



## LAMPIRAN

## OSC2.PAS

```
{ PROGRAM OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD
```

```
Author      : Sondag P H
Compiler    : Turbo Pascal 7.0
CPU        : Intel 486 Dlx2-66MHz
Operating System : DOS 6.22, WIN95/97
Abstrak     : Program ini merupakan osiloskop sederhana dengan memanfaatkan Sound card pada komputer multimedia. Frekuensi penyampelan yang digunakan sebesar 44100Hz dengan resolusi data 16 bit pada mode mono. Pemrograman menggunakan standar Sound Blaster Pro dan Sound Blaster 16. Sinyal diubah ke digital pada ADC Sound Card, kemudian data ini ditransfer ke memori menggunakan DMA dan ditampilkan ke layar.
Linker      : bmp2.pas, mousesb.pas, sb.pas, gb.pas, bmp.pas, wavfmt.pas
}
```

```
uses timer,dos,Graph,Crt;
{$M 64000,0,655360}
var warna          : FillPatternType;
    Gd, Gm, i2,tmp,tmp2 : Integer;
    P3,P,P2,a1,a2,a3,a4,rev : Pointer;
    Size,ii         : word;
    ext             : boolean;
    pl,vch1,vch2,pch1,pch2 : integer; {Variable volt/div & Position}
    tpd,ch1,ch2     : byte;
    data,dai2      : array[1..8192] of word;
    vmin,vmax,vmin2,vmax2,vref,vref2 : word;
    comp,comp2,max1,max2 : real;
    stereo,phase,run,spic,swav : boolean;
    vol            : byte;
    nama          : string[8];
    t;text;
{$i bmp2.pas}
{$i mousesb.pas}
{$i sb.pas}
{$i gb.pas}
{$i bmp.pas}
{$i wavfmt.pas}
function hasil(z:real):string;
var v:string;
begin
    Str(z:2:6,v);
    hasil:=v;
end;
```

```
{PROGRAM UTAMA}
```

```
begin
Gd := Detect;{Inialis Grafik VGA 640 x 480 }
InitGraph(Gd, Gm, '');{Buka mode grafik }
for i2:=1 to 8 do
    warna[i2]:=SFF;
cleardevice; {membersihkan layar}
SetColor(15);
setttextstyle(0,0,2);
outtextxy(200,1,'OSCILOSCOPE DIGITAL');{menggambar panel}
layar: {menggambar panel}
display(20,440,60,460,15,red,'exit');
display(100,440,220,460,1,7,' Single Trace');
display(240,440,360,460,1,7,' Dual Trace');
display(380,440,500,460,1,7,' Phase Trace');
display(540,20,600,40,15,0,' 1ms');
putImage(602,19,p^,NormalPut);
putImage(602,30,p2^,NormalPut);
outtextxy(545,50,'TIME/DIV');
```

```

display(540,70,600,90,15,0,' 5V');
putImage(602,69,p1,NormalPut);
putImage(602,80,p2,NormalPut);
outtextxy(545,100,'VOLT/DIV');
outtextxy(545,110,' ch 1');

display(540,125,600,145,15,0,' 5V');
putImage(602,123,p1,NormalPut);
putImage(602,135,p2,NormalPut);
outtextxy(545,155,'VOLT/DIV');
outtextxy(545,165,' ch 2');

putImage(530,179,a1,NormalPut);
putImage(530,221,a2,NormalPut);
putImage(580,179,a1,NormalPut);
putImage(580,221,a2,NormalPut);
outtextxy(540,265,'ch1');
outtextxy(590,265,'ch2');
putImage(532,275,a3,NormalPut);
putImage(574,275,a4,NormalPut);

display(530,320,570,360,1,7,"");
putImage(530,320,a4,xorput);
display(580,320,620,360,1,7,"");
display(590,330,610,350,1,6,"");
display(530,380,625,420,1,cyan,'Save to BMP');
display(530,430,625,470,1,cyan,'Save to WAV');
{Akhir pembuatan gambar panel}
Mousetest; {Test mouse & reset mouse}
init; {Inisial Sound Blaster}
run:=true; {inisialis nilai awal}
tanda:=false;
getmem(buffer,sizeof(data));{Pesan tempat pada memori untuk tempat data}
Assign(t,'OSC.CFG'); {membuka file OSC.CFG}
reset(t);
{readln(t.vol);}
sbrset; {reset sound blaster}
versi; {mengambil versi DSP}
port[mix]:=$0A;port[mixdata]:=$11;{Microphone Off}
port[mix]:=$0C;port[mixdata]:=$27;{Filter}
port[mix]:=$38;port[mixdata]:=vol and $F8;{Line in vol SB16}
port[mix]:=$39;port[mixdata]:=vol and $F8;{Line in vol SB16}
port[mix]:=$3d;port[mixdata]:=$10;{Line Control in}
port[mix]:=$3e;port[mixdata]:=$08;{Line Control in}
readln(t.vol);
port[mix]:=$3f;port[mixdata]:=vol;{Line in Gain}
port[mix]:=$40;port[mixdata]:=vol;{Line in Gain}
readln(t.vol);
port[mix]:=$2e;port[mixdata]:=vol;{Line in}
port[mix]:=$28;port[mixdata]:=vol;{CD}
port[mix]:=$43;port[mixdata]:=$00;{AGC}
port[$21]:=port[$21] and (255 xor (1 shl sb.irq));{Enable IRQ}
getintvec(sb.irq+8,rev);{simpan alamat irq asli}
setintvec(sb.irq+8,@baca);{set ke alamat program irq SB}
i:=seg(buffer^)shl 4+ofs(buffer^); {ambil alamat fisik buffer}
page:=(seg(buffer^)+ofs(buffer^))shr 4)shr 12;
if channel>3 then begin
  dma:=dma div 2;
  page:=page div 2;
end;
vch1:=300;vch2:=300;pch1:=0;
{vch1 & vch2 adalah kontanta u/ kalibrasi amplituda secara program }
max1:=5;

```

```

max2:=5;
readln(t,vref2);
vref:=vref2 div 2+256;
close(t);
{if (t1<4) and (t2<13) then vref:=128 else vref:=-514;}
tpd:=1;
stereo:=false;
repeat
if phase then tpd:=1;
MouseOn: {mouse On}
repeat
setViewport(0,0,520,420,true);
setcolor(15);
layer:{gambar background};
tanda:=false;
if run then begin
vmin:=Sfff;vmax:=S0;{set nilai awal vmax & vmin}
vmin2:=Sfff;vmax2:=S0;{set nilai awal vmax & vmin}
comp:=0;
baca2;{Program Sound blaster & DMA}
repeat {tunggu hingga ada interupsi}
if keypressed then
begin
setintvec(sb.irq+8,rev);{kembalikan alamat irq asli}
halt(1);{kembali ke system}
end;
until (tanda);
if not stereo then {mono}
for ii:=1 to dma-2 do begin
{mengambil data dari memori}
data[ii]:=memw[seg(buffer^):ofs(buffer^)+ii];
if data[ii]>vmax then vmax:=data[ii];{ambil nilai max}
if data[ii]<vmin then vmin:=data[ii];{ambil nilai min}
end else begin {stereo}
i2:=1;ii:=1;
repeat{mengambil data dari memori}
data[i2]:=memw[seg(buffer^):ofs(buffer^)+ii];
inc(ii);
dat2[i2]:=memw[seg(buffer^):ofs(buffer^)+ii] and Sfff0;
inc(ii);
{mengambil nilai max & min}
if data[i2]>vmax then vmax:=data[i2];
if data[i2]<vmin then vmin:=data[i2];
if dat2[i2]>vmax2 then vmax2:=dat2[i2];
if dat2[i2]<vmin2 then vmin2:=dat2[i2];
inc(i2);
until i2>2048;
end;
end;
if (t1>=4) and (t2<13) then begin
comp:=((vmax-vmin-vref)/Sfff)*max1{-0.005874};
comp2:=((vmax2-vmin2-vref2)/Sfff)*max2{versi SB16}
end else begin
comp:=((vmax-vmin{-vref2})/12800)*max1;{versi SB PRO 2}
comp2:=((vmax2-vmin2{-vref2})/12800)*max2;
end;
{menampilkan data tegangan}
outtextxy(20,30,'Vamp ch1 : '+hasil( comp) );
outtextxy(20,40,'Vamp ch2 : '+hasil( comp2) );
outtextxy(20,50,'Vmin : '+hasil( vmax-vmin) );
outtextxy(20,60,'Vmin : '+hasil( vmax2-vmin2{-vref2}));
moveto(pl,220);
setcolor(14);
if phase then begin {Beda phase}

```

```

ii:=20;
{mengambar data Ch1 terhadap Ch2}
moveto(data[ii]div vch1 +270-((Sfff-512) div (vch1*2))+pch1,
dat2[ii]div vch2 +220-((Sfff-512) div (vch2*2))+pch2);
for ii:=pl to 520 do
  lineto(data[ii]div vch1 +270-((Sfff-512) div (vch1*2))+pch1,
  dat2[ii]div vch2 +220-((Sfff-512) div (vch2*2))+pch2)
end else if not stereo then {mengambar single trace}
  for ii:=pl to 520 do
    lineto(ii,data[ii]div tpd]div vch1 +220-((Sfff-512) div (vch1*2))+pch1)
  else begin {mengambar double trace}
    setcolor(14);
    for ii:=pl to 520 do
      lineto(ii,data[ii]div tpd]div vch1 +220-((Sfff-512) div (vch1*2))+pch1);
      setcolor(lightgreen);
      moveto(pl,220);
      for ii:=pl to 520 do
        lineto(ii,dat2[ii]div tpd]div vch2 +220-((Sfff-512) div (vch2*2))+pch2);
      end;
    ii:=0;
  repeat {delay}
    mouseget; {mengambil informasi mouse}
    inc(ii);
  until (button<>0) or (ii>60000);
  {end of delay}
  if spic then begin {simpan ke BMP}
    save_bmp(nama);
    spic:=false;
  end;
  if swav then begin {simpan ke WAV}
    save_wav(nama);
    swav:=false;
  end;
  setcolor(8); {Hapus gambar}
  outtextxy(20,30,' Vamp ch1 : '+hasil( comp) );
  outtextxy(20,40,' Vamp ch2 : '+hasil( comp2) );
  outtextxy(20,50,' Vmin : '+hasil( vmax-vmin{-0.005874} ) );
  outtextxy(20,60,' Vmin : '+hasil( vmax2-vmin2{-vre12} ) );
  moveto(pl,220);
  if phase then begin
    ii:=20;
    moveto(data[ii]div vch1 +270-((Sfff-512) div (vch1*2))+pch1,
    dat2[ii]div vch2 +220-((Sfff-512) div (vch2*2))+pch2);
    repeat
      lineto(data[ii]div vch1 +270-((Sfff-512) div (vch1*2))+pch1,
      dat2[ii]div vch2 +220-((Sfff-512) div (vch2*2))+pch2);
      ii:=ii+2;
    until ii>520;
  end else if not stereo then
    for ii:=pl to 520 do
      lineto(ii,data[ii]div tpd]div vch1 +220-((Sfff-512) div (vch1*2))+pch1)
    else begin
      for ii:=pl to 520 do
        lineto(ii,data[ii]div tpd]div vch1 +220-((Sfff-512) div (vch1*2))+pch1);
        moveto(pl,220);
        for ii:=pl to 520 do
          lineto(ii,dat2[ii]div tpd]div vch2 +220-((Sfff-512) div (vch2*2))+pch2);
        end;
      until button<>0; {akhir menghapus layar}
      MouseOff; {Mouse Off}
      setViewport(0,0,520,420,false);
      if (x>=20) and (x<=60) and (y>=440) and (y<=460) then begin
        size:=ImageSize(20,440,60,460);
        getmem(p3,size); {}

```

```

getimage(20,440,60,460,p3');
putimage(20,440,p3',notput);
delay(100);
putimage(20,440,p3',normalput);
freemen(p3,size);
ext:=true;
end;
if cek(530,380,625,420,1)=1 then begin
  nama:=key;{BMP}
  spic:=true;
end;
if cek(530,430,625,470,2)=2 then begin
  nama:=key;{WAV}
  swav:=true;
end;
if cek(580,320,620,360,3)=3 then run:=false;
if cek(530,320,170,360,4)=4 then run:=true;
if cek(532,275,573,316,5)=5 then pl:=pl-2;
if cek(574,275,615,316,6)=6 then pl:=pl+2;
{mengatur posisi Ch 1}
if cek(530,179,571,220,7)=7 then pch1:=pch1-2;
if cek(530,221,571,262,8)=8 then pch1:=pch1+2;
{Mangatur Posisi Ch2}
if cek(580,179,621,220,9)=9 then pch2:=pch2-2;
if cek(580,221,621,262,10)=10 then pch2:=pch2+2;
if cek(100,440,220,460,11)=11 then
begin
  com($A0);{set Mono}
  stereo:=false;
  phase:=false;
end;
if cek(240,440,360,460,12)=12 then
begin
  com($A8);{set stereo}
  stereo:=true;
  phase:=false;
end;
if cek(380,440,500,460,13)=13 then
begin
  com($A8);{set stereo}
  stereo:=true;
  phase:=true;
end;
if cek(602,19,613,30,14)=14 then tpd:=tpd-5;{mengatur posisi}
if cek(602,30,613,41,15)=15 then tpd:=tpd+5;{mengatur posisi}
if cek(602,69,613,80,16)=16 then dec(ch1);{atur V/div}
if cek(602,80,613,91,17)=17 then inc(ch1);{atur V/Div}
if tpd<1 then tpd:=1;
if tpd>11 then tpd:=11;
case tpd of
  1:begin
    display(540,20,600,40,14,0,"");
    display(540,20,600,40,15,0,' 1ms');
  end;
  6:begin
    display(540,20,600,40,14,0,"");
    display(540,20,600,40,15,0,' .2ms');
  end;
  11:begin
    display(540,20,600,40,14,0,"");
    display(540,20,600,40,15,0,' .1ms');
  end;
end;
case ch1 of

```

```

t:=port[rdstat];
tanda:=true;
inline($fb);{kembalikan flag dari stack}
end;

function bacadsp:byte; {Membaca data DARI sound blaster}
begin
  repeat until port[rdstat] and $80 = $80;
  bacadsp:=port[sbread];
end;

procedure versi; {mengambil versi DSP;}
begin
  com($E1);
  t1:=bacadsp;
  t2:=bacadsp;
end;

procedure baca2;
begin
  com($40);com($E9);{Set time constant}
  com($42);com(hi(44100));com(lo(44100));{Set f sample}
  {com($A4);}
  Com($48);Com(lo(dma-1));Com(hi(dma-1));
  {Install DMAC}
  port[dma_data[channel,1]]:=505;{ pilih channel dmaC}
  port[dma_data[channel,2]]:=0;{reset channel DMA}
  port[dma_data[channel,3]]:=45;{45}
  port[dma_data[channel,4]]:=lo(i);{kirim alamat memori}
  port[dma_data[channel,4]]:=hi(i);
  port[dma_data[channel,5]]:=page;{Kirim halaman memori}
  port[dma_data[channel,6]]:=lo(dma);
  port[dma_data[channel,6]]:=hi(dma);
  port[dma_data[channel,1]]:=501;
  {Program SB ke Tranfer high speed (44100kHz)}
  if (t1>3) and (stereo or phase) then begin {DMA tranfer w/ SB 16}
    com($CA);Com($20);com(lo(dma-1));com(hi(dma-1));end
  else begin com($CA);Com($0);com(lo(dma-1));com(hi(dma-1));end;
  if t1<4 then com($99);{High Speed DMA tranfer w/ SB PRO}

  {end DMAC}
  {com($99);{Set High Speed DMAC chanel 1}
end;

```

### BMP.PAS

{ BAGIAN DARI PROGRAM OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD

```

Author           : Sondag P H
Compiler         : Turbo Pascal 7.0
CPU              : Intel 486 Dx2-66MHz
Operating System : DOS 6.22, WTN95/97
Abstrak          : Procedure meminta masukkan nama file}

```

function key:string;

```

var z           : string[8];
    c           : array[1..8]of char;
    b,k         : byte;
    p           : char;
begin
  {Masukkan nama File}
  p:=#2; z:="";
  display(200,100,440,150,14,0,'nama file :');
  setviewport(200,100,440,150,true);
  b:=1;
  repeat

```

```

p:=readkey;{ambil data dari keyboard}
if p=#0 then p:=readkey
if p=#8 then b:=b-2 else c[b]:=-p;
if b>9 then b:=0;
setcolor(0);
outtextxy(100,22,z);
z:="";{set nama file awal kosong}
for k:=1 to b do
  z:=z+c[k];{ubah char ke string}
setcolor(14);
outtextxy(100,22,z);
inc(b);
until (p=#13) or (b=#9);
setviewport(200,100,440,150,false);
setviewport(0,0,639,479,true);
display(200,100,440,150,8,8;nama file:);{Menampilkan nama file}
z:="";
for k:=1 to b-2 do
  z:=z+c[k];
key:=z;
end;

```

#### **BMP2.PAS**

**{ BAGIAN DARI PROGRAM OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD**

```

Author      : Sondag P H
Compiler    : Turbo Pascal 7.0
CPU         : Intel 486 Dlx2-66MHz
Operating System : DOS 6.22, WIN95/97
Abstrak     : Procedure menyimpan data dari ADC SB ke file BMP}
}
{ Save a screen of 640x480 pixels in 16 colors to a BMP image }
{ Size of this image is 153.718 bytes }
{ 640x480 = 307200/2 = 153.600+118 = 153.718 bytes }
const
  bmphead :array[1..118] of byte =

```

```

{ BM } (66,77,
  118,
  8,
  0,
  0,0,0,0,0,
  118,0,0,0,40,
  0,0,0,
{ br } 128,2,   { (2x256)+128 = 640 pixels }
  0,0,
{ ho } 224,1,   { (1x256)+224 = 480 pixels }
  0,0,1,0,4,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
{ 0} 0,0,0,0,
{ 1} 170,0,0,0,
{ 2} 0,170,0,0,
{ 3} 170,170,0,0,
{ 4} 0,0,170,0,   { dit zijn de 16 kleuren(colors) }
{ 5} 170,0,170,0, { die gelden voor de standaard }
{ 6} 0,85,170,0,  { pascal vga scherm }
{ 7} 170,170,170,0,
{ 8} 85,85,85,0,
{ 9} 255,0,0,0,
{10} 0,255,0,0,
{11} 255,255,0,0,
{12} 0,0,255,0,
{13} 255,0,255,0,
{14} 0,255,255,0,
{15} 255,255,255,0);

```



```
procedure save_bmp(filename:string);
```

```
var
  f      :file;
  x,y,p  :integer;
  r,rr,pal :byte;

begin
  assign(f,filename+'.BMP');
  rewrite(f,1);
  for p:=1 to 118 do blockwrite(f,bmphead[p],1);
  for y:=479 downto 0 do begin
    for x:=0 to 639 do begin
      r:=getpixel(x,y);
      rr:=getpixel(x+1,y);
      pal:=(r*16)+rr;
      blockwrite(f,pal,1);
      inc(x);
    end;
  end;
  close(f);
end;
```

#### WAVFMT.PAS

{ BAGIAN DARI PROGRAM OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD

```
Author      : Sondag P H
Compiler    : Turbo Pascal 7.0
CPU        : Intel 486 Dx2-66MHz
Operating System : DOS 6.22. WIN95/97
Abstrak     : Procedure menyimpan datan dari ADC SB ke file WAV}
}
```

```
Type id_t = array[1..4] of char;{header WAV}
```

```
riff_1 = Record
  R_Ident : id_t;
  length : Longint;
  C_Ident : id_t;
  S_Ident : id_t;
  s_length: Longint;
  Format ,
  Modus : Word;
  freq ,
  byte_p_s: Longint;
  byte_sam,
  bit_sam : Word;
  D_Ident : id_t;
  d_length: Longint;
End;
```

```
Var id      : riff_t;
    wav,wav2 : file;
    wavbuff  : array[1..8192]of byte;
```

```
{Main Program}
procedure header;
begin
  with id do begin{Header WAV}
    R_Ident:='RIFF';
    {length:= {Longint of File}
    C_Ident := 'WAVE';
    S_Ident := 'fmt ';
```

```

s_length:= $10; {Always}
Format :=1;
{Modus := 2; 1=mono,2=stereo}
freq := 44100; {Sample Rate}
{byte_p_s:= modus*freq*bit_sam;}
{byte_sam:=4; 1=8bit mono
                2=8bit stereo
                3=16bit mono
                4=16bit stereo}
bit_sam := 16; {sum bit / sample}
D_Ident := 'data';
{d_length:= length file-44 (header); }
end;
end;

```

```

Procedure Save_Wav(name:string);
var f:file;p:byte;
    data:array[1..8192] of integer;
begin
header:
assign(f,name+'.WAV');
rewrite(f,1); {Buka file}
if stereo then begin {Header WAV stereo}
    id.length:=44+196608-8{(6*dma*4)};
    id.d_length:=199608;
    id.byte_p_s:=2*44100*2;
    id.byte_sam:=4;
    id.modus:=2;
end else begin {header WAV mono}
    id.length:=44+98304{3*dma*2};
    id.d_length:=98304;
    id.byte_p_s:=2*44100;
    id.byte_sam:=3;
    id.modus:=1;
end;
blockwrite(f,id,44);
for ii:=1 to dma do {mengambil data di memori}
    data[ii]:=memw[seg(buffer^):ofs(buffer^)+ii]-$8000;
if stereo then
    for ii:=0 to 23 do
        blockwrite(f,data,dma){Tulis ke file}
else
    for ii:=0 to 11 do
        blockwrite(f,data,dma); {Tulis ke file}
close(f);
end;
end;

```

#### MOUSES.BPAS

{ BAGIAN DARIPROGRAM OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD

```

Author           : Sondag P H
Compiler         : Turbo Pascal 7.0
CPU              : Intel 486 Dx2-66MHz
Operating System : DOS 6.22, WIN95/97
}

```

```
var x,y,button:word;
```

```

Procedure Mousetest; {Memeriksa driver mouse}
begin
asm
    mov ax,0
    int 33h
    mov x,ax
end; {ax = 0 mouse tidak ditemukan}

```

```

if x=0 then begin write('Mouse Tidak ditemukan');
  halt(0); end else
  writeLn('Mouse terinstall');
end;

```

```

Procedure Mouserreset;Assembler;
asm
  mov ax,0 {AX=ffffh mouse ada bx: jml tombol}
  int 33h{reset mouse}
end;

```

```

Procedure MouseOn;Assembler;{Mouse On}
asm
  mov ax,1
  int 33h
end;

```

```

Procedure MouseOff;Assembler;{Mouse Off}
asm
  mov ax,2
  int 33h
end;

```

```

Procedure MouseGet;Assembler;{Mengambil informasi koordinat dan tombol mouse}
asm
  mov ax,3;
  int 33h
  mov x,cx
  mov y,dx
  and bx,7
  mov button,bx
end;

```

## SETUP.PAS

```

{ PROGRAM SETUP OSILOSKOP DIGITAL SEDERHANA DENGAN MENGGUNAKAN SOUND CARD
Author           : Sondag P H
Compiler         : Turbo Pascal 7.0
CPU              : Intel 486 Dx2-66MHz
Operating System : DOS 6.22, WIN95/97
Abstrak          : Program ini merupakan setup untuk kalibrasi program osiloskop sederhana OSC2.PAS}
Linker           : sb.pas
}
Program Setup;
Uses Timer,Crt,Dos;
Var stereo,phase : boolean;
    c             : char;
    y             : byte;
    rev           : pointer;
    d             : byte;
    ii            : word;
    data          : array[1..8192] of word;
    vmax.vmin    : word;
    t             : text;
{$i SB.PAS}
begin
  Clrscr;
  init;{inisialis Sound blater}
  sbreset;{reset sound blaster}
  versi;c:='';{mengambil versi DSP}
  {set nilai volume awal}
  Assign(t,'OSC.CFG');{buka file}
  reset(t);
  port[mix]:=3F;

```

```

readln(t,d);{baca nilai data pada file}
port[mixdata]:=d;
port[mix]:= $21;
readln(t,d);
port[mixdata]:=d;
close(t);
Clrscr;
Writeln('Setup Oscilloscope Digital');
y:=1;
getintvec(sb.irq+8,rev);{simpan alamat irq asli}
setintvec(sb.irq+8,@baca);{set alamat irq ke program}
port[$21]:=port[$21] and (255 xor (1 shl sb.irq));{Enable IRQ}
getmem(buffer,sizeof(data));{pesan besar memori}
i:=seg(buffer)shl 4 + ofs(buffer);{ambil alamat memori}
page:=(seg(buffer) + ofs(buffer) shr 4) shr 12;{ambil jumlah halaman}
if channel=3 then begin
  dma:=dma div 2;
  page:=page div 2;
end;
dma:=4096;{set ukuran blok}
stereo:=true;{mode stereo}
Repeat
  tanda:=false;
  gotoxy(1,2);write('[ ] Volume Line In');{Tampilan menu}
  gotoxy(1,3);write('[ ] Gain Line In Left: Right');
  gotoxy(1,4);write('[ ] Volume CD ');
  gotoxy(1,5);write('[ ] Input Filter ');
  gotoxy(2,y+1);write('v');
  port[mix]:= $2e;{Line in}
  d:=port[mixdata];
  gotoxy(26,2);write(d shr 4:2,'-',d and $f:2);
  port[mix]:= $3f;{Line in Gain Left}
  d:=port[mixdata];
  gotoxy(26,3);write(d shr 6:2);
  port[mix]:= $40;{Line in Gain Right}
  d:=port[mixdata];
  gotoxy(41,3);write(d shr 6:2);
  port[mix]:= $28;{CD Vol}
  d:=port[mixdata];
  gotoxy(26,4);write(d shr 4:2,'-',d and $f:2);
  gotoxy(1,8);
  writeln('Sound Blaster Pro Only');
  gotoxy(1,9);
  port[mix]:= $c;
  d:=port[mixdata];
  case d and $6 of
    $0:writeln('Microphone ');
    $2:writeln('CD ');
    $4:writeln(' ');
    $6:writeln('Line In ');
  end;
  if d and $8 = $8 then
    writeln('Lowpass Filter ') else
    writeln('Highpass Filter');
  if d and $20 = $20 then
    writeln('Filter : Off') else
    writeln('Filter : On');
  vmax:= $0;{set nilai awal max dan min}
  vmin:= $FFFF;
  baca2:={program SB & DMA}
  repeat until (tanda);{tunggu interupsi}
  for ii:=1 to dma-2 do
    begin{mengambil data dari memori}
      data[ii]:=memw[seg(buffer)+ofs(buffer)+ii];
    end;
end;

```

```

    if data[i] > vmax then vmax := data[i];
    if data[i] < vmin then vmin := data[i];
end;
{ gotoxy(100,20); write(' '); }
gotoxy(100,19); write('12800');
gotoxy(100,20); write(vmax-vmin:6); {menampilkan data ADC ke layar}
if keypressed then c:=readkey; {jika ada tombol ditekan baca tombol tsb}
if c=#0 then c:=readkey;
case c of
    #72:dec(y); {tombol panah atas dec c}
    #80:inc(y); {tombol panah atas dec c}
end;
if c=#75 then
    case y of
        1:begin
            port[mix]:= $2e; {Line in}
            d:=port[mixdata];
            d:=d-$22;
            if d shr 4 <> d and $0f then d:= $00;
            port[mixdata]:=d;
        end;
        2:begin
            port[mix]:= $3f; {Line in}
            d:=port[mixdata];
            d:=d-$40;
            port[mixdata]:=d;
            port[mix]:= $40; {Line in}
            port[mixdata]:=d;
        end;
        3:begin
            port[mix]:= $28; {Line in}
            d:=port[mixdata];
            d:=d-$22;
            if d shr 4 <> d and $0f then d:= $00;
            port[mixdata]:=d;
        end;
        4:begin
            port[mix]:= $c; {Line in}
            d:=port[mixdata];
            d:=(d-2) and 6;
            if d shr 4 <> d and $0f then d:= $00;
            port[mixdata]:=d;
        end;
    end;
end;
if c=#77 then
    case y of
        1:begin
            port[mix]:= $2e; {Line in}
            d:=port[mixdata];
            d:=d+$22;
            if d shr 4 <> d and $0f then d:= $ff;
            port[mixdata]:=d;
        end;
        2:begin
            port[mix]:= $3f; {Line in}
            d:=port[mixdata];
            d:=d+$40;
            port[mixdata]:=d;
            port[mix]:= $40; {Line in}
            port[mixdata]:=d;
        end;
        3:begin
            port[mix]:= $28; {Line in}
            d:=port[mixdata];

```

```

    d:=d+$22;
    if d shr 4 <> d and $0f then d:=$ff;
    port[mixdata]:=d;
end;
4:begin
    port[mix]:= $c; {Line in}
    d:=port[mixdata];
    d:=(d+2) and 6;
    if d shr 4 <> d and $0f then d:= $00;
    port[mixdata]:=d;
end;
end;
if y=0 then y:=1;
if y>4 then y:=4;
if c<=#27 then c:=';
until c=#27;
setintvec(sb.irq+8.rev);{mengembalikan alamat asli}
Assign(t,'OSC.CFG');
rewrite(t);{buka file}
port[mix]:= $3F;{mengatur Gain line in}
writeln(t,port[mixdata]);
port[mix]:= $2E;{mengatur Volume line in}
writeln(t,port[mixdata]);
port[mix]:= $2e;
port[mixdata]:= $00;
baca2;{Program SB & DMA}
repeat until (tanda);{tunggu sampai ada interupsi}
for ii:=1 to dma-2 do
begin
    data[ii]:=memw[seg(buffer^):ofs(buffer^)+ii];
    {mengambil data dari memori}
    if data[ii]>vmax then vmax:=data[ii];{ambil nilai max}
    if data[ii]<vmin then vmin:=data[ii];{ambil nilai min}
end;
writeln(t,vmax-vmin);{tuliskan data ke file}
close(t);{Tutup file}
end.

```

Tek

Stop

Stop

M Pos: 0.000s

MEASURE

Source

Type

Freq

999.0Hz

Mean

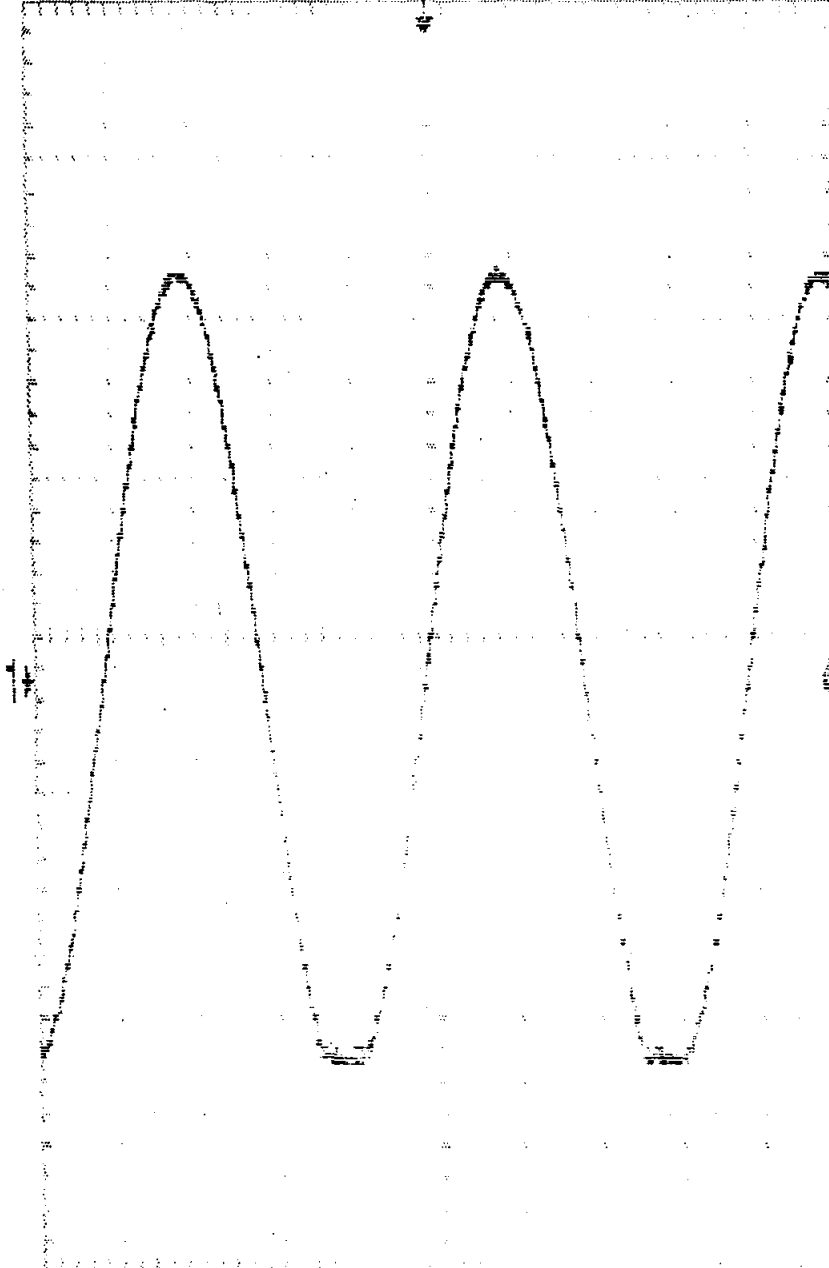
0.00V

Period

1.002ms

Period

1.002ms



CH1 20.0V

CH2 200mV

M 250µs

CH1 5 -42.2mV

PERPUSTAKAAN  
 Universitas Negeri Widya Mandala  
 SURABAYA