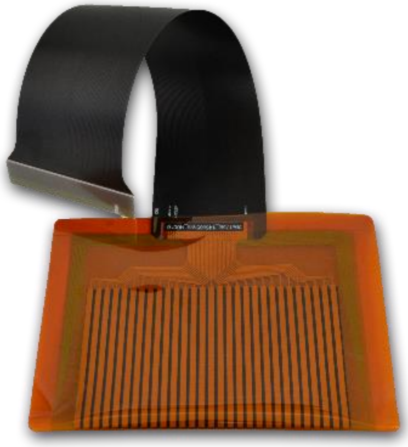
FSR-Matrix TM29x17 (Type 1) - PI

Product Brief

InnovationLab GmbH
Speyerer Straße 4
69115 Heidelberg
www.innovationlab.de
info@innovationlab.de

Key Features

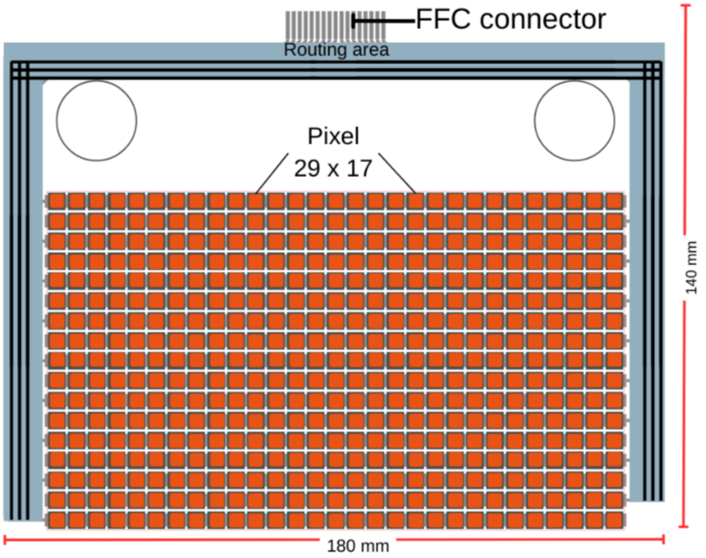
- Piezoresistive sensor foil
- Compact design
- High spatial resolution to detect small differences in pressure distributions
- Compatible with iL-readout electronics, Plug & Play
- Temperature stable PI-substrate
- Large pressure range
- Cost-efficient roll-to-roll manufacturing
- Available for:
  - DevKit electronics
  - Universal electronics



Mechanical data

Foil thickness & material	50 µm Polyimide	**Spatial resolution** (# of pixels)	29 x 17
Foil dimensions (L x W x H)	18 x 14 x 0.12 cm³	**Pixel size**	0.32 x 0.32 cm^2
Active area	8.5 x 14.5 cm^2	**Pixel pitch**	0.5 cm
Approved temperature range*	-20 °C to 100 °C	**Sensitive matrix part** (w/o concentrator)	41 %

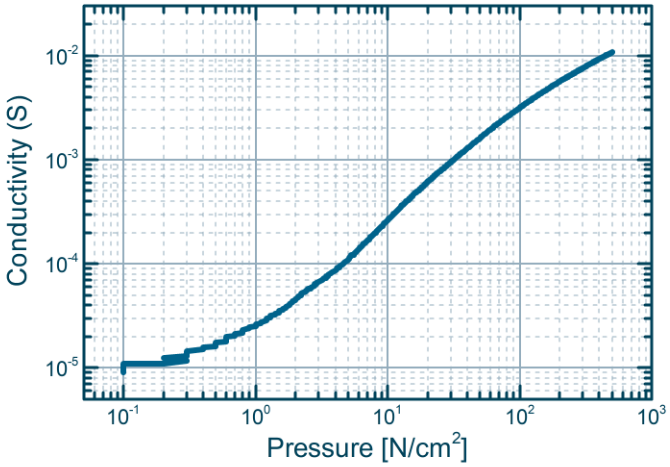
*in non-condensing solvents

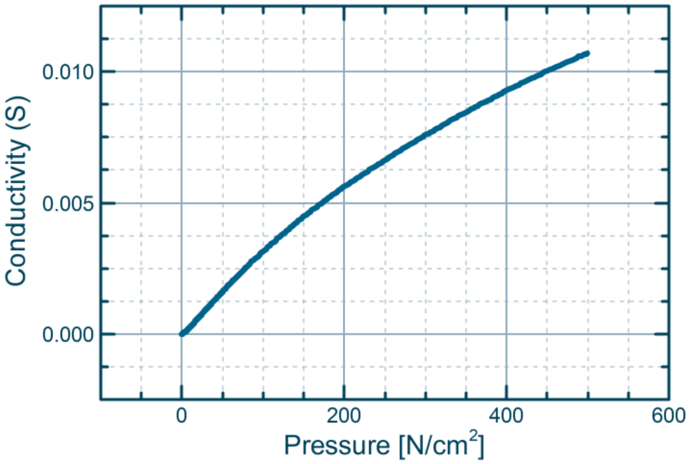


Sensor characteristics

The following data was obtained without preconditioning of the sensor. Preconditioning, i.e. loading the sensor with a high pressure for a certain time prior to the measurement, can improve the specs.

<table>
<tr><th></th><th>@ 1 Ncm^{-2} / 10 kPa</th><th colspan="2">@ 10 Ncm^{-2} / 100 kPa</th><th>@ 100 Ncm^{-2} / 1 000 kPa</th></tr>
<tr><td><b>Repeatability Error (1 σ)</b>
(20 repetitions)</td><td>16.5 %</td><td colspan="2">3.5 %</td><td>2.7 %</td></tr>
<tr><td><b>Drift (average)</b>
(% / $\log_{10}$ time scale)</td><td>10.2 %</td><td colspan="2">10.3 %</td><td>9.2 %</td></tr>
<tr><td><b>Hysteresis</b>
(at 50% of maximal actuation)</td><td colspan="4">20.5 %</td></tr>
<tr><td><b>Recommended max. actuation</b></td><td colspan="2">2 000 Ncm^{-2}</td><td colspan="2">20 000 kPa</td></tr>
<tr><td><b>Dynamic range</b></td><td colspan="2">0.1 – 2 000 Ncm^{-2}</td><td colspan="2">1 – 20 000 kPa</td></tr>
<tr><td><b>Temperature coefficient</b></td><td colspan="4">~ 0.9 %/K</td></tr>
<tr><td><b>Pixel-to-Pixel variation</b></td><td colspan="4">~ 5% (w/o compensation)</td></tr>
</table>





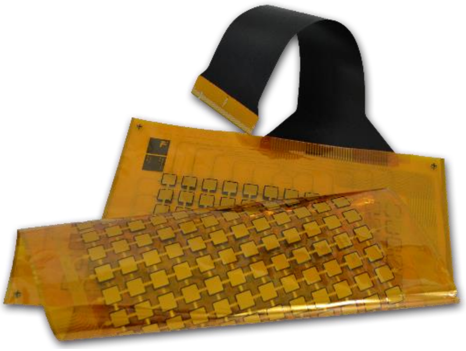
FSR-Matrix TM20x12 (Type 6) - PI

Product Brief

InnovationLab GmbH
Speyerer Straße 4
69115 Heidelberg
www.innovationlab.de
info@innovationlab.de

Key Features

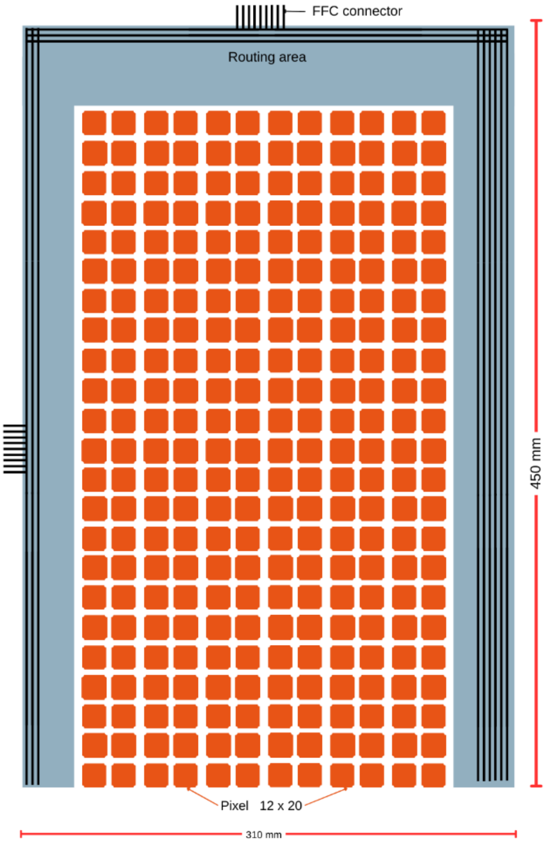
- Large area design
- Low drift and high sensitivity to measure detailed differences in pressure distributions
- Temperature stable PI-substrate
- Extra thin design
- Available for:
  - DevKit electronics



Mechanical data

Foil thickness & material	25 µm PI	**Spatial resolution** (# of pixels)	20 x 12
Foil dimensions (L x W x H)	45 x 31 x 0.007 cm^3	**Pixel size**	1.1 x 1.1 cm^2
Active area	37 x 22 cm^2	**Pixel pitch**	1.8 cm
Approved temperature range*	-20 °C to 100 °C	**Sensitive matrix part** (w/o concentrator)	36 %

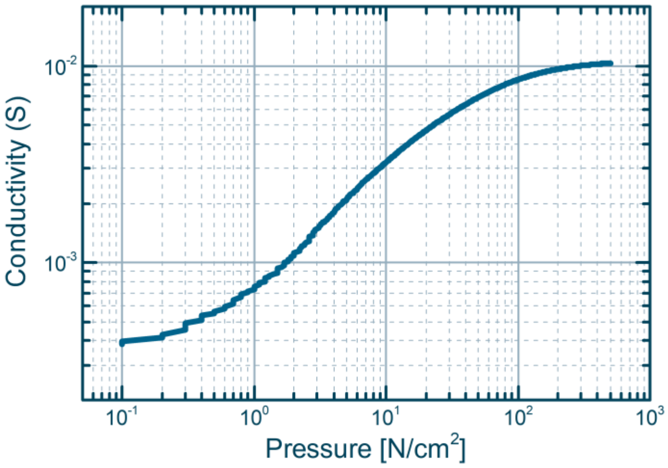
*in non-condensing solvents

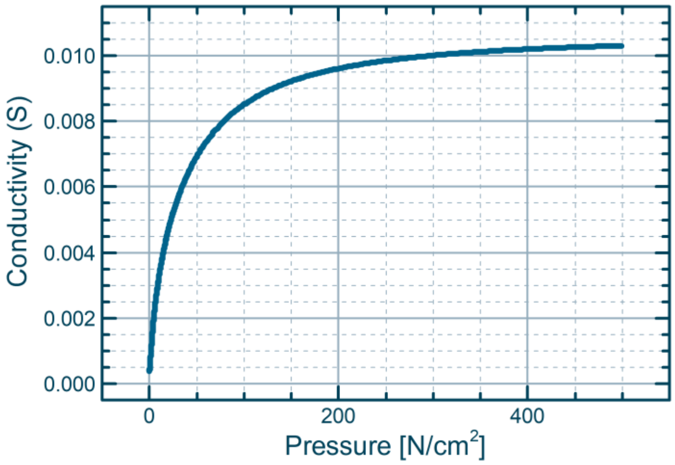


Sensor characteristics

The following data was obtained without preconditioning of the sensor. Preconditioning, i.e. loading the sensor with a high pressure for a certain time prior to the measurement, can improve the specs.

	@ 1 Ncm^{-2} / 10 kPa	**@ 10 Ncm^{-2} / 100 kPa**	**@ 100 Ncm^{-2} / 1 000 kPa**
Repeatability Error (1 σ) (20 repetitions w/o preconditioning)	8.9 %	2.3 %	0.7 %
Drift (average) (% / log$_{10}$ time scale)	10.4 %	5.2 %	1.7 %
Hysteresis (at 50 % of maximal actuation)	5.4 %		
Recommended max. actuation	2 000 Ncm^{-2}	20 000 kPa	
Dynamic range	0.1 – 500 Ncm^{-2}	1 – 5 000 kPa	
Temperature coefficient	~ 0.9 %/K		
Pixel-to-Pixel variation	~ 5% (w/o compensation)		





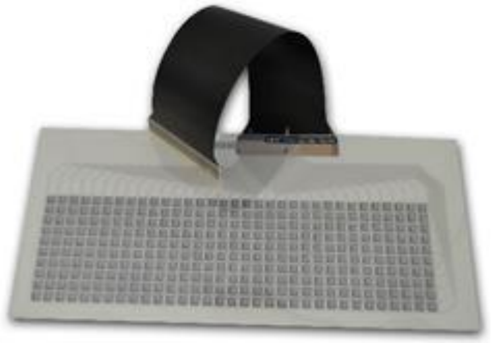
FSR-Matrix SM32x10 (Type 4) - PET

Product Brief

InnovationLab GmbH
Speyerer Straße 4
69115 Heidelberg
www.innovationlab.de
info@innovationlab.de

Key Features

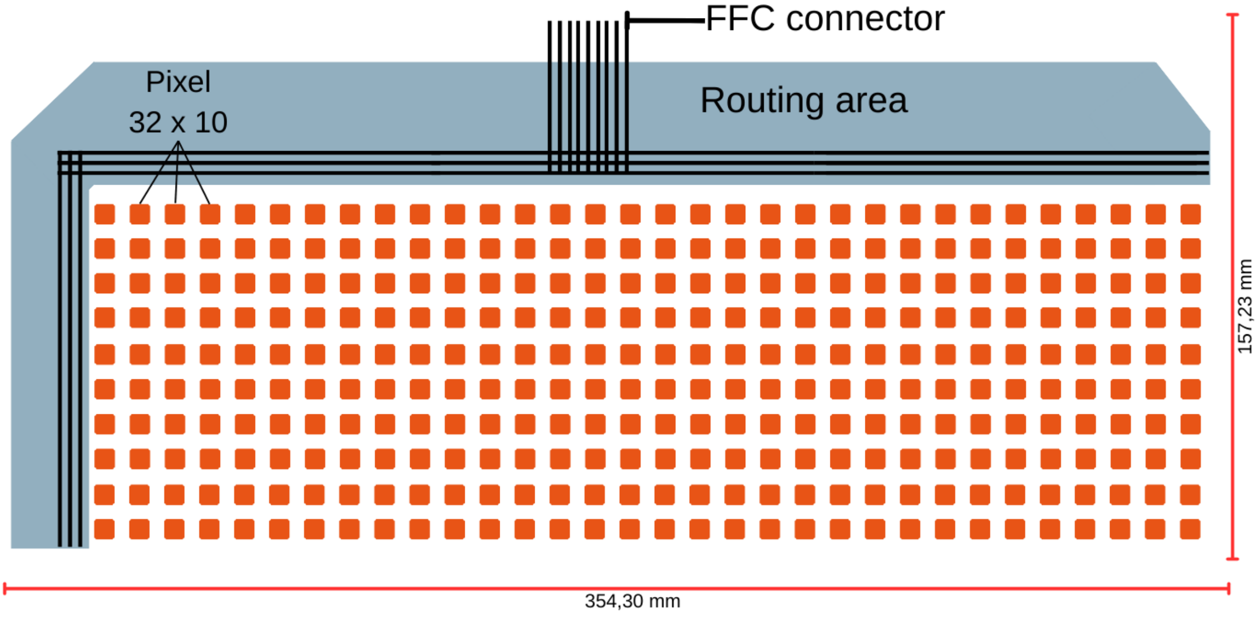
- Piezoresistive sensor foil
- Wide range
- Compatible with iL-readout electronics, Plug & Play
- Cost-efficient roll-to-roll manufacturing
- Available for:
  - DevKit electronics
  - Universal electronics



Mechanical data

Foil thickness & material	75 µm PET	**Spatial resolution** (# of pixels)	32 x 10
Foil dimensions (L x W x H)	36 x 17 x 0.021 cm^3	**Pixel size**	0.62 x 0.57 cm^2
Active area	32.5 x 9.6 cm^2	**Pixel pitch**	1 cm
Approved temperature range*	-20 °C to 100 °C	**Sensitive matrix part** (w/o concentrator)	36 %

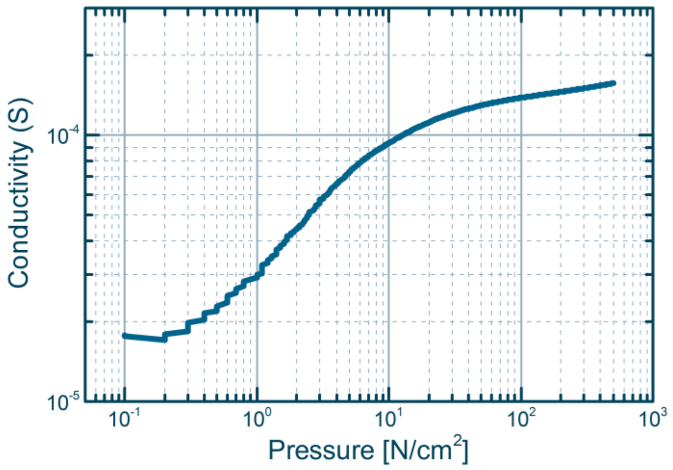
*in non-condensing solvents

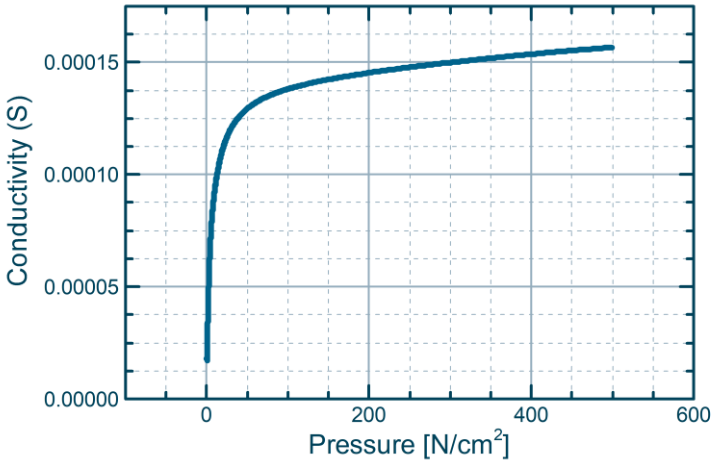


Sensor characteristics

The following data was obtained without preconditioning of the sensor. Preconditioning, i.e. loading the sensor with a high pressure for a certain time prior to the measurement, can improve the specs.

<table>
<tr><th></th><th colspan="2">@ 1 Ncm^{-2} / 10 kPa</th><th colspan="2">@ 10 Ncm^{-2} / 100 kPa</th><th colspan="2">@ 100 Ncm^{-2} / 1 000 kPa</th></tr>
<tr><td><b>Repeatability Error (1 σ)</b>
(20 repetitions w/o preconditioning)</td><td colspan="2">6.7 %</td><td colspan="2">1.3 %</td><td colspan="2">0.14 %</td></tr>
<tr><td><b>Drift (average)</b>
(% / log$_{10}$ time scale)</td><td colspan="2">11.2 %</td><td colspan="2">3.2 %</td><td colspan="2">0.5 %</td></tr>
<tr><td><b>Hysteresis</b>
(at 50 % of maximal actuation)</td><td colspan="6">1.7 %</td></tr>
<tr><td><b>Recommended max. actuation</b></td><td colspan="3">2 000 Ncm^{-2}</td><td colspan="3">20 000 kPa</td></tr>
<tr><td><b>Dynamic range</b></td><td colspan="3">0.1 – 2 000 Ncm^{-2}</td><td colspan="3">1 – 20 000 kPa</td></tr>
<tr><td><b>Temperature coefficient</b></td><td colspan="6">~ 0.9 %/K</td></tr>
<tr><td><b>Pixel-to-Pixel variation</b></td><td colspan="6">~ 5% (w/o compensation)</td></tr>
</table>





DevKit Electronics

Product Brief

InnovationLab GmbH
Speyerer Straße 4
69115 Heidelberg
www.innovationlab.de
info@innovationlab.de

Key Features

- State-of-the-art read-out electronics without crosstalk between pixels
- 12-bit ADC resolution
- Supports sensor matrices with up to 32x32 sensor pixels
- Scan frequency for maximum resolution (32x32): 8 Hz
- Up to 8 individual FSR sensors supported
- Simple data readout via USB Serial
- Implementation as an extension board for Arduino Nano BLE ©
- BLE connectivity, 3D accelerometer, gyroscope and magnetometer of the Arduino can be used
- Programmable via Arduino IDE, source code available
- Two client app options: feature-rich GUI software for Windows (SensorMatrixLAB) or open-source Unix console client
- Powering options: via USB port or battery powered

Usage

InnovationLab's DevKit Electronics is an entry-level crosstalk-less solution for capturing and visualizing data from printed sensor matrices. It allows to easily evaluate capabilities of resistive sensors like FSR (force-sensitive resistor) technology and quickly develop prototypes of various usage scenarios. It is based on a high-performing analog module, which is also used in InnovationLab's universal electronics. This results in state-of-the-art readout precision with simple Arduino-based programming interface.

What's in the package?

Included:

- Main board with 1 Analog Module (for scans of up to 32x32 pixel matrices)
- Arduino Nano BLE (preprogrammed)
- Open-source Unix console client (github)
- Open-source Arduino Code (github)

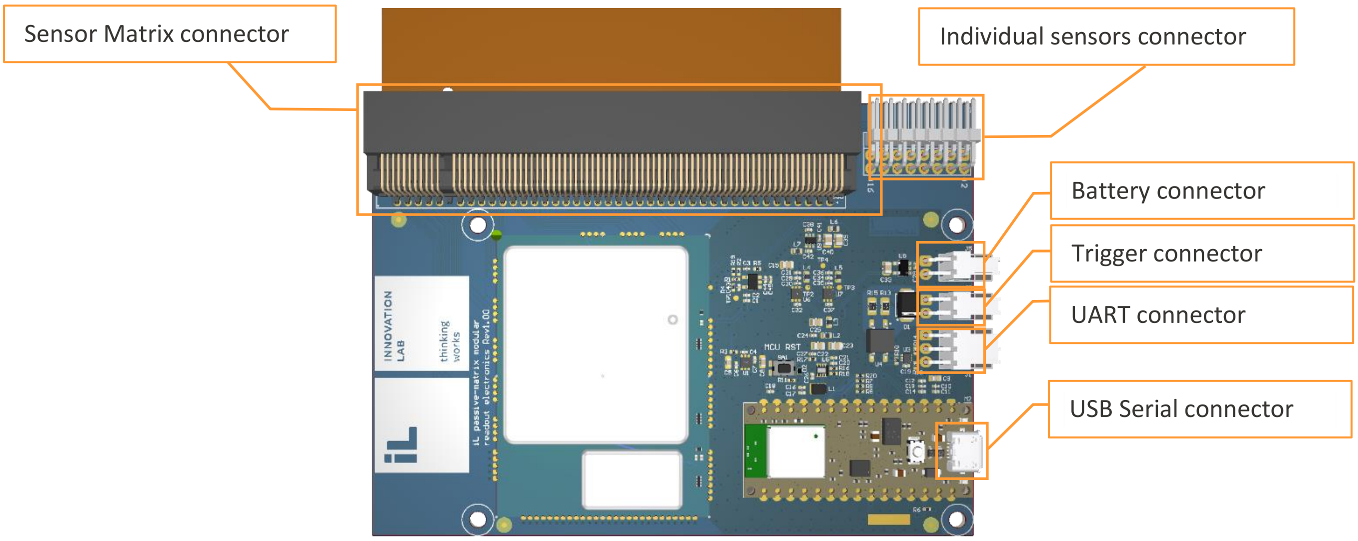
Not included:

- Micro USB cable
- Battery
- Sensors (check our DevKit)
- Dedicated Software: SensorMatrixLAB (licenses available upon request)

Dimensions

LxWxH: 115x82x14 mm³

On-board connectors



Sensor Matrix connector

The PCI express connector is used for attaching printed sensor matrices to the DevKit electronics. Pins A01-A32 are connected to matrix rows (drive), pins B01-B32 are connected to matrix columns (read).

Individual sensors connector

The right angle two rows pin header connector is used for attaching to the DevKit individual resistive sensors. Pin mapping is as follows:

15 – S8D	**13** – S7D	**11** – S6D	**9** – S5D	**7** – S4D	**5** – S3D	**3** – S2D	**1** - S1D
16 – S8R	**14** – S1R	**12** – S6R	**10** – S5R	**8** – S4R	**6** – S3R	**4** – S2R	**2** - S1R

Where **SxD** corresponds to **Sensor *x* Drive** and **SxR** corresponds to **Sensor *x* Read.**

USB Serial Connector

The connector of an Arduino Nano BLE is used for both programming the DevKit electronics via Arduino IDE and the data exchange with the software client. When configuring the software client, the data rate of 1 000 000 bps should be used.

Trigger Connector

Used for synchronizing the DevKit with an external equipment.

Pin	Label	Arduino connection	Description
1	TRIG-	GND	Trigger relay negative input (decoupled)
2	TRIG+	AIN5_A3/GPIO23	Trigger relay positive input (decoupled)

1 mA required for activating the relay, 60 V max.

UART Connector

Pin	Label	Arduino connection	Description
1	GND	GND	Signal and System ground
2	TX	Arduino TX, pin 17	Transmit Data
3	RX	Arduino RX, pin 16	Receive Data

For DevKit electronics the voltage level of 3.3V is used. Use Arduino ***Serial*** object to access the port.

Battery Connector

Voltage range: 2,5 – 5.5 V (only for supply, no charging possible)

SensorMatrixLAB

Software for printed Sensor Matrices

User Guide

Full Edition

Manual version:	SML 5 – Rev 1.0.1
Software version:	5.0.23.x
Supported hardware version:	Rev 1.01
Supported device firmware:	7.3.x (Universal Matrix)
Supported device firmware:	0.5.x (Devkit)

INNOVATION LAB

thinking works

Impressum

InnovationLab GmbH

Speyerer Str. 4

69115 Heidelberg, Germany

www.innovationlab.de

Customer Support

Telephone number: +49 (0) 6221-54 19 100

Fax number: +49 (0) 6221-54 19 110

Email: info@innovationlab.de

Copyright ©

Table of Contents

1 Introduction

This document is an integral part of the SensorMatrixLAB Software.
To ensure proper operation and full functionality of this product, read this manual carefully and completely before taking any action related to the use of this program.

1.1 How to Use This Manual

This manual provides the necessary information about the purpose, system requirements and settings needed to install, run, and effectively use SensorMatrixLAB software.

1.2 Typographic Conventions Used in This Document

In this user guide, certain words and phrases are distinguished by special formatting:

- Keywords and important information are highlighted in bold (e.g., **NOTE:** this is some important information).
- Software interface texts referenced in this manual are highlighted in italics (e.g., Use the *Apply* button to save the size).

2 Safety Instructions and Recommendations

These instructions facilitate the safe and efficient use of the product and should be available to the operator at user site at any time. Failure to follow these instructions may result in errors in the functioning of the program and in the incorrect processing and storage of data.

2.1 User Qualifications and Intended Use

This software is intended for the specific purpose as described. It may only be used by users with at least a basic IT knowledge, and only with the equipment and under the operating conditions described in this documentation.

The user is fully responsible for the consequences resulting from the use of the product beyond its intended use and the safety rules described in this manual.

2.2 General Software Safety Instructions

To ensure effective and safe use of SensorMatrixLAB software:

- Always install the latest operating system updates.
- Clear the cache in your Internet browser(s) regularly.
- Keep your anti-virus software up to date.

3 Overview and Requirements

The SensorMatrixLAB software is a Microsoft Windows application for visualizing and processing data from InnovationLab's pressure sensor matrices.

The software supports the following key features:

- Advanced data visualization
- High-speed data acquisition and playback
- API access to sensor data
- Communication interfaces: USB Serial, Wi-Fi, Ethernet
- Support for custom printed sensor matrices and different hardware types
- Advanced hardware configuration (including standalone operation via CAN/RS232)
- Sensor calibration (including individual calibration of sensors in a matrix)
- Full support of proprietary protocols

3.1 Hardware Requirements

Unless otherwise specified, these minimum hardware requirements apply to all installation options:

- InnovationLab’s hardware (electronics) with the latest firmware version
- InnovationLab’s printed sensor matrix.
- Windows-based personal computer with x86/x64 processor
- USB port with USB-to-serial driver installed.
- Optional: Internet connection
- Optional: Local network connection via Ethernet (for ETH connectivity)

3.2 Software Requirements

The software requires the following conditions to function properly:

1. Microsoft Windows 10 or higher
2. Microsoft .NET Framework 4.7.2
3. The latest version of a web browser (Mozilla Firefox, Google Chrome, or Microsoft Edge)
4. Microsoft Excel (2016 or later)

4 Installation and Maintenance

4.1 Software Installation

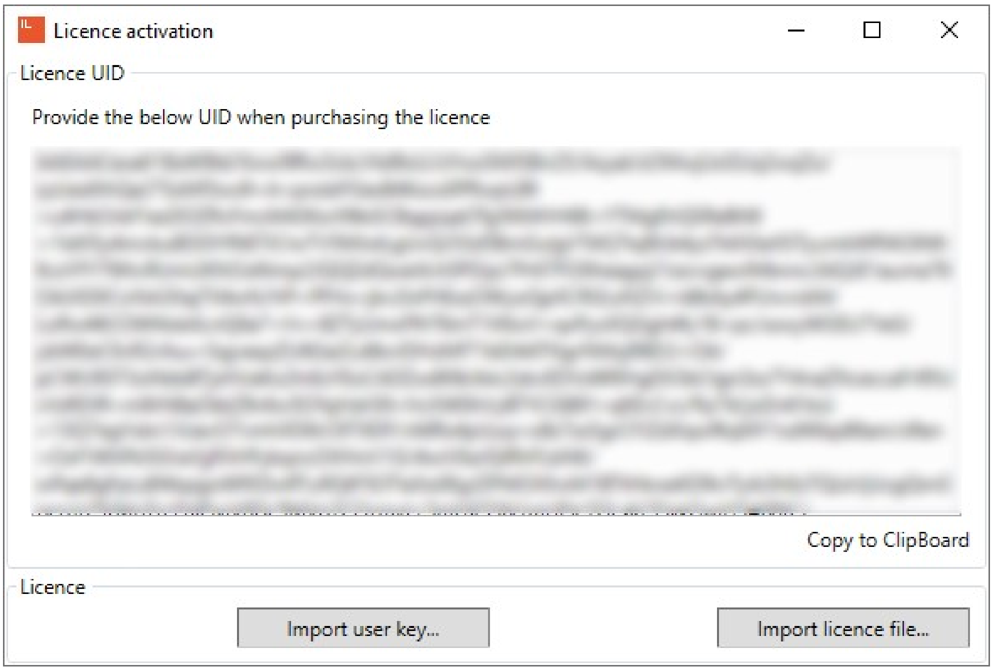
To install the SensorMatrixLAB software on your PC, run the provided installer executable and follow the installation instructions.

NOTE: SensorMatrixLAB does not require a local user to have administrator privileges on the computer to install or use the software. However, administrator privileges may be required to use the auto-discovery feature for ethernet connection.

4.2 Software Activation

Once installed, the software has to be activated.

1. Run the software
2. Read and confirm the license agreement
3. Copy your license UID to the clipboard (**Figure 1:** Activation window).



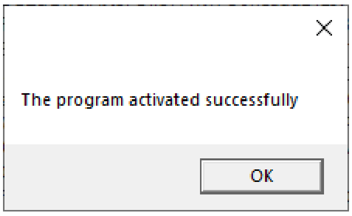
Licence activation

Licence UID

Provide the below UID when purchasing the licence

Figure 1: *Activation window*

4. Send the UID including your company's name and order number to software@innovationlab.de.
5. You will receive two files for activation:
    - A user key file (KEY)
    - A license file (LIC)
6. To activate the program import both files (**Figure 1:** Activation window)
7. If all operations have been performed correctly, you will see the following confirmation.



The program activated successfully

Figure 2: *Activation confirmation*

- ✓ Congratulations, your license activation is complete! If you start the software, the main window should appear.

4.3 Update

To update SensorMatrixLAB to a newer version:

1. Reinstall the current version from your PC.
2. Request activation files for a new software version.
3. Proceed with the activation as described above.

5 User Interface

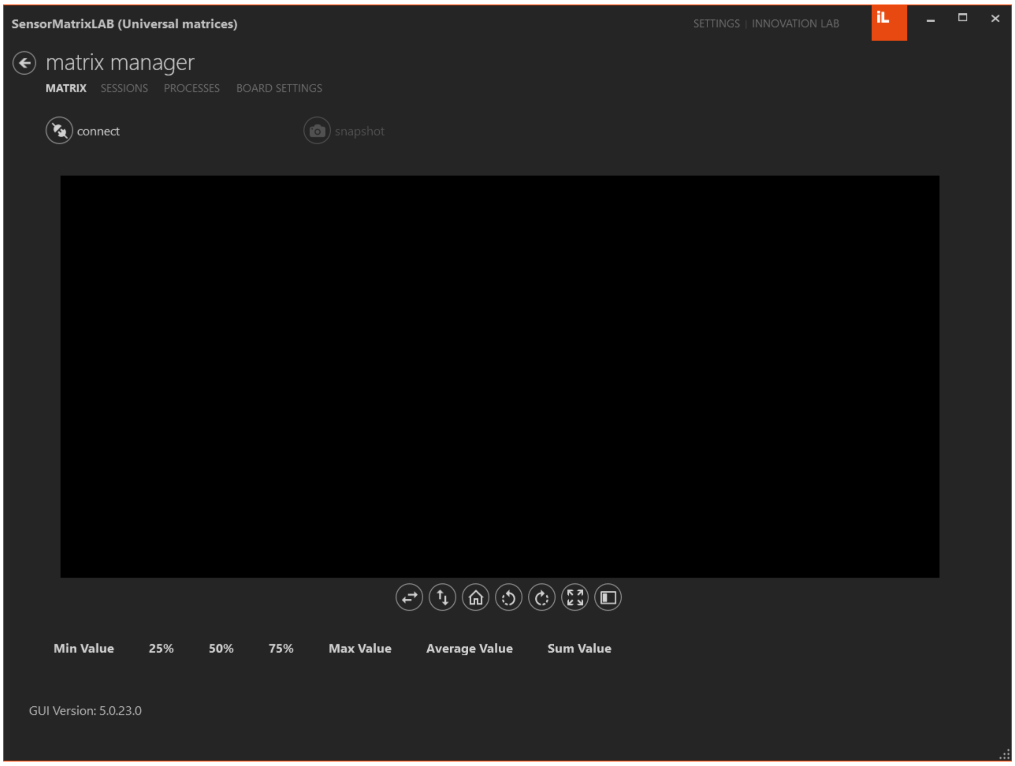


Figure 3: *SensorMatrixLAB start screen (Universal Matrix).*

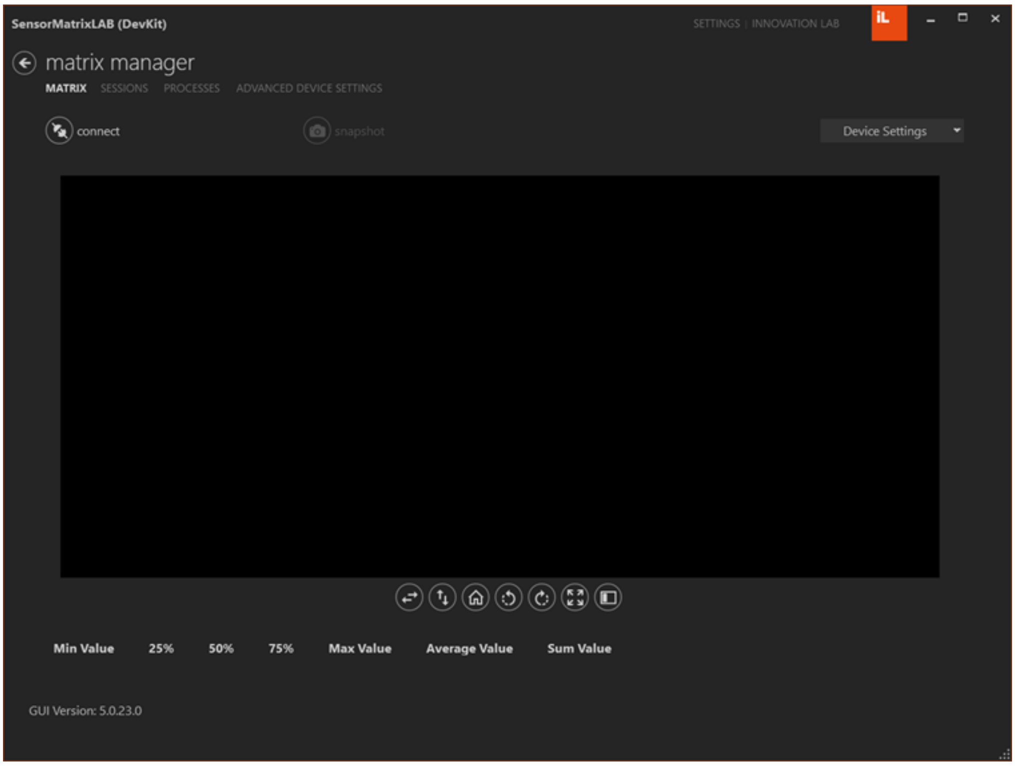


Figure 4: *SensorMatrixLab start screen (Devkit).*

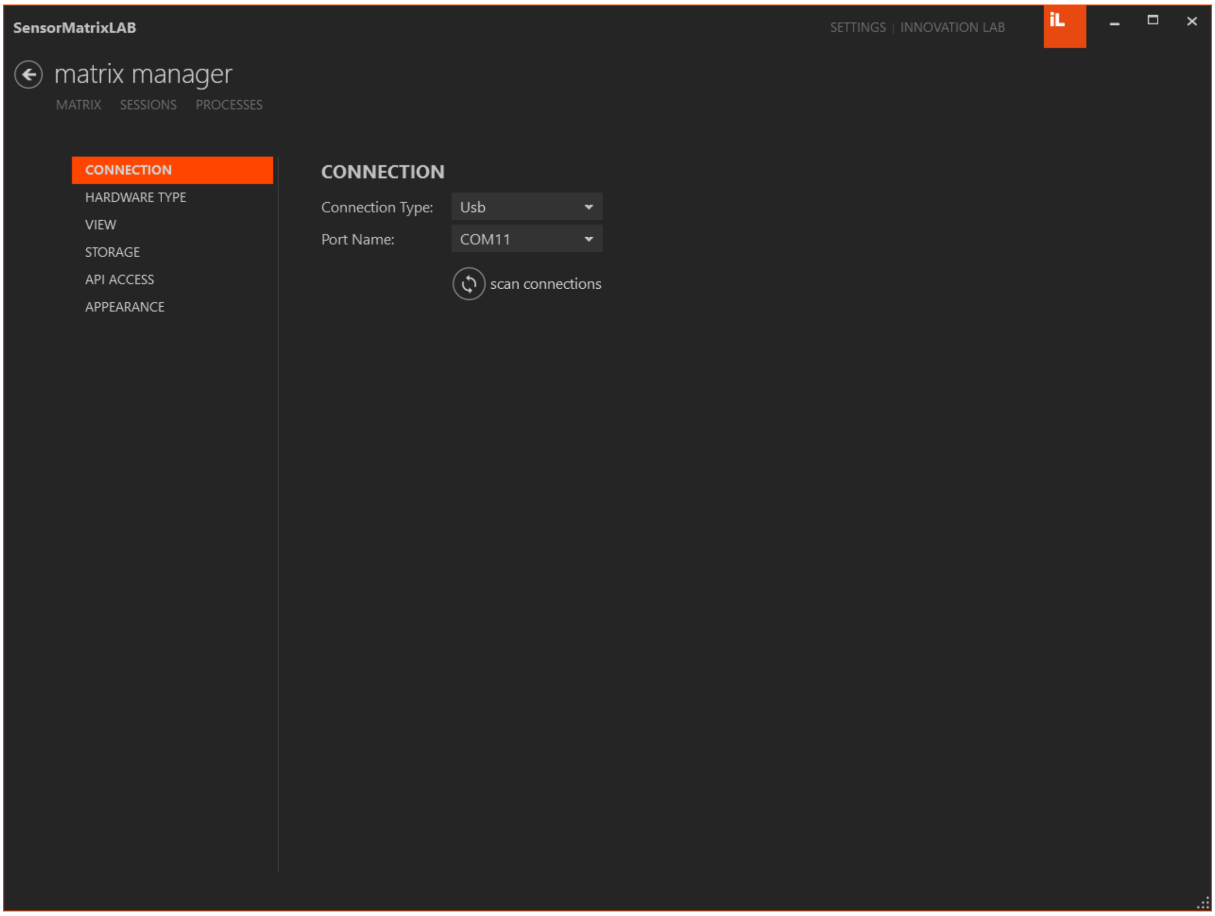
Figure 3 & **Figure 4** shows the main window of the SensorMatrixLAB software after startup. The different sections of the main window are described from top to bottom:

1. *SETTINGS* (left to the iL logo) refer to the global settings of the application. For detailed information on the settings and each menu, see the global settings section (**section 6**).
2. Tabs (below *matrix manager*) allows switching between several core features of the application. For more information, see the **section 7**:
    - *MATRIX*: main view for matrix visualization
    - *SESSIONS:* export and visualization of previously captured measurement sessions
    - *PROCESSES:* information about ongoing processes
    - *BOARD SETTINGS (Universal Matrix Only):* read, modify, and upload board parameters and settings.
    - *ADVANCED DEVICE SETTINGS (DevKit Only):* read, modify, and upload advanced board parameters.

6 Global Settings

Global settings related to the application and hardware connection can be changed under *SETTINGS* in the upper right corner of the application window.

NOTE: The *SETTINGS* button will disappear if a measurement session is running. Make sure that the device is disconnected to change the global settings.



CONNECTION

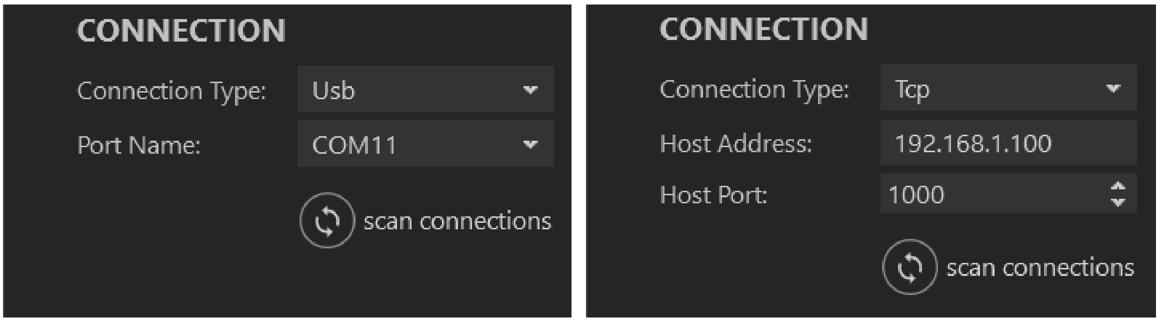
Connection Type: Usb

Port Name: COM11

***Figure 5:** SensorMatrixLAB – Settings (Universal Matrix)*

Figure 5 gives an overview of the *SETTINGS* window. The subsections are described in the following. For Devkit, the tab Frame processing is added.

6.1 Connection



CONNECTION

Connection Type: Usb

Port Name: COM11

CONNECTION

Connection Type: Tcp

Host Address: 192.168.1.100

Host Port: 1000

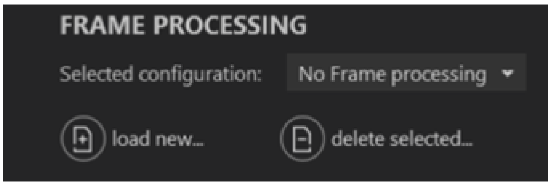
Figure 6: *CONNECTION settings for USB and TCP*

This subsection allows you to select and specify the connection interface, either USB or TCP. For USB, the corresponding COM port must be chosen. For TCP it is required to specify the IP address and port of the device. Use the *scan connections* button to search for possible connections.

NOTE: Not all hardware support TCP. Please refer to your hardware manual for more information.

NOTE: The *scan connections* button for TCP might require running the application with admin rights.

6.2 Frame Processing (Devkit only)



FRAME PROCESSING

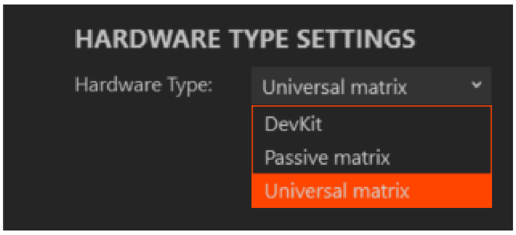
Selected configuration: No Frame processing

Figure 7: *Frame processing settings*

Per default the complete device’s readout resolution is visualized in the application. Usually, the connected sensor covers a resolution smaller than this. In addition, depending on the sensor layout, the arrangement of the readout does not correspond to the sensor matrix layout.

To reduce the amount of data and correct the arrangement, a sensor matrix specific JSON file can be loaded to the application. This frame conversion file can be uploaded using the *load new* button. After that it is selectable in the dropdown menu.

6.3 Hardware Type



HARDWARE TYPE SETTINGS

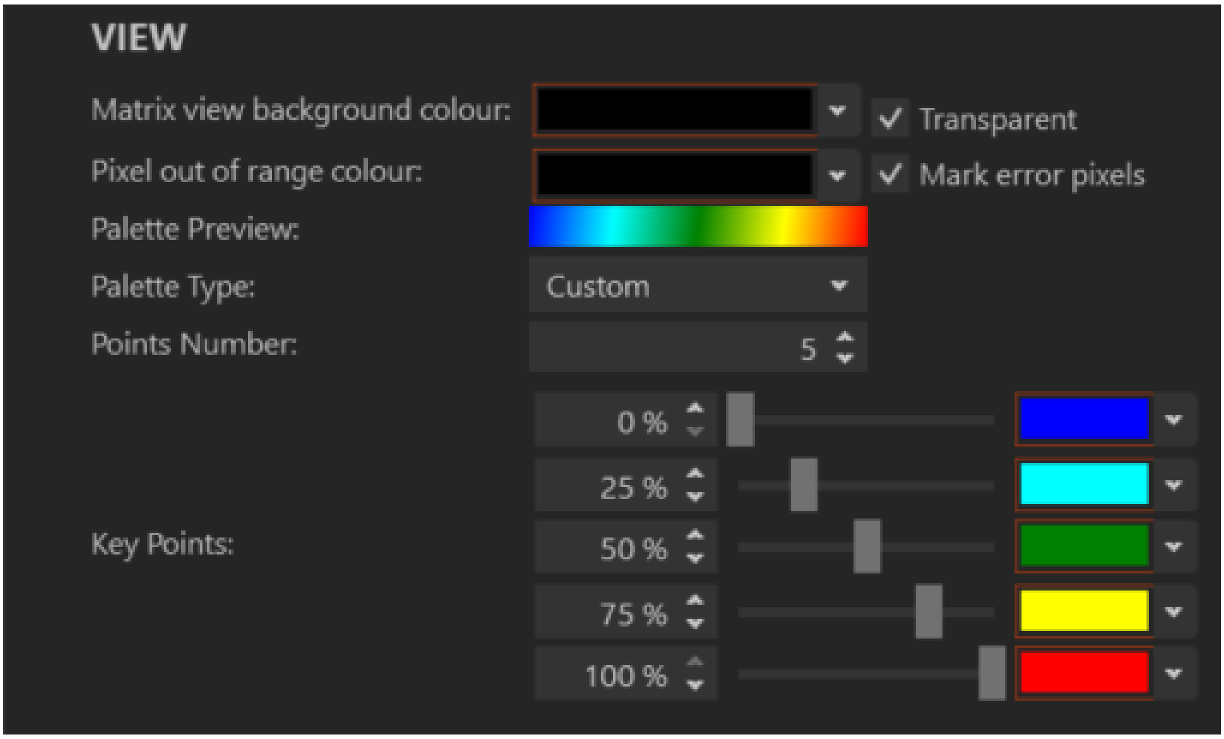
Hardware Type: Universal matrix

Figure 8: *HARDWARE TYPE Settings*

Depending on your order, several hardware types might be able to select. Choose the connected device from the dropdown menu. Typically, only one hardware type is selectable.

NOTE: If you are interested in the different hardware types and their capabilities, please visit www.innovationlab.de/en.

6.4 View Color Palette



VIEW

Matrix view background colour: ☑ Transparent

Pixel out of range colour: ☑ Mark error pixels

Palette Preview:

Palette Type: Custom

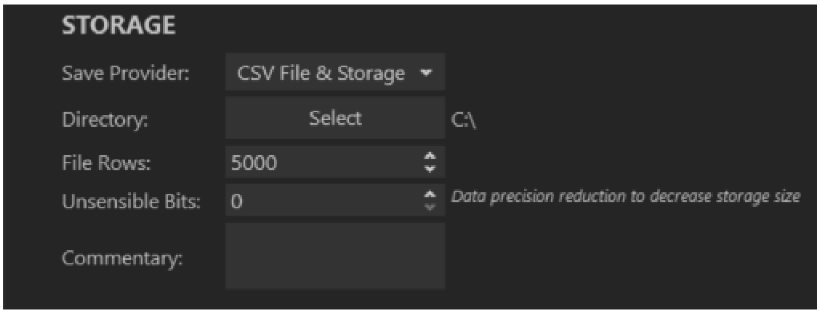
Points Number: 5

Key Points: 0 % 25 % 50 % 75 % 100 %

Figure 9: *VIEW Settings*

The *VIEW* section allows to select and configure a color map for data display. A variety of predefined color palettes is provided. In addition, a customized color palette can be configured (**Figure 9**).

6.5 File Storage



STORAGE

Save Provider: CSV File & Storage

Directory: Select C:\

File Rows: 5000

Unsensible Bits: 0 *Data precision reduction to decrease storage size*

Commentary:

Figure 10: *STORAGE settings*

The *STORAGE* subsection allows to save a session in two ways: *CSV file* (CSV) or *Storage* (DB). These modes can be selected under the *Save Provider* option. During a measurement, the data is written in the selected format to the directory specified.

In addition to the data, a session.txt or session.db file is generated, encapsulating crucial session information, including the Commentary, if specified.

NOTE: Make sure that a valid directory is specified.

6.5.1 CSV file type

The data in CSV files is arranged in the following format:

Column	1	2	3	4 and following
Description	Time Stamp (in ms)	Sample Number	CAN Time Stamp (if configured)	Data of sensor pixel

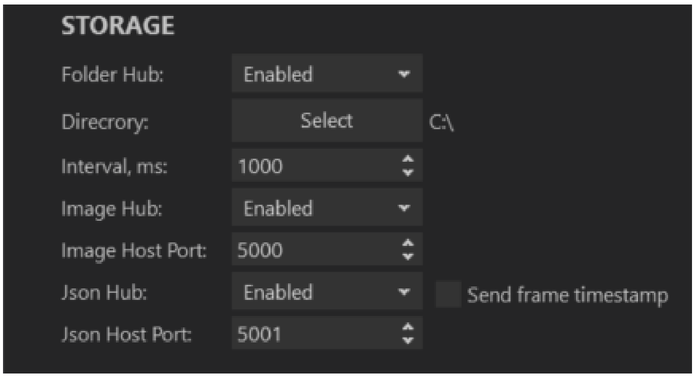
Each row corresponds to a single matrix snapshot. The two-dimensional matrix data is flattened and starts at the fourth column. The *File Rows* setting allows to change the maximum amount of CSV file rows before a new file is created.

6.5.2 DB file type

The database (DB) files provide a smaller sized storage option. Those files can be read, visualized, and exported to CSV in the *SESSIONS* tab (section 7.2).

To reduce the file size even more, the *Unsensible Bits* option can be used. It specifies a tolerance window a sensor's readout must leave before a new value is stored. E.g., setting this option to 3 requires a value to change by at least 8 (2^3) before a new value is saved. This is particularly useful for long-term measurements.

6.6 API Access



STORAGE

Folder Hub: Enabled

Direcrory: Select C:\

Interval, ms: 1000

Image Hub: Enabled

Image Host Port: 5000

Json Hub: Enabled □ Send frame timestamp

Json Host Port: 5001

Figure 11: *API Access settings*

The API ACCESS subsection provides several options to access the measurement data stream outside SensorMatrixLAB. Three APIs are provided:

- *Folder Hub*
- *Image Hub*
- *Json Hub*

6.6.1 Folder Hub

This mode creates grayscale PNG images while the session is running. To enable it, use the *Folder Hub* setting.

Specifications:

- The data is written to a selected *Directory* at a selected *Interval* (in ms).
- Each pixel of the generated graphics corresponds to a pixel of the sensor matrix.
- The luminosity contains information about the sensor value (0 to 4095).

6.6.2 Image Hub

To enable this mode, use the *Image Hub* setting.

Specifications:

- The matrix state is generated as a PNG image on demand for incoming HTTP requests.
- Images can be loaded using the HTTP GET request via the selected *Image Host Port*.
- The data format is the same as for Folder Hub.
- Try to access in a web browser via http://localhost:<SELECTED_PORT>/
  (e.g., http://localhost:5000/)

6.6.3 Json Hub

To enable this mode, use the *Json Hub* setting.

Specifications:

- The matrix state is generated as a JSON file on-demand for incoming HTTP requests.

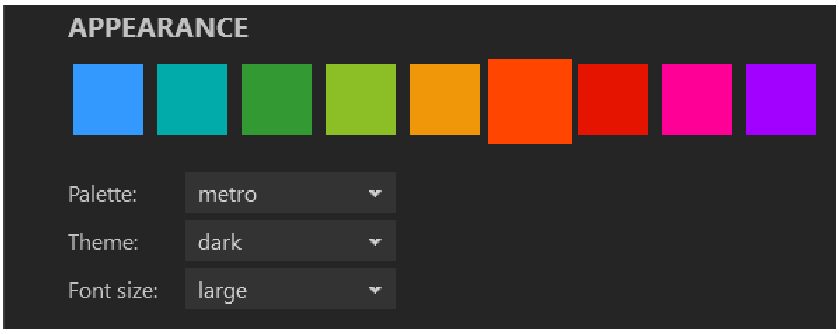
- The data can be loaded by third-party software using the HTTP GET request through the selected *JSON Host Port*.
- The matrix data is given in two-dimensional form as a listed list.
- Use the *send timestamp* checkbox to include the measurement start time and timestamp since measurement start in ms.
- Try to access in a web browser via http://localhost:<SELECTED_PORT>/ (e.g., http://localhost:5001/)

```
{"StartTime":"30-08-2023
20:51:26","CurrentFrameTimestamp":10157.0,"FrameData":[[0,0,0,1,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0],[0,0,0,1,0,0,1,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1
,0,0,0,0],[0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0]
,[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[1,0,0,0
,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
0,1,0,0,0,0],[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0],[0,0,0,0,0,0,0
,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,
0,0,0],[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[1
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0],[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
,[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]}
```

Figure 12: Example of a JSON file containing 32x32 matrix data including time stamp.

6.7 Application Style

All settings related to the application style can be found in the *APPEARANCE* section.



APPEARANCE

Palette: metro

Theme: dark

Font size: large

Figure 13: APPEARANCE *Settings*

7 Tabs

7.1 Matrix

This window controls the measurement to the connected device and provides various options to modify the live measurement presentation. In the following, its features are described from top to bottom (**Figure 14**).

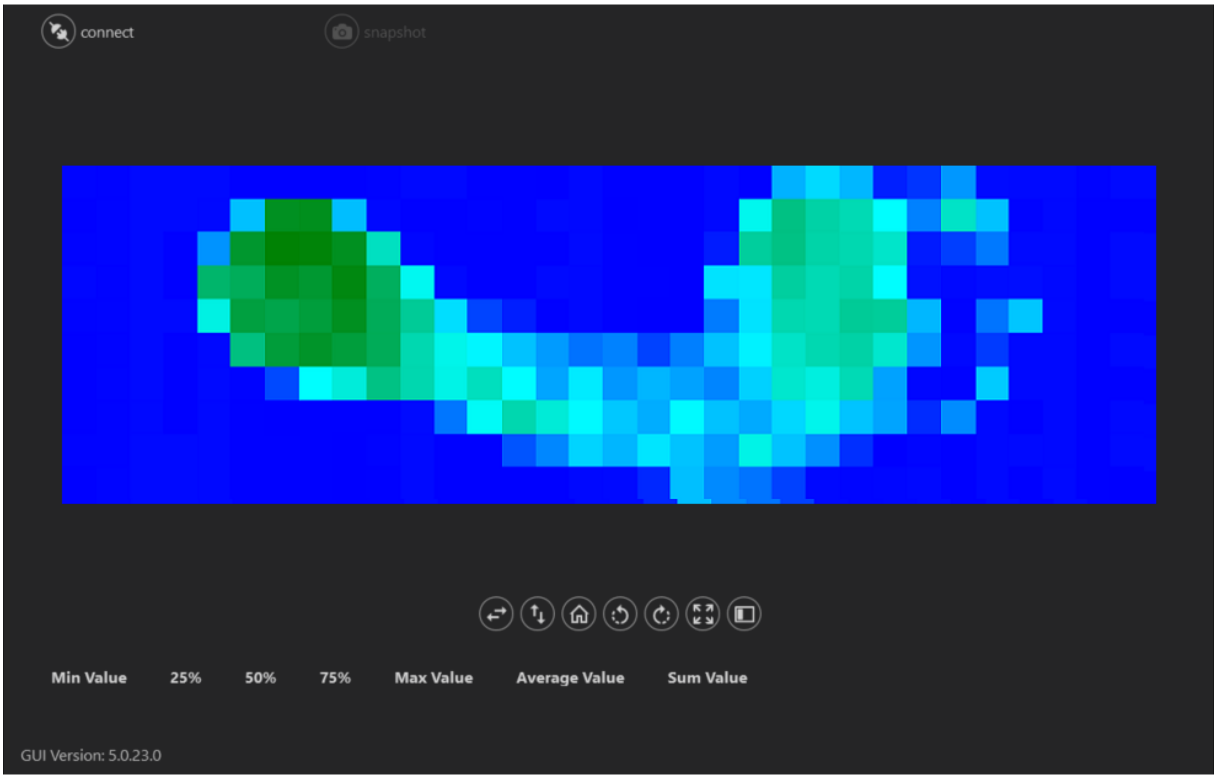


Figure 14: *MATRIX tab*

7.1.1 Control Buttons

The *connect/disconnect* button establishes a connection the device using the interface specified in the connection settings (section 6.1). After a connection is established, the *snapshot* button can be used to capture a screenshot of the current measurement visualization. To stop the current measurement session the *connect/disconnect* must be clicked again.

7.1.2 Device Settings (Devkit only)

The device settings are stored in the non-volatile memory of the hardware and the same settings are used to control the device via all supported interfaces (e.g., CAN, ETH).

NOTE: To update the device configuration, an existing connection to the hardware is required and the session must be stopped.

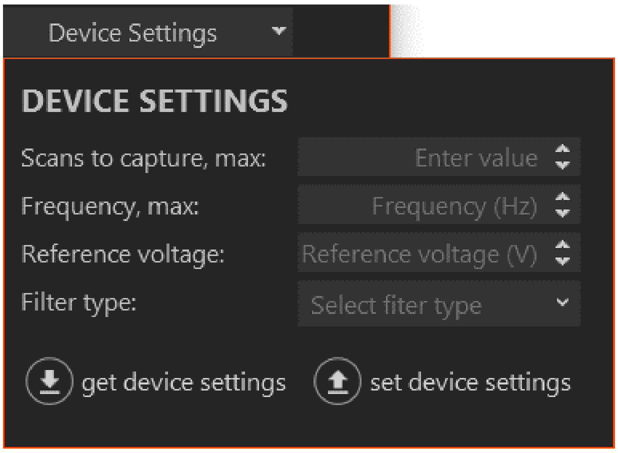


Figure 15: *Device settings window*

If the USB connection to the board is established successfully, the board settings can be changed in the *MATRIX* window under *Device Settings*.

The *Device Settings window* shown in **Error! Reference source not found.** contains several options:

- The *get device settings* button loads the current settings from the board, *set device settings* sends the user's input to the board.
- *Scans to capture, max* allows you to limit the number of captured images per session. When this number is reached, the session will be stopped. If this value is set to 0 (default), the session is not limited and can only be stopped using the *disconnect* button.
- *Frequency max* describes the upper limit for the measurement frequency of a session in Hz (samples per second). If this value is set to 0, the measurement will be performed at the maximum frequency possible for the current hardware and sensor matrix.
  NOTE: Some hardware does not support frequency limit. Refer to the hardware datasheet for more information.
- *Reference voltage* allows you to set a reference voltage value on the board. In general, the higher the reference, the higher the sensitivity of the sensor system to external stimulus. It is recommended to start measurements with a low reference voltage and to increase it if necessary. The unit of this value is Volts.
- *Filter type* allows you to apply time-domain filtering to live sensor data. This means that each matrix pixel is filtered as a function of time, independent of other pixels. The filter parameter can be set in the *ADVANCED DEVICE SETTINGS* tab (section 7.4).
  NOTE: Not all hardware supports live filtering. Please refer to your hardware manual for more information.

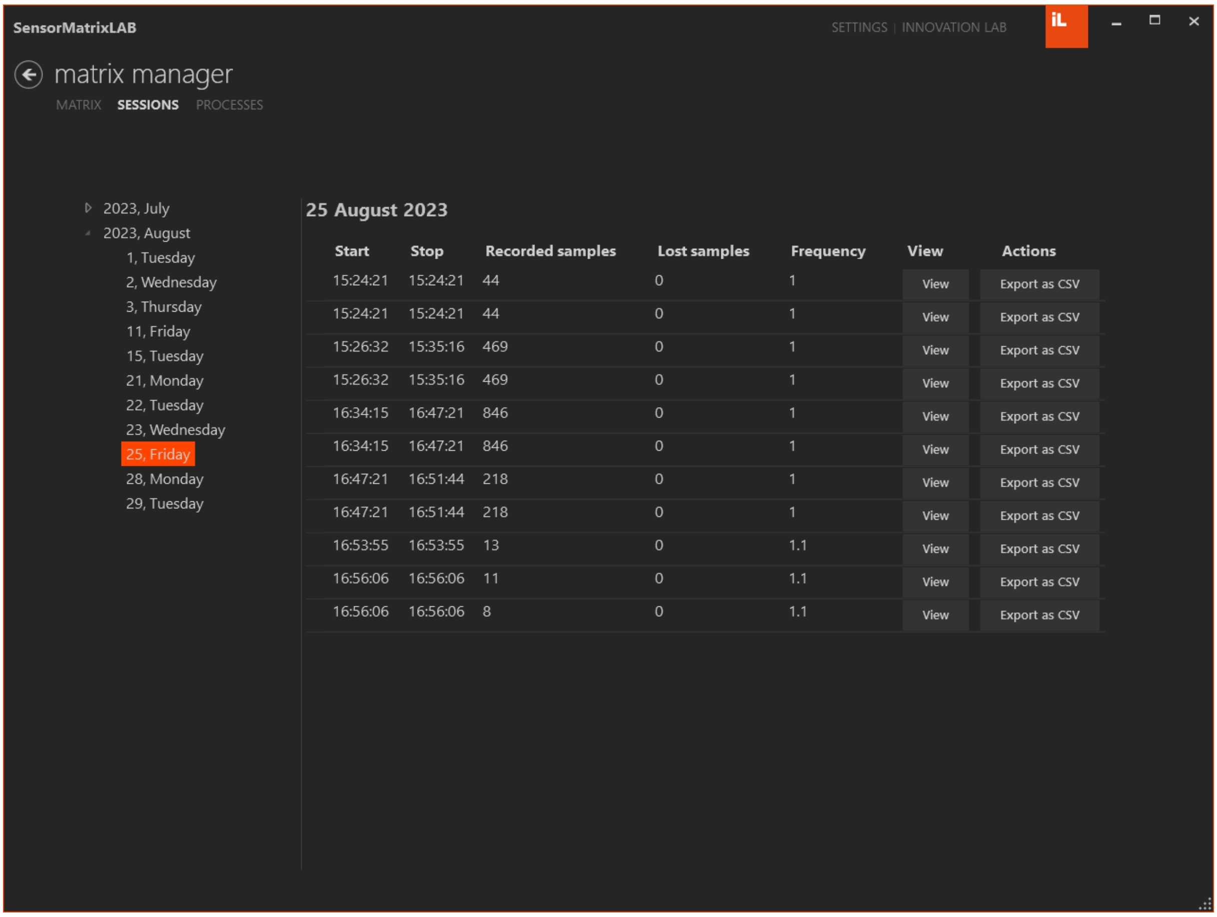
7.1.3 Matrix View

The matrix view provides a variety of features to visualize the measured data in real time. A detailed description of the matrix view can be found in the Visualization section (section 8).

7.1.4 Software and Firmware Version

The current software version is displayed in the bottom right of the *MATRIX* window. If a connection to the device is established and a measurement running, the device's firmware version is displayed in the bottom right of the window.

7.2 Sessions



- 2023, July
- 2023, August
  - 1, Tuesday
  - 2, Wednesday
  - 3, Thursday
  - 11, Friday
  - 15, Tuesday
  - 21, Monday
  - 22, Tuesday
  - 23, Wednesday
  - 25, Friday
  - 28, Monday
  - 29, Tuesday

25 August 2023

Start	Stop	Recorded samples	Lost samples	Frequency	View	Actions
15:24:21	15:24:21	44	0	1	View	Export as CSV
15:24:21	15:24:21	44	0	1	View	Export as CSV
15:26:32	15:35:16	469	0	1	View	Export as CSV
15:26:32	15:35:16	469	0	1	View	Export as CSV
16:34:15	16:47:21	846	0	1	View	Export as CSV
16:34:15	16:47:21	846	0	1	View	Export as CSV
16:47:21	16:51:44	218	0	1	View	Export as CSV
16:47:21	16:51:44	218	0	1	View	Export as CSV
16:53:55	16:53:55	13	0	1.1	View	Export as CSV
16:56:06	16:56:06	11	0	1.1	View	Export as CSV
16:56:06	16:56:06	8	0	1.1	View	Export as CSV

Figure 16: *SESSIONS window*

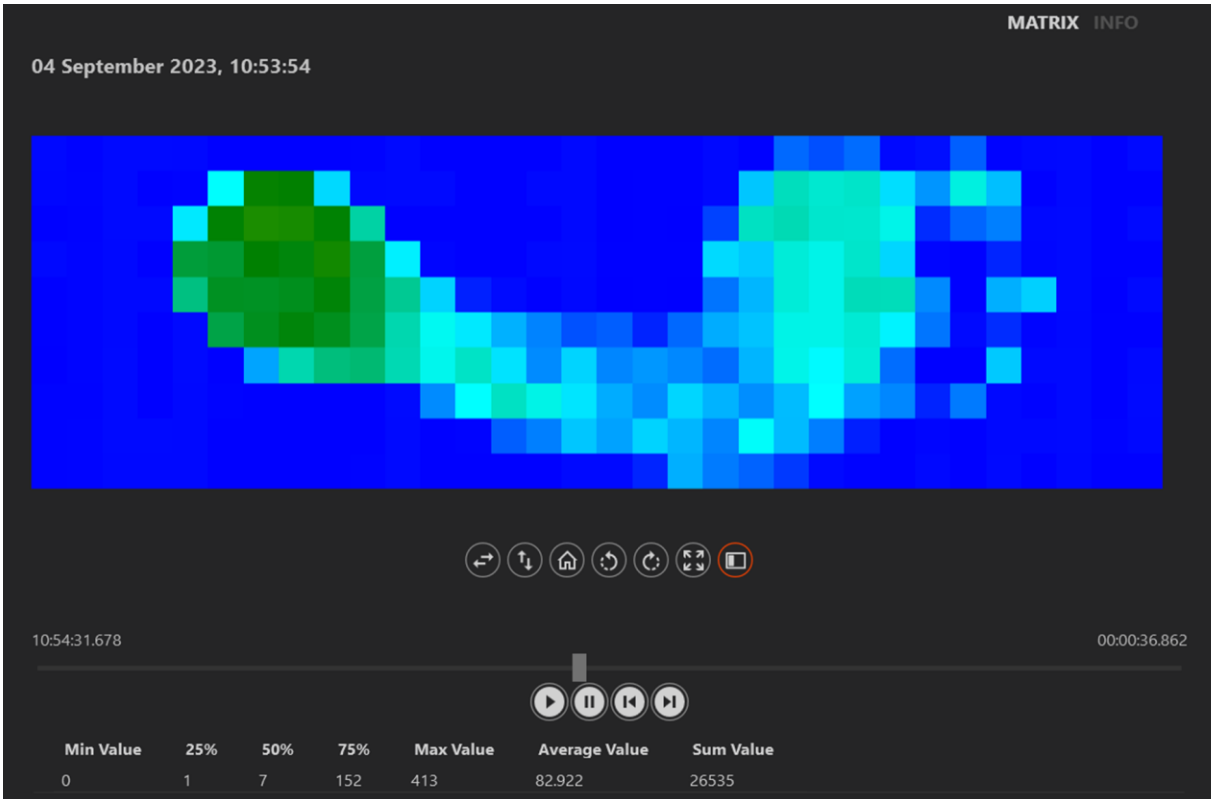
The *SESSIONS* window allows to review and export previous measurement sessions. Recordings are categorized by date and time of the measurement session. In addition, the number of recorded samples and the measurement frequency is displayed.

Lost samples describe the number of samples that have been lost during transmission. This might exclusively happen for USB connection, as a broken data package will not lead to a request for another one. However, the number of lost packages is typically very low.

NOTE: Only sessions saved in database format are displayed.

Note: Only sessions saved in the *STORAGE directory* are shown. To show different files, switch the directory (section 6.5).

Clicking on the *View* button will lead to the session view window which provides two tabs (*MATRIX* and *INFO*).

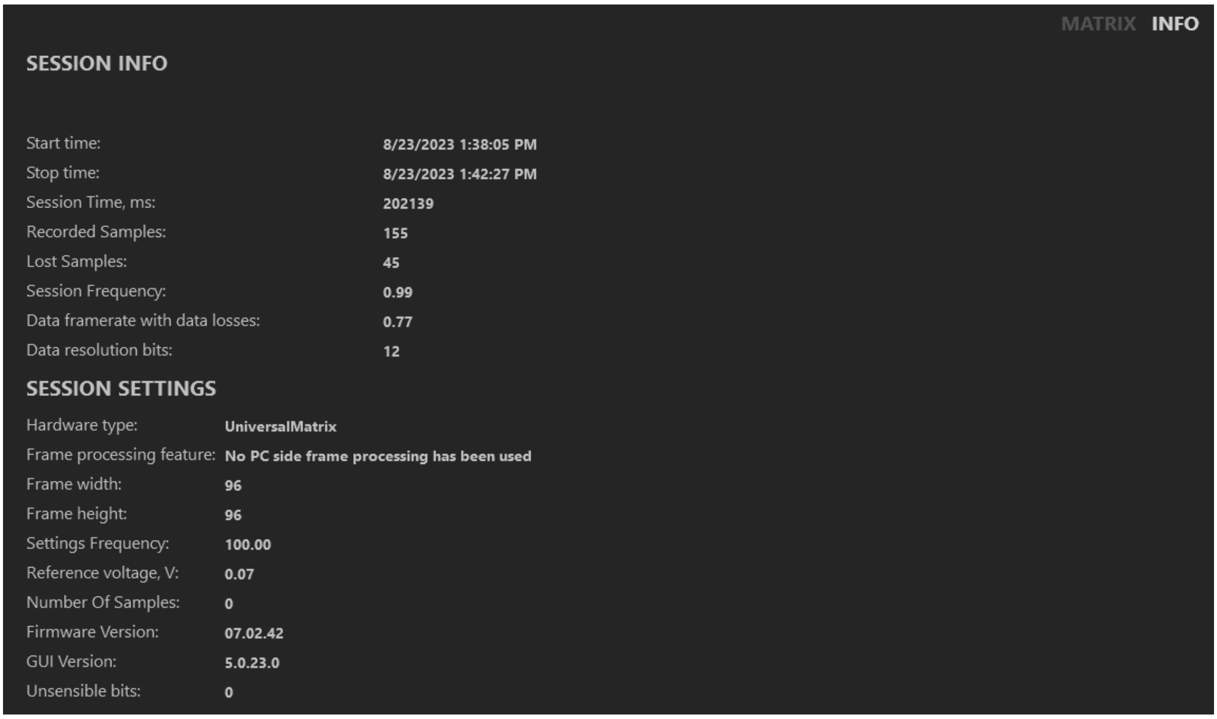


04 September 2023, 10:53:54

Min Value	25%	50%	75%	Max Value	Average Value	Sum Value
0	1	7	152	413	82.922	26535

__Figure 17:__ Session view MATRIX window showing the pressure distribution of a foot.

Figure 17 shows the *MATRIX* tab for a selected measurement session. It shares all visualization features with the live data view. Please see section 8 for more information. In addition, a data player is provided to select specific samples or automate the data play.



SESSION INFO

Start time:	8/23/2023 1:38:05 PM
Stop time:	8/23/2023 1:42:27 PM
Session Time, ms:	202139
Recorded Samples:	155
Lost Samples:	45
Session Frequency:	0.99
Data framerate with data losses:	0.77
Data resolution bits:	12

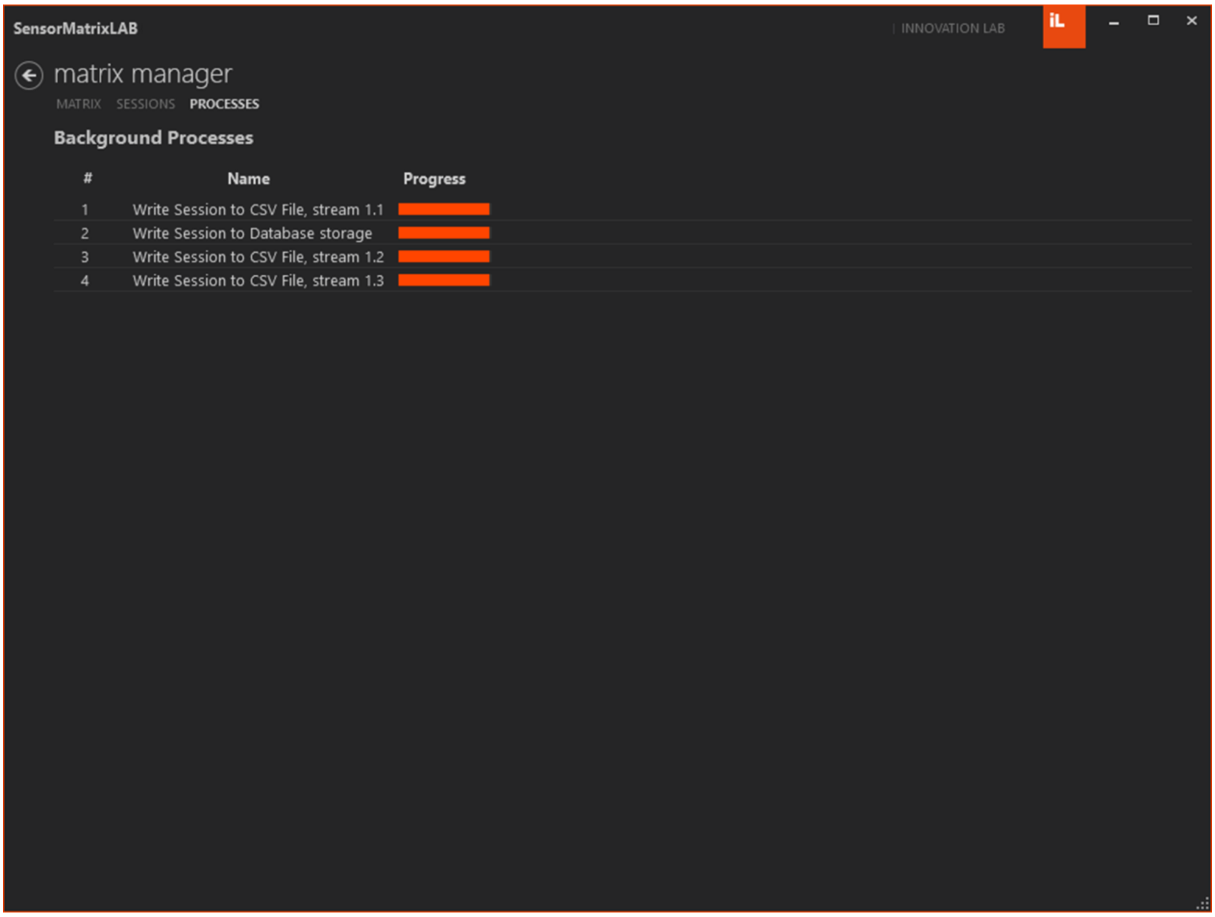
SESSION SETTINGS

Hardware type:	UniversalMatrix
Frame processing feature:	No PC side frame processing has been used
Frame width:	96
Frame height:	96
Settings Frequency:	100.00
Reference voltage, V:	0.07
Number Of Samples:	0
Firmware Version:	07.02.42
GUI Version:	5.0.23.0
Unsensible bits:	0

__Figure 18:__ Session view INFO window

The INFO tab provides the meta data of the selected measurement session, like frame size, measurement frequency, etc. (**Figure 18**).

7.3 Processes



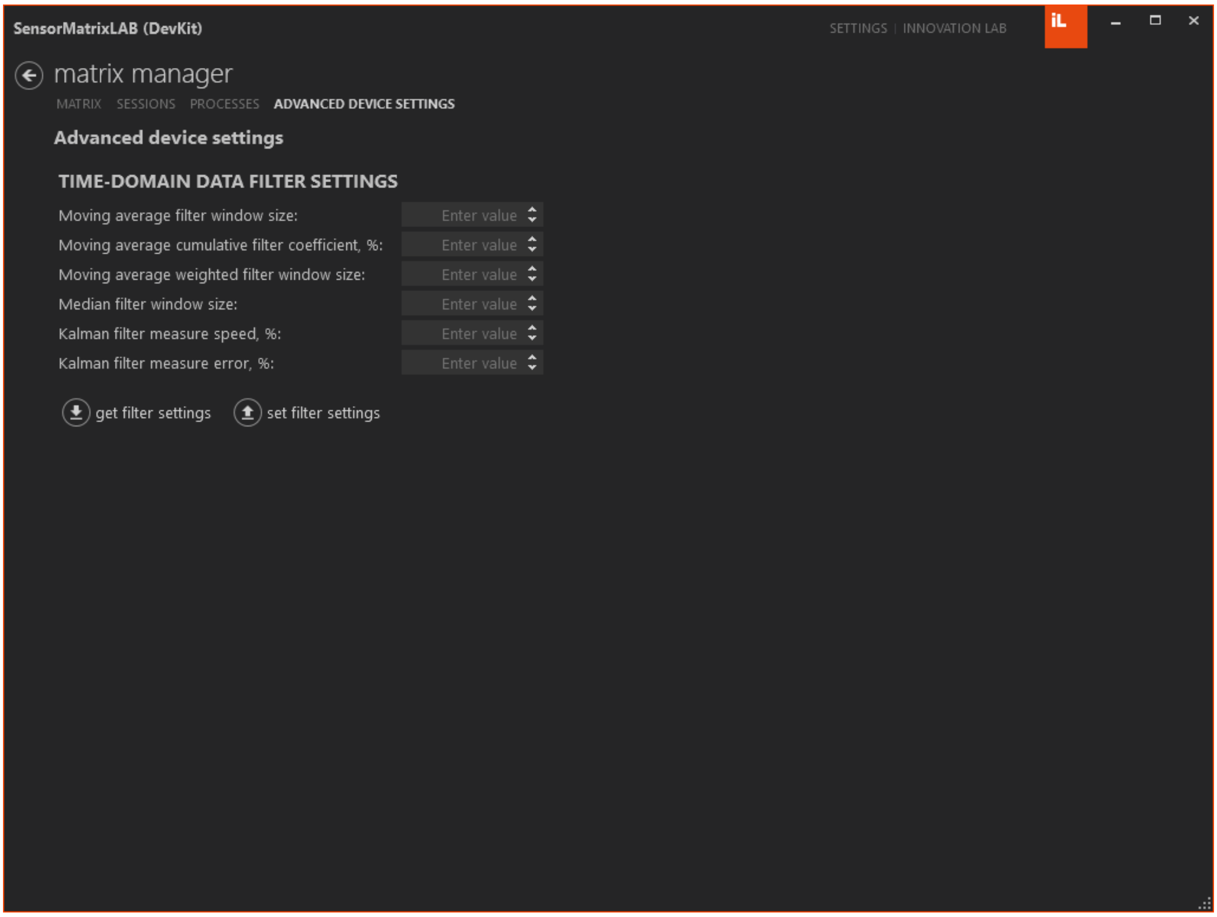
#	Name	Progress
1	Write Session to CSV File, stream 1.1	
2	Write Session to Database storage	
3	Write Session to CSV File, stream 1.2	
4	Write Session to CSV File, stream 1.3	

Figure 19: *PROCESSES window*

This tab contains a list of running subprocesses, such as CSV and database file writing. A progress bar shows the subprocess' progress.

NOTE: The performance of file writing subprocesses can be affected by the accessibility of the save destination. In case of file writing subprocesses adding up, consider selecting a different directory in the *STORAGE* settings (6.5).

7.4 Advanced Device Settings (Devkit only)


SensorMatrixLAB (DevKit)

matrix manager

Advanced device settings

TIME-DOMAIN DATA FILTER SETTINGS

Moving average filter window size: Enter value

Moving average cumulative filter coefficient, %: Enter value

Moving average weighted filter window size: Enter value

Median filter window size: Enter value

Kalman filter measure speed, %: Enter value

Kalman filter measure error, %: Enter value

Figure 20: *ADVANCED DEVICE SETTINGS window*

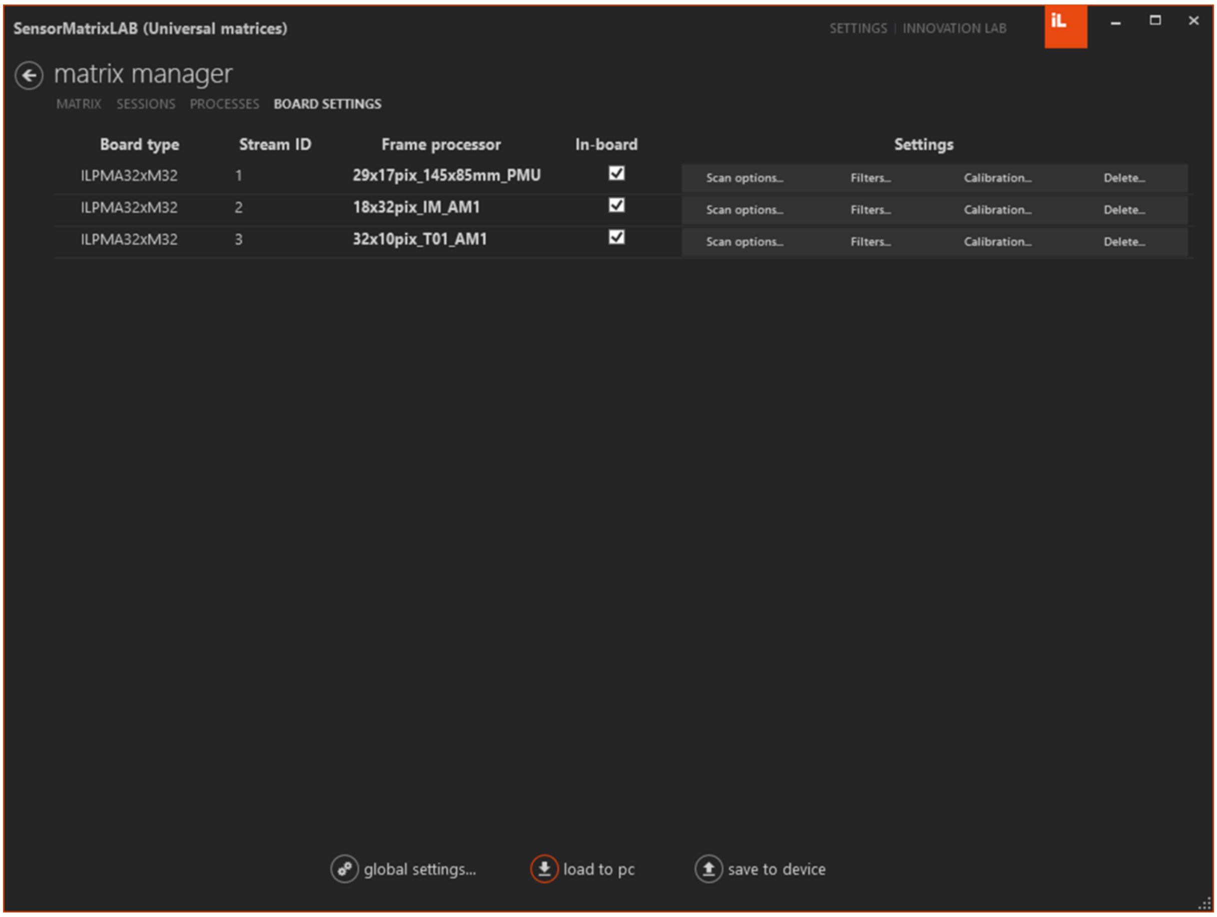
Figure 20 shows the *ADVANCED DEVICE SETTINGS* tab. It provides additional on-board features which are described in detail in the following subsections.

7.4.1 Time Filter Settings

In this section, the on-board time domain filtering of the sensor data can be enabled and configured. This reduces measurement noises and delivers a steady output. Filter computation is performed for each pixel individually.

A detailed description about the filters and their configuration is given in section 9.1.

7.5 Board Settings (Universal matrix only)



Board type	Stream ID	Frame processor	In-board	Settings			
ILPMA32xM32	1	29x17pix_145x85mm_PMU	☑	Scan options...	Filters...	Calibration...	Delete...
ILPMA32xM32	2	18x32pix_IM_AM1	☑	Scan options...	Filters...	Calibration...	Delete...
ILPMA32xM32	3	32x10pix_T01_AM1	☑	Scan options...	Filters...	Calibration...	Delete...

Figure 21: *BOARD SETTINGS window*

The board settings are stored in the non-volatile memory of the hardware and the same settings are used on all supported interfaces (e.g., CAN, ETH). Several streams can be configured for a single board. Those streams can read different (or same) readout regions with different settings (e.g., reference voltage, frequency). The stream information is stored on an MicroSD card applied to the PMU board (see the hardware manual for more information).

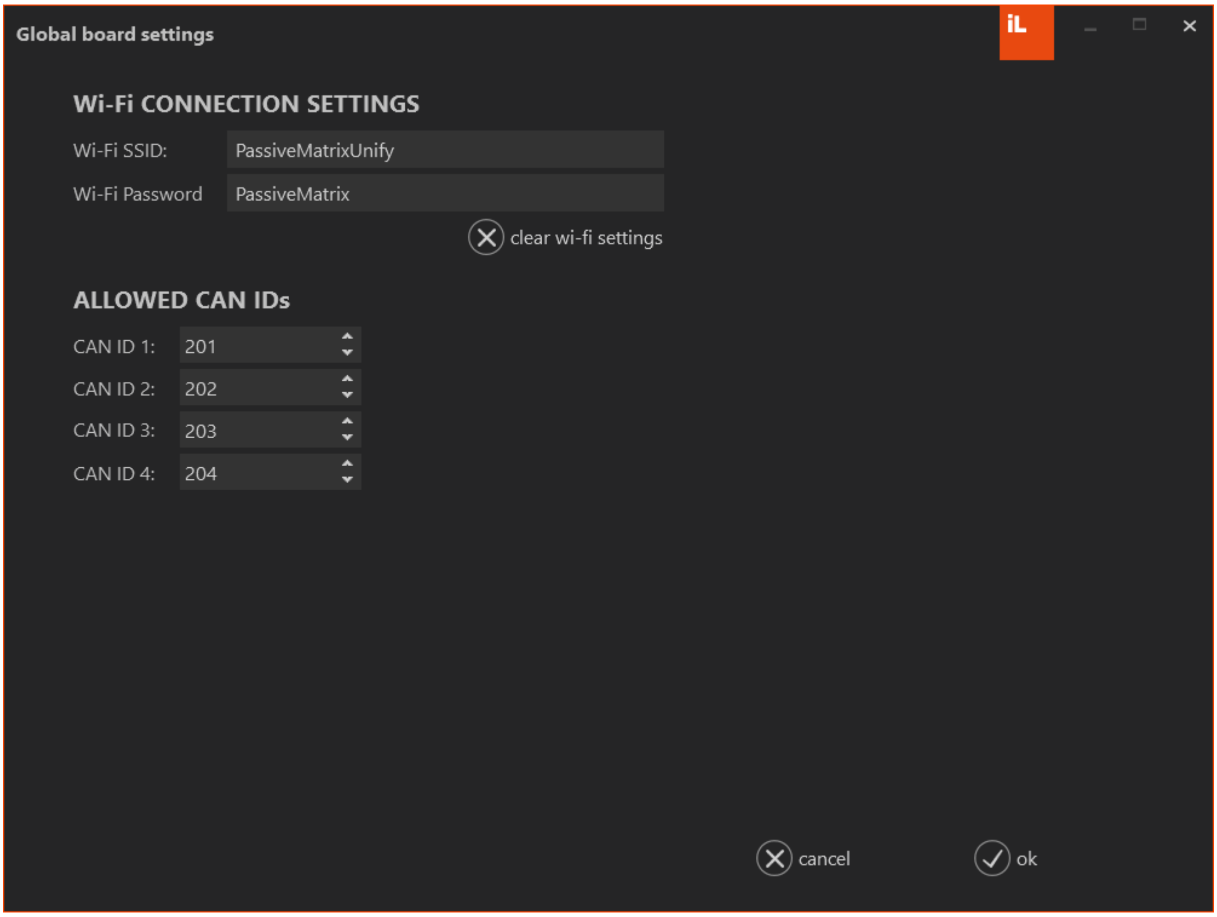
NOTE: To read and update the device's stream configurations, an existing connection to the hardware is required and the session must be stopped.

Use the *load to pc button* to read the stream configurations of the connected device. This process might take a while. Afterwards, all streams available are displayed in the *BOARD SETTINGS* window (**Figure 21**). After some changes have been made, they can be written to the device using the *save to device button*.

NOTE: Some changes might require the board to restart. In this case, if the *connect* button is pressed, the board will restart automatically. Wait until the device's LED is blinking green and try to connect again.

In the following, all settings will be explained in detail.

7.5.1 Global Device Settings



Global board settings

Wi-Fi CONNECTION SETTINGS

Wi-Fi SSID: PassiveMatrixUnify

Wi-Fi Password PassiveMatrix

ALLOWED CAN IDs

CAN ID 1: 201

CAN ID 2: 202

CAN ID 3: 203

CAN ID 4: 204

Figure 22: *Global board settings window*

Click the *global settings* button on the device settings window (**Figure 21**) to open the global board settings window (**Figure 22**). This window contains non stream specific board parameters. For Wi-Fi functionality the SSID and password can be set in the *Wi-Fi CONNECTION SETTINGS* section. If the CAN interface is used, the *ALLOWED CAN IDs* specify up to four CAN IDs from whom the board will accept commands.

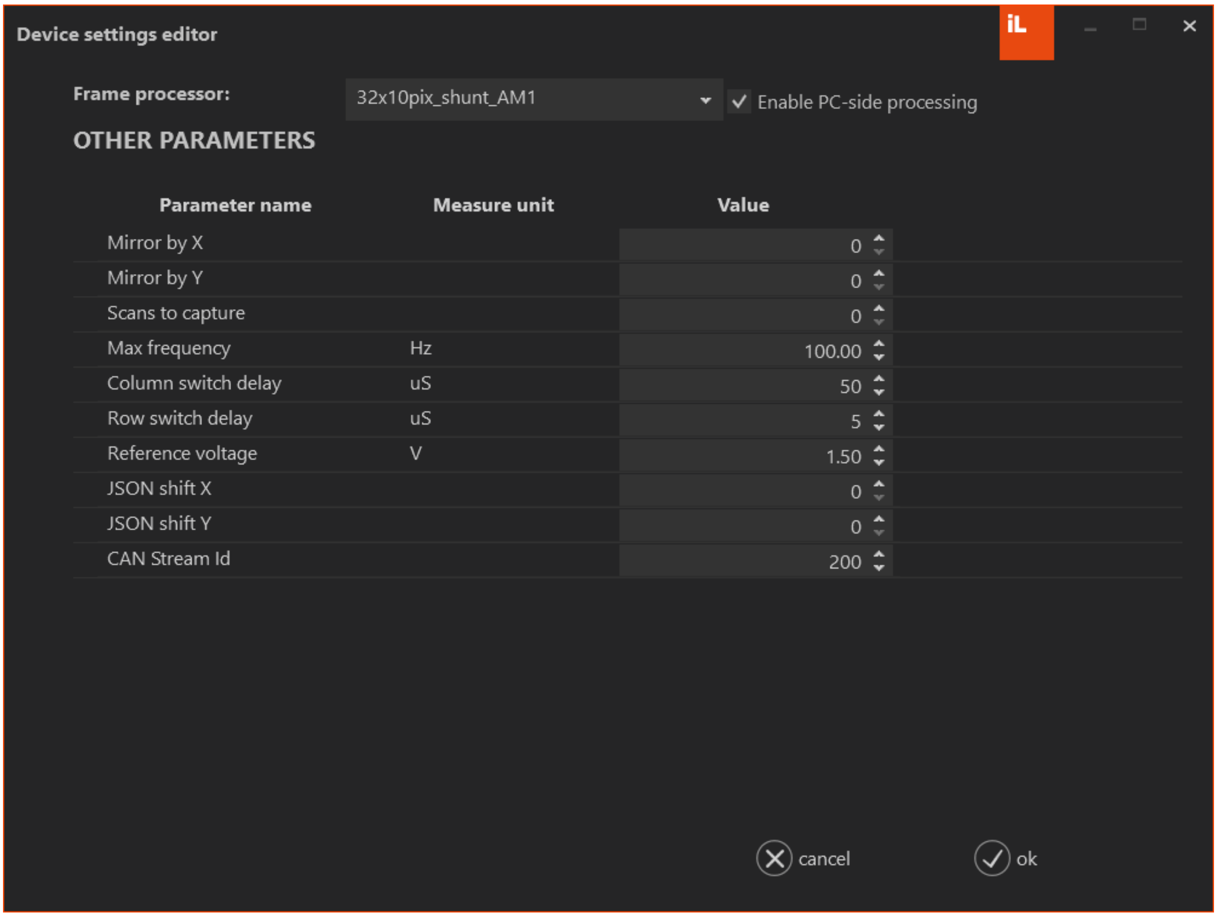
NOTE: For additional information regarding the CAN interface, please see the hardware manual.

7.5.2 In-Board Frame Processing

In the device settings window (**Figure 21**), for each stream a frame processor is declared. This frame processor specifies the readout region according to the selected sensor matrix. The frame processing information is stored on the MicroSD card and is typically pre-configured according to the ordered sensor matrices. To disable the frame processing the *In-board* checkbox can be disabled.

NOTE: The frame processing can also be performed software sided. See section 7.5.3 for more information.

7.5.3 Scan Options



Device settings editor

Frame processor: 32x10pix_shunt_AM1

☑ Enable PC-side processing

OTHER PARAMETERS

Parameter name	Measure unit	Value
Mirror by X		0
Mirror by Y		0
Scans to capture		0
Max frequency	Hz	100.00
Column switch delay	uS	50
Row switch delay	uS	5
Reference voltage	V	1.50
JSON shift X		0
JSON shift Y		0
CAN Stream Id		200

Figure 23: *Scan Options editor*

Click the *Scan options* button (**Figure 21**) for a specific stream to open the scan options editor (**Figure 23**).

The *Frame processor* option allows the selection of a specific processor for software-sided frame processing. It must be activated with the *Enable PC-side processing* checkbox.

NOTE: Make sure that *In-board* frame processing is disabled to prevent unwanted behavior by doubled frame processing.
NOTE: Compared to In-board frame processing, PC-sided frame processing reduces the maximum readout frequency because the complete readout area (96x96) must be measured before transmission to the PC. In-board frame processing is recommended.

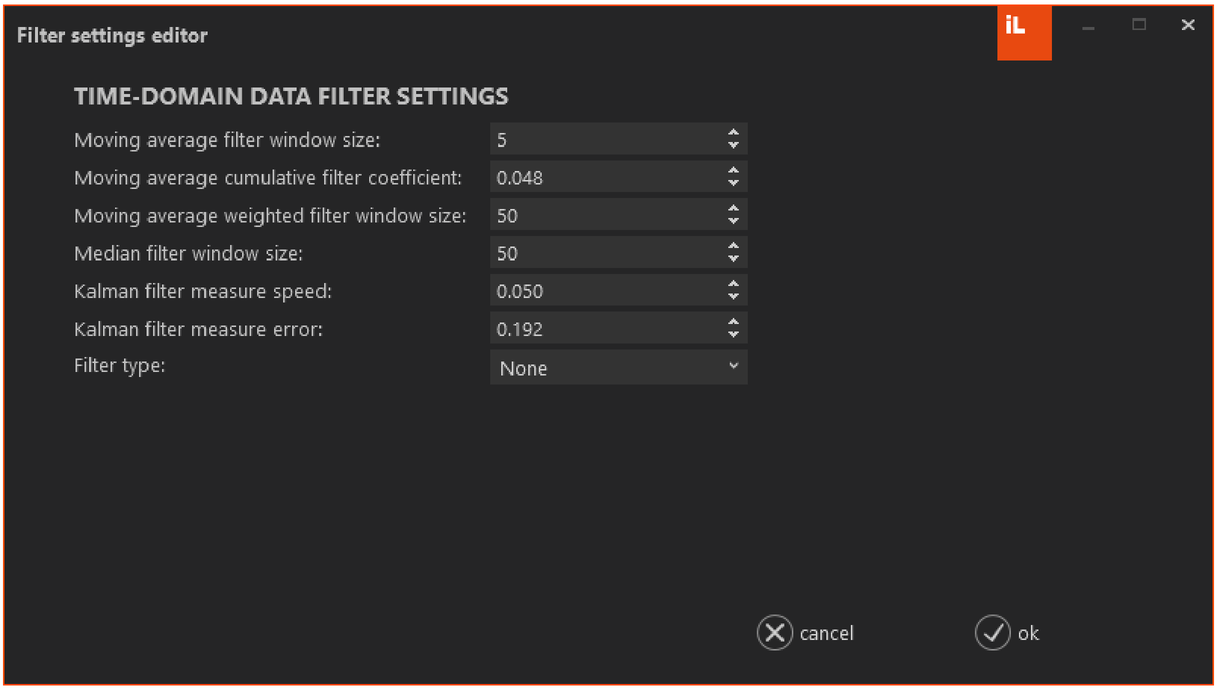
The other stream parameter’s descriptions and details are listed as follows.

Name	Description	Unit	Type	Min. Value	Max. Value
Mirror by X	mirrors the view by the x-axis	None	bool	0	1
Mirror by Y	mirrors the view by the y-axis	None	bool	0	1
Scans to capture	allows you to limit the number of captured images per session. When this number is reached, the session will be stopped. If this value is set to 0, the session is not limited and can only be stopped using the *disconnect* button.	None	uint	0	65535
Max Frequency	describes the upper limit for the measurement frequency of a session in Hz (samples per second). If this value is set to 0, the measurement will be performed at the maximum frequency possible.	Hz	float	0.0	1000.0
Column switch delay	allows you to set a delay between multiplexing the readout area’s columns. This value can affect image quality and provides control over timing.	µs	uint	0	65535
Row switch delay	allows you to set a delay between multiplexing the readout area’s rows. This value can affect image quality and provides control over timing.	µs	uint	0	65535
Reference voltage	allows you to set a reference voltage value on the board. In general, the higher the reference, the higher the sensitivity of the sensor system to external stimulus. It is recommended to start measurements with a low reference voltage and to increase it if necessary.	V	float	0.0	3.2885
JSON shift X	Allows to shift the frame processing area (provided as a JSON file) in x-direction.	None	bool	0	1
JSON shift Y	Allows to shift the frame processing area (provided as a JSON file) in y-direction.	None	bool	0	1
CAN Stream Id	Specifies a CAN frame ID for the selected stream. If data is transmitted via the CAN interface, data for the selected stream is transmitted with the ID given.	None	uint	0	2047

NOTE: If frame processing is configured for any matrix on analog module 1, then it can also be read from the analog module 2 connector if JSON shift X and JSON shift Y is set to 32.

NOTE: For additional information regarding the CAN interface, please see the hardware manual.

7.5.4 Time Filter Settings



Filter settings editor

TIME-DOMAIN DATA FILTER SETTINGS

Moving average filter window size: 5

Moving average cumulative filter coefficient: 0.048

Moving average weighted filter window size: 50

Median filter window size: 50

Kalman filter measure speed: 0.050

Kalman filter measure error: 0.192

Filter type: None

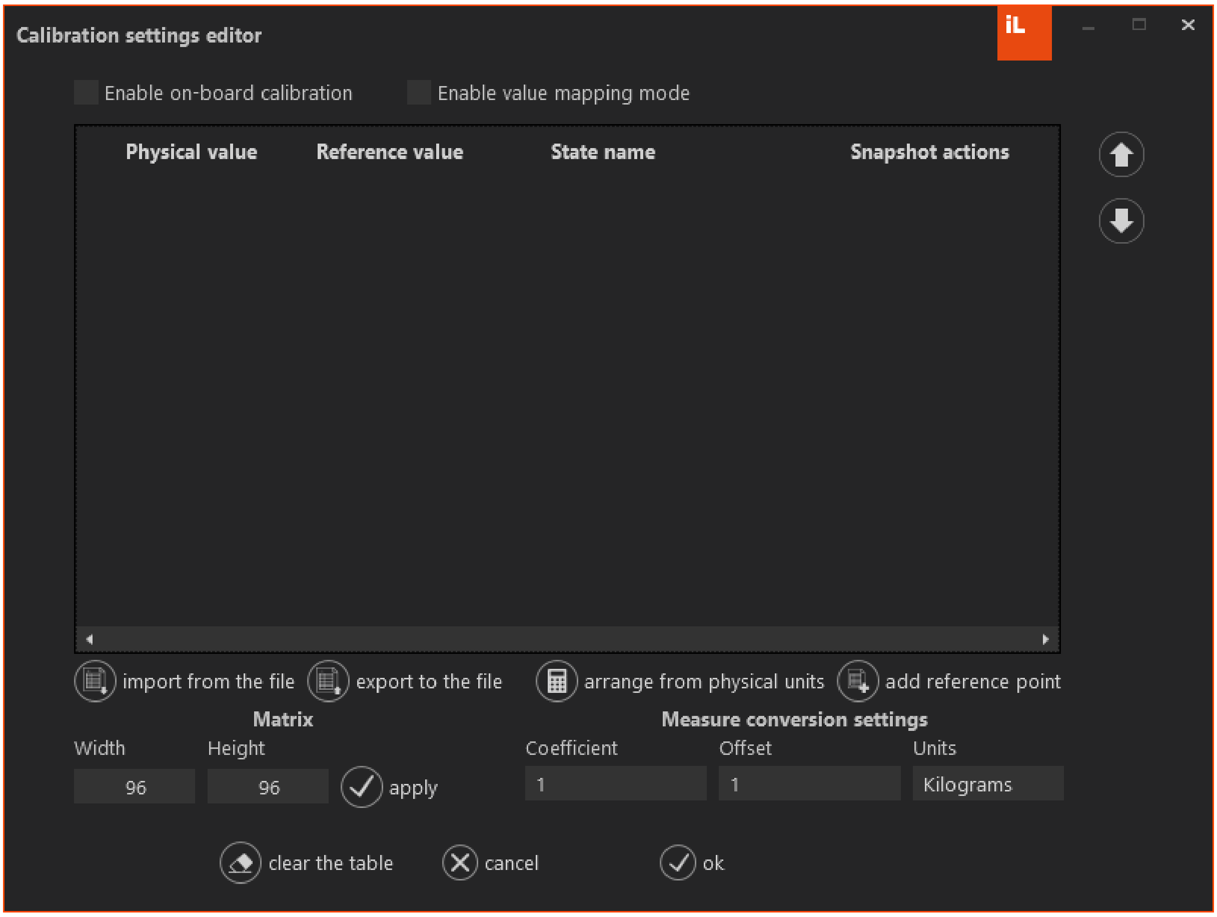
Figure 24: *Filter settings editor*

Click the *Filters* button (**Figure 21**) for a specific stream to open the filter settings editor (**Figure 24**).

In this window, the on-board time domain filtering of the sensor data can be enabled and configured. This reduces measurement noises and delivers a steady output. Filter computation is performed for each pixel individually.

A detailed description about the filters and their configuration is given in section 9.1.

7.5.5 Calibration Settings



□ Enable on-board calibration □ Enable value mapping mode

Physical value	Reference value	State name	Snapshot actions

Width: 96
Height: 96
Coefficient: 1
Offset: 1
Units: Kilograms

__Figure 25:__ Calibration settings editor

Click the *Calibration* button (**Figure 21**) for a specific stream to open the calibration settings editor (**Figure 25**).

This window allows to import, export, and generate calibration data. Typically, a mapping from original measurements to physical values based on calibration data is performed in two steps. First, the original 12-bit measurement values are adjusted on-board. Second, those modified 12-bit values are transferred to a physical value on software side. Both steps can be enabled and disabled using the *Enable on-board calibration* and *Enable value mapping mode* checkboxes.

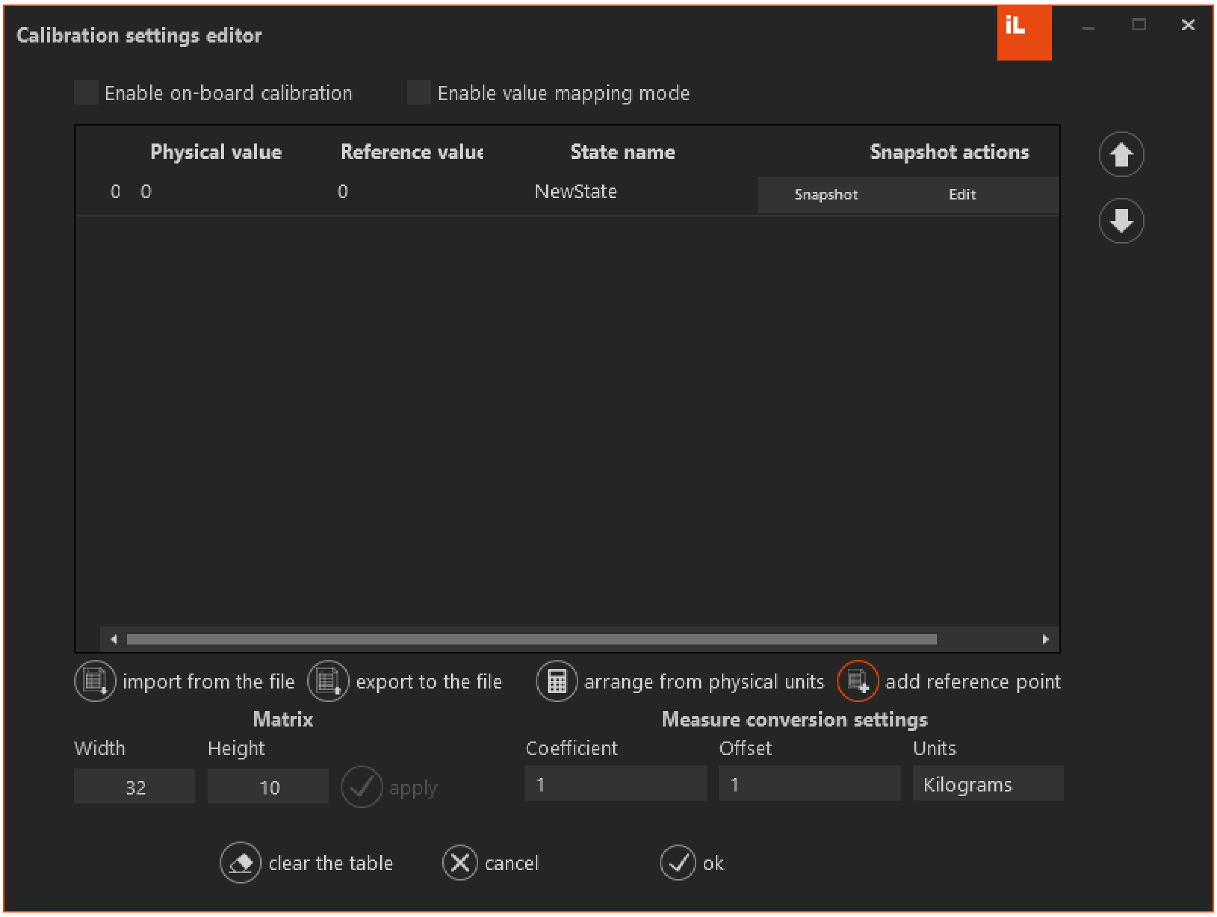
NOTE: A detailed explanation about the calibration concept is given in section 0.

The calibration data can be stored in an MS-Excel File (XLSX). To import or export calibration data, the *import from the file* and *export to the file* buttons are used. To confirm the calibration settings hit the *ok* button and load the new settings to the device to apply them.

7.5.5.1 Reference Points

Reference points are used to assign matrix measurement data to a single physical value. To generate a new snapshot, first make sure that the correct width and height is selected. Change the *Width* and *Height* fields according to your stream configuration and apply. Next, click the *add reference point* button and a new reference point entry is added to the list (**Figure 26**).

NOTE: Make sure that on-board calibration is disabled on your board. If on-board calibration is enabled, disable the checkbox, click ok and press the *save to device* button before continuing.



Calibration settings editor

□ Enable on-board calibration □ Enable value mapping mode

Physical value	Reference value	State name	Snapshot actions
0 0	0	NewState	Snapshot Edit

Matrix

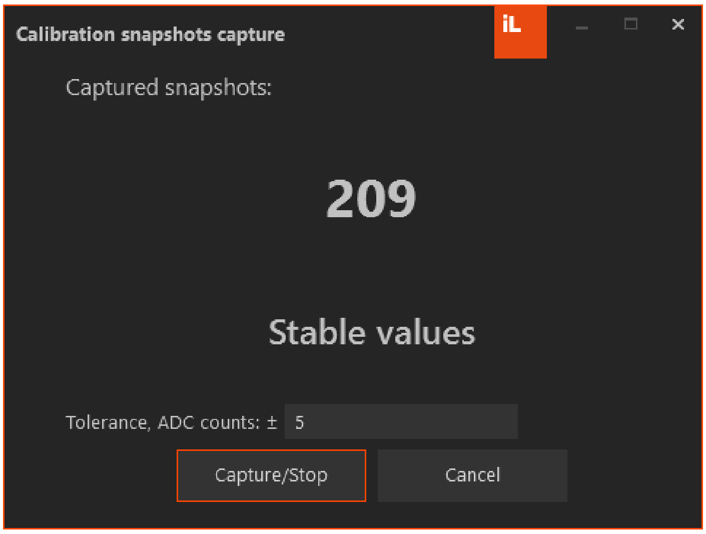
Width: 32 Height: 10

Measure conversion settings

Coefficient: 1 Offset: 1 Units: Kilograms

Figure 26: *Adding a snapshot.*

To capture the matrix data for the current reference point, click the *Snapshot* button. The calibration snapshot capture window will open (**Figure 27**).



Calibration snapshots capture

Captured snapshots:

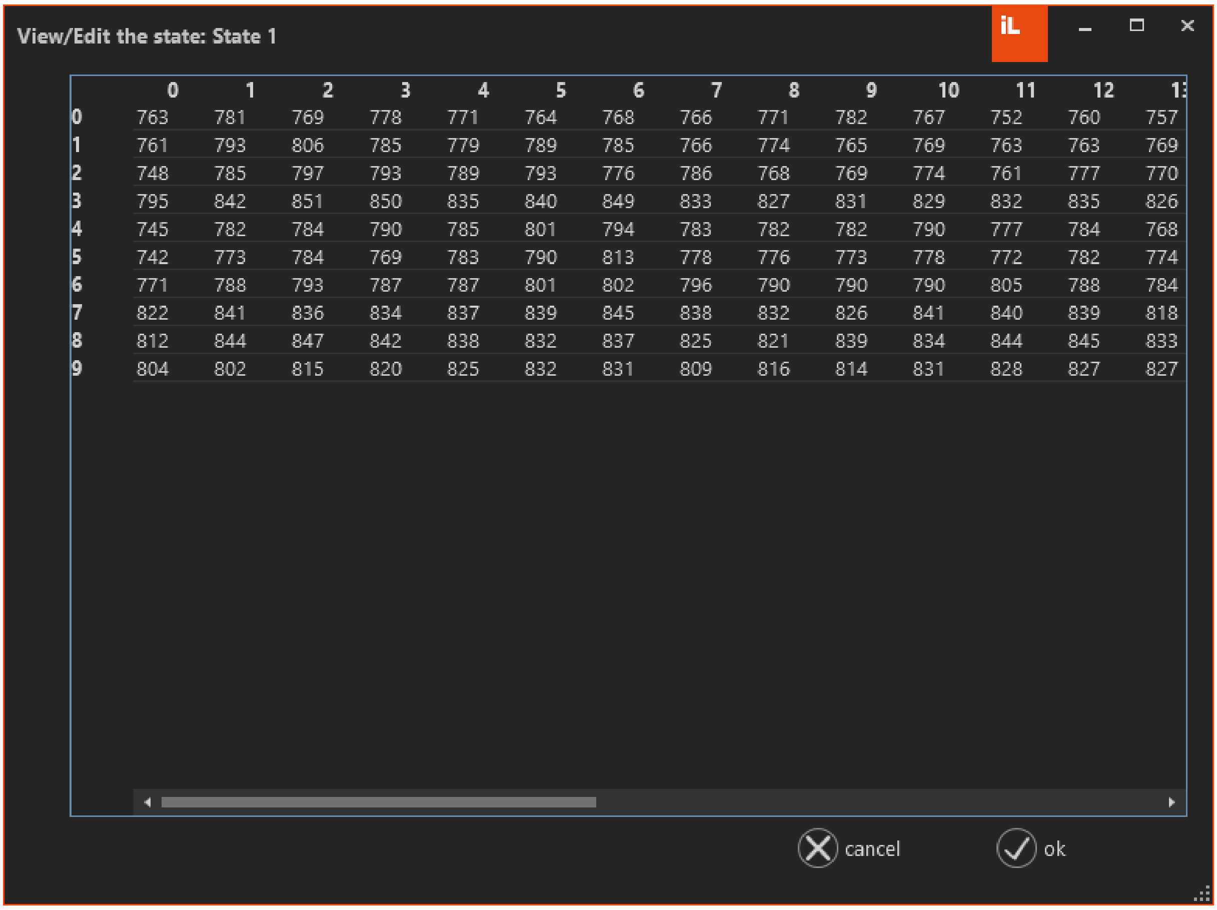
209

Stable values

Tolerance, ADC counts: ± 5

Figure 27: *Calibration snapshot capture window*

With the Capture/Stop button, matrix measurements are started and stopped. The number in the window's center indicates the number of recorded matrix samples. To guarantee stable measurement conditions, a tolerance window can be set. If the measurements for some sensor cells will leave this window, their position is indicated, and no snapshot is obtained. After leaving the window, the average matrix over all recorded samples is stored. This matrix can be reviewed and edited using the *Edit* button (**Figure 28**).



View/Edit the state: State 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	763	781	769	778	771	764	768	766	771	782	767	752	760	757
1	761	793	806	785	779	789	785	766	774	765	769	763	763	769
2	748	785	797	793	789	793	776	786	768	769	774	761	777	770
3	795	842	851	850	835	840	849	833	827	831	829	832	835	826
4	745	782	784	790	785	801	794	783	782	782	790	777	784	768
5	742	773	784	769	783	790	813	778	776	773	778	772	782	774
6	771	788	793	787	787	801	802	796	790	790	790	805	788	784
7	822	841	836	834	837	839	845	838	832	826	841	840	839	818
8	812	844	847	842	838	832	837	825	821	839	834	844	845	833
9	804	802	815	820	825	832	831	809	816	814	831	828	827	827

***Figure 28:** View/Edit state window.*

Optionally, a physical value, a reference value (at least one of both) and a state name can be assigned to the reference point.

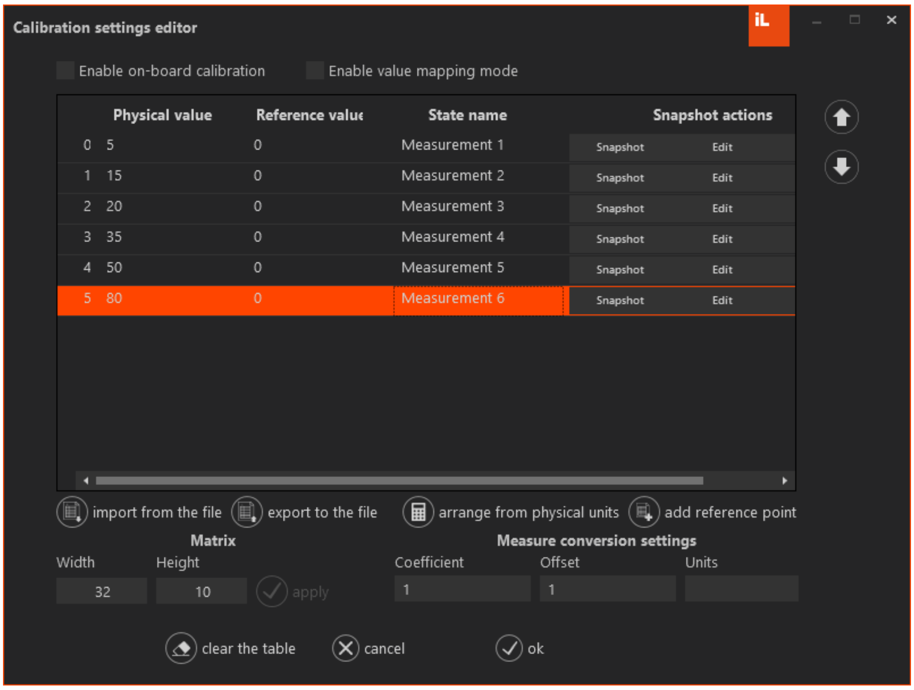
NOTE: At least two reference points are required to use the calibration data saved to a device.

7.5.5.2 Calibration Example

In the following, an example for calibration of a temperature sensor matrix is given. Consider the sensor matrix is placed in a temperature-controlled setup, that applies the same temperature to all sensor cells of the matrix.

NOTE: No matter which physical values are referenced to a sensor matrix (e.g., pressure, temperature, etc.), the setup for calibration requires to apply the same value to each single sensor cell. In addition, it is required to know the physical reference value as precise as possible.

1. Select the corresponding stream for the temperature matrix and open the calibration settings editor.
2. Apply several temperatures to the sensor matrix, record snapshots and notate the corresponding temperature values in the *physical value* fields (**Figure 29**).



Calibration settings editor

□ Enable on-board calibration □ Enable value mapping mode

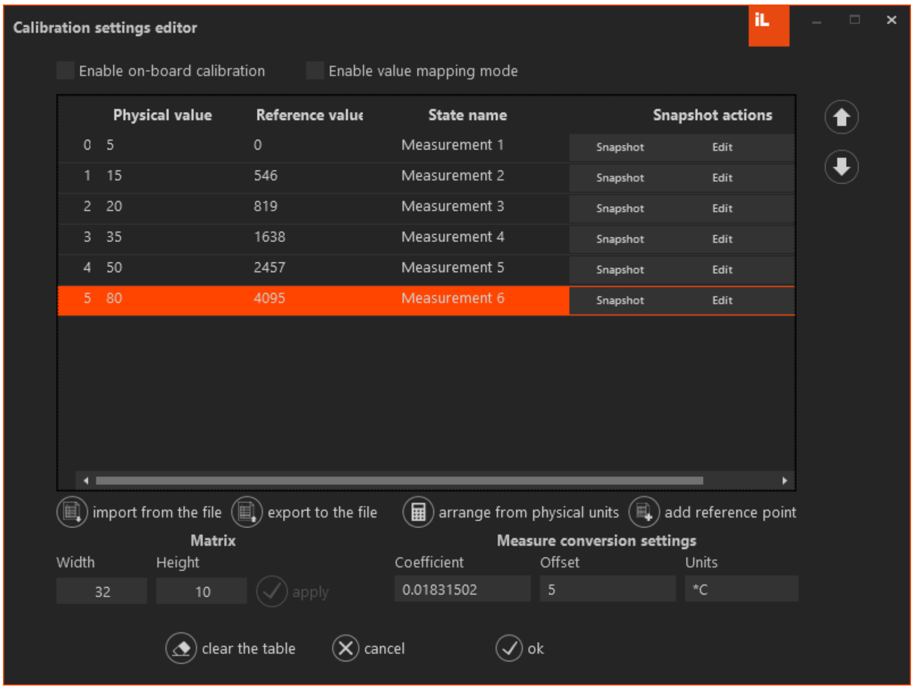
	Physical value	Reference value	State name	Snapshot actions	
0	5	0	Measurement 1	Snapshot	Edit
1	15	0	Measurement 2	Snapshot	Edit
2	20	0	Measurement 3	Snapshot	Edit
3	35	0	Measurement 4	Snapshot	Edit
4	50	0	Measurement 5	Snapshot	Edit
5	80	0	Measurement 6	Snapshot	Edit

Matrix: Width 32, Height 10

Measure conversion settings: Coefficient 1, Offset 1, Units

Figure 29: *Calibration example step 2*

3. Click the *arrange from physical units* button to automatically compute the reference values for on-board calibration as well as the parameter for value mapping (*Coefficient* and *Offset*). Assign the correct unit (in this case °C) to the Units field (**Figure 30**).



Calibration settings editor

□ Enable on-board calibration □ Enable value mapping mode

	Physical value	Reference value	State name	Snapshot actions	
0	5	0	Measurement 1	Snapshot	Edit
1	15	546	Measurement 2	Snapshot	Edit
2	20	819	Measurement 3	Snapshot	Edit
3	35	1638	Measurement 4	Snapshot	Edit
4	50	2457	Measurement 5	Snapshot	Edit
5	80	4095	Measurement 6	Snapshot	Edit

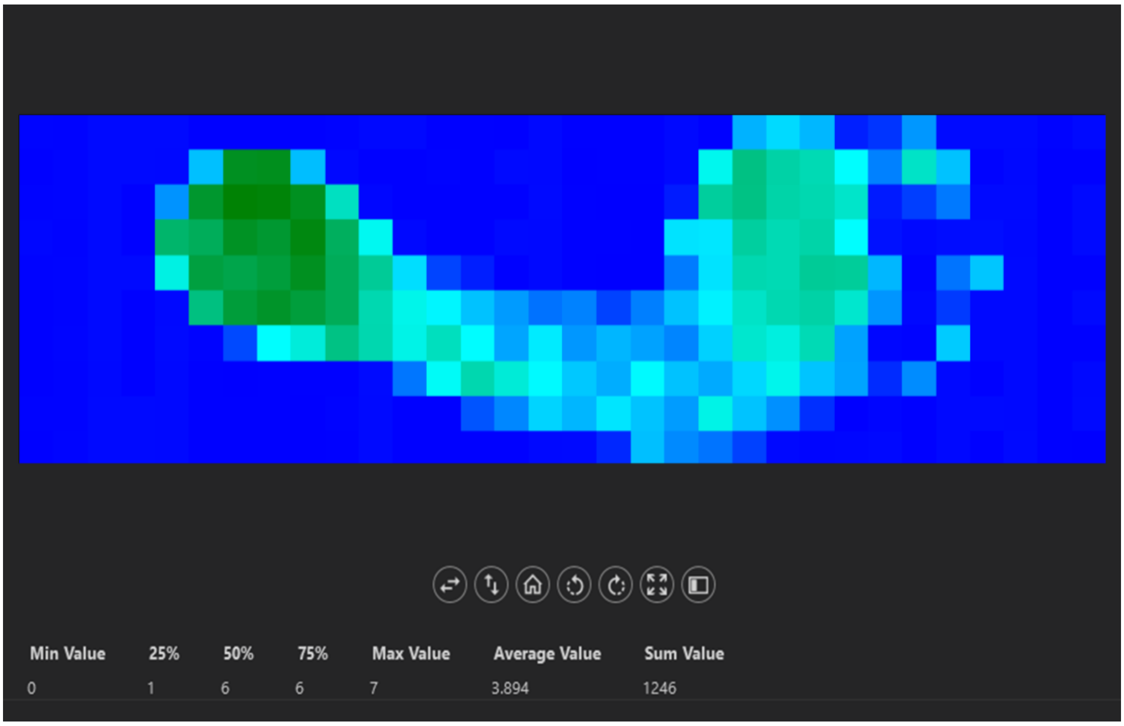
Matrix: Width 32, Height 10

Measure conversion settings: Coefficient 0.01831502, Offset 5, Units °C

Figure 30: *Calibration example step 3.*

4. Enable both calibration checkboxes. Click *ok* and save the settings to the device. The next time a measurement session is started, the board will restart. Afterwards, the physical values will be displayed in the matrix visualization during a measurement.

8 Visualization



Min Value	25%	50%	75%	Max Value	Average Value	Sum Value
0	1	6	6	7	3.894	1246

Figure 31: *Visualization area showing the pressure distribution of a foot.*

Figure 31 shows the visualization area, accessible in the *MATRIX* main tab (section 7.1) and session view (section 7.2). It provides several features which are described in the following subsections in detail.

8.1 Control Panel

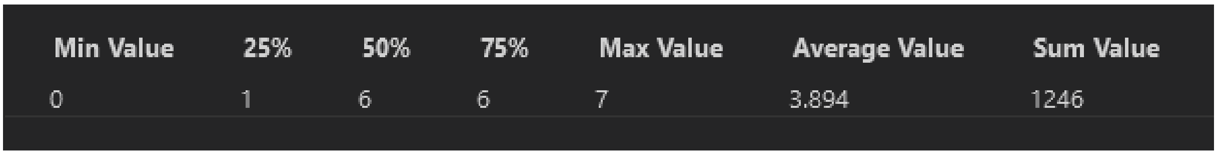
The control panel is placed directly below the view port. Its functions are listed below.

Button	Description
⇄	Mirror the view vertically
⇅	Mirror the view horizontally
⌂	Return to default view
↺	Rotate the view by a 90° left turn
↻	Rotate the view by a 90° right turn
⛶	Switch to fullscreen mode
◧	Open/close the advanced visualization settings panel (section 8.3)

8.2 Matrix Statistics

Below the control panel, the statistics panel is placed (**Figure 32**). The current view’s matrix statistics are displayed here. These are (from left to right):

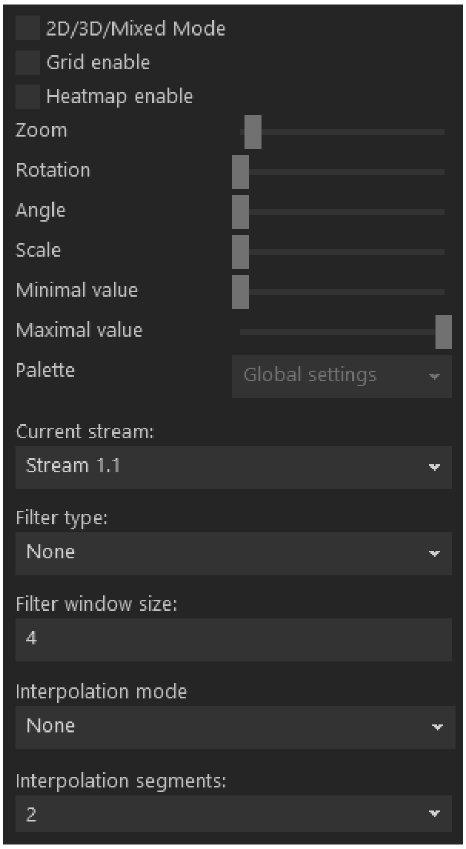
- Minimum value
- 25% percentile
- 50% percentile (median)
- 75% percentile
- Maximum value
- Average value
- Sum of all values



Min Value	25%	50%	75%	Max Value	Average Value	Sum Value
0	1	6	6	7	3.894	1246

__Figure 32:__ Matrix statistics panel

8.3 Advanced Visualization Settings



□ 2D/3D/Mixed Mode

□ Grid enable

□ Heatmap enable

Palette: Global settings

Current stream: Stream 1.1

Filter type: None

Filter window size: 4

Interpolation mode: None

Interpolation segments: 2

__Figure 33:__ Advanced visualization settings panel

Figure 33 shows the advanced visualization settings panel which can be accessed using the rightest button in the control panel (section 8.1). Its features are described in the following subsections.

8.3.1 General Settings

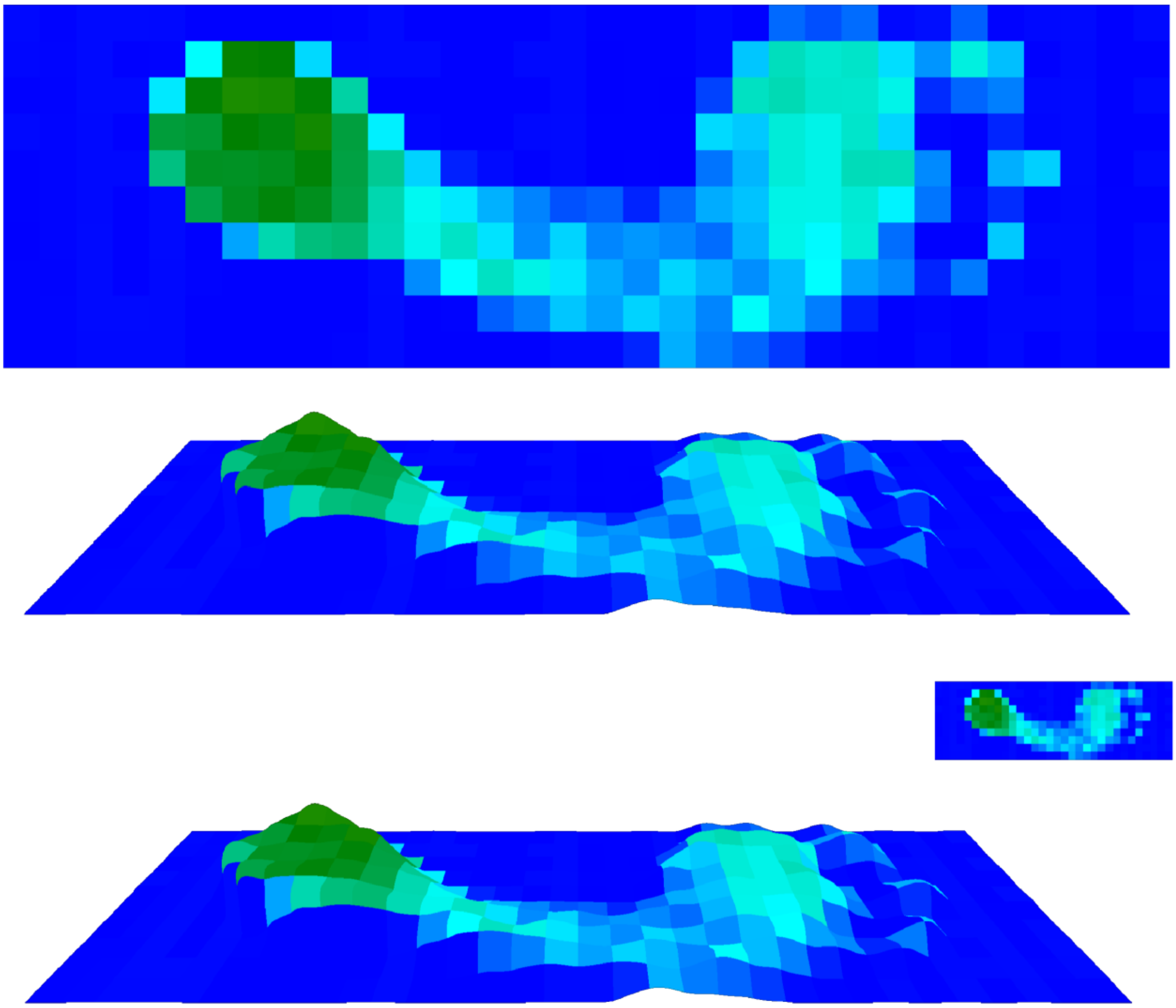


Figure 34: *Presentation of the three visualizations modes on the example of a foot’s pressure distribution. Top to bottom: 2D mode, 3D mode, mixed mode*

The *2D/3D/Mixed Mode* checkbox allows to switch between the three visualizations modes (**Figure 34**). The *Grid enable* checkbox controls if a grid is projected onto the matrix visualization or not. To switch between pixelated and interpolated visualization, the *Heatmap enable* checkbox can be used. The sliders *Minimal value* and *Maximum value* adjust the boundaries for the colormap.

With 3D mode or mixed mode enabled, four slider controls are provided to adjust the view in the 3D scene. Those are *Zoom*, *Rotation*, *Angle* and *Scale*.

If several streams are configured, the *Current stream* dropdown menu can be used to select a specific stream for visualization.

8.3.2 Image Filtering

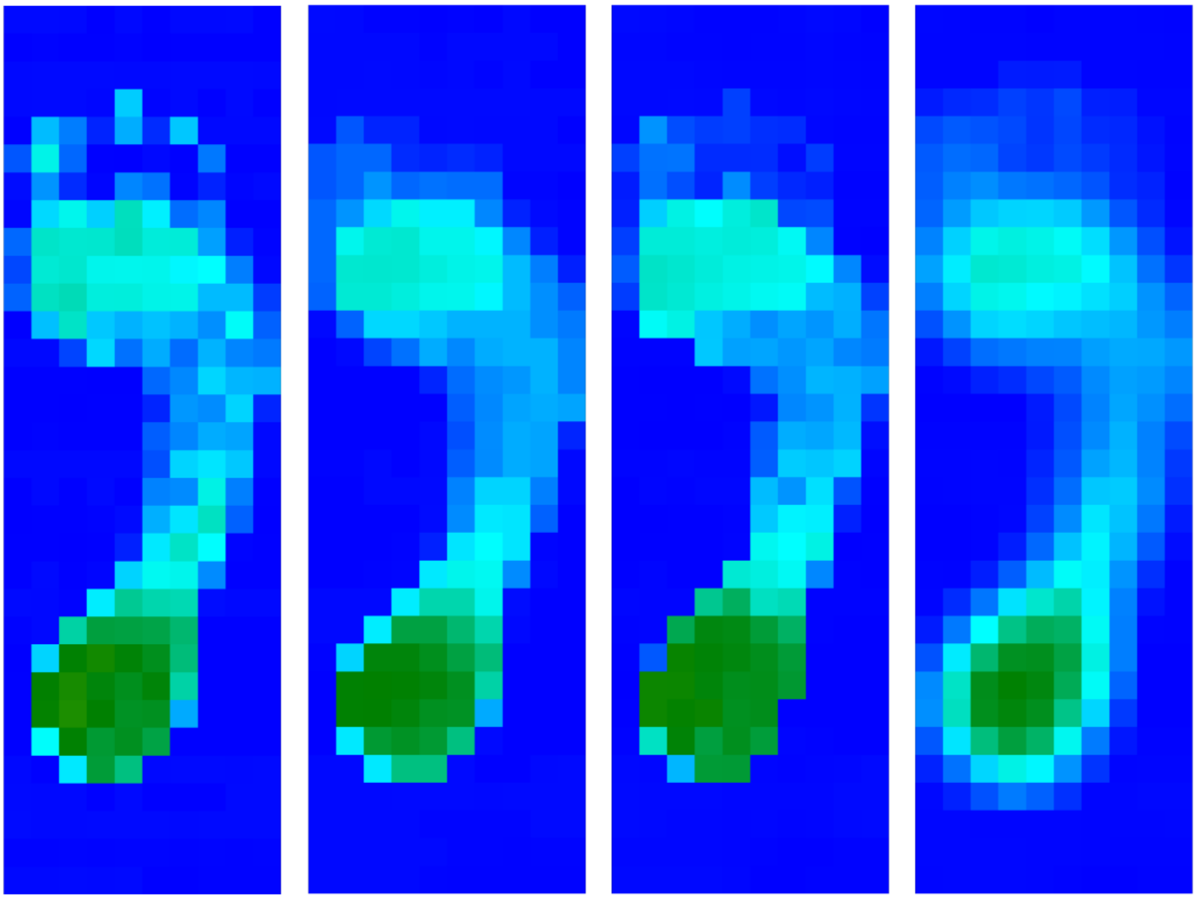


Figure 35: *Presentation of the filter methods on the example of a foot's pressure distribution. Left to right: original, median filter, SNN filter, mean filter.*

The *Filter type* dropdown menu provides three image filters:

- Median filter
- Statistical Nearest Neighbor (SNN) filter
- Mean (Average) filter.

The filter's window size can be set below. **Figure 35** shows a comparison of the different filter types for window size 3.

8.3.3 Image Interpolation

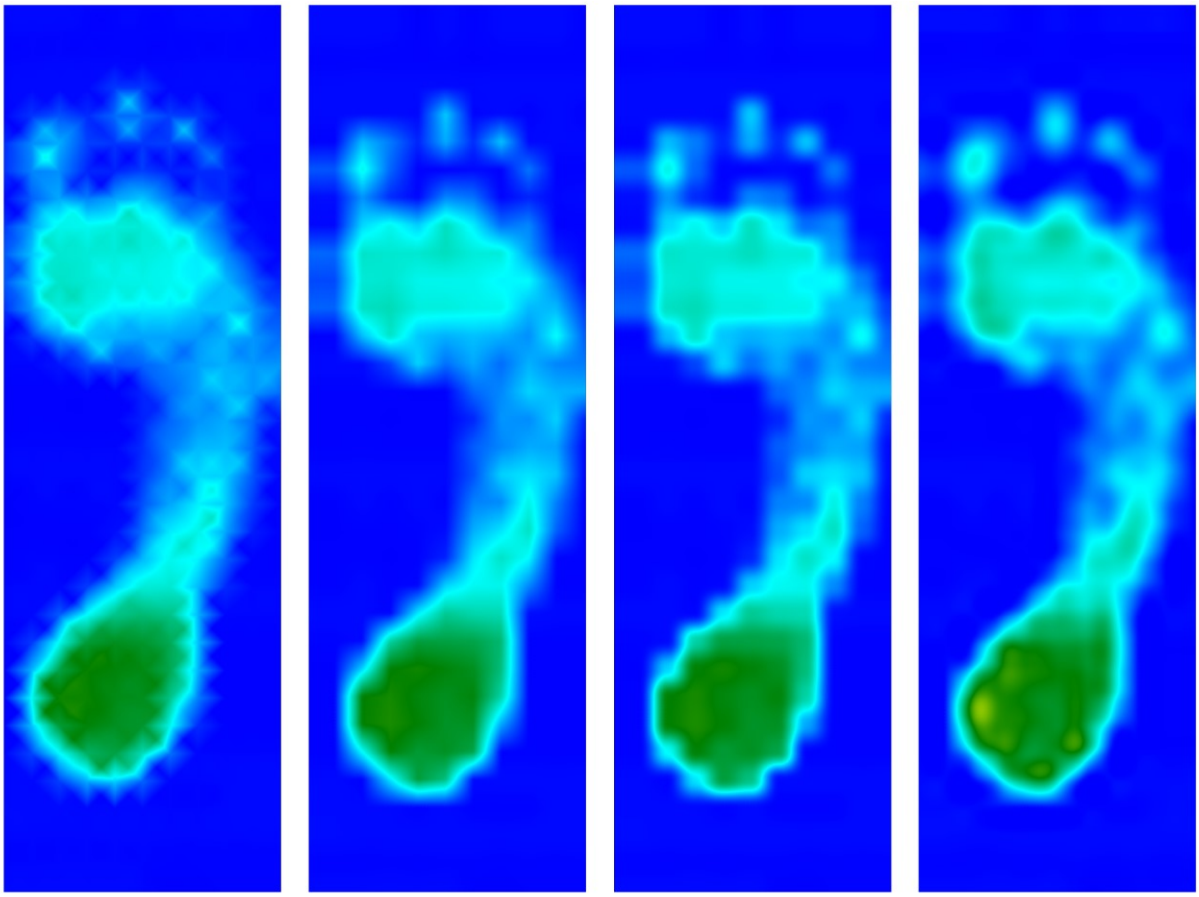


Figure 36: *Presentation of the three filter methods on the example of a foot's pressure distribution. Top to bottom: None, Bilinear, Cosine, Bicubic*

The *Interpolation mode* dropdown menu provides three interpolation types:

- Bilinear
- Cosine
- Bicubic

The number of interpolation segments can be set below. **Figure 36** shows a comparison of the different filter types for 8 segments.

8.4 Real-time Charts

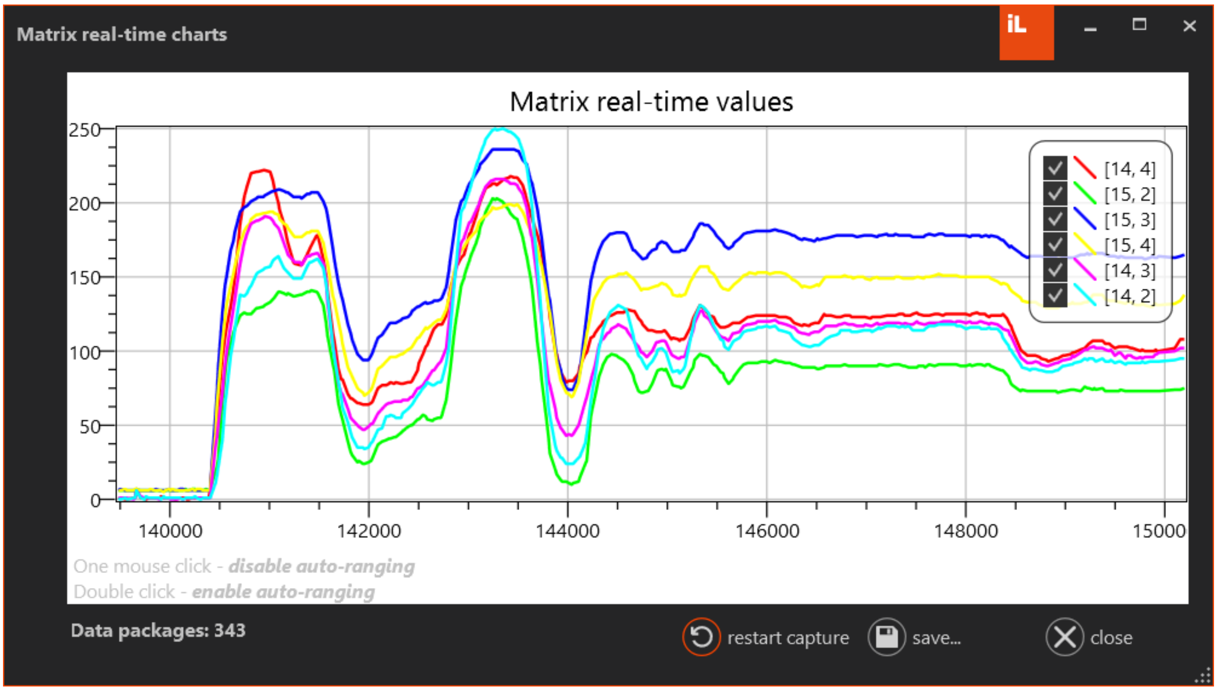


Figure 37: *Matrix real-time charts*

It is possible to visualize the measurement data for a specific selection of sensor cells in a line plot. Therefore, double click onto the pixel (sensor cell) in the view port. The real-time charts window opens (**Figure 37**).

NOTE: This function is only available in 2D mode.

Several pixels can be selected and displayed at the same time. Use the buttons to restart the plot or to save it to an image file.

9 Advanced Features

9.1 Time-Series Filtering

This section provides information about the device’s time filtering functionalities and how to configure them. Five filter algorithms are provided, which are described in the following subsections. The timestamp index of a measured sample x is notated with i. The filtered sample is notated with y.

NOTE: Time filtering is performed on-board. If enabled, the original data cannot be accessed after a measurement session. If it is required to keep the original measurement data, disable time filtering, and perform it manually.

The parameter details for the filter configuration are listed below.

Name	Notation	Unit	Type	Min. Value	Max. Value
Moving average filter window size	k_a	None	uint	1	100
Moving average cumulative filter coefficient	α	None	float	0.0096	1.0
Moving average weighted filter window size	k_{aw}	None	uint	1	100
Median filter window size	k_m	None	uint	1	100
Kalman filter measure speed	v	None	float	0.0096	1.0
Kalman filter measure error	r	None	float	0.0096	1.0

9.1.1 Moving Average Filter

For each sample x this filter outputs an unweighted average y_a of the current and previous $k - 1$ data points (if available). The k_a parameter represents the window size.

$$y_{a,i} = \begin{cases} \frac{1}{i}\sum_{j=0}^{i-1} x_{i-j} & i < k_a \\ \frac{1}{k_a}\sum_{j=0}^{k_a-1} x_{i-j} & i \geq k_a \end{cases}$$

9.1.2 Moving Average Cumulative Filter

An Exponentially Weighted Moving Average (EWMA) applies weighting factors to input data points that decrease exponentially. The coefficient α controls the effect of the filtering. For $\alpha = 1$ no filtering takes place.

$$y_{ac,i} = \alpha \sum_{j=0}^{i} (1-\alpha)^j \, x_{i-j}$$

9.1.3 Moving Average Weighted Filter

A weighted average is an average that has multiplication factors to give different weights to data at different positions in the sample window. The most recent sample is weighted the most. The parameter k_{aw} represents the window size.

$$y_{aw,i} = \begin{cases} \dfrac{1}{i(i+1)} \sum_{j=0}^{i-1} (i-j)\, x_{i-j} & i < k_{aw} \\ \dfrac{1}{k_{aw}(k_{aw}+1)} \sum_{j=0}^{k_{aw}-1} (k_{aw}-j)\, x_{i-j} & i \geq k_{aw} \end{cases}$$

9.1.4 Median Filter

The median filter returns the median of the last k_m samples, where k_m notates the window size. The median of a given set of values is the center value of the sorted set of those values. In the following, an example for $k_m = 5$ is given.

$$\begin{matrix} \dots \\ \end{matrix}\begin{pmatrix} x_{i-5} = 2765 \\ x_{i-4} = 3135 \\ x_{i-3} = 2912 \\ x_{i-2} = 3829 \\ x_{i-1} = 3430 \\ x_i = 3731 \end{pmatrix} \rightarrow \begin{pmatrix} x_{i-3} = 2912 \\ x_{i-4} = 3153 \\ x_{i-1} = 3430 \\ x_i = 3731 \\ x_{i-2} = 3829 \end{pmatrix} \rightarrow y_{m,i} = x_{i-1} = 3430$$

9.1.5 Kalman Filter

Implementation of a one-dimensional Kalman filter based on iterative computation of an estimate for measured values in a time series. K notates the Kalman gain, p the estimate error. The Kalman filtering can be configured using the measurement error r and the measurement speed v. The start conditions are $p_0 = 40$ and $y_0 = x_0$.

For each time stamp index i, three computation steps are required.

1. In the first step, the Kalman Gain K_i is computed.

$$K_i = \frac{p_{i-1}}{p_{i-1} + r}$$

2. Second, the filtered value (estimate) y_i is computed.

$$y_i = y_{i-1} + K_i(x_i - y_{i-1})$$

3. Finally, the error estimate for the next iteration p_i is computed.

$$p_i = (1 - K_n)\, p_{i-1} + v|y_i - y_{i-1}|$$

This process is repeated iteratively for each new data sample.

9.2 Calibration (Universal Matrix only)

The conversion from 12-bit integer measurement values to physical values is performed using calibration data. This conversion is split in two parts, an on-board integer to integer conversion and a software sided value mapping. The basic idea is to choose the reference values such that the software-sided value mapping can be performed using a linear function.

9.2.1 Introduction

In the following, for simplicity the process is explained for a single sensor cell. In general, this process is performed for each cell in the sensor matrix individually. Consider having a sorted set of n physical values P which correspond to a set of n original measurements M (12-bit integers). This kind of corresponding data usually arises from a calibration scenario. In addition, a set of n reference values R (12-bit integer) is defined, with their parameters still unknown.

$$M = [m_1, m_2, \dots, m_n]$$

$$P = [p_1, p_2, \dots, p_n]$$

$$p_1 < p_2 < \cdots < p_{n-1} < p_n$$

$$R = [r_1, r_2, \dots, r_n]$$

To compute a physical output y for any given measurement x, first an on-board conversion based on the reference values must be performed (f_1). Second, given the slope a and offset b of a linear function, value mapping is performed (f_2).

$$y = f_2(a, b, \hat{x}) = a \cdot \hat{x} + b$$

$$\hat{x} = f_1(R, M, x)$$

The on-board conversion is computed by linear interpolation of the measurement set M to the reference set R. For $m_i \leq x < m_{i+1}$ the computation of $\hat{x}$ derives as follows.

$$\hat{x} = r_i + (x - m_i)\frac{r_{i+1} - r_i}{m_{i+1} - m_i}$$

For a functioning value mapping, R, a and b must be chosen such that $p_i = f_2(r_i) = f_2\big(f_1(m_i)\big)$ for every i. In other words, the reference values must provide a linear correlation to p. The task is to define R such that this constraint is satisfied.

NOTE: Saved data (CSV or DB) does not contain physical values but converted 12-bit values. Use the parameters a and b to convert them to physical values manually.

9.2.2 Parameter Computation

1. First, the reference boundaries must be determined. Typically, the boundaries are set to $r_1 = 0$ and $r_n = 2^{12} - 1 = 4095$ to provide maximum resolution. However, as the on-board conversion provides also linear extrapolation, this range can be reduced. In this case measurements that lie outside of the calibration range will be linearly extrapolated instead of clipped to the boundaries.
   NOTE: The *arrange from physical units* button in the calibration settings editor performs the computation for $r_1 = 0$ and $r_n = 4095$ by default.
   NOTE: The reference values R are a 12-bit integer. Therefore, only integer values between 0 and 4095 are valid entries.

2. Second, the linear function parameters a and b are computed. Each pair (r_i, p_i) must satisfy the following condition.

   $$p_i = a \cdot r_i + b$$

   With r_1 and r_n already defined, the parameters a and b can be derived.

   $$\begin{pmatrix} p_1 \\ p_n \end{pmatrix} = a \begin{pmatrix} r_1 \\ r_n \end{pmatrix} + b$$

   $$a = \frac{p_1 - p_n}{r_1 - r_n}$$

   $$b = p_1 - a \cdot r_1 = p_1 - \frac{p_1 - p_n}{r_1 - r_n} \cdot r_1$$

   NOTE: In the calibration settings editor a refers to the *Coefficient* and b to the *Offset*.

3. Finally, the remaining reference values are computed directly from the physical values.

   $$r_i = \frac{p_i - b}{a}$$

10 List of Figures

11 Index