INOGENI CAM230 User Guide

User guide

Version 1.4

August 14, 2024

VERSION HISTORY

Version	Date	Description
1.0	April 12, 2023	First release.
1.1	June 2, 2023	• Added new REMOTE connectivity. • Added power consumption limit for USB cameras. • Added RCM statement. • Added minor modifications to RESTAPI. • Added firmware update procedure.
1.2	August 15, 2023	• Added information for future mounting options. • Specified M2.5 mounting holes on enclosure. • Added information for RS232 command delimiters. • Added more precision regarding the “disableSerialInterface” command over RESTAPI. • Added UKCA statement. • Simplified RESTAPI presentation.
1.3	September 25, 2023	• Adding the “Auto (disable remote)” video switching mode. • Adding the “Audio follows video” audio switching mode.
1.4	August 14, 2024	• Adding description on REST API token format. • Adding missing commands over RESTAPI and RS232.

CONTENTS

TYPICAL APPLICATION

Here is a typical connection diagram used for the CAM230 device in a videoconferencing setup.

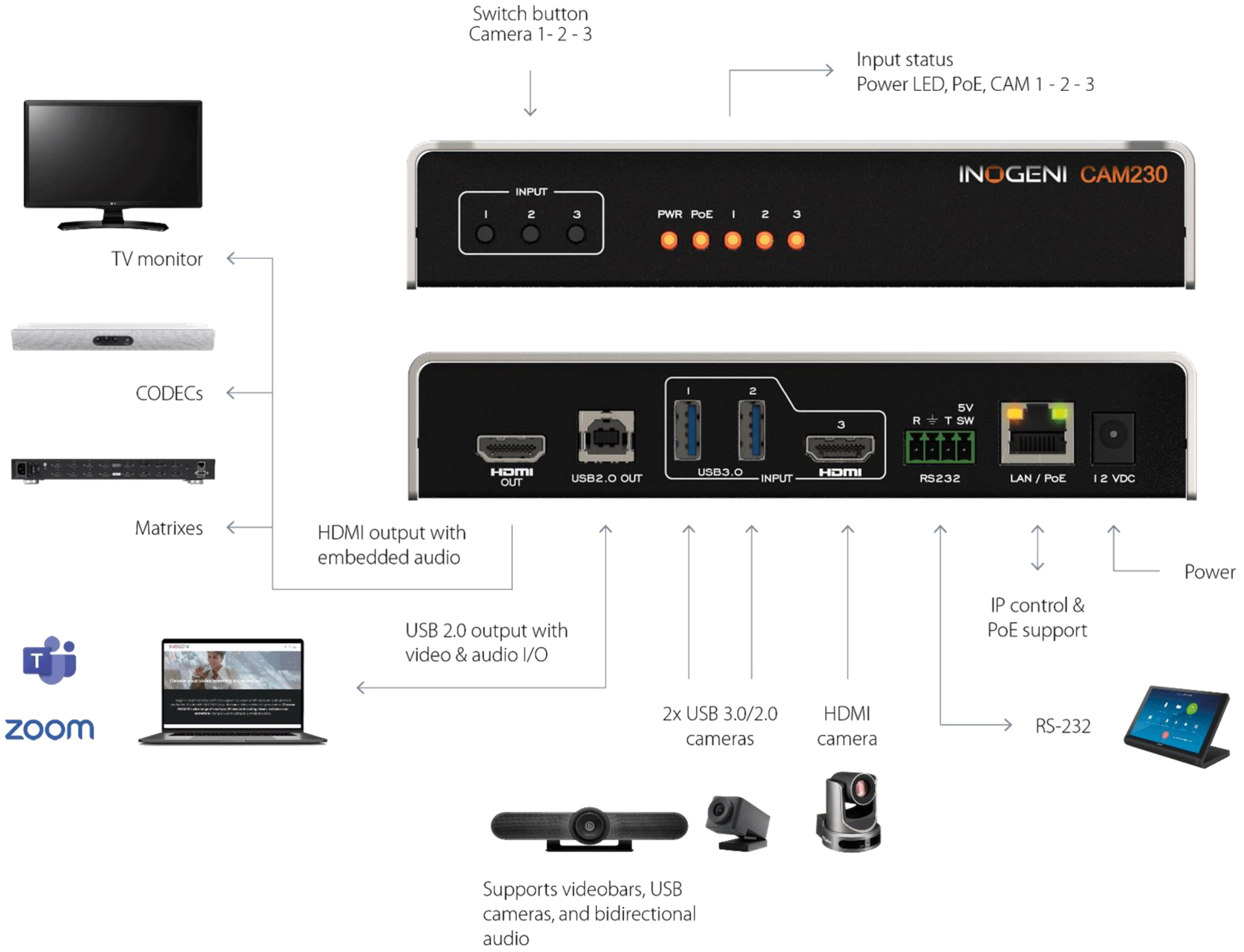


Figure 1: Common use case

BLOCK DIAGRAM

Here is a simple block diagram to better understand the usage of the product.

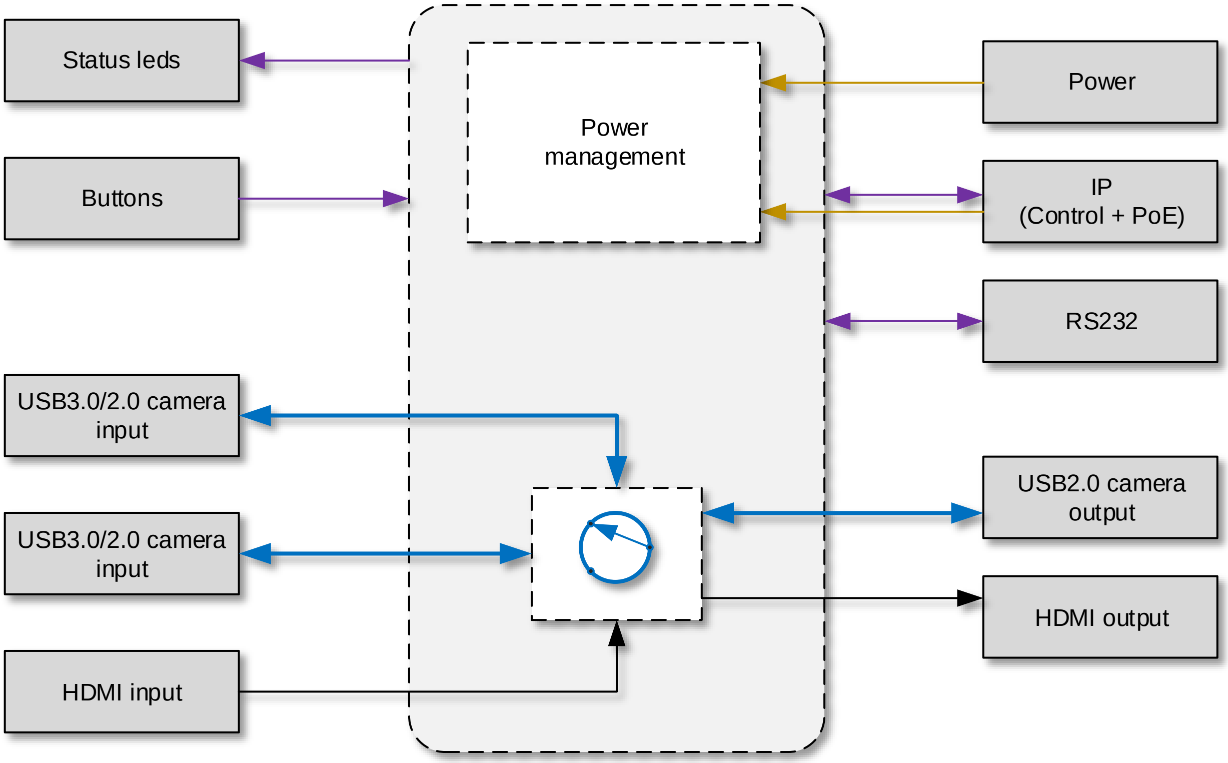


Figure 2: Basic block diagram

The device embeds a video switch that can connect to USB3.0/2.0 and HDMI cameras. The device will output video content from the selected source over HDMI and USB2.0 outputs simultaneously.

This can give you a great asset to your videoconferencing equipment to switch camera interface easily without going into your UC settings.

AUDIO ROUTING

The device can connect to USB microphones/speakerphones and on a HDMI feed that includes audio. These sources can be sent over the HDMI output and to the USB2.0 output.

The USB2.0 interface embeds a digital audio interface which implements a microphone and a speaker device. Ultimately, the UC software connected to the USB2.0 camera interface can receive and transmit audio from/to the connected USB videobar.

DEVICE INTERFACES

Here are the devices interfaces.

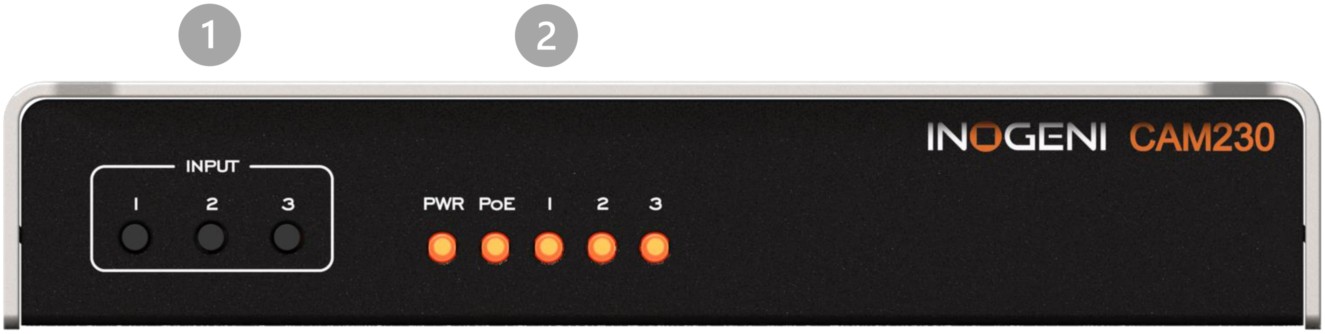


Figure 3: Front side connections

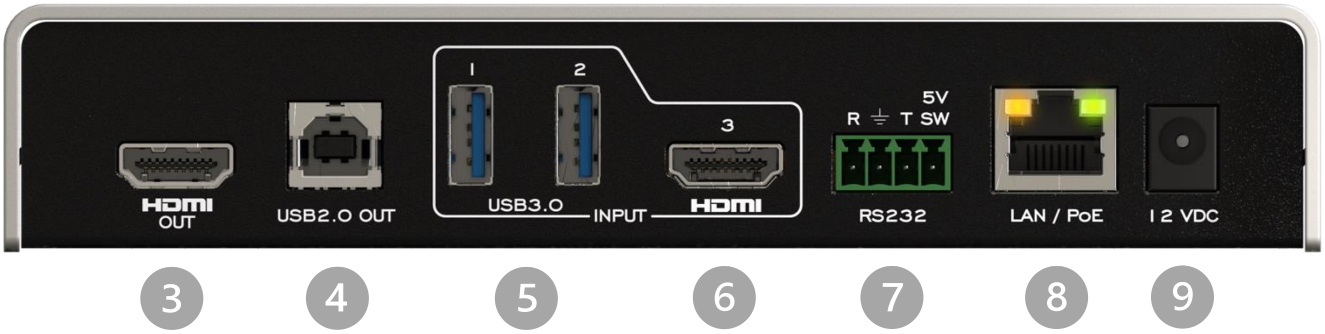


Figure 4: Back side connections

1 Input selection buttons
2 Input status LEDs
3 HDMI output
4 USB2.0 camera output
5 USB3.0 camera inputs
6 HDMI camera input
7 RS232 and remote interface
8 LAN/PoE interface
9 +12VDC power input

LEDS BEHAVIOR

Here are the LEDs behavior:

Power input	
OFF	No power.
SOLID	Device is powered up.
PoE	
OFF	Not powered from PoE.
SOLID	Powered from PoE.
Input led	
OFF	Input not detected and not selected.
LOW	Input detected and not selected.
HIGH	Input selected.

SPECIFICATIONS

Here is the complete specification.

Physical details	
Dimensions (W x L x H)	17.33 cm x 11.57 cm x 3.26 cm 6.82” x 4.55” x 1.28”
Power supply	12V (100-240 VAC 50/60Hz to 12V/1.2A DC) – or – PoE source compliant with IEEE 802.3af (802.3at Type 1)
Weight	600 g
Package content	1 x USB 2.0 Type-B to Type-A cable (3ft). 1 x terminal block connection. 1 x 12V power supply.
Operating temperature	0° to 45° C (32° to 113° F)
Storage temperature	-40° to 105° C (-40° to 221° F)
Relative humidity	0% to 90% non-condensing
Mounting options	Ability to mount under the table or on a wall. More details will follow shortly
UPC code	51497302825
Origin	Canada
Warranty	2 years

USB inputs	
2x USB 3.0 inputs	**1x USB 3.0/2.0 camera** Uncompressed, YUYV/NV12/I420 – 1080p30 MJPEG – 1080p30
USB Power	Up to 1.2A shared between the USB ports

HDMI input	
Resolution	1080p50/60 fps, 720p50/60 fps
Connector	HDMI

HDMI output	
Resolution	3840x2160p23.98/24/25/29.97/30 fps, 1080p50/60 fps, 720p50/60 fps
Connector	HDMI

USB output	
USB connector	1x USB 2.0 Type-B UVC interface up to 1080p30 MJPEG. UAC interface with audio I/O. PTZ control supported.

Audio	
Digital audio I/O	Embedded in HDMI or USB

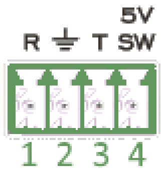
Control	
Control options	Front buttons RS-232 LAN USB
IP interface	100 Mbps half-duplex (autonegotiation not supported) Supports DHCP or static IP addressing
RS232 interface	Baud rate: 9600 Data bits: 8 Stop bits: 1 Parity: None Flow control: None

Certifications	
HDCP compliant	The device does not decrypt BD/DVD movies, satellite/cable receivers or other encrypted sources.
Certifications	FCC, CE, UKCA, RoHS, IEC62368, SoV, RCM
TAA-compliant	Yes

Compatibility	
Operating system	NO driver installation necessary Windows 7 and above (32/64-bit) macOS 10.10 and above Linux (kernel v2.6.38 and above)
Cameras Supported	Cameras (or video Source) with a HDMI or USB outputs
Software Compatibility	UVC-compliant. Runs with all software compatible to DirectShow/MediaFoundation, V4L2, QuickTime and AVFoundation.

SERIAL COMMUNICATION PROTOCOL

Here is the complete list of commands provided through the serial connection. As written on the back of the device, here is the pinout of the terminal block.



Pin 1: Receive
Pin 2: GND
Pin 3: Transmit
Pin 4: 5V supply (for INOGENI Remote)

There must be a space character between command and arguments.

Typically, commands will return ACK in case of success and NACK in case of failure.

You need to add a carriage return character and a line feed "**\r\n**" at the end of the command string.

Note that if serial interface was disabled using REST API, commands will not be parsed, and no response will be provided.

Baud rate: 9600 // **Data bits**: 8 // **Stop bits:** 1 // **Parity:** None // **Flow control:** None

Command	Argument(s)
`HELP` Return command list with description.	None
`RSTR` Restore default settings (including password and REST API token).	None
`RESET` Reset/reboot the device.	None
`IP` Returns IP address.	None
`VERSION` Returns firmware version.	None
`STATUS` Return devices, video/audio inputs and HDMI output status.	None
`PAN` Relative pan of the selected camera.	1 argument (integer) The sign specifies the direction. We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allows. We recommend using values between `-10` and `10`.
`TILT` Relative tilt of the selected camera.	1 argument (integer) The sign specifies the direction. We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allows. We recommend using values between `-10` and `10`.
`ZOOM` Relative zoom of the selected camera.	1 argument (integer) The sign specifies the direction. We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allows. We recommend using values between `-10` and `10`.

`SETHDMI`
Set HDMI output mode.
	1 argument (integer)
	Possible parameters:
	`0 => 1080P60`
	`1 => 1080P50`

	2 => 720P60 3 => 720P50 4 => 4K24 5 => 4K25 6 => 4K30
SETVIDEOFORMAT Set video input format.	2 arguments (integer) 1st arg: <inputPort> 1 => USB input #1 2 => USB input #2 4 => Test pattern 2nd arg: <formatIndex>
GETVIDEOFORMATS Get video input format.	1 argument (integer) The argument specifies the port of the video input for which we want to get the available formats. 1 => USB input #1 2 => USB input #2 3 => HDMI input 4 => Test pattern
SETAUDIOINPUT Set audio input.	1 argument (integer) The argument specifies the index (start at 0) of the audio input. To see available inputs, use STATUS command.
SETAUDIOINPUTMODE Set audio input switching mode.	1 argument (integer) The argument specifies the audio input mode. 0 => First connected audio device with fallback 1 => Audio follows video selection
SETVIDEOINPUT Set video input.	1 argument (integer) The argument specifies the port of the video input. 1 => USB input #1 2 => USB input #2 3 => HDMI input 4 => Test pattern To see available inputs, use STATUS command.
SETVIDEOINPUTMODE Set video input switching mode.	1 argument (integer) The argument specifies if we want manual or automatic switching. 0 => automatic switching 1 => manual switching 2 => automatic switching (disable remote)
ENUSBCDCNCM Enable/Disable the USB CDC-NCM interface over USB2.0 output.	1 argument (integer) The argument specifies if we want to enable of disable the interface. 0 => OFF 1 => ON
ENUSBOUTSPEAKER Enable/Disable the USB speaker interface over the USB 2.0 output.	1 argument (integer) The argument specifies if we want to enable of disable the interface. 0 => OFF 1 => ON
ENUSBOUTMIC Enable/Disable the USB speaker interface over the USB 2.0 output.	1 argument (integer) The argument specifies if we want to enable of disable the interface. 0 => OFF 1 => ON

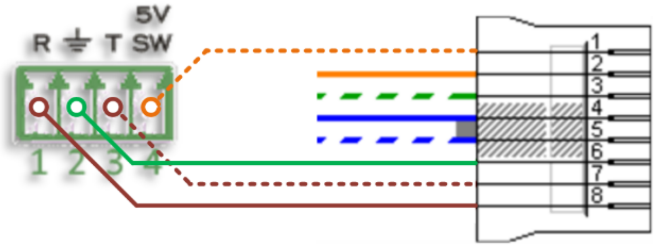
INOGENI REMOTE

The INOGENI Remote needs to be connected to the terminal block port in order to operate. Apply wiring accordingly. This remote is sending serial commands to the device. Make sure to set the **DIP SW6** below the device to ON in order to apply power to the remote before going further. See “DIP SWITCHES” section and user manual of the INOGENI REMOTE for more details.

⚠ **Do not plug a RJ45 cable between the INOGENI device and the REMOTE.**

Terminal block:
Pin 1: Receive
Pin 2: GND
Pin 3: Transmit
Pin 4: 5V supply

RJ45:
Pin 1: 5V supply
Pin 2,3,4,5: NC
Pin 6: GND
Pin 7: RX
Pin 8: TX



LAN COMMUNICATION PROTOCOL

You can access the device settings through its LAN interface. The LAN interface uses DHCP (default) and static IP addressing. You can obtain the IP from the INOGENI Control App or from the serial port IP command. Note that LAN is set to 100Mbps half-duplex.

CDC-NCM COMMUNICATION PROTOCOL

The device can also be controlled through CDC-NCM interface exposed on the USB2.0 device port.

This interface has the same functions as the LAN interface, except the requests are done through USB to ease configuration.

CDC-NCM IP address: 169.254.10.10

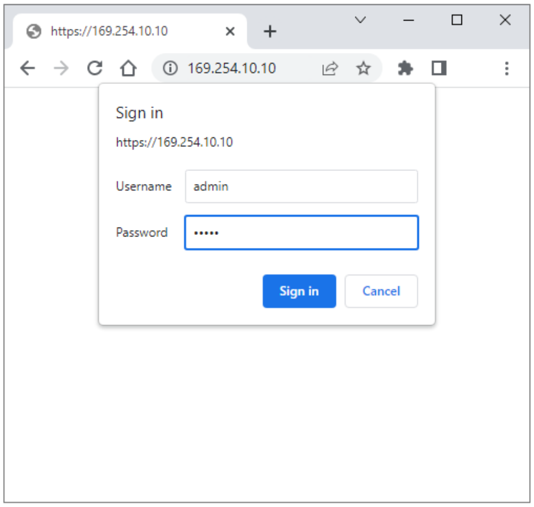
POE

The device can be powered from a 12V power supply or from a PoE compliant source. If the 12V power supply is connected, this one is taken in priority.

WEBPAGE

Here is the webpage that can be used to configure and upgrade the device. This webpage is accessible through IP or through the CDC-NCM interface over USB2.0.

The username is “**admin**”, and the default password is “**admin**”.



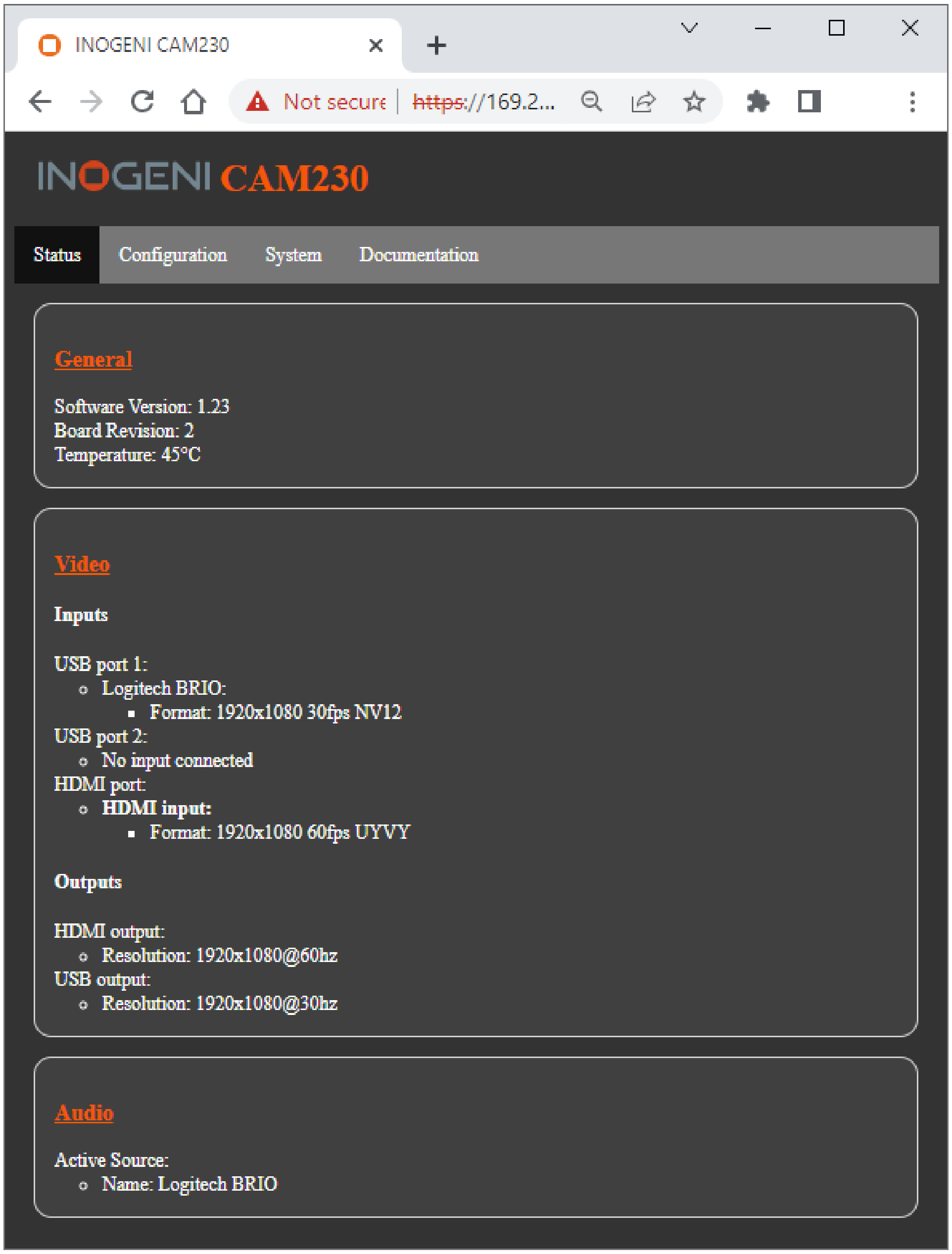
Sign in

https://169.254.10.10

Username admin

Password •••••

The **STATUS** page will give you information about the firmware installed. video and audio devices that you can monitor.



General

Software Version: 1.23
Board Revision: 2
Temperature: 45°C

Video

Inputs

USB port 1:
- Logitech BRIO:
  - Format: 1920x1080 30fps NV12

USB port 2:
- No input connected

HDMI port:
- **HDMI input:**
  - Format: 1920x1080 60fps UYVY

Outputs

HDMI output:
- Resolution: 1920x1080@60hz

USB output:
- Resolution: 1920x1080@30hz

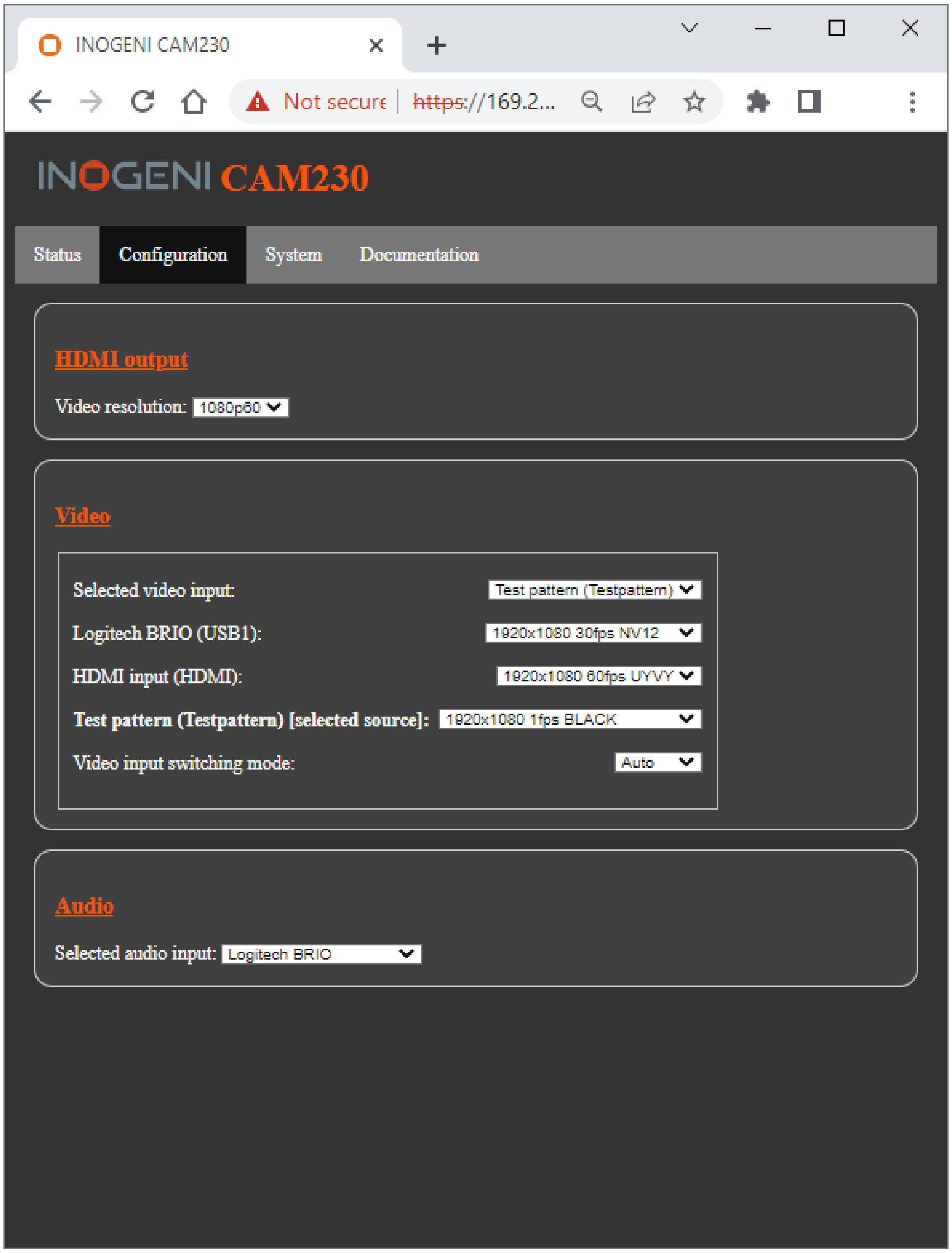
Audio

Active Source:
- Name: Logitech BRIO

Figure 5: Status preview

The **CONFIGURATION** tab will allow you to :

- Set the HDMI resolution over HDMI
- Set the selected camera source
- Set the video input switching mode
  - AUTO : Device will switch to newly detected video source
  - MANUAL : Device will only switch when we get the control to do it.
  - AUTO (DISABLE REMOTE)
    - Support switching between inputs.
    - If any button is pressed, disable remote switching, and stay on camera selected. If API calls are done after to switch the input, API returns message that the command was not applied due to the current mode.
    - If the currently selected input/button is pressed again, enable remote switching. There will be retroaction to ALL 3x leds (flashes 3 times / rapidly) indicating the device re-enabled remote switching.
    - If Teams/Zoom call ends (video streaming stopping), enable remote switching.
- Set the audio input from USB sources or HDMI input.



HDMI output

Video resolution: 1080p60

Video

Selected video input: Test pattern (Testpattern)

Logitech BRIO (USB1): 1920x1080 30fps NV12

HDMI input (HDMI): 1920x1080 60fps UYVY

Test pattern (Testpattern) [selected source]: 1920x1080 1fps BLACK

Video input switching mode: Auto

Audio

Selected audio input: Logitech BRIO

Figure 6: Configuration preview

The **SYSTEM** tab will allow you to :

- Change the current password for accessing device settings.
- Get/Set REST API access token needed using REST API interface.
- Change network settings of your device.
- Restore default settings and reboot the system.
- Update your system.

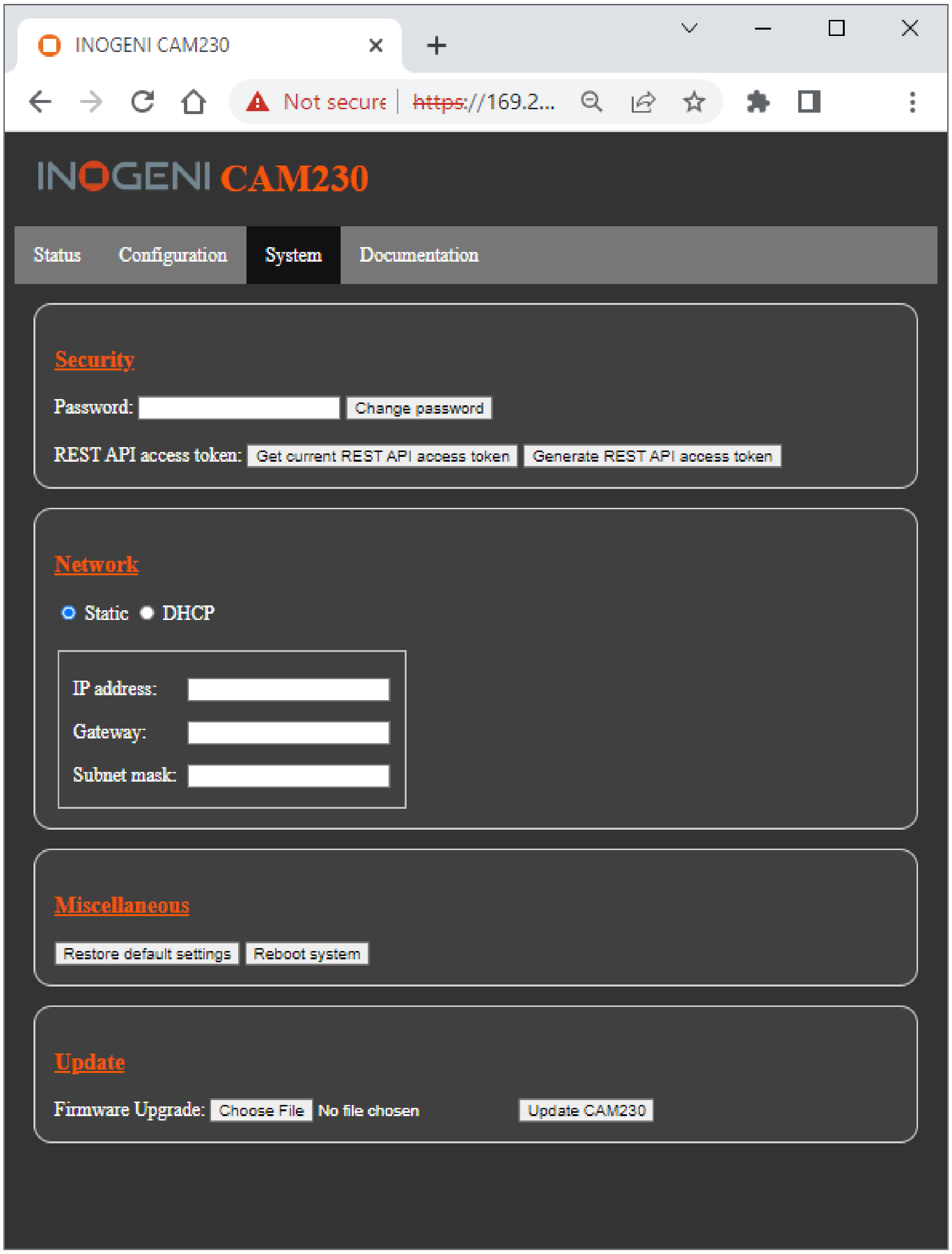
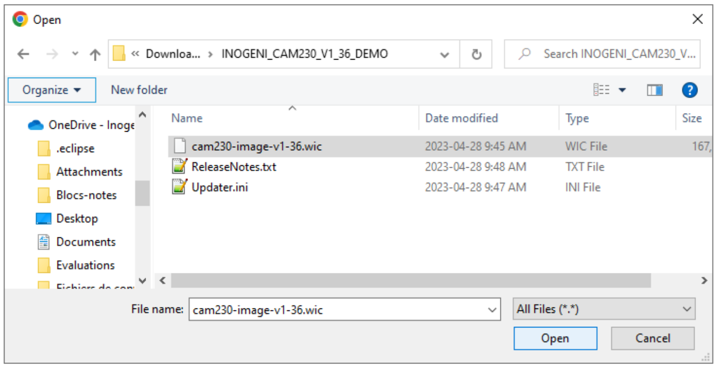


Figure 7: System preview

To update your system, please do the following :

- Click on the “Choose File” button and browse to the WIC file downloaded from our website.

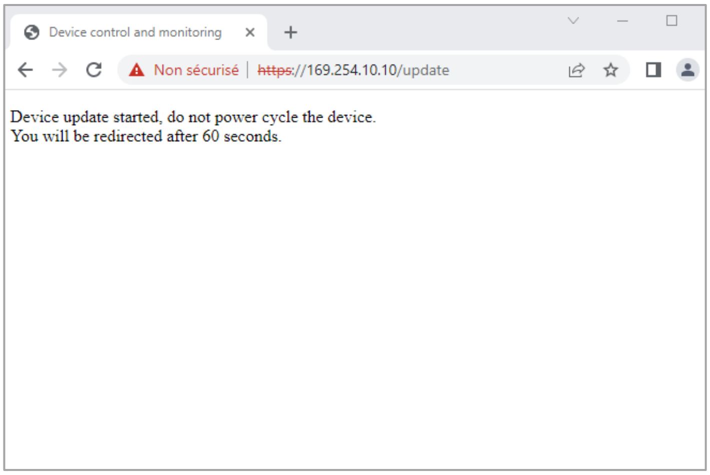


Name	Date modified	Type	Size
cam230-image-v1-36.wic	2023-04-28 9:45 AM	WIC File	167,
ReleaseNotes.txt	2023-04-28 9:48 AM	TXT File	
Updater.ini	2023-04-28 9:47 AM	INI File	

File name: cam230-image-v1-36.wic

All Files (*.*)

- Click on “Update CAM230” button to proceed to the update. The operation can take up to 1 minute. The device will reboot and browser will be refreshed.


Device control and monitoring
Non sécurisé | ~~https~~://169.254.10.10/update

Device update started, do not power cycle the device.
You will be redirected after 60 seconds.

The **DOCUMENTATION** tab will allow you :

- Get to the latest user guide.
- Go to product webpage.

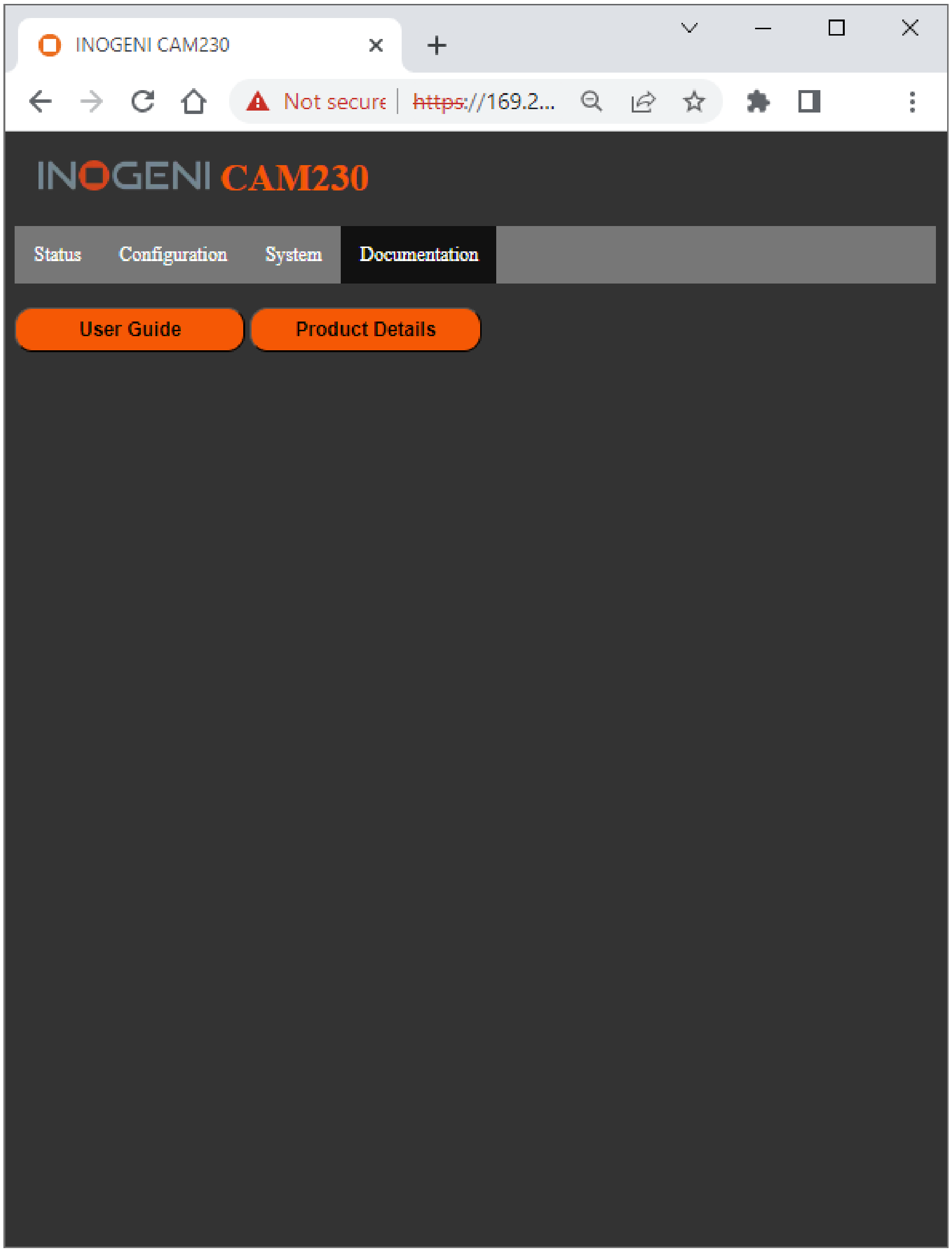


Figure 8: Documentation preview

The first time you access the webpage, your web browser is likely to complain that the connection is insecure. The reason for this is because we are using self-signed HTTPS certificates, because certificate providers will not provide certificates for addresses that are not globally accessible.

The webpage can set HDMI resolution, USB video input format, webpage password, or the REST API access token. Please note that in the case of the REST API token, we can only ask for the device to generate a new randomly generated token. It can also be used to upgrade the device firmware.

REST API

The response will usually be JSON formatted with a “message” field containing a JSON string explaining the cause of the error or “success” in case of success. Note that we are using self-signed certificates.

You can enable a bearer authentication in the HTTP header (Authorization: Bearer <token>) through our configuration page to increase security on the API.

There will be a return code to each call with the following commands:

```
200 => success
400 => error
401 => authorization error
```

Here is the complete list of commands supported through the REST API (excluding password change, firmware update, bearer token get/set):

Command URL / Description	Body arguments	Return body

HTTP GET/POST
https://<IP>/api/v1/rstr
Restore default settings (including password and REST API token)
		{
		"message": <String>
		}

HTTP GET/POST
https://<IP>/api/v1/reset
Reset/reboot the device
		{
		"message": <String>
		}

HTTP GET
https://<IP>/api/v1/version
Returns firmware version
		{
		"major": <Integer>,
		"minor": <Integer>
		}

HTTP GET **https://<IP>/api/v1/status** Returns devices, video/audio inputs and HDMI output status		JSON object with multiple fields

HTTP GET/POST
https://<IP>/api/v1/pan
Relative pan of the selected camera
	pan=<pan>
	The sign specifies the direction.
	We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allow. We recommend using values between -10 and 10.
		{
		"message": <String>
		}

HTTP GET/POST
https://<IP>/api/v1/tilt
Relative tilt of the selected camera
	tilt=<tilt>
	The sign specifies the direction.
	We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allow. We recommend using values between -10 and 10.
		{
		"message": <String>
		}

HTTP GET/POST
https://<IP>/api/v1/zoom
Relative zoom of the selected camera
	zoom=<zoom>
	The sign specifies the direction.
	We multiply the argument by the camera smallest step, and if the speed is too fast, we go as fast as the camera allow. We recommend using values between -10 and 10.
		{
		"message": <String>
		}

HTTP GET/POST https://<IP>/api/v1/setHdmi
Set HDMI output mode.
	`mode=<mode>`
	`<mode>` options:
	`0 => 1080P60`
	`1 => 1080P50`
	`2 => 720P60`
	`3 => 720P50`
	`4 => 4K24`
	`5 => 4K25`
	`6 => 4K30`
		`{ "message": <String> }`

HTTP GET/POST https://<IP>/api/v1/setVideoFormat
Set video input format.
	`x-www-form-urlencoded`
	`input=<inputPort>`
	`format=<formatIndex>`
	`<inputPort>` options:
	`1 => USB input #1`
	`2 => USB input #2`
	`4 => Test pattern`
	`<formatIndex>` options:
	See `getVideoFormats` command for available formats.
		`{ "message": <String> }`

HTTP GET https://<IP>/api/v1/getVideoFormats
Get video input format.
	`input=<input>`
	If input argument is not provided, will use currently selected input
	`<input>` options:
	`1 => USB input #1`
	`2 => USB input #2`
	`3 => HDMI input`
	`4 => Test pattern`
		JSON array containing all formats for requested input.

HTTP GET/POST https://<IP>/api/v1/setAudioInput Set audio input.	`input=<inputIndex>` The argument specifies the index of the audio input. To see available inputs, use “status” command.	`{ "message": <String> }`

HTTP GET/POST https://<IP>/api/v1/setAudioInputMode
Set audio input switching mode.
	`mode=<mode>`
	`<mode>` options:
	`0 => First connected audio device with fallback`
	`1 => Audio follows video selection`
		`{ "message": <String> }`

HTTP GET/POST https://<IP>/api/v1/setVideoInput
Set video input.
	`input=<input>`
	`<input>` options:
	`1 => USB input #1`
	`2 => USB input #2`
	`3 => HDMI input`
	`4 => Test pattern`
	The argument specifies the index of the video input. To see available inputs, use “status” command.
		`{ "message": <String> }`

HTTP GET/POST https://<IP>/api/v1/setVideoInputMode
Set video input switching mode.
	`mode=<mode>`
	`<mode>` options:
	`0 => automatic switching`
	`1 => manual switching`
	`2 => automatic switching (disable remote)`
		`{ "message": <String> }`

HTTP GET/POST https://<IP>/api/v1/disableSerialInterface Disable serial interface.	`disable=<disable>` If `<disable>` is `1`, will disable serial interface API to give IP interface exclusive access to serial port, otherwise serial interface API is enabled. When IP interface has exclusive access to serial port, user must use the “serialRead” and “serialWrite” commands.	`{ "message": <String> }`

HTTP GET https://<IP>/api/v1/serialRead
 	
Read serial data from RS232.
		{
		"message": <String>
		}
		
		Message field contains characters read from RS232

HTTP GET/POST https://<IP>/api/v1/serialWrite
 	
Write serial data to RS232. Giving content to write in URL is not supported.
	<Content to write>	{
		"message": <String>
		}

HTTP GET/POST https://<IP>/api/v1/setNetwork
 	
Configure network settings.
	x-www-form-urlencoded
	<mode> options:
	static => addressing is static
	dhcp => use DHCP addressing
	
	If mode is static, must provide following args:
	<ip> option:
	String defined IP address. Example:
	192.168.0.20
	
	<netmask> option:
	String defined netmask address. Example:
	255.255.0.0
	
	<gateway> option:
	String defined gateway address. Example:
	192.168.0.1
		{
		"message": <String>
		}

HTTP GET/POST https://<IP>/api/v1/enUsbCdcNcm
 	
Enable/Disable the USB CDC-NCM interface over USB2.0 output.
	enable=<enable>
	
	<enable> options:
	0 => OFF
	1 => ON
		{
		"message": <String>
		}

HTTP GET/POST https://<IP>/api/v1/enUsbOutSpeaker
 	
Enable/Disable the USB speaker interface over the USB 2.0 output.
	enable=<enable>
	
	<enable> options:
	0 => OFF
	1 => ON
		{
		"message": <String>
		}

HTTP GET/POST https://<IP>/api/v1/enUsbOutMic
 	
Enable/Disable the USB microphone interface over the USB 2.0 output.
	enable=<enable>
	
	<enable> options:
	0 => OFF
	1 => ON
		{
		"message": <String>
		}

It is also possible to embed arguments to an API call inside the URL to ease configuration with some control systems with the following topology:

GET https://<IP>/api/v1/**<COMMAND>**?**<ARG1>=value**&**<ARG2>=value**

where **<COMMAND>**, **<ARG1>** and **<ARG2>** are command and associated arguments.

For example, using the **setVideoFormat** command, you can issue the following request:

GET https://<IP>/api/v1/**setVideoFormat**?**inputPort=1**&**formatIndex=0**

This request will set the input #1 to format index 0.

The following commands allow to perform password management, bearer token management and firmware update. The authentication used is basic auth, and we use the same user and password as the webpage (default user=admin password=admin).

Command URL / Description	Body arguments	Return body

`HTTP POST`
`https://<IP>/api/v1/changePassword?password=<newPassword>`
Change the webpage password to <newPassword>.
		`{`
		`    "message": <String>`
		`}`

`HTTP GET`
`https://<IP>/api/v1/getAccessToken`
Return the bearer token.
		`{`
		`    "token": <String>`
		`}`
		If no bearer token is set, the “token” field will be null.

`HTTP POST`
`https://<IP>/api/v1/generateAccessToken`
Generate random access token.
		`{`
		`    "message": <String>`
		`}`

`HTTP POST`
`https://<IP>/api/v1/update`
Sends update file to device.
	Must use formdata body.
	The key must be myFile, and the value is of type file. We expect a .wic file that should be present in our official update packages
		`{`
		`    "message": <String>`
		`}`

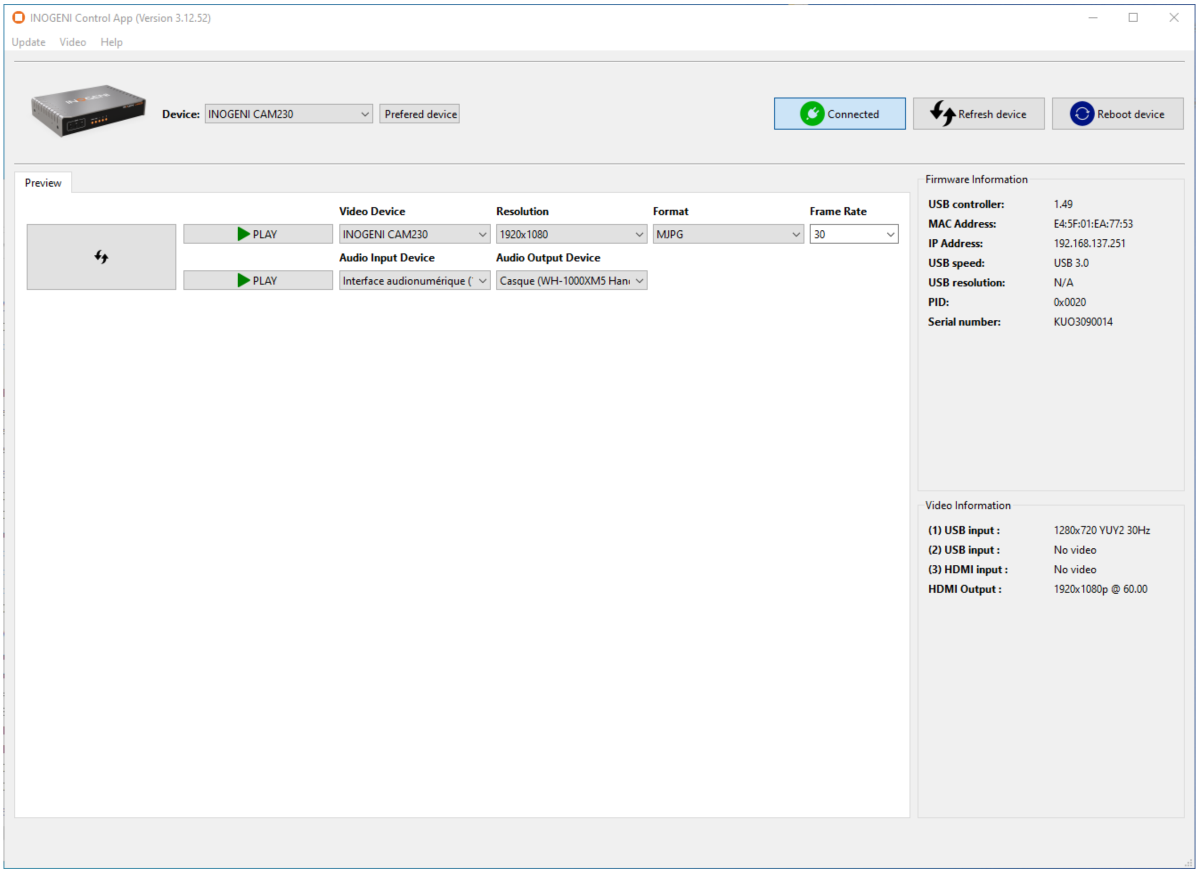
The bearer token is generated using a random process. The format of the bearer token only supports the following:

- Alphanumeric (`A` to `Z`) upper and lowercase characters.
- Numbers `0-9`.

INOGENI CONTROL APP

You can use our Control App to monitor firmware information and upgrade your unit.

NOTE: You need to use the USB-B to USB-A cable provided with the box for the Control App to detect the unit.



Device: INOGENI CAM230

Video Device	Resolution	Format	Frame Rate
INOGENI CAM230	1920x1080	MJPG	30

Audio Input Device	Audio Output Device
Interface audionumérique (	Casque (WH-1000XM5 Han

Firmware Information

USB controller:	1.49
MAC Address:	E4:5F:01:EA:77:53
IP Address:	192.168.137.251
USB speed:	USB 3.0
USB resolution:	N/A
PID:	0x0020
Serial number:	KUO3090014

Video Information

(1) USB input :	1280x720 YUY2 30Hz
(2) USB input :	No video
(3) HDMI input :	No video
HDMI Output :	1920x1080p @ 60.00

Figure 9: INOGENI Control App preview

MECHANICAL SPECIFICATIONS

You can find the mechanical specification here that lists the holes. All dimensions are in **mm [in]**.

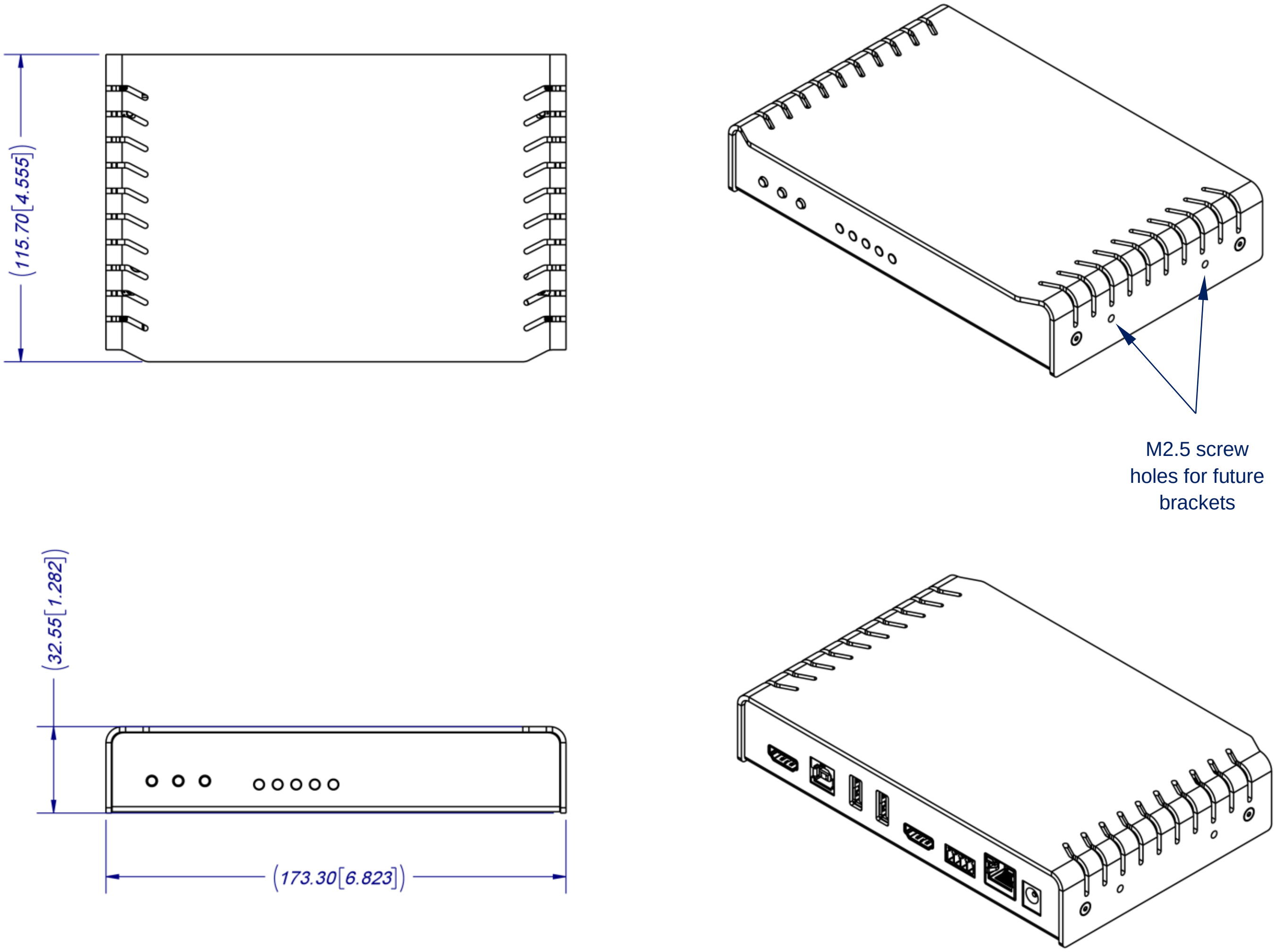


Figure 10: Top plate dimensions

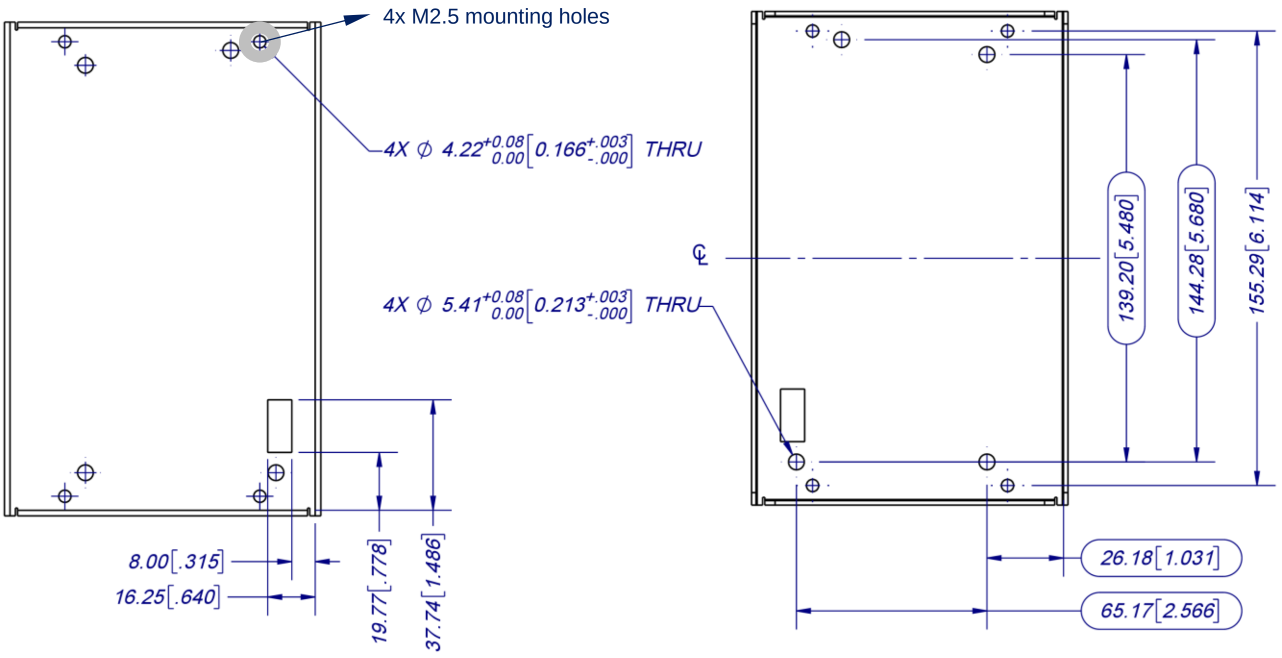


Figure 11: Bottom plate dimensions and holes positions

DIP SWITCHES

Here you can find the behavior of the DIP switches located at the back of the unit.

Switch	Position	Description
SW1	OFF	For future use.
	ON	
SW2	OFF	For future use.
	ON	
SW3	OFF	For future use.
	ON	
SW4	OFF	For future use.
	ON	
SW5	OFF	Reserved.
	ON	
SW6	OFF	Disable 5V on terminal block
	ON	Enable 5V on terminal block. This switch must be set to power up the connected remote.

TROUBLESHOOTING SECTION

Here is the troubleshooting section for the device.

Problem	Resolution
The device keeps rebooting when we connect USB powered cameras. The device is connected to a PoE switch.	If you connect high consuming USB cameras that are powered by the USB bus, we recommend using the provided power supply to accommodate the power demand.
We cannot detect the USB output on the PC when using it with a USB extender	The device has a USB2.0 interface only. Make sure that the USB extender you intend to use supports USB2.0. You can use our own set of USB extenders that are proven to work with our devices. https://inogeni.com/support/compatible-usb-extenders/
We cannot get the HDMI feed to go through the device.	Please do the following checklist : - Make sure HDMI connection is correct. HDMI cable should be within 30ft. Otherwise, you need to have an active HDMI extension. - Monitor if the HDMI led is ON. - Monitor the video resolution through our app or through the web interface if it is detected properly. - Make sure that the HDMI feed is not an encrypted HDCP source like a blu-ray or set-top box. The HDMI input is compatible with HDMI cameras.
My camera software running on my computer is not detecting the camera while the CAM230 is connected in line.	Unfortunately, this is the expected behavior since the PC is agnostic of the USB camera. However, the device supports all UVC controls (pan, tilt and zoom controls) and can route them to the selected camera.
LAN interface is not working properly. **Firmware updates from LAN interface takes a long time or hang/timeout.**	Make sure the port that is connected to the LAN interface is using 100Mbps half-duplex. Slow updates through LAN interface can be caused by duplex mismatch. The LAN interface does not do any autonegotiation.

SUPPORT

Engineered by video professionals, for video professionals, it is your most compatible USB 3.0 device. INOGENI expertise at your fingertips:

- **Expert Technical Support team** at support@inogeni.com for immediate help or if you have any technical question about our products.
- Extensive **Knowledge Base** to learn from other customers' experiences.

INOGENI name and logo are trademarks or registered trademarks of INOGENI. Use of this product is subject to the terms and conditions of the license and limited warranty in effect at the time of purchase. Product specifications can change without notice.

INOGENI, Inc.
1045 Avenue Wilfrid-Pelletier
Suite 101
Québec, QC, Canada, G1W0C6
(418) 651-3383

CERTIFICATIONS

FCC Radio Frequency Interference Statement Warning
This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:
(1) this device may not cause harmful interference, and
(2) this device must accept any interference received including interference that may cause undesired operation.

IC Statement
This Class A digital apparatus complies with Canadian CAN ICES-3(A)/NMB-3(A).

CE Statement
We, INOGENI Inc., declare under our sole responsibility that the CAM230, to which this declaration relates, is in conformity with European Standards EN 55032, EN 55035, and RoHS Directive 2011/65/EU + 2015/863/EU.

UKCA Statement
This device is compliant with the Electromagnetic Compatibility Regulations 2016 No. 1091 as part of the requirements leading to the UKCA marking.

WEEE Statement
The European Union has established regulations for the collection and recycling of all waste electrical and electronic equipment (WEEE). Implementation of WEEE regulations may vary slightly by individual EU member states. Please check with your local and state government guidelines for safe disposal and recycling or contact your national WEEE recycling agency for more information.

RCM Statement
This device is compliant with Regulator Compliance Mark (RCM) certification.