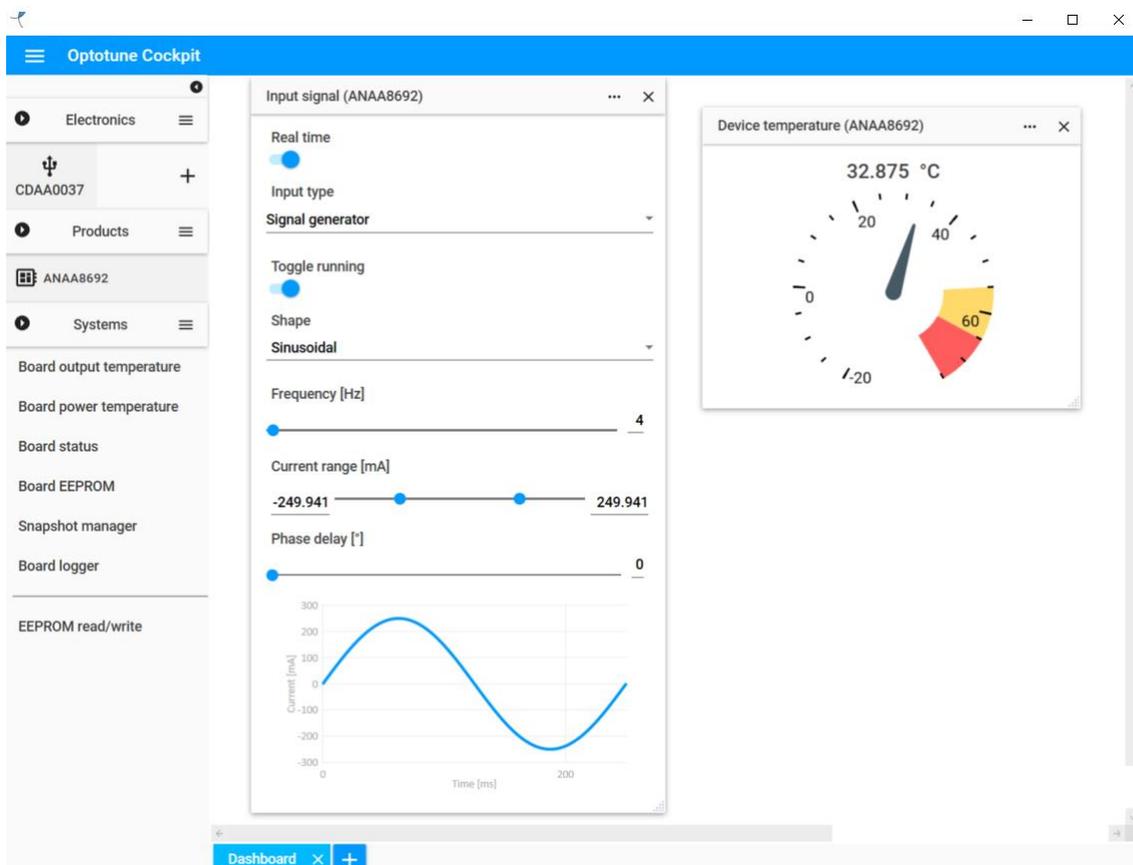


# Optotune Cockpit User Manual



Copyright ©2023 Optotune Switzerland AG. All Rights Reserved. Software Optotune Cockpit shall only be used in connection with the evaluation of the Optotune products such as liquid lenses, optical actuators and its compatible controllers. Any other use is not permitted without the prior written authorization of Optotune Switzerland AG.

IN NO EVENT SHALL OPTOTUNE BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS OPTOTUNE COCKPIT SOFTWARE, ITS DOCUMENTATION AND OPTOTUNE PRODUCTS, EVEN IF OPTOTUNE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

## Table of Contents

1. Overview .....	4
1.1 Software Installation .....	5
2. Optotune Cockpit introduction .....	5
2.1 Application main window .....	5
2.2 Flashing firmware and Device recovery .....	6
2.3 Dashboard .....	7
2.4 Device status .....	7
3. Operating ICC-4C controller with Optotune Cockpit .....	8
3.1 Board output temperature & Board power temperature .....	8
3.2 Board status .....	8
3.3 Board EEPROM .....	8
3.4 Snapshot manager .....	9
3.5 Board logger .....	9
3.6 Unknown device detected .....	10
3.7 Autodetect .....	10
4. Available systems for Focus Tunable Lenses connected via ICC-4C controller .....	11
4.1 Input Signal .....	11
4.1.1 Static input mode .....	11
4.1.2 Signal generator .....	13
4.1.3 Analog voltage input .....	13
4.1.4 Custom vector .....	14
4.1.5 External trigger signal .....	15
4.2 Device temperature .....	16
4.3 EEPROM read/write .....	16
5. Operating MR-E-2 and MR-E-3 controller with connected mirror head .....	16
5.1 Input signal .....	16
5.2 Input conditioning .....	17
5.3 Control mode .....	18
5.4 Linear output .....	18
5.5 Device status .....	19
5.6 EEPROM key-values .....	19
5.7 Mirror EEPROM read/write .....	19
5.8 Snapshot manager .....	20
5.9 Temperature sensor mirror .....	21
5.10 Temperature sensor proxy .....	21
6. Setups .....	21
6.1 Autofocus .....	23

## 1. Overview

Optotune Cockpit is an application (compatible with Windows 10 and 11) designed to control Optotune devices that use Optotune and Gardasoft controllers. With the application, users can connect to the controllers and devices, read device statuses, and perform firmware updates. Any connected device is referred to as "electronics" (e.g., ICC-4C or ECC-1C controller). Systems are displayed as widgets on the dashboard. The application supports multiple working dashboards, which can be customized by moving and resizing the widgets.

Supported controllers (electronics):

- Universal current controller ICC-4C
- Embedded current controller ECC-1C
- EL-E-4 and EL-E-4i
- Industrial lens controller TR-CL180 by Gardasoft
- Mirror controller MR-E-2 and MR-E-3

Supported devices (products):

- EL-16-40, EL-12-30, EL-10-30, EL-3-10 series
- XPR-9-2, XPR-20, XPR-33
- MR-E-2 and MR-E-3 Mirror head
- MR-15-30, MR-10-30
- FMR-20, BSW-20

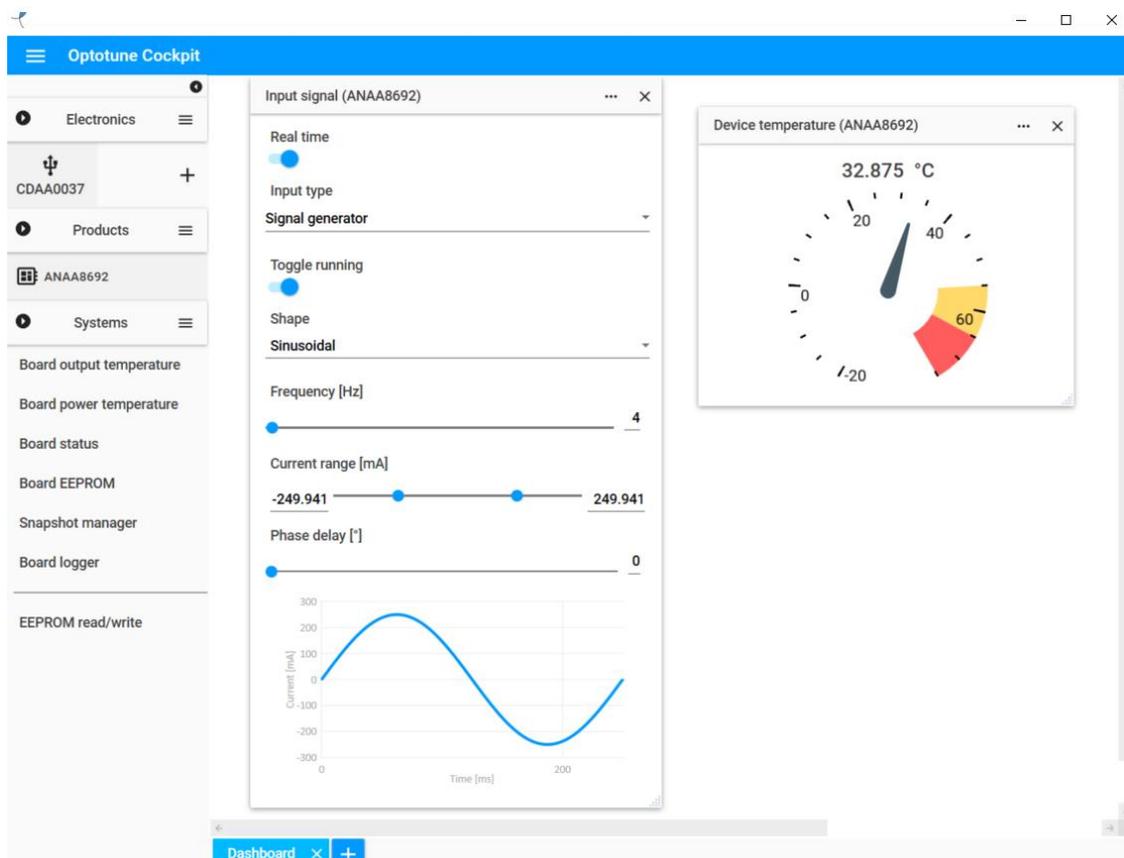


Figure 1: Optotune Cockpit with a connected ICC-4C controller driving an EL-16-40-TC lens

## 1.1 Software Installation

- Visit Optotune website: <https://www.optotune.com/registration-for-software-download>
- After registration, you will be forwarded to the software download page
- Download the Optotune Cockpit installation package
- Run Optotune cockpit installation
- Follow the installation wizard

## 2. Optotune Cockpit introduction

Launch Optotune Cockpit and click on "Connect Device." A new window will be displayed where the user can choose from a list of supported controllers connected via virtual COM port or Ethernet. More information about supported interfaces can be found in the controller's datasheets. Click "Connect" to establish the hardware connection.

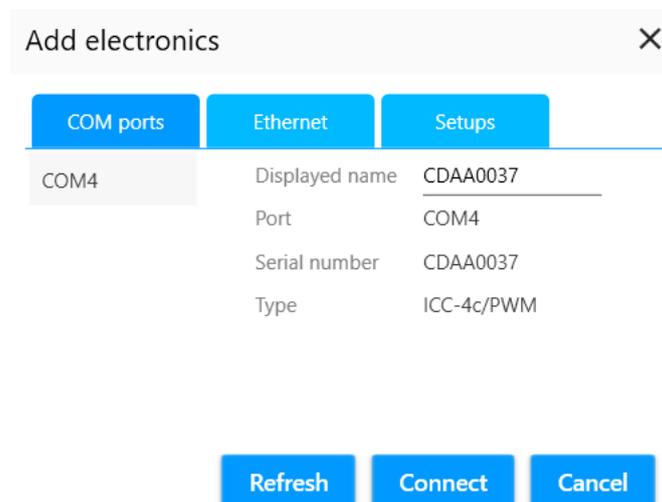


Figure 2: Add electronic Pop-up window.

### 2.1 Application main window

Figure 3 below shows the main control menu window. The software displays the serial numbers of the connected controllers and devices. If a product is not recognized or is unknown, no serial number is shown (e.g., if the connected device has no EEPROM or the EEPROM data is corrupted). In some cases, it's still possible to control such a device via Optotune Cockpit. After right-clicking on the unknown device, an extra menu allows the user to rename, disconnect (only available for controllers), and view the device properties.

The "System" section is split into two parts. The top part contains widgets related to the controller, and the bottom part is allocated for product-related widgets. In case several products are connected at the same time, selecting (left mouse click) the desired product is needed to display its systems. If a widget name does not appear anymore in the Optotune Cockpit main window menu, it means that it is already open on the software dashboard.

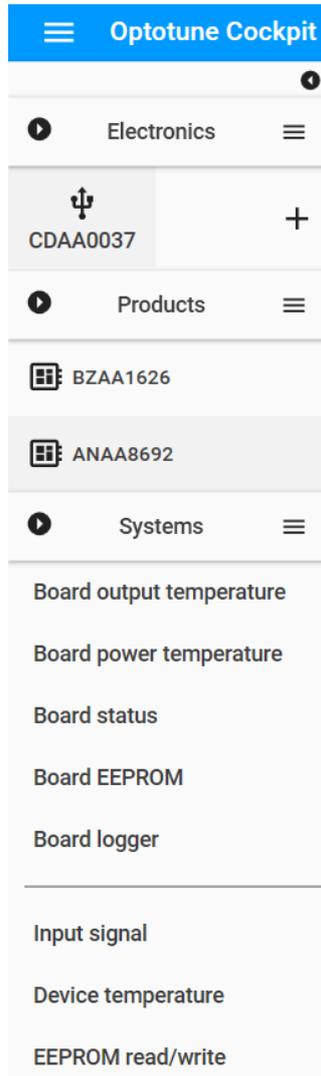


Figure 3: The main control menu of the Optotune Cockpit with one controller and two devices connected

## 2.2 Flashing firmware and Device recovery

The application allows the user to update the firmware of the selected device. To do so, open the drawer menu in the top left corner of Optotune Cockpit and choose the “Firmware Update” option. A new widget will be added to the current dashboard. On the widget, click on the folder icon to browse for the corresponding .hex file (the file can be downloaded after registration at <https://www.optotune.com/registration-for-software-download>) and then click on “Update”. **Please double check the version of Firmware after its update.**

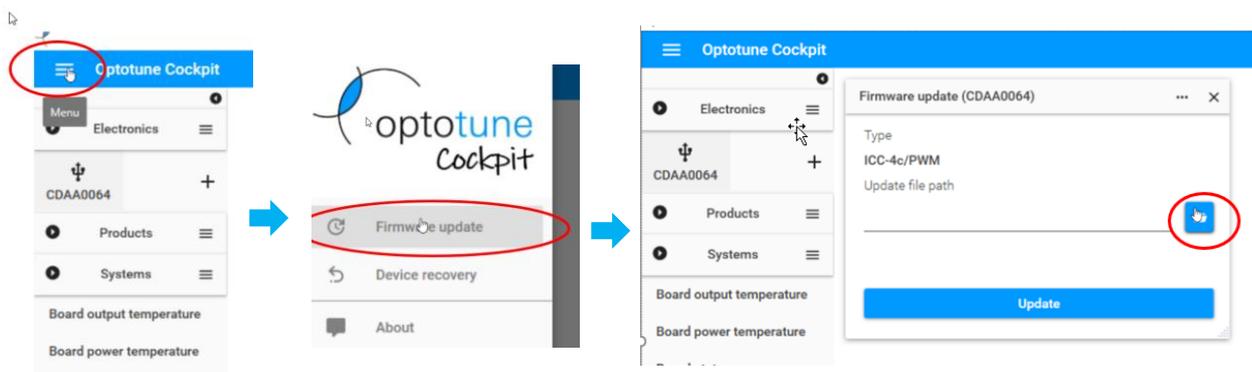


Figure 4: Firmware flashing procedure

If the device is stuck in DFU mode during the firmware update, you can use the recovery option. To use this feature, open the drawer menu and choose the “Device Recovery” option. A new widget will be added to the dashboard that shows all the connected devices in DFU mode. The list will be empty if no devices in DFU mode are found. Select the device to recover and choose “Recover.”

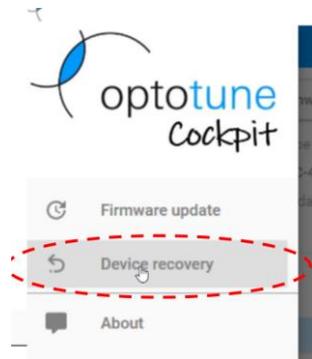


Figure 5: Device recovery

## 2.3 Dashboard

The software has a dashboard-oriented user interface. All systems are displayed in widgets that open on the dashboard. Users can have multiple widgets on the dashboard, but only one widget of the selected system can be open on the same dashboard. The same system can have another widget on other dashboards. Users can add a new dashboard by clicking the “+” button next to the dashboard names at the bottom of the software window. The selected dashboard is then highlighted at the bottom of the application workspace. Each dashboard can be renamed, removed, or duplicated. These actions are available from a menu that appears by right-clicking on the dashboard name. By duplicating the dashboard, the entire dashboard, including all widgets already open, is duplicated onto the new dashboard.

## 2.4 Device status

The software periodically scans the status of the connected devices. In case of a failure, the device reports this failure to the application, and an error icon appears next to the faulty electronic device. The icon color varies based on the severity of the problem (red or orange).

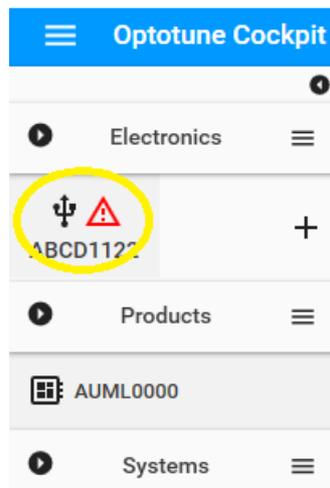


Figure 6: Error icon of connected controller

More information about the device is shown in the “Device Status” window by clicking the warning icon. The corresponding window will list the errors related to the device. Errors can be cleared by pressing the “Clear” button. However, if those errors persist on the device side, the error icon will remain visible.

### 3. Operating ICC-4C controller with Optotune Cockpit

Each Optotune controller supports a specific set of systems that can be accessed by the user. Systems provide a graphical interface to communicate with electronics. The ICC-4C controller provides the following systems for the user (the ECC-1C controller provides all the same systems, except for the Board Status widget).

#### 3.1 Board output temperature & Board power temperature

Two temperature reading widgets show the output temperature of a connected product and the board temperature. The user can set temperature thresholds for both temperature readings to protect the controller against overheating. This setting is accessible by clicking on the "Meatball menu" icon in the top right corner of the widget. By default, 80 °C is preconfigured in the controller firmware for both temperature readings.

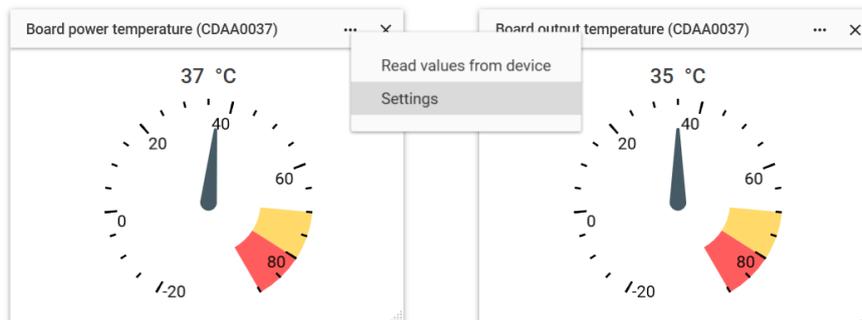


Figure 7: Board output and power temperature

#### 3.2 Board status

The Board Status widget offers three different reading values: input voltage, input current, and frontend voltage. It also shows the MCU load as a percentage. The frontend voltage can be manually adjusted within a range of 7V to 20V.

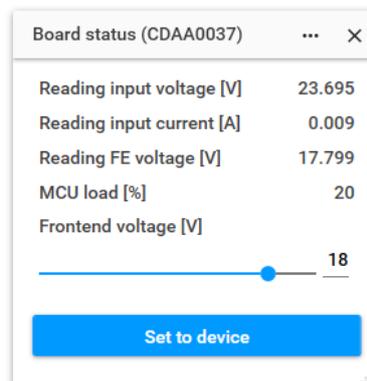


Figure 8: Board status readings

#### 3.3 Board EEPROM

With the board EEPROM control system, the Ethernet connection settings of the controller can be configured. IPv4 or IPv6 values can be set manually, or the DHCP option can be chosen to automatically assign the IP address and other communication parameters to the device from the DHCP server.

To update the device with the desired settings, the user must click the "Set to Device" button. If the Ethernet settings were changed manually by the user, the device must be restarted for those changes to take effect in the controller.

It's also possible to enable FE autostart and edit FE voltage, PWM frequency, and Device Detect Mask.

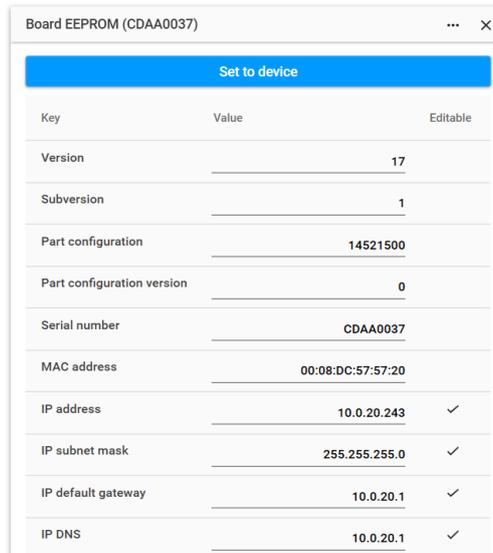


Figure 9: Ethernet settings in Board EEPROM system widget

### 3.4 Snapshot manager

The software provides functionality to control the snapshot manager present in the device firmware. This functionality is provided by the “Snapshot Manager” widget, allowing users to retrieve settings previously stored in the firmware even after power cycling the device. These snapshots are identified by numbers, and each snapshot can be loaded onto the device. Snapshot number one is “Factory Settings,” which allows the user to reload the factory settings on the device (this snapshot cannot be edited or overwritten). Users can store the current configuration into any of the other available snapshot numbers. It is also possible to choose which snapshot should be loaded during device start-up. **For example, a user-defined snapshot can be saved as number 2 by pressing "Save," then "Load" (to check for any failures), and then using "Set the Startup Snapshot" as number 2.** This option saves the snapshots into non-volatile memory, so the snapshots will not be lost after powering off the device.



Figure 10: Snapshot manager

### 3.5 Board logger

The embedded board logger allows the user to log up to 5000 points and six different values at a time. The sampling rate ranges from 20000 Hz to 0.001 Hz. It’s possible to monitor the current, focal power, or resistance for each channel.

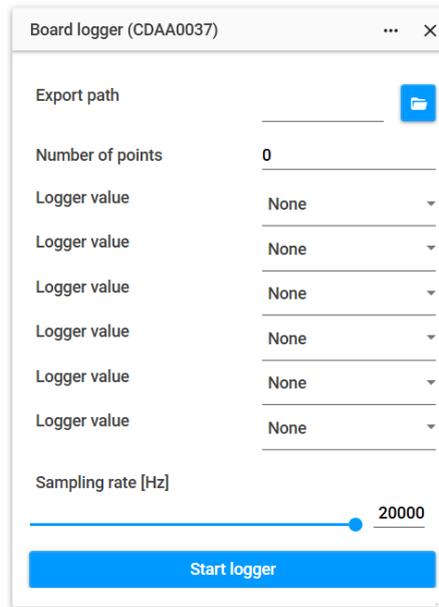


Figure 11: Board logger widget windows

### 3.6 Unknown device detected

Some special development applications may involve the use of non-conventional products (e.g., LED light sources) that are unknown to the controller. In this case, in the “Board Status” section, the appropriate channel must be activated. Refresh the products list to show the unknown device. Afterwards, in the “Input Signal” widget, the user will be asked to configure the device, which can be chosen from the presets (EL-3-10, XPR-9) or set as a custom setting – Unknown Device.

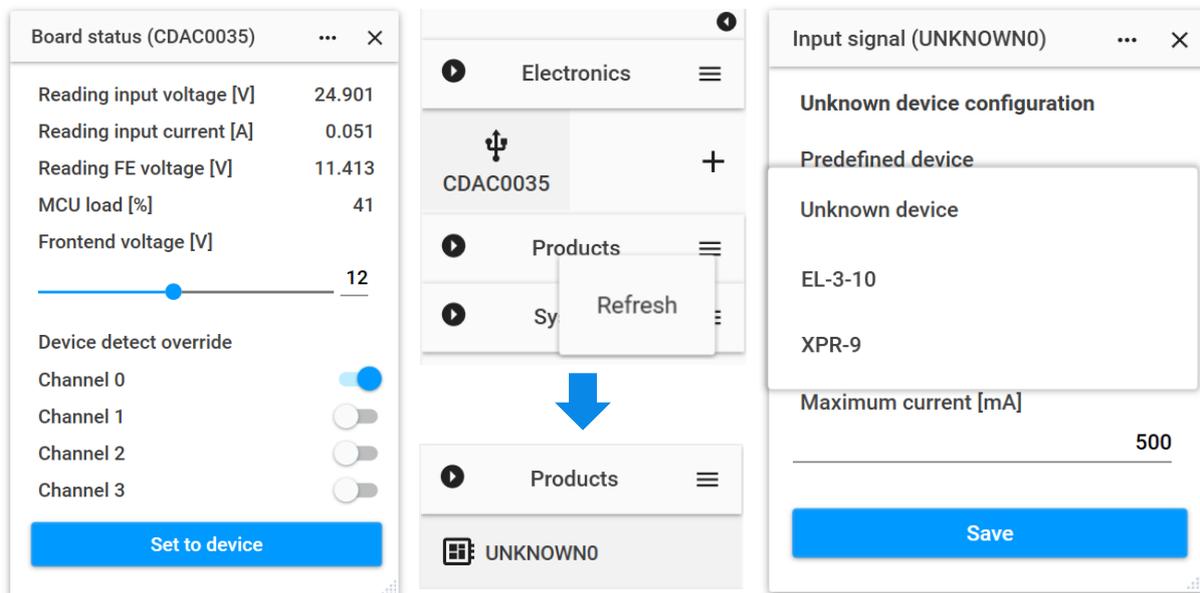


Figure 12: Unknown device settings

### 3.7 Autodetect

This widget allows enabling or disabling the autodetect device process on all available channels. By default, all channels are enabled. **All unoccupied channels should have autodetection disabled to achieve the best performance of the driver. When saving a snapshot using the "Snapshot Manager" widget, the occupied channels must have autodetection enabled.**

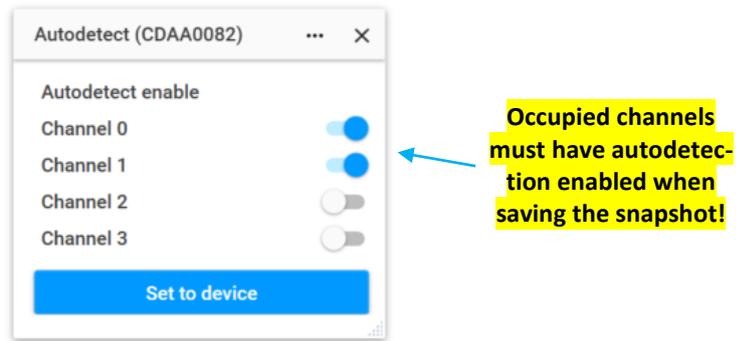


Figure 13: Autodetect widget with enabled autodetection on channels 0 and 1

## 4. Available systems for Focus Tunable Lenses connected via ICC-4C controller

### 4.1 Input Signal

Controlling a connected focus tunable lens is possible via the “Input Signal” widget. The widget offers four different operational modes, and the device can be controlled in real-time or manually in all four modes. To enable real-time operation, click on the toggle switch under “Real Time.” Any changes to the control parameters will be applied automatically (by default, real-time operation is disabled). To manually operate the lens, first set the desired parameters and then press the “Set to Device” button.

Via the “Meatball menu” (top right corner of the widget), it’s possible to configure the minimum and maximum current or optical power available for the connected lens. The default values are pre-set based on the connected lens version, and it’s safe to operate the given lens within that range. **Incorrectly set limits can potentially damage the connected device! If in doubt, please consult with Optotune.** In case the connected lens is not recognized by the controller, the current limits need to be set before operating the lens. After manually inserting the values, click “Save.”

#### 4.1.1 Static input mode

This feature allows the user to set a certain “Current” or “Focal Power” (if applicable for the given lens version) to the connected device. First, select “Current” or “Focal Power” from the “Unit Type” drop-down menu and then set a value. The value can be changed by dragging the slider or manually overwriting the shown value on the right side of the slider. The minimum step size can be achieved by clicking on the plus and minus buttons next to the slider.

Optotune lenses are current-controlled. By using focal power mode, temperature-independent lens operation is ensured. Depending on the current temperature, the controller automatically adjusts the current applied to the lens to compensate for the temperature drift of the lens. A look-up table with the calibration data for temperature compensation is stored directly on the EEPROM of each individual lens. With temperature compensation enabled, the absolute repeatability achieved over an operating temperature range of 10 to 50 °C amounts to typically 0.05 diopters or 0.1 diopters for the EL-16-40-TC and EL-10-30-TC series, respectively.

**Achievable Range of Focal Power:** When the lens heats up, the optical fluid expands in volume, causing the focal power to increase. The focal power increases linearly by approximately 0.2 to 0.7 dioptres per 10°C temperature increase, depending on the lens model. The color bar under the slider identifies the values of the focal power that are **Guaranteed**, **Achievable** and **Not achievable** with the current temperature limits. The operational temperature limits can be adjusted via the settings button on right bottom side. By default, the range is set to 20 - 45 °C.



Figure 14: Real time mode disabled with focal power achievable range

The limits for the different ranges of focal power are determined by the temperature limits and the maximum and minimum current. The maximum focal power value limit is set at the highest encoded temperature limit and maximum current, while the minimum focal power limit is set at the lowest encoded temperature and minimum current. This is explained in Figure 15. The highest and lowest temperatures define a sector for the linear relationship between focal power and current, indicated by the light and dark grey lines. The coloured line represents the focal power range at the current temperature between the highest and lowest temperatures, with the colours representing the following states:

- **Green:** Focal power can successfully be maintained at predefined temperatures limits in this range.
- **Yellow:** Focal power can be maintained at the current temperature and/or current drift state.
- **Red:** Focal power cannot be maintained at the current temperature and/or drift state.

The minimum green/yellow limit is determined by the focal power value at the highest temperature and minimum current. The maximum green/yellow limit is determined by the focal power value at the lowest temperature and maximum current. Both limits are indicated by the dashed horizontal lines in Figure 15. The closer the minimum and maximum temperature limits are, the larger the guaranteed range (green) will be. Additionally, the yellow/red limits are determined by the focal power values at the minimum and maximum current at the current temperature.

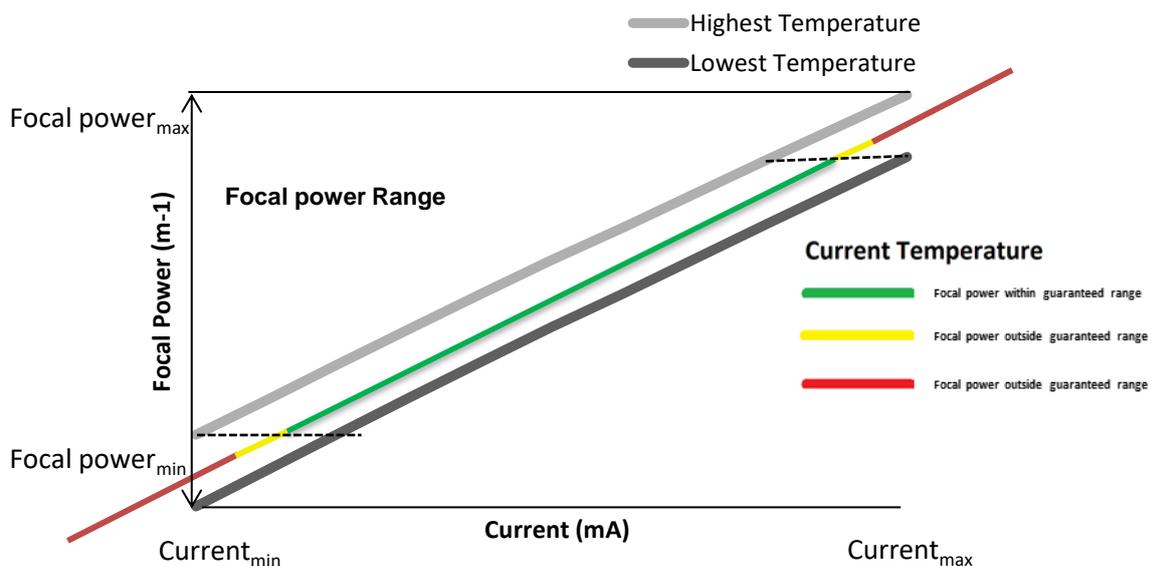


Figure 15: Illustration of the achievable range of focal power (vertical axis) versus current (horizontal axis). The guaranteed range (green line) also depends on the highest and lowest temperature, indicated by the grey lines.

### 4.1.2 Signal generator

Several driving waveforms can be chosen from the “Shape” dropdown menu. The user first selects current or focal power (if applicable) from the “Unit Type” dropdown menu. For each waveform type, the amplitude, frequency, and other parameters (depending on the shape) can be adjusted. Values can be changed by dragging the slider or inserted manually. The “Toggle Running” button allows the waveform to be run or paused.

Available waveforms: sinusoidal, triangular, square, sawtooth, pulse and staircase.

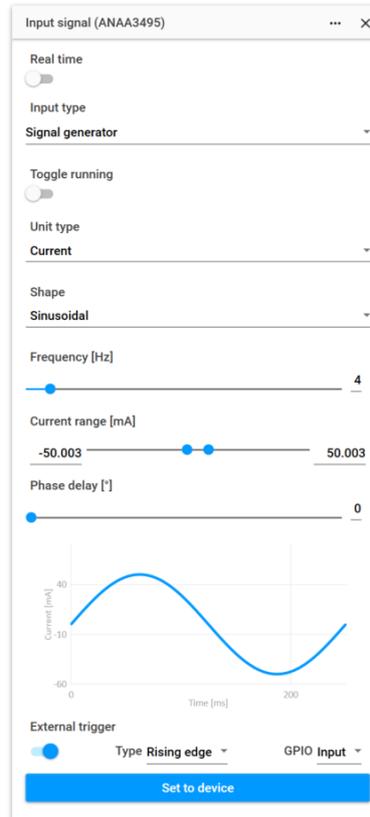


Figure 16: Sinusoidal waveform signal generator

### 4.1.3 Analog voltage input

The analog voltage input (0-10 V, one per channel) can be mapped to the driving current or the focal power (if applicable) of the connected lens on the corresponding channel. Both linear and non-linear mapping are possible. The applied analog voltage can be read by pressing the refresh button. With linear interpolation, the controller linearly interpolates the analog voltage input range (0-10 V) to the corresponding pre-set current or focal power range. With non-linear analog voltage transition, mapping points can be added manually by the user. For each point, the user sets the input voltage and the corresponding output current or focal power. A toggle button is available to enable constant or linear extrapolation.

**Analog voltage input mode with the ECC-1C controller is often used with the Snapshot Manager. The setting is saved to non-volatile memory, so the snapshot with mapping points will not be lost after powering off the device. Please check section 3.4 Snapshot Manager for more information.**

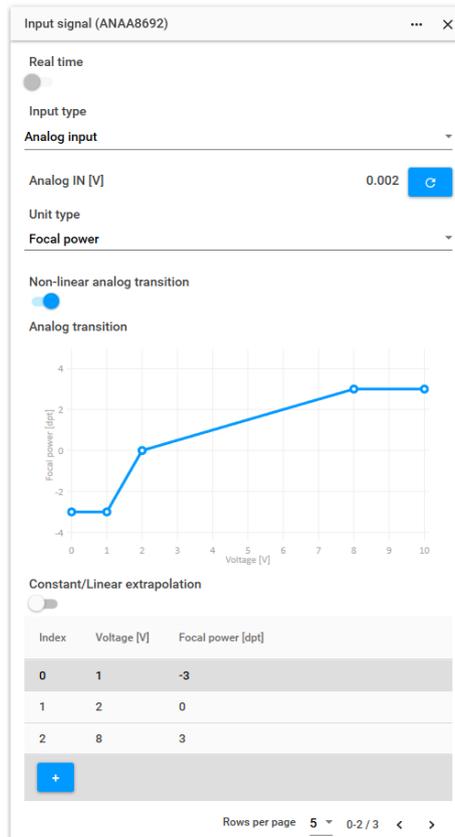


Figure 17: Analog voltage input with Non-linear mapping of current values

#### 4.1.4 Custom vector

The Custom Vector subsystem allows the user to pre-set a certain number of customized input current and dwell time values and then cycle through them.

The system offers two modes: Easy and Pro mode. In Easy mode, the user sets the static current or focal power (if applicable) value and dwell time for each point. By right-clicking on any given row in the table, the user can delete, add, or insert values. Each value can be manually rewritten. As with other input types, the widget offers both real-time operation and manual value setting for the device. The header menu allows the configuration of minimum and maximum current or optical power limits. It also contains the function to read the current values that were pre-set in the device. Pro mode allows the import of current values from an external CSV file, enabling the import of multiple current stages. Once the values are imported, the user can choose the sampling rate.

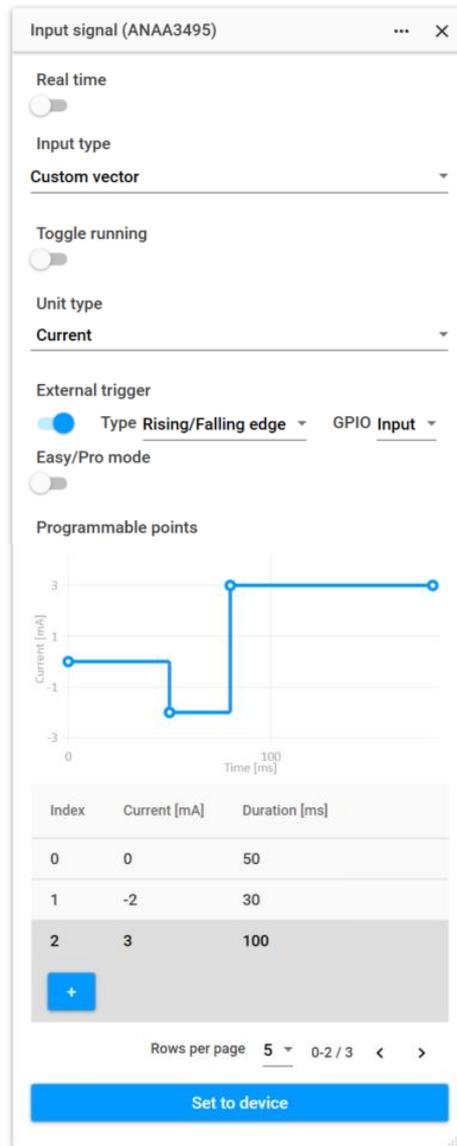


Figure 18: Custom vector widget in Easy mode

#### 4.1.5 External trigger signal

The Signal Generator and Custom Vector input types allow the use of a trigger signal for synchronizing the lens with other parts of the system. The user has different options for whether the trigger starts on the Rising edge or the Rising/Falling edge using a GPIO pin as an input. Additionally, the controller can generate the trigger signal as an output on the GPIO pin.



Figure 19: External trigger signal settings

## 4.2 Device temperature

The software allows monitoring the temperature of the connected focus tunable lenses. The user can set temperature thresholds to prevent damaging the lens. This setting is accessible by clicking on the “Meatball menu” icon in the top right corner of the widget. By default, 80 °C is preconfigured in the controller firmware.

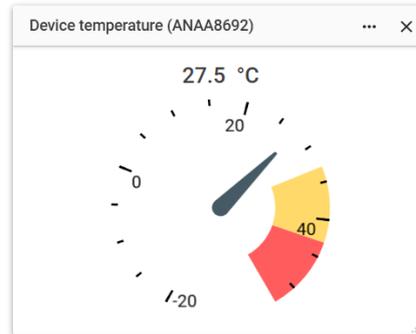


Figure 20: Lens temperature reading

## 4.3 EEPROM read/write

The EEPROM widget displays the data stored in the lens EEPROM. It shows two plot series with calibration points measured at 30 °C and 45 °C. In case the EEPROM of a connected lens is corrupted, Optotune can provide the original calibration data for the given lens. Via the settings in the “Meatball menu,” the user can import EEPROM data from a CSV file, export EEPROM data to a CSV file in byte form or export it in a human-readable format. After importing a new EEPROM, it’s necessary to click on the “Write” button to flash the new EEPROM data to the connected lens.

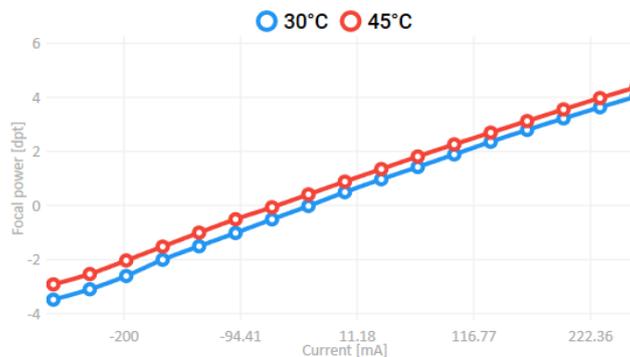


Figure 21: Lens calibration data

## 5. Operating MR-E-2 and MR-E-3 controller with connected mirror head

Optotune Cockpit provides the following systems for MR-E-2 and MR-E-3. Systems can be divided into signal flow systems, which control the signal flow stages of the device, and other systems, which mostly provide status information about the device such as temperatures, EEPROM values, etc.

### 5.1 Input signal

The application provides a widget for the input signal system. This widget contains a graphical interface for setting the values of this system. Values are set by clicking the “Set to Device” button at the bottom of the widget. The widget supports real-time input functionality, allowing the user to change values in the device in real-time by using the sliders or by entering values into the fields and pressing Enter or losing focus on that field. The widget restricts the user from entering invalid values. The view of the widget changes based on the system selected for each axis (X and Y). There are four options for each axis: “Static Value,” “Signal Generator,” “Arbitrary Vector,” and “Analog Input.” By selecting one of these options, the widget layout changes to correspond to the selected input signal system. When the Arbitrary Vector input signal is used, the widget allows the

user to change the vector values by clicking the “Edit Values” button. Values are edited for both axes simultaneously. The widget also provides a menu with the option to read values from the device. This menu is opened by clicking on the three dots icon in the header of the widget. More options are added to this menu while editing the vector values, such as importing and exporting values from a CSV file.

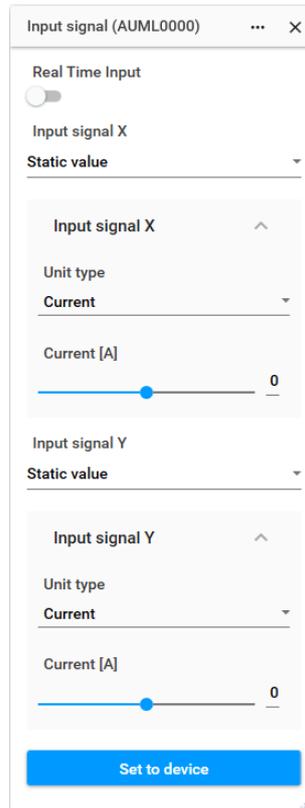


Figure 22: Input signal widget (Static value both axis)

## 5.2 Input conditioning

The Input Condition widget allows the user to set values for scale and offset. These values are then applied to the signal generated in the Input Signal widget. This widget also contains real-time input functionality and menu options to read values from the device.

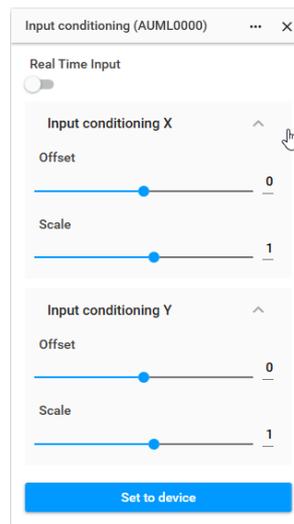


Figure 23: Input conditioning widget

### 5.3 Control mode

The Control Mode widget allows the user to change the control mode for both axes. This widget also provides a menu option to read values from the device. When the “Closed Loop” option is chosen, the widget allows the user to change the values of PID gains. It is not recommended to change these values.

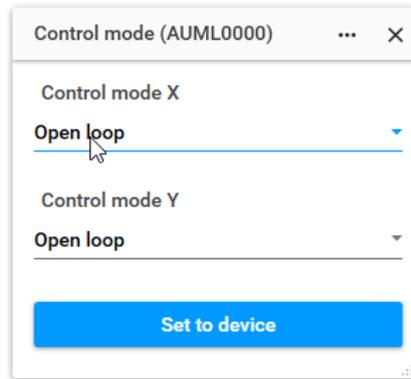


Figure 24: Control mode widget

### 5.4 Linear output

The Linear Output widget provides a graphical interface for the linear output system. It allows the user to change the values of the last stage of the signal flow, such as the current limit. This widget supports real-time input functionality and contains a menu option to read values from the device. Values for each axis are changed separately.

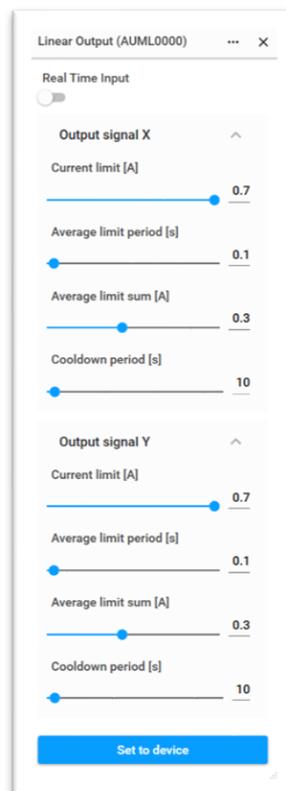


Figure 25: Linear output widget

## 5.5 Device status

The application provides a widget to show the actual status of the device. The widget displays information about the temperature of the connected mirror, the output stage current for both axes, and the actual X and Y values. It also contains a plot that shows the actual position of the mirror (the point where the mirror is pointing). It is only possible to move the mirror within the red bounds of the plot.

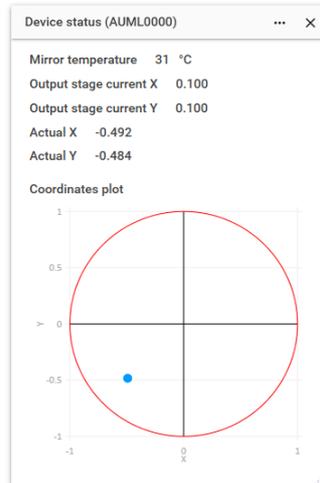
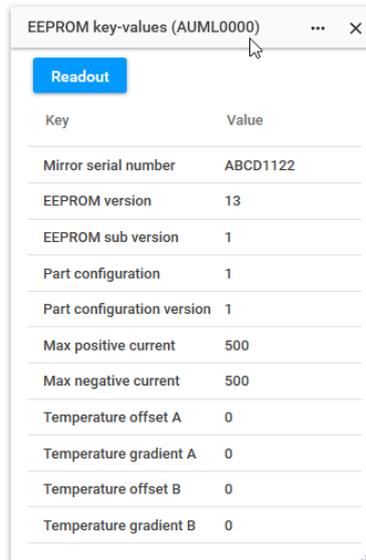


Figure 26: Device status widget

## 5.6 EEPROM key-values

The application provides a widget to display key values stored in the mirror EEPROM. These values can be read from the device by clicking the "Readout" button. They can also be exported or imported using the menu in the header of the widget.



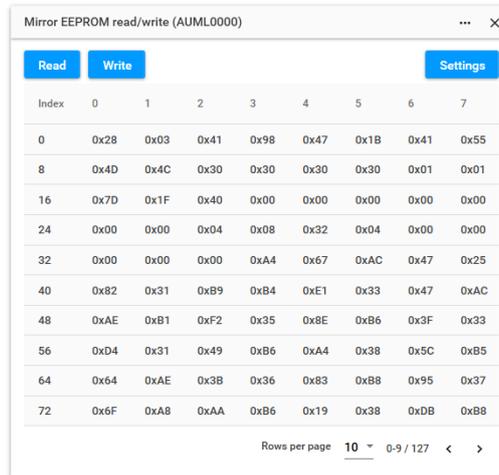
Key	Value
Mirror serial number	ABCD1122
EEPROM version	13
EEPROM sub version	1
Part configuration	1
Part configuration version	1
Max positive current	500
Max negative current	500
Temperature offset A	0
Temperature gradient A	0
Temperature offset B	0
Temperature gradient B	0

Figure 27: EEPROM key-values widget

## 5.7 Mirror EEPROM read/write

The application provides an option to view and change the EEPROM data stored in the mirror. This data is read from the device and shown in a table. The data can then be edited and written back into the device. The mirror EEPROM needs to have valid calculated CRC values in the correct positions to work properly. After clicking the "Write" button, all the values in the table are written into the mirror EEPROM, and the CRC values are calculat-

ed and written into the EEPROM. Data can be exported to or imported from a CSV file. The user can also edit the number of rows to be read from the device and the start index by clicking the "Settings" button. Additionally, the user can change the number of rows per page in the table. By right-clicking on the table, the user can add, delete, or insert a new row into the table. Values are written into the device sequentially and according to the indices shown in the table.

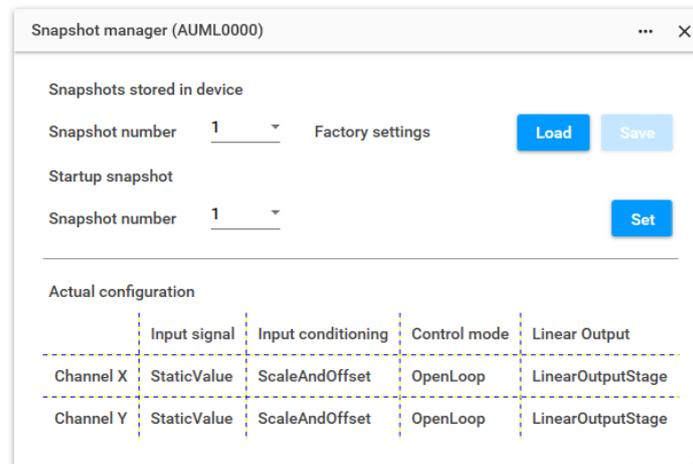


Index	0	1	2	3	4	5	6	7
0	0x28	0x03	0x41	0x98	0x47	0x1B	0x41	0x55
8	0x4D	0x4C	0x30	0x30	0x30	0x30	0x01	0x01
16	0x7D	0x1F	0x40	0x00	0x00	0x00	0x00	0x00
24	0x00	0x00	0x04	0x08	0x32	0x04	0x00	0x00
32	0x00	0x00	0x00	0xA4	0x67	0xAC	0x47	0x25
40	0x82	0x31	0xB9	0xB4	0xE1	0x33	0x47	0xAC
48	0xAE	0xB1	0xF2	0x35	0x8E	0xB6	0x3F	0x33
56	0xD4	0x31	0x49	0xB6	0xA4	0x38	0x5C	0xB5
64	0x64	0xAE	0x3B	0x36	0x83	0xB8	0x95	0x37
72	0x6F	0xA8	0xAA	0xB6	0x19	0x38	0xDB	0xB8

Figure 28: Mirror EEPROM read/write widget

## 5.8 Snapshot manager

The application provides functionality to control the snapshot manager present in the device firmware. This functionality is provided by a widget that allows the user to load snapshots stored in the firmware. These snapshots are identified by numbers, and each snapshot can be loaded into the device. Snapshot number one is the "Factory Settings" snapshot, which allows the user to load factory settings into the device. This snapshot cannot be erased, and therefore, the user cannot store their own snapshots in snapshot number one. The user can store the current configuration into any other available snapshot number. The user can also choose which snapshot should be loaded at start-up of the device. Snapshots are stored in temporary memory; therefore, the user needs to choose the "Persistence Save" option from the widget header menu. This option saves the snapshots into persistent memory so they will not be lost after powering off the device. The current configuration can be seen at the bottom of this widget. The user can also export the current configuration into a file, which can be imported into the device later. This way, the user can store multiple configurations on their computer and use the application to load these configurations. The widget also provides an option to read values from the device in the widget header menu.



Snapshots stored in device

Snapshot number  Factory settings

Startup snapshot

Snapshot number

---

Actual configuration

	Input signal	Input conditioning	Control mode	Linear Output
Channel X	StaticValue	ScaleAndOffset	OpenLoop	LinearOutputStage
Channel Y	StaticValue	ScaleAndOffset	OpenLoop	LinearOutputStage

Figure 29: Snapshot manager widget

## 5.9 Temperature sensor mirror

The application provides an option to monitor the temperature of the connected mirror. The widget for monitoring the temperature shows the actual temperature of the mirror in °C and the temperature threshold. After reaching this threshold, the device stops and starts cooling itself. Once the temperature drops below the threshold minus 10 °C, it starts working again. This threshold can be changed using the slider or input field by clicking the "Set to Device" button.

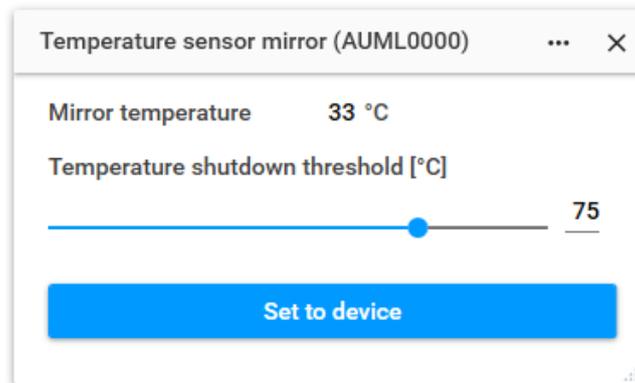


Figure 30: Temperature sensor mirror widget

## 5.10 Temperature sensor proxy

The device also supports monitoring the proxy board temperature. The widget for monitoring this temperature shows the temperature in °C and the temperature threshold, which works the same way as the temperature threshold for the mirror.

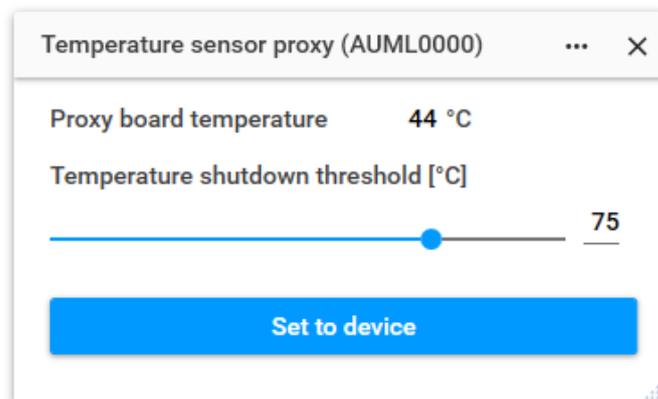


Figure 31: Temperature sensor proxy widget

## 6. Setups

Optotune Cockpit provides solutions for controlling setups consisting of GenICam compliant cameras and Optotune products. All these devices must be connected to a PC and detected correctly by the Cockpit software. In the "Add Electronics" window under "Setups," find the camera with associated Optotune devices and press "Connect." First, configure the camera settings under "Camera Properties." The camera's "Live View" is also available.

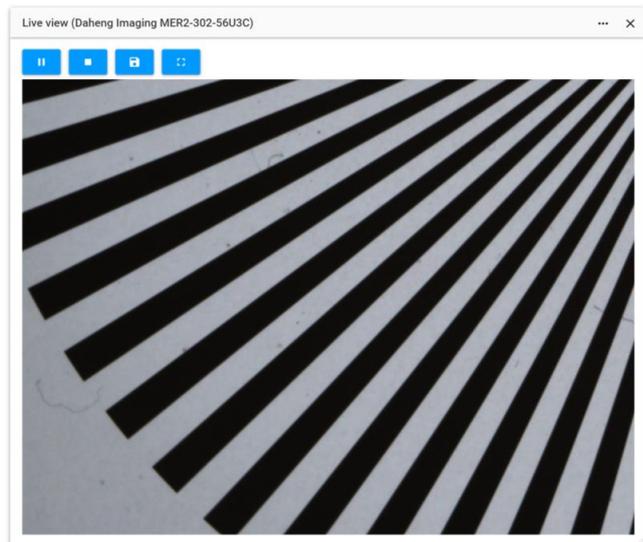
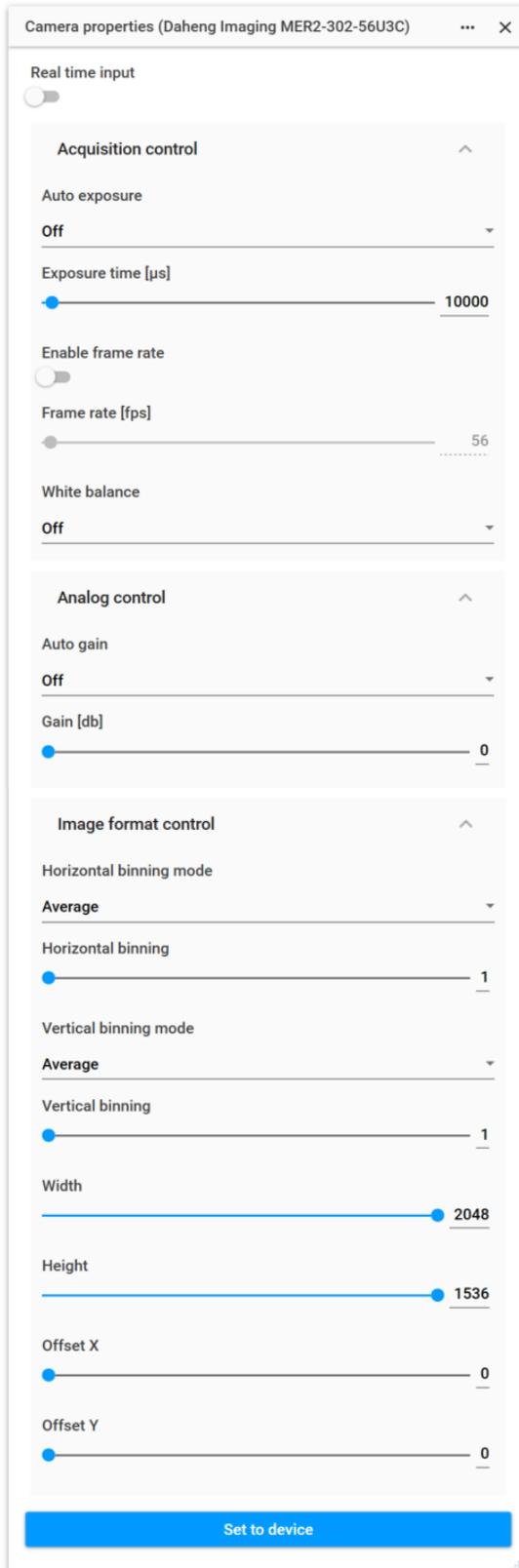


Figure 32: Camera properties and Live view widgets

## 6.1 Autofocus

The camera lens setup provides an automated focus algorithm in the “Autofocus” module as follows: The camera takes several images of a small area of interest at different diopter values. The diopter increment step depends on the set coarse sweep step. The algorithm then calculates the contrast of the taken images. A small interval in the vicinity of the maximum contrast value is then swept with a fine sweep step. Again, the algorithm calculates the contrast of the taken images. It subsequently fits a Lorentzian curve to the contrast values versus the optical powers. The sharpest image corresponds to the peak of the Lorentzian curve and defines the required diopter/current setting.

Autofocus Flowchart for Coarse sweep step 0.4 dpt and Fine sweep step 0.04 dpt within diopter range <-2, 3> is shown below.

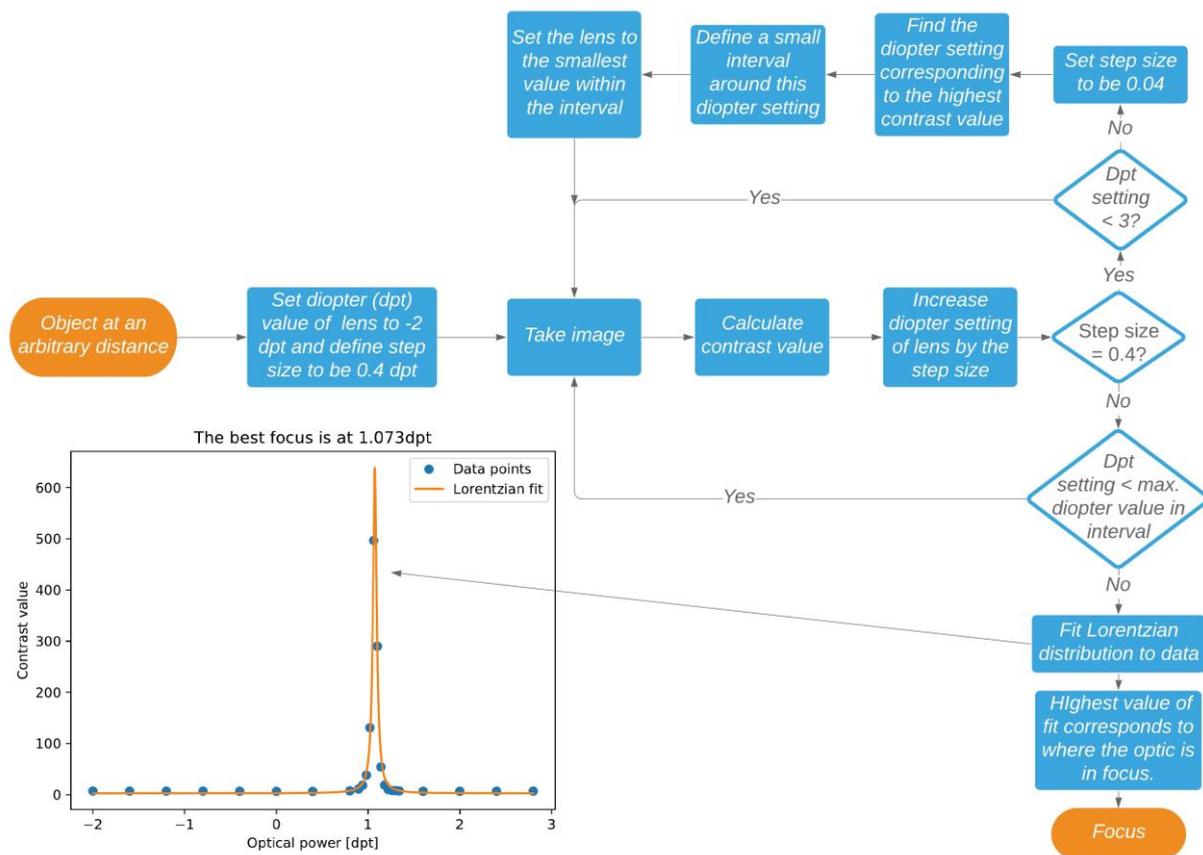


Figure 33: Program flow of the autofocus algorithm

Users can choose between Simple and Pro modes of Autofocus. These modes offer parameter settings (such as Unit Type, Coarse and Fine sweep step, Range, etc.) and display the Lorentzian fit. The concept of this feature is based on simply clicking and choosing an area of interest that should be in focus.

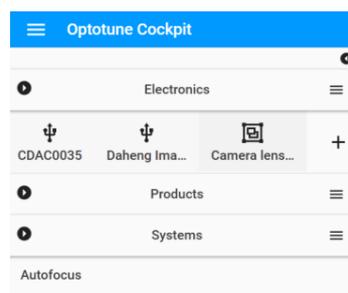


Figure 34: Camera and lens setup in Cockpit software

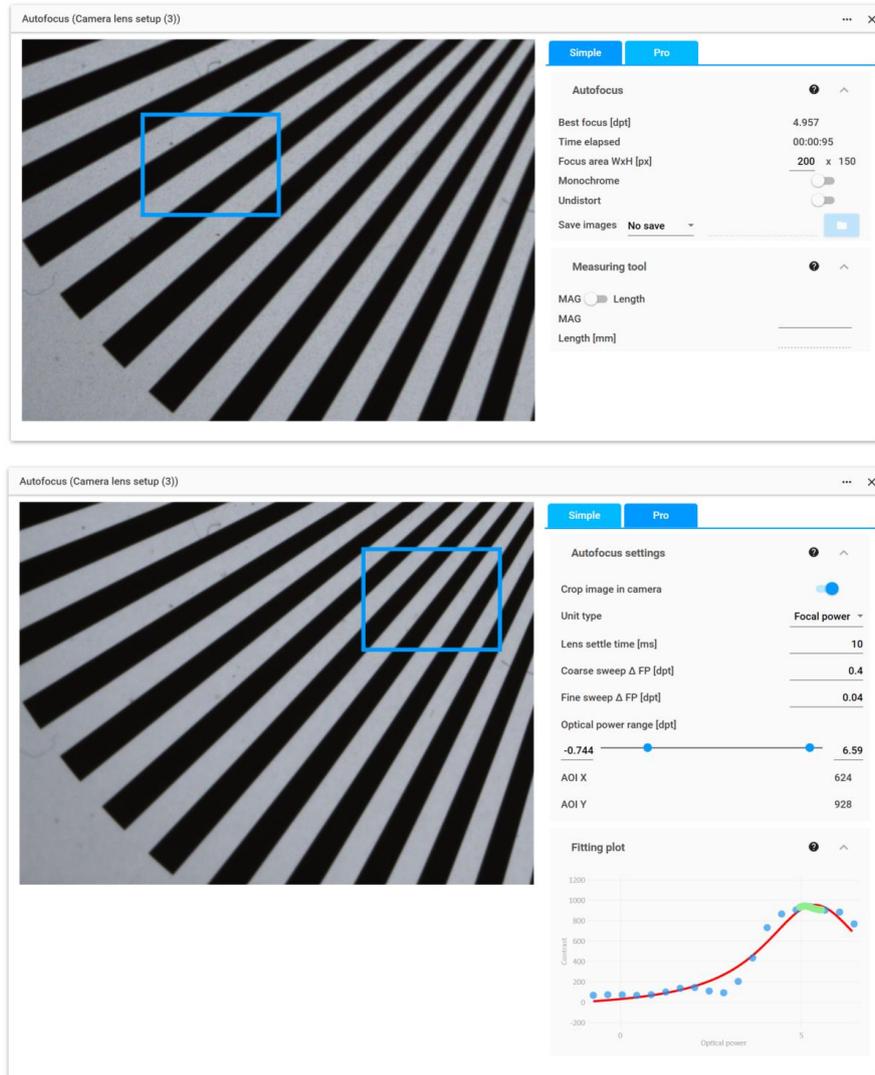


Figure 35: Autofocus module, Simple and Pro mode with appropriate settings

Autofocus module SDK examples for C# and Python are available at [Software Download Center](#) in section “SDKs for Optotune Products”. For further information, please contact [sales@optotune.com](mailto:sales@optotune.com).