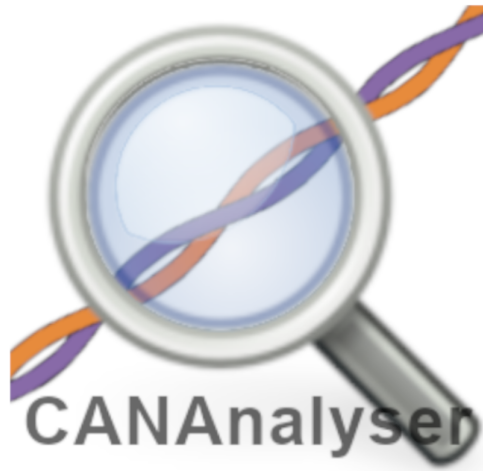
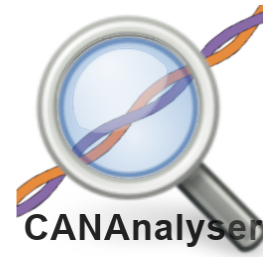




SLSS CANAnalyser Pro Software A2B Qwiic interface addendum

For use with Serosys Technologies hardware dongles

Document: V1.2.8.0 - October 2026



Users Manual

Based on software version 1.2.8.0L

The information contained in this Publication is Proprietary to SeRoSys Technology LLC and Sebastian Langer Software Solutions UG (haftungsbeschränkt)



Table of Contents

Users Manual	1
1. Logging Methods Overview	3
2. Common Information for Both Methods	3
2.1. Hardware Setup	3
2.2. Bus Monitor Status LEDs	7
2.2.1. Startup Check and Qwiic Cable Errors	8
2.3. SLSS CANAnalyser ID cross-reference to A2B EventID	9
2.4. A2B EventID logging legend	10
3. Method 1: Software-Connected Logging	11
3.1. Hardware Configuration	11
4. Method 2: Standalone A2B + CAN Logging	13
4.1. Hardware Configuration	13
4.1.1. Setting the Lua Script to Autostart	14
4.2. Standalone Logging Types	18
4.2.1. Constant Logging	18
4.2.2. Triggered Pre-Buffered Logging	20
4.2.3. Reconnecting to the Software for Post-Logging Analysis	22
4.3. Standalone Logging FAQ and Troubleshooting	23
4.3.1. Key Concepts	23
4.3.2. Frequently Asked Questions	24
4.3.3. Troubleshooting	27
4.3.4. Best Practices Checklist	29

1. Logging Methods Overview

The SeRoSys Lab CAN Pro dongle and the Flextech A2B Bus Monitor can be combined to capture A2B and CAN data together, with both kinds of messages synchronized on one timeline. Two methods are available. Choose the one that matches your test setup, then jump to the matching chapter.

	Method 1: Software-Connected Logging	Method 2: Standalone Logging
What it is	Synchronized A2B + CAN logging with the SeRoSys dongle and Flextech Bus Monitor connected to a PC running the SLSS CANAnalyser software.	Standalone A2B + CAN synchronized logging with no PC connected. Two types are available: constant logging of CAN traffic + A2B events, or triggered pre-buffered logging that also stores a matching audio WAV file on the Bus Monitor.
How the data is captured	Live in the software. A2B and CAN messages appear chronologically merged in the Incoming Data and Logfile Recorder tabs.	Recorded by the devices themselves. Constant logging records everything. Triggered pre-buffered logging saves the last (x) seconds / minutes when an external trigger occurs. Logs are retrieved later in the software once the SeRoSys dongle is connected.
Jump to	Chapter: Method 1	Chapter: Method 2

2. Common Information for Both Methods

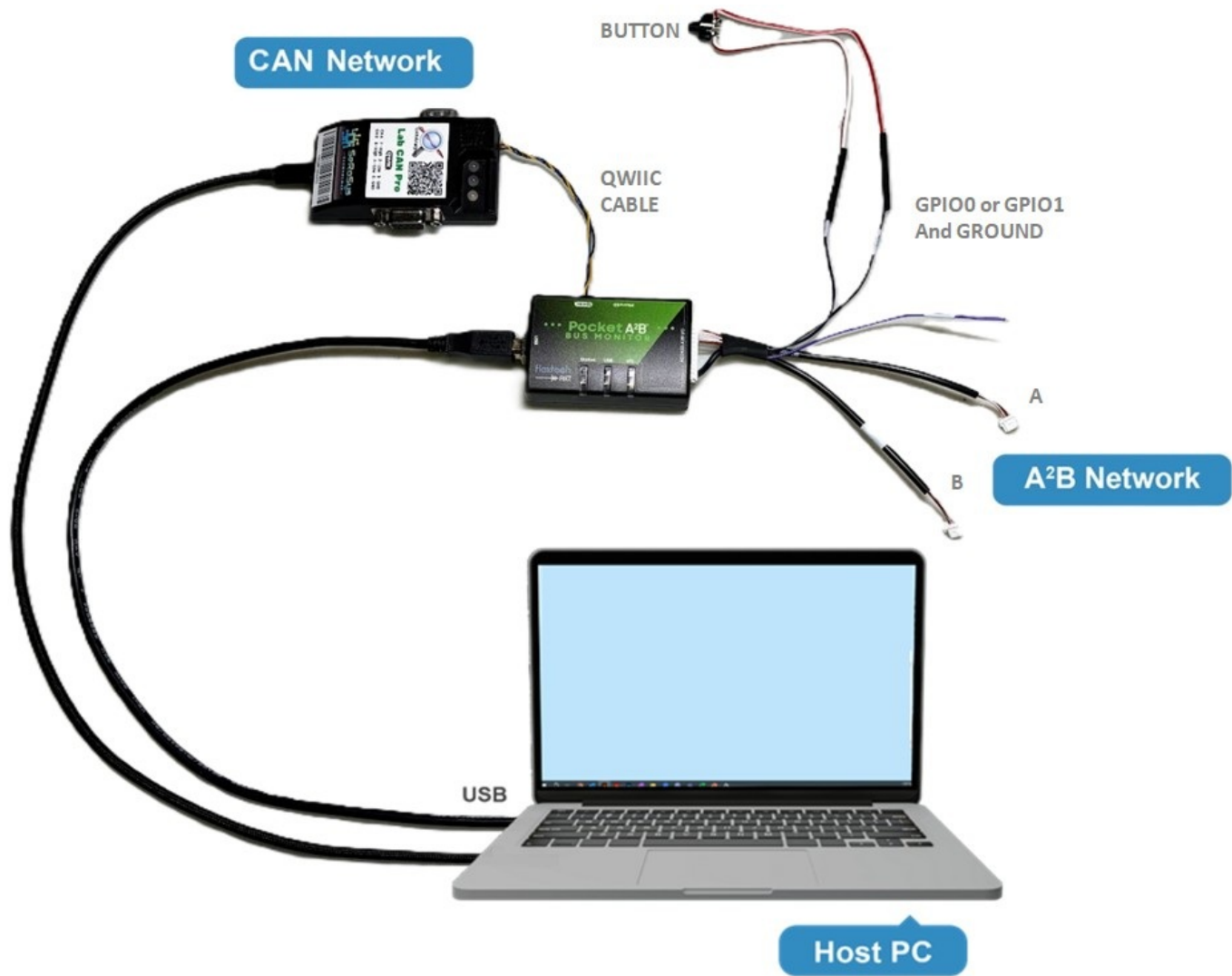
The topics in this chapter apply to both Method 1 and Method 2.

2.1. Hardware Setup

This is the hardware interface for the SeRoSys dongles to work in parallel with the Pocket A2B Bus Monitor products by Flextech AKT.



Ensure that USB cables are not too long to ensure a minimal voltage loss to keep the power at 5V and a minimum of 150mA of current for proper operation.



For logging while connected to the SLSS CANAnalyser software, follow the diagram for the 2 USB cables to be connected to the Host PC.

For standalone logging, the Host PC is omitted and the 2 USB cables are connected to standard power capable USB ports that provide standard adequate 5V with proper current capability.

If buffered logging is not required, the Button connected to the GPIO0 / GPIO1 and GROUND pins on the Flextech Bus Monitor (not on the SeRoSys dongle) may be omitted.

For details on proper A2B network connections via the A and B connectors, please consult the proper Flextech Bus Monitor user's manual.

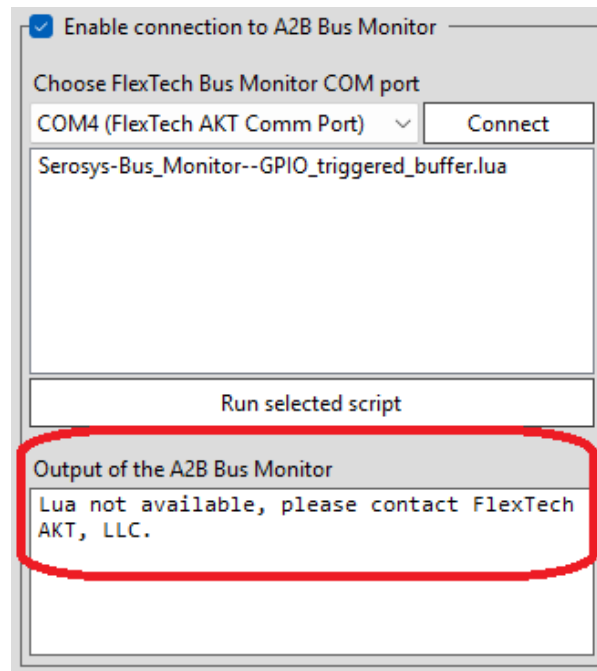
This interface works on both Flextech A2B Bus Monitor versions which support A2B 1.0 or A2B 2.0.

<https://flextechakt.com/products/a2b-pocket-bus-monitor>

<https://flextechakt.com/products/a2b-2-0-bus-monitor>



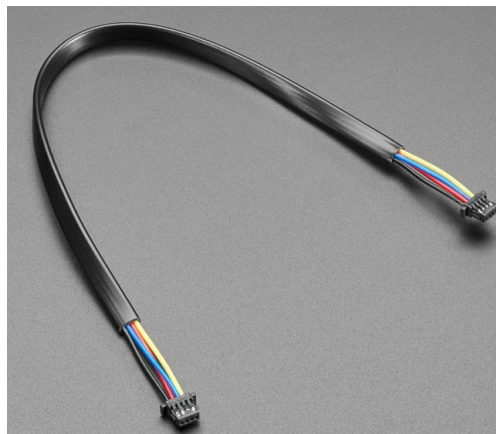
To support this interface, users must add "AKT Automation power by Lua" to the purchase of the Flextech A2B Bus Monitor. When connecting to the Bus Monitor in the Methods below, if the "Output of the A2B Bus Monitor" window shows this information in the text box window then that means that the Lua support was not purchased with the Flextech device and Flextech AKT must be contacted for unlocking Lua access on the A2B Bus Monitor.



A Qwiic cable is required to connect the SeRoSys products to the Flextech products. This is available to purchase as an add-on from our purchase page here: <https://serosys-tech.com/product/qwiic-cable/>



A standard Qwiic cable purchased online contains 4 wires. If purchased separately, you **MUST** cut the red wire from this cable for usage with this setup to avoid power backfeed between the 2 devices.





This interface requires a Lua script to be copied on to the microSD card that is provided with the Flextech Pocket A2B Bus Monitor. The Lua script may be downloaded from our downloads page (under Scripts and Automation) at <https://serosys-tech.com/downloads/>

Upon downloading the Lua file. Please copy the Lua file to the microSD card and insert the card into the Flextech A2B Bus Monitor. **This Lua file is NOT for the microSD card in the SeRoSys dongle**

2.2. Bus Monitor Status LEDs



LED / State	Meaning
Status LED: solid green	Everything is working and no errors have been detected.
Status LED: blinking red	Error state. An error was detected in one of the major operations.
Status LED: blinking red for about 5 seconds, followed by a restart of the Bus Monitor	The built-in startup check of the Lua script could not communicate over the Qwiic (I2C) cable. Whether the Bus Monitor retries afterward depends on how the Lua script was started. See Startup Check and Qwiic Cable Errors below.
I/O LED: on (red) for 1 second, then off	The GPIO trigger was received (the button was pressed).

2.2.1. Startup Check and Qwiic Cable Errors

When the Lua script starts, it performs a built-in startup check. It tries to send an I2C message over the Qwiic cable to the SeRoSys hardware.

- **If this succeeds**, the Bus Monitor goes to operational mode. A solid green Status LED then means everything is working.
- **If this fails**, the Status LED blinks red for about 5 seconds to show that something is wrong. The Bus Monitor then performs a hard reset.

What happens after the hard reset depends on how the Lua script was started:

How the Lua script was started	Behavior after a failed startup check
Manually, by pressing Run selected script in the Hardware Configuration window of the SLSS CANAnalyser software	The Bus Monitor does not retry after the restart. After you resolve the Qwiic cable problem, you must open the Hardware Configuration in the SLSS CANAnalyser again, reconnect to the Bus Monitor, and run the Lua script again.
Automatically, using the Set as autostart button in the Hardware Configuration of the SLSS CANAnalyser software (see Setting the Lua Script to Autostart)	The cycle repeats indefinitely (5 seconds of red blinking, then a restart) until you resolve the Qwiic cable problem. Once it is resolved, the next automatic hard reset clears the error condition, and the Bus Monitor starts normally with the Status LED solid green.

If the Status LED blinks red for about 5 seconds and the Bus Monitor then restarts, check that the Qwiic cable is fully connected at both ends (the Bus Monitor and the SeRoSys dongle). You can simulate this error by unplugging the Qwiic cable. This makes it easy for a tester to confirm that everything is ready before starting a test.



In Standalone mode, power the SeRoSys dongle **FIRST**, and then the Bus Monitor. If the Bus Monitor is powered before the SeRoSys dongle, it is likely that the error condition would be seen once on the first detection and after the 5 red LED blinks and the Bus Monitor resets, the SeRoSys dongle would be powered up and ready to go and therefore the error condition would be cleared.



Solid green means everything is fine. When using triggered pre-buffered logging, if the I/O LED does not flash red for about one second when you press the button, the trigger did not reach the Bus Monitor and no triggered recording will be created for that press.

2.3. SLSS CANAnalyser ID cross-reference to A2B EventID



The ID column in the Incoming Data and the Logfile recorder needs to be set as the ID display format of HEX in order to properly see the 0xA2Bxxx ID's, otherwise they will be the DEC or Binary version and more difficult to use.

ID	EventID Message
0xA2B001	Bias OK
0xA2B002	Bias OK removed
0xA2B003	A2B Discovery
0xA2B004	A2B NEWSTRCT
0xA2B005	A2B RegisterAccess (I2C_REG)
0xA2B006	A2B Peripheral I2C Transaction (I2C_PERIPHERAL)
0xA2B007	A2B Downstream Frame Error (SCF)
0xA2B008	A2B Upstream Frame Error (SRF)
0xA2B009	A2B Slave Error
0xA2B010	A2B Interrupt (IRQ)
0xA2B011	A2B Sequence Error
0xA2B012	A2B SPI Transaction
0xA2B013	Bias reverse detected
0xA2B014	Bias reverse removed
0xA2B015	Bus locked
0xA2B016	Bus unlocked
0xA2B032	Bus Monitor GPIO trigger message (Designation shows the filename counter of the linked WAV file)
0xA2B4095	Buffer overflow error

2.4. A2B EventID logging legend

A2B EventID logging structure will work the same as with the CAN bus monitoring. A2B messages received by the A2B Bus Monitor can be seen on the Incoming Data tab and the Logfile Recorder tab. The logs may be exported and analyzed offline or additionally imported into the Data Viewer.

Time	Δt Start [μs]	Δt Message [μs]	ID	Channel	Data type	Data length	Data	ASCII Text
23.01.2026 18:10:14.728	0	0	0xA2B001	CH_A	A2B	0		
23.01.2026 18:10:14.732	4915	4915	0xA2B015	CH_A	A2B	0		
23.01.2026 18:10:14.735	7991	3076	0xA2B003	CH_A	A2B	2	0x01 0x7B	
23.01.2026 18:10:14.736	8381	390	0xA2B007	CH_A	A2B	4	0x00 0x01 0x01 0x01	
23.01.2026 18:10:14.747	19943	11562	0xA2B003	CH_A	A2B	2	0x00 0x00	
23.01.2026 18:10:14.748	20359	416	0xA2B005	CH_A	A2B	4	0x00 0x02 0xAD x00	-
23.01.2026 18:10:14.749	21256	897	0xA2B005	CH_A	A2B	4	0x00 0x03 0x29 0x00	)
23.01.2026 18:10:14.749	21656	400	0xA2B005	CH_A	A2B	4	0x00 0x04 0x00 0x00	
23.01.2026 18:10:14.750	22050	394	0xA2B005	CH_A	A2B	4	0x00 0x05 0x01 0x00	
23.01.2026 18:10:14.750	22505	455	0xA2B005	CH_A	A2B	4	0x00 0x0B 0x80 0x00	
23.01.2026 18:10:14.750	22901	396	0xA2B005	CH_A	A2B	4	0x00 0x0C 0x02 0x00	♦
23.01.2026 18:10:14.751	23296	395	0xA2B005	CH_A	A2B	4	0x00 0x41 0xE0 0x00	Aâ
23.01.2026 18:10:14.751	23871	575	0xA2B005	CH_A	A2B	4	0x00 0x42 0x19 0x00	B
23.01.2026 18:10:14.752	24272	401	0xA2B005	CH_A	A2B	4	0x00 0x4A 0x10 0x00	J
23.01.2026 18:10:14.752	24670	398	0xA2B005	CH_A	A2B	4	0x00 0x4D 0x10 0x00	M



A2B messages will now be chronologically integrated with CAN messages. To identify them, the Data Type column will have a red **[A2B]** icon.

In the ID column uses the unique ID of 0xA2Bxxx where xxx is the A2B Event ID.

In the raw data, the various bytes are identified per the details above.

Incoming Data	Logfile Recorder [ID FILTER ACTIVATED]		Send CAN Messages	Send LIN Messages	Standalone Send Mode	Reverse Engineering	Incoming Database Signals	Send Database Signals	
Time	Δt Start [μs]	Δt Message [μs]	ID	Channel	Data type	Data length	Data	ASCII Text	Designation
13.02.2026 04:22:51.651	2431402	15549	0x156	CH_A	STD	8	0x5D 0xC4 0x00 0x00 0x03 0x00 0x00 0x00	J A	
13.02.2026 04:22:51.666	2446383	14981	0x159	CH_A	STD	8	0x3A 0x62 0x00 0x18 0xC6 0x00 0xAD 0x00	b E -	
13.02.2026 04:22:51.671	2451681	5298	0xA2B005	CH_A	A2B	4	0x00 0x5A 0x81 0x00		A2B Register Access (I2C_REG) * Node: 0 * Address: CLK2CFG (0x5A) * Access mode: WRITE * Data: 0x81
13.02.2026 04:22:51.681	2461386	9705	0x420	CH_A	STD	8	0x00 0x00 0x00 0x00 0x00 0x00 0x06 0x00		
13.02.2026 04:22:51.696	2476360	14974	0x340	CH_A	STD	8	0x01 0x1A 0x00 0xC2 0x00 0x0E 0x6A 0xAA	A ?	
13.02.2026 04:22:51.711	2491356	14996	0x165	CH_A	STD	8	0x00 0x00 0x00 0x40 0x00 0x00 0x00 0x05	@	
13.02.2026 04:22:51.722	2502387	11031	0xA2B005	CH_A	A2B	4	0x00 0x65 0x03 0x00		A2B Register Access (I2C_REG) * Node: 0 * Address: DNMSK0 (0x65) * Access mode: WRITE * Data: 0x03
13.02.2026 04:22:51.726	2506349	3962	0x3DA	CH_A	STD	8	0x05 0x64 0x00 0x00 0x00 0x00 0x00 0x00	d	
13.02.2026 04:22:51.764	2544687	38338	0x201	CH_A	STD	8	0x0B 0x57 0x41 0x3C 0x00 0x00 0x18 0x00	WA<	
13.02.2026 04:22:51.773	2553097	8410	0xA2B005	CH_A	A2B	4	0x00 0x1B 0x00 0x00		A2B Register Access (I2C_REG) * Node: 0 * Address: INTMSK0 (0x1B) * Access mode: WRITE * Data: 0x00

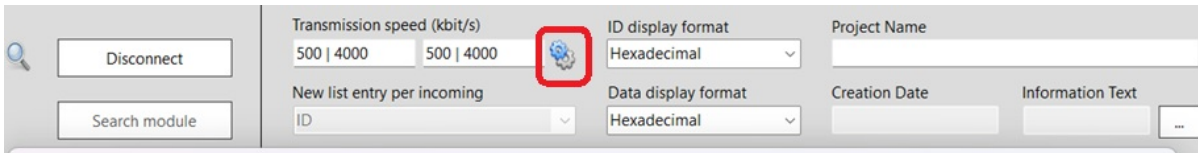
The Designation column includes the human readable data interpretation for easy analysis.

3. Method 1: Software-Connected Logging

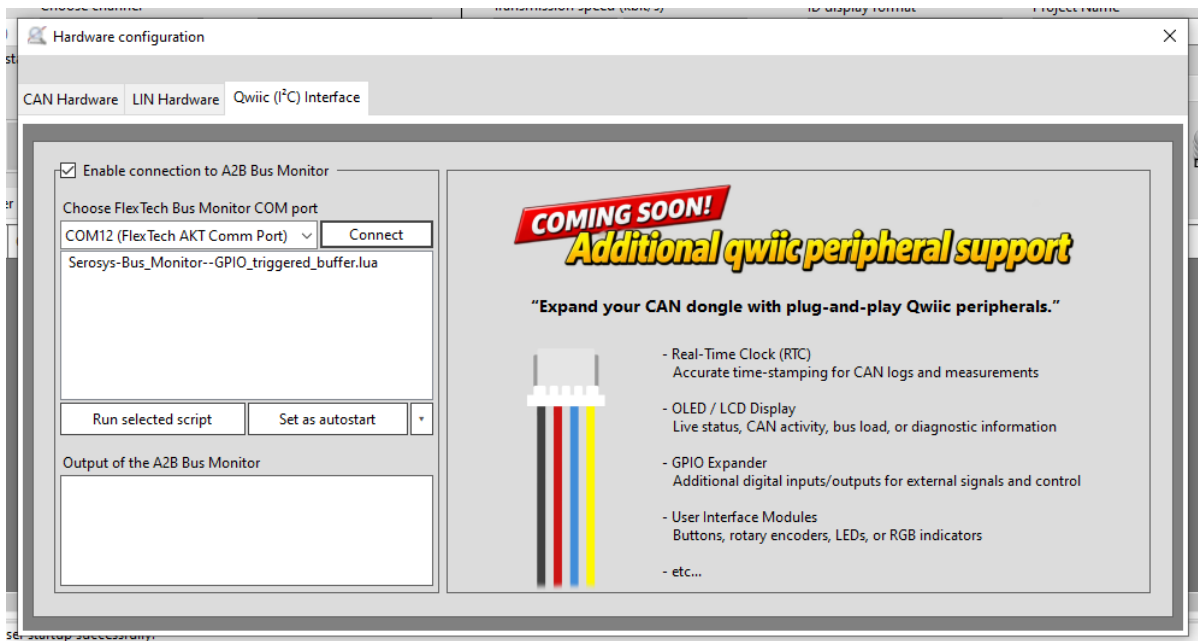
In this method the Lab CAN Pro dongle and the Flextech A2B Bus Monitor are used together with the SLSS CANAnalyser software running on a PC. A2B and CAN data are captured live and synchronized in the Incoming Tab / Log Recorder of the SLSS CANAnalyser software. To log without a PC, see [Method 2: Standalone Logging](#).

3.1. Hardware Configuration

Press the setup gear icon to enter the Hardware configuration screen.



Select the tab "Qwiic (I²C) Interface"



1. Ensure the checkbox "Enable connection to A2B Bus Monitor" is checked.



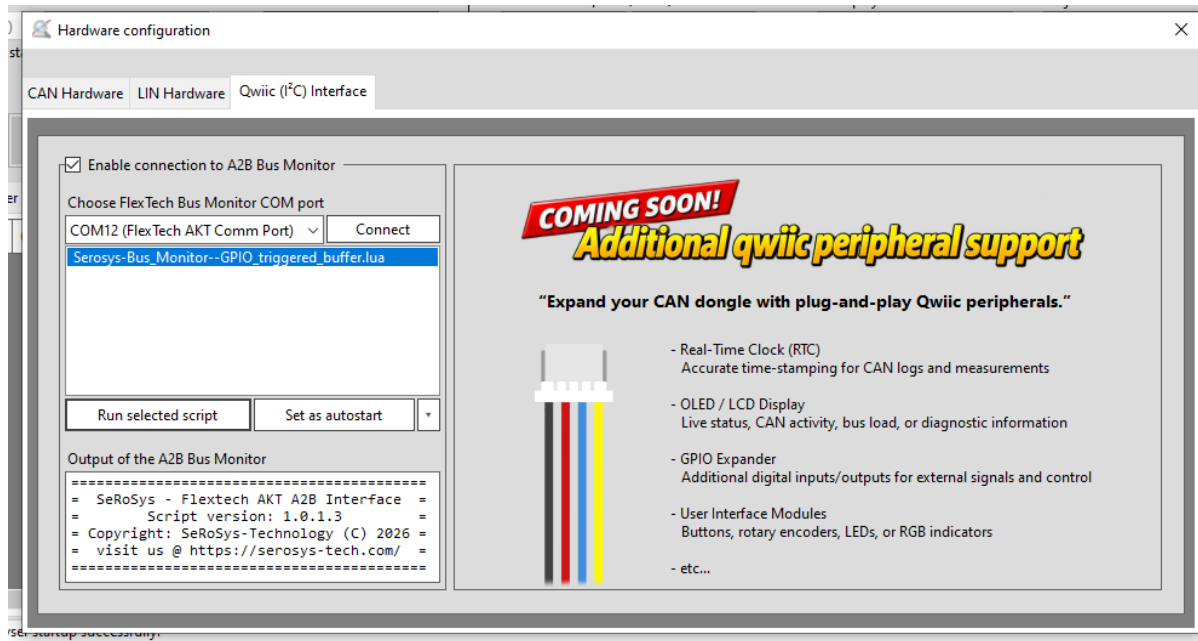
This checkbox MUST be checked. It is required for the SeRoSys dongle to access the connection to the Flextech A2B Bus Monitor.

2. Select the Comm port from the drop-down that matches the description (**FlexTech AKT Comm Port**)

3. Press **Connect**. All of the **.lua** files on the microSD card that is inserted in the A2B Bus Monitor are listed. This is where the downloaded Lua file must reside (see [Hardware Setup](#)).
4. Choose / highlight the file called **"Serosys-Bus_Monitor—GPIO_triggered_buffer.lua"**
5. Press **Run selected script**. This is a one-time load of the Lua script on the Bus Monitor, which configures it for a live, software-connected monitoring and logging session.



It is an option to press "Set as autostart" if the user desires the Lua script to always run on the Flextech Bus Monitor instead of this one-time load to avoid doing this step each time. That method is defined in [Method 2: Standalone Logging](#).



6. The lower box titled **"Output of the A2B Bus Monitor"** must be filled with this output text after the script is run, as shown above. If it stays empty, there is an issue (see [Startup Check and Qwiic Cable Errors](#)). If the output text is shown, everything is ready to go!



If you have already set the Lua script to start automatically (see [Setting the Lua Script to Autostart](#) in Method 2), the Bus Monitor runs it every time it starts. When you connect, the output box will show that an autostart configuration was detected, and the script is already loaded and running.

Close the Hardware Configuration window and now the connection is complete and the tools will work together.

4. Method 2: Standalone A2B + CAN Logging

In this method the Lab CAN Pro dongle and the Flextech A2B Bus Monitor log A2B and CAN data on their own, without a PC connected. Afterwards, the dongle is reconnected to the SLSS CANAnalyser software, where the logs can be analyzed. If you are logging with the software connected, see [Method 1: Software-Connected Logging](#).

There are **two types of standalone logging**. Choose the one that matches your test:

	Constant Logging	Triggered Pre-Buffered Logging
What it does	Constantly logs A2B events and CAN traffic.	Saves a buffered log of the last (x) seconds / minutes each time an external trigger occurs including A2B events and CAN traffic as well as a WAV file.
Button needed	No	Yes, or any other qualifying electrical input to pull the GPIO on the Bus Monitor to ground. For details on the Bus Monitor electrical GPIO characteristics, please consult the proper Flextech Bus Monitor user's manual.
Jump to	Constant Logging	Triggered Pre-Buffered Logging



Both types use the same Qwiic cable, Lua add-on, and Lua script described in [Common Information](#).

4.1. Hardware Configuration

Before starting a standalone logging session (either type), make sure that:

1. The Bus Monitor and the dongle are both running and configured correctly.
2. The Lua script has been set to autostart (see [Setting the Lua Script to Autostart](#)).
3. The Bus Monitor Status LED is solid green. A blinking red Status LED indicates an error state (see [Bus Monitor Status LEDs](#)).

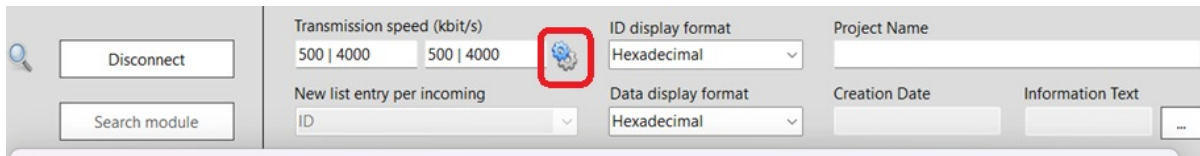
For **triggered pre-buffered logging** only, also make sure that:

4. The button (or other electrical trigger) is connected to the Bus Monitor trigger input that is selected for triggering (GPIO0 by default, see [Selecting the Trigger GPIO](#) below). The I/O LED turns on red for 1 second each time the trigger is received.

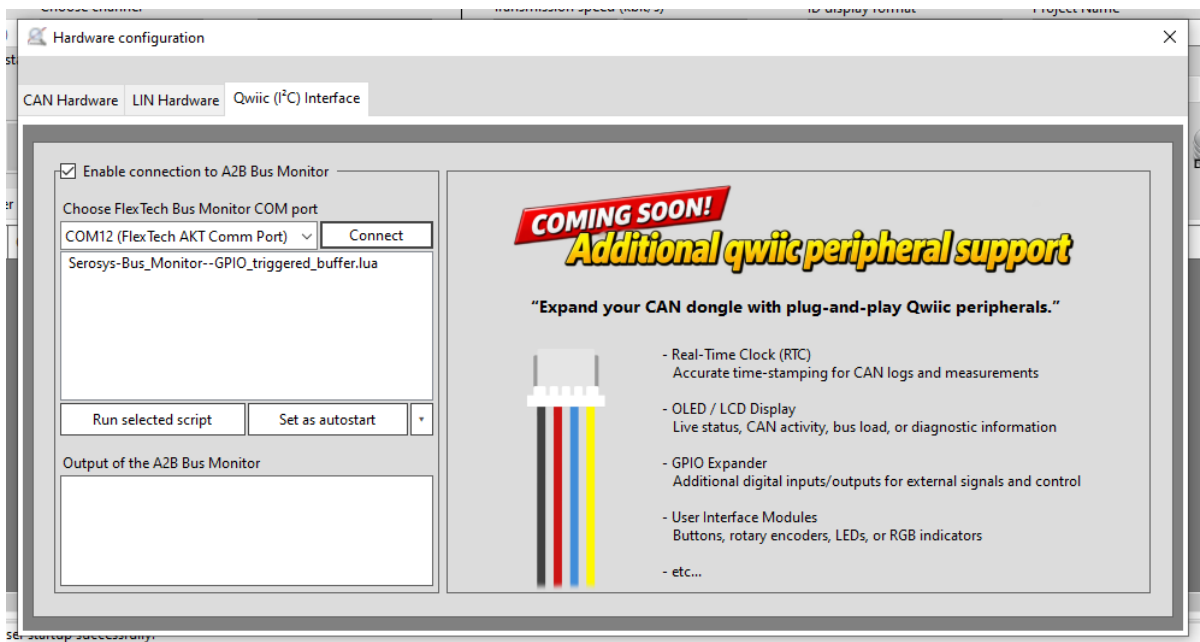
4.1.1. Setting the Lua Script to Autostart

In Method 1, the Lua script is loaded once with **Run selected script** for a live session. For standalone logging there is no PC connected to run the script, so the Lua script can be set to run automatically every time the Bus Monitor starts.

Press the setup gear icon to enter the Hardware configuration screen.



Select the tab "Qwiic (I²C) Interface"

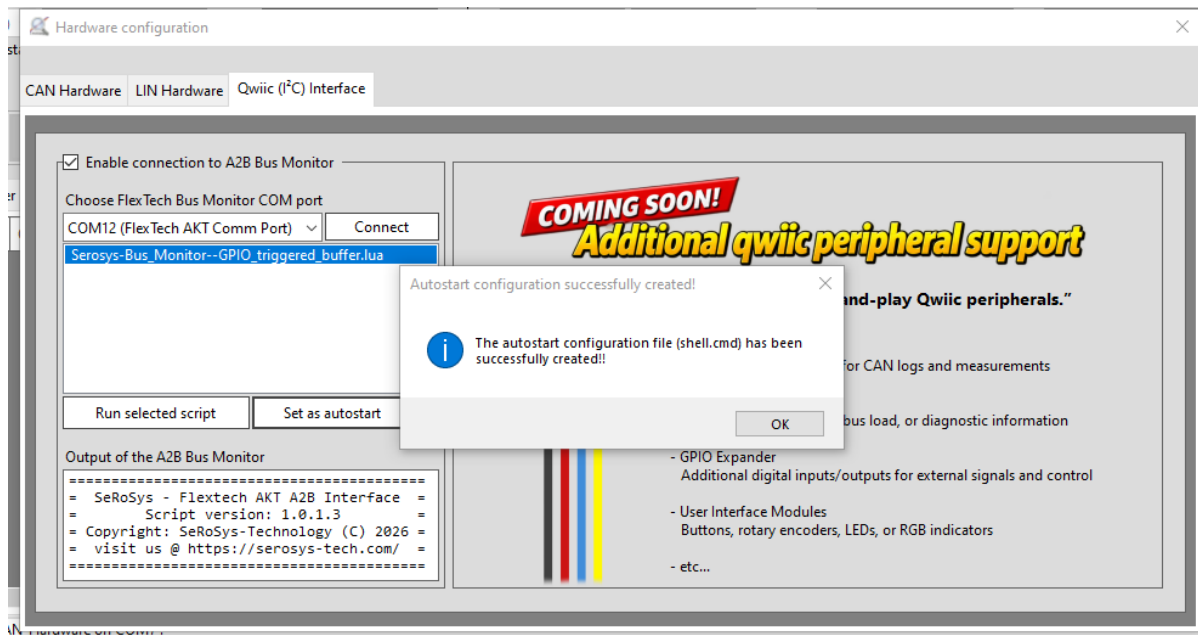


1. Ensure the checkbox "**Enable connection to A2B Bus Monitor**" is checked.



This checkbox MUST be checked. It is required for the SeRoSys dongle to access the connection to the Flextech A2B Bus Monitor.

2. Select the Comm port from the drop-down that matches the description (**FlexTech AKT Comm Port**)
3. Press **Connect**. All of the **.lua** files on the microSD card that is inserted in the A2B Bus Monitor are listed. This is where the downloaded Lua file must reside (see [Hardware Setup](#)).
4. Choose / highlight the file called "**Serosys-Bus_Monitor—GPIO_triggered_buffer.lua**"
5. Press **Set as autostart**.



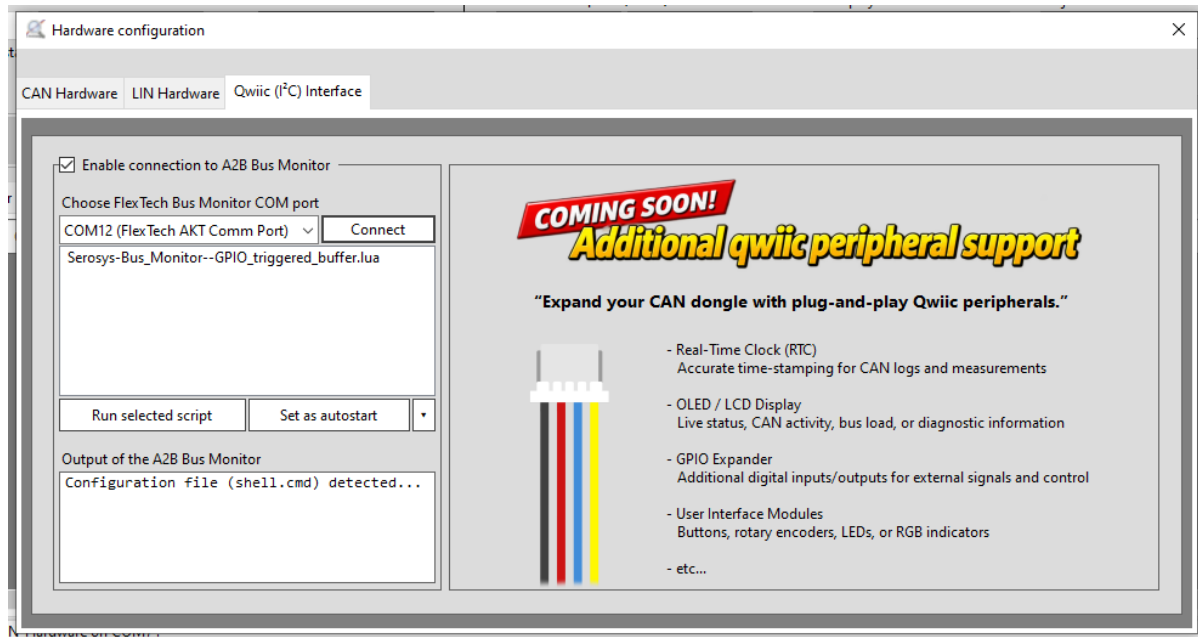
The output box shows the same text as after **Run selected script**. A pop-up message also confirms that the autostart configuration file (`shell.cmd`) has been successfully created. This configuration file is now loaded into the Bus Monitor, which will automatically run the Lua script every time it is started up, both for standalone logging and for software-connected logging, until the configuration file is removed.



If the startup check fails while the Lua script runs from autostart, the red blinking and restart cycle repeats until the Qwiic cable problem is resolved. See [Startup Check and Qwiic Cable Errors](#).

Detecting an existing autostart configuration

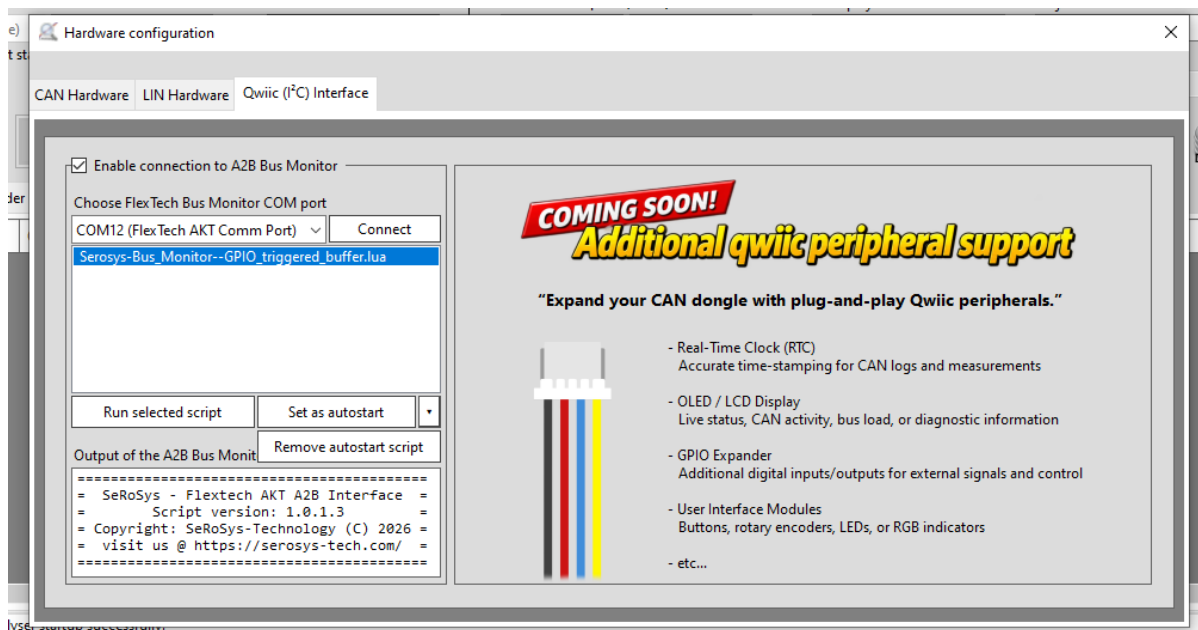
When you connect to the Bus Monitor serial connection and an autostart Lua script has already been loaded, the output box shows that a configuration file (`shell.cmd`) was detected. The script is already running.



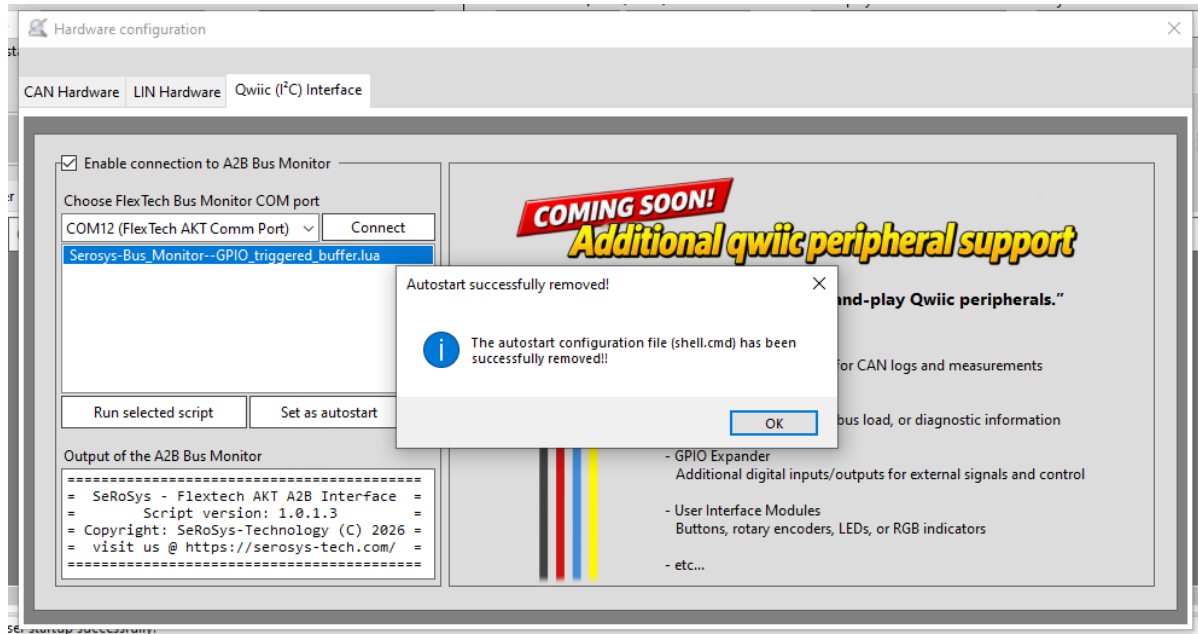
Removing the autostart configuration

To stop the Bus Monitor from running the Lua script automatically each time it receives USB power:

1. Connect to the Bus Monitor as described above.
2. Press the small down-arrow button to the right of **Set as autostart**, and select **Remove autostart script**.



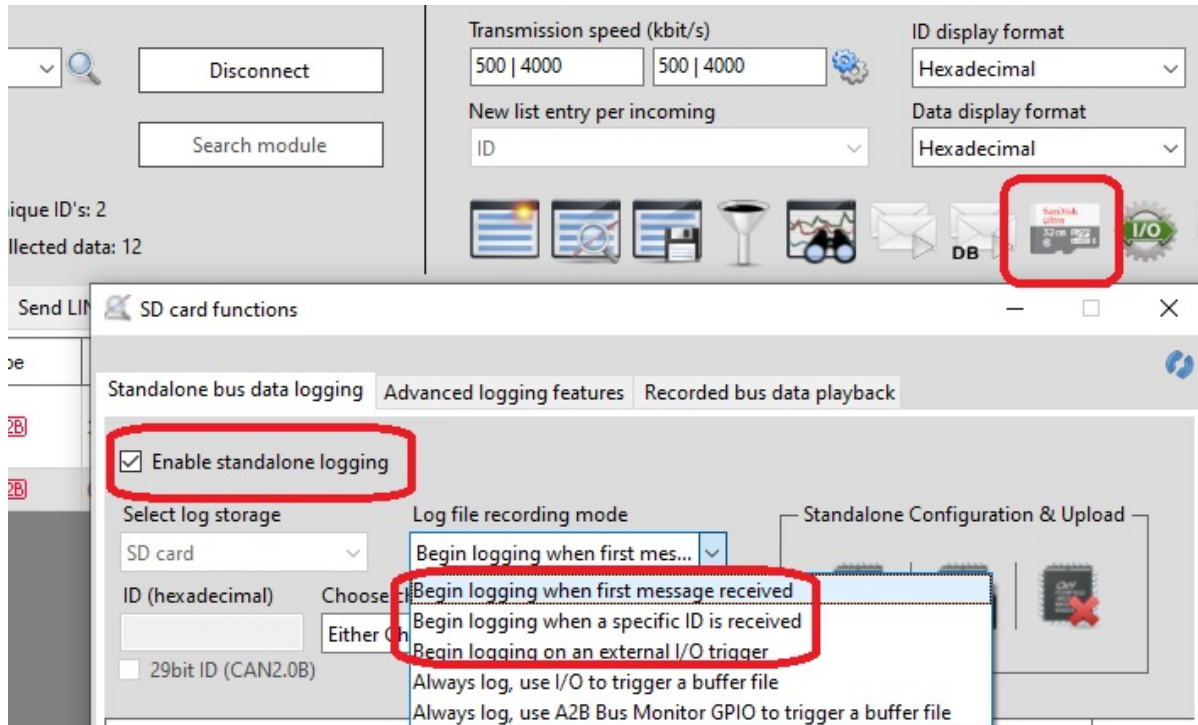
3. A pop-up message confirms that the autostart configuration file (`shell.cmd`) has been successfully removed.



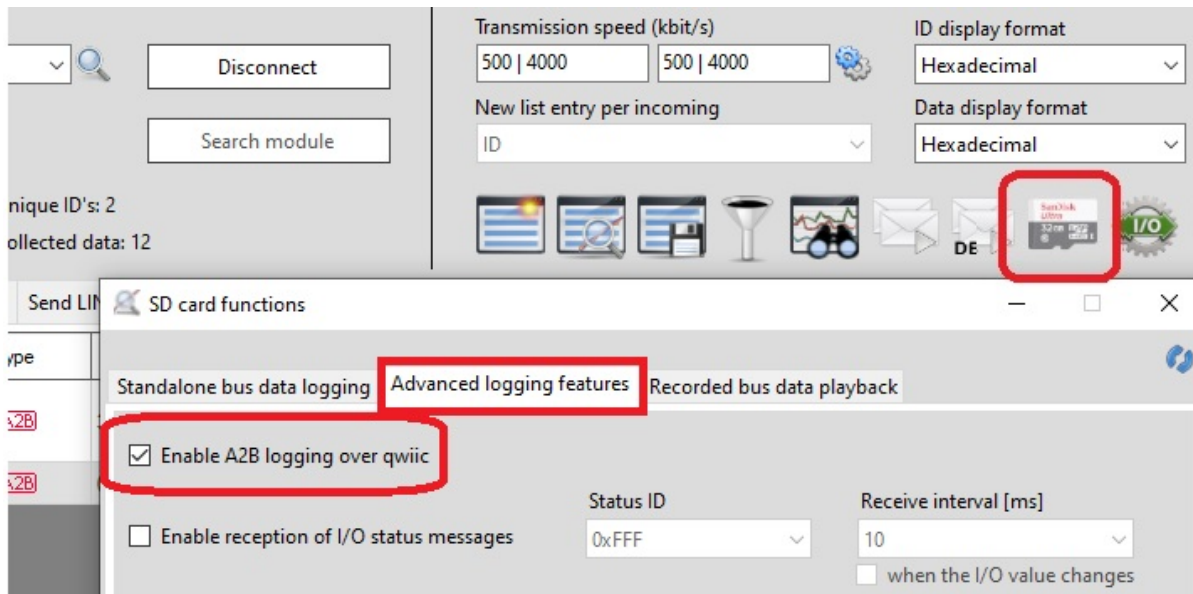
4.2. Standalone Logging Types

4.2.1. Constant Logging

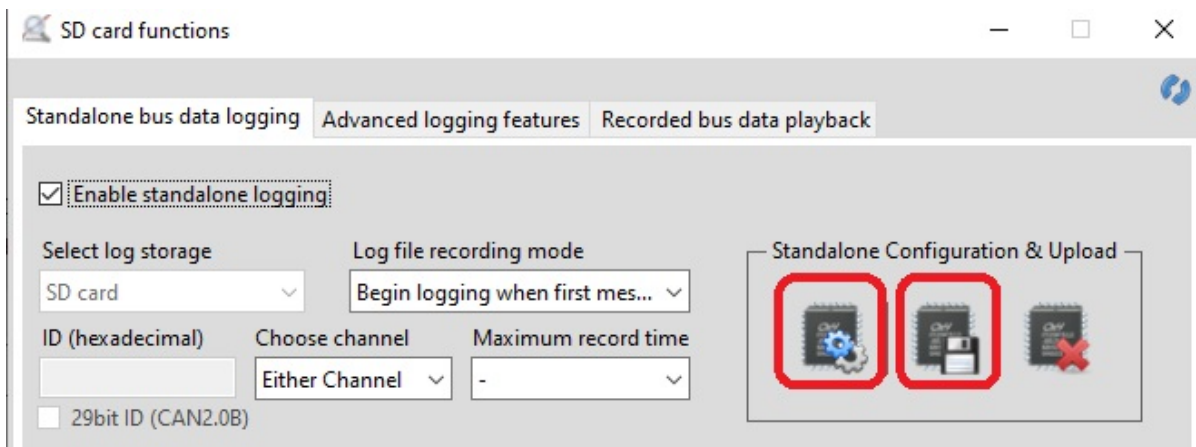
Constant logging only logs the A2B events and CAN traffic. It does not respond to triggers and does not store WAV files. No button is needed, and the trigger settings and WAV file information in the rest of this chapter do not apply to this type.



1. Press the SD card icon
2. In the "Standalone bus data logging" tab, ensure to check the box next to "Enable standalone logging"
3. From the drop-down menu titled "Log file recording mode", choose any one of the top 3 options based on the CAN message triggering desired choice to begin the logging session based on CAN input triggering as a method.



1. In the "Advanced logging features" tab, ensure to check the box next to "Enable A2B logging over qwiic"



1. Ensure the hardware settings are correct for the CAN bus connection (left red box)
2. Save the settings to the SeRoSys dongle (right red box)

The Software will now disconnect from the SeRoSys dongle and the power / connection LED will be solid purple.

At this point, the SeRoSys dongle and the Flextech Bus Monitor need to be both connected to powered USB cables and the qwiic cable must be solidly installed between them. Ensure to power the SeRosys dongle first, and then the Bus Monitor.

If the SeRoSys dongle shows a purple LED and CAN traffic is showing on the green LEDs, and on the Bus Monitor, the Status LED becomes solid green, then the setup is ready to log CAN + A2B events onto the SD card on the SeRoSys dongle.

4.2.2. Triggered Pre-Buffered Logging

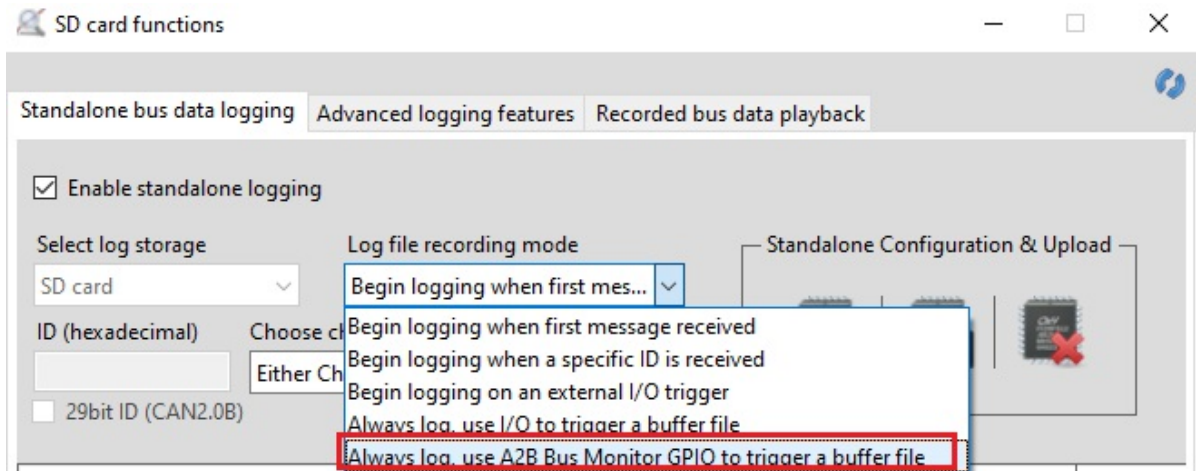
Triggered pre-buffered logging lets you use an external electrical trigger, such as a push-button or any other qualifying electrical GPIO input **on the Flextech Bus Monitor (not on the SeRoSys dongle)**, to store a log file for CAN + A2B on the SeRoSys dongle and a WAV file for A2B on the Bus Monitor. Instead of a continuous log, each trigger saves only the last (x) seconds / minutes of A2B and CAN data, as defined in the **Pre-trigger duration** drop-down box. All other traffic data is discarded as the intention for this mode is to only store notable events for easier analysis.

- After a log and WAV file have been stored, it is ready to use again. Every subsequent trigger stores another set of files (log + WAV)
- GPIO0 or GPIO1 may be used on the Bus Monitor.
- Each trigger also stores an audio WAV file on the Flextech Bus Monitor. The WAV file name matches the duration of the buffered event log file that is stored on the SeRoSys dongle.
- The dongle log file and the Bus Monitor WAV file are linked, as described in [Standalone Logging FAQ and Troubleshooting](#).

Software setup

All steps are the same as in [Method 1 for Constant Logging](#) except for 2 different items.

- Different selection for "Log file recording mode"



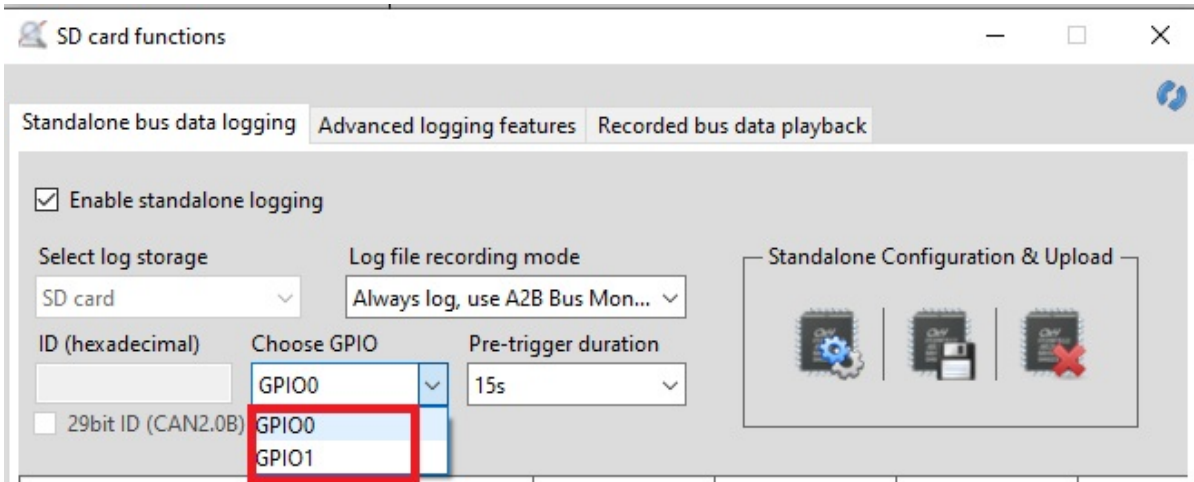
1. Chose "Always log, use A2B Bus Monitor GPIO to trigger a buffer file"

Configure the Bus Monitor GPIO inputs



This must be followed very carefully. The Bus Monitor GPIO number must be specified in 2 places and they must match to avoid failures

a) Select the correct one from the software menu



b) Specify the correct matching GPIO from within the Lua script

The Lua script supports two Bus Monitor inputs for the standalone trigger: **GPIO0** and **GPIO1**. The default is **GPIO0**. If you want to use GPIO0, the Lua script does not need to be edited.

To use GPIO1, you must manually edit the Lua script (*Serosys-Bus_Monitor-GPIO_triggered_buffer.lua*) in a text editor and change the value on the following line near the top of the lua file:

```
local used_trigger_gpio = 0 -- the Bus Monitor GPIO pin for the standalone trigger activation
```

Value of TRIGGER_GPIO	Trigger input used
0 (default)	GPIO0
1	GPIO1

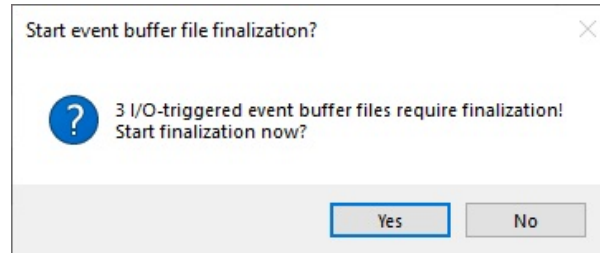


Edit ONLY this line of the Lua script, and change only the number. Do not change anything else in the script.

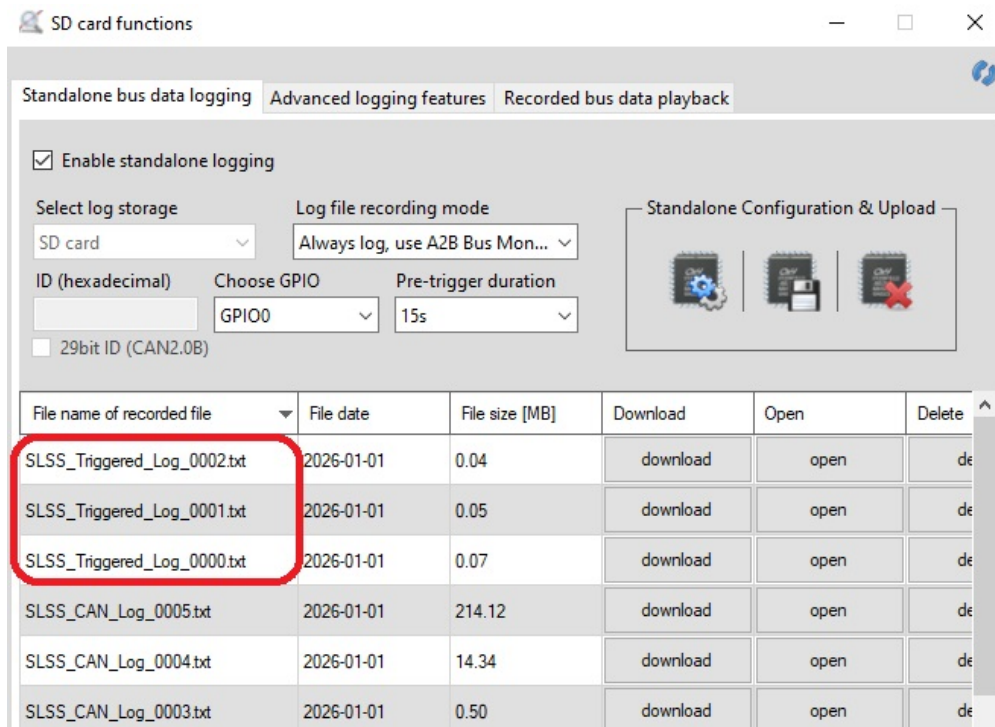
After editing, save the file and make sure the edited script is the one on the microSD card that is inserted in the Flextech A2B Bus Monitor (see [Common Information](#)). Connect your button to the GPIO that you selected. A button connected to the other GPIO will not trigger a recording.

4.2.3. Reconnecting to the Software for Post-Logging Analysis

After logging, reconnect the dongle to the SLSS CANAnalyser software and open the SD card window. If triggers were set, the software asks whether to finalize the pending recordings.



Finalization creates **SLSS_Triggered_Log_XXXX.txt** files on the dongle SD card. These can be opened in the software the same way as other logs, with A2B and CAN messages integrated chronologically.



The last entry of each triggered log is the **0xA2B032** (Bus Monitor GPIO trigger message) event. Its Designation column shows the filename counter that identifies the matching WAV file in the **SLSS_wav_files** directory on the Bus Monitor SD card.



Always finalize the logs with the CANAnalyser software before removing the SD card or clearing WAV files from the Bus Monitor SD card.

For common questions and problem solving, see [Standalone Logging FAQ and Troubleshooting](#) in this chapter.

4.3. Standalone Logging FAQ and Troubleshooting

This section answers common questions about standalone (offline) logging when a SeRoSys dongle is used together with a Flextech A2B Bus Monitor (BM). It also covers what to check when something does not behave as expected. Unless stated otherwise, it applies to **triggered pre-buffered logging**. Constant logging does not use triggers, finalization of triggered logs, or WAV files.



Everything in this section assumes that the Bus Monitor and the SeRoSys dongle are both running and have been configured correctly, as described in [Hardware Configuration](#).

4.3.1. Key Concepts

Trigger

An electrical input on the Flextech Bus Monitor (not on the SeRoSys dongle), such as a button press, is the only event that tells the system there is something worth keeping. Each trigger stores a triggered recording.

Pre-trigger duration

The length of the buffered log (the last (x) seconds / minutes before the trigger) that is saved on each trigger. It is set with the **Pre-trigger duration** drop-down box.

Temporary storage (**_tlb** sub-directory)

While logging, the dongle stores data in a temporary sub-directory named **_tlb** on its SD card. This folder is created automatically by the dongle firmware whenever it is not present.

Finalization

Finalization is the step where the CANAnalyser software converts the data held in **_tlb** into the permanent **SLSS_Triggered_Log_XXXX.txt** files. Finalization is only needed (and only offered) when at least one trigger was set.

Linked WAV file

Each triggered log is linked to a WAV file recorded on the Bus Monitor SD card, in the sub-directory **SLSS_wav_files**. The last entry of every **SLSS_Triggered_Log_XXXX.txt** file is the event **0xA2B032** (*Bus Monitor GPIO trigger message*). Its Designation column shows the filename counter, which identifies the WAV file number the log belongs to. For example, a filename counter of 10 means that the WAV file whose name ends in "10" is the matching recording.



The dongle and the Bus Monitor each keep their own file counter, independently of each other. The link between a CAN log and its WAV file is the filename counter stored in the log (event 0xA2B032). Always finalize the CAN logs with the CANAnalyser software before removing the SD cards or clearing WAV files (see the questions below). For the meaning of the Bus Monitor LEDs, see [Bus Monitor Status LEDs](#).

4.3.2. Frequently Asked Questions

What is the difference between constant logging and triggered pre-buffered logging?

Constant logging only logs the A2B events and CAN traffic. It does not respond to triggers and does not store WAV files. Triggered pre-buffered logging stores a log of the last (x) seconds / minutes each time the trigger input on the Bus Monitor is activated, and also stores a matching audio WAV file on the Bus Monitor. See [Standalone Logging Types](#).

Can I connect the trigger button to the SeRoSys dongle?

No. The trigger input is on the Flextech Bus Monitor, not on the SeRoSys dongle. Connect the button to the selected input (see [Selecting the Trigger GPIO](#)) and ground on the Bus Monitor.

What happens if the button is never pressed during a session?

Nothing of lasting value is kept. The dongle does write its data to the temporary `_tlb` sub-directory, but if no trigger was set, that data is automatically deleted the next time the dongle is connected to the software. No `SLSS_Triggered_Log_xxxx.txt` file is created, no finalization message appears in the SD card window, and no matching WAV file is kept.

Why do I see no "_triggered" files on the SD card when I remove it from the dongle?

If the SD card is removed before finalization is done, the data is still in the `_tlb` sub-directory and has not yet been converted into `SLSS_Triggered_Log_xxxx.txt` files. This can be confusing, because the card looks empty of logs.



Always connect the dongle to the CANAnalyser software and complete finalization before removing the SD card or copying files from it.

Deleting or modifying the `_tlb` sub-directory by hand removes any data that has not yet been finalized. If the folder is missing, the dongle firmware recreates it automatically.

I pressed the button several times in one session. Do I get one log or several?

Each button press creates a separate triggered recording. Pressing the button three times in a session means three files to finalize.

I did several logging sessions on different days before connecting to the software. Will the files add up?

Yes. Unfinalized triggered recordings accumulate until you finalize them. If you press the button 3 times in one session, power everything down, and later press the button 3 more times in another session, the SD card window will report 6 files to process. This is intentional, so that tests can be run over multiple sessions or days before the data is processed in one go.

The software asks me to finalize every time I open the SD card window. What if I say "No"?

You can decline, and nothing is lost. The software will ask again each time you open the SD card window, and will keep asking until the files have been finalized. If you continue logging in the meantime, the new triggered recordings are simply added to the pending ones. For example, if you decline to finalize 2 logs and then record 2 more, the next prompt will offer to finalize 4 logs.

Why does the software ask before finalizing? Why not just do it automatically?

Finalization can take several minutes when there are multiple or longer recordings. Asking first lets you reach your standard log files, the SD card settings, or the playback functionality immediately, without waiting for finalization to complete. Decline now and finalize when you have time.

Does deleting a file on one of the SD cards break the file numbering?

No. The dongle and the Bus Monitor file counters work completely independently of each other, so you can delete whatever you want on either side without disturbing the other counter. The Bus Monitor automatically fills in missing numbers first. For example, if the WAV files 3, 4, 5, and 6 exist and you delete 3, the next recording will be numbered 3, and the one after that will be 7.

The important point is that the number is determined first and then transferred into the CAN log file. The number stored in the log is therefore always the correct reference to the WAV file that was recorded.

Is there any way the link between a CAN log and its WAV file can be lost?

The link itself is stored in the CAN log, so deleting files on either card does not change it. There is one situation to be aware of: if the WAV files are deleted from the Bus Monitor SD card **before** the CAN logs have been finalized, the Bus Monitor WAV file counter starts again from 1. You will then have CAN logs with multiple identical WAV file numbers.



Finalize the CAN logs with the CANAnalyser software first, and only then clear files from the SD cards.

How can I tell that the Bus Monitor and the dongle are communicating over the Qwiic cable?

When the Lua script starts, it checks the connection by sending an I2C message over the Qwiic cable to the SeRoSys hardware. If this succeeds, the Bus Monitor goes to operational mode and a solid green Status LED indicates everything is working. If it fails, the Status LED blinks red for 5 seconds and the Bus Monitor then restarts. If the Lua script was started with **Run selected script**, it does not retry, and you must reconnect and run the script again after fixing the cable. If it was started with **Set as autostart**, it keeps repeating the cycle until the cable problem is resolved. See [Startup Check and Qwiic Cable Errors](#).

Why do I hear a high-pitched sound from the Bus Monitor?

A high-pitched sound is a sign that the Bus Monitor is not getting adequate USB power. Check that the USB cable is not too long (a long cable causes voltage loss, and the voltage must stay at 5V) and that the USB power source can supply sufficient current (a minimum of 150 mA). See the Troubleshooting table below and [Hardware Setup](#).

How do I choose which GPIO the button is connected to?

The default trigger input is GPIO0. To use GPIO1, edit only the `TRIGGER_GPIO` line in the Lua script as described in [Selecting the Trigger GPIO](#).

4.3.3. Troubleshooting

Symptom	Likely cause	What to do
No finalization prompt appears when I open the SD card window	No trigger was set during the logging session, so the temporary data was discarded. Or the data was already finalized.	Confirm that the button was pressed while logging. Check that the I/O LED on the Bus Monitor flashes red for 1 second on each press.
The SD card shows no "SLSS_Triggered_Log" files after logging	Finalization has not been completed yet. The data is still in the <code>_tlb</code> sub-directory.	Insert the SD card back in the dongle, connect to the CANAnalyser software, open the SD card window, and finalize. In future, finalize before removing the SD card.
The SD card window reports more files to process than expected	Triggered recordings from earlier sessions were never finalized, and they accumulate.	Finalize them all. If you need to keep sessions separate, finalize at the end of each session.
The prompt to finalize keeps coming back	Finalization was declined earlier. The prompt returns every time until the files are finalized.	Choose to finalize when you are ready to wait for it to complete.
Several CAN logs reference the same WAV file number	The WAV files on the Bus Monitor SD card were deleted before the CAN logs were finalized, so the WAV counter restarted at 1.	For future sessions, finalize the CAN logs before clearing WAV files from the Bus Monitor SD card.
A log's referenced WAV file cannot be found	The WAV file was removed from the Bus Monitor SD card, or you are looking in the wrong location.	Open the log and read the filename counter in the last entry (0xA2B032). Check the <code>SLSS_wav_files</code> directory on the Bus Monitor SD card for the file ending in that number.
The Bus Monitor does not function properly, or a high-pitched sound can be heard from the Bus Monitor	The Bus Monitor is not receiving adequate USB power. The USB cable may be too long, causing a voltage loss below 5V, or the USB power source may not supply enough current (a minimum of 150 mA is required).	Use a shorter USB cable. Connect the Bus Monitor to a USB port or power supply that provides a stable 5V with adequate current capability. See the warning in Hardware Setup .

Symptom	Likely cause	What to do
Erratic or unexpected behavior of the Bus Monitor or the SeRoSys dongle when they are connected by the Qwiic cable	The red wire of the Qwiic cable was not cut. 3.3V can then backfeed between the 2 hardware modules and cause erroneous behavior.	Check the Qwiic cable. A standard Qwiic cable purchased separately contains 4 wires, and the red wire MUST be cut for this setup. Make sure that the red wire is cut. See Hardware Setup .
Status LED blinks red for about 5 seconds, then the Bus Monitor restarts	The startup check could not communicate with the SeRoSys hardware over the Qwiic (I2C) cable.	Check that the Qwiic cable is fully connected to both devices (the Bus Monitor and the SeRoSys dongle). Also confirm that the red wire of the Qwiic cable has been cut (see above). What to do next depends on how the Lua script was started. If you started it with Run selected script , the Bus Monitor will not retry: open the Hardware Configuration in the SLSS CANAnalyser, reconnect to the Bus Monitor, and run the Lua script again. If it was started with Set as autostart , the 5-second red blinking and restart repeat until the cable problem is resolved. After that, the next automatic restart starts normally with a solid green Status LED.
Bus Monitor Status LED is blinking red	The Bus Monitor detected an error in one of its major operations.	Stop logging and check the Bus Monitor SD card and connections. See the Flextech documentation for details.
Bus Monitor Status LED is solid green but no WAV file is created	No trigger was received, or the trigger did not reach the Bus Monitor. (Constant logging does not store WAV files.)	Check the trigger wiring and verify the I/O LED flashes red for 1 second when the button is pressed. Make sure the button is connected to the GPIO selected in the Lua script (GPIO0 by default, see Selecting the Trigger GPIO).

4.3.4. Best Practices Checklist

1. Make sure the USB cables are not too long and that the USB power source provides a stable 5V with adequate current.
2. Make sure the red wire of the Qwiic cable has been cut, and that the cable is fully connected to both the Bus Monitor and the SeRoSys dongle.
3. Start with both devices running and configured correctly (solid green Status LED on the Bus Monitor).
4. For triggered pre-buffered logging, connect the button to the selected trigger input on the Bus Monitor (GPIO0 by default).
5. For triggered pre-buffered logging, press the button each time you want to keep a recording. Confirm the I/O LED flashes red for 1 second.
6. After testing, reconnect the dongle to the CANAnalyser software and open the SD card window.
7. Finalize all pending files before removing the SD card or copying files from it.
8. Do not clear WAV files from the Bus Monitor SD card until the CAN logs have been finalized.
9. Keep a backup copy of both SD cards' files for important test sessions.