GPRS OTA Protocol

(for VT600)

V1.0

Content

1. Data Format

GPRS data format is as below：

From Server to Tracker:

@@<length(2bytes)><device ID (7bytes)><command (2bytes)><data><Checksum(2bytes)>\r\n

From Tracker to Server：

$$< length(2bytes))><device ID (7bytes>< command (2bytes)>< data >< Checksum (2bytes)>\r\n

Note:

'<' and '>' are the list separators, do not input '<' and '>' when writing a packet.

All multi-byte data complies with the sequence: High byte prior to low byte.

Item	Specification
@@	2 bytes. It is the header of packet from server to tracker. It is in ASCII code (Hex code is 0x40)
$$	2 bytes. It is the header of packet from tracker to server, It is in ASCII code (Hex code is 0x24)
L	2 bytes. It is the length of the whole packet including the header and ending character, it is in hex code
ID	7 bytes. It is in hex code, the unused byte will be stuffed by 'f' or '0xff'. It is in the format of hex code. For example, if ID is 12345678901, then it will be shown as follows: 0x12, 0x34, 0x56, 0x78, 0x90, 0x1f, 0xff.
Command	2 bytes. The command code is in hex code. Please refer to the command list below.
Data	Min 0 byte, max 130 bytes.
Checksum	2 bytes. It indicates CRC-CCITT(default is 0xffff) checksum of all data (not including checksum itself and the ending character). It is in hex code.

	For example: 24 24 00 11 12 34 56 78 90 1F FF 50 00 6F 35 0D 0A, which the checksum 0x6F35 = CRC-CCITT (24 24 00 11 12 34 56 78 90 1F FF 50 00)
\r\n	2 bytes. It is the ending character and in hex code (0x0d,0x0a)

2. OTA Command

1. Get tracker info – 0x9500

Command:	@@<L><Device ID><0x9500><cheksum>\r\n
Description:	Get tracker info，to check if tracker is compatible for the OTA protocol
Example:	40 40 00 11 10 11 30 92 36 33 FF 95 00 3F 9C 0D 0A
Response:	$$<L><Device ID><0x9500><device code (2 bytes)><Maximum length of OTA data (2bytes)><firmware version><S/N><IMEI><checksum>\r\n
Description:	**Device Code**：code of tracker type eg. VT310 code is 0x0010； Maximum length of OTA data: tracker's maximum length of OTA data, for example, VT600's maximum length is 0x0080, 128byte. Platform will send data as per tracker's maximum length Firmware version: support VT600 firmware V1.38 or above **S/N**：tracker device ID **IMEI**：tracker IMEI number
Sample	24 24 00 1A 10 11 30 92 36 33 FF 95 00 00 10 00 80 56 31 2E 33 38 27 0E 0D 0A

2. Device OTA Authorization– 0x9501

Command:	@@<L><Device ID><0x9501><checksum>\r\n
Description:	Server send GPRS command to tracker to enter in OTA status. Device will exit OTA status if no receive OTA data package in 20mins. Then you need reauthorize it. Tracker will stop sending location data till tracker exit OTA status
Sample:	40 40 00 11 10 11 30 92 36 33 FF 95 01 2F BD 0D 0A
Response:	$$<L><Device ID><0x9501><flag(1byte)> <checksum>\r\n

Description:	**Flag**： = 0x01，Authorization successful，server could send OTA data package = 0x00，Authorization failed, need reauthorization.
Sample:	24 24 00 12 10 11 30 92 36 33 FF 95 01 01 39 96 0D 0A

3. Server send OTA data package – 0x9502

Command:	@@<L><Device ID><0x9502><OTA data package address offset (4bytes)><OTA Length (2bytes)>< OTA data package><checksum>\r\n
Description:	Server send OTA data package OTA data package address offset：the address offset of the OTA data package in the whole data, 4bytes, High byte prior OTA length: 2bytes, the length of OTA data package OTA Data package: N bytes of the OTA data package
Sample:	40 40 00 97 10 11 30 92 36 33 FF 95 02 00 00 00 00 00 80 1C FC A0 D4 94 34 7C 7C C4 C4 BC 6C 34 D4 63 DA 02 2A 25 74 34 D4 06 D8 88 20 13 66 CE FE 33 A8 D8 10 A6 C4 5C 3C 9B D2 9C 72 BD CC 5C 3C 9B D2 CA 52 9D AC 3C 1C FB 32 2A B2 49 82 60 D8 87 A6 B6 06 2A 3F E8 80 FF DE BE 4E 4A 1B C8 60 DF BE 9E 2E 59 C0 68 00 7F 5E 3E CE F9 60 08 A0 9F 7E 5E EE 99 00 A8 40 3F 1E FE 8E 39 A0 48 E0 5F 3E 1E AE D9 40 E8 80 5E 04 9C 7C B0 5A 08 A0 9F 7E A9 3E 0D 0A
Response:	$$<L><device ID><0x9502><OTA data package address offset (4bytes)><flag(1byte)> <checksum>\r\n
Description:	OTA data package address offset：address offset of the data package from corresponding server **Flag:** = 0x01，received and stored correctly, server could send next OTA data package = 0x00，data receive failed, server need send the current OTA data package again = 0xFF，OTA unauthorized
Sample:	24 24 00 16 10 11 30 92 36 33 FF 95 02 00 00 00 00 01 FA 25 0D 0A

4. OTA data check – 0x9503

Command:	@@<L><device ID><0x9503><OTA data length(2bytes)><checksum>\r\n
Description:	After sever send all the OTA data package, check if the data received in tracker is correct. OTA Data length: the length of all OTA datum (not include the 2bytes of OTA checksum), 2bytes, high data prior
Sample:	40 40 00 13 10 11 30 92 36 33 FF 95 03 E1 00 2B A7 0D 0A
Response:	$$<L><Device ID><0x9503><OTA Checksum(2bytes)><checksum>\r\n
Description:	**OTA data checksum** :it is the checksum of all OTA datum. If the checksum replied from tracker is same as the last two bytes of OTA data package, then all the datum received in tracker is correct. Then server could send 0x9504 to let tracker upgrade with the new firmware. If the checksum replied from tracker is different, server need cancel the OTA command and do the whole procedure again.
Sample:	24 24 00 13 10 11 30 92 36 33 FF 95 03 B9 79 24 75 0D 0A

5. Tracker upgrade with OTA firmware – 0x9504

Command:	@@<L><device ID><0x9504><checksum>\r\n
Description:	Request tracker to upgrade the firmware data after server receive right checksum replied from tracker. Note: only send this command after server receive right checksum
Sample:	40 40 00 11 10 11 30 92 36 33 FF 95 04 7F 18 0D 0A
Response:	$$<L><Device ID><0x9504><Flag(1byte)><checksum>\r\n
Description:	**Flag：** = 0x01，tracker received SMS command and going to upgrade firmware = 0x02，tracker firmware upgrade finished = 0xFF，OTA unauthorized
Sample:	24 24 00 12 10 11 30 92 36 33 FF 95 04 01 C6 63 0D 0A 24 24 00 12 10 11 30 92 36 33 FF 95 04 02 F6 00 0D 0A

6. Cancel OTA – 0x9505

Command:	@@<L><Device ID><0x9505><checksum>\r\n
Description:	OTA authorized, sever could cancel the command of OTA
Sample:	40 40 00 11 10 11 30 92 36 33 FF 95 05 36 19 0D 0A
Response:	$$<L><Device ID><0x9505><flag(1byte)><checksum>\r\n
Description:	**Flag：** ＝0x01，OTA command cancel successful；
Sample:	24 24 00 12 10 11 30 92 36 33 FF 95 05 01 36 19 0D 0A

3. OTA procedure demo:

Follow up following OTA procedure：

0x9500 -> 0x9501 -> 0x9502 -> 0x9503 -> 0x9504

Sample of OTA data package：

1C FC A0 D4 94 34 7C 7C C4 C4 BC 6C 34 D4 63 DA 02 2A 25 74 34 D4 06 D8 88 20 13 66 CE FE 33 A8 D8 10 A6 C4 5C 3C 9B D2 9C 72 BD CC 5C 3C 9B D2 CA 52 9D AC 3C 1C FB 32 2A B2 49 82 60 D8 87 A6 B6 06 2A 3F E8 80 FF DE BE 4E 4A 1B C8 60 DF BE 9E 2E 59 C0 68 00 7F 5E 3E CE F9 60 08 A0 9F 7E 5E EE 99 00 A8 40 3F 1E FE 8E 39 A0 48 E0 5F 3E 1E AE D9 40 E8 80 5E 04 9C 7C B0 5A 08 A0 9F 7E 5E EE 99 00 A8 40 3F 1E FE 8E 39 A0 4C 26 6F 09 17 FC 0F D3 6B F2 7B EC F5 75 1D 42 2A 87 B6 7B C9 AF F2 63 7F 9A 1E 22 07 7C 8A 9A 2A 45 64 04 A4 44 EE EA EF 46 6A CA FD 7E C8 47 D2 D7 D5 B8 92 B2 BD 7B 27 48 6C 27 A7 95 B3 CA F4 58 2B 5D D4 56 39 5E 2F 96 A0 EE F3 12 C3 D1 C2 4E AB 72 DC 72 AF 6E 6D EF 00 F3 0A 34 13 B9 A1 BE 4B 62 86 D7 C4 2B F2 5C B1 E4 CE 21 67 6B 22 94 9B 9C

……

……

39 47 D6 B1 57 10 5A C1 8F 65 CD EA 3A 04 4E 1E 0C AB 53 4F E8 69 F4 41 E3 8F A6 B8 3B EF E2 06 51 6F 86 98 1C 29 83 28 C0 7B 94 F7 9C BF 9E B3 0C EA 0F C5 90 3B 95 CB 0D 91 18 A2 50 C2 AC FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA DC 82 52 A2 72 C2 92 E2 B2 02 D2 22 F2 42 12 62 B9 79

Note: we only get head and end data as OTA data package sample, the last two bytes 0xB9 0x79 is the checksum of OTA data, which no need send to tracker. The OTA Length do not include checksum

Step 1: send 0x9500 to read tracker's version

Data from server：40 40 00 11 10 11 30 92 36 33 FF 95 00 3F 9C 0D 0A

Data from tracker：24 24 00 1A 10 11 30 92 36 33 FF 95 00 00 10 00 80 56 31 2E 33 38 27 0E 0D 0A

Step 2: send 0x9501 for authorization

Data from server：40 40 00 11 10 11 30 92 36 33 FF 95 01 2F BD 0D 0A

Data from tracker：24 24 00 12 10 11 30 92 36 33 FF 95 01 01 39 96 0D 0A

Step 3: send 0x9502 to transmit OTA data package

Data from server：40 40 00 97 10 11 30 92 36 33 FF 95 02 00 00 00 00 00 80 1C FC A0 D4 94 34 7C 7C C4

C4 BC 6C 34 D4 63 DA 02 2A 25 74 34 D4 06 D8 88 20 13 66 CE FE 33 A8 D8 10 A6 C4 5C 3C 9B D2 9C 72 BD CC 5C 3C 9B D2 CA 52 9D AC 3C 1C FB 32 2A B2 49 82 60 D8 87 A6 B6 06 2A 3F E8 80 FF DE BE 4E 4A 1B C8 60 DF BE 9E 2E 59 C0 68 00 7F 5E 3E CE F9 60 08 A0 9F 7E 5E EE 99 00 A8 40 3F 1E FE 8E 39 A0 48 E0 5F 3E 1E AE D9 40 E8 80 5E 04 9C 7C B0 5A 08 A0 9F 7E A9 3E 0D 0A

Data from tracker：24 24 00 16 10 11 30 92 36 33 FF 95 02 00 00 00 00 01 FA 25 0D 0A

Note: the address offset in the first OTA data package is 00 00 00 00 , OTA length is 00 80, the data in red color is OTA data package.

Step 4: repeat step 3 till finish the whole transmitting of OTA data package

Data from server：40 40 00 97 10 11 30 92 36 33 FF 95 02 00 00 00 80 00 80 5E EE 99 00 A8 40 3F 1E FE 8E

39 A0 4C 26 6F 09 17 FC 0F D3 6B F2 7B EC F5 75 1D 42 2A 87 B6 7B C9 AF F2 63 7F 9A 1E 22 07 7C 8A 9A 2A 45 64 04 A4 44 EE EA EF 46 6A CA FD 7E C8 47 D2 D7 D5 B8 92 B2 BD 7B 27 48 6C 27 A7 95 B3 CA F4 58 2B 5D D4 56 39 5E 2F 96 A0 EE F3 12 C3 D1 C2 4E AB 72 DC 72 AF 6E 6D EF 00 F3 0A 34 13 B9 A1 BE 4B 62 86 D7 C4 2B F2 5C B1 E4 CE 21 67 6B 22 94 9B 9C CF EB 0D 0A

Data from tracerr：24 24 00 16 10 11 30 92 36 33 FF 95 02 00 00 00 80 01 E1 BD 0D 0A

……

……

Data from server：40 40 00 97 10 11 30 92 36 33 FF 95 02 00 00 E0 80 00 80 39 47 D6 B1 57 10 5A C1 8F 65

CD EA 3A 04 4E 1E 0C AB 53 4F E8 69 F4 41 E3 8F A6 B8 3B EF E2 06 51 6F 86 98 1C 29 83 28 C0 7B 94 F7 9C BF 9E B3 0C EA 0F C5 90 3B 95 CB 0D 91 18 A2 50 C2 AC FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA 8A 9A 2A 3A CA DA 6A 7A 0A 1A AA BA 4A 5A EA FA DC 82 52 A2 72 C2 92 E2 B2 02 D2 22 F2 42 12 62 ED BD 0D 0A

Data from tracker：24 24 00 16 10 11 30 92 36 33 FF 95 02 00 00 E0 80 01 41 8C 0D 0A

Step 5: send 0x9503 to get OTA data checksum from tracker

Data from server：40 40 00 13 10 11 30 92 36 33 FF 95 03 E1 00 2B A7 0D 0A

Data from tracker：24 24 00 13 10 11 30 92 36 33 FF 95 03 B9 79 24 75 0D 0A

Note：E1 00 is the full length of whole OTA data（not including OTA last two bytes' checksum B9 79），checksum replied from tracker is same as the last two bytes data of OTA data.

Step 6: send 0x9504 to start firmware upgrade

Data from Server：40 40 00 11 10 11 30 92 36 33 FF 95 04 7F 18 0D 0A

Data from Tracker：24 24 00 12 10 11 30 92 36 33 FF 95 04 01 C6 63 0D 0A

Data from tracker after finish firmware upgrade：24 24 00 12 10 11 30 92 36 33 FF 95 04 02 F6 00 0D 0A