Find The Place – Extended

March 18, 2007

Contents

1 Introduction

This document describes a heavily extended version of the *Find The Place* J2ME Midlet originally written and released by *Per Kofod Hansen* for Motorola A920/A925 mobile phones.

This version is mainly tailored and targeted towards the Motorola E1000 3G mobile phone, but may well run on other handsets which provide the appropriate APIs (specifically **com.motorola.location** as well as **com.motorola.io**).

Find The Place is a J2ME Midlet that accesses the internal GPS of a (Motorola) mobile phone to provide the user with some basic location-finding functionality, much like a first generation GPS receiver (GPSr) would.

This modified version of *Find The Place* provides additional capabilities on top of that, using mobile data access to retrieve map data provided by *Google Maps* for either a specific given location or the current location as determined by GPS. In that regard it performs a job similar to MGMaps by *Cristian Streng*.

1.1 A few words on GPS in general

There are an amazing number of misconceptions around when it comes to GPS. Most people in this day and age think of GPS as something you just turn on and then you'll know just where you are, no matter where you are. And indeed, most modern day GPS appliances will actually seem to perform like that.

This, of course, is due to a wide variety of reasons. Improvements in GPSr sensitivity and antenna efficiency are just two of the main factors helping current units to achieve this almost magic-like quality, thus obscuring the fact that GPS is highly complex, almost fragile system when it comes to reliability.

GPS is based on a fairly large number of earth satellites in low orbit, constantly broadcasting a complex data stream towards earth. By receiving and evaluating this data stream from several satellites at a time, a GPSr can calculate the precise geographical position it is currently located at.

So far, so good – but the frequencies used to transmit the data were chosen as to be suitable for fairly low powered transmissions and be fairly undisturbed by atmospheric phenomena. In other words, unless you have a direct line of sight to the satellite, you won't be able to catch its signal. Clouds, rain and snow usually don't matter, but as soon as something with somewhat more substance comes between the receiver and the satellite, that's it – no more signal. Signals can *only* be received from satellites which are currently visible in the segment of sky that can be seen from the receiver's point of view.

Now, to get an initial position (a so called *first fix*) after powering on a GPSr, the unit needs the signals from four (!) different satellites at the very same time. Once the *first fix* has been performed, all subsequent fixes can be performed with the signals from just three satellites.

This usually should not be a big problem, as there are quite a number of those *Navstar* satellites orbiting the globe – around 30 are operational these days. One drawback is that they are not geostationary, that is they are moving across the sky constantly. So at a specific point on earth, satellites will rise and set all the time, so the visible satellites will change continuously over time, and their virtual position in the sky above will change as well.

In a perfect world (that is one without valleys, hills, mountains, buildings and even people) that wouldn't be a problem. But in reality, there are always parts of the sky that are not visible – and the more urbanized an area is, the less you will usually get to see of the sky. This is where the problems start.

The typical number of satellites visible in the *entire* sky at a spot is something between 6 and 10, with rare exceptions when there's even two or so more or less. And even that's not entirely true, since they're not really in the *entire* sky. There is a large, circular "blind spot" in the sky which will never ever contain any GPS satellites. In the northern hemisphere this spot is north, with an elevation that rises with the latitude of the observer's position, and of course vice versa in the southern hemisphere.

Beginning to see the problem yet? Imagine going to a window of your house or flat. How much of the sky do you see? Not to mention the fact that it might be facing north! Imagine what the number of visible satellites might be. I'll give you a hint: *not nearly enough*! Think about standing outside of your dwellings. In urban areas you will usually have sky just right above you, starting around an elevation of, say, 30-45°. If you're close to a building, it may be even worse, at least for one half of the sky.

To make things even more difficult, the receiver needs to know where the individual satellites currently are. This can be calculated from the orbital parameters. Problem is that these are constantly undergoing minor changes, so they have to be updated regularly. They fall in two major groups: parameters that change slowly and stay approximately correct over several weeks to several months. They are referred to as the *almanac*, whereas the parameters that are valid for only several hours are known as the *ephemeris*.

Without this data, the GPSr can not perform its function. So the data is provided within the data stream transmitted by the satellites. But due to the fact that the signals are so weak, the data rate available is a measly 50 Baud. This makes updating the ephemeris and almanac quite a lengthy process – 30 seconds to 2 minutes for the ephemeris and 10 to 15 minutes (!) for the almanac, not even taking reception errors into account. This problem is aggravated by the fact that the signal strength needed for download is significantly higher than the one needed for acquiring fixes.

All of these factors tend to come together when trying to work with GPS, not exactly making things easy. Keep them in mind and act accordingly before blaming “that stupid GPS”.

1.2 GPS and the E1000

The Motorola E1000 3G mobile phone features an integrated A-GPS receiver. The “A” stands for *Assisted*. And as with GPS in general, there are a huge amount of misconceptions when it comes to Assisted GPS.

Some people think that A-GPS is a stripped-down version of GPS, not providing full GPS capabilites, thus claiming it’s not a “real” GPSr. While this *may* be true, it needn’t necessarily be that way.

First and foremost, “assisted” means that external data can be supplied to the GPSr to improve or speed up the *Time To First Fix* (TTFF). Imagine a GPSr that is rarely used. And all of a sudden, you need to know your location. In traditional (*autonomous*) GPS operation, this process may easily take 20 to 30 minutes, since the unit has neither a current almanac, current ephemeris or even an approximate idea of where it currently is located.

By providing almanac and ephemeris data as well as an approximation of its location to the GPSr externally, this process can be speeded up to an amazing few seconds. In addition DGPS (*Differential GPS*) information can be used to further improve precision of the fix. In this *assisted* mode, the data needed is provided via the mobile network when needed. Problem: the network operator has to specifically support this. If this is not the case, assisted operation is not available.

In addition, it is conceivable that the A-GPSr itself is incapable of performing the necessary calculations and will transmit the received data to a server on the mobile network to do this task on behalf of the GPSr. These units are completely incapable of autonomous operation and can indeed be regarded as stripped down and inferior.

Thankfully the Phoenix GPS Acquisition Module (GAM) used in the Motorola E1000 is capable of autonomous as well as assisted operation. The former is proven to work with recent releases of the firmware (R26LD), if the necessary requirements are met (i.e. AGPS available and activated using the appropriate tools and menu settings).

But even then, using the GPS feature of the phone is not an easy task. For one thing, the only way to access the functionality is by using a J2ME Midlet, which in turn can only use Motorola's proprietary API to do so. And as if this wasn't bad enough, a Midlet doing so needs special permissions that cannot be granted by the user without the aid of specific tricks and/or tools.

So Motorola didn't make it easy for users to take advantage of the built-in GPSr. And who knows, maybe they had a very good reason for this. They most likely knew from the start what users have found out the hard way by now: the E1000 has an *extremely* lousy GPS antenna.

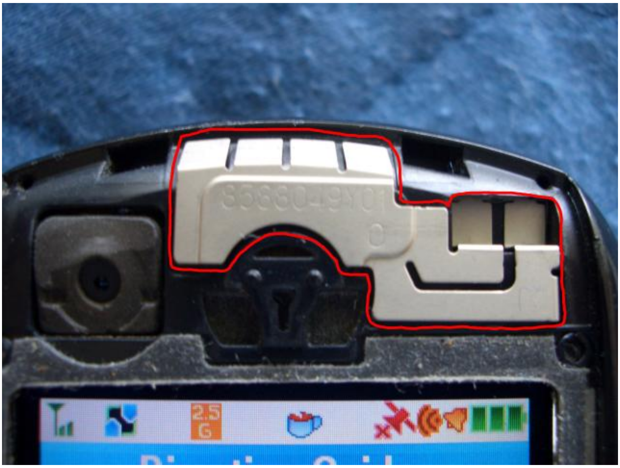


Figure 1: The E1000's GPS antenna: a certified joke.

Instead of opting for a decent embedded quadrifilar helix or at least an acceptable microstrip patch antenna, the engineers incorporated a piece of bubble gum wrap on steroids into the design (see Figure 1 on page 7). This of course makes GPS operation with the E1000 *without* the use of an external GPS antenna (the proprietary connector for which can be found top right on the backside of the phone – see Figure 2 on page 8) tiresome at best.

Add to that the fact that in virtually all E1000s the GPSr has been dor-

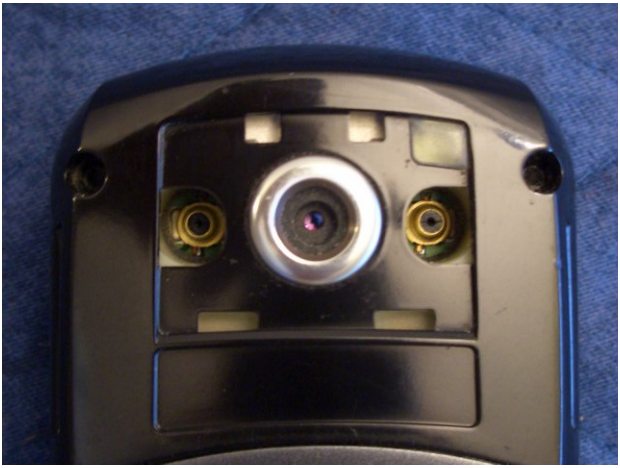


Figure 2: External antenna connectors on the E1000.

mant since the phone left the factory, and you are getting a nice recipe for disaster. The almanac data provided in production is seriously outdated by now, so the GPSr needs to cold start. As they say on *French Maid TV*: “Zis may take a vile, so find zomesing else to do.”

Nonetheless, it *is* possible and, once the phone has access to current data, even works fairly well in somewhat open terrain. So come on – do you *really* want to look that gift horse in the mouth?

1.3 Some History

First reports of people finding a way to make use of the E1000’s builtin GPSr popped up in late October 2005, with *EmGPS* being the first proof-of-concept Midlet. At the same time, people made use of *Find The Place*, a Midlet written some time before by *Per Kofod Hansen* for his Motorola A920 mobile phone (which, incidentally, uses the same proprietary location API as provided on the E1000).

Some people were also toying with *Locator*, a J2ME application published by *Geomaia Ltd.* for the Motorola A925 handset. But since it was shareware selling around $25, it never seemed to really catch on within the E1000 community.

Once the news spread, a lot of people showed interest in gaining GPS functionality for their phones. Probably most of these people never used (let alone owned!) a GPSr before in their life, thus not understanding the complex nature of the system as described in section 1.1 of this document. Add to that the facts about the antenna and the outdated almanac as well as the fact that setting up the necessary access rights for Midlets accessing the GPSr was a non-trivial task. And to top it all off, the reliability of GPS operation depends on the firmware release installed on the phone.

This, of course, led to the fact that only a few people managed to actually make things work, while most never really got even close. And even those who got lucky were *extremely* unhappy about the lack of reliability. At one point it would work, just to stop working for days or weeks after that. Then, out of the blue, it would start working again. So most people simply gave up on it, and just a precious few kept tinkering.

In late September 2006 *Cristian Streng* added support for using the **com.motorola.location** API to his excellent *Mobile GMaps* application. So from now on there was actually something remotely related to a navigation system available for Motorola phones. This in turn somewhat intensified the efforts to improve reliability of the E1000's GPSr.

External antennae were tried. The almanac data file found in the file system watched and analyzed. Theories were created and discarded as to what the main problems were. People using the GPS feature frequently reported that it became more reliable over time. Further analysis showed that the whenever the phone was performing GPS operations it would, albeit slowly, update its almanac. In the end it looked as if feeding a complete current almanac data to the GPSr would help things along a great deal.

Problem was that simply copying a file containing almanac data to replace the one already on the phone wouldn't work, even though the format was known to be correct. The phone would simply replace the new data with the previous contents again.

In early 2007, *k500zm* (a really smart cookie) finally figured it out how to make the externally fed almanac data stick. By feeding the almanac file to the phone using a process known to Motorola modders as *flexing*, it is accepted by the handset.

And thus it turned out that once the phone has a fairly current almanac, reliability increases dramatically. And even with its ridiculous antenna, TTFFs of 30s to 2 minutes are nothing unusual upon warm start – provided that the requirements for GPS operation as pointed out in section 1.1 are met.

And while I don't recommend making your life depend on it, the GPSr feature of the Motorola E1000 3G mobile phone has finally become somewhat usable and useful.

1.4 The Application

So why this application instead of the aforementioned usual suspects, *Mobile GMaps* and the original *Find The Place*? Well, for one thing *Find The Place* always seemed to be more reliable in producing fixes, and the source code was readily available for tinkering.

Add to that the fact that data charges for map downloads were (and still *are* in many parts of the world, especially when roaming) prohibitively high and that sometimes traditional navigation using waypoints and direction is preferable to a moving map display (e.g. when hiking or hunting for a Geocache), and you have more than enough reasons to try and roll your own application in order to get the best of both worlds.

This was the original motivation in extending the functionality of *Find The Place*. Combining simple navigation with maps. Not having to pay exorbitant amounts of money for downloading the same map tiles over and over again. Being able to get an on-demand map of your immediate surroundings at the push of a button, without the need of a GPS application constantly downloading map tiles. You get the drift.

So at this point, *Find The Place – Extended* (a.k.a. *My Find The Place*), Release 2.2.X features:

- A clean *Position* screen, which gives all location, status and waypoint information without the need for scrolling around.
- A new and redesigned *Direction* screen featuring a large, easy to see direction indicator as well as some additional information like course deviation and *Estimated Time Enroute* (ETE).
- Easier waypoint selection and handling, as well as the ability to edit waypoint information.
- Support for controlling the display backlight from the application in order to improve battery life.
- An averaging mode to improve precision while standing still.
- Support for writing timestamped position information in CSV format to a file at configurable intervals, thus creating a track log that can easily be further processed at a later point in time.
- A *Map* screen fed online by *Google Maps* with an additional underlying two level cache mechanism. Downloaded tiles will be cached in RAM and in the file system, given correct configuration.
- A switchable trailplot and waypoint overlay for the map display.
- Easy switch between static and moving map display, normal and full screen display as well as zooming in and out of the map.
- Manual panning through maps.
- Automatic or on-demand loading of missing map tiles.

. . . and more. Intrigued? Then maybe *Find The Place – Extended* is the GPS application for you.

2 Installation

There are probably more ways to install a J2ME Midlet on a Motorola phone than there are ways to skin a cat. Which one is the best for you depends on an innumerable amout of factors and is something that is beyond the scope of this manual. Chances are that unless you got your phone very recently, you already figured out the best way for you to do it. So just do whatever you feel most comfortable with.

If you need a recommendation, use *MotoMidMan*. Why? Since it will not only be able to install the Midlet to your phone, but in addition provides an easy way to perform the required next step.

Ed. note: more needs to be written. Screenshots needed.

2.1 Granting Full Access

Since *Find The Place – Extended* makes use of proprietary Motorola APIs which can only be invoked if the Midlet is given sufficient privileges, an additional step is needed so that the application can access the GPSr and the file system.

This can either be done the traditional way, by downloading internal Java files, patching them using a tool known as *MotoMu* and re-uploading them (for a long time this has been the only way to grant full rights to a midlet – no longer recommended unless you know what you are doing) or by using *MotoMidMan*, which is able to grant full privileges to a Midlet with a simple mouse click.

Ed. note: more needs to be written. Screenshots needed. Explain the Background option if you want it to be able to run in the background.

2.2 Updating The Almanac

Strictly speaking, this step is not really part of the installation procedure and technically not necessary. But it is highly recommended if you haven't used the GPS feature of your phone before or have been very dissatisfied with the time it takes to get the initial fix.

Ed. note: more needs to be written. Possibly screenshots are needed. Explain how and where to get an updated SEM almanac and use the sem2moto tool to convert it to Flex format as well as flex the updated almanac using RSD Lite.

3 Using The Application

3.1 Before you start

Before running the application for the first time, it is highly recommended to create a directory structure in the phone's file system (preferably on the *MicroSD / Transflash* card) to hold the data files created by the application (i.e. track logs and cached map tiles).

The defaults assumed by the application are **/b/GPS** (a folder named "GPS" in the root of your storage card) for the folder in which track logs are created and **/b/GPS/tcache** (a subfolder named "tcache" within the aforementioned "GPS" folder) as base folder for the tile cache.

This base folder in turn needs to contain 18 subfolders for the individual zoom levels provided by Google Maps. These subfolders must be named **z0**, **z1**, **z2** . . . **z15**, **z16** and **z17**.

These folders can either be created by connecting the flash card to a computer and performing the necessary steps there, by hooking up the phone using the USB cable and using something like *p2kcommander* or *moto4lin* to access the file system or on the phone using another Midlet capable of doing so (e.g. a modified *Opera Mini* or *FileCommander*).

You may create these directories someplace else, since the path for the track log folder as well as the tile cache base folder can be configured using the application's setup dialog (see section 3.5.4 on page 27 for details). It may even possible to use the internal phone memory. Changing the defaults is highly discouraged, though.

WARNING!

FAILURE TO CREATE THE NECESSARY DIRECTORIES WILL RESULT IN MALFUNCTION OF THE RELATED FEATURES (TRACK LOG, SECONDARY TILE CACHE). DOUBLE CHECK THAT THE ENTIRE DIRECTORY STRUCTURE EXISTS AND IS PROPERLY SET UP!

Once you have performed all the aforementioned preparatory steps and have made sure that the necessary folders are all accounted for, you can the proceed to start the application for the first time.

3.2 The Main (Position) Screen

Whenever you start the application, it will display this screen (see Figure 3 on page 19). This screen provides basic status, location and waypoint information. In particular, these are (from top to bottom):

- GPSr Status

  This indicates the current status of the internal GPSr, based on the status codes returned by the **com.motorola.location** API. This will change frequently in order to give some insight into GPSr operation. Messages you are likely to see during normal operation are:

  - initializing

    You will see this message only very briefly after startup, if at all. It is shown while access to the GPSr is established and the acquisition is initialized. Usually it is immediately replaced by one of the following messages.

  - unavailable

    This indicates that the GPSr is operating, but is either not yet able to acquire a signal (still trying to download ephemeris data after startup) or there has been no signal available at all for some time (trying to reacquire a signal).

  - no response

    This message shows that the location API is still busy processing the last call and has not returned with a result yet – so this is more or less an informative message to allow some insight into the internal workings of the acquisition thread.

  - timeout XXs

    A message occasionally shown to let the user know how long the application has been trying to get a fix without a positive result. So this will give the number of seconds since startup or since the last fix was acquired.

- not enough satellites

  Indicates that there are GPS signals available and/or that current ephemeris data is available to the GPSr, but there haven't been enough visible satellites available for evaluation to produce a position fix.

  While this may sound like a bad thing, this message is one of the very few indications that things are working and indeed moving along after startup. Usually the application will go from *initializing* right to alternating between *no response*, *unavailable* and *timeout XXs*. Then, after some time, *unavailable* will be replaced with *not enough satellites*, showing that signals are indeed available and the ephemeris data downloaded.

  Given an up to date almanac and a sufficient number of available satellites in the visible portion of sky, this process to this point shouldn't take longer than two minutes, three at most. If there is no progress within that time, you most likely have an unsuited spot or a bad time for the spot you're at. Either try again later or move to a location in which the sky is less occluded.

  Should the almanac data in the phone be outdated, all bets are off, though. In that case it may take a long time (even hours) to get to this point. But when you do, you'll at least know that you're getting a closer to a fix.

- fixed ±XXm

  This of course is the one message you are waiting for. It shows that the GPSr was able to produce a position fix and gives an estimate on the precision of it. Note that this *Dilution Of Precision* (DOP) estimate does not mean that the position caclulate will indeed be that precise or may be that far off. It just tries to tell the user that there is a high probability his location might be within the indicated radius around the position shown.

  So this needs to be take with a grain of salt – sometimes it will say *fixed ±57m* and be right on the spot, at other times it will claim *fixed* ±*3m* and be off over 50m. So when in doubt, try to get multiple fixes for the same point at different times of day. Using *Averaging* may also be an option.

There are two distinct status messages that you may get to see that, depending on the circumstances are accidental (thus indicating you made a mistake) or intentional (if you produced the condition on purpose). These are:

- disabled

  This will be shown if the AGPS feature of the phone has been disabled by the user in the phone's menu (*Settings – Phone Status – AGPS Service*). If this is intentional, regard it as an informative message. Otherwise quit the application, enable AGPS and restart the application.

- open failed

  Connection to the GPSr using the location API could not be established. The reason for this almost always is that the Midlet does not have the access rights needed for doing so. This is invariably due to errors while installing the application. See section 2.1 on page 12 for details. Follow the necessary steps described and then restart the application.

The remaining messages are some you should never get to see, unless there is something seriously wrong either with the hard- or software of the phone. They are listed here mainly for completeness' sake:

- not found

  The GPSr could not be found. Someone possibly diked it out while you weren't watching your phone.

- receiver error

  The GPSr detected an error while going about its business.

- internal error

  Uhm. Yeah. No idea when this might possibly pop up.

- status unknown

  The location API returned a status code unknown to the application.

- Position

  These two lines give the WGS84 coordinates (Latitude and Longitude) as reported by the GPSr. The format this is reported in is subject to the choice made in the application's setup. See section 3.5.4 on page 27 for details.

  An asterisk (*) displayed behind the coordinates indicates that *Averaging* is enabled and the position reported is not the result of the last fix but rather an average position calculated from all fixes since averaging mode has been turned on.

- Elevation

  Whenever there is a 3D fix available, this will give the altitude of the current location in metres above sea level. This is known to be inaccurate especially with poor reception and/or satellite constellations with a high DOP.

  As with the *Position* fields, an asterisk (*) means that *Averaging* is active and that the given altitude is not the result of the last fix but rather an average altitude calculated from all fixes since averaging mode has been enabled.

- Speed and Heading

  While moving, this gives the current speed in km/h at which the unit is moving, as well as the compass direction it is moving in. Note that both of these values may be fairly inaccurate at slow speeds. Occasionally the GPSr will report these values even if the unit is stationary, in which case they will of course be completely random and meaningless.

- Destination

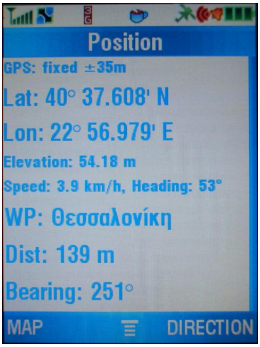
  This shows the name of the currently selected destination waypoint as specified when the waypoint was stored. When first starting the application, a hard-coded default is used for the current destination. This is likely to change in the future.

- Distance

  Indicates the current distance to the selected destination waypoint in metres. This of course is the distance "as the crow flies", not the actual distance you may have to travel on the ground.

- Bearing

  The compass direction the selected destination waypoint can be found in, with the scale starting with 0° for true north, running clockwise, wrapping around at 360°.



Position

GPS: fixed ±35m

Lat: 40° 37.608' N

Lon: 22° 56.979' E

Elevation: 54.18 m

Speed: 3.9 km/h, Heading: 53°

WP: Θεσσαλονίκη

Dist: 139 m

Bearing: 251°

Figure 3: Example of the main application screen.

For access to the other features the application provides, there are three function keys active in the Main/Position Screen:

- The right softkey (marked “DIRECTION”) takes you to the Direction Screen (see section 3.3 on page 19).
- The left softkey (marked “MAP”) will take you to the Map Screen (see section 3.4 on page 21).
- The center key will display the Main Menu, as described in section 3.5 on page 25.

3.3 The Direction Screen

This screen can be reached from the Main/Position Screen by pressing the right softkey (marked “DIRECTION”). It is intended as a visual navigational aid while *moving* towards a waypoint. Due to the nature of the data provided by the GPSr, it more or less becomes meaningless while the unit is stationary.

The screen consists of the following items:

- Compass

  A graphical visual navigation aid styled after a compass. Note however that in fact it is *not* a compass. There are three indicators present within this element:

  - Dotted black line

    Indicates the direction you're currently moving in ("Heading", as numerically displayed on the Main/Position Screen). This can in fact be used to find true north. By rotating the unit so that this indicator is pointing straight in the direction you're moving in, the top of the display (i.e. the 0° mark) will be facing towards true north.

  - Solid purple line

    Gives the direction the currently selected destination waypoint can be found in ("Bearing", as numerically displayed on the Main/Position Screen). If you rotate the unit until the top of the display (again the 0° mark) points towards true north, this indicator will point exactly into the direction of your destination.

  - Red/Blue "compass" needle

    The main indicator of this element. The red end of the needle will show the current deviation (i.e. the difference between "Heading" and "Bearing"). That way if you hold the unit in a way that the top of the display is pointing in the direction you're moving in (which obviously will be most of the time), the red end of the needle will always point towards your destination as long as you are moving.

- Deviation

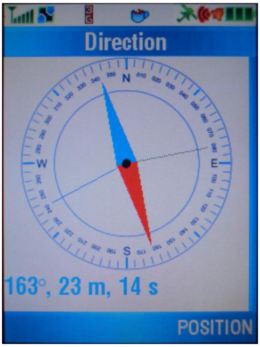
  This is the numeric equivalent of the value shown by the red and blue "compass" needle. It is provided in case a more precise readout is desired.

- Distance

  Current distance to the selected destination waypoint in metres, as also displayed on the Main/Position Screen. Repeated here for convenience, so that constant switching between the screens is not necessary.

- ETE

  Estimated Time Enroute will give you an estimate of the time remaining until you reach your destination, based on the current distance and speed. This of course is only useful if you are actually moving straight in the direction towards your destination.



163°, 23 m, 14 s

Figure 4: Example of the direction screen.

The Direction Screen provides no further access to other features of the application. The only selectable user function available is the return to the Main/Position Screen by pressing the right softkey (marked “POSITION”).

3.4 The Map Screen

This screen can be reached from the Main/Position Screen by pressing the left softkey (marked “MAP”). It is intended to provide the user with a basic map-browser capable of accessing maps provided by the *Google Maps* service.

In addition, pinpointing the current location is supported as well as a graphic overlay indicating the location of waypoints and the track traveled (if the track log feature is active).

By default, displaying maps is handled in a *very* restrictive manner as to avoid excessive data charges caused by the application constantly downloading map tiles. So unless configured otherwise (see section 3.5.4 on page 27), the application will download map tile *only* if prompted to do so by user interaction.

Ed. note: a lot more needs to be written, especially about tile cache handling, the display of missing tiles, indication of download failures and the Update In Progress indicator.

A number of options are available to the user in the Map Screen. These options are selected by pressing the appropriate keys while the Map Screen is active. Key assignments are as follows:

- Joystick / 2 4 6 8

  The joystick as well as the number keys 2, 4, 6 and 8 can be used to pan around the map by changing the current map location (indicated by the location marker in the centre of the map screen and the position overlay).

- Fire / 5

  By either pushing down the joystick or pressing the number key 5, the automatic refresh of the current location is toggled. When on, the map window will update the current map location whenever a new position fix becomes available.

  This effectively provides the user with a “moving map” that always shows the immediate surroundings while traveling.

  It is *not recommended* to combine this with the waypoint/trailplot overlay, since this would seriously slow down the refresh rate of the map display.

- * / #

  These two keys control the zoom level or map scale of the display. Zooming in is performed by pressing the # key (effectively showing a smaller area in more detail), whereas pressing the * key will zoom out (showing less detail of a larger area).

  Currently the *Google Maps* service provides 18 zoom levels.

  Once the minimum (or maximum) level is reached, further requests to zoom out (or in) are simply ignored.

- 1

  Toggles full screen mode. Usually, the bottom of the screen contains the labels for the currently active softkeys. This of course takes away quite a bit of display area that could be used for displaying more of the map – at the expense of an extra keystroke to access the softkey functions.

  So by pressing the number key 1, the map display will switch from the regular display to full screen mode and vice versa back to regular mode while full screen mode is active. To access the softkey functions while in full screen mode, press the centre softkey first to make your choice.

- 0

  Pressing the number key 0 will toggle the display backlight on or off. Since display illumination is one of the major factors in battery life, it is recommended to turn it off whenever possible to conserve energy, thus maximizing battery life.

- (7)

  (Shows the currently remaining free memory for debugging purposes as an overlay in the upper lefthand corner. Will be removed eventually.)

- 9

  The waypoint/trailplot overlay can be toggled on and off by pressing the number key 9. It is discouraged to activate this function while the automatic refresh feature is active, as it seriously slows down the continuous update of the map.

- SEND

  Pressing the SEND key will force the download of map tiles missing in the current map display. This is not needed if the application has been set up to automatically download tiles whenever necessary (see section 3.5.4 on page 27 for details).

The position indicator in the centre of the Map Screen consists of two parts: the position indicator itself, which is either shown as a small red bullet (while the unit is stationary) or a triangle (while moving) and a red circle, giving a graphical indication of the current precision (DOP) estimate. In other words, it is very likely your current position is somewhere in the area within the circle, whereas the position indicator itself gives the location of the last position fix). If the precision given is higher than can sensibly be displayed at the currently chosen map scale, the circle is omitted.

As mentioned before, the position indicator will change into a triangle if the unit is moving at a sufficient speed, with the acute angle of the triangle indicating the direction of travel. The accuracy of the direction shown will depend on the current travel speed. At slow speeds the result may be fairly inaccurate to the point of being virtually random.

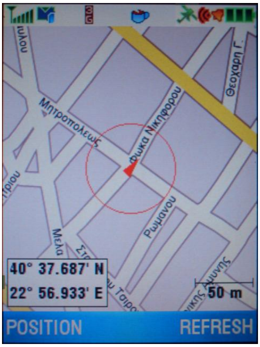


Figure 5: Example of the map screen.

To refresh the current map position using the last position fix provided by the GPSr, press the right softkey (marked “REFRESH”). To exit the Map

Screen and return to the Main/Position Screen, press the left softkey (marked “POSITION”).

If full screen mode is active, these sofkeys are disabled. Use the centre softkey instead to access the menu and make the desired choice from there.

3.5 Main Menu

When pressing the center softkey while the Main/Position Screen is displayed, the Main Menu is opened. It provides the following selections:

1. Waypoints

   Opens the *Waypoints* list (see section 3.5.1 on page 26 for details).

2. Light On/Off

   Toggles display illumination on and off (see section 3.5.2 on page 26 for details).

3. Averaging On/Off

   Toggles *Averaging* mode on and off (see section 3.5.3 on page 27 for details).

4. Setup Application

   Opens the application’s Setup Screen (see section 3.5.4 on page 27 for details).

5. Exit

   Terminates the application.

Ed. note: more needs to be written?

3.5.1 Waypoints

A *Waypoint* is an entity describing a location, typically consisting of at least a pair of coordinates and a name. The application is able to make use of these as navigational points, giving direction and distance as well as ETE to such a point if selected as current destination.

In addition waypoints can be used as bookmarks and shown as an overlay to the map display. This allows for fast and precise selection of a specific map area even without using GPS.

Ed. note: more needs to be written?

1. Show on Map
2. New
3. Edit
4. Delete

Ed. note: explain Select/Position actions.

3.5.2 Light On/Off

Toggles the display illumination on and off. Note that the backlight is *the* main consumer of electrical power. So in order to maximize battery life while using the application, try to turn it off as often and as long as possible.

Ed. note: more needs to be written?

3.5.3 Averaging On/Off

Toggles *Averaging* mode on and off. This is used to acquire more precise coordinates by averaging fixes over a long period of time while the unit is stationary.

Ed. note: needs to be written?

3.5.4 Setup Application

- Position Format
- Tracklog
- Log Interval
- Log Directory
- Map Zoom
- Map Download
- Map Autorefresh
- Cache Directory
- GPS Timeout
- GPS Accuracy
- GPS Age

Ed. note: more needs to be written.

3.5.5 Exit

Ed. note: needs to be written.

4 Troubleshooting

Ed. note: needs to be written.