0x0c);
    lcd_cmd(0x06);
}
//-----
void lcd_dly(void)
{

```

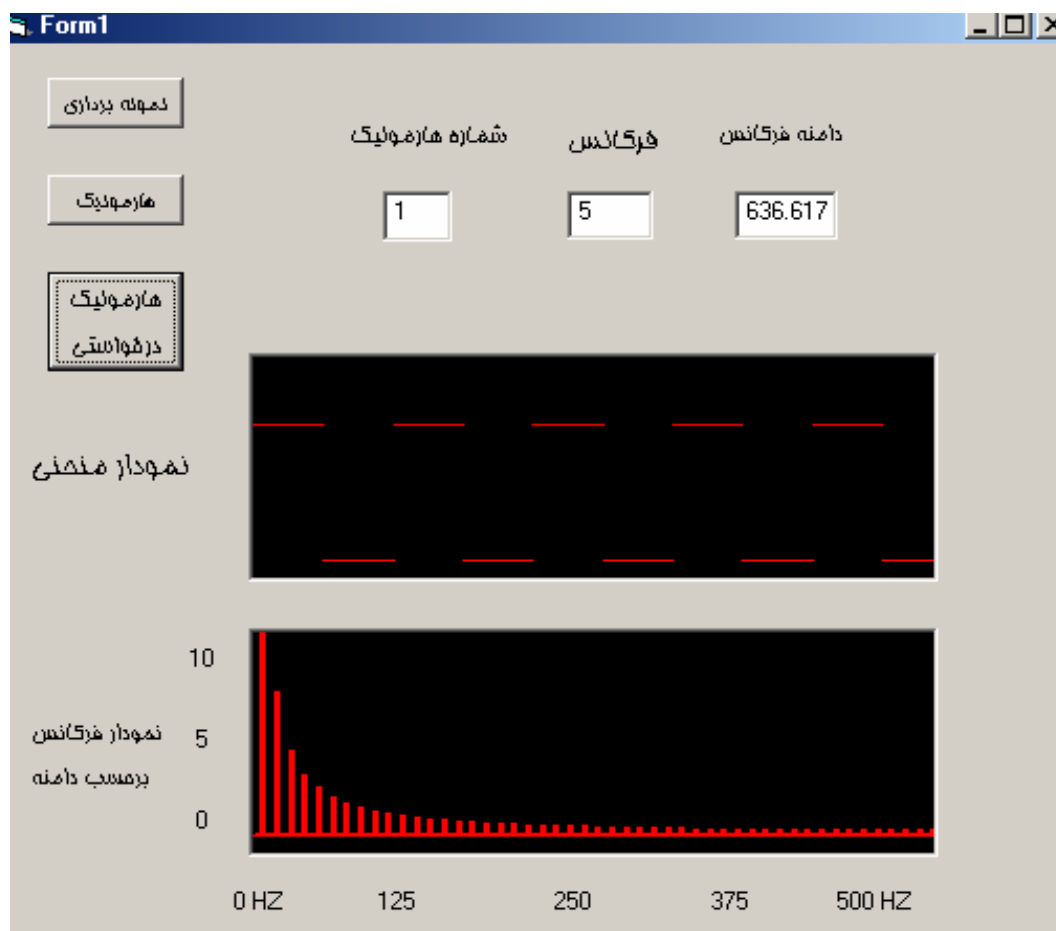


```
char i,j;  
for(i=0;i<50;i++)  
    for(j=0;j<50;j++)  
        {}  
}  
//-----
```

نرم افزارهای تشریح شده در بالا به زبان C می باشند ، به منظور ایجاد محیط گرافیکی مناسبتر با قابلیت های مورد نظر و همچنین برای راحت تر شدن کاربرد آن برای کاربر این نرم افزار به زبان VBASIC نیز نوشته شده است .



قالب کلی نرم افزار به زبان VB :



در نرم افزار ترجمه شده به زبان VB سه کلید روی فرم اصلی قرار دارد که نامهای آن (نمونه برداری ، هارمونیک و هارمونیک درخواستی) می باشد . کلید اول (نمونه برداری) به مدت یک ثانیه از سیگنال مورد نظر نمونه برداری کرده و همزمان آن را نشان می دهد .

پس از آن با زدن کلید هارمونیک ، هارمونیک های موج تا فرکانس 499HZ و با دامنه های مشخص رسم می شود .

هارمونیک درخواستی کلیدی است که برای اطلاع دقیق از فرکانس و دامنه یک هارمونیک به کار می رود . با نوشتن شماره هارمونیک درخواستی در جایگاه (شماره هارمونیک) و زدن این کلید ، فرکانس و دامنه آن به ترتیب در جایگاه های (فرکانس و دامنه فرکانس) درج می شود .

در زیر هر یک از رویدادهای نوشته شده در نرم افزار را به طور جداگانه بررسی می کنیم .

رویداد Command1_Click :

این رویداد به منظور شروع نمونه برداری از موج فراخوانی می شود . در حالت آزمایشی برای تشخیص از صحت کارکرد برنامه دو موج یکی سینوسی و دیگری پالس مربعی به طور نرم افزاری در آرایه S1 ذخیره شد و با زدن کلیدهای دیگر ، نرم افزار به طور دستی تست شد .
بعد از اطمینان از صحت کارکرد نرم افزار برنامه اصلی آن در تایمر نوشته شد .

رویداد Timer1_Timer :

در این رویداد با یک تاخیر حدود 1 ms به مدت یک ثانیه از موج نمونه برداری و همزمان در صفحه نمایش نشان داده می شود .

رویداد Fory :

این رویداد به منظور انجام محاسبات سری فوریه و به دست آوردن هارمونیک ها می باشد .

رویداد Command2_Click :

در این رویداد پس از فراخوانی رویداد Fory و ذخیره سازی دامنه فرکانسها نموداری برحسب دامنه و فرکانس نمایش داده می شود .

رویداد Command3_Click :

در این رویداد پس از خواندن شماره هارمونیک درخواستی از Text1 ، شروع به جستجو در آرایه Rharmony برای بدست آوردن دامنه و فرکانس هارمونیک درخواستی می کند . پس از یافتن آن دامنه و فرکانس آن به ترتیب در Text2 و Text3 درج می شود .

رویداد Delay :

این رویداد برای ایجاد تاخیر به هنگام خواندن پورت می باشد .

تابع Lpt_convert :

این تابع پورت موازی با آدرس 0x378 را تبدیل به پورت ورودی می کند ، اگر این کار به درستی انجام شود ، مقدار True و اگر این تبدیل به درستی انجام نشود مقدار False برگردانده می شود .

نکته :

باید توجه داشت که برای استفاده از متغیرهای در چندین رویداد و تابع باید این متغیر ها به صورت عمومی تعریف شود . و همچنین برای استفاده از دستورات ورودی و خروجی در VB باید DLL آن را به پروژه اضافه نمود . تمامی کارهای بالا در MODULES به صورت زیر نوشته شود :

```
Declare Sub vbOut Lib "win95io.dll" (ByVal nPort As Integer, ByVal nData As Integer)
```

```
Declare Function vbInp Lib "win95io.dll" (ByVal nPort As Integer) As Integer
```

```
-----'
```

```
Public Const lpt = &H378
```

```
Public Const pi = 3.14159265358979
```

```
Public s1(1000) As Double
```

```
Public uf As Double
```

```
Public numspec As Integer
```

```
Public hurmony(1000) As Double
```

```
Public rhurmony(500) As Double
```

```
Public t As Boolean
```

```
Public temp As Integer
```

```
Public Function lpt_convert() As Boolean
```

```
Dim i As Integer
```

```
i = vbInp(lpt + 2 + &H400)
```

```
vbOut (lpt + 2 + &H400), i Or 32
```

```
i = vbInp(lpt + 2)
```

```
vbOut (lpt + 2), i Or 33
```

```
End Function
```

نام DLL استفاده شده در این نرم افزار WIN95IO می باشد که با همین نام قابل دریافت از اینترنت می باشد . VBINP نام تابعی است که برای خواندن از پورت از آن استفاده می شود و VBOUT نام تابعی است که برای نوشتن در پورت از آن استفاده می شود . تابع LPT_CONVERT برای تبدیل پورت 0x378 به یک پورت ورودی می باشد . برای انجام این کار باید بیت پنجم بایت با آدرس 0x37A + 0x400 و همچنین بیت اول و پنجم بایت با آدرس 0x37A را در منطق یک نشانند .

توجه :

اگر در هنگام تبدیل پورت موازی به پورت ورودی پورت به جایی متصل باشد ، این تبدیل به درستی انجام نخواهد شد و تابع LPT_CONVERT مقدار False را بر می گرداند . برای حل این مشکل باید یک بار هنگامی که پورت متصل به مدار است بیت های آن به دور دستی خواند و در هنگام تست پورت در تابع LPT_CONVERT برای صحت ورودی شدن پورت به جای عدد ۲۵۵ عدد بدست آمده از مدار را ذخیره کرد (البته می توان در زمان اجرای تابع تبدیل کابل پورت را از مدار یا کامپیوتر جدا کرد) .

نرم افزار پروژه به زبان VB :

```
Private Sub Command1_Click()
Dim i As Integer
Dim j As Integer
Dim g As Double
Dim f As Integer
Dim z As Integer
Dim t As Byte
g = 0
f = 50
temp = 1
Timer1.Enabled = True
Picture1.BackColor = vbBlack
'----- sin
Dim k As Integer
k = 0
For i = 1 To 1000
s1(i) = 500 * Cos(2 * pi * f * g)
g = g + 0.001
k = k + 30
Picture1.PSet (i + k, 500 + s1(i) / 2), vbRed
'-----
For i = 0 To 800 Step 200
For j = 0 To 100
s1(i + j) = 0
Picture1.PSet ((i + j) * 5, 500 + s1(i + j)), vbRed
Next j
For j = 100 To 200
s1(i + j) = 1000
Picture1.PSet ((i + j) * 5, 500 + s1(i + j)), vbRed
Next j
Picture1.PSet (i * 5, 500 + s1(i)), vbRed
Next i
End Sub
'-----
Private Sub Command2_Click()
Dim i As Integer
Dim j As Integer
```

تابع سینوسی

تابع پالس

```

Dim z As Integer
Dim s As Integer
j = 0
s = 10
Picture1.BackColor = vbBlack
'-----

fory
For i = 1 To 500
rhurmony(i) = hurmony(i) + hurmony(1000 - i)
Next i
For i = 1 To 5000 Step 10
For z = (5 * rhurmony(j)) To 0 Step -1
For s = 1 To 30
Picture1.PSet (i + s, (1500 - z)), vbRed
Next s
Next z
j = j + 1
'-----

If rhurmony(j) <= 1 Then
s = 1
Else:
s = 10
End If
'-----

Next i
End Sub

Private Sub fory()
Dim kc As Double
Dim ks As Double
Dim i As Integer
Dim j As Integer
numspec = 1000
For i = 0 To numspec
kc = 0
ks = 0
For j = 0 To numspec
kc = kc + Cos((2 * pi * i * j) / numspec) * s1(j)
ks = ks + Sin((2 * pi * i * j) / numspec) * s1(j)
Next j

```

**رویداد سری
فوری**

```
hurmony(i) = (Sqr((kc ^ 2) + (ks ^ 2))) / numspec
```

```
Next i
```

```
End Sub
```

```
'-----
```

```
Private Sub Command3_Click()
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Dim h As Integer
```

```
Dim n As Integer
```

```
Dim z As String
```

```
z = Text1.Text
```

```
If z <> "" Then
```

```
h = Text1.Text
```

```
n = 1
```

```
'-----
```

```
For i = 1 To 500
```

```
If rhurmony(i) >= 1 Then
```

```
If n = h Then
```

```
Text2.Text = i
```

```
Text3.Text = rhurmony(i)
```

```
Exit Sub
```

```
End If
```

```
n = n + 1
```

```
End If
```

```
Next i
```

```
Text2.Text = "no"
```

```
Text3.Text = "no"
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
lpt_convert
```

```
temp = 1
```

```
t = False
```

```
Picture1.BackColor = vbBlack
```

```
End Sub
```

رویداد مناسبه فرکانس

و دامنه ی هارمونیک ها


```

'-----
Private Sub Timer1_Timer()
Dim r As Integer
Dim r1 As Double
Dim k As Integer
Timer1.Enabled = False
k = 0
For temp = 0 To 1000
s1(temp)= vbInp(lpt)
k = k + 30
Picture1.PSet (temp + k, 500 + (s1(temp))) , vbRed
delay
Next temp
End Sub
'-----

```

این رویداد برای خواندن
نمونه ها از پورت است

```

Private Sub delay()
Dim i As Integer
Dim j As Integer
For i = 0 To 1000
For j = 0 To 300
Next j
Next i
End Sub

```

رویداد تأخیر

نکته :

برای نمایش بهتر نمودار منحنی های مختلف با فرکانس های گوناگون از دو کلید به عنوان TIME/DIV و VOLT/DIV استفاده شده است .

پل ارتباطی دریافت داده ها در این پروژه پورت موازی می باشد . در زیر توضیحات مختصری در مورد نحوه عملکرد پورت موازی آورده شده است .

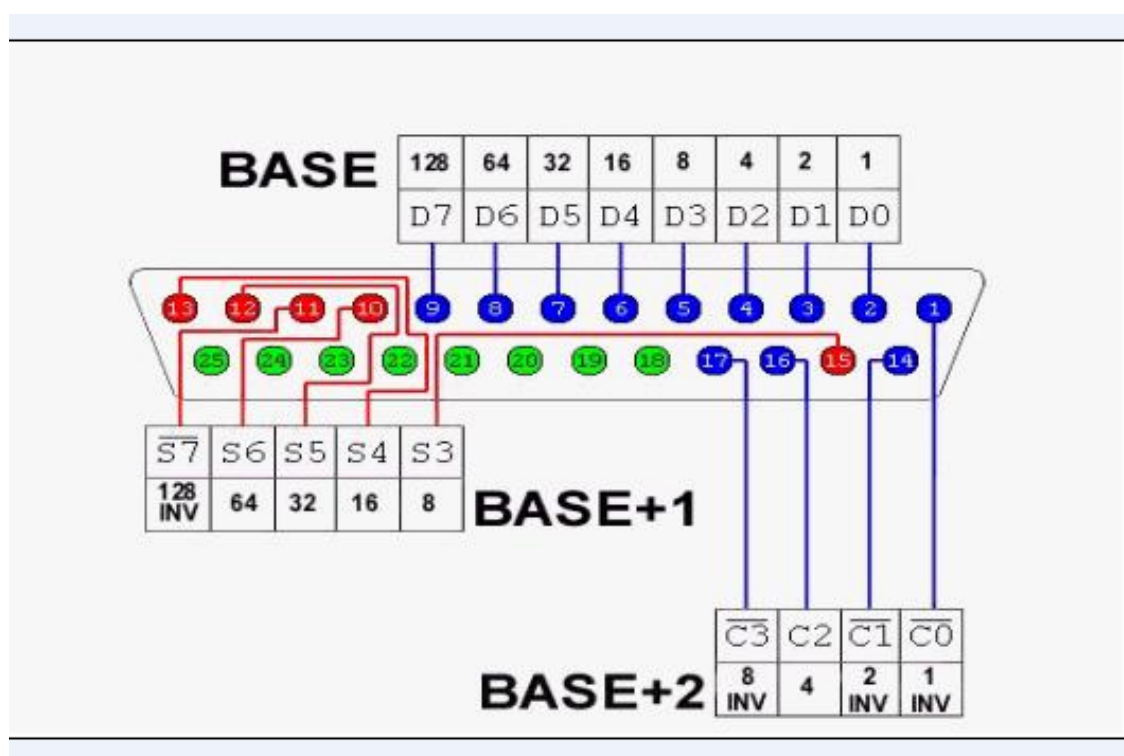
پورت موازی :

این پورت دارای 25 پین است و BIOS در IBM PC ، تا 4 چاپگر را اجازه وصل می دهد ، یعنی LPT1 تا LPT4 . BIOS INT 17H به برنامه نویس امکان مقدار دهی اولیه به پورت های چاپگر را می دهد تا کارکترها را در چاپگر بنویسد ، وضعیت چاپگر را بررسی می کند و دیگر اعمال را انجام می دهد. پایه های پورت LPT را دسته بندی نموده ایم و در زیر توضیح می دهیم:

خطوط داده و زمین ها :

پایه های ورودی DATA1 تا DATA8 مسیری 8 بیتی و موازی را برای داده ارسالی از PC به چاپگر فراهم می آورند و در واقع هر یک متعلق به یکی از خطوط داده می باشند.

پایه های چاپگر DB-25



در شکل بالا BASE عدد ۳۷۸ هگز و BASE+1 و BASE+2 نیز به ترتیب شماره ۳۷۹ هگز و 37a هگز میباشد . پورت ۳۷۸ در قدیم فقط برای نوشتن بود که برای استفاده بهینه از این پورت قابلیت ورودی

بودن این هشت بیت نیز در آن تعبیه شد. پورت شماره ۳۷۹ فقط ورودی است و پورت با شماره 37a هم ورودی و هم خروجی می باشد . توجه داشته باشید که ۳ بیت از بیت‌های این پورت در درون خود NOT دارد .

محدودیت های پروژه :

از محدودیت های این پروژه سرعت ADC استفاده شده است . اگر سرعت ADC استفاده شده بالا باشد می توان فرکانسهای بالاتری را نیز نمونه گیری کرد . نکته دیگر برای مقابله با نویز ایزوله کردن یا جدا سازی بیت های پورت موازی توسط اپتوکوپلر است . البته باید محدودیت پورت را نیز در نظر گرفت ، سرعت پورت موازی حداکثر 1MHZ است .
با استفاده از شکافهای PCI می توان سرعت پورت را تا 33 MHZ نیز ارتقا داد.

پیشنهاد :

روش تشخیص هارمونیک های یک موج دریافتی به وسیله یک فرمول ریاضی معروف به FFT می باشد . از این روش سری فوریه یا همان FFT برای تشخیص حروف در سیگنال های صوتی یا همان پردازش صدا نیز استفاده می شود . هر حرفی که از دهان خارج می شود شامل تعدادی هارمونیک با فرکانس و دامنه های مختلف می باشد ، اگر برای تمامی حروف یک DATA BASE بر حسب هارمونیک های به کار رفته برای هر حرف درست کرده و از سیگنال صوتی سری فوریه گرفته شود می توان حروف و در نتیجه کلمات گفته شده را به صورت متن استخراج کرد . به عنوان مثال اگر فرض کنیم حرف " آ " یی که از دهان خارج می شود شامل فرکانس های 1.5KHZ , 1KHZ , 3KHZ است اگر بهنگام تحلیل سیگنال صوتی با این فرکانسها برخورد کردیم به معنی حرف " آ " می باشد .

امید است این ایده روزنه ای برای کمک به دانشجویان در پروژه پردازش سیگنال های صوتی باشد .