```
CLEAR
CLOSE

ON ERROR GOTO transfererror
OPEN "transfer.dat" FOR INPUT AS #1
LINE INPUT #1, temp$

CLOSE 1
ON ERROR GOTO 0
'SHELL "erase transfer.dat"
'SHELL "erase temp19.dat"

desig$ = RTRIM$(LTRIM$(UCASE$(temp$)))
IF desig$ = "" THEN GOTO terror2
GOTO gotadesig2


transfererror:
RESUME terror2
terror2:
CLS
LOCATE 12, 1: PRINT "                This module requires X-CNTRY.EXE to run."
LOCATE 14, 1: PRINT "          Call Lamar Instrument (805) 933 3202 for further
information."

END


'TheVeryBeginning:
'APXXXXXX.BAS
'07-24-1992
'12-14-1992
'04-29-1993
'05-08-1993
'06-23-1994
'CLS
'LOCATE 10, 1:
'PRINT "         APPROACH  Copyright 1992 1993 1994 Paul Lamar (805) 933 3202"
'PRINT
'PRINT
'PRINT

'INPUT "         Type designator for airport. Example: LAX "; desig$

'gotadesig:

'desig$ = UCASE$(LTRIM$(MID$(desig$, 1, 4)))

gotadesig2:

'OPEN "ddd.bas" FOR OUTPUT AS #25
OPEN "TEMP45.DAT" FOR RANDOM AS #45

'gtemp$ = "$GPRMC,221842,A,3300.98,N,11859.17,W,900.0,045.0,261092,014.2,E*6A"

CLS
LOCATE 12, 1
PRINT "               "; desig$; " selected."
PRINT
```

```
PRINT
PRINT


'NS******************* NO SIMULATION *********************

'nosimulation:
CLS
LOCATE 12, 1
PRINT "                Please wait. Searching for data."

DIM airportlist(1 TO 24) AS STRING * 100
DIM plinedata(1 TO 800)
DIM pcircledata(1 TO 100)
pli = 1
pci = 1

winsize = 420
corner0 = 10
corner1 = 30                'corner1 = 30    scr 12
corner2 = corner0 + 400     'corner2 = 430   scr 12
corner3 = corner1 + 400     'corner3 = 450   scr 12

'com1:4800,N,8,1,OP5,RS,RB2048,CD0,DS0,CS0

ON ERROR GOTO errorfile1
OPEN "comunkat.dat" FOR INPUT AS #1
LINE INPUT #1, ctemp$
ON ERROR GOTO 0
CLOSE 1
GOTO comunkat1

errorfile1:
CLS
LOCATE 10, 1
PRINT "         COMUNKAT.DAT file is missing from this directory."
PRINT "         Please find the file and copy it to this directory."
PRINT "         It should consist of one line and look something like this:"
PRINT "         COM1:4800,N,8,1,OP5,RS,RB4096,CD0,DS0,CS0"
PRINT "                                                                 "
PRINT "         This file is specific to the type of GPS you have."
PRINT
PRINT "       Call Lamar Instruments for further information (805) 933 3202"
END

comunkat1:

OPEN ctemp$ FOR RANDOM AS #9                    'open the com program

temp$ = DATE$
mon$ = MID$(temp$, 1, 2)
day$ = MID$(temp$, 4, 2)
yr$ = MID$(temp$, 9, 2)
temp2$ = TIME$
hr$ = MID$(temp2$, 1, 2)

filenamegen$ = mon$ + day$ + yr$ + hr$
filenamesim$ = filenamegen$ + ".PB"
```

```
OPEN filenamesim$ FOR OUTPUT AS #12 LEN = 8000


'12345678901234567890
'"A","TOA ",
'"A", Ident$, Name$, City$, State$, Lat$, GPSLon$, elevat$, var$, PatAlt$, Lights$,
fueltype$, tower$, ctaf$
'"R", RLength$, Wide$, BaseEndIdent$, Alignment$, Lat$, GPSLon$, RHTraf$, Lights$,
ILSSys$
'"F","TOA ","118.9  ","125.6  ","111.9  "


desiglen = LEN(desig$)

'TBS********************** THE BIG SEARCH ************

'DEFDBL A-Z               'timer will qui

trial$ = desig$           '"TOA "
'CLS

SHELL "dir runway.dat > temp22.dat"
OPEN "temp22.dat" FOR INPUT AS #22

DO WHILE NOT EOF(22)
LINE INPUT #22, tmp20$
IF MID$(UCASE$(tmp20$), 1, 3) = "RUN" THEN GOTO gotfile20
LOOP
CLOSE 22
CLS
LOCATE 15, 1
PRINT "                         RUNWAY.DAT file missing."
PRINT
PRINT "                  Call Lamar Instruments (805) 933 3202 "
END

gotfile20:

CLOSE 22
KILL "temp22.dat"

tmp20a$ = LTRIM$(MID$(tmp20$, 15, 30))
final20$ = ""
FOR gf20i = 1 TO 30
tmpchr20$ = MID$(tmp20a$, gf20i, 1)
IF tmpchr20$ = " " THEN GOTO gotblank20
final20$ = final20$ + tmpchr20$
NEXT gf20i
gotblank20:

filelen = VAL(final20$)           ''' 4946280
recordlen = 188

numrecords = INT(filelen / recordlen)
'print #25, numrecords
low = 0
high = numrecords + 1
mid = INT((low + high) / 2)      'low starts at 0   mid starts at count/2
mid2 = mid * recordlen           'so it starts in the middle
```

```
OPEN "runway.dat" FOR INPUT AS #35 LEN = 2000
time1 = TIMER
ON ERROR GOTO tbserror1

start:
IF mid2 = oldmid2 THEN GOTO nomatchfound
oldmid2 = mid2

SEEK #35, mid2
LINE INPUT #35, temp$

test$ = LTRIM$(MID$(temp$, 7, 4))

'         1         2
'12345678901234567890123456789012345678901234567890
'1     00A  STONE MOUNTAIN-BRITT MEMORIAL          STONE MOUNTAIN           GA
'3     00CA GOLDSTONE /GTS/                        BARSTOW                  CA

'print #25, "       test$ "; test$
'print #25, "      trial$ "; trial$
'print #25, "        mid2 "; mid2

IF LTRIM$(RTRIM$(test$)) = LTRIM$(RTRIM$(trial$)) THEN GOTO tbsprintit

IF test$ > trial$ THEN GOTO lowerit
'         ISGT
IF test$ < trial$ THEN GOTO raiseit
'          ISLT
CLS
PRINT "something wrong"

END

tbsprintit:
'print #25, "tbsprintit"; TIMER
'print #25, temp$
OPEN "temp40.dat" FOR OUTPUT AS #40

index = mid2 - (10 * recordlen)            'go back to insure you get all
IF mid2 < (10 * recordlen) THEN index = 1  'neccessary to get the first ten

SEEK #35, index

'12345676890
'"A","XXXX"
ON ERROR GOTO 0
ON ERROR GOTO tbsprterror
FOR x = 1 TO 50
LINE INPUT #35, temp$
desig$ = RTRIM$(LTRIM$(UCASE$(desig$)))
test3$ = RTRIM$(LTRIM$(MID$(UCASE$(temp$), 6, desiglen)))
IF test3$ = desig$ THEN GOSUB print40
'print #25, "test3$ "; test3$
'print #25, "desig$ "; desig$
'print #25,
NEXT x

tbse2:
ON ERROR GOTO 0
```

```
time2 = TIMER
'print #25, " time 1 "; time1
'print #25, " time 2 "; time2
'print #25, "delta time "; time2 - time1
CLOSE #35
CLOSE #40
GOTO endgotdesig:

tbsprterror:
RESUME tbse2:

print40:
PRINT #40, temp$

RETURN


raiseit:
low = mid
mid = INT((low + high) / 2)         'mid will be higher
mid2 = mid * recordlen              'INT after the divide is crucial
GOTO start

lowerit:
high = mid                          'mid will get lower
mid = INT((low + high) / 2)
mid2 = mid * recordlen
GOTO start

tbserror1:
PRINT : PRINT : PRINT "error "
PRINT "mid "; mid
PRINT "mid2 "; mid2
PRINT MID$(temp$, 1, 60)
INPUT z$
END

nomatchfound:
CLS
LOCATE 15, 1
PRINT "                     NOT A VALID AIRPORT DESIGNATOR"
PRINT
PRINT
ON ERROR GOTO 0
CLOSE
CLEAR
'GOTO TheVeryBeginning
END



'EGD**************** END GOT DESIG *************************
endgotdesig:
'print #25, "endgotdesig"; TIMER

'"R", RLength$, Wide$, BaseEndIdent$, Alignment$, Lat$, Lon$, RHTraf$, Lights$,
ILSSys$
'"F","TOA ","118.9  ","125.6  ","111.9  "
```

```
OPEN "temp40.dat" FOR INPUT AS #40
'SEEK #40, 1
DO WHILE NOT EOF(40)
INPUT #40, nft$
'print #25, nft$

IF MID$(nft$, 1, 1) = "L" THEN GOSUB storealine
IF MID$(nft$, 1, 1) = "A" THEN GOSUB gotanairport
IF MID$(nft$, 1, 1) = "R" THEN GOSUB GotARunway
IF MID$(nft$, 1, 1) = "C" THEN GOSUB StoreACircle
IF MID$(nft$, 1, 1) = "F" THEN GOSUB gotafreq

LOOP

CLOSE #40
'123456789
'xx:xx.xxx
'print #25, "RUNWAY 1"; TIMER
' RUNWAY 1
runwaylatdeg1 = VAL(MID$(runwaylat1$, 1, 2))
runwaylatmin1 = VAL(MID$(runwaylat1$, 4, 6))
RunWayLatAbsCoord1 = (runwaylatdeg1 * 60) + runwaylatmin1

runwaylondeg1 = VAL(MID$(runwaylon1$, 1, 3))
runwaylonmin1 = VAL(MID$(runwaylon1$, 5, 6))
RunWayLonAbsCoord1 = (runwaylondeg1 * 60) + runwaylonmin1

'print #25, "RUNWAY 2"; TIMER
' RUNWAY 2
runwaylatdeg2 = VAL(MID$(runwaylat2$, 1, 2))
runwaylatmin2 = VAL(MID$(runwaylat2$, 4, 6))
RunWayLatAbsCoord2 = (runwaylatdeg2 * 60) + runwaylatmin2

runwaylondeg2 = VAL(MID$(runwaylon2$, 1, 3))
runwaylonmin2 = VAL(MID$(runwaylon2$, 5, 6))
RunWayLonAbsCoord2 = (runwaylondeg2 * 60) + runwaylonmin2

'print #25, "RUNWAY 3"; TIMER
' RUNWAY 3
runwaylatdeg3 = VAL(MID$(runwaylat3$, 1, 2))
runwaylatmin3 = VAL(MID$(runwaylat3$, 4, 6))
RunWayLatAbsCoord3 = (runwaylatdeg3 * 60) + runwaylatmin3

runwaylondeg3 = VAL(MID$(runwaylon3$, 1, 3))
runwaylonmin3 = VAL(MID$(runwaylon3$, 5, 6))
RunWayLonAbsCoord3 = (runwaylondeg3 * 60) + runwaylonmin3

'print #25, "RUNWAY 4"; TIMER
'RUNWAY 4
runwaylatdeg4 = VAL(MID$(runwaylat4$, 1, 2))
runwaylatmin4 = VAL(MID$(runwaylat4$, 4, 6))
RunWayLatAbsCoord4 = (runwaylatdeg4 * 60) + runwaylatmin4

runwaylondeg4 = VAL(MID$(runwaylon4$, 1, 3))
runwaylonmin4 = VAL(MID$(runwaylon4$, 5, 6))
RunWayLonAbsCoord4 = (runwaylondeg4 * 60) + runwaylonmin4

'print #25, "RUNWAY 5"; TIMER
' RUNWAY 5
```

```
runwaylatdeg5 = VAL(MID$(runwaylat5$, 1, 2))
runwaylatmin5 = VAL(MID$(runwaylat5$, 4, 6))
RunWayLatAbsCoord5 = (runwaylatdeg5 * 60) + runwaylatmin5

runwaylondeg5 = VAL(MID$(runwaylon5$, 1, 3))
runwaylonmin5 = VAL(MID$(runwaylon5$, 5, 6))
RunWayLonAbsCoord5 = (runwaylondeg5 * 60) + runwaylonmin5

'print #25, "RUNWAY 6"; TIMER
' RUNWAY 6
runwaylatdeg6 = VAL(MID$(runwaylat6$, 1, 2))
runwaylatmin6 = VAL(MID$(runwaylat6$, 4, 6))
RunWayLatAbsCoord6 = (runwaylatdeg6 * 60) + runwaylatmin6

runwaylondeg6 = VAL(MID$(runwaylon6$, 1, 3))
runwaylonmin6 = VAL(MID$(runwaylon6$, 5, 6))
RunWayLonAbsCoord6 = (runwaylondeg6 * 60) + runwaylonmin6

'print #25, "RUNWAY 7"; TIMER
' RUNWAY 7
runwaylatdeg7 = VAL(MID$(runwaylat7$, 1, 2))
runwaylatmin7 = VAL(MID$(runwaylat7$, 4, 6))
RunWayLatAbsCoord7 = (runwaylatdeg7 * 60) + runwaylatmin6

runwaylondeg7 = VAL(MID$(runwaylon7$, 1, 3))
runwaylonmin7 = VAL(MID$(runwaylon7$, 5, 6))
RunWayLonAbsCoord7 = (runwaylondeg7 * 60) + runwaylonmin6

magvar = VAL(magvar$)

'CAL***************** CACULATE AIRPORT LOCATION **********************
MapLat$ = MID$(FileLat$, 1, 2)
FileLatDeg = VAL(MapLat$)

FileLatMin = VAL(MID$(FileLat$, 4, 6))

'filelatmin100 = FileLatMin / 60         'convert min to dec parts of a degree
'filelatangle = (FileLatDeg + filelatmin100) / 57.295  'convert to radians

FileLatCoord = (FileLatDeg * 60) + FileLatMin
'naut. miles north of the equator
MapLon$ = MID$(FileLon$, 1, 3)
filelondeg = VAL(MapLon$)
filelonmin = VAL(MID$(FileLon$, 5, 6))
FileLonCoord = (filelondeg * 60) + filelonmin
'naut. miles west of the meridan

fieldelevation = VAL(fieldelevation$)

GOTO finishedwitharray

'GAA**********  GOT AN AIRPORT ******************************

'12345678901234567890
'"A","TOA ",
'"A", Ident$, Name$, City$, State$, Lat$, Lon$, elevat$, var$, PatAlt$, Lights$,
fueltype$, tower$, ctaf$

gotanairport:
```

```
INPUT #40, Ident$, Name$, City$, State$, FileLat$, FileLon$, fieldelevation$,
magvar$, patternaltitude$, Lights$, fueltype$, tower$, CTAF$

EorW$ = MID$(magvar$, 3, 1)
ft1$ = RTRIM$(MID$(fueltype$, 1, 5)) + " "
ft2$ = RTRIM$(MID$(fueltype$, 6, 5)) + " "
ft3$ = RTRIM$(MID$(fueltype$, 11, 5)) + " "
ft4$ = RTRIM$(MID$(fueltype$, 16, 5)) + " "
ft5$ = RTRIM$(MID$(fueltype$, 21, 5)) + " "
ft6$ = RTRIM$(MID$(fueltype$, 26, 5)) + " "
ft7$ = RTRIM$(MID$(fueltype$, 31, 5)) + " "
ft8$ = RTRIM$(MID$(fueltype$, 36, 5))

fueltype$ = RTRIM$(ft1$ + ft2$ + ft3$ + ft4$ + ft5$ + ft6$ + ft7$ + ft8$)
RETURN


' ******************** GOT A FREQUENCY *************************

'"F","TOA ","118.9  ","125.6  ","111.9  "
gotafreq:
INPUT #40, Ident$, towerfreq$, atisfreq$, ilsfreq$
RETURN


'GAR************************ GOT A RUNWAY ************************

GotARunway:
runwaynum = runwaynum + 1
'print #25, "runwaynum "; runwaynum
IF runwaynum = 1 THEN GOTO gotrunway1
IF runwaynum = 2 THEN GOTO gotrunway2
IF runwaynum = 3 THEN GOTO gotrunway3
IF runwaynum = 4 THEN GOTO gotrunway4
IF runwaynum = 5 THEN GOTO gotrunway5
IF runwaynum = 6 THEN GOTO gotrunway6
IF runwaynum = 7 THEN GOTO gotrunway7
RETURN

gotrunway1:
'PRINT #25, "gotrunway1 "; TIMER
INPUT #40, Ident$, RunwayLength1$, Wide1$, BaseEndIdent1$, RunwayHeading1$,
runwaylat1$, runwaylon1$, RHTraf1$, Lights1$, ILSSys1$
'WRITE #25, Ident$, RunwayLength1$, Wide1$, BaseEndIdent1$, RunwayHeading1$,
runwaylat1$, runwaylon1$, RHTraf1$, Lights1$, ILSSys1$
IF MID$(runwaylat1$, 1, 2) = "  " THEN GOSUB getrunwayloc1    'missing data
IF MID$(RunwayHeading1$, 1, 2) = "  " THEN GOSUB getrunwayloc1 'missing data

RETURN

'"A","00A ","STONE MOUNTAIN-BRITT MEMORIAL             ","STONE MOUNTAIN
","GA","33:48.673N","084:07.161W","  986","02W","    ","DUSK-DAWN","MOGAS100LL
","  ","122.800"
'"R","00A ","  2806","  45","17 ","   ","         ","          ","N","LOW  ","
","
"
'"R","00V ","  5999","  60","15 ","150","38:57.208","104:34.476","N","MED  ","
","
"
```

```
'The following is a correction in case runway data is missing from the FAA data
base
getrunwayloc1:
'print #25, FileLat$
'print #25, FileLon$
'PRINT #25, "Bas End identifier "; BaseEndIdent1$
RunwayHeading1$ = RTRIM$(BaseEndIdent1$) + "0"
'PRINT #25, "Runway heading 1 "; RunwayHeading1$
IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
RunwayAngle = ((VAL(RunwayHeading1$)) + EorWCor * VAL(magvar$)) / 57.295

IF (VAL(RunwayHeading1$)) + EorWCor * VAL(magvar$) >= 90 AND (VAL(RunwayHeading1$))
+ EorWCor * VAL(magvar$) < 270 THEN anglatcor = 1 ELSE anglatcor = -1
IF (VAL(RunwayHeading1$)) + EorWCor * VAL(magvar$) >= 0 AND (VAL(RunwayHeading1$))
+ EorWCor * VAL(magvar$) < 180 THEN angloncor = 1 ELSE angloncor = -1

'PRINT #25, "Runway angle "; RunwayAngle
latcorr = ((COS(RunwayAngle) * VAL(RunwayLength1$)) / 2) / 6080
'print #25, "latcorr in minutes "; latcorr
'print #25, "latcorr in feet "; ((COS(RunwayAngle) * VAL(RunwayLength1$)) / 2)

LonCorr = ((SIN(RunwayAngle) * VAL(RunwayLength1$)) / 2) / 6080
'print #25, "loncorr in minutes "; loncorr
'print #25, "loncorr in feet "; ((SIN(RunwayAngle) * VAL(RunwayLength1$)) / 2)

IF MID$(FileLat$, 4, 1) = "0" THEN fix0$ = "0" ELSE fix0$ = ""
runwaylat1$ = MID$(FileLat$, 1, 2) + ":" + fix0$ + LTRIM$(STR$(VAL(MID$(FileLat$,
4, 5)) + anglatcor * latcorr))
'print #25, runwaylat1$
IF MID$(FileLon$, 5, 1) = "0" THEN fix0$ = "0" ELSE fix0$ = ""
runwaylon1$ = MID$(FileLon$, 1, 3) + ":" + fix0$ + LTRIM$(STR$(VAL(MID$(FileLon$,
5, 5)) + angloncor * LonCorr))
'print #25, runwaylon1$
RETURN


gotrunway2:
INPUT #40, Ident$, runwaylength2$, Wide2$, BaseEndIdent2$, runwayheading2$,
runwaylat2$, runwaylon2$, RHTraf2$, Lights2$, ILSSys2$
RETURN

gotrunway3:
INPUT #40, Ident$, runwaylength3$, Wide3$, BaseEndIdent3$, runwayheading3$,
runwaylat3$, runwaylon3$, RHTraf3$, Lights3$, ILSSys3$
RETURN

gotrunway4:
INPUT #40, Ident$, runwaylength4$, Wide4$, BaseEndIdent4$, runwayheading4$,
runwaylat4$, runwaylon4$, RHTraf4$, Lights4$, ILSSys4$
RETURN

gotrunway5:
INPUT #40, Ident$, runwaylength5$, Wide5$, BaseEndIdent5$, runwayheading5$,
runwaylat5$, runwaylon5$, RHTraf5$, Lights5$, ILSSys5$
RETURN

gotrunway6:
```

```
INPUT #40, Ident$, runwaylength6$, Wide6$, BaseEndIdent6$, runwayheading6$,
runwaylat6$, runwaylon6$, RHTraf6$, Lights6$, ILSSys6$
RETURN

gotrunway7:
INPUT #40, Ident$, RunwayLength7$, Wide7$, BaseEndIdent7$, runwayheading7$,
runwaylat7$, runwaylon7$, RHTraf7$, Lights7$, ILSSys7$
'print #25, RunwayLength7$
RETURN


' ********************** GOT A CIRCLE ************************

'C Circle Number 1 ,33:57.068,118:24.65754,11,214,290,
'C Circle Number 2 ,33:55.862,118:25.47945,10,65,90,


StoreACircle:         'return with colon position in cpos
circleflag = 1
FOR x = 1 TO 2
INPUT #35, test$
FOR cpos = 1 TO 4
degpos$ = MID$(test$, cpos, 1)
IF degpos$ = ":" THEN GOTO endfindcircolonpos
NEXT cpos
endfindcircolonpos:
deg = VAL(MID$(test$, 1, cpos))
min = VAL(MID$(test$, (cpos + 1), 6))
abscoord = (deg * 60) + min
pcircledata(pci) = abscoord
pci = pci + 1
NEXT x

FOR x = 1 TO 3
INPUT #35, test$
abscoord = VAL(test$)
pcircledata(pci) = abscoord
pci = pci + 1
NEXT x
lastpci = pci
RETURN

'GAL******************** GOT A LINE ************************

'L- 7  ,33:57.586,118:21.517   ,34:00.689,118:14.896,
'L- 8  ,33:57.586,118:26.482   ,34:00.344,118:37.655,

storealine:
lineflag = 1
FOR x = 1 TO 4
INPUT #35, test$
FOR cpos = 1 TO 4
degpos$ = MID$(test$, cpos, 1)
IF degpos$ = ":" THEN GOTO endfindcolonpos
NEXT cpos
endfindcolonpos:
deg = VAL(MID$(test$, 1, cpos))
min = VAL(MID$(test$, (cpos + 1), 6))
abscoord = (deg * 60) + min
```

```
plinedata(pli) = abscoord
pli = pli + 1
NEXT x
lastpli = pli
RETURN

'FWA*********** FINISHED WITH ARRAY ********************

finishedwitharray:
CLOSE #19

distogo = 14

'ST*************************  START TIMER **********************

ON TIMER(2) GOSUB calculatedata
TIMER ON
screenwidth30:
widthofscreen = 30           'initial width of screen in miles
GOTO twbts:

screenwidth15:
widthofscreen = 15
GOTO twbts

screenwidth75:
widthofscreen = 7.5
GOTO twbts

screenwidth375:
widthofscreen = 3.75
GOTO twbts

screenwidth1825:
widthofscreen = 1.825
GOTO twbts

twbts:
TIMER STOP
scale = 400 / widthofscreen     'number of pixels per n-mile or per minute
GOSUB FindLowerRightCorner
FltStartedFlag = 1
'GOSUB debugplotscreen
PlotScreenFlag = 1
'AirportDatScrFlag = 0
CLS 0
SCREEN 12
WIDTH 80, 60

VIEW (corner0, corner1)-(corner2, corner3), , 15

GOSUB LayoutPlotScreen

IF lineflag = 1 THEN GOSUB PlotAllLines
IF circleflag = 1 THEN GOSUB PlotAllCircles
GOSUB PlotRunway1
IF runwaynum > 1 THEN GOSUB PlotRunway2
IF runwaynum > 2 THEN GOSUB PlotRunway3
IF runwaynum > 3 THEN GOSUB PlotRunway4
```

```
IF runwaynum > 4 THEN GOSUB PlotRunway5
IF runwaynum > 5 THEN GOSUB PlotRunway6
IF runwaynum > 6 THEN GOSUB PlotRunway7
'IF runwaynum > 7 THEN GOSUB PlotRunway8

TIMER ON

'PS*******************  PLOT INFO SCREEN LOOP ******************************

PlotScreenInfLoop:
key$ = INKEY$
IF key$ = CHR$(27) THEN GOTO endit                          'esc to exit

IF distogo <= 7.5 AND widthofscreen = 30 THEN GOTO screenwidth15
IF distogo <= 3.75 AND widthofscreen = 15 THEN GOTO screenwidth75
IF distogo <= 1.825 AND widthofscreen = 7.5 THEN GOTO screenwidth375
IF distogo <= .912 AND widthofscreen = 3.75 THEN GOTO screenwidth1825
IF distogo > .912 AND widthofscreen = 1.825 THEN GOTO screenwidth375
IF distogo > 1.825 AND widthofscreen = 3.75 THEN GOTO screenwidth75
IF distogo > 3.75 AND widthofscreen = 7.5 THEN GOTO screenwidth15
IF distogo > 7.5 AND widthofscreen = 15 THEN GOTO screenwidth30
IF key$ = CHR$(27) THEN GOTO endit                          'esc to exit

GOTO PlotScreenInfLoop

'********  CACLULATE DATA  SUB HAPPENS EVERY TWO SECONDs
calculatedata:                              'happens every two seconds

'GOSUB debugcalcdata

GOSUB eraseplane    'must be in front of garmin

GOSUB garmin

'xdif = (GPSLonCoord * (COS(GPSLatAngle)) - fileloncoord * (COS(filelatangle)))

LonCorr = COS(((((GPSLatCoord + FileLatCoord) / 2)) \ 60) / 57.295)
xdif = LonCorr * (GPSLonCoord - FileLonCoord)
ydif = GPSLatCoord - FileLatCoord
distogo = (xdif ^ 2 + ydif ^ 2) ^ .5
IF distogo > 99 THEN distogo = 99
' > is greater than

gslopealt = (distogo * 318.3488) + fieldelevation
IF gslopealt > 9999 THEN gslopealt = 9999

'sin(3/57.295)*6080  3 degree glide slope
heading = VAL(heading$)
magheading = heading + magvar
IF magheading > 360 THEN magheading = magheading - 360
magheading$ = LTRIM$(STR$(magheading))
GOSUB plotplane                     '726
GOSUB PrtPlotScreen                 '979
endcalc:
RETURN            'go back to back ground


'******************  GARMIN ***************************************
garmin:
```

```
index1 = 0
IF LOC(9) < 1 THEN GOTO nodatacomingin   'real important
gartemp$ = INPUT$(LOC(9), #9)    'this is the only thing that works
FOR x = 1 TO LEN(gartemp$)
gartemp2$ = MID$(gartemp$, x, 1)
IF gartemp2$ = "$" THEN GOSUB checkit
NEXT x
gtemp$ = gartemp4$



'BFDR******************* BACK FROM DEAD RECKONING *******************
'BackFromDR:

SEEK #45, 1               'temp 45 has been opened for random for this trick
PRINT #45, gtemp$
SEEK #45, 1

IF LEN(gtemp$) <> 70 THEN GOTO skipgtmp

INPUT #45, sent$, sattime$, aye$, GPSLat$, LatNorS$, GPSLon$, LonEorW$, airspeed$,
heading$, satdate$, blank$, checksum$

WRITE #12, sattime$, GPSLat$, LatNorS$, GPSLon$, LonEorW$, airspeed$, heading$,
satdate$ '12 is the play back file

skipgtmp:

'                123456789 123456789  12345 12345        12345
'$GPRMC,221842,A,3455.98,N,11823.17,W,089.0,000.0,261092, ,E*6A
' 12345678901234567890123456789012345678901234567890123456789012345678901234567890
'          1         2         3         4         5         6


IF aye$ = "A" THEN datavalid$ = "GPS DATA VALID  "
IF aye$ = "V" THEN datavalid$ = "GPS DATA NO GOOD"

'IF simulationflag = 1 THEN datavalid$ = " PLAYBACK"

'The trouble with lat min is a degree boundry could be right in the
'middle of your approach plate

GPSLatMin$ = MID$(GPSLat$, 3, 5)     'needed by lay out print screen
GPSLatMin = VAL(GPSLatMin$)
GPSLatDeg$ = MID$(GPSLat$, 1, 2)
GPSLatDeg = VAL(GPSLatDeg$)
GPSLatCoord = (GPSLatDeg * 60) + GPSLatMin   '=naut miles above the equator

GPSLonDeg$ = MID$(GPSLon$, 1, 3)
GPSLonMin$ = MID$(GPSLon$, 4, 5)
GPSLonMin = VAL(GPSLonMin$)
GPSLonDeg = VAL(GPSLonDeg$)
GPSLonCoord = (GPSLonDeg * 60) + GPSLonMin   'miles west of the meridan

airspeed = VAL(airspeed$)

second$ = MID$(sattime$, 5, 3)   'time from the GPS
minute$ = MID$(sattime$, 3, 2)
```

```
hour$ = MID$(sattime$, 1, 2)

'set GMT to atomic clock in satellite.

GMT$ = hour$ + ":" + minute$ + ":" + second$
endgarminsub:
RETURN

'Old checkit prior to Bob's loran
'checkit:
'gartemp3$ = MID$(gartemp$, (x + 3), 3)
'IF gartemp3$ = "RMC" THEN GOSUB gotlatlon
'RETURN

'gotlatlon:
'gartemp4$ = MID$(gartemp$, x, 70)
'TIMER OFF
'TIMER ON
'RETURN


checkit:
'PRINT #25, "check it sub "; TIMER
gartemp3$ = MID$(gartemp$, (x + 3), 3)
IF gartemp3$ = "RMC" THEN GOTO RMC

IF gartemp3$ = "GLL" THEN GOTO GLL

'PRINT #25, " end check it sub "; TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
endcheckit:
RETURN


RMC:
gartemp4$ = MID$(gartemp$, x, 70)
TIMER OFF
TIMER ON
GOTO endcheckit

GLL:
IF LEN(gartemp$) <> 140 THEN GOTO endcheckit
gartemp5$ = MID$(gartemp$, x, 140)

'         1         2         3         4         5         6         7         8
9         10        11        12        13
'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890123456789012345678901234567890
'$LCGLL,3016.50,N,08350.63,W  $LCBWC,,3500.00,N,12000.00,W,000,T,001,M,000.0,N,099
$LCAPA,A,A,0.00,R,N,V,A,348,M,099  $LCVTX,001,M,000,N--
'$LCGLL,3045.17,N,08129.38,W  $LCBWC,,3016.51,N,08350.63,W,257,T,260,M,125.1,N,099
$LCAPA,A,A,5.04,L,N,V,A,347,M,099  $LCVTX,115,M,001,N


garlat$ = MID$(gartemp5$, 8, 7) + "0,N,"
garlon$ = MID$(gartemp5$, 18, 8) + "0,W,"
garhead$ = MID$(gartemp5$, 59, 3)
garspeed$ = MID$(gartemp5$, 132, 3)
```

```
gartemp4$ = "$GPRMC,221355.756536,A," + garlat$ + garlon$ + "0" + garspeed$ + ".0,"
+ garhead$ + ",030793,,, *20"

'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
34567890
            '         1         2         3         4         5         6         7

'CLS
'PRINT gartemp4$
'PRINT "LEN gartemp4$ "; LEN(gartemp4$)
'PRINT "LEN gartemp$ "; LEN(gartemp$)
'PRINT "$GPRMC,221355.756536,A,4712.114,N,12213.465,W,0096.5,177,030793,,,*20"

'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
34567890
       '         1         2         3         4         5         6         7
'garcnt = garcnt + 1
'PRINT garcnt
TIMER OFF
TIMER ON
GOTO endcheckit


'NDCI************** NO DATA COMMING IN ******************************
nodatacomingin:
'IF simulationflag = 1 THEN GOTO simulation:
BEEP
LOCATE 8, 5: PRINT "                                                  "
LOCATE 9, 5: PRINT "                                                  "
LOCATE 10, 5: PRINT "***** NO DATA COMING IN FROM GPS SENSOR *****"
delayone = TIMER
ndci:
key$ = INKEY$
IF key$ = CHR$(27) THEN GOTO endit                          'esc to exit
delaytwo = TIMER
IF delaytwo < delayone + 4 THEN GOTO ndci
LOCATE 10, 5: PRINT "                                                  "
GOTO endgarminsub


'PR1************************  PLOT RUN WAY 1 *******************************
PlotRunway1:

xpixcoordf = 400 - ((FileLonCoord - lowrcorlon) * scale)
ypixcoordf = 400 - ((FileLatCoord - lowrcorlat) * scale)

CIRCLE (xpixcoordf, ypixcoordf), 1    'mark the location of the Airport
'print #25, "run way length string "; RunwayLength1$

RnyLen1 = VAL(LTRIM$(RunwayLength1$))
IF RnyLen1 = 0 THEN RnyLen1 = 800
rnyhdg1 = VAL(LTRIM$(RunwayHeading1$))       'this is the mag heading

'22 ? could be 223 225 227  229 ?  you need to know.

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg21 = rnyhdg1 + EorWCor * INT(magvar)    'convert to true north +fudge
```

```
IF rnyhdg21 >= 360 THEN rnyhdg21 = rnyhdg21 - 360

opprnyhdg1 = rnyhdg1 + 180                              'get opposite heading
IF opprnyhdg1 >= 360 THEN opprnyhdg1 = opprnyhdg1 - 360
runwayheadings1$ = STR$(rnyhdg1) + "-" + LTRIM$(STR$(opprnyhdg1))

rnyhdg21 = rnyhdg21 + 90
IF rnyhdg21 > 360 THEN rnyhdg21 = rnyhdg21 - 360 ELSE rnyhdg21 = rnyhdg21
rnyhdg21 = rnyhdg21 * (-1)       'to get clockwise degrees

'All this crap to get it to start at strait up and rotate clockwise
'other wise the stupid DRAW program starts at 270 and rotates counter
'clock wise.


xpixcoord1 = 400 - ((RunWayLonAbsCoord1 - lowrcorlon) * scale)
ypixcoord1 = 400 - ((RunWayLatAbsCoord1 - lowrcorlat) * scale)

PSET (xpixcoord1, ypixcoord1)

RnyScaleFactor1 = 105263 / RnyLen1
scale21 = scale / RnyScaleFactor1    '/10 for 5000 feet   x/10 =5000
IF scale21 < 1 THEN scale21 = 1


RS$ = "S" + STR$(CINT(scale21))
DRAW "TA=" + VARPTR$(rnyhdg21) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'*************************  PLOT RUN WAY 2 *********************************
PlotRunway2:

RnyLen2 = VAL(LTRIM$(runwaylength2$))
rnyhdg2 = VAL(LTRIM$(runwayheading2$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg22 = rnyhdg2 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg22 >= 360 THEN rnyhdg22 = rnyhdg22 - 360

opprnyhdg2 = rnyhdg2 + 180                              'get opposite heading
IF opprnyhdg2 >= 360 THEN opprnyhdg2 = opprnyhdg2 - 360
runwayheadings2$ = STR$(rnyhdg2) + "-" + LTRIM$(STR$(opprnyhdg2))

rnyhdg22 = rnyhdg22 + 90
IF rnyhdg22 > 360 THEN rnyhdg22 = rnyhdg22 - 360 ELSE rnyhdg22 = rnyhdg22
rnyhdg22 = rnyhdg22 * (-1)       'to get clockwise degrees

xpixcoord2 = 400 - ((RunWayLonAbsCoord2 - lowrcorlon) * scale)
ypixcoord2 = 400 - ((RunWayLatAbsCoord2 - lowrcorlat) * scale)

PSET (xpixcoord2, ypixcoord2)

RnyScaleFactor2 = 105263 / RnyLen2
scale22 = scale / RnyScaleFactor2    '/10 for 5000 feet   x/10 =5000
IF scale22 < 1 THEN scale22 = 1      'x = 50000

RS$ = "S" + STR$(CINT(scale22))
DRAW "TA=" + VARPTR$(rnyhdg22) + RS$ + "D1 L80 U2 R80 D1"
RETURN
```

```
'************************  PLOT RUN WAY 3 *******************************
PlotRunway3:

RnyLen3 = VAL(LTRIM$(runwaylength3$))
rnyhdg3 = VAL(LTRIM$(runwayheading3$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg23 = rnyhdg3 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg23 >= 360 THEN rnyhdg23 = rnyhdg23 - 360

opprnyhdg3 = rnyhdg3 + 180                              'get opposite heading
IF opprnyhdg3 >= 360 THEN opprnyhdg3 = opprnyhdg3 - 360
runwayheadings3$ = STR$(rnyhdg3) + "-" + LTRIM$(STR$(opprnyhdg3))

rnyhdg23 = rnyhdg23 + 90
IF rnyhdg23 > 360 THEN rnyhdg23 = rnyhdg23 - 360 ELSE rnyhdg23 = rnyhdg23
rnyhdg23 = rnyhdg23 * (-1)       'to get clockwise degrees

xpixcoord3 = 400 - ((RunWayLonAbsCoord3 - lowrcorlon) * scale)
ypixcoord3 = 400 - ((RunWayLatAbsCoord3 - lowrcorlat) * scale)

PSET (xpixcoord3, ypixcoord3)

RnyScaleFactor3 = 105263 / RnyLen3
scale23 = scale / RnyScaleFactor3    '/10 for 5000 feet   x/10 =5000
IF scale23 < 1 THEN scale23 = 1      'x = 50000

RS$ = "S" + STR$(CINT(scale23))
DRAW "TA=" + VARPTR$(rnyhdg23) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'************************  PLOT RUN WAY 4 *******************************
PlotRunway4:

RnyLen4 = VAL(LTRIM$(runwaylength4$))
rnyhdg4 = VAL(LTRIM$(runwayheading4$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg24 = rnyhdg4 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg24 >= 360 THEN rnyhdg24 = rnyhdg24 - 360

opprnyhdg4 = rnyhdg4 + 180                              'get opposite heading
IF opprnyhdg4 >= 360 THEN opprnyhdg4 = opprnyhdg4 - 360
runwayheadings4$ = STR$(rnyhdg4) + "-" + LTRIM$(STR$(opprnyhdg4))

rnyhdg24 = rnyhdg24 + 90
IF rnyhdg24 > 360 THEN rnyhdg24 = rnyhdg24 - 360 ELSE rnyhdg24 = rnyhdg24
rnyhdg24 = rnyhdg24 * (-1)       'to get clockwise degrees

xpixcoord4 = 400 - ((RunWayLonAbsCoord4 - lowrcorlon) * scale)
ypixcoord4 = 400 - ((RunWayLatAbsCoord4 - lowrcorlat) * scale)

PSET (xpixcoord4, ypixcoord4)

RnyScaleFactor4 = 105263 / RnyLen4
scale24 = scale / RnyScaleFactor4    '/10 for 5000 feet   x/10 =5000
IF scale24 < 1 THEN scale24 = 1      'x = 50000
```

```
RS$ = "S" + STR$(CINT(scale24))
DRAW "TA=" + VARPTR$(rnyhdg24) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'************************  PLOT RUN WAY 5 *******************************
PlotRunway5:

RnyLen5 = VAL(LTRIM$(runwaylength5$))
rnyhdg5 = VAL(LTRIM$(runwayheading5$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg25 = rnyhdg5 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg25 >= 360 THEN rnyhdg25 = rnyhdg25 - 360

opprnyhdg5 = rnyhdg5 + 180                              'get opposite heading
IF opprnyhdg5 >= 360 THEN opprnyhdg5 = opprnyhdg5 - 360
runwayheadings5$ = STR$(rnyhdg5) + "-" + LTRIM$(STR$(opprnyhdg5))

rnyhdg25 = rnyhdg25 + 90
IF rnyhdg25 > 360 THEN rnyhdg25 = rnyhdg25 - 360 ELSE rnyhdg25 = rnyhdg25
rnyhdg25 = rnyhdg25 * (-1)       'to get clockwise degrees

xpixcoord5 = 400 - ((RunWayLonAbsCoord5 - lowrcorlon) * scale)
ypixcoord5 = 400 - ((RunWayLatAbsCoord5 - lowrcorlat) * scale)

PSET (xpixcoord5, ypixcoord5)

RnyScaleFactor5 = 105263 / RnyLen5
scale25 = scale / RnyScaleFactor5    '/10 for 5000 feet   x/10 =5000
IF scale25 < 1 THEN scale25 = 1      'x = 50000

RS$ = "S" + STR$(CINT(scale25))
DRAW "TA=" + VARPTR$(rnyhdg25) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'************************  PLOT RUN WAY 6 *******************************
PlotRunway6:

RnyLen6 = VAL(LTRIM$(runwaylength6$))
rnyhdg6 = VAL(LTRIM$(runwayheading6$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg26 = rnyhdg6 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg26 >= 360 THEN rnyhdg26 = rnyhdg26 - 360

opprnyhdg6 = rnyhdg6 + 180                              'get opposite heading
IF opprnyhdg6 >= 360 THEN opprnyhdg6 = opprnyhdg6 - 360
runwayheadings6$ = STR$(rnyhdg6) + "-" + LTRIM$(STR$(opprnyhdg6))

rnyhdg26 = rnyhdg26 + 90
IF rnyhdg26 > 360 THEN rnyhdg26 = rnyhdg26 - 360 ELSE rnyhdg26 = rnyhdg26
rnyhdg26 = rnyhdg26 * (-1)       'to get clockwise degrees

xpixcoord6 = 400 - ((RunWayLonAbsCoord6 - lowrcorlon) * scale)
ypixcoord6 = 400 - ((RunWayLatAbsCoord6 - lowrcorlat) * scale)

PSET (xpixcoord6, ypixcoord6)

RnyScaleFactor6 = 105263 / RnyLen6
```

```
scale26 = scale / RnyScaleFactor6    '/10 for 5000 feet   x/10 =5000
IF scale26 < 1 THEN scale26 = 1      'x = 50000

RS$ = "S" + STR$(CINT(scale26))
DRAW "TA=" + VARPTR$(rnyhdg26) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'************************  PLOT RUN WAY 7 *******************************
PlotRunway7:

RnyLen7 = VAL(LTRIM$(RunwayLength7$))
rnyhdg7 = VAL(LTRIM$(runwayheading7$))       'this is the mag heading

IF EorW$ = "E" THEN EorWCor = 1 ELSE EorWCor = -1
rnyhdg27 = rnyhdg7 + EorWCor * INT(magvar)        'convert to true north +fudge
IF rnyhdg27 >= 360 THEN rnyhdg27 = rnyhdg27 - 360

opprnyhdg7 = rnyhdg7 + 180                              'get opposite heading
IF opprnyhdg7 >= 360 THEN opprnyhdg7 = opprnyhdg7 - 360
runwayheadings7$ = STR$(rnyhdg7) + "-" + LTRIM$(STR$(opprnyhdg7))

rnyhdg27 = rnyhdg27 + 90
IF rnyhdg27 > 360 THEN rnyhdg27 = rnyhdg27 - 360 ELSE rnyhdg27 = rnyhdg27
rnyhdg27 = rnyhdg27 * (-1)       'to get clockwise degrees

xpixcoord7 = 400 - ((RunWayLonAbsCoord7 - lowrcorlon) * scale)
ypixcoord7 = 400 - ((RunWayLatAbsCoord7 - lowrcorlat) * scale)

PSET (xpixcoord7, ypixcoord7)

RnyScaleFactor7 = 105263 / RnyLen7
scale27 = scale / RnyScaleFactor7    '/10 for 5000 feet   x/10 =5000
IF scale27 < 1 THEN scale27 = 1      'x = 50000

RS$ = "S" + STR$(CINT(scale27))
DRAW "TA=" + VARPTR$(rnyhdg27) + RS$ + "D1 L80 U2 R80 D1"
RETURN

'PAL*********     PLOT ALL LINES

PlotAllLines:
scrsize = 400
pli = 1
pal2:
ycat = plinedata(pli)
yc1 = scrsize - ((ycat - lowrcorlat) * scale)
pli = pli + 1
xcat = plinedata(pli)
xc1 = scrsize - ((xcat - lowrcorlon) * scale)
pli = pli + 1
ycbt = plinedata(pli)
yc2 = scrsize - ((ycbt - lowrcorlat) * scale)
pli = pli + 1
xcbt = plinedata(pli)
xc2 = scrsize - ((xcbt - lowrcorlon) * scale)
pli = pli + 1
LINE (xc1, yc1)-(xc2, yc2)
IF pli < (lastpli + 1) GOTO pal2
RETURN
```

```
'PAC************************** PLOT ALL CIRCLES
PlotAllCircles:
pci = 1
pac2:
circcorlat = pcircledata(pci)
pci = pci + 1
circcorlon = pcircledata(pci)
pci = pci + 1
radius = (pcircledata(pci) * scale) * 1.18
pci = pci + 1
angle1 = pcircledata(pci)
pci = pci + 1
angle3 = pcircledata(pci)
GOSUB plotcircle
pci = pci + 1
IF pci < lastpci THEN GOTO pac2
RETURN


' ************************    PLOT CIRCLE

'|  field 1            |    2    |    3     |   4    |  5   | 6    |  7  |  8   |
'C-   A Circle         , 33:56.00, 118:25.87, 11     , 215  , 300  ,     ,

plotcircle:        'Don't dick with this! It is very tricky indeed.
xcoord1 = (circcorlon - lowrcorlon) * scale  'now in pixels
xcoord2 = 400 - xcoord1
xcpixcor = CINT(xcoord2)
ycoord1 = (circcorlat - lowrcorlat) * scale  'now in pixels
ycoord2 = 400 - ycoord1
ycpixcor = CINT(ycoord2)

'|  field 1            |    2    |    3     |   4    |  5   | 6    |  7  |  8   |
'C-   A Circle         , 33:56.00, 118:25.87, 11     , 215  , 300  ,     ,

IF angle1 > 90 THEN angle2 = 90 + 360 - angle1 ELSE angle2 = 90.1 - angle1
 startc = (angle2 / 57.295)   '* (-1)  'if you make it - useful for debug
 IF angle3 > 90 THEN angle4 = 90 + 360 - angle3 ELSE angle4 = 90.1 - angle3
endc = (angle4 / 57.295)   '* (-1)   'if you make it minus useful for debug

'           x     ,    y   ,            ,start, end      Normally
CIRCLE (xcpixcor, ycpixcor), radius, , endc, startc
'x is horz increasing to left  and y is verticle increasing down
'GOSUB debugcircle
RETURN

'The trick is the end of the circle becomes the start because this
'stupid function plots counter clockwise, start and end are reversed
'                        str end
'CIRCLE (100, 200), 50, , 0, 2
'This bloody fucking function starts at 90 due east by God & for Christ
'sakes and rotates counter clockwise jeees
'DRAW does just the opposite for fuck sakes on the start of the angle


'**********************       PLOT VOR
plotvor:
plotvorlat3 = winsize - ((VAL(MID$(field2$, 4, 6)) * scale)) 'in minutes
```

```
plotvorlong$ = field3$
temp$ = field3$: GOSUB findcolon      'enter with string in temp$
plotvorlong3 = winsize - (VAL(MID$(plotvorlong$, (cp + 1), 2)) * scale)
'in minutes
pcolnum% = INT(plotvorlong3 / 8)                        '640/80 = 8
PSET (plotvorlong3, plotvorlat3)
DRAW "BL3 U3 R6 D6 L6 U3"       'vor
prownum% = INT(plotvorlat3 / 8)
prownum% = (prownum% + 3)
pcolnum% = (pcolnum% + 2)
IF prownum% < 5 THEN prownum% = 5
IF pcolnum% <= 0 THEN pcolnum% = 3
IF pcolnum% >= 50 THEN pcolnum% = 50
IF prownum% >= 56 THEN prownum% = 56
Ident$ = MID$(field1$, 3, 3)
LOCATE prownum%, pcolnum%: PRINT Ident$;
RETURN

findcolon:
FOR cp = 1 TO 5
p$ = MID$(temp$, cp, 1)
IF p$ = ":" THEN GOTO gotit44:
NEXT cp
gotit44:
RETURN           'return with colon position in cp


'EP******************       ERASE PLANE

eraseplane:
 ON ERROR GOTO ErasePlaneError
 headingpr = -1 * heading

 phorzpos = 400 - (GPSLonCoord - lowrcorlon) * scale
 pvertpos = 400 - (GPSLatCoord - lowrcorlat) * scale

 PSET (phorzpos, pvertpos)         'erase horz, vert
 DRAW "C0"
RS$ = "S" + STR$(4)
DRAW "TA=" + VARPTR$(headingpr) + RS$ + "L10 R20 L10 D1 L10 R20 L10 U1 U5 D15 L4 R8
L4 U15 D5"

enderaseplane:
ON ERROR GOTO 0
RETURN

ErasePlaneError:
'print #25, "eraseplane error "
'print #25, "phorzpos "; phorzpos
'print #25, "pvertpos "; pvertpos
RESUME enderaseplane

'PA******************* PLOT AIRPLANE
plotplane:
'GOSUB debugplotplane
headingpr = -1 * heading
 phorzpos = 400 - (GPSLonCoord - lowrcorlon) * scale
 pvertpos = 400 - (GPSLatCoord - lowrcorlat) * scale
PSET (phorzpos, pvertpos)         'horz, vert
```

```
DRAW "C15"
RS$ = "S" + STR$(4)
DRAW "TA=" + VARPTR$(headingpr) + RS$ + "L10 R20 L10 D1 L10 R20 L10 U1 U5 D15 L4 R8
L4 U15 D5"
RETURN



'************    GET PIXEL

'return with lat pixel in templat and lon pixel in templon

getpix:
templat = ((VAL(MID$(templat$, 1, 2))) * 60) + VAL(MID$(templat$, 4, 5))
temp$ = templon$: GOSUB findcolon
templon = ((VAL(MID$(templon$, 1, cp))) * 60) + VAL(MID$(templon$, (cp + 1), 5))

templat = winsize - (templat - lowbndry) * scale
templon = winsize - (templon - rightbndry) * scale
RETURN

'PO************        PLOT OBSTRUCTION
plotobstruction:
obslat$ = "33:46.35"
obslon$ = "118:19.75"
'getpix:  Enter with NNNlat$ in templat$ and NNNlon$ in templon$
'return with lat pixel in templat and lon pixel in templon
templat$ = obslat$
templon$ = obslon$
GOSUB getpix
PSET (templon, templat)
obsang = 0
RS$ = "S" + STR$(CINT(scale / 8))
DRAW "TA=" + VARPTR$(obsang) + RS$ + "BL2 E3 F3"

DIM arrayn(1 TO 10) AS STRING * 80    'This is in leu of a bunch of DATA
OPEN "NUMBERS" FOR INPUT AS #3        'statements.
FOR x = 1 TO 10                       'for letters I think we will use
LINE INPUT #3, arrayn(x)              'the ascii decimal values -90
NEXT x
CLOSE #3

ht$ = "305"

templat = templat - 11
templon = templon - 10

ht1 = VAL(MID$(ht$, 1, 1))
IF ht1 = 0 THEN ht1 = 10
PSET (templon, templat)
DRAW "S8" + arrayn(ht1)
templon = templon + 9

ht2 = VAL(MID$(ht$, 2, 1))
IF ht2 = 0 THEN ht2 = 10
PSET (templon, templat)
DRAW "S8" + arrayn(ht2)
templon = templon + 9
```

```
ht3 = VAL(MID$(ht$, 3, 1))
IF ht3 = 0 THEN ht3 = 10
PSET (templon, templat)
DRAW "S8" + arrayn(ht3)

ht4 = VAL(MID$(ht$, 4, 1))              'All this to handle 3 or 4 digits
IF ht4 = 0 THEN ht4 = 10
IF MID$(ht$, 4, 1) = "" THEN ht4 = 11   'Give it a phoney error
templon = templon + 9
PSET (templon, templat)
ON ERROR GOTO DrawObjError:
DRAW "S8" + arrayn(ht4)                'Phoney error pays off :)
ON ERROR GOTO 0
RETURN

DrawObjError:
RESUME NEXT

'****************     GET LINE
GetLine:
 temp$ = ""                            'Erase the temp string.
 FOR ColNum = colgetlin TO 24 + NumChar       'Get all characters on the line.
 charactval = SCREEN(rownum, ColNum)   'Read the screen memory.
 Charact$ = CHR$(charactval)
 temp$ = temp$ + Charact$              'Concatenate all the characters.
 NEXT ColNum
 temp$ = RTRIM$(temp$)                 'Trim trailing right blanks
 temp$ = LTRIM$(temp$)                 'Trim leading left blanks.
 RETURN                                'Go back with the data in the Temp$


'LPS***************** LAYOUT  PLOT SCREEN  *********************

LayoutPlotScreen:
cn2 = 53
LOCATE 3, 1: PRINT " <----------- SCREEN WIDTH MILES"; '
LOCATE 3, 38: PRINT " ------------->";


LOCATE 7, cn2: PRINT "   LAT POSITION  ";
LOCATE 9, cn2: PRINT "   LON POSITION  ";
LOCATE 11, cn2: PRINT " GREENWICH TIME  ";
LOCATE 13, cn2: PRINT "   TRUE  HEADING  ";
LOCATE 15, cn2: PRINT "GROUND SPEED KTS  ";

LOCATE 19, cn2: PRINT "  DEPARTURE FREQ  ";
LOCATE 21, cn2: PRINT "   APPROACH FREQ  ";
LOCATE 23, cn2: PRINT "       ATIS FREQ  ";
LOCATE 25, cn2: PRINT "      TOWER FREQ  ";
LOCATE 27, cn2: PRINT "        ILS FREQ  ";
LOCATE 29, cn2: PRINT "     GROUND FREQ  ";
LOCATE 31, cn2: PRINT "            CTAF  ";
LOCATE 33, cn2: PRINT " RUNWAY LIGHTING  ";

LOCATE 35, cn2: PRINT "  FIELD ELEVATION ";
LOCATE 37, cn2: PRINT " PATTERN ALTITUDE ";

LOCATE 40, cn2: PRINT "   DISTANCE TO GO ";
```

```
LOCATE 44, cn2: PRINT " GLIDE SLOPE ALT. ";

LOCATE 48, cn2: PRINT " FUEL TYPES AVAILABLE";

'LOCATE 54, cn2: PRINT MapLat$; ":00"

LOCATE 56, 1: PRINT "                  PRESS ESC TO EXIT            " '; MapLon$;
":00"

LOCATE 59, 1: PRINT " "; Name$; City$; State$; "  "; Ident$;

RETURN

' Comment            lat       lon       twr    atis   ils    apr  dep  gnd

'********************** PRINT PLOT SCREEN DATA  **********************
PrtPlotScreen:
cn1 = 72
LOCATE 3, 33: PRINT "       ";
LOCATE 3, 33: PRINT widthofscreen;
LOCATE 7, (cn1 - 1): PRINT "        ";     'erase
LOCATE 7, (cn1 - 1): PRINT GPSLatDeg$; ":"; GPSLatMin$;
LOCATE 9, (cn1 - 1): PRINT "        ";    'erase
LOCATE 9, (cn1 - 2): PRINT GPSLonDeg$; ":"; GPSLonMin$;
LOCATE 11, (cn1 - 1): PRINT GMT$;
LOCATE 13, cn1: PRINT "        ";
LOCATE 13, (cn1 + 1)
PRINT USING "###"; heading;
LOCATE 15, cn1: PRINT "        ";
LOCATE 15, (cn1 + 1): PRINT airspeed$;
LOCATE 23, cn1: PRINT "       ";
LOCATE 23, (cn1 + 1): PRINT atisfreq$;
LOCATE 25, cn1: PRINT "       ";
LOCATE 25, (cn1 + 1): PRINT towerfreq$;
LOCATE 27, cn1: PRINT "       ";
LOCATE 27, (cn1 + 1): PRINT ilsfreq$;
LOCATE 31, cn1: PRINT "       ";
LOCATE 31, (cn1 + 1): PRINT CTAF$;

IF MID$(Lights$, 1, 4) = "    " THEN Lights$ = " NONE " ELSE Lights$ = Lights$
LOCATE 33, (cn1): PRINT Lights$;

LOCATE 35, cn1: PRINT "       ";
LOCATE 35, (cn1 + 1): PRINT fieldelevation$;
LOCATE 37, cn1: PRINT "       ";
LOCATE 37, (cn1 + 1): PRINT patternaltitude$;
LOCATE 40, cn1: PRINT "         ";
LOCATE 40, (cn1 + 1)
'IF distogo > 30 THEN GOTO endprintit:
PRINT USING "##.###"; distogo;
LOCATE 44, cn1: PRINT "         ";
LOCATE 44, (cn1 + 1)
'IF gslopealt > 50000 THEN GOTO endprintit
PRINT USING "####"; gslopealt;

LOCATE 50, (cn1 - 18): PRINT fueltype$;

'                         GPS DATA NO GOOD
'                          GPS DATA VALID
```

```
LOCATE 8, 20: PRINT "             ";
LOCATE 8, 20: PRINT datavalid$;

endprintit:
RETURN

'"A", Ident$, Name$, City$, State$, Lat$, GPSLon$, elevat$, var$, PatAlt$, Lights$,
fueltype$, tower$, ctaf$

' ****************** END IT ***************************************
endit:
TIMER OFF
CLOSE
KILL "temp40.dat"
KILL "temp45.dat"
CLS 0
CLS
SYSTEM

'*****************************************************************
'                   FIND LOWER RIGHT CORNER

FindLowerRightCorner:

lowrcorlat = FileLatCoord - (widthofscreen / 2)  'in nmiles or minutes
lowrcorlon = FileLonCoord - (widthofscreen / 2)
'print #25, "Find corner "
'print #25, "Lower right lat"; lowrcorlat
'print #25, "Lower right lon"; lowrcorlon
RETURN

'*************************************************************

'  END OF PROGRAM
```