FINDING GPS COORDINATES ON A MAP USING PDA

Nicolae-Iulian ENESCU[1]

PhD Candidate, University Assistant, Computer Systems and Communications Department,
Faculty of Automation, Computers and Electronics
University of Craiova, Craiova, Romania

E-mail: nenescu@cs.ucv.ro

Abstract: *The purpose is defining algorithms for the elaboration of GPS coordinates using the PDA. The C++ and C# programming languages have been used for this. A software was elaborated which: runs on PDAs that have Windows Mobile 2003 OS, assumes GPS information and implements algorithms for establishing the coordinates on a map represented as a TIF image. The precision for the coordinates is less than 10 meter. The program was tested and used by a Japanese Company.*

Key words: PDA; GPS; coordinates; map; algorithms

Requests

A software application was desired that should run on PDA with GPS antenna attached. The application must locate the GPS coordinates on a map from Japan.

The coordinates known by longitude and latitude must be established over a given image in TIF format which can represent a map with scale 1:200000 or 1:25000. The maps contain information in their headers that refer to the map limits. This data includes the left-top and right-bottom coordinates of the corners, the dimensions of width and height in pixels and the lengths of the maps. It was necessary an algorithm that would also allow map synchronization when the user goes beyond the limits of the current map.

Informatics solution

As a result of many PDAs (on the Japanese market) having Windows Mobile OS, the informatics solution for implementing the requests was to use .NET CF C# with Microsoft Visual Studio .NET 2003. Due to .NET CF having some limitations in loading a TIF image, a dynamic library for the module that loads and saves a TIF image was built in C++ with Microsoft eMbedded Visual C++ 4.

Because the program needed additional functions that are not in .NET CF there was used an application framework named "Smart Device Framework 1.2" which enriches and extends the .NET Compact Framework. There are many new class libraries and controls, along with all the existing class libraries and controls, available free on www.opwnnetcf.org.

Algorithms

Map algorithms

Algorithm for Map Index

200000 and 25000 maps structure is *mesh type*. It is called "*Standard Area Mesh*". This is the standard way to divide an area by constant intervals of longitude and latitude.

In the case of Japan, the latitude and longitude used since 2002 are WGS84 type, but the map data stored in the application map is Tokyo Datum type. The "*Standard Area Mesh*" has 3 levels.

1) <u>The 1st Area Block (Primary Area Division or the ***First Mesh***)</u>

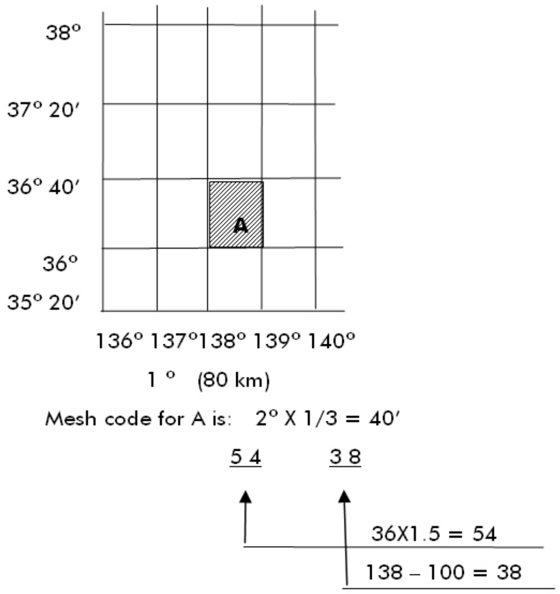
The Primary Area Division is made by dividing Japan into rectangles with the width of 1-degree longitude and height of 2/3-degree latitude (1° X 40').

The 1st Area Block has a correspondent area of 6,400km^2, which is the area of 1 unit from 1:200000 map. Each unit has a correspondent 1:200000 map image.

The mesh code for the 1st area is 4 digits long. The value is composed from 2 parts of 2 digits each.

The first 2 digits are obtained by multiplying the south edge latitude degrees with 1.5.

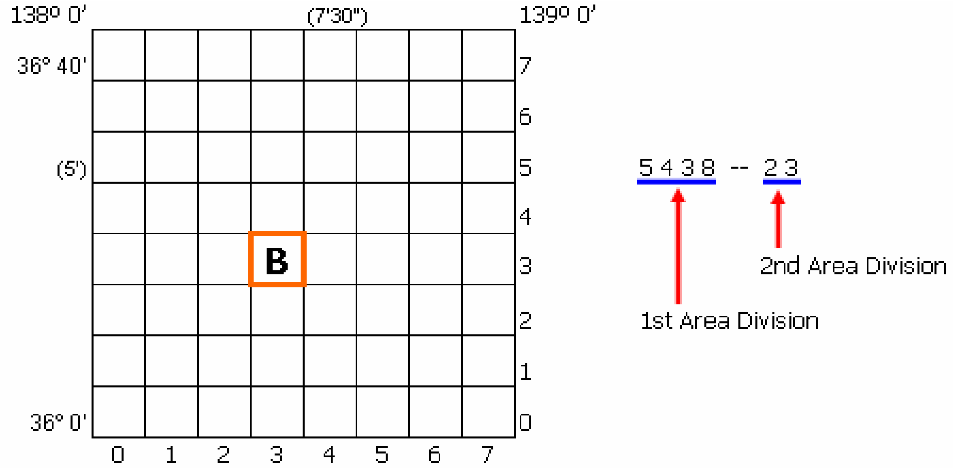
The next 2 digits are obtained by subtracting the west edge longitude degrees with 100.



2) <u>The 2nd Area Block (Secondary Area Division or the ***Second Mesh***)</u>

Dividing the 1st area division vertically – horizontally by 8 makes the 2nd area. Each unit is 7' 30" X 5' and corresponds to 1 unit of the 1:25000 map, which has an area of 100 km^2.

The mesh code for the 2nd area division is a 2-digit code. The first digit is numbered from 0 for south edge to 7 for north, along longitude line. The second digit is numbered from 0 to west edge to 7 for east edge, along latitude line.

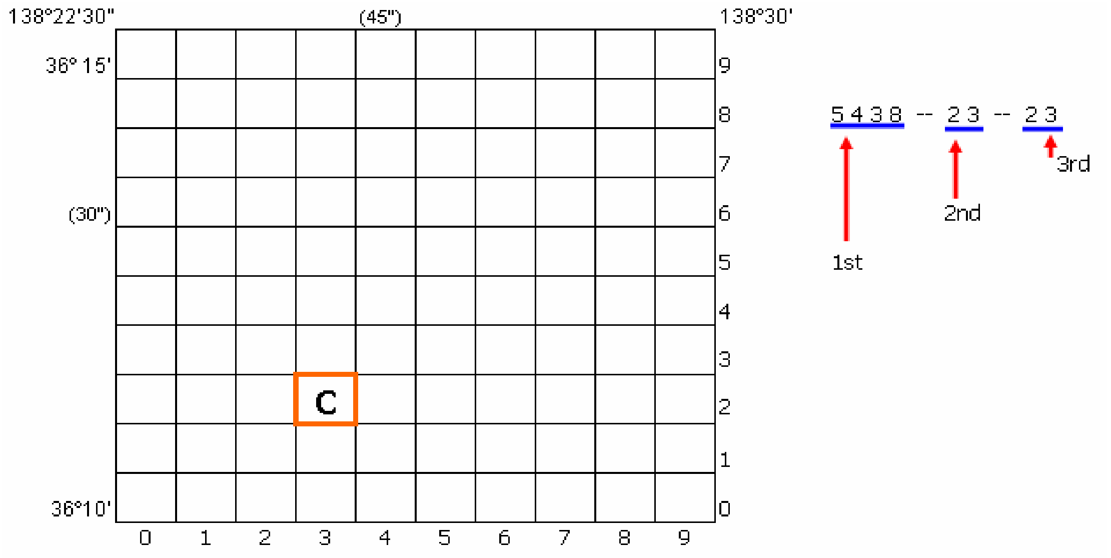


3) The 3rd Area Block (Third Area Division or the ***Base Mesh***)

Dividing the 2nd area division vertically – horizontally by 10 makes the 3rd area. Each unit is 45" X 30" and it corresponds to an area of 1 km^2.

The mesh code for the 3rd area division is a 2 digits code. The first digit is numbered from 0 for south edge to 9 for north edge, along longitude line. The second digit is numbered from 0 for west edge to 9 for east edge, along latitude line.

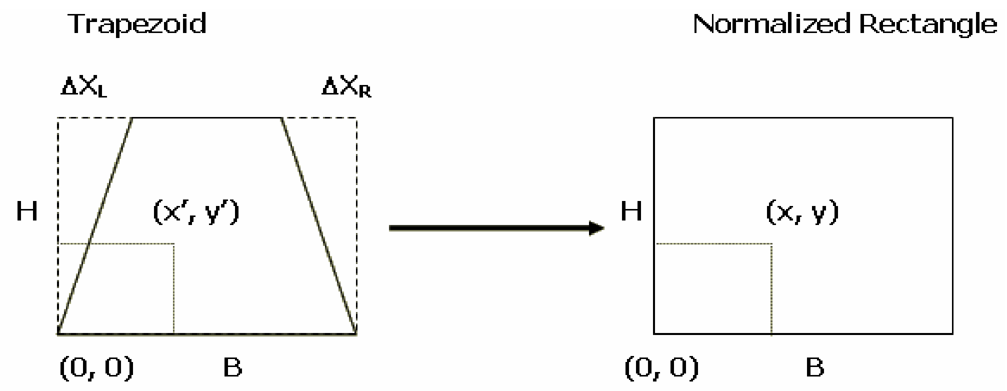
The 3rd Area Division is called the "Base Mesh".



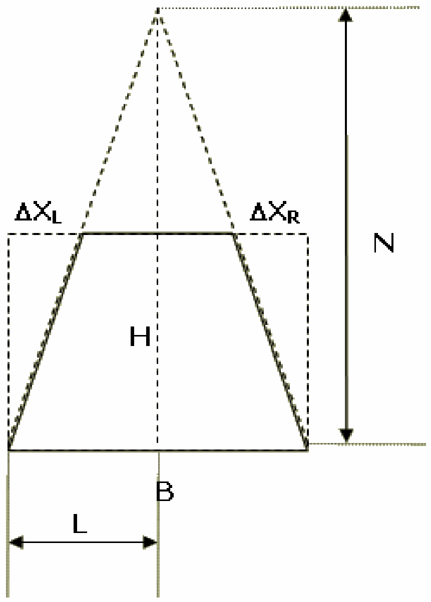
Algorithm for map management

The major concern is that there are irregular maps and the maps have a trapezoidal shape instead of a rectangular one as we might expect. The first step towards a good map joining quality is to adjust the maps' shape.

The program must cut images exactly on the edges of the map. The problem was that the images are not rectangles, and the information about the corners is not accurate. In order to solve those problems the application must find the real map corners and must transform the maps in rectangles using affine transformation.

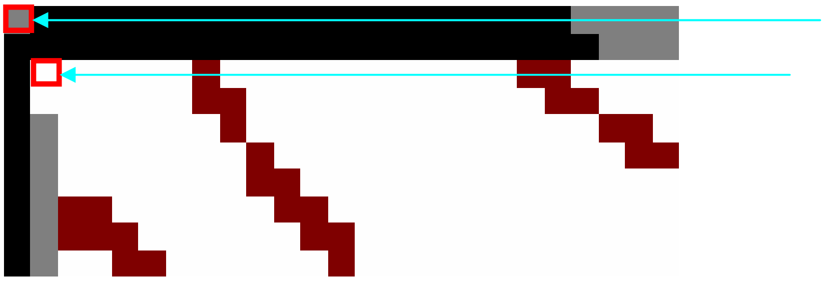


(x', y') = (L – (N – y) * (L – x) / N, y)



$N = (H * B) / (\Delta X_L + \Delta X_R)$
$L = (\Delta X_L * B) / (\Delta X_L + \Delta X_R)$

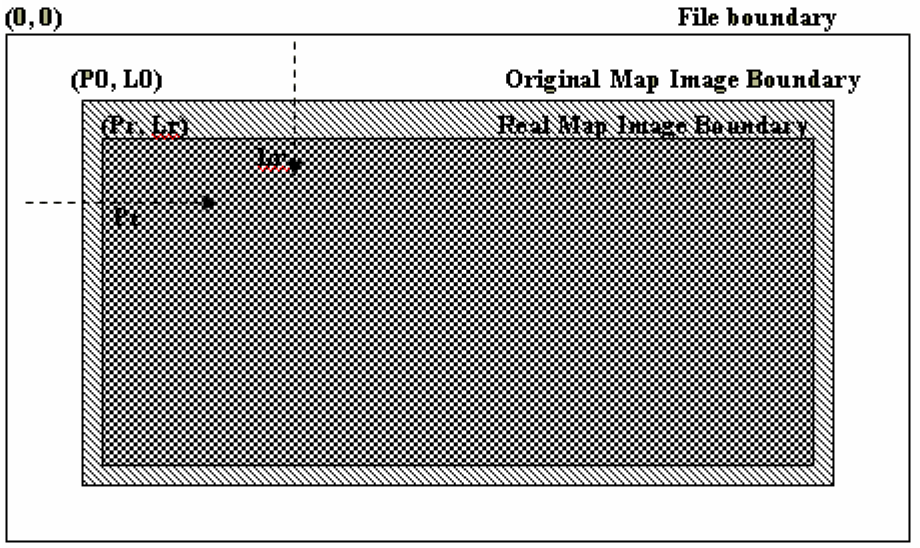
Real corners



In theory, every map image

should have a 2-pixel width boundary, gray colored. Unfortunately this is not always true. There are many cases when the boundary line is missing, or is only 1-pixel width or the color is not gray.

The program searches the real corner by checking the color of pixels.



The two arrows show the direction of searching. Every pixel, starting from the file boundary, is checked to see if it has a color different than Black (text color), Brown (heights), White (none).

There are 2 directions: one for finding the "X" coordinate and the other for finding the "Y" coordinate. The same procedure is used for all the 4 corners.

There are a couple of conditions to be met in order to decide if a pixel contributes to the real corner:

- the search is done separately on the line and on the column;
- if a pixel has the Gray color we assume that it is on the map border;
- if a pixel has a different color than Gray, Black and White, we assume that the border is missing and that the pixel is where the map starts. The next 10 (10 is a number experimentally issued) pixels' color is checked to be sure they are image pixels.
- if the distance between the default value and the found value is bigger than 15 pixels, than the default value is kept.

GPS Algorithms

This section provides guidance in the implementation of measurement processing algorithms.

The discussions include:

- Mathematical constants used in GPS position determination computations.
- The GPS parity algorithm implementation to permit the user to detect demodulation errors within the decoded navigation message.
- Interpretation of the satellite transmitting URA parameter.
- Satellite position determination using broadcast ephemeris parameters.
- Correction of the code phase time received from the satellite with respect to both satellite code phase offset and relativistic effects.

- Compensation for the effects of satellite group delay differential.
- Correction for ionospheric propagation delay.
- Performing time transfer to UTC.

Mathematical Constants

The speed of light used for generating the data described in the above paragraphs is:

$$c = 2.99792458 \times 10^{8} \text{ meters per second}$$

which is the official WGS-84 speed of light. The user should use the same value for the speed of light in computations. Other WGS-84 constants the user is required to use for satellite ephemeris calculations are:

$\mu = 3.986005 \times 10^{14}$ meters3 / sec2 WGS-84 value of the Earth's universal gravitational parameter

$\Omega e = 7.2921151467 \times 10^{-5}$ rad / sec WGS-84 value of the Earth's rotation rate

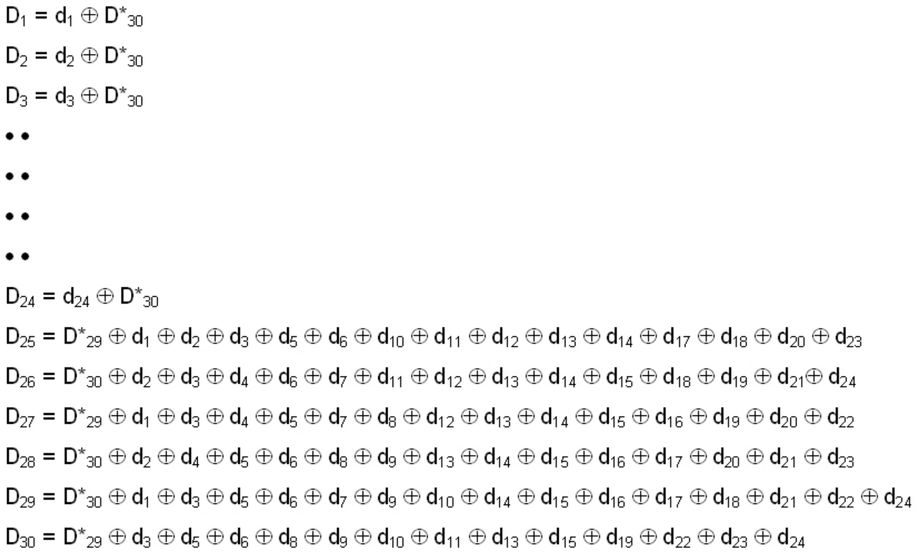
The sensitivity of the satellite's antenna phase center position to small perturbations in most ephemeris parameters is extreme. The sensitivity of position to the parameters $(A)^{1/2}$, Crc and Crs is about one meter/meter. The sensitivity of position to the angular rate parameters is on the order of 10^{8} meters/semicircle, and to the angular rate parameters is on the order of 10^{12} meter/semicircle/second. Because of this extreme sensitivity to angular perturbations, the value of π used in the curve fit is given here. π is a mathematical constant, the ratio of a circle's circumference to its diameter. Here π is taken as

$$\pi = 3.1415926535898$$

Parity Algorithm

The user must perform error detection of the decoded navigation data using the parity algorithm equations provided in Table 1. Figure 1 presents an example flow chart that defines one way of recovering data (d_n) and checking parity. The parity bit D^*_{30} is used for recovering raw data. The parity bits D^*_{29} and D^*_{30}, along with the recovered raw data (d_n) are modulo-2 added in accordance with the equations appearing in Table 1 for $D_{25} \ldots D_{30}$, which provide computed parity to compare with transmitted parity $D_{25} \ldots D_{30}$.

Table 1. Parity Encoding Equations



$D_1 = d_1 \oplus D^*_{30}$

$D_2 = d_2 \oplus D^*_{30}$

$D_3 = d_3 \oplus D^*_{30}$

• •

• •

• •

• •

$D_{24} = d_{24} \oplus D^*_{30}$

$D_{25} = D^*_{29} \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_{10} \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{17} \oplus d_{18} \oplus d_{20} \oplus d_{23}$

$D_{26} = D^*_{30} \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{18} \oplus d_{19} \oplus d_{21} \oplus d_{24}$

$D_{27} = D^*_{29} \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_5 \oplus d_7 \oplus d_8 \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{19} \oplus d_{20} \oplus d_{22}$

$D_{28} = D^*_{30} \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{20} \oplus d_{21} \oplus d_{23}$

$D_{29} = D^*_{30} \oplus d_1 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_9 \oplus d_{10} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{18} \oplus d_{21} \oplus d_{22} \oplus d_{24}$

$D_{30} = D^*_{29} \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{10} \oplus d_{11} \oplus d_{13} \oplus d_{15} \oplus d_{19} \oplus d_{22} \oplus d_{23} \oplus d_{24}$

where:
$d_1, d_2, ... d_{24}$ are the source data bits
the symbol (*) is used to identify the last 2 bits of the previous word of the subframe,
$D_{25}, ... D_{30}$ are the computed parity bits
$D_1, D_2, D_3, ... D_{29}, D_{30}$ are the bits transmitted by the satellite, and
$\oplus$ is the "Modulo-2" or "Exclusive-Or" operation.

User Range Accuracy (URA)

The URA reported in the navigation message will correspond to the maximum value anticipated during each subframe fit interval with uniform SA levels invoked. Referring to the decimal equivalent of the transmitted four-bit binary number as N -- with N a positive integer in the range of 0 through 15 -- the accuracy value is defined to mean "no better than X meters", in accordance with the following relationships:

- If the value of N is 6 or less, $X = 2^{(1 + N/2)}$,
- If the value of N is 6 or more, but less than 15, $X = 2^{(N-2)}$,
- N = 15 will indicate the absence of an accuracy prediction and will advise the SPS user to use that satellite at the user's own risk.

For N = 1, 3, and 5, X is rounded to 2.8, 5.7, and 11.3 meters respectively; the above relationships yield integer values of X for all other values of N. Using these values of X the user may utilize a look-up table approach for interpreting the URA message.

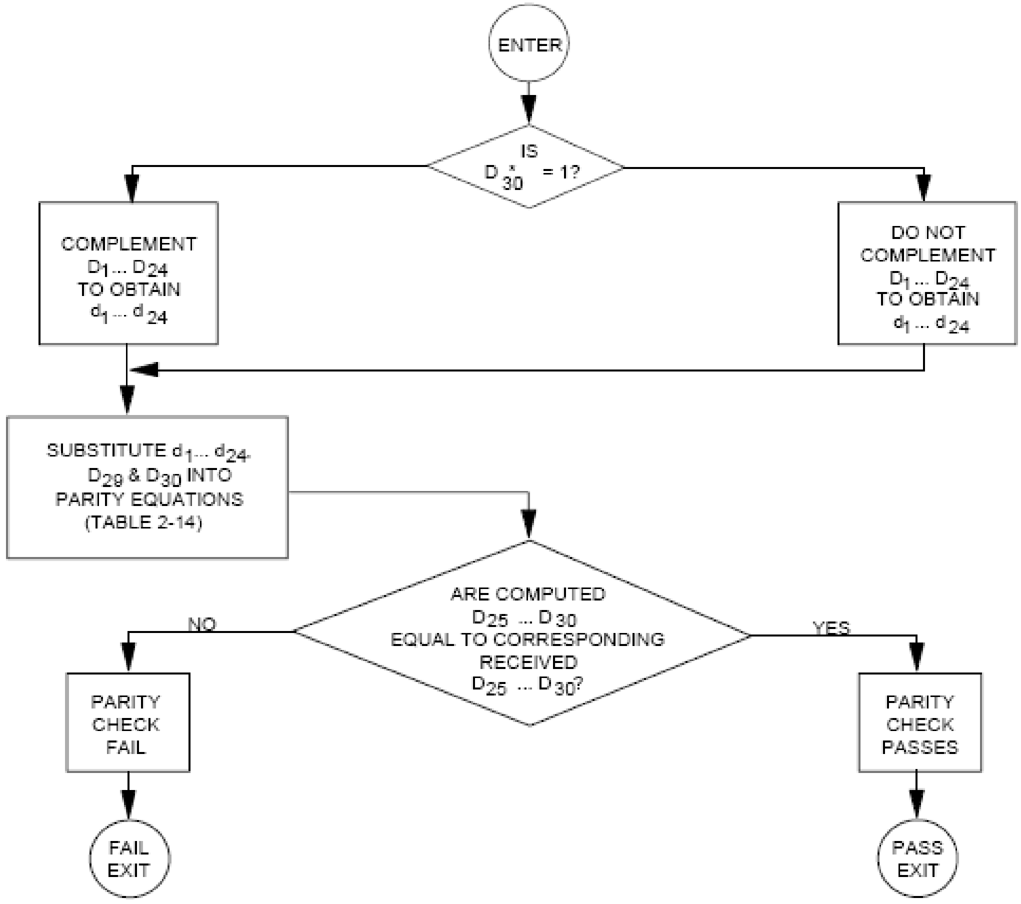

ENTER
IS $D_{30}^{*} = 1$?
COMPLEMENT $D_1 ... D_{24}$ TO OBTAIN $d_1 ... d_{24}$
DO NOT COMPLEMENT $D_1 ... D_{24}$ TO OBTAIN $d_1 ... d_{24}$
SUBSTITUTE $d_1 ... d_{24}$, D_{29} & D_{30} INTO PARITY EQUATIONS (TABLE 2-14)
ARE COMPUTED $D_{25} ... D_{30}$ EQUAL TO CORRESPONDING RECEIVED $D_{25} ... D_{30}$?
NO
YES
PARITY CHECK FAIL
PARITY CHECK PASSES
FAIL EXIT
PASS EXIT

Figure 1. Example Flow Chart for User Implementation of Parity Algorithm

User Algorithm for Ephemeris Determination

The user will compute the ECEF coordinates of position for the phase center of each satellite's L-Band antenna utilizing a variation of the equations shown in Table 2. Subframes 2 and 3 parameters are Keplerian in appearance; the values of these parameters, however, are obtained via a least squares curve fit of the predicted ephemeris for the phase center of the satellite's antenna (time-position quadruples; t, x, y, z).

a. Coordinate System

The equations given in Table 2 provide the satellite's antenna phase center position in the WGS-84 Earth-Centered Earth-Fixed reference frame defined as follows:

ORIGIN = Earth's center of mass*

Z-AXIS = Parallel to the direction of the CONVENTIONAL INTERNATIONAL ORIGIN (CIO) for polar motion, as defined by the BUREAU INTERNATIONAL DE L'HEURE (BIH) on the basis of the latitudes adopted for the BIH stations**

X-AXIS = Intersection of the WGS-84 reference meridian plane and the plane of the mean astronomic equator, the reference meridian being parallel to the zero meridian defined by the BUREAU INTERNATIONAL DE L'HEURE (BIH) on the basis of the longitudes adopted for the BIH stations***

Y-AXIS = Completes a right-handed Earth-Centered, Earth-Fixed orthogonal coordinate system, measured in the plan of the mean astronomic equator 90 degrees east of the X-axis***

* Geometric center of WGS-84 ellipsoid

** Rotation axis of WGS-84 ellipsoid

*** X, Y axis of WGS-84 ellipsoid

b. Geometric Range Correction

When computing the geometric range, the user will account for the effects due to earth rotation rate (reference Table 2) during the time of signal propagation so as to evaluate the path delay in an inertially stable coordinate system. Specifically, if the user works in Earth-fixed coordinates the user should add $(-\Omega_e y\Delta t\ ,\Omega_e x\Delta t,0)$ to the position estimate (x, y, z).

Application of Correction Parameters

In order to properly account for satellite clock bias and propagation delays, the user receiver must perform corrections to observed pseudo range measurements. The pseudo range is defined as:

$$PR_{measured} = c(t_{received} - t_{transmitted})$$

where

$PR_{measured}$ = measured pseudo range

$t_{received}$ = time that ranging measurement was received at the user location

$t_{transmitted}$ = time that ranging signal was transmitted from the satellite

Group Delay Application

The SPS user who utilizes the L1 frequency will modify the code phase offset with the equation:

$$(\Delta t_{SV})_{L1} = \Delta t_{SV} - T_{GD}$$

where TGD is provided to the user as subframe 1 data.

Table 2. Elements of Coordinate Systems

$A = \left(\sqrt{A}\right)^2$	Semi-major axis
$n_0 = \sqrt{\frac{\mu}{A^3}}$	Computed mean motion - rad/sec
$t_k = t - t_{oe}*$	Time from ephemeris reference epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e \sin E_k$	Kepler's equation for eccentric anomaly (may be solved by iteration) - radians
$\nu_k = \tan^{-1}\left\{\frac{\sin\nu_k}{\cos\nu_k}\right\} = \tan^{-1}\left\{\frac{\sqrt{1 - e^2}\sin E_k \ / \ (1 - e\cos E_k)}{(\cos E_k - e) \ / \ (1 - e\cos E_k)}\right\}$	True anomaly
$E_k = \cos^{-1}\left\{\frac{e + \cos\nu_k}{1 + e\cos\nu_k}\right\}$	Eccentric anomaly
$\Phi_k = \nu_k + \omega$	Argument of latitude
	<u>Second Harmonic Perturbations</u>
$\delta u_k = C_{us}\sin 2\Phi_k + C_{uc}\cos 2\Phi_k$	Argument of latitude correction
$\delta r_k = C_{rc}\cos 2\Phi_k + C_{rs}\sin 2\Phi_k$	Radius correction
$\delta i_k = C_{ic}\cos 2\Phi_k + C_{is}\sin 2\Phi_k$	Correction to inclination
$u_k = \Phi_k + \delta u_k$	Corrected argument of latitude
$r_k = A(1 - e\cos E_k) + \delta r_k$	Corrected radius
$i_k = i_0 + \delta i_k + (\mathrm{IDOT})\, t_k$	Corrected inclination
$\left.\begin{aligned} x_k' &= r_k \cos u_k \\ y_k' &= r_k \sin u_k \end{aligned}\right\}$	Positions in orbital plane
$\Omega_k = \Omega_0 + \left(\dot{\Omega} - \dot{\Omega}_e\right) t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node
$\left.\begin{aligned} x_k &= x_k' \cos\Omega_k - y_k' \cos i_k \sin\Omega_k \\ y_k &= x_k' \sin\Omega_k + y_k' \cos i_k \cos\Omega_k \\ z_k &= y_k' \sin i_k \end{aligned}\right\}$	Earth-Centered, Earth-Fixed coordinates

* t is GPS system time at time of transmission, i.e., GPS time corrected for transit time (range/speed of light). Furthermore, t_k shall be the actual total time difference between the time t and the epoch time t_{oe}, and must account for beginning or end of week crossovers. That is, if t_k is greater than 302,400 seconds, subtract 604,800 seconds from t_k. If t_k is less than -302,400 seconds, add 604,800 seconds to t_k.

Satellite Clock Correction

The polynomial defined in the following allows the user to determine the effective satellite PRN code phase offset referenced to the phase center of the satellite antennas (Δt_{sv}) with respect to GPS system time (t) at the time of data transmission.

The coefficients transmitted in the subframe 1 describe the offset apparent to the control segment two-frequency receivers for the interval of time in which the parameters are transmitted. This estimated correction accounts for the deterministic satellite clock error characteristics of bias, drift and aging, as well as for the satellite implementation characteristics of group delay bias and mean differential group delay. Since these coefficients do not include corrections for relativistic effects, the user's equipment must determine the requisite relativistic correction. Accordingly, the offset given below includes a term to perform this function.

The user will correct the time received from the satellite with the equation (in seconds)

$$t = t_{sv} - (\Delta t_{sv})L1 \qquad (1)$$

where

t = GPS system time (seconds),

t_{sv} = effective SV PRN code phase time at message transmission time (seconds),

(Δt_{sv}) L1 = SV PRN code phase time offset (seconds).

The satellite PRN code phase offset is given by

$$(\Delta t_{sv})_{L1} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_r - T_{GD} \qquad (2)$$

where

a_{f0}, a_{f1}, and a_{f2} are the polynomial coefficients given in subframe 1, t_{oc} is the clock data reference time in seconds, and Δt_r is the relativistic correction term (seconds) which is given by

$$\Delta t_r = F\, e\, (A)^{1/2} \sin E_k$$

The orbit parameters (e, A, E_k) used here are described in discussions of data contained in subframes 2 and 3, while F is a constant whose value is

$$F = -2\, (\mu)^{1/2} / c^2 = -\, 4.442807633\, (10)^{-10} \text{ sec} / (\text{meter})^{1/2}$$

Note that equations (1) and (2), as written, are coupled. While the coefficients a_{f0}, a_{f1}, and a_{f2} are generated by using GPS time as indicated in equation (2), sensitivity of tsv to t is negligible. This negligible sensitivity will allow the user to approximate t by t_{sv} in equation (2). The value of t must account for beginning or end of week crossovers. That is, if the quantity t - t_{oc} is greater than 302,400 seconds, subtract 604,800 seconds from t. If the quantity t - t_{oc} is less than -302,400 seconds, add 604,800 seconds to t.

Ionospheric Model

The SPS user should correct the time received from the satellite for ionospheric effect by utilizing parameters contained in page 18 of subframe 4 in the model given below. It is estimated that the use of this model will provide at least a 50 percent reduction in the SPS user's RMS error due to ionospheric propagation effects.

The ionospheric correction model is given by

$$T_{iono} = \begin{Bmatrix} F*\left[5.0*10^{-9} + (AMP)\left(1-\frac{x^2}{2}+\frac{x^4}{24}\right)\right], & |x| < 1.57 \\ F*(5.0*10^{-9}) & , |x| \geq 1.57 \end{Bmatrix} (sec)$$

where

$$AMP = \begin{Bmatrix} \sum_{n=0}^{3} \alpha_n \ \phi_m^n, \ AMP \geq 0 \\ \text{if } AMP < 0, \ AMP = 0 \end{Bmatrix} (sec)$$

$$x = \frac{2\pi(t - 50400)}{PER}, \ (radians)$$

$$PER = \begin{Bmatrix} \sum_{n=0}^{3} \beta_n \ \phi_m^n, \ PER \geq 72{,}000 \\ \text{if } PER < 72{,}000, \ PER = 72{,}000 \end{Bmatrix} (sec)$$

$F = 1.0 + 16.0[0.53 - E]^3$ and

α_n and β_n are the satellite transmitted data words with n = 0, 1, 2, and 3.

Other equations that must be solved are

$$\phi_m = \phi_i + 0.064 \cos(\lambda_i - 1.617) \ (semi-circles),$$

$$\lambda_i = \lambda_u + \frac{\psi \sin A}{\cos \phi_i} \ (semi-circles),$$

$$\phi_i = \begin{Bmatrix} \phi_u + \psi \cos A(semi-circles), \ |\phi_i| \leq 0.416 \\ \text{if } \phi_i > 0.416, \text{ then } \phi_i = +0.416 \\ \text{if } \phi_i < -0.416, \text{ then } \phi_i = -0.416 \end{Bmatrix} (semi-circles),$$

$$\psi = \frac{0.00137}{E + 0.11} - 0.022 \ (semi-circles),$$

$$t = 4.32 * 10^4 \ \lambda_i + \text{GPS time (sec)}$$

where

$0 \leq t < 86400$, therefore: if $t \geq 86400$ seconds, subtract 86400 seconds;

if $t < 0$ seconds, add 86400 seconds.

The terms used in computation of ionospheric delay are as follows:

- Satellite Transmitted Terms

α_n the coefficients of a cubic equation representing the amplitude of the vertical delay

(4 coefficients = 8 bits each)

β_n the coefficients of a cubic equation representing the period of the model

(4 coefficients = 8 bits each)

- Receiver Generated Terms

E elevation angle between the user and satellite (semi-circles)

A azimuth angle between the user and satellite, measured clockwise positive from the true North (semi-circles)

ϕ_u user geodetic latitude (semi-circles) WGS-84

λ_u user geodetic longitude (semi-circles) WGS-84

GPS time receiver computed system time

- Computed Terms

x phase (radians)

F obliquity factor (dimensionless)

t local time (sec)

ϕ_m geomagnetic latitude of the earth projection of the ionospheric intersection point (mean ionospheric height assumed 350 km) (semicircles)

λ_i geomagnetic latitude of the earth projection of the ionospheric intersection point (semi-circles)

ϕ_i geomagnetic latitude of the earth projection of the ionospheric intersection point (semi-circles)

ψ earth's central angle between user position and earth projection of ionospheric intersection point (semi-circles)

Universal Coordinated Time (UTC)

Depending on the relationship of the effectivity date to the user's current GPS time, the following three different UTC/GPS-time relationships exist:

a. Whenever the effectivity time indicated by the WNLSF and the DN values is not in the past (relative to the user's present time), and the user's present time does not fall in the timespan which starts at DN + 3/4 and ends at DN + 5/4, the UTC/GPS-time relationship is given by

$$t_{UTC} = (t_E - \Delta t_{UTC}) \text{ \{Modulo 86400 seconds\}}$$

where t_{UTC} is in seconds and

$$\Delta t_{UTC} = \Delta t_{LS} + A_0 + A_1 (t_E - t_{ot} + 604800 (WN - WN_t)), \text{ (seconds);}$$

tE = GPS time as estimated by the user on the basis of correcting tsv for factors described in previous paragraph as well as for ionospheric and SA (dither) effects;

Δt_{LS} = delta time due to leap seconds;

A0 and A1 = constant and first order terms of polynomial;

tot = reference time for UTC data;

WN = current week number (derived from subframe 1);

WNt = UTC reference week number.

The estimated GPS time (t_E) is in seconds relative to end/start of week. The reference time for UTC data (tot) is referenced to the start of that week whose number (WNt). The WNt value consists of the eight LSBs of the full week number. The user must account for the truncated nature of this parameter as well as truncation of WN, WNt, and WLSF due to rollover of the full week number. The absolute value of the difference between the untruncated WN and WNt values will not exceed 127.

b. Whenever the user's current time falls within the timespan of DN + 3/4 to DN + 5/4, proper accommodation of the leap second event with a possible week number transition is provided by the following expression for UTC:

$$t_{UTC} = W \text{ [Modulo } (86400 + \Delta t_{LSF} - \Delta t_{LS})] \text{ , (seconds);}$$

where

$$W = (t_E - \Delta t_{UTC} - 43200) \text{ [Modulo 86400]} + 43200 \text{ , (seconds);}$$

and the definition of Δt_{UTC} (as given in "a" above) applies throughout the transition period. Note that when a leap second is added, unconventional time values of the form 23: 59: 60.xxx are encountered. Some user equipment may be designed to approximate UTC by decrementing the running count of time within several seconds after the event, thereby promptly returning to a proper time indication. Whenever a leap second event is encountered, the user equipment must consistently implement carries or borrows into any year/week/day counts.

c. Whenever the effectivity time of the leap second event, as indicated by the WNLSF and DN values, is in the "past" (relative to the user's current time), the relationship previously given for t_{UTC} in "a" above is valid except that the value of Δt_{LSF} is substituted for Δt_{LS}. The CS will coordinate the update of UTC parameters at a future upload so as to maintain a proper continuity of the t_{UTC} time scale.

Implementation for the solution

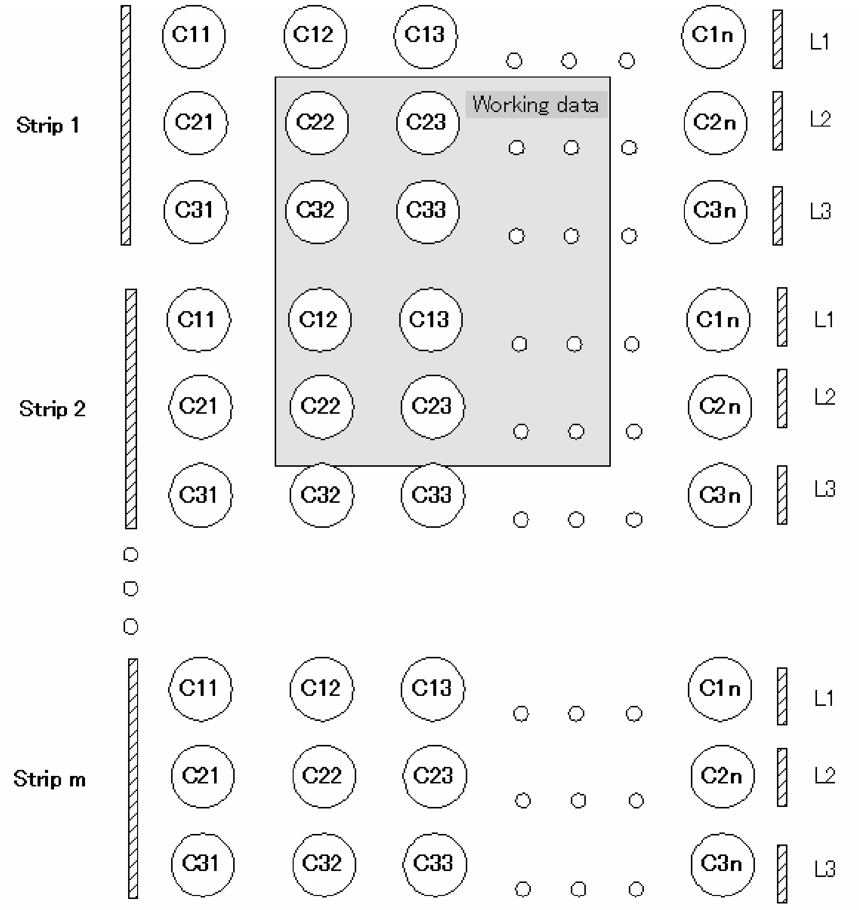
The application is built around the Multiple Document Interface (MDI) specification. The purpose of this application is providing the user a tool to interact with map images. The images are TIF files and represent maps of all regions in Japan. There are two (2) kinds of maps the application can work with. The classification is based on the map scale specification and these are 1:200000 and 1:25000.

The information in the TIF file can have 2 types of organization:

- by strips (see figure 2)
- by tiles (see figure 3)

Both types are supported.

Figure 2. TIF Strip Organization



Because the PDA memory is limited, not the entire TIF file is loaded in the main memory. Only an image of 900x900 pixels is kept in the main memory. This image is named "Working data".

When the user drags the map outside of the "Working data" image, then this image is updated with a new sub-zone of the TIF map.
When this happens, the hour glass cursor appears on the screen for a short time.

For each TIF file organization (strips or tiles) there is an algorithm to extract a "Working data" sub-zone.

In the above picture a "Cij" element is referred to as the color element (pixel) at the row i and column j in a certain strip. Li is referred to as the line(row) i in a strip.

A strip contains one or more lines of pixels. When a "Working data" covers a strip, the entire strip is loaded from the file and then only the necessary zone is extracted from the strip. A "Working data" zone can cover one or more strips – all of them are read from the file. This leads to slow performance if the TIF file has a big width. For this reason, the TIF tile organization is preferred because it is much faster.

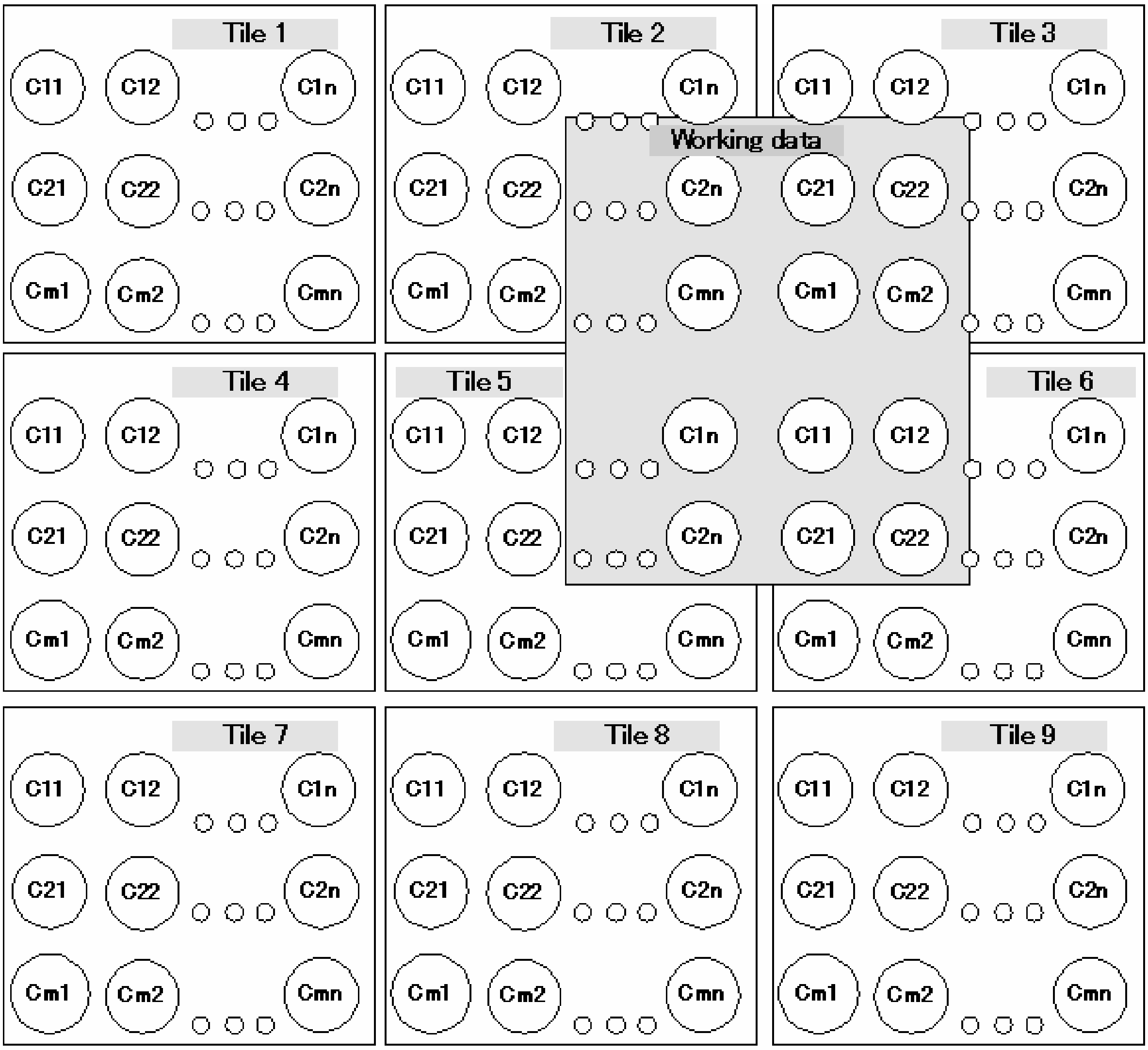


Figure 3. TIF Tile Organization

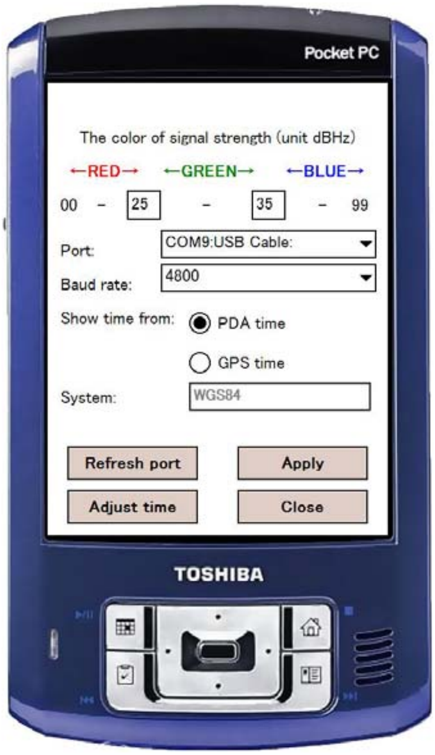
The tile number increases from left to right and from top to bottom.

Each tile has a fixed number of rows and columns. These parameters are established when the map is exported. The “Cij” elements are referred to as color elements (pixels) in a tile.

A working data can cover one or more tiles. Only the covered tiles are loaded from the file and after that only the area of interest is kept in the main memory. The tile organization is faster than the strip organization because less information has to be loaded from the file at each “Working data” update.

In the program, for configuring the GPS and receiving GPS sentences it was used the framework “Smart Device Framework 1.2”.

The next three figures represent the main screens of the program used for: set GPS characteristics, display GPS information and find GPS coordinates on a TIF map.



The color of signal strength (unit dBHz)

←RED→ ←GREEN→ ←BLUE→

00 - 25 - 35 - 99

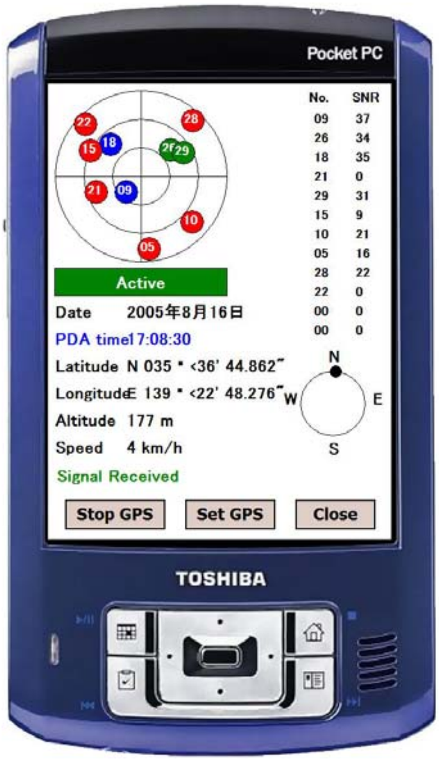
Port: COM9:USB Cable:

Baud rate: 4800

Show time from: ● PDA time ○ GPS time

System: WGS84

Figure 4. Screen from application used for GPS Settings



No.	SNR
09	37
26	34
18	35
21	0
29	31
15	9
10	21
05	16
28	22
22	0
00	0
00	0

Active

Date 2005年8月16日

PDA time17:08:30

Latitude N 035 ° <36' 44.862"

LongitudeE 139 ° <22' 48.276"

Altitude 177 m

Speed 4 km/h

Signal Received

Figure 5. Screen from application used for displaying GPS Information

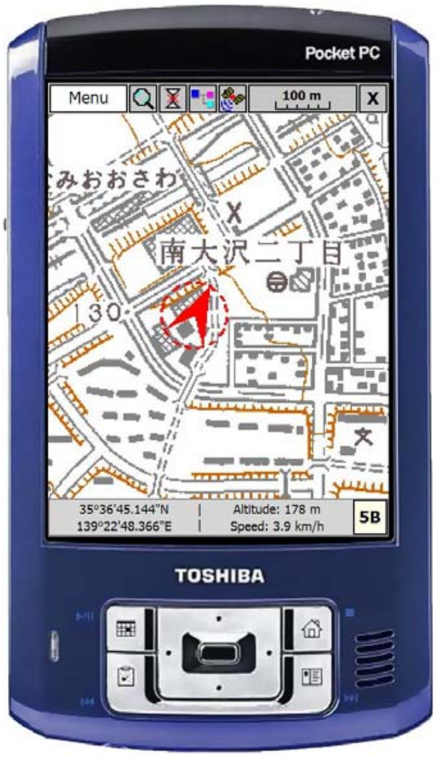


Figure 6. Screen for finding GPS coordinates on map

About PDA

Personal digital assistants (PDAs) are handheld devices that were originally designed as personal organisers, but became much more versatile over the years. PDAs have many uses: calculating, use as a clock and calendar, playing computer games, accessing the Internet, sending and receiving e-mail, use as a radio or stereo, video recording, recording notes, use as an address book, GPS and use as a spreadsheet. Newer PDAs also have both color screens and audio capabilities, enabling them to be used as mobile phones (smartphone), web browsers or media players. Many PDAs can access the Internet, intranets or extranets via Wi-Fi, or Wireless Wide-Area Networks (WWANs). One of the most significant PDA characteristic is the presence of a touch screen.

Touch screen PDAs, including Windows Pocket PC devices, usually have a detachable stylus that can be used on the touch screen. Interaction is then done by tapping the screen to activate buttons or menu choices, and dragging the stylus to, for example, highlight text.

Text input is usually done in one of two ways:

- Using a virtual keyboard, where a keyboard is shown on the touch screen. Input is done by tapping the letters.
- Using letter or word recognition, where letters or words are written on the touch screen, and then "translated" to letters in the currently activated text field. Despite rigorous research and development projects, this data input method still requires much patience from the user since it tends to be rather inaccurate.

The currently major PDA operating systems are:

- Palm OS - owned by PalmSource

- Windows Mobile (Pocket PC), (based on the Windows CE kernel) - owned by Microsoft
- RIM for the BlackBerry - owned by Research In Motion
- Many operating systems based on the Linux kernel - free (not owned by any company). These include:
    - GPE - Based on GTK+/X11
    - OPIE user interface/Qtopia - based on Qt/E Qtopia is developed by Trolltech, OPIE is a fork of Qtopia developed by volunteers
- Symbian OS (formerly EPOC) owned by Ericsson, Motorola, Panasonic, Nokia, Samsung, Siemens and Sony Ericsson

Many PDAs run using a variation of the ARM architecture (usually denoted by the Intel XScale trademark). This encompasses a class of RISC microprocessors that are widely used in mobile devices and embedded systems, and its design was influenced strongly by a popular 1970s/1980s CPU, the MOS Technology 6502.

About GPS

The Global Positioning System, usually called GPS, is the only fully-functional satellite navigation system. A constellation of more than two dozen GPS satellites broadcasts precise timing signals by radio, allowing any GPS receiver (abbreviated to GPSr) to accurately determine its location (longitude, latitude, and altitude) in any weather, day or night, anywhere on Earth.

GPS has become a vital global utility, indispensable for modern navigation on land, sea, and air around the world, as well as an important tool for map-making and land surveying. GPS also provides an extremely precise time reference, required for telecommunications and some scientific research, including the study of earthquakes. GPS receivers can also gauge altitude and speed with a very high degree of accuracy.

Calculating positions

To calculate its position, a receiver first needs to know the precise time. To do this, it uses an internal crystal oscillator-based clock that is continually updated by the signals being sent in L1 from various satellites. At that point the receiver identifies the visible satellites by the distinct pattern in their C/A codes. It then looks up the ephemeris data for each satellite, which was captured from the NM and stored in memory. This data is used in a formula that calculates the precise location of the satellites at that point in time.

Finally the receiver must calculate the time delay to each satellite. To do this, it produces an identical C/A sequence from a known seed number. The time delay is calculated by increasingly delaying the local signal and comparing it to the one received from the satellite; at some point the two signals will match up, and that delay is the time needed for the signal to reach the receiver. The delay is generally between 65 and 85 milliseconds. The distance to that satellite can then be calculated directly, the so-called pseudorange.

The receiver now has two key pieces of information: an accurate estimate of the position of the satellite, and an accurate measurement of the distance to that satellite. This tells the receiver that it lies on the surface of an imaginary sphere whose radius is that distance. To calculate the precise position, at least four such measurements are taken simultaneously. This places the receiver at the intersection of the four imaginary spheres. Since the C/A pattern repeats every millisecond, it can only be used to place the user within

300 kilometers (180 mi). Thus the multiple measurements are also needed to determine whether the receiver has lined up its internal C/A code properly, or is "one off".

The calculation of the position of the satellite, and thus the time delay and range to it, all depend on the accuracy of the local clock. The satellites themselves are equipped with extremely accurate atomic clocks, but this is not economically feasible for a receiver. Instead, the system takes redundant measurements to re-capture the correct clock information.

To understand how this works, consider a local clock that is off by 0.1 microseconds, or about 30 meters (100 ft) when converted to distance. When the position is calculated using this clock, the range measurements to each of the satellites will read 30 meters too long. In this case the four spheres will not overlap at a point, instead each sphere will intersect at a different point, resulting in several potential positions about 30 meters apart. The receiver then uses a mathematical technique to calculate the clock error that would produce this offset, in this case 0.1 microseconds, adjusts the range measurements by this amount, and then updates the internal clock to make it more accurate.

This technique can be applied with any four satellites. Commercial receivers therefore attempt to "tune in" to as many satellites as possible, and repeatedly make this correction. In doing so, clock errors can be reduced almost to zero. In practice, anywhere from six to ten measurements are taken in order to round out errors, and civilian receivers generally have 10 to 12 channels in total.

Calculating a position with the P(Y) signal is generally similar in concept, assuming one can decrypt it. The encryption is essentially a safety mechanism; if a signal can be successfully decrypted, it is reasonable to assume it is a real signal being sent by a GPS satellite. In comparison, civil receivers are highly vulnerable to spoofing since a set of navigationally consistent C/A signals can be generated using readily available off the shelf signal generators. Even if the victim receiver incorporates RAIM features it will still "buy in' to the spoofing signals since RAIM only checks to make sure the signals make sense from a navigational perspective.

Accuracy

Best case. The position calculated by a GPS receiver relies on three accurate measurements: the current time, the position of the satellite, and the time delay for the signal. Errors in the clock signal can be reduced using the method above, meaning that the overall accuracy of the system is generally based on the accuracy of the position and delay.

The measurement of the delay requires the receiver to "lock onto" the same sequence of bits being sent from the satellite. This can be made relatively accurate by timing comparing the rising or trailing edges of the bits. Modern electronics can lock the two signals to about 1% of a bit time, or in this case about 1% of a microsecond. Since light travels at 299,792,458 m/s, this represents an error of about 3 meters (10 ft), the minimum error possible given the timing of the C/A signal.

This can be improved by using the higher-speed P(Y) signal, assuming the same 1% accuracy in locking the retrieved P-code to the internally generated version. In this case the same calculation results in an accuracy of about 30 centimeters (1 ft). Since the P-code repeats at 10.23 MHz, it has a "repeat range" of about 30 kilometers (20 mi). This explains the terminology; when using the P-code, it was first necessary to calculate a coarse position with the C/A code in order to determine how to line up the P-code with the internally generated copy.

Atmospheric effects. One of the biggest problems for GPS accuracy is that changing atmospheric conditions change the speed of the GPS signals unpredictably as they pass through the ionosphere. The effect is minimized when the satellite is directly overhead and becomes greater toward the horizon, since the satellite signals must travel through the greater "thickness" of the ionosphere as the angle increases. Once the receiver's rough location is known, an internal mathematical model can be used to estimate and correct for the error.

Because ionospheric delay affects the speed of radio waves differently based on their frequencies, the second frequency band (L2) can be used to help eliminate this type of error. Some military and expensive survey-grade civilian receivers can compare the difference between the P(Y) signal carried in the L1 and L2 frequencies to measure atmospheric delay and apply precise corrections. This correction can be applied even without decrypting the P(Y) signal, as long as the encryption key is the same on both channels. In order to make this easier, the U.S. Government has added a new civilian signal on L2, called L2C, starting with the Block IIR-M satellites. The first Block IIR-M was launched in 2005. It allows a direct comparison of the L1 and L2 signals for ionospheric correction.

The effects of the ionosphere are generally slow-moving and can easily be tracked. The effects for any particular geographical area can be easily calculated by comparing the GPS-measured position to a known surveyed location. This correction, say, "10 meters to the east" is also valid for other receivers in the same general location. Several systems send this information over radio or other links to the receivers, allowing them to make better corrections than a comparison of L1 and L2 alone could.

The amount of humidity in the air also has a delaying effect on the signal, resulting in errors similar to those generated in the ionosphere but located much closer to the ground in the troposphere. The areas affected by these problems tend to be smaller in area and faster moving than the billows in the ionosphere, making accurate correction for these effects more difficult.

Multipath effects. GPS signals can also be affected by multipath issues, where the radio signals reflect off surrounding terrain; buildings, canyon walls, hard ground, etc. This delay in reaching the receiver causes inaccuracy. A variety of receiver techniques, most notably narrow correlator spacing, have been developed to mitigate multipath errors. For long delay multipath, the receiver itself can recognize the wayward signal and discard it. To address shorter delay multipath from the signal reflecting off the ground, specialized antennas may be used. This form of multipath is harder to filter out since it is only slightly delayed as compared to the direct signal, causing effects almost indistinguishable from routine fluctuations in atmospheric delay.

Multipath effects are much less severe in dynamic applications such as cars and planes. When the GPS antenna is moving, the false solutions using reflected signals quickly fail to converge and only the direct signals result in stable solutions.

Ephemeris and clock errors. The navigation message from a satellite is sent out only every 12.5 minutes. In reality, the data contained in these messages tends to be "out of date" by an even larger amount. Consider the case when a GPS satellite is boosted back into a proper orbit; for some time following the maneuver, the receiver’s calculation of the satellite's position will be incorrect until it receives another ephemeris update. Additionally, the amount of accuracy sent in the ephemeris is limited by the bandwidth; using the data from the satellites alone limits its accuracy.

Further, while it is true that the onboard clocks are extremely accurate, they do suffer from clock drift. This problem tends to be very small, but may add up to 2 meters (6 ft) of inaccuracy.

These sorts of errors are even more "stable" than ionospheric problems and tend to change on the order of days or weeks, as opposed to minutes. This makes correcting for these errors fairly simple by sending out a more accurate almanac on a separate channel.

Conclusions

GPS allows receivers to accurately calculate their distance from the GPS satellites. The receivers do this by measuring the time delay between when the satellite sent the signal and the local time when the signal was received. This delay, multiplied by the speed of light, gives the distance to that satellite. The receiver also calculates the position of the satellite based on information periodically sent in the same signal. By comparing the two, position and range, the receiver can discover its own location.

Several "real world" effects intrude and degrade the accuracy of the system. These are outlined in the table below (Table 3), with descriptions following. When all of these effects are added up, GPS is typically accurate to about 15 meters (50 ft).

Table 3. Sources of GPS error

Source	*Effect*
Ionospheric effects	± 5 meter
Ephemeris errors	± 2.5 meter
Satellite clock errors	± 2 meter
Multipath distortion	± 1 meter
Tropospheric effects	± 0.5 meter
Numerical errors	± 1 meter or less

Differential GPS (DGPS) helps correct these errors. The basic idea is to gauge GPS inaccuracy at a stationary receiver station with a known location. Since the DGPS hardware at the station already knows its own position, it can easily calculate its receiver's inaccuracy. The station then broadcasts a radio signal to all DGPS-equipped receivers in the area, providing signal correction information for that area. In general, access to this correction information makes DGPS receivers much more accurate than ordinary receivers.

The most essential function of a GPS receiver is to pick up the transmissions of at least four satellites and combine the information in those transmissions with information in an electronic almanac, all in order to figure out the receiver's position on Earth.

Once the receiver makes this calculation, it can tell you the latitude, longitude and altitude (or some similar measurement) of its current position.

References

1. Federal Aviation Administration, **FAA WAAS fact-sheet,** EGNOS, the European equivalent, went on-line in 2005.
2. Hydrographic Journal. **Developments in Global Navigation Satellite Systems.** April 2002.
3. United States Department of Defense. **Announcement of Initial Operational Capability.** December 8, 1993.
4. National Archives and Records Administration. **U.S. Global positioning system policy.** March 29, 1996.
5. Federal Highway Administration. **Nationwide DGPS Program Fact Sheet.**
6. Office of Science and Technology Policy. **Presidential statement to stop degrading GPS.** May 1, 2000.
7. HowStuffWorks. **How GPS Receivers Work.**
8. Dana, Peter H. **GPS Orbital Planes.** August 8, 1996.
9. USNO. **NAVSTAR Global Positioning System.**
10. NMEA **NMEA 2000**
11. Rizos, Chris. University of New South Wales. **GPS Satellite Signals.** 1999.
12. Physics Today. **Relativity and GPS**. May 2002.
13. Deines, **Uncompensated relativity effects for a ground-based GPSA receiver,** Position Location and Navigation Symposium, 1992. Record. '500 Years After Columbus — Navigation Challenges of Tomorrow'. IEEE PLANS '92.
14. United States Naval Research Laboratory. **National Medal of Technology for GPS.** November 21, 2005
15. Phrack. **Issue 0x3c (60), article 13.** December 28, 2002.
16. GPS World. **The hunt for an unintentional GPS jammer.** January 1, 2003.
17. Ruley, John. AVweb. **GPS jamming.** February 12, 2003.
18. Space Environment Center. **SEC Navigation Systems GPS Page.** August 26, 1996.
19. American Forces Press Service. **CENTCOM charts progress.** March 25, 2003.
20. Peter H. Dana: **Global Positioning System Overview** — Large amount of technical information and discussion.
21. **Simplified explanation of GPS** at howstuffworks.com
22. **GPS SPS Signal Specification, 2nd Edition** — The official (civilian) signal specification.
23. **Satellite Navigation: GPS & Galileo (PDF)** — 16-page paper about the history and working of GPS, touching on the upcoming Galileo
24. **History of GPS**, including information about each satellite's configuration and launch.
25. **The GPS Joint Program Office (GPS JPO)** — Responsible for designing and acquiring the system on behalf of the US Government.
26. **University of New Brunswick, In Simple Terms, How Does GPS Work?**

[1] Nicolae - Iulian Enescu is Teaching Assistant at the Computer Systems and Communications Department at the Faculty of Automation, Computers and Electronics, University of Craiova, Romania.
In July 2000 he graduated the Faculty of Automation, Computers and Electronics at the University of Craiova.
From 2001 he is PhD student at the Academy of Economic Studies, Faculty of Cybernetics, Statistics and Economic Informatics.
He is author and coauthor of more than 20 journal articles and scientific presentations at conferences, and coauthor of 3 books.
His work focuses on the data communication, assembler, programming techniques, high level programming languages, databases and software testing.
He collaborate with many IT&C Companies for developing some software projects.