

# Photon controller user manual

Version 1.0 – 18.12.2019

## Term description

Shortpress – press the button for 50 ms to 1 s.

Longpress – press and hold the button between 1 s and 2 s.

Device config – configuration file located in the controller memory where user can rewrite device parameters. The file is accessible by connecting controller to PC via USB.

## Power on

Three shortpresses in a row wake up the controller from off state. Shortpresses must be closer than 500 ms to each other. After the wake-up, the control light is yellow for a short time (less than 1 s) until controller wakes up to standby state. In standby state, the control light is green first. After a short time, the control light changes its color according to its wifi state. The color mapping is shown in Table 1.

<b>color</b>	<b>signaling</b>	<b>state</b>
green	lit	standby with disabled or offline wifi
magenta	lit	standby with wifi MASTER
blue	lit	standby with wifi MASTERSLAVE
light-blue	lit	standby with wifi SLAVE
orange	flashes	firmware download
yellow	lit	controller is starting
yellow	flashes	shutting down or button stuck - controller shut-down after release
magenta	flashes	controller USB is connected to PC in MSC state (file transfer state)
green	flashes	charging is finished
red	flashes	slow charging (preparation for fast charging)
red	lit	fast charging
white	lit	error
no color	-	program is running or controller off

Table 1 - Control light states

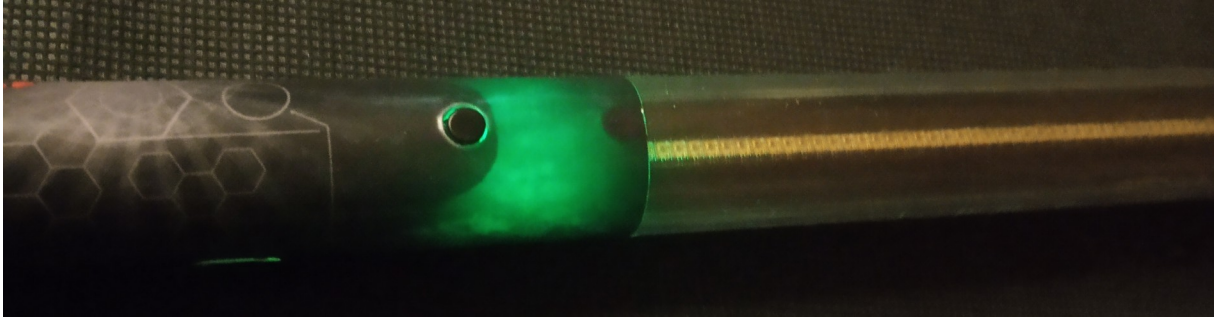


Figure 1 - Standby state

## Power off

Controller can be powered off by holding the button for longer than 5 seconds. After this time, control light start flashing yellow. Release the button and the controller turns off. Another way to power off the controller is waiting for the standby timer expiration. The standby timer runs only in the standby state and can be set in *device.conf* configuration file Figure 9. In power off state, controller draws a current up to 50 uA so during long storage time without charging it can discharge complete battery in two or three years. For this reason we recommend to avoid such a long storage time and charge the battery at least every six months. Longer time without charging can cause a shortening of the battery lifetime.

## Program selection

In standby state, press and hold the button. Controller now toggles between *program select bar*, brightness select bar and battery state indicator.

First the led bar shows a *program select bar shown in Figure 2*. Release the button to stop toggling to another menu. Then user can increment program number on each shortpress. After selecting the right program, confirm the selected program by longpress. Controller returns to standby state where program can be started by shortpress. Selected program is written into non-volatile memory and persists after controller shutdown.

Programs are numbered in alphabetical order according to their filenames. Numbers have bigger priority than letters e.g.: *001\_show.pix* is before *show.pix*.

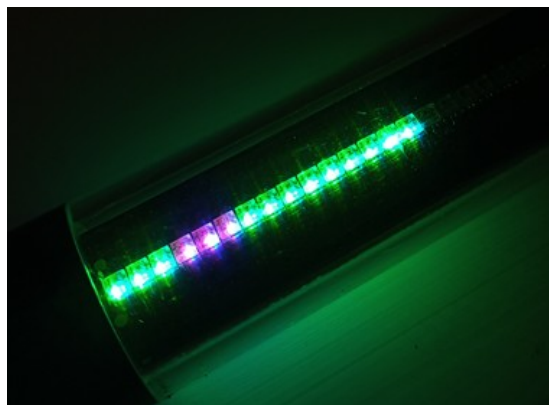


Figure 2 - program select bar

## Brightness adjustment

In standby state, press and hold the button. Controller now toggles between program select bar, brightness select bar and battery state indicator. Brightness select bar is the second menu shown when holding the button.

Release the button to stop toggling to another menu. Then user can increment brightness level by shortpress. After selecting, confirm the selected brightness by longpress. Controller returns to standby state. Selected brightness is written into non-volatile memory and persists after controller shutdown.

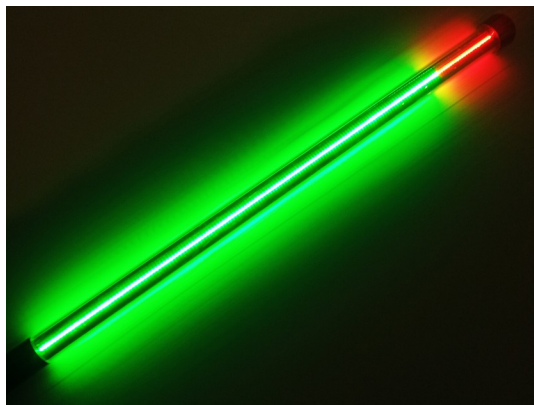


*Figure 3 - brightness adjustment*

## Battery state indicator

In standby state, press and hold the button. Controller now toggles between program select bar, brightness select bar and battery state indicator. Battery state indicator is the third menu shown when holding the button.

Green to red ratio shows the battery charge status. Complete green means battery full charge, complete red means battery discharged.



*Figure 4 - battery state indicator*

## Running the program

The program is started by shortpress in standby state. Controller runs the last selected program. On wake-up from off state controller loads the last selected program number from the non-volatile memory. During program run, the control light is disabled to avoid disturbing the show. When program ends, controller moves to standby state and the control light is enabled again.

Program can be interrupted by longpress. On longpress, controller moves to standby state immediately. To avoid an unintended interruption of the program, shortpress has no influence on the program run so hit the button accidentally does not stop the program.

During program run when battery is almost discharged, the brightness level is decreased to level 1 to prolong the show while inform the performer about the almost discharged battery state. Note that the prolonged time heavily depends on the actually shown picture and can vary between 10 s to a couple of minutes.

## Uploading program to controller

Program is created by IGNIS Studio application which can be downloaded at <http://ignisshop.com>

Program is a file of the IGNISSHOP proprietary format with “.pix” suffix. It can be uploaded to the controller manually or via the dedicated “Export” tab in the IGNIS Studio. To upload the program, connect the controller USB to the PC via a micro USB cable. Use a cable with data wires (some of charging cables have power wires only and data wires are not connected). The control light starts flashing magenta. It means that the controller is connected to PC and waits for data.



Figure 5 - Micro USB cable

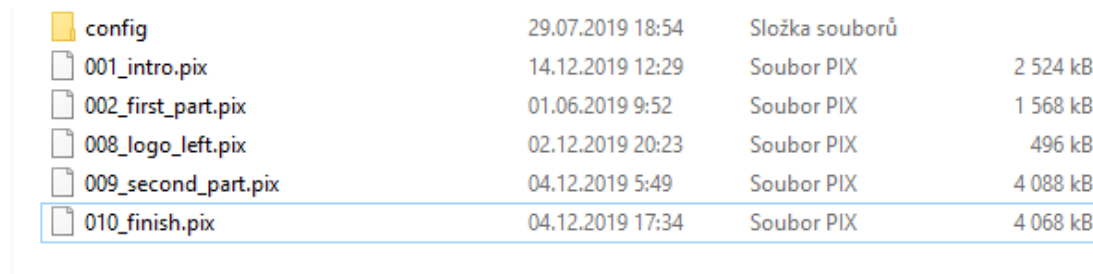
Windows then show the “Found new hardware” wizard (first time only). The controller is standard MSC device (like USB flash drive) so no driver is needed. Wait for windows to install its native driver. After installation the device is shown in the windows “device manager” under “Disk drives” like “IGNISSHOP.COM Photon Prog Mem”.

Then the .pix file containing program can be uploaded by IGNIS Studio or by drag and drop in windows explorer. All programs must be copied to the main root folder. **Do not delete or rename the “config” folder which is also in the main root of the memory.**

When program is uploaded by IGNIS Studio, the prefix number is added automatically to the filename. For example, program named *Intro* is uploaded as *001\_Intro.pix* file. The *001* prefix is selected automatically according to the highest prefix actually found in the memory, so the program is put at the end of the list e.g.: if highest program found in the memory is *015\_End.pix*, the *Intro*

program is uploaded by IGNIS Studio like 016\_Intro.pix. User can change the order in IGNIS Studio Export tab by clicking up/down arrows.

When program is uploaded by drag and drop or any other way, user must take care of the right prefixes by itself and change name of the file manually (select *rename* and add the right prefix followed by underscore “XXX\_”). Controller also runs programs without the prefix, but these cannot be synchronized via wifi (because wifi sync commands uses the numbers in prefixes).



config	29.07.2019 18:54	Složka souborů	
001_intro.pix	14.12.2019 12:29	Soubor PIX	2 524 kB
002_first_part.pix	01.06.2019 9:52	Soubor PIX	1 568 kB
008_logo_left.pix	02.12.2019 20:23	Soubor PIX	496 kB
009_second_part.pix	04.12.2019 5:49	Soubor PIX	4 088 kB
010_finish.pix	04.12.2019 17:34	Soubor PIX	4 068 kB

Figure 6 - Controller internal memory content

After uploading programs to the controller (by IGNIS Studio or any other way), click to “safely remove device” icon to finish a potential incomplete copy to the controller memory (Figure 7). **Note: Disconnecting the controller USB without the safe remove may cause controller internal memory malfunction, loss of data or harm the device.**

After safe remove is done, USB can be disconnected. After USB disconnecting, controller turns off. Triple click the button to start it again if necessary.

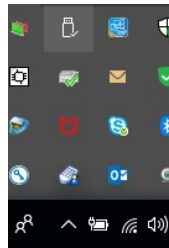


Figure 7 - Safely remove device

## Controller settings

Controller has a device settings stored in a configuration file in the internal non-volatile memory. The configuration is in file “device.conf” stored at “config” folder. The folder can be accessed by the same procedure like in “Uploading program to controller” chapter.

Parameters shown in Table 2 are written in the “device.conf” text file shown in Figure 8 and can be changed by rewriting in a text editor. Use a standard simple text editor like notepad, notepad++, PSPad. Note that some text editors adds a special non-visible characters to the text which can cause problems to controller during parameter read. Controller parser looks for {0x0D, 0x0A} end of line sequence. Some text editors or operating systems can use a different line termination and must be properly set then.

Character semicolon “;” marks a comment. It means that all the text after this character is skipped until end of the current row. Whitespaces are omitted. Parameter names must be written precisely according to Table 2. Whitespaces in names are not allowed. These are replaced by underscore “\_” character. Do not write to the file any other text except of necessary comments. Each line must be terminated by end of line sequence described above.

Open the “device.conf” file in a text editor, change the value and then save it. **After finishing changing of parameters, the controller must be safely removed from the USB bus like in the mentioned chapter.**

parameter name	type	default value	min	max	Description
wifi_mode	integer	1	1	4	Set the wifi mode. 1=disabled, 2=master, 3=slave, 4=masterslave
wifi_ssid	string	-	-	-	The wifi SSID (network name) which the controller connects to (max length is 31 characters)
wifi_password	string	-	-	-	The wifi password (max length is 31 characters)
wifi_group	integer	1	1	255	The wifi group where controller sends and receive commands. Each controller can be in one group only.
standby_timer_s	integer	180	10	9999	Time to stay in standby state before switch off the controller. The value is in seconds. NOTE: Settings this value to high may lead to discharge the battery when controller is not switched off manually.
control_led_brightness	integer	4	1	31	Brightness of the control led. NOTE: High values can cause faster battery discharging. In warm environments, do not set the value above 16.

Table 2 - Device configuration parameters

```
[Wifi]
wifi_ssid = wifi_ssid_name
wifi_password = password
wifi_mode = 4 ; 1 = disabled, 2 = master, 3 = slave, 4 = masterslave
wifi_group = 1

[Timers]
standby_timer_s = 180 ; Time how long the device stays in the standby mode before sleep
```

Figure 8 - device.conf file

## Wifi synchronization

Controller uses wifi standard 802.11 b / g / n and can be connected only in wifi client mode (does not transmit its own wifi network, but connect to an existing one). Controller acts as a DHCP client so a DHCP server must be enabled at the Access Point (AP) which controller connects to.

To connect the controller to a wifi network first switch on an AP and write its SSID name and password to the controller "device.conf" configuration file (described in chapter Controller settings). AP can be a hardware based or a smartphone software (known as wifi tethering). This way you can create a wifi network by enabling this function in your smartphone. When properly connected to a wifi network, the control led in standby state change its color according to wifi mode settings described in Table 1.

Wifi modes:

- MASTER – controller sends commands to other controllers in the same wifi group, but does not listen to anyone.
- MASTERSLAVE – controller sends commands to other controller in the same wifi group and listen to commands from the same wifi group.
- SLAVE – controller does not send any commands but listen to commands from the same wifi group.

**NOTE: Set at most one MASTER in each wifi group. Otherwise SLAVES and MASTERSLAVES will be confused when two MASTERS sends its command at the same time or use a MASTERSLAVE instead of a MASTER when multiple masters are needed.**

```
wifi_mode = 4 ; 1 = disabled, 2 = master, 3 = slave, 4 = masterslave
wifi_group = 1

[Timers]
standby_timer_s = 180 ; Time how long the device stays in the standby mode before sleep
```

Figure 9 - *device.conf* configuration file

When user starts a program by shortpress in standby state and controller is set as MASTER or MASTERSLAVE, it starts sending commands. Command holds information about program number and actual program time. Program number is taken from the program prefix e.g.: when user runs program named "005\_Jane left show.pix", the controller sends command which says "run program number 5". Then all SLAVES or MASTERSLAVES looks to their internal memory and seeks for any program with "005\_" prefix. The rest of the filename is skipped. So only a program prefixes are synchronized together and multiple users can arbitrary name their programs. E.g.: program "012\_Jane blue intro.pix" is synchronized with "012\_John\_left.pix". When program filename has no prefix (not uploaded by IGNIS Studio), no synchronization is available.

Controllers are synchronized on picture basis, not on picture line basis (pictures change in the same time, but actual line of the picture may differ). First picture after start of synchronization can be shifted about hundreds of millisecond to one second. Synchronization then takes about 1-3 seconds to precisely synchronize time in all controllers so if the next picture comes after this time, it is precisely synchronized once begins. If precise synchronization of the show from first picture is needed, it is necessary to start all synchronized programs with a black image (void place at the timeline in IGNIS Studio) for at least 3 seconds. Then first picture can begin on the timeline and it is precisely synchronized.

MASTER and MASTERSLAVE controller sends its synchronization commands all the time when program is running so other SLAVE or MASTERSLAVE controller can be powered on during program run and synchronize to already running program once its wifi connect to the AP.

If two programs with a different length are synchronized, then following behavior is observed:

- Master has longer program than slave – after end of the slave program, slave returns to its standby state, but it is unable to run any other program until the master program finishes.
- Slave has longer program than master – after end of the master program, slave continues with the current program until it ends or until master starts any other program.

## Picture stabilization

Controller is equipped by inertial unit with accelerometer, gyroscope and electronic compass. These tools are used together to predict the staff position and the rotation speed. The information is then used to stabilize picture to the same position according to the rotation. In IGNIS Studio user can enable this feature by “accelerometer” button and then set other parameters such a picture period.

The position recognition algorithm has a delay of 1-2 turns. So when user starts spinning the staff, the picture become synchronized after these spins. As more precisely user spins the better the algorithm can predict the position. If precise synchronization to spinning is necessary from the first spin of the picture, the user must start spinning 2 turns prior the picture change.

## Battery charging

To charge the battery, connect the standard wall charger to controller micro USB connector. Control led starts lit red. Max charging current is constrained by controller to 1.5 A, but it can be charged also by stronger or weaker charger. Weaker charger prolongs charging time.

If control led starts flashing red after charger connect, controller signalizes a slow charging which precedes to the fast charging. Slow charging prepares battery for fast charging and fast charging starts immediately when it is done. The red light then change from flashing to lit.

Once charging is finished, control led flashes green and the USB can be disconnected. After disconnecting the USB, controller turns off.

When controller is connected to PC as a storage device, battery is also charged, but maximum current draw from USB port is constrained to 500 mA. So this charging is very slow in comparison with charging with a standard wall charger.



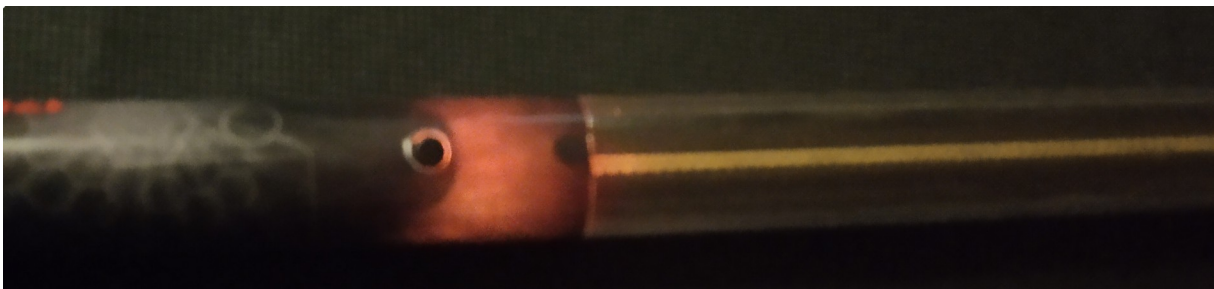


*Figure 10 - Battery charging*

## Firmware update

Controller firmware can be updated at customers. It is done by downloading a .hex file and uploading to the controller via USB (MSC). Updating firmware is described in following steps.

- Connect controller to PC via micro USB cable.
- Control led starts flashing pink color and PC found the controller memory like an external storage device
- Open the controller memory by windows explorer or some similar file transfer application
- Copy the .hex file with the new firmware to the “**config**” folder located in the controller memory. Controller looks for the firmware file here in this folder on every startup.
- Disconnect the USB, controller shuts down.
- Start controller by triple shortpress described in “Power on” part of this manual.
- Now control led starts flashing orange. The firmware is uploading. This can take about two minutes.
- Once uploading is done, control led shows the uploading result. Green color for 5 s means success (new firmware was uploaded properly), red color for 5 s means failure (new firmware was not uploaded).
- After that controller is powered off.
- Tripple shortpress can start the controller again with the new firmware if uploading was succeeded.



*Figure 11 - Orange flashing during firmware upload*

## Technical data

<b>Parameter</b>	<b>value</b>	<b>unit</b>
Current consumption (wifi disabled)	200	mA
Current consumption (wifi enabled)	300	mA
Current consumption (power off)	50	uA
Max charging current	1500	mA
Non-volatile memory size	512	MB
Wifi standard	2.4 GHz, 802.11b/g/n	-
Wifi encryption support	WPA, WPA2	-
Accelerometer range	+ - 16	g
Gyroskop range	+ - 2000	dps
Magnetometer range	+ - 16	gauss
Operating temperature range	-20 to 85	°C