

There are two free Pascal compilers available with which this program has been tested and works.

1. [www.bloodshed.net devpascal.html](http://www.bloodshed.net/devpascal.html)

an 8mb downloadable fully integrated development environment (IDE)
which is my first choice as its windows are more compatible with Widows

2. www.freepascal.org down i386 win32-ftp.freepascal.org.var

a 28mb downloadable fully integrated development environment (IDE)
which is my second choice as its windows are more compatible with DOS

The 28mb version has included with full documentation.

rtl.pdf	run time library document	(eg SIN...)
ref.pdf	language reference document	(eg IF...)

Some trigonometric functions need the MATH library, some use the default or SYSTEM library.
This is why the program says "uses math;"

An excellent online tutorial is found at

<http://www.taoyue.com/tutorials/pascal/contents.html>

and this provides better examples than are found in RTL.PDF and REF.PDF

Additionally, Lazarus is a free open source Pascal system which is also GUI and the illustrating shadows simulators for the IBM 1401 and IBM 360 were written using Lazarus.

Windows XP win32
Windows Vista win64

I have tested the Lazarus system and the programs I wrote for it as well
as the IBM 1401 and 360 simulators and their sundial programs
and also the Lazarus based ALLDIALS. These work on XP and Vista.
The Vista programs do not work on XP, XP programs work on Vista.

I have tested the 8mb Pascal IDE system and my sundial programs on
Both Vista and XP.

The program for horizontal dials follows.

=====

program HorizontalDial;

{ Educational purposes only: Author Simon WHeaton-Smith
Date March 5, 2007 1930 mst

This is a conversational program, it is a lot easier for new programmers to see what is going on rather than the transaction oriented programming techniques or the object oriented concepts.
}

{
There are two free Pascal compilers available with which this program has been tested and works.

1. [www.bloodshed.net devpascal.html](http://www.bloodshed.net/devpascal.html)
an 8mb downloadable fully integrated development environment (IDE) which is my first choice as its windows are more compatible with Widows
2. www.freepascal.org down i386 win32-ftp.freepascal.org.var
a 28mb downloadable fully integrated development environment (IDE) which is my second choice as its windows are more compatible with DOS

This has some useful files even though you would probably use the 8mb version of the Pascal IDE.

rtl.pdf Run time library Has the functions and the "uses" invocation for them
ref.pdf Reference Has the language structures such as IF... and so on.

The 28mb version has included with full documentation.

Some trigonometric functions need the MATH library, some use the default or SYSTEM library. This is why the program says "uses math;"

An excellent online tutorial is found at

<http://www.taoyue.com/tutorials/pascal/contents.html>
This has better examples of some of the useful things not well explained in RTL.PDF or REF.PDF

}

{
This program runs using the DOS feature of Windows, rather than the more modern GUI or graphical user interface. This is to make the progrsm clear and simple.

To copy the program's output, click on the top left of the DOS window and select EDIT then SELECT ALL, and repeat but select COPY. This captures the

program output and allows it to be saved to any file.

The output may be in a notation that requires you to interpret the data as follows...

```
*----- Horizontal Dial -----*
Enter latitude   32.75
Enter longitude  108.2
Enter legal meridian 105

Corrected hour line angles.
morning hours first
Hour: -2 hour angle: 3.3199E+01 Angle: 1.9494E+01
Hour: -1 hour angle: 1.8199E+01 Angle: 1.0085E+01
noon
Hour: 0 hour angle: 3.19999E+00 Angle: 1.7323E+00
noon
Hour: 1 hour angle: -1.18000E+01 Angle: -6.4479E+00
Hour: 2 hour angle: -2.68000E+01 Angle: -1.5283E+01
afternoon hours last
```

where

```
1.0085E+01 would be 10.085
1.7323E+00 would be 1.7323
```

The uncorrected hour lines are in the above notation, the corrected hour lines are truncated for easy reading. This is for educational purposes.

}

uses math;

```
var lng : single ;
    ref : single ;
    corh: single ;
    lat : double ;
    hlat: float ; { hour line angle as a tan }
    hla : float ; { hour line angle itself }
var sinlat : float ;

var xxx : string ;
var i : integer ;
    ii : integer ;
    j : float ;

begin
  writeln (*----- Horizontal Dial -----*);
  writeln ('www.illustratingshadows.com h-dial-pgm.pas');
  writeln (*----- March 5, 2007 1930 -----*);
  writeln ('Enter latitude   ');
  readln (lat);
  writeln ('Enter longitude  ');
  readln (lng);
  writeln ('Enter legal meridian');
  readln (ref);
  writeln ('Design latitude:   ', lat:6:2);
```

```
writeln ('Design longitude: ', lng:6:2);
writeln ('Reference meridian: ', ref:6:2);
writeln ('Correction to shadow ', 4*(lng-ref):6:2, ' minutes. ');
corh := (4*(lng-ref)) / 60 ;
writeln ('Correction to shadow ', corh:6:1, ' hours. ');
writeln ( ' ');

writeln ( 'Uncorrected hour line angles (hit enter)' );
readln (xxx);
writeln ( 'morning hours first' );
for i := -6 to +6 do

begin
  if i=0 then writeln ('noon');
  if i=1 then writeln ('noon');
  { get sin(lat) documented on page 1291 of rtl.pdf }
  sinlat := sin( degtorad(lat) );

  { get the hour angle of the sun }
  ii := - 1*i ;
  j := 15 * ii ;

  { get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15) }
  { tan is defined on page 699 of rtl.pdf it is a MATH function }
  hlat:= sinlat * tan(degtorad(j)) ;

  { get the hour line angle back to degrees }
  { arctan2 is defined on page 668 of rtl.pdf it is a
    MATH function and needs a "uses" }
  { arctan is defined on page 1159 of rtl.pdf it is a
    SYSTEM function and needs no "uses" }
  hla := radtodeg( arctan(hlat) ) ;

  writeln ('Hour: ', i, ' hour angle: ', j , ' Angle: ', hla );
end ;
writeln ( 'afternoon hours last' );
writeln ( ' ');

writeln ( 'Corrected hour line angles (hit enter)' );
readln (xxx);

writeln ( 'morning hours first' );
for i := -6 to +6 do
begin
  if i=0 then writeln ('noon');
  if i=1 then writeln ('noon');
  { get sin(lat) }
  sinlat := sin( degtorad(lat) ) ;

  { get the hour angle of the sun }
  ii := - 1*i ;
  j := 15 * (ii+corh) ;

  { get the resulting hour line angle ~ atan(sin(lat)*tan(hr*15) }
  hlat:= sinlat * tan(degtorad(j)) ;
```

```
{ get the hour line angle back to degrees }  
hla := radtodeg( arctan(hlat) );  
  
  writeln ('Hour: ', i, ' hour angle: ', j:6:2 , ' Angle: ', hla:6:2 );  
end ;  
writeln ( 'afternoon hours last' );  
writeln ( ' ');  
writeln ( '*** END ***' );  
  
  readln (xxx);  
end.
```