The Wayback Machine - https://web.archive.org/web/20210906072141/https://dev.intelrealsense.com/docs/external-synchronization-of-intel-reals...

Documentation ⌄   🔍

External Synchronization of Intel® RealSense™ Depth cameras

[✎ SUGGEST EDITS]

Authors: Anders Grunnet-Jepsen, Aki Takagi, John Sweetser, Tri Khuong, Dave Tong
Rev 1.3

> ℹ️ This article is also available in PDF format

1. Introduction

In this whitepaper we introduce a new Genlock function for all Intel® RealSense™ depth cameras that are based on Global Shutter imagers. This currently includes the D435 and D435i cameras. "Genlock", or generator locking, describes the technique whereby one or more video streams can be exactly synchronized in time to an external trigger. With this feature, depth frames can be triggered from an external source to capture at almost arbitrary times or frequencies, within an allowable time and frequency window. This includes being able to trigger at irregular intervals as well as offering the option to start a specified burst of frames with each single trigger.

In a previous whitepaper (https://dev.intelrealsense.com/docs/multiple-depth-cameras-configuration) we had covered a similar capability where it was possible to make one Intel RealSense camera act as the Master and others as Slaves and have them hardware synchronized via a connecting cable. This was extremely helpful for multi-camera operations that required simultaneous capture from multiple cameras. However, the limitation of that approach was that only discrete pre-defined frequencies could be used, like 6, 15, 30, 60 and 90 frames per second. Moreover, the allowable frequency windows for the synchronization trigger signal was so strict as to make it nearly unfeasible to use an external signal generator as source. The new Genlock mode lifts some of these restrictions but adds others, as will be detailed below. Appendix A

provides a table with a more detailed comparison of the differences between the different synchronization techniques.

Documentation ▾   Q

2. Principles of Operation

We start by defining the characteristics of the external HW SYNC trigger: Inputs are high impedance, trigger voltage level is 1.8 V, and pulse width should be wider than 100 microseconds. The Genlocked camera triggers frames on the positive upward slope.

When the Genlock mode is selected, the camera will wait indefinitely for a trigger. Once the trigger arrives, exposure will start, as shown in Figure 1. This detail is important to note when temporally synching with events from other cameras or devices, as there is up to 1 frame delay between trigger and readout. Since the stereo camera exposure is global shutter, the camera sensor will accumulate charge during the exposure time, and when done, it will read it out. The exposure time can either be set manually or can be determined inside the device by its autoexposure algorithm. However, if using auto-exposure, note that this is based on previous frame brightness and the auto-exposure algorithm requires a few frames of data to stabilize. This means that the autoexposure will only respond at the trigger rate and should therefore only be used if there are expected to be only minor lighting changes between triggers.
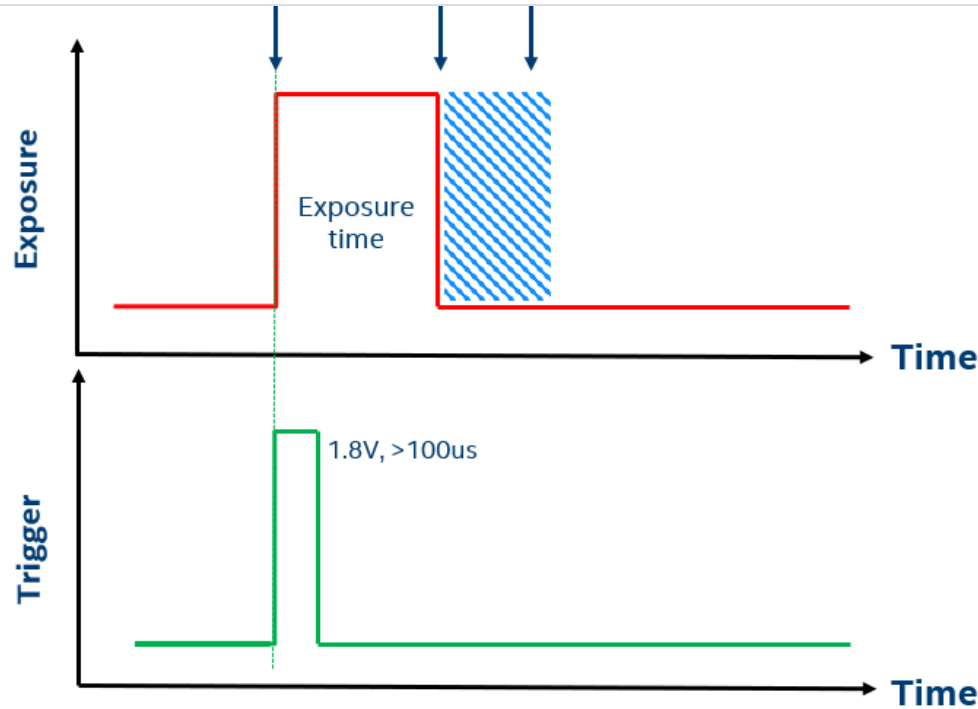
Documentation ⌄   🔍

Figure 1. Timing diagrams for trigger and camera frame.

When streaming in Genlock mode, the camera can be configured during initialization to any of the normal input resolutions and frame rates. Specifically, the "native" frame rate of the camera can be set to 6 fps, 15fps, 30fps, 60fps, or 90fps. This native frame rate has an impact on two significant aspects:

First, it does not set the trigger rate, but it does set an upper limit on the frame rate that can be supported by an external trigger. When sending a trigger pulse train, the trigger signal cannot be repeated faster than HALF the native camera mode frame rate, as shown in the table below. In other words, once a trigger is received the camera will be blind to any other trigger signals for 2x the frame time of the "native mode" of the camera. This means that the trigger frequency can be any value inside the allowable range in the Table 1.

Native Camera Mode, FPS	Allowed External Trigger, FPS
60	0-30
30	0-15
15	0-7
6	0-3

Table 1: The allowed trigger frequencies as a function of the camera mode.

Second, when the Genlock mode is selected, there is the option to adjust a "burst count" from 1 to 255 frames. This means that once a trigger is received, a specified number of frames will be burst at the native frame rate. For example, if the native camera frame rate is 90fps, and the burst count is set to 3, then each time a trigger is received the camera will burst 3 frames at 90fps (i.e. with 11.1ms separation). The camera will then wait for the next trigger, as shown in Figure 1. This does mean that it is possible to approach the full frame rate, in this case 90fps, with a 1Hz trigger, by setting the native camera mode to 90 fps and setting the count to 89 while triggering at 1Hz, for example.

Documentation ⌄   🔍
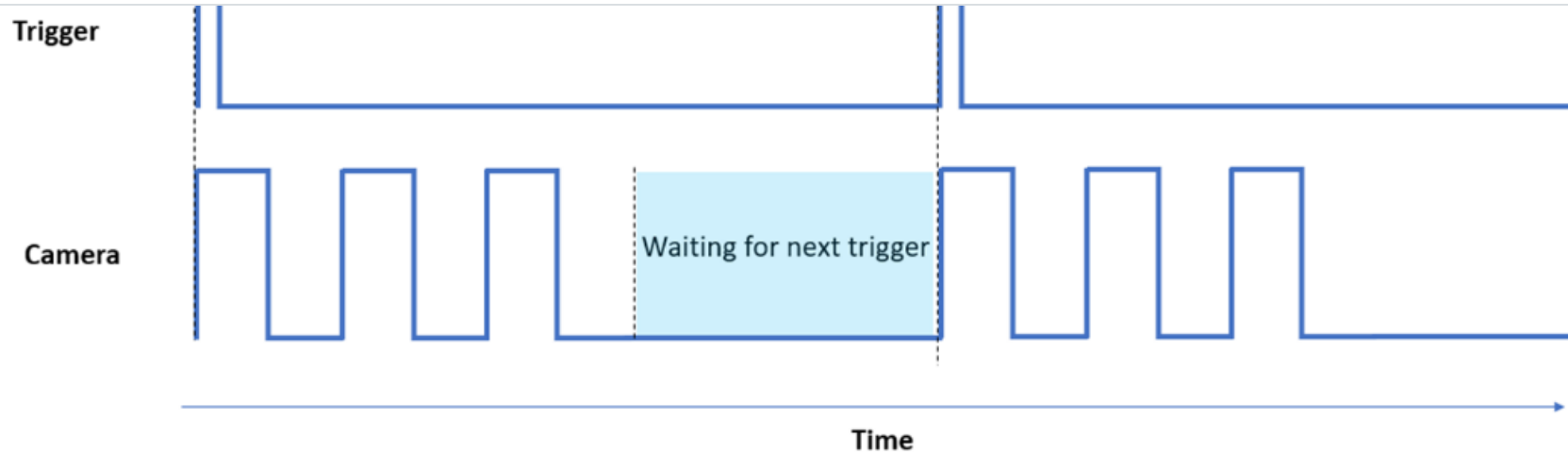

Figure 2. Timing diagrams for multiple triggers and camera frames. The frames will be burst at the native camera frame rate. The burst count can be specified from 1-255.

The next question to address is whether an Intel RealSense camera can act as the external trigger to another Intel RealSense camera. Yes, this is possible. One Intel RealSense camera can be placed into "Master" mode, while all other cameras are placed into "Genlock" mode. However, the same frame rate restrictions apply as outlined in Table 1. So, in order to have all cameras stream at 30fps, for example, the Master must be set to 30fps, and all Genlock cameras must be natively set to 90fps modes with burst count set to 1. The master camera will then deliver the trigger at 30fps, and all Genlock cameras will trigger in sync with the master. This does limit the maximum exposure time of all the slave cameras to 1/90 seconds, instead of 1/30 seconds.

There are several additional restrictions we need to address.

1. **Global Shutter**: The Genlock feature only works on Intel RealSense cameras that are based on Global Shutter stereo imagers. This means that it currently works on the D435 and D435i, but not on D415.

2. **Color**: For the D435/D435i the color imager is rolling shutter and it is not hardware synced to the depth. This is an HW limitation, so the color

not pulse at the frame rate but will automatically be set to be on continuously.

4. **Auto-exposure**: Auto-exposure works but will respond proportional to the camera frame rate, so it is not advisable to be in auto-exposure when the frame rate is very low and lighting conditions may change during exposure.

5. **Firmware and SDK**: This new feature is supported in **FW 5.12.4.0**, and later. It requires Intel RealSense SDK 2.34+.

3. Camera Connections

In this section we provide more details on how to connect a signal generator to a camera. Figure 3 shows a picture of the D435. It has a 9-connector that is hidden by a small magnetic latch. The sync signal needs to be connected to pin 5 (SYNC). Pin 9 is Ground.

Figure 3. Connector a sync cable to the D435.

To connect multiple cameras, you can use a daisy-chain or a star-configuration. For connections longer than 3m, some care must be taken with
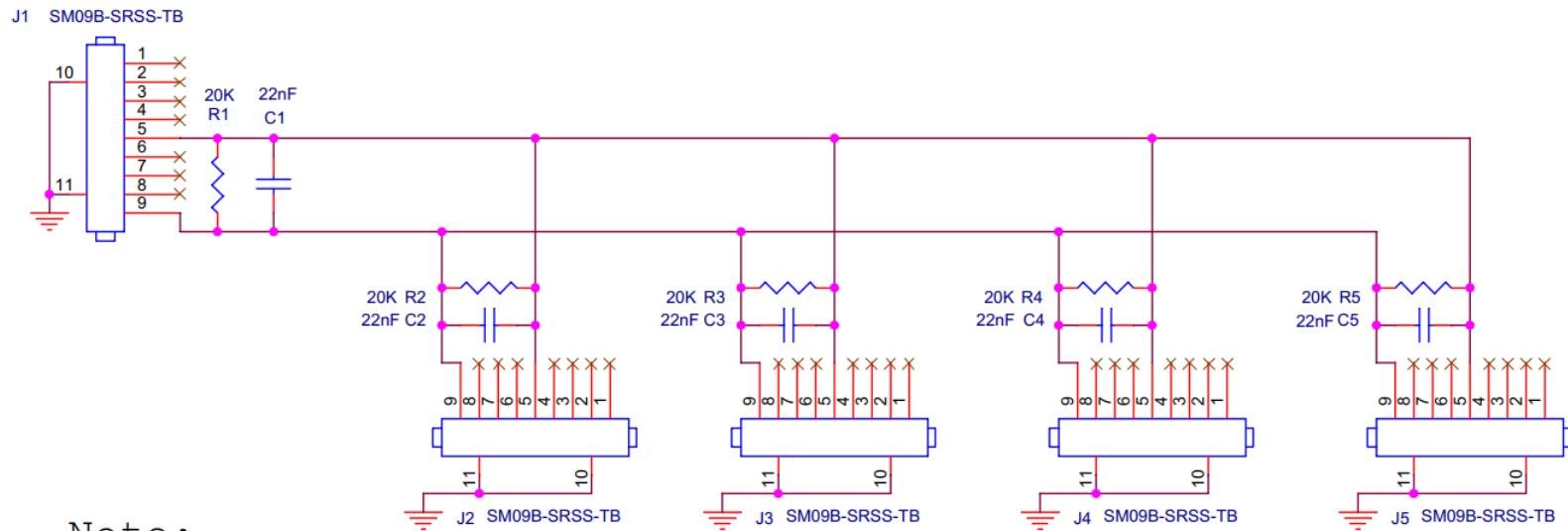
Figure 4. Connecting a MASTER camera to multiple GENLOCKed cameras. The proposed wiring helps mitigate electrical noise.

The connector has manufacture part number ASSHSSH28K152, and the housing has part number SHR-09V-S. Manufacturer web site is www.jst-mfg.com. Figure 5 shows an example of a D435 camera wired to a very simple computer-controlled signal generator – an Arduino Micro with a level-shifter to bring the sync signal to 1.8V.
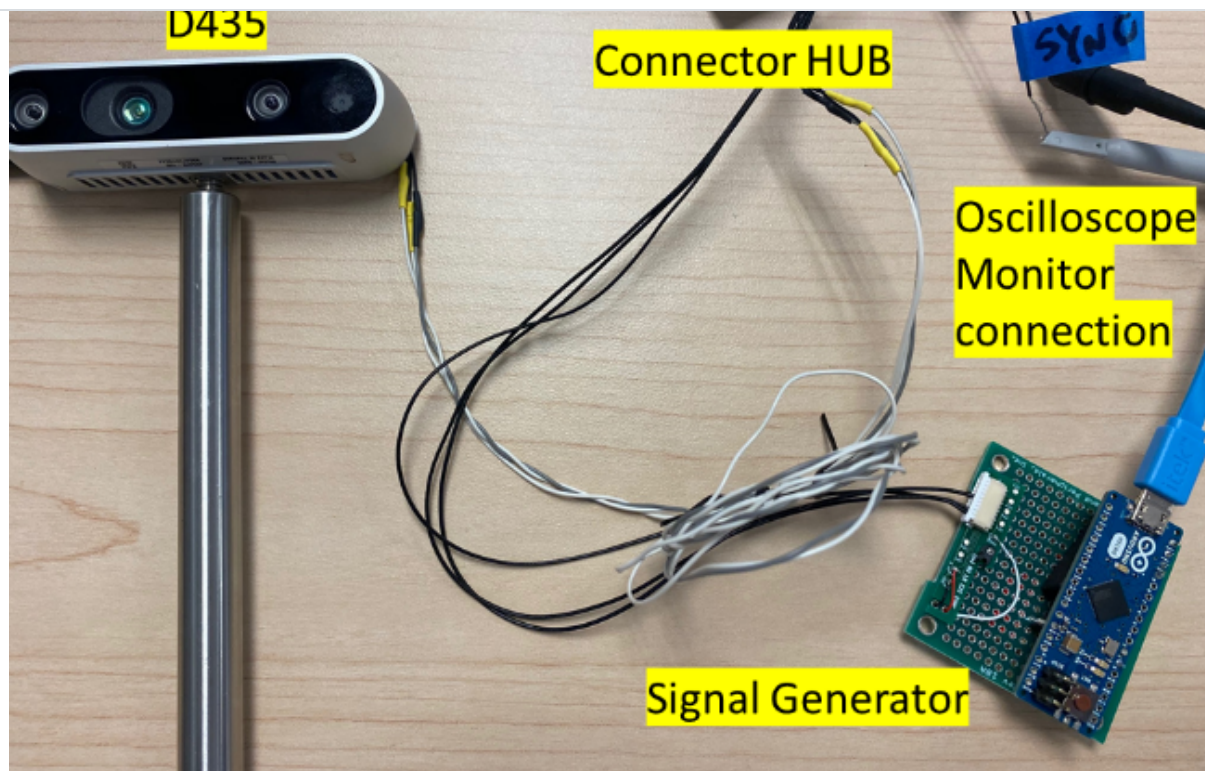
Documentation ⌄

🔍

Fig 5. Example of connecting a D435 to a computer-controlled signal generator.

In our example, it was possible to control multiple cameras simultaneously. Alternatively, one can also create a signal generator that can control the individually multiple output sync signals, with control over the frequency as well as the phase (time delay) between each trigger. See why in the next section.

4. Genlock usages

There are numerous reasons why one might be interested in this Genlock operation, and we will address the most common ones here:

1. **Unique frequencies:** As stated, the Intel RealSense cameras only currently support a few predefined "native" frequencies. These frequencies

Documentation ⌄   Q

2. **Arbitrary times:** For Genlock, no regular repetition rate is needed. The external source could be erratic or random in time, and the capture would still work and be synced exactly to the stimulus trigger. For example, it could be synced to a human heartbeat or a door-entry pressure pad.

3. **Externally controlled Sync:** Probably the most common need for Genlock is when you need to synchronize multiple sources. For example, users may have a separate RGB camera that they want to be the master, and they want Intel RealSense cameras to exactly synchronize to its capture. This is often needed for applications like 3D scene reconstruction which may use a very high-resolution color camera and an Intel RealSense depth camera.

4. **Connecting multiple cameras:** A very common application of Genlock is to link cameras together to be exactly HW synchronized to all fire at the same time. While it is certainly possible to use free-running cameras and software-synchronize by aligning frames according to time-stamps and frame counter, this will only guarantee synchronization to within a frame time. To capture scenes in motion much finer synchronization is normally needed. One example use-case is to have multiple cameras pointing in different directions away from each other (i.e. with non-overlapped views), in order to create a single synchronized point-cloud capture. Another common usage is to scan objects, people or scenes, by having multiple cameras arrayed in a circle around a scene, all looking inwards to capture the scene from different angles. One great advantage of Intel RealSense depth cameras is that they do no interfere with each other, so they can all point at the same scene and capture depth. In fact, the genlock mode automatically sets the projector into CW (continuous wave) mode, so it is on all the time. For more details on multi-camera usages we encourage the user to read the white paper entitled "Using the Intel RealSense Depth Camera 4xx in Multi-Camera configuration.". While this paper details a different frame synchronization mode, it also covers in more detail the many other considerations that need to be made when trying to connect to many Intel RealSense cameras at the same time, including power, bandwidth, and CPU limitations.

5. **Capture high-speed motion:** It is possible to create a synthetic high-speed "bullet-time" camera array through time-staggered triggering. Assuming the scene is bright enough, requiring for example <1ms exposure for an outdoor scene, it is possible to trigger multiple cameras to fire staggered with, say, 1ms delays. So, if 10 cameras are arrayed to capture the same scene at 90fps each, with a 1.1ms delay between each

camera trigger, it is possible to capture motion at 900fps. In fact with the newly release 300fps mode, the same camera configuration would

Documentation ⌄   🔍

observation of very slowly changing scenes, or simply to reduce the computational load on a CPU platform. When using this mode, it is of course necessary to make sure the timeout for the SDK "wait for frame" call is set to longer than the exposure time. We have verified this works for inter-trigger times of 33 seconds.

5. Programming Camera Sync

To get started using this feature, we have included it in the Intel RealSense Viewer 2.34+. It is an option that can be selected BEFORE streaming is started. The feature is called "Inter Cam Sync Mode" and is mode 4-258. Mode 4 will trigger a single frame capture with each trigger. Mode 5 with capture 2, and so on (ex: Burst count = Mode – 3).
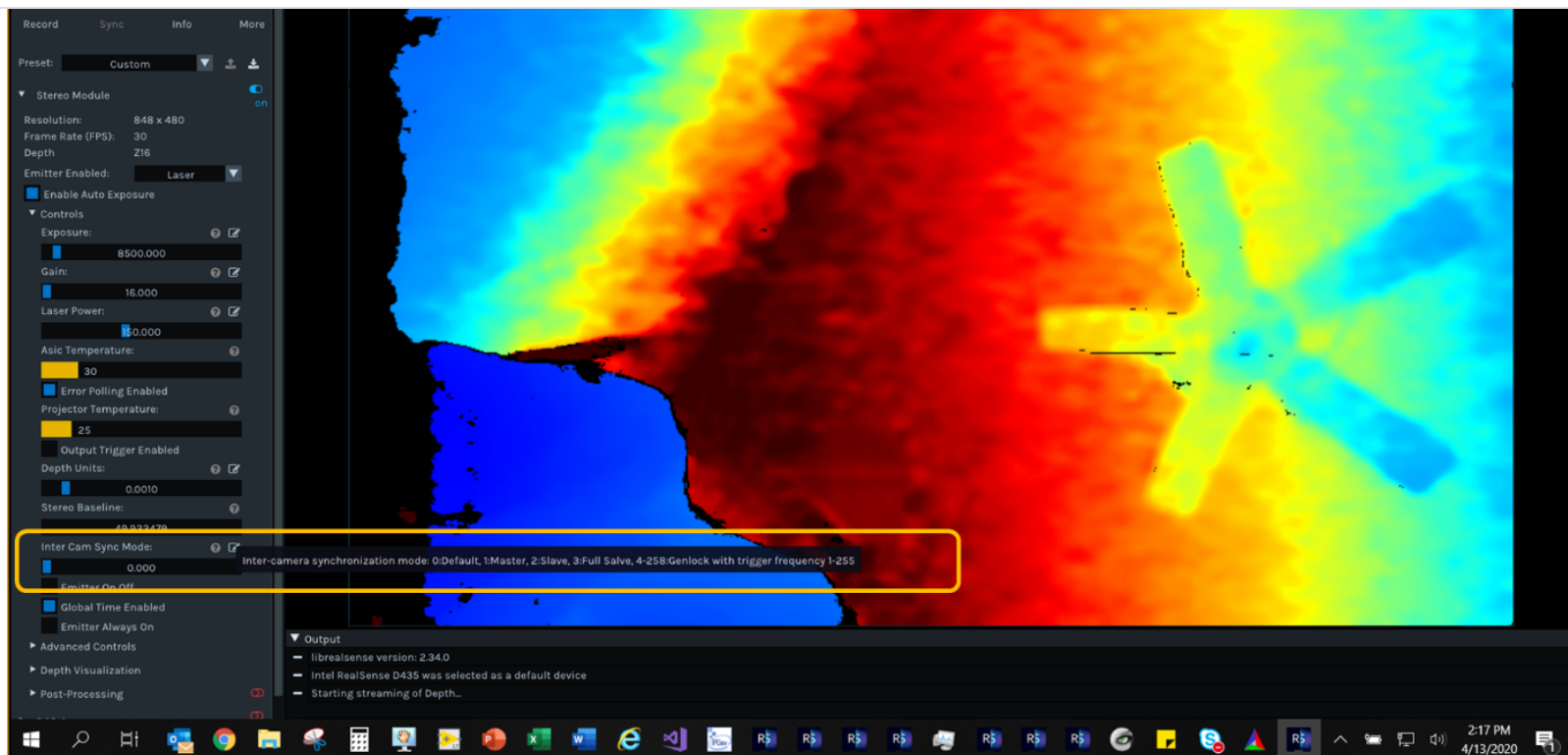
Documentation   ⌄   Q

Fig 6. Activating the Genlock feature via the Intel RealSense Viewer. Selecting 4 will trigger a single frame per external Trigger. Selecting N>4 will trigger N-3 frames per trigger pulse.

To access the feature via the SDK is equally simple. Simply use the RS2_Set_Option

C

```
* write new value to sensor option
* \param[in] options   the options container
* \param[in] option    option id to be queried
```

```
* \param[in] value    new value for the option
```

and for the option select the enumerated flag 42:

C

```
RS2_OPTION_INTER_CAM_SYNC_MODE, /**< Impose Inter-camera HW synchronization mode. Applicable for D400/L500/Rolling S
```

As stated above the value to enter is:

C

```
enum inter_cam_sync_mode
    {
        INTERCAM_SYNC_DEFAULT, //0
        INTERCAM_SYNC_MASTER, //1
        INTERCAM_SYNC_SLAVE, //2
        INTERCAM_SYNC_FULL_SLAVE, //3
        INTERCAM_SYNC_MAX = 258 // 4-258 are for Genlock with trigger frequency 1-255
    };
```

Again, we emphasize that the command should be sent to the device, BEFORE streaming is started, or the mode will not take effect until after streaming has been stopped and restarted. To read back the INTERCAM SYNC value, use the RS_Get_Option.

C

```
 * \return value of the option
 */
float rs2_get_option(const rs2_options* options, rs2_option option, rs2_error** error);
```

6. Conclusion

The ability to synchronize and slave an Intel RealSense Depth camera to an almost arbitrary external input signal has been demonstrated. This feature can be an enabler for applications that require exact hardware time synchronization to an external RGB master camera or to where any other external sync signal is desired to control the exact frame rate and capture time electronically. This can be useful for numerous usages ranging from ultra-slow stop-motion applications to ultra-fast bullet-time 3D captures with multiple cameras.

Appendix A

There are two distinct HW sync slave modes that can be selected in the SDK using the RS2_Set_Option command with

RS2_OPTION_INTER_CAM_SYNC_MODE

Setting Option value to 2 or 3 will set the Depth Camera to SLAVE or FULL SLAVE mode. The only difference between these two modes is that full slave mode also synchronizes the color camera.

The Genlock Slave mode is selected using value 4 (and above). In the table below we explain some of the differences in more detail.

Capability	GenLock (Mode=4+)	HW Sync Slave (Mode=2 or 3)
Operate cameras at arbitrary frame rates	Yes, any external frequency is supported. However, this mode is only supported for Intel RealSense Global Shutter cameras.	No, but can support the few predefined distinct frequencies. It works for all Intel RealSense cameras, whether based on Rolling

| | (not just 6, 15, 30, 60, | | Shutter or Global Shutter. Depth & RGB, can |

	external sources (like other cameras)	rate, as detailed in document. So, if a native frequency is 90fps, the supported sync frequencies will be 0-45fps.	close in frequency to that of the sensor frame rate, or it will fire on its own and wont actually be synchronized in capture.
3	HW sync multiple cameras	Yes, one Intel RealSense camera can be set to Master while the others are in Genlock mode. However, as noted in document, the Master frame needs to be less than half the native frame rate of the Genlocked cameras (unless burst count >1 is used).	Yes, one Intel RealSense camera is set to Master, and the others to SLAVE. Master and Slave cameras will be synchronized to the exact same frequency.
4	Capture at ultra-high speed using multiple cameras with staggered triggers	Yes.	Yes, in principle. However, the sync would need to spawn from the Master Intel RealSense camera, but with electronically configured delays.
5	Capture at very low rates for time-lapse photography	Yes	No, currently the lowest frame rate is 6 FPS.
6	Auto Exposure	Yes, BUT autoexposure needs a few captured frames to settle on the correct value.	Yes
7	Frame Capture Delay	Optical capture starts with Trigger.	Readout (not capture) starts with Trigger.
8	Laser On	Laser ON all the time. This mode does not work with the mode of Toggling of laser on/off every frame.	Normal mode (can be also set to ON all the time).

Documentation ∨   🔍

Updated 2 months ago