Date: 2010/10/01 **By: Wen-Hsin Chen / FW Engineer**

Communication Specification

Model: DG200

Product Application GPS data logger

Revision History

Version	Change History	Issue Date	Remark
0.1	Initiation	2010/10/1	Preliminary

GlobalSat Technology Corporation
16F., NO.186, Jian-Yi Road, 235 Chung Ho City,
Taipei Hsien, Taiwan, R.O.C.
Tel: +886-2-8226-3799 Fax: +886-2-8226-3899

The Format of communication between PC software and DG 200

1. General Format & Rules

For Sending Format

The general format of communication sent to DG 200 is:

Start Sequence	Payload Length	Payload	Message Checksum	End Sequence
0xA0,0xA2	Two-bytes	Up to 1023	Two-bytes	0xB0,0xB3

For Returning Format

The general format of communication received from DG 200 is:

Start Sequence	Payload Length	Payload	Message Checksum	End Sequence
0xA0,0xA2	Two-bytes	Up to 1023	Two-bytes	0x0D,0x00

Note:

All information from/to DG200 is Big-Endian.

The information exchanged between PC software and DG 200 is located in the **PAYLOAD field**.

For RETURNED COMMAND, the length of PAYLOAD is ignored.

How to calculate checksum

Assume **PAYLOAD** is a array that is transmitted, and **PAYLOAD_LEN** is the length of this array:

```
Checksum = PAYLOAD [0];
For ( i =1; I < PAYLOAD_LEN; i++)
{
      Checksum = Checksum + PAYLOAD [i];
}
```

Checksum = Checksum & $(2^{15} - 1)$;

Payload Format

The exchange data format used in DG is:

Command ID	Parameter
One-byte	n-bytes value

2. Communication Interface

2.1 Get Configuration

Send Command:

Command ID	Parameter
0xB7	None

Return Value:

Command ID	Parameter
0xB7	45-bytes value

Parameter description:

Information Type	Speed Threshold Flag	Speed Threshold	Distance Threshold flag	Distance Threshold
Byte 0	Byte 1	Byte 2 – byte 5	Byte 6	Byte7-byte10
Time Interval 1	Time Interval 2	Time Interval 3	Not used	DataSaveOverwrite
Byte11-byte14	Byte15-byte18	Byte19-byte22	Byte23	Byte 24
Interval by time1/distance1 flag	Interval by time2/distance2 flag	Interval by time3/distance3 flag	Interval by distance1	Interval by distance2
Byte25	Byte26	Byte27	Byte28-byte31	Byte32-byte35
Interval by distance3	Opration Mode	WAAS	Memory usge	Not used
Byte36-byte39	Byte40	Byte41	Byte42	Byte43
Model type				
Byte44				

Note:

The unit of time is seconds.

The unit of distance is meter.

The value of flag: 0 is DISABLE, 1 is ENABLE.

Information type: 0: position only, 1: position, speed and date/time, 2: position, speed, date/time and altitude.

2.2 Set Configuration

Send Command:

Command ID	Parameter
0xB8	42-bytes value

Parameter description:

Information Type	Speed Threshold Flag	Speed Threshold	Distance Threshold flag	Distance Threshold
Byte 0	Byte 1	Byte 2 – byte 5	Byte 6	Byte7-byte10
Time Interval 1	Time Interval 2	Time Interval 3	Not used	Not used
Byte11-byte14	Byte15-byte18	Byte19-byte22	Byte23	Byte 24
Interval by time1/distance1 flag	Interval by time2/distance2 flag	Interval by time3/distance3 flag	Interval by distance1	Interval by distance2
Byte25	Byte26	Byte27	Byte28-byte31	Byte32-byte35
Interval by distance3	Opration Mode	WAAS		
Byte36-byte39	Byte40	Byte41		

Return Value:

Command ID	Parameter
0xB7 or 0xB8	4-bytes value

Parameter description:

Result
Byte 0-byte3

Note:

If it is OK, result = 1.

2.3 Get Track file header

Send Command:

Command ID	Parameter
0xBB	2-bytes value

Parameter description:

The index of the first train file in this iteration
Byte 0-byte1

Return Value:

Command ID	Parameter
0xBB	(12*N+4)-bytes value

Note:

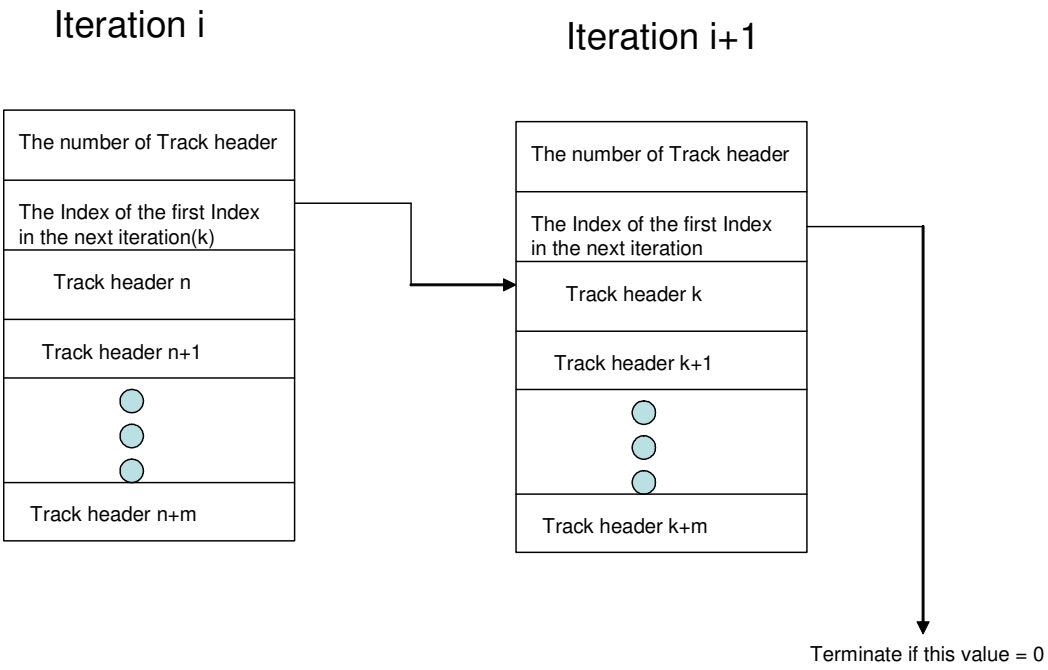
N is the number of track file headers

Parameter description:

The number of track file headers returned (assume it is N)	The index of the next track file can be got	The track file header
Byte 0-byte1	Byte2-byte3	Byte4- byte(12*N+3)

Note:

Continue to call the function until N = 0. The index of the first track file in this iteration is the index of the next track file returned in last iteration. The index of the first track file is 0 in the first iteration. Refer the flow chart shown in the next page.



Header Format:

Byte0-Byte3: Time=>32 bit integer:
Value: HHMMSS
Byte4-Byte7: Date=>32 bit integer
Value:YYMMDD
Byte8-Byte11: The index of file=>32 bit integer

For example:
Time: 17:10:10, the value is 171010
Date: 2006/12/13, the value is 061213

2.4 Get a track file

Send Command:

Command ID	Parameter
0xB5	2-bytes value

Parameter description:

The index of the track file
Byte 0-byte1

Return Value:

The track file is return in two sessions, so the two sessions should be combined together to parse data.

Command ID	Parameter
0xB5	1024-bytes value

Parameters:

Track file
Byte 0-byte1023

Note:

The End sequence for this command is 6-bytes values.

Parse track records in a track file:

1. Get the parameter(1032-bytes value) from two sessions
2. Strip the GPS information from the parameter(1024-bytes value, byte0-byte1023)
3. Combine two 1024-bytes value then parse GPS information:
4. The format of a track file will like as:

Track record 1	Track record 2	Track record 3 … Track record n
Byte 0-byte31	Byte32-byteM	byteM-byte2047

The format of the first track record must be FORMAT C (32 bytes). The format of other track records in this track file is decided by the field “STYLE” of the first track record.

There are three formats for storing track records:

a. Position, date/time and speed(20 bytes):

Latitude	Longitude	UTime	UDate	Speed
Byte 0-byte3	Byte4-byte7	Byte8-byte11	Byte12-byte15	Byte16-byte19

b. Position, date/time, speed and altitude(32 bytes):

Latitude	Longitude	UTime	UDate	Speed
Byte 0-byte3	Byte4-byte7	Byte8-byte11	Byte12-byte15	Byte16-byte19
Altitude	Not used	Style		
Byte20-byte23	Byte24-byte27	Byte28-byte31		

Note:

The format of all fields are described as following, and the type of all fields are number.

Latitude: ddmmmmm, N/S indicator: N: if this field > 0, S: if this field < 0

Longitude: dddmmmmm, E/W indicator: E: if this field >0, W: if this field >0

UTime: hhmmss

UDate: ddmmyy

Speed: (km/hour) * 100

Altitude: (meter)*10000

Style: 0: Position only, 1: Position, date/time and speed, 2: Position, date/time, speed and altitude

Example: How the get the track record?

The raw data (HEX format) is like that: it is FORMAT C

0026239B00B95F380001B2070001D97600000000000000000000000100000002

So parsing:

(0026239B)(00B95F38)(0001B1A1)(0001D912)(00000064)(00006400)(00000001)(00000002)

Latitude = 0x0026239B = 2499483 =>24°99.483

Longitude=0x00B95F38=12148536=>121°48.536

UTime=0x0001B1A1=111009 =>11:10:09(hour:minute:second)

UDate=0x0001D912=121106=>2006/11/12

Speed=0x00000064=100=> 1 (km/hour)

Altitude=0x00006400=25600=> 2.56 meter

Style=0x0000002 =>Format C

2.5 Delete All track files

Send Command:

Command ID	Parameter
0xBA	0xFF,0xFF

Return Value:

Command ID	Parameter
0xBA	4-bytes value

Parameter description:

Result
Byte 0-byte3

Note:

Result = 1, it is OK.

2.6 Get the ID of DG

Send Command:

Command ID	Parameter
0xBF	None

Return Value:

Command ID	Parameter
0xBF	8-bytes value

Parameter description:

ID of DG200
Byte 0-byte7

2.7 get SW version

Send Command:

Command ID	Parameter
0xBC	1-bytes value

Parameter description:

G-Mouse on/off
Byte 0

If G-Mouse on/off is 1, NMEA output will send.

Return Value:

$PSRFTXT,Version: sirf chip sw version
$PSRFTXT,Version2: DG-200 sw version

2.8 cold start, warm start, factory reset

Send Command:

Command ID	Parameter
0x80	24-bytes value

Parameter description:

Set 0x00	Channel cnt set 0x0c	Init type
Byte 0-byte21	Byte 22	Byte 23

Cold start, Init type is 0x84
warm start, Init type is 0x83
hot start, Init type is 0xc0
factory reset , Init type is 0x88

2.9 Set the ID of DG

Send Command:

Command ID	Parameter
0xC0	8-bytes value

Return Value:

Command ID	Parameter
0xC0	4-bytes value

Parameter description:

Result
Byte 0-byte3

Note:

1. **one byte represents a digit(0-9), so there 8 bytes for 8 digits.**
2. **Result = 1, then the save action is OK**