PUMATRONIX

PUMATRONIX

CAPTURE DEVICES

ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+, VTR 600, ITSCAM 450 AND ITSCAM 450+

Integration

Pumatronix Equipamentos Eletrônicos Ltda.

Rua Bartolomeu Lourenço de Gusmão, 1970. Curitiba, Brasil

Visit our website https://www.pumatronix.com

Send comments on this document to suporte@pumatronix.com

Information contained in this document is subject to change without prior notice.

Pumatronix reserves the right to modify or improve this material without obligation to notify the changes or improvements.

Pumatronix grants permission to download and print this document, provided that the electronic or physical copy of this document contains the full text. Any changes to this content are strictly prohibited.

Changes History

Date	Revision	Updated content
06/28/2022	1.0.0	First issue
04/02/2024	1.1.0	Inclusion of ITSCAM 600 FHD, VTR 600 and ITSCAM 450 products; Updates for firmware versions 1.3.0 to 1.6.0
04/30/2024	1.1.1	Inclusion of ITSCAM 450+ product; Update for firmware version 1.7.1
05/24/2024	1.1.2	Inclusion update of ITSCAM 450+ product; Inclusion of the Automatic Iris application
08/09/2024	1.2.0	Update of REST API information
09/17/2024	1.3.0	Update for firmware version 1.7.3; Updates for Cougar Protocol; Update of digital signature description; Update of image configuration values
10/03/2024	1.4.0	Update for firmware version 1.7.4
11/30/2024	1.4.1	Inclusion of ITSCAM 600+ product; Reordering of chapters according to the web interface; Detailing of the Maintenance and Upgrade menus; Insertion of the Digital Signature validation process; Addition of available REST API fields
02/04/2025	1.4.2	Updates for firmware version 1.7.5
05/23/2025	1.4.3	Updates for firmware version 1.7.6 (SAD-836)

Overview

The purpose of this document is to guide the developer in the use of the operating interfaces that allow configuration of the behavior of the ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+, VTR 600, ITSCAM 450 and ITSCAM 450+ devices. This document details the options available via the web interface, the Pumatronix Protocol or the Cougar Protocol.

Depending on the firmware version applied to the device being accessed, the web access interface differs, and some functions may only be available in the latest versions.

Summary

1. Presentation of the Web Interface

The Web interface makes it possible to evaluate the images generated and configure the devices. Access to the interface requires information:

User	*admin*
Password	*1234*

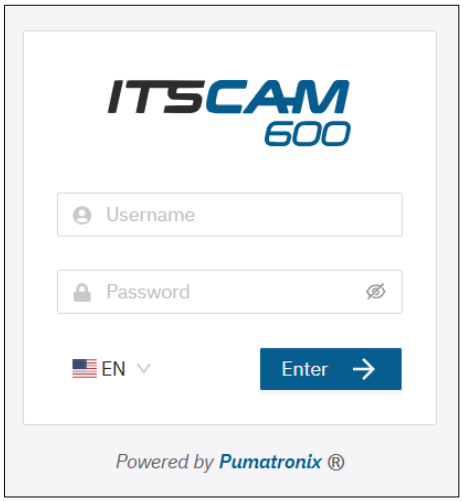


Figure 1 - Login screen

As a form of security, it is recommended to change the device's default password by accessing the System > Users menu:

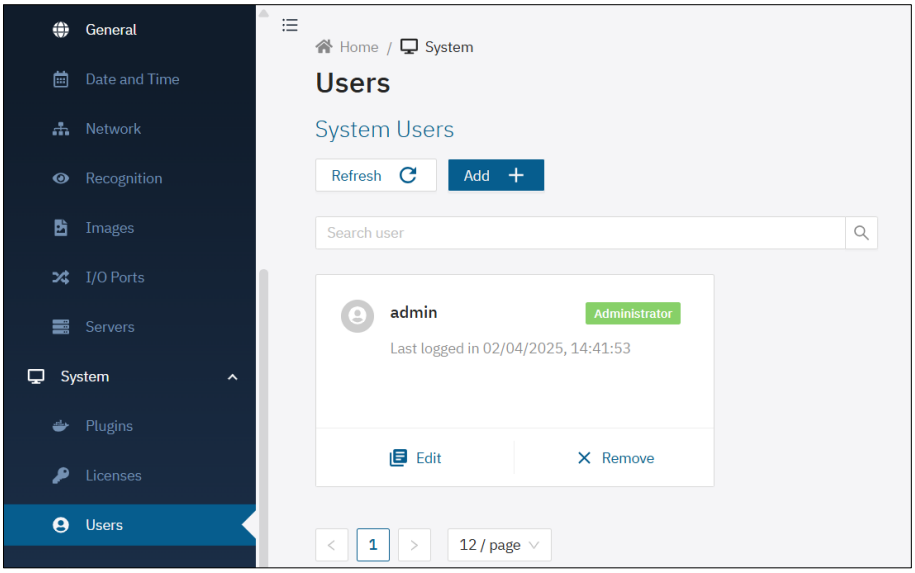
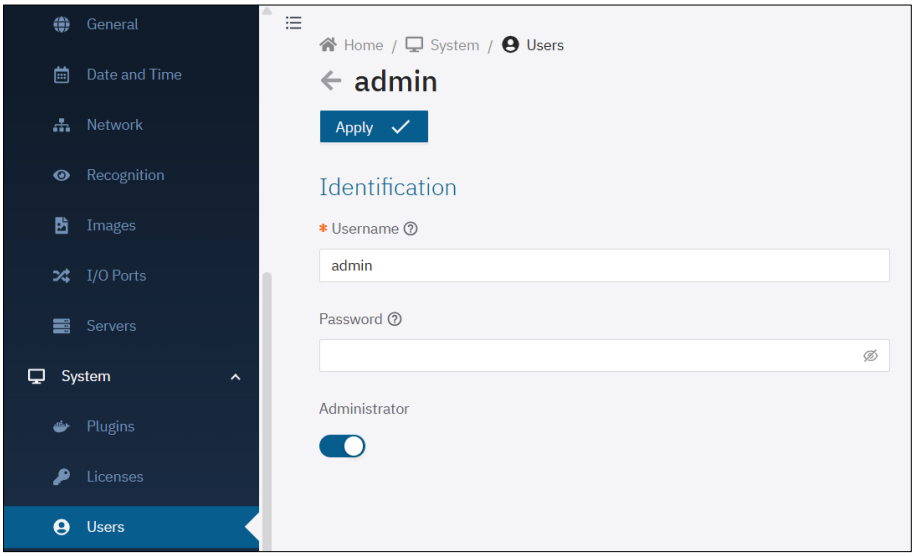


Figure 2 - Screen displayed when accessing System>Users

1) Click on *Edit*;
2) Enter a *Name* and create a new *Password*, which can contain between 4 and 200 characters, including numbers, letters and special characters;
3) Finish by clicking on *Apply*:



1.1. Web Interface Help

If you still have doubts about the functionality of any configuration in the web interface, the question mark icon displays help with an explanation, examples or the recommended configuration for the device when the cursor is placed on it:

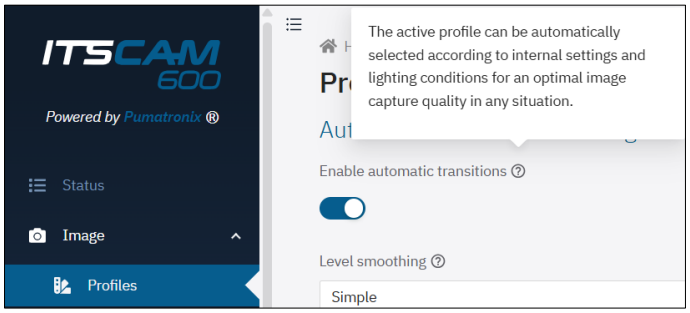


Figure 3 - Help displayed by positioning the cursor on the icon

Figure 4 – Interface help

1.2. Changing the Interface Language

The Web interface can be displayed in Portuguese or English by selecting it on the login screen or by accessing it from the Equipment > General menu, on *Standard language*:

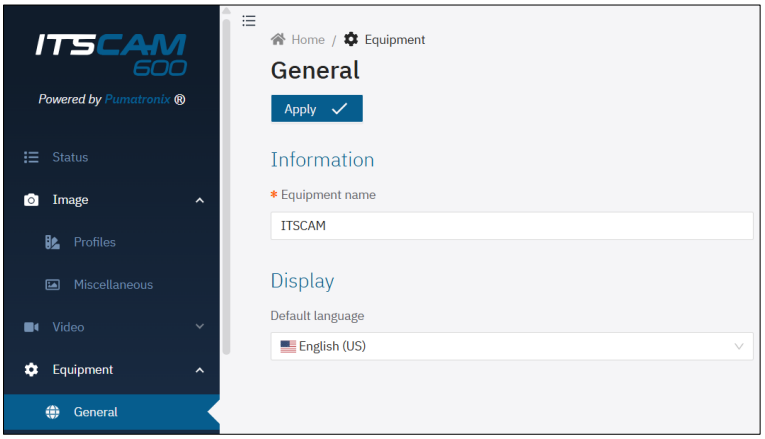


Figure 5 – Equipment > General menu screen

1.3. Home Screen

The home screen displays the *Status* screen of the device in operation, as well as the permanent usability options on the interface:

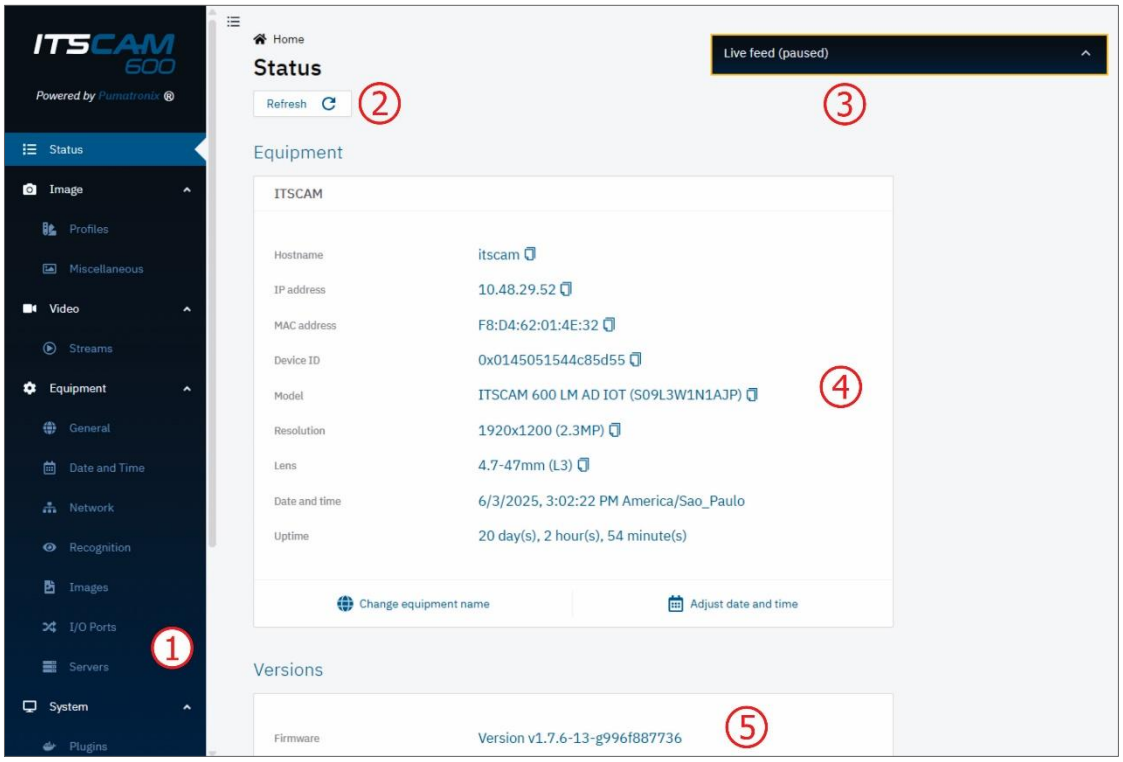


Figure 6 - Options available on the interface and in the Status window: 1) Menu bar; 2) Action button available for functionality; 3) Pop-up live view window; 4) Equipment data, including detailed Runtime and quick setting buttons, 5) Data on installed firmware versions

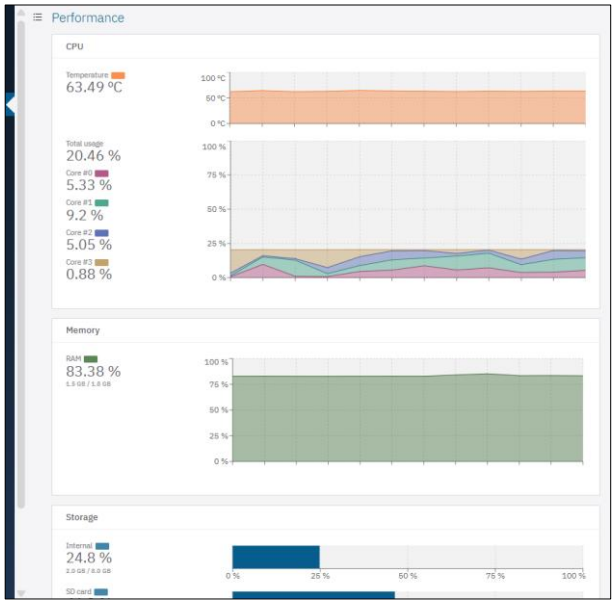


Figure 7 - Available data on the Current Status of CPU, Memory and Storage Performance

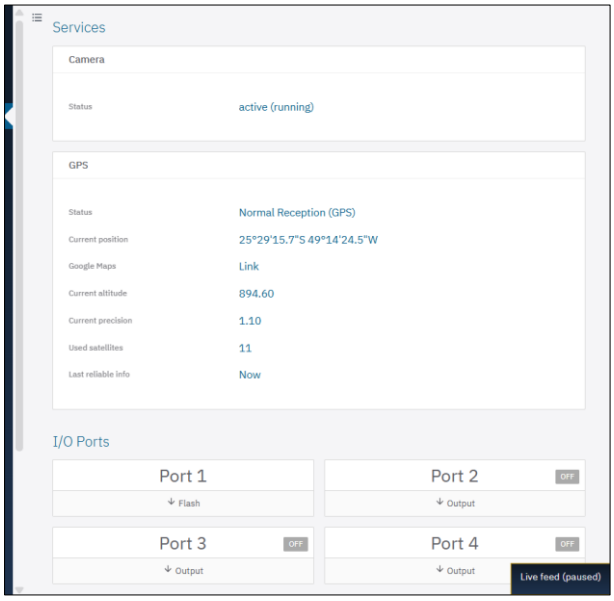


Figure 8 - Available data on the Current Status of Camera and GPS Services and port connections*

* When the device is able to establish communication with a GPS network, the main Geolocation information is displayed as the Figure 8 above.

1.3.1. Live Viewing

The web interface has a live image viewer, available in a pop-up window that can be moved to any place on the interface screen or even expanded to occupy the entire window.

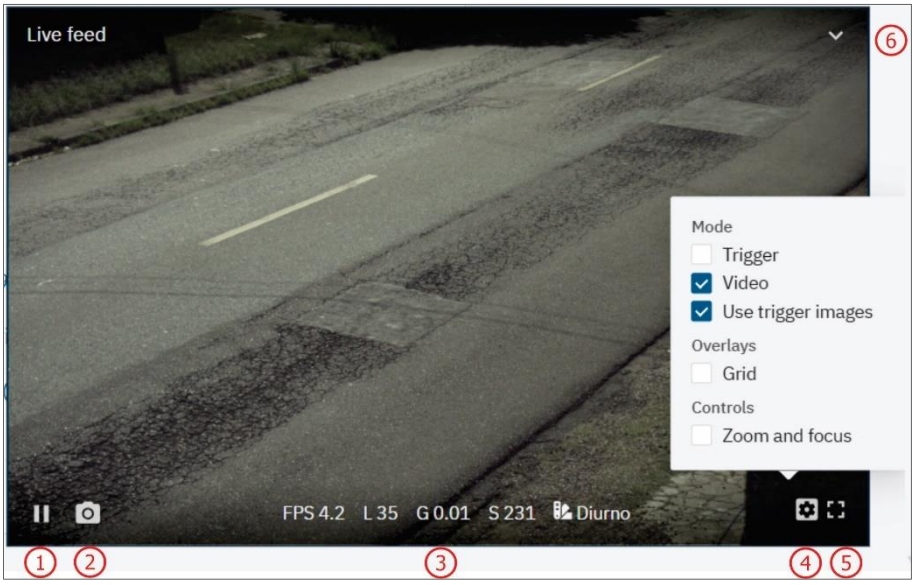
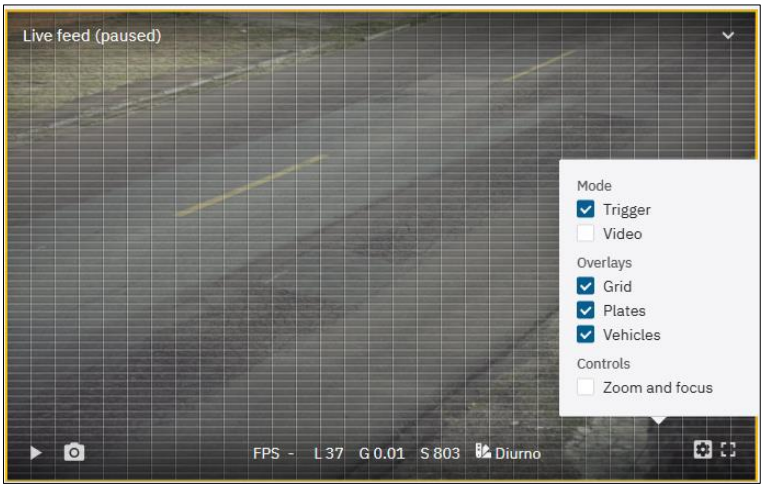


Figure 9 – Pop-up live view window: 1) Pause live view; 2) Take photo; 3) Level of FPS, Gain, Shutter and Photo Profile values active; 4) View settings; 5) Full screen; 6) Minimize window

1) Click *Pause live view* (1) to pause the live video and the last captured image remains fixed;
2) Click on the *Take photo* (2) button to download an instant JPEG file generated from the image being captured by the device;

3) View, at the bottom of the pop-up window (3), the *Level, Gain and Shutter* values that are being applied to the device and the *Image Profile* that is active;
4) Click on the Display settings *(4)* option to access the image display options in the pop-up window:
    a. Selecting *Video Mode* displays the real-time image captured by the device and enables the options:
        i. *Use trigger images*: when enabled, it is possible to have video during the night synchronized with the triggering of the illuminators (the trigger must be enabled), when disabled, it is possible to have more fluid video (it does not depend on trigger configuration). This option is automatically enabled when configured in the *Video Settings (Streams)*;
        ii. *Grid*: overlaying a grid on the image;

    b. Selecting *Trigger* displays the last image captured in the configured parameters and enables the options:
        i. *Grid*: overlaying a grid on the image;
        ii. *Plates*: overlaying a recognition area on the plates;
        iii. *Vehicles*: displays a recognition area about vehicles;



    c. *Zoom and focus*: displays two bars on the side of the pop-up window, which allows the adjust of the image Zoom and Focus.

5) Click on the *Full Screen* option (5) to expand the view and make interface full screen.

2. Image Settings

The steps for configuring the image capture devices are presented according to the sequence of the interface and must be carried out after physical installation, respecting the Prerequisites and Necessary Conditions for installation presented in the Product Manual.

2.1. Image Profiles

The *Daytime* and *Nighttime* image capture configuration profiles are the factory defaults. The settings applied to each parameter of an image profile can be accessed via the *Image > Profiles* menu:

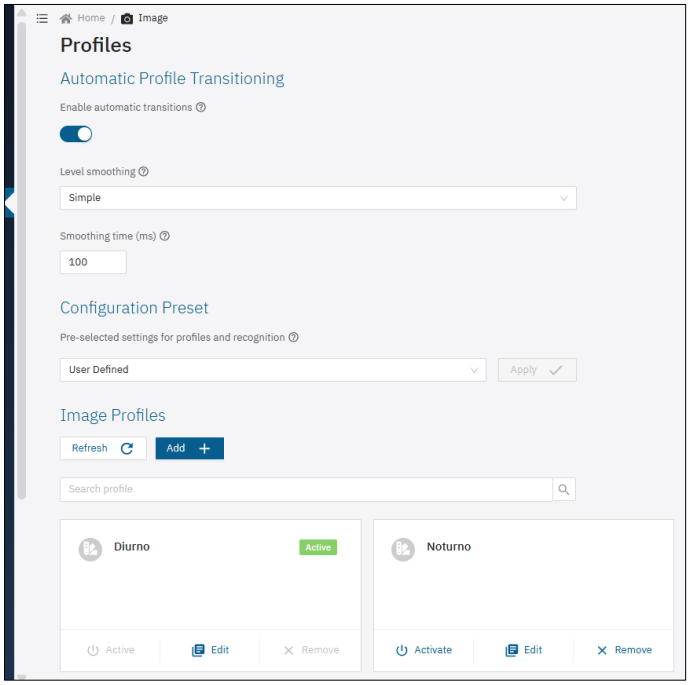
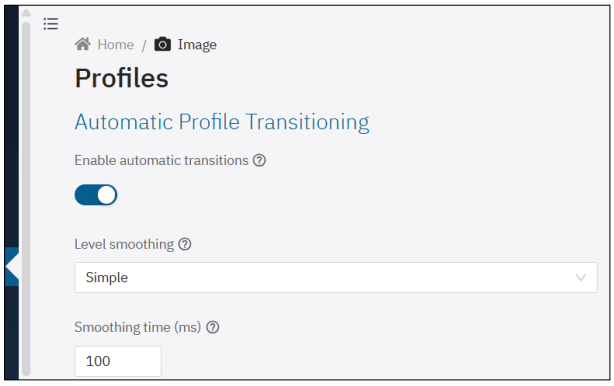


Figure 10 – Image Menu > Profiles

1) Enable the *Enable automatic transitions* option so that the profile change is applied automatically according to the internal settings and lighting conditions, which helps to achieve optimum image quality in any situation:
    a. Use one profile for captures with ambient light (daytime) and another for captures with artificial lighting (nighttime), as recommended, and it is possible to register up to four *Image Profiles*, each with its own set of settings;
    b. Select the *Level Smoothing* to be applied when switching between profiles;
        i. *None:* Level changes are made instantly when profiles are changed;
        ii. *Simple:* level change is performed using a smoothing time during the profile change;
2) Set the *Smoothing tome* in the automatic transition between profiles, in milliseconds;



3) Select any *Configuration Preset* with the pre-selected settings for profile and recognition, from the options *Mobile Equipment*, *Fixed Equipment* or *User-Defined*, which allows manual adjustments by the user. They only apply if there are profiles named *Day* and *Night*;
4) Click on *Edit* for one of the *Image Profiles* (Day or Night, for example) and the settings are available in tabs and saved automatically:
    a. In the tab labeled *General*, access the *Identification* and *Text Overlay* settings in the image;
    b. Identify the Profile by assigning a *Name* and adding a *Description*;
    c. Enable the *Snapshot overlay* option to apply a text overlay to all generated photos;
        i. Enter a String of up to 1024 characters in the *Overlay text* field. Go to the *Detailed help* field to check the values that can be entered as captions;

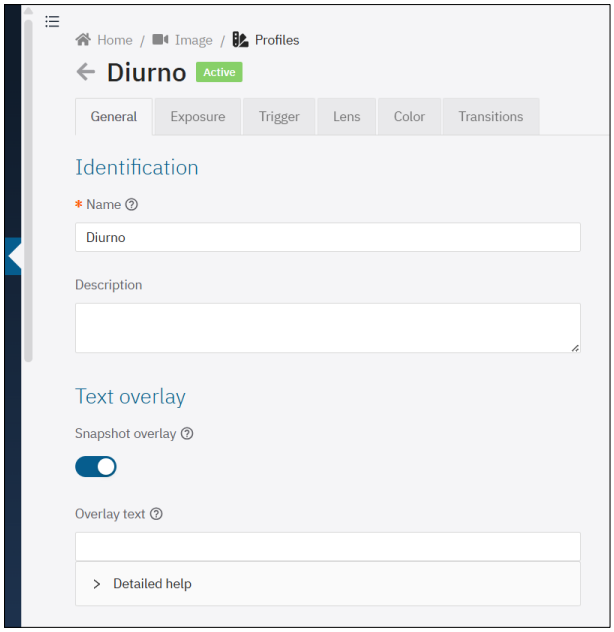
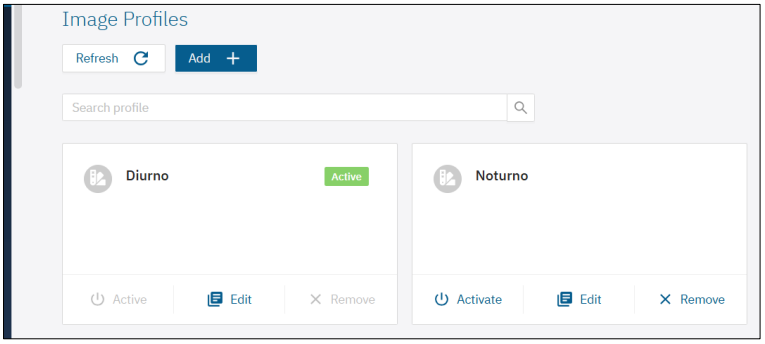


Figure 11 - Initial screen for editing the Day profile

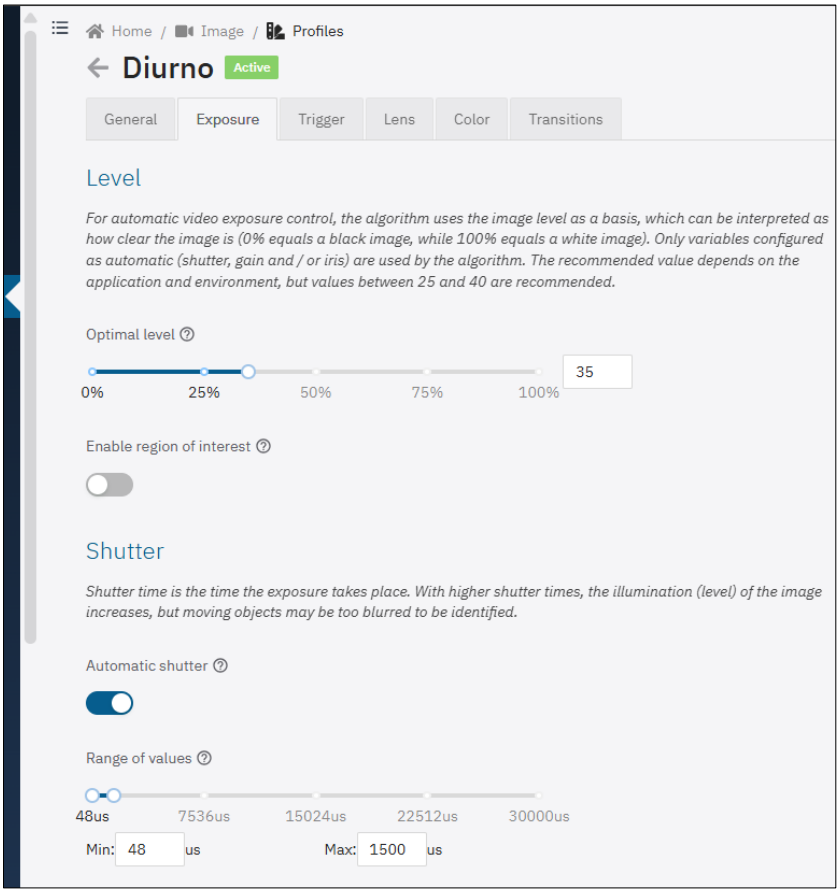
5) Click *Add +* on *Image Profiles* to create new image profiles (it is recommended to configure existing profiles before creating new ones).

2.1.1. Exposure

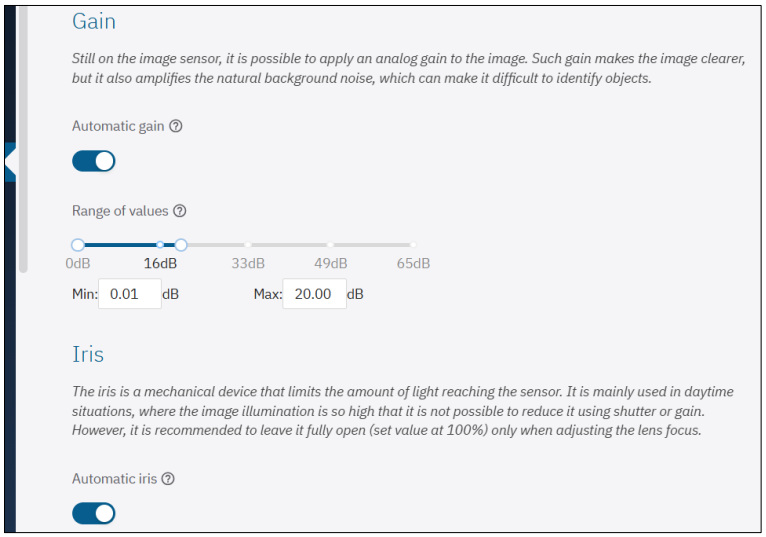
1) Find the profile to set up in the *Image > Profiles* menu and click *Edit*;



2) Select the *Exposure* tab and on this screen, the settings are automatically saved;
3) Select the target value for the level of image brightness that the device should try to achieve using the variables configured as automatic (shutter, gain and/or iris) in *Optimal level*, considering that 0% is equivalent to a black image, while 100% is equivalent to a white image, with values between 25 and 40 being recommended, depending on the environment;
4) Select the *Enable region of interest* option and define which region of the image should be considered for level calculation, especially in scenarios with regions that could hinder the level algorithm (e.g. lamps that are too bright, flashing or regions that are too dark);
5) Enable the *Automatic shutter* or select the value for the *Fixed shutter* in microseconds;
6) Select the automatic *Range of values* that can be applied by the *Optimal Level* algorithm, in microseconds;

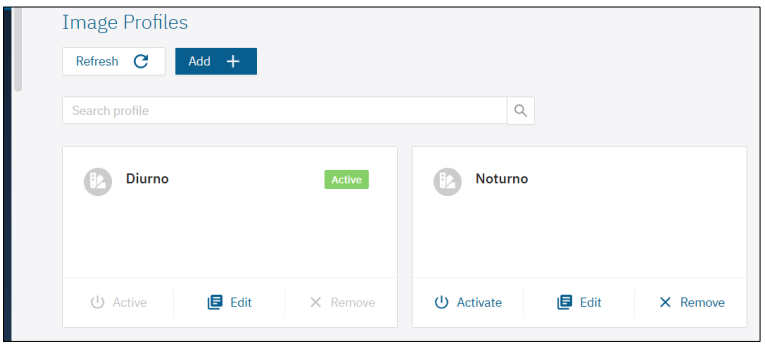


7) Enable *Gain* by clicking on the *Automatic Gain* option, in which the equipment will adjust the *Gain* (within the limits set by the user) to keep the image at the optimal level. When this option is deactivated, the *Gain* to be applied is a fixed value also defined by the user;
8) Select the *Range of values* that can be applied by the automatic level algorithm, in decibels;
9) Enable the *Automatic iris* option:

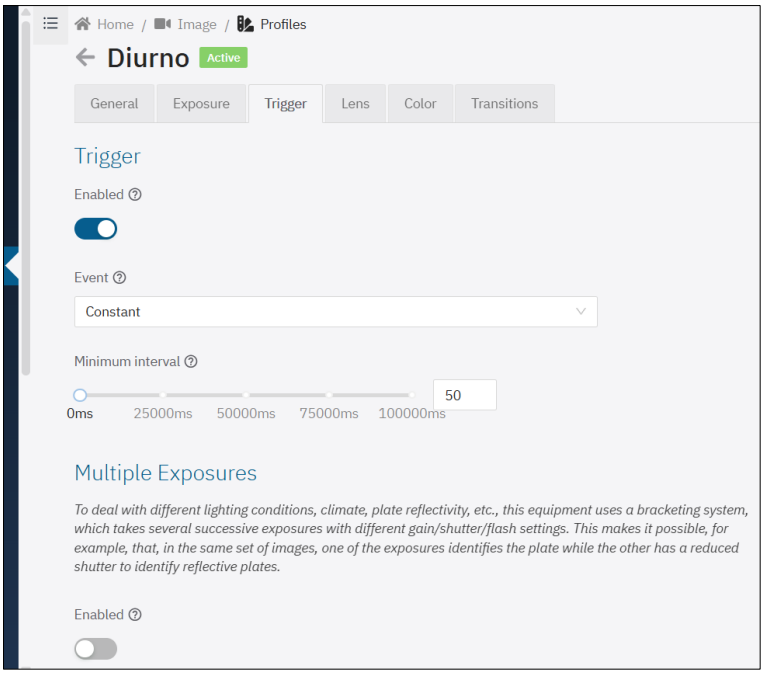


2.1.2. Trigger

1) Find the profile to set up in the *Image* > *Profiles* menu and click *Edit*;



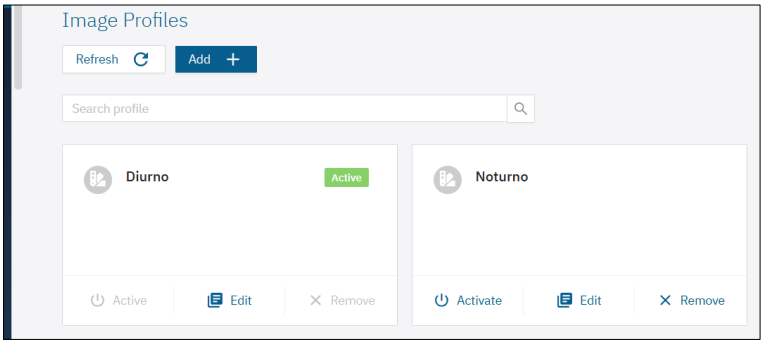
2) Select the *Trigger* tab and on this screen, the settings are automatically saved;



3) Select the *Enabled* option to configure image capture according to the configured *Event*
4) Select *Event* as *Constant* to generate a continuous stream of image captures at the highest rate the device can handle;
5) Select signal input *Event* from the options:
    a. *Rising edge:* generates an image capture when the signal is activated;
    b. *Falling edge:* generates an image capture when the signal is deactivated;
    c. *Rising and falling edge:* generates an image capture when the signal is activated or deactivated;
    d. *High level*: generates image captures continuously as long as the signal level is high;
    e. *Low level*: generates image captures continuously as long as the signal level is low;
    f. Select the input *Port* of the signal used for the trigger, configured in Inputs and Outputs;
6) Select the *Event* as *Motion* to activate the *Motion Detector* functionality, which generates image captures without the need for external sensors, while motion is being detected:
    a. Set the *Minimum interval* for separation between triggers, in milliseconds;
    b. Set the *Threshold* value (from 0 to 100) required to activate the *Motion Detector*;
    c. Select *Enable region of interest* to delimit the area of the image that should be considered in the motion calculation.

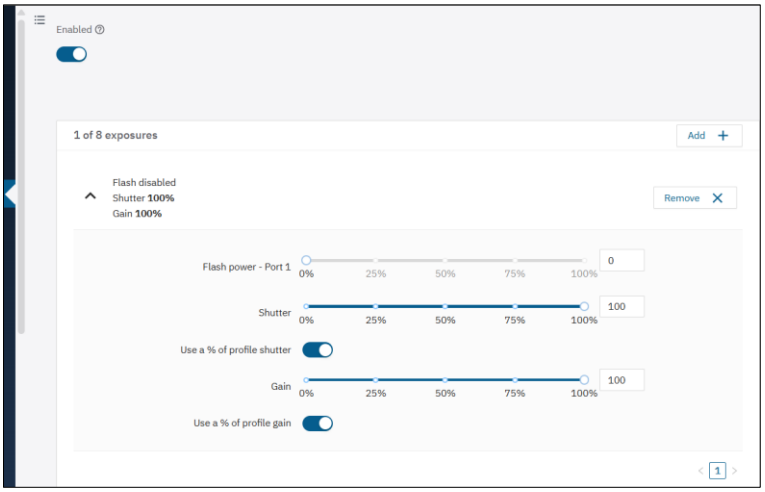
2.1.2.1. Multiple Exposures

1) Find the profile to set up in the Image > Profiles menu and click *Edit*;



2) Select the *Trigger* tab and on this screen, the settings are automatically saved;
3) Enable *Multiple Exposures* by clicking *Enabled* to take multiple shots simultaneously with different flash, shutter and gain settings. If disabled, only one exposure is made per trigger, without flash and with shutter and video gain;
4) Click on the *Add +* button to create an exhibition (up to 8 per profile);

5) Select the exposure to display its settings;



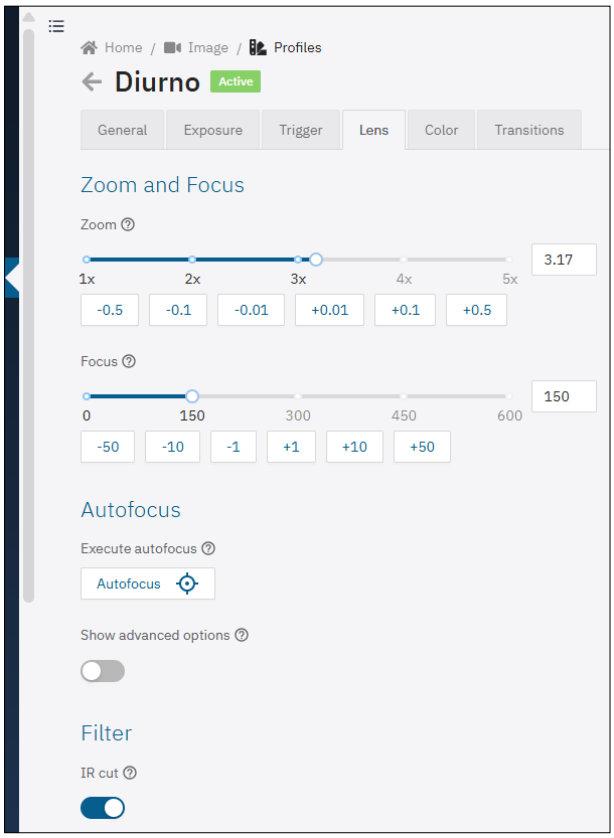
6) Select the *Flash power*, always corresponding to a percentage of the initial shot (this option is available for the Pumatronix ITSLUX range of lighting devices);
7) Select the *Shutter* percentage (exposure time of the image sensor), generating images with a variation in the amount of light captured;
8) Select the *Gain* percentage (digital post-processing), which allows to lighten or darken the images.

2.1.3. Lens

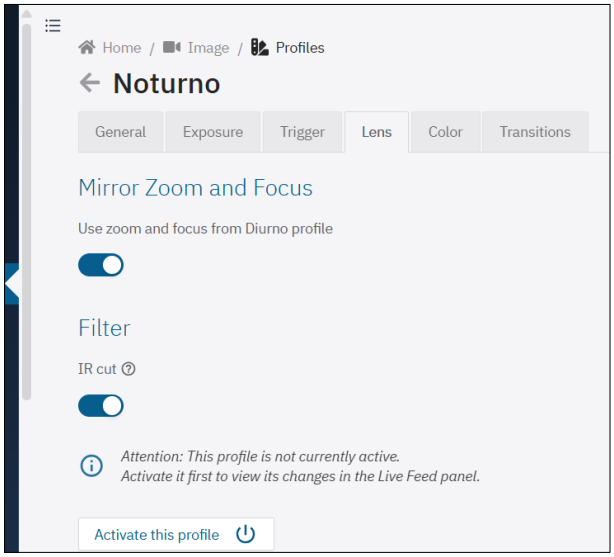
The *Lens* tab contains the *Zoom* and *Focus*, *Autofocus* and *Filter* settings.

Keep the Auto Iris option (available on the ITSCAM 600, ITSCAM 600 FHD and ITSCAM 450 products) disabled during the focus adjustment process for best results during the image setup process (available in the Exposure tab).

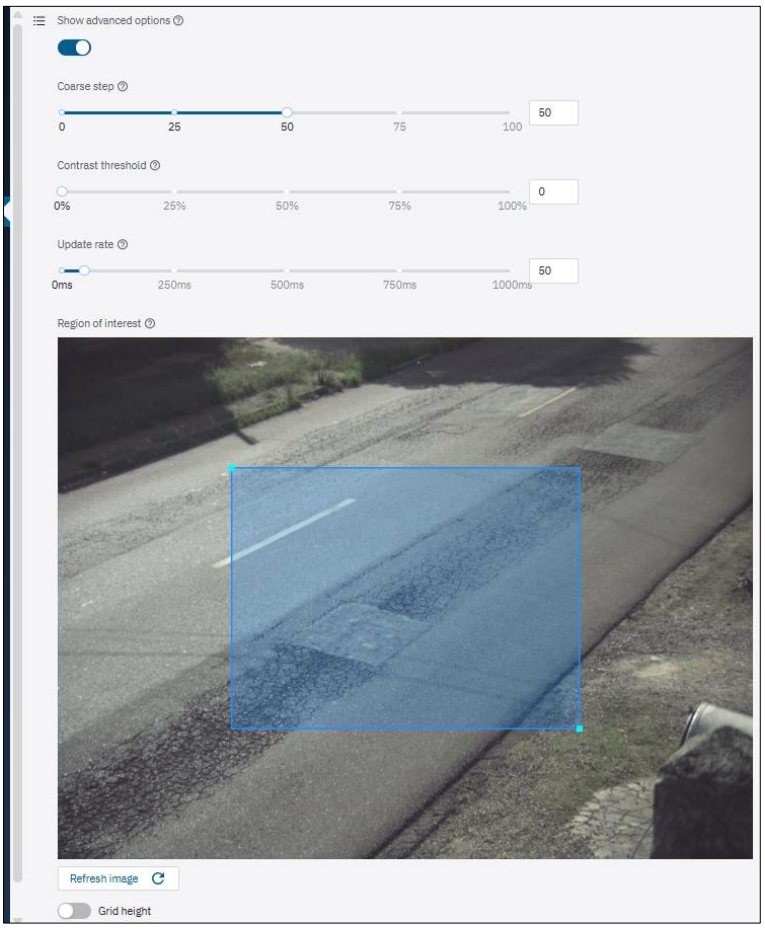
1) Select the *Lens* tab;



2) Check the *Zoom* and *Focus* settings in real time using the Pop-up Preview window;
3) Select the *Zoom* rate value, gradually increasing the available buttons until finding the ideal value;
4) Select the distance of the *Focus* lens, gradually increasing the available buttons until finding the ideal value;
    a. Copy the *Zoom* and *Focus* settings from the first profile configured, via the *Mirror Zoom and Focus* option, by enabling the *Use zoom and focus option from Daytime profile**, available on the *Lens* tab of the other profiles;



5) Adjust the automatic focus by clicking on *Autofocus* (only available on models with a powered lens), which modifies the focus position until finding the point of greatest image contrast. The button remains active while the algorithm is running and resets when it finishes. To stop the process, simply deactivate it by clicking the button again. It is suggested to place objects with notable edges at the place of use to help the algorithm;
6) Click *Show advanced options* to display the advanced *Autofocus* settings, only in the event of an algorithm malfunction;
    a. *Coarse step:* if the image is not in focus to begin with, the algorithm first runs through the entire focus range, with this value as the increment. Cannot edit this setting;
    b. *Contrast threshold:* the minimum contrast for the image to be considered in focus. Cannot edit this setting;
    c. *Update rate:* the time it takes for the focus to be correctly applied to the image;
    d. *Region of interest:* defines the region in which *Autofocus* will be executed, by default only the central region of the image is considered and can be edited by dragging the vertices of the default rectangle.



The advanced *Autofocus* options should only be adjusted if the algorithm is malfunctioning.

The *Long Step* and *Proximity Threshold* values are already factory-defined for the best performance and operation of the equipment, and it is not possible to edit these fields. If necessary, contact Pumatronix Technical Support.

7) Enable the *IR Filter* during the daytime to protect the image sensor and because infrared light can distort the image colors. When using infrared lighting devices, keep the *IR Filter* disabled, ensuring that the image is formed on the sensor;

8) Adjust the positioning of the device based on the images displayed in *Live View*. The suggested framing for two lanes in the image is to show the ends of the lane, as shown in the examples:

Figure 12 – Example of a daytime image with the suggested framing

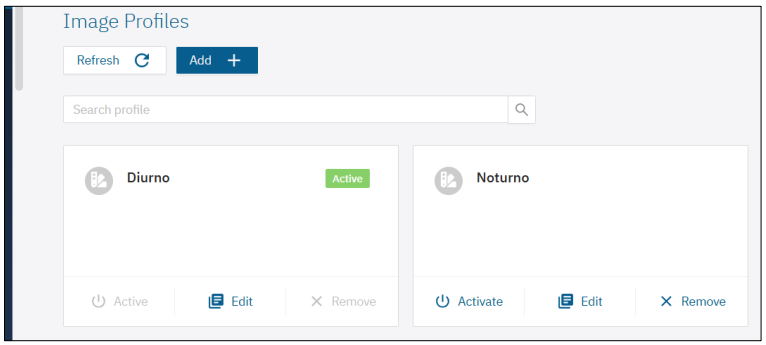
Figure 13 – Example of a nighttime image with the suggested framing

The Zoom and Focus adjustment corresponds to the Image Profile: Zoom and focus adjustment must be performed on all Profiles enabled on the device.

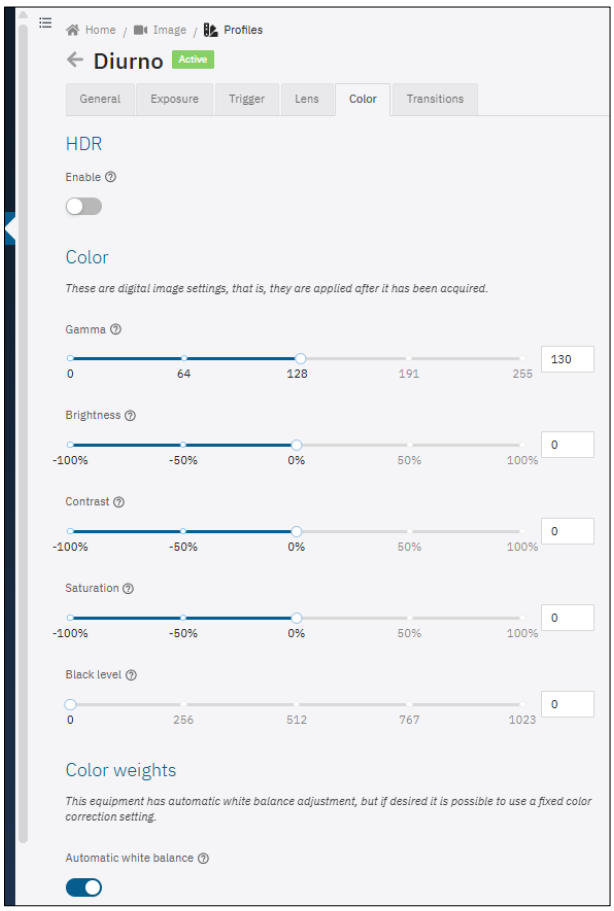
2.1.4. Color

It is not recommended to change the color settings, with the exception of the *Gamma* option.

1) Find the profile to set up in the *Image > Profiles* menu and click Edit;



2) Select the *Color* tab and on this screen, the settings are automatically saved;
3) Set the Gamma value, which adjusts the lighting of the image by increasing the gain to darker places than lighter ones;
4) Enable the *Auto White Balance* option and the Red, Green and Blue balance in the image will be adjusted automatically.

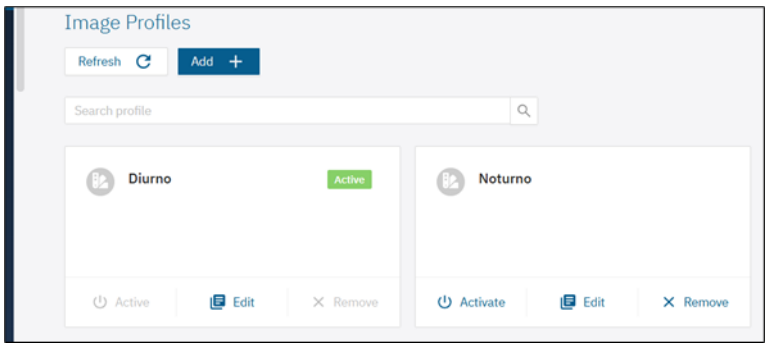


The *Brightness*, *Contrast*, *Saturation and Black Level* values are factory-set to the ideal values in normal image lighting conditions and adjustments are restricted in cases of images generated with poor quality. Please contact Technical Support for guidance on these color settings.

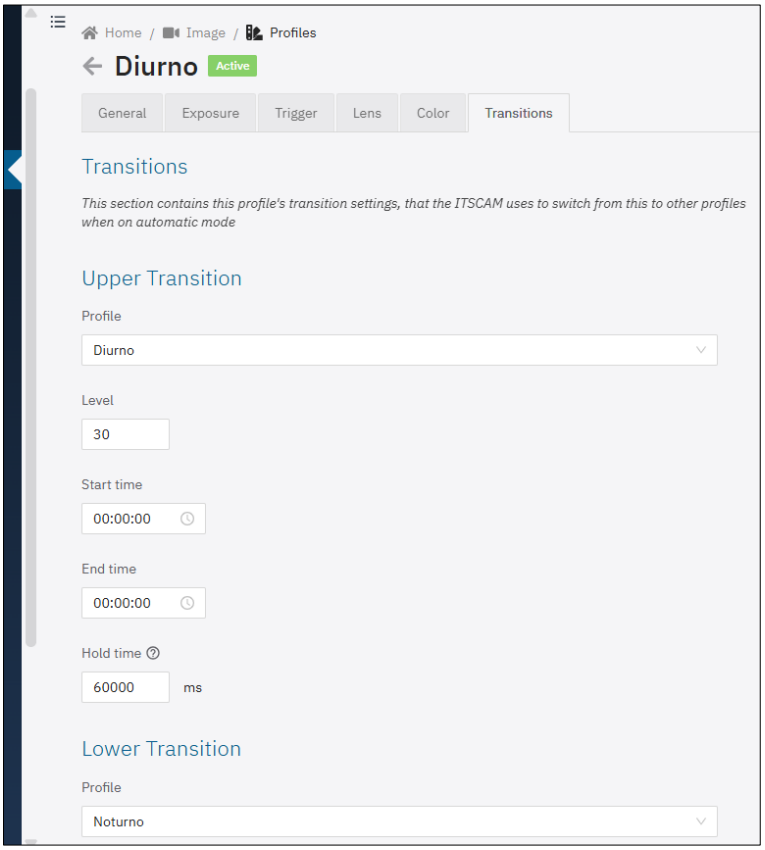
2.1.5. Transitions between Image Profiles

When the *Enable automatic transitions* between Image *Profiles* option is selected, the *Level* values and times applied are used as the basis for the transition.

1) Find the profile to setup in the *Image* > *Profiles* menu and click *Edit*;



2) Select the *Transitions* tab and on this screen, the settings are automatically saved;



3) Set the *Upper Transition* options for the profile being edited:
   a. Select which *Profile* the upper transition occurs;
   b. Set the *Level* value at which the upper transition should occur;
   c. Set a *Start time* and/or an *End time* for the upper transition;
   d. Set the *Timeout* for the upper transition so that the device does not oscillate unnecessarily between profiles by entering a value between 10,000 milliseconds (10s) and 100,000 milliseconds (100s);
4) Set the *Lower Transition* options for the profile being edited:
   a. Select which *Profile* the lower transition occurs;
   b. Set the *Level* value at which the lower transition should occur;
   c. Set a *Start time* and/or an *End time* for the lower transition;
   d. Set the *Hold time* for the lower transition so that the device does not oscillate unnecessarily between profiles by entering a value between 10,000 milliseconds (10s) and 100,000 milliseconds (100s).

2.1.6. Suggested Image Configuration for Profiles

Capturing quality images is fundamental to get the full potential out of capture devices. For this reason, a suggested image configuration is presented in the Visible Light (Daytime Profile) and Infrared Light (Nighttime Profile) situations, which can be used as a starting point for adjusting the equipment.

Images taken at night and/or with the aid of artificial infrared lighting (Nighttime Profile) require image settings that highlight the details of vehicles and license plate characters, and the suggested image setting can be used as a starting point for adjusting the equipment and checking that the lighting device is activated.

Tab	Parameter	Suggested value in Daytime Profile (Visible Light)	Suggested value in Nighttime Profile (Infrared Light)
Exposure	Optimal level	40%	
	Region of Interest	Disabled	
	Automatic Shutter	Enabled	
	Minimum Shutter	80µs	
	Maximum Shutter	500µs	1200µs
	Automatic Gain	Enabled	
	Minimum Gain	0.01dB	
	Maximum Gain	8.50dB	10.50dB
	Automatic iris	Enabled	
Trigger	Trigger*	Enabled/Disabled	
	Multiple Exposures	Enabled/Disabled	
Lens	IR Filter	Enabled	Disabled
Color	HDR**	Disabled	Enabled
	Gamma	130	25
	Gloss	0%	
	Contrast	0%	
	Saturation	0%	-100%
	Black level	20	
	Automatic white balance	Enabled	Disabled
	Red	N/A	100%
	Green		100%
	Blue		100%
Transitions	Enable automatic transitions***	Selected	
	Profile (Upper Transition)	Daytime	
	Level (Upper Transition)	30	10
	Timeout****	60,000ms	1,000ms
	Profile (Lower Transition)	Nighttime	
	Level (Lower Transition)	20	10
	Timeout****	60,000ms	1,000ms

*When enabled, the suggested minimum interval between shots is 100 ms. The captures generated by ITSCAM are determined by the *Event* configured and within the specified *Minimum interval* between triggers.

**Only devices with the S6 sensor have this functionality.

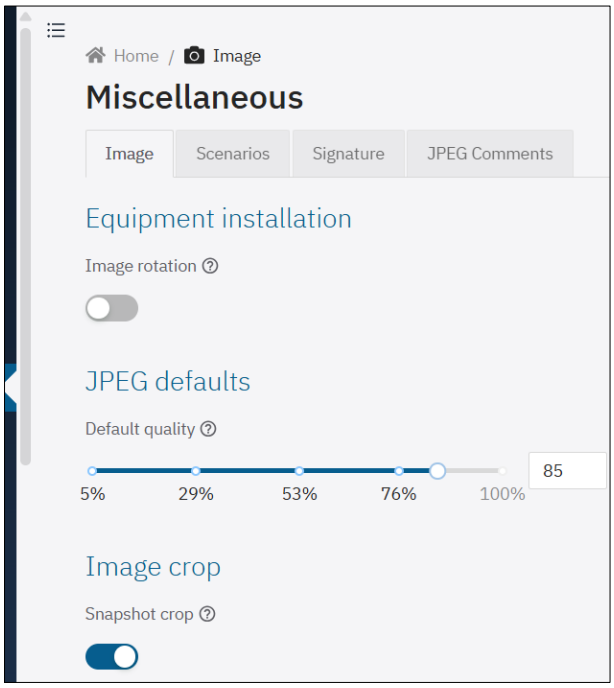
***The settings shown for the automatic transition between profiles must be made in the *Transitions* tab of the corresponding profile.

****It is advisable to set the *Timeout* close to 1 minute (by entering a value of 60,000ms in the interface) so that the device does not oscillate unnecessarily between configuration profiles.

2.2. Image Framing

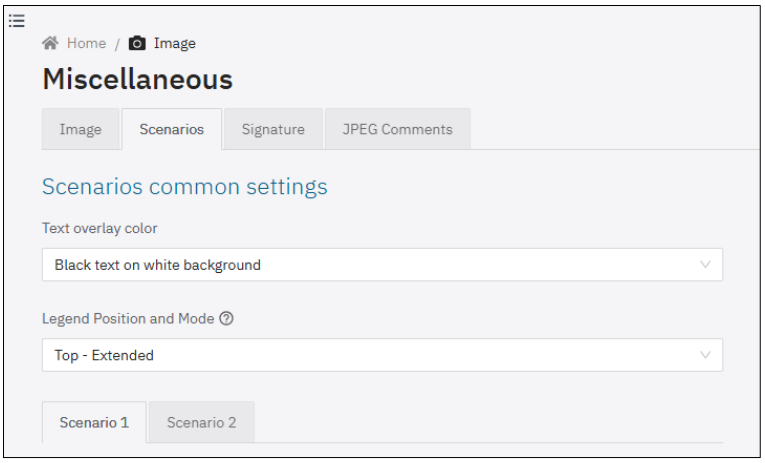
When the device is attached using a suction cup to the windshield of a vehicle or when it is attached to the roof, the image of the device will be upside down in the view, which can be corrected in *Image rotation*:

1) Go to the *Image* > *Miscellaneous* menu and the settings will be saved automatically;
2) Go to the *Image* tab and enable the *Image rotation* option when the equipment is installed upside down;
3) Set the *Default Quality* value at which the JPEG will be generated;



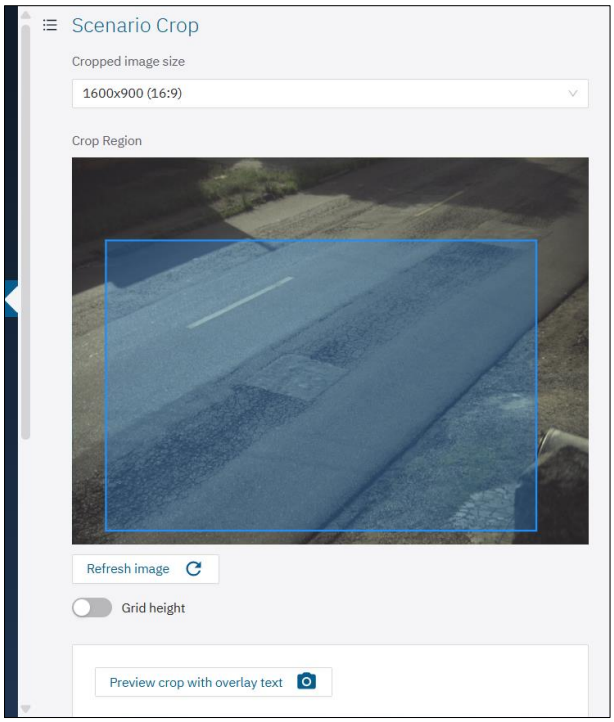
4) Enable the *Photo cropping* option and images will only be generated of the portion of the ROI delimited over the preview image. This option does not affect streams and live image viewing and is intended to improve the conversion time to JPEG;
    a. Select the *Cropping Mode* from the options:
        i. Static (cropping used to imitate a model with lower resolution);
        ii. OCR (cropping uses the ROI as a default position and if a plate is recognized, keeps the vehicle centered, adjusting the position and maintaining the same size);
    b. Display the license plate size reference by clicking on *Grid height* and specifying the value in pixels, following the template shown in OCR;
    c. Check that the *Cropped image size* is the final desired size;

5) Click on the *Scenarios* tab to configure two-lane surveillance, defining two different scenarios in the image generated by the capture device, simulating the operation of two different capture devices and also define the settings common to both scenarios;
    a. Select *Caption Color Disabled*, *Black Text on White Background*, *and White Text on Black Background;*
    b. Select the *Caption Position and Mode from the Top – Overlay, Bottom – Overlay, Top – Extended* and *Bottom – Extended* options to define whether the caption will be inserted at the top (top) or bottom (bottom) of the image, overlapping part of it (overlay), or will extend the image so that parts of the image are not lost (extended). The *Extended* option consumes more processing resources.



    c. In the *Scenario 1* tab, enter a text at the top of the image, up to 1024 characters, for *Scenario 1*, by clicking on the *Caption text in photos* field. Go to the *Detailed help* field to check the values that can be entered as captions;
    d. Select the *Text Size (in pixels*) that will be used in the caption from the available size options;

e. Select the *Size of the cropped image* for *Scenario 1* from the available size options;
    i. Click on the vertex to drag the cropped image to the ideal frame;
f. Display the license plate size reference by clicking on *Grid height* and specifying the value in pixels, following the template shown in OCR;
g. Check that the caption and the size of the crop are properly configured by clicking *Preview crop with caption*;



h. Repeat the steps above, accessing the *Scenario 2 tab* and defining the specifications for *Scenario 2*.

6) It is possible to obtain images of the *Scenarios* through the *COUGAR Communication Protocol (Socket)* or the Snapshot function, available in *Checking the Images Generated*.

2.2.1. Digital Signature

Description of the *Digital Signature* Validation Process:

When the ITSCAM 600 converts the captured image into JPEG format, it calculates the hash of the image using the SHA-256 algorithm. This "hash" is then signed by the RSA algorithm with a key of 1024, 2048 or 4096 bits. In the final image, which will be transmitted over the network, the following fields are appended to the image:

- *Sign*: RSA hash signature;
- *ExpoenteRSA*: exponent used to calculate the signature;
- *ModuloRSA*: module used to check the signature;
- *Sha256*: SHA-256 hash of the image.

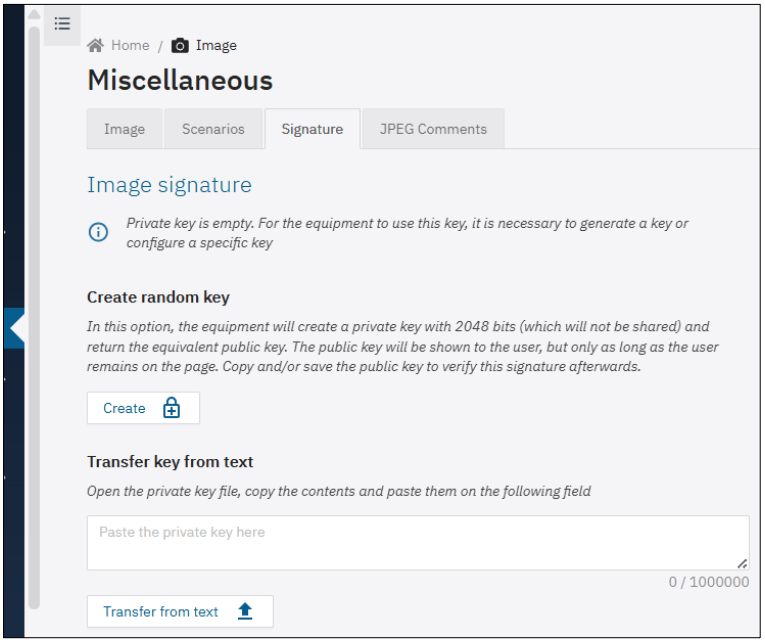
To ensure high security, the exponent used is always 65537. This data is appended, in text format, to the comment field of the JPEG image, defined in the "FF FE" marker according to the ISO/IEC 10918 specification.

To check the signature, carry out the following procedure:

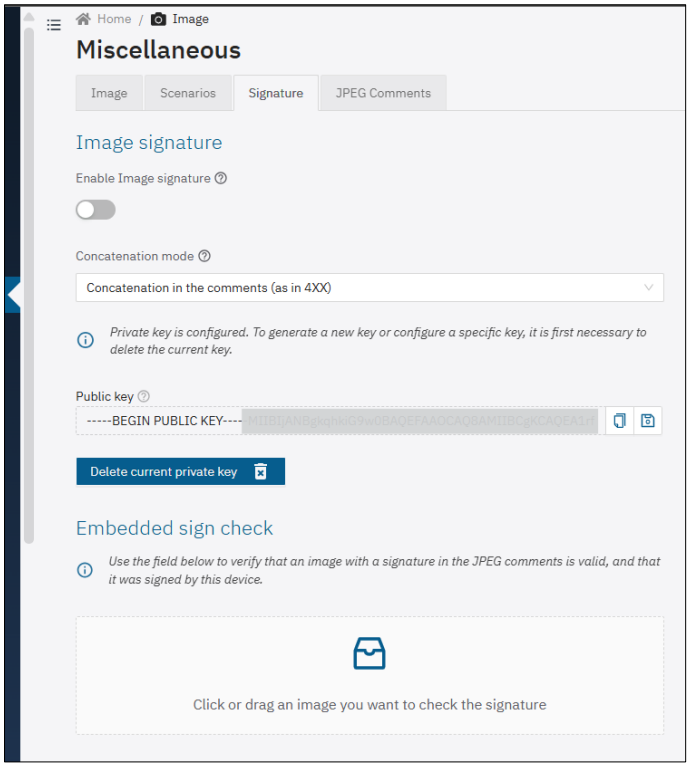
a) Remove the texts referring to *Sign*, *ExpoenteRSA*, *ModuloRSA* and hash (Sha256) from the comments field;
b) Calculate the hash of the image with the text specified in "a" removed;
c) Extract the hash from the signature (*Sign* field) considering the exponent (*ExpoenteRSA*) and the modulus (*ModuloRSA*);
d) Compare the hash obtained in "b" with the hash obtained in "c". If you get the same hash in both cases, the image is authentic. Otherwise, the image has been tampered with.

On the interface of the device with this functionality applied, follow the steps to generate a *Digital Signature* or for verification:

1) Go to the Image > Miscellaneous menu;
2) Click on the *Signature* tab and generate or register a private key, which allows you to digitally sign the images. After registration, the public key remains available to validate the signature of the images generated by the device;
3) Choose between the options for sending a private key:
    a. Create a new private key by clicking *Create+* to *Create a random key*. In this option, the device will create a 2048-bit private key (which will not be shared) and return the equivalent public key;
    b. Enter the text of the copied private key in the field available under *Transfer key from text*, sending it by clicking *Transfer text*;
    c. Upload the saved private key file by clicking on the *Transfer key* file field, which supports files in PEM or P12 format (from 1024 to 4096 bits);

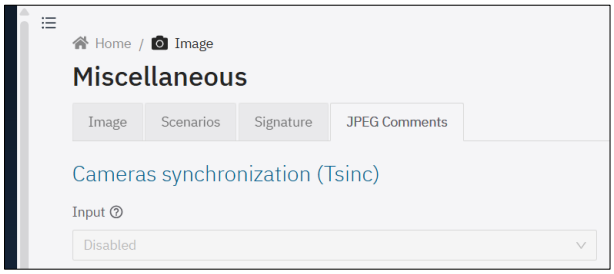


4) Select to *Enable digital signature* on images and the *Concatenation mode*, to join the signature to the generated JPEG file;
5) Click or drag a file into the *Embedded Signature Check* to check whether an image with a signature in JPEG comments is valid and has been signed by the device;



2.2.2. Jpeg comments

1) Enable the option that uses the physical input of the device to count the synchronization time of captures made by multiple devices. To select a synchronization option, some input port must be configured on the ITSCAM 450, ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+ and VTR 600. With this setting active, the "Tsinc" and "TsincDT" parameters are added to the image's jpeg comments. This information can be used by a server when applying synchronization.

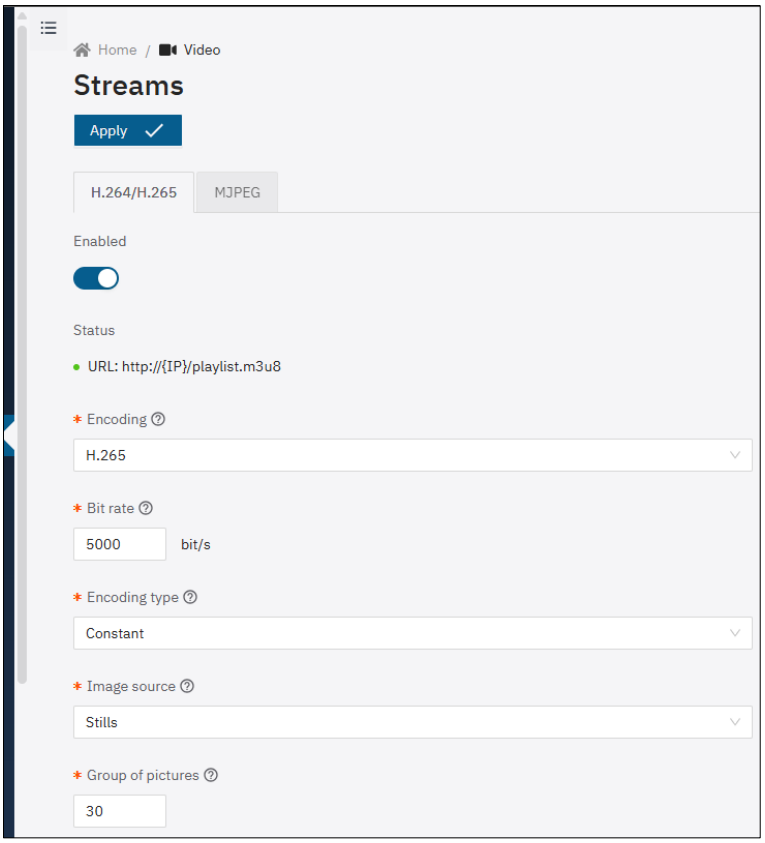


2.3. Video Settings (Streams)

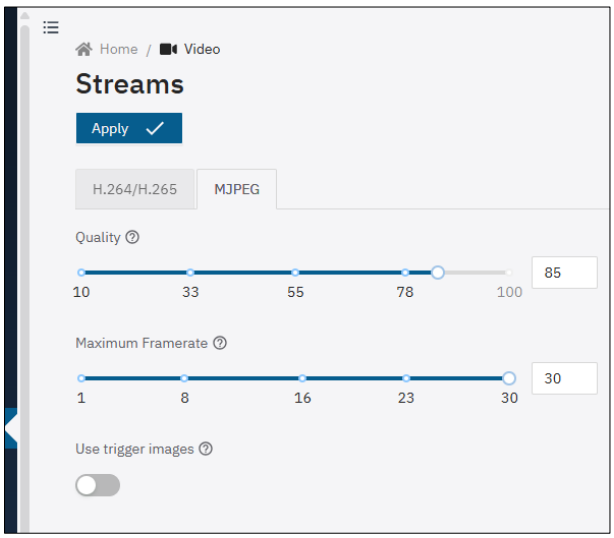
The devices offer the option to follow live images captured by the device outside the web interface, integrated into a DVR or video monitoring system by setting it in the *Video > Streams* menu. ITSCAM 450 and ITSCAM 450+ devices only allow MJPEG streaming.

1) Access the settings in the *H.264/H.265* tab by clicking on *Enabled*;
2) Choose between the H.264 or H.265 *encoding* options;
3) Set the Data transfer *Bit rate* for the stream in bit/s;

4) Select the *Encoding type* from the options:
    a. *Constant*: the value remains the same throughout the duration of the video;
    b. *Variable*: the value decreases in parts that need less information and increases in sections that need a greater amount of data.
5) Select the *Profile* that defines the compression standard (only available for H.264 coding) from the options: *High*, *Main* or *Baseline*;
6) Select the *Image Source* for the flow from the options:
    a. *Preview*: using continuous image;
    b. *Stills*: using image captures (or high trigger rate);
7) Enter a number of frames from an *Group of pictures* block. This block is used as a reference for the stream encoding process;



8) Access the stream configuration options in the *MJPEG* tab;
9) Adjust the quality of the images sent by the stream in the *Quality* field;
10) Set the *Maximum framerate* of images sent by the stream;
11) Enable the option to *Use trigger images* to display in the stream. With this function enabled, it is possible to have video during the night synchronized with the triggering of the illuminators. If disabled, the preview images will be used;
12) Click *Apply* after confirming the settings entered:



3. Equipment Settings

3.1. General

The *Equipment* > *General* screen allows you to configure the equipment's display information:

1) Enter a device ID under *Device name*;
2) Select the interface display language, available in *Portuguese* (BR) and *English* (US).

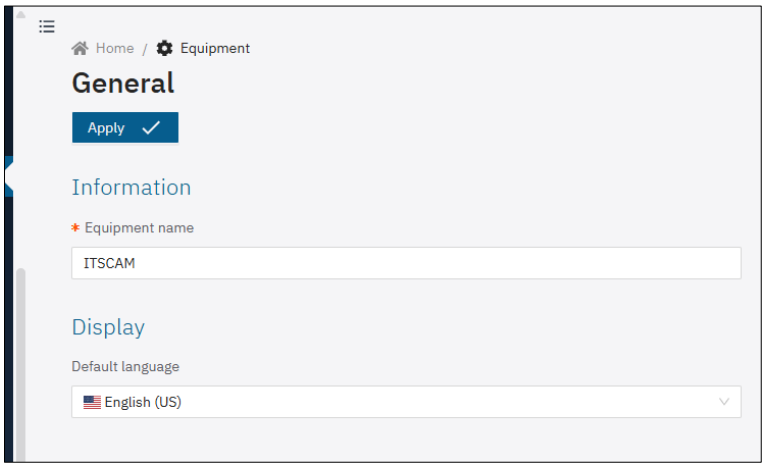


Figure 14- Equipment > General menu screen

3.2. Date and Time

The *Date* and *Time* settings can be accessed from the *Equipment* > *Date and Time* menu and can be made manually or via an NTP server by enabling the *Use NTP server* option and entering the *NTP server address*.

In the *Time zone* option, the continent and the time zone of the respective city must be selected when clicking, considering that Daylight Saving Time is automatically applied according to the current rules. To deactivate automatic Daylight Saving Time, select the Etc option and the desired fixed *GMT* (with opposite sign, due to historical patterns).

It is possible to use the device itself as an NTP server by selecting the *Enable* internal NTP server option:

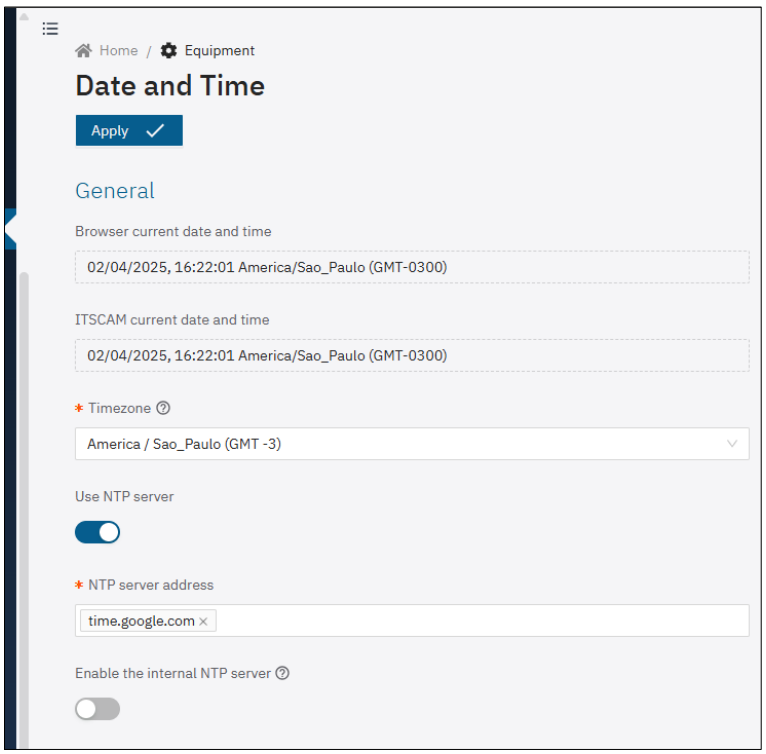


Figure 15 – Equipment > Date and Time menu screen

3.3. Network Settings

Network settings require the use of an *Auxiliary Configuration Equipment*.

1) Enter the factory default IP address in the browser;
2) Enter *user* admin and *password* 1234;
3) Access the menu *Equipment > Network*;
4) Access the tab for the network to be configured.

The initial network settings screen allows the visualization of the connected networks *Status* and the diagram of the configured *Routes*:

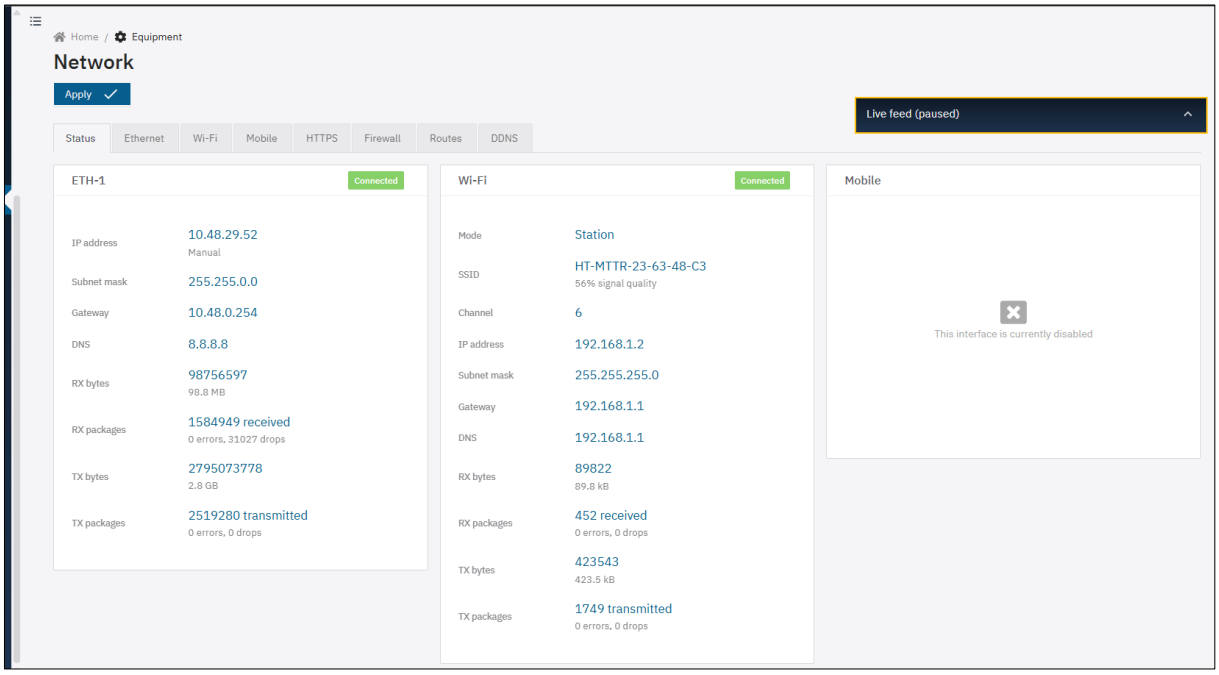


Figure 16 - Network settings home screen

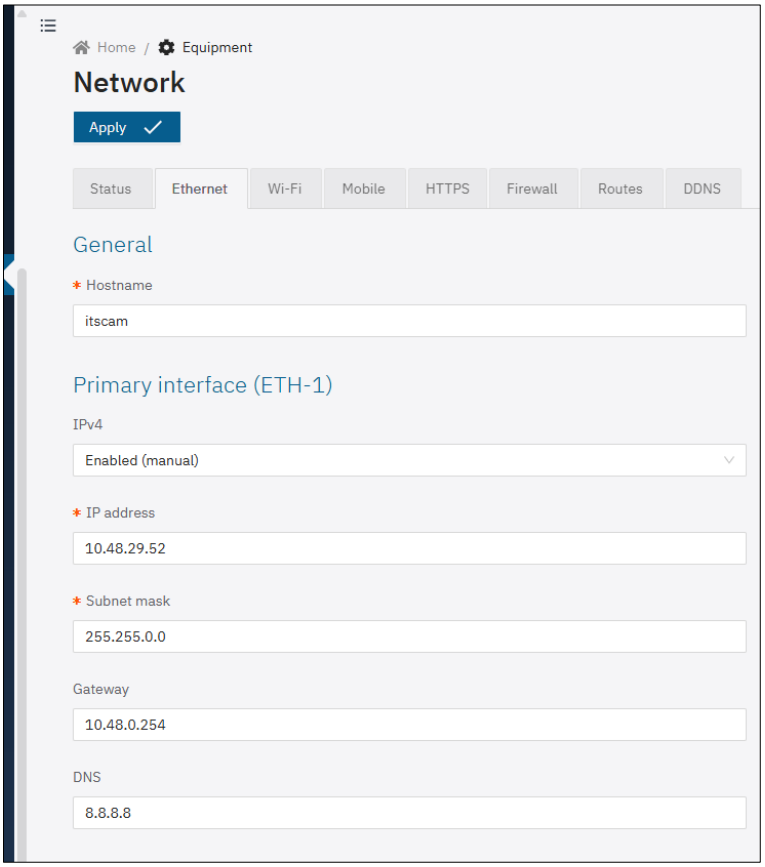
3.3.1. Ethernet Network Settings

Some models of the ITSCAM 600 device have only one Ethernet connector, and the ITSCAM 600 FHD and ITSCAM 600+ devices have 2 Ethernet connectors. For the ITSCAM 450, ITSCAM 450+ and VTR 600 there is only 1 Ethernet interface for settings:

1) Select the *Ethernet* tab;
2) Identify the network in *Hostname*;
3) Fill in the data for the *Primary Interface (ETH-1)* when connecting to *ETH-1*;
    a. Use a different IP address to the one used to access the device to avoid conflicts and malfunctioning of the data network;

The maintenance IP (192.168.254.254) is used to recover the connection in extraordinary situations when the primary IP is lost. For this reason, when manually configuring the device's network interface (Ethernet or Wi-Fi), values different from the maintenance IP must be applied.

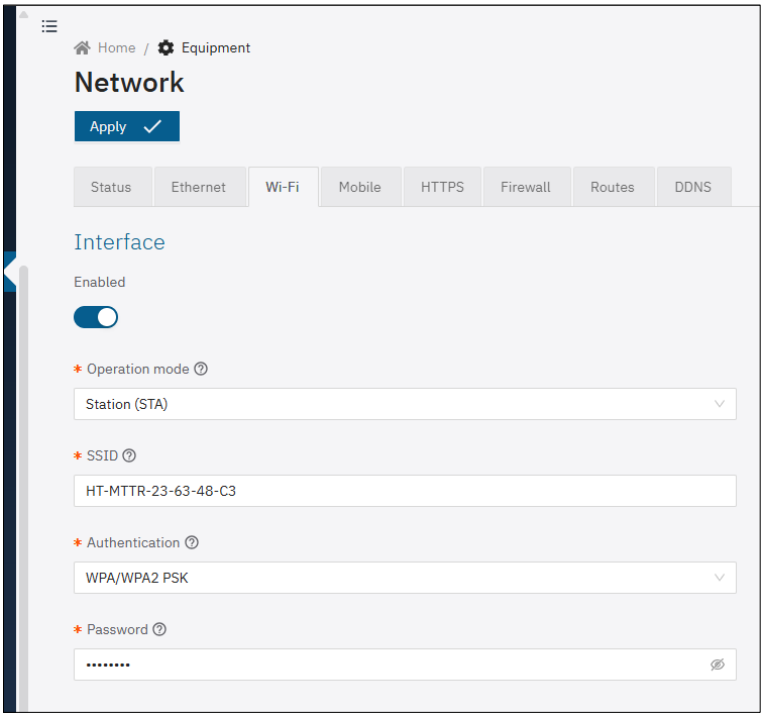
4) Click *Apply* after confirming the data entered:



3.3.2. Wi-Fi Network Settings

Only the VTR 600, ITSCAM 600+ and some models of the ITSCAM 600 and ITSCAM 600 FHD devices can be connected to a Wi-Fi network and configured:

1) In the Wi-Fi tab, select *Enabled* for the interface;
2) Under *Operating Mode* select the option *Station (STA)* to connect to an existing Wi-Fi network;
3) Click on the *SSID* field and select the Wi-Fi network that will be used;
4) Under *Authentication*, select the Authentication protocol to be used: *Open (no authentication)*, *WEP* or *WPA/WPA2 PSK*;
5) Enter the *Password* to access the selected Wi-Fi network;
6) Click *Apply* after confirming the data entered:

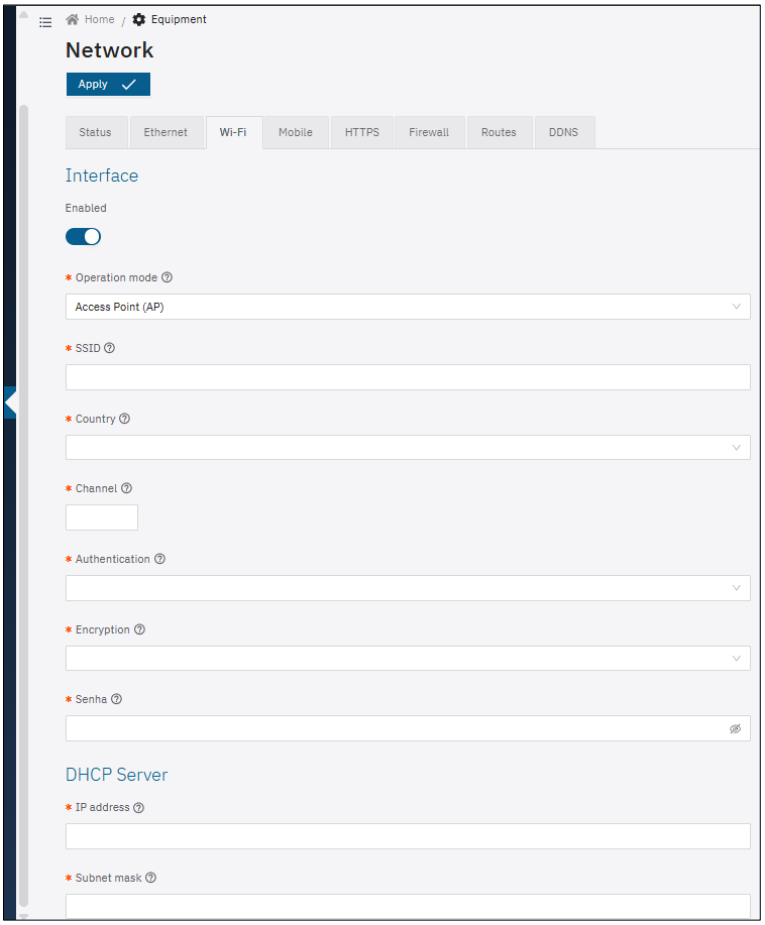


Select the *Access Point (AP)* option in the *Operation Mode* field only when the device is to be used as an access point to an available Wi-Fi network:

1) Enter the *SSID* identification data, the *Country*, the *Channel* and the type of *Authentication* that will be applied when distributing the Wi-Fi connection;
2) Select the type of *Encryption* on the connection between TKIP, AES or TKIP & AES;
3) Set a *Password* to access the *Access Point* network;
4) Enter the *IP Address* and *Subnet Mask* of the *DHCP Server* to determine the address range for the devices connecting to the *Access point*;

The maintenance IP (192.168.254.254) is used to recover the connection in extraordinary situations when the primary IP is lost. For this reason, when manually configuring the device's network interface (Ethernet or Wi-Fi), values different from the maintenance IP must be applied.

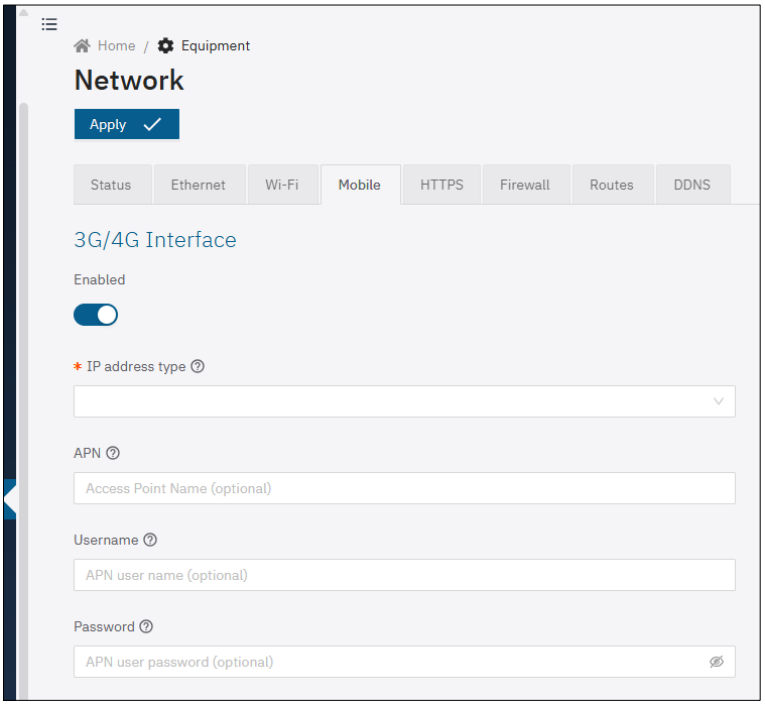
5) Click *Apply* after confirming the entered data:



3.3.3. 3G/4G Network settings

Only the VTR 600, ITSCAM 600+ some models of the ITSCAM 600 and ITSCAM 600 FHD devices can be enabled for the mobile network, on the 3G/4G interface, and configured:

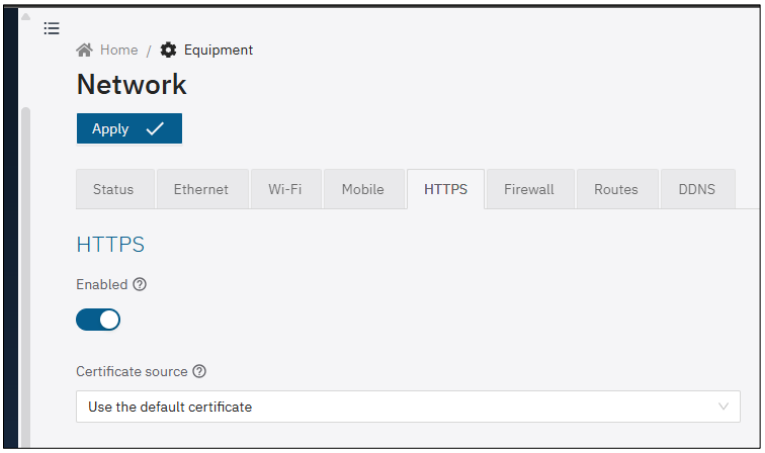
1) In the Mobile tab, select the option *Enabled* for 3G/4G interface;
2) Fill in the APN data if there is no automatic detection, entering custom data whenever you need to configure the operator's information. By default, the information is:
    a. *APN*: http://[provider's name].com.br;
    b. *User*: [provider's name];
    c. *Password*: [provider's name];
3) Click the *Apply* after confirming the data entered.



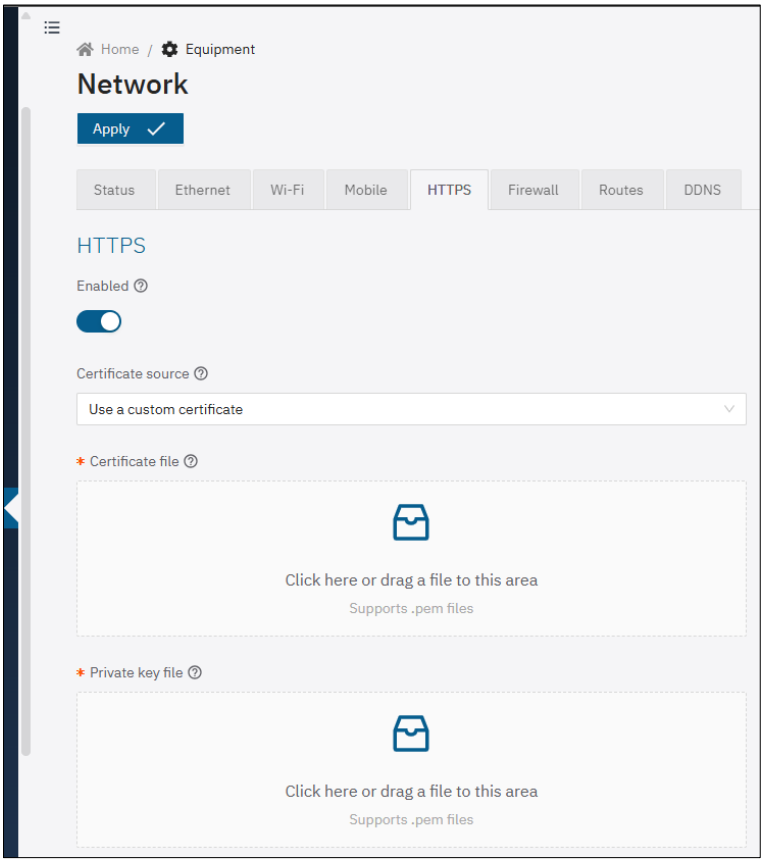
3.3.4. HTTPS settings

ITSCAM 600, ITSCAM 600 FHD, VTR 600 and ITSCAM 600+ capture devices support the HTTPS protocol (port 443), which allows communication to be carried out in encrypted form, with a certificate created by the device itself or supplied by the user. ITSCAM 450 and ITSCAM 450+ devices up to firmware version 1.7.0 do not support the HTTPS protocol.

1) In the HTTPS tab, select *Enabled*;



2) Specify the provider of the key files by selecting *Certificate source*:
   a. *Use the standard certificate:* The system will use an embedded certificate generated by Pumatronix and a security message will appear in the browser on first access. For the settings to be applied, the device must be *Restarted* by going to *System > Update*;
   b. *Use a specific certificate:* The system will use a certificate provided by the user, via a pair of public & private key files, by dragging in the *Certificate file* and *Private key file* areas;
3) Click *Apply* after confirming the data entered:

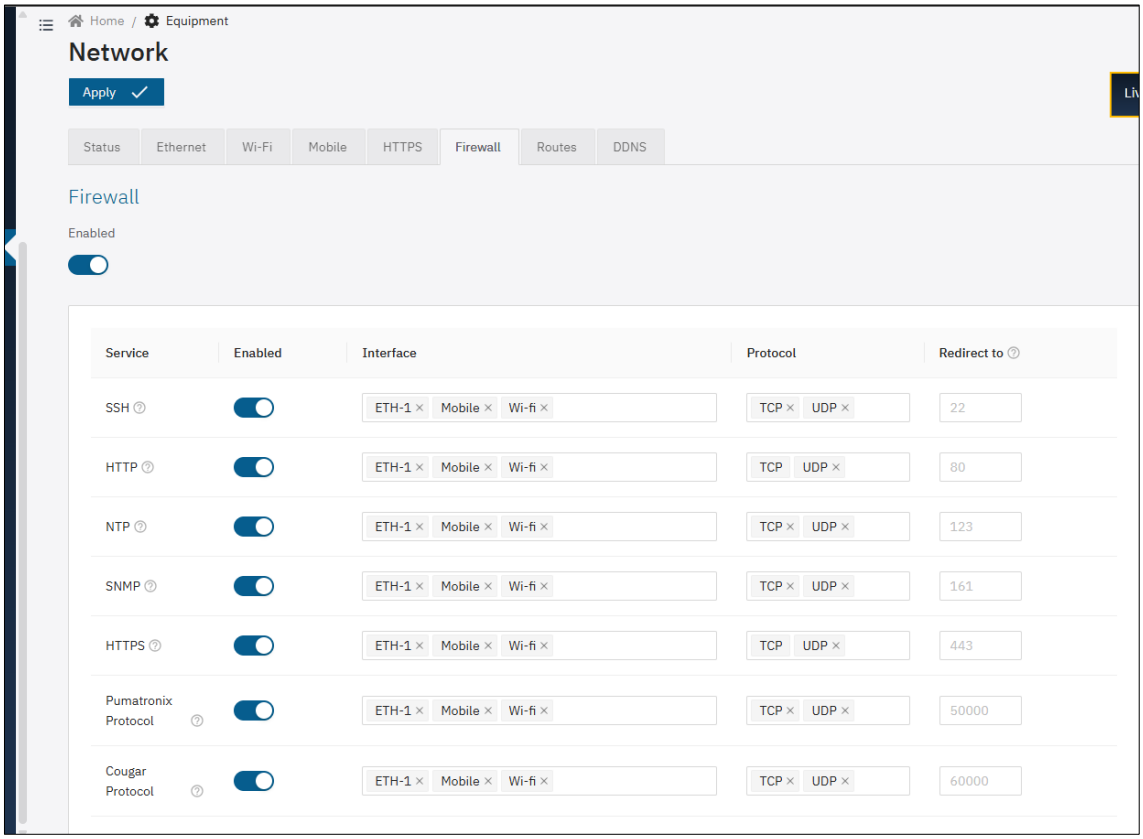


The use of HTTPS changes the access link to the device from `http://` to `https://` and the device must be *Restarted* for the change to be applied, by going to *System > Update*.

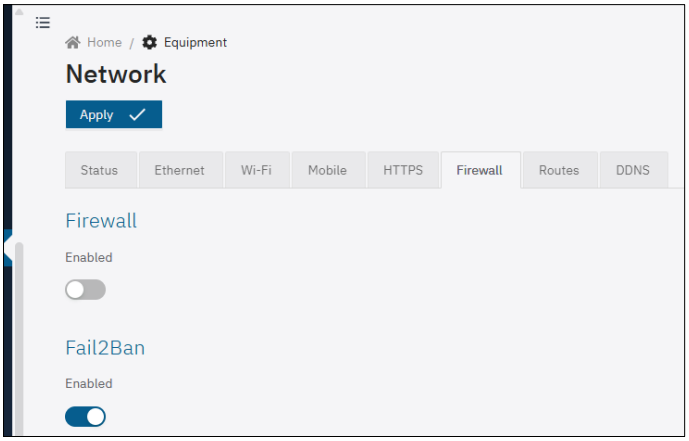
3.3.5. Firewall settings

Configure the network security firewall by going to the *Equipment* > *Network* menu:

1) In the *Firewall* tab, select *Enabled*;
    a. Select the connection *Interface* used with the port;
    b. Select the *Protocol* to be applied;
    c. In the *Redirect to* field, enter a value between 100 and 65535 for which port the service should use instead of the default (optional), or leave it blank to not redirect (the value should be different from other ports used by the system or plugin and other redirects);



2) Click on the switch that enables the *Fail2Ban* option, which offers additional protection against repeated authentication/login failures to the device by blocking the user's IP for 10 minutes after 5 failed attempts;

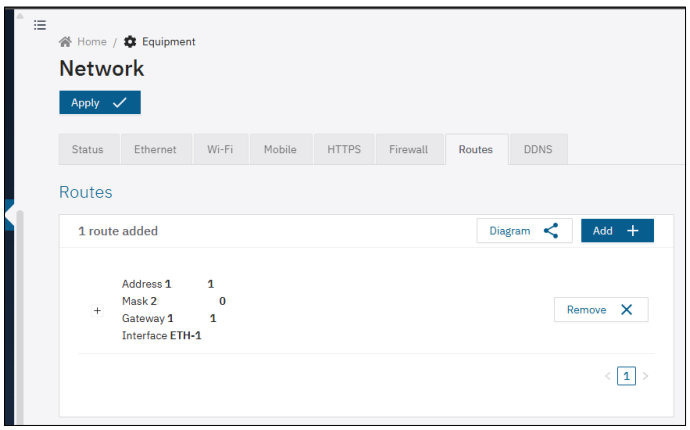


3) Click the *Apply* after confirming the data entered.

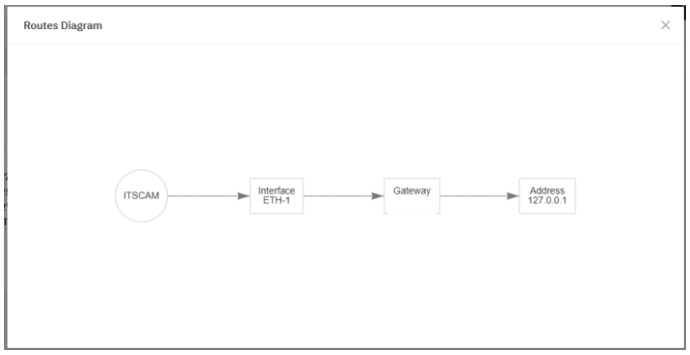
3.3.6. Route settings

Configure *Routes* when it is necessary to access remote IP subnets or those not directly connected to the network interface, which can be accessed using the Default Gateway or the specified route shown in the *Diagram*.

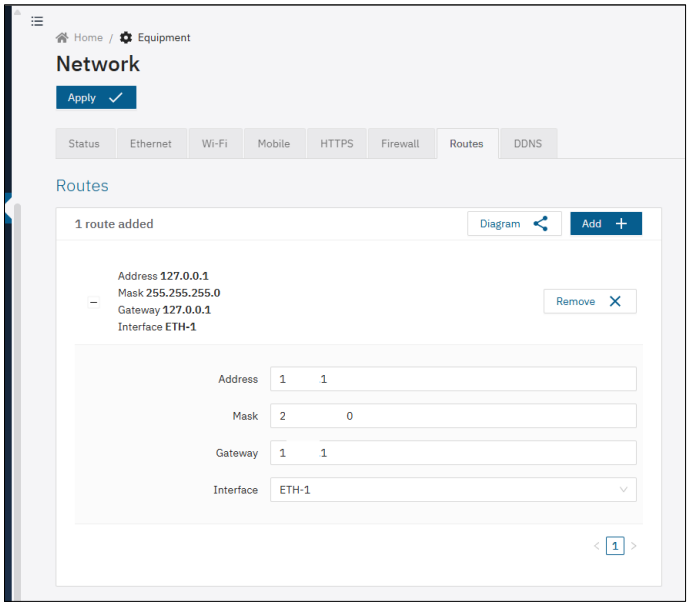
1) Select the *Route* tab;
2) Add a route by clicking *Add+*;



3) View by clicking on *Diagram*;



4) Edit the *Address, Mask, Gateway and Interface* data by clicking on the "+" to the left of the route;
5) Click *Apply* after checking the data entered.



3.3.7. Integration with DDNS service

By upgrading to firmware version 1.4.0 and above, it is possible to integrate the network connection with a DDNS (dynamic DNS) service provider:

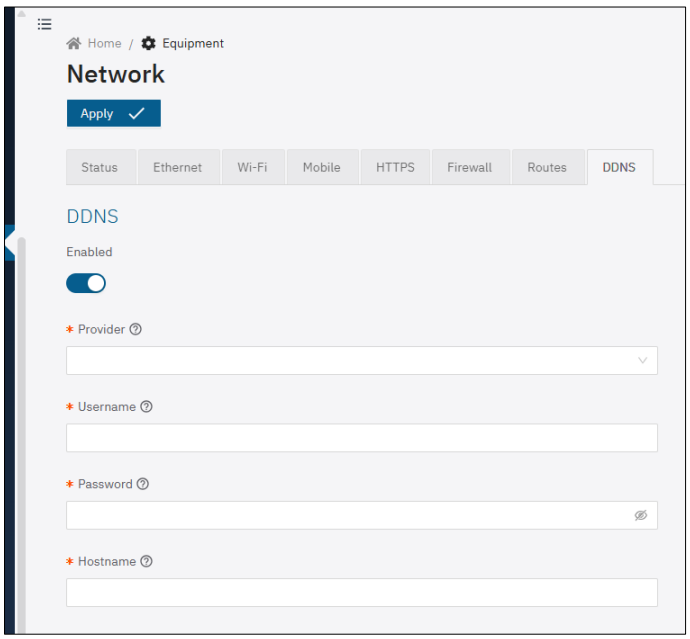


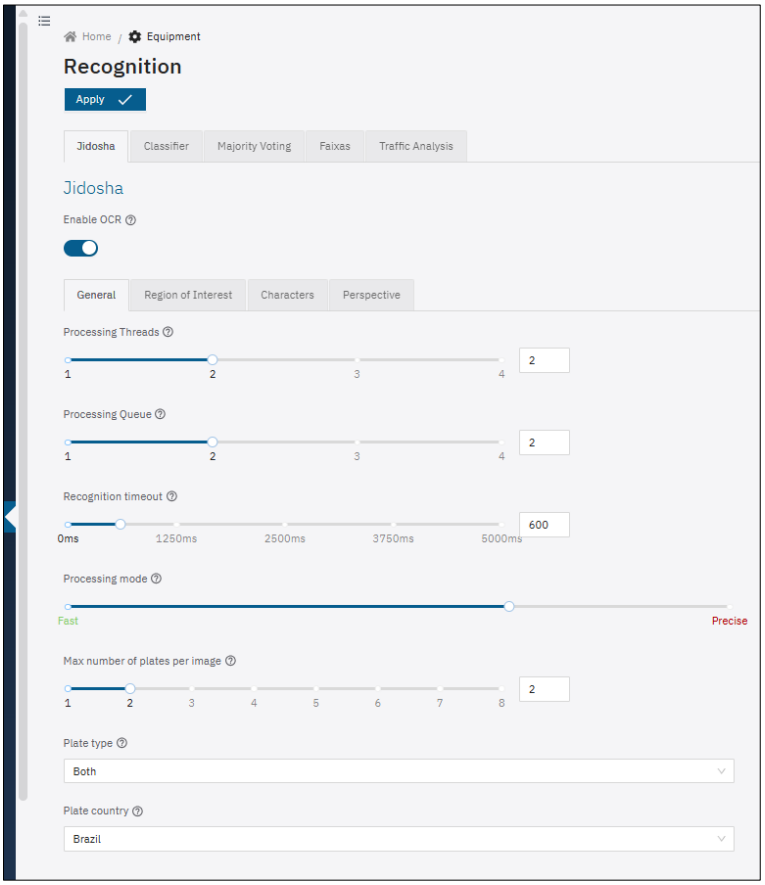
Figure 17 - Equipment > Network menu screen when configuring the DDNS

1) In the *DDNS* tab, select the *Enabled* option;
2) Fill in the fields with the details of the DNS Service *Provider*, the *User* and *Password* for accessing the provider and the *Hostname*, including the domain name;
3) Click *Apply* after checking the data entered.

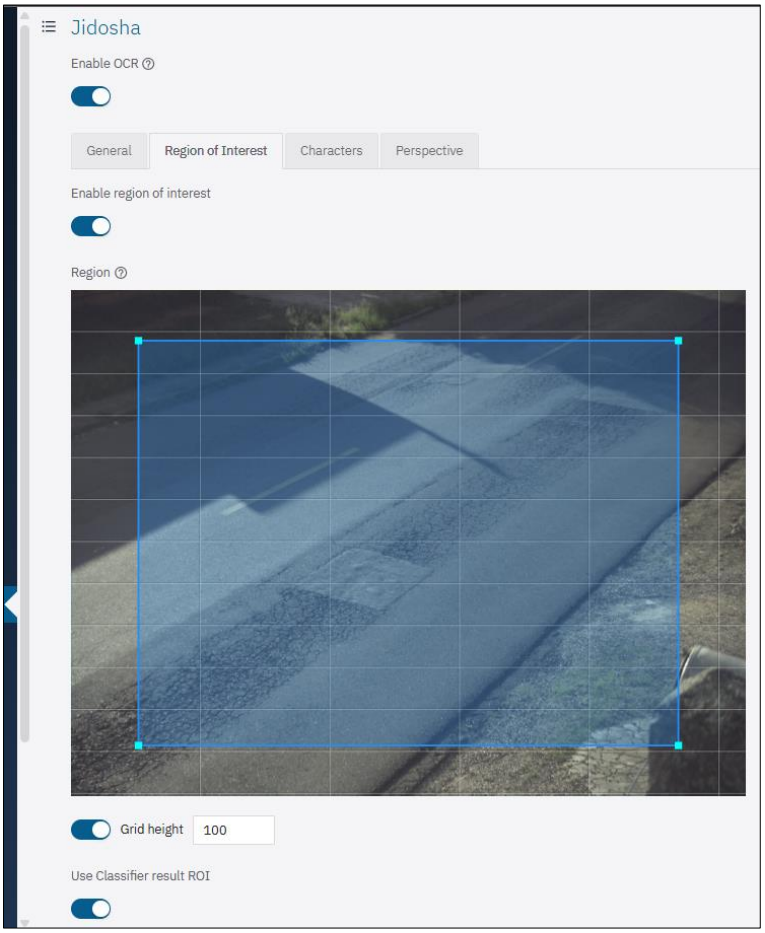
3.4. Recognition (OCR Reading)

The captured images can provide automatic recognition of the license plate of the vehicles in the images (OCR) and the type of vehicle identified (*Classifier*). In this way, JPEG images are delivered with the comments field of the file containing the license plates identified, the type of vehicle identified (motorcycle, car, truck and bus) and the instantaneous settings of the equipment.

1) Go to the *Equipment > Recognition* menu;
2) Select the *Jidosha* tab and click *Enable OCR*;
3) Access the setting options in the *General* tab:
    a. Configure the number of *Processing threads* that will be used to process the OCR;
    b. Set the maximum size of the OCR *Processing Queue* (smaller values reduce latency while larger values reduce frame loss in high flow situations);
    c. Set the *Recognition Timeout* (images without a license plate or with hidden characters cause the algorithm to continue searching for the license plate until the specified time is reached);
    d. Set the OCR *Processing Mode* to faster or more precise;
    e. Set the *Maximum number of plates per image*, determining the maximum number of plates that can be read in the same image;
    f. Select the *Plate type* of the vehicles that should be searched for by the algorithm in the images, considering the pattern of license plate letters and numbers, which are different for vehicles and motorcycles (in frontal installations, motorcycle license plates are not captured);
    g. Select the *Plate country* of the vehicles in the images. For OCR processing of license plates from countries other than Brazil, please contact Pumatronix technical support. ITSCAM with embedded OCR performs the processing for Brazilian license plates only.



4) Select the *Region of Interest* tab and click *Enable region of interest*:
   a. Enable the *Grid Height* option, located just below the image;



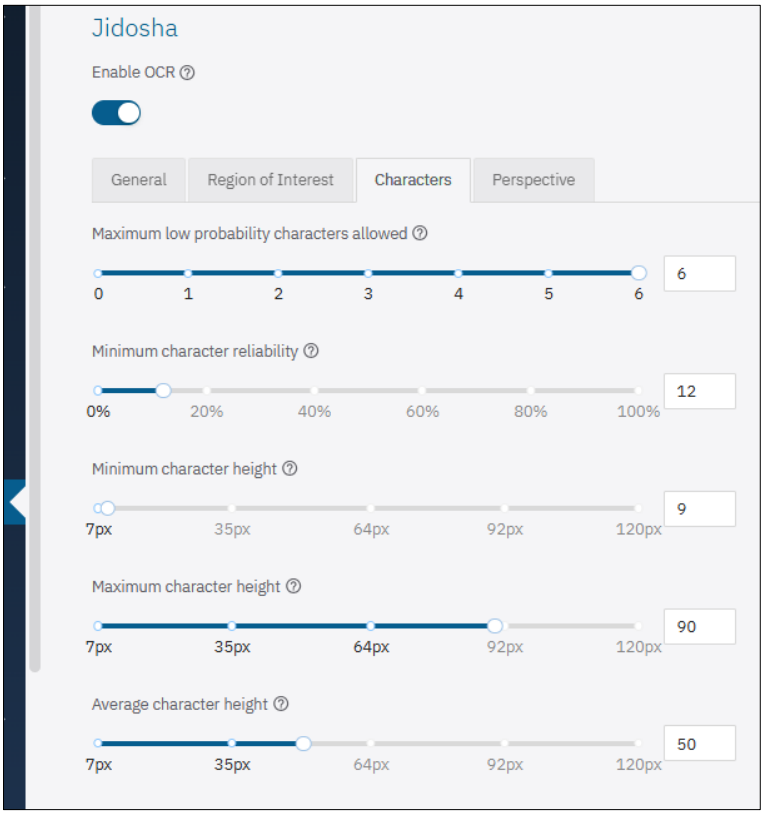
b. Set 10 for the *Grid height* (in pixels)
    i. Note the size of the license plate characters in the region of the image marked as 1 in the example, i.e. when the vehicles are in the most distant region to the capture device. The characters should be approximately the grid's height. If they are very different, it is necessary to review the zoom settings;

c. Set 40 for the *Grid height* (in pixels);
    i. Note the size of the license plate characters in the region of the image marked as 3 in the example, i.e. when the vehicles are in the region closest to the capture device. The characters should be approximately the grid's height. If they are very different, it is necessary to review the zoom settings;
d. Set 25 for the *Grid height* (in pixels);

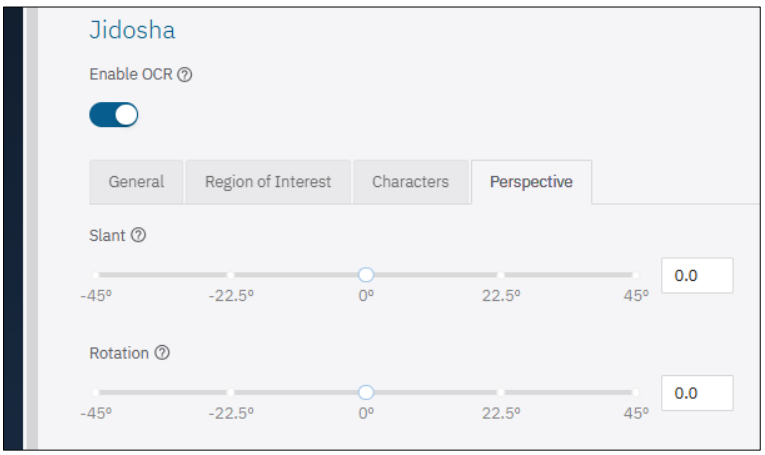
i. Note the size of the license plate characters in the region of the image marked as 2 in the example, i.e. when the vehicles are in the central region of the image. The characters should be approximately the grid's height. If they are very different, it is necessary to review the zoom settings;

OCR in Region of Interest: OCR is performed only within the drawn region, which is slightly shaded in the image.

5) Access the setting options in the *Characters* tab:
   a. Set the *Maximum low probability characters allowed* and characters that are identified with reliability lower than the minimum value set are represented by the '?' character;
   b. Define the *Minimum Character Reliability* with the minimum OCR reliability percentage, which considers the degree of similarity between the letter identified in the processing and a letter in perfect capture conditions. It is recommended to maintain the factory standard of 85% minimum reliability;
   c. Set the *Minimum character height* as 9;
   d. Set the *Maximum character height* as 60;
   e. Set the *Average character height* as 20;



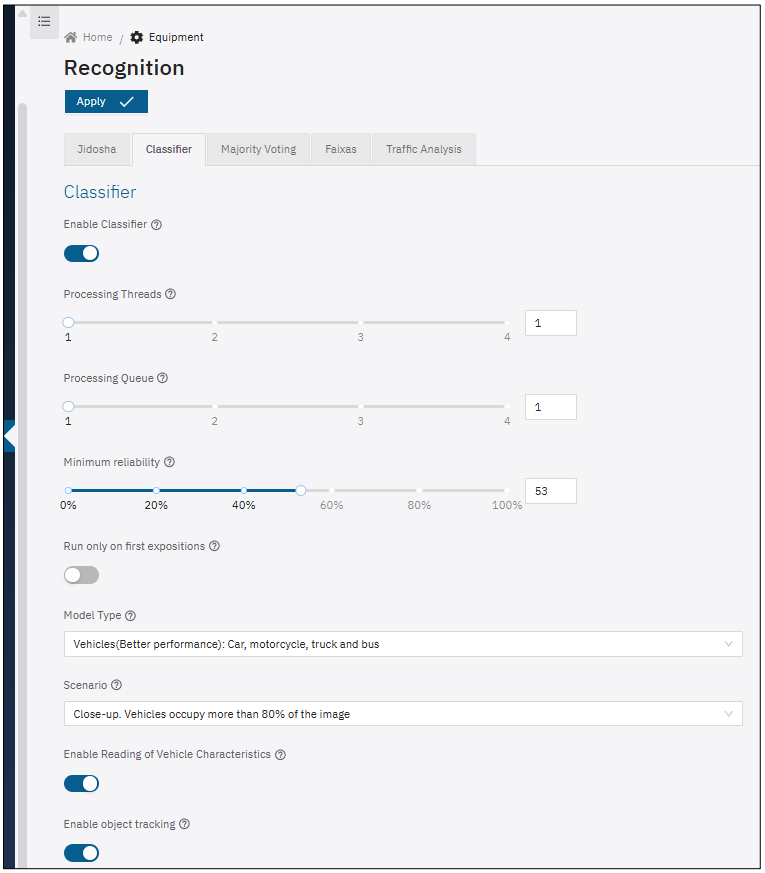
6) Access the setting options in the Perspective tab:
   a. Adjust the *Slant* angle of the plates as they appear in the image, removing the 'italics' effect, in order to improve OCR efficiency:
      i. Take a photo of a plate in the desired position, using the functionality in *Equipment>Images>Snapshot*, and adjust the tilt value until the characters on the plate are aligned with the vertical axis of the grid;
   b. Adjust the *Rotation* angle of the plates as shown in the image to improve OCR efficiency:
      i. Take a photo of a plate in the desired position, using the functionality in *Equipment>Images>Snapshot*, and adjust the rotation value until the characters on the plate are aligned with the horizontal axis of the grid;



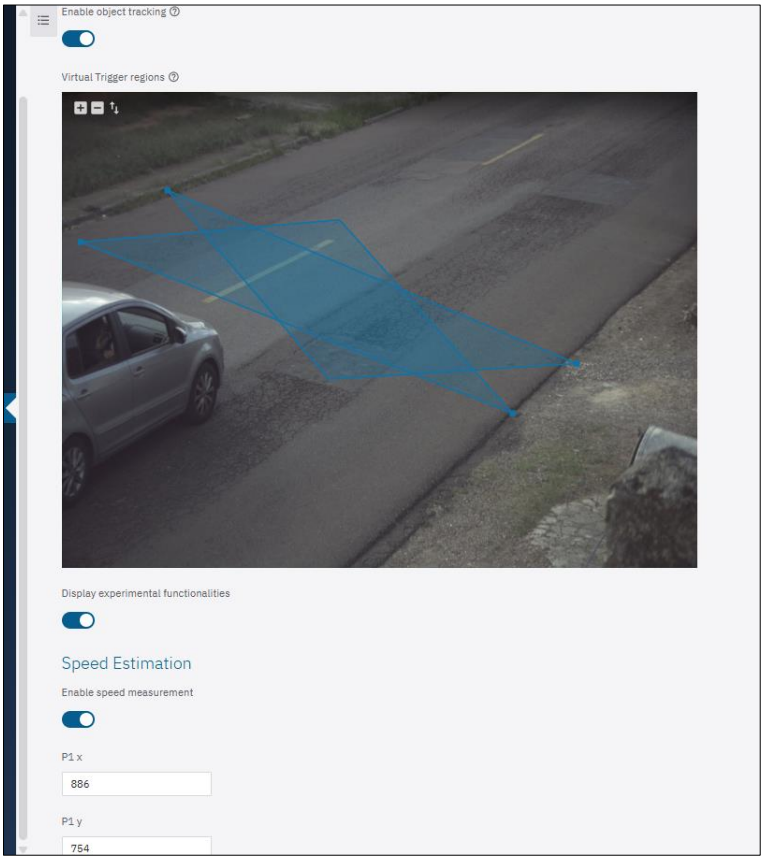
7) Click *Apply* to validate the entered information.

3.4.1. Classifier

1) Go to the *Equipment > Recognition* menu;
2) Select the Classifier tab and click *Enable Classifier* for the device to analyze the captured images in real time and evaluate the content present in the images. This analysis aims to distinguish motorcycles, cars, trucks and buses from images that show only the road;
    a. Set the number of *Processing threads* used to process the Classifier;
    b. Configure the size of the Classifier's image *Processing queue*, considering that smaller values reduce latency while larger values reduce frame loss in situations of high vehicle flow;
    c. Set the *Minimum Reliability* in the identification/classification of vehicles in images, considering the degree of similarity between the vehicle identified in the processing and a vehicle in perfect capture conditions. It is recommended to maintain the factory standard of 20% minimum reliability;
    d. Enable the option to *Run only on first exposures* and the *Classifier* recognition will be processed only on the first capture, when the *Multiple Exposures* option is enabled, and the first is sufficiently illuminated to distinguish the vehicle and the other exposures are adequately illuminated for plate recognition;
3) Specify the *Model Type*, selecting according to the installation of the equipment;
4) Select the *Scenario* considering the installation of the equipment:
    a. *Panoramic* for capturing up to two track lanes;
    b. *Close-up* for a single lane;
5) Select *Enable Reading of Vehicle Characteristics* to have *Classifier* process the reading of vehicle characteristics such as brand, model and color. This option can increase processing time;

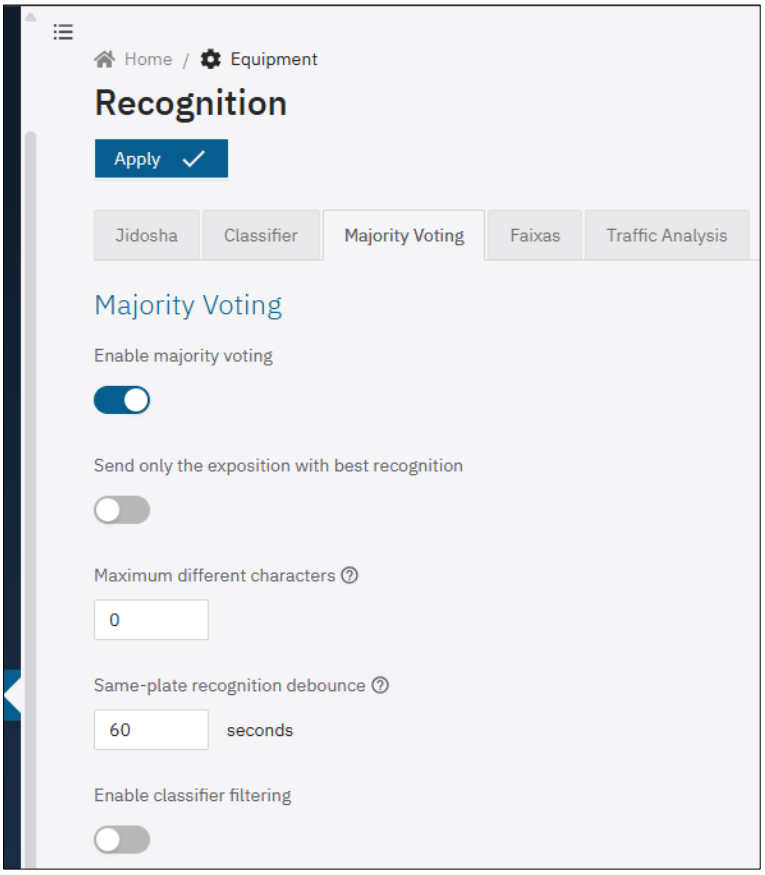


6) Select the *Enable object tracking* option to generate a high-precision *virtual loop* (virtual trigger):
7) Create the *Virtual Trigger* Regions which are made up of 3 pieces of information: 2 points (indicating a line) and a direction (represented by the third point of the triangle), indicating that vehicles crossing the line defined by the two points and following the direction indicated by the triangle generate the capture of an image, characterizing the *Virtual Loop* (virtual trigger):
    a. **+** (Add region): Add a new region. This configuration is limited to a maximum of 4 regions;
    b. **-** (Remove region): Remove selected region. This action can also be performed using the 'delete' key on the keyboard;
    c. ⇅ (Rotate region): Changes the direction of the selected region. This function allows the creation of separate loop points for each lane and indicates the direction of vehicle flow in order to generate a precise virtual loop, based on the class of the objects;
    d. Click on Display experimental features to display the Speed estimator settings:
        i. Select to *Enable* speed measurement and the coordinates P1 to P6 are made available for adjustment. This functionality is based on the timing between frames and the vehicle's tracking coordinates, and the capture viewer displays the vehicle's path and the estimated speed along the way when object tracking is used. This feature is in the experimental phase and only available in selected situations.



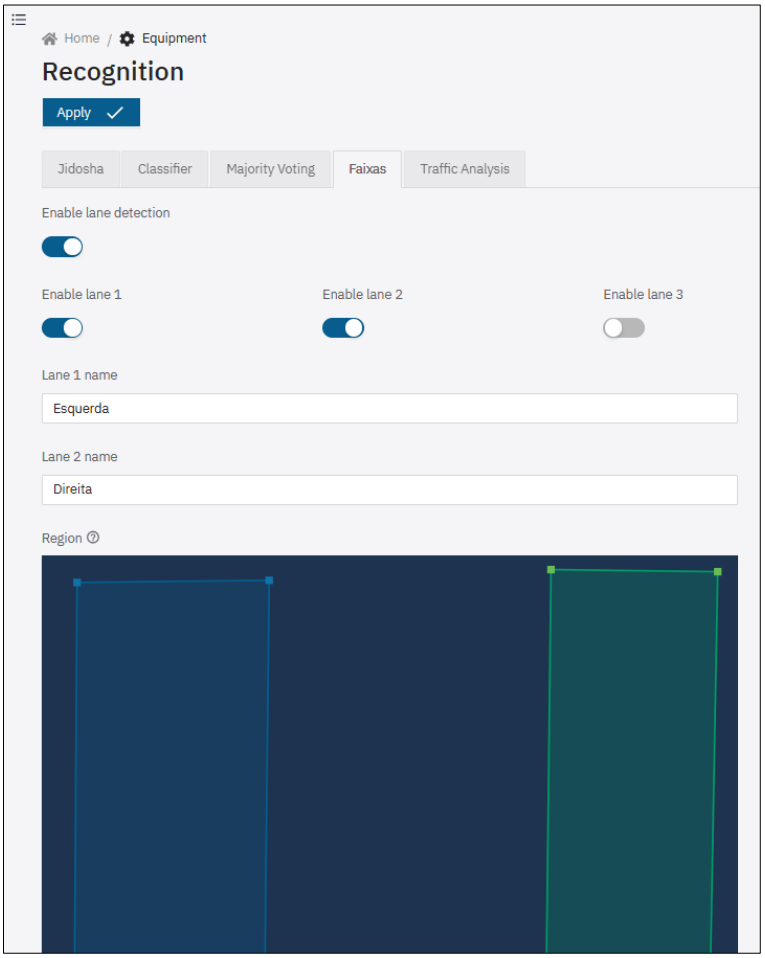
3.4.2. Majority Vote

1) In the *Majority Vote* tab, access the adjustment options by clicking on *Enable majority vote* when the *Multiple Exposures* feature is enabled and with the aim of the plate resulting from the OCR reading considering the most reliable detection for each character;
2) Enable the option *Send only the exposure with the best recognition* so that only the image with the best recognition index is sent;
3) Set the *Maximum number of different characters* tolerated to consider two plates as being equal in the *Multiple Exposures* reading;
4) Set a time, in seconds, for the *Same-plate recognition debounce*;
5) Select *Enable classifier filtering* which allows the use of the *Classifier* recognition to be used when enabled, and only images with a vehicle are generated;
6) Enable the option to *Forward images* without valid plates if triggered by the *Classifier tracking*, when the filtering option with the *Classifier* is disabled, and the images will be forwarded by the *Classifier* trace even when the detected plates are not considered valid;
7) Click *Apply* to validate the entered information.



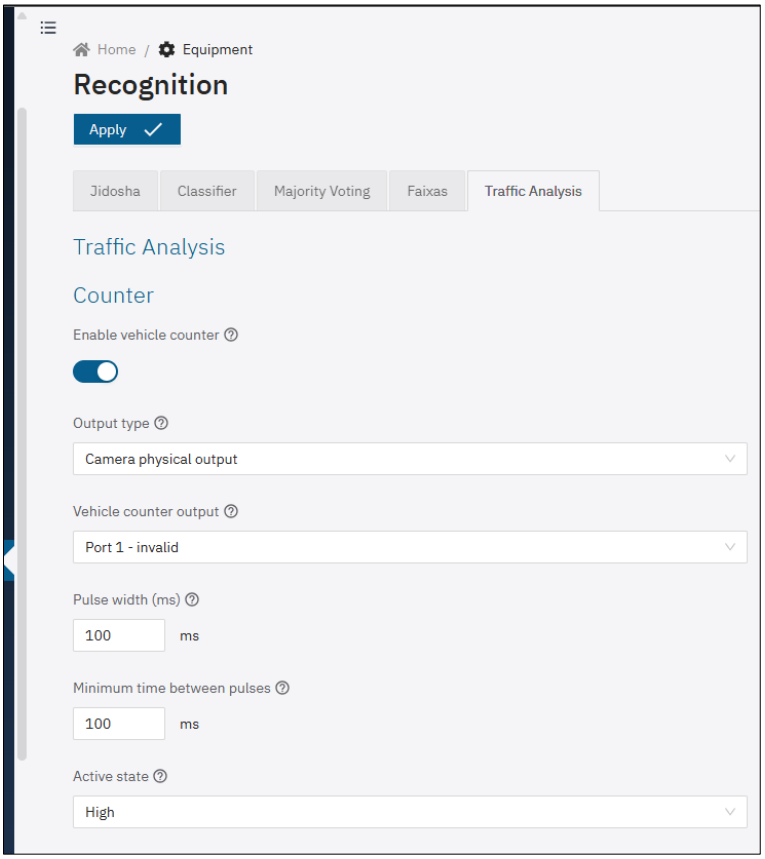
3.4.3. Lanes

1) Access the adjustment options in the *Lanes* tab by clicking on *Enable Lane Detection* to configure and name up to 3 lanes. Detected vehicles will have their lane estimated and displayed in the "vehicleList" comment of the images;
2) Click on which tracks will be enabled in the *Enable Track 1, Enable Track 2* and *Enable Track 3* options;
3) Set a name for the tracks that have been enabled;
4) Define the region of the image that corresponds to each track;
    a. Click on the vertex to drag the clipping and define the range detection area;

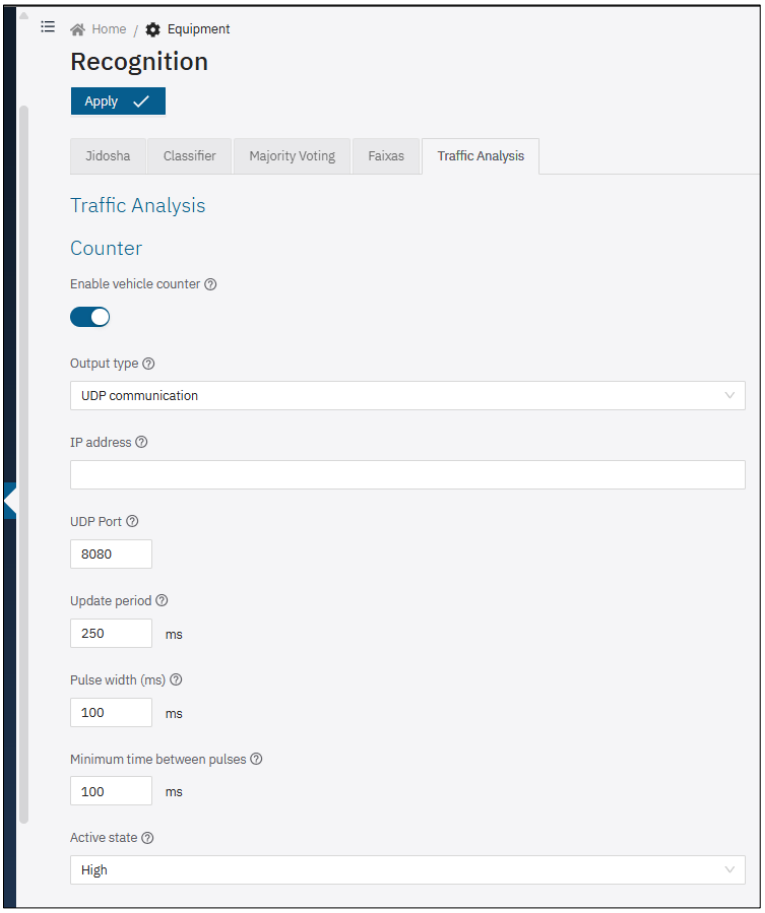


3.4.4. Traffic Analysis

1) Click on *Enable vehicle counter* and the pulse output for vehicle counting will be applied via the physical output port or UDP. Majority vote required;
2) Select which *Output Type* will be used from the options:
    a. *Camera Physical output*: set the *Vehicle counter output*, the *Pulse width (ms)*, the *Minimum time between pulses* and the *Active state* which defines whether the output pulse should follow positive or negative logic;



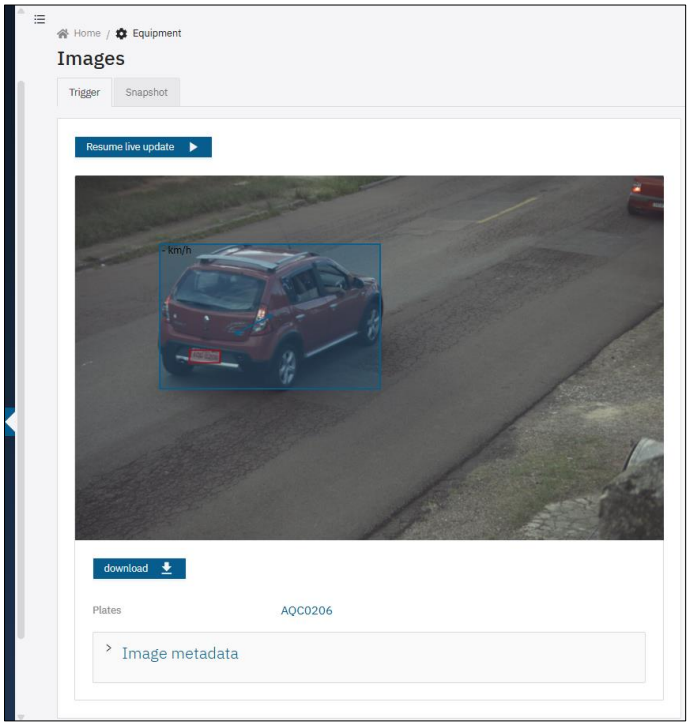
b. UDP Communication: configure by indicating the *IP Address* of the *UDP server* and the UDP Port receiving the pulses, the *Update Period* with the time between each send to the server indicating vehicle count or not, the *Pulse Width (ms)*, the *Minimum Time Between Pulses* with the minimum time for which the signal should be kept in the inactive state between two consecutive count pulses and the *Active State* indicating whether the output pulse should follow positive or negative logic, by selecting the *High* or *Low* option.



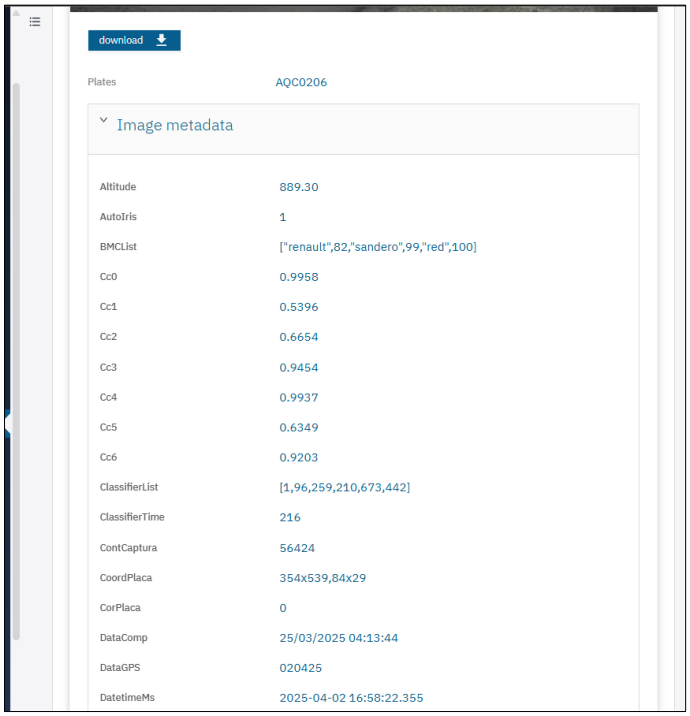
3.5. Checking the Images Generated

It is possible to manually generate a capture and view it, in order to check the local lighting, daytime or nighttime, by accessing the *Equipment > Images* menu:

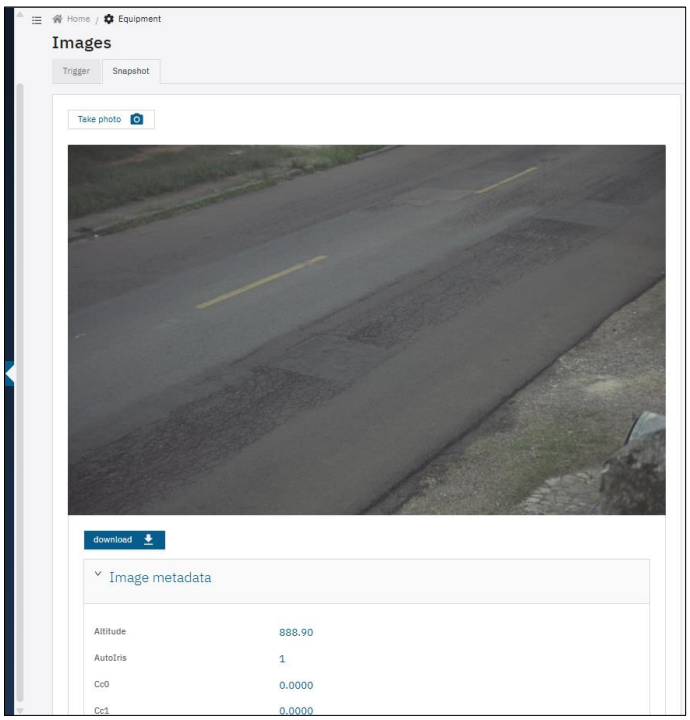
1) Select the *Trigger* tab and the last capture made is displayed, waiting for the last capture to be updated when necessary;



a. Click on *download* to download an instant JPEG image of the capture made by the device;
b. Click on *Image metadata* to view the metadata of the generated image;



2) Go to the *Snapshot* tab to access the current settings applied to the image;
   a. Click on *Take photo* and the generated image will be displayed;
   b. View by clicking on the *Metadata of the generated image* field;

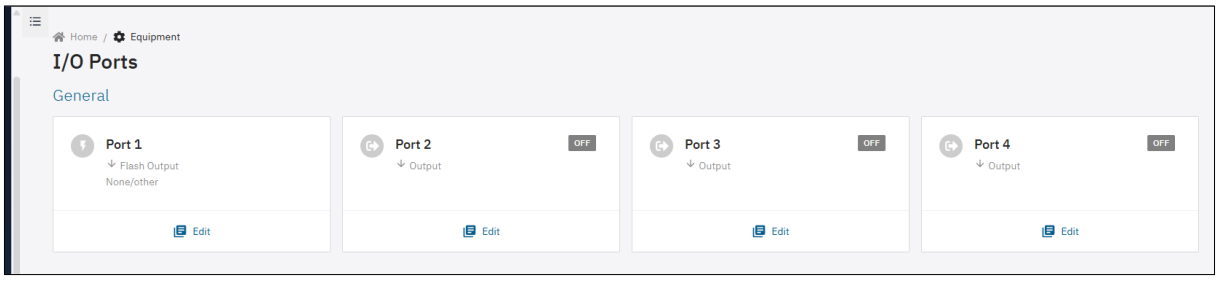


3) Make the necessary adjustments using the information from the current image settings.

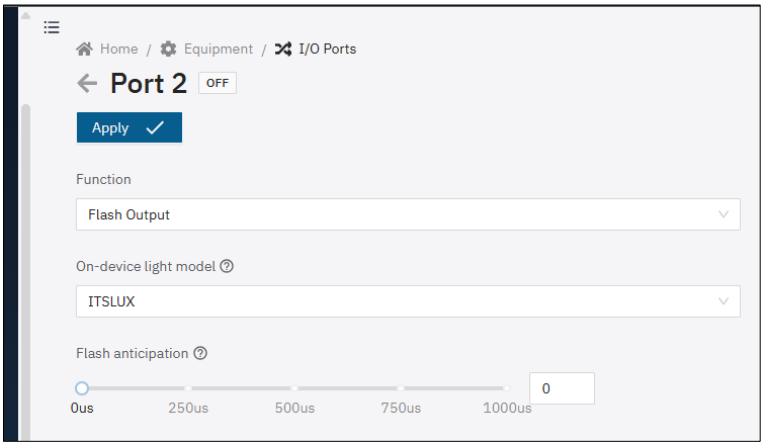
3.6. I/O Ports

ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+ and VTR 600 devices have 4 ports that must be configured by software as inputs or outputs (IOs), that are available to controlling the illuminator trigger or for the installation of external sensors, such as loops and light barriers, which identify the moment of image capture (trigger). To set up the use of each IO port:

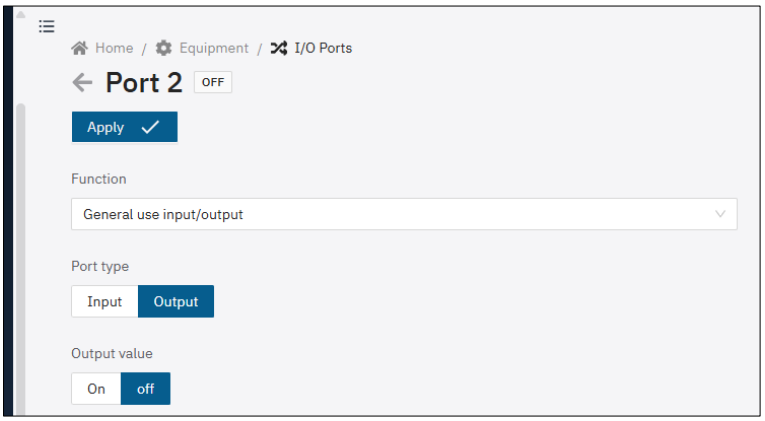
1) Access the settings in the menu *Equipment > I/O Ports*;
2) Locate the *Port* to be configured;
3) Click on the corresponding *Edit* button;



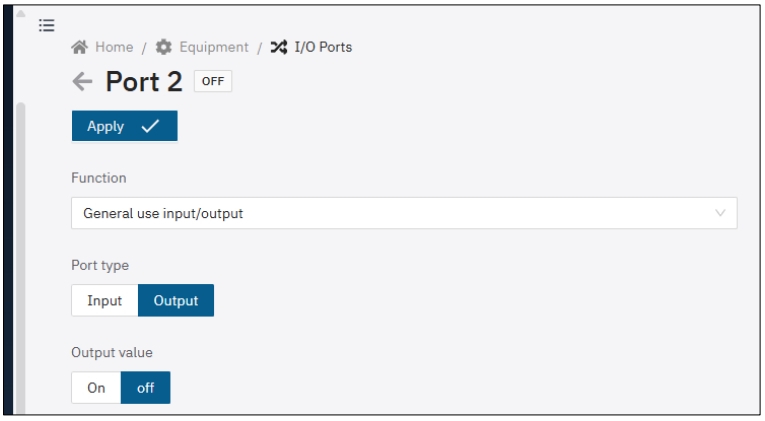
4) Select the *Function* as *Flash Output* when the IO port is used to flash the lighting device:
   a. Select the *On-device light model*: ITSLUX, WHITELUX, WHITELUX (video) or, if the lighting device is from another manufacturer, None/other;
   b. Set the *Flash anticipation* value to align the image capture with the flash at its peak light output on equipment that has a trigger delay.



5) Select the *Function* as a *General-purpose input/output* when receiving a loop signal or light barrier:
   a. Select the *Port Type* clicking on Input (example: when receiving a loop signal or light barrier);



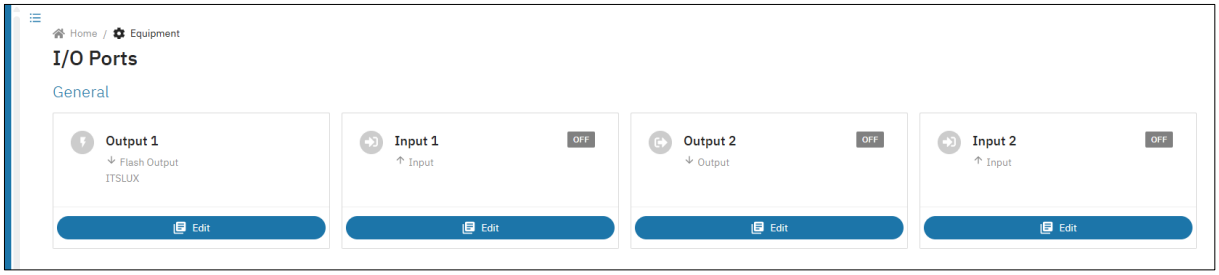
   b. Select the *Port Type* as *Output* (example: when triggering gates or alarms);
      i. Select *Output value* as *On* to activate the output signal;
      ii. Select *Output value* as *Off* to deactivate the output signal;
6) Apply the port settings by clicking *Apply*.



3.6.1. I/O Ports on the ITSCAM 450 and ITSCAM 450+

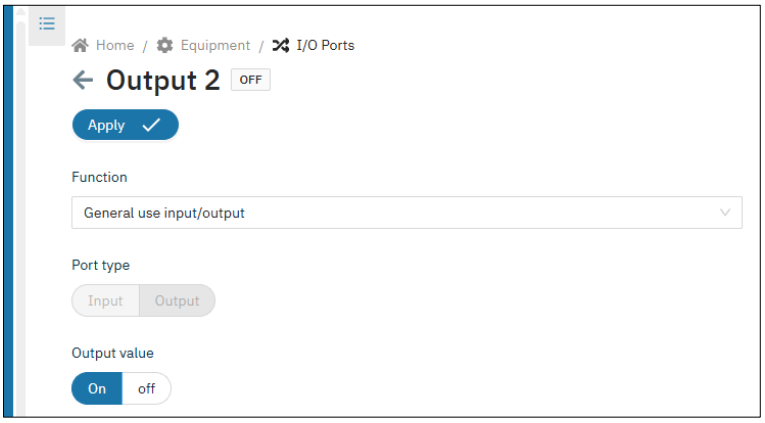
The ITSCAM 450 and ITSCAM 450+ devices have 2 inputs and 2 outputs, which are available for controlling the triggering of the illuminator or for installing external sensors, such as loops and light barriers, which identify the moment of image capture (trigger). To configure the outputs, access the menu *Equipment > I/O Ports*:

1) Locate the *Port* to be configured and click the *Edit* button;



2) Select the *Function* as *Flash Output* when the output port is used to trigger the lighting device flash;
   a. Select the *On-device light model* as *ITSLUX* and, if the lighting device is from another manufacturer, *None/other*;
   b. Set the *Flash anticipation* value to align the image capture with the flash at its peak light output on equipment that has a trigger delay.

3) Select the *Function* as *General-purpose input/output* when used, for example, to activate a gate;



   a. Select *Output value* as *On* to activate the output signal;
   b. Select *Output value* as Off to deactivate the output signal;

4) Apply the port settings by clicking *Apply*.

3.7. Servers

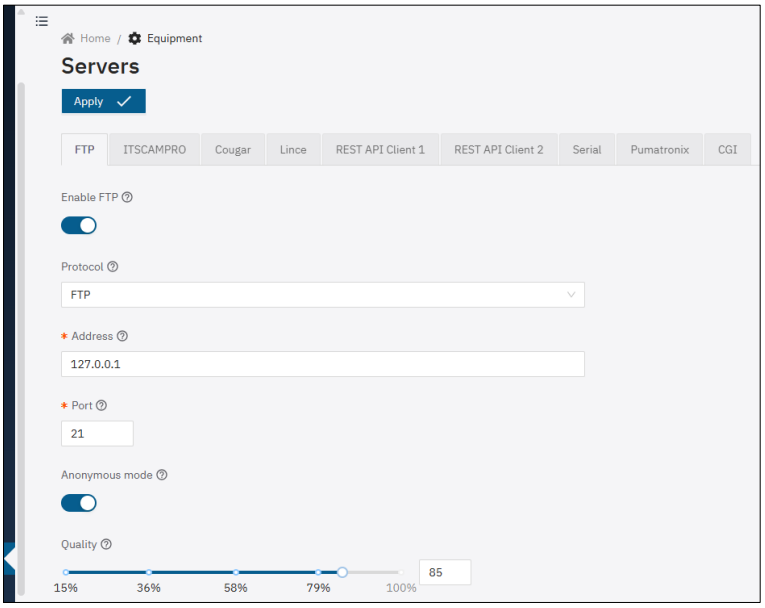
The devices can automatically send the images to an FTP server or to ITSCAMPRO, for example.

1) Go to the *Equipment > Servers* menu;

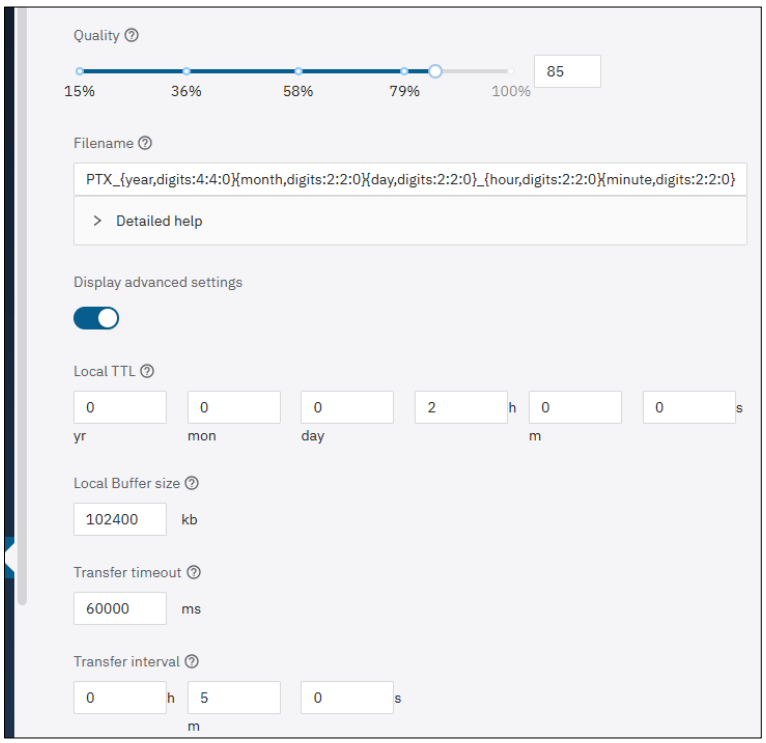
2) Select the tab corresponding to the server to be configured: *FTP*, *ITSCAMPRO*, *Cougar*, *Lince*, *REST API Client* or *Serial*;
3) Check Maintenance for the steps to restore the system to factory defaults, should any errors occur while configuring the servers.

3.7.1. FTP servers

1) Enable the FTP server by clicking *Enable FTP*;
2) Select the Protocol to be used from the options:
    a. *FTP*: Basic file transfer protocol;
    b. *FTPS*: SSL/TLS secure file transfer protocol;
    c. *SFTP*: SSH file transfer protocol.
3) Fill in the IP *Address* and access *Port*;
4) Define a *User* and *Password*;
5) Enable *Anonymous Mode* when not using username and password;



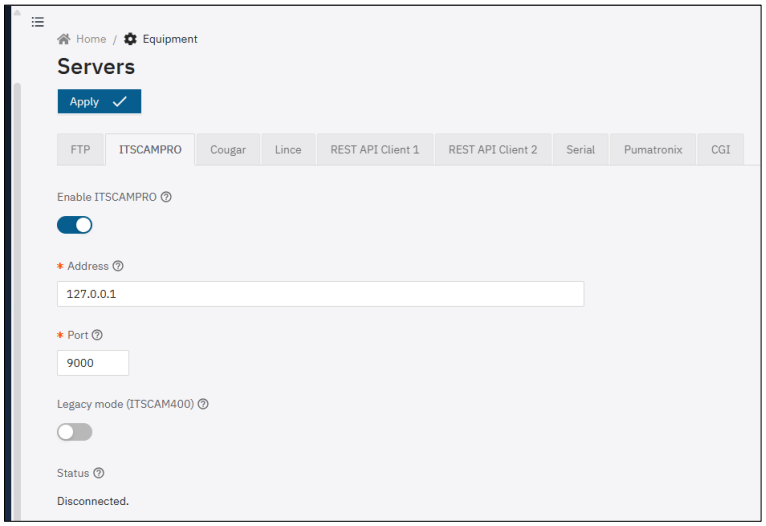
6) Adjust the quality of the JPEG image when saving via FTP by selecting it in the *Quality* field;
7) Change the code in *Filename* to customize the name of the file with information from the capture. To ensure that the name is in the correct format, a validation mechanism has been implemented that indicates if any field is filled in incorrectly. To consult, expand the *Detailed help* option;
8) Enable the *Display advanced options* option and check/adjust the storage options that the FTP upload service will use:
    a. Local *TTL*: If it is not possible to connect, the device will temporarily keep the images for the specified time;
    b. Local buffer size: If it is not possible to connect, the device will keep the images temporarily, limited by the size specified;
    c. Transfer timeout: time limit for an individual FTP transfer;
    d. Transfer interval: After uploading all the images, the device disconnects from the FTP server and only reconnects after the specified time;
9) Click *Apply* after checking the data entered.



3.7.2. ITSCAMPRO Server

Devices can be enabled to send captures to an external ITSCAMPRO server:

1) Select the *Enable* ITSCAMPRO option on the ITSCAMPRO tab;
2) Fill in the IP *Address* of the ITSCAMPRO server, which must be a domain name or a valid IPv4 address;
3) Enter the access *Port*, between 1 and 65535;



4) Select *Legacy Send Mode (ITSCAM 400)* when the P0 protocol is to be used for sending data. This mode is compatible with ITSCAM400 and should only be used on legacy systems;
5) Click *Apply* to validate the data entered;
6) Check in *Status* that the server is *Connected*.

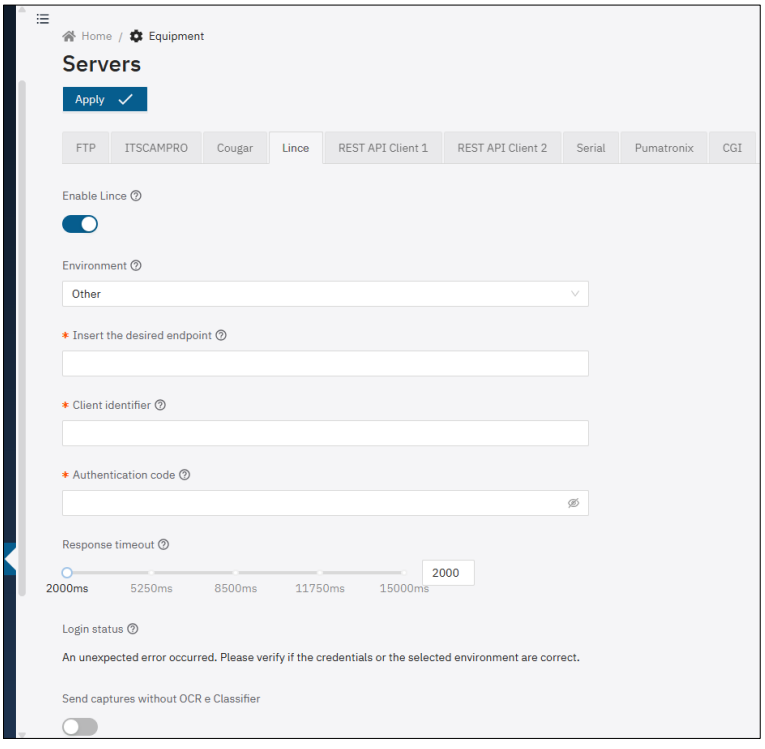
3.7.3. Cougar Server

1) In the Cougar tab, select the *Enable* authentication option;

2) Set a *Password* to authenticate with Cougar, using up to 64 characters, as *Cougar* can configure multiple aspects of the device;
3) Click *Apply* after checking the data entered.

3.7.4. Lince server

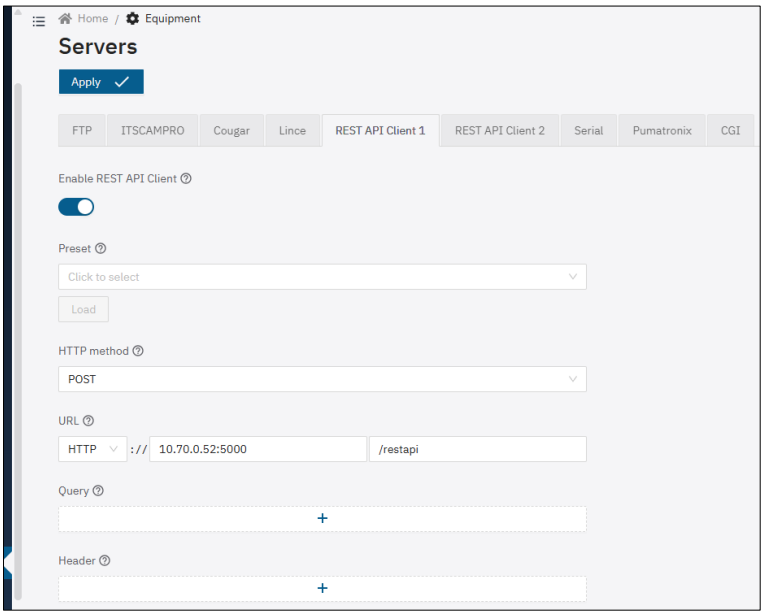
1) In the *Lince* tab, select the *Enable Lince* option;
2) Select the server operating *Environment* from the options: *Development, Homologation, Production* or *Other*;
3) *Enter the desired endpoint* by entering the desired URL for sending the captures via the Lince server. Example: lince.app.br or lince.app.br:1443;
4) Enter a *Client Identifier* for the Lince server;
5) Enter an *Authentication Code* for the Lince server;
6) Set the Lince server *Response Timeout* between 2,000 and 15,000 milliseconds;
7) Check the *Login status* of the last login attempt to send records;
8) Select the *Send captures without OCR* and *Classifier* option so that images without recognition are also sent;
9) Click *Apply* after checking the data entered.



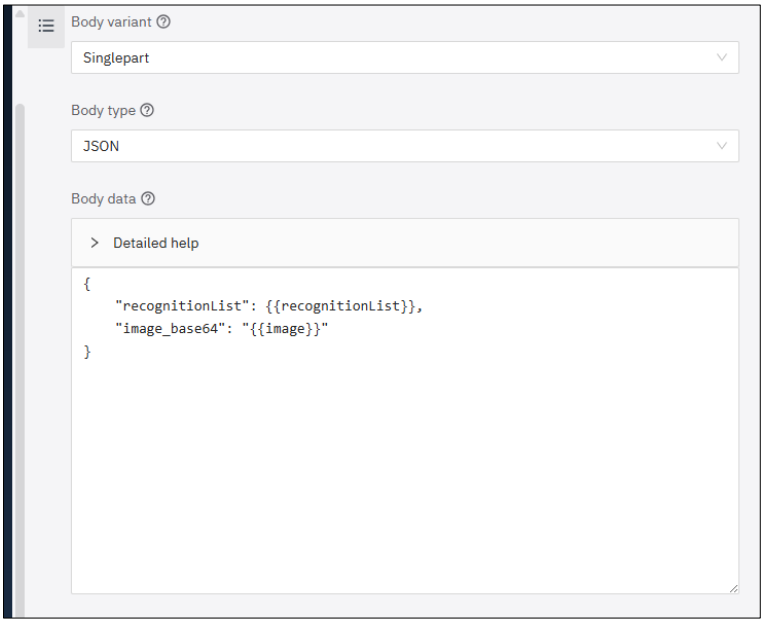
3.7.5. REST API Client Server

The devices support sending captures to a generic HTTP server, resizing the image and retrying sending.

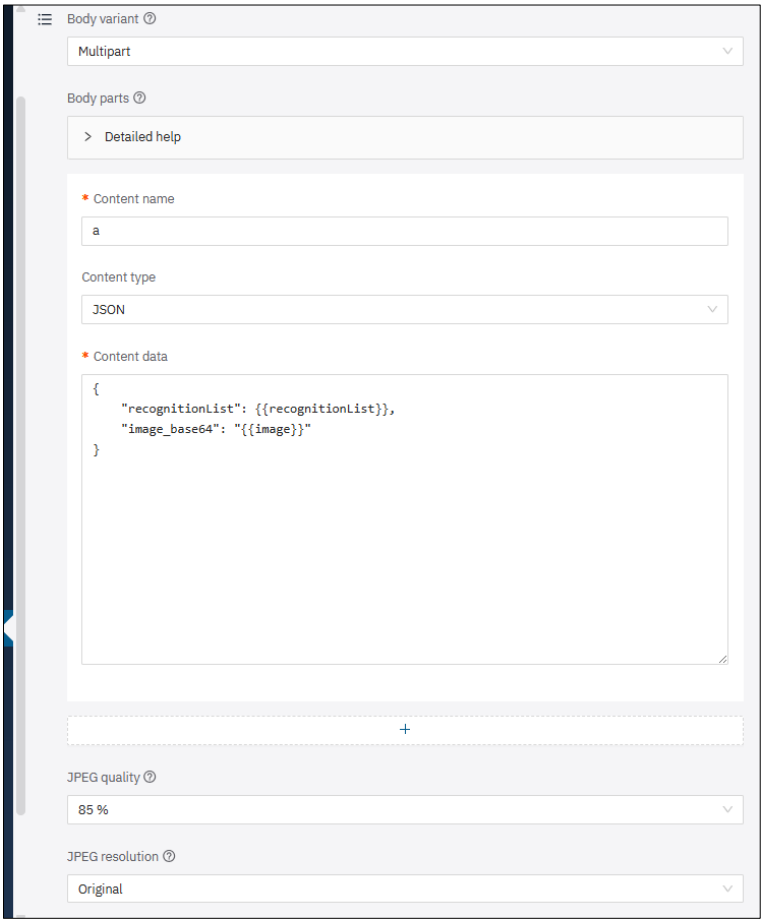
1) In the *REST API Client* tab, select the *Enable REST API Client* option, which allows sending captures to a generic HTTP server;
    a. Select a Preset under *Presets* to apply a preset to some fields by clicking the *Load* button;
2) Select the *HTTP Method* of the custom request from GET, POST and PUT;
3) Enter the URL for the custom request, indicating the schema, host and path filled in separately;
4) Enter the *Query* parameters for the custom request by clicking +;
5) Enter additional headers for the custom request in *Header* by clicking + and entering name and value;



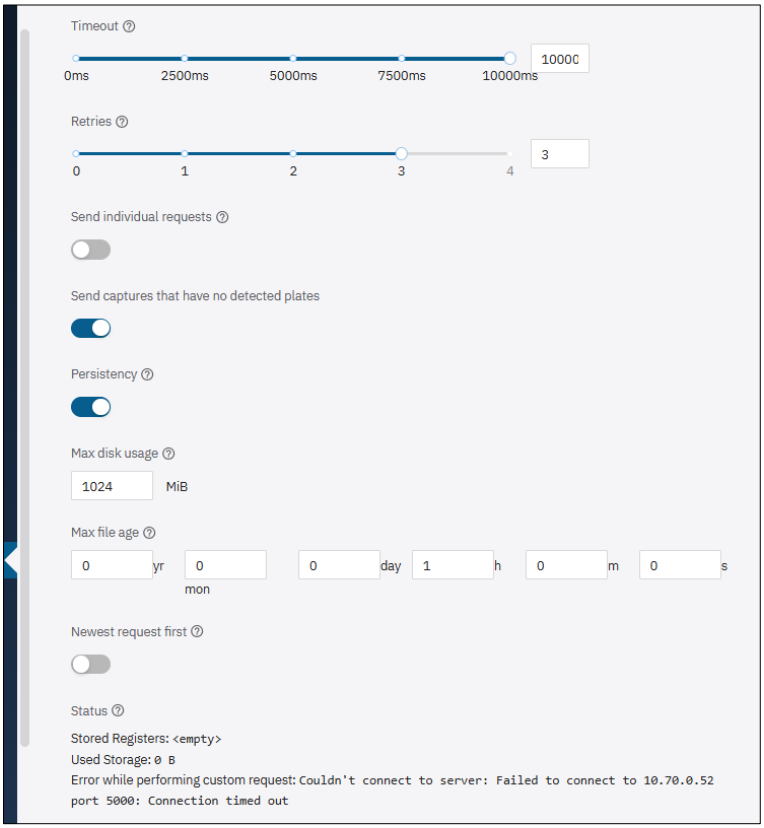
6) Select the *Body variant* request variant as *Singlepart*;
    a. Select the custom request *Body Type* (Content-Type header is added automatically) from the *JSON*, *JPEG* and *Form* (*URL encoding*) options;
    b. Check and edit the custom request *Body Data* (Content-Length header is added automatically) by replacing the variables with variable names surrounded by double braces, considering the variables available in the *Detailed help or in* *Rest API Available Fields*;



7) Select *Body variant* as *Multipart* and the request body is made up of several 'contents', each requiring a name, type and data:
    a. Identify with a *Name of the content*;
    b. Specify the *Type of the custom* request content by selecting between the *JSON*, *JPEG* and *Form (URL encoding)* options;
    c. Check and edit the custom request *Content Data* (Content-Length header is added automatically) by replacing the variables with variable names surrounded by double braces, considering the variables available in the *Detailed Help* for *Body Parts*;
8) Select the *JPEG Quality* that is sent in the body between *Standard* or between 5% and 95%;
9) Select the *JPEG Resolution* that is sent in the body, bearing in mind that an image ratio different from the original will cause stretching when resizing (if this is a problem, it will be required to select an image crop with the same ratio);



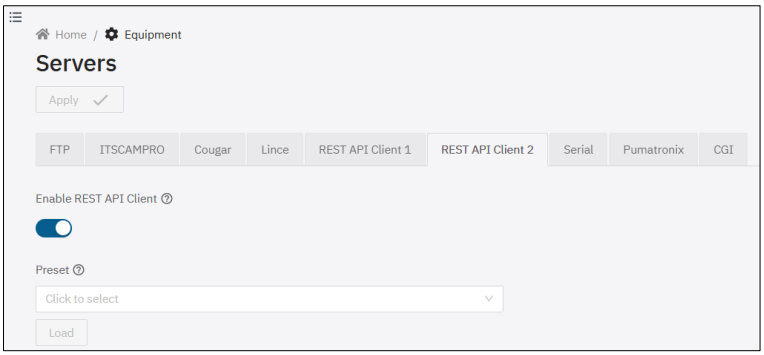
10) In *Timeout*, enter the time interval, in milliseconds, in which the custom request is canceled if there is no response from the server;
11) Indicate the number of *Retries* that are performed on the personalized request again in the event of a failure. Note that variable substitution errors do not count as failures;
12) Select whether the device should *Send individual requests*, considering that a request will be sent for each plate instead of one request per group of exposures;
13) Select whether the device should *Send captures that have no plates detected*, considering that captures in which there is no plate character recognition will be sent;
14) Select *Persistency* to save information on disk when the request fails and try to send it again later;
15) Enter the amount in Mib of *Maximum disk usage* to persist failed requests;
16) Enter the *Maximum file age* of failed requests, considering that failed requests older than this value are discarded;
17) Select whether the device will make the *Newest request first*, considering that requests will be made from the newest to the oldest rather than from the oldest to the newest;
18) Check *Status* for information about the last custom request made by the REST API Client;
19) Click *Apply* after checking the data entered.



3.7.6. REST API Client Server 2

It is possible to create a second REST API client, allowing data to be sent to up to 2 servers with different APIs.

1) Select the *Enable REST API Client* option on the *REST API 2 Client* tab;
2) Perform the steps indicated in the *REST API Client Server* configuration, but inserting the information of the other API server.



With the inclusion of the second *REST API Client* on the servers page, the *Client* data storage structure has been changed. Old records will be ignored and will take up disk space. Therefore, if the service is used (even if only one *Client*), it is necessary to *Delete capture data* by accessing *Storage Maintenance*, after updating to version 1.7.6.

3.7.6.1. Rest API Available Fields

It is possible to use variable substitution using variable names surrounded by double braces. The variables available are:

Variable	Description
cameraId	Equipment name
equipmentId	MAC address of the equipment
gpsHdop	Dilution of GPS horizontal accuracy
image	JPEG of the capture, base64 encoded. It is possible to send a data URL by prefixing this field with extra information: "{{image}}"
imageList	JSON list of the JPEGs (base64 encoded) of each exposure. This variable does not need to be enclosed in square brackets in the content model.
imageRaw	JPEG of the capture, in "raw bytes". This variable is treated in a special way and is only replaced when the content is exactly "{{imageRaw}}", since otherwise an invalid JSON would be generated.
imageRawList	List of JPEGs of all exposures, in "raw bytes". This variable is also treated in a special way, being replaced only when the content is exactly "{{imageRawList}}". It should be used in multipart forms, causing multiple files to be sent.
latitude	Equipment coordinate (decimal degree format)
localDay	Day ("DD" format) of the current date/time (local time zone)
localHours	Hours ("HH" format) of the current date/time (local time zone)
localMilliseconds	Milliseconds ("mmm" format) from the current date/time (local time zone)
localMinutes	Minutes ("MM" format) from the current date/time (local time zone)
localMonth	Month ("MM" format) of the current date/time (local time zone)
localSeconds	Seconds ("SS" format) from the current date/time (local time zone)
localYear	Year ("YYYY" format) of the current date/time (local time zone)
longitude	Equipment coordinate (decimal degree format)
plate	Characters of the detected plate
plateBoundingBox	Coordinates of the plate in the image ("x,y,w,h" format)
plateProbability	Reliability of each character on the plate, separated by a comma
recognitionList	List of all recognitions. Each recognition is an object containing the "imageIndex" field and the optional "plateInfo" and "vehicleInfo" fields. "plateInfo" is an object with fields "plate", "plateProbability" and "plateBoundingBox". "vehicleInfo" is an object with fields "vehicleBoundingBox", "vehicleType" and "vehicleTypeProbability" and optional fields "vehicleBrand", "vehicleBrandProbability", "vehicleColor", "vehicleColorProbability", "vehicleModel" and "vehicleModelProbability" when vehicle characteristics are enabled. This variable does not need to be enclosed in square brackets in the content model.
registerId	Current record identifier
utcDay	Day ("DD" format) of the current date/time (UTC time zone)
utcHours	Hours ("HH" format) of the current date/time (UTC time zone)
utcMilliseconds	Milliseconds ("mmm" format) from the current date/time (UTC time zone)
utcMinutes	Minutes ("MM" format) from the current date/time (UTC time zone)

Variable	Description
utcMonth	Month ("MM" format) from the current date/time (UTC time zone)
utcSeconds	Seconds ("SS" format) from the current date/time (UTC time zone)
utcYear	Year ("YYYY" format) of the current date/time (UTC time zone)
vehicleBoundingBox	Coordinates of the vehicle in the image ("x,y,w,h" format)
vehicleBrand	Brand of detected vehicle
vehicleBrandProbability	Reliability of the detected vehicle brand
vehicleColor	Color of the detected vehicle
vehicleColorProbability	Reliability of the detected vehicle color
vehicleModel	Vehicle model detected
vehicleModelProbability	Reliability of the detected vehicle model
vehicleType	Type of vehicle detected
vehicleTypeProbability	Reliability of the type of vehicle detected

Attention: The "plate*" and "vehicle*" fields show the data of the first vehicle detected in the image. For multiple vehicles, use the "recognitionList" field.

It is possible to insert also static data (such as lane direction, public key and so on) into the JSON in the creation field of the JSON template, as in the "direction" field in the example below.

JSON example with all fields included:

```
JavaScript
{
            "cameraId": "{{cameraId}}",
            "equipmentId": "{{equipmentId}}",
            "registerId": "{{registerId}}",
            "timestamp":
"{{utcYear}}-{{utcMonth}}-{{utcDay}}T{{utcHours}}:{{utcMinutes}}:{{utcSecond
s}}.{{utcMilliseconds}}Z",
        "local_timestamp":
"{{localYear}}-{{localMonth}}-{{localDay}}T{{localHours}}:{{localMinutes}}:{
{localSeconds}}.{{localMilliseconds}}Z",
        "latitude": {{latitude}},
        "longitude": {{longitude}},
        "gpsHdop": {{gpsHdop}},
        "recognitionList": {{recognitionList}},

        "plate": "{{plate}}",
        "plateBoundingBox": [{{plateBoundingBox}}],
        "plateProbability": [{{plateProbability}}],
        "vehicleBoundingBox": [{{vehicleBoundingBox}}],
        "vehicleBrand": "{{vehicleBrand}}",
        "vehicleBrandProbability": {{vehicleBrandProbability}},
        "vehicleColor": "{{vehicleColor}}",
        "vehicleColorProbability": {{vehicleColorProbability}},
```

```
        "vehicleModel": "{{vehicleModel}}",
        "vehicleModelProbability": {{vehicleModelProbability}},
        "vehicleType": "{{vehicleType}}",
        "vehicleTypeProbability": {{vehicleTypeProbability}},
        "sentido": "crescente",
        "imageList": {{imageList}}
}
```

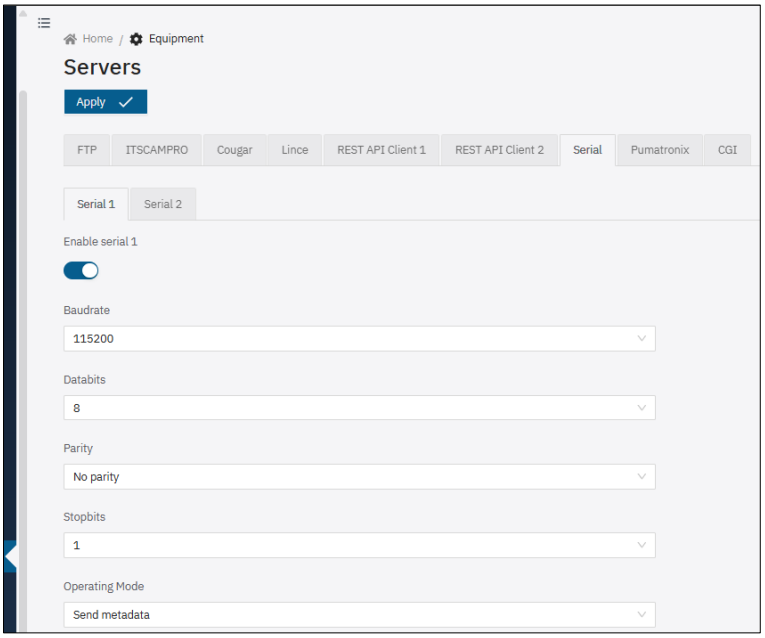
Example of data sent:

```
JavaScript
{
  "cameraId": "ITSCAM 600 - Carlos Laet - (Thiago Trannin: teste long run
gerenciador de capturas)",
  "equipmentId": "F8-D4-62-01-4E-32",
  "registerId": "4856387",
  "timestamp": "2024-11-26T13:42:39.145Z",
  "local_timestamp": "2024-11-26T10:42:39.145Z",
  "latitude": -25.48764228820801,
  "longitude": -49.24016952514648,
  "gpsHdop": 0.7,
  "recognitionList": [
        {
              "imageIndex": 0,
              "plateInfo": {
                    "plate": "CBH0599",
                    "plateBoundingBox": [358, 182, 78, 23],
                    "plateProbability": [
                          0.9999926090240, 0.9999926090240, 0.9999926090240,
                          0.9999926090240, 0.9999926090240, 0.9999926090240,
                          0.9999926090240
                    ]
              }
        },
        {
        "imageIndex": 0,
              "vehicleInfo": {
                      "vehicleBoundingBox": [541, 0, 658, 438],
                      "vehicleBrand": "fiat",
                      "vehicleBrandProbability": 0.8452616333961487,
                      "vehicleColor": "silver",
                      "vehicleColorProbability": 0.5213572978973389,
                      "vehicleModel": "unknown",
```

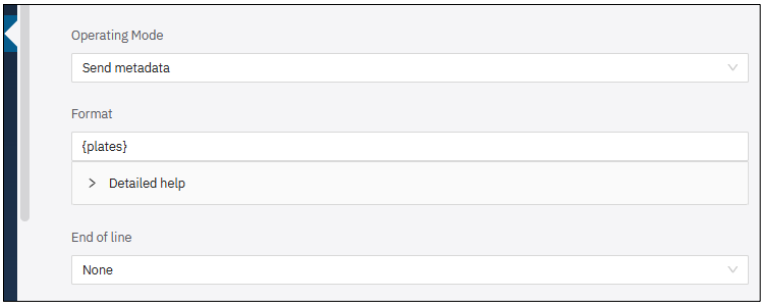
```
                    "vehicleModelProbability": 0.0,
                    "vehicleType": "car",
                    "vehicleTypeProbability": 0.0
                }
            }
    ],
    "plate": "CBH0599",
    "plateBoundingBox": [358, 182, 78, 23],
    "plateProbability": [1.0, 0.96, 0.98, 0.99, 1.0, 1.0, 1.0],
    "vehicleBoundingBox": [0, 0, 0, 0],
    "vehicleBrand": "",
    "vehicleBrandProbability": 0.0,
    "vehicleColor": "",
    "vehicleColorProbability": 0.0,
    "vehicleModel": "",
    "vehicleModelProbability": 0.0,
    "vehicleType": "",
    "vehicleTypeProbability": 0.0,
    "sentido": "crescente",
    "imageList": ["/9j/4A<REST OF IMAGE IN BASE64 format>ABA=="]
}
```


3.7.7. Serial Interface

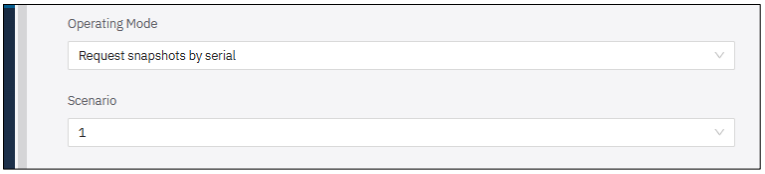
1) Select the *Serial* tab to configure the device's serial interfaces;
2) In the *Serial 1* tab, select the *Enable serial 1* option;
3) Select the *Baudrate* from the options: 4800, 9600, 19200, 38400, 57600 or 115200;
4) Select the *Databits* from the options: 5, 6, 7, 8 or 9;
5) Select *Parity* from the options: No parity, Odd or Even;
6) Select the *Stopbits* from the options: 0, 1, 1.5 or 2.
7) Select the *Operation Mode* between the options *Send metadata* or *Request captures by serial*;



8) In *Operation Mode > Sending metadata*, indicate the *Format* by replacing the variables with variable names surrounded by double braces, taking into account the variables available in the *Detailed Help*;
   a. Select *End of line* in the options: None, <CR>(\r), <LF>(\n) or <CR><LF>(\r\n);



9) In Operation Mode > Request captures via serial, select Scenario from the options: No scenario, 1 or 2;



10) Select the *Serial 2* tab to configure the device's serial 2 interface;
11) Click *Apply* after checking the data entered.

3.7.8. Pumatronix Server

1) Enable the *Legacy Mode for Pumatronix Protocol* and the *PhotoIndex* field in the image comments indicates the numbering 0 for video/preview frames and 1,2... for trigger/snapshot. When disabled, the PhotoIndex field is numbered 0 for video or trigger and 1,2... for trigger.

3.7.9. Authentication for config.cgi and reboot.cgi

1) Select *Enable authentication for config.cgi and reboot.cgi* to protect access to *config.cgi* and *reboot.cgi* and they will require authentication for access. User and password for authentication are the same as for the web interface.

4. System Settings

4.1. Plugins

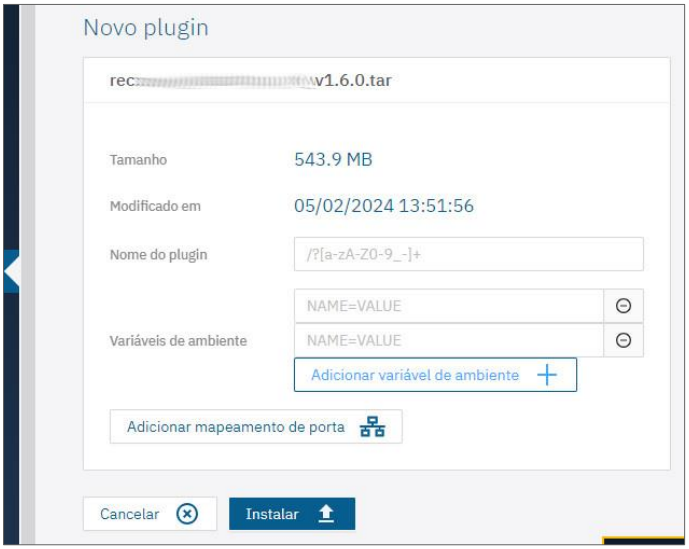
Importing plugins directly via the web interface and configuring more than one port with external mapping is possible for ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+ and VTR 600 devices. VTR 600 receives the ITSCAMPRO Mobile plugin installation from the factory.

Proceed with the installation of plugins after formatting the SD card, following the steps:

1) Go to *System* > *Plugins*;

2) Click or drag a *.tar* file into the *New Plugin* area;
3) Enter the *Name of the plugin* that describes its use;
4) Configure the plugin's *Environment variables* by clicking on *Add environment variable*;
5) Click *Add port mapping* when an internal container port needs to be exposed on the device, according to the plugin in use;

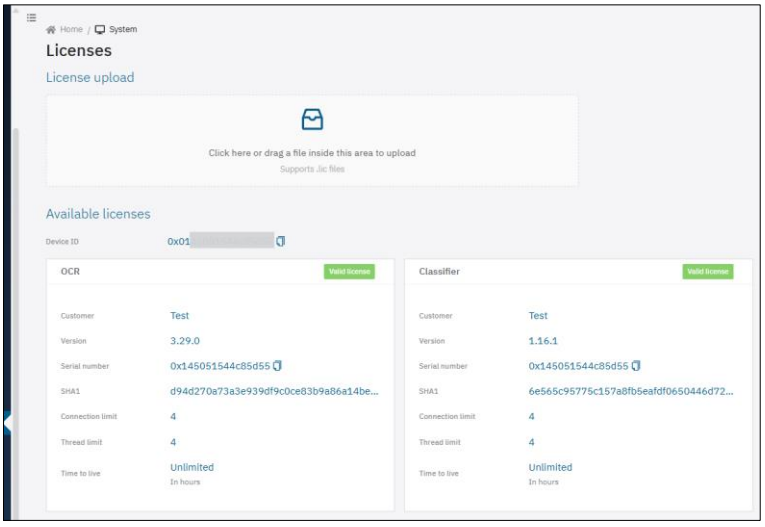
6) Click *Install* after checking the data entered.



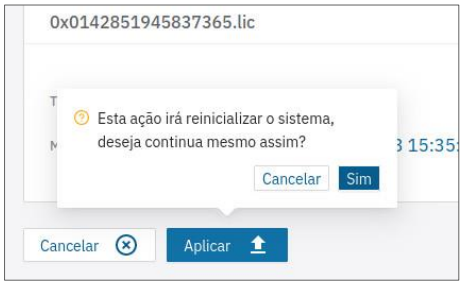
4.2. Licenses

Analytics licenses can be updated directly via the web interface, either for automatic recognition of the license plate of the vehicles in the images (OCR) or for recognition of the type of vehicle identified (*Classifier*).

1) Go to the *System* > *Licenses* menu to update the licenses;
2) Click or drag the file in *.lic* format to the *License Upload* area, provided by Support or Commercial departments, when available;



3) Click on *Apply*;
4) Confirm on the reboot prompt if it is possible to reboot the system after applying the license:



5) Wait for the file to load completely.

4.3. User Access Management

Capture devices allow greater control of access and changes made to the device, as multiple users can be created. Users configured with the *Administrator* profile can configure the equipment, users and view images. Users with an *Operator* profile can view images and settings. To manage active users, access *System > Users*:

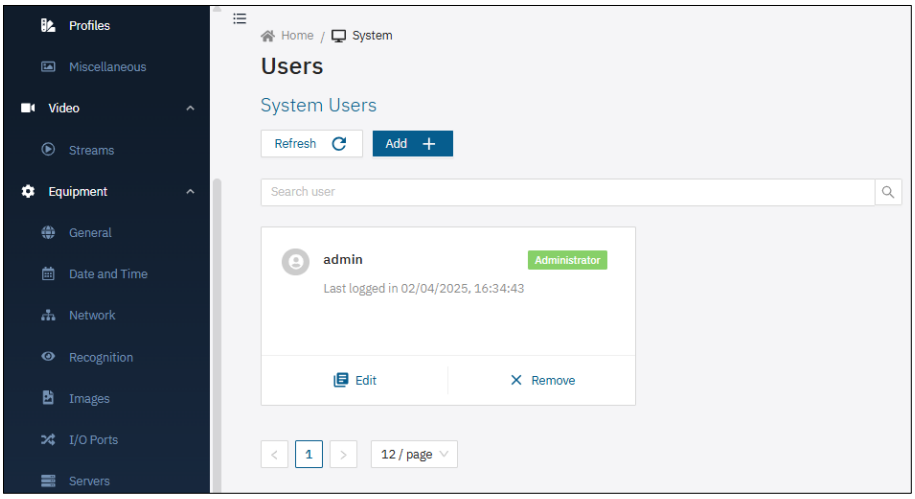
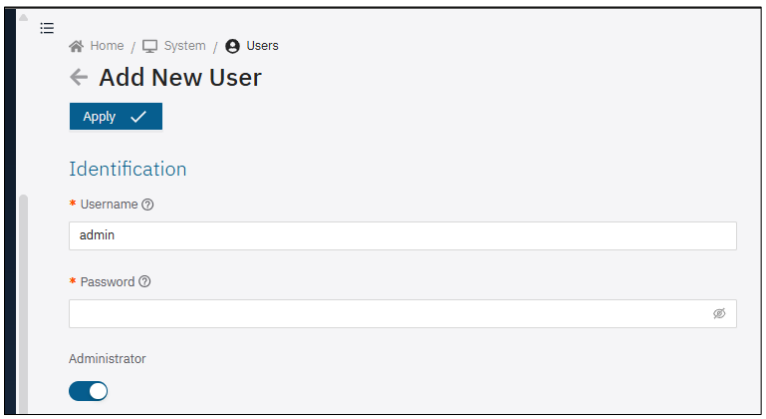


Figure 18 - User management home screen

1) Create a new user by clicking *Add+*;
2) Edit the data of the existing user by clicking on the *Edit* button;
3) Identify with a unique *Name* using between 4 and 200 characters, with letters and numbers and no spaces;
4) Create an access *Password* containing between 4 and 200 characters, numbers, letters and special characters or leave it blank to keep the current password;
5) Validate the edit by clicking *Apply*.

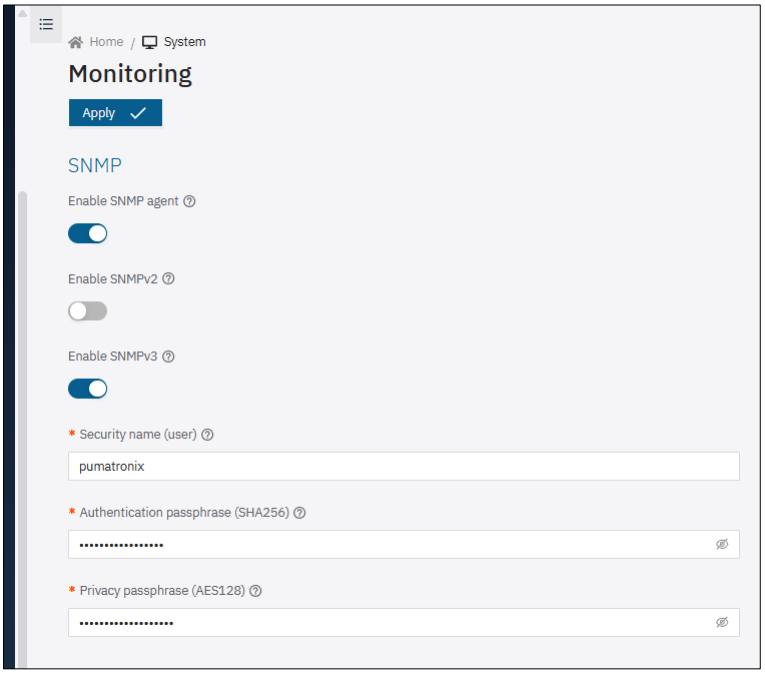


The factory default user and password should be changed for better access control and greater security.

4.4. Monitoring

The capture device can be monitored remotely using the *SNMP* protocol, configured directly via the web interface.

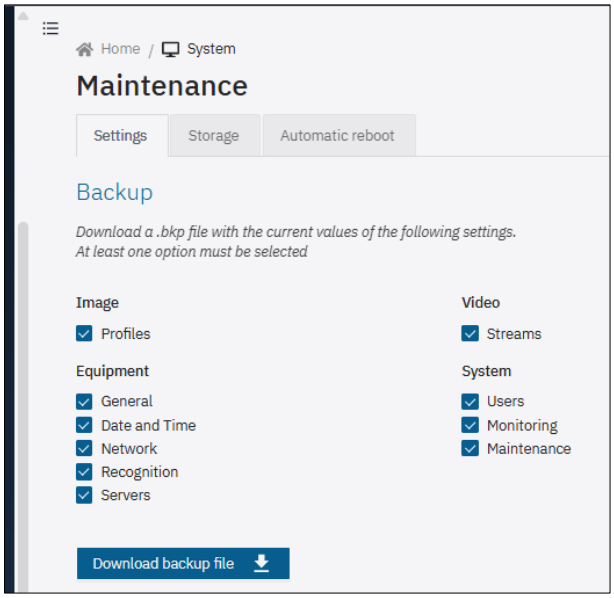
1) Go to the System > Monitoring menu;
    a. Select the *Enable SNMP agent* option to enable the integrated SNMP agent;
    b. Enable version 3 of the SNMP protocol in the *Enable SNMPv3* option:
        i. Define a *Security Name (user)*, also called "user". The context name is an empty string (mandatory field);
        ii. Set an *Authentication Password* (SHA256) with the SHA256 authentication protocol (mandatory field);
        iii. Set a *Privacy password* (AES128) with the AES128 privacy protocol (mandatory field);
    c. Enable version 2 of the SNMP protocol in the *Enable SNMPv2* option, since SNMPv2 is insecure by default, SNMPv3 should be used whenever possible:
        i. Enter the String in the *Community* field;
2) Click *Apply* after confirming the data entered:



4.5. Maintenance

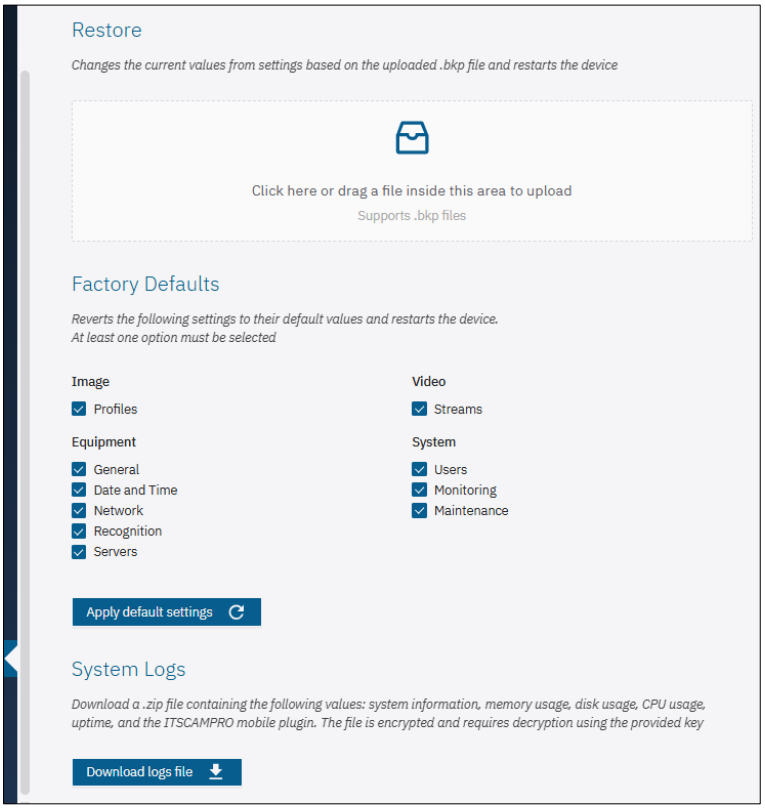
The *Backup*, *Restore*, *Factory Defaults and Automatic Reboot* maintenance options are available via the *System > Maintenance* menu.

1) Locate the *Backup* field to save a backup file of the settings, which can be restored on the device itself or to import the configuration to other devices;
    a. Select the settings that will be saved in the backup file;
    b. Click on *Download log file*;



2) Locate the *Restore* field to use a backup, in which case the current settings will be overwritten by the information saved in the file;
    a. Click or drag a file in *.bkp* format;
    b. Wait for the file to load and for the device to reboot;

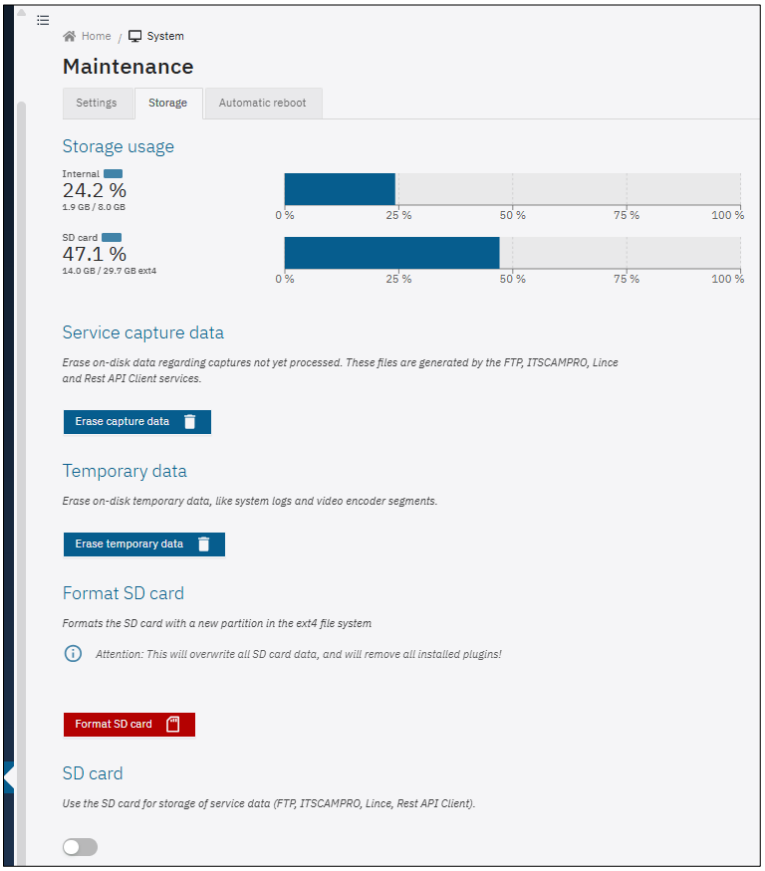
3) Locate the *Factory Defaults* field to restore the factory settings in the event of a device malfunction or incorrect settings;
    a. Select the settings that will be reset to the factory default;
    b. Click on *Apply default settings*;
    c. Wait for the device to reboot;



4) Locate the *System Logs* field to download a *.zip* file with the main system logs: system, memory usage, disk usage, CPU usage, connected time and the ITSCAMPRO mobile plugin. The file is encrypted, so it must be decrypted using the key provided:
    a. Click on *Download log file*;
    b. Unzip the zipped file;
    c. Access the text files, locating the data from each log separately.

4.5.1. Storage Maintenance

1) In the *System > Maintenance* menu, go to the *Storage* tab:
2) Check under *Storage Usage* the space in use of the *Internal* storage;
3) Check *Storage Usage* to see if there are any files saved on the SD card;
4) Only click *Format SD card* if it is certain that the plugin files can be overwritten and replaced. The import of plugins requires that an *ext4* formatted SD card be inserted in the capture device;



a. If the SD card is not present, the device will display the following error message:

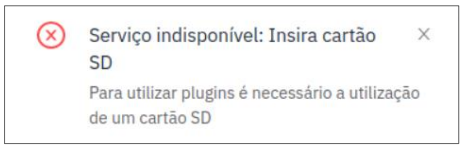
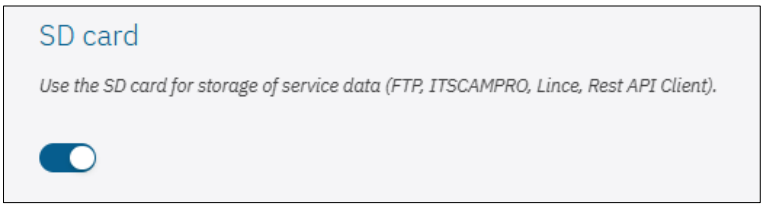


Figure 19 – Error message displayed if SD card is not present

5) Select the option to delete the *services' Capture Data* and the data saved on disk relating to captures that have not yet been processed will be deleted. These files are generated by the FTP, ITSCAMPRO and Lince services;
6) Select the option to delete *Temporary Data* and the temporary data saved on disk, such as system logs and video encoder segments, will be deleted.
7) On ITSCAM 600 and ITSCAM 600 FHD devices, it is possible to enable the option Use SD card to store service data (FTP, ITSCAMPRO, Lince, Rest API Client).

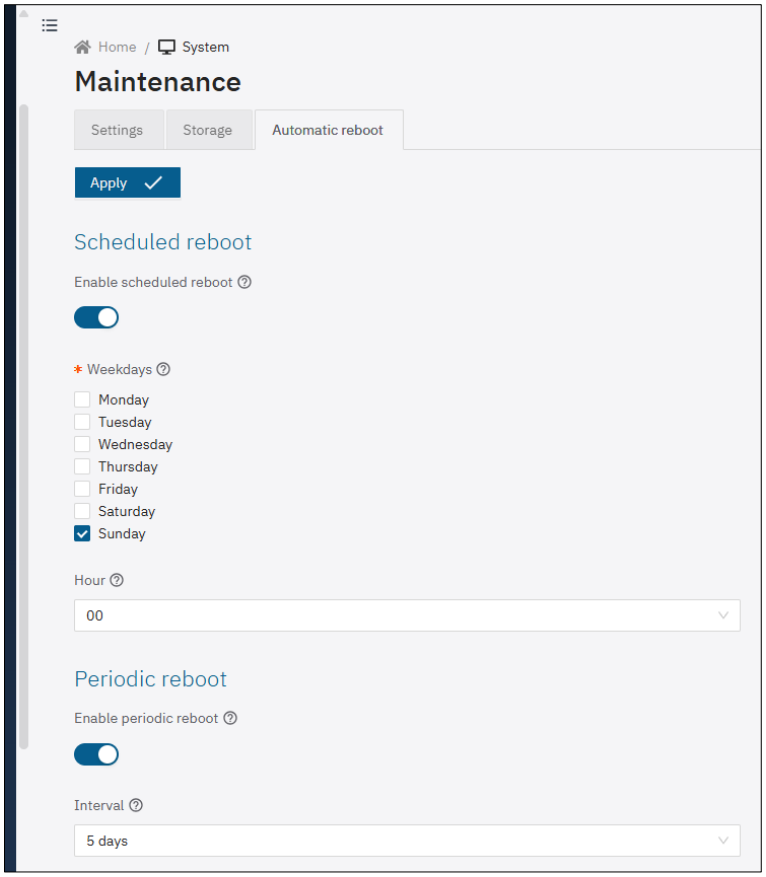


4.5.2. Automatic Reboot

1) Program the Automatic Reboot of the device on a scheduled day and time or periodically for improved system operation:
    a. Select *Enable scheduled reboot* to restart the system at the specified days and times;
        i. Choose one or more *Weekdays* for the system restart;

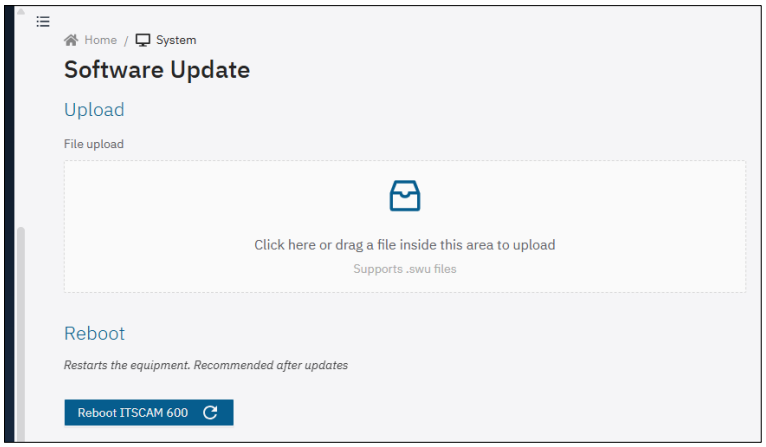
ii. Set a time for rebooting the system;

b. Select *Enable periodic restart* to restart the system whenever it is on for longer than the specified interval;
   i. Set a range between the available options.



4.6. Update

1) *Upload the file* automatically when connected to *SoftwareUpdate* and the installation is carried out automatically when the file is valid;
2) Click on *Restart ITSCAM600* (or ITSCAM450) when the update process is complete, so that the new version goes live;



3) Check the version of the installed file name by going to the *Current Status* > *Versions* > *Firmware* screen.

5. API REST

The capture devices have a API REST to access the images and settings of the equipment. The API is documented in *OpenAPI 3.0* format and the latest version is available in the devices' own web interface via the *API Documentation* option in the left-hand menu or directly via the endpoint http://192.168.254.254/protected/itscam.yaml. The itscam.yaml file can be imported by API testing tools such as Postman and Insomnia.

6. COUGAR Communication Protocol (Socket)

The Cougar protocol is an API for integrating capture devices, based on TCP socket connections. The main functionalities made available through this API are the control and configuration of the device and the receipt of images and metadata from vehicle ticket records.

The protocol was designed with the following premises in mind:

- Modular Implementation:
  - All messages share a common header, making the low-level parser easy to implement;
  - Data formatted mainly in JSON format, without the need to fill in all the fields of a given configuration;
  - By using JSON, metadata and extra functionality can be implemented in the same command without interfering with the way it currently works;
- Asynchronous messages:
  - Events/metadata can be sent by the device without interfering with configuration communication;
  - Facilitates the creation of GUIs or other services with a high rate of simultaneous events;
- Connection customization:
  - Information is only sent when requested, reducing bandwidth consumption;
  - Image metadata can be sent separately, making it possible to receive only metadata, only the image or both;
  - Complete metadata is sent during JPEG processing, improving event timing and bandwidth usage;
  - The connection can use other types of binary JSON to reduce the amount of data transmitted;
  - Connections can request a password, making attacks more difficult.

The breakdown of the protocol following this manual presents the basic structure of the protocol, documents the commands, their arguments and how they work, describes the client APIs and presents a set of general code and usage recommendations to make the most of the protocol and APIs.

6.1. Connection and Messages

Cougar is implemented using TCP/60000 port. By default, the equipment (which will be called the server) will not send any data (except for the server shutdown indication) until it is configured to do so, or in response to a request.

Any data sent in any direction will always be encapsulated in a message. Every message contains a header and a body, if there is one. All data is formatted with the most significant byte (MSB) first (also called Network Byte Order). The message structure is:

Cabeçalho													Corpo						
Byte Inicial	Tamanho (32bits)				Operação (16bits)		ID (32bits)				CRC (16bits)		Corpo				CRC corpo (16bits)	
(102) 0x66	MSB			LSB	MSB	LSB	MSB			LSB	MSB	LSB	0	1	...	Tamanho-1	MSB	LSB

- Initial byte:
  - 8 bits;
  - Fixed value of 102 (0x66);
- Body size:
  - 32 bits with no signal;
  - Contains the size of the message body (without CRC);
- Operation:
  - 16 bits with no signal;
  - Describes operation performed or type of message in the body;
- ID:
  - 32 bits with no signal;
  - "Unique" transaction identifier (possible reuse of IDs with count scrolling);
    - Responses to requests have the same ID as the request.
  - Incremental value, with an increment of 2;
  - The client must start counting at 0. Server starts counting at 1;
- CRC:
  - 16 bits with no signal;
  - XMODEM format (polynomial: 0x1021, initial value: 0x0000, residue: 0x0000);
    - See CRC16 XMODEM calculation example;
  - Calculation for sending done from the initial byte to the ID (11 bytes);
  - Enables checking by running the algorithm only once on the entire header (13 bytes);
    - The CRC of the header will always return 0;
- Body (optional):
  - Size described in bytes in the header;
- Body CRC (when body exists):
  - 16 bits with no signal;
  - XMODEM format (polynomial: 0x1021, initial value: 0x0000, residue: 0x0000);
  - Calculation made only for the body.

6.2. General definitions

The definition of operations is related to the characteristics of the implementation for the ITSCAM 600, ITSCAM 600 FHD, ITSCAM 450 and ITSCAM 450+ devices:

- ITSCAM 600 and ITSCAM 600 FHD have 4 "dry I/Os" (usually called GPIOs), which can be set to:
  - Input of image capture signals (Trigger);
  - Output for activating equipment (gates, signaling, etc.);
  - Flash drive output;

- ITSCAM 450 and ITSCAM 450+ have 4 IOs, 2 inputs and 2 outputs, which cannot be configured;
- ITSCAM 600 and ITSCAM 600 FHD have 2 serial interfaces that can be set up as RS-232 or RS-485 (usually RS-232 on serial 1 and RS-485 on serial 2);
- ITSCAM 450 and ITSCAM 450+ have 2 RS-232 serial interfaces;
- Captured images can be separated into 3 categories:
    - Preview:
        - When nothing else is configured, all images are Preview;
        - Main source of video streams;
        - No flash trigger;
        - No OCR processing or vehicle detection;
        - Used for motion detection and image brightness adjustment;
    - *Snapshot*:
    - Request made by the client (Cougar, WEB or Pumatronix Protocol);
    - Multiple Exposures: Flash, shutter and gain can be customized in the request;
    - Processing pipeline where OCR and other analyses are carried out;
    - The image is never filtered (there is always a response to the client);
    - *Trigger*:
        - Request made automatically (via rising/falling edge of signal, motion detection, etc.);
        - Multiple Exposures: Flash, shutter and gain customized in advance via REST/WEB or Cougar;
        - Processing pipeline where OCR and other analyses are carried out;
        - Image can be filtered if configured not to forward images without vehicles;
- The equipment can capture between 1 and 8 exposures (ITSCAM 600 and ITSCAM 600FHD) or between 1 and 4 exposures (ITSCAM 450 and ITSCAM 450+) from a single event, each with its own shutter and gain (which can be fixed by the user or depend on the current value);
- ITSCAM 600 and ITSCAM 600 FHD can drive up to 4 flashes independently and with power control;
- ITSCAM 450 and ITSCAM 450+ can trigger up to 2 flashes independently.

6.3. Operations

Type	Name	Value	Comments
General	NACK	1 (0x0001)	Message failure / keep-alive
Events	SHUTDOWN	256 (0x0100)	Equipment/connection disconnecting
	EVT_TRIGGER	257 (0x0101)	Trigger metadata
	JPEG_TRIGGER	258 (0x0102)	Trigger image
	EVT_SNAPSHOT	259 (0x0103)	Snapshot metadata
	JPEG_SNAPSHOT	260 (0x0104)	Snapshot image
	EVT_PREVIEW	261 (0x0105)	Preview metadata
	JPEG_PREVIEW	262 (0x0106)	Preview image
	EVT_PIPE_START	263 (0x0107)	Pipeline image input
	EVT_GPIO	264 (0x0108)	Changing GPIO inputs
	EVT_SERIAL	265 (0x0109)	Data received on serial

Type	Name	Value	Comments
	IMGPKG_TRIGGER	266 (0x010A)	Trigger Image with Metadata
	IMGPKG_SNAPSHOT	267 (0x010B)	Snapshot Image with Metadata
Requests	SET_OPT_STR	512 (0x0200)	Section setup
	SET_CALLBACKS	513 (0x0201)	Callback setup (events)
	SET_JPEG_CFGS	514 (0x0202)	JPEG conversion settings
	TRIGGER_SNAPSHOT	515 (0x0203)	Request Snapshot
	GET_LASTFRAME	516 (0x0204)	Request last frame (Preview)
	AUTHENTICATE	517 (0x0205)	Client authentication
	SET_SERIAL_CFGS	518 (0x0206)	Serial interfaces setting
	SEND_SERIAL_DATA	519 (0x0207)	Sending data via serial interfaces
	SET_EQUIP_CFGS	520 (0x0208)	General equipment settings
	CMD_REBOOT	521 (0x0209)	Requests a device restart

The body of all messages has one of the following formats:

- None (empty body [0 bytes]):
  - Used for NACK as a check-alive (to check more quickly if the connection to the server has dropped);
- Binary: Body contains only data in binary format (i.e.: JPEG image);
  - Used for GET_LASTFRAME;
- JSON: Body contains data only in JSON format or one of the binary variants, set using SET_OPT_STR;
  - Used for all other requests and events without a JPEG image (i.e: EVT_SNAPSHOT);
- Mixed: Body contains, in order:
  - Metadata size in 32 bits (4 bytes);
    - MSB first, as well as for the header;
  - Metadata in JSON format (or variant);
  - Data in binary format, occupying the rest of the space specified by the header;
  - Used in image events (i.e.: JPEG_SNAPSHOT).

Descriptions of JSON fields come with the data type in square brackets, along with the default value or an example. For example:

- "field" *[string, "value"]*: Field explanation.

For most fields, the response contains all the fields that can be set for a given request. Therefore, to read the current settings, send an empty request.

1) NACK: NACK is sent by the server in response to a command when:

- Response to the NACK command (sent by the client): As the TCP connection often assumes that the server is still operational, using a check-alive can check whether the server has shut down more quickly;
- Invalid operation or not implemented by the server;
- Unauthenticated client (if the option is enabled);
- The required operand is not present or is incorrectly formatted;

- Internal failure;

A NACK is not sent when:

- Message header is malformed, with invalid header or body CRC;
- Non-critical operation fails or is incorrectly formatted:
    - i.e.: when trying to set the "trigger" field in the SET_CALLBACKS command to 1, the operation fails because it expects a boolean value (true or false) but does not generate errors. It is necessary to check the command response to determine whether the operation was successful.

The body of the response is a JSON with a "reason" field that describes the error encountered for debugging purposes, unless it is a response to the NACK command itself (in which case the body is empty).

2) SHUTDOWN: Event sent when the server is shutting down (for example, if the device is rebooting). Sent without arguments and does not need to be set to be sent.
3) SET_OPT_STR: SET_OPT_STR sets options related to the client section/connection. For security reasons, this is the only setting that only works with the plain JSON format (e.g: {"string":"value"}). The settings for this option are:

- "json" *[string, "plain"]*: how all other messages are sent. That could be:
    - "plain": Common JSON (ascii string);
    - "bson": Binary JSON (BSON);
    - "cbor": Concise Binary Object Representation (CBOR);
    - "messagepack": MessagePack;
    - "ubjson": Universal Binary JSON;
- "respondCfgPath" *[bool, false]*: When set, the SET_EQUIP_CFGS operation returns the path that was sent to the request (more details in the operation);
- "timeoutMs" [int, 1296000000]: Timeout value in milliseconds (default equals 15 days), values accepted between 1000 and 2147483647 (2^31-1). The cougar server will disconnect/close the socket if it doesn't receive any messages from the client in the meantime. To maintain connectivity, a command must be sent to the device at a shorter interval than the defined timeout. To do this, you can use the empty NACK command as a keepalive signal. (From firmware version 1.7.2/1.8.0).

4) AUTHENTICATE: If the server is set up as such, the client will need to send a password for authentication. This option is highly recommended for any operating system, as Cougar has access to most of the device's settings. The parameter used in the request (which can be omitted if the status is only to be checked) is:

- "pass" *[string]*: Password for client access.

This command does not return NACK. The answer will always contain:

- "auth" *[bool]*: True if access to other messages is enabled;
- "msg" *[string]*: Message related to the status of the operation. It could indicate, for example:
    - If authentication is not required;
    - If it's already authenticated;
    - If authentication was successful;
    - If the field is poorly formed;
    - If the password is incorrect;

The only commands that work without requiring authentication are:

- NACK (check-alive);
- SET_OPT_STR (to configure JSON mode);

- AUTHENTICATE.

5) SET_CALLBACKS: SET_CALLBACKS sets which events are sent to the client. The fields are:

- "pipeline" *[bool, false]*: Enables EVT_PIPE_START events;
- "trigger" *[bool, false]*: Enables EVT_TRIGGER events;
- "snapshot" *[bool, false]*: Enables EVT_SNAPSHOT events;
- "preview" *[bool, false]*: Enables EVT_PREVIEW events;
- "gpio" *[bool, false]*: Enables EVT_GPIO events;
- "triggerjpeg" *[bool, false]*: Enables JPEG_TRIGGER events;
- "snapshotjpeg" *[bool, false]*: Enables JPEG_SNAPSHOT events;
- "previewjpeg" *[bool, false]*: Enables JPEG_PREVIEW events;
- "triggerimgpkg" *[bool, false]*: Enables IMGPKG_TRIGGER events;
- "snapshotimgpkg" *[bool, false]*: Enables IMGPKG_SNAPSHOT events;
- "serial1" *[bool, false]*: Enables EVT_SERIAL events for serial 1;
- "serial2" *[bool, false]*: Enables EVT_SERIAL events for serial 2;

6) SET_JPEG_CFGS: SET_JPEG_CFGS sets the minimum quality and timing for generating JPEGs. The default quality is configurable via the REST/WEB interface. The fields are:

- "trigger":
  - "quality" *[int, configurável]*: Quality of images generated from Trigger;
- "snapshot":
  - "quality" *[int, configurável]*: Quality of images generated from Snapshot;
- "preview":
  - "quality" *[int, configurável]*: Quality of images generated from Preview;
  - "mindt" *[int, 100]*: Minimum time (in milliseconds) between Preview images, to reduce the maximum frame rate.

7) EVT_PIPE_START: Event sent when a frame enters the photo pipeline (Trigger and Snapshot). Used to facilitate the timing of photo capture with an external server. Only the simplest metadata is sent:

- "framecount" *[uint64]*: Unique image value, implemented using a counter that increments with all images captured (Snapshots, Triggers and/or Previews), reset to zero at server startup;
- "rid" *[uint64]*: Snapshot sync value. Implemented so that the client can correlate the request made (which returns the same value) with the image captured;
- "multexp":
  - "len" *[int]*: Number of exposures in the multiple exposure group;
  - "pos" *[int]*: Position of the image in multiple exposures, starting with 0.

8) EVT_TRIGGER, EVT_SNAPSHOT, and EVT_PREVIEW: All these events are sent when the images have finished being processed internally and are delivered to the Server. The moment these events are sent, the JPEG conversion of the image also begins, if enabled. For Trigger and Snapshot images it is possible to correlate EVT_PIPE_START events using "framecount". The metadata for all of them is similar, with the exception of the lack of "rid", "jidosha" and "classifier" data for Preview images. The fields are:

- "framecount" *[uint64]*: Unique image value, implemented using a counter that increments with all images captured (Snapshots, Triggers and/or Previews), reset to zero at server startup;
- "rid" *[uint64]*: Snapshot sync value. Implemented so that the client can correlate the request made (which returns the same value) with the image captured;
- "ogSize":
  - "w" *[int]*: Original image width, in pixels;

  - "h" *[int]*: Original image height, in pixels;
- "size":
  - "w" *[int]*: Image width. Different from the original if the image has been cropped, in pixels;
  - "h" *[int]*: Image height. Different from the original if the image has been cropped, in pixels;
- "multExp":
  - "len" *[int]*: Number of exposures in the multiple exposure group;
  - "pos" *[int]*: Position of the image in multiple exposures, starting with 0;
- "shutter" *[int]*: Shutter exposure in microseconds;
- "gain" *[int]*: Analog gain of exposure in millibels (100x decibels);
- "stats":
  - "level" *[int]*: Image level, with a value from 0 to 1000;
  - "meanr" *[int]*: Average value of the color Red, with a value from 0 to 255;
  - "meang" *[int]*: Average value of the color Green, with a value from 0 to 255;
  - "meanb" *[int]*: Average value of the color Blue, with a value from 0 to 255;
  - "stddev" *[int]*: Standard deviation squared, with a value from 0 to 65535;
- "time":
  - "setup" *[uint64]*: Monotonic exposure scheduling time, in microseconds;
  - "exp" *[uint64]*: Monotonic exposure time, in microseconds;
  - "dma" *[uint64]*: Monotonic entry time of the exposure into the pipeline, in microseconds;
  - "now" *[uint64]*: Internal monotonic time of this message, in microseconds;
- "date":
  - "year" *[int]*: Year the image was captured;
  - "month" *[int]*: Month the image was captured;
  - "day" *[int]*: Day the image was captured;
  - "hour" *[int]*: Hour the image was captured;
  - "min" *[int]*: Minute the image was captured;
  - "sec" *[int]*: Second the image was captured;
  - "msec" *[int]*: Millisecond the image was captured (taken at the same moment as the "time.dma" field);
- "crop":
  - "x" *[int]*: Horizontal position of the start of the image crop, in pixels;
  - "y" *[int]*: Vertical position of the start of the image crop, in pixels;
- "jidosha": List / Vector:
  - "plate" [string]: Identified plate;
  - "probs" [float list]: Probabilities of each character identified. Same size as the identified plate. Values from 0.0 to 1.0;
  - "pos":
    - "x" [int]: Horizontal position of the plate, in pixels;
    - "y" [int]: Vertical position of the plate, in pixels;
    - "w" [int]: Plate width, in pixels;
    - "h" [int]: Plate height, in pixels;
  - "color" [int]: Plate color, 0 for white and 1 for red;
  - "moto" [int]: Value 1 for when the identified license plate has the Motorcycle format, 0 otherwise;
  - "country" [int]: OCR country code, using ISO 3166-1;
- "classifier": List / Vector:

- "type" *[int]*: Type of vehicle identified, based on the Classifier interface:
  - 0: Unknown;
  - 1: Car;
  - 2: Motorcycle;
  - 3: Truck;
  - 4: Bus;
- "prob" *[float]*: Identification probability, from 0.0 to 1.0
- "pos":
  - "x" *[int]*: Horizontal position of the vehicle, in pixels;
  - "y" *[int]*: Vertical position of the vehicle, in pixels;
  - "w" *[int]*: Vehicle width, in pixels;
  - "h" *[int]*: Vehicle height, in pixels.

9) JPEG_TRIGGER, JPEG_SNAPSHOT, JPEG_PREVIEW, IMGPKG_TRIGGER and IMGPKG_SNAPSHOT: All of these events load the converted JPEG image along with some metadata relating to the capture, so that the events can be correlated (using the Mixed formatting described above). The quality of the conversion is set via the "quality" fields of SET_JPEG_CFGS. For the JPEG_PREVIEW event, only the "framecount" and "quality" fields are filled in. For the others, the fields are:

- "framecount" *[uint64]*: Unique image value, implemented using a counter that increments with all images captured (Snapshots, Triggers and/or Previews), reset to zero at server startup;
- "quality" *[int]*: JPEG conversion quality, in percent;
- "rid" *[uint64]*: Snapshot sync value. Implemented so that the client can correlate the request made (which returns the same value) with the image captured;
- "multExp":
  - "len" *[int]*: Number of exposures in the multiple exposure group;
  - "pos" *[int]*: Position of the image in multiple exposures, starting with 0.

10) TRIGGER_SNAPSHOT: This operation queues a Snapshot request. By default, when called without any parameters, it uses the default parameters configured for the Trigger in the WEB/REST interface. Multiple exposure settings should be formatted as follows:

- "multexp": List / Vector (between 1 and 8 items):
  - "shutter" (only one of the options, 100% of the current shutter if not populated):
    - "percent" *[int, 100]* : Percentage of the current shutter;
    - "value" *[int]* : Shutter in microseconds;
  - "gain" (only one of the options, 100% of the current gain if not populated):
    - "percent" *[int, 100]* : Percentage of the current gain;
    - "value" *[float]* : Gain in decibels;
  - "flash":
    - "1" *[int, 0]*: Flash percentage at output 1;
    - "2" *[int, 0]*: Flash percentage at output 2;
    - "3" *[int, 0]*: Flash percentage at output 3;
    - "4" *[int, 0]*: Flash percentage at output 4.

The "shutter" and "gain" fields expect only one of the available options. If both are filled in, the percentage will always be used. In addition, in order for the flash to work properly, the outputs used as flash must be pre-configured.

The request response only contains the "rid" (64-bit integer field), which is a unique identifier value for the request, serving to correlate requests with the resulting images/processing/metadata.

It is also possible to pass the following parameters in the request, inside a "stringMap" object, in the form of string pairs, to customize the requests:

- "stringMap":
  - "Cenario" *[string]*: It is possible to pass the value "1" or "2" to use one of the scenarios configured on the device (in the *Image>Miscellaneous* menu, *Scenarios* tab);
  - "User_*" *[string]*: Any text string to replace the tag with the same name in the banner configured for the selected scenario. The name of this field can be chosen by the user (such as User_Speed, User_Id), accepting any alphanumeric string in place of the *;
  - "UserCrop" *[string]*: String in the format "x1,y1,x2,y2", where x1 and y1 are the position in pixels from the top left corner of the image, and x2 and y2 are the position in pixels from the bottom right corner of a customized cropped image;
  - "TextOverlay" *[string]*: Desired custom text for the image banner.

11) GET_LASTFRAME: This operation uses the last Preview exposure to generate a JPEG. The only input parameter is:

- "quality" [*int, 80]* : JPEG conversion quality. The response to this command fills the body entirely with the converted image or returns NACK if there is a problem. For continuous use (video or preview) we recommend using the JPEG_PREVIEW event to generate the images.

12) EVT_GPIO: This event is sent when there is a change in the status of the device's "dry" inputs (GPIO). The data sent is:

- "framecount" *[uint64]*: Frame in which the event occurred, useful for correlating with exposures;
- "rising" *[int]*: Combination of all inputs where the logic level ranged from 0 to 1;
- "falling" *[int]*: Combination of all inputs where the logic level ranged from 1 to 0;
- "state" *[int]*: Final state of inputs.

To shorten the messages, the data from all the inputs is placed in the same integer variable, setting the bit with the corresponding offset. For example, if input 3 changes state from 0 to 1 and input 1 is high, the corresponding event (in "plain" mode) would be:

```
{"framecount":1234, "rising":4, "falling":0, "state":5}
```

Therefore, setting the bit corresponding to input 3 would be equivalent to `1<<(3-1) => 4` ("<<" being the shift-left operation) and the combination of bits 1 and 3 would be `(1<<(3-1))|(1<<(1-1)) => 5` ("|" being the bit-by-bit OR operation).

13) SET_SERIAL_CFGS: SET_SERIAL_CFGS sets the serial interfaces. They can be used by more than one client and reconfiguration using different parameters can lead to data loss. It is recommended (especially for RS485 connections) that all devices on the same bus use the same settings. The available settings are:

- "serial1":
  - "baud" *[int, 115200]*: Interface speed in bits per second (baud);
  - "bits" *[int, 8]*: Number of bits in each frame. Valid values are between 5 and 8;
  - "stop" *[int, 1]*: Number of stop-bits. Valid values are 1 or 2;
  - "parity" *[string, "n"]*: Parity of each frame. Valid values are:
    - "n": No generation or checking;
    - "o": Odd parity;
    - "e": Even parity.
- "serial2":
  - Same parameters as "serial1".

14) EVT_SERIAL: Event generated when data is received via the serial interface. As there is more than one interface, the message format is mixed. In the metadata, it is sent:

- "pipe" *[string]*: Data source ("serial1" or "serial2"). In the rest of the message (binary format), the received data is sent. Between 1 and 2048 bytes of raw data can be sent in each message. Due to internal timings and other factors, messages being received may be sent in fragments smaller than the maximum size.

15) SEND_SERIAL_DATA: This operation sends data to the specified interface. The required fields are:

- "pipe" *[string]*: Target interface ("serial1" or "serial2");
- "ascii" *[string]*: Data in ASCII format (no special characters/bytes) or;
- "base64" *[string]*: Data in base64 format or;
- "hex" *[string]*: Data in hexadecimal format (2 characters per byte, 0-9;A-F).

Only one of the data fields is used (using primarily "ascii", then "base64" and then "hex" if multiple are sent). It is recommended to send only the most convenient format. If everything goes as expected, the command will return:

- "len" *[int]*: Number of frames (usually bytes) sent.

16) SET_EQUIP_CFGS: SET_EQUIP_CFGS is the main interface for controlling equipment settings on the Cougar. To approximate the formatting of the REST interface, it controls the image capture settings, making it equivalent to REST endpoints:

- /camera/misc/readonly/volatile → "equip.volatile";
- /camera/misc/readonly/constants → "equip.miscRO";
- /camera/misc → "equip.misc";
- /camera/autofocus → "equip.autofocus";
- /camera/profiles/:id (except creating and deleting profiles) → "equip.profiles";
- /camera/profiles → "equip.currProfile";
- /camera/transitioner → "equip.transitioner";
- /camera/ios → "equip.io".

It also controls the Services endpoints:

- /camera/services/stream → "h264" and "mjpeg";
- /camera/services/ocr → "ocr";
- /camera/services/classifier → "classifier";
- /camera/services/analytics → "analytics";
- /camera/services/ftp → "ftp";
- /camera/services/itscampro → "itscampro";
- /camera/services/licenses → "license".

The (optional) input parameters for this command are:

- "path" *[string,""]*: Root path to consider for the data;
- "data" *[object/value, {}]*: Data to be written to the settings.

All the settings mentioned above are available in a complete settings tree. Therefore, like all the other Cougar settings, it is possible to configure the elements sparsely, configuring only what it is need. This means that, from a writing point of view, the following options are equivalent:

- {"path":"analytics.voting.roi1.x0", "data":123}
- {"path":"", "data":{"analytics":{"voting":{"roi1":{"x0":123}}}}}

- {"path":"analytics.voting", "data":{"roi1":{"x0":123}}}

Due to the sheer volume of configurations, when using an empty path in a conventional case, around 800 elements are answered. This unnecessarily increases the consumption of processing and network resources and is therefore not recommended for continuous use. For the examples above, the answers are:

- 123;
- all settings (~800 elements);
- only majority voting settings (~25 elements).

Due to the volume of configurations, it is recommended to use the REST interface documentation as a reference for the parameters of this command.

17) CMD_REBOOT: This command forces an immediate reboot of the capture device.

6.4. APIs available

Since the protocol is open, it is possible to implement its own version of the client, especially if the application does not use one of the languages in which they have been implemented, or if the version made available does not meet the requirements of the rest of the application (e.g. python 2 vs. 3, C++ 98 vs. 11 vs. 17 vs. 20). Pumatronix provides clients in C++ and Python, for code reference only (contact technical support to obtain them).

6.5. General Recommendations

Configurations can be changed partially/sparsely, without the need to read, modify and write a fixed structure. This way, even if new fields are added, client behavior remains consistent.

For the SET_EQUIP_CFGS operation, the path closest to the settings to be adjusted should be entered, so that the return from the operation is as small as possible. It is recommended to get all the settings ("path":"") only when necessary (for example: when initializing the client logic).

To implement the protocol independently, it is highly recommended to use a multi-threaded environment to handle message reception and interpretation in a separate thread, to reduce latency between data reception. If necessary, it is possible to use a *threadpool* to make interpretation faster in multi-core environments.

All messages are asynchronous, meaning that during the processing of a command (waiting between sending and receiving) other messages (events, images or even commands from other threads) can be received.

6.6. CRC16 XMODEM calculation example

```
// Example implementation of the CRC16 XMODEM algorithm. This example uses a
// "CRC tab" to precalculate coefficients and optimize performance.
// It is possible to define the tab as a "magic" value constant.
// It uses variables defined in <stdint.h> that can be overridden
// for the system equivalent. They usually are:
// - uint8_t -> unsigned char
// - uint16_t -> unsigned short int
```

```
uint16_t crcXMODEM(const uint8_t *c, int numBytes, uint16_t currCrc = 0) {
    static uint16_t crcTab[256];
    static bool init = 0;
    if(!init){
        init = true;
        for (int i=0; i<256; i++) {
            uint16_t short_c = i<<8;
            uint16_t crc = 0;
            for (int j=0; j<8; j++) {
                if ( ((crc ^ short_c) & 0x8000) != 0 ) {
                    crc <<= 1;
                    crc ^= 0x1021;
                } else {
                    crc <<= 1;
                }
                short_c <<= 1;
            }
            crcTab[i] = crc;
        }
    }
    for (int i = 0; i < numBytes; i++) {
        currCrc = (currCrc << 8) ^ crcTab[((currCrc >> 8) ^ c[i])];
    }
    return currCrc;
}
```

7. Open Source Pumatronix Communication Protocol (Socket)

Communication with the devices is via the Ethernet interface, using the Pumatronix UDP and TCP/IP communication protocols. The port used to communicate with the external device is **50000**. Therefore, the application developed to communicate with the device must be configured to send commands using this port in the TCP and UDP protocols. There is compatibility and some differences in the application on each device model.

The UDP protocol is only used to identify devices connected to the network, as it allows *broadcast-type* packets to be sent, which are received by all devices. This allows the image capture device to send its ID when it receives this packet.

All other commands use the TCP protocol, which establishes a point-to-point connection between the control device and the image capture device. Receiving and changing settings is secured by transmitting a *CRC* code. However, most of the responses sent by the equipment do not have a *CRC*.

The Pumatronix Protocol supports connections that have been developed based on the Dynamic Library (dll) and the C++ class for Linux. A *Development Kit (SDK)* is available at https://www.pumatronix.com/ with the files needed to develop the application, that can be downloaded from the site at *Customer Area > Technical Support.*

Commands			
VIDEO	REQ_DATA	SETA_TIPO_SHUT	REQ_POSICAO_ZOOM
VIDEO_CONTINUO	REQ_ENTRADA	SETA_SHUT	REQ_POSICAO_FOCO
FOTO	REQ_GAMMA	SETA_SHUT_MAX	
FOTO_INFO_ADIC	REQ_GANHO_VISIVEL	SETA_TIPO_GANHO	
REQ_CONF_IMAGEM	REQ_TEMPO_LIGADO	SETA_GANHO	
REQ_CONF_GER	REQ_MOV_LENTE	SETA_GANHO_MAX	
SETA_DEFAULT	REQ_FOCO_IR	SETA_NIVEL_IMG	
REQ_CONF_REDE	REQ_IO_VIGIA	SETA_ZOOM	
REQ_SITUACAO_DAY_NIGHT	REQ_DIR_IO_VIGIA	SETA_FOCO	
REQ_ROTACAO	SETA_ROTACAO	SETA_POSICAO_ZOOM	
REQ_LENTE_AUTO_IRIS	SETA_LENTE_AUTO_IRIS	SETA_POSICAO_FOCO	
REQ_MODO_DAY_NIGHT	SETA_MODO_DAY_NIGHT	SETA_IO_VIGIA	
REQ_SATURACAO	SETA_SATURACAO	SETA_DIR_IO_VIGIA	
REQ_WHITE_BALANCE	SETA_WHITE_BALANCE	SETA_GAMMA	
REQ_FOTO_COLORIDA	SETA_FOTO_COLORIDA	REQ_WB_ATUAL	
REQ_HORA	SETA_SAIDA	SETA_ZOOM_FOCO	


PUMATRONIX
www.pumatronix.com