**两种方式跟服务器通讯：**

**第一种：GPRS数据包，直接往server：track.willcheer.com发数据。**

From server to tracker:

**@@<package flag><L>,<IMEI>,<command>,<data><\*checksum>\r\n**

From tracker to server:

**$$<package flag><L>,<IMEI>,<command>,<data><\*checksum>\r\n**

**详细见第二页。**

注意事项：

Note: $$<packageflag><L>,<IMEI>,<command>,<event code>,<(-)yy.dddddd>,<(-)xxx.dddddd>,<yymmddHHMMSS>,<Z>,<N>,<G>,<Speed>,<Heading>,<HDOP>,<Altitude> ,<Journey>,<Runtime>,<Base ID>,<State>,<AD>,<RFID>/<Picture>/<Fence>/<Temperature Sensor Index No..>,<Customize Data>,<Protocol Version>,<Fuel Percentage>,<Temperature Sensor Index No.1 Value|Temperature SensorIndex No.2 Value|......Temperature Sensor Index No.n Value><\*checksum>\r\n

Note: ‘,’ is list separator in ASCII (0x2C); Do not input ‘<’ and ‘>’ when writing a command; All multi-byte data complies with the following sequence: High byte prior to low byte; GPRS package (including data) flow is about 160 bytes.

第二种用WEB地址模式（简易模式），OsmAnd Live Tracking web address

Format:

http://track.willcheer.com:5055/?id=123456&lat={0}&lon={1}&timestamp={2}&hdop={3}&altitude={4}&speed={5}

where:

* track.willcheer.com – server address or domain name
* 123456 – your device unique identifier (e.g., IMEI number，或者身份识别)

**Description of GPRS Data**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Parameter |  |  | Description |  |  | Example |
| **@@** | | | 2 bytes. Header of the package from server to tracker. It is in ASCII(0x40) | | | @@ | |
| **$$** | | | 2 bytes. Header of the package from tracker to server, It is in ASCII(0x24) | | | $$ | |
| **package flag** | | | 1 byte. In ASCII from 0x41 to 0x7A | | | Q | |
| **L** | | | Length from its following separator ‘,’ to the ending character ‘\r\n’. It is decimal. $$<package flag><L>,<IMEI>,<command>,<data><\*checksum>\r\n | | | 25 | |
| **IMEI** | | | Tracker ’s IMEI is normally 15 digitals. | | | 353358017784062 | |
| **command** | | | Command code in Hex string. Please refer to the Command List and Command Details below. | | | AAA | |
| **Code** | | | Event code. Decimal. | | | 1 | |

|  |  |  |
| --- | --- | --- |
| **<->xxx.dddddd** | Longitude: in unit of degree. Decimal. ‘-’means west, no minutes means east xxx = degrees; dddddd = decimal part of degree | 114.752146 -114.821453 |
| **yymmddHHMMSS** | yy = year mm = month dd = date HH = hour MM = minute SS = second Decimal digit | 091221102631 |
| **Z** | GPS status indicator: A = valid, V = invalid | A = Valid |
| **N** | Numbers of satellites available. Decimal. | 5 |
| **G** | GSM signal. Decimal(0~31) | 12 |
| **Speed** | KM/h. Decimal. | 58 |
| **Heading** | Heading, in unit of degree. Decimal.(0~359) | 275 |
| **HDOP** | Horizontal Dilution of Precision, 0.5-99.9. Decimal. HDOP Values below 4 are great and value above 8 are bad HDOP is blank when no GPS fix. | 5 |
| **Altitude** | MSL Altitude, in unit of meter. Decimal. | 118 |
| **Mileage** | In unit of meter. Decimal. The total accumulated mileage and maximumof 4294967295 meters. | 564870 |
| **Runtime** | In unit of second. Decimal. The total accumulated runtime and maximum4294967295 seconds. | 2546321 |
| **Base ID** | ID of the base station including MCC|MNC|LAC|CI Note: for SMS report, the Base ID is empty. MCC and MNC are decimal; LAC and CI are Hex. | 460|0|E166|A08B |
| **State** | Status of 8 inputs and 8 outputs. Hex. Bit0...Bit7 is output state, Bit0 is Ouput1 state Bit8...Bit15 is input state, Bit8 is Input1 state | 0421(HEX String) = 0000010000100001 |
| **AD** | Separated by ‘|’. Hex. AD1|AD2|AD3|Battery AD|External Power AD | 123|456|235|1234|324| 654|1456|222(HEX String) |

|  |  |  |
| --- | --- | --- |
| **RFID** | IC Card identity code. Hex. Only shown in GPRS Event Code 37 | 42770680(HEX String) |
| **Picture** | Picture name Only shown in GPRS Event Code 39 | 0918101221\_C2E03 |
| **Fence** | Fence Number Only shown in Event Code 20 and 21 | 2 |
| Temperature Sensor Index No. | The Index No. is set by command C40 Format: 2 Hex. Note: Only event 50,51 contain Temperature Sensor Index No. | 08 Means Temperature SensorIndex No.8 |
| Customize Data | Reserved separator remained |  |
| Protocol Version | Default: empty Format: Decimal = empty, for MT90/MVT100/ MVT340/MVT380/TC68S, and T1/MVT600/ MVT800 without temperature and fuel sensors =1, for T1/MVT600/ MVT800 with temperature and fuel sensor =2-49, reserved for general Meitrack protocl | 1 Means:protocol version with Fuel Percentage and Temperature value |
| Fuel Percentage | Fuel Percentage Format: 4 Hex.High byte is the Integerbit of the percentage,lower byte is the decimal of the percentage. Fuel type=0, no Fuel sensor, value is empty. | 241E Means fuel percentage: 36.30%. |
| Temperature Index No. &Value | Temperature Index No. Format: 6 Hex. The highest byte is the Index No. of sensor. The middle byte is the integer of temperature (-127~+127) Lower bytes is the decimal of the temperature. | 011A09|021A15|061E20 Means 3 temperature sensors, numbers are 1,2, and 6; Temperature values are 26.09°C,26.21°C,30.32°C |

|  |  |  |
| --- | --- | --- |
| **\*** | 1 byte. A separator between data and checksum.It is in ASCII(Hex0x2A) | \* |
| **checksum** | 2 bytes. Indicating the sum of all data (exclude checksum itself and the ending character). It is in HEX String.  $$<package flag><L>,<IMEI>,<command>,<data><\*checksum>\r\n | BE |
| **\r\n** | 2 bytes. Ending character in ASCII (0x0d,0x0a) | \r\n |