Allied Vision

APPLICATION NOTE

Image Timestamp on Allied Vision GigE Cameras

2015-Sep-23

Description

Each frame returned from the camera contains a timestamp:

- For Vimba users: *GevTimestampValue*
- For PvAPI users: *TimestampHi* and *TimestampLo*

This timestamp is assigned either at the start or the end of exposure (depending upon the camera) for the corresponding frame. The number is in clock ticks, and counts up continuously from the start of camera booting. Timestamp frequency, the clock rate in Hz, is represented by the following read-only parameter on the camera.

- For Vimba users: *GevTimestampTickFrequency*
- For PvAPI users: *TimeStampFrequency*

$$\text{RealWorldTime} = \frac{TimeStamp}{TimeStampFrequency}$$

The camera clock crystal, a CITIZEN CS10-25.000MABJTR has a drift specification of ± 50 PPM.

Users interested in synching the camera Timestamp to another clock should note that the camera clock will drift from the other clock, and therefore will need to be periodically re-synched by resetting the Timestamp.

For Vimba

See the *GevTimestampControlReset*, *GevTimestampControlLatch*, and *GevTimestampValue* features.

For details on camera controls as seen from the Vimba Viewer, please refer to *GigE Features Reference* document

For PvAPI ≥ 1.22

See the *TimeStampReset*, *TimeStampValueLatch*, *TimeStampValueHi* and *TimeStampValueLo* attributes.

For details on camera controls as seen from the PvAPI GigE SampleViewer, please refer to *GigE camera and driver attributes* document

ADVANCED: For PvAPI < 1.22

1. **To RESET the timestamp**: Timestamp Control Register address: 0x0944, length 4 bytes. Bit 31 = reset. WRITE only. Do not READ this register.
2. **To READ the timestamp**: Timestamp Control Register address: 0x0944, length 4 bytes. Bit 30 = Latch current timestamp counter into Timestamp Value register. WRITE only. Do not READ this register.

   Timestamp Value Register address: High part: 0x0948, length 4 bytes. Low part: 0x094C, length 4 bytes. It is necessary to latch the 64-bit timestamp value to guarantee its integrity when performing the two 32-bit read accesses. Access high first, then low. READ only.

Example code

(See PvRegIo.h and examples/siotest for more on PvRegisterRead/Write):

```
unsigned long control_address = 0x0944;
// NOTE: Bit order is reversed:   [0 ---------------------- 31]
unsigned long reset_data = 0x1;
unsigned long latch_data = 0x1 << 1;
unsigned long ts_address[2] = {0x0948, 0x094C};
unsigned long ts_data[2];
unsigned long num_complete;
tPvErr return_value;

// reset
return_value = PvRegisterWrite(handle, 1, &control_address,
&reset_data, &num_complete);

// latch
return_value = PvRegisterWrite(handle, 1, &control_address,
&latch_data, &num_complete);

// read timestamp
return_value = PvRegisterRead(handle, 2, &ts_address[0], &ts_data[0],
&num_complete);
```

For technical support, please contact support@alliedvision.com.
For comments or suggestions regarding this document, please contact info@alliedvision.com.

Disclaimer

Due to continual product development, technical specifications may be subject to change without notice. All trademarks are acknowledged as property of their respective owners. We are convinced that this information is correct. We acknowledge that it may not be comprehensive. Nevertheless, Allied Vision cannot be held responsible for any damage in equipment or subsequent loss of data or whatsoever in consequence of this document.