# ELIIXA+ 8k/4k CL Color

## Cmos Multi-Line Color Camera

# User Manual

# Summary

# 1 CAMERA OVERVIEW

## 1.1 Features

- Cmos Colour Sensor :
  - 8192 RGB Pixels, 5 x 5µm (Full Definition)
  - 4096 RGB Pixels 10x10µm (True Colour)
- Interface : CameraLink® (up to 10 Taps at 85MHz)
- Line Rate :
  - Up to 50000 l/s In 8k Full Definition Mode
  - Up to 66000 l/s in 4k True Colour Mode
- Bit Depth : 24bits (RGB 8bits)
- Scan Direction
- Flat Field Correction
- Low Power Consumption : <9W
- F-Mount compliance

## 1.2 Key Specifications

| Characteristics | Typical Value | Unit |
|---|---|---|
| **Sensor Characteristics at Maximum Pixel Rate** | | |
| Resolution | 8192 or 4096 | RGB Pixels |
| pixel size (square) | 5 or 10 | µm |
| Max line rate | | |
| 4096 RGB Pixels True Color Mode | 66 | kHz |
| 8192 RGB Pixels Full Definition Modes | 50 | kHz |
| **Radiometric Performance at Maximum Pixel Rate and minimum camera gain** | | |
| Bit depth | 3 x 8 | Bits |
| Response (Peak) : True Color or Full Def Enhanced | | |
| Red | 11.8 | LSB 8bits/(nJ/cm²) |
| Green | 11.2 | LSB 8bits/(nJ/cm²) |
| Blue | 7.8 | LSB 8bits/(nJ/cm²) |
| Response non linearity | < 1 | % |
| PRNU HF Max | 3 | % |
| Dynamic range | 65 | dB |

| | | |
|---|---|---|
| **Functionality (Programmable via GenICam Control Interface)** | | |
| Analog Gain | Up to 12 (x4) | dB |
| Offset | -4096 to +4096 | LSB |
| Trigger Mode | Timed (Free run) and triggered (Ext Trig, Ext ITC) modes | |
| Sensor Modes | • True Color : 4096 RGB Pixels of 10x10µm<br>• Full Definition Enhanced : 8192 RGB Pixels 5x5µm<br>• Full Definition Single : 8192 RGB Pixels 5x5µm | |
| **Mechanical and Electrical Interface** | | |
| Size (w x h x l) | 126 x 60 x 35 | mm |
| Weight | 360 | g |
| Lens Mounts | F, T2, M42 | - |
| Sensor alignment ( see chapter 4 ) | ±100 | µm |
| Sensor flatness | ±35 | µm |
| Power supply | 12 - 24 | V |
| Power dissipation – Typ. while grabbing | < 9 | W |
| **General Features** | | |
| Operating temperature | 0 to 55 (front face) or 70 (Internal) | °C |
| Storage temperature | -40 to 70 | °C |
| Regulatory | CE, FCC and RoHS compliant | |

## 1.3 Description

e2v's next generation of line scan cameras are setting new, high standards for line rate and image quality. Thanks to e2v's recently developed multi-line CMOS technology, the camera provides an unmatched 100,000 lines/s and combines high response with an extremely low noise level; this delivers high signal to noise ratio even when short integration times are required or when illumination is limited. The 5µm pixel size is arranged in four active lines and dual line filter configuration allowing the camera to be operated in several modes: True colour mode with 10µm RGB pixels to provide equivalent colour fidelity to 10µm pixel tri-linear solutions with advanced immunity to web variation or Full definition mode with a 8192 RGB pixel resolution.

## 1.4 Typical Applications

- Raw material surface inspection
- Flat panel display inspection
- PCB inspection
- Solar cell inspection
- Parcel and postal sorting
- High resolution document scanning
- Print and paper inspection

# 2 CAMERA PERFORMANCES

## *2.1 Camera Characterization*

| | Unit | True Color (8k) | | Full Definition Single | | Full Definition Enhanced | |
|---|---|---|---|---|---|---|---|
| | | Typ. | Max | Typ. | Max | Typ. | Max |
| **Dark Noise RMS** | LSB | 0.12 | 1.2 | 0.11 | 1.2 | 0.12 | 1.2 |
| **Dynamic Range** | - | 2125:1 | - | 2125:1 | - | 2125:1 | - |
| **RMS Noise** (3/4 Sat) | LSB | 2.2 | - | 2.15 | 4 | 2.2 | 4 |
| **Full Well Capacity** | e- (per color) | 13650 | - | 13650 | - | 13650 | - |
| **SNR** (3/4 Sat) | dB | 40 | - | 40 | - | 40 | - |
| **Peak Response** (460/530/660nm) | LSB 8bits/ (nJ/cm2) | 8/10/12 | - | 4/5/6 | - | 8/10/12 | - |
| **Non Linearity** | % | 0,3 | - | 0,3 | - | 0,3 | - |
| **Without Flat Field Correction :** | | | | | | | |
| **FPN rms** | LSB | 0.21 | 1 | 0.23 | 1 | 0.22 | 1 |
| **FPN pk-pk** | LSB | 1 | 2 | 1 | 2 | 1 | 2 |
| **PRNU hf** (3/4 Sat) | % | 0.13 | 0,35 | 0.123 | 0,35 | 0.14 | 0,35 |
| **PRNU pk-pk** (3/4 Sat) | % | 1.1 | 3 | 1 | 3 | 1.25 | 3 |

Test conditions :

- All values are given at Nominal Gain (0dB) : Preamp Gain x1, Amp Gain 0dB
- Figures in LSB are for a 8bits format
- Measured at exposure time = 400µs and line period = 400µs in Ext Trig Mode (Max Exposure Time)
- Maximum data rate

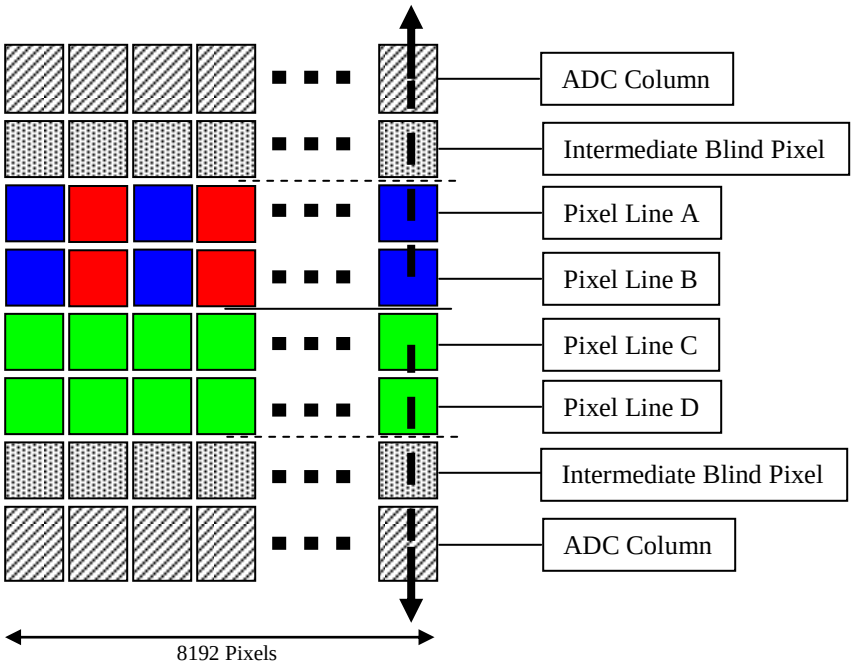
## 2.2 Image Sensor

The Eliixa+ Colour 8k sensor is composed of two pairs of sensitive lines.

The Colour version has been completed with RGB colour Filter and disposed as detailed beside.

Each pair of lines use the same Analog to Digital Column converter (ADC Column). An appropriate (embedded) Time delay in the exposure between each line this allows to combine two successive exposures in order to double the sensitivity of a single line.

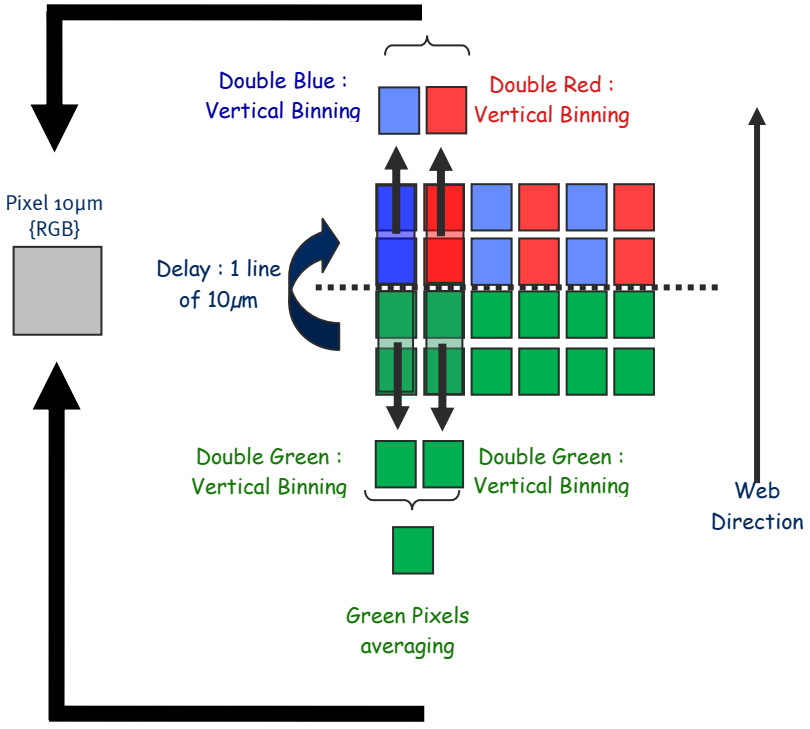
This Time Delay Exposure is used only in the Full Definition Enhanced mode (See Below).




### 2.2.1 True Colour Mode (TC)

10µm pixels (R,G,B)
Twice less pixels than B/W
Requires x3/2 the data flow of B&W

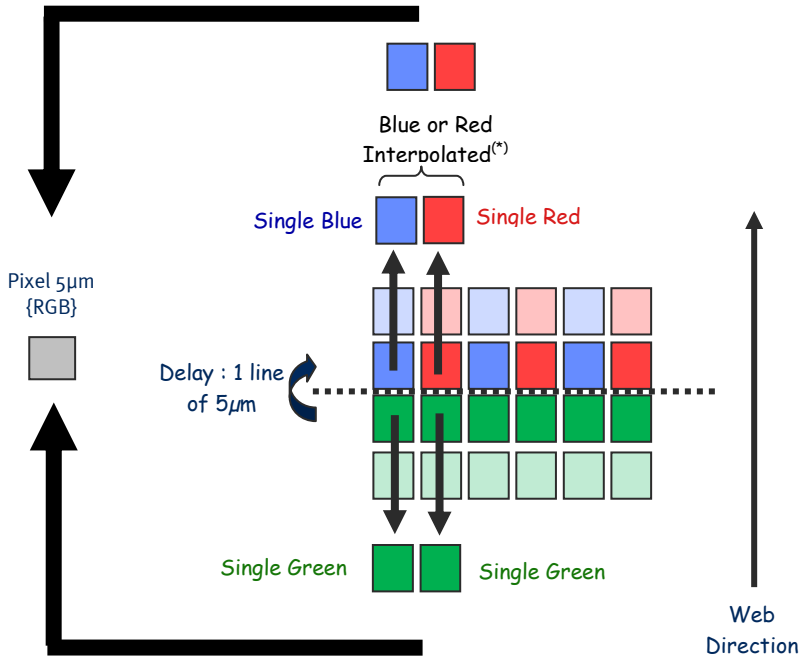
- Only one (High Sensitivity) TC mode available : Equivalent to 6 x Pixels of 5µm (with their respective colour filters).
- "Full Exposure control" not needed in TC as the TDI is not active (only binning). The Exposure time can be control as for a single line mode.



## 2.2.2 Full Definition Single Mode (FD Single)

5µm pixels (R,G,B)
Same definition than B&W
Requires x3 the data flow of the B&W

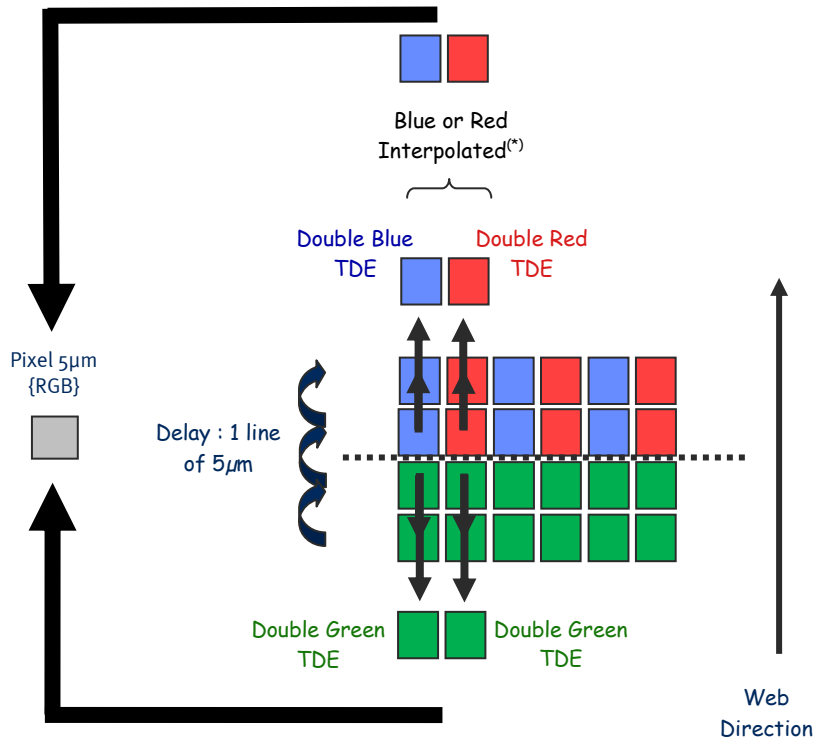
- Sensitivity is half of the TC mode available : Equivalent to 3 x Pixels of 5µm (with their respective colour filters).
- "Full Exposure control" not needed in this mode as the Time Delay Exposure is not active. The Exposure time can be control as for a single line mode.




## 2.2.3 Full Definition Enhanced Mode (FD Enhanced)

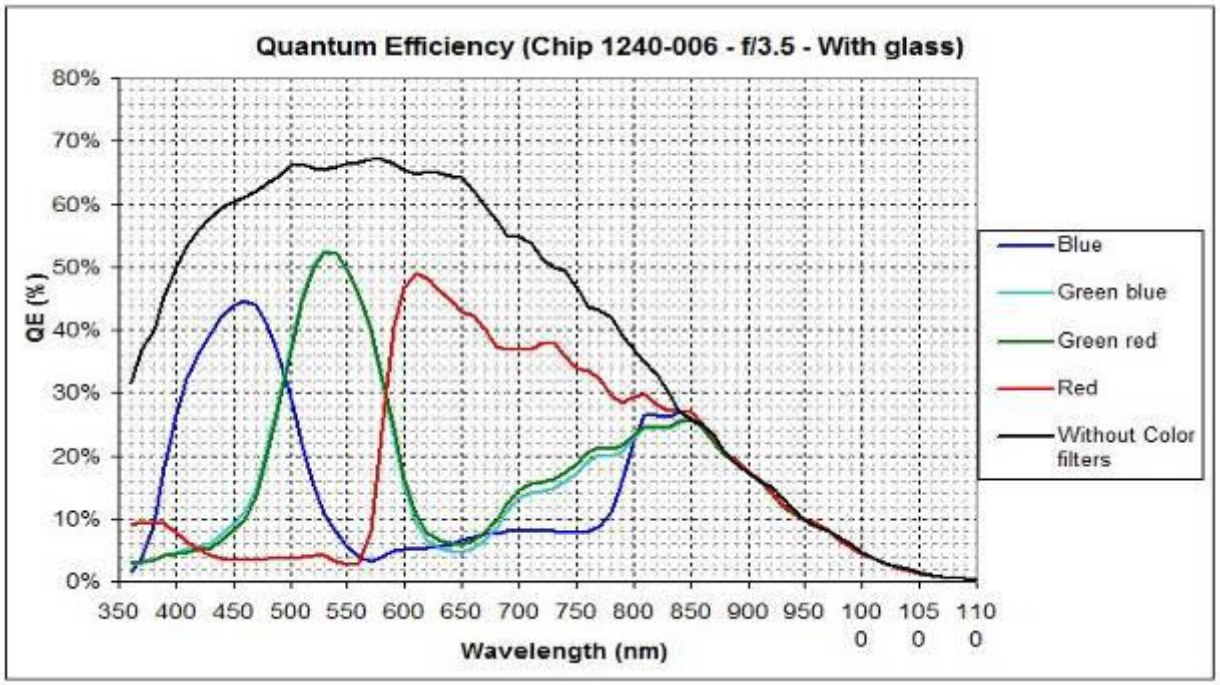
5µm pixels (R,G,B)
Same definition than B&W
Requires x3 the data flow of the B&W

- Sensitivity is the same as the TC mode available : Equivalent to 6 x Pixels of 5µm (with their respective colour filters).
- "Full Exposure control" is activated in this mode as the Time Delay Exposure is active.



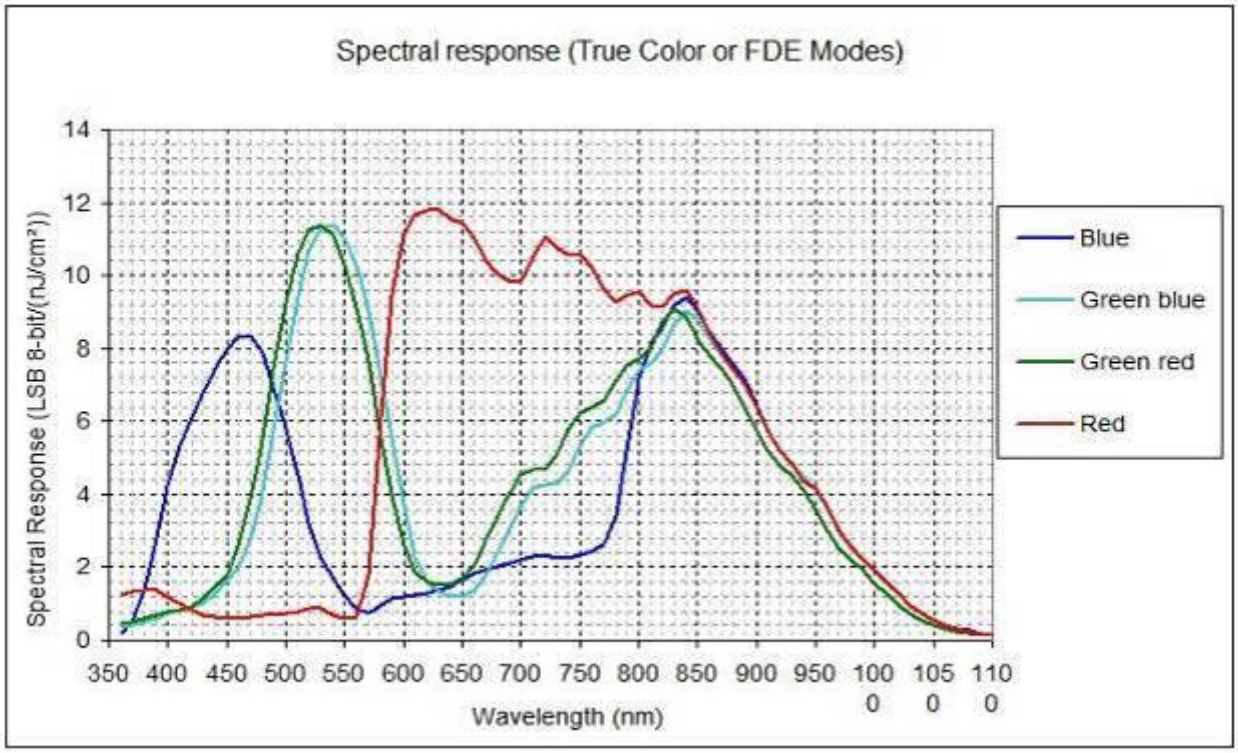
## *2.3 Response & QE curves*

### 2.3.1 Quantum Efficiency



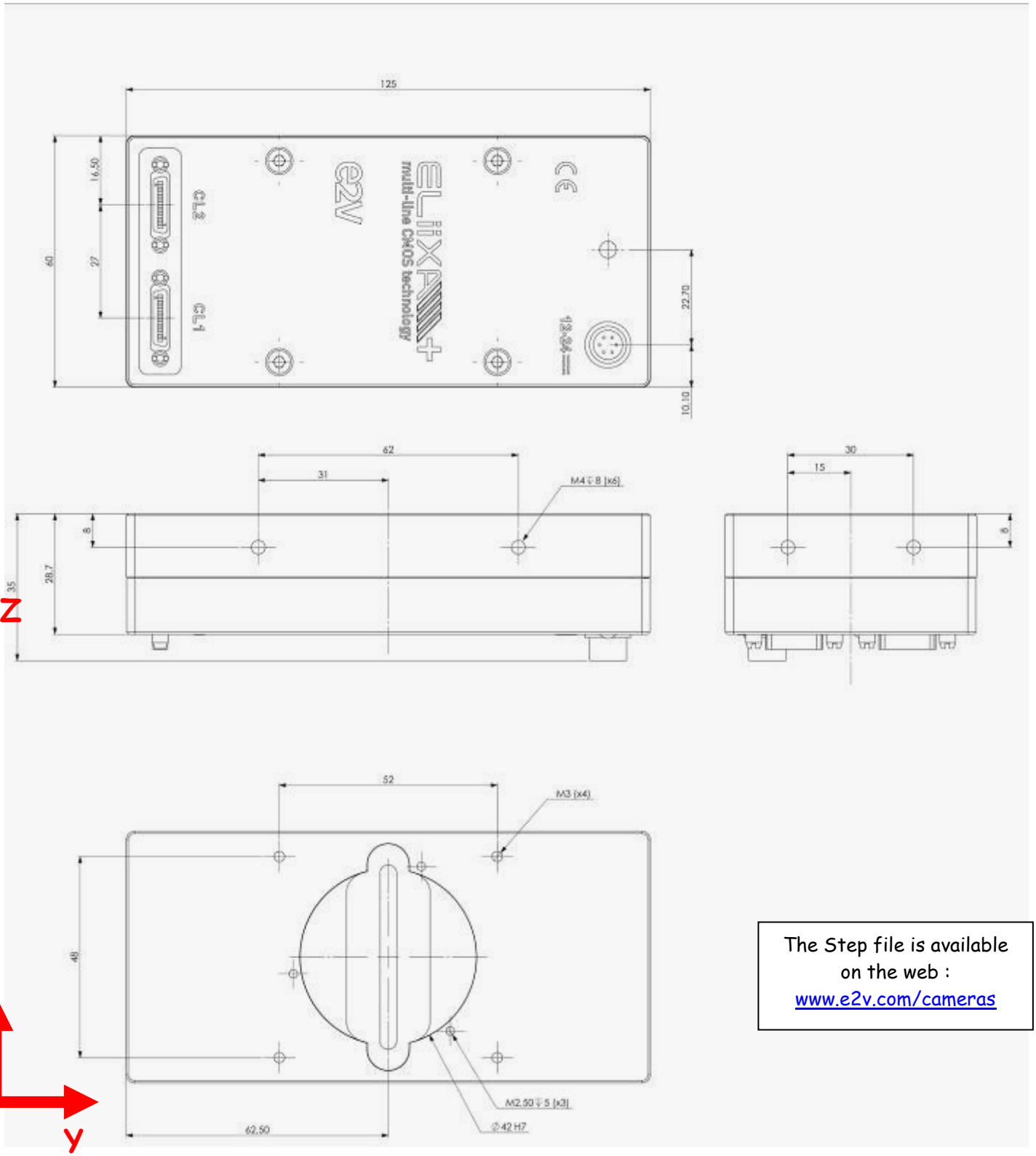

### 2.3.2 Spectral Response

*Mode "True color" or "Full definition Enhanced"*



# 3 CAMERA HARDWARE INTERFACE

## *3.1 Mechanical Drawings*



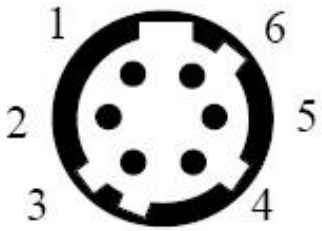

The Step file is available on the web : www.e2v.com/cameras

| Sensor alignment | |
|---|---|
| Z = -10.3 mm | ±100µm |
| X = 9.5 mm | ±100 µm |
| Y = 62.5mm | ±100 µm |
| Flatness | 50 µm |
| Rotation (X,Y plan) | ±0,15° |
| Tilt (versus lens mounting plane) | 50µm |

## 3.2 Input/output Connectors and LED

### 3.2.1 Power Connector

Camera connector type: Hirose HR10A-7R-6PB (male)
Cable connector type: Hirose HR10A-7P-6S (female)



Camera side description

| Signal | Pin | Signal | Pin |
|---|---|---|---|
| PWR | 1 | GND | 4 |
| PWR | 2 | GND | 5 |
| PWR | 3 | GND | 6 |

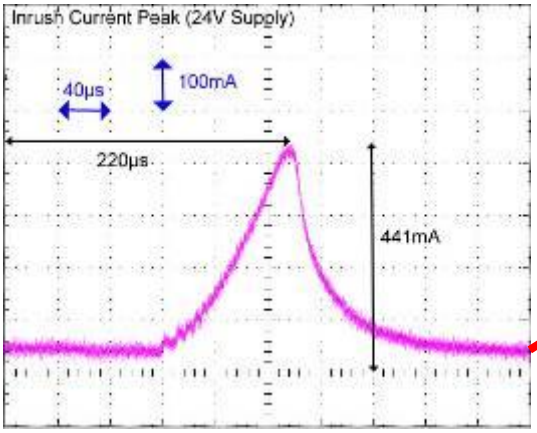
Power supply from 12 to 24v
Power 7,5W max with an typical inrush current peak of **1A** during power up

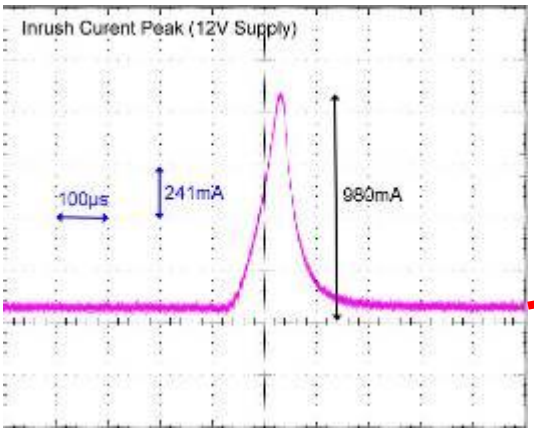
Typical current/Power during the grab (possible variation : +/- 5%)

| Camera supply (Line Period Minimum) | Supply 12V | | Supply 24V | |
|---|---|---|---|---|
| | I(mA) | P(W) | I(mA) | P(W) |
| **Full 8Taps** | 605 | 7.26 | 303 | 7.272 |
| **Deca 10Taps** | 613 | 7.356 | 308 | 7.392 |
| **Base 3Taps RGB** | 589 | 7.068 | 298 | 7.152 |
| **Medium 2x 3Taps RGB** | 598 | 7.176 | 302 | 7.248 |

**Power Time : Max 40s (Green Light)**

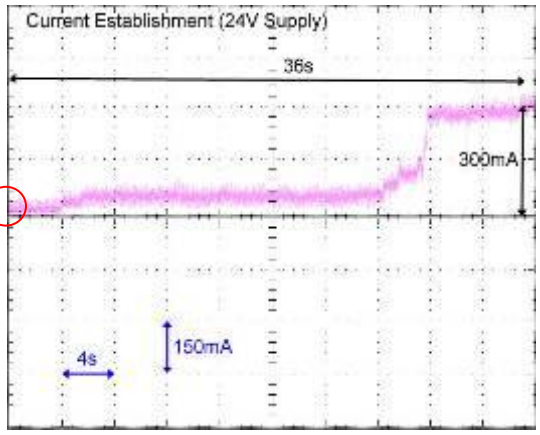
**Inrush Current Peak**



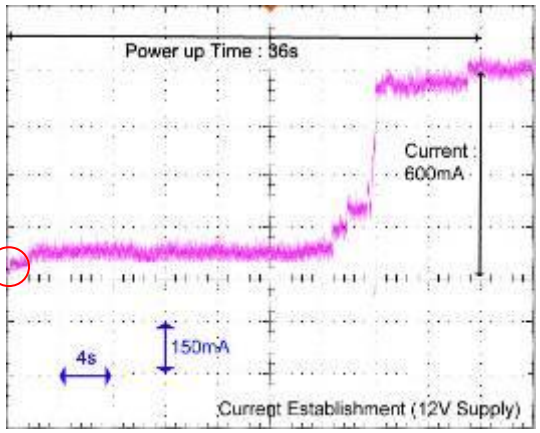




**Current Establishment time and level**

24V




12V



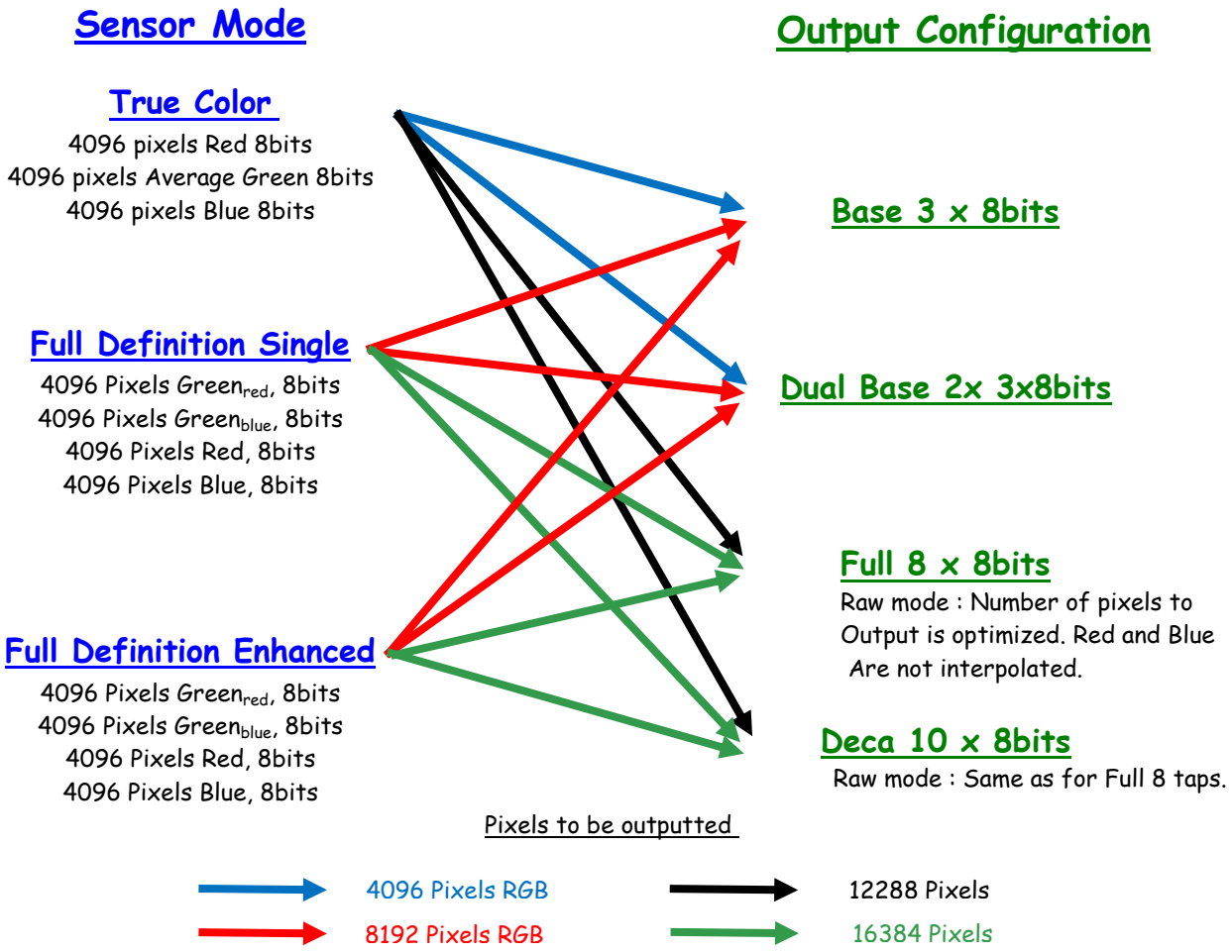
### 3.2.2 Status LED Behaviour

After less than 2 seconds of power establishment, the LED first lights up in ORANGE. Then after a Maximum of 40 seconds, the LED must turn in a following colour :

| Colour and state | Meaning |
|---|---|
| Green and continuous | OK |
| Green and blinking slowly | Waiting for Ext Trig (Trig1 and/or Trig2) |
| Red and continuous | Camera out of order : Internal firmware error |

### 3.2.3 CameraLink Output Configuration

| | Adjacent Channels |
|---|---|
| **Base** : 3 Channels RGB 8bits | 3 x 85MHz |
| **Medium** : 2 x 3 Channels RGB 8bits | 2x 3 x 85MHz |
| **Full** : 8 Channels 8bits | 8 x 85MHz |
| **Deca** : 10 Channels 8bits | 10 x 85MHz |



### 3.2.3.1 True Color

In Base Output Mode

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap1 | RED 1 | RED 3 | RED 5 | RED 7 | RED 9 | RED 11 | RED 13 | RED 15 | RED 8181 | RED 8183 | RED 8185 | RED 8187 | RED 8189 | RED 8191 |
| Tap2 | GREEN 1 | GREEN 2 | GREEN 3 | GREEN 4 | GREEN 5 | GREEN 6 | GREEN 7 | GREEN 8 | GREEN 4091 | GREEN 4092 | GREEN 4093 | GREEN 4094 | GREEN 4095 | GREEN 4096 |
| Tap3 | BLUE 2 | BLUE 4 | BLUE 6 | BLUE 8 | BLUE 10 | BLUE 12 | BLUE 14 | BLUE 16 | BLUE 8182 | BLUE 8184 | BLUE 8186 | BLUE 8188 | BLUE 8190 | BLUE 8192 |

In Medium or Dual Base Output Mode

**Connector 1**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap1 | RED 1 | RED 2 | RED 3 | RED 4 | RED 5 | RED 6 | RED 7 | RED 8 | RED 2043 | RED 2044 | RED 2045 | RED 2046 | RED 2047 | RED 2048 |
| Tap2 | GREEN 1 | GREEN 2 | GREEN 3 | GREEN 4 | GREEN 5 | GREEN 6 | GREEN 7 | GREEN 8 | GREEN 2043 | GREEN 2044 | GREEN 2045 | GREEN 2046 | GREEN 2047 | GREEN 2048 |
| Tap3 | BLUE 1 | BLUE 2 | BLUE 3 | BLUE 4 | BLUE 5 | BLUE 6 | BLUE 7 | BLUE 8 | BLUE 2043 | BLUE 2044 | BLUE 2045 | BLUE 2046 | BLUE 2047 | BLUE 2048 |

**Connector 2**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap4 | RED 2049 | RED 2050 | RED 2051 | RED 2052 | RED 2053 | RED 2054 | RED 2055 | RED 2056 | RED 4091 | RED 4092 | RED 4093 | RED 4094 | RED 4095 | RED 4096 |
| Tap5 | GREEN 2049 | GREEN 2050 | GREEN 2051 | GREEN 2052 | GREEN 2053 | GREEN 2054 | GREEN 2055 | GREEN 2056 | GREEN 4091 | GREEN 4092 | GREEN 4093 | GREEN 4094 | GREEN 4095 | GREEN 4096 |
| Tap6 | BLUE 2049 | BLUE 2050 | BLUE 2051 | BLUE 2052 | BLUE 2053 | BLUE 2054 | BLUE 2055 | BLUE 2056 | BLUE 4091 | BLUE 4092 | BLUE 4093 | BLUE 4094 | BLUE 4095 | BLUE 4096 |

In Full 8 Taps Output Mode

**Connector 1**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap1 | RED 1 | BLUE 3 | GREEN 6 | RED 9 | BLUE 11 | GREEN 14 | RED 17 | BLUE 19 | RED 4081 | BLUE 4083 | GREEN 4086 | RED 4089 | BLUE 4091 | GREEN 4094 |
| Tap2 | GREEN 1 | RED 4 | BLUE 6 | GREEN 9 | RED 12 | BLUE 14 | GREEN 17 | RED 20 | GREEN 4081 | RED 4084 | BLUE 4086 | GREEN 4089 | RED 4092 | BLUE 4094 |
| Tap3 | BLUE 1 | GREEN 4 | RED 7 | BLUE 9 | GREEN 12 | RED 15 | BLUE 17 | GREEN 20 | BLUE 4081 | GREEN 4084 | RED 4087 | BLUE 4089 | GREEN 4092 | RED 4095 |

**Connector 2**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap4 | RED 2 | BLUE 4 | GREEN 7 | RED 10 | BLUE 12 | GREEN 15 | RED 18 | BLUE 20 | RED 4082 | BLUE 4084 | GREEN 4087 | RED 4090 | BLUE 4092 | GREEN 4095 |
| Tap5 | GREEN 2 | RED 5 | BLUE 7 | GREEN 10 | RED 13 | BLUE 15 | GREEN 18 | RED 21 | GREEN 4082 | RED 4085 | BLUE 4087 | GREEN 4090 | RED 4093 | BLUE 4095 |
| Tap6 | BLUE 2 | GREEN 5 | RED 8 | BLUE 10 | GREEN 13 | RED 16 | BLUE 18 | GREEN 21 | BLUE 4082 | GREEN 4085 | RED 4088 | BLUE 4090 | GREEN 4093 | RED 4096 |
| Tap7 | RED 3 | BLUE 5 | GREEN 8 | RED 11 | BLUE 13 | GREEN 16 | RED 19 | BLUE 21 | RED 4083 | BLUE 4085 | GREEN 4088 | RED 4091 | BLUE 4093 | GREEN 4096 |
| Tap8 | GREEN 3 | RED 6 | BLUE 8 | GREEN 11 | RED 14 | BLUE 16 | GREEN 19 | RED 22 | GREEN 4083 | RED 4086 | BLUE 4088 | GREEN 4091 | RED 4094 | BLUE 4096 |

In Deca 10 Taps Output Mode

**Connector 1**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tap1** | RED 1 | GREEN 4 | BLUE 7 | RED 11 | GREEN 14 | BLUE 17 | RED 21 | GREEN 24 | | BLUE 4077 | RED 4081 | GREEN 4084 | BLUE 4087 | RED 4091 | GREEN 4094 |
| **Tap2** | GREEN 1 | BLUE 4 | RED 8 | GREEN 11 | BLUE 14 | RED 18 | GREEN 21 | BLUE 24 | | RED 4078 | GREEN 4081 | BLUE 4084 | RED 4088 | GREEN 4091 | BLUE 4094 |
| **Tap3** | BLUE 1 | RED 5 | GREEN 8 | BLUE 11 | RED 15 | GREEN 18 | BLUE 21 | RED 25 | | GREEN 4078 | BLUE 4081 | RED 4085 | BLUE 4088 | BLUE 4091 | RED 4095 |

**Connector 2**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tap4** | RED 2 | GREEN 5 | BLUE 8 | RED 12 | GREEN 15 | BLUE 18 | RED 22 | GREEN 25 | | BLUE 4078 | RED 4082 | GREEN 4085 | BLUE 4088 | RED 4092 | GREEN 4095 |
| **Tap5** | GREEN 2 | BLUE 5 | RED 9 | GREEN 12 | BLUE 15 | RED 19 | GREEN 22 | BLUE 25 | | RED 4079 | GREEN 4082 | BLUE 4085 | RED 4089 | GREEN 4092 | BLUE 4095 |
| **Tap6** | BLUE 2 | RED 6 | GREEN 9 | BLUE 12 | RED 16 | GREEN 19 | BLUE 22 | RED 26 | | GREEN 4079 | BLUE 4082 | RED 4086 | GREEN 4089 | BLUE 4092 | RED 4096 |
| **Tap7** | RED 3 | GREEN 6 | BLUE 9 | RED 13 | GREEN 16 | BLUE 19 | RED 23 | GREEN 26 | | BLUE 4079 | RED 4083 | GREEN 4086 | BLUE 4089 | RED 4093 | GREEN 4096 |
| **Tap8** | GREEN 3 | BLUE 6 | RED 10 | GREEN 13 | BLUE 16 | RED 20 | GREEN 23 | BLUE 26 | | RED 4080 | GREEN 4083 | BLUE 4086 | RED 4090 | GREEN 4093 | BLUE 4096 |
| **Tap9** | BLUE 3 | RED 7 | GREEN 10 | BLUE 13 | RED 17 | GREEN 20 | BLUE 23 | RED 27 | | GREEN 4080 | BLUE 4083 | RED 4087 | GREEN 4090 | BLUE 4093 | |
| **Tap10** | RED 4 | GREEN 7 | BLUE 10 | RED 14 | GREEN 17 | BLUE 20 | RED 24 | GREEN 27 | | BLUE 4080 | RED 4084 | GREEN 4087 | BLUE 4090 | RED 4094 | |

### 3.2.3.2 Full Definition (Single or Enhanced)

In Base Output Mode

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tap1** | RED 1 | RED 2 | RED 3 | RED 4 | RED 5 | RED 6 | RED 7 | RED 8 | | RED 8187 | RED 8188 | RED 8189 | RED 8190 | RED 8191 | RED 8192 |
| **Tap2** | GREEN 1 | GREEN 2 | GREEN 3 | GREEN 4 | GREEN 5 | GREEN 6 | GREEN 7 | GREEN 8 | | GREEN 8187 | GREEN 8188 | GREEN 8189 | GREEN 8190 | GREEN 8191 | GREEN 8192 |
| **Tap3** | BLUE 1 | BLUE 2 | BLUE 3 | BLUE 4 | BLUE 5 | BLUE 6 | BLUE 7 | BLUE 8 | | BLUE 8187 | BLUE 8188 | BLUE 8189 | BLUE 8190 | BLUE 8191 | BLUE 8192 |

## In Medium or Dual Base Output Mode

### Connector 1

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap1 | RED 1 | RED 2 | RED 3 | RED 4 | RED 5 | RED 6 | RED 7 | RED 8 | | RED 4091 | RED 4092 | RED 4093 | RED 4094 | RED 4095 | RED 4096 |
| Tap2 | GREEN 1 | GREEN 2 | GREEN 3 | GREEN 4 | GREEN 5 | GREEN 6 | GREEN 7 | GREEN 8 | | GREEN 4091 | GREEN 4092 | GREEN 4093 | GREEN 4094 | GREEN 4095 | GREEN 4096 |
| Tap3 | BLUE 1 | BLUE 2 | BLUE 3 | BLUE 4 | BLUE 5 | BLUE 6 | BLUE 7 | BLUE 8 | | BLUE 4091 | BLUE 4092 | BLUE 4093 | BLUE 4094 | BLUE 4095 | BLUE 4096 |

### Connector 2

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap4 | RED 4097 | RED 4098 | RED 4099 | RED 4100 | RED 4101 | RED 4102 | RED 4103 | RED 4104 | | RED 8187 | RED 8188 | RED 8189 | RED 8190 | RED 8191 | RED 8192 |
| Tap5 | GREEN 4097 | GREEN 4098 | GREEN 4099 | GREEN 4100 | GREEN 4101 | GREEN 4102 | GREEN 4103 | GREEN 4104 | | GREEN 8187 | GREEN 8188 | GREEN 8189 | GREEN 8190 | GREEN 8191 | GREEN 8192 |
| Tap6 | BLUE 4097 | BLUE 4098 | BLUE 4099 | BLUE 4100 | BLUE 4101 | BLUE 4102 | BLUE 4103 | BLUE 4104 | | BLUE 8187 | BLUE 8188 | BLUE 8189 | BLUE 8190 | BLUE 8191 | BLUE 8192 |

## In Full 8 Taps Output Mode

### Connector 1

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap1 | BLUE 1 | RED 2 | BLUE 3 | RED 4 | BLUE 5 | RED 6 | BLUE 7 | RED 8 | | BLUE 2043 | RED 2044 | BLUE 2045 | RED 2046 | BLUE 2047 | RED 2048 |
| Tap2 | GREEN 1 | GREEN 2 | GREEN 3 | GREEN 4 | GREEN 5 | GREEN 6 | GREEN 7 | GREEN 8 | | GREEN 2043 | GREEN 2044 | GREEN 2045 | GREEN 2046 | GREEN 2047 | GREEN 2048 |
| Tap3 | BLUE 2049 | RED 2050 | BLUE 2051 | RED 2052 | BLUE 2053 | RED 2054 | BLUE 2055 | RED 2056 | | BLUE 4091 | RED 4092 | BLUE 4093 | RED 4094 | BLUE 4095 | RED 4096 |

### Connector 2

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tap4 | GREEN 2049 | GREEN 2050 | GREEN 2051 | GREEN 2052 | GREEN 2053 | GREEN 2054 | GREEN 2055 | GREEN 2056 | | GREEN 4091 | GREEN 4092 | GREEN 4093 | GREEN 4094 | GREEN 4095 | GREEN 4096 |
| Tap5 | BLUE 4097 | RED 4098 | BLUE 4099 | RED 4100 | BLUE 4101 | RED 4102 | BLUE 4103 | RED 4104 | | BLUE 6139 | RED 6140 | BLUE 6141 | RED 6142 | BLUE 6143 | RED 6144 |
| Tap6 | GREEN 4097 | GREEN 4098 | GREEN 4099 | GREEN 4100 | GREEN 4101 | GREEN 4102 | GREEN 4103 | GREEN 4104 | | GREEN 6139 | GREEN 6140 | GREEN 6141 | GREEN 6142 | GREEN 6143 | GREEN 6144 |
| Tap7 | BLUE 6145 | RED 6146 | BLUE 6147 | RED 6148 | BLUE 6149 | RED 6150 | BLUE 6151 | RED 6152 | | BLUE 8187 | RED 8188 | BLUE 8189 | RED 8190 | BLUE 8191 | RED 8192 |
| Tap8 | GREEN 6145 | GREEN 6146 | GREEN 6147 | GREEN 6148 | GREEN 6149 | GREEN 6150 | GREEN 6151 | GREEN 6152 | | GREEN 8187 | GREEN 8188 | GREEN 8189 | GREEN 8190 | GREEN 8191 | GREEN 8192 |

In Deca 10 Taps Output Mode

**Connector 1**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tap1** | BLUE | RED | BLUE | RED | BLUE | RED | BLUE | RED | | BLUE | RED | BLUE | RED | BLUE | RED |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 1633 | 1634 | 1635 | 1636 | 1637 | 1638 |
| **Tap2** | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 1633 | 1634 | 1635 | 1636 | 1637 | 1638 |
| **Tap3** | BLUE | RED | BLUE | RED | BLUE | RED | BLUE | RED | | BLUE | RED | BLUE | RED | BLUE | RED |
| | 1639 | 1640 | 1641 | 1642 | 1643 | 1644 | 1645 | 1646 | | 3271 | 3272 | 3273 | 3274 | 3275 | 3276 |

**Connector 2**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tap4** | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN |
| | 1639 | 1640 | 1641 | 1642 | 1643 | 1644 | 1645 | 1646 | | 3271 | 3272 | 3273 | 3274 | 3275 | 3276 |
| **Tap5** | BLUE | RED | BLUE | RED | BLUE | RED | BLUE | RED | | BLUE | RED | BLUE | RED | BLUE | RED |
| | 3277 | 3278 | 3279 | 3280 | 3281 | 3282 | 3283 | 3284 | | 4909 | 4910 | 4911 | 4912 | 4913 | 4914 |
| **Tap6** | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN |
| | 3277 | 3278 | 3279 | 3280 | 3281 | 3282 | 3283 | 3284 | | 4909 | 4910 | 4911 | 4912 | 4913 | 4914 |
| **Tap7** | BLUE | RED | BLUE | RED | BLUE | RED | BLUE | RED | | BLUE | RED | BLUE | RED | BLUE | RED |
| | 4915 | 4916 | 4917 | 4918 | 4919 | 4920 | 4921 | 4922 | | 6547 | 6548 | 6549 | 6550 | 6551 | 6552 |
| **Tap8** | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN |
| | 4915 | 4916 | 4917 | 4918 | 4919 | 4920 | 4921 | 4922 | | 6547 | 6548 | 6549 | 6550 | 6551 | 6552 |
| **Tap9** | BLUE | RED | BLUE | RED | BLUE | RED | BLUE | RED | | BLUE | RED | BLUE | RED | BLUE | RED |
| | 6553 | 6554 | 6555 | 6556 | 6557 | 6558 | 6559 | 6560 | | 8185 | 8186 | 8187 | 8188 | 8189 | 8190 |
| **Tap10** | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN | | GREEN | GREEN | GREEN | GREEN | GREEN | GREEN |
| | 6553 | 6554 | 6555 | 6556 | 6557 | 6558 | 6559 | 6560 | | 8185 | 8186 | 8187 | 8188 | 8189 | 8190 |

# 4 STANDARD CONFORMITY

The ELIIXA+ cameras have been tested using the following equipment:

- A shielded power supply cable
- A Camera Link data transfer cable ref. MVC-1-1-5-2M from CEI (Component Express, Inc.)

e2v recommends using the same configuration to ensure the compliance with the following standards.

## *4.1 CE Conformity*

The ELIIXA+ cameras comply with the requirements of the EMC (European) directive 2004/108/CE (EN50081-2, EN 61000-6-2).

## *4.2 FCC Conformity*

The ELIIXA+ cameras further comply with Part 15 of the FCC rules, which states that: Operation is subject to the following two conditions:

- This device may not cause harmful interference, and
- This device must accept any interference received, including interference that may cause undesired operation

This equipment has been tested and found to comply with the limits for Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference 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 instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

**Warning**: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

## *4.3 RoHs Conformity*

ELIIXA+ cameras comply with the requirements of the RoHS directive 2002/95/EC.

**Declaration Number: NE31S208701**

---

**We,**

**e2v semiconductors**
**rue de Rochepleine**
**38120 Saint-Egrève**
**France**

**declare the product(s)**

---

Product Family: ***EliiXA+ 16k Cameras***

Model Identification: **EV71YC4MCL1605-Bxx / EV71YC4MCP1605-Bxx**
**EV71YC4CCP1605-Bxx**

x = 0-9-A-Z

---

**in conformance with the requirements of the following standards:**

---

EN55022 : ed. 2006, A class

EN61000-6-2 : ed. 2005
- IEC 61000-4-2 : ed.2009
- IEC 61000-4-3 : ed.2006 + A1/2008 +A2/2011
- IEC 61000-4-4 : ed.2004
- IEC 61000-4-5 : ed.2006
- IEC 61000-4-6 : ed.2009
- IEC 61000-4-11 : ed.2004

when used in conformity with the recommended set-up (as per the Product Specification or Data Sheet).

---

**applicable to:** **Information Technology Equipments (I.T.E.)**

---

**This (These) product(s) complies(y) with the requirements of the:**
- Electromagnetic Compatibility Directive 2004/108/EC,
- CE Marking European Directive 93/68/EEC

**and carry the CE marking accordingly.**

Saint-Egrève, France, on June 13rd, 2013

Martine WOOLF,
Quality Manager

# 5 GETTING STARTED

## 5.1 Out of the box

The contains of the Camera box is the following :

- One Camera ELIIXA+
- Power connector (Hirose HR10A-7P-6S -female)

*There is no CDROM delivered with the Camera : Both User Manual (this document) and CommCam control software have to be downloaded from the web site : This ensure you to have an up-to-date version.*

*Main Camera page : **www.e2v.com/cameras***

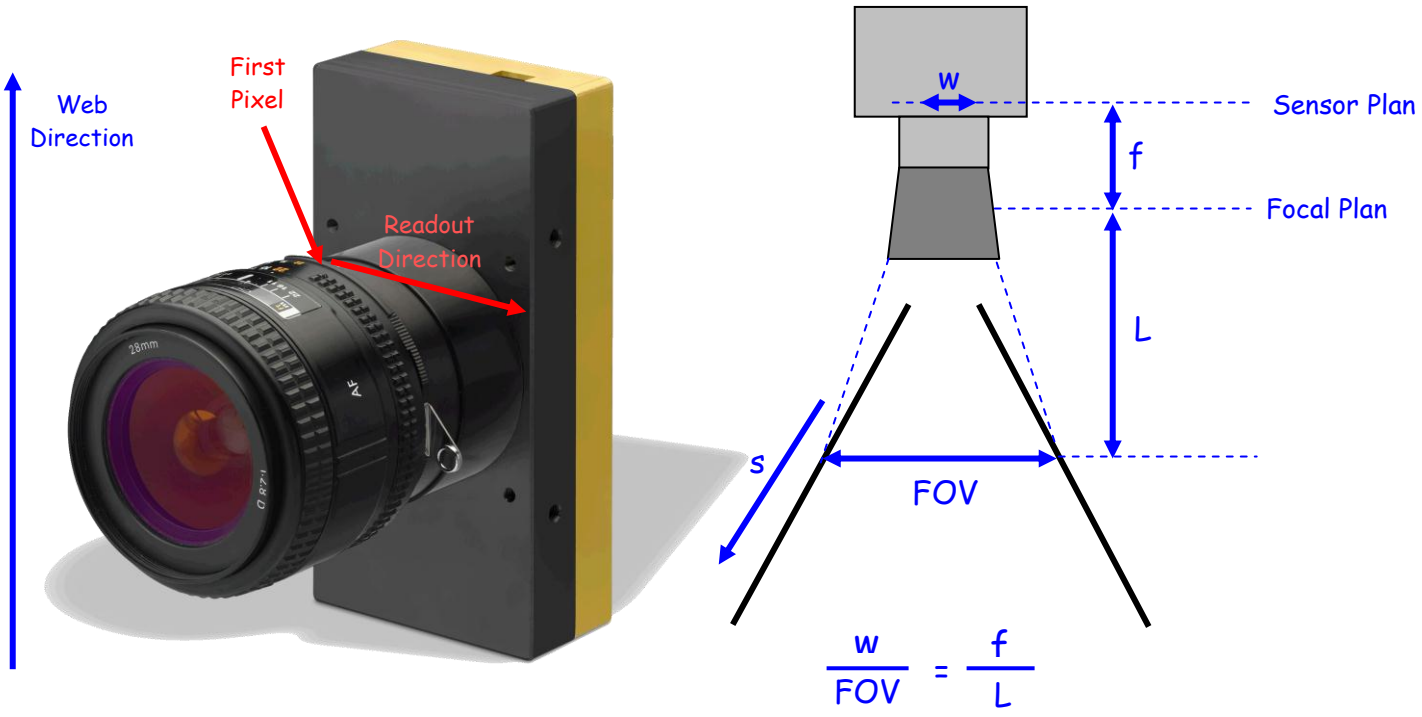
*On the appropriate Camera Page (ELIIXA+ 8k/4k color) you'll find a download link*

*first version of CommCam compliant is indicated in the last Chapter*

*CommCam download requires a login/password :*

- *Login : **commcam***
- *Password : **chartreuse***

## 5.2 Setting up in the system

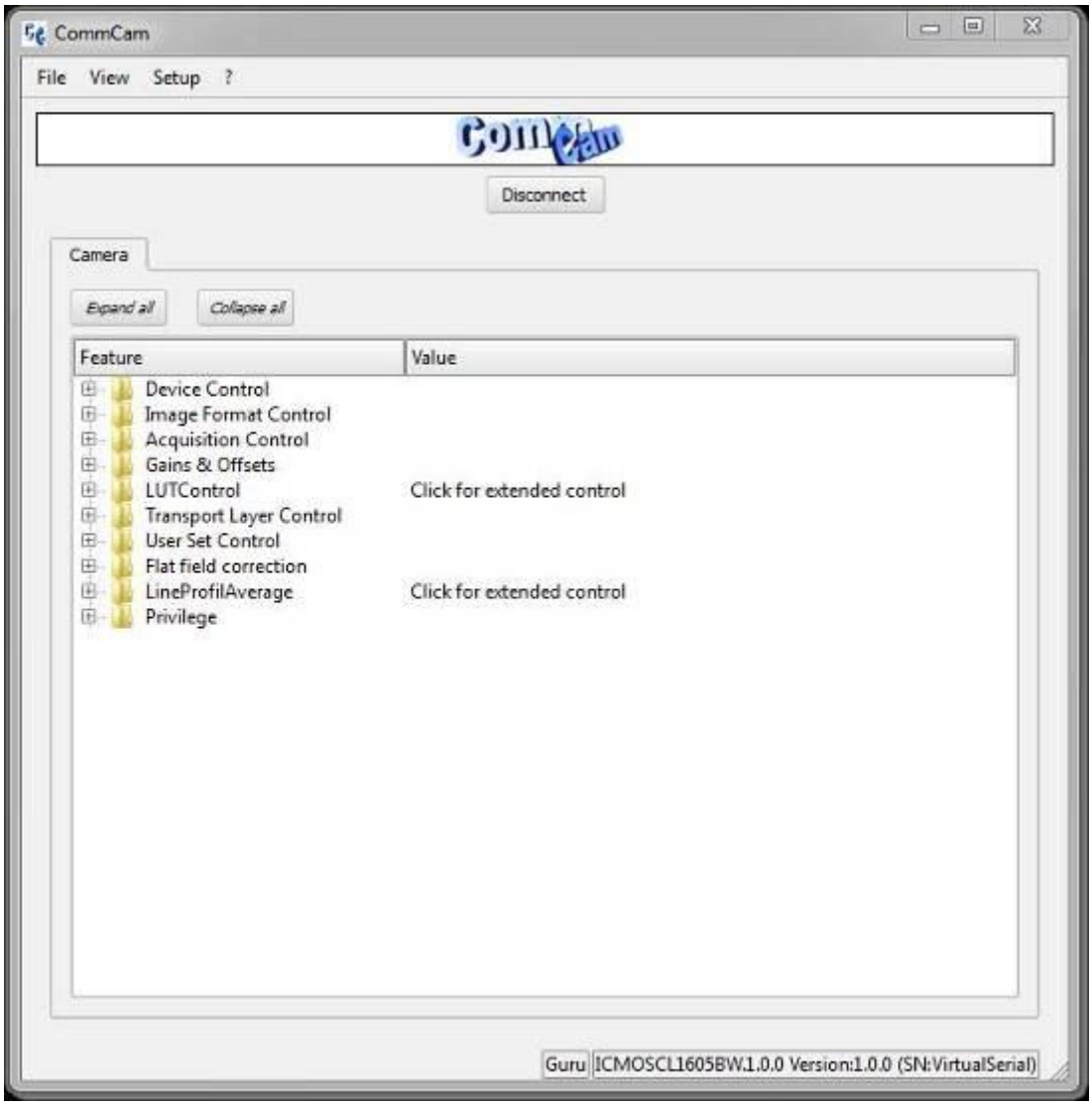



The Compliant Lenses Mounts are detailed in Appendix D

# 6 CAMERA SOFTWARE INTERFACE

## 6.1 Control and Interface

As all the e2v Cameras, the ELIIXA+ CL is delivered with the friendly interface control software COMMCAM.UCL (as "Ultimate Camera Link") which is based on the GenICam standard
COMMCAM recognizes and detects automatically all the UCL Cameras connected on any transport layers (Camera Link or COM ports) of your system.
Once connected to the Camera you have an easy access to all its features. The visibility of these features can be associated to three types of users: Beginner, Expert or Guru. Then you can make life easy for simple users.
Minimum version of CommCam is 2.2.0 in order to recognize the ELIIXA+ 8k/4k color Camera.



## 6.2 Serial Protocol and Command Format

The Camera Link interface provides two LVDS signal pairs for communication between the camera and the frame grabber. This is an asynchronous serial communication based on RS-232 protocol.
The serial line configuration is:

- Full duplex/without handshaking
- 9600 bauds (default), 8-bit data, no parity bit, 1 stop bit. The baud rate can be set up to 115200

### 6.2.1 Syntax

Internal camera configurations are activated by write or readout commands.
The command syntax for write operation is:

w <command_name> <command_parameters><CR>

The command syntax for readout operation is:

r <command_name><CR>

### 6.2.2 Command Processing

Each command received by the camera is processed:

- The setting is implemented (if valid)
- The camera returns ">"<return code><CR>

The camera return code has to be received before sending a new command.


The camera return code has to be received before sending a new command. Some commands are longer than the others : Waiting for the return code ensure a good treatment of all the commands Without saturating the buffer of the camera

**Table 5-1.** Camera Returned Code

| Returned code | meaning |
|---|---|
| >0 | (or ">OK") : All right, the command will be implemented |
| >3 | Error Bad CRC (for write command only) |
| >16 | Invalid Command ID (Command not recognized or doesn't exist) |
| >33 | Invalid Access (the receipt of the last command has failed). |
| >34 | Parameter out of range (the parameter of the last command sent is out of range). |
| >35 | Access Failure (bad communication between two internal devices). |

### 6.2.3 GenICam ready

The CameraLink Standard is not yet compliant with GenICam Standard, but as much as possible, each command of the ELIIXA+ will have its correspondence with the Standard Feature Naming Convention of the GenIcam Standard.
This correspondence is given in parenthesis for each feature/command as the following example :

- **Vendor name** (*DeviceVendorName*) : "**e2v**"

## 6.3 Camera Commands

### 6.3.1 Information

These values allow to indentify the Camera. They can be accessed in CommCam software in the "Info" section

All these values are fixed in factory and can't be changed (shaded) except the Camera User ID which can be fixed by the Customer :

- **Vendor name** (*DeviceVendorName*) : "e2v"
  - ⇨ Read function : "**r vdnm**";
    Returned by the camera : "e2v", string of 32 bytes (including "/0")
  - ⇨ Can not be written

- **Model Name** (*DeviceModelName*) : Internal name for GenICam :
  - ⇨ Read function : "**r mdnm**";
    Returned by the camera : String of 32 bytes (including "/0") :
  - ⇨ Can not be written

- **Device Manufacturer Info** (*DeviceManufacturerInfo*) : Get Camera ID
  - ⇨ Read function : "**r idnb**";
    Returned by the camera : String of 128 bytes (including "/0")
  - ⇨ Can not be written

- **Device Version** (*DeviceVersion*) : Get Camera Hardware version
  - ⇨ Read function : "**r dhwv**";
    Returned by the camera : String of 32 bytes (including "/0")
  - ⇨ Can not be written

- **Device Firmware Version** (*DeviceFirmwareVersion*): Get camera synthetic firmware
  - ⇨ Read function : "**r dfwv**";
    Returned by the camera : String of 16 bytes (including "/0")
  - ⇨ Can not be written

- **Device SFNC Version : 1.5.0**
  These Parameters (Major, Minor, Sub Minor) are only virtual ones in order to give the SFNC compliance of the Camera.

- **Device ID** (*DeviceID*) : Camera Factory identifier ID
  - ⇨ Read function : "**r cust**";
    Returned by the camera : String of 128 bytes (including "/0")
  - ⇨ Write function : "**w cust <idstr>**"

- **Device User ID** (*DeviceUserID*) : Camera user identifier ID
  - ⇨ Read function : "**r cust**";
    Returned by the camera : String of 128 bytes (including "/0")
  - ⇨ Write function : "**w cust <idstr>**"

- **Electronic board ID** (*ElectronicBoardID*) : Get PcB Board ID
  - ⇨ Read function : "**r boid**";
    Returned by the camera : String of 32 bytes (including "/0")
  - ⇨ Can not be written

- **Device Temperature Selector** (*DeviceTemperatureSelector*) : **MainBoard**
  - ⇨ Can not be written

- **Device Temperature** (*DeviceTemperature*) : Get Main Board Temperature
  - ⇨ Read function : "**r temp**";
    Return by the camera : Temperature in Q10.2 format (8 bits signed + 2 bits below comma). Value is between -512 to 511 in °C.

- **Device Serial Port Selection** : Indicates the Serial Port on which the Camera is connected.

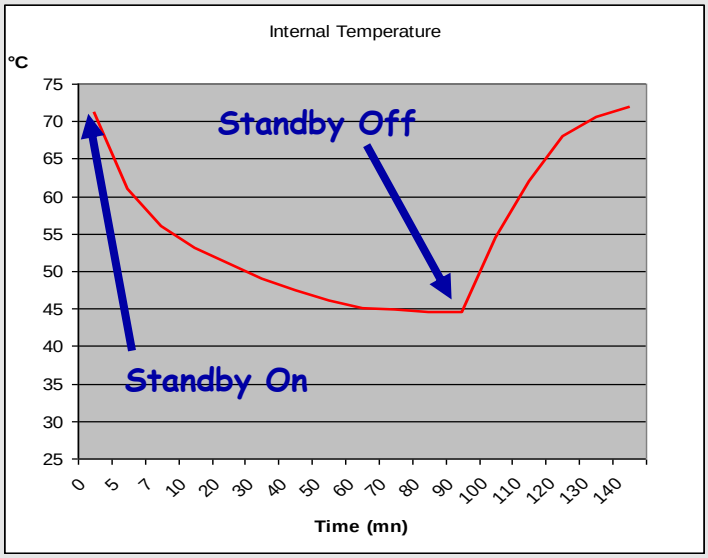
- **Device Serial Port Baud Rate** (*ComBaudRate*): Set the Camera BaudRate
  - ⇨ Read function : "**r baud**";
    Returned by the camera : Value of the Baud Rate
  - ⇨ Write function : "**w baud**" <index> with the index as follows :
    - 1 : 9600 Bauds (default value at power up)
    - 2 : 19200Bauds
    - 6 : 57600Bauds
    - 12 : 115200Bauds

- **Standby Mode** (*Standby*) : Activation of the Standby mode of the Camera
  - ⇨ Read function : "**r stby**";
    Returned by the camera : Boolean.
    - 0 : Disable Standby mode (False)
    - 1 : Enable stanby mode (True)
  - ⇨ Write function : "**w stby <val>**"; <val> is 0 or 1.

**A standby mode, what for ?**

The Standby mode stops all activity on the sensor level. The power dissipation drops down to about **6W**. During the standby mode, **the grab is stopped**

Once the Standby mode turned off, the Camera recovers in less than **1ms** to send images again from the sensor.



- **Camera status** : Get the Camera status register (32bits Integer)
  - ⇨ Read function : "**r stat**";
    Returned by the camera : 32bits integer :

    - **Bit 0** : (*StatusWaitForTrigger*) : True if no trig received from more than 1sec
    - **Bit 1** : (*StatusTriggerTooFast*) : Missing triggers. Trig signal too fast
    - **Bit 2** : (*StatusSensorConnection*) : True is the Sensor pattern is checked as failed.
    - Bit 3, 4, 5, 6, 7 : Reserved
    - **Bit 8** : (*StatusWarningOverflow*) : True is an overflow occurs during FFC or Tap balance processing.
    - **Bit 9** : (*StatusWarningUnderflow*) : True is an underflow occurs during FFC or Tap balance processing
    - Bits 10 : Reserved
    - **Bits 11** : Scrolling Direction : 0 = Forward, 1 = Reverse. Updated only by external CC3 (CameraLink)
    - Bits, 12, 13, 14, 15 : Reserved
    - **Bit 16** : (*StatusErrorHardware*) : True if hardware error detected
    - Bits 17 to 31 : Reserved

### 6.3.2 Image Format

- **Sensor Width** (*SensorWidth*) : Get the physical width of the Sensor. This value is available in the CommCam "Image Format Control" section :
  - ⇨ Read function : "**r snsw**";
    Return by the sensor : Integer 8192.
  - ⇨ Can not be written;

- **Sensor Height** (*SensorHeight*) : Get the physical height of the Sensor. This value is available in the CommCam "Image Format Control" section :
  - ⇨ No Access. Virtual command in xml"; Value always = 1

- **Width Max** (*WidthMax*) : Get the Maximum Width of the Sensor. This value is available in the CommCam "Image Format Control" section :
  - ⇨ No Access. The value is mapped on "SensorWidth"

- **Height Max** (*HeigthMax*) : Get the Maximum height of the Sensor. This value is available in the CommCam "Image Format Control" section :
  - ⇨ No Access. Virtual command in xml"; Value always = 1

- **Output mode** (*OutputMode*) : Set the CameraLink Output mode (refer also to Chapter : CameraLink Output Configuration). This command is available in the CommCam "Image Format Control" section :
  - ⇨ Read function : "**r mode**";
    Returned by the camera : Output mode from 0 to 3 (see table below).
  - ⇨ Write function : **"w mode"** <value> :
    detailed in the table below :

| Modes | Connector CL1 | Connector CL2 | Mode value |
|---|---|---|---|
| **Base 3 Channels RGB 8 bits** | 3 x 8 bits | - | **0** |
| **Dual Base 3 Channels RGB 8 bits** | 3 x 8 bits | 3 x 8 bits | **1** |
| **Full 8 Channels 8bits** | 8 x 8 bits | | **2** |
| **Full+ 10 Channels 8bits** | 10 x 8 bits | | **3** |

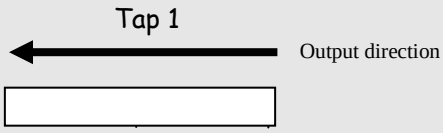
  - "0" : BaseRGB8bits
  - "1" : DualBaseRGB8bits
  - "2" : RawFull8Outputs8bits
  - "3" : RawFullPlus10Outputs8bits

***Switching between Sensor modes***

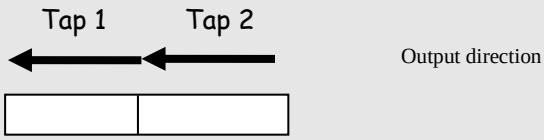
*The "Raw" output modes (8 or 10Taps) are achieved by loading another FPGA firmware. Then the switch time between Base or Dual Base modes and Full 8taps or Full+ 10Taps mode is about several seconds (**maximum 9s**).*

## Structure of the Camera Link Channels for interfacing

- **Base Mode :** 1 Tap RGB 24 bits (3 channels), outputted from Left to Right.
  **3x4096 pixels (1 RGB Tap) in True Color Mode**
  **3x8192 pixels (1 RGB Tap) in Full Definition Modes**

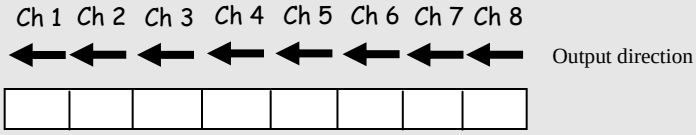



- **Dual Base Mode :** 2 Taps RGB 24 bits (2 x 3 channels), outputted from Left to Right
  **2 x (3x2048) pixels (2 RGB Taps) in True Color Mode**
  **2 x (3x4096) pixels (2 RGB Taps) in Full Definition Modes**

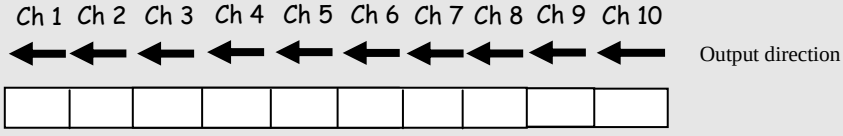



**The two following output modes are considered as "Monochrome" on the Frame Grabber side.**
**A specific interpolation on the application level is required to get back the color buffer.**

- **FULL Mode :** 8 Taps Separate, outputted from Left to Right.
  **8x1536 pixels each Channel in True Color Mode : 12290 pixels total.**
  **- 4096 Green, 4096 Red and 4096 Blue pixels**
  **8x2048 pixels each Channel in Full Definition Modes : 16380 pixels total.**
  **- 8192 Green, 4096 Red and 4096 Blue pixels**




- **FULL+ (Deca) Mode :** 10 Taps Separate, outputted from Left to Right.
  **10x1229 pixels each Channel in True Color Mode : 12290 pixels total.**
  **- 4096 Green, 4096 Red and 4096 Blue pixels.** The last pixel of Tap9 and Tap10 are valid but black.
  **10x1638 pixels each Channel in Full Definition Modes : 16380 pixels total.**
  **- 8190 Green, 4095 Red and 4095 Blue pixels**

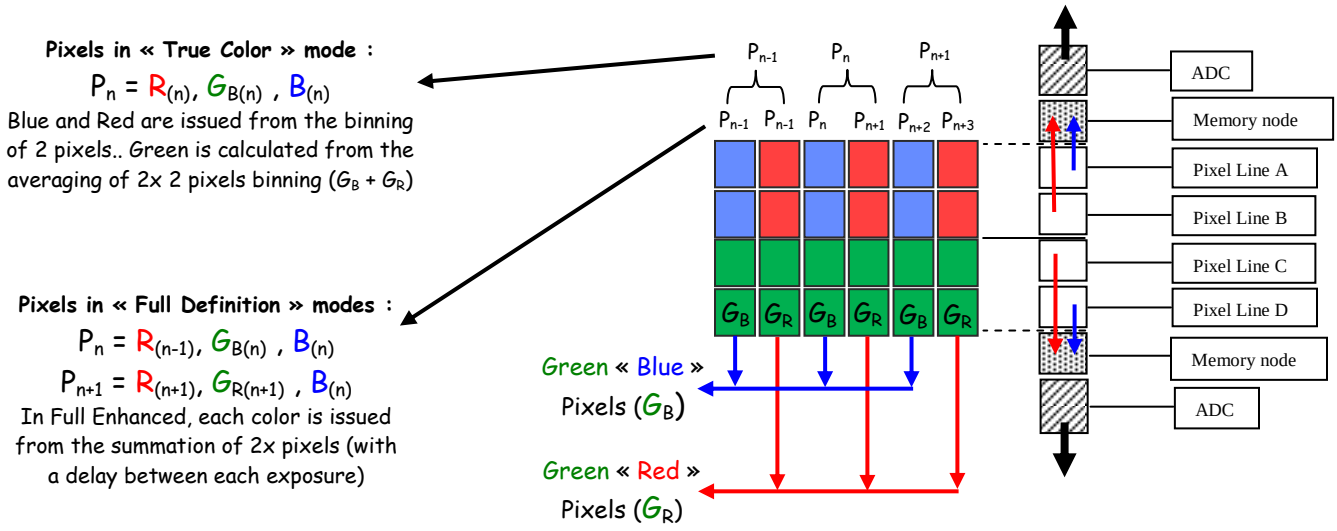


- **Output Frequency** (*OutputFrequency*) : Set the CameraLink Data Output Frequency. This value is available in the CommCam "Image Format Control" section :
  - ⇨ Read function : "**r clfq**";
    Return by the Camera : Frequency from 0 to 5
  - ⇨ Write Function : "**w clfq** <value>"
    - "0" : 85MHz (default).
    - "1" : 60MHz.
    - "2" : 65MHz.
    - "3" : 70MHz.
    - "4" : 75MHz.
    - "5" : 80MHz.

- **Sensor Mode** (*SensorMode*) : Defines the number of Line used on the Sensor. This command is available in the CommCam "Image Format Control" section :

⇨ Read function : "**r smod**";
Returned by the camera : Integer from 0 to 2

⇨ Write function : "**w smod**" <value> :
- "0" : True Color
- "1" : Full definition Single
- "2" : Full Definition Enhanced



***Full Exposure Control***

*As the « Full Definition Enhanced » color mode is performing an internal Time delay exposure on the Four Color lines, normally, the variation of the Exposure time should not possible in this sensor mode.*
*Thanks to an e2v licensed solution, two of the Exposure controlled mode (Ext Trig with internal or External exposure control) are still available in this color mode.*

*The "Free Run" synchronization mode (line Trigger and Exposure time controlled internally) is not available in the "FDE" color mode.*

- **Scan Direction** (*ScanDirection*) : Set the scan direction for the sensor. This value is available in the CommCam "Image Format Control" section :
  - ⇨ Read function : "**r scdi**";
    Return by the Camera : 0, 1 or 2 (Forward/reverse/external)

  - ⇨ Write function : "**w scdi** <value>";
    - "0" : Forward.
    - "1" : Reverse
    - "2" : Externally controlled (by CC3 of the CameraLink Sync signals)

Forward/reverse information has to be set correctly For the re-ordering of the colors.

*The Forward direction is defined as detailed beside* :

**Note** : The delay for the Camera to take in account a change in the ScanDirection value is minimum **100ms**.

This information can be set dynamically by using the **CC3 Trig signal** of the CameraLink connector (change the direction "on the fly").
In these case, the Trigger level signification is :
- "0" : Forward.
- "1" : Reverse

- **Test Image Selector** (*TestImageSelector*) : Defines if the data comes from the Sensor or the FPGA (test Pattern). This command is available in the CommCam "Image Format" section :

  ⇨ Read function : "**r srce**";
  Returned by the camera : "0" if Source from the Sensor and "1 to 5" if test pattern active
  ⇨ Write function : "**w srce**" <value> :
  - "0" : To switch to CCD sensor image
  - "1" : Grey Horizontal Ramp (Fixed) : **See AppendixA**
  - "2" : White Pattern (Uniform white image : 255)
  - "3" : Grey Pattern (Uniform middle Grey : 128 on each color))
  - "4" : Black Pattern (Uniform white image : 0)
  - "5" : Grey vertical Ramp (moving)

  The test pattern is generated in the FPGA : It's used to point out any interface problem with the Frame Grabber.
  When any of the Test pattern is enabled, the whole processing chain of the FPGA is disabled.

- **Color Selection** :
  Disables each of the 3 colors.. This command is available in the CommCam "Image Format" section.

  ⇨ Read function : "**r cold**";
  Returned by the camera : Integer corresponding to one of the 3 different step values :
  - Bit 0 : Red color disabled if set to 1
  - Bit 1 : Blue color disabled if set to 1
  - Bit 2 : Green (both $Green_{Red}$ and $Green_{Blue}$) color disabled if set to 1

  ⇨ Write function : "**w cold**" <val> ;

### 6.3.3 Acquisition Control

This section deals with all the Exposure, Line period and synchronisation modes

- **Synchronisation Mode** (TriggerPreset) : Timed or Triggered, it defines how the grabbing is synchronized. This command is available in the CommCam "Acquisition Control" section :

  ⇨ Read function : "**r sync**";
  Returned by the camera :
  - "**0**" : Internal Line Trigger with Exposure time Internally Controlled (Free Run). *Not available when Sensor mode is set in "Full Definition Enhanced"*
  - "**1**" : External Trigger with Exposure Time Internally Controlled. *Available also when Sensor mode is set in "Full Definition Enhanced".*
  - "**2**" : External Trigger with maximum Exposure time
  - "**3**" : One External with Exposure Time Externally Controlled. The same Trigger signal defines the line period and its low level defines the exposure time. *Available also when Sensor mode is set in "Full Definition Enhanced".*
  - "**4**" : Two External Triggers with Exposure Time Externally Controlled : CC2 defines the start of the exposure (and also the start Line) and CC1 defines the Stop of the exposure. *Not available when Sensor mode is set in "Full Definition Enhanced".*
  - "**5**" : Internal Line Trigger with maximum Exposure Time

  ⇨ Write function : "**w sync**" <value>

*The Timing diagrams associated to each Synchronization mode and the Timing values associated are detailed in the APPENDIX B of this document.*

- **Exposure time** (*ExposureTime*): Defines the exposure time when set in the Camera. This command is available in the CommCam "Acquisition Control" section :

  ⇨ Read function : "**r tint**";
  Returned by the camera : Integer from 15 to 65535 (=1,5µs to 6553,5µs by step o 0,1µs)

  ⇨ Write function : "**w tint**" <value> ;

  This value of exposure time is taken in account only when the synchronisation mode is "free run" (0) or "Ext Trig with Exposure time set" (1). Otherwise it's ignored.

Due to the limitation of the timing pixel inside the sensor, the Exposure time has to be set by taking in account the limitation detailed in the APPENDIX B of this document.
The **Minimum exposure time** which can be set is **1,5µs**

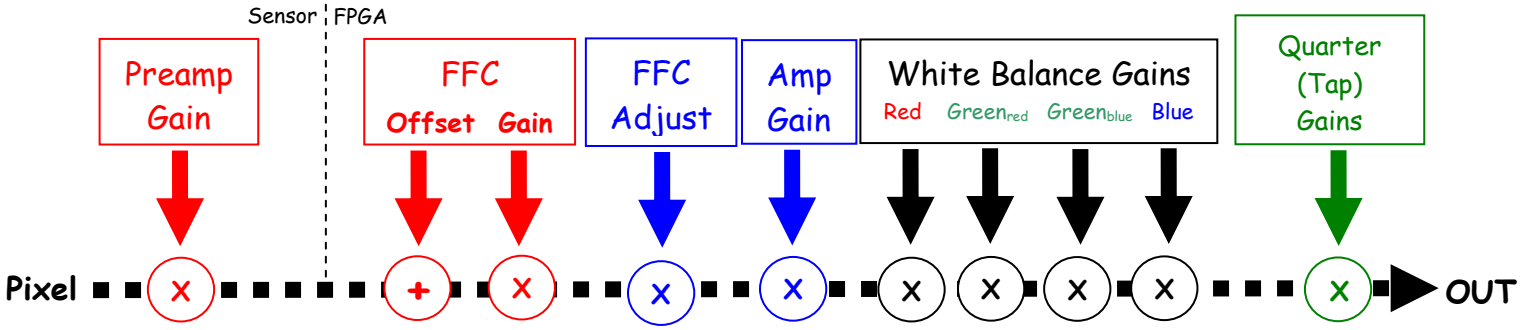
- **Line Period** (*LinePeriod*) : Defines the Line Period of the Camera in Timed mode. This command is available in the CommCam "Acquisition Control" section :

  ⇨ Read function : "**r tper**";
  Returned by the camera : Integer from 151 to 65536 (=15,1µs to 6553,6µs by step o 100ns)
  ⇨ Write function : "**w tper**" <value> ;

  The line period is active only in Free Run modes. It's also disabled if in this mode, the Integration time is set higher than the Line Period.


The Tables of the minimum Line Period (Max Line Rate) versus the Data rate and the output mode chosen are given in Appendix C (Chap. 9.2) of this document.

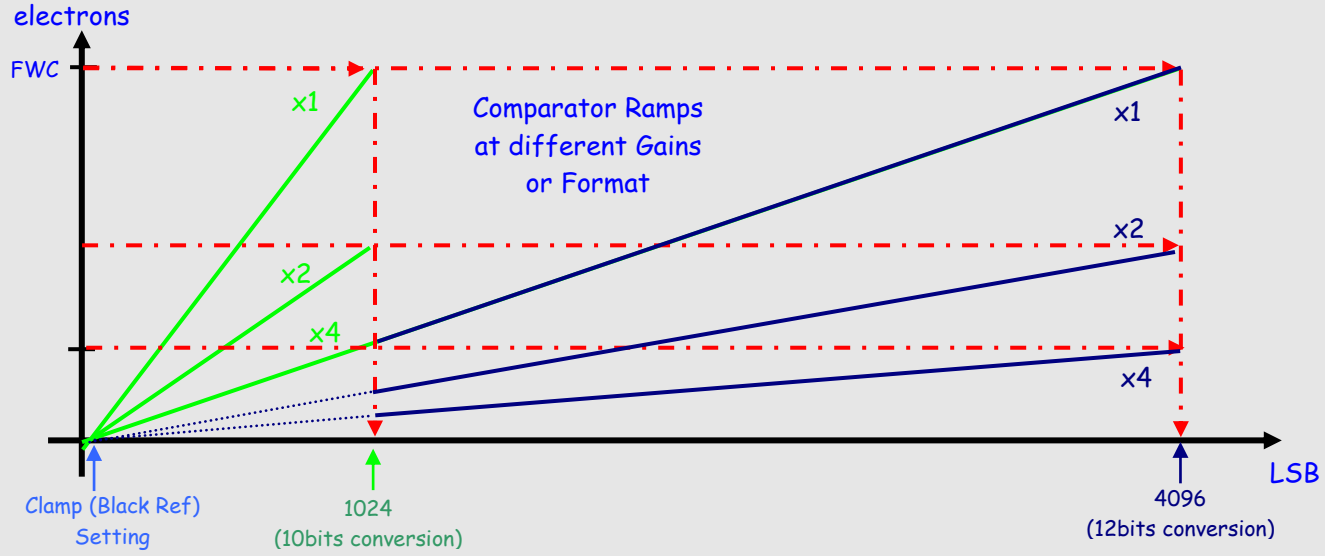
## 6.3.4 Gain and Offset




**Action on whole line** .....
**Action per pixel** .....
**Action per Sensor's Quarter** ..... *(Not available on BA0 models)*
**Action per Color** .....

### Analog Gain in the ADC

The only analog Gain available in the ELIIXA+ is located at the sensor level, in the ADC converter.
This "Preamp Gain" is in fact a variation of the ramp of the comparator of the ADC.
Then 3 Values are available : x1, x2 and x4. A gain x1 in a 12 bits conversion is equivalent to x4 in 10 bits.




- **Preamp Gain** : (*Gain* with *GainSelector*= *AnalogAll*)
  Set the Pre-amplification Gain. This command is available in the CommCam "Gain & Offset" section.

  ⇨ Read function : "**r pamp**";
  Returned by the camera : Integer corresponding to one of the 3 different step values :
  - 0 : x1 (0dB)
  - 1 : x2 (6dB)
  - 2 : x4 (12dB)

  ⇨ Write function : "**w pamp**" <int> ;

- **Gain:** (*Gain* with *GainSelector*= *GainAll*)
  Set the Amplification Gain. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gain**";
    Returned by the camera : Value from 0 to 6193 corresponding to a Gain range of 0dB to +8dB calculated as following : Gain(dB) = 20.log(1+ Gain/4096).
  - ⇨ Write function : "**w gain**" <int> ;

- **Tap Gain (*Gain* with *GainSelector*= *Tap*X) :**
  - ⇨ Read function : "r **fga**<tap>"; <tap> is 1 or 2
    Returns the Gain value for the tap. Ex : "*r fga1*" returns Gain value Tap1.
  - ⇨ Write function : "w **fga**<tap> <value>"
    - **<tap>** : 1 or 2
    - **<value>** : from -128 to +127 by step of 1 (0,0021dB each step)

- **Digital Gain** (*Gain* with *GainSelector*=*DigitalAll*) : Set the global Digital Gain. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gdig**";
    Returned by the camera : Integer value from 0 to 255. The corresponding Gain is calculated as 20log(1+val/64) in dB
  - ⇨ Write function : "**w gdig**" <int> ;

- **Digital Offset** (*BlackLevelRaw* with *BlackLevelSelector*=*All*) : Set the global Digital Offset. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r offs**";
    Returned by the camera : Value from –4096 to +4095 in LSB
  - ⇨ Write function : "**w offs**" <int> ;

- **Tap Balance Gains Enable Switch (*TapBalanceGainEnable*) :**
  - ⇨ Read function : "r **fgae**";
    Returns the Tap Balance Status.
  - ⇨ Write function : "w **fgae** <val>" with <val> : 0 or 1
    - 0 : Disables the Tap Balance Gains
    - 1 : Enables the Tap Balance Gains

#### 6.3.4.1 White Balance

As described in chapter 6.3.2, the structure of the sensor differentiates Green pixels facing Blue or Red pixels. Then the white balance is associated with 4 color Gains :

- Red Gain
- $Green_{Red}$ Gain
- $Green_{Blue}$ Gain
- Blue Gain

The Color Selection or enabling (Image Format Chapter) can affect the way you're performing the white balance : For example, if you disable the Blue and the Red color, the "White Balance" will be performed only between the two Green Gains.

The dissociation of Green (blue) and Green (Red) is justified by the possible difference of response of the two types of Green because of their respective neighbor color influence and then the necessity to tune them separately.

As usual, for a perfect White balance, provide to the Camera a non-saturating white (gray) target in the center of the sensor.
The White balance has to be performed after the Flat Field Correction as each color is performing its own FFC with its own reference.
In any case, the best tuning of the Camera Gains is performed from the left to the right of the Gain Chain described above : Preamp Gain first and quarter Gains last (if required).

The Following Gains are enabled by the White balance Enable switch :

- **Digital Red Gain** (*Gain* with *GainSelector*=*DigitalRed*) : Set the Red Gain for the white balance. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gwbr**";
    Returned by the camera : Integer value from 0 to 1548. The corresponding Gain is calculated as 20.log( 1 + <val>/1024) in dB
  - ⇨ Write function : "**w gwbr**" <val> ;

- **Digital Blue Gain** (*Gain* with *GainSelector*=*DigitalBlue*) : Set the Blue Gain for the white balance. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gwbb**";
    Returned by the camera : Integer value from 0 to 1548. The corresponding Gain is calculated as 20.log( 1 + <val>/1024) in dB
  - ⇨ Write function : "**w gwbb**" <val> ;

- **Digital $Green_{Red}$ Gain** (*Gain* with *GainSelector*=*DigitalGreenR*) : Set the $Green_{Red}$ Gain for the white balance. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gwbg**";
    Returned by the camera : Integer value from 0 to 1548. The corresponding Gain is calculated as 20.log( 1 + <val>/1024) in dB
  - ⇨ Write function : "**w gwbg**" <val> ;

- **Digital Green$_{Blue}$ Gain** (*Gain* with *GainSelector=DigitalGreenB*) : Set the Green$_{Blue}$ Gain for the white balance. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "**r gwbj**";
    Returned by the camera : Integer value from 0 to 1548. The corresponding Gain is calculated as 20.log( 1 + <val>/1024) in dB
  - ⇨ Write function : "**w gwbj**" <val> ;

- **White Balance Enable Switch** (*WhiteBalanceEnable*) : Enables the White Balance Gains. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ Read function : "r **gwbe**";
    Returns the White Balance Gain Enable Status.
  - ⇨ Write function : "w **gwbe** <val>" with <val> : 0 or 1
    - 0 : Disables the White Balance Gains
    - 1 : Enables the White Balance Gains

- **White Balance Calibration Control** (*AutoWhiteBalanceStart*) : Launch or abort of the White Balance process for the RGB Gains calculation. This command is available in the CommCam "Gain & Offset" section :
  - ⇨ No Read Function
  - ⇨ Write function :
    - "**w awbc 1**" : Launch the White Balance Calibration Process.
    - "**w awbc 0**" : Abort the White Balance Calibration Process.

## 6.3.5 Flat Field Correction

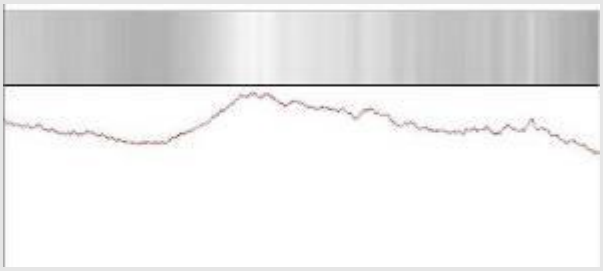
### How is performed the Flat Field Correction ?

***What is the Flat Field correction (FFC) ?***

The Flat Field Correction is a digital correction on each pixel which allows :

- To correct the Pixel PRNU (Pixel Response Non Uniformity) and DSNU (Dark Signal Non Uniformity)
- To Correct the shading due to the lens
- To correct the Light source non uniformity

**Before**



**After**

***How is calculated / Applied the FFC ?***

The FFC is a digital correction on the pixel level for both Gain and Offset.

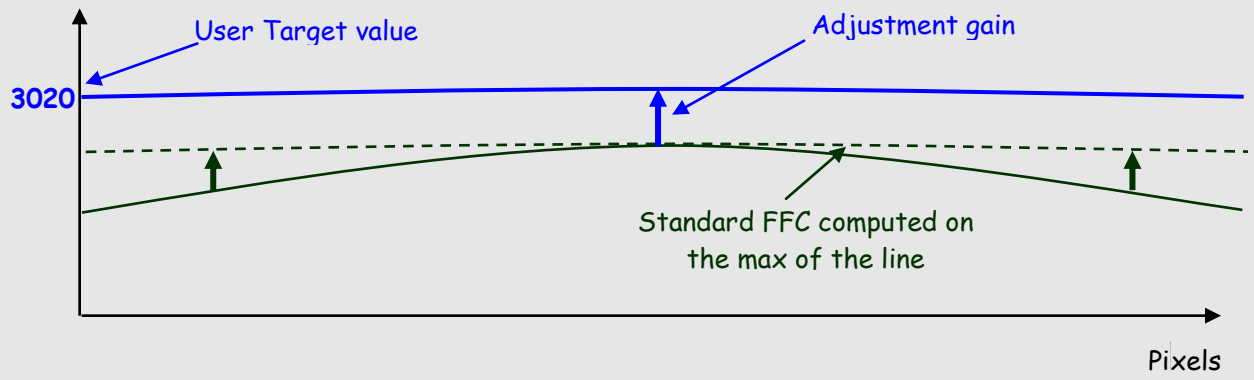
- Each Pixel is corrected with :
  - An Offset on 8 bits (Signed Int 5.3). They cover a dynamic of ±16LSB in 12bits with a resolution of 1/8 LSB 12bits.
  - A Gain on 12 bits with a max gain value of x5. Gain : U12 (1+Gain/1024) => x1 to x3.999 by step of 1/1024

The calculation of the new pixel value is : **P' = ( P + Off).(1 + Gain/1024)**

The FFC is processed independently for each Color (Red, Blue, $Green_{Blue}$, $Green_{Red}$). A white balance is required after any FFC process.

The FFC is always processed with the max pixel value of the line as reference. If enabled, the FFC adjust module (located at the output of the FFC module) calculates the adjustment gain to reach the target defined by the User.

When the FFC result is saved in memory, the adjust gain and target are saved in the same time in order to associate this gain value with the FFC result.




***How to perform the Flat Field Correction ?***

**FPN/DSNU Calibration**

- Cover the lens
- Launch the FPN Calibration : Grab and calculation is performed in few seconds
- Offset format : S9.1 => -256..+255.5 by step of ½

**PRNU Calibration**

The User must propose a white/gray uniform target to the Camera (not a fixed paper).
The Gain/Light conditions must give a non saturated image in any Line.
The Camera must be set in the final conditions of Light/ Gain and in the final position in the System.
I f required, set a user target for the FFC adjust and enable it.

- White uniform (moving) target
- Launch the FFC
- Enable the FFC
- You can save the FFC result (both FPN+PRNU in the same time) in one of the x4 FFC User Banks.
- The user target and Gain are saved with the associated FFC in the same memory.

**Advices**

The ELIIXA+ Cameras have 4 x FFC Banks to save 4 x different FFC calibrations. You can use this feature if your system needs some different conditions of lightning and/or Gain because of the inspection of different objects : You can perform one FFC per condition of Gain/setting of the Camera ( 4 Max) and recall one of the four global settings (Camera Configuration + FFC + Line Balance) when required.

#### 6.3.5.1 Activation

- **FFC Activation** (*FFCEnable*) : Enable/disable the Flat Field Correction. This command is available in the CommCam "Flat Field Correction" section :

  ⇨ Read function : "**r ffcp**" : Returns the FFC Status (0 if disabled, 1 if enabled)
  ⇨ Write function :
  - "**w ffcp 1**" : Enable the FFC.
  - "**w ffcp 0**" : Disabled the FFC

#### 6.3.5.2 Automatic Calibration

- **FPN/DSNU Calibration :**

  - **FPN Calibration Control** (*FPNCalibrationCtrl*) : Launch or abort of the FPN process for the Offsets calculation. These commands are available in the CommCam "Flat Field Correction / Automatic Calibration " section :

    ⇨ Read function : "**r calo**" : Returns the FPN Calculation Process Status (0 if finished, 1 if processing)
    ⇨ Write function :
    - "**w calo 1**" : Launch the FPN Calibration Process.
    - "**w calo 0**" : Abort the FPN Calibration Process.
  - **FPN Coefficient Reset** (*FPNReset*) : Reset the FPN (Offsets) coefficient in Memory. This command is available in the CommCam "Flat Field Correction / Manual Calibration " section :

    ⇨ Write function : "**w rsto 0**" : Reset (set to 0) the FPN coefficients in memory. This doesn't affect the FFC User Memory Bank but only the active coefficients in Memory.

- **PRNU Calibration :**
  - **PRNU Calibration Control** (*FFCCalibrationCtrl*) : Launch or abort of the PRNU process for the Gains calculation. This command is available in the CommCam "Flat Field Correction / Automatic Calibration " section :

    ⇨ Read function : "**r calg**" : Returns the PRNU Calculation Process Status (0 if finished, 1 if processing)
    ⇨ Write function :
    - "**w calg 1**" : Launch the PRNU Calibration Process.
    - "**w calg 0**" : Abort the PRNU Calibration Process.

  - **PRNU coefficient Reset** (*PRNUReset*) : Reset the PRNU (Gains) coefficient in Memory. This command is available in the CommCam "Flat Field Correction / Manual Calibration " section :

    ⇨ Write function : "**w rstg 0**" : Reset (set to "x1") the PRNU coefficients in memory. This doesn't affect the FFC User Memory Bank but only the active coefficients in Memory.

Some Warnings can be issued from the PRNU/FPN Calibration Process as "pixel Overflow" of "Pixel Underflow" because some pixels have been detected as too high or too low in the source image to be corrected efficiently.
The Calculation result will be proposed anyway as it's just a warning message.
The Status Register is the changed and displayed in CommCam "Status" section :
Register status is detailed chap §6.3.3.

### 6.3.5.3 Manual Flat Field Correction

The FFC Coefficients can also be processed outside of the Camera or changed manually by accessing directly their values in the Camera : This is the "Manual" FFC.

In CommCam, the User can access to a specific interface by clicking on "click for extended control" in both "Manual FFC calibration" and "Manual FPN calibration sections" :



This will allow the user to upload/download out/in the Camera the FFC coefficients in/from a binary or text file that can be processed externally.

It is recommended to setup the baud rate at the maximum value possible (115000 for example) otherwise the transfer can take a long time.

- **FPN coefficients modification :** Direct access to the FPN coefficients for reading or writing.
  The FPN coefficients are read packets of x128 coefficients : Format: S9.1 => -256..+255.5 step ½

  ⇨ Read function : "**r ffco <addr>**" : Read 128 consecutive FPN user coefficients starting from **<addr>** address. Returned value is in hexadecimal, without space between values (one unsigned short per coefficient).
  - Coefficient from address 0 to 4095 are for red pixels
  - Coefficient from address 4096 to 8191 are for blue pixels
  - Coefficient from address 8192 to 12287 are for $green_{Red}$ pixels
  - Coefficient from address 12288 to 163837 are for $green_{Blue}$ pixels

  ⇨ Write function :" **w ffco <addr><val>** : Write 128 consecutive FPN user coefficients starting from the **<addr>** address. **<val>** is the concatenation of individual FPN values, without space between the values (one unsigned short per coefficient).

- **PRNU coefficients modification :** Direct access to the PRNU coefficients for reading or writing.
  The PRNU coefficients are read packets of x128 coefficients. Format : U1.13 (1+coeff/8192) => x1 to x2.999877 by step of 1/8192

  ⇨ Read function : "**r ffcg <addr>**" : Read 128 consecutive PRNU user coefficients starting from **<addr>** address. Returned value is in hexadecimal, without space between values (one unsigned short per coefficient).
  - Coefficient from address 0 to 4095 are for red pixels
  - Coefficient from address 4096 to 8191 are for blue pixels
  - Coefficient from address 8192 to 12287 are for $green_{Red}$ pixels
  - Coefficient from address 12288 to 163837 are for $green_{Blue}$ pixels

  ⇨ Write function :" **w ffcg <addr><val>** : Write 128 consecutive PRNU user coefficients starting from the **<addr>** address. **<val>** is the concatenation of individual PRNU values, without space between the values (one unsigned short per coefficient).

#### 6.3.5.4 FFC User Bank Management

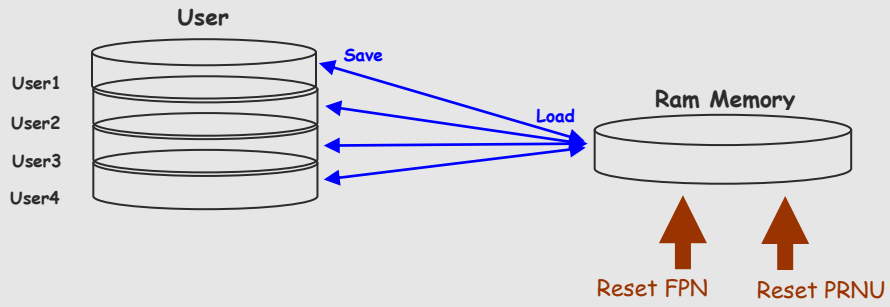
The new-processed FFC values can be saved or restored in/from 4 x User banks.
Both Gains and Offsets in the same time but also the FFC Adjust User target and associated gain.
These functions are available in the Flat Field correction/Save & Restore FFC section :

- **Restore FFC from Bank** (*RestoreFFCFromBank*) : Restore the FFC from a Bank in the current FFC.
  ⇨ Read function : "**r rffc**" : Get the current FFC Bank used
  Returned by the camera : 0 for Factory bank or 1 to 4 for User banks
  ⇨ Write function : "**w rffc <val>**" : Bank <val> 1 to 4 for User banks

  Note : Factory means neutral FFC (no correction).

- **Save FFC in User Bank** (*SaveFFCToBank*) : Save current FFC in User Bank
  ⇨ Can not de read
  ⇨ Write function : "**w sffc <val>**" : User bank <val> if from 1 to 4.

## FFC User Bank Usage




**At the power up :**
- Last User Bank used is loaded in RAM

**Reset a User bank :**
- Reset the RAM (FPN/PRNU individually)
- Save in the bank to reset

## 6.3.6 Statistics and Line Profile

This function allows the User to get some statistics on a pre-defined ROI. On request, the Camera acquires and then calculates some key values as the min, the max, the average or the standard deviation in this Region of Interest.
The grab and calculation command and also the collection of the results is not performed in real time as it is done through the serial connection.
This function and the results are available in CommCam in the "Line Profile Average" Section :

- **Line Profile average measurement** (*LineAverageProfile*) : Control the grab and computation of the statistics.
  ⇨ Read function : "**r pixs**" : Get the status of the calculation
  Returned by the camera : 0 : finished, 1: running
  ⇨ Write function :
    - "**w pixs 1**" : Start the accumulation and then the computing
    - "**w pixs 0**" : Abort the computing.

The Calculated values are detailed as following :

**For the Red Pixels**

- **Pixel average Value** (*RedPixelROIMean*) : Average Red level value calculated on whole Region of interest
  ⇨ Read function : "**r pavr**" : Get the average value
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Standard deviation** (*RedPixelROIStandardDeviation*) : standard deviation of all the Red pixel level values of Region of interest
  ⇨ Read function : "**r pstr**" : Get the standard deviation
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Min value** (*RedPixelROIMin*) : Minimum Red level pixel value on the whole region of interest.
  ⇨ Read function : "**r pmir**" : Get the Minimum value
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Max Value** (*RedPixelROIMax*) : Maximum Red level pixel value on the whole region of interest
  ⇨ Read function : "**r pmar**" : Get the maximum value
  Returned by the camera : Unsigned format value : U12.4

**For the Blue Pixels**

- **Pixel average Value** (*BluePixelROIMean*) : Average Blue level value calculated on whole Region of interest
  ⇨ Read function : "**r pavb**" : Get the average value
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Standard deviation** (*BluePixelROIStandardDeviation*) : standard deviation of all the Blue pixel level values of Region of interest
  ⇨ Read function : "**r pstb**" : Get the standard deviation
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Min value** (*BluePixelROIMin*) : Minimum Blue level pixel value on the whole region of interest.
  ⇨ Read function : "**r pmib**" : Get the Minimum value
  Returned by the camera : Unsigned format value : U12.4
- **Pixel Max Value** (*BluePixelROIMax*) : Maximum Blue level pixel value on the whole region of interest
  ⇨ Read function : "**r pmab**" : Get the maximum value
  Returned by the camera : Unsigned format value : U12.4

**For the $Green_{Red}$ Pixels**

- **Pixel average Value** (*Green(r)BluePixelROIMean*) : Average $Green_{Red}$ level value calculated on whole Region of interest
  - ⇨ Read function : "**r pavg**" : Get the average value
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Standard deviation** (*Green(r)PixelROIStandardDeviation*) : standard deviation of all the $Green_{Red}$ pixel level values of Region of interest
  - ⇨ Read function : "**r pstg**" : Get the standard deviation
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Min value** (*Green(r)PixelROIMin*) : Minimum $Green_{Red}$ level pixel value on the whole region of interest.
  - ⇨ Read function : "**r pmig**" : Get the Minimum value
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Max Value** (*Green(r)PixelROIMax*) : Maximum $Green_{Red}$ level pixel value on the whole region of interest
  - ⇨ Read function : "**r pmag**" : Get the maximum value
    Returned by the camera : Unsigned format value : U12.4

**For the $Green_{Blue}$ Pixels**

- **Pixel average Value** (*Green(b)BluePixelROIMean*) : Average $Green_{Blue}$ level value calculated on whole Region of interest
  - ⇨ Read function : "**r pavj**" : Get the average value
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Standard deviation** (*Green(b)PixelROIStandardDeviation*) : standard deviation of all the $Green_{Blue}$ pixel level values of Region of interest
  - ⇨ Read function : "**r pstj**" : Get the standard deviation
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Min value** (*Green(b)PixelROIMin*) : Minimum $Green_{Blue}$ level pixel value on the whole region of interest.
  - ⇨ Read function : "**r pmij**" : Get the Minimum value
    Returned by the camera : Unsigned format value : U12.4
- **Pixel Max Value** (*Green(b)PixelROIMax*) : Maximum $Green_{Blue}$ level pixel value on the whole region of interest
  - ⇨ Read function : "**r pmaj**" : Get the maximum value
    Returned by the camera : Unsigned format value : U12.4

- **Pixel access Line number** (*PixelAccessLineNumer*) : Set the number of lines to accumulate.
  ⇨ Read function : "**r pixl**" : Get the number of line
  Returned by the camera : 1, 256, 512 or 1024
  ⇨ Write function : "**w pixl <val>**" : Set the number of lines. <val> is 1, 256, 512 or 1024.

- **Pixel ROI Start** (*PixelRoiStart*) : Set the Region of Interest start position.
  ⇨ Read function : "**r prod**" : Get the starting pixel
  Returned by the camera : value between 0 and 16383
  ⇨ Write function : "**w prod <val>**" : Set the starting pixel. <val> is between 0 and 16383

- . **Pixel ROI Width** (*PixelRoiWidth*) : Set the Width of the Region of Interest.
  ⇨ Read function : "**r prow**" : Get the width in pixel
  Returned by the camera : value between 1 and 16384
  ⇨ Write function : "**w prow <val>**" : Set the ROI width in pixels. <val> is between 1 and 16384

After performing a line profile measurement, all the values computed which are described below are not refreshed automatically in CommCam : You have to right-click on each value and ask for an individual refresh.

### 6.3.7 Privilege Level

There are 3 privilege levels for the camera :
- Factory (0) : Reserved for the Factory
- Integrator (1) : Reserved for system integrators
- User (2) : For all Users.

The Cameras are delivered in Integrator mode. They can be locked in User mode and a specific password is required to switch back the Camera in Integrator mode. This password can be generated with a specific tool available from the hotline (hotline-cam@e2v.com)

This function is available in the Privilege section :

- **Privilege level Management** (*PrivilegeLevel*) : Get the current Camera privilege level..
  ⇨ Read function : "**r lock**" : Get the current privilege
  Returned by the camera : 0 to 2
  ⇨ Write function : "**w lock <val>**" : <val> is as follow
    - **2** : Lock the Camera in Integrator or "privilege User"
    - **<computed value>** : Unlock the Camera back in Integrator mode

## 6.3.8 Save & Restore Settings

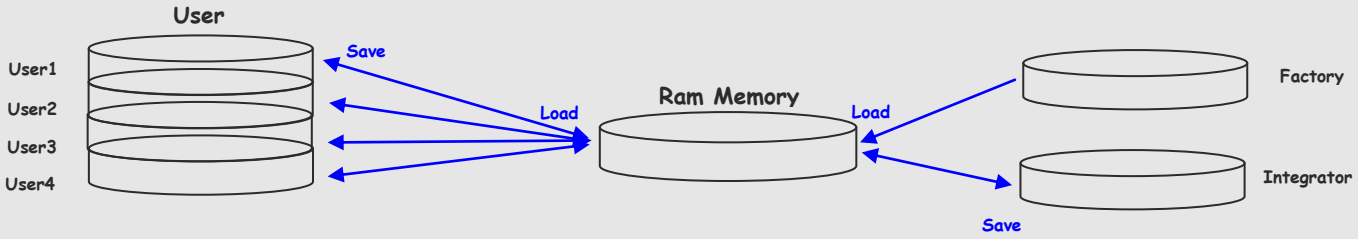
The settings (or Main configuration) of the Camera can be saved in 4 different User banks and one Integrator bank. This setting includes also the FFC and LUT enable
This function is available in the Save & Restore Settings section :

- **Load settings from Bank :** Allows to restore the Camera settings.
  - ⇨ Read function : "**r rcfg**" : Get the current Tap Bank in use
  - ⇨ Write function : "**w rcfg <val>**" : Load settings from bank <val> (0: Factory , 1 to 4 for Users, 5 for Integrator)

- **Save settings to Bank :** Allows to save the Camera settings in User or Integrator Bank
  - ⇨ Write function : "**w scfg <val>**" : Save the current settings in the User bank <val> (1 to 4 for User, 5 for Integrator)

The integrator bank (User Set5) can be written only if the Camera is set in integrator mode (Privilege level = 1). This integrator bank can be used as a « Factory default » by a system integrator.

**Configuration Bank Usage**



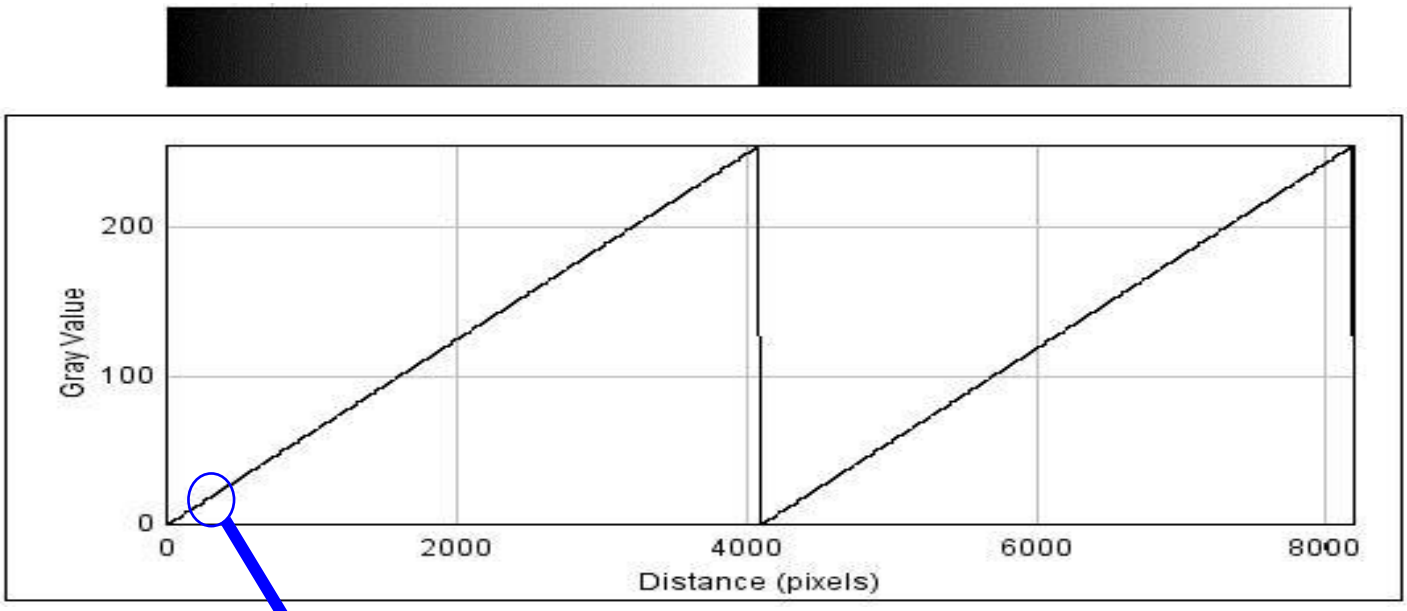

**At the power up :** Last User Bank used is loaded in RAM
"Integrator" Bank (5) can be locked by switching the Camera in "User" mode (cf : Privilege feature). Then it can't be saved any more without switching back the Camera in "Integrator" Mode.

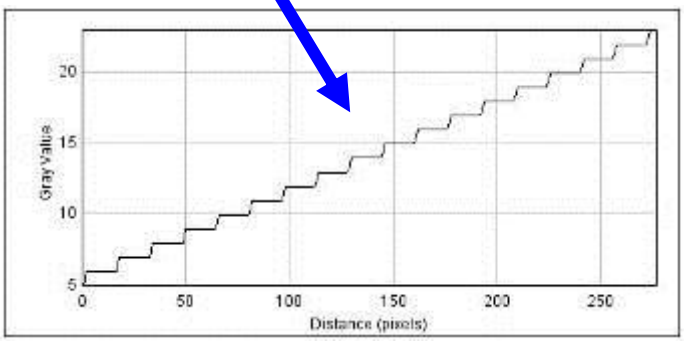
# 7 APPENDIX A: Test Patterns

## *7.1 Test Pattern 1: Vertical wave*

The Test pattern 1 is a vertical moving wave : each new line will increment of 1 gray level in regards with the previous one : level reaches 255 before switching down to 0

## *7.2 Test Pattern 2: Fixed Horizontal Ramps*

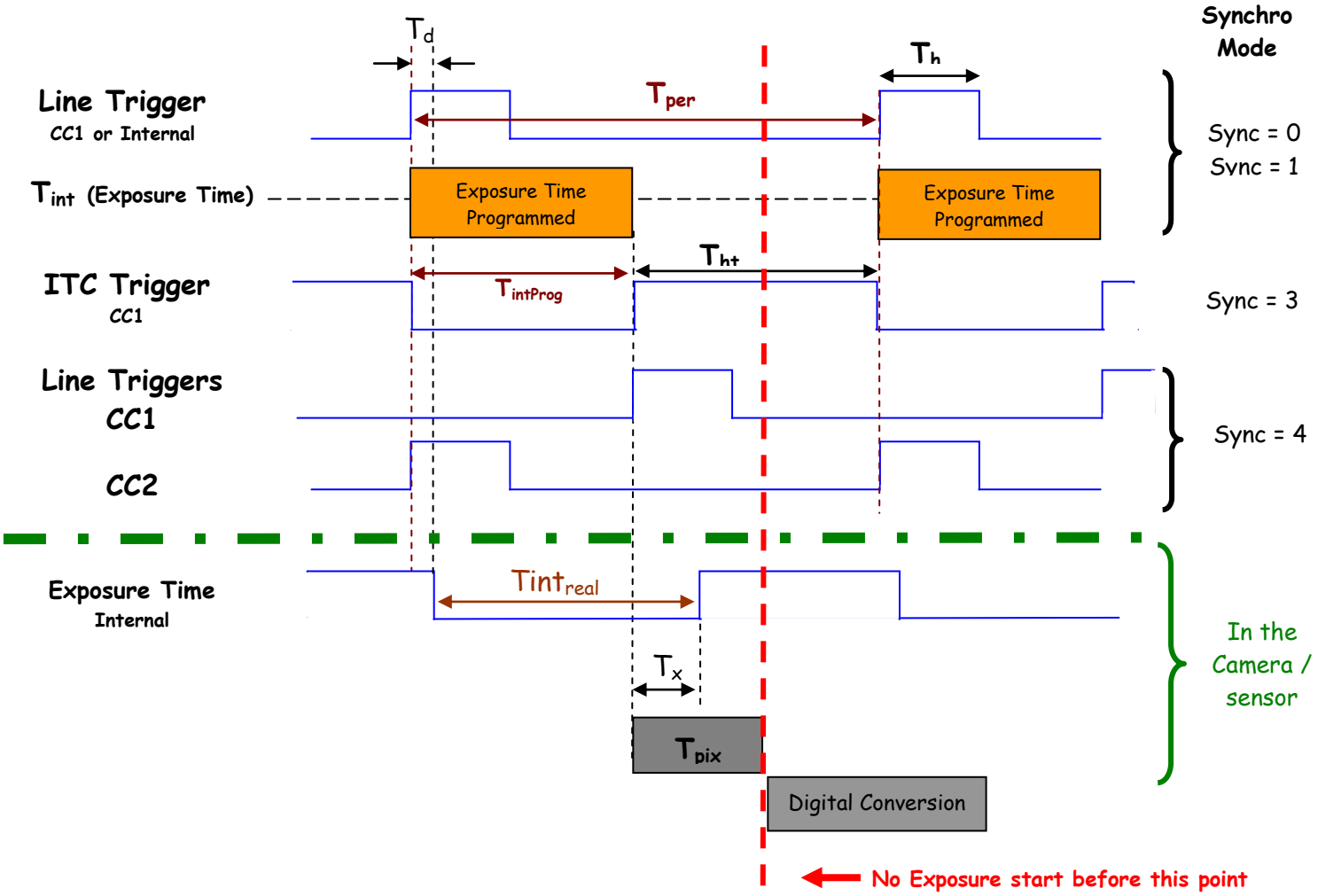



Starting at 0, an increment of 1 LSB is made every 16 pixels. When it reaches 255, turns back to 0 and starts again.

# 8 APPENDIX B: Timing Diagrams

## 8.1 *Synchronization Modes with Variable Exposure Time*




**$T_{pix}$** : Timing Pixel. During this uncompressible period, the pixel and its black reference are read out to the Digital converter. During the first half of this timing pixel (read out of the black reference), we can consider that the exposure is still active.

**Digital Conversion** : During the conversion, the analog Gain is applied by the gradient of the counting ramp (see next chapter : Gain & Offset). The conversion time depends on the pixel format :

- 8 or 10 bits : **6µs**

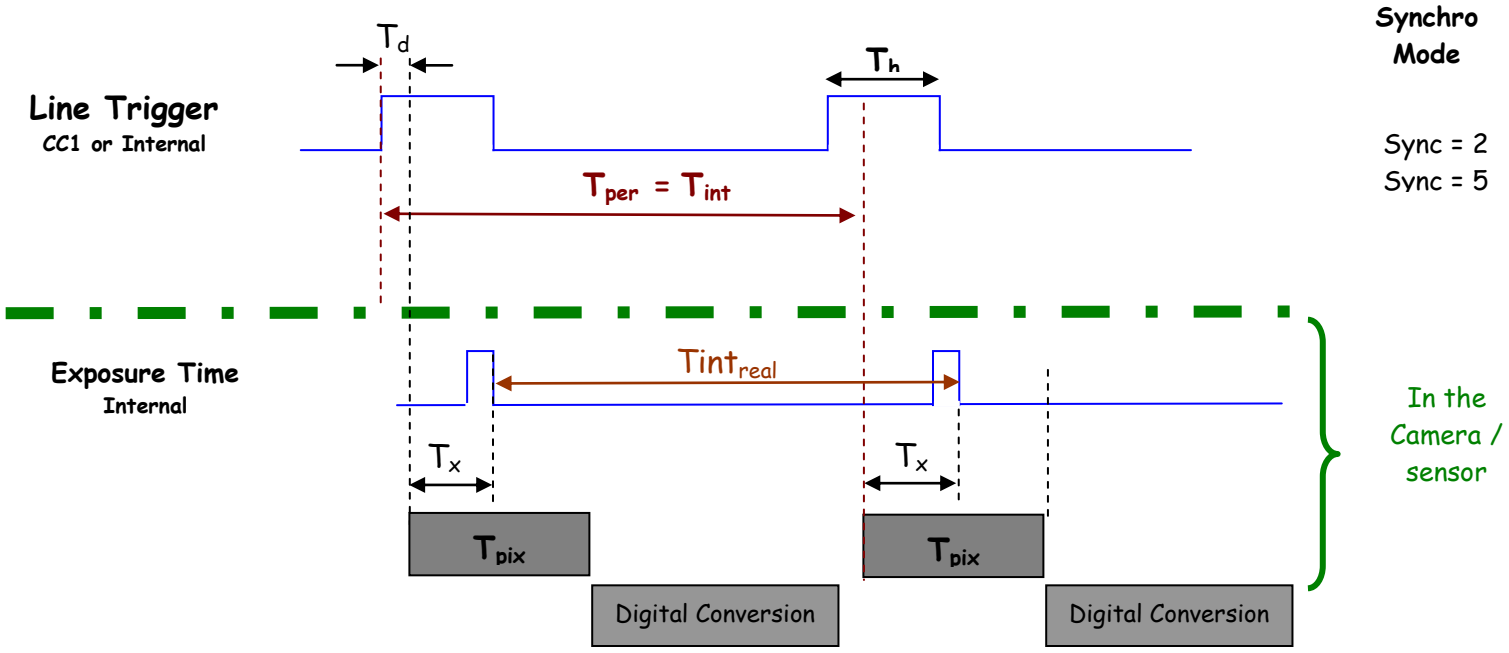
This conversion is done in masked time, eventually during the next exposure period.

**$T_d$** : Delay between the Start exposure required and the real start of the exposure.

*If $T_{per}$ is the Line Period (internal or external coming from the Trigger line), in order to respect this line Period, the Exposure Time as to be set by respecting : $T_{int} + T_{pix} <= T_{per}$*
*Then, the real exposure time is : $Tint_{real} = T_{int} + T_x - T_d$.*
*In the same way, The high level period of the Trig signal in sync=3 mode, $T_{ht} >= T_{pix}$*

For a Line Period of *LinePer*, the maximum exposure time possible without reduction of line rate is : $Tint_{max} = T_{per} - T_{pix}$ ($T_{pix}$ is defined above) but the effective Exposure Time will be about $Tint_{real} = T_{int} + T_x - T_d$.

## 8.2 Synchronisation Modes with Maximum Exposure Time

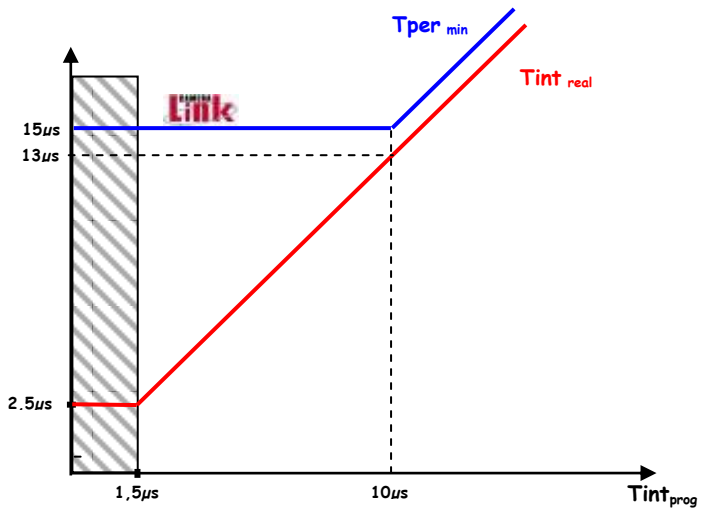



In these modes, the rising edge of the Trigger (internal or External) starts the readout process ($T_{pix}$) of the previous integration. The Real exposure time ($Tint_{real}$) is finally equal to the Line Period ($T_{per}$) even if it's delayed from ($T_x + T_d$) from the rising edge of the incoming Line Trigger.

## 8.3 Timing Values

| Label | Min | Unit |
|---|---|---|

| | | |
|---|---|---|
| $T_{pix}$ | 5 | µs |
| $T_x$ | 3,1 | µs |
| $T_h$ | 0,120 | µs |
| $T_{ht}$ | $T_{pix}$ | µsec |
| $T_d$ | 1.1 | µs |

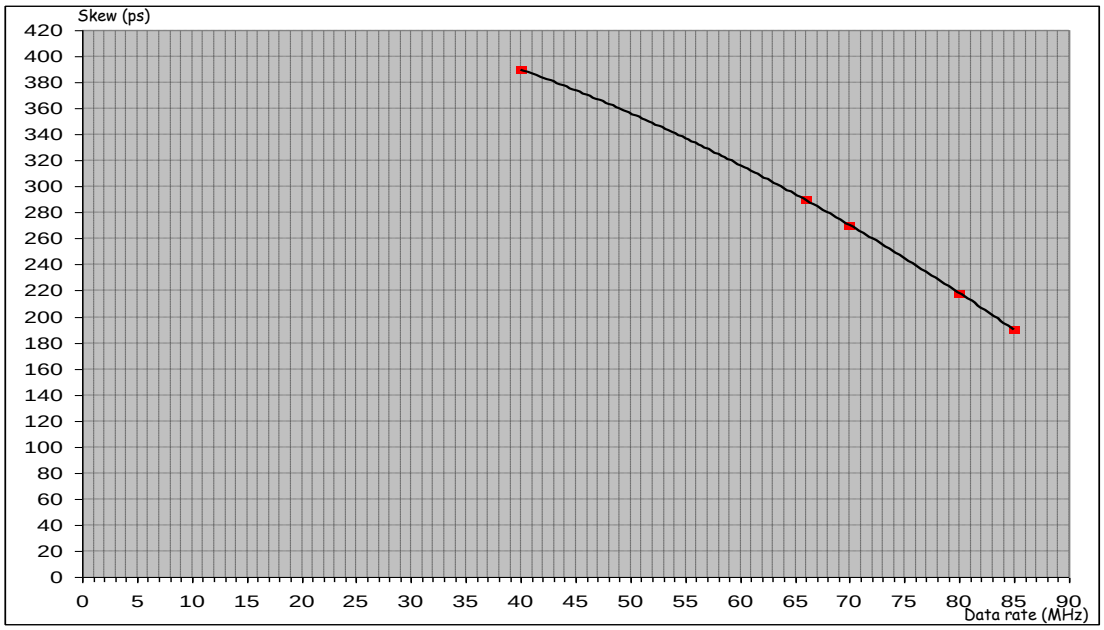


# 9 APPENDIX C: CameraLink Data Cables

## *9.1 Choosing the Cable*

You may check the compliance of your CameraLink cables with the transportation of the 85MHz data rate.
The main parameter to be checked in the cable specification is the skew (in picoseconds)
This parameter is given for a dedicated maximum value per meter of cable (as max : 50ps/m)

The CameraLink Standards defines the maximum total skew possible for each data rate :




Here is a following example of cable and the cable length limitation in accordance with the standard :

| | |
|---|---|
| Conductor Size: | 28 AWG Stranded |
| Propogation Velocity: | 1.25 ns/ft [4.1 ns/m ] |
| Skew (within pair): | 50 ps/meter maximum |
| Skew (channel skew per chipset): | 50 ps/meter maximum |

| DataRate | Skew | Cable Length |
|---|---|---|
| 40Mhz | 390ps | 7,8m |
| 66MHz | 290ps | 5,8m |
| 70MHz | 270ps | 5,4m |
| 80MHz | 218ps | 4,36m |
| 85MHz | 190ps | 3,8m |

## 9.2 Choosing the Data Rate

Maximum Line Rates tables versus Data rate and Pixel Format

| Data Frequency : 85MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 20 | 50 | 40 | 25 | 52.9 | 18.9 | 66.2 | 15.1 |
| Full Def. (8K 5µm) | 10 | 100 | 20 | 50 | 40 | 25 | 50 | 20 |

| Data Frequency : 80MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 18.8 | 53.2 | 37.6 | 26.6 | 49.8 | 20.1 | 62.1 | 16.1 |
| Full Def. (8K 5µm) | 9.4 | 106.3 | 18.8 | 53.2 | 37.6 | 26.6 | 46.9 | 21.3 |

| Data Frequency : 75MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 17.6 | 56.7 | 35.2 | 28.4 | 46.5 | 21.5 | 58.1 | 17.2 |
| Full Def. (8K 5µm) | 8.8 | 113.4 | 17.6 | 56.7 | 26 | 38.4 | 44 | 22.7 |

| Data Frequency : 70MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 16.4 | 60.8 | 32.9 | 30.4 | 43.5 | 23 | 54.3 | 18.4 |
| Full Def. (8K 5µm) | 8.2 | 121.5 | 16.4 | 60.8 | 32.9 | 30.4 | 41.1 | 24.3 |

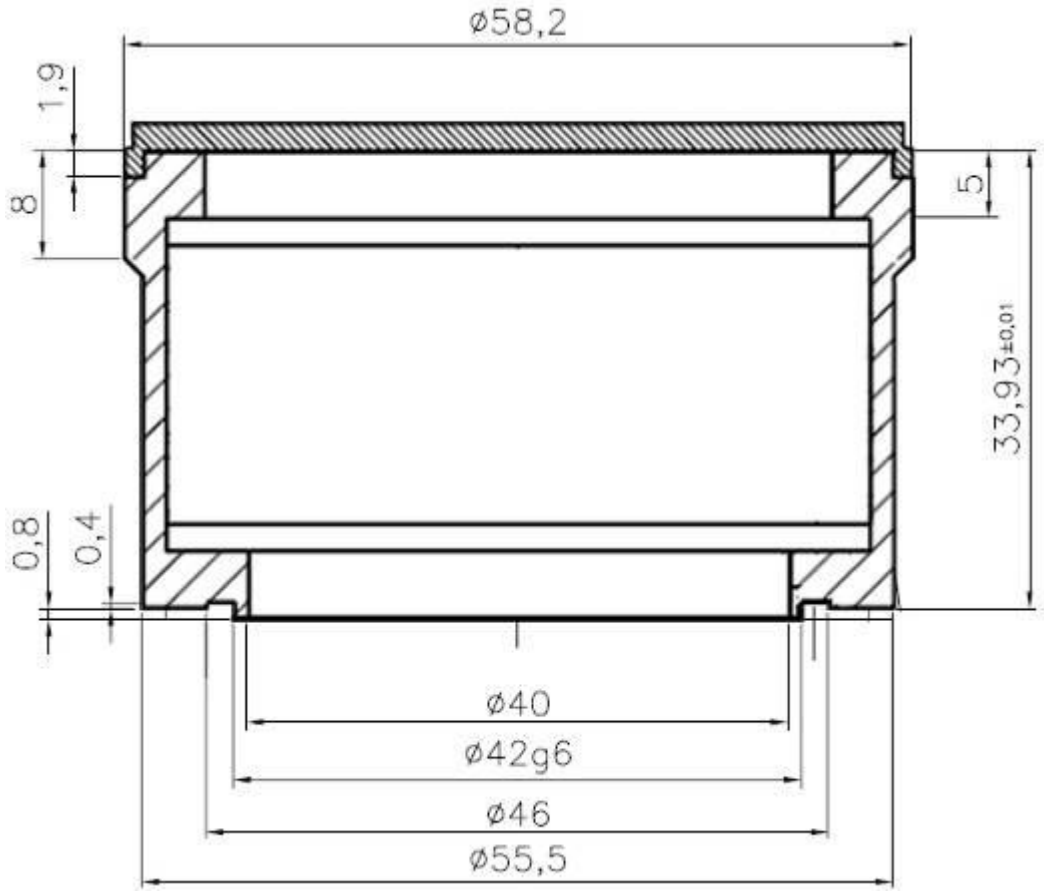
| Data Frequency : 65MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 15.3 | 65.4 | 30.5 | 32.7 | 40.3 | 24.8 | 50.5 | 19.8 |
| Full Def. (8K 5µm) | 7.6 | 130.8 | 15.3 | 65.4 | 30.5 | 32.7 | 38.1 | 26.2 |

| Data Frequency : 60MHz | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sensor Mode | Base : 3x8bits | | Dual Base : 2x 3x8bits | | Full 8x8bits | | Full+ : 10x8bits | |
| | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) | Line Rate Max (kHz) | Tper Min (µs) |
| True Color (4k 10µm) | 14.1 | 70.9 | 28.1 | 35.5 | 37.3 | 26.8 | 46.7 | 21.4 |
| Full Def. (8K 5µm) | 7 | 141.7 | 14.1 | 70.9 | 28.1 | 35.5 | 35.2 | 28.4 |

# 10 APPENDIX D: Lens Mounts

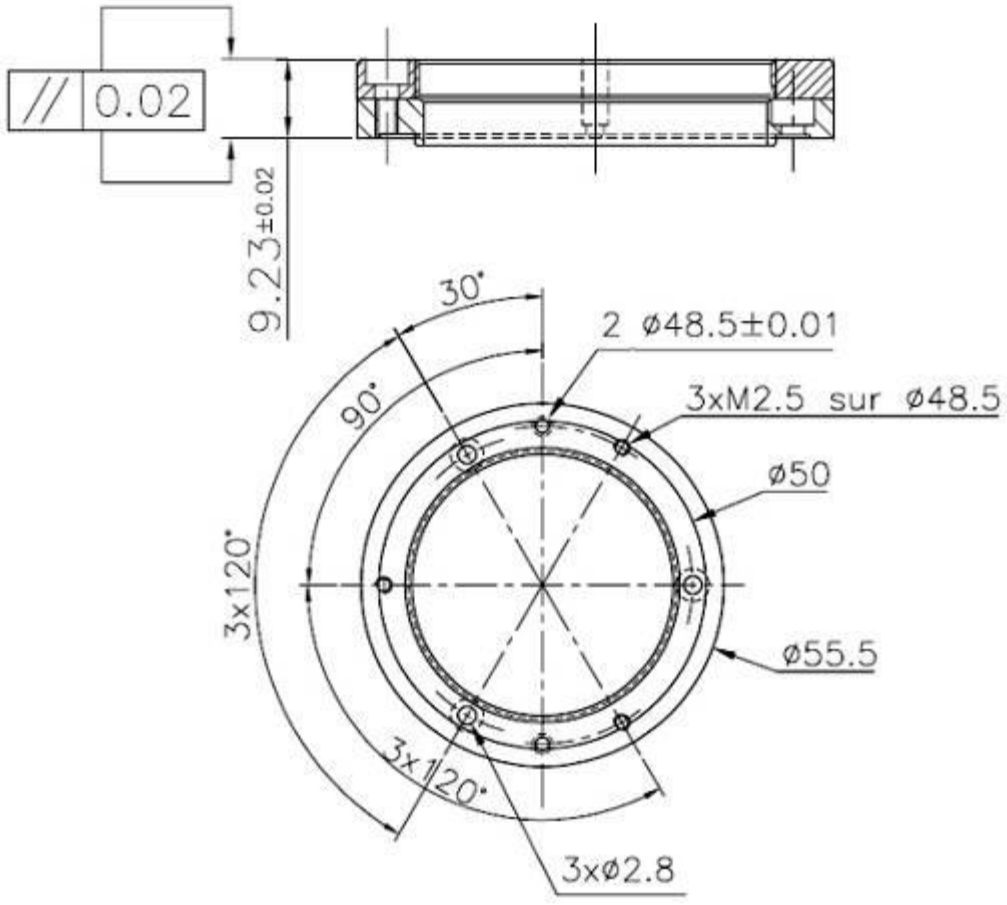
## *10.1 F-Mount*

F Mount : Kit10 (Part number EV71KFPAVIVA-ABA)



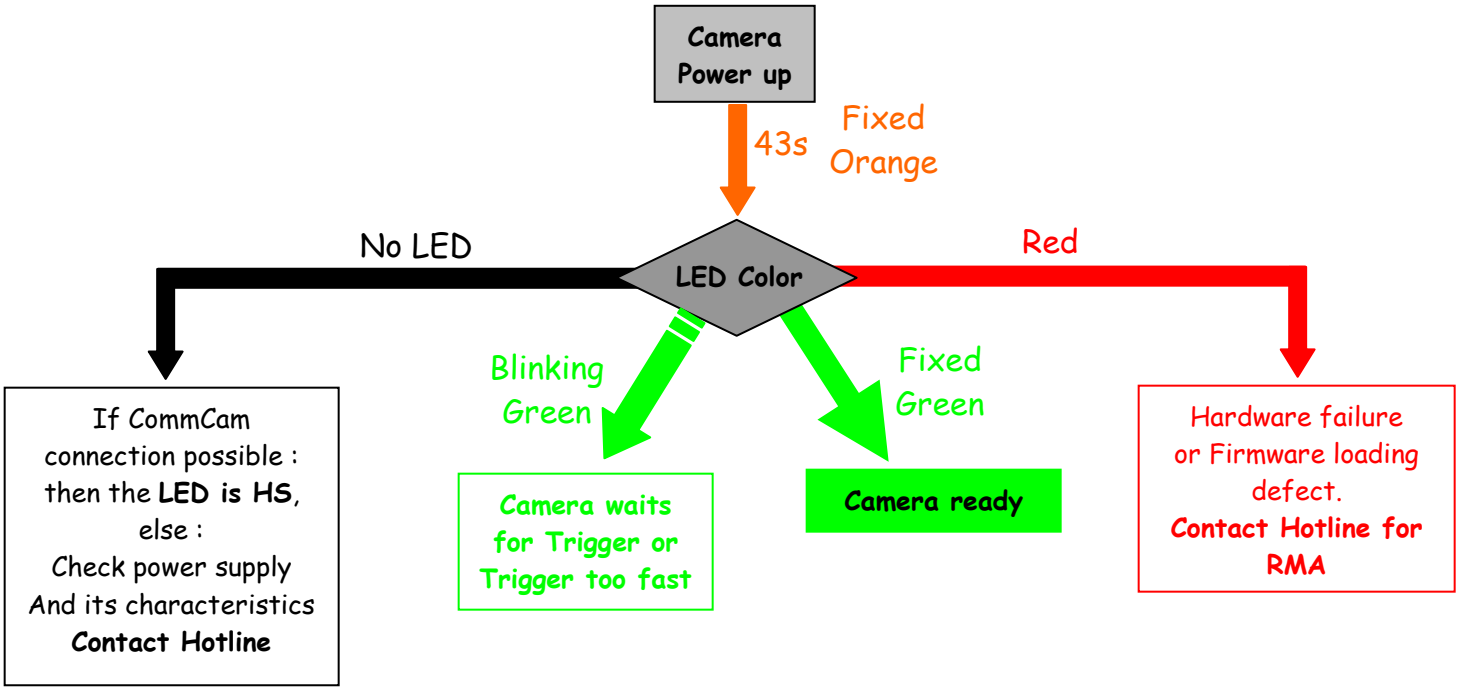
## *10.2 T2 & M42x1 Mounts*

M42x0,75 (T2 Mount) : Kit30 (Part number AT71KFPAVIVA-AKA)
M42x1 Mount : Kit40 (Part number AT71KFPAVIVA-ADA)



# 11 APPENDIX E: TROUBLESHOOTING

## *11.1 Camera*




## *11.2 CommCam Connection*

Refer to CommCam software Help for the connection issues.

# 12 APPENDIX F: Revision History

## *12.1 Device Control*

| Feature | CL Command | Description |
|---|---|---|
| **DeviceVendorName** | r vdnm | Get camera vendor name as a string (32 bytes long including '\0') |
| **DeviceModelName** | r mdnm | Get camera model name as a string (32 bytes long including '\0') |
| **DeviceFirmwareVersion** | r dfwv | Get camera synthetic firmware version (PKG version) as a string (**32** bytes long including '\0') |
| **DeviceVersion** | r dhwv | Get camera version as a string (hardware version) (32 bytes long including '\0') |
| **DeviceManufacturerInfo** | r idnb | Get camera ID as a string (**48** bytes long including '\0') |
| **DeviceUserID** | r cust | Get device user identifier as a string (**16** bytes long including '\0') |
| | w cust <idstr> | Set camera identifier to <idstr> |
| **DeviceID** | r deid | Read Serial Nb |
| **ElectronicBoardID** | r boid | Read Electronic Board ID |
| **DeviceSFNCVersionMajor** | Xml Virtual | 1 |
| **DeviceSFNCVersionMinor** | Xml Virtual | 5 |
| **DeviceSFNCVersionSubMinor** | Xml Virtual | 0 |

## *12.2 Image Format*

| Feature | Command | Description |
|---|---|---|
| **SensorWidth** | r snsw | Get sensor physical width. |
| **SensorHeight** | Xml virtual | |
| **WidthMax** | Map on SensorWidth | |
| **HeightMax** | Xml virtual | |
| **Height** | Xml virtual | |
| **Width** | Xml virtual | Depends on (OuputRegion, OuputRegionWidth) and SensorWidth |
| **SensorMode** | r smod | Get sensor mode |
| | w smod 0 | Set sensor mode to "True Color" |
| | w smod 1 | Set sensor mode to "Full Definition single" |
| | w smod 2 | Set sensor mode to "Full Definition Enhanced" |
| **ScanDirection** | r scdi | Get scan direction |
| | w scdi 0 | Set scan direction to "forward" |
| | w scdi 1 | Set scan direction to "reverse" |
| | w scdi 2 | Set scan direction to "Externally controlled direction via CC3 Camera Link (CC3=0 forward, CC3=1 reverse)" |
| **OutputMode** | r mode | Get output mode (CameraLink configuration and CMOS sensor resolution) |
| | w mode 0 | Set output mode to "BaseRGB8bits" |
| | w mode 1 | Set output mode to "DualBaseRGB8bits" |
| | w mode 2 | Set output mode to "RawFull8Outputs8bits" |
| | w mode 3 | Set output mode to "RawFullPlus10Outputs8bits" |
| **OutputFrequency** | r clfq | Get Camera Link frequency |
| | w clfq 0 | Set Camera Link frequency to 85MHz |
| | w clfq 1 | Set Camera Link frequency to **60MHz** |
| | w clfq 2 | Set Camera Link frequency to **65MHz** |
| | w clfq 3 | Set Camera Link frequency to **70MHz** |
| | w clfq 4 | Set Camera Link frequency to **75MHz** |
| | w clfq 5 | Set Camera Link frequency to **80MHz** |

| Feature | Command | Description |
|---|---|---|
| **TestImageSelector** | r srce | Get test (output FPGA) image pattern |
| | w srce 0 | Set test (output FPGA) image pattern to "Off", processing chaine activated |
| | w srce 1 | Set test (output FPGA) image pattern to "GreyHorizontalRamp", processing chaine desactivated |
| | w srce 2 | Set test (output FPGA) image pattern to "White pattern", processing chaine desactivated |
| | w srce 3 | Set test (output FPGA) image pattern to "gray pattern", processing chaine desactivated |
| | w srce 4 | Set test (output FPGA) image pattern to "Black pattern", processing chaine desactivated |
| | w srce 5 | Set test (output FPGA) image pattern to "GreyVerticalRampMoving", processing chaine desactivated |
| **Color Selection** | r cold | Read the color selection |
| | w cold <val> | Set the color selection. Val :<br>- Bit 0 : Disables the Red color<br>- Bit 1 : Disables the Blue color<br>- Bit 2 : Disables both $Green_{Red}$ and $Green_{Blue}$ colors |

## *12.3 Synchro and Acquisition*

| Feature | Commands | Description |
|---|---|---|
| **LinePeriod** | r tper | Get current line period |
| | w tper <val> | Set line period, from from 150 (15µs) to 65535 (6553,5µs), step 1 (0,1µs) |
| **LinePeriodMin** | r tpmi | Get current line period min (15…65535 step 0,1µs) |
| **AcquisitionLineRate** | Xml Virtual | = 1 / LinePeriod en Hertz |
| **ExposureTime** | r tint | Get exposure time |
| | w tint <val> | Set exposure time, from 1 (0,1µs) to 65535 (6553,5µs), step 1 (0,1µs) |
| **TriggerPreset** | r sync | Get trigger preset mode |
| | w sync 0 | Set trigger preset mode to Freerun timed mode, with exposure time and line period programmable. Not available in FDE sensor mode. |
| | w sync 1 | Set trigger preset mode to Triggered mode with exposure time settings |
| | w sync 2 | Set trigger preset mode to Triggered mode with maximum exposure time |
| | w sync 3 | Set trigger preset mode to Triggered mode with exposure time controlled by one signal |
| | w sync 4 | Set trigger preset mode to Triggered mode with exposure time controlled by two signals. Not available in FDE sensor mode. |
| | w sync 5 | Set trigger preset mode to Freerun mode, with max exposure time and programmable line period |

## 12.4 Gain & Offset

| Feature | Commands | Description |
|---|---|---|
| **Gain**<br>**GainSelector= AnalogAll** | r pamp | Get the current pre-amp gain |
| | w pamp <val> | Set pre amplifier gain to: 0 (-12dB), 1 (-6dB), 2 (0dB) (analog gain)<br>Change balances and compensation |
| **Gain**<br>**GainSelector= gainAll** | r gain | Get current digital gain |
| | w gain <val> | Set gain from 0dB(0) to +8 dB (6193) |
| **Gain**<br>**GainSelector=DigitalAll** | r gdig | Get contrast expansion digital gain |
| | w gdig <val> | Set contrast expansion digital gain from 0 (0 dB) to 255 (+14 dB), step 1 (TBD dB) |
| **BlackLevelRaw**<br>**BlackLevelSelector=All** | r offs | Get common black level. |
| | w offs <val> | Set common black from -4096 to 4095, step 1 |
| **GainAbs**<br>**GainSelector=DigitalTap<j>** | r fga<j> <val> | Get tap<j> digital gain. Dynamically updated on AnalogAll gain changes |
| | w fga<j> <val> | Set tap<j> digital gain from -128 to 127 by step 1 (0.0021dB). Dynamically updated on AnalogAll gain changes |
| **TapBalanceGainEnable** | r fgae | Get the status of the Tap balance |
| | w fgae <val> | Enables the Tap Balance :<br>- 0 : Disables the Tap Balance Gains<br>- 1 : Enables the Tap Balance Gains |
| **Gain**<br>**GainSelector= DigitalRed** | r gwbr | Get the Red Gain for the white balance |
| | w gwbr <val> | Set the Red Gain from 0 (0dB) to 1548 (8dB) : (1 + <val>/1024) |
| **Gain**<br>**GainSelector= DigitalBlue** | r gwbb | Get the Blue Gain for the white balance |
| | w gwbb <val> | Set the Blue Gain from 0 (0dB) to 1548 (8dB) : (1 + <val>/1024) |
| **Gain**<br>**GainSelector= DigitalGreenR** | r gwbg | Get the $Green_{Red}$ Gain for the white balance |
| | w gwbg <val> | Set the $Green_{Red}$ Gain from 0 (0dB) to 1548 (8dB) : (1 + <val>/1024) |
| **Gain**<br>**GainSelector= DigitalGreenB** | r gwbj | Get the $Green_{Blue}$ Gain for the white balance |
| | w gwbj <val> | Set the $Green_{Blue}$ Gain from 0 (0dB) to 1548 (8dB) : (1 + <val>/1024) |
| **AutoWhiteBalanceStart** | w awbc 0 | Stops the Auto white Balance calibration process |
| | w awbc 1 | Starts the Auto white Balance calibration process |
| **AutoWhiteBalanceEnable** | w gwbe 0 | Disables the White Balance |
| | w gwbe 1 | Enables the White Balance |

## 12.5 Flat Field Correction

| Feature | Commands | Description |
|---|---|---|
| **FFCEnable** | r ffcp | Get Flat Field Correction processing status |
| | w ffcp 0 | Disable Flat Field Correction ("False") |
| | w ffcp 1 | Enable Flat Field Correction ("True") |
| **FPNReset** | w rsto 0 | Reset FPN coefficients |
| **PRNUReset** | w rstg 0 | Reset PRNU coefficients |
| No direct feature | r ffco <addr> | Read 128 Fpn coefficients starting from address <addr>. Return value is in hexadecimal, without space between values (one unsigned short per coef). Format: S9.1 => -256 to +255.5 by step of 1/2 |
| | w ffco <addr> <val> | Write 128 Fpn coefficients (straight to FPGA) starting from address <addr>. <val> is the concatenation of individual Fpnvalue, without space between values.<br>- Coefficient from address 0 to 4095 are for red pixels<br>- Coefficient from address 4096 to 8191 are for blue pixels<br>- Coefficient from address 8192 to 12287 are for $green_{Red}$ pixels<br>- Coefficient from address 12288 to 163837 are for $green_{Blue}$ pixels |
| No direct feature | r ffcg <addr> | Read 128 Prnu coefficients (straight from FPGA) starting from address <addr>. Return value is in hexadecimal, without space between values. (one unsigned short per coef)<br>U12 (1+coeff/1024) => x1 to x3.999 by step of 1/1024 |
| | w ffcg <addr> <val> | Write 128 Prnu coefficients (straight to FPGA) starting from address <addr>. <val> is the concatenation of individual PRNUvalue, without space between values.<br>- Coefficient from address 0 to 4095 are for red pixels<br>- Coefficient from address 4096 to 8191 are for blue pixels<br>- Coefficient from address 8192 to 12287 are for $green_{Red}$ pixels<br>- Coefficient from address 12288 to 163837 are for $green_{Blue}$ pixels |
| **FFCCalibrationCtrl** | r calg | Get the PRNU calibration status |
| | w calg 0 | Abort PRNU calibration by setting it to "Off" (no effect if already stopped) |
| | w calg 1 | Launch PRNU calibration by setting it to "Once" (no effect if already launched) |
| **PrnuCalibrationCtrl** | r calo | Get the fpn calibration status |
| | w calo 0 | Abort fpn calibration by setting it to "Off" (no effect if already stopped) |
| | w calo 1 | Launch fpn calibration by setting it to "Once" (no effect if already launched) |

## 12.6 Save and Restore

| Feature | Commands | Description |
|---|---|---|
| **UserSetLoad** | r rcfg | Get the current user configuration bank (saved or restored) |
| | w rcfg <val> | Restore current UserSet from UserSet bank number <val>, from 0 to 5; <val> comes from UserSetSelector. |
| **UserSetSave** | w scfg <val> | Save current UserSet to UserSet bank number <val>, from 1 to 5; <val> comes from UserSetSelector. 0 cannot be saved. 5 (Integrator) can't be saved in User mode |
| **UserSetControl** | Xml virtual | |
| **RestoreLUTFromBank** | r rlut | Get the current LUT bank (saved or restore) |
| | w rlut <val> | Restore current LUT from LUT bank number <val>, from 1 to 4; <val> comes from LUTSetSelector. |
| **SaveLUTToBank** | w slut <val> | Save current LUT to LUT FFC bank number <val>, from 1 to 4; <val> comes from LUTSetSelector. |
| **RestoreFFCFromBank** | r rffc | Get the current FFC bank (save or restore) |
| | w rffc <val> | Restore current FFC (including FPN and FFCGain) from FFC bank number <val>, from 1 to 4; <val> comes from UserFFCSelector (XML feature). |
| **SaveFFCToBank** | w sffc <val> | Save current FFC (including FPN and FFCGain) to FFC bank number <val>, from 1 to 4; <val> comes from FFCSelector (XML feature). |

## 12.7 Camera Status

| Feature | Commands | Description |
|---|---|---|
| **PrivilegeLevel** | r lock | Get camera running privilege level<br>0 = Privilege Factory<br>1 = Privilege Advanced User<br>2 = Privilege User |
| **ChangePrivilegeLevel** | w lock 1 | Lock camera privilege to "Advanced User" |
| | w lock 2 | Lock camera privilege to "User" |
| | w lock <val> | Unlock camera privilege depending on <val> (min=256; max=$2^{32}$-1) |
| **DeviceTemperature** | r temp | Read Mainboard internal temperature (format signed Q10.2 = signed 8 bits, plus 2 bits below comma. Value from -512 to +511) in °C |
| **DeviceTemperatureSelector** | Xml Virtual | |
| **Standby** | r stby | Read Standby state (CMOS sensor) |
| | w stby 0 | Disable standby mode ("False") |
| | w stby 1 | Enable standby mode ("True"), no more video available but save power and temperature |
| | r stat | Get camera status (see below for details) |
| **StatusWaitForTrigger** | | Bit 0: true if camera waits for a trigger during more than 1s |
| **Satus trigger too fast** | | Bit 1: true if camera trigger is too fast |
| **StatusWarningOverflow** | | Bit 8: true if a an overflow occurs during FFC calibration or Tap balance (available only for integrator/user mode) |
| **StatusWarningUnderflow** | | Bit 9: true if a an underflow occurs during FFC calibration or Tap balance (available only for integrator/user mode) |
| **Cc3 Scrolling direction** | | Bit 11: 0 : forward, 1: reverse |
| **StatusErrorHardware** | | Bit 16 : true if hardware error detected |

## 12.8 Communication

| Feature | Commands | Description |
|---|---|---|
| **ComBaudRate** | r baud | Get current baud rate (This feature is not saved in camera) |
| | w baud 1 | Set baud rate to "9600Bds" |
| | w baud 2 | Set baud rate to "19200Bds" |
| | w baud 6 | Set baud rate to "57600Bds" |
| | w baud 12 | Set baud rate to "115200Bds" |
| | | |

## 12.9 Line Profile Average

| Feature | Commands | Description |
|---|---|---|
| **LineAverageProfile** | r pixs | Get the line Line Average Profile status<br>- 1 : running<br>- 0 : finished |
| | w pixs 0 | Abort the Line Average Profile |
| | w pixs 1 | Run the Line Average Profile |
| **PixelAccessLineNumer** | r pixl | Get the number of line for average |
| | w pixl <val> | Set the number of line to accumulate<br>- <val> : 1,256,512,1024 |
| **No direct feature** | r pixv <addr> | Read 128 pixel values starting from address <addr>, from SensorWidth-128-1. Return value is in hexadecimal, without space between values. (one unsigned short per coef) |
| **PixelRoiStart** | r prod | Get Roi start |
| | w prod <val> | Set Roi start for pixel statistic computing (0 to SensorWidth -1-1) |
| **PixelRoiWidth** | r prow | Get Roi width |
| | w prow <val> | Set Roi width for pixel statistic computing (1 to SensorWidth) |
| **RedPixelROIMean** | r pavr | Get ROI Mean value for Red Pixels (format U12.4) |
| **RedPixelROIStandardDeviation** | r pstr | Get ROI Stand deviation for Red Pixels (format U12.4) |
| **RedPixelROIMin** | r pmir | Get ROI Min value for Red Pixels (format U12.4) |
| **RedPixelROIMax** | r pmar | Get ROI Max value for Red Pixels (format U12.4) |
| **BluePixelROIMean** | r pavb | Get ROI Mean value for Blue Pixels (format U12.4) |
| **BluePixelROIStandardDeviation** | r pstb | Get ROI Stand deviation for Blue Pixels (format U12.4) |
| **BluePixelROIMin** | r pmib | Get ROI Min value for Blue Pixels (format U12.4) |
| **BluePixelROIMax** | r pmab | Get ROI Max value for Blue Pixels (format U12.4) |
| **Green(r)PixelROIMean** | r pavg | Get ROI Mean value for $Green_{Red}$ Pixels (format U12.4) |
| **Green(r)PixelROIStandardDeviation** | r pstg | Get ROI Stand deviation for $Green_{Red}$ Pixels (format U12.4) |
| **Green(r)PixelROIMin** | r pmig | Get ROI Min value for $Green_{Red}$ Pixels (format U12.4) |
| **Green(r)PixelROIMax** | r pmag | Get ROI Max value for $Green_{Red}$ Pixels (format U12.4) |
| **Green(b)PixelROIMean** | r pavg | Get ROI Mean value for $Green_{Blue}$ Pixels (format U12.4) |
| **Green(b)PixelROIStandardDeviation** | r pstg | Get ROI Stand deviation for $Green_{Blue}$ Pixels (format U12.4) |
| **Green(b)PixelROIMin** | r pmig | Get ROI Min value for $Green_{Blue}$ Pixels (format U12.4) |
| **Green(b)PixelROIMax** | r pmag | Get ROI Max value for $Green_{Blue}$ Pixels (format U12.4) |

# 13 APPENDIX G: Revision History

| Manual Revision | Comments / Details | Firmware version | 1st CommCam compliant Version |
|---|---|---|---|
| Rev A | First release | 1.0.3 | **2.2.1** |