

FMS iSNIFFER FD

- technical description -

Contents:

BASIC DESCRIPTION	2
WIRE CONNECTIONS	3
CONFIGURATION	3
COMMANDS	5
FIRMWARE UPDATE	8
LIST OF SUPPORTED VEHICLES	12
EXAMPLE CONFIGURATION AND DATA READING FROM VW GROUP ELECTRIC VEHICLES – ŠKODA ELROQ	13
EVENTS	14

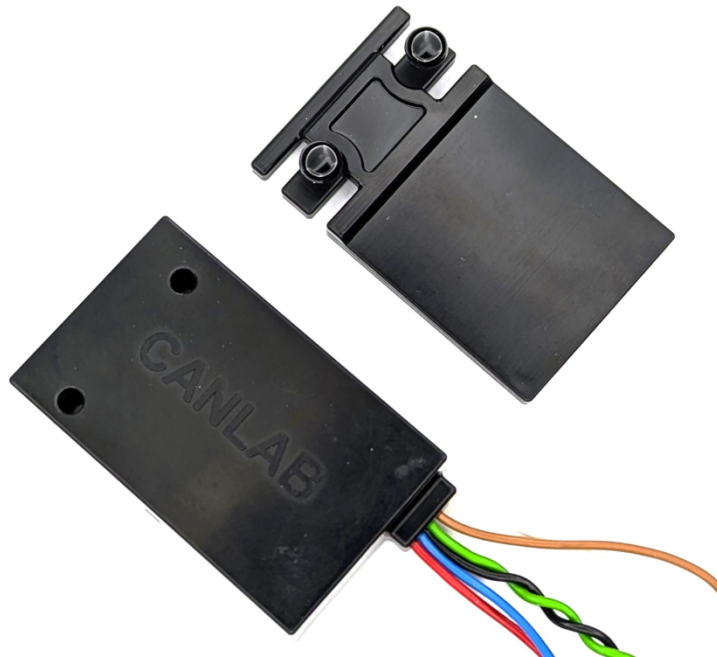
CANLAB team

Basic description

This is a variant of the CAN SNIFFER – a contactless, non-invasive (non-intrusive) CAN bus reader, where data output is provided via a serial line (UART). Vehicle data is read non-invasively from the CAN bus, processed and decoded by the MCU in the CAN sniffer, and can be read via the serial port. The data and command format is the same as for the FMS OEM CHIP. The FMS iSNIFFER additionally allows connection of, for example, the D8 output of a digital tachograph via an input with an optical isolator. Custom firmware can also include reading of other serial data, such as IBIS, an RFID reader, a float sensor with RS232, etc.

This solution combines a CAN sniffer for safe CAN bus reading with the CAR2COM FMS OEM CHIP V7. Combining these 2 devices into one allows safe non-invasive data reading and decoding at a very competitive price. The FMS iSNIFFER covers all settings and data reading in the same way as the current FMS OEM CHIP V7 (for example, statistics are not supported, tachograph downloads are not possible, error code reading via queries is not available), however it is fully sufficient for a large proportion of use cases. Command compatibility with the FMS OEM CHIP V7 allows easy switching between this and the full solution.

Both classic CAN and the new CAN FD are supported. The output serial line can be implemented as classic RS232 (positive/negative voltage) or in 5V or 3.3V variants.



Wire connections

Colour	Description
Red	Power supply 8–32V
Blue	GND
Black	Serial line – RX
Green	Serial line – TX
Brown	GND D8 *
White	D8 signal *
* not present in the economy version	

Configuration

Configuration is performed using similar commands as for the FMS OEM CHIP V6 and V7.

Character	Interface support		Description
	C	T	
A	Red		Accelerator pedal position.
B	Red		Brake pedal position.
C	Red		Average fuel consumption.
D	Red		Daily mileage / or trip mileage.
E	Red		Error codes
F	Red		Total fuel consumed value.
G	Red		Current and recommended gear
H	Red		Total engine hours.
I		Blue	Driver identification from tachograph card
J	Red		Fuel flow
K	Red	Blue	Exceeding driving time without a mandatory break according to the tachograph.
L	Red		Fuel tank level. Percentage for trucks, litres for VW.
M**	Red		Engine load
N	Red		Coolant temperature.
O	Red		Ad blue
P	Red		Current fuel consumption.
Q	Red		Bit states for doors, boot, etc. 16-bit number expressed in hexadecimal. Bit 0 – closed. From FW version 6.2, an additional 32-bit word with bit flags follows. Format:Qabcd-klmnopqr
R	Red	Blue	Engine RPM
S	Red	Blue	Vehicle speed
T	Red	Blue	Total mileage.
U*	Red		Tyre pressure
V	Red		Distance in kilometres to the next service.
W*	Red		Axle load.
Y	Red		Fuel consumed during trip
Z	Red	Blue	Tachograph speed

Data assignment – extended CAN data

Character	Description
A	Vehicle acceleration – longitudinal and lateral
B	Range
C	Clutch
D	

F	
E	Total_fuel_used_gaseous
G	Gross weight
H	Vehicle charging flag (EV)
I	Trailer axle load sum
J	
K	
L	
M	
N	
O	
P	PTO
Q	HVESS
R	
S	Steering angle
T	Cruise control
U	
V	Trailer weightp
W	Cargo weight
Y	
Z	Electric range (EV)

Example configuration:

```
$PCAN,C,SET,C8,F2,EXT,LISO,P0+1,T0,#RTL##VW###,*17
```

C8 – same as for the FMS OEM CHIP, setting the CAN speed for either classic frames or the arbitration phase of CAN FD frames. Supported values: C7-125kbit, C8-250kbit, C9-500kbit, C10-1Mbit, C11-666.67kbit.

F2 – as the iSniffer can also work with CAN FD frames, the data phase speed must also be configured when the frame is transmitted with the baud rate switch flag. Supported values: F0-500kbit, F1-1Mbit, F2-2Mbit, F3-4Mbit, F4-8Mbit.

If EXT is specified, the CAN message receive buffers are optimised for receiving 29-bit identifiers.

LISO – not applicable for the iSniffer; vehicle connection is safe.

P0+1 – the configured data is periodically sent to the serial line every 1 second (0 minutes, 1 second)

T0 – vehicle code/type, 0 – vehicles using the SAE J1939 protocol

RTL – value codes to be periodically transmitted in the first sentence \$PCAN,C – CAN data.

VW – kW – value codes to be periodically transmitted in the first sentence \$PCAN,E – Extended CAN data.

Periodic CAN data is generated separately for CAN data and Extended CAN data. The reason is to keep these strings to a reasonable length when reading large amounts of data, and to simplify parsing. The generated data looks like this, for example:

```
$PCAN,C,GET,R1242L48.0:0.0T231564.0,*59
```

```
$PCAN,E,GET,V5488W7844,*00
```

If periodic data needs to be disabled, the command is as follows:

\$PCAN,C,SET,C8,EXT,LISO,P0+0,T0,###,*26

Period of 0 minutes and 0 seconds. Between between # and ## no data characters are specified

In addition to reading data from the CAN bus, it is also possible to read data from the D8 pin of a digital tachograph. This example command sets a request for periodic generation of data I (driver identification), K (driving time exceeded) and S (tachograph speed) from the tachograph. Data is generated every 5 seconds (P0+5) and this is a VDO tachograph (T0).

\$PCAN,T,SET,T0,P0+5,#SIK##,*62

Example of data that is then automatically generated periodically:

\$PCAN,T,GET,S0K00-00-00I@00000000004LZ400@0000000002FK5001@,*22

Commands

Each command begins with the header \$\$PCAN, followed by a letter identifying the interface or function for which the command is intended. The entire string is then terminated with a checksum using an XOR operation. The checksum always consists of 2 characters and the value is expressed in hexadecimal. More information about the checksum can be found in the FMS OEM CHIP / CAR2COM manual.

After power-on and expiry of the bootloader timeout, the iSniffer automatically sends firmware version information in the following format:

\$PCAN,C,VER,2.17,FMS_CHIP_ISNIFFER_FD,CANLABsro,????????,*67

Firmware version is 2.17. This is an ISNIFFER with CAN FD support.

The question marks indicate that no identification string has been set. This can be configured in the iSniffer using the command:

\$PCAN,C,CID,6BZ5590,*36

This sets the identification to 6BZ5590, in this case the vehicle registration plate.

Then, on a firmware version query or after power-on, the message looks like this:

\$PCAN,C,VER,2.17,FMS_CHIP_ISNIFFER_FD,CANLABsro,6BZ5590,*7F

The firmware version and identification can also be queried as follows:

\$PCAN,C,VER,*32

PTO information source:

\$PCAN,C,PTO,a,b,*checksum

a: 0=0xFDA4, 1=0xFEf1

b: for compatibility with FMS OEM CHIP only

Ignoring zero fuel level:

\$PCAN,C,IZF,?,*35

Enable – tank spikes to zero value are ignored

\$PCAN,C,IZF,1,*3B

Enable – tank spikes to zero value are ignored when RPM is non-zero

\$PCAN,C,IZF,2,*38

Disable zero ignoring

\$PCAN,C,IZF,0,*3A

For compatibility with FMS OEM CHIP, calculation of fuel consumed during the trip

Query settings

\$PCAN,C,STT,F?,*75

Response, calculation is performed from high resolution total fuel used:

\$PCAN,C,STT,F?0,*45

Switch to calculation from high resolution total fuel used:

\$PCAN,C,STT,F0,*7A

Switch to calculation from fuel rate:

\$PCAN,C,STT,F1,*7B

This command performs a software reset of the processor:

\$PCAN,C,PWR,RST,*5F

For each vehicle profile, an operating variable is selected upon receipt of which the indicator LED flashes. This is usually vehicle speed. At the same time, an internal counter is reset indicating how old this value is. If data reception is not occurring or stops, this counter begins to increment every second. The maximum value is 65535

\$PCAN,C,OLD,*34

Response, last data received 7 seconds ago:

\$PCAN,C,OLD,7,*2F

Query for DPF clogging in heavy goods vehicles:

\$PCAN,C,DPF,*21

The response contains two parameters: the first is soot clogging, the second is ash clogging. In this case, the second parameter is not read.

\$PCAN,C,DPF,22.8:X,*79

Diagnostics of received CAN data:

\$PCAN,C,STA,*35

Response:

\$PCAN,C,STA,a,b,c,d,e,f,g,h,i,*<checksum>

- a- vadfaulty CAN frame
- b- received classic 11-bit
- c- received classic 29-bit
- d- received FD 11-bit
- e- received FD 29-bit

- f- received FD 11-bit with DLC≤8 queued as classic
- g- received FD 29-bit with DLC≤8 queued as classic
- h- lost classic and FD 11-bit due to full FIFO
- i- lost classic and FD 29-bit due to full FIFO

In some cases, the CAN bus carries 2 total mileage values with different figures. The discrepancy is usually caused by a mismatch between the engine ECU and the tachograph data at the time of vehicle sale / tachograph installation, ECU replacement and similar events.

It is possible to configure filtering by source address:

Data from tachograph (SA 238 – tachografu (SA 238 – 0xEE):

\$PCAN,C,SFL,T,1,238,*5A

Engine ECU:

\$PCAN,C,SFL,T,1,0,*73

Disable filtering:

\$PCAN,C,SFL,T,0,*4E

Query settings:

\$PCAN,C,SFL,T,?,*41

Enable function that accepts the odometer reading only when it is increasing. Intended for J1939-based profiles for ECUs that round the distance on shutdown.

\$PCAN,C,SFL,T,M,1,*41

Disable function:

\$PCAN,C,SFL,T,M,0,*41

Query:

\$PCAN,C,SFL,T,M,?,*41

Also for J1939-based profiles, some vehicles may provide an odometer reading in low resolution. Activating this function ignores this and uses only the high resolution value.

\$PCAN,C,SFL,T,H,1,*41

Filtering of total fuel consumed – similar to the previous case, multiple messages with different resolutions may be present on the CAN bus.

\$PCAN,C,SFL,T,F,<value>,*41

Value: 0 – low i high resolution

1 – low resolution only

2 – high resolution only

Similar to total mileage filtering, but this time for engine hours – the ECU source address is configured:

\$PCAN,C,SFL,H,1,23,*5A

Engine ECU:

\$PCAN,C,SFL,H,1,0,*73

Disable filtering:

\$PCAN,C,SFL,H,0,*4E

Reset FMS iSniffer settings to factory defaults:

\$PCAN,C,RESET,0,*3A

The FMS iSniffer also allows reading of certain error codes that are broadcast on the CAN bus without the need to send queries for them. Query for the number of stored error codes:

\$PCAN,C,DTR,S,*4E

Response, 3 errors are stored:

\$PCAN,C,DTR,S,3,*51

Query for the first error:

\$PCAN,C,DTR,0,*2D

Response:

\$PCAN,C,DTR,DM1 174 F3 O1 E0,*69

DM1 type error, SPN 174, FMI 3, occurrence counter 1, error reported by ECU with SA E0. SPN, FMI and OC are in decimal; SA is in hexadecimal.

Query for the second stored error:

\$PCAN,C,DTR,1,*2C

Response:

\$PCAN,C,DTR,BOB M0116,*39

Specific error in BOBCAT format, code M0116:

Query and response for the third stored error:

\$PCAN,C,DTR,2,*2F

\$PCAN,C,DTR,DM1 172 F3 O1 E0,*6F

Query for the fourth error, response that the buffer is empty:

\$PCAN,C,DTR,3,*2E

\$PCAN,C,DTR,3,EMPTY,*57

Vehicle data does not have to be read only via periodic data but also via queries. Example query for speed (S) and RPM (R):

\$PCAN,C,GET,#SR##,*2B

Response, speed 0, RPM 1242.

\$PCAN,C,GET,S0R1242,*3D

Firmware update

To update the firmware, the CANLAB terminal application is required, and the converter must be connected to a PC via the serial line – typically using a USB-to-COM port adapter, depending on the FMS iSniffer variant (RS232 / 5V / 3.3V levels). We recommend using converters / cables based on FTDI chips.

The current version of the terminal is available here:

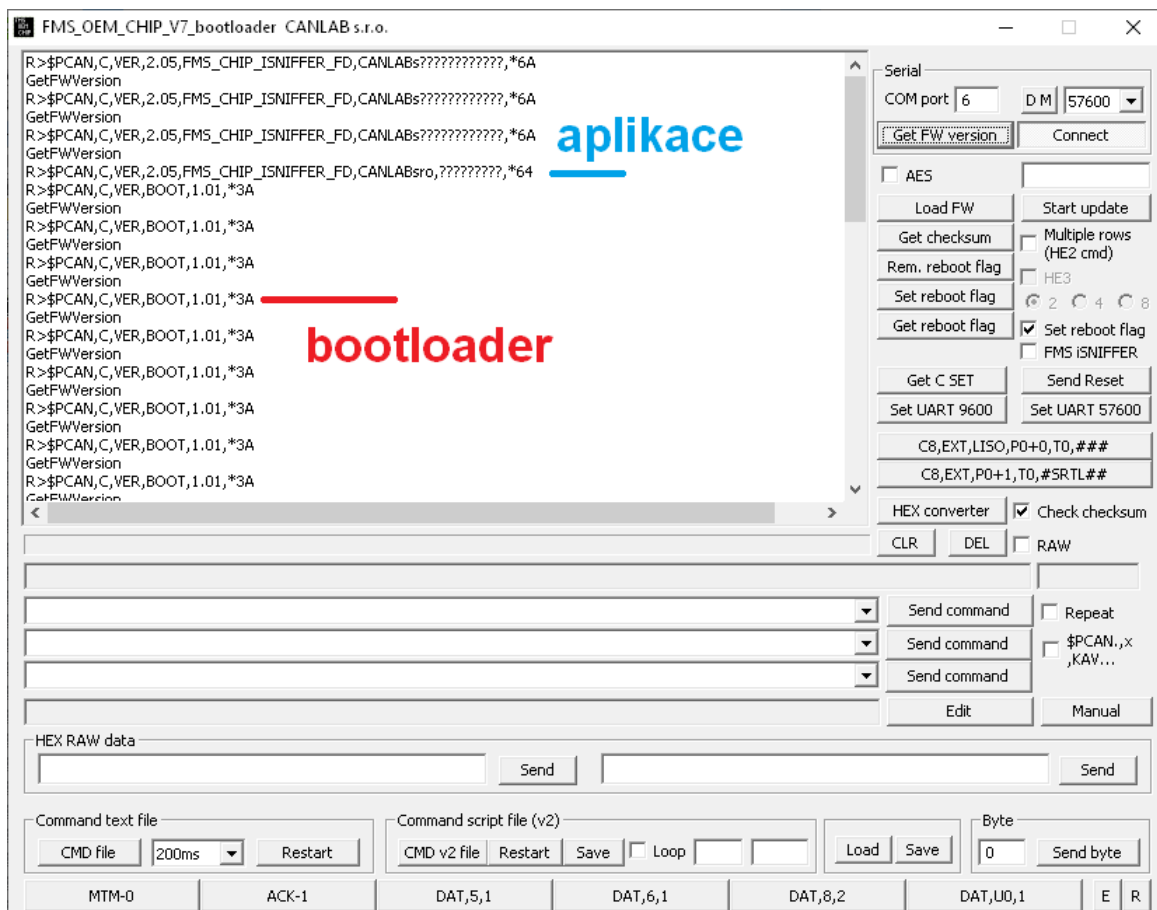
https://www.canlab.cz/sites/default/files/FMS_OEM_CHIP_V7_PC_bootloader.zip

It is advisable to use the latest version of the terminal.

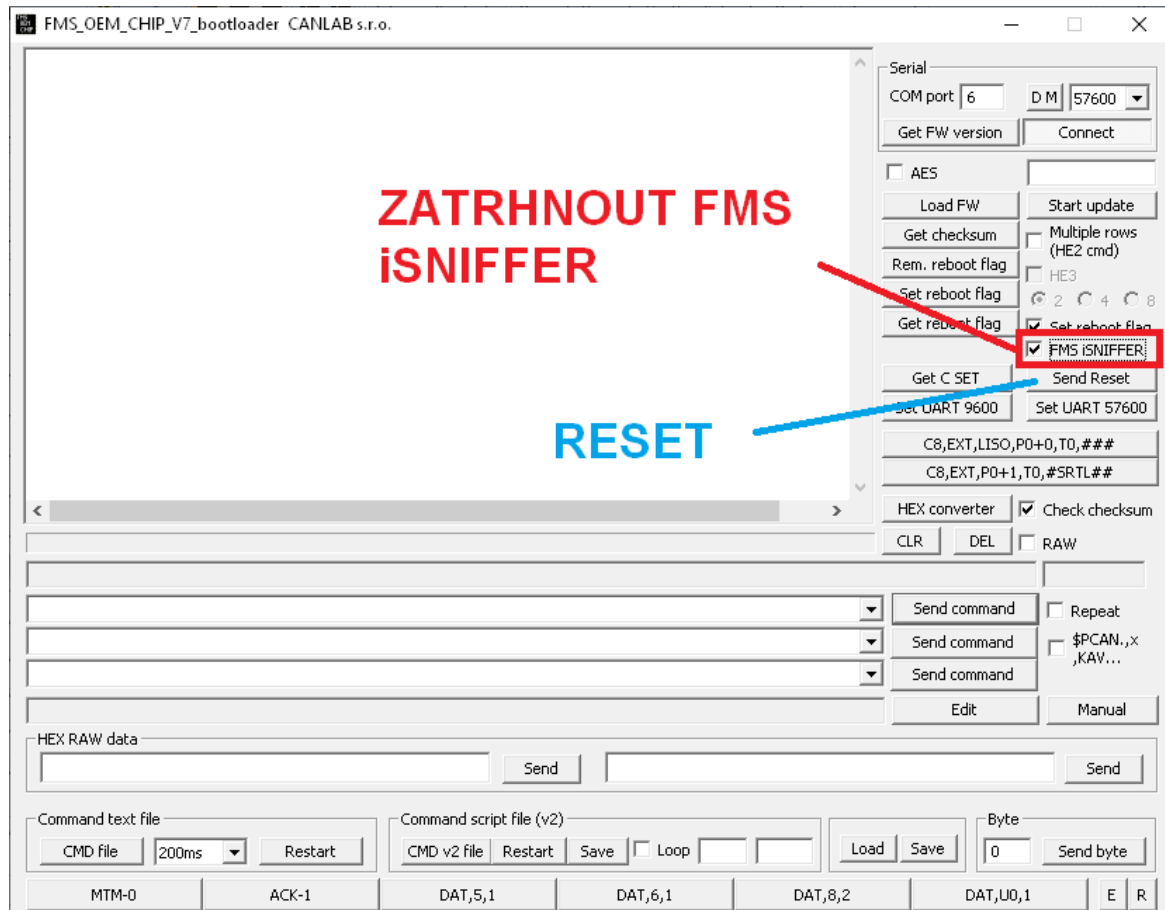
Note, the following applies:

- the bootloader is fixed at 57600 baud
- booting is possible for 10 seconds after power is connected
- the application firmware has a fixed speed of 9600 (as per customer configuration)

Uploading firmware to the FMS iSniffer uses a different protocol than that used for the FMS chip/CAR2COM (different MCU architecture). Whereas the normal FMS OEM chip uploads the FW to external FLASH at runtime, the iSniffer has no such memory. Therefore, uploading is only possible for approximately 10 seconds after connecting power or after a command-triggered restart. This can also be seen in the image below. After power-on, several FW version queries are visible. Whilst the iSniffer is in the bootloader, it returns a different response – the bootloader version – compared to after switching to the application.



To upload, it is therefore necessary to start the terminal, first tick the option indicating that this is an FMS iSNIFFER (see the second image), and set the speed to 57600.



Then load the HEX file with the firmware (Load FW). Do not use the Multiple rows option etc.; the iSniffer does not support this.

It is then recommended to reset the iSniffer by cycling the power, so that it enters bootloader mode for 10 seconds (or use the Send Reset button if the serial line speed is correct, then change the speed to 57600), and to start the FW upload (Start update) within 10 seconds.

Once the upload is complete, switch the speed to 9600 for the application firmware to verify communication, and confirm that the firmware has been updated by querying the firmware version.

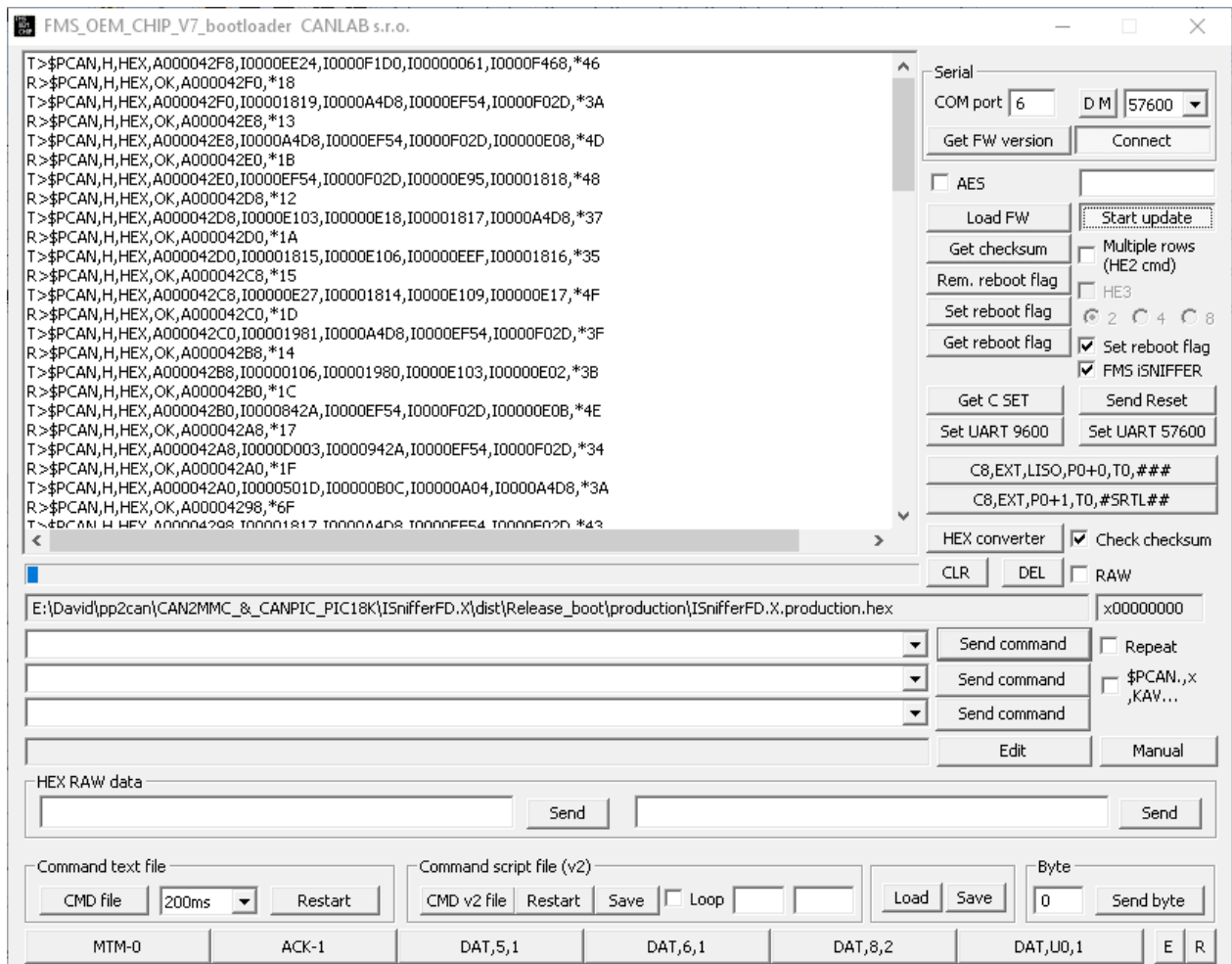
Procedure:

- Start the terminal
- Set 57000
- Connect
- Select FMS ISNIFFER
- Load FW
- Power reset
- Wait 1 second
- Start update
- After upload, switch to 9600
- Verify communication with the application by querying the firmware version

If immediately programming another FMS iSniffer unit without restarting the terminal:

- Disable Start update

- Set 57000
- Load FW
- Power reset
- Wait 1 second
- Start update
- After upload, switch to 9600
- Verify communication with the application



List of supported vehicles

The profile name is derived from the vehicle on which the data was measured. In practice, the profile may also work on other models from the same manufacturer, or on models from other manufacturers that use the same components / platform.

Profile name / vehicle	Code T
Heavy goods vehicles SAE J1939	0..14
Volvo Truck from 2013	7
J1939, trip fuel calculated from fuel rate	10
Mercedes truck E5 and older	15
Volvo Truck 2021	570
VW older, pre-MQB (Fabia I and similar)	16
VW MQB	192..194
VW Egolf	195,197,198
VW MQB Radio	196
VW MQB CNG	199
VW MQN LPG	200
VW MQB Hybrid	201
VW MQB ID3	202
VW MEB	205,206
Toyota	96,97,98
Toyota Hybrid	99
Peugeot	208,209
Peugeot	224,225,227,228
PSA	229
Toyota Hilux 2021	100
Older Hyundai and Kia	320
Hyundai and Kia from 2016	321
Hyundai Ioniq 2019	322
Hyundai KOna	323
Kia EV6	324
Hyundai Tuscon 2023	325
Hyundai Tuscon 2025	326
Mercedes Vito 2007	55
Mercedes Sprinter older	48,49
Mercedes Sprinter2018	53
Citroen Jumper	147,148
Opel Astra J	80
Opel Movano Vivaro	81,82,83.289,291
Renault Megane	288
Renault Escape Talisman	290
Renault Kangoo ZE	292
Fiat Ducato	145,146,150,
Fiat Doblo 50kb	149
Fiat Fullback	151
Fiat Tipo	152
Fiat Ducato 50kb	153,154
Fiat Ducato 2022	155

Renault Master	240,241,242,243
Ford Mondeo	128
Ford Tranzit 2015	134
Ford Tranzit Turneo	129,133
Ford CMax	130
Ford SMax	131
Ford Fusion	132
Ford Ranger	135
Ford Turneo Connect	136
Ford Tranzit 2017	137,138,141,142,143
Ford 2011	139
Ford Focus 2019	140,144
Linde H50 Evo	2000
Still RX 70_50	2100
Takeuschi	520
Bobcat	525
Faresin 2021	534
Kubota 2023	535
Kubota 2023 5t	536
Karcher	537
LGMGP AR14J	2200
T26JH	2201
AR24J	2202
T20J	2203

Example configuration and data reading from VW Group electric vehicles – Škoda Elroq

The VW Group MEB platform uses CAN FD for communication. The FMS iSniffer is compatible with the CAN FD protocol. However, the communication speed must be configured for both the arbitration phase and the data phase. The base speed during arbitration is 500kbit/s (C9), and the data phase speed is 2Mbit (F2).

The profile for MEB platform vehicles is 205 or 206. They differ in how battery energy is read. Connect to the orange-brown / pink wires.

```
$PCAN,C,SET,C9,F2,P0+1,T205,#LT##ZLK###,*4F
```

With this configuration, we specify that the PCAN,C sentence should send data L (tank level – battery state of charge for an electric vehicle) and T (total kilometres). The PCAN,E sentence then sends K (energy consumed), L (recuperated energy) and Z (electric range), along with additional information about maximum battery capacity and current battery capacity. These two values are not included in the tank/battery level, as the tank level has multiple combinations (two tanks, auxiliary tank with external float sensor when using CAR2COM) and parsing would be complicated. For this reason, the values were placed under the letter Z.

The FMS iSniffer then sends the following data every second:

\$PCAN,C,GET,L68.0T7096.0,*17

\$PCAN,E,GET,K29844490L8411380Z319:193500000:281664000,*51

PCAN,C sentence:

L68.0 – battery state of charge is 68 per cent

T7096.0 – odometer reading is 7096.0 km

PCAN,E sentence:

K29844490

Energy consumed since start: 29,844,490 Ws (watt-seconds)

$29844490 \text{ Ws} / 3600 = 8290 \text{ Wh} / 1000 = 8.290 \text{ kWh}$

L8411380

Energy recovered through recuperation since start: 8,411,380 Ws

$8,411,380 \text{ Ws} \div 3600 = 2,336 \text{ Wh} \div 1000 = 2.336 \text{ kWh}$

Z319:193500000:281664000 – range is 319 km, current battery energy is 193,500,000 Ws, and maximum capacity is 281,664,000 Ws.

Conversion to kWh:

$193500000 \text{ Ws} = 53.75 \text{ kWh}$

$281,664,000 \text{ Ws} = 78.24 \text{ kWh}$

Start of journey:

\$PCAN,E,GET,K690L0Z366:214830000:281664000,*6D

End of journey:

\$PCAN,E,GET,K29844490L8411380Z319:193500000:281664000,*51

From sentence E, letter Z:

Battery at start:

$214,830,000 \text{ Ws} = 59.675 \text{ kWh}$

Battery at end:

$193500000 \text{ Ws} = 53.75 \text{ kWh}$

Consumed: 5.925 kWh

From sentence E, letters K and L

Consumed: 8.290 kWh

Recuperated: 2.336 kWh

Total consumption – recuperation = battery decrease

$8.290 - 2.336 = 5.954 \text{ kWh}$, which approximately matches the battery decrease of 5.925 kWh

Events

From firmware version 2.32, basic event support has been added to the firmware – RPM and speed.

An event means exceeding a set value for a duration longer than the configured threshold. The function can be used, for example, for monitoring how vehicles or machines are used in rental companies or by employees, without the need to periodically read and check data.

Example: RPM exceeding 4000 rpm for more than 5 seconds

Configuration:

\$PCAN,C,EVN,R,V,4000,*02 - V – hodnota, 4000rpm

\$PCAN,C,EVN,R,T,5,*31 - T – doba, 5s

Query – the response has the same format as the configuration:

\$PCAN,C,EVN,R,V,?,*39

\$PCAN,C,EVN,R,T,?,*3B

If the RPM is continuously exceeded for 5 seconds, the FMS iSniffer sends an event as follows:

\$PCAN,C,EVN,RPM,*4D

Event generation can be cancelled by setting the event duration to 0.

The maximum configurable event duration is 254 seconds.

If, after sending the event, the RPM does not drop below the set value, the event is generated again after a further 5 seconds.

For speed-exceeded events, the following commands are used:

\$PCAN,C,EVN,S,V,<value>,*<checksum>

\$PCAN,C,EVN,S,V, <value>,*<checksum>

The generated event then has the following format:

\$PCAN,C,EVN,SPEED,*45