

4K@60 4:4:4 対応

モジュール式シームレスビデオプロセッサ

DVP500x シリーズ

コマンドガイド Ver.1.0.0

- この度は、本製品をお買い上げいただきまして誠にありがとうございます。
- 本製品の性能を十分に引き出してご利用いただくために、ご使用前に必ずこの取扱説明書をお読みください。また、お読みになった後は、本製品近くの見やすい場所に保管してください。

この取扱説明書をお読みいただく前に

- この取扱説明書の無断転載を禁じます。
- お客様がお持ちの製品のバージョンによっては、この取扱説明書に記載される外観図やメニュー項目などが、一部異なる場合がありますのでご了承ください。
- 取扱説明書は改善のため、事前の予告なく変更することがあります。最新の取扱説明書は、弊社のホームページからダウンロードすることができます。

<http://www.arvanics.com>

商標について

- HDMI、High-Definition Multimedia Interface、および HDMI ロゴ は、米国およびその他の国における HDMI Licensing, LLC の商標または、登録商標です。
- その他、記載されている会社名、製品名は、各社の登録商標または商標です。なお、本文中において、®マークや™マークを省略している場合があります。

目次

1	外部制御プログラミング	4
1.1	通信方法について	4
1.2	メッセージのフォーマット	4
2	コミュニケーションプロトコル	6
2.1	アドミニストレーション	6
2.2	入力	10
2.3	出力	16
2.4	レイアウト	22
2.5	ビデオウィンドウステータス	24
2.6	音声割り当て	26
2.7	その他	27

1 外部制御プログラミング

1.1 通信方法について

本機を外部から制御するには、TCP/IP ソケットと RS-232C シリアル の 2 種類のインターフェースを利用できます。両インターフェースは同じ XML フォーマットのコミュニケーションプロトコルを利用します。本機はサーバーとして動作します。クライアントアプリケーションから送信された要求メッセージを本機が受信し、要求されたタスクが実行されます。応答メッセージが本機からクライアントへ送信されます。

【TCP/IP】

TCP/IP ソケットインターフェースを使って本機を制御できます。ソフトウェアはポート番号 6464 で待ち受け、クライアントアプリケーションの TCP コネクションを受け付けます。ソフトウェアはクライアントの要求メッセージを受け取って処理を行い、要求されたタスクを実行します。非同期のイベントはイベントポート経由で送信されます。

【RS-232C】

RS-232C インターフェースを使って本機を制御することもできます。シリアルポートの初期パラメータはフロントパネルのメニュー、もしくは上述したコントロールアプリケーション等の TCP/IP ソケットインターフェースの外部プログラムによって設定ができます。デフォルトのパラメータは以下になります。

ボーレート : 9600bps
データビット : 8bit
パリティ : no
ストップビット : 1bit
フローコントロール : no

要求と応答の各メッセージは、TCP/IP インターフェースと同様のフォーマットになります。シリアルインターフェースでは、非同期のイベントはクライアントアプリケーションに送信されません。

1.2 メッセージのフォーマット

要求、応答、イベントの各メッセージは、XML ベースの共通フォーマットで構成されています。複雑さによって、異なったタイプのメッセージがあります。

【パラメータのないメッセージ】

<msg_name />
例 : <query_device_info />

【1つのパラメータを有するメッセージ】

```
<msg_name>parameter</msg_name>
```

```
例： <echo>message</echo>
```

【2つ以上のパラメータを持つメッセージ】

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

例：

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

【属性と複雑なパラメータを有するメッセージ】

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

例：

```
<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>
```

後述の章にて、プロトコルメッセージの詳細について説明します。

使用できるパラメータは、メッセージの説明内に記載されています。要求メッセージからパラメータを省略することも可能です。省略されたパラメータは、修正されません。同様に、応答メッセージやイベントメッセージは変更されたパラメータのみが含まれます。

いくつかの属性は省略することができますが、必須の属性もあります。

2 コミュニケーションプロトコル

2.1 アドミニストレーション

SETUP

接続時のユーザー認証に使用します。	
詳細	<p>本機の設定変更やステータス確認を行う要求メッセージ（例：入出力の設定、再起動、シャットダウンなど）を送信するためには、最初にユーザー認証が必要です。認証が完了すると、設定されているすべての権限がクライアントに対して有効になります。非同期のメッセージは権限に応じてクライアントに送信されます。</p> <p>Needack パラメータのデフォルト値が設定可能です。要求結果の応答メッセージを持たないコマンドには、needack 属性を設定することで、送信したコマンドに対する ack メッセージを受け取ることができます。この応答メッセージの形式は以下の通りです。</p> <pre><ack request="{tag_name}" /></pre> <p>tag_name は、応答対象の要求メッセージ名になります。</p> <p>デフォルトでは、ack メッセージは送信されません。認証時に ack メッセージが要求された場合は、以降の要求メッセージに needack を設定しなくても、ack メッセージが返されます。</p>
要求メッセージ	<pre><setup version="1" > <username>xxx</username> <password>xxx</password> <needack>Yes</needack> </setup></pre>
応答メッセージ	<pre><setup name="xxx" admin="{No Yes}" session="id"> <permissions> <config_input>{No Yes}</config_input> <config_output>{No Yes}</config_output> <custom_output_timing>{No Yes}</custom_output_timing> <video_control>{Yes No}</video_control> <audio_control>{Yes No}</audio_control> <modify_layout>{Yes No}</modify_layout> <run_layout>{Yes No}</run_layout> </permissions> </setup></pre>
属性	<ul style="list-style-type: none"> ● version: プロトコルバージョン
要素	<ul style="list-style-type: none"> ● username: 認証に使用するユーザー名 ● password: ユーザーのパスワード ● needack: ack 要求メッセージのためのグローバルパラメータ

QUERY_DEVICE_INFO

本機の全般的な情報を取得します。	
詳細	本機は query_device_info 要求メッセージを受信すると、 device_info 応答メッセージをクライアントに送信します。
要求メッセージ	<code><query_device_info /></code>
応答メッセージ	<pre> <device_info> <type>{DIVIP404 DIVIP808 ...}</type> <sw_label>{DXyyyymmdd}</sw_label> <state>{Ready RestartNeeded FwUpdateNeeded FwUpdating Error}</state> <num_layouts>n</num_layouts> <num_inputs>n</num_inputs> <num_outputs>n</num_outputs> <num_overlays>n</num_overlays> <num_windows>n</num_windows> <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>{DM1U4 ...}</decoder> <port>DVI</port> <port>HDMI</port> <port>RGB</port> <port>DP</port> <port>SV</port> <port>CVBS</port> <port>YUV</port> <port>SDI</port> <port>CAT6</port> <port>Optical</port> <port>Quad Video</port> <audio_port>Embedded</audio_port> <audio_port>Analogue</audio_port> <audio_port>S/PDIF</audio_port> <standard>NTSC</standard> <standard>PAL</standard> <standard>SECAM</standard> </wss /> <brightness min="x" max="y" default="z" /> <contrast min="x" max="y" default="z" /> <hue min="x" max="y" default="z" /> <saturation min="x" max="y" default="z" /> </pre>

	<pre> <sharpness min="x" max="y" default="z" /> <preview hls="{Yes No}" udp="{Yes No}" rtsp="{Yes No}" /> <hdcp_config /> <edid_config /> </input> ... <output id="n"> <background_image_cap>{No Yes}</background_image_cap> <custom_timing_cap>{No Yes}</custom_timing_cap> <output_mode_cap>{No Yes}</output_mode_cap> <port>DVI</port> <port>HDMI</port> <port>RGB</port> <port>DP</port> <port>SV</port> <port>CVBS</port> <port>YUV</port> <port>SDI</port> <port>CAT6</port> <port>Optical</port> <output_modes> <mode id="{resolution_id}">{1920x1080@60Hz ...}</mode> ... </output_modes> </output> ... <output_grid> <output_modes> <mode id="{resolution_id}">{1920x1080@60Hz ...}</mode> ... </output_modes> </output_grid> </device_info> </pre>
<p>要素</p>	<ul style="list-style-type: none"> ● type: 本機のサブシステムタイプ (video processor) ● state: DIVIP ソフトウェアのステータス (Ready, FwUpdateNeeded, FwUpdating, Error) ● num_layouts: レイアウトの数 ● num_inputs: 入力の数 ● num_outputs: 出力の数 ● num_windows: ウィンドウの最大数 ● num_scheduler_events: スケジューライベントの最大数 ● transition_effect: 本機で対応している切替効果 ● open_effect: 本機で対応しているオープン時の切替効果 ● close_effect: 本機で対応しているクローズ時の切替効果 ● scheduler_type: 本機で対応しているスケジュールタイプ ● input: ID によって識別される利用可能な入力の詳細情報

- **decoder**: 入力のタイプ
- **fw**: ファームウェアバージョン
- **port**: 利用可能なビデオポート
- **audio_port**: 利用可能なオーディオポート
- **brightness**: ブライツネスのデフォルト値
- **contrast**: コントラストのデフォルト値
- **hue**: ヒューのデフォルト値
- **saturation**: サチュレーションのデフォルト値
- **sharpness**: シャープネスのデフォルト値
- **preview**: プレビューイメージ (HLS, RTSP, UDP) の可否
- **hdcp_config**: 利用可能な HDCP 設定
- **edid_config**: 利用可能な EDID 設定
- **background_image_cap**: 出力のバックグラウンドイメージ利用可否
- **output_modes**: 利用可能な出力モードのリスト

2.2 入力

QUERY_INPUT_STATUS

指定した入力チャンネルのステータス情報を取得します。	
詳細	<p>本機は query_input_status 要求メッセージを受信すると、input_status 応答メッセージをクライアントに送信します。</p> <p>本機のフロントパネルボタンを使ってパラメータが変更された場合にも、クライアントに input_status イベントメッセージを送信します。</p>
要求メッセージ	<code><query_input_status id="n" /></code>
応答メッセージ	<pre> <input_status id="n"> <name>xxxx</name> <port>{DVI RGB SV ...}</port> <audio_port>{Embedded Analogue S/PDIF Mute}</audio_port> <brightness>n</brightness> <contrast>n</contrast> <hue>n</hue> <saturation>n</saturation> <sharpness>n</sharpness> <overscan>{Off Medium Full}</overscan> <standard>{NTSC PAL SECAM}</standard> <wss>{No Yes}</wss> <hdcp_enable>{Yes No}</hdcp_enable> <crop> <left>n</left> <top>n</top> <right>n</right> <bottom>n</bottom> </crop> </input_status> </pre>
属性	id : 入力識別子
要素	<ul style="list-style-type: none"> ● name: 入力の名前 ● port: 現在選択されているビデオボード ● audio_port: 現在選択されているオーディオポート ● brightness: ブライトネスの現在値 ● contrast: コントラストの現在値 ● hue: ヒューの現在値 ● saturation: サチュレーションの現在値 ● sharpness: シャープネスの現在値 ● overscan: オーバースキャン (Off, Medium, Full) の現在値 ● standard: ビデオ入力のビデオスタンダード (NTSC, PAL, SECAM) ● wss: ワイドスクリーン信号。コンポジットビデオ信号の形式。 ● crop: 現在のクロッピング設定。Left (左)、top (上)、width (幅)、height (高さ) のパラメータが入力信号のフルサイズに対するパーセンテージで表されます。例えば、クロッピングがないフルサイズの場合、left:0, top:0, width:100, height:100, となります。また、左上 1/4 サイズの入力信号の場合、left:0, top:0, width:50, height:50 となります。クロッピングパラメータは、left:0.45, width:99.23 というように小数点第二位の正確さで変更することが可能です。 ● hdcp_enable: HDCP の有無

CONFIG_INPUT

指定した入力チャンネルのパラメータを設定します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみ利用できます。
要求メッセージ	<pre><config_input id="n" needack="{Yes No}"> <name>xxxx</name> <port>{DVI RGB SV ...}</port> <audio_port>{Embedded Analogue S/PDIF Mute}</audio_port> <brightness>n</brightness> <contrast>n</contrast> <hue>n</hue> <saturation>n</saturation> <sharpness>n</sharpness> <overscan>{Off Medium Full}</overscan> <standard>{NTSC PAL SECAM}</standard> <wss>{No Yes}</wss> <hdcp_enable>{Yes No}</hdcp_enable> <crop> <left>n</left> <top>n</top> <right>n</right> <bottom>n</bottom> </crop> </config_input></pre>
応答メッセージ	query_input_status の説明をご確認ください。
属性	id : 入力識別子
要素	input_status の説明をご確認ください。

QUERY_INPUT_INFO

指定した入力信号のタイミング情報を取得します。	
詳細	本機は query_input_info 要求メッセージを受信すると、 input_info 応答メッセージをクライアントに送信します。 入力信号が変更された場合にも、本機は input_info イベントメッセージをクライアントに送信します。
要求メッセージ	<pre><query_input_info id="n" /></pre>
応答メッセージ	<pre><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></pre>

	<pre><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> <sync_type>{SOG DCS DSS n/a}</sync_type></input_info></pre>
属性	<ul style="list-style-type: none"> ● id: 入力識別子 ● valid: 入力信号が適切なら Yes、入力信号が不適切なら No となります
要素	<ul style="list-style-type: none"> ● hres: 水平解像度 ● vres: 垂直解像度 ● hfreq: 水平周波数 ● vfreq: 垂直周波数 ● htotal: 水平合計 ● vtotal: 垂直合計 ● hstart: 水平スタート ● vstart: 垂直スタート ● hactive: 水平アクティブ ● vactive: 垂直アクティブ ● hpol: 水平ポラリティ ● vpol: 垂直ポラリティ ● interlaced: インターレース。値: Yes もしくは No ● hdcp: HDCP の有効 / 無効。値: Yes もしくは No ● hdcp_version: HDCP バージョン。値: n/a, 1.X, 2.0, 2.2 Type 0, 2.2 Type 1 ● sync_type: 同期信号の形式。値: SOG, DCS, DSS

QUERY_INPUT_CUSTOM_EDID

<p>指定した入力チャンネルのカスタム EDID 情報を取得します。</p>	
詳細	<p>本機は query_input_custom_edid 要求メッセージを受信すると、input_custom_edid 応答メッセージをクライアントに送信します。 入力の EDID が変更された場合、本機は input_custom_edid イベントメッセージをクライアントに送信します。</p>
要求メッセージ	<pre><query_input_custom_edid id="n" /></pre>
応答メッセージ	<pre><input_custom_edid id="n" valid="{No Yes}" [output_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> <hpol>{Positive Negative}</hpol> <vpol>{Positive Negative}</vpol> <multichannel_audio>{Yes No}</multichannel_audio> </input_custom_edid></pre>
属性	<ul style="list-style-type: none"> ● id: 入力識別子 ● valid: EDID が適切なら Yes、無効なら No となります

	<ul style="list-style-type: none"> ● output_id: カスタム EDID ブロックが特定の出力から来ている場合利用可能です。
要素	<ul style="list-style-type: none"> ● pixelclock: ピクセルクロック (MHz) ● interlaced: 利用可能な変数は Yes もしくは No ● hactive: 水平アクティブ ● vactive: 垂直アクティブ ● hblank: 水平ブランク ● vblank: 垂直ブランク ● hfrontporch: 水平フロントポーチ ● vfrontporch: 垂直フロントポーチ ● hsyncwidth: 水平同期幅 ● vsyncwidth: 垂直同期幅 ● hpol: 水平ポラリティ。値: Positive もしくは Negative ● vpol: 垂直ポラリティ。値: Positive もしくは Negative ● multichannel_audio: マルチチャンネルサウンドの対応。値: Yes もしくは No

SET_INPUT_CUSTOM_EDID

指定した入力チャンネルの EDID を設定します。	
詳細	<p>入力の EDID が変更された場合、本機は input_custom_edid イベントメッセージをクライアントに送信します。</p> <p>この要求メッセージは、適切な権限を持つクライアントのみが利用できます。</p>
要求メッセージ	<pre><set_input_custom_edid id="n" valid="{No Yes}" [output_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> <hpol>{Positive Negative}</hpol> <vpol>{Positive Negative}</vpol> <multichannel_audio>{Yes No}</multichannel_audio> </set_input_custom_edid></pre>
応答メッセージ	query_input_custom_edid メッセージの説明をご確認ください。
属性	<ul style="list-style-type: none"> ● id: 入力識別子 ● valid: Yes はカスタム EDID が有効、No は無効となります。 ● output_id: 特定の出力からカスタム EDID を取得する場合に入力します。
要素	input_custom_edid 応答メッセージの説明をご確認ください。

ADC_CALIBRATION

入力チャンネルの ADC (アナログ→デジタル変換) キャリブレーションを実行します。	
詳細	id 属性によって指定された入力チャンネルに対して、ADC キャリブレーションの手順を実行します。RGB、SV、CVBS、YUV の入力のみ実行可能です。この要求メッセージは、適切な権限を持つクライアントのみ使用できます。
要求メッセージ	<adc_calibration id="n" needack="{Yes No}"/>
応答メッセージ	-
属性	id: 入力識別子

AUTO_ADJUST

入力チャンネルの自動フェーズキャリブレーションを実行します。	
詳細	id 属性によって指定された入力チャンネルに対して、自動フェーズキャリブレーション手順を実行します。
要求メッセージ	<auto_adjust id="n" needack="{Yes No}"/>
応答メッセージ	-
属性	id: 入力識別子

REQUEST_INPUT_PREVIEW

入力チャンネルのプレビューセッションの開始 / 停止を実行します。	
詳細	id 属性によって指定された入力チャンネルに対して、入力信号のプレビューセッションの開始 / 停止を行います。この要求メッセージは、プレビュー機能が有効な場合のみ動作します。プレビュー機能の利用可否は、 device_info 応答メッセージで確認できます。
要求メッセージ	<request_input_preview id="n" type="{hls png rtsp udp}" width="n" height="n" needack="{Yes No}"> {Start Stop}</request_input_preview>
応答メッセージ	-
属性	<ul style="list-style-type: none"> ● id: 入力識別子 ● type: プレビューの形式。 <ul style="list-style-type: none"> ○ png: Motion PNG (fps: 5, delay minimal) ○ hls: http Live Stream ○ rtsp: real time streaming protocol ○ udp: ユーザーデータプロトコル (fps: 24, delay: 4-5 secs, codec: h264, maximum width 640 pixel) ● width, height: 要求されたプレビュー映像のサイズ。開始時に必要です。
備考	<p>プレビューのストリーム URL は以下の通りです。</p> <p>HLS : http://{server_ip_address}/hls/in{input_id}_preview.m3u8</p> <p>UDP : udp://224.0.1.1:1001 (1001 ポートは入力#1 を意味します)</p> <p>RTSP : rtsp://{server_ip_address}:8001/input (8001 ポートは input #1 のを意味します)</p> <p>ストリームの URL は、以下の応答メッセージでも確認できます。</p> <p><input_preview url="udp://224.0.1.1:1006" /></p>

GET_INPUT_PREVIEW (PNG 形式のみ)

id 属性で指定された入力チャンネルのプレビュー画像を取得します。	
詳細	この要求メッセージは、 request_input_preview の開始 (Start) 要求が成功した後に送信できます。 本機は get_input_preview 要求メッセージを受信すると、 input_preview 応答メッセージをクライアントに送信します。この応答メッセージには、 id 属性で指定された入力チャンネルのプレビュー画像が含まれています。
要求メッセージ	<code><get_input_preview id="n" /></code>
応答メッセージ	<code><input_preview id="n" valid="{Yes No}"> <![CDATA[...]]> </input_preview></code>
属性	<ul style="list-style-type: none"> ● id: 入力識別子 ● valid: Yes であれば有効となり、No であれば無効となります。
備考	cdata セクションには、png フォーマットのプレビュー画像が格納されています。

2.3 出力

QUERY_OUTPUT_STATUS

出力チャンネルの HDCP ステータスを取得します。	
詳細	本機は query_output_status 要求メッセージを受信すると、 output_status 応答メッセージをクライアントに送信します。 1 つまたは複数の出力チャンネルの HDCP ステータスが変更された場合、 output_status イベントメッセージをクライアントに送信します。
要求メッセージ	<query_output_status />
応答メッセージ	<output_status> <hdcp id="n" version="n">{Yes No}</hdcp> ... </output_status>
属性	id : 出力識別子 version : HDCP バージョン
要素	hdcp : HDCP が有効の場合は Yes 、HDCP が有効でない場合は No

QUERY_OUTPUT_CONFIG

出力チャンネルの設定情報を取得します。	
詳細	本機は query_output_config と config_output 要求メッセージを受信すると、 output_config 応答メッセージをクライアントに送信します。 1 つまたは複数の出力パラメータが変更された場合、 output_config イベントメッセージをクライアントに送信します。
要求メッセージ	<query_output_config />
応答メッセージ	<output_config type="{Grid Individual}"> <!-- IF TYPE = GRID --> <horizontal_grid>n</horizontal_grid> <vertical_grid>n</vertical_grid> <horizontal_gap>n</horizontal_gap> <vertical_gap>n</vertical_gap> <background_color>0xhhhhhh</background_color> <resolution>n</resolution> <output_mode>{Auto Force DVI Force HDMI}</output_mode> <hdcp_enable>{Yes No}</hdcp_enable> <!-- IF TYPE = INDIVIDUAL --> <display id="n"> <left>n</left> <top>n</top> <background_color>0xhhhhhh</background_color> <resolution>n</resolution> <output_mode>{Auto Force DVI Force HDMI}</output_mode> <hdcp_enable>{Yes No}</hdcp_enable> </display> ... </output_config>

属性	type: グリッドサイズもしくは個別
要素	<p>Type がグリッドの場合</p> <ul style="list-style-type: none"> ● horizontal_grid / vertical_grid: ビデオウォールを構成するモニタの行と列の数 ● horizontal_gap / vertical_gap: ベゼルもしくはオーバーラップの設定情報を取得できます。 <p>Type が個別設定の場合</p> <ul style="list-style-type: none"> ● left / top: ビデオウォール上の表示器の位置。それぞれの表示器を個別に表します。 <p>共通の表示器パラメータ</p> <ul style="list-style-type: none"> ● background_colour: バックグラウンドカラー ● resolution: 解像度の識別子。識別子は device_info 応答メッセージの output_modes で確認できます。 ● output_mode: 利用可能な変数は Auto、Force DVI、Force HDMI ● hdcp_enable: 利用可能な変数は Yes もしくは No

CONFIG_OUTPUT

指定した出力チャンネルの設定情報を変更します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<pre><config_output type="{Grid Individual}"> <!-- IF TYPE = GRID --> <horizontal_grid>n</horizontal_grid> <vertical_grid>n</vertical_grid> <horizontal_gap>n</horizontal_gap> <vertical_gap>n</vertical_gap> <background_colour>0xhhhhhh</background_colour> <resolution>n</resolution> <output_mode>{Auto Force DVI Force HDMI}</output_mode> <hdcp_enable>{Yes No}</hdcp_enable> <!-- IF TYPE = INDIVIDUAL --> <display id="n"> <left>n</left> <top>n</top> <background_colour>0xhhhhhh</background_colour> <resolution>n</resolution> <output_mode>{Auto Force DVI Force HDMI}</output_mode> <hdcp_enable>{Yes No}</hdcp_enable> </display> ... </config_output></pre>
応答メッセージ	詳細は query_output_config メッセージをご確認ください。
属性	type: グリッドもしくは個別設定
要素	Type がグリッドの場合

	<ul style="list-style-type: none"> ● horizontal_grid / vertical_grid: ビデオウォールを構成するモニタの行と列の数 ● horizontal_gap / vertical_gap: 本パラメータにより、ベゼルもしくはオーバーラップの設定情報を設定できます。 <p>Type が個別設定の場合</p> <ul style="list-style-type: none"> ● left / top: ビデオウォール上の表示器の場所となります。それぞれの表示器を個別に表します。mm もしくは inch により小数点第二位まで表示します。 <p>共通の表示器パラメータ</p> <ul style="list-style-type: none"> ● background_colour: バックグラウンドカラー ● resolution: 解像度の識別子。識別子は device_info 応答メッセージリスト上の output_modes で確認できます。 ● output_mode: 利用可能な変数は Auto、Force DVI、Force HDMI ● hdcp_enable: 利用可能な変数は Yes もしくは No
--	---

QUERY_OUTPUT_TIMING_CONSTRAINTS

指定した出力チャンネルの制限情報を取得します。	
詳細	本機は query_output_timing_constraints 要求メッセージを受信すると、 output_timing_constraints 応答メッセージをクライアントに送信します。このメッセージには、カスタム出カタイミング設定に必要な出力チャンネルの制限情報を取得できます。
要求メッセージ	<code><query_output_timing_constraints id="n" /></code>
応答メッセージ	<pre> <output_timing_constraints id="n" > <pixelclock min="n" max="n" /> <hactive min="n" max="n" mask="0xhhhhhhhh" /> <vactive min="n" max="n" mask="0xhhhhhhhh" /> <hblank min="n" max="n" mask="0xhhhhhhhh" /> <vblank min="n" max="n" mask="0xhhhhhhhh" /> <hfrontporch min="n" max="n" mask="0xhhhhhhhh" /> <vfrontporch min="n" max="n" mask="0xhhhhhhhh" /> <hsyncwidth min="n" max="n" mask="0xhhhhhhhh" /> <vsyncwidth min="n" max="n" mask="0xhhhhhhhh" /> <interlaced /> <DSS /> <CSYNC /> <SOG /> </output_timing_constraints> </pre>
属性	id : 出力識別子
要素	<ul style="list-style-type: none"> ● pixelclock min, max: ピクセル周波数制限 ● hactive min, max, mask: 水平稼働制限及びマスク ● vactive min, max, mask: 垂直アクティブ制限及びマスク ● hblank min, max, mask: 水平ブランク制限及びマスク ● vblank min, max, mask: 垂直ブランク制限及びマスク ● hfrontporch min, max, mask: 水平フロントポーチ制限及びマスク ● vfrontporch min, max, mask: 垂直フロントポーチ制限及びマスク

	<ul style="list-style-type: none"> ● hsyncwidth min, max, mask: 水平同期幅制限及びマスク ● vsyncwidth min, max, mask: 垂直同期幅制限及びマスク ● interlaced: 出力のインターレースタイミング処理 ● DSS: 出力のセパレートシンクフォーマット処理 ● CSYNC: 出力のコンポジットシンクフォーマット処理 ● SOG: 出力のシンクオングリーンフォーマット処理
--	---

QUERY_CUSTOM_OUTPUT_TIMINGS

カスタム出力タイミング設定の情報を取得します。	
詳細	本機は query_custom_output_timings 要求メッセージを受信すると、 custom_output_timings 応答メッセージをクライアントに送信します。このメッセージには、追加されたすべてのカスタム出力タイミングの情報が含まれていません。
要求メッセージ	<query_custom_output_timings/>
応答メッセージ	<pre> <custom_output_timings> timing 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> </timing> ... </custom_output_timings> </pre>
属性	id : カスタム出力タイミング識別子
要素	<ul style="list-style-type: none"> ● pixelclock: ピクセル周波数 ● interlaced: インターレース。値 : Yes, No ● hactive: 水平アクティブ ● vactive: 垂直アクティブ ● hblank: 水平ブランク ● vblank: 垂直ブランク ● hfrontporch: 水平フロントポーチ ● vfrontporch: 垂直フロントポーチ ● hsyncwidth: 水平同期幅 ● vsyncwidth: 垂直同期幅 ● hpol: 水平ポラリティ ● vpol: 垂直ポラリティ ● synctype: 同期タイプ。値 : DSS (セパレートシンク)、CSYNC (コンポジットシンク)、SOG (シンクオングリーン)

ADD_CUSTOM_OUTPUT_TIMING

新しいカスタム出力タイミングを作成し、出力タイミングリストに追加します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<pre><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></pre>
応答メッセージ	<custom_output_timing_changed />
要素	query_custom_output_timings の内容をご確認ください。

MODIFY_CUSTOM_OUTPUT_TIMING

指定したカスタム出力タイミングを修正します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<pre><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></pre>
応答メッセージ	<custom_output_timing_changed />
属性	id : カスタム出力タイミング識別子
要素	query_custom_output_timings の内容をご確認ください。

DELETE_CUSTOM_OUTPUT_TIMING

指定したカスタム出力タイミングをリストから削除します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<code><delete_custom_output_timing id="n" needack="{Yes No}" /></code>
応答メッセージ	<code><custom_output_timing_changed /></code>
属性	id: カスタム出力タイミング識別子

QUERY_OUTPUT_TIMING

指定した出力タイミングの情報を取得します。	
詳細	本機は query_output_timing 要求メッセージを受信すると、 output_timing 応答メッセージをクライアントに送信します。この応答メッセージには、指定した出力タイミングのすべての情報が格納されています。
要求メッセージ	<code><query_output_timing id="n" mode_id="n" /></code>
応答メッセージ	<pre> <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_timings> </pre>
属性	<ul style="list-style-type: none"> ● id: 出力識別子 ● mode_id: 出力タイミング識別子
要素	<ul style="list-style-type: none"> ● pixelclock: ピクセル周波数 ● interlaced: インターレース。値 : Yes, No ● hactive: 水平アクティブ ● vactive: 垂直アクティブ ● hblank: 水平ブランク ● vblank: 垂直ブランク ● hfrontporch: 水平フロントポーチ ● vfrontporch: 垂直フロントポーチ ● hsyncwidth: 水平同期幅 ● vsyncwidth: 垂直同期幅 ● hpol: 水平ポラリティ ● vpol: 垂直ポラリティ

2.4 レイアウト

QUERY_LAYOUT_STATUS

レイアウトパラメータの設定情報を取得します。	
詳細	本機は query_layout_status 要求メッセージを受信すると、 layout_status 応答メッセージをクライアントに送信します。 1つ以上のレイアウトパラメータが変更された場合、本機は、 layout_status イベントメッセージをクライアントに送信します。
要求メッセージ	<query_layout_status />
応答メッセージ	<layout_status running="{No Yes}"> <layout id="n" name="xxx" timeout="n" advance="{Off Next n}" color="0xhhhhhh" active="{No Yes}">{Not Used Used}</layout> ... </layout_status>
属性	<ul style="list-style-type: none"> ● id: レイアウト識別子 ● name: レイアウト名 ● active: レイアウトがアクティブか非アクティブかを示します。 ● running: レイアウトの自動実行が現在実行されているか否かを示します。
値	<ul style="list-style-type: none"> ● Not used: id で識別されるレイアウトが利用されていないことを示します。 ● Used: レイアウトの設定が保存されています。

MODIFY_LAYOUT_PARAMETERS

レイアウトのパラメータを設定します。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<modify_layout_parameters id="n" timeout="n" advance="{Off Next n}" name="xxx" color="0xhhhhhh" needack="{Yes No}"/>
応答メッセージ	-
属性	<ul style="list-style-type: none"> ● id: レイアウト識別子 ● name: レイアウト名 ● timeout: シナリオ実行時における二つのレイアウト間の遅延 ● advance: シナリオ実行時における次のステップ ● color: レイアウトボタンの色

SAVE_LAYOUT

現在のレイアウトを保存します。	
要求メッセージ	<save_layout id="n" timeout="n" advance="{Off Next n}" name="xxx" color="0xhhhhhh" needack="{Yes No}" name="x"/>
応答メッセージ	-
属性	<ul style="list-style-type: none"> ● id: レイアウト識別子 ● name: レイアウト名 ● timeout: シナリオ実行時における二つのレイアウト間の遅延 ● advance: シナリオ実行時における次のステップ ● color: レイアウトボタンの色

CLEAR_LAYOUT

レイアウトに保存されている情報を削除します。	
要求メッセージ	<clear_layout id="n" needack="{Yes No}"/>
応答メッセージ	-
属性	id: レイアウト識別子

SWAP_LAYOUT

from と to で指定された 2 つのレイアウト間のパラメータを変更します。	
要求メッセージ	<swap_layout from="n" to="n" needack="{Yes No}"/>
応答メッセージ	-
属性	<ul style="list-style-type: none"> ● from: レイアウト識別子 ● to: レイアウト識別子

RECALL_LAYOUT

id で指定された保存済みレイアウトを呼び出します。	
要求メッセージ	<recall_layout id="n" advance="{Yes No}" needack="{Yes No}"/>
応答メッセージ	-
属性	<ul style="list-style-type: none"> ● id: レイアウト識別子 ● advance: レイアウト呼び出し時のシナリオ自動実行

SAVE_LAYOUTS

全レイアウトをクライアントにバックアップします。	
詳細	本機は save_layouts 要求メッセージを受信すると、 save_layouts 応答メッセージをクライアントに送信します。 この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<save_layouts/>
応答メッセージ	<save_layouts> ... </save_layouts>

LOAD_LAYOUTS

save_layouts 要求メッセージでバックアップした全レイアウト設定を本機に読み込みます。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<load_layouts needack="{Yes No}" > ... </load_layouts>
応答メッセージ	-

2.5 ビデオウィンドウステータス

QUERY_WINDOWS_STATUS

ウィンドウ情報を取得します。	
詳細	<p>本機は、query_windows_status 要求メッセージを受信すると、windows_status 応答メッセージをクライアントに送信します。</p> <p>ウィンドウの設定が変更された場合、本機は windows_status イベントメッセージをクライアントに送信します。</p>
要求メッセージ	<query_windows_status />
応答メッセージ	<pre> <windows_status> <window id="n" input_id="{n Close}" zorder="n"> <keepaspectratio keep_dimension="{Width Height}" keep_position="{Center Start End}">{No Yes}</keepaspectratio> <placement> <left>n</left> <top>n</top> <width>n</width> <height>n</height> </placement> <crop> <left>n</left> <top>n</top> <width>n</width> <height>n</height> </crop> <osd left="n" top="n" background="{No Yes}" bgcolor="0xhhhhhh" line1color="0xhhhhhh" line2color="0xhhhhhh" line1="xxx" line2="xxx">{Yes No}</osd> <frame left="n" top="n" right="n" bottom="n" color ="0xhhhhhh">{No Yes}</frame> <transition_eff duration="n">{Cut Fade}</transition_eff> <open_eff duration="n">{Cut Fade}</open_eff> <close_eff duration="n">{Cut Fade}</close_eff> </window> ... </windows_status> </pre>
属性	<ul style="list-style-type: none"> ● id: ウィンドウ識別子 ● input_id: 開いているウィンドウ識別子の入力チャンネル ● zorder: 複数の重複ウィンドウの配置を示します。
要素	<ul style="list-style-type: none"> ● keepaspectratio: 入力信号のアスペクト比の保持 ● placement: ウィンドウ配置パラメータ ● frame: ウィンドウフレームパラメータ ● osd: ウィンドウ OSD パラメータ ● crop: ウィンドウトリミングパラメータ ● transition_eff: ウィンドウのソース切替効果 ● open_eff: ウィンドウオープン効果 ● close_eff: ウィンドウクローズ効果

TAKE

ビデオウィンドウの展開、またはウィンドウの修正を行います。	
詳細	ウィンドウのステータスが変更されると、本機は windows_status イベントメッセージをクライアントに送信します。
要求メッセージ	<pre> <take needack="{Yes No}"> <window id="n" input_id="{n Close}" zorder="n"> <keepaspectratio keep_dimension="{Width Height}" keep_position="{Center Start End}">{No Yes}</keepaspectratio> <placement> <left>n</left> <top>n</top> <width>n</width> <height>n</height> </placement> <crop> <left>n</left> <top>n</top> <width>n</width> <height>n</height> </crop> <osd left="n" top="n" background="{No Yes}" bgcolor="0xhhhhh" line1color="0xhhhhh" line2color="0xhhhhh" line1="xxx" line2="xxx">{Yes No}</osd> <frame left="n" top="n" right="n" bottom="n" color ="0xhhhhh">{No Yes}</frame> <transition_eff duration="n">{Cut Fade}</transition_eff> <open_eff duration="n">{Cut Fade}</open_eff> <close_eff duration="n">{Cut Fade}</close_eff> </window> ... </take> </pre>
応答メッセージ	詳細は、query_windows_status メッセージをご参照ください。
属性	<ul style="list-style-type: none"> ● id: ウィンドウ識別子 ● input_id: 開いているウィンドウ識別子の入力チャンネル ● zorder: 複数の重複ウィンドウの配置を示します。
要素	query_windows_status メッセージの説明をご確認ください。

TAKE_CLOSE_ALL

ウォールに表示しているすべてのウィンドウを閉じます。	
詳細	ウィンドウのステータスが変更されると、本機は windows_status イベントメッセージをクライアントに送信します。
要求メッセージ	<code><take_close_all /></code>
応答メッセージ	-

2.6 音声割り当て

QUERY_AUDIO_ASSIGNMENT

音声の割り当てに関する情報を取得します。	
詳細	本機は query_audio_assignment 要求メッセージを受信すると、 audio 応答メッセージをクライアントに送信します。 音声割り当てが変更された場合、本機は audio イベントメッセージをクライアントに送信します。
要求メッセージ	<query_audio_assignment />
応答メッセージ	<audio> <connected input_id="n" output_id="n" /> ... <disconnected output_id="n" /> ... </audio>
属性	-
要素	<ul style="list-style-type: none"> ● input_id: 入力識別子 ● output_id: 出力識別子

AUDIO

音声割り当てを設定し、新しい接続の作成、既存の接続の削除を実行します。	
要求メッセージ	<audio needack="{Yes No}"> <connect input_id="n" output_id="n" /> ... <disconnect output_id="n" /> ... </audio>
応答メッセージ	詳細は、 query_audio_assignment メッセージをご参照ください。

2.7 その他

QUERY_HARDWARE_MONITOR_STATUS

ハードウェアモニタの情報を取得します。	
詳細	本機は query_hardware_monitor_status 要求メッセージを受信すると、 hardware_monitor_status 応答メッセージをクライアントに送信します。
要求メッセージ	<query_hardware_monitor_status />
応答メッセージ	<pre><hardware_monitor_status> <temperature source="xxx" min="n" max="n" unit="Celsius">n</temperature> ... <tachometer source="xxx" min="n">n</tachometer> ... <voltage source="xxx" min="n" max="n" Nominal="n">n</voltage> ... <power source="xxx">{Ok Failure}</power> ... </hardware_monitor_status></pre>
要素	<ul style="list-style-type: none"> ● temperature: 温度センサが感知した現在の温度。センサは source 属性により識別されます。有効な値は min および max 属性により指定され、測定値の単位は unit で指定されます。 ● tachometer: 内部ファンから受けた現在の測定速度。ファンは source 属性により識別され、最低速度は min 属性によって指定されます。 ● voltage: 電源電圧の実測値。センサは source 属性により識別され、min、max、および Nominal の値で指定されます。 ● power: 電源供給のステータス。電源は source 属性により識別されます。

STORE_CUSTOM_FIELD

クライアント固有の情報を本機に保存します。	
詳細	この欄のデータ形式は指定されないため、どの文字列も使用可能です。
要求メッセージ	<pre><store_custom_field id="n" needack="{Yes No}"> ... </store_custom_field></pre>
応答メッセージ	-
属性	id: データを識別する一意の英数字の文字列

QUERY_CUSTOM_FIELD

本機に保存済みのクライアント固有の情報を取得します。	
詳細	本機は query_custom_field 要求メッセージを受信すると、 custom_field reply メッセージをクライアントに送信します。カスタムフィールドデータの値は、クライアントから供給されたものと同じ形式で応答します。
要求メッセージ	<query_custom_field id="n" />

応答メッセージ	<custom_field id="n"> ... </custom_field>
属性	id: 保存済みデータを識別する英数字の文字列

ECHO

本機との通信の確認に使用できます。	
詳細	本機は echo 要求メッセージを受信すると、 echo 応答メッセージで message_content の内容をクライアントに返送します。
要求メッセージ	<echo>message_content</echo>
応答メッセージ	<echo>message_content</echo>

REBOOT

本機を再起動させます。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<reboot needack="{Yes No}" />
応答メッセージ	-

SHUTDOWN

本機をシャットダウンさせます。	
詳細	この要求メッセージは、適切な権限を持つクライアントのみが利用できます。
要求メッセージ	<shutdown needack="{Yes No}" />
応答メッセージ	-

DVP500x シリーズ 取扱説明書

<コマンドガイド>

Ver.1.0.0

発行日 : 2021 年 9 月 7 日



株式会社アルバニクス

本 社 〒242-0021 神奈川県大和市中心 7-9-1
TEL: (046) 259-6920
FAX: (046) 259-6930
E-mail: info@arvanics.com
URL: <http://www.arvanics.com>