

IPS Protocol Frame Format

Applicable Model: BTT One



Change History

Filename	IPS Protocol Frame Format	Created By	Faiiz
Project	BTT One	Creation date Update date	10 – 09 – 2020
Version	V1.0		

Document History

Revision	Date	History

General Structure of TCP Messages

- All data is received in text format as a packet which looks as follows:

```
#PT#msgCRC\r\n
```

Parameter	Description
#	Start byte
PT	Packet type (see the <i>Packet types</i> table)
#	Delimiter
Msg	Text of the message
CRC	CRC16 checksum
\r\n	End of the packet (0x0D0A in HEX)

Packet Types

Type	Description	Sender
L	Login packet	Device
AL	Answer to the login packet	Server
D	Extended data packet	Device
AD	Answer to the extended data packet	Server
C	Send command to device	Server
M	Response command from the device	Device
AM	Answer to the response command from device	Server

Login Packet

- The packet is used for the device authorization on the server. Every TCP connection starts with sending this packet from the device to the server. Other data should be transferred only after the server confirms the successful authorization of the device.
- The login package looks as follows:

```
#L#Protocol_version;IMEI;Password;CRC16\r\n
```

Parameter	Description
L	Packet type: login packet.
Protocol_version	Current protocol version. In this case, 2.0.
;	Delimiter
IMEI	IMEI, ID or serial number of the controller.
Password	Password to access the device. If there is none, NA is transmitted.
CRC16	Checksum. See the <i>Checksum</i> section.

Server Response to the L Packet

Type	Code	Meaning	Example
AL	1	Unit successfully authorized.	#AL#1\r\n
	0	Connection rejected. Possible reasons: <ul style="list-style-type: none"> • Incorrect protocol version. The current one is 2.0; • The unit is not created on the server; • Incorrect packet structure. 	#AL#0\r\n

Extended Data Packet

- The packet contains additional data structures and looks as follows:

```
#D#Date;Time;Lat1;Lat2;Lon1;Lon2;Speed;Course;Alt;Sats;HDOP;Inputs;
Outputs;ADC;lbutton;Params;CRC16\r\n
```

Parameter	Description
D	Packet type: extended data packet.
Date	Date in the DDMMYY format, UTC±00:00. If there is no data, NA is transmitted.
Time	Time in the HHMMSS format, UTC±00:00. If there is no data, NA is transmitted.
Lat1;Lat2	Latitude. If there is no data, NA is transmitted. See <i>Annex</i> .
Lon1;Lon2	Longitude. If there is no data, NA is transmitted. See <i>Annex</i> .
Speed	Speed value, integer (km/h). If there is no data, NA is transmitted.
Course	Direction of movement, integer (from 0 to 359 degrees). If there is no data, NA is transmitted.

Alt	Altitude, integer (meters). If there is no data, NA is transmitted.
Sats	Number of satellites, integer. If there is no data, NA is transmitted.
HDOP	Horizontal Dilution of Precision. It shows the accuracy of the coordinates transmitted by the device. The smaller this value is the more accurate the coordinates are. If there is no data ,NA is transmitted.
Inputs	Digital inputs. Every bit of the number (beginning from the low-order one) corresponds to one input. Integer. If there are none, NA is transmitted.
Outputs	Digital outputs. Every bit of the number (beginning from the low-order one) corresponds to one output. Integer. If there are none, NA is transmitted.
ADC	Analogue inputs. Fractional numbers separated by commas. Numbering from 1. If there are none, an empty string is transmitted. Example: 14.77,0.02,3.6
Reserve	NA is transmitted.
Params	Additional parameters. Separated by commas. See <i>Additional Parameters</i> .
CRC16	Checksum. See the <i>Checksum</i> section.

Additional Parameters (Params)

- Each parameter has the following structure:

Name:Type:Value

Examples of additional parameters: count1:1:564, fuel:2:45.8, hw:3:V4.5, SOS:1:1

Parameter	Description
Name	<p>Parameter name. In lowercase. The maximum number of characters is 38.</p> <p>hdop : Hdop fix : gps fix type (0=Nofix , 2=2D fix , 3=3D fix) csq : gsm signal level dist : total distance event : event temp1 : temperature channel 1 pwr_ext : External power voltage pwr_batt : Battery power voltage imei : device imei option : external device connect (NULL=Not connect) driver : DLT driver (1=Connected , 0=Not Connect) T1 : DLT card track 1 (show when DLT driver is connected) T2 : DLT card track 2 (show when DLT driver is connected) T3 : DLT card track 3 (show when DLT driver is connected)</p>
Type	<p>Parameter type:</p> <p>1 — Integer / Long; 2 — Double; 3 — String (the maximum number of characters: 1344).</p>
Value	<p>Parameter value. Depends on the parameter type.</p>

Server Response to the D Packet

Type	Code	Meaning	Example
AD	1	Packet successfully registered.	#AD#1\r\n
	0	Error receiving or incorrect packet	#AD#0\r\n

Send Command to Device

- The command is used to send command from server to device for action or configuration device:

```
#C#Command;\r\n
```

Parameter	Description
C	Packet type: Send Command to device.
Command	AT Command Interface See the <i>vehaa command interface guide</i> document.
;	End packet

Response Command from device

- The command is response command from device:

```
#M#Msg;CRC16\r\n
```

Parameter	Description
M	Packet type: Response command from the device.
Msg	Response command from device.
CRC16	Checksum. See the <i>Checksum</i> section.

Server Response to the M Packet

Type	Code	Meaning	Example
AM	1	Message received.	#AM#1\r\n
	0	Error receiving or incorrect packet	#AM#0\r\n

Checksum

- The CRC16 checksum should be added to the message as a hexadecimal number in ASCII characters. The byte order is big-endian.

Example: 0xFC45 => 0x46433435

Checksum Calculation

Packet type	Explanation
L	<p>Message example: #L#Protocol_version;IMEI;Password;CRC16\r\n</p> <p>The checksum is calculated for the following part of the packet: Protocol_version;IMEI;Password;</p>
D	<p>Message example: #D#Date;Time;Lat1;Lat2;Lon1;Lon2;Speed;Course;Alt;Sats;HDOP;Inputs;Outputs;ADC;lbutton;Params;CRC16\r\n</p> <p>The checksum is calculated for the following part of the packet: Date;Time;Lat1;Lat2;Lon1;Lon2;Speed;Course;Alt;Sats;HDOP;Inputs;Outputs;ADC;Reserve;Params;</p>
M	<p>Message example: #M#Msg;CRC16\r\n</p> <p>The checksum is calculated for the following part of the packet: Msg;</p>

C Code Example for CRC16 Calculation

```
static const unsigned short crc16_table[256] =
{
    0x0000,0xC0C1,0xC181,0x0140,0xC301,0x03C0,0x0280,0xC241,
    0xC601,0x06C0,0x0780,0xC741,0x0500,0xC5C1,0xC481,0x0440,
    0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCFC1,0xCE81,0x0E40,
    0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,
    0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDBC1,0xDA81,0x1A40,
    0x1E00,0xDEC1,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,
    0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,
    0xD201,0x12C0,0x1380,0xD341,0x1100,0xD1C1,0xD081,0x1040,
    0xF001,0x30C0,0x3180,0xF141,0x3300,0xF3C1,0xF281,0x3240,
    0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,
    0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,
    0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF9C1,0xF881,0x3840,
    0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,
    0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xEDC1,0xEC81,0x2C40,
    0xE401,0x24C0,0x2580,0xE541,0x2700,0xE7C1,0xE681,0x2640,
    0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,
    0xA001,0x60C0,0x6180,0xA141,0x6300,0xA3C1,0xA281,0x6240,
    0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,
    0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,
    0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA9C1,0xA881,0x6840,
    0x7800,0xB8C1,0xB981,0x7940,0xBB01,0x7BC0,0x7A80,0xBA41,
    0xBE01,0x7EC0,0x7F80,0xBF41,0x7D00,0xBDC1,0xBC81,0x7C40,
    0xB401,0x74C0,0x7580,0xB541,0x7700,0xB7C1,0xB681,0x7640,
    0x7200,0xB2C1,0xB381,0x7340,0xB101,0x71C0,0x7080,0xB041,
    0x5000,0x90C1,0x9181,0x5140,0x9301,0x53C0,0x5280,0x9241,
    0x9601,0x56C0,0x5780,0x9741,0x5500,0x95C1,0x9481,0x5440,
    0x9C01,0x5CC0,0x5D80,0x9D41,0x5F00,0x9FC1,0x9E81,0x5E40,
    0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x99C0,0x9880,0x9841,
    0x8801,0x48C0,0x4980,0x8941,0x4B00,0x8BC1,0x8A81,0x4A40,
    0x4E00,0x8EC1,0x8F81,0x4F40,0x8D01,0x4DC0,0x4C80,0x8C41,
    0x4400,0x84C1,0x8581,0x4540,0x8701,0x47C0,0x4680,0x8641,
    0x8201,0x42C0,0x4380,0x8341,0x4100,0x81C1,0x8081,0x4040
};

unsigned short crc16 (const void *data, unsigned data_size)
{
    if (!data || !data_size)
        return 0;
    unsigned short crc = 0;
    unsigned char* buf = (unsigned char*)data;

    while (data_size--)
        crc = (crc >> 8) ^ crc16_table[(unsigned char)crc ^ *buf++];

    return crc;
}
```

Annex

The coordinates are compliant with the NMEA 0183 standard.

DDMM.MM is the format of latitude. Two digits of degrees (DD). If the degree value consists of one digit, the degree field still contains two digits. That is the field is filled with zeros.

For example, 01. The degrees are followed by two digits of integer minutes, a point, and a fractional part of minutes of variable length. The leading zeros are not omitted. N denotes north (positive) latitude, S denotes south (negative) latitude.

Example: 5544.6025;N

55 is a degree value.

44.6025 / 60 = 0,743375 is a minute value.

N is north latitude (positive sign).

$55 + 0,743375 = +55,743375$

DDDMM.MM is the format of longitude. Three digits of degrees (DDD). If the degree value consists of one digit, the degree field still contains three digits. That is the field is filled with zeros.

For example, 001. The degrees are followed by two digits of integer minutes, a point, and a fractional part of minutes of variable length. The leading zeros are not omitted. E denotes east (positive) longitude, W denotes west (negative) longitude.

Example: 03739.6834;E

037 is a degree value.

39.6834 is a minute value.

E is east longitude (positive sign).

$037 + 39.6834 = +37,66139$