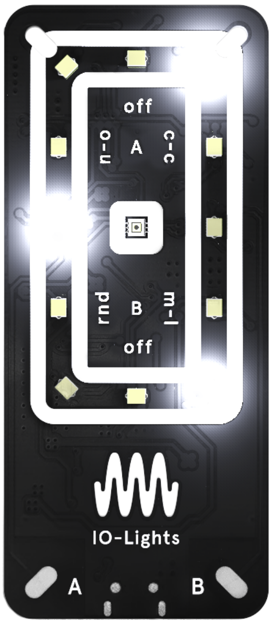
IO-Lights

Light-Sensitive MIDI Controller



User Manual

Rev. B - March 2021 (Firmware version 1.1)

Index

Safety Instructions

Please follow the instructions for the use of IO-Lights below to guarantee proper operation and ensure warranty from Instruments of Things.

Water
2.4SINK shouldn’t be used in humid environments to avoid damaging electrical components. However, our branded Movesense sensors are waterproof.

Fire
2.4SINK shouldn’t be operated in environments deceeding 0°C or exceeding 50°C.

Transport
To avoid mechanical damages 2.4SINK should be always transported in the original package without the antenna mounted.

What’s included

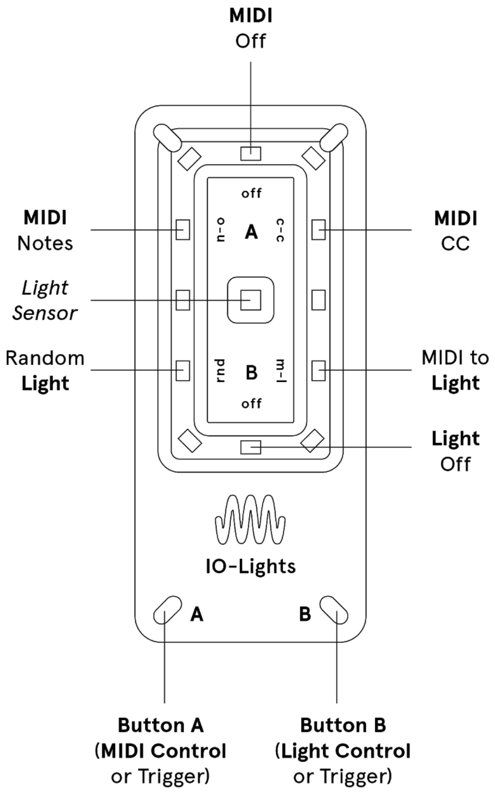
- IO-Lights light sensitive MIDI controller
- USB cable (USB-C to USB Micro)

About IO-Lights

IO-Lights is a MIDI controller that controls MIDI CC or MIDI Note messages via environmental light. A high-resolution light sensor in the center of the product makes it possible. In addition, MIDI notes can also be sent to IO-Lights, which are then represented as light by the 12 LEDs on the device.

IO-Lights is used as a common MIDI controller. Just connect it via USB to the music software of your choice and start controlling sounds with light. You want to connect IO-Lights to your hardware synthesizer? Then simply use an additional USB MIDI host (e.g. this). Configuration and firmware updates can be done directly in a web app.

Getting Started



Light Sensor

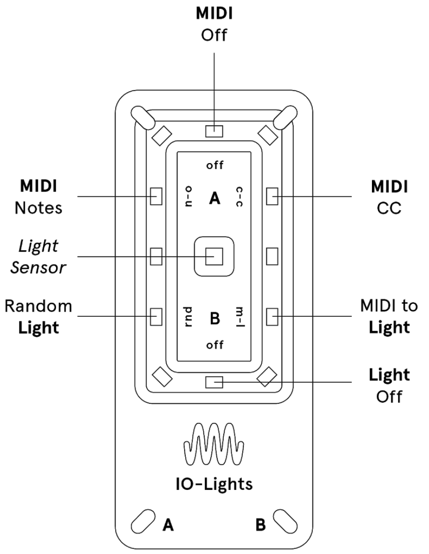
Based on the high-precision light sensor, IO-Lights generates and sends MIDI (Note or Control Change) messages to a USB-MIDI compatible host. The incoming brightness range can be either set manually or to automatically adapt to the environmental light. The outgoing note range can be set in the web app as well. Additionally, internal note quantization can be configured in the web app to send out notes in a specific scale (e.g. major, minor, chromatic, etc.). If clock synchronization is disabled, the minimum and maximum note length can be customized.

Furthermore, the sensitivity of the light sensor can be configured up to 0.25 Lux precision. Note, that depending on the light sensor sensitivity, the latency increases due to a higher measurement integration time.

LEDs

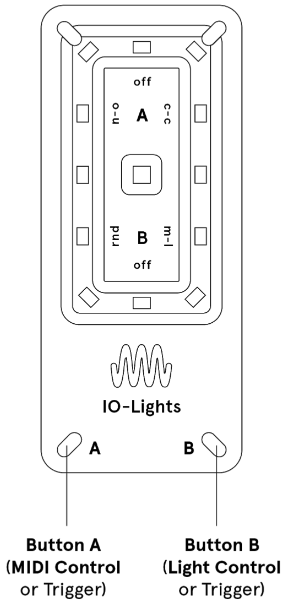
The twelve, white LEDs on the UI of IO-Lights are intended to be used as a light source to influence the light sensor. The LEDs can be either controlled externally by incoming MIDI messages (m-l) or via a built-in LED mode (random or feedback loop). If MIDI notes are sent to IO-Lights, a single LED represents a halftone starting from the top independent of the octave. If enabled, automations (i.e. Control Change) are visualized in a clockwise circle animation starting from the to.

This is what makes IO-Lights so unique: By creating a visual bridge between incoming and outgoing data over a different, physical medium.



Touch Buttons

IO-Lights offers two capacitive touch buttons to select MIDI output mode (Button A) and LED mode (Button B). When touching a button, the according LED shows the current mode. Additionally, the touch buttons can be used to trigger a fixed note or send Control Change messages. The notes and controller numbers can be customized in the web app.



Random LED Mode

In random (rnd) LED mode, the LEDs create random light patterns with a random number of LEDs turned on, making IO-Lights a visual, random MIDI source, which can be influenced by reflecting the LED light or an external light source. A time randomness parameter can be configured from several seconds to just a few microseconds in the web app. The number of active LEDs can be set to a random value or to a specific number in the web app, as well.

Feedback Loop Mode

During the development of IO-Lights, we realized the power of creating a visual bridge between incoming and outgoing MIDI data. At some point, the IO-Lights prototype (accidentally) started to oscillate, as we were creating a MIDI loopback with an external audio software. It was wild, and interesting! However, the visual feedback loop was quite unstable and only worked in a few situations and environments. Furthermore, the additional latency of the external audio software (or rather USB-MIDI host) led to a low oscillation frequency. So we started to focus on this characteristic resulting in the feedback loop mode.

The solution was quite simple: The higher the measured brightness of the light sensor, the less LEDs are turned on (in clockwise circle). Based on this principle, a stable, visual feedback loop can be created in most environments and influenced by reflecting material or external light.

Depending on the measured brightness, influenced by the feedback loop, random values are generated and converted into MIDI notes or Control Change messages. To keep control over the wildness, the internal note quantizer, automation slew limiter and external clock synchronization can be optionally enabled.

With all these features combined, IO-Lights evolves into an organic, visual instrument with many surprises, and still does what you want.

The feedback loop mode is always used with fixed brightness range. Otherwise, the oscillation becomes unstable. If desired, one can change the brightness range to adapt the feedback loop to any environment condition. Furthremore, the oscillation frequency depends on the light sensor sensitivity. The higher the sensitivity, the lower the oscillation frequency.

Clock Synchronization

IO-Lights can be synchronized to an external clock via MIDI to always play in time with your other instruments. Based on the external clock (and clock divider), IO-Lights sends the current measured brightness value in regular intervals as MIDI Note or Control Change message.

For better usability, IO-Lights will use the internal clock until an external clock has been received if clock synchronization is enabled. From this time point, IO-Lights will always stay in sync with the external clock. Hence, if the external clock is stopped, IO-Lights won’t send any MIDI messages.

Web App (Configuration)

To allow configuration with a wide range of end devices, we have created a responsive web app, hosted on our website (instrumentsofthings.com/io-app). All available parameters can be changed in real-time. New configurations can be persisted in the internal flash memory of IO-Lights.

To keep IO-Lights USB-MIDI class compliant (no additional drivers needed), the configuration parameters are transmitted via MIDI SysEx.

Note: If you require our MIDI SysEx protocol for your own application, feel free to email us: support@instrumentsofthings.com

Furthermore, new firmware images are automatically discovered from our server and can be flashed directly to IO-Lights in the web app via WebUSB. At the moment of this writing, only Chrome/Chromium, Edge and Opera support both USB protocols.

A description of the different parameters can be found in the following tables.

MIDI Notes	
Output Note Range	Range slider to set minimum and maximum output note. Default: C2 - C4
Minimum Note Length	Minimum note length in milliseconds Default: 50
Maximum Note Length	Maximum note length in milliseconds. If value is set to 0 (infinite), the current note is held until the measured brightness reaches the value range of another note. Default: 200
Quantizer	The root note and scale to which outgoing MIDI notes are quantized to. Default: C - Chromatic

MIDI Control Change	
Controller Index	Typically, manufacturers of MIDI compatible hardware use different indices for specific sound parameters. To control and automate the desired parameter of an external instrument via IO-Lights, the controller index can be customized (0 - 127). If you are not familiar with MIDI CC, here is an overview of common controller indices. Default: 74 (Frequency Cutoff)
Visualize MIDI CC	If enabled, the incoming MIDI Control Change values are

	visualized with the UI LEDs in a clockwise circle animation, starting from the top. Note: LED mode (Button B) has to be in m-l (MIDI to Light) mode for CC visualization. Default: Off
CC Slew Limiter	If enabled, the outgoing MIDI Control Change value is smoothed. Default: Off
Slew Limit	Sets the slew rate. The higher the slider value, the higher the smoothing of the outgoing CC value. Default: 98%

Light Sensor	
Automatic Brightness Adaption	If enabled, the minimum and maximum measured brightness corresponds to full scale input brightness range. Thus, the brightness limits are dynamically adapted to the environment. Default: On
Brightness Range	Corresponds to the fixed input brightness range. Default: 0% - 50%
Sensitivity	Configures the sensitivity of the light sensor (i.e. integration time). Note: The sensitivity parameter affects the latency. The higher the sensitivity, the higher the latency. Default: 25%

Random Mode	
Level of Randomness	If IO-Lights is not synchronized to an external clock, this parameter controls the time variation between two random LED states. The level of randomness parameter affects the average time and its variance. The higher the randomness, the lower the average time and variance. Default: 60%
Number Active	Corresponds to number of active LEDs. If set to 0, the number

LEDs	of active LEDs is random for each state. Default: 0 (= Random)

Buttons	
Touch Notes	If enabled, the touch buttons can be used individually to play a single note (with fixed velocity). Default: Off
Button A Note	Fixed note, which is sent when pressing Button A. Default: C1
Button B Note	Fixed note, which is sent when pressing Button B. Default: D1
Touch Controller	If enabled, the touch buttons can be used individually to control a parameter via MIDI Control Change. Default: Off
Button A Controller	Button A Controller index. Default: 1 (Mod Wheel)
Button B Controller	Button B Controller index. Default: 11 (Expression)

Extras	
Clock Sync	If enabled, IO-Lights will synchronize to an external MIDI clock and send the current brightness value in regular intervals. The synchronization becomes active (including start/stop) as soon as an external clock tick has been received. Default: On
Clock Divider	Sets the clock divider (1 - 96). Default: 16 => 1/16
Note Hold	If enabled, a note will be only sent if it has changed during the last clock cycle. If disabled, a note is always sent in regular intervals based on the external clock. This option is only used, if

	clock sync is enabled. Default: On
Persist Button	Saves the current configuration parameters to IO-Lights internal flash memory to make them available after reboot. After successful persistation, a short LED animation appears (clockwise circle animation).
Upgrade Button	If a new firmware is available on our server, the upgrade button can be clicked, which causes IO-Lights to switch to upgrade (DFU) mode. After selecting the device (STM32 BOOTLOADER), the recent firmware is transmitted from our server directly to IO-Lights. After successful upgrade, IO-Lights will automatically reboot and show the LED startup animation. Note: The internal settings are set to factory defaults after each upgrade.

Firmware Update

IO-Lights firmware can be easily updated with any web browser supporting WebMIDI and WebUSB in our web app (instrumentsofthings.com/io-app). The current firmware version of attached IO-Lights is shown at the bottom of the web app. If a new firmware is available on our website, the new version number is shown at the bottom.

Caution! Do not disconnect or turn off your host device during upgrade! IO-Lights will automatically reboot after successful upgrade (startup LED animation appears).

Note: IO-Lights settings are set to factory defaults after each upgrade.

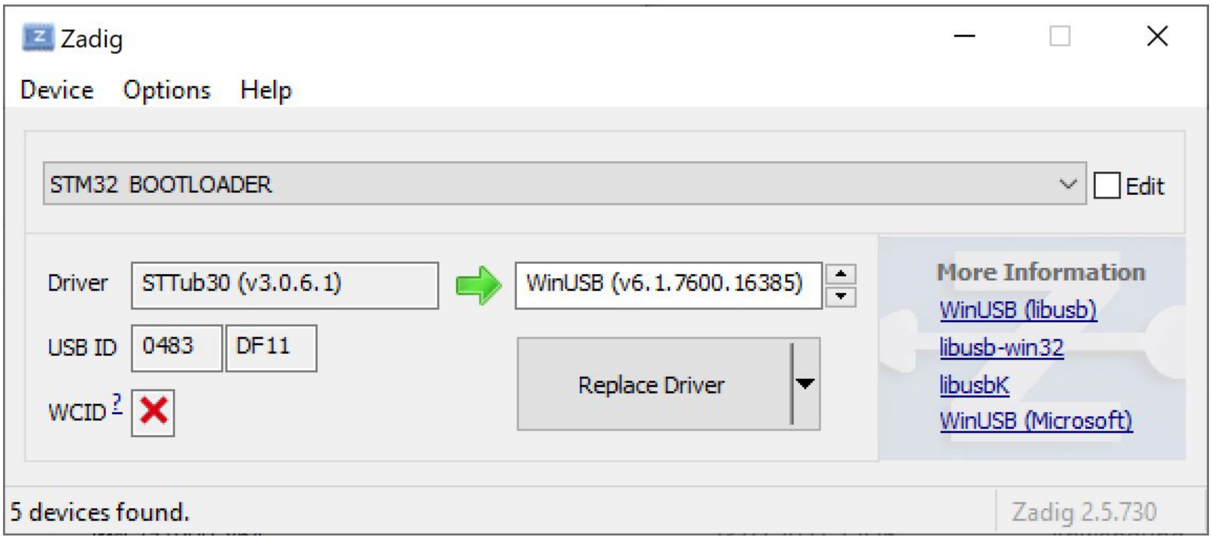
After clicking the Upgrade Button, an alert appears which has to be confirmed. Afterwards IO-Lights switches into upgrade (DFU) mode. Thus, IO-Lights does not appear as a MIDI device anymore. In a new dialog window, you have to select ***STM32 BOOTLOADER*** as device and confirm. Now the new firmware from our website is directly written to the flash memory of IO-Lights (takes a few seconds). After successful upgrade, IO-Lights automatically reboots and shows a startup LED animation.

Note: Windows needs additional drivers for firmware upgrades

Firmware upgrades do not require any additional drivers for Mac OS and Linux. Unfortunately, the default Windows USB drivers do not support DFU though. To be able to upgrade IO-Lights firmware on Windows, we recommend the free tool Zadig (https://zadig.akeo.ie/), which automatically installs the correct driver.

1. Connect IO-Lights to your (Windows) computer, open IO-App and click the upgrade button (confirm the alert dialog, to set IO-Lights to upgrade mode).
2. Open Zadig tool
3. Click on Options -> List all devices
4. Select ***STM32 BOOTLOADER*** from the dropdown list
5. Click on Replace Driver and wait until installation has finished

Now everything is set up to easily upgrade IO-Lights firmware on Windows.



STM32 BOOTLOADER

Driver STTub30 (v3.0.6.1) → WinUSB (v6.1.7600.16385)

USB ID 0483 DF11

WCID ? ✗

5 devices found.

Zadig 2.5.730

Change Log

v1.1

- Improves touch button robustness

v1.0

- Initial firmware