Goke AGPS

User Manual

Legal notices

If you receive this document from Hunan Guoke Microelectronics Co., Ltd. (hereinafter referred to as "Guoke Micro"), it means that you have agreed to the following terms. If you do not agree to the following terms, please stop using this document.

This document is copyrighted by Hunan Guoke Microelectronics Co., Ltd., and reserves any rights not expressly granted in this document. Without the prior written permission of Guoke Micro, no unit or individual may copy, transmit, distribute, use or leak the document and any pictures and forms contained in the document , data and other information.

The Company reserves the right to make modifications and improvements to the products described in this manual without prior notice; We reserve the right to revise or retract this manual at any time.

If there is any unclear text in this user manual, please consult the company or agents or sellers in time.

1 GPD introduce

GPD is an implementation defined by Goke for AGPS-assisted positioning. It mainly uses the current navigation data Rinex file to obtain from the IGS website, and then convert it into the current ephemeris and transmit it to the chip through the serial port This accelerates the positioning of the GPS chip.

1. How to obtain GPD files

Download the GPD file for the current time (http://www.goke-agps.com:7777/brdcGPD.dat by visiting Goke's GPD server website)。 Since the real-time ephemeris published by the IGS website is updated every 2 hours, the relative GPD files are also updated every 2 hours 。

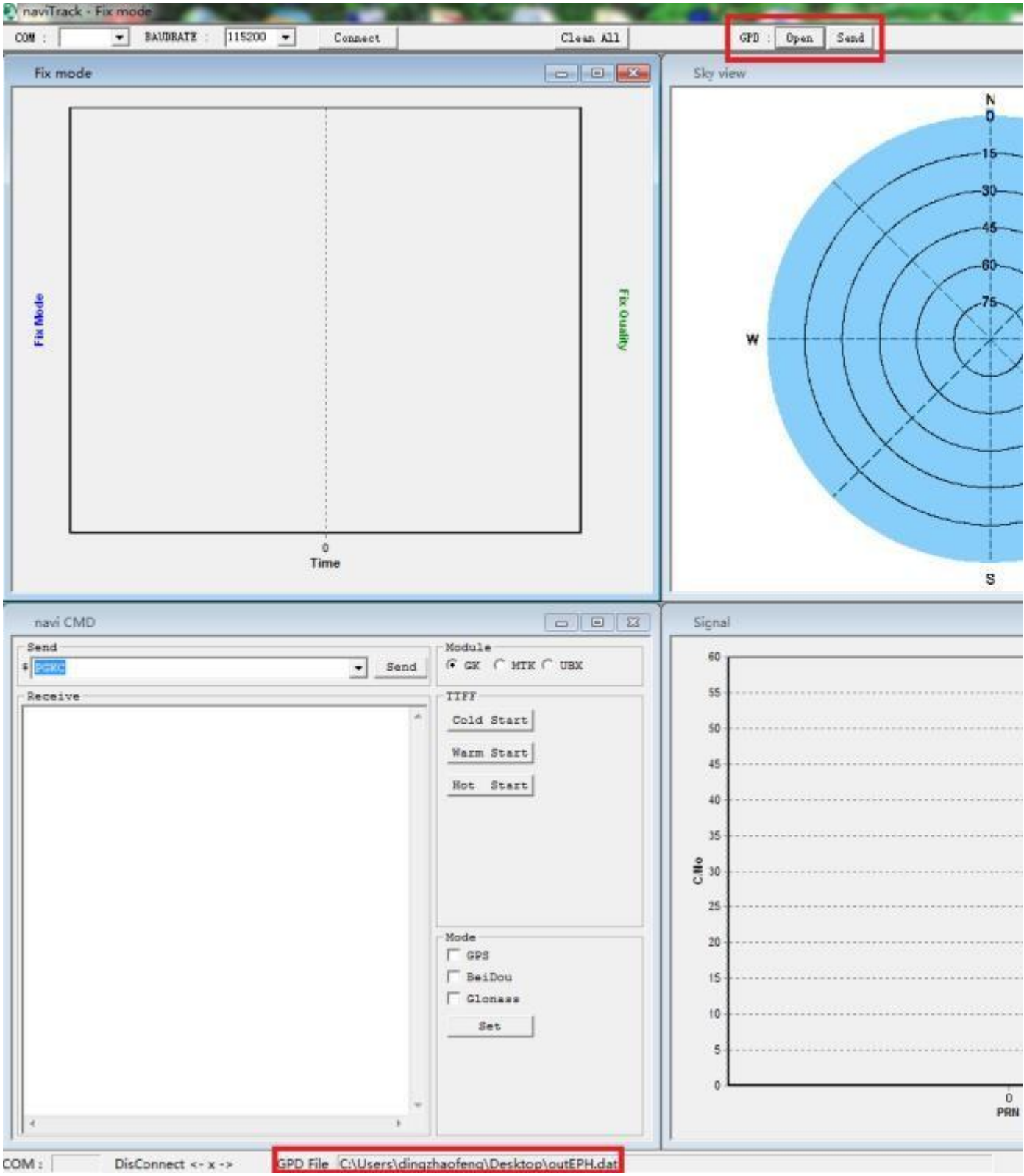
1. How to use GPD files

Uploaded to the chip via the naviTrack tool provided by GOKE.

After the chip is powered on, click the "open" button next to the GPD icon above and select the GPD file downloaded online. The selected file information is displayed below the tool.

After successfully selecting the file, click the "send" button and the tool starts uploading.

After waiting for a few moments, a completion prompt box appears stating that the upload was successful, otherwise it failed and you can re-upload.



1. How to delete GPD data in the chip

Since GPD data is only valid for 6 hours, if the time limit is exceeded, the presence of GPD data on the chip will not work. The user can clear the GPD data in the chip by sending serial port commands. Of course, every time you upload new GPD data, you will also clean up the old GPD data first.

Delete GPD Data command: Enter in the command input box for the naviTrack navigation tool

"PGKC047"。

1. Effects after using GPD

Since the ephemeris data of the currently visible satellites has been obtained, the positioning time can be effectively reduced. After using GPS, you can increase the cold start positioning time by about 10 to 15 seconds. Especially in the case of a weaker signal, it is possible to increase the positioning speed more.

1. An auxiliary positioning method that accelerates GPD

Since pure GPD assistance also relies on searching for GPS time information in the sky, sometimes it consumes too much time when the signal is poor. Faster positioning times can be achieved by setting the current time information and coarse coordinate information with the PGKC639 command.

Note: When set by the 639 command, the range of latitude and longitude should deviate from the actual position by 20km

Within, the time deviation should not exceed 5 minutes.

Command:639

Set approximate location and time information to speed up positioning.

Arguments:

Arg1: Latitude, for example:

28.166450

Arg2: Longitude, for example:

120.389700

Arg3: Height, for
example: 0
Arg4: year
Arg5: month
Arg6: day
Arg7: hour, The time is UTC time
Arg8: minute
Arg9: second

Example:

$PGKC639,28.166450,120.389700,0,2017,3,15,12,0,0*33<CR><LF>

639 After the command is successful, GK9501 returns the following format:

$PGKC001,639,3*21

2 GPD Communication transmission

Mainly through serial communication to transfer GPD data to the chip in chunks. The main process is as follows：

1. Switch NMEA receive to BINARY receive mode (gK9501 input and output format .pdf for command formats).

   Send: Message type + switch mode + baud rate + CheckSum

   Data: $PGKC 149, 1, 115200*15

   (The message type transmitted by GPD is 149).

   Receive: Packet header (2B) + Packet length (2B) + ACK type (2B) + Message type (2B) + Valid flag (1B) + CheckSum(1B)+ Bag tail (2B).

   data：0xaa,0xf0,0x0c,0x00,0x01,0x00,0x95,0x00,0x03,(chk),0x0d,0x0a

   (Checksum is a byte-byte XOR from the packet length field to the field

before checksum).)

1. Sends the first block of GPD data and waits for the ACK to reply

Send: Packet header (2B) + Packet length (2B) + Transmission Type (2B) + GPD Packet Sequence Number (2B).
+ Data payload(512B) + CheckSum(1B) + Package tail (2B).
data：0xaa,0xf0,0x0b,0x02,0x066,0x02,0x00,0x00,……,(chk),0x0d,0a

Receive: Packet header (2B) + packet length (2B) + ACK type (2B) + GPD packet sequence number (2B).
+ Valid flag (1B) + CheckSum (1B) + End of bag (2B).
data：0xaa,0xf0,0x0c,0x00,0x03,0x00,0x00,0x00,0x01,(chk),0x0d,0x0a

1. Send the remaining chunks in turn and wait for the ACK to answer

Send: Packet header (2B) + Packet length (2B) + Transmission Type (2B) + GPD Packet Sequence Number (2B).
+ Data payload(512B) + CheckSum(1B) + Package tail (2B).
data：0xaa,0xf0,0x0b,0x02,0x066,0x02,0x01,0x00,……,(chk),0x0d,0a
(The GPD file is split into 512-byte chunk transfers, with the last chunk less than 512 bytes padded with 0.)

Receive: Packet header (2B) + packet length (2B) + ACK type (2B) + GPD packet sequence number (2B).
+ Valid flag (1B) + CheckSum (1B) + End of bag (2B).
data：0xaa,0xf0,0x0c,0x00,0x03,0x00,0x01,0x00,0x01,(chk),0x0d,0x0a

1. Sends a GPD transfer end statement, waiting for a reply

Send: Packet header (2B) + Packet length (2B) + Transmission type (2B) + GPD terminator (2B).

+CheckSum（1B）+Package tail（2B）

data：0xaa,0xf0,0x0b,0x00,0x066,0x02,0xff,0xff,(chk),0x0d,0a

Receive: Packet header (2B) + packet length (2B) + ACK type (2B) + GPD terminator (2B).

+ Valid flag (1B) + CheckSum (1B) + End of bag (2B).

data：0xaa,0xf0,0x0c,0x00,0x03,0x00,0xff,0xff,0x01,(chk),0x0d,0x0a

1. Toggles binary receive to NMEA receive mode

Send: Packet header (2B) + Packet length (2B) + Message type (2B) + Transport type (1B) +

Baud rate (4B) + CheckSum (1B) + end of package (2B).

Data: 0xaa, 0xf0, 0x0e, 0x00, 0x95, 0x00, 0x00, 0x00, 0xc2, 0x01, 0x00, (chk), 0x0d, 0x0a

Receive: Packet header (2B) + Packet length (2B) + ACK type (2B) + Message type (2B) + Valid flag (1B) + CheckSum(1B)+ Bag tail (2B).

Data: 0xaa, 0xf0, 0x0c, 0x00, 0x01, 0x00, 0x95, 0x00, 0x03, (chk), 0x0d, 0x0a

(Status flags: 0 for invalid, 1 for unsupported, 2 for failed, 3 for success).