

BCi4-6600 Manual

6.6 M Pixel camera



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

Research Park
B-3001 Leuven
Belgium

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

email info@c-cam.be
<http://www.c-cam.be>

Contents

1	Introduction	4
2	Camera Specification	5
2.1	Windowing	5
2.2	Sub-sampling	6
2.3	Frame Rate and Integration Time	9
2.3.1	Integration Time	9
2.3.2	Frame Rate	9
2.4	Offset, Offset Fine	9
2.5	Gain	10
2.6	Data Mode	11
2.6.1	LS interface	11
2.6.2	USB interface	11
2.6.3	CameraLink interface	11
3	Camera Parameters	12
3.1	Using the SDK with LS and USB interface	12
3.2	Camera commands when using CameraLink interface	12
4	External triggering	12
5	Programming example	13
5.1	Control Using SDK	13
5.2	Control by Camera Link interface	14
6	Non-volatile memory with Camera Link interface	15
7	Color filter geometry	15
8	Camera variations	15
9	Appendix A : SDK parameters	16
10	Appendix B : Camera Link commands	18
11	Appendix C : Camera Link communication	21
11.1	Camera Link serial channel	21
11.2	Command Message format	21
11.3	Camera Response format	21

1 Introduction

The BCi4-6600 camera is a linear response camera using an ibis 4-6600 CMOS imager which is a 6.6Megapixel "portrait"-format sensor with a matrix of 2208 horizontal and 3000 vertical pixels.

The camera is available in color and monochrome versions. Both versions are the same except that the color version has a Bayer color filter array overlaid on top of the pixels.

The sensor has a rolling shutter and features a 12-bit digital output. The camera can be equipped with different interfaces and at this moment available types are:

1. USB2.0,
2. IEEE1394a (supporting the DCAM V1.31 specification)
3. LS interface which is a proprietary interface that, in combination with a PCI-LS interface, gives complete real-time control over the camera.

Other features of the camera are:

- External trigger input and flash output
- Non-volatile storage of camera parameters
- Built-in 1st order calibration algorithms

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 shown for reference. For a descriptive explanation of these functions refer to the Programmers Reference Manual. Only functions and function parameters specific to this camera are described here.

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 Specification

2.1 Windowing

Flexible Window Of Interest (WOI) can be set in the camera, allowing the selection of almost all possible windows in the focal plane of the sensor. The advantage of windowing is that the smaller the window is, the higher the frame rate will be. The word 'almost' is used because there are some rules that have to be followed when setting up the WOI.

The WOI can be set with the `CC_SetWOI` function. This function allows you to set the upper left corner and the lower right corner of the window (alternatively you can use width and height values in another mode. See the Programmers Reference Guide).

The maximum window size is 2208 (H) x 3000 (V) pixels, which is equal to a full frame. The left and top edges of the WOI must be a multiple of 24 pixels. This gives a total of 92 horizontal start columns and 125 vertical start rows.

The right edge must be a multiple of 2 and the bottom edge has total freedom and can be on any line.

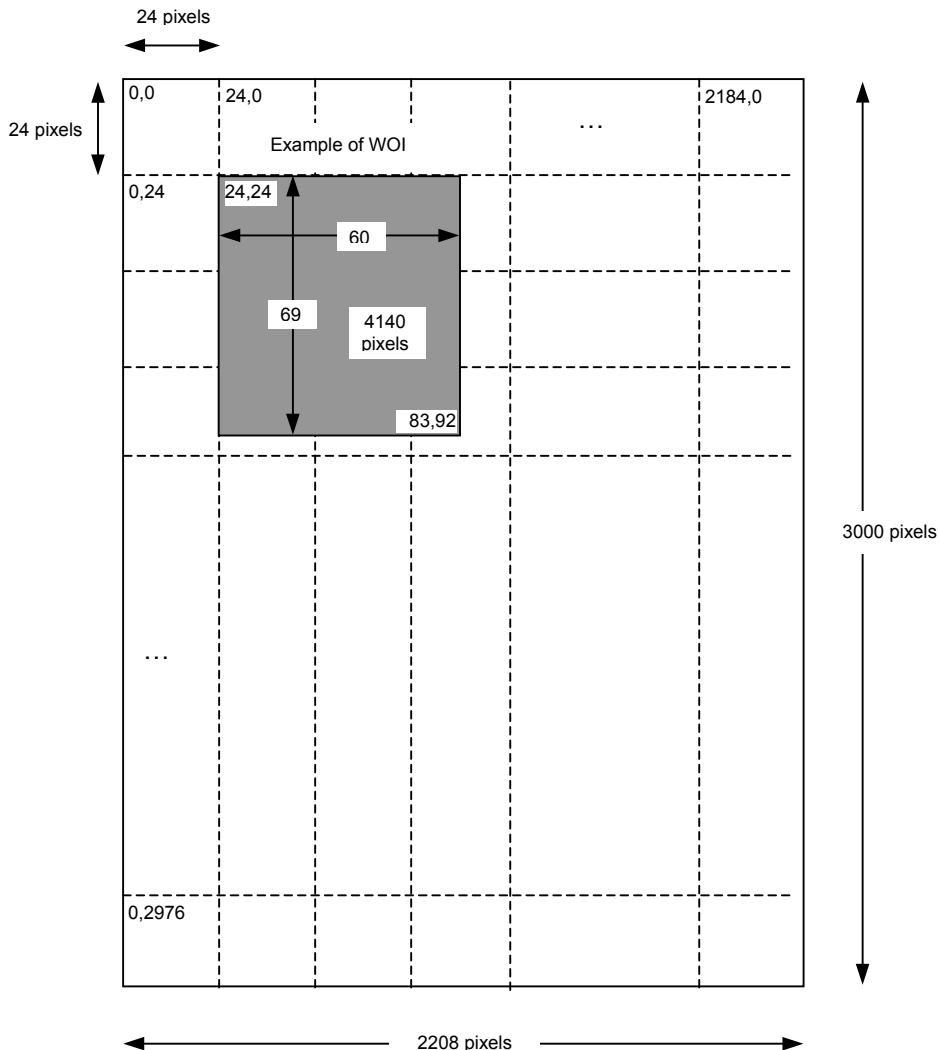


Figure 1. WOI example

2.2 Sub-sampling

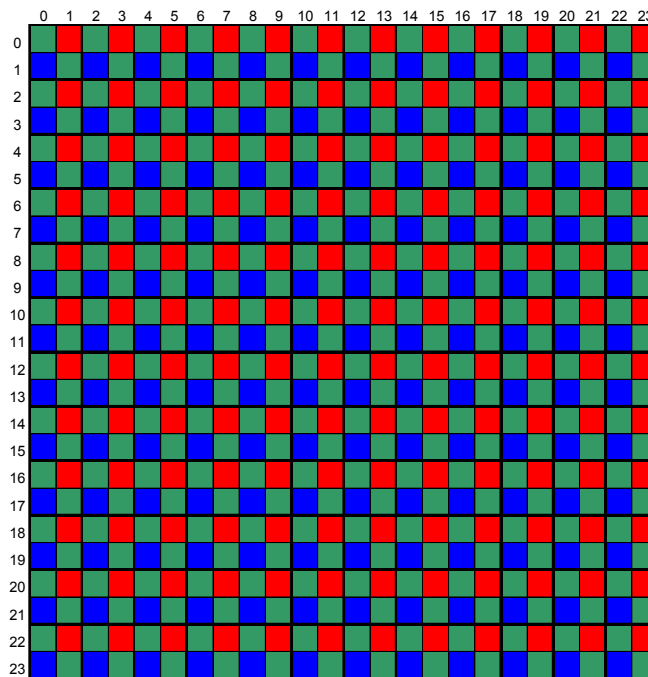
The camera supports several sub-sampling modes. By sub-sampling, the resolution can be reduced while maintaining the full field of view of the imager. The pixels are grouped in blocks of 2 by 2 pixels. This is done to preserve the information in the colour version as it represents one Bayer element: One red, one blue and 2 green pixels.

When sub-sampling, several groups of pixels are skipped during read out in horizontal and/or in vertical direction. The possible sub-sampling modes are 1, 2, 3, 4 and 6.

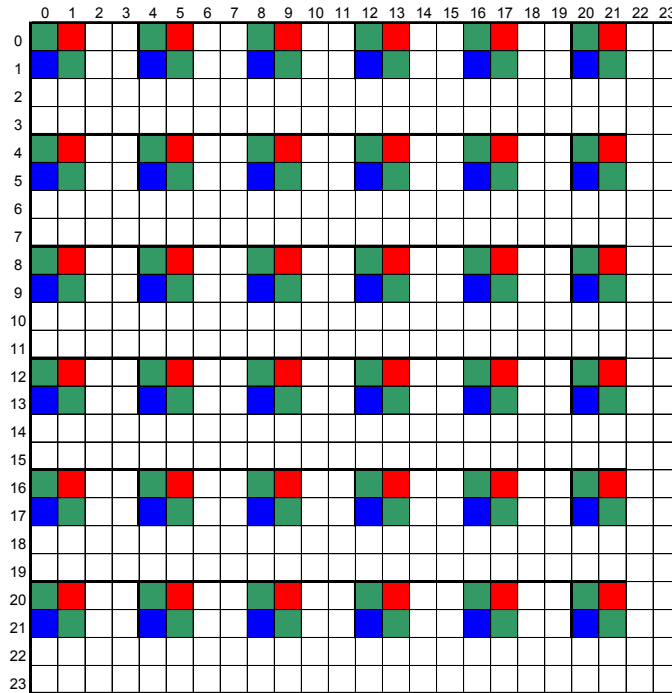
- Mode 1 every pixel group is read (no sub-sampling).
 - Mode 2 1 pixel group is read and 1 is skipped.
 - Mode 3 1 pixel group is read and 2 are skipped.
 - Mode 4 1 pixel group is read and 3 are skipped.
 - Mode 6 1 pixel group is read and 5 are skipped.
- (Say: "one-in-X are read, where X is the mode. e.g. one-in-six)

The modes can be set in the `CC_SetWOI` function in the form of the `XInc` and `YInc` parameter. The horizontal and vertical sub-sampling can be set independently. Each mode divides the total number of pixels in a frame by the corresponding factor. The maximum reduction is $6 \times 6 = 36$. This reduction results in the maximum frame rate. All sub-sampling modes can be combined with all WOI's.

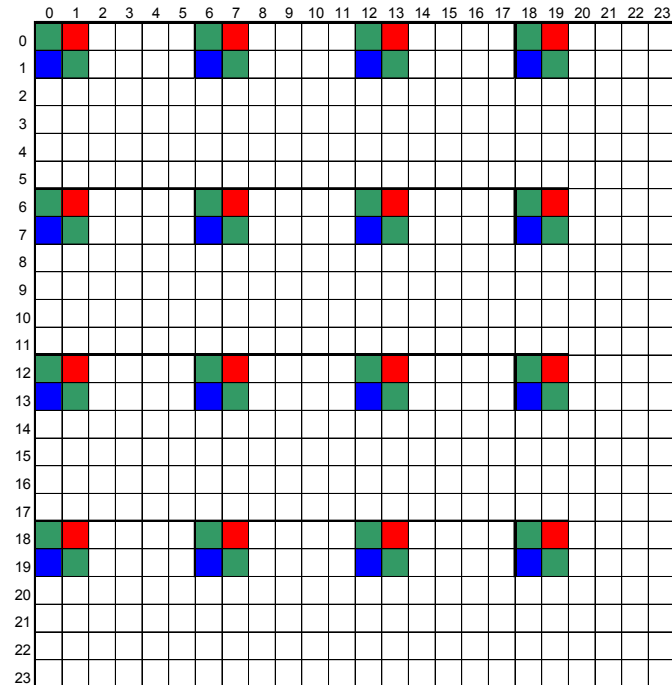
The sub-sampling repeats in blocks of 24 x 24 pixels, the next figures display the sub-sampling graphically :



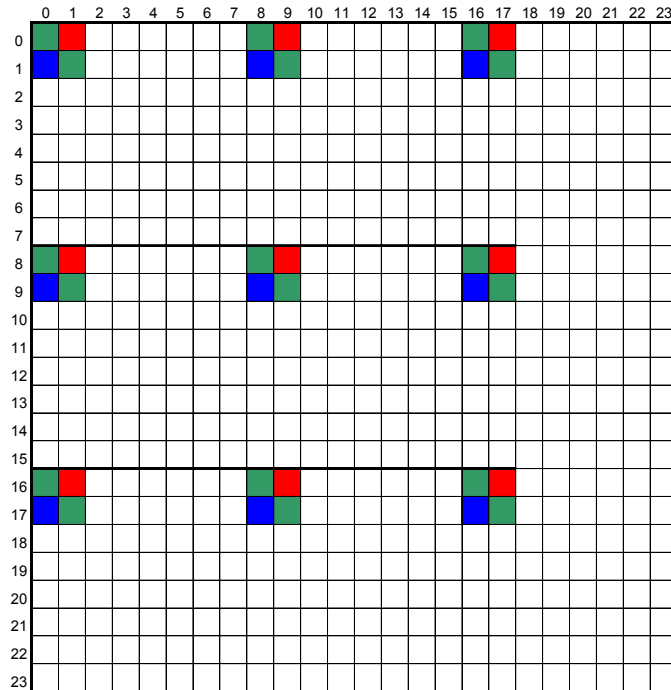
Sub-sample mode 1
 (Xinc = Yinc = 1)



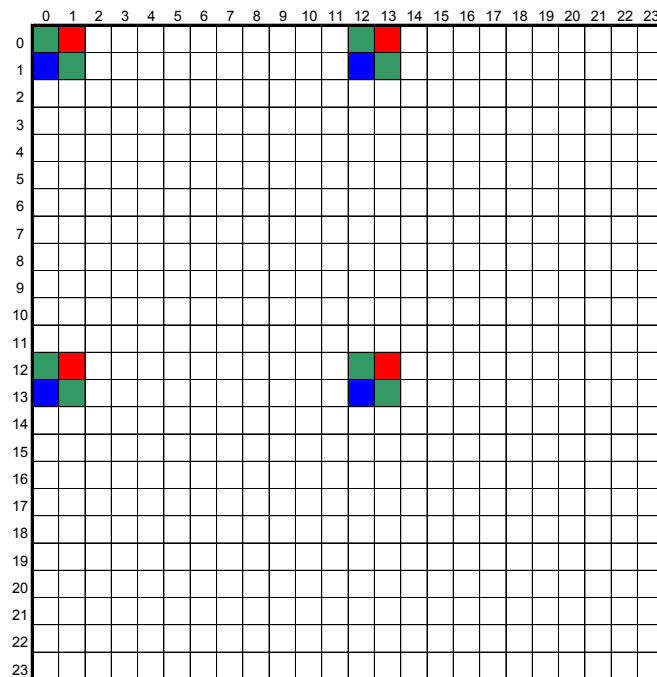
Sub-sample mode 2
(Xinc = Yinc = 2)



Sub-sample mode 3
(Xinc = Yinc = 3)



Sub-sample mode 4
(Xinc = Yinc = 4)



Sub-sample mode 6
(Xinc = Yinc = 6)

2.3 Frame Rate and Integration Time

2.3.1 Integration Time

The integration time can be set in microseconds, with the smallest time being 63 us. Due to the rolling shutter, each line is integrated for the same duration, but shifted in time. This gives distorted images for moving objects. The integration time and the currently set WOI determine how many lines are integrating at a certain point in time.

The rolling shutter has the advantage of reading out pixels while at the same time integrating other pixels for the next frame to be read out. This means that, when the integration time is shorter than the frame readout time, the frame rate is proportional to the frame readout time.

The camera can operate in continuous mode where the maximum frame rate can be achieved. There is also a snapshot mode (default mode) where a frame is captured following a software or hardware trigger. The disadvantage of this mode is that the integration time and the readout time occur in sequence (one after the other). This means that the frame time will be twice the frame readout time or twice the integration time, depending which one is the longest in duration. Therefore the maximum frame rate in snapshot mode will be half the frame rate of the continuous mode.

2.3.2 Frame Rate

The pixel rate of the camera is 40 M pixel per second (25 ns pixel period). The row blanking time is 7.15 us. The frame time can be calculated as follows :

$$frame-time = number-of-lines * (row-blanking-time + (number-of-pixels * pixel-period))$$

$$i.e. frame-time = number-of-lines * (7.15 + (number-of-pixels * 0.025)) \mu S$$

If the integration time is shorter than the frame time, then

$$frame_rate = \frac{1000000}{frame_time} \text{ (frames per second)}$$

If the integration time is longer than the frame time, then

$$frame_rate = \frac{1000000}{Integration_time} \text{ (frames per second)}$$

The frame time for a full WOI (2208 x 3000) is 187.05 ms which gives a maximum frame rate of 5.34 FPS.

2.4 Offset, Offset Fine

The offset can be set in 256 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 different odd and even column amplifier in the ibis 4-6600 sensor, differences in intensity can occur between the odd and even columns. The difference can vary from sensor to sensor in the manufacturing process. This 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 determined and set in the camera's non-volatile memory during production and should therefore not to be modified by the customer.

2.5 Gain

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

Gain setting	DC gain
0	1.36
1	1.64
2	1.95
3	2.35
4	2.82
5	3.32
6	3.93
7	4.63
8	5.40
9	6.35
10	7.44
11	8.79
12	10.31
13	12.36
14	14.67
15	17.38
127	1

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 no longer 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 region or the brighter region.

Unity gain can be set by writing 127 to the `CC_PAR_GAIN` parameter.

2.6 Data Mode

The BCi4-6600 camera is a 12-bit camera and transmits its image data with a pixel depth of 12 bits by default. The transfer time of an image depends on the number of bytes contained in the image and the type of interface being used. The number of bytes in an image is determined by the size of the WOI, possibly reduced by sub-sampling, and the pixel depth.

Reducing the pixel depth can decrease the transfer time of an image, which leads to a higher frame-rate in snapshot mode.

2.6.1 LS interface

The BCi4-6600 camera equipped with an LS interface can transfer a full frame with pixel depth of 12 bits in 331.2 msec. If the pixel depth is reduced to 10 bits, then the transfer time will be halved, to approximately 165.6 msec.

You can switch between 10-bit 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 these 2 modes.

BCi4-6600 Camera Mode	PCI interface card logic	Camera logic
12 bit	Pcilshu.ttb	Bci6lslogic2.ttb
10 bit	Pcilsrx10.ttb	Bci6lslogic2.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 USB camera is connected directly to the PC without an interface card, 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 Camera Parameters

3.1 Using the SDK with LS and USB interface

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

3.2 Camera commands when using CameraLink interface

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

4 External triggering

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

5 Programming example

5.1 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 which functions are needed and in which order. For a more complete (and working) example see the sample programs in the application directory.

This example opens the camera, initializes it, and then captures an image into a buffer and finally the camera is closed:

```
USHORT    buffer[2208*3000] ;
ULONG     picture_size ;
BOOL      ret ;
HANDLE    MyCam ;

MyCam = CC_Open( "BCi4-6600 USB", 0, CC_CAPTURE_WAIT ) ;
ret = CC_LoadCamera( MyCam, "bci6usblogic2.ttb" ) ;
ret = CC_SetWOI( MyCam, 0, 0, 2207, 2999, 1, 1, CC_WOI_LEFTTOP_RIGHTBOTTOM,
                &picture_size ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL0, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL1, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL2, 0 ) ;
ret = CC_SetParameter( MyCam, CC_PAR_ANAVAL3, 110 ) ;
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. (In this example CC_LoadInterface is not required as the camera has a USB interface.)

All other functions before the CC_CaptureSingle function can be called in any order. And the last function to call should be CC_Close.

Note that CC_SetWOI calculates and returns the size of the picture in pixels for later use.

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

5.2 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
F606	Y-start = 0
F760	
8000	
F6B7	Number of lines = 3000
F72B	
9BB7	
F603	X-start = 0
F750	
A000	
F64F	Width = 2208
F714	
B89F	
F600	No sub sampling
F770	
F6C4	Integration lines = 2500
F739	
E000	Line delay = 0
E100	
E200	
FC10	Frame time = 500 ms
FE20	
FEA1	
FE07	
FE00	
F600	Gain = 0
F780	
F664	Offset = 100
F790	
E800	Reset = 0
E900	Dual slope reset = 0
EB6E	Vmid = 110
FF82	Start the camera frame time controlled

6 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 5.2.

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.

7 Color filter geometry

All BCi4-6600 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;2207) and then all subsequent rows. The layout of the Bayer pattern:

```
GRGRGRG ...
BGBGBGB ...
GRGRGRG ...
BGBGBGB ...
...
```

8 Camera variations

The BCi4-6600 camera is available in the following variations :

Camera	Camera interface	PCI interface board	Camera Logic file	PCI board logic file
BCi4-6600 LS	Serial LVDS	PCI-LS	Bci6lslogic2.ttb	Pcilshu.ttb Pcilsrx10.ttb
BCi4-6600 USB	USB 2.0	-	Bci6usblogic2.ttb	-

Use the name in the "Camera" column in the CC_Open command to open the corresponding camera.

9 Appendix A : SDK parameters

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

Parameter	Values	Description
CC_PAR_XSTART	0 .. 2184 (multiples of 24)	See "Programmers Reference Manual"
CC_PAR_XEND	0 .. 2208 (even)	
CC_PAR_XINC	1, 2, 3, 4 or 6	
CC_PAR_YSTART	0 .. 2976 (multiples of 24)	
CC_PAR_YEND	0 .. 3000	
CC_PAR_YINC	1, 2, 3, 4 or 6	
CC_PAR_CAMERA_MODE	CC_CAMERA_NORMAL CC_CAMERA_DIAG_X	See "Programmers Reference Manual"
CC_PAR_DATA_MODE		See "Programmers Reference Manual"
CC_PAR_INTEGRATION_TIME	63 ... 107374180	Sets the integration time of the 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.
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 .. 255	Adds an analog offset to the output signal of the sensor.
CC_PAR_OFFSET_B_FINE	0 .. 255	The ibis 4-6600 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 highest reset voltage. 0 = 3.3V; 255 = 1.5V. The higher the voltage, the more the full well capacity. Set to zero.
CC_PAR_ANAVAL1	0 .. 255	Sets the reference voltage for dual slope operation. 0 = 3.3V; 255 = 1.5V
CC_PAR_ANAVAL2	0 .. 255	Sets the GND_AB voltage. 0 = 0V; 255 = 2.5V. Set to zero.

CC_PAR_ANAVAL3	0 .. 255	Sets the centre voltage for sensors output signal. This acts as an additional offset parameter. Set to 110.
CC_PAR_CTRLBIT	Bit 0, 2	<p>Bit setting 0 is default :</p> <p>Bit 0 = 0 sets the camera in 12 bit mode. 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)</p> <p>Bit 2 = 0 enable snapshot operation Bit 2 = 1 enable continuous operation</p>

10 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)	Range	Description
WOI Y start	F6xx F760	6 to 131	Start row = (value – 6) * 24
	8xxx	0 to 2976	multiples of 24 (must be equal as start row)
WOI Y height	F6xx F72x	0 to 2999	WOI height = value + 1
	9xxx	1 to 2999	must be equal as WOI height
WOI X start	F6xx F75x	3 to 95	Start column = (value – 3) * 24
	Axxx	0 to 2184	multiples of 24 (must be equal as start column)
WOI X width	F6xx F71x	0 to 1103	WOI width = (value + 1) * 2
	Bxxx	0 to 2207	must be equal as WOI width
Sub sampling	F6xx F770	0 to 4	0 = inc 1, 1 = inc 2, 2 = inc 3, 3 = inc 4, 4 = inc 6 bits (4:2) = X subsample, bits (7:5) = Y subsample
	C0xx D0xx	1,2,3,4,6	Must be equal to inc x or inc y respectively
Integration lines	F6xx F73x	0 to 3013	1) integration time < frame time (*) $WOI_HEIGHT - 1 - \text{inttime}(\text{ns}) / (WOI_WIDTH * 25 + 7050)$ 2) integration time >= frame time 3013
Line delay	E0xx (LSB) E1xx E2xx E3xx (MSB)	32 bit	1) integration time < frame time (*) 0 2) integration time >= frame time $(\text{inttime} * 40) / WOI_HEIGHT$
Frame time	FC10 FExx (LSB) FExx FExx (MSB)	24 bit	Frame time in microseconds

Offset	F6xx F790	0 to 255	
Offset fine	F6xx F7A0	0 to 255	
Gain	F6xx F780	0 to 15	Bits (3:0) = gain Bit 4 = unity gain
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 = x pattern, bit 5 = center value, bit(3:0) = data mode
Reset voltage	E8xx	0 to 255	Reset voltage
Multislope pixel reset voltage	E9xx	0 to 255	Multislope reset voltage
Vmid voltage	EBxx	0 to 255	Set to 210
Flash delay	FC04 FExx (LSB) FExx FExx (MSB)	24 bit	Flash delay in micro seconds
Flash width	FC08 FExx (LSB) FExx FExx (MSB)	24 bit	Flash width in micro seconds
Trigger control	D5xx		bit 0 = External input polarity, 0 = low, 1 = high bit 1 = Trigger output polarity, 0 = low, 1 = high bit 2 = Enable external trigger without start command bit 3 = Use trigger out as arm ready bit 4 = bit 5 = bit 6 = bit 7 =

Control bits	FFCx	<p>Set / Clear control switches Where x denotes: Bits 3,2,1 = bit number Bit 0 = 0 for Clear, = 1 for Set</p> <p>bit 0 = Cameralink Base bit mode (1 = 8-bit, 0 = 10/12-bit) bit 1 = not used bit 2 = continuous run bit 3 .. 7 not used</p> <p>e.g. FFC0 Cameralink mode 10/12-bit FFC1 Cameralink mode 8-bit</p>
--------------	------	--

11 Appendix C : Camera Link communication

11.1 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.

11.2 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 = $256 - \text{Mod}^{256}(02 + 00 + 00 + 0xBC + 0xFF + 0x80) = 0xC3$

11.3 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
0.1	30/6/2005	First Issue
0.2	14/7/2005	Minor changes (PMB)
0.3	28/10/2005	Changed unity gain value to 127
1.0	22/11/2007	Added paragraph 2.6.3 Changed chapter 5 Inserted chapter 6 Added Appendices A through C
1.1	08/04/2008	Minor changes (JB)
1.2	04/02/2009	Sect. 5.1 changed Vmid to 110 (PMB) Appendix 9 (same)