SOME NOTES ON PROGRAMMING IN "C" for the Palm Pilot-

There is a free trial C compiler available for the Palm Pilot, and after 45 days you can purchase it. Also available is the Integrated Development Environment (IDE) which can compile programs for standalone execution on a Palm Pilot. I use the option to add in all the libraries, so no extra work other than a hot synch is needed.

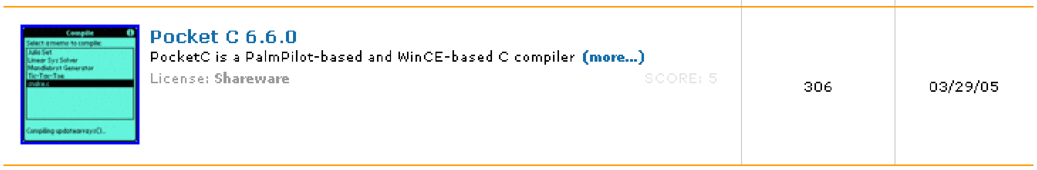
http://www.orbworks.com/pcpalm/download.html
http://orbworks.com/pcpalm/buy.html

See notes at back for Windows VISTA win64 and how to hotsynch

Alternatively, for a free earlier version, go to:-

http://www.softwarearchives.com/PDA/

and select PDA and then enter "PocketC" in the search field and click SEARCH, then you will find PocketC:-



Pocket C 6.6.0
PocketC is a PalmPilot-based and WinCE-based C compiler (more...)
License: Shareware SCORE: 5
306
03/29/05

The free version from www.softwarearchives.com/PDA is an older version, and this download site requires free registration.

The free trial from www.orbworks.com is the latest and greatest, having a 45 day free trial before purchase. Also, this version has a desktop and emulator for the PDA, so it is a great package.

The program for horizontal dials is shown below. This is a conversational program and does not use graphics nor a GUI (graphic user interface). This does not use object oriented features, it is as simple as it can be.

NOTE: PocketC uses MATHLIB for trigonometric functions. Many PDAs have this already. If not, then add it into the next HotSynch by double clicking MATHLIB.PRC.

A program is created as a MEMO in the PDA, and PocketC then invoked. When PocketC comes up, it shows a list of compiled programs.

If you have already compiled the program, simply select it and click execute.
If you have not already compiled the program, ignore that list and click compile,
then the entire memo files are searched for C programs, and a list
displayed. Select from that list and click compile. After the very quick compile,
you are returned to the initial screen showing compiled programs.
As you have compiled the program, simply select it and click execute.

The program will run, and you answer the prompts.
When the program ends, click on output to see the output.

The next few pages show programs for compile and execute on the PDA, compile and hot synch from the PC to the PDA as standalone programs, and they have text only, text and graphics, or pure graphics. Also discussed are emulator as well as simulator testing.

Non graphical horizontal dial

Block select then copy this, and then paste it into a new memo file on your Palm desktop, and then perform a hot synch.

This is designed for compile and execute on the PDA using PocketC.

```
// hdial
/*  horizontal dial  */
main()
{           float       fLat, snLat, fLng;
            string      sLat, sLng;
            float       fRef;
            string      sRef;
            float       hlat, hla;
            int         i, ii;
            float       corh, j;

            clear();
            puts ("When finished ");          puts ("click OUTPUT\n");

            sLat=     getsd ("Lat","32");
            puts ("Lat  ");
            fLat = (float)sLat;
            puts(format(fLat,2));             puts ("\n");

            sLng=     getsd ("Lng","108.2");
            puts ("Lng  ");
            fLng = (float)sLng;
            puts(format(fLng,2));             puts (" ...and... ");

            sRef=     getsd ("Ref","105");
            puts ("Ref.Lng  ");
            fRef = (float)sRef;
            puts(format(fRef,2));             puts ("\n");

            snLat=sin(2*3.1416*fLat/360);
            corh = (4*(fLng-fRef)) / 60 ;

            for    (i = -6 ; i <7 ; i++  )
            {
                        // get the hour angle of the sun
                        ii = (-1)*i  ;
                        j  = 15 * (ii+corh) ;

                        /* get the resulting hour line angle */
                        hlat = snLat * tan(2*3.1416*j/360) ;

                        /* get the hour line angle back to degrees */
                        hla  = 360*(atan(hlat))/(2*3.1416) ;

                        puts ("Hour: "); puts ( i);  puts ("  ");
                        puts ("hLineAngle: ");
                        puts (format(hla,2) );
                        puts ("\n");
            }
}
```

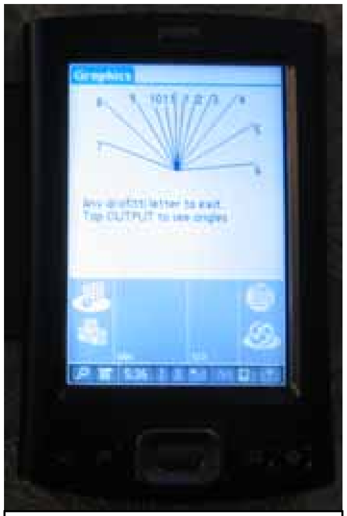
```
* T E S T    V A L U E S
        Design latitude:          32.75
        Design longitude:        108.20
        Reference meridian:      105.00
                Hour:  -6 hour angle:  93.20  Angle: -84.10
                Hour:  -5 hour angle:  78.20  Angle:  68.88
                Hour:  -4 hour angle:  63.20  Angle:  46.96
                Hour:  -3 hour angle:  48.20  Angle:  31.18
                Hour:  -2 hour angle:  33.20  Angle:  19.49
                Hour:  -1 hour angle:  18.20  Angle:  10.09
                Hour:   0 hour angle:   3.20  Angle:   1.73
                Hour:   1 hour angle: -11.80  Angle:  -6.45
                Hour:   2 hour angle: -26.80  Angle: -15.28
                Hour:   3 hour angle: -41.80  Angle: -25.81
                Hour:   4 hour angle: -56.80  Angle: -39.58
                Hour:   5 hour angle: -71.80  Angle: -58.71
                Hour:   6 hour angle: -86.80  Angle: -84.10
```

```
// hdialG
/*  horizontal dial with graphics */
main()
{       float   fLat, snLat, fLng;
        string  sLat, sLng, s;
        float   fRef;
        string  sRef;
        float   hlat, hla;
        int     i, ii, x, y, xc, yc;
        float   corh, j;

        clear();
        graph_on();

        puts ("Horizontal Dial:  ");

        sLat=     getsd ("Lat","32");
        puts ("Lat  ");
        fLat = (float)sLat;
        puts(format(fLat,2));

        sLng=     getsd ("Lng","108.2");
        puts ("\nLng  ");
        fLng = (float)sLng;
        puts(format(fLng,2));

        sRef=     getsd ("Ref","105");
        puts (" Ref.Lng  ");
        fRef = (float)sRef;
        puts(format(fRef,2));
        puts ("\n");

        snLat=sin(2*3.1416*fLat/360);
        corh = (4*(fLng-fRef)) / 60 ;

        for    (i = -6 ; i <7 ; i++  )
        {
                // get the hour angle of the sun
                ii = (-1)*i   ;
                j  = 15 * (ii+corh) ;

                /* get the resulting hour line angle */
                hlat = snLat * tan(2*3.1416*j/360) ;

                /* get the hour line angle back to degrees */
                hla  = 360*(atan(hlat))/(2*3.1416) ;

                xc = 80; yc = 20;
                if ( hla>=-45 && hla<=45) // middle hours
                {       y=60;
                        x=-y*hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        if ( i>0 ) { text(x+xc,100-(y+yc), i); }
                        if ( i<0 ) { text(x+xc,100-(y+yc), 12+i); }
                }
                if ( hla>45 && i<0 )                    // early hours
                {       x=-60;                          // good angles
                        y=-x/hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        text(x+xc,100-(y+yc), 12+i);
                        text(x+xc-3,100-(y+yc-9), format(hla,1));
                }
                if ( hla<-45 && i>0)                    // late hours
                {       x=60;                           // good angles
                        y=-x/hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        text(x+xc,100-(y+yc), i);
                        text(x+xc,100-(y+yc-9), format(hla,1));
                }
                puts ("Hour: "); puts ( i);  puts ("  ");
                puts ("hLineAngle: ");
                puts (format(hla,2) );
                puts ("\n");

                if ( i <0 )
                {       text(10,100-(i*10), 12+i);
                        text(30,100-(i*10), format(hla,1));
                }
                if ( i >0 )
                {       text(80 ,100+(i*10), i);
                        text(100,100+(i*10), format(hla,1));
                }
        }
        text(10,85, "Any grafitti letter to exit");
        text(10,95, "Tap OUTPUT to see angles");
        s=     getc ();
}
```

Graphical horizontal dial

Block select then copy this, and then paste it into a new memo file on your Palm desktop, and then perform a hot synch.

This is designed for compile and execute on the PDA using PocketC.



This program also displays the text of hour line angles on the text output area.

Graphical horizontal dial

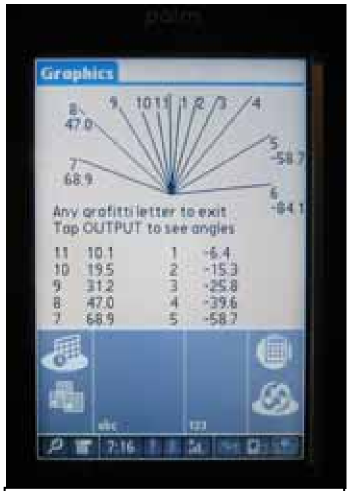
This is compiled with the IDE provided by orbworks, then the program is double clicked

hdialgSA.prc

to get it into the hot synch queue, then hot synched. This is not compiled on the PDA. It is a standalone application.

This is designed for compile on the PC then synching with the PDA as a standalone program.

```
// hdialgSA
@cid "ILS1";
@name "hdialgSA";
@dbname "hdialgSA";
@category "Main";
@ver "1.0";

/*  horizontal dial with graphics */
main()
{       float   fLat, snLat, fLng;
        string  sLat, sLng, s;
        float   fRef;
        string  sRef;
        float   hlat, hla;
        int     i, ii, x, y, xc, yc;
        float   corh, j;

        clear();
        graph_on();

        sLat=     getsd ("Lat","32");
        fLat = (float)sLat;

        sLng=     getsd ("Lng","108.2");
        fLng = (float)sLng;

        sRef=     getsd ("Ref","105");
        fRef = (float)sRef;

        snLat=sin(2*3.1416*fLat/360);
        corh = (4*(fLng-fRef)) / 60  ;

        for     (i = -6  ; i <7  ; i++   )
        {
                // get the hour angle of the sun
                ii = (-1)*i   ;
                j  = 15 * (ii+corh) ;

                /* get the resulting hour line angle */
                hlat = snLat * tan(2*3.1416*j/360) ;

                /* get the hour line angle back to degrees */
                hla  = 360*(atan(hlat))/(2*3.1416) ;

                xc = 80; yc = 20;
                if ( hla>=-45 && hla<=45)     // middle hours
                {       y=60;
                        x=-y*hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        if ( i>0 ) { text(x+xc,100-(y+yc), i); }
                        if ( i<0 ) { text(x+xc,100-(y+yc), 12+i); }
                }
                if ( hla>45 && i<0 )          // early hours
                {       x=-60;                               // good angles
                        y=-x/hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        text(x+xc,100-(y+yc), 12+i);
                        text(x+xc-3,100-(y+yc-9), format(hla,1));
                }
                if ( hla<-45 && i>0)          // late hours
                {       x=60;                                // good angles
                        y=-x/hlat;
                        line (1,xc,100-yc,x+xc,100-(y+yc));
                        text(x+xc,100-(y+yc), i);
                        text(x+xc,100-(y+yc-9), format(hla,1));
                }

                if ( i <=0 )
                {       text(10,100-(i*10), 12+i);
                        text(30,100-(i*10), format(hla,1));
                }
                if ( i >0 )
                {       text(80 ,100+(i*10), i);
                        text(100,100+(i*10), format(hla,1));
                }
        }
        text(10,90, "Any grafitti letter to exit");
        s=     getc ();
}
```



```
Any grafitti letter to exit
Tap OUTPUT to see angles
11   10.1        1   -6.4
10   19.5        2   -15.3
9    31.2        3   -25.8
8    47.0        4   -39.6
7    68.9        5   -58.7
```

This program only displays hour line angles on the graphics area.

Graphical vertical dial

This is compiled with the IDE provided by orbworks, then the program is double clicked

vdialgSA.prc

to get it into the hot synch queue, then hot synched. This is not compiled on the PDA. It is a standalone application.

This is designed for compile on the PC then synching with the PDA as a standalone program.

This exact code can be saved as a memo in the Palm desktop, hot synched, and then compiled and executed on the PDA itself. However, if the standalone version is there also (i.e. as compiled with the PocketC IDE) then the name must be changed from:-

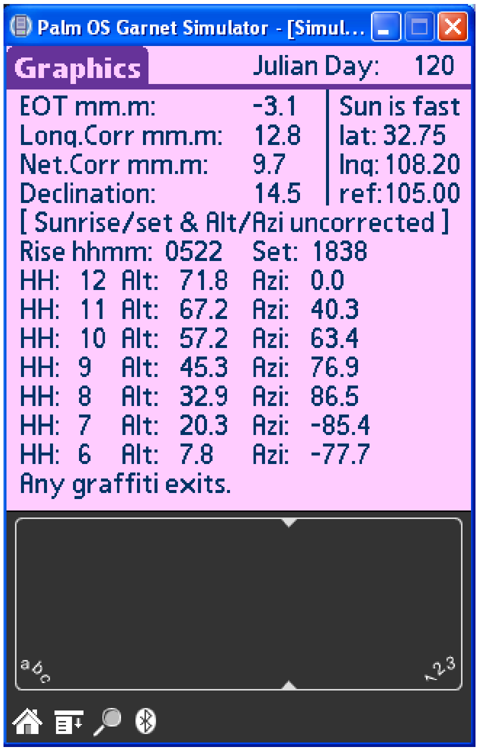
vdialgSA   to, for example:-
             vdialg

```
// vdialgSA
@cid "ILS1";
@name "vdialgSA";
@dbname "vdialgSA";
@category "Main";
@ver "1.0";

/*  vertical dial with graphics */
main()
{       float   fLat, csLat, fLng;
        string  sLat, sLng, s;
        float   fRef;
        string  sRef;
        float   hlat, hla;
        int     i, ii, x, y, xc, yc;
        float   corh, j;

        clear();
        graph_on();

        sLat=     getsd ("Lat","32");
        fLat = (float)sLat;

        sLng=     getsd ("Lng","108.2");
        fLng = (float)sLng;

        sRef=     getsd ("Ref","105");
        fRef = (float)sRef;

        csLat=cos(2*3.1416*fLat/360);
        corh = (4*(fLng-fRef)) / 60 ;

        for     (i = -6 ; i <7 ; i++  )
        {
                // get the hour angle of the sun
                ii = (-1)*i  ;
                j  = 15 * (ii+corh) ;

                /* get the resulting hour line angle */
                hlat = csLat * tan(2*3.1416*j/360) ;

                /* get the hour line angle back to degrees */
                hla  = 360*(atan(hlat))/(2*3.1416) ;

                xc = 80; yc = 20;
                if ( hla>=-45 && hla<=45)      // middle hours
                {       y=60;
                        x=-y*hlat;
                        line (1,xc,yc,x+xc,(y+yc));
                        if ( i>0 ) { text(x+xc,(y+yc), i); }
                        if ( i<0 ) { text(x+xc,(y+yc), 12+i); }
                }
                if ( hla>45 && i<0 )           // early hours
                {       x=-60;                                 // good angles
                        y=-x/hlat;
                        line (1,xc,yc,x+xc,(y+yc));
                        text(x+xc,(y+yc), 12+i);
                        text(x+xc-3,(y+yc-9), format(hla,1));
                }
                if ( hla<-45 && i>0)           // late hours
                {       x=60;                                  // good angles
                        y=-x/hlat;
                        line (1,xc,yc,x+xc,(y+yc));
                        text(x+xc,(y+yc), i);
                        text(x+xc,(y+yc-9), format(hla,1));
                }

                if ( i <=0 )
                {       text(10,100-(i*10), 12+i);
                        text(30,100-(i*10), format(hla,1));
                }
                if ( i >0 )
                {       text(80 ,100+(i*10), i);
                        text(100,100+(i*10), format(hla,1));
                }
        }
        text(10,90, "Any grafitti letter to exit");
        s=     getc ();
}
```

An almanac as a standalone program

```
// almanacSA
@cid "ILS1";
@name "almanacSA";
@dbname "almanacSA";
@category "Main";
@ver "1.0";

#define RADIANS (2*3.1416/360)
#define DEGREES (360/(2*3.1416))

/*  almanac but text is done with graphics */
main()
{       float   fLat, csLat, fLng;
        string  sLat, sLng, s;
        float   fRef;
        string  sRef;
        float   hlat, hla, sha;
        int     i, ii, x, y, xc, yc, mm, dd, jd, amh, shh, shm;
        float   corh, corm, decl, j, jdf, eot, alt, azi, ss;

        clear();
        graph_on();

        // start coordinates for printing to the PDA
        x = 5;
        y = 5;
        text (x,y+50, "Ensure MATHLIB.PRC synchd");

        mm =     (int)getsd ("Month 1-12","1");
        dd =     (int)getsd ("Day 1-31","1");
        if (mm==1) {   jd = 0+dd; }
        if (mm==2) {   jd = 31+dd; }
        if (mm==3) {   jd = 31+28+dd; }
        if (mm==4) {   jd = 31+28+31+dd; }
        if (mm==5) {   jd = 31+28+31+30+dd; }
        if (mm==6) {   jd = 31+28+31+30+31+dd; }
        if (mm==7) {   jd = 31+28+31+30+31+30+dd; }
        if (mm==8) {   jd = 31+28+31+30+31+30+31+dd; }
        if (mm==9) {   jd = 31+28+31+30+31+30+31+31+dd; }
        if (mm==10){    jd = 31+28+31+30+31+30+31+31+30+dd; }
        if (mm==11){    jd = 31+28+31+30+31+30+31+31+30+31+dd; }
        if (mm==12){    jd = 31+28+31+30+31+30+31+31+30+31+30+dd; }
        jdf = (float)jd;

        sLat=     getsd ("Lat","32.75");
        fLat = (float)sLat;

        sLng=     getsd ("Lng","108.2");
        fLng = (float)sLng;

        sRef=     getsd ("Ref","105");
        fRef = (float)sRef;

        csLat=cos(2*3.1416*fLat/360);
        corm = 4*(fLng-fRef) ;
        corh = corm / 60 ;

        eot=-1*(9.84*sin(RADIANS*(2*(360*(jdf-81)/365)))-7.53*cos(RADIANS*(360*(jdf-
81)/365))-1.5*sin(RADIANS*(360*(jdf-81)/365)))-0.3;

        // start coordinates for printing to the PDA
        x = 5;
        y = 15;

        // display EOT
        text (x,y, "EOT mm.m:  ");
        text (x+80,y, format(eot,1));
        //text (x+100,y, mm);
        //text (x+120,y, dd);
        text (x+120,y, "JD:");
        text (x+135,y, jd);
        y=y+10; // next line

        // display eot correction
        text (x,y, "Long.Corr mm.m:");
        text (x+80,y, format(corm,1));
        y=y+10; // next line
        // display net correction
        text (x,y, "Net.Corr mm.m:");
        text (x+80,y, format(corm+eot,1));
        y=y+10; // next line

        // display declination
        decl = DEGREES*(0.006918 - 0.399912*cos(2*3.1416*(jdf-1)/365) +
0.070257*sin(2*3.1416*(jdf-1)/365)  - 0.006758*cos(2*2*3.1416*(jdf-1)/365) +
0.000907*sin(2*2*3.1416*(jdf-1)/365)  - 0.002697*cos(3*2*3.1416*(jdf-1)/365) +
0.00148*sin(3*2*3.1416*(jdf-1)/365));
        text (x,y, "Declination:");
        text (x+80,y, format(decl,1));
        y=y+10; // next line

        // display rise/set from noon uncorrected
        sha = 12-((DEGREES*(acos( tan(RADIANS*fLat) * tan(RADIANS*decl) )))/15) ;
        shh = (int)(sha);
        shm = (sha-(int)(sha))*60;
        text(x,y, "[Sunrise/set & Alt/Azi uncorrected]");
        y=y+10; // next line
        text (x,y, "Rise/Set from noon hh mm:");
        text (x+120,y, shh);
        text (x+130,y, "+");
        text (x+135,y, shm);
        y=y+10; // next line

        for    (i = 6  ; i <13 ; i++  )
        {
                amh = i;                      // am hour 6 to 12
                alt =
DEGREES*(asin(sin(RADIANS*(decl))*sin(RADIANS*fLat)+cos(RADIANS*decl)*cos(RADIANS*fLat)*c
os(RADIANS*(15*(12-(amh))))));
                azi = DEGREES*(atan(sin(RADIANS*(15*(12-
amh)))/(sin(RADIANS*fLat)*cos(RADIANS*(15*(12-amh)))-
tan(RADIANS*decl)*cos(RADIANS*fLat))));
                text (x,y, "HH:");
                text (x+20,y, amh);
                text (x+35,y, "Alt:");
                text (x+55,y, format(alt,1));
                text (x+80,y, "Azi:");
                text (x+100,y, format(azi,1));
                y=y+10; // next line
        }
        text(x,y, "Any graffiti exits.");
        s=     getc ();
}
```

Month:	4
Day:	30
Latitude:	32.75
Longitude:	108.2
Ref. Long:	105



```
Julian Day:    120
EOT mm.m:          -3.1  | Sun is fast
Long.Corr mm.m:    12.8  | lat: 32.75
Net.Corr mm.m:     9.7   | lng: 108.20
Declination:       14.5  | ref:105.00
[ Sunrise/set & Alt/Azi uncorrected ]
Rise hhmm:  0522   Set:  1838
HH:  12  Alt:  71.8   Azi:  0.0
HH:  11  Alt:  67.2   Azi:  40.3
HH:  10  Alt:  57.2   Azi:  63.4
HH:  9   Alt:  45.3   Azi:  76.9
HH:  8   Alt:  32.9   Azi:  86.5
HH:  7   Alt:  20.3   Azi:  -85.4
HH:  6   Alt:  7.8    Azi:  -77.7
Any graffiti exits.
```

NOTE: An optional graphics box can be added around the dial plate, this entails four lines at the end which are the box, and for aesthetic purposes, the "Any grafitti..." text is moved over a bit.

```
        ...
        ...
        ...
        ...
        line (1,5,15,159,15);
        line (1,5,90,159,90);
        line (1,5,15,5,90);
        line (1,159,15,159,90);

        text(30,90, "Any grafitti letter to exit");
        s=     getc ();
}
```

NOTE: The zip file that goes with this document has several files. The convention is...

xxxxx.prc — a standalone application compiled by the PocketC Integrated Development Environment (IDE) which is available at:-

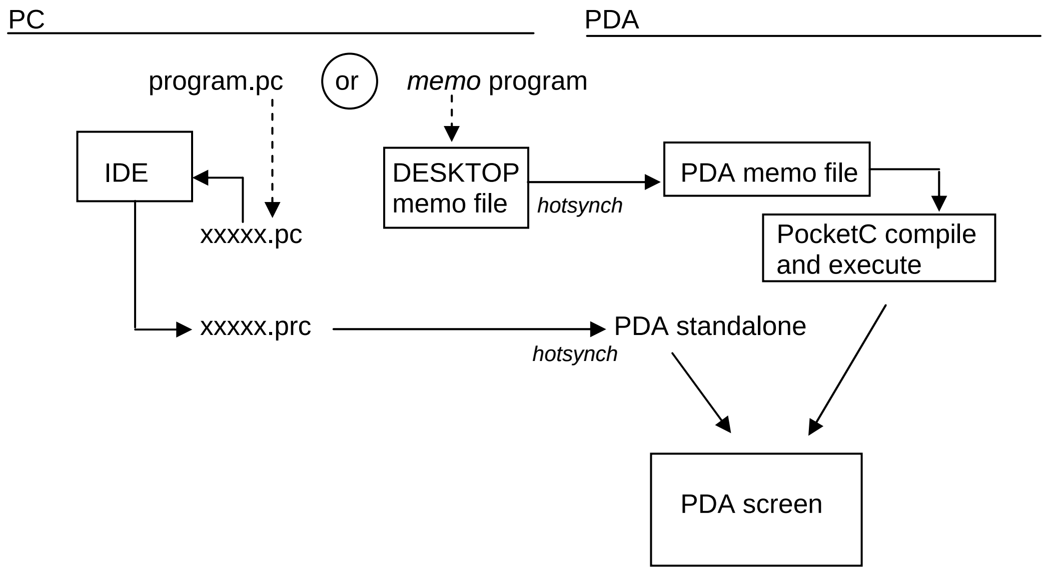
http://www.orbworks.com/pcpalm/download.html
http://orbworks.com/pcpalm/buy.html

xxxxx.pc — the source code for the IDE which will turn it into a xxxxx.prc

xxxxx.c — code to be saved in a Palm Pilot *memo* file on the Palm Desktop and then hot synched. This requires PocketC be installed on the PDA, and that this program be compiled and then executed within PocketC.

xxxxxG.yy — This has graphics output

xxxxxGSA.yy — This has graphics output, and is a standalone program built by the IDE.



TESTING ~ no special software except PocketC on the PDA ~ or ! PocketC's IDE on your PC

You have two choices.

SIMPLEST

Install PocketC by downloading it, unzipping it, and then installing PocketC on the PDA.

Then copy the programs xxxxx.c to the MEMO area on the Palm Desktop, and perform a hot synch.

Then on the PDA, execute the PocketC application, enter COMPILE, let it locate the programs (it looks for "// name" at the start of each memo.

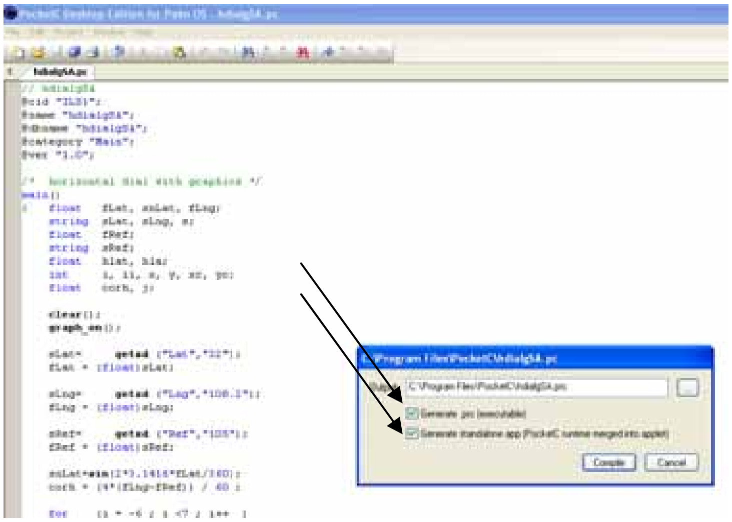
Then compile that program, then execute it.

If the hour line angles are zero or the hour lines drawn are all vertical, then double click on MATHLIB.PRC and hot synch it. Then repeat teh process.

NEXT SIMPLEST

Use the PocketC Integrated Development Environment (IDE) and load the xxxxx.pc programs into the IDE and compile them.

NOTE: After the source is loaded, go to PROJECT, SET PROJECT FILE, and click on the name of your program. Then do the build.


PocketC Desktop Edition for Palm OS - hdialg5A.pc
File Edit Project Window Help
hdialg5A.pc

```
// hdialg5A
@cid "ILD1";
@name "hdialg5A";
@dbname "hdialg5A";
@category "Main";
@ver "1.0";

/*  horizontal dial with graphics */
main()
{   float    fLat, snLat, fLng;
    string   sLat, sLng, s;
    float    fRef;
    string   sRef;
    float    hlat, hla;
    int      i, ii, x, y, xc, yc;
    float    corh, j;

    clear();
    graph_on();

    sLat=     getsd ("Lat","32");
    fLat = (float)sLat;

    sLng=     getsd ("Lng","108.2");
    fLng = (float)sLng;

    sRef=     getsd ("Ref","105");
    fRef = (float)sRef;

    snLat=sin(2*3.1416*fLat/360);
    corh = (4*(fLng-fRef)) / 60 ;

    for     (i = -6 ; i <7 ; i++  )
```

C:\Program Files\PocketC\hdialg5A.pc

Output: C:\Program Files\PocketC\hdialg5A.prc

☑ Generate .prc (executable)

☑ Generate standalone app (PocketC runtime merged into applet)

Use the two options to ensure the fullest software goes with the compiled program.

The resulting xxxxx.prc is complete and is hot synched to the PDA where it can be run as any application.

That same program can be moved to the emulator or to the simulator PDA depictions and emulated or simulated.

Where is that xxxxx.prc program to be found? Go to

PROGRAM FILES
POCKETC

and locate that program.

TESTING ~ using a PDA simulator (not emulator) (easiest to setup, has ROMS already)

Hints on testing:- http://www.pdacortex.com/palm_emulator.htm

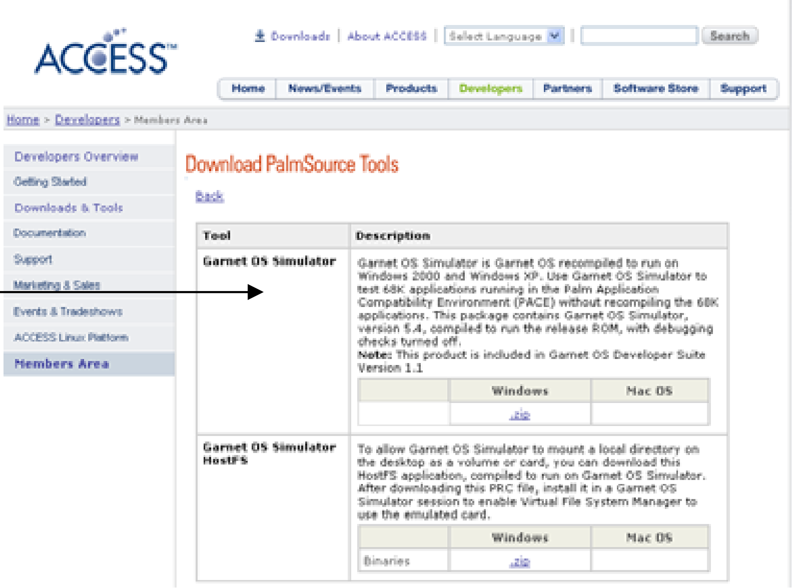
Test harnesses:- http://www.access-company.com/developers/

▼

http://www.access-company.com/products/downloads/index.html

▼

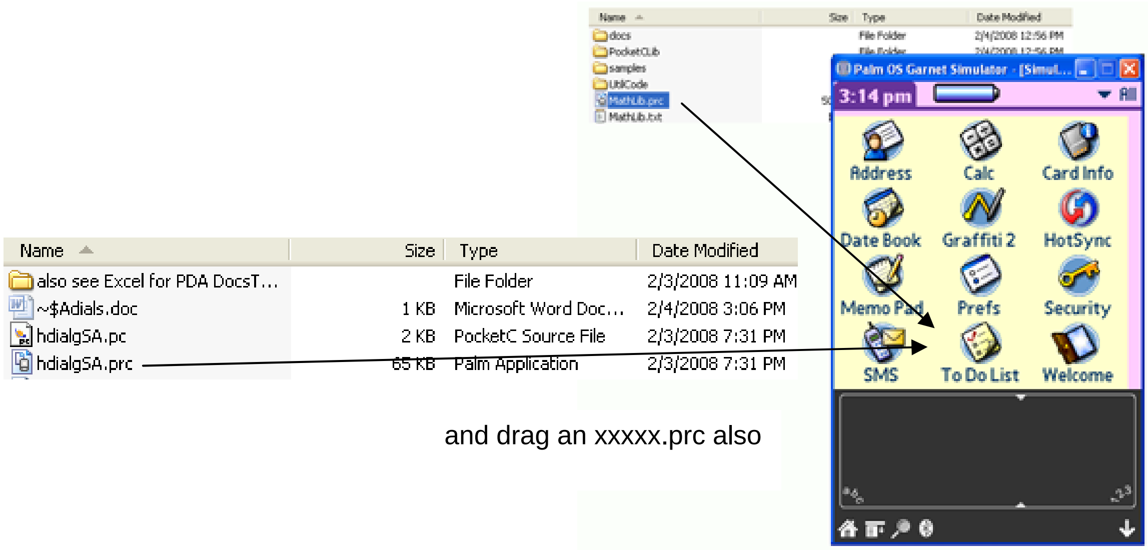
then login (you register for free)
then select Garnet OS Development Tools
then select SIMULATOR (not emulator, etc)
NOTE: you have to resign in sometimes


ACCESS
Downloads | About ACCESS | Select Language | Search
Home | News/Events | Products | Developers | Partners | Software Store | Support
Home > Developers > Members Area
Developers Overview
Getting Started
Downloads & Tools
Documentation
Support
Marketing & Sales
Events & Tradeshows
ACCESS Linux Platform
Members Area
Back

Download PalmSource Tools

Tool	Description
Garnet OS Simulator	Garnet OS Simulator is Garnet OS recompiled to run on Windows 2000 and Windows XP. Use Garnet OS Simulator to test 68K applications running in the Palm Application Compatibility Environment (PACE) without recompiling the 68K applications. This package contains Garnet OS Simulator, version 5.4, compiled to run the release ROM, with debugging checks turned off. **Note:** This product is included in Garnet OS Developer Suite Version 1.1 Windows: .zip Mac OS:
Garnet OS Simulator HostFS	To allow Garnet OS Simulator to mount a local directory on the desktop as a volume or card, you can download this HostFS application, compiled to run on Garnet OS Simulator. After downloading this PRC file, install it in a Garnet OS Simulator session to enable Virtual File System Manager to use the emulated card. Binaries – Windows: .zip Mac OS:

then download the simulator, and unzip it, and
then invoke it, and follow its startup like any new PDA
then drag MATHLIB.PRC



Name	Size	Type	Date Modified
docs		File Folder	2/4/2008 12:56 PM
PocketCLib		File Folder	2/4/2008 12:56 PM
samples			
UtilCode			
MathLib.prc			
MathLib.txt			

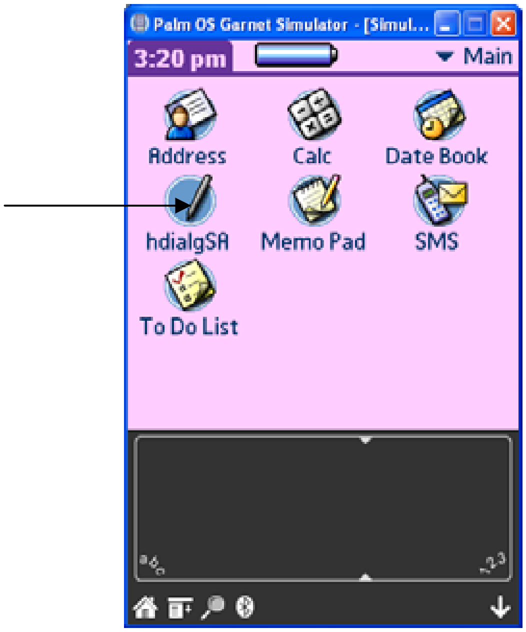
Name	Size	Type	Date Modified
also see Excel for PDA DocsT…		File Folder	2/3/2008 11:09 AM
~$Adials.doc	1 KB	Microsoft Word Doc…	2/4/2008 3:06 PM
hdialgSA.pc	2 KB	PocketC Source File	2/3/2008 7:31 PM
hdialgSA.prc	65 KB	Palm Application	2/3/2008 7:31 PM

and drag an xxxxx.prc also

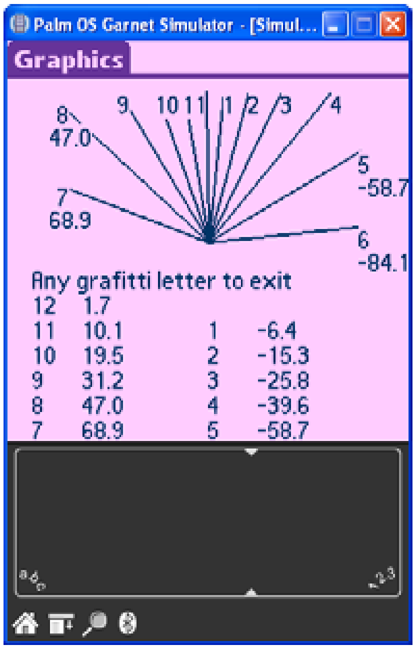
May 18, 2009 ~ this may be distributed freely provided the web site credit
and this notice are retained

if you don't drag MATHLIB.PRC then all trigonometric functions return zero, and thus hour line angles, and hour lines on the graph area will be vertical lines.

then locate the program in the application area of the simulated PDA and run the program



and see the results



```
Any grafitti letter to exit
12   1.7
11   10.1          1    -6.4
10   19.5          2    -15.3
9    31.2          3    -25.8
8    47.0          4    -39.6
7    68.9          5    -58.7
```

TESTING PROGRAMS ~ using a PDA emulator (not simulator) (harder to setup, needs ROMs)

Hints on testing:- http://www.pdacortex.com/palm_emulator.htm

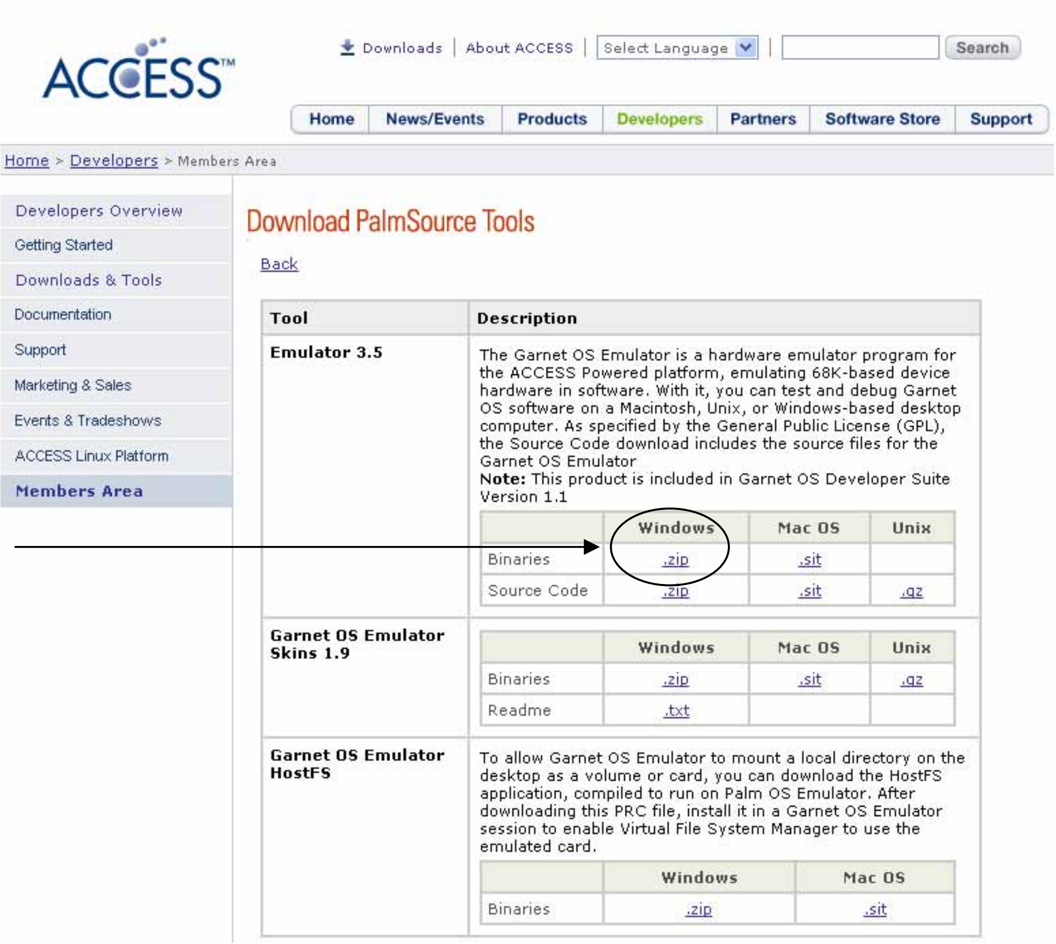
Test harnesses:- http://www.access-company.com/developers/

▼

http://www.access-company.com/products/downloads/index.html

▼

then login (you register for free)
then select Garnet OS Development Tools
then select EMULATOR (not simulator, etc)
NOTE: you have to resign in sometimes



Download PalmSource Tools

Tool	Description
Emulator 3.5	The Garnet OS Emulator is a hardware emulator program for the ACCESS Powered platform, emulating 68K-based device hardware in software. With it, you can test and debug Garnet OS software on a Macintosh, Unix, or Windows-based desktop computer. As specified by the General Public License (GPL), the Source Code download includes the source files for the Garnet OS Emulator **Note:** This product is included in Garnet OS Developer Suite Version 1.1

	Windows	Mac OS	Unix
Binaries	.zip	.sit	
Source Code	.zip	.sit	.gz

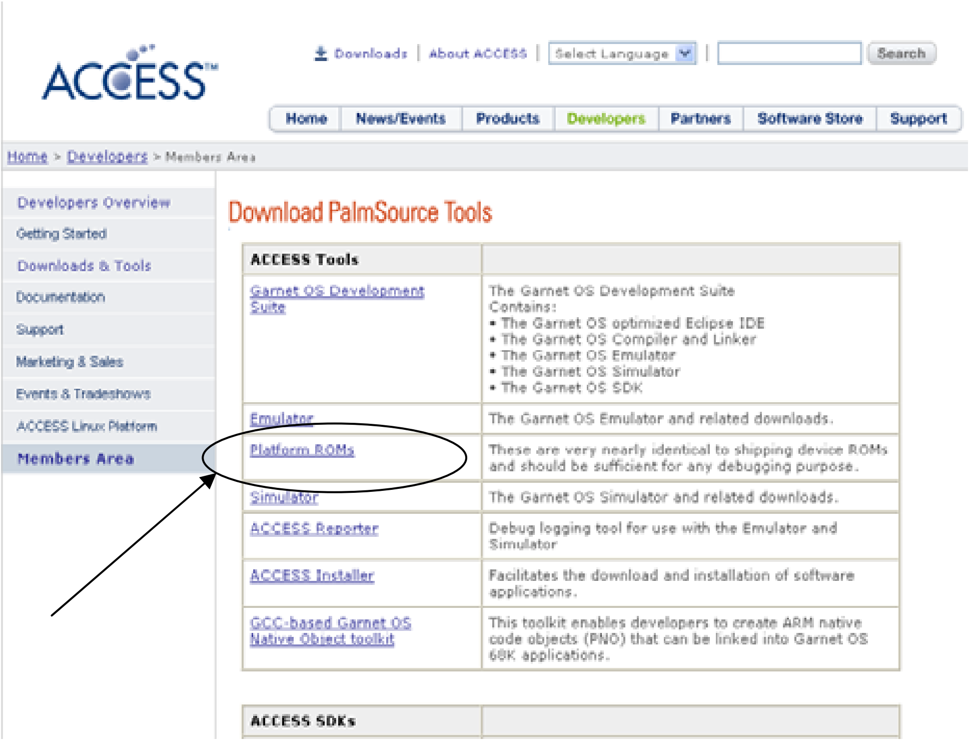
Garnet OS Emulator Skins 1.9

	Windows	Mac OS	Unix
Binaries	.zip	.sit	.gz
Readme	.txt		

Tool	Description
Garnet OS Emulator HostFS	To allow Garnet OS Emulator to mount a local directory on the desktop as a volume or card, you can download the HostFS application, compiled to run on Palm OS Emulator. After downloading this PRC file, install it in a Garnet OS Emulator session to enable Virtual File System Manager to use the emulated card.

	Windows	Mac OS
Binaries	.zip	.sit

then download the emulator, but you will also need a ROM file, I had trouble downloading it from my Palm TX, however, you can find ROMs at:-

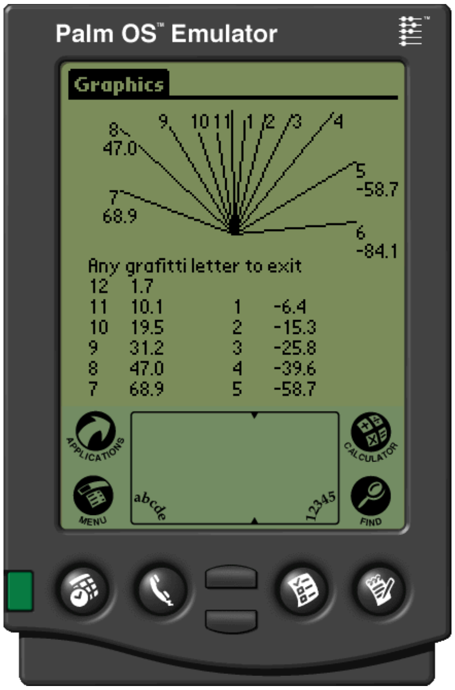

ACCESS
Downloads | About ACCESS | Select Language | Search
Home | News/Events | Products | Developers | Partners | Software Store | Support
Home > Developers > Members Area
Developers Overview
Getting Started
Downloads & Tools
Documentation
Support
Marketing & Sales
Events & Tradeshows
ACCESS Linux Platform
Members Area

Download PalmSource Tools

ACCESS Tools	
Garnet OS Development Suite	The Garnet OS Development Suite Contains: • The Garnet OS optimized Eclipse IDE • The Garnet OS Compiler and Linker • The Garnet OS Emulator • The Garnet OS Simulator • The Garnet OS SDK
Emulator	The Garnet OS Emulator and related downloads.
Platform ROMs	These are very nearly identical to shipping device ROMs and should be sufficient for any debugging purpose.
Simulator	The Garnet OS Simulator and related downloads.
ACCESS Reporter	Debug logging tool for use with the Emulator and Simulator
ACCESS Installer	Facilitates the download and installation of software applications.
GCC-based Garnet OS Native Object toolkit	This toolkit enables developers to create ARM native code objects (PNO) that can be linked into Garnet OS 68K applications.

ACCESS SDKs	

then download and unzip the roms, then invoke the emulator
then locate the ROM (pick one) when asked
then drag MATHLIB.PRC from a folder to the emulator
then drag xxxxx.PRC also, then run it.



```
Any grafitti letter to exit
12    1.7
11    10.1        1    -6.4
10    19.5        2    -15.3
9     31.2        3    -25.8
8     47.0        4    -39.6
7     68.9        5    -58.7
```

WINDOWS VISTA, WIN64 AND HOT SYNCH

Palm Pilot TX

Windows Vista win64 will not hot synch with the USB nor with WIFI. It will synch with Bluetooth.

I bought a $15 Bluetooth USB from Walmart and it installed first time. And I turned WIFI off on the PDA and Bluetooth on.

The PC needed restarting after the Bluetooth driver install, and I did it again after Bluetooth setup just in case.

I configured Hotsynch to use the Bluetooth (COM3 in my case).

Several Hotsynch attempts failed. I rebooted my PDA (warm start) after a serial port in use error on the Palm.

Eventually after much random trial and error things happened.

After several synch atrtempts on my PDA, Vista win64 suddenly asked for a code (like WIFI), I have mine something like:

d2d3a1a2b0

and suddenly the PDA asked for a code, so I entered:

d2d3a1a2b0   (same as above)

And Vista asked for permission to proceed, then the hotsync worked.

I am still noit sure how or why this finally worked, there are many mysteries in this universe, and PDAs and VISTA is one of them.

NOTE: You not need the blue light to be on for the Bluetooth machine to be listening, so just because there is no blue light does not mean that “it needs turning on”.

NOTE: Of course, I expect it would work quicker if I read the booklets, but I didn't.