

User's Manual

DIMAX804KC ***Presentation Matrix*** ***The All-in-One Device***



DEXON® Systems

Document: ***User's Manual
DEXON DIMAX804KC
Presentation Matrix***
Document identifier: **228047**
Version: **3.2.1**

Copyrights (C) **DEXON® Systems**
All rights reserved.

DEXON Systems Ltd.

Baross u. 165. H-2040 Budaörs, Hungary

Tel: +36 23 422 804

Fax: +36 23 445 199

sales@dexonsystems.com

<http://www.dexonsystems.com>

You may not copy this documentation in whole or in part without permission from DEXON Systems Ltd. The authors reserve the rights to make any changes to this product and to revise the information about the products contained in this guide without an obligation to notify any persons about such revisions or changes.

DEXON® and DXWall® are registered trademarks of DEXON Systems

Intel and Pentium are trademarks of Intel Corporation.

Windows 7® and Windows 10® are registered trademarks of Microsoft Corporation

UNIX ® is a registered trademark of the Open Group in the United States and other countries.

VNC is Copyright RealVNC Ltd. 2002 and Copyright AT&T Laboratories Cambridge 1996-2001.

All other trademarks are properties of their respective owners.

CONTENTS

1	DEXON'S SMALL MATRICES	3
1.1	DEXON's Presentation Matrix	3
1.2	Main characteristics	3
2	INTERFACES AND OPERATION	4
2.1	General Overview	4
2.2	Front Panel.....	4
2.3	Back Panel and Interfaces	5
2.4	Typical Application	5
3	SETUP AND CONFIGURATION	6
3.1	Push Button and LCD panel.....	6
3.1.1	Advanced.....	6
3.1.2	Layouts	7
3.1.3	User Presets	8
3.2	Setup through HTML5 Web Browser	8
3.2.1	Advanced Panel.....	12
3.3	Custom Output Timing	14
3.4	Media Player Panel	16
3.4.1	Create Playlist Presentation.....	16
3.4.2	Create Scheduler Presentation.....	17
3.5	Layout Handling	18
3.5.1	The Calendar based DXScenario User Interface	18
4	EXTERNAL PROGRAMMING	20
4.1	Message Format	20
4.2	Communication Protocol.....	21
4.2.1	Administration	21
4.2.2	Input.....	24
4.2.3	Output.....	27
4.2.4	Advanced.....	36
4.2.5	Layout	38
4.2.6	Preset.....	41
4.2.7	Video Assignment.....	43
4.2.8	Audio Assignment.....	44
4.2.9	Miscellaneous	45
5	SOFTWARE AND FIRMWARE UPDATES	49
6	SPECIFICATIONS	50
6.1	Electrical, Mechanics and Environment	50
6.2	Input Signals	50
6.3	Output Signals.....	52
7	PROGRAMMING SAMPLES	53
7.1	Programming Samples for Matrix Switcher Subsystem	53

FIGURES

Figure 2-1. Front Panel	4
Figure 2-2. Interfaces and Connections	5
Figure 2-3. Application	5
Figure 3-1. LCD panel and push buttons	6
Figure 3-2. DXWebControl of DIMAX804 application overview	9
Figure 3-3. Input Settings of Input 1.....	9
Figure 3-4. Output Settings of Output 1.....	10
Figure 3-5. Output Settings of each display	11
Figure 3-6. DXWebControl DIMAX804KC Advanced Panel	12
Figure 3-7. Custom Output Timing Editor.....	14
Figure 3-8. Media Player Panel.....	16
Figure 3-9. Add/modify playlist presentation panel	17
Figure 3-10. Add/modify calendar based presentation panel	18
Figure 3-11. Calendar based DXScenario User Window	18
Figure 3-12. Add/modify calendar based events panel.....	19

1 DEXON'S SMALL MATRICES

1.1 DEXON's Presentation Matrix

DEXON Systems' DIMAX Matrix Switchers are vital components in demanding AV projects where different input signals must be switched to any or multiple outputs. They serve presentation, digital signage or board room applications that demand best equipment for successful projects. DEXON's Intelligent Matrix (DIMAX) provides perfect solution with smart features as quality scaling, seamless switching and picture in picture visualisation. DIMAX Family offers superior design, excellent price/performance balance, brilliant video quality and high reliability tailored to challenging AV projects. Easy installation and control are guaranteed by DEXON's intuitive man-machine interface.

DIMAX versions:

- **DIMAX804KC** – 8xHDMI inputs, 4xHDMI outputs, 4 x mirrored HDBaseT signals

1.2 Main characteristics

The DIMAX804KC presentation matrix is an all-in-one solution of a standalone box with the following features:

- Input signal standard:
 - 8 x digital HDMI with HDCP - progressive up to 4K30Hz
- Output signal standards:
 - 4 x digital HDMI with HDCP - progressive up to 4K30Hz
 - 4 x HDBaseT with HDCP - progressive up to 4K30Hz
- High quality scaling up or scaling down of input signals to the outputs
- Picture-in-picture display of second input to any or all output screens
- Programmable seamless switching and cross-fade transition effects
- Full HDCP support for input signals and output connections
- Audio de-embedding and embedding to HDMI output signals
- Video Wall mode for large area visualization
- Internal processor for advanced visualization functions when using 9th input
 - Internal Media Player, playing local or LAN based remote contents, supporting AVI and H.264 formats
 - Programmable automatic scheduling for media player
 - Web page rendering of pre-defined WEB page in live
 - IP camera display in real time, RTSP protocol, H.264
 - Remote operator screen's display through the LAN
- Programmable local or remote sleep and wake up functions - WoL
- Easy-to-use graphics user interface for configuration process and wall management from laptops over the LAN

2 INTERFACES AND OPERATION

2.1 General Overview

Input Signals Processed

- **DVI**
DVI 1.0
from 640x480 up to 1920x1200@60Hz
- **HDMI**
HDMI 2.0 and HDCP 2.2 specifications
from 640x480 up to 4096x2160@30Hz

Output Signals

- **HDMI**
HDMI 2.0 and HDCP 2.2 specifications
from 640x480 up to 4096x2160@30 Hz
- **HDBaseT outputs**
Cat5/6 cable up to 100m (330 feet)
from 640x480 up to 4096x2160@30Hz

Audio handling

- R, L analogue audio input
- R, L analogue audio output
- HDMI embedded audio output

2.2 Front Panel



Figure 2-1. Front Panel

2.3 Back Panel and Interfaces

DIMAX804KC Back Panel

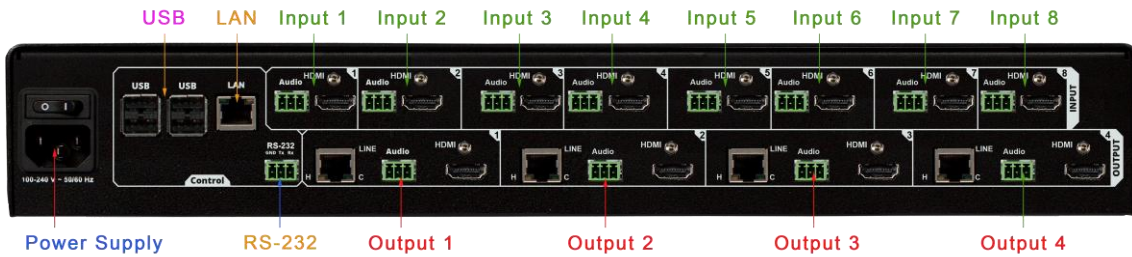


Figure 2-2. Interfaces and Connections

2.4 Typical Application

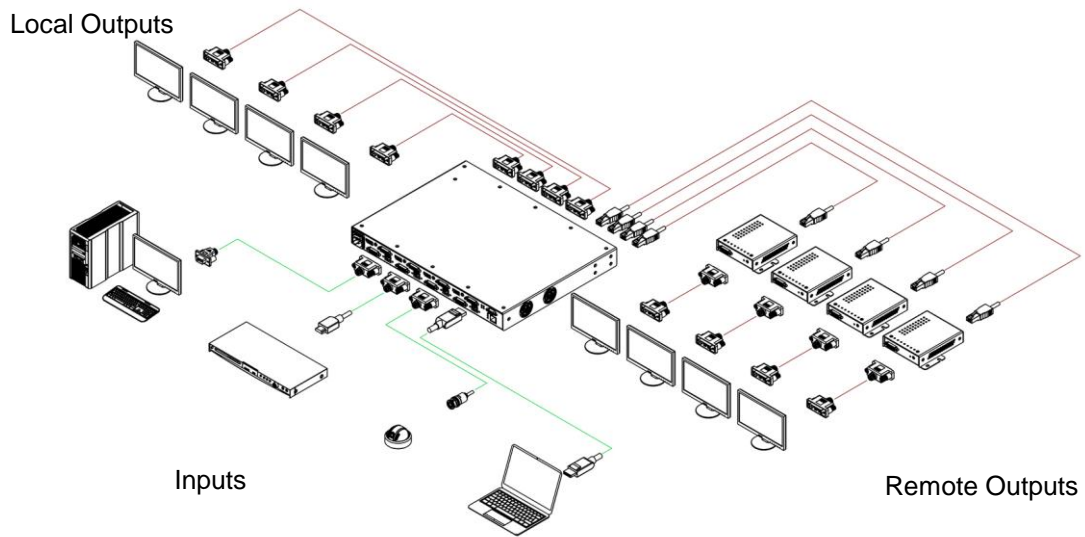


Figure 2-3. Application

3 SETUP AND CONFIGURATION

The DIMAX804KC device provides perfect solution for control and setup. The user has the following options to control:

- Push Button and LCD panel
- Easy-to-use Graphics User Interface via web browser
- RS-232 interface

3.1 Push Button and LCD panel

The DIMAX804KC devices can be configured by push buttons and LCD panel for important settings. The device has three main menu items that can be selected by ▼▲ buttons after pushing the MENU button:

- Advanced
- Layouts
- User Presets

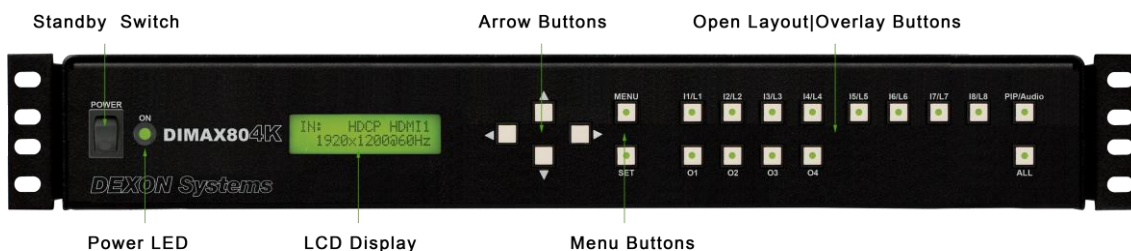


Figure 3-1. LCD panel and push buttons

3.1.1 Advanced

Button Functions

The user can set the function of the push buttons as Overlay Open or Recall Layout.

- Overlay – The buttons operate on individual overlay windows. An input source selected by an input button can be opened on output screen selected by an output button.
- Layout – The buttons operate on predefined layouts. Eight available layouts can be recalled by a simple button click.

Remote Control

This menu item contains the parameters of remote control:

- DHCP enable
- Specify IP address
- Set network mask

You should press the SET button to save the settings.

OSD

There is a built-in information text about the current output settings and input connections. This information text can be visualized on the **OSD** (On Screen Display) panel. The visibility settings of the OSD panel can be changed here.

- **Auto**: the information panel is automatically displayed in case of any changes in the settings.
- **Yes**: the information panel is visualized permanently.
- **No**: the information panel is not visible.

By default, the information panel can be seen on the top-left corner of each display output. The display time in case of Auto mode is 4 seconds. These parameters can be changed from the controlling application.

Serial Port

The adjustable parameters are: Baud Rate, Databits, Parity, Stopbits and Flow Control.

Factory Reset

The factory reset is a full restore of device to its factory settings.

3.1.2 Layouts

Layout is a set of parameters describing arrangement of the input signals on the DIMAX device. The layout determines:

- Input video and audio signals
- Outputs assigned to that inputs
- Input and output cropping values
- Picture-in-picture signals and positions

The **Layouts** menu contains the list of available layouts. There are the very same three items inside each layout submenu:

- Layouts
 - Layout #1
 - Store
 - Recall
 - Clear
 - Layout #2
 - Store
 - Recall
 - Clear
 - ...
 - Layout #32
 - Store

- Recall
- Clear

If you want to store the current layout for future usage, select the **Store** menu item. An '*' sign before the Store keyword will show that the given layout contains the arrangement. Any previously stored layout can be restored by the help of **Recall** menu item. Select the **Clear** menu item if a layout is not needed any more. An '*' sign before Clear keyword shows that the given layout is empty.

You can use the push buttons (L1-L8) to select certain layouts only if you are not navigating in the main menu and the Button Function is in Layout mode.

3.1.3 User Presets

User Presets mean set of configuration parameters that include all the input configurations, output configurations and advanced configurations. The structure of the **Presets** menu is very similar to the Layouts menu. The user can do the same operations on each preset:

- Store
- Recall
- Clear

You may save 4 sets of parameters.

3.2 Setup through HTML5 Web Browser

The matrix switcher can be configured and set up by the Web application program. Please connect this device to a network. LAN Communication format:

Category	Data
Physical Layer	Ethernet 10/100 Mbps interface (802.3)
Network Layer	TCP/IP, WebSocket
Port	6464 (Communication Request Port, TCP/IP) 6465 (Communication Event Port, TCP/IP) 6466 (Communication Request Port, WebSocket) 6467 (Communication Event Port, WebSocket) 80 (HTTP)
Initial IP Address	DHCP
Fix Username	Administrator
Default Password	dexon (adjustable via DXHTMLControl in advanced panel)

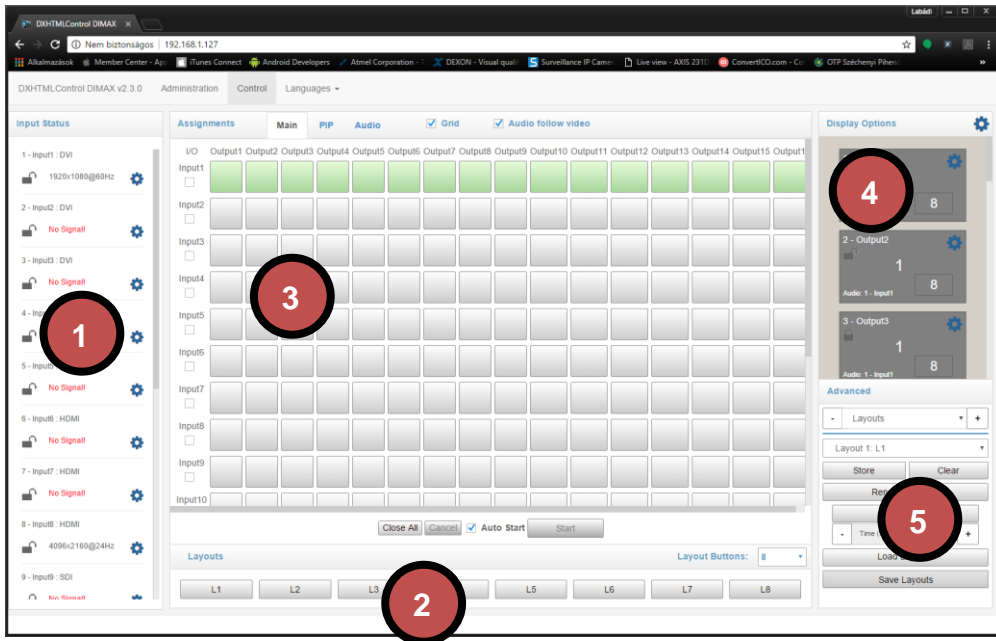


Figure 3-2. DXWebControl of DIMAX804 application overview

1. Input Status

This section shows the status of the input channels. You can see here the detected input signal timings (signal resolution and refresh rate) and the state of the HDCP encryption.

Click over the cog icon will open the input settings window, where user may set several parameters of the selected input:

- source's name
- source cropping
- audio handling
- MediaPlayer Scheduler
- Custom EDID, etc.

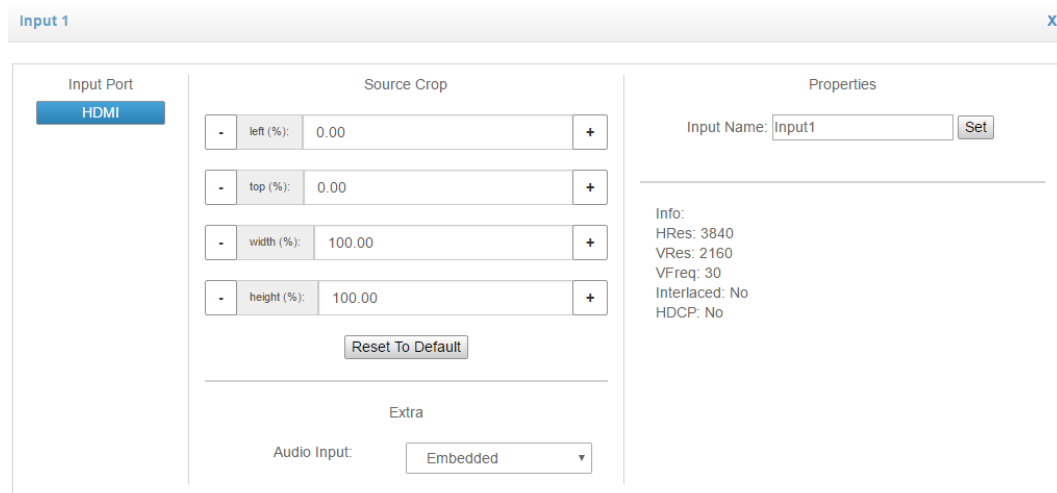


Figure 3-3. Input Settings of Input 1

Note: The usage of custom EDID: You can add a custom EDID Detailed Timing into the certain input or you can select an output which will be the source of the full EDID block. In case of DIMAX804KC the Cat output port has higher priority.

2. Layouts

The previously saved layout can be recalled by clicking on the selected button.

3. Assignments area

- **Main Channel Assignment tab**
This panel visualizes a matrix for intuitive assignment between inputs and outputs. The inputs are listed vertically the outputs are horizontally. Click on the desired rectangles to assign the inputs to the outputs. After completing the desired assignment push Start.
- **Close All**
The Close All button deletes all assignment.
- **Auto Start**
Set Auto Start ON and each user action takes effect immediately (otherwise pushing of Start button is required).
- **PIP Channel Assignment tab**
The PIP assignment can be modified on this tab that works like the main channel assignment. One PIP overlay is possible for each output display.
- **Audio Assignment tabs**
User may switch the audio channels separately from the main channel on this tab.

4. Display Options area

The output parameters are displayed and can be modified in this section. This shows the assignments of the output screens (main channel, PIP, audio).

User may modify the **output setting for each monitor separately** by clicking the cog icon inside the output monitor window.

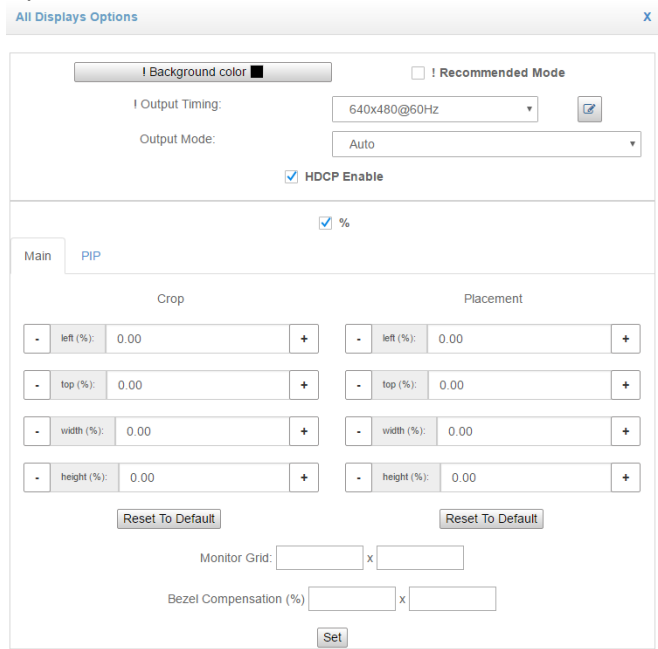


Figure 3-4. Output Settings of Output 1

The adjustable parameters are Output Name, Recommended Mode, Output Timing, HDCP, Test Pattern, Keep Aspect Ratio and the Background Colour. You can also modify here the PIP settings such as size and position of the PIP overlay, the Cropping of the output.

User can also **modify common setting of the output monitors** by clicking on the cog icon in the heading next to the Display Option text.

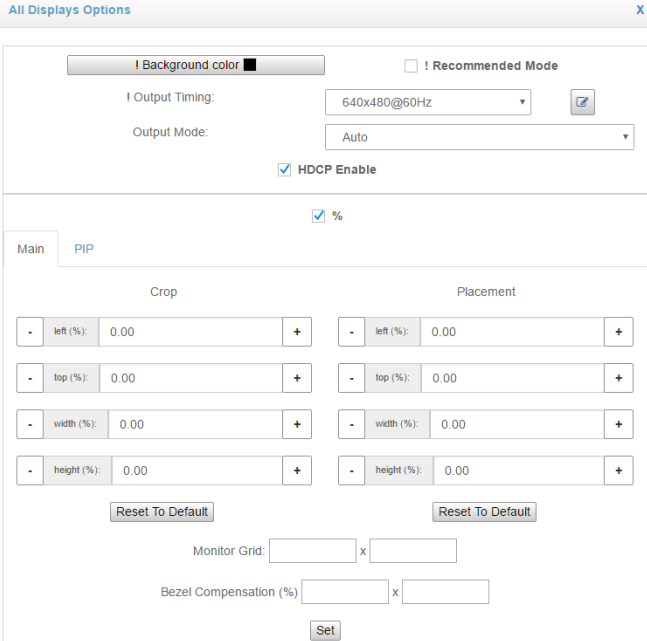


Figure 3-5. Output Settings of each display

The commonly adjustable parameters are Recommended Mode, Output Timing, HDCP, Test Pattern, Keep Aspect Ratio and the Background Colour. You can also modify here the PIP settings as size and position of the PIP overlay and the Cropping of the output.

User may configure video wall mode by choosing the dimension of the monitor grid and/or bezel compensation and clicking on the Set button.

5. Advanced panel

This section contains group of other special parameters. Please see next section.

3.2.1 Advanced Panel

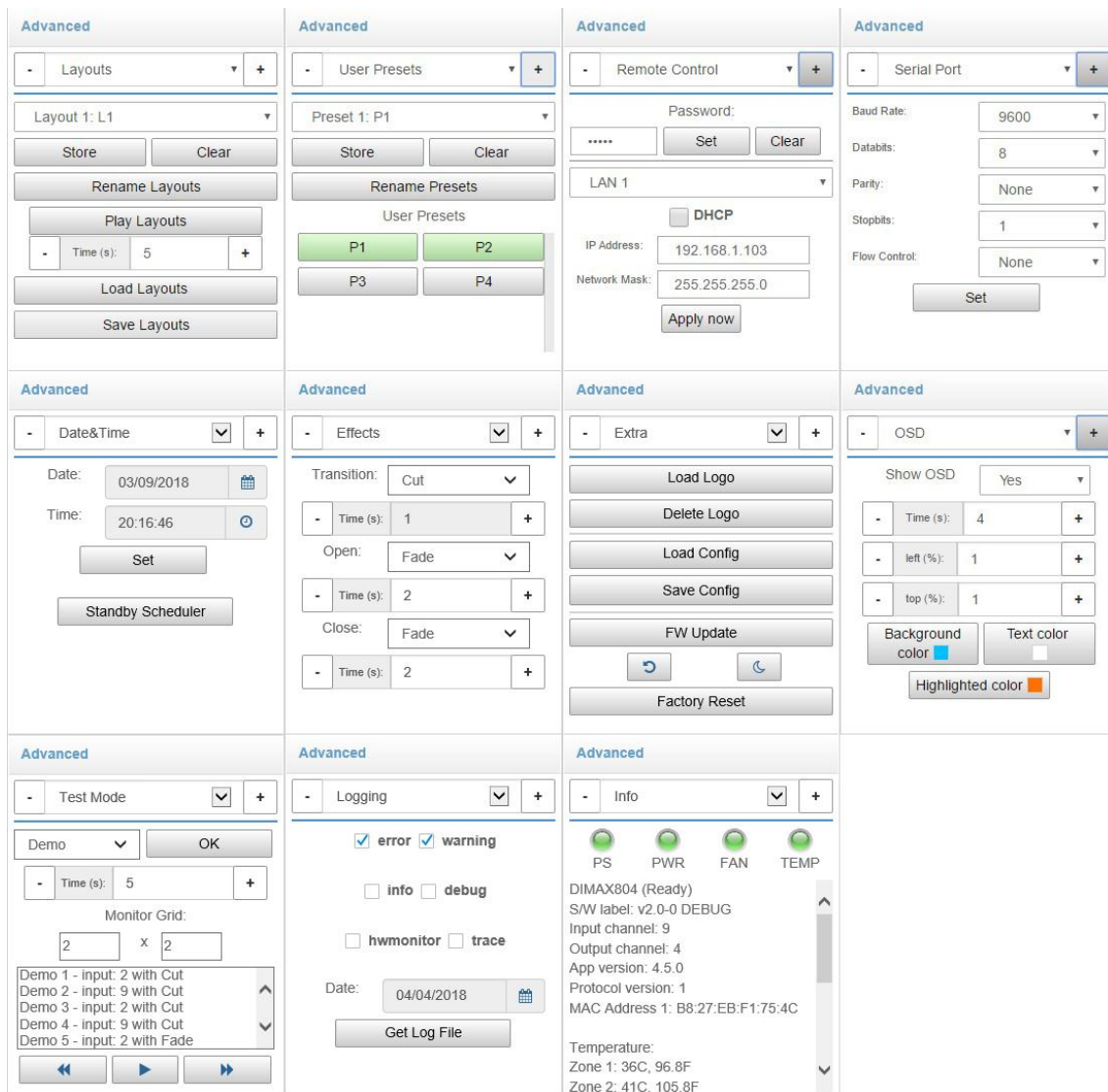


Figure 3-6. DXWebControl DIMAX804KC Advanced Panel

Layout Tab:

To save current output layout either as a shortcut button (Store button) or locally on the control PC (Save Layout button). Previously saved layouts can be loaded from the control PC with the Load Layout button. The shortcut button of a layout can be deleted by pressing the Clear button. Layout buttons can be renamed (Rename Layout).

User Presets Tab:

The User Presets can be named, saved and deleted on this tab.

Remote Control Tab:

The password can be changed in this section. The default password is "dexon".

The IP address uniquely identifies the module.

Serial Port Tab

The preferences of the serial port are adjustable on this tab. The adjustable parameters are: Baud Rate, Databits, Parity, Stopbits and Flow Control.

Date & Time Tab

You can set the current time and date. You can change the parameters of Wake Up and Standby function.

Effects Tab:

The desired transition, open and close effect can be chosen: Cut or Fade.

Extra Tab:

You can upload a small image for user interface logo (maximum 1 MB, *.png).

You can export the current configuration to a file through this web based application, and load the configuration from a local file (*.xml)

You can reset the product's settings to its factory default values.

You can activate the standby mode or start restart or shutdown method.

OSD Tab

The OSD panel visualization parameters can be set on this tab. The OSD panel can be seen on all the output displays. With the help of Show OSD parameter the visualization can be turned on or off permanently, or can be set to display automatically. In this case the OSD panel is visualized temporarily in every situation when the content of the panel changes. The display period can be set in seconds with the time parameter.

The position of the OSD panel can be specified with the coordinates of top-left corner. The left and top parameters are counted in percentage of the full dimensions of the output image. The background colour, the text colour and the highlighted colour can also be specified here.

Test Mode Tab

This page can be used to test or demonstrate with automatic settings of different parameters of the device. The sequence of different settings can be generated with a single mouse click over the OK button. To recall the generated sequence of settings can be automatically scheduled by pressing the Play button. The operator can walk through the list settings manually also.

Logging Tab

You can switch on or out the logging procedures. The device has 6 levels of logging. Log files can be recalled for each day by choosing the date and pressing Get Log File button.

Info Tab:

This tab displays the current input timing and provides special information about the product (App version, temperature, etc.). If the tab is visible, it is updated in every 5 seconds.

3.3 Custom Output Timing

The device contains several computer resolutions used by standard display systems today. However, some applications (LED walls) may require a native format that isn't enumerated on device's output timing list. The user can expand the output timing list with custom specified timing parameters.

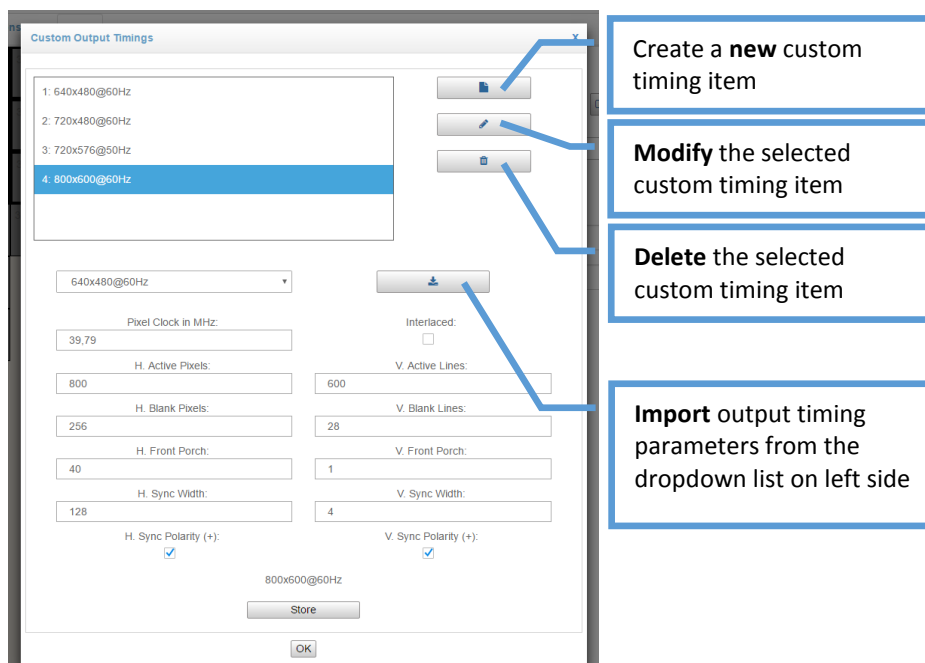



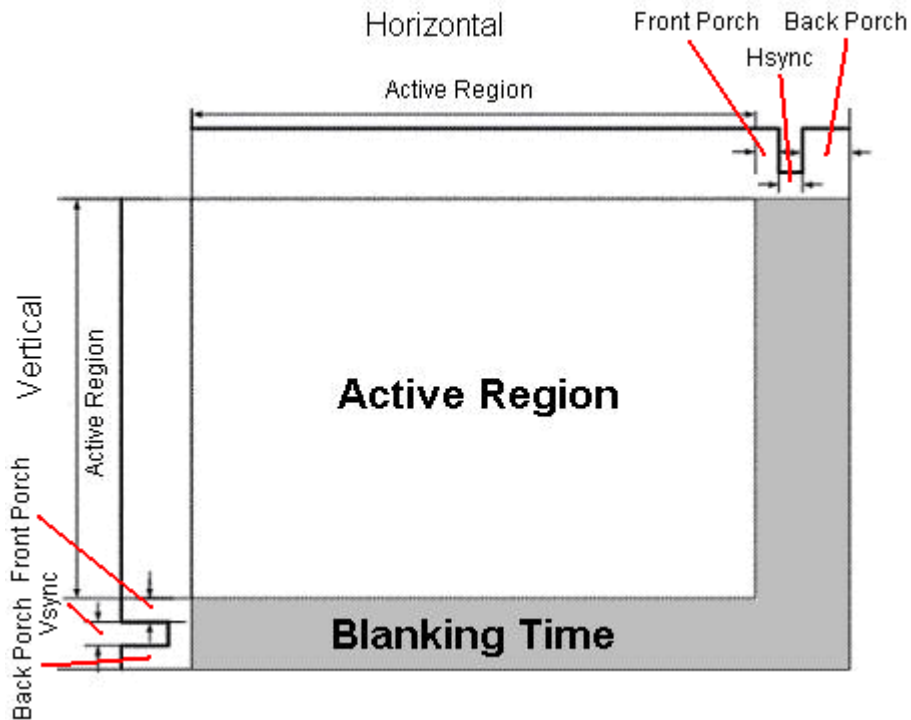
Figure 3-7. Custom Output Timing Editor

Custom Output Timing Editor helps to create a new output timing, modify a created timing or delete one of created timings. This editor can be achieved by the  button on output settings page (this button will be hidden if the output channel is not capable to set custom output timing parameters)

The Custom Output Timing feature will expand the output timing list on output settings page, if it is compliant with the output card capabilities. Several custom timings can be created and they can be used for several output channels also.

This editor gives total access to the parameters in order to create a custom output timing:

- The **Pixel Frequency** (unit: MHz) - the number of pixels per second
- The **Active Area Width** (unit: pixels) - the number of visible pixels per line
- The **H Blanking Interval** (unit: pixels)
- The **H Front Porch** (unit: pixels)
- The **H Sync Width** (unit: pixels)
- The **Active Area Height** (unit: lines) - the number of visible lines
- The **V Blanking Interval** (unit: lines)
- The **V Front Porch** (unit: lines)
- The **V Sync Height** (unit: lines)
- The **Sync polarities** (positive or negative)



Simple mathematical relationships link the different values together:

- $V \text{ Blanking Interval} = V \text{ sync} + V \text{ Back Porch} + V \text{ Front Porch}$
- $H \text{ Blanking Interval} = H \text{ sync} + H \text{ Back Porch} + H \text{ Front Porch}$
- $V \text{ Total Size} = V \text{ Blanking Interval} + \text{Active Area Height}$
- $H \text{ Total Size} = H \text{ Blanking Interval} + \text{Active Area Width}$
- $\text{Line Frequency} = \text{Frame Rate} \times V \text{ Total Size}$

$\text{Pixel Frequency} = \text{Line Frequency} \times H \text{ Total Size} = \text{Frame Rate} \times V \text{ Total Size} \times H \text{ Total Size}$

3.4 Media Player Panel

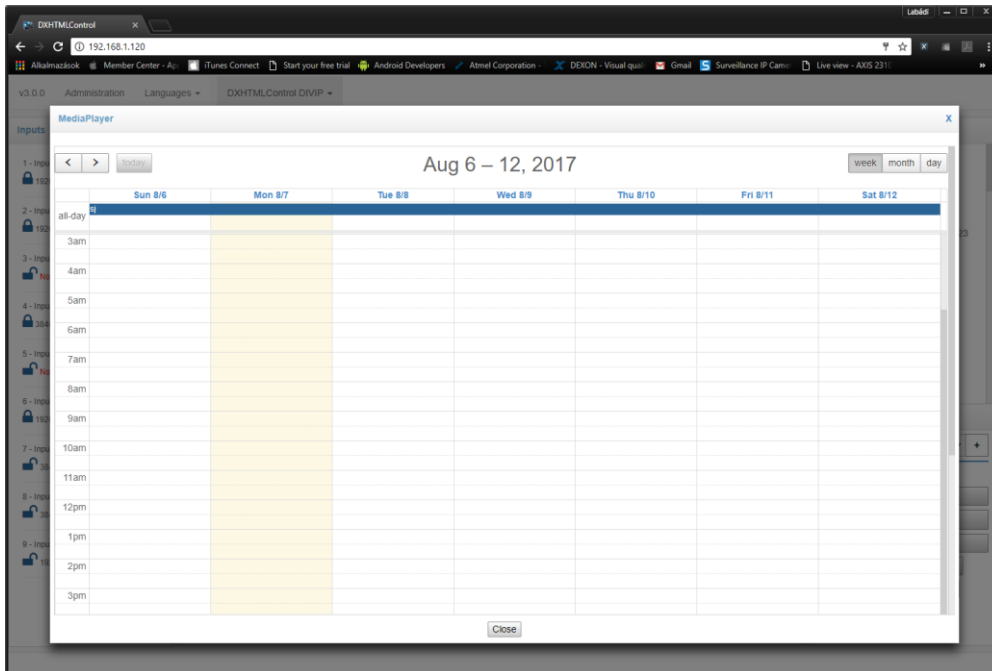


Figure 3-8. Media Player Panel

The **Media Player** feature allows you to create digital-signage presentations. The presentations can be performed with various types of content:

- videos (file or directory) (FHD, H.264)
- network video (file or directory) (FHD, H.264)
- IP streaming (over rtsp protocol)
- HTML5 web pages
- VNC Client

The Media Player source can be accessed by selecting the last input source (DIMAX804KC: #9)

3.4.1 Create Playlist Presentation

Follow these steps to add a presentation:

1. Click the **Playlist** button
2. Click the **New** button
3. Add a **name** of the presentation
4. Select a Presentation type from the **Protocol** dropdown list
5. Fill data form
6. Click **Store** once you have finished creating playlist presentation.

Note: You can activate a presentation with **checkbox**. One presentation can be active at one time, this activation can be change in control section also.

This playlist presentation priority is higher than calendar based presentation.

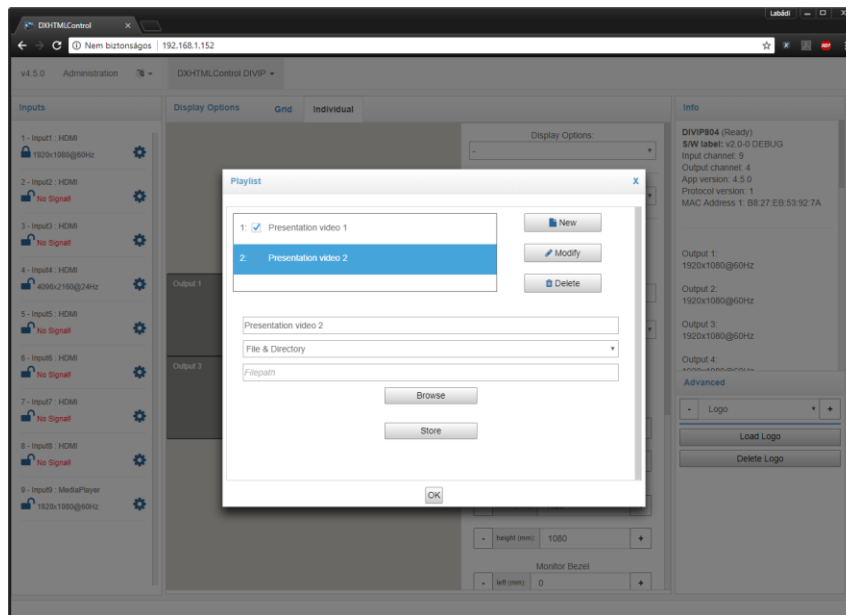


Figure 3-9. Add/modify playlist presentation panel

3.4.2 Create Scheduler Presentation

Follow these steps to add a presentation:

1. Click the **Scheduler** button
2. Click the calendar area
3. Add a **name** of the presentation
4. Select a Presentation type from the **Protocol** dropdown list
5. Fill data form
6. Check the **Active All Day Every Day** box if you would like the presentation to play at all times. This disables all other scheduling options in the Schedule Presentation window.
7. Use the **Start Time** and **End time** fields to specify a time range for the presentation.
8. Check the **Recurring Event** box if you want the presentation to play on a repeating schedule during a specific date range. Check All Days or specify certain days of the week. It is necessary to specify a **Start Date** and **End Date** if the Recurring Event box is checked.
9. Click **OK** once you have finished configuring the calendar based presentation.

Note: You can disable all presentation with **Enable** checkbox, which is located in left top corner.

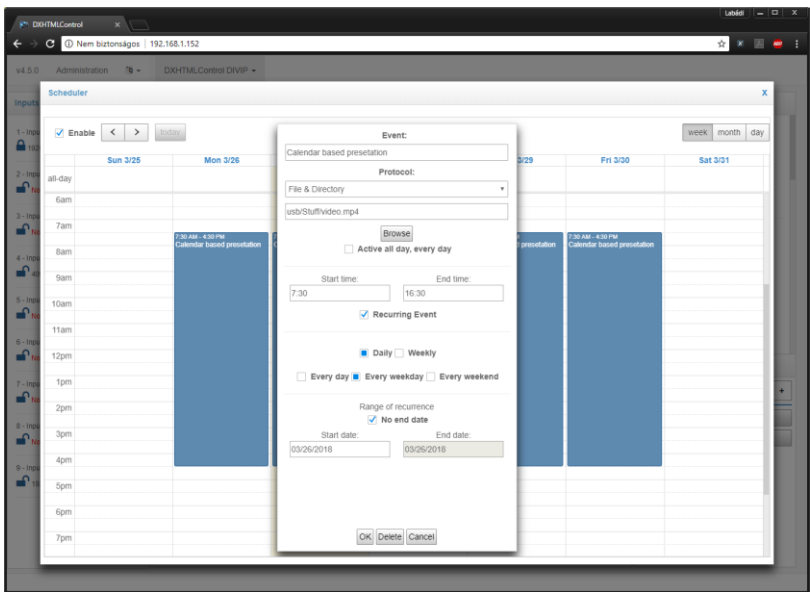


Figure 3-10. Add/modify calendar based presentation panel

3.5 Layout Handling

Users can save certain layout of video windows and create a scheduler of different layouts that is easily accessible. The DXServer stores these layouts and the maximum number of the layouts are 32.

3.5.1 The Calendar based DXScenario User Interface

Users may create calendar based scenarios.

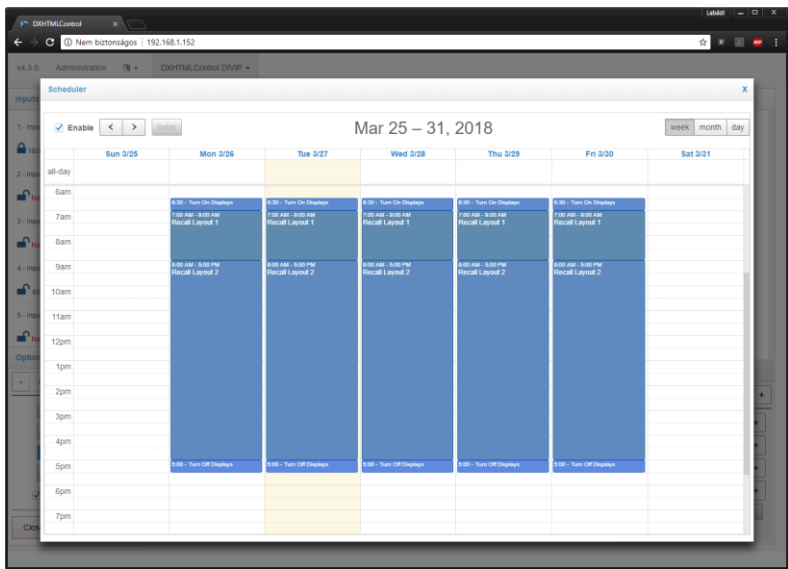


Figure 3-11. Calendar based DXScenario User Window

3.5.1.1 Create Scheduler Event

Follow these steps to add a presentation:

1. Click the **Calendar** button
2. Click the calendar area
3. Add a **name** of the event
4. Select an Event type from the **Protocol** dropdown list (Recall Layout or send string/hex over Serial Port, for example: hex: 01 A4 05 30 01, means 1 byte = 2 hex digit and 1 space)
5. Fill data form
6. Check the **Active All Day Every Day** box if you would like the presentation to play at all times. This type of events will start every start up and every midnight.
7. Use the **Start Time** and **End time** fields to specify a time range for the presentation.
8. Check the **Recurring Event** box if you want the presentation to play on a repeating schedule during a specific date range. Check All Days or specify certain days of the week. It is necessary to specify a **Start Date** and **End Date** if the Recurring Event box is checked.
9. Click **OK** once you have finished configuring the calendar based presentation.

Note: You can disable all events with **Enable** checkbox, which is located in left top corner.

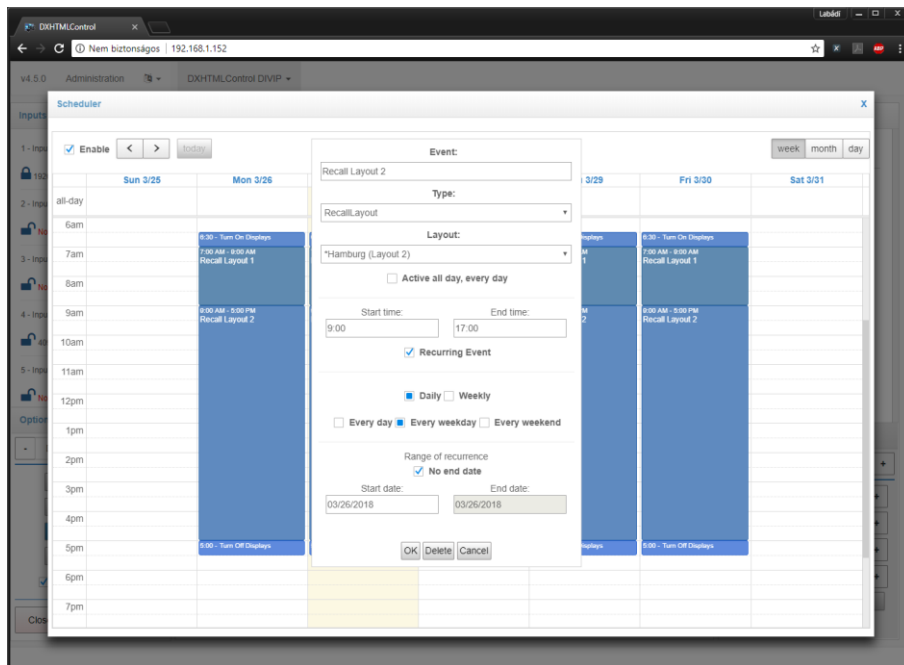


Figure 3-12. Add/modify calendar based events panel

4 EXTERNAL PROGRAMMING

There are three different interfaces to control DIMAX device externally: TCP/IP socket interface, WebSocket interface and RS-232 serial interface. All interfaces use the same XML format communication protocol.

The software of DIMAX device behaves as a server. The client application's request messages are received and processed by the DIVIP server. The requested tasks are performed and reply messages are sent back to the client by the server.

TCP/IP

The DIMAX device can be controlled through TCP/IP socket interface. The software listens to the port 6464 and accepts TCP connections of client applications. It receives and processes the clients' request messages and performs the requested tasks.

Asynchronous events can be sent by the DIVIP server through the port 6465.

WebSocket

The DIMAX device can be controlled through WebSocket interface. The software listens to the port 6466 and accepts WebSocket connections of client applications. It receives and processes the clients' request messages and performs the requested tasks.

Asynchronous events can be sent by the DIVIP server through the port 6467.

RS-232

The DIMAX device can be controlled through RS-232 interface, too.

The parameters of the serial port initially can be set in the front panel menu or from an external program through TCP/IP socket interface, e.g.: the control application described in the previous section.

The default parameters of the RS-232 port:

Baud rate: 9600 bps

Data bits: 8 bits

Parity: no

Stop bits: 1 bit

Flow control: no

The requests and replies have the same format as in case of TCP/IP interface.

Asynchronous events are not sent to the client application through the serial interface.

Note: When the TCP/IP or WebSocket socket is closed the authenticated connection is terminated, therefore you have to keep the socket interface open while you would like to control the Device. If the authentication is terminated you will get this error message: Client is not authenticated! The serial port does not require the authenticated connection.

4.1 Message Format

The request, reply and event messages have common XML based formats. There are different types of messages in terms of complexity:

Message without any parameter:

<msg_name />, e.g: <query_device_info />

Message with one parameter:

```
<msg_name>parameter</msg_name>, e.g. <echo>message</echo>
```

Message with more parameters:

```
<msg_name>
  <param_name1>param1</param_name1>
  <param_name2>param2</param_name2>
</msg_name>
```

e.g.:

```
<config_misc>
  <date>05/01/2015</date>
  <time>10:20:30</time>
</config_misc>
```

Message with attributes and complex parameters

```
<msg_name attr1="a1" attr2="a2">>
  <param_name1>
    <param_name11>param11</param_name11>
    <param_name12>param12</param_name12>
  </param_name1>
</msg_name1>
```

e.g.:

```
<config_input id="1">
  <name>i1</name>
  <port>DVI</port>
  <crop>
    <left>0</left>
    <top>0</top>
    <width>100</width>
    <height>100</height>
  </crop>
</config_input>
```

The detailed description of the protocol messages can be found in the following paragraphs. All the possible parameters are explained in the description of messages. It is allowed to leave out parameters from a request message. The omitted parameters will not be modified. Similarly, the reply and event messages could contain only the changed parameters.

Some attributes can be omitted also, but there are mandatory attributes, too.

4.2 Communication Protocol

4.2.1 Administration

SETUP

setup - Request

Request message:

```
<setup version="1" mfg="Dexon" prod="DIMAX-WebControl" rel="1.0.0">
  <username>Administrator</username>
```

```
<password>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX </password>
<needack>Yes</needack>
</setup>
```

Description:

This request is used to set the authentication level. If the administrator password is set, all the administration messages will be enabled for the client. The authentication is required for all those messages that will change the configuration and status of the device, as **config_input**, **config_output**, **config_misc**, **reboot**, **recall_preset**, etc.

The default value of **needack** parameter can be set up here. There is an attribute called **needack** for all the requests that have no reply message to receive as a result of the request. With the **needack** attribute the client can request an acknowledge reply to the request. The form of the acknowledge reply is the following:

```
<ack request="{tag_name}" />
```

where the tag_name is the name of the request in relation to the message is sent back.

By default, the **ack** message is not sent. If the need of acknowledge is requested here, the **ack** message will be sent as a reply for the following messages without further **needack** setting.

Attributes:

- **version**: protocol version
- **mfg**: manufacturer of the client program
- **prod**: product info of the client program
- **rel**: release number of the client program

Elements:

- **username**: administrator username (Fix: "Administrator")
- **password**: administrator **password** what must be an **md5 hash** string (32 hexadecimal digits)
- **needack**: global parameter for ack request

setup - Reply**Reply message:**

```
<setup />
```

Description:

The DIMAX server sends this empty reply message back to the client.

QUERY_DEVICE_INFO**query_device_info - Request****Request message:**

```
<query_device_info />
```

Description:

The **query_device_info** request is used to obtain general information on the DIMAX device

device_info - Reply

Reply message:

```

<device_info>
  <type>{DIMAX804K|...}</type>
  <state>{Ready|Error}</state>
  <num_presets>n</num_presets>
  <num_layouts>n</num_layouts>
  <num_inputs>n</num_inputs>
  <num_outputs>n</num_outputs>
  <num_overlays>n</num_overlays>
  <num_network_adapters>n</num_network_adapters>
  <num_mediaplayer_events>n</num_mediaplayer_events>
  <num_scheduler_events >n</num_scheduler_events>
  <transition_effect>Cut</transition_effect>
  <transition_effect>Fade</transition_effect>
  <open_effect>Cut</open_effect>
  <open_effect>Fade</open_effect>
  <close_effect>Cut</close_effect>
  <close_effect>Fade</close_effect>
  <scheduler_type>RecallLayout</scheduler_type>
  <scheduler_type>TextOverSerialPort</scheduler_type>
  <scheduler_type>HexOverSerialPort</scheduler_type>
  <input id="n">
    <decoder>{DX804K-IN}</decoder>
    <fw required="n.n.n.n" current="n.n.n.n" />
    <port>HDMI</port>
    <audio_port>Embedded</audio_port>
    <audio_port>Analogue</audio_port>
    <media_player/>
    <hdcp_config/>
    <edid_config/>
    <enable/>
    <volume/>
  </input>
  ...
  <output id="n">
    <fw required="n.n.n.n" current="n.n.n.n" />
    <enable_cap>{Yes|No}</enable_cap>
    <frame_cap>{Yes|No}</frame_cap>
    <opacity_cap>{Yes|No}</opacity_cap>
    <custom_timing_cap>{Yes|No}</custom_timing_cap>
    <rotation_cap>{Yes|No}</rotation_cap>
    <audio_port>Embedded</audio_port>
    <audio_port>Analogue</audio_port>
    <port>HDMI</port>
    <output_modes>
      <mode id="{resolution_id}">{1920x1080@60Hz|...}</mode>
      ...
    </output_modes>
  </output>
  ...
</device_info>

```

Description:

The DIMAX server sends this reply message to the client in response to the **query_device_info** request.

Elements:

- **type:** DIMAX device type
- **state:** status of DIMAX software. (Ready, Error)
- **num_presets:** number of presets
- **num_layouts:** number of layouts
- **num_inputs:** number of inputs
- **num_outputs:** number of outputs
- **num_network_adapters:** number of network adapters
- **num_mediaplayer_events:** number of maximum media player events
- **num_scheduler_events:** number of maximum scheduler events
- **transition_effect:** transition effect supported by the device
- **open_effect:** open effect supported by the device
- **close_effect:** close effect supported by the device
- **scheduler_type:** scheduler event type supported by the device

- **input:** detailed information about the available input identified by **id**:
 - **decoder:** type of the input
 - **fw:** firmware version
 - **port:** available video port
 - **audio_port:** available audio port
 - **hdcp_config:** available HDCP configuration
 - **edid_config:** available EDID configuration
 - **media_player:** available Media Player configuration
 - **enable:** available turn on/off feature
 - **volume:** available analogue volume configuration

- **output:** detailed information about the available output identified by **id**:
 - **fw:** firmware version
 - **port:** available video port
 - **audio_port:** available audio port
 - **enable_cap:** available enable capability
 - **frame_cap:** available frame capability
 - **opacity_cap:** available opacity capability
 - **custom_timing_cap:** available custom timing capability
 - **rotation_cap:** available rotation capability
 - **output_modes:** list of available output modes

4.2.2 Input

QUERY_INPUT_STATUS

query_input_status - Request

Request message:

```
<query_input_status id="n" />
```

Description:

The **query_input_status** request is used to obtain status information on a specific input.

Attributes:

- **id:** input identifier

input_status - Reply**Reply message:**

```
<input_status id="n">
  <name>xxxx</name>
  <port>{HDMI|...}</port>
  <audio_port>{Embedded|Analogue}</audio_port>
  <hdcp_enable>{Yes|No}</hdcp_enable>
  <crop>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </crop>
</input_status>
```

Description:

The DIMAX server sends **input_status** reply message to the client in response to the **query_input_status** request. The DIMAX server sends **input_status** event message to the client if one or more parameters have been changed by another client program or by the operator using the front panel menu.

Attributes:

- **id:** input identifier

Elements:

- **name:** name of the input
- **port:** currently selected video port
- **audio_port:** currently selected audio port
- **crop:** current cropping settings. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the input image. (E.g.: the full picture without any cropping: left:0, top:0, width:100, height:100, the top left quarter of the input image: left:50, top:0, width:50, height:50). The cropping parameters can be floating point values with 2 decimal point accuracy. (E.g.: left:0.45, width: 99.23)
- **hdcp_enable:** HDCP enable state

CONFIG_INPUT**config_input - Request****Request message:**

```
<config_input id="n" needack="{Yes|No}">
  <name>xxxx</name>
  <port>{HDMI|...}</port>
  <audio_port>{Embedded|Analogue}</audio_port>
  <hdcp_enable>{Yes|No}</hdcp_enable>
  <crop>
    <left>n,nn</left>
    <top>n,nn</top>
    <right>n,nn</right>
    <bottom>n,nn</bottom>
```

```
</crop>
</config_input>
```

Description:

The **config_input** request is used to set parameters of a specific input. This request is available only for clients with administrator privilege.

Attributes:

- **id**: input identifier

Elements:

See details in description of **input_status** reply message.

input_status - Reply

See details in description of **query_input_status** message.

QUERY_INPUT_INFO**query_input_info - Request****Request message:**

```
<query_input_info id="n" />
```

Description:

The **query_input_info** request is used to obtain timing information on a specific input.

Attributes:

- **id**: input identifier

input_info - Reply**Reply message:**

```
<input_info id="n" valid="{Yes|No}">
  <hres>n</hres>
  <vres>n</vres>
  <hfreq>{n|n/a}</hfreq>
  <vfreq>n</vfreq>
  <htotal>{n|n/a}</htotal>
  <vtotal>{n|n/a}</vtotal>
  <hstart>{n|n/a}</hstart>
  <vstart>{n|n/a}</vstart>
  <hactive>{n|n/a}</hactive>
  <vactive>{n|n/a}</vactive>
  <hpol>{Positive|Negative|n/a}</hpol>
  <vpol>{Positive|Negative|n/a}</vpol>
  <interlaced>{Yes|No|n/a}</interlaced>
  <hdcp>{Yes|No}</hdcp>
  <hdcp_version>{n/a|1.X|2.0|2.2 Type 0|2.2 Type 1}</hdcp_version>
</input_info>
```

Description:

The DIMAX server sends **input_info** reply message to the client in response to the **query_input_info** request. The DIMAX server sends **input_info** event message to the client if the input signal has been changed.

Attributes:

- **id**: input identifier
- **valid**: **Yes**, if input signal is valid, **No**, if the input signal is invalid.

Elements:

- **hres**: horizontal resolution
- **vres**: vertical resolution
- **hfreq**: horizontal frequency
- **vfreq**: vertical frequency
- **htotal**: horizontal total
- **vtotal**: vertical total
- **hstart**: horizontal start
- **vstart**: vertical start
- **hactive**: horizontal active
- **vactive**: vertical active
- **hpol**: horizontal polarity
- **vpol**: vertical polarity
- **interlaced**: Possible values: **Yes**, **No**
- **hdcp**: Possible values: **Yes**, **No**
- **hdcp_version**: Possible values: **n/a**, **1.X**, **2.0**, **2.2 Type 0**, **2.2 Type 1**

4.2.3 Output

QUERY_OUTPUT_STATUS

query_output_status - Request

Request message:

```
<query_output_status />
```

Description:

The **query_output_status** request is used to obtain the HDCP status of the outputs

output_status - Reply

Reply message:

```
<output_status>
  <hdcp id="n">{Yes|No}</hdcp>
  ...
</output_status>
```

Description:

The DIMAX server sends **output_status** reply message to the client in response to the **query_output_status** request. The DIMAX server sends **output_status** event message to the client if the HDCP status on one or more outputs has been changed.

Attributes:

- **id**: output identifier

Elements:

- **hdcp**: **Yes**, if output HDCP is enabled, **No**, if the output HDCP is disabled.

QUERY_OUTPUT_CONFIG

query_output_config - Request

Request message:

```
<query_output_config id="n" />
```

Description:

The **query_output_config** request is used to obtain configuration information on a specific output.

Attributes:

- **id**: output identifier

output_config - Reply

Reply message:

```
<output_config id="n">
  <name>xxxx</name>
  <background_color>0xhhhhh</background_color>
  <resolution recommended_mode="{Yes|No}">n</resolution>
  <hdcp_enable>{Yes|No}</hdcp_enable>
  <keepaspectratio keep_dimension="{Width|Height}"
keep_position="{Center|Start|End}">{No|Yes}</keepaspectratio>
  <pip_keepaspectratio keep_dimension="{Width|Height}"
keep_position="{Center|Start|End}">{No|Yes}</pip_keepaspectratio>
  <frame left = "n" top = "n" right = "n" bottom = "n" color = "0xhhhhh">{No|Yes}</frame>
  <pip_frame left = "n" top = "n" right = "n" bottom = "n" color
= "0xhhhhh">{No|Yes}</pip_frame>
  <opacity>n</opacity>
  <pip_opacity>n</pip_opacity>
  <crop>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </crop>
  <placement>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
```



```

        <height>n,nn</height>
    </placement>
    <pip_crop>
        <left>n,nn</left>
        <top>n,nn</top>
        <width>n,nn</width>
        <height>n,nn</height>
    </pip_crop>
    <pip_placement>
        <left>n,nn</left>
        <top>n,nn</top>
        <width>n,nn</width>
        <height>n,nn</height>
    </pip_placement>
</output_config>

```

Description:

The DIMAX server sends **output_config** reply message to the client in response to the **query_output_config** and the **config_output** requests. The DIMAX server sends **output_config** event message to the client if one or more parameters of an output have been changed.

Attributes:

- **id**: output identifier

Elements:

- **name**: name of the output
- **background_colour**: background colour
- **resolution**: identifier of resolution. The identifier can be found in the **output_modes** list of the **device_info** reply message
- **hdcp_enable**: name of the output
- **keepaspectratio**: display the video content of a window without any distortion on main channel
- **pip_keepaspectratio**: display the video content of a window without any distortion on second channel
- **frame**: display the video content of a window with frame on main channel
- **pip_frame**: display the video content of a window with frame on second channel
- **opacity**: opacity describes the transparency, where 100 is not transparent at all, 50 is 50% see-through, and 0 is completely transparent on main channel
- **pip_opacity**: opacity describes the transparency, where 100 is not transparent at all, 50 is 50% see-through, and 0 is completely transparent on second channel
- **placement**: the position of the main picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image.
- **crop**: output cropping settings. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image. (E.g.: the full picture without any cropping: left:0, top:0, width:100, height:100, the top left quarter of the output image: left:50, top:0, width:50, height:50). The cropping parameters can be floating point values with 2 decimal point accuracy. (E.g.: left:66.67, width: 33.33)
- **pip_placement**: the position of the second picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image.
- **pip_crop**: the crop of the second picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image. (E.g.: the full picture without any cropping: left:0, top:0, width:100, height:100, the top left quarter of the output image: left:50, top:0, width:50, height:50). The cropping parameters can be floating point values with 2 decimal point accuracy. (E.g.: left:66.67, width: 33.33)

CONFIG_OUTPUT

config_output - Request

Request message:

```

<config_output id="n">
  <name>xxxx</name>
  <background_color>0xhhhhh</background_color>
  <resolution recommended_mode="{Yes|No}">n</resolution>
  <hdcp_enable>{Yes|No}</hdcp_enable>
  <keepaspectratio keep_dimension="{Width|Height}"
keep_position="{Center|Start|End}">{No|Yes}</keepaspectratio>
  <pip_keepaspectratio keep_dimension="{Width|Height}"
keep_position="{Center|Start|End}">{No|Yes}</pip_keepaspectratio>
  <frame left="n" top="n" right="n" bottom="n" color="0xhhhhh">{No|Yes}</frame>
  <pip_frame left="n" top="n" right="n" bottom="n" color
="0xhhhhh">{No|Yes}</pip_frame>
  <opacity>n</opacity>
  <pip_opacity>n</pip_opacity>
  <crop>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </crop>
  <placement>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </placement>
  <pip_crop>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </pip_crop>
  <pip_placement>
    <left>n,nn</left>
    <top>n,nn</top>
    <width>n,nn</width>
    <height>n,nn</height>
  </pip_placement>
</config_output>

```

Description:

The **config_output** request is used to set configuration information on a specific output. This request is available only for clients with administrator privilege.

Attributes:

- **id**: output identifier

Elements:

- **name**: name of the output
- **background_colour**: background colour

- **resolution**: identifier of resolution. The identifier can be found in the **output_modes** list of the **device_info** reply message
- **hdcp_enable**: name of the output
- **keepaspectratio**: display the video content of a window without any distortion on main channel
- **pip_keepaspectratio**: display the video content of a window without any distortion on second channel
- **frame**: display the video content of a window with frame on main channel
- **pip_frame**: display the video content of a window with frame on second channel
- **opacity**: opacity describes the transparency, where 100 is not transparent at all, 50 is 50% see-through, and 0 is completely transparent on main channel
- **pip_opacity**: opacity describes the transparency, where 100 is not transparent at all, 50 is 50% see-through, and 0 is completely transparent on second channel
- **placement**: the position of the main picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image.
- **crop**: output cropping settings. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image. (E.g.: the full picture without any cropping: left:0, top:0, width:100, height:100, the top left quarter of the output image: left:50, top:0, width:50, height:50). The cropping parameters can be floating point values with 2 decimal point accuracy. (E.g.: left:66.67, width: 33.33)
- **pip_placement**: the position of the second picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image.
- **pip_crop**: the crop of the second picture on the output. The **left**, **top**, **width** and **height** parameters are counted in percentage of the full dimensions of the output image. (E.g.: the full picture without any cropping: left:0, top:0, width:100, height:100, the top left quarter of the output image: left:50, top:0, width:50, height:50). The cropping parameters can be floating point values with 2 decimal point accuracy. (E.g.: left:66.67, width: 33.33)

output_config - Reply

See details in description of **query_output_config** message.

QUERY_OUTPUT_TIMING_CONSTRAINTS

query_output_timing_constraints- Request

Request message:

```
<query_output_timing_constraints id="n" />
```

Description:

The **query_output_timing_constraints** request is used to obtain constraint information on a specific output.

Attributes:

- **id**: output identifier

output_timing_constraints - Reply

Reply message:

```
<output_timing_constraints id="n" >
  <pixelclock min="n" max="n" />
```

```

<hactive min="n" max="n" mask="0xhhhhhhh" />
<vactive min="n" max="n" mask="0xhhhhhhh" />
<hblank min="n" max="n" mask="0xhhhhhhh" />
<vblank min="n" max="n" mask="0xhhhhhhh" />
<hfrontporch min="n" max="n" mask="0xhhhhhhh" />
<vfrontporch min="n" max="n" mask="0xhhhhhhh" />
<hsyncwidth min="n" max="n" mask="0xhhhhhhh" />
<vsyncwidth min="n" max="n" mask="0xhhhhhhh" />
<interlaced />
<DSS />
<CSYNC />
<SOG />
</output_timing_constraints>

```

Description:

The DIMAX server sends **output_timing_constraints** reply message to the client in response to the **query_output_timing_constraints** request. This reply adds the limits of the specified output for custom output timing settings.

Attributes:

- **id**: output identifier

Elements:

- **pixelclock min, max**: pixel frequency limits
- **hactive min, max, mask**: horizontal active limits and mask
- **vactive min, max, mask**: vertical active limits and mask
- **hblank min, max, mask**: horizontal blank limits and mask
- **vblank min, max, mask**: vertical blank limits and mask
- **hfrontporch min, max, mask**: horizontal front porch limits and mask
- **vfrontporch min, max, mask**: vertical front porch limits and mask
- **hsyncwidth min, max, mask**: horizontal sync limits and mask
- **vsyncwidth min, max, mask**: vertical sync limits and mask
- **interlaced**: output handles interlaced timing
- **DSS**: output handles Separate sync format
- **CSYNC**: output handles Composite sync format
- **SOG**: output handles Sync-on-green format

QUERY_CUSTOM_OUTPUT_TIMINGS

query_custom_output_timings- Request

Request message:

```
<query_output_custom_timings />
```

Description:

The **query_custom_output_timings** request is used to obtain custom output timings information.

custom_output_timings - Reply

Reply message:

```

<custom_output_timings>
  <timing id="n">
    <pixelclock>n</pixelclock>
  </timing>
</custom_output_timings>

```

```

<interlaced>{Yes|No}</interlaced>
<hactive>n</hactive>
<vactive>n</vactive>
<hblank>n</hblank>
<vblank>n</vblank>
<hfrontporch>n</hfrontporch>
<vfrontporch>n</vfrontporch>
<hsyncwidth>n</hsyncwidth>
<vsyncwidth>n</vsyncwidth>
<syncctype>{DSS|CSYNC|SOG}</syncctype>
<hpol>{Positive|Negative}</hpol>
<vpol>{Positive|Negative}</vpol>
</timing>
...
</custom_output_timings>

```

Description:

The DIMAX server sends **custom_output_timings** reply message to the client in response to the **query_custom_output_timings** request. This reply contains all added custom output timings.

Attributes:

- **id**: custom output timing identifier

Elements:

- **pixelclock**: pixel frequency
- **interlaced**: interlaced, values: Yes, No
- **hactive**: horizontal active
- **vactive**: vertical active
- **hblank**: horizontal blank
- **vblank**: vertical blank
- **hfrontporch**: horizontal front porch
- **vfrontporch**: vertical front porch
- **hsyncwidth**: horizontal sync
- **vsyncwidth**: vertical sync
- **hpol**: horizontal polarity
- **vpol**: vertical polarity
- **syncctype**: synchronization type, values: DSS (Separate sync), CSYNC (Composite sync), SOG (Sync-on-green)

ADD_CUSTOM_OUTPUT_TIMING**add_custom_output_timing - Request****Request message:**

```

<add_custom_output_timing needack="{Yes|No}">
<pixelclock>n</pixelclock>
<interlaced>{Yes|No}</interlaced>
<hactive>n</hactive>
<vactive>n</vactive>
<hblank>n</hblank>
<vblank>n</vblank>
<hfrontporch>n</hfrontporch>
<vfrontporch>n</vfrontporch>
<hsyncwidth>n</hsyncwidth>

```

```

<vsyncwidth>n</vsyncwidth>
<synctype>{DSS|CSYNC|SOG}</synctype>
<hpol>{Positive|Negative}</hpol>
<vpol>{Positive|Negative}</vpol>
</add_custom_output_timing>

```

Description:

The **add_custom_output_timing** request is used to create a new custom output timing, which will expand the output timing list. This request is available only for clients with administrator privilege.

Elements:

See details in description of **query_custom_output_timings** message.

add_custom_output_timing - Reply

```
<custom_output_timing_changed />
```

MODIFY_CUSTOM_OUTPUT_TIMING**modify_custom_output_timing – Request****Request message:**

```

<modify_custom_output_timing id="n" needack="{Yes|No}">
  <pixelclock>n</pixelclock>
  <interlaced>{Yes|No}</interlaced>
  <hactive>n</hactive>
  <vactive>n</vactive>
  <hblank>n</hblank>
  <vblank>n</vblank>
  <hfrontporch>n</hfrontporch>
  <vfrontporch>n</vfrontporch>
  <hsyncwidth>n</hsyncwidth>
  <vsyncwidth>n</vsyncwidth>
  <synctype>{DSS|CSYNC|SOG}</synctype>
  <hpol>{Positive|Negative}</hpol>
  <vpol>{Positive|Negative}</vpol>
</modify_custom_output_timing>

```

Description:

The **modify_custom_output_timing** request is used to modify a specified custom output timing. This request is available only for clients with administrator privilege.

Attributes:

- **id**: custom output timing identifier

Elements:

See details in description of **query_custom_output_timings** message.

modify_custom_output_timing - Reply

```
<custom_output_timing_changed />
```

DELETE_CUSTOM_OUTPUT_TIMING

delete_custom_output_timing - Request

Request message:

```
<delete_custom_output_timing id="n" needack="{Yes|No}" />
```

Description:

The **delete_custom_output_timing** request is used to remove a specified custom output timing, which will expand the output timing list. This request is available only for clients with administrator privilege.

Attributes:

- **id:** custom output timing identifier

add_custom_output_timing - Reply

```
<custom_output_timing_changed />
```

QUERY_OUTPUT_TIMING

query_output_timing- Request

Request message:

```
<query_output_timing id="n" mode_id="n" />
```

Description:

The **query_output_timing** request is used to obtain specified output timings information.

output_timing - Reply

Reply message:

```
<output_timings id="n" mode_id="n">
  <pixelclock>n</pixelclock>
  <interlaced>{Yes|No}</interlaced>
  <hactive>n</hactive>
  <vactive>n</vactive>
  <hblank>n</hblank>
  <vblank>n</vblank>
  <hfrontporch>n</hfrontporch>
  <vfrontporch>n</vfrontporch>
  <hsyncwidth>n</hsyncwidth>
  <vsyncwidth>n</vsyncwidth>
  <synctype>{DSS|CSYNC|SOG}</synctype>
  <hpol>{Positive|Negative}</hpol>
  <vpol>{Positive|Negative}</vpol>
</output_timing>
```

Description:

The DIMAX server sends **output_timing** reply message to the client in response to the **query_output_timing** request. This reply contains all information of specified output timings.

Attributes:

- **id**: output identifier
- **mode_id**: output timing identifier

Elements:

- **pixelclock**: pixel frequency
- **interlaced**: interlaced, values: Yes, No
- **hactive**: horizontal active
- **vactive**: vertical active
- **hblank**: horizontal blank
- **vblank**: vertical blank
- **hfrontporch**: horizontal front porch
- **vfrontporch**: vertical front porch
- **hsyncwidth**: horizontal sync
- **vsyncwidth**: vertical sync
- **hpol**: horizontal polarity
- **vpol**: vertical polarity

4.2.4 Advanced

QUERY_MISC_CONFIG

query_misc_config - Request

Request message:

```
<query_misc_config />
```

Description:

The **query_misc_config** request is used to obtain configuration information on advanced parameters.

misc_config - Reply

Reply message:

```
<misc_config>
  <password protected="{No|Yes}" />
  <mac_address id="n">n:n:n:n:n:n</mac_address>
  <dhcp id="n">{Yes|No}</dhcp>
  <ipv4_address id="n">n.n.n.n</ipv4_address>
  <network_mask id="n">n.n.n.n</network_mask>
  <wakeup_scheduler time="h:i" recurring="{No|Yes}">{No|Yes}</wakeup_scheduler>
  <standby_scheduler time="h:i" recurring="{No|Yes}">{No|Yes}</standby_scheduler>
  <osd left="n,nn" top="n,nn" time="n" bgcolor="0xhhhhh" textcolor="0xhhhhh"
  highlightedcolor="0xhhhhh">{Auto|Yes|No}</osd>
  <transition duration="n">{Cut|Fade}</transition>
  <serial_port>
    <baud_rate>{115200|57600|38400|19200|9600}</baud_rate>
```



```

< databits>{8|7}</ databits>
< parity>{None|Even|Odd|Mark|Space}</ parity>
< stopbits>{1|1.5|2}</ stopbits>
< flow_control>{None|XonXoff|Hardware}</ flow_control>
</ serial_port>
< log_config error="{Yes|No}" warning="{Yes|No}" info="{Yes|No}" hwmonitor="{Yes|No}"
debug="{Yes|No}" trace="{Yes|No}">{Yes|No}</ log_config>
< date>mm/dd/yyyy</ date>
< time>h:i:s</ time>
</ misc_config>

```

Description:

The DIMAX server sends **misc_config** reply message to the client in response to the **query_misc_config** and the **config_misc** requests. The DIMAX server sends **misc_config** event message to the client if one or more advanced parameters have been changed.

Elements:

- **password**: protected=Yes means that the administrator password is set. The password itself is not published, of course.
- **mac_address**: mac address of the network adapter
- **dhcp**: **Yes**: dhcp enabled, **No**: dhcp disabled
- **ipv4_address**: ipv4 address
- **network_mask**: network mask
- **serial_port**: RS-232 port settings.
- **osd**: the **On Screen Display** information panel visualization can be permanently turned **On** or **Off**, or it can be displayed **Automatically** in case of changes in the settings. The duration of automatic display can be set in seconds in the **time** attribute. The position of the panel can be configured with the **top** and **left** coordinates of the area. The coordinates are counted in percentage of the full dimensions of the output image. The colouring of the panel can be set the RGB specifications of the background, text and highlight colours.
- **transition**: possible values: **Cut**, **Fade**. In case of **Fade** the duration of the transition period is specified in seconds by the **Duration** attribute.
- **log_config**: logging configuration. The logging can be turned on or off generally. Further, the different types of log messages can be turned on or off individually: **error**, **warning**, **info**, **hwmonitor**, **debug**, **trace**
- **date** and **time**: the current date and time set on the device
- **wakeup_scheduler**: a time value when the device will deactivate the standby mode
- **standby_scheduler**: a time value when the device will activate the standby mode

CONFIG_MISC

config_misc - Request

Request message:

```

< config_misc needack="{Yes|No}">
< password clear="{No|Yes}">xxxx</ password>
< wakeup_scheduler time="h:i" recurring="{No|Yes}">{No|Yes}</ wakeup_scheduler>
< standby_scheduler time="h:i" recurring="{No|Yes}">{No|Yes}</ standby_scheduler>
< dhcp id="n">{Yes|No}</ dhcp>
< ipv4_address id="n">n.n.n.n</ ipv4_address>
< network_mask id="n">n.n.n.n</ network_mask>
< osd left="n,nn" top="n,nn" time="n" bgcolor="0xhhhhhh" textcolor="0xhhhhhh"
highlightedcolor="0xhhhhhh">{Auto|Yes|No}</ osd>

```

```

<transition duration="n">{Cut|Fade}</transition>
<serial_port>
  <baud_rate>{115200|57600|38400|19200|9600}</baud_rate>
  <databits>{8|7}</databits>
  <parity>{None|Even|Odd|Mark|Space}</parity>
  <stopbits>{1|1.5|2}</stopbits>
  <flow_control>{None|XonXoff|Hardware}</flow_control>
</serial_port>
<log_config error="{Yes|No}" warning="{Yes|No}" info="{Yes|No}" hwmonitor="{Yes|No}"
debug="{Yes|No}" trace="{Yes|No}">{Yes|No}</log_config>
<date>mm/dd/yyyy</date>
<time>h:i:s</time>
</config_misc>

```

Description:

The **config_misc** request is used to set configuration information on advanced parameters. This request is available only for clients with administrator privilege.

Elements:

- **password**: administrator password. The **clear** attribute is used to remove the current password
- **dhcp**: enable or disable dhcp
- **ipv4_address**: ipv4 address. It is required if **dhcp** is disabled.
- **network_mask**: network mask
- **serial_port**: RS-232 port settings.
- **osd**: the **On Screen Display** information panel visualization parameters.
- **button_function**: possible values: **Normal Open**, **Auto Open**
- **transition**: possible values: **Cut**, **Fade**. In case of **Fade** the duration of the transition period is specified in seconds by the **Duration** attribute.
- **log_config**: logging configuration. The logging can be turned on or off generally. Further, the different types of log messages can be turned on or off individually: **error**, **warning**, **info**, **hwmonitor**, **debug**, **trace**
- **date** and **time**: the current date and time set on the device
- **wakeup_scheduler**: a time value when the device will deactivate the standby mode
- **standby_scheduler**: a time value when the device will activate the standby mode

misc_config - Reply

See details in description of **query_misc_config** message.

4.2.5 Layout**QUERY_LAYOUT_STATUS****query_layout_status - Request****Request message:**

```
<query_layout_status />
```

Description:

The **query_layout_status** request is used to obtain configuration information on layout parameters.

layout_status - Reply

Reply message:

```
<layout_status>
  <layout id="n" name="x" active="{No|Yes}">{Not Used|Used}</layout>
  ...
</layout_status>
```

Description:

The DIMAX server sends **layout_status** reply message to the client in response to the **query_layout_status** request. The DIMAX server sends **layout_status** event message to the client if the states of one or more layouts have been changed.

Attributes:

- **id**: layout identifier
- **name**: layout name
- **active**: shows that the layout is currently active or not

Value:

- **Not Used**: the layout identified by **id** is not in use, i.e. does not store any information
- **Used**: the layout identified by **id** stores a layout configuration

RENAME_LAYOUT

rename_layout - Request

Request message:

```
<rename_layout id="n" needack="{Yes|No}" name="x"/>
```

Description:

The **rename_layout** request is used to set name parameter to a layout. This request is available only for clients with administrator privilege.

Attributes:

- **id**: layout identifier
- **name**: layout name

SAVE_LAYOUT

save_layout - Request

Request message:

```
<save_layout id="n" needack="{Yes|No}"/>
```

Description:

The **save_layout** request is used to save the current arrangement to a layout.

Attributes:

- **id**: layout identifier

CLEAR_LAYOUT

clear_layout - Request

Request message:

```
<clear_layout id="n" needack="{Yes|No}"/>
```

Description:

The **clear_layout** request is used to remove the information stored in the layout identified by **id**.

Attributes:

- **id**: layout identifier

RECALL_LAYOUT

recall_layout - Request

Request message:

```
<recall_layout id="n" needack="{Yes|No}"/>
```

Description:

The **recall_layout** request is used to set the arrangement stored in the layout identified by **id** to the DIMAX device.

Attributes:

- **id**: layout identifier

SAVE_LAYOUTS

save_layouts - Request

Request message:

```
<save_layouts/>
```

Description:

The **save_layouts** request is used to query the whole layout configuration in order to save it for future loading. This request is available only for clients with administrator privilege.

save_layouts - Reply

Reply message:

```
<save_layouts>
```

...

```
</save_layouts>
```

Description:

The DIMAX server sends **save_layouts** reply message to the client in response to the **save_layouts** request.

LOAD_LAYOUTS

load_layouts - Request

Request message:

```
<load_layouts needack="{Yes|No}" >
```

...

```
</load_layouts>
```

Description:

The **load_layouts** request is used to load the whole layout configuration information previously saved with the help of **save_layouts** message. This request is available only for clients with administrator privilege.

4.2.6 Preset

QUERY_PRESET_STATUS

query_preset_status - Request

Request message:

```
<query_preset_status />
```

Description:

The **query_preset_status** request is used to obtain configuration information on preset parameters.

preset_status - Reply

Reply message:

```
<preset_status>
```

```
<preset id="n" name="x" active="{No|Yes}">{Empty|Used}</preset>
```

...

```
</preset_status>
```

Description:

The DIMAX server sends **preset_status** reply message to the client in response to the **query_preset_status** request.

The DIMAX server sends **preset_status** event message to the client if the states of one or more presets have been changed.

Attributes:

- **id**: preset identifier
- **name**: preset name
- **active**: shows that the preset is currently active or not

Value:

- **Empty**: the preset identified by **id** is not in use, i.e. does not store any information

- **Used:** the preset identified by **id** stores a set of settings

RENAME_PRESET

rename_preset - Request

Request message:

```
<rename_preset id="n" needack="{Yes|No}" name="x"/>
```

Description:

The **rename_preset** request is used to set name parameter to a preset. This request is available only for clients with administrator privilege.

Attributes:

- **id:** preset identifier
- **name:** preset name

SAVE_PRESET

save_preset - Request

Request message:

```
<save_preset id="n" needack="{Yes|No}"/>
```

Description:

The **save_preset** request is used to save the current configuration and connection information as a preset. This request is available only for clients with administrator privilege.

Attributes:

- **id:** preset identifier

CLEAR_PRESET

clear_preset - Request

Request message:

```
<clear_preset id="n" needack="{Yes|No}"/>
```

Description:

The **clear_preset** request is used to remove the information stored in the preset identified by **id**. This request is available only for clients with administrator privilege.

Attributes:

- **id:** preset identifier

RECALL_PRESET

recall_preset - Request

Request message:

```
<recall_preset id="n" needack="{Yes|No}"/>
```

Description:

The **recall_preset** request is used to set the configuration and connection settings stored in the preset identified by **id** to the DIMAX device. This request is available only for clients with administrator privilege.

Attributes:

- **id**: preset identifier

4.2.7 Video Assignment

QUERY_VIDEO_ASSIGNMENT

query_video_assignment - Request

Request message:

```
<query_video_assignment />
```

Description:

The **query_video_assignment** request is used to obtain information on video assignment.

video - Reply

Reply message:

```
<video>
  <connected input_id="n" output_id="n" channel="{Main|PIP}" />
  ...
  <disconnected output_id="n" channel="{Main|PIP}" />
  ...
</video>
```

Description:

The DIMAX server sends **video** reply message to the client in response to the **query_video_assignment** request. The DIMAX server sends **video** event message to the client if the video assignment has been changed.

Elements:

- **connected**: existing connection between the input identified by **input_id** and the output identified by **output_id** on **channel Main** or **PIP**
- **disconnected**: currently disconnected **channel** on output identified by **output_id**

VIDEO

video - Request

Request message:

```
<video needack="{Yes|No}">
  <connect input_id="n" output_id="n" channel="{Main|PIP}" />
  ...
  <disconnect output_id="n" channel="{Main|PIP}" />
  ...
</video>
```

Description:

The **video** request is used to set video assignment. With the help of this request new video connections can be created and existing connections can be removed.

video - Reply

See details in description of **query_video_assignment** message.

4.2.8 Audio Assignment

QUERY_AUDIO_ASSIGNMENT

query_audio_assignment - Request

Request message:

```
<query_audio_assignment />
```

Description:

The **query_audio_assignment** request is used to obtain information on audio assignment.

audio - Reply

Reply message:

```
<audio>
  <connected input_id="n" output_id="n" />
  ...
  <disconnected output_id="n" />
  ...
</audio>
```

Description:

The DIMAX server sends **audio** reply message to the client in response to the **query_audio_assignment** request. The DIMAX server sends **audio** event message to the client if the audio assignment has been changed.

Elements:

- **connected:** existing connection between the input identified by **input_id** and the output identified by **output_id**
- **disconnected:** currently disconnected output identified by **output_id**

AUDIO**audio - Request****Request message:**

```
<audio needack="{Yes|No}">
  <connect input_id="n" output_id="n" />
  ...
  <disconnect output_id="n" />
  ...
</audio>
```

Description:

The **audio** request is used to set audio assignment. New audio connections can be created and existing connections can be removed by the help of this request.

audio - Reply

See details in description of **query_audio_assignment** message.

4.2.9 Miscellaneous**QUERY_HARDWARE_MONITOR_STATUS****query_hardware_monitor_status - Request****Request message:**

```
<query_hardware_monitor_status />
```

Description:

The **query_hardware_monitor_status** request is used to obtain information on hardware monitor status.

hardware_monitor_status - Reply**Reply message:**

```
<hardware_monitor_status>
  <temperature source="xxx" min="n" max="n" unit="x">n</temperature>
  <tachometer source="xxx" min="n">n</tachometer>
</hardware_monitor_status>
```

Description:

The DIMAX server sends **hardware_monitor_status** reply message to the client in response to the **query_hardware_monitor_status** request.

Elements:

- **temperature:** the currently measured value received from temperature sensor. The sensor is identified by **source** attribute, the range of valid values is specified by **min** and **max** attributes. The measuring **unit** is also specified.
- **tachometer:** the currently measured speed received from an internal fan. The fan is identified by **source** attribute, the minimum speed is specified by **min** attribute.

ECHO

echo - Request

Request message:

`<echo>message_content</echo>`

Description:

In response to the **echo** request the DIMAX server echoes the **message_content** in the **echo** reply.

Reply message:

`<echo>message_content</echo>`

Description:

The DIMAX server sends the same **echo** message back to the client.

SAVE_CONFIG

save_config - Request

Request message:

`<save_config/>`

Description:

The **save_config** request is used to query the whole configuration set in order to save it for future loading.

This request is available only for clients with administrator privilege.

save_config - Reply

Reply message:

`<save_config>`

...

`</save_config>`

Description:

The DIMAX server sends **save_config** reply message to the client in response to the **save_config** request.

LOAD_CONFIG

load_config - Request

Request message:

```
<load_config needack="{Yes|No}" >
...
</load_config>
```

Description:

The **load_config** request is used to load the whole configuration set previously saved with the help of **save_config** message. This request is available only for clients with administrator privilege.

GET_LOG_FILE

get_log_file - Request

Request message:

```
<get_log_file />
```

Description:

The **get_log_file** request is used to ask the logging information saved. This request is available only for clients with administrator privilege.

log_file - Reply

Reply message:

```
<log_file>
<![CDATA[
...
]]>
</log_file>
```

Description:

The DIMAX server sends **log_file** reply message to the client in response to the **get_log_file** request. The logged information is placed to a CDATA section to keep xml format valid.

CUSTOM_FIELD

store_custom_field - Request

Request message:

```
<store_custom_field id="fieldname" needack="{Yes|No}">
...
</store_custom_field>
```

Description:

The **store_custom_field** request is used to store client-specific information on the server for future usage. The format of the custom field data is not specified, it can be any kind of character string.

Attributes:

- **id**: any unique alphanumeric string to identify the data.

query_custom_field - Request

```
<query_custom_field id="n" />
```

Description:

The **query_custom_field** request is used to retrieve previously stored client-specific information from the server.

Attributes:

- **id**: unique alphanumeric string to identify the previously stored data.

custom_field - Reply

Reply message:

```
<custom_field id="n">
```

...

```
</custom_field>
```

Description:

The DIMAX server sends **custom_field** reply message to the client in response to the **query_custom_field** request. The value of the custom field data will be replied in the same format as it was supplied by the client.

REBOOT

reboot - Request

Request message:

```
<reboot needack="{Yes|No}" />
```

Description:

The **reboot** request is used to perform a restart on DIMAX device. This request is available only for clients with administrator privilege.

WAKEUP

wakeup - Request

Request message:

```
<wake_up needack="{Yes|No}" />
```

Description:

The **wakeup** request is used to deactivate the standby mode. This request is available only for clients with administrator privilege.

STANDBY

standby - Request

Request message:

```
<standby needack="{Yes|No}" />
```

Description:

The **standby** request is used to activate the standby mode.
This request is available only for clients with administrator privilege.

5 SOFTWARE AND FIRMWARE UPDATES

This operating software and firmware may be updated for possible improvements and new features.

- Step 1 **Create an installation USB memory stick**
The update package contains folder called DUA_pollingPi. Copy this folder to the root folder of the pendrive.
Be sure that the memory stick is not write protected and file system is FAT32. The software has to remove some installation files (*.sh) during the update process from the pendrive.
- Step 2 **Plug the USB stick into the USB port**
The software will automatically recognize the installation material within 1-2 minutes. The upgrade process will be started automatically.
During the upgrade process the device will be restarted two times.
- Step 3 The device automatically checks the firmware version. If there is a version mismatch, then the F/W updating start automatically.

The progress of the update will be displayed on the LCD screen.

Do not interrupt this process. When the update process is finished, the LCD screen displays the initial login message – hopefully without the firmware warning.

6 SPECIFICATIONS

6.1 Electrical, Mechanics and Environment

Size	435 x 328 x 45 mm
Weight	3200 g
Power supply	100 – 240 V Auto ranging
Dissipation	max. 90 W
Cooling	cooling controller fan
Operating temperature range	5- 40 degrees centigrade
Operating humidity range	10 – 90% non-condensing
Storage temperature range	-20 - +60 degrees centigrade
Storage humidity range	10 – 90% non-condensing
Altitude	0 – 2000 m

6.2 Input Signals

HDMI signal timings and standards

Mode	Horizontal Frequency	Pixel Frequency	Standard
640 x 350 @ 85Hz	37.9 kHz	31.5 MHz	DMT ID 01h
640 x 400 @ 85Hz	37.9 kHz	31.5 MHz	DMT ID 02h
640 x 480 @ 60Hz	31.5 kHz	25.175 MHz	DMT ID 04h
640 x 480 @ 72Hz	37.9 kHz	31.5 MHz	DMT ID 05h
640 x 480 @ 75Hz	37.5 kHz	31.5 MHz	DMT ID 06h
640 x 480 @ 85Hz	43.3 kHz	36.0 MHz	DMT ID 07h
720 x 400 @ 85Hz	37.9 kHz	35.5 MHz	DMT ID 03h
800 x 600 @ 56Hz	35.23 kHz	36.0 MHz	DMT ID 08h
800 x 600 @ 60Hz	37.9 kHz	40.0 MHz	DMT ID 09h
800 x 600 @ 72Hz	48.1 kHz	50.0 MHz	DMT ID 0Ah
800 x 600 @ 75Hz	46.9 kHz	49.5 MHz	DMT ID 0Bh
800 x 600 @ 85Hz	53.7 kHz	56.25 MHz	DMT ID 0Ch
800 x 600 @ 120Hz RB	76.3 kHz	73.25 MHz	DMT ID 0Dh
848 x 480 @ 60Hz	31.0 kHz	33.75 MHz	DMT ID 0Eh
1024 x 768 @ 60Hz	48.4 kHz	65.0 MHz	DMT ID 10h
1024 x 768 @ 70Hz	56.5 kHz	75.0 MHz	DMT ID 11h
1024 x 768 @ 75Hz	60.0 kHz	78.75 MHz	DMT ID 12h
1024 x 768 @ 85Hz	68.7 kHz	94.5 MHz	DMT ID 13h
1024 x 768 @ 120Hz RB	97.6 kHz	115.5 MHz	DMT ID 14h
1152 x 864 @ 75Hz	67.5 kHz	108.0 MHz	DMT ID 15h
1280 x 768 @ 60Hz RB	47.4 kHz	68.25 MHz	DMT ID 16h
1280 x 768 @ 60Hz	47.8 kHz	79.5 MHz	DMT ID 17h
1280 x 768 @ 75Hz	60.3 kHz	102.25 MHz	DMT ID 18h

1280 x 768 @ 85Hz	68.6 kHz	117.5 MHz	DMT ID 19h
1280 x 768 @ 120Hz RB	97.4 kHz	140.25 MHz	DMT ID 1Ah
1280 x 800 @ 60Hz RB	49.3 kHz	71.0 MHz	DMT ID 1Bh
1280 x 800 @ 60Hz	49.7 kHz	83.5 MHz	DMT ID 1Ch
1280 x 800 @ 75Hz	62.8 kHz	106.5 MHz	DMT ID 1Dh
1280 x 800 @ 85Hz	71.6 kHz	122.5 MHz	DMT ID 1Eh
1280 x 800 @ 120Hz RB	101.6 kHz	146.25 MHz	DMT ID 1Fh
1280 x 960 @ 60Hz	60.0 kHz	108.0 MHz	DMT ID 20h
1280 x 960 @ 85Hz	85.9 kHz	148.5 MHz	DMT ID 21h
1280 x 1024 @ 60Hz	64.0 kHz	108.0 MHz	DMT ID 23h
1280 x 1024 @ 75Hz	80.0 kHz	135.0 MHz	DMT ID 24h
1280 x 1024 @ 85Hz	91.1 kHz	157.7 MHz	DMT ID 25h
1360 x 768 @ 60Hz	47.7 kHz	85.5 MHz	DMT ID 27h
1360 x 768 @ 120Hz RB	97.5 kHz	148.25 MHz	DMT ID 28h
1366 x 768 @ 60Hz	47.7 kHz	85.5 MHz	DMT ID 51h
1400 x 1050 @ 60Hz RB	64.7 kHz	101.0 MHz	DMT ID 29h
1400 x 1050 @ 60Hz	65.3 kHz	121.75 MHz	DMT ID 2Ah
1400 x 1050 @ 75Hz	82.3 kHz	156.0 MHz	DMT ID 2Bh
1440 x 900 @ 60Hz RB	55.5 kHz	88.75 MHz	DMT ID 2Eh
1440 x 900 @ 60Hz	55.9 kHz	106.5 MHz	DMT ID 2Fh
1440 x 900 @ 75Hz	70.6 kHz	136.75 MHz	DMT ID 30h
1440 x 900 @ 85Hz	80.4 kHz	157.0 MHz	DMT ID 31h
1600 x 1200 @ 60Hz	75.0 kHz	162.0 MHz	DMT ID 33h
1680 x 1050 @ 60Hz RB	64.7 kHz	119.0 MHz	DMT ID 39h
1680 x 1050 @ 60Hz	65.3 kHz	146.25 MHz	DMT ID 3Ah
1920 x 1200 @ 60Hz RB	74.0 kHz	154.0 MHz	DMT ID 44h
1920 x 1080 @ 60Hz	67.5 kHz	148.5 MHz	DMT ID 52h
640 x 480 @ 59.94Hz	31.469 kHz	25.175 MHz	CEA-861-E VIC 1
720 x 480 @ 59.94Hz	31.469 kHz	27.0 MHz	CEA-861-E VIC 2, 3
720 x 576 @ 50Hz	31.25 kHz	27.0 MHz	CEA-861-E VIC 17, 18
1280 x 720 @ 50Hz	37.5 kHz	74.25 MHz	CEA-861-E VIC 19
1280 x 720 @ 60Hz	45.0 kHz	74.25 MHz	CEA-861-E VIC 4
1440 x 288 @ 50Hz	15.625 kHz	27.0 MHz	CEA-861-E VIC 23, 24 *
1440 x 240 @ 60Hz	15.734 kHz	27.0 MHz	CEA-861-E VIC 8, 9 *
1440 x 480 @ 59.94Hz	31.469 kHz	54.0 MHz	CEA-861-E VIC 14, 15
1440 x 576 @ 50Hz	31.25 kHz	54.0 MHz	CEA-861-E VIC 29, 30
1920 x 1080 @ 24Hz	27.0 kHz	74.25 MHz	CEA-861-E VIC 32
1920 x 1080 @ 25Hz	28.125 kHz	74.25 MHz	CEA-861-E VIC 33
1920 x 1080 @ 30Hz	33.75 kHz	74.25 MHz	CEA-861-E VIC 34
1920 x 1080 @ 60 Hz	67.5 kHz	148.5 MHz	CEA-861-E VIC 16
2560 x 1400 @ 60 Hz			
3840 x 2160 @ 24Hz			
3840 x 2160 @ 30Hz			
4096 x 2160 @ 30Hz			
All the CVT compliant timings that have pixel frequency <= 300 MHz			

* with overscan

6.3 Output Signals

Supported output resolutions

Resolution	Note
640x480@60Hz	
720x480@60Hz	
720x576@50Hz	
800x600@60Hz	
848x480@60Hz	
960x540@60Hz	
960x600@60Hz	
1024x768@60Hz	
1280x720@60Hz	
1280x768@60Hz	
1280x800@60Hz	Reduce Blanking
1280x800@60Hz	
1280x960@60Hz	
1280x1024@60Hz	
1360x768@60Hz	
1366x768@60Hz	Reduce Blanking
1366x768@60Hz	
1400x900@60Hz	Reduce Blanking
1400x900@60Hz	
1400x1050@60Hz	Reduce Blanking
1400x1050@60Hz	
1600x900@60Hz	Reduce Blanking
1680x1050@60Hz	Reduce Blanking
1680x1050@60Hz	
1600x1200@60Hz	Reduce Blanking
1920x1080@50Hz	
1920x1080@60Hz	
1920x1200@60Hz	Reduce Blanking
1920x1200@60Hz	
1920x1440@60Hz	
2048x1152@60Hz	Reduce Blanking
2560x1440@60Hz	Reduce Blanking
2560x1600@60Hz	Reduce Blanking
2848x1602@60Hz	Reduce Blanking
3840x2160@24Hz	Reduce Blanking
3840x2160@25Hz	Reduce Blanking
3840x2160@30Hz	Reduce Blanking
3840x2160@30Hz	
4096x2160@30Hz	

7 PROGRAMMING SAMPLES

The first steps should be the following:

Log in with username and password into 6464 port (WS: 6466):

```
<setup version="1">
  <username>Administrator</username>
  <password> f24850daaaca9ec1e55e43fa50d006e3</password>
</setup>
```

The reply message will describe the permissions for the user:

```
<setup standalone="Yes" admin="Yes">
  <permissions>
    <config_input>Yes</config_input>
    <config_output>Yes</config_output>
    <custom_output_timing>Yes</custom_output_timing>
    <video_control>Yes</video_control>
    <audio_control>Yes</audio_control>
    <modify_layout>Yes</modify_layout>
    <run_layout>Yes</run_layout>
  </permissions>
</setup>
```

Note: When the socket is closed the authenticated connection is terminated, therefore you have to keep the socket interface open while you would like to control the Device. If the authentication is terminated you will get this error message: Client is not authenticated!

7.1 Programming Samples for Matrix Switcher Subsystem

The reply message of the <query_device_info /> request will be like this:

```
<device_info>
  <type>DIMAX804K</type>
  <sw_label>DX20170915</sw_label>
  <state>Ready</state>
  <num_layouts>32</num_layouts>
  <num_inputs>4</num_inputs>
  <num_outputs>4</num_outputs>
  <num_overlays>1</num_overlays>
  <transition_effect>Cut</transition_effect>
  <transition_effect>Fade</transition_effect>
  <input id="1">
    <decoder>D4IN-HA4</decoder>
    <port>HDMI</port>
    <audio_port>Embedded</audio_port>
    <audio_port>Analogue</audio_port>
    <preview />
    <hdcp_config />
    <edid_config />
  </input>
  ...

  <output id="1">
    <background_image_cap>No</background_image_cap>
```

```
<custom_timing_cap>No</custom_timing_cap>
<output_mode_cap>No</output_mode_cap>
<port>HDMI</port>
</output>
...
</device_info>
```

To query the actual assignments, send the following:
<query_video_assignment />

The reply message is something similar:

```
<video>
<connected input_id="1" output_id="1" channel="Main"/>
<connected input_id="2" output_id="2" channel="Main"/>
<connected input_id="3" output_id="3" channel="Main"/>
<connected input_id="4" output_id="4" channel="Main"/>
</video>
```

Note: channel can be Main or PIP.

To open and close video

To open input #1 on output #1 channel:

```
<video><connect input_id="1" output_id="1" channel="Main"/></video>
```

To open input #1 on all outputs:

```
<video>
<connect input_id="1" output_id="1" channel="Main"/>
<connect input_id="1" output_id="2" channel="Main"/>
<connect input_id="1" output_id="3" channel="Main"/>
<connect input_id="1" output_id="4" channel="Main"/>
</video>
```

To change input #1 to input #2 on output #1:

```
<video>
<connect input_id="2" output_id="1" channel="Main"/>
</video>
```

To change the output #2 to input #4 and close output #3.

```
<video>
<connect input_id="4" output_id="2" channel="Main"/>
<disconnect output_id="3" channel="Main"/>
</video>
```

To open and close audio

To open input #1 audio on output #1 audio channel:

```
<audio><connect input_id="1" output_id="1"/></audio>
```