Astir2
User Guide

December 2017

Version 1.4624

Astrohn Technology Ltd.
Naugarduko str. 102, LT-03160 Vilnius, Lithuania
Phone +370 5 2034336
Fax +370 5 2034683
E-Mail info@astrohn.eu

Table of Content

Chapter 1. Introduction

Astir2 Infrared core is the latest generation uncooled thermal imaging camera core manufactured by Astrohn Technology, Ltd.

The core is based on a modern microbolometer from Ulis. It does not require temperature stabilization. The core does not have mechanical shutter either which means silent and reliable operation throughout the whole operating temperature range. Advanced image processing algorithms guarantee vivid and detailed image.

Through the use of additional expansion boards, core supports many different types of analog and digital video output formats, has flexible control interfaces. Highly customizable functionality along with small form factor and weight ensures its suitability for a vast range of applications.

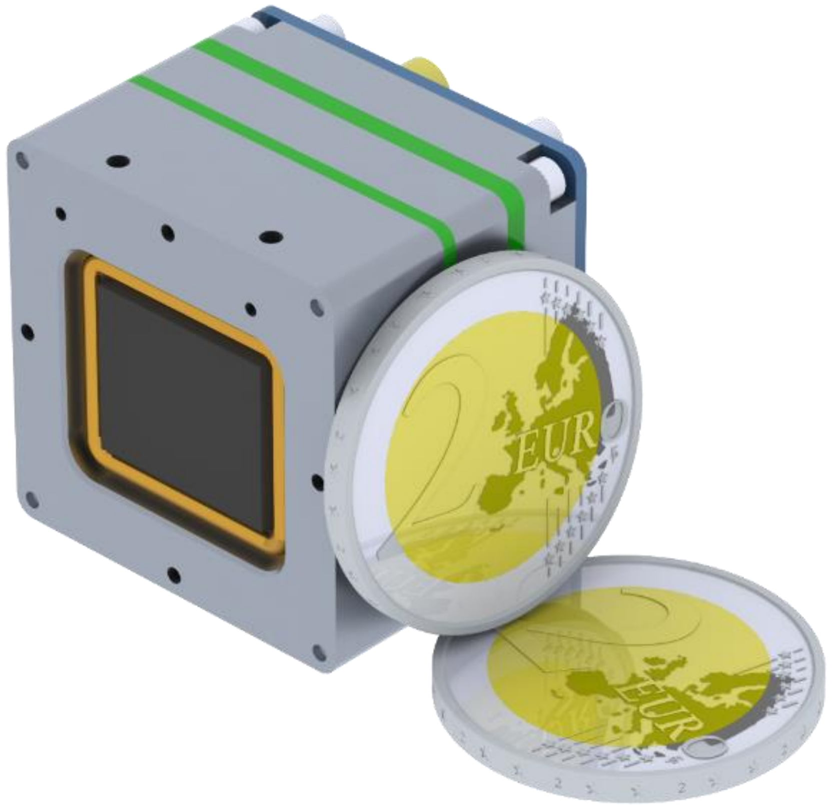


Figure 1. Astir2 thermal imaging core

Chapter 2. Technical Specifications

Table 1 below summarizes main technical characteristics for the Astir2 thermal imaging core.

Table 1. General Astir2 core characteristics

General	
Storage ambient temperature	-55^o to +105^o
Operating ambient temperature	-40^o to +60^o
Power-up time	<3.5 s
Weight (bare core without lens)	32 g (640 x 480); 36 g (384 x 288)
Size (bare core without lens)	30 x 30 x 23.5 (640 x 480) / 30 x 30 x 25.0 (384 x 288)
Mounting holes	4 x M1.6 (front) and 4 x 2 x M2 (sides)
Power supply	4.5 - 5.5 V or 3 - 12 V (1)
Sensor Characteristics	
Sensor type	Uncooled amorphous silicon microbolometer
Spectral range	8 – 14 µm
Typical sensitivity (NETD)	< 55 mK
Resolution	640 x 480 or 384 x 288
Image frequency	25 Hz (640 x 480), 50 Hz (384 x 288)
Pixel pitch	17 µm
Image processing	
Digital zoom	x1, x2, x4, x8, Continuous
Temporal noise filtering	Yes
Image sharpening	Yes (variable)
AGC	Advanced analysis of separate scene parts
Bad pixel detection/correction	List based; automatic list based; real time
Graphics overlay	Pixel based with 256 color palette (2)
Interfaces	
Video output	Analog (PAL), Digital (BT.656, Custom-1, Custom-2, CameraLink) (3)
Control	UART, RS232, SPI, QuadSPI, USB (4)

Notes:

1. Power supply voltage range depends on the extension boards used.
2. Some of the 256 colors from the graphics overlay color palette are reserved for special purposes (transparency, invertible color, etc.)
3. Available video output interfaces depends on the extension boards used.
4. Available control interfaces depends on the extension boards used.

Core features dependant on the extension board are given in Table 2.

Table 2. Extension board dependent features of the Astir2 core

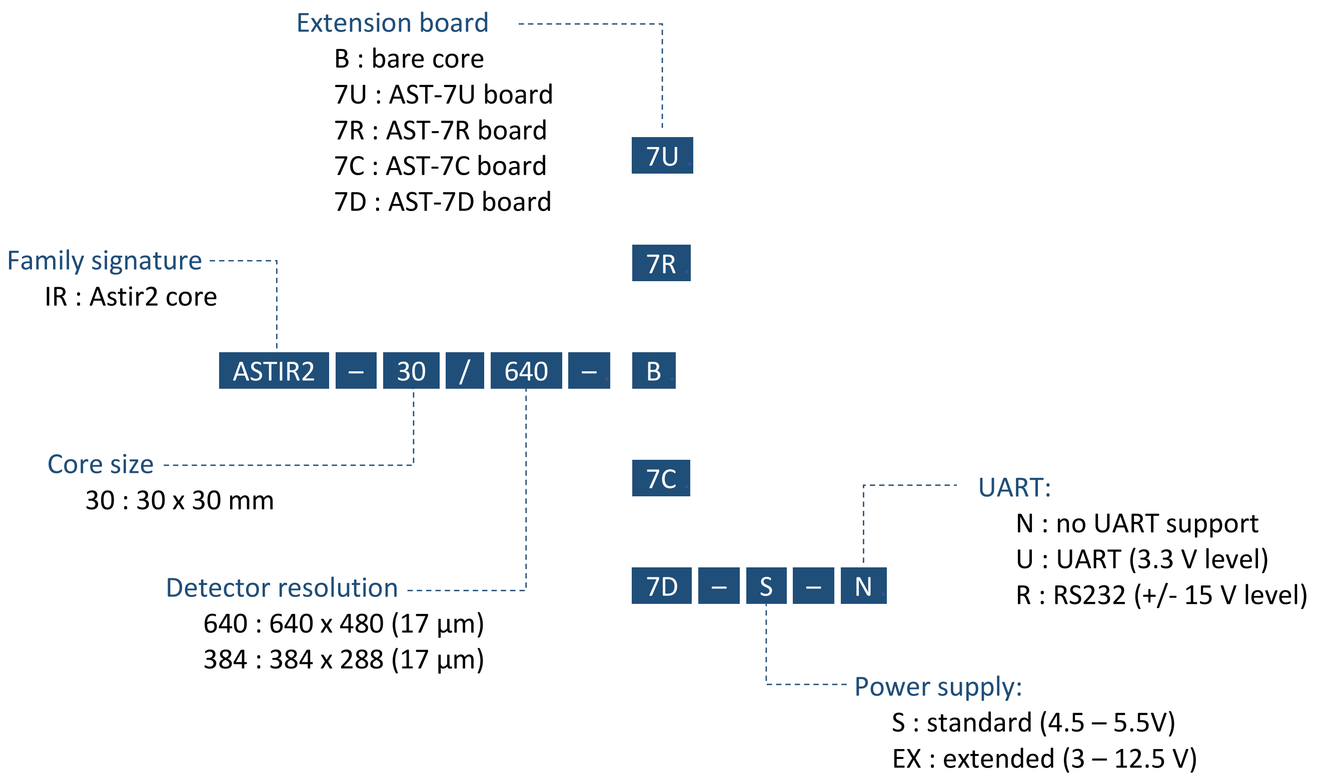
	Bare core	AST-7D	AST-7U	AST-7R	AST-7C
Power supply					
Voltage	4.5 - 5.5 V	4.5 - 5.5 V / 3 - 15 V(1)	4.5 - 5.5 V	3 - 15 V	3 - 15 V
Power consumption	1.2 W	1.4 W	1.2 W	1.4 W	1.5 W
Video output					
Analog PAL	Yes	Yes		Yes	
Digital BT.656	Yes		Yes		
Digital CameraLink					Yes
Digital Custom-1	Yes				
Digital Custom-2	Yes				
Control interface					
USB		Yes			
RS232		Yes (2)		Yes	
UART	Yes	Yes (2)	Yes	Yes	Yes (3)
SPI	Yes				
QuadSPI	Yes				

Notes:

1. Power supply voltage range depends on the configuration of the board (-S or -EX)
2. Available interface type (RS232 vs UART) depends on the configuration of the board (-R or -U)
3. With AST-7C board UART commands can be sent and response received over two differential pairs on CL connector.

Chapter 3. Naming

Astir2 core can be ordered as bare core (without any extension boards) or with certain extension boards that simplify interfacing with the core. Diagram below summarizes different options available. Extension boards can be ordered separately from the core.



Chapter 4. Extension boards

Bare core

Bare core is the smallest possible version of the Astir2 core consisting of two PCBs (analog sensor board and digital processor board). At the back of the bare core there is a 70 pin Hirose connector DF17(4.0)-70DP-0.5V(57) exposed. Through this connector core is powered, controlled (UART, SPI, QuadSPI interfaces) and outputs video (Analog video output (PAL) and/or Digital video output) signal. Pinout for the 70-pin connector is given in the table below. Refer to the bare core mechanical interface drawing.

Table 3. Astir2 bare core 70-pin connector pinout

Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.
1	A_VIDEO_COM	Out	19	GND	Pwr	37	D_VIDEO_D3	Out	55	Core status LED	Out
2	A_VIDEO_OUT	Out	20	GND	Pwr	38	D_VIDEO_D7	Out	56	SPI_SCK	In
3	Reserved		21	Reserved		39	GND	Pwr	57	Reserved	
4	Reserved		22	UART_RX	In	40	GND	Pwr	58	SPI_SDO	Out
5	Reserved		23	Reserved		41	D_VIDEO_CLK	Out	59	GND	Pwr
6	Reserved		24	D_VIDEO_MODE	Out	42	QSPI_D0	Bi	60	GND	Pwr
7	Reserved		25	Reserved		43	Reserved		61	Reserved	
8	Reserved		26	D_VIDEO_VS	Out	44	QSPI_D1	Bi	62	SPI/QSPI_READY	Out
9	GND	Pwr	27	D_VIDEO_HS	Out	45	Reserved		63	QSPI_CLK	In
10	GND	Pwr	28	Reserved		46	QSPI_D2	Bi	64	SPI_SDI	In
11	Reserved		29	GND	Pwr	47	Reserved		65	GND	Pwr
12	Reserved		30	GND	Pwr	48	QSPI_D3	Bi	66	Reserved	
13	Reserved		31	D_VIDEO_D0	Out	49	GND	Pwr	67	Power supply - GND	Pwr
14	Reserved		32	D_VIDEO_D4	Out	50	GND	Pwr	68	Power supply	Pwr
15	Reserved		33	D_VIDEO_D1	Out	51	Reserved		69	Power supply - GND	Pwr
16	Reserved		34	D_VIDEO_D5	Out	52	Reserved		70	Power supply	Pwr
17	Reserved		35	D_VIDEO_D2	Out	53	External sync. input	In			
18	UART_TX	Out	36	D_VIDEO_D6	Out	54	Reserved				

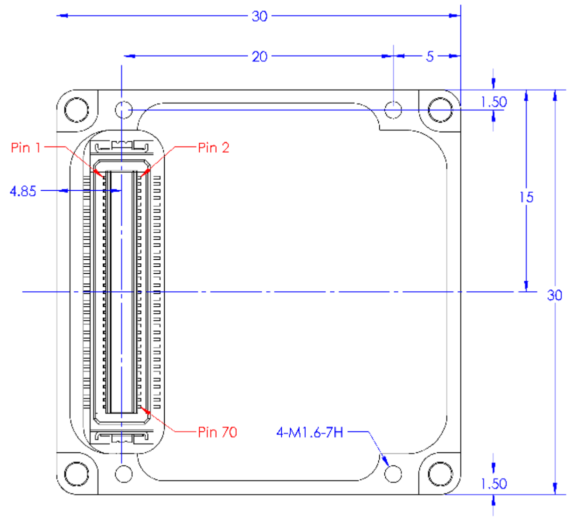


Figure 2. Astir2 bare core mechanical interface drawing

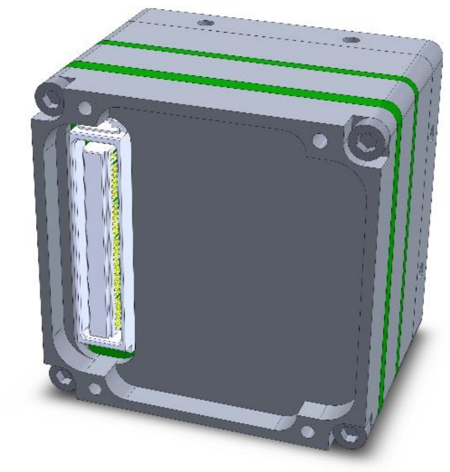


Figure 3. Astir2 bare core

AST-7D

AST-7D board is connected to bare core's 70-pin connector. The board provides easier access to cores power supply and analog video output. Additionally, it allows the core to be connected to the PC over the USB. PC Software can then be used to adjust all core's settings, perform gain calibration, capture digital snapshots, update firmware, etc.

The board also features connector for UART/RS232 interface. Depending on the board version voltage levels can be +3.3V (UART) or +/-15V (RS232). The +/-15V levels are useful if for instance the core is to be connected to the PC's RS232 port. For more information regarding UART/RS232 interface settings to communicate with the core see UART/RS232 interface section.

Power supply range can also be chosen when ordering. Default version uses 4.5 – 5.5 V power supply input. Extended power supply range is 3 – 15 V (in which case power consumption will increase by approximately 10 %).

The board has LEDs to indicate power supply, core, and communication over USB status.

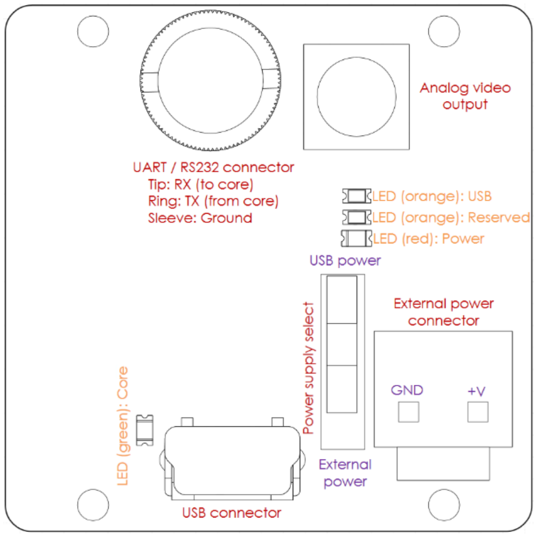


Figure 5. AST-7D board functional view

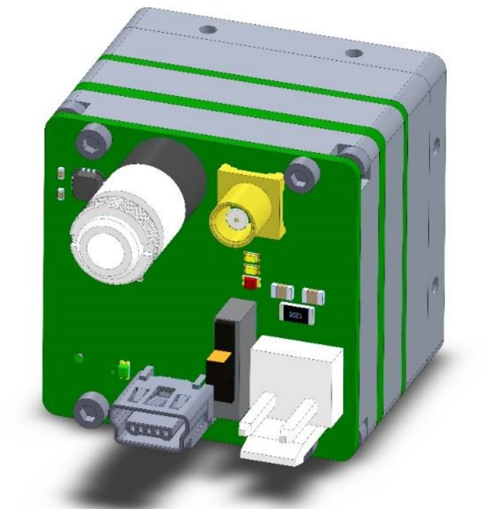


Figure 4. Astir2 core with AST-7D board

AST-7U

AST-7U board has single 20-pin flex cable connector (Mollex 52745-2097). Example flex cable: 0210200211 from Molex.

Power supply, digital video output (without external synchronization signals) and UART core control are all available on the single connector. Connector pinout is given in the table below.

Power supply range: 4.5 – 5.5 V. All video output signals as well as core control signals (UART_RX and UART_TX) have 3.3V levels. With AST-7U board TX and RX signals have inverted polarity (compared to bare core version).

Table 4. AST-7U board flex cable connector pinout

Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.
1	UART_RX	In	6	D_VIDEO_D0	Out	11	GND	Pwr	16	D_VIDEO_D5	Out
2	UART_TX	Out	7	Power supply	Pwr	12	D_VIDEO_D3	Out	17	GND	Pwr
3	Power supply	Pwr	8	D_VIDEO_D1	Out	13	GND	Pwr	18	D_VIDEO_D6	Out
4	D_VIDEO_CLK	Out	9	GND	Pwr	14	D_VIDEO_D4	Out	19	GND	Pwr
5	Power supply	Pwr	10	D_VIDEO_D2	Out	15	GND	Pwr	20	D_VIDEO_D7	Out

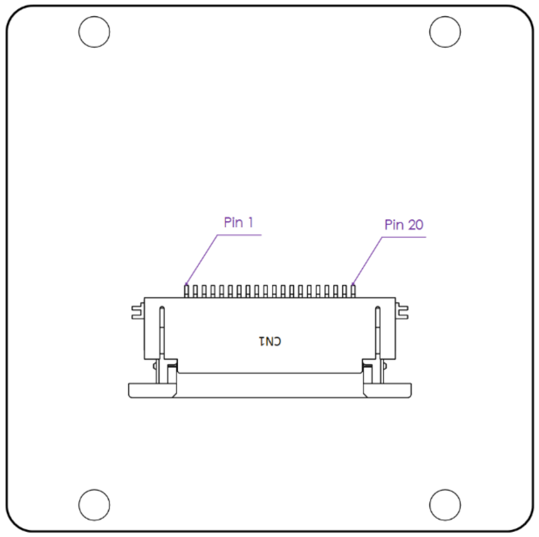


Figure 7. AST-7U board functional view

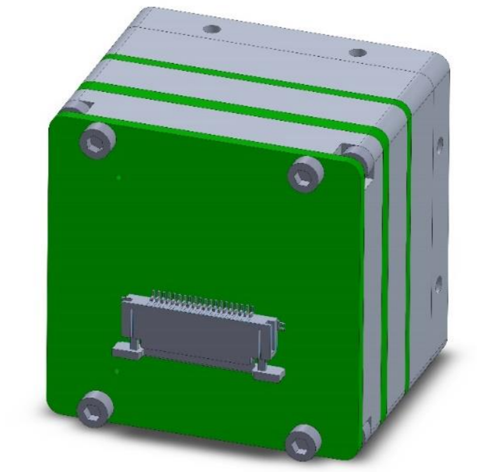


Figure 6. Astir2 core with AST-7U board

AST-7R

The board has three connectors:

- B2P-VH(LF)(SN) – power supply for the core
- B3P-VH(LF)(SN) – analog video output
- B4P-VH(LF)(SN) – UART / RS232 interface

Power supply has to be in the range of 3 - 15 V. UART signal levels are 3.3V. Two LEDs show power supply and core status.

Refer to the board drawing below for the pinout of the connectors.

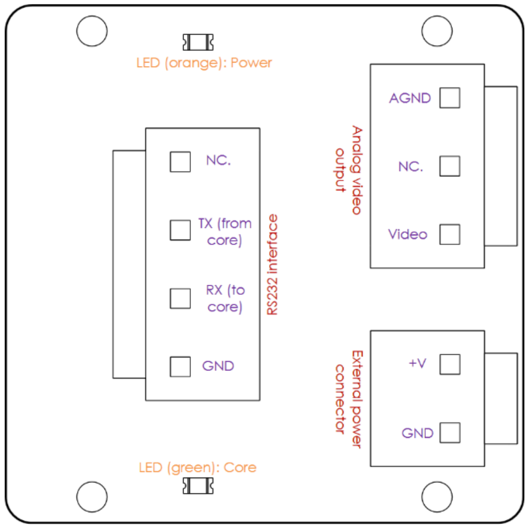


Figure 9. AST-7R board functional view

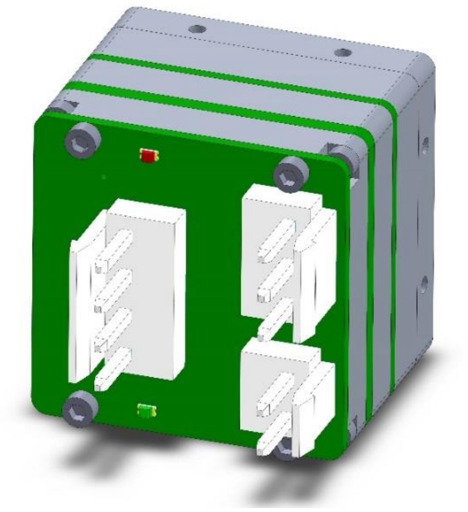
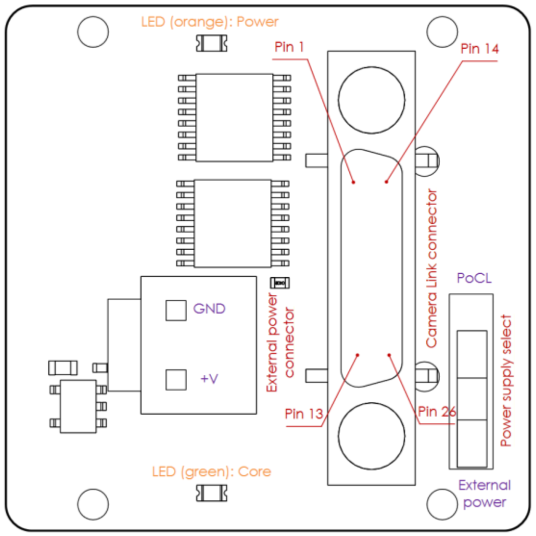


Figure 8. Astir2 core with AST-7R board

AST-7C

The board has female SDR connector through which the core can be controlled and digital CameraLink video output can be accessed. The core can be powered through either external power supply connector or the same CameraLink connector.

Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.	Pin	Description	Dir.
1	NC		8	TFG_n	Out	15	X0_p	Out	22	NC	
2	X0_n	Out	9	NC		16	X1_p	Out	23	NC	
3	X1_n	Out	10	NC		17	X2_p	Out	24	NC	
4	X2_n	Out	11	NC		18	CLOCK_p	Out	25	Reserved	
5	CLOCK_n	Out	12	Reserved		19	X3_p	Out	26	Power supply	Pwr
6	X3_n	Out	13	GND	Pwr	20	TC_n	In			
7	TC_p	In	14	NC		21	TFG_p	Out			



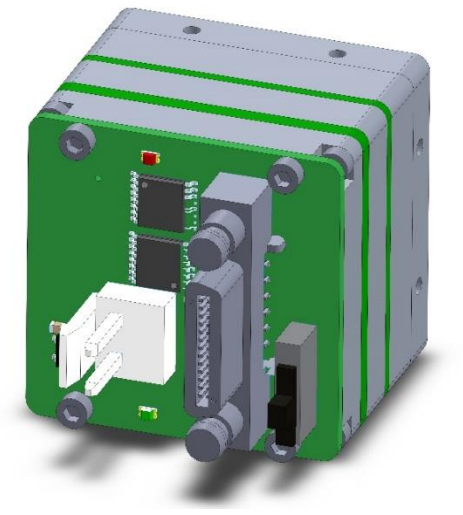


Figure 10. AST-7C board functional view

Chapter 5. Video output

Analog video output (PAL)

PAL is default video output for the core. Pal video data contains interlaces rows. Astir2 core with 640 x 480 resolution sensor uses video data form the same sensor frame for odd and even video rows. Astir2 core with 384 x 288 resolution sensor uses video data from different sensor frames for odd and even video rows. Each full frame (two interlaced half frames) consists of 625 rows (576 of which contains active video data).

Digital video output

Table 5. Summary of digital video output interfaces

Output type	Pixel clock frequency	Full resolution	Image resolution	Interlaced	Frame rate (full frame)	Color subsampling	Bit depth
Custom-1	108 MHz	800x600	800x600	No	50	4:4:4	8
Custom-2	27 MHz	Sensor resolution	Sensor resolution	No	Sensor frame rate	4:2:2	8
BT.656	27 MHz	720x576	720x576	Yes	25	4:2:2	8
BT.656 centered	27 MHz	720x576	Sensor resolution	Yes	25	4:2:2	8
CameraLink	20 MHz	640x480	640x480	No	25	Mono	8

Notes:

- For 640 x 480 resolution Astir2 cores (where sensor operates at 25 Hz) and 50 Hz video output mode every sensor frame is repeated twice in video output.
- For 384 x 288 resolution Astir2 cores (where sensor operates at 50 Hz) and 25 Hz video output mode every second sensor frame is omitted (data before omitting the frame is still used in temporal video filter)
- CameraLink video output is currently only supported by 640 x 480 resolution Astir2 core.

BT656

Core video output can be set to BT656 digital mode. Output signal is compliant with ITU-R recommendation for BT.656.

Video is clocked out using D_VIDEO_CLK signal (see bare core pinout) at 27 MHz frequency. Video data is output on pins D_VIDEO_D0 through D_VIDEO_D7.

Each row consists of 864 double words, with 720 double words of active video data. 4 double words are used for timing reference code.

Video rows are interlaced (Data from the same raw frame for odd and even video rows in Astir2 cores with 640 x 480 resolution sensor. Data from two consecutive sensor frames is used for odd and even video rows in Astir2 cores with 384 x 288 resolution sensor), with 625 rows comprising full frame (two interlaced half frames). Each frame contains of 576 rows of active video data. Full frame refresh frequency is 25 Hz.

BT.656 video is available on Bare core or on the flex cable connector on AST-7U board.

BT656 centered

In BT656 centered video output mode video frames have the same resolution and timing parameters as in standard BT656 mode. Output image however is not being scaled up to match video output resolution – it is kept at sensor resolution (either 640 x 480 or 384 x 288) and positioned at the center of full video output frame.

Custom-2

Custom-2 video output mode uses the same color subsampling (4:2:2), pixel clock frequency (27 MHz) and timing reference code format as BT656 video output mode. Video output, however, is not interlaced and the frame rate matches sensor’s frame rate (50 Hz for 384 x 288 core and 25 Hz for 640 x 480 core).

Other timing parameters are as follows. For 640 x 480 core:

- Full line duration: 2 x 720 clock cycles
- Active line duration: 2 x 640 clock cycles
- Total number of lines: 750
- Active lines: 480

For 384 x 288 core:

- Full line duration: 2 x 600 clock cycles
- Active line duration: 2 x 384 clock cycles
- Total number of lines: 450
- Active lines: 288

Custom-1

Custom-1 video output mode outputs video data with 108 MHz pixel clock (D_VIDEO_CLK signal) frequency. Three color components (red, green, blue) are transferred for each individual pixel over 8 bit data bus (signals D_VIDEO_D0 through D_VIDEO_D7). Video output resolution is 800 x 600, frame rate: 50 Hz (non-interlaced).

Below are horizontal and vertical timing diagrams.

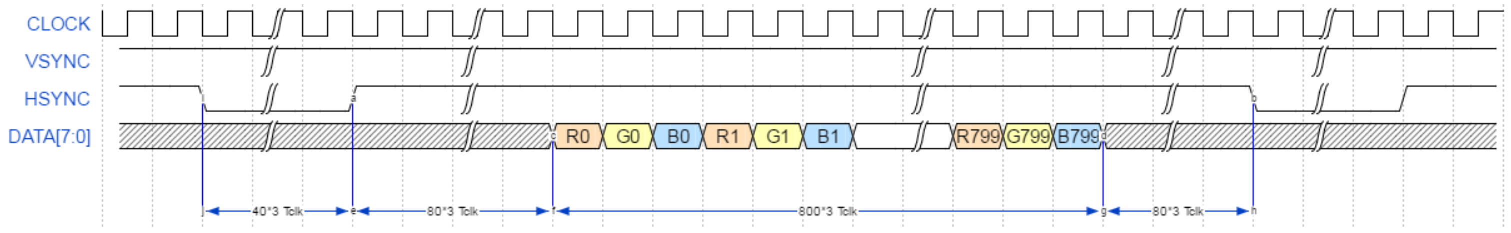


Figure 11. Horizontal timing diagram for Custom-1 digital video output

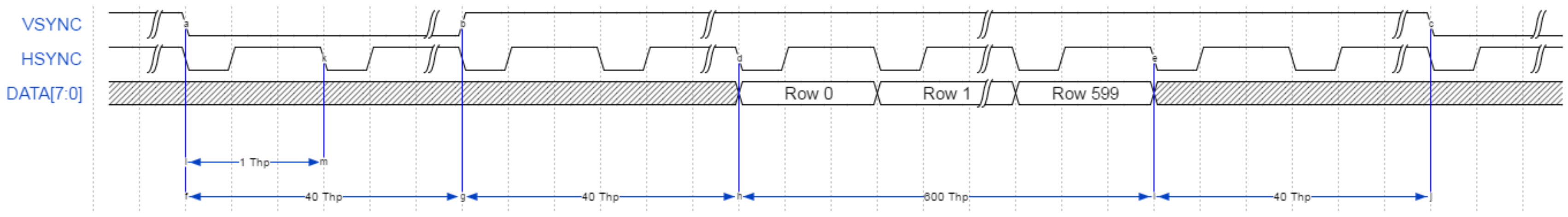


Figure 12. Vertical timing diagram from Custom-1 digital video output

Camera Link

CameraLink video output is available only for 640 x 480 resolution Astir2 core if AST-7C extension board is used. CameraLink configuration is 1X2-1Y (12 bits per pixel, least significant 4 bits are equal to zero). CameraLink clock frequency: 20 MHz.

Figure 13 below shows the order CameraLink bits are transferred over four differential pairs on CameraLink connector on AST-7C board. Similarly, Figure 12 shows the order pixel value bits (D0[n] and D1[n]) along with synchronization signals (data valid, frame valid, line valid) are transferred over CameraLink differential signal pairs. Additionally, the mapping between CameraLink bits and pixel value bits is given in the Table 6.

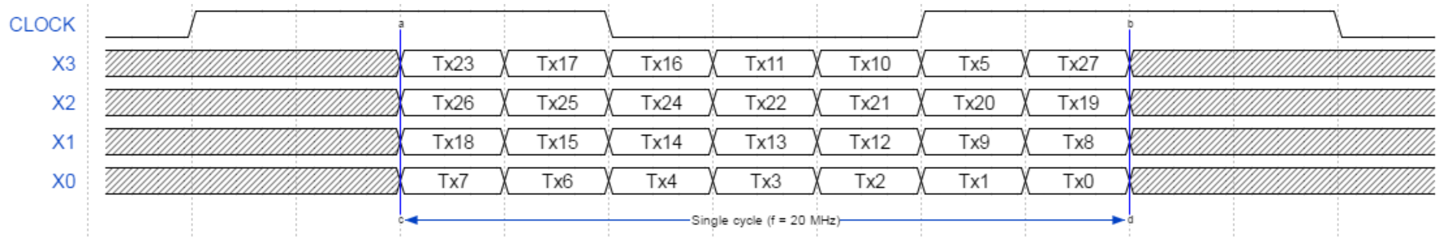

CLOCK
X3
X2
X1
X0
Tx23 Tx17 Tx16 Tx11 Tx10 Tx5 Tx27
Tx26 Tx25 Tx24 Tx22 Tx21 Tx20 Tx19
Tx18 Tx15 Tx14 Tx13 Tx12 Tx9 Tx8
Tx7 Tx6 Tx4 Tx3 Tx2 Tx1 Tx0
Single cycle (f = 20 MHz)

Figure 13. CameraLink bit order during single clock cycles

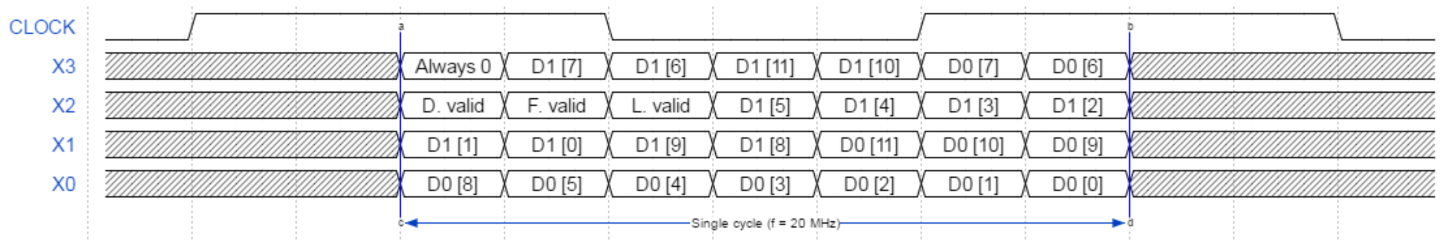


Figure 14. Pixel data bit order during single clock cycles

Table 6. CameraLink bit mapping

Video pixel	CameraLink port	CameraLink signal	Video pixel	CameraLink port	CameraLink signal
D0 [0]	PortA[0]	Tx0	**D1 [0]**	PortC[0]	Tx15
D0 [1]	PortA[1]	Tx1	**D1 [1]**	PortC[1]	Tx18
D0 [2]	PortA[2]	Tx2	**D1 [2]**	PortC[2]	Tx19
D0 [3]	PortA[3]	Tx3	**D1 [3]**	PortC[3]	Tx20
D0 [4]	PortA[4]	Tx4	**D1 [4]**	PortC[4]	Tx21
D0 [5]	PortA[5]	Tx6	**D1 [5]**	PortC[5]	Tx22
D0 [6]	PortA[6]	Tx27	**D1 [6]**	PortC[6]	Tx16
D0 [7]	PortA[7]	Tx5	**D1 [7]**	PortC[7]	Tx17
D0 [8]	PortB[0]	Tx7	**D1 [8]**	PortB[4]	Tx13
D0 [9]	PortB[1]	Tx8	**D1 [9]**	PortB[5]	Tx14
D0 [10]	PortB[2]	Tx9	**D1 [10]**	PortB[6]	Tx10
D0 [11]	PortB[3]	Tx12	**D1 [11]**	PortB[7]	Tx11

Figure 15, shows vertical timing for the CameraLink video output. Each row is 320 clock cycles long (two pixels are transmitted simultaneously in 1X2-1Y CameraLink configuration). Inter-row time is equal to 1280 clock cycles. After last row, Frame valid is active for 4096 clock cycles. After additional 29184 clock cycles new frame is started. Total frame time is 800000 clock cycles.

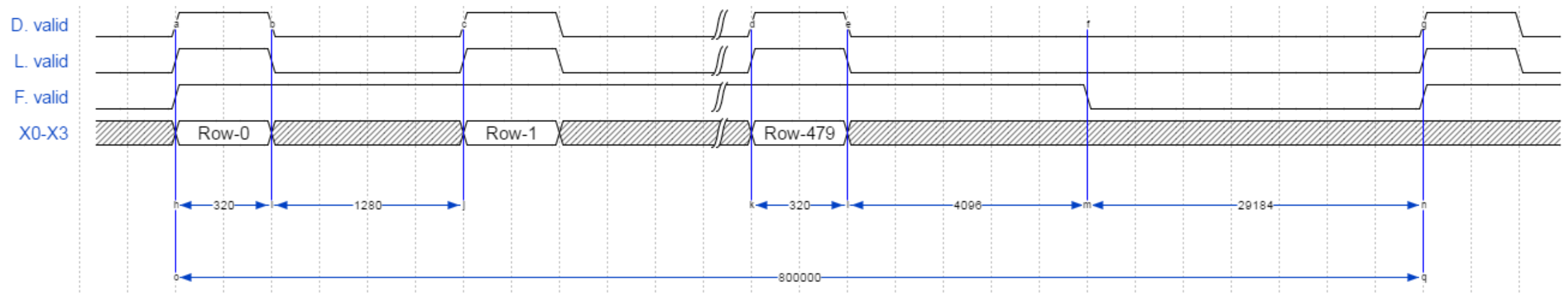


Figure 15. Vertical timing diagram for CameraLink video output

Chapter 6. Core control

Core status can be monitored and operation can be controlled by accessing internal core registers or memories.

Direct access to the registers is available over SPI or QuadSPI interfaces.

Additionally, register values can be changed (core can be controlled) over UART interface. In this case sending command to the core will set one or more registers to appropriate values. Commands available and registers modified by each command are described in UART section of the document. Core status (values of the main registers which determine core operation) can also be read over UART interface. One particular UART command (*"rw <p1> <p2> <CR>"*) can be used to set any register to any value. However, arbitrary register values cannot be read over UART.

Internal core memories can only be accessed over SPI or QuadSPI interfaces.

To avoid image change mid-frame most of the registers related to video output are synchronized with the start of video frame.

Registers can be virtual or real. Real registers store their value and in many cases can be read back (for example output image's brightness register). Some real registers are read only registers (for example the ones containing firmware version of the core). Virtual registers, on the other hand, are used to activate certain operations within the core (for example to start automatic bad pixel detection). The operation is started every time appropriate value is written to the virtual register – it does not have to be reset.

All registers are 1 byte in size. Certain settings may require values that exceed the range of 0 to 255. In those cases two or more registers are grouped together to store single value. Then register with the lowest address contains 8 least significant bits of the value.

List of core registers

Table 7 is a summary of all core registers. Register address is given in the address column. Several addresses are written in cases where more than one register is used to store certain core settings value. Real / Virtual column shows whether the register is real or virtual. For some registers several bits are real and the rest are virtual (for example register 0x45). Whether the register is readable, writable or both is shown in R/W column. Bits column shows the effective size (in bits) of a register or group of registers. Some registers have default values after power up (however most of the registers related to image settings are overwritten during power up with values read from the core's non-volatile memory). If applicable, default values are also given in the table. Valid values column shown values (or range of values) that can be written to the register(s).

Registers that can be changed by sending dedicated UART command to the core are marked with '+' in UART cmd. column. Note that all writable registers can be changed by using single write register UART command, even if dedicated UART command is not available. Finally, description column briefly describes the purpose of the register (or group or registers). More detailed description of each of the registers is given below the table.

Table 7. List of Astir2 core registers

Address	Real / Virtual	R/W	Bits	Default value	Valid values	UART cmd.	Description
0x01, 0x00	R	R	[15:0]	N/A	N/A	+	Firmware version
0x03, 0x02	R	R	[15:0]	N/A	N/A	+	Firmware revision
0x04	R	R	[7:0]	N/A	N/A	+	Sensor ID code
0x08, 0x07, 0x06	R	R	[23:0]	N/A	N/A	+	FPA temperature code
0x08	V	W	[7:0]	N/A	1	+	Save user settings to flash
0x0F, 0x0E	R	R	[15:0]	N/A	N/A	+	Firmware variant
0x20	R	W	[7:0]	0	0, 1	-	Suspends list based bad pixel corrections.
0x44	R	R/W	[7:0]	0	0 - 3	-	Digital video output clock phase
0x45	R, V	W	[7:0]	0	0, 1, 16, 17	-	Graphics overlay control
0x46	R	W	[7:0]	0xFF	0x00 – 0xFF	-	Graphics overlay clear color
0x47	R	W	[7:0]	0x20	0x00 – 0xFF	-	Graphics overlay trigger area size
0x48	R	W	[7:0]	0x80	0x00 – 0xFF	-	Graphics overlay trigger hysteresis
0x4A, 0x49	R	W	[9:0]	0	0x000 – 0x3FF	-	Digital image offset (horizontal)
0x4C, 0x4B	R	W	[9:0]	0	0x000 – 0x3FF	-	Digital image offset (vertical)
0x63	R, V	W	[7:0]	N/A		+	External sync trigger
0x75, 0x74	R	R/W	[15:0]	1	0x0000 – 0xFFFF	+	Contrast bias
0x76	R	R/W	[7:0]	0x28	0x00 – 0xFF	+	Contrast
0x77	R	R/W	[7:0]	0x00	0x00 – 0xFF	+	Brightness
0x78	R	R/W	[7:0]	0x02	0x00 – 0x04	+	BC mode
0x79	R	R/W	[7:0]	0x00	0x00 – 0x02	+	Division factor
0x7B, 0x7A	R	R/W	[15:0]	0x00	0x0001 – 0xFFFF	+	Extra contrast
0x7C	R	R/W	[7:0]	0x7D	0x00 – 0xFF	+	Maximum contrast

Address	Real / Virtual	R/W	Bits	Default value	Valid values	UART cmd.	Description
0x7D	R	R/W	[7:0]	1		+	Video output
0x7F	R	R/W	[7:0]	0x07		+	Temporal filter
0x80	R	R/W	[7:0]	0x00		+	Reserved for gamma (not used)
0x82	R	R/W	[7:0]	0x07		+	Destriping filter
0x83	R	R/W	[7:0]	1		+	Sharpening filter
0x86	R	R/W	[7:0]	0		+	Image flip
0x9C	V	W	[7:0]	N/A	0x01, 0x02	+	Bad pixel detection (flash mode) control
0x9D	R	W	[7:0]	0x87		+	Bad pixel detection (real time) control
0x9E	R	W	[7:0]	20		+	Bad pixel detection sensitivity (real time)
0x9F	R	W	[7:0]	30		+	Bad pixel detection sensitivity (flash mode)
0xA0	V	W	[7:0]	N/A	1	+	Re-initialize the core
0x0A5, 0xA4, 0xA3, 0xA2	R	R	[31:0]	N/A		+	Core serial number
0xA9, 0xA8, 0xA7, 0xA6	R	R	[31:0]	N/A		+	Sensor serial number
0xB0	R	R/W	[7:0]	1		-	Image resolution mode
0xB1	R	R/W	[7:0]	0x5A		-	Image resolution horizontal
0XB2	R	R/W	[7:0]	0x90		-	Image resolution vertical
0xD3, 0xD2	R	R/W	[15:0]	2328		+	Pixel threshold
0xD5	R	R/W	[7:0]	0		+	Palette and polarity
0xD6	R	R/W	[7:0]	0	0, 1, 2, 4, 8, 254, 255	+	Discrete zoom factor
0xD9, 0xD8	R	W	[15:0]	0		-	Rows start
0xDB, 0xDA	R	W	[15:0]	*		-	Rows count (default value is equal to sensors rows count)
0xDD, 0xDC	R	W	[15:0]	0		-	Column start
0xDF, 0xDE	R	W	[15:0]	*		-	Column count (default value is equal to sensors column count)
0xE3, 0xE2	R	W	[15:0]	0		+	Continuous zoom value

Registers 0x01, 0x00

Read only register pair contains firmware version of the core.

Registers 0x03, 0x02

Read only register pair contains firmware revision of the core.

Register 0x04

Read only register contains sensor ID code. For Astir2 core the following values are valid:

- 0x05: Sensor used is 640 x 480 resolution, 17 um pixel pitch, amorphous silicon detector from Ulis
- 0x06: Sensor used is 384 x 288 resolution, 17 um pixel pitch, amorphous silicon detector from Ulis

Registers 0x08, 0x07, 0x06 (readable)

Three readable registers contain temperature code. Read temperature code can be converted to temperature expressed in degrees Celsius by using the following equations:

For 0x05 sensor (640 x 480): $T = 94.4306 - {}^{T_{code}}/_{37044.1}$

For 0x06 sensor (384 x 288): $T = {}^{T_{code}}/_{99321.1} - 58.2162$

Registers 0x08 (writable)

Writing 0x01 to this register will store current core settings in non-volatile memory. For the list of settings that are saved to / loaded from non-volatile memory refer to Saving configuration chapter. Make sure that the core is powered for at least 3 seconds after the command has been issued. Otherwise settings stored in non-volatile memory will get corrupt. If corrupt settings are loaded after next power up, the output image will most likely be corrupt / absent. From this condition core can be recovered by either setting all core settings (registers) that are loaded from non-volatile memory to valid values or by connecting the core to the PC with AST-7D board and reflashing core's firmware using Chapter 7. CoreConfig PC Software. Reflashing core's firmware will restore core's settings to working values.

Registers 0x0F, 0x0E

Read only register pair contains firmware variant of the core. Different core hardware requires appropriate firmware variant. Differences can be sensor or other hardware features related. So far the following firmware variants have been used:

- 0x0010: Astir2 with 0x06 sensor and large FPGA (hardware configuration is obsolete and no longer manufactured; firmware file name: starts with ‘Astir2_a7_3a’)
- 0x0020: Astir2 with 0x05 sensor and large FPGA (hardware configuration is obsolete and no longer manufactured; firmware file name: starts with ‘Astir2_a7_3b’)
- 0x0020: Astir2 with 0x06 sensor and small FPGA (firmware file name: starts with ‘Astir2_a5_3a’)
- 0x0021: Astir2 with 0x05 sensor and small FPGA (firmware file name: starts with ‘Astir2_a5_3b’)

Register 0x20

Writing 0x01 to this register will disable list based bad pixel correction. Real time bad pixel correction can however still be functional (it is separately controlled by 0x9D register). When both types of bad pixel correction are disabled all bad pixels will be visible in the output image. Refer to Bad pixel correction chapter for more detailed description of different bad pixel correction modes in Astir2 core.

Register 0x44

Value in this register determines digital video output data (and synchronization signals’) output phase relative to clock signal. For video output modes with clock running at 27 MHz (BT656, Custom-2) there are four possible options:

- 0x00: Data changes on the falling clock edge.
- 0x01: Data changes in the middle between rising and falling clock edges
- 0x02: Data changes on the rising clock edge
- 0x03: Data changes in the middle between falling and rising clock edges

For video output modes with clock running at 108 MHz (Custom-1), two options are available:

- 0x00 or 0x02: Data changes on the falling clock edge

- 0x01 or 0x03: Data changes on the rising clock edge

Register 0x45

Register is used to control graphics overlay buffer. Least significant bit (bit 0) in the register is used to enable or disable display of graphics overlay content on top of infrared image – writing 1 will enable and writing 0 will disable it. Bit 5 is virtual bit. Writing 1 will switch graphics overlay buffers. This is used to prevent any flickering during buffer update- while one graphics overlay buffer is being updated by external device, content of another buffer is used to render video output. Once buffer update has finished, buffer switch command can be issued to make new content visible.

Register 0x46

Register code contains color code (from graphics overlay color palette) used to initialize new buffer after buffer switch command. Default value is core 0xFF, which corresponds to the transparent color code.

Register 0x47

Value in this register determines half-size of the square area located at the center of the image frame, content of which is used to determine one of two possible states for invertible graphics overlay color code (code 0xFE). If given area contains more bright pixels than dark, any graphics overlay elements drawn in 0xFE color will be displayed as black and wise versa.

Register 0x48

Register is used to implement the hysteresis for the invertible 0xFE graphics overlay color code. The higher the value in the register the more stable will 0xFE color code be rendered, but on the other hand larger change in medium brightness in the center zone (defined by register 0x47) will be required for color inversion to take place.

Registers 0x4A, 0x49

In custom-1 digital video output mode thermal image can be scaled up from sensor resolution to resolution lower than the actual video output resolution (controlled with register 0xB0). If it is, register pair will store horizontal shift for the scaled up thermal image within the video output frame. Similarly, registers 0x4B and 0x49 are used to shift the image vertically. By default (with horizontal and vertical offsets set to 0) the image is located in the top left corner of the video output frame.

Registers 0x4C, 0x4B

Register pair stores vertical shift for the scaled up thermal image within the video output frame (see Registers 0x4A, 0x49).

Register 0x63

Register controls external synchronization mode. For more information refer to External synchronization mode section of this document.

Registers 0x75, 0x74

Register pair sets the contrast bias parameter, i.e. pixel value used as a reference point for contrast adjustment: pixel values greater than contrast bias will be increased after contrast multiplication and pixel values lower than contrast bias will be decreased.

Parameter is used only in brightness/contrast mode 0x04. For the picture to become visible, parameter value should be close to the average of all raw pixels' values. In other brightness/contrast mode contrast reference point is determined automatically.

Register 0x76

Register adjusts overall contrast of the image. Parameter works in all brightness/contrast mode. For brightness/contrast mode *0x0*0, 0x01 and 0x04. It is the main contrast determining parameter, while for modes 2 and 3 automatic contrast adjustment does a good job and this parameter usually has to be set to a value close to a default 64 for best result.

Register 0x77

Register adjusts overall brightness of the image. Parameter works in all brightness/contrast mode.

Register 0x78

Register sets contrast/brightness mode (BC mode) of the image. There are five different BC modes available:

- 0x00: Mean
- 0x01: Median
- 0x02: Linear
- 0x03: Advanced
- 0x04: Manual

For modes 0, 1 and 4 pixel output value is calculated using the following formula:

$$pixel_a = \left(pixel_p - contrast\_bias\right) * contrast + contrast\_bias + brightness$$

Where $pixel_a$ – adjusted pixel value,

$pixel_p$ – pixel value prior to contrast/brightness adjustment,

contrast – manually set contrast value,

brightness – manually set brightness value.

Parameter contrast_bias indicates pixel value which is going to be a reference point for contrast adjustment; i. e. pixel values greater than contrast_bias will be increased after contrast multiplication and pixel values lower than contrast_bias will be decreased.

In mean BC mode (0x00) the core automatically sets contrast bias value to the mean value of all pixels in the frame. In median BC mode (0x01) contrast bias value is automatically set to the median value of all pixels in the frame. Median mode is more suitable when the scene contains small object which have temperature much smaller or higher compared to the rest of the scene and which we don’t want to focus on. In such instances using *average* mode (mode 0) would make the rest of the scene become hardly visible, while “median” mode would help to reduce influence of these objects to the visibility of the rest of the scene.

Setting parameter to 4 places the core in fully manual mode. In this mode contrast bias is set manually. Usually this mode is not suitable for autonomous core operation because raw pixel value depends on core and lens temperature. Under normal operating conditions with slowly changing environment temperature it makes average of all pixel values drift away from the initially set contrast bias. Which in turn makes core output too bright or too dark.

Mode 2 will not only optimize brightness of the image, but the contrast as well. In this mode two additional parameters are used: maximum contrast (register 0x7C) and histogram cropping (register pair 0xD3, 0xD2). Histogram cropping parameter sets number of brightest and darkest pixels to be removed before proceeding with the adjustment. Maximum contrast parameter sets maximum allowed contrast for this mode.

Mode 3 is the most advanced picture optimization mode which usually gives the best result. Additionally to parameters used in mode 2, there are two more parameters: extra contrast and Register 0x79.

Register 0x79

Parameter sets number of blocks that the image is divided into for analysis purposes in advanced brightness/contrast mode (brightness/contrast mode set to 0x03). The higher the value the better contrast and brightness optimization can be achieved. Under some conditions (when for instance scene contains a small number of large objects with different temperatures) higher values may cause some artifacts (for instance boundaries between image blocks may become slightly visible), but it's rarely the case and in general the higher the division factor the better is visibility of the smaller objects in the scene with many different temperature objects.

Register 0x7B, 0x7A

Parameter sets signal gain level (extra contrast) when advanced brightness/contrast mode is used (brightness/contrast mode set to 0x03). When this mode is used, it is usually best to have basic *contrast* value set to around 64, and use *extra contrast* parameter to adjust image gain. Higher value means more contrast, at the expense of increased noise.

Register 0x7C

When brightness/contrast mode is set to 0x02 or 0x03, parameter determines maximum gain that can be applied to the signal prior to further processing. The effect of parameter is noticeable in low contrast scenes. For such scenes, without gain limitation signal may be amplified to a point where only sensor noise becomes visible. To prevent this, use smaller parameter values. Too small values on the other hand may limit detectability of objects in low contrast scenes.

Register 0x7D

Register controls video outputs of the core.

Least significant bit (bit 0) enables/disables analog video output. Currently only PAL analog output is supported.

Bit 1 enables/disables digital video output.

Bits 5:2 are used to select digital video output type. The following options are available:

- 0x0: BT656
- 0x1: Custom-1
- 0x2: CameraLink
- 0x3: BT656 centered
- 0x4: Custom-2

Both analog and digital video output modes can be active simultaneously.

Register 0x7F

Temporal filter parameter sets the amount of temporal filtering applied to the core's output image. Higher values filter high frequency temporal noise better, but may filter out fast moving small objects with a temperature similar to the background. Setting parameter to 0 disables temporal filtering. Values 1, 2 and 3 correspond to temporal filter of increasing intensity.

Register 0x80

Register is reserved for gamma adjustment and is not being used currently.

Register 0x82

Parameter is used to enable/disable image destriping and choose its level. Because of how infrared sensor works, there is a vertical striped pattern superimposed on the image. This pattern slowly changes with time. Vertical stripes become apparent in low contrast scenes when gain/contrast has to be increased. To reduce visibility of the stripes the core has special algorithm implemented. Setting destriping parameter to 0 disables destriping algorithm, while setting it to value greater than 0 – enables. Destriping level can be chosen up to 15. Higher value will filter vertical stripes better with a tradeoff that in some scenes, with many thin vertical objects (such as fences), destriping algorithm may introduce some additional artifacts. However the artifacts are rare and usually much less noticeable then the original stripes themselves.

Register 0x83

Register stores sharpening level for the image. Higher values mean more sharpening, but it also increase noise sharpness (as well as perceivable noise level). Sharpness is only supported when brightness/contrast mode is set to 2 or 3, otherwise image sharpening is automatically disabled.

Register 0x86

Register is used to enable/disable flipping of the output image. Depending on the register value, image can be flipped as follows:

- 0x00: standard output (no flipping),
- 0x01: image is flipped vertically,
- 0x02: image is flipped horizontally,
- 0x03: image is flipped both vertically and horizontally (equivalent to rotation by 180 degrees).

Register 0x9C

Virtual register controls automatic bad pixel detection (list mode). Writing 0x01 to the register will initiate automatic bad pixel detection procedure. Writing 0x02 to the register will erase the table containing the list of bad pixels detected during previous automatic bad pixel detection.

Register 0x9D

Register controls automatic bad pixel detection and correction (real time mode). Possible register values are described in real time bad pixel detection section of this document.

Register 0x9E

Register stores sensitivity value used by real time automatic bad pixel detection and correction algorithm. Optimal values are in the range of 0x20 to 0x30.

Register 0x9F

Register stores sensitivity value used by automatic bad pixel detection (flash mode) algorithm. Optimal values are in the range of 0x20 to 0x30.

Register 0xA0

Writing 1 to this virtual register will restart the core (reloads all calibration data, filters, palettes and user settings) from non-volatile flash memory.

Registers 0xA5, 0xA4, 0xA3, 0xA2

Four registers contain serial number of the core.

Registers 0xA9, 0xA8, 0xA37, 0xA6

Four registers contain serial number of the sensor within the core.

Register 0xB0

Register determines size of the infrared image within the digital video output frame. Possible values are:

- 0x00: infrared image will be scaled up to fill whole video output frame.
- 0x01: infrared image size will be determined by values stored in registers 0xB1 and 0xB2.
- 0x02: infrared image will be left at sensor resolution.
- 0x03: infrared image will have resolution of 640 x 480 (In case of 384 x 288 core the image will be scaled up. For 640 x 480 image will be left unchanged)
- 0x04: Infrared image will have resolution of 720 x 576
- 0x05: Infrared image will have resolution of 768 x 576

Register 0xB1

Register contains horizontal infrared image resolution when register 0xB0 is set to 0x01. Prior to writing to 0xB1 register preferred horizontal resolution value (in pixels) has to be divided by 8.

Note: the core supports only scaling thermal image up. It cannot be scaled down.

Register 0xB2

Register contains vertical infrared image resolution when register 0xB0 is set to 0x01. Prior to writing to 0xB2 register preferred horizontal resolution value has to be divided by 4.

Registers 0xD3, 0xD2

Register pair contains number of brightest and darkest pixels (histogram cropping value) to be removed before proceeding with the image analysis and adjustments in brightness/contrast mode 0x02 and 0x03. If real time bad pixel correction is not being used, this prevents new bad pixels from influencing analysis results. It also prevents noisy pixels at the sides of the global histogram from making optimization results vary too much from frame to frame. It's a good idea to have this value set to at least several hundreds. Higher values may increase the stability, but details of the small hot or cold objects may become invisible (saturated).

Register 0xD5

Register selects color palette and its polarity. Polarity is determined by the least significant bit (bit 0). If set to 0 palette will have positive polarity. If set to 0, polarity will be negative.

Rest of the bits in the register contain active palette ID number. Default grayscale palette has code 0x00.

List of palettes available are given in Palettes section.

Register 0xD6

Register controls digital zoom. Writing values 0x01, 0x02, 0x04, 0x08 will result in thermal image being magnified x1, x2, x4, x8 times respectively. Value 0x00 is equivalent to 0x01 (x1 time magnification).

Custom thermal image cropping (custom zoom) is available when value written to register 0xD6 is equal to 0xFF. In this mode the following registers have to be initialized prior to writing 0xD6 register:

- 0xD9, 0xD8: Register pair holds cropping window's top left corner's Y coordinate within the sensor frame.
- 0xDB, 0xDA: Register pair holds cropping window's height.
- 0xDD, 0xDC: Register pair holds cropping window's top left corner's X coordinate within the sensor frame.
- 0xDF, 0xDE: Register pair holds cropping window's width.

Cropping window's position and size are expressed in 1/16th of a sensor pixel – with this precision cropping window can be positioned within the sensor frame. In custom cropping mode area of the sensor frame cropped by the cropping window will be scaled up to video output resolution (determined by register 0xB0).

Continuous zoom is enabled by writing value 0xFE to register 0xD5. Prior to this, register pair 0xE3, 0xE2 has to be written.

To smoothen zoomed in image bilinear interpolation is used.

Registers 0xD9, 0xD8

Register pair stores cropping window's top left corner's Y coordinate (measured in 1/16th of a sensor pixel) within the sensor frame in custom zoom mode.

Registers 0xDB, 0xDA

Register pair stores cropping window's height (measured in 1/16th of a sensor pixel) within the sensor frame in custom zoom mode.

Registers 0xDD, 0xDC

Register pair stores cropping window's top left corner's X coordinate (measured in 1/16th of a sensor pixel) within the sensor frame in custom zoom mode.

Registers 0xDF, 0xDE

Register pair stores cropping window's width (measured in 1/16th of a sensor pixel) within the sensor frame in custom zoom mode.

Registers 0xE3, 0xE2

Register pair contains continuous zoom factor (mode enabled by writing 0xFE to register 0xD6). In continuous mode cropping window is centered in respect to sensor frame. Cropping window's top left corner's X value is determined by the register pair 0xE3, 0xE2 (measured in 1/8th of a sensor pixel). The rest of the cropping window's parameters are calculated by the core automatically to keep it centered and maintain aspect ratio (4:3).

Chapter 7. Control Interfaces

Apart from USB interface there two main methods to control the core:

- ASCII based serial control (UART/RS232 interfaces)
- Packet based control (SPI/QuadSPI interfaces)

ASCII based serial control is simpler to use. However it is slower and not all core functions are accessible (see list of available commands).

Packet based control on the other hand gives access to all core's settings and functions.

UART/RS232

UART/RS232 interface can be used to adjust core's parameters while the core is in operation.

If -R modification of AST-7D extension board is used, the core can be connected directly to the PC over RS232 interface. Otherwise (if AST-7D modification used is –U or different board is used) UART interface voltage level is 3.3V.

Interface settings are list below:

- Baud rate: 57600
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none

Control commands to the core are sent by sending command word and parameters (if specific command requires any) separated by spaces and finished by <CR> symbol (carriage return, hex code 0x0D). If the core recognizes and executes command successfully it replies with the following string:

"<LF>Done<CR><LF>",

where <LF> is line feed symbol (hex code 0x0A). Otherwise core replies with one of the following strings:

- "<LF>Err: Command not recognized<CR><LF>"
- "<LF>Err: Command too long<CR><LF>"
- "<LF>Err: Invalid parameter(s) <CR><LF>"
- "<LF>Err: Too many parameters<CR><LF>"

First reply is sent if command word was not recognized. Second reply is sent if too many symbols have been sent to the core without <CR> symbol (indicating end of command). If parameters are out of range for a given command, third reply will be sent. Fourth reply is sent if number of parameters does not match the number expected for a specific command.

List of supported UART/RS232 interface commands is given in Table 8. Commands to the core have to be sent without quotation marks. "<p1>" denotes first parameter, "<p2>" denotes second parameter, etc. All parameter values are in decimal form. Any of the UART/RS232 commands is equivalent to writing one or more core registers. Registers that each of the commands modifies are listed below the table.

Table 8. List of supported RS232 interface commands

Command word	No of params	Equivalent registers operation
"config save<CR>"	0	Writing 0x01 to register 0x08
"restart<CR>"	0	Writing 0x01 to register 0xA0
"bc mode <p1> <CR>"	1	Writing <p1> to register 0x78
"contrast bias <p1> <CR>"	1	Writing <p1> to register pair 0x75, 0x74
"contrast <p1> <CR>"	1	Writing <p1> to register 0x76
"bright <p1> <CR>"	1	Writing <p1> to register 0x77
"noise filter <p1> <CR>"	1	Writing <p1> to register 0x7F
"video output <p1> <CR>"	1	Writing <p1> to register 0x7D
"gamma <p1> <CR>"	1	Writing <p1> to register 0x80 (not implemented)
"min window <p1> <CR>"	1	Writing <p1> to register 0x7C
"palette <p1> <CR>"	1	Writing <p1> to register 0xD5
"zoom <p1> <CR>"	1	Writing <p1> to register 0xD6
"dismiss levels <p1> <CR>"	1	Writing <p1> to register pair 0xD3, 0xD2
"division <p1> <CR>"	1	Writing <p1> to register 0x79
"extra contrast <p1> <CR>"	1	Writing <p1> to register pair 0x7B, 0x7A
"sharpen <p1> <CR>"	1	Writing <p1> to register 0x83
"destripe <p1> <CR>"	1	Writing <p1> to register 0x82
"flip <p1> <CR>"	1	Writing <p1> to register 0x86

Command word	No of params	Equivalent registers operation
"shift <p1> <p2> <p3> <p4><CR>"	4	Reserved
"sync <p1> <CR>"	1	Writing <p1> to register 0x63
"rw <p1> <p2> <CR>"		Writing <p2> to register at address <p1>
"azoom <p1> <p2> <p3> <p4> <CR>"		Writing <p1> to register pair 0xD9, 0xD8; <p2> to register pair, 0xDB, 0xDA; <p3> to register pair 0xDD, 0xDC; <p4> to register pair 0xDF, 0xDE; and 0xFF to register 0xD6
"czoom <p1> <p2> <CR>"	2	Writing <p1> to register pair 0xE3, 0xE2 and 0xFE to register 0xD6
"abp detect <p1> <p2> <CR>"		Writing <p1> to register 0x9F and 0x01 to register 0x9C
"abp erase <p1> <p2> <CR>"		Writing 0x02 to register 0x9C
"abp mode <p1> <p2> <CR>"		Writing <p1> to register 0x9E and <p2> to register 0x9D
"read info<CR>"	0	Core responds with a string indicating sensor type and resolution
"read status<CR>"	0	Reserved
"read config<CR>"	0	Core responds with the current settings instead of "<LF>Done<CR><LF>" acknowledgment.

"read config" command

Upon reception of "read config" command over UART or RS232 interface the core will respond with current settings (values of the appropriate core registers). Table below shows the structure of core's response. Column Offset contains offset in the response packet.

Table 9. Structure of the core response to "read config" command

Offset	Size	Register address	Notes
0x00	0x01		Always 0x00
0x01	0x03		Always 0x000000
0x04	0x04		Always 0x12345678
0x08	0x01	0x78	BC mode parameter
0x09	0x01	0x76	Contrast
0x0A	0x01	0x77	Brightness
0x0B	0x02	0x75, 0x74	Contrast bias
0x0D	0x01	0x7F	Temporal filter

Offset	Size	Register address	Notes
0x0E	0x01	0x7D	Video output
0x0F	0x01	0x80	Reserved for gamma
0x10	0x01	0x7C	Maximum gain
0x11	0x01	0xD5	Palette
0x12	0x01	0xD6	Zoom factor
0x13	0x02	0xD3, 0xD2	Histogram cropping
0x15	0x01	0x79	AGC blocks
0x16	0x02	0x7B, 0x7A	Extra contrast
0x18	0x01	0x83	Sharpening
0x19	0x01	0x82	Destriping
0x1A	0x01	0x86	Image flip
0x1B	0x01		Reserved
0x1C	0x01		Reserved
0x1D	0x01		Reserved
0x1E	0x01		Reserved
0x1F	0x01	0x63	External sync. settings
0x20	0x01		Always 0x13
0x21	0x01		Always 0x10

SPI

Communication over SPI (as well as QuadSPI) interfaces is packet based. This chapter describes physical layer of the interface. For information regarding transfer protocol refer to Chapter 8. The core acts as a slave SPI device; the clock has to be supplied by external device.

SPI interface supports two modes: 8bit and 32bit. The mode is determined by the four first bytes in the header of request packet. Every request packet starts with four ASCII characters: ‘T’ (0x54), ‘U’ (0x55), ‘S’ (0x53), ‘B’ (0x42). If these four characters are received in the following order: ‘TUSB’ (see Figure 16), SPI interface for the current transaction (remaining part of the request packet, and full response packet) will operate in 8 bit mode. In 8bit mode data is sent byte by byte, most significant bit first. If on the other hand initial four characters are received in the following order: ‘BSUT’ (see Figure 17), the interface will operate in 32 bit mode. In this mode data stream is divided into 32 bit words and sent most significant bit first.

SPI mode for response packet will be kept the same as it was for request packet (Figure 18 and Figure 19 show the waveform when reading four bytes 0x12, 0x34, 0x56 and 0x78 in 8bit and 32bit SPI modes respectively). First data bit has to be read at the time of first clock rising edge. Next data bit to be read from the core will be ready in 4 to 16 ns after the last rising edge.

Timeout value for the request/response packet exchange is 10 ms. Timeout counter within the core is suspended during the command execution phase (in between the request and response packets). This is done to allow longer commands (for instance writing to core’s non-volatile flash memory) to be executed without timeout occurring.

Additionally 32 bits of data over SPI interface (regardless of 8bit or 32bit interface mode is used) to the core have to be sent (or read) in less than 10 µs. This requirement limits lowest possible SPI clock frequency to 3.2 MHz. Maximum allowed frequency for SPI interface is 22.5MHz.

When writing data to core (sending request packet), the following requirements have to be met (Figure 20): SETUPmin time should be 2 ns or more; HOLDmin time should be 10 ns or more.

When data is being read from the core (receiving response packet, Figure 21), HOLDmin time is equal to 4 ns, HOLDmax time is equal to 16 ns.

All SPI signals have the following levels: VILmin=-0.3V, VILmax=0.8 V, VIHmin=2.0 V, VIHmax=3.45 V, VOLmax = 0.400 V, VOHmax = 2.4 V.

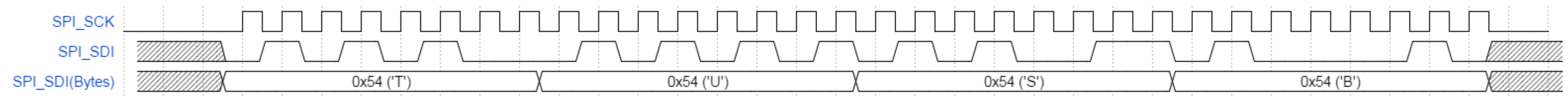


Figure 16. SPI interface / 8bit / write

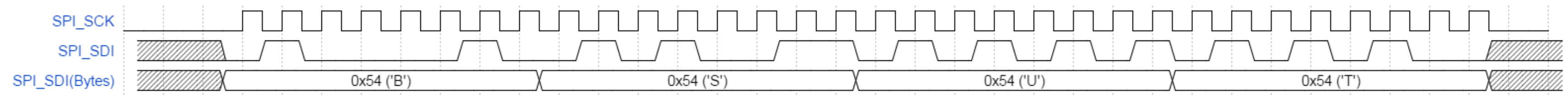


Figure 17. SPI interface / 32bit / write

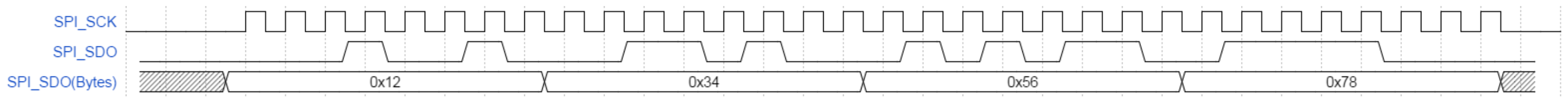


Figure 18. SPI interface / 8bit / read

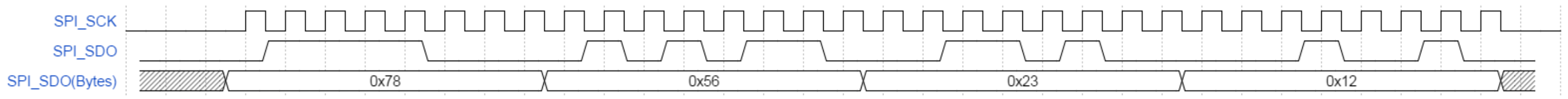


Figure 19. SPI interface / 32bit / read

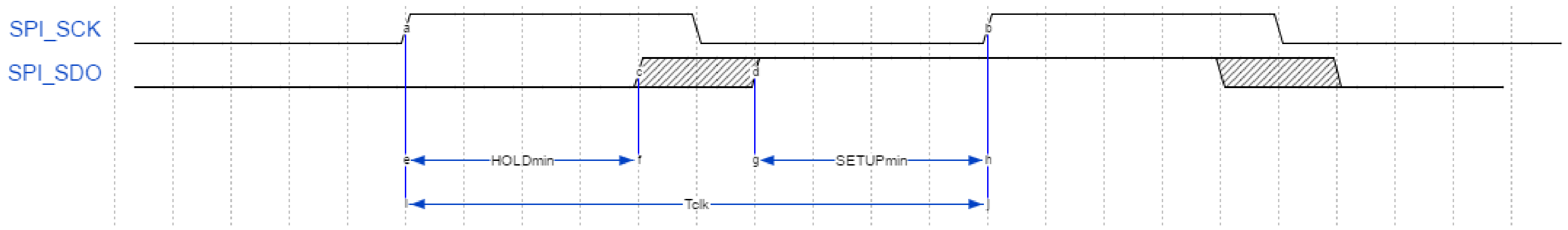


Figure 20. SPI bit timing (sending data to core)

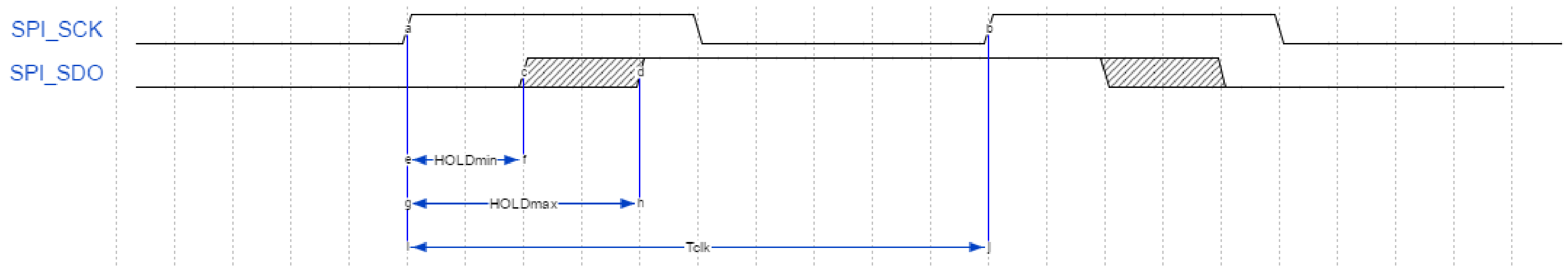


Figure 21.Bit timing (reading data from core)

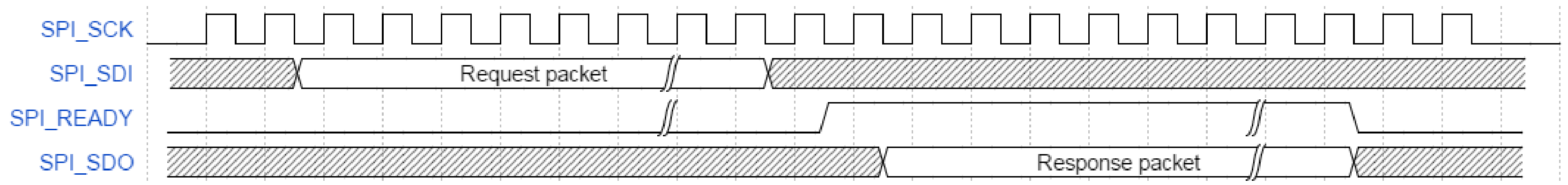


Figure 22. Request/response packet exchange over SPI

QuadSPI

Communcation over QuadSPI (as well as SPI) interfaces is packet based. This chapter describes physical layer of the interface. For information regarding transfer protocol refer to Chapter 8. The core acts as a slave QuadSPI device; the clock has to be supplied by external device.

To communicate with the core over QuadSPI interface 4 bidirectional data lines, clock and ready signal (to indicate when the response packet) are used. Over four data lines data is transferred four bits per clock cycle. Data is transferred most significant nibble (half-byte) first.

When response data packet is being received from the core, first nibble of data is to be read at the time of first rising clock edge. Next nibble of data will be ready in 4 to 16 ns after the last rising clock edge.

Timeout value for the request/response packet exchange is 10 ms. Timeout counter within the core is suspended during the command execution phase (in between the request and response packets). This is done to allow longer commands (for instance writing to core’s non-volatile flash memory) to be executed without timeout occurring.

Additionally 32 bits of data over QuadSPI interface to the core have to be sent (or read) in less than 10 µs. This requirement limits lowest possible QuadSPI clock frequency to 3.2 MHz. Maximum allowed frequency for SPI interface is 45 MHz.

When writing data to core (sending request packet), the following requirements have to be met (Figure 25): SETUPmin time should be 2 ns or more; HOLDmin time should be 10 ns or more.

When data is being read from the core (receiving response packet, Figure 26), HOLDmin time is equal to 4 ns, HOLDmax time is equal to 16 ns.

All SPI signals have the following levels: VILmin=-0.3V, VILmax=0.8 V, VIHmin=2.0 V, VIHmax=3.45 V, VOLmax = 0.400 V, VOHmax = 2.4 V.

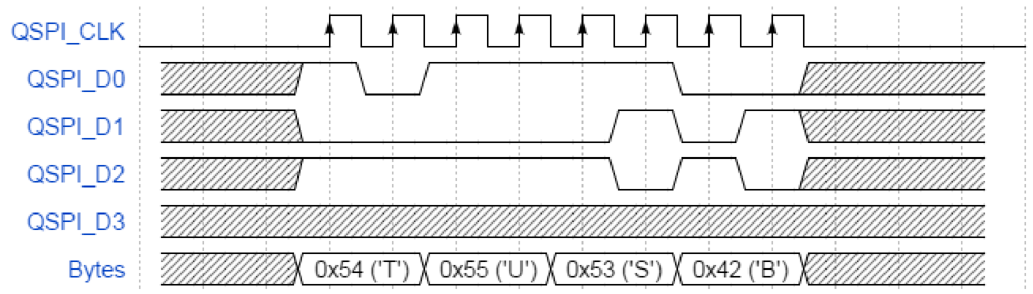


Figure 23. QuadSpi write timing

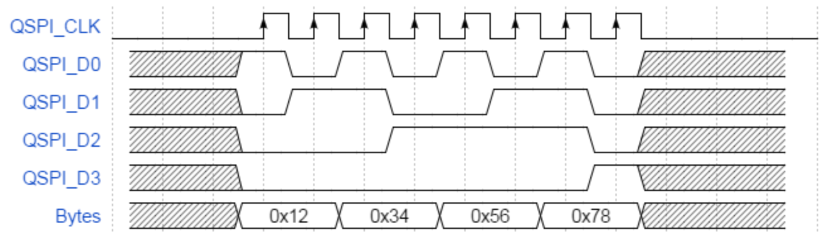


Figure 24. QuadSpi read timing

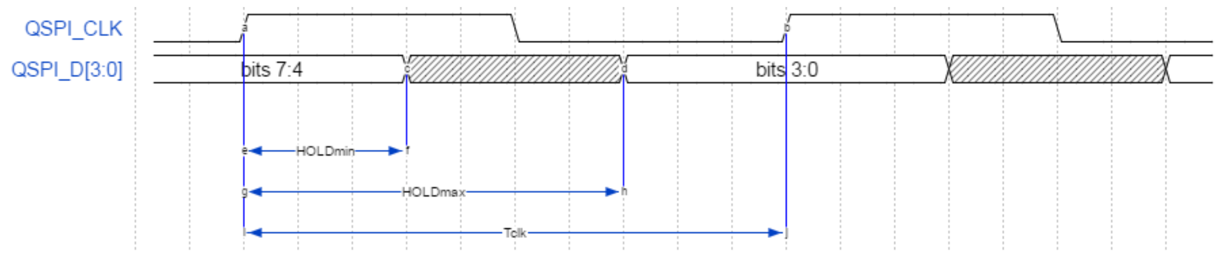


Figure 25. QuadSpi bit timing write

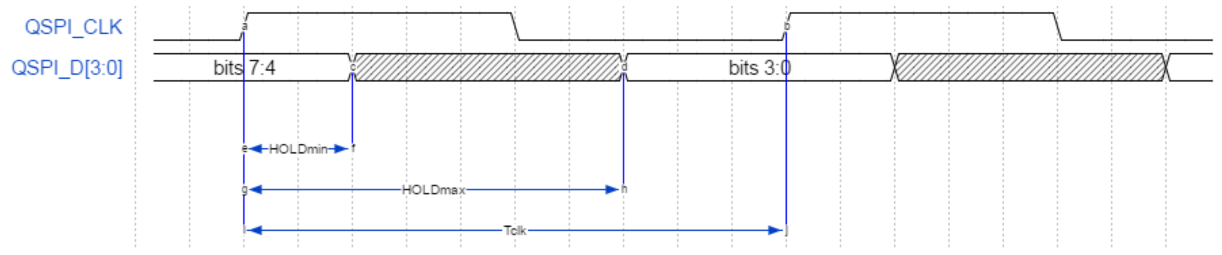


Figure 26. QuadSpi bit timing read

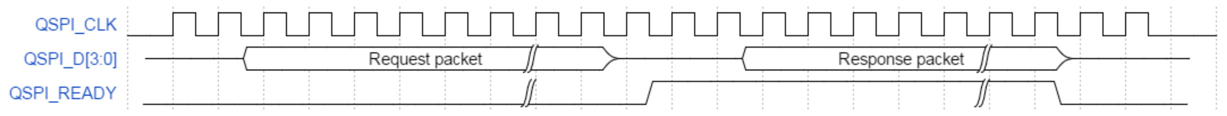


Figure 27. Request response packet exchange over QuadSPI

USB

USB interface is available only when using AST-7D board. It is the simplest and fastest way to get started with Astir2 core. Once the Astir2 core is connected to the PC, CoreConfig PC Software can be used to access all core's settings and features. Alternatively, the core can be connected to the PC using RS232 interface, however not all features are then accessible in CoreConfig (among inaccessible features is manual bad pixel correction, snapshots, etc.)

Chapter 8. Packet based communication

In packet based communication mode core communicates with the external master device over SPI or QuadSPI interface by exchanging packets. Each transaction is always initiated by external master device by sending a request packet. Upon successful reception of the request packet, the core responds with response packet.

Packet structure

Both request and response packets consist of two parts – packet header and packet body.

Request packet header has a fixed size of 64 bytes. Request packet body can have a size of 1024 or 0 bytes (determined by the value at 0x12 byte in request packet header). Response packet header can have a size of 64, 16, 4 or 0 bytes (determined by the value at 0x12 byte in request packet header). Response packet body can have a size of 1024 or 0 bytes. Figure 28 depicts single transaction (request / response packet exchange between master and slave).

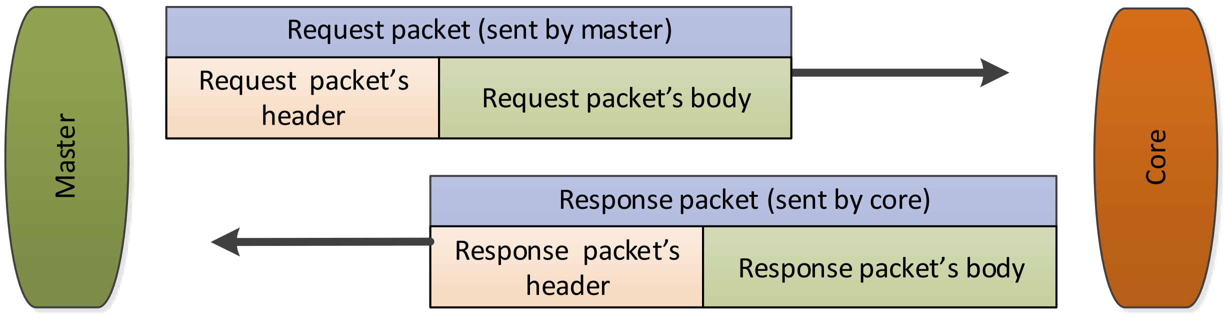


Figure 28. Request / response packet exchange between master and slave

Request packet header

All request packets start with a header. Request header size is fixed to 64 bytes. Header structure is given in Table 10. Some fields of the header are reserved for future use. All bytes belonging to the reserved data fields must be set to 0x00.

Table 10. Request packet header structure

Field offset	Field size	Field name	Field description
0x00	0x04	Title	Always: 0x54, 055, 0x53, 0x42 (four ASCII characters: 'TUSB')
0x04	0x04	Checksum	Contains the sum of packet's body words (data in packet body is interpreted as 16-bit words)
0x08	0x02	Command	Contains command code. Command code can have one of four valid values (given in Table 10). It determines the action the core will take upon reception of data packet and type of response packet it will respond with.
0x0A	0x02	Data size	Indicates size of valid data in packet's body. Packet body size can have a size of 0 or 1024 bytes. If it is non-zero, valid data size may vary, even though packet body size is fixed to 1024 bytes.
0x0C	0x02	Memory select	For memory writing or memory reading commands indicates which core's memory will be accessed. List of available memories is given in Table 14.
0x0E	0x04	Memory offset	Memory offset in bytes for memory writing and memory reading commands.
0x12	0x01	Packet sizes	Bits [1:0] must be set to 0x00 Bits [3:2] indicate request body size: 0x0 : request packet's body has a size of 1024 bytes 0x3 : request packet's body has a size of 0 bytes Bits [5:4] indicate response header size: 0x0 : response packet's header has a size of 64 bytes 0x1 : response packet's header has a size of 16 bytes 0x2 : response packet's header has a size of 4 bytes 0x3 : response packet's header has a size of 0 bytes Bits [7:6] indicate response body size: 0x0 : response packet's body will have a size of 1024 bytes for memory/register read commands and a size of 0 bytes for memory/register write commands. 0x1 : response packet's body will have a fixed size of 1024 bytes 0x3 : response packet's body will have a fixed size of 0 bytes
0x13	0x01	Reserved	Must be set to 0x00.

Field offset	Field size	Field name	Field description
0x14	0x02	Window X	Window's upper left corner X coordinate (multiple of 8). The field is only used for windowed memory writing operations.
0x16	0x02	Window Y	Window's upper left corner Y coordinate (multiple of 1). The field is only used for windowed memory writing operations.
0x18	0x02	Window width	Window width in bytes (multiple of 8). The field is only used for windowed memory writing operations.
0x1A	0x02	Frame width	Frame width in bytes (multiple of 8). The field is only used for windowed memory writing operations.
0x1C	0x01	Window mode	Windowed memory writing mode.
0x1D	0x01	Reserved	All bytes in this field must be set to 0x00.
0x1E	0x01	Reserved	Will be used to indicate packet destination. Must be set to 0x00.
0x1F	0x01	Reserved	Will be used for request event synchronization. Must be set to 0x00.
0x20	0x20	Reserved	All bytes in this field must be set to 0x00.

Valid command codes are given in Table 11.

Table 11. Valid command codes used in request packet's header

Command code	Transaction purpose
0x00	Read core's registers
0x01	Read core's memory
0x02	Write core's registers
0x03	Write core's memory

Request packet body

Content of the request packet's body depends on the *command* field in request packet's header.

For register read command, bytes at even addresses in request packet's body contain addresses of core's registers to be read from, while bytes at odd addresses in request packet's body are ignored. For instance, to read the content of three core's registers at addresses 0x05, 0x20 and 0xAB, request packet's body will look like this: [0x05, 0x00, 0x20, 0x00, 0xAB, 0x00, 0x00, 0x00, …]. With request packet's body size of 1024 bytes, maximum number of registers read during a single transaction is 512.

For register write command, bytes at even addresses in request packet's body contain addresses of core's registers to be written to, while bytes at odd addresses in request packet's body contain data to be written. For instance, to write value 0x12 to register at address 0x34 and value 0x56 to register at address 0x78, request packet's body will look like this: [0x34, 0x12, 0x78, 0x56, 0x00, 0x00 …]. With request packet's body size of 1024 bytes, maximum number of register written during a single transaction can be 512.

For memory write command, request packet's body contains data to be written to one of core's memories.

For memory read command, request packet's body data is ignored.

Response packet header

Response packet header contains status code, indicating whether request packet has been successfully received and command executed. Response code is located at the first byte of response packet's header. Meaning of the status code is given in the Table 12. Meaning of different status code values. below.

Table 12. Meaning of different status code values.

Status code	Status code meaning
0x00	Command execution successful
0x01	Wrong request packet's header title (bytes 0x00 – 0x03 in request packet's header)
0x02	Wrong checksum in request packet's header
0x03	Unrecognized command (*command* code field in request packet's header).

Response packet body

For read core's registers command, bytes at even addresses in the response packet body contain value of the registers which addresses where transferred to the core with request packet containing read core's registers command.

For read core's memory command, request packet's body contains data from the core's memory.

Read core's registers command

Following fields are used in request packet's header when core's register reading is intended:

- *Title*
- *Checksum*
- *Command*
- *Data size*
- Packet sizes

Values in other fields are ignored, except for reserved fields which must have all bytes set to 0x00.

Command field has to be set to 0x00, while *data size* field contains number of registers to be read.

Write core's registers command

Following fields are used in request packet's header when core's register writing is intended:

- *Title*
- *Checksum*
- *Command*
- *Data size*
- Packet sizes

Values in other fields are ignored, except for reserved fields which must have all bytes set to 0x00.

Command field has to be set to 0x02, while *data size* field contains number of registers to be written.

Read core’s memory command

Following fields are used in request packet's header when core’s memory reading is intended:

- *Title*
- *Checksum*
- *Command*
- *Data size*
- Packet sizes
- *Memory select*
- *Memory offset*
- *Window mode*

Values in other fields are ignored, except for reserved fields which must have all bytes set to 0x00.

Command field has to be set to 0x01. *Data size* field indicates number of bytes to be read from the memory. Possible values for *memory select* field are given in Table 14. *Memory offset* field contains address we want memory reading to start at. Memory reading in windowed mode is not supported and *window mode* field must be set to 0x00.

Write core’s memory command

Following fields are used in request packet's header when core’s register reading is intended:

- *Title*
- *Checksum*
- *Command*
- *Data size*
- *Memory select*
- *Memory offset* (for some *memory select* field values, memory offset is chosen by the core automatically and value in the *memory offset* field is ignored)
- *Window mode*

Values in other fields are ignored, except for reserved fields which must have all bytes set to 0x00.

Command field has to be set to 0x03. *Data size* field indicates number of bytes to written to the memory (has to be a multiple of 8). Possible values for *memory select* field are given in Table 13. *Memory offset* field contains address we want memory writing to start at.

If data is written to flash memory (indicated as “nonvolatile flash” memory in Table 13) at least 30ms have to pass between two 1024 byte write operations.

If windowed memory writing is to be used (*window mode* field value greater than 0) the following additional fields are used:

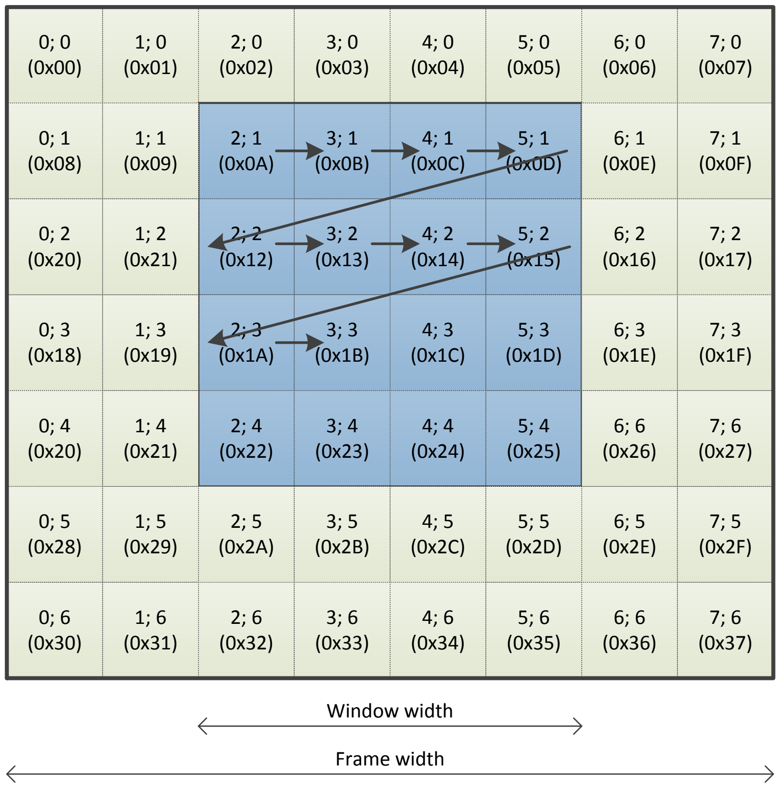
- *Window X*
- *Window Y*
- *Window width*
- *Frame width* (for some *memory select* values frame width is chosen by the core automatically and value in the *frame width* field is ignored)

If *windowed mode* field is set to 1, new virtual window will be created within the selected memory and data will be written to the memory starting at the beginning of the window. If *windowed mode* field is set to 2, data will be written further to the previously created window.

Windowed memory writing is visualized in **Error! Reference source not found.**. Continuous region of memory starting at address 0x00 is v isualized as two dimensional 8 bytes wide array (array width corresponds to *frame width* field in request packet’s header). Within the two dimensional memory array virtual window is created at location with two dimensional coordinates (2;1) (in request packet's header *window X* field set to 2 and *window Y* field set to 1). Window width is set to 4 bytes (*window width* field in request packet's header). In the figure, each memory cell contains its absolute address within memory (in parenthesis) as well as its two dimensional coordinates within arbitrary chosen two dimensional memory array. Data written to the memory using windowed mode with parameters described above will be written to the memory locations colored in blue.

Windowed memory writing is particularly useful when using graphics overlay functionality.

Important: care must be taken during memory write operations not to go over the allowed memory region limits as this will corrupt other parts of core's memory. However, unless writing to flash memory, core's operation after memory corruption can be fully restored by re-powering it.



Nonvolatile flash memory sector erase

Before core's nonvolatile flash memory can be written with new content, sector erase operation has to be performed (however it does not have to be performed if write operation only involves changing flash memory bits from '1' to '0' – this might be the case when new bad pixel is to be marked in bad pixel map for instance). Flash sector size is 65536 bytes.

To perform flash sector erase, the following core's registers (in the exact given order) have to be written to:

Table 13. Core's nonvolatile flash sector erase procedure

Register address	Register data	Explanation
0x25	0x03	Switch to manual flash memory control
0x27	0x00	Deselect flash chip
0x27	0x01	Select flash chip
0x26	0x06	Set write enable
0x27	0x00	Deselect flash chip
0x27	0x01	Select flash chip

Register address	Register data	Explanation
0x26	0xD8	Sector erase command
0x26	Address2	Third (most significant) byte of the flash sector address
0x26	Address1	Second byte of the flash sector address
0x26	Address0	First (least significant) byte of the flash sector address
0x27	0x00	Deselect flash chip
0x25	0x00	Switch to default flash memory control

After flash sector erase command has been issued sector is guaranteed to be erased in 3 seconds – no other operation requiring flash access can take place during this time. At least one millisecond delay has to be made between each register write operation for the core to perform serial transaction to flash memory. It means that registers have to be written to the core using separate request packets (one register per packet).

Core’s memories

Table below lists all available memories to be used with read core’s memory and write core's memory command.

Table 14. List of core’s memories

Memory select	Memory description	Memory type
0x01	Reserved	N/A
0x02	Reserved	N/A
0x03	Reserved	N/A
0x04	Reserved	N/A
0x05	Graphics overlay hidden buffer	LPDDR2 memory
0x06	Reserved	N/A
0x07	Reserved	N/A
0x08	Reserved	N/A
0x09	Reserved	N/A
0x0A	Reserved	N/A
0x0B	Reserved	N/A
0x0C	Reserved	N/A

Chapter 9. Bad pixels

During the operation new bad pixels may appear; especially if the core experiences shocks or impacts. To mask bad pixels out the core uses two methods:

- List based. Information about bad pixels is stored in the bad pixel list. During the operation all pixels in the list are replaced with interpolated information from the good neighboring pixels. There are three bad pixel lists in the core:
    - Factory list. The list stored in core's flash memory during the manufacture/calibration. It cannot be modified by the user.
    - User list. The list can be modified by the user.
    - Auto list. The list will be automatically filled by the core once the appropriate command is sent.
- Real time. During the operation core will determine pixels that are likely to be bad and fix them in real time.

Information about bad pixel positions in the three lists is used during operation simultaneously. One or more of the lists cannot be selectively disabled.

Real time pixel detection can be disabled or enabled.

Bad pixels (those that are marked as bad in one of the three lists or those determined to be bad using real time detection algorithms) are corrected by taking an average value of up to 8 neighboring good pixels for each bad pixel. In a situation, where at least 9 bad pixels form a 3 x 3 square, center pixel is approximate using different methods.

User bad pixel list

The list can be modified by the user either by using CoreConfig software (when the core is connected to the PC over USB using AST-7D board) or by directly modifying user bad list in the flash memory of the core (which can be done over SPI or QuadSPI interfaces).

Auto bad pixel list

It is recommended to cover the lens while the command is being executed. Automatic bad pixel detection process will take up to 2 seconds.

The command can be invoked by writing 0x01 to register 0x9C or by sending "*abp detect <sensitivity>*" command over UART. If command is invoked by writing to register 0x9C, sensitivity value has to be written to register 0x9F first.

Sensitivity value will determine how sensitive automatic bad pixel search is. Lower value means more pixels are marked as bad. It will decrease the probability of bad pixels being missed. On the other hand too low value will result in good pixels marked as bad which at the extremely low values will result in the blurred final image. Optimal sensitivity values are in the range 20 to 30.

Auto bad pixel list can be erased by writing 0x02 to the register 0x9C or by sending "*abp erase*" command over UART.

Automatic bad pixel detection and deletion can also be activated from the CoreConfig software's Bad Pixel Correction window (buttons "*Auto BP detect*" and "*Auto BP erase*"). It is equivalent to sending to the core the two commands described above.

The core will detect groups of bad pixels as long as group size is not greater than 4.

Real time BP detection

The core can recognized and fix bad pixels during the operation in real time. The mode is controlled by the value stored in register 0x9D. Valid values are:

- 0x00 – real time pixel detection is off
- 0x01 – bad pixels detected in real time are marked as black.
- 0x02 – bad pixels detected in real time are marked as white.
- 0x03 – bad pixels detected in real time are blinking (very fast).
- 0x04 – bad pixels detected in real time are blinking (fast).
- 0x05 – bad pixels detected in real time are blinking (at medium rate).
- 0x06 – bad pixels detected in real time are blinking (slowly).
- 0x07 – bad pixels detected in real time mode are automatically fixed.

Note: For values 0x01 – 0x06 to work image sharpening has to be disabled (set to 0).

Sensitivity of the automatic bad pixel detection is determined by the value written to register 0x9E. Effect of the sensitivity parameter is described in the Auto bad pixel list section.

Real time bad pixel detection and correction can be controlled using UART command "*abp mode <sensitivity> <mode>*".

Real time bad pixel detection sensitivity and mode can also be set using CoreConfig software ("Advanced" tab).

Chapter 10. Palettes

Both digital and analog video outputs can output color image. Thermal image is colored using one of the 26 available palettes, selectable via 0xD5 register. Figure 30 shows the list of available palettes.

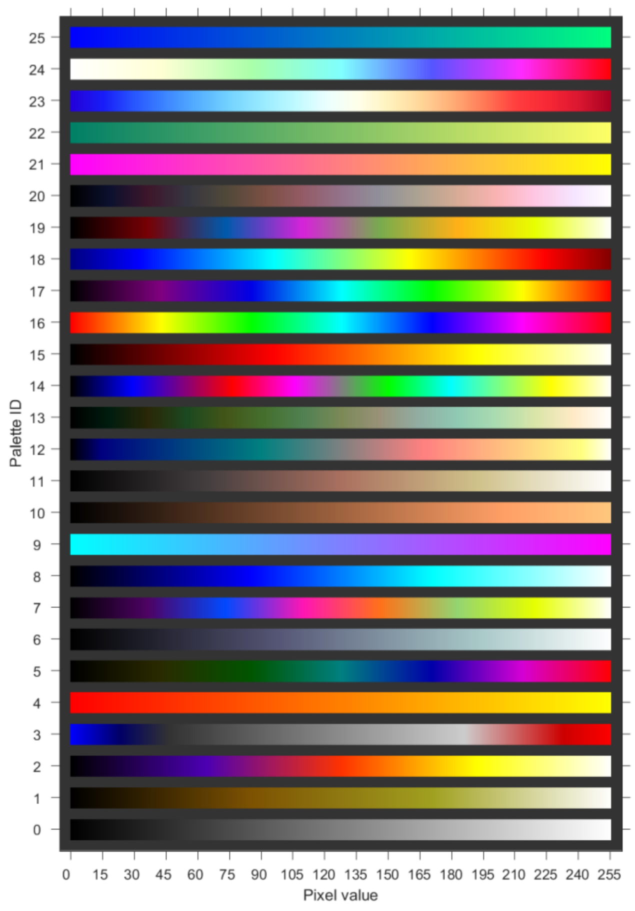


Figure 29. List of available palettes for thermal image

Graphics overlay image uses single palette shown in the figure below. 0xFF is transparent color. 0xFE is invertible graphics overlay color code. Color codes 0xE0 to 0xFD are reserved.

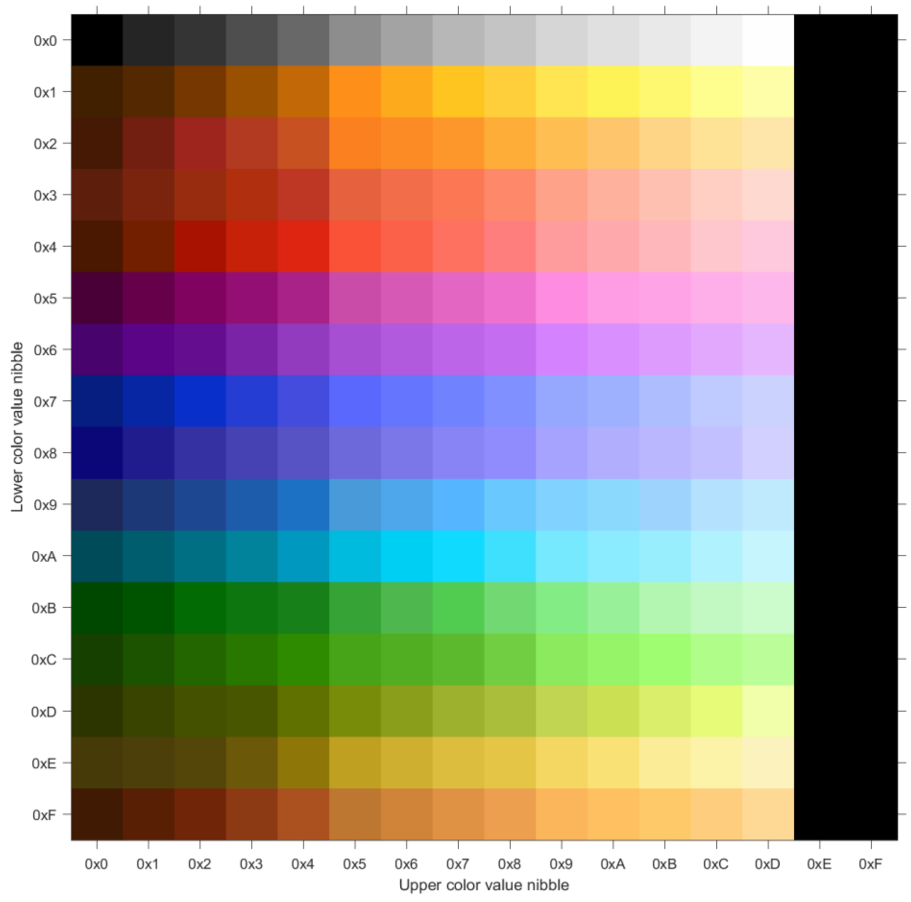


Figure 30. Palette used for graphics overlay

Chapter 11. Graphics overlay

In order to write to graphics overlay buffer, memory select field value in the request packet’s header (sent over SPI or QuadSPI interfaces) has to be set to 0x05. When graphics overlay buffer is selected, memory offset field in packet header is ignored and frame width is set automatically inside the core to appropriate value (640 for PAL video output and 720 for BT.656 video output). Data in graphics overlay buffer is superimposed on the infrared scene. Single byte in graphics overlay buffer represents single pixel.

Pixels with value 0xFF in graphics overlay buffer are transparent (infrared scene content is visible at those locations).

Pixels with value 0xFE in graphics overlay buffer change color depending on the infrared scene content at the center of the image. Size of the square window at the center of the image used to determine what color all 0xFE pixels will be displayed in, is defined by the register 0x47 (each side of the square window is twice the value written to register 0x47). If number of pixels with brightness above medium within the window is greater than the number of pixels with brightness below medium plus register 0x48 value, color for all 0xFE pixels is switched to black. If number of pixels with brightness above medium within the window is less than the number of pixels with brightness below medium minus register 0x48 value, color for all 0xFE pixels is switched to white. If difference between number of pixels with greater brightness and number of pixels with lower brightness is within the [– ‘value in register *0x48’; +* ‘value in register *0x48’]* interval, color for 0xFE pixels is not changed. This assures color is not inverted too frequently when number of dark and bright pixels is about equal.

Pixel values 0xE0 – 0xFD are reserved.

The rest of graphics overlay pixel values (0x00 – 0xDF) represent graphics overlay content. For PAL video output signal 0x00 represents black overlaid pixel, while 0xDF represent white overlaid pixel.

Chapter 12. Miscellaneous features

External synchronization mode

This mode only works 640 x 480 resolution Astir2 core.

Mode is controlled with register 0x63.

Setting the least significant bit (bit 0) to 1 will enable external sync mode. In this mode Astir2 core will wait for external synchronization event. After this event has been received the core will read four frames from the sensor. First three frames will be disregarded, because after a break of undetermined duration sensor needs to output at least a few frames before it recovers to its fully working condition. Fourth frame will be used as an output from the core.

Bit 1 in the 0x63 register is used to select between continuous and single frame video output in external synchronization mode. Setting this bit to 1 will only output single frame after synchronization event. If the bit is set to 0, the core will continuously output the same frame after the synchronization event until the next event occurs.

Synchronization event can be hardware or software. Hardware synchronization event is positive pulse on the external sync. input for at least 50 ns. Software event is generated by setting bit 7 (virtual bit) in 0x63 register (bit 0 has to be kept equal to 1, otherwise external sync mode will be disabled).

Firmware update

Core’s firmware can be updated with newer versions (if available) to add new features. Firmware update procedure takes around 2 minutes. For update to be successful it is important not to interrupt it. In case of power, connection or any other failure main firmware will get corrupt.

However the core will still be able to start up using fail-safe copy of factory programmed firmware. This copy of fail-safe firmware cannot be modified by user. Fail-safe firmware is fully functional firmware matching the one that the core was programmed in factory with. It will take several seconds longer to power up if main firmware is corrupt. After power up, update procedure can be repeated.

Firmware can only be updated using AST-7D board and CoreConfig PC Software. The procedure is described in Core Flash tab section.

Capturing snapshots

The core is able to capture and store in its volatile memory single frame. Frame size matches sensor resolution. Stored frame has all the image processing applied except for zoom and flip functions. Stored frame can be downloaded to a PC and saved to a file.

Snapshot capture only works using AST-7D board and CoreConfig PC Software. The procedure is described in Core Status section.

Gain calibration

Gain calibration is used to determine gain values for each individual sensor pixel. Gain calibration should be performed whenever the core is to be used with another lens than that it arrived from the factory with. To perform gain calibration two black bodies are required. Typically one at room temperature (cold black body) and second one at 20 or 30 degrees Celsius higher temperature (hot black body). Before performing gain calibration make sure sensor surface is clean, there are no dust or any other particles on it. Lens should be focused to infinity.

Gain calibration can be performed using AST-7D board and CoreConfig PC Software. The procedure is described in Core Flash tab section.

Saving configuration

Save configuration command will store core parameters' values to flash memory. It will take up to 3.5 seconds to complete and should not be interrupted. In case of power supply interruption during the saving operation, the core might have its parameters set to unknown values after next power up. To fix this, set all the parameters to desired values and save configuration again.

Command can be issued by writing 0x01 to register 0x08 or over UART interface (command "*config save<CR>*").

The list containing all settings (register values) which are stored in non-volatile flash memory and loaded after power up or reinitialization event can be found in the section describing reinitialization of the core.

Reinitialization

Reinitialization (restart command) forces the core to reinitialize with the parameters stored in flash memory. After restarting, core has the same settings as if it was just powered up. All unsaved modifications to any of the core's parameters will be lost.

Command can be issued by writing 0x01 to register 0xA0 or over UART interface (command "*restart<CR>*").

The following list contains all settings (register values) which are stored in non-volatile flash memory and loaded after power up or reinitialization event:

- register 0x78 (BC mode),
- register 0x76 (contrast),
- register 0x77 (brightness),
- registers 0x75, 0x74 (contrast bias),
- register 0x7F (temporal filter),
- register 0x7D (video output),
- register 0x80 (gamma – not used),
- register 0x7C (maximum gain),
- register 0xD5 (palette),
- register 0xD6 (zoom),
- registers 0xD3, 0xD2 (extreme pixel removal),
- register 0x79 (division factor),
- register 0x7B, 0x7A (signal gain level),
- register 0x83 (sharpening level),
- register 0x82 (destriping level),
- register 0x86 (image flip),
- register 0x63 (external sync),
- register 0xB0 (output resolution mode),
- register 0xB1 (output frame width),
- register 0xB2 (output frame height),
- register 0x44 (digital output phase),
- register 0x9D (real time bad pixel detection mode).

Chapter 7. CoreConfig PC Software

Prerequisites

CoreConfig PC software requires the following two packages to be installed on the system:

- .Net Framework 4.5.2 for x86 system (can be downloaded from https://www.microsoft.com/en-us/download/details.aspx?id=42642)
- C++ 2015 Redistributable package for x86 system (https://www.microsoft.com/en-us/download/details.aspx?id=48145)

Installation

CoreConfig software does not require installation. To start the software run CoreConfig.exe file in CoreConfig folder.

If USB interface available on the AST-7D board is to be used, connect AST-7D board to the PC and install USB drivers. Drivers are not digitally signed. On Windows 7/8/10 systems digital driver signature enforcement might have to be temporarily disabled during the driver installation process.

Connecting to the core

After CoreConfig software is started connection with the core can be established and core can be configured. First, select connection method to use in *Connect using* drop box. If the core is connected over USB (using AST-7D board), select *TUSB board*. Drop box below will list all the cores found connected to a PC using AST-7D board(s). Select the one you wish to configure (first one is selected automatically by default). If PC's RS232 interface is going to be used, select *COM port* from *Connect using* drop box. Drop box below will list all available COM ports on the system. Select the one that has Astri2 core connected.

After successful connection (using either TUSB or RS232 interface) core information will be displayed under Info tab page and all other tab pages will become enabled.

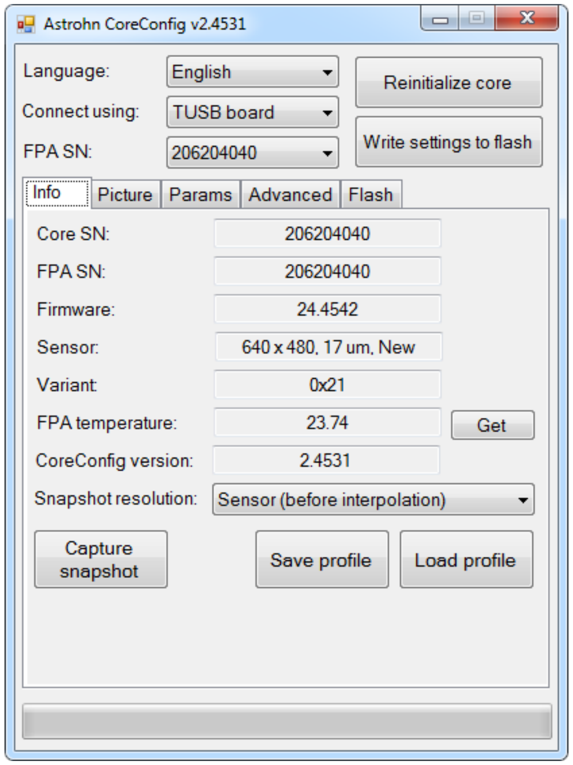
Core Status

After successful connection, *Info* tab (Figure 32) of the software will show core information: core serial number, sensor serial number, firmware version and sensor temperature. Sensor temperature can be updated by clicking *Get* button close the temperature field.

Button *Capture snapshot* allows Capturing snapshots (one frame) of what core sees at a given moment to a file.

Two buttons at the top right corner allow saving current settings to the core's non-volatile flash memory and performing reinitialization of the core.

Additionally, buttons *Save profile* and *Load profile* allow saving current core settings as seen in CoreConfig software to a file and reading them back from a file.



Language:	English
Connect using:	TUSB board
FPA SN:	206204040

Core SN:	206204040
FPA SN:	206204040
Firmware:	24.4542
Sensor:	640 x 480, 17 um, New
Variant:	0x21
FPA temperature:	23.74
CoreConfig version:	2.4531
Snapshot resolution:	Sensor (before interpolation)

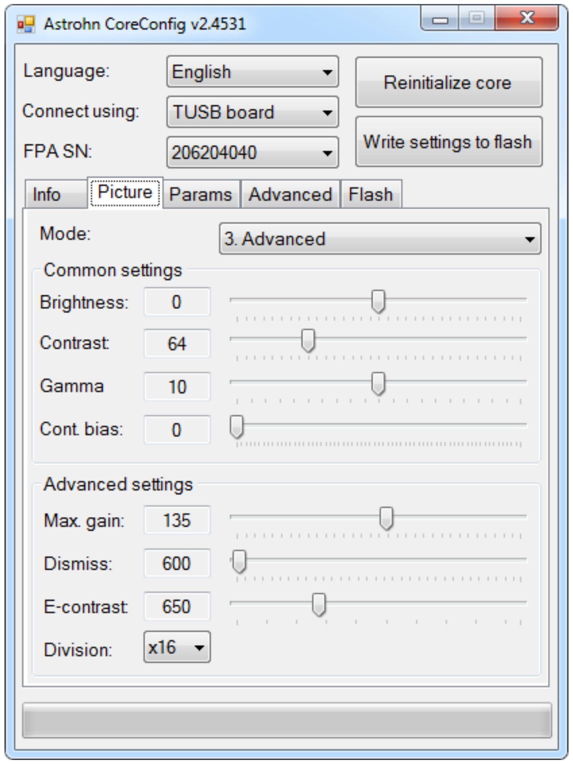
Figure 31. Info tab

Core image settings

All the adjustable core's parameters responsible for such picture features as brightness, contrast, etc. are available in *Picture* tab page (see Figure 33).

List of parameters available there:

- Brightness/contrast mode (register 0x78)
- Brightness (register 0x77)
- Contrast (register 0x76)
- Gamma (register 0x80 – not implemented)
- Contrast bias (registers 0x75, 0x74)
- Maximum gain (register 0x7C)
- Levels to dismiss (registers 0xD3, 0xD2 – extreme pixel removal)
- Extra contrast (register 0x7B, 0x7A)
- Division factor (register 0x79)



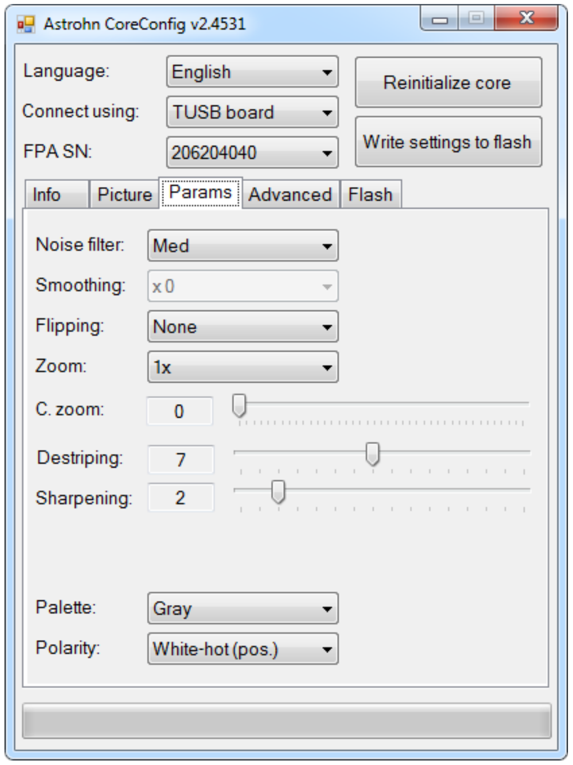
Language: English
Connect using: TUSB board
FPA SN: 206204040
Mode: 3. Advanced
Brightness: 0
Contrast: 64
Gamma: 10
Cont. bias: 0
Max. gain: 135
Dismiss: 600
E-contrast: 650
Division: x16

Figure 32. Picture settings tab

Additional core settings

Parameters responsible for image filtering, quality improvement, video output mode and others are available in *Params* tab page (see Figure 34):

- Noise filter (register 0x7F - temporal filter)
- Flipping (register 0x86)
- Zoom factor (register 0xD6)
- Continuous zoom value (registers 0xE3, 0xE2)
- Destriping (register 0x82)
- Sharpening (register 0x83)
- Palette (register 0xD5)
- Polarity (register 0xD5)



Language: English
Connect using: TUSB board
FPA SN: 206204040
Noise filter: Med
Smoothing: x 0
Flipping: None
Zoom: 1x
C. zoom: 0
Destriping: 7
Sharpening: 2
Palette: Gray
Polarity: White-hot (pos.)

Figure 33. Params tab

Advanced core settings

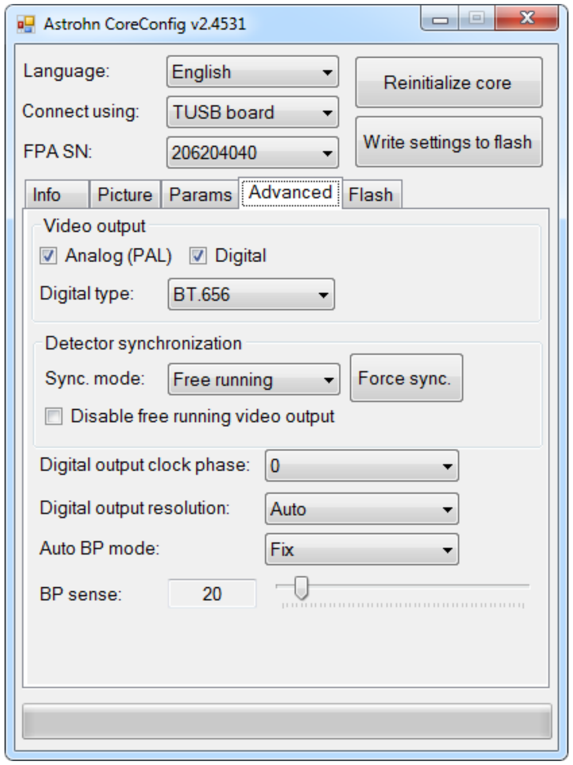
Advanced tab (Figure 35) gives access to video output type selection (register 0x7D). Analog and digital video outputs can be enabled independently of each other.

External synchronization mode can also be controlled and tested here.

Digital output clock phase dropdown box lets the user to change digital video output signal's clock phase (register 0x44).

Digital output resolution dropdown box controls digital video output resolution (register 0xB0).

Additionally Advanced tab contains controls for real time BP detection – *Auto BP mode* dropdown box (register 0x9D) and *BP sense* slider bar (register 0x9E).



Language: English
Connect using: TUSB board
FPA SN: 206204040
[✓] Analog (PAL) [✓] Digital
Digital type: BT.656
Sync. mode: Free running
[ ] Disable free running video output
Digital output clock phase: 0
Digital output resolution: Auto
Auto BP mode: Fix
BP sense: 20

Figure 34. Advanced tab

Core Flash tab

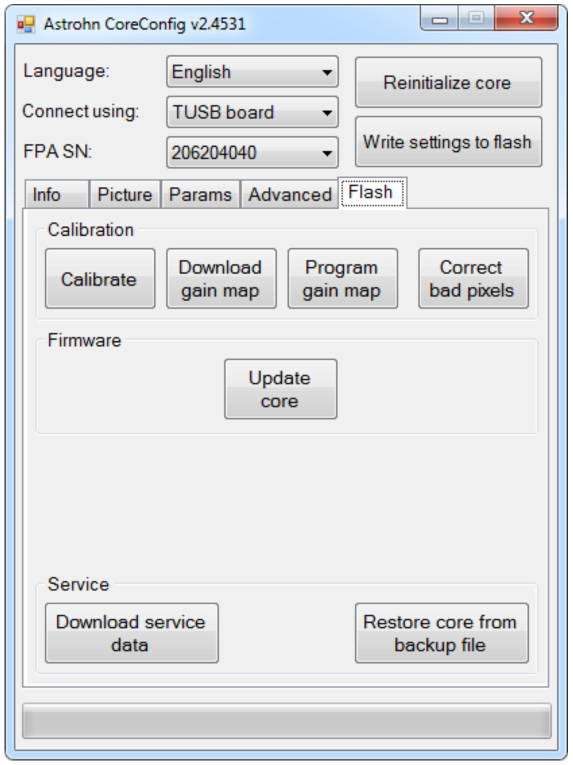
See Gain calibration section about why and when gain calibration should be performed. Gain calibration can be performed using CoreConfig PC Software and AST-7D board.

To calibrate gain open *Calibrate* tab (Figure 36) when the core is connected to the PC over USB using AST-7D board and press *Calibrate* button. Software will ask to place Astir2 core in front of a hot black body. Once the core is positioned click *Ok* in the dialog box to confirm. After about 15 seconds the software will ask to place cold black body in front of the core and to confirm by clicking *Ok*. It will take several seconds longer to process cold black body data (compared with hot black body). At the same time it will update gain correction table in core's working memory and ask to save it to file. Saving to a file is a convenient way to have this table available for use later if, for instance, core is used with more than one lens. The software will then ask whether new gain correction table has to be written to non-volatile flash core's memory. After confirmation previous gain map in the core will be lost and replaced with new one. If new gain map is not written to non-volatile core's memory it will only be used until the core is reinitialized or power-cycled.

There is also an option to download gain map currently stored in non-volatile core's memory by clicking *Download gain map* button.

To program gain map from file on the PC without going through the calibration procedure with two black bodies, click *Program gain map* button.

Correct bad pixels button opens additional window which lets the user to manually identify bad pixels.



Language: English

Connect using: TUSB board

FPA SN: 206204040

Figure 35. Core flash tab

Core firmware can be updated by clicking *Update core* button under *Update* tab and following instructions. There is a fail-safe mechanism implemented which lets the core power up even after failed update (caused by power supply interruption, etc.) – see Firmware update section for more information. When the *Update core* button is clicked the window will opened were the user has to select a folder containing firmware update files. Firmware folder for the 640x480 core will contain "Astir2_a5_3b" in its name. Firmrware for the 384x288 core will contain "Astir2_a5_3a" in its name.

Button *Download service data* downloads service information from the core's memory. It may be useful when trying to identify potential issues with the core.

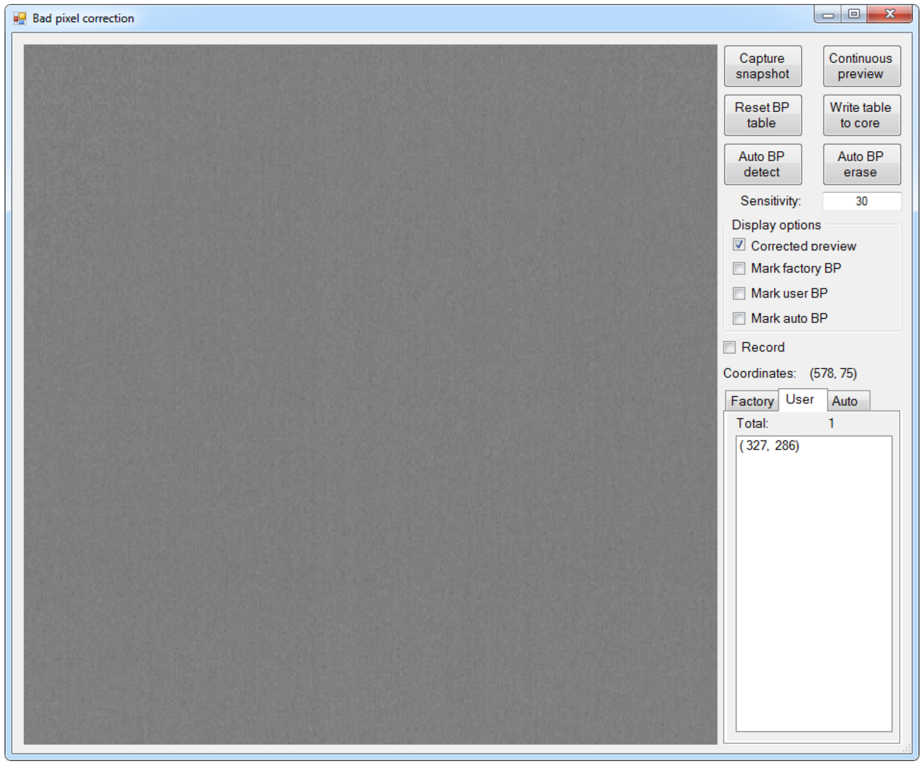
Restore core from backup file button restores core flash memory content from a file.

Bad pixels

To manually identify new bad pixels go to *Flash* tab and click *Correct bad pixels button*. This will open bad pixel correction window (Figure 37). After the window opens, wait for several more seconds until the image from the cores becomes visible. New snapshot can be taken at any time by clicking *Capture snapshot* button. It is usually a good idea to check for bad pixels in several scenes with different temperatures – for instance when the core is pointed at the table and when it's covered with hand. The image from core can be zoomed and panned. Zooming is performed with mouse wheel (rotate mouse wheel up to zoom in and down to zoom out. To pan the image drag it while holding left mouse button.

Right mouse button is used to mark bad pixels. Clicking on a good pixel will mark it as bad, and vice-versa.

There are three lists with bad pixels on the right hand side of the bad pixel correction window. First list contains factory marked bad pixels. This list cannot be modified by the user. Second list contains user marked bad pixels. Third list is filled automatically by the core (see Auto bad pixel list). Two buttons (*Auto BP detect* and *Auto BP erase*) are used to start automatic bad pixel identification and to clear the automatic bad pixel list.



Sensitivity: 30

[✓] Corrected preview
[ ] Mark factory BP
[ ] Mark user BP
[ ] Mark auto BP
[ ] Record

Coordinates: (578, 75)

Total: 1

(327, 286)

Figure 36. Bad pixel correction window

To see the image without bad pixel correction uncheck *Corrected preview* check box. Unchecking it will suspend factory, user and auto list based bad pixel correction. List of user bad pixels can be cleared by clicking *Reset BP table* button. The list will be cleared only in bad pixel correction window (not in the core). Closing and opening this window will show original list again.

To write any changes made to the bad pixel correction list to the core's non-volatile memory click *Write table to core* button. It will take several seconds. After dialog box pops up click *Ok* and the core will reinitialization using updated bad pixel correction table.

Bad pixels contained in any of the three lists can be marked and visualized in the preview pane by clicking on the checkboxes: *Mark factory BP*, *Mark user BP*, *Mark auto BP*.