WolfVision GmbH
Oberes Ried 14
A-6833 Klaus / AUSTRIA
Tel. ++43-5523-52250
Fax ++43-5523-52249
E-Mail: wolfvision@wolfvision.com
www.wolfvision.com

WolfProt Programmer´s Guide

Table of contents

1 Introduction

WolfProt protocol: A simple and fast way to operate Cynap from your Room Management System solution.

Our WolfProt Protocol aims to offer versatility and sophisticated access when it comes to develop your next Room Management System integration.

1.1 Disclaimer

This manual is intended for Room Management System developers. Therefore it is assumed that you already have a good understanding of AMX/Crestron/Cue-system programming and integration (see http://www.howtoprogramcrestron.com/resources.html for Crestron integration or http://www.amx.com/products/NetLinxStudio.asp for AMX integration).

Our provided templates contain the whole range of Cynap functions, most of the time not all are needed for each implementation at your customers' site. We strongly advise to adapt our demonstration templates to your customer needs before installing the provided demonstration template.

Knowledge of TCP/IP networking is beneficial. It is also assumed that your processor and touch panel/terminals are installed and do function correctly and reside on the same network (Wi-Fi or Ethernet) as Cynap.

2 WolfProt Basics

Wolfprot is the WolfVision command protocol to control WolfVision devices. WolfProt is an easy accessible Protocol that allows you to control various functions on Cynap and Wolfvision Visualizers.

The connection between your Room Management System and Cynap/Visualizer uses Ethernet and TCP/IP and is password protected and first needs to be activated and configured (Enable and set Room Management System user).

The commands consist of GET and SET commands.
GET commands retrieve information from the devices; SET commands change settings or initiate a direct command.

WolfProt commands are organized in request-reply pairs (SET/GET).
They start with a header, followed by the command, the length and data.

If you send a SET command you need to follow up with the appropriate GET command to check, if the SET command got executed.

2.1 Cynap Introduction

Cynap is a collaboration device which supports a large number of resources (videos, office formats or images), several input sources (on 2 HDMI in) on up to 4 windows as well as internal functions such as Recording/Streaming and Annotation, Whiteboard or WebRTC.

2.1.1 Cynap specifications

System

Operating System	Linux
Memory	8GB
Internal storage	64GB
Output resolutions	Up to 4K UHD: 2160p60 (4.2.0), 2160p30 (4.4.4), 1080p60 (4.4.4), 1080p30 (4.4.4)
Compatible mobile device operating systems	iOS, Android, Mac OS, Windows, Windows Mobile, current HTML5 browser
Supported image file formats	GIF, JPEG, BMP, PNG
Supported document file formats	PDF, Word, PowerPoint, Excel, Text, HTML
Supported video file formats	AVI, WMV, MOV, MP4, DivX, MKV, M4V, OGV
Supported audio file formats	MP3, WMA, MKA, OGA, OGG
Supported mirroring protocols	AirPlay, Google Cast, Miracast (no HDCP support), vSolution Cast (up to 30 fps)
HDCP support	Yes, (HDCP 1.4)

Table 1: System specifications

Inputs and Outputs

Video input	HDMI x2 (HDMI 2.0)
HDBaseT 1.0 Input	x 1 [1]
Video output	HDMI x2 (HDMI 2.0)
HDBaseT 1.0 Output	x 1 [2]
Audio	Line in / Line out (3.5mm mini jacks)
USB ports	Rear USB 3.0 ports x4, front USB 2 port x1, FAT32 limited to 4 GB files

Table 2:Input/Outputs

2.1.2 Cynap functionalities

Features

Max. no. of devices simultaneously displaying content on screen	4
Max. simultaneous receiver connections via Capture app	Virtually unlimited (dependent on network infrastructure)
Wireless device mirroring	Yes
Streaming protocols	RTSP, RTP (Unicast/Multicast)
Local video recording	Yes, 1080p, 30fps
Cloud services	Yes, Box, Google gDrive, Dropbox
Access to network drives	Yes
Document and media player	Yes
Whiteboard and annotation	Yes
Presentation modes	Protected and open mode
On-screen content arrangement modes	Dynamic
Web conferencing	WebRTC
Dual screen modes	Yes
Integrated web browser	Yes
Customizable background image	Yes

Table 3: Cynap features

2.2 Cynap Network Connectivity

Wi-Fi standards	802.11 ac/a/b/g/n
Wi-Fi Bands	2.4 and 5 GHz
Data rate	Wireless up to 900Mbps
Network protocols	TCP/IP, FTP, HTTP, HTTPS, SNTP/NTP
IP configuration	DHCP, Static, network interface priority
Security encryption	WEP, WPA2, WPA Enterprise
Max. wireless coverage	Environment dependent
Ethernet/LAN port – 1 GigE	Yes
Ethernet/LAN default settings	IP address, subnet mask, default gateway, DNS server, interface priority

Table 4: Network connectivity

[1] can be used only when HDMI input 1 is not in use
[2] can be used only when HDMI output 2 is not in use

Cynap supports WolfProt only on Ethernet interface. Before you´re able to issue a command you have to make sure that you´re connected to Cynap and logged in as Room Management System user (open socket).

Cynap offers TLS encrypted connections – please make sure that you use the correct port (see Table 5: Firewall settings).

Cynap, once in standby mode, needs to be reactivated by Wake-On-LAN (sending a magic packet to Cynap´s MAC address).

2.2.1 WolfProt for Cynap and Visualizers

WolfProt has been extended and simplified for use with Cynap. It remains compatible for former and new Visualizer models.

To address a Visualizer directly, please be aware that communication runs directly over Port 50916 (re-routed and untouched by Cynap).

Cynap offers both an unencrypted and encrypted socket connection at the same time. You´re able to switch off unencrypted connections ***by enabling "Only allow secure connection"*** in the security settings in which case, the socket uses Port 50917 and blocks the connection on Port 50915.

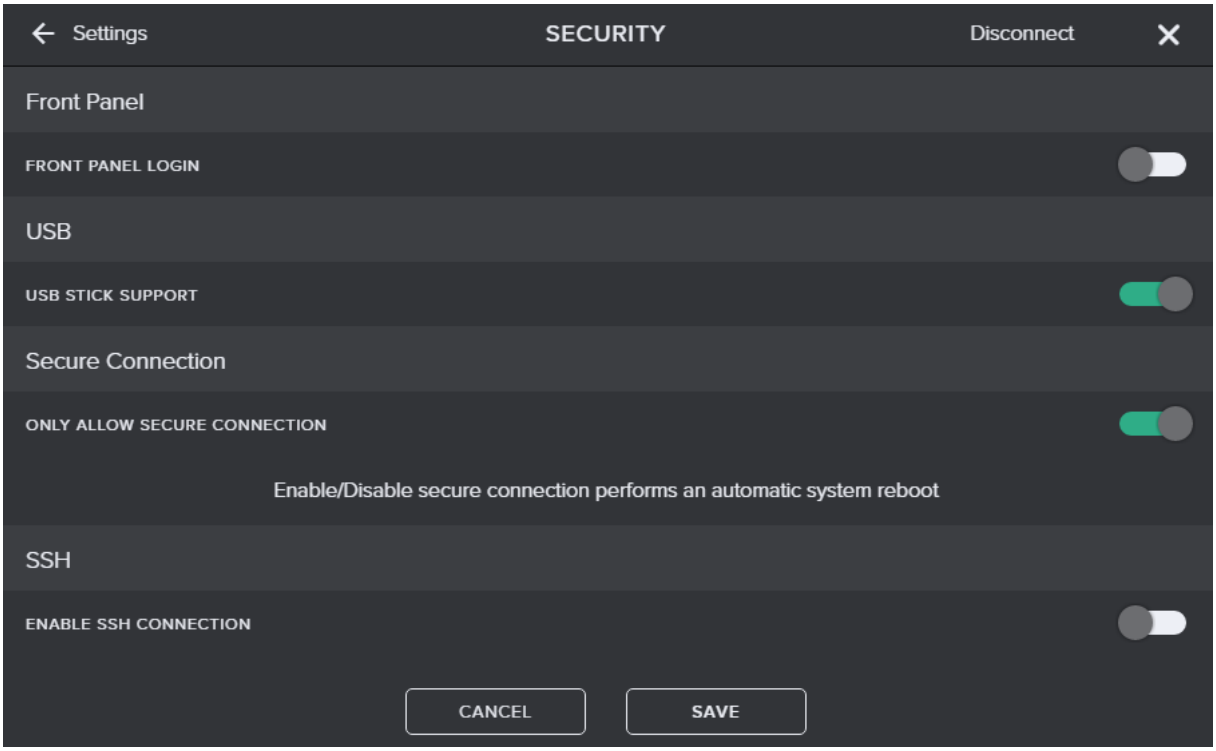


Figure 1: SSL Configuration

2.2.2 Firewall settings

Port #	Feature		Description
50913	UDP/ TCP	WolfVision Device Discovery	Device Discovery (WOL, Wake-On-LAN) Port for WolfVision devices
50915	TCP	Cynap WolfProt	Communication between Room Management System and Cynap
50916	TCP/ UDP	Visualizer WolfProt	Communication between Room Management System and Visualizer
50917	TCP	Cynap WolfProt protected by TLS	SSL encrypted connection to Cynap

Table 5: Firewall settings

Portscan
It requires a port scanner to verify if the necessary ports are available. The laptop (or other scanning device) has to be in the same network and not be obstructed by an activated firewall.

Examples
Portscanner used in examples: nmap (https://nmap.org)
Port 50916 has no Visualizers attached; therefore the service is not running and the port not responding.

UDP Scan

```
C:\Users\rgraemer>nmap --system-dns --reason -sU -p U:50913,60916 10.0.6.9
Starting Nmap 7.31 ( https://nmap.org ) at 2016-12-05 13:23
Nmap scan report for 10.0.6.9
Host is up, received arp-response (0.00013s latency).
PORT      STATE         SERVICE REASON
50913/udp open|filtered unknown no-response
50916/udp closed        unknown port-unreach ttl 64
```

TCP Scan

```
C:\Users\rgraemer>nmap --system-dns --reason -sT -p T:50915-50917 10.0.6.9
Starting Nmap 7.31 ( https://nmap.org ) at 2016-12-05 13:21
Nmap scan report for 10.0.6.9
Host is up, received arp-response (0.00034s latency).
PORT      STATE    SERVICE REASON
50915/tcp open     unknown syn-ack
50916/tcp filtered unknown no-response
50917/tcp open     unknown syn-ack
```

For additional port descriptions and networking requirements please read the vSolution Network Integration manual (available online).

The connection between your Room Management System and Cynap is authenticated by *UserType* and *password*. The session on port 50915 is unencrypted and to use an encrypted session you have to use 50917.

Visualizer connected to Cynap´s Visualizer Ethernet port: Cynap forwards your Visualizers commands without interfering directly to the Visualizer. The response from the Visualizer will be sent back directly. The connection to your Visualizer is not encrypted.

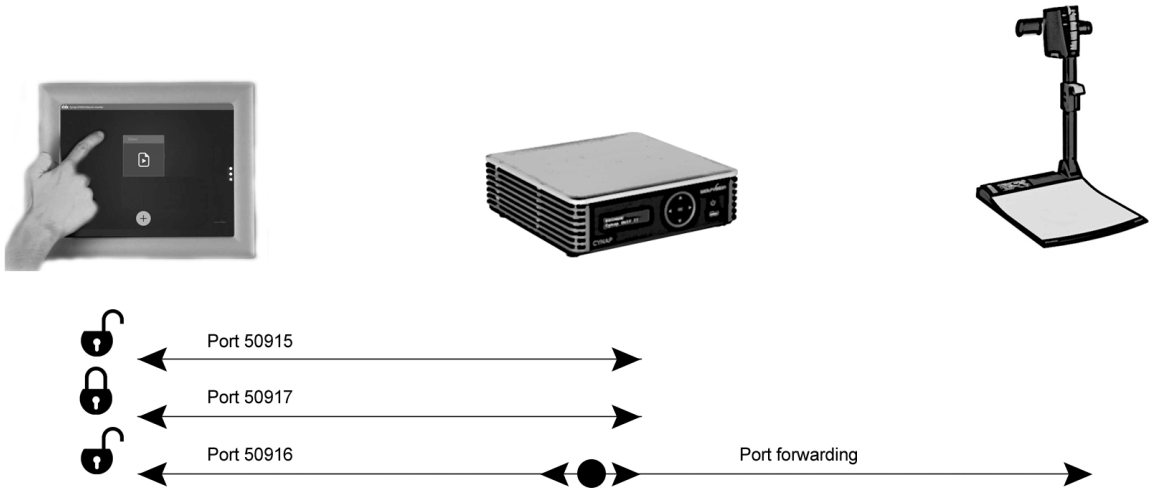
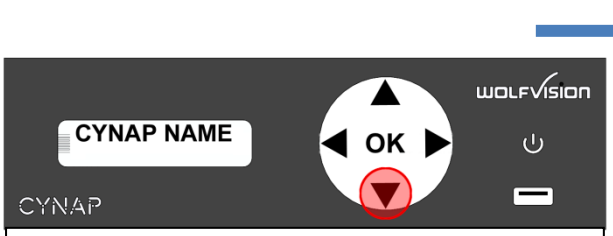


Figure 2: Room Management System Cynap connection to Visualizer (Port forwarding)

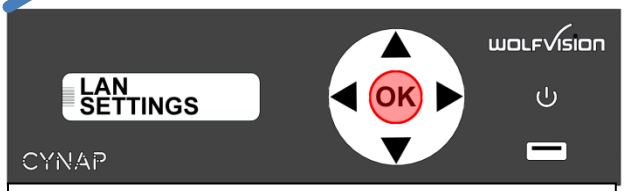
WolfVision devices are communicating on Ethernet. To successfully operate and send/ commands, some ports are required to be open.

To send and receive data between Cynap you´re able to choose between an SSL encrypted connection or not. To address a Cynap connected Visualizer you still need to use an unencrypted session on Port 50916.

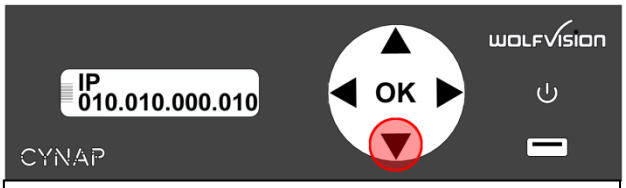
2.2.3 Get network information from the front panel



Press Down Arrow-Key until you see **LAN Settings**



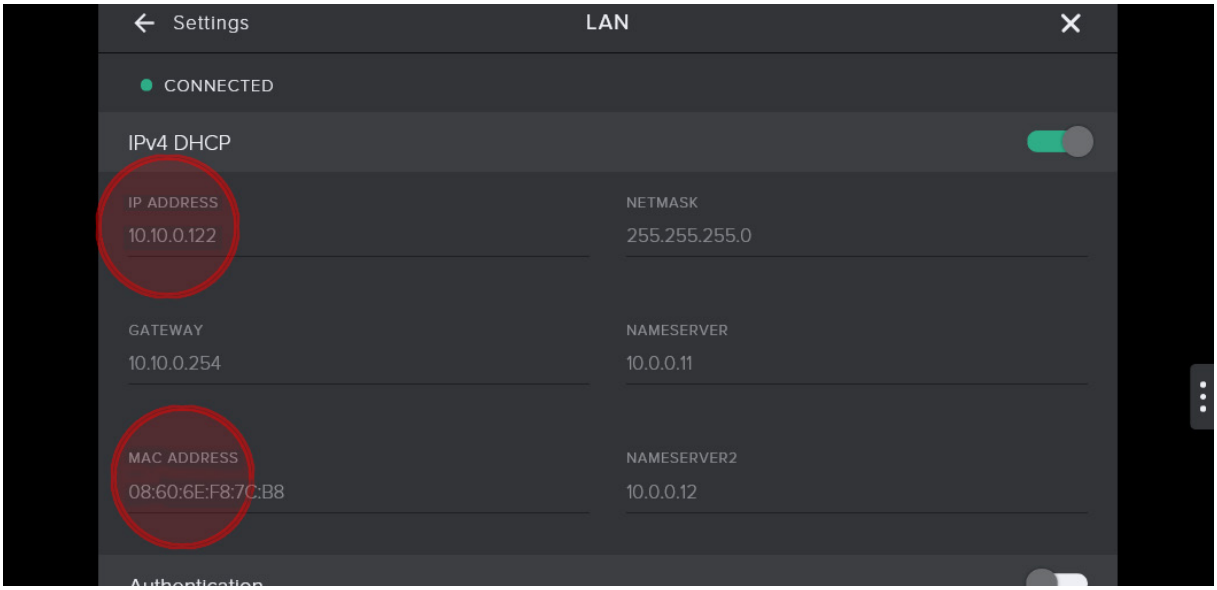
When you see LAN Settings, **press OK**



Again, press Down-Arrow-Key until you see the **IP address**

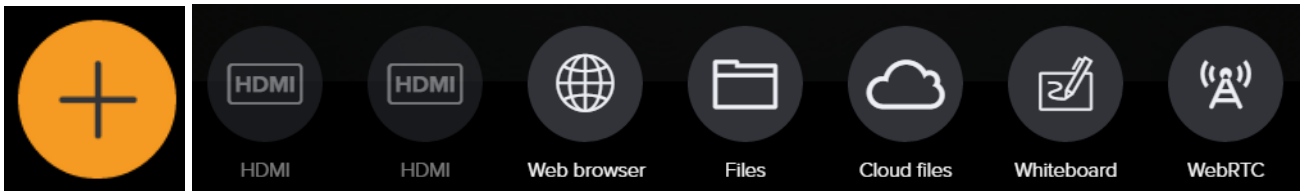
Once again, press **Down**-Arrow-Key until you can read the **MAC address**.

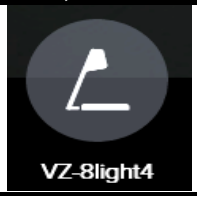
The same information can also be fetched from the Browser when you´re connected to Cynap´s Settings (Settings -> LAN).

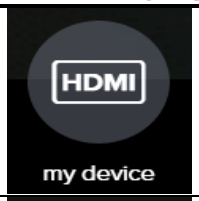
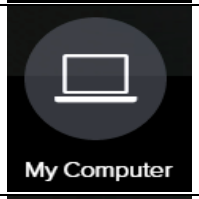
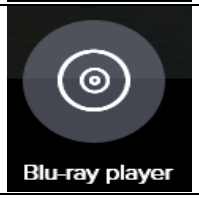


2.3 Cynap Sources

Cynap offers various applications, these applications are called sources. Each source, when pressed on the yellow circle (source button) open up in a new window (total of 4 windows/sources usable at once).



Cynap Source	Description
HDMI input 1 and HDMI input 2	Opens up a window with the connected HDMI1 or 2 input sources. Name (label) of HDMI input and icon (four types of HDMI input selectable) can be changed in the settings of Cynap. HDMI input type: WolfVision Visualizer Icon, label: 

	<table><tr><td>General HDMI</td><td>  </td></tr><tr><td>Computer</td><td>  </td></tr><tr><td>Disc Player</td><td>  </td></tr></table>
Web browser	Opens a web browser
Files	Opens the file dialog
Cloud files	Opens the Cloud file list or the login screens if not logged in (login details can be saved in toolbar menu)
Whiteboard	Opens an empty whiteboard
WebRTC	Opens a configured WebRTC session

2.4 Help/Support

We do provide a number of support documents to help you integrate Cynap into your implementation.

We are, however, unable to support you with skills in programming languages needed to build your Room Management System solution.

3. Setup Room Management System User

Figure 3: User Levels

Your room management system shares the same user level as the moderator of Cynap. The same WolfProt commands are also being used for our range of software (e.g. vSolution Link etc.).

To avoid confusion between the Cynap moderator and your Room Management System solution and offer the same level of access, we provide an additional user for your Room Management implementation.

It gives the moderator the possibility to use a randomized and changing PIN instead of a static password.

The user Room Management System offers the same moderator-user rights; but without a randomized PIN.

However, if you don´t configure the Room Management System user to protect your access to Cynap, your opened web socket connection will automatically be granted the level USER instead of just UserLevel NONE.

This convenient login feature only works as long as the Moderator USER has no password set.

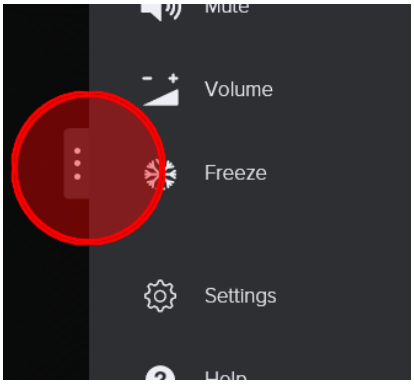
As soon as the Moderator password has been activated, your solution won´t work anymore and a login procedure or the setup of the Room Management System user will be required.

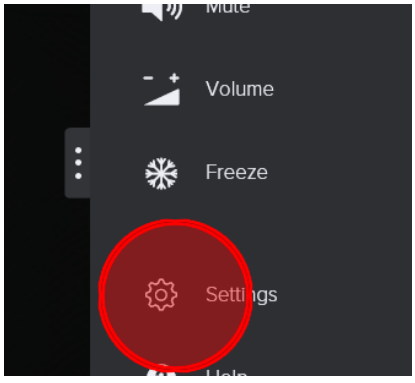
Your anonymous access will be automatically downgraded to user None and most functions will cease to work.

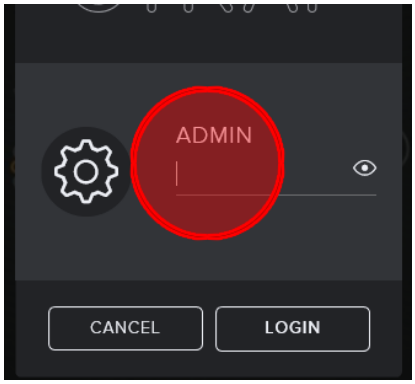
3.1 Tutorial

Start a browser and type in Cynap´s IP address to gain access to Cynap´s menu.

First click on the Cynap toolbar indicator and then click on Settings. Enter your admin password on the Login-Popup and click Login to access the Settings.

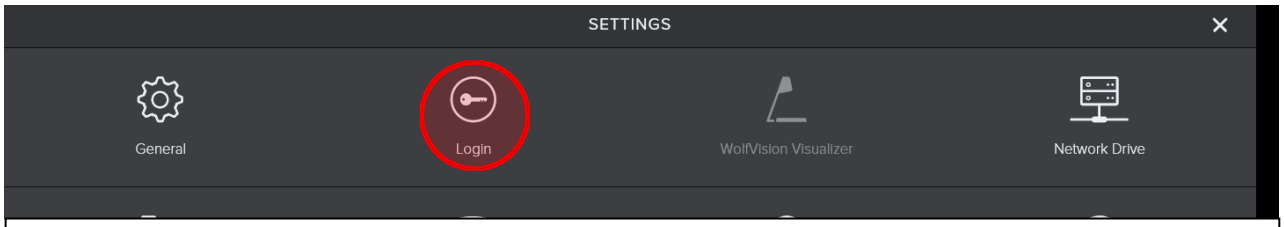






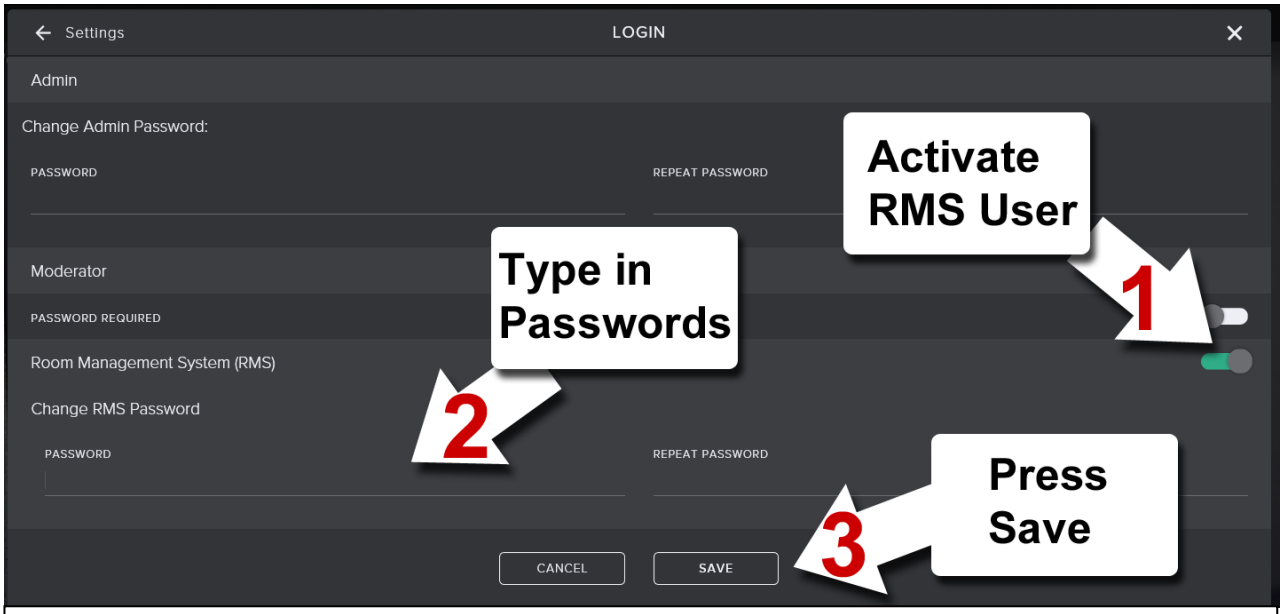
WolfVision GmbH
Oberes Ried 14
A-6833 Klaus / AUSTRIA
Tel. ++43-5523-52250
Fax ++43-5523-52249
E-Mail: wolfvision@wolfvision.com

Click on Toolbar	Click on Settings	Enter Password



Click on Login

Activate Room Management System User option and set your desired password



1. Enable Room Management System(RMS) – creates RMS_USER
2. Type in password. Once to create and a second time to verify it
3. Don´t forget to SAVE to commit the changes on Cynap

WolfVision GmbH
Oberes Ried 14
A-6833 Klaus / AUSTRIA

Tel. ++43-5523-52250
Fax ++43-5523-52249
E-Mail: wolfvision@wolfvision.com
www.wolfvision.com

4. WolfProt Command API´s

4.1 Command categories

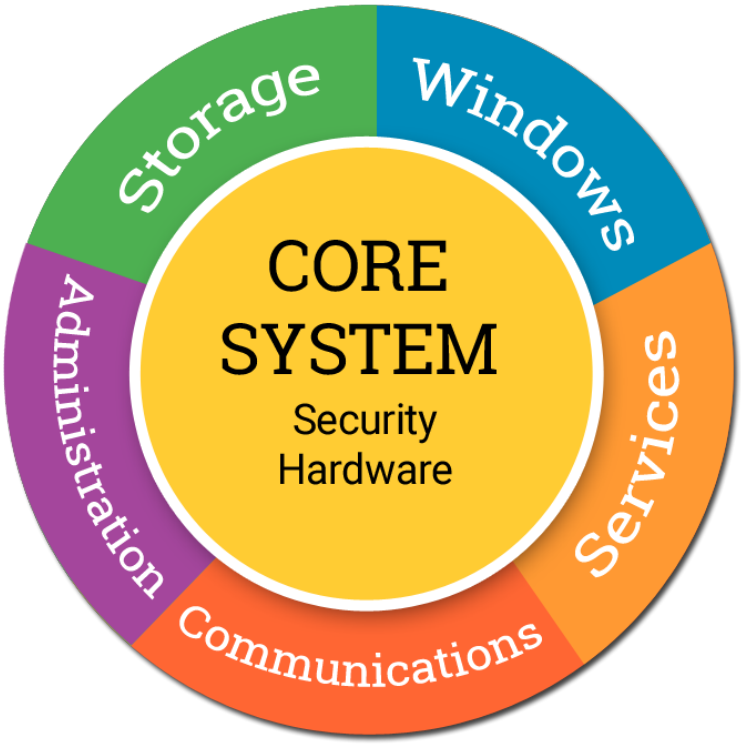


Figure 3: Cynap System Structure

WolfProt API´s are structured into 6 different categories:

Category	Description	Example
Core System	System commands	Send Standby command
Administration	Basically all the configurable settings (SET commands)	Enable and set GoogleDrive Session
Communications	LAN/WI-FI commands	Stop LAN connection
Services	Cynap services	Start/Stop Recording
Windows	Windows commands	Open red window
Storage	File commands	Write to USB

Table 6: Command structure

4.2 Command Structure

The commands are binary hex codes and have to be sent with an application. Each command has a defined reply. Commands should be send in SET and GET pairs to receive the status of an issued command.

Communication should use BSD sockets.
Connection: Ethernet or Wi-Fi, Protocol is TCP/IP.
Cynap: Port 50915 (unencrypted connection) or Port 50917 (encrypted connection)
Visualizer connected to Cynap: Port 50916 (unencrypted connection)

It is recommended to open a connection, issue various commands and then close the connection.
Response time: 15 to 100 ms on average.

Issuing a set command requires a follow up with its get command to be informed of the status or progress of the set command.

Some commands require an authorized user session before being issued. Please check **Fehler! Verweisquelle konnte nicht gefunden werden.** to see, if your command needs a login.

4.3 Returned Error Codes

There are 10 return codes in total. The return codes are placed in Header Bit 7 in the return message. To know, if a command succeeded you need to send a get command to receive the status of your request.

The reply packet to each command tells you:

- on set commands: the command got executed
- on get commands: the status or parameters that have been set on Cynap

Otherwise you will receive an error code if you did violate the protocol.
Be aware that if you don´t get a reply packet after sending a set command, you´re not violating the protocol but the socket is still expecting some additional values and waits forever for completion.

Example: when sending a larger number on “parameter length” but not providing the necessary parameter (e.g. size of *cynap.net* of 9 chars as being 20 chars).

Code	Explanation
1	Timeout
2	Invalid Command Unknown command
3	Invalid Parameter Unknown parameter
4	Invalid Length Check sizeof parameter
5	Queue Full Wait and repeat
6	Firmware Update Error Just firmware update command
7	Access Denied Wrong UserType or password
8	Authorization Required

	Command needs authorized session (issue login)
9	Busy

Table 7: Return Codes

A common mistake is using the wrong AccessLevel; not being logged in at all or logged in as User when an Administrator log in is required.

Example:
Request the streaming resolution (Administrator log in required) and being connected to Cynap as User and issuing a GET command.

Request streaming resolution >> 08 CB 23 00

When logged in as Administrator you will receive one of:
- 08 CB 23 01 00 (Full HD)
- 08 CB 23 01 01 (HD)
- 08 CB 23 01 00 (qHD)
- 08 CB 23 01 00 (nHD)

When not logged in or logged in as User you will receive
- 88 CB 23 07

Same happens with a SET command when you´re not logged in as Admistrator

Set streaming resolution >> 09 CB 23 01 00 (for Full HD)

When not properly logged in you will receive
- 89 CB 23 07

4.4 Authentication

Cynap supports authentication and most of the commands require an authorized session to be executed (please read chapter 7 to see a list of APIs and the required authorization).
The passwords can be set/changed in the GUI of Cynap's Settings screen.

Please read more about authentication on 5.1 authentication (UserLevels/UserTypes)

The default password for admin, moderator and Room Management System users is **Password** (case sensitive with a minimum length of 6 characters including blanks).

The admin password as well as the moderator and Room Management System passwords can be changed in the settings.

The passwords are hidden (***) and must be entered correctly to perform any changes.

The moderator login can also be set to PIN, then the four digits random PIN would be required for successful logins. The collaboration user requires a PIN and it is available in Annotation-Collaboration mode only.

If you forgot your admin password, please get in touch with support@wolfvision.com and have your Cynap´s Support PIN ready
>> visible on Front Panel -> DEVICE –> SUPPORT PIN

5. Special Types

5.1 Authentication (UserLevel/UserType/Access Levels)

There are 5 Users (or 4 User Types) to control Cynap via WolfProt Commands – if you want to just control Cynap (no changing of configuration settings) then we recommend using the Room Management System_User.

If you need to change settings via Terminal then the Admin user is the one you need to get access to Cynap´s settings.

Passwords have no minimum length; maximum length is 63 bytes (<=63).

	Role	Access Level	UserType	Description
	Admin	2	Admin	Allowed to use all get and set commands
	Room Management System_User	1	User	Allowed to set and get commands to operate Cynap (e.g. everything that´s being available via web interface minus the settings section)
	Moderator_User	1	User	Same as Room Management System_User with possible different login behaviour
	Annotation_User	3	Annotation	Allowed to use some get commands
	None		None	Allowed to use limited get commands

Table 8: UserType

The API´s which require User Level None will be processed without a required prior login.

The difference between Moderator_User and Room Management System_User is, that the Moderator uses a protective layer, forcing web browser visitors/users to log in before they can operate or see content on Cynap.

Additionally the password for the Moderator User can be set to a randomized PIN, making it more difficult to implement a dynamically changing PIN into your Room Management System solution.

Note: you always need to log in (at least on *Access Level 1* – User) to be able to operate with the same commands as a Cynap user.

The **Room Management System User**, having the same rights as the Moderator, on the other hand, is having its own password and offers no dynamically changed PINs.

The **Annotation** is the level used for Cynap's annotation functions.

The **Admin user** requires the Password set on Cynap.
The Room Management System User needs to be activated before it can be used and differentiates from the Admin on following authorizations:

The **User None** in general is useful for a quick testing, when you lack the password, and only need to see if you can get a response from Cynap.

Note: It is possible to develop a Cynap controller without a login function. But it requires switching off the login features (user/password combination) and enables everybody with a browser connection to Cynap to possibly interfering with a Cynap presentation.

Login Set Command
// open web socket
// 09 CB 42 {password_length + 2} {access_level} {password_length <=63} {password}
Send 09 CB 42 6 1 4 t e s t

5.2 Windows
There are four controllable windows on Cynap. They are differentiated on ID which represents always the same color and on the content they provide (set in parameters).

5.2.1 Windows Control
In general windows can be opened/closed, muted, put in full screen or set, if configured, on a second screen (HDMI2 out) called **Dualscreen** feature.

WindowControl gives you control to manipulate the 4 Cynap windows. In the API, the set command to open a window uses following structure:

Color and corresponding WindowID:

Column 3 (D0) represents the ID of the window (always first to fourth, 1 - 4) and it also represents the colors on your Cynap remote. Autoarrange (ff) creates the next available window.

Priority (same on remote)	Color	WindowID (d0)
1st	Red	00
2nd	Green	01
3rd	Yellow	02
4th	Blue	03
SourceButton	autoarrange	-1 (ff)

Table 9: Remote Window Color ID

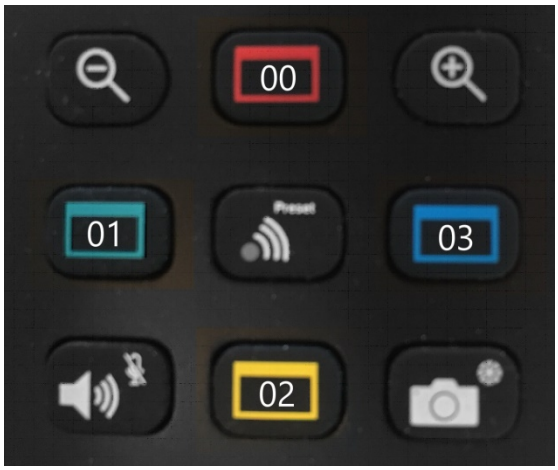


Figure 4: color coded windows on remote

Common Commands when selecting a specific window.

Set	Command	Window ID (d0)	Action	Function
09	CB 28 02	D0	00	Close
09	CB 28 02	D0	01	Size: Full screen
09	CB 28 02	D0	02	Size: Window
09	CB 28 02	D0	03 00	Audio: not muted
09	CB 28 02	D0	03 01	Audio: muted

09	CB 28 02	D0	04 nn	Audio: Volume level in % (0 to 99)
09	CB 28 02	D0	05	Toggle Full screen/Window
09	CB 28 02	D0	06	HDMI2 Copy: On
09	CB 28 02	D0	07	HDMI2 Copy: Off
09	CB 28 02	D0	08	HDMI2 Copy: Toggle

Table 10: Window Commands

Content-Aware commands such as to pause a video are being processed on WindowTypes (see below) and its specific command

5.2.2 WindowTypes

Based on the content the window changes its type. For example, a video content behaves differently than a PDF in that manner that the video player can be paused and a PDF file can be zoomed in. The parameter of a web browser consists of the URL and the parameter of an image of the file source location.

WindowTypes are:

Parameter-ID	Name	Functionality	Parameter
01	Visualizer	Wolfvision Visualizer support	none
02	HDMI Input	Select HDMI1 or HDMI2 input	HDMI1 or HDMI2
03	Web Browser	Start web browser	URL
04	Miracast	Microsoft´s Wi-Fi direct connection	none
05	AirPlay/GoogleCast	AirPlay and GoogleCast	none
06	Video	.mp4, .mkv, .avi	Location and name
07	vSolutionCast	BYOD for Windows 7	none
08	Image	Gif, jpg, png, bmp	Location and name
09	PDF	PDF files	Location and name
0a	Office presentation	Powerpoint files	Location and name
0b	Office text	Word files	Location and name
0c	Office calculation	Excel files	Location and name
0d	Whiteboard	Drawing	None
0e	Audio	.mp3, .oga	Location and name
0f	Webconference	WebRTC client	None
10	Webcam	USB Camera support	None

Table 11: Cynap Window Types

5.2.3 WindowType Example

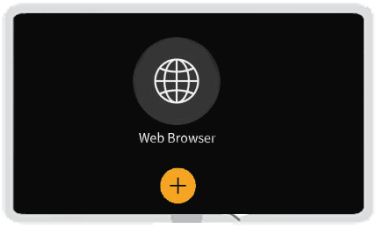
Scenario:
Provide a simple Web Browser function on your Room Management System where the user is able to use the keyboard to enter a URL and then see the URL on Cynap´s web browser.

On Cynap we expect a newly opened web browser window with a destination URL already loaded.

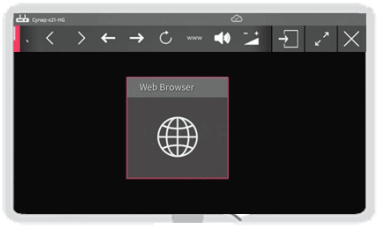
For simplicity we use the parameter autoarrange (-1 or FF) where Cynap open up the next possible window (you´re also allowed to use one of the four ID´s to open a specific window).

First design your layout and combine the layout with your implementation. In this example, most of the work needs to be done within your RMS IDE (creating icons and link them with internal functions).

Layout



1. Implement Menu



2. Create window context sensitive toolbar (e.g. button for web address)

3. Show keyboard and catch URL

4. After you store the URL you are able to issue the WolfProt Command : set the *WindowType* to *Browser* with a specific web address (cynap.net).

The URL will not be parsed by the WolfProt Command Agent – a malformed URL will be displayed as not found – the developer of the controller implementation needs to take care of validating the user entered URL.

WolfProt Command:

```
// Your RMS layout/implementation done
// Loop: Open socket and poll public GET command
//                              (e.g. polling a get command to receive a PIN,
//                              see PIN Status Room, 08 CB 54 00)
// Loop: Check if AuthorizationLevel is set to RMS_User
//                              (Set Login, 09 CB 42 01, see Login)
// All OK? Then send “Window Start, WindowID=auto arranged, WindowType=browser”

While (socketOpen)
  While (RMS_User_logged_in)
    Send 0D CB 2C 00 0D FF 03 00 09 c y n a p . n e t
```

0D	This command doesn´t start with the usual 09 visible in the set command, as the length of the parameter could exceed 256 (e.g. URL > 256 chars).
CB 2C	the call for Window Start
00 0D	the size of the parameter is 13 or in hex 0D. Since we have to follow protocol,

	the first 00 must be set to 0
FF	-1 for auto arranging the window
03	type of window equals browser
00 09	size of parameter, in this case, size of the following web address
c y n a p . n e t	doesn´t need to be converted into hex (c equals to 0x63)

Window Start command based on API reference (see chapter 8 APIs by functionality)

Window Start	Command	Parameters		Reply Package
	0D CB 2C ***ab cd d0 d1 d2 d3 d4 dn***	**Param** ***ab, cd*** ***d0*** ***d1*** ***d2, d3*** ***d4 to dn***	**Description** Length of parameter + 4 Window number (-1/0xff for auto arrange) **Window types:** **Value Type** 0x01 WolfVision Visualizer 0x02 HDMI Input 0x03 Browser 0x04 Miracast 0x05 AirPlay 0x06 Video 0x07 vSolution Cast 0x08 Image 0x09 PDF 0x0a Office presentation 0x0b Office text 0x0c Office spreadsheet 0x0d Whiteboard 0x0e Audio 0x0f WebRTC 0x10 Web Cam Length of parameter (2 bytes) Name in characters	09 CB 2C 00

6 File Operations

API	Command	Description
Get Mounts List	08 CB 3D 00	Information on available mounted file system and their status (JSON array)
Get File List	0C CB 3E	Provide the root path and you will get a list of files (JSON array)
Get File Download List	08 CB 7B 01	If you send a length of 0 you will get a merged list of all downloads otherwise (length == 1) you´re able to filter 0: DropBox 1: Google Drive 2: Box
Get File Upload List	08 CB C1 01	JSON array of all file uploads in progress
Get Cloud Mode	08 CB 8F 00	Disabled/enabled returns per cloud service
Get Cloud Status	08 CB 4C 00	Status on all cloud services returned
Get FTP Mode	08 CB 62 01	FTP server configured – no status on availability

Table 12: File commands

The array of the mounted drive not only tells you the name but also if the drive is writeable or available in the cloud.

name | name of mount	Id | unique ID	type | local, net	status | mounted or not	perms | read/write
Internal	Internal	Local	mounted/notMounted	ro: read only
System	System	system	notMounted	
USB	USB	usb	mounted/notMounted	rw: read, write
Dropbox	dropbox	cloud	mounted/notMounted	rw: read, write
Google Drive	gdrive	cloud	mounted/notMounted	rw: read, write
Box	Box	cloud	mounted/notMounted	rw: read, write
Network Drive 0	netdrive0	netdrive	mounted/disabled	ro/rw
Network Drive 1	Netdrive1	netdrive	mounted/disabled	ro/rw
Network Drive 2	Netdrive2	netdrive	mounted/disabled	ro/rw
Network Drive 3	Netdrive3	netdrive	mounted/disabled	ro/rw
Network Drive 4	Netdrive4	netdrive	mounted/disabled	ro/rw
Network Drive 5	Netdrive5	netdrive	mounted/disabled	ro/rw
Network Drive 6	Netdrive6	netdrive	mounted/disabled	ro/rw
Network Drive 7	Netdrive7	netdrive	mounted/disabled	ro/rw
Network Drive 8	Netdrive8	netdrive	mounted/disabled	ro/rw
Network Drive 9	Netdrive9	netdrive	mounted/disabled	ro/rw
FTP	FTP	ftp	mounted/disabled	wo

The ftp session won´t be checked every 10 secs unlike the network drives – the status mounted therefore doesn´t tell you if the ftp-connection is working; the status mounted informs you that a ftp connection has been configured.

Cynap´s operating system does not allow file manipulations. Temporarily sharing media files content from local and remote resources is allowed.

Downloading video images office documents from the cloud services
Displaying video images office documents from internal/external/remote locations
Uploading recordings and snapshots only from internal location

The file list array consists of two fields. Field one is filled with the file name and field two contains the file type.

The file types supported by Cynap have a specific identifier and file types not supported by Cynap are specified as unknown.

To open a file you have to prepare its fully qualified name. The fully qualified name is generated from the mounted storage device, 3 leading slashes, the path and the filename itself.

e.g. USB:///Folder1/video1.mp4

File Type	Description
Audio	Audiofile (e.g. .mp3, oga, …)
calc	Spreadsheet files (.xls, xlsx)
dir	Directory
html	Locally saved webpage
image	Pictures in gif, png or other supported formats
pdf	PDF file
presentation	Powerpoint
text	Word or text files
unknown	Unknown file format - please hide or mark as unknown
video	Supported video format

6.1 File Transfer (FTP)

When uploading a file from Cynap to an FTP server please make sure, that your FTP server is already set up with a user and password combination and the proper rights to create/replace files – FTP file transfer offers a user/password combination but no further settings such as ACTIVE/PASSIVE or specific parameters such as Kerberos login or other sFTP options.
Cynap´s FTP client connection is not polled and uploading a file to the FTP server requires your implementation to handle connection continuity.

6.2 Cloud (box, Dropbox and Google Drive)

The cloud as read-write device, once logged in by the user, behaves like a common network file share (CIFS).

Mount names of type Cloud

Mount	Id	URL
Box:///	Box	https://www.box.com
Dropbox:///	Dropbox	https://www.dropbox.com/
Google Drive:///	Gdrive	https://www.google.com/drive/

Cloud status: WPC_Cloud_Status (0xCB4C)

0x00	Disconnected	Connect if required
0x01	Oauth (Open Authorization 2.0)	wait
0x02	Connected	file list access pending: wait before fetching the cloud file list
0x03	Connection failed	Network/authorization problem
0x04	Synced	Cloud directory read:

		Ready to
0x05	Disabled (not configured)	

Cloud API´s

	WPC_CLOUD_CONNECT	0xCB45
	WPC_CLOUD_PRELOAD	0xCB46
	WPC_CLOUD_STATUS	0xCB4C
Enable/Disable cloud services	WPC_CLOUD_MODE	0xCB8F
	WPC_CLOUD_UPLOAD	0xCBA1
	WPC_BOX_CLOUD_DATA	0xCBCC

6.3 Network file share (CIFS)

Cynap´s network file share is based on CIFS (Common Internet File Share). Once configured (up to 10 drives) it allows you to up/download files from Microsoft servers.

6.4 USB external HDD or memory stick

A single FAT32 formatted drive can be attached and accessed via Cynap.

6.5 Internal storage

There are two types of files:
1. Cynap created content such as *snapshots* and *recordings* and
2. Temporary files such as downloaded content from attached remote locations

Only Cynap created files can be deleted, uploaded or copied to a USB storage device.

Example: JSON Array of list of mounted storage


```
[
{"name":"Internal","id":"Internal","type":"local","status":"mounted","perms":"ro"},
{"name":"System","id":"System","type":"system","status":"notMounted"},
{"name":"USB","id":"USB","type":"usb","status":"mounted","perms":"rw"},
{"name":"Dropbox","id":"dropbox","type":"cloud","status":"notMounted"},
{"name":"Google Drive","id":"gdrive","type":"cloud","status":"mounted","perms":"rw"},
{"name":"Box","id":"box","type":"cloud","status":"notMounted"},
{"name":"Network Drive 0","id":"netdrive0","type":"netdrive","status":"mounted","perms":"rw"},
{"name":"Network Drive 1","id":"netdrive1","type":"netdrive","status":"disabled"},
{"name":"Network Drive 2","id":"netdrive2","type":"netdrive","status":"disabled"},
{"name":"Network Drive 3","id":"netdrive3","type":"netdrive","status":"disabled"},
{"name":"Network Drive 4","id":"netdrive4","type":"netdrive","status":"disabled"},
{"name":"Network Drive 5","id":"netdrive5","type":"netdrive","status":"disabled"},
{"name":"Network Drive 6","id":"netdrive6","type":"netdrive","status":"disabled"},
{"name":"Network Drive 7","id":"netdrive7","type":"netdrive","status":"disabled"},
{"name":"Network Drive 8","id":"netdrive8","type":"netdrive","status":"disabled"},
{"name":"Network Drive 9","id":"netdrive9","type":"netdrive","status":"disabled"},
{"name":"FTP","id":"FTP","type":"ftp","status":"disabled","perms":"wo"}
]
```

WolfVision GmbH
Oberes Ried 14
A-6833 Klaus / AUSTRIA
Tel. ++43-5523-52250
Fax ++43-5523-52249
E-Mail: wolfvision@wolfvision.com
www.wolfvision.com

Example: JSON Array of file listing (root USB:///)

```
[
{ "name": "wolfvision", "type": "dir" },
{ "name": "bigbunny.avi", "type": "video" },
{ "name": "compressed.rar", "type": "unknown" },
{ "name": "cynap.log", "type": "unknown" },
{ "name": "music.mp3", "type": "audio" },
{ "name": "test.txt", "type": "text" },
{ "name": "user.data, "type": "unknown" },
{ "name": "video1.mp4", "type": "video" }
{ "name": "video2-h265.mkv", "type": "video" },
]
```

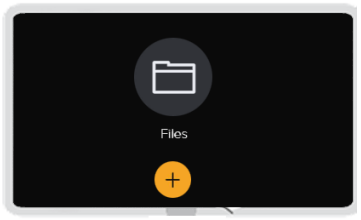
6.6 Tutorial: File Operations

Scenario:
To fetch the file list of an inserted USB Memory stick and open/use a specific file on it. Before a file list can be requested, it is necessary to get information on the mounted drives.

Some files aren´t supported by Cynap and we recommend hiding them or at least mark them as unsupported.

User behavior:
The Cynap user presses the orange Plus-Icon (Source-Button) and selects the files icon. All available sources, including the file source (indicated by “Files” and an icon of a folder), are displayed.

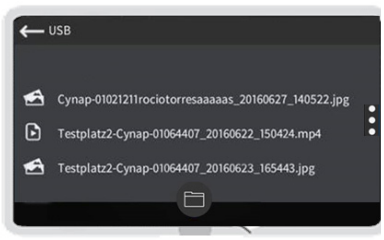
Layout
Create File list button
When pressed, show the mounted drives
Open drive and request the list of files
Open selected file with the appropriate window type



1. Implement File dialogue

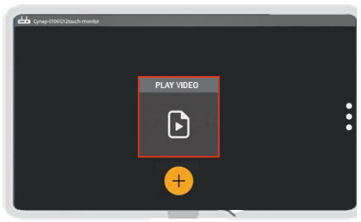
2. Get the list of mounted drives

Execute command
GET Mounts List, 08 CB 3D 00



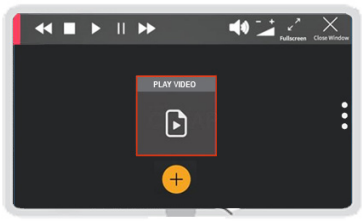
3. Show file list – filter out files of type unknown which can not be displayed on Cynap

Execute command
GET File List, 0C CB 3E xx



4. Play the selected file

Execute command
SET Open File, 0D CB 3C xx
Use fully qualified name:
e.g. USB:///video.mp4



5. Set controls based on context of content

Execute command
GET Windows , 08 CB BA 00

WolfProt Commands:

```
// Your layout/implementation done
// Loop: Open socket and poll public GET command
//                              (e.g. polling a get command to receive a PIN,
//                              see PIN Status Room, 08 CB 54 00)
// Loop: Check if AuthorizationLevel is set to RMS_User
//                              (Set Login, 09 CB 42 01)
// All OK? Then send "Window Start, WindowID=auto arranged, WindowType=browser"

While (socketOpen)
  While (RMS_User_logged_in)

    //get and use list of mounted drives
    Send 08 CB 3D 00   //get mounted storage devices
    // process returned JSON array   (e.g. USB:///)

    //fetch event and get list of selected storage device
    // user clicks on USB storage device

    //send mounted device and parent path to show files and directories
    Send 08 CB 3E 00 07 u s b : / / /
    // process returned JSON array  (e.g. video1.mp4)
    // hide files of type unknown or clearly mark them as inaccessible

    //fetch event and open requested file
    // next available window on Cynap will open itself
    Send 0D CB 3C 00 10 u s b : / / / v i d e o . m p 4
```

6.7 Tutorial: Visualizer through Cynap

Scenario:
Changing the Autofocus from off to on

The existence of a Visualizer must be checked and also if the Autofocus setting isn´t activated yet.

User behavior:
The user just pressed the autofocus on button on the visualizer.

WolfProt Commands:

```
//
// VISUALIZER ZOOM VIA PORT 50916 (INSTEAD OF CB_EVENT WHERE THE REMOTE BUTTONS ARE USED
//
//
// the Visualizer can be controlled without an open Visualizer window on cynap. However it
// is useful if the user needs visual confirmation on Cynap (e.g. zooming in and see reaction on
// Cynap window.

while (connected and logged in) {
    // Visualizer connectd/disconnected?
    send 08 CB 68 00
    if return == 08 CB 68 01 00 // 00 disconnected 01 connected
    {
    feedback: no visualizer connected
    }
    else {
       open socket to Visualizer (Port 50916) // no login necessary, the Visualizer commands are directly routed over
        while (connected to Visualizer) {
        // First open a Visualizer window on Cynap
        open new window
        // then check if Autofocus is on or off
        send
        if autofocus is off
          (
          send 09 CB 29 02 d0 0
          )
        else
          (
          feedback: autofocus already on
          )
        }
     }
 }
```

7 Cynap in Standby

Figure 5: Standby process

Once the user presses the Standby button, Cynap will enter the Standby mode by checking if vital background processes need to be completed (such as an initiated upload on a recording) and if or if not waits 30 seconds in case a restart got received.

The networking stack and all other monitoring activities are shut down. The only way to start a Cynap from standby is to use the remote, the power button on the front panel or cable based network* to send a Wake-On-LAN command.

Since your Room Management System is connected to Cynap via Ethernet, you will need to issue a Wake-On-LAN command.

Polling to keep a connection on Cynap via ping or echo command does not work- you will have to issue a GET command to get the proper status (esp. if Cynap is entering standby)

To wake Cynap from standby a user needs to press the power button on the remote or on Cynap.

Your Room Management System implementation needs to send a broadcast to the MAC address of Cynap issued in a magic packet to initiate the start-up process.

Figure 6: Wake-On-LAN process

*Wake on WI-FI is not supported as antennas are not powered on standby to receive signals

```
// Prepare Broadcasting
// IP = 255.255.255.255  Broadcast IP
// Port = 0xC6E1 Port 50913 decimal to hex
Cynap.connect (
       (0xffffffff), // 255.255.255.255 in hex
       (0xc6e1)      // Port 50913 in hex

// prepare SendBuffer
Init SendBuffer[1024] //address buffer of size of 1024 bytes
int counter=0  // pointer to be used to add required information at correct buffer pointer
location

// first 6 bytes should be 0xff
for (int c=0;c<6;c++)
       SendBuffer[counter++] = "0xff"

// next add 16 times the CYNAP_MAC_ADRESS
CYNAP_MAC_ADDRESS ="086066ff66ff" // (should look like 08:60:66:FF:66:FF or 086066FF66FF)
for  (int c=0; c<16;c++){
       int d=0
       // convert CYNAP_MAC_ADDRESS into hex-format
       for (int e=0;e<6;e++) //6 bytes of CYNAP_MAC_ADDRESS{
              SendBuffer[counter++] = ConvertToHex (CYNAP_MAC_ADDRESS.substring(d,2),HexValue)
              d+=2}}

// Send Broadcast to activate MAC addressed device
send status=Cynap.send (SendBuffer,1024)
```

Figure 7: Example Wake-On-LAN

8 APIs by functionality

Cynap System Commands		UserLevel
08 CB 00 00	Get Cynap Model Type Syntax: 08 CB 00 00 Variables: None Returns: 08 CB 00 00 Examples 08 CB 00 00 -	
	Set Cynap Model Type n/a	
08 CB 60 00	Get Cynap serial number Syntax: 08 CB 60 00 Variables: None Returns: 08 CB 60 04 D0 – D3 Variables: Serial number D0 – D3 Examples: >> 08 CB 60 00 << 08 CB 60 04 AA 11 BB 22	
08 CB 03 00	Get Name of Cynap Syntax: 08 CB 03 00 Variables: None Returns: 08 CB 03 LEN D0 … Dn Variables: LEN= size of returned name, D0 to Dn = name (array of characters) Examples: >> 08 CB 03 00 << 08 CB 03 04 N A M E	

Cynap System Commands		UserLevel
09 CB 03	Set the name of Cynap Synax: 09 CB 03 LEN, D0 …Dn Variables: LEN : [size of name to be inserted], [D0 to Dn : name, array of characters] Returns: 09 CB 03 00 Examples: 09 CB 03 4 N A M E	
08 CB 77 01 00	Verify support Syntax: 08 CB 77 01 00 Variables: None Returns: 08 CB 77 02 00 D0 [D0 Status: 00: failed, 01: OK] Examples: >> 08 CB 77 01 00 << 08 CB 77 02 00 00 << 08 CB 77 02 00 01	
08 CB 77 01 01	Fetch support key Syntax: 08 CB 77 01 01 Variables: None Returns 08 CB 77 LEN 01 D0 … Dn [Length of key +1] 01 [D0 … Dn: Key] Examples: >> 08 CB 77 01 01 << 08 CB 77 0B 01 1 2 3 4 5 6 7 8 9 0	
08 CB 77 01 02	End of support period (last calendar year) Syntax 08 CB 77 01 02 Variables: None Returns: 08 CB 77 03 02	

Cynap System Commands		UserLevel
08 CB 7F 01 00	Verify License Syntax: 08 CB 7F 01 00 Variables: None Returns: 08 CB 7F 02 00 D0 [D0 Status: 00: failed, 01: OK] Examples: >> 08 CB 7F 01 00 << 08 CB 7F 02 00 00 << 08 CB 7F 02 00 01	
08 CB 7F 01 01	Fetch license key and license features Syntax: 08 CB 7F 01 01 Variables: none Returns 08 CB 7F LEN 01 D0 … Dn [Length of key +1] 01 [D0 … Dn: Key] Examples: >> 08 CB 7F 01 01 << 08 CB 7F 0B 01 1 2 3 4 5 6 7 8 9 0	
08 CB 7F 01 01	Fetch license features Syntax: 08 CB 7F 01 02 Variables: none Returns 08 CB 7F LEN 01 D0 … Dn [Length of key +1] 01 [D0 … Dn: Key] Examples: >> 08 CB 7F 01 02 << 08 CB 7F 0B 01 1 2 3 4 5 6 7 8 9 0	
08 CB 1D 00	Is Time Server switched On or Off Syntax: 08 CB 1D 00 Variables: None Returns: 08 CB 1D 01 [Length, always 01] 00 [D0: Time Server Off] 08 CB 1D 01 [Length, always 01] 01 [D0: Time Server On] Examples: >> 08 CB 1D 00 << 08 CB 1D 01 01	

Cynap System Commands		UserLevel
09 CB 1D 01	Activate or deactivate Time Server Syntax: 09 CB 1D 01 Variables 00 [Set Time Server Off] 01 [Set Tie Server On] Returns: 09 CB 1D 00 Examples: 09 CB 1D 01 00	ADM
08 CB 1F 00	Get IP Address of Time Server Syntax: 08 CB 1F 00 Variables: None Returns: 08 CB 1F 04 [D0 – D3] each octet of IP, 0xC0 0xA8 0x00 0x0A : 192.168.0.10 Examples >> 08 CB 1F 00 << 08 CB 1F 04 C0 A8 00 0A	MOD
09 CB 1F	Set IP Address of Time Server Syntax: 09 CB 1F Variables: [Length: always 04] [Octet 1] [Octet 2] [Octet 3] [Octet 4] Returns: 09 CB 1F 00 Examples 09 CB 1F 04 0C A8 00 0A	ADM
08 CB 1E	Get Date Format: DDMMYYYY/24H or MMDDYYYY/AM-PM Syntax: 08 CB 1E 00 Variables: None Returns: 08 CB 1E 01 00 DDMMYYYY/24 08 CB 1E 01 01 MMDDYYYY/AM-PM Examples: >> 08 CB 1E 00 << 08 CB 1E 01 00	MOD

Cynap System Commands		UserLevel
09 CB 1E	Set Date Format: DDMMYYYY/24H or MMDDYYYY/AM-PM Syntax: 09 CB 1E 01 Variables: 00 DDMMYYYY/24 01 MMDDYYYY/AM-PM Returns: 09 CB 1E 00 Examples: >> 08 CB 1E 01 01	
08 CB 1C	Get Cynap´s date and time Syntax: 08 CB 1C Variables: None Returns: 08 CB 1C [Length always 0x05] [Year: 0x00 to 0x63][Month: 0x01 to 0x0C] [Day: 0x01 to 1F] [Hours:0x00 to 0x17] [Minutes: 0x00 to 0x3B] >> 08 CB 1C << 08 CB 1C 05 11 0B 07 0B 0F (171107-11.15)	
09 CB 1C	Set Cynap´s date and time Syntax: 09 CB 1C Variables: [Length always 0x05] [Year: 0x00 to 0x63][Month: 0x01 to 0x0C] [Day: 0x01 to 1F] [Hours:0x00 to 0x17] [Minutes: 0x00 to 0x3B] Returns: 08 CB 1C 00 Send << 08 CB 1C 05 11 0B 07 0B 0F (171107-11.15) Receive >> 09 CB 1C 00	

Cynap System Commands		UserLevel
09 CB 1C	Set the Time Zone of Cynap Syntax: 09 CB 82 01 00 Pacific/Apia 09 CB 82 01 01 Pacific/Honolulu 09 CB 82 01 02 America/Anchorage 09 CB 82 01 03 America/Los_Angeles 09 CB 82 01 04 America/Tijuana 09 CB 82 01 05 America/Phoenix 09 CB 82 01 06 America/Denver 09 CB 82 01 07 America/Mazatlan 09 CB 82 01 08 America/Chicago 09 CB 82 01 09 America/Regina 09 CB 82 01 0A America/Mexico_City 09 CB 82 01 0B America/Bogota 09 CB 82 01 0C America/New_York 09 CB 82 01 0D America/Caracas 09 CB 82 01 0E America/Halifax 09 CB 82 01 0F America/Santiago 09 CB 82 01 10 America/La_Paz 09 CB 82 01 11 America/St_Johns 09 CB 82 01 12 America/Sao_Paulo 09 CB 82 01 13 America/Buenos_Aires 09 CB 82 01 14 America/Montevideo 09 CB 82 01 15 Atlantic/South_Georgia 09 CB 82 01 16 Atlantic/Azores 09 CB 82 01 17 Atlantic/Cape_Verde 09 CB 82 01 18 Africa/Casablanca 09 CB 82 01 19 UTC 09 CB 82 01 1A Europe/London 09 CB 82 01 1B Europe/Berlin 09 CB 82 01 1C Africa/Kinshasa 09 CB 82 01 1D Europe/Istanbul 09 CB 82 01 1E Africa/Cairo 09 CB 82 01 1F Asia/Jerusalem 09 CB 82 01 20 Africa/Johannesburg 09 CB 82 01 21 Europe/Moscow 09 CB 82 01 22 Africa/Nairobi 09 CB 82 01 23 Asia/Baghdad 09 CB 82 01 24 Asia/Tehran 09 CB 82 01 25 Asia/Muscat 09 CB 82 01 26 Asia/Baku 09 CB 82 01 27 Asia/Kabul 09 CB 82 01 28 Asia/Karachi 09 CB 82 01 29 Asia/Kolkata 09 CB 82 01 2A Asia/Katmandu 09 CB 82 01 2B Asia/Almaty 09 CB 82 01 2C Asia/Dhaka 09 CB 82 01 2D Asia/Rangoon 09 CB 82 01 2E Asia/Bangkok 09 CB 82 01 2F Asia/Hong_Kong 09 CB 82 01 30 Asia/Singapore 09 CB 82 01 31 Australia/Perth 09 CB 82 01 32 Asia/Taipei 09 CB 82 01 33 Asia/Tokyo 09 CB 82 01 34 Asia/Seoul 09 CB 82 01 35 Australia/Adelaide 09 CB 82 01 36 Australia/Darwin 09 CB 82 01 37 Australia/Sydney 09 CB 82 01 38 Australia/Brisbane 09 CB 82 01 39 Pacific/Noumea 09 CB 82 01 3A Pacific/Auckland 09 CB 82 01 3B Pacific/Kwajalein Variables None Returns: 09 CB 82 00 Example 09 CB 82 01 30 Asia/Singapore	ADM

Cynap System Commands		UserLevel
08 CB 82	Get the Time Zone of Cynap Syntax: 08 CB 82 00 Returns: 09 CB 82 01 00 Pacific/Apia 09 CB 82 01 01 Pacific/Honolulu 09 CB 82 01 02 America/Anchorage 09 CB 82 01 03 America/Los_Angeles 09 CB 82 01 04 America/Tijuana 09 CB 82 01 05 America/Phoenix 09 CB 82 01 06 America/Denver 09 CB 82 01 07 America/Mazatlan 09 CB 82 01 08 America/Chicago 09 CB 82 01 09 America/Regina 09 CB 82 01 0A America/Mexico_City 09 CB 82 01 0B America/Bogota 09 CB 82 01 0C America/New_York 09 CB 82 01 0D America/Caracas 09 CB 82 01 0E America/Halifax 09 CB 82 01 0F America/Santiago 09 CB 82 01 10 America/La_Paz 09 CB 82 01 11 America/St_Johns 09 CB 82 01 12 America/Sao_Paulo 09 CB 82 01 13 America/Buenos_Aires 09 CB 82 01 14 America/Montevideo 09 CB 82 01 15 Atlantic/South_Georgia 09 CB 82 01 16 Atlantic/Azores 09 CB 82 01 17 Atlantic/Cape_Verde 09 CB 82 01 18 Africa/Casablanca 09 CB 82 01 19 UTC 09 CB 82 01 1A Europe/London 09 CB 82 01 1B Europe/Berlin 09 CB 82 01 1C Africa/Kinshasa 09 CB 82 01 1D Europe/Istanbul 09 CB 82 01 1E Africa/Cairo 09 CB 82 01 1F Asia/Jerusalem 09 CB 82 01 20 Africa/Johannesburg 09 CB 82 01 21 Europe/Moscow 09 CB 82 01 22 Africa/Nairobi 09 CB 82 01 23 Asia/Baghdad 09 CB 82 01 24 Asia/Tehran 09 CB 82 01 25 Asia/Muscat 09 CB 82 01 26 Asia/Baku 09 CB 82 01 27 Asia/Kabul 09 CB 82 01 28 Asia/Karachi 09 CB 82 01 29 Asia/Kolkata 09 CB 82 01 2A Asia/Katmandu 09 CB 82 01 2B Asia/Almaty 09 CB 82 01 2C Asia/Dhaka 09 CB 82 01 2D Asia/Rangoon 09 CB 82 01 2E Asia/Bangkok 09 CB 82 01 2F Asia/Hong_Kong 09 CB 82 01 30 Asia/Singapore 09 CB 82 01 31 Australia/Perth 09 CB 82 01 32 Asia/Taipei 09 CB 82 01 33 Asia/Tokyo 09 CB 82 01 34 Asia/Seoul 09 CB 82 01 35 Australia/Adelaide 09 CB 82 01 36 Australia/Darwin 09 CB 82 01 37 Australia/Sydney 09 CB 82 01 38 Australia/Brisbane 09 CB 82 01 39 Pacific/Noumea 09 CB 82 01 3A Pacific/Auckland 09 CB 82 01 3B Pacific/Kwajalein Example 08 CB 82 00	 MOD

Cynap System Commands		UserLevel
08 CB 02	Get Cynap Picture, a screenshot of the screen output in JPEG format. There are 6 types of preview pictures available: **Source 0**: Cynap main screen **Source 1**: HDMI 1 input (standard Visalizer HDMI port) **Source 2**: HDMI 2 input **Source 3**: Cynap main screen with overlays (such as toolbar, source button) **Source 4**: Visualizer defined HDMI input **Source 5**: Recording (depending on Output HDMI2 Mode settings either HDMI1 or HDMI 2 output) Syntax: 08 CB 02 06 [D0 D1] [D2 D3] [D4] [D5] Variables: Length, always 06, D0 D1 picture width, 1920 = hex 780 = [D0] 07 [D1] 80 [D2 D3] picture height,[D4] picture type JPG 0x00 only, [D5] Picture Source, see above Returns: 08 CB 02 [LEN + 12] [I0 I1 I2 I4]: Packet Length [D0 D1] picture width [D2 D3] picture height [D4 D5 D6 D7] picture length [D8 - Dn] picture data Example: >> 08 CB 02 06 07 80 04 38 00 03 << 08 CB 02 Note: if a source is not defined, you will receive a black picture.	
09 CB 01 08 CB 34 00	Remote Control Commands Get Block **TO BE DISCUSSED – might become obsolete or expanded**	
0B 01 CB 80	Changing the background image of Cynap Mandatory: JPEG format, 1920x1080 pixels, 8-bit color Syntax: 0B 01 CB 80 [D0 D1 D2 D3] [D4] [D5 –Dn] picture data Variables: Length, always 80, [D0 D1 D2 D3] Length [D4] 0:delete, >0:save [D5 -Dn] picture data Returns: 09 CB 80 00 Example: >> 0B 01 CB 80 [size] 01 [picture data] << 09 CB 80 00	
08 CB 85	Status if Cap-Lock/Shift key is pressed or not. Syntax: 08 CB 85 00 Variables: None Returns: 08 CB 85 01 [D0] Parameters: [D0] 0: OFF, 1: ON Example: >> 08 CB 85 00 << 08 CB 85 01 00 (Caps lock Off)	

Cynap System Commands		UserLevel
08 CB 2E	Firmware-Update USB: Get Progress and Status Syntax: 08 CB 2E 00 Variables: None Returns: 08 CB 2E 02 NN D0 Parameters: [NN] 00 Idle, 01 Upload, 02 Programming, 03 Finished, 04 Failed [D0] Progress 0 – 100 (%) Example: Status Programming 99% >> 08 CB 2E 00 << 08 CB 2E 02 02 63	MOD
08 CB 7A	Firmware-Update Online: Get Progress and Status Syntax: 08 CB 7A 00 Variables: None Returns: 08 CB 7A LEN D0 D1 D2 … Dn Parameters: [LEN] Length of online firmware + 2 [D0] Status: 0: idle, 1: failed, 2: no connection, 3: checking, 4: up to date, 5: out of date, 6: downloading, 7: updating [D1] Progress 0 – 100 (%) [D2] Firmware version Example: >> 08 CB 7A 00 << 08 CB 7A 02 00 00	MOD
08 CB 38	Get Mirroring Presentation Mode. There are 2 modes, Meeting and Lecture. In Lecture Presentation Mode, the moderator has to press the Mirroring Icon on the bottom toolbar in Meeting Presentation Mode. Syntax: 08 CB 38 00 Variables: None Returns: 08 CB 38 01 D1 Parameters: [D1] 0, Meeting Presentation Mode, 1 Lecture Presentation Mode Example: >> 08 CB 38 00 << 08 CB 38 01 01	MOD

Cynap System Commands		UserLevel
09 CB 38	Set the Mirroring Presentation Mode. There are 2 modes, Meeting and Lecture. In Lecture Presentation Mode, the presente has to press the Mirror Icon in the bottom toolbar in Meeting Presentation Mode not. Syntax: 09 CB 38 01 D1 Variables: [D1] 00 Meeting Presentation Mode, 01 Lecture Presentation Mode Returns: 09 CB 38 00 Parameters: None Example: 09 CB 38 01 00	
09 CB 49	Ends the Presentation – ends the current presentation and gives the options to keep or delete recording/snapshots and if Cynap enters Standby mode. Syntax: 09 CB 49 LEN D0 D1 D2 Variables: [LEN] 03, fixed length [D0] Delete recordings, 0 keep, 1 delete [D1] Delete recordings, 0 keep, 1 delete [D2] Enter Standby mode, 0 no, 1 yes Returns: 09 CB 49 00 Parameters: None Example: 09 CB 49 03 00 00 00	
08 CB 5E	Get status bar network indicator – tells which network indicators are on or off with a single return status. Syntax: 08 CB 5E 00 Variables: None Returns: 08 CB 5E 01 D0 Parameters: [D0] 00, No IP shown, 01 LAN only, 02 WI-FI only, 03 LAN and WI-FI Example: >> 08 CB 5E 00 Receive<< 08 CB 5E 01 03	

Cynap System Commands		UserLevel
09 CB 5E	Set status bar network indicator – set the network indicators. Syntax: 09 CB 5E 01 D0 Variables: [D0] 00, No IP shown, 01 LAN only, 02 WI-FI only, 03 LAN and WI-FI Returns: 09 CB 5E 00 Parameters: None Example: 08 CB 5E 01 03	ADM
0D CB 6B	Create/replace the settings file – for seeding multiple Cynaps or an online restore of a previously saved configuration. The data will be decrypted and replaces an existing settings file. Syntax 0D CB 6B AB CD D0 .. Dn Variables: [AB][CD] sizeof Binary Settings File [D0] [Dn] multiple bytes containing settings Returns: 09 CB 6B 00 Parameters: None Example: 0D CB 6B 09 FF {array of data} // filesize = 2559 bytes	ADM
08 CB 6B	Retrieve the settings file – for automated periodical backups. This file is encrypted and cannot be read without being decrypted first. Syntax: 08 CB 6B 00 Variables: None Returns: 0C CB 6B AB CD D0 … Dn Parameters: [AB] [CD] sizeof settings file [D0]…[Dn] encrypted settings file Example: >> 08 CB 6B 00 << 0C CB 6B 09 FF {array of data}	ADM
09 CB 84	Activates a keyboard layer on Cynap´s web interface	ADM

Cynap System Commands		UserLevel
08 CB 84	Check if keyboard has been activated or not	 MOD
08 CB 68	Check if Visualizer is connected or not	 MOD

Cynap Security Commands		UserLevel
09 CB 41	Activate or Deactivate Moderator User If you activate this user, then please make sure that you also enter a password immediately (using command 09 CB 40). Syntax: 09 CB 41 01 D0 Variables: [D0] 00 Deactivate, 01 Activate PIN Returns: 09 CB 41 00 Parameters: None Example: 09 CB 41 01 01	 ADM
08 CB 41	Verify if Moderator user has been activated or not. Syntax: 08 CB 41 00 Variables: None Returns: 08 CB 41 01 D0 Parameters: [D0] 00 Deactivated, 01 Activated Example: >> 08 CB 41 00 << 08 CB 41 01 01	 ADM

09 CB 51	Set type of password – randomized and changing PIN or plain text Syntax: 09 CB 51 01 D0 Variables: [D0] Password type 00, text or 01 PIN Returns: 09 CB 51 00 Parameters: None Example 09 CB 51 01 01	 ADM
08 CB 51	Get type of password – randomized and changing PIN or plain text Syntax: 09 CB 51 00 Variables: None Returns: 08 CB 51 01 D0 Parameters: [D0] Password type 00, text or 01 PIN Example 09 CB 51 01 01	 ADM
09 CB 52	Set the PIN display destinations – allows to include wanted or exclude unwanted destinations where the login PIN is shown Syntax: 09 CB 52 04 D0 D1 D2 D3 Variables: [D0] On Visualizer display 00 excluded 1 included [D1] On Cynap display 00 excluded 1 included [D2] On Main Screen 00 excluded 1 included [D3] On Room Management System 00 excluded 1 included Returns: 09 CB 52 00 Parameters: None Example: 09 CB 52 04 01 01 01 01	 ADM

08 CB 52	Get the PIN display destinations – shows where the PINs are displayed Syntax: 08 CB 52 00 Variables: None Returns: 08 CB 52 04 D0 D1 D2 D3 Parameters: [D0] On Visualizer display 00 excluded 1 included [D1] On Cynap display 00 excluded 1 included [D2] On Main Screen 00 excluded 1 included [D3] On Room Management System 00 excluded 1 included [D4] On HDMI2 Out 00 excluded 1 included Example: >> 08 CB 52 00 << 08 CB 52 04 01 01 01 01	
09 CB 91	Activate or deactivate Moderator Login PIN Syntax: 09 CB 91 01 D0 Variables: [D1] 00 Hide PIN, 01 Show PIN Returns: 09 CB 91 00 Parameters: None Example: 09 CB 91 01 01	
08 CB 53	Get Moderator Login PIN and PIN display status Syntax: 08 CB 53 00 Variables: None Returns: 0C CB 53 AB D0 D1 … Dn Parameters: [AB] sizeof PIN + 1 [D0] 00 PIN hidden, 01 PIN visible [D1] … [D4] PIN Example >> 0C CB 53 00 << 0C CB 53 05 01 01 02 03 04	

09 CB D6	Activate/Deactivate AirPlay PIN Syntax: 09 CB D6 01 D1 Variables: [D1] 00 Deactivate, 01 Activate PIN Returns: 09 CB D6 00 Parameters: None Example: 09 CB D6 01 01	 ADM
08 CB E4	Get AirPlay PIN and PIN display status Syntax: 08 CB E4 00 Variables: None Returns: 0C CB E4 AB D0 D1…Dn Parameters: [AB] sizeof PIN + 1 [D0] 00 Hide PIN, 01 Show PIN, 02 PIN Popup [D1]…[Dn] PIN Example: >> 08 CB E4 00 << 0C CB E4 05 02 02 02 02	 MOD
09 CB CD	Activate/Deactivate Annotation PIN Syntax: 09 CB CD 01 D0 Variables: [D0] 00 Deactivate, 01 Activate PIN Returns: 09 CB CD 00 Parameters: None Example: 09 CB CD 01 01	 ADM

08 CB CE	Get Annotation PIN and PIN display status Syntax: 08 CB CE 00 Variables: None Returns: 08 CB CE AB D0 D1…Dn Parameters: [AB] sizeof PIN + 1 [D0] 00 Hide PIN, 01 Show PIN, 02 PIN Popup [D1]…[Dn] PIN Example: >> 08 CB CE 00 << 0C CB CE 05 02 02 02 02	 MOD
08 CB E7	Retrieve Login PIN and PIN display status on HDMI2 Syntax: 08 CB E7 00 Variables: None Returns: 0C CB E7 AB D0 D1…Dn Parameters: [AB] sizeof PIN + 1 [D0] 00 Hide PIN, 01 Show PIN, 02 PIN Popup [D1]…[Dn] PIN Example: >> 08 CB E7 00 << 0C CB E7 05 02 02 02 02	 MOD
08 CB 54	Retrieve Login PIN and PIN display status on your implementation Syntax: 08 CB 54 00 Variables: None Returns: 0C CB 54 AB D0 D1…Dn Parameters: [AB] sizeof PIN + 1 [D0] 00 Hide PIN, 01 Show PIN, 02 PIN Popup [D1]…[Dn] PIN Example: >> 08 CB 54 00 << 0C CB 54 05 02 02 02 02	 MOD

09 CB 40	Set or overwrite moderator user password – maximum of 63 bytes allowed – passwords are case sensitive! Syntax: 09 CB 40 AB D0..Dn Variables: [AB] size of password [D0]…[Dn] password String (zero terminated) Returns: 09 CB 40 00 Parameters: none Example: 09 CB 40 09 P A S S W O R D 00	
08 CB 40	Get moderator user password Syntax: 08 CB 40 00 Variables: None Returns: 08 CB 3F AB D0…Dn Parameters: [AB] size of password [D0]…[Dn] password String (zero terminated) Example: >> 08 CB 40 00 << 08 CB 40 09 P A S S W O R D 00	
09 CB 3F	Set or overwrite administrator password – maximum of 63 bytes allowed – passwords are case sensitive! Syntax: 09 CB 3F AB D0..Dn Variables: [AB] size of password [D0]…[Dn] password String (zero terminated) Returns: 09 CB 3F 00 Parameters: none Example: 09 CB 3F 09 P A S S W O R D 00	

08 CB 3F	Get administrator password Syntax: 08 CB 3F 00 Variables: None Returns: 08 CB 3F AB D0…Dn Parameters: [AB] size of password [D0]…[Dn] password String (zero terminated) Example: >> 08 CB 3F 00 << 08 CB 3F 09 P A S S W O R D 00	 ADM
09 CB D0	Activate or deactivate the Room Management System User If you activate this user, then please make sure that you also follow up with issuing a password immediately (using command 09 CB CF). Syntax: 09 CB CD 01 D1 Variables: [D1] 00 Deactivate, 01 Activate PIN Returns: 09 CB CD 00 Parameters: None Example: 09 CB CD 01 01	 ADM
08 CB D0	Verify if Room Management System (RMS) user has been activated or not. Syntax: 08 CB D0 00 Variables: None Returns: 08 CB D0 01 D0 Parameters: [D0] 00 Deactivated, 01 Activated Example: >> 08 CB D0 00 << 08 CB D0 01 01	 ADM

09 CB CF	Set or overwrite Room Management System (RMS) user password – maximum of 63 bytes allowed Syntax: 09 CB CF AB D0..Dn Variables: [AB] size of password [D0]…[Dn] password String (zero terminated) Returns: 09 CB CF 00 Parameters: none Example: 09 CB CF 09 P A S S W O R D 00	 ADM
08 CB CF	Get administrator password Syntax: 08 CB 3F 00 Variables: None Returns: 08 CB 3F AB D0…Dn Parameters: [AB] size of password [D0]…[Dn] password String (zero terminated) Example: >> 08 CB 3F 00 << 08 CB 3F 09 P A S S W O R D 00	 ADM
09 CB 42	Cynap Login – as soon as you configure a moderator or a Room Management System user, you need to log in to execute AccessLevel user commands. Syntax: 08 CB 3F 00 Variables: None Returns: 08 CB 3F AB D0…Dn Parameters: [AB] size of password [D0]…[Dn] password String (zero terminated) Example: >> 08 CB 3F 00 << 08 CB 3F 09 P A S S W O R D 00	 ADM  MOD

09 CB EE	Admin Log out – simply disconnects the logged in administrator Syntax: 09 CB EE 00 Variables: None Returns: 09 CB EE 00 Parameters: None Example: 09 CB EE 00	
09 CB 48	When Cloud services are activated and a moderator added his/her credentials, then these credentials can be saved as user data on a USB memory stick. User.data is being saved on the root directory of a connected USB memory stick. Syntax: 09 CB 48 00 Variables: None Returns: 09 CB 48 00 Parameters: None Example: 09 CB 48 00	
09 CB 8B	Loads Cloud credentials from an existing user.data file. Syntax: 09 CB 48 00 Variables: None Returns: 09 CB 48 00 Parameters: None Example: 09 CB 48 00	

09 CB 8E	Front panel lock, enables or disables a login feature on the front panel. It forces to enter the administrator´s password before changes are applied. Keep in mind that only a limited set of non-alphanumerical signs are available. Available characters are: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz ÄäÖöÜü 0123456789_-:$& () Syntax: 09 CB 8E 01 D0 Variables: [D0] 00 front panel unlocked, 01 front panel locked Returns: 09 CB 8E 00 Parameters: None Example: 09 CB 8E 01 00	 ADM
09 CB AC	USB storage mounting enabled or disabled – if disabled, storage connected via Cynap´s USB ports are not mounted. Syntax: 09 CB AC 01 D0 Variables: [D0] 00 disabled, 01 enabled Returns: 09 CB AC 00 Parameters: None Example: 09 CB AC 01 01	 ADM

Cynap Audio Commands		UserLevel
09 CB 56	Set Cynap´s Master Volume in percent Syntax: 09 CB 56 01 d0 Variables: D0 Returns: 09 CB 56 00 Parameters: None Example: 09 CB 56 01 32	 MOD

08 CB 56	Get Master Volume in percent Syntax: 08 CB 56 00 Variables: None Returns: 08 CB 56 01 D0 Parameters: [D0] volume in percent Example: >> 08 CB 56 00 << 08 CB 56 01 32	
09 CB AF	Set Line Out volume in percent Syntax: 09 CB AF 01 D0 Variables: [D0] volume in percent Returns: 09 CB AF 00 Parameters: None Example: 09 CB AF 01 32	
08 CB AF	Get Line Out Volume in percent Syntax: 08 AF AF 00 Variables: None Returns: 08 CB AF 01 D0 Parameters: [D0] volume in percent Example: >> 08 CB AF 00 << 08 CB AF 01 32	
09 CB B0	Set HDMI1 Out volume in percent Syntax: 09 CB B0 01 D0 Variables: [D0] volume in percent Returns: 09 CB B0 00 Parameters: None Example: 09 CB B0 01 32	

08 CB B0	Get HDMI1 Out Volume in percent Syntax: 08 CB B0 00 Variables: None Returns: 08 CB B0 01 D0 Parameters: [D0] volume in percent Example: >> 08 CB B0 00 << 08 CB B0 01 32	ADM
09 CB B1	Set HDMI2 Out volume in percent Syntax: 09 CB B1 01 D0 Variables: [D0] volume in percent Returns: 09 CB B1 00 Parameters: None Example: 09 CB B1 01 32	ADM
08 CB B1	Get HDMI2 Out Volume in percent Syntax: 08 CB B1 00 Variables: None Returns: 08 CB B1 01 D0 Parameters: [D0] volume in percent Example: >> 08 CB B1 00 << 08 CB B1 01 32	ADM
09 CB E8	Set Line In Level Syntax: 09 CB E8 01 D0 Variables: [D0] level in percent Returns: 09 CB E8 00 Parameters: None Example: 09 CB E8 01 32	ADM

08 CB E8	Get Line In Level in percent Syntax: 08 CB E8 00 Variables: None Returns: 08 CB E8 01 D0 Parameters: [D0] volume in percent Example: >> 08 CB E8 00 << 08 CB E8 01 32	
09 CB DD	Enable/Disable Line In mixed with local output. Switch on or off a line in connected audio device such as a microphone on HDMI output Syntax: 09 CB DD 01 D0 Variables: [D0] 00 mix disabled, 01 mix enabled Returns: 09 CB DD 00 Parameters: None Example: 09 CB DD 01 01	
08 CB DD	Get status of Line In mixed with local output Syntax: 08 CB DD 00 Variables: None Returns: 08 CB DD 01 D0 Parameters: [D0] 00 mix disabled, 01 mix enabled Example: >> 08 CB DD 00 << 08 CB DD 01 01	
09 CB DE	Enable/Disable Line In mixed with remote output. Switch on or off a line in connected device such as a microphone on recording/streaming output Syntax: 09 CB DE 01 D0 Variables: [D0] 00 mix disabled, 01 mix enabled Returns: 09 CB DE 00 Parameters: None Example: 09 CB DE 01 01	

08 CB DE	Get status of Line In mixed with remote output. Syntax: 08 CB DE 00 Variables: None Returns: 08 CB DE 01 D0 Parameters: [D0] 00 mix disabled, 01 mix enabled Example: >> 08 CB DE 00 << 08 CB DE 01 01	
09 CB DF	Enable/Disable USB Microphone Syntax: 09 CB DF 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB DF 00 Parameters: None Example: 09 CB DF 01 01	
08 CB DF	Get status if USB Microphone got enabled/disabled Syntax: 08 CB DF 00 Variables: None Returns: 08 CB DF 01 D0 Parameters: [D0] 00 mix disabled, 01 mix enabled Example: >> 08 CB DF 00 << 08 CB DF 01 01	
09 CB E0	Enable/disable USB microphone to be mixed with WebRTC. Prior to enable/disable this option, USB Microphone has to be enabled (09 CB DF 01 01). Syntax: 09 CB E0 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB E0 00 Parameters: None Example: 09 CB E0 01 01	

08 CB E0	Get status if USB Microphone got enabled/disabled Syntax: 08 CB E0 00 Variables: None Returns: 08 CB E0 01 D0 Parameters: [D0] 00 mix disabled, 01 mix enabled Example: >> 08 CB E0 00 << 08 CB E0 01 01	 ADM
09 CB 58	Set Master Mute Syntax: 09 CB 58 01 D0 Variables: [D0] 00 off, 01 on, 02 toggle Returns: 09 CB 58 00 Parameters: None Example: 09 CB 58 01 01	 MOD
08 CB 58	Get Master Mute Syntax: 08 CB 58 00 Variables: None Returns: 08 CB 58 01 D0 Parameters: [D0] 00 off, 01 on, 03 toggled Example: >> 08 CB 58 00 << 08 CB 58 01 01	 MOD
09 CB 59	Set Microphone Mute – USB microphone and line in source are being muted. Syntax: 09 CB 59 01 D0 Variables: [D0] 00 off, 01 on, 02 toggle Returns: 09 CB 59 00 Parameters: None Example: 09 CB 59 01 01	 MOD

08 CB 59	Get Microphone Mute status Syntax: 08 CB 59 00 Variables: None Returns: 08 CB 59 01 D0 Parameters: [D0] 00 off, 01 on, 02 toggled Example: >> 08 CB 59 00 << 08 CB 59 01 01	
09 CB E3	Play audio test sample Syntax: 09 CB E3 01 D0 Variables: [D0] 01 start playing Returns: 09 CB E3 00 Parameters: None Example: 09 CB E3 01 01	

Cynap Streaming Commands		UserLevel
09 CB 20	Start or stop Unicast/Multicast Audio/Video-Streaming Streaming in 4K not supported Syntax: 09 CB 20 01 D0 Variables: [D0] 00 stop streaming, 01 start streaming Returns: 09 CB 20 00 Parameters: None Example: 09 CB 20 01 01	

08 CB 20	Verify if Unicast/Multicast streaming is running Syntax: 08 CB 20 00 Variables: None Returns: 08 CB 20 01 D0 Parameters: [D0] 00 streaming stopped, 01 streaming in progress Example: >> 08 CB 20 00 << 08 CB 20 01 01	 MOD
09 CB 93	Activates or deactivates Audio/Video-Streaming functionality – if the administrator deactivates streaming then the “Start Streaming” button on the toolbar will disappear. Syntax: 09 CB 93 01 D0 Variables: [D0] 00 disable streaming, 01 enable streaming Returns: 09 CB E3 00 Parameters: None Example: 09 CB 93 01 01	 MOD
08 CB 93	Verify if Unicast/Multicast Streaming functionality is enabled or disabled Syntax: 08 CB 93 00 Variables: None Returns: 08 CB 93 01 D0 Parameters: [D0] 00 streaming disabled, 01 streaming enabled Example: >> 08 CB 93 00 << 08 CB 93 01 01	 MOD
09 CB 23	Change resolution of Unicast/Multicast streaming. Syntax: 09 CB 23 01 D0 Variables: [D0] 00 Full HD – 1080p, 02 HD – 720p, 03 qHD – 540p, 04 nHD – 360p Returns: 09 CB 23 00 Parameters: None Example: 09 CB 23 01 00	 ADM

08 CB 23	Get resolution of Unicast/Multicast streaming. Syntax: 08 CB 23 00 Variables: None Returns: 08 CB 23 01 D0 Parameters: [D0] 00 Full HD – 1080p, 02 HD – 720p, 03 qHD – 540p, 04 nHD – 360p Example: >> 08 CB 23 00 << 08 CB 23 01 01	ADM
09 CB 24	Change frame rate of Unicast/Multicast streaming. Syntax: 09 CB 24 01 D0 Variables: [D0] 00 Low (10 fps), 01 Medium (20 fps), 02 High (30 fps) Returns: 09 CB 24 00 Parameters: None Example: 09 CB 24 01 00	ADM
08 CB 24	Get frame rate of Unicast/Multicast streaming. Syntax: 08 CB 24 00 Variables: None Returns: 08 CB 24 01 D0 Parameters: [D0] 00 Low (10 fps), 01 Medium (20 fps), 02 High (30 fps) Example: >> 08 CB 24 00 << 08 CB 24 01 01	ADM
09 CB D4	Activate or deactivate Real Time Transport Protocol (RTP) streaming mode. If RTP mode is activated, please make sure that Real Time Streaming Protocol (RTSP) mode is deactivated. Syntax: 09 CB D4 01 D0 Variables: [D0] 00 RTP mode OFF, 01 RTP mode ON Returns: 09 CB D4 00 Parameters: None Example: 09 CB D4 01 00	ADM

08 CB D4	Check if Real Time Transport Protocol streaming mode is on or off. Syntax: 08 CB D4 00 Variables: None Returns: 08 CB D4 01 D0 Parameters: [D0] 00 RTP mode OFF, 01 RTP mode ON Example: >> 08 CB D4 00 << 08 CB D4 01 01	
09 CB 21	Set RTP streaming IP address. Unlike setting the IP address via remote or web browser interface, the IP address won´t be validated. Syntax: 09 CB 21 04 D0 D1 D2 D3 Variables: [D0] first octet, [D1] second octet, [D2] third octet, [D3] fourth octet Returns: 09 CB 21 00 Parameters: None Example: 09 CB 21 04, E1, 00, 00, 32 //IP: 225.0.0.50	
08 CB 21	Receive RTP streaming IP address. Syntax: 08 CB 21 00 Variables: None Returns: 08 CB 21 04 D0 D1 D2 D3 Parameters: [D0] first octet, [D1] second octet, [D2] third octet, [D3] fourth octet Example: >> 08 CB 21 00 << 08 CB 21 04, E1, 00, 00, 32 //IP: 225.0.0.50	

09 CB 22	Set RTP streaming video port number. Allowed and reserved streaming ports are in the range of Port 8800 to 9000 (0x2260 to 0x 2328). Audio port number will be automatically generated based on your port number (video port number + 2). Unlike setting the IP address via remote or web browser interface, the IP address won´t be validated. Syntax: 09 CB 21 02 D0 D1 Variables: [D0] [D1] byte-pair to represent 8800 to 9000 Returns: 09 CB 22 00 Parameters: None Example: 09 CB 22 02 22 62 //Port: 8802	
08 CB 22	Receive RTP streaming video port number. The audio port number resides in the system as the received video port number + 2 (e.g video port 8800, audio port 8802). Syntax: 08 CB 22 00 Variables: None Returns: 08 CB 22 02 D0 D1 Parameters: [D0] [D1] byte-pair to represent 8800 to 9000 Example: >> 08 CB 22 00 << 08 CB 22 02 22 62 //Port: 8802	
09 CB 5C	Set the RTP streaming network interface. Using RTP you can send a multicast stream on LAN or a unicast stream on LAN or Wi-Fi. Syntax: 09 CB 5C 01 D0 Variables: [D0] 00 streaming on LAN, 01 streaming on Wi-Fi Returns: 09 CB 5C 00 Parameters: None Example: 09 CB 5C 01 00	

08 CB 5C	Check if Real Time Transport Protocol streaming mode is sent over LAN (Multicast) or WI-FI (Unicast). Syntax: 08 CB 5C 00 Variables: None Returns: 08 CB 5C 01 D0 Parameters: [D0] 00 RTP mode OFF, 01 RTP mode ON Example: >> 08 CB 5C 00 << 08 CB 5C 01 01	ADM
09 CB D9	Set the RTP Time to Live (TTL) parameter on Unicast. Syntax: 09 CB D9 01 D0 Variables: [D0] 1 to 255 Returns: 09 CB D9 00 Parameters: None Example: 09 CB D9 01 00	ADM
08 CB D9	Set the RTP Time to Live (TTL) parameter on Unicast. Syntax: 08 CB D9 00 Variables: none Returns: 08 CB D9 01 D0 Parameters: D0: 1 to 255 Example: >> 08 CB D9 00 Receive: << 08 CB 9D 01 01	ADM
09 CB DA	Set the RTP Time to Live (TTL) parameter on Multicast. Syntax: 09 CB D9 01 D0 Variables: [D0] 1 to 255 Returns: 09 CB D9 00 Parameters: None Example: 09 CB D9 01 00	ADM

08 CB D9	Set the RTP Time to Live (TTL) parameter on Multicast. Syntax: 08 CB D9 00 Variables: none Returns: 08 CB D9 01 D0 Parameters: D0: 1 to 255 Example: >> 08 CB D9 00 Receive: << 08 CB 9D 01 01	ADM
09 CB D5	Activate or deactivate Real Time Transport Protocol (RTP) streaming mode. If RTP mode is activated, please make sure that Real Time Streaming Protocol (RTSP) mode is deactivated. Syntax: 09 CB D5 01 D0 Variables: [D0] 00 RTP mode OFF, 01 RTP mode ON Returns: 09 CB D5 00 Parameters: None Example: 09 CB D5 01 00	ADM
08 CB D5	Check if Real Time Streaming Protocol streaming mode is on or off. Syntax: 08 CB D5 00 Variables: None Returns: 08 CB D5 01 D0 Parameters: [D0] 00 RTP mode OFF, 01 RTP mode ON Example: >> 08 CB D5 00 << 08 CB D5 01 01	ADM

09 CB E5	Set the RTSP streaming network interface. Using RTSP you can send a multicast or unicast over LAN and Wi-Fi. Syntax: 09 CB E5 01 D0 Variables: [D0] 00 streaming on LAN, 01 streaming on Wi-Fi Returns: 09 CB E5 00 Parameters: None Example: 09 CB E5 01 00	
08 CB E5	Check on which interface Real Time Transport Protocol streaming is sent. Syntax: 08 CB E5 00 Variables: None Returns: 08 CB E5 01 D0 Parameters: [D0] 00 RTP mode OFF, 01 RTP mode ON Example: >> 08 CB E5 00 << 08 CB E5 01 01	

Cynap Video Recording Commands		UserLevel
09 CB 25	Start or stop video recording Syntax: 09 CB 25 05 D0 D1 D2 D3 D4 Variables: [D0] 00 stop recording, 01 pause/resume recording, 02 stop recording, 03 recording Returns: 09 CB 25 00 Parameters: None Example: 09 CB 25 01 01	

08 CB 25	Get info on video recording in progress Syntax: 08 CB 25 00 Variables: None Returns: 08 CB 25 05 D0 D1 D2 D3 D4 Parameters: [D0] 00 stop recording, 01 pause/resume recording, 02 stop recording, 03 recording Opencast [D1]…[D4] recording time in ms Example: >> 08 CB 25 00 << 08 CB 25 01 01	
09 CB 26	Set the recording resolution – from nHD up to FullHD 4K recording not available Syntax: 09 CB 26 01 D0 Variables: [D0] 00 Full HD, 01 HD, 02 qHD, 03 nHD Returns: 09 CB 26 00 Parameters: None Example: 09 CB 26 01 00	
08 CB 26	Query on resolution of video recording 4K recording not available Syntax: 08 CB 26 00 Variables: None Returns: 08 CB 26 01 D0 Parameters: [D0] 00 Full HD, 01 HD, 02 qHD, 03 nHD Example: >> 08 CB 26 01 00 << 08 CB 26 01 02	

09 CB 94	Set the frame rate of the video recording. Low equals 10 fps, medium 20 fps and high 30 fps. Syntax: 09 CB 24 01 D0 Variables: [D0] 00 low, 01 medium, 02 high Returns: 09 CB 24 00 Parameters: None Example: 09 CB 24 01 00	ADM
08 CB 94	Get the frame rate information of video recording. Low equals 10 fps, medium 20 fps and high 30 fps. Syntax: 08 CB 24 00 Variables: None Returns: 08 CB 24 01 D0 Parameters: [D0] 00 low, 01 medium, 02 high Example: >> 08 CB 24 01 00 << 08 CB 24 01 02	ADM
09 CB 92	Enable or disable video recording function Syntax: 09 CB 92 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB 92 00 Parameters: None Example: 09 CB 92 01 00	ADM
08 CB 92	Check if video recording is enabled Syntax: 08 CB 92 00 Variables: None Returns: 08 CB 92 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: >> 08 CB 92 00 << 08 CB 92 01 01	ADM

Cynap WebRTC Commands		UserLevel
09 CB B9	Enable or disable WebRTC Syntax: 09 CB B9 01 D0 Variables: [D0] 00 disable WebRTC, 01 enable WebRTC Returns: 09 CB B9 00 Parameters: None Example: 09 CB B9 01 01	ADM
08 CB B9	Check if WebRTC has been switched off or not. Syntax: 08 CB B9 00 Variables: None Returns: 08 CB B9 01 D0 Parameters: [D0] 00 disable WebRTC, 01 enable WebRTC Example: >> 08 CB B9 00 << 08 CB B9 01 01	ADM
09 CB B8	Set up URL of WebRTC also called Webconferencing Room. Syntax: 09 CB B8 Ab D0…Dn Variables: [AB]: sizeof URL [D0]…[DN] URL Returns: 09 CB Ba 00 Parameters: None Example: >> 09 CB B8 10 w e b r t c . o r g << 09 CB B8 00	ADM

09 CB BB	WebRTC Camera Autostart, set setting to automatically activate the camera when the presenter starts a WebRTC session. Syntax: 09 CB 26 01 D0 Variables: [D0] 00 Full HD, 01 HD, 02 qHD, 03 nHD Returns: 09 CB 26 00 Parameters: None Example: 09 CB 26 01 00	
08 CB 26	Check if video recording function is enabled or disabled Syntax: 08 CB 26 00 Variables: None Returns: 08 CB 26 01 D0 Parameters: [D0] **00** Full HD, **01** HD, **02** qHD, **03** nHD Example: >> 08 CB 26 01 00 << 08 CB 26 01 02	

Output HDMI1 and HDMI2		UserLevel
09 CB 4D	Sets the resolution of HDMI 1 Output Syntax: 09 CB 4D 01 D0 Variables: [D0] **00** Automatically detected, **01** 1080p60, **02** 2160p30, **03** 2160p60 Returns: 09 CB 4D 00 Parameters: None Example: 09 CB 4D 01 01	

<table>
<tr>
<td>08 CB 4D</td>
<td>Gets the resolution of HDMI 1 Output

Syntax:
08 CB 4D 00

Variables:
None

Returns:
08 CB 4D 01 D0

Parameters:
[D0] 00 Automatically detected, 01 1080p60, 02 2160p30, 03 2160p60

Example:
>> 08 CB D4 00
<< 08 CB D4 01 01</td>
<td>ADM</td>
</tr>
<tr>
<td>09 CB 5B</td>
<td>Setting of Output Mode/Resolution of HDMI 2
There are 3 terms defining the possible modes:
<table>
<tr><td>Mirror</td><td>a digital copy of HDMI 1 Output in following resolutions
1:1 same as HDMI 1 Output
1080p FullHD or 720p HD</td></tr>
<tr><td>Content</td><td>single window display of pre-configured or set window without top status bar – this window will not be streamed or recorded
annotation not available on HDMI2 output</td></tr>
<tr><td>Moderator</td><td>single window display of pre-configured or set window including the top status bar – just this window will be streamed or recorded
annotation available on HDMI2 output</td></tr>
</table>
Note: 4K Output on HDMI 2 output only supported if HDMI 1 output is 4K too.

Syntax:
09 CB 5B 01 D0

Variables:
[D0] 00 Mirror 1:1, 01 Mirror 1080p60, 02 Mirror 720p60, 03 Content 1080p60,
04 Content 720p60, 05 Moderator 1080p60

Returns:
09 CB 5B 00

Parameters:
None

Example:
09 CB 5B 01 00</td>
<td>ADM</td>
</tr>
</table>

<table>
<tr>
<td>08 CB 5B</td>
<td>Get settings of Output Mode/Resolution of HDMI 2

There are 3 terms defining the possible modes:
<table>
<tr><td>Mirror</td><td>a digital copy of HDMI 1 Output in following resolutions
1:1 same as HDMI 1 Output
1080p FullHD or 720p HD</td></tr>
<tr><td>Content</td><td>single window display of pre-configured or set window without top status bar – this window will not be streamed or recorded
annotation not available on HDMI2 output</td></tr>
<tr><td>Moderator</td><td>single window display of pre-configured or set window including the top status bar – just this window will be streamed or recorded
annotation available on HDMI2 output</td></tr>
</table>
Note: 4K Output on HDMI 2 output only supported if HDMI 1 output is 4K too.

In Content or Moderator mode: To send a window to HDMI2 you need to use API Window Control, 09 CB 28.

Syntax:
08 CB 5B 00

Variables:
None

Returns:
09 CB 5B 01 D0

Parameters:
[D0] 00 Mirror 1:1, 01 Mirror 1080p60, 02 Mirror 720p60, 03 Content 1080p60,
04 Content 720p60, 05 Moderator 1080p60

Example:
>> 08 CB 5B 00
<< 08 CB 5B 01 00</td>
<td> 
</td>
</tr>
<tr>
<td>09 CB D7</td>
<td>Enable HDMI 2 Mirror Override in Moderator Mode
To offer the moderator the possibility to toggle between Mirroring and Moderator mode, the HDMI 2 Mirror Overator mode needs to be enabled.

Syntax:
09 CB D7 01 D0

Variables:
[D1] 00 Disable, 01 Enable
None

Returns:
09 CB D7 00

Parameters:
None

Example:
>> 09 CB D7 01 01
<< 09 CB D7 00</td>
<td> 
</td>
</tr>
</table>

08 CB D7	Check if HDMI 2 Mirror Override in Moderator Mode is enabled/disabled Syntax: 08 CB D7 00 Variables: None Returns: 08 CB D7 01 D0 Parameters: [D1] **00** Disable, **01** Enable Example: >> 08 CB D7 09 << 09 CB D7 01 D0	
09 CB D8	Toggle HDMI2 Mirror Override Mirror Mode in Moderator Mode To offer the moderator the possibility to toggle between Mirroring and Moderator mode, the HDMI 2 Mirror Override mirror mode needs to be enabled (09 CB D7/08 CB D7). Syntax: 09 CB D8 01 D0 Variables: [D1] **00** Start mirror override, **01** stop mirror override Returns: 09 CB D8 00 Parameters: None Example: >> 09 CB D8 01 00	
08 CB D8	Check HDMI2 Mirror Override Mode in Moderator Mode started/stopped Check, if HDMI2 Mirror Override has been started or stopped Syntax: 09 CB D8 00 Variables: None Returns: 09 CB D8 01 D0 Parameters: [D1] **00** Mirror override started, **01** mirror override stopped Example: >> 08 CB D8 00 << 08 CB D8 01 01	

<table>
<tr>
<td>09 CB B3</td>
<td>
Three slots of pre-defined sources are customizable to be automatically sent to HDMI2 without sending a windows output control command to change the HDMI destination (09 CB 28 D0 06, Window Control). Such a source will automatically be sent to HDMI 2.

Syntax:
09 CB B3 02 D0 D1

Variables:
[D0] Slot # (00 to 02)
[D1] Source type
<table>
<tr><td>00 None</td><td>01 Visualizer</td><td>02 HDMI input</td><td>03 Browser</td><td>04 Miracast</td></tr>
<tr><td>05 AirPlay</td><td>06 Video</td><td>07 vSolution Cast</td><td>08 Image</td><td>09 PDF</td></tr>
<tr><td>10 Office presentation</td><td>11 Office text</td><td>12 Office spreadsheet</td><td>13 Whiteboard</td><td>14 Audio</td></tr>
<tr><td>15 WebRTC</td><td>16 Webcam</td><td></td><td></td><td></td></tr>
</table>
Returns:
09 CB B3 00

Parameters:
None

Example:
Customize the 3 available slots with video, image and whiteboard sources
>> 09 CB B3 02 00 06
<< Receive 09 CB B3 00
>> 09 CB B3 02 00 08
<< Receive 09 CB B3 00
>> 09 CB B3 02 00 13
<< Receive 09 CB B3 00
</td>
<td>ADM</td>
</tr>
<tr>
<td>08 CB B3</td>
<td>
Three slots of pre-defined sources are customizable to be automatically sent to HDMI2 without sending a windows output control command to change the HDMI destination (09 CB 28 D0 06, Window Control). Such a source will automatically be sent to HDMI 2.

Default values are set to 00, none even when HDMI Mode has not been set to Content Mode.

Syntax:
08 CB B3 01 D0

Variables:
[D0] Slot # (00 to 02)
[D1] Source type
<table>
<tr><td>00 None</td><td>01 Visualizer</td><td>02 HDMI input</td><td>03 Browser</td><td>04 Miracast</td></tr>
<tr><td>05 AirPlay</td><td>06 Video</td><td>07 vSolution Cast</td><td>08 Image</td><td>09 PDF</td></tr>
<tr><td>10 Office presentation</td><td>11 Office text</td><td>12 Office spreadsheet</td><td>13 Whiteboard</td><td>14 Audio</td></tr>
<tr><td>15 WebRTC</td><td>16 Webcam</td><td></td><td></td><td></td></tr>
</table>
Returns:
09 CB B3 00

Parameters:
None

Example:
Customize the 3 available slots with video, image and whiteboard sources
>> 09 CB B3 02 00 06
<< Receive 09 CB B3 00
>> 09 CB B3 02 00 08
<< Receive 09 CB B3 00
>> 09 CB B3 02 00 13
<< Receive 09 CB B3 00
</td>
<td>ADM</td>
</tr>
</table>

09 CB 79	Set HDMI Input Sources Cynap offers 2 HDMI input sources – the labelling of these sources by name and type shows an icon and a name to identify each source. The name itself is limited to 16 characters. Syntax: 09 CB 79 AB D0 D1 D2…Dn Variables: [AB] **size** of variables (e.g. name + 2) [D0] Input, **00** HDMI Input 1, **01** HDMI Input 2 [D1] Type, **00** Visualizer, **01** Generic HDMI, **02** Computer, **03** Disc Player [D2]…[Dn] **Name** (max. 16 chars). Returns: 09 CB 79 00 Parameters: None Example: 09 CB 79 10 00 01 A B C D E F G H	
08 CB 79	Get HDMI Input Sources Cynap offers 2 HDMI input sources – the labelling of these sources by name and type shows an icon and a name to identify each source. The name itself is limited to 16 characters. Syntax: 08 CB 79 00 Variables: None Returns: 08 CB 79 AB D0 D1 D2…Dn Parameters: [AB] **size** of variables (e.g. name + 2) [D0] Input, **00** HDMI Input 1, **01** HDMI Input 2 [D1] Type, **00** Visualizer, **01** Generic HDMI, **02** Computer, **03** Disc Player [D2]…[Dn] **Name** (max. 16 chars). Example: >> 08 CB 79 00 << 08 CB 79 10 00 01 08 CB 79 10 00 01 A B C D E F G H	
09 CB BC	Set HDMI input to automatically start if power is activated Set or unset HDMI input sources to open a window as soon as a signal on HDMI is detected. Syntax: 09 CB BC 02 D0 D1 Variables: [D0] Input, **00** HDMI Input 1, **01** HDMI Input 2 [D1] Start Window on signal detection, **00** off, **01** on Returns: 09 CB BC 00 Parameters: None Example: 09 CB BC 02 00 00	

08 CB BC	Check if selected HDMI input is set to automatically start if power is activated. Syntax: 08 CB BC 01 D0 Variables: [D0] Input, **00** HDMI Input 1, **01** HDMI Input 2 Returns: 08 CB BC 02 D0 D1 Parameters: [D0] Input, **00** HDMI Input 1, **01** HDMI Input 2 [D1] Start Window on signal detection, **00** off, **01** on Example: >> 08 CB BC 00 Receive >> 08 CB BC 02 00 00	
09 CB 6F	Activate/Deactivate HDCP Main Only available on models supporting HDCP out Syntax: 09 CB 6F 01D0 Variables: [D0] Output, **00** HDCP main off, **01** HDCP main on Returns: 09 CB 6F 00 Parameters: None Example: 09 CB 6F 01 01	
08 CB 6F	Check if HDCP Main has been activated/deactivated Only available on models supporting HDCP out Syntax: 08 CB 6F 00 Variables: None Returns: 08 CB 6F 01 D0 Parameters: [D0] Output, **00** HDCP main off, **01** HDCP main on Example: >> 08 CB 6F 00 Receive >> 08 CB 6F 01 01	

09 CB 70	Activate/Deactivate HDCP Aux Only available on models supporting HDCP out Syntax: 09 CB 70 01D0 Variables: [D0] Output, **00** HDCP aux off, **01** HDCP aux on Returns: 09 CB 70 00 Parameters: None Example: 09 CB 70 01 01	
08 CB 70	Check if HDCP Aux has been activated/deactivated Only available on models supporting HDCP out Syntax: 08 CB 70 00 Variables: None Returns: 08 CB 670F 01 D0 Parameters: [D0] Output, **00** HDCP out aux off, **01** HDCP out aux on Example: >> 08 CB 70 00 Receive >> 08 CB 70 01 01	
09 CB 71	Activate/Deactivate HDCP in 1 Only available on models supporting HDCP in Syntax: 09 CB 71 01D0 Variables: [D0] Input, **00** HDCP in 1 off, **01** HDCP in 1 on Returns: 09 CB 71 00 Parameters: None Example: 09 CB 71 01 01	

08 CB 71	Check if HDCP in 1 has been activated/deactivated Only available on models supporting HDCP in Syntax: 08 CB 71 00 Variables: None Returns: 08 CB 71 01 D0 Parameters: [D0] Input, **00** HDCP in 1 off, **01** HDCP in 1 on Example: >> 08 CB 71 00 Receive >> 08 CB 71 01 01	ADM
09 CB 72	Activate/Deactivate HDCP in 2 Only available on models supporting HDCP in Syntax: 09 CB 72 01D0 Variables: [D0] Input, **00** HDCP in 2 off, **01** HDCP 2 on Returns: 09 CB 72 00 Parameters: None Example: 09 CB 72 01 01	ADM
08 CB 72	Check if HDCP in 2 has been activated/deactivated Only available on models supporting HDCP in Syntax: 08 CB 72 00 Variables: None Returns: 08 CB 72 01 D0 Parameters: [D0] Input, **00** HDCP in 2 off, **01** HDCP in 2 on Example: >> 08 CB 72 00 Receive >> 08 CB 72 01 01	ADM

<table>
<tr><th>Window commands</th><th></th><th>UserLevel</th></tr>
<tr><td>08 CB 8D</td><td>Check if there is a free window available
Once the 4 windows slots are taken, no new window can be opened without another one being closed first.

This status request tells if a new window can be opened without causing a “You have reached the maximum number of free windows”-pop-up.

Syntax:
08 CB 8D 00

Variables:
None

Returns:
08 CB 8D 01 D0

Parameters:
[D0] windows, 00 no free window slot available, 01 free window slot available

Example:
>> 08 CB 8D 00
Receive >> 08 CB 8D 01 01</td><td>MOD</td></tr>
<tr><td>0D CB 2C</td><td>Starts a new window – the context menu will be based on your defined WindowType. WindowType 00, none, exists as an empty window and is used in querying the number of active windows (valid are 01 to 16).

All of the four possible windows are initialized and active as soon as Cynap starts. These four windows carry the WindowType none (00) and aren´t visible on the user interface. WindowType none is also the parameter to check if a specific window is available to use.

Syntax:
0D CB 2C AB CD D0 D1 D2 D3 D4…Dn

Variables:
[AB][CD] size of parameter (+4)
[D0] WindowID (FF for auto arrange)
[D1] WindowType
<table>
<tr><td>00 None</td><td>01 Visualizer</td><td>02 HDMI input</td><td>03 Browser</td><td>04 Miracast</td></tr>
<tr><td>05 AirPlay</td><td>06 Video</td><td>07 vSolution Cast</td><td>08 Image</td><td>09 PDF</td></tr>
<tr><td>0A Office presentation</td><td>0B Office text</td><td>0C Office spreadsheet</td><td>0D Whiteboard</td><td>0E Audio</td></tr>
<tr><td>0F WebRTC</td><td>10 Webcam</td><td></td><td></td><td></td></tr>
</table>
[D2][D3] size of following parameter
[D4]…[Dn] parameter (e.g. URL for browser)

Returns:
09 CB 2C 00

Parameters:
None

Example:
0D CB 2C 00 20 FF 08 00 1B file:///media/usb/image.jpg</td><td>MOD</td></tr>
</table>

Window commands	UserLevel

08 CB BA

Get information about all windows (hidden and active). Active and therefore visible windows are all windows except the ones carrying a 00 in field WindowType (none). This command returns common and content specific information, e.g. general information about its position on the screen but also the duration of a video in a window of WindowType 06 (video).

MOD

Syntax:
08 CB BA 00

Variables:
None

Returns:
General information about resolution and the size of the complete dataset and 4 times the common window and content type information in one dataset per window.

Parameters:
Common window information and specific content type information build a single block per window.

General	W1 General	W1 Specific	W2 General	W2 Specific	W3 General	W3 Specific	W4 General	W4 Specific

General information:
Size of total information data set, screen resolution

Common window information:
Size of common block, WindowType, WindowState, X Position, Y position (both from upper left corner), Width, Height, Mute (on or off), Volume (in percent), Shared with HDMI 2

Type specific block
Based on WindowType the specific information block various in size and amount of returned parameters.

Types

00 None	**01** Visualizer	**02** HDMI input	**03** Browser	**04** Miracast
05 AirPlay	**06** Video	**07** vSolution Cast	**08** Image	**09** PDF
0A Office presentation	**0B** Office text	**0C** Office spreadsheet	**0D** Whiteboard	**0E** Audio
0F WebRTC	**10** Webcam			

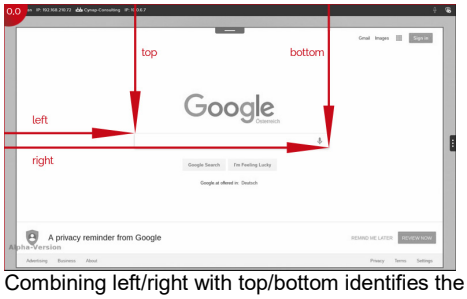
Type 00 None
Does not contain information, also common window information such as position are set to 0

Type 01 Visualizer and Type 02 HDMI input
[D0][D1][D2][D3] Index, [AB][CD] size of HDMI input name [E0]…[En]HDMI input name

Type 03 Internet Browser
[AB][CD] size of URL, [D0]…[Dn]URL, [AB][CD] size of title, [D1]…[Dn] title,
[D2] type of input entry field, any value beside **00** means that a keyboard is required and that the keyboard layout could be adapted (e.g. show @ in entry field of type email).

00 no input field	**01** text	**02** text area	**03** password	**04** number	**05** URL	**06** date	**07** month
08 week	**09** time	**0A** datetime	**0B** datetime local	**0C** email	**0D** search	**0E** phone	**FF** unknown field

[D3] PDF viewer activated [D4] [D5] [D6] [D7] page number of PDF
[D8] [D9] [DA] [DB] number of total pages of PDF
[DC] web page loading, **00** no, **01** yes
[DD][DE][DF][E0] distance in pixels from offset to top border
[E1][E2][E3][E4] distance in pixels from offset to bottom border
[E5][E5][E7][E8] distance in pixels from offset to left border
[E9][EA][EB][EC] distance in pixels from offset to right border



Combining left/right with top/bottom identifies the borders and corners of an input entry field.

Window commands | **UserLevel**

MOD

Type 04 and 05 Miracast and Airplay
[AB][CD] size of name
[D0]…[Dn] name

Type 06 Video
[D0][D1][D2][D3] duration in milliseconds
[D4][D5][D6][D7] position in milliseconds
[D8] speed: 00 paused, 01 normal, -1 reverse, 2 – 5 speed multiplier
[D9] looping: 00 off, 01 on

Type 07 vSolution Cast
[AB][CD] size of name
[D0]…[Dn] name

Type 08 image
[D0][D1][D2][D3] number of image
[D4][D5][D6][D7] total number of images
[D8][D9] size of file name
[DA]…[Dn] file name

Type 09 PDF
[D0][D1][D2][D3] number of page
[D4][D5][D6][D7] total number of pages

Type 0A, 0B and 0C, Office: Word, Presentation and Spreadsheets
All 3 Office formats carry the same structures although spreadsheets uses worksheets instead of pages and page numbers every content type of 0A, 0B and 0C

Type 0D Whiteboard
[D0] number of participating users

Type 0E Audio
[D0][D1][D2][D3] duration in milliseconds
[D4][D5][D6][D7] position in milliseconds
[D8] speed: 00 paused, 01 normal, -1 reverse, 2 – 5 speed multiplier
[D9] looping: 00 off, 01 on

Type 0F WebRTC
[AB][CD] size of URL, [D0]…[Dn]URL, [AB][CD] size of title, [D1]…[Dn] title,
[D2] type of input entry field, any value beside **00** means that a keyboard is required and that the keyboard layout could be adapted (e.g. show @ in entry field of type email).

00 no input field	**01** text	**02** text area	**03** password	**04** number	**05** URL	**06** date	**07** month
08 week	**09** time	**0A** datetime	**0B** datetime local	**0C** email	**0D** search	**0E** phone	**FF** unknown field

[D3] PDF viewer activated [D4] [D5] [D6] [D7] page number of PDF
[D8] [D9] [DA] [DB] number of total pages of PDF
[DC] web page loading, **00** no, **01** yes

[DD][DE][DF][E0] distance in pixels from offset to top border
[E1][E2][E3][E4] distance in pixels from offset to bottom border
[E5][E5][E7][E8] distance in pixels from offset to left border
[E9][EA][EB][EC] distance in pixels from offset to right border

Combining left/right with top/bottom identifies the borders and corners of an input entry field.

Type 10 Web Cam
No specific information returned

Example:
Single window open on Cynap – single window contains google.at as URL
>> 08 CB DA 00
<<
0C CB BA 00 D7 0F 00 08 70 0D 03 00 00 6A 00 64 0E 28 07 F6 00 50 00 00 93 00 3A 68 74 74 70 73 3A 2F 2F 77 77 77 2E 67 6F 6F 67 6C 65 2E 61 74 2F 3F 67 66 65 5F 72 64 3D 63 72 26 65 69 3D 4D 4B 36 49 57 4C 6D 68 44 49 66 5A 38 41 65 7A 7A 36 75 77 44 77 00 3A 68 74 74 70 73 3A 2F 2F 77 77 77 2E 67 6F 6F 67 6C 65 2E 61 74 2F 3F 67 66 65 5F 72 64 3D 63 72 26 65 69 3D 4D 4B 36 49 57 4C 6D 68 44 49 66 5A 38 41 65 7A 7A 36 75 77 44 77 01 00 00 00 00 00 00 00 00 00 00 00 AF 3D 00 00 C1 AF 00 00 71 11 00 01 0A 92 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Window commands		UserLevel
09 CB 28	Manipulate one of the four open windows (e.g. maximise, close, etc.) in general – no WindowType specified actions available. Syntax: 09 CB 28 AB D0 D1 Variables: [AB] size of variable [D0] WindowID [D1] Command [D2] Value (if necessary)	

[AB]	[D0]	[D1]	[D2]	Action
02	WindowID	00		Close window
02		01		Fullscreen
02	WindowID	02		Normal
03	WindowID	03	**00** off | **01** on	Mute
03	WindowID	04	**0** to **100**	Volume in %
02	WindowID	05		Toggle full screen/normal
02	WindowID	06		Display window on HDMI 2
02	WindowID	07		Hide window on HDMI 2
02	WindowID	08		Toggle display/hide window on HDMI 2

Return:
09 CB 28 00

Parameters:
None

Example:
Close window number four (blue colored)

09 CB 28 02 03 00

Window commands		UserLevel
09 CB 29	Controlling the Visualizer window – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 09 CB 29 AB D0 D1 Variables: [AB] size of variable [D0] WindowID [D1] Command [D2] Value [D3] 2nd value pair	 MOD

[AB]	[D0]	[D1]	[D2]	[D3]	Action
04	WindowID	00	0x001…0x000f	0x001…0x000f	Zoom wide (speed 0x000 = Stop)
02		01	0x001…0x000f	0x001…0x000f	Zoom tele (speed 0x000 = Stop)
02	WindowID	02	0x001…0x000f	0x001…0x000f	Focus far (speed 0x000 = Stop)
03	WindowID	03	0x001…0x000f	0x001…0x000f	Zoom near (speed 0x000 = Stop)
03	WindowID	04			Autofocus on
02	WindowID	05			Autofocus off
02	WindowID	06			Freeze on
02	WindowID	07			Freeze off
02	WindowID	08			Preset set
02	WindowID	09			Preset recall

Return:
09 CB 29 00

Parameters:
None

Example:
Autofocus on a second window of WindowType Visualizer

09 CB 29 02 01 04

Window commands		UserLevel
0D CB 2A	Controlling the browser window – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB 2A AB CD D0 D1 D1 D2 D3 Variables: [AB] size of variable [D0] WindowID [D1] Command [D2] Value [D3] 2nd value pair	 MOD

[AB][CD]	[D0]	[D1]	[D2]	[D3]	Action
Dynamic	WindowID	00	URL [D2]…[Dn]		URL (max. 512 characters)
00 02	WindowID	01			Reload
00 02	WindowID	02			Stop
00 03	WindowID	03			Zoom in
00 03	WindowID	04			Zoom out
00 02	WindowID	05			Cursor down
00 02	WindowID	06			Cursor up
00 02	WindowID	07			Cursor right
00 02	WindowID	08			Cursor left
00 02	WindowID	09			Back
00 02	WindowID	0A			Forward
00 02	WindowID	0B			PDF: next page
00 02	WindowID	0B			PDF: previous page
00 04	WindowID	0C	page number	Page number	PDF: set page
00 02	WindowID	0E			PDF: full height
00 02	WindowID	0F			PDF: full width
00 02	WindowID	10			PDF: full page
00 02	WindowID	11			Scroll down
00 02	WindowID	12			Scroll up
00 02	WindowID	13			Scroll right
00 02	WindowID	14			Scroll left

Return:
0D CB 2A 00

Parameters:
None

Example:
Reload browser window
0D CB 2A 00 02 00 01

Window commands		UserLevel
0D CB 39	Controlling the image window – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB 39 AB CD D0 D1 D2 D3 Variables: [AB] size of variable [D0] WindowID [D1] Command [D2] Value [D3] 2nd value pair **[AB][CD] / [D0] / [D1] / [D2]…[Dn] / Action** Dynamic (Path + name + 2) / WindowID / 00 / Filepath + name / Open image (max. 512 characters) 00 02 / WindowID / 01 / / Previous image 00 02 / WindowID / 02 / / Next image 00 03 / WindowID / 03 / / First image 00 03 / WindowID / 04 / / Last image Return: 0D CB 39 00 Parameters: None Example: Open file image.jpg from USB memory stick on WindowID 00 0D CB 39 00 1D 00 f i l e : / / / m e d i a / u s b / i m a g e . j p g	MOD
0D CB 2B	Controlling the video window, opening, pausing, etc. – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB 2B AB CD D0 D1 D2 D3 Variables: [AB] size of variable [D0] WindowID [D1] Command [D2] Value …[Dn] increase on bytes (file name) **[AB][CD] / [D0] / [D1] / [D2]…[Dn] / Action** 00 02 / WindowID / 00 / / Play 00 02 / WindowID / 01 / / Pause 00 02 / WindowID / 02 / / Stop 00 03 / WindowID / 03 / / Forward 00 03 / WindowID / 04 / / Rewind Dynamic (Path + name + 2) / WindowID / 05 / Filepath + name / Open image (max. 512 characters) 00 06 / WindowID / 06 / D1 D2 D3 D4: positions in ms / Seek 00 02 / WindowID / 07 / / Loop off 00 02 / WindowID / 08 / / Loop on 00 02 / WindowID / 09 / / Loop toggle Return: 0D CB 39 00 Parameters: None Example: Loop video on WindowID 00 0D CB 2B 00 02 00 08	MOD

Window commands		UserLevel
0D CB A2	Controlling the audio window, playing etc. – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB A2 AB CD D0 D1 D2 Dn Variables: [AB] [CD] size of variable [D0] WindowID [D1] Command [D2] Value …[Dn] increase bytes (e.g. file names)	

[AB][CD]	[D0]	[D1]	[D2]…[Dn]	Action
00 02	WindowID	00		Play
00 02	WindowID	01		Pause
00 02	WindowID	02		Stop
00 03	WindowID	03		Forward
00 03	WindowID	04		Rewind
Dynamic (Path + name + 2)	WindowID	05	Filepath + name	Open image (max. 512 characters)
00 06	WindowID	06	D1 D2 D3 D4: positions in ms	Seek
00 02	WindowID	07		Loop off
00 02	WindowID	08		Loop on
00 02	WindowID	09		Loop toggle

Return:
0D CB 39 00

Parameters:
None

Example:
Pause audio track on WindowID 00
0D CB A2 00 02 00 01

Window commands		UserLevel
0D CB 4A	Controlling a PDF window, page turning etc. – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB 4A AB CD D0 D1 D2 Dn Variables: [AB] [CD] size of variable [D0] WindowID [D1] Command [D2] Value …[Dn] increase bytes (e.g. file names)	

[AB][CD]	[D0]	[D1]	[D2]…[Dn]	Action
Dynamic (Path + name + 2)	WindowID	00	Mount + path + name	Open PDF file (max. 512 characters)
00 02	WindowID	01		Zoom in
00 02	WindowID	02		Zoom out
00 02	WindowID	03		Cursor down
00 02	WindowID	04		Cursor up
00 02	WindowId	05		Cursor right
00 02	WindowID	06		Cursor left
00 02	WindowID	07		Next page
00 02	WindowID	08		Previous page
00 04	WindowID	09	D2, D3 page number	Set page number
00 02	WindowID	0A		Full height
00 02	WindowID	0B		Full width
00 02	WindowID	0C		Full page
00 02	WindowID	0D		Scroll down
00 02	WindowID	0E		Scroll up
00 02	WindowID	0F		Scroll right
00 02	WindowID	10		Scroll left

Return:
0D CB 4A 00

Parameters:
None

Example:
Use full width on PDF on WindowID 00
0D CB 4A 00 02 00 0B

Window commands		UserLevel
0D CB 4B	Controlling an office window (word, excel and powerpoint got combined into this single API), next worksheet/page etc. – for general window manipulations (non WindowType specified action) please use 09 CB 28. Syntax: 0D CB 4B AB CD D0 D1 D2 Dn Variables: [AB] [CD] size of variable [D0] WindowID [D1] Command [D2] Value …[Dn] increase bytes (e.g. file names)	 MOD

[AB][CD]	[D0]	[D1]	[D2]…[Dn]	Action
Dynamic (Path + name + 2)	WindowID	00	Mount + path + name	Open office file (max. 512 characters)
00 02	WindowID	01		Zoom in
00 02	WindowID	02		Zoom out
00 02	WindowID	03		Cursor down
00 02	WindowID	04		Cursor up
00 02	WindowId	05		Cursor right
00 02	WindowID	06		Cursor left
00 02	WindowID	07		Next page
00 02	WindowID	08		Previous page
00 02	WindowID	09		Next worksheet
00 02	WindowID	0A		Previous worksheet
00 02	WindowID	0B		Full page
00 02	WindowID	0C		Full width
00 02	WindowID	0D		Scroll down
00 02	WindowID	0E		Scroll up
00 02	WindowID	0F		Scroll right
00 02	WindowID	10		Scroll left

Return:
0D CB 4A 00

Parameters:
None

Example:
Use full width on word file on WindowID 00
0D CB 4B 00 02 00 0C

09 CB 2D	Sending a key on a specified window – for example sending a space bar event to a browser window where URL == youtube.com to pause/play a video. Syntax: 09 CB 2D 08 D0 D1 D2 D3 D4 D5 D6 D7 Variables: [D0] WindowID [D1] Action: **00** Key hit (key down), **01** Key tip (key down/up), **02** key release (key up), **03** move pointer relative, **04** move pointer absolute [D2][D3] key [D4][D5] x position [D6][D7] y position Returns: 09 CB 2D 08 Parameters: None Example: Send space bar (ASCII hex 20) to WindowID 00 09 CB 2D 08 00 01 20 00 00 00 00	 MOD

Window commands		UserLevel
09 CB 5A	Freezes the screen (creates a picture of the actual screen and blocks user operations on web GUI and via remote). You are still able to open new windows via WolfProt commands as Cynap did not stop listening for events such as AirPlay Mirroring. These windows will shop up once the user presses the unfreeze button or when you send the freeze off command. Syntax: 09 CB 5A 01 D0 Variables: [D0] Action: **00** Freeze off, **01** freeze on, **02** toggle freeze Returns: 09 CB 2D 00 Parameters: None Example: Freeze! 09 CB 5A 01 01	 MOD

File operation commands		UserLevel
0D CB 6A	Copies a defined source file on a connected USB memory stick. Available files to copy are recordings (h.264/.mp4) and snapshots (.jpg) residing on volume SYSTEM. If a Syntax: 0D CB 6A AB CD D0…Dn Variables: [AB][CD] size of file name (+2 for AB CD bytes) [D0] …[Dn] mount, file path and file name Returns: 09 CB 6A 00 Parameters: None Example: Copy an existing 0D CB 6A 00 31 system:///Cynap-Consulting_20170113_102639.jpg	 MOD
0D CB 3C	Opens a file from a defined source (local HDD or USB). Syntax: 0D CB 3C AB CD D0…Dn Variables: [AB][CD] size of file name (+2 for AB CD bytes) [D0] …[Dn] mount, file path and file name Returns: 09 CB 3C 00 Parameters: None Example: Copy an existing 0D CB 3C 00 31 system:///Cynap-Consulting_20170113_102639.jpg	 MOD

<table>
<tr>
<td>0D CB 7C</td>
<td>Opens a file from a defined source (local HDD or USB).

Syntax:
0D CB 7C AB CD D0…Dn

Variables:
[AB][CD] size of file name (+2 for AB CD bytes)
[D0] …[Dn] mount, file path and file name

Returns:
09 CB 7C 00

Parameters:
None

Example:
Copy an existing
0D CB 7C 00 31 system:///Cynap-Consulting_20170113_102639.jpg</td>
<td> 
</td>
</tr>
<tr>
<td>0D CB AB</td>
<td>Set up networking drives
The URL field contains the name of the server – if the server has not been allocated on a name server you could also use its IP address.

Syntax:
0D CB AB AB CD D0 D1 D2 D3 Dn D4 D5 Dn D6 D7 Dn D8 D9 Dn DA DB Dn DC DD

Variables:
<table>
<tr><td>[AB][CD] total size of all variables
(+2 to include AB CD bytes)</td><td>[D0] index (1 – 9)</td></tr>
<tr><td colspan="2">[D1] type of networking drive, 01 CIFS, 02 NFS</td></tr>
<tr><td>[D2] size of name</td><td>[D3] …[Dn] label/name (255 chars max)</td></tr>
<tr><td>[D4] size of URL</td><td>[D5] …[Dn] URL (255 chars max)</td></tr>
<tr><td>[D6] size of username</td><td>[D7] …[Dn] username (255 chars max)</td></tr>
<tr><td>[D8] size of password</td><td>[D9] …[Dn] password (255 chars max)</td></tr>
<tr><td>[DA] size of domain</td><td>[DB] …[Dn] domain (255 chars max)</td></tr>
<tr><td>[DC] Access mode, 00 Read only 01 Read-write</td><td>[DD] Default upload 00 no, 01 yes</td></tr>
</table>
Returns:
09 CB AB 00

Parameters:
None

Example:
Set up a CIFS drive and set it as default upload drive (read only-drive can´t be set as default upload drive).

0D CB AB 00 00 01 07 mydrive 13 //server01/resource 04 user
08 password 06 domain 01 01</td>
<td> 
</td>
</tr>
</table>

08 CB 7E	Check free disk storage – get the amount of available space on your Cynap Syntax: 08 CB 7E 01 D0 Variables: [D0] unit 00 MB Returns: 08 CB 7E D0 D1 D2 D3 D4 Parameters: [D0] unit status, [D1][D2][D3][D4] free space left Example: >> 08 CB 7E 01 00 << 08 CB 7E 00 00 00 00 DC	 MOD
08 CB 3D	Available mounted file systems – returned bytes are in structured in JSON array. JSON structured array: example of mounted file systems [{"id":"Internal","name":"Internal","type":"local","status":"mounted","perms":"ro"}, {"id":"System","name":"System","type":"system","status":"mounted","perms":"ro"}, {"id":"USB","name":"USB","type":"usb","status":"notMounted"}, {"id":"dropbox","name":"Dropbox","type":"cloud","status":"notMounted"}, {"id":"gdrive","name":"GoogleDrive","type":"cloud","status":"notMounted"}, {"id":"box","name":"Box","type":"cloud","status":"notMounted"}, {"id":"netdrive0","name":"training","type":"netdrive","status":"mounted","perms":"rw"}, {"id":"netdrive1","name":"QNAP","type":"netdrive","status":"mounted","perms":"ro"}, {"id":"netdrive2","name":"Network Drive 2","type":"netdrive","status":"disabled"}, {"id":"netdrive3","name":"Network Drive 3","type":"netdrive","status":"disabled"}, {"id":"netdrive4","name":"Network Drive 4","type":"netdrive","status":"disabled"}, {"id":"netdrive5","name":"Network Drive 5","type":"netdrive","status":"disabled"}, {"id":"netdrive6","name":"Network Drive 6","type":"netdrive","status":"disabled"}, {"id":"netdrive7","name":"Network Drive 7","type":"netdrive","status":"disabled"}, {"id":"netdrive8","name":"Network Drive 8","type":"netdrive","status":"disabled"}, {"id":"netdrive9","name":"Network Drive 9","type":"netdrive","status":"disabled"}, {"id":"FTP","name":"FTP","type":"ftp","status":"mounted","perms":"wo"}] Syntax: 08 CB 3D 00 Variables: None Returns: 0A 01 CB 3D AB CD D0 …Dn Parameters: [AB][CD] size of JSON array [D0] …[Dn] JSON array Example: 08 CB 3D 00	 MOD

0C CB 3E	Fetch file list from selected directory (root path) Example file list: [{ "name": "Neuer Ordner", "type": "dir" }, { "name": "System Volume Information", "type": "dir" }, { "name": "wolfvision", "type": "dir" }, { "name": "01 Documents", "type": "dir" }, { "name": "02 Pictures", "type": "dir" }, { "name": "03 Video", "type": "dir" }, { "name": "04 Software", "type": "dir" }, { "name": "05 Audio", "type": "dir" }, { "name": "06 Certification Content", "type": "dir" }, { "name": "background.jpg", "type": "image" }, { "name": "Cynap-after-reset.txt", "type": "text" }, { "name": "test-video.mp4", "type": "video" }, { "name": "Cynap-01068795.log", "type": "unknown" }] Syntax: 0C CB 3E AB CD D0 …Dn Variables: [AB][CD] size of directory [D0] …[Dn] root path (e.g. usb:///folder/) Returns: 0A 01 CB 3E AB CD Parameters: JSON array - Name: file name - Type: dir, video, audio, image, text, office, unknown - Preload: download progress in % - (drive)uploadStatus: aborted, done, failed, running, pending - (drive)uploadProgress: progress in % - (drive) gdrive, dropbox, netdrive0-9, internal, usb, ftp Example: >> 0C CB 3E 00 07 USB:/// << 0A 01 CB 3E 00 00 01 00 FF content of JSON array	 MOD
08 CB 7B	Fetch list of current downloads Syntax: 08 CB 7B AB D0 Variables: [AB] size of selected cloud service indicator, 00 all (merged), 01 selected cloud service [D0] 00 DropBox,01 gDrive, 02 Box Returns: 0C CB 7B AB CD D0 …Dn AB CD size of JSON array (+2) D0 …Dn JSON array Array: - Filename: File name - Preload: progress of download in % - Status: download status: aborted, done, failed, running, pending Example: Get complete list of downloads >> 08 CB 7B 00 << 0C CB 7B 00 55 content of JSON array	 MOD

08 CB C1	Query list of current file uploads Uploads of current presentation is available on AccessLevel User and all uploads (e.g. still active uploads of former recordings) requires Admin user rights. Syntax: 08 CB C1 01 D0 Variables: [D0] 00 uploads of current presentation, 01 all uploads (admin only) Returns: 0C CB C1 AB CD D0 …Dn AB CD size of JSON array (+2) D0 …Dn JSON array Array: - Filename: File name - progress: progress of download in % - Status: download status: aborted, done, failed, running, pending - Target: target path Example: Show uploads of current presentation >> 08 CB C1 01 00 << 0C CB C1 00 55 content of JSON array	 

General commands such as file lists are combined within the general file operation commands – listed below are the cloud specific commands.

Cloud commands		UserLevel
09 CB 8F	Enable/disable one of several cloud services. Syntax: 09 CB 8F 02 D0 D1 Variables: [D0] cloud service: 00 DropBox, 01 gDrive, 02 Box, 03 坚果云/jian guo yun [D1] enable, disabled: 00 disable, 01 enable Returns: 09 CB 8F 00 Parameters: None Example: 09 CB 8F 02 01 01	
08 CB 8F	Get information which cloud services are enabled (all at once). Syntax: 08 CB 8F 00 Variables: None Returns: 08 CB 8F 04 D0 D1 D2 D3 Parameters: [D0] gDrive: 00 disabled, 01 enabled [D1] Dropbox: 00 disabled, 01 enabled [D2] Box: 00 disabled, 01 enabled [D3] 坚果云/jian guo yun: 00 disabled, 01 enabled Example: >> 08 CB 8F 00 << 08 CB 8F 04 01 01 01 00	

<table>
<tr><th>Cloud commands</th><th></th><th>UserLevel</th></tr>
<tr><td>09 CB 45</td><td>Start or stop a cloud service connection.
To see, if and which cloud services are connected use command 08 CB 4C (cloud status)

Syntax:
09 CB 45 02 D0 D1

Variables:
[D0] cloud service: 00 DropBox, 01 gDrive, 02 Box, 03 坚果云/jian guo yun
[D1] connect/disconnect: 00 disconnect, 01 connect

Returns:
09 CB 45 00

Parameters:
None

Example:
Connect to Google´s gDrive
09 CB 45 02 01 01</td><td>MOD</td></tr>
<tr><td>09 CB 46</td><td>Preload a file from the cloud

Syntax:
09 CB 45 AB CD D0 D1 …Dn

Variables:
[AB][CD] size of variables
[D0] action: 00 abort, 01 download, 02 download and open
[D1] …[Dn] path + file name (512 CHARS max)

Returns:
09 CB 46 00

Parameters:
None

Example:
09 CB 46 00 12 02 gdrive:///file.doc</td><td>MOD</td></tr>
<tr><td>08 CB 4C</td><td>Get live status information on configured cloud services

Syntax:
08 CB 4C 00

Variables:
None

Returns:
08 CB 4C 04 D0 D1 D2 D3

<table><tr><td>Parameters:
[D0] gDrive: status
[D1] Dropbox: status
[D2] Box: status
[D3] 坚果云/jian guo yun: status</td><td>Status:
00 disconnected
01 authenticating
02 connected
03 failed
04 synced</td></tr></table>
Example:
>> 08 CB 4C 00
<< 08 CB 4CF 04 02 02 03 00</td><td>MOD</td></tr>
</table>

Cloud commands		UserLevel
0D CB BF	File upload to cloud services To upload a file you need to specify the source file location and the target path – there´s also the possibility to cancell or restart a former initiated upload via index. Based on your chosen action (indexed or non-indexed/) the variables change slightly. Syntax: Non-indexed: 0D CB BF AB CD D0 D1 D2 D3 …Dn D4 D5 D6…Dn Indexed (if action == 02 or 03) 0D CB BF AB CD D0 D1 D2 D3 D4 Variables: Non-indexed: [AB][CD] size of variables (+2 for AB CD bytes) [D0] Action: 00 cancel, 01 start, 02 cancel index, 03 restart index [D1][D2] size of source path and file name [D3] …[Dn] source (path + file name) [D4][D5] size of target path [D6] …[Dn] target (path only) Indexed: [AB][CD] size of variables (+2 for AB CD bytes) [D0] Action: 00 cancel, 01 start, 02 cancel index, 03 restart index [D1][D2] size of index [D3][D4] index Returns: 09 CB 7C 00 Parameters: None Example: Copy an existing 0D CB BF 00 3A 01 00 23 internal:///snapshot/snapshot88.jpg 00 0E gdrive:///pics	 MOD
0D CB 87	Remove file item from the list of pending downloads Syntax: 0D CB 87 AB CD D0…Dn Variables: [AB][CD] size of file name (+2 for AB CD bytes) [D0] …[Dn] mount, file path and file name (512 CHARS max) Returns: 09 CB 87 00 Parameters: None Example: 0D CB 87 00 31 gdrive:///Cynap-Consulting_20170113_102639.jpg	 MOD

Cloud commands		UserLevel
0D CB CC	Log in to cloud service Box Set user name and password. Syntax: 0D CB CC AB CD D0 D1 …Dn D2 D3 …Dn Variables: [AB][CD] size of file name (+2 for AB CD bytes) [D0] size of user name [D1] …[Dn] user name (63 CHARS max) [D2] size of password [D3] …[Dn] password Returns: 09 CB CC 00 Parameters: None Example: Copy an existing 0D CB CC 00 0E 04 user 08 password	 MOD
0D CB EF	Log in to cloud service坚果云/jian guo yun Set user name and password. Syntax: 0D CB EF AB CD D0 D1 …Dn D2 D3 …Dn Variables: [AB][CD] size of file name (+2 for AB CD bytes) [D0] size of user name [D1] …[Dn] user name (63 CHARS max) [D2] size of password [D3] …[Dn] password Returns: 09 CB EF 00 Parameters: None Example: Copy an existing 0D CB EF 00 0E 04 user 08 password	 MOD

General commands such as file lists and transfer are combined within the general file and storage operation commands – listed below are the FTP specific commands.
Note: FTP is solely used for uploads and only a single FTP connection can be stored.

FTP commands		UserLevel
09 CB 62	Enable FTP service Syntax: 09 CB 62 01 D0 Variables: [D0] enable/disable: 00 disable, 01 enable Returns: 09 CB 62 00 Parameters: None Example: 09 CB 62 01 01	 ADM

FTP commands		UserLevel
08 CB 62	Verify if FTP service has been enabled Syntax: 08 CB 62 00 Variables: None Returns: 08 CB 62 01 D0 Parameters: [D0] enabled/disabled: 00 disabled, 01 enabled Example: >> 08 CB 62 00 << 08 CB 62 01 01	ADM
09 CB 63	Set or change a FTP-server location. The last updated FTP server will also be updated on the global settings file. Syntax: 09 CB 63 AB D0 …Dn Variables: [AB] size of ftp server address (+1 for AB bytes) [D0] …[Dn] FTP address (255 CHARS max) Returns: 09 CB 63 00 Parameters: None Example: 0D CB 63 13 ftp.our-server.com	ADM
08 CB 63	Get the stored FTP-server location. Syntax: 08 CB 63 00 Variables: None Returns: 09 CB 63 00 Parameters: [AB] size of ftp server address (+1 for AB bytes) [D0] …[Dn] FTP address Example: >> 08 CB 63 00 << 08 CB 63 13 ftp.our-server.com	ADM

FTP commands		UserLevel
09 CB 64	Set or change a FTP-server user name. The last updated FTP user got updated on the global settings file. Syntax: 09 CB 64 AB D0 …Dn Variables: [AB] size of ftp server user (+1 for AB bytes) [D0] …[Dn] FTP user (63 CHARS max) Returns: 09 CB 64 00 Parameters: None Example: 09 CB 63 05 user	ADM
08 CB 64	Get the stored FTP- user name. The last updated FTP user got updated on the global settings file. Syntax: 08 CB 64 00 Variables: None Returns: 08 CB 64 AB D0 …Dn Parameters: [AB] size of ftp server user (+1 for AB bytes) [D0] …[Dn] FTP user (63 CHARS max) Example: >> 08 CB 64 00 Receive 08 CB 64 05 user	ADM
09 CB 65	Set or change a FTP-server password. The last updated FTP password got updated on the global settings file. Syntax: 09 CB 64 AB D0 …Dn Variables: [AB] size of ftp password (+1 for AB bytes) [D0] …[Dn] FTP password (63 CHARS max) Returns: 09 CB 65 00 Parameters: None Example: 09 CB 65 09 password	ADM

FTP commands		UserLevel
08 CB 65	Get the stored FTP- password. The last updated FTP password got updated on the global settings file. Syntax: 08 CB 65 00 Variables: None Returns: 08 CB 65 AB D0 …Dn Parameters: [AB] size of ftp server user (+1 for AB bytes) [D0] …[Dn] FTP password (63 CHARS max) Example: >> 08 CB 64 00 Receive 08 CB 64 09 password	ADM
09 CB 69	Test the FTP connection to see, if combination of user, password and server name are correct. A file of 0 byte size will be uploaded. To see if the connection try was successful you need to send 08 CB 67 (query on set command, see below). Syntax: 09 CB 69 00 Variables: [None Returns: 09 CB 69 00 Parameters: None Example: 09 CB 69 00	ADM
09 CB 67	Get the result of the FTP connection test. Syntax: 09 CB 67 00 Variables: None Returns: 08 CB 67 01 D0 Parameters: [D0] Status: **00** test pending, **01** connection failed, **02** test successful Example: 08 CB 67 00 08 CB 67 01 02	ADM

Collaboration commands (whiteboard/annotation)		UserLevel
09 CB E6	Collaboration User Show or hide user connection status Syntax: 09 CB E6 01 D0 Variables: [D0] user connection: 00 hidden, 01 visible Returns: 09 CB E6 00 Parameters: None Example: 09 CB E6 01 01	
08 CB E6	Query Collaboration User Get information on user connection status Syntax: 08 CB E6 00 Variables: None Returns: 08 CB E6 01 D0 Parameters: [D0] user connection: 00 hidden, 01 visible Example: 09 CB E6 01 01	
09 CB 8C	Paint on whiteboard or selected window Syntax: 09 CB 8C 06 D0 D1 D2 D3 D4 D5 Variables: [D0] Paint: 01 stopped, 02 in progress [D1] layer: -1 separate window (white board) or 0 – 3 WindowID [D2][D3] x position [D4][D5] y position Returns: 09 CB 8C 00 Parameters: None Example: Copy an existing 09 CB 8C 06 00 -1 00 00 00 00	

Collaboration commands (whiteboard/annotation)		UserLevel
09 CB 96	Annotation/whiteboard action Depending of the submitted size and D2 the command changes its structure. Syntax: 09 CB 96 **02** D0 D1 or 09 CB 96 **0A** D0 D1 D2 D3 or 09 CB 96 **AB** D0 D1 D2 …Dn Variables: Size of variables: 02 [D0] WindowID, 0, 1, 2 or 3 [D1] Action: 00 redo / 01 undo / 02 clear layer / 03 clear all 04 enable annotation / 05 stop / 06 snapshot / 07 connect using default nickname 08 disconnect me / 09 disconnect all / 0B pause Size of variables: 0A [D0] WindowID, 0, 1, 2 or 3 [D1] Action: 02 clear user / 08 disconnect user [D2][D3]D4[D5]D6[D7]D8[D9]: client id (D2 higher, D9 lower byte) Size of variables: dynamic [AB] size of variables [D0] WindowID, 0, 1, 2 or 3 [D1] Action: 07 connect using nickname / 0A set nickname [D2] …[Dn] nickname (32 chars max) Returns: 09 CB 96 00 Parameters: None Example: 09 CB 96 02 00 06	MOD
09 CB 95	Set drawing tool Changes/sets drawing attributes (size, colour, alpha level and 3 shapes) – Cynap colours are not fixed and can be set using the standard RRGGBB nomination. Syntax: 09 CB 95 07 D0 D1 D2 D3 D4 D5 D6 Variables: [D0] WindowID, -1 annotation/whiteboard layer, 0 – 3 WindowID [D1] Tool select: 00 pen, 01 line, 02 circle, 03 rectangle, 04 eraser [D2] red [D3] green [D4] blue levels e.g. FF FF FF equals white [D5] alpha level [D6] size of drawing vector Returns: 09 CB 95 00 Parameters: None Example: Set drawing color to orange and the shape to a circle 09 CB 95 07 -1 02 FF 7D 00 FF 05	MOD

Collaboration commands (whiteboard/annotation)		UserLevel
08 CB 95	Get drawing settings Get information on status, numbers of user etc. Syntax: 08 CB 95 01 D0 Variables: [D0] WindowID, -1 annotation/whiteboard layer, 0 – 3 WindowID Returns: 08 CB 95 05 D0 D1 D2 D3 D4 Parameters: [D0] Status: -2: no annotation or whiteboard window -1: client_id unknown 0: client_id connected [D1] number of redo operations [D2] number of undo operations [D3] clear all: 00 impossible, 01 possible [D4] number of users Example: >> 08 CB 95 01 -1 << 08 CB 95 05 00 00 0A 05	 MOD
09 CB BD	Set annotation/whiteboard user visibility If sent with variable length of 02 then all users are set to hidden/visible and when using a size of 0A then you´re able to hide a specific user Syntax 09 CB BD 02 D0 D1 09 CB BD 0A D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 Variables: [D0] WindowID [D1] 00 hidden, 01 visible [D2] [D3] [D4] [D5] [D6] [D7] [D8] [D9] ClientID, D2 higher byte, D9 lower byte) Returns: 09 CB BD 00 Parameters: None Example: Set a user session visible 09 CB BD 02 -1 01 00 00 A0 C3 8F FF 1D 05	 MOD
08 CB BD	Get list of annotation users Returns a JSON array listing all visible users and theirs nicknames Syntax 08 CB BD 02 D0 D1 Variables: [D0] WindowID [D1] type of user: 00 calling user, 01 all users Returns: 09 CB BD AB CD D0 …Dn Parameters: [AB][CD] size of parameters [D0] …[Dn] JSON array Array: client_id, nickname, visible yes|no: 00 no, 01 yes Example: >> 08 CB BD 02 -1 01 << 08 CB BD 00 FF JSON array	 MOD

Collaboration commands (whiteboard/annotation)		UserLevel
08 CB 97	Annotation function active or stopped Sends back the information if annotation is currently running, stopped or paused. Syntax: 08 CB 97 00 Variables: None Returns: 09 CB 7C 01 D0 Parameters: [D0] Status: 01 not running, 01 started, 02 paused Example: >> 08 CB 97 00 << 08 CB 97 01 01	MOD

Mirroring service commands		UserLevel
09 CB 36	Enable/disable Miracast service Syntax: 09 CB 36 01 D0 Variables: [D0] 00 disable Miracast, 01 enable Miracast Returns: 09 CB 36 00 Parameters: None Example: 09 CB 36 01 01	ADM
08 CB 36	Check if Miracast service has been enabled or disabled Syntax: 08 CB 36 00 Variables: None Returns: 08 CB 36 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: 08 CB 36 00 08 CB 36 01 01	ADM

Mirroring service commands		UserLevel
09 CB 37	Enable/disable AirPlay service Syntax: 09 CB 37 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB 37 00 Parameters: None Example: 09 CB 37 01 01	
08 CB 37	Check if AirPlay service has been enabled or disabled Syntax: 08 CB 37 00 Variables: None Returns: 08 CB 37 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: 08 CB 37 00 08 CB 37 01 01	
09 CB D6	Enable/disable AirPlay Meeting Room PIN Syntax: 09 CB 37 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB 37 00 Parameters: None Example: 09 CB D6 01 01	
08 CB D6	Check if AirPlay Meeting Room PIN has been enabled or disabled Syntax: 08 CB 37 00 Variables: None Returns: 08 CB 37 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: >> 08 CB D6 00 << 08 CB D6 01 01	

Mirroring service commands		UserLevel
09 CB B2	Enable/disable Chromecast service Syntax: 09 CB B2 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB B2 00 Parameters: None Example: 09 CB B2 01 01	ADM
08 CB B2	Check if Chromecast service has been enabled or disabled Syntax: 08 CB B2 00 Variables: None Returns: 08 CB B2 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: >> 08 CB B2 00 << 08 CB B2 01 01	ADM
09 CB 86	Enable/disable vSolution Connect service Syntax: 09 CB 86 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB 86 00 Parameters: None Example: 09 CB 86 01 01	ADM
08 CB 86	Check if vSolution Connect service has been enabled or disabled Syntax: 08 CB 86 00 Variables: None Returns: 08 CB 86 01 D0 Parameters: [D0] 00 disabled, 01 enabled Example: >> 08 CB 86 00 << 08 CB 86 01 01	ADM

Mirroring service commands		UserLevel
09 CB 3B	Enable/disable mirroring services Syntax: 09 CB 86 01 D0 Variables: [D0] 00 disable, 01 enable Returns: 09 CB 86 00 Parameters: None Example: 09 CB 86 01 01	ADM
08 CB 3B	Check if mirroring services have been enabled or disabled Syntax: 08 CB 3B 00 Variables: None Returns: 08 CB 3B 01 D0 Parameters: [D0] 00 disabled, 01 enabled [D1][D2] time left Example: >> 08 CB 3B 00 << 08 CB 3B 01 01	ADM

9 Limitations

Cynap's streaming services can cause increased network traffic which might interfere with the Crestron or AMX processor.

It is highly recommended to follow your Room Management System brand's network guide lines in any setup.

Protocol changes happen and new commands will be added with every new release of an updated firmware. We therefore advise to check for obsolete and changed commands in your implementation before you apply a new firmware release.

The data sent to Cynap via WolfProt won´t be neither parsed nor validated. Please mind this fact before changing vital settings (e.g. network configuration) when using Administrator Access Level commands.

10 Troubleshooting

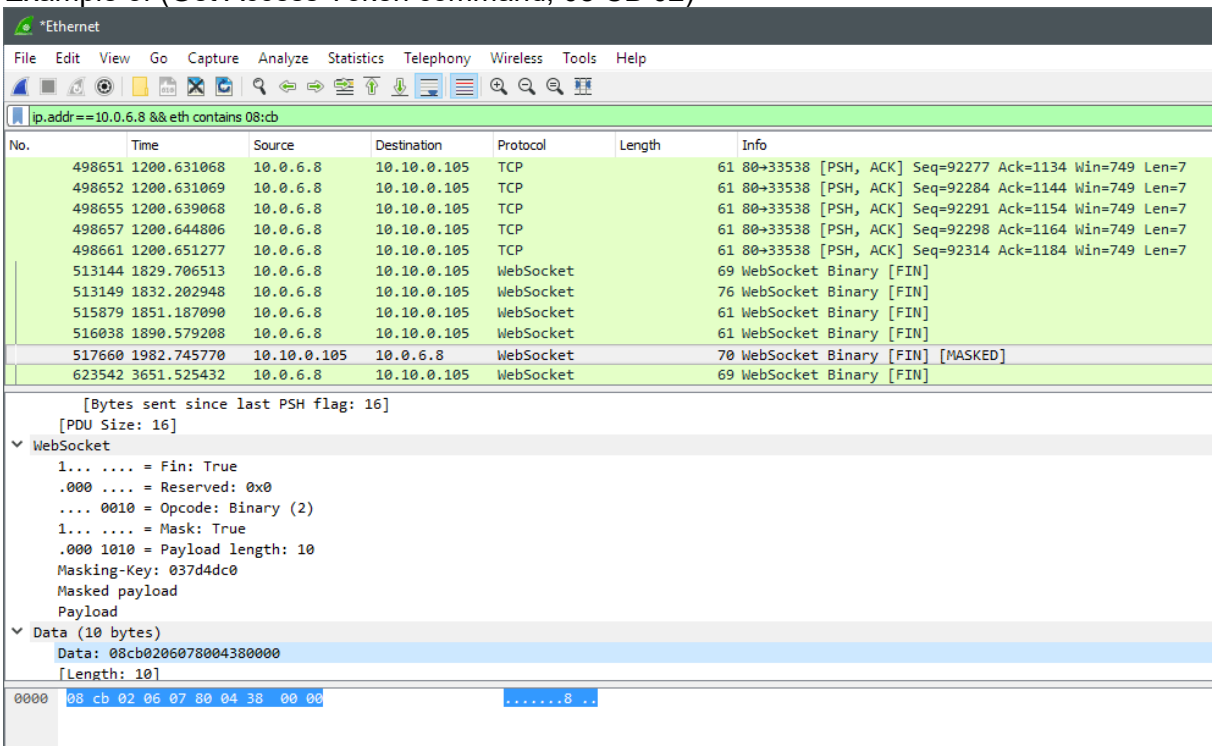
Command issues
Check that the MAC and IP address as well as all TCP/UDP ports are configured correctly.

Start Wireshark
Using Wireshark to verify your get and set commands

Start collecting the traffic on the interface used for Cynap communication (e.g. Ethernet) – Important: connect to Cynap after the start of traffic collection, otherwise Wireshark won´t be able to initiate properly and won´t catch traffic on the WebSocket protocol.

Filter the IP address of your Cynap and your command in hex.
For instance: ip.addr == 192.168.10.10 && eth contains 08:cb

Example of (Get Access Token command, 08 CB 02)



Authorization issues
Check that the Room Management System user has been set up with the same corresponding password (on both ends).

Make sure that the command you´re sending (e.g SET command) is being covered by the necessary Access Level of your login.

Network issues
Commandline: ping the gateway of your network and issue an arp –a to get a list of IP and MAC address combinations to check if your Cynap MAC address is found.

Device issues
Check that the Cynap module is executed on the processor.

Check that the layout file has been transferred to your touch terminal.

11 Glossary

Term	Description
AirPlay	AirPlay is a proprietary protocol stack/suite developed by Apple Inc. that allows wireless streaming between devices of audio, video, device screens, and photos, together with related metadata
AMX	AMX LLC, a manufacturer of commercial and residential control systems
Google Cast	GoogleCast enables users with a mobile device or personal computer to initiate and control playback of Internet-streamed audio/visual content through mobile and web apps that support the Google Cast technology. Alternatively, content can be mirrored from the Google Chrome web browser running on a personal computer, as well as from the screen of some Android devices.
Crestron	Manufacturer of home automation systems, building and campus control systems.
WolfProt	High Definition Media Interface, standardized interface to digitally transmit picture and sound
Miracast	Miracast is a standard for wireless connections from devices to display
BYOD	Bring Your Own Device: mobile devices
WOL	Wake-On-LAN: broadcast message to specific MAC address

12 Figures

13 Changes

Ver 1.0	Initial release	30.8.2016	rg
Ver 1.1	Separating into 2 parts (RMS template integration and coding with WolfProt) Adding latest WolfProt development	16.9.2016	rg
Ver 1.2	Separate two parts into two separate documents – changing structure for template integrators or WolfProt developers	12.11.2016	rg
Ver 1.3	Added APIs by AuthorizationLevel	16.11.2016	rg
Ver 1.4	Added fully fleshed out tutorials	01.12.2016	rg
Ver 1.5	WolfProt commands based on input from AMX dev	12.12.2016	Rg
Ver 1.6	Userlevels added, every cmd verified	30.01.2017	Rg