Protocol Manual
OBDV2_V1.5

Description: Protocol manual for OBDV2	Document Name: Protocol_Manual_OBDV2_V1.5

REVISION HISTORY

S. NO	REV. NO	DATE	CHANGE DESCRIPTION	Revised By
1	1.0	17-09-2021	First Version	Pavan Naik
2	1.1	27-09-2021	1. Added Trip Time in protocol	Prasanth Ravi
3	1.2	05-01-2022	1. Added an extra comma in login packet 2. Added | just before * in Tracking Packet 3. Added | as last character of the packet in PID Data string 4. Added e.g. data in PID data section. 5. Added e.g. data in VIN data section 6. Added e.g. data in DTC data section 7. Modified server command section 8. Removed NAP event in Event code section.	Prasanth Ravi
4	1.3	20-01-2022	1. Added Accelerometer data in Tracking packet 2. Added The Tracking data when PKT_PID_IGNFILT is enabled	Prasanth Ravi
5	1.4	25-01-2022	1. Modified the Event code list 2. Edited the Protocol format	Prasanth Ravi
6	1.5	05-03-2022	1. Added DTC condition in OBD data section. 2. Added DTC decoding.	Prasanth Ravi

PREPARED BY	VERIFIED BY	APPROVED BY
Prasanth Ravi	Pavan Naik	Kiran A R

Contents

1   Introduction

This document explains the communication protocol between the OBD V2 tracking device and the tracking Server. This document defines specifics for communication such as communication packet formats and data type of the information which are being transferred.

2   Intended Audience

This document is intended for reference for system designers, customers who wish to integrate the OBD V2 device with their applications.

3   Communication Protocol

The communication between the OBD V2 and Tracking server is socket based single threaded client server communication on TCP/IP. OBD V2 is considered as client and initiates the communication request once and tries keeps the connection alive. On every transmission, OBD V2 push the data on destination server on the established connection or request for new connection in case of no connection. Different Packet Formats are as explained in the following sections.

4   Packet Format

    a)   Login packet (Connection packet)
    b)   Tracking format

4.1   Login Packet

This packet is the first packet which is sent to the server after successful connection with the server.

$$CLIENT_1NS,862843041050881,15,1_37OB04A05QREL_48_16,106.51.81.99,17672,internet,T1:10 S,T2:1 M,Ad1:0000000000,Ad2:0000000000,TOF:0 S,,OSC:100 KM,OST:10 S,GPS:NO,Ignition:ON,*6D

SL NO	String	Parameter	Description	Data Type
1	$$	Packet Header	$$ indicates beginning of the packet	String
2	CLIENT_1NS	Client ID		String
3	862843041050881	Serial Number/IMEI number of the Device	Unique Serial Number or IMEI of the device	Unsigned Integer
4	15	Event/ Message code	Refer event/ Message code Table:2	Unsigned Integer
5	1_37OB04A05QREL_48_16	Firmware Version	Firmware version number currently installed in the device	String

6	106.51.81.99	IP Address/DNS	IP address/DNS where the device is reporting to.	String
7	17672	Port Number	Port number where data is being sent	Unsigned Integer
8	Internet	APN	APN provided by the service provider	String
9	T1:10 S	Ignition ON reporting interval	Duration in which data is being sent during IGN ON	String
10	T2:1 M	Ignition OFF reporting interval	Duration in which data is being sent during IGN OFF	String
11	Ad1:0000000000	Admin number 1	Currently configured Admin 1 number	String
12	Ad2:0000000000	Admin number 2	Currently configured Admin 2 number	String
13	TOF:0 S	GMT Offset	GMT (00)/ IST (19800) time configured on the device	String
14	Reserved	Reserved bit	Reserved for future use	N/A
15	OSC:100 KM	Over speed Limit	Over speed configuration in the device	String
16	OST:10 S	Over speed Duration	Duration for which the Over speed is configured	String
17	GPS:YES	GNSS Fix Status	YES/NO	String
18	Ignition:ON	Ignition Status	OFF/ON	String
19	*	Data separator		Character
20	6D	Checksum	XOR of all data excluding data separator (*), checksum itself and ending characters. Value displayed in HEX	Integer

*All parameters are comma separated from each other in the data string except for the Header and client ID.

4.2   Tracking Packet format

4.2.1   Tracking packet When Accelerometer data is dis-abled

This packet will be sent to the server as per the recording interval that is configured in the devices.

This packet will contain both tracking and OBD related data.

$$CLIENT_1NS,862843041050881,1,12.976515,77.54966,211220112256,A,20,0,583,3,6,1.09,0,0,12181,2050,12181,3960,10023,21,1|0104:064104FB000000|010D:06410D15000000|010C:06410C2B4E0000|0101:064101000E8000|0102:06410200000000|0103:06410300005500|0105:0641057C000000|0106:06410600000000|0107:06410700000000|0108:06410800000000|0109:06410900000000|010A:06410A00000000|010B:06410B00000000|0111:06411100000000|0122:06412200000000|0133:0641335B000000|*66

SL NO	String	Parameter	Description	Data Type
1	$$	Package Header.	$$ indicates beginning of the packet	String
2	**CLIENT_1NS**	Client ID		String
3	**190803159**	Device serial number/IMEI number	Unique Serial Number or IMEI of the device	Unsigned Integer
4	101	Message/Event code (reason for packets)	Refer Message/Event Code Table:2	Unsigned Integer
5	**12.976515**	Latitude		Float
6	**77.54966**	Longitude		Float
7	**190920093311**	Date and time	In YYMMDDHHMMSS format.	Unsigned Integer
8	A	GNSS fix status	A= valid. V = invalid.	Character
9	**20**	GSM Signal	0 to 31. 99: Not known/not detectable	Unsigned Integer
10	0	Speed	With Decimal digit. In KM/H.	Unsigned Integer
11	583	Accumulated Distance	Accumulated Distance (Odometer based on GPS).	Unsigned Integer
12	3	Course in degree	This value represents the heading direction in degree value. (0-360degrees)	Unsigned Integer
13	6	Satellites being used	Shows the total number of satellites being used for Location points (0-24)	Integer
14	1.09	HDOP	Horizontal Dilution of Precision	Float
15	0	Reserved for expansion	Reserved	NA
16	0	Reserved for expansion	Reserved	NA
17	12181	Voltage equivalent of Analog I/P reading	Analog input value measured in mill volts	Unsigned Integer
18	2050	Event Flag	Displayed in decimal, this field can be converted to binary	Unsigned Integer

			to get the positional data for the 32 information fields as explained in the table below. (Please refer table:1)	
19	12181	External Battery Voltage	This value indicates the external battery voltage in millivolt. Here it is 12.267V	Unsigned Integer
20	3960	Internal Battery Voltage	This value represents the internal battery voltage in millivolt. Here it is 3.96V	Unsigned Integer
21	10023	Trip Time	Time Counter in Seconds	Unsigned Integer
22	21	Accessory ID	This represents the Type of OBD Packet. Refer OBD Packet Type Table for the reference.	Unsigned Integer
23	Refer OBD Data Section	OBD Data	Pipe Separated OBD Data	Pipe Separated Strings
24	*	Separator b/w data & checksum	Indicated End of the frame	Character
25	2F	Checksum	XOR of all data excluding data separator (*), checksum itself and ending characters. Value displayed in HEX	Integer

4.2.2 Tracking Data When Accelerometer data is enabled

This packet will be sent to the server as per the recording interval that is configured in the devices. This packet will contain both tracking and OBD related data.

$$CLIENT_1NS,862843041086125,111,12.976598,77.549461,220120064049,A,12,0,3699,43,5,1.330 000,0,0,0,0,12376,3933,58,-996,-
31,0,21,1|010C:06410C10660000|010D:06410D00000000|0101:064101000E8000|011F:06411F000 00000|0121:06412100000000|0105:06410576550000|0114:06411400000000|010B:06410B000000 00|010F:06410F55000000|0131:06413100000000|0111:06411100000000|010E:06410E00000000| 0149:06414900550000|0123:0641230B7C0000|014A:06414A00000000|0110:06411002520000|*3 5

SL NO	String	Parameter	Description
1	$$	Package Header.	$$ indicates beginning of the packet
2	**CLIENT_1NS**	Client ID	
3	**190803159**	Device serial number/IMEI number	Unique Serial Number or IMEI of the device
4	101	Message/Event code (reason for packets)	Refer Message/Event Code Table:2
5	**12.976515**	Latitude	
6	**77.54966**	Longitude	
7	**190920093311**	Date and time	In yymmddHHMMSS format.
8	A	GNSS fix status	A= valid. V = invalid.
9	**20**	GSM Signal	0 to 31. 99: Not known/not detectable
10	0	Speed	With Decimal digit. In KM/H.
11	583	Accumulated Distance	Accumulated Distance (Odometer based on GPS).
12	3	Course in degree	This value represents the heading direction in degree value. (0-360degrees)
13	6	Satellites being used	Shows the total number of satellites being used for Location points (0-24)
14	1.09	HDOP	Horizontal Dilution of Precision
15	0	Reserved for expansion	Reserved
16	0	Reserved for expansion	Reserved
17	12181	Voltage equivalent of Analog I/P reading	Analog input value measured in mill volts
18	2050	Event Flag	Displayed in decimal, this field can be converted to binary to get the positional data for the 32 information fields as explained in the table below. (Please refer table:1)
19	12181	External Battery Voltage	This value indicates the external battery voltage in millivolt. Here it is 12.267V
20	3960	Internal Battery Voltage	This value represents the internal battery voltage in millivolt. Here it is 3.96V
21	3933	ACC X-axis	signed integer
22	53	ACC Y-axis	signed integer
23	-98	ACC Z-axis	signed integer
24	10023	Trip Time	Time Counter in Seconds
25	21	Accessory ID	This represents the Type of OBD Packet. Refer OBD Packet Type Table for the reference.
26	Refer OBD Data Section	OBD Data	Pipe Separated OBD Data

27	*	Separator b/w data & checksum	Indicated End of the frame
28	2F	Checksum	XOR of all data excluding data separator (*), checksum itself and ending characters. Value displayed in HEX

4.2.3   OBD Data

The OBD data in the tracking packet will be pipe separated in which all the 16 PIDs (Based on configuration) data will be sent to the server. Below is the data of OBD section from the tracking packet. The OBD data will be having the accessory ID 21.

The Accessory ID 21 might contain any data with below type of data.

1. PID Data (1)
2. VIN Data (51)
3. DTC Data (52)

4.2.3.1   PID Data when IGN ON

This section contains the data format of PID data. PID data will be sent in interval which is configured by the user.

The format of PID data will be as follows.

|PIDINFO1:DATA1|PIDINFO2:DATA2|………………………………….|PIDINFO16:PIDDATA16|*

Where
PIDINFO:MODE(2Characters)PID(2Characters)
DATA:Length(2Characters)ResposneType(2Characters)PID(2Characters)

e.g., of only PID Data
1|010C:06410C11780000|010D:06410D03000000|0101:0641010107E100|011F:06411F01A10000|
0121:06412100050000|0105:0641057B550000|0114:06411400000000|010B:06410B18000000|01
0F:06410F3F000000|0131:06413105540000|0111:06411129000000|010E:06410E7E000000|0149:0
6414931550000|0123:06412300000000|014A:06414A18000000|0110:06411000000000|

e.g. of complete packet
$$CLIENT_1NS,862843041081449,1,12.916952,77.596909,220105035657,A,25,0,2898,124,5,1.6200
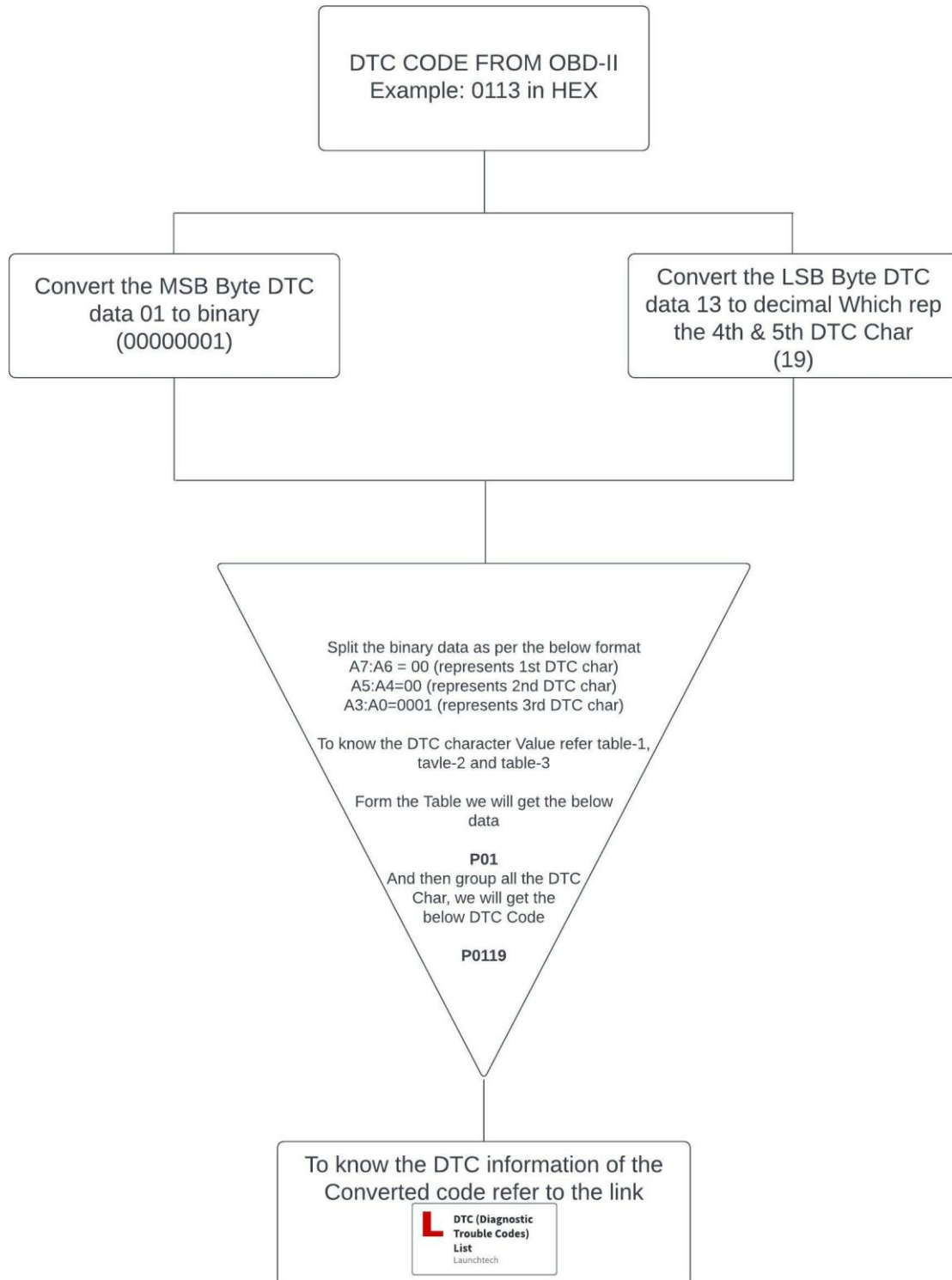00,0,0,0,2048,13572,3933,23,-
75,134,360,21,1|010C:06410C11780000|010D:06410D03000000|0101:0641010107E100|011F:064
11F01A10000|0121:06412100050000|0105:0641057B550000|0114:06411400000000|010B:06410
B18000000|010F:06410F3F000000|0131:06413105540000|0111:06411129000000|010E:06410E7E
000000|0149:06414931550000|0123:06412300000000|014A:06414A18000000|0110:0641100000
0000|*22

4.2.3.2   PID data when PKT_PID_IGNFILT is enabled at IGN-OFF state

During Ignition off state when PKT_PID_IGNFILT is enabled the PID data will not be added in the tracking packet instead the packet will end with the two PIPES below is the example of full complete Tracking packet during IGN OFF when PKT_PID_IGNFILT is enabled.

e.g. of complete packet

$CLIENT_1NS,862843041058421,125,0.000000,0.000000,220119165339,V,19,0,465563,0,0,0.000000,0,0,0,512,12406,3927,0,21,1||*03

4.2.3.3   PID data when PKT_PID_IGNFILT is disabled at IGN-OFF state

During Ignition OFF state When the PKT_PID_IGNFILT is disabled the device sends the data along with the PID data as per the below format.

$$CLIENT_1NS,862843041081449,1,12.916952,77.596909,220105035657,A,25,0,2898,124,5,1.620000,0,0,0,0,13572,3933,123,-75,33,0,21,1|0100:06410CFFFFFFFF|0100:06410DFFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|0100:0641FFFFFFFF|*22

4.2.3.4   VIN Data

This section contains the data format of VIN data. The VIN data will be sent every time when the ignition is turned ON.

The format of VIN Data as follows.
51|VIN

e.g. of only VIN data
51|MALA351ALBM016132|

e.g. of complete packet
$$CLIENT_1NS,862843041081449,20,0.000000,0.000000,220105035030,V,21,51|MAKGM653EEN003463l*05

4.2.3.5   DTC Data

This section contains the data format of DTC data. The DTC data will be sent to the server every minute in below condition.

1. Ignition On
2. DTC is generated in the vehicle

The format of DTC Data as follows.

52|DTC1|DTC2|…………………………………|DTCN|

e.g. of only DTC data

52|0123|1124|………………………….|1132|

e.g. of complete packet

$$CLIENT_1NS,862843041058421,20,12.913817,77.593842,220104105107,A,21,52|C121|C028|*4F

4.2.3.5.1 DTC Data Decoding

Below diagram explains the decoding logic of the DTC that is being received from the device.

A7-A6	First DTC character
00	**P** - Powertrain
01	**C** - Chassis
10	**B** - Body
11	**U** - Network

Table: 1

A5-A4	Second DTC character
00	**0**
01	1
10	**2**
11	**3**

Table: 2

A3-A0	Third DTC character
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8

Table: 3

4.3   Server Command

The commands pushed to the device via GPRS/Server send status response back to the server in packet format. The response format is as explained below.

4.3.1   Command Format

4.3.1.1   Server to Device

The device has the facility to execute the command from the server end. The same command which will be used for Bluetooth and SMS mode should be used with # at the beginning and should be terminated by \r\n to execute the command from the server end.

#set$120000000@aquila123#SRC_IGN:2,5,15,200,11500*\r\n

4.3.1.2   Response From Device to Server

Each command those are executed from the server will be responded with success and failure response. The response format is different for get and set commands.

4.3.1.2.1   Set

Below is the format of the response for set command.

Response Format:

##<Client ID>,<SerialNumber>,<CommandID>,<Success/Failure>,*<Checksum>

*CommandName- Please refer the commands table for the different command names.

Example:

##CLIENT_1NS,862843041058421,SRC_IGN,Success,*48

##CLIENT_1ZF,123456789,SET_ODOM,Success,*0D

4.3.1.2.2 Get

Below is the format of the response for the get command.

Response Format:

$$<Client_ID>,<serial/IMEI_Number>,<event_code>,<Command_ID>,<Value_as_per_set_command>,*<checksum>

Example:

$$CLIENT_1NS,862843041058421,50,CFG_HA,0,10,*76

5 Event Flag Explanation

As shown below, the data for the 32-bit fields can be got by converting the unsigned decimal integer in place of the Event Flag, to binary.

e.g. Event Flag- 8272

32-bit Binary representation of 8272 = (MSB) 00000000000000000010000001010000

Bit	Description	Value
31(MSB)	Reserved	0
30	Reserved	0
29	Reserved	0
28	Reserved	0
27	Harsh braking event	0
26	Harsh accelerometer event	0
25	Accident Status	0
24	DO Status	0
23	Reserved	0
22	Reserved	0
21	Reserved	0
20	Reserved	0
19	Angle Polling bit	0
18	Internal Battery In Low Power Mode	0
17	Reserved	0
16	Reserved	1
15	Sleep Status	0
14	Tilt Status	0
13	Towing Status	0
12	Reserved	0
11	Ignition Current Status	0
10	Reserved	0
9	Reserved	0
8	Reserved	0
7	Power Fail Status	0
6	Immobilizer Violation Status	1
5	Reserved	0
4	Reserved	1
2	Over Speed Stop	0
3	Over Speed Start	0
2	Case open switch status	0
1	Digital Input 1 Status	0

Table 1: Event Code Table

6 Event Code /Message Code Table (Field 4 of Protocol)

Event Code	Description

1	Default Packet due to tracking interval
2	Reserved
3	Power Fail Event
8	Over Speed Started Event.
9	Over Speed ended Event.
10	Ignition ON Event
11	Ignition OFF Event
12	Harsh Braking
14	Towing start event
15	Login Packet
16	Towing end event
17	Vehicle accident event
18	Sleep mode start event
19	Harsh Acceleration
21	Power ON/Hard restart event
22	Illegal Power ON event
23	Harsh cornering
24	Soft restart event
25	NAP mode wakeup event
26	Sleep mode wakeup event

Table 2: Event Code Table

7 Dos and Don'ts

Dos

➢ Read the data till null at the listener end.
➢ Read the data in ASCII format.

Don'ts

➢ Don't disconnect the connection, forcefully/teardown the socket connection.