

## Using the PConLan protocol to remote control an EyeVision system

The PConLan protocol is an ASCII text based protocol to operate EyeVision systems by remote control over Ethernet. As transport layer the UDP protocol is used. For most commands the EyeVision system sends a response indicating if the command was accepted or not. The PConLan reference describes all available commands. It is easily possible to control an EyeVision system with any programming language that supports basic string handling and network sockets. The programming language used as example is Python . The syntax highlighting uses gray for comments, which start with an hash mark.

### E1 remote start or stop the checkprogram

This example can be used with any checkprogram in the simplest case an empty one. If you execute this script, it will start the checkprogram inside the running EyeVision Software on the same system. By changing the value of the variable UDP\_IP the command can be sent to any other system in the same network. To remote stop the checkprogram again, simply change the value of the variable cmd from 002 to 003.

```
import socket

#The IP address of the EyeVision system
UDP_IP = '127.0.0.1'
#Standard PConlan udp port
UDP_PORT = 5952
#PConLan command
cmd = '#002#' # start
#cmd = '#003#' # stop

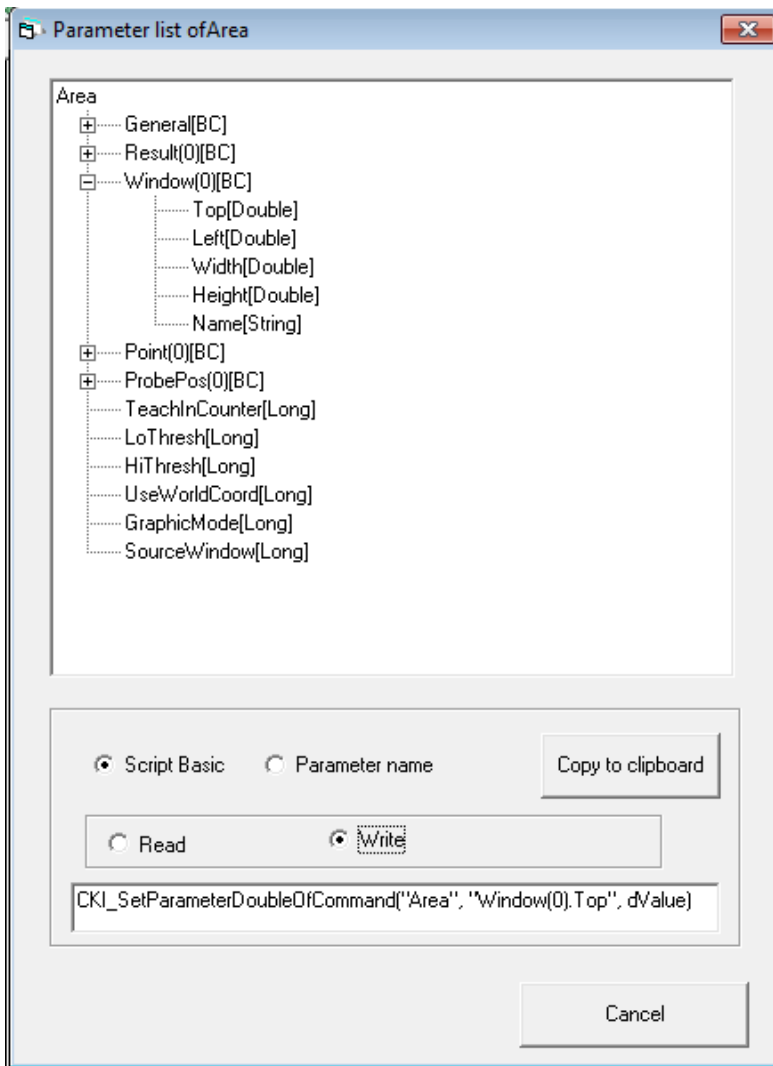
#Open socket
sock = socket.socket(socket.AF_INET,
                     socket.SOCK_DGRAM)

#Sends the command
sock.sendto(cmd, (UDP_IP, UDP_PORT))
print 'command send to the EyeVision system : ' + cmd

sock.settimeout(5) # 5 sec
#Read the response on the socket
resp, addr = sock.recvfrom(64) # max 64 byte
print addr, resp
```

### E2 remote change the parameter of a command inside the checkprogram

Most of the commands inside a checkprogram expose a set of parameters that are changeable. The list of parameters of a command, is in the manual of each command as subfile in the navigation tree on the left of this page. This list is also in the EyeVision software, accessed by a right click on the command icon in the program editor, context menu entry parameter list. There you also get the command line to use in the script interpreter. It has the form CKI\_SetParameterDoubleOfCommand(<CmdName>, <ParName>, <Value>). This line contains the information that is needed to build the UDP packet. The command name is the entry in the comment column of the program editor.



To test this script you can use this checkprogram : Make sure that you have selected an image with dimensions of at least 200×500. If you execute this script, it will set the y position of the evaluation area to 100 and increase it by 10 on every press of the enter key, till 400.

```
import signal
import sys
import socket

#The following four lines are to exit by pressing ctrl + c
def signal_handler(signal, frame):
    print ''
    sys.exit(0)
signal.signal(signal.SIGINT, signal_handler)

#The IP address of the EyeVision system
UDP_IP = '127.0.0.1'
#Standard PConlan udp port
UDP_PORT = 5952
#PConLan command
#CKI_SetParameterDoubleOfCommand('Area', 'Window(0).Top', dValue)
#021;<CmdName>;<ParType>;<ParName>;<Value>#

CmdName = 'Area'
ParType = '1' # 0 = String, 1 = Double, 2 = Long
ParName = 'Window(0).Top'
Value = 100
cmd = '#021;' + CmdName + ';' + ParType + ';' + ParName + ';' + str(Value) + '#'

#Open socket
```

```

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

while 1:
    cmd = '#021;' + CmdName + ';' + ParType + ';' + ParName + ';' + str(Value) + '#'
    Value = Value + 10 if Value < 400 else 100
    #Sends the command
    sock.sendto(cmd, (UDP_IP, UDP_PORT))
    print 'command send to the EyeVision system : ' + cmd
    inp = raw_input('press enter key to continue')

```

### E3 remote read a parameter of a command inside the checkprogram

The response on the port that is used to send a command, is only a receipt. To get the value a command sends back a second port is used. This functionality needs to be enabled in the Eyevision.ini file. Add or extend your ini file with the following lines:

```

[PCONLAN]
Enabled=1
Port=5952
UsePConLANResponse = 1
PConLANResponsePort= 4558
PConLANResponseIP= 127.0.0.1

```

This script sends the command to read the gain parameter from the 'Camera settings' command. The checkprogram from E2 already contains the command for testing.

```

import socket

#The IP address of the EyeVision system
UDP_IP = '127.0.0.1'
#Standard PConlan udp port
UDP_PORT = 5952
#The IP of the interface to listen for packets, empty address listens on all
UDP_IP_RECV = ''
#Standard port for extended response
UDP_PORT_RECV = 4558

#PConLan command
#CKI_SetParameterLongOfCommand("Camera settings", "Gain", lValue)
#021;<CmdName>;<ParType>;<ParName>#
CmdName = 'Camera settings'
ParType = '2' # 0 = String, 1 = Double, 2 = Long
ParName = 'Gain'
#ParName = 'Shutter'
cmd = '#023;' + CmdName + ';' + ParType + ';' + ParName + '#'

#Open socket to send command
sock = socket.socket(socket.AF_INET,
                     socket.SOCK_DGRAM)

sock.settimeout(5) # 5 sec

#Open socket to receive extended response
listenSock = socket.socket(socket.AF_INET,
                           socket.SOCK_DGRAM)

#Make it listen for packets from the defined IP
listenSock.bind((UDP_IP_RECV, UDP_PORT_RECV))
listenSock.settimeout(5) # 5 sec

#Sends the command

```

```
sock.sendto(cmd, (UDP_IP, UDP_PORT))
print 'command send to the EyeVision system : ',cmd

#Read the response on the socket
try:
    resp, addr = sock.recvfrom(64) # max 64 byte
    print 'response : ',addr,resp
except socket.timeout:
    print 'sock timed out'

#Read the extended response on the additional socket
try:
    resp, addr = listenSock.recvfrom(1024)
    print 'response : ', addr,resp
except socket.timeout:
    print 'listenSock timed out'
```