Digiteq Automotive

We make car revolution happen

ScreenLinq FPD-Link III

HDMI to FPD-Link III Video Converter

User Manual, Version 1.3.0

Classification: Public
Author: Matěj Bartík (MAB)
Last update: 22.09.25

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. 26466023
VAT ID No.: CZ26466023
registered by the Municipal Court in Prague,
Section C, Insert 84128

MANAGEMENT BOARD
Pavel Štěpánek
Marco Jähnisch

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

A Volkswagen Group Company

Contents

List of Figures

List of Tables

List of Code Listings

Changes

Version	Date	Change description	Changed by	Approved by
0.9.3	26.09.23	Initial revision, no spell-check.	MAB	MAB
0.9.4	12.10.23	Default display profiles update, DFU tutorial, basic spell-check	MAB	MAB
0.9.5	17.10.23	RGB LEDs statuses explained	MAB	MAB
0.9.6	08.02.24	Improved FAQ (RPi 4 issues)	MAB	MAB
0.9.7	28.02.24	Touchscreen operations explained, link/QR code to the product page	MAB	MAB
1.1.0	20.08.24	Heartbeat, display profile management, E-EDID template updated	MAB	MAB
1.1.1	28.08.24	Minor firmware improvements	MAB	MAB
1.1.2	12.09.24	Minor firmware improvements (EEPROM)	MAB	MAB
1.3.0	22.09.25	Dimming application and interface, (AR-)HUD support, ScreenLinq FPD-Link III Multiplexer support. Tutorials have been updated for Windows 11. Described issues with ABT 13.6" vertical.	TPE, MAB	MAB

Preface

About the ScreenLinq

Dear customer,

Thank you for purchasing ScreenLinq. The ScreenLinq presents a powerful tool whenever it is necessary to display any image from your computer on an infotainment display (ABT), selected types of combined (FPK, ILV), or head-up displays in a vehicle interior.

ScreenLinq is intended to be used mainly in the conceptual phase of HMI/UX and application development, during concept presentations, and user testing. ScreenLinq offers processing of user interactions – touches and gestures – with the projected content for UX testing. ScreenLinq can also be used in serial testing when it allows displaying any testing content on ABT display and interacting with the displayed content (e.g., triggering).

ScreenLinq can be used for testing on test benches as well as directly in a real car.

It fully supports modern software and operating systems and is easy to use. To start using ScreenLinq, you only need to connect the relevant cables and select the desired display profile.

In addition, ScreenLinq supports the processing of messages from the CAN-FD, LIN, or USB interfaces and a wide range of supported and preset screen resolutions, including a selection of the most widely used displays in the Volkswagen Group. It is possible to set up further display profiles based on a request.

This user manual provides instructions for the smooth operation of ScreenLinq. We hope that you will be satisfied with our product.

Digiteq Automotive s.r.o.

1 About this User Manual

The company reserves the right to make technical changes to the equipment or this document without 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 publisher's written permission. All technical information, drawings, screenshots, etc., are liable to the law of copyright protection.

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

2 Safety Instructions

The following safety instructions are intended not only for protecting your health but also for protecting the product.

This section overviews all essential safety aspects for protecting individuals and ensuring safe and trouble-free operations.

The warranty/guarantee becomes 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 voids!

Therefore, please read the following items 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 Instructions.
- The products, equipment, and devices must only be used indoors.
- Use the products, equipment, and devices only for their intended purpose as described in the Product Specification.
- The products, equipment, and devices should not be operated in potentially explosive atmospheres.
- During the operation of the products, equipment, and devices, do not permit any work method that hinders the safety of the products, equipment, and devices.

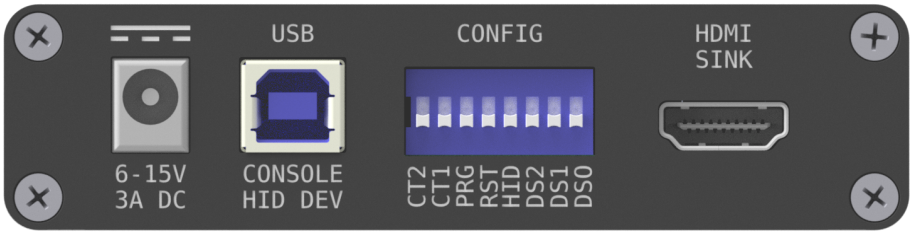
- Always keep the working area of a unit clean and orderly to avoid danger from dirt or scattered parts.
- Do not exceed the technical performance data specified for each product, equipment, and device.
- Keep all safety precautions and danger hazard labels on the products, equipment, and devices in legible condition and replace the descriptions as needed.
- Operation and work on the products, equipment, and devices must only be carried out by trained personnel.
- In case of malfunction, immediately stop the unit.
- Only qualified personnel could approve a unit as operational after a fault occurs.

3 Product Specification

3.1 General Description

The ScreenLinq product family consists of powerful tools for processing video streams (and auxiliary data) from a computer to an in-car display. This particular variant implements an *HDMI to FPD-Link III Video Converter*. Besides this primary function, it can also provide access to the in-car display via several bidirectional communication interfaces to transmit/receive the auxiliary data (touches, gestures, settings, etc.).

(a) Front panel



(b) Rear panel

Figure 1: ScreenLinq's front and rear panel depicting available interfaces.

A ScreenLinq device comes in a **fail-safe** mode (all features are disabled); thus, the device needs to be set-up through the user configuration DIP switch (see Section 3.3.2) located on the rear panel of the device (see Fig. 1b) and a wiring diagram (see Section 4) must be followed as well.

3.2 Mechanical and Electrical Properties

Table 1: Mechanical and electrical properties

Interfaces	HDMI 1.4b Sink (pixel clock up to 210 MHz) FPD-Link III Source (dual-link capable) 2x CAN-FD 1x LIN Master (*reserved for future use*)
Operating Voltage	12 – 15 VDC nominal (6 — 15 VDC when an in-car display has an independent power supply).
Operating Current	Maximum of 100 mA, power consumption of an in-car display is not incorporated. Maximum rating: 3 A in total.
Operating temperature	0 °C to 60 °C while preventing condensation
Storage temperature	-40 °C to 85 °C
Dimensions (w × h × d)	105 × 95 × 26 mm (including connectors and control elements)
Weight	200 g
CAN physical layer	In accordance with ISO 11898
Built-in CAN terminators	120 Ω, activated by DIP switches on the rear panel
EMC Compliance	CISPR 32/EN 55032
Water resistance	IP30

WARNING

In case a ScreenLinq device is used to power an in-car display, the ScreenLinq's input voltage must not exceed the operating voltage of the display used.

3.3 Interfaces

3.3.1 Power Input

ScreenLinq uses an industry-standard power barrel jack with dimensions of 2.1 mm inner diameter and 5.5 mm outer diameter (see Fig. 2). The power input accepts a DC (Direct

Current) power, where positive voltage is tied to the center pin of the power barrel jack connector.

The operating voltage (see Tab. 1) is 12 – 15 VDC. The ScreenLinq can withstand surges or peaks up to 25 VDC for a short time, allowing the ScreenLinq to connect to a car's on-board power grid easily.

It is recommended to use the provided CUI Inc. SWI36-12-E-P5[1] AC/DC power adapter (see Fig. 7) In case the provided AC/DC adapter can not be used; it is necessary to provide sufficient power when ScreenLinq is used to provide power to an in-car display. A typical current drawn by an in-car display usually ranges between 1.5 and 2 amperes.

Figure 2: A power barrel jack 2.1/5.5 mm.

3.3.2 User DIP Switch, Button, and RGB LEDs

To allow user interaction with ScreenLinq, several ways exist to provide control and visualize status information. The primary user configuration interface is the DIP switch (see Fig. 3) located on the rear panel (see Fig. 1b). The factory state of the DIP switch is all *OFF* state. To transit an individual switch to *ON* state, it is needed to slide the respective switch toward the white **ON** marking. The DIP switch positions **DS2, DS1, and DS0** are used to select a desired display profile stored in a configuration memory (see factory pre-loaded profiles – Tab. 5).

There is the **USER** button (also labeled as *INIT* in the first hardware revision) on the front panel (see Fig. 1a). In the current firmware release (1.3.0) pressing the USER button triggers an action (see Table 2).

However, some cabling might need to be reattached to propagate the new settings to a computer (generating a Hot-Plug event, for example) or to an in-car display.

Alternatively, the USER button can be re-configured to serve different purposes, such as restarting an in-car display, generating user events (triggers), and possibly more.

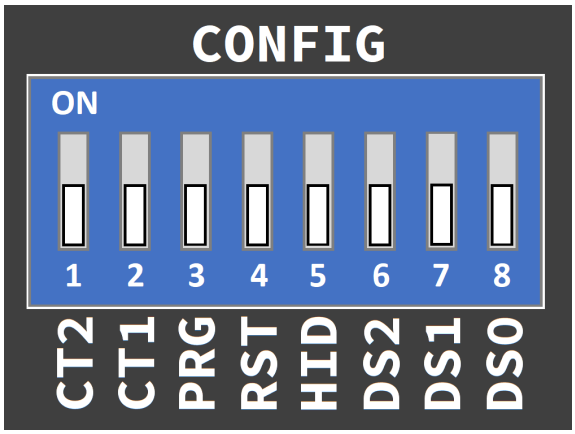
Besides the USER button, two tri-colored (RGB) LEDs are on the front panel to indicate the device's status. As the LED labels indicate, each visualizes the status of the ScreenLinq device's respective part.

[1] https://www.cui.com/product/external-ac-dc-power-supplies/wall-plug/swi36-e-series/swi36-12-e

Table 2: User button actions (functions)

User button actions (functions)		
Pressed duration	**Action triggered**	**FPD-L3 Status LED**
0 < t < 1 seconds	Transmits a device identification over JSON based interface (see Section 4.6)	Glowing white steadily
1 < t < 4 sec	Restarts a ScreenLinq unit	Blinking white slowly
4 < t < 7 sec	Clears all user uploaded display profiles and reloads pre-loaded profiles (see Table 5) instead	Blinking white rapidly
7 < t sec	Cancellation of any action	LED is off
	Selected action is going to executed	Glowing red

The device will start with both LED glowing blue. The device is considered fully operational when both LEDs are green. For more details, see Table 3.



CT2	CAN-FD Interface 2 Termination	**HID**	USB HID Enable
CT1	CAN-FD Interface 1 Termination	**DS2**	Display Select (bit 2)
PRG	Programming mode	**DS1**	Display Select (bit 1)
RST	Reset (device in idle)	**DS0**	Display Select (bit 0)

Figure 3: User configuration DIP switch (factory defaults).

Table 3: Device status based on RGB LEDs

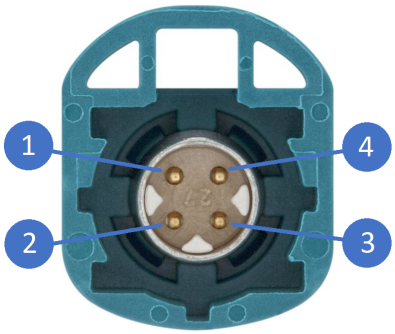
Device status based on RGB LEDs			
FPD-L3 Status		**CAN FD Status**	
Colour	**Meaning**	**Colour**	**Meaning**
Blue	Idle (Powered-up)	Blinking Blue	Idle (Powered-up)
Green	HDMI & FPD-Link III interfaces are locked and provide video	Blinking Green	Normal (transmitting CAN-FD messages)
Red	HDMI & FPD-Link III interface are not locked	Blinking Yellow	Malfunction of FRAM (I^2C3 Bus) but still transmitting CAN-FD messages
		Green & Yellow alternating	CAN simulation active, but *Touches & Gestures* format wasn't recognized
		Red	Malfunction of CAN-FD interface(s)
		Green	HDMI cable needs to be connected

3.3.3 FPD-Link III interface

ScreenLinq utilizes *Flat Panel Display Link* technology in its third generation (referred to as FPD-Link III[2]) to implement a video source interface. The implementation of the FPD-Link III interface supports a dual link (see pin-out on Fig. 4) feature, which enables image resolution up to 2880x1080 at 60 FPS (for example). The maximum pixel clock of a video stream is 210 MHz. Please note an attached sink device (display) might not be capable to receive the maximum pixel clock frequency. The physical layer uses *Shielded Twisted Pairs* with *LVDS*[3].

ScreenLinq's FPD-Link III interface uses Rosenberger HSD[4] connectors (*D4S20D-40MA5-Z* is used) and cables commonly used by in-car displays. The used solution is also compatible with other products of Digiteq Automotive, such as Modular FrameGrabber (MGB)[5] or FrameGraber4 (FG4)[6].

[2] https://www.ti.com/lit/an/slyt581/slyt581.pdf
[3] Low Voltage Differential Signaling
[4] https://www.rosenberger.com/product/rosenberger-hsd/
[5] https://www.digiteqautomotive.com/en/product/modular-framegrabber-mgb
[6] https://www.digiteqautomotive.com/en/product/fg4-pcie-card



FPD-Link III Signals

1	DOUT0 (−)	4	DOUT1 (+)
2	DOUT1 (−)	3	DOUT0 (+)

Figure 4: Pin-out of HSD connector used for the FPD-Link III source interface.

3.3.4 HDMI

A standard 19-pin (Type-A) HDMI connector (see Fig. 5) is present to implement a video sink device to receive a multimedia stream from a computer. ScreenLinq implements the HDMI 1.4b standard. It is also possible to use other video interfaces, such as DisplayPort (with DP++ feature) or DVI (limited by a maximum pixel clock of 165 MHz).

Since HDMI 1.4b pixel clock can reach the maximum of 340 MHz, an incoming video stream from the HDMI interface must respect the pixel clock limit of the FPD-Link III interface (see Section 3.3.3).

Some HDMI 1.4b features, such as the Ethernet channel (HEC) or Audio Return Channel (ARC), are not supported.

> **!** **WARNING**
>
> We highly recommend not to use USB-C to HDMI adapters (or USB-C docking stations) as they are generally incapable of generating pixel clock accurately. Use a native HDMI/DP/DVI port instead.

3.3.5 CAN-FD Connector

The CAN-FD Connector (using industry standardized D-Sub (Canon) 9 connector) provides two CAN-FD[7] interfaces (CAN-FD is fully backward compatible with CAN Classic), one LIN[8] (Master) interface, and a power output (VB+) which equals the ScreenLinq's input voltage (see Section 3.3.1) minus a drop-out on a reverse-blocking diode ($\approx -0.5\ V$). Due to this

[7] https://www.can-cia.org/can-knowledge/can/can-fd/
[8] https://lin-cia.org/standards/

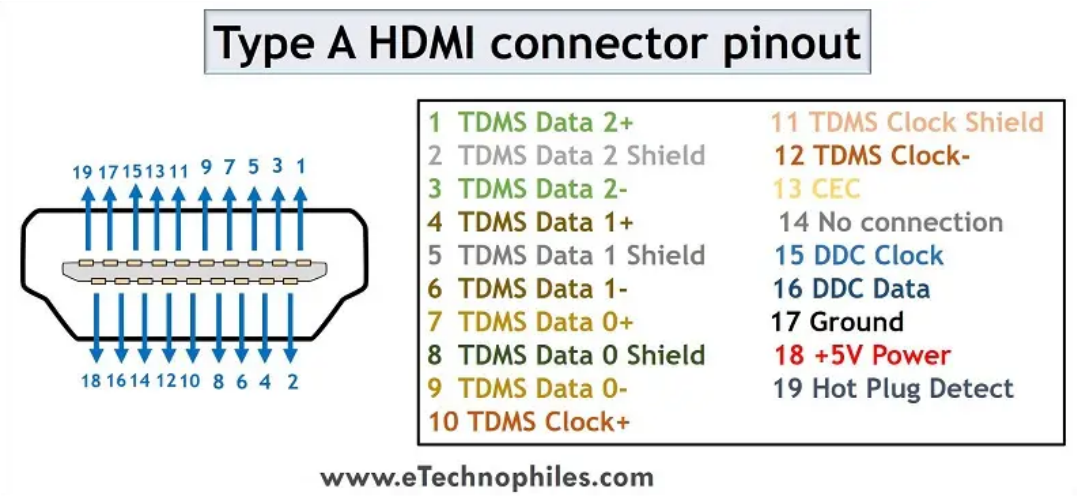


Figure 5: Pin-out of HDMI connector.

fact and also to the fact that a car on-board power grid utilizes a voltage slightly higher than +12 V, an in-car display can be dual-powered from a provided AC/DC adapter and the car on-board grid simultaneously.

The respective pin-out (see Fig. 6) is (partially compatible) with products from Vector[9] or PEAK-System[10] to enable a seem-less integration into existing projects. Each of the CAN-FD interfaces has been internally terminated with a 120 Ω resistor, which can be (de)activated by CT1 (or CT2 respectively) DIP switch position (see Section 3.3.2).

> **!**
>
> **WARNING**
>
> Do not attach or detach the D-Sub (Canon) 9 connector when a ScreenLinq device is powered on. Shielding of the D-Sub (Canon) 9 connector may cause a short circuit between VB+ and ground pins.
>
> In case the power needs to be re-applied to an in-car display, re-attach the connector on the in-car display's side.

CAN-FD interfaces implement necessary commands to keep an in-car display in a power-on state. In addition to this, they also receive commands from the in-car display and translate user-triggered events (such as touches and gestures) to USB HID[11] commands (see Section 3.3.6).

[9] https://www.vector.com/int/en/products/products-a-z/hardware/network-interfaces/cables/can-cables/
[10] https://www.peak-system.com/Contact.54.0.html?&L=1
[11] https://www.usb.org/sites/default/files/hid1_11.pdf

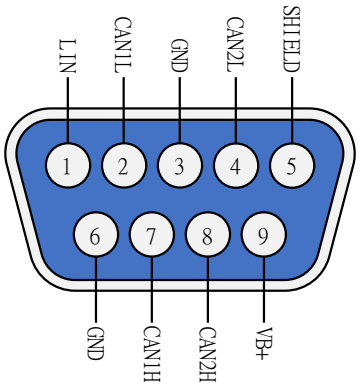


Figure 6: Pin-out of the ScreenLinq's CAN-FD Connector.

3.3.6 USB interface

ScreenLinq implements a USB 2.0 (Full-speed) device with different interfaces and end-points depending on the feature selected via the user DIP switch, see Section 3.3.2:

- USB HID class provides touches or gestures to an operating system (see Section 4.5),
- USB DFU class enables updating a new firmware (see Appendix B),
- USB CDC class allows JSON based communication:
  - Uploading or downloading display profiles via our application (see Section 4.3.3),
  - Accessing communication over CAN-FD and LIN interfaces, CAN simulation parameters, and ScreenLinq unit properties via a custom user application (see Section 4.6).

4 Operations

This section provides various helpful information to operate your ScreenLinq device successfully, including fundamental procedures such as:

- Describing the provided cables and accessories,
- Wiring the ScreenLinq to a computer or an in-car display,
- Configuring the ScreenLinq device,
- Using custom display profiles,
- Uploading a new firmware,
- Troubleshooting a problem.

4.1 Package Content

Besides this user manual, ScreenLinq's package (see Fig. 7) contains the device itself, an AC/DC power adapter, three data cables (HDMI, HSD, USB), and a Power & CAN-FD harness.

1	ScreenLinq device	4	AC/DC adapter 230/12 V
2	HSD cable 2.5 m	5	HDMI cable 2 m
3	Power & CAN-FD harness 2.5 m	6	USB cable (A-to-B type) 2 m

Figure 7: ScreenLinq package content.

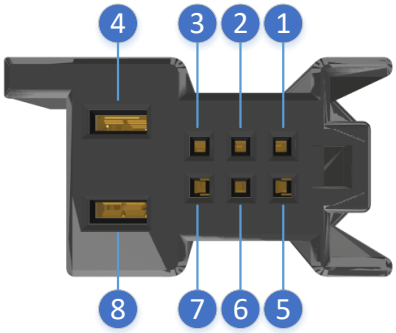
4.1.1 Power & CAN-FD harness

The provided Power & CAN-FD harness (DQ P/N: 153-101-YYY-ZZZ) has two D-Sub (Canon 9) female connectors, which are equal in function and pin-out (see Fig. 6); therefore, either of them can be attached to the ScreenLinq's CAN-FD connector (see Section 3.3.5).

The remaining D-Sub (Canon 9) female connector can be used to provide access to a compatible system (CAN analyzer, for example).

The last connector of the Power & CAN-FD harness is a block connector (see Fig. 8) intended to provide power and CAN-FD 1 interface to an in-car display. Such block connector (P/N: 1K0.972.924) has been used in several car platforms in the Volkswagen group.

In case an in-car display uses a different connector, check a list of existing alternative harnesses (see Table 4) or a new harness should be designed to fit your needs. In that case, please don't hesitate to contact us for additional support.



4	VB+ (KL.30)	3	*n.c.*	2	*n.c.*	1	CAN-FD High
8	GND (KL.31)	7	*n.c.*	6	*n.c.*	5	CAN-FD Low

Figure 8: Pin-out of the in-car display block connector (P/N: 1K0.972.924).

Table 4: Available harnesses

Available harnesses for ScreenLinq		
Harness type	**Terminal**	**DQ product code**
i.ID/FID	1EA.035.750	DQ 151-102-YYY-ZZZ
Old platforms	8W0.972.112	DQ 152-102-YYY-ZZZ
Standard	1K0.972.924	DQ 153-102-YYY-ZZZ
AR-HUD Bundle	5Q0.972.706	DQ 157-100-YYY-ZZZ

4.2 User Configuration & Wiring Diagram

Prior to the first run (or when a different configuration is required), the ScreenLinq device needs to be configured via the *user configuration DIP switch* (see Section 3.3.2). The configuration setup can be divided into three separate steps:

1. Select an operation mode:
   - Normal operation mode (video interfaces are active) (RST=0, PRG=0)
     - USB HID function activated (HID=1)
     - USB HID function disabled (HID=0)
   - Firmware update mode (RST=0, PRG=1, HID=X[12])
   - Idle mode (RST=1, PRG=X, HID=X)
2. Activate or deactivate termination for CAN-FD interfaces – it is recommended to activate termination for the used CAN-FD interface by the *CT1* or *CT2* switch because

[12] **X** – represents *Don't Care* value.

a majority of in-car displays expect the termination to be implemented by a head unit (such as ICAS3). In case no such unit is attached to the ScreenLinq device nor the in-car display, the CAN-FD interface **must** be terminated.

3. Select the desired in-car display profile via DS2, DS1, and DS0 switches. It is possible to invoke a custom user profile or a pre-loaded one (see Tab. 5).

With the ScreenLinq properly configured, it is possible to proceed to a wiring diagram (see Fig. 9) as the first step has already been completed. The numbers in the wiring diagram represent both step number and part number (they match each other) as they were described (see Fig. 7) in the previous section.

For certain operation modes, there might be some deviations from the general wiring diagram:

- For *normal mode*, attaching the USB cable is optional (USB HID commands or auxiliary data cannot be processed).
- For *firmware update mode*, attaching an in-car display is optional.
- For *idle mode*, all cabling beside the power input is optional.

4.3 Using (Custom) Display Profiles

ScreenLinq has a total of eight display profile slots, each can be used to upload (or download) custom profiles to support many in-car displays. Thanks to the effort of the ScreenLinq's team, a desired profile might exist already.

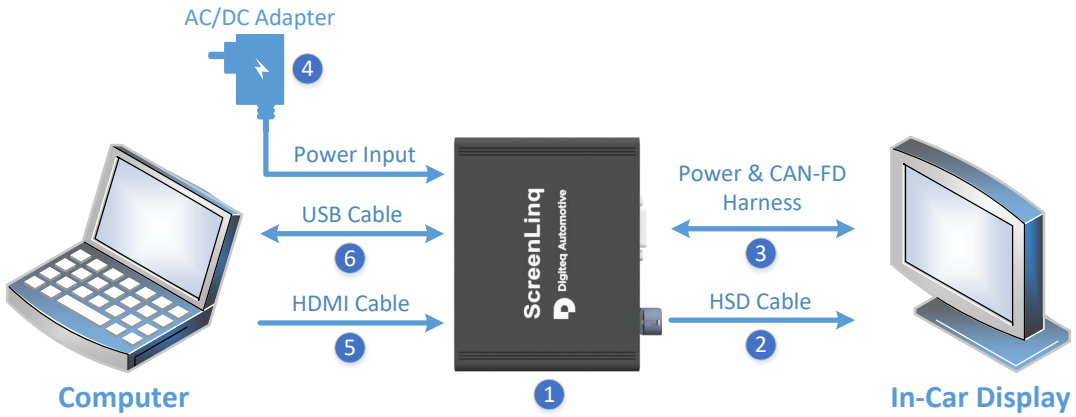
If the desired profile is unavailable, you can ask the ScreenLinq's team to create such profiles (see Appendix A) or create your own display profiles, as explained below.

4.3.1 Setting Up a New (Custom User) Display Profile

Before uploading a custom user display profile into a ScreenLinq device, it is necessary to create the profile first. This step requires knowledge of video timing parameters (see Appendix A). ScreenLinq's display profiles have been using *VESA Enhanced - Extended Display Identification Data (E-EDID) format*[13]. ScreenLinq supports E-EDIDs in version 1.3 and 1.4 with a maximal size of 256 bytes.

It is recommended to use an E-EDID editor, such as DELTACAST.TV E-EDID Editor[14] alongside a provided E-EDID template (see Listing 3) to prevent creating an invalid E-EDID structure.

[13] https://glenwing.github.io/docs/
[14] https://www.deltacast.tv/products/free-software/e-edid-editor



1. Configure a ScreenLinq device via the *User Configuration DIP Switch* (see Section 4.2) including individual steps such as:
   1. Select an operating mode,
   2. Select an in-car display profile,
   3. Activate CAN-FD 1 interface termination by sliding **CT1** switch to **ON** position,
2. Attach *HSD cable* to both the ScreenLinq and an in-car display,
3. Attach *Power & CAN-FD harness* (see Section 4.1.1) to both the ScreenLinq and an in-car display (both D-Sub (Canon 9) female connectors of the harness are equal),
4. Apply power to ScreenLinq device and wait for blue LEDs indicating a *Ready* status,
5. Attach *HDMI cable* to both the ScreenLinq and a computer **(it's not recommended to use USB-C to HDMI converters or USB-C docking stations as they are often unable to generate precise video timing causing issues – use a native HDMI port instead)**,
6. Attach *USB cable* to both the ScreenLinq and a computer when these features are required:
   1. Processing of USB HID commands or auxiliary data over USB interface,
   2. Uploading or downloading a display profile,
   3. Updating a firmware.

!

WARNING

In case the HDMI cable is attached to a computer before powering the ScreenLinq device up, an incorrect (generic) display profile (see Fig. 17 with it's timing parameters or respective E-EDID as Listing 2) might be loaded instead the desired display profile.

Figure 9: ScreenLinq's wiring diagram including instructions for basic operation.

Table 5: Pre-loaded in-car displays parameters (firmware version 1.3.0)

Profile Slot	DS2	DS1	DS0	ResX (Pixels)	ResY (Lines)	Pixel Clock (MHz)	In-car Displays	Part Number(s)
0	0	0	0	1560	700	77.85	VW 10.0", VW 12.0", SA 13.1" (H01, H03, H05, H07, H08) with scaler	10A.919.606.K 10A.919.606.T 5LA.919.606(.A)
1	0	0	1	1442[1,2]	700	77.85	SA 13.1" (H01, H03, H05, H07, H08) with scaler	5LA.919.606(.A)
2	0	1	0	1920	1080	136.90	SA 13.1" UNECE (H07, H21, H22)	5LA.919.606(.B)
3	0	1	1	1920	932[1,3]	136.90	SA 13.1" UNECE (H07, H21, H22)	5LA.919.606(.B)
4	1	0	0	1920	1080	136.83	SA 12.9", SA E3 13.0" (H32)	3P0.919.605 XXX.XXX.XXX.X
5	1	0	1	1560	878	92.26	SA 10.4", SA E3 10.0" (H30, H40)	XXX.XXX.XXX.X 14A.919.603
6	1	1	0	1280	640	61.43	SA 9.2" (H05)[4]	655.919.606.A
7	1	1	1	2240	1260	178.42	VW E3 15.0" (H43)[5]	14A.919.606(.A)

1. Such display profile is not guaranteed to work as the video signal parameters deviates from the original source's parameters.
2. Reduced horizontal resolution from 1560 px to match visible area.
3. Reduced vertical resolution from 1080 px to match visible area.
4. Requires a different *Power & CAN-FD harness* (DQ P/N: 152-XXX-YYY-ZZZ).
5. Might not work as a significant number of laptops (or docking stations) have a built-in resolution or pixel clock limit[a].

[a] https://www.dell.com/support/kbdoc/en-us/000126548/resolution-on-external-monitor-limited-to-1920-x-1080-using-hdmi

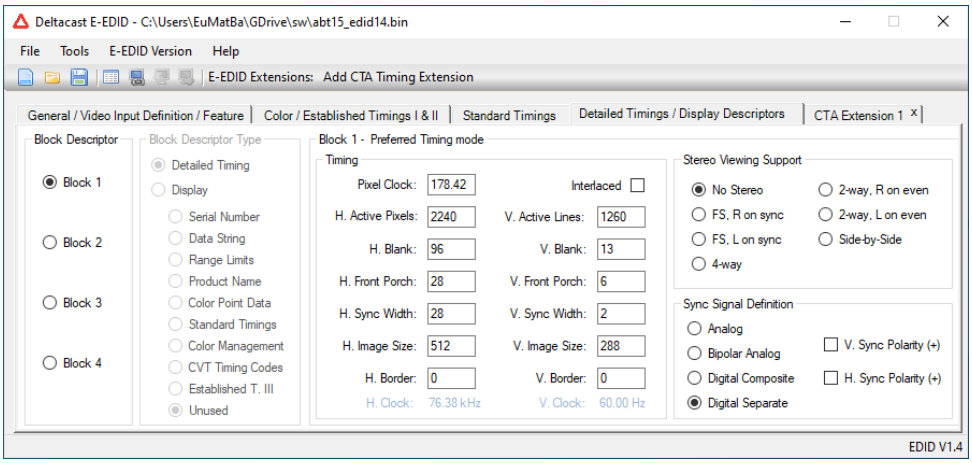


Figure 10: GUI of the DELTACAST.TV E-EDID Editor.

However, it is also possible to create an E-EDID structure from scratch. The recommended E-EDID editor allows to set various details, including video timing parameters (see Fig 10).

> **!** **WARNING**
>
> Uploading an invalid E-EDID structure to a ScreenLinq device will cause a GPU driver display identification process to fail.

E-EDID structure also supports *CTA Extensions*, which can be used to specify additional features. One of the features is bridging up to eight audio streams over the FPD-Link III interface (see Section 3.3.3) from the HDMI interface (see Section 3.3.4).

> **!** **WARNING**
>
> Advanced features such as CAN simulations require to add some additional information (to *Vendor Specific* blocks) via use of the *CTA Extensions*. Without such blocks, only video will be generated. Please check our documentation for creating a custom user display profile.

4.3.2 ScreenLinq Application

The ScreenLinq application (see Section 5) is designed to show, upload, and download display profiles; shows connected ScreenLinq devices (including their details). To use the application, the ScreenLinq device must be in *Normal operation mode*. The basic command syntax follows:

```
screenlinq [-c <com_port_device> | -l]
           profile <options...> |
           system <options...>
```

It is possible to list all connected devices with option **-l**:.

```
screenlinq -l
Found devices:
COM3
```

It this example, the only ScreenLinq device is identified on a serial port **COM3**. Therefore, it is possible to manage the device with following commands:

> **WARNING**
>
> Don't forget to change a serial port identifier to match your setup.

```
screenlinq -c COM3 system information [-t <timeout_in_s>=2]
```

This command shows basic information (a hardware and firmware version) about the ScreenLinq device. There is an optional parameter **-t** which specifies a timeout for the command (default value is 2 seconds to finish the issued command).

```
screenlinq -c COM3 system information
system information:
hardware version: 1.0
firmware version: 1.3
json schema version: 0.4
schema url: https://products.digiteqautomotive.com/schemas/
                    screenlinq/COM.protocol.0.4.schema.json
```

4.3.3 Uploading and Downloading a Display Profile to ScreenLinq

A display profile can be uploaded to one of eight display profile slots, where the content of the selected slot will be overwritten (including the pre-loaded display profiles – see Table 5). To see the profiles currently present in the ScreenLinq device, issue a command:

```
screenlinq -c COM3 profile list
```

The **list** command prints a table of present display profiles (see Fig. 11).

To upload a display profile (or multiple profiles), issue command:

```
screenlinq -c COM3 profile profile upload [-t <timeout_in_s>=2]
 <profile_number> <edid_file>
[<profile_number> <edid_file> [...]]
```

```
d:\>screenlinq -c COM3 profile list
+-----------------------------------------------------------------------------------------------------+
| Available profiles:                                                                                 |
|-----------------------------------------------------------------------------------------------------|
| profile slot | resolution   | screen size | part number    | data string      | ECU      | CAN BUS |
|-----------------------------------------------------------------------------------------------------|
|      0       | 1560x700  px| 10.0"       | 10A.919.606.T  | "H132S6081P01"   | ABT      | MIB CAN |
|      1       | 1442x700  px| 13.1"       | 5LA.919.606.A  | "H007S0226P01"   | ABT      | MIB CAN |
|      2       | 1920x1080 px| 13.1"       | 5LA.919.606.B  | "H023S0226P01"   | ABT      | MIB CAN |
|      3       | 1920x932  px| 13.1"       | 5LA.919.606.B  | "H023S0226P01"   | ABT      | MIB CAN |
|      4       | 1920x1080 px| 12.9"       | 3P0.919.605    | "HH30SC031P01"   | ABT      | MIB CAN |
|      5       | 1560x878  px| 10.3"       | 14A.919.603    | "HH20SB010P01"   | ABT      | MIB CAN |
|      6       | 1280x640  px|  9.2"       | 655.919.606.A  | "HH05S0081P01"   | ABT      | MIB CAN |
|      7       | 2240x1260 px| 15.0"       | 14A.919.606.A  | "H030SA030P02"   | ABT      | MIB CAN |
+-----------------------------------------------------------------------------------------------------+
```

Figure 11: ScreenLinq application – Table of available display profiles.

where command arguments are following:

1. **-t** optional timeout for the command,
2. **<profile_number> <edid_file>** a pair of arguments describing a profile slot and a filename of an E-EDID file of a display profile.
3. the last pair of arguments can be issued multiple times to upload multiple profiles at the same time.

```
screenlinq -c COM3 profile upload 0 5LA.920.320.C_SA_iID_5i3.bin
```

This example upload a profile (with an **i.ID** display profile) to the profile slot **0**. It is possible to verify the display profile has been uploaded successfully by issuing **profile list** command (see Fig. 12).

```
d:\>screenlinq -c COM3 profile upload 0 5LA.920.320.C_SA_iID_5i3_800x400px_20240813.bin

d:\>screenlinq -c COM3 profile list
+-----------------------------------------------------------------------------------------------------+
| Available profiles:                                                                                 |
|-----------------------------------------------------------------------------------------------------|
| profile slot | resolution   | screen size | part number    | data string      | ECU      | CAN BUS |
|-----------------------------------------------------------------------------------------------------|
|      0       |  800x400  px|  5.3"       | 5LA.920.320.C  | "H031S0630P01"   | iID/FID  | AB CAN  |
|      1       | 1442x700  px| 13.1"       | 5LA.919.606.A  | "H007S0226P01"   | ABT      | MIB CAN |
```

Figure 12: ScreenLinq application – Checking the uploaded profile.

It is also possible to download display profiles to your computer for checking or editing such profiles.

```
screenlinq -c COM3 profile profile download [-t <timeout_in_s>=2]
 <profile_number> <edid_file>
[<profile_number> <edid_file> [...]]
```

1. **-t** optional timeout for the command,
2. **<profile_number> <edid_file>** a pair of arguments describing a profile slot which content will be downloaded to a file with a given filename.
3. the last pair of arguments can be issued multiple times to download multiple profiles at the same time.

To reload the pre-loaded display profiles back to their original slots it is possible to issue command **profile reset**:

```
screenlinq -c COM3 profile profile reset
[-t <timeout_in_s>=2]
all | profile number [profile number]
```

It is possible to reload every available slot with argument **all** or to specify which profile slot(s) will be restored. As usual, there is **-t** optional argument for the timeout.

4.4 Uploading a New Firmware

This topic is covered in the Appendix B.

4.5 Configuring and Using USB HID Function on a Computer

To activate the feature, a ScreenLinq unit must be in *Normal operation mode* with *USB HID function* activated (see Section 4.2). Beside that, the ScreenLinq unit must be connected with a USB cable (see Fig. 9) to the same computer which provides an image signal over HDMI interface. The HDMI cable should be attached prior to the attachment of the USB cable (see Fig. 9).

Such devices (a display and a USB HID peripheral) shall be detected by an operating system and paired automatically. This process is further explained in the Appendix C.

4.6 JSON Based Communication and Control Interface over USB

ScreenLinq implements a JSON-based communication interface available via a USB VCP (Virtual COM Port) to manage a unit's settings, manage display profiles, or to adjust simulation properties (via CAN-FD and LIN messages) from user space.

To activate the feature, a ScreenLinq unit must be in *Normal operation mode*. Now, it is possible to open a COM port on your computer with suitable software (PuTTY[15], for example). The COM port setting follows:

[15] https://www.putty.org

- Baudrate: 115200
- Databits: 8
- Stopbits: 1
- Parity: None
- Flow control: None

The implemented protocol (see the latest available JSON schema[16]) provides access to the display's system information as well its events, such as touch, key, slide, etc. However, please keep in mind that only some displays are currently supported, and only some of their features.

> **INFO**
>
> ScreenLinq Communication JSON Schema version 0.4 has been implemented and supported in the current firmware (version 1.3.0).

Beside the ScreenLinq's product page, a link to a JSON schema can be obtained via the ScreenLinq application (see Section 4.3.2), with the command arguments **system information** to be specific.

4.7 Heartbeat Status Message

ScreenLinq unit supports remote monitoring of it's status. CAN message is being transmitted over both interfaces. The used bitrate respects a setting of an used CAN simulation (see Section 4.8). Properties of such message follows:

> **INFO**
>
> A signals definitions file is available for Vector CANoe[a]. It can be downloaded from ScreenLinq's product page (see Section 5).
>
> [a] https://www.vector.com/int/en/products/products-a-z/software/canoe/

- Format: CAN 2.0B (backward compatible with CAN-FD format)
- CAN ID: 0x1FFFEF01 (29-bit Extended)
- Cycle time: 1000 ms
- Data length: 8 bytes:

[16] https://products.digiteqautomotive.com/schemas/screenlinq/COM.protocol.0.4.schema.json

0. FPD-Link III serializer status (byte 0) (see Table 6)
1. FPD-Link III serializer status (byte 1) (see Table 7)
2. State of switches located on the rear panel (see Section 3.3.2 an Table 8)
3. State of user profiles (see Table 9)
4. Reserved for future use
5. Reserved for future use
6. Reserved for future use
7. Reserved for future use

Table 6: Heartbeat status message (byte 0): FPD-Link III serializer status (byte 0)

Bit(s)	Description
7	HDMI RX +5V detect: Indicates status of the RX_5V pin. When asserted, indicates the HDMI interface has detected valid voltage on the RX_5V input.
6	HDMI Interrupt Status: Indicates an HDMI Interrupt is pending.
5	Reserved
4	Initialization Done: Initialization sequence has completed. This step will complete after configuration complete (2^{th} bit).
3	Remote EDID Loaded: Indicates EDID SRAM has been loaded from a remote EDID EEPROM device over the Bidirectional Control Channel. The (0^{th} bit) value indicates if the EDID load was successful.
2	Configuration Complete: Indicates automatic configuration has completed. This step will complete prior to initialization complete (4^{th} bit).
1	Configuration checksum status: Indicates result of Configuration checksum during initialization. The device verifies the 2's complement checksum in the last 128 bytes of the EEPROM. A value of 1 indicates the checksum passed.
0	EDID checksum Status: Indicates result of EDID checksum during EDID initialization. The device verifies the 2's complement checksum in the first 256 bytes of the EEPROM. A value of 1 indicates the checksum passed.

Table 7: Heartbeat status message (byte 1): FPD-Link III serializer status (byte 1)

Bit(s)	Description
7	Bit value 1 indicates that the FPD-Link III has detected a valid downstream connection and determined capabilities for the downstream link.
6	This bit indicates that the FPD-Link III transmitter is active and the receiver is LOCKED to the transmit clock. It is only asserted once a valid input has been detected, and the FPD-Link III transmit connection has entered the correct mode (Single vs. Dual mode).
5:4	FPD-Link III Port Status: If the 6th bit is set to a 1, this field indicates the port mode status as follows: 00: Dual FPD-Link III Transmitter mode. 01: Single FPD-Link III Transmit on port 0. 10: Single FPD-Link III Transmit on port 1. 11: Replicate FPD-Link III Transmit on both ports.
3	HDMI TMDS Valid: Bit value 1 indicates the TMDS interface is recovering valid TMDS data from the HDMI interface.
2	HDMI PLL lock status: Bit value 1 indicates the HDMI PLL has locked to the incoming HDMI clock.
1	No HDMI Clock Detected: Bit value 1 indicates the Frequency Detect circuit did not detect an HDMI clock greater than 6 MHz
0	HDMI Frequency is Stable: Indicates the Frequency Detection circuit has detected a stable HDMI clock frequency

Table 8: Heartbeat status message (byte 2): State of switches

Bit(s)	Description
7:4	Reserved
3	USB **HID** position.
2:0	Display (profile) Select positions **DS2:DS0**.

Table 9: Heartbeat status message (byte 3): State of user profiles

Bit(s)	Description
7:0	Each bit position indicates a respective display profile slot (see Table 5) contains a display profile (EDID) with invalid checksum.

4.8 CAN Simulation

There are three CAN simulations implemented in the ScreenLinq FPD-Link III firmware. When a simulation has been selected, a CAN interface(s) parameters are set accordingly. The available options are explained in detail in the document *"EDID Guideline - How to make a display profile"* (see Section 5).

Currently supported CAN simulations:

- None
- MIB CAN (CAN-FD 500/2000)
- AB CAN (CAN-FD 500/500)

> **!**
>
> **INFO**
>
> In case the *"None"* CAN simulation has been used, CAN interface(s) will be configured with the following (fail-safe) setting:
>
> - Format: CAN 2.0B
> - Bitrate: 500 kbit/s

Such CAN simulations support these types of displays:

- ABT (CID)
- CDD
- i.ID/FID
- HUD
- PID

4.8.1 Adjusting CAN simulation's signals

There are three independent methods of adjusting a CAN message/signal of a CAN simulation:

1. A display profile preset (automatically loaded during power-up).
2. A communication over the JSON based interface (see Section 4.6).
   - ScreenLinq Dimming Application provides a user-friendly GUI for the JSON based interface.
3. A ScreenLinq request–response protocol over CAN interface.

Depending on the used CAN simulation (which has been selected by a display profile), some simulation signals (see Table 10) can be adjusted while simulation running. The signals values are loaded from the respective display profile if they are present (see Section 4.3.1). Otherwise, the simulation starts with defaults values (see Table. 10). Then, the signal's values can be adjusted during runtime via the *ScreenLinq Dimming Application* or *ScreenLinq request–response protocol*.

Table 10: List of adjustable signals of a CAN simulation

Sim.	Message	Signal	Sig. ID	Length	Default value
MIB	Dimmung_01	DI_KL_58xd	0x01	1 Byte	0xFD
MIB	Dimmung_01	DI_KL_58xs	0x02	1 Byte	0x64
MIB	Dimmung_01	DI_Display_Nachtdesign	0x03	1 Byte	0x00
MIB	Dimmung_01	DI_KL_58xt	0x04	1 Byte	0x64
AB	RLS_02	LS_Helligkeit_FW	0x05	2 Bytes	0xFD; 0x03
AB	RLS_02	RLS_Vorfeldhelligkeit_Boost	0x06	1 Byte	0x0F

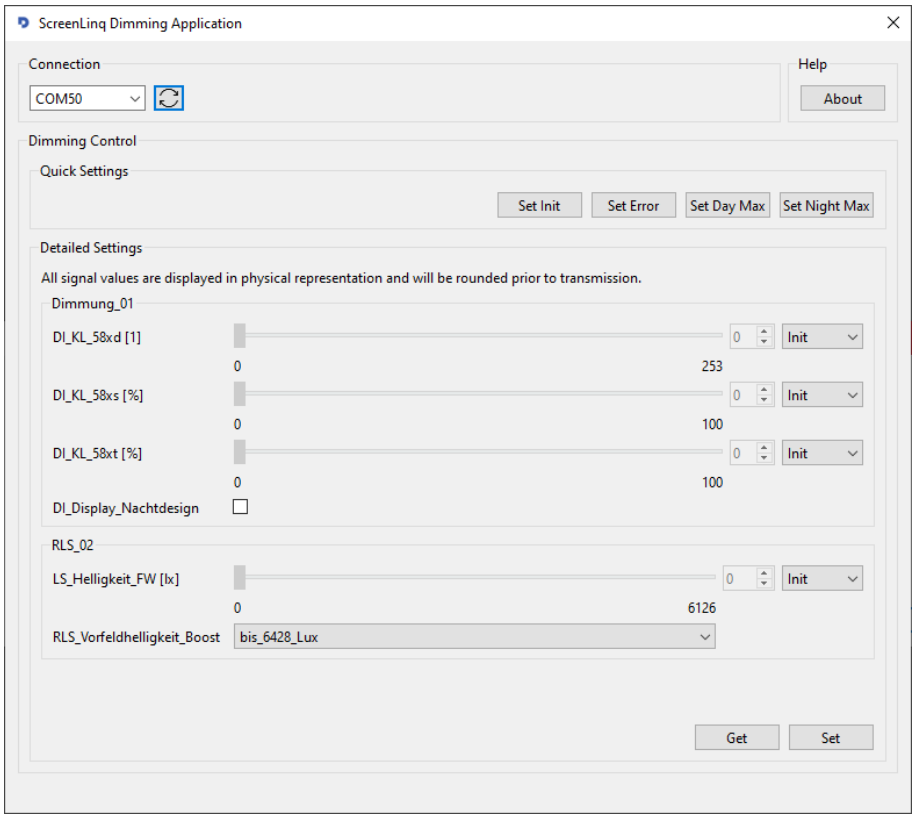


Figure 13: ScreenLinq Dimming Application.

ScreenLinq Dimming Application

The *ScreenLinq Dimming Application* (see Fig. 13) was designed to provide a user-friendly way of adjusting CAN simulation signal's values over the ScreenLinq JSON based interface (see Section 4.6). The application automatically detects all connected ScreenLinq devices (and provides a list of available ScreenLinq's *COM* ports) on start-up.

The application has two main areas with control elements: *"Quick Settings"* and *"Detailed Settings"*. In *"Quick Settings"* area, each button sets the signals values to predefined values (see Table 11).

In *"Detailed Settings"* area, each region represent one individual signals. The gray colored signal's name indicates the respective signal is not part of the used CAN simulation; thus, can't be adjusted. If a signal is part of the active CAN simulation, but it has an inactive slider, pick a **Number** option from a drop down menu (see Fig. 14).

Figure 14: ScreenLinq Dimming Application – Detail of a drop down menu.

A status bar at the bottom indicates a state of each user interaction with the application (see Fig. 15).

Figure 15: Status bar messages.

Table 11: Signal's values of certain presets (*"Quick Settings"*)

Message	Signal	Set Init	Set Error	Set Day Max	Set Night Max
Dimmung_01	DI_KL_58xd	INIT	FEHLER	253	100
Dimmung_01	DI_KL_58xs	INIT	FEHLER	100	100
Dimmung_01	DI_Display_Nacht-design	N/A	N/A	0	1
Dimmung_01	DI_KL_58xt	INIT	FEHLER	100	100
RLS_02	LS_Helligkeit_FW	INIT	FEHLER	6126	0
RLS_02	RLS_Vorfeld-helligkeit_Boost	N/A	N/A	15	0

ScreenLinq request–response protocol

ScreenLinq implements its own protocol which can be used for configuration and monitoring a device over a CAN interface(s). A subsequent part can be used for adjusting the CAN simulation signals. To request a service from a ScreenLinq, a message CAN ID (extended format) *0x1FFFEF08* need to be transmitted. A message with a response has CAN ID (extended format) *0x1FFFEF09*.

The response message respects the CAN format of the request (if a request was transmitted in CAN-FD format, a respective response will be transmitted back also in CAN-FD format). Examples of the a message structure when no segmentation is needed are presented below.

> **!**
>
> **INFO**
>
> Please keep in mind a value with a range greater than 256 (1 byte) must be encoded in the Little-endian[a] format across multiple bytes. For example, value of *1021* (0x03FD hexadecimally) will encoded as *0xFD; 0x03* bytes.
>
> [a] https://en.wikipedia.org/wiki/Endianness

Set signal's value request *(ID = 0x1FFFEF08)*:

```
0x02 0x0C <SIGNAL_ID> <SIGNAL_VALUE>
```

Get signal's value request *(ID = 0x1FFFEF08)*:

```
0x01 0x0C <SIGNAL_ID>
```

Response format (when no errors occur) *(ID = 0x1FFFEF09)*:

```
0x24 0x0C <SIGNAL_ID> <SIGNAL_VALUE>
```

If an error occurred, the response format is *(ID = 0x1FFFEF09)*:

```
0x1F 0x0C <ERROR_CODE> <SIGNAL_ID>
```

where the error codes have the following meaning:

- 0x01: wrong length of the requested signal value,
- 0x02: unknown signal id,
- 0x03: the signal with the requested id is not currently transmitted,
- 0x04: the requested signal value is wrong (most probably out of range).

Beside these messages, an asynchronous notification message will be transmitted is a signal has been adjusted by another method (e.g. ScreenLinq Dimming Application) with the following structure *(ID = 0x1FFFEF09)*:

```
0x23 0x0C <SIGNAL_ID> <SIGNAL_VALUE>
```

INFO

A signals definitions file is available for Vector CANoe[a]. It can be downloaded from ScreenLinq's product page (see Section 5).

[a] https://www.vector.com/int/en/products/products-a-z/software/canoe/

4.9 Cooperation with ScreenLinq FPD-Link III Multiplexer

It is possible to make a setup of ScreenLinq FPD-Link III and ScreenLinq FPD-Link III Multiplexer devices. A system harness is provided as a part of the ScreenLinq FPD-Link III Multiplexer delivery.

The ScreenLinq FPD-Link III attempts to receive the multiplexer's status message from its both CAN interfaces (see Section 3.3.5). When successful, the converter automatically disables the running CAN simulation (including HID events) on the other CAN interface. The CAN interface on which the detection has been successful at first stays the only active interface. However, both the CAN simulation and HID events are *independently* adjusted according to the state of specific signals of the Multiplexer's status message.

The adjustment is processed according to the following bullets:

- CAN simulation:
  - If the Multiplexer's status message indicates that CAN is switched to channel 1, the device sends no simulation.
  - If the Multiplexer's status message indicates that CAN is switched to channel 2, the device sends its simulation (see Section 4.8) with no restriction.
- HID events:
  - If the Multiplexer's status message indicates that HID is switched to channel 1, the device sends no simulation.
  - If the Multiplexer's status message indicates that HID is switched to channel 2, the device processes HID events (see Section 4.5).
- The ScreenLinq status message (see Section 4.7) is always transmitted.

Please check the ScreenLinq FPD-Link III Multiplexer's User manual for obtaining more details regarding these topics (but not limited to):

- configuring and monitoring a ScreenLinq FPD-Link III Multiplexer,
- ScreenLinq FPD-Link III Multiplexer status message's format,
- routing between physical and logical channels (video signals; CAN interfaces; simulations; HID events).

5 Troubleshooting, Support, & Resources

This section presents a list of known issues and their possible causes or solutions. The list has been divided into sections dealing with general issues and issues related to a specific in-car display.

Whenever is possible, update a ScreenLinq device to the latest firmware, which can be found in the respective product page (among other information) at `https://products.digiteqautomotive.com/screenlinq/` (also available as a QR code – see Fig. 16).

Figure 16: QR code for accessing the ScreenLinq's product page.

Still looking for a solution?

In case you are unable to find an answer to your question, do not hesitate to contact us via email: support.products@digiteqautomotive.com

5.1 General Issues

Issue 1: ScreenLinq's user LEDs are not glowing still or blinking

Possible causes:

- ScreenLinq device is not (properly) powered.
- ScreenLinq device is in *Idle* or *Firmware update mode*.
- ScreenLinq's internal configuration process has failed.
- ScreenLinq device is faulty.

Issue 2: ScreenLinq is not identified by an OS or a GPU driver at all

Possible causes:

- ScreenLinq device is not (properly) powered.
- HDMI cable is not attached properly.
- ScreenLinq's internal configuration process has failed.
- ScreenLinq device is faulty.

Issue 3: ScreenLinq is identified by an OS or a GPU driver, but the resolution can not be set to any value

Possible causes:

- Display profile with an invalid E-EDID structure has been selected (re-upload a valid E-EDID structure).
- E-EDID 1.3 structure is used for pixel clock greater than 165 MHz (switch to E-EDID 1.4 structure instead).
- Resolution or pixel clock exceeds the HDMI interface's capabilities (see Note 4 in Table 5 and contact the manufacturer of such HDMI source for additional support).
- ScreenLinq device is faulty.

Issue 4: ScreenLinq is being identified by an OS or a GPU driver with a different resolution than requested

- Does the resolution match other display profiles? If so, detach the HDMI cable and press INIT (or power cycle the ScreenLinq device) to load the new configuration and attach the HDMI cable back.
- Is ScreenLinq recognized as **TI-DS90Ux949** device? Is the primary resolution 1280x720 (see Fig. 17 for additional video timing information)? If so, detach the HDMI cable, power cycle the ScreenLinq device, and attach the HDMI cable back.

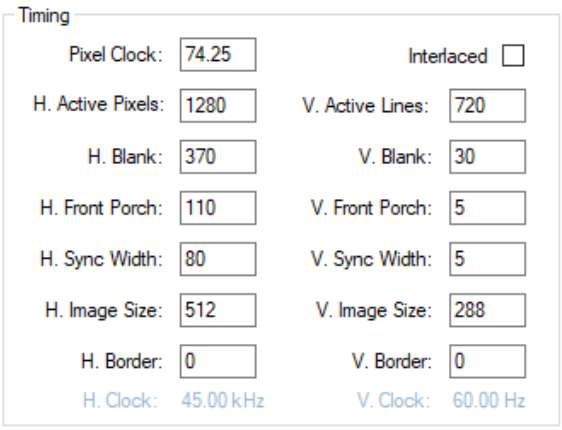


Figure 17: Video timing of the generic display profile.

Issue 5: ScreenLinq is being correctly identified by an OS or a GPU driver, but no image can be seen

Possible causes:

- An in-car display might not be powered up – check Power & CAN-FD harness (see Fig. 7, number 3) connectors are attached properly.
- CAN-FD 1 interface might not be terminated; therefore, an in-car display has not been awakened from sleep mode – Slide **CT1** switch on the rear panel (see Fig. 1b) to **ON** position to activate termination for the CAN-FD 1 interface.
- HSD cable is not attached properly.
- Some in-car displays can (incorrectly) lock onto an FPD-Link III signal with no active video stream; thus, they cannot re-lock when a valid video signal appears. Power cycle such in-car display via re-attaching display block connector (see Fig. 8).

Issue 6: I can see an image on an in-car display, but it looks like a “gray grain“

A GPU driver (nVidia driver seems to be the most problematic) starts a video stream and temporarily stops the stream, causing the FPD-Link III interface to glitch. Consequently, an FPD-Link III deserializer on an in-car display side cannot lock (and generates the “gray grain“ image).

To rectify such behavior, there are several options:

- Power cycle the in-car display via re-attaching the display block connector (see Fig. 8).
- Re-attach the HDMI cable.
- Try a different computer.

Issue 7: I can see image on an in-car display, but there is a black padding surrounding the image

An OS or a GPU driver has set up a *Cloning* (Duplication) for secondary available display (ScreenLinq), but resolution does not match the primary display. Therefore, the image is being scaled to the ScreenLinq’s resolution while keeping the aspect ratio. When aspect ratio differs, a black padding is being added to the image.

To rectify such behavior, use displays in *Extended* (Individual) mode or adjust the resolution on the primary display to match the ScreenLinq’s resolution.

Issue 8: I can see image on an in-car display, but it is “unstable“

Possible causes:

- If an experimental display profile is used, switch to a profile that matches the video timing parameters of the display’s original head unit.
- An incorrect display profile has been selected.
- A display profile is intended for a different hardware or software revision listed as supported.

Issue 9: There are graphical artifacts or display doesn't behave correctly

Such behavior is often caused by USB-C to HDMI adapters (and by many docking stations) as such adapters are unable to generate precise video timing.

Connecting ScreenLinq to a native HDMI port of a graphic card of your laptop/PC whenever is possible is highly recommended solution.

Issue 10: Image has been corrupted while using Raspberry Pi 4 or newer

Please check the used profile works correctly while using a regular GPU of your computer. It is likely the profile contains horizontal timing parameters non-divisible by two resulting that Raspberry Pi HDMI pipeline can not generate an image correctly[a].

[a] https://www.raspberrypi.com/documentation/computers/config_txt.html#raspberry-pi-4-hdmi-pipeline

Issue 11: ScreenLinq Application fails to work (running, but no output visible etc.)

Please make sure that appropriate C++ runtime has been installed on your PC. You can check it in Settings / Apps / Installed apps[a]. The correct one is "Microsoft Visual C++ 2015-2022 Redistributable (x64)" or newer (see Fig. 18). The installer of the latest version is available here[b].

[a] ms-settings:appsfeatures
[b] https://aka.ms/vs/17/release/vc_redist.x64.exe

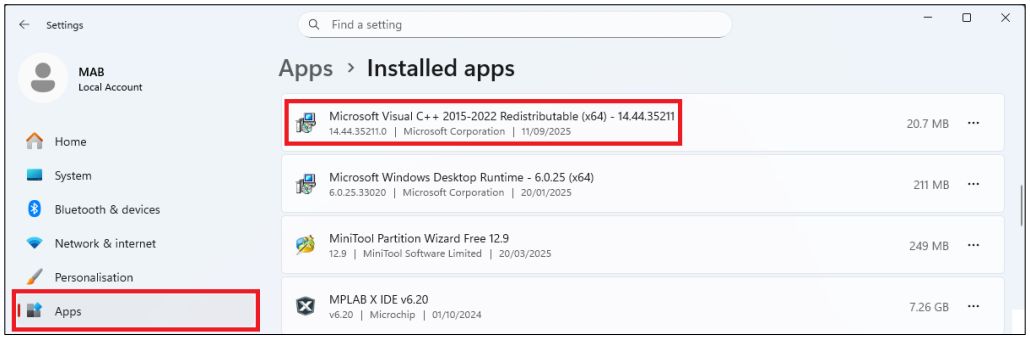


Figure 18: The proper version installed in the Windows 11 Apps manager.

5.2 Issues Related to a Specific In-Car Display

Issue 12: Skoda Auto ABT 13.1"

The Skoda Auto ABT 13.1" has been produced in many variants (both hardware and software), where only the 9 most common combinations are known to the ScreenLinq's team.

Furthermore, this display is prone to generate a visual error when the exact display timing is not met (which could differ between each revision). Also, relying solely on hardware revision to determine a display profile is impossible, as used image resolution can vary within one hardware revision.

Issue 13: Volkswagen ABT 15.0"

The Volkswagen ABT 15.0" display stays black, and ScreenLinq is not recognized correctly by an OS or a GPU driver. The problem is caused by many laptop vendors, which limit the capabilities of an HDMI source interface to a certain resolution (FullHD/1080p usually).

It is possible to use the VW 10.0" or SA 12.9" profile to test the setup/wiring, but the displayed image will be corrupted.

If you were able to display any image, the only solution is to find a computer with no such limitations. It seems discrete GPUs in a PC have no such limits.

Issue 14: Skoda Auto 13.6" Vertical

At this moment, we reported two issues of the ABT (*P/N: 50A.919.606*) back to the Skoda Auto and the ABT's supplier. We hope a newer firmware will be released with these two issues rectified.

It seems the ABT is going to lock onto an unstable video signal (introduced by a graphic card) and can't recover even after receiving a valid video signal after updating the ABT's firmware to version higher than the *D030*.

Therefore, there are two options how to operate this ABT with ScreenLinq FPD-Link III successfully:

- Use a graphic card which doesn't generate glitches at the beginning of the HDMI transmission.
- Use an ABT with firmware revision *D030* or lower.
- Use a ScreenLinq FPD-Link III Repeater *(DQ P/N: 155-XXX-XXX-XXX)* to filter out glitches from the incoming video signal.
- Use an adapter from DisplayPort/USB-C to HDMI (despite the general recommendation not to use such adapters with the ScreenLinq FPD-Link III device).

Appendix A: Request for a New In-Car Display Profile

In case an in-car display is not supported (respectively, the display profile does not exist), it is possible to tailor a custom user profile to fit your needs. To create such a profile, the following information are required:

1. Part number of an in-car display,
2. Video timing parameters of the used display/screen/LCD panel:
   - These parameters are known to you, and it is possible to provide them to us (via filling up a form – see Table 12).
   - You have a matching head unit (ICAS3, HCP3, for example) and the in-car display in a working setup, and you are willing to share the setup with us for the measurement to obtain the video timing parameters.

   Please keep in mind that the resolution of an active video (visible area) is not sufficient to create a display profile.
3. Any other known details such as manuals, datasheets, car platform, hardware revision, or software version (see Fig. 20).

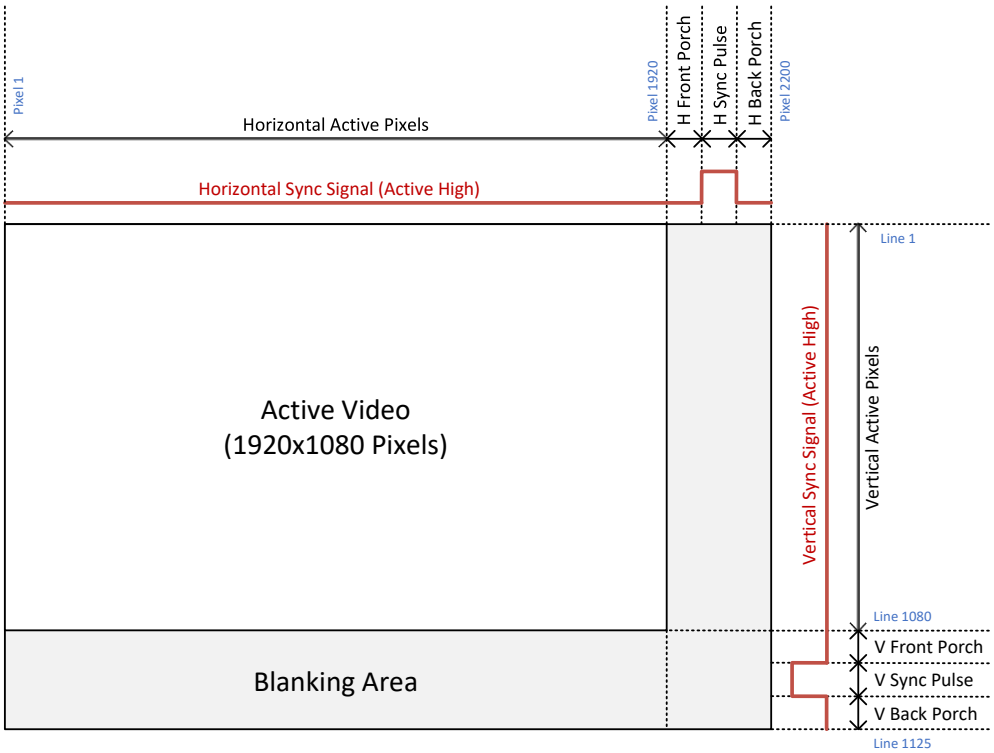


Figure 19: Visualization of video timing parameters - Full HD (1080p) resolution as defined by CEA-861 standard.

Table 12: Parameters required for creating a custom display profile

Video Timing Parameter	Parameter Value	Unit
Pixel Clock		Hz
Framerate		Hz (FPS)
H Active		Pixels
H Total		Pixels
H Blanking		Pixels
H Front Porch		Pixels
H Back Porch		Pixels
H Sync Pulse		Pixels
H Sync Polarity[1]		Logical Level
V Active		Lines
V Total		Lines
V Blanking		Lines
V Front Porch		Lines
V Back Porch		Lines
V Sync Pulse		Lines
V Sync Polarity[1]		Logical Level
Display Parameter	**Parameter Value**	**Example/Unit**
Part number		XXX.XXX.XXX[.X]
Screen size (active area) – X axis		Millimeters
Screen size (active area) – Y axis		Millimeters
Hardware revision		3-digit code
Software revision		4-digit code
Car platform		MEB; MQB

1. Only two values are possible: Active-Low or Active-High logic levels.

In case some of the parameters are unknown, they still can be calculated from other parameters by following these equations:

$$H_{Blanking} = H_{Front\ Porch} + H_{Back\ Porch} + H_{Sync\ Pulse}$$

$$H_{Total} = H_{Active} + H_{Blanking}$$

$$V_{Blanking} = V_{Front\ Porch} + V_{Back\ Porch} + V_{Sync\ Pulse}$$

$$V_{Total} = V_{Active} + V_{Blanking}$$

$$Pixel\ Clock = H_{Total} \times V_{Total} \times Framerate$$

Figure 20: An example of a sticker (of an in-car display) with depicted part number (5E3.920.770), software version (M230), and hardware version (E03).

Appendix B: How to upload a firmware via USB

To upload a firmware to your ScreenLinq device, you need to make these four steps:

1. Download and unpack a provided ZIP archive (see an example of a ZIP archive's file system on Listing 1), which usually includes:
   - A driver for ScreenLinq (an STM32 Bootloader driver),
   - A firmware upload utility (dfu-util)[17],
   - A DFU file containing the ScreenLinq's firmware,
   - A batch file for easy execution of firmware upload process,
   - A file for the CANoe software with signals definitions,
   - An executable file of the ScreenLinq application.
2. Configure and power-on the ScreenLinq device into "Firmware update mode".
3. Install the driver by following a standard procedure.
4. Upload a firmware via the DFU interface.

```
[DIR]  ScreenLinq_Firmware
[DIR]  ScreenLinq_Dimming_App
[DIR]  STM32_Bootloader_Driver
[FILE] flash_screenlinq_fw_v1.3.0.bat
[FILE] screenlinq.exe
[FILE] ScreenLinq_User_Manual_v1.3.0.pdf
[FILE] ScreenLinq_Protocol_v1.0.dbc
[FILE] ScreenLinq_StatusMessage_v1.3.0.dbc
```

Listing 1: Example of a the ZIP archive's file-system.

B.1 Invoking ScreenLinq's "Firmware update mode"

To activate ScreenLinq's *"Firmware update mode"*, find a switch *"PRG"* on the *User DIP switch* (see Section 3.3.2) on the rear side of your ScreenLinq device (see Fig.1b) and slide it into **ON** position. Then, power-on the ScreenLinq device and connect it to your computer with the provided USB cable.

B.2 Installing an STM32 Bootloader Driver for a ScreenLinq Device

To gain access to *Device Firmware Upgrade (DFU)* interface over a USB interface, it is necessary to install an appropriate driver (STM32 Bootloader) as they might not be included nor supported by your operating system.

[17] https://dfu-util.sourceforge.net

> **WARNING**
>
> You might need administration rights to install or update a driver. In case you don't have such rights, contact an IT department of your organization to let them install the driver on your computer.

In the case of Microsoft Windows 11, a ScreenLinq device will appear in *Device Manager* as a device with a *Warning* symbol (see Fig. 21), indicating that such a device might not operate as desired.

The STM32 Bootloader driver can be installed by following steps:

1. Invoking ScreenLinq's *"Firmware update mode"* (see Section B.1).
2. Find a ScreenLinq device in *Device Manager* (see Fig. 21),
3. Use the option *Browse my computer for drivers* to pick the driver manually (see Fig. 22).
4. Browse for the *"STM32_Bootloader_Driver"* directory (have the option *"Include sub-folders"* checked (see Fig. 23) and press *Next*).
5. The driver should now be installed successfully (see Fig. 24 and Fig. 25).
6. Close all dialogues and *Device Manager* itself.

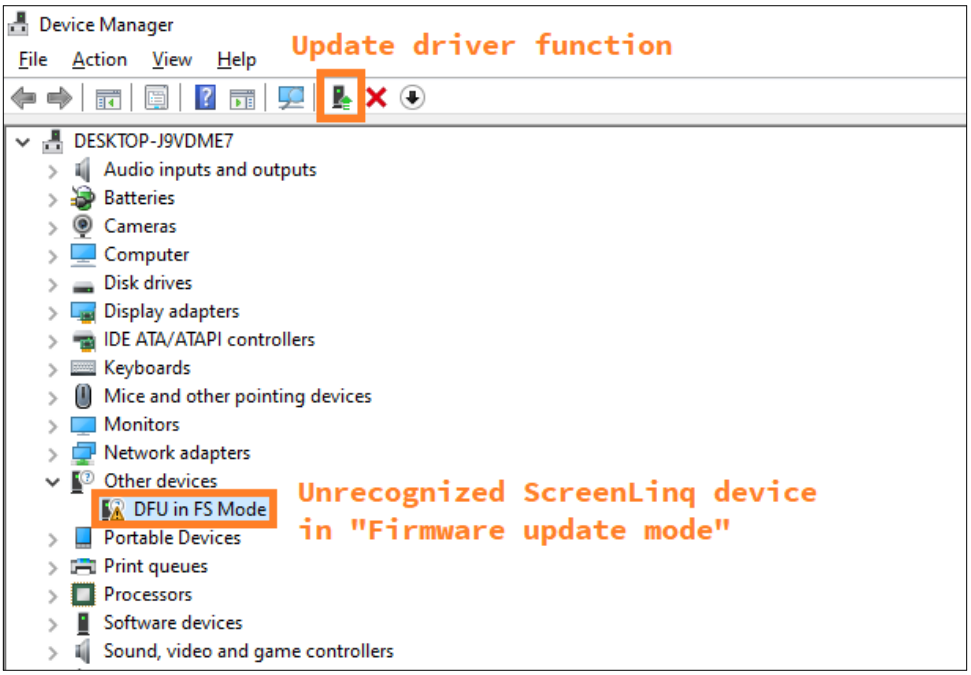


Figure 21: An unrecognized ScreenLinq device in Windows 11 Device Manager.

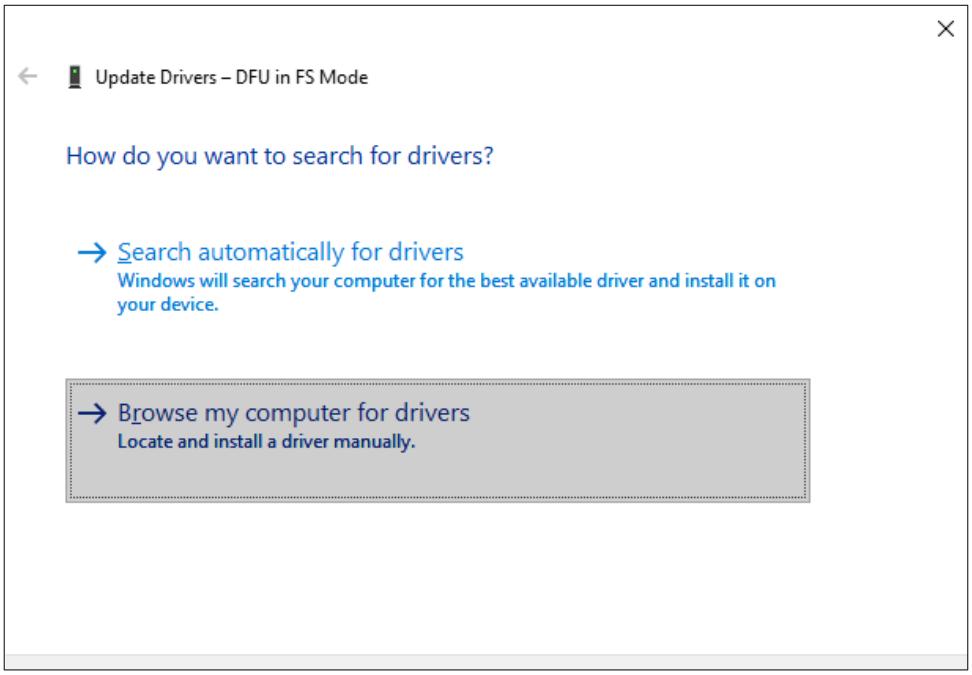


Figure 22: Use the option *"Browse my computer for drivers"* to pick the driver manually.

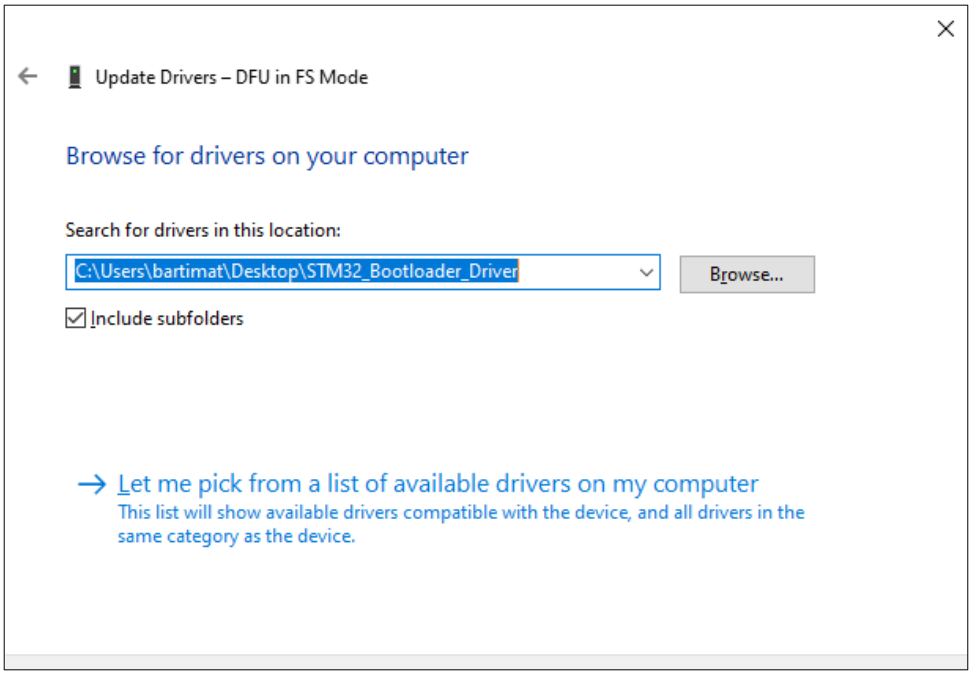


Figure 23: Locate the provided *"STM32_Bootloader_Driver"* directory.

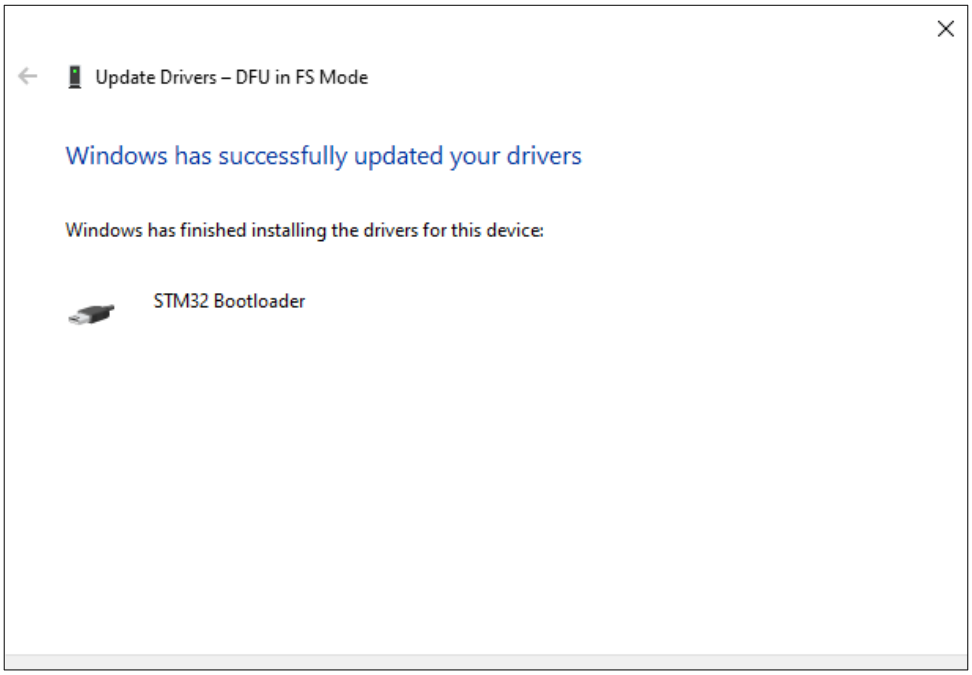


Figure 24: The STM32 Bootloader driver has been installed successfully.

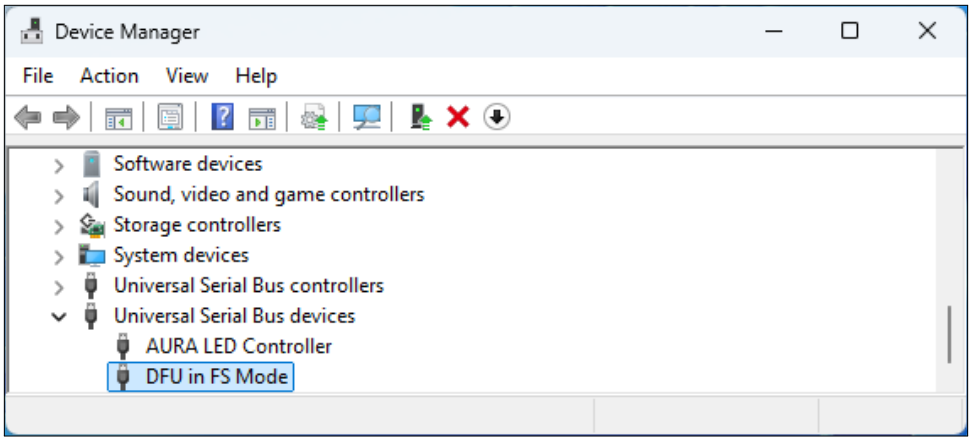


Figure 25: The ScreenLinq device in the *Firmware update mode* has been correctly recognized by the Windows 11.

B.3 Uploading a Firmware via DFU Interface

To upload a firmware (provided as DFU file) to your ScreenLinq device, it is necessary to invoke the *"Firmware update mode"* (see Section B.1) first. Therefore, the second step is to execute **flash_screenlinq_fw_v1.3.0.bat** simply by double-clicking on the batch file with a computer's mouse. The bat file will execute the *dfu-util* with the required parameters to flash the ScreenLinq device successfully (see Fig. 26 for an example run).

WARNING

Do not forget to deactivate ScreenLinq's *"Firmware update mode"* in the end by sliding *"PRG"* switch to **OFF** position and to power-off ScreenLinq device before using other ScreenLinq's operating modes (see Section 4.2).

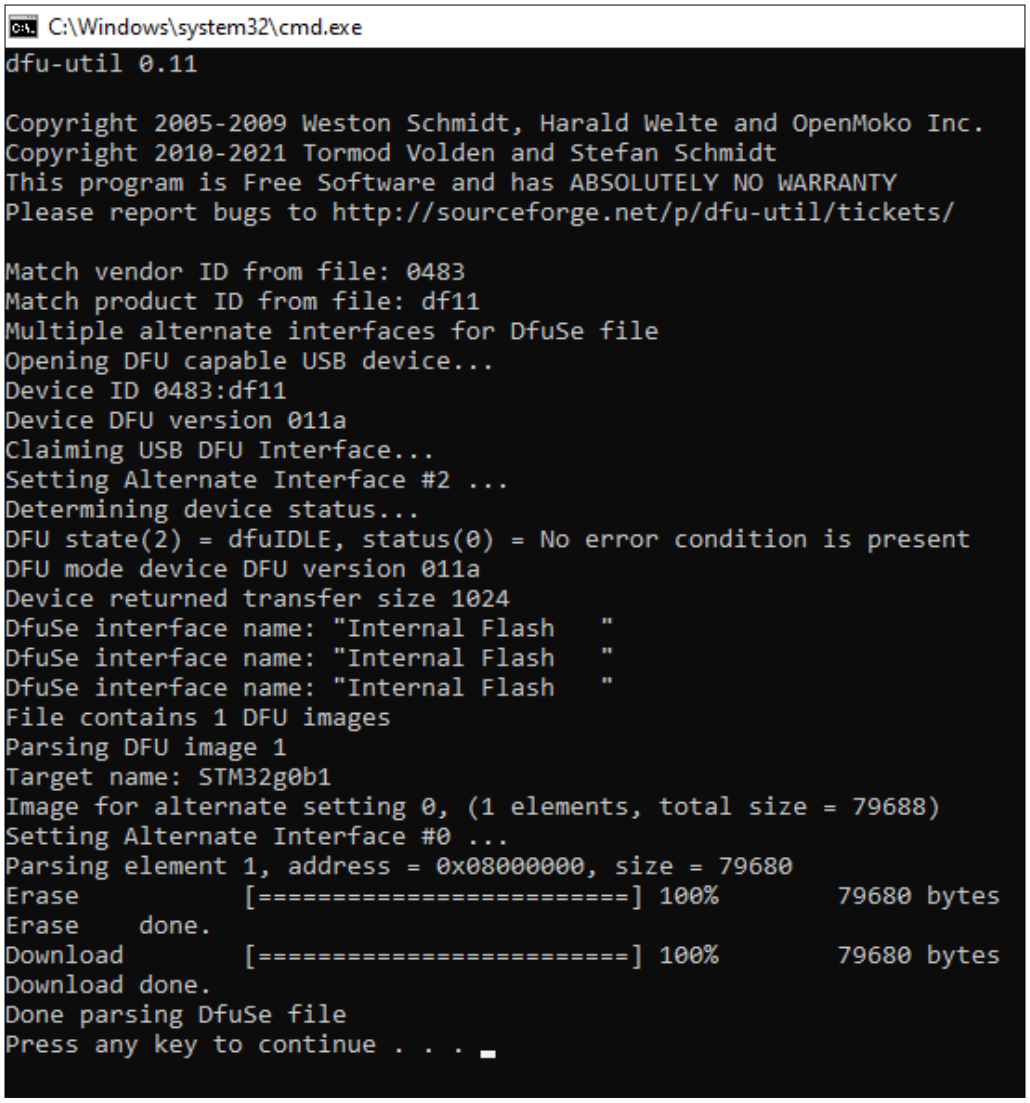


Figure 26: An example of executing the batch file successfully including the **dfu-util** command.

Appendix C: How to Synchronize Image from Screen and Touches from Digitizer

First, please make sure your ScreenLinq device has been detected fully in operating system you use. In the particular case of Windows 11, you can make a check via *"Settings"* and *"Bluetooth & other devices"* dialogues (see Fig. 27) as two new devices should appear on the list:

- Generic Monitor (indicating a P/N of the connected display)
- DQ Automotive ScreenLinq Touchscreen (indicating a resolution of the connected display)

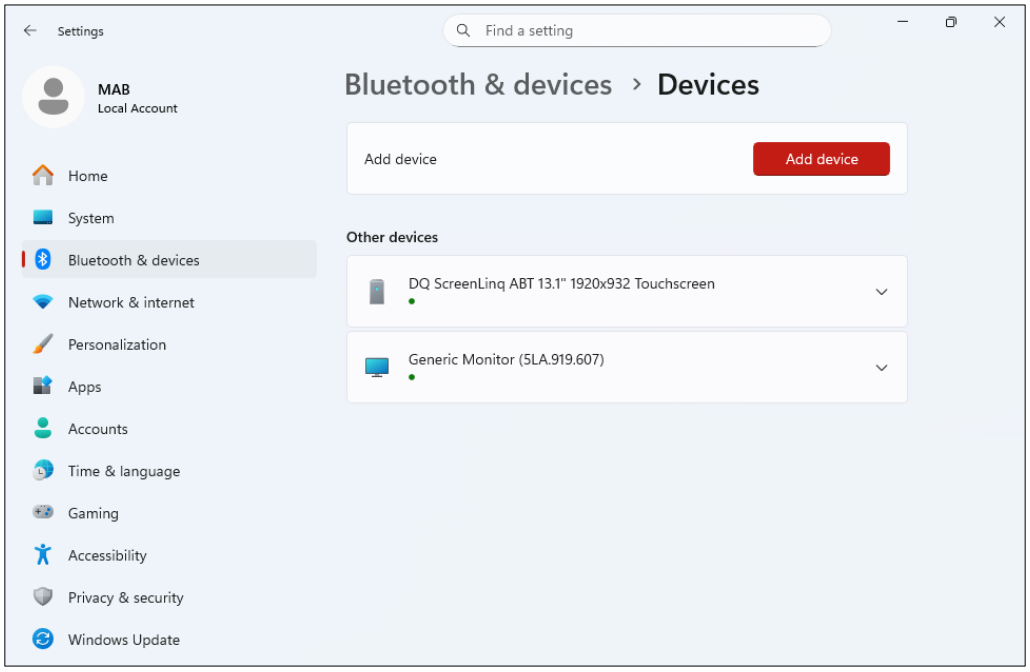


Figure 27: A fully recognized ScreenLinq device in Windows 11 "Bluetooth & other devices" dialogue.

WARNING

In case only the *"Generic Monitor"* has been detected, please check the *"HID"* switch is in **ON** position, USB cable is attached, and re-power ScreenLinq device to activate the USB HID feature (see Section 4.2).

Under normal circumstances, Windows 11 pairs a newly attached display with a HID touch screen automatically. In the unlikely case the pairing process has been unsuccessful, it is possible to pair such devices manually by following these steps:

1. Open legacy *"Control Panel"* dialogue (see Fig. 28),
2. Switch the dialogue's interface to *"Small icons"* (see Fig. 29),
3. Open *"Tablet PC Settings"* (see Fig. 30),
4. Click on the *"Setup"* (see Fig. 31),
5. Assign a screen to a touchscreen interface (see Fig. 32).

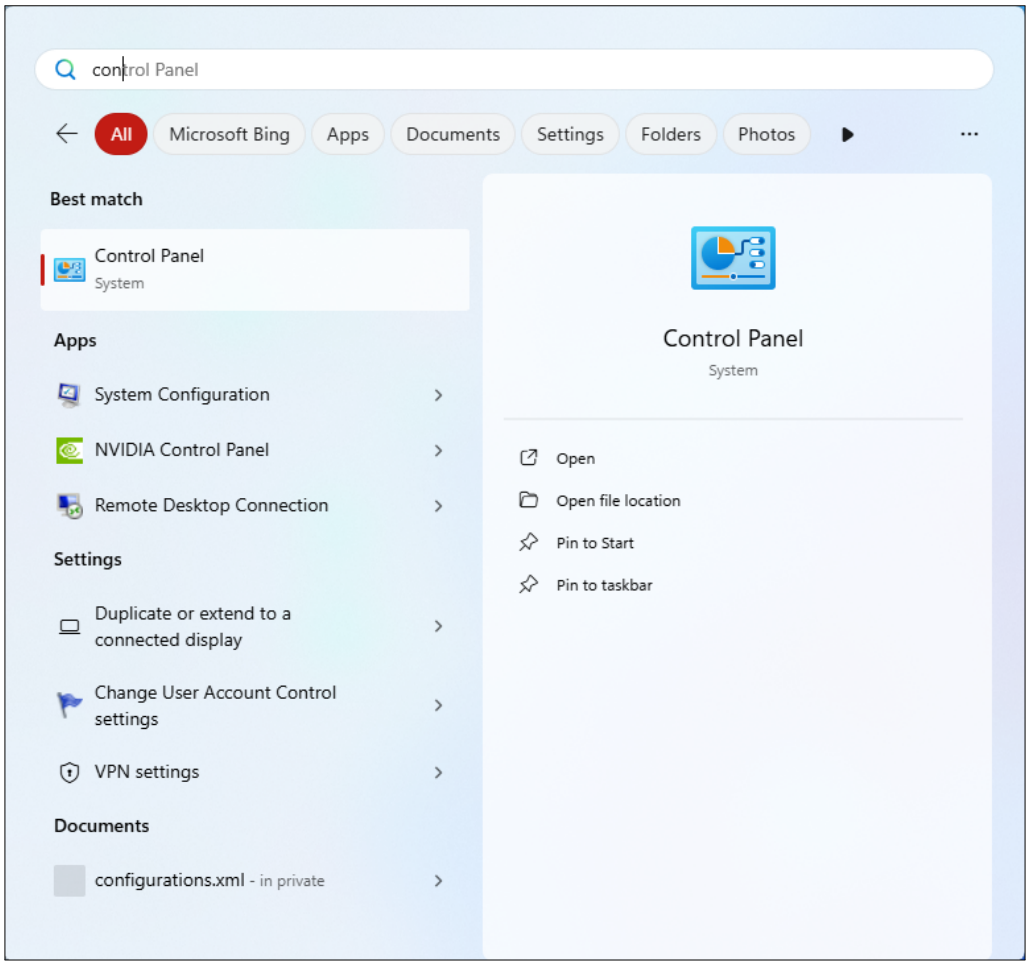


Figure 28: Invoking the *"Control Panel"* from the *"Start"*.

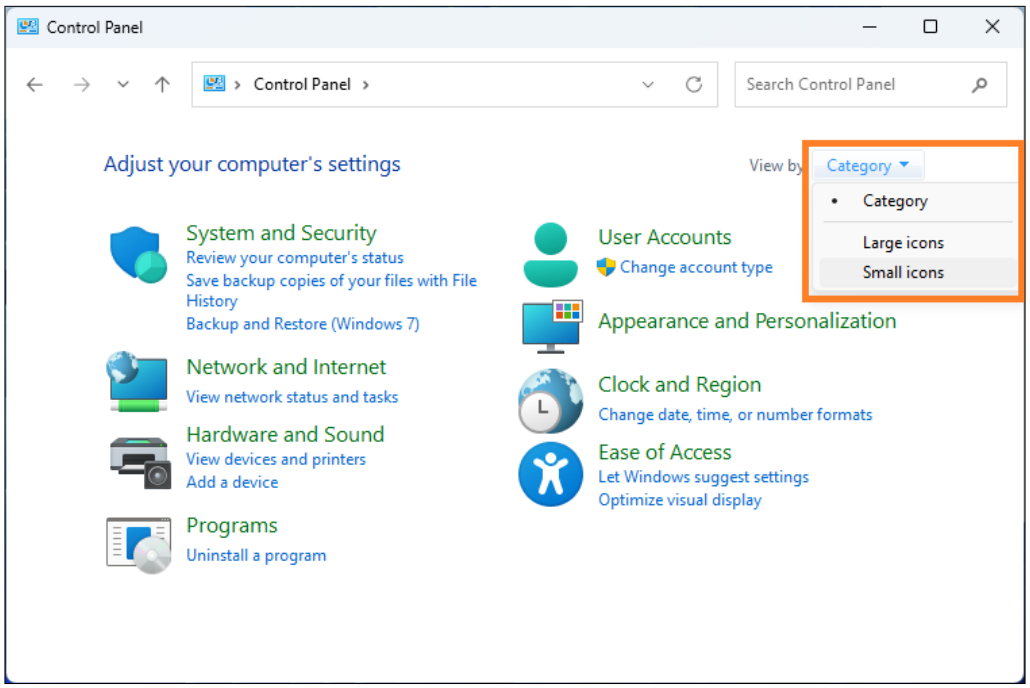


Figure 29: *"Control Panel"* dialogue

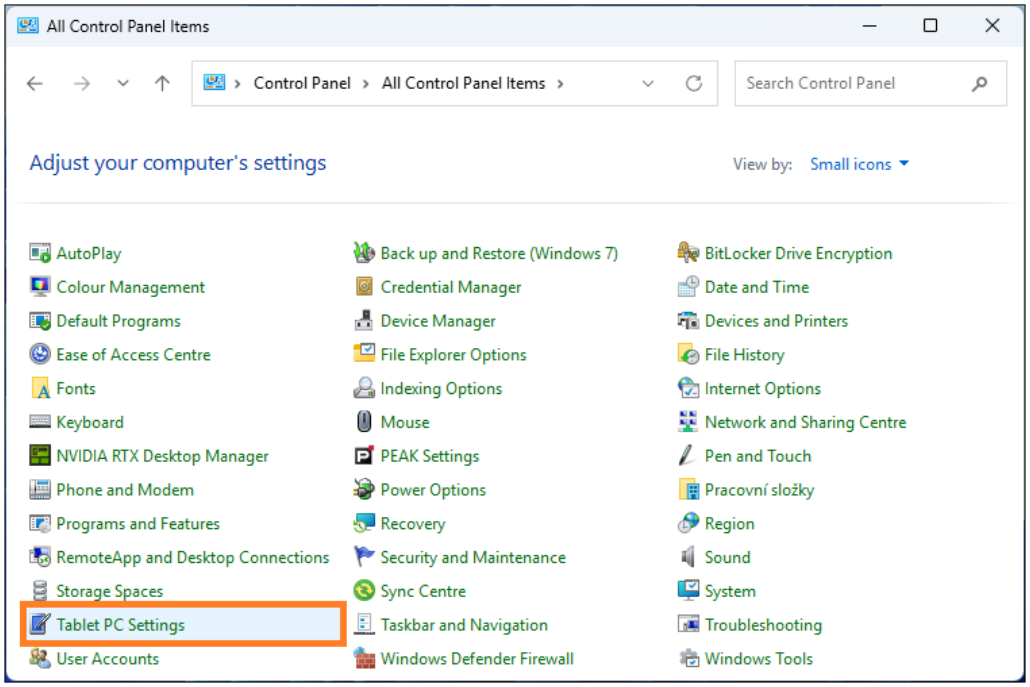


Figure 30: *"Control Panel"* dialogue – Tablet PC Settings.

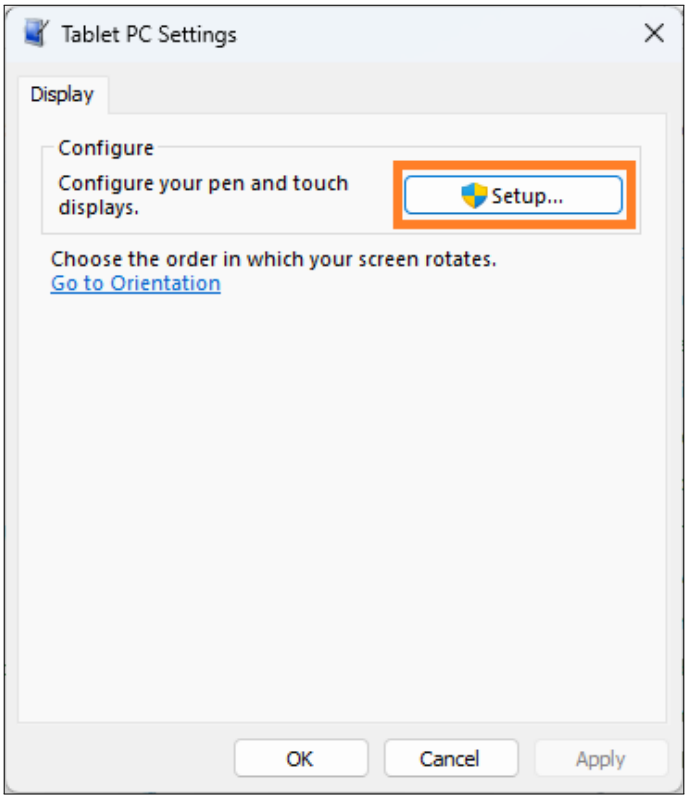


Figure 31: *"Tablet PC Settings"* dialogue – Setup.

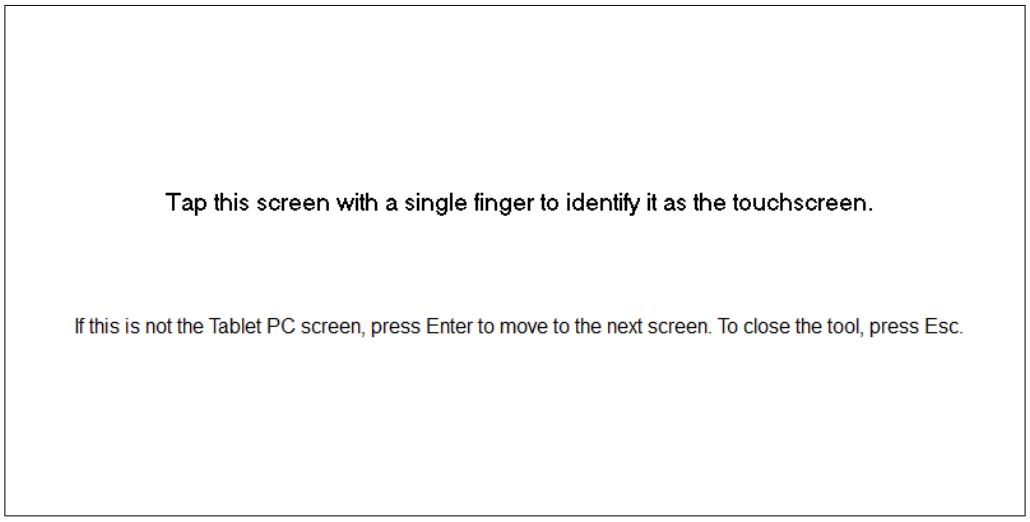


Figure 32: Choosing the appropriate screen to be paired with the touchscreen interface.

Appendix D: E-EDID 1.3 structure of the generic (incorrect) display profile

```
0x00 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x00 0x53 0x0E 0x49 0x09 0x01 0x00 0x00 0x00
0x1C 0x18 0x01 0x03 0x80 0x34 0x20 0x78 0x0A 0xEC 0x18 0xA3 0x54 0x46 0x98 0x25
0x0F 0x48 0x4C 0x00 0x00 0x00 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x1D 0x00 0x72 0x51 0xD0 0x1E 0x20 0x6E 0x50
0x55 0x00 0x00 0x20 0x21 0x00 0x00 0x18 0x00 0x00 0x00 0xFD 0x00 0x3B 0x3D 0x62
0x64 0x08 0x00 0x0A 0x20 0x20 0x20 0x20 0x20 0x20 0x00 0x00 0x00 0xFC 0x00 0x54
0x49 0x2D 0x44 0x53 0x39 0x30 0x55 0x78 0x39 0x34 0x39 0x0A 0x00 0x00 0x00 0x10
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x57
0x02 0x03 0x15 0x40 0x41 0x84 0x23 0x09 0x7F 0x05 0x83 0x01 0x00 0x00 0x66 0x03
0x0C 0x00 0x10 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x28
```

Listing 2: E-EDID 1.3 structure of the generic (incorrect) display profile.

Appendix E: Display Profile Template – E-EDID 1.4

```
0x00 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x00 0x53 0x0E 0x49 0x09 0x01 0x00 0x00 0x00
0xFF 0x22 0x01 0x04 0xA2 0x21 0x13 0x78 0x02 0xEC 0x18 0xA3 0x54 0x46 0x98 0x25
0x0F 0x48 0x4C 0x00 0x00 0x00 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0xB2 0x45 0xC0 0x60 0x80 0xEC 0x0D 0x40 0x1C 0x1C
0x62 0x00 0x4C 0xBB 0x10 0x00 0x00 0x18 0x00 0x00 0x00 0xFC 0x00 0x31 0x34 0x41
0x2E 0x39 0x31 0x39 0x2E 0x36 0x30 0x36 0x0A 0x20 0x00 0x00 0x00 0xFE 0x00 0x48
0x30 0x33 0x30 0x53 0x41 0x30 0x33 0x30 0x50 0x30 0x32 0x0A 0x00 0x00 0x00 0x10
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x5E
0x02 0x03 0x2B 0x00 0x67 0x03 0x0C 0x00 0x00 0x00 0x00 0x24 0xF1 0x01 0x00 0x00
0x00 0x44 0x51 0x01 0x10 0x01 0x01 0x00 0x01 0x00 0x03 0x02 0x00 0x00 0xEC 0x01
0x00 0x00 0x00 0x44 0x51 0x02 0x10 0x02 0x02 0x01 0x02 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFB
```

Listing 3: Recommended E-EDID 1.4 structure template in hexadecimal format.

User's Notes:

ScreenLinq's Product Page

https://products.digiteqautomotive.com/screenlinq/

Still looking for a solution?

In case you are unable to find an answer to your question, do not hesitate to contact us via an email: support.products@digiteqautomotive.com

Do you need an offer?

Do you need to order a new device or accessories, or are you looking for a custom-tailored solution to satisfy your requirements? Do not hesitate to contact us via an e-mail: business.products@digiteqautomotive.com