

Streamlogging

FAQ

What is Streamlogging? → [more information](#)

At Streamlogging **all on CAN bus received messages/identifiers are stored to raw measurement data** with an exact time stamp and without any setting of loggers CAN-IN channels required.

The generation of a 2D measurement from the raw measurement data, i.e. the assignment of the CAN identifiers, is now assigned to the channel names on the computer via a channel settings file and can thus be carried out as often as required and with different channel settings files.

What are the advantages of Streamlogging? → [more information](#)

Streamlogging basically works like common 2D Sticklogging but additionally provides advantages:

- 1) By assigning the CAN identifiers to the channel names on the computer, it is possible to generate 2D measurements with an **“unlimited” number of OFFLINE 2D CAN channels** from the raw measurement data due to the now irrelevant memory size of the measurement on the computer.
- 2) This assignment and thus the generation of a 2D measurement on the computer **can be carried out as often as desired and in each case with different channel setting files.**

What is the difference from Streamlogger to Sticklogger?

The difference in Stick-and Streamlogging is **only in behaviour of CAN-Bus channel logging** because of another firmware! With Streamlogger **still 128 ONLINE CAN bus channels can be used** for recording and other purposes like sending or rerouting channels via CAN or using in online calculation channels (CALC)!

With Streamlogging only advantages are gained because every time all on CAN bus sent messages are recorded automatically, even if they are not specified for recording in loggers CAN-In group!

Therefore, at Streamlogging basically no CAN-IN channels must be set for recording because every available and at Streamlogging automatically recorded CAN-messages can be linked to channel setting file (DBC/CAN-DB) on computer when creating 2D measurements.

Hence, if logger setting was made via WinIt and a CAN-channel setting was done wrong or CAN channel was completely forgotten at logger setting, with Streamlogging this missing or wrong set CAN-channel can be restored by applying correct channel setting file (DBC/CAN-DB file) on computer.

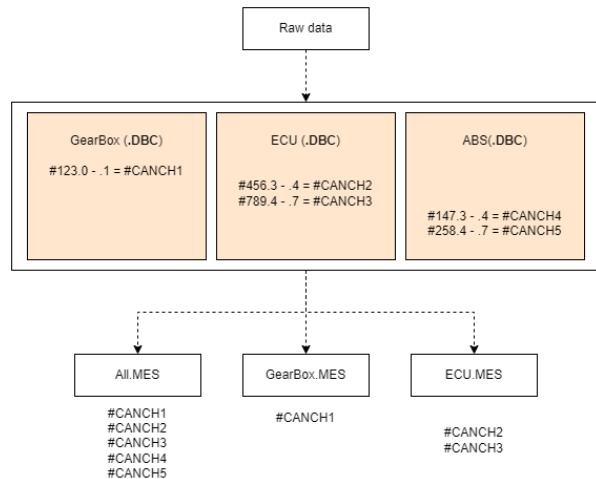
What are channel setting files (DBC/CAN-DB files) and for what are they used?

[→ more information](#)

A DBC/CAN-DB file is an ASCII based translation file used to apply identifying names, scaling, offsets, and defining information, to data transmitted within a CAN frame.

For any given CAN ID, a DBC file can identify some or all the data within the CAN frame.

Channel setting files (DBC/CAN-DB files) are therefore used to link any required information to raw data CAN-IDs to be able to create 2D measurement with respective channels.



DBC/CAN-DB files can be applied to raw measurement data **as often as desired** and in combined forms to create different 2D measurements (e.g., All.MES with where GearBox.DBC and ABS.DBC was used, GearBox.MES only with GearBox related channels defined in GearBox.DBC, ECU.MES with only ECU related channels defined in ECU.DBC, ...)

At 2D this process of linking DBC/CAN-DB file to raw measurement data to create 2D measurement is called decompressing.

Where do I find a suitable channel setting file (DBC/CAN-DB file)?

2D modules: At every *Apply* in Winlt the DBC/CAN-DB file for all CAN bus of all CAN-modules currently loaded in Winlt are stored into 2D folder structure.

WinARace → [CTRL]+[ALT]+[d] → System → CAN-DB

Normally all ECU manufacturer are providing DBC/CAN-DB files for their respective ECU's.

Where can the raw measurement data be found after the download? → [more information](#)

After inserting a USB-Stick into PC for downloading the containers with raw measurement data (.000) files are copied to the *Temporary directory* of Race-Software (WinARace → *Folders - Protocol* → *Temporary directory*) into download folder (.DL).

How can I apply another channel setting file (DBC/CAN-DB-files) to raw measurement data to create new measurements? → [more information](#)

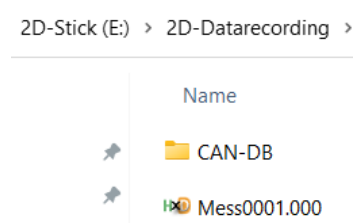
Drop respective download folder (.DL) on WinARace. The respective download folder (.DL) can located on every directory on computer.

To apply channel settings on multiple download folder (.DL) at WinARacet-tab *Modules* the *2D USB-Stick Handler* must be opened. Then drop the selected downloader folders (.DL) on opened *2D USB-Stick Handler* window to start decompressing.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

Downloading measurement data from USB-Stick – Where do I need to store DBC/CAN-DB file on USB-Stick? → [more information](#)

In order to be able to apply a channel setting file (DBC/CAN-DB-file) of raw stream logger data at **initial data download from USB-Stick**, the channel setting file must be located **on the 2D-USB stick in the CAN-DB directory within the 2D-Datarecording directory**.



To define which CAN-DB should be used for which CAN-Bus at decompressing, the file name extensions of CAN-DB files on USB-Stick are checked:

..._CAN1.DBC → used for channels from CAN 1

..._CAN2.DBC → used for channels from CAN 2

Which channel setting file (DBC/CAN-DB file) is used when decompressing already downloaded measurement data again? → [more information](#)

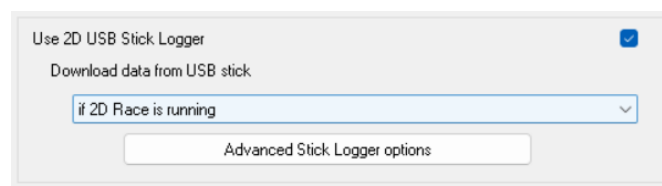
With **activated** User-Interface, the DBC/CAN-DB files can be selected during decompressing process.

With **deactivated** User-Interface, the DBC/CAN-DB files must be copied into to respective download folder (.DL)

If another event should be used for decompressing than at initial download from USB-Stick, please delete .ini-file from .DL folder to create 2D measurement in currently selected event. Otherwise data is decompressed in event which is specified in .ini-file inside .DL.

How can the User-Interface be enabled/disabled? → [more information](#)

WinARace → Settings → System → Advanced Stick Logger options



How is it possible to determine whether a logger is a Streamlogger?

WinIt → read in system with respective logger → Tab *Version* → Firmware

If the firmware name contains ...*STRM*... the logger works as Streamlogger.

DashTFT modules with USB-Recording option are only available with Streamlogger-firmware.

What are the requirements for using Streamlogging? → [more information](#)

Streamlogging firmware > **105** or DashTFT with **USB-Stick-Logging option Race2022**

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

Table of content

| | |
|--|----------|
| FAQ | 1 |
| PREFACE | 4 |
| 1.1 REQUIREMENTS | 4 |
| 1.1.1 Software..... | 4 |
| 1.1.2 Hardware | 4 |
| FURTHER INFORMATION | 5 |
| 1.2 INTRODUCTION | 5 |
| 1.3 IMPORTANT STICKLOGGING DOWNLOAD/DECOMPRESSING INFORMATION..... | 6 |
| 1.3.1 Find DL folder for respective measurement on computer | 7 |
| 1.4 DOWNLOADING DATA..... | 7 |
| 1.4.1 Sticklogger firmware | 7 |
| 1.4.2 Streamlogger firmware | 8 |
| 1.4.3 Comparison Sticklogger/Streamlogger..... | 9 |
| 1.5 POSSIBLE WAYS TO APPLY CHANNEL SETTING FILE (DBC/CAN-DB-FILE) TO RAW DATA | 10 |
| 1.5.1 USB-Stick | 10 |
| 1.5.2 Manual decompressing: | 11 |
| 1.6 STREAM-USER-INTERFACE..... | 12 |
| 1.6.1 Decompress options for stream logger - Add DBC/CAN-DB files..... | 12 |
| 1.6.2 Channel rates to extract from stream | 14 |
| 1.7 DEBUG-FILES | 15 |
| 1.7.1 Streamreport..... | 15 |
| 1.7.2 Streamcontent | 16 |
| 1.7.3 Activate/Deactivate Stream-user-Interface..... | 17 |
| 1.8 PLAYBACK FUNCTIONALITY | 21 |
| 1.8.1 USB-Stick preparations | 21 |
| 1.8.2 Streamlogger preparations | 22 |
| 1.8.3 Start playback..... | 22 |

Preface

1.1 Requirements

1.1.1 Software

When *Streamlogging* is used at least Race 2022 software is used.

Please ensure that latest software version is installed!



- To check for new updates:
 - a) WinARace → Help → Search for software updates
 - b) Download [update file](#) and drop on WinARace

1.1.2 Hardware

Streamlogger firmware > **105** or DashTFT with **USB-Stick-Logging option**

Further information

1.2 Introduction

An important difference to Sticklogger is that Streamlogger, in addition to all known Sticklogger functions, can **also record all messages sent on the CAN bus**, allowing 2D measurements to be made **with an unlimited number of OFFLINE CAN channels!**

At Stickloggers only the CAN channels are recorded which are specified for recording in CAN-IN group in logger setting (.LDD).

So, the difference in Stick-and Streamlogging **is only in behaviour of CAN-Bus channel logging** because of another firmware!

With Streamlogger **still 128 ONLINE CAN bus channels can be used** for recording and other purposes like sending or rerouting channels via CAN or using in online calculation channels (CALC)!

The “unlimited” number of OFFLINE CAN channels are created by using DBC/CAN-DB files at decompressing on computer.

Stream- and Stickloggers can be still used for recording multiple CAN bus.

Because DashTFT is also able to record data, Streamlogging firmware is also available for DashTFT!

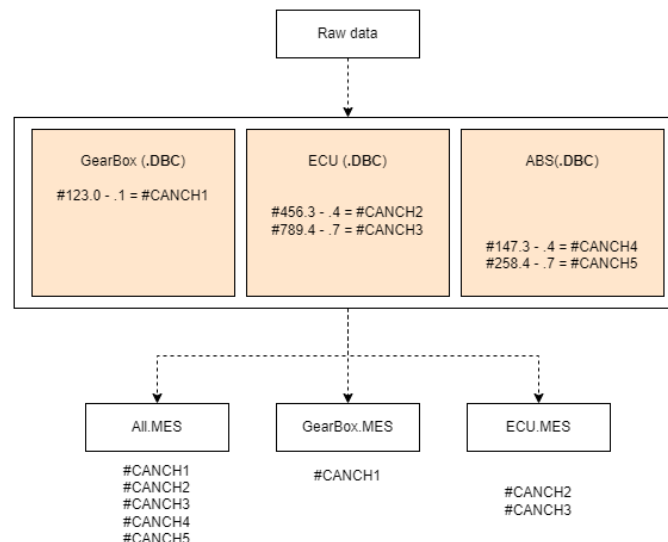
With Streamlogging only advantages are gained because every time all on CAN bus sent messages are recorded, even if they are not specified for recording in loggers CAN-In group!

Advantages Streamlogging

1. Streamlogging has only advantages, because every time all messages sent on the CAN bus are recorded in the measurement raw data, even if they were not intended for recording in the CAN-In group of the logger!

This makes it possible, even after the actual measurement at any time and as often as desired, to generate all required CAN channels as a 2D measurement from the stored raw data using a DBC/CAN-DB file!

2. In addition, Streamlogging and the use of different DBC/CAN DB files can be used to provide different team members with only the required 2D channels in a measurement.



1.3 Important Sticklogging download/decompressing information

At all Stick- and Streamlogger-firmware **from 2021 on**, 2D One File Recording format is when logging data on USB Stick.

Thereby, data security/integrity is improved when recording to the USB stick when it is unplugged, or the system is switched off while data is being written to it.

This is achieved by completely avoiding writing to the file system table of the USB stick.



Further Information

- One File Recording firmware can be identified by extension ...OF....hex.
- One File Recording is also used at Streamlogging.
- One File Recording is available from FW 95 and required at least Race2021.

When logging data to USB-Stick, the logger setting, and raw measurement data are now both written into the same container file (Mess____.000) on USB-Stick.

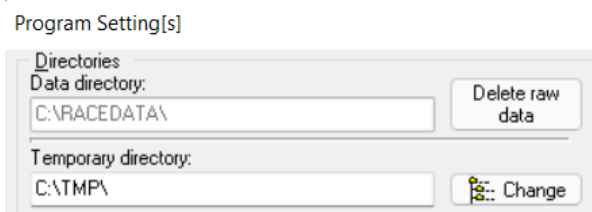
When USB-Stick is plugged into computer for data download, the container files (Mess____.000) are decompressed by 2D software tool (Decompress.exe) whereby the different files are extracted.

1. Measurement data (.MES-folder) → is created inside selected event
2. Logger setting file (.LDD-file) → is stored inside respective .MES in event
3. Raw download data (.DL-folder) → is stored in TMP directory

Via WinARace → Settings → Destination of raw data it can be chosen if and where raw measurement data is stored.

| | |
|--|---|
| Temporary folder – delete after decompress | Raw measurement data is deleted after successful decompression |
| Temporary folder – keep files | Raw measurement data is kept after successful decompression in <i>Temporary directory</i> |
| In measurement folder | Raw measurement data is kept after successful decompression in respective .MES directory |

For security reasons, the raw download data from each measurement should always be kept in download folder (.DL) *Temporary directory* which is set at every software by default to ...\\TMP\ (WinARace → Settings → Folders – Protocol)







This makes it possible to access the raw measurement data of the USB stick again and again.



Important information

Please not change or delete the .DL folders from *Temporary directory*!
 Inside *SpecSheet* of every measurement the name of respective download folder (.DL) is stored!

Example: OneFileRec0547.DL in Temporary directory

| | |
|--|---|
|  OneFileRec0547.LDD | Logger setting file |
|  OneFileRec0547.ini | Decompress information (decompress path of created measurement) |
|  OneFileRec0547.000 | Raw measurement data |
|  0329-RMSTK3-20.HED | SpecSheet of decompressed measurement (important for re-decompressing) |
|  29CC635E_BC-GNSS2CAN_IMU_Send_CAN1.DBC | DBC/CAN-DB-file which was used at for decompressing data (can be more than one) |

1.3.1 Find DL folder for respective measurement on computer

In SpecSheet group DOWNLOAD of every measurement the information is stored where the .DL folder is stored **on the computer which was used for downloading the data.**

| GRUPPEN | 1124-CAR_17-01.HED |
|--------------------|--------------------------|
| LapMeter [m] | U |
| Date | 22.1124 |
| Time | 133249 |
| Run total [s] | 0 |
| Run recording [s] | 0 |
| Start of download | 24.11.2022 13:32:49 |
| End of download | 24.11.2022 13:32:49 |
| Number of errors | 0 |
| Time to decompress | 0.0 sec |
| Loggername | Car_17 |
| Logger serial | 2A3357CA |
| Mileage calculated | (0km) |
| MainSamplingRate | 1000 |
| DL folder | C:\TMP\OneFileRec0184.DL |

1.4 Downloading data

After inserting a USB-Stick into PC for downloading the containers with raw measurement data (.000) files are decompressed.

At decompressing by Decompress.exe, the different information from raw measurement data (.000) is extracted and a 2D measurement (.MES) is created.

1.4.1 Sticklogger firmware

With normal Sticklogger firmware, the software automatically decompresses the raw measurement data with the channel information given by logger setting file (.LDD).



Further Information

At Streamloggers, the channels defined for recording in logger setting file (.LDD) are always preselected for decompressing!

1.4.2 Streamlogger firmware

With Streamlogging firmware, additionally to the channel information given by logger setting file (.LDD), DBC/CAN-DB files can be used to extract also channels from raw measurement data which were not defined for recording in logger setting file but are available on the different CAN bus! Here it is also distinguished which DBC-/CAN-DB-file should be used for which CAN bus.



Further Information

Depending on the settings of the software, different windows are displayed here during the decompress process, in which, for example, different DBC/CAN DB files can be used for decompression.

In addition to the integration of DBC-/CAN-DB-files during the decompress process, it is also possible to store DBC-/CAN-DB-files on the USB-Stick used.

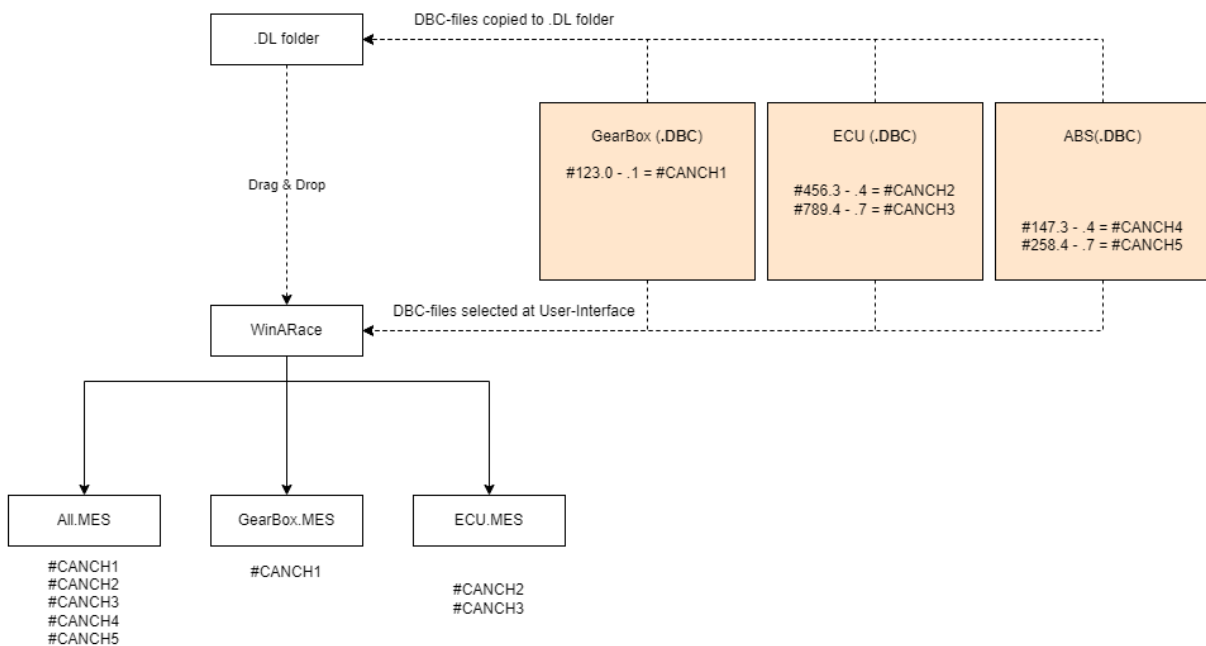
A DBC-/CAN-DB-file stored on the 2D-Stick in the CAN-DB folder is always preselected during the decompress process.



Further Information

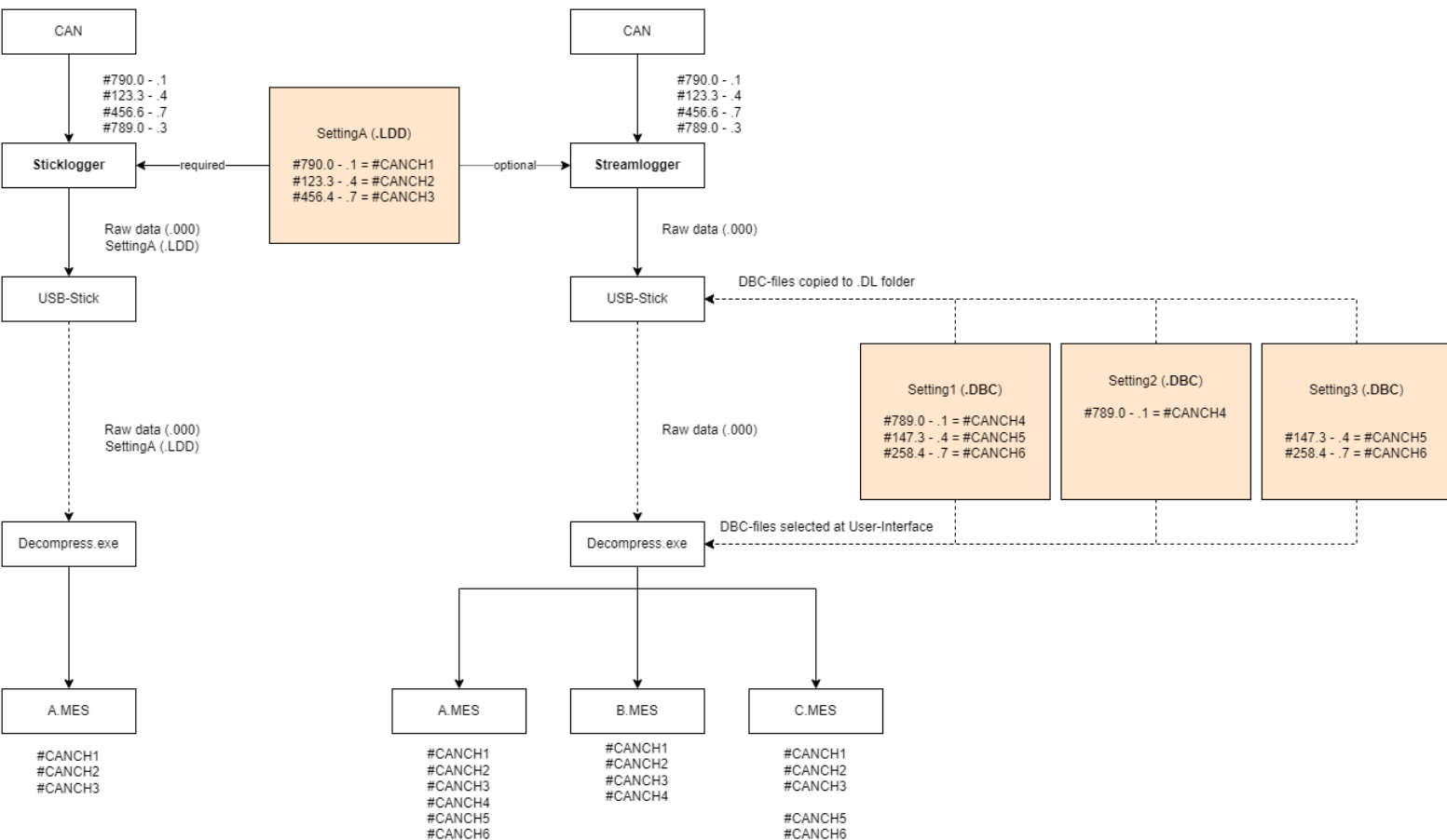
Folder CAN-DB is already created by Stick-preparation program or created manually! If all Streamlogger user interface windows are deactivated, the DBC/CAN DB files stored in the folder on the USB stick are automatically used for decompressing the raw measurement data.

If the extension _CAN1 is appended to a DBC-/CAN-DB-file in this folder, the DBC-/CAN-DB-file is only used for decompressing the channels on CAN1 (the same applies to the other CAN-buses).



1.4.3 Comparison Sticklogger/Streamlogger

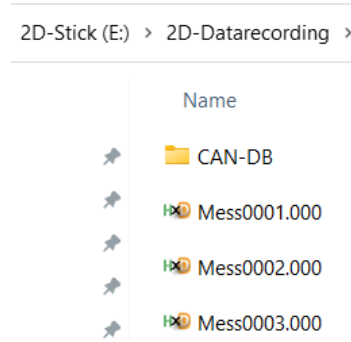
| Sticklogger | Streamlogger |
|---|--|
| Only records its CAN channels which are <u>defined by the</u> previously done setting (.LDD) | Records <u>all</u> on CAN Bus <u>received</u> CAN identifiers with unique time stamp. |
| Stickloggers can only record max. 128 ONLINE CAN channels | With Streamloggers max. 128 ONLINE CAN channels can be recorded <u>plus</u> a measurement with an <u>unlimited</u> number of OFFLINE CAN 2D Analyzer channels can be created at decompressing! |
| Recorded CAN channels are <u>immediately</u> available after download | Recorded (raw) CAN identifier data must be linked to a CAN-DBC file at decompressing <i>(only in CAN-DBC file used channels will be created as 2D-channels for Analyzer)</i> |
| | Recorded (raw) CAN identifier data can be used for creating different 2D measurements multiple times |
| | Streamlogger can also record <u>certain</u> defined CAN channels (.LDD) and all on CAN Bus received CAN channels, which must be decompressed with DBC, at same |



1.5 Possible ways to apply channel setting file (DBC/CAN-DB-File) to raw data

1.5.1 USB-Stick

In order to be able to apply a channel setting file (DBC/CAN-DB-file) of raw stream logger data at initial data download from USB-Stick, the channel setting file must be located **on the 2D-USB stick in the CAN-DB directory within the 2D-Datarecording directory**.



Important information

Since the setting file should already be on the 2D-USB stick before the measurement data download, the channel setting file should already be copied to the stick before the measurement. A channel setting file does not necessarily already have to be on the 2D-USB stick during the download, it can also be applied to the raw measurement data later when manually decompressing.



Further Information

Folder *CAN-DB* is already created by Stick-preparation program or created manually! If the extension *_CAN1* is appended to a DBC-/CAN-DB-file in this folder, the DBC-/CAN-DB-file is only used for decompressing the channels on CAN1 (the same applies to the other CAN-buses).

1.5.2 Manual decompressing:

After the raw measurement data has already been downloaded to the PC, channel setting files can be applied to the raw measurement data as often as desired to create 2D measurements.

a) Open *WinARace*

b) Open directory where DL-folder with raw measurement data is stored (...\\TMP\\...)



Important information

If another event should be used for decompressing than at initial download, please delete .ini-file from .DL folder.

Selected event is then used for decompressing.

Otherwise data is decompressed in event which is specified in .ini-file inside .DL.

c) Drag and drop respective .DL folder on opened *WinARace* to start decompressing



Important information

With deactivated Stream-User-Interface, the channel setting files inside .DL folder are used for creating 2D measurement.

d) The measurement is decompressed



Important information

Only those channels are generated in the 2D measurement, from which the respective CAN identifier was also received in the raw measurement file!

Channels which are defined for recording in logger setting file (.LDD) are decompressed every time!

1.6 Stream-User-Interface

If the user interface is **activated** for decompressing (see 1.7.3 how to activate), it is possible to display a user interface when decompressing (after step c from previous chapter) the data (whether manually or when downloading the data initially from the stick), in which various DBC/CAN-DB files can be added to the decompression process in order to create additional channels from the raw measurement data.



Further Information

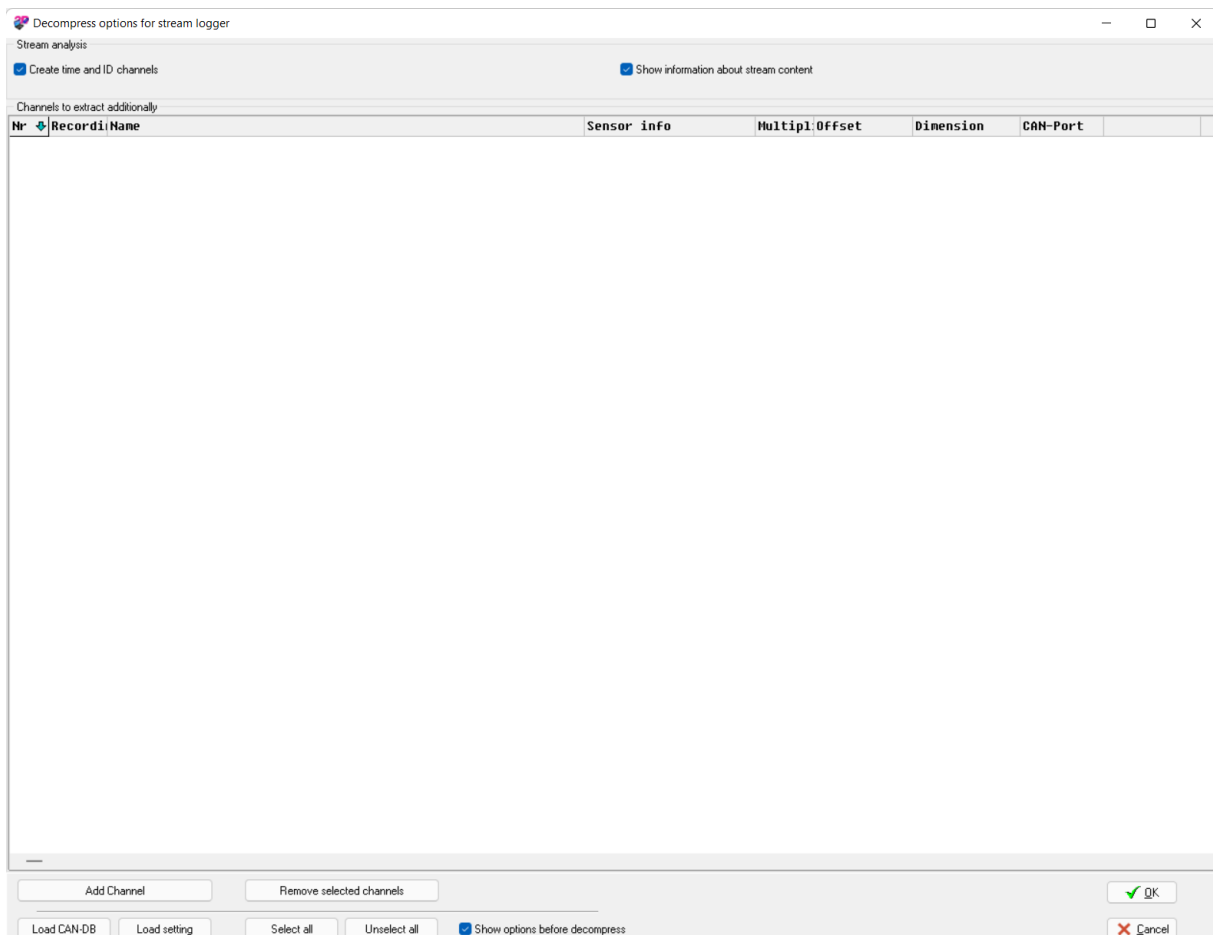
When the user interface is **deactivated**, the DBC/CAN-DB files are automatically used which are located in the CAN-DB directory (1.5.1) when the raw measurement data are downloaded initially to the USB stick, or in the download folder (.DL) when the data are decompressed manually (1.5.2).



Important information

Stream-User-Interface is **deactivated** by default, that at download of measurement data from USB-Stick only to common 2D-Download windows are shown!

1.6.1 Decompress options for stream logger - Add DBC/CAN-DB files



Further Information

Channels defined in DBC/CAN-DB file in .DL folder (at manual decompression) or from CAN-DB folder on USB-Stick (at initial decompression from USB-Stick) are preselected in this list.

1.6.1.1 Load CAN-DB

Via the buttons *Load CAN-DB* further DBC/CAN-DB-files can be loaded which are used for decompressing data.

At column Sensor Info the source of respective channel can be distinguished:

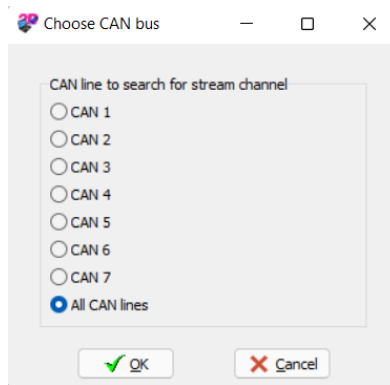
| | | | | | | | | | |
|----|-------------------------------------|------------------|---------------------|-----|------------|---|-----|---|----|
| 47 | <input checked="" type="checkbox"/> | STEA_FTAX_EFFU | OrgLoggerChannel1 | 0.0 | -89.9° | 2 | 302 | 2 | 16 |
| 48 | <input checked="" type="checkbox"/> | LAPTIME_GNSS2CAN | OrgLoggerChannel1 | 0.0 | 0.0 Sec | 1 | 100 | 4 | 32 |
| 49 | <input checked="" type="checkbox"/> | MILE_KM | OrgLoggerChannel1 | 1.0 | 0.0 km | 2 | 330 | 0 | 24 |
| 50 | <input checked="" type="checkbox"/> | ACC_X_GPS | Rotated Acc_x value | 0.0 | -163.8:m/s | 1 | 498 | 0 | 16 |
| 51 | <input checked="" type="checkbox"/> | ACC_Y_GPS | Rotated Acc_y value | 0.0 | -163.8:m/s | 1 | 498 | 2 | 16 |



Further Information

Channels defined for recording in logger setting file (.LDD) are marked with sensor info OrgLoggerChannel

If necessary, select respective CAN-bus where the channels from selected CAN-DB/DBC-file is received



If all channels from CAN-DB/DBC-file are added to list, please click on OK for decompressing data.



Further Information

If no channels are selected in this overview only the originally in WinIt-Setting recorded channels (LDD-File) are created at decompressing.

1.6.2 Channel rates to extract from stream

Because at decompressing the software detects on which rate the CAN channels are really send on CAN-bus by analysing receiving time stamps, it can be selected which channel rate should be used for decompressing.

Only select the necessary sampling rates.



Further Information

If at decompressing a channel rate is detected by the software (see Stream-Report) which is not selected in list, the **next higher, activated** rate is selected for respective channel to not lose any information of this channel!

Press OK to start decompressing to create 2D measurement.

Channel rates to extract from stream

| Frequency | Hz | Divisor of base rate |
|-------------------------------------|-------|----------------------|
| <input checked="" type="checkbox"/> | 16000 | 1 |
| <input type="checkbox"/> | 8000 | 2 |
| <input type="checkbox"/> | 4000 | 4 |
| <input type="checkbox"/> | 3200 | 5 |
| <input checked="" type="checkbox"/> | 2000 | 8 |
| <input type="checkbox"/> | 1600 | 10 |
| <input checked="" type="checkbox"/> | 1000 | 16 |
| <input type="checkbox"/> | 800 | 20 |
| <input type="checkbox"/> | 640 | 25 |
| <input checked="" type="checkbox"/> | 500 | 32 |
| <input type="checkbox"/> | 400 | 40 |
| <input type="checkbox"/> | 320 | 50 |
| <input checked="" type="checkbox"/> | 200 | 80 |
| <input type="checkbox"/> | 160 | 100 |
| <input checked="" type="checkbox"/> | 100 | 160 |
| <input type="checkbox"/> | 80 | 200 |
| <input type="checkbox"/> | 64 | 250 |
| <input checked="" type="checkbox"/> | 50 | 320 |
| <input type="checkbox"/> | 40 | 400 |
| <input type="checkbox"/> | 32 | 500 |
| <input checked="" type="checkbox"/> | 25 | 640 |
| <input checked="" type="checkbox"/> | 20 | 800 |
| <input type="checkbox"/> | 16 | 1000 |
| <input checked="" type="checkbox"/> | 12.5 | 1280 |
| <input checked="" type="checkbox"/> | 10 | 1600 |
| <input checked="" type="checkbox"/> | 6.25 | 2560 |
| <input checked="" type="checkbox"/> | 5 | 3200 |
| <input type="checkbox"/> | 3.125 | 5120 |
| <input checked="" type="checkbox"/> | 2 | 8000 |
| <input checked="" type="checkbox"/> | 1 | 16000 |

OK Cancel

1.7 Debug-Files

For the analysis of presence, actual transmitted rate or many other information of a CAN channel on a CAN bus, different debug outputs can be activated in the 2D software.



Further Information

Many debug outputs are already stored in the respective measurement data directory by default!

1.7.1 Streamreport

The stream report can be used to check which CAN IDs are received on which CAN bus. In addition, the sampling rates at which the respective CAN identifiers were actually received are determined here.



Further Information

This file contains an overview of **all** (and not only of recorded) CAN IDs that can be received on the respective CAN bus!

Inside every .MES folder a Streamreport.txt is stored automatically at download/decompression!

CAN-IDs received on respective CAN-bus + internal channels:

```
Number of internal channels found : 52
[ADC#5, vExt, vUSB, vOUT, V_Sat, ValidSat, SSHH, Course, Lat_dez, Lon_dez, Altitude, MMDD, HHMM, HorAccu, VerAccu, SpAccu, CourAccu, GAP, EVENTTRG,

Number of base identifiers on CAN1 (Number of occurrences in brackets): 3
[0x498(78758), 0x499(78758), 0x794(9845)]

Number of base identifiers on CAN2 (Number of occurrences in brackets): 25
[0x121(39375), 0x123(39376), 0x125(39376), 0x127(39375), 0x200(168672), 0x201(168673), 0x202(33734), 0x203(33734), 0x204(33734), 0x205(33734), 0x20
```

List of base identifiers in order of occurrence:

Here is shown in histogram form with scaling of 0.1 ms, how many CAN packets were received with which actual sampling rate.

Based on this detection of the actual sampling rate, a sampling rate is then assigned to the respective channel.

```
List of base identifiers in order of occurrence
*****
0x200 (512) CAN port: 2
Determined sampling rate: 500.000 Number of occurrences: 168672
Rate fitted to preselected frequencies: 500.000
Time spectrum in stream:
Time difference in ms | 0.9| 1.0| 1.1| 1.2| 1.3| 1.4| 1.5| 1.6| 1.7| 1.8| 1.9| 2.0| 2.1| 2.2| 2.3| 2.4| 2.5| 2.6| 2.7| 2.8| 2.9| 3.0| 3.1| 3.2| 3.3| 3.4| 3.5| 1000.0|
Number of occurrences | 8| 6| 11| 57| 394| 586| 8438| 805| 3278| 9055| 7884| 104595| 6848| 8784| 2831| 718| 9085| 696| 432| 163| 180| 1999| 3256| 81| 64| 18| 2| 1| 1|
```

In this case 104595 of 168672 CAN-packages of ID 0x200 are received with time difference of 2.0 ms.

1.7.1.1 Show Streamreport after decompression

This Streamreport.txt located in each measurement folder can be automatically displayed each time raw measurement data is downloaded/decompressed.



Further Information

To active the automatically displaying of Streamreport after every download/decompression, the Decompress2D.ini entry *ShowStreamStatistics* must be set to 1. For more information about Decompress2D.ini please refer to chapter 1.7.3.1.

1.7.2 Streamcontent

After the software has determined the actual sampling rate of each CAN identifier, the software checks whether the determined sampling rate is also in the *channel rates to extract* selection (see chapter 1.6.2).

As already mentioned in the linked chapter, if the detected actual sampling rate is not activated, the next higher activated sampling rate is used.

The Streamcontent.txt then contains the respective assignment of the actual sampling rate to the sampling rate used.

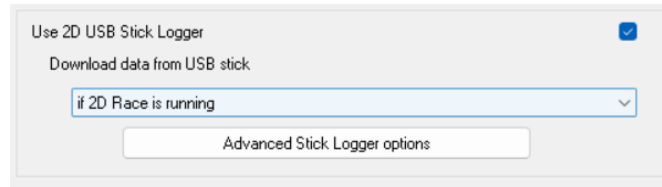


Further Information

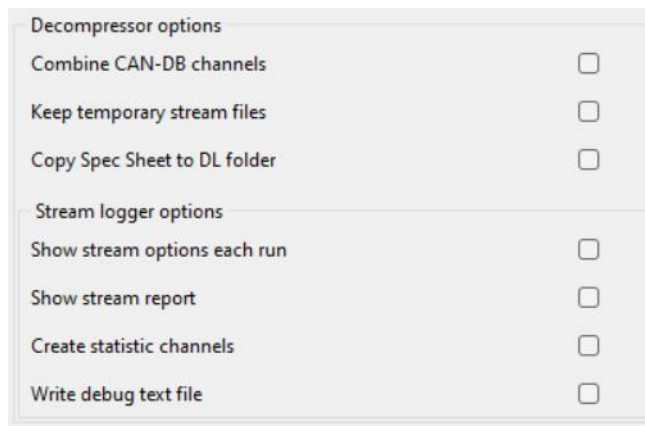
Inside every .MES folder a Streamcontent.txt is stored automatically at download/decompression!

1.7.3 Activate/Deactivate Stream-user-Interface

1. WinARace → Settings → System → Advanced Stick Logger options



2. Switch on or off the required options



| Decompressor options | | Link |
|------------------------------------|---|------------------------|
| Combine CAN-DB channels | | |
| Keep temporary stream files | Keep StreamFile folder in download folder | |
| Copy SpecSheet to temporary folder | Copy SpecSheet created at initial download to download folder. Especially important with activated SpecSheet-User-Interface | Manual |
| Stream logger options | | |
| Show stream options each run | Activate to show Stream-User-Interface at each decompression process | 1.6 |
| Show stream report | Activate to show Streamreport.txt at download/decompressing (MES folder) | 1.7.1 |
| Create statistic channels | Activate to create Analyzer channels which containing time (#t_Ch_), identifier (#ID_ch_) and length (#L_Ch_) information of every received CAN message | |
| Write debug text file | Create human readable CAN identifier debug report <MessName>.DBG at decompressing in MES folder | |

1.7.3.1 Decompress2D.ini flags

Underlined entries are default settings!

| | | | |
|--------------------------------|---|-------------|--|
| [Special] | | | |
| CombineCanDBChannels | = | <u>0</u> /1 | |
| KeepTempStreamFiles | = | <u>0</u> /1 | Keep StreamFile folder in download folder (for debugging) |
| CopySpecSheetToTemp | = | <u>0</u> /1 | Copy SpecSheet created at initial download to download folder (Especially important with activated SpecSheet-User-Interface → Manual) |
| [CAN-Stream] | | | |
| <u>CreateStatisticChannels</u> | = | <u>0</u> /1 | Activate to create Analyzer channels which containing time (#t_Ch_), identifier (#ID_ch_) and length (#L_Ch_) information of every received CAN message |
| <u>ShowStreamStatistics</u> | = | <u>0</u> /1 | Activate to show Streamreport.txt at download/decompressing (MES folder) |
| <u>ShowStreamOptions</u> | = | <u>0</u> /1 | Activate to show channel selection to select DBC file at decompressing |
| WriteDebugText | = | <u>0</u> /1 | Create human readable CAN identifier debug report <MessName>.DBG at decompressing in MES folder |
| ExtractTextFiles | = | <u>0</u> /1 | When CAN identifier sends its data as serial data stream via CAN identifier, serial data stream can be additionally extracted as text file for debugging |
| ExtractBinFiles | = | <u>0</u> /1 | When CAN identifier sends its data as serial data stream via CAN identifier, serial data stream can be additionally extracted as BIN file for debugging |
| TimeInExtract | = | <u>0</u> /1 | Additionally write time of occurrence to text file |
| [SIO2CAN] | | | |
| ExtractPairs | = | | If <i>ExtractTextFiles</i> or <i>ExtraxtBinFiles</i> was activated, the CAN identifiers to extract in text/bin file can be selected. |

| | | | |
|-----------------|---|----------------|---|
| TextFileIDsCAN1 | = | 0x1D0, ... | Extract all bytes from CAN identifier 0x1D0 from CAN bus 1 to text file |
| TextFileIDsCAN2 | = | 0x1D0, ... | Extract all bytes from CAN identifier 0x1D0 from CAN bus 2 to text file |
| BinFileIDsCAN1 | | 0x01,0x100,... | Extract all bytes from selected CAN identifier from CAN bus 1 to BIN file |
| BinFileIDsCAN2 | | 0x01,0x100,... | Extract all bytes from selected CAN identifier from CAN bus 2 to BIN file |
| Pairing | | | Display two identifiers in a file over time (who sends what and when) |
| | | | |
| | | | |

| | | |
|-------------|---|---|
| [Stream] | | |
| OVSelection | = | 1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,0,0,1,0,0,1,1,0,1,1,1,1,0,0,1 |
| | | Selection of possible sampling rates for stream decoding |

Example of debug file:

| Time | Bus | Type | ID | Length | Status | Data |
|------|-----|------|-------|--------|--------|-------------------------|
| 111 | 2 | B | 0x181 | 8 | 0 | 00 00 48 A5 B5 62 01 14 |
| 231 | 2 | B | 0x7F6 | 8 | 0 | 00 00 19 97 00 00 02 4E |
| 346 | 2 | B | 0x161 | 8 | 0 | 67 23 AC D8 39 E9 5F 06 |
| 468 | 2 | B | 0x181 | 8 | 0 | 24 00 00 00 00 00 9C 18 |
| 800 | 2 | B | 0x181 | 8 | 0 | 00 00 00 00 00 00 00 00 |
| 1016 | 2 | B | 0x161 | 8 | 0 | 00 00 00 00 00 00 02 48 |
| 1145 | 2 | B | 0x181 | 8 | 0 | 00 00 00 00 00 00 98 BD |
| 1320 | 2 | B | 0x181 | 8 | 0 | FF FF 00 00 00 00 FF FF |
| 1497 | 2 | I | 193 | 8 | 0 | 00 00 00 00 00 00 78 02 |
| 1757 | 2 | B | 0x7F8 | 8 | 0 | 00 00 19 97 00 00 02 50 |
| 1879 | 2 | B | 0x7F6 | 8 | 0 | 79 9E 53 9B 3B BF 3D F7 |
| | | | | | | |

1.8 Playback functionality

With streamlogger playback functionality a streamlogger can send out (play back) all on CAN-busses received data from a previously done measurement.

This can be used to setup HIL (Hardware in the loop) tests and test and setup (2D-)dashboards with real race-track measurement data on desk!



Further Information

Any **Streamlogger** can be used for playback of data.

Also DashTFT can playback recording files.

The same Streamlogger does not have to be used for playback as for recording.

1.8.1 USB-Stick preparations

1. Open temporary directory of measurement to be played back



Further Information

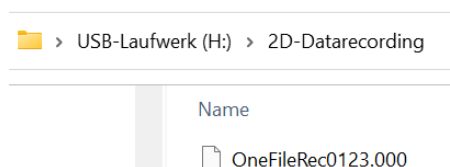
Temporary directory: ...\\TMP\ (WinARace → Settings → Folders – Protocol)

⇒ chapter 1.3

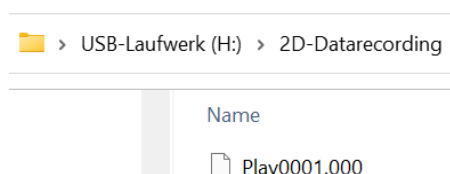
DL-folder: Open SpecSheet of .MES to be played back and see in SpecSheet group DOWNLOAD which OneFileRec____.DL folder belongs to which .MES

⇒ chapter 1.3.1

2. Copy OneFileRec____.000 to an arbitrary USB-Stick in folder 2D-Datarecording

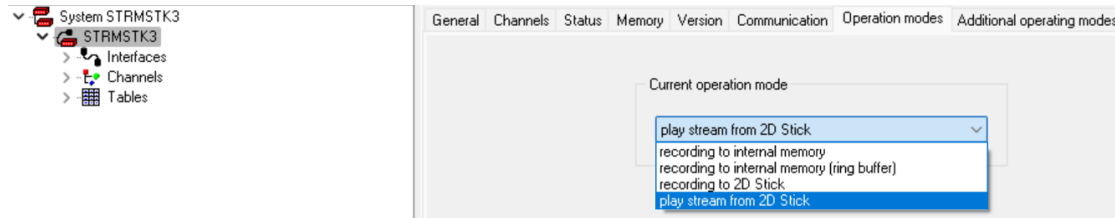


3. Rename OneFileRec.000 on Stick to Play0001.000



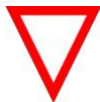
1.8.2 Streamlogger preparations

1. Read in streamlogger which should play back measurement data in Winlt
2. Navigate to the following tab and set record mode to *play stream from 2D stick*



DashTFT tab: *Additional operating modes* → *Logger record mode*

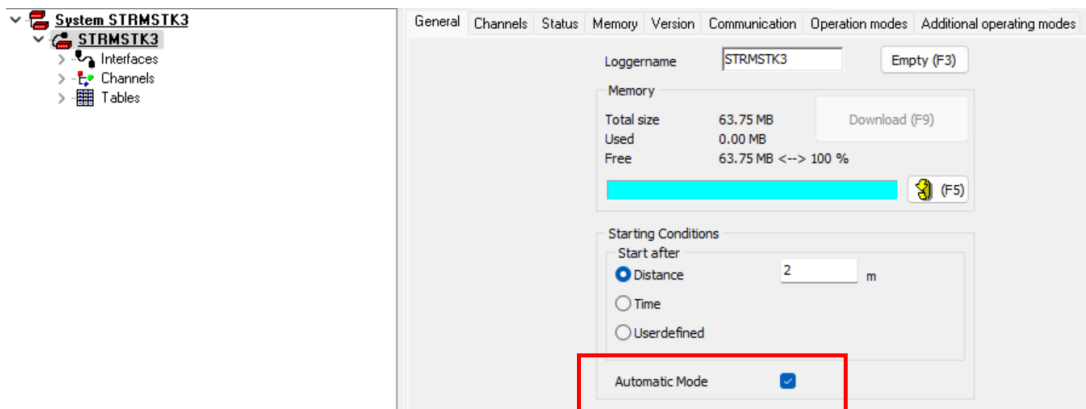
Logger tab: *Operation modes* → *Current operation mode*



Important information

When inserting 2D-USB-Stick at mode *play stream from 2D Stick* with normally for 2D-recording formatted Mess___.000 files the logger will start recording if starting condition is fulfilled!

3. With activated logging mode *Automatic Mode* the streamlogger plays back the Play0001.000 from file automatically again and again.
With deactivated logging mode *Automatic Mode* the Play0001.000 file is only replayed once. Then Stick has to be re-plugged or power cycle has to be applied to logger.



Further Information

Automatic Mode can only be activataed with *Distance* or *Userdefined* starting condition.

1.8.3 Start playback

1. Insert USB-Stick with Play0001.000 file in 2D-Datarecording folder. Logger will blink red (like at recording) when playback is active!