Protocol Frame Format

Applicable Model: BTT One

AERIAL™
Communications Co., Ltd.

Change History

Filename	Protocol Frame Format	Created By	Faiiz
Project	BTT One	Creation date Update date	12 – 12 – 2019 11 – 06 – 2020
Version	V1.7		

Document History

Revision	Date	History
003	March 19, 2020	Update binary protocol ➢ Analog input value ➢ Temperature value ➢ Battery voltage value ➢ External power voltage value Update description
004	May 7, 2020	Update binary protocol ➢ Digital input value
005	May 11, 2020	Update external sensor protocol ➢ Example camera snapshot ➢ 14 input channels protocol ➢ Temperature module protocol
006	May 22,2020	Update external sensor protocol ➢ 14 input channels protocol
007	June 11,2020	➢ Fix satellite in use description ➢ Fix latitude , longitude calculation

Frame Format

- TCP frame format sent from device to server

Original
$<Original Binary>#

With Card Reader
$<Original Binary>#$ <Length-Decimal 4 Bytes><DriverLicense String Data Begin With '%'>#

With External Sensor
$<Original Binary># $<Length-Decimal 4 Bytes><JSON String Data>#

With Card Reader & External Sensor
$<Original Binary>#$ <Length-Decimal 4 Bytes><DriverLicense String Data Begin With '%'>#$<Length-Decimal 4 Bytes><JSON String Data>#

Note:

- Symbols "<" and ">" will not be present in actual data , only for documentation purpose only.
- The size of original binary data packet is 49 bytes.
- External 14 input channel sensor will be present in original format (see detail below)

Original Binary Protocol

Parameter	Description	Example
$	Header sent from device to server. Type is ASCII. (Hexadecimal as 0x24)	$
Data length	Length of binary from header to end . Hexadecimal 1 Byte.	0x31
Version	Version of firmware . Hexadecimal 1 Byte.	0x01
IMEI	Device IMEI . Little endian decimal 8 byte.	imei is 353358017784062 data in frame is 6240781780355303

Satellite in use and fix status	Hexadecimal 1 Byte. Bit0 to Bit4 corresponds to number of received gps satellite Bit5 to Bit7 corresponds to fix status fix status (Hex) 00 : No fix 20 : Fix 2D 40 : Fix 3D	0x49 (hexadecimal) = 0100 1001 sat received = 01001 = 9 fix status = 010 = Fix 3D
Latitude	Little endian Hexadecimal 4 Byte. (See detail in table 1)	5AFBCD00
Longtitude	Little endian Hexadecimal 4 Byte. (See detail in table 1)	1503FC05
GPS Date DDMMYYyy	Hexadecimal 4 Byte. DD indicates date. MM indicates month. YY indicates year (first 2 digit). yy indicates year (last 2 digit).	12031410 indicates 18 March 2016
GPS Time HHMMSS	Hexadecimal 3 Byte. HH indicates hour. MM indicates minute. SS indicates second.	08350E indicates 08 : 53 : 14
Speed	Unit : Km/hr Hexadecimal 1 Byte.	0x64 indicates 100 km/hr
Direction	Little endian Hexadecimal 2 Byte. Indicates driving direction. Unit: degree. Value ranges from 0 to 359.	
Distance	Unit : Meter Hexadecimal 3 Byte. The maximum value is 1000000000. If value exceeds the maximum value, it will be automatically cleared.	
Input	Little endian Hexadecimal 2 Byte. Bit0 to Bit2 corresponds to status of input port 1 to 3 Bit3 to Bit15 reserve when external 14 input channel sensor connected. Bit0 to Bit1 corresponds to status of input port 1 to 2 Bit2 to Bit15 corresponds to status of external 14input sensor channel 1 to 14	0x07 (Hexadecimal) = 0000 0000 0000 0111
Analog input value	Little endian Hexadecimal 2 Byte. Value / 1000	

Temperature value	Little endian Hexadecimal 2 Byte. Value less than 0x0800 Temperature= Value / 16 Value equal or more 0x0800 Temperature=(Value-0x0800) / 16 Temperature range: -127°C to +127°C	
Battery Voltage	Little endian Hexadecimal 2 Byte. Value / 1000	
External power Voltage	Little endian Hexadecimal 2 Byte. Value / 1000	
Output	Hexadecimal 1 Byte. Bit0 to Bit1 corresponds to status of input port 1 to 2 Bit2 to Bit7 reserve	0x03 (Hexadecimal) = 0000 0011
Event	Hexadecimal 1 Byte. (See detail in table 2)	
Reserve	Reserve	
Reserve	Reserve	
Reserve	Reserve	
#	Ending character sent from device to server. Type is ASCII. (Hexadecimal as 0x23)	#

Example data send to server

24 31 20 70 84 93 60 60 26 52 03 47 57 49 CE 00 7D 01 FA 05 09 08 14 13 0A 05 0E 00 00 00 00 8C
33 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 23

Table 1. Latitude and Longitude

Format	Description
00000000 (4 byte)	Ex. 5AFBCD00 = 0x00CDFB5A (hex) 0x00CDFB5A (hex) = 13499226 (decimal) Value = 13499226 / 1000000 = 13.499226 Decimal Degree Format Decimal = Value in integer = 13 Degree = ((Value - Value in integer) *100000000) / 60 (13.499226 – 13) *1000000 = 49922600/60 = 832043 Decimal (integer) = 13 Degree (integer) = 832043 **Latitude = 13.832043 (DD Format)** Degree, minutes seconds Format Degree = Value in integer = 13 Minutes = ((Value - Value in integer) *100 (13.499226 – 13) *100 = 49.9226 Minutes in integer = 49 Seconds = (Minutes – Minutes in integer) * 60 (49.9226 – 49) * 60 = 55.35 Seconds = 55.35 **Latitude = N 13° 49’ 55.35” (DMS Format)**
	Ex. 1503FC05 = 0x05FC0315 (hex) 0x05FC0315 (hex)= 100401941 (decimal) Value = 100401941 / 1000000 = 100.401941 Decimal Degree Format Decimal = Value in integer = 100 Degree = ((Value - Value in integer) *100000000) / 60 (100.401941 – 100) *1000000 = 40194100/60 = 669901 Decimal (integer) = 100 Degree (integer) = 669901 **Longitude = 100.669901 (DD Format)** Degree, minutes seconds Format Degree = Value in integer = 13 Minutes = ((Value - Value in integer) *100 (100.401941 – 100) *100 = 40.1941 Minutes in integer = 40 Seconds = (Minutes – Minutes in integer) * 60 (40.1941 – 40) * 60 = 11.64 Seconds = 11.64 **Latitude =E 100° 40’ 11.64” (DMS Format)**

Table 2. Event

Event code	Event
0x10	Driving
0x20	IO changed
0x30	Overspeed
0x40	Parking (Engine off)
0x41	Idle (Engine on)
0xD4	Card login
0xD5	Card logout
0x81	GPS loss
0x99	Power loss
0x98	Power Enter

Card reader Protocol

Item	Fotmat	Example	Description
Start header	Ascii		$
length	Ascii	0149	Data 149 byte
Text of card	Ascii		
End frame	Ascii		#

Text of card

Example:

% ^DRIVING LICENSE$TEST$MR.^^?;6007641111111111119=180919770411=?
+ 11 1 99999580 10001 ?

Data track 1: starts with "%" and ends with "?" This track contains 3 type of content fields.

Sample raw data	Start delimiter	End delimiter	Field length	Meaning
DRIVING LICENSE	^	$	flexible	Surname
TEST	$	$	flexible	Name
MR.	$	^^	flexible	Prefix

Data track 2: starts with ";" and ends with "?" This track contains 2 type of content fields.

Sample raw data	Start delimiter	End delimiter	Field length	Meaning
600764 (fix)	;		6	Thailand (Fix)
1111111111119		=	13	Thai ID card number
1809	=		4	Expiration date (YY MM)
19770411		=	8	Birthday (YYYY MM DD)

Data track 3: starts with "+" and ends with "?" This track contains 4 type of content fields.

Sample raw data	Start delimiter	End delimiter	Field length	Meaning
11	blank	Blank	2 or 4	Type of Driving License
1	blank	Blank	1	Sex
99999580	Blank	blank	8	ID Driving License
10001	Blank	blank	5	Province

External Sensor Protocol

Item	Fotmat	Example	Description
Start header	Ascii		$
length	Ascii	0149	Data 149 byte
Text of sensor	Ascii		
End frame	Ascii		#

Text of sensor

- camera snapshot protocol

  {"IMG00": "filename"}

  Find the file in following directory: *…/imei of device/image-filename.JPG*

  Example:

  $0027{"IMG00": "20200510123050"}#

  Image path:

  http://kkj.aerialcoms.com:3000/images/358887093585618/image-20200510123050.JPG

- 14 input channels protocol (standard firmware)

  {"EXP14IN":version,"IN1":status,"IN2":status,"IN3":status,"IN4":status,"IN5":status, "IN6":status,"IN7":status,"IN8":status,"IN9":status,"IN10":status,"IN11":status, "IN12":status, "IN13":status,"IN14":status,"SUM":"checksum"}

Item	Description
status	0 : inactive , 1 : active

- 14 input channels protocol (rotation firmware)

  {"EXP12RT":version,"IN1":status,"IN2":status,"IN3":status,"IN4":status,"IN5":status, "IN6":status,"IN7":status,"IN8":status,"IN9":status,"IN10":status,"IN11":status, "IN12":status,"RT":direction,"TM":time,"RR":round,"SUM":"checksum"}

Item	Description
status	0 : inactive , 1 : active
direction	0 : Stop , 1 : Left , 2 : Right
time	Time per round
round	Rotation round (Max : 1000)

- Temperature module with display protocol

  {"EXP1W":version,"T1":value,"T2":value,"T3":value,"F1A":value,"F1D":value,"F2A":value, "F2D":value,"SUM":"checksum"}

  (See detail in Temperature module with display user guide)

Acknowledge from server

When server already received each data packet , Server must send response text "oK" or "ERR" for acknowledged. If server send "ER" , device will send that data packet again.