Multi-Camera Sync Hub Dev/Pro

Datasheet v1.0

1. Product Brief

Product Overview

Multiple cameras are used to increase camera coverage, fill in occlusions where a single camera might have blind spots or blockage, capture multiple images of the same scene from different viewpoints or angles or increase the effective frame rate. Orbbec's Sync Hubs are available in two versions, the Sync Hub Dev and Sync Hub Pro, and are designed to reduce the complexity and cost of a multi-camera network of compatible Orbbec cameras and external sensors. The trigger voltage is switchable, and the number of secondary devices can be extended using multiple hubs.

The Developer edition, compact and flexible, uses a GPIO interface, perfectly meeting rapid prototyping needs during early-stage development. The Professional edition, designed for commercial use, employs a reliable RJ45 interface, guaranteeing a more stable and longer-range signal connection using standard CAT5 or better cables.

Product Features

Sync Hub Dev, Star and Daisy Chain:

- Compact, dedicated versions for Star or Daisy Chain connections
- 1 Primary and up to 4 Secondary devices, extensible by connecting additional hubs
- Uses 8-pin flat cables
- Switchable voltages when used with USB-C power supply or 5V input

Sync Hub Pro:

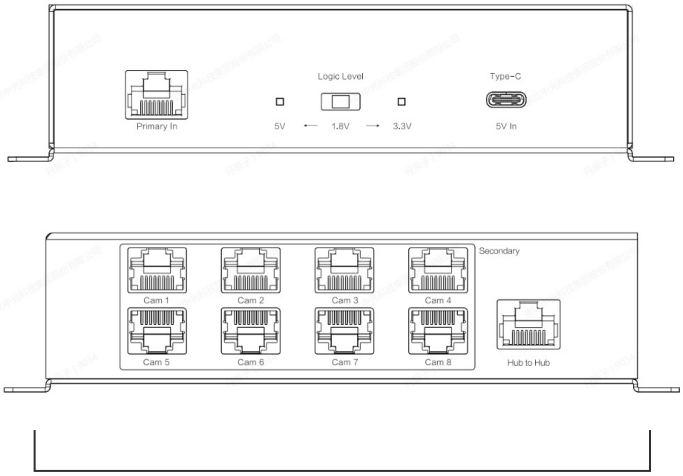
- Compact, mountable enclosure
- Ingenuous use of industry standard RJ45 interfaces and CAT5 or better cables
- Supports 1 Primary and up to 8 secondary devices in Star format, extensible by connecting additional hubs
- Switchable voltages when used with USB-C power supply

Contact us: info@orbbec3d.com | 2800 Livernois Rd, Troy, MI 48083 USA

2. Product Information

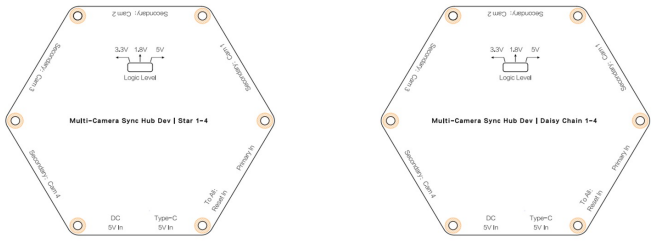
2.1 Multi-Camera Sync Hub Pro Wireframe Diagram

Top view of the Multi-Camera Sync Hub Pro



Front and Rear view of the Multi-Camera Sync Hub Pro

2.2 Multi-Camera Sync Hub Dev Wireframe Diagram



Top view of the Multi-Camera Sync Hub Dev

Contact us: info@orbbec3d.com | 2800 Livernois Rd, Troy, MI 48083 USA

3. Multi-Camera Sync Hub Internal Connections

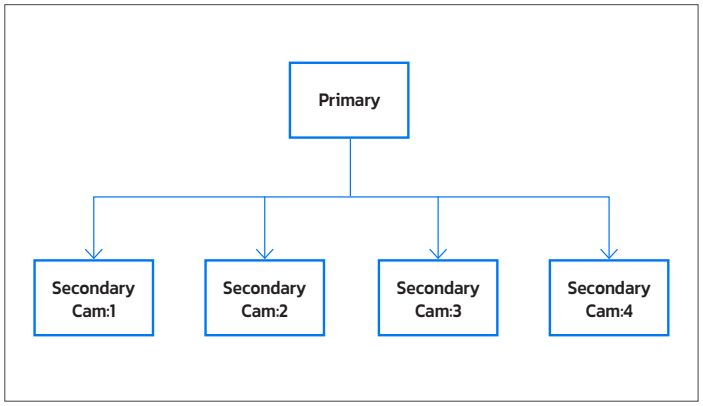
To avoid possible interference when pointing at the same direction, the minimum delay recommended is listed below for compatible Orbbec cameras based on the exposure setting. To achieve the closest exposure between 2 devices, the first pulse of the second camera should fall in the first idle period of the first camera.

Camera	Minimum Delay (μs)
Femto Bolt/Mega (ToF)	160
Gemini 2 Series (Stereo Vision)	0
Gemini 330 Series (Stereo Vision)	0
Astra 2 (Structured Light)	4000

3.1 Star Connections:

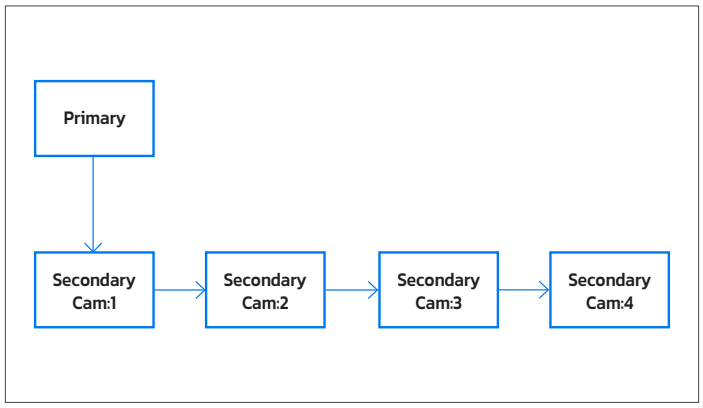
Available on Sync Hub Pro and Sync Hub Dev

When using a Star connection, the Primary device can deliver the trigger pulse simultaneously to each secondary camera or device in the network. However, each secondary camera with an overlapping Field of View will need to be programmed individually with a delay offset from the trigger to match its sequence in capturing.



3.2 Daisy Chain Connections:

Available only for Sync Triger Hub Dev



4. Multi-Camera Sync Interface and Electrical Parameters

Multi-Camera Sync Hub pin definition, applicable to both Multi-Camera Sync Hub Pro and Multi-Camera Sync Hub Dev.

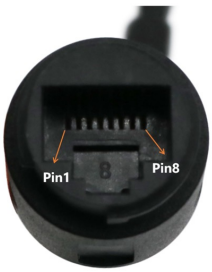
Pin	Definition	Function
Pin_1	SYNC_VCC	• Input Continuous Voltage Level Signal • I/O Voltage Level Setup Signal: The I/O voltage level for the 8-Pin low-speed interface is pre-set to default at 1.8V. When the hub is provided with a 5V supply, this I/O level can be adjusted to either 3.3V or 5V to improve signal stability.
Pin_2	GPIO_OUT	• Output Pulse Signal • Synchronization Drive Signal: IR exposure synchronization signal; commonly used to drive external supplementary lights.
Pin_3	VSYNC_OUT	• Output Pulse Signal • Synchronization Trigger Signal: Used to trigger subsequent devices with a programmable delay.
Pin_4	TIMER_SYNC_OUT	• Output Pulse Signal • Hardware Timestamp Reset Signal: Used to reset the hardware timestamp of subsequent secondary devices.
Pin_5	RESET_IN	• Input Pulse Signal • Camera Hard Reset Signal: Used to trigger the camera to power off and automatically reset upon power on.
Pin_6	VSYNC_IN	• Input Pulse Signal • Synchronization Trigger Signal: A synchronous data collection trigger signal sent from preceding devices.
Pin_7	TIMER_SYNC_IN	• Input & Pulse Signal • Hardware Timestamp Reset Signal: A command to reset hardware timestamp sent from preceding devices.
Pin_8	GND	• Input/Output & Continuous Level Signal • Ground Signal

5. Cables

5.1 Multi-Camera Sync Hub Pro Adapter Cable

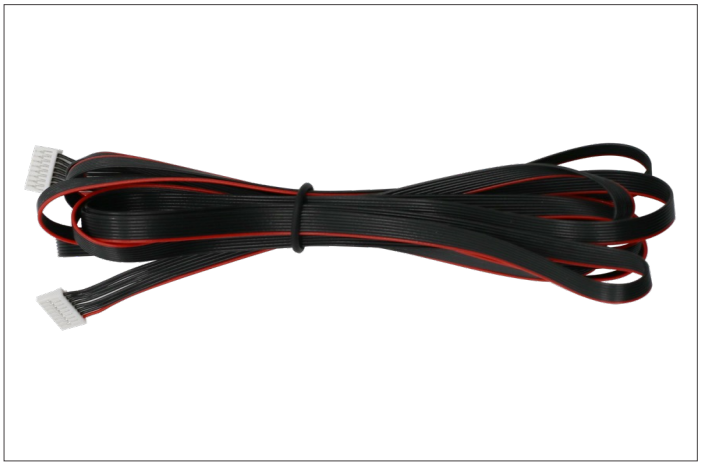
Hub Type	Cable	Version	Specifications	Compatible Models	Note
Multi-Camera Sync Hub Pro	Multi-Camera Sync Cable	GH1.25 8P GPIO to RJ45 (0.15m)	Pitch 1.25mm	Femto Mega Series	Adapter cable: T568B, CAT5e, CAT6
		SH1.0 8P GPIO to RJ45 (0.15m)	Pitch 1.0mm	Gemini 2/330 Series, Astra 2	

Pin	Function	Line Color	Camera Sync Interface Pinout
1	SYNC_VCC	Blue	
2	GPIO_OUT	Blue & White	
3	VSYNC_OUT	Orange	
4	TIMER_SYNC_OUT	Orange & White	
5	RESET_IN	Brown & White	
6	VSYNC_IN	Green	
7	TIMER_SYNC_IN	Green & White	
8	GRD	Brown	

Pin	Function	Line Color	RJ45 (Female) Port Pinout
1	TIMER_SYNC_OUT	Orange & White	
2	VSYNC_OUT	Orange	
3	TIMER_SYNC_IN	Green & White	
4	SYNC_VCC	Blue	
5	GPIO_OUT	Blue & White	
6	VSYNC_IN	Green	
7	RESET_IN	Brown & White	
8	GRD	Brown	

5.2 Multi-Camera Sync Hub Dev Sync Cable

Hub Type	Cable	Version	Specifications	Compatible Models
Multi-Camera Sync Hub Dev	8-pin Cable	GH1.25-8P to SH1.0-8P (1m)	Pitch 1.0mm	Gemini 2/330 Series Astra 2
		GH1.25-8P to GH1.25-8P (1m)	Pitch 1.25mm	Femto Mega Series

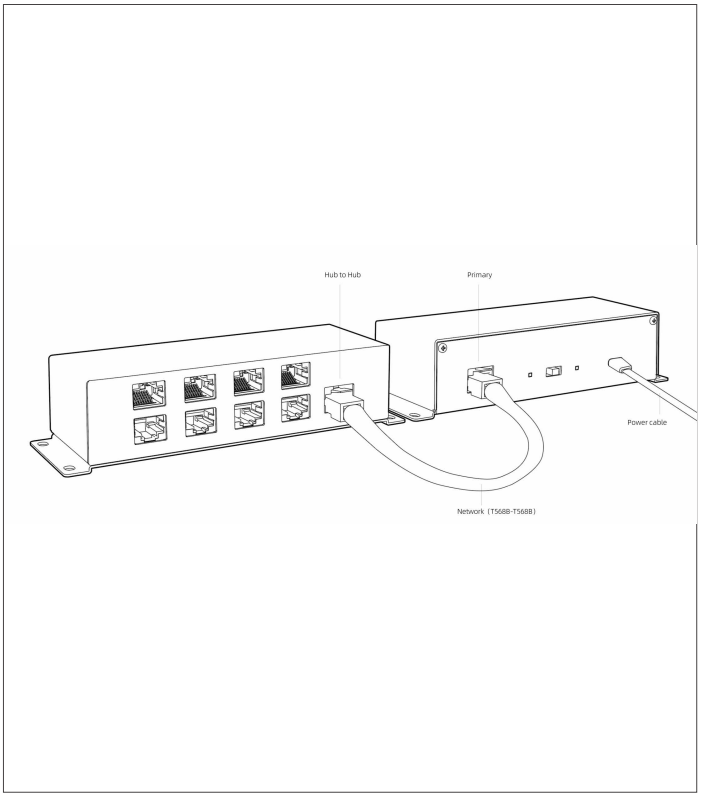


6. Multi-Camera Setup Extended Connection

6.1 Multi-Camera Sync Hub Pro

Each Multi-Camera Sync Hub Pro supports up to 8 cameras or compatible devices. The number of secondary devices can be expanded using a Hub-to-Hub connection.

Hub to Hub cable: T568B standard network cable can be used for direct connection.



6.2 Multi-Camera Sync Hub Dev

The Single Multi-Camera Sync Hub Dev supports the connection of 1 to 4 cameras, with the capability to increase this number through its Hub-to-Hub expansion feature. By utilizing crossover cables in the Hub-to-Hub extension scheme, the crossover end of the cable is connected to the Primary In connector of the additional hub, while the uncrossover end connects to the last secondary connector of the preceding hub.

1	2	3	4	5	6	7	8	Connections
SYNC_ VCC (Grey)	GPIO_ OUT (Purple)	VSYNC_ OUT (Blue)	TIMER_ SYNC_ OUT (Green)	RESET_ IN (Yellow)	VSYNC_ IN (Orange)	TIMER SYNC_ IN (Red)	GND (Brown)	Connect to Hub Secondary Interface
SYNC_ VCC (Grey)	GPIO_ OUT (Purple)	VSYNC_ IN (Orange)	TIMER_ SYNC_ IN (Red)	RESET_ IN (Yellow)	VSYNC_ OUT (Blue)	TIMER SYNC_ OUT (Green)	GND (Brown)	Connect to Hub Primary Interface

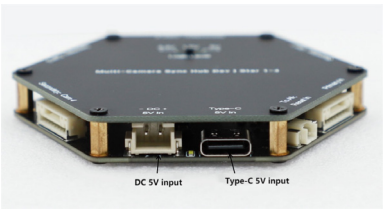
7. Additional Interfaces of Multi-Camera Sync Hub Dev

7.1 Reset In

A power reset for all cameras can be triggered through the Reset In interface.

7.2 Power Input

The developer version supports 5V input to either USB Type-C input or a dedicated DC 5V input connector that is specified as: XH2.54mm 2P terminal wire.



8. Software Configuration

8.1 Software Configuration for USB Type-C Connection

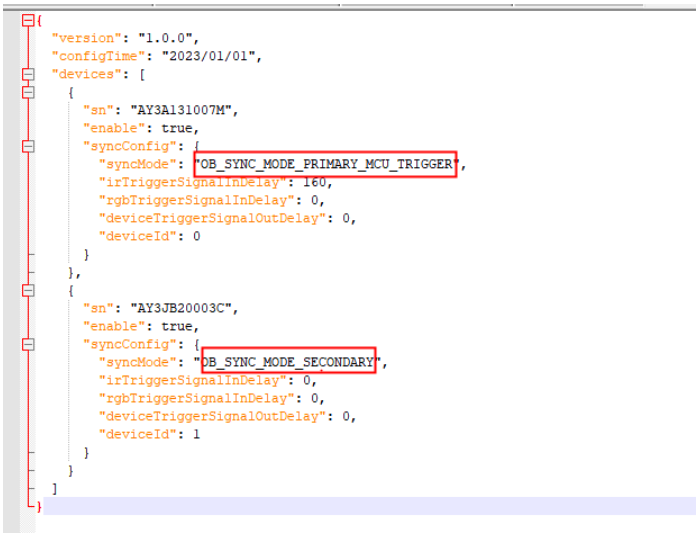
Before the test, it is necessary to set the working mode of each device, mainly Primary/Secondary mode, Depth Delay (irTriggerSignalInDelay) and other parameters. The following uses MultiDeviceSync as an example to illustrate how to configure USB Type-C multi-device synchronization.

Device Configuration

1. Find the configuration file `MultiDeviceSyncConfig.json` in the Orbbec SDK program folder and modify the content of the configuration file according to the actual situation.

```
// MultiDeviceSyncConfig.json
{
    "sn": "AY3A1310007M", // device serial number
    "syncConfig": {  // For details, please refer to the definition and description of the OBDeviceSyncConfig structure in the ObTypes.h file
      "syncMode": "OB_SYNC_MODE_MCU_PRIMARY",  // Synchronous Mode Configuration
      "irTriggerSignalInDelay": 160,//ir/depth/tof trigger signal input delay, in microseconds; in order to prevent laser interference, it is recommended to configure this delay between different devices to make the laser stagger 160us
      "rgbTriggerSignalInDelay": 0, //rgb trigger signal input delay, in microseconds
      "deviceTriggerSignalOutDelay": 0,  // Device trigger signal output delay, in microseconds
      "deviceId": 0  // Device ID, can be used for device number
    }
}
```

For example, configure the Primary and Secondary of 2 devices as follows:



2. Run `MultiDeviceSync.exe`, enter option `0` and press Enter. Wait for the device to complete configuration and restart.

```
Please select options:
0 --> config devices
1 --> start stream
input: 1

config[0]: SN=CL2LC2P0089, mode=OB_SYNC_MODE_PRIMARY
config[1]: SN=CL2LC2P00C0, mode=OB_SYNC_MODE_SECONDARY
Device sn[CL2LC2P0089] is configured, rebooting...
Device sn[CL2LC2P00C0] is configured, rebooting...
```

Software Configuration: Continued from previous page

Run Multi-Device Synchronization

There are two ways to start: either 1) configure cameras as primary or secondary first, or 2) start the stream directly. The second is recommended unless you need to reconfigure for device synchronization, either to switch cameras from primary to secondary or to change sequence between the secondary devices.

1. After the device synchronization configuration is completed, the depth and RGB data stream of the device will automatically start.
2. You can also run `MultiDeviceSync.exe`, enter option `1` and press Enter to start the depth and RGB data stream of the device.

```
Please select options:
0 --> config devices
1 --> start stream
input: 1

start secondary devices...
start primary devices...
```

Notice

1. After the device synchronization configuration is completed, the parameters will be written to the flash memory in the device. It does not need to be reconfigured every time it is used. Frequent configuration will decrease the service life of the flash memory.
2. Some models of devices will take some time to automatically restart after the configuration is completed.
3. After starting the device, please press the `ESC` key in the image preview window to stop the data flow and exit the program. Force stopping or abnormal program exit may cause the device to shut down incompletely, which will cause the Secondary device to be triggered all the time (restart the device to solve it).

Key code description

1. Device clock synchronization

The main purpose of device clock synchronization is to ensure the synchronization of data frame time stamps between different devices, to facilitate subsequent synchronization and matching of data frames. The device clock synchronization uses an asynchronous time service scheme, that is, the PC computer provides time service to each connected device, which can be completed by the following code:

```
context.enableMultiDeviceSync(3600000); // This operation will immediately synchronize the time service of all created devices, and will automatically time service every hour (3600000 seconds)
```

2. Device synchronization configuration

Please refer to the definition and description of the `OBDeviceSyncConfig` structure in the `ObTypes.h` file:

```
dev->getSyncConfig(); // Get the synchronization configuration of the current device

dev->setSyncConfig(SyncConfig); // Set the device synchronization configuration, this operation will write the parameters to the device Flash, and the device will take effect after restarting
```

3. Primary/Secondary device start stream

Please refer to the definition and description of the `OBDeviceSyncConfig` structure in the `ObTypes.h` file:

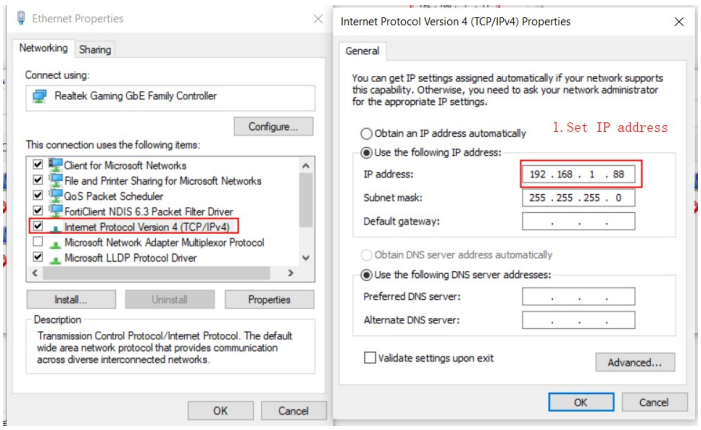
```
std::cout << "Secondary devices start..." << std::endl;
startStream(secondary_devices, OB_SENSOR_DEPTH);
startStream(secondary_devices, OB_SENSOR_COLOR);

// Delay and wait for 5s to ensure that the initialization of the Secondary device is completed
std::this_thread::sleep_for(std::chrono::milliseconds(5000));

std::cout << "Primary device start..." << std::endl;
startStream(primary_devices, OB_SENSOR_DEPTH, secondary_devices.size());
startStream(primary_devices, OB_SENSOR_COLOR, secondary_devices.size());
```

8.2 Network Software Configuration

(Only for Cameras with Ethernet Data Support)

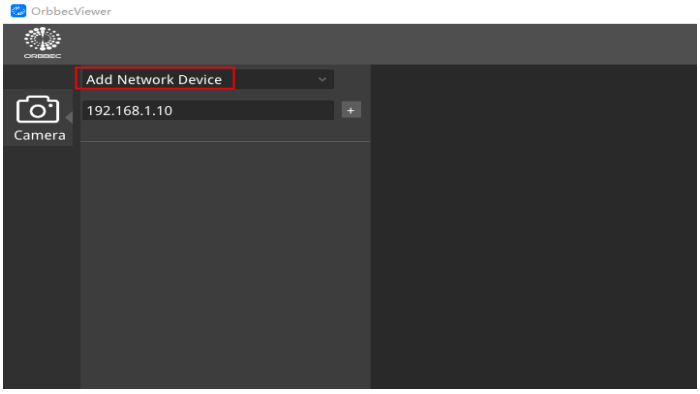
Configure Computer IP

1. After the computer recognizes the network from the router, the IP needs to be reconfigured, and the Ethernet IP and the device must be configured on the same network segment. The default IP of the camera is 192.168.1.10, and the wired network IP of the computer can be changed to 192.168.1.x.

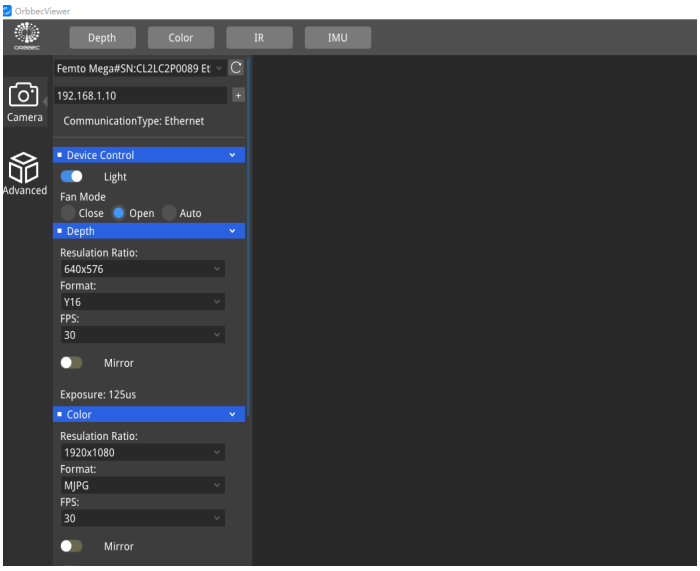


Software Configuration: Continued from previous page

2. After setting up the device IP, you can ping it first on the terminal to confirm that the PC is connected to the device.
3. Open OrbbecViewer, as shown below:

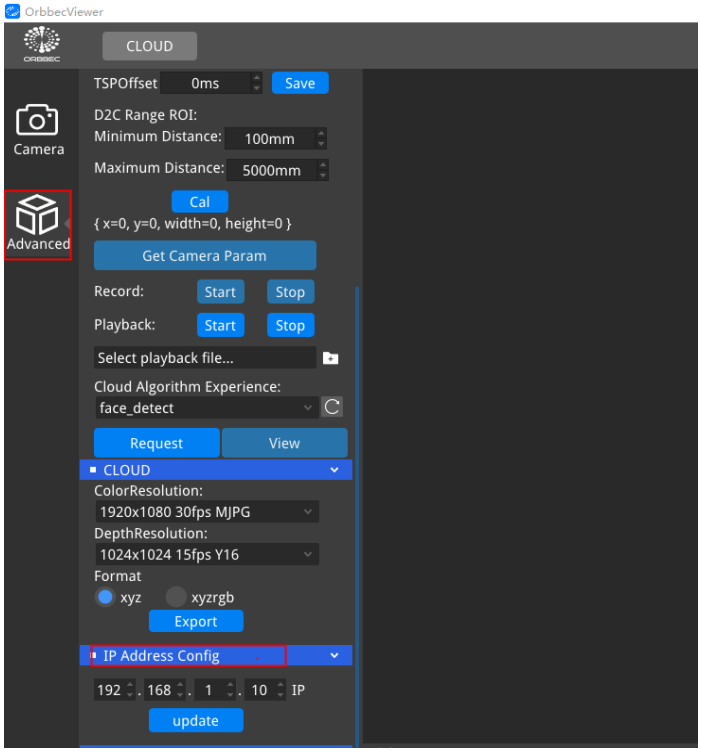


4. Click to add a network device, and the following box will pop up. Enter the device IP, then click + to connect the device. The successful connection interface is as follows:



Configure Each Device's IP address

Find "Advanced Mode" → "IP Address Config", then click "update" after modifying the IP. It is recommended to set the last digit according to the order of placement, such as Primary Device IP: 192.168.1.11, Secondary Device IP: 192.168.1.12, Secondary Device2 IP: 192.168.1.13.



**Note: An error will be returned after the update, because the device cannot get the return value after the IP is updated. You need to manually connect to the newly set IP again.*

Network (PoE) Multi-Device Synchronization

Network connection with PoE multi-device synchronization is different from USB Type-C multi-device synchronization, and the way to connect devices is different. Please refer to the NetDevice example for network stream acquisition. After obtaining the data stream, the configuration of multi-device synchronization is the same as that of USB Type-C.

For network mode multi device synchronization, please combine the NetDevice and MultiDeviceSync examples to complete the process.