

```
//SIDI1TDD.C TIME DOMAIN DISPLAY
```

```
// Marko Cebokli feb 2004 GNU/GPL licence
```

```
// www.gnu.org/licences/licences.html#GPL
```

```
// version 1 - software sample timing
```

```
// priklop na LPT port:
```

```
//          pin 11 channel 1 I      bit 7
//          pin 10 channel 1 Q      bit 6
//          pin 12 channel 2 I      bit 5
//          pin 13 channel 2 Q      bit 4
//          pin 18..25 ground
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <dos.h> //inportb
```

```
#include <time.h>
```

```
#include <graphics.h>
```

```
#include <conio.h> //getch() kbhit()
```

```
#include <stdlib.h> //rand()
```

```
double PI=3.14159265358979;
```

```
unsigned int sig[32767];
```

```
int xr[32767],xi[32767];
```

```
//-----
```

```
void vzorcenje(long n) //this one does the sampling
```

```
{ //array is written each time to equalize timing
```

```
int i;
```

```
long j;
```

```
unsigned char a;
```

```
for (j=0;j<n;j++) //131072
```

```
{
  i=(int)(j>>2);
```

```

a=inportb(0x379)>>4; //D25: *pin11 pin10 pin12 pin13 (pin15)
sig[i]=(sig[i]<<4)+a;
}
}

```

```
//-----
```

```
void hitrovz(long n) //fast sampling without bit concatenation
```

```

{
long j;

for (j=0;j<n;j++)
{
sig[j]=inportb(0x379);
}
}

```

```
//-----
```

```
void preflintaj1(long n, int k) //podatki iz "hitrovz" v real/imag vhodne podatke FFT
```

```

{
long j;
int w;
for (j=0;j<n;j++)
{
w=1;
if (k==1)
{
if ((sig[j]&0x40)==0x40) xi[j]=w; else xi[j]=-w; //Q1
if ((sig[j]&0x80)==0x80) xr[j]=w; else xr[j]=-w; //I1
//xr[j]=0;
}
else
{
if ((sig[j]&0x10)==0x10) xi[j]=w; else xi[j]=-w; //Q2
if ((sig[j]&0x20)==0x20) xr[j]=-w; else xr[j]=w; //I2
//xr[j]=0;
}
}
}
}

```

```
//-----
```

```

void checkspeed() //measure speed of computer (sample rate etc)
{
clock_t sta,mid,sto;
float srate,brate;

//printf("\n clktck=%8.2f \n",CLK_TCK);
sta=clock();
vzorcenje(131072); //do the sampling
mid=clock();
hitrovz(32767);hitrovz(32767);hitrovz(32767);hitrovz(32767);
sto=clock();
//printf("\n cas vzorcenja= %d",mid-sta);
printf("\n cas vzorcenja= %d",sto-mid);
srate=131072.0/(mid-sta)*CLK_TCK;
//printf("\n sampling rate= %8.0f /sec",srate);
srate=131072.0/(sto-mid)*CLK_TCK;
printf("\n sampling rate= %8.0f /sec",srate);
}

```

```
//-----
```

```

void autocor(int x[], int n, int ofs, float* k)
{
int i;

*k=0;
for (i=50;i<(n-50);i++)
{
*k=*k+x[i]*x[i+ofs];
}
*k=*k/(float)(n-100);
}

```

```
/**-----
```

```

int main()
{
int gdriver = DETECT, gmode, errorcode;
int i,n;
char irqmask;
float k;
long ii;

```

```
n=32767;
printf("\n\n-----\n");
checkspeed();
getch();

hitrovz(n);
preflintaj1(n,1);

irqmask=inportb(0x21);
outportb(0x21,irqmask+1);    //stop timer interrupt

initgraph(&gdriver, &gmode, "c:\\bc\\bgi");

do
{
clearviewport();
hitrovz(n);
preflintaj1(n,1);

//risanje casovnih slik (oscilogramov)

setcolor(GREEN);line(60,14,580,14);line(60,46,580,46);
for (i=64;i<576;i=i+2)line(i,29,i,31);
setcolor(WHITE);
moveto(64,15);
for (i=0;i<256;i++)
{
lineto(64+2*i,30-15*xr[i]);lineto(64+2*i+2,30-15*xr[i]);
}
setcolor(GREEN);line(60,54,580,54);line(60,86,580,86);
for (i=64;i<576;i=i+2)line(i,69,i,71);
setcolor(WHITE);
moveto(64,55);
for (i=0;i<256;i++)
{
lineto(64+2*i,70-15*xi[i]);lineto(64+2*i+2,70-15*xi[i]);
}
preflintaj1(n,2);
setcolor(GREEN);line(60,94,580,94);line(60,126,580,126);
for (i=64;i<576;i=i+2)line(i,109,i,111);
setcolor(WHITE);
moveto(64,95);
```

```

for (i=0;i<256;i++)
{
  lineto(64+2*i,110-15*xr[i]);lineto(64+2*i+2,110-15*xr[i]);
}
setcolor(GREEN);line(60,134,580,134);line(60,166,580,166);
for (i=64;i<576;i=i+2)line(i,149,i,151);
setcolor(WHITE);
moveto(64,135);
for (i=0;i<256;i++)
{
  lineto(64+2*i,150-15*xi[i]);lineto(64+2*i+2,150-15*xi[i]);
}

```

// risanje avtokorelacij

```

preflintaj1(n,1);
setcolor(GREEN);
moveto(50,200);lineto(150,200);lineto(150,300);lineto(50,300);lineto(50,200);
line(50,250,150,250);
moveto(50,250);setcolor(WHITE);
for (i=0; i<20; i++)
{
  autocor(xr,n,i-10,&k);
  lineto(50+5*i,250-50.0*k);lineto(55+5*i,250-50.0*k);
}
setcolor(GREEN);
moveto(200,200);lineto(300,200);lineto(300,300);lineto(200,300);lineto(200,200);
line(200,250,300,250);
moveto(200,250);setcolor(WHITE);
for (i=0; i<20; i++)
{
  autocor(xi,n,i-10,&k);
  lineto(200+5*i,250-50.0*k);lineto(205+5*i,250-50.0*k);
}
preflintaj1(n,2);
setcolor(GREEN);
moveto(350,200);lineto(450,200);lineto(450,300);lineto(350,300);lineto(350,200);
line(350,250,450,250);
moveto(350,250);setcolor(WHITE);
for (i=0; i<20; i++)
{
  autocor(xr,n,i-10,&k);
  lineto(350+5*i,250-50.0*k);lineto(355+5*i,250-50.0*k);
}

```

```
    }
setcolor(GREEN);
moveto(500,200);lineto(600,200);lineto(600,300);lineto(500,300);lineto(500,200);
line(500,250,600,250);
moveto(500,250);setcolor(WHITE);
for (i=0; i<20; i++)
    {
        autocor(xi,n,i-10,&k);
        lineto(500+5*i,250-50.0*k);lineto(505+5*i,250-50.0*k);
    }
for (ii=0;ii<50000000;ii++) i=2; //delay zanka, ker ni timer interrupta
}
while(kbhit()==0);

getch();

closegraph();

outportb(0x21,irqmask);

return 0;
}
```