Digiteq Automotive

TraceBox Lite

User Manual 1.3

Author: Tomas Lustyk, Robert Nedved,

December 17, 2021

HEAD OFFICE
Digiteq Automotive s.r.o.
Novodvorská 994/138
142 21, Prague 4
Czech Republic

CONTACTS
+420 239 046 750
info@digiteqautomotive.com
www.digiteqautomotive.com

BILLING INFORMATION
Company ID. No. 26 46 60 23
VAT ID No.: CZ 26 46 60 23
registered by the Municipal Court in
Prague, Section C, Insert 84128

MANAGEMENT BOARD
Andreas Hauptvogel
Milan Klaus

BANK ACCOUNT
Commerzbank Aktiengesellschaft
Jugoslávská 1, 120 21 Prague 2
Account No.: 10634787/6200 (EUR)
IBAN: CZ4262000000000010634787
SWIFT: COBACZPXXXX

A Volkswagen Group Company

Digiteq Automotive

Contents

List of tables

Changes

Version	Date	Change description	Changed by	Approved by
1.0	5.1.2021	First version	Tomáš Lustyk	Radek Pravda
1.1	2.5.2021	Chapter Connector description added	Robert Nedvěd	Jiří Breitkopf
1.2	1.12.2021	Chapter Output file format added	Jan Dittrich	Robert Nedvěd

1 About this User Manual

Company reserves the right to make technical changes to the equipment or changes to this document without any prior notice. No guarantee is given for the information provided. No part of this manual may be reproduced in any form or by any means without the written permission of the publisher. All technical information, drawings, screenshots etc. are liable to law of copyright protection.

We are grateful for references to mistakes, or suggestions for improvement, to be able to offer you even more efficient products in the future.

2 Safety Instruction

Dear Customer, the following safety instructions are intended not only for the protection of your health but also for the protection of the product.

This section gives an overview of all important aspects of safety for the protection of individuals and to ensure safe and trouble-free operation.

The warranty/guarantee become void if damage is incurred resulting from non-compliance with these operating instructions. We do not assume any liability for consequential damage!

We also do not assume any liability for damage to other property or personal injury caused by improper use or failure to observe the safety instructions. In such cases the guarantee/warranty will become void!

Therefore, read the following items very carefully before connecting the product and taking it into operation.

2.1 General Safety Instructions

- The product may only be set up, started or serviced after gaining familiarity with the appropriate Operating Instruction.
- The products, equipment and device must only be used indoor.
- Use the products, equipment and device only for its intended purpose as described in Product Specification.
- The products, equipment and devices should not be operated in potentially explosive atmospheres.
- During operation of the products, equipment and device, do not permit any work method that hinders the safety of the products, equipment and device.
- Always keep the working area of the unit clean and in good order so as to avoid any risk posed by dirt or mislaid parts.
- Do not exceed the technical performance data specified for each products, equipment and device.
- Keep a legible copy of all safety precautions and descriptions of dangers and hazards at the location of the product, equipment and device, and replace them as needed.
- Both the operation of and work on the product, equipment and device must only be carried out by trained personnel.
- In case of malfunction, immediately stop the unit.
- Have the fault corrected by appropriately trained personnel.

3 Product Specification

3.1 General Description

The TraceBox Lite is a modular solution for tracing communication between ECUs (Electronic Control Unit). It is primary used as data logger extension for test benches and cars. Main advantage of this solution is possibility to adapt tracing capacities and capabilities of the trace box according to agreed requirements. Which means that user can have for example one, two, or three Automotive Ethernet interfaces.

3.2 Trace Interfaces

- 4x CAN / CAN FD
- 4x Serial ports
- 1x HTTP stream port (Internal infotainment logs)
- Up to 2x Automotive Ethernet – 1 Gbps (1000BASE-T1) or 100 Mbps (100BASE-T1)

3.2.1 CAN / CAN FD

Four CAN / CAN FD buses can be traced simultaneously by the Tracebox. The speed of the buses is set to 500 kbit for CAN bus and 2000 kbit for CAN-FD bus. Tracebox is in passive mode, it sends no data to CAN bus, which makes it invisible to other nodes on the bus. There is no terminating resistor on the Tracebox CAN connectors.

3.2.2 Serial Port

Four serial busses (RS232) can be traced simultaneously by Tracebox Lite. Baud rates 9600 up to 230400 baud are supported.

3.2.3 HTTP Stream

HTTP stream uses Ethernet interface, a compatible USB network card is connected to the MIB unit (to its USB HUB). The compatible USB network card is included in the package. The USB network card is then connected to the TraceBox into the HTTP stream. User has to specify IP address and port (number) of MIB trace in the web interface of the TraceBox.

3.2.4 Automotive Ethernet

The device is able to trace up to 2 1000BASE-T1 or 100BASE-T1 Automotive Ethernet interfaces. The line between two ECUs is interrupted and the TraceBox Automotive Ethernet interface is inserted between these two ECUs. Therefore, there are two ports for each Automotive Ethernet interface on the Tracebox. In other words, Tracebox behaves as a repeater, it does not alter the communication. See Figure 1 for details.

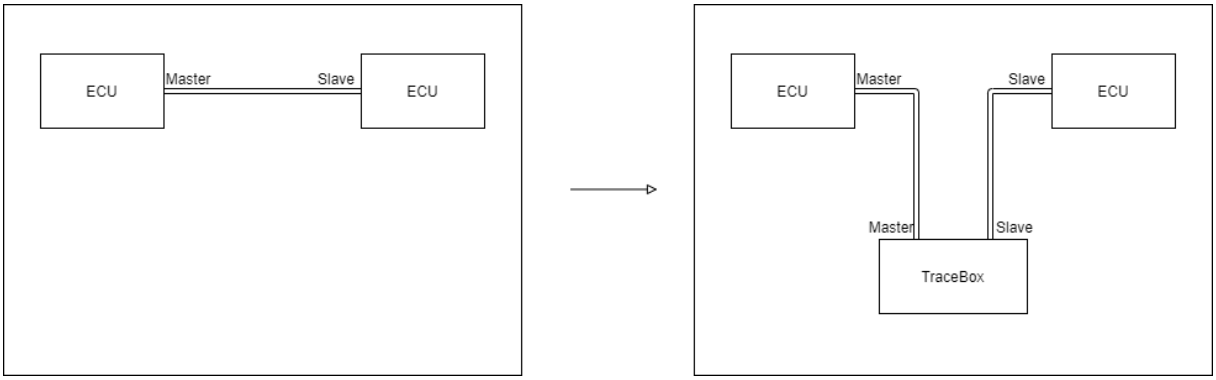


Figure 1: Automotive Ethernet Connection

3.3 Control Interfaces

- 2.4 GHz WiFi Hotspot
- 2.5 Gbit Ethernet

3.4 Mechanical and Electrical Properties

Power supply	6 V to 28 V
Power consumption	max 70 W
Operating temperature	0 °C to 50 °C while preventing condensation
Storage temperature	-20 °C to 70 °C while preventing condensation
Dimensions (W x D x H)	330 x 409 x 96 mm
Weight	5 kg

Table 1: Mechanical and Electrical Properties

3.5 Connector Description

The figure Figure 2 show layout of the connectors on the TraceBox-Lite case. The connectors are also labelled on the case with small stickers.

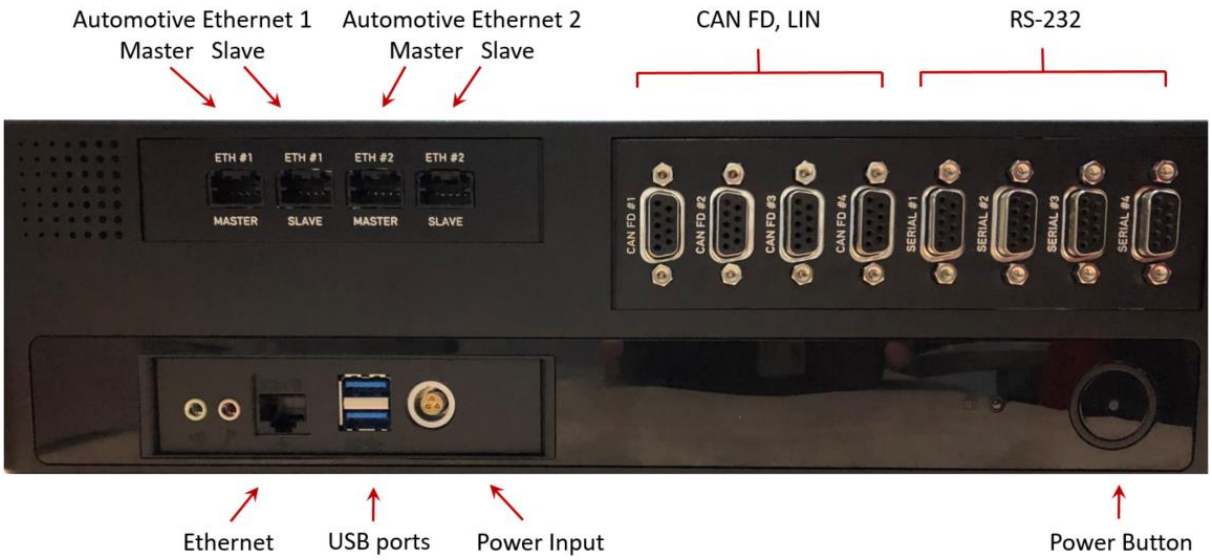
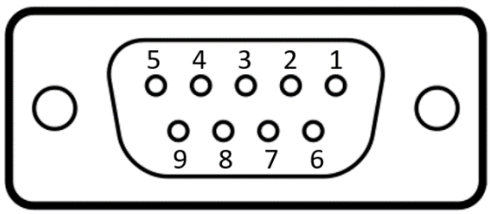


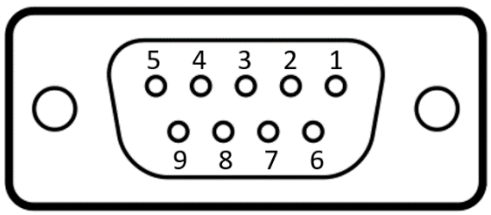
Figure 2: Tracebox-Lite Connector Layout

Pinouts of individual interfaces are described on following figures: , , and .



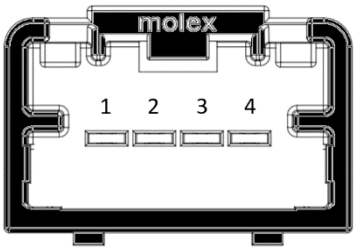
1. No Connection
2. CAN_L
3. GND
4. LIN_2
5. Shield
6. No Connection
7. CAN_H
8. LIN_1
9. No Connection

Figure 3: Can Connector Pinout (D-Sub 9 Female)



1. Data Carrier Detect
2. Receive Data
3. Transmit Data
4. Data Terminal Ready
5. GND
6. Data Set Ready
7. Request to Send
8. Clear to Send
9. Ring Indicator

Figure 4: Serial Port Connector Pinout (D-Sub 9 Female)



1. No Connection
2. TRD+
3. TRD-
4. No Connection

Figure 5: 1000BASE-T1 Connector (Molex Mini50 Female)

4 First time Power on

4.1 Included in the box

- TraceBox PC according to configuration of the customer.
- Collection of cables for test bench, or cable cable adaptors for in-car usage.
- Power supply cable according to configuration (12 V for in-car use case or 230 V for test bench use case).
- Ethernet network cable, 2 m.
- Manual.

4.2 Starting with the device

The first step is to connect a power supply cable. Then, start the device by pressing the power button. A blue indicator light should be ON after the start of the device. The device has its own network, user can connect to the network. The trace box computer software manages the network.

Verify communication between TraceBox and your computer. The TraceBox communicates with your computer via WI-FI or common Ethernet cable connection. The TraceBox has its own WiFi hotspot “Tracebox”. The password to the WiFi hotspot is “*tracebox*”. The user should start web-browser and type “*trace.box*” into the address line. Then the web page with graphical user interface (GUI) of the TraceBox should start, see Figure 6. In case of some DNS errors, type “192.168.4.1” instead.

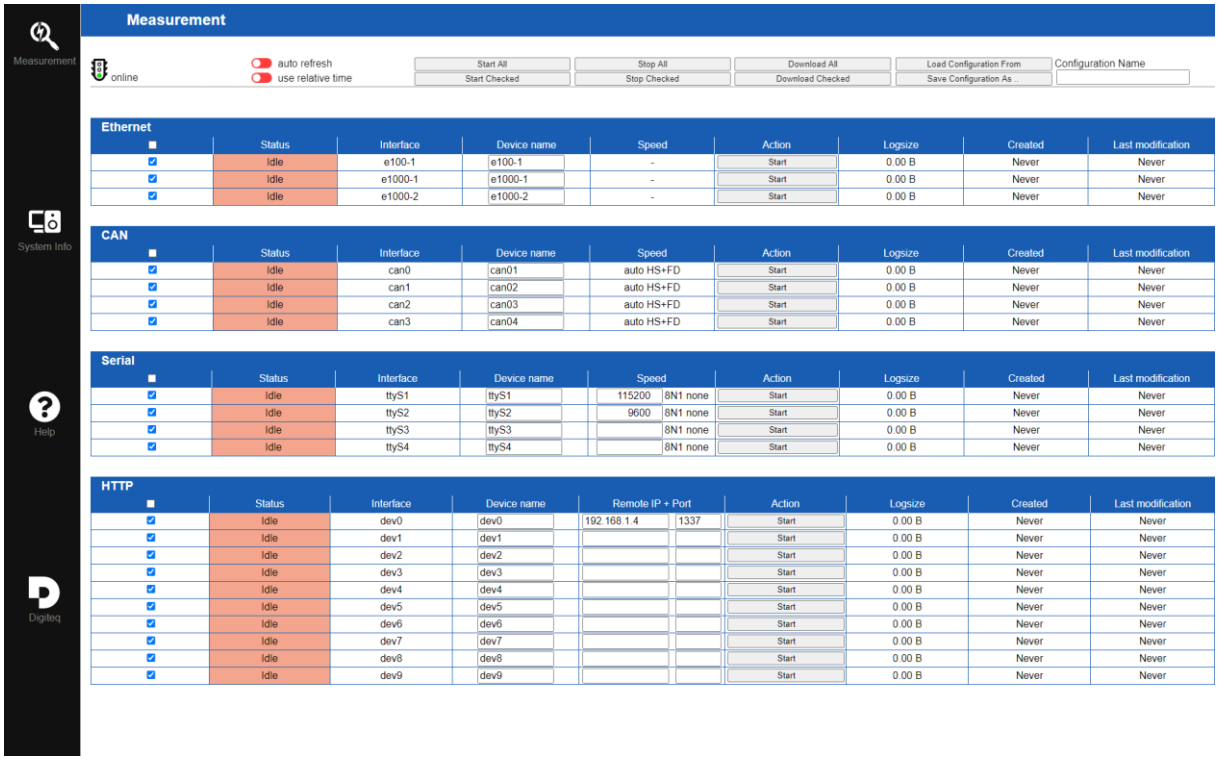


Figure 6: Graphical user interface of TraceBox

Next step is to connect the cables of interfaces that should be traced. Follow the instruction mentioned in individual chapters of interfaces to correctly connect traced interfaces.

To terminate the work with TraceBox, the power button should be pressed. The device will correctly stop to work. The indicating blue light goes out.

5 TraceBox Controlling

The TraceBox remote control is done via the web page graphical user interface. An overall picture of the GUI can be seen in Figure 6.

5.1 Web interface

Tracebox page can be accessed using URL: 192.168.4.1 or "trace.box" in your web browser. The web interface should load immediately. There are three main panels in the web interface panels: "**Measurement**", "**System Info**", and "**Help**". The panel "**Measurement**" is loaded as the first.

The panel "**Measurement**" (Figure 6) includes all information about the traced interfaces, mainly overall status of the device and overview of traced automotive interfaces. In the left upper corner of the web interface there are traffic lights that indicator. This shows the connection status between your web-browser and TraceBox (to an API backend). *Green light* indicates the connection is all right. *Red Light* stands for connection problems. There are also two indicators of "*auto refresh*" of the page and "*use relative time*". The main part of the window consists of all available trace interfaces. An interface (for example Ethernet) is describe by:

- Check box – enables to select interface for tracing.
- Status – Idle or Running
- Device name – user can change the name at will, this is also the name of the log file
- Speed – communication speed used by the device
- Action – starts or stops tracing individually.
- Log size – size of the log files in [Bytes].
- Created – date and time, when log file was created.
- Last modification – date and time, when log file was modified.
- Remote IP + Port (for HTTP traces)

At the top of the web page, there is a section with 8 buttons. There is a "Start All" button that starts all available tracing. "Start Checked" button to start only selected ones. "Stop All" button is for stopping all the tracing and "Stop Checked" will stop only the selected ones. "Download All" will download zip-file with all available traces to your computer. "Download Checked" only for the selected ones.

The device is platform independent, it enables to trace all car/project platforms available (MQB, 37W, MEB). In case that device is used with more than one platform, it enables to save and load configuration of traces (selection of interfaces). This should simplify work with the device and more platforms/projects. User can have for example one configuration file for MEB platform, one for MQB, and another one for 37W.

The "**System**" section is used for the tracebox system control. There are three buttons available. "Restart" for the tracebox system reboot. "Shutdown" for the tracebox system power off. And "Update" which will let user choose debian package that will be installed into the tracebox system.

Figure 7: Tracebox Web Interface – System Section.

The "**System Info**" section shows information about the TraceBox itself. There is information about the current version of the TraceBox-Lite Daemon and current version of the Web interface. Also there is a link to update packages. Information about system follow: Operational system information, how long the trace box is running since the last start, and statistics about memory, and disc usage. See the Figure 8 for details.

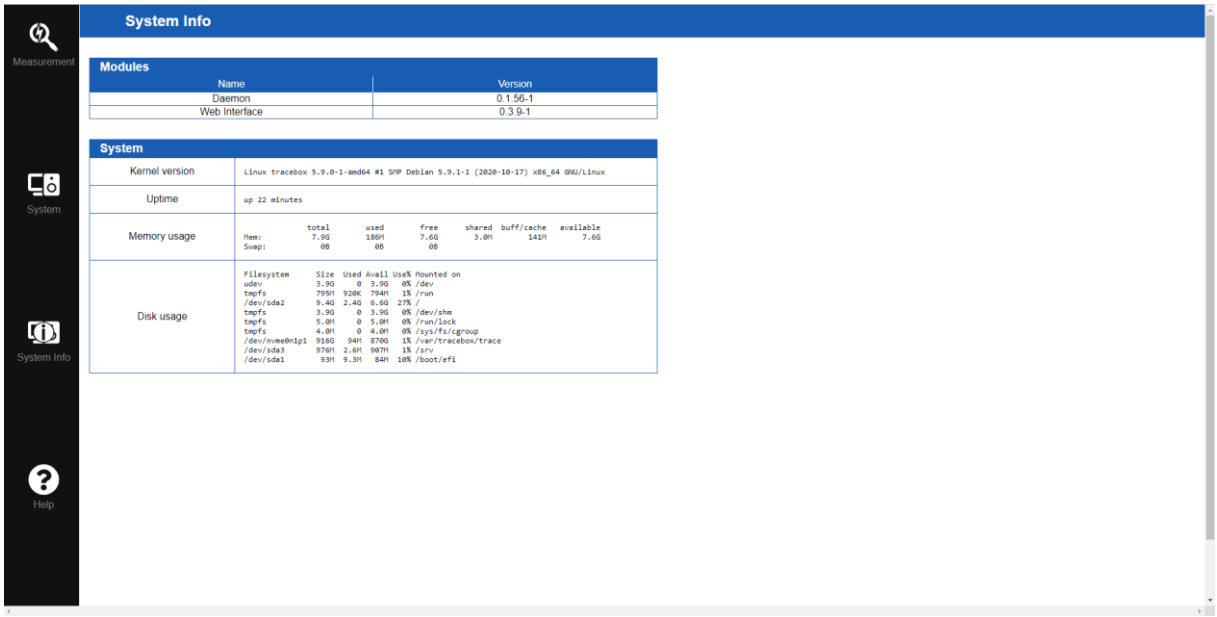


Figure 8: Tracebox Web Interface – System Information section

The “**Help**” section provides user with clear help and support. There is a link to this manual as it provides entire documentation to the TraceBox-Lite. Also there is link to the API commands guide. In case of any troubles there is an e-mail contact for the tracebox support.

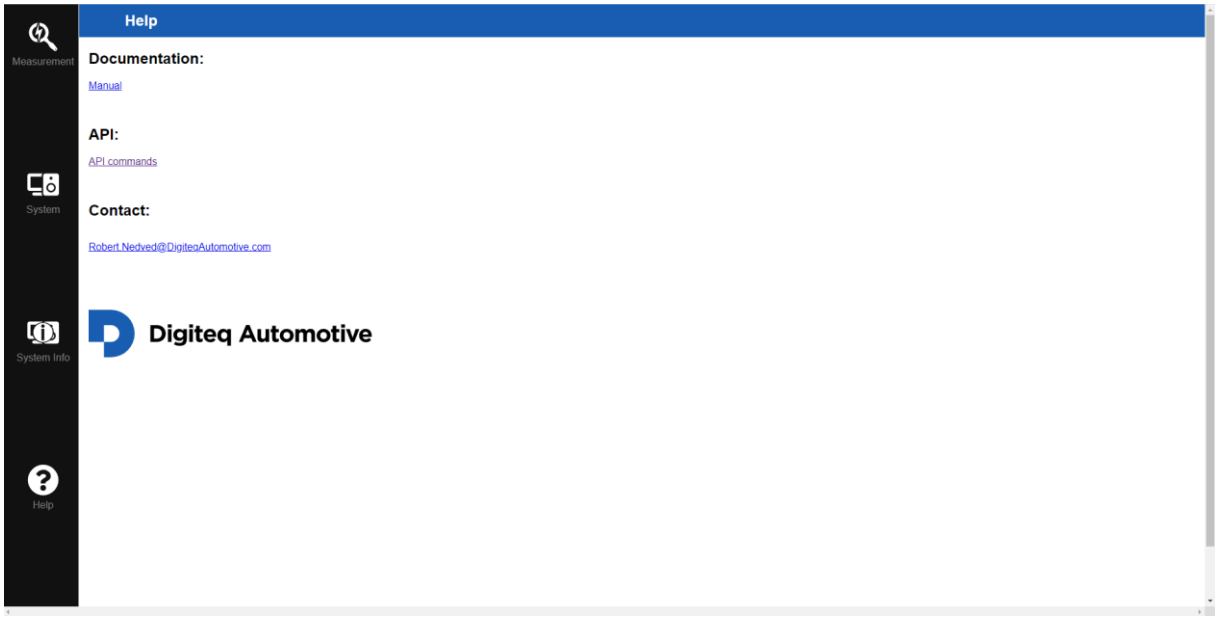


Figure 9: Tracebox Web Interface – Help Section.

5.1.1 Connections

This section provides detailed description of procedure, how the traces of individual interfaces can be performed.

5.1.1.1 CAN

Connect TraceBox to CAN bus you want to trace. The description of the interface is in section 3.2.1, pinout of the CAN connector is in .

The CAN bus interface in GUI is in Figure 8. Optionally change device name in web interface to customize log filename. Click on “*Start*” button in row of device you want to trace. The “*Start*” button will start tracing for the individual device for interface CAN. To terminate tracing use “*Stop*” button on individual device.

CAN

	Status	Interface	Device name	Speed	Action	Logsize	Created	Last modification
☑	Idle	can0	can01	auto HS+FD	Start	0.00 B	Never	Never
☑	Idle	can1	can02	auto HS+FD	Start	0.00 B	Never	Never
☑	Idle	can2	can03	auto HS+FD	Start	0.00 B	Never	Never
☑	Idle	can3	can04	auto HS+FD	Start	0.00 B	Never	Never

Figure 8: Graphical User Interface – CAN Traces Detail

5.1.1.2 Serial port

Connect TraceBox to Serial line you want to trace. The description of the interface is in section 3.2.2, pinout of the Serial port connector is in

Figure 9 shows the serial interface section. Optionally change the device name to customize log filename. The "*Start*" button will start tracing the individual device. To terminate tracing use "*Stop*" button on individual device.

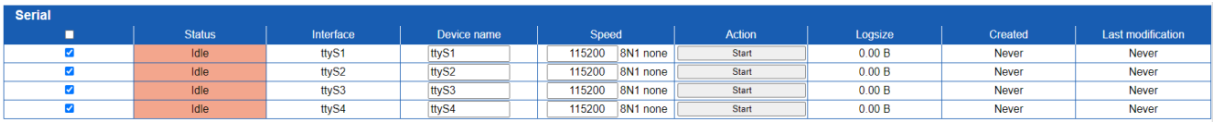


Figure 9: Graphical User Interface- Serial Port Tracing Detail

5.1.1.3 HTTP trace

Connect TraceBox HTTP trace port to device with HTTP traces. The description of the interface is in section 3.2.3.

The HTTP trace interface in graphical user interface is in Figure 10. Set correct remote IP address and port. Optionally change device name in web interface to customize log filename. Click on "*Start*" button in row of device you want to trace. The "*Start*" button will start tracing for the individual device for interface HTTP Traces. To terminate tracing use "Stop" button on individual device.

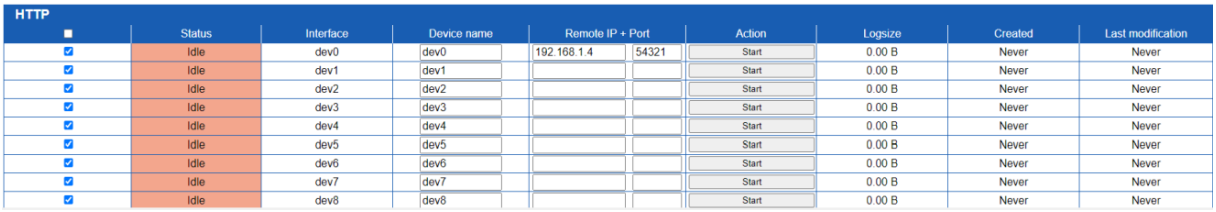


Figure 10: Graphical User Interface- HTTP Traces Detail.

5.1.1.4 Automotive Ethernet

Connect TraceBox between Ethernet nodes you want to trace. To connect the device, follow the instruction in section 3.2.4.

The CAN bus interface in graphical user interface is in Figure 11. Optionally change device name in web interface to customize log filename. Click on "*Start*" button in row of device you want to trace. The "*Start*" button will start tracing for the individual device for interface Ethernet. To terminate tracing use "*Stop*" button on individual device.

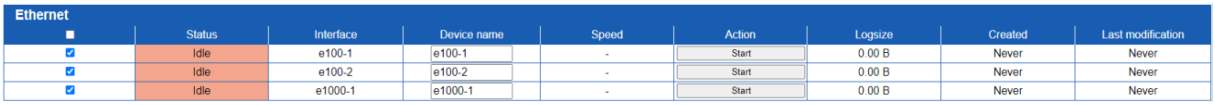


Figure 11: Graphical User Interface – Automotive Ethernet Tracing Detail.

5.1.1.5 Trace all interfaces

TraceBox-Lite enable to trace *all* interfaces and devices in one go. To start tracing all devices on all interfaces use button "*Start all*" in the upper part of GUI. The button "*Stop all*" enables to terminate tracing on all devices. See panel in Figure 12.

Figure 12: Graphical User Interface – Panel with control buttons to start, stop, download traces on all or selected devices.

5.1.2 Save Configuration

The TraceBox-Lite web interface enables to configure what devices are traced by using checkboxes. There is possibility to save these configuration into a file. To save the configuration follow steps:

- Fill configuration name, the configuration file name is specified as item "*Configuration Name*".
- Click on button "*Save Configuration*".

The situation with saving configuration is also depicted in Figure 13.

Figure 13: Graphical User Interface – Panel with control buttons to save and load configuration.

5.1.3 Load Configuration

The save configuration can be loaded when needed. To load the configuration follow steps:

- Click on "*Configuration Name*" text area.
- Select configuration name.
- Click on button "*Load Configuration*".

The situation with loading configuration is also depicted in Figure 13.

5.1.4 Auto Refresh

The TraceBox-Lite supports web page auto-refresh. To enable auto-refresh by sliding the "Auto-refresh" switch. Green color indicates Auto-refresh is enabled. A red icon means that the auto refresh is disabled. The position of "*auto refresh*" icon can be found in Figure 12.

5.1.5 Use relative time

A relative time can be used when tracing. By default, time of an event in traces is recorded as absolute, one line of traces will start with for example – *6.11.2020 10:30*. The relative time tracking uses a time offset specified by user. Slide the "Use relative time" slide switch to enable or disable the relative time

tracking mode (pattern "*before 8 minutes*" will be used). Green color indicates relative time mode is active. The position of "use relative time" icon can be found in Figure 12.

5.2 API

Tracebox can be controlled by third party software using its API. In other words, the API was designed to be accessed with test automation tools. Commands accessible through the API are described in this section.

5.2.1 Connection

The API commands are accessible by standard web browser on the port number 8080.

5.2.2 System

URL	Action
`/info`	Returns actual version of the daemon.
`/shutdown`	Shuts down the tracebox.
`/reboot`	Reboots the tracebox.

5.2.3 Tracing

5.2.3.1 CAN

A command list for CAN interface is shown in Table 2.

URL	Action
`/can/device_list`	Returns available CAN devices
`/can/format_list`	Returns available CAN trace formats
`/can/can0/start`	Starts trace on device *can0*
`/can/can0/stop`	Stops trace on device *can0*
`/can/can0/set_speed/500000`	Sets speed on device *can0* to 500kbit/s
`/can/can0/is_active`	Returns if tracing on device *can0* is active
`/can/can0/change_logfile`	Changes trace output for device *can0* into new file
`/can/can0/info`	Returns *can0* device info
`/can/can0/enable`	Enable device in platform tracing
`/can/can0/disable`	Disable device in platform tracing

URL	Action
/can/can0/status	Returns status of the can0 device
/can/can0/set_name/log_name	Sets name for can0 log file
/set_format/can/can_format	Sets can log file format (“asc” or “pcapng”)
/get_format/can	Returns current log file format (“asc” or “pcapng”)

Table 2: API CAN Interface.

device_list

Possible response: ["can0", "can1", …]

start

Optional arguments:

- speed (Example /can/can0/start?speed=500000)

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already started !" }

stop

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already stopped !" }

set_speed

Possible response: { "return": "0", "status": "OK" }

is_active

Possible responses: { "return": "0", "status": "NOT_ACTIVE" }

{ "return": "1", "status": "ACTIVE" }

change_logfile

Possible response: { "return": "0", "status": "OK" }

info

Possible response: { "interface": "can0", "speed": "500000" }

enable

Possible response: { "return": "0", "status": "OK" }

disable

Possible response: { "return": "0", "status": "OK" }

5.2.3.2 Automotive Ethernet

The

Table 3 shows command list of the interface for Automotive Ethernet.

URL	Action
/ethernet/device_list	List of available Ethernet devices
/ethernet/format_list	Returns available Ethernet trace formats
/ethernet/enp30s0/start	Start tracing the device *enp30s0*
/ethernet/enp30s0/stop	Terminate tracing the device *enp30s0*
/ethernet/enp30s0/is_active	Returns status whether tracing on device *enp30s0* is active
/ethernet/enp30s0/change_logfile	Change of the file where the traces from device *enp30s0* are saved
/ethernet/enp30s0/info	Writes information about the device *enp30s0*
/ethernet/enp30s0/status	Returns status of the enp30s0 device
/ethernet/enp30s0/enable	Enable device in platform tracing
/ethernet/enp30s0/disable	Disable device in platform tracing
/ethernet/enp30s0/set_name/log_name	Sets name for enp30s0 log file

Table 3: API Automotive Ethernet Interface.

device_list

Possible response: ["enp0s3"]

start

Possible response: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already started !" }

stop

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already stopped !" }

is_active

Possible responses: { "return": "0", "status": "NOT_ACTIVE" }

{ "return": "1", "status": "ACTIVE" }

change_logfile

Possible response: { "return": "0", "status": "OK" }

info

Possible response: { "interface": "enp0s3"}

5.2.3.3 HTTP Trace

The Table 4 shows command list of the interface for HTTP.

URL	Akce
/http/device_list	List of available HTTP Traces.
/http/format_list	Returns available HTTP trace formats
/http/dev0/start	Start tracing the device *dev0*
/http/dev0/stop	Terminate tracing the device *dev0*
/http/dev0/set_address/192.168.1.4	Set IP Address for tracing 192.168.1.4 on device *dev0*
/http/dev0/set_port/54321	Set port 54321 for tracing on device *dev0*
/http/dev0/is_active	Returns status whether tracing on device d*ev0* is active
/http/dev0/change_logfile	Change of the file where the traces from device *dev0* are saved
/http/dev0/info	Writes information about the device *dev0*

URL	Akce
/http/dev0/status	Returns status of the dev0 device
/http/dev0/enable	Enable device in platform tracing
/http/dev0/disable	Disable device in platform tracing
/http/dev0/set_name/log_name	Sets name for dev0 log file
/set_format/http/http_format	Sets http log file format (“raw” or “pcapng”)
/get_format/http	Returns current log file format (“raw” or “pcapng”)

Table 4: API HTTP Traces Interface

device_list

Possible response: ["dev2", "dev1", "dev0"]

start

Optional arguments:

- ip_address (Example /http/dev0/start?ip_address=192.168.1.4)
- port (Example /http/dev0/start?port=54321)

Possible response: { "return": "0", "status": "OK" }

stop

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already stopped !" }

set_address

Possible response: { "return": "0", "status": "OK" }

set_port

Possible response: { "return": "0", "status": "OK" }

is_active

Possible responses: { "return": "0", "status": "NOT_ACTIVE" }

{ "return": "1", "status": "ACTIVE" }

change_logfile

Possible response: { "return": "0", "status": "OK" }

info

Possible response: { "ip_address": "192.168.1.4", "port": "54321" }

5.2.3.4 Serial port

The Table 5 shows command list of the interface for Serial Port.

URL	Akce
/serial/device_list	List of available serial devices
/serial/format_list	Returns available HTTP trace formats
/serial/ttyS0/start	Start tracing the device *ttyS0*
/serial/ttyS0/stop	Terminate tracing the device *ttyS0*
/serial/ttyS0/set_speed/115200	Set speed 115200 baud/s on device *ttyS0*
/serial/ttyS0/is_active	Returns status whether tracing on device *ttyS0* is active
/serial/ttyS0/change_logfile	Change of the file where the traces from device *ttyS0* are saved
/serial/ttyS0/info	Writes information about the device *ttyS0*
/serial/ttyS0/status	Returns status of the ttyS0 device
/serial/ttyS0/enable	Enable device in platform tracing
/serial/ttyS0/disable	Disable device in platform tracing
/serial/ttyS0/set_name/log_name	Sets name for ttsyS0 log file
/set_format/serial/serial_format	Sets serial log file format (“log” or “pcapng”)
/get_format/serial	Returns current log file format (“log” or “pcapng”)

Table 5: API Serial Port.

device_list

Possible response: ["ttyS0", "ttyS2", "ttyS1", "ttyS3"]

start

Optional arguments

- speed (Example /serial/ttyS0/start?speed=11500)

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already started !" }

stop

Possible responses: { "return": "0", "status": "OK" }

{ "return": "1", "status": "Device is already stopped !" }

set_speed

Possible response: { "return": "0", "status": "OK" }

is_active

Possible responses: { "return": "0", "status": "NOT_ACTIVE" }

{ "return": "1", "status": "ACTIVE" }

change_logfile

Possible response: { "return": "0", "status": "OK" }

info

Possible response: { "interface": "ttyS0", "speed": "115200" }

5.2.4 Configuration

API part configuration enables to set configuration of individual trace interfaces. There is possibility to save these configuration into a file and load previously saved configuration. The functionality is the save as in the Web interface:

start

Possible response: { "return": "0", "status": "OK" }

{ "return": "1", "status": " `Platform is already started !` " }

stop

Possible response: { "return": "0", "status": "OK" }

{ "return": "1", "status": " `Platform is already stopped !` " }

save

Possible response: { "return": "0", "status": "OK" }

URL	Akce
/platform/start	Starts all interface which are enabled in the platform/configuration
/platform/stop	Stops all interface which are enabled in the platform/configuration
/platform/save/platform_name	Saves current configuration under chosen name
/platform/load/platform_name	Load configuration
/platform/remove/platform_name	Remove configuration
/platform/get/configuration_name	Downloads chosen configuration
/platform/prepare_download/	Compresses and downloads all available log files into one *.zip file
/platform/prepare_download/tar.gz	Compresses and downloads all available log files into one *.tar.gz file
/platform/list	Writes what platforms are available
/platform/is_active	Returns status whether tracing is running

Table 6: API Configuration Interface.

load

Possible response: { "return": "0", "status": "OK" }

remove

Possible response: { "return": "0", "status": "OK" }

prepare_download

Possible response: { "return": "0", "status": "OK" }

list

Possible response: ["platform0", " platform1", " platform2"]

is_active

Possible response: { "return": "0", "status": "NOT_ACTIVE" }

{ "return": "1", "status": "ACTIVE" }

5.2.5 Triggers

The markers in API provides user possibility to mark/indicate an event in the log files. When an event occurs while tracing the MIB, it is possible to label this event by using marker/trigger through the API. The API provides methods to generate such an event when tracing. There will be unique label of this event in log files.

URL	Akce
`/trigger/trigger_name/trigger`	Add trigger/marker from *trigger_name* into log file
`/trigger/device_list`	List of available triggers
`/triger/trigger_name/info`	Writes information about the trigger with name *trigger_name*
`/triger/trigger_name/enable`	Enable trigger *trigger_name*
`/triger/trigger_name/disable`	Disable trigger *trigger_name*
`/triger/trigger_name/set_name/new_name`	Change name *trigger_name* to *new_name*

Table 7: API Marker/Trigger Interface.

trigger

Possible response: { "return": "0", "status": "OK" }

{ "return": "1", "status": "DEVICE_DISABLED" } trigger turned off

info

Possible response: { "enabled": true, "interface": "trigger0", "device_name": "trigger0"}

enable

Possible response: { "return": "0", "status": "OK" }

disable

Possible response: { "return": "0", "status": "OK" }

set_name

Possible response: { "return": "0", "status": "OK" }

5.3 Output file format

Tracebox stores the traced data into several different formats. User can choose “pcapng” format for all devices. Additionaly user can choose also “asc” format for CAN, “log” format for Serial and “raw” formatss for HTTP. Traced files can be read by many applications including CANoe, Wireshark, tcpdump, Snort, and Ethereal.

5.4 System update

1. Open “System Info” from the left menu.
2. Click on button “Update” and you will be redirected to software update page.
3. Browse your filesystem to select package updates.
4. Click to “Update”