Great River Technology
6121 Indian School Rd. NE Ste 141
Albuquerque, NM 87110
www.greatrivertech.com

White Paper

Best Practices for Implementing a HOTLink Video Protocol for IR and Optical Applications

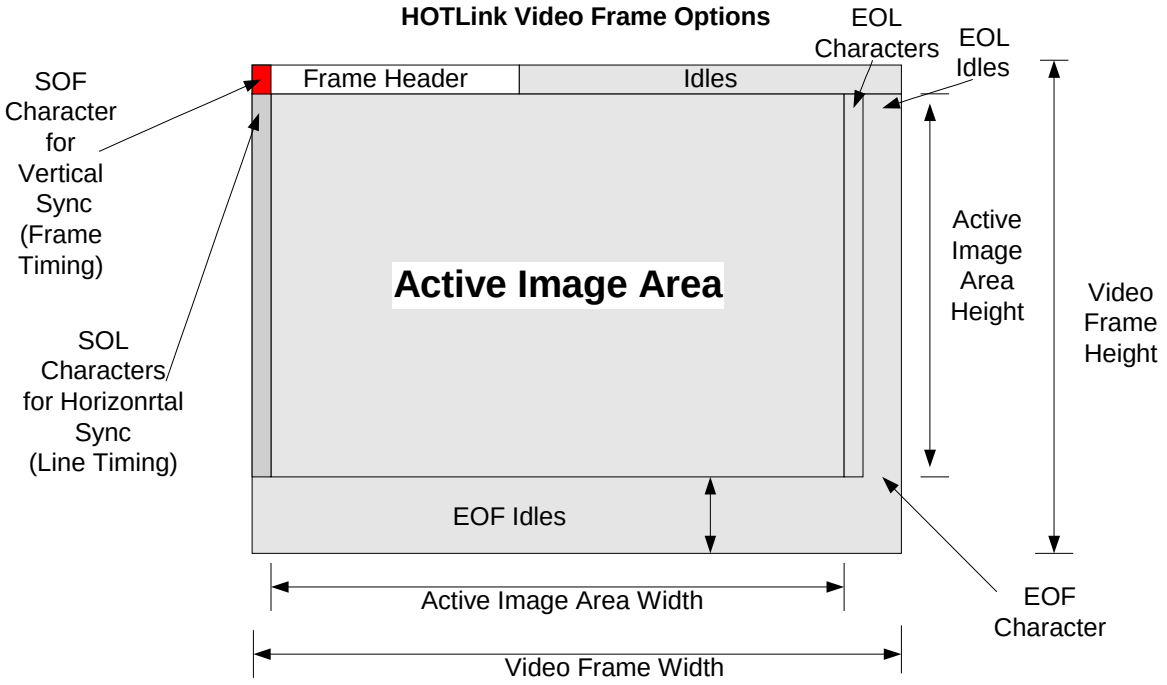
August 30, 2005

Introduction

HOTLink has some features that make it the interface of choice for digital, point-to-point, uncompressed video transmission: two wire interface, good noise immunity, low latency, and low protocol overhead and complexity. Many infrared cameras use HOTLink as their digital output, for example, the Phoenix Indigo FLIR, WESCAM MX-15 Imaging Turrent, and many others. The HOTLink chips are made by Cypress. HOTLink is an 8B/10B physical layer with special characters defined, but no HOTLink protocol standard exists. The fact that no standard exists gives designers ultimate flexibility; however, following some simple design guidelines will insure a robust design. This paper offers some protocol guidelines based on years of experience interfacing to numerous HOTLink sources.

Overview

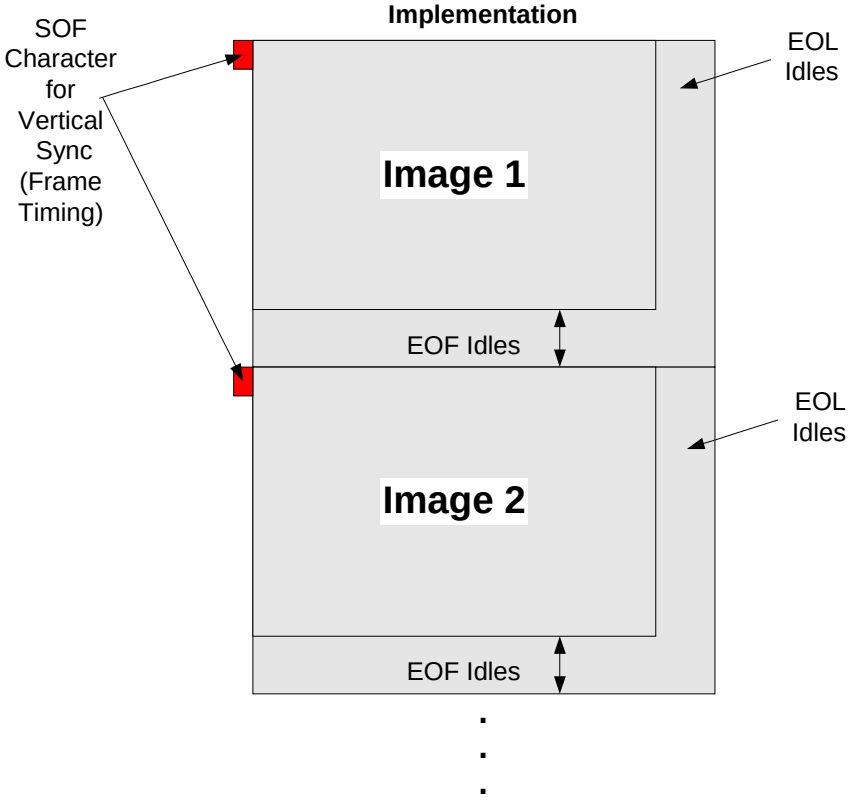
This paper will define the use of features/character sets built into the Cypress to create a simple and robust video link. Special characters (as distinguished from pixel data) can be used to indicate the start of frame (SOF), Start of Line (SOL), End of Line (EOL), End of Frame (EOF), and Idles (EOL or EOF). Figure 1 below shows how special characters can be used to delineate a video frame, and typical characteristics associated with a video frame. Figure 1 shows a maximum set of framing option; however, robust transmission can be achieved with a much smaller set of framing characters.



Minimum Implementation

A minimum implementation is shown in Figure 2 below. Only five things are required to use this scheme: Frame Size (#columns & #rows), pixel depth (bits/pixel), SOF character, EOL Idles, and EOF idles.

Figure 2: Minimum HOTLink Video Implementation



Special Characters: When using 8B/10B encoding, there are two types of characters, data characters, and special characters. When a character is converted from 8 bits to 10 bits, the serializer must be told if a byte is Data or a Special Character (by setting the SC/D pin HIGH on the HOTLink chip)**.** Common special characters are: K28.0, K28.1, K28.2, K28.3, K28.4, K28.5, K28.6, K28.7, K23.7, K27.7, K29.7, and K30.7. These characters can be used to indicate Start of Frame, End of Frame, Start of line, End of Line, Idle or other needed control characters. The "K" designation as in "K28.0" comes from the technical history of 8B/10B encoding, and the names are unimportant. A K28.0 character is a 00H (together with the SC/D pin = HIGH), and each character in the list above increases by 1, so the K30.7 character is 0BH.

Special Character Designations: Any of the special characters (except K28.5) can be used to define any of the video framing functions (SOF, SOL…). The K28.5 character is ***ALWAYS*** an IDLE or NULL character.

Recommended Designations:

Start of Frame (SOF): K28.0
Start of Line (SOL): K28.1

End of Line (EOL): K28.2
End of Frame (EOF): K28.3
All IDLEs or NULLS: K28.5

Design Guidelines:
Long word boundaries: Any video source that may be captured to a PC must observe long word (32 bit) boundaries, that is, data size in bytes must be divisible by 4 to move it over the PCI bus.

Interline Idles: Six idle characters (K28.5) are needed to safely maintain synchronization between transmitter and receiver. This is needed due to the tolerance on the oscillators between the transmitter and receiver. A receiver will typically use an elastic buffer to derive a clock signal based on the incoming data. For data steams longer than several thousand bytes, synchronization can be lost.

Interframe Idles: A minimum of six idle characters are needed, but the time between the end of the frame and the start of the next frame is filled with idle characters, so typically, there are thousands of interframe idles.

Protocol Example:
A 640 x 480, 8 bit mono IR camera at 60 Hz with line timing of 31.496kHz on a 240Mbps physical link.

Given a line timing of 31.496Khz (31.75us/line), the number of bytes per line is: 240Mbps/31.496 Khz = 7620bits/line, where each 8 bit byte is 8B/10B encoded as a 10bit byte, so the number of bytes/line = 762.

Total Number of idle characters per line = 762-640 = 122.

Total number of idle characters between frames =
Frame time = (line time * # lines)
16.67ms = (31.75us *Total #lines)
Total #lines = 525
idle lines = 525 – 480 = 45 lines
45 lines = 34,290 bytes

Figure 3: Example HOTLink Video Implementation