CANedge1 Intro and Tools

Release FW 01.04.01

Oct 12, 2021

CONTENTS

0.1 CANedge1 - get started

This guide provides a simple intro to each step of your workflow - including software & API tools.

0.1.1 About this manual

0.1.1.1 Purpose

The CANedge1 Intro focuses on the following:

- How to get started with the CANedge1
- How to use relevant software & API tools

The document is structured by the steps you go through when using the device for the first time.

0.1.1.2 Other documentation

The CANedge1 Docs and CANedge2 Docs serve as the product manual. Those docs detail the hardware, configuration and concepts beyond the scope of the CANedge Intro.

0.1.1.3 Notation used

The following notation is used throughout this documentation:

Admonitions

Note: Used to highlight supplementary information

Warning: Used if incorrect use may result in major loss of data and/or time

Danger: Used if incorrect use may result in personal injury or death

0.2 Configure your device

Below we outline how to update your device firmware and configuration.

0.2.1 Check for firmware updates

Before you start, we suggest that you check if a newer firmware exists. You can compare the `fw_ver` in your `device.json` file on the SD card vs. the latest firmware in the CANedge1 Docs.

0.2.2 Configure your CANedge

1. Extract the device SD card, insert it into your PC and open the config editor tool
2. Load your SD card `config-XX.YY.json` file in the editor via the Configuration File dropdown[1]
3. Select the CANedge1 Rule Schema via the editor sidebar dropdown
4. Modify your config, press "Review changes" and verify your edits
5. Download the Configuration File and copy/paste it to your SD card (overwrite the original file)
6. Safely eject the SD card and re-insert it into your CANedge1

The new Configuration File is now loaded by your device the next time it is powered on.

Note: Optionally open the online editor, right-click and "Save as. . . " to store the editor for offline use

Note: An invalid Configuration File will be defaulted by the device upon power up

Note: The UIschema lets you toggle advanced configuration settings on/off (off by default)

0.2.3 Configuration tools

0.2.3.1 Editor tools

For the best experience, we recommend to use one of the configuration editors below[1].

Simple config editor (offline/online)

The 'simple editor' lets you load and edit your Configuration File[2]:

Note: Optionally open the online editor, right-click and "Save as. . . " to store the editor for offline use

[1] The Rule Schema (`schema-XX.YY.json`) determines the structure of the editor, while the Configuration File (`config-XX.YY.json`) contains your actual configuration. The optional UIschema (`uischema-XX.YY.json`) determines the styling of the editor (e.g. for toggling advanced settings). For further details on the JSON Schema concept, see the CANedge Docs.

[1] The Configuration File is a JSON text file. This means you can in principle perform updates via any standard text editor. However, in doing so you will not benefit from the Rule Schema, which ensures that you perform valid edits. For most purposes, we therefore recommend to use a Schema based editor tool.

[2] The CANedge browser tools like the configuration editors (online/offline) work on modern browser like Chrome, Firefox and Internet Edge, but not Internet Explorer.

0.2.3.2 Using the config editor

Below we provide details on how to use the config editor.

Documentation

We strongly recommend that you review the Configuration section of the CANedge docs. This explains how the JSON Schema concept works incl. the role of the Configuration File, Rule Schema and UIschema. Further, it provides detailed examples for some of the more advanced configuration settings like CAN ID filters.

Presentation mode - simple vs. advanced

The editor tools will by default hide advanced settings for simplicity. To show all the available settings, you can switch the Presentation Mode in the sidebar.

Support tools

The editors add a number of configuration tools in the bottom toolbar:

- **Encryption tool**: This tool helps you encrypt passwords. For details see the CANedge Docs and the encryption tool section. For batch encryption, see the OTA batch manager section
- **Filter checker**: When setting up CAN ID filters, this tool can help evaluate if a given CAN ID will pass through the filter or not. It also provides guidance for setting up J1939 PGN filters
- **Partial config loader**: This lets you load, schema-validate and merge a partial Configuration File (e.g. a transmit list) into your active Configuration File
- **Bit timing calculator**: You can use the "Bit-timing (advanced)" mode to set a custom bit rate for your application and this calculator can be useful in checking your settings
- **Schema & config loader**: This will be open by default when using the editor and lets you load the UIschema, Rule Schema and Configuration File for use in configuration of the device

New major/minor Firmware

If you need to update to a new major/minor CANedge Firmware (e.g. from `00.07.04` to `01.02.04`), you will need to add a matching Configuration File to the device SD card. You can use the config editor to help update an existing Configuration File to a new Firmware structure:

1. Download the new Firmware zip from the CANedge docs
2. Load the new Rule Schema via the editor sidebar Rule Schema dropdown
3. Load your old Configuration File via the editor sidebar Configuration File dropdown
4. Perform updates if needed to ensure validity with the new Rule Schema
5. Review and download your new Configuration File

For details on how to update the Firmware, see the CANedge docs.

0.2.3.3 Encryption tool

The CANedge supports encryption of passwords - see the CANedge Docs for details.

Below we outline how to easily encrypt fields using the encryption tool within the config editor.

Generating keys & encrypting plain text data

Within the config editor, click the "Encryption tool" (lock icon) to open the tool.

In the tool, paste the public key (`kpub`) value from your `device.json` file and click "Create keys"[3].

This produces two keys:

1. **Server public key**: Should be added in the security section of the device Configuration File
2. **Encryption key**: Used for the encryption of plain text data (e.g. passwords)

You are now able to encrypt plain text data as follows:

1. Enter a plain text password and click "Encrypt"
2. Copy the encrypted password to the relevant field in the Configuration File
3. Set the corresponding key format to "Encrypted" in the Configuration File

Re-using an encryption key

You can securely store the encryption key for later use. This lets you use the second mode of the encryption tool to encrypt fields using an existing key.

This way you can later update/add passwords without changing the server public key or any pre-encrypted data in your Configuration File.

[3] You can extract the `kpub` from the `device.json` file on the device SD card

0.3 Record raw data

Below we outline how to record raw data to the device SD card.

0.3.1 Preparation

Before you connect your device, it is important that you do the following:

1. Read the introduction and hardware installation guide in the CANedge1 Docs
2. Verify that the pin-out of your application, adapter cable & CANedge match

0.3.2 Verify that you can log data

1. Connect & power the CANedge in your application via Channel 1 (green LED lights up)
2. Verify that the device records data to the SD card (yellow & red LEDs blink)
3. Disconnect the device, extract the SD and confirm that the `LOG/` folder now contains data[1]

If you're logging data from cars or heavy-duty vehicles, see our OBD2/J1939 sections. If you're having trouble logging data, see our troubleshooting section.

Note: Once you're done testing we recommend that you configure filters, prescalers & compression to reduce your file size (often by 90%+) - see also our tips & tricks section

See the online documentation for more details on logging OBD2 and J1939 data

[1] It is important that you disconnect your device before extracting the SD card to avoid SD card corruption. Similarly, when ejecting the SD card from your PC, make sure to use the 'safe eject' functionality

0.4 Transfer data

0.4.1 Transfer data via SD card

When you're ready to process your data, you'll need to transfer it from the device SD card:

1. Disconnect the logger from your application (the CANedge is 100% power safe)
2. Extract the SD card, insert it into your PC and open the `LOG/` folder
3. Transfer the log files you wish to process to your PC

0.5 Process your MDF4 data

The CANedge logs raw data in the popular MDF4 (`.MF4`) format - supported by many CAN tools.

In this section we outline some useful tools for processing your raw data:

1. *MF4 converters* - easily convert MF4 data to other formats (e.g. `csv`, `asc`, `trc`)
2. *asammdf GUI* - load, edit, DBC convert and plot your MDF4 data
3. *Python API* - automate your data processing at scale
4. Browser dashboards - visualize your data in customizable browser dashboards

You can of course also process your data in other tools, e.g. MATLAB.

0.5.1 MDF4 converters

Feature Intro

The below open source C++ executables let you easily convert raw MDF4 files into other formats.

If you're using data compression/encryption, the converters also **decompress/decrypt** your data.

Simply drag & drop files/folders (incl. nested) onto a converter - or use it in your CLI/scripts.

0.5.1.1 Download & source code

The Linux/Windows builds and source code can be found below:

Source code | README

0.5.1.2 Converter types

mdf2finalized

The CANedge records raw data as 'unsorted' and 'unfinalized'. This ensures performance and power safety. Some tools natively support unfinalized/unsorted MF4 (e.g. asammdf and our Python API), while others require that you use the `mdf2finalized` converter first (see below).

- **Vector CANalyzer** supports only finalized & sorted MF4 files, hence the mdf2finalized converter can be used to make your log files compatible (as of Vector's SP2 update)[1]
- **MATLAB's Vehicle Network Toolbox** (VNT) supports unfinalized MF4 files[2]. However, some advanced VNT use cases require that the data is already finalized & sorted (e.g. MF4 data stores). See also our MATLAB sample script

mdf2asc

Vector's ASC format is supported by various CAN tools, e.g. CANalyzer and CANape. The `mdf2asc.exe` lets you easily convert files into the ASC format for loading in such tools. With the latest SP2 for Vector's tools, you can also simply finalize the MF4 files (see above).

[1] The `mdf2finalized` tool can be used if you're using the latest SP2 for your Vector tools. If you're using older versions of Vector tools, we recommend using the `mdf2asc` converter instead as older versions of Vector's tools do not support the extended CAN ID syntax used in the CANedge MDF4 files

[2] MATLAB's Vehicle Network Toolbox supports unfinalized MF4 files as of release 2021b. In short, the tool lets you finalize/sort files as part of your script, rather than e.g. using the `mdf2finalized` converter

mdf2csv

The `mdf2csv.exe` enables quick conversion of the raw MDF4 data into a simple CSV format that you can load in text editors, Excel and other tools.

mdf2peak

The `mdf2peak.exe` lets you convert your MDF4 data to PEAK's TRC format, for use in e.g. PCAN Explorer and other PEAK tools. Default output is the `2.1` format, but you can specify `-f 1.1` via the command line.

mdf2pcap

The `mdf2pcap.exe` lets you convert raw MDF4 files into pcap format, for loading in Wireshark. Wireshark offers a range of powerful filter/analysis tools and can handle large log files seamlessly. Further, you can utilize our Wireshark plugin to e.g. convert OBD2/DBC data as in asammdf.

mdf2clx000

If you wish to convert your CANedge MDF4 log files to the CLX000 log file format, this converter lets you do that. When it runs, the converter reads an INI file in the same folder (mdf2clx000_config.ini) that follows the rules used in CLX000 configuration files. This means that you can use the settings in your old CLX000 configuration files to ensure that the output from the CANedge matches your preferred format. You can directly copy the relevant lines from the `[log]` section in your CLX000 `CONFIG.INI` file (removing any comments).

mdf2socketcan

The popular socketCAN format is supported by various open source CAN based software. With the `mdf2socketcan.exe` you can convert raw MDF4 log files for easy loading in these tools.

0.5.1.3 Encryption & compression

If you're encrypting or compressing your CANedge data, the converters will serve as a simple method for decrypting and/or decompressing the data again. All converters will natively recognize if your file is encrypted/compressed.

Note that if you have encrypted your files, you will need to add your plain form encryption password in the `passwords.json` file. You can add a single password as `default`, or e.g. add a list of device specific passwords by entering the serial number and the password as below:

```
{
    "0245BF81": "MySecret22BczPassword1234@482",
    "13FC798A": "MyOtherSecretPassword512312zZ"
}
```

0.5.1.4 CLI options

To use a converter via the CLI type the name in the command prompt to display the options.

Example: Convert a data folder (e.g. `log/`) into an output folder (e.g. `output/`) via below:

```
mdf2csv -I log -O output
```

Note: For examples on automating the converter usage, see the API examples library on github

0.5.2 asammdf GUI

Feature Intro

The asammdf GUI lets you easily load, review, DBC convert, plot and export your CANedge data.

Simply download and open the tool to use it (**no installation required**):

You can also install asammdf as below (reducing the load time for opening the GUI by 80%):

1. Install Python 3.7 for Windows (64 bit) or Linux (add to PATH)
2. Open your command prompt and write: `pip install asammdf[gui]`
3. Open your start menu, write 'asammdf' and open the GUI via the asammdf icon

0.5.2.1 Load raw data

You can directly load your raw MDF4 log file data from the CANedge in asammdf.

Loading your MDF4 data

To open a single file, click "File/Open" and browse to your `.MF4` log file.

To open multiple files (e.g. for concatenation), first click "Mode/Batch processing".

Note: If your data is compressed/encrypted (`.MFC/.MFE/.MFM`), use *mdf2finalized* to convert it to `.MF4`

Review your raw data

Once you've loaded a raw MDF4, you'll see an overview of the file in the **Channels** tab. Channel group 0 contains CAN data, while remaining channel groups contain e.g. LIN data, RTR frames etc.[1].

If you select a channel group and click the plot icon, you can display the data in a tabular form. Here you can quickly filter and analyze your data, both with relative and absolute timestamps. You can also use e.g. the CAN Trace or LIN Trace views to show multiple channel groups in one tabular display.

0.5.2.2 DBC conversion

To analyse your data, you'll need to convert it to human-readable (aka physical) form. To do so, you'll need a DBC file (CAN Database) with the decoding rules.

Converting a raw MDF4 with a DBC file

1. Select the "Bus Logging" tab, click "Load CAN database" and load your DBC
2. Click "Extract CAN signals" to save a new MDF4[1]

[1] See the CANedge Docs for details on the MDF4 log file structure and the role of each column

[1] You can optionally enable 'Ignore invalid signals', which is useful for J1939 data as it removes signals that are not actually containing valid data. For J1939/NMEA2000 data you may also consider disabling the 'Consolidated J1939' setting. By disabling this, CAN IDs that share the same PGN are no longer bundled, but are instead separated in the signal output.

The resulting MDF4 will be opened as a new tab in the GUI - ready for e.g. plotting.

J1939 & OBD2 DBC files

Usually, you'll need to be an OEM to have access to a full DBC file detailing the data parameters of a specific application, though exceptions exist:

J1939

Most heavy duty vehicles today use the standardized J1939 protocol. This means that you can typically use a J1939 DBC to decode a large share of signals across vehicle brands. We offer a *demo J1939 DBC* and a full J1939 DBC.

OBD2

Most cars let you request OBD2 PID data, which can be decoded using our free *OBD2 DBC*.

0.5.2.3 Graphical plots

Once you've converted your raw MDF4 data, you can start analysing it - e.g. via plots.

Plotting parameters

1. In the Settings tab (top menu), enable sub-plots
2. From Channels, drag & drop a parameter into the gray area to plot it
3. Optionally click the "window" icon and add more parameters to the gray area
4. Press "Shift + V" to tile the sub-plots vertically

0.5.2.4 Export

See the online documentation for details on exporting MDF4 files via asammdf.

0.5.3 Python API tools

The Python API tools let you easily automate and scale your CANedge data processing.

0.5.3.1 API modules overview

The three tools below enable most data processing use cases:

- `canedge_browser`: List log files for selected devices & time periods (from local disk or S3)
- `mdf_iter`: Extract raw CAN data from the CANedge log files (as iterable or dataframe)
- `can_decoder`: DBC-decode raw CAN data to physical values

0.5.3.2 Get started

To get started with the API tools, check out the API examples library on github.

In particular, the `data-processing/` examples show how to combine the data processing modules.