

# **RIFA-B Easy Plug Tracker** Support OBDII/J1939

**USER MANUAL** 



Version 4.0.0 Oct 30<sup>th</sup>, 2018

ANTZER TECH CO., LTD. 11F-1, No.475, Guangfu S. Rd., Xinyi Dist., Taipei City 11074, Taiwan (R.O.C.) Tel: +886-2-2723-3131 www.antzer-tech.com



## **Important User Information**

#### Liability

Every care has been taken in the preparation of this document. Please inform ANTZER TECH CO., LTD. any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, ANTZER TECH CO., LTD. reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and therefore should not be considered as a binding description of the range of functions (neither for future product versions). ANTZER TECH CO., LTD. assumes no responsibility for any errors that may appear in this document. There are many applications of the described product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards. ANTZER TECH CO., LTD. will under no circumstances assume liability or responsibility for any problems that may arise as a result from improper use or use that is not in accordance with the documented features of this product.

The examples and illustrations in this document are included solely for illustrative purposes.

#### **Intellectual Property Rights**

ANTZER TECH CO., LTD. has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in Taiwan and other countries.

Copyright © 2017 ANTZER TECH CO., LTD. All rights reserved. Antzer Tech RIFA Series User Manual

# **Table of Contents**

1.	Product Description	. 3
1.1.	Introduction	. 3
1.2.	Document History	. 3
1.3.	Hardware Specification	. 4
2.	Hardware Feature	. 5
2.1.	OBD Connector	. 5
2.2.	Micro-USB Connector	. 6
2.3.	LED Indication	. 7
2.4.	Buzzer Operation	. 8
2.5.	SIM Card Slot	. 8
2.6.	Power Supply	. 8
2.7.	Backup Battery	. 9
3.	Configuration	11
3.1.	USB Driver Installation	11
3.2.	Configuring RIFA-B	11
3.3.	BLE Function for Android App Application	15
3.3.1	1 System Requirement	15
3.3.2	2 APK File	15
3.3.3	3 Installation	15
3.3.4	4 Connect Procedures	16
3.3.5	5 Bluetooth Disconnection Procedures	19
3.3.6	6 Known Issues and troubleshooting (depends on Android Version)	20
4.	Firmware Update	21
5.Dii	mension Drawing	21
6.Re	liability Specifications	22
6.1	Environmental	22
6.2	Certification and Compliance	22
7.	Ordering Information	23
8.	Appendix A - The default WWAN data transmitting format	23
9.	Appendix B – AT command list of RIFA	25
10.	Appendix C – RIFA BLE Transmit Protocol Format	35

# **o**antzer

## 1. Product Description

### 1.1. Introduction

Antzer-Tech Automotive-Grade RIFA-B OBDII/J1939 vehicle tracker provides a compact, economic, easy plug and track dongle solution for fleet management. RIFA's self-designed firmware not only supports OBDII & J1939 protocols, but integrates superior cellular modem, highly sensitive GPS, gyroscope and accelerometer in one end-to-end solution. With built-in antennas for both GPS and Cellular modules, RIFA engineers a truly robust connectivity. RIFA-B is combining with the most comprehensive and economical vehicle diagnostics technology, which provides real-time engine monitoring and GPS location. The engine diagnostic data is collected through the vehicle's OBD-II communication port and is transmitted via cellular network to the back-end center. In this way, potential engine problems can be identified earlier before the vehicle breaks down at an inopportune time. Furthermore, you may configure other advanced driving behavior events such as harsh braking, sudden acceleration, speeding, cornering, and much more in order to reduce the risks of vehicle damage and drive down the costs of fuel. This user manual is intended to guide you through the installation and configuration process.

Version	Date	Author	Description
1.0	14-Mar-18	Leopold Chen	First version of this document
2.0	11-July-18	Leopold Chen	Revision of this document
3.0	30-July-18	Leopold Chen	Add BLE application
4.0	30-Oct-18	Leopold Chen	Update Configurator Software

### 1.2. Document History

# 1.3. Hardware Specification

General	Vehicle Interface	Built-in OBD-II(J1962) Compliant Connector		
	Messages	12,000 buffered messages		
	Geofencing	Geofences Zones (Polygon, Rectangular or Circle Setting)		
	Configuration	Over-The-Air Firmware and Remote Maintenance API		
Vehicle	Connector Type	SAE J1962, Male		
Network	OBD-II(J1962)	CANbus, Power and Ground		
	Connector			
	Vehicle Protocols	ISO15765-4 On-Board Diagnostic and J1939		
Cellular	Frequency Band	GSM/GPRS: 850,900,1800,1900 Mhz		
Network		HSPA/UMTS : 800,850,900,1700,1900,2100 Mhz		
		LTE Cat 1 : Band 4, 13 or Band 3,7,20		
		LTE Cat M1/NB 1 : Band 2,3,4,5,8,12,13,20,28		
		LoRa <sup>+</sup> : US915,EU868,AS923 channel plan		
	Data Protocol	TCP, UDP, HTTPs , MQTT , LoRaWAN		
Wireless	Bluetooth	2.4GHz Low Energy Bluetooth Class 2		
Network				
GPS	Chipset	Ublox Neo M8 Engine, 72Channels support GPS, Galileo, GLONASS,		
		BeiDou		
	Deed Deelessing	Outline like Compared UDD. Terable provide CDC Cineral Land		
	Dead Reckoning			
Sensor 3-Axis G-sensor with Auto-Calibrating Fur		3-Axis G-sensor with Auto-Calibrating Function		
LED Indicator		x2 LEDS (2 Colors each for GPS, Cellular Network, and System Status)		
SIM Card	Form Factor	Mini SIM 2FF (25 x 15mm)		
Configuration		MicroUSB Port for RS-232 Configuration Tool		
Port				
Bower	Power Input	0~221/ DC Dower Input from OPDII Connector		
Power	Power input	9 S2V DC Power input noin OBDII connector		
	Range			
	Power Mode	Operating Mode, Sleep Mode, and Battery Mode		
	Min. Power	<3mA @ 12V (Sleep mode), Support CAN Wakeup Function		
	Consumption			
Buzzer		Built-in Buzzer for System Status		
Environment	Operating	-30 ~ 70 °C (Without Battery); -20 ~ 70 °C (With Battery)		
	Temperature			
	Storage	-40 ~ 85 °C		
	Temperature			
	Certificate and	CE, FCC, RoHS and MIL-STD-810G 514.6		
	Vibration			
	Humidity	10% to 90% R.H. (Non-Condensing) Compliant		
Dimensions		86 x 56 x 28 mm		
Battery	Lithium Ion	Optionally support Built-in 3.7V 130mAh Battery		
	Polymer Battery			
Accessories	Maintenance Kit	RS232-to-MicroUSB Cable Length: 1M		

1.External antenna is needed



## 2. Hardware Feature

### 2.1. OBD Connector

Antzer-Tech RIFA-B can support ISO 15765-4, SAE J2284 and SAE J1939 on HD OBD by pin 6/14 with CAN High/Low signal. Also, RIFA-B support wide range power input from 9V to 32V by pin 16/5.

	123	4 5	6 7 8
	9 10 11	12 13	14 15 16
PIN	DESCRIPTION	PIN	DESCRIPTION
1	NC.	9	NC.
2	NC.	10	NC.
3	NC.	11	NC.
4	NC.	12	NC.
5	Ground	13	NC.
6	CAN High	14	CAN Low
-	NC	15	NC.
1	110.	A WARD TO A	

OBD-II Connector and Pinout



### 2.2. Micro-USB Connector

There is an RS-232 port on the RIFA-B through the Micro-USB port for configuration usage. This topic shows you how to connect PC to RIFA-B by the Micro-USB port. Please follow below steps:

1. Prepare one USB to RS-232 cable



Note: You can buy this cable from FTDI website: http://www.ftdichip.com/Products/Cables/USBRS232.htm

2. Prepare one RIFA-B Micro-USB to RS-232 cable (Antzer PN: T170000001)



3.

3.1 Connect the RS-232 side of USB to RS-232 cable to the RS-232 side of RIFA Micro-USB to RS-232 cable



- 3.2 Connect the Micro-USB side of RIFA-B Micro-USB to RS-232 cable to the Micro-USB port of the RIFA-B device
- 3.3 Connect the USB side of USB to RS-232 cable to the USB port of PC



4. Make sure the COM port location of the USB to RS-232 cable recognized by PC (Win OS, Device Manage)

Run the RIFA-B Configuration Tool to execute the settings on RIFA-B (see chapter 3).

2.3. LED Indication

There is a LED indicator of RIFA to show the status of WWAN(cellular network), GPS and System power.



Link(LED1) Green Light: to show RIFA's WWAN module works fine.

Link(LED1) Yellow Light: to show RIFA's GPS module works fine.

System(LED2) Red Light: to show RIFA's power-input works fine.



### 2.4. Buzzer Operation

There is a buzzer equipped on RIFA to indicate (1) System reboot (2) Waring on unsafe driver behavior (3) Waring on driving to restricted area (Geofence) (4) GPS signal detection

### 2.5. SIM Card Slot

User can insert the SIM card (MiniSIM,2FF) to RIFA's SIM card slot showing by below photo.



### 2.6. Power Supply

If you want to test RIFA's function in your office, you can power up RIFA-B by the Micro-USB port with 5V input following below instruction.



### 2.7. Backup Battery

The RIFA-B is optionally equipped a backup battery which allows RIFA-B keep operating for about 30 min. and sending the alarm to central management center at a power loss. However, you may not want to enable the battery after testing. You can switch off the power following below procedure.

1. If you want to do the on-road test, please follow below steps after finished your testing. It can help RIFA to keep the battery life cycle.



Below photo shows the jumper's location on RIFA



2. For real application, you just need to check below steps and then use RIFA on your vehicle.





wake up powered by battery.



## 3. Configuration

Antzer Tech supply a simplicity and clarity configuration tool which is so user-friendly. RIFA collects the common setting by default that customer only need to define the impactive setting.

### 3.1. USB Driver Installation

Following section 2.2. Checking the driver of RS232-to-USB cable is already installed on your laptop or notebook PC which you want to use it to configure the RIFA.

### 3.2. Configuring RIFA-B

Checking the hardware set-up is finished by section2.2. And then executing the "RIFA configure Tool" to configure RIFA.

First, you need to select the COM port correctly with your host device. And then click "Connect" button to start your RIFA's configuration.

💀 SerialPortSetting	_		×
COM29 COM29 Connect	EXIT	~	.:

💀 RIFA Configure Tool v1.6.5			_						
Port Setting									
Configure Trobuleshooting									
Message OK+00032301a3		FW Version Clear Message	Set ID :	τρίγ					
GPS Control	Sensor Control	Set OBDII/J1939 WW	VAN Transmit C	ontrol					
CAN TO ADR : Get Exist Geofence Data	Sensor :  On Off	Per	riod : 5000	ms <sup>3G</sup>					
On  Off Configure Geofence Data	Sensitivity High	Configure PID Tra		) On () Off					
Geofence :	Sensitivity. Ingh		13111331011. C						
On Off Apply	Apply		Apply						
WWAN Configuration									
IP Addr. :	Remote Port :	Local Port T	CP/UDP : TCP	· ~					
APN : PIN :	ID :	PWD :							
SMS phone NO. Authentication : 0-NONE V WWAN : O On () Off Apply									
Enter sleep mode after idling									
60 sec		Apply							
	Save all setting	Get all Setting Load D	efault	EXIT					
FW Version : 00.03.23.01.a3   🔮 Connected   CO	OM27								

#### Message

You can check the system status in message box.

FW Version: Click this button to check the firmware version in message box.

You can clean all the messages by clicking "Clear Message".

#### Device ID

You can set up the ID of the RIFAs from 00000~99999, it can help you to track the device by ID and show IDs in your backend server.

#### **GPS Control**

(1) CAN To ADR: Set "On" to execute CAN-to-ADR function (Optional). If your RIFA equips ADR function, then you can activate the CAN-to-ADR function. It can improve the performance of GPS locating under poor signal circumstance.



- (2) Geofence: Set "On" to execute Geofence function
- (3) Get Exist Geofence Data: Check now the geofence region (coordinates) which is setting.
- (4) Configure Geofence Data: Set up a new geofence region (coordinates).
- (5) Apply: Click this button to save all the "GPS control" settings.

#### **Sensor Control**

- (1) Sensor: Set "On" to execute G-sensor (or Gyroscope) function
- (2) Sensitivity: Select the sensitivity to trigger RIFA's buzzer (or send event record to backend center) by unsafe driving behavior such ad harsh braking and over-acceleration.
- (3) Apply: Click this button to save all the "Sensor control" settings.

#### Set OBDII/J1939

(1)Set OBDII/J1939:Select OBDII or J1939 for your vehicle protocol

(2)Set PID: Click this button and it will help you to choose what PID data you want to get form vehicles (OBDII or J1939).

💀 PIDSetting		_	×
Set OBDII/J1939 : OBDII ~ Set PID : 22-VIN ~ < < < <	OD-VSS OC-RPM		
Ok			



Click "OK" to exit from "Configure PID" and save the configurations.

#### WWAN Transmit Control

- (1) Period: Type the period value (milliseconds) you want to send data via cellular network. (System will show the network type such as 3G, 4G or NBIoT)
- (2) Transmission: Set "On" to request WWAN module to send data by defined time interval.
- (3) Apply: Click this button to save all the "WWAN Transmit control" settings.

#### WWAN configuration

- (1) IP Addr/Remote Port/Local Port : Type the IP address, remote port and local port(if needed) of your server uploading by RIFA
- (2) TCP/UDP: Select the data transmission protocol by RIFA's data uploading
- (3) APN/PIN/ID/PWD: Type the APN,PIN,ID and password provided by your WWAN SIM card carrier
- (4) SMS phone No.: Type the phone number used which you want to remotely configure (dial into) RIFA by SMS
- (5) Authentication: Select the authentication type by data transmission
- (6) WWAN: Set "On" to wake up WWAN module
- (7) Apply: Click this button to save all the "WWAN configuration" settings.

#### **RTC Control**

- (1) Date/Time: Set RTC timer(0-None) or you can select GPS UTC time (1-GPS) for your system
- (2) Enter sleep mode after idling: Set the time period to enter sleep mode while there is no OBD or J1939 data input to RIFA.
- (3) Apply: Click this button to save all the "RTC Control" settings.



Save all setting : Click this button to save all of your settings.

**Noted**:Please don't forget to click "Save all setting" button, it will keep all your settings after finished configuration procedure.

Get all setting : Click this button to check the previous settings of your RIFA.

Load Default: Click this button to set factory settings to the RIFA.

**EXIT:** Click this button to close configuration tool.

- 3.3. BLE Function for Android App Application
  - 3.3.1 System Requirement

Software version needed is Android 6.0.0 with BLE 4.0 (or above version)

3.3.2 APK File

Please install Antzer BLE software APK : Antzer\_BLE\_vX\_X\_X.apk

#### 3.3.3 Installation

1.Copy above .apk file into Android cell phone.

2.Click the apk and following the installing steps to install the BLE application.

3. It appears an ANTZER-BLE icon on Android cell phone Apps list as Figure 1.





#### Figure 1

#### 3.3.4 Connect Procedures

1. Plug in the power adapter and power on RIFA BLE dongle.

2.Wait for booting ready buzzer beep (1 high and 1 low beep).

3.Launching ANTZER BLE ANTZER-BLE app as Figure 2

Φ	≉ ∯. <b>⊪</b> 191% <b>⊠</b>	19:55
୕ପ	О STOP	3
Unknown device	(RSSI: -88)	
Unknown device	(RSSI: -75)	
Unknown device	(RSSI: -89)	
NINA-B1-B4CA92	(RSSI: -58)	
Unknown device	(RSSI: -98)	
Unknown device	(RSSI: -97)	
Unknown device	(RSSI: -99)	
CONN	ECT	



4. Select RIFA BLE device on the list (check BLE mac address) and click the "CONNECT" button as Figure 3.

<b>© =</b>	2	≸ <b>∜ .⊪I</b> 91%	<b>a</b> 19:56
୕ପ	C	STOP	0
Unknown devic	e	(RSSI: -89)	
Unknown devic	e	(RSSI: -72)	
Unknown devic	e	(RSSI: -91)	
a NINA-B1-B4CA9	92	(RSSI: -58)	
Unknown devic	e	(RSSI: -98)	
Unknown devic	e	(RSSI: -97)	
Unknown devic	e	(RSSI: -99)	
Unknown devic	е	(RSSI: -93)	
b c	ONNECT		



5. Waiting for RSSI value appearance (BLE signal checking) and then click "CHAT" tab button to read CAN bus data as Figure 4.

			7	11.11.9	1/0 24 19.0
ପ	NINA-E	81-B4CA	92	D	ISCONNEC
OVE	RVIEW	ALL C	AN DATA	b	CHAT
3	Tempera	ture (°C)			
Ŵ	RSSI (dB	)			a -62
Ô	Battery L	evel (%)			-
ス	Accelero	meter Rar	nge (G)		12
x			_		
Y	_		-		
z			-		
Red			Green	關	
Light	開	11	Light	Contraction of the local distance of the loc	



6. CAN bus data appears by hex data format and in a row by data scrolling as Figure 5.

# **o**antzer

	* 1	<sup>ም</sup> . <b>ብ</b> 84% 🖬 12:00
	B1-B4CA92	DISCONNECT
	ALL CAN DATA	CHAT
5,0,6,8,0,0	,10,3	,,
	Fri Jul 20 11:59:59 G	BLE \$ MT+08:00 2018
10,2,a2,e,0	),c,	
8,5,0,0,0,0	,0,0,0,13,88,0,	10,3
		BLE ¥
	Fri Jul 20 11:59:59 G	MT+08:00 2018
10,2,a3,b,a 5,0,0,0,10,	a,5,0,a,0,0,0,0, 3	0,a,
		BLE *
	Fri Jul 20 11:59:59 G	MT+08:00 2018
10,2,b1,6,3	37,0,75,44,d0,	5,0,0,0,0,0
,-,-,-,-,-0,0	Fri Jul 20 12:00:00 G	BLE \$ MT+08:00 2018
Write a mess	sage	Send

Figure 5

### For OBDII/J1939 Data Collection, please follow below steps

7. Waiting for RSSI value appearance (BLE signal checking) and then click "ALL CAN DATA" sheet to read CAN(J1939) data as Figure 6.

<b>\$</b>			*	49 . <b>nl</b> 91%	a 🖬 19:56
ିପ	NINA-	B1-B4C	A92	DIS	CONNECT
ovi	ERVIEW	b ALL	CAN DATA		нат
л	Temper	ature (°C	2)		
Q	RSSI (d	B)	<i>.</i> ,		a -62
Ê	Battery	Level (%)	)		-
ス	Acceler	ometer F	Range (G)		
х	-		_		_
Y	-		_		_
z					
Red Ligh	t Ra		Green Light	關	



8. CAN (OBDII/J1939) bus data will appear by readable format as Figure 7.

	* *	<sup>ទ្ធ</sup> .៨ 84% 🖬 12:00	
O NINA-B1-B4CA92 DISCONNECT			
OVERVIEW	ALL CAN DATA	CHAT	
ALTITUDE	0.00		
SPEEDDEGREE			
DATETIME	ũ.		
LAITITUDE	0000.000	000E	
LONGITUDE	0000.000	W00W	
SPEED	90		
RPM	2500		
G-SENSOR	27.0.35,44,10,6		
GYRO	00.00.00		
FLI	24.8		
AAT	-1.5		
EFR	6.8		
PCT	0.0		
VEP	12.8		
ECT	5		
ETP	0.0		
MAF	0.0		
DIST	5000.0		
ABP	10.5		
IAT	10.0		
FRP	0.0		
EOT1	10.5		

Figure 7

### 3.3.5 Bluetooth Disconnection Procedures

1. 1. Click "DISCONNECT" button as Figure 9(a).





2. The App will stop data appearance scrolling, and RIFA BLE dongle will have a short beep buzzer after completing BLE disconnection between smart phone



and RIFA.

3. 3. The "DISCONNECT" button will change to "CONNECT" button around 30 seconds as Figure 10(a).



Figure 10

4. For re-connection of BLE device, click Android device button () to initialize connection(as Figure 2).

3.3.6 Known Issues and troubleshooting (depends on Android Version)

 sometimes in disconnection procedure, the "DISCONNECT" button may not change to "CONNECT" button after 30 seconds waiting.
 Troubleshooting – click Android device button (

2. Sometimes in BLE re-connection, Android app may not read CAN bus data.

<u>**Troubleshooting**</u> – click Android device button ( ) to return to initial connection screen and doing connection procedures again.

3. In disconnection procedure, it could not beep the buzzer of disconnection and can't be re-connected anymore. (It may happen in some specific Android devices with low possibility).

Troubleshooting - Power off RIFA dongle (re-connect from vehicle) for forcingly



disconnect BLE connection and then re-connect it.

### 4. Firmware Update

- 1. Launch RIFAUpdTool\_V2\_1x.exe
- 2. Select the appropriate serial port from the column "Serial Port".
- 3. Press "Open Port" to open the serial port.
- 4. Press "Reboot" to enter to firmware update mode and then it will display update messages. Once "Waiting for the file to be sent..." appears, then you can
- 5. Press "Update" to start firmware update and waiting for firmware update completely.
- 6. It will display "Start Program execution..." when firmware update and reset completely, and then press "Close Port" to finish RIFA firmware update procedures.



## **5.Dimension Drawing**



## **6.**Reliability Specifications

### 6.1 Environmental

Environmental specifications of RIFA-B follow MIL-STD-810G, as indicated in the following table.

Environment	Specifications
Operating	-40°C to 85°C (without Battery)
Temperature	-20°C to 60°C (with Battery)
Vibration	Operating: Random, 7.69(Grms), 20~2000(Hz)
	Compliant with MIL-STD-810G

### 6.2 Certification and Compliance

Antzer-Tech RIFA-B complies with the following standards:

- CE
- FCC Class B
- RoHS
- MIL-STD-810G Vibration Compliant



# 7. Ordering Information

**RIFA-B Product Part Number** 

Part Number	Description
RIFA-B73Q-G01000	OBDII CONN, OBDII, J1939, 3G, GPS
RIFA-B7LQ-E01000	OBDII CONN, OBDII, J1939, 4G C1(EU), GPS
RIFA-B7LU-E01000	OBDII CONN, OBDII, J1939, 4G C1(EU), GPS UDR
RIFA-B7LQ-A01000	OBDII CONN, OBDII, J1939, 4G C1(US), GPS
RIFA-B7LU-A01000	OBDII CONN, OBDII, J1939, 4G C1(US), GPS UDR
RIFA-B7MQ-G01000	OBDII CONN, OBDII, J1939, 4G CM1, NBIoT GPS
RIFA-B7MU-G01000	OBDII CONN, OBDII, J1939, 4G CM1, NBIoT GPS UDR

Part Number	Description
T170000001	RS232-to-MicroUSB Cable (1M)

# 8. Appendix A - The default WWAN data transmitting format

<b>RIFA 3G/LTE Transmission Protocol Format</b>		
Default: UART_1, 115200 bps		
SendMessage		
Syntax:		
\$FRCMD,IMEI,_Se	ndMessage,,latitude,hemi,longitude,hemi,alt,speed,heading,date,time,valid,Analo	
g1=value,Analog2=	value*XX	
Field	Descriptions	
Device ID	Device ID, e.g. 24680 (from 00000 to 99999)	
latitude	Latitude (NMEA format), e.g. DDMM.mmmm	
hemi	Hemisphere N or S	
longitude	Longitude DDMM.mmmm (NMEA format)	
hemi	Hemisphere E or W	
alt	Altitude in meters above sea level, e.g. AA.a	
speed	Speed over ground in knots, e.g. SSS.ss	
heading	Heading over ground in degrees, e.g. HHH.h	
date	Date, DDMMYY	
time	Time (UTC), hhmmss.dd	

valid	1 if a vlid fix. 0 if not a valid fix.	
Analog1=value	Status signals. Analog1=VSS (Vehicle Speed, 2 bytes)	
Analog2=value	Status signals. Analog2=RPM (Vehicle RPM, 2 bytes)	
Analog3=value	Status signals. Analog3=FLI (Fuel Level, 2 bytes, FLI.f_FLI)	
A palog 4-value	Status signals. Analog4=AAT (Ambient air temperature, 3 bytes,	
Analog4_value	AAT_H_L.f_AAT)	
Analog5=value	Status signals. Analog5=EFR(Engine fuel rate, 3 bytes, EFR_H_L.f_EFR)	
Analog6=value	Status signals. Analog6=PCT(Calculated engine load, 2 bytes, PCT.f_PCT)	
Apolog7-voluo	Status signals. Analog7=VEP(Vehicle Electrical Power, 3 bytes,	
Analog / _ value	VEP_H_L.f_VEP)	
Analog8=value	Status signals. Analog8=ECT(Engine coolant temperature, 1 byte, ECT)	
Analog9=value	Status signals. Analog9=ETP(Engine Throttle Position, 2 bytes, ETP.f_ETP)	
Analog10=value	Status signals. Analog10=MAF(Mass Air Flow, 3 bytes, MAF_H_L.f_MAF)	
Apolog 11-voluo	Status signals. Analog11=DIST(Vehicle Distance, 5 bytes,	
Analog11=value	DIST_H1_H2_L1_L2.f_DIST)	
Apolog12-voluo	Status signals. Analog12=ABP(Absolute Barometric Pressure, 2 bytes,	
Allalog12–value	ABP.f_ABP)	
Analog13=value	Status signals. Analog13=IAT(Intake air temperature, 3 bytes, IAT_H_L.f_IAT)	
Apolog14-voluo	Status signals. Analog14=FRP(Engine Fuel Valve 1 Inlet Absolute Pressure, 3	
Allalog14–value	bytes, FRP_H_L.f_FRP)	
Analog 15-yalue	Status signals. Analog15=EOT1(Engine oil temperature, 3 bytes,	
Analog15-value	EOT1_H_L.f_EOT1)	
Analog16=value	Status signals. Analog16= ()	
Analog17=value	Status signals. Analog17=()	
Analog18=value	Status signals. Analog18= ()	
Analog19=value	Status signals. Analog19= ()	
Analog20=value	Status signals. Analog20=()	
Analog21=value	Status signals. Analog21= ()	
Analog22=value	Status signals. Analog22= ()	
Analog23=value	Status signals. Analog23= ()	
Analog24=value	Status signals. Analog24=()	
Analog25=value	Status signals. Analog25= ()	
Analog26=value	Status signals. Analog26= ()	
Analog27=value	Status signals. Analog27=()	
Analog28=value	Status signals. Analog28= ()	



Analog29=value	Status signals. Analog29= ()
Analog30=value	Status signals. Analog30=()
Analog31=value	Status signals. Analog31= ()
Analog32=value	Status signals. Analog32= ()
Analog33=value	Status signals. Analog33=()
Analog34=value	Status signals. Analog34=()
XX	Checksum, 1 byte
Total	~512 bytes

# 9. Appendix B – AT command list of RIFA

### **RIFA** Configuration List

Field	Description	CMD	Notes
			DUT should return
			firmware information
Einner Vansion	Get Firmware	> A1 + VEK?	correctly to Host PC.
	Version	A(0x41) T(0x54) 0x2B V(0x50) E(0x45) B(0x52) 2(0x2E)	e.g.
		E(0x45) R(0x52) ?(0x3F)	OK+010900A1 =
			v01.09.00.A1
Decet		<b>Δ.Τ. DCT 1</b>	DUT should auto-reboot
Reset DUI	Reset DUI	> A1+K51,1	itself.
	Eismaurona un data		After finished update
Update	firmware update		procedure, the system
	function	> A1+UPD,2	should boot successfully.
Reset to default	Initialize settings		1 System reset
setting	about DUT.	> A1+N01,3	1. System reset.

			1. If the range of setting
			parameter is 10 to 300
			Sec., the return value is
	Set the range of		OK.
Sleeping Time	Sleeping Time Out	> AT+CFGS,x	2. If the setting parameter
Out	Configure is 10 to	(x=010s ~ 300s, Unit: mS.)	is other value, the return
	300 seconds.		value will show FAIL.
			3. When the set time is
			reached, the DUT should
			enter sleep mode.
			1. If the range of PIN
			Code is correct, the
	Setup the specific	> AT+PIN, ' <b>xxxxxxx</b> '	return value is OK.
PIN Code	PIN code for the SIM	(Set the PIN Code range is from	2. If the range of PIN
	card	4 to 8 characters.)	Code is abnormal, the
			return value will show
			FAIL.
			1. If the Protocol type is
			correct, the return value
			is OK.
			2. If the Protocol type is
2G Pagistor	Setup WWAN		abnormal, the return
SG Register	communication	> AT+GPRS, <protocol type=""></protocol>	value will show FAIL.
- Hotocol type	protocol type		3. The Mapping Table
			about 0x08004032 (1
			Byte) value is correct.
			e.g.: 0x31 = IPV4(defalt
			value), 0x30 ~ 0x39
			1. If the APN is correct,
3G Register		ΔT+ΔPN < ΔPN <	the return value is OK.
- APN	Setup WWAN APN	$(\Delta DN = 16 \text{ bytes})$	2. If the Protocol type is
		$(\mathbf{M} \mathbf{N} = 10 \text{ bytes})$	abnormal, the return
			value will show FAIL.
			1. If the UNAME is
3G Register	Setup WWAN Login	SAT⊥IINAME ZUGer Nomes	correct, the return value
- User name	user name	> 111 + UTVINIU, NUSCI INdilit>	is OK.
			2. If the Protocol type is

			abnormal, the return value will show FAIL.
3G Register - Password	Setup WWAN Login password	> AT+PWD, <password></password>	<ol> <li>If the password is correct, the return value is OK.</li> <li>If the Protocol type is abnormal, the return value will show FAIL.</li> </ol>
3G Register - Authentication	Setup WWAN Authentication	> AT+AUTH, <authentication></authentication>	<ol> <li>If the Authentication is correct, the return value is OK.</li> <li>If the Protocol type is abnormal, the return value will show FAIL.</li> </ol>
3G Register - Service Type	Setup WWAN Service Type	> AT+STYPE, <service type=""></service>	<ol> <li>If the Service Type is correct, the return value is OK.</li> <li>If the Protocol type is abnormal, the return value will show FAIL.</li> </ol>
3G Register - Host IP Address	Setup WWAN Host IP Address	> AT+HIP, <host address="" ip=""></host>	<ol> <li>If the Host IP Address is correct, the return value is OK.</li> <li>If the Protocol type is abnormal, the return value will show FAIL.</li> </ol>
3G Register - Remote Port	Setup WWAN Remote Port	> AT+RPORT, <remote port=""></remote>	<ol> <li>If the Remote Port is correct, the return value is OK.</li> <li>If the Protocol type is abnormal, the return value will show FAIL.</li> </ol>

			1. If the Remote Port is
3G Register	Setup WWAN Local	> AT+LPORT, <local port=""></local>	correct, the return value
			is OK.
- Local Port	Port		2. If the Protocol type is
			abnormal, the return
			value will show FAIL.
			1. DUT should return
			Gyro Threshold Data to
			Host PC.
			Data format:
			O K + 0xXX_H 0xXX_L
Gyro Threshold	Gat Gura Thrashold		0xYY_H 0xYY_L
Data	Data	> AT+GTHS?	0xZZ_H 0xZZ_L (Total
Data	Data		9 Bytes)
			2. If communication fail
			or appear any errors, the
			return value will show
			FAIL.
			1. If the Gyro Threshold
			Data is correct, the return
Set Gyro Threshol	Set Gyro Threshold	> AT+STHS, <gyro threshold<br="">Data&gt;</gyro>	value is OK.
	Data		2. If communication fail
	2		or appear any errors, the
			return value will show
			FAIL.
			1. DUT should return
			Gyro Duration Data to
			Host PC.
Gyro Duration			Data format:
	Get Gyro Duration		O K + 0xXX (Total 4
Data	Data	> AT+GDUR?	Bytes)
			2. If communication fail
			or appear any errors, the
		return value will show	
			FAIL.

			1. If the Gyro Duration
		> AT+SDUR, <gyro duration<="" td=""><td>Data is correct, the return</td></gyro>	Data is correct, the return
	Set Gyro Duration	Data>	value is OK.
	Data	e.g.:	2. If communication fail
	Data	A T + SDUR , 0xXX (Total 9	or appear any errors, the
		Bytes)	return value will show
			FAIL.
			1. DUT should return
			Accelerometer Data to
			Host PC.
			Data format:
			O K + 0xXX (Total 4
Accelerometer	Get Accelerometer	> AT+EATHS1?	Bytes)
Data	Threshold 1		
			2. If communication fail
			or appear any errors, the
			return value will show
			FAIL.
			1. DUT should return
		> AT+EATHS2?	Accelerometer Data to
			Host PC.
			Data format:
			O K + 0xXX (Total 4
	Get Accelerometer		Bytes)
	Threshold 2		
			2. If communication fail
			or appear any errors, the
			return value will show
			FAIL.
			1. DUT should return
Accelerometer Duration Data			Accelerometer Duration
			Data to Host PC.
	Get Accelerometer		Data format:
	Duration 1	> AT+EADUR1?	O K + 0xXX (Total 4
			Bytes)
			2. If communication fail



		or appear any errors, the return value will show FAIL.
Get Accelerometer Duration 2	> AT+EADUR2?	<ol> <li>DUT should return Accelerometer Duration Data to Host PC.</li> <li>Data format: O K + 0xXX (Total 4 Bytes)</li> </ol>
		2. If communication fail or appear any errors, the return value will show FAIL.
Set Accelerometer Threshold 1	> AT+SATHS1, <accelerometer Threshold_1 value&gt;</accelerometer 	<ol> <li>If the Accelerometer Threshold 1 Data is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> </ol>
Set Accelerometer Threshold 2	> AT+SATHS2, <accelerometer Threshold_2 value&gt;</accelerometer 	<ol> <li>If the Accelerometer Threshold 2 Data is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> </ol>

			1. If the Accelerometer
			Duration 1 Data is
			correct, the return value
	Set Accelerometer	> AT+SADUR1, <accelerometer< td=""><td>is OK.</td></accelerometer<>	is OK.
	Duration 1	Duration_1 value>	2. If communication fail
			or appear any errors, the
			return value will show
			FAIL.
			1. If the Accelerometer
			Duration 2 Data is
			correct, the return value
	Set Accelerometer	> AT+SADUR2, <accelerometer< td=""><td>is OK.</td></accelerometer<>	is OK.
	Duration 2	Duration_2 value>	2. If communication fail
			or appear any errors, the
			return value will show
			FAIL.
			FAIL. 1. If the AVL Sensor
			FAIL. 1. If the AVL Sensor Data Transmission
			FAIL. 1. If the AVL Sensor Data Transmission function is correct, the
			FAIL. 1. If the AVL Sensor Data Transmission function is correct, the return value is OK.
			FAIL. 1. If the AVL Sensor Data Transmission function is correct, the return value is OK. 2. If communication fail
	Dicable / Enable		FAIL. 1. If the AVL Sensor Data Transmission function is correct, the return value is OK. 2. If communication fail or appear any errors, the
AVL Sensor Data	Disable / Enable	> AT+SENS,x	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data	> AT+SENS,x (x= 0/1, Disable/Enable)	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data Transmission	> AT+SENS,x (x= 0/1, Disable/Enable)	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> <li>3. Mobile phone device</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data Transmission	> AT+SENS,x (x= 0/1, Disable/Enable)	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> <li>3. Mobile phone device</li> <li>should receive Sensor</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data Transmission	> AT+SENS,x (x= 0/1, Disable/Enable)	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> <li>3. Mobile phone device</li> <li>should receive Sensor</li> <li>Data via BT from DUT.</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data Transmission	> AT+SENS,x (x= 0/1, Disable/Enable)	<ul> <li>FAIL.</li> <li>1. If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>2. If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> <li>3. Mobile phone device</li> <li>should receive Sensor</li> <li>Data via BT from DUT.</li> <li>4. Cloud server should</li> </ul>
AVL Sensor Data Transmission	Disable / Enable AVL Sensor Data Transmission	> AT+SENS,x (x= 0/1, Disable/Enable)	<ol> <li>FAIL.</li> <li>If the AVL Sensor</li> <li>Data Transmission</li> <li>function is correct, the</li> <li>return value is OK.</li> <li>If communication fail</li> <li>or appear any errors, the</li> <li>return value will show</li> <li>FAIL.</li> <li>Mobile phone device</li> <li>should receive Sensor</li> <li>Data via BT from DUT.</li> <li>Cloud server should</li> <li>receive Sensor Data via</li> </ol>

l	I	1	
Transmitted Time Interval Configure	Setup Transmitted Time Interval Configure	> AT+CFGI, <transmitted interval time&gt; (Transmitted interval time=1ms ~ 65535ms)</transmitted 	<ol> <li>If the AVL Transmitted interval tim is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> <li>Mobile phone device should receive Sensor Data by Transmitted interval time via BT fron DUT.</li> <li>Cloud server should receive Sensor Data by Transmitted interval tim via WWAN from DUT.</li> </ol>
RTC Configure	RTC Configure - Date	> AT+RTCD, <date> (date format (HEX) = 00YYMMDD YY: Year, MM: Month, DD: Day)</date>	<ol> <li>If the AVL RTC(date is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> </ol>
	RTC Configure - Time	> AT+RTCT, <date> (date format (HEX) = 00hhmmss hh: Hour, mm: Minute, ss: Second)</date>	<ol> <li>If the AVL RTC(time is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> </ol>
OBDII PID	Configure OBDII PID	> AT+PID, <pid code=""> (PID Code (HEX) = xxxxxxxxx)</pid>	<ol> <li>If the OBDII PID is correct, the return value is OK.</li> <li>If communication fail or appear any errors, the</li> </ol>



			return value will show				
			FAIL.				
			1. DUT should return				
			Gyro Angular Rate Data				
			to Host PC.				
			Data format:				
			$O K + 0xXX_H 0xXX_L$				
	Get Gyro Angular		0xYY_H0xYY_L				
Gyro Angular		> AT+GOUT?	0xZZ_H 0xZZ_L (Tota				
Rate	Rate Data		9 Bytes)				
			2. If communication fail				
			or appear any errors, the				
			return value will show				
			FAIL.				
			1. DUT should return				
	Get Accelerometer		Accelerometer Data to				
			Host PC.				
			Data format:				
			O K + 0xXX_H 0xXX_L				
A			0xYY_H 0xYY_L				
Accelerometer		> AT+AOUT?	0xZZ_H 0xZZ_L (Total				
Data	Data		9 Bytes)				
			2. If communication fail				
			or appear any errors, the				
			return value will show				
			FAIL.				
			1. If the writed gofence				
Goofanca Data	Stora Goofanca Data		command is correct, the				
(SPI ROM)	Store Georence Data	> AT+GEOF, <geofence data=""></geofence>	return value is OK.				
			2. If communication fail				
			or appear any errors, the				



			return value will show FAIL.
	Get Geofence Data (SPI ROM)	> AT+GEOFxxx? (xxx = 000 ~ 255)	<ol> <li>DUT should return Geofence Data to Host PC.</li> <li>If communication fail or appear any errors, the return value will show FAIL.</li> </ol>
OBD/J1939 request time interval Configure	Setup OBD/J1939 request time interval Configure	> AT+CFGI, <obd j1939<br="">requested interval time&gt; (OBD/J1939 requested interval time = ??? ~ ???, Unit: ms: OBDII default vale: 50 ms J1939 default vale: 2 s)</obd>	<ol> <li>If the range of setting parameter is ??? to ???, the return value is OK.</li> <li>If the setting parameter is other value, the return value will show FAIL.</li> </ol>

Froming	Total = 20	butoe																	
DI F		CMD	DLC							D	ata							DLE	FTY
0x10	0x02	0xA1	14							141	ovtes							0x10	0x03
0	1	2	3	4	5	6	7	8	0	10	11	12	13	14	15	16	17	18	19
0	1	2			5	0	1	0	,	10		12	15	14	15	10	17	10	15
Response	= 0x91: R	eceive GP	S Latitude	+ Hemisp	here Data	(ASCII)													
DLE	STX	CMD	DLC					Lati	itude					Hemi		0x00		DLE	ETX
0x10	0x02	0x91	11					10 t	oytes					N or S		3 bytes		0x10	0x03
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Response	e = 0x92: R	leceive GP	S Longitu	de + Hemi	sphere Da	ta (ASCII)													
DLE	STX	CMD	DLC						Longitude						Hemi	0x	(00	DLE	ETX
0x10	0x02	0x92	12						11 bytes						E or W	2 t	oyte	0x10	0x03
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Deenenen	- 0-02. B	CT	C Altituda	Data (AS	CID														
DIE	STV	CMD	DLC	Data (AS	(11)		Altitude							0x00				DIE	ETV
DLE 0x10	0x02	0x02	7				7 bytes							7 bytes				0x10	0x02
0.10	1	2	3	4	5	6	7 0 y te 3	8	0	10	11	12	13	14	15	16	17	18	10
0	1	2	5	4	5	0	/	0	,	10	11	12	1.5	14	15	10	17	10	19
Response	= 0x94: R	eceive GP	S Speed +	Degree D	ata (ASCI	D													
DLE	STX	CMD	DLC		(	Sp	eed					Degree				0x00		DLE	ETX
0x10	0x02	0x94	11			6 b	ytes					5 bytes				3 bytes		0x10	0x03
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Response	= 0x96: R	eceive GF	S Date + 7	Time (ASC	II)														
DLE	STX	CMD	DLC			D	ate						Time	(UTC)				DLE	ETX
0x10	0x02	0x96	14			6 bytes (D	DMMYY)						8 bytes (l	nhmmss.d)				0x10	0x03
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Response	= 0xA1: H	Receive Ve	hicle VSS	+ RPM +	FLI + AA	T + EFR +	PCT Data	(Binary)											
Response DLE	e = 0xA1: H STX	Receive Ve CMD	ehicle VSS DLC	+ RPM + VSS (1	FLI + AA unsign)	T + EFR + RPM (	PCT Data unsign)	(Binary) FLI (t	unsign)		AAT (sign	1)	F	EFR (unsig	1)	PCT (	unsign)	DLE	ETX
Response DLE 0x10	e = 0xA1: H STX 0x02	Receive Ve CMD 0xA1	ehicle VSS DLC 14	+ RPM + VSS (u 2 b	FLI + AA unsign) ytes	T + EFR + RPM ( 2 b	PCT Data unsign) ytes	(Binary) FLI (u 1 byte	insign) 1 byte	2 b	AAT (sign	i) 1 byte	E 2 b	FR (unsign ytes	1) 1 byte	PCT (	unsign) 1 byte	DLE 0x10	ETX 0x03
Response DLE 0x10	e = 0xA1: H STX 0x02	Receive Ve CMD 0xA1	ehicle VSS DLC 14	+ RPM + VSS (t 2 b VSS_H	FLI + AA unsign) ytes VSS_L	T + EFR + RPM ( 2 b RPM_H	PCT Data unsign) ytes RPM_L	(Binary) FLI (u 1 byte FLI	insign) 1 byte f_FLI	2 b AAT_H	AAT (sign ytes AAT_L	ı) 1 byte f_AAT	EFR_H	EFR (unsign ytes EFR_L	1) 1 byte f_EFR	PCT ( 1 byte PCT	unsign) 1 byte f_PCT	DLE 0x10	ETX 0x03
Response DLE 0x10	= 0xA1: H STX 0x02	Receive Ve CMD 0xA1	chicle VSS DLC 14	+ RPM + VSS (t 2 b VSS_H	FLI + AA unsign) ytes VSS_L	T + EFR + RPM ( 2 b RPM_H	PCT Data unsign) ytes RPM_L	(Binary) FLI (t I byte FLI FLI.:	Insign) 1 byte f_FLI f_FLI	2 b AAT_H	AAT (sign ytes AAT_L AAT.f_AA	1) 1 byte f_AAT	EFR_H	EFR (unsign ytes EFR_L EFR.f_EFF	1) 1 byte f_EFR 2	PCT (1 1 byte PCT PCT.	unsign) 1 byte f_PCT f_PCT	DLE 0x10	ETX 0x03
Response DLE 0x10	= 0xA1: H STX 0x02	CMD 0xA1	blicle VSS DLC 14 3	+ RPM + VSS (t 2 b VSS_H 4	FLI + AA unsign) ytes VSS_L 5	T + EFR + RPM ( 2 b RPM_H 6	PCT Data unsign) ytes RPM_L 7	(Binary) FLI (t 1 byte FLI FLI 8	insign) 1 byte f_FLI f_FLI 9	2 b AAT_H 10	AAT (sign ytes AAT_L AAT.f_AA 11	n) <u>1 byte</u> f_AAT xT 12	EFR_H	EFR (unsign ytes EFR_L EFR.f_EFF 14	1) 1 byte f_EFR 2 15	PCT ( 1 byte PCT PCT. 16	unsign) 1 byte f_PCT f_PCT 17	DLE 0x10	ETX 0x03
Response DLE 0x10 0	= 0xA1: H STX 0x02	Receive Ve CMD 0xA1 2	bicle VSS DLC 14 3	+ RPM + VSS (t 2 b VSS_H 4	FLI + AA unsign) ytes VSS_L 5 ETP + M/	T + EFR + RPM ( 2 b RPM_H 6	PCT Data unsign) ytes RPM_L 7 Data (Bin	(Binary) FLI (t 1 byte FLI FLI 8	Insign) 1 byte f_FLI f_FLI 9	2 b AAT_H 10	AAT (sign ytes AAT_L AAT.f_AA 11	n) 1 byte f_AAT NT 12	EFR_H 13	EFR (unsign ytes EFR_L EFR.f_EFF 14	1) 1 byte f_EFR 2 15	PCT ( 1 byte PCT PCT. 16	unsign) 1 byte f_PCT f_PCT 17	DLE 0x10	ETX 0x03
Response DLE 0x10 0 Response DLE	= 0xA1: I STX 0x02 1 = 0xA2: I STX	Receive Ve CMD 0xA1 2 Receive Ve CMD	ehicle VSS DLC 14 3 ehicle VEP DLC	+ RPM + VSS (t 2 b VSS_H 4 + ECT +	FLI + AA unsign) ytes VSS_L 5 ETP + MA	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n)	PCT Data unsign) ytes RPM_L 7 Data (Bina ECT (sign)	(Binary) FLI (t 1 byte FLI FLI 8 ary) ETP (t	Insign) 1 byte f_FLI f_FLI 9 unsign)	2 b AAT_H 10	AAT (sign ytes AAT_L AAT.f_AA 11	1) 1 byte f_AAT XT 12 20)	EFR_H	EFR (unsign ytes EFR_L EFR.f_EFF 14 D	1) 1 byte f_EFR 2 15 IST (unsig	PCT () 1 byte PCT PCT.: 16	unsign) 1 byte f_PCT f_PCT 17	DLE 0x10 18 DLE	ETX 0x03
Response DLE 0x10 0 Response DLE 0x10	= 0xA1: H STX 0x02 1 = 0xA2: H STX 0x02	CMD 0xA1 2 Receive Ve CMD 0xA2	ehicle VSS DLC 14 3 ehicle VEP DLC 14	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V	FLI + AA unsign) ytes VSS_L 5 ETP + MA /EP (unsig ytes	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte	PCT Data unsign) ytes RPM_L 7 Data (Bina ECT (sign) 1 byte	(Binary) FLI (t I byte FLI FLI: 8 ary) ETP (t I byte	insign) 1 byte f_FLI f_FLI 9 unsign) 1 byte	2 b AAT_H 10 M 2 b	AAT (sigr ytes AAT_L AAT.f_AA 11 IAF (unsig ytes	n) <u>1 byte</u> f_AAT AT 12 gn) 1 byte	EFR_H	EFR (unsign ytes EFR_L EFR.f_EFF 14 D 4 b;	1) 1 byte f_EFR 15 IST (unsig ytes	PCT ( 1 byte PCT PCT. 16 n)	unsign) 1 byte f_PCT f_PCT 17 1 byte	DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2	bhicle VSS DLC 14 3 ehicle VEP DLC 14	+ RPM + VSS (t 2 b VSS_H 4 2 + ECT + V 2 b VEP_H	FLI + AA unsign) ytes VSS_L 5 ETP + M/ /EP (unsig ytes VEP_L	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP	PCT Data unsign) ytes RPM_L 7 Data (Bin: ECT (sign) 1 byte ECT	(Binary) FLI (t FLI FLI FLI: 8 ary) ETP (t 1 byte ETP	insign) 1 byte f_FLI f_FLI 9 unsign) 1 byte f_ETP	2 b AAT_H 10 M 2 b MAF_H	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L	n) 1 byte f_AAT T 12 gn) 1 byte f_MAF	EFR_H 13 DIST_H1	EFR (unsign ytes EFR_L EFR.f_EFR 14 DIST_H2	1) 1 byte f_EFR 15 IST (unsig ytes DIST_L1	PCT ( 1 byte PCT PCT. 16 n)	unsign) 1 byte f_PCT f_PCT 17 1 byte f_DIST	DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10	= 0xA1; H STX 0x02 1 = 0xA2; H STX 0x02	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2	hicle VSS DLC 14 3 hicle VEP DLC 14	+ RPM + VSS (t 2 b VSS_H 4 + + ECT + V 2 b VEP_H	FLI + AA insign) ytes VSS_L 5 ETP + MA 'EP (unsig ytes VEP_L VEP_L	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P	PCT Data unsign) ytes RPM_L 7 Data (Bina ECT (sign) 1 byte ECT	(Binary) FLI (t FLI FLI FLI. 8 ary) ETP (t 1 byte ETP ETP.	Insign) 1 byte f_FLI f_FLI 9 unsign) 1 byte f_ETP f_ETP	2 b AAT_H 10 MAF_H MAF_H	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsign ytes MAF_L IAF.f_MA	1) 1 byte f_AAT T 12 1 byte f_MAF AF	EFR_H 13 DIST_H1	EFR (unsign ytes EFR_L EFR.f_EFF 14 D 4 by DIST_H2 D	1) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS	PCT ( 1 byte PCT PCT: 16 n) DIST_L2 T	unsign) 1 byte f_PCT 17 17 1 byte f_DIST	DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10	= 0xA1; H STX 0x02 1 = 0xA2; H STX 0x02 1	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2	shicle VSS DLC 14 3 shicle VEP DLC 14	+ RPM + VSS (t 2 b VSS_H 4 2 + ECT + V 2 b VEP_H 4	FLI + AA insign) ytes VSS_L 5 ETP + MA /EP (unsig ytes VEP_L VEP_L VEP_L 5	T + EFR + RPM ( 2 b RPM_H 6 1 1 byte f_VEP P 6	PCT Data unsign) ytes RPM_L 7 Data (Bini ECT (sign) 1 byte ECT 7	(Binary) FLI (t I byte FLI FLI 8 ary) ETP (t 1 byte ETP ETP. 8	Insign) 1 byte f_FLI f_FLI 9 unsign) 1 byte f_ETP f_ETP 9	2 b AAT_H 10 MAF_H MAF_H N 10	AAT (sigr ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L IAF.f_MA 11	1) 1 byte f_AAT T 12 1 byte f_MAF AF 12	EFR_H 13 DIST_H1 13	EFR (unsign ytes EFR_L EFR.f_EFF 14 D 4 by DIST_H2 D 14	1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15	PCT () PCT PCT: 16 n) DIST_L2 T 16	unsign) 1 byte f_PCT 17 1 byte f_DIST 17	DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03 19
Response DLE 0x10 0 Response DLE 0x10	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02 1	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2	hicle VSS DLC 14 3 hicle VEP DLC 14 3	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4	FLI + AA insign) ytes VSS_L 5 ETP + MA /EP (unsig ytes VEP_L VEP_L VEP.f_VE 5	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6	PCT Data unsign) ytes RPM_L 7 Data (Binn ECT (sign) 1 byte ECT 7	(Binary) FLI (t I byte FLI FLI 8 ary) ETP (t 1 byte ETP ETP.1 8	Insign) 1 byte f_FLI 9 1 byte f_ETP 9 1 byte f_ETP 9	2 b AAT_H 10 MAF_H N 10	AAT (sigr ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L IAF.f_M/ 11	1) 1 byte f_AAT 12 1 byte f_MAF AF 12	EFR_H 13 DIST_H1 13	EFR (unsign ytes EFR_L EFR.f_EFF 14 DIST_H2 DIST_H2 14	1) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 T 16	unsign) 1 byte f_PCT f_PCT 17 1 byte f_DIST 17	DLE 0x10 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19
Response DLE 0x10 0 Response DLE 0x10 0 Response	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02 1 = 0xA3: I	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve	blicle VSS DLC 14 3 blicle VEP DLC 14 3 chicle ABF	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 + IAT + 1	FLI + AA ansign) ytes VSS_L 5 ETP + MA /EP (unsig ytes VEP_L VEP_L VEP_L FRP + EO	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F	PCT Data unsign) ytes RPM_L 7 Data (Binn ECT (sign) 1 byte ECT 7 8inary)	(Binary) FLI (t 1 byte FLI FLI 8 ary) ETP (t 1 byte ETP.t 8	Insign) 1 byte f_FLI 9 1 byte f_ETP 9 1 byte f_ETP 9	2 b AAT_H 10 MAF_H N 10	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L AAF.f_M/ 11	1) 1 byte f_AAT 12 1 byte f_MAF AF 12	EFR_H 13 DIST_H1 13	EFR (unsign ytes EFR_L EFR.f_EFF 14 DIST_H2 DIST_H2 14	1) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 T 16	unsign) 1 byte f_PCT f_PCT 17 1 byte f_DIST 17	DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03 19
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE	= 0xA1: F STX 0x02 1 = 0xA2: F STX 0x02 1 = 0xA3: F STX	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve CMD	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABF DLC	+ RPM + VSS (t 2 b VSS_H 4 2 + ECT + VEP_H 4 2 - ECT + VEP_H 4 2 - ECT + 1 2 - ECT + 1 2 - ECT + 2 - E + 2 - ECT + 2 - E + 2 - E + -	FLI + AA msign) ytes VSS_L 5 ETP + M4 /EP (unsig ytes VEP_L VEP_L 5 FRP + EO unsign)	T + EFR + RPM ( 2 b RPM_H 6 1 1 byte f_VEP P 6 T1 Data (F	PCT Data unsign) ytes RPM_L 7 Data (Binn ECT (sign) 1 byte ECT 7 Binary) IAT (sign)	(Binary) FLI (t 1 byte FLI FLI 8 ary) ETP (t 1 byte ETP.( 8	Insign) 1 byte f_FLI f_FLI 9 unsign) 1 byte f_ETP 9 F FTP 9 F	2 b AAT_H 10 MAF_H N 10 RP (unsign	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L MAF.f_M/ 11	a) 1 byte f_AAT T 12 1 byte f_MAF AF 12	EFR_H 13 DIST_H1 13 EOT1 (sign	EFR (unsign ytes EFR_L EFR_L EFR.f_EFF 14 DIST_H2 DIST_H2 DIST_H2 DIST_H2 DIST_H2	a) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15 Run Tim	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 ST 16 e (unsign)	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00	DLE 0x10 18 DLE 0x10 18 DLE	ETX 0x03 19 ETX 0x03 19 19 ETX
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE 0x10	= 0xA1; I STX 0x02 1 = 0xA2; I STX 0x02 1 = 0xA2; I STX 0x02	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve CMD 0xA3	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABF DLC 11	+ RPM + VSS (t 2 b VSS_H 4 4 + ECT + V 2 b VEP_H 4 4 + IAT + I ABP ( 1 byte	FLI + AA Insign) ytes VSS_L 5 ETP + M// ZEP (unsign) VEP_L VEP_L VEP.f_VE 5 FRP + EO unsign) 1 byte	T + EFR + RPM ( 2 b RPM_H 6 NF + DIST n) 1 byte f_VEP P 6 T1 Data (F 2 b	PCT Data unsign) ytes RPM_L 7 Data (Binn ECT (sign) 1 byte ECT 7 Binary) IAT (sign) ytes	(Binary) FLI (t I byte FLI FLI 8 ary) ETP (t I byte ETP. 8 8	Insign) 1 byte f_FLI f_FLI 9 1 byte f_ETP f_ETP 9 F ETP 9 F ETP 9 F ETP 9 F ETP 9 F ETP 9 F 2 b F F F F F F F F F F F F F	2 b AAT_H 4 10 MAF_H N 10 RP (unsign ytes	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L IAF.f_M/ 11 1 1 byte	a) 1 byte f_AAT T 12 1 byte f_MAF AF 12 2 b	EFR_H 13 DIST_H1 13 EOT1 (sign ytes	EFR (unsign ytes EFR_L EFR_f_EFF 14 D 4 b; DIST_H2 D 14 1 1 byte	a) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15 Run Timo 2 b	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 ST 16 e (unsign) ytes	unsign) 1 byte f_PCT f_PCT 17 1 byte f_DIST 17 0x00 1 byte	DLE 0x10 18 DLE 0x10 18 18 DLE 0x10	ETX 0x03 19 ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE 0x10	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02 1 = 0xA3: I STX 0x02	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve CMD 0xA3	hicle VSS DLC 14 3 hicle VEP DLC 14 3 chicle ABF DLC 11	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 2 + IAT + I ABP ( 1 byte ABP	FLI + AA insign) ytes VSS_L 5 ETP + M/ /EP (unsig ytes VEP_L VEP.f_VE 5 FRP + EO unsign) 1 byte f_ABP	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F 2 b IAT_H	PCT Data unsign) ytes RPM_L 7 Data (Bini ECT (sign) 1 byte ECT 7 Binary) IAT (sign) ytes IAT_L	(Binary) FLI (t 1 byte FLI FLI 8 ary) ETP (t 1 byte ETP ETP 8 1 byte f_IAT	Insign) 1 byte f_FLI f_FLI 9 1 byte f_ETP f_ETP 9 FETP 2 b FRP_H	2 b AAT_H / 10 MAF_H N 10 RP (unsign ytes FRP_L	AAT (sign ytes AAT_L AAT.f_AA 11 IAF (unsig ytes MAF_L AAF.f_MA 11 1 1 J byte f_FRP	1) 1 byte f_AAT T 12 1 byte f_MAF AF 12 2 b EOTI_H	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L	EFR (unsign ytes EFR.f_EFF 14 D DIST_H2 D IST_H2 D I 14 1 byte f_EOT1	a) 1 byte f_EFR 15 15 15 15 15 15 Run Time 2 b RUNTM_H	PCT ( 1 byte PCT PCT: 16 DIST_L2 T 16 e (unsign) ytes RUNTM_L	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00 1 byte	DLE 0x10 18 DLE 0x10 18 18 DLE 0x10	ETX 0x03 19 ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE 0x10	= 0xA1: F STX 0x02 1 1 = 0xA2: F STX 0x02 1 = 0xA3: F STX 0x02	CMD 0xA1 2 CMD 0xA2 CMD 0xA2 2 CMD 0xA3	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABF DLC 11	+ RPM + VSS (t 2 b VSS_H 4 + ECT + VEP_H 4 VEP_H 4 + IAT + 1 ABP (t 1 byte ABP.	FLI + AA insign) ytes VSS_L 5 ETP + M/ ZEP (unsig ytes VEP.f_VE 5 FRP + EO unsign) 1 byte f_ABP	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F 1 Data (F 1 Data (F) 1 Data (F)	PCT Data unsign) ytes RPM_L 7 Data (Bin: ECT (sign) 1 byte ECT 7 inary) IAT (sign) ytes IAT_L IAT.f_IAT	(Binary) FLI (t 1 byte FLI FLI 8 ary) ETP (t 1 byte ETP 1 8 8 1 byte f_IAT	Insign) 1 byte f_FLI f_FLI 9 uunsign) 1 byte f_ETP 9 FETP 9 FETP 9 1 byte f_ETP 9 1 byte f_FLI 1 byte f_FLI 1 byte f_FLI f_FFLI f_FLI	2 b AAT_H / 10 MAF_H 10 RP (unsign ytes FRP_L FRP.f_FR	AAT (sigr ytes AAT_L AAT_F_AA 11 IAF (unsig ytes MAF_L AF.f_M/ 11 1 byte f_FRP P	n) 1 byte f_AAT T 12 1 byte f_MAF AF 12 2 b EOT1_H E	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L OT1.f_EO	FR (unsign ytes EFR_LEF 14 DIST_H2 DIST_H2 D 14 1 1 1 byte f_EOT1 T1	1) 1 byte f_EFR 15 15 IST (unsig ytes DIST_L1 IST.f_DIS 15 Run Tim 2 b RUNTM_H Resc	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 T 16 e (unsign) ytes RUNTM_L erved	unsign) 1 byte f_PCT f_PCT 17 1 byte f_DIST 17 0x00 1 byte	DLE 0x10 18 DLE 0x10 18 DLE 0x10	ETX 0x03 19 ETX 0x03 19 ETX 0x03
Response DLE 0x10 Response DLE 0x10 0 Response DLE 0x10 0 0	= 0xA1; F STX 0x02 1 1 = 0xA2; F STX 0x02 1 = 0xA3; F STX 0x02 1	CMD 0xA1 2 CMD 0xA2 CMD 0xA2 2 CMD 0xA3 2 2	chicle VSS DLC 14 3 chicle VEP DLC 14 3 chicle ABF DLC 11 3 3	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 + IAT + I ABP ( 1 byte ABP ABP, 4	FLI + AA insign) ytes VSS_L 5 ETP + M/ ZEP (unsig ytes VEP_L VEP.f_VE 5 FRP + EO unsign) 1 byte f_ABP 5	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F 1 Data (F 1 AT_H 6	PCT Data unsign) ytes RPM_L 7 Data (Bini ECT (sign) 1 byte ECT 7 Sinary) IAT (sign) ytes IAT_L IAT.f_IAT 7	(Binary) FLI (t 1 byte FLI FLI.1 8 ary) ETP (t 1 byte ETP ETP.1 8 1 byte f_LAT 8	msign) 1 byte f_FLI f_FLI 9 1 byte f_ETP f_ETP 9 E E F 9 F F P 9 F F P 9 F F P 9 F F P 9 F P 9 F P 9 F P 9 F P 9 F P 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 9 F 7 9 F 7 F 7 9 F 7 F 9 F 7 F 7 9 F 7 P 7 P 7 P 7 P 7 P 7 P 7 P 9 P P P P P P P P P P P P P	2 b AAT_H 4 10 MAF_H N 10 RP (unsign ytes FRP_L FRP_F_FRI 10	AAT (sigr ytes AAT_f_AA AT_f_AA 11 IAF (unsig ytes MAF_L IAF.f_M/ 11 1 byte f_FRP P 11	) 1 byte f_AAT T 12 1 byte f_MAF 12 1 byte f_MAF 12 2 b EOTI_H EOTI_H EOTI_H	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L EOT1.f_EO 13	EFR (unsign ytes EFR_L EFR_L EFR_T_EFR_T_14 DIST_H2 DIST_H2 DIST_H2 DIST_H2 T 14 14 1 byte f_EOT1 T1 14	1) 1 byte f_EFR 3 15 15 15 15 Run Time 2 by RUNTM by RUNTM by RUNTM by RUNTM by RUNTM by	PCT ( 1 byte PCT PCT. 16 DIST_L2 T 16 e (unsign) ytes RUNTM_L reved 16	unsign) 1 byte f_PCT 17 17 10 1 byte f_DIST 17 0x00 1 byte 17 17 17 17 17 17 17 17 17 17	DLE 0x10 18 DLE 0x10 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19
Response DLE 0x10 Response DLE 0x10 Response DLE 0x10 0 Response DLE 0x10	= 0xA1: L STX 0x02 1 1 = 0xA2: L STX 0x02 1 = 0xA3: L STX 0x02 1 = 0xA3: L STX 0x02 1 = 0xA3: L STX	CMD OxA1 2 Receive Ve CMD 0xA2 2 CMD 0xA2 2 CMD 0xA3 2 2 CMD 0xA3	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABB DLC 11 3 Second 2	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 2 + IAT + 1 ABP ( 1 byte ABP.4 4 4	FLI + AA ansign) ytes VSS_L 5 ETP + MA ZEP (unsig Ytes VEP_L VEP_f_VE 5 FRP + EO unsign) 1 byte f_ABP 5	T + EFR + RPM ( 2 b RPM_H 6 NF + DIST n) 1 byte f_VEP P 6 T1 Data (F 1 Data (F 1 AT_H 6	PCT Data unsign) ytes RPM_L 7 Data (Bin ECT (sign) 1 byte ECT 7 Sinary) IAT (sign) ytes IAT_L IAT_L IAT_L IAT_L	(Binary) FLI (t I byte FLI FLI. 8 ary) ETP (t I byte ETP. ETP. 8 1 byte f_IAT 8	Insign) 1 byte f_FLI f_FLI 9 1 byte f_ETP f_ETP f_ETP 9 F FTP 9 1 9 9 1 9 9 1 9 9 9 1 9 9 1 9 9 9 1 9 9 1 9 9 1 9 9 1 9 9 1 9 1 9 9 1 1 1 1 1 1 1 1 1 1 1 1 1	2 b AAT_H / 10 MAF_H N 10 RP (unsigg FRP_L FRP_L FRP_L 10	AAT (sigr ytes AAT.f_AA AT.f_AA 11 IAF (unsig ytes MAF_L IAF.f_M/ 11 1 byte f_FRP P 11	) 1 byte f_AAT T 12 1 byte f_MAF AF 12 EOT1_H EOT1_H EOT1_H	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L OT1.f_EO 13	EFR (unsign ytes EFR_L EFR.f_EFF 14 DIST_H2 DIST_H2 DIST_H2 14 1 1 byte f_EOT1 T1 14	a) 1 byte f_EFR 3 15 IST (unsigned tes DIST_L1 IST.f_DIS 15 15 Run Time 2 b RUNTM_H Rese 15	PCT ( 1 byte PCT PCT 16 n) DIST_L2 T 16 e (unsign) ytes RLNTM_L reved 16	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00 1 byte 17 17	DLE 0x10 18 DLE 0x10 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE 0x10 0 0 Response	= 0xA1: F STX 0x02 1 = 0xA2: F STX 0x02 1 = 0xA3: F STX 0x02 1 = 0xA3: F STX 0x02 1 = 0xA3: F STX	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve CMD 0xA3 2 Receive Ve CMD 0xA3	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABF DLC 11 3 Sensor Da	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 + IAT + 1 ABP ( 1 byte ABP ABP. 4 4 -	FLI + AA ansign) ytes VSS_L 5 ETP + M/ ZPP (unsig ytes VEP_f VE 5 FRP + EO unsign) 1 byte f_ABP f_ABP 5 5	T + EFR + RPM ( 2 b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F 2 b IAT_H 6 4 6	PCT Data unsign) ytes RPM_L 7 Data (Bin ECT (sign) 1 byte ECT 7 Binary) IAT (sign) ytes IAT_L IAT_L IAT_F_IAT 7	(Binary) FLI (t FLI t FLI FLI FLI FLI 8 ary) ETP (t 1 byte ETP 4 8 1 byte f_IAT 8	Insign) 1 byte f_FLI f_FLI 9 1 byte f_ETP f_ETP 9 9 FETP 9 FETP 9 9 FRP_H 9	2 b AAT_H / 10 MAF_H N 10 RP (unsign ytes FRP_L FRP.f_FR1 10	AAT (sigr ytes AAT_L AAT_L 11 IAF (unsig ytes MAF_L AAF.f_M/ 11 1 byte f_FRP P 11	a) 1 byte f_AAT T 12 1 byte f_MAF 12 2 b EOT1_H EDT1_H E	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L OT1.f_EO 13	FR (unsign FFR_L EFR_L EFR_f_EFF 14 D 4 by DIST_H2 D 14 1 1 1 byte f_EOT1 T1 14	1) 1 byte f_EFR 15 15 IST (unsig ytes DIST_L1 IST.f_DIS 15 Run Tim 2 b RUNTM_H Ress 15	PCT ( 1 byte PCT PCT. 16 n) DIST_L2 T 16 c (unsign) ytes RUNTM_L reved 16	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00 1 byte 17 17	DLE 0x10 18 DLE 0x10 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19 19
Response DLE 0x10 0 Response DLE 0x10 0 Response 0x10 0 Response (available DLE	= 0xA1: F STX 0x02 1 = 0xA2: F STX 0x02 1 = 0xA3: F STX 0x02 1 = 0xA3: F STX 0x02 1 = 0xA3: F STX 0x02 1 = 0xA2: F STX 0x02 	CMD CMD 0xA1 2 2 CMD 0xA2 2 2 CMD 0xA3 2 2 CMD 0xA3 2 2 CMD 0xA3 2 2 CMD 0xA1	hicle VSS DLC 14 3 hicle VEP DLC 14 3 hicle ABF DLC 11 3 Sensor Da t, lowest 4 DLC	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H - + IAT + I ABP ( 1 byte ABP ABP. 4 ta (Binary, bbit is una	FLI + AA insign) ytes VSS_L S ETP + M/ EP (unsig ytes VEP_L VEP.f_VE 5 FRP + EO unsign) 1 byte f_ABP f_ABP 5 valiable d	T + EFR + RPM ( 2b RPM_H 6 AF + DIST n) 1 byte f_VEP P 6 T1 Data (F IAT_H 6 ata) XX YY	PCT Data unsign) ytes RPM_L 7 Data (Bin: ECT (sign) 1 byte ECT 7 Binary) IAT (sign) ytes IAT_L IAT.f_IAT 7 ZZ (sign)	(Binary) FLI (t FLI FLI FLI FLI ETP (t 1 byte ETP ETP. 8 1 byte f_IAT 8	Insign) 1 byte f_FLI 9 1 byte f_ETP 9 FETP 9 FRP_H 9 9	2 b AAT_H 10 2 b MAF_H N 10 RP (unsign ytes FRP_L FRP_f_FRI 10	AAT (sign ytes AAT_L 11 IAF (unsig ytes MAF_L 11 11 MAF_f_MAF_L 11 11 11 11 11 11 11 11 11 11 11 11 11	) 1 byte f_AAT T 12 1 byte f_MAF 12 1 byte f_MAF F 12 1 byte f_MAF 12 1 byte f_12 1	EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L OT1.f_EO 13 0,	FR (unsign test EFR_L EFR.f_EFR 14 DIST_H2 DIS	1) 1 byte f_EFR 15 IST (unsig ytes DIST_L1 IST.f_DIS 15 Run Timm 2 b RUNTM_H Reso 15	PCT ( 1 byte PCT PCT: 16 n) DIST_L2 TT 16 e (unsign) ytes RUNTM_L erved 16	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00 1 byte 17 17	DLE 0x10 18 DLE 0x10 18 18 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19 19 19 19
Response DLE 0x10 0 Response DLE 0x10 0 Response 0LE 0x10 0 Response (available DLE 0x10	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02 1 = 0xA2: I STX 0x02 1 = 0xA3: I STX 0x02	Receive Ve CMD 0xA1 2 Receive Ve CMD 0xA2 2 Receive Ve CMD 0xA3 2 2 Receive Ve CMD 0xA3	hicle VSS DLC 14 3 bhicle VEP DLC 14 3 bhicle ABF DLC 11 3 sensor Da t, lowest 4 DLC 6	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V2 b VEP_H 4 - + IAT + 1 ABP ( 1 byte ABP J ABP J 4 - - - - - - - - - - - - -	FLI + AA           ytes           VSS_L           5           ETP + M/           ETP + M/           VEP_L           VEP_L           VEP_L           PRP + EO           I hyde           5           yatalable d	T + EFR + RPM ( 2 b RPM_H 6 1 byte f_VEP P 6 T1 Data (E 1AT_H 6 IAT_H 6 XX YY XX YY	PCT Data unsign) ytes RPM_L 7 Data (Bin ECT (sign) I byte ECT 7 Binary) IAT (sign) ytes IAT_L IAT_L IAT_L IAT_L IAT_L IAT_L Y_L/H(6.7)	(Binary) FLI (i FLI FLI FLI B Try) ETP (i ETP (i ETP) ETP ETP ETP B B ETP ETP ETP S B I byte S S S S S S S S S S S S S S S S S S S	Insign) 1 byte f_FLI 9 1 byte f_FTP 9 FETP 9 FETP 9 FRP_H 9 (8.9)	2 b AAT_H // 10 MAF_H MAF_H N 10 RP (unsign RP (unsign RP (res FRP_L 10	AAT (sign ytes AAT_L AAT_L AAT_f_AA 11 IAF (unsig ytes MAF_L MAF_f_M/ 11 1 byte f_FRP 9 11	a) 1 byte f_AAT T 12 1 byte f_MAF 12 2 b EOTI_H EOTI_H 12 12	EFR_H 2 bb EFR_H 13 DIST_H1 13 EOT1 (sign ytes EOT1_L OT1.f_EO 13 0x 8 bb	EFR (unsign ytes EFR_L EFR_f_EFF 14 DIST_H2 DIST_H2 DIST_H2 0 14 14 1 1 1 byte f_EOT1 T1 14	1) 1 byte 1 [EFR 15 15 15 15 15 15 15 Run Tim 2 b RUNTM, H <b>Rese</b> 15	PCT (1 1 byte PCT PCT: 16 n) DIST_L2 T T 16 e (unsign) ytes R(NTM_L) 16	unsign) 1 byte f_PCT 7 17 1 byte f_DIST 17 0x00 1 byte 17 17	DLE 0x10 18 DLE 0x10 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19 ETX 0x03
Response DLE 0x10 0 Response DLE 0x10 0 Response DLE 0x10 0 Response (available DLE 0x10 0 0	= 0xA1: I STX 0x02 1 = 0xA2: I STX 0x02 1 = 0xA3: I STX 0x02 1 = 0xB1: F e data len STX 0x02 1	Receive Vc MD OxA1 2 Receive Vc CMD 0xA2 2 2 Receive Vc CMD 0xA3 2 2 Receive G- CMD 0xA1 2 2 Receive Vc CMD 0xA3 2 2 Receive Vc	hicle VSS DLC 14 3 Chicle VEP DLC 14 3 3 Chicle ABI DLC 11 3 Sensor Da C DLC 11 0 C 6 6 3	+ RPM + VSS (t 2 b VSS_H 4 + ECT + V 2 b VEP_H 4 - ABP ( 1 byte ABP ABP. 4 4 - - - - - - - - - - - - -	FLI + AA msign) VtsS_L ETP + ML/ ETP (unsign) T byte T VEP.f_VEE 5 FRP + EOC f_ABP A	T + EFR + ( RPM ( 2 b RPM_H 6 1 byte f_VEP P 6 11 Data (f VEP P 6 11 Data (f 1 J 2 b 1 AT_H 6 XX YY (H(4,5), Y 6	PCT Data unsign) ytes RPM_L 7 7 1 byte ECT (sign) 1 byte ECT 7 inary) I LAT (sign) ytes IAT_L IAT (sign) ytes IAT_L ZZ (sign) 7	(Binary) FLI( 1 byte FLI FLI 8 ary) ETP (f ETP (f ETP f I byte f I byte f I a S a S a S a S a S a S a S a S a S a S	Insign) 1 byte f_FL1 9 1 byte f_ETP 9 FETP 9 FETP 9 1 byte f_ETP 9 0 1 byte f_FL1 9 1 byte f_FTP 9 1 byte f_FTP 9 9 1 byte f_FTP 1 byte f_FTP 9 9 1 byte f_FTP 1 byte f_FTPP 1 byte f_FTPP 1 byte f_FTPP 1 byte f_FTPP 1 byte f_FTPP 1 byte f_FTPP 1 byte f_FTPPP 1 byte f_FTPPPPPPP 1 byte f_	2 b AAT_H / 10 MAF_H N 10 RP (unsign res FRP_L 10 10	AAT (sigr ytes AAT_L AAT_L AAT_L AAT_L AAT_L AAT_L AAT_L II II II II II II II II II II II II II	a) 1 byte f_AAT T 12 m) 1 byte f_MAF 12 12 12 12 12 12 12 12 12 12	E E 2 b 2 b 2 b 2 b 2 b 2 b 2 b 2 b 2 b	EFR (unsign ytes EFR_L EFR_f_EFF 14 DD 4 b; DIST_H2 D 14 14 14 14 14 14 14 14 14 14 14 14 14	1) 1 byte f_EFR 15 15 15 15 15 15 15 15 15 15	PCT (i 1 byte PCT PCT. 16 n) DIST_L2 T 16 e (unsign) ytes RLNTM_1 16 16	unsign) 1 byte f_PCT 17 1 byte f_DIST 17 0x00 1 byte 17 17 17 17 17 17 17	DLE 0x10 18 DLE 0x10 18 DLE 0x10 18 18 DLE 0x10 18	ETX 0x03 19 ETX 0x03 19 ETX 0x03 19 ETX 0x03 19

# 10. Appendix C – RIFA BLE Transmit Protocol Format