Description of the CAN-SNOOPER
for display terminals with integrated CAN-LOGGER
by MKT Systemtechnik
Version 1.3

CAN-SNOOPER

Dokument-Nr: 85118
Original: C:\CBProj\CanLoggerUtility\DOKU\art85118_CAN_Snooper_Description.odt
Online: www.mkt-sys.de/MKT-CD/handbuecher/art85118_CAN_Snooper_Description.pdf

This document is available in english and german language.
Dieses Dokument ist in englischer und deutscher Sprache verfügbar.

Contents

Revision History

Revision number	Date (ISO8601)	Author	Notes, modifications
V1.0	2008-11-17	W.Büscher	Creation (first in englisch, german translation will follow)
V1.1	2009-09-08	W.Büscher	Various additions for devices with 32-bit CPU
V1.2	2011-02-17	W.Büscher	"Max-period detector" for cyclic messages, and error frames displayed in the message list.
V 1.3	2017-08-31	W. Büscher	Updated description of 'signal decoders' / CAN Snooper can access the imported signal-database *from the display application* now

Preface

This document describes the operation of the CAN-SNOOPER, which is implemented as an additional function in several display terminals by MKT Systemtechnik, which have an integrated CAN-LOGGER, too.

The CAN snooper is a simple CAN analyzer on the CAN message level. The intended use is to monitor traffic in an "unknown" CAN network, with the possibility to *decode* or *highlight* certain signals in certain messages on the terminal screen without any premises like the availability of a CAN database file (but an imported CAN database helps a lot - see 'Signal Decoders')

In devices with 32-bit CPU (MKT-View III / IV), the CAN snooper can additionally record received CAN messages in a disk file (at moderate speed, in contrast to the CAN-Logger), and can send those recorded messages to the CAN-bus, or replay them internally (for the display application). This often helps for debugging, but can also be used for presentations when no 'real' CAN network is available: In that case, just record some "live data" with the Snooper, and play them back in an endless loop. No PC is required for this.

Up-to-date information can be found in the online help system (HTML format) of the terminal programming tool .

1 First steps

This chapter only describes a subset of the CAN Snooper's capabilities. Especially in the MKT-View II (available since 01 / 2009), there may be additional features which are not described here. Anyway, try to follow this chapter step-by-step to get a first impression about the CAN SNOOPER's possibilities. Advanced users will find a few hints and notes (indented and in smaller characters), which can be skipped for a quick start.

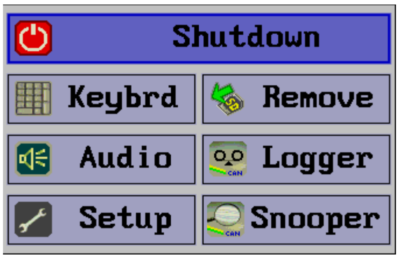
1.1 Invocation of the CAN Snooper

1.1.1 MKT-View I (and similar devices with 16-Bit-CPU)

First enter the system menu (press and hold F2+F3 simultaneously). If the terminal firmware contains a CAN-Snooper, you will find an entry for it in the system menu ("CAN Snooper"). Select the menu item with the rotary knob, then press ENTER (or the rotary knob).

1.1.2 MKT-View II (and similar devices with 32-Bit-CPU)

The CAN-Snooper can be invoked similar as above: Select the menu item with the rotary knob, cursor keys, or touchscreen; then press ENTER, or the rotary knob, or touch the menu item a second time.
In addition, the MKT-View II (and similar devices with a touchscreen) can also invoke the CAN Snooper through a graphic button on the "Shutdown"-screen. To enter the shutdown screen, press and hold the power key (usually F1) until the following window appears :



Tap on the button are labelled "Snooper", to start the CAN snooper.

Notes:

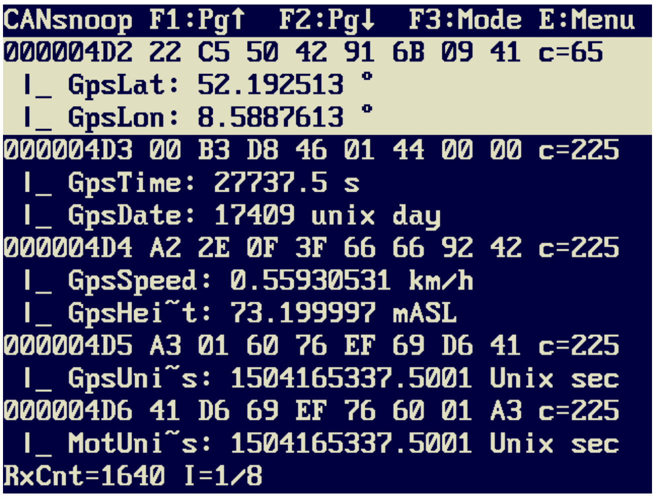
- If you don't find the item "CAN snooper mode" in the terminal's system menu, the CAN snooper is not supported by the firmware.
- The CAN snooper function may be supported by the firmware but not unlocked yet. To unlock a firmware function please contact the manufacturer for details.
- The CAN snooper display works best if the display runs in landscape mode. Using it in portrait mode is possible, but requires some tedious horizontal scrolling.

- The CAN snooper can also be invoked from the user-defined application (the "UPT application"), using the interpreter command snooper.run .

- The CAN-Snooper settings can be saved in a configuration file (SNOOPER.CFG), but only for devices with a 32-bit CPU (for example, "MKT-View II"). To save the snooper configuration permanently, select 'Other Functions' .. 'Save Snooper-Config' in the snooper menu. The config will be loaded automatically, next time you invoke the CAN snooper.

1.2 Message List

The received messages are displayed in hexadecimal form, sorted by identifier.



```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Mode E:Menu
000004D2 22 C5 50 42 91 6B 09 41 c=65
 |_ GpsLat: 52.192513 °
 |_ GpsLon: 8.5887613 °
000004D3 00 B3 D8 46 01 44 00 00 c=225
 |_ GpsTime: 27737.5 s
 |_ GpsDate: 17409 unix day
000004D4 A2 2E 0F 3F 66 66 92 42 c=225
 |_ GpsSpeed: 0.55930531 km/h
 |_ GpsHei~t: 73.199997 mASL
000004D5 A3 01 60 76 EF 69 D6 41 c=225
 |_ GpsUni~s: 1504165337.5001 Unix sec
000004D6 41 D6 69 EF 76 60 01 A3 c=225
 |_ MotUni~s: 1504165337.5001 Unix sec
RxCnt=1640 I=1/8
```

(combined display of 'raw' CAN frames
and a few decoded signals contained in them)

The leftmost column shows the CAN identifiers, usually in hexadecimal format. The most significant bit(s) show the bus number and the 11/29-bit-flag:

- 0x00000000 (bits 31..29 clear) indicates its a normal 11-bit-ID received from the first CAN interface (bus 1)
- 0x20000000 (bit 29 set) indicates its a 29-bit-identifier, also called "extended message frame"
- 0x40000000 (bit 30 set) indicates the message has been received with the 2nd CAN interface (bus 2)
- 0x80000000 (bit 31 set) indicates a message from the 3rd CAN interface (CAN3)
- 0xC0000000 (bits 31 and 30 set) indicates a message from the 4th CAN interface (CAN4)

Note: "CAN3" and "CAN4" are available when an external CAN adapter is connected, which provides two additional CAN ports (in addition to the device's two 'local' CAN ports). The external CAN interface is connected via Ethernet (CAN-via-UDP); details about CAN-via-UDP in document #85140.

After the message ID, up to 8 data bytes from the last received CAN message are shown in hexadecimal form (decimal won't fit on the small screen!). For future devices with CAN FD, up to 64 bytes may be possible (using multiple text lines).

If signal decoders exist for a received CAN message (see next chapter), the CAN message data cannot only be displayed in raw hexadecimal form, but also as scaled (decoded) signal values in an operator friendly form (along with signal name, signal value, and physical unit). Since 08/2017, the CAN-snooper can access the CAN signal database in the display application (if those signal definitions have not been deleted in the programming tool). Such signal definitions were traditionally imported from a *.dbc file ('database for CAN'), using the appropriate programming tool for your device. An extract from the imported database is stored in the device's onboard Flash memory, so you don't need a memory card for this purpose. For a combined display with both 'raw AN message' and 'signal values decoded from that message', open the CAN snooper menu, enter 'Display Options', and set 'View' to 'Both, combined' (instead of 'Raw CAN frames'). More about the menu and signal decoders will follow in the next chapters.

Certain devices (like MKT-View II) support the detection of CAN error frames. Error frames are displayed and counted like ordinary CAN messages, but with the text "ERROR FRAME" instead of the CAN-ID and data field. For experts: A CAN error frame is a sequence of at least six dominant bits, indicating a problem with the CAN bus network.
To avoid 'flooding' the CAN receive buffer with error frames, a maximum of 100 ***error frames per second*** will be buffered (for the display). If the error frame rate exceeds this limit, additional errors will be counted but not listed/logged.

At the right side of the screen, some additional info can be displayed. By default, only the frame counters for all individual IDs are displayed. A one-letter-token indicate which "additional info" is displayed at the moment. Possible tokens are:

- c= Counter. Every received CAN ID has its own counter, so there may be up to 2048(?) different counters.
- p= Period. This is the time interval between the last two received messages with the same CAN-ID, displayed in milliseconds (or seconds with "s" appended, if the value gets too large)
- P= Maximum Period. Only for devices with firmware compiled 2011-02-15 or later. This is the maximum measured period between any two messages with the same CAN-ID, displayed in milliseconds, since the message list was last cleared.
  Intended to watch for 'missing frames' for periodically transmitted messages.
- f= Frequency or Frame rate. This is the count of messages per second.
- t= Timestamp. Because the screen is too narrow, you may have to scroll horizontally to see the complete timestamp.

You can select which of these 'additional infos' shall be displayed via the CAN Snooper's menu (see next chapter).

The selection bar can be moved through the message list with the cursor up/down keys, or -if the terminal has no cursor keys- with the rotary button. The selection bar will remain on the same CAN ID even if "new" messages are merged into the sorted list.

The upper line shows the function key assignments. The function keys can be used to scroll the message list rapidly ("page up", "page down"), to mark or hide certain messages, and to scroll the display horizontally though we don't recomment this. The set of function key assignments can be changed with a function key itself (usually F3), or through the menu. Some abbreviations for the key functions are:

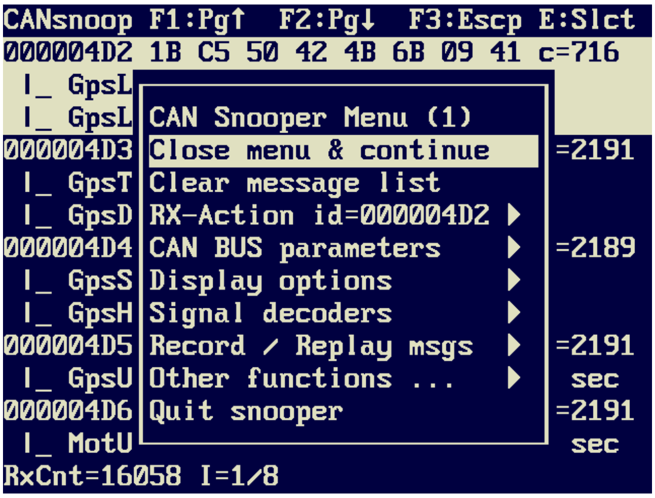
- PgUp: Scroll the list one page up
- PgDw: Scroll the list one page down

- Mode: Switch to the next 'mode', or set or key functions
- Menu: Enter the CAN Snooper's main menu
- Dump: Dumps the complete message buffer into a text file (not just the screen)
- Mark: Mark the currently selected message (because it's something special)
- Hide: Hide the currently selected message (because I don't want to see it any longer)
- Slct: (select) Do something else with the currently marked item, depends on what the item is

The Snooper is entirely controlled from a menu on the terminal screen. Use the function keys and the rotary button to navigate through the menu and to modify parameters if necessary. The CAN Snooper's menu structure is explained in the next chapter.

1.3 Snooper Main Menu

To open the CAN Snooper's main menu, press the ENTER key (or press the rotary button down, which is the same throughout this document). The currently selected message will be scrolled to the top, and a popup menu similar to this will open:



```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct
000004D2 1B C5 50 42 4B 6B 09 41 c=716
  |_ GpsL
  |_ GpsL CAN Snooper Menu (1)
000004D3 Close menu & continue    =2191
  |_ GpsT Clear message list
  |_ GpsD RX-Action id=000004D2 ▶
000004D4 CAN BUS parameters     ▶ =2189
  |_ GpsS Display options        ▶
  |_ GpsH Signal decoders        ▶
000004D5 Record / Replay msgs   ▶ =2191
  |_ GpsU Other functions ...    ▶  sec
000004D6 Quit snooper             =2191
  |_ MotU                           sec
RxCnt=16058 I=1/8
```

(screenshot of CAN Snooper main menu, beta version)

To navigate through the menu, use the cursor up/down keys. If your terminal has no cursor keys, turn the rotary button.

To open one of the submenus, or modify a numeric parameter in a menu, press the enter key.

The functions in the main menu are:

- Close menu window
  Closes the snooper's popup menu to have a full-screen view of the received messages again
- Clear message list
  Erases the internal list of received messages, including all message counters etc.
  Also prepares
- Define an RX-action
  Tell the snooper "what to do when a message with a certain ID has been received". This may be an acoustic signal, etc.
- CAN bus parameters
  Lets you modify the CAN baudrate and -possibly- other CAN parameters in future.
  CAUTION ! The standard terminal is always an ACTIVE CAN NODE. If you select the wrong baudrate, the CAN controller sends thousands of error frames which may cause a catastrophy if you try this in a vehicle at 200km/h ! There may be a special hardware with a disabled CAN output driver which avoids this problem. Unfortunately we cannot turn off the CAN output driver by software in older devices - see hardware specification.
- Display options
  Select what shall be displayed besides the hexadecimal message data, and show/hide certain unwanted messages.

- Signal decoders
  Explained in another chapter
- Other functions
  Less frequently used functions like the file counter for CAN message dumps.
- Quit snooper
  Exit snooper mode and resume normal operation as "programmable terminal". If the snooper has been invoked through the system menu, "Quit" takes you back to that menu. If the snooper has been invoked from a user-programmed display page, you will return to that page.

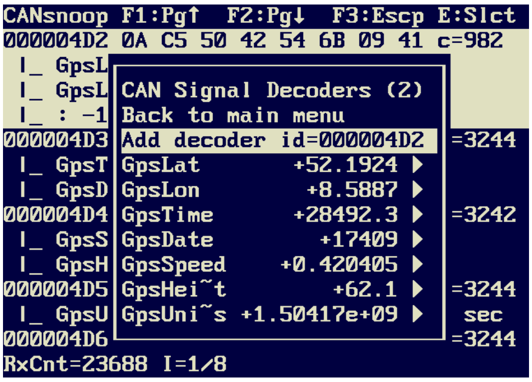
Caution: Using the snooper disables the "normal" CAN message processing of the application !

1.4 Signal Decoders in CAN Snooper mode

From the received CAN message traffic, a few signals can be isolated for display purposes.

You define a few ***signal parameters***, and the CAN snooper will turn bits and bytes from a CAN message into ***human-readable decimal numbers***.

To configure a Signal Decoder for a particular message, select the message which shall be decoded (so we know the ID) and open the CAN Snooper's main menu. Select "Signal decoders". A popup menu similar to the one shown below will appear:

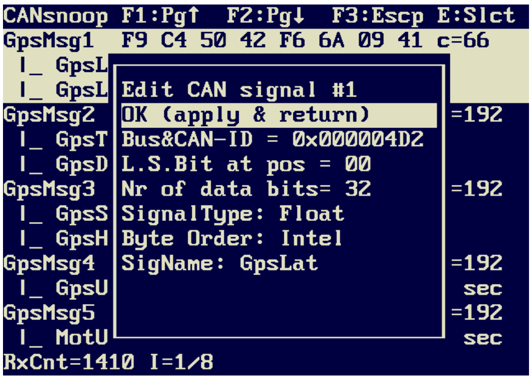


```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct
000004D2 0A C5 50 42 54 6B 09 41 c=982
  I_ GpsL
  I_ GpsL CAN Signal Decoders (2)
  I_ : -1 Back to main menu
000004D3 Add decoder id=000004D2  =3244
  I_ GpsT GpsLat         +52.1924 ▶
  I_ GpsD GpsLon          +8.5887 ▶
000004D4 GpsTime        +28492.3 ▶  =3242
  I_ GpsS GpsDate          +17409 ▶
  I_ GpsH GpsSpeed     +0.420405 ▶
000004D5 GpsHei~t          +62.1 ▶  =3244
  I_ GpsU GpsUni~s +1.50417e+09 ▶   sec
000004D6                            =3244
RxCnt=23688 I=1/8
```

(CAN snooper menu "Signal Decoders")

Select the menu item "Add decoder id=XXXXX" to define a new signal decoder. We already know the CAN-ID of the message, because that message has been selected in the message list. Otherwise, you would have to enter a 29-bit value without a numeric keyboard which is tricky ;-)

To define or modify one of the CAN Signal Decoders, another menu will appear on the terminal's screen:



```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct
GpsMsg1  F9 C4 50 42 F6 6A 09 41 c=66
  I_ GpsL
  I_ GpsL Edit CAN signal #1
GpsMsg2  OK (apply & return)        =192
  I_ GpsT Bus&CAN-ID = 0x000004D2
  I_ GpsD L.S.Bit at pos = 00
GpsMsg3  Nr of data bits= 32        =192
  I_ GpsS SignalType: Float
  I_ GpsH Byte Order: Intel
GpsMsg4  SigName: GpsLat            =192
  I_ GpsU                            sec
GpsMsg5                             =192
  I_ MotU                            sec
RxCnt=1410 I=1/8
```

(CAN snooper menu "Edit Signal Decoder")

As you see, the CAN Snooper needs some more information to decode a CAN signal. These are

- message ID
- position of the least significant bit within the message. Valid bit positions are 0..63 (!!) because a CAN message has up to 8 data bytes.

- count of databits for the signal. Often 8, 16, or 32 bits.
- the signal's byte order ("Intel" / "Motorola" format as in a CANdb signal)
- the signal type („Signed“, „Unsigned“, „Float“, „Double“)

In most cases, 16-bit-values will be on even byte boundaries within the CAN data field. Values in *Intel* format (little endian) are sent low-byte-first, *Motorola* format (big endian) is high-byte-first. Forget about "Motorola forwards" and "Motorola backwards" - we don't know the difference. Here, the bits inside a CAN are ALWAYS numbered as follows, no matter if the sender is *"Intel"*, *"Motorola"* or whatever :

Byte-Index	bit 7 (MSB)	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0 (LSB)
[0]	7	6	5	4	3	2	1	0
[1]	15	14	13	12	11	10	9	8
[2]	23	22	21	20	19	18	17	16
[3]	31	30	29	28	27	26	25	24
[4]	39	38	37	36	35	34	33	32
[5]	47	46	45	44	43	42	41	40
[6]	55	54	53	52	51	50	49	48
[7]	63	62	61	60	59	58	57	56

(numbering of data bits in a CAN frame)

Like in many programming languages, bits and indices are numbered from zero to "N minus one". The very first bit in the data field is BIT ZERO, not "BIT ONE".

After specifying all signal parameters, apply the changes and return to the previous menu by selecting "OK". This takes you back to the signal decoder menu, where the decoded values will be displayed as soon as new CAN messages with a matching identifier have been received. Decoders which have not received a signal yet will show the value '-99999999' which means 'invalid'.

1.5 Message Dump in CAN Snooper mode

For a convienient examination of the received traffic on a CAN MESSAGE level, the contents of the message buffer can be 'dumped' into a textfile on the memory card. You can examine, edit, and print the dump file on a PC later, all you need is a simple text editor.

The dump file format is the same as on the message screen, which means the dump file will show the last received CAN messages ***sorted by identifier***. Don't confuse this file format with the one used for the snooper's optional CAN message recorder.

Sample dump file:

```
    ; CANsnoop dump nr. 7
    ; started 2003-08-28 13:36:59
    ; dumped  2003-08-28 14:00:28
    00000003 08 BD 28 5E 4B 94 11 AF c=233
    000003FF 08 BC D3 5E 7E E9 7F 2F c=231
     ...
```

The header of a dump file includes date and time information, taken from the terminal's battery-buffered clock.We use *the one and only 'logical' date- and time format*, which is YYYY-MM-DD hh:mm:ss, the International Standard Date and Time Notation proposed by ISO 8601. Do a web search on "ISO 8601" for more info. For the future, avoid formats like YY/DD/MM, DD/MM/YY etc in text data files which may have to be processed on machines in other countries.

The filenames are produced automatically in the format "candmpNN.txt", where NN is a decimal number (index) from zero to 99. Whenever a message dump has been written to the CF Memory Card, the index is incremented. You can modify the current dump index in the *Other Functions* menu.

To generate a CAN message dump, you can..

- select a function key mode (set) which contains the DUMP function, then press the associated key to start a dump,

or

- select the function *Dump messages to file* in the *Other Functions* menu.

Note: The message dump function may take a few seconds to produce the text file. During this time, some messages will not be counted if there is a high bus load. You will notice an increasing *Lost Message Counter* in that case !

1.6 Defining "Actions" on reception of a certain CAN message

(T.B.D.)

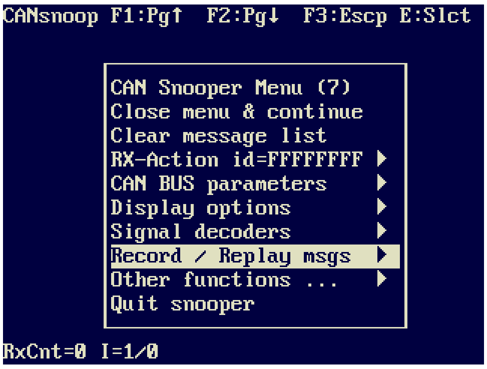
1.7 Recording and playing CAN messages with the CAN-Snooper

In devices with 32-bit CPU (like "MKT-View II"), the Snooper can record and play CAN messages, without the need for a PC.

The file format used by the snooper's recorder is a simple, but widely used[1], text-based format (in contrast to the CAN-Logger). Each CAN-message occupies about 80 characters in this file format, which is the reason that the CAN-Snooper's *CAN-Recoder* is not as fast as the *CAN-Logger* (which uses a binary format, and only 16 bytes per CAN-message).

Despite the limitation of a few hundred messages per second, the CAN-Snooper's recorder can be a very helpful tool for debugging and presentations. For example, you may record a few minutes of "live data" in a vehicle, and play it back later in an endless loop. That way, you can show the display terminal "in action" without the need for a PC, or a "real" CAN-bus.

To use the snooper's recorder, open the snooper's main menu as described in an earlier chapter, and select the item "Record / Replay messages" :

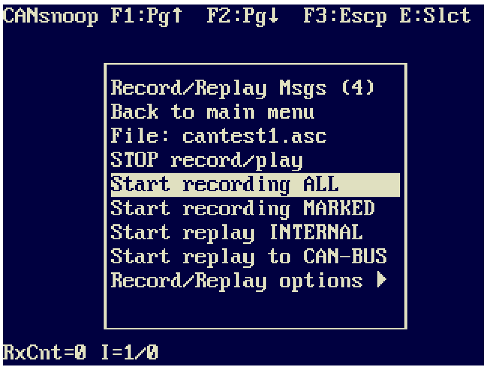


```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct

        CAN Snooper Menu (7)
        Close menu & continue
        Clear message list
        RX-Action id=FFFFFFFF ▶
        CAN BUS parameters    ▶
        Display options       ▶
        Signal decoders       ▶
        Record / Replay msgs  ▶
        Other functions ...   ▶
        Quit snooper

RxCnt=0 I=1/0
```

This will open a submenu like the one shown below (it may look different, depending on the display hardware):



```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct

        Record/Replay Msgs (4)
        Back to main menu
        File: cantest1.asc
        STOP record/play
        Start recording ALL
        Start recording MARKED
        Start replay INTERNAL
        Start replay to CAN-BUS
        Record/Replay options ▶

RxCnt=0 I=1/0
```

The items in this menu should speak for themselves.. if not:

- File: cansn**1.asc
  Can be used to modify the name of the recorded (or replayed) file.

[1] The CAN-snooper's message recorder / player uses a format which may be compatible with Vector's ASCII logfile format; in fact it's a subset with less spaces for the ID-field.

The file sequence number starts at 1 (one) by default, and will be incremented when stopping after *record* . This allows recording a sequence of up to 999 different files, without the need to select a new name for each new file.

- STOP record/play :
  Stops recording or playing a recorded file. See notes on the file sequence number in the previous paragraph: If recording was active (before 'stop'), the filename will be modified for the next sequence number.
- Start recording ALL
  Begins to record ALL received CAN messages, on all buses, regardless of the CAN-ID. Caution, the max message rate which can be recorded by the snooper (in contrast to the CAN logger) is limited to approx. 2000 ... 3000 messages per second, depending on the storage medium (SD memory card speed). A single CAN message occupies about 60 bytes in an ASCII log (*.asc, Vector CAN message format), so the file will grow quickly, and you may run out of disk space sooner or later !
- Start recording MARKED
  Begins to record only those CAN messages, which have earlier been marked in the CAN snooper's main window (the CAN message list, sorted by CAN-IDs).
- Start replay INTERNAL
  Starts to "play" the previously recorded messages, but doesn't transmit them physically. Instead, the messages from the logfile are only internally routed to the various CAN-message-handlers. The purpose of this function is to test / debug / demonstrate the terminal's display application without the need for a real CAN-bus network.
- Start replay to CAN-BUS
  Reads the CAN-messages from the selected file, and sends them out to the CAN-bus (physically).

Caution: Be aware of the consequences !
Make sure you know what you're doing !
Don't try this in a vehicle !
You have been advised .

Note: The CAN transceiver in some devices, for example "MKT-View II", may be configured for 'listen-only' mode through the system setup. In this case, playback to the CAN bus will not work. You will need to enable CAN transmission through the setup menu - see document #85115, titled "System Menu and Setup Options for User Programmable Displays", "CAN-TxEnable" .

- Record/Replay options ...
  This menu item opens another submenu, where some less frequently used parameters can be modified, as explained in the next chapter.

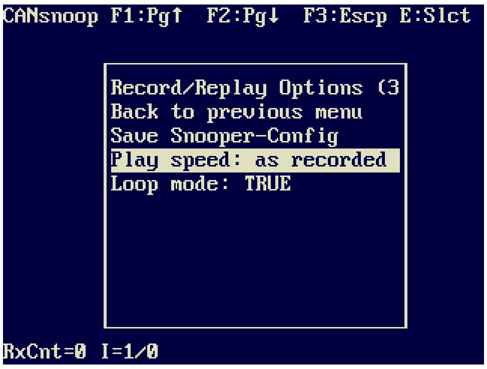
Alternatively, the CAN-Snooper's "playback" function can also be started through the following interpreter commands:

- snooper.play_file("cantest1.asc")
  Starts playing the specified file "internally", i.e. without sending it to the CAN-bus. This command may be very helpful for presentations, to let a program run in "demo mode"

without the need to use the snooper's menu. For example, put this command inside a button, or invoke it from a programmable function key, etc.

- snooper.send_file("cantest1.asc")
  Starts sending (replaying) the specified file to the CAN-bus.

1.7.1 CAN Snooper Record / Replay Options



```
CANsnoop F1:Pg↑  F2:Pg↓  F3:Escp E:Slct

            Record/Replay Options (3
            Back to previous menu
            Save Snooper-Config
            Play speed: as recorded
            Loop mode: TRUE

RxCnt=0 I=1/0
```

The Record/Replay Options menu contains (at least) the following items:

- Save Snooper-Config:
  Saves the current SNOOPER settings permanently in a file named SNOOPER.CFG .
- Play speed:
  Defines the CAN message interval when playing a previously recorded file back to the CAN-bus, or (as a simulation) into the display application. Possible values are listed further below.
- Loop mode
  FALSE = During playback, only play the contents of the logfile *once*. Don't repeat.
  TRUE = When reaching the end of the file during playback, rewind back to the start, and play again.

The REPLAY SPEED is usually set to "as recorded", which means during playback, the gaps between two CAN transmissions are as close to the original recording as possible. The timing accuracy depends on the CPU load; the author expected it to be accurate to fractions of a millisecond.
Alternatively, a recorded logfile can be played back with one of the following fixed timings. This feature turns the CAN-Snooper into a powerful testing tool, because it allows to to play back a recording faster or slower than life, for example to put a CAN node 'under stress' (to see how it handles CAN receive buffer overruns, etc).
Available playback speed options are:

- Playback timing "as recorded" : use the timestamp in the recorded file to decide when to send the next message.
- "10 Messages/second" : Very low bus load, 10 messages per second.
- "100 Messages/sec" : Similar as above. Causes a moderate bus load.
- "1000 Messages/sec", "2000 Messages/sec", "5000 Messages/sec" :
  Plays the messages with the specified frequency, if possible; regardless of the original timestamps in the recording. This may cause a high bus load, depending on the CAN bitrate. 5000 Messages/sec only works gapless if the entire file fits in a buffer (see notes below), and is intended only for "stress-testing" at the manufacturer !

Note:
Keeping the replay function running at high speed while returning to the display application may slow down the display functions. Consider this: While playing a logfile in Vector ASCII format, the CPU must read and parse 5000 text lines per second (with 80 characters per line) from the file. But

if the logfile is sufficiently small to fit inside an internal "file buffer", it only needs to be parsed once (in loop mode), and can be played from the buffer without parsing it again.
For this reason, if you need to play logfiles back at high speed without gaps, make sure the file contains less than 32768 CAN-messages. This can easily be achieved with a text editor, since the snooper uses ASCII CAN logfiles by default, which contain one CAN message per text line.