CLX000 Intro

Release FW 5.85

CSS Electronics

Mar 23, 2023

CONTENTS

0.1 CLX000 - get started

This guide provides a simple intro to each step of your workflow - incl. key software tools.

0.1.1 About this manual

0.1.1.1 Purpose

The CLX000 Intro focuses on the following:

- How to get started with the CLX000
- How to use relevant software 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 CLX000 Docs serve as the product manual. The Docs detail the hardware, configuration and concepts beyond the scope of the CLX000 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

0.2.1 Configure device

Below we describe how to configure your CLX000:

1. Use a mini USB adapter to connect the CLX000 to your PC
2. Verify that the green + red LED light up (see the LED video)
3. Download and open CANvas, go to 'Configure logger' and select your logger model number
4. Make a basic change - e.g. change the file split size to 10 MB
5. Press 'Export' and save the `CONFIG.INI` file to the device SD card (overwriting the original file)
6. Safely eject the device e.g. using the 'Eject logger' button in CANvas

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

For details on each configuration field, see the CLX000 Docs.

Note: If your Configuration File contains invalid data (or is deleted from the SD), the device will create a default Configuration File to replace it once powered on

Note: You can also open the `CONFIG.INI` file directly via a text editor to edit the fields

Note: While you can modify the log file format (delimiters, fields etc), we recommend keeping the default settings to ensure compatibility with CANvas for processing the data

0.2.2 Update Firmware

In some cases you may want to update the Firmware of the CLX000 device. Below we describe the steps in brief.

1. First, check your current Firmware by opening a log file and checking the `FW rev`
2. Open the CLX000 Firmware page and download the latest (if it is newer than your current)
3. Follow the Firmware Upgrade steps of the CLX000 Docs section 7 (p 13)

When updating, make sure that you do not put a wrong Firmware on your device (e.g. mixing a CL1000 and CL2000). Further, on some PCs, you may need to update your DFU drivers[1].

Note: You can optionally use CANvas to support in the firmware update process

[1] You can download the DFU drivers here. To install them, open your Device Manager and locate the CLX000 port. Right-click the port, choose to install/update the driver software and browse to the `dfu/` folder of the driver zip.

0.3 Record raw data

0.3.1 Preparation

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

1. Read the introduction and installation guide in the CLX000 Docs
2. Verify that the pin-out of your application, adapter cable & CLX000 match

0.3.2 Verify that you can log data

1. Connect & power the CLX000 in your application via the DB9 connector (green LED lights up)
2. Verify that the device records data to the SD card (yellow & red LEDs blink)
3. Disconnect the device, connect it via USB to your PC and confirm that the SD contains data

Before you deploy your unit in the field, we strongly recommend to review our tips & tricks. If you're having trouble logging data, see our troubleshooting section.

0.3.3 Logging OBD2 data

In this section we outline how to log OBD2 data with your CLX000.

0.3.3.1 Configure your device

First you configure your device to transmit OBD2 'request messages':

1. In the 'CAN-bus' section, disable silent mode and set the bit-rate to 500K (for cars)
2. In the 'Transmit list' section, click 'OBD' to add up to 20 PIDs to be requested by the device[1]
3. Optionally adjust the request period/delay or add custom OBD2 PID requests
4. Export the configuration file to your device SD card

0.3.3.2 Record data from your car

1. Connect the CLX000 to the OBD2 connector in your car via the DB9-OBD2 adapter[2]
2. Verify that the device turns on and logs data (see the LED section of the CLX000 Docs)
3. Disconnect the device and review the `TXT` log file via a text editor
4. If your car responds[3] you should see CAN frames with ID `7E8`[4] in your data
5. Once confirmed, you can optionally optimize your Configuration File[5]

[1] Note that the default list in CANvas matches most of the Mode 01 PIDs, but you can manually add custom OBD2 PIDs. Further, if your vehicle uses extended IDs (e.g. for light trucks), you may need to use the request ID `18DB33F1`. The responses in this case may come on `18DAF110`

[2] We recommend using one of our DB9-OBD2 adapters. If using a 3rd party cable, it's important to verify the pin-out.

[3] Note that some older cars do not support OBD2 data acquisition via CAN bus, while some newer cars block CAN access via the OBD2 connector. For cars that do support OBD2 data, the extent of coverage varies. The "supported PID" single-shot requests can help provide information on what PIDs are supported. If your car does not respond to the OBD2 requests, we recommend to test in other cars to determine if the issue is specific to the car or e.g. the Configuration File

[4] In some cases you'll see IDs like `7E9`. In this case, you may need to modify the OBD2 conversion database to use alternative CAN IDs

[5] For example, you may want to add filters to only record OBD2 responses. Also, you may want to add custom OBD2 PID requests - see our simple intro to OBD2 and the OBD2 PID Wikipedia page for details on this

0.3.3.3 Analyze & plot your OBD2 data

The *CANvas conversion intro* details how you can convert raw data from the CLX000 into human-readable form - and e.g. export it as CSV. You can decode raw OBD2 data with the built-in OBD2 database in CANvas (see the black database icon) in the 'Create/load database' section.

0.3.3.4 OBD2 & battery consumption

The CLX000 consumes <1W, which is not an issue for your car battery in practical use cases. In most cases, the device also turns off with your car (or 10-20 min after). However, if this is not the case and you're requesting OBD2 data, the device may "wake up" the car sensors.

In such scenarios, there are a couple of options:

1. You can simply disconnect the device between trips
2. You can re-wire your vehicle's OBD2 connector so that the power pin is linked to the ignition
3. If your use case requires a dynamic toggling of the OBD2 requests, consider the CANedge

0.4 Transfer data

0.4.1 Transfer data from SD card via USB

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 CLX000 is 100% power safe)
2. Connect the device via USB to your PC to view the SD contents
3. Transfer `TXT` log files to your PC for processing

0.4.2 Transfer data via WiFi

In addition to SD card transfer, the CL3000 lets you *access data via WiFi*.

0.4.2.1 WiFi (CL3000)

CL3000 WiFi modes

The CL3000 supports three WiFi modes for remote log file access.

1. Access point: Use e.g. a laptop to connect to the CL3000 like a WiFi access point (AP)
2. Station: CL3000 connects to WiFi AP making the SD accessible by devices on the network
3. Push: CL3000 connects to WiFi AP and pushes log files to an FTP server

By default, the CL3000 will operate in access point mode. You can change mode and other WiFi settings via the `CONFIG.INI` directly or via CANvas. For details, see the CLX000 Docs.

Access point and station mode

In access point and station mode, the CL3000 lets you access the SD card from your browser by entering the device IP or hostname (e.g. `id0001/`). Here, you can download, delete & split log files. Note that on some networks, you need to use the IP rather than hostname. Note also that the web user interface will only show up to ~500 log files (oldest files shown first).

Push mode

The push mode lets you auto-upload data to an FTP server. For example, you can set up an Exavault cloud FTP server or use your own local/dedicated FTP server.

Note: If you aim to use the 'push' mode at larger scale, we recommend to check out the CANedge2, the 2nd generation of the CL3000

WiFi tips & tricks

Below we list a few tips when using the CL3000:

1. Use the CANvas 'test' buttons to verify your WiFi/FTP details before testing via the CL3000
2. If uploading data via push mode, use a small split size (e.g. 1-10 MB) for smooth uploads
3. If connecting multiple devices in station mode to the same WiFi AP use *unique hostnames*
4. To auto-fetch log files from station mode CL3000 units, see for inspiration this `Ruby script`
5. To access a CL3000 in station mode outside the WiFi network, you can port forward it
6. For push mode local/dedicated servers, ensure that no firewalls block the access to the device

0.5 Process your log file data

The CLX000 logs raw CAN data in a simple `TXT` format.

In this section we outline useful tools for viewing/decoding your raw data.

1. *CANvas* - load multiple raw log files, DBC decode to physical values and export to CSV
2. *SavvyCAN* - load/stream raw CAN data, DBC decode it, create plots and more

Note: CANvas is ideal for processing multiple log files. SavvyCAN is recommended for analyzing single log files and/or streaming CAN data via USB[1]

Note: You can also simply load the raw `TXT` log files in e.g. custom scripts, Excel, MATLAB etc.

Note: CANvas refers to Wireshark in the streaming section. This is in principle still possible to use, but we have since migrated to SavvyCAN which we consider superior

Sample raw CAN data & DBC files

0.5.1 CANvas

Feature Intro

CANvas lets you load raw CLX000 log files, decode the data to physical values and export it.

0.5.1.1 Installation

Download & unzip, then open via the CANvas.exe[1]:

0.5.1.2 Converting raw CAN data to physical values

To decode raw CAN data, you need a database with the conversion rules - see our CAN bus intro.

CANvas supports basic DBC files for loading your decoding rules[2].

Below are the steps to convert log files via DBC files:

1. Open CANvas and go to 'Convert data', click 'DBC' and load your DBC file[3]
2. Next, load your log file(s) and then select the 'Scaled engineering values' output
3. Click the 'Output' tab and the green button to start the conversion
4. The converted data can be loaded in 3rd party tools (e.g. Excel, scripts) for plots/analysis

[1] We recently added support for SavvyCAN, for loading log files and streaming data. We believe SavvyCAN is superior to Wireshark for most use cases. As such, Wireshark is primarily recommended if you are familiar with the tool already.

[1] CANvas is Windows-only. For Linux you can consider using *SavvyCAN* - or process the log files directly in e.g. scripts

[2] You can in principle create a decoding database directly in CANvas using the GUI editor. However, we strongly recommend that you use the DBC file format if possible when creating a new database. The DBC format is also supported in SavvyCAN, the CANedge software tools and most other CAN software available. Various free DBC editors exist, including one in SavvyCAN. See our DBC intro for details. Note that some DBC syntax features are not supported by CANvas, e.g. multiplexed messages and transport protocol decoding. The CANedge software tools have more extensive DBC support

[3] If loading a J1939 DBC, make sure to select the J1939 matching criteria. For OBD2 decoding, note that you should use the built-in OBD2 database instead of loading a DBC file

Note: You can modify settings related to the output, e.g. file merging, downsampling or timestamps

0.5.1.3 Converting raw log files to other formats

To convert your log files to other formats (`ASC` or `TRC`), load the log files and select the 'Vector CANalyzer' or 'PEAK' output options[4].

0.5.1.4 Troubleshooting

If CANvas returns only empty lines, it typically reflects that there is not a match between your log file data and the conversion rule database. Review your raw log file to see if the CAN IDs match your database entries.

0.5.2 SavvyCAN

Feature Intro

SavvyCAN is ideal for analyzing raw and/or DBC decoded CAN bus data.

You can load the raw CLX000 log files (`TXT`) or *stream CAN data* via USB.

0.5.2.1 Installation

Windows: Download below build (which includes the CLX000 plugin)[1].

Linux: Download below build and copy the Linux plugin into the `canbus/` folder.

docs | 30 min intro video | CLX000 plugin source code

0.5.2.2 Stream data via the CLX000

1. Setup the CLX000 to stream CAN data (see the *Stream* section)
2. Click 'Connection/Open Connection Window/Add New Device Connection'
3. Select 'QT SerialBus Devices/csscan'
4. Click 'Create New Connection' - the raw data stream should now appear in the main window[2]

[4] Note that you can also save to Vector ASC or PEAK TRC via SavvyCAN, though SavvyCAN does not have a feature for converting a batch of log files

[1] The CLX000 plugin allows for streaming the CLX000 data into SavvyCAN. To use the latest release of SavvyCAN, simply download it and add the CLX000 plugin to the `canbus/` folder manually on either Windows or Linux. For convenience, our Windows download link has the plugin pre-included. Note that if your PC does not have Visual Studio redistributables installed, you may get a `vcruntime140_1.dll was not found` error. In this case, use the `vc_redist.x64.exe` to install the redistributables

[2] If the data does not appear, ensure that your device is correctly set up for streaming data as per the *Stream* guide

0.5.2.3 Load log file and DBC file(s)

To load a raw `TXT` log, click File/Load Log File, select the CLX000 format and load your file[3].

To load a DBC file, click File/DBC File Manager/Load and set the ID match criteria (`Exact/J1939`).

0.5.2.4 Main window

Below we outline a number of key functionalities of the main window:

- Auto Scroll Window: Toggle if the window should scroll with the data (streaming only)
- Overwrite Mode: Show only a single line for each unique ID (with the latest frame information)
- Interpret Frames: Use the loaded DBC to decode matched frames (see the Data column)
- Frame Filtering: Show/hide specific CAN IDs

0.5.2.5 Useful analysis tools

Below we outline some of the 'RE Tools' that can be useful for analysis or reverse engineering. For a more detailed documentation, see the above links for the official docs and intro video.

Graph Data

This lets you plot specific CAN signals either real-time (for streamed data) or based on a log file. Simply right click the plot and select 'Add new graph'. You can select a signal based on your DBC file, or you can manually add one[4]. You can also right-click the graph to reset the view and/or follow the end of the graph (for streaming).

Sniffer

This tool shows you the unique CAN IDs and helps identify changing bits/bytes - e.g. in response to physical events you perform while streaming data. It is ideal for reverse engineering state parameters (open/closed, on/off, ...).

[3] For some CLX000 log file formats, the SavvyCAN file format 'auto-detect' feature may work, in which case you can simply directly load the log files without first selecting the CLX000 format via the dropdown.

[4] Note that for J1939 signals, you'll need to manually overwrite the CAN ID to ensure it matches the exact 29-bit CAN ID in your data. The Graph Data does not currently implement the J1939 match type that is used in the main window. Further, for OBD2 data the plot tool is not ideal as it does not yet support multiplexing - as such it will only yield useful plots if your data contains only a single OBD2 PID

Range State

This is a great tool for reverse engineering continuous parameters like speed, RPM, SoC% etc. You add your hypothesis on the signal details (CAN ID(s), length, ...) and press 'Recalculate Candidate Signals'. The tool then produces plots for all possible signals that match your criteria[5].

Flow View

This view lets you control the playback of a log file (or an active CAN stream) to analyze changes to bits and bytes. Bits will be colored based on changes vs. the start of the session/log or vs. the previous frame (toggle via 'Auto Reference'). The tool is useful for e.g. reverse engineering if you're looking for correlating specific events or patterns.

Frame Data Analysis

This lets you analyze the CAN frames in your data using histograms, e.g. to understand the distribution of specific data bytes throughout a log file.

0.5.2.6 Send frames via the CLX000

As of Firmware 5.85, you can freely control the CLX000 CAN frame transmit functionality in real-time. This can be done via custom script integrations (see the CLX000 Docs), or via the SavvyCAN 'Send Frames' tools. Below we describe the relevant tools in brief.

Danger: Transmitting data via CAN bus can be dangerous. Do not transmit data into your CAN bus unless your setup is safe and you are 100% sure that you know what you are doing. This is particularly so for the Playback and Fuzzing tools.

Custom frame sender

This tool lets you specify one or more custom CAN frames to be transmitted by the CLX000. You can fully control the ID, payload, transmit frequency - and even apply advanced functions.

Detailed docs (Custom Window)

[5] The tool plots the decimal values of the extracted bits based on your criteria (using an offset of 0 and a scale factor of 1). To get the final signal, you'll need to identify how to offset/scale the chart, though this is typically a fairly simple task. The tool is ideal for post processing - e.g. if you've logged data during a trip while you've made timestamped notes of the physically observable values (e.g. speed in km/h). Your notes can be used to produce a 'rough graph', which can then be compared vs the suggested signal plots

Playback

The Playback window lets you load a CAN bus log file (e.g. from your CLX000) - and replay it back onto a CAN bus. This can be useful in e.g. reverse engineering use cases or if you want to use the CLX000 as a simulation device. You can use the 'original frame timing' when playing it back and optionally filter the data to only replay a subset of the CAN frames.

Detailed docs (Playback Window)

Fuzzing

The fuzzing window can be used to transmit "controlled randomness" onto the CAN bus. We mainly recommend using this for simulation purposes or for highly controlled reverse engineering, as sending random CAN data into e.g. a vehicle is dangerous.

Detailed docs (Fuzzing Window)

0.6 Stream data in real-time via USB

The CLX000 can be used as a CAN interface to stream data in real-time via USB to your PC.

To stream CAN data via the CLX000, follow the below steps:

1. Power the CLX000 via the DB9 and verify that it is logging CAN data (red/yellow LED flashing)
2. Once verified, connect the CLX000 to your PC via USB
3. Follow the installation/streaming guide for *SavvyCAN*[1]

Warning: You will not be able to start the streaming if the CLX000 is stuck in auto-detect bit-rate mode. Make sure the device is able to log data before attempting to stream

Warning: We recommend using a laptop (disconnected from power) for streaming to avoid differences in ground voltage between the USB and DB9 connection (see the CLX000 Docs for details)

[1] We recently added support for streaming data into SavvyCAN, which we consider superior vs. Wireshark as it's easier to setup/use and provides more powerful features