



FOX AVL

COMMUNICATION PROTOCOL

Document version: **1.7.0**

Last revision: **25.04.2013**

Applicable with: **Fox2.1** firmware version: 2.0.25. / 0.1.0.31,

FoxLite1.1 firmware version: 1.1.27,

FoxLite2.0 firmware version: 1.2.11

FoxLite3.0 firmware version: 1.0.1223

FoxBasic firmware version: 1.0.1223



geneko
RB GENERALEKONOMIK

www.geneko.rs
support@geneko.rs
+ 381 (11) 3340 591
+ 381 (11) 3340 178
Bulevar despota Stefana 59a/1 11000 Belgrade, Serbia

1 TABLE OF CONTENTS

1	Table of Contents	1
2	Release notes	2
3	Introduction	3
3.1	General Note	3
3.2	Copyright Note	3
4	Communication Package Formats	4
4.1	Old Binary protocol	4
4.1.1	Message Type - description	5
4.1.2	GPS data – reserved field	5
4.1.3	Vehicle status subfield description	5
4.1.4	Checksum field	6
4.1.5	Message ID field	7
4.1.6	CAN data field	12
4.2	XML data protocol	14
4.2.1	Example of three messages in a row	15
4.2.2	List of events and statuses options	15
4.3	Binary data protocol (BINv1)	16
4.4	Binary data protocol with offset (BINv2)	17
4.5	Binary data protocol with encryption (CryptoBINv1)	18
4.6	Binary data protocol with offset and encryption (CryptoBINv2)	19
4.7	Binary data protocol with authentication	20
4.8	Gateway data protocol	21
4.8.1	Fox to Server gateway message format	21
4.8.2	Server to Fox gateway message format	22
5	Remote Configuration Package Format	23

2 RELEASE NOTES

New features comparing to previous version Communication protocol **1.5**

- Remote configuration commands inserted.
- New event 39 – green drive data packet.
- Fixed typo in event field length (field may be longer than 64 bytes).
- Input / Output states description updated.
- New event 32 – fuel flow analog meter data packet. Consist of 20 values (0..255) – 12.5ml pulse counts.
- New events 33 and 34 – general purpose A/D conversion data packets, for inputs 0 and 1, respectively.
- New event code 41 used for CAN/FMS data transport. Event code 40 is now backward compatibility only.
- Introduction of CAN data field – related to event code 41.
- Updated event code 110 – respond on iButton List (IBL) command.
- New event codes 111 and 112 used to inform of iButton UID insertion / deletion.
- New event codes 210, 211 and 212 used to transfer phonebook listing, insertion & deletion data.
- New event codes 120, 121, 122, 123 and 124 supporting GeoFence functionality. Codes states for GF_IN, GF_OUT events and as responses to: GF_ADD, GF_DELETE & GF_READ, respectively.
- New event code 1130 generated on crash detection (breach of preconfigured acceleration level).
- Status update on event codes support on FoxBasic and FoxLite3.0 devices.
- New document format.

New features comparing to previous version Communication protocol **1.3**

- New events 1201 and 1202 private drive started/stopped.
- In binary protocol new subfield EngineTemp inside field vehicle status
- In binary protocol new subfield Distance inside field vehicle status
- In binary protocol new subfieldADGP1 and ADGP2 inside field vehicle status
- In binary protocol correction status 29,51,52 has event data 8Bytes long
- In binary protocol all events can have vehicle status data
- In XML protocol vehicle status has 9 subfields separated by comas instead of 8
- In XML protocol new subfield EngineTemp inside field vehicle status
- In XML protocol new subfield Distance inside field vehicle status
- In XML protocol new subfield ADGP1 and ADGP2 inside field vehicle status

3 INTRODUCTION

3.1 GENERAL NOTE

We would like to thank you for your interest in Geneko Fox AVL FoxLite AVL product. This document is a description of communication protocol between Fox/FoxLite/FoxBasic devices and server to which data are sent, as well as a protocol for parameters remote change of Fox/FoxLite/FoxBasic devices.

Please note that protocol support for each device depends on product's featured functionalities and firmware revision. For more complete information, look at the products feature matrix, device datasheet or contact our sales department. Also note that product series Fox and FoxLite prior to 3.0 will soon meet EOP and the support for them shall remain legacy. Future revisions of protocols and this document will be related to FoxBasic, FoxLite3.0 and newer devices.

The information in this guide is subject to change without notice. This document may include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Geneko may make improvements in or changes to the protocols or software described in this document at any time.

3.2 COPYRIGHT NOTE

THIS DOCUMENT, THE SOFTWARE AND PROTOCOLS DESCRIBED WITHIN THIS DOCUMENT ARE OWNED BY RB "GENERAL EKONOMIK" D.O.O., ("GENEKO") AND ARE FURNISHED UNDER AND SUBJECT TO THE TERMS OF A LICENSE AGREEMENT AND/OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, GENEKO PROVIDES THIS DOCUMENT, PROTOCOLS AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OR EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE THIS STATEMENT MAY NOT APPLY TO YOU. YOU SHOULD CAREFULLY REVIEW THE TERMS OF THE LICENSE AGREEMENT AND/OR NON-DISCLOSURE AGREEMENT THAT ACCOMPANIED THIS DOCUMENT OR SOFTWARE.

This guide contains proprietary technical information regarding Fox devices product series, protected by copyright. No part of this guide may be photocopied or reproduced in any form without prior written consent from Geneko.

Copyright ©2009 – 2012 RB "GENERAL EKONOMIK", d.o.o. All rights reserved.

Geneko is a trademark of RB "GENERAL EKONOMIK", d.o.o. All other trademarks listed in this document are the property of their respective owners.

4 COMMUNICATION PACKAGE FORMATS

Fox AVL device sends data to remote server using TCP connection. Data are sent to server with static IP address through one of free IP ports (server address and data port parameters are configurable by Fox AVL device menu, remote commands or FoxConfigurator program). Socket connection time out by default is set to 0s, as soon as data are sent Fox device will close the connection. Connection time out can be configured to 60s. Fox AVL device can send data in two predefined opened data formats:

1. Old binary data protocol
2. XML data protocol
3. Binary data protocol (version 1)
4. Binary data protocol (version 1) with offset
5. Binary data protocol (version 1) with encryption
6. Binary data protocol (version 1) with offset and encryption
7. Binary data protocol with authentication

4.1 OLD BINARY PROTOCOL

Fox AVL Binary format is designed to decrease data traffic costs. Using binary format data, load can be reduced up to 60%. 100 bytes in XML format can be translated to 40 bytes in binary format.

Applicable to:

- Fox AVL devices with firmware versions 2.0.8 and above,
- FoxLite AVL devices
- FoxBasic AVL devices.

Note that there are five more different binary protocols, still in development, which shall be used with future devices. Unless stated otherwise, default data storage in binary protocol is **big endian**.

Old binary protocol message format:

bytes	binary value	description
0-1	0x5C 0x72	Start of message
2	variable	Total number of bytes in message including start of message bytes
3	variable	Message type,
4-7	variable	Vehicle ID number, 4-MSB ... 7-LSB
8-9	variable	Message ID, 8-MSB 9-LSB
10	variable	Year & 0x7f [0-99], GPS valid flag & 0x80 [0 valid, 0x80 not valid]
11	variable	month
12	variable	day
13	variable	hour [24 hours encoding]
14	variable	minutes
15	variable	seconds

16-27	variable	GPS data, format MIN GPS data aaaabbbbccccddddeeeeffff MSB(aaaa) & 0x80 latitude northing 0-N, 1-S MSB(aaaa) & 0x7f latitude MSB [min] LSB(aaaa) latitude LSB [min] MSB(bbbb) latitude MSB [min/10000] 4 decimal places LSB(bbbb) latitude LSB [min/10000] 4 decimal places MSB(cccc) & 0x80 longitude easting 0-E, 1-S MSB(cccc) & 0x7f longitude MSB [min] LSB(cccc) longitude LSB [min] MSB(dddd) longitude MSB [min/10000] 4 decimal places LSB(dddd) longitude LSB [min/10000] 4 decimal places MSB(eeee) MSB speed [knots] [1 knot = 1 km/h * 250/463] LSB(eeee) MSB speed [knots] MSB(ffff) MSB direction [deg] LSB(ffff) MSB direction [deg]
28-var	variable	Reserved field, depends of message type and message ID, max length = 256-29
28 +res.len		Checksum

4.1.1 MESSAGE TYPE - DESCRIPTION

This field describes current message type. Reserved values are:

0x56 Fox/FoxLite not encrypted GPS data

0x5a Fox/FoxLite not encrypted parameter data

4.1.2 GPS DATA – RESERVED FIELD

Field Reserved is field with variable length consist of up to four subfields in this specific order: Cell Info, Vehicle Status, Event Data and, optional, CAN Data.

Cell Info - format (1 byte Cell Info length [0-25] + Cell Name) max total 26 bytes

Vehicle Status - format (1 byte Status field length [0 or 16] + Status) max total 17 bytes

Event Data - format (1 byte Event Datalength [0-123] + eventData) max total 124 bytes

CAN Data - format (1 byte CAN Data length [0-117] + 4 byte mask + masked CAN data) max total 118 bytes

Reserved field *theoretical* total max length = 26 + 17 + 124 + 118 = 285B (*real* up to 192B with current FW version).

4.1.3 VEHICLE STATUS SUBFIELD DESCRIPTION

Total length 16 bytes

bytes	description
0	Digital inputs states LSB (MSB I ₇ I ₆ I ₅ I ₄ I ₃ I ₂ I ₁ I ₀ LSB) I ₀ – Driver door status 0-opened, 1 closed I ₁ – External presence sensor 1 Input, 0-activated, 1- not activated I ₂ – Alarm status/external Alarm status, 0-activated, 1- not activated I ₃ – Fox GPI2 default/FoxLite GPIO default, 0-activated, 1- not activated I ₄ – Trunk door status, 0-opened, 1- closed I ₅ – Hood door status, 0-opened, 1- closed I ₆ – Panic taster, 0-activated, 1- not activated I ₇ – Private drive, 0-activated, 1- not activated

1	Digital inputs states MSB (MSB I ₁₅ I ₁₄ I ₁₃ I ₁₂ I ₁₁ I ₁₀ I ₉ I ₈ LSB) I ₈ – Contac Key, 0-off, 1-on I ₉ – Engine on, 0-activated, 1- not activated I ₁₀ – Fox GPIO1 default, 0-activated, 1- not activated I ₁₁ – Internal acceleration sensor, 0-activated, 1- not activated I ₁₂ – FoxLite GPIO1 default, 0-activated, 1- not activated I ₁₃ – FoxLite GPIO2 default, 0-activated, 1- not activated I ₁₄ – FoxLite GPIO3 default, 0-activated, 1- not activated I ₁₅ – FoxLite GPIO4 default, 0-activated, 1- not activated
2	Digital outputs states (MSB O ₇ O ₆ O ₅ O ₄ O ₃ O ₂ O ₁ O ₀ LSB) : Older Fox devices / FoxBasic, FoxLite3.0 O ₀ – Head Lights / Digital output 1 state O ₁ – Output for activation external sensors / Digital output 2 state O ₂ – Central lock / unused O ₃ – Engine blocked / Engine blocked O ₄ – LED signalization / unused O ₅ – Alarm siren / unused O ₆ – Flashes / unused O ₇ – User defined output / unused
3	User analog AD value MSB (on FoxBasic, FoxLite3.0 and newer devices -> external voltage)
4	User analog AD value LSB (on FoxBasic, FoxLite3.0 and newer devices -> external voltage)
5	RPM [rounds/min] value MSB
6	RPM [rounds/min] value LSB
7	FUEL [1%-100%], 0 - no fuel measuring
8	Engine Temperature MSB (for future use;only FoxLite3.0 and newer devices with CAN connection)
9	Engine Temperature LSB (for future use; only FoxLite3.0 and newer devices with CAN connection)
10	User defined AD input 1 (for future use)
11	User defined AD input 2 (for future use)
12 ¹	Distance Traveled MSB
13 ¹	Distance Traveled
14 ¹	Distance Traveled
15 ¹	Distance Traveled LSB

4.1.4 CHECKSUM FIELD

Checksum byte calculation includes all message bytes except start of message bytes.

Checksum calculation algorithm is:

$$CB = 256 - \left(\sum_{i=2}^{len-2} bytes[i] \right) \bmod 256$$

To test the correctness of message summarize all bytes including check sum byte and **without start of message** bytes. If result is 0, message is correct.

4.1.5 MESSAGE ID FIELD

This field describes GPS data message

Message ID	Description
0	Regular GPS data, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(4B or 6B) If 4B (MSB,...,LSB) total distance traveled [m] If 6B (MSB,...,LSB) total distance traveled [m] + (2B)max speed between two position, MSB-integer value of speed and LSB integer value of decimal fraction multiplied with 100
1	Over speed event, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
11	WDT start (Fox Start), reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(1B or 2B) reset code
17 ¹	Internal alarm armed, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
18 ^{2,3}	RPM over limit event, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(2B) – (MSB, LSB RPM value)
19 ¹	Unauthorized vehicle access, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
22 ¹	External alarm activated, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
24 ¹	Internal alarm disarmed, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
27 ³	External power supply below limit event, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(2B) – (MSB,LSB power supply value [100mV])
29 ¹ (deprecated)	Dallas key event, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(8B) - (iButton code 8B)
30 ¹	External alarm deactivated, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
31 ²	Fuel data packet, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(64B) – (32*2B, 32 measured values of fuel, MSB1, LSB1,..., MSB32, LSB32)
32 ²	Fuel flow data packet, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(20B) – (20*1B, 20 measured values of fuel, val1,...,val32)
33 ⁴	ADC0 data packet, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(64B) – (32*2B, 32 values of GP A/D converter, inp0, MSB1, LSB1,..., MSB32, LSB32)
34 ⁴	ADC1 data packet, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(64B) – (32*2B, 32 values of GP A/D converter, inp0, MSB1, LSB1,..., MSB32, LSB32)
39	Green drive packet, reserved – cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(120B) – (15*(3*2B + 2B), 15 values structured values containing: 3 signed short – presenting X, Y & Z coordinates of zero-normalized acceleration vector and 1 unsigned short – current RPM)
40 ² (deprecated)	Additional FMS data, cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB string)
41 ⁵	Additional FMS data with CAN field, cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB string) + can_data_field(max 118B)
45 ¹	GPS Receiver Error 24h without valid signal, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)

46 ¹	Sim Card Connection Error, reserved - cell_info_len(1B) + cell_name(max25B) + (deprecated) status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
47 ¹	Backup Battery Low Message, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(0B) + event_data_len(1B) + event_data(0B)
51	iButton Login Message, reserved - cell_info_len(1B) + cell_name(max16B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(8B) - (iButton code 8B)
52	iButton Logout Message, reserved - cell_info_len(1B) + cell_name(max16B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(8B) - (iButton code 8B)
59 ¹	Euroscan temperature sensor readings, reserved - cell_info_len(1B) + cell_name(max16B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB: string format: sddd.d;sddd.d;sddd.d;sddd.d;sddd.d;sddd.d)
75 ¹ (future use)	Unofficial msg response, reserved - cell_info_len(1B) + cell_name(max16B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB: string)
79 ¹	Official msg response, reserved - cell_info_len(1B) + cell_name(max16B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB: string)
80	Response on parameter change command , reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(0B) + event_data_len(1B) + event_data(nB) (list of parameters n bytes)
81 ¹ (deprecated)	Parameters from group 1 has changed, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
82 ¹ (deprecated)	Parameters from group 2 has changed, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
83 ¹ (deprecated)	Parameters from group 3 has changed, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
84	Delete log file response, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data("ok","err")
85 ¹ (future use)	Ibutton remote change response, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB string)
86	Remote reset of device, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
87 ¹ (deprecated)	RTC has changed remotely, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
88	Firmware updated, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(2B)(updating result)
89	GPS data on demand, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
90	Unknown remote command, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
91	Irregular remote command format, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
98 ¹	Net-list changed response, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(text response)
100	Response on parameter read command, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of parameters n bytes)
101	Response on debug command, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of parameters n bytes)
110(updated)	Response on remote iButton change/list request ⁶ , reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of UIDs, format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 15 UIDs per message

111 ⁴	List of newly added iButton UUIDs, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of UUIDs, format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 15 UUIDs per message
112 ⁴	List of deleted iButton UUIDs, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of UUIDs, format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 15 UUIDs per message
120 ⁴	Geo-fence in event, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(4B) (activated geo-fence zone ID)
121 ⁴	Geo-fence out event, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(4B) (activated geo-fence zone ID)
122 ⁴	Response on geo-fence insertion command, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(5B) (4B geo-fence zone ID + 1B error code)
123 ⁴	Response on geo-fence deletion command, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(5B) (4B geo-fence zone ID + 1B error code)
124 ⁴	Response on geo-fence read command, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (1B count of geo-fence zones in message + count * 4B geo-fence zone IDs)
210 ⁴	Response on remote phonebookread request, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of phonebook entries, each entry consist of: 1B – reserved, 1B flags, 13B name, 16B number; format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 4 items per message
211 ⁴	List of newly added phonebookitems, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of phonebook entries, each entry consist of: 1B – reserved, 1B flags, 13B name, 16B number; format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 4 items per message
212 ⁴	List of deleted phonebookitems, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (list of phonebook entries, each entry consist of: 1B – reserved, 1B flags, 13B name, 16B number; format n*m:0123456789ABCDEF...; n: number of message, m: total messages) max 4 items per message
383 ⁷	Garmin device arrival, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(nB) (2B-product ID(uint16) + 2B- sw version(sint16) + m*2B-m protocols supported (array of sint16))
384 ⁷	Garmin device removal, reserved - cell_info_len(1B) + cell_name(max25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1011/1012 ¹	Driver's door opened/closed, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(0B) + event_data_len(1B) + event_data(0B)
1021	Eksternal alarm sensor activated, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(0B) + event_data_len(1B) + event_data(0B)
1051/1052 ¹	Trunk opened/closed, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(0B) + event_data_len(1B) + event_data(0B)
1061/1062 ¹	Hood opened/closed, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1071	Panic taster activated, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1092	Contact key on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(4B)(distance traveled [m])
1091	Contact key off, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(4B+2B)(distance traveled [m] MSB..LSB, max section speed format aaa.bb[km/h], MSB aaa, and LSB bb)
1122	Internal acceleration sensor event, reserved - cell_info_len(1B) + cell_name(max 25B) +

	status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1130 ⁴	Crash detection event, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1201/1202 ¹	Start/stop private drive, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2010/2011 ¹	Digital output 1 (user defined output GPO) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2020/2021 ¹	Digital output 2 (flash lights) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2030/2031 ¹	Digital output 3 (alarm siren) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2040/2041 ¹	Digital output 4 (LED signalization) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2050/2051	Digital output 5 (engine block) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2060/2061 ¹	Digital output 6 (central lock) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2070/2071 ¹	Digital output 7 (output for external sensors activation) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2080/2081 ¹	Digital output 8 (head lights) turned off/turned on, reserved - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
1000–1990 or ≥3000	Input (X) event code, configurable, defaults to 30Xb (<i>b-logic state</i>) - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)
2000–2990	Output (X+1) event code, configurable, defaults to 21Xb (<i>b-logic state</i>) - cell_info_len(1B) + cell_name(max 25B) + status_info_len(1B) + status(16B) + event_data_len(1B) + event_data(0B)

¹FoxBasic, FoxLite3.0: Not supported in the current firmware

²Not supported on FoxBasic devices and FoxLite prior to 2.0

³FoxBasic, FoxLite3.0 and later devices send data in status field - no event data

⁴Supported only on FoxBasic, FoxLite3.0 and newer devices

⁵ FoxLite3.0 and newer: event_data_len equals 0 or 1. Even_data (if exists) represents CAN protocol revision.

⁶With FoxBasic, FoxLite3.0 and newer devices this is list command (IBL) response

⁷Allowed only on FoxLite3.0 from firmware version 1.0.1223. and above.

4.1.6 CAN DATA FIELD

With new generation of Fox devices (FoxLite3.0 and later), FMS data (previously stored in the event data field section of message ID 40) is sent within newly introduced CAN data field.

CAN data field is sent as part of message ID 41, after `even_data_field`. Indication that can field exists is: `event_data_len = 1` and `event_data[0]` (first and only byte) containing FMS revision = 33 for the current device and firmware version. FMS revision is subject to change with CAN interface updates and FMS data supported.

CAN field has a variable length, depending on data contained within (present data is masked with appropriate bit). For current FMS revision (33) maximal field length is 118 bytes. Note that *bitmask value is served in little endian format*, while other CAN field values are stored in the big endian format (as the rest of old binary protocol values).

Field header (5B): 1B Length + 32bit mask, allowing each parameter to be presented (bit = 1) or defined as not existing (bit = 0). Default value for non-existing parameters is zero („0“).

Rest of the field description

Mask bit num	Length[B]	Name	Unit	Scale / Offset
0	2	Average Speed	Km/h	10
0	2	Maximal Speed	Km/h	10
1	2	Average RPM	rpm	1
1	2	Maximal RPM	rpm	1
2	2	Average Fuel Consumption	l/h	10
2	2	Maximal Fuel Consumption	l/h	10
3	2	Axel Weight 1	Kg	1
4	2	Axel Weight 2	Kg	1
5	2	Axel Weight 3	Kg	1
6	2	Axel Weight 4	Kg	1
7	2	Average Turbo Pressure	bar	10
7	2	Maximal Turbo Pressure	bar	10
8	1	Minimal Engine Temp	deg	1 / +40
8	1	Maximal Engine Temp	deg	1 / +40
9	1	Average Acceleration Pedal	%	1
9	1	Maximal Acceleration Pedal	%	1
10	1	Average Torque	%max	1
10	1	Maximal Torque	%max	1
11	4	Total Fuel Used	l	1
12	4	Fuel Used at Cruise	l	1
13	4	Fuel Used at Drive	l	1
14	4	Idle Longer than 5min	times	1
15	4	Idle Longer than 10min	times	1
16	4	Total Idle Time	sec	1
17	4	Total Time PTO	sec	1
18	4	Time Cruise	sec	1
19	4	RPM > treshold_RPM1	sec	1

20	4	RPM > treshold_RPM2	sec	1
21	4	Speed > treshold_Speed1	sec	1
22	4	Speed > treshold_Speed2	sec	1
23	4	Speed > treshold_Speed3	sec	1
24	4	Brake Applications	times	1
25	4	Clutch Applications	times	1
26	4	Engine On	sec	1
27	4	Time Torque > 90%	sec	1
28	4	Mileage	Km	10
29	11	Tachograph Data	*	/
30		RFU		
31		RFU		

* Tachograph Data is given as binary frame described in the table below:

Byte(s)	Bit(s)	Description	Unit	Scale / Offset
0,1	15-0	Output Shaft Speed (avg)	rpm	1
2,3	15-0	Output Shaft Speed (max)	rpm	1
4,5	15-0	Vehicle Speed (avg)	Km/h	10
6,7	15-0	Vehicle Speed (max)	Km/h	10
8	0	[MD] Motion Detected	/	
	1	[OS] Overspeed Detected	/	
	2	[DI] Direction	/	
	3	[TP] Tacho performance	/	
	4	[HI] Handling Information	/	
	5	[EV] Tachograph Event	/	
	6	n/a	/	
	7	n/a	/	
9	0	[D1A] Driver1 Info (present)	/	
	3-1	[D1WS] Driver1 Working State	/	
	7-4	[D1TS] Driver1 Time State	/	
10	0	[D2A] Driver2 Info (present)	/	
	3-1	[D2WS] Driver2 Working State	/	
	7-4	[D2TS] Driver2 Time State	/	

Notes:

- 1) The appropriate fields (current speed, current engine temperature, fuel level & distance traveled) will be filled with CAN derived data if present.
- 2) For details on field description, read the user manual for external device used.

4.2 XML DATA PROTOCOL

Data is sent from Fox device to server in the following format:

```
<fox>
  <gps id= "vehicleID"
    data= "statusID,Valid[V/A],Date,Time,Latitude,LatDirection,Longitude,LonDirection,Speed,Course,
          GsCellInfo,VehicleStatus,AdditionalDataOnStatus"
    can= "BitMask,AvgSpeed,MaxSpeed,AvgRPM,MaxRPM,AvgFuelConsumption,MaxFuelConsumption,
          AxelWeight1,AxelWeight2,AxelWeight3,AxelWeight4,AvgTurboPress,MaxTurboPress,
          MinEngTemp,MaxEngTemp,AvgAccelPedal,MaxAccelPedal,AvgTorque,MaxTorque,FuelTotal,
          FuelCruise,FuelDrive,Idle5min,Idle10min,TotalIdle,TotalLPT0,TimeCruise,TimeRPM1,TimeRPM2,
          TimeSpeed1,TimeSpeed2,TimeSpeed3,BrakeApps,ClutchApps,TimeEngineOn,TimeHiTorque,Mileage,
          T_Avg0ShaftSpeed,T_Max0ShaftSpeed,T_AvgSpeed,T_MaxSpeed,T_Flags,T_Driver1,T_Driver2"
  />
</fox>
```

In upper format:

- tag **id** defines vehicle identification number,
- tag **data** is standard NMEA message received from GPS receiver in modified format. Beside NMEA sequence it contains:
 - GSM station name,
 - Vehicle status,
 - Data on time and date read from built-in device real time clock in format DDMMYY and HHMMSS (in case data from GPS are valid, date and time are read from GPS receiver).
- tag **can** is optional (sent only with *statusID*=40) and holds the data picked-up from the vehicle CAN bus. Since most vehicles do not support all of the listed values, present data is masked with an appropriate *BitMask* which is always listed at the beginning. All values are presented in hexadecimal number format. For details on each field, refer to section 4.1.6.

Basic message VehicleStatus has the following format:

$I_0I_1I_2I_3I_4I_5I_6I_7I_8I_9I_{10}I_{11}I_{12}I_{13}I_{14}I_{15}$ AccVoltage*10 EngineTemp RPM FuelLevel ADGP₁ ADGP₂ O₁O₂O₃O₄O₅O₆O₇O₈ Distance

Those are nine (9) subfields, separated by space, where I_n [$n=0, 1 \dots 15$] is device input states holder and O_n [$n=1 \dots 8$] is device output states holder. States are expressed in values "0" (logical zero) and "1" (logical one).

Please note that number of supported digital inputs / outputs varies depending on type of FOX device.

Some of the I/O fields may hold special indication functionality, depending on device type & configuration:

- I_0 - Driver door status 0-opened, 1 closed
- I_1 - External presence sensor 1 Input, 0-activated, 1- not activated
- I_2 - Alarm status/external Alarm status, 0-activated, 1- not activated
- I_3 - Fox GPI2 default/FoxLite GPIO0 default, 0-activated, 1- not activated
- I_4 - Trunk door status, 0-opened, 1- closed
- I_5 - Hood door status, 0-opened, 1- closed
- I_6 - Panic taster, 0-activated, 1- not activated
- I_7 - Private drive, 0-activated, 1- not activated
- I_8 - Contac Key, 0-off, 1-on
- I_9 - Engine on, 0-activated, 1- not activated

- I₁₀ - Fox GPI1 default, 0-activated, 1- not activated
- I₁₁ - Internal acceleratino sensor, 0-activated, 1- not activated
- I₁₂ - FoxLite GPIO1 default, 0-activated, 1- not activated
- I₁₃ - FoxLite GPIO2 default, 0-activated, 1- not activated
- I₁₄ - FoxLite GPIO3 default, 0-activated, 1- not activated
- I₁₅ - FoxLite GPIO4 default, 0-activated, 1- not activated

O₁ - Head Lights

O₂ - Output for activation external sensors

O₃ - Central lock

O₄ - Engine blocked

O₅ - LED signalization

O₆ - Alarm siren

O₇ - Flashes

O₈ - User defined output

Notes:

- 1) Invalid data on vehicle position will be sent to server, too. The third window in GPS data structure indicates whether the position is accurate or not (**V** stands for invalid data, **A** for valid ones). Server will decide how these data will be interpreted.
- 2) Time and date sent, represents UTC time and date (UTC time is the same as GMT or differs from it by 1 hour, depending Daylight Saving Time)
- 3) Usage of general purpose A/D fields (ADGP₁& ADGP₂) has been deprecated with devices FoxBasic1.0 & FoxLite3.0. On all newer devices, values from ADC₀& ADC₁ are sent within *AdditionalDataOnStatus* with *statusID* 33 & 34, respectively.
- 4) Although cross-protocol support may vary depending on device type & functionality, data sent within *AdditionalDataOnStatus* aims to be compatible with *event_data* field of binary protocol for same values of *statusID* (XML) and *messageID* (binary). Default data separator is character space. For more information on each *statusID*, please refer to section 4.1.5.

4.2.1 EXAMPLE OF THREE MESSAGES IN A ROW

FoxBasic1.0, FoxLite3.0 & newer devices, send single <gps> node under one <fox> node, oppositely to prior devices which send multiple <gps> nodes under the same <fox> node. Example:

Fox, FoxLite1.x, FoxLite2.0	<pre><fox> <gps id="5" data="0,A,271006,130450,4448.9987,N,02028.2615,E,0,96,Palilula,,0" /> <gps id="5" data="11,A,271006,130454,4448.9987,N,02028.2615,E,0,96,Palilula,,1" /> <gps id="5" data="0,A,271006,130456,4448.9987,N,02028.2615,E,0,96,Palilula,,7" /> </fox></pre>
FoxBasic1.0, FoxLite3.0	<pre><fox><gps id="5" data="0,A,271006,130450,4448.9987,N,02028.2615,E,0,96,Palilula,,0" /></fox> <fox><gps id="5" data="11,A,271006,130454,4448.9987,N,02028.2615,E,0,96,Palilula,,1" /></fox> <fox><gps id="5" data="0,A,271006,130456,4448.9987,N,02028.2615,E,0,96,Palilula,,7" /></fox></pre>

4.2.2 LIST OF EVENTS AND STATUSES OPTIONS

For list of event codes & descriptions, please refer to section 4.1.5.

4.3 BINARY DATA PROTOCOL (BINV1)

This protocol is still in development and will be the default protocol for future Fox series devices.

Binary protocol message format:

bytes	binary value	description
0-1	0x5C 0x65	Start of message
2-4	variable	Total number of bytes in message including start of message bytes (max message length 2^{24-1}), 2-MSB ... 4-LSB
5	variable	Session number, semi-random number
6	variable	Message type
7-16	variable	Vehicle ID number, 10 bytes, 7-MSB ... 16-LSB
17-26	variable	Fox Id tag, 10 bytes, 17-MSB ... 26-LSB
27-28	variable	Message ID, 2 bytes, 8-MSB ... 9-LSB
29-46	variable	GPS data, 18 bytes, format MIN GPS data aaaabbbbccccddddeeeeffff MSB(aaaa) & 0x80 latitude northing 0-N, 1-S MSB(aaaa) & 0x7f latitude MSB [min] LSB(aaaa) latitude LSB [min] MSB(bbbb) latitude MSB [min/10000] 4 decimal places LSB(bbbb) latitude LSB [min/10000] 4 decimal places MSB(cccc) & 0x80 longitude easting 0-E, 1-S MSB(cccc) & 0x7f longitude MSB [min] LSB(cccc) longitude LSB [min] MSB(dddd) longitude MSB [min/10000] 4 decimal places LSB(dddd) longitude LSB [min/10000] 4 decimal places MSB(eeee) MSB speed [knots] [1 knot = 1 km/h * 250/463] LSB(eeee) MSB speed [knots] MSB(ffff) MSB direction [deg] LSB(ffff) MSB direction [deg]
47-var	variable	Reserved field, depends of message type and message ID, max length = 256-29
47+res.len		Checksum

4.4 BINARY DATA PROTOCOL WITH OFFSET (BINV2)

This protocol is still in development.

4.5 BINARY DATA PROTOCOL WITH ENCRYPTION (CRYPTOBINV1)

This protocol is still in development.

4.6 BINARY DATA PROTOCOL WITH OFFSET AND ENCRYPTION (CRYPTOBINV2)

This protocol is still in development.

4.7 BINARY DATA PROTOCOL WITH AUTHENTICATION

This protocol is still in development.

4.8 GATEWAY DATA PROTOCOL

This protocol is designed for communication with external devices connected to fox device. Protocol supports bidirectional transfer of data over GSM network, using fox device as a gateway. Protocol is applicable on FoxLite3.0 device with firmware version 1.0.1223 and above.

Currently, only connection to Garmin GPS devices that support the FMI control specification (*Garmin Fleet Management Interface*, please, refer to Garmin FMI-CS 001-00096-00 Rev. H) can be established. Communication is done with protocol stacks divided in two layers: physical (link) & application.

FMI-enabled device handling is done with protocol stacks divided in two layers: physical (link) & application. Link layer handling, as well as some device-related and driver-related functions of application layer handling is done by Fox devices.

FMI-related (pID = 161) application layer payload should be handled by Server (Section 5.1 in Garmin FMI Specification).

Besides all Packet IDs different from FMI (161), following application-layer FMI packets are handled by Fox:

ID	direction	description	packet data type
0x0000	S -> G	Enable Fleet Protocol Request	fleet_features_data_type
0x0001	S -> G	Product ID Request	-
0x0002	S <- G	Product ID	product_id_data_type
0x0003	S <- G	Protocol Support Data	protocol_array_data_type
0x0813	S <-> G	Server to Client Driver ID	driver_id_d607_data_type
0x0260	F <-> G	Ping Packet ID	-
0x0261	F <-> G	Ping Response Packet ID	-
0x1010	F <-> G	Remote Reboot Request	-

Note:

Base FMI protocol stack version to be supported is 2.6 – features declared as A607 or below. Since there is no plan to support older devices (not supporting FMI 2.6) all features that has deprecated implementation (e.g. A604) should use newer (e.g. A607) declaration only!

4.8.1 FOX TO SERVER GATEWAY MESSAGE FORMAT

bytes	binary value	description
0	0x10	Start of message
1-2	variable	Msg_len (Total number of bytes in message without 0x10 , msg_len , Checksum and 0x03 . 1-MSB 2-LSB)
3	variable	Bit mask (0x80 if only garmin packet is included, high-order bit is enabled)
4-7	variable	Vehicle id (Big-endian, 4-MSB...7-LSB)
8-11	variable	Date and time Unix timestamp (number of seconds from 1.1.1970) (Big-endian, 8-MSB...11-LSB)
12-23	variable	GPS position data, format MIN GPS data aaaabbbbccccddddeeeeffff MSB(aaaa) & 0x80 latitude northing 0-N, 1-S MSB(aaaa) & 0x7f latitude MSB [min] LSB(aaaa) latitude LSB [min]

		MSB(bbbb) latitude MSB [min/10000] 4 decimal places LSB(bbbb) latitude LSB [min/10000] 4 decimal places MSB(cccc) & 0x80 longitude easting 0-E, 1-S MSB(cccc) & 0x7f longitude MSB [min] LSB(cccc) longitude LSB [min] MSB(dddd) longitude MSB [min/10000] 4 decimal places LSB(dddd) longitude LSB [min/10000] 4 decimal places MSB(eeee) MSB speed [knots] [1 knot = 1 km/h * 250/463] LSB(eeee) MSB speed [knots] MSB(ffff) MSB direction [deg] LSB(ffff) MSB direction [deg]
24-var	optional	Gateway data
msg_len + 3	variable	Checksum (CRC8 Dallas Maxim) Polynomial: $x^8 + x^5 + x^4 + x^0$
msg_len + 4	0x03	End of message

Gateway data – Garmin message

bytes	binary value	description
0	variable	Garmin message length
1-2	variable	FMI protocol ID (1-LSB 2-MSB)
3-var	variable	FMI protocol message (section 5.1 of Garmin specification)

4.8.2 SERVER TO FOX GATEWAY MESSAGE FORMAT

bytes	binary value	description
0	0x10	Start of message
1	variable	Msg_len (Total number of bytes in message without 0x10 , Checksum and 0x03)
2-var	variable	Garmin message
msg_len + 1	variable	Checksum(CRC8 Dallas Maxim) Polynomial: $x^8 + x^5 + x^4 + x^0$
msg_len + 2	0x03	End of message

5 REMOTE CONFIGURATION PACKAGE FORMAT

Fox AVL devices support remote commands sent by SMS message or through TCP connection over GPRS. SMS commands are accepted from server phone number (defined with parameter **SN**), or admin flagged phonebook entries. For older device, without smart phonebook support, all phonebook entries are considered to have admin privileges. Note that devices with empty phonebook and blank server phone number parameter will *accept remote command from any phone number*.

Device can also be configured to periodically checks for remote commands from server using TCP connection. This is determined by parameter **auto command check timeout**(Fox Basic, Fox Lite 2.0, Fox Lite 3.0 and newer devices – can be set by FoxConfigurator application) and **time period for server command check** (Fox 2.1 – can be set by terminal AVL device menu). This parameter can also be altered remotely. For Fox Basic, Fox Lite 3.0 and newer devices, this value is profile-dependent and (in the current FW version) relative to data package sending operation (meaning that command check is actually performed upon data upload if timeout condition has been met).

The *check-for-command* request is sent to the server IP through one of free IP ports (configured for data transmission) with the following format:

LSB **STX (0x02)** **Vehicle_ID** **ETX (0x03)** MSB












































Fox expects commands (if present) to be sent as response, in same manner: **STXcommand(s)ETX**.



























The command part of message is independent from transport channel (SMS or TCP). Fox devices recognize two types of commands:

- Remote executive commands – identified with **command mnemonic** + semicolon “:” + **[value]**. Note that *value* field is required only by some executive commands.
- Parameter alteration commands – identified with **parameter mnemonic** + semicolon “:” + **new value**

Multiple commands may be sent within single request by server. If that is the case, commands are separated with comma character “,”. Total command request length is limited to **160** characters.

Fox commands support matrix:

Command		Device				
Mnemonic	Description	Fox 2.1	FoxLite 1.1	FoxLite 2.0	FoxLite 3.0	FoxBasic 1.0
Legend:  Supported as remote command  Supported as parameter						
DEL	Delete all parameters					
DL	Delete data log					
DMP	Dump file to standard output (hex)					
DR	Start logged data transmission					
FD	Print debug info to standard output					
FFW	Restore factory firmware					
FW	Invoke firmware update OTA					
FR	Invoke firmware read & upload					
GFA	Geo-fence zone add					
GFD	Geo-fence zone delete					
GFR	Geo-fence zone read					
HIB	Hibernate					
P	Send position with debug info					
R	Send SMS formatted response					
RP	Read one or more parameters					
RS	Reset device					
RSS	Restore Settings from SIM					
MKF	Format file system					

MKG	Format geo-fence storage database					
MKL	Format data log storage database					
MKP	Format phonebook database					
MKI	Format iButton UIDs database					
IBA	Add iButton UIDs					
IBR	Remove iButton UIDs					
IBL	Read list of iButton UIDs					
PBA	Add phonebook entry					
PBR	Remove phonebook entry					
PBL	Read list of phonebook entries					
CGR	Reset CAN data values					
SLU	Upload debug log to FTP server					
<i>Parameter alteration commands</i>						
	<i>To be done. Look at the user manual...</i>					
						



Contact us

General	<p>RB "GENERAL EKONOMIK", d.o.o. Bulevar Despota Stefana 59a/1, 11000 Belgrade, Serbia</p> <p>web www.geneko.rs www.geneko.co.rs</p> <p>e-mail office@geneko.rs</p> <p>phone + 381 (11) 3340 591 + 381 (11) 3340 178</p> <p>fax + 381 (11) 3224 437</p>
Sales	sales@geneko.rs
Support	foxsupport@geneko.rs support@geneko.rs