

LIST PROCESSING LANGUAGE (progeCAD and AutoCAD)

LISP is in use in both ProgeCAD and AUTOCAD. ProgeCAD has a free 2D version which is appropriate for dial plate design. The following notes may not be fully current and they refer to the free version that is available for non commercial use at:-

<http://www.progesoft.com/en/products/progecad-smart/>

They are included here because of the references to LISP, as opposed to proegCAD itself.

The process was tested in January of 2013 to ensure that most of this information was valid.

First, download "ProgeCAD" from ProgeSOFT, then register it. This is some 90mb, and the registration process which used to take some time is now instant.

Second, go to the: illustratingShadowsCD folder and then to LISP

and then locate the ".dcl" and files in the Illustrating Shadows CD folder called "LISP"

Name	Date modified	Size
hDial.lsp	1/26/2008 8:10 AM	11 KB
hDialForm.dcl	1/4/2008 1:30 PM	2 KB
LISP read me first.pdf	1/26/2008 8:15 AM	114 KB
vdec.lsp	1/26/2008 8:12 AM	16 KB
vdecForm.dcl	1/6/2008 2:54 PM	2 KB
vDial.lsp	1/26/2008 8:13 AM	10 KB
vDialForm.dcl	1/5/2008 6:37 PM	2 KB

and copy them into:-

c:\Program Files (x86)\progeSOFT\progeCAD 2009 Smart! ENG

This is required because the dialog panels (*.dcl files) are only loaded from this folder

hdialForm.dcl vdialForm.dcl vdecForm.dcl

In fact, you can copy the entire LISP folder from Illustrating Shadows into the progeCAD folder if you so desire.

Third, then, run "ProgeCAD"

- in TOOLS "load LISP or SDS Application"
- use the ADD FILE and locate the "hdial.lsp" program or "vDial.lsp", or "vdec.lsp"
- then LOAD only one of them, it doesn't remember them if you load several at a time.

Then back in "ProgeCAD"

look at the bottom of the screen which has two lines,
the top line (which you can expand) is what the system has done
the bottom line is what you enter.

the system, in response to your "LOAD" it will show you the name
of the application it loaded, in this sample it will be "HDIAL"
on the top line

on the bottom line enter "HDIAL", this is not case sensitive, the "hdial" is entered
because it is named in the hdial.lsp program file. Changing the file name
does not change the invoking name, unless you change that also.

The program then runs.

ditto with VDIAL	which is NOT a decliner, it is a pure south dial.
ditto with VDEC	dial which is a decliner

PROGRAMMING NOTES:

[frustrating messages when debugging LISP]

a malformed list:	mismatched () may cause this but the message may also be invalid floating point value happens on a LOAD of a program, but can be () problems, as in... (setq vhr ctr) vs (setq vhr ctr))
-------------------	---

too many arguments	often in (IF... if a (PROGN group is missing
--------------------	--

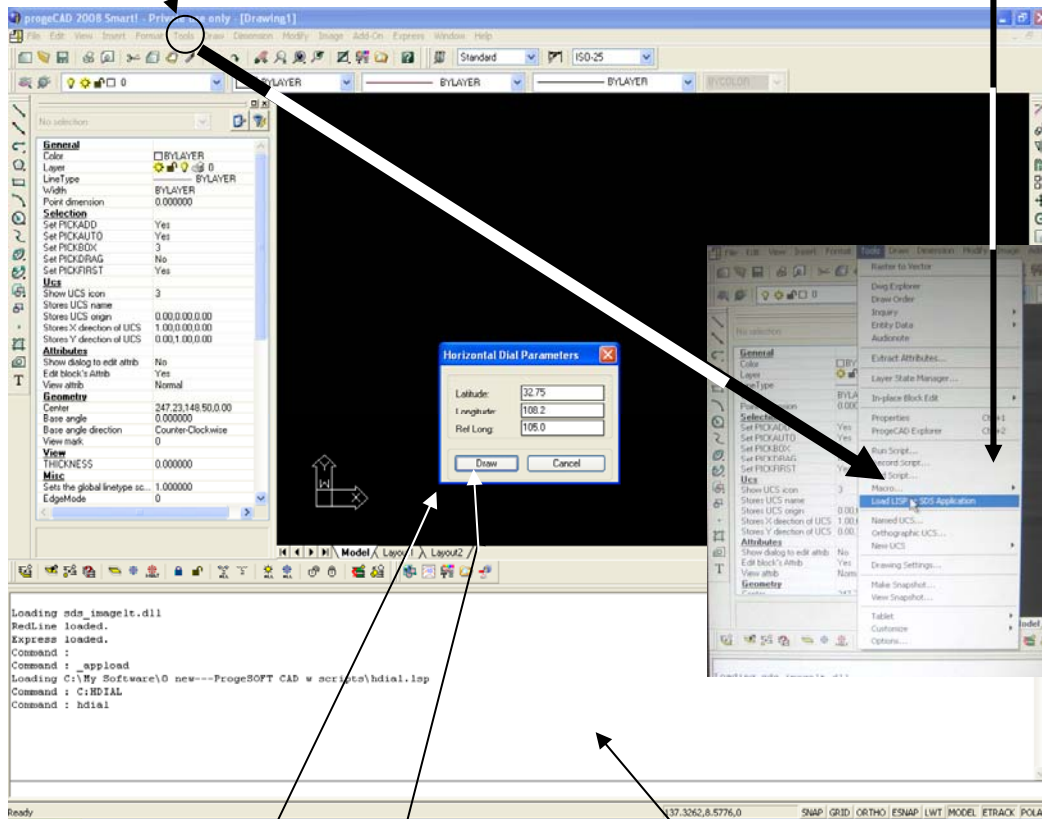
error messages	are not intuitive
----------------	-------------------

TOOLS

load LISP or SDS Application

is about half way down the menu

then you get a blank panel, and you must ADD the hdial program (each time you start ProgeCAD) and the first time you must locate it with BROWSE, then ADD it.

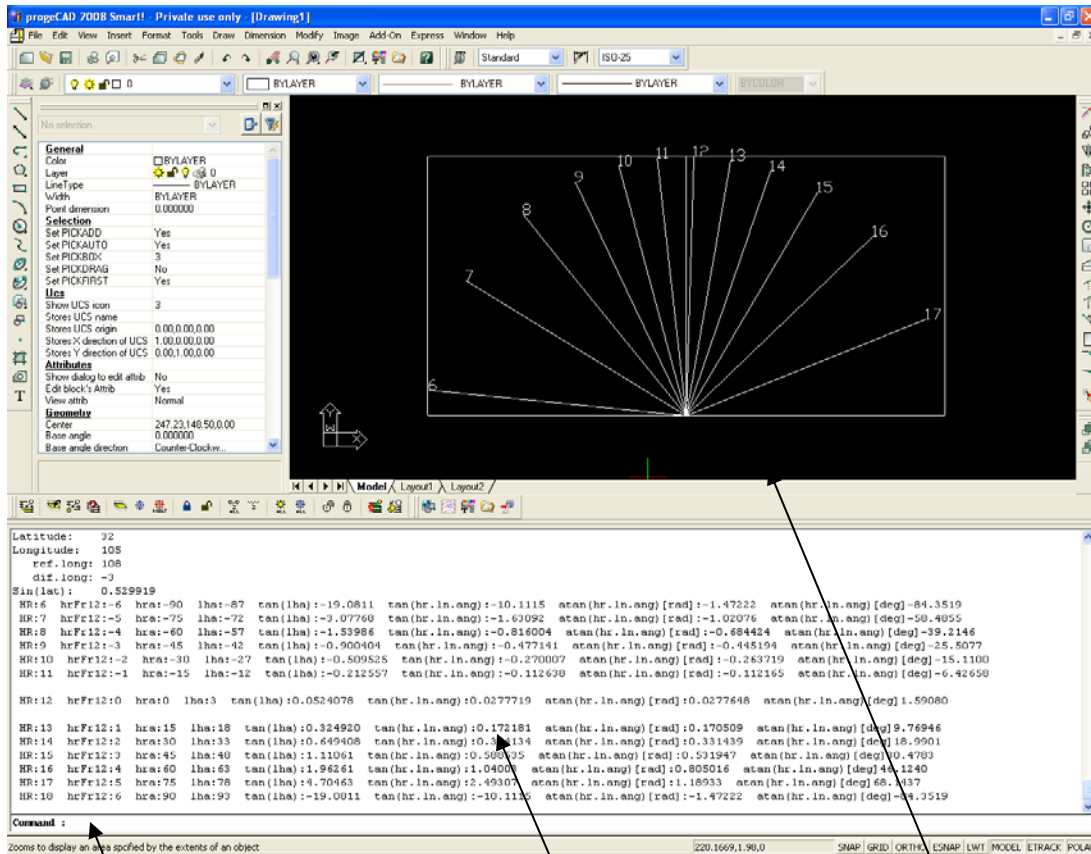


The first data entry screen

If you change the latitude etc, then the DRAW button is no longer the default, so ENTER doesn't do its thing, you will need to actually click on DRAW.

Results of commands and also the hour line angle data is displayed here.

This is the command line where, after a LOAD from TOOLS, you will type in HDIAL, VDIAL, or VDEC,



The display of data, and the graphical depiction.

This is the command line where, after a LOAD from TOOLS, you will type in HDIAL, VDIAL, or VDEC

The Data Entry panel

(for the h-dial)

This is saved in c:\Program Files\progeSOFT

or wherever the program was installed into.

```
hDialForm : dialog {
  label = "Horizontal Dial Parameters";
  : column {
    : boxed_column {
      : edit_box {
        key = "dLat";
        label = "Latitude:";
        edit_width = 15;
        value = "32.75";
      }
      : edit_box {
        key = "dLng";
        label = "Longitude:";
        edit_width = 15;
        value = "108.2";
      }
      : edit_box {
        key = "dRef";
        label = "Ref Long:";
        edit_width = 15;
        value = "105.0";
      }
    }
    : boxed_row {
      : button {
        key = "draw";
        label = " Draw ";
        is_default = true;
      }
      : button {
        key = "cancel";
        label = " Cancel ";
        is_default = false;
        is_cancel = true;
      }
    }
  }
}
```

The Main Program

(for the h-dial)

This is saved anywhere, but in ProgeSOFT, do a:-

TOOLS

load LISP or SDS Application

then browse, locate the folder with this in it, highlight this file, and click LOAD

```
; =====
; FUNCTION: to move data from the panel to the program
; =====

(defun gDialParms()                ; Invoked after
  (setq lat(atof(get_tile "dLat"))) ; the DRAW
  (setq lng(atof(get_tile "dLng"))) ; button is used
  (setq ref(atof(get_tile "dRef")))
)

; =====
; FUNCTION: to get user parameters and display a dial plate
; =====

; This is a horizontal dial program written in AutoCAD LISP
; This is intended for ProgeCAD from ProgeSOFT CAD system
; This is a skeleton only, but the data for hour line angles validates
;
; KNOWN BUG:  hours between 7 and 5pm are shown, this is because
;             tan fails when angles approach 90 degrees in this version
;             of LISP. And there is a d.long test to limit it.
;
; HISTORY:    Jan 4, 2008 I only started Jan 3 on this and finished
;             this skeleton Jan 4, in one day, never having used
;             LISP before. LISP uses polish notation.
;             Jan 5, 2008 added text for the hours
; AUTHOR:     Simon Wheaton-Smith FRI, MBCS, CITP   Jan 7, 2008 0842

(defun C:hDial()

  ; dont echo graphical commands to the log
  (setvar "cmdecho" 0)

  ; ----- Load the h-dial form
  (setq dcl_id (load_dialog "hDialForm.dcl"))

  ; ----- If the form wasnt there them do bad things
  (if (not (new_dialog "hDialForm" dcl_id))
    (progn
      (alert "The hDialForm.DCL file could not be loaded!")
      (exit)
    )
  )

  ; ----- do this set of commands when something happens
  (action_tile "draw" "(setq ddiag 2) (gDialParms) (done_dialog)")
  (action_tile "cancel" "(setq ddiag 1) (done_dialog)")

  ; ----- Display the panel
  (start_dialog)

  ; ----- Unload the panel
  (unload_dialog dcl_id)

  ; ----- If Cancel
  (if(= ddiag 1)
    (princ "\n ... no dial will be generated") ; LISP uses Polish
    ; notation, get over it.
  )

  ; ----- If draw
  ;[lat]    is a design latitude
  ;[lng]    is a design longitude
  ;[ref]    is a reference longitude
  ;[dlong]  is longitude correction
  ;[slat]   is the SINE of the latitude
  ;[thra]   is the tan of the hour angle
```

[illegible]

```

; (princ hr)
(setq hra (* hr 15))
; (princ " hra:")
; (princ hra)
(setq lha (- hra dlong))
; (princ " lha:")
; (princ lha)

; =====
; we have the sun's local hour angle in degrees
; now we need the hour line angle itself
; =====

; [tlha] is the tangent of the local hour angle
; get TANGENT(HOUR ANGLE)
; Convert degrees to radians
; Rad = (n * 2 * 3.1416) / 360
(setq tlha (* lha 2)) ; get lha times 2
(setq tlha (* tlha pi)) ; then times 3.1416
(setq tlha (/ tlha 360)) ; divided by 360

; - - - - -
; LISP TAN returns bad data on tan(radians(90))
; so we set flag=1 meaning ok to do a tan, flag=0 not ok
(setq flag 1) ; default is ok
( if (= lha -90)
  (setq flag 0)
)
( if (= lha 90)
  (setq flag 0)
)
; before doing a tan, check and see if tan will work
(if (= flag 1)
  (setq tlha (tan tlha))
)
; - - - - -

(if (= flag 1)
  (progn
    ; ~ ~ ~ ~ ~
    ; print the real or false tan (flase is one we made up)
    ; (princ " tan(lha):") (princ tlha)

    ; the guts of this part of the program is here
    ; [thrlan] is the tangent of the hour line angle
    ;
    ; F O R M U L A F O R H O U R L I N E A N G L E
    ;
    ; hrLnAng = atan ( sin(lat) * tan (lha) )
    ; slat tlha
    ;
    ; F O R M U L A F O R H O U R L I N E A N G L E
    ;
    (setq thrlan (* slat tlha))
    ; (princ " tan(hr.ln.ang):") (princ thrlan)

    ; [hrlan] is the hour line angle in radians
    (setq hrlan (atan thrlan))
    ; (princ " atan(hr.ln.ang)[rad]:") (princ hrlan)
    ; >>>> [hrlan] is the hour line angle, radians for polar coordinates

    ; [hrlanD] is the the hour line angle in degrees
    (setq hrlanD (/ hrlan 2))
    (setq hrlanD (/ hrlanD pi))
    (setq hrlanD (* hrlanD 360))
    (princ " hour line angle[deg]") (princ hrlanD)

    ; ABOVE FIGURES have been validated for 32.57 lat, 108.2 lng, ref 105

    ; now we need to convert the angle in radians >>> hrlan
    ; to coordinates for line drawing, with dial center at >>> x=250 y=50
    (setq Pt0 '(250 50))
    ; i.e. cx cy
    (setq rotation (- hrlan 1.571))
    ; 1.570796 is 90 degrees in radians, and LISP here needs a 90 shift
    (setq pt1 (polar Pt0(- rotation) (* 200)))

    ; dont show hour lines off the plate area
    ( if(> lha -90)
      (progn

```