



Canon® EF/EF-S Lens Controller (LC-2) Application Program Interface

Programmers Reference Manual
Copyright © March 2018
Version 1.7.4

Notice

Copyright © 2018 Innovative Scientific Solutions Inc. All rights reserved.

ISSI does not warrant that the LC-2 API will function properly in every hardware/software software environment. This software is inherently complex, and users are cautioned to verify the results of their work.

ISSI has tested the software and reviewed the documentation. ISSI MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS SOFTWARE OR DOCUMENTATION, THEIR QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS SOFTWARE AND DOCUMENTATION ARE LICENSED "AS IS" AND YOU, THE LICENSEE ARE ASSUMING THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE.

IN NO EVENT WILL ISSI BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, even if advised of the possibility of such damages. In particular, API shall have no liability for any programs or data stored or used with ISSI software, including the costs of recovering such programs or data.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1.Redistributions of source code must retain the above copyright notice, this list of conditions and the above disclaimer.
- 2.Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the above disclaimer listed in this license in the documentation and/or other materials provided with the distribution.
- 3.Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this API without specific prior written permission.

ISSI Contact Information ***Innovative Scientific Solutions Inc.***

7610 McEwen Road
Dayton, OH 45459
Ph.: (937) 630-3012
Fax: (937) 630-3015
Website www.psp-tsp.com
Email : issi-sales@innssi.com

ISSI EF/EF-S LENS CONTROLLER-2 API

(API version 1.7.4)

API based on UDP protocol, 1339/UDP port is used for communication.

Action: Get controller f/w version

Command: "ver" (hex: 76:65:72)

Answer: "ISSI LC-2=1x.y (s/n:150xx)"

Action: Change controller IP address to 192.168.1.2

Command: "ChangelP=192.168.1.2"

Answer: no answer, controller will set IP and restart

Action: Initialize lens, and get current motors values and ranges (aperture ranges would change when zoom changed)

Command: "ping" (hex: 70:69:6e:67)

Answer: contains 7 packets:

#1 "zRange=minX,maxX"

where minX- minimum zoom value, maxX – maximum zoom value

#2 "fRange=minZ,maxZ"

where minY- minimum focus value, maxY – maximum focus value

#3 "aRange=minY,maxY"

where minZ- minimum aperture value, maxZ – maximum aperture value

#4 "Current=XX,YY,ZZ"

where XX- current zoom value, YY- current focus value, ZZ- current aperture value

#5 "AF=X"

where X =1 if AF selected on lens and 0 if MF selected.

#6 "IS=X"

where X=1 if Image Stabilization(IS) switch is enabled, 0 if disabled or lens have no IS

#7 "ISactive=X"

where X=1 if IS is activated and 0 if it deactivated

"nolensfound" - in case no lens found

Action: Move Focus motor on X units

Command: "moveFocus=X"

where X could be as positive as negative values.

Answer: "Focus=Y", where Y – current value of Focus motor
or "errorFocus" when focus value is not reachable;

Action: Move Aperture motor on X step (one-quarter-stop f-number)

Command: "moveAper=X"

where X could be as positive as negative values.

Answer: "Iris=Y", where Y – current value of Aperture
or "errorAper" (in old f/w) and "errorAperLimits" when aperture value is not reachable

Action: Set desired value for Focus motor

Command: "setFocus=X"

where X is positive value

Answer: Focus =XXX, where XXX is the current value, and next packet "focusDone" - indicate that desired focus is set (for non USM lens could take more time)

Action: Set desired value for aperture (in f-number)

Command: "setAper=X"

where X is in quarter-stop f-number scale, please use pre-calculated f-stop numbers: [1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.7, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.1, 3.4, 3.7, 4.0, 4.4, 4.8, 5.2, 5.6, 6.2, 6.7, 7.3, 8.0, 8.7, 9.5, 10.4, 11.3, 12.3, 13.5, 14.7, 16, 17.4, 19, 20.7, 22.6, 24.7, 26.9, 29.3, 32.0, 34.9, 38.1, 41.5, 45.3, 49.4, 53.8, 58.7] for firmware 1.7.3 and earlier, and [1.0, 1.1, 1.3, 1.4, 1.6, 1.8, 2.0, 2.2, 2.5, 2.8, 3.2, 3.5, 4.0, 4.5, 5.0, 5.6, 6.3, 7.1, 8.0, 9.0, 10, 11, 13, 14, 16, 18, 20, 22, 25, 29, 32, 36, 40, 45, 51, 57, 64, 72, 80, 90]

for f/w versions after 1.7.4.

Answer: "Iris=Y", where Y – current value of Aperture, if X is out of range will answer with "errorAper-Limits" message

Action: detect IS (Image Stabilization) availability

Command: "isIS"

Answer: "IS=Y"

where Y could be 0 or 1, "0" - means lens does not have IS function, "1" - lens have IS function. Also will answer "0" if IS switch is in OFF position.

Action: activate IS (Image Stabilization) function

Command: "enableIS=X"

Answer: "ISactive=Y"

where X is the amount of seconds to keep IS active, from [0-3600]. "0" - disable IS, and Y is 0 or 1 - current IS status

Action: Set nickname for LC (stored in LC memory)

Command: "setName=XXXXXXX"

where XXXXXXX is 7-symbols name for this controller.

Answer: no answer

Action: Get LC nickname (stored in LC memory)

Command: "getName"

Answer: "NAME=XXXXXX"

Action: soft restart LC-2

Command: "Reboot" (hex: 52:65:62:6f:6f:74)

Answer: OK

Action: hard restart LC-2

Command: "Restart" (hex: 52:65:74:73:74:61:72:74)

Answer: OK

Action: explore focus limits

Command: "refRange"

Answer: "fRange=minY,maxY"

where minY– minimum focus value, maxY – maximum focus value

Action: get lens name

Command: "getLens"

Answer: "Lens=XXX", where XXX is the lens name stored in internal lens memory, lens should support this command.

Action: alternative way to set lens focus, lens will move focus motor to minimum position, reset encoder counter and then position the focus. Helps to suppress accumulative lens drift effect for some lens.

Command: "setFocus2=X"

where X is positive value

Answer: Focus =XXX, where XXX is the current value, and next packet "focusDone" - indicate that desired focus is set (for non USM lens could take more time)

PERL EXAMPLE SCRIPT – SEND COMMAND

```
#!/usr/bin/perl -w
use IO::Socket;
use strict;
my $sock = IO::Socket::INET->new(
    Proto => 'udp',
    PeerPort => 1339,
    PeerAddr => '192.168.2.252',
) or die "Could not create socket: $!\n";
my $size = $sock->send($ARGV[0]) or die "Send error: $!\n";
print "$size bytes sent\n";
my $response = "";
$sock->recv( $response, 1024 );
print "Response: $response";
$sock->close();
```

Usage: ./udpsend.pl setFocus=1000

will send command "setFocus=1000" to 192.168.2.252 to port 1339/UDP

PYTHON EXAMPLE SCRIPT – SEND COMMAND

```
import socket
UDP_IP = "192.168.2.252"
UDP_PORT = 1339
print ("ISSI :: Canon Python Script Example\n")
MESSAGE = "setFocus=1000"
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # UDP
sock.sendto(MESSAGE.encode(), (UDP_IP, UDP_PORT))
try:
    while True:
        data, addr = sock.recvfrom(1024)
        if not data: break
        print ("Received:", data)
except socket.error:
    print("")
sock.close()
```

Usage: ./udpsend.py

will send command "setFocus=1000" to 192.168.2.252 to port 1339/UDP