Appendix

ISSI LC-1S :: API v.3.0.1

Based on UDP protocol, 1337/UDP port is used for communication.

Action: Get controller f/w version
Command: "ver" (hex: 766572)
Answer: "ISSI LC-1s=3.0.1 (s/n:19001)"

Action: Change controller IP address to 192.168.1.2
Command: "ChangeIP=192.168.1.2"
(hex: 4368616e676549503d3139322e3136382e312e32)
Answer: no answer, controller will set IP and restart

Action: Get current motors values
Command: "Current" (hex: 43757272656e74)
Answer: "Current=XXX,YYY,ZZZ"
Where XXX – zoom position value, YYY- focus position value, ZZZ – iris position value.

Action: Move Zoom Narrow during X ms
Command: "ZoomN=X"
Answer: "Zoom=Y", where Y – current value of Zoom motor

Action: Move Zoom Wide during X ms
Command: "ZoomW=X"
Answer: "Zoom=Y", where Y – current value of Zoom motor

Action: Move Focus Far during X ms
Command: "FocusF=X"
Answer: "Focus=Y", where Y – current value of Focus motor

Action: Move Focus Near during X ms
Command: "FocusN=X"
Answer: "Focus=Y", where Y – current value of Focus motor

Action: Move Iris Open during X ms
Command: "IrisO=X"
Answer: "Iris=Y", where Y – current value of Iris motor

Action: Move Iris Close during X ms
Command: "IrisC=X"
Answer: "Iris=Y", where Y – current value of Iris motor

Action: Set zoom motor value to X
Command: "setZoom=X"
Answer: "zoomDone", it may take a while to position the motors.

Action: Set focus motor value to X
Command: "setFocus=X"
Answer: "focusDone", it may take a while to position the motors.

Action: Set iris motor value to X
Command: "setIris=X"
Answer: "irisDone", it may take a while to position the motors.

Action: Stop all motors (during setFocus, setIris or setZoom)
Command: "setStop"
Answer: same as for 'Current" command

Action: Motors limits detection (for all tree motors)
Command: "FindLimits"
Answer: in separate packets:
zoomMax=XXX
irisMax=XXX
focusMax=XXX

Action: Set Iris mode - Motorized Iris(X=2), DC Iris(X=1) iris or Video Iris(X=0).
Command: "setIrisMode=X"
Answer: "mode=X"

Action: Move DC Iris to close
Command: "DIrisC=X"
Answer: "DIrisC=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move DC Iris to open
Command: "DIrisO=X"
Answer: "DIrisO=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move DC Iris to close faster

Command: "DIrisCX2=X"
Answer: "DIrisCX2=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move DC Iris to open faster
Command: "DIrisOX2=X"
Answer: "DIrisOX2=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move Video Iris to close
Command: "VIrisC=X"
Answer: "VIrisC=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move Video Iris to open
Command: "VIrisO=X"
Answer: "VIrisO=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move Video Iris to close faster
Command: "VIrisCX2=X"
Answer: "VIrisCX2=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Move Video Iris to open faster
Command: "VIrisCX2=X"
Answer: "VIrisOX2=OK", DC Iris lenses has no potentiometer, there is no feedback with actual position.

Action: Set nickname for LC (stored in LC memory)
Command: "setNAME=XXXXXXX" where XXXXXXX is 7-symbols name
Answer: no answer

Action: Get nickname for LC (stored in LC memory)
Command: "getNAME" Answer:
Answer: "NAME=XXXXXX"

Action: Change controller IP address to 192.168.1.2
Command: "ChangeIP=192.168.1.2"
Answer: "IPchanged" need restart to apply changes

==========

Python3 Example Script – LC-1S Send Command

==========

```
import socket
import time

# LC2 IP address
UDP_IP = "192.168.2.251"
UDP_PORT = 1337

print ("ISSI :: LC-1S Python Script Example\n")

# Command list
MESSAGES = ["ver", "IrisMode=2", "Current"]

# UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # UDP

sock.setblocking(0)
sock.settimeout(0.05)
counter = 0;
# Loop on command list
for i in range(len(MESSAGES)): # LC2 command loop
        print ("#",i+1,"Sent:", MESSAGES[i])
        start = time.time()
        sock.sendto(MESSAGES[i].encode(), (UDP_IP, UDP_PORT))
        try:
                while True:
                        data, addr = sock.recvfrom(1024)
                        if not data: break
                        print ("Received:", data)
                        end = time.time()
                        print(end - start, " seconds")
        except  socket.error:
           print("")
        time.sleep(0.15)
```

==========