

```

'Set waveforms with two channels
var lastT, now;           'to record where we got to
var doneSetupFlag%:=0;
                        'Flag to set when setup performed
var doneMarkerFlag% := 0;
                        'Flag to set when target marker displayed
var doneSoundFlag% := 0;
                        'Flag when sound is output
var doneSoundFlag1% := 0;
                        'Flag when sound is output
var newChan%;
var sampleFlag% := 0;
var fTime;

var mChan%[1];

'Associated files      - Edit accordingly
var seqFile$ := "C:\\Spike6\\scripts\\seq2.pls";           'Output
sequencer file
var pi;
pi := ATan(1.0)*4;                                         '3.14159...

var time;
var time1 := 5;
var time2 := 6;
var time3 := 9;
var targetMarker%;
var startTargetFlag% := 0;
var target0%;

var vf%,vfl%,xy%,xy1%, x[100], y[100];
var ax[4], ay[4];           'start posn'
var bx[4], by[4];           '12'
var cx[4], cy[4];           '1'
var dx[4], dy[4];           '2"
var ex[4], ey[4];           '3'
var fx[4], fy[4];           '4'
var gx[4], gy[4];           '5'
var hx[4], hy[4];           '6'
var ix[4], iy[4];           '7'
var jx[4], jy[4];           '8'
var kx[4], ky[4];           '9'
var sx[4], sy[4];           '10'
var tx[4], ty[4];           '11'

ax[0]:=-3; ax[1]:=-3; ax[2]:=3; ax[3]:=3;

```

```

ay[0]:=-3; ay[1]:=3; ay[2]:=3; ay[3]:=-3;

bx[0]:=-3; bx[1]:=-3; bx[2]:=3; bx[3]:=3;
by[0]:=12; by[1]:=18; by[2]:=18; by[3]:=12;

cx[0]:=6.5; cx[1]:=6.5; cx[2]:=8.5; cx[3]:=8.5;
cy[0]:=11.99; cy[1]:=13.99; cy[2]:=13.99; cy[3]:=11.99;

dx[0]:=11.99; dx[1]:=11.99; dx[2]:=13.99; dx[3]:=13.99;
dy[0]:=6.5; dy[1]:=8.5; dy[2]:=8.5; dy[3]:=6.5;

ex[0]:=12; ex[1]:=12; ex[2]:=18; ex[3]:=18;
ey[0]:=-3; ey[1]:=3; ey[2]:=3; ey[3]:=-3;

fx[0]:=11.99; fx[1]:=11.99; fx[2]:=13.99; fx[3]:=13.99;
fy[0]:=-8.5; fy[1]:=-6.5; fy[2]:=-6.5; fy[3]:=-8.5;

gx[0]:=6.5; gx[1]:=6.5; gx[2]:=8.5; gx[3]:=8.5;
gy[0]:=-13.99; gy[1]:=-11.99; gy[2]:=-11.99; gy[3]:=-13.99;

hx[0]:=-3; hx[1]:=-3; hx[2]:=3; hx[3]:=3;
hy[0]:=-18; hy[1]:=-12; hy[2]:=-12; hy[3]:=-18;

ix[0]:=-8.5; ix[1]:=-8.5; ix[2]:=-6.5; ix[3]:=-6.5;
iy[0]:=-13.99; iy[1]:=-11.99; iy[2]:=-11.99; iy[3]:=-13.99;

jx[0]:=-13.99; jx[1]:=-13.99; jx[2]:=-11.99; jx[3]:=-11.99;
jy[0]:=-8.5; jy[1]:=-6.5; jy[2]:=-6.5; jy[3]:=-8.5;

kx[0]:=-18; kx[1]:=-18; kx[2]:=-12; kx[3]:=-12;
ky[0]:=-3; ky[1]:=3; ky[2]:=3; ky[3]:=-3;

sx[0]:=-13.99; sx[1]:=-13.99; sx[2]:=-11.99; sx[3]:=-11.99;
sy[0]:=6.5; sy[1]:=8.5; sy[2]:=8.5; sy[3]:=6.5;

tx[0]:=-8.5; tx[1]:=-8.5; tx[2]:=-6.5; tx[3]:=-6.5;
ty[0]:=11.99; ty[1]:=13.99; ty[2]:=13.99; ty[3]:=11.99;

DoToolbar();

Proc DoToolbar()
ToolbarSet(1,"Quit");
ToolbarSet(3,"Setup",DoSetup%);
Toolbar("",1023);
end;

```

```

Func DoSetup%(
lastT:=0;
SampleSettings();           'set the channel specifications
if vf% then
    View(vf%);
    FileClose(0,-1);
    vf%:=0;
endif;
vf%:=FileNew(0,1);
if vf%<=0 then message("No data file"); halt endif;
Window(0,0,50,100);        'position it
WindowVisible(0);          'hides data window
DoXY();
SampleStart();
ToolbarSet(0,"Idle",Idle%);
ToolbarClear(3);
ToolbarSet(3,"Sample Data",DoSample%);
return 1;
end;

```

```

Func DoSample%(
ToolbarClear(0);
lastT:=0;
SampleSettings();           'set the channel specifications
if vf% then
    View(vf%);
    SampleStop();
    FileClose(0,-1);
    vf%:=0;
endif;
SampleSequencer("seqFile$");
vf%:=FileNew(0,1);
if vf%<=0 then message("No data file"); halt endif;
Window(0,0,50,100);        'position it
WindowVisible(0);          'hides data window
mChan%[0] := MemChan(9,0,BinSize(1));
ChanTitle$(mChan%[0],"Derived");
ChanUnits$(mChan%[0],ChanUnits$(1));
ChanShow(mChan%[0]);
MemSave(101,3,-0,0);        'Save derived channel
'ChanDelete(mChan%[0]);
ChanShow(3);
DoXY();

```

```

SampleStart();
sampleFlag% := 1;
ToolbarSet(0,"Idle",Idle%);
ToolbarClear(3);
ToolbarSet(3,"Restart",DoSample%);
ToolbarSet(5,"Finish",DoSampleStop%);
startTargetFlag% += 1;
Seconds(0);
return 1;
end;

```

```

Proc DoXY()
if xy% then
    View(xy%);
    FileClose(0,-1);
    xy%:=0;
endif;
xy% :=FileNew(12,1);
if xy%<=0 then message("No xy view"); halt endif;
Window(50,0,100,100);      'position it
XRange(-30,30);             'set
scales
YRange(-1,-30,30);
WindowVisible(3);           'maximises xy window

DoTarget0();
DoHours();

```

```

XYSetChan(1, -100, 0, 0);   'chan 1, fixed 100 points, no sort
XYDrawMode(1, 2, 3);        'set size 1 dots
XYDrawMode(1,5,0);          'Stop XY view axes expanding if data
exceeds range
end;

```

```

Func Dosamplestop%()
View(xy%);FileClose(0,0);
View(vf%);
SampleStop();

```

```

ChanProcessAdd(4,2); 'Add Rectify to channel 3
'ChanProcessAdd(4,0); 'Add Smooth to channel 3
'ChanProcessAdd(4,1); 'Add DC Remove to channel 3
'ChanProcessArg(4,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(4,3,1,0.005); 'Set DC Remove Time constant (s)
ChanProcessAdd(5,2); 'Add Rectify to channel 4
'ChanProcessAdd(5,0); 'Add Smooth to channel 4
'ChanProcessAdd(5,1); 'Add DC Remove to channel 4
'ChanProcessArg(5,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(5,3,1,0.005); 'Set DC Remove Time constant (s)
'ChanProcessAdd(6,2); 'Add Rectify to channel 5
'ChanProcessAdd(6,0); 'Add Smooth to channel 5
'ChanProcessAdd(6,1); 'Add DC Remove to channel 5
'ChanProcessArg(6,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(6,3,1,0.005); 'Set DC Remove Time constant (s)
'ChanProcessAdd(7,2); 'Add Rectify to channel 6
'ChanProcessAdd(7,0); 'Add Smooth to channel 6
'ChanProcessAdd(7,1); 'Add DC Remove to channel 6
'ChanProcessArg(7,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(7,3,1,0.005); 'Set DC Remove Time constant (s)'ChanProcessAdd(7,2);
'Add Rectify to channel 7
'ChanProcessAdd(7,0); 'Add Smooth to channel 7
'ChanProcessAdd(7,1); 'Add DC Remove to channel 7
'ChanProcessArg(7,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(7,3,1,0.005); 'Set DC Remove Time constant (s)
'ChanProcessAdd(8,2); 'Add Rectify to channel 8
'ChanProcessAdd(8,0); 'Add Smooth to channel 8
'ChanProcessAdd(8,1); 'Add DC Remove to channel 8
'ChanProcessArg(8,2,1,0.005); 'Set Smooth Time constant (s)
'ChanProcessArg(8,3,1,0.005); 'Set DC Remove Time constant (s)
XRange(0,MaxTime());
FrontView(vf%);
FileClose(0,0);
return 0;
return;
end;

```

```

func Idle%()
var n1%,n2%;
time := Seconds();                                'Get current time
if startTargetFlag% then                          'If started recording (not calibration)
docase
    case time >= time1 and time < time2 then      'between t1
and t2
        if not doneMarkerFlag% then

```

```

        DoRandMarker();
        'Position target and trigger output
        DoneMarkerFlag% := 1;
    'Set flag
    endif;
    case time >= time2 and time < time3 then
        if not doneSoundFlag% then
            'Play sound
            SampleKey("N");
            doneSoundFlag% := 1;
            'Set flag
            endif;
            case time >= time3 then
                '3 seconds later
                if not doneSoundFlag1% then
                    'Play sound
                    SampleKey("S");
                    ClearTarget();
                    'Clear target
                    Seconds(0);
                    'Reset timer
                endif;
            endcase;
        endif;
        View(vf%);
        now := maxTime();
        'find where we are
        if (now - lastT > 1.0) then lastT := now-1.0 endif;
        n1% := ChanData(1, x[], lastT, now, fTime);
        n2% := ChanData(2, y[], lastT, now);
        n1% := Min(n1%, n2%);
        'get minimum points
        if n1% > 0 then
            view(xy%).XYAddData(1,x[:n1%], y[:n1%]);
            if sampleFlag% then
                ArrMul(x[:n1%],x[:n1%]);
                array
                ArrMul(y[:n1%],y[:n1%]);
                array
                ArrAdd(x[:n1%],y[:n1%]);
                Sqrt(x[:n1%]);
                square root of array
                MemSetItem(mChan%[0],0,fTime,x[:n1%]);
                MemSave(101,3,0,1);
                ChanShow(3);
            endif;
            'lastT := now;
            'update start point
        endif;
    '1 second later
    'Add arrays
    'Get
    'Square x
    'Square y

```

```

return 1;
end;

'func rectify%(wChan%, sTime, eTime)      ' returns the memory channel number
'var mc%;
'mc% := MemChan(1,0,BinSize(wChan%)); 'Create channel
'if mc%<=0 then return mc% endif;
'ChanScale(mc%, ChanScale(wChan%));  'Copy scale...
'ChanOffset(mc%, ChanOffset(wChan%)); '...and offset...
'ChanUnits$(mc%, ChanUnits$(wChan%)); '...and units
'ChanTitle$(mc%, "Rectified");
'ChanComment$(mc%,"Rectified from channel "+Str$(wChan%));
'var work%[8000]; ' buffer to process data with
'var read%;      ' the number of bins read

'repeat
'  read% := ChanData(wChan%, work%[], sTime, eTime, sTime);
'  if (read%>0) then ' if we got some data
'    Abs(work%[:read%]); ' rectify the data
'    MemSetItem(mc%, 0, sTime, work%[:read%]); 'copy it
'    sTime := sTime + BinSize(wChan%)*read%; 'next start time
'  endif;
'until read%<=0;
'ChanShow(-1);
'return mc%; 'Return the new channel number
'end;

Proc DoTarget0()
XYSetChan(0,-4, 0, 2);
XYDrawMode(2,2,0);
XYDrawMode(2,4,2);
'no marker points

XYAddData(2, ax[], ay[]);
return;
end;

Proc DoHours()      'add 12 hour markers
var angle;
for angle := 0 to 2*pi step pi/6 do
  XYSetChan(0,-12,0,0);
XYAddData(3, 15*cos(angle), 15*sin(angle));
  XYDrawMode(3, 2, 0)
next;
return;

```

end;

```
'set config for sampling; 2 channels at 100 Hz
Proc SampleSettings()
SampleClear();
SampleSequencer("C:\\Spike6\\scripts\\seq2.pls");
SampleUsPerTime(5);
SampleTimePerAdc(5);
SampleMode(1);
SampleLimitTime(-600);
SampleLimitSize(-1024);
SampleWaveform(1,0,100);
SampleCalibrate(1," degrees",33,0);
SampleTitle$(1,"X");
SampleComment$(1,"No comment");
SampleWaveform(2,1,100);
SampleCalibrate(2," degrees",33,0);
SampleTitle$(2,"Y");
SampleComment$(2,"No comment");
SampleWaveform(4,2,2000);
SampleTitle$(4,"EMG 1");
SampleWaveform(5,3,2000);
SampleTitle$(5,"EMG 2");
```

```
SampleEvent(6,0,0,1);
SampleTitle$(6,"Bit 0");
```

```
SampleEvent(7,1,0,1);
SampleTitle$(7,"Bit 2");
```

```
SampleEvent(8,2,0,1);
SampleTitle$(8,"Bit 4");
SampleEvent(9,3,0,1);
SampleTitle$(9,"Bit 8");
SampleEvent(10,4,0,1);
SampleTitle$(10,"Bit 16");
SampleEvent(11,5,0,1);
SampleTitle$(11,"Bit 32");
SampleEvent(12,6,0,1);
SampleTitle$(12,"Bit 64");
SampleEvent(13,7,0,1);
SampleTitle$(13,"Bit 128");
SampleTitle$(31,"Keyboard");
SampleComment$(31,"No comment");
SampleDigMark(2);
```

```

SampleTitle$(32,"Digital marker");
SampleComment$(32,"No comment");
end;

```

```

Proc DoRandMarker()
View(xy%);
if targetMarker% > 0 then                                'Clear target if
not done so
    ChanDelete(targetMarker%);
    targetMarker% := 0;
endif;
targetMarker% := XYSetChan(0, -4, 0,2);                  'Set new XY channel
XYDrawMode(targetMarker%,2,1);

XYDrawMode(targetMarker%,4,1);
XYJoin(targetMarker%, 2);
var marker;
marker := Rand();                                        'Get
random number
docase
    'Set target according to marker and trigger sequencer

    case marker <= 0.25 then
        XYAddData(targetMarker%,ex[], ey[]);
        View(vf%).SampleKey("3");
    case (marker <= 0.5) and (marker > 0.25) then
        XYAddData(targetMarker%,hx[], hy[]);
        View(vf%).SampleKey("6");

    case (marker > 0.5) and (marker <= 0.75) then

        XYAddData(targetMarker%,kx[], ky[]);

        View(vf%).SampleKey("9");

    case marker > 0.75 then

        XYAddData(targetMarker%,bx[], by[]);

        View(vf%).SampleKey("T");
endcase;
end;

```

```
Proc ClearTarget()
View(xy%);
if targetMarker% > 0 then
    ChanDelete(targetMarker%);
    targetMarker% := 0;
endif;
doneMarkerFlag% := 0;
doneSoundFlag% := 0;
doneSoundFlag1% := 0;
end;
```