Magellan

eXplorist

Communication Specification

Revision 0.7

August 30, 2005

P/N 600644

TABLE OF CONTENTS

1. General

1.1. Purpose

The Magellan eXplorist Communication Specification contains all information for developers who want to benefit from providing Thales Navigation's Magellan eXplorist GPS units support in their custom software.

The communication package contains all information for exchange user data. This includes:

- The descriptions to transfer waypoints/routes/tracks between a PC and GPS unit.
- The file formats needed to create waypoints/ routes files
- The driver set and functions to realize real-time logging on the PC using the eXplorist Unit as a receiver

1.2. Legal Disclaimer

The information described in this document and used by Thales Navigation equipment are subject to change without notice. Thales Navigation does not assume any liability or responsibility for any use made of the information in this document

2. eXplorist description

The major difference between the Magellan eXplorist series and previous Magellan devices (Meridian/SporTrak e.g.) is the communication between the unit and the PC. The eXplorist 400, 500, 500 and XL devices communicate through USB while the previous units communicate through serial port using NMEA protocol.

When an eXplorist is connected to the PC, it will be treated as a mass storage device, with file transfer completed simply by copying to and from the disk. So for uploading/downloading waypoints, routes and tracks, no specific software or functions has to be used. This makes it easy for third party to exchange data between the Unit and the PC.

The only exception is the real-time GPS Logging function, which uses a USB-to-serial emulation and still uses the standard NMEA protocol.

3. Standard Folder Structure

This section describes the folder structure in eXplorist internal storage and how the eXplorist organize waypoints, routes and track files.

The eXplorist manages the USB link by a set of USB protocol stack software that does all of the link and layer level maintenance and implements a USB mass storage device driver. The PC OS (nominally Windows) USB host software makes the storage device in the unit, internal NAND storage or SD card, appear to be a standard disk drive attached to the PC. The application tasks in the unit also view the same storage device as a DOS/Windows FAT compatible storage system.

To activate the mass storage mode the unit has to be wired correctly to the PC and turned on. The user has to choose in the appearing Communications Menu the File Transfer Mode and select which mass storage device, internal memory or SD he wants to access. It is not possible to access both the SD card and the NAND flash in parallel from the PC. The switch between the different communication modes could also be reached under Menu: ->Adv Features->Communications

The following folder structure is used in eXplorist storage - the internal storage or an initialized SD card:

1. Background Maps (\Background Maps*.mgi)
2. Detail Maps (\Detail Maps*.imi)
3. Geocaches (\Geocaches*.gs)
4. My POIs (\My POIs*.upt)
5. Regions (\Regions*.rgn)
6. Routes (\Route*.rte)
7. Track Logs (\Track Logs*.trk)

Initialization means here that this standard folder structure will be created, when a new SD card is inserted in the unit.

This structure helps users to maintain their files. There are no restrictions for creating new folders or save a file in any folder.

Next to these folders firmware creates another hidden folder (\USBTRANS\) which contains two files: language.dat and Unit_ID.dat.

The language.dat file contains the folder names in a custom language, so the standard folder could be created in other languages. For example in German:

1. BackgroundMaps=Hintergrundkarten
2. DetailMaps=Detailkarten
3. MyPOIs=Meine Sonderz
4. Routes=Routen
5. TrackLogs=Track-Prot
6. Geocaches=Geocaches

The format of each line is KEY = string value. To find the proper folder, the application need to first check if the hidden file (\USBTRANS\language.dat) exists. If the file exists, the application reads the folder name in the file, and verifies if the folder name exists. If the file does not exist, the application verifies if the standard folder name in English.

The Unit_ID.dat file contains unit ID with “Serial Number”key and other information. The firmware will create the UNIT_ID file in the NAND or SD card, if it is inserted into the unit. Every time an SD card is inserted into the unit, the UNIT_ID file for the card will be over-written. This is done to ensure the card corresponds to the last unit using it. If the storage medium is full or the unit is somehow unable to write the file, it will not be created. Even when a card is locked, eXplorist disregards the write-protect mechanism. Therefore, eXplorist will still be able to write to the card.

To ease the communication with the device, the internal storage or an initialized SD card could be detected through the UNIT_ID file or through a search for the standard folders in English. Therefore the 3rd party software can point automatically to the correct destination.

4. File Formats

This section describes the file format required by eXplorist units. This description enables developer easily to understand and to create waypoints, routes and track files in Magellan format in their custom software.

The file formats and structure are traditionally derived from an extension of the *National Marine Electronics Association NMEA 0183 Standard for Interfacing Marine Electronic Devices* Version 2.10 dated October 15, 1995 and Data Transmission Protocol Specification for Magellan Products. Therefore the file structures keep the formats of NMEA messages even these messages are not used between the USB unit and the host.

This section does not deal with the hardware issues necessary to support this file format. Electrical issues involved in connecting a GPS unit to a PC, or other device, is not covered.

The intended user of this section is assumed to have some level of familiarity with converting data from one format to one another used by PCs and other devices. The user is assumed to be familiar with the concepts of bytes, bit order, general parsing and string manipulation concepts.

All data formats in this document are assumed to be based on an 8 bit byte as the fundamental unit of transfer. All byte (0 to 255) may be used in a record, unless otherwise indicated all bytes are encoded as per ASCII standards. Although any arbitrary byte value could be used in a message that is part of this protocol, an effort has been made to stick to the limitations described in the NMEA standards. The primary affect of this is to limit most characters to ASCII characters with values of 20 (hex) to 60 (hex). This comprises numbers, uppercase letters and most of the punctuation characters.

4.1. General message description

In general, the messages and file structure defined by this specification follow the NMEA Message structure in that they consist of a header, one or more fields, followed by a hex checksum, and ended by a Carriage Return Line Feed delimiter as follows:

```
$PMGNxxx,<fields>*hh<CR><LF>
```

The Header portion of the message, in conformance with NMEA standards, consists of the dollar sign and the letter “P” indicating that this is a private message. Magellan’s registered private message identifier of MGN is next, followed by the command name. In the example above, the lower case “xxx” is replaced with the command identifier in upper case. A comma terminates the Header field.

The data part of the message consists of one or more fields as described in this document. Commas separate each field in the data. Each field, and the data values that they can contain, are described under the various commands as shown in this document.

The tail of the message, separated by an asterisk (*), consists of a two character checksum (hexadecimal notation) followed by a carriage return then a line feed. The checksum consists of the byte-wise exclusive OR of all bytes in the message. This includes all characters between the leading dollar sign and the asterisk immediately before the checksum (the dollar sign and asterisk are not included in the checksum). The checksum is disregarded by eXplorist GPS units.

4.2. General Rules for Position and Time

Unless otherwise indicated all position information is referenced to the WGS-84 datum. All altitude information is referenced to the geoid (e.g. it is height above mean sea level). All date and time information is in UTC.

4.3. Waypoints

Waypoint information is stored in ASCII format. The extension for waypoints is *.upt, which stands for User Points of Interest.

The first line of waypoint files contains a format message, listing the data type and describing the fields within the message. Because the line is comprised of multiple sections (format, metadata), fields beginning with % with denote a new section. An example of the message is:

```
$PMGNFMT,%WPL,LAT,HEMI,LON,HEMI,ALT,UNIT,NAME,MSG,ICON,%META,ASCII
```

The string “%WPL” indicates a waypoint file. It is followed by a list of field names. The string “%META” indicates the metadata section. The only metadata supported now is codepage. It is ASCII in this example. The header line is followed by data lines, one for each waypoint, in the following data field, introduced by the %WPL section of $PMGNFMT message.

An example of this message is given below:

```
$PMGNWPL,llll.lll,N,lllll.lll,W,aaaa,F,c----c,c---c,c—c,xx*hh<CR><LF>
```

The first field consists of two digits representing the degrees of Latitude, with a leading zero if necessary, followed by two digits representing the minutes of Latitude, with a leading zero if necessary, followed by fractional minutes. For example, a latitude of 38° 15’ 30” would be encoded as 3815.5. The next field consists of either the letter “N” or “S” to indicate North or South Latitude. The next field consists of the Longitude formatted like the Latitude, but with 3 digits to represent the degrees of Longitude. For example, a Longitude of 118° 5’ 15” would be encoded as 11805.25. This field is followed by an “E” or “W” to indicate East or West Longitude. The next field is the Altitude, followed by its units (“F” for feet, “M” for meters). Three text fields follow the position fields. The first is the name of the waypoint (20-character limit, not including the null terminator), the next is a message or comment for this waypoint, and the last is an Icon identifier.

A blank icon field indicates a default icon. Magellan Products use lower case to reference the icon associated with the waypoint. See the icon appendix for details. The final data field is a waypoint type indicator (a numerical value particular to a given model of GPS units), which is used in some GPS units. Trailing fields need not be transmitted if they are empty.

The following line indicates the end of the file

```
$PMGNCMD,END*3D
```

4.3.1. Magellan waypoint file example

```
$PMGNFMT,%WPL,LAT,HEMI,LON,HEMI,ALT,UNIT,NAME,MSG,ICON,CHKSUM,%META,ASC
II
$PMGNWPL,3402.77945,N,11750.83783,W,227,M,WPT001,,a*2B
$PMGNWPL,3359.81504,N,11746.68163,W,355,M,WPT002,,a*2F
$PMGNWPL,3358.79483,N,11742.98982,W,204,M,WPT003,,a*2F
$PMGNWPL,3401.73998,N,11740.48218,W,249,M,WPT004,,a*20
$PMGNWPL,3405.57061,N,11740.66793,W,367,M,WPT005,,a*2B
$PMGNWPL,3408.88150,N,11750.00194,W,356,M,WPT006,,a*26
$PMGNWPL,3403.37618,N,11742.89695,W,273,M,WPT007,,a*21
$PMGNWPL,3401.81698,N,11751.34864,W,226,M,WPT008,,a*2D
$PMGNWPL,3405.68610,N,11751.83624,W,201,M,WPT009,,a*2C
$PMGNCMD,END*3D
```

4.4. Routes

The extension for route files is *.rte. A route file can contain multiple routes. Route files contain information of all waypoints used in the routes. This makes the route file self-contained and ensures that the file does not have any dependencies on other files.

In the beginning of all route files, a format message will be included, similar to waypoint file, to describe the fields within the message. Because the route file contains route and waypoint data, the header will contain format information for both types of data:

```
$PMGNFMT,%RTE,NUM_MSG,ID,FLAG,NUM,NAME,WPL,LAT,HEMI,LON,HEMI,ALT,UNIT,NA
ME,MSG,ICON,%WPL,LAT,HEMI,LON,HEMI,ALT,UNIT,NAME,MSG,ICON,%META,ASCII
```

The %RTE section lists the fields which will appear in the $PMGNRTE message.

Every route is introduced by the RTE Message.

```
$PMGNRTE,xx,xx,c,n,c----c,c,c---c,c,c---c,c,.......*hh<CR><LF>
```

The message itself consists of two groups of fields. The first set of fields consists of header information and includes the Magellan Proprietary Message Identifier, followed by the number of messages that make up this route, the individual Id number for this message, the letter “c”, the route number, and the Route Name (max. of 20 characters, not including the null terminator). The second set of fields are the names, in order, of the waypoints that make up the route, followed by Icon information in lower case The Icon information applies to the waypoint name immediately to it’s left. If no waypoint Icon information is being transmitted, then the icon field should be left empty. For examples,

```
$PMGNRTE,2,1,c,1,FOO,POINT1,b,POINT2,c,POINT3,d*6C<CR><LF>

$PMGNRTE,2,2,m,1,FOO,THIS IS A ROUTE MESSAGE*1F<CR><CLF>
```

“$PMGNRTE” identifies these two lines as route messages, the next two fields, “2,1” indicate that this is the first line for this route, which is described in two messages. The next field, “c” indicates that this is a complete route description. “FOO” is the name of this route. The first waypoint that makes up this route is “POINT1” and has a special Icon identified as “b” attached to it. The second point of this route is “POINT2”. The third point of the route is “POINT3”.

The second line is decoded as follows: “$PMGNRTE” indicates that this is a Magellan Private Message. The next two fields indicate that this is the second message of two messages that make up this route. The lower case ‘m’ indicates that this is a message record. The next field, “FOO” is the name of the route. This name field must be the same for all records that make up a route. The next field contains the message that is to be attached to the route. This message may be in mixed case, however, only certain Thales Navigation products that support lower case letters may be able to use a mixed case message. The message is terminated by the usual checksum field followed by carriage return/line feed pair.

Lines for one route must be together in increasing numerical order. A line with 1 of ‘x’ indicates the start of a route description, a line with ‘x’ of ‘x’ indicates that the route is complete. Missing sequence numbers indicate that some data is missing, and the route should not be processed.

The route files end with the following line

```
$PMGNCMD,END*3D
```

4.4.1. Magellan route file example

```
$PMGNFMT,%RTE,NUM_MSG,ID,FLAG,NUM,NAME,WPT_NAME1,ICON1,WPT_NAME2,ICON
2,CHKSUM%WPL,LAT,HEMI,LON,HEMI,ALT,UNIT,NAME,MSG,ICON,CHKSUM,%META,ASCII
$PMGNWPL,3408.88150,N,11750.00194,W,356,M,WPT006,,a*26
$PMGNWPL,3403.37618,N,11742.89695,W,273,M,WPT007,,a*21
$PMGNWPL,3401.81698,N,11751.34864,W,226,M,WPT008,,a*2D
$PMGNWPL,3405.68610,N,11751.83624,W,201,M,WPT009,,a*2C
$PMGNRTE,2,1,c,1,Route 1,WPT006,a,WPT007,a*63
$PMGNRTE,2,2,c,1,Route 1,WPT008,a,WPT009,a*60
$PMGNCMD,END*3D
```

4.5. Tracks

Tracks files have the *.trk extension. A Track file is basically a sequence of GPS position fixes. At the beginning of all track files, a format message will be included, describing the fields within the message:

```
$PMGNFMT,%TRK,LAT,HEMI,LON,HEMI,ALT,UNIT,TIME,VALID,NAME,%META,ASCII
```

The header line is followed by the following data message

```
$PMGNTRK,llll.ll,a,yyyyy.yy,a,xxxxx,a,hhmmss.ss,A,c----c,ddmmyy*hh<CR><LF>
```

The first field in this message is the Latitude, followed by N or S. The next field is the Longitude followed by E or W. The next field is the altitude followed by “F” for feet or “M” for meters. The next field is the UTC time of the fix. The next field consists of a status letter of “A” to indicate that the data is valid, or “V” to indicate that the data is not valid. The last character field is the name of the track. The last field contains the UTC date of the fix. Note that this field is (and its preceding comma) is optional.

NOTE: The Latitude and Longitude Fields are shown as having two decimal places. As many additional decimal places may be added as long as the total length of the message does not exceed 82 bytes.

4.6. Magellan Geocache File Format

Geocaching points are stored in files with *.gs extension. The PMGNGEO message is based upon the above described WPL message with the addition of altitude, name, dates, and type information fields.

An example of this message is given:

```
$PMGNGEO,llll.lll,N,lllll.lll,W,aaaa,F,c----c,c---c,c—c,c—c, ddmmyyy,ddmmyyy, xx*hh<CR><LF>
```

The first field consists of two digits representing the degrees of Latitude, with a leading zero if necessary, followed by **two** digits representing the minutes of Latitude, with a leading zero if necessary, and followed by fractional minutes. For example, a latitude of 38° 15’ 30” would be encoded as 3815.5. The next field consists of either the letter “N” or “S” to indicate North or South Latitude. The next field consists of the Longitude formatted like the Latitude, but with 3 digits to represent the degrees of Longitude. For example, a Longitude of 118° 5’ 15” would be encoded as 11805.25. An “E” or “W” to indicate East or West Longitude follows this field. The next field is the Altitude, followed by its units (“F” for feet, “M” for meters). Four text fields follow the position; the first is the Cache Id of the geo point.

Next is a Name or comment for this geo point, then the Owner and Hint followed by the Type identifier:

Identifier
Traditional Caches
Multi-Caches
Mystery Caches
Puzzle Caches
Letterbox Hybrid
Virtual Caches
Reverse Virtual
Locationless Caches
Webcam Caches
Event Caches
CITO Event Caches
Unknown Caches

Followed by the date placed and date found.

dd	Day of file creation (UTC)
mm	Month (numerical) of file creation (UTC)
yy	Year of file creation, expressed relative to 1900 (2000 would be 100) (UTC)

The following field is Difficulty and Terrain

4.6.1. Magellan gs file example

```
$PMGNGEO,3348.027,S,15109.135,E,0000,F,GCGC56,Grumpy,Leek,,Traditional
Cache,0407103,0907103,,*32
$PMGNGEO,3343.072,S,15029.674,E,0000,F,GCGC5V,Sleepy's Legacy,Riblit,,Multi-
Cache,0407103,1207103,,*0E
$PMGNGEO,3403.742,S,15103.740,E,0000,F,GCGC68,Bashful,WauloK,,Mystery
Caches,2906103,1207103,,*76
$PMGNGEO,3402.526,S,15103.583,E,0000,F,GCGC6C,Blue Midget,WauloK,,Letterbox
Hybrid,2906103,1207103,,*71
$PMGNGEO,3402.278,S,15112.362,E,0000,F,GCGCEC,Heigh-ho Heigh-ho,The Rats,,Virtual
Caches,3006103,1207103,,*41
$PMGNGEO,3342.886,S,15113.843,E,0000,F,GCGCEQ,Snow White,Bushrat,,Locationless
Caches,0407103,1207103,,*30
$PMGNGEO,3409.531,S,15058.351,E,0000,F,GCGCEX,DOPEY and the mystery of the holy
smoke,odlids,,Webcam Caches,0407103,1907103,,*2B
$PMGNGEO,3332.254,S,15037.985,E,0000,F,GCGCEY,Bugger! - EcoTeam
#10,EcoTeam,,Event Caches,3006103,1907103,,*03
```

```
$PMGNGEO,3351.000,S,15112.200,E,0000,F,GCGD1D,MS-06: What's Up Doc,Mind
Socket,,Unknown Cache,0307103,1807103,,*5E
$PMGNCMD,END*3D
```

The size of the file is only limited by the size of the media. The unit will only read records and may truncate fields based on the memory available on the unit.

4.6.2. Geocache Raw Data Field Extraction

Geocache raw data may be attained in *.LOC or *.GPX files from various providers, for example www.geocaching.com . The Magellan PMGNGEO record supports the following fields from the *.LOC OR *.GPX raw data:

Field	Comment
Cache Id*	Alphanum
Name*	Alphanum (Key for Alpha Query on the unit)
Owner	Alphanum
Location*	Lat/Lon: Numeric (Key for Nearest Query on the unit, WGS84)
Type	Char
Date Placed	DD-MM-YYYY
Date Last Found	DD-MM-YYYY
Difficulty	Numeric
Terrain	Numeric
Hint	Alphanum

*The minimum number of fields required for a valid Magellan geocache record is Cache Id, Name and Location.

5. GPS Logging

In the eXplorist series, the GPS Logging or Tracking function is the only function which needs the serial NMEA messages/ protocol. It requires to install an USB to Serial emulation driver, magellanexplorist.inf, on PC. After the driver is installed, the emulation driver can be found in the Device Manager, and the USB connection will act just like a regular serial port.

The USB to Serial emulation is realized over USB communication device classes (UBS CDC). The NMEA messages are sent across USB line through a pipe and the pipe supports a packet size of 32 bytes. A bigger NMEA sentence will be divided into multiple USB packets. A packet might contain the next NMEA sentence; differentiate with a new line character (“\n”). The CDC class provides the mechanism to store all the data coming from the USB device in an intermediate buffer and provide to the 3rd party software like a serial device. The intermediate buffering and baud-rate etc. can be configured by user from the device properties, but this is normally not needed.

The NMEA data communication mode could be activated and turned off on the GPS units through the communications menu. It appears automatically on the screen when the unit is turned on with existing wiring to PC USB port or through the menu path: Menu: ->Adv Features->Communications->NMEA Data Comm-> 2.1GSA. With this selection the GPS unit reboots. For the first time, in a plug-n/play environment, the GPS unit should be recognized as new device (Magellan eXplorist to serial).

Once activated the Unit starts sending NMEA Messages every second. From developer point of view the GPS device can be found by scanning the COM ports for GPS devices. After finding the device the position can be tracked.

The eXplorist supports/ sends the following Magellan commands

Sentence/Command	**Description**
$PMGNST	Magellan status message (not described in here)
NMEA	Turn on/off logging
USB_FILE_XFER	Switching to USB File Transfer Mode

And the following standard NMEA messages

Sentence	**Description**
$GPGGA	Global positioning system fixed data
$GPGLL	Geographic position - latitude / longitude
$GPGSA	GNSS DOP and active satellites
$GPGSV	GNSS satellites in view
$GPRMC	Recommended minimum specific GNSS data

5.1. Messages: NMEA and USB_File_XFER

The eXplroist supports the Magellan $PMGNCMD command for turning NMEA mode and USB mode on and off. The commands are sent as ASCII text, in upper case.

```
$PMGNCMD,NMEAON*74<CR><LF>
$PMGNCMD,NMEAOFF*3A<CR><LF>
$PMGNCMD,USB_FILE_XFER*39<CR><LF>
```

For detailed description of $PMGNCMD please refer to Magellan Data Transmission protocol.

5.2. $GPGGA Sentence (Fix data)

Example (signal not acquired):
$GPGGA,235947.000,0000.0000,N,00000.0000,E,0,00,0.0,0.0,M,,,,0000*00
Example (signal acquired):
$GPGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,,,0000*1F

Field	**Example**	**Comments**
Sentence ID	$GPGGA	
UTC Time	092204.999	hhmmss.sss
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Position Fix	1	0 = Invalid, 1 = Valid SPS, 2 = Valid DGPS, 3 = Valid PPS
Satellites Used	04	Satellites being used (0-12)
HDOP	24.4	Horizontal dilution of precision
Altitude	19.7	Altitude in meters according to WGS-84 ellipsoid
Altitude Units	M	M = Meters
Geoid Seperation		Geoid seperation in meters according to WGS-84 ellipsoid
Seperation Units		M = Meters
DGPS Age		Age of DGPS data in seconds
DGPS Station ID	0000	
Checksum	*1F	
Terminator	CR/LF	

5.3. $GPGLL Sentence (Position)

Example (signal not acquired):
$GPGLL,0000.0000,N,00000.0000,E,235947.000,V*2D
Example (signal acquired):
$GPGLL,4250.5589,S,14718.5084,E,092204.999,A*2D

Field	**Example**	**Comments**
Sentence ID	$GPGLL	
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West

UTC Time	092204.999	hhmmss.sss
Status	A	A = Valid, V = Invalid
Checksum	*2D	
Terminator	CR/LF	

5.4. $GPGSA Sentence (Active satellites)

Example (signal not acquired): $GPGSA,A,1,,,,,,,,,,,,,0.0,0.0,0.0*30
Example (signal acquired): $GPGSA,A,3,01,20,19,13,,,,,,,,,40.4,24.4,32.2*0A

Field	**Example**	**Comments**
Sentence ID	$GPGSA	
Mode 1	A	A = Auto 2D/3D, M = Forced 2D/3D
Mode 1	3	1 = No fix, 2 = 2D, 3 = 3D
Satellite used 1	01	Satellite used on channel 1
Satellite used 2	20	Satellite used on channel 2
Satellite used 3	19	Satellite used on channel 3
Satellite used 4	13	Satellite used on channel 4
Satellite used 5		Satellite used on channel 5
Satellite used 6		Satellite used on channel 6
Satellite used 7		Satellite used on channel 7
Satellite used 8		Satellite used on channel 8
Satellite used 9		Satellite used on channel 9
Satellite used 10		Satellite used on channel 10
Satellite used 11		Satellite used on channel 11
Satellite used 12		Satellite used on channel 12
PDOP	40.4	Position dilution of precision
HDOP	24.4	Horizontal dilution of precision
VDOP	32.2	Vertical dilution of precision
Checksum	*0A	
Terminator	CR/LF	

5.5. $GPGSV Sentence (Satellites in view)

Example (signal not acquired): $GPGSV,1,1,01,21,00,000,*4B
Example (signal acquired):
$GPGSV,3,1,10,20,78,331,45,01,59,235,47,22,41,069,,13,32,252,45*70

Field	**Example**	**Comments**
Sentence ID	$GPGSV	

Number of messages	3	Number of messages in complete message (1-3)
Sequence number	1	Sequence number of this entry (1-3)
Satellites in view	10	
Satellite ID 1	20	Range is 1-32
Elevation 1	78	Elevation in degrees (0-90)
Azimuth 1	331	Azimuth in degrees (0-359)
SNR 1	45	Signal to noise ration in dBHZ (0-99)
Satellite ID 2	01	Range is 1-32
Elevation 2	59	Elevation in degrees (0-90)
Azimuth 2	235	Azimuth in degrees (0-359)
SNR 2	47	Signal to noise ration in dBHZ (0-99)
Satellite ID 3	22	Range is 1-32
Elevation 3	41	Elevation in degrees (0-90)
Azimuth 3	069	Azimuth in degrees (0-359)
SNR 3		Signal to noise ration in dBHZ (0-99)
Satellite ID 4	13	Range is 1-32
Elevation 4	32	Elevation in degrees (0-90)
Azimuth 4	252	Azimuth in degrees (0-359)
SNR 4	45	Signal to noise ration in dBHZ (0-99)
Checksum	*70	
Terminator	CR/LF	

5.6. $GPRMC Sentence (Position and time)

Example (signal not acquired):
$GPRMC,235947.000,V,0000.0000,N,00000.0000,E,,,041299,,*1D
Example (signal acquired):
$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,*25

Field	Example	Comments
Sentence ID	$GPRMC	
UTC Time	092204.999	hhmmss.sss
Status	A	A = Valid, V = Invalid
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Speed over ground	0.00	Knots
Course over ground	0.00	Degrees
UTC Date	211200	DDMMYY
Magnetic variation		Degrees

Magnetic variation		E = East, W = West
Checksum	*25	
Terminator	CR/LF	

Appendix A: Icons

The eXplorist units support to attach a characterizing icon to waypoints. In the waypoint file they are aligned to the waypoint over one or two characters as described above. In the following table you find the allocation between characters and icons.

No	Lowercase character	Icon Name	Icon	File name
1	a	Crossed box	⊞	Udefault
2	b	box	▣	Box
3	c	house	🏠	House
4	d	aerial	🎈	Aerial
5	e	airport	✈	Airport
6	f	amusement park	🎢	amusesement
7	g	ATM	$	ATM
8	h	auto repair	🔧	Auto_Repair
9	I	boating	⛵	boating
10	j	camping	⛺	camps
11	k	exit ramp	▨	Exit_ramps
12	l	first aid	✚	First_aid
13	m	nav aid	✢	Fixed_nav_aid
14	n	buoy	⚲	Float_buoy
15	o	fuel	⛽	Fuel
16	p	garden	✿	gardens
17	q	golf	⛳	Golf_courses
18	r	hotel	🏨	hotel
19	s	hunting/fishing	🐟	Hunt_fish
20	t	large city	●	Large_cities
21	u	lighthouse	Lt	Light_house
22	v	major city	◘	Major_cities
23	w	marina	⚓	marina
24	x	medium city	▪	Medium_cities
25	y	museum	🏛	Museum
26	z	obstruction	☠	obstruction
27	aa	park	🌲	Parks
28	ab	resort	R	resort
29	ac	restaurant	🍴	Restaurants
30	ad	rock	☗	Rocks
31	ae	scuba	🤿	Scuba
32	af	RV service	RV	RV_Services
33	ag	shooting	◎	shooting
34	ah	sight seeing	🔭	Sight_seeing
35	ai	small city	·	Small_cities
36	aj	sounding	⋮	soundings
37	ak	sports arena	🏟	Sports_areana
38	al	tourist info	🟥	tourist
39	am	Truck service	🚚	Truck_services
40	an	winery	🍇	Wineries
41	ao	wreck	🚢	wreck
42	ap	zoo	🐾	zoos