

The Camera Link camera interface

[Introduction](#) -- [Electrical](#) -- [Channel Link](#) -- [Data transmission](#) -- [FPGA notes](#) -- [External links](#)

A brief technical description with notes on an FPGA grabber implementation

Introduction

Camera Link is a standard interface used in industrial cameras. It consists of a number of serial links using differential signaling. Depending on the amount of data to be transferred, one or two cables can be used per camera.

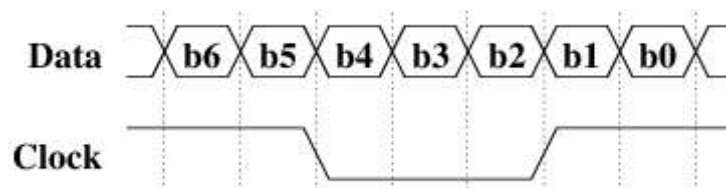
The Automated Imaging Association, the creators of the Camera Link standard, seem to invest some energy into removing hyperlinks from Wikipedia and pressuring members to remove the standard from the web. The resulting scarcity of readily available information on Camera Link was the main reason for me to write this guide.

Electrical

The electrical standard used by Camera Link is LVDS, low voltage differential signaling. Every signal is carried by two wires, with a voltage difference of 0.35 V. When the nominally positive wire has a higher potential than the negative one, this represents a high logic level, when the negative wire has a higher potential, a low.

The "link layer" - Channel Link

The serial links of which Camera Link is composed follow the Channel Link design. Seven bits are transmitted per link, MSB first. The accompanying serial frame clock has a duty cycle of 4:3 and has its falling edge after the second bit (bit 5).



Channel Link transmitter and receiver chips typically handle three or four serial data links together with a common clock. Camera Link uses four per chip.

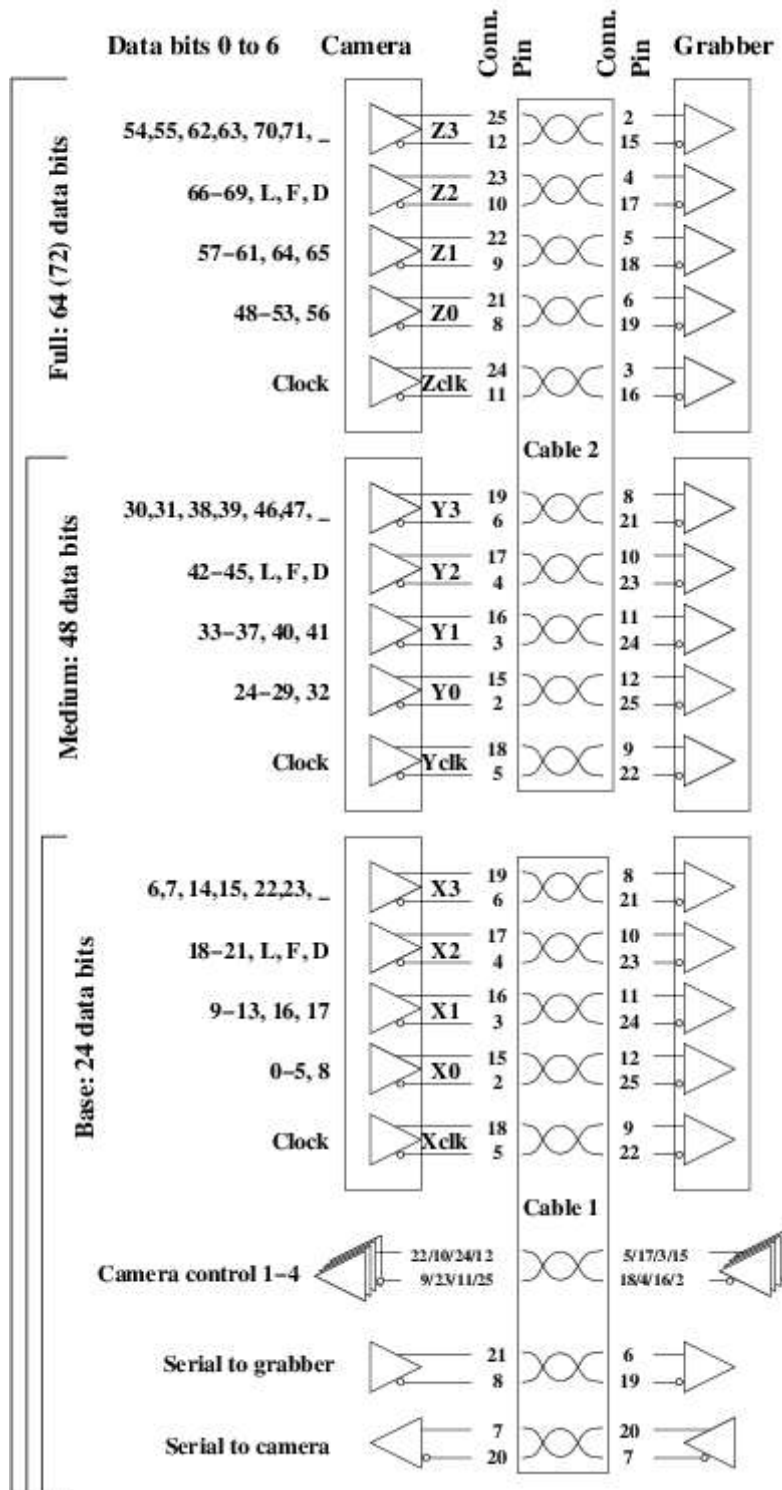
Channel Link recommends twinax (shielded pair) or twisted pair cable for transmission. I have constructed a cable for a camera which was not Camera Link compatible but did use Channel Link out of old Cat5 networking cable. If you want/have to patch together your own cable for Camera Link transmission, this is worth a try.

For more information on Channel Link, read the [Channel Link Design Guide](#) from National Semiconductors, most of which is also relevant for Camera Link.

Camera Link uses serial frame clocks between 40 and 85 MHz, even though some Channel Link chips support higher frequencies.

Data transmission

The following complex diagram shows how Camera Link is built up from between one and three Channel Link transmitters and receivers. Click on it to view a scalable SVG image. **Important: Before February 2018, the bit assignments in this diagram were wrong!**



Three configurations

Camera Link comes in three variants, Base, Medium and Full. The Base configuration uses only one 4-channel Channel Link transmitter (the one at the bottom) and requires only one cable. Medium uses two transmitters and receivers, Full uses three, and both require two cables.

Ports, taps and bits

Each chip can transmit $4 \times 7 = 28$ bits per clock cycle. Each transmits a frame sync, line sync and data valid bit, and one bit remains unassigned. This gives 24 pixel data bits per transfer for the Base configuration and 48

bits per transfer for the Medium configuration. The maximum for the Full configuration would be 72 bits per transfer, but the standard limits this to 64.

For reasons that escape me, the data bits are not assigned to the serial links in sequence, but are heavily permuted within each transmission chip. (This may help lower crosstalk or peak driver power when neighbouring serial links toggle simultaneously.) The numbers on the left of the diagram list which data bits are transmitted on each link, from the lowest-valued link bit (b0 **above**) to the highest. F, L and D stand for the frame valid, line valid and data valid bits, and the underscore denotes the spare bit.

In the standard, the 64 possible data bits are divided into eight 8-bit "ports", which everyone else calls "taps". They are denoted by capital letters starting with port A, which contains bits 0 to 7. The standard prescribes exactly how pixel data are assigned to the ports. When the values of pixels (or colour channels) are a multiple of 8 bits wide, they are assigned to the ports in order. For example, 16-bit pixel values would use ports A and B, assigning bits 0 to 15 without permutation. 24-bit RGB pixels use port A (red), B (blue) and C (green).

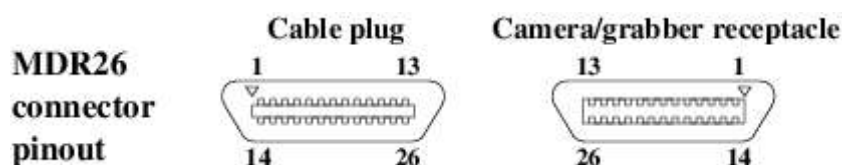
Values with fractional byte widths are split up. For 12-bit values, the lower 8 bits are assigned to ports A, C, E and D, respectively and in that order. Ports B and F contain the remaining 4 bits of each value, in order, starting with those from the first value in bits 0 to 3 of port B (or bits 8 to 11 by my global count). Pixel bit widths of 10 and 14 are upward compatible to 12 and 16, respectively.

Lastly, it should be mentioned that some cameras and frame grabbers claim to support 10-tap, 80-bit transmissions via Camera Link. As I have not got my hands on such a device, I can only speculate how they work. I assume that the bits I denote with 64 to 71 are used in violation of the standard, and the redundant F, L, D and spare bits on the chips Y and Z are replaced by data.

Pins and cable direction

The diagram lists the connector pin numbers of all wires at the camera and grabber end, respectively. Beside the data links, there are four camera control signals and a bidirectional serial interface. The pins of the camera control lines are given in the order from CC1 to CC4. The serial interface serves to adjust camera settings and is RS232 compatible except for the electrical standard, which is LVDS.

Below is the pinout of the normal-sized connectors used for Camera Link. Pin 1 is usually marked with a small triangle. These connectors are called MDR26, for 26-pin Mini Delta Ribbon. A smaller connector is also allowed by the standard, called Mini-Camera-Link or SDR26 (Shrunk Delta Ribbon).



The non-data connector pins have the following functions:

Camera pin	Grabber pin	Signal	Comment
1, 13, 14, 26	1, 13, 14, 26	Inner shield	Digital ground in camera
1, 26	1, 26	PoCL: +12V	Power over Camera Link $\leq 4W$
7, 20	20, 7	Unused in cable 2	Terminate with 100 Ω

An observant reader will have noticed that two wires belonging to the same signal always always face each other on opposite sides of the connector (≤ 13 resp. > 13), and that their sides are reversed at the other end of the cable. This is designed to prevent the skew which would otherwise result from the two rows of a PCB connector being attached by leads of different lengths (see the [Channel Link Design Guide](#) page 6).

Purely from the standpoint of electrical connection, Camera Link cables are symmetrical. However, if you reverse the direction of a cable used for the Base configuration, the wires previously used for a serial interface will carry the data link X3 and therefore much higher frequencies. Some cable manufacturers cut costs by not shielding the serial interface wires as well as the data link wires. Those cables cannot be reversed, and their camera and grabber ends are normally marked as such. For related reasons, those cables cannot be used as a second cable in a Medium or Full configuration. If you want to buy a cable for that purpose (or do not want to bother about matching ends), do not forget to specify that.

Notes on an FPGA grabber implementation

Building an integrated circuit to receive data via Camera Link is relatively straightforward. Other than undoing the bit permutations, all it takes is a deserialiser. The serial frame clock could be multiplied by 7 to register the incoming bits, but multiplying by just 3.5 and using DDR flip-flops is probably a better idea.

The main non-trivial task is finding the correct alignment, both of the sampling point for the bits and the first bit of each 7-bit word. Neither is usually hard-coded (though you could probably do that if you are building a system with a specific camera). The precise sampling point can be adjusted by a variable phase shift in the component which multiplies the clock.

Determination of the bit alignment is based on the correlation of the data bits with the clock (see [diagram above](#)). It *should* really be possible to constrain the correct alignment, but there seems to be no way to impose that particular constraint on the tools. The crux of the matter is that clock and data nets are separate in FPGAs and that we need the serial frame clock as both a clock (for multiplication and sampling) and as data for the bit alignment comparison. This seems to introduce a significant delay, which the vendor tools must be aware of but do not allow constraints to compensate. As a consequence, an adjustable shift between the sampled clock and the data bits is implemented, which has to be varied by software (or a state machine) until valid data are received.

To determine the alignment of the sampled serial frame clock itself, it is simplest to compare it with a 7-bit word containing 4 high and 3 low bits. This ensures that alignment happens exactly once in each data word for a standard conformant clock. I have seen one camera with an apparently non-standard transmitter chip which did not reliably produce a clock duty cycle of 4:3. To support it, I matched the rising clock edge only, and added a state bit to prevent reading the serial shift register multiple times during each word.

External links

- [Camera Link Technolog Brief](#) from Basler
- [Channel Link Design Guide](#) from National Semiconductors
- A vendor of Camera Link signal boosters includes a diagram of maximum standard cable lengths at different frequencies in [their data sheet](#)

Licensed under the [Creative Commons Attribution-Share Alike 3.0 Germany License](#)



[TOS / Impressum](#)