Application Note

PCIe8 DV C-Link

Using the Timestamp Function

For IRIG-B time code input

Rev. 2010 April 09

Engineering Design Team (EDT), Inc.
1400 NW Compton Drive, Suite 315
Beaverton, OR 97006
p 503-690-1234 / 800-435-4320
f 503-690-1243
www.edt.com

Contents

PCIe8 DV C-Link – Using the Timestamp Function

Overview

The timestamp function on the EDT PCIe8 DV C-Link board enables a precise IRIG-B timestamp to be inserted, directly into the data, for each image frame at the moment of capture.

This timestamp is included in a 32-byte footer after the end of the image data; no image data is overwritten. For example, a camera with 1000 x 1000 8-bit pixels would produce 1,000,032 bytes of DMA data.

Although the footer is at the end of the DMA data, the time is captured at the beginning of the frame on the rising edge of the frame valid signal. The time thus captured combines the time derived from an IRIG-B input with a high-resolution counter for computing fractional seconds.

NOTE EDT's current IRIG-B format (IRIG2) is the second such format used for the PCIe8 DV CLink board. EDT's former IRIG-B format (IRIG1) is now obsolete.

In addition to the IRIG time value, the footer contains a frame counter, as well as space for a 64-bit timestamp computed as Unix seconds and fractional seconds.

Setting Up the Board

The IRIG-B function is supported by an EDT-built cable (part number 016-02894). This cable is built from two purchased cables and wired as shown in Figure 1.

Figure 1. EDT-built cable (p/n 016-02894-00)

Notes

Lemo pin 5 (white wire) to center conductor of coax cable
Lemo pin 7 (black wire) to shield of coax cable

2-ft. Lemo cable (016-02650)

Lemo
1 = green
2 = red
3 = orange
4 = brown
5 = white
6 = blue
7 = black

72-in. BNC cable, male to male (016-02735), cut in half to create the EDT-built cable

Camera Configuration Directives

To enable the IRIG timestamp, add this line to a camera configuration file:

```
method_header_type: IRIG2
```

To enable packed BCD timestamps in the configuration file, use:

```
irig_raw: 1
```

To specify that this board is using input from another board rather than direct input, use:

```
irig_slave: 1
```

Footer Format

The IRIG2 footer is 32 bytes appended to the image data transferred by DMA from the PCIe8 DV C-Link board. It includes a frame counter and a timestamp from IRIG input.

Time values down to the microsecond are counted using the 40 MHz oscillator on the board. The time is latched at the beginning of transfer from the camera when the board sees the rising edge of frame valid.

After frame capture, the time in Unix seconds is computed, including the fractional time determined by dividing the current 40 MHz count by the number of 40 MHz clocks in a second. The IRIG time is determined on the board as either Unix seconds, or as the IRIG BCD values packed into a 32-bit integer.

Table 1 shows the elements of the IRIG2 footer.

Table 1. Elements of IRIG2 Footer – part 1 of 2

Structure element	Offset	Size	Type	Notes
Magic	0	4	u_int	This should be ASCII “EDT” followed by 01 (or 0x45445401).
Frame	4	4	u_int	This is the frame counter, which is reset when `reset_intfc bit` is cleared – normally at the start of acquisition.
IRIG time	8	4	One of two types: - packed raw BCD; or - Unix seconds. Type is indicated by a bit in the status register.	For IRIG time, the raw format is: 6 bits seconds 6 bits minutes 5 bits hours 9 bits days 6 bits years Unix seconds = seconds since 1/1/1970 (without leap seconds)
40 MHz count	12	4	u_int	Counts using onboard 40 MHz clock since last pulse per second
40 MHz maximum ticks at last pulse	16	4	u_int	Counts at last pulse per second (pps).
Status	20	1	u_char	Status bits: 0–3 = footer type (a value of 3 = Unix seconds; a value of 5 = IRIG BCD) 4 = has valid IRIG data 5 = is synched with pps 6 = has seen IRIG error 7 = has seen pps error
Reserved	21	3	u_char	Reserved.

Table 1. Elements of IRIG2 Footer – part 2 of 2

Structure element	Offset	Size	Type	Notes
Monotonic Unix time with fractional seconds	24	8	double	This value is computed and filled in by the library after DMA. Fractional time is the 40 MHz count divided by 40 MHz max count

The following C++ struct encapsulates this footer structure, including showing the struct with bit fields representing the packed BCD values from the IRIG signal.

```
// ts_raw_t is defined in libedt_timing.h
// Packed BCD from IRIGB
typedefstruct { // Raw timecode format
    u_long seconds:6;
    u_long minutes:6;
    u_long hours:5;
    u_long days:9; // days in the year
    u_long years:6;
} ts_raw_t;
// This structure is defined in pdv_irig.h
typedef struct Irig2Record {
    u_int magic; /* magic */
    u_int framecnt; /* starts at 0 */
/* There are two modes - seconds from 1970 (Unix time) or BCD mode, in which the BCD
values from IRIG are packed into 32 bits in the raw structure */
union {
    u_int seconds;
    ts_raw_t raw;
} t;
u_int clocks; /* how many 40 MHz ticks in last second */
u_int tickspps; /* 40 MHz ticks since last second */
    struct { /* status bits */
    u_char type: 4;
    u_char irig_ok:1;
    u_char pps_ok:1;
    u_char had_irig_error:1;
    u_char had_pps_error:1;
} status;
u_char reserved[3];
double timestamp; /* holds a 64-bit unix seconds time */
    /* This must be filled in by software */
} Irig2Record;
```

Library API

In the EDT PDV library routines, extra data added by the software or the board (outside of the defined image) typically is called a *header*. In this case, however, such data is at the end, so it is called a *footer*.

EDT's software development kit for the IRIG option includes two new files: `pdv_irig.c` and `pdv_irig.h`.

libpdv.c

The functions below are included in `libpdv.c`.

The following function returns the full size of the DMA (image data plus the 32-byte IRIG footer):

```
int pdv_get_dmasize(PdvDev *pdv_p)
```

The following function returns the footer's byte offset from the beginning of the image data:

```
int pdv_get_header_offset(PdvDev *pdv_p)
```

For the IRIG footer, this will be:

```
image width x height x bytes per pixel
```

NOTE The PCIe8 DV C-Link requires the DMA size to be a multiple of 8 bytes. Therefore, with or without the IRIG option enabled, the image size must be a multiple of 8 bytes.

pdv_irig.c

The functions below are included in `pdv_irig.c`.

The following function will reset the frame counter to 0:

```
int pdv_reset_dma_framecount(PdvDev *pdv_p)
```

The following function sets the IRIG format to packed BCD values, if the value *bcd* is non-zero; otherwise the format is Unix seconds (seconds since 1/1/1970):

```
int pdv_irig_set_bcd(PdvDev *pdv_p, int bcd)
```

The following function will compute the offset of the IRIG footer and return a pointer to its memory (since this in the DMA stream from the board, this pointer will be overwritten when this buffer is used again):

```
irig2Record *pdv_irig_get_footer(PdvDev *pdv_p, u_char *imagedata)
```

The following function computes the Unix time from the counts values in the footer and places the result in footer > timestamp (and the timestamp value is also returned by the function):

```
double pdv_irig_process_time(Irig2Record *footer)
```

The following function can be used to clear the error flags `had_irig_error` and `had_pps_error` (both flags are "sticky" bits, showing that an error has occurred in the IRIG transfer or the pps signal respectively):

```
void pdv_irig_reset_errors(PdvDev *pdv)
```

If the IRIG / pps input is coming from another PCIe8 DV C-Link board, the following function sets the board in slave mode to use the input values from the other board, rather than from the Lemo connector (note – this function requires a special cable from EDT):

```
void pdv_irig_set_slave(PdvDev *pdv_p, int onoff)
```

This function uses the direct register access to the current time value to fill in an Irig2Record structure (the valid elements are: counts, tickspps, status, and the IRIG seconds / bcd values):

```
void pdv_irig_get_current(PdvDev *pdv_p, Irig2Record *footer)
```

irigdump.c

The EDT PDV driver also contains a simple example program named `irigdump.c`. This program, based on `simple_take.c`, exercises most of the functions described above.

To display the command line arguments, run `irigdump -help`.

Revision Log

Below is a history of modifications to this guide.

Date	By	Pp	Detail
20100408	PH, SC	All	Created guide for new app note (IRIG-B) on PCIe8 DV C-Link.