

Algorithms for Locating GPS Coordinates on PDA

NICOLAE-IULIAN ENESCU

DAN MANCAS

ECATERINA - IRINA MANOLE

Computer and Communication Engineering Department

Faculty of Automation, Computers and Electronics

Bvd. Decebal, Nr. 107, 200440, Craiova, Dolj

ROMANIA

dan.mancas@ucv.ro, nenescu@cs.ucv.ro, catya_ace@yahoo.com

Abstract: - The purpose of our research is to find algorithms for loading large maps on PDA and place on it the geographical location given by latitude and longitude. C++ and C# programming languages have been used for implementation. It was elaborated a software which runs on PDA with Windows Mobile 2003 OS and implements algorithms for finding the coordinate on a map represented like a TIF image. The application must localize the coordinates known by longitude and latitude over a given image in TIF format which can represent a map with scale 1:200000 or 1:25000. The map headers contain information that refers to map limit. An algorithm that permits map synchronization when the value of current map is out of limit it was necessary.

Key-Words: GPS, map, TIF, coordinates, PDA, Windows Mobile 2003

1 Informatic solution

As a result of many PDA have Windows Mobile OS, the informatics solution for implement the requests was to use .NET CF C# with Microsoft Visual Studio .NET 2003. Due to .NET CF has some limitations regarding loading a TIF image, a dynamic library for the module which load and save a TIF image it was build in C++ with Microsoft embedded Visual C++ 4.

Because the program needed additional functions that are not in .NET CF it 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.

2 Algorithm

2.1 Map algorithms

200000 and 25000 maps structure is *mesh type*. It is called "*Standard Area Mesh*". This is the standard way to divide the area by constant interval of longitude and latitude. "*Standard Area Mesh*" has 3 levels.

2.1.1 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 the area by constant interval of longitude and latitude. "*Standard Area Mesh*" has 3 levels.

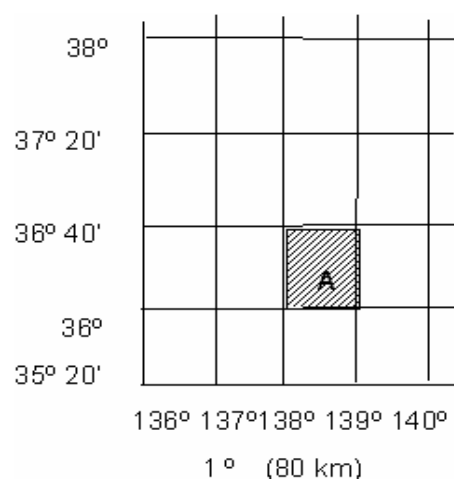


Fig. 1 – Primary Area Division

The Primary Area Division (the 1st Area Block or the First Mesh) is made by dividing the country map into rectangles with the width of 1-degree longitude and height of 2/3-degree latitude (1° X 40'), Fig. 1. The 1st Area Block has a correspondent area of 6,400km², 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.

First 2 digits are obtained by multiplying the south edge latitude degrees with 1.5, Fig. 2.

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

Mesh code for A is:

$$2^{\circ} \times 1/3 = 40'$$

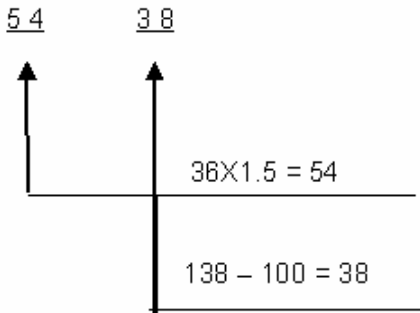


Fig. 2 – Mesh code for Primary Area Division

Dividing the 1st area division vertically – horizontally by 8 makes the 2nd area block (Secondary Area Division or the Second Mesh).

Each unit is 7' 30" x 5' and correspond to 1 unit of the 1:25000 map, which has an area of 100 km².

The mesh code for the 2nd area division is a 2-digit code. First digit is numbered from 0 for south edge to 7 for north along longitude line. Second digit is numbered from 0 to west edge to 7 for east edge along latitude line, Fig. 3.

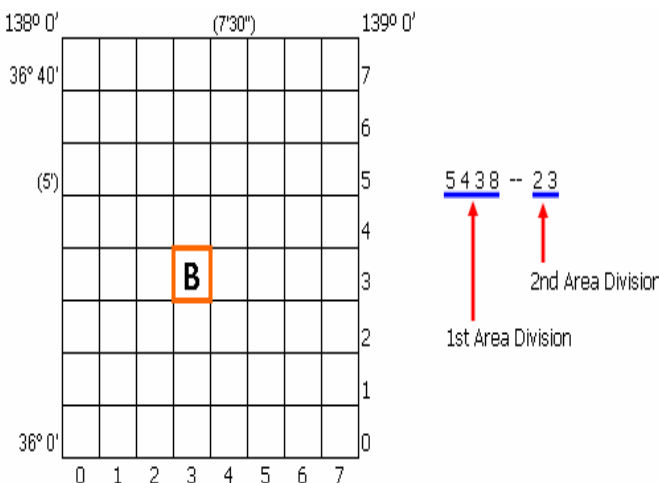


Fig. 3 – Secondary Area Division

Dividing the 2nd area division vertically – horizontally by 10 makes the 3rd area block (Third Area Division or the Base Mesh). Each unit is 45" x 30" and is corresponding to an area of 1 km².

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

The 3rd Area Division is called “Base Mesh”, Fig. 4.

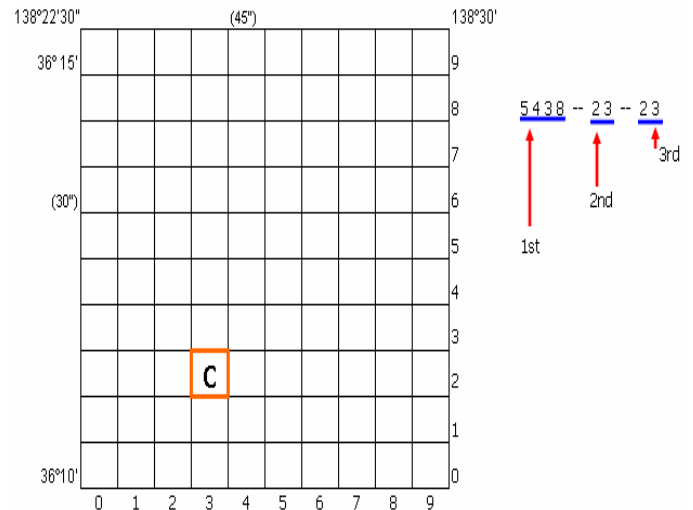


Fig. 4 – Third Area Division

2.1.2 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 toward 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, Fig. 5.

In theory, every map image should have a 2-pixel width boundary, gray colored. Unfortunately this is not true all the time. There are many cases when the boundary line is missing, or is only 1-pixel width or the color is not gray, Fig. 6.

The two arrows show the direction of searching. Every pixel, starting from the file boundary, it 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 it is used for the all-4 corners.

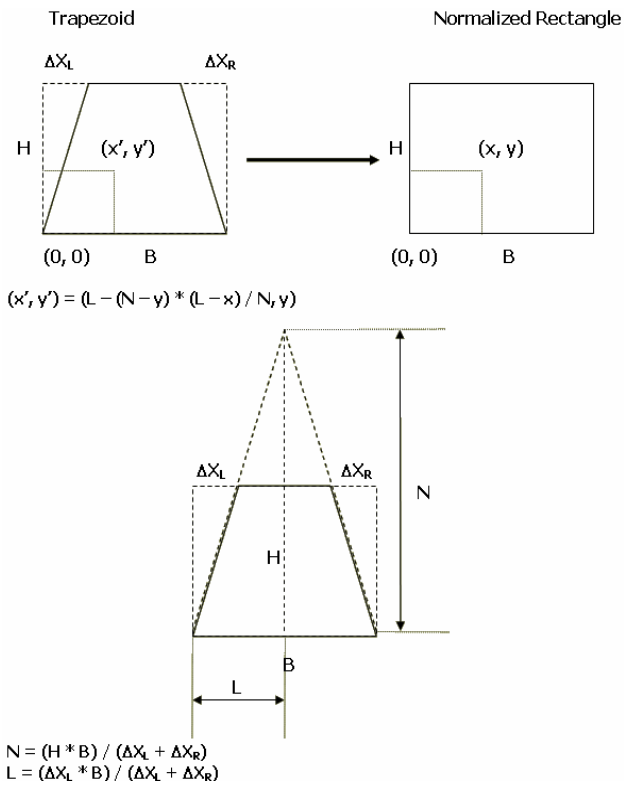


Fig. 5 - Transform maps in rectangles

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

- the search is done separate 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 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 founded value is bigger than 15 pixels, than the default value is kept.

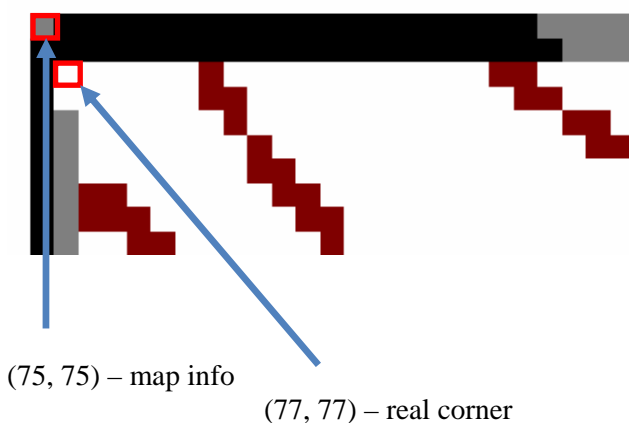


Fig. 6. Map boundary

The program searches the real corner by checking the color of pixels, Fig. 7.

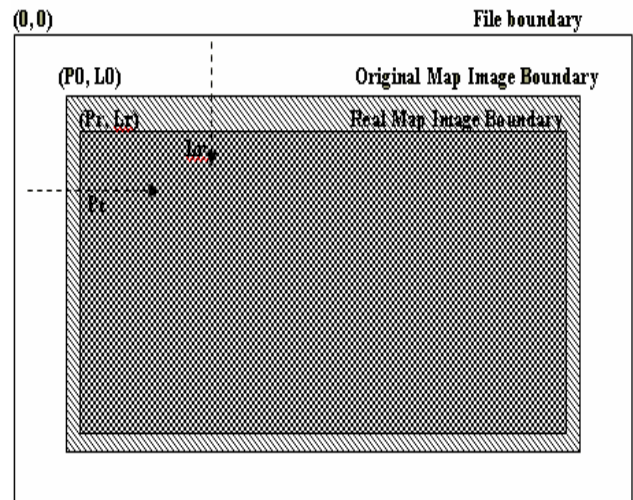


Fig. 7. Searching the real corner of a map

2.2 GPS algorithms

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.

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

2.2.2 Mathematical Constants

The speed of light used for generating the data is $c = 2.99792458 \times 10^8$ meters per second which is the official WGS-84 speed of light. We use the same value for the speed of light in computations.

Other WGS-84 constants required to use for satellite ephemeris calculations are:

- $\mu = 3.986005 \times 10^{14}$ meters³ / sec² which is WGS-84 value of the Earth's universal gravitational parameter
- $\Omega_e = 7.2921151467 \times 10^{-5}$ rad / sec which is 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 10⁸ meters/semicircle, and to the angular rate parameters is on the order of 10¹² 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 and is taken as

$$\pi = 3.1415926535898 \tag{1}$$

Parity Algorithm

We perform error detection of the decoded navigation data using the parity algorithm equations provided in table 1.

Table 1. Parity Encoding Equations[1]

$$D_1 = d_1 \oplus D_{30}^*$$

$$D_2 = d_2 \oplus D_{30}^*$$

$$D_3 = d_3 \oplus D_{30}^*$$

••

••

••

••

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

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

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

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

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

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

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

where:

- d_1, d_2, \dots, 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}, \dots, D_{30} are the computed parity bits
- $D_1, D_2, D_3, \dots, D_{29}, D_{30}$ are the bits transmitted by the satellite, and
- \oplus is the "Modulo-2" or "Exclusive-Or" operation.

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 $D_{25} \dots D_{30}$, which provide computed parity to compare with transmitted parity $D_{25} \dots D_{30}$.

2.2.3 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 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 we utilize a look-up table approach for interpreting the URA message [2].

2.2.4 Algorithm for Ephemeris Determination

We will compute the ECEF coordinates of position for the phase center of each satellite's L-Band antenna utilizing a variation of the equations.

The values of parameters 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) [1].

a. Coordinate System

The equations 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

b. Geometric Range Correction

When computing the geometric range, we will account for the effects due to earth rotation rate 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) .

2.2.5 Application of Correction Parameters

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

$$PR_{\text{measured}} = c(t_{\text{received}} - t_{\text{transmitted}}) \quad (2)$$

Where:

- PR_{measured} = measured pseudo range
- t_{received} = time that ranging measurement was received at the user location

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

2.2.6 Satellite Clock Correction

The polynomial defined in the following allows us 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 [2].

The coefficients transmitted 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 equipment must determine the requisite relativistic correction. Accordingly, the offset given below includes a term to perform this function.

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

$$t = t_{sv} - (\Delta t_{sv})_{L1} \quad (3)$$

where:

- t = GPS system time (seconds),
- t_{sv} = effective SV PRN code phase time at message transmission time (seconds),
- $(\Delta t_{sv})_{L1}$ = SV PRN code phase time offset (seconds).

The satellite PRN code phase offset is given by

$$(\Delta t_{sv})_{L1} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_r - T_{GD} \quad (4)$$

Where:

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

$$\Delta t_r = F e (A)^{1/2} \sin E_k \quad (5)$$

F is a constant whose value is

$$F = -2 (\mu)^{1/2} / c^2 = - 4.442807633 (10)^{-10} \text{ sec} / (\text{meter})^{1/2} \quad (6)$$

While the coefficients a_{f0} , a_{f1} , and a_{f2} are generated by using GPS time as indicated, sensitivity of t_{sv} to t is negligible. This negligible sensitivity will allow the user to approximate t by t_{sv} .

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 .

2.2.7 Universal Coordinated Time (UTC)

Depending upon the relationship of the effectivity date to the user's current GPS time, the following three different UTC/GPS-time relationships exist [3]:

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

$$t_{UTC} = (t_E - \Delta t_{UTC}) \{ \text{Modulo } 86400 \text{ seconds} \} \quad (7)$$

Where:

- t_{UTC} is in seconds and
-

$$\Delta t_{UTC} = \Delta t_{LS} + A0 + A1 (t_E - \text{tot} + 604800 (WN - WNt)), \text{ (seconds);} \quad (8)$$

- t_E = 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;
- 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:

$$t_{UTC} = W [\text{Modulo } (86400 + \Delta t_{LSF} - \Delta t_{LS})], \text{ (seconds);} \quad (9)$$

Where:

$$W = (t_E - \Delta t_{UTC} - 43200) [\text{Modulo } 86400] + 43200, \text{ (seconds);} \quad (10)$$

and the definition of Δt_{UTC} 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.

3 Problem Solution

3.1 Implementation for the solution

The application is built around the Simple Document Interface (SDI) specification. The purpose of this application is to provide the user a tool to interact with map images. The images are TIF files. There are two 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 8)
- by tiles (see figure 9)

Both types are supported.

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.

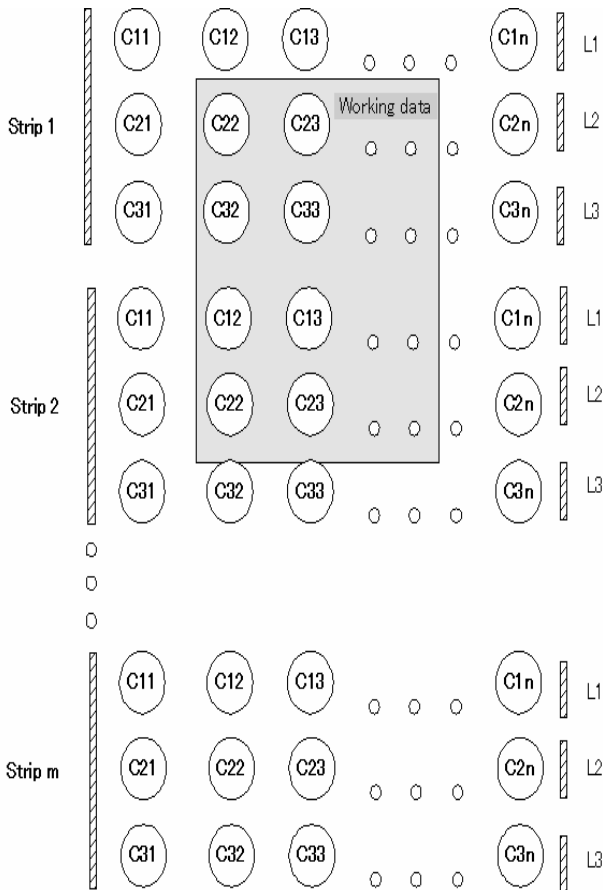


Fig. 8. TIF Strip Organization

In the above picture a “Cij” element is referred as the color element (pixel) at the row i and column j in a certain strip. Li is referred 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. Due to this reason, the TIF tile organization is preferred because is much faster.

The tile number increase 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 as color elements (pixels) in a tile.

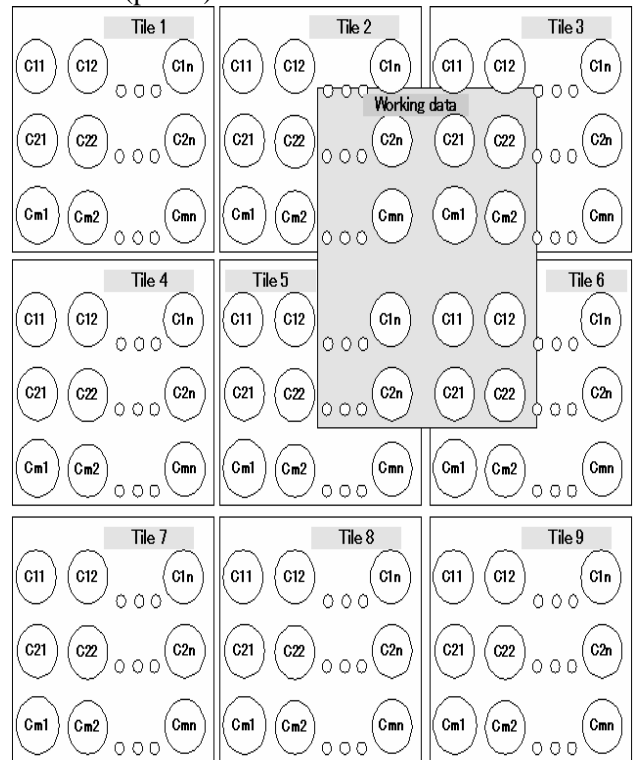


Fig. 9. TIF Tile Organization

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.

The application has the following main features:

- ❖ Load JMap - custom made TIFF images. These maps are obtained by exporting the regular TIFF images and by adding the contour lines data (height information). A special stand-alone application has been developed to encrypt the JMap images.
- ❖ JMap registration – the user loads a map by using the “File open” dialog. When the map is loaded for the first time the application saves all the data regarding the map in a special file (binary encrypted file) so the next time the application will run the user is no need to manually select again the map, he can see it in the mesh code where it belongs and directly load it. Also, the user can rename and un-register a jmap.
- ❖ Display the JMaps and provide functionality to manipulate this maps as normal TIFF

images (marker management, printing, saving map selections, etc). The information about the map coordinates are stored in the TIFF file tags and the application reads these data and uses it to provide the desired functionality.

- ❖ GPS navigation on JMaps – jmaps are totally integrated with the GPS module and can be used (if available) for GPS navigation as the 25000 or 200000 images are used. There is a synchronization mechanism between all these types of maps so the GPS position can be seen on any kind of map.

The following diagram shows different module components of the program.

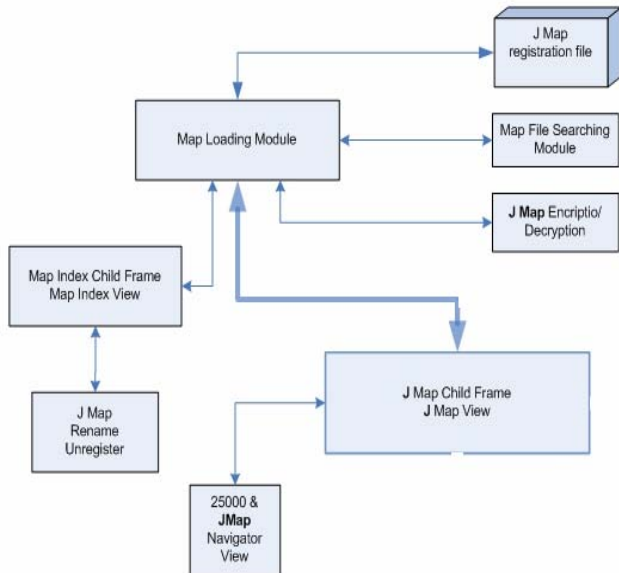


Fig. 10. Module for loading maps

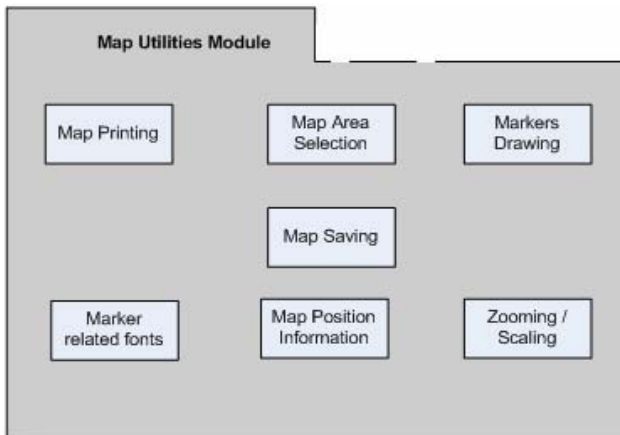


Fig. 11. Map utilities module

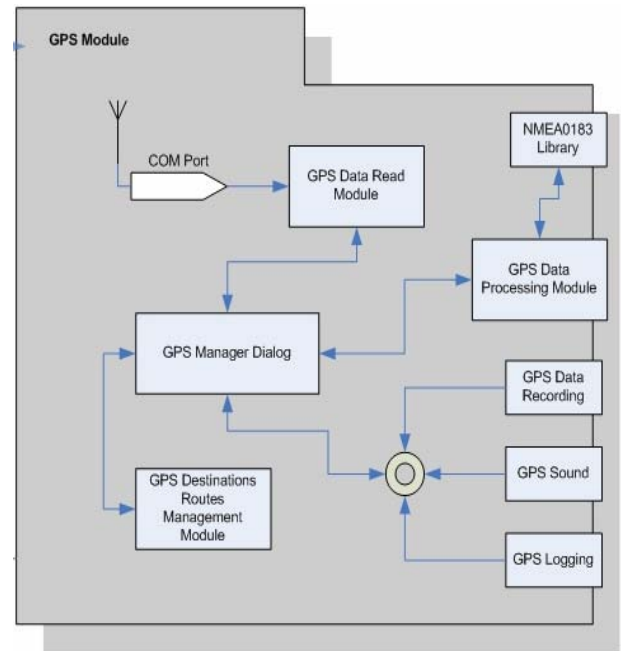


Fig. 12. GPS Module

The next three figures represent the main screens from application and are used for set GPS characteristics, display GPS information and locating GPS coordinates on TIF map.

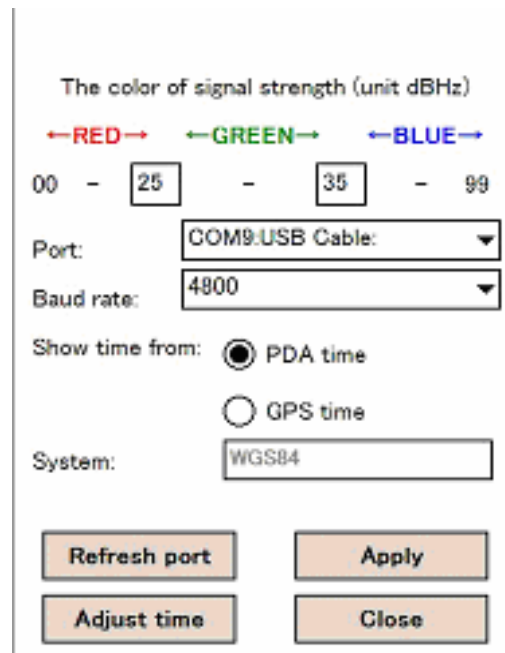


Fig. 13. Screen used for set GPS characteristics

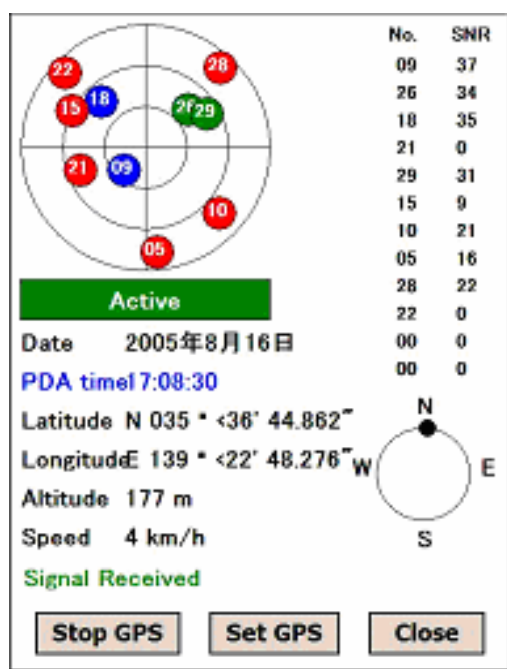


Fig. 14. Screen used for display GPS information

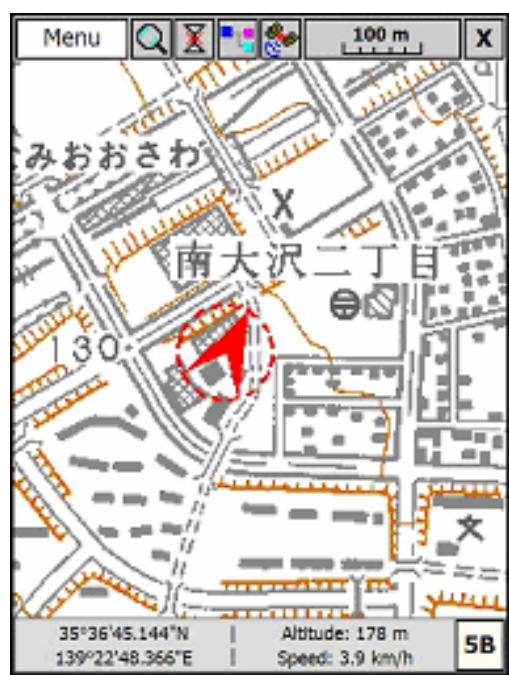


Fig. 15. Locating GPS coordinates on map

3.2 Testing the solution

The functional testing verification represents a part of the testing process of software applications. The functional verification has the following objectives: to verify the behavior of application, to detect the defects of software, to test robustness and the performance of application in operational nominal context and degraded context.

The verification strategy required for an item (symbol or object of interface) defines four categories of tests:

- Nominal – functional verification in nominal mode
- Degraded – functional verification in degraded mode
- Robustness – verification the functional robustness
- Performance – verification of performance

The model used to test the functionality of an application is IVV (Integration Verification and Validation) and the main method is IADT (Integration Analysis Demonstration and Test).

The objective of Integration is to assemble the software product from the components, ensuring that it functions properly.

The objective of Verification is to ensure that the final and intermediate software product satisfy their technical requirement.

Verification covers nominal operation, but also degraded operation and robustness, whether or not that specifically related requirement is included in the specification.

The objective of Validation is to determine that the software product satisfies the requirements in its intended operational environment.

The IADT method represents:

- Inspection is a visual verification of an item.
- Analysis is a verification based upon analytical evidences obtained by calculation, without intervention on the verified item. The techniques used are modeling, simulation and forecasting.
- Demonstration is a verification of operational characteristic observable by the overseeing of the functioning component without physical measurement.
- Test is a verification of functional characteristics that are measurable and directly or indirectly reachable.

The *nominal functional tests* verify the nominal behavior for an item in the context of valid data. In this test, we verify the graphical interface for all display states of item for nominal symbolism like position, colors, shape, text; logic of display; and the behavior for all operational values of inputs data.

The *degraded functional tests* verify the degraded behavior in the context of invalid data. In this test we verify the graphical interface for all display states for invalid symbolism; logic and behavior of display for invalid inputs data

The *robustness tests* verify how the operational behavior of an item is not degraded when the input data are out of operational context.

The *performance tests* verify how the operational behavior of an item is not degraded when the input data are forced to values limit.

For each item (symbol or object from user interface) we create a top test procedure with tests according to the described model presented above. Each top procedure contains intermediate procedures and each of these contains scenarios of test with one or many test cases.

In case that many items form a part of the interface, these are tested together. In some cases, one or many procedures or scenarios could be absent.

The graphical user interface that we refer to is depicted in the Figure 14, where we present the satellites' positions and the color of strength of their signals, data and GPS time, the coordinates (longitude, latitude and altitude), the pointing of walk and the list of satellites and their strength signal. The color of the strength signal is set from the configuration signal interface presented in the Figure 13.

For the graphical interface of the application from Figure 14 we apply the verification strategy model for certain symbols or graphical objects as latitude and longitude.

3.2.1 Latitude – top procedure

This symbol is a label, which indicates the latitude received from satellite.

Graphical description for valid state:

- "Latitude" in black color
- "N" or "S" black flag
- 3 black digits, right justified, leading zero, for degrees, followed by ° unit.
- 2 black digits, right justified, leading zero, for minutes, followed by ' unit.
- 2 black digits, right justified, leading zero, for seconds, followed by black decimal point, 3 black digits represent the thousandth of seconds and by " unit
- located in right of the diagram with satellites positions, below GPS time

Graphical description for invalid state:

- "Latitude" in black color
- 3 black dashes followed by ° unit.
- 2 black dashes followed by ' unit.
- 2 black dashes followed by point, 3 black dashes and " unit

The input parameter for latitude is *positionLat* that has the operational range $[-90^{\circ}00'00.000", +90^{\circ}00'00.000"]$ and the codable range $[-180^{\circ}00'00.000", +180^{\circ}00'00.000"]$

a) Valid state display and input increase in operational range (*Nominal and Regression procedure*)

i) Graphical display scenario

Test case no. 1: *positionLat* = +35°36'46.140"

Display in black "Latitude N 035°36'46.140" "

Test case no. 2: *positionLat* = -35°36'46.140"

Display in black "Latitude S 035°36'46.140" "

ii) Increase input in operational range

Test case no. 1: *positionLat* from -90°00'00.000" to -0°59'59.999" with step 0°00'00.001"

Display in black "Latitude S" and values according with evolution from -90°00'00.000" to -0°59'59.999".

Test case no. 2: *positionLat* from 00°00'00.000" to +90°00'00.000" with step 0°00'00.001"

Display in black "Latitude N" and values according with evolution from 00°00'00.000" to +90°00'00.000".

b) Input decrease in operational range (*Nominal procedure*)

i) Decrease input in operational range

Test case no. 1: *positionLat* from +90°00'00.000" to 0°00'00.000" with step -0°00'00.001"

Display in black "Latitude N" and values according with evolution from +90°00'00.000" to 0°00'00.000".

Test case no. 2: *positionLat* from -0°59'59.999" to -90°00'00.000" with step -0°00'00.001"

Display in black "Latitude N" and values according with evolution from -0°59'59.999" to -90°00'00.000".

c) Invalid state display for invalid input (*Degraded and Regression procedure*)

i) Invalid input

Test case no. 1: *positionLat* is invalid (not receive from satellite)

Display in black "Latitude ", the flag "N" or "S" is not displayed and 3 black dashes followed by ° unit, then 2 black dashes followed by ' unit, then 2 black dashes followed by point, 3 black dashes and " unit.

Like that: "Latitude ---°--'--.--" "

- d) Input increase outside operational range
(*Robustness and Regression procedure*)
- i) Increase input outside operational range
Test case no. 1: *positionLat* from -180°00'00.000" to -90°00'00.001" with step 0°00'00.001"
Display in black "Latitude S 090°00'00.000" "
Test case no. 2: *positionLat* from +90°00'00.001" to +180°00'00.000" with step +0°00'00.001"
Display in black "Latitude N 090°00'00.000" "
- e) Input decrease outside operational range
(*Robustness procedure*)
- i) Increase input outside operational range
Test case no. 1: *positionLat* from +180°00'00.000" to +90°00'00.001" with step -0°00'00.001"
Display in black "Latitude N 090°00'00.000" "
Test case no. 2: *positionLat* from -90°00'00.001" to -180°00'00.000" with step -0°00'00.001"
Display in black "Latitude S 090°00'00.000" "

3.2.2 Longitude – top procedure

This symbol is a label, which indicates the longitude received from satellite.

Graphical description for valid state:

- "Longitude" in black color
- "E" or "W" black flag
- 3 black digits, right justified, leading zero, for degrees, followed by ° unit.
- 2 black digits, right justified, leading zero, for minutes, followed by ' unit.
- 2 black digits, right justified, leading zero, for seconds, followed by black decimal point, 3 black digits represent the thousandth of seconds and by " unit
- located in right of the diagram with satellites positions, below Latitude

Graphical description for invalid state:

- "Longitude" in black color
- 3 black dashes followed by ° unit.
- 2 black dashes followed by ' unit.
- 2 black dashes followed by point, 3 black dashes and " unit

The input parameter for latitude is *positionLong* that has the operational range and also the codable range [-180°00'00.000", +180°00'00.000"].

- a) Valid state display and input increase in operational range (*Nominal and Regression procedure*)

- i) Graphical display scenario

Test case no. 1: *positionLong* = +139°22'50.082"

Display in black "Longitude W 139°22'50.082" "

Test case no. 2: *positionLong* = -139°22'50.082"

Display in black "Longitude E 139°22'50.082" "

- ii) Increase input in operational range

Test case no. 1: *positionLong* from -180°00'00.000" to -0°59'59.999" with step 0°00'00.001"

Display in black "Longitude E" and values according with evolution from -180°00'00.000" to -0°59'59.999".

Test case no. 2: *positionLong* from 00°00'00.000" to +180°00'00.000" with step 0°00'00.001"

Display in black "Latitude W" and values according with evolution from 00°00'00.000" to +180°00'00.000".

- b) Input decrease in operational range (*Nominal procedure*)

- i) Decrease input in operational range

Test case no. 1: *positionLong* from +180°00'00.000" to 0°00'00.000" with step -0°00'00.001"

Display in black "Longitude W" and values according with evolution from +180°00'00.000" to 0°00'00.000".

Test case no. 2: *positionLong* from -0°59'59.999" to -180°00'00.000" with step -0°00'00.001"

Display in black "Longitude E" and values according with evolution from -0°59'59.999" to -180°00'00.000".

- c) Invalid state display for invalid input (*Degraded and Regression procedure*)

- i) Invalid input

Test case no. 1: *positionLong* is invalid (not receive from satellite)

Display in black "Longitude ", the flag "W" or "E" is not displayed and 3 black dashes followed by ° unit, then 2 black dashes followed by ' unit, then 2 black dashes followed by point, 3 black dashes and " unit.

Like that: "Longitude ---°--'---,---" "

Since the operational range is identical to codable range, we don't have Robustness test procedures.

4 Conclusion

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.

In order to find the correct coordinates for a point given by its latitude and longitude on a TIF map there were identified three types of problems that might arise:

- sometimes, the maps have a trapezium and have to be brought to a rectangular form
- not always the limits of the maps is defined by two gray pixels and this limit has to be found in order to correctly localize the points
- as the maps are very large, having over 10 MB usually and because the PDA has limited memory, two algorithms that use the maps width were chosen to be implemented so that if the width won't exceed 1024 pixels the strip organization is chosen, else the tile organization is used.

The testing of software programs is done for two reasons: detecting errors and estimating reliability.

References:

- [1] HowStuffWorks. *How GPS Receivers Work*.
- [2] Rizos, Chris. University of New South Wales. *GPS Satellite Signals*. 1999.
- [3] Physics Today. *Relativity and GPS*. May 2002.
- [4] Phrack. *Issue 0x3c (60), article 13*. December 28, 2002.
- [5] *Simplified explanation of GPS at howstuffworks.com*
- [6] *GPS SPS Signal Specification*, 2nd Edition — the official (civilian) signal specification.
- [7] University of New Brunswick, *In Simple Terms, How Does GPS Work?*
- [8] Enescu N., Dumitraşcu E., Ivan I., Siniros P., The Classification of Software Testing Techniques, *The Proceedings of the Seventh International Conference on Informatics in Economy*
- [9] Roper, M., *Software testing*, McGraw–Hill, 1994
- [10] Sommerville, I., *Software engineering*, Addison–Wesley, 1996