# PICOLO U4 H.264™
# PICOLO U8 H.264™
# PICOLO U16 H.264™

## Documentation

Download the
PICOLO U4/U8/U16 H.264
driver from
www.euresys.com

PCI EXPRESS™

EURESYS™
Excellence in vision

## Disclaimer



## Contact Us

### Europe

**Euresys s.a.** (Corporate Headquarters)

Avenue du Pré-Aily, 14
B-4031 Angleur
BELGIUM

Phone: (32) (4) 367-72-88
Fax: (32) (4) 367-74-66

### United States

**Euresys Inc.**

27126B Paseo Espada, Suite 704
San Juan Capistrano, CA 92675
USA

Toll free: (1) (866) 387-3797
Phone: (1) (630) 250-2300 EXT 2303
Fax: (1) (630) 626-1619

### Asia

**Euresys Pte. Ltd.**

627A Aljunied Road
#08-09 BizTech Centre
Singapore 389842
SINGAPORE

Phone: (65) 6748-0085
Fax: (65) 6841-2137

# Contents

# Installation

# 1. Precautions of Use

## General Precautions

- **Electrostatic Sensitive Device.** Boards may be damaged by electrostatic discharges. Follow the procedure hereby described and apply any general procedure aimed at reducing the risk associated to electrostatic discharge. Damage caused by improper handling is not covered by the manufacturer warranty.
- **Electromagnetic Compatibility.** Euresys boards are compliant with electromagnetic compatibility regulatory requirements. To ensure this compliance, it is mandatory to secure the board bracket with the relevant screw according to the procedure hereby described.
- **Risk of Electrical Shock.** Do not operate the computer with any enclosure cover removed. During the hardware installation, ensure the AC power cord is unplugged before touching any internal part of the computer.
- **Heating Device.** In operation, it is normal that a board dissipates some heat. To ensure the adequate cooling effect of the fan equipping your computer, it is mandatory to correctly fit all enclosure covers, including blank brackets.
- **Hot Plugging Forbidden.** Uncontrolled plugging and unplugging of equipment may damage a board. Always switch-off the computer, the cameras and any relevant system device when connecting or disconnecting a cable at the frame grabber or auxiliary board bracket.
- **Poor Grounding Protection.** The computer and the camera can be located in distant areas with distinct ground connections. Poor ground interconnection, ground loop or ground fault may induce unwanted voltage between equipments, causing excessive current in the interconnecting cables. This faulty situation can damage the frame grabber or the camera electrical interface. The user must follow proper equipment grounding practices at all ends of the interconnecting cables. In addition, it is recommended to use cable assemblies with overall shield solidly connected to the conductive shell of all connectors. Besides the beneficial effect of cable shielding on electromagnetic compatibility, the shield connection can increase the protection level against grounding problems in temporarily absorbing unwanted fault current.

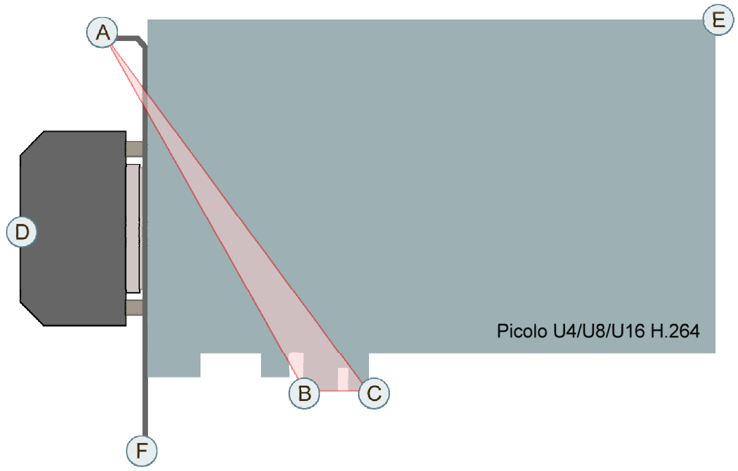
## Specific Recommendations

- Don't warp or bend the PCB assembly excessively.
- Don't apply excessive torque or force on the connectors.
- Don't apply excessive shocks on the assembly.
- Ensure that the PCI connector on the PC mother board, and chassis slot are properly aligned according to PCI Express specification.
- Ensure locking of the bracket screw.
- Use chassis limiting the displacement of the bracket tip (point F in drawing below).
- Use chassis with additional board retainers; however, take care that additional board retainers are properly aligned and don't stress mechanically the PCB assembly.

Following limits shall never be exceeded:

- A torque applied around AC axis (see drawing below) on point E of the board may not result into a lateral displacement exceeding +/- 2 mm.

⚠ The maximal torque applied around AB axis (see drawing below) on the VIDEO connector is 0.5 Newton-meter. For instance, this corresponds to a force of 10 Newton applied at point D along an axis perpendicular to the ACB seating plane.



# 2. Supported Platforms

For the supported CPU architectures, operating systems and development tools, please refer to the Release Notes that come with your driver on the Euresys website.

# 3. PCI Board Installation

Euresys PCI frame grabbers —Conventional PCI or PCI Express— must be physically inserted in your computer. Before operation, check if your computer has an available slot. Multiple Euresys boards can be inserted in the same computer.

### Recommended Installation Procedure

- Switch off the computer and all connected peripherals (monitor, printer...).
- Discharge any static electricity that could be accumulated by your body. You can achieve this by touching an unpainted metal part of the enclosure of your computer with a bare hand. Make sure that the computer is linked to the AC power outlet with proper earth connection.
- Disconnect all cables from your computer, including AC power.
- Open the computer enclosure, according to the manufacturer instructions, to gain access to the PCI slots. Locate an available PCI slot —Conventional PCI or PCI Express, according to your board.
- Remove the blank bracket associated with this location. To achieve this, remove the securing screw and keep it aside for later use in the procedure. Keep the blank bracket in a known place for possible re-use.
- Unwrap the Euresys board packing, take the board and carefully hold it. Avoid any contact of the board with unnecessary items, including your clothes.
- Gently insert the board in the selected PCI slot, taking care to push it down fully into the slot. If you experience some resistance, remove the board and repeat the operation. You should attempt to make a perfect board-to-slot mechanical alignment for best results. Ensure that the lower part of the bracket is inserted into the corresponding enclosure fastening.
- Secure the board with the saved screw.
- Install the modules and auxiliary board, if any. Refer to the module or board documentation for configuration and connection requirements.
- Close the computer enclosure according to the manufacturer instructions.
- Establish the camera connection.

### Board Cooling

- To guarantee proper operation and longer board reliability, ensure an adequate cooling of the board. The cooling is improved by a higher air flow circulating around the board. This air flow is increased, for example, by using computer case fans.
- In addition, avoid placing a frame grabber next to other heat dissipating boards.

# 4. Driver Installation

The DirectShow driver is distributed on the Euresys website download area: www.euresys.com > DOWNLOAD. The first time access requires a profile creation to obtain a user ID and a password.

Once the package is downloaded, install the driver. Note that if you have an existing Picolo U4/U8/U16 H.264 driver already installed, you will be prompted to uninstall it before being able to continue. The following lists the files (**C:\Program Files\Euresys\UxH264** directory by default):

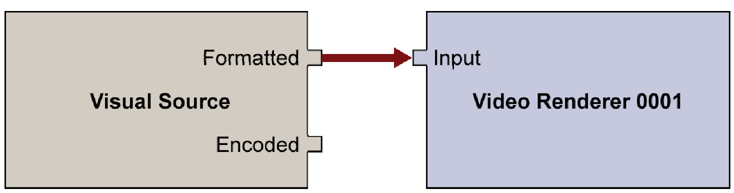
- **C:\Program Files\Euresys\UxH264** contains script files for driver installation.
- **C:\Program Files\Euresys\UxH264\DriversDS** contains the driver.
- **C:\Program Files\Euresys\UxH264\Include** contains the header file (**Picolo_UxH264.h**).

# 5. Testing Your Board

You can use the GraphEdit application to test the working condition of the board. For more information on the DirectShow filters, refer to DirectShow Reference.

## Formatted Video Testing

- Open GraphEdit.
- Select Graph > Insert Filters.
- Select WDM Streaming Capture Devices > UxH264 Visual Source.
- Click [Insert Filter]. The filter is displayed with two output pins named "Formatted" and "Encoded" respectively.
- Select Graph > Insert Filters.
- Select DirectShow Filters > Video Renderer.
- Click the "Formatted" pin, and drag the arrow to the "Input" pin of the Video Renderer.
- Right-click the UxH264 Visual Source Filter, and select [Filter Properties].
- Select [UxH264 Video Source] tab.
- Select the Input Channel that your video source is currently connected to, and click [OK].
- Select [UxH264 Video Formatter] tab.
- Select the [Flip Vertical] checkbox. Take note, selecting this checkbox depends on the type of video renderer that is used.
- Click [Play]. The video should appear. If not, test with all the input channels.

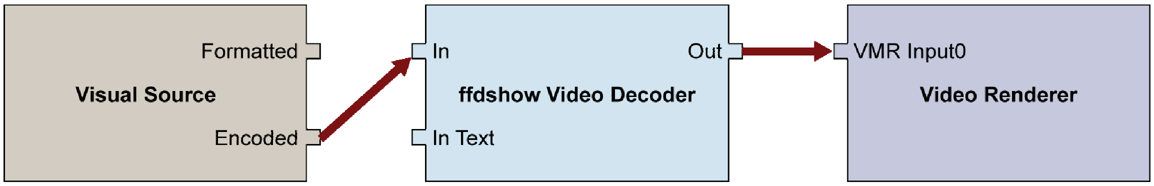



Formatted video testing

## Encoded Video Testing

- Repeat steps 2 to 4 of the formatted video testing.
- Insert the H.264 decoder filter, for example, **ffdshow** Video Decoder. If you do not have the decoder filter, download any other DirectShow H.264 decoder.
- Click the "Encoded" pin, and drag the arrow to the "In" pin of the **ffdshow** Video Decoder.
- Right-click the "Out" pin, and select [Render Pin].
- Right-click the UxH264 Visual Source filter, and select [Filter Properties].
- Select [UxH264 Video Source] tab.
- Select the Input Channel that your video source is currently connected to, and click [OK].

- Click [Play]. The encoded video should appear. If not, test with all the input channels.




Encoded video testing

**Hint.** Once you built the graphs, select File > Save As to save the graphs, so you don't need to rebuild them every time.

# Hardware Reference

# 1. PICOLO U4/U8/U16 H.264 Overview

The Picolo U4 H.264, Picolo U8 H.264 and Picolo U16 H.264 are outstanding video capture cards featuring real-time H.264 on-board compression for 4, 8 or 16 video channels along with audio capability. Each video input can be configured independently. These cards offer cost effective acquisition and compression delivering three independently configurable output streams for each video input: two H.264 encoded streams and one raw uncompressed stream.

Picolo U4/U8/U16 H.264 —1-lane PCI Express— offers 180 MByte/s data delivery bandwidth. The following lists the main features of the board.

### System

- Interface
  - Full height half length, 1-lane PCI Express board
- Connectors
  - 1 VIDEO I/O HD44F 44 pins High density Sub-D connector
  - 1 VIDEO OUTPUT & CASCADE PH6M 6 pins header
  - 1 AUDIO FTSH34M 34 pins High density header
  - 2 PROFESSIONAL I/O FTSH34M 34 pins High density headers
  - 1 WATCHDOG PH4M 4-pins header
  - 1 PCI Express 1-lane connector
  - 1 VIDEO FTSH34M 34 pins High-density header
- Temperature monitor

### Video Capture

- Video decoder — analog video decoding
  - Multi-standard PAL-B/D/G/H/I, or NTSC-M
  - 4/8/16 independent video acquisition channels
  - Independently programmable frame rate and acquisition parameters for each video acquisition channels
    - Image size: 4CIF, 2CIF, CIF and QCIF
    - Contrast, brightness and saturation controls available
    - Video presence and video standard detection, overlay caption text, privacy masking regions

### On-Board Compression

- H.264 (MPEG-4 Part 10) Baseline Profile (Level 3)
- 3 simultaneous and independently configurable output streams for each video channel
  - Two independently configurable H.264 encoded streams
    - Resolution settings: 4CIF, 2CIF, CIF, QCIF
    - Configurable reduction of the frame rate
    - Configurable bit rate control: CPQ, CBR, VBR

- One raw uncompressed stream - subject to PCI bus available bandwidth
  - Resolution settings: 4CIF, 2CIF, CIF, QCIF
  - Configurable reduction of the frame rate
- Image storage formats available:
  - YUV420PL (native format), Y8
  - Other formats possible: YUV422PL, YUV422, RGB15, RGB16, RGB24 and RGB32

- Data transfer
  - High-performance DMA transfer
  - Scattered transfer

### Audio Capture

- 4/8/16 independent and high-quality audio acquisition channels
- Line-level analog audio input signals
- Selectable audio formats:
  - 64 kbps G.711 A-law, 64 kbps G.711 μ-law
  - Linear PCM @ 8, 16, 22.05, 44.1, or 48 kHz
- Audio-video synchronization supported by accurate time stamping of audio and video data

### Video Pass-Through Selector

Any of 4/8/16 video inputs, or the cascade video input, can be routed to video output.

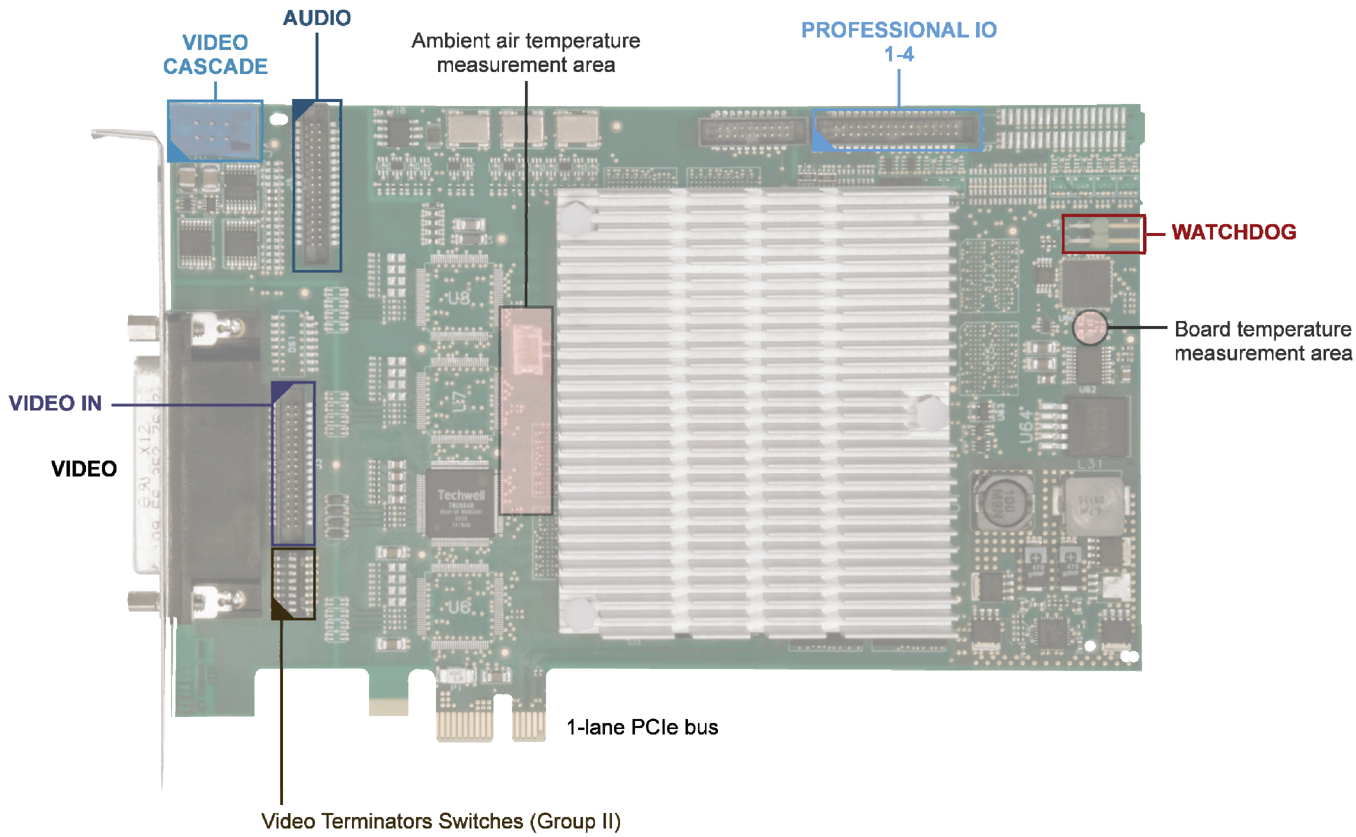
### IO Sub-System

- 4/8/16 solid-state relay outputs
- 4/8/16 contact-closure inputs
- 1 Watchdog output

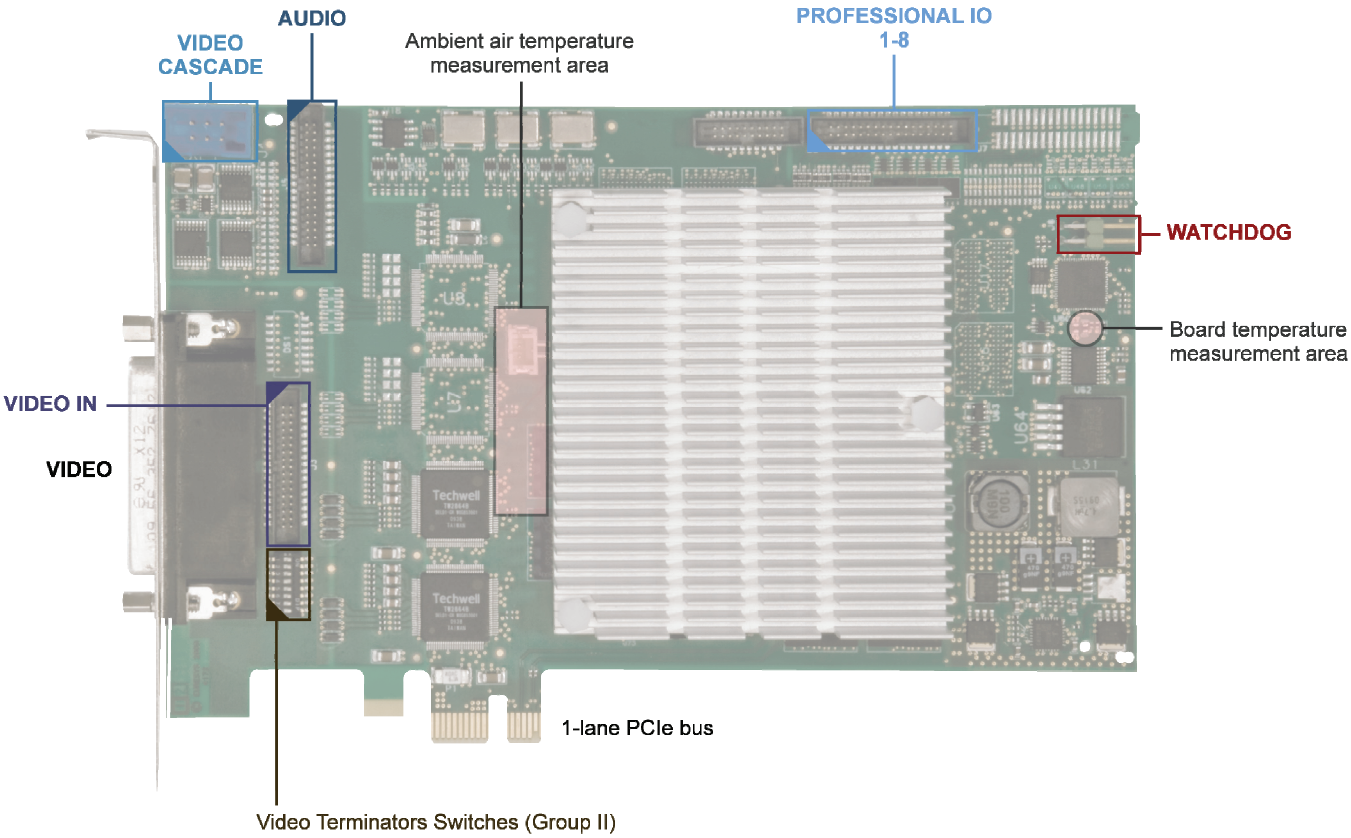
# 2. Board Specifications

## 2.1 Board Layout and Block Diagram

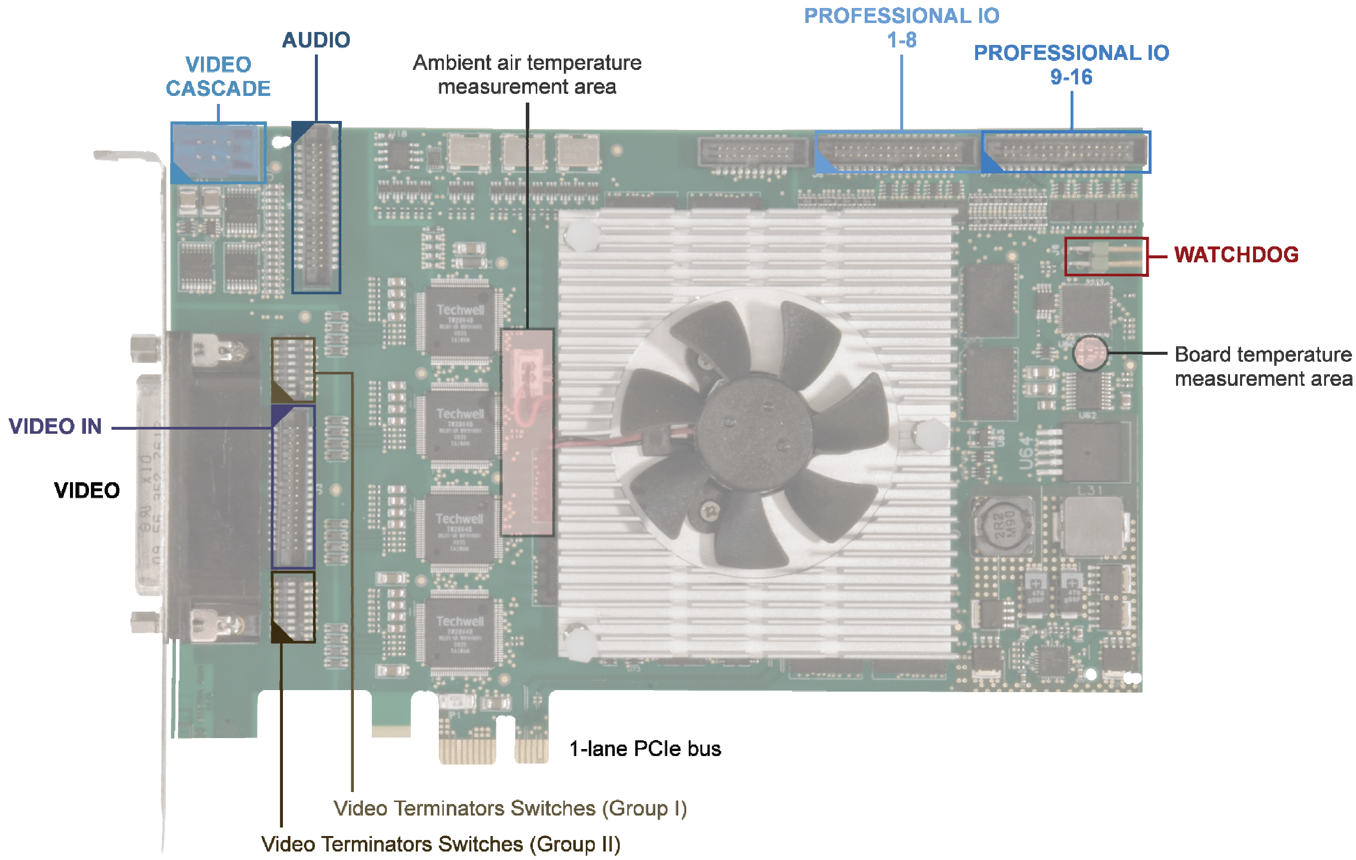
| | |
|---|---|
| **Dimensions** | 168 mm x 111 mm |
| **PCIe connector** | 1-lane |
| **Video connections** | Up to 4/8/16 cameras |



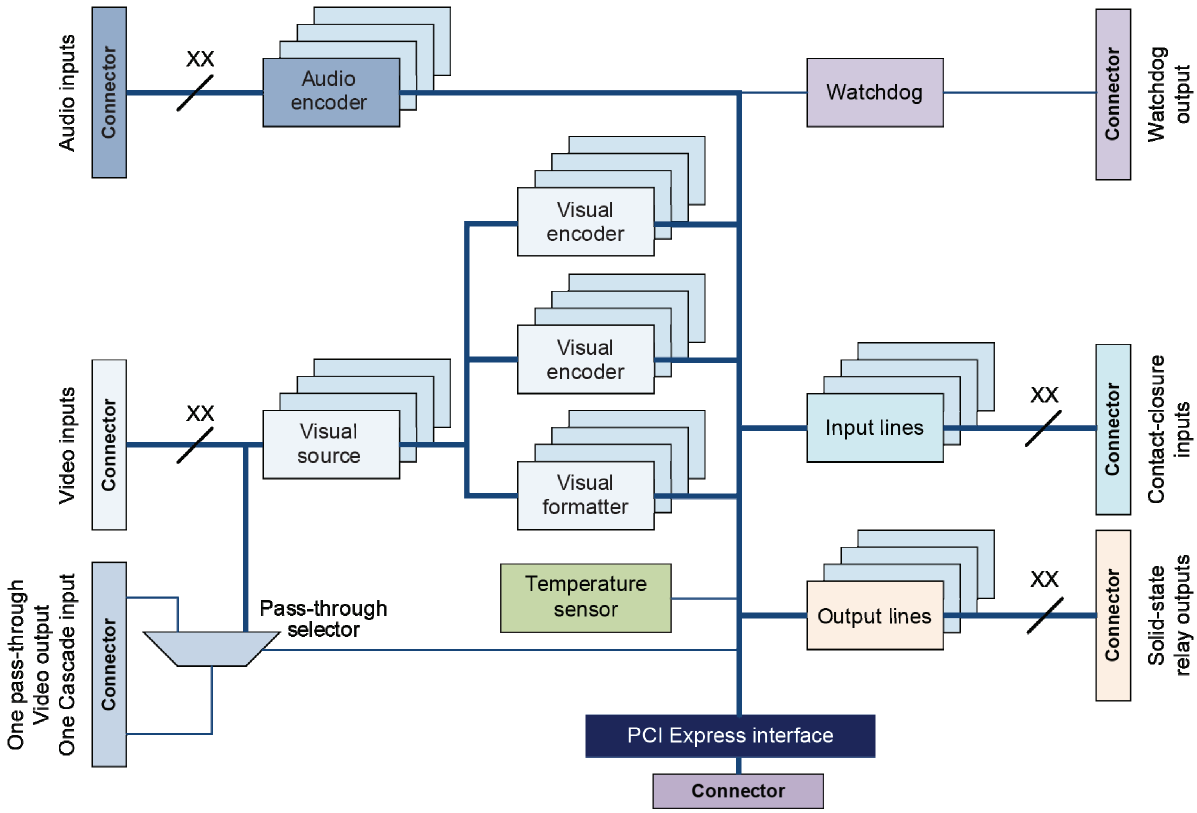

Picolo U4 H.264 board layout




Picolo U8 H.264 board layout




Picolo U16 H.264 board layout




Picolo U4/U8/U16 H.264 block diagram (xx = 4/8/16)

# 2.2 Board Power Supply

### Board consumption

| Supply connector | Parameter | Min | Typ. | Max | Units |
|---|---|---|---|---|---|
| PCI Express | **Supply voltage for +3.3 V** | 3.0 | 3.3 | 3.6 | V |
| | **Supply current for +3.3 V** | | | | A |
| | **Supply voltage for +12 V** | 11.0 | 12 | 13.0 | V |
| | **Supply current for +12 V** | | | | A |
| | **Power rail requirement** | | | | W |

### Notes

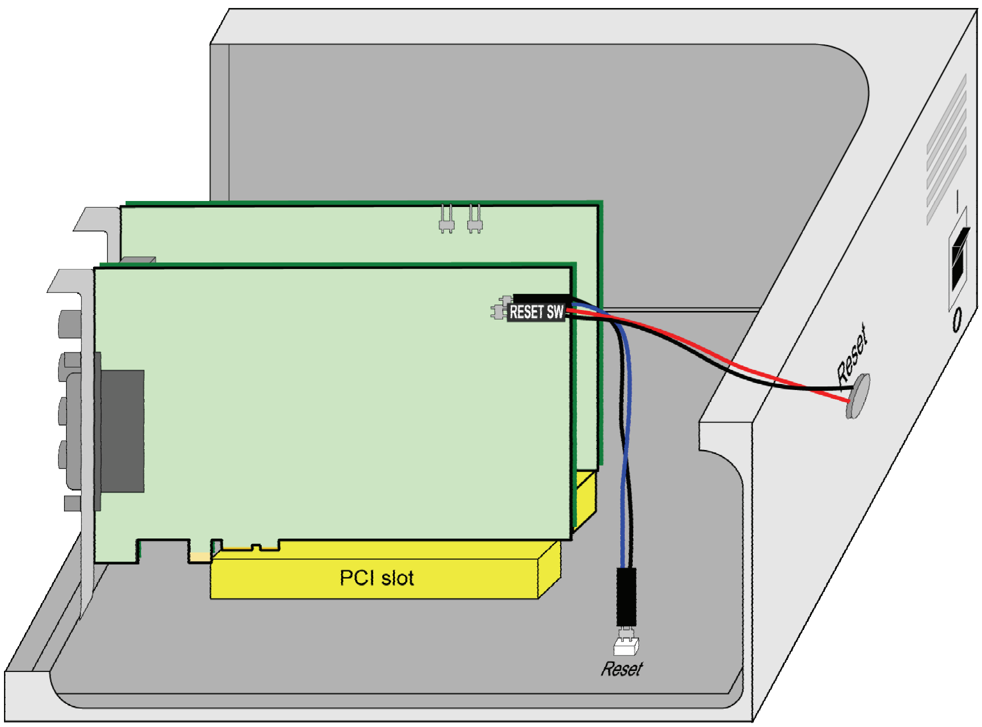
- Typical supply current values are measured during normal board operation at 25°C ambient temperature and nominal supply voltages.
- Picolo U4/U8/U16 H.264 exceeds the PCI Express 10W maximum heat dissipation limit of a half-length standard-height x1 PCI Express card. Refer to Environmental for more details.

## 2.3 Watchdog Connector

The watchdog closes the PC reset relay during 1 second when it detects a malfunction of the system. You need to install an electrical connection between the PC reset relay and the PC reset connection on the motherboard. For this purpose, the Euresys frame grabbers are fitted with a PC reset header.

For the location of the PC reset header, refer to Board Layout and Block Diagram. If several frame grabbers are installed in the PC, one reset connection is sufficient as shown on the following figure. Two PC reset connectors are provided on the frame grabbers to help cabling. The picture below shows a typical system.

After connecting the watchdog, you need to configure the watchdog parameters using the UxH264Watchdog interface. Refer to Watchdog for further information.




Typical Picolo U4/U8/U16 H.264 system setup

## 2.4 PCI Express Bus

### PCI Express Bus Compatibility

Picolo U4/U8/U16 H.264 can be inserted in a PCI Express slot inside a PC. It is compliant with the PCI Express Base Specification 1.1.

- **1-lane PCI Express**
  - The PCIe edge connector is1-lane wide. It operates at 2.5 GHz.

- Picolo U4/U8/U16 H.264 can be used in any 4-lane or larger PCIe slot. It can be used in a 16-lane PCIe slot that is not reserved for a graphical board.

## PCI Express Bus Power Supplies

As per the PCI Express Base Specification 1.1, the following power supplies are mandatory on the motherboard PCIe connectors.

- +3.3 V
- +12 V

According to the board specification, only some of these power supplies are required for the board and/or the system and cameras powering. Picolo U4/U8/U16 H.264 requires both power rails.

## DMA Transfer

Picolo U4/U8/U16 H.264 performs audio and video data transfer to the Host PC memory over the PCI Express link using Direct-Memory-Access – DMA. The Host CPU is relieved of data moving tasks.

The multi-channel DMA controller of Picolo U16 H.264 handles respectively 48 concurrent data streams: 16 encoded video, 16 raw video, and 16 audio streams.

The scatter-gather DMA controller transfers data directly into memory buffers composed with non-contiguous blocks of physical memory, allocation of contiguous physical memory blocks and buffer copy are avoided.

The 64-bit enabled DMA controller is compliant with both 32-bit and 64-bit Operating Systems and is capable to address any physical memory in the 64-bit address range.

## PCI Congestion Management

In case of PCI congestion, a built-in regulation mechanism reduces automatically the frame rate of the video streams by dropping individual frames.

The frame rate reduction affects the raw video streams by giving priority to the encoded video streams. If PCI congestion still persists when all the raw video streams are stopped, the PCI congestion manager will reduce the frame rate of the encoded video streams.

In any case, the integrity of the transferred frames is preserved.

The audio streams are not affected by the PCI congestion manager.

## Additional Information

For more information about PCIe busses, their performance and their usage, refer to the Euresys PCI Express technology note.

# 2.5 Environmental

### Storage Conditions

| Specification | Min | Max | Units |
|---|---|---|---|
| **Temperature range** | -20 / -4 | +70 / +158 | °C / °F |
| **Humidity range** | 10 | 90 | % RH non-condensing |

### Operating Conditions

| Specification | Min | Max | Units |
|---|---|---|---|
| **Ambient air temperature range(*)** | 0 / +32 | +50 / +122 | °C / °F |
| **Ambient humidity range** | 10 | 90 | % RH non-condensing |
| **Recommended limit for the board temperature(**)** | | +75 / +167 | °C / °F |
| **Absolute limit for the board temperature** | | +80 / +176 | °C / °F |

(*) The ambient air temperature is measured in the close vicinity of the fan inlet of the active heatsink, at a distance of about 10 mm above the PCB. See also Board Layout.

(**) The board temperature is measured on the surface of the PCB, at the location of the temperature monitor. See also Board Layout.

You must ensure the PC cooling system be efficient to keep the Picolo U4/U8/U16 H.264 boards below the recommended limit and also the maximum ambient air temperature. Picolo U4/U8/U16 H.264 is not operable when board temperature exceeds the absolute limit. Any attempt to operate Picolo U4/U8/U16 H.264 when board temperature exceeds the absolute limit can damage the board.

Refer to Temperature Monitor for more details.

### Extreme Storage Recovery Conditions

The board may not be powered immediately after leaving storage conditions that are outside the normal operating conditions.

A recovery time is required to ensure that:

- all components have reached the normal temperature conditions and,
- eventually, all traces of condensed water have disappeared.

For boards having a fan, such as the Picolo U16 H.264, a recovery time of 24 hours from cold storage conditions is required to avoid damages to the ball bearings and reduction of the lifetime of the fan.

### Certifications

See Declaration of Conformity.

## 2.6 Declaration of Conformity

**Notice for USA**

Compliance Information Statement (Declaration of Conformity Procedure)
DoC FCC Part 15

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules.

These limits are designed to provide reasonable protection against harmful interference in a residential installation or when the equipment is operated in a commercial environment.

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

**Notice for Europe**

This product is in conformity with the Council Directive 89/336/EEC
amended by 92/31/EEC and 93/68/EEC

This equipment has been tested and found to comply with EN55022/CISPR22 and EN55024/CISPR24. To meet EC requirements, shielded cables must be used to connect a peripheral to the board. This product has been tested in a typical class B compliant host system. It is assumed that this product will also achieve compliance in any class B compliant unit.

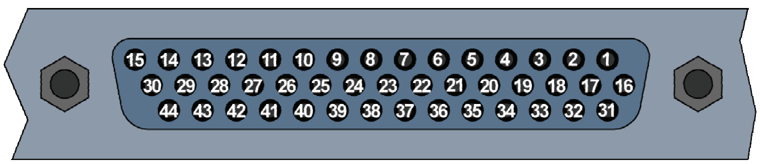
This product is in conformity with the European Union RoHS Directive, that stands for "the restriction of the use of certain hazardous substances in electrical and electronic equipment". This directive will ban the placing on the EU market of new electrical and electronic equipment containing more than agreed levels of lead, cadmium, mercury, hexavalent chromium, polybrominated biphenyl (PBB) and polybrominated diphenyl ether (PBDE) flame retardants, from 1 July 2006.

# 3. Connectors and Switches Specifications

## 3.1 VIDEO/VIDEO IN Connectors

### VIDEO Connector Layout

The **VIDEO** connector is a HD44F —high density 44-pins female— Sub-D connector. The connector shell is fitted with two UNC4-40 female screw locks.




VIDEO connector layout

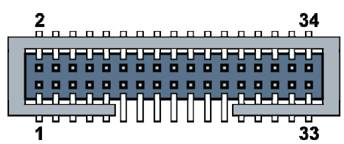
### VIDEO connector pins assignment

| Pin # | Pin name | Function (Picolo U4 H.264) | Function (Picolo U8 H.264) | Function (Picolo U16 H.264) |
|---|---|---|---|---|
| 1 | VRN_IN_CAS | Cascade Video Input Return | Cascade Video Input Return | Cascade Video Input Return |
| 2 | VID_IN_CAS | Cascade Video Input | Cascade Video Input | Cascade Video Input |
| 3 | VID_IN12 | — | Video Input 5 | Video Input 12 |
| 4 | VID_IN4 | — | Video Input 1 | Video Input 4 |
| 5 | VRN_IN15 | Video Input 4 Return | Video Input 8 Return | Video Input 15 Return |
| 6 | VID_IN15 | Video Input 4 | Video Input 8 | Video Input 15 |
| 7 | VID_IN7 | Video Input 2 | Video Input 4 | Video Input 7 |
| 8 | — | — | — | — |
| 9 | VRN_IN10 | — | — | Video Input 10 Return |
| 10 | VID_IN10 | — | — | Video Input 10 |
| 11 | VID_IN2 | — | — | Video Input 2 |
| 12 | VID_IN13 | — | — | Video Input 13 |
| 13 | VRN_IN5 | — | — | Video Input 5 Return |
| 14 | VID_IN5 | — | — | Video Input 5 |
| 15 | — | — | — | — |
| 16 | VRN_OUT | Video Output Return | Video Output Return | Video Output Return |
| 17 | VRN_IN16 | — | Video Input 7 Return | Video Input 16 Return |
| 18 | VRN_IN12 | — | Video Input 5 Return | Video Input 12 Return |
| 19 | VRN_IN4 | — | Video Input 1 Return | Video Input 4 Return |
| 20 | — | — | — | — |
| 21 | VRN_IN11 | Video Input 3 Return | Video Input 6 Return | Video Input 11 Return |
| 22 | VRN_IN7 | Video Input 2 Return | Video Input 4 Return | Video Input 7 Return |
| 23 | — | — | — | — |
| 24 | VRN_IN14 | — | — | Video Input 14 Return |
| 25 | VRN_IN6 | — | — | Video Input 6 Return |
| 26 | VRN_IN2 | — | — | Video Input 2 Return |
| 27 | VRN_IN13 | — | — | Video Input 13 Return |
| 28 | VRN_IN9 | — | — | Video Input 9 Return |
| 29 | VRN_IN1 | — | — | Video Input 1 Return |
| 30 | — | — | — | — |
| 31 | VID_OUT | Video Output | Video Output | Video Output |

| | | | | |
|---|---|---|---|---|
| **32** | **VID_IN16** | — | Video Input 7 | Video Input 16 |
| **33** | **VRN_IN8** | — | Video Input 3 Return | Video Input 8 Return |
| **34** | **VID_IN8** | — | Video Input 3 | Video Input 8 |
| **35** | — | — | — | — |
| **36** | **VID_IN11** | Video Input 3 | Video Input 6 | Video Input 11 |
| **37** | **VRN_IN3** | Video Input 1 Return | Video Input 2 Return | Video Input 3 Return |
| **38** | **VID_IN3** | Video Input 1 | Video Input 2 | Video Input 3 |
| **39** | **VID_IN14** | — | — | Video Input 14 |
| **40** | **VID_IN6** | — | — | Video Input 6 |
| **41** | — | — | — | — |
| **42** | — | — | — | — |
| **43** | **VID_IN9** | — | — | Video Input 9 |
| **44** | **VID_IN1** | — | — | Video Input 1 |

## *VIDEO IN Connector Layout*

The **VIDEO IN** connector is a straight FTSH34M connector: a vertical mount dual-row 34-pin micro header with end shroud.




VIDEO IN connector layout

### VIDEO IN connector pins assignment

| Pin # | Pin name | Function (Picolo U4 H.264) | Function (Picolo U8 H.264) | Function (Picolo U16 H.264) |
|---|---|---|---|---|
| 1 | GND | Ground | Ground | Ground |
| 2 | GND | Ground | Ground | Ground |
| 3 | VID_IN1 | — | — | Video Input 1 - Signal |
| 4 | VRN_IN1 | — | — | Video Input 1 - Return |
| 5 | VID_IN5 | — | — | Video Input 5 - Signal |
| 6 | VRN_IN5 | — | — | Video Input 5 - Return |
| 7 | VID_IN9 | — | — | Video Input 9 - Signal |
| 8 | VRN_IN9 | — | — | Video Input 9 - Return |
| 9 | VID_IN13 | — | — | Video Input 13 - Signal |
| 10 | VRN_IN13 | — | — | Video Input 13 - Return |
| 11 | VID_IN2 | — | — | Video Input 2 - Signal |
| 12 | VRN_IN2 | — | — | Video Input 2 - Return |
| 13 | VID_IN6 | — | — | Video Input 6 - Signal |
| 14 | VRN_IN6 | — | — | Video Input 6 - Return |
| 15 | VID_IN10 | — | — | Video Input 10 - Signal |
| 16 | VRN_IN10 | — | — | Video Input 10 - Return |
| 17 | VID_IN14 | — | — | Video Input 14 - Signal |
| 18 | VRN_IN14 | — | — | Video Input 14 - Return |
| 19 | VID_IN3 | Video Input 1 | Video Input 2 | Video Input 3 - Signal |
| 20 | VRN_IN3 | Video Input 1 - Return | Video Input 2 - Return | Video Input 3 - Return |
| 21 | VID_IN7 | Video Input 2 | Video Input 4 | Video Input 7 - Signal |
| 22 | VRN_IN7 | Video Input 2 - Return | Video Input 4 - Return | Video Input 7 - Return |
| 23 | VID_IN11 | Video Input 3 | Video Input 6 | Video Input 11 - Signal |
| 24 | VRN_IN11 | Video Input 3 - Return | Video Input 6 - Return | Video Input 11 - Return |
| 25 | VID_IN15 | Video Input 4 | Video Input 8 | Video Input 15 - Signal |
| 26 | VRN_IN15 | Video Input 4 - Return | Video Input 8 - Return | Video Input 15 - Return |
| 27 | VID_IN4 | — | Video Input 1 | Video Input 4 - Signal |
| 28 | VRN_IN4 | — | Video Input 1 - Return | Video Input 4 - Return |
| 29 | VID_IN8 | — | Video Input 3 | Video Input 8 - Signal |
| 30 | VRN_IN8 | — | Video Input 3 - Return | Video Input 8 - Return |
| 31 | VID_IN12 | — | Video Input 5 | Video Input 12 - Signal |
| 32 | VRN_IN12 | — | Video Input 5 - Return | Video Input 12 - Return |

| | | | | |
|---|---|---|---|---|
| **33** | **VID_IN16** | — | Video Input 7 | Video Input 16 - Signal |
| **34** | **VRN_IN16** | — | Video Input 7 - Return | Video Input 16 - Return |

# Video Inputs Electrical Specifications

The **VIDEO** connector has **4/8/16** identical Video Inputs. Each port has two pins named **VID_INxx** and **VRN_INxx** respectively, where **xx** is a number ranging from **1** to **4/8/16**. The "hot conductor" of the transmission line transporting the video signal **xx** has to be connected to the **VID_INxx** pin, while the "cold conductor" of the transmission line has to be connected to the corresponding **VRN_INxx** pin. If the transmission line is a coaxial line, the "hot conductor" is the inner conductor, and the "cold conductor" is the outer conductor.

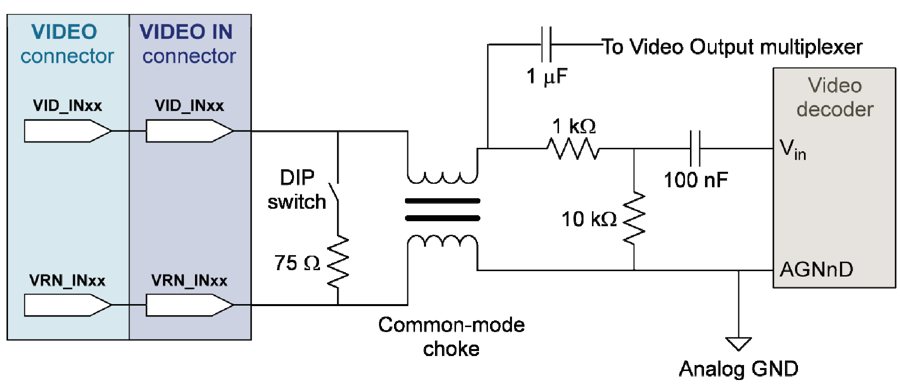
A switchable 75Ω termination resistor is inserted directly across **VID_INxx** and **VRN_INxx**. The termination resistor can be disconnected by a DIP switch.

The factory settings of the termination resistor DIP switch are **ON**.

The video signal flows through a common-mode choke which attenuates any common-mode noise present on the video signal. The **VRN_INxx** pin is connected to the analog **GND** through one winding of the common-mode choke. The **VID_INxx** pin is connected to the second winding.

The filtered video signal (slightly attenuated by the 1kΩ/10kΩ resistor network) is AC-coupled to a video input of the video decoder. The 1 kΩ resistor protects the video decoder input against excessive currents in case of abnormal signal levels applied on the Video inputs.

The filtered video signal is also AC-coupled to an input of the video multiplexer circuit.




Video Input circuit

### Video inputs - DC characteristics

| Parameter | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Input voltage range - Absolute max rating** | -0.55 | | +2.2 | V |
| **Input impedance - Terminator** OFF | | 11 | | Ω |
| **Input impedance - Terminator** ON | | 75 | | Ω |

For a correct operation of the video decoder, the video signal must satisfy all following electrical requirements.

### Video signal electrical requirements

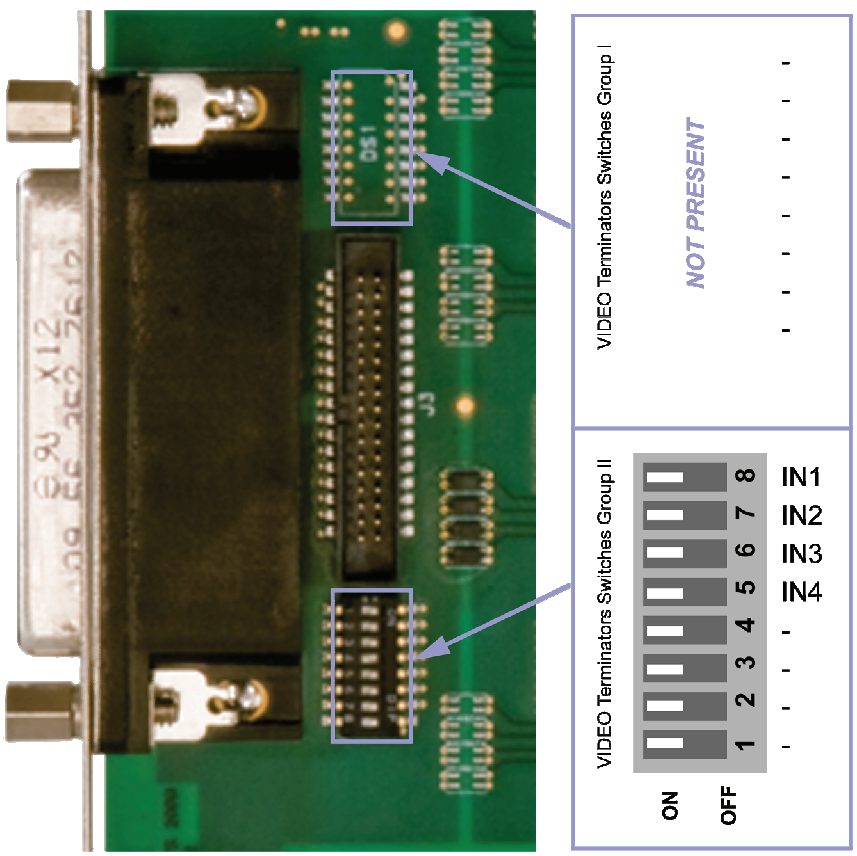
| Parameter | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Overall peak-to-peak amplitude** | 0.5 | 1.0 | 1.5 | V |
| **Sync amplitude** | 150 | 300 | 400 | mV |
| **Rise/Fall time of sync edges** | 50 | | 300 | ns |

Lower video signal amplitudes are not recommended.

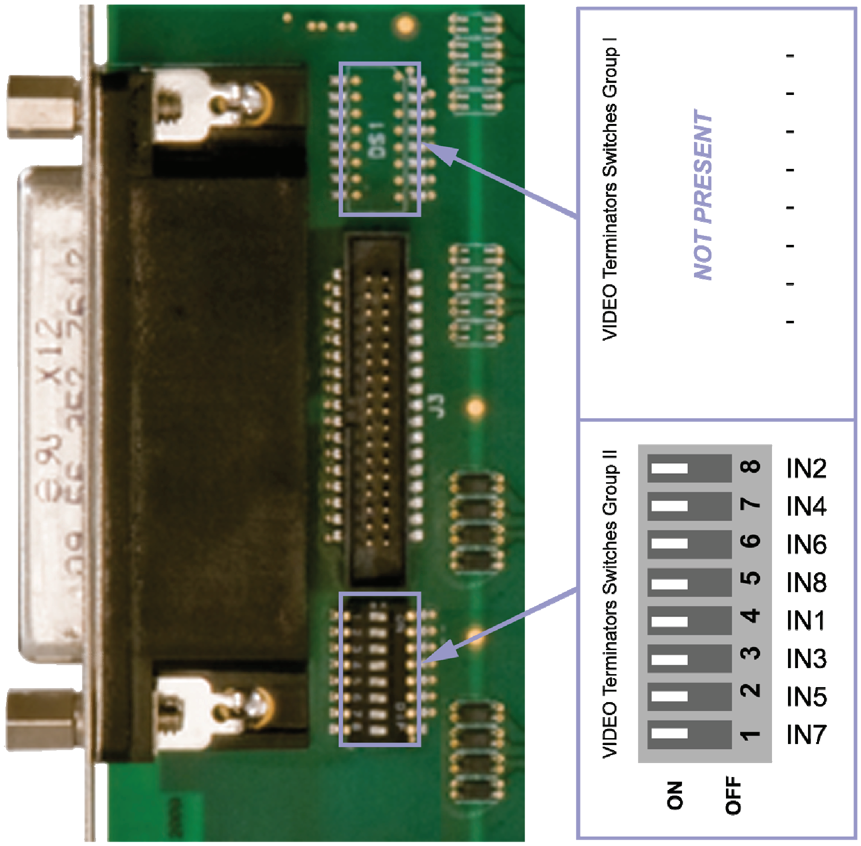
The signal attenuation induced by the 1kΩ/10kΩ resistor network circuit is compensated by the front-end amplifier in the input stage of the video decoder.

## 3.2 Video Terminators Switches

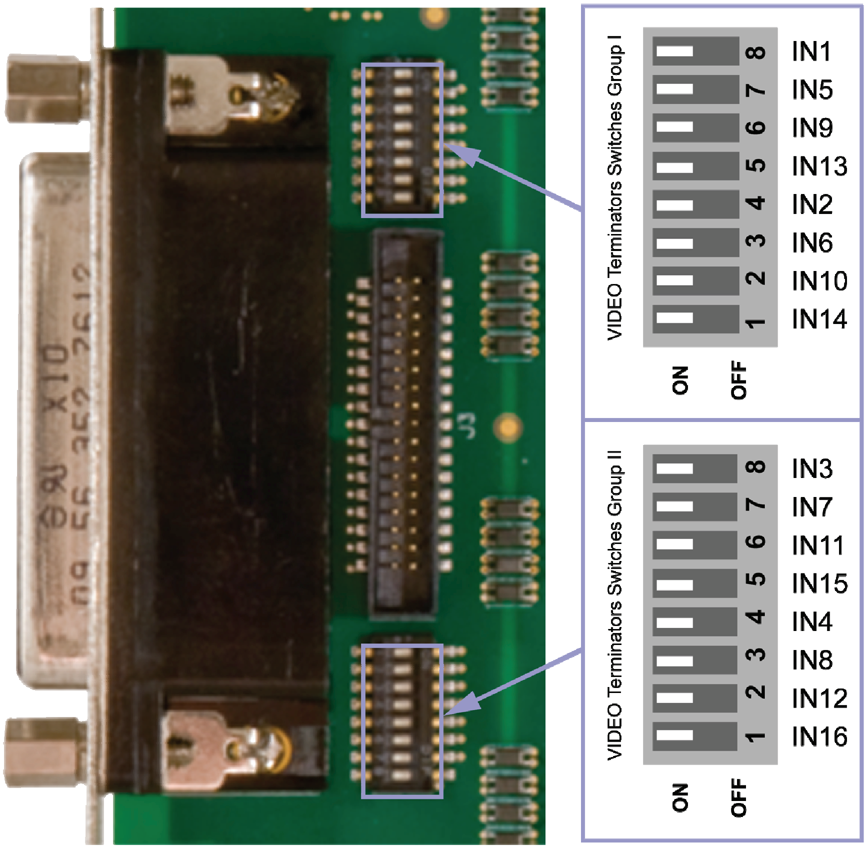
The Video Terminators Switches are split in two groups of 8 switches: Group **I** and Group **II**.




Picolo U4 H.264 Video Terminators Switches assignments




Picolo U8 H.264 Video Terminators Switches assignments

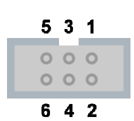



Picolo U16 H.264 Video Terminators Switches assignments

## 3.3 VIDEO CASCADE Connector

### VIDEO CASCADE Connector Layout

The **VIDEO CASCADE** connector is a straight PH64M connector: a vertical mount dual-row 6-pin header without shroud.




VIDEO CASCADE connector layout

### VIDEO CASCADE connector pins assignment

| Pin # | Pin name | Function |
|---|---|---|
| 1 | VID_OUT | Video Output |
| 2 | VRN_OUT | Video Output Return |
| 3 | VID_IN_CAS | Video Cascade Input |
| 4 | VRN_IN_CAS | Video Cascade Input Return |
| 5 | — | Not connected |
| 6 | — | Not connected |

# Cascade Video Input Electrical Specifications

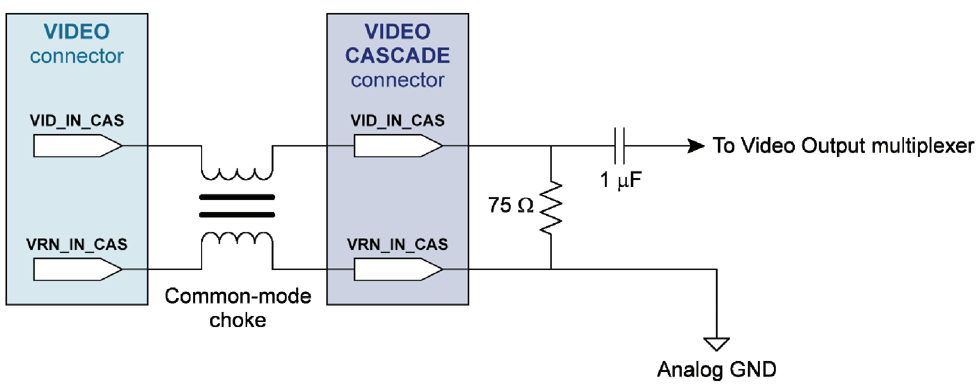
Picolo U4/U8/U16 H.264 has one Cascade Video Input port for both VIDEO and VIDEO CASCADE connectors.

The ports have two pins named **VID_IN_CAS** and **VRN_IN_CAS** respectively. The "hot conductor" of the transmission line transporting the video signal **xx** has to be connected to the **VID_IN_CAS** pin, while the "cold conductor" of the transmission line has to be connected to the corresponding **VRN_IN_CAS** pin. If the transmission line is a coaxial line, the "hot conductor" is the inner conductor, and the "cold conductor" is the outer conductor.

A fixed 75Ω termination resistor is inserted across **VID_IN_CAS** and **VRN_IN_CAS**.

When this is applied on the **VIDEO** connector port, the cascade video signal flows through a common-mode choke, which attenuates the common-mode noise present on the video signal. The **VRN_IN_CAS** pin is connected to the analog **GND** through one winding of the common-mode choke. The **VID_IN_CAS** pin is connected to the second winding.

The filtered video signal is available on the **VIDEO CASCADE** connector. It is applied to the Video Output multiplexer through a coupling capacitor.




Cascade Video Input circuit

### DC characteristics

| Cascade Video Input - DC characteristics | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Input voltage range - Absolute max rating** | -2 | | +2 | V |
| **Input impedance** | | 75 | | Ω |

# Video Output Electrical Specifications

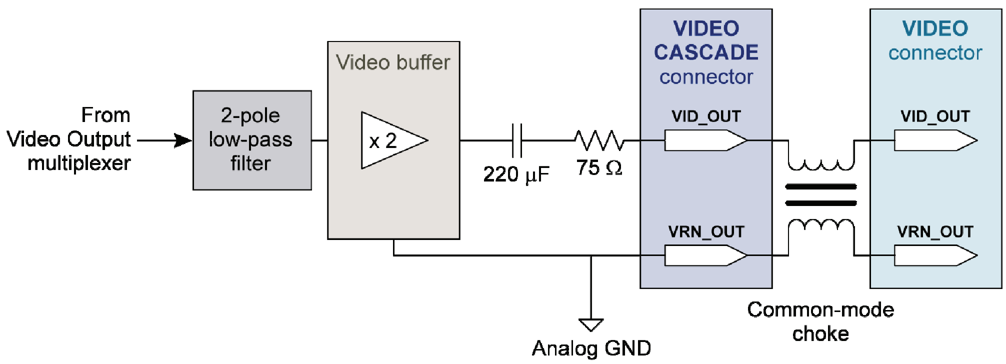
Picolo U4/U8/U16 H.264 has one Video Output port on both VIDEO and VIDEO CASCADE connectors.

Each port has two pins named **VID_OUT** and **VRN_OUT** respectively. The "hot conductor" of the transmission line transporting the video signal has to be connected to the **VID_OUT** pin, while the "cold conductor" of the transmission line has to be connected to the **VRN_OUT** pin. If the transmission line is a coaxial line, the "hot conductor" is the inner conductor, and the "cold conductor" is the outer conductor.

The output of the Video Output multiplexer is low-pass filtered, then amplified by a video buffer with a fixed gain of 2.

The output of the buffer is AC-coupled and serially terminated by a 75Ω termination resistor inserted in the output path.

The video output is further filtered by a common-mode choke, which prevents common-mode noise present on the video signal to penetrate inside. The **VRN_OUT** pin is connected to the analog **GND** through one winding of the common-mode choke; The **VID_OUT** pin is connected to the second winding.




Video Output circuit

## DC characteristics

| Video Output - DC characteristics | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Output impedance** | | 75 | | Ω |

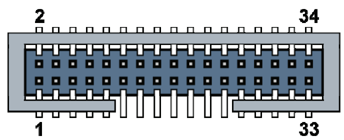
## AC characteristics

| Video Output - AC characteristics | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Bandwidth** | 50 | | 7,000,000 | Hz |

# 3.4 AUDIO Connector

## AUDIO Connector Layout

The **AUDIO** connector is a straight FTSH34M connector: a vertical mount dual-row 34-pin micro header with end shroud.




AUDIO connector layout

### AUDIO connector pins assignment

| Pin # | Pin name | Function (Picolo U4 H.264) | Function (Picolo U8 H.264) | Function (Picolo U16 H.264) |
|---|---|---|---|---|
| 1 | GND | Ground | Ground | Ground |
| 2 | GND | Ground | Ground | Ground |
| 3 | AUD_IN1 | — | — | Audio Input 1 - Signal |
| 4 | GND | — | — | Audio Input 1 - Return |
| 5 | AUD_IN5 | — | — | Audio Input 5 - Signal |
| 6 | GND | — | — | Audio Input 5 - Return |
| 7 | AUD_IN9 | — | — | Audio Input 9 - Signal |
| 8 | GND | — | — | Audio Input 9 - Return |
| 9 | AUD_IN13 | — | — | Audio Input 13 - Signal |
| 10 | GND | — | — | Audio Input 13 - Return |
| 11 | AUD_IN2 | — | — | Audio Input 2 - Signal |
| 12 | GND | — | — | Audio Input 2 - Return |
| 13 | AUD_IN6 | — | — | Audio Input 6 - Signal |
| 14 | GND | — | — | Audio Input 6 - Return |
| 15 | AUD_IN10 | — | — | Audio Input 10 - Signal |
| 16 | GND | — | — | Audio Input 10 - Return |
| 17 | AUD_IN14 | — | — | Audio Input 14 - Signal |
| 18 | GND | — | — | Audio Input 14 - Return |
| 19 | AUD_IN3 | Audio Input 1 | Audio Input 2 | Audio Input 3 - Signal |
| 20 | GND | Audio Input 1 - Return | Audio Input 2 - Return | Audio Input 3 - Return |
| 21 | AUD_IN7 | Audio Input 2 | Audio Input 4 | Audio Input 7 - Signal |
| 22 | GND | Audio Input 2 - Return | Audio Input 4 - Return | Audio Input 7 - Return |
| 23 | AUD_IN11 | Audio Input 3 | Audio Input 6 | Audio Input 11 - Signal |
| 24 | GND | Audio Input 3 - Return | Audio Input 6 - Return | Audio Input 11 - Return |
| 25 | AUD_IN15 | Audio Input 4 | Audio Input 8 | Audio Input 15 - Signal |
| 26 | GND | Audio Input 4 - Return | Audio Input 8 - Return | Audio Input 15 - Return |
| 27 | AUD_IN4 | — | Audio Input 1 | Audio Input 4 - Signal |
| 28 | GND | — | Audio Input 1 - Return | Audio Input 4 - Return |
| 29 | AUD_IN8 | — | Audio Input 3 | Audio Input 8 - Signal |
| 30 | GND | — | Audio Input 3 - Return | Audio Input 8 - Return |
| 31 | AUD_IN12 | — | Audio Input 5 | Audio Input 12 - Signal |
| 32 | GND | — | Audio Input 5 - Return | Audio Input 12 - Return |

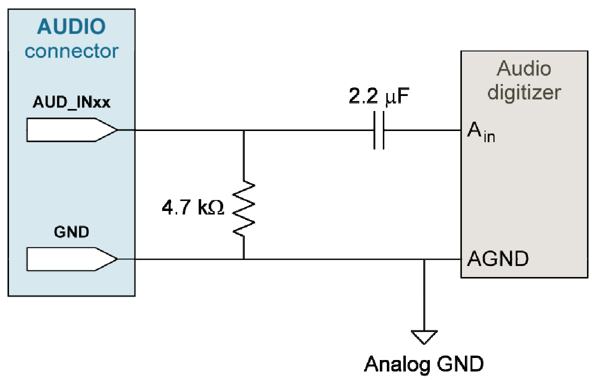
| | | | | |
|---|---|---|---|---|
| **33** | **AUD_IN16** | — | Audio Input 7 | Audio Input 16 - Signal |
| **34** | **GND** | — | Audio Input 7 - Return | Audio Input 16 - Return |

# Audio Inputs Electrical Specifications

There are 4/8/16 identical line-level analog Audio Input ports on the AUDIO connector.

Each port has two pins named **AUD_INxx** and **GND** respectively, where **xx** is a number ranging from **1** to **4/8/16**.

The line-level audio signal is applied to the ADC input through a AC coupling capacitor.




Audio Input circuit

### Audio Input - DC characteristics

| Parameter | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Input voltage range - Absolute max rating** | -30 | | +30 | V |
| **Input impedance at 1 kHz** | | >4.7 | | kΩ |

For a correct operation of the audio decoder, the audio signal must satisfy the following electrical requirements:

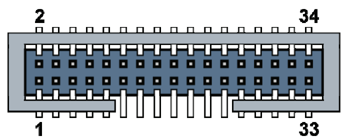
### Audio signal characteristics

| Parameter | Min | Typ. | Max | Units |
|---|---|---|---|---|
| **Audio level range** | 0.5 | 1.0 | 1.65 | $V_{ptp}$ |

# 3.5 PROFESSIONAL I/O Connectors

## PROFESSIONAL I/O 1-8 Connector Layout

The **PROFESSIONAL I/O 1-8** connector is a straight FTSH34M connector: a vertical mount dual-row 34-pin micro header with end shroud.




PROFESSIONAL I/O 1-8 connector layout

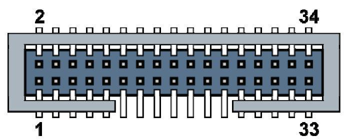
### PROFESSIONAL I/O 1-8 pins assignment

| Pin # | Pin name | Function (Picolo U4 H.264) | Function (Picolo U8 H.264 Picolo U16 H.264) |
|---|---|---|---|
| 1 | GND | Ground | Ground |
| 2 | GND | Ground | Ground |
| 3 | IN1-A | Input 1 - A side | Input 1 - A side |
| 4 | IN1-B | Input 1 - B side | Input 1 - B side |
| 5 | IN2-A | Input 2 - A side | Input 2 - A side |
| 6 | IN2-B | Input 2 - B side | Input 2 - B side |
| 7 | IN3-A | Input 3 - A side | Input 3 - A side |
| 8 | IN3-B | Input 3 - B side | Input 3 - B side |
| 9 | IN4-A | Input 4 - A side | Input 4 - A side |
| 10 | IN4-B | Input 4 - B side | Input 4 - B side |
| 11 | IN5-A | — | Input 5 - A side |
| 12 | IN5-B | — | Input 5 - B side |
| 13 | IN6-A | — | Input 6 - A side |
| 14 | IN6-B | — | Input 6 - B side |
| 15 | IN7-A | — | Input 7 - A side |
| 16 | IN7-B | — | Input 7 - B side |
| 17 | IN8-A | — | Input 8 - A side |
| 18 | IN8-B | — | Input 8 - B side |
| 19 | OUT1-A | Output 1 - A side | Output 1 - A side |
| 20 | OUT1-B | Output 1 - B side | Output 1 - B side |
| 21 | OUT2-A | Output 2 - A side | Output 2 - A side |
| 22 | OUT2-B | Output 2 - B side | Output 2 - B side |
| 23 | OUT3-A | Output 3 - A side | Output 3 - A side |
| 24 | OUT3-B | Output 3 - B side | Output 3 - B side |
| 25 | OUT4-A | Output 4 - A side | Output 4 - A side |
| 26 | OUT4-B | Output 4 - B side | Output 4 - B side |
| 27 | OUT5-A | — | Output 5 - A side |
| 28 | OUT5-B | — | Output 5 - B side |
| 29 | OUT6-A | — | Output 6 - A side |
| 30 | OUT6-B | — | Output 6 - B side |
| 31 | OUT7-A | — | Output 7 - A side |

| | | | |
|---|---|---|---|
| **32** | **OUT7-B** | — | Output 7 - B side |
| **33** | **OUT8-A** | — | Output 8 - A side |
| **34** | **OUT8-B** | — | Output 8 - B side |

## PROFESSIONAL I/O 9-16 Connector Layout

The **PROFESSIONAL I/O 9-16** connector is a straight FTSH34M connector: a vertical mount dual-row 34-pin micro header with end shroud.




PROFESSIONAL I/O 9-16 connector layout

## PROFESSIONAL I/O 9-16 pins assignment

| Pin # | Pin name | Function (Picolo U16 H.264 only) |
|---|---|---|
| 1 | **GND** | Ground |
| 2 | **GND** | Ground |
| 3 | **IN9-A** | Input 9 - A side |
| 4 | **IN9-B** | Input 9 - B side |
| 5 | **IN10-A** | Input 10 - A side |
| 6 | **IN10-B** | Input 10 - B side |
| 7 | **IN11-A** | Input 11- A side |
| 8 | **IN11-B** | Input 11 - B side |
| 9 | **IN12-A** | Input 12 - A side |
| 10 | **IN12-B** | Input 12 - B side |
| 11 | **IN13-A** | Input 13 - A side |
| 12 | **IN13-B** | Input 13 - B side |
| 13 | **IN14-A** | Input 14 - A side |
| 14 | **IN14-B** | Input 14 - B side |
| 15 | **IN15-A** | Input 15 - A side |
| 16 | **IN15-B** | Input 15 - B side |
| 17 | **IN16-A** | Input 16 - A side |
| 18 | **IN16-B** | Input 16 - B side |
| 19 | **OUT9-A** | Output 9 - A side |
| 20 | **OUT9-B** | Output 9 - B side |
| 21 | **OUT10-A** | Output 10 - A side |
| 22 | **OUT10-B** | Output 10 - B side |
| 23 | **OUT11-A** | Output 11 - A side |
| 24 | **OUT11-B** | Output 11 - B side |
| 25 | **OUT12-A** | Output 12 - A side |
| 26 | **OUT12-B** | Output 12 - B side |
| 27 | **OUT13-A** | Output 13 - A side |
| 28 | **OUT13-B** | Output 13 - B side |
| 29 | **OUT14-A** | Output 14 - A side |
| 30 | **OUT14-B** | Output 14 - B side |
| 31 | **OUT15-A** | Output 15 - A side |

| | | |
|---|---|---|
| **32** | **OUT15-B** | Output 15 - B side |
| **33** | **OUT16-A** | Output 16 - A side |
| **34** | **OUT16-B** | Output 16 - B side |

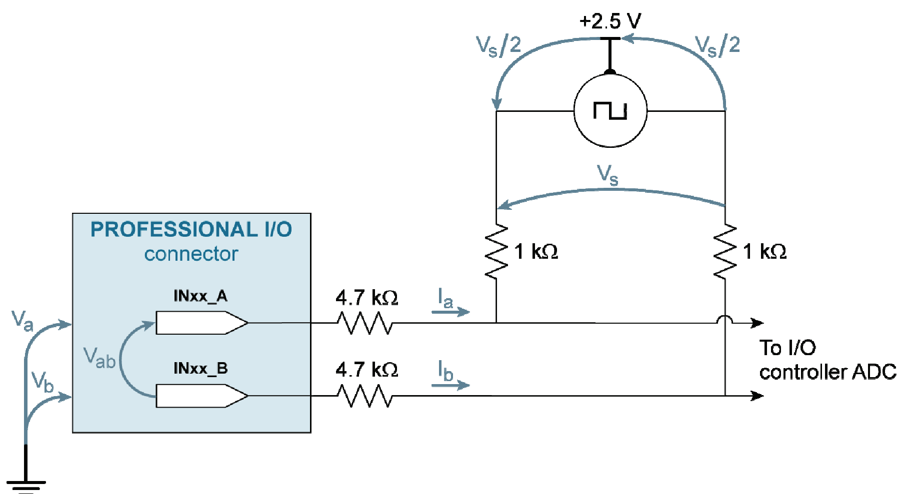
## *Professional Inputs 1-16 Electrical Specifications*

There are 4/8/16 identical general purpose digital inputs on the **I/O** connector. Each port exposes two pins named **INxx-A** and **INxx-B** respectively, where **xx** is a number ranging from **1** to **4/8/16**.

This digital non-isolated differential input is polarity insensitive. Following type of devices are accepted:

- Digital Totem-Pole drivers at TTL, 5V CMOS, or 12V levels
- Potential-free contact closure (dry contacts)
- Fixed-potential contact closure (one pin of the contact at fixed potential)

The I/O controller measures the differential voltage and the differential impedance across both pins of each port sequentially. Therefore, it stimulates the external devices by applying alternatively positive and negative stimulation voltage $V_s$, as shown on the following diagram.




**Professional Inputs 1-16 circuit**

- When the measured impedance is above the impedance threshold, the I/O controller reports an **OPEN** state.
- When the measured impedance is below the impedance threshold, the I/O controller considers the differential voltage.
- When the measured differential voltage is above the voltage threshold, the I/O controller returns a **HIGH** state, else it returns a **LOW** state.

The impedance threshold is not adjustable. The voltage threshold is selectable among three values to determine the logic level of TTL, 5V CMOS, or 12V signals.

**Hint.** When you connect a low-impedance digital source (for instance a TTL gate), the capability to detect an open state can be used to detect a broken line.

### Absolute max ratings

| Parameter | Symbol | Min | Typ. | Max | Units |
|---|---|---|---|---|---|
| **Absolute max voltage** | $Va_{AbsMax}$, $Vb_{AbsMax}$ | | | 25 | V |
| **Absolute min voltage** | $Va_{AbsMin}$, $Vb_{AbsMin}$ | -25 | | | V |

The port does not operate correctly over the specified range. For normal operating conditions, refer to the DC characteristics table below.

The specification applies for both on and off power conditions.

### DC characteristics

| Parameter | Test condition(*) | Symbol | Min | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| **AC stimulation voltage** | | \|Vs\| | | 1.67 | | $V_{ptp}$ |
| **DC bias voltage** | | $V_{bias}$ | | 2.5 | | V |
| **Input voltage range** | "Logic input" operation | $Va_{range}$,$Vb_{range}$ | -7.8 | | 12.8 | V |
| **Input current** | Va (or Vb) = - 5.0 V | Ia (or Ib) | | 1,32 | | mA |
| | Va (or Vb) = - 0.0 V | | | 0,44 | | mA |
| | Va (or Vb) = 2.5 V | | | 0,00 | | mA |
| | Va (or Vb) = 5.0 V | | | -0,44 | | mA |
| | Va (or Vb) = 12.0 V | | | -1,67 | | mA |
| TTL Logic input operation (3V p.t.p signal) | | | | | | |
| **Differential voltage threshold** | Va in Vb within input voltage range | $V_{TTLth}$ | | 1.5 | | V |
| **Common-mode voltage(**)** | | $V_{TTLcmv}$ | -6.3 | 1.5(***) | 11.3 | V |
| 5V CMOS logic input operation (5V p.t.p. signal) | | | | | | |
| **Differential voltage threshold** | Va in Vb within input voltage range | $V_{CMOSth}$ | | 2.5 | | V |
| **Common-mode voltage(**)** | | $V_{CMOScmv}$ | -5.3 | 2.5(***) | 10.3 | V |
| 12V logic input operation (12V p.t.p signal) | | | | | | |
| **Differential voltage threshold** | Va in Vb within input voltage range | $V_{12Vth}$ | | 6 | | V |
| **Common-mode voltage(**)** | | $V_{12Vcmv}$ | -1.8 | 6(***) | 6.8 | V |
| Contact operation - both pins floating | | | | | | |
| **Differential impedance threshold** | Potential-free contact | $R_{PFth}$ | | 5 | | kΩ |
| **Common-mode voltage(**)** | | $V_{PFcmv}$ | | 2.5 | | V |
| Contact operation - one pin at fixed potential | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Differential impedance threshold** | | $R_{Fxth}$ | | 8.5 | | kΩ |
| **Common-mode voltage(****)** | | $V^{FXcmv}$ | -7.8 | | 12.8 | V |

(*) T = 25 °C, unless specified.

(**) Common-mode voltage = (Va + Vb)/2.

(***) One pin at GND.

(****) Common-mode voltage = voltage Va (or Vb) of pin at fixed potential.

### AC characteristics

| Parameter | Test condition(*) | Symbol | Value | Units |
|---|---|---|---|---|
| **Sampling rate** | | f | 200 | Hz |
| **Deglitching filter tap count(**)** | Filter setting at 10 ms | tc | 2 | Samples |
| | Filter setting at 100 ms | | 20 | Samples |
| **Minimum pulse width(***)** | Filter setting at 10 ms | $pw_{Min}$ | 35 | ms |
| | Filter setting at 100 ms | | 115 | ms |

(*) $T_{amb}$ = 25°C, unless otherwise specified.

(**) 'tc' is the number of taps of the deglitching filter. The deglitching filter removes all transient input states having a number of consecutive identical samples smaller or equal to the value tc.

(***) This indicates the smallest pulse that the input lines can safely detect. This minimal value cannot be guaranteed in all circumstances; however, it can serve as guide lines for the system design.

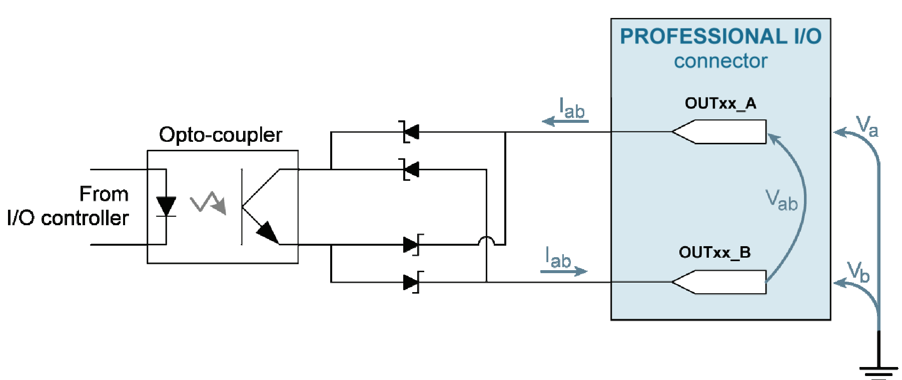
# *Professional Outputs 1-16 Electrical Specifications*

There are 4/8/16 general purpose outputs on the **I/O** connector.

Each port has two pins named **OUTxx-A** and **OUTxx-B** respectively, where **xx** is a number ranging from **1** to **4/8/16**.

This output port emulates a potential-free and polarity-free solid-state contact.

The contact remains in the **OPEN** state during board initialization procedure.




Professional Outputs 1-16 circuit

### Absolute max ratings

| Parameter | Test condition(*) | Symbol | Min | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| **Maximum voltage** | Contact open | $\|Vab\|_{max}$ | | | 30 | V |
| **Maximum current** | Contact closed | $\|Iab\|_{max}$ | | | 50 | mA |

(*) $T_{amb}$ = 25 °C, unless specified.

### DC characteristics

| Parameter | Test condition(*) | Symbol | Min | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| **Voltage drop** | Contact closed, \|Iab\| = 0.5 mA | $\|Vab\|_{drop}$ | | 0.55 | | V |
| | Contact closed, \|Iab\| = 1 mA | | | 0.6 | | V |
| | Contact closed, \|Iab\| = 3 mA | | | 0.75 | | V |
| | Contact closed, \|Iab\| = 5 mA | | | 0.9 | | V |
| | Contact closed, \|Iab\| = 7 mA | | | 1.5 | | V |
| | Contact closed, \|Iab\| = 10 mA | | | 2.8 | | V |
| | Contact closed, \|Vab\|  15 V | | | 0.5 | 0.7 | µA |
| | | | | | 500 | $V_{ACrms}$ |

### AC characteristics

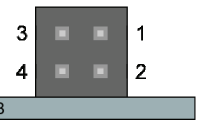
| Parameter | Test condition (*) | Symbol | Min | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| **Turn-off time** | Contact closed, \|Iab\| = 1 mA | $t_{off}$ | | 60 | | µs |
| | Contact closed, \|Iab\| = 5 mA | | | 20 | | µs |
| | Contact closed, \|Iab\| = 10 mA | | | 5 | | µs |
| **Turn-on time** | Contact closed, \|Iab\| = 1 mA | $t_{on}$ | | 2 | | µs |
| | Contact closed, \|Iab\| = 5 mA | | | 5 | | µs |
| | Contact closed, \|Iab\| = 10 mA | | | 10 | | µs |

(*) $T_{amb}$ = 25 °C, unless specified.

# 3.6 WATCHDOG Connector

## WATCHDOG Connector Layout

The **WATCHDOG** connector is a right-angled PH4M connector: a horizontal mount dual-row 4-pin header without shroud.




WATCHDOG connector layout

WATCHDOG connector pins assignment

| Pin # | Pin name | Function |
|---|---|---|
| 1 | **RST-A** | Reset output - A side |
| 2 | **RST-B** | Reset output - A side |
| 3 | **RST-B** | Reset output - B side |
| 4 | **RST-A** | Reset output - B side |

**Note.** You can connect up to two reset switches vertically or horizontally. For example, across 1-2 and 3-4, or 1-3 and 2-4 pins.

## Watchdog Output Electrical Specifications

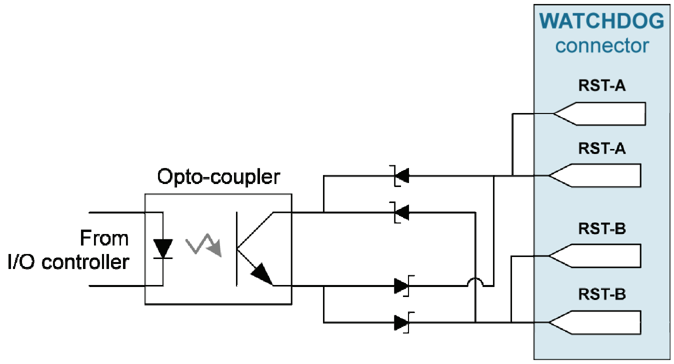
Picolo U4/U8/U16 H.264 has a dual Watchdog Output on the WATCHDOG connector.

Each port has two pins named **RST-A** and **RST-B**.

This output port emulates a potential-free and polarity-free solid-state contact.

The contact remains in the **OPEN** state during board initialization procedure.

The Watchdog Output shares the same electrical specifications as the I/O outputs.




Watchdog Output circuit

# 3.7 PCI Express Connector

The PCI Express connector is a vertical card edge connector. The side B is the primary (component) side; the side A is the secondary (solder) side.

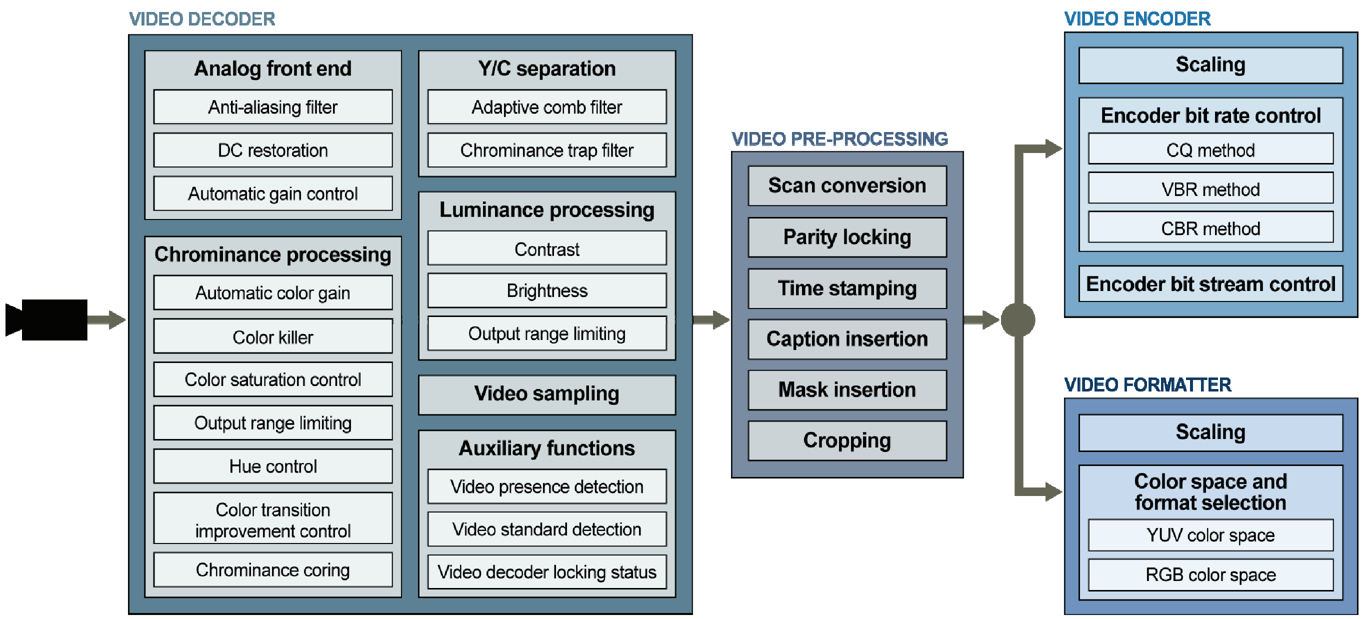
## PCI Express connector pins assignment

| Side B pin # | Pin name | Function | Side A pin # | Pin name | Function |
|---|---|---|---|---|---|
| **1** | **+12V** | 12V Power input | **1** | **NPRSTN1** | Connected to **nPRSNT2** (pin B17) |
| **2** | **+12V** | 12V Power input | **2** | **+12V** | 12V Power input |
| **3** | **+12V** | 12V Power input | **3** | **+12V** | 12V Power input |
| **4** | **GND** | Ground | **4** | **GND** | Ground |
| **5** | **SMBCLK** | - | **5** | **TCK** | 3.3V Power input |
| **6** | **SMBDAT** | - | **6** | **TDI** | Connected to **TDO** (pin A7) |
| **7** | **GND** | - | **7** | **TDO** | Connected to **TDO** (pin A6) |
| **8** | **+3.3V** | 3.3V Power input | **8** | **TMS** | - |
| **9** | **nTRST** | - | **9** | **+3.3V** | 3.3V Power input |
| **10** | **+3.3V** | - | **10** | **+3.3V** | 3.3V Power input |
| **11** | **nWAKE** | - | **11** | **nPERST** | PCI Reset input |

## Mechanical key

| Side B pin # | Pin name | Function | Side A pin # | Pin name | Function |
|---|---|---|---|---|---|
| **12** | **RSVD** | - | **12** | **GND** | Ground |
| **13** | **GND** | Ground | **13** | **REFCLK+** | Reference clock Differential input pair |
| **14** | **PETp(0)** | PCI Express Lane 0 Differential input pair | **14** | **REFCLK-** | |
| **15** | **PETn(0)** | | **15** | **GND** | Ground |
| **16** | **GND** | Ground | **16** | **PERp(0)** | PCI Express Lane 0 Differential output pair |
| **17** | **nPRSNT2** | Connected to **nPRSNT1** (pin A1) | **17** | **PERn(0)** | |
| **18** | **GND** | Ground | **18** | **GND** | |

# 4. Frame Grabber Operation




Frame grabber operation block diagram

## 4.1 Video Decoder

## Multi-Standard Video Decoder

The video decoder of Picolo U4/U8/U16 H.264 supports the following popular analog color television standards:

- **PAL-B/D**
- **NTSC-M**

You can use the DirectShow API to determine the video standard.

Setting the video standard enforces the following settings of the video decoder:

- The video system setting
- The color killer configuration.

The following table shows the enforced settings of the decoder for each one of the 4 allowed values of the standard:

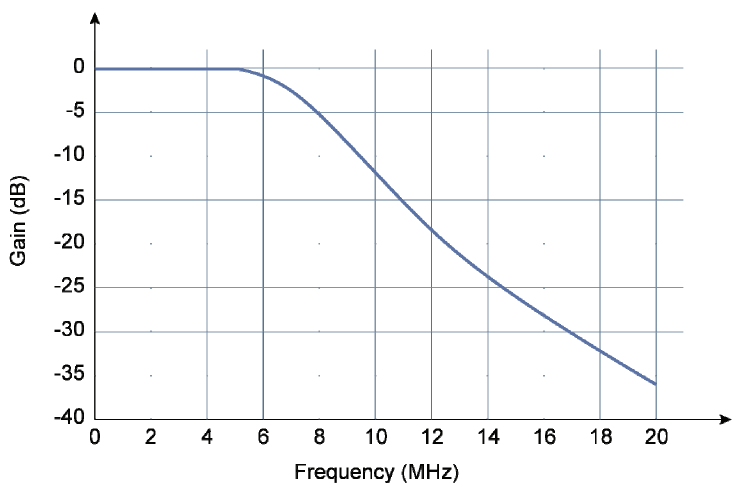
| Standard | Video system | Color killer |
|---|---|---|
| **PAL** | PAL-B/D | Color is always alive |
| **NTSC** | NTSC-M | |
| **CCIR** | PAL-B/D | Color is always killed |
| **EIA** | NTSC-M | |

# *Analog Front End*

The following sections show the functional blocks that process the analog video signal.

## Anti-Aliasing Filter

The analog front end of the video decoder includes an anti-aliasing low-pass filter that attenuates undesirable high frequencies in the video signal before converting to digital video signal.

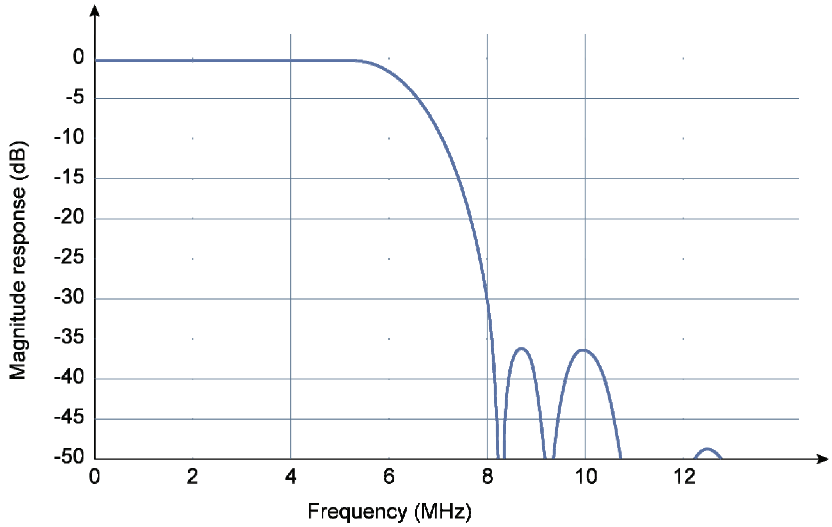



**Frequency response of the anti-aliasing filter**

The filter attenuation is always set to -3 dB at 7 MHz, and -10 dB at 10 MHz.

## Decimation Filter

The over-sampled digitized composite video data are decimated using a digital decimation filter.




Decimation filter

### Automatic Gain Control (AGC)

The analog front end of the video decoder includes a variable gain amplifier and an automatic gain control loop that restore the nominal sync amplitude of the signal before converting to digital video signal. In other words, the video decoder can receive incoming signals at various strength and yet maintain a constant video signal by adjusting the weaker signal to receive more gain, stronger signals to receive less gain or none at all.

The AGC is always enabled. The gain range of the variable gain amplifier ranges from -6 dB up to 18 dB, allowing a wide range of video signal amplitudes to be correctly decoded.

### DC Restoration

The DC level of the video is restored using back-porch clamping technique.

## *Y/C Separation*

### Adaptive Comb Filter

This filter extracts the luminance and the chrominance out of the color composite PAL/NTSC signals. This filter improves the luminance resolution, and reduces noise such as cross-luminance (rain bowling) and cross-color (dot crawl). The adaptive comb filter is always enabled.

## *Luminance Processing*

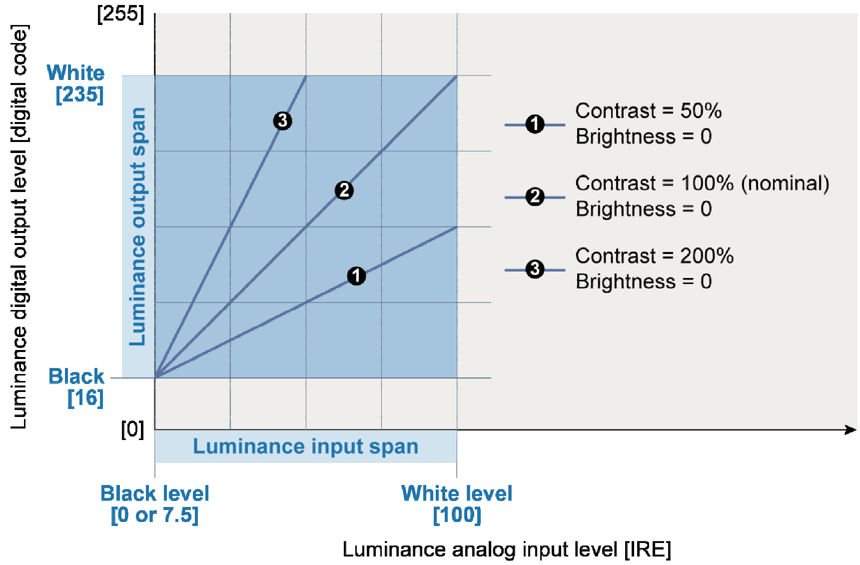
The following sections show the functional blocks used for luminance processing.

## Contrast

You can adjust picture contrast by applying gain on luminance component through the Contrast property. This contrast control can be used at anytime, even during acquisition. The decoder provides a luminance gain adjustment in 256 steps, ranging from 0 up to 200 %, with a default value of 100 %.

The following figure shows the luminance transfer function for three contrast settings and the nominal brightness setting.

The luminance output span is expressed in 8-bit digital codes according to ITU-R BT.601-4 specification.

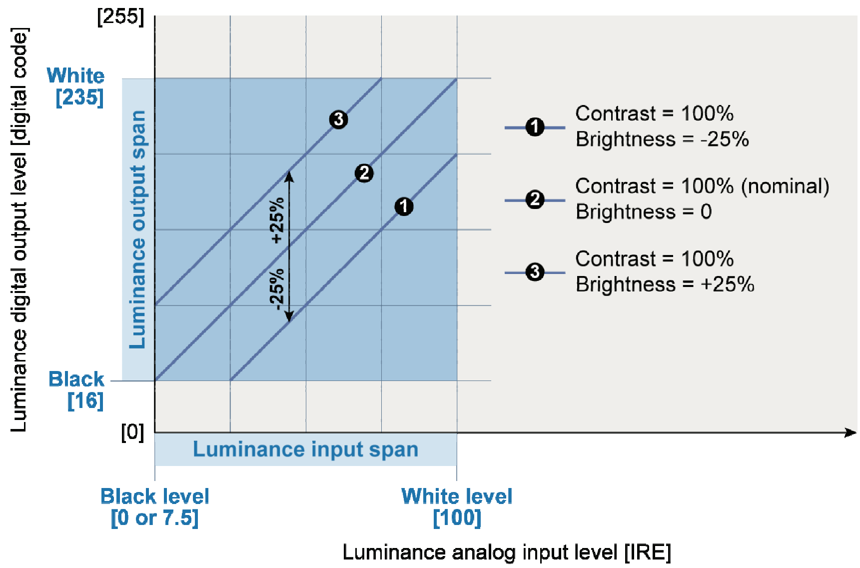



Contrast (luminance gain) control

## Brightness

You can adjust picture brightness by applying offset on luminance component through the Brightness property. This contrast control can be used at anytime, even during acquisition. Picolo U4/U8/U16 H.264 provides an offset gain adjustment in 256 steps, ranging from -25 % up to 25 % of the output span.

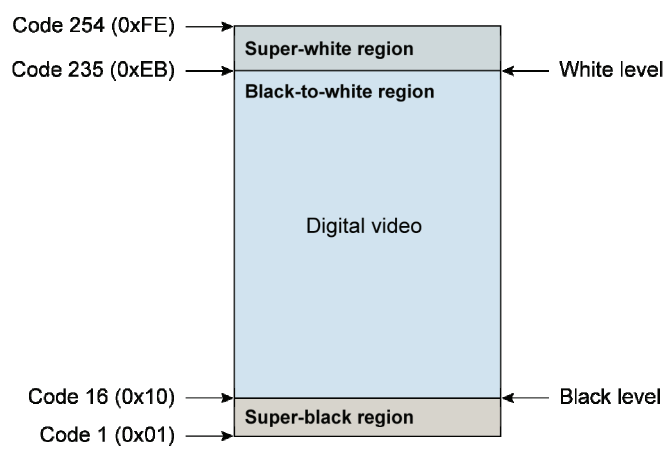
The following figure shows the luminance transfer function for three brightness settings and the nominal brightness setting.




Brightness (luminance gain) control

## Luminance Output Range Limiting

The luminance component Y is delivered in 8-bit digital codes, according to ITU-R BT.601-4 specification. The luminance spans over 220 levels, with the black level corresponding to code 16 (0x10), and white level corresponding to code 235 (0xEB). The following diagram shows the luminance output range.




Luminance output range

The range of luminance output is divided into three regions, as shown in the above drawing.

- The **super-black** region (code from 1 up to 15) and the **super-white** region (code from 236 up to 254) provide some extra margins for occasional excursion of the signal beyond the normal limits, or for some contrast and/or brightness settings.

- The **black-to-white** region is the effective luminance output span: a correctly adjusted digital video renderer considers exclusively this region.

The luminance output range is normally bounded in the range [2..254]. The darkest luminance levels will not produce a luminance output code below 2, and the brightest luminance levels will not produce a luminance output code above 254 (0xFE).

# *Chrominance Processing*

The following sections show the functional blocks used for chrominance processing.

## Automatic Color Gain

The automatic color gain restores the nominal chrominance levels. The gain range extends from -6 dB (attenuation by a factor of 2) up to +30 dB (amplification by a factor of 32). The automatic color gain is always enabled.

## Color Killer

Automatic color killer disables the chrominance decoding when the incoming video signal is a black-and-white signal, or when the incoming signal is very weak or noisy.

You can configure the automatic color killer through the DirectShow API.

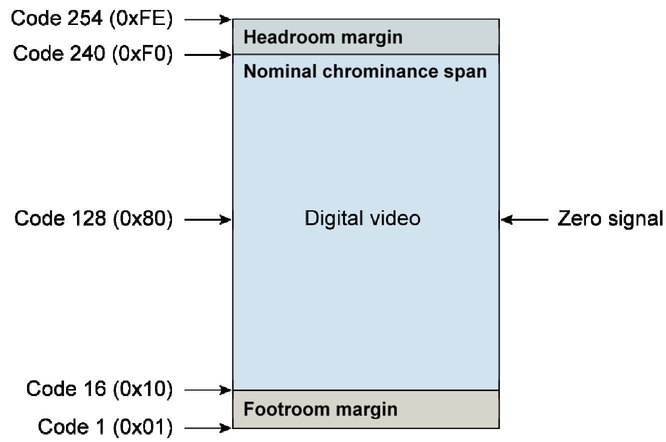
## Color Saturation Control

You can adjust color saturation by applying gain on chrominance component through the Saturation property. his color saturation control can be used at anytime, even during acquisition. The decoder provides a luminance gain adjustment in 256 steps, ranging from 0 up to 200 %, with a default value of 100 %.

## Output Range Limiting

The chrominance components U and V are delivered in 8-bit digital codes, according to ITU-R BT.601-4 specification. The chrominance spans over 225 levels in the center part of the [0..255] scale, with zero signal corresponding to 128 (0x80). The nominal range of chrominance levels is 16 (0x10) up to 240 (0xF0).

The chrominance output range is bounded in the decoder. The lowest value for a chrominance component is 2. The highest value for a chrominance component is 254 (0xFE).




ITU-R BT.601-4 chrominance output range

The chrominance output range is normally bounded in the range [2..254]. The darkest chrominance levels will not produce a chrominance output code below 2. The brightest chrominance levels will not produce a chrominance output code above 254 (0xFE).

## Hue Control

The hue control is always fixed to 0 %.

## Color Transition Improvement Control

CTI gain is always set at 75 %.

## Chrominance Coring

The coring is always set to 128 ± 2. This means that chrominance codes 126, 127, 129 and 130 are substituted by chrominance code 128.

# Video Sampling

The on-board video decoder digitizes the analog video signal using the YUV 4:2:2 sampling method. The luminance component Y is sampled at 13.5 MHz, and the chrominance components U and V are sampled at half-frequency at 6.75 MHz. The sampling rates are identical for both 625-line and 525-line television systems.

The sampling rates are not adjustable, the actual luminance sampling rate is proportional to the line frequency of the video signal, in order to produce a fixed integer amount of samples per line. There are 858 samples per line in case of 525-line television systems, and 864 samples per line in case of 625-line television systems. The sampling clock is a phase-locked on the falling edge of the sync tip.

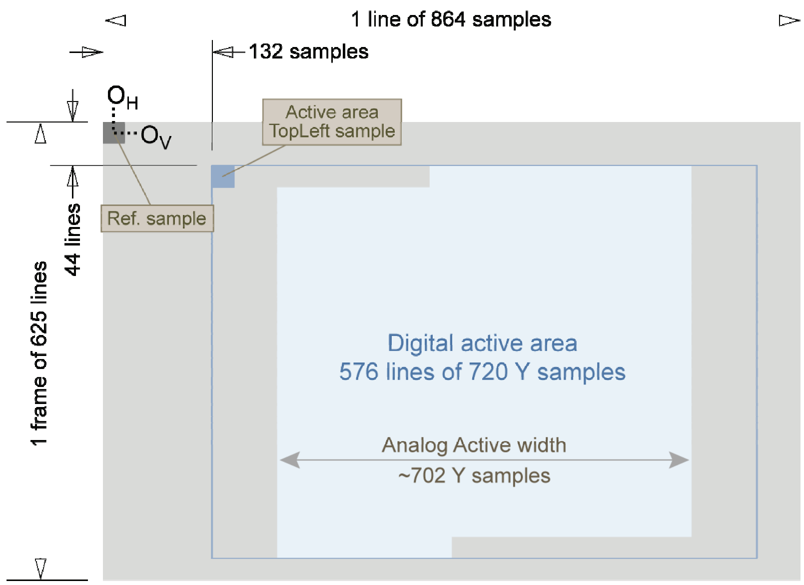
The sample aspect ratio is about 11/10 for 525-line system, and about 54/59 for 625-line system.

## Digital Active Area

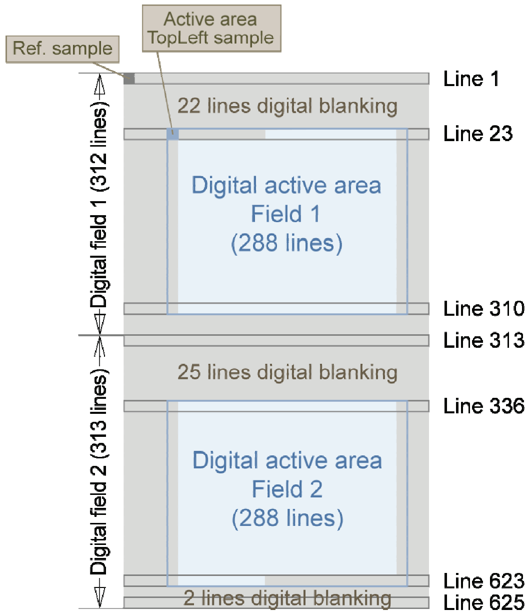
The digital active area is a rectangular area of digitally active samples. Its size and position depend on the actual television system of the video signal.

### Digital Active Area (625-line television system)

For 625-line television system, the size and position of the digital active area is shown in the next figure.




**Digital active area definition — 625-line system**
One frame after scan conversion (spatial line ordering)



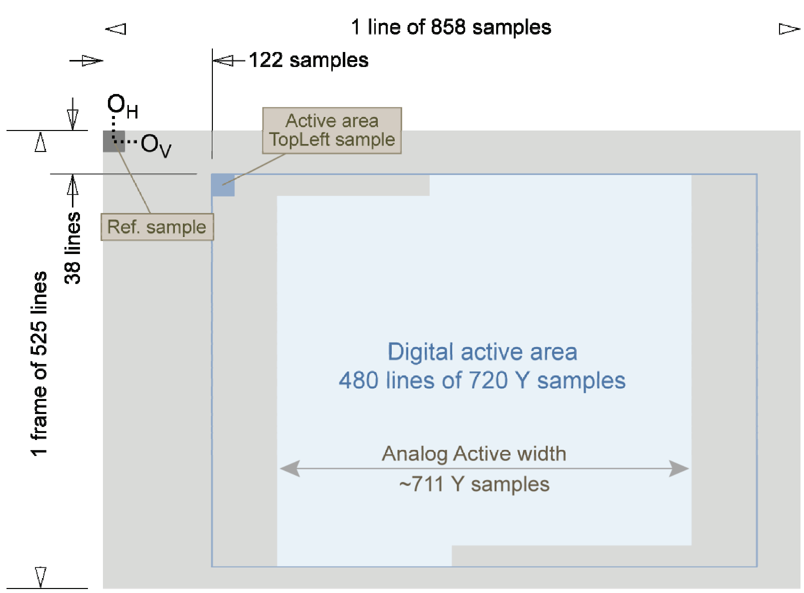

**Digital active area definition — 625-line system**
Two successive fields before scan conversion (temporal line ordering)

The digital active area size and position are defined according to ITU-R BT.601 standard.

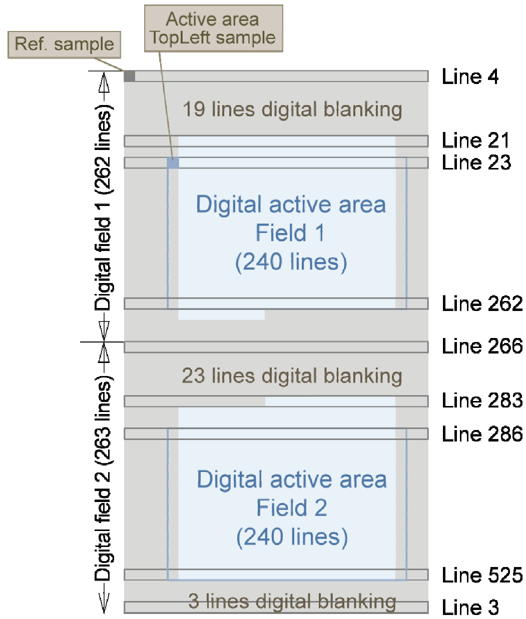
The analog active width is about 18 pixels narrower than the digital active width, leaving a blanked margin of 8~9 pixels on the left and right sides. The first and the last lines of the digital active area contain partial video lines.

### Digital Active Area (525-line television system)

For 525-line television system, the size and position of the digital active area are shown in the next figure.




**Digital active area definition — 525-line system**
One frame after scan conversion (spatial line ordering)




**Digital active area definition — 625-line system**
Two successive fields before scan conversion (temporal line ordering)

The digital active area size and position are defined according to ITU-R BT.601 standard.

The digital active height and the vertical position are defined according to SMPTE recommended practice RP-202, "Video Alignment for MPEG coding". The digital active area includes the 240 lowest full lines of each field, for a total of 480 lines.

The analog active width is about 9 pixels narrower than the digital active width, leaving a blanked margin of 4~5 pixels on the left and right sides.

# Auxiliary Functions

The following sections show the auxiliary functions available for the video decoder.

## Video Decoder Locking Status

You can use the DirectShow API to report the video decoder locking status.

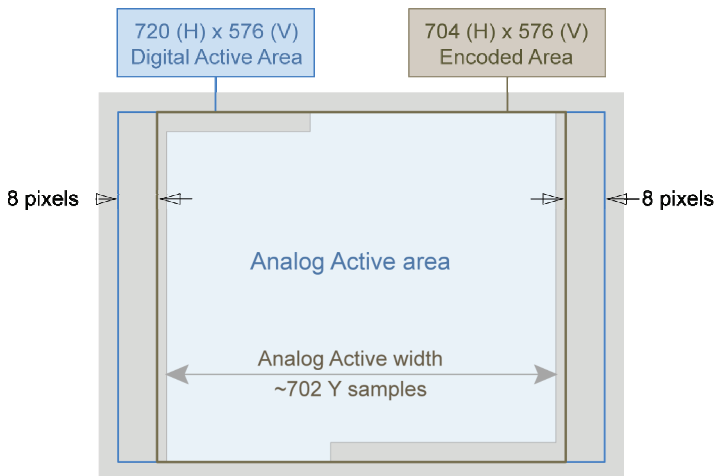
## Video Standard Detection

You can use the DirectShow API to report the detected standard.

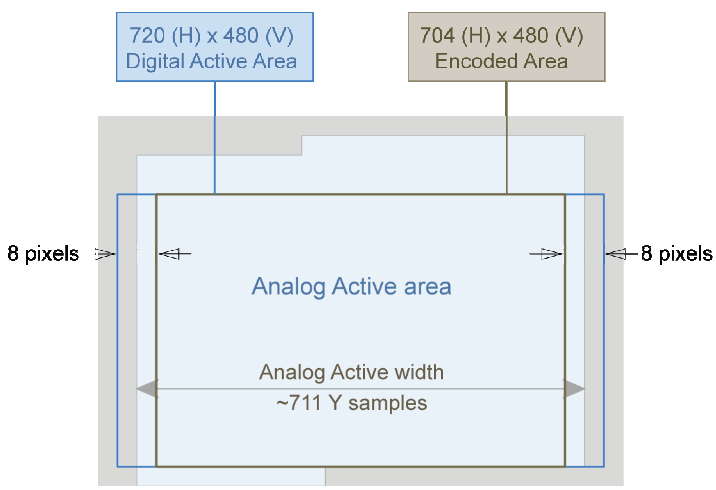
# 4.2 Video Pre-Processing

## Cropping

The source image is always cropped horizontally to retain 704 pixels per line, for both 525-line and 625-line television systems. The cropped area is centered on the digital active area. Eight pixels are cropped at both and right sides.




**Cropping**
625-line television system




**Cropping**
525-line television system

The cropped area is:

- 704 (H) x 576 (V) for 625-line television system

- 704 (H) x 480 (V) for 525-line television system

For 625-line system:

- The cropped area still contains some blank pixels at both left and right sides, typically 1 pixel wide in case of a nominal analog video signal.
- The cropped area still contains some blank pixels at both top and bottom sides, typically 1/2 line at left side of topmost line, 1/2 line at right side of bottommost line in case of a nominal analog video signal.

For 525-line system:

- There are no blank pixels at left and right sides of the cropped area, assuming a nominal analog video signal.
- There are no blank half-lines at top and bottom sides of the cropped area, assuming a nominal analog video signal.

## Mask Insertion

Every video channel of Picolo U4/U8/U16 H.264 embeds a privacy mask insertion function.

Up to four rectangular regions of the Digital Active Area can be masked. The masked regions are colored in black. The masked regions are inserted on both the formatted and compressed streams.

The mask boundaries fit the macro block borders; the mask area is automatically modified to the smallest enclosing region which fits the 16 x 16 macro blocks border.

The digital active area is divided into square macro blocks of 16 x 16 pixels. The PAL digital active area is composed with 44 x 36 macroblocks; the NTSC digital active area is composed with 44 x 30 macroblocks.

You can define the position and the size of the four regions using the Mask0, Mask1, Mask2, Mask3 property.

Picolo U4/U8/U16 H.264 can achieve 4/8/16 privacy masks updates per second for all inputs.

## Caption Insertion

Every video channel of Picolo U4/U8/U16 H.264 embeds a text caption insertion function.

Up to 5(*) lines of up to 47 characters can be engraved in the image. The same text is inserted on both the formatted and compressed streams.

(*)The number of caption text lines may be limited to 2 according to the driver's specification. For example, 5 lines is applicable with the version 2.0 of the Virtual File System driver.

The character set is the 7-bit ASCII character set. Non-printable characters are displayed as small rectangles. The font is not configurable. The character cell size is 16x26 pixels.

The text can be located at four pre-defined positions, or anywhere in the active area (custom position):

- **Top left**: the text line is top justified with a top margin of 16 rows of pixels. The text line is left justified with a left margin of 16 columns of pixels.
- **Top right**: the text line is top justified with a top margin of 16 rows of pixels. The text line is right justified with a right margin of 16 columns of pixels.
- **Bottom left**: the text line is bottom justified with a bottom margin of 16 rows of pixels. The text line is left justified with a left margin of 16 columns of pixels.

- **Bottom right**: the text line is bottom justified with a bottom margin of 16 rows of pixels. The text line is right justified with a right margin of 16 columns of pixels.
- **Custom**: the text line is left justified. The position of the top-left pixel of the first character is freely configurable.

The caption foreground is white. The character background is semi-transparent: the excursion of the video luminance is limited upwards

You can control the caption insertion by using the Caption0, Caption1 property.

Picolo U4/U8/U16 H.264 can achieve 4/8/16 caption text updates per second for all inputs.

## Scan Conversion

The video decoder delivers an ITU-R BT.656 YUV422 digital video stream to the data input port of the on-board processor.

Two consecutive fields of the incoming stream are assembled to build a YUV420 planar image frame, that is stored in a temporary buffer —the video buffer— located into the memory space of the on-board processor.

The video buffer serves as data source for both the video encoder and the video formatter.

## Time Stamping

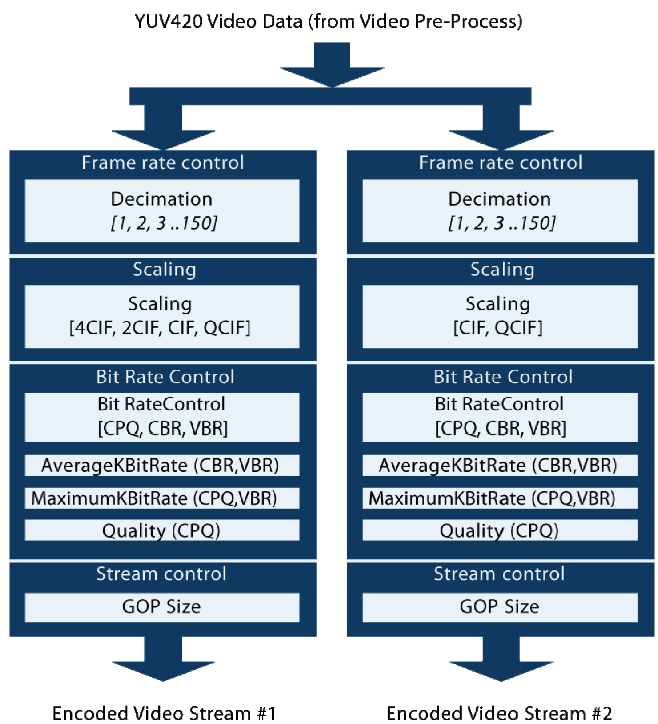
Picolo U4/U8/U16 H.264 embeds a local time clock. The local time clock is implemented with a counter that increments at a rate of 90 kHz.

Every video frame (consisting of a pair of fields) is time-stamped. The time stamp is the local time when the first sample of field 1 is delivered by the video decoder through the bus. The video time stamp period is typically 33 milliseconds for NTSC video signals, and 40 milliseconds for PAL video signals.

Every audio frame (consisting of 160 milliseconds of audio data) is time-stamped. The time stamp is actually the local time when the first audio sample is delivered by the audio interface through the bus.

# 4.3 Video Encoder

The Video Encoder is capable of delivering simultaneously two encoded video streams from a single source.




## Frame Rate Control (Video Encoder)

The frame rate of each encoded videos stream can be reduced by adjusting the Decimation property.

$$\text{frame rate} = \frac{\text{full frame rate}}{\text{decimation}}$$

where full frame rate is:

- 29.97 Hz for 525-line television systems
- 25.00 Hz for 625-line television systems

**Decimation** is any integer value within **[1..150]**. By default, the value is **1**. You can modify the value at anytime even during the acquisition.

This simple decimation method offers the theoretical advantage of a constant interval between captured frames.

However, in case of congestion of the output path, the frame rate control circuit may occasionally increase the interval between captured frames. Such occurrence is not signaled to the application.

To minimize the risk of a congestion of the video encoder output path, the available PCI bandwidth is first allocated to the video encoder output path, the remaining bandwidth is allocated to the video formatter output path.

# Scaling (Video Encoder)

The Video Encoder embeds a scaling function that scales down the images from the native 4CIF resolution to one of the following resolutions 2CIF, CIF, and QCIF.

The resolution of each Encoded Video Stream can be defined independently.The first stream can be configured for any one of the four available resolutions: 4CIF, 2CIF, CIF, and QCIF.The second encoded stream can also be configured for any one of the four available resolutions but both encoded streams are subject to the amount of instantaneously available encoding power for the considered video channel.

You are able to reduce the image size on both horizontal and vertical axis by using the DirectShow API.

The default resolution is 4CIF for the first stream and CIF for the second stream.

Following table shows the size of the images, the television system and the selected scaling factor.

| Scaling | Scaling factor (H x V) | 625-line system (H x V) | 525-line system (H x V) |
|---|---|---|---|
| **4CIF** | 1 x 1 | 704 x 576 | 704 x 480 |
| **2CIF** | 1 x 2 | 704 x 288 | 704 x 240 |
| **CIF** | 2 x 2 | 352 x 288 | 352 x 240 |
| **QCIF** | 4 x 4 | 176 x 144 | 176 x 112(*) |

(*) In QCIF resolution, the video encoder delivers 8 lines less than the video formatter. These lines are the 4 top and the 4 bottom lines which are not encoded.

# Encoder Bit Rate Control

Three methods are provided to control the bit rate, VBR, CPQ and CBR. You can select the bit rate control method by assigning the BitRateControl property. By default, the value is CPQ.

## VBR Method

The VBR —Variable Bit Rate— method delivers an encoded video bit stream at a variable bit rate, depending on the level of motion detected in the scene. This method is a variant of the CPQ method, where you specify a maximum bit rate and an average bit rate, using the MaximumKBitRateControl property and AverageKBitRateControl property.

Typically, the average setting is 50% of the maximum setting. Full frame rate is not anymore possible when setting values above 2,000.

## CPQ Method

The CPQ —Constant Picture Quality— method delivers an encoded video bit stream at a constant picture quality level. You can adjust the picture quality level by using the
Quality property.

The maximum bit rate is specified by the application by means of the MaximumKBitRateControl property.

Full frame rate is not anymore possible when setting values above 2,000.

**Note.** The effective bit rate is continuously varying, in order to maintain a constant subjective image quality.

### CBR Method

The CBR —Constant Bit Rate— method delivers an encoded video bit stream at a constant bit rate. You can adjust the targeted bit rate by using the MaximumKBitRateControl property. Full frame rate is not anymore possible when setting values above 2,000.

**Note.** The effective bit rate is not strictly constant, it may suffer from large short-term variations. However, the long-term average bit rate matches relatively closely the setting.

## *Encoder Bit Stream Controls*

### I PERIOD (GOP Size)

I PERIOD defines the period, in frames, of the I-frames. This is also known as the GOP size, that is the number of frames in a Group Of Pictures.

You can select the GOP size by using the GOPSize property. By default, the value of GOPSize is 30.

### Encoded Bit Stream Structure

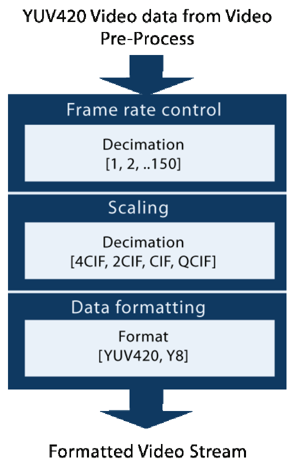
On Picolo U4/U8/U16 H.264, the encoded bit stream is structured as follows:

- SPS NAL Unit
- PPS NAL Unit
- IDR NAL Unit
- (GOPSize – 1) P-frames
- SPS NAL Unit
- PPS NAL Unit
- IDR NAL Unit

Note: Each GOP starts with an IDR slice preceded by SPS and PPS NAL Units allowing an H.264 decoder to start decoding anywhere in the stream.

# 4.4 Video Formatter

The Video Formatter is capable of delivering one raw uncompresses stream from a single source




## Frame Rate Control (Video Formatter)

The frame rate of the formatted videos stream can be reduced by adjusting the Decimation property.

$$\text{frame rate} = \frac{\text{full frame rate}}{\text{decimation}}$$

where full frame rate is:

- 29.97 Hz for 525-line television systems
- 25.00 Hz for 625-line television systems

**Decimation** is any value within [**1...150**]. By default, the value is **1**. You can modify the value at anytime even during the acquisition.

This simple decimation method offers the theoretical advantage of a constant interval between captured frames.

However, in case of congestion of the output path, the frame rate control circuit may occasionally increase the interval between captured frames. Such occurrence is not signaled to the application.

To minimize the risk if a congestion of the video encoder output path, the available output bandwidth is first allocated to the video encoder output path, the remaining bandwidth is allocated to the video formatter output path.

## Scaling (Video Formatter)

You are able to reduce the image size on both horizontal and vertical axis by using the DirectShow API.

### Size of the images, the television system and the selected scaling factor

| Scaling | Scaling factor (H x V) | 625-line system (H x V) | 525-line system (H x V) |
|---|---|---|---|
| **4CIF** | 1 x 1 | 704 x 576 | 704 x 480 |
| **2CIF** | 1 x 2 | 704 x 288 | 704 x 240 |
| **CIF** | 2 x 2 | 352 x 288 | 352 x 240 |
| **QCIF** | 4 x 4 | 176 x 144 | 176 x 120 (*) |

(*) The width and the height of the delivered image are a multiple of 16 except for the height of the QCIF image, which is multiple of 8 but not multiple of 16.

# Color Space and Format Selection

Picolo U4/U8/U16 H.264 is capable of delivering images either in RGB or in the YUV color spaces. with the following images formats:

<table>
<tr><th>Format</th><th>Color Space</th><th>Data packing</th></tr>
<tr><td>Y8</td><td rowspan="4">YUV</td><td>N/A</td></tr>
<tr><td>YUV422</td><td>Packed</td></tr>
<tr><td>YUV420PL</td><td rowspan="2">Planar</td></tr>
<tr><td>YUV422PL</td></tr>
<tr><td>RGB15</td><td rowspan="4">RGB</td><td rowspan="4">Packed</td></tr>
<tr><td>RGB16</td></tr>
<tr><td>RGB24</td></tr>
<tr><td>RGB32</td></tr>
</table>

You can select the format through the DirectShow API.

## YUV Color Space

If a format of the YUV color space is selected, the pixel data is delivered using the YCbCr color space. The video formatter does not need to perform any color space transformation, and the output levels of Y, U, V components delivered are identical to the output levels of Y, Cb, Cr components delivered by the video decoder. No additional gamma correction is performed.

#### Output values of Y, Cb and Cr components for a "75% amplitude, 75% saturated YCbCr color bar" test pattern

| Component | Light gray | Yellow | Cyan | Green | Magenta | Red | Blue | Black |
|---|---|---|---|---|---|---|---|---|
| **Y** | 180 | 162 | 131 | 112 | 84 | 65 | 35 | 16 |
| **Cb (U)** | 128 | 44 | 156 | 72 | 184 | 100 | 212 | 128 |
| **Cr (V)** | 128 | 142 | 44 | 58 | 198 | 212 | 114 | 128 |

The above results are obtained under the following conditions.

- The **Contrast** setting must be nominal (100% luminance gain).
- The **Brightness** setting must be nominal (null luminance offset).
- The **Saturation** setting must be nominal (100% color saturation).

## RGB Color Space

If a format of the RGB color space is selected, the pixel data is delivered using the RGB color space. The video formatter converts the ITU-R BT.601 digital YCbCr pixel data in a RGB pixel data using the following formulas.

```
R' = 1.164(Y - 16) + 1.596(Cr - 128)
G' = 1.164(Y - 16) - 0.813(Cr - 128) - 0.392(Cb - 128)
B' = 1.164(Y - 16) + 2.017(Cb - 128)
```

**R'**, **G'**, **B'** must be saturated at the 0 and 255 levels.

The output range of R, G, B components occupies the full digital range, as shown in the following table.

#### Output range of R, G, B components

| Format | Output code for black | Output code for white |
|---|---|---|
| **RGB15** | R = 0, G = 0, B = 0 | R = 31, G = 31, B = 31 |
| **RGB16** | R = 0, G = 0, B = 0 | R = 31, G = 63, B = 31 |
| **RGB24** | R = 0, G = 0, B = 0 | R = 255, G = 255, B = 255 |
| **RGB32** | R = 0, G = 0, B = 0 | R = 255, G = 255, B = 255 |

No additional gamma correction is performed.

**Output values of R, G, B components for a "75% amplitude, 75% saturated YCbCr color bar" test pattern**

| Component | Light gray | Yellow | Cyan | Green | Magenta | Red | Blue | Black |
|---|---|---|---|---|---|---|---|---|
| **R (Red)** | 191 | 191 | 0 | 0 | 191 | 191 | 0 | 0 |
| **G (Green)** | 191 | 191 | 191 | 191 | 0 | 0 | 0 | 0 |
| **B (Blue)** | 191 | 0 | 191 | 0 | 191 | 0 | 191 | 0 |

The above results are obtained under the following conditions.

- The **Contrast** setting must be nominal (100% luminance gain).
- The **Brightness** setting must be nominal (null luminance offset).
- The **Saturation** setting must be nominal (100% color saturation).

## 4.5 Audio Digitizer and Encoder

Picolo U4/U8/U16 H.264 has 4/8/16 high-quality audio acquisition channels. Each audio channel converts a line-level analog audio signal into a digital audio stream.

A frequency synthesizer generates the master audio clock signal from a stable oscillator. Two master audio clock frequencies are available:

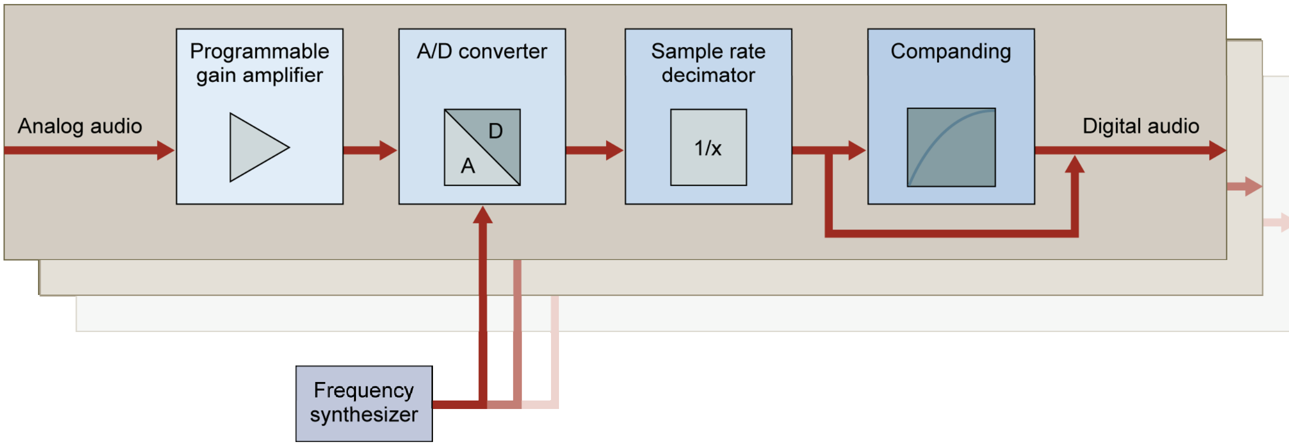
- 44.1 kHz
- 48 kHz

The frequency selection of the master audio clock is common to all the audio channels of a board. All the audio ADC's of the Picolo U4/U8/U16 H.264 are clocked at the same frequency.

The processing chain of each audio channel is composed of the following stages:

- A programmable gain amplifier for audio volume control.
- An analog-to-digital converter clocked at the master audio clock frequency.
- A sample rate decimator for the sample rate control.

- A companding block used exclusively for the generation of G.711 digital audio.




4/8/16 audio channels

## Configurations

### Available configurations of the audio channels

| Sample rate (kHz) | Sample depth (bits) | Companding law | ADC clock |
|---|---|---|---|
| **8** | 8 | µ-law | 48 kHz |
| **8** | 8 | A-law | 48 kHz |
| **8** | 16 | PCM | 48 kHz |
| **16** | 16 | Linear | 48 kHz |
| **22.05** | 16 | Linear | 44.1 kHz |
| **44.1** | 16 | Linear | 44.1 kHz |
| **48** | 16 | Linear | 48 kHz |

Each audio channel is configured individually through the DirectShow API.

The audio sampling frequencies of all the audio channels of a board must belong to one of the following two groups:

- The '48 kHz group' containing the following frequencies: 8, 16, and 48 kHz.
- The '44.1 kHz group' containing the following frequencies: 22.05, and 44.1 kHz.

## Volume Control

Each audio channel has an independent audio volume that can be modified any time. The volume adjusts the gain of the gain of the input amplifier.

The volume is controlled through the DirectShow API.

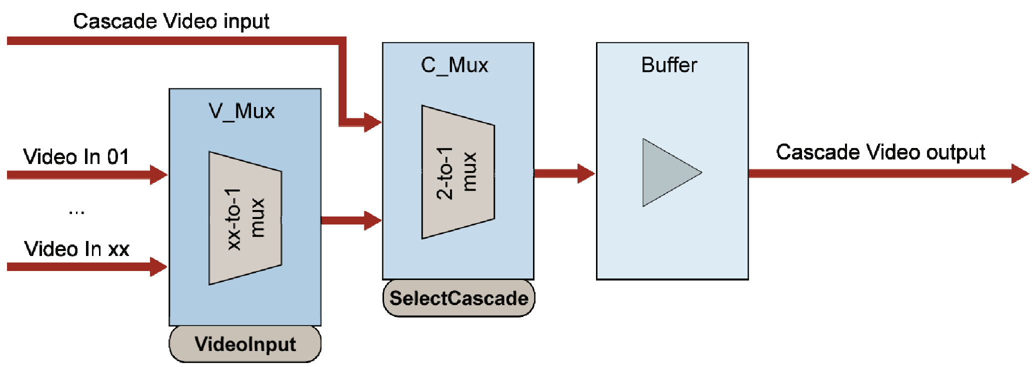
The gain range of the volume control is sufficiently large to accept both "professional" and "consumer" line-level audio signal; however, it is not sufficient to allow direct connection of microphones. Refer to Audio Inputs Electrical Specifications applicable to the audio inputs of Picolo U4/U8/U16 H.264.

In order to optimize the signal-to-noise ratio, the user is invited to adjust the volume such that the highest amplitude of the incoming analog audio produces digital codes just below the maximum value.

The amplifier delivers a level-corrected analog audio signal to the A/D converter.

## 4.6 Video Pass-Through Selector

The video pass-through selector is capable of selecting any of the 4/8/16 video signals applied on **VID_IN1**, ..., **VID_IN4/8/16** and Cascade Video input ports. The selected video signal is then buffered and routed to the Cascade Video output port.




Video pass-through selector functional block diagram (xx = 4/8/16)

The video multiplexer has 2 stages of selectors. The first stage —V_Mux— selects one out of the 4/8/16 video inputs; the second stage —C_Mux— selects either the Cascade Video input, either the output of the first selector stage.

At power-on or reset, both stages revert to their default value: the V-Mux selects **VID_IN01** and the C_Mux selects the Cascade Video Input.

Note that a selection of the Cascade Video input preserves the selection of V_Mux.

**Note.** The buffered Video Cascade allows all video input signals of multiple Picolo U4/U8/U16 H.264 to be multiplexed on a single Video output for connection to a single display monitor.

The video signals are switched asynchronously regardless of the video timing; consequently, a video monitor connected on the Cascade Video output may temporally unlock after a channel switching.

The video input signals are not significantly altered by the channel switching, allowing channel switching during image acquisition.

## 4.7 Format Description

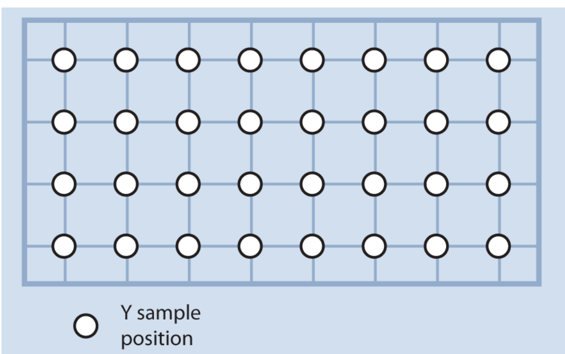
### Y8

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **Y8** | Y800 | PACKED | 1 Bytes/pixel |

#### Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **Y** | 1 | 1 |

#### Spatial map




#### Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | Y | 1 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

| Plane 0 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Word 0** | Pixel 3: Y | | | | | | | | Pixel 2: Y | | | | | | | | Pixel 1: Y | | | | | | | | Pixel 0: Y | | | | | | | |
| **Byte #** | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| **Bit#** | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

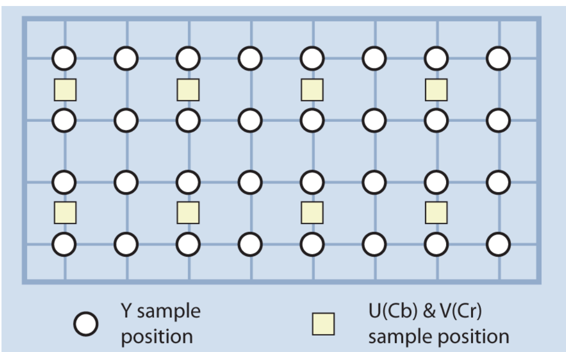
# YUV420Planar

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **YUV420 Planar** | I420, IYUV, YV12 | PLANAR | 1.5 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **Y** | 1 | 1 |
| **U (Cb)** | 2 | 2 |
| **V (Cr)** | 2 | 2 |

## Spatial map




Note:

- The sampling patternof the YUV420format is as specified by the MPEG-2 and the MPEG-4 Part 2 standards.
- The sampling is orthogonal for both the luminance and chrominance samples.
- The chrominance spatial frequency is one half of the luminance spatial frequency in both the horizontal and vertical directions.
- There is one chrominance sample for every quadruplet of luminance samples.
- The position of the chrominance sample is exactly in the center of the rectangle defined by the 4 nearest luminance sampling positions.

## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | Y | 1 Byte/pixel |
| **1** | U | 0.25 Byte/pixel |
| **2** | V | 0.25 Byte/pixel |
| **3** | - | - |

| Plane 0 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | Pixel 3: Y | | | | | | | | Pixel 2: Y | | | | | | | | Pixel 1: Y | | | | | | | | Pixel 0: Y | | | | | | | |
| Byte # | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| Bit# | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Plane 1 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | Pixel 6: U (Cb) | | | | | | | | Pixel 4: U (Cb) | | | | | | | | Pixel 2: U (Cb) | | | | | | | | Pixel 0: U (Cb) | | | | | | | |
| Byte # | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| Bit# | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Plane 2 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Word 0** | Pixel 6: V (Cr) | | | | | | | | Pixel 4: V (Cr) | | | | | | | | Pixel 2: V (Cr) | | | | | | | | Pixel 0: V (Cr) | | | | | | | |
| **Byte #** | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| **Bit#** | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

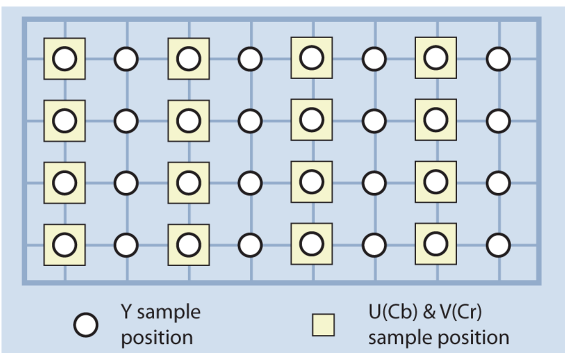
# YUV422 Packed

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **YUV422 Packed** | Y42P, YUYV, YUY2 | PACKED | 2 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **Y** | 1 | 1 |
| **U (Cb)** | 2 | 1 |
| **V (Cr)** | 2 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | YUV | 2 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

| Plane 0 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | Pixel 0: V(Cr) | | | | | | | | Pixel 1: Y | | | | | | | | Pixel 0: U(Cb) | | | | | | | | Pixel 0: Y | | | | | | | |
| Byte # | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| Bit# | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

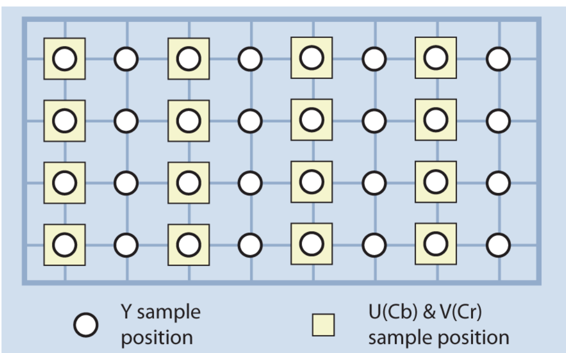
# YUV422 Planar

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **YUV420 Planar** | Y42B | PLANAR | 2 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **Y** | 1 | 1 |
| **U (Cb)** | 2 | 1 |
| **V (Cr)** | 2 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | Y | 1 Byte/pixel |
| **1** | U | 0.5 Byte/pixel |
| **2** | V | 0.5 Byte/pixel |
| **3** | - | - |

| **Plane 0** | **Memory layout** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Word 0** | Pixel 3: Y | | | | | | | | Pixel 2: Y | | | | | | | | Pixel 1: Y | | | | | | | | Pixel 0: Y | | | | | | | |
| **Byte #** | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| **Bit#** | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| **Plane 1** | **Memory layout** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Word 0** | Pixel 6: U (Cb) | | | | | | | | Pixel 4: U (Cb) | | | | | | | | Pixel 2: U (Cb) | | | | | | | | Pixel 0: U (Cb) | | | | | | | |
| **Byte #** | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| **Bit#** | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Plane 2 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | Pixel 6: V (Cr) | | | | | | | | Pixel 4: V (Cr) | | | | | | | | Pixel 2: V (Cr) | | | | | | | | Pixel 0: V (Cr) | | | | | | | |
| Byte # | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| Bit# | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

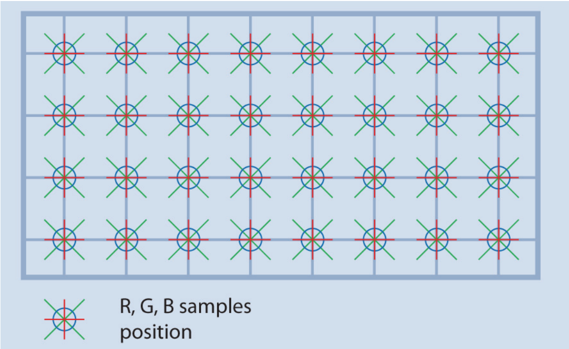
# RGB24 Packed

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **RGB24 Packed** | BI_RGB,RGB | PACKED | 3 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **R** | 1 | 1 |
| **G** | 1 | 1 |
| **B** | 1 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | RGB | 3 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

<table>
<tr><th>Plane 0</th><th colspan="32">Memory layout</th></tr>
<tr><td>Word 2</td><td colspan="8">Pixel 3: R</td><td colspan="8">Pixel 3: G</td><td colspan="8">Pixel 3: B</td><td colspan="8">Pixel 2: R</td></tr>
<tr><td>Word 1</td><td colspan="8">Pixel 2: G</td><td colspan="8">Pixel 2: B</td><td colspan="8">Pixel 1: R</td><td colspan="8">Pixel 1: G</td></tr>
<tr><td>Word 0</td><td colspan="8">Pixel 1: B</td><td colspan="8">Pixel 0: R</td><td colspan="8">Pixel 0: G</td><td colspan="8">Pixel 0: B</td></tr>
<tr><td>Byte#</td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td>Bit#</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

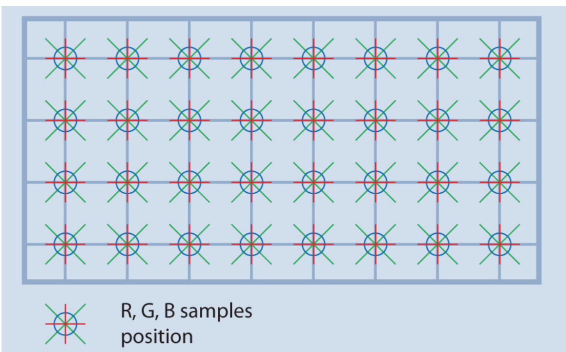
# RGB32 Packed

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **RGB32 Packed** | BI_RGB,RGB | PACKED | 4 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **R** | 1 | 1 |
| **G** | 1 | 1 |
| **B** | 1 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | XRGB | 4 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

<table>
<tr><th>Plane 0</th><th colspan="32">Memory layout</th></tr>
<tr><td>Word 0</td><td colspan="8">"1111 1111"</td><td colspan="8">Pixel 0: R</td><td colspan="8">Pixel 0: G</td><td colspan="8">Pixel 0: B</td></tr>
<tr><td>Byte #</td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td>Bit#</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

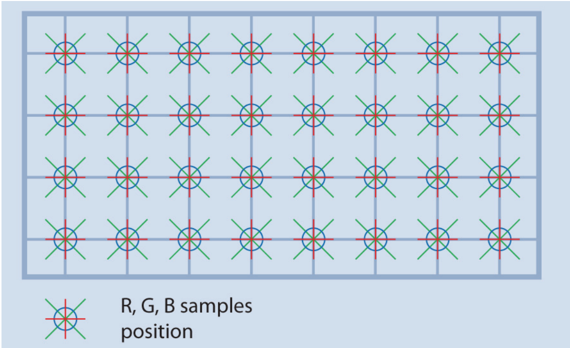
# RGB15

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **RGB15** | BI_RGB,RGB | PACKED | 2 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **R** | 1 | 1 |
| **G** | 1 | 1 |
| **B** | 1 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | RGB | 2 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

<table>
<tr><th>Plane 0</th><th colspan="32">Memory layout</th></tr>
<tr><td>Word 0</td><td>0</td><td colspan="5">Pixel 1:R</td><td colspan="5">Pixel 1:G</td><td colspan="5">Pixel 1:B</td><td>0</td><td colspan="5">Pixel 0:R</td><td colspan="5">Pixel 0:G</td><td colspan="5">Pixel 0:B</td></tr>
<tr><td>Byte #</td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td>Bit#</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

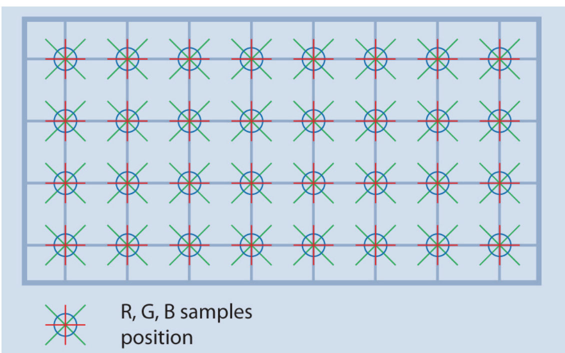
# RGB16

| Format | FourCC | Storage type | Storage requirement |
|---|---|---|---|
| **RGB16** | BI_RGB,RGB | PACKED | 2 Bytes/pixel |

## Spatial sampling periods

| Component | Horizontal | Vertical |
|---|---|---|
| **R** | 1 | 1 |
| **G** | 1 | 1 |
| **B** | 1 | 1 |

## Spatial map




## Plane assignment

| Plane# | Plane name | Storage requirement |
|---|---|---|
| **0** | RGB | 2 Byte/pixel |
| **1** | - | - |
| **2** | - | - |
| **3** | - | - |

| Plane 0 | Memory layout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Word 0 | Pixel 1:R | | | | | Pixel 1:G | | | | | | Pixel 1:B | | | | | Pixel 0:R | | | | | Pixel 0:G | | | | | | Pixel 0:B | | | | |
| Byte # | 3 | | | | | | | | 2 | | | | | | | | 1 | | | | | | | | 0 | | | | | | | |
| Bit # | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

# 5. Board I/O Operation

## 5.1 General Purpose Inputs

The I/O controllers of Picolo U4/U8/U16 H.264 manage:

- 4/8/16 general purpose inputs
- 4/8/16 general purpose outputs
- 1 watchdog output
- 1 temperature monitor

### Configuration

The inputs are initially configured at power-on as follows:

- 4 kΩ impedance threshold
- TTL voltage threshold
- 100 millisecond de-bounce filter off

The initial configuration is suitable for both TTL signaling and contact closure devices. Alternate threshold voltages can be selected, for each input style individually through the DirectShow API.

| Threshold | Threshold voltage | Notes |
|---|---|---|
| **TTL** | 1.5 V | For TTL signaling and contact closure devices. This is the default configuration. |
| **CMOS** | 2.5 V | For 5V CMOS signaling. |
| **12V** | 6.0 V | For 12V signaling. |

Alternate time constants can be selected, for each input filter individually, through the DirectShow API.

| InputFilter | Time constant | Notes |
|---|---|---|
| **Off** | — | Filter is turned off. |
| **10ms** | 10 ms | Filter is turned on with a 10ms time constant . |
| **100ms** | 100 ms | A 100ms filter is very efficient to reject 50/60Hz perturbation. This is the default configuration. |

Alternate impedance thresholds cannot be selected.

### State

The I/O controller measures both the voltage and the impedance across every input at a rate of 200 Hz. It compares the measured values with the threshold and determines the input state according to the following resolution table:

| Measured impedance | Measured voltage | State |
|---|---|---|
| **Below impedance threshold** | **Below voltage threshold** | Low |
| **Below impedance threshold** | **Above voltage threshold** | High |
| **Above impedance threshold** | **Don't care** | Disconnected |

The input state is reported through the DirectShow API. The value must be interpreted differently according to the type of device attached to the input:

| Level | Meaning for contact devices | Meaning for logic devices |
|---|---|---|
| **Low** | Contact is closed. | The logic level is 0. |
| **High** | N/A. | The logic level is 1. |
| **Disconnected** | Contact is open. | The line is disconnected. |

### Events Signaling

Following events can be registered, for each input individually, to report changes on the **State** property:

| StateChange | Meaning |
|---|---|
| **GoLow** | The line is entering the **Low** state. |
| **GoHigh** | The line is entering the **High** state. |
| **GoDisconnected** | The line is entering the **Disconnected** state. |

A signaling event of a particular pin can be register once.

## 5.2 General Purpose Outputs

The I/O controllers of Picolo U4/U8/U16 H.264 manage 4/8/16 general purpose "solid-state relay" outputs, the electrical style is SSRLY —Solid-State Relay.

The state of the output can be changed at anytime through the DirectShow API.

| State | Description |
|---|---|
| **Closed** | The solid-state relay is closed (i.e. it makes a short circuit as its outputs). |
| **Open** | The solid-state relay is open (i.e. it makes an open circuit). |

All solid-state relays of professional outputs and watchdog are placed in the Open state:

- in the I/O controller initialization phase, after a Power On
- after the watchdog I/O controller issues a PC reset

## 5.3 Temperature Monitor

The I/O controllers of Picolo U4/U8/U16 H.264 embed a temperature monitor. An I/O controller performs a board temperature measurement every 1.28 seconds, and reports the result in the Temperature property.

The application should check for regularly temperature and:

- produces a warning message when the temperature exceeds the recommended limit.
- prepares for a system shutdown when the temperature is approaching the absolute limit.

### Overheating Protection

When the temperature monitor detects a temperature sensor malfunction, or measures a temperature exceeding the absolute limit, Picolo U4/U8/U16 H.264 is automatically placed into the reduced power mode.

In the reduced power mode:

- The board power consumption is significantly reduced, which will limit or stop the board temperature rise.
- All acquisitions are stopped and usually cause the application to fail.

The reduced power mode remains until the PC is restarted. The normal operation is resumed after board initialization, providing that the temperature sensor operates normally and the measured board temperature has sufficiently decreased, namely:

- below the recommended limit in case of a PC reset,
- below the absolute limit in case of a Power-On reset.

## 5.4 Watchdog

Maintaining a reliable video-surveillance operation on a PC for several years is difficult. For example, mains electrical perturbations can lead to a failure in the PC using the video-surveillance system. Hence, manual intervention is required to restart it, but this will lead to video-surveillance operation breakdown due to the delay. To ensure that video-surveillance task is continuously monitored, a hardware device, called **watchdog**, is implemented on this frame grabber.

The watchdog is a hardware device which monitors the video-surveillance software application running on the PC. If there is no activity of the software application during a defined time, called **time-out**, the watchdog concludes in a system failure, and closes a PC reset relay which restarts the system.

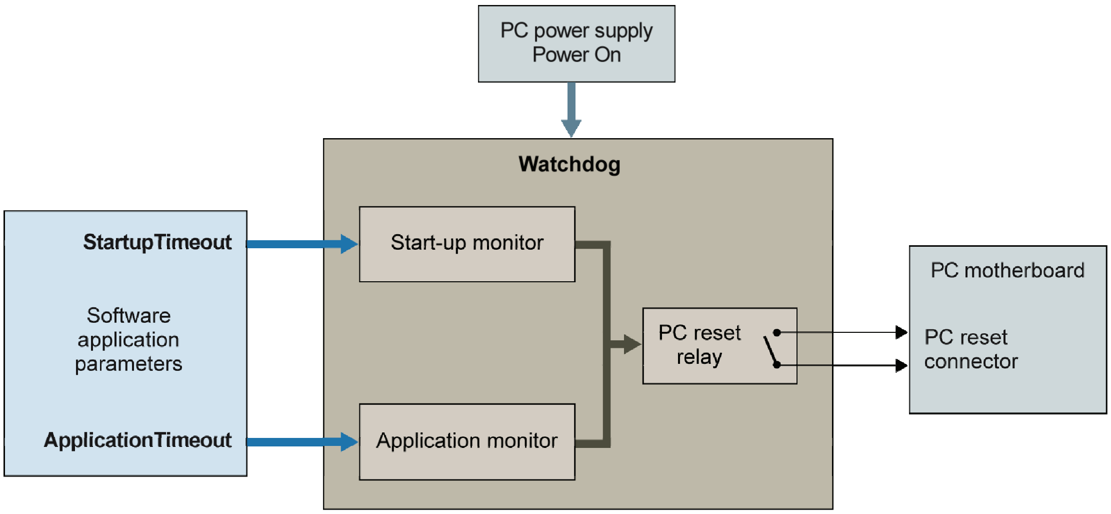
The watchdog consists of two main devices, called monitors:

- The **start-up monitor** is responsible for monitoring the system when the computer is booting, and until the software application is launched.
- The **application monitor** is responsible for monitoring the system during the normal operation of the software application.

The monitors are based on a time-out mechanism. The software application must signal to the monitor that it is operational before a time-out period. If the application does not respond within this time-out period, the monitor will conclude that a system malfunction has occurred.

# Watchdog Block Diagram

The block diagram of the watchdog is shown below.



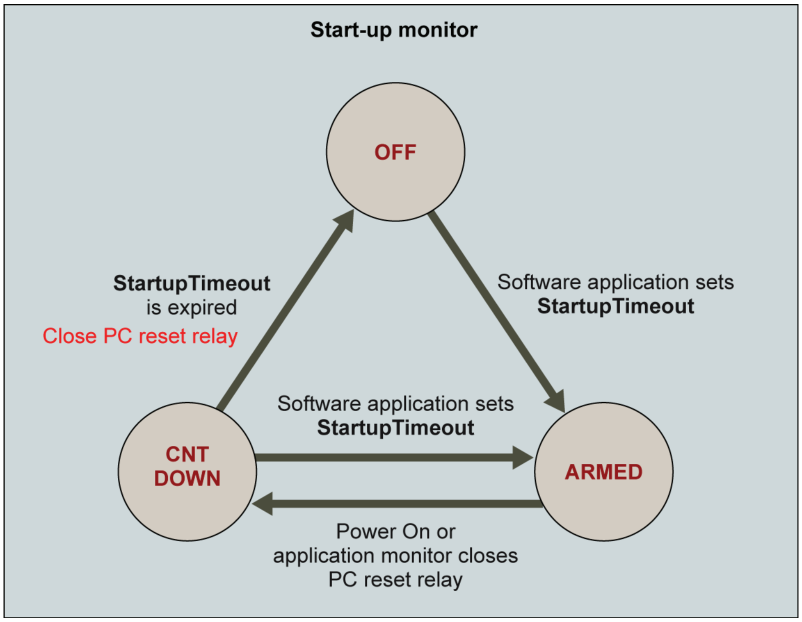

Watchdog block diagram

The above block diagram connects with three inputs and one output:

- Power On: the watchdog requires the power supply of the monitored PC. At Power On, the watchdog is operational, and starts to monitor the system.
- **StartupTimeout**: the software application is responsible to adequately define a maximum time. It allows the system to complete the start-up sequence by setting the **StartupTimeout** property.
- **ApplicationTimeout**: the software application is responsible to adequately define a maximum time. It allows the application to periodically signal its correct operation by setting the **ApplicationTimeout** property.
- PC reset connector: when the watchdog concludes that the start-up has failed, or that the application is no longer running, it restarts the system by closing the PC reset relay. For this purpose, an electrical connection has to be installed between the PC reset relay and the reset connector of the PC motherboard.

# Start-up Monitor

The start-up monitor is graphically represented by a state-device as follows:




Start-up monitor block diagram

The factory preset of the start-up monitor is the **OFF** state. When the software application sets the **StartupTimeout** property, the start-up monitor goes in the **ARMED** state. In the **ARMED** state, the start-up monitor waits for a new start-up sequence. The start-up monitor is able to memorize its state and the **StartupTimeout** property. In other words, its fully operational when power is applied, and before the launch of the software application.

The start-up monitor can be in three states: **OFF**, **ARMED**, or **CNT DOWN**.

## OFF

In the **OFF** state, the start-up monitor is disabled.

## ARMED

In the **ARMED** state, the start-up monitor is waiting for a start-up sequence to monitor. For example, the start-up monitor detects a new start-up sequence when power is applied (Power On), or a start-up sequence can be engaged following a PC reset relay closure generated by the application monitor.

## CNT DOWN

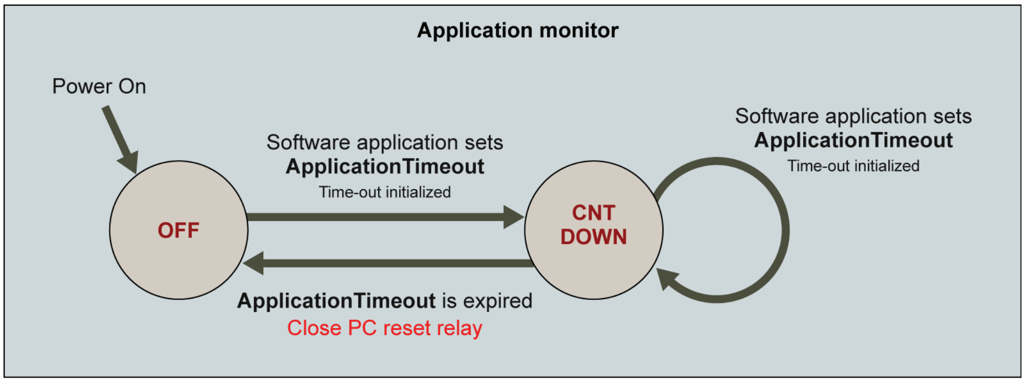
The start-up monitor goes in the **CNT DOWN** state, and begins to count the time that elapses. Two situations may occur in the **CNT DOWN** state:

- The application is working correctly, and sets the **StartupTimeout** property within the time-out expiration. The start-up monitor concludes that the start-up sequence has been completed. The start-up monitor goes in the **ARMED** state, waits for a new start-up sequence, and the application monitor is triggered.

- The start-up is abnormally long, and the application could not set the **StartupTimeout** property before time-out expiration. The start-up monitor concludes that the start-up sequence has not been completed. The start-up monitor goes in the **OFF** state, and closes the PC reset relay for about 1 second.

# *Application Monitor*

The application monitor is graphically represented by a state-device as follows:




**Application monitor block diagram**

The application monitor cannot memorize its state when power is switched off. During Power On, it always starts in the **OFF** state. The application monitor becomes active, and goes in the **CNT DOWN** state under the software application control by simply setting the **ApplicationTimeout** property to a desired value.

The application monitor can be in two states: **OFF**, or **CNT DOWN**.

## OFF

In the **OFF** state, the application monitor is disabled.

## CNT DOWN

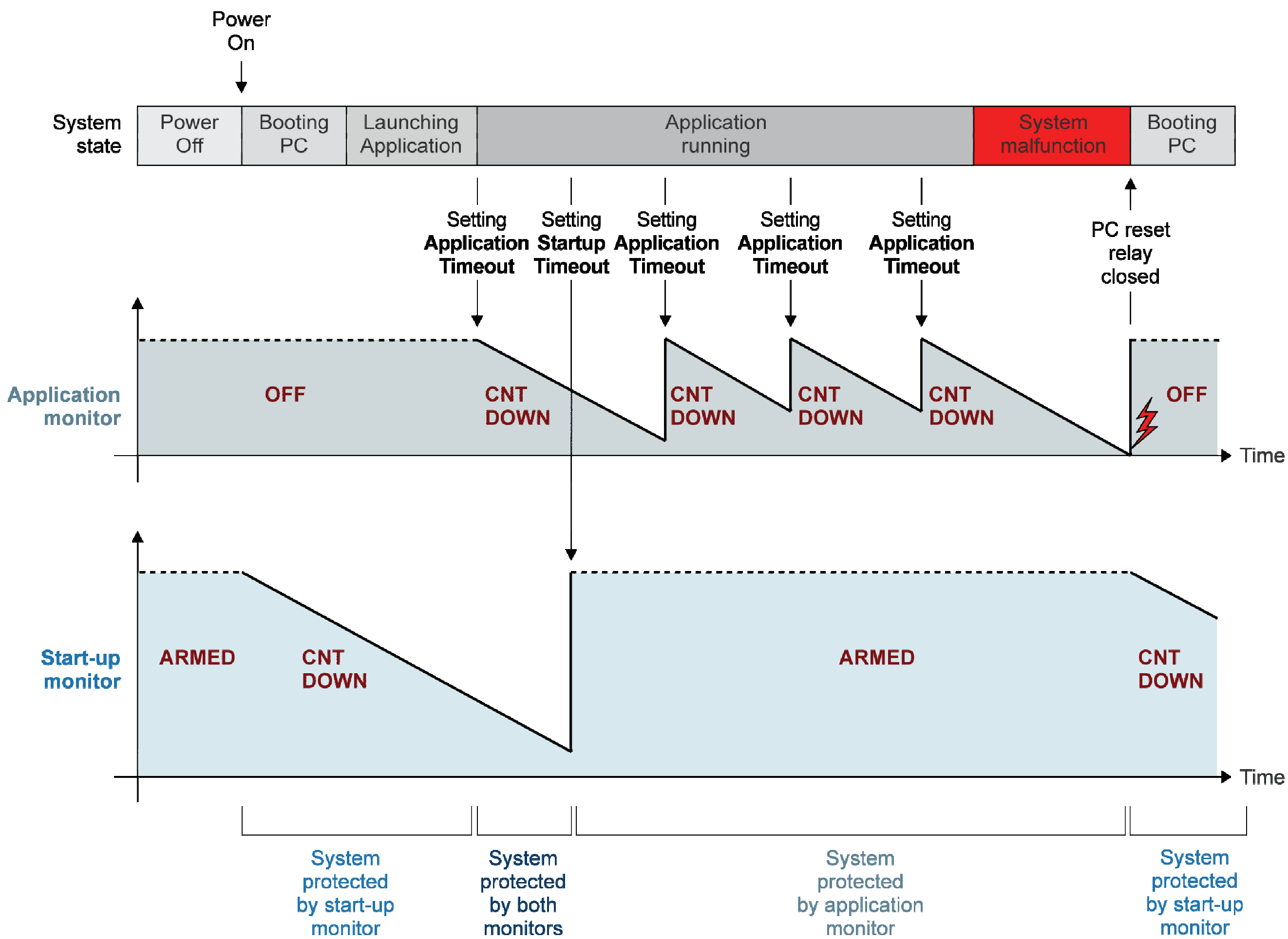
Once in the **CNT DOWN** state, the application monitor waits for a new setting of the **ApplicationTimeout** property. Two situations may occur in the **CNT DOWN** state:

- The software application is alive, and sets the **ApplicationTimeout** property before the time-out expiration. The application monitor concludes that the application is correctly running. The application monitor stays in the **CNT DOWN** state, and waits for the next **ApplicationTimeout** property setting during the time-out period.
- A malfunction occurs, and the application does not run the appropriate code which sets the **ApplicationTimeout** property. Time-out expiration occurs, and the application monitor concludes that the application is no longer running. The application monitor goes in the **OFF** state, and closes the PC reset relay during about 1 second.

# Watchdog Timing Diagram

The implementation of two distinct monitors allows the programmer to define two separate watchdog time-out, one for the start-up phase, and one for the normal operation phase.

The timing diagram of the watchdog operation is shown below.




Watchdog timing diagram

1. It is assumed that the software application has set the **StartupTimeout** property during a previous PC operation. Hence the start-up monitor has memorized an **ARMED** state. When the PC power supply is switched on, the start-up monitor goes in the **CNT DOWN** state.
2. After the PC has booted up, the application is launched and begins its normal operation. At this point, the start-up monitor counting down is stopped when the application sets the **StartupTimeout** property. The start-up monitor goes in the **ARMED** state, and is ready to monitor the next start-up sequence.
3. The application monitor goes in the **CNT DOWN** state when the application sets the **ApplicationTimeout** property.
4. The system is now in normal operation and the application periodically sets the **ApplicationTimeout** property to inform the application monitor that everything is working properly.

5. If a system malfunction occurs —for example, the application is frozen and the internal counter of the application monitor reaches 0 after the time-out period— the application monitor closes the PC reset relay during 1 second, and goes in the **OFF** state. The relay closure generates a reboot of the PC. The start-up monitor changes from the **ARMED** state to the **CNT DOWN** state, and monitors the coming start-up sequence.

To ensure system monitoring without interruption, the software application must perform a initial setting of **ApplicationTimeout** property before **StartupTimeout** property.

## *Watchdog Resets Logging*

The advanced I/O sub-system counts the occurrences of watchdog resets, and delivers the result in the ResetCount property. This is a way to count the number of malfunctions the system has encountered. You can clear this value by using the ClearResetCount property.

# DirectShow Reference

This documentation focuses on the Euresys DirectShow filters. It does not replace Microsoft's DirectShow SDK documentation. A sound knowledge of DirectShow concepts is needed before starting.

# 1. Drivers Model

The device driver of Picolo U4/U8/U16 H.264 is an AVStream streaming mini driver providing several filters. The filters provided by the driver are accessible in DirectShow through **KsProxy**. This makes all boards functionalities accessible in DirectShow.

The kernel streaming filter exposes its configuration API through the **property set** mechanism. There are several **standard** property sets, as well as **custom** property sets.

In DirectShow, the property sets become COM interfaces. Standard property sets are supported by standard COM interfaces and standard property pages (dialog boxes). Custom property sets are supported through custom COM interfaces and custom property pages.

# 2. PICOLO U4/U8/U16 H.264 Configuration

## 2.1 Immediate and Cached Settings

The settings of Picolo U4/U8/U16 H.264 are controlled through properties of interfaces belonging to the Picolo U4/U8/U16 H.264 filters. Two types of settings are considered regarding the timing of application.

In case of **immediate** settings:

- Any change of the corresponding property is immediately applied. The value is stored in non-volatile memory, and restored automatically after power-on reset.
- Reading the property value reports the current state of the setting.

In case of **cached** settings:

- Any change of the corresponding property is cached, it becomes effective the next time the Filter graph leaves the **STOP** state.
- Reading the property value reports the cached value, and **NOT** the actual state of the setting.

Cached settings affect deeply the hardware and/or driver configuration. A reconfiguration of the hardware and driver is required when one or several cached setting has changed.

## 2.2 Filter Graph

Filter Graph is the central component in DirectShow. Applications use it to build and control filter graphs. Filter Graph also handles synchronization, event notification, and other aspects of managing the filter graph.

There are **3 Filter Graph states**:

- **STOP**. No media data is streaming, and the Picolo U4/U8/U16 H.264 filters are idling.
- **PAUSE**. The Picolo U4/U8/U16 H.264 source filters are streaming data.
- **RUN**. The Picolo U4/U8/U16 H.264 source filters are streaming data.

When the Filter Graph state changes from **RUN** or **PAUSE** towards **STOP**, data streaming is stopped in all Picolo U4/U8/U16 H.264 filters present in the graph.

When the Filter Graph state changes from **STOP** towards **PAUSE** or **RUN**, the delivery of records is started in all Picolo U4/U8/U16 H.264 filters present in the graph.

**Filter Property Pages**

There is a property sheet for every programmatic interface.

## 2.3 Graph Reference Clock

Picolo U4/U8/U16 H.264 attaches a time stamp on each Media Sample. The time stamps of all audio and video streams delivered by a Picolo U4/U8/U16 H.264 board are obtained from a local time base, clocked by a stable X-Tal controlled oscillator. The time is expressed in 100 ns unit, and conforms to the DirectShow requirements.

According to the user setting or to the DirectShow decision rules, the graph reference clock can be provided by various time bases, different from the Picolo U4/U8/U16 H.264 local time base (for example, the system clock or an audio renderer clock).

**Note.** Despite the accuracy of the time bases, the time bases are desynchronizing progressively (long-term drifting), and significant differences in the time measurements can appear when running the graph over a long period.

Picolo U4/U8/U16 H.264 automatically compensates the long-term drift of the time bases. Therefore, it measures the differences that can exist between the graph reference clock and the local clock used to perform time stamp, and maintains this difference below 100 ms by smoothly adjusting the local time base on the graph reference clock. This compensation is performed independently for each pin.

This long-term drift compensation is designed to operate correctly if the accuracy of the graph reference clock is within +/-1%.

## 2.4 Batch Installation

The following procedures shows how to automate the installation of the driver on several target machines by using command line.

1. Install driver on one machine using the supplied installation tool. Run the installer tool and follow on-screen instructions.
2. From the installation directory, navigate to the following files and directories: **/setup** directory, **/driversDS** directory, **install.bat** file and **uninstall.bat** file. Use these files to make your own package.
3. Copy your own package on the target machine in a directory of your choice.
4. On the target machines, run install.bat for the installation of the driver. Un-install the driver by using install.bat.

# 3. DirectShow Filters

The DirectShow® Interface of the DirectShow® drivers exposes all functionalities of the boards through a collection of DirectShow® Filters.

The transport and delivery of media data are performed via DirectShow standard means such as Pin Interface, Media Sample Interface, etc.

The configuration settings and the control of Picolo U4/U8/U16 H.264 features are done through a set of both standard and custom COM interfaces and property pages.

## 3.1 Filters Instantation (UxH264)

The number of operable filters is limited by Picolo U4/U8/U16 H.264 hardware resources. The filter creation fails and reports and error when the operable instances count is exceeded.

**Operable instances count for each filter**

| Filter | Maximum operable instances count |
|---|---|
| **Visual Source** | Two per Video Input(*) |
| | Three per Video Input(**) |
| | One per Audio Input |
| | One per Professional Input |
| | One per Professional Output |
| | One per board |
| | One per board |
| | N/A(***) |

(*) A graph may contain only one instance of visual source filter per video input.Two instances of visual source filter for the same video Input can be created in separate graphs providing that the encoded pin and formatted pin are connected only once.

(**) A graph may contain up to 3 instances of the Visual Source Filter per Video Input. Three instances of the Visual Source Filter for the same Video Input can be created in the same graph or in separate graphs provided that the Encoded pin is not connected more than two times and that the Formatted Pin is connect only once. The second encoded stream of a Video Input may not be configured for the 4CIF and the 2CIF resolution.

(***) The maximum number of board filter instances is solely limited by system resources.

# 3.2 Visual Source Filter

## Encoded Pins

The **Encoded** output pin streams data in H.264 format. It exhibits **MEDIATYPE_Video** and following **Media_SubTypes**:

- **FourCC H264**
- **FourCC h264**
- **FourCC AVC1**
- **FourCC avc1**

Choosing any of the H.264 variants enables connection with various third-party filters supporting H.264, and does not affect the encoder settings.

The Encoded pin has one custom interface, IUxH264VisualEncoder, and has also a custom property page.

### Standard Pin Interfaces

- IAMBufferNegotiation
- IAMStreamConfig
  - Scaling
  - Format
- IAMStreamControl
- IksPin
- IKsPropertySet
- IMediaSeeking
- IPin
- IQualityControl
- ISpecifyPropertyPages

### Notes

- Method **IMediasample::IsSyncPoint** returns true on frame that contains IDR NAL unit, false otherwise.
- The SPS, PPS, and IDR NAL units are merged into the one single media sample.
- 4CIF and the 2CIF resolutions are not allowed on the second encoded stream

## Formatted Pins

The **Formatted** output pin streams data in several uncompressed video formats. It exhibits **MEDIATYPE_Video** and following **Media_SubTypes**:

| MEDIASUBTYPE | FourCC | Euresys format name | Description | Ranking (**) |
|---|---|---|---|---|
| **MEDIASUBTYPE_I420** | I420 | YUV420PL | YUV 4:2:0 video stored in planar format, with the following plane order: Y,U, and V. | 1 |
| **MEDIASUBTYPE_YV12** | YV12 | YUV420PL | YUV 4:2:0 video stored in planar format, with the following plane order: Y,U, and V.<br>YV12 is identical to I420 except that the U and V planes are swapped. | 2 |
| **MEDIASUBTYPE_RGB32** | N/A | RGB32 | ARGB video in 8-8-8-8 Packed format | (***) |
| **MEDIASUBTYPE_RGB24** | N/A | RGB24 | RGB video in 8-8-8 Packed format | (***) |
| **MEDIASUBTYPE_RGB565** | N/A | RGB16 | RGB video in 5-6-5 Packed format | (***) |
| **MEDIASUBTYPE_RGB555** | N/A | RGB15 | RGB video in 5-5-5 Packed format | (***) |
| **MEDIASUBTYPE_RGB8 (*)** | Y800 | Y8 | 8-bit monochrome video. | 3 |
| **MEDIASUBTYPE_YUY2** | YUY2 | YUY422 | YUV 4:2.2 video stored in Packed format | (***) |

(*) MEDIASUBTYPE_RGB8 operates from the Color Space Converter delivered by Microsoft.

(**) It is recommended to select a format with the lowest ranking value.

(***) The RGB video formats are not recommended on Picolo Ux Series

The media type is negotiated by the pins at connection time. Setting the format through the custom pin interfaces restricts the possible formats to the selected one. The Formatted pin has one custom interface, IUxH264VisualEncoder, and has also a custom property page.

### Standard Pin Interfaces

- IAMBufferNegotiation
- IAMStreamConfig
  - Scaling
  - Format
- IAMStreamControl
- IKsPin
- IKsPropertySet
- IMediaSeeking
- IPin
- IQualityControl
- ISpecifyPropertyPages

# Standard Interfaces (Visual Source Filter)

## KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## Other standard interfaces

| Interface | Description | Notes |
|---|---|---|
| **IAMAnalogVideoDecoder** | Controls video digitization on a WDM video capture card. | The **put_TVFormat** method sets the analog video format.. Accepts following standards in the **AnalogVideoStandard** enumeration: **NTSC_M**, **PAL_B**, **PAL_D**, **PAL_H**, **PAL_I**.<br>These methods are not supported:<br>**get_HorizontalLocked**<br>**get_VCRHorizontalLocking**<br>**get_OutputEnable**<br>**put_VCRHorizontalLocking**<br>**put_OutputEnable** |
| **IAMVideoProcAmp** | Adjusts the qualities of a video signal, such as brightness, contrast, and saturation. | Also available in **IUxH264VisualSource** interface. Both interfaces need to be maintained in sync. |
| **IAMVideoCompression** | Sets and retrieves video compression properties. | |
| **ICodecAPI** | Configures an encoder or decoder. | |

# IUxH264VisualSource Custom Interface

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **Camera** | Camera attached to the input pin. (*) | Integer, Set(Get) | **[U4:0...3, U8:0...7, U16:0...15 ]** | VIDEO/VIDEO IN Connectors |
| **DetectedStandard** | Reports the detected standard of the video signal. | Enum, Get Only | **PAL**<br>**NTSC**<br>**CCIR**<br>**EIA** | Auxiliary Functions |
| **SignalDetected** | Reports the video signal detection state | Bool, Get Only | | Auxiliary Functions |
| **Contrast** | Luminance gain (%). Default: **100**. (**)(***) | Int, Set(Get) | **[0..200]** | Luminance Processing |
| **Brightness** | Luminance offset (%). Unit is one gray level. Default: **0**. (**)(***) | Int, Set(Get) | **[-100..100]** | Luminance Processing |
| **Saturation** | Color saturation (%). Default: **100**. (**)(***) | Int, Set(Get) | **[0..200]** | Chrominance Processing |
| **Caption** | Caption settings in the encoded stream. (**)<br>The caption text is inserted simultaneously on formatted and compressed streams.<br>See details in the Caption property table below. | | | Caption Insertion |
| **Masks** | Masks collection in the encoded stream. (**)<br>The privacy mask is inserted simultaneously on formatted and compressed streams.<br>See details in the Masks property table below. | | | Mask Insertion |
| **NumberOfInputs** | Reports the maximum number of visual sources per board | Data: Integer, Get Only | **U4:4, U8:8, U16:16** | |
| **NumberOfCaptions** | Reports the maximum number of captions per visual source | Data: Integer, Get Only | **5** | |

(*) Cached setting.

(**) Immediate setting that can be changed at any time.

(***) Contrast, Brightness, and Saturation and also available through the IAMVideoProAmp standard interface.

### Caption property

| Property | Description | Type, access | Value range |
|---|---|---|---|
| **Caption.Data** | Index of the caption line. Two lines of caption text are independently configurable. | Enum, Set Only | **[CAPTION_INDEX_0, .CAPTION_INDEX_1, CAPTION_INDEX_2, CAPTION_INDEX_3, CAPTION_INDEX_4]** |
| **Caption.StringBuffer** | Caption text. An empty string inhibits the caption insertion. | Enum, Set(Get) | **<string of up to 47 7-bit ASCII characters>** |
| **Caption.PositionType** | Selects the type of the position definition. | Enum, Set(Get) | **[CAPTION_POSITION_TOP_LEFT = 0, CAPTION_POSITION_BOTTOM_LEFT, CAPTION_POSITION_TOP_RIGHT, CAPTION_POSITION_BOTTOM_RIGHT, CAPTION_POSITION_CUSTOM]** |
| **Caption.X** | X-coordinate of the upper-left corner pixel of the leftmost character. (The coordinate [0, 0] is the upper left corner of the full image.)<br>Relevant only when **PositionType=Custom**. | Integer, Set(Get) | **[0..703]** |
| **Caption.Y** | Y-coordinate of the upper-left corner pixel of the leftmost character. (The coordinate [0, 0] is the upper left corner of the full image.)<br>Relevant only when **PositionType=Custom**. | Integer, Set(Get) | **[0..575]** |

The above properties are Immediate setting that can be changed at any time.

**Notes:**

- Each line of caption text is independently configurable
- Caption text is inserted simultaneously on formatted and compressed stream
- Coordinates are expressed in pixel units of the full-res image,coordinates origin is the top-left corner of the image
- Properties X and Y are relevant only with custom position type

### Masks property

| Property | Description | Type, access | Value range |
|---|---|---|---|
| **Masks.Data** | Index of the mask region. Four privacy regions are independently configurable. | Enum, Set Only | **[0..3]** |
| **Masks.X** | X-coordinate of the upper-left corner of the rectangular masked region. (The coordinate [0, 0] is the upper left corner of the full image.) (*) | Integer, Set(Get) | **[0..703]** |
| **Masks.Y** | Y-coordinate of the upper-left corner of the rectangular masked region. (The coordinate [0, 0] is the upper left corner of the full image.) (*) | Integer, Set(Get) | **[0..575]** |
| **Masks.Width** | Width of the rectangular masked region. A width of 0 inhibits the mask insertion. (*) | Integer, Set(Get) | **[0..704]** |
| **Masks.Height** | Height of the rectangular masked region. A height of 0 inhibits the mask insertion. (*) | Integer, Set(Get) | **[0..576]** |

The above properties are Immediate setting that can be changed at any time.

(*) The mask area is automatically modified to the smallest enclosing region which fits the 16x16 macroblocks border.

| Event | Description | Related topics |
|---|---|---|
| **KSEVENT_VisualSource_SignalDetect** | Informs on video signal status change. | Auxiliary Functions |

## Code Snippets

Refer to Naming Conventions before setting up your code.

### Camera Index Property

```
//////////////////////////////////////////////////////////
// Set the camera index property                        //
// cameraIndex holds value to be written                //
// pFilter is a pointer to the Ux H.264 Filter          //
//////////////////////////////////////////////////////////

// Declaration
IUXH264VisualSource *pVS;
PICOLO_PROPERTY Control;

// Get the VisualSource interface
pFilter->QueryInterface( CLSID_IUXH264VisualSource, (void **)&pVS );

// Fill control structure
Control.Property.Id = KSPROPERTY_VisualSource_Camera;
Control.PropertySetting.Data = cameraIndex;
```

```
// Set the camera property according the to control structure value
pVS->Set(&Control);

// Release interface
pVS->Release();
```

**Caption Property**

```
//////////////////////////////////////////////////////////////////////
// Setting the Caption property                                     //
// captionIndex holds the Caption index                             //
// captionPosition holds the Caption position(if used)              //
// captionX holds the x position where the text should be displayed //
// captionY holds the y position where the text should be displayed //
// captionText holds the text to be displayed                       //
// pFilter is a pointer to the Ux H.264 Filter                      //
//////////////////////////////////////////////////////////////////////

// Declaration
IUXH264VisualSource *pVS;
PICOLO_PROPERTY Control;

// Get the VisualSource interface
pFilter->QueryInterface( CLSID_IUXH264VisualSource, (void **)&pVS );

// Fill control structure
Control.Property.Id = KSPROPERTY_VisualSource_Caption;
Control.PropertySetting.Data = captionIndex;
Control.PropertySetting.PositionType = captionPosition;
Control.PropertySetting.X = captionX;
Control.PropertySetting.Y = captionY;

// Hardware does not accept unicode strings.
StringCbCopyA( Control.PropertySetting.StringBuffer, sizeof(
Control.PropertySetting.StringBuffer ), captionText );

// Set the caption property according to the control structure values
pVS->Set(&Control);

// Release interface
pVS->Release();
```

### Masks Property

```
//////////////////////////////////////////////////////////////
// Setting the Mask property                               //
// maskIndex holds the Mask index                          //
// maskX holds the x position top left corner of the mask  //
// maskY holds the y position top left corner of the mask  //
// maskWidth holds width of the mask                       //
// maskHeight holds height of the mask                     //
// pFilter is a pointer to the Ux H.264 Filter             //
//////////////////////////////////////////////////////////////

// Declaration
IUXH264VisualSource *pVS;
PICOLO_PROPERTY Control;

// Get the VisualSource interface
pFilter->QueryInterface( CLSID_IUXH264VisualSource, (void **)&pVS );

// Fill control structure
Control.Property.Id = KSPROPERTY_VisualSource_Mask;
Control.PropertySetting.Data = maskIndex;
Control.PropertySetting.X = maskX;
Control.PropertySetting.Y = maskY;
Control.PropertySetting.Width = maskWidth;
Control.PropertySetting.Height = maskHeight;

// Set the mask property according to the control structure values
pVS->Set(&Control);

// Release interface
pVS->Release();
```

## *IUxH264VisualEncoder Custom Interface*

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **Decimation** | Frame rate decimation process in the encoded stream. Default: **1**. (**) | Int, Set(Get) | **[1..150]** | Frame Rate Control (Video Encoder) |
| **BitRateControl** | Bit rate control method of the encoder. Default: **CPQ**. (*) | Enum, Set(Get) | **CPQ** **CBR** **VBR** | Encoder Bit Rate Control |
| **Quality** | Quality level of the picture. Default: **30**. (*) | Int, Set(Get) | **[1..100]** | Encoder Bit Rate Control |
| **AverageKBitRate** | Average bit rate of the encoder. Default: **2000**. (*) | Int, Set(Get) | **[0..4000]** | Encoder Bit Rate Control |
| **MaximumKBitRate** | Maximum bit rate of the encoder. Default: **4000**. (*) | Int, Set(Get) | **[0..4000]** | Encoder Bit Rate Control |
| **GOPSize** | Number of frames per group. Default: **30**. No specified upper limit. (*) | Int, Set(Get) | **[1..30..128]** | Encoder Bit Stream Controls |

(*) Cached setting.

(**) Immediate setting that can be changed at any time.

## *IUxH264VisualFormatter Custom Interface*

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **Decimation** | Frame rate decimation process of the formatted stream. Default: **1**. (*) | Int, Set(Get) | **[1..150]** | Frame Rate Control (Video Formatter) |
| **FlipVertical** | Controls the top/down image flippinf. Default: **OFF**. | Enum, Set(Get) | **[OFF,ON]** | |

(*) Immediate setting that can be changed at any time.

# *3.3 Audio Source Filter*

## *Encoded Pins (Audio Encoder)*

The **Encoded** output pin streams audio data in PCM. It exhibits **MEDIATYPE_Audio**.

### Standard Pin Interfaces

- IAMBufferNegotiation
- IAMStreamConfig
  - Available media subtypes are:
    - **MEDIASUBTYPE_ALAW**
    - **MEDIASUBTYPE_muLAW**
    - **MEDIASUBTYPE_pcm**
- IAMStreamControl
- IKsPin
- IKsPropertySet
- IMediaSeeking
- IPin
- IQualityControl
- ISpecifyPropertyPages

The audio sampling rate is configured through the IAMStreamConfig interface

Picolo U4/U8/U16 H.264 supports the following audio configurations:

- 8-bit µ-Law companded audio at 8 kHz (G.711)
- 8-bit A-Law companded audio at 8 kHz (G.711)
- 16-bit linear PCM audio at 8 kHz
- 16-bit linear PCM audio at 16 kHz
- 16-bit linear PCM audio at 22.05 kHz
- 16-bit linear PCM audio at 44.1 kHz
- 16-bit linear PCM audio at 48 kHz.

The default audio configuration is 8-bit A-law 8 kHz

Each audio source can be configured individually. However, the following restriction applies:

The audio sampling frequencies of all the audio sources of a board must belong to one frequency groups. There are two groups:

- The '48 kHz group' contains the following frequencies: 8, 16, and 48 kHz.
- The '44.1 kHz group' contains the following frequencies: 22.05, and 44.1 kHz.

# Standard Interfaces (Audio Source Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

### Other standard interfaces

| Interface | Description | Notes |
|---|---|---|
| **IBasicAudio** | Set audio output volumes | Balance control is not applicable.<br>Volume control is also available through the customer filter property **Volume**. |

# IUxH264AudioSource Custom Interface

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **Input** | Audio input sourcing the audio encoder. Default: **1**.(*) | Integer, Set(Get) | **[U4:0...3, U8:0...7, U16:0...15 ]** | AUDIO Connector |
| **Volume** | Volume (%). Default: **100**. (**) | Int, Set(Get) | **[25..275]** | Audio Digitizer and Encoder |
| **NumberOfInputs** | Reports the maximum number of audio sources per board | Data: Integer, Get Only | **[U4:4, U8:8, U16:16]** | |

(*) Cached setting.

(**) Immediate setting that can be changed at any time.

# 3.4 Input Line Filter

## Standard Interfaces (Input Line Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## IUxH264InputLine Custom Interface

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **Input** | Input line sourcing the input controller. Default: **1**. (*) | Integer, Set(Get) | **[U4:0...3, U8:0...7, U16:0...15 ]** | General Purpose Inputs |
| **State** | State of the input line. | Enum, Get Only | **Low**<br>**High**<br>**Disconnected** | General Purpose Inputs |
| **InputFilter** | Time constant of the debouncing filter. Default: **_100ms**. (**) | Enum, Set(Get) | **Off**<br>**_10ms**<br>**_100ms** | General Purpose Inputs |
| **InputStyle** | Electrical style of the input line. Default: **TTL**. | Enum, Set(Get) | **TTL**<br>**CMOS**<br>**_12V** | General Purpose Inputs |
| **NumberOfInputs** | Reports the maximum number of input lines per board | Data: Integer, Get Only | **[U4:4, U8:8, U16:16]** | |

(*) Cached setting.

(**) Immediate setting that can be changed at any time.

| Event | Description |
|---|---|
| **KSEVENT_InputLine_GoLow** | Informs on line state going **Low**. |
| **KSEVENT_InputLine_GoHigh** | Informs on line state going **High**. |
| **KSEVENT_InputLine_GoDisconnected** | Informs on line state going **Disconnected**. |

# 3.5 Output Line Filter

## Standard Interfaces (Output Line Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## IUxH264OutputLine Custom Interface

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **Output** | Output line driven by the output controller. Default: **1**. (*) | Integer, Set(Get) | **[U4:0...3, U8:0...7, U16:0...15 ]** | PROFESSIONAL I/O connectors |
| **State** | Output state. Default: **Open**. (*) | Enum, Set(Get) | **Closed Open** | General Purpose Outputs |
| **NumberOfOutputs** | Reports the maximum number of output lines per board | Data: Integer, Get Only | **[U4:4, U8:8, U16:16]** | |

(*) Immediate setting that can be changed at any time.

# 3.6 Watchdog Filter

## Standard Interfaces (Watchdog Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## IUxH264Watchdog Custom Interface

Refer to Naming Conventions before setting up your code.

The watchdog interface properties comply with the standard Euresys watchdog API. This interface controls only the watchdog for a single board.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **StartupTimeout** | Start-up time-out duration (seconds). (*) | Int, Set(Get) | **[1..65535]** | Watchdog |
| **ApplicationTimeout** | Application time-out (seconds). (*) | Int, Set(Get) | **[1..65535]** | |
| **ResetCount** | Number of watchdog-initiated resets. (*) | Int, Get(Set) | **[0..65535]** | |

(*) Immediate setting that can be changed at any time, it's value is stored in Non-volatile memory and restored automatically after power-on reset.

# 3.7 Pass-Through Selector Filter

## Standard Interfaces (Pass-Through Selector Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## IUxH264PassThroughSelector Custom Interface

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **BoardIndex** | Board index in the system, from 0 on. | Integer, Get Only | **[0..]** | Board Filter |
| **SelectCascade** | If **TRUE**, the Cascade Video Input is selected. Default: **TRUE**. (*) | Bool, Set(Get) | **TRUE** **FALSE** | Video Pass-Through Selector |
| **VideoInput** | Video Input to pass through (if **SelectCascade** is **FALSE**). | Integer, Set(Get) | **[U4:0...3, U8:0...7, U16:0...15 ]** | Video Pass-Through Selector |
| **NumberOfInputs** | Reports the maximum number of video inputs per board | Data: Integer, Get Only | **[U4:4, U8:8, U16:16]** | |

(*) Immediate setting that can be changed at any time.

# 3.8 Board Filter

## Standard Interfaces (Board Filter)

### KsProxy standard interfaces

| Interface | Description |
|---|---|
| **IAMFilterMiscFlags** | Queries whether a filter is a source filter or a renderer. |
| **IBaseFilter** | Is the primary interface for all DirectShow filters. |
| **IKsPropertySet** | Sets properties on a kernel-mode filter. |
| **IMediaFilter** | Controls the streaming state of a filter |
| **IPersist, IPersistStream** | Saves and loads objects that use a simple serial stream for their storage needs. |
| **ISpecifyPropertyPages** | Indicates that an object supports property pages. |

## IUxH264Board Custom Interface

Refer to Naming Conventions before setting up your code.

| Property | Description | Type, access | Value range | Related topics |
|---|---|---|---|---|
| **Position** | Position of the board in the system (*) | Integer, Get Only | | |
| **SerialNumber** | Board serial number. (**) | String, Get Only | String of 16 digits corresponding to the serial number label | |
| **PartNumber** | Board part number. (**) | String, Get Only | String of 16 digits corresponding to the part number label | |
| **Type** | Reports the board type. | String, Get Only | **Picolo U4 H.264, Picolo U8 H.264, Picolo U16 H.264** | |
| **Temperature** | Reports the board temperature (°C). | Integer, Get Only | **[0...149]** | Temperature Monitor |

(*) An index, from 0 on, in the list of Picolo U4/U8/U16 H.264 boards enumerated by the driver.

(**) The serial and part numbers are assigned at factory. Their combination is unique for a board of a given type.

# Resources Identification (Software)

The resources are split into the following classes:

- Video resources including
  - Video inputs
  - Video cascade input
  - Video output
- Audio resources
- General purpose inputs resources
- General purpose outputs resources

For both the VFS and the DirectShow drivers, the resources are identified by a zero-based index. Example, on the Picolo U8 H.264 board, the indexes 0… 7 designate a particular resource of one class.

# 1. Video Resources

| Function | Driver Identification | | | Video Terminators Switch# | |
|---|---|---|---|---|---|
| | Picolo U16 | Picolo U8 | Picolo U4 | Group | Sw# |
| Video Inputs | 0 | | | I | 8 |
| | 1 | | | I | 4 |
| | 2 | 1 | 0 | II | 8 |
| | 3 | 0 | | II | 4 |
| | 4 | | | I | 7 |
| | 5 | | | I | 3 |
| | 6 | 3 | 1 | II | 7 |
| | 7 | 2 | | II | 3 |
| | 8 | | | I | 6 |
| | 9 | | | I | 2 |
| | 10 | 5 | 2 | II | 6 |
| | 11 | 4 | | II | 2 |
| | 12 | | | I | 5 |
| | 13 | | | I | 1 |
| | 14 | 7 | 3 | II | 5 |
| | 15 | 6 | | II | 1 |

Legend:

- Driver Identification: this is the index in the application program designating a video input resource.

**Note** : Each resource requires a pin pair on every connector: one pin for the signal, one pin for the return path of the signal.

# 2. Audio Resources

| Function | Zero-based Index | | | FTSH34 PH34 Pins # | |
|---|---|---|---|---|---|
| | Picolo U16 | Picolo U8 | Picolo U4 | Sig. | .Ret. |
| Audio Inputs | 0 | | | 3 | 4 |
| | 1 | | | 11 | 12 |
| | 2 | 1 | 0 | 19 | 20 |
| | 3 | 0 | | 27 | 28 |
| | 4 | | | 5 | 6 |
| | 5 | | | 13 | 14 |
| | 6 | 3 | 1 | 21 | 22 |
| | 7 | 2 | | 29 | 30 |
| | 8 | | | 7 | 8 |
| | 9 | | | 15 | 16 |
| | 10 | 5 | 2 | 23 | 24 |
| | 11 | 4 | | 31 | 32 |
| | 12 | | | 9 | 10 |
| | 13 | | | 17 | 18 |
| | 14 | 7 | 3 | 25 | 26 |
| | 15 | 6 | | 33 | 34 |

Legend:

- Driver Identification: this is the index in the application program designating an audio resource

Note :Each resource requires a pin pair on every connector: one pin for the signal, one pin for the return path of the signal.

# 3. IO Resources

| Function | Zero-based Index | | |
|---|---|---|---|
| | U16 | U8 | U4 |
| **Inputs 1-8** | 0 | 0 | 0 |
| | 1 | 1 | 1 |
| | 2 | 2 | 2 |
| | 3 | 3 | 3 |
| | 4 | 4 | |
| | 5 | 5 | |
| | 6 | 6 | |
| | 7 | 7 | |
| **Outputs 1-8** | 0 | 0 | 0 |
| | 1 | 1 | 1 |
| | 2 | 2 | 2 |
| | 3 | 3 | 3 |
| | 4 | 4 | |
| | 5 | 5 | |
| | 6 | 6 | |
| | 7 | 7 | |
| **Inputs 9-16** | 8 | | |
| | 9 | | |
| | 10 | | |
| | 11 | | |
| | 12 | | |
| | 13 | | |
| | 14 | | |
| | 15 | | |
| **Outputs 9-16** | 8 | | |
| | 9 | | |
| | 10 | | |

| | | | |
|---|---|---|---|
| | 11 | | |
| | 12 | | |
| | 13 | | |
| | 14 | | |
| | 15 | | |

# Glossary

## A

### Absolute audio power levels

Can be expressed using the dBm logarithmic scale. The 0 dBm level is defined as an absolute power level of 1 mW. To convert watts into dBm, use the formula:

```
PowerdBm = 10 * log(1000 * PowerWatt)
```

Note that for a 600 Ω load, dBu level = dBm level.

### Absolute audio voltage levels

Independently of the load impedance, can be expressed using the dBu logarithmic scale. The 0 dBu level is defined as an absolute voltage level that produces 1 mW of power across a 600 Ω load. To convert Vrms into dBu, use the formula:

```
VoltagedBu = 20 * log(VoltageVoltRMS / 0.7746)
```

Can also be expressed using the dBV logarithmic scale. The 0 dBV level is defined as an absolute voltage of 1 Vrms.

### Analog

A type of signal in an electronic circuit that takes on a continuous range of values.

### Audio digitizer

Digitizes an analog audio signal, and delivers a digital audio stream.

## *B*

### Bandwidth

A measurement of the rate of data transfer, in bits or bytes per second.

### Back-porch clamping

During the back-porch interval (between the sync tip and the begin of a video line) of the video signal, also known as "blank level", the DC-restoration circuit maintains the blank level at a constant DC voltage, using a servo-loop that minimizes the error.

### Brightness

In NTSC and PAL video signals, the brightness information at any particular instant in a picture is conveyed by the corresponding instantaneous DC level of active video.

### Bit

The smallest unit of information, that can be represented either as 1 or 0.

### Byte

Consists of 8 bits. Values from 0 to 255, commonly used to represent the gray-scale value of a pixel.

## *C*

### CCIR

CCIR is the standard monochrome video format used in most of Europe, Israel, and some other places in the world. CCIR products are also generally referred to as PAL because all PAL products can also handle black and white CCIR video.

### Chrominance

A signal that represents the color information of an image.

### CIF formats

CIF formats, commonly used in video teleconferencing systems for specific resolution, 352 x 288 for PAL and 352 x 240 for NTSC.

### Contrast

The difference between the darkest and the brightest parts of an image.

### COM

The Component Object Model was introduced by Microsoft in 1993. It is an object-oriented programming model, used to enable inter-process communication and dynamic object creation, in any programming language that supports the technology.

### Comb filter

An electrical filter circuit that passes a series of frequencies and rejects the frequencies in between, it combs out the frequencies. Used on a composite video signal, to separate the chrominance signal and reject the luminance signal, or to select the luminance signal and reject the chrominance signal.

### Cropping

Remove any unwanted areas in an image.

### Cross luminance

See Dot crawl.

## D

### D-1 resolution

D-1 is a resolution standard for TV specifications. It means 720 x 486 in NTSC systems, and 720 x 576 in PAL and SECAM systems.

### dBm

dB(1 mW)

A dBm is a standard unit for measuring levels of power in relation to a 1 milliwatt reference signal. Similar to dB, except that dB is relative to the power of the input signal, dBm always relates to a 1 milliwatt signal. In other words, dB is a relative measurement, and dBm is an absolute measurement.

```
XdBm = XdBW + 30
```

### dBu

dB(0.775 Vrms)

Voltage relative to 0.775 volts.

Originally dBv, it was changed to dBu to avoid confusion with dBV. The "v" comes from "volt", while "u" comes from "unloaded". dBu can be used regardless of impedance, but is derived from a 600 Ω load dissipating 0 dBm (1 mW).

### dBV

dB(1 Vrms)

A logarithmic voltage ratio with a reference voltage relative to 1 volt, regardless of impedance.

### Deinterlace

The process of converting interlaced analog video signal into a non-interlaced form.

### Digital

A method of storing, processing and transmitting information through the use of distinct electronic or optical pulses that represent the binary digits 0 and 1.

### DirectShow

A multimedia framework and API produced by Microsoft for software developers to perform various operations with media files or streams.

### Dot crawl

A visual defect consisting of animated checkerboard patterns, which appears along horizontal color transitions. This occurs when the video decoder mixed up the high-frequency luminance information as chrominance information.

## *E*

### EIA

Electrical Industries Association. Monochrome video signal for North America and Japan TV standard (525 lines 60 Hz).

## *F*

### Frame

One frame is made up of two fields.

### Frame grabber

A device that interfaces with a camera and, on command, samples the video, converts the sample to a digital value (if the frame grabber is analog instead of digital), and stores that in a computer's memory.

### Field

One half of a frame.

### Frequency

The number of completed waveform in a given time, measured in cycles per second (Hz).

## *G*

### G.711 PCM audio encoder

Digitizes audio according to ITU-G.711, using either μ-law or A-law, and delivers a PCM encoded audio stream.

### Gain

Any increase or decrease in strength of an electrical signal and is often measured in terms of decibels.

### GUID

Globally Unique Identifier is a distinct 128-bit reference number that is used to identify a particular software application.

### GOP

Group Of Pictures.

## *H*

### H.264

A high quality video compression with different ranges of bit rates. Also known as **MPEG-4 Part 10**, or **MPEG AVC** (Advanced Video Coding). This standard can be applied to a wide variety of applications, networks and systems due to it flexibility.

### Hz

The measurement unit of frequency (cycles per second).

# I

## IDR picture

When the decoder receives an IDR picture, all subsequent transmitted slices can be decoded without references to any frame decoded prior to IDR picture.

## Image aspect ratio

The ratio of the width to the height of a frame of a video image. For conventional PAL/NTSC television standards, the aspect ratio is 4:3, or 1.333.

## Impedance

The total of the resistive and reactive opposition, measured in ohms, that a circuit presents to the flow of alternating current at a given frequency.

## Interlaced

A video storage mode consisting of fields (odd or even lines) with each field containing half of the lines in a frame.

## I/O

Input/Output.

## ISO

International Standards Organization.

## ITU

International Telecommunications Union.

# J

(empty)

# K

(empty)

# L

## LED

Light Emitting Diode.

## Line level audio

Line level is a term used to describe the strength of an audio signal used to transmit analog sound information between audio components and sending to recording devices. It is used as it minimizes noise and distortion when processing, transferring or reproducing recorded sound.

## Luminance

A signal that represents the scene brightness of the video signal information. The difference between luminance and brightness is that the latter is non-measurable and sensory. The color video picture information contains two components, luminance (brightness and contrast) and chrominance (hue and saturation).

# M

## Macroblock

Block of 16x16 pixels, used in image H.264 compression for instance.

## Mask

Block out certain portions of an image to prevent viewing.

## Moiré pattern

An unwanted effect that appears in the video picture when high frequencies are folding back to the lower frequencies.

## Monochrome

A black and white picture.

# N

## Noise

An unwanted signal produced by all electrical circuits. Noise cannot be eliminated but only minimized.

### NTSC

American committee that sets the standards for color television as used today in the US, Canada, Japan and parts of South America. NTSC television uses a 3.57945 MHz sub-carrier whose phase varies with the instantaneous hue of the televised color, and whose amplitude varies with the instantaneous saturation of the color. NTSC employs 525 lines per frame and 59.94 fields per second.

## *O*

### OEM

Original Equipment Manufacturer.

## *P*

### PAL

Phase alternating line. Describes the color phase change in a PAL color signal. PAL is a European color TV system featuring 625 lines per frame, 50 fields per second and a 4.43361875- MHz sub-carrier. Used mainly in Europe, China, Malaysia, Australia, New Zealand, the Middle East and parts of Africa.

### PCI

Peripheral Component Interconnect. A personal computer local bus designed by Intel, which runs at 33 MHz and supports Plug and Play.

### PCI Express

An emerging (2004/2005) standard for high-speed graphics, likely to result in a 20 % boost over 2003-era AGP 8x performance.

### PC reset relay

The PC reset relay is a component of the watchdog installed on Euresys frame grabbers. It is closed when the PC has to be reset.

### Pulse-code modulation

It is a digital representation of an analog signal in a numeric code, usually in binary code. This is achieved by sampling the signal at a regular interval and then convert it into a digital signal.

# Q

## QCIF

Quarter CIF. 176 x 144 for PAL and  176 x 120 for NTSC.

# R

## Resolution

A measure of number of dots in the horizontal lines of the camera or television system capability to reproduce image detail. For example, 720 x 576 is capable of displaying 720 dots on each 576 lines, or about 420,000 pixels.

## RGB color space

RGB is based on three primary colors, red, green and blue. It is used for video color representation.

# S

## Saturation

The intensity of the colors in the active picture. The degree by which the eye perceives a color as departing from a gray or white scale of the same brightness. For example, red is highly saturated, whereas a pale pink is not. A 100 % saturated color does not contain any white, adding white reduces saturation.

## Scaling

Enlarging an image according to a scale.

## SECAM

Sequential Color And Memory. Color television broadcast system used in France.

## SIF formats

SIF is a version of CIF for exchanging videos images for NTSC, PAL and SECAM. 240 x 352 for NTSC and 288 x 352 for PAL and SECAM.

## Signal-to-noise (S/N) ratio

An S/N ratio can be given for the luminance signal, chrominance signal and audio signal. The S/N ratio is the ratio of noise to actual total signal, and it shows how much higher the signal level is than the level of noise. It is expressed in decibels (dB), and the bigger the value is, the crisper and clearer the picture and sound will be during playback.

### Start-up

The start-up is a sequence of operations starting from the booting of the PC and ending with the launch of the application.

## T

(empty)

## U

(empty)

## V

### Video decoder

The video decoder recovers the PAL or NTSC color composite analog video signal, and converts it into a YUV422 digital video stream.

### Video encoder

The video encoder superimposes digital signal information on other digital signal, according to ITU-T rec. H.264 Baseline profile video compression standard, and delivers an encoded video stream.

### Video formatter

The video formatter formats digital video in either RGB or YUV color spaces, and delivers a raw video stream.

### Video pass-through selector

The video pass-through selector selects any of the 16 analog Video Inputs, or the Video Cascade Input, and delivers a buffered analog video signal to the Video Output.

### Vrms

For audio signals, and AC signals in general, the voltage level is often expressed as the **r**oot **m**ean **s**quare of the voltages.

Note that, in case of a sinusoidal signal:

```
VPeakToPeak ≈ 2.8 * Vrms
```

# W

## Watchdog

This device restarts the system in the event of a system failure, for instance a malfunction of the software application.

## Waveform

The shape of the electromagnetic wave. A graphical representation of voltage and current in relation to time.

# X

(empty)

# Y

## Y/C

Video signal containing separate luminance and chrominance components.

## YUV

A color encoding scheme for natural pictures in which luminance and chrominance are separate. YUV allows the encoding of luminance information at full bandwidth, and chrominance information at half bandwidth.

# Z

(empty)

# Index

## F



## G



## H



## I



## J



## K



## L



## M



## N



## O



## P



## Q



## R