# BCi5 Manual

## C-Cam Technologies

a division of

## Vector International

## Copyright

C-Cam Technologies is a division of Vector International.

This document contains proprietary and confidential information of C-Cam Technologies, division of vector International.

No part of this document may be translated or reproduced in any form without prior written permission from Vector International.
All rights reserved.

## Disclaimer

The information contained within this document has been carefully checked and is believed to be entirely reliable and consistent with the product that it describes. However, no responsibility is assumed for inaccuracies. C-Cam Technologies division of Vector International assumes no liability arising from of the application or use of any product or circuit described herein. C-Cam Technologies reserves the right to make changes to any product and product documentation in an effort to improve performance, reliability or design.

## Trademarks

IBM, PC/AT, VGA and SVGA are registered trademarks of International Business Machine Corporation. MS-DOS is a registered trademark of Microsoft Corporation.

## Restriction

This code is restricted in reproduction, use and transfer. See the Vector International conditions of use. The license is granted for use of the software on a single computer. By using the software, the user implies agreement to the conditions of use and agrees to settle all disputes through the court in Leuven Belgium.

## Distribution

Distribution is only allowed through registered representatives. A list of these representatives can be found on our web site.

## Contact address

**C-Cam Technologies**
division of
**Vector International**

Technologielaan 4,
B-3001 Leuven
Belgium

Tel. +32 (0)16 40 20 16
Fax +32 (0)16 40 03 23

email info@vector-international.be
http://www.vector-international.be

**IMPORTANT NOTICE**

The screw-locking connectors used on our cameras have been chosen for their industrial qualities and are not intended to be “hot-pluggable”.
The data interface cable should never be plugged or unplugged at the camera end while under power. Failure to observe this restriction can result in damage to the camera’s interface.

## Contents

# 1 Introduction

The BCi5 camera is a linear response camera using a ibis 5 sensor with 1280 x 1024 pixels.
It has a synchronous shutter and features a 12 bit digital output. With the multiple slope mode, the camera can reach a higher dynamic range almost like a logarithmic response sensor.

This document describes the usage of this camera and a software example. It also supposes that the reader is familiar with the Programmers Reference Manual ( or CCAPI manual ). The library functions used in this manual are only references, for a descriptive explanation of these functions refer to this Programmers Reference Manual. Only functions and parameters for functions that are specific for this camera are described.

If you have questions regarding this document, please e-mail to c-cam@vector-international.be. We will be glad to help you.

The engineering team of C-Cam hopes you enjoy their effort in enhancing the industrial digital camera revolution.

C-Cam Technologies

# 2 Camera operation

## 2.1 Windowing

Flexible window of interest (WOI) can be set to read out a specific part of the image sensor pixel array.
A maximum width of 1280 pixels and a maximum height of 1024 pixels can be set. The width and height of the WOI can be changed to any value for the height ranging from 1 to 1024 and to even values for the width ranging from 2 to 1280. The start position of the WOI in the x direction varies from 0 to 1278 (only even values). The start position in the y direction varies from 0 to 1023 and all values are valid.

### **2.1.1** Sub-sampling

Sub-sampling can be activated for the X and Y direction separately by setting the `CC_PAR_XINC` and `CC_PAR_YINC` parameter respectively to 2. Setting these parameters back to 1 disables sub-sampling. With sub-sampling activated 2 pixels are read and 2 pixels are skipped in both directions. The number of pixels is halved in each direction when the corresponding sub-sampling is activated. I.e. the maximum frame size will be 640 by 512. Note that the WOI parameters still require setting the size as if it was not sub-sampled.
The frame rate will also increase up to a factor of 4 when sub-sampling in both directions.
Be careful when setting the Y-start position of a WOI that is sub-sampled. Valid values are 0, 1, 4, 5, 8, 9, …, 1020, 1021. The pixel numbers missing are the one that are skipped.
For the X-start position valid values are 0, 4, 8, …, 1276.

### **2.1.2** Multiple WOI readout in a single pass

It is possible to read multiple WOI’s out of the pixel array which are all exposed in the same time period. I.e. the camera only integrates once, and multiple sections can be read out. This can improve speed as only the parts that are of interest are read out.

Some rules need to be followed :

- The X-start and width position must be the same for all WOIs
- The sections in the Y-direction may not overlap (it is allowed but the overlapped section will be black)

The size of the frame that will leave the camera is equal to the width multiplied by the sum of all heights of the programmed WOIs. The number of WOIs can be programmed by the `CC_PAR_NLIST_OPS` parameter. When one WOI is required, it is not necessary to set this parameter. The `CC_PAR_NLIST_OPS` parameter must be set before calling the `CC_SetWOI()` function.
See also frame rate calculations in section 2.3.

## 2.2 Integration time

The BCi5 camera has a synchronous shutter. This means that all pixels are exposed to light at the same instant. Readout of the exposed pixels can only start after the integration time. Therefore the integration time has a direct influence on the frame rate. The maximum frame rate can only be achieved when the integration time is zero.
Four sets of integration time parameters can be used in multi-slope operation (see multi-slope section). The default mode is single-slope operation when only one set of parameters is used.

## 2.3 Frame rate

The frame period is equal to *integration time* + *readout time*.

Frame period = integration time + (height * (RBT + pixel_period * width))

with : integration time in microseconds
height of the WOI in pixels
RBT the row blanking time being 3.5 microsecs
pixel_period being 25 ns
width of the WOI in pixels

The maximum frame period with zero integration time is

1024 * (3.5 + 0.025 * 1280) = 36.352 ms which results in a frame rate of 27.5 frames per second.

Note that this is the frame rate at which the camera can store images in internal memory, as long the memory is not full. The frame rate that the camera can output depends on the interface, see paragraph 2.6

## 2.4 Offset, Offset Fine

The offset can be set in 128 steps and adds an analog voltage to the output signal of the sensor. Increase this value if too many pixels in the image stay "black" (a value of zero). Decrease this value if a dark frame cannot reach the black level. Due to the dual column amplifiers in the IBIS 5 sensor, differences in intensity can occur between the odd and even columns. This difference can vary from sensor to sensor in the manufacturing process and can be compensated for by setting the `CC_PAR_OFFSET_B_FINE` parameter, a fixed and unique value for each camera.
This parameter will be set in the camera's non-volatile memory during production and has therefore not to be determined by the customer.
When a color sensor is used, the CFA pattern makes the sensor less sensitive and the offset range may not produce a satisfactory result. This can be solved using the `CC_PAR_ANAVAL0` parameter. This parameter adds an additional offset to the sensor signal before it is fed to the sampling electronics.

## 2.5 Gain

The gain value, `CC_PAR_GAIN`, can be set in 15 steps according to the following table:

| Gain setting | DC gain |
|---|---|
| 0 | 1.37 |
| 1 | 1.62 |
| 2 | 1.96 |
| 3 | 2.33 |
| 4 | 2.76 |
| 5 | 3.50 |
| 6 | 4.25 |
| 7 | 5.20 |
| 8 | 6.25 |
| 9 | 7.89 |
| 10 | 9.21 |
| 11 | 11.00 |
| 12 | 11.37 |
| 13 | 11.84 |
| 14 | 12.32 |
| 15 | 12.42 |

Unity gain can be set by writing 255 to the `CC_PAR_GAIN` parameter.

The unity gain setting maps the complete output range of the sensor onto the ADC including the non-linear part, near saturation of the sensor. For all other gain settings, the output range of the sensor will not fit in the ADC range. The `CC_PAR_ANAVAL0` parameter can be used to shift the sensor output signal over the ADC range for example to see the darker or brighter regions.

## 2.6 Data Mode

The BCi5 camera is a 12-bit camera and transmits its image data with a pixel depth of 12-bit by default. The transfer time of an image (not included the integration time) depends on the number of bytes in the image and the interface type. The number of bytes in an image is determined by the size of the WOI and the pixel depth. Reducing the pixel depth can decrease the transfer time of an image, giving a higher frame-rate.

### 2.6.1 LS interface

A BCi5 camera equipped with an LS interface can transfer a full frame with pixel depth of 12 bits in 65.5 msec. If the pixel depth is reduced to 10 bits, then the transfer time will be halved to about 33 msec.
You can switch between the 10 and 12 bit mode by setting or clearing control bit 0 (`CC_PAR_CTRLBIT`).
The logic file for the PCI interface card must be changed when switching between those 2 modes.

| BCi5 Camera Mode | PCI interface card logic | Camera logic |
|---|---|---|
| 12 bit | Pcilshu.ttb | Bci5ls40a.ttb |
| 10 bit | Pcilsrx10.ttb | Bci5ls40a.ttb |

Further bit manipulation is done in the PCI interface card and can be adjusted by using the `CC_PAR_DATA_MODE` parameter. If the data mode is reduced to 8 bits in the interface card, then the transfer of the image from interface card to the user memory buffer is also halved compared to a higher bit-depth setting.

### 2.6.2 USB interface

Since a BCi5 USB camera is connected directly to the PC without an interface card in between, all bit manipulation with the `CC_PAR_DATA_MODE` parameter is done inside the camera. This means that if the camera is set to 8 bit mode, the transfer of an image will be half the time of a transfer with a higher bit depth setting.

### 2.6.3 CameraLink interface

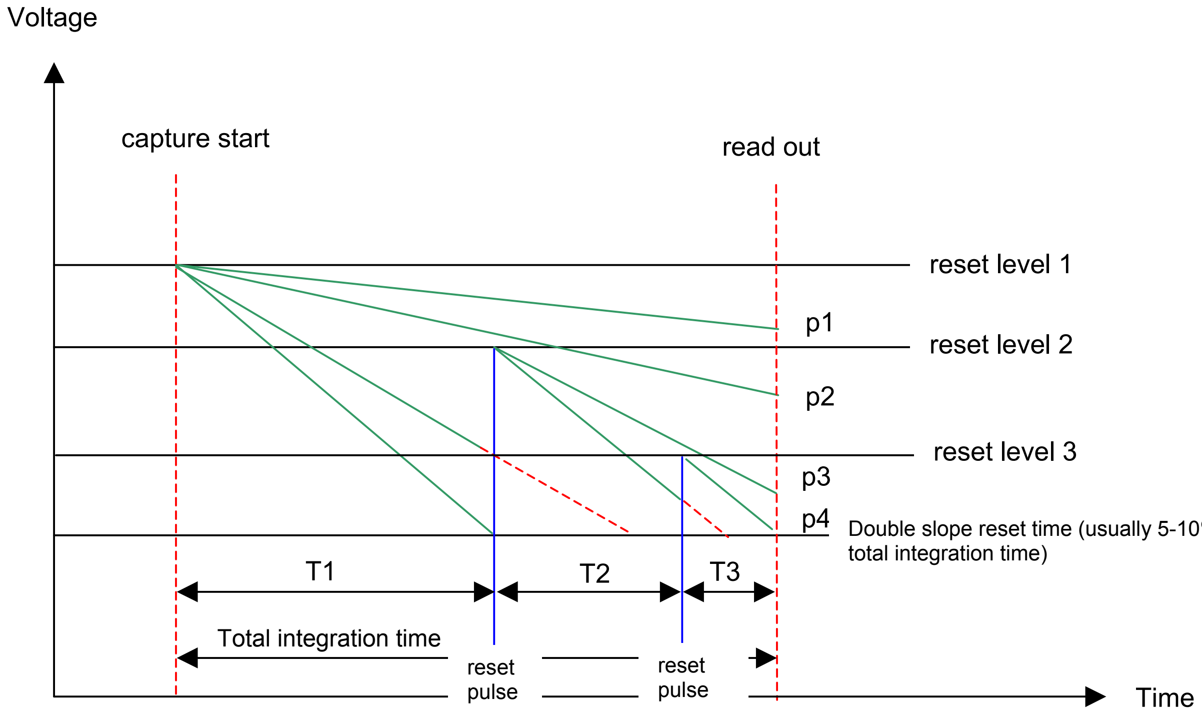
The camera uses a single tap “Base” configuration. The data mode of the image can be switched between 8 bit and 12 bit operation by using the following commands :

| Bitmode | Command |
|---|---|
| 8 bit (default) | E700 |
| 12 bit | E708 |

Note that the camera starts up in 8 bit mode, but this can be changed by a command file programmed into the camera which is executed at startup.
See also the Application Note document on setting CameraLink bitmodes because older versions works a bit different.

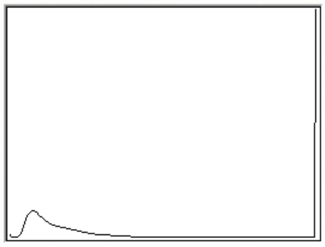
# 3 Multislope operation

The BCi5 camera is able to capture images in multi-slope operation. With this method, the camera increases its dynamic range from 67 dB to about 100 dB. Using multi-slope operation requires some tuning of several parameters and varies from scene to scene. The way this mode works is explained below:



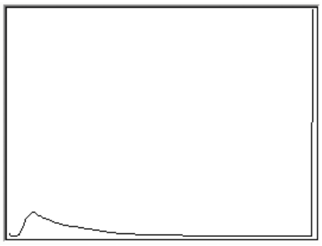

The figure above shows the integration progress of four pixels that are all illuminated with a different intensity of light and with a 3-slope integration time.
The green lines are the analog signal on the photodiode (pixel), which decrease as a result of exposure. The slope is determined by the amount of light at each pixel (the more light, the steeper the slope). When the pixels reach the saturation level the analog signal will not change despite further exposure. As you can see, without multi-slope operation, pixels p3 and p4 will reach saturation before the end of the integration time. No signal will be acquired without multi-slope. When multi-slope is enabled intermediate reset pulses will occur (blue line) at a certain time before the end of the total integration time (T1 and T2). These multi-slope reset pulses reset the analog signal of the pixels BELOW this level to the reset level. After the reset the analog signal starts to decrease with the same slope as before the multi-slope reset pulse. At the end of the total integration time, none of the pixels are saturated and this will result in an increase of the optical dynamic range. It is important to notice that pixel signals above the multi-slope reset levels will not be influenced by these multi-slope reset pulses (p1 and p2 for both reset pulses, and p3 for only the second reset pulse).
The integration time for each slope and the corresponding reset level can all be programmed using the `CC_SetParameter` function. As a rule of thumb, for a quick result, set the integration time of slope Ti+1 to 5-10 % of slope Ti.

The following images show the results of a 3-slope integration. The chosen scene has a high dynamic range because the intensity of the office lights differs greatly from the intensity of light visible through the window. With a single slope camera one has to choose between the window area exposed correctly leading to a black office, or the office exposed correctly leading to a bright spot in the window.

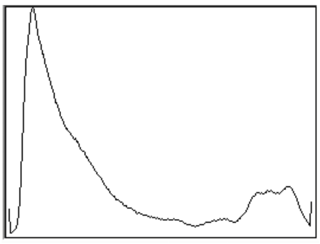


<u>1 slope operation</u>
1[st] slope, 8 msec integration time.

The histogram shows that the window is saturated, visible in the steep line at the right.



<u>2 slope operation</u>
1[st] slope 8 msec, 2[nd] slope 1 msec, 2[nd] slope reset level = 3.0V
Now some pixels in the window are not saturated, but the histogram still shows that not much more information is available than before. An additional slope is needed. Also notice that the office environment has become a bit brighter because the total integration time is now 9 msec.



<u>3 slope operation</u>
1[st] slope 8 msec, 2[nd] slope 1 msec and 3.0V reset level, 3[rd] slope 0.08 msec and 2.6V
Almost no pixels are saturated. The corresponding histogram shows that there is a lot more information than previously. The total integration time is now 9.08 msec.

In the figure explaining the multi-slope operation, reset level 1 is the same level used in the standard single slope operation that is the default mode at startup. This level can be programmed with parameter `CC_PAR_ANAVAL3`.
To switch to multi-slope operation, the following line of code has to be used :

```
CC_SetParameter(MyCam, CC_PAR_CTRLBIT, 257); // sets control bit 1 in the camera
```

A value of 256 resets bit 1 and switches back to single slope mode.
For the programming of the parameters for generating the last image in the 3 slope example the following lines of code has to be used :

```
CC_SetParameter(MyCam, CC_PAR_INTSEL, 0); // select first slope
CC_SetParameter(MyCam, CC_PAR_INTEGRATION_TIME, 100000); // 100 msec
CC_SetParameter(MyCam, CC_PAR_ANAVAL3, 0); // reset voltage = 4 V

CC_SetParameter(MyCam, CC_PAR_INTSEL, 1); // select second slope
CC_SetParameter(MyCam, CC_PAR_INTEGRATION_TIME, 20000); // 20 msec
CC_SetParameter(MyCam, CC_PAR_ANAVAL1, 70); // reset voltage = 3.7 V

CC_SetParameter(MyCam, CC_PAR_INTSEL, 2); // select third slope
CC_SetParameter(MyCam, CC_PAR_INTEGRATION_TIME, 1500); // 1.5 msec
CC_SetParameter(MyCam, CC_PAR_ANAVAL1, 110); // reset voltage = 3.5 V
```

Note that the reset level for the first slope is not set by the `CC_PAR_ANAVAL1` parameter but with the `CC_PAR_ANAVAL3` parameter as this is the same value used in single slope operation.
Also note that the last value written to the `CC_PAR_INTSEL` parameter also determines how many slopes will be used during integration. A value of 2 means 3 slopes. A maximum of 4 slopes can be used.

# 4 Camera Parameters

## 4.1 Using the SDK with LS and USB interface

For a list of all parameters refer to Appendix A : SDK parameters

## 4.2 Camera commands when using CameraLink interface

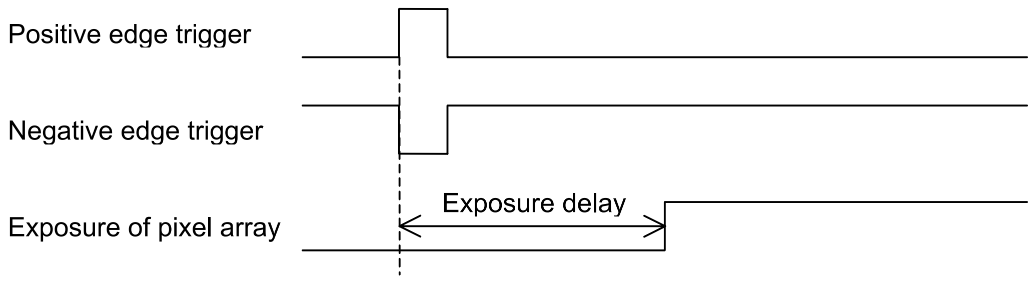
For a list of all commands refer to Appendix B : Camera Link commands

# 5 External triggering

For a complete description on using the external triggering of the camera refer to the “Trigger IO” manual.

## 5.1 Exposure delay

After receipt of a pulse on the Trigger Input there is a delay before exposure (the integration time) starts. This delay depends on the polarity of the trigger pulse.




| Trigger event | Exposure delay (us) |
|---|---|
| Positive edge | 170 |
| Negative edge | 215 |

The delays are measured with trigger pulses of 5V.

# 6 Programming example

## 6.1 Control Using SDK

This programming example uses only basic functions from the programmers interface (API) with no error checking. This example is only meant to show what functions are needed and in which order. For a more complete and working example see the examples in the application directory.

This example opens the camera and initialises it, then it captures an image into a buffer and finally it closes the camera :

```
USHORT      buffer[1280*1024] ;
ULONG       picture_size ;
BOOL        ret ;
HANDLE      MyCam ;

MyCam = CC_Open( "BCi5 LS", 0, CC_CAPTURE_WAIT ) ;
ret = CC_LoadInterface( MyCam, "pcilshu.ttb" ) ;
ret = CC_LoadCamera( MyCam, "bci5ls40a.ttb" ) ;
ret = CC_SetWOI( MyCam, 0, 0, 1279, 1023, 1, 1, CC_WOI_LEFTTOP_RIGHTBOTTOM,
                 &picture_size ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL0, 200 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL1, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL2, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL3, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_GAIN, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_OFFSET, 100 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_DATA_MODE, CC_DATA_16BIT_11_DOWNTO_0 );
ret = CC_SetParameter( MyCam , CC_PAR_CAMERA_MODE , CC_CAMERA_NORMAL ) ;
ret = CC_CaptureSingle( MyCam, buffer, picture_size*2, CC_NO_TRIGGER, 1,
                        NULL ) ;
ret = CC_Close( MyCam ) ;
```

CC_Open must be the first function to call before you can access any other function. (Note that the handle returned by CC_Open is used by all other functions). The next functions to call should be CC_LoadInterface and CC_LoadCamera in that order. All other functions before the CC_CaptureSingle function can be called in any order. And the last function to call should be CC_Close.

For proper error correction, the `ret` value should be checked for TRUE. If the `ret` value is FALSE, then an error occurred and you should call `GetLastError` to find out what went wrong. You can find the matching error-value in CCAPIERR.H

## 6.2 Control by Camera Link interface

The following command sequence starts the camera running at full frame full speed.
The commands must be formatted according to the format specified in Appendix C : Camera Link communication.

| Command | Remarks |
|---|---|
| F600 | Y-start = 0 |
| F750 | |
| F600 | Y-start = 0 |
| F760 | |
| F6FF | Number of lines = 1023 (= 1024 lines) |
| F723 | |
| F600 | X-start = 0 |
| F740 | |
| F67F | Width = 639 (= 1280 pixels) |
| F712 | |
| F600 | Sub sampling off |
| F770 | |
| F640 | Gain = 0 |
| F780 | |
| F64B | Offset = 75 |
| F790 | |
| F680 | Offset fine = 128 |
| F7A0 | |
| FC10 | Frame time = 0 |
| FE00 | |
| FE00 | |
| FE00 | |
| FE00 | |
| E080 | Integration time = 10 ms (value = 0x61A80) |
| E11A | |
| E206 | |
| E300 | |
| E8C8 | Anaval0 = 200 |
| EA78 | Anaval2 = 120 |
| FF82 | Start the camera frame time controlled |

# 7 Non-volatile memory with Camera Link interface

Since the BCi5-CL camera can not be used with the SDK directly, but has to be controlled by a Camera Link frame-grabber with a slow command channel, the camera is equipped with non-volatile memory for stand alone operation. Note that this non-volatile memory is only available with BCi5 cameras with a Camera Link interface. This memory can contain a configuration file for the programmable logic which define the operation of the camera, and can also contain a command list to set the correct camera parameters at start-up. An example of a command list is shown in section 6.2.

When the camera is powered up, a camera management CPU will first check if a configuration file is present and will then program the FPGA of the camera. This operation takes about 1 second. After this, the CPU checks if a command file is present and will then execute the command list. When either the configuration file or the command file is missing, it has to be loaded via the Camera Link frame-grabber' serial communication channel. The configuration file is factory programmed but can be changed or erased by the user.

The non-volatile memory can be maintained with the CTLoad program. For more information refer to the CTLoad manual.

# 8 Color filter geometry

All BCi5 versions are available with a monochrome sensor or a sensor with a color filter array applied.

Starting from pixel coordinate (0;0) to the end of the first line (0;1279) and then all subsequent rows.

The layout of the Bayer pattern:

**GBGBGBGB** ...
**RGRGRGRG** ...
**GBGBGBGB** ...
**RGRGRGRG** ...
...

Please note that BCi5 cameras with a color sensor will still output gray-scale or intensity values for each color. Color recombination needs to be done in image processing software.

# 9 Camera variations

The BCi5 camera is available in the following variations :

| Camera | Camera interface | PCI interface board | Camera Logic file | PCI board logic file |
|---|---|---|---|---|
| BCi5 LS | Serial LVDS | PCI-LS | Bci5ls40a.ttb | Pcilshu.ttb<br>Pcilsrx10.ttb |
| BCi5 USB | USB 2.0 | - | Bci5usb.ttb | - |
| BCi5 CL | Camera Link Base | Camera Link frame grabber | Bci5cl40a.ttb | - |

Use the name in the “Camera” column in the `CC_Open` command to open the corresponding camera.

# 10 Appendix A : SDK parameters

List of parameters that can be used with the `CC_SetParameter` function:

| Parameter | Values | Description |
|---|---|---|
| CC_PAR_XSTART | Max 1280 | Use CC_SetWOI function with these parameters.<br>See "Programmers Reference Manual" |
| CC_PAR_XEND | Max 1280 | |
| CC_PAR_XINC | 1, 2 | |
| CC_PAR_YSTART | Max 1024 | |
| CC_PAR_YEND | Max 1024 | |
| CC_PAR_YINC | 1, 2 | |
| CC_PAR_CAMERA_MODE | CC_CAMERA_NORMAL<br>CC_CAMERA_DIAG_X<br>CC_CAMERA_DIAG_Y<br>CC_CAMERA_DIAG_X_XOR_Y | See "Programmers Reference Manual" |
| CC_PAR_DATA_MODE | | See "Programmers Reference Manual" |
| CC_PAR_INTEGRATION_TIME | 1 … 107374180 | Sets the integration time of the ibis5 sensor in microseconds. Although you can program this value to about 107 seconds, max 10 seconds will give you a meaningful image. For higher integration times, extra efforts have to be done like cooling the sensor.<br>In combination with the CC_PAR_INTSEL parameter, multiple integration times can be defined for multislope operation. See multislope section. |
| CC_PAR_INTSEL | 0 .. 3 | Selects the number of slopes to use. Use this parameter also for setting the appropriate integration time and reset level voltage. |
| CC_PAR_NLIST_OPS | 0 .. 63 | Sets the number of WOIs that has to be read out in a single pass. 0 = 1 = one WOI, 2 = 2 WOIs, etc… Only used in global shutter mode. Not needed to be set when using one WOI. |
| CC_PAR_GAIN | 0 .. 15, 255 | Sets the output amplifier of the sensor. 0 = lowest, 15 = highest. 255 = unity gain. See table for exact amplification factor. This parameter must be set once for correct camera operation. |
| CC_PAR_OFFSET | 0 .. 127 | Adds an analog offset to the output signal of the sensor. |

| | | |
|---|---|---|
| CC_PAR_OFFSET_B_FINE | 0 .. 127 | The ibis 5 has 2 amplifiers, one for all even columns and one for all odd columns. This parameter can be used to correct the deviation mismatch between these two amplifiers. |
| CC_PAR_ANAVAL0 | 0 .. 255 | Sets the centre voltage for sensors output signal. This acts as an additional offset parameter. Set to 200. |
| CC_PAR_ANAVAL1 | 0 .. 255 | Sets the external reference voltage for multislope operation.<br>In combination with the CC_PAR_INTSEL parameter, multiple reset voltages can be defined for multislope operation. See multislope section. 0 = 4V; 255 = 1V |
| CC_PAR_ANAVAL2 | 0 .. 255 | Sets the pixel core voltage. 0 = 4V; 255 = 2.5V. Set to 120. |
| CC_PAR_ANAVAL3 | 0 .. 255 | Sets the highest reset voltage. 0 = 4.5V; 255 = 2.5V. The higher the voltage, the more the full well capacity. Set to zero. |
| CC_PAR_CTRLBIT | Bit 0, 1 | Bit 0 = 0 sets the camera in 12-bit mode.<br>Bit 0 = 1 sets the camera in 10-bit mode, transfer of images will be twice as fast as in 12-bit mode. (Only applicable in combination with a PCI-LS card)<br>Bit 1 : enable multi-slope operation |

# 11 Appendix B : Camera Link commands

List of commands for controlling the camera via a Camera Link interface.
The commands are always 16-bit values with an op-code field of 4, 8, 12 or 16 bits and a parameter field of respectively 12, 8, 4 or 0 bits depending on the type of command. The parameter field is indicated by the 'x' symbol in the Command column.

| Parameter | Command (HEX) | Description |
|---|---|---|
| WOI Y left start | F6xx<br>F75x | 0 to 1023 (y start position) |
| WOI Y right start | F6xx<br>F76x | 0 to 1023 (y start position) |
| WOI Y height | F6xx<br>F72x | 0 to 1023 (number of lines - 1) |
| WOI X start | F6xx<br>F74x | 0 to 639 (start position = multiple of 2) |
| WOI X width | F6xx<br>F71x | 0 to 639 (value = width/2 - 1) |
| Sub sampling | F6xx<br>F770 | bit 2 = X subsample<br>bit 5 = Y subsample |
| Integration time | E0xx (LSB)<br>E1xx<br>E2xx<br>E3xx (MSB) | Integration time in units of 25 ns |
| Frame time | FC10<br>FExx (LSB)<br>FExx<br>FExx (MSB) | Frame time in microseconds |
| Offset | F6xx<br>F790 | 0 to 127 |
| Offset fine | F6xx<br>F7A0 | 0 to 127 |
| Gain | F6xx<br>F780 | 0 to 15<br>bit assignments : (nc, standby_n, 0, unity, gain[3..0]) |
| Sequencer | F6xx<br>F70x | bit 0 = shutter mode : 0 = global, 1 = rolling |
| Integration slopes | F2xx | 0 to 3 |
| Snapshot | FF80 | start the camera, single shot |
| Start triggered | FF81 | start the camera, triggered |

| | | |
|---|---|---|
| Start continuous | FF82 | start the camera continuous, frame rate controlled |
| Stop | FFFC | stop the camera |
| Reset camera | FFFD | reset the internal logic and sensor |
| Diagnostics | E7xx | Diagnostic bits, bit 6 = diagnostic on, bit 5 = Y pattern |
| Vmid | E8xx | 0 to 255 |
| Multislope pixel reset voltage | E9xx | 0 to 255 |
| Pixel core voltage | EAxx | 0 to 255 |
| Reset voltage | EBxx | 0 to 255 |
| Flash delay | FC04<br>FExx (LSB)<br>FExx<br>FExx (MSB) | Flash delay in units of 25 ns |
| Flash width | FC08<br>FExx (LSB)<br>FExx<br>FExx (MSB) | Flash width in units of 25 ns |
| Trigger control | D5xx | bit 0 = External input polarity, 0 = low, 1 = high<br>bit 1 = Trigger output polarity, 0 = low, 1 = high<br>bit 2 = Enable external trigger without start command<br>bit 3 = Use trigger out as arm ready<br>bit 4 =<br>bit 5 =<br>bit 6 =<br>bit 7 = |
| Control bits | FFCx | Set / Clear control switches<br>Where x denotes:<br>Bits 3,2,1 = bit number<br>Bit 0 = 0 for Clear, = 1 for Set<br><br>bit 0 = Cameralink Base bit mode<br>(1 = 8-bit, 0 = 10/12-bit)<br>bit 1 = Enable multi-slope operation<br>bit 2 .. 7 not used<br><br>e.g.<br>FFC0 Cameralink mode 10/12-bit<br>FFC1 Cameralink mode 8-bit<br>FFC2 Disable multi-slope<br>FFC3 Enable multi-slope |

# 12 Appendix C : Camera Link communication

## 12.1 Camera Link serial channel

Camera commands are sent via the Camera Links serial channel. All Camera Link compatible frame grabbers provide user-functions that can be used to send commands to the camera. These functions all reside in a separate library or DLL file. The filename of this DLL is CLSERxxx.DLL where xxx is specific to each vendor. These functions are:

- `clSerialInit`
- `clSerialWrite`
- `clSerialRead`
- `clSerialClose`

Please refer to the Camera Link specification for more information.

## 12.2 Command Message format

Command messages use the standard (Intel) format for HEX records.

The message consists of ASCII characters 0 to 9 and A to F except for the first character, which is a colon.

Commands sent to the camera with the `clSerialWrite` function have the following message format:

| | | | | | |
|---|---|---|---|---|---|
| : | Length | Address | Record type | Data | Checksum |

Field explanation :

| | |
|---|---|
| **:** | The first character indicating the start of a new message |
| **Length** | The length of the Data field in bytes (= number of characters / 2). Is always '02' |
| **Address** | This field always contains '0000' |
| **Record type** | Must always be 'BC' |
| **Data** | This field contains 2 bytes (4 characters) that contains the actual camera command (see command list) |
| **Checksum** | 8 bit checksum code. Sum, modulo 256, of all fields (bytes) except the ':' including checksum field must be zero. |

Example :

`:020000BCFF80C3` (Camera start command = FF80 hexadecimal)

Checksum = 256 – $Mod^{256}$(02 + 00 + 00 + 0xBC + 0xFF + 0x80) = 0xC3

## 12.3 Camera Response format

The camera always responds with one of two characters:

| | | |
|---|---|---|
| ACK | 0x06 | Command accepted |
| NACK | 0x15 | Checksum or length error |

**Document History**

| Issue | Date | Changes |
|---|---|---|
| 1.0 | 4/9/2003 | First Issue |
| 1.1 | 19/1/2004 | • Changed document layout<br>• Added “External Triggering” paragraph |
| 1.2 | 20/4/2004 | • Made changes in the camera parameters list<br>• Changed gain settings table<br>• Added “data mode” paragraph<br>• Added “color filter geometry” paragraph<br>• Added “camera variations” paragraph<br>• Added Document History |
| 1.3 | 15/11/2006 | • Changed anaval0 default value to 200<br>• Unity gain added in parameter table<br>• Moved camera parameters to appendix<br>• Added paragraphs in chapter 2<br>• Added appendices<br>• Added Camera Link information through whole document<br>• Inserted chapter 7 |
| 1.4 | 27/11/2007 | • Added paragraph 5.1<br>• Changed paragraph 2.6.3 |
| 1.5 | 24/01/2008 pmb | • Postal address change<br>• USB cable warning |