



- **APPLICATION NOTE**



- **SUPPORTED *FSM* PARAMETERS**

- **STEPPIII / FOX / FOX-LT**

VERSION HISTORY:

This table provides a summary of the document revisions.

Version	Author	Changes	Created
		-	
1.0.0	F. Beqiri	- Initial version	09/12/2009

TABLE OF CONTENTS

1	ABOUT THIS DOCUMENT.....	5
1.1	<i>Supported FMS parameters.....</i>	<i>5</i>
1.2	<i>Connecting the AVL device to a vehicle FMS gateway?.....</i>	<i>8</i>
1.3	<i>Reading FMS parameters form the CAN-Bus and sending them via TCP to a remote server or storing inside the device.....</i>	<i>9</i>

CAUTIONS

Information furnished herein by FALCOM is believed to be accurate and reliable. However, no responsibility is assumed for its use. Please, read carefully the safety precautions.

If you have any technical questions regarding this document or the product described in it, please contact your vendor.

General information about FALCOM and its range of products are available at the following Internet address: <http://www.falcom.de/>

TRADEMARKS

Some mentioned products are registered trademarks of their respective companies.

COPYRIGHT

This document is copyrighted by **FALCOM WIRELESS COMMUNICATIONS GmbH** with all rights reserved. No part of this documentation may be produced in any form without the prior written permission of **FALCOM WIRELESS COMMUNICATIONS GmbH**.

FALCOM WIRELESS COMMUNICATIONS GMBH.

No patent liability is assumed with respect to the use of the information contained herein.

NOTE

Specifications and information given in this document are subject to change by FALCOM without notice.

1 ABOUT THIS DOCUMENT

This application note provides information how to connect a FALCOM device with CANBus option to an external CANBus and read the FMS parameters running on this bus. *The STEPIII device is used as an example in this application note.*

1.1 Supported FMS parameters

The Fleet Management Systems Interface (FMS) is a standard interface to vehicle data of commercial vehicles. The amount of data is dependent on the manufacturer and model of the vehicle and might be different. If some data are not available at the FMS interface they are read as not available (n/a).

Starting from the firmware version 2.6.4 releases and newer, the FALCOM devices with CANBus option support some of the CAN parameters defined by FMS standard. The following is a list of supported parameters that can be read on the FMS interface:

FMS Parameters (<i>dynamic entries</i>)	Name	Responses/Description
FMS.BREAK_SWITCH	Brake switch	Reports the current status of the break switch: 0 - Brake switch is off 1 - Brake switch is on err - Device error n/a - Value not available
J1939.PARK_BREAK_SWITCH	Parking brake switch	Reports the current status of the parking brake switch (additional - it is not in FMS-standard) 0 - Parking brake switch is off 1 - Parking brake switch is on err - Device error n/a - Value not available
FMS.SPEED_WB_KMPH	Vehicle speed (wheel based)	Reports the wheel based vehicle speed in km/h
FMS.SPEED_WB	Vehicle speed (wheel based)	Displays the wheel based vehicle speed in cm/s
FMS.CRUISE_CONTROL	Cruise control status	Reports the current status of the cruise control 0 - Cruise control is off 1 - Cruise control is on err - Device error n/a - Value not available
FMS.CLUTCH_SWITCH	Clutch switch	Reports the current status of the clutch switch 0 - Clutch switch is off 1 - Clutch switch is on err - Device error n/a - Value not available
FMS.PTO	PTO status	Reports the current status of the power take off governor. 00 - Off / disabled 01 - Hold 02 - Remote hold 03 - Standby 04 - Remote standby 05 - Set 06 - Decelerate/Coast 07 - Resume 08 - Accelerate 09 - Accelerator override 10 - Preprogrammed set speed 1 11 - Preprogrammed set speed 2 12 - Preprogrammed set speed 3 13 - Preprogrammed set speed 4 14 - Preprogrammed set speed 5 15 - Preprogrammed set speed 6 16 - Preprogrammed set speed 7 17 - Preprogrammed set speed 8

FMS Parameters (<i>dynamic entries</i>)	Name	Responses/Description
		18 – PTO set speed memory 1 19 – PTO set speed memory 2 31 – Value not available
FMS.ACCEL	Accelerator pedal position	Reports the current accelerator pedal position in percent (%).
J1939.FUEL_TRIP	Trip fuel used	Reports the used fuel in litre (l). – trip based not in FMS-standard
FMS.FUEL_TOTAL	Fuel used	Reports the total used fuel in litre (l).
FMS.FUEL	Fuel used	Displays the fuel level in % (percent).
J1939.FUEL_ECO_INST	Instant fuel eco	Reports the instant fuel consumption in litre/1000 kilometres (l/1000 km). - not in FMS-standard.
J1939.FUEL_ECO_AVRG	Average fuel eco	Reports the average fuel consumption in litre/1000 kilometres (l/1000 km). - not in FMS-standard.
FMS.ENGINE_SPEED	Engine speed	Reports the engine speed in rotations per minute (rpm).
FMS.VER_DIAG_SUPP	Diagnostics supported	Reports the diagnostics state. 0 – Diagnostics not supported 1 – Diagnostics supported err – Device error n/a – Value not available
FMS.VER_REQU_SUPP	Requests supported	Reports the requests state. 0 – Requests not supported 1 – Requests supported err – Device error n/a – Value not available
FMS.VERSION	Software version	Reports the FMS software version as a string.
FMS.VEHICLE_DIST	High resolution total vehicle distance	Reports the high resolution total vehicle distance in meters (m).
J1939.VEHICLE_DIST_TRIP	High resolution trip vehicle distance	Reports the high resolution total vehicle distance in meters (m). - not in FMS-standard.
FMS.MAINTANCE	Next regular maintenance	Reports the remaining distance to the next regular maintenance in kilometres (km).
FMS.TC_DRV1_STATE	Driver 1 working state	Reports the current Driver 1 working state: 0 - Rest 1 - Available 2 - Work 3 - Drive 6 - ERROR 7 - n/a
FMS.TC_DRV2_STATE	Driver 2 working state	Reports the current Driver 2 working state: 0 - Rest 1 - Available 2 - Work 3 - Drive 6 - ERROR 7 - n/a
FMS.TC_DRV1_TIME	Driver 1 time related states	Reports the current driver 1 time related state: 00 - Normal 01 - 15 min bef. 4½ h 02 - 4½ h reached 03 - 15 min bef. 9 h 04 - 9 h reached 05 - 15 min bef, 16 h 06 - 16 h reached 09 - Other 14 - ERROR 15 - n/a

FMS Parameters (<i>dynamic entries</i>)	Name	Responses/Description
FMS.TC_DRV2_TIME	Driver 2 time related states	Reports the current driver 2 time related state: 00 - Normal 01 - 15 min bef. 4½ h 02 - 4½ h reached 03 - 15 min bef. 9 h 04 - 9 h reached 05 - 15 min bef, 16 h 06 - 16 h reached 09 - Other 14 - ERROR 15 - n/a
FMS.TC_MOTION	Vehicle motion	Reports the current vehicle motion status: 0 – Not moving 1 – Moving err – Device error n/a – Value not available
FMS.TC_DRV1_CARD	Driver 1 card	Reports the current driver 1 card status: 0 – Not present 1 – Present err – Device error n/a – Value not available
FMS.TC_DRV2_CARD	Driver 2 card	Reports the current driver 2 card status: 0 – Not present 1 – Present err – Device error n/a – Value not available
FMS.TC_OVERSPEED	Vehicle overspeed	Reports whether or not the vehicle speed exceeds the speed limit registered in digital tachograph memory: 0 – Not exceeding 1 – Exceeding err – Device error n/a – Value not available
FMS.TC_EVENTS	System events	0 – No system events 1 – System events err – Device error n/a – Value not available
FMS.TC_HANDLING	Handling information	0 – No handling information present 1 – Handling information present err – Device error n/a – Value not available
FMS.TC_PERF	Tachograph performance	Reports the current tachograph performance status: 0 – Normal performance 1 – Performance analysis err – Device error n/a – Value not available
FMS.TC_DIR	Direction indicator	Reports the motion direction: 0 – Forward 1 – Reverse err – Device error n/a – Value not available
FMS.TC_SHAFT_SPEED	Tachograph output shaft speed	Reports the calculated output shaft speed in rotations per minute (<i>rpm</i>).
FMS.TC_SPEED_KMPH	Tachograph vehicle speed	Reports the calculated output shaft speed in rotations per minute (<i>rpm</i>).
FMS.TC_SPEED	Tachograph vehicle speed	Reports the calculated output shaft speed in rotations per minute (<i>rpm</i>).
FMS_ENGINE_TEMP	Engine coolant temperature	Reports the current engine coolant temperature in degree Celsius (°C).

Table 1: Supported FMS parameters

1.2 Connecting the AVL device to a vehicle FMS gateway?

The Bus-FMS-Standard interface is designed according the SAE J1939 Standard. Third parties are able to connect and to get these data from the truck bus system. For more details about the Bus-FMS-Standard, please visit the web site: <http://www.bus-fms-standard.com>. The connector shown on the figure below (Fig. 1) corresponds to the connector defined by Bus FMS Standard and not by all trucks bus system as it can be different from truck to truck. The location of the Bus FMS Standard connector is not standardized. Please contact the manufacturers to know the exact location of the Bus FMS connector in your vehicle.

To read the data of the internal bus defined in Bus-FMS-Standard you'll need:

- ✓ A FMS interface,
- ✓ The CAN bus must be terminated at both ends by a 120-ohm resistor to prevent signal reflections. Use an ohm meter to check wiring on the vehicle side. With the power off, verify 60-ohm across CAN_H and CAN_L (if only one resistor is installed you will read 120-ohm instead of 60-ohm).
- ✓ A FALCOM device with CANbus (e.g. STEPPIII-B1-CA) running with the firmware version 2.6.4 releases and later,
- ✓ A SIM card (supporting GPRS for transmitting data over TCP),
- ✓ Vehicle installation cable (refer to the document "AppNotes_Vehicle_mounting.pdf"),
- ✓ Operating voltage.

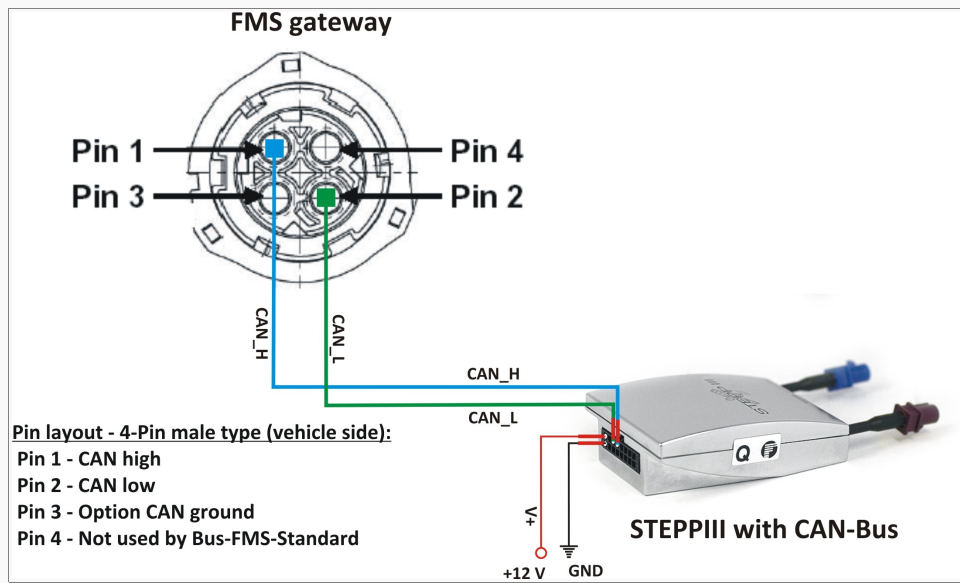


Figure 1: Connecting STEPPIII to the in-vehicle FMS gateway.

How to install a FALCOM AVL device in a vehicle, refer to the document "AppNotes_Vehicle_mounting.pdf".

The CAN lines on STEPPIII and FOX/-LT:

STEPPIII - on the 16pin connector:	
DI0 (Pin 10)	CAN_H line
DI1 (PIN 12)	CAN_L line
FOX/-LT - on the 8-pin connector:	
I/O2 (Pin 5 - Yellow)	CAN_L line
I/O3 (Pin 6 - Green)	CAN_H line

Table 1: Supported FMS parameters

1.3 Reading parameters form the FMS gateway and sending them to a remote server via TCP or storing inside the device

After connecting the FALCOM device to the vehicle FMS interface and applying power to the device you are able to get the data from the truck bus system either by recording these data inside the device or sending them to a remote server via SMS, TCP etc.. To do it, you have first to enable the CANBus and FMS interface inside the device, restart the AVL device and then you can get the data either on request or send them out periodically via alarm configuration.

The following examples show how to configure an AVL device to get some of the FMS parameters running on the CANBus data steam. Additional help for each example/command is also given.

Enable or disable FMS functionality:

```
$PFAL,Sys.Can.Enable,250K,RW
```

To use the FMS functionality, first you have to enable the CAN interface, set the baudrate to 250k baudrate (*the same as FMS on vehicle side*) and enter it into the ReadWrite mode. The RW (ReadWrite) setting is required **ONLY** when connecting an AVL device to the **FSM GATEWAY** and **NOT to the CAN Bus** (vehicle side). This is used to send an acknowledgement when the requested message is successful received.

```
$PFAL,Sys.Can.FMS.Enable // enables FMS interface
```

```
$PFAL,Sys.Can.FMS.Disable // disables FMS interface
```

These PFAL commands enable/disable FMS functionality on the AVL device after rebooting the AVL device.

```
$PFAL,Sys.Device.Reset
```

Execute this command to reset the AVL device and enable or disable FMS functionality.

Sending out FMS parameters values by timer or SMS text events:

```
$PFAL,CNF.Set,AL99=Sys.Device.eStart:Sys.Timer0.Start=cyclic,60000&Sys.Timer1.Start=cyclic,600000
```

This alarm starts two cyclic timers with 60 seconds and 60 minutes when the device starts up.

```
$PFAL,CNF.Set,AL98=SYS.Timer.e0:TCP.Client.Send,8,"Fuel:          &(FMS.FUEL_TOTAL);      Speed:
&(FMS.SPEED_WB_KMPH)"
```

This alarm transmits via TCP the total fuel used and current speed of the vehicle to a remote server whenever the Timer0 expires.

```
$PFAL,CNF.Set,AL97=GSM.SMS.eIncoming.Text="POS":GSM.SMS.Send,"&(SMSNumber)",8,"Fuel:
&(FMS.FUEL_TOTAL); Speed: &(FMS.SPEED_WB_KMPH)"
```

This alarm transmits back via SMS the total fuel used and current speed of the vehicle whenever the device receives an SMS with text "POS".

```
$PFAL,CNF.Set,AL97=SYS.Timer.e1:GSM.SMS.Send,"&(SMSNumber)",8,"Fuel: &(FMS.FUEL_TOTAL); Speed:
&(FMS.SPEED_WB_KMPH)"
```

This alarm transmits every 1 hour (when Timer1 expires) via SMS the total fuel used and current speed of the vehicle.

Getting FMS parameters values on request:

```
$PFAL,TCP.Client.Send,8,"Fuel: &(FMS.FUEL_TOTAL); Speed: &(FMS.SPEED_WB_KMPH)"
```

This PFAL command should be sent from a remote server. Once the AVL device receives this PFAL command it transmits back to that server the total fuel used and current speed of the vehicle.