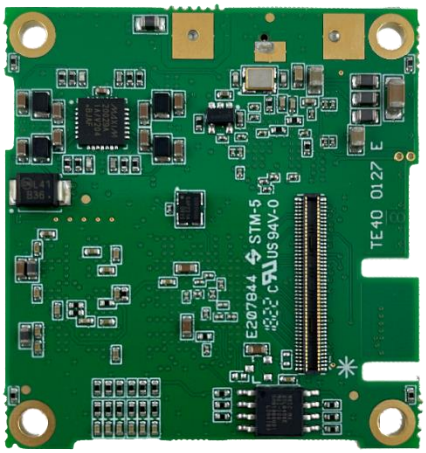
USB3 Neo

Technical Manual



P/N – TV10 0083: USB3 **Neo** interface board for LVDS zoom cameras

P/N – TV50 0025: Mounting kit for TV10 0083 – USB3 Neo I/F board
Includes: 30-way micro-coax camera cable, 2-way cable (power supply), 3-way cable (UART TTL), 7-way cable (GPIOs), right angle black anodized bracket, screws and spacers

P/N – TV50 0026: Cable kit for TV10 0083 – USB3 Neo I/F board
Includes: 30-way micro-coax camera cable, 2-way cable (power supply)

Table of content

Revision History

Date	Revision	Description	Modified by	Note
09/02/22	A	Creation of the document	CBO	
23/08/22	B	Add internal registers	CBO	
07/04/23	C	Update board and kit references	CBO	
27/02/25	D	Change document organization and graphical chart	CBO	
25/08/25	E	Update register table according to v3.0	CBO	

Key features

- Super speed USB 3.2 Gen 1
- Uncompressed HD video stream
- No latency involved by the board
- Video resolution up to 1080p60
- USB Video Class (UVC) compliant
- Communication:
  - USB CDC (COM port / tty)
  - VISCA camera control via local UART
  - GPIOs for keypad connection
- Camera external power supply 6V-12VDC*
- Consumption:
  - Board only 2W
  - Camera only around 3W
- Supports Windows and Linux OS
- USB3 cable length up to 15m or 30m adding boost
- Operating temperature [0°C; 60°C]

*Refer to the camera technical manual for max power supply input

General description

USB3 is the 3rd major version of the Universal Serial Bus standard for interfacing computers and electronic devices. The SuperSpeed USB provides a transfer rate that allows to transmit uncompressed high-definition video signals from camera to a computer.
A simple USB3 cable is enough to power, to control and to stream HD video from the camera. This enables easy implementation of image capture solutions on most devices with a USB3 port (single board computers, NVidia platforms or PC using Windows or Linux OS).

The USB3 Neo converts the native LVDS video signal from zoom bock cameras to USB3 standard. You will take advantage of new visualization, processing, and image transfer solutions by using the USB3 streaming possibilities.

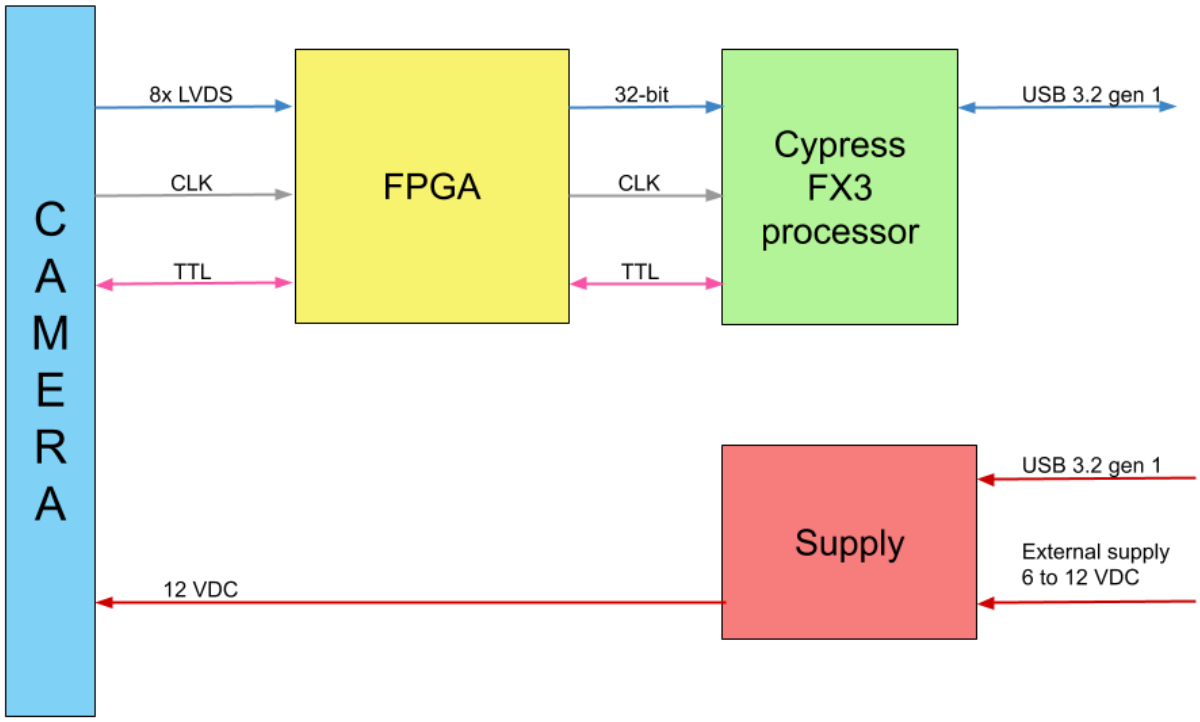
The board provides uncompressed UVC compliant video stream. This enables you to use standard application (i.e VLC, Amcap, Windows camera) for image capture, recording and viewing. It also allows to develop custom application and image processing based on DirectShow, GStreamer or OpenCV.

The add-on connector on the back side offers an infinity of new functionalities like to plug-in additional boards to get more capabilities (i.e 4K camera compatibility), or to implement your own functions (i.e Sync information, strobe pulse, external trigger, accelerometer).

Benefits of this solution

- USB3 standard offering many possibilities
- Plug and play solution, one cable for video and communication
- Ultra-low latency involved by the board (< 1ms)
- Video stream up to 1080p60
- Compatible with Windows and Linux
- Camera external power supply connector to solve power issue
- Always keep up to date with an easy software update
- Addon connector for custom needs and 4K camera support
- GPIOs connector to easily send basic VISCA commands (zoom in / out, freeze on / off, focus)

Block diagram



Video acquisition

The main components are the FPGA and the FX3 for video acquisition.
The board acquires LVDS video from the camera block with no latency deserialization to provide uncompressed UVC video stream available via an USB3 connector.

Communication

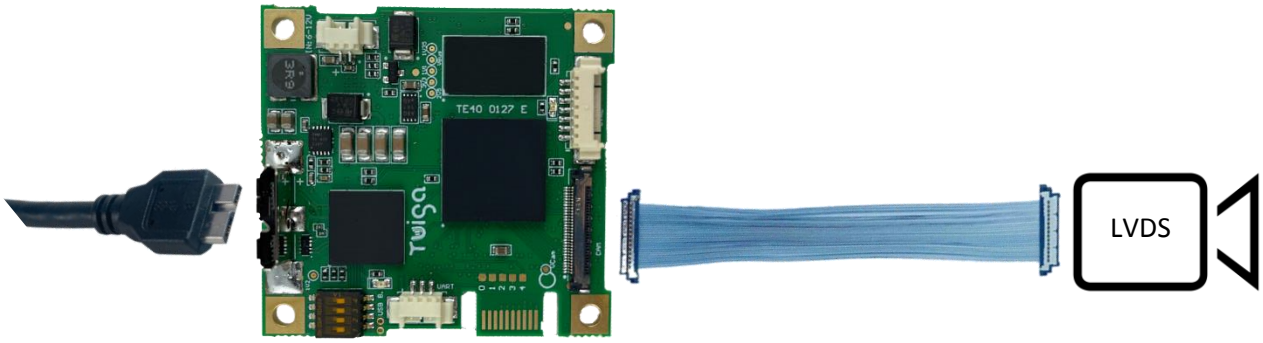
The communication is possible using the CDC (COM port on Windows, tty on Linux) through the FX3 and the FPGA. An UART TTL is available on a 3 points Molex connector directly connected to the FPGA.
A flashing green LED helps to know if an error occurred according to its number of blinks.

Power supply

The power supply is done via the USB3.0. It supplies the camera too.
However, the camera only can be powered via a 2 points Molex connector for external power supply (6 to 12VDC).
The board switches automatically to the camera external power supply if you provide it.

Accessing the video

Quick setup



Installation steps:

1. Connect the KEL cable between the board J203 and the camera.
2. If needed (depending on the camera consumption and on your system power), connect the external camera power supply and power it on.
3. Connect the USB3.0 A type cable between the board J204 and the computer on a USB3.0 port. For tower computer please connect on a back side USB3.0 port.

Computer configuration

The USB3 Neo board has been tested on several Exploitation Systems:

- Windows: 7, 8, 8.1, 10, 11
- Linux: Ubuntu 16.04, Ubuntu 20.04, Fedora 33
- Nvidia: Jetson Xavier NX

On Windows

Several software tools are available to display the video:

- AMCap: in Devices select the Twiga board
- Yawcam: in Settings select the Twiga board
- VLC: in Media select “Open a capture device” and select the Twiga board as Video device name
- Other software tested: Gstreamer, Camera application from Windows, OBS, Debut from NCH Software

On Linux

The board is automatically detected as a video device in /dev/, to find it use the command “ls /dev/”. By default, the device is called video0 but it could have another number if you already have a video device connected or according to the Exploitation System version.
To display the stream, you can use a Gstreamer pipeline:

gst-launch-1.0 v4l2src device=/dev/video0 ! videoconvert ! autovideosink

Video characteristics

Introduction on video formats

You have two video format types:

- Progressive: displays both the even and odd scan lines (the entire video frame) at the same time. The video formats are listed with the letter ‘p’.
- Interlaced: displays even and odd scan lines as separate fields. The even scan lines are drawn on the screen, then the odd scan lines are drawn on the screen. Two of these even and odd scan line fields make up one video frame. The video formats are listed with the letter ‘i’.

Notion of LVDS mode:

- It is controlled by the register 74 of the camera (0x00: Single mode, 0x01: Dual mode).
- It is used to increase the video output from 4x LVDS data lines to 8x LVDS data lines. The output clock frequency is still 74,25MHz but with twice more date lanes.
- It is needed to process video formats 1080p50, 1080p59.94 and 1080p60. If the camera itself does not have 4x additional LVDS data lanes, it will output data at 148,5MHz for video formats 1080p50, 1080p59.94 and 1080p60.

On LVDS Full HD cameras blocks you can have several video formats available:

- Full HD Interlaced 1920x1080i: it can be at 50, 59.94 or 60 FPS, the camera must be in Single mode.
- Full HD Progressive 1920x1080p: it can be at 25, 29.97 or 30 FPS, the camera must be in Single mode. It can also be at 50, 59.94 or 60 FPS, with these video formats only, the camera must be in Dual mode to be able to send more data.
- HD Progressive 1280x720p: it can be at 25, 29.97, 30, 50, 59.94 or 60 FPS, the camera must be in single mode.

LVDS video input supported resolutions

The video format from the LVDS camera can be configured by sending VISCA command using the register 72.

	25	29.97	30	50	59.94	60
1280x720p	✓	✓	✓	✓	✓	✓
1920x1080p	✓	✓	✓	✓*	✓*	✓*

* The video formats 1080p50, 1080p59.94 and 1080p60 require the camera configured in dual lane: register 74 set to 0x01. The others video formats require the register 74 set to 0x00 for single lane.

USB3 video output

The output video stream is using UVC protocol and is recognized as a video device on the computer like a webcam. The video is transmitted via a USB3.0 link up to 1080p60.
If you connect the board to a USB2.0 port a black screen will be displayed.
If the input video format is not supported a black and white pattern will be displayed.

System configuration

Communication

Communication with the camera can be done through 2 connectors:

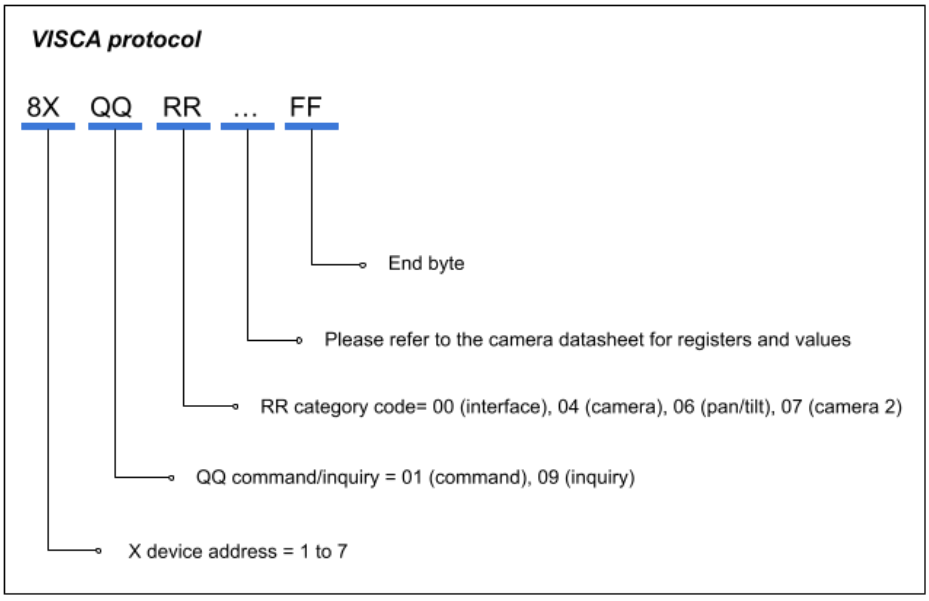
1. USB3.0 connector (J204) using CDC protocol: it allows you to send commands (VISCA) to the camera through the USB3.0 cable. You can change video format, zoom, manage camera parameters such as focus, iris, shutter… You can use basic communication software or specific software according to the camera block you use. You can communicate with internal registers through this connector.

2. UART TTL 3 points connector (J202): you can also use this way to send VISCA commands to the camera or communicate with internal registers. The UART TTL level is 3V3.

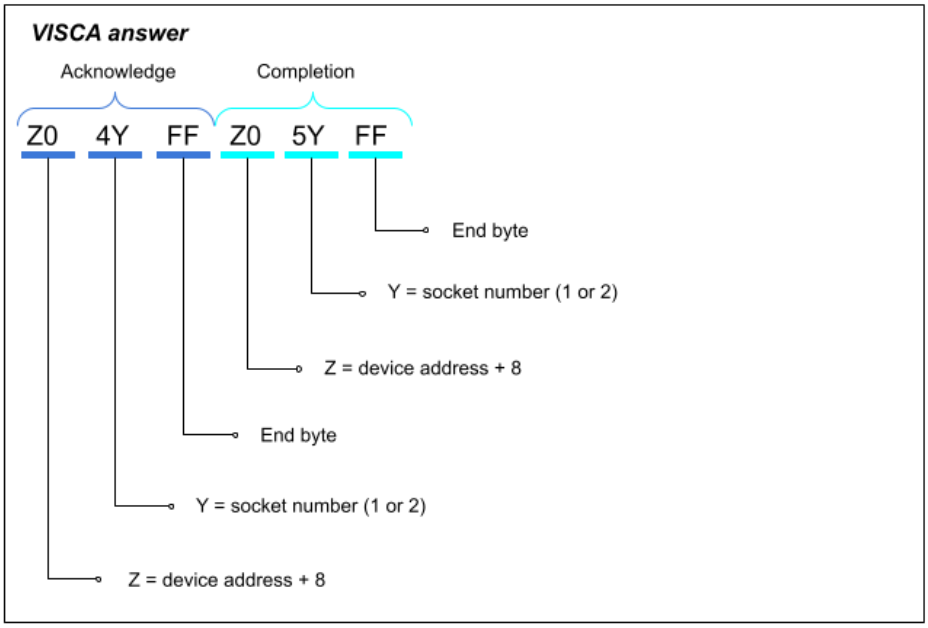
The baud rate of the serial interface will follow the baud rate configured in the camera.

To the camera

The camera communication uses VISCA protocol and will follow camera specifications. It is a standard protocol for camera blocks following this structure:

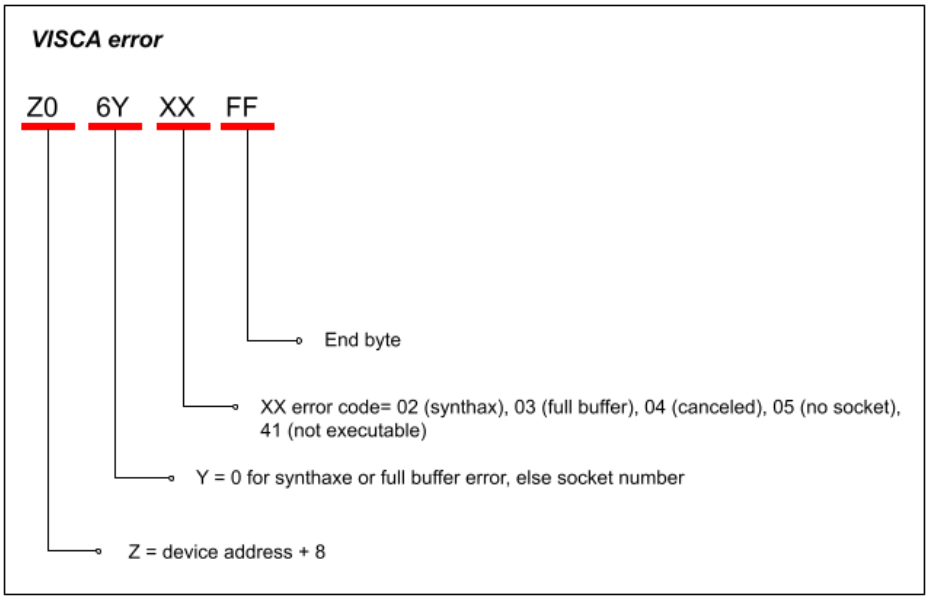


The camera answer follows this structure:



The time between the acknowledgement and the completion packet depends on the command. The answer for an inquiry is Z0 5Y followed by the information requested with FF as end byte.

If an error occurs, here the answer structure:



Example: Zoom In command with a speed of 7 is 0x81 01 04 07 27 FF and the expected answer is 0x90 41 FF followed by 0x90 51 FF.

You can communicate using communication software like Termite or the camera brand communication tool like Sony FCB Control software.

To the internal registers

You can access to internal registers via both communications (CDC and UART TTL) with VISCA protocol at Address 0x82.

Register name	Addr	Saved	Set command	Comments	Inquiry command	Inquiry answer
REG_FPGA_VERSION_H	0x01	No	NA	Software version MSB	0x82 09 06 01 FF	0xA0 50 01 00 0y FF
REG_FPGA_VERSION_L	0x02	No	NA	Software version LSB	0x82 09 06 02 FF	0xA0 50 02 00 0y FF
REG_FPGA_TEMPERATURE	0x04	No	NA	See Annex 1	0x82 09 06 04 FF	0xA0 50 04 0y 0y FF
REG_FPGA_REBOOT	0x05	No	0x82 01 06 05 00 01 FF	Reboot the FPGA	NA	NA
REG_FPGA_RUNTIME_H	0x0A	No	NA	Runtime High	0x82 09 06 0A FF	0xA0 50 0A 0y 0y FF
REG_FPGA_RUNTIME_MH	0x0B	No	NA	Runtime Middle High	0x82 09 06 0B FF	0xA0 50 0B 0y 0y FF
REG_FPGA_RUNTIME_ML	0x0C	No	NA	Runtime Middle Low	0x82 09 06 0C FF	0xA0 50 0C 0y 0y FF
REG_FPGA_RUNTIME_L	0x0D	No	NA	Runtime Low	0x82 09 06 0D FF	0xA0 50 0D 0y 0y FF
REG_DBG_LED	0x12	Yes	0x82 01 06 12 0y 0y FF	0x00: LED OFF 0x01: LED ON	0x82 09 06 12 FF	0xA0 50 12 00 0y FF
REG_CONFIG_SAVE	0x13	No	0x82 01 06 13 00 01 FF	Save registers values to flash	NA	NA
REG_CONFIG_LOAD	0x14	No	0x82 01 06 14 00 01 FF	Reload registers values from flash	NA	NA
REG_CAM_VENDOR_ID_H	0x20	No	NA	Camera vendor ID MSB	0x82 09 06 20 FF	0xA0 50 20 0y 0y FF
REG_CAM_VENDOR_ID_L	0x21	No	NA	Camera vendor ID LSB	0x82 09 06 21 FF	0xA0 50 21 0y 0y FF
REG_CAM_MODEL_ID_H	0x22	No	NA	Camera model ID MSB	0x82 09 06 22 FF	0xA0 50 22 0y 0y FF
REG_CAM_MODEL_ID_L	0x23	No	NA	Camera model ID LSB	0x82 09 06 23 FF	0xA0 50 23 0y 0y FF
REG_CAM_UART_BYPASS	0x24	No	0x82 01 06 24 0A 05 FF	Bypass the communication for camera update *	NA	NA
REG_CAM_FORCE_FMT	0x25	No	0x82 01 06 25 00 01 FF	Format the camera format to 1080p30 single lane	NA	NA
REG_CAM_DETECT_RESET	0x26	No	0x82 01 06 26 00 01 FF	Reset camera detection	NA	NA
REG_VIDEO_DETECTED	0x28	No	NA	0x01: video detected Other: video not detected	0x82 09 06 28 FF	0xA0 50 28 00 0y FF
REG_ZOOM_IN_SPEED	0x2A	Yes	0x82 01 06 2A 00 0y FF	GPIO zoom in speed configuration from 0 to 7	0x82 09 06 2A FF	0xA0 50 2A 00 0y FF
REG_ZOOM_OUT_SPEED	0x2B	Yes	0x82 01 06 2B 00 0y FF	GPIO zoom out speed configuration from 0 to 7	0x82 09 06 2B FF	0xA0 50 2B 00 0y FF
REG_USB_MODE	0x30	Yes	0x82 01 06 30 00 0y FF	0x00: Isochronous 0x01: Bulk	0x82 09 06 30 FF	0xA0 50 30 00 0y FF
REG_USB_REG_OR_SW	0x31	Yes	0x82 01 06 31 00 0y FF	0x00: from register 0x01: from DIP switch 4	0x82 09 06 31 FF	0xA0 50 31 00 0y FF

* Note: The board needs to be restarted after the camera update to enable the communication again. Be sure that your camera can be updated only using the Kel cable (no need to connect any FFC cable).

Vendor and Product IDs can be changed by internal registers, such as the serial number and the device name. Please contact us for more information if you need to change these parameters.

Computer configuration

On Windows

If your Windows is older than Windows 10 please ask for the driver installation procedure. The board will be automatically detected as a COM port. You can use Termite or putty to open the communication with the dedicated COM port and send VISCA commands. According to your camera brand you can use the specific tool provided by the camera manufacturer like Sony FCB control software.

On Linux

The communication port is recognized on Linux as a /dev/ttyACMx device, where "x" is the number of the device.
When a *ttyACM* device appears on Linux, some daemons software analyze it and it could not be available for about 10sec. After that the device is released and the communication works as a COM Port on Windows.

You can use a serial communication terminal to send commands to the camera (e.g. gtkterm).

You can also use the "**echo**" command with a Linux terminal:

- Configure the tty with the correct baud rate: ***stty 9600 -F /dev/ttyACMx***
- For checking the configuration use: ***stty -a -F /dev/ttyACMx***
- Send commands like zoom plus as it ***echo –en '\|x81\|x01\|x04\|x07\|x02\|xff' > /dev/ttyACMx*** where x is the device number and 8101040702FF the zoom in command in hexadecimal

GPIOs

Six GPIOs are available on J200 connector, each one is dedicated to a specific camera function:

Pin	Action	Control	VISCA command sent
Ctrl1	Press Release	Zoom + Zoom stop	0x81 01 04 07 23 FF 0x81 01 04 07 00 FF
Ctrl2	Press Release	Zoom – Zoom stop	0x81 01 04 07 33 FF 0x81 01 04 07 00 FF
Ctrl3	Press Release	Focus Auto / Manual	0x81 01 04 38 10 FF
Ctrl4	Press Release	Focus near Focus stop	0x81 01 04 08 33 FF 0x81 01 04 08 00 FF
Ctrl5	Press Release	Focus far Focus stop	0x81 01 04 08 23 FF 0x81 01 04 08 00 FF
Ctrl6	Press Release	Image freeze toggle	0x81 01 04 62 02 FF 0x81 01 04 62 03 FF

To activate it you need to connect the pin to the ground. ESD filters and anti-bounce have been added. You can use existing keyboard to easily control them.
Note that zoom in and out speed is 3 by default but can be modified by internal registers.

LED signalization

Number of Fx3 Status LED blink per 2 seconds	Meaning
1	Error
2	Communication error
3	Format error
4	Configuration ok

Number of FPGA Status LED blink per 3 seconds	Meaning
2	FX3 not detected
4	Video error
6	Configuration Ok

Multi switch

Switch number	Meaning
1	Unused
2	Unused
3	Unused
4	Transmission mode : OFF = Bulk mode (default), ON = Isochronous mode

The DIP switch SW800 is used with the switch n°4 to select the data transmission mode between Bulk (default) and Isochronous. The change is active after a restart of the board only.

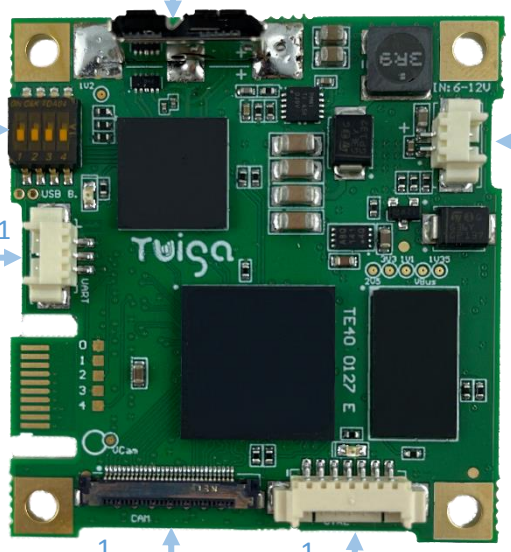
Please check the internal registers *REG_USB_REG_OR_SW* to be sure that the priority is given to the DIP switch 4 and not to the internal register *REG_USB_MODE*.

- **Bulk mode:** for transmission that requires guaranteed delivery without specific bandwidth or latency requirements. In the absence of any other demands for bus bandwidth, bulk transfers are processed as quickly as possible.

- **Isochronous mode:** provides guaranteed amounts of bandwidth and latency, used for real-time applications. For isochronous transfers, timely data delivery is much more important than perfectly accurate or complete data transfer.

Connectors

J204	USB3.2



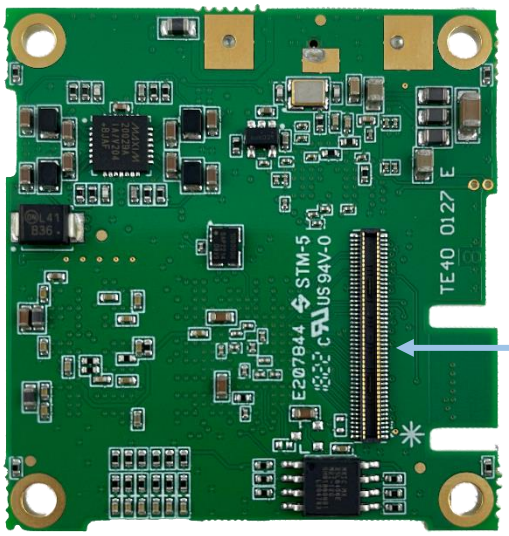
DIP switches				Configuration
1	2	3	4	
/	/	/	**OFF**	Bulk
/	/	/	**ON**	Isochronous

J202	UART TTL
1	GND
2	Rx
3	Tx

J203	LVDS input
1	TX4-
2	TX4+
3	TX5-
4	TX5+
5	Reset
6	NC
7	TX6-
8	TX6+
9	TX7-
10	TX7+
11	GND
12	GND
13	VCAM
14	VCAM
15	VCAM
16	VCAM
17	VCAM
18	RxD (TTL camera input)
19	TxD (TTL camera input)
20	GND
21	TX0-
22	TX0+
23	TX1-
24	TX1+
25	TX2-
26	TX2+
27	TXCLKOUT-
28	TXCLKOUT+
29	TX3-
30	TX3+

J300	External Camera Power supply
1	GND
2	V+

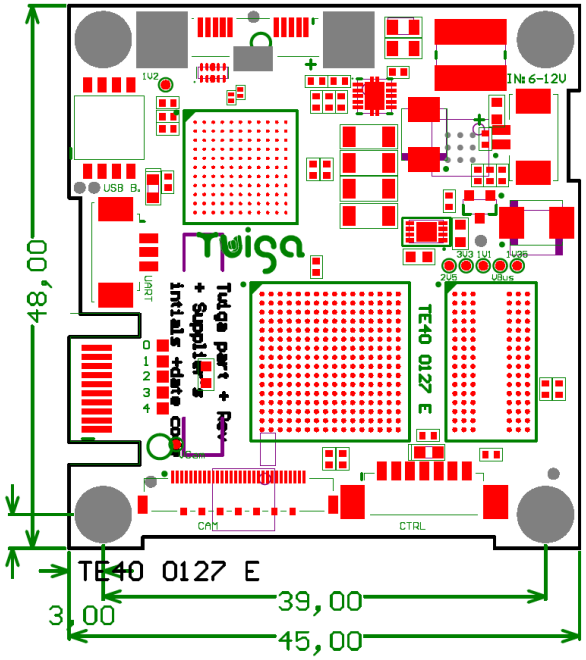
J200	GPIOs
1	GND
2	Zoom +
3	Zoom -
4	Focus Auto/Manual
5	Focus Near
6	Focus Far
7	Freeze On/Off



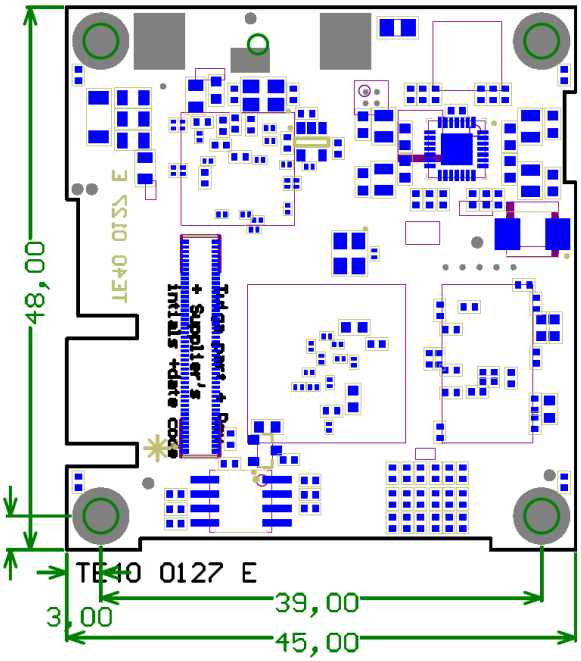
J201	Board to Board connector
Extended functionalities	

Form factor

TOP



BOTTOM



48mm (H) x 45mm (W) x 11mm (D)

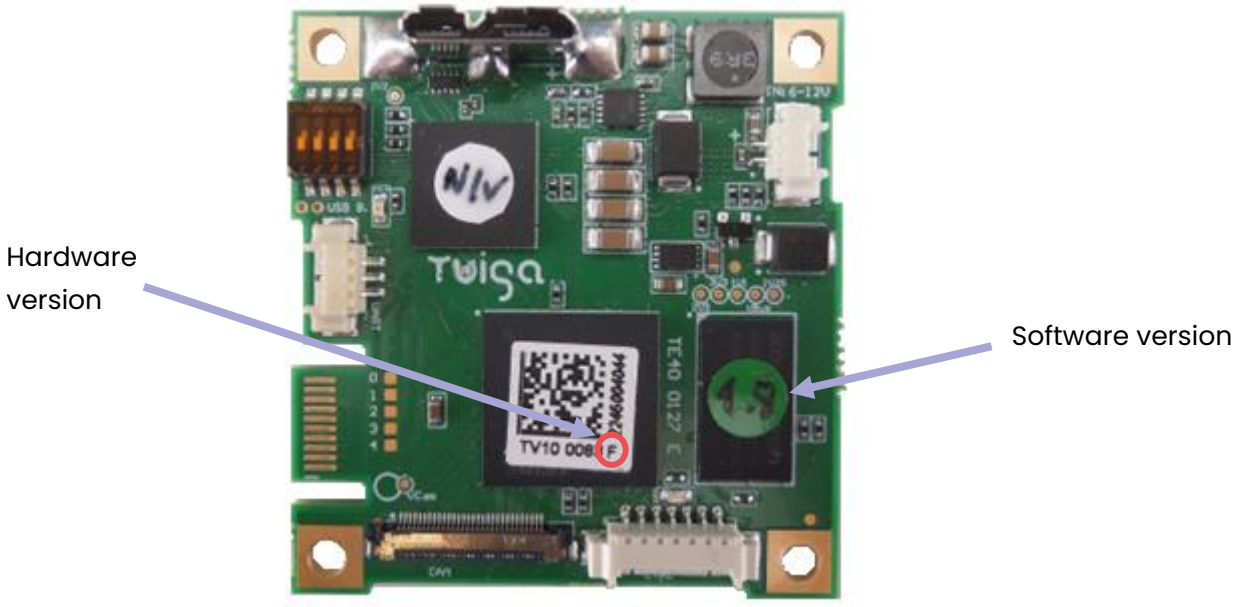
4 holes Ø 3mm

13g

Troubleshooting

Get hardware and software version

The hardware version is a letter written close to the reference of the board TV10 0083. The software version is written on a green sticker stuck on the top side of the board. Be careful, USB3 Neo can be updated by the customer, in this case the green sticker could be at a wrong version. You can still read internal registers to get the correct version.



Update via UART

An update of the board is possible by distance, you need an UART connection with the board and a Java application made by i2S.
If you do not have this tool, please send us a mail at info@i2s.fr specifying which board you are working with. This way we will give you the right tool to perform the software update and the changes involved by the different software.

Common issues

If you have any problem getting the video, here some points you need to check:

- Power supply is correctly connected to the board, no consuming issue or overheating of the board.
- If the camera restarts in loop, in this case the power coming from your USB3 port is not enough to power supply the board and the camera. Please use the 2 points Molex connector for external camera power supply (12V, 1,5A).
- If the video is a black screen, it could mean that your board is recognized as USB2.0 device. Please use USB3.0 cable and USB3.0 port, if it is the case try with another USB3.0 cable.
- The video format of the camera is correct and supported by the board, if not you will see a black and white pattern. If the video format is correct, try another Kel cable. Maybe try with another LVDS compatible camera to be sure the issue is not coming from the camera.
- The LVDS mode of the camera (register 74) is adapted to your video format: dual mode (value 0x01) for 1080p50, 1080p59.94 and 1080p60, or single mode (value 0x00) for other video formats.
- If you have random freeze or video loss very often and only a reboot of the board helps, please try setting the isochronous transmission mode. You probably have an environment with EMC perturbations which is not correctly manage on all exploitation systems in bulk transmission mode.

If you are not able to find the cause of the issue, please contact us at info@i2s.fr and we will give you support. Explain us the problem you are facing with as much details as possible and please add the hardware and software version of your board.

Annex

Annex 1: FPGA temperature table

Here the table to get the FPGA temperature (°C) from the value read in the register 0x15.

Register value read	FPGA temperature (°C)
0x00	−58
0x01	−56
0x02	−54
0x03	−52
0x04	−45
0x05	−44
0x06	−43
0x07	−42
0x08	−41
0x09	−40
0x0A	−39
0x0B	−38
0x0C	−37
0x0D	−36
0x0E	−30
0x0F	−20
0x10	−10
0x11	−4
0x12	0
0x13	4
0x14	10
0x15	21
0x16	22
0x17	23
0x18	24
0x19	25
0x1A	26
0x1B	27
0x1C	28
0x1D	29
0x1E	40
0x1F	50
0x20	60
0x21	70
0x22	76
0x23	80
0x24	81
0x25	82
0x26	83
0x27	84
0x28	85
0x29	86
0x2A	87
0x2B	88
0x2C	89
0x2D	95

0x2E	96
0x2F	97
0x30	98
0x31	99
0x32	100
0x33	101
0x34	102
0x35	103
0x36	104
0x37	105
0x38	106
0x39	107
0x3A	108
0x3B	116
0x3C	120
0x3D	124
0x3E	128
0x3F	132