MR-E-3 Development Kit

Operation Manual

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

Table of Contents

1 Overview

The MR-E-3 is a fully integrated driving solution for the Optotune MR-series beam steering mirrors. It provides access to the full functionality of the mirrors, including both open and closed loop control. Users can control the mirror and perform operations in various modes, such as waveform operation or static pointing. This document serves as an operation manual for the mirror driver.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

2 Software and System requirements

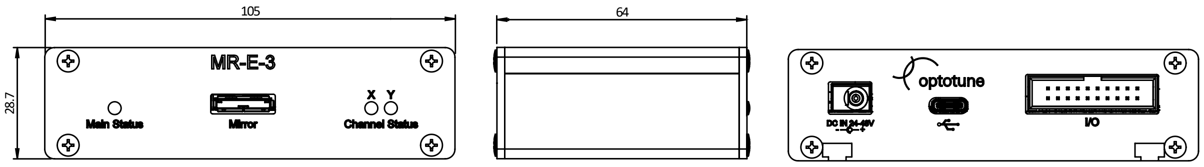
- Windows 10 or later
- USB Driver: USB 2.0 port
- Optotune Cockpit Control Software (https://www.optotune.com/registration-for-software-download)
- Software Development Kits – Python, C#
- Optional: Serial communication terminal software such as Termite Terminal (only required for simple serial communication mode)

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

3 Mechanical Details

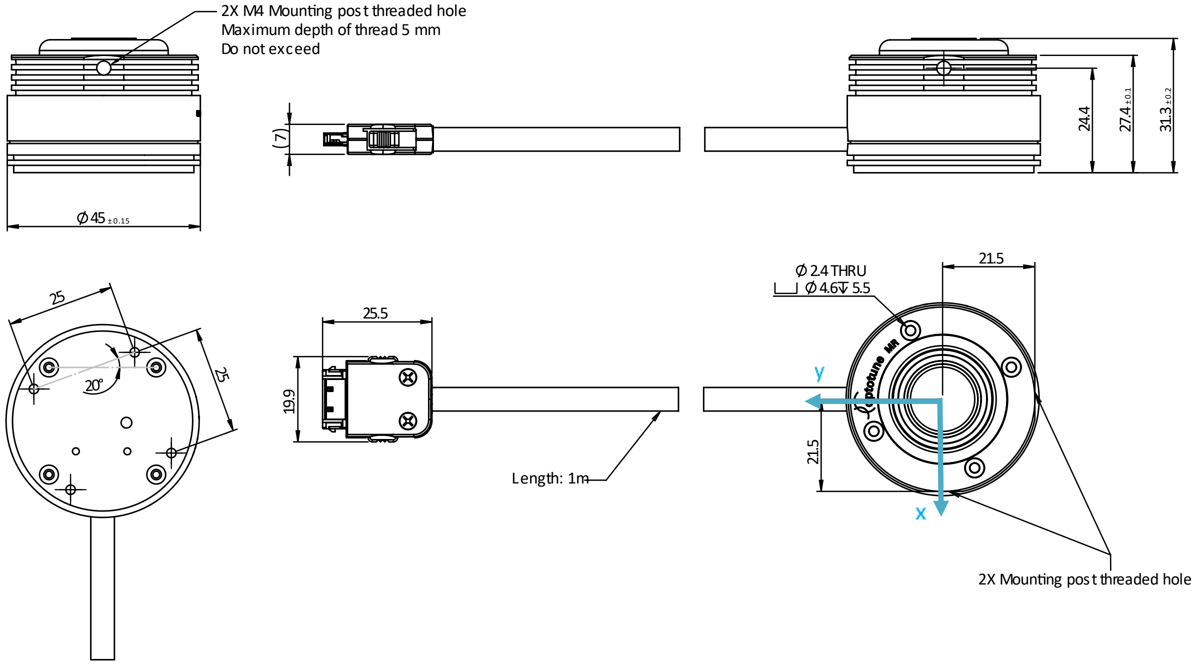
All numbers in the mechanical drawings are in millimeters.

3.1 Base Unit



3.2 Mirror Head Unit

The two mounting holes on the mirror head are M4 tapped.



***Warning:** During operation the Mirror Head Unit can become hot (up to 85°C). Avoid touching the Mirror Head Unit during operation and ensure not to place heat-sensitive equipment in the immediate vicinity of the Mirror Head Unit.*

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

4 Hardware Operation

4.1 Package Contents and Description

The figure above shows all the parts included in the MR-E-3 Development Kit package, consisting of the MR-E-3 Base Unit and the MR-E-3 Mirror head unit. The base unit consists of control electronics and the necessary accessories – a power supply and a USB cable. The head unit assembly includes the MR-series mirror, its anodized casing, and the mirror cable, which connects to the MR-E-3 controller.

Important: *Depending on how the mirror operates, the Mirror Head Unit alone may not be able to fully dissipate the generated heat into the surrounding air, leading to potential shutdowns due to overheating. Ensure proper heat dissipation by mounting the Mirror Head Unit on a heat sink with good thermal contact.*

Note: *Strong external magnetic fields may cause offsets in current (in closed-loop) or position (in open-loop).*

4.2 Power supply specifications

Both the MR-E-3 Base Unit and the driver hardware include a power supply. Specifications for the power supply are:

Specifications	Value	Units
Output Voltage (Vdc)	24	Vdc
Output Current (A)	1	A
Output Power (W)	24	W

The model number of the power supply is GE24I24-P1J from MEAN WELL manufacturer and can be purchased directly from Optotune.

4.3 Hardware Connections

The back panel of the MR-E-3 base unit has three connectors. In addition to the power connector, it features a USB Type-C connector for control through Optotune Cockpit software, software development kits, and for simple and pro mode serial communication. The I/O connector offers miscellaneous features and provides connections for communication interfaces such as SPI, UART, I^2C, or analog voltage operation. The pin-out is given in Table 1. The front panel has only mirror connector. The status LED reflects the state of the driver. Red light indicates an active error, orange warns about various behavior and green light indicates no active error (Table 2).

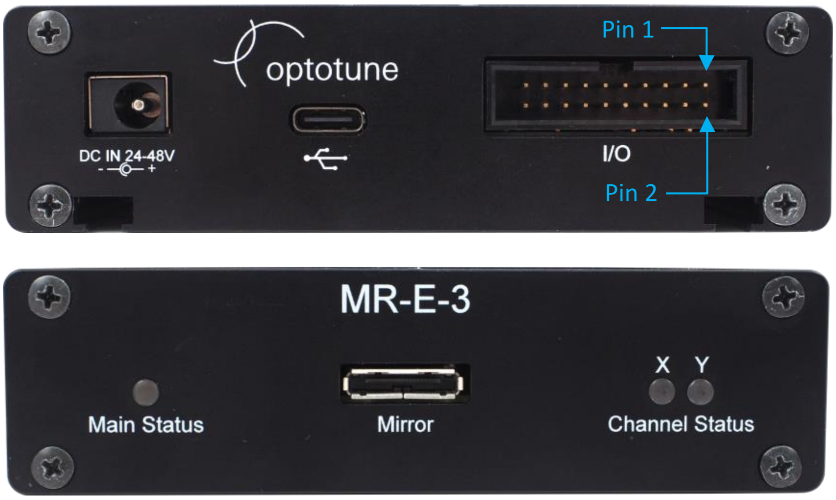


Table 1. I/O Connector pin-out.

Position	Function	Description
1	AI_X	Analog Input for X axis
2	AI_Y	Analog Input for Y axis
3	Signal GND	Digital and analog ground
4	External VCC Enable	Enable signal for external power supply (connect to Power GND to activate)
5	NRST	Reset signal for driver (connect to GND to activate)
6	SYNC_Y	Trigger Input/Output for Y axis[1]
7	UART TX/ I2C SCL	Serial interface transmitter line / I2C clock line[2]
8	SYNC_X	Trigger Input/Output for X axis[1]
9	UART RX/ I2C SDA	Serial interface receiver line / I2C data line[2]
10	SPI_DATA_NRDY	SPI Data Not Ready
11	Proxy SPI_CLK	Proxy Board SPI CLK output[3]
12	SPI_MOSI	SPI Master Output Slave input
13	Proxy SPI_CS	Proxy Board SPI Chip Select output – conversion start signal
14	SPI_MISO	SPI Master Input Slave output
15	Proxy SPI_MOSI	Proxy Board SPI MOSI output[3]
16	SPI_CS	SPI Chip select
17	STABILITY	Mirror stable
18	SPI_CLK	SPI Clock
19	External VCC	External power supply input[1]
20	Power GND	Power GND of driver
		[1] *configurable input/output* [2] *configurable external serial interface UART or I2C* [3] *Proxy Board SPI digital output with raw data*

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

Table 2. Description of status LEDs.

LED	Color	Legend
Main status	Red	Power on, no connection
	Orange	Operation OK (status register has non-zero value)
	Green	Operation OK (status register is zero)
Channel status	Red	Device error
	Green	Device detected; operation OK

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

5 Mirror Coordinate System

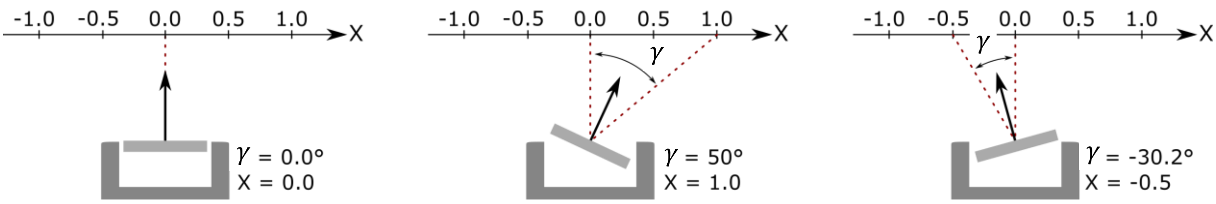
In the following section, the mirror coordinate system is introduced, and various coordinate transformations are described. You can download an example Python script that implements these transformations here.

5.1 Definition of the XY Coordinate System

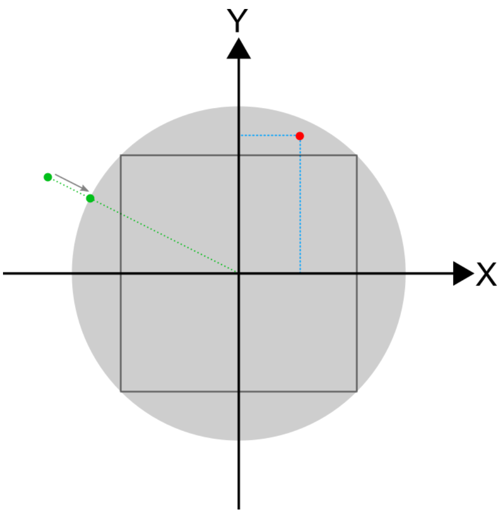
We define a coordinate system to calibrate the internal optical feedback mechanism and relate it to the physical mirror position. This coordinate system is a Cartesian coordinate system with axes X and Y. The X-axis is perpendicular to the cable protruding from the mirror head unit, and the Y-axis is parallel to this cable. The numerical values on the axes are unitless and are defined by the maximum deflection of the mirror in optical angles, i.e., 50°. Along each axis, a deflection of +50° corresponds to a value of +1, and a deflection of -50° corresponds to a value of -1. This corresponds to the projection observed on a screen if the mirror reflects a laser beam traveling along the Z-axis, incident on its center. For example, the analytical relationship between the deflection angle γ (optical angle) and the coordinate system value x along the X-axis is:

$$x = \frac{\tan(\gamma)}{\tan\,(50°)}$$

The figure below illustrates the relationship between deflection on-axis and coordinate system value using three examples:



The mirror has a maximum deflection of 25° (mechanical) in every direction. Therefore, in the XY coordinate system, a unit circle with a radius of 1 contains all accessible values, as shown in gray in the figure below:



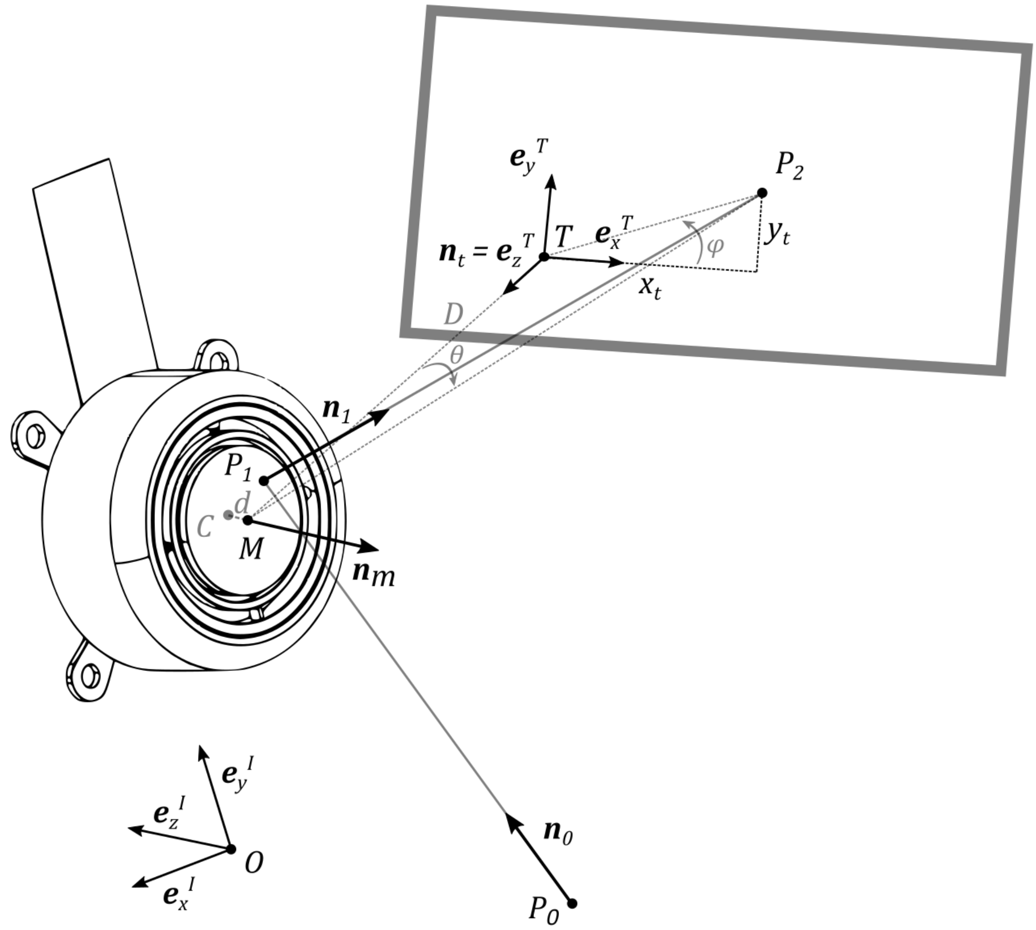
The mirror can access every possible combination of X and Y values for both X and Y < 0.7 (black square in the figure above). If one of the X or Y values exceeds 0.7, the other value must decrease so that at any time $X^2 + Y^2 \leq 1$ (the red dot in the figure above is an example). The firmware automatically reduces XY positions outside the accessible unit circle by moving them to the nearest edge of the unit circle (green dots in the figure above). This behavior prevents mirror and driver damage.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

When driving the mirror in open loop mode, it is possible to reach angles larger than 25°, which correspond to XY values greater than 1. However, these angles are outside the guaranteed and calibrated range of the mirror, and calibration accuracy can vary significantly.

5.2 Transformation Between Different Cartesian Projection Coordinates

When using the mirror in a scanning system, typically the arrangement with 0° AOI is not practical, since incident and reflected beam paths would overlap. The figure below shows the general case: The mirror with its middle point M and normal vector $\boldsymbol{n}_m$ rotates around the center of rotation C and reflects an incoming beam, specified by a point P_0 and a unit vector $\boldsymbol{n}_0$. The reflected beam starts at point P_1 and has the direction of the unit vector $\boldsymbol{n}_1$. Finally, the reflected beam hits a target plane at the point P_2, where the target plane is located at a distance D from the undeflected mirror center and a has the normal vector $\boldsymbol{n}_t$.



This general arrangement captures distortion effects due to

- AOI of the incoming beam
- Rotated target plane
- Off-centered incoming beam
- Distant center of rotation

The vector analysis to calculate the beam path is straightforward. First, inserting the equation for the incoming beam $\boldsymbol{r} = \boldsymbol{r}_{OP_0} + t \cdot \boldsymbol{n}_0$, $t \in \mathbb{R}$ into the equation for the mirror plane $(\boldsymbol{r} - \boldsymbol{r}_{OM}) \cdot \boldsymbol{n}_m = 0$, yields the intersection point P_1

$$\boldsymbol{r}_{OP_1} = \boldsymbol{r}_{OP_0} + t_1 \cdot \boldsymbol{n}_0, \qquad \text{where} \quad t_1 = \frac{(\boldsymbol{r}_{OM} - \boldsymbol{r}_{OP_0}) \cdot \boldsymbol{n}_m}{\boldsymbol{n}_0 \cdot \boldsymbol{n}_m}.$$

Note that $\boldsymbol{r}_{OM} = \boldsymbol{r}_{OC} + d \cdot \boldsymbol{n}_m$.

Then, the reflected beam is obtained by applying the law of reflection

$$\boldsymbol{n}_1 = \boldsymbol{n}_0 - 2 \cdot (\boldsymbol{n}_0 \cdot \boldsymbol{n}_m) \cdot \boldsymbol{n}_m.$$

Finally, we calculate the intersection point with the target plane P_2

$$\boldsymbol{r}_{OP_2} = \boldsymbol{r}_{OP_1} + t_2 \cdot \boldsymbol{n}_1, \qquad \text{where} \quad t_2 = \frac{\left(\boldsymbol{r}_{OT} - \boldsymbol{r}_{OP_1}\right) \cdot \boldsymbol{n}_t}{\boldsymbol{n}_1 \cdot \boldsymbol{n}_t}.$$

We can now express the vector $\boldsymbol{r}_{TP_2} = \boldsymbol{r}_{OP_2} - \boldsymbol{r}_{OT}$ in target plane coordinates

$${}_T\boldsymbol{r}_{TP_2} = \boldsymbol{A}_{TI\,I}\boldsymbol{r}_{TP_2} = \begin{bmatrix} x_t \\ y_t \\ 0 \end{bmatrix},$$

where $\boldsymbol{A}_{TI} = \boldsymbol{A}_{IT}{}^{-1} = \boldsymbol{A}_{IT}{}^{\mathrm{T}}$ is the orthogonal transformation matrix between the frames of reference I and T.

In a simplified case, for an incoming beam hitting the mirror centered and d assumed to be zero, we have $P_1 = M = C$ and one can explicitly calculate the mirror orientation from projected coordinates.

$$\boldsymbol{n}_m = \frac{\boldsymbol{n}_1 - \boldsymbol{n}_0}{\|\boldsymbol{n}_1 - \boldsymbol{n}_0\|}, \qquad \text{where} \ \boldsymbol{n}_1 = \frac{\boldsymbol{r}_{MP_2}}{\|\boldsymbol{r}_{MP_2}\|} \ \text{and} \ \boldsymbol{r}_{MP_2} = \boldsymbol{r}_{MT} + \boldsymbol{r}_{TP_2}.$$

Note that this simplification still captures the distortion introduced by the AOI of the incoming beam, which is by far the most important one to consider.

For convenience, in the following, the origin O is placed at the undeflected mirror center: $O = P_1 = M = C$.

The above equations can be used to transform between normalized mirror coordinates $(x,\ y)$ and target plane coordinates:

$(x,\ y) \rightarrow \boldsymbol{n}_m \rightarrow (x_t,\ y_t)$:

1. By definition of the mirror coordinates set ${}_I\boldsymbol{n}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and ${}_I\boldsymbol{r}_{MP_2} = \begin{bmatrix} x\,D\tan 50° \\ y\,D\tan 50° \\ -D \end{bmatrix}$. So, ${}_I\boldsymbol{n}_1$ is obtained by normalizing $\begin{bmatrix} x \\ y \\ -1/\tan 50° \end{bmatrix}$ and the mirror normal by calculating $\boldsymbol{n}_m = \frac{\boldsymbol{n}_1 - \boldsymbol{n}_0}{\|\boldsymbol{n}_1 - \boldsymbol{n}_0\|}$.
2. Redefine the actual incoming beam $\boldsymbol{n}_0$ and target plane ($\boldsymbol{A}_{TI}$ and D). Using the known mirror normal $\boldsymbol{n}_m$ calculate ${}_T\boldsymbol{r}_{TP_2}$ using the above equations and extract (x_t, y_t), i.e. the first two components of ${}_T\boldsymbol{r}_{TP_2}$.

$(x_t,\ y_t) \rightarrow \boldsymbol{n}_m \rightarrow (x,\ y)$:

1. Specify the actual incoming beam $\boldsymbol{n}_0$ and target plane ($\boldsymbol{A}_{TI}$ and D). Calculate ${}_I\boldsymbol{r}_{MP_2} = \boldsymbol{A}_{TI}{}^{\mathrm{T}} \begin{bmatrix} x_t \\ y_t \\ -D \end{bmatrix}$ and normalize to get ${}_I\boldsymbol{n}_1$, then obtain the mirror normal from $\boldsymbol{n}_m = \frac{\boldsymbol{n}_1 - \boldsymbol{n}_0}{\|\boldsymbol{n}_1 - \boldsymbol{n}_0\|}$.
2. By definition of the mirror coordinates redefine ${}_I\boldsymbol{n}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and $\boldsymbol{A}_{TI} = \mathbf{1}$. Using the mirror normal $\boldsymbol{n}_m$ from the previous step, calculate ${}_I\boldsymbol{n}_1 = {}_I\boldsymbol{n}_0 - 2 \cdot ({}_I\boldsymbol{n}_0 \cdot {}_I\boldsymbol{n}_m) \cdot {}_I\boldsymbol{n}_m$. Scale this vector with a constant factor to get the vector $\begin{bmatrix} x \\ y \\ -1/\tan 50° \end{bmatrix}$, from which x and y can be extracted.

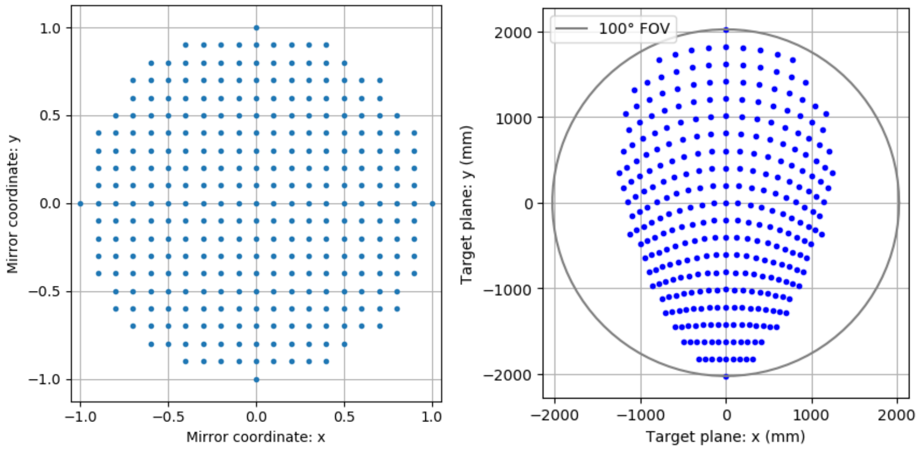
Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

As an example, consider the following typical arrangement:

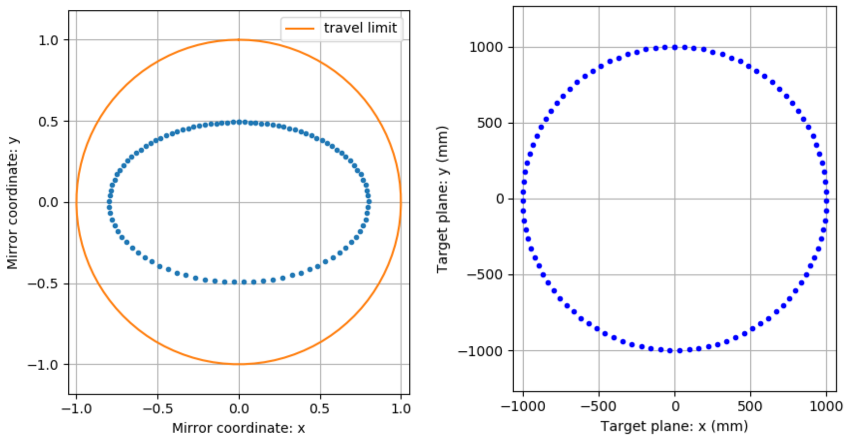
The incoming beam in the yz-plane hits the mirror centered at a 45° incidence angle. The target plane is placed perpendicular to the reflected beam for an undeflected mirror, i.e. when $x = y = 0$.

$$ {}_I\boldsymbol{n}_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}, {}_I\boldsymbol{r}_{OP_0} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \quad \boldsymbol{A}_{TI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 45° & -\sin 45° \\ 0 & \sin 45° & \cos 45° \end{bmatrix}, \quad D = 1700\ \text{mm}, \quad d = 1.3\ \text{mm} $$

The below plots show the distortions introduced in this case. For reference, also the 100° FOV is shown, which the mirror would enable for 0° AOI.

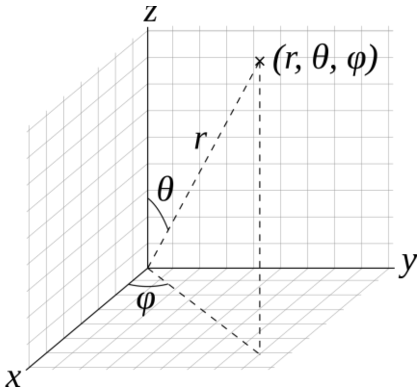


If we want to scan a circle with radius 1 m in the target plane, we can apply the transformation in the other direction to find the necessary mirror coordinates. The fact that the resulting points all lie within the travel limit, shows that this pattern is feasible.



5.3 Transformation to and from Spherical Coordinates

It is also possible to express the mirror position by using spherical coordinates instead of the XY coordinates introduced above. Spherical coordinates are not available in the MR-E-3 firmware and software. In this coordinate system, two angles define the mirror position: A polar angle between the z-axis and the reflected beam, together with an azimuthal angle defined as the angle between the x-axis and the projection of the reflected beam onto the xy-plane. The angles θ and φ are also depicted in section 5.2.



The following equations transform XY coordinates into spherical coordinates:

$$\varphi = \text{atan2}(y_t, x_t)$$

$$\theta = \text{acos}\left(\frac{D}{\sqrt{x_t^2 + y_t^2 + D^2}}\right)$$

Where $\text{atan2}(y, x)$ is the 2-argument arctangent with $\text{atan2}(y, x) = \text{atan}\left(\frac{y}{x}\right)$ for positive values of x.

The following equations transform spherical coordinates into cartesian coordinates:

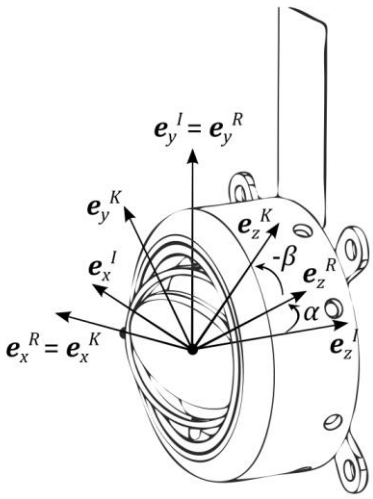
$$r = \frac{D}{\cos(\theta)}$$

$$x_t = r\sin(\theta)\cos(\varphi) = D\tan\theta\cos\varphi$$

$$y_t = r\sin(\theta)\sin(\varphi) = D\tan\theta\sin\varphi$$

5.4 Euler angles

When using Euler angles (sometimes called Cardan angles) to describe the orientation of the mirror, it is necessary to specify the sequence of rotations. One example of such a representation is depicted below. Three different coordinate systems are introduced. The inertial frame of reference I, the intermediate frame R, aligned with the outer gimbal ring, and the mirror-fixed frame K.



The inertial frame of reference I is rotated around its y-axis by the angle α. Following that rotation, the resulting intermediate frame R is rotated around its x-axis by the angle $-\beta$. The negative sign accounts for the fact that the rotation shown here acts in the negative direction, as defined by the right-hand rule. Using the two transformation matrices

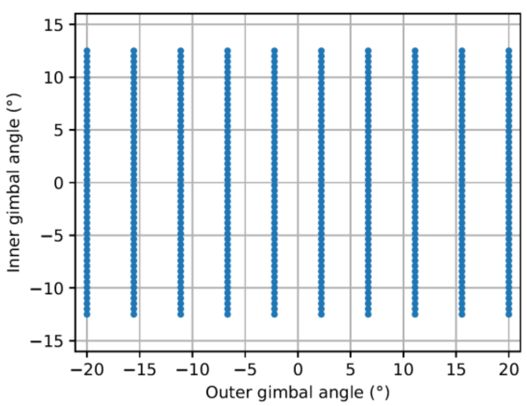
$$\boldsymbol{A}_{IR} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \text{ and } \boldsymbol{A}_{RK} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}$$

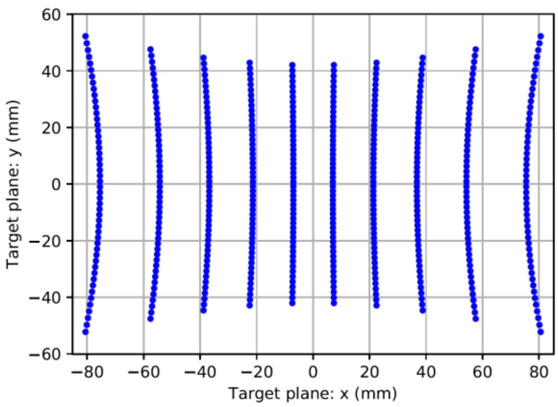
the mirror normal vector, expressed in the inertial frame of reference, can then be written as

$$_I\boldsymbol{n}_m = -{}_I\boldsymbol{e}_z{}^K = -\boldsymbol{A}_{IK}\,{}_K\boldsymbol{e}_z{}^K = -\boldsymbol{A}_{IR}\,\boldsymbol{A}_{RK}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\alpha\ \cos\beta \\ \sin\beta \\ -\cos\alpha\ \cos\beta \end{bmatrix},$$

which can be inserted into the above calculations to obtain projected coordinates.

As an example, consider scanning the resonant axis of the MR-10-30 at full angular stroke at fixed outer gimbal angles in the range -20° to 20°. The plot on right hand side shows the scanned trajectories on a screen placed perpendicular to the incoming beam (0° AOI):





Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

Amplitude control

This chapter only applies to the MR-10-30 mirror that has a resonant axis. This chapter describes how the resonant axis can be driven reliably and at stable amplitude by an automatic adjustment of the driving frequency of the sinusoidal excitation.

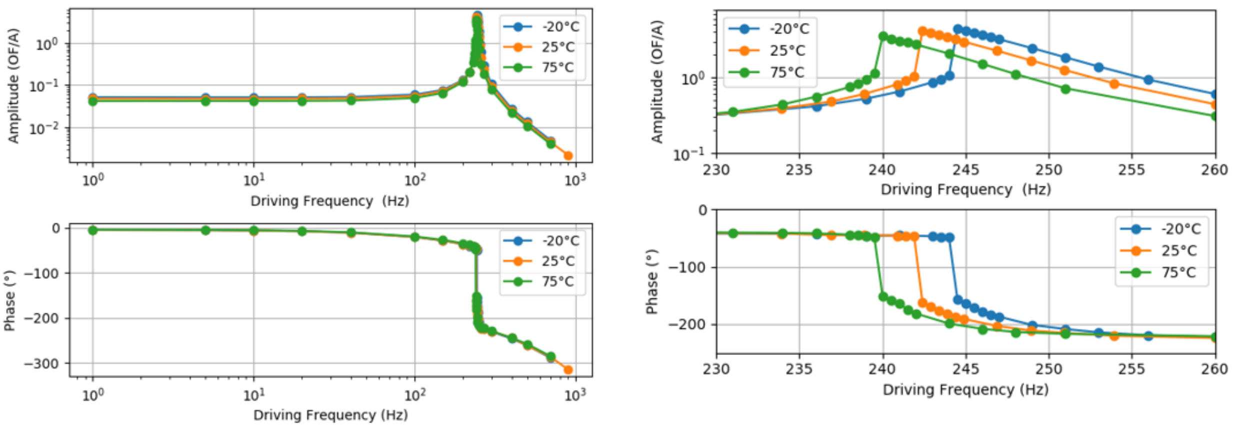
5.5 Motivation

Due to its large Q-factor of typically $Q \approx 400$ the time constant of the resonant axis at $f_0 = 250$ Hz is significant:

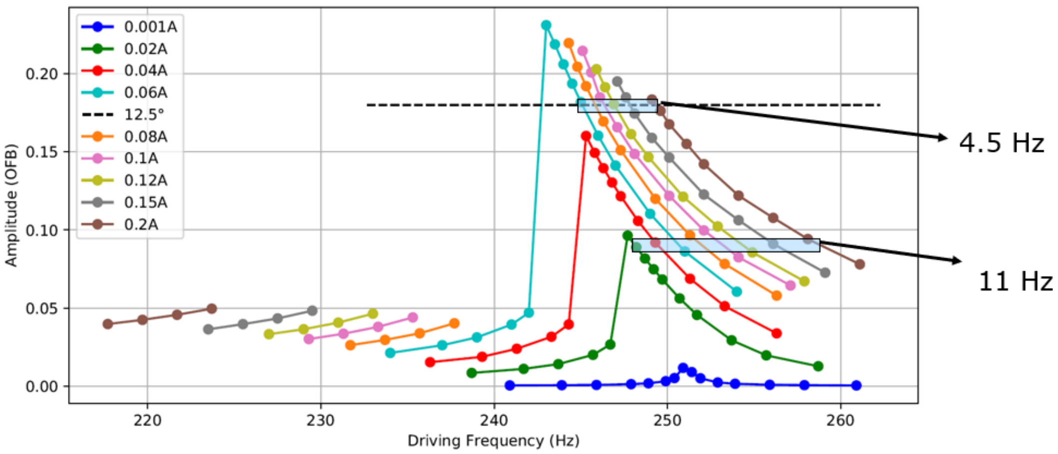
$$\tau = \frac{Q}{\pi f_0} = 0.5\ s$$

This means that the resonant axis cannot change amplitude quickly. Typically, it is therefore not possible to change the amplitude during a frame that lasts at the order of 100 ms. This means that the resonant axis angle β will be near-constant across a frame, if the frame lasts less than τ, which is typically the case. It also means that the curved scan lines depicted in section 5.4 are fundamental.

The controller should therefore focus on keeping the amplitude stable across different frames and for changing environmental conditions, namely temperature, which shifts resonance by about 5 Hz per 100 K:



There are two ways to compensate a shift in resonance. One is to use a higher nominal excitation current and then to reduce or increase the current as needed. This approach is limited to a few Hz of resonance frequency shift and increases power consumption.



The second approach changes the driving frequency and hence allows to operate always close to resonance with lowest power consumption. This approach is implemented in MR-E-3 Firmware.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

5.6 Implementation

Amplitude control follows these steps:

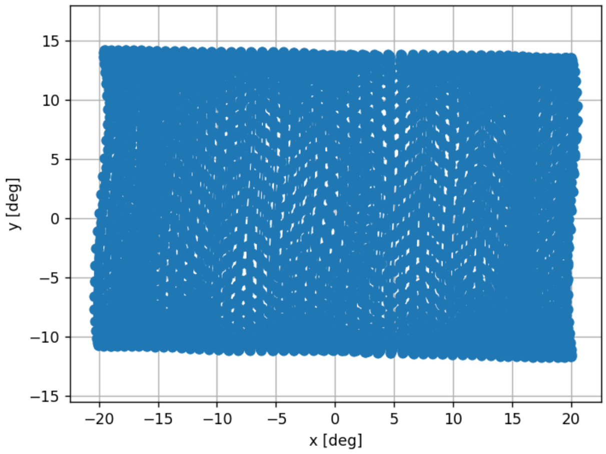
1. Controller measures natural frequency at power up of the controller.
2. User sets desired FOV (or desired mechanical scan amplitude) and activates amplitude control.
3. Suitable driving frequency and driving current are chosen based on the desired FOV and the natural frequency of the MR-10-30.
4. Resonant axis amplitude (Euler angle β) is periodically measured and the drive frequency is updated according to:

$$f_{\text{drive}} = f_{\text{drive}} - k \cdot \frac{A_{\text{meas}} - A_{\text{set}}}{S}$$

The firmware system 0x35 lists all the parameters for amplitude control in detail. Here a only the

- Enable amplitude control (register 0x3500):
  Activates amplitude control. All parameters need to be configured prior to activation. During operation changing parameters of the algorithm is not possible.
- Desired amplitude or desired FOV (registers 0x3505 and 0x3507):
  Desired amplitude. FOV equals 4x the mechanical amplitude. Only one of these registers needs to be specified.
- Frequency shift, 0x3509:
  Defines starting value for drive frequency above resonance frequency. Use higher value to smoothly ramp-up to the target amplitude. For FOVs close to the maximum this can avoid triggering the safety-shutdown.
- Resonance slope S and control gain k (registers 0x350C and 0x350A):
  Tune these to get a good compromise between smoothness of scan and convergence speed to target amplitude.
- Time window, averaging cycles, and control cycles (registers 0x350D, 0x350E, 0x350F), affect the timing of the control update rate:
  control timing = Time window * Averaging cycles * Control cycles
  The control timing interval should be larger than 1s.

Example code for amplitude control using the Python SDK can be shared upon request. A resulting scan in Euler angles looks as follows. Note that in the scan below there is a residual offset angle in the resonant axis:



5.7 Safety shutdown

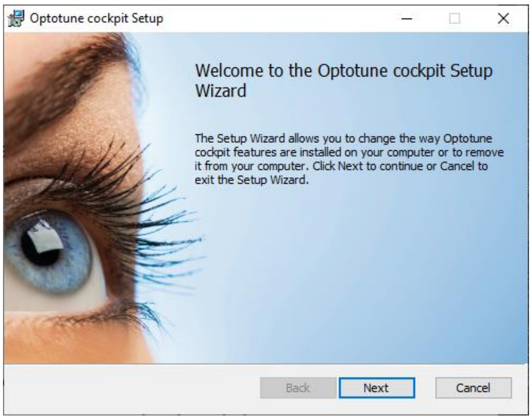
The amplitude measurement for the resonant axis is also active when amplitude control is disabled. This makes it possible to zero the output current in case an excessive amplitude was detected, which could overstress or permanently damage the resonant axis. To benefit from this safety feature, users are strongly advised to use firmware version 2.7 or later.

6 Optotune Cockpit

This section provides a short overview of the Optotune Cockpit software. It describes the main functions to control the mirror.

6.1 Installation of Optotune Cockpit

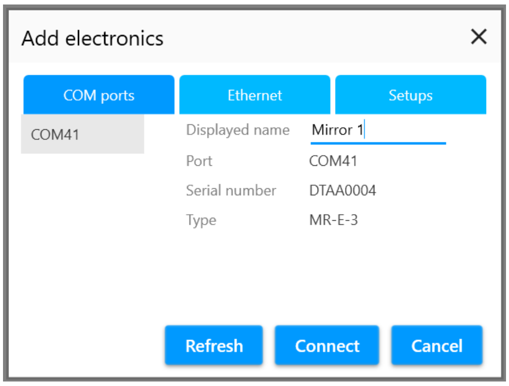
Download the Optotune Cockpit Installer from the Optotune website (https://www.optotune.com/registration-for-software-download). Open the installer file and follow the instructions on the screen.



6.2 MR-E-3 Connection

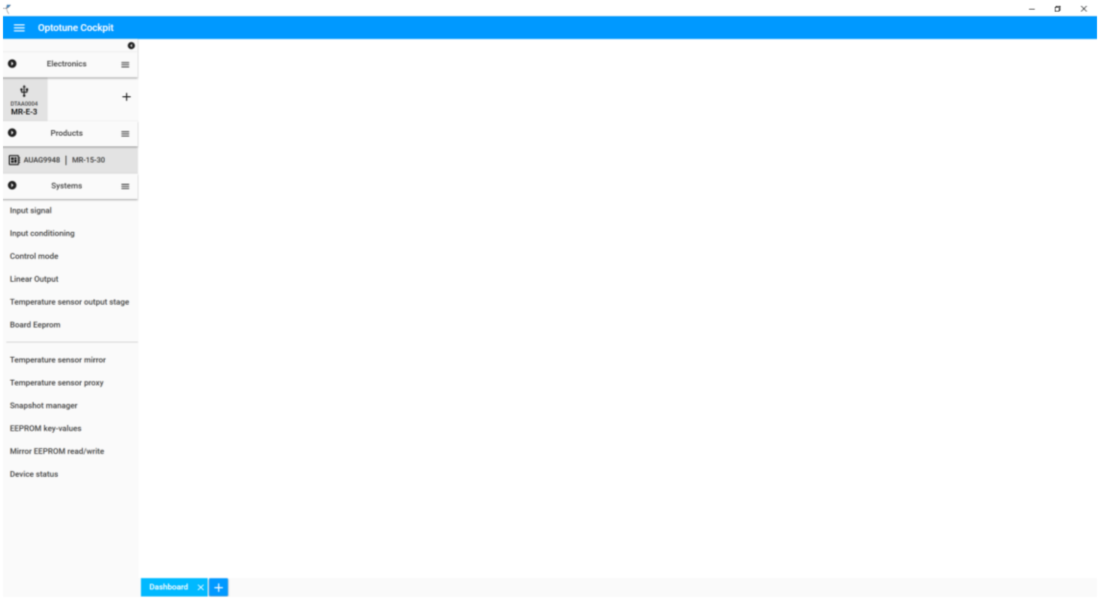
Make sure that the MR-E-3 is connected via USB to the computer and powered. Open Optotune Cockpit and click on *Connect Device.*

A window will appear, allowing you to select the device. You can also change the displayed name of the device for this session. This makes it easy to control several devices with the same computer.



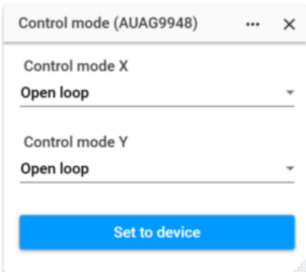
After clicking connect, the device will appear on the left side of the Optotune Cockpit window. You can add additional devices at any time by clicking + and repeating the steps above.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

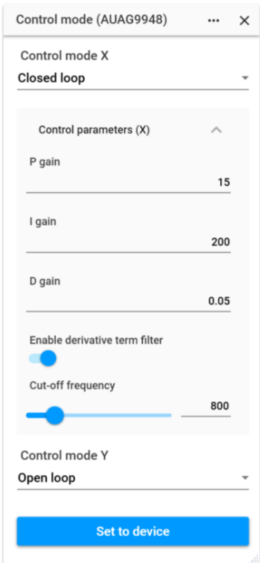


6.3 Mirror Control

The two main panels for controlling the mirror are *Input Signal* and *Control Mode*. In the *Control Mode* window, you can choose to operate the mirror in closed loop or open loop via the drop-down menus. The X and Y axes are independent and can run with different control modes. After choosing the desired modes, click *Set to device* to send the settings to the driver.

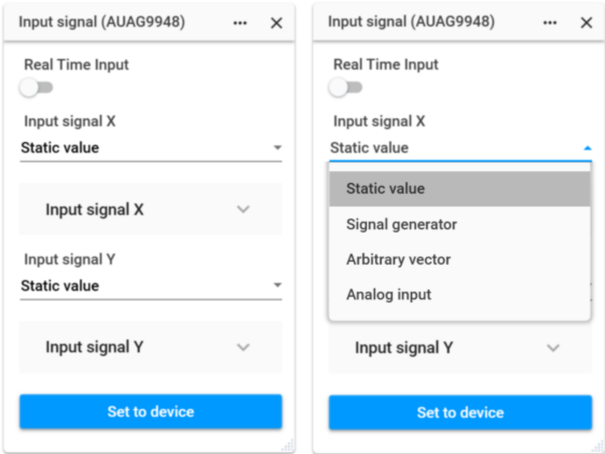


Advanced users can also set the P, I and D gain of the closed-loop PID controller. It is possible to enable/disable low pass filter for derivative term and set cutoff frequency. We recommend keeping the standard values optimized by Optotune, unless for a specific application tuning is necessary. Contact Optotune support if you believe that the standard parameters are not suitable for your application, and you are unsure how to tune them.



Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

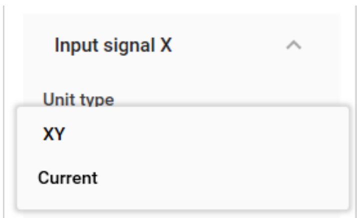
Having set the control modes for both axes, you can then use the Input Signal panel to start controlling the mirror. There are two ways to control the mirror through this panel, either with Real Time Input on or off. If Real Time Input is set to "on," any changes in the panel are transferred directly to the driver and take effect immediately. If it is turned "off," you need to press "Set to Device" after making changes for them to be sent to the driver.



MR-E-3 can be set to four different operational modes:

- **Static value:** Optotune Cockpit sends a single position or current value to the driver.
- **Signal generator:** The internal signal generator within the driver sends a time-dependent stream of position or current values until it is interrupted. Available signal generating functions are Sinusoidal, Rectangular, Sawtooth and Triangular.
- **Arbitrary vector:** The driver reads a set of user-defined positions or currents one after another and sends them to the mirror.
- **Analog input:** The driver switches to Analog input mode (see section on analog control).

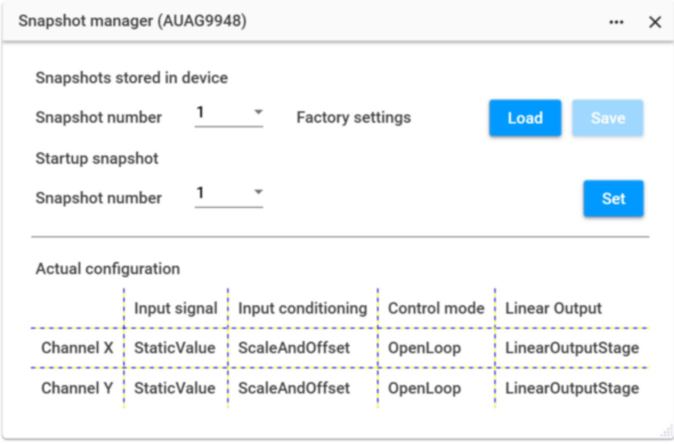
Important: *Make sure that the unit type in the Signal Input panel matches the control mode set for the corresponding axis. Closed loop control is only possible with XY unit type. Open loop control is only possible with Current unit type.*



6.4 Factory Reset

You can always return to factory setting by opening the *Snapshot manager* panel and loading snapshot 1. Snapshot 1 contains the factory settings and cannot be overridden.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

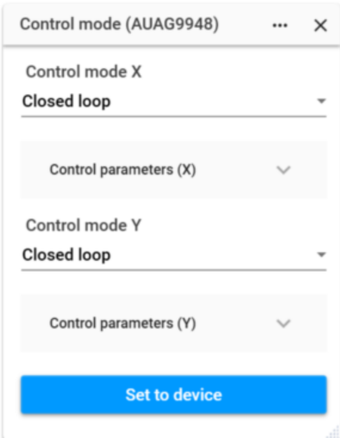


6.5 Examples

In this section, we show two examples of how to control the mirror with Optotune Cockpit. The first example includes setting both axes to closed-loop control and moving the mirror to a specific point. The second example demonstrates how to operate in mixed-mode waveform operation.

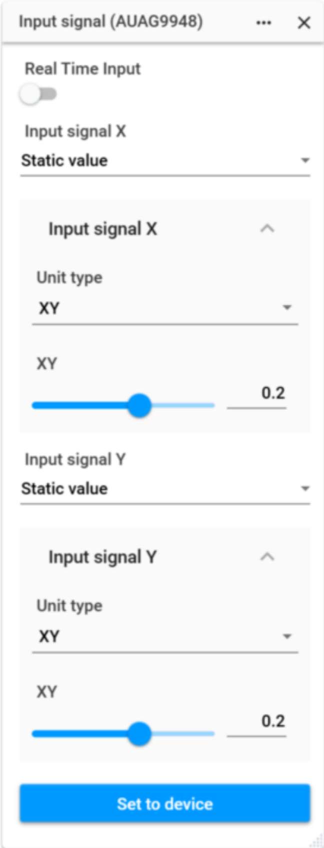
6.5.1 Closed loop static operation

We will move the mirror to the position [0.2, -0.2]. First, set both axes to closed loop control in the *Control Mode* panel and click *Set to device*.



Next, open the *Signal input* panel. Under *Input signal X* choose Static value. Repeat the same step for the Y-axis. In the *Input signal X* drop-down menu, choose unit type XY and set the value 0.2 either using the slider or typing it in directly. Repeat this for the Y-axis and set a value of -0.2. The panel should now look like this:

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

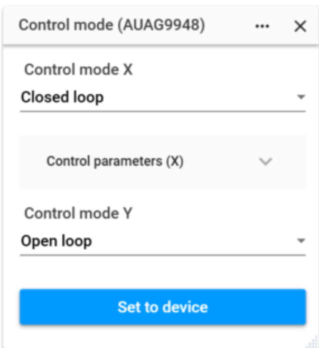


Now click *Set to device* and observe the mirror moving to the [0.2, -0.2] position.

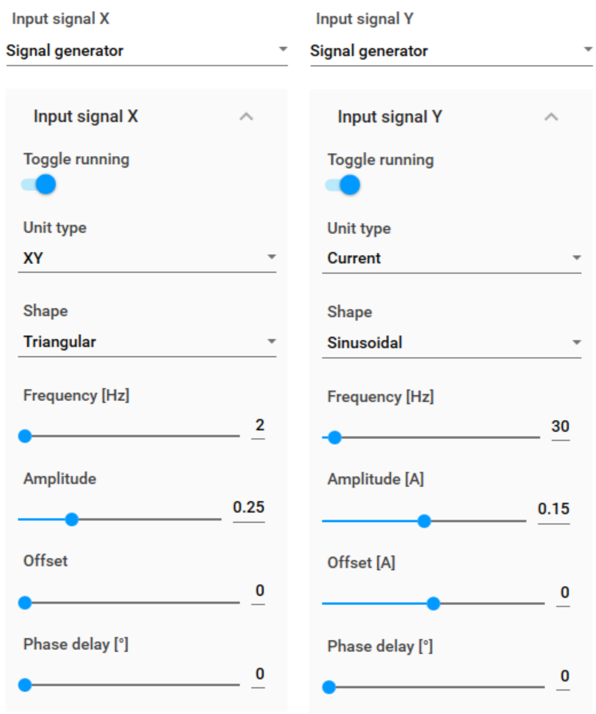
6.5.2 Mixed-Mode Waveform Operation

We will set the X-axis to closed-loop triangular waveform mode with an amplitude of 0.25 and a frequency of 2 Hz. The Y-axis will operate in open-loop sinusoidal waveform mode with an amplitude of 0.1 A and a frequency of 30 Hz. This situation, with one axis operating in closed-loop and the other axis operating in open-loop, is called mixed-mode operation.

First, set the X-axis to closed loop mode and the Y-axis to open loop mode in the *Control Mode* panel. Click *Set to device*.



Next, move to the *Input signal* panel. Set the input signal of the X-axis to *Signal generator* and activate *Toggle running*. Set *Unit type* to XY and *Shape* to Triangular. Set *Frequency* to 2 Hz and *Amplitude* to 0.25. Set the input signal of the Y-axis to *Signal generator* and activate *Toggle running*. Set *Unit type* to Current and *Shape* to Sinusoidal. Set *Frequency* to 30 Hz and *Amplitude* to 0.15. The settings should now look like this:



Now click *Set to device* and observe the mirror movement.

7 Simple Serial Communication Mode

The MR-E-3 firmware provides simple serial mode operation. When activated, the user can send string (ascii) commands (Table 3) via serial communication to control the mirror. It only offers basic control functionality. This section describes a step-by-step procedure to execute a simple position change of the mirror. Subsequently, we list all the commands available along with their input range and syntax examples. Commands sent via serial are not case sensitive. The maximum message size is 64 bytes.

7.1 Termite Terminal

Termite is a third-party software available online (https://www.compuphase.com/software_termite.htm). Termite does not have an installation file. To run it, simply run the executable file located within the downloaded files.

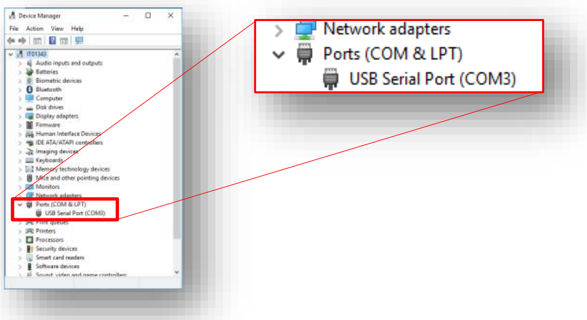
A serial communication terminal can be used to control the mirror by simply sending string (ascii) commands to the driver via the USB connection. The driver accepts ascii commands after startup, until Optotune Cockpit establishes a connection. If the user disconnects the driver through Optotune Cockpit, it will accept string commands again.

7.2 Step-by-Step procedure

Make sure that the driver is connected to the computer and powered. Do not have an open connection to the driver with Optotune Cockpit that would block the COM port.

First, find the port corresponding to the driver:

1. On the computer, open the *Run* dialog box by pressing and holding the Windows key, then press the R key.
2. Type *devmgmt.msc* and press Enter.
3. Make sure that you have no other USB devices connected.
4. Observe the *Ports (COM & LPT)* section of the Device Manager dialog window. Note down the COM port corresponding to the driver. In the case of this example the port is COM3 as shown below.



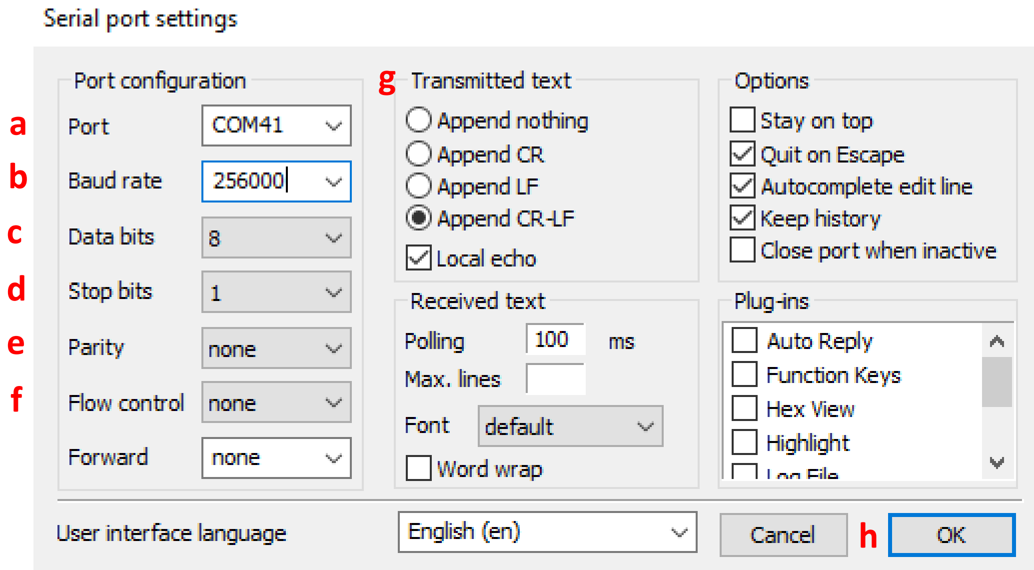
Next, configure a serial communication terminal:

1. Open the serial communication terminal by executing *Termite.exe* (see icon below).

2. Follow the steps below to configure the terminal panel. All steps are indicated by the red marking in the figure.
    a. *Port*: Make sure that the COM port is the one specified in the Device Manager (COM7 in this example)

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

b. *Baud rate*: 256000 bps
c. *Data*: 8
d. *Stop Bit*: 1
e. *Parity Bit*: None
f. *CTS Flow Control*: None (i.e. in this case unticked)
g. *Transmitted text:* All commands sent to the hardware need to be terminated with the end-of-line character combination *\r\n*.
For example, when sending the command *Start*, the End-of-Line character combination must be included at the end of the string (i.e. *start\r\n*). By setting the field *Send on enter* to the option CR-LF we ensure that every string (ascii) command you type terminates correctly with an *\r\n*.
h. *Connect*: Press *OK* and *Connect* to establish a connection between the host PC and the driver.



Now we can start to control the mirror driver via serial commands.

Type the following command and hit enter:

>> Start (commands are not case sensitive)

Make sure that the *Received Data* section of the terminal displays the following:

<< OK\r\n

This command does not execute any move. It simply works as a handshake between hardware and host PC.

To check any error codes, type:

>> Status

If there is no active error, the following text should appear (see Table 5 with error codes):

<< 0000000000\r\n

To tilt the mirror in the X-axis, type the following command in the terminal and hit enter:

>> x=0.5

In the received section of the terminal, the following text should appear:

<< OK\r\n

To reset the mirror to its zero position, type the following text into the terminal:

>> xy=0;0

The received section of the terminal should display the following text:

<< OK\r\n

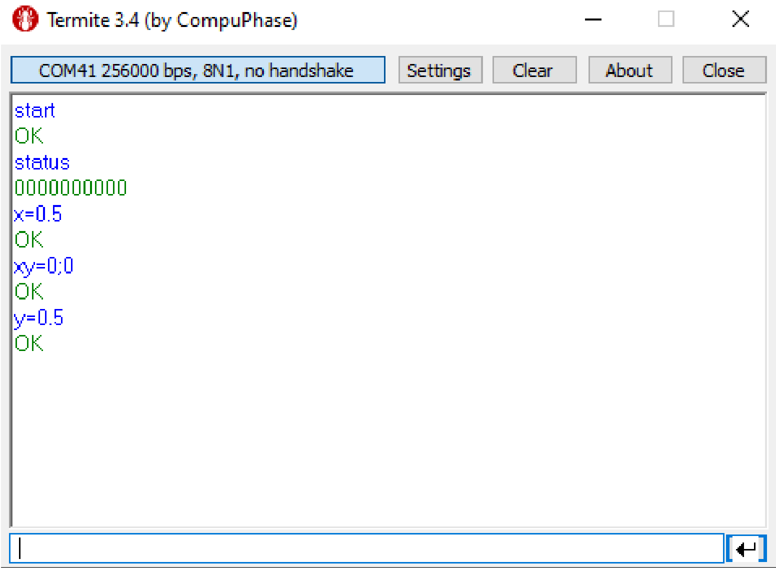
The mirror will then return to the zero-position.

To tilt the mirror in the Y-axis, type the following command in the terminal and hit enter:

>> y=0.5

In the received section of the terminal should display the following text:

<< OK\r\n



7.3 List of commands available

Table 3. List of commands used in "Simple Mode".

Simple mode command	Description
START[CR][LF]	Returns "OK", just a simple handshake command to confirm that the communication is working. Otherwise, no response is received.
STATUS[CR][LF]	Returns the status register in hex format, which encodes the status and errors of the device. Example: "0000000008[CR][LF]". See Table 3 for further description of the status bytes.
ACKNOWLEDGE[CR][LF]	Clears history error flags in the status register. Answers "OK".
RESET[CR][LF]	Restarts controller's firmware. Note: no answer is sent via serial line.
GOPRO[CR][LF]	Starts binary protocol-based mode of serial communication. Serial message CRC is not checked.
GOPROCRC[CR][LF]	Starts binary protocol-based mode of serial communication. Serial message CRC is checked.
GETID[CR][LF]	Answers with firmware serial number. Example: "14352500-00-A[CR][LF]".
GETVERSION[CR][LF]	Answers with firmware version number. Example: "1.3.741632 [CR][LF]".
GETSN[CR][LF]	Answers with board and device serial number. Example: "Board: CDAA1234, Device: ANAA1234[CR][LF]".
GETGITSHA1[CR][LF]	Answers with 40 bytes hexadecimal GIT build identification. Example : "eb8115e6b04814f0c37146bbe3dbc35f3e8992e0[CR][LF]".
GOTODFU[CR][LF]	Starts controller's loader for firmware update. Note: no answer is sent via serial line.
CURRENTX=%f[CR][LF]	Drives channel X with a current of float milliamperes, float should be in the range (-1136.0, 1136.0). Note: Value should be within set current limit.
CURRENTY=%f[CR][LF]	Drives channel Y with a current of float milliamperes, float should be in the range (-1136.0, 1136.0). Note: Value should be within set current limit.
PIDOFX=%f[CR][LF]	Drives channel X with a PID with a target optical feedback of float.
PIDOFY=%f[CR][LF]	Drives channel Y with a PID with a target optical feedback of float.
PIDOFXY=%f;f[CR][LF]	Drives channels X and Y with a PID with target optical feedback of float types.
X=%f[CR][LF]	Drives channel X to achieve calibrated coordinate of float, float should be in the range (-1.0, 1.0).
Y=%f[CR][LF]	Drives channel Y to achieve calibrated coordinate of float, float should be in the range (-1.0, 1.0).

XY=%f;%f[CR][LF]	Drives channels X and Y to achieve calibrated coordinates XY of float types.
GETTEMP[CR][LF]	Answers with actual temperature of connected mirror. Returned temperature is a decimal value in units of degree Celsius. Example: "28.250[CR][LF]".
SETTEMPLIM=%f[CR][LF]	Sets operational temperature limit in degree Celsius.
GETDEVICESN[CR][LF]	Answers with serial number of a device connected. Example: "Device: ANAA1234[CR][LF]".
DETECTDEVICE[CR][LF]	Runs autodetection of device, answers with device name. Example: "MR-15-30[CR][LF]".
SETCURLIMIT=%f;f[CR][LF]	Sets current limit value on both channels. Command supports decimal parameter value in mA units. Current value is limited either by the power capabilities of MR-E-3 controller itself or connected device. First argument is positive limit (0, Max current), second one is negative (- Max current, 0>.
GETCURLIMIT[CR][LF]	Answers with set current limits – (Max current, -Max current). Example: "500, -500".

For each command the terminal transmits to the driver, there should be a response from the driver. Table 4 lists all possible responses.

Table 4. List of commands returned by the mirror driver.

Simple mode reply	Description
OK[CR][LF]	Command accepted and performed without limits.
NO[CR][LF]	Command not accepted, for any reason.
OL[CR][LF]	Command not accepted, because parameter reached lower limit.
OU[CR][LF]	Command not accepted, because parameter reached upper limit.
ERROR[CR][LF]	Command not available.

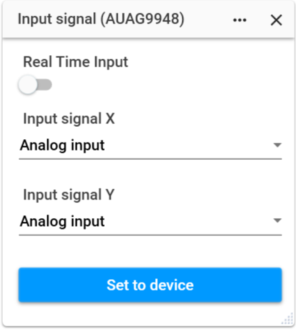
Table 5 below details the meaning of each bit returned by the *status\r\n* command.

Table 5. Status register map.

Bit#	Description
0	Proxy not connected
1	Proxy temperature threshold is reached
2	Mirror temperature threshold is reached
3	Mirror EEPROM not valid
4	Mirror not stable
5	Output current limit is reached
6	Output current average limit is reached
7	XY input is trimmed
8	Proxy was disconnected
9	Proxy temperature threshold was reached
10	Mirror temperature threshold was reached
11	Output current limit was reached
12	Output current average limit was reached
13	XY input was trimmed
14..31	Reserved

8 Analog Mode

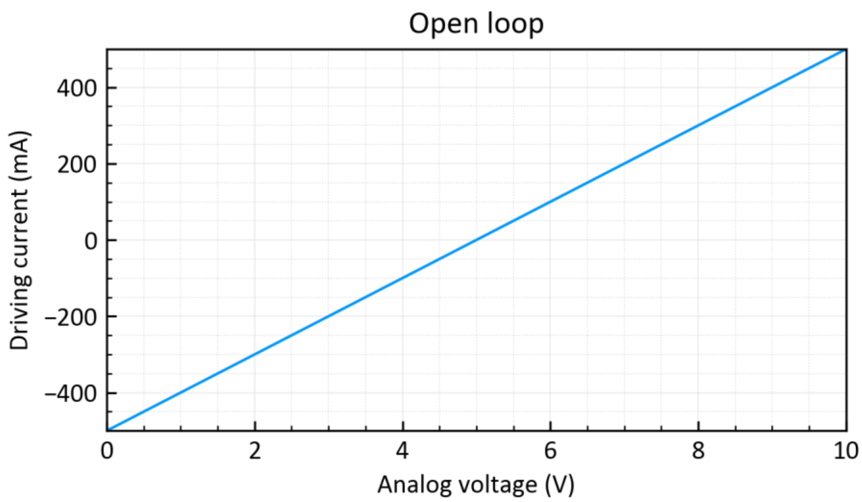
The analog mode must be explicitly activated through Optotune Cockpit or SPI. When analog mode is active, the analog input pins should be connected and not be left floating. Choose analog mode in the *Signal input* panel and click *Set to device*:

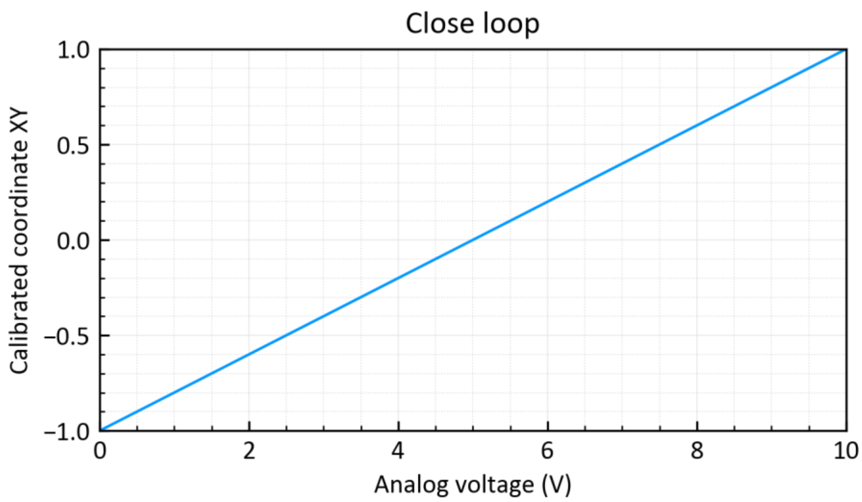


8.1 Input Pins and Signal Levels

The input pins are located on the I/O connector. Pin 1 controls the X-axis and pin 2 controls the Y-axis. The analog inputs are bipolar single ended signals to ground.

The range of AI_X and of AI_Y is from 0 V to 10 V. The input signals map linearly to this voltage. The figure below shows how the whole voltage range is mapped to the open loop and closed loop input range.





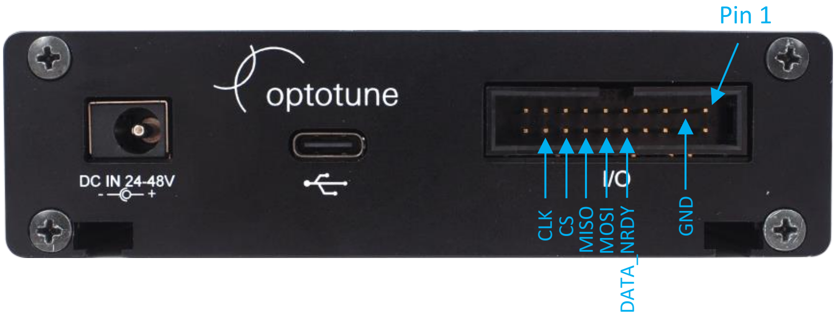
Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

9 Software development kits

Python and C# SDKs are available in the Software section here: https://www.optotune.com/downloads.

10 SPI Interface

This section describes how the MR-E-3 operates as SPI slave with the following pin assignments on the I/O connector:



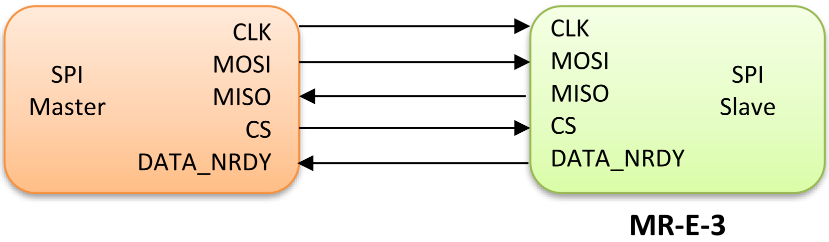
SPI is always active and does not need to be activated or deactivated through Optotune Cockpit. The mirror can be driven through SPI (pinout in Table 6) while the master is receiving real time data. The master has full access to all MR-E-3 registers through SPI. The logic level is 3.3V (CMOS).

Table 6. SPI signals pinout on IO connector.

Signal	Pin number	IO Direction
GND	3	-
SPI_DATA_NRDY	10	Out
SPI_MOSI	12	Out
SPI_MISO	14	In
SPI_CS	16	In
SPI_CLK	18	In

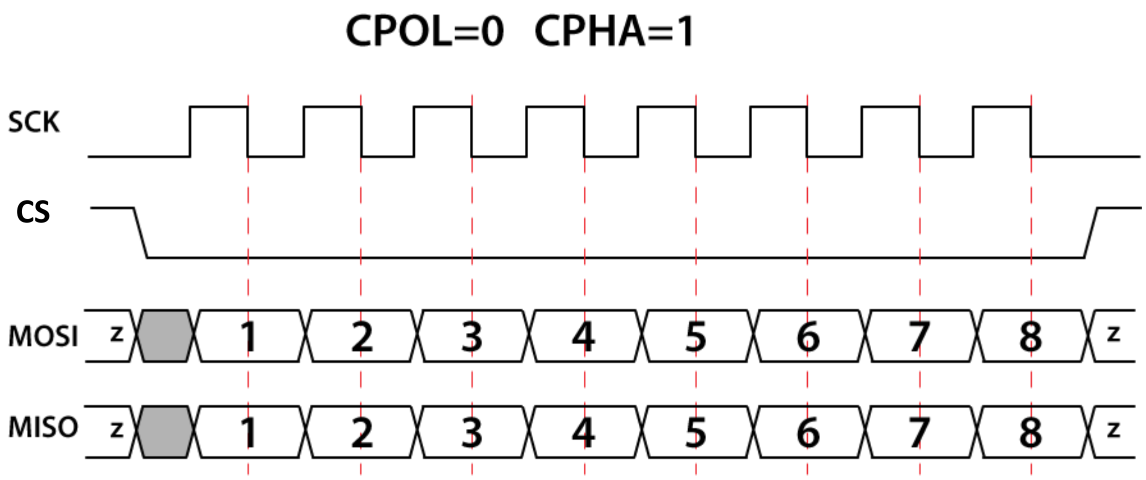
10.1 Interface

MR-E-3 driver has a four wire SPI slave interface as shown below:



MR-E-3 uses SPI Mode 1 (CPOL = 0, CPHA = 1). The Interface accepts data on the positive edge of the clock and samples the data at the negative edge. Additionally, MR-E-3 provides a SPI data not ready signal for accurate time synchronization with the master. The following timing diagram shows the 8-bit data frame.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com



10.2 Framing

Each transfer comprises 14 bytes with 8 bits each. A negative edge on the CS line marks the beginning of a new frame. A frame ends with a positive edge on the CS line. Note that MSB comes first in the transmitted data.

10.3 Timing and synchronization

- The interface is designed to work up to 32 MHz clock frequency, depending on the wire impedance and environment.
- To ensure a proper operation of the interface, the time between the negative edge of the CS signal and any edge on the clock signal must be at least 20 ns.

In order to achieve accurate time synchronization between the slave and the master, the MR-E-3 provides a SPI_DATA_NRDY signal. SPI_DATA_NRDY signal is driven high, signaling that the receiving frame is being processed, working independently of CS pin logic level. Once the frame is processed and the data requested are ready for transmission, the SPI_DATA_NRDY is driven low. Master should not send a new SPI frame while the SPI_DATA_NRDY is high. MR-E-3 driver updates its registers at a frequency of 40 kHz. Frames received at a higher frequency cannot be processed.

10.4 Data Structure

The following sections explain the 14 bytes (seven 16-bit words) for each data transfer (MISO and MOSI). The master can write and read registers to control the driver. The master can write up to two registers and read up to three registers with one frame transaction. Regardless of a read or write request, the last eight bytes of the SPI data frame slave response is reserved for reading back registers. For full description of the MR-E-3 register map, please refer to MR-E-3 Firmware documentation (https://www.optotune.com/registration-for-software-download).

10.4.1 Master Out, Slave In (MOSI)

A data frame corresponding to a write request contains the following words:

Word	Description
0	Write flag (0x0001)
1	'System ID1' + 'Register ID1' to perform write
2	'System ID2' + 'Register ID2' to perform write
3-4	Data to write in address 'System ID1' + 'Register ID1'
5-6	Data to write in address 'System ID2' + 'Register ID2'

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

A data frame corresponding to a read request contains the following words:

Word	Description
0	Read flag (0x0000)
1	'System ID' + 'Register ID' to perform read
2-6	0

10.4.2 Master In, Slave Out (MISO)

The slave responds to a write request with the following frame:

Word	Description
0	Write flag (0x0001)
1	'System ID1' + 'Register ID1', by write fail: 0x0000
2	'System ID2' + 'Register ID2', by write fail: 0x0000
3-4	Read back data according to SPI read pointer register 0, by read back fail: 0x7cf0bdc2
5-6	Read back data according to SPI read pointer register 1, by read back fail: 0x7cf0bdc2

The slave responds to a read request with the following frame:

Word	Description
0	Read flag (0x0000)
1-2	Read back data of address 'System ID' + 'Register ID' asked from previous read request, by read back fail: 0x7cf0bdc2
3-4	Read back data according to SPI read pointer register 0, by read back fail: 0x7cf0bdc2
5-6	Read back data according to SPI read pointer register 1, by read back fail: 0x7cf0bdc2

For full description of the MR-E-3 controlling and configuring capabilities using SPI, please refer to MR-E-3 Firmware manual.

10.5 Examples

Refer to MR-E-3 Firmware documentation for complete register references.

10.5.1 Setting open loop to both axis and reading back X, Y

By default, the driver is configured in open loop mode and the read back of the SPI is pointing to the optical feedback read registers (0x2300, 0x2301). Therefore, no initial configuration is needed for this example. The SPI master can directly send a SPI frame to command a current in both channels.

In the following example frame the master commands 0.050 A in X channel and 0.080 A of opposite direction in Y channel.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x5000	0x5100	0x3d4ccccd	0xbda3d70a

Explanation:

- Word 0: This example is a write request. 0x0001 is the SPI write flag.
- Word 1: 0x5000 is the register address in the static input system of X axis that sets current
- Word 2: 0x5100 is the register address in the static input system of Y axis that sets current
- Words 3-4: 0x3d4ccccd is the single precision floating point representation of 0.050.
- Words 5-6: 0xbda3d70a is the single precision floating point representation of -0.080.

While the master sends the above frame in the MOSI line, the driver will send the optical feedback values in Words 3-4 and Bytes 5-6 in the MISO line, see subsection 10.4.2.

10.5.2 Setting closed loop triangular to X axis and open loop sinusoidal to Y axis

For this example, the operation mode of the driver needs to be configured accordingly. The following frames have to be sent.

1. Select signal generator system as active input for X and Y axis.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x4000	0x4005	0x00000060	0x00000061

Explanation:

- Word 0: This example is a write request. 0x0001 is the SPI write flag.
- Word 1: 0x4000 is the register address for the active input system of X axis.
- Word 2: 0x4005 is the register address for the active input system of Y axis.
- Word 3-4: 0x00000060 is the ID of the signal generator system for X axis.
- Word 5-6: 0x00000061 is the ID of the signal generator system for Y axis.

2. Activate closed loop control for the X axis and open loop control for Y axis. See Operation mode register 0x2526 in Miscellaneous features in Firmware documentation. Can be sent twice.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x2526	0x2526	0x00000005	0x00000005

Explanation:

- Word 1: 0x2526 is the register address for the operation mode.
- Word 2: 0x2526 is the register address for the operation mode, sent again.
- Word 3-4: 0x00000005 is the operation mode value for Close loop X only.
- Word 5-6: 0x00000005 is the operation mode value for Close loop X only, sent again.

3. Configure signal unit. XY unit for X axis and current for Y axis.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x6000	0x6100	0x00000002	0x00000000

Explanation:

- Word 1: 0x6000 is the register address for the unit of the signal generator for X axis.
- Word 2: 0x6100 is the register address for the unit of the signal generator for Y axis.
- Words 3-4: 0x00000002 is the ID for XY unit.
- Words 5-6: 0x00000000 is the ID for current unit.

4. Configure signal shape. Triangular for X axis and sinusoidal Y axis.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x6002	0x6102	0x00000001	0x00000000

Explanation:

- Word 1: 0x6002 is the register address for the shape of the signal generator for X axis.
- Word 2: 0x6102 is the register address for the shape of the signal generator for Y axis.
- Words 3-4: 0x00000001 is the ID for triangular shape.
- Words 5-6: 0x00000000 is the ID for sinusoidal shape.

5. Configure signal frequencies (e.g. 5.0 Hz for X axis and 10.0 Hz for Y axis) with sending float values.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x6003	0x6103	0x40a00000	0x41200000

Explanation:

- Word 1: 0x6003 is the register address for the frequency of the signal generator for X axis.
- Word 2: 0x6103 is the register address for the frequency of the signal generator for Y axis.
- Words 3-4: 0x40a00000 is the single precision floating point representation of 5.0.
- Words 5-6: 0x41200000 is the single precision floating point representation of 10.0.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com

6. Configure signal amplitudes (e.g. 0.6 for X axis and 0.05 A for Y axis) with sending float values.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x6004	0x6104	0x3f19999a	0x3d4ccccd

Explanation:

- Word 1: 0x6004 is the register address for the amplitude of the signal generator for X axis.
- Word 2: 0x6104 is the register address for the amplitude of the signal generator for Y axis.
- Words 3-4: 0x3f19999a is the single precision floating point representation of 0.6.
- Words 5-6: 0x3d4ccccd is the single precision floating point representation of 0.05.

7. Set the run flag for the signal generators.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x6001	0x6101	0x00000001	0x00000001

Explanation:

- Word 1: 0x6001 is the register address for the run flag of the signal generator for X axis.
- Word 2: 0x6101 is the register address for the run flag of the signal generator for Y axis.
- Words 3-4: 0x00000001 sets flag to true.
- Words 5-6: 0x00000001 sets flag to true.

10.5.3 Activate analog mode

The default active input system of the driver is the static input. Analog input mode can be explicitly configured through SPI with the frame below.

Word 0	Word 1	Word 2	Words 3-4	Words 5-6
0x0001	0x4000	0x4005	0x00000058	0x00000059

Explanation:

- Word 0: This example is a write request. 0x0001 is the SPI write flag.
- Word 1: 0x4000 is the register address for the active input system of X axis.
- Word 2: 0x4005 is the register address for the active input system of Y axis.
- Word 3-4: 0x00000058 is the ID of the analog input mode system for X axis.
- Word 5-6: 0x00000059 is the ID of the analog input mode system for Y axis.

Optotune AG | Bernstrasse 388 | CH-8953 Dietikon | Switzerland
Phone +41 58 856 3000 | www.optotune.com | info@optotune.com