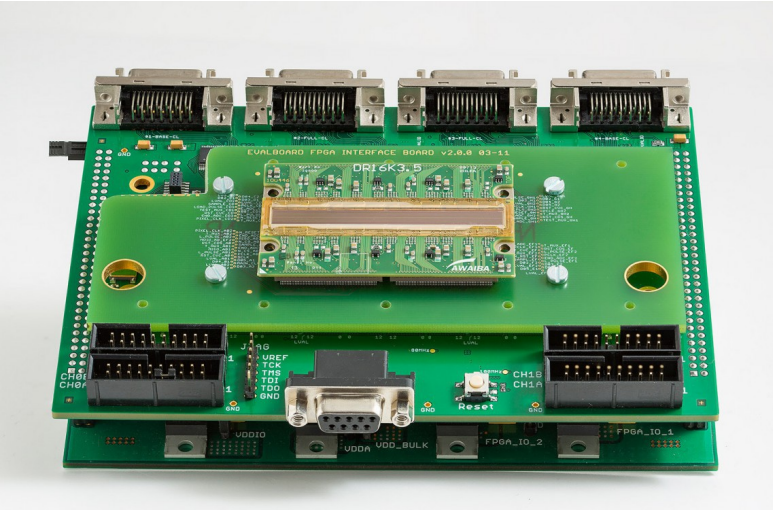
DRAGSTER
Evaluation Board



AWAIBA

Revision History:

Version	*Date*	*Modifications*	*Author*
4.0	19/05/15	Updated document	Fátima Gouveia

Table of Contents

Index of Tables

Index of Figures

1 General Concept

The evaluation system features a highly configurable hardware which enables an easy setup of Awaiba's Dragster line scan family for a quick sensor evaluation.

Image data is transferred to a frame grabber over high speed camera link. Any grabber that supports at least a camera link base configuration can directly acquire data from any Dragster version. The system controls sensor operation using an FPGA to define the state machine timings, it acquires data synchronously and multiplexes that data to Camera-Link Full interfaces.

Over an RS232 serial interface the user can have access to the state machine configuration and all sensor registers. It is possible to read back those sensor registers. Please refer to the serial communication section for further details of the serial communication protocol.

For debugging process, some digital IO sensor control signals sensor are routed as CMOS TTL signals to the parallel connectors.

The Evaluation board features two clock possibilities, one at 100MHz and another at 80MHz. In the current implementation the 80MHz oscillator is used to generate a sensor global clock of 20MHz, which is used for the FPGA state machine, while the Camera-Link interfaces will run at 40Mhz. All timing information in the reminder of this document are with respect to the 20MHz clock.

The evaluation of DRAGSTER sensors uses a stack of three boards:

- Sensor Head Board – Contains the DRAGSTER sensor and connects to the FPGA board
- AWAIBA Evaluation Board – versatile and easy to program, this board includes a SPARTAN 3AN FPGA, a micro controller for RS232 communication and 2 Camera Link full interfaces. It is the main board where the Sensor Head Board is connected.
- Evaluation Board Power Supply - Under the main board there is a power supply board to create a group of filtered voltages to apply to the sensor, micro controller and RS232 sensor operation and camera-link

By default we have an FPGA code that reads out all sensors over 1 Camera Link, base connector, capable of operating the sensor with maximal line rate however discarding lines if the data rate coming from the sensor exceeds the data rate that can be transmitted over one base Camera Link connection. Alternatively we have specific FPGA codes that can support the sensors with full data transmission using up to 4 Camera Link base or 2 Camera Link medium configurations.

1.1 Architecture Diagram

Resume of the interface connections between the different components of the Evaluation Board.

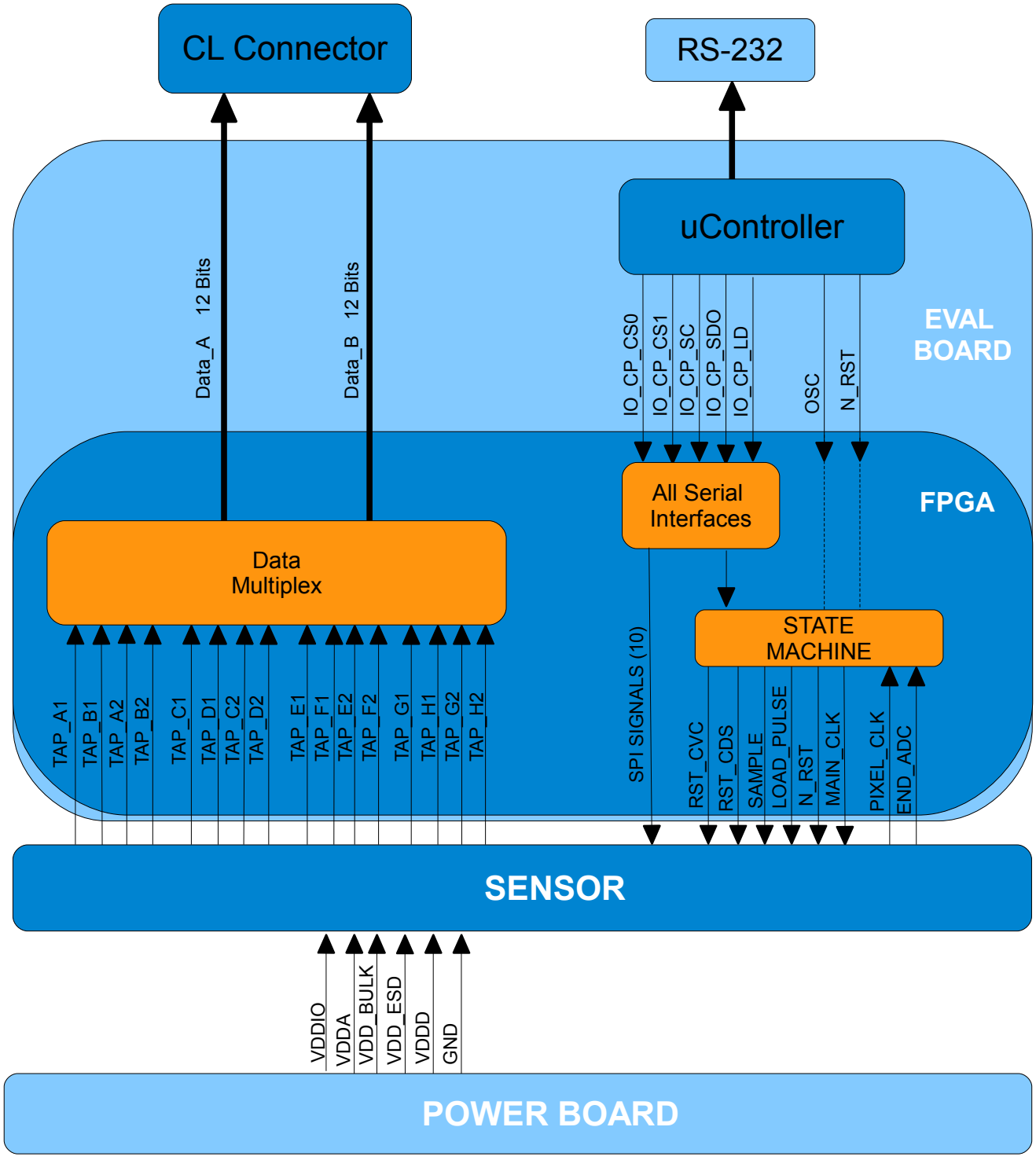


Figure 1: Block Diagram

1.2 Power Connection

The system main input voltage is 12V DC (Figure 2). A green LED in the power board will turn on when the power is connected. It has also a 1A soldered fuse for user safety and board protection. After any wrong manipulation check its connectivity and if the LED does not start please replace the fuse by another with same specification.
All voltages are generated in this board.

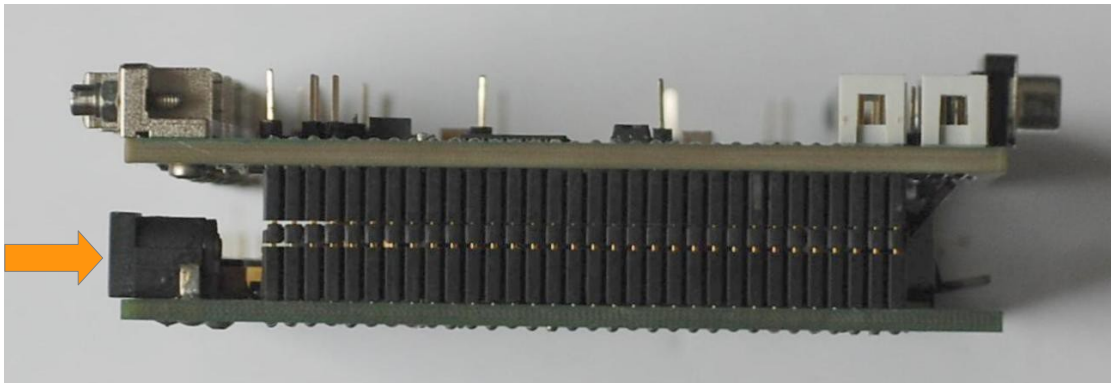


Figure 2: Evaluation Board Power connection

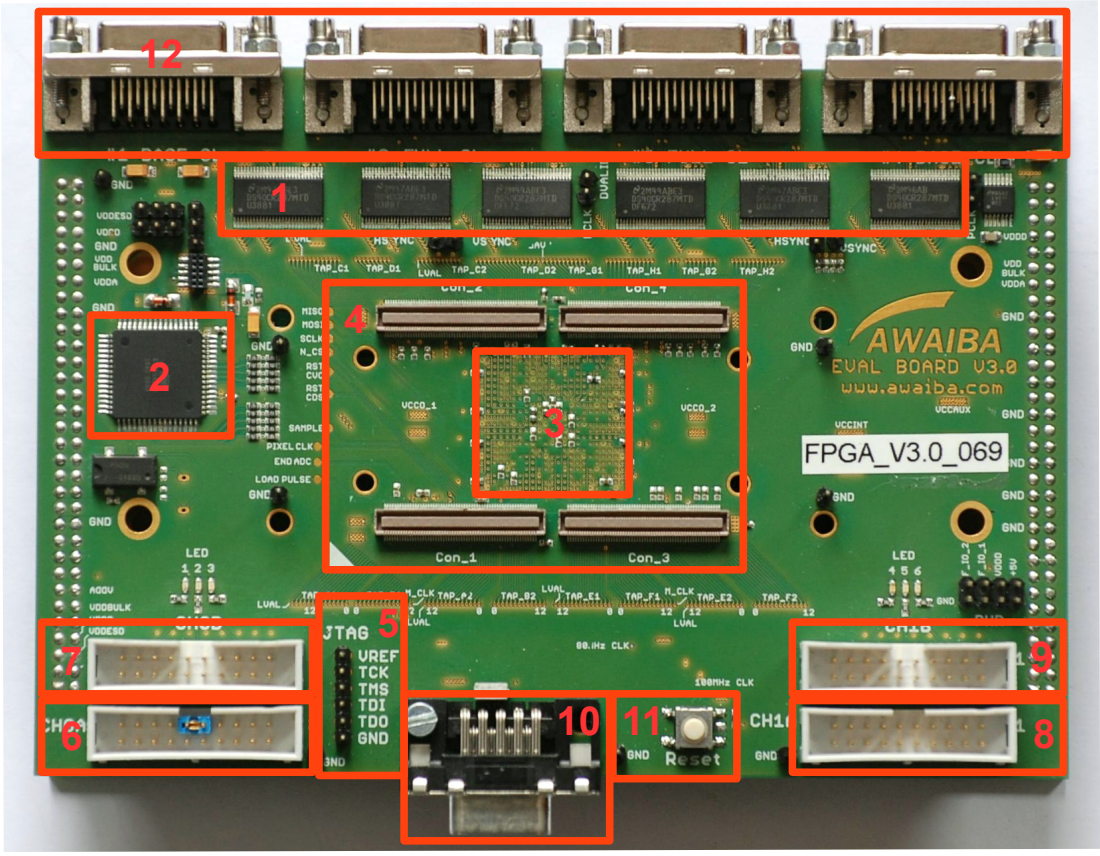
1.3 Board Reset

It's a good practise to perform a full system reset before starting evaluating the sensor. In the AWAIBA Evaluation Board there are two possibilities to make this reset: by pressing the Reset button or by sending the reset command. Both have the same effect of restarting the state machine's operation. After this the sensor is loaded with the default values defined in the datasheet, which may obviously need to be reconfigured to match the current setup.

Note: *Always perform a reset to the system after power up to guarantee a defined power up status!*

If the board does not start operation after power up, if the sensor was wrong configured or if data is not consistent, please press the board reset button and reconfigure the sensor.

2 Evaluation Board components and connectors pinout



Legend:	
1. Camera Link Drivers	7. Parallel TTL Connector Ch0B
2. ATMega 128 Controller	8. Parallel TTL Connector Ch1A
3. Spartan 3AN FPGA (bottom side)	9. Parallel TTL Connector Ch1B
4. Sensor Connectors Base	10. Serial Communication Connector
5. JTAG Connector	11. Board Reset Button
6. Parallel TTL Connector Ch0A	12. Camera-Link Connector

2.1 Debug points in the Parallel TTL Connectors

TTL Connector	S-Connectors Channel	Signal Name
CH0A	S00	80MHz clock
	S01	100MHz clock
	S02	Sensor clock
	S03	Camera Link drivers clock
	S04	VCC
	S05	Trigger selector jumper
	S06	Internal trigger signal
	S07	External trigger signal (input)
CH0B	S10	Sensor SCLK
CH1A	S16	Test Mode Jumper (Load Pulse Off)
	S17	VDD
	S18	END ADC AB1
	S19	LVAL AB1
	S20	Sensor Load Pulse
	S21	Sensor Sample
	S22	Sensor RST CVC
	S23	Sensor RST CDS
CH1B	S30	Pixel CLK AB1

Table 1: TTL Connectors Debug Signals

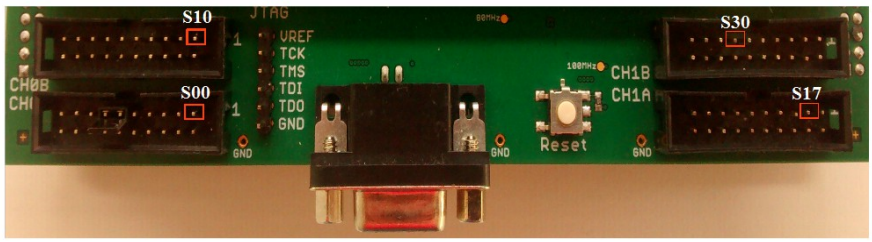


Figure 3: TTL Connectors Debug Signals Identification

3 Dragster Sensors

3.1 Assembling DR_4K_7um on the Evaluation board

The DR_4K_7um Sensor is assembled in connector 1 of the Evaluation Board, with the triangle marks aligned (Figure 4).

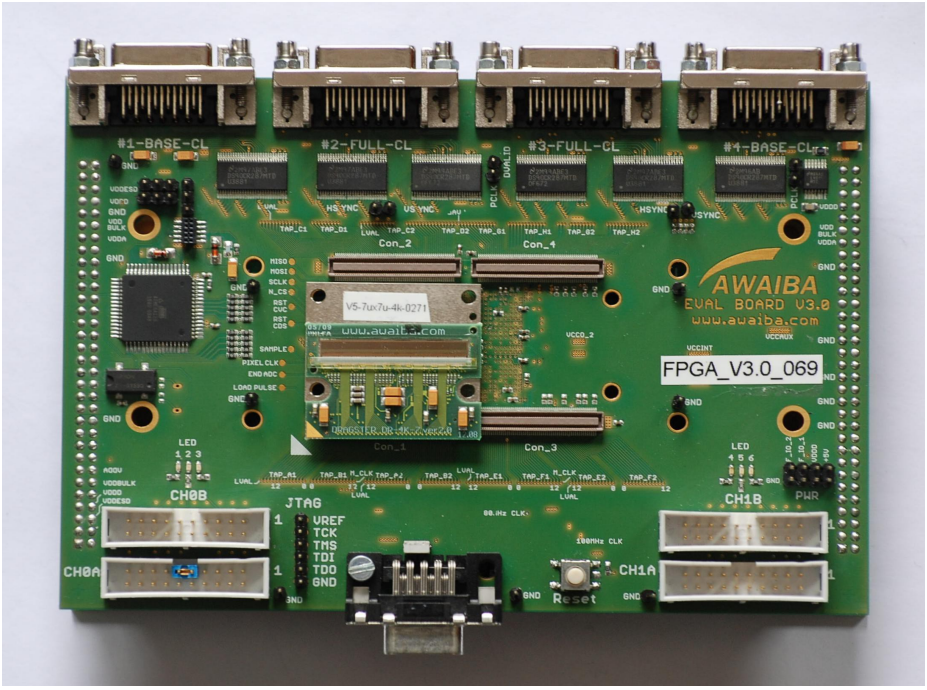


Figure 4: Assembling DR_4K_7um on the Evaluation board

3.2 Assembling DR_8K_7um on the Evaluation board

The DR_8K_7um sensor is assembled in the BOTTOM SIDE connectors (Connector 1 and Connector 3) of the Evaluation Board, with the triangle marks aligned (Figure 5).

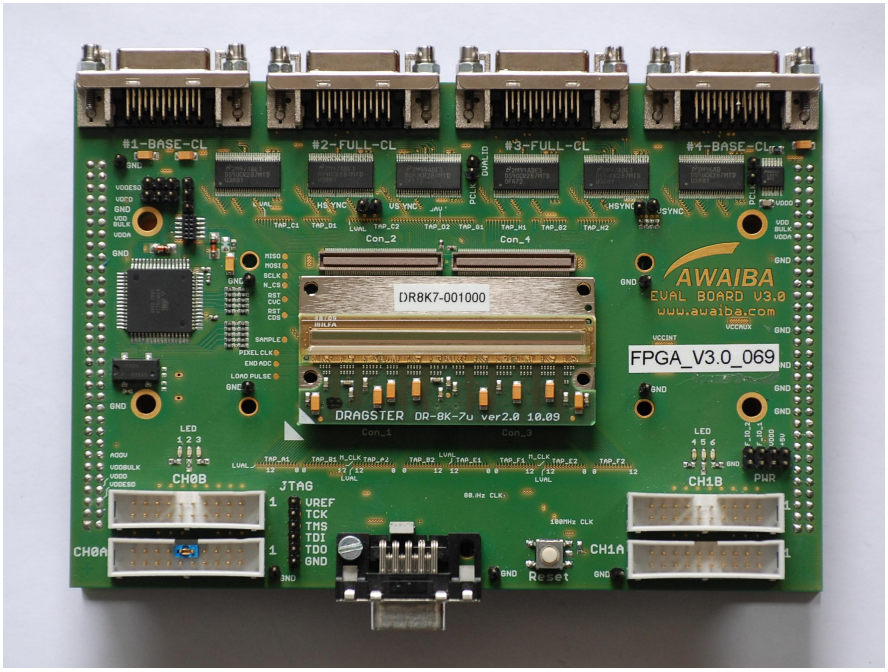


Figure 5: Assembling DR_8K_7um on the Evaluation Board

3.3 DR_8K_7um

Dragster 8K_7um sensor is made out of segments with 2K pixels and 7um pitch. Each 2K segment has 2 taps outputs of 12 bits width. At sensor's beginning (readout direction) there are 32 extra black pixels that are read out and make part of the first 2K segment (taps A1, B1).

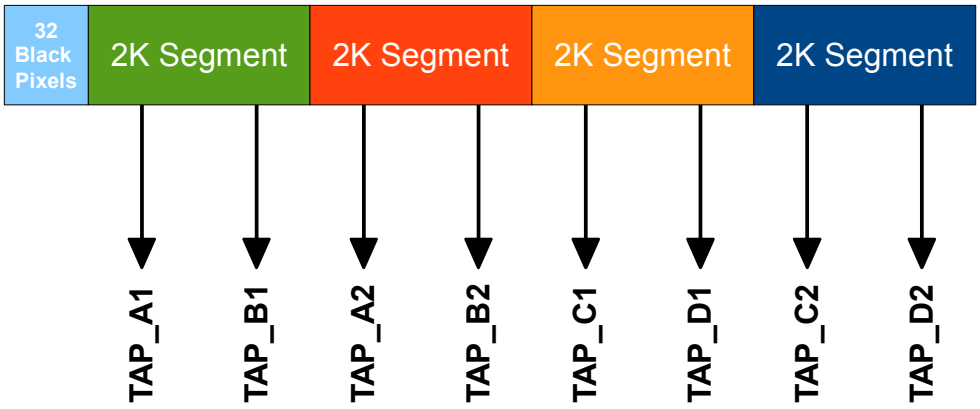


Figure 6: TAP Organization DR 8K_7um

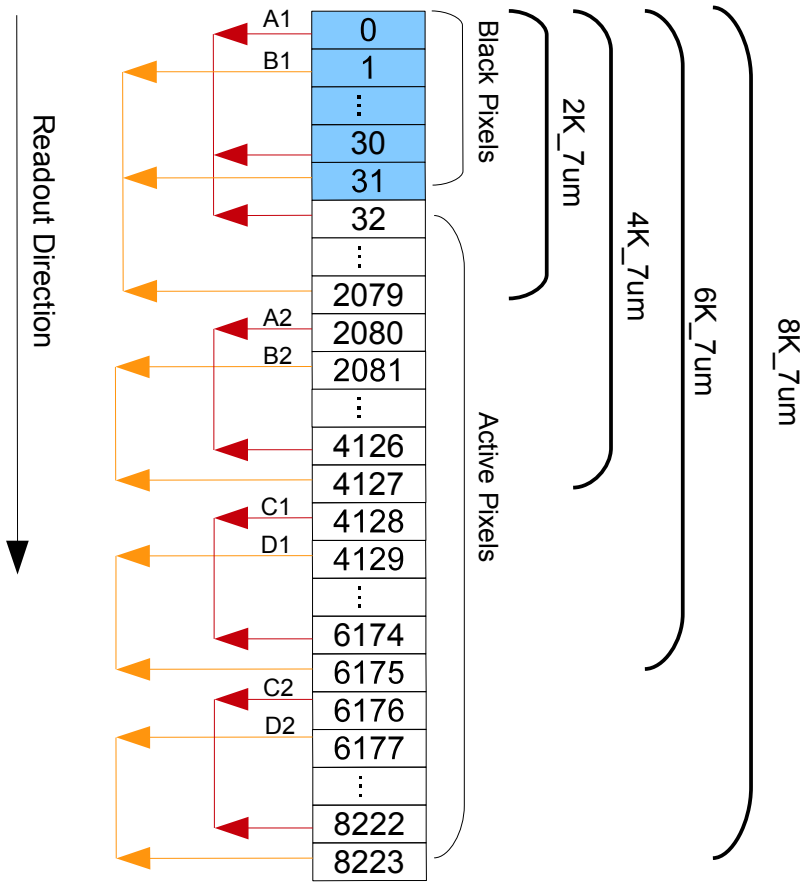


Figure 7: TAP Organization according to pixel numbers

3.4 Assembling DR_8K_3.5um on the Eval board

The DR_8K_3.5um sensor is assembled in connectors Connector 1 and Connector 2 of the Evaluation Board, with the triangle marks aligned (Figure 8).

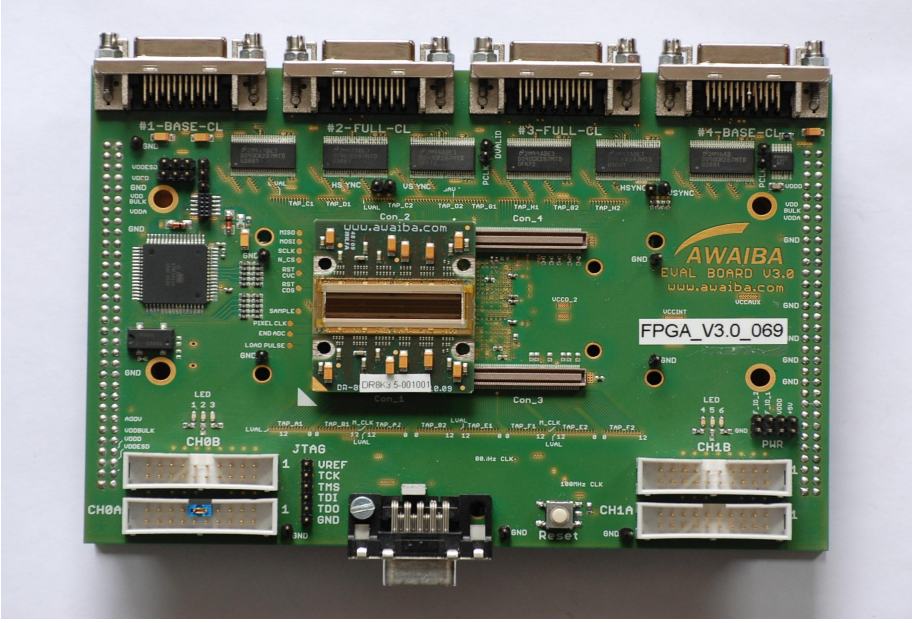


Figure 8: Assembling DR_8K_3.5um on the Evaluation board

3.5 Assembling DR_16K_3.5um on the Eval board

The DR_16K_3.5um sensor is assembled in the 4 connectors of the Evaluation Board. The sensor orientation is such that the triangle marker on the sensor head board aligns with the white triangle on the Evaluation board (Figure 9).

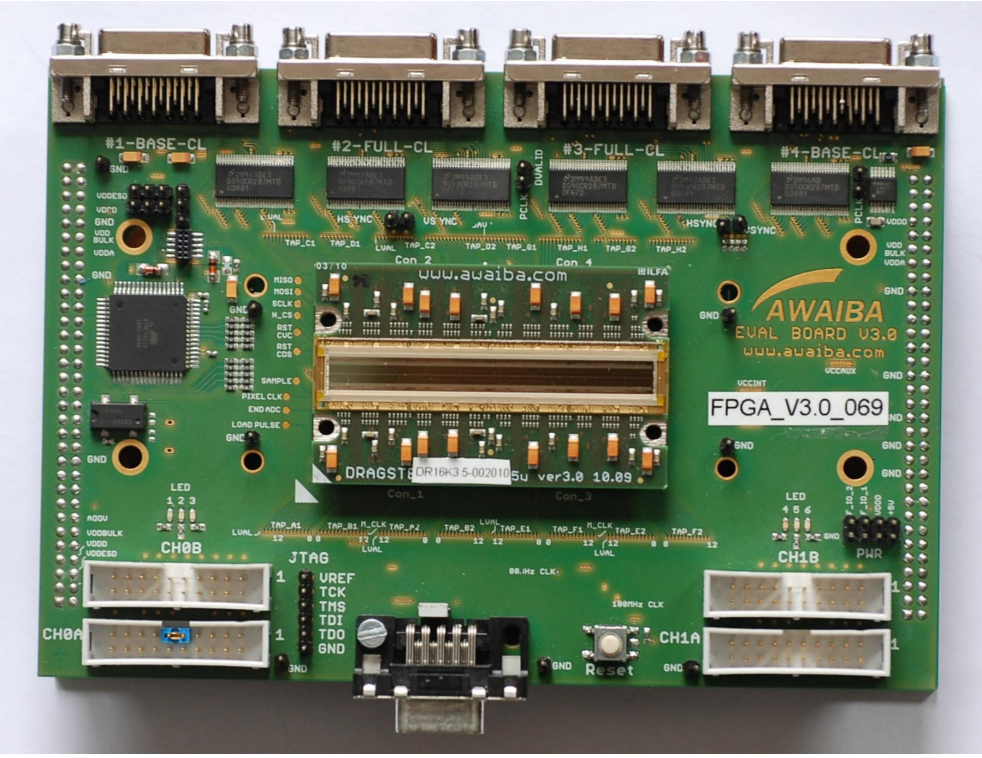


Figure 9: Assembling DR_16K_3.5um on the Evaluation Board

3.6 DR_16K_3.5um

Dragster 16K sensor is organized in segments of 4K pixels, described like this due to the 4 taps with 12 bit each. At the sensor's beginning (readout direction) there are 64 black extra pixels placed that are read out and make part of the first 4K segment (taps A1, B1, E1 and F1).

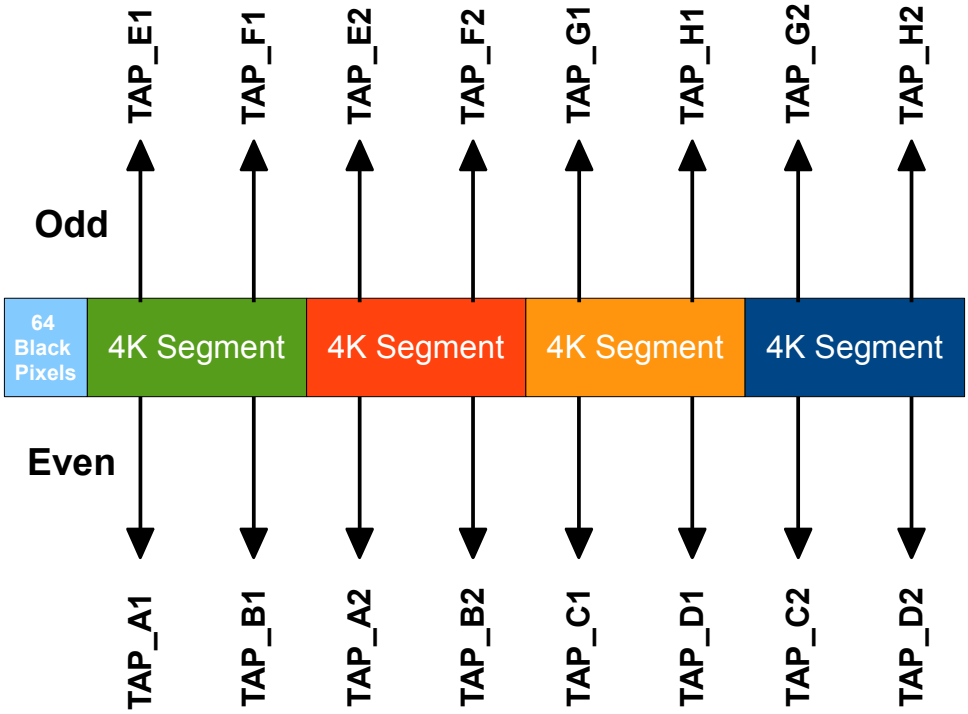


Figure 10: TAP organization DR_16K_3.5um

The Dragster 16K sensor has in total 16 TAPS of 12bits.

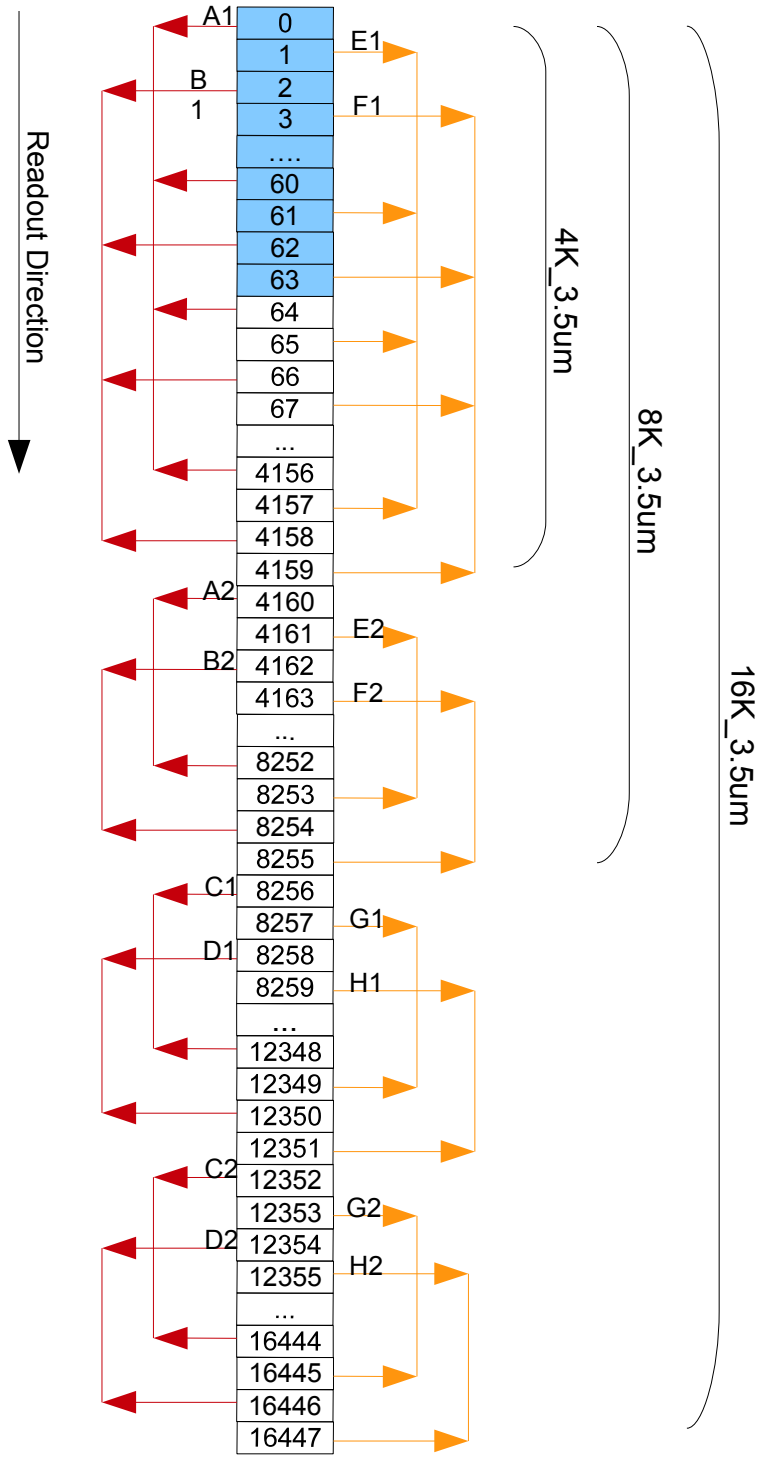


Figure 11: TAP Organization according to pixel numbers

4 Camera-Link Data Interface

4.1 TAP Assignment DR_16K_3.5um

The FPGA board can receive up to 16TAPS of 12 Bits each so that it reads out all DR_16K_3.5um sensor. The data received in the FPGA are multiplexed and sent to one Camera Link Connector through Data_A and Data_B.

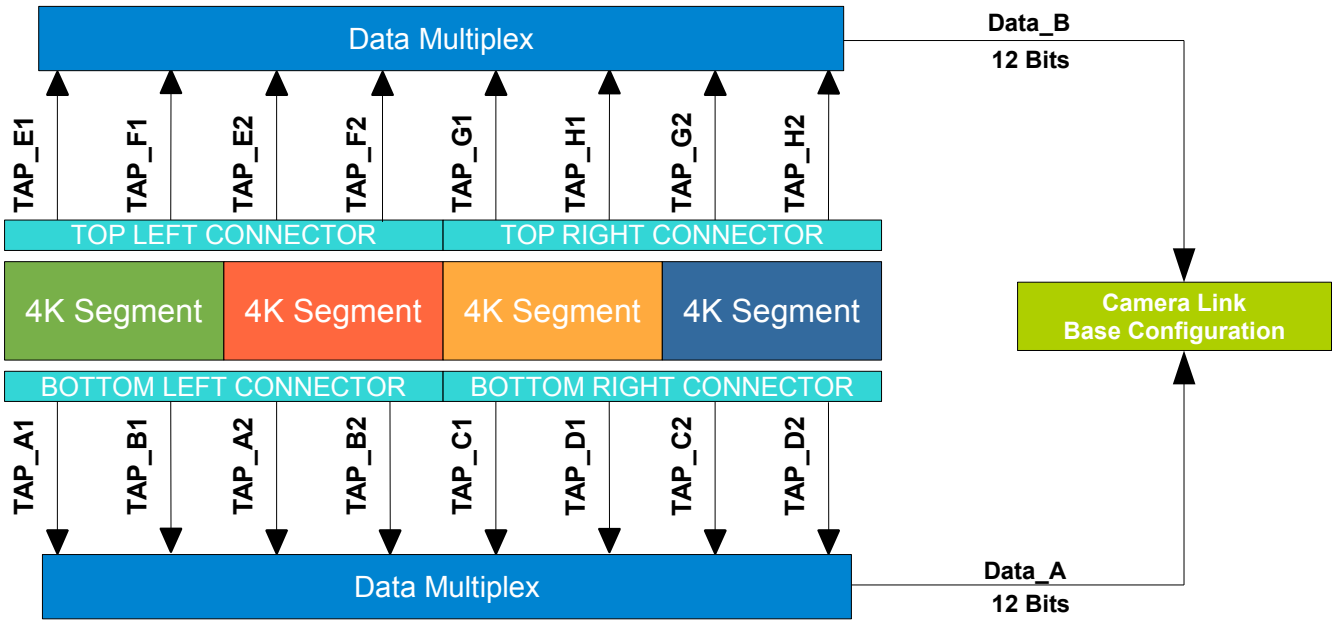


Figure 12: Camera Link Taps

The explanation above is valid for DR_16K_3.5um sensors. Because the FPGA implementation is compatible with all sensor variations, that means that in case other sensors are tested, like DR_4K or DR_8K, not all the TAPs will contain valid data.

The Camera-Link driver chips can be enabled or disabled via the serial interface. Please refer to the Serial communication interface section for this details.

Note: *Make sure the Camera-Link drivers are enabled when trying to capture image data!*

4.2 TAP Assignment DR_8K_7um

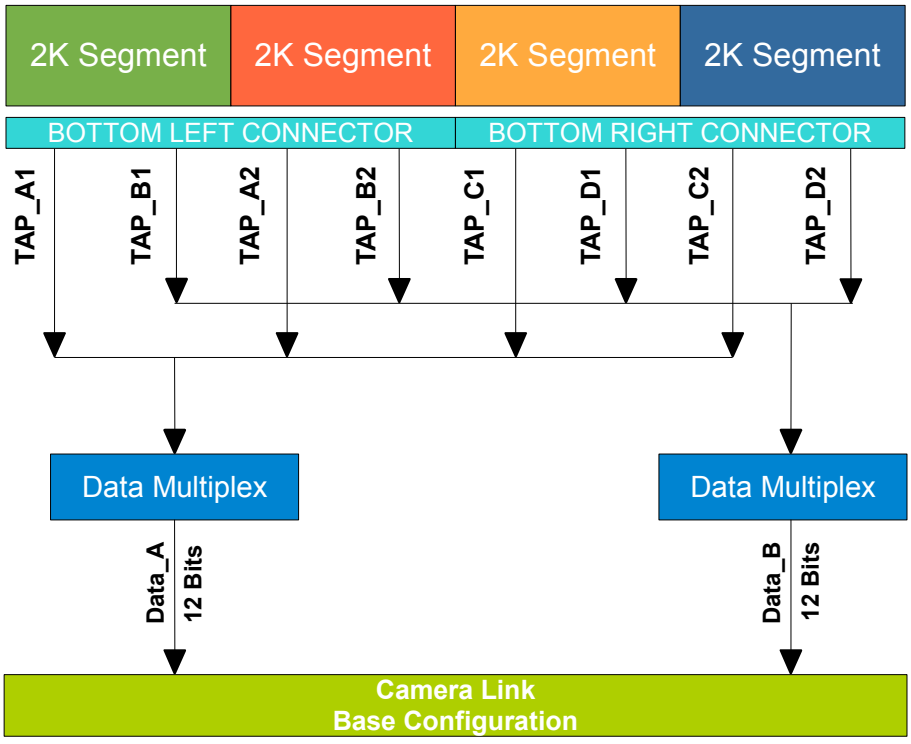


Figure 13: Camera Link Taps

5 Serial configuration interface

State machine's basic parameters, like line and integration time, together with all sensor's registers are accessible via a RS232 interface.

The serial communication uses a standard serial communication port on the PC side and a D-SUB 13 connector on board's side. (Serial communication over Camera-Link is not implemented).

5.1 Serial Port Configuration

- Baud Rate: 19200
- Databits: 8
- Stopbits: 1
- Parity: none
- Flow control: none

5.2 Communication data structure

HEX Format	Function
23	Start of transmission
80	Address of Evaluation Board
40 / 41 / 44	Command
DATA	MSBs Data for line/integration times or Register address of the sensor
ADDRESS	LSBs Data for line/integration times or Data to send to the register
0D	End of transmission

Table 2: Data structure for uController

5.3 Command Structure

Command in HEX	Data	Valid Data Range	Description
0x30	None	none	Enable 4k-side Camera Link Chips
0x31	None	none	Disable 4k-side Camera Link Chips
0x32	None	none	Enable 8k-side Camera Link Chips
0x33	None	none	Disable 8k-side Camera Link Chips
0x40	<HI_byte> <LO_byte>	0x0000..0xFFFF	Set Integration time
0x41	<HI_byte> <LO_byte>	0x0000..0xFFFF	Set Line time
0x44	<HI_byte> <LO_byte>	0x0000..0x0FFF	Write to AB1 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x1000..0x1FFF	Write to CD1 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x2000..0x2FFF	Write to AB2 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x3000..0x3FFF	Write to CD2 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x4000..0x4FFF	Write to EF1 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x5000..0x5FFF	Write to GH1 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x6000..0x6FFF	Write to EF2 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x7000..0x7FFF	Write to GH2 segment SPI channels
0x44	<HI_byte> <LO_byte>	0x8000..0x8FFF	Write to all bottom side SPI channels
0x44	<HI_byte> <LO_byte>	0x9000..0x9FFF	Write to all top side SPI channels
0x44	<HI_byte> <LO_byte>	0xA000..0xAFFF	Write to all SPI channels
0x44	None	0xBXXX	Send a reset

Table 3: Command Structure

Please note that the Camera-Link interface is made of a chip set with internal PLL's. In order to reset the PLL's for the case they might have booted in illegal state, use the commands 31 and 30.

Note: *Make sure the Camera-Link drivers are enabled when trying to capture image data!*

5.3.1 Setting the Line Period

To set the line time, use command 0x41 and a 16bit value. The line time is adjusted in multiples of 50ns (20MHz main clock). It is user responsibility to keep the line time above the minimum value permitted with the applied sensor register setting. The line time must in any case be longer than the maximum of:

- ADC time (controlled by sensor register 0x09) + 1us
- Integration Time (controlled by command 0x40) + 1us

but never larger than 3276us.

Note: *Setting the line period shorter than the integration time or the time required for ADC conversion shorter than data readout time will set the state machine in an undefined state with unpredictable output, and require a global reset to restore proper operation!*

5.3.2 Integration Time

The integration time is controlled by command 0x40 and it's value is also multiple of 50ns, like the line period.

The minimum allowed integration time for the evaluation board is 3us or 0x0078h. (the sensor itself permits 2us minimal integration time). The maximum programmable integration time is 0xFFFF which corresponds to 3276us. It is the users responsibility to assure that the integration time is set shorter than the line period, otherwise the state machine will fall in an illegal state.

	Start	Command	Integration Time	End of communication
Hex	2380	40	<0050> - <FFFF>	0D

Table 4: Integration Time Configuration

5.3.3 Registers content

The Dragster Sensors features a separate SPI for each segment (2K). The SPI "segments" can be selected according to the following table.

1st 4 bits Data	Selected SPI
0	AB1 segment SPIs
1	CD1 segment SPIs
2	AB2 segment SPIs
3	CD2 segment SPIs
4	EF1 segment SPIs
5	GH1 segment SPIs
6	EF2 segment SPIs
7	GH2 segment SPIs
8	All BOTTOM SIDE segments SPIs
9	All TOP SIDE segments SPIs
A	Write to all SPIs

Table 5: SPI selection

Note: *Normally the best results should be obtained by writing identical content to all SPI registers. Writing conflicting operation modes or settings to the respective segments may stop the sensor or Evaluation Board from functioning or lead to severely distorted image data!*

The sensor SPI is controlled with the command "D" and the bit structure is:

	Start	Command	Data MSB	4bit Register Address (0-F)	8bit Register data	End of communication
Hex	2380	44	8	<0-F>	<00>-<FF>	0D

Table 6: SPI Command Structure

Sensor registers must be written in first place and the new settings will only take effect after setting the "update request bit" in the control register (address 0x01)

Please refer to the Dragster Datasheet Section 9 for more detailed description of the sensor registers.

5.3.4 Reset to Evaluation Board and Sensor

To send a reset to the Evaluation Board and sensor a command must be sent via RS232.

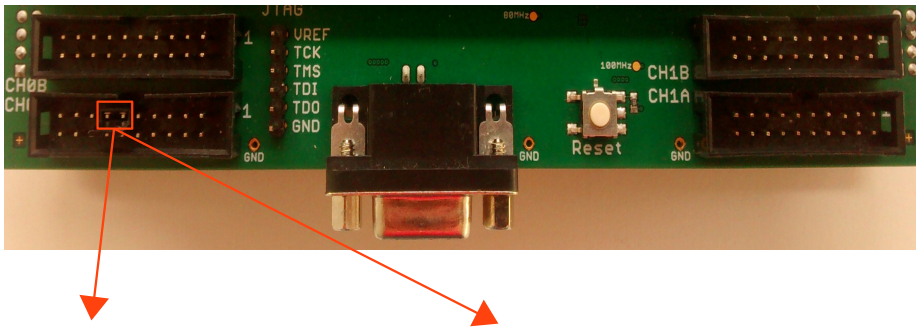
	Start	Command	Data MSB	4bit Register Address (0-F)	8bit Register data	End of communication
Hex	2380	44	B	0xX	0xXX	0D

Table 7: Reset to Evaluation Board and Sensor

Note: *The 12 less significant data bits are "don't care" bits!*

6 Trigger Modes

The state machine implemented for timing control allows internal or external trigger modes and to switch between these modes is used a jumper.



Jumper between S05 and S06
Internal Trigger mode

Remove the jumper
External Trigger applied on S07

Figure 14: Trigger Modes description

The S07 Channel is used to input the trigger signal and the pin right under can be used to connect GND.

6.1 Internal Trigger Mode

For internal trigger mode a jumper must be placed between S05 and S06, pulling the S06 to HIGH state. The internal trigger is generated according to the line time.

6.2 External Trigger Mode

For external trigger operation the jumper for internal trigger mode must be removed and the external trigger signal must be supplied in the S07 pin in the Parallel TTL Connector Ch0A. To restore to internal mode the jumper must be placed like described in the previous point.

Note: *The implemented state machine will discard a trigger event that occurs while the sensor is still in a readout/integration mode. The trigger event will be discarded if it is sent before the line period counter reached it's end state. The user can use the internal signals of the state machine available on the Parallel TTL Connectors to define the correct trigger moment!*

7 Load pulse

When the Auto Generation bit, from register 0x02, is set to “0” the sensor sends the output data without the need of sending the external Load Pulse.

The Load Pulse generation is turned off on the FPGA, when the Test Mode Jumper is connected as in figure 15.

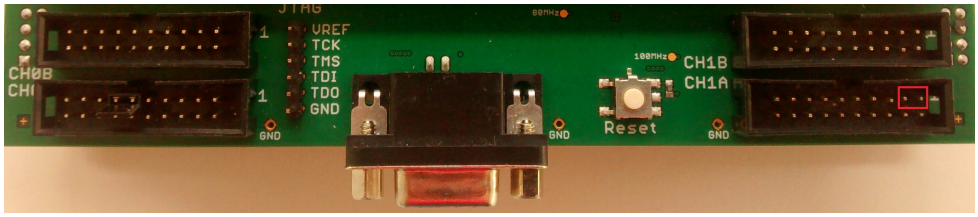


Figure 15: Jumper on S16 and S17 to remove the Load Pulse.

8 ISAC – RS232 Communication Tool

8.1 Software Installation

The ISAC tool can be used to communicate with Dragster line-scan sensors and its a simple software that allows the user to send commands to sensor registers using a RS232 communication port. With this software the user can configure all the sensor operation modes in order to achieve the best configuration settings for the set up and take advantage over it's great performance.

To run the installation package please launch the executable sent with the Evaluation system, follow the instructions and one ISAC icon will appear in the desktop. This software is distributed only under the operating system Windows.

The following sequence of steps are mandatory before sending a command to the sensor:

- Configure the serial port under “EvalBoard” → “Configure” → “Serial Port” - choose the right port of your system from the drop menu (figure 17)
- Choose the sensor under test in the “Dragster Version” tab (figure 16)
- Send registers in the “Registers” tab (figure 18)
- As extra options, it's possible to check the Ini file considered by the software (“File” → “IniFile...”), to activate and de-activate both camera link interfaces (“Evalboard” → ”CL drivers”) and to double check the software version (“Help” → “About...”)

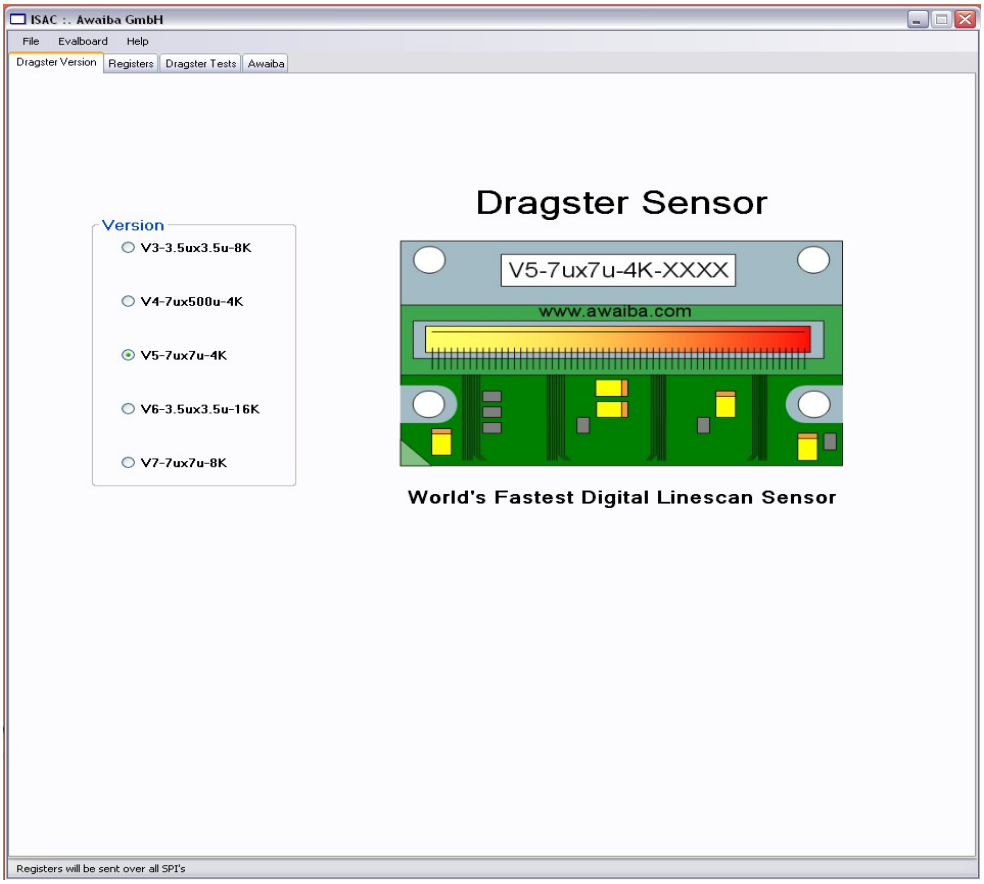


Figure 16: ISAC software

8.2 Configuration of the communication port

Before start communicating with the sensor, it's necessary to set the communication port parameters.

Click on “Evalboard” → “Configure” → “Serial Port...” menu and set the COM port number where the sensor is connected to.

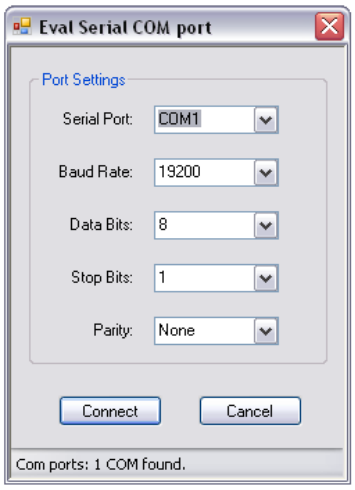


Figure 17: ISAC - Serial Port Configuration

All other parameters (baud rate, data bits, stop bits and parity) are well selected by default so just press “Connect”.

8.3 Choose the sensor version under test

The sensor version is chosen in ISAC's first tab. It is important to choose the right version because of SPI compatibilities since they have different address assignments.

8.4 Register Configuration

In the **Registers** tab all 10 Dragster register values are accessible together with the state machine timings.

Note: *Edit the respective register content but take in account that some modes are not compatible and can lead to sensor wrong function. Please refer to the respective sections of Dragster sensor datasheet and Evaluation board documentation for description of the register content and functionality of the bit switches in the control registers!*

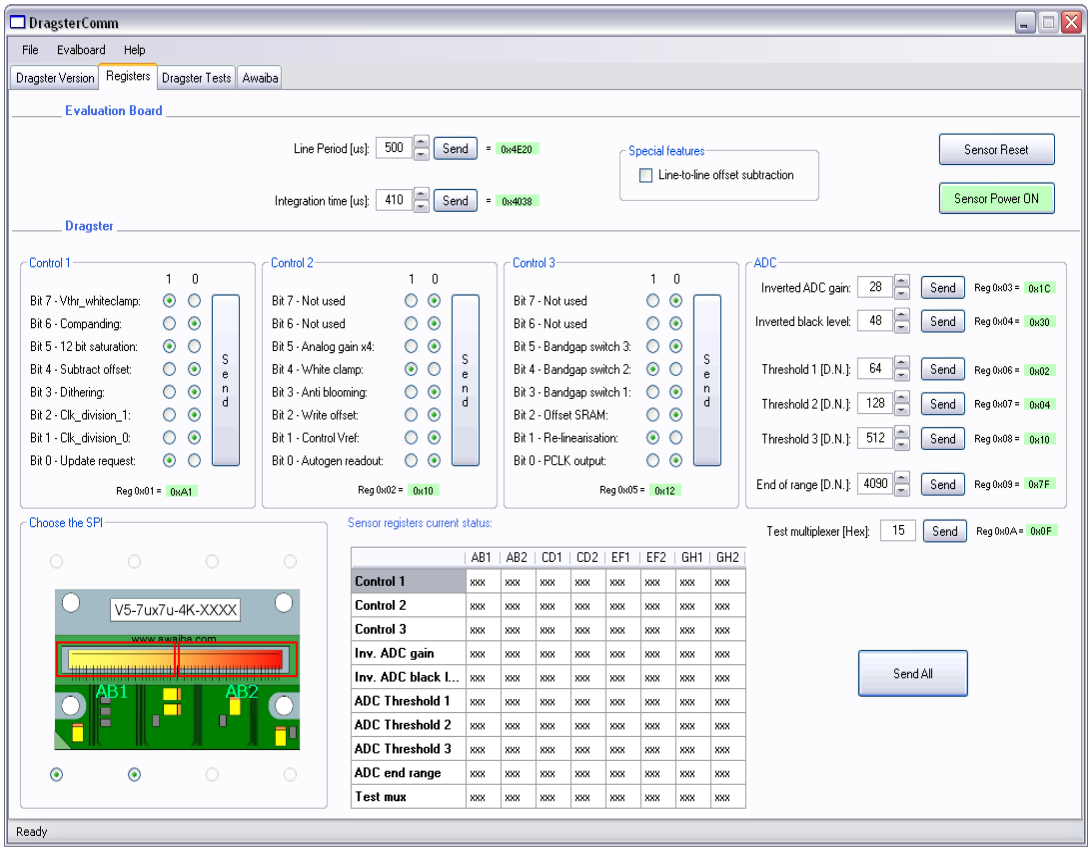


Figure 18: ISAC - Registers Configuration

After defining the register content the hexadecimal representation value is converted and shown with green highlight. In case the input value is not allowed because exceeds register maximum or minimum value or because a wrong character was inserted, a red highlight will appear.

There is no global register validation before sending each register over the serial port so it's user responsibility to ensure that sensor performs right. In case a wrong state is set, a sensor reset is recommended and the button **Reset sensor** must be pressed.

There is also available a **Send All** button, that writes all register's settings one after the other, finishing with the Control 1.

Register values are, by default, send to all SPI's at a time. In case of a fine tune of the sensor, for example a gain or black level tune, a different SPI can be chosen. By clicking the buttons around the Dragster image, all SPI's can be selected/unselected and programmed individually. The last value of the registers changed will be recorded in the status table.

After each **Send** click, the status of the sensor will be kept in the table seen in the middle of the software window. A not programmed or a default value is defined as “xxx”. This table is based in the commands send.

Next Table 8 shows the addresses of all Dragster registers, labelled as shown in ISAC's interface.

Label	Address
Control 1	0x01
Control 2	0x02
Control 3	0x05
Inverted ADC Gain	0x03
Black Level	0x04
ADC Threshold 1	0x06
ADC Threshold 2	0x07
ADC Threshold 3	0x08
ADC End of Range	0x09
Test Multiplexer	0x0A

Table 8: Dragster Registers Labels and Addresses

Important considerations:

- When selecting to write only to one SPI be aware that all settings, including update request will only be sent to the respective sensor half. And when using **Send All** together with **All SPI** the respective settings will again be over written by the activated entries.
- By accessing individual SPI's the sensor can be fine tuned to compensate deviations in gain and black level references along different segments but only small value differences in will lead to useful image data.

8.4.1 Enabling on chip offset subtraction

To enable the on chip offset subtraction a sequence of commands should be sent to the sensor over all SPI's (use the Send buttons of the individual registers):

1. Activate the second bit “Offset SRAM” in Control 3 register. Click “Send”.
2. Set in Control 1 "Subtract offset" bit to 0 and "Update request" to 1. Click “Send”
3. Set the sensor to dark condition and adjust the black level output value around 100DN. Avoid any pixel value clipping to 0DN.
4. Set in Control 2 "Write Offset" bit to 1. Click “Send” and in Control 1 "Update request" to 1. Click Send.
5. Set in Control 1 "Subtract offset" bit to 1 and "Update request" to 1. Click “Send”.
6. Set in Control 2 "Write Offset" bit to 0. Then “Send”.
7. Keep in Control 1 "Subtract offset" bit to 1 and "Update request" to 1. Click “Send”.

8. Reduce the setting in “Black Level” (register 0x04) by one or two counts, to avoid clipping of the noise to 0DN.

Now the dark image will be subtracted on chip from the illuminated image. Avoid for further operation to change the ADC analogue gain, without recalibration of the offset image.

When turning "Subtract offset" bit in Control 1 to 0 the stored offset image is kept saved in the SRAM block, so it doesn't need to be re-acquired.

8.4.2 Relation between ADC gain and ADC end of range

The ADC gain is programmable in a wide range to accommodate the different operation scenarios. Please refer to the ASIC Specification for full details.

The most important relation when adjusting the conversion gain is the relation between "ADC end of range", set over the respective register in multiples of 32 DN, and the register entry of the "inverse ADC gain".

- Too low ADC quantization steps (low values on "inverse ADC gain") will lead to a very high AD conversion gain, covering only a small range of the analogue signal swing, and mainly amplifying noise. The setting of the “Black level” is subject to ADC gain, and thus with a very low value for the ADC quantization step, will be difficult to adjust correctly.
- Too high ADC quantization step (high values on "inverse ADC gain") will lead to the ADC covering an input range exceeding the ADC's linear response range, and may lead to artefacts. If artefacts close to saturation are observed, please reduce the setting in “inverse ADC gain”. The ADC quantization step should therefore be adjusted together with the ADC end of range register as indicated in Table 9.

For linear operation and 10 bit resp. 12 bit digitalisation with the ADC clock set to the same frequency as the readout clock, the following table gives an overview of the ADC gain settings:

	ADC End of Range	
Inversed ADC Gain	0x80 (4096) 12bit	0x20 (1024) 10bit
0x80	x0.25 (do not use)	x1
0x20	x1	x4
0x10	x2	x8
0x08	x4	x16

Table 9: Relation between ADC gain and ADC end of range

8.4.3 Using analogue gain

For high sensitivity, especially at high line rates we recommend to use the on chip analogue gain of x4. The analogue gain performs signal amplification before ADC noise is added, therefore the noise increase due to this additional gain is typically only of x2 while the signal increases by x4.

8.5 Frame Rate

We recommend that Line Period must in any case be longer than the maximum of:

- ADC time (controlled by sensor (register 0x09 * 32) / M_Clock) + 1us
- Integration Time + 2us
- LVAL period (Readout Time = 1040 clks) + 8 + 4

The following table states all relevant timing conditions, depending on the operating frequency and ADC resolution.

Resolution	Frequency [MHz]	ADC Time * [us] + 1us	Readout Time [us]	Min Line Period [us]	MAX Integration Time * [us]	FPS
12 Bit	100	41,96	10,52	41,96	39,96	23832
	80	52,2	13,15	52,2	50,2	19157
	40	103,4	26,3	103,4	101,4	9671
	20	205,8	52,6	205,8	203,8	4859
10 Bit	100	11,24	10,52	11,24	9,24	88968
	80	13,8	13,15	13,8	11,8	72464
	40	26,6	26,3	26,6	24,6	37594
	20	52,2	52,6	52,6	50,6	19011
8 Bit	100	3,56	10,52	10,52	8,52	95057
	80	4,2	13,15	13,15	11,15	76046
	40	7,4	26,3	26,3	24,3	38023
	20	13,8	52,6	52,6	50,6	19011

Table 10: Frame per second description

By activating the **Companding Mode** the ADC conversion time is significantly reduced while no information is lost. ADC thresholds recommended in this configuration are the following.

	Threshold 1	Threshold 2	Threshold 3	ADC End Range	ADC Main Clk Times
Reg Set HEX	1	4	10	7F	620

Table 11: Companding Mode, registers configuration

Resulting then in the following timings.

Resolution	Frequency [MHz]	ADC Time* [us] +1us	Readout Time [us]	Min Line Period [us]	MAX Integration Time* [us]	FPS
12 Bit	100	7,2	10,52	10,52	8,52	95057
	80	8,75	13,15	13,15	11,15	76046
	40	16,5	26,3	26,3	24,3	38023
	20	32	52,6	52,6	50,6	19011

Table 12: Companding Mode, registers configuration

8.6 Configure the ini file

The ISAC tool provides an ini file where is possible to define the values to be loaded by default each time the program runs. The variables should be changed in a common text editor and the file saved before the changes take effect next time the software runs.

8.7 Debug Console

To have some indications about the program execution there is a debug console used as output of several messages. Although these messages have only significant importance in the software development they should be reported to Awaiba GmbH every time there is a bug found in ISAC tool. This will help us to better identify the error cause and to correct the software flaw.

8.8 Known issues

Under some installations of Windows XP some system libraries required to run the application may be missing. If you get an error message when invoking the application please install the "Visual C++ 2005 express edition" from:

http://msdn.microsoft.com/en-us/express/aa975050.aspx

9 SPARTAN 3AN JTAG Programming

The JTAG connector in the Evaluation Board is used to program the SPARTAN 3AN FPGA through a JTAG chain. This AN version has integrated flash memory so you don't need to reprogram it after being disconnected from power supply.

To program the FPGA you need to connect a JTAG programmer and use the Xilinx Impact software. In the Impact software you should:

- Click in **Boundary Scan**
- Right-click anywhere in the canvas and choose the **Initialize Chain** option. This option will show the device that is connected via JTAG (figure 19).
- The user can either choose to load the configuration files, by pressing **Yes**, or, if the message box does not appear, it can choose **Assign new configuration file** option by right clicking in the canvas. Choose the bit file that we sent you.
- Click with the mouse right button in the FPGA icon and choose **Program Flash and Load FPGA**. In other dialogue boxes that might appear you don't need to change any settings.

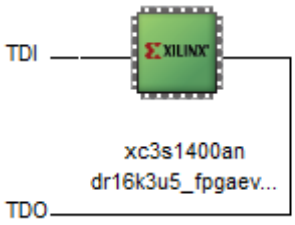


Figure 19: SPARTAN JTAG Chain

To change from one sensor to another you need to turn off the board and restore the power again. The FPGA board will automatic recognize the new sensor and a test image will be displayed in the Framer Grabber program.

End of Document

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

ams:
EVALBOARDDRAGSTER