PUMATRONIX

PUMATRONIX

# CAPTURE DEVICES

ITSCAM 600, ITSCAM 600 FHD, VTR 600, ITSCAM 450 E 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 |
|---|---|---|
| 11/30/2024 | 1.4.1 | Inclusion of ITSCAM 600+ product;<br>Reordering of chapters according to the web interface;<br>Detailing of the Maintenance and Upgrade menus;<br>Insertion of the Digital Signature validation process;<br>Addition of available REST API fields |
| 02/04/2025 | 1.4.2 | Updates for firmware version 1.7.5; Jpeg comments; Vehicle indicator; Pumatronix servers; |
| 05/23/2025 | 1.4.3 | Updates for firmware version 1.7.6 (SAD-836) |
| 17/10/2025 | 1.5.0 | Description of SDK commands; Cougar Protocol updates; Insertion of Pumatronix Open Source Communication Protocol (Socket) commands |
| 10/15/2025 | 1.5.1 | Updates for firmware version 1.7.7 (SAD-948) |
| 01/09/2025 | 1.5.2 | Update regarding firmware version 1.7.8 (SAD-1015) |

# 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* |

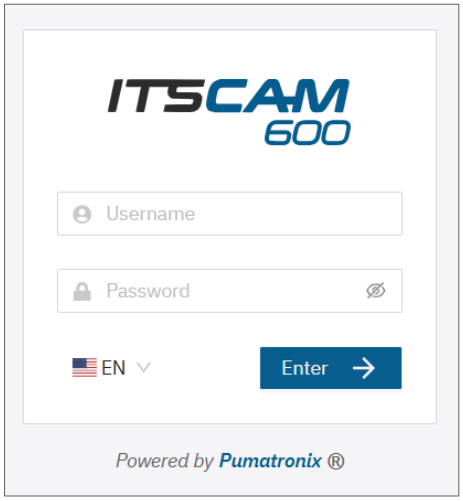



*Figure 1 - Login screen*

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

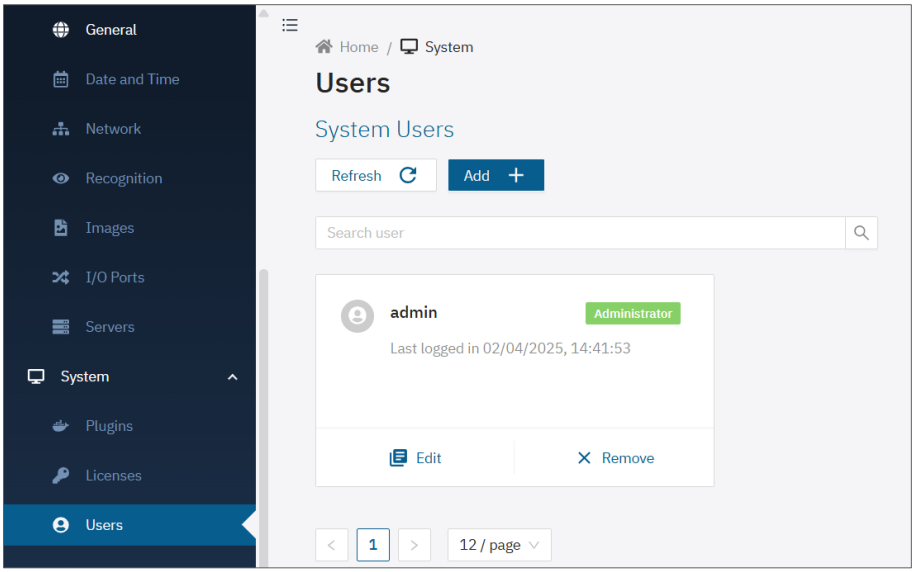
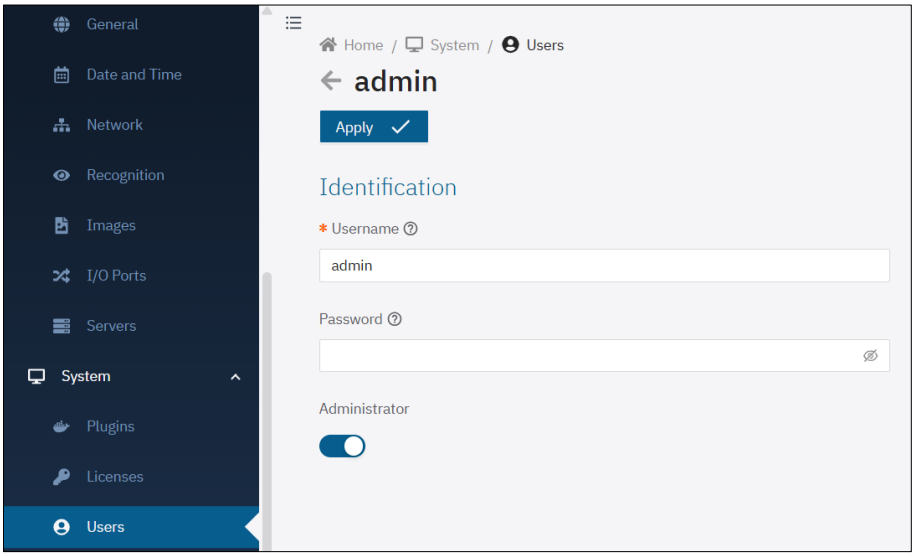



*Figure 2 - Screen displayed when accessing System>Users*

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:

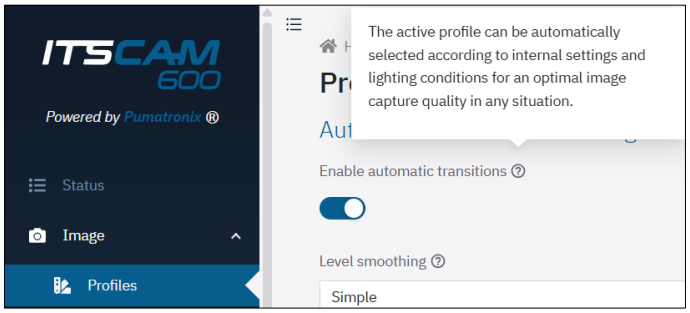



*Figure 3 - Help displayed by positioning the cursor on the icon*

*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*:

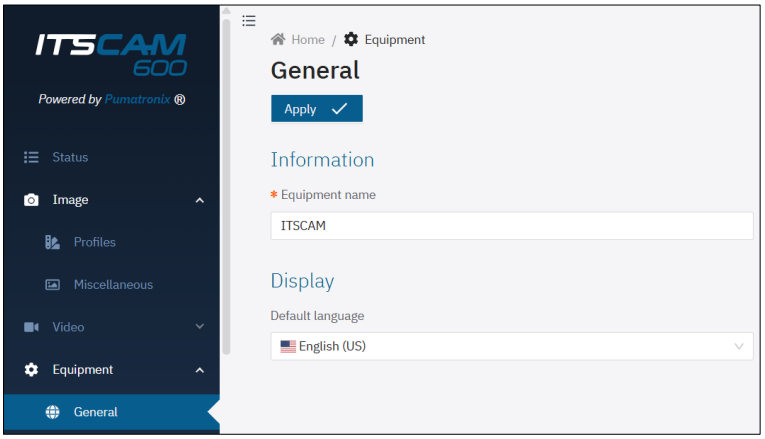



*Figure 5 – Equipment > General menu screen*

## 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:

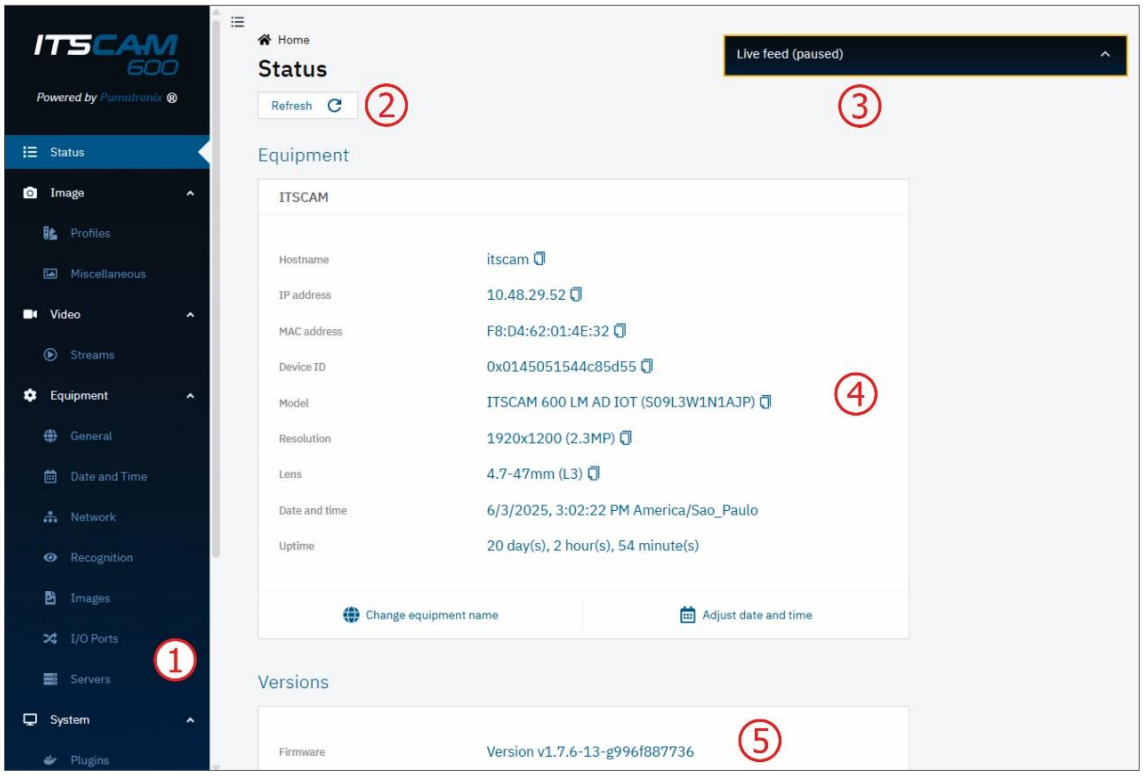



*Figure 6 - Options available on the interface and in the Status window: 1) Menu bar; 2) Action button available for functionality; 3) Pop-up live view window; 4) Equipment data, including detailed Runtime and quick setting buttons, 5) Data on installed firmware versions*

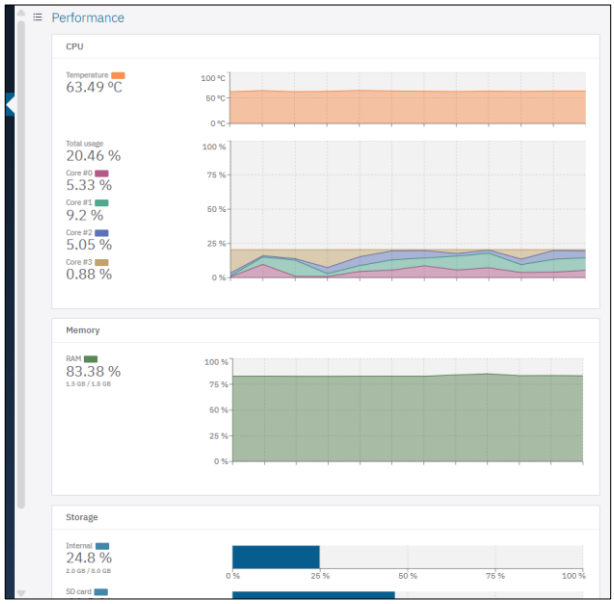



*Figure 7 - Available data on the Current Status of CPU, Memory and Storage Performance*

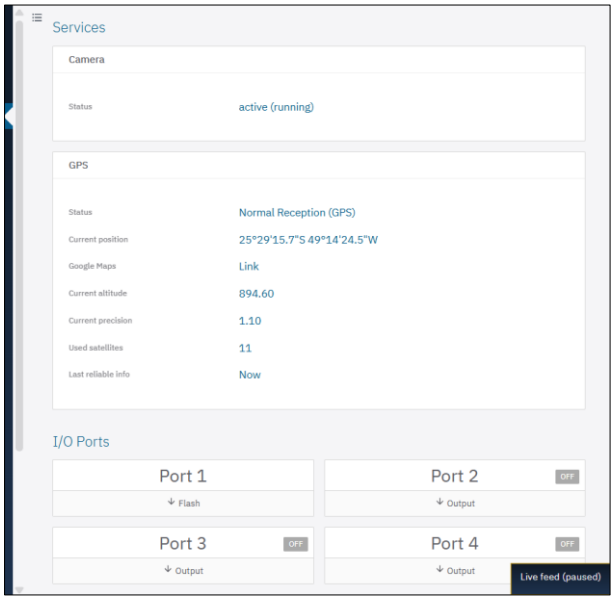



*Figure 8 - Available data on the Current Status of Camera and GPS Services* and port connections*

* 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 – Live View Floating Window: 1) Pause live view; 2) Take photo; 3) Run autofocus; 4) Show/Hide Grid; 5) Video or Capture Mode; 6) Slider Control; 7) Show/Hide Settings; 8) Full screen; 9) Minimize window; 10) Active FPS, Level, Gain, Shutter and Image Profile values; 11) Zoom and Focus adjustments*

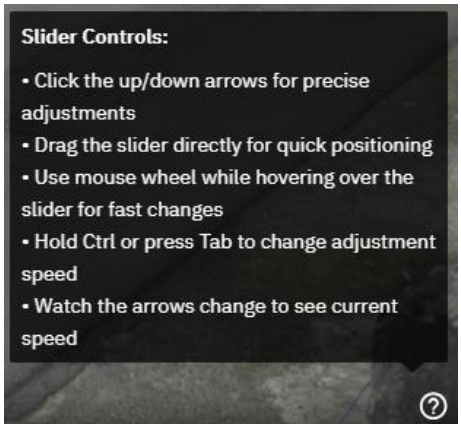
1) Click *Pause live view* (1) to pause the live video and the last captured image remains fixed;
2) Click the *Take Photo* button (2) to download an instant JPEG file generated from the image currently being displayed;
3) Click the button to *Run autofocus* (3);

4) Click the button to show the *Grid* (4):
    a. On the right side of the window, the *Grid* size configuration will be available;


5) Select the Image display *Mode* (5) from the options:
    a. Select *Video Mode* to display the real-time image captured by the device, enabling the option:
        i. Synchronized: when enabled, it is possible to have nighttime video synchronized with the illuminator trigger (the trigger must be enabled). When disabled, smoother video is possible (it does not depend on trigger configuration). This option is automatically enabled when configured in the Video Settings (Streams) tab;


    b. Select Capture Mode to display the plates/vehicles detected in the capture:
        i. Plates: click to hide/show the plates detected in the capture;
        ii. Vehicles: click to hide/show the vehicles detected in the capture;


6) Hover the mouse over the *Slider Control* (6) to view keyboard and mouse shortcut options for better Zoom and Focus configuration.




7) Click the Show *Display Settings option (7)* to access all available options in the floating window:
8) Click the *Full Screen* option (8) to expand the view and occupy the entire interface window;
9) Click the arrow to *Minimize* the window (9):
    a. The viewer can be locked when minimized by clicking the padlock:


10) View, at the top of the floating window (10), the *Level, Gain, and Shutter* values being applied to the device and the active *Image Profile.*
11) Make the *Zoom and Focus* adjustments (11) using the two bars displayed on the left side of the floating window.

# 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:

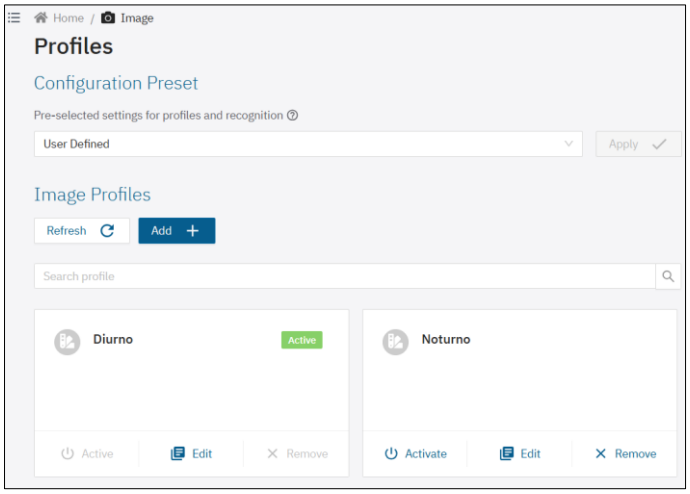



*Figure 10 – Image Menu > Profiles*

1) 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*;
2) 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;

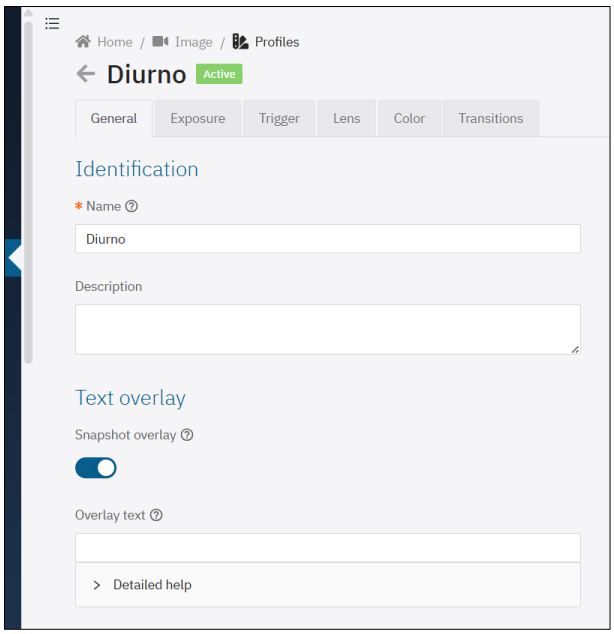
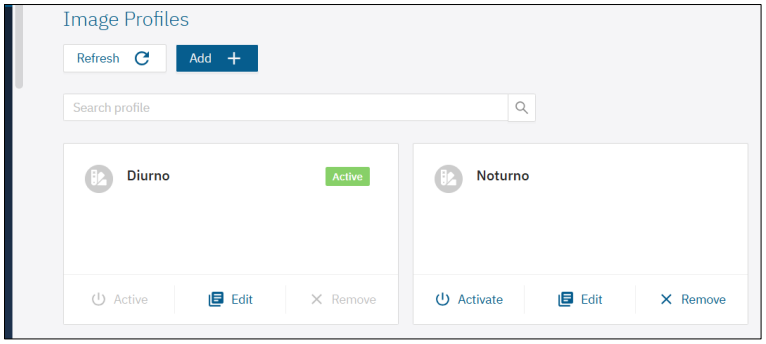



*Figure 11 - Initial screen for editing the Day profile*

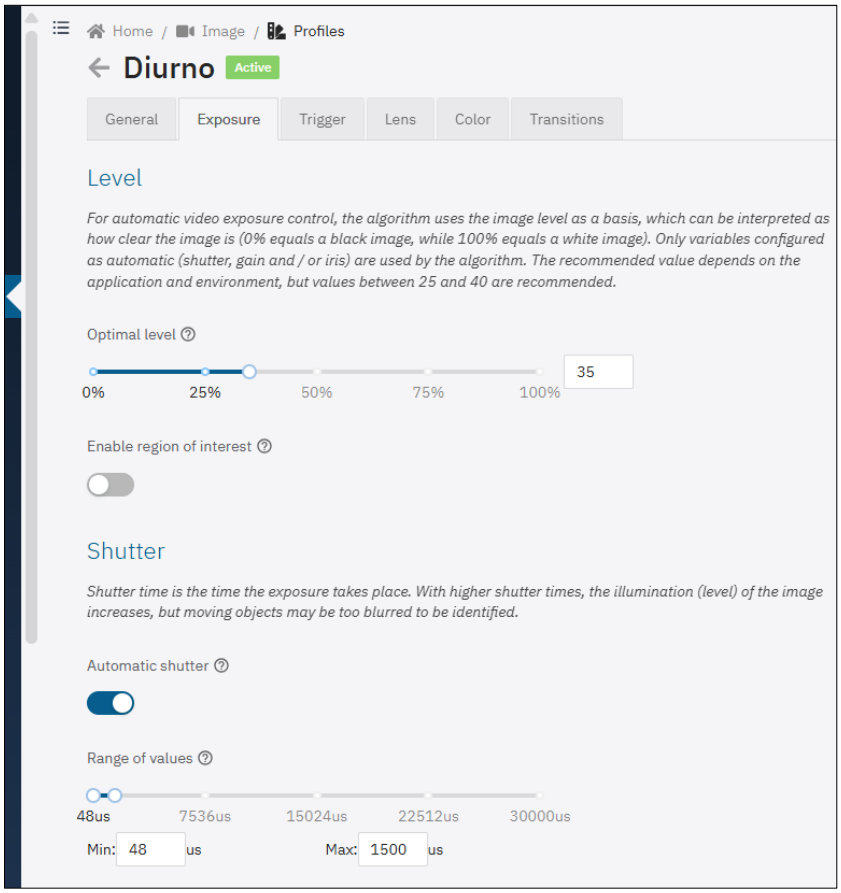
3) Click *Add +* on *Image Profiles* to create new image profiles (it is recommended to configure existing profiles before creating new ones).

## 2.1.1. Exposure

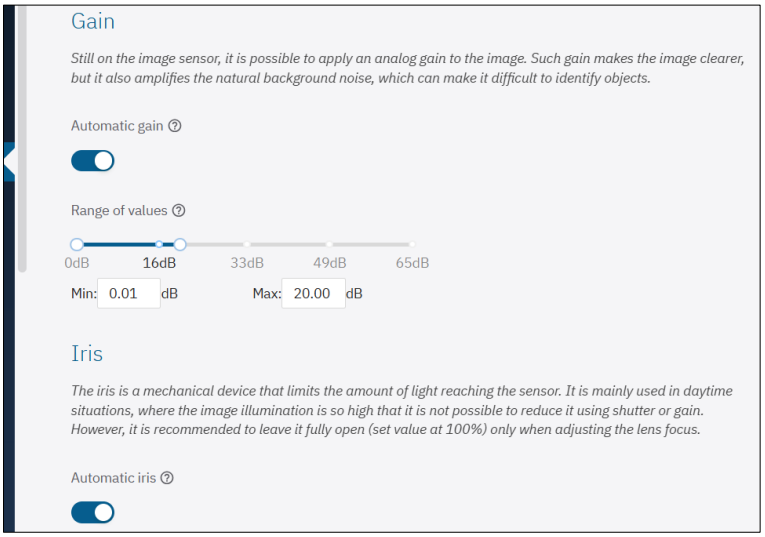
1) Find the profile to set up in the *Image > Profiles* menu and click *Edit*;




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;

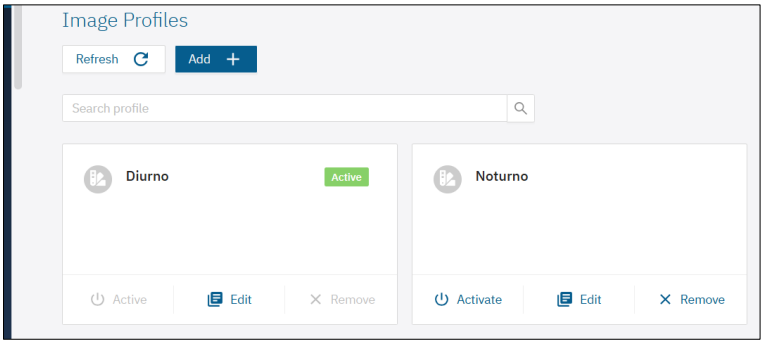


7) Enable *Gain* by clicking on the *Automatic Gain* option, in which the equipment will adjust the *Gain* (within the limits set by the user) to keep the image at the optimal level. When this option is deactivated, the *Gain* to be applied is a fixed value also defined by the user;
8) Select the *Range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

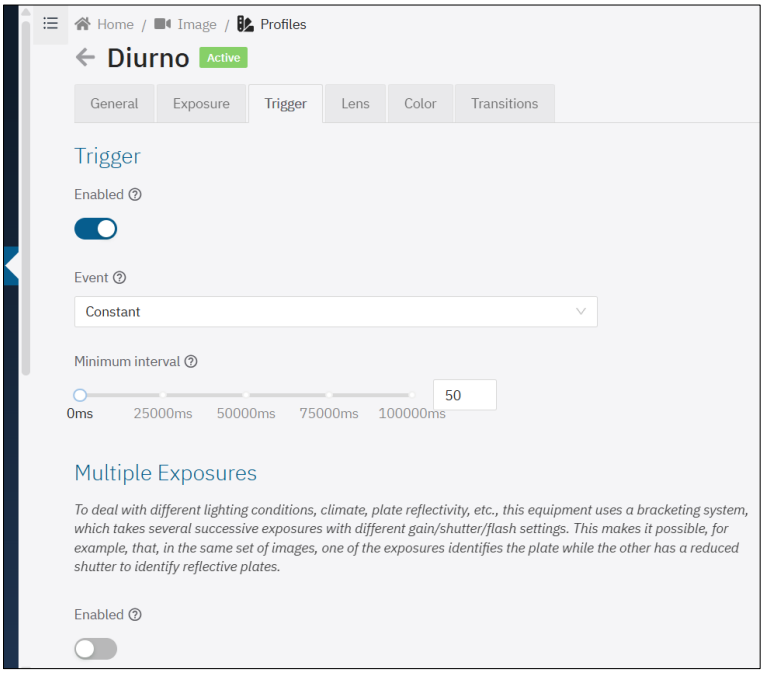



## 2.1.2. Trigger

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



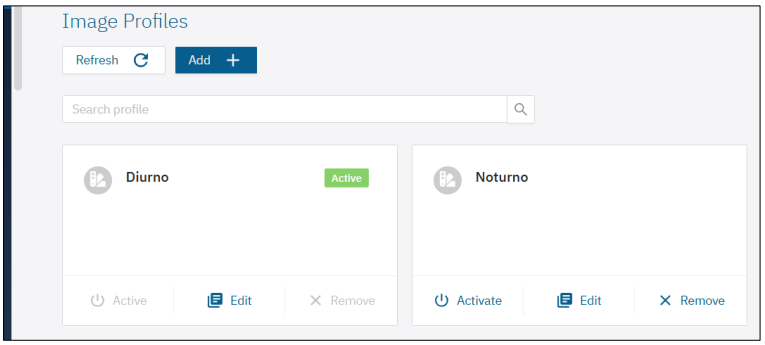

2) Select the *Trigger* tab and on this screen, the settings are automatically saved;




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

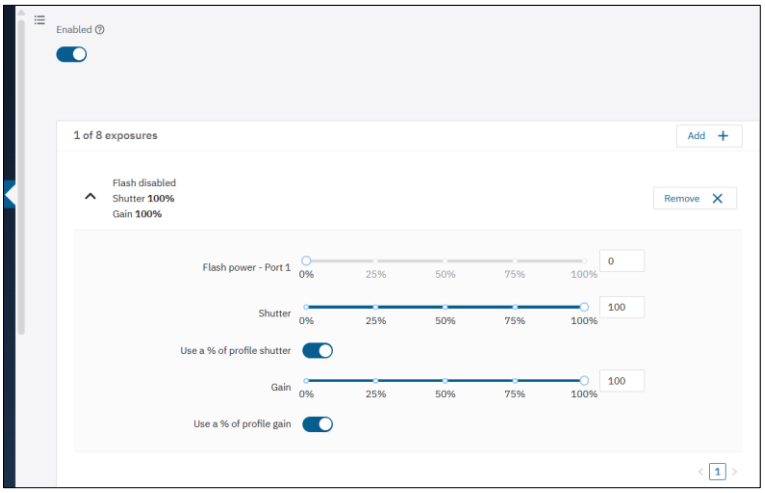
1) Find the profile to set up in the Image > Profiles menu and click *Edit*;




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;



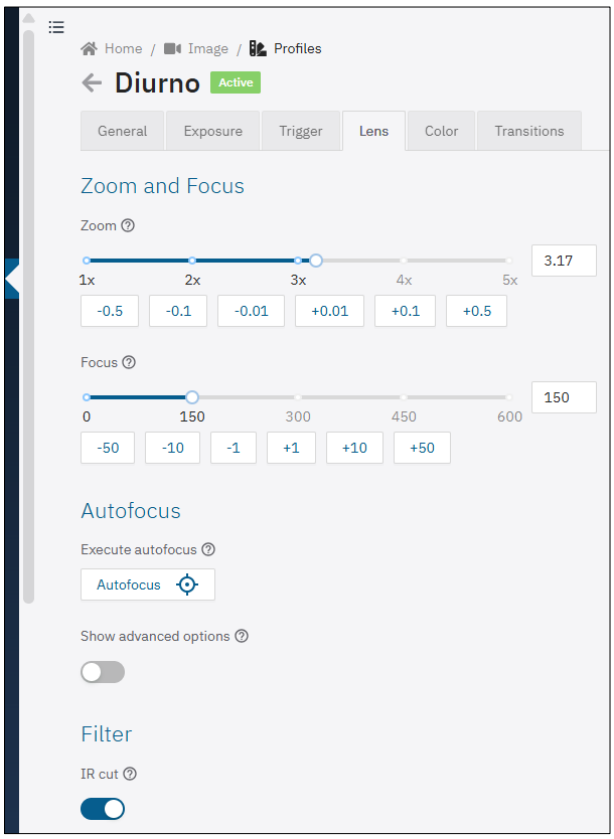

6) Select the *Flash power*, always corresponding to a percentage of the initial shot (this option is available for the Pumatronix ITSLUX range of lighting devices);
7) Select the *Shutter* percentage (exposure time of the image sensor), generating images with a variation in the amount of light captured;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

## 2.1.3. Lens

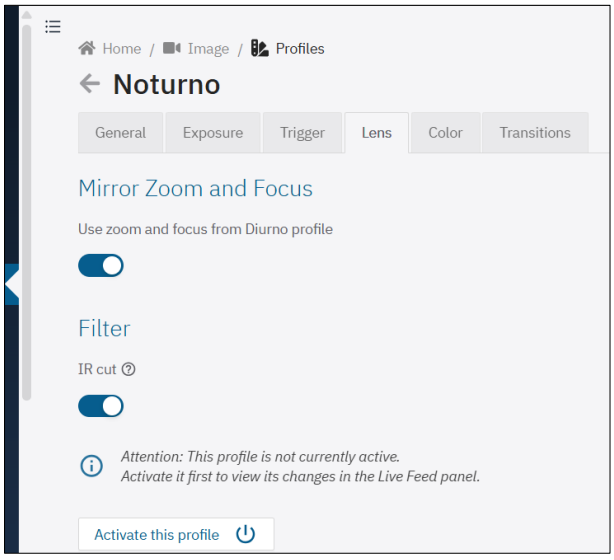
The *Lens* tab contains the *Zoom* and *Focus*, *Autofocus* and *Filter* settings.

**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;

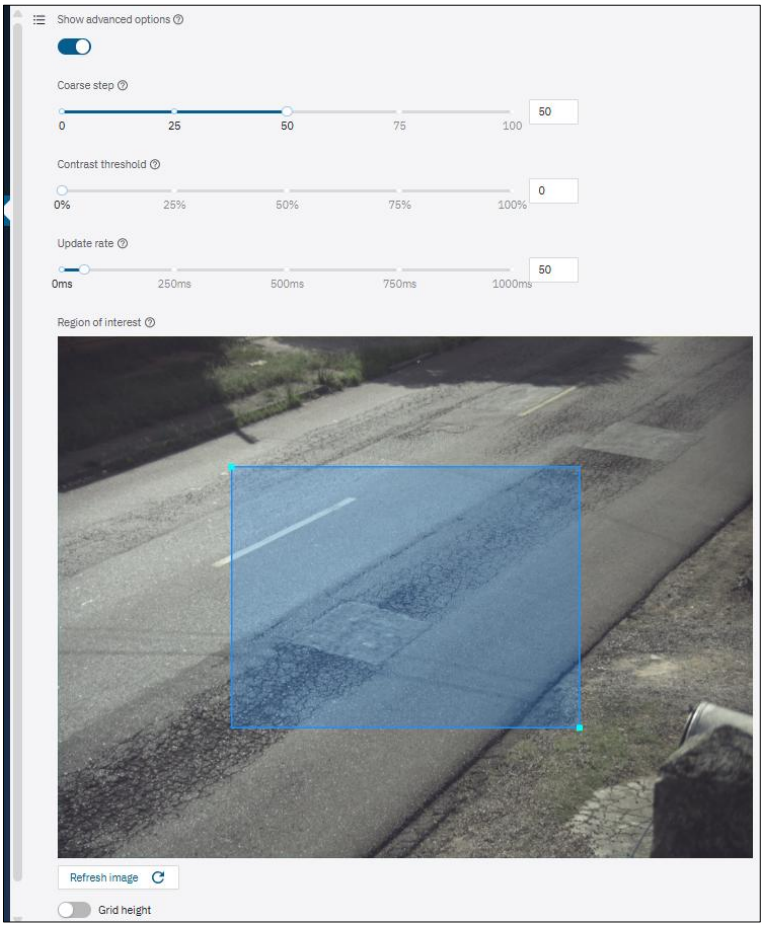



2) Check the *Zoom* and *Focus* settings in real time using the Pop-up Preview window;
3) Select the *Zoom* rate value, gradually increasing the available buttons until finding the ideal value;
4) Select the distance of the *Focus* lens, gradually increasing the available buttons until finding the ideal 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*

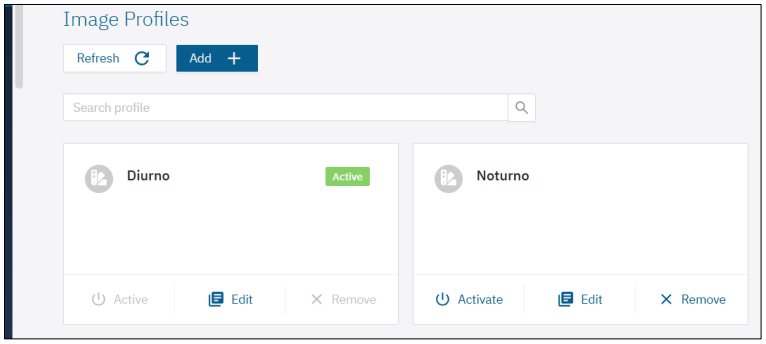
*Figure 13 – Example of a nighttime image with the suggested framing*

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

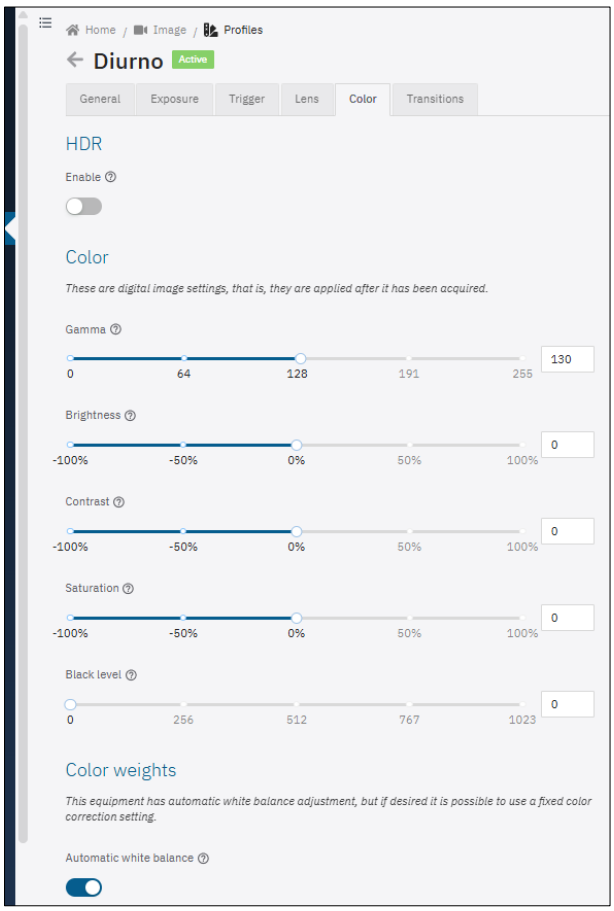
## 2.1.4. Color

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.

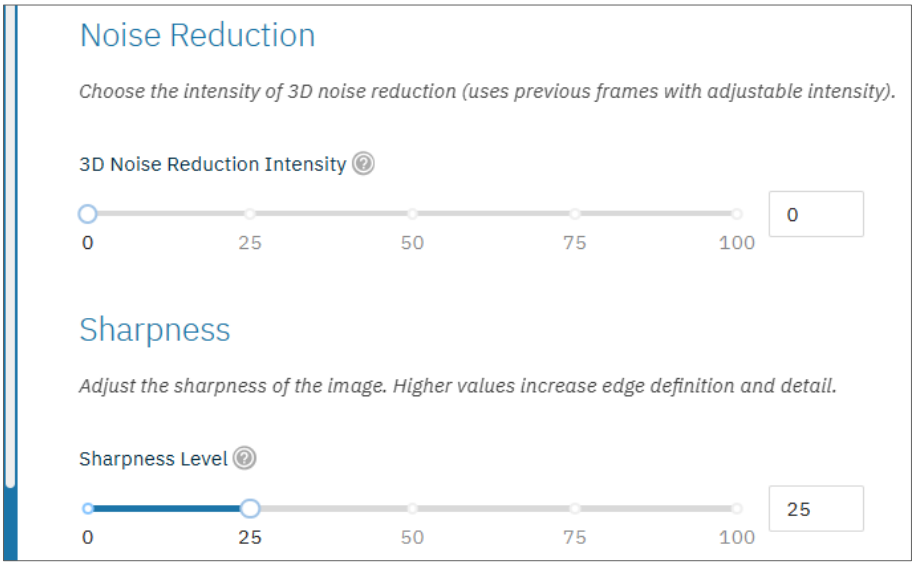



**The *Brightness*, *Contrast*, *Saturation and Black Level* values are factory-set to the ideal values in normal image lighting conditions and adjustments are restricted in cases of images generated with poor quality. Please contact Technical Support for guidance on these color settings.**

On the ITSCAM 450 and ITSCAM 450+ devices, additional *Noise Reduction* and *Sharpness* settings are available under the *Color tab.* These settings are excellent for improving video and image visual quality, especially in low-light conditions.

1) Set the *Noise Reduction* from 0 to 100 to soften electronic noise ("grain" in the image), which is common in dark environments (3D noise reduction may cause trails on moving objects;
2) Set the *Sharpness Level* from 0 to 100 to make the image sharper, enhancing edges and image details. During use, consider the following situations:

- Slight Sharpness enhancement: Causes an enhancement of image edges, which is useful in cases such as recognizing vehicle license plate characters.
- Excessive Sharpness: If too strong, sharpness may cause a grainy effect on displayed figures, canceling the effect of noise reduction.




### 2.1.4.1. Suggested Noise Reduction and Sharpness Configuration

The balance between Noise Reduction and Sharpness must always prioritize the device's main monitoring objective (e.g., capturing static plates such as at a toll plaza, capturing images with movement — such as on a high-traffic roadway, capturing clean/smooth images at night, etc.). Below, we detail optimized settings for various scenarios, aiming for maximum image performance in each application.

- Identification of static details (e.g., License Plates)

In this scenario, the main requirement is maximum edge fidelity. For this, we recommend using the 2D noise reduction filter or configuring the 3D filter at minimum intensity (0% to 20%). The 3D filter should be used at minimized intensity to prevent trails from obscuring characters. The Sharpness Level should be set to a higher value, between 25% and 35%, as higher values are crucial for enhancing edge definition, prioritizing the structural information of the image.

- Monitoring in Dynamic Environments (High Traffic/Movement)

In this scenario, the focus is preventing trails and blurring. The 2D noise reduction filter is the most suitable, as it is a spatial filter (single-frame based), making it more reliable in high-movement scenes. The Sharpness Level should be kept between 15% and 20% to ensure detail without exaggeration, mitigating the temporal effect of the 3D filter and ensuring clarity in capturing moving events.

- Standard Night Monitoring (Low Light)

The goal is effective smoothing of electronic noise. The 3D noise reduction filter is recommended at an intensity of 30% to 50%. Temporal processing is more effective against nighttime noise. The Sharpness Level should be slightly reduced to 10% to 15% to prevent residual noise from being enhanced, seeking the ideal balance between noise suppression and acceptable sharpness.

- Constant Lighting Environment (Controlled Light)

In this case, the priority is performance optimization. If the environment is well-lit and noise is minimal, Noise Reduction may be turned off or configured using the 2D filter. The Sharpness Level should be conservative, between 10% and 15%, since noise reduction is not mandatory under adequate lighting, optimizing the camera's processing cycle.

## 2.1.5. 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% |

*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.

## 2.2. Transitions Between Image Profiles

The *Transitions* menu allows configuring automatic transitions between configuration profiles of the capture device based on lighting conditions (light level) and time. This functionality is essential for automating the application of configuration sets throughout the day, such as switching between daytime and nighttime profiles according to the ambient lighting conditions.

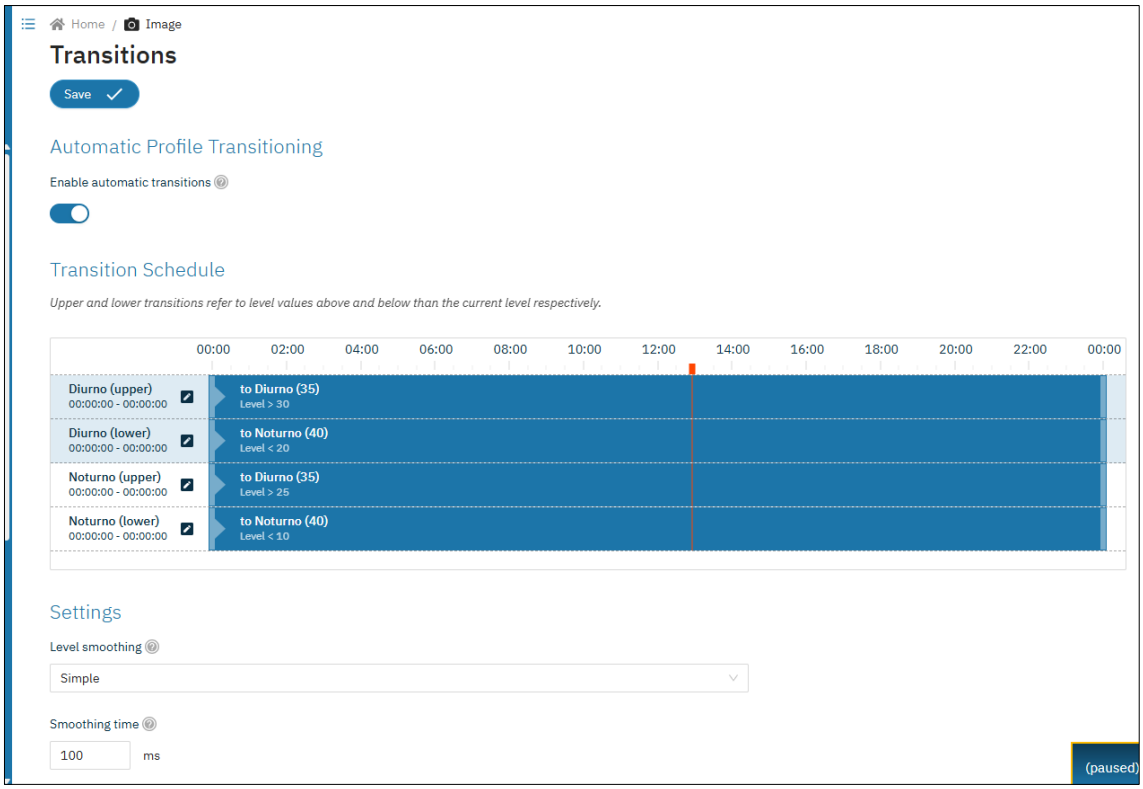
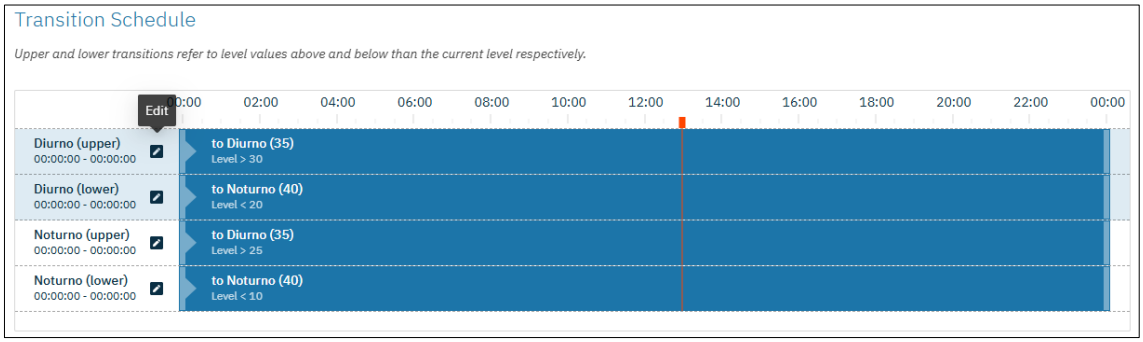



*Figure 14 - Image Menu > Transitions*

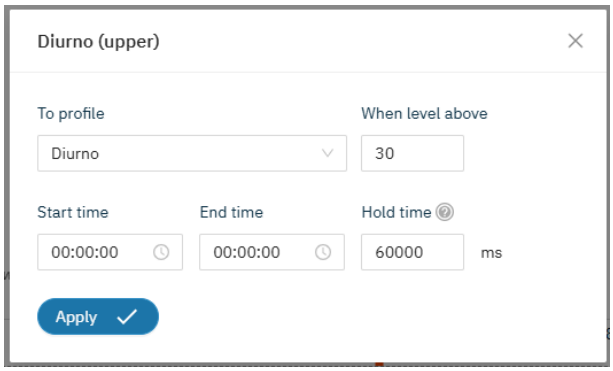
1) Enable the Enable automatic transitions option so that profile changes are applied automatically according to internal settings and lighting conditions, which helps to obtain ideal image quality in any situation:
    a. Use one profile for captures with ambient light (daytime period) and another for captures with artificial lighting (nighttime period), as recommended, with the possibility of registering up to four Image Profiles, each with its own configuration set.
2) Configure the *Transition Schedule* by clicking the *Edit* button next to the profile name;



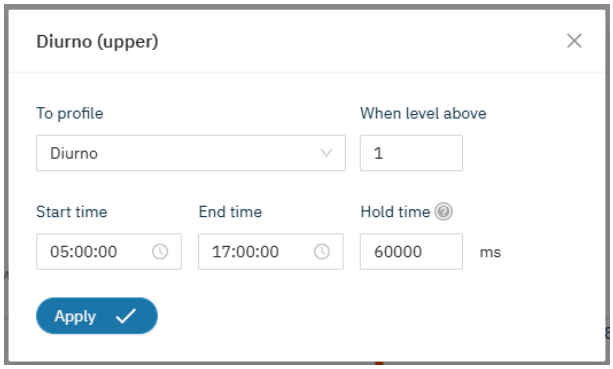
3) Define the Upper Transition options for the profile being edited:
    a. Select to which Profile the upper transition will occur;
    b. Define the Level value at which the upper transition should occur;
    c. Define a Start Time and/or an End Time for the upper transition;
    d. Define the Wait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

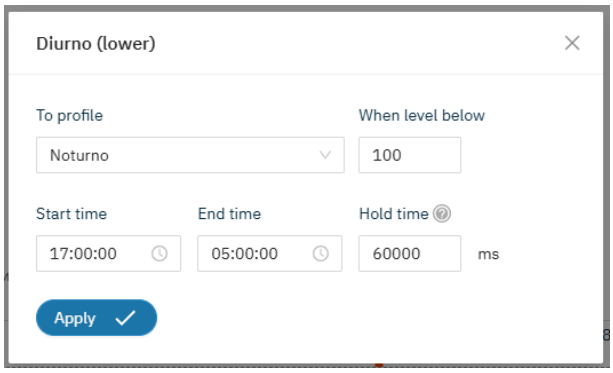
It is possible to configure the operation of automatic profile transitions in three different ways:

1) Transition by level only: If a level value is defined, the capture device changes profiles as soon as the configured level is reached, without considering the time (that is, without defining start and end times);

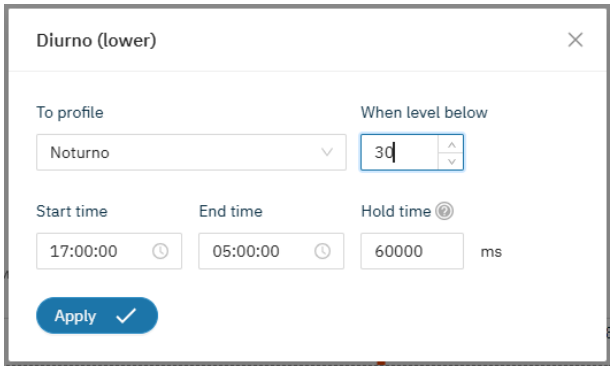



2) Transition by time only: If a level value greater than 100 or less than 1 is defined in the transition settings, it is possible to use profile transition based only on the defined start and end times. (In the example below, between 05:00 and 17:00 the daytime profile will remain active, and from 17:00 to 05:00 the nighttime profile will remain active);






3) Transition by level + time: If both the level value and the start/end time are configured, the transition will only occur when both conditions are true — that is, the image level must reach the defined value and the current time must be within the selected interval;




4) Select the *Level Smoothing* to be applied when switching between profiles;
   a. None: the level change is performed instantly when switching profiles;
   b. Simple: the level change is performed using a smoothing time during the profile switch;
5) Define the *Smoothing Time* for the automatic level transition between profiles, to set the interval in milliseconds corresponding to the filter time required to move from the level of the source profile to the desired level of the current profile;

### 2.2.1. Suggested Image Configuration for Transitions

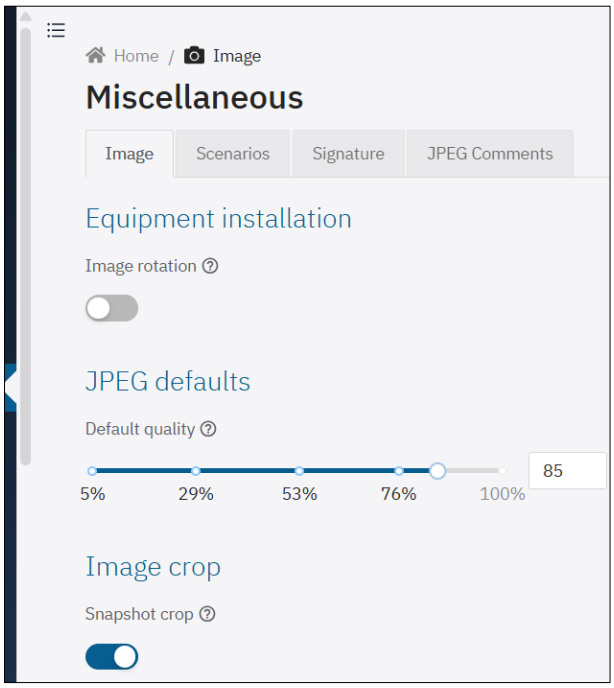
| Parameter | Suggested Value for Daytime Profile (Visible Light) | Suggested Value for Nighttime Profile (Infrared Light) |
|---|---|---|
| Enable automatic transitions | Selected | |
| Profile (Upper Transition) | Daytime | |
| Level (Upper Transition) | 30 | 10 |
| Wait Time* | 60,000 ms | 1,000 ms |
| Profile (Lower Transition) | Nighttime | |
| Level (Lower Transition) | 20 | 10 |
| Wait Time* | 60,000 ms | 1,000 ms |

* It is recommended to set the *Wait Time* close to 1 minute (entering the value 60,000 ms in the interface) so that the device does not oscillate unnecessarily between configuration profiles.

## 2.3. 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*:

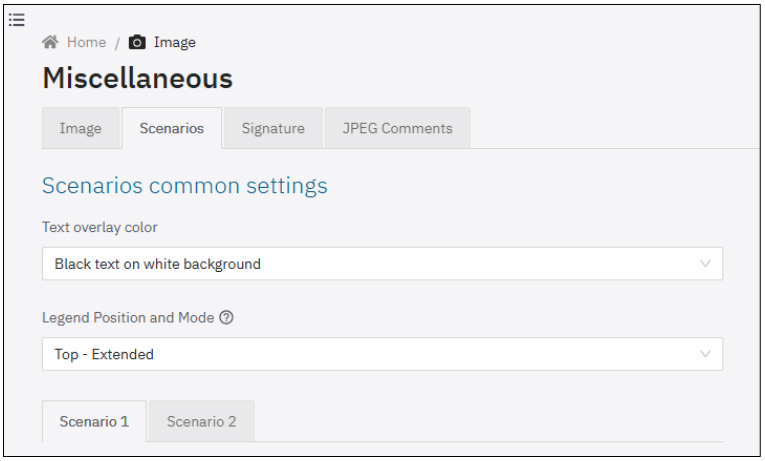
1) Go to the *Image > Miscellaneous* menu and the settings will be saved automatically;
2) Go to the *Image* tab and enable the *Image rotation* option when the equipment is installed upside down;
3) Set the *Default Quality* value at which the JPEG will be generated;




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;

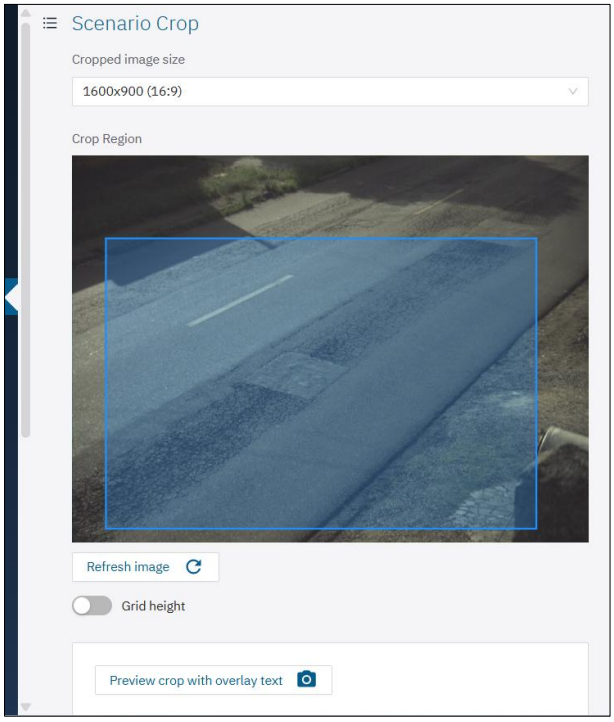
a. Select the *Cropping Mode* from the options:
    i. Static (cropping used to imitate a model with lower resolution);
    ii. OCR (cropping uses the ROI as a default position and if a plate is recognized, keeps the vehicle centered, adjusting the position and maintaining the same size);
b. Display 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*.

## 2.3.1. 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.

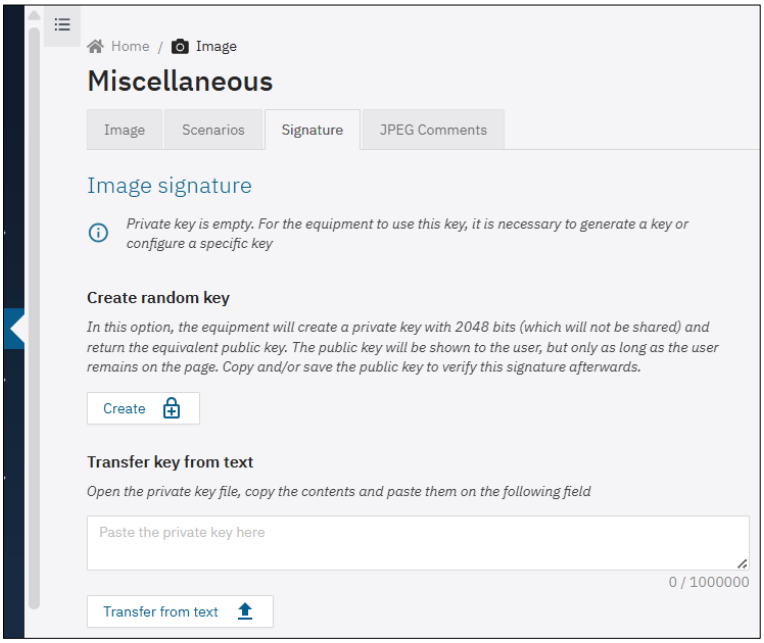
To ensure high security, the exponent used is always 65537. This data is appended, in text format, to the comment field of the JPEG image, defined in the "FF FE" marker according to the ISO/IEC 10918 specification.

To check the signature, carry out the following procedure:

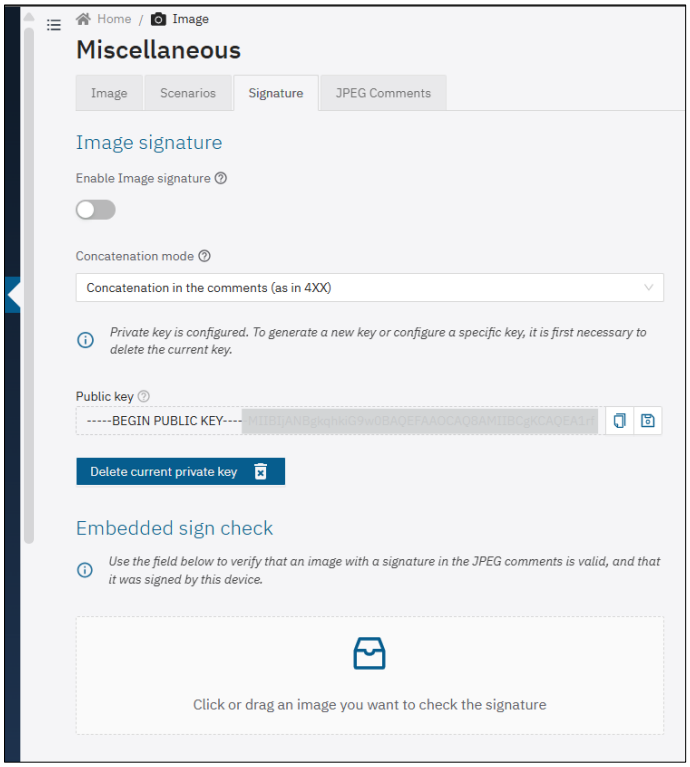
a) Remove the texts referring to *Sign*, *ExpoenteRSA*, *ModuloRSA* and hash (Sha256) from the comments field;
b) Calculate the hash of the image with the text specified in "a" removed;
c) Extract the hash from the signature (*Sign* field) considering the exponent (*ExpoenteRSA*) and the modulus (*ModuloRSA*);
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;



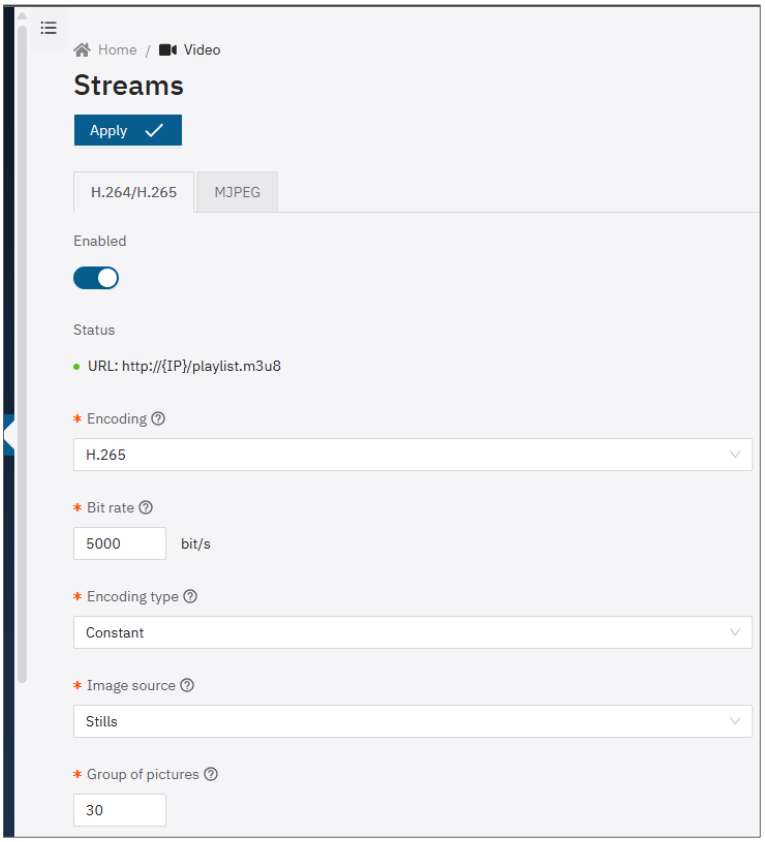

## 2.3.2. Jpeg comments

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.

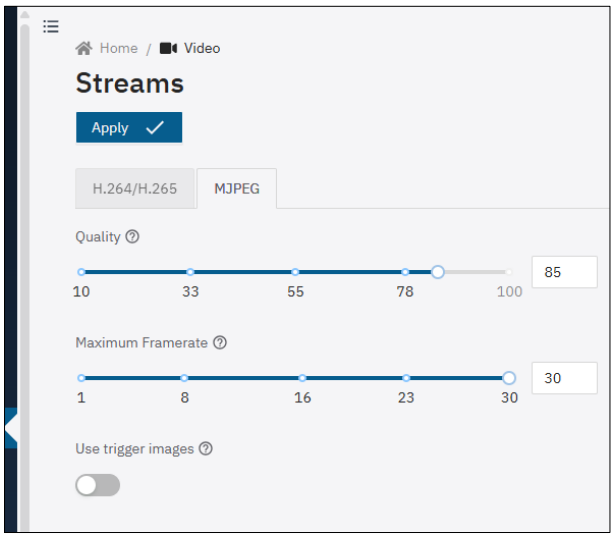

## 2.4. Video Settings (Streams)

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;



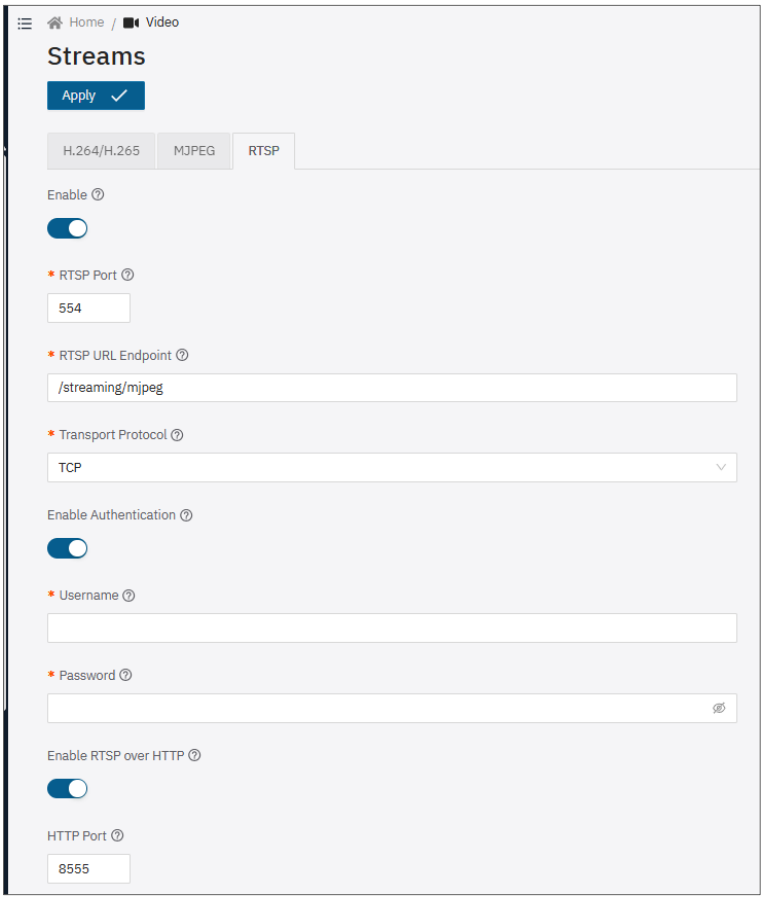

8) Access the stream configuration options in the *MJPEG* tab;
9) Adjust the quality of the images sent by the stream in the *Quality* field;
10) Set the *Maximum framerate* of images sent by the stream;
11) Enable the option to *Use trigger images* to display in the stream. With this function enabled, 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:




13) Access the settings in the *RTSP* tab by clicking *Enabled*;
14) Enter in *RTSP Port* the port number for the connection;
15) Enter the *RTSP URL endpoint;*
16) Select the *Transport Protocol,* from the options:
    a. UDP

b. TCP
c. TCP/UDP

17) Enable the *Enable Authentication option;*
    a. Define a User for RTSP access;
    b. Define a Password for RTSP access;
18) Enable the *Enable RTSP over HTTP* option to enable RTSP-over-HTTP connection (valid only for TCP transport);
    a. Enter the *HTTP Port number*




# 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).

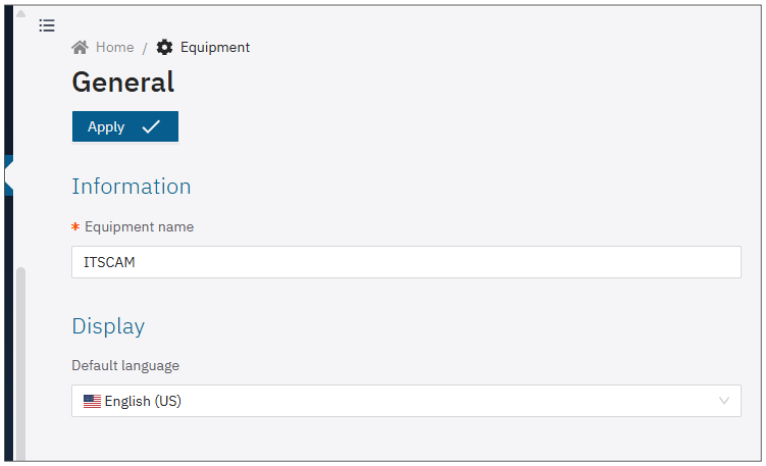



*Figure 15 - Equipment > General menu screen*

## 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:

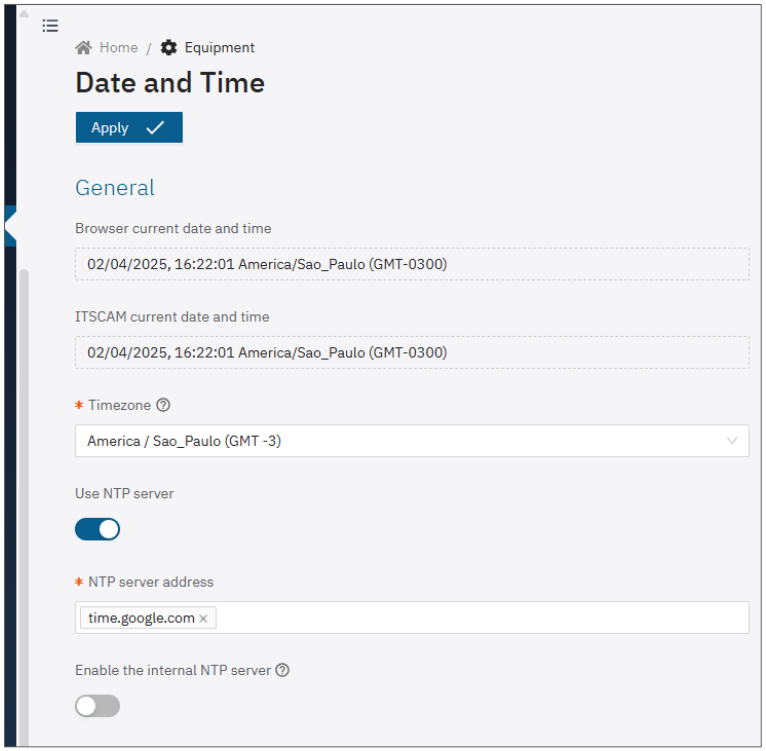



*Figure 16 – Equipment > Date and Time menu screen*

## 3.3. Network Settings

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

3) Enter the factory default IP address in the browser;
4) Enter *user* admin and *password* 1234;
5) Access the menu *Equipment > Network*;
6) 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*:

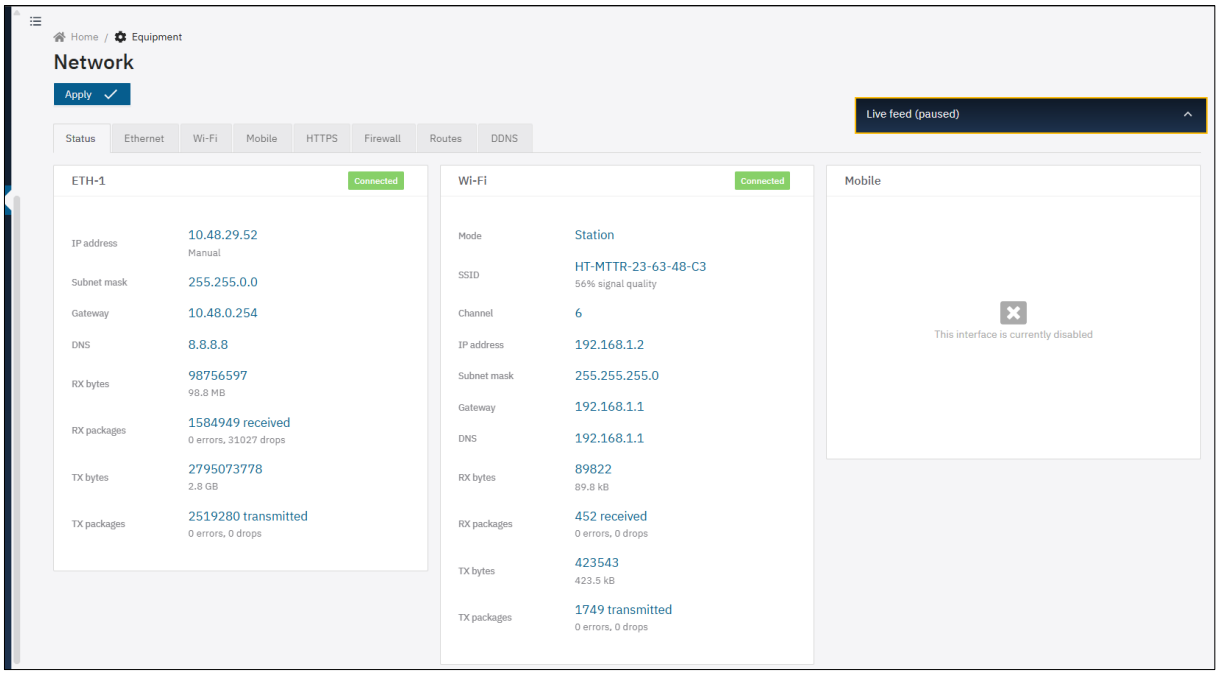



*Figure 17 - Network settings home screen*

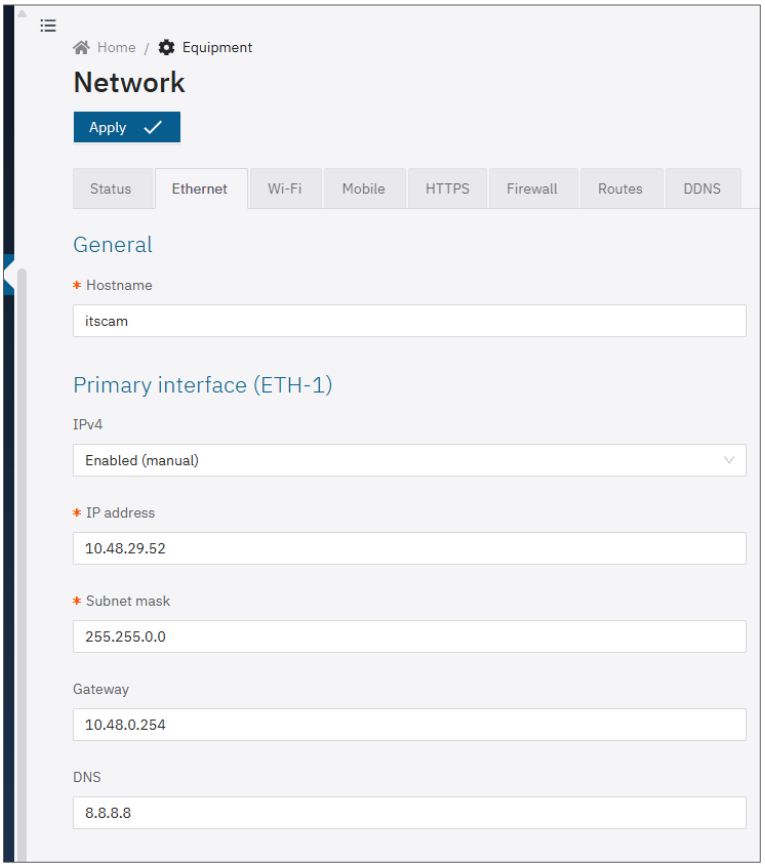
### 3.3.1. Ethernet Network Settings

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.**

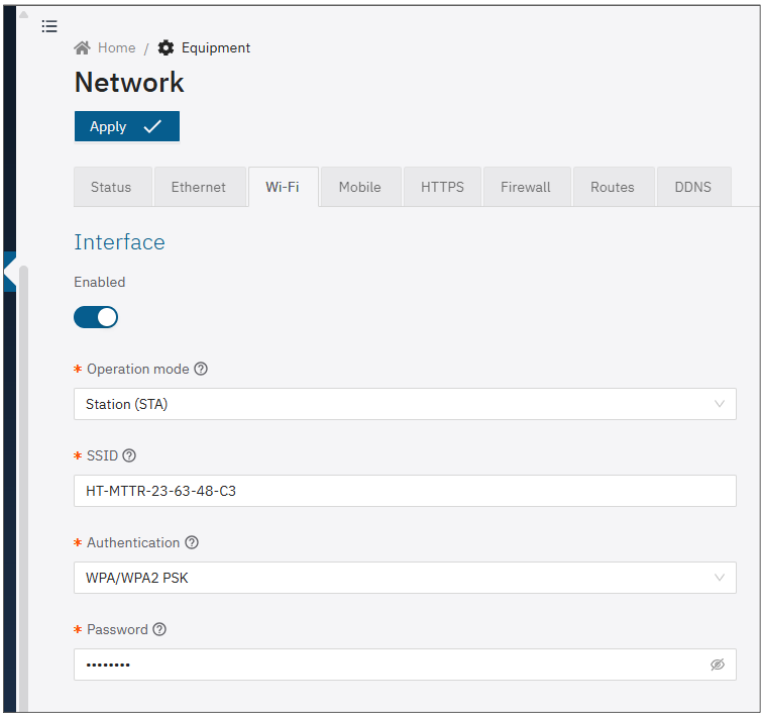
4) Click *Apply* after confirming the data entered:




## 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:

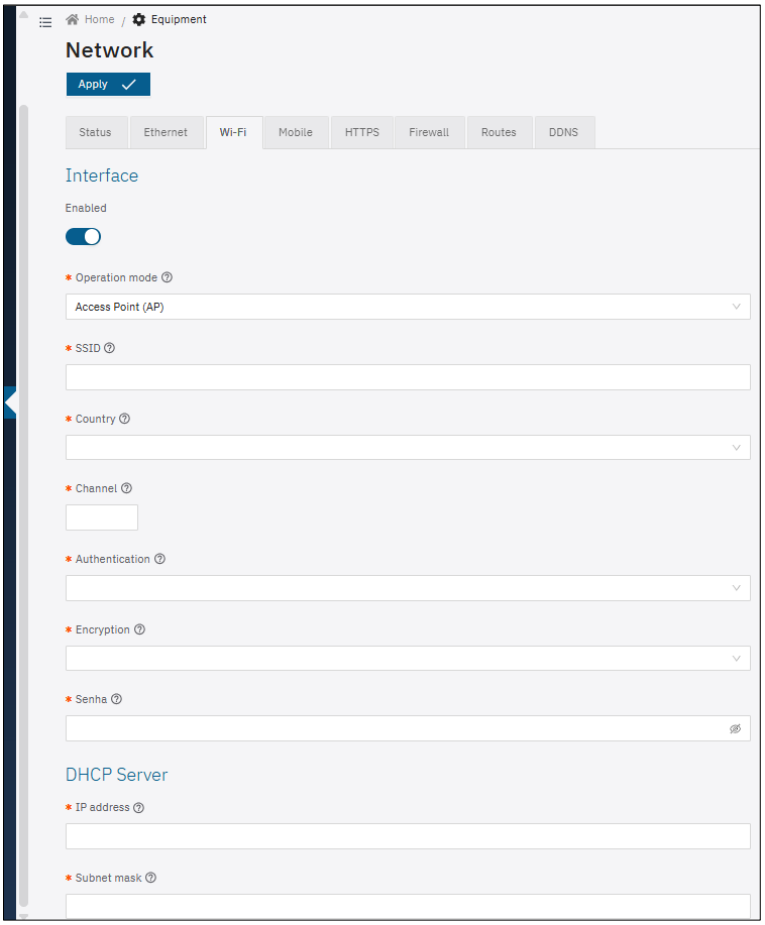



Select the *Access Point (AP)* option in the *Operation Mode* field only when the device is to be used as an access point to an available Wi-Fi network:

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.**

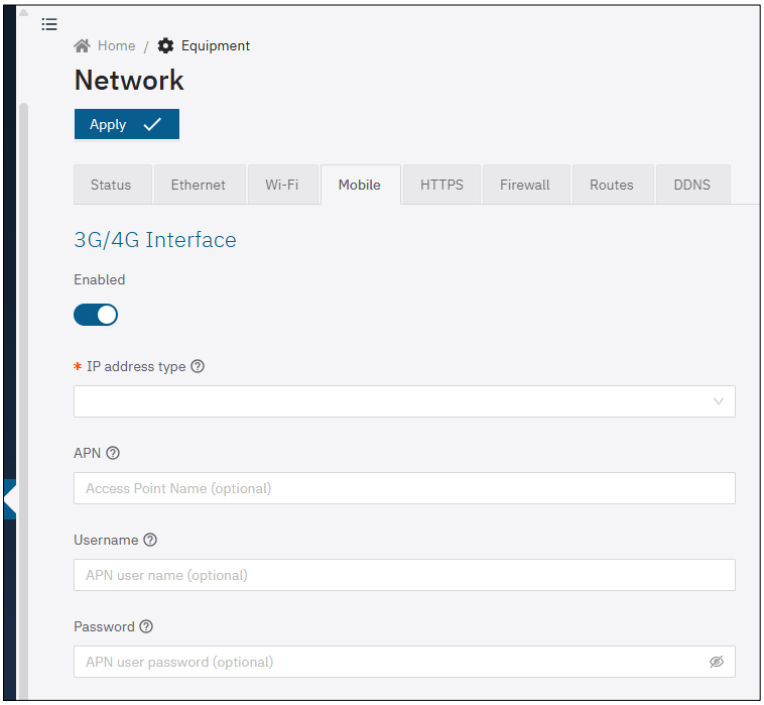
5) Click *Apply* after confirming the entered data:




## 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:

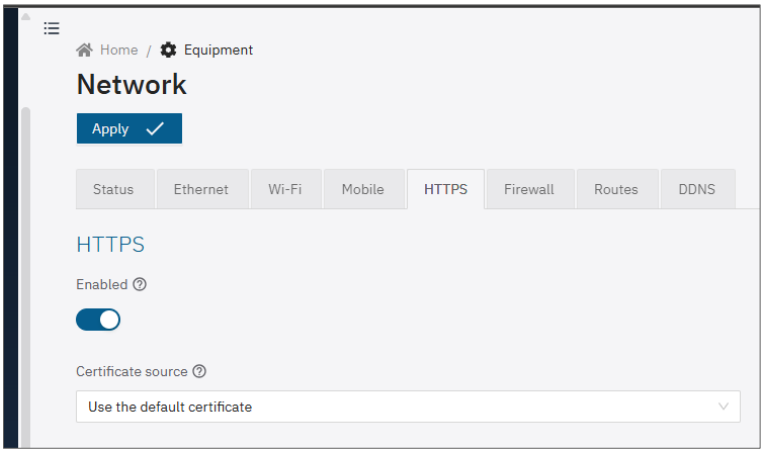
1) In the Mobile tab, select the option *Enabled* for 3G/4G interface;
2) Fill in the APN data if there is no automatic detection, entering custom data whenever you need to configure the operator's information. By default, the information is:
    a. *APN*: http://[provider's name].com.br;
    b. *User*: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



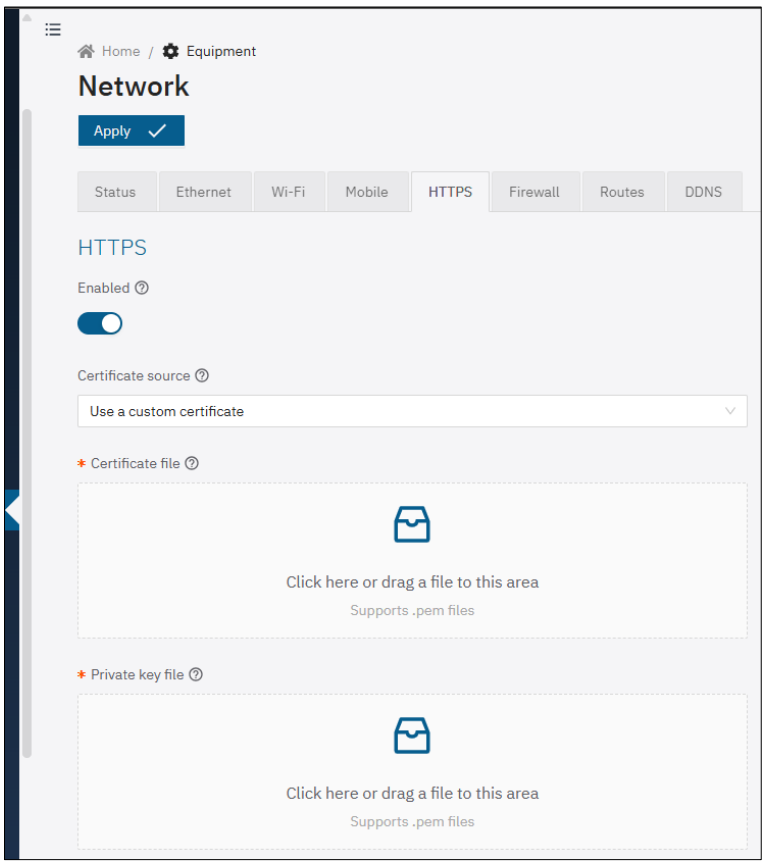

## 3.3.4. HTTPS settings

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:

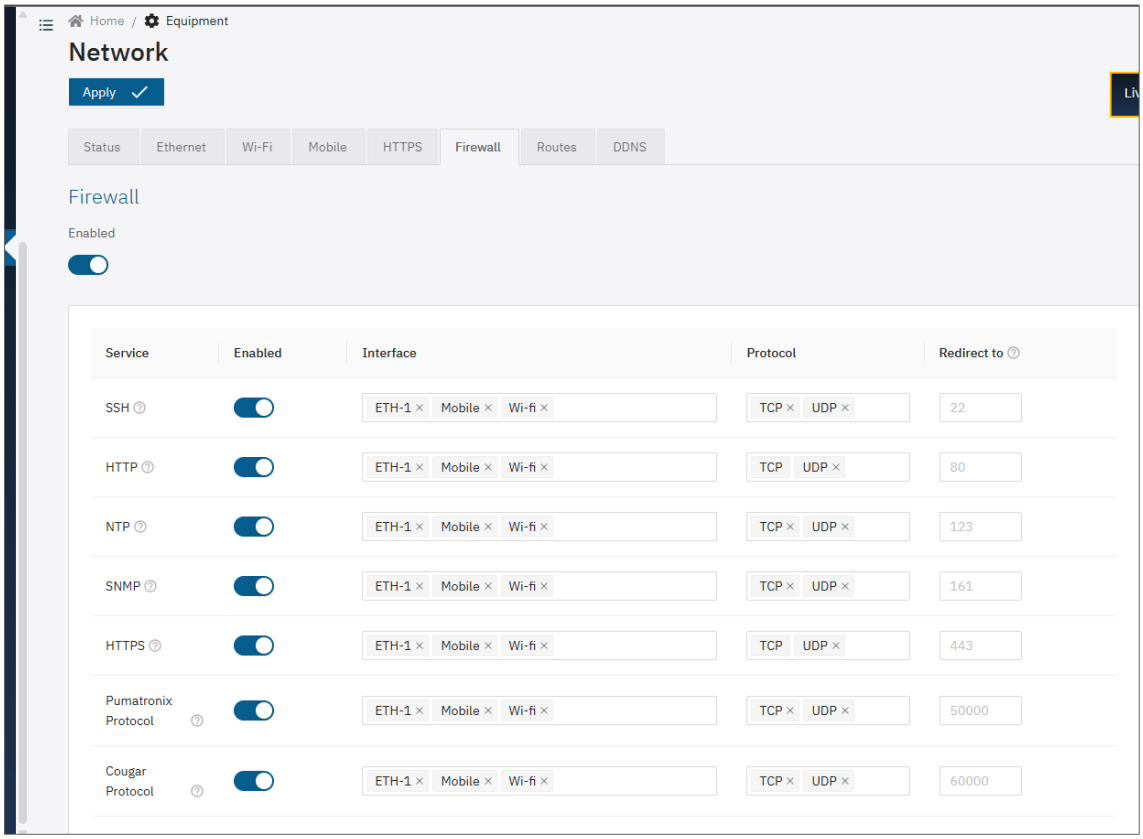



**The use of HTTPS changes the access link to the device from `http://` to `https://` and the device must be *Restarted* for the change to be applied, by going to *System > Update*.**

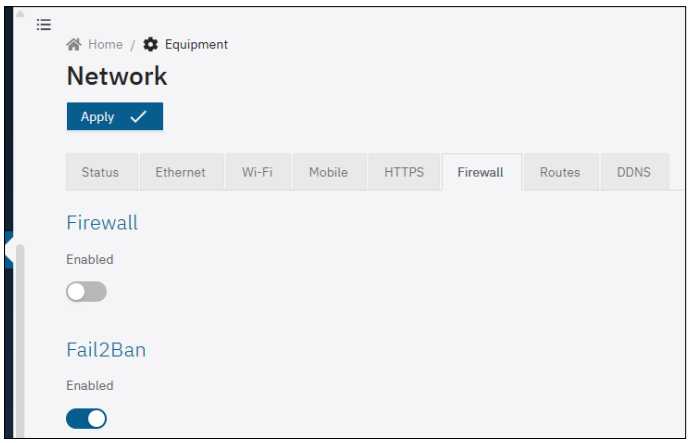
## 3.3.5. Firewall settings

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;

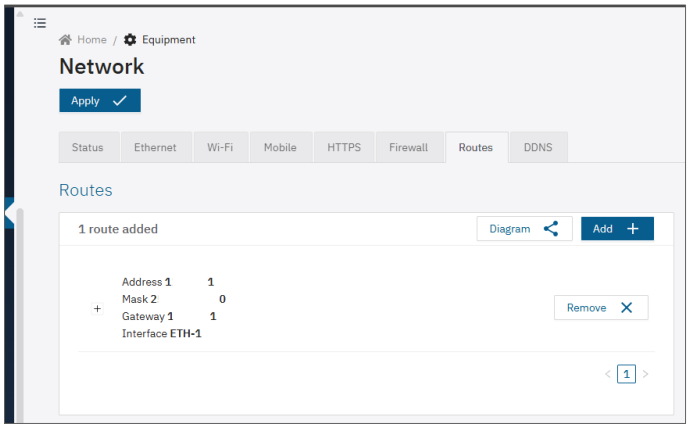



3) Click the *Apply* after confirming the data entered.

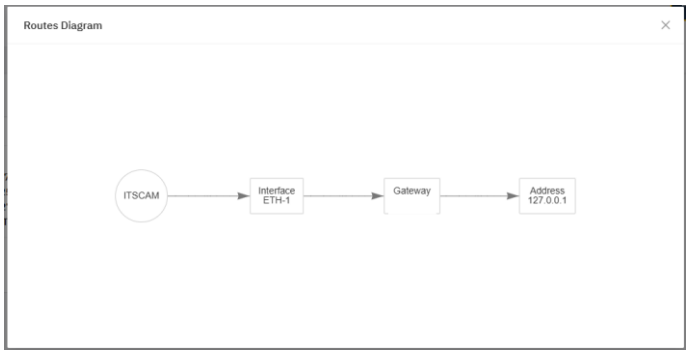
## 3.3.6. Route settings

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*.

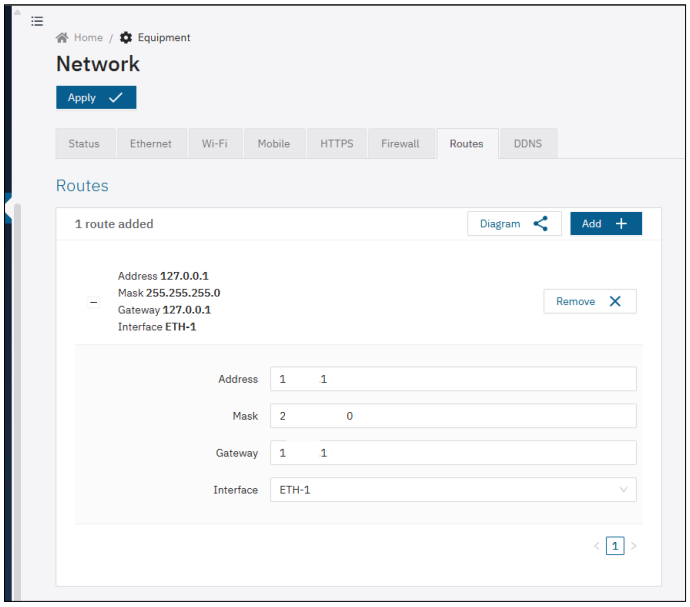
1) Select the *Route* tab;
2) Add a route by clicking *Add+*;




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:

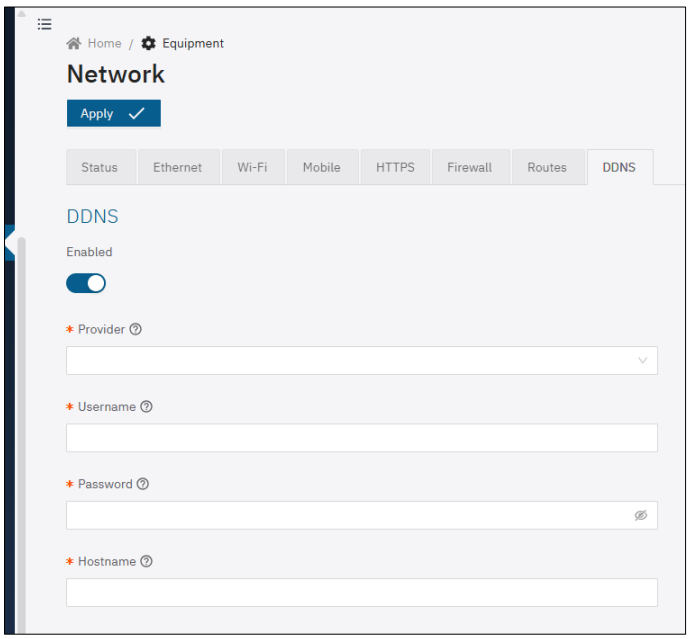


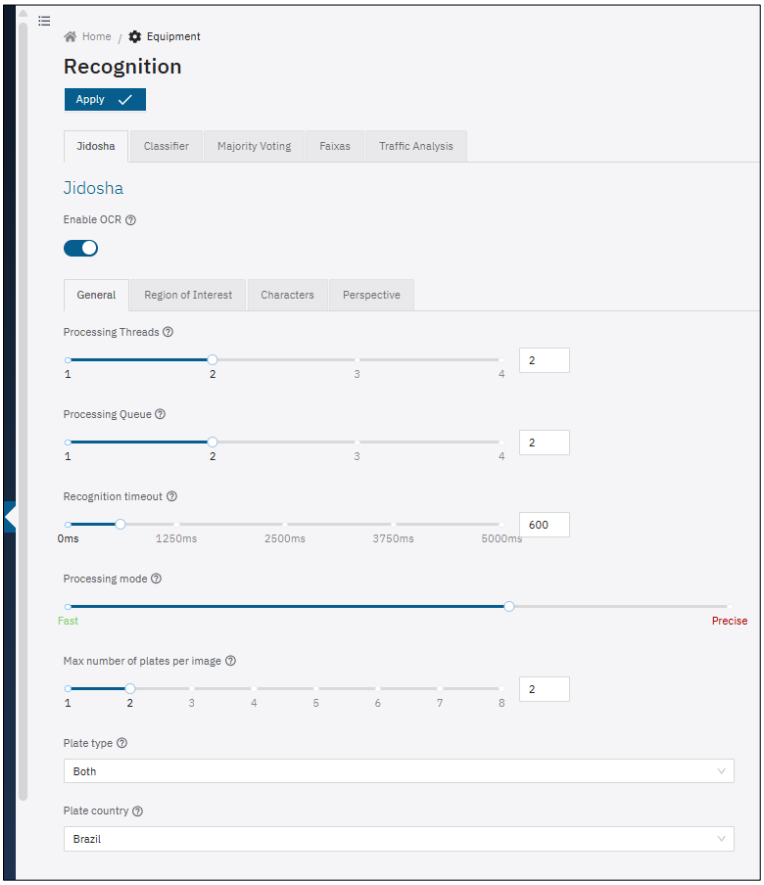

*Figure 18 - Equipment > Network menu screen when configuring the DDNS*

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.

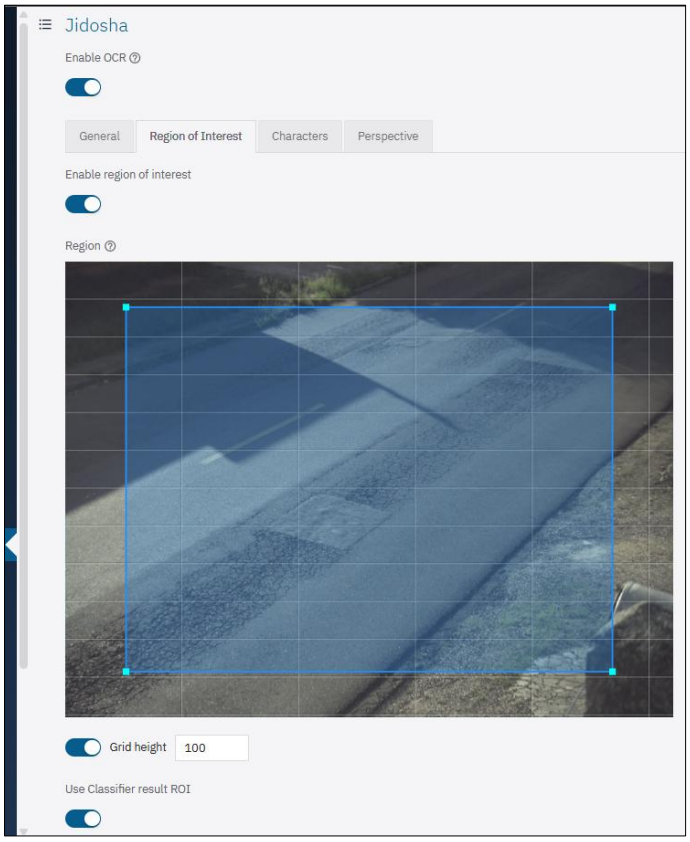
## 3.4. Recognition (OCR Reading)

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;

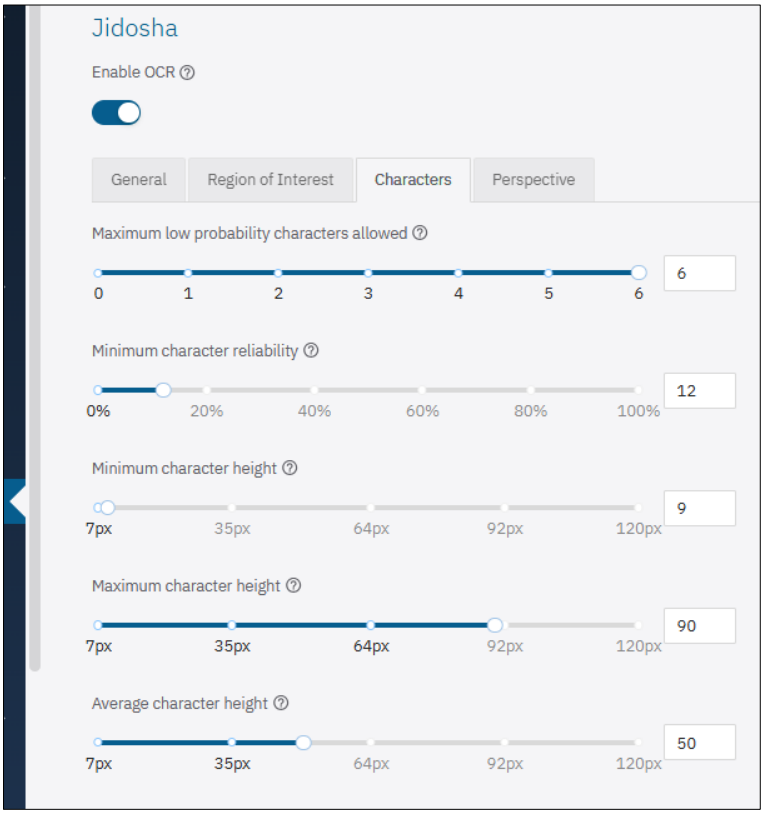


b. Set 10 for the *Grid height* (in pixels)
    i. Note the size of the license plate characters in the region of the image marked as 1 in the example, i.e. when the vehicles are in the most distant region to the capture device. The characters should be approximately the grid's height. If they are very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

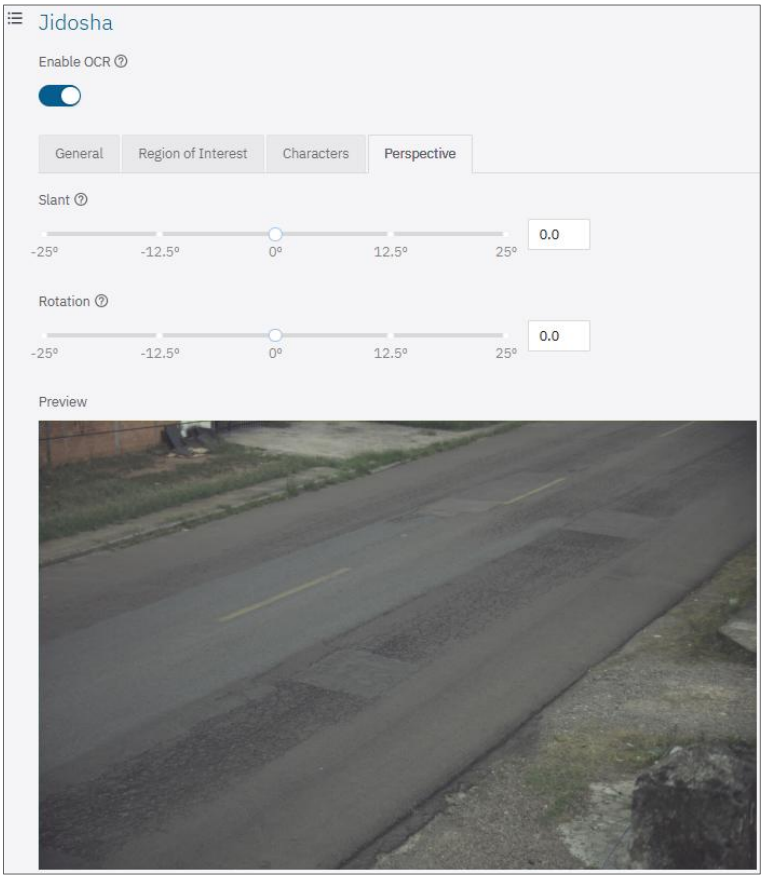

c. Set 40 for the *Grid height* (in pixels);
    i. Note the size of the license plate characters in the region of the image marked as 3 in the example, i.e. when the vehicles are in the region closest to the capture device. The characters should be approximately the grid's height. If they are very different,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



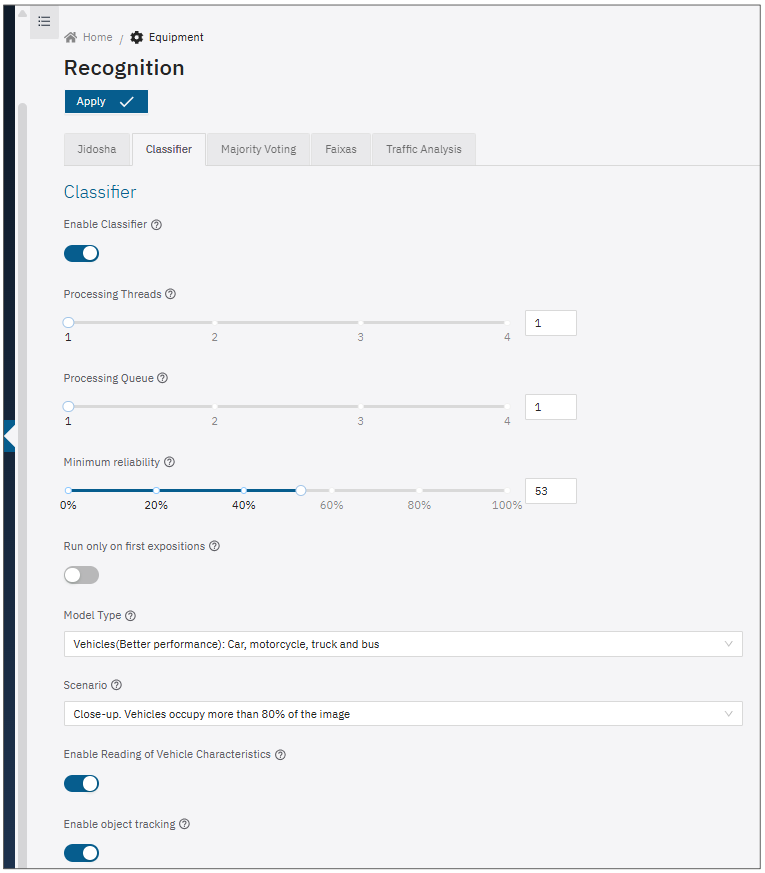

6) Access the setting options in the Perspective tab:
   a. Adjust the *Slant* angle of the plates as they appear in the image, removing the 'italics' effect, in order to improve OCR efficiency:
      i. Take a photo of a plate in the desired position, using the functionality in *Equipment>Images>Snapshot*, 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) View the effect of the adjustments in Preview




8) 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;



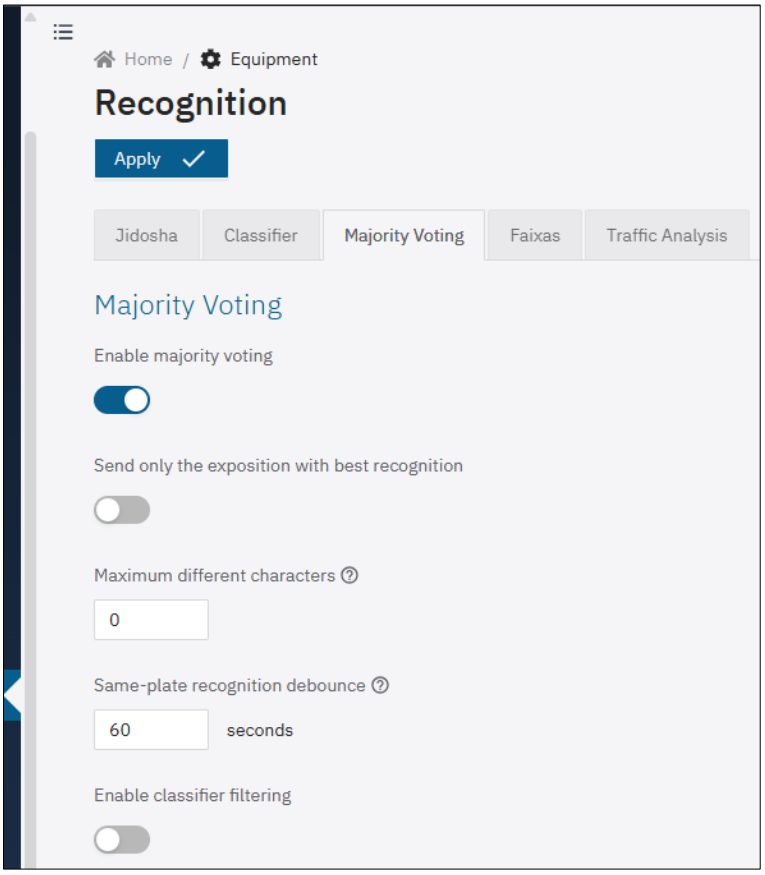

6) Select the *Enable object tracking* option to generate a high-precision *virtual loop* (virtual trigger):
7) Create the *Virtual Trigger* Regions which are made up of 3 pieces of information: 2 points (indicating a line) and a direction (represented by the third point of the triangle), indicating 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;


8) Click on Display experimental features to display the Speed estimator settings:

a. 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.

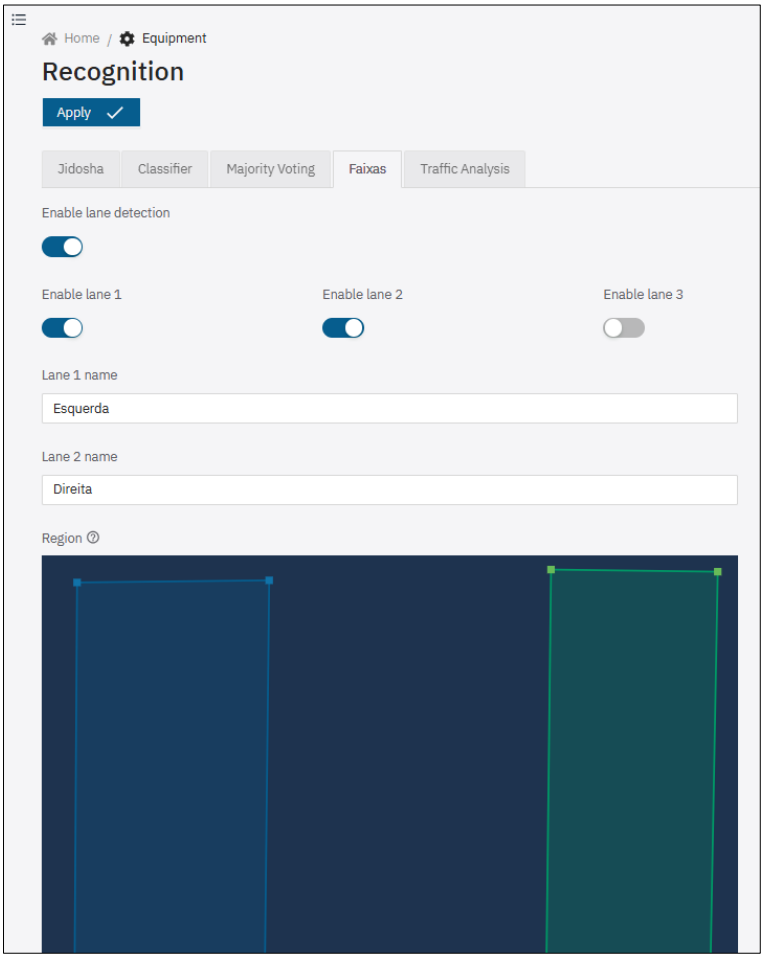

## 3.4.2. Majority Vote

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.



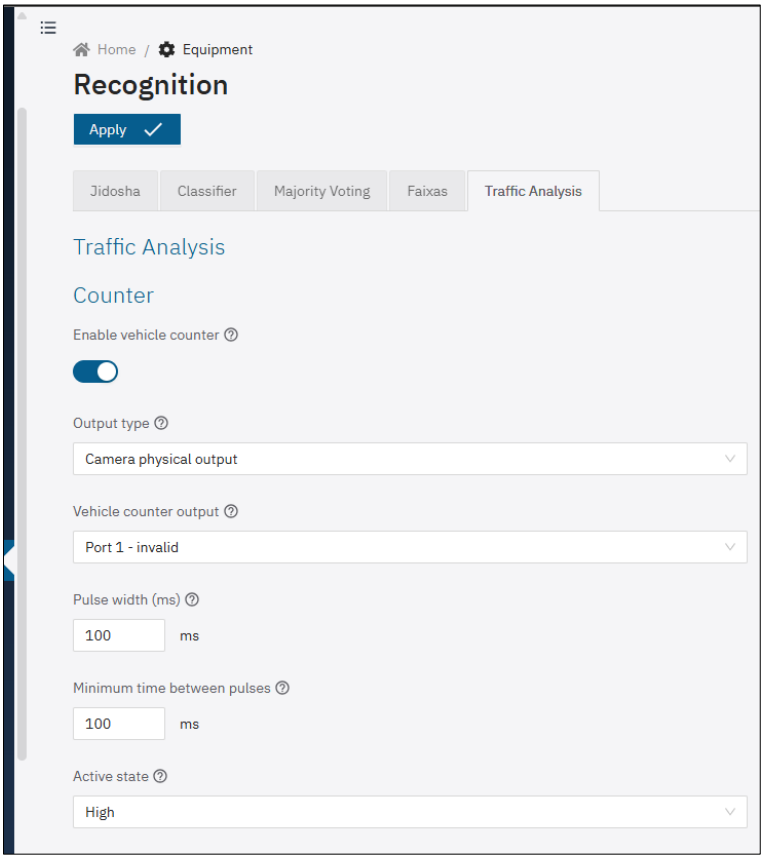

### 3.4.3. Lanes

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;

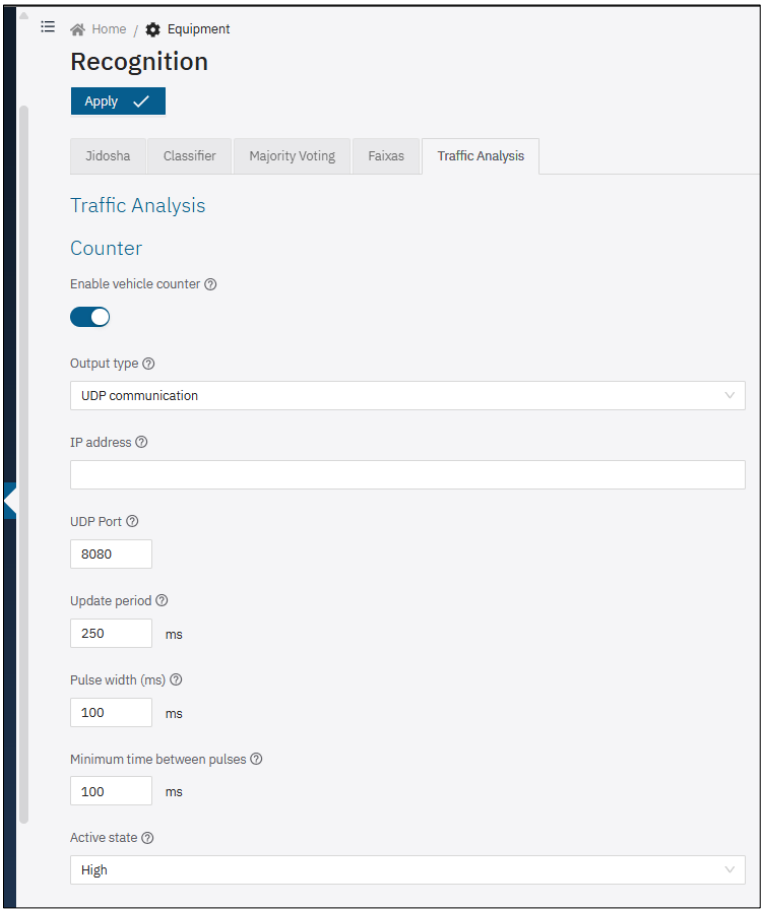



## 3.4.4. Traffic Analysis

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;



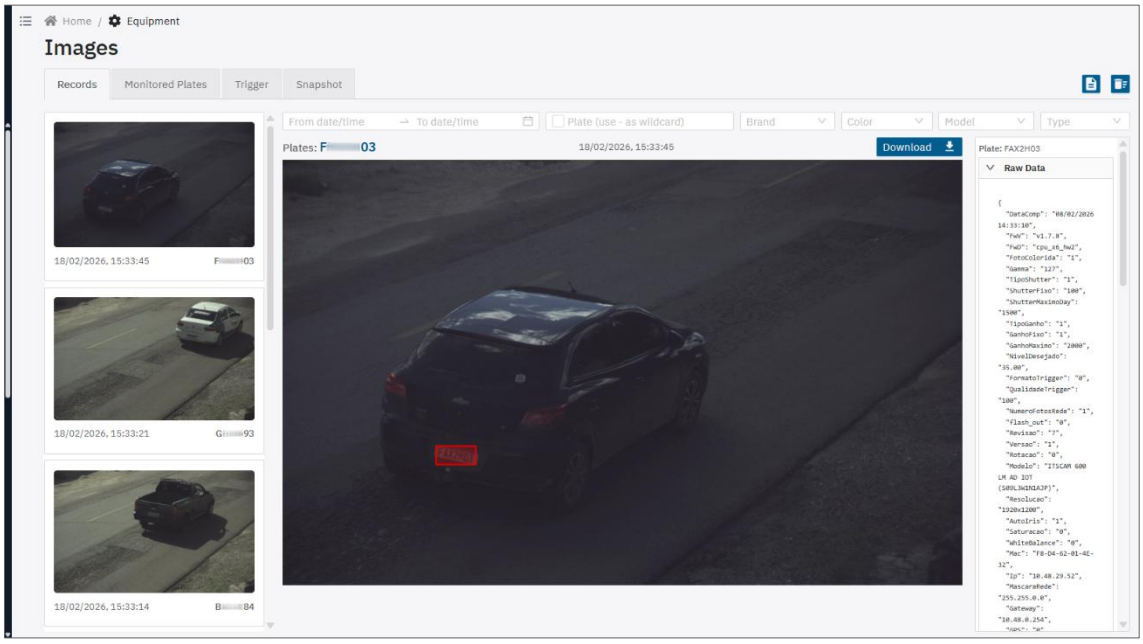

b. UDP Communication: configure by indicating the *IP Address* of the *UDP server* and the UDP Port receiving the pulses, the *Update Period* with the time between each send to the server indicating vehicle count or not, the *Pulse Width (ms)*, the *Minimum Time Between Pulses* with the minimum time 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.



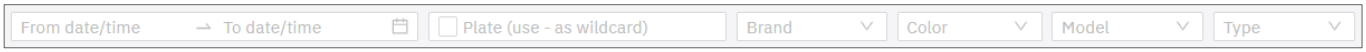

## 3.5. Checking the Images Generated

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:

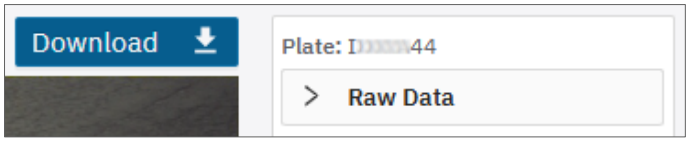
1) Select the *Records* tab to view the history of captures while the screen is open, displaying image data and also allowing you to search by license plate or vehicle characteristics;



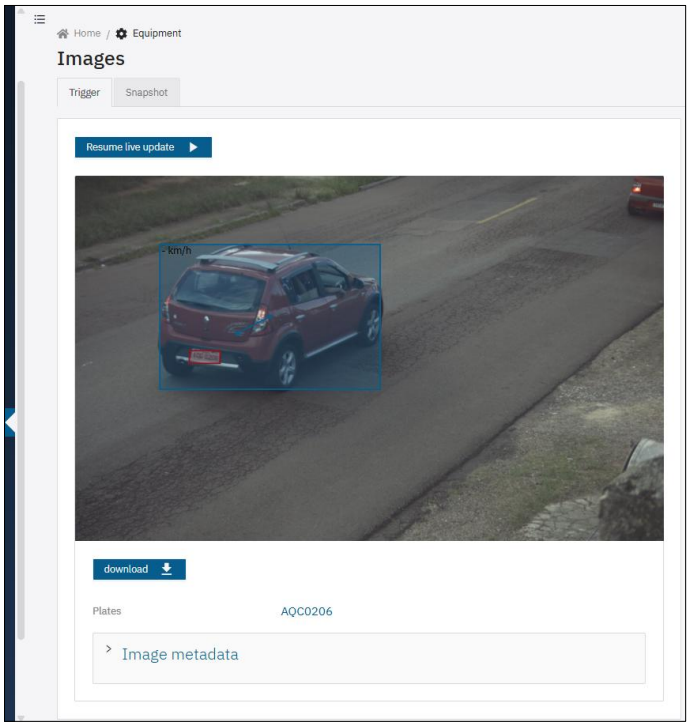
a. Select from the filters *From Date/Time – To Date/Time*; *Plate*; *Brand*; *Color*; *Model*, and *Type* to search for custom captures;
b. Click the *Download Records* button to download all captures for the day;
c. Click the *Clear All Local Data* button to delete the day's captures;



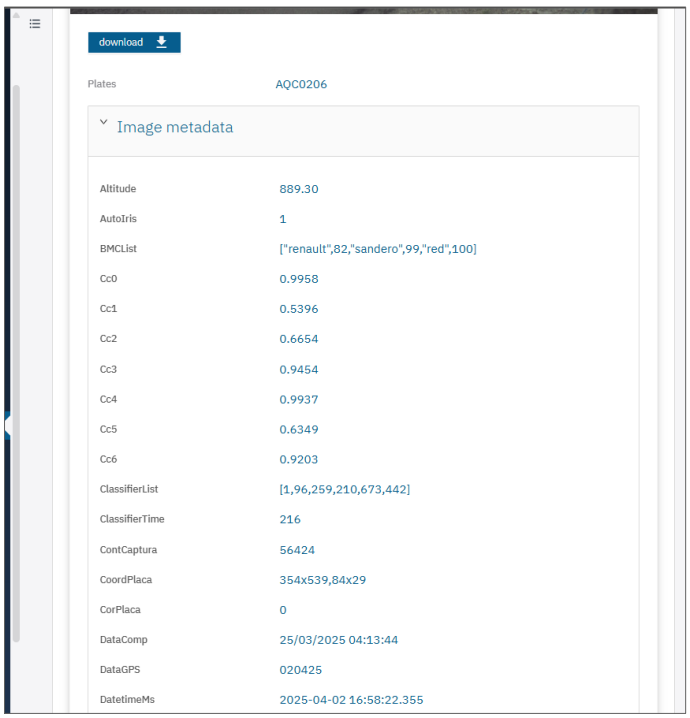

d. Click the *Download* button to download only the last capture displayed;
e. View the image metadata by clicking on the *Raw Data* field.



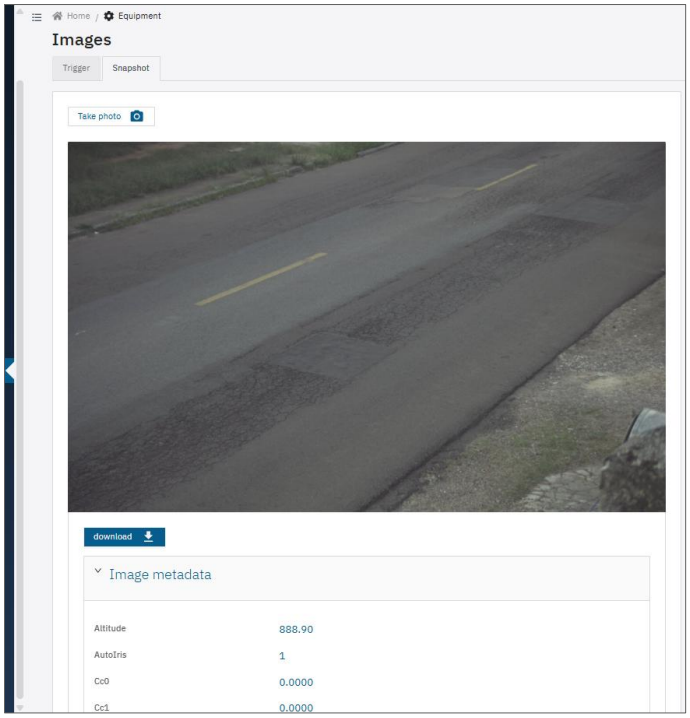

2) Select the *Trigger* tab and the last capture made is displayed, waiting for the last capture to be updated when necessary;




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;




3) 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;



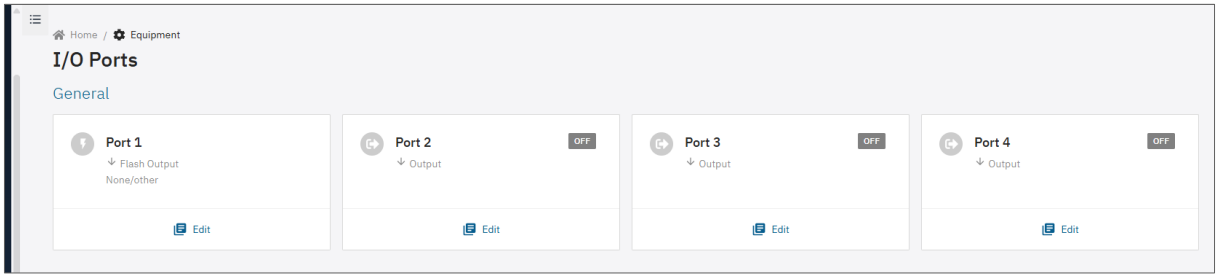

4) Make the necessary adjustments using the information from the current image settings.

## 3.6. I/O Ports

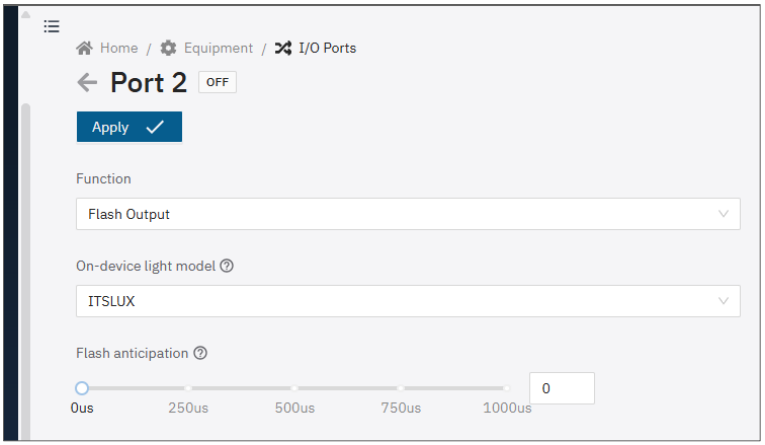
ITSCAM 600, ITSCAM 600 FHD, ITSCAM 600+ and VTR 600 devices have 4 ports that must be configured by software as inputs or outputs (IOs), that are available to controlling the illuminator trigger or for the

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:

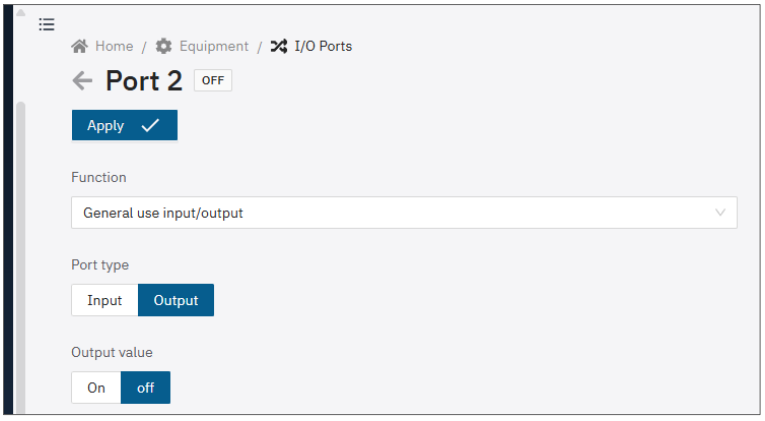
1) Access the settings in the menu *Equipment > I/O Ports*;
2) Locate the *Port* to be configured;
3) Click on the corresponding *Edit* button;



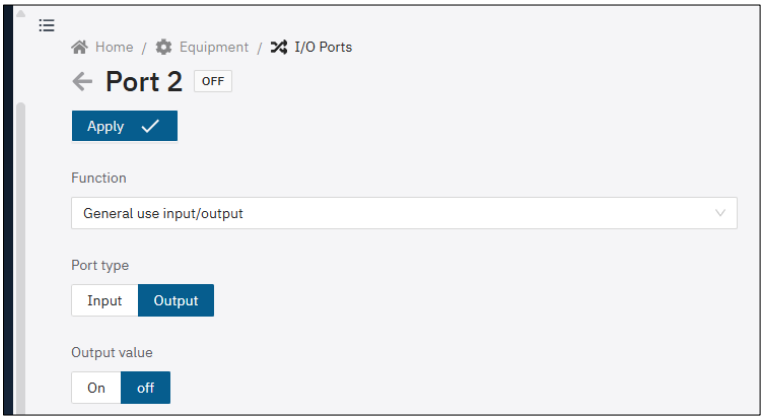

4) Select the *Function* as *Flash Output* when the IO port is used to flash the lighting device:
   a. Select the *On-device light model*: ITSLUX, WHITELUX, WHITELUX (video) or, if the lighting device is from another manufacturer, None/other;
   b. Set the *Flash anticipation* value to align the image 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);



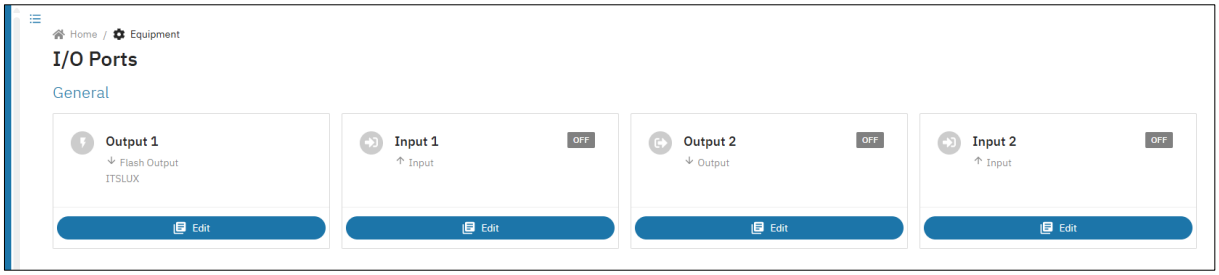

   b. Select the *Port Type* as *Output* (example: when triggering gates or alarms);
      i. Select *Output value* as *On* to activate the output signal;
      ii. Select *Output value* as *Off* to deactivate the output signal;
6) Apply the port settings by clicking *Apply*.




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

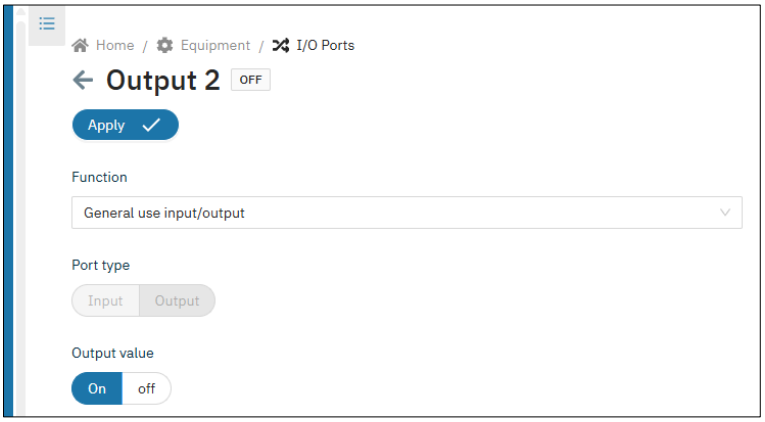
The ITSCAM 450 and ITSCAM 450+ devices have 2 inputs and 2 outputs, which are available for controlling the triggering of the illuminator or for installing external sensors, such as loops and light barriers, which identify the moment of image capture (trigger). To configure the outputs, access the menu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

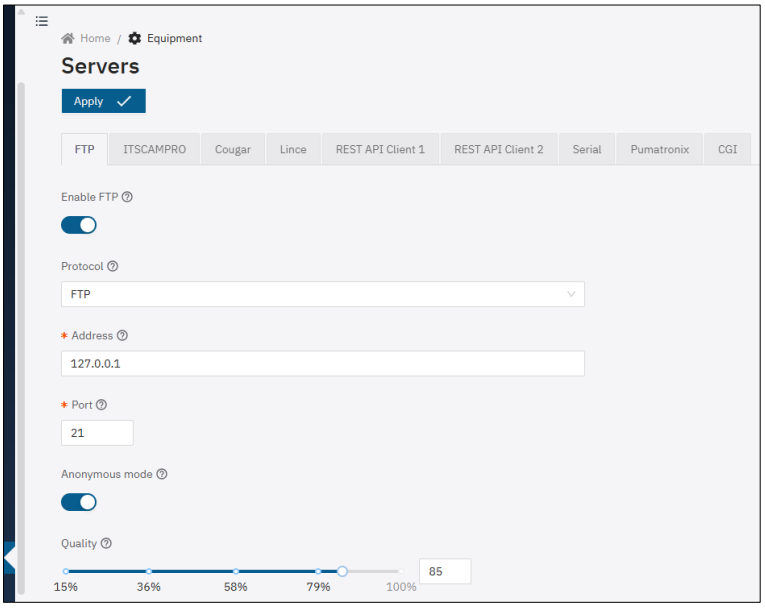
## 3.7. Servers

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

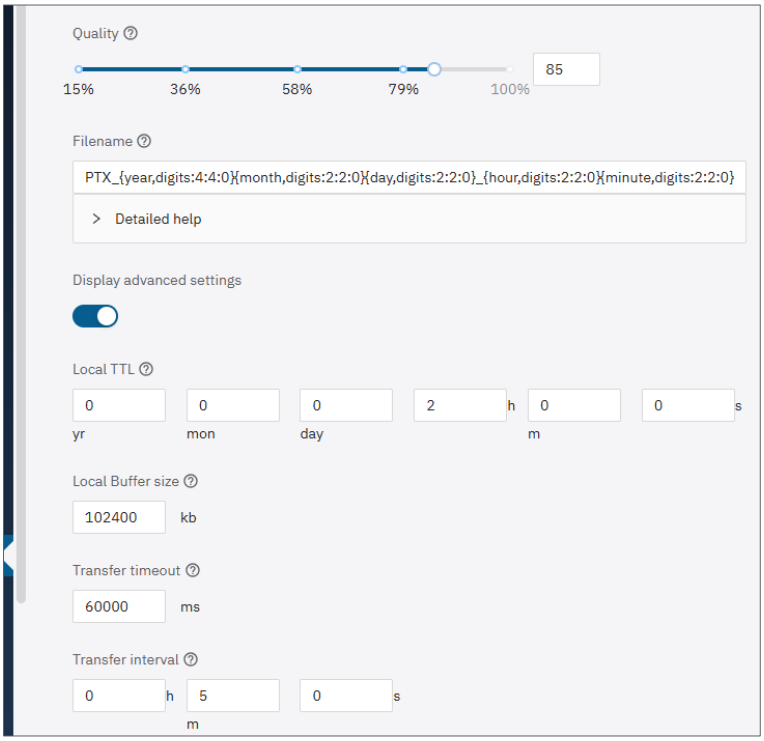
1) Go to the *Equipment > Servers* menu;
2) Select the tab corresponding to the server to be configured: *FTP*, *ITSCAMPRO*, *Cougar*, *Lince*, *REST API Client* or *Serial*;
3) Check Maintenance for the steps to restore the system to factory defaults, should any errors occur while configuring the servers.

### 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;



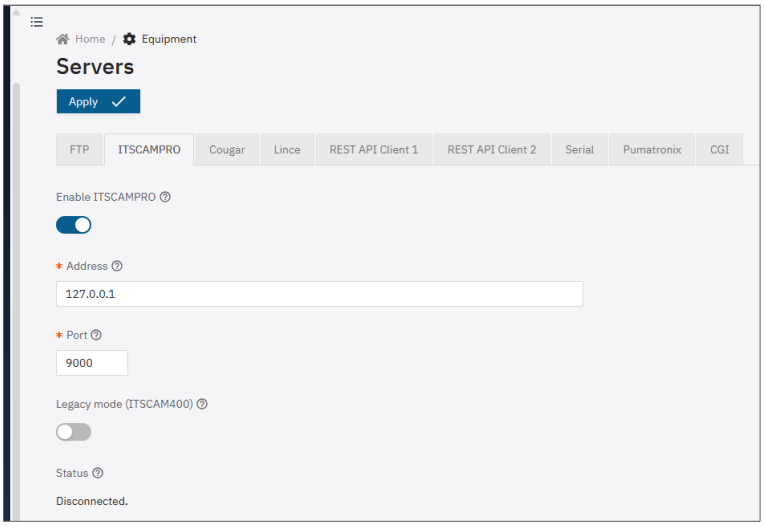

6) Adjust the quality of the JPEG image when saving via FTP by selecting it in the *Quality* field;
7) Change the code in *Filename* to customize the name of the file with information from the capture. To ensure that the name is in the correct format, a validation mechanism has been implemented that 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:

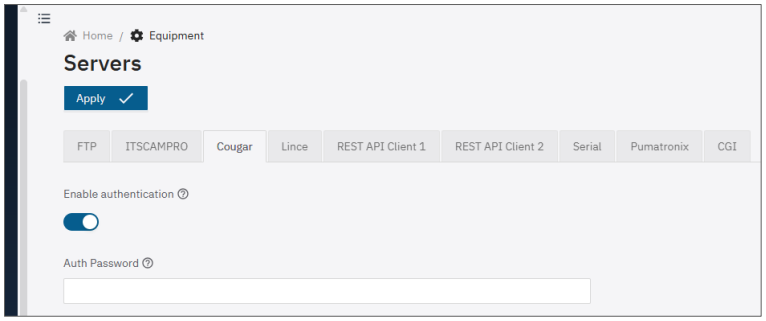
1) Select the *Enable* ITSCAMPRO option on the ITSCAMPRO tab;
2) Fill in the IP *Address* of the ITSCAMPRO server, which must be a domain name or a valid IPv4 address;
3) Enter the access *Port*, between 1 and 65535;




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

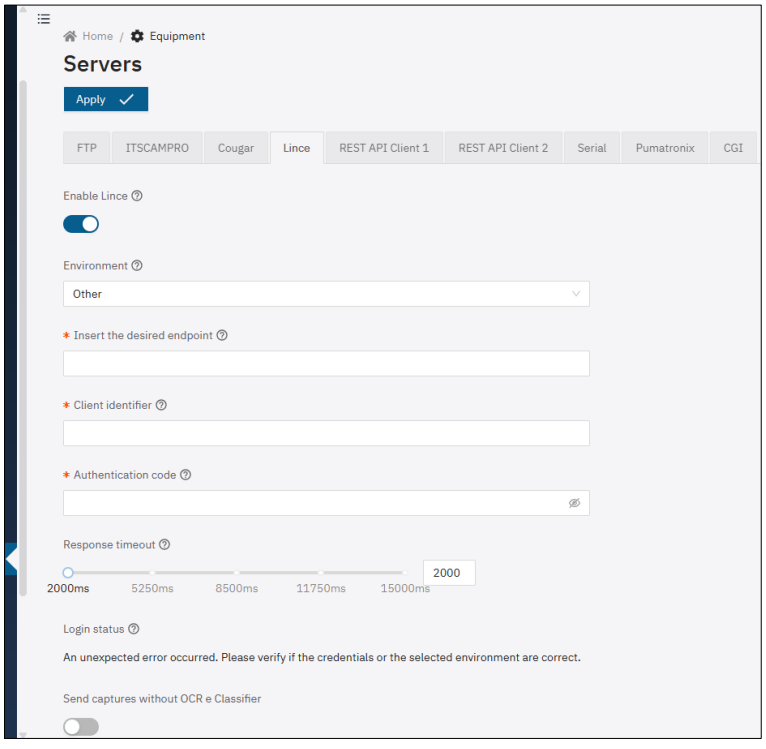
1) In the Cougar tab, select the *Enable* authentication option;




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

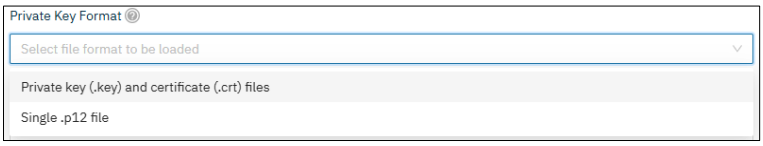
1) In the *Lince* tab, select the *Enable Lince* option;
2) Select the server operating *Environment* from the options: *Development, Homologation, Production* or *Other*;
3) *Enter the desired endpoint* by entering the desired URL for sending the captures via the Lince server. Example: lince.app.br 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.



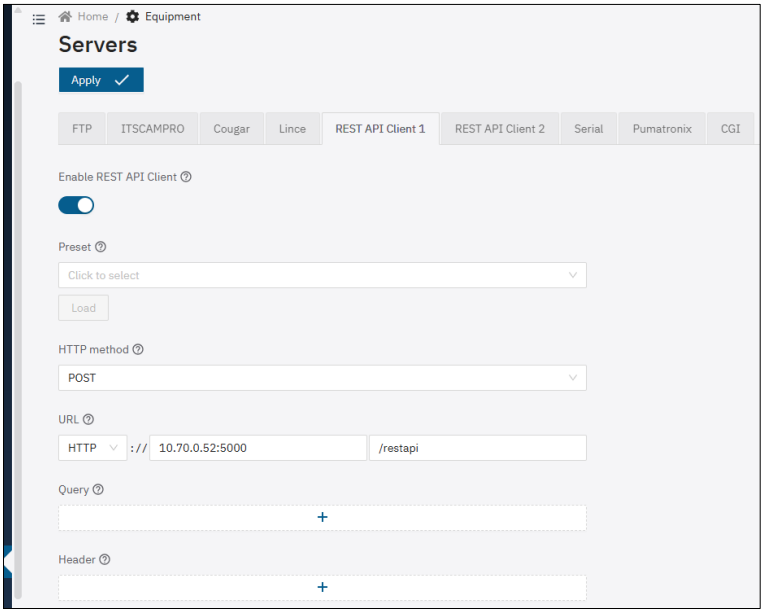
## 3.7.5. REST API Client Server

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

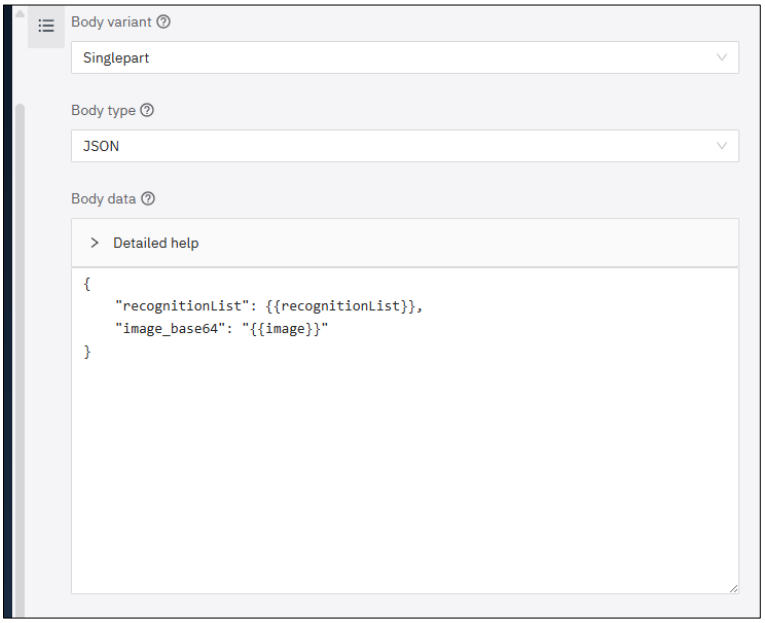
1) In the *REST API Client* tab, select the *Enable REST API Client* option, which allows sending captures to a generic HTTP server;
    a. Select a Preset under *Presets* to apply a preset to some fields by clicking the *Load* button;
2) Select the *HTTP Method* of the custom request from GET, POST and PUT;
3) Enter the URL for the custom request, indicating the schema, host and path filled in separately;
4) If a Trusted Certificate or Private Key is registered in Equipment > TLS, the Key for mTLS option will become available:



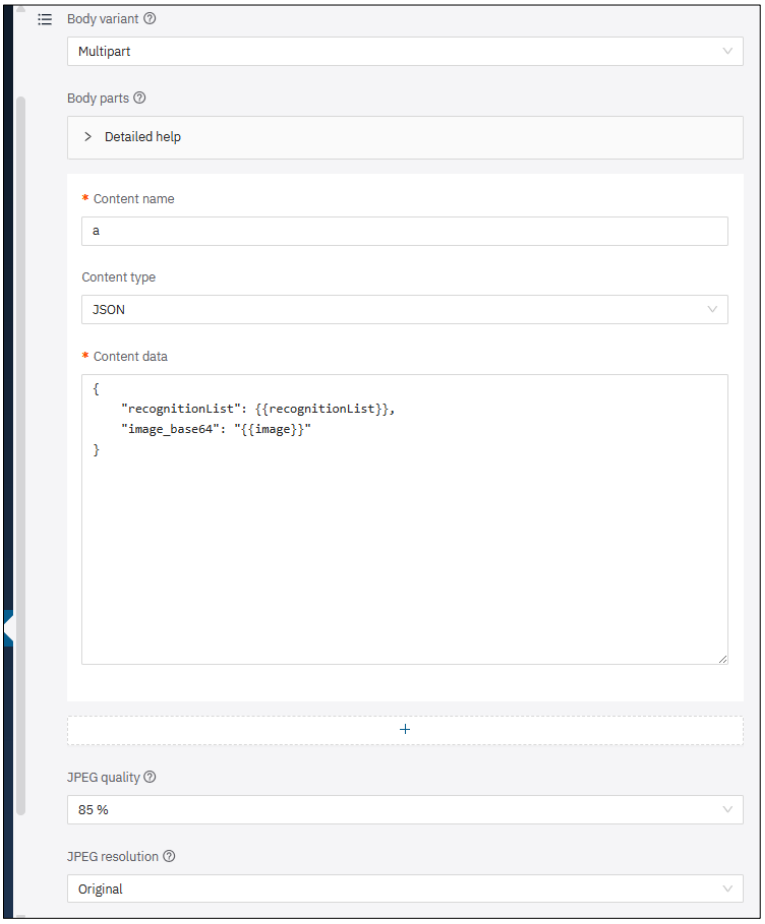

5) Enter the *Query* parameters for the custom request by clicking +;
6) Enter additional headers for the custom request in *Header* by clicking + and entering name and value;



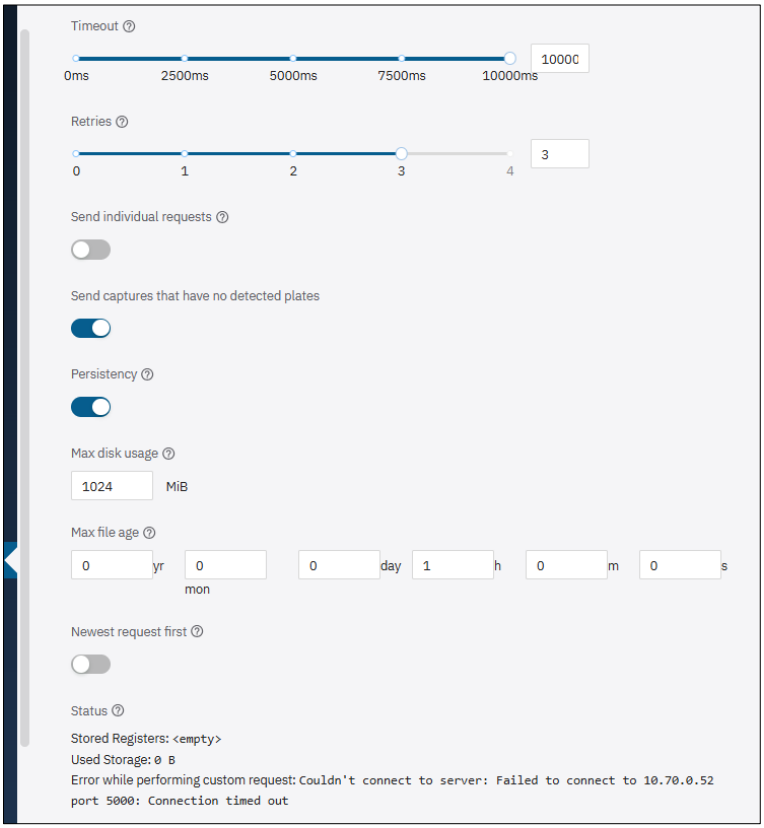

7) Select the *Body variant* request variant as *Singlepart*;
    a. Select the custom request *Body Type* (Content-Type header is added automatically) from the *JSON*, *JPEG* and *Form* (*URL encoding*) options;
    *b.* Check and edit the custom request *Body Data* (Content-Length header is added automatically) 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*;




8) 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*;
9) Select the *JPEG Quality* that is sent in the body between *Standard* or between 5% and 95%;
10) 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);



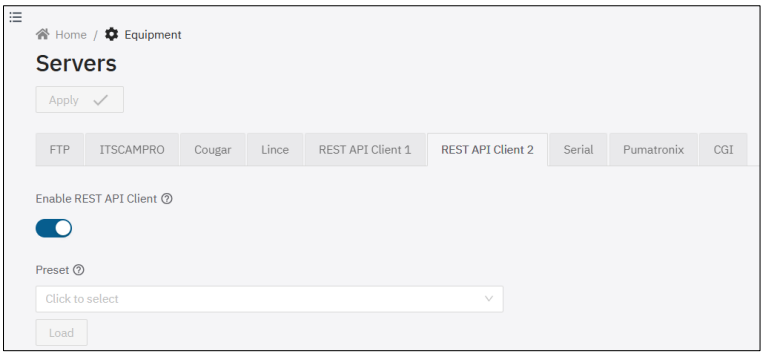

11) In *Timeout*, enter the time interval, in milliseconds, in which the custom request is canceled if there is no response from the server;
12) Indicate the number of *Retries* that are performed on the personalized request again in the event of a failure. Note that variable substitution errors do not count as failures;
13) 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;
14) 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;
15) Select *Persistency* to save information on disk when the request fails and try to send it again later;
16) Enter the amount in Mib of *Maximum disk usage* to persist failed requests;
17) Enter the *Maximum file age* of failed requests, considering that failed requests older than this value are discarded;
18) 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;
19) Check *Status* for information about the last custom request made by the REST API Client;
20) 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}}
}
```

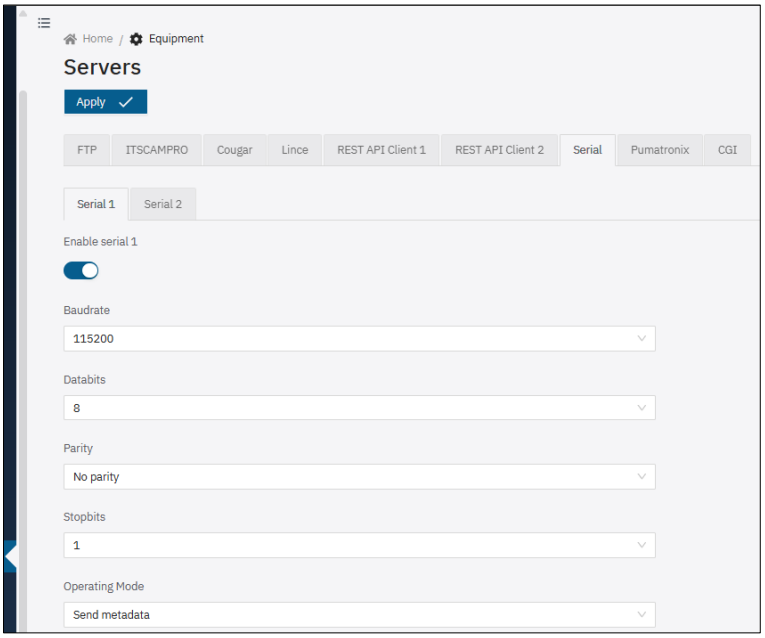
**Example of data sent:**

```
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",
```

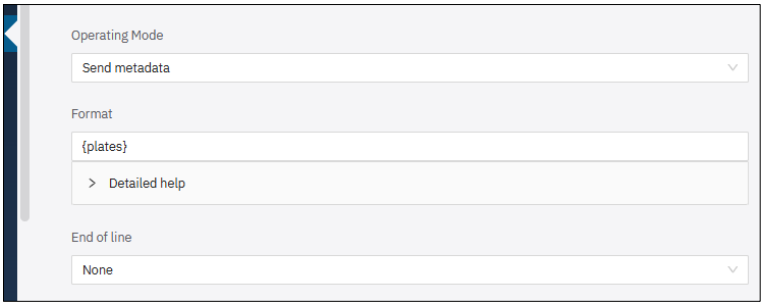
```
                    "vehicleModelProbability": 0.0,
                    "vehicleType": "car",
                    "vehicleTypeProbability": 0.0
                }
            }
    ],
    "plate": "CBH0599",
    "plateBoundingBox": [358, 182, 78, 23],
    "plateProbability": [1.0, 0.96, 0.98, 0.99, 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

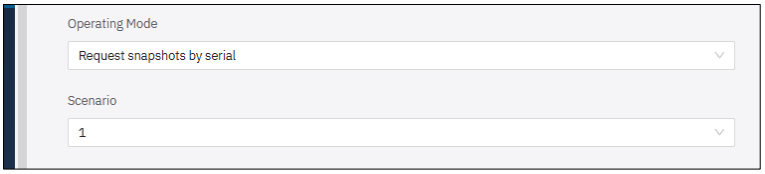
1) Select the *Serial* tab to configure the device's serial interfaces;
2) In the *Serial 1* tab, select the *Enable serial 1* option;
3) Select the *Baudrate* from the options: 4800, 9600, 19200, 38400, 57600 or 115200;
4) Select the *Databits* from the options: 5, 6, 7, 8 or 9;
5) Select *Parity* from 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*;



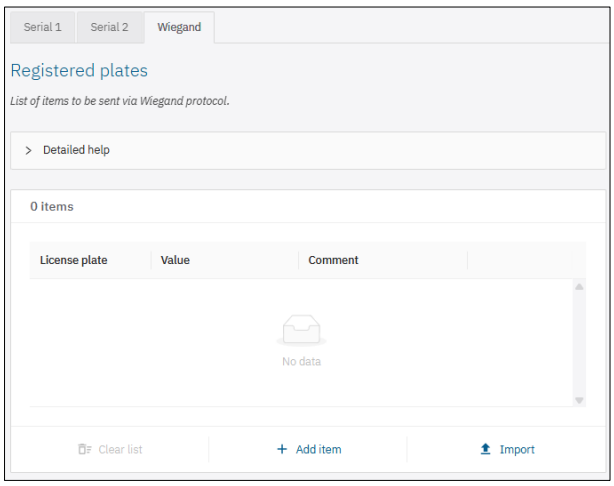

8) In *Operation Mode > Sending metadata*, indicate the *Format* by replacing the variables with variable names surrounded by double braces, taking into account the variables available in the *Detailed Help*;
    a. Select *End of line* in the options: None, <CR>(\r), <LF>(\n) or <CR><LF>(\r\n);



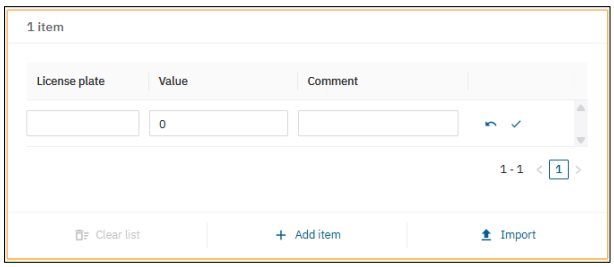

9) In Operation Mode > Request captures via serial, select Scenario from the options: No scenario, 1 or 2;



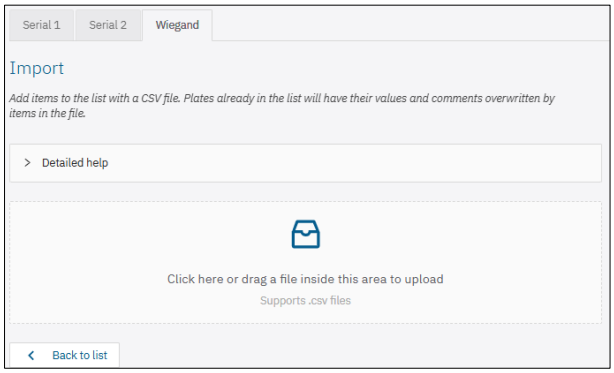

10) Select the *Serial 2* tab to configure the device's serial 2 interface;
11) Select the Wiegand tab to manage a list of boards and their corresponding codes, which will be transmitted via the serial interface after detection;
    a. Access the Detailed Help section to check the possible values that can be entered;




12) Click on + Add item to manually add the License Plate, Value, and Comment;



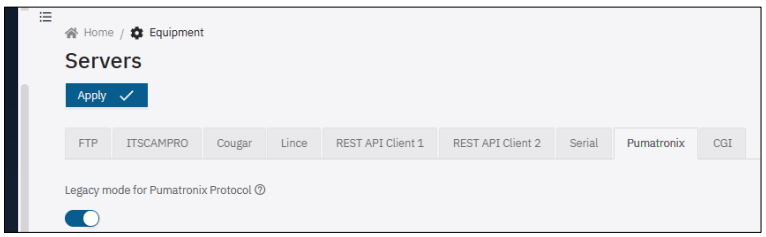

13) Click Import to upload data from a .CSV file;




14) Click Save after verifying the entered data.

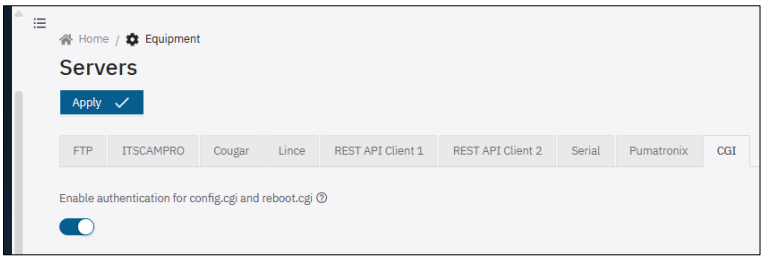
## 3.7.8. Pumatronix Server

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

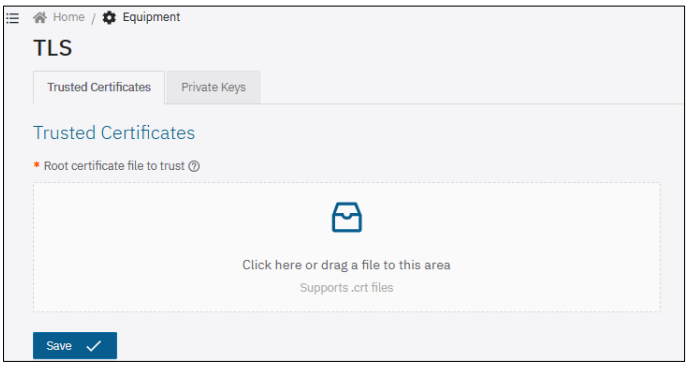
1) Select *Enable authentication for config.cgi and reboot.cgi* to protect access to *config.cgi* and *reboot.cgi* and they will require authentication for access. User and password for authentication are the same as for the web interface.
2) Click Save after verifying the entered data.



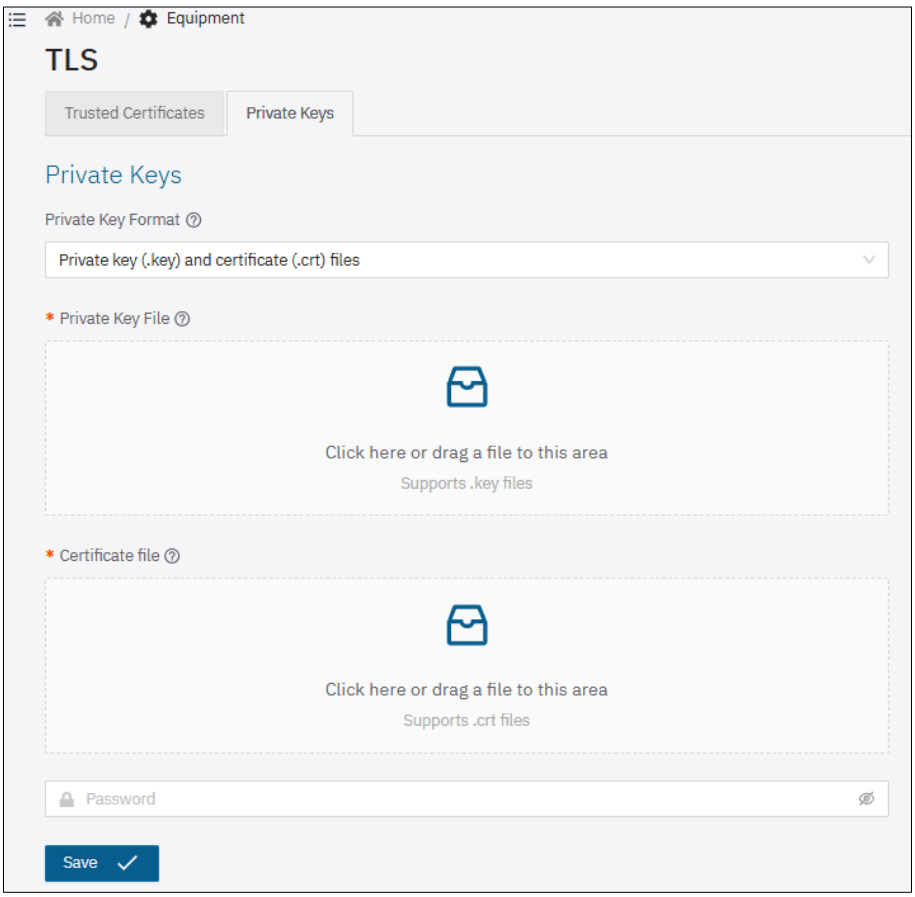

## 3.8. TLS

The devices can send data to a REST server using mutual TLS authentication (mTLS), loading certificates and private keys.

1) In the *Trusted Certificates* tab, click or drag a certificate file to trust in *.CRT format; (*Trusted certificates are used to validate self-signed certificates from remote servers so that a data transfer can be initialized)




2) In the Private Keys tab, select the Private Key Format from the options:
    a. *Private key files (.key) and certificate (.crt);*
        i. Click or drag a private key file in *.KEY* format and a certificate to trust in *.CRT* format into their respective fields;
    b. *Single .p12 file;*
        i. Click or drag a PKCS12 file in *.p12 format;*
3) Define a Password, which may contain between 4 and 200 characters, including numbers, letters, and special characters;




4) Click *Save* after checking the entered data

# 4. System Settings

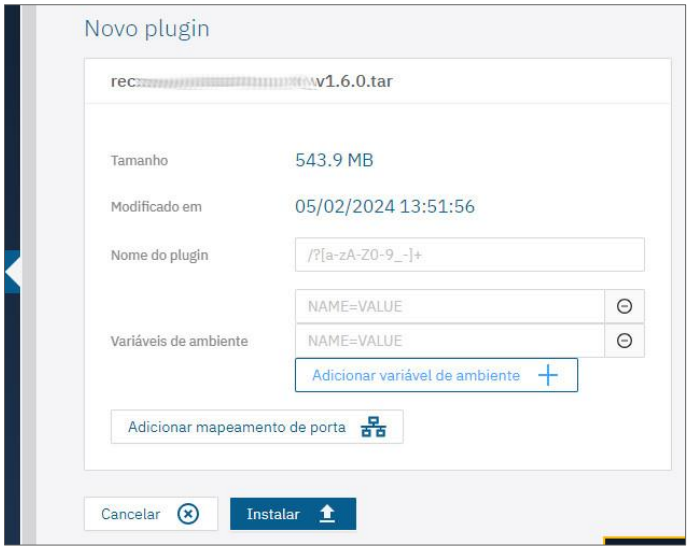
## 4.1. Plugins

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*;

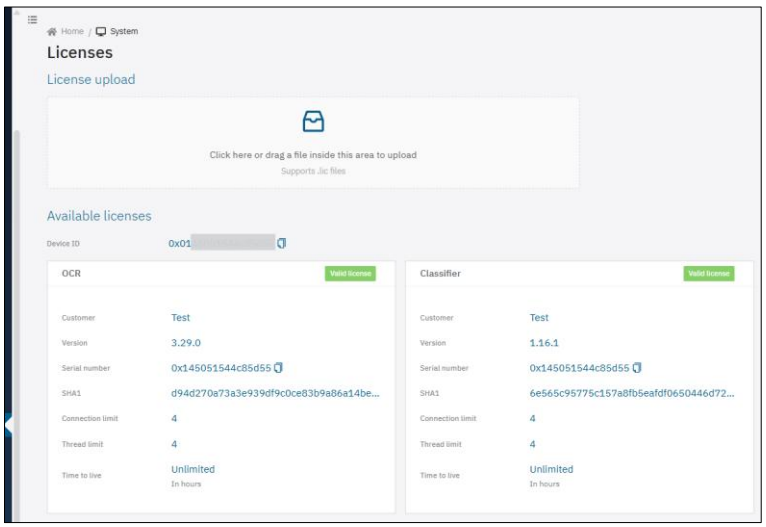

2) Click or drag a *.tar* file into the *New Plugin* area;
3) Enter the *Name of the plugin* that describes its use;
4) Configure the plugin's *Environment variables* by clicking on *Add environment variable*;
5) Click *Add port mapping* when an internal container port needs to be exposed on the device,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



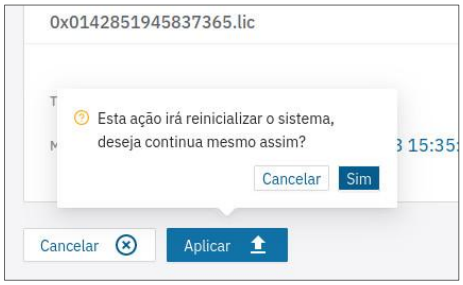

## 4.2. Licenses

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*:

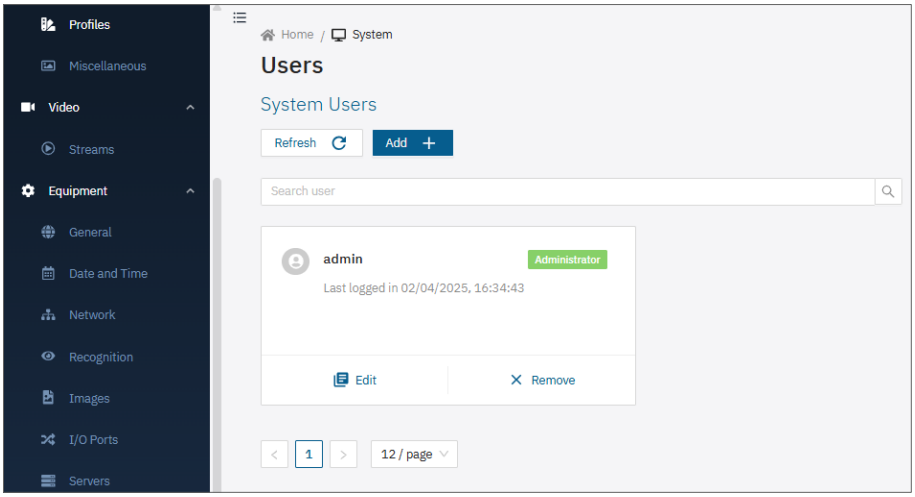
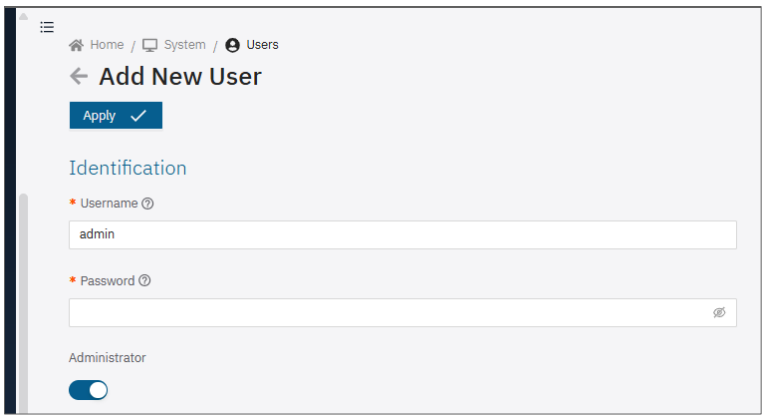



*Figure 19 - User management home screen*

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*.

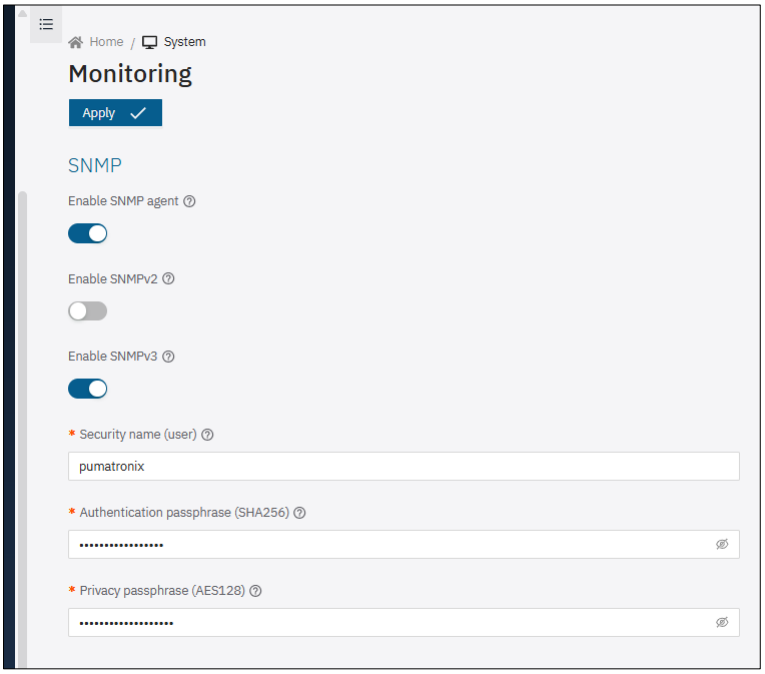



**The factory default user and password should be changed for better access control and greater security.**

## 4.4. Monitoring

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

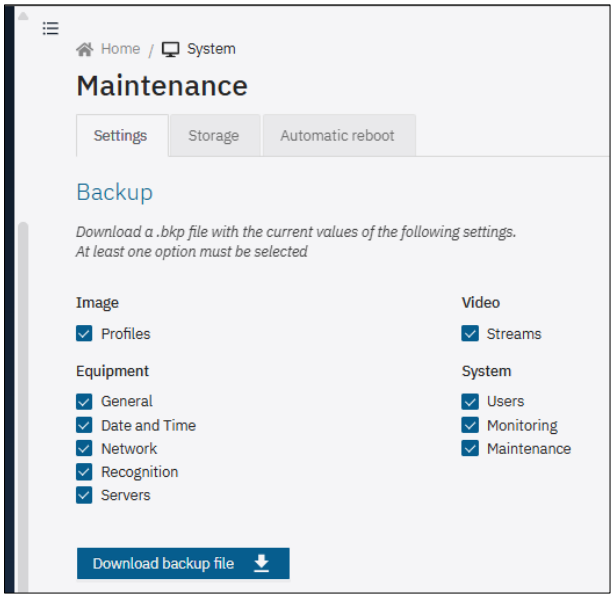
1) Go to the System > Monitoring menu;
    a. Select the *Enable SNMP agent* option to enable the integrated SNMP agent;
    b. Enable version 3 of the SNMP protocol in the *Enable SNMPv3* option:
        i. Define a *Security Name (user)*, also called "user". The context name is an empty string (mandatory 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

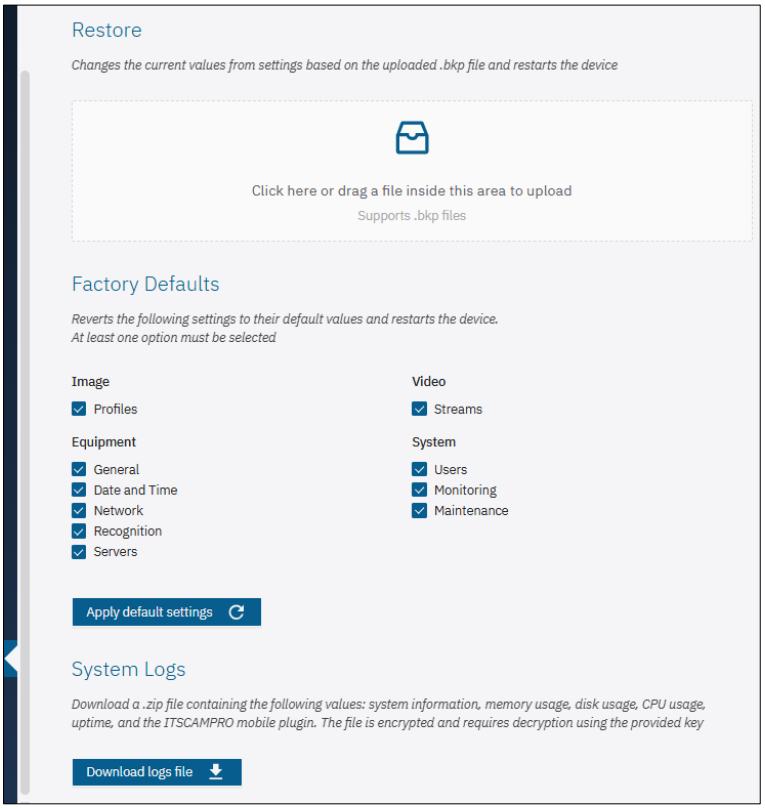
The *Backup*, *Restore*, *Factory Defaults and Automatic Reboot* maintenance options are available via the *System > Maintenance* menu.

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;

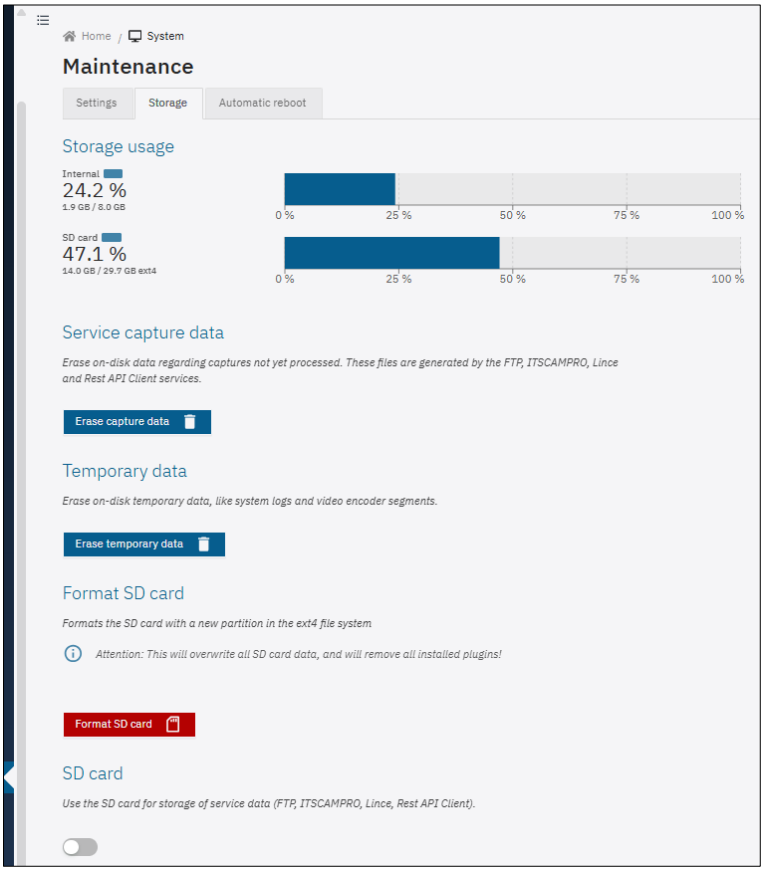
3) Locate the *Factory Defaults* field to restore the factory settings in the event of a device malfunction or incorrect settings;
    a. Select the settings that will be reset to the factory default;
    b. Click on *Apply default settings*;
    c. Wait for the device to reboot;




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:

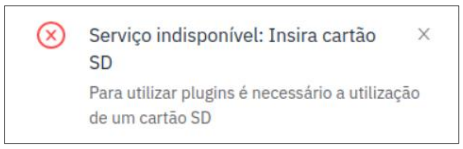
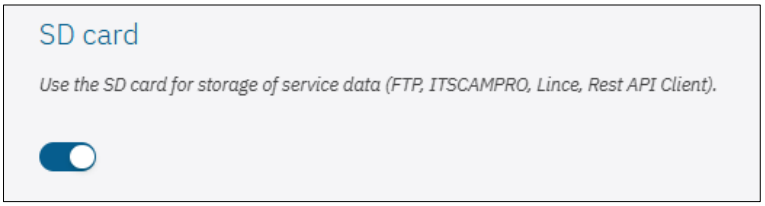



*Figure 20 – Error message displayed if SD card is not present*

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, including the capture data stored on the SD card. 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).

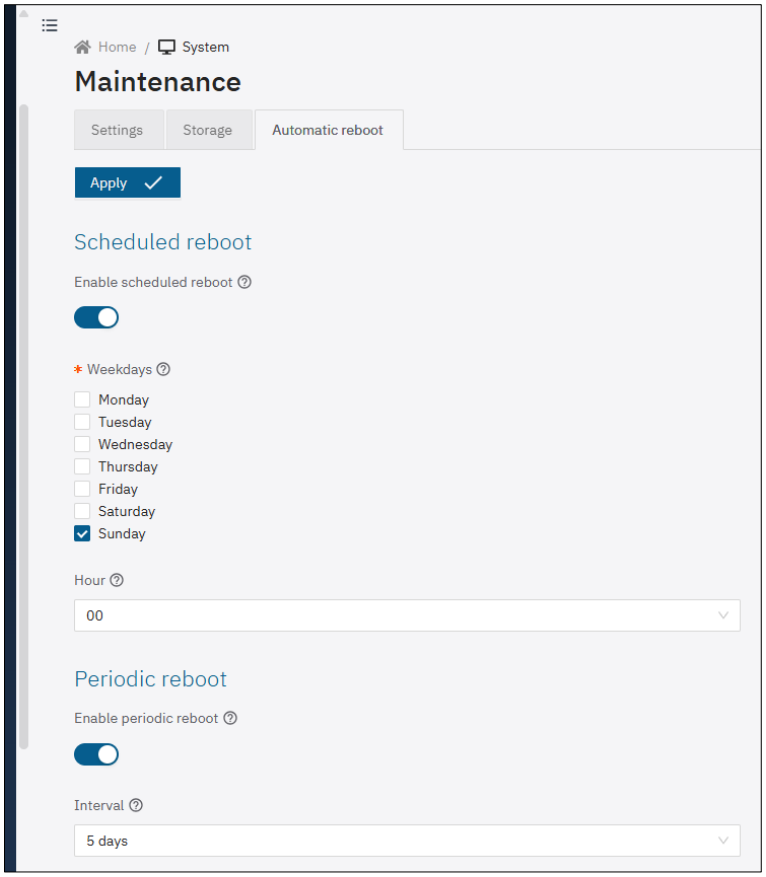



## 4.5.2. Automatic Reboot

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.

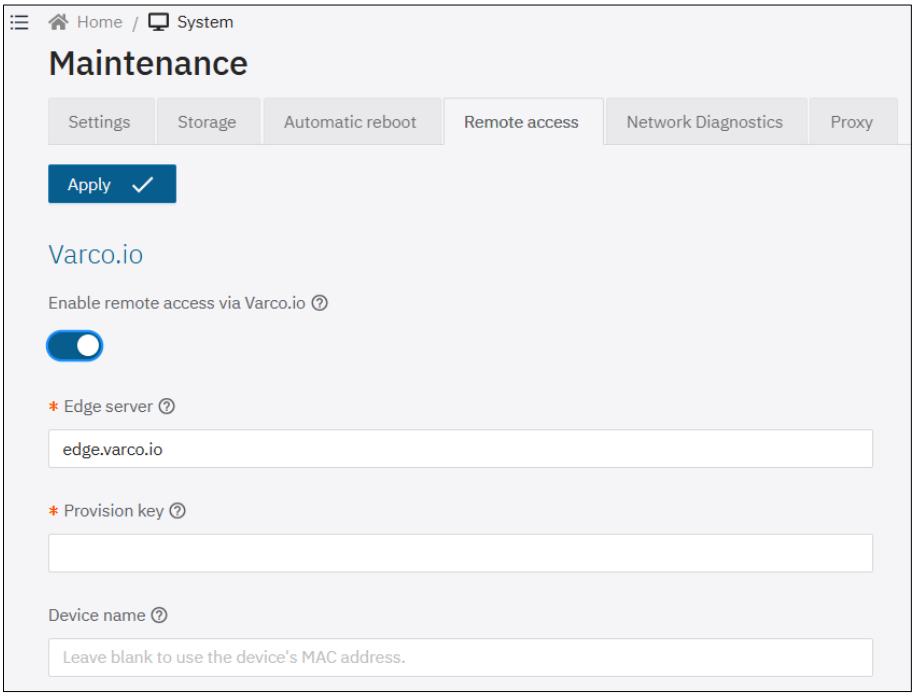



## 4.5.3. Remote access

The device can connect remotely through varco.io.

Varco.io is a remote access platform for cameras. To use it, you must have a Varco.io account and a provisioning key.

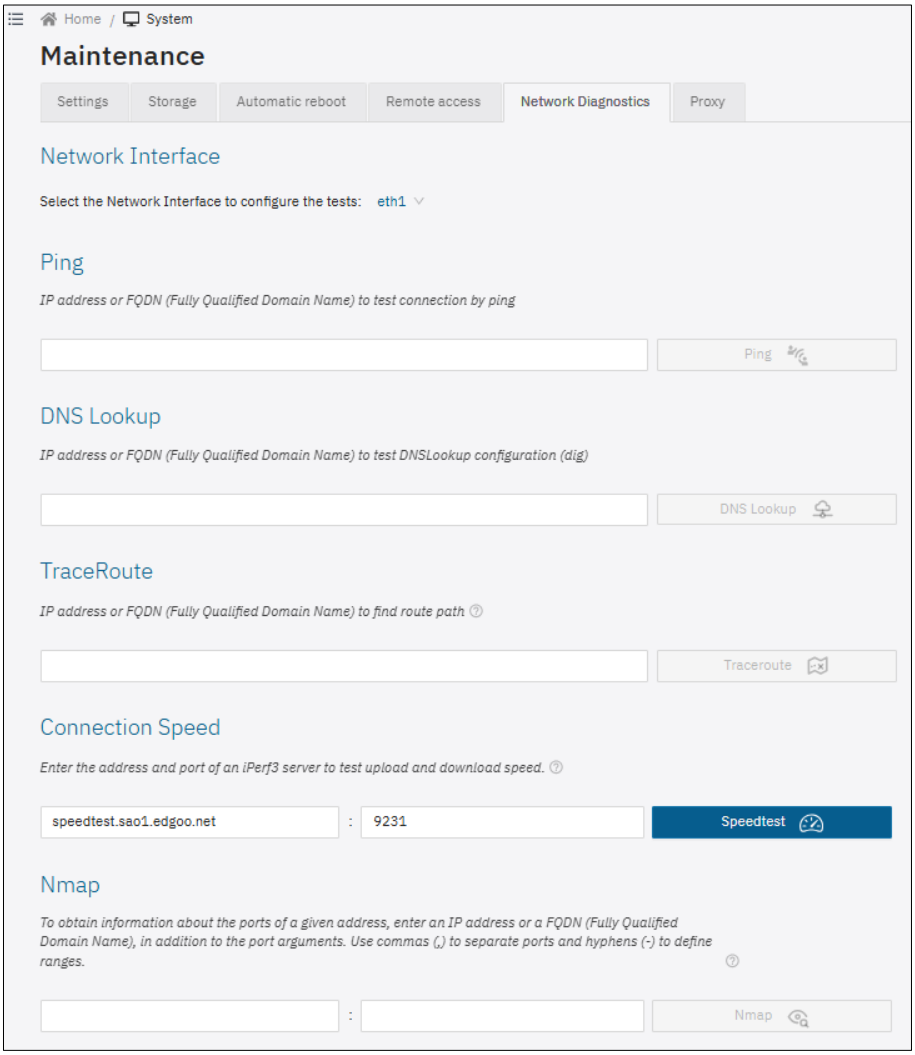
1) Enable the *Enable remote access via Varco.io* option to connect to Varco.io;
2) Enter the *Edge Server* to be used by Varco.io;
3) Enter the *Provisioning Key* to be used by Varco.io;
4) Define the *Device Name* to be displayed in Varco.io. Leave it blank to use the device's MAC address;
5) Click Apply to confirm the entered data.




## 4.5.4. Network Diagnostics

In the *Network Diagnostics* tab, several tools are included to help troubleshoot network issues.

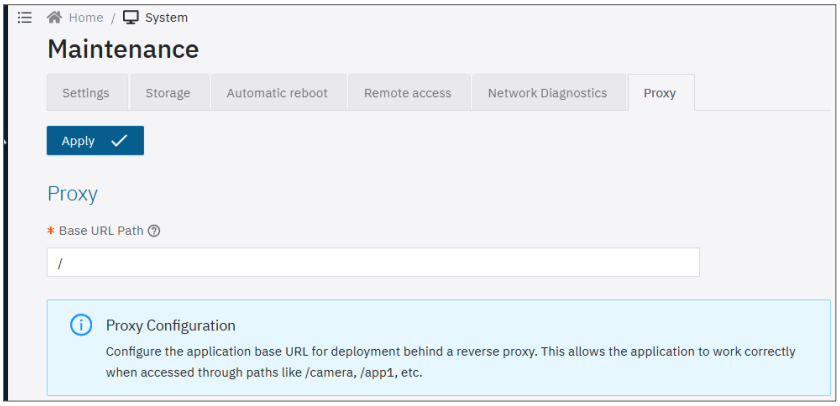
1) Select the Network Interface on which the tests will be performed, from the options:
    a. eth1;
    b. wifi
2) Enter an IP Address or FQDN (Fully Qualified Domain Name) to test connection via ping;
3) Enter an IP Address or FQDN (Fully Qualified Domain Name) to test the DNSLookup (dig) configuration;
4) Enter an IP Address or FQDN (Fully Qualified Domain Name) to trace the route path. In case of failure, some possible errors are:
    a. !A: Destination network administratively prohibited
    b. !F: Fragmentation needed
    c. !H: Host unreachable
    d. !N: Network unreachable
    e. !P: Protocol unreachable
    f. !S: Source route failed
    g. !X: Administratively prohibited
5) In *Connection Speed*, enter the address and port of an *iPerf3* server to test upload and download speed. (For this, an iPerf3 server is required as a reference point. You can start a local server by running the command: iperf3 -s);
6) In *Nmap*, to obtain information about the ports of a given address, enter an IP address or an FQDN (Fully Qualified Domain Name), in addition to the port arguments. Use commas (,) to separate ports and hyphens (-) to define ranges. (For this, you must provide an IP address or domain name (FQDN) and the ports you want to scan.)




## 4.5.5. Proxy

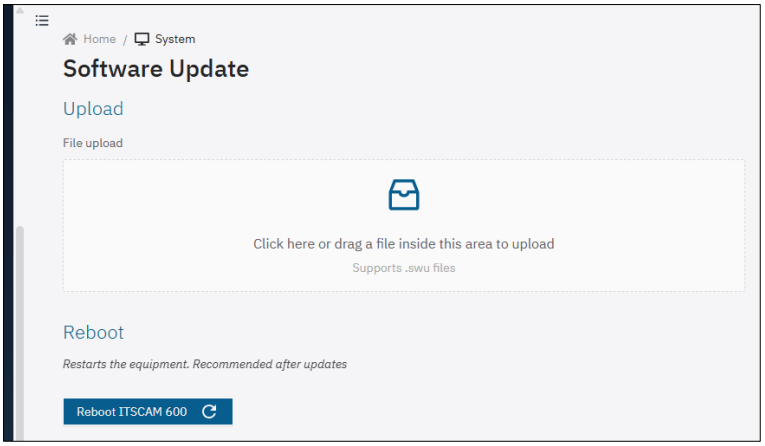
In the *Proxy* tab, you can change the webapp path so that an external proxy can be used to access different capture devices behind the same address.

7) Enter the Base URL Path where the application is accessed (e.g., / for root, /camera for path with proxy);
    a. Configure the base URL of the application for deployment behind a reverse proxy. This allows the application to function correctly when accessed through paths such as /camera, /app1, etc.



## 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 |
| | 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.
- "imgpkg"
    - "embedexif" *[bool, false]*: Inserts EXIF data into the image.
    - "embedcomments" *[bool, false]*: Insert the image metadata into the image's "comments" field.

- "embedsign" *[bool, false]*: Insert the signature into the images, i.e., include the tags in the metadata: "ExpoenteRSA", "ModuloRSA", "Sha256", and "Sign".

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. Since no image processing has been performed at this stage, 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.
- "tags" [json]: includes all tags already present in the comments of images obtained by traditional snapshot or trigger methods, but to include signature attributes, it is necessary to use IMGPKG_* specifically configured to accept the signature: SET_JPEG_CFGS.

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.
- "DatetimeUTCMs" *[string]*: Date and time (in UTC, considering the time zone as 0) when the metadata was sent. ("YYYY-MM-DD hh:mm:ss.000")
- "uptimeMs" *[int]*: Time, in milliseconds, since the last startup. It has the same time source as the "TempoLigado" metadata in the image comments.

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, however, 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)

**This protocol is only available for ease of migration from the 400 to the 450/600 line, and has been integrated within the compatibility limitations between models. Priority should be given to the Cougar Protocol when using it for development.**

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.*

The description of the commands applied to the ITSCAM 600 and ITSCAM 450 family devices are described according to their compatibility with the original commands of the Dynamic Library (itscam.dll).

## 7.1. Dynamic Library Error Codes

In most dynamic library functions, an integer value is returned indicating success or error in execution. The list of error values can be found in the *Error Code Table* of the ITSCAM Dynamic SDK Library. Successful execution of the functions is represented by a positive return. However, if the return is not listed in the error codes, please contact Technical Support.

## 7.2. Accepted and Fully Compatible Commands

### 7.2.1. broadcast( lista : char *, tamanhoLista : int ) : void

| |
|---|
| Returns the IP addresses of the ITSCAMs on the network (IP of the connection on ETH0) |

| | |
|---|---|
| lista | Memory region that starts the list |
| tamanhoLista | Size of the memory block that will contain the list |

### 7.2.2. criarConexaoItscam( ip : const char * ) : int

| |
|---|
| Establishes a connection with an ITSCAM connected to the network. The return of a value other than zero does not mean that the connection to ITSCAM has been successfully established.<br>This means that the identifier has been created. To determine whether the connection has been established correctly, use the *leStatus* function. |

| | |
|---|---|
| ip | ITSCAM IP address |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Connection not established |
| 1 to 64 | Connection ID |

### 7.2.3. criarConexaoItscamTimeout( ip : const char *, timeout : int ) : int

| |
|---|
| Establishes a connection with an ITSCAM connected to the network, specifying a timeout in seconds. The return of a value other than zero does not mean that the connection to ITSCAM has been successfully established. This means that the identifier has been created. To determine whether the connection has been established correctly, use the *leStatus* function. |

| | |
|---|---|
| ip | ITSCAM IP address |
| timeout | Socket connection timeout (in seconds) |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Connection not established |
| 1 to 64 | Connection ID |

## 7.2.4. fecharConexaoItscam( id : int ) : int

| |
|---|
| Closes a specific connection to an ITSCAM on the network. The identifier becomes invalid after calling this function and should no longer be used. |

| | |
|---|---|
| id | ITSCAM connection index |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.5. leAutoFoco( id : int ) : int

| |
|---|
| Returns the autofocus status of the ITSCAM. The autofocus function automatically changes the focus when the zoom is modified. |

| | |
|---|---|
| id | ITSCAM connection index |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Disabled |
| 1 | Enabled |

## 7.2.6. leAutoIris( id : int ) : int

| |
|---|
| Returns the value of the ITSCAM auto iris lens setting when in Day profile. In Night mode, the iris is always kept open. |

| | |
|---|---|
| id | ITSCAM connection index |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | ITSCAM without DC auto iris lens |
| 1 | ITSCAM with DC auto iris lens |

## 7.2.7. leBalancoBranco( id : int, bb : int* ) : int

| |
|---|
| Returns the ITSCAM white balance value. |

| | |
|---|---|
| id | ITSCAM connection index |
| bb | Vector with three 32-bit integer values, the first corresponding to red, the second to green, and the third to blue. |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.8. leDataAtual( id : int ) : int

| |
|---|
| Returns the current date from ITSCAM. |

| | |
|---|---|
| id | ITSCAM connection index |

| | |
|---|---|
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | ITSCAM without the clock module |
| Positive value | Date in DDMMYY format |

## 7.2.9. leFocoInfraVermelho( id : int ) : int

| Description | |
|---|---|
| Motorized lenses do not have infrared correction, so separate focusing must be performed for visible light and infrared light. This function returns the type of focus currently being used. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Current focus for visible light (infrared filter enabled) |
| 1 | Current focus for infrared light (infrared filter disabled) |

## 7.2.10. leGamma( id : int ) : int

| Description | |
|---|---|
| Returns the gamma value of ITSCAM. Gamma is an operation performed on the image that produces a logarithmic response to pixel intensity, with the aim of improving contrast and clarity. It can be useful for enhancing images in the shadows.<br>For devices in the ITSCAM 450 and ITSCAM 600 families, the gamma values are indicated for the active profile. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Gamma disabled and linear image sensor response |
| 1 | Gamma enabled and standard logarithmic response of the image sensor |
| 70 to 255 | Gamma enabled with logarithmic curve according to the returned value |

## 7.2.11. leGateway( id : int, gateway : char* ) : int

| Description | |
|---|---|
| Returns the ITSCAM gateway. | |
| **Parameters** | |
| id | ITSCAM connection index |
| gateway | Character vector in which the ITSCAM gateway of the specified connection will be filled |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.12. leHoraAtual( id : int ) : int

| Description | |
|---|---|
| Returns the current time of ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| -1 | ITSCAM without the clock module |
| Positive value | Time in HHMMSS format |

## 7.2.13. leIp( id : int, ip : char* ) : int

| Description | |
|---|---|
| Returns the IP address of the ITSCAM that is connected to the specified ID. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ip | Character vector in which the ITSCAM IP address of the specified connection will be filled in * |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

* Examples of using the connection IP address read function can be found in the *SDK Dynamic Library*.

## 7.2.14. leMac( id : int, mac : char* ) : int

| Description | |
|---|---|
| Returns the MAC address of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| mac | Character vector in which the MAC address of the ITSCAM of the specified connection will be filled in (with a minimum size of 18 characters) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.15. leMascaraRede( id : int, mascara : char* ) : int

| Description | |
|---|---|
| Returns the network mask of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| mascara | Character vector in which the ITSCAM network mask of the specified connection will be filled in |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.16. leModelo( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM model. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Value from 1 to 524699 | 441 = ITSCAM 450 or ITSCAM 600 |

## 7.2.17. leModoDayNight( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM Day/Night operating mode setting. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Automatic Transition (switches between Day and Night mode depending on brightness) |
| 1 | Profile 0 active (Day, default) |
| 2 | Another active profile |

## 7.2.18. leModoOCR( id : int ) : int

| Description | |
|---|---|
| Returns the OCR mode of ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | OCR disabled |
| 1 | OCR in fast mode (faster execution and lower recognition) |
| 2 | OCR in normal mode |
| 3 | OCR in slow mode |
| 4 | OCR in very slow mode (longer execution time and higher recognition rate) |

## 7.2.19. leNivelDesejado( id : int ) : int

| Description | |
|---|---|
| Returns the desired brightness level for the ITSCAM image. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 7 to 62 | Desired level value |

## 7.2.20. leNumeroFotos( id : int ) : int

| Description | |
|---|---|
| Returns the number of photos set per request via the network. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 to 16 | Varies depending on the ITSCAM model (refer to the Integration Manual for limits) |

## 7.2.21. leNumeroFotosIO( id : int ) : int

| Description | |
|---|---|
| Returns the number of photos set per request via I/O. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 to 16 | Varies depending on the ITSCAM model (refer to the Integration Manual for limits) |

## 7.2.22. lePortaServidor( id : int ) : int

| Description | |
|---|---|
| Returns the port set in ITSCAM. The system may have several active servers, but it only returns the port of the server set by the *setaTipoServidor* command. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| 1 to 65535 | Port |

## 7.2.23. lePosicaoFoco( id : int ) : int

| Description | |
|---|---|
| Returns the absolute position of the focus if ITSCAM has a motorized lens. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Positive or negative value | Motorized lens focus position |

## 7.2.24. lePosicaoZoom( id : int ) : int

| Description | |
|---|---|
| Returns the absolute position of the zoom if ITSCAM has a motorized lens. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Positive or negative value | Motorized lens zoom position |

## 7.2.25. leQualidadeFotoIO( id : int ) : int

| Description | |
|---|---|
| Returns the quality of JPEG photos when requested via I/O. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 100 | Quality when the photo format used is JPEG. |

## 7.2.26. leRevisaoFirmware( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM firmware revision. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | ITSCAM firmware revision |

## 7.2.27. leRotacao( id : int ) : int

| Description | |
|---|---|
| Returns the value of the ITSCAM rotation setting. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Normal image |
| 1 | Image rotated 180° |

## 7.2.28. leSituacaoDayNight( id : int ) : int

| Description | |
|---|---|
| Returns the current status of the ITSCAM Day/Night profile, according to the active profile. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Night profile (or any profile other than Day) |
| 1 | Day Profile |

## 7.2.29. leStatus( id : int ) : int

| Description | |
|---|---|
| Returns the status of the connection with ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Connection OK |

## 7.2.30. leTipoGanho( id : int ) : int

| Description | |
|---|---|
| Returns the status of the connection with ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Fixed gain |
| 1 | Automatic Gain |

## 7.2.31. leTipoServidor( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM server settings. The system may have several active servers, but it returns only one of them. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | None |
| 1 | FTP |
| 2 | ITSCAMPRO |

## 7.2.32. leTipoShutter( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM shutter settings. Value valid only after definition using the *setaTipoShutter* function. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Fixed shutter on both profiles |
| 1 | Automatic shutter on both profiles |
| 2 | Fixed shutter in Day mode and automatic shutter in Night mode |

## 7.2.33. leValorEntrada( id : int ) : int

| Description | |
|---|---|
| Returns the value of inputs 2 and 4, when set as inputs, on the ITSCAM 600. In ITSCAM 450, it returns the value of inputs 1 and 2. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Inputs disabled |
| 1 | Only first input activated |
| 2 | Only second input activated |
| 3 | Both inputs activated |

## 7.2.34. leValorNivel( id : int ) : int

| Description | |
|---|---|
| Returns the current brightness level of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 62 | Current level value |

## 7.2.35. leVersaoFirmware( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM firmware version with the value referring only to the first two fields of the firmware. Example: versions v1.7.7 and v1.7.6 return the same value. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Firmware version |

## 7.2.36. requisitaFotoId( id : int, slot : int, idFoto : int, buf : unsigned char*, res : int ) : int

| Description | |
|---|---|
| Retrieves a photo from the identifier returned by *requisitaMultiplasFotosSemEspera* or *requisitaMultiplasFotosIOSemEspera*.. | |
| **Parameters** | |
| id | ITSCAM connection index |
| Slot | Package identifier, returned by the *requisitaMultiplasFotosSemEspera* or *requisitaMultiplasFotosIOSemEspera* function. |
| idFoto | Photo of the desired sequence, starting at 0 |
| buf | Pre-allocated byte vector, large enough to contain the image. For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| res | Reserved parameter, always send 0 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.2.37. setaBalancoBranco( id : int, bb : int*) : int

| Description | |
|---|---|
| Assigns the ITSCAM white balance value. | |
| **Parameters** | |
| id | ITSCAM connection index |
| bb | Vector with three integers representing the weight of the red, green, and blue components (in that order). Values range from 0 (automatic adjustment) to 255. |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.38. setaBalancoBrancoAlternativo( id : int, tipo : int ) : int

| Description | |
|---|---|
| Defines whether ITSCAM will use a different white balance value for Day and Night modes. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: Same white balance for Day and Night mode<br>1: Alternative/differentiated white balance enabled for Day mode<br>2: Alternative/differentiated white balance enabled for Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.39. setaFoco( id : int, foco : int ) : int

| Description | |
|---|---|
| In ITSCAMs with motorized lenses, it moves the lens focus. | |
| **Parameters** | |
| id | ITSCAM connection index |
| foco | 1 to 999: Move the lens to focus on objects at infinity<br>1000: Does not move the lens<br>1001 to 1999: Move the lens to focus on nearby objects |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.40. setaFotoColorida( id : int, cor : int ) : int

| Description | |
|---|---|
| Set ITSCAM to produce color photos in Night mode. Generating color photos in Night mode is not the same as forcing ITSCAM to always operate in Day mode. | |
| **Parameters** | |
| id | ITSCAM connection index |
| cor | 0: grayscale photos in Night mode; 1: color photos in Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.41. setaGamma( id : int, gamma : int ) : int

| Description | |
|---|---|
| Assigns the ITSCAM gamma value to profiles according to the setaGammaAlternativo selection. Gamma is an operation performed on the image that produces a logarithmic response to pixel intensity, with the aim of improving contrast and clarity. It can be useful for enhancing images in the shadows. | |
| **Parameters** | |
| id | ITSCAM connection index |
| gamma | 0: Gamma disabled and linear image sensor response<br>1: Gamma enabled and standard logarithmic response of the image sensor<br>70 to 255: Gamma enabled with logarithmic curve according to the returned value |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.42. setaGammaAlternativo( id : int, tipo : int ) : int

| Description | |
|---|---|
| Defines whether ITSCAM will use an alternative/differentiated gamma value for Day and Night modes. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: Same gamma in Day and Night mode<br>1: Alternative/differentiated gamma enabled for Day mode<br>2: Alternative/differentiated gamma enabled for Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.43. setaGanhoAlternativo( id : int, tipo : int ) : int

| Description | |
|---|---|
| Assigns different maximum gains for Day and Night modes. When working with automatic gain, in some situations, it is important that the maximum gain in Day mode is different from the maximum gain in Night mode. ITSCAM has a filter between the lens and the sensor that cuts infrared rays, significantly reducing the amount of light captured by the sensor. In Night mode, there is no such filter, and infrared flash is generally used. As a result, the amount of light captured by the sensor is much greater, and the maximum gain in Night mode needs to be lower than that used in Day mode. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: Equal gain in Day and Night mode<br>1: Maximum alternative/differentiated gain enabled for Day mode<br>2: Alternative/differentiated maximum gain enabled for Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.44. setaGanhoSegundaFotoLuzInfravermelha( id : int, ganho : int ) : int

| Description | |
|---|---|
| Assigns the gain value of the second photo when the predominant light is infrared and the multiple photo request functions are being used. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ganho | 0 to 72 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.45. setaGanhoSegundaFotoLuzVisivel( id : int, ganho : int ) : int

| Description | |
|---|---|
| Assigns the gain value of the second photo when the predominant light is visible and the multiple photo request functions are being used. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ganho | 0 to 72 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.46. setaIntervaloTriggersPeriodicos( id : int, intervalo : int ) : int

| Description | |
|---|---|
| Assigns the time ITSCAM waits to perform a new capture, modifying the minimum time between triggers, when set with trigger 9: Periodical, 10: High Level or 11: Low Level. | |
| **Parameters** | |
| id | ITSCAM connection index |
| intervalo | Time (in minutes for periodic triggers, or in milliseconds for high or low levels) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.47. setaLenteAutoIris( id : int, autoiris : int ) : int

| Description | |
|---|---|
| Assigns the ITSCAM lens settings. Valid for Day profile, as Night always has Iris open. | |
| **Parameters** | |
| id | ITSCAM connection index |
| autoiris | 0: Lens without DC auto iris; 1: Lens with DC auto iris |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.48. setaLimiarDayNight( id : int, limiar : int ) : int

| Description | |
|---|---|
| Assigns the level value that ITSCAM should use for switching from Day to Night mode. This change occurs when the shutter and gain reach their maximum values and the level falls below the established threshold. | |
| **Parameters** | |
| id | ITSCAM connection index |
| limiar | 5 to 40: TSCAM level for input into Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.49. setaLimiarNightDay( id: int, limiar : int ) : int

| Description | |
|---|---|
| Assigns the percentage of the maximum shutter value that ITSCAM should display for changing from Night to Day mode. | |
| **Parameters** | |
| id | ITSCAM connection index |
| limiar | 0 to 100: Percentage of maximum ITSCAM shutter for input to Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.50. setaModoDayNight( id : int, daynight : int ) : int

| Description | |
|---|---|
| Assigns the Day/Night mode setting of the ITSCAM | |
| **Parameters** | |
| id | ITSCAM connection index |
| daynight | 0: Automatic Mode; 1: Day mode; 2: Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.51. setaModoOCR( id : int, ocr : int) : int

| Description | |
|---|---|
| Assigns the OCR mode of ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ocr | 0: OCR disabled<br>1: OCR in fast mode (faster execution and lower recognition)<br>2: OCR in normal mode<br>3: OCR in slow mode<br>4: OCR in very slow mode (longer execution time and higher recognition rate) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.52. setaNumeroFotos( id : int, num : int ) : int

| Description | |
|---|---|
| Assigns the number of photos to be captured to each request via the network. This function does not need to be called directly, as the photo request functions already call it internally. | |
| **Parameters** | |
| id | ITSCAM connection index |
| num | 1 to the limit of the ITSCAM model (refer to the Integration Manual to verify the value) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.53. setaNumeroFotosIO( id : int, num : int ) : int

| Description | |
|---|---|
| Assigns the number of photos to be captured to each request via I/O. This function does not need to be called directly, as the photo request functions already call it internally. | |
| **Parameters** | |
| id | ITSCAM connection index |
| num | 1 to the limit of the ITSCAM model (refer to the Integration Manual to verify the value) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.54. setaPosicaoFoco( id : int, foco : int ) : int

| Description | |
|---|---|
| In ITSCAMs with motorized lenses, it changes the absolute focus position to the one specified in this function. Absolute position is measured relative to a fixed reference. | |
| **Parameters** | |
| id | ITSCAM connection index |
| foco | -1000 to 1000: Absolute focus position |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.55. setaPosicaoZoom( id : int, zoom : int ) : int

| Description | |
|---|---|
| In ITSCAMs with motorized lenses, it changes the absolute zoom position to the one specified in this function. Absolute position is measured relative to a fixed reference. | |
| **Parameters** | |
| id | ITSCAM connection index |
| zoom | -1000 to 2000: Absolute zoom position |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.56. setaQualidadeFotoIO( id : int, qualidade : int ) : int

| Description | |
|---|---|
| Assigns the quality of the JPEG photo acquired by ITSCAM I/O. | |
| **Parameters** | |
| id | ITSCAM connection index |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.57. setaRotacao( id : int, rotacao180 : int ) : int

| Description | |
|---|---|
| Assigns 180° rotation in ITSCAMs with 752x480 pixel resolution. | |
| **Parameters** | |
| id | ITSCAM connection index |
| Rotacao180 | 0: Does not rotate the image 180°<br>1: Rotate the image 180° |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.58. setaTempoEntreTriggers( id : int, tempo : int ) : int

| Description | |
|---|---|
| Assigns the minimum time between two I/O pulses for the second pulse to be considered valid (in milliseconds) by modifying the minimum time between triggers. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tempo | 0 to 60,000 (milliseconds) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.59. setaTipoGanho( id : int, automatic : int) : int

| Description | |
|---|---|
| Assigns the ITSCAM gain type relative to all profiles. | |
| **Parameters** | |
| id | ITSCAM connection index |
| automatic | 0: Fixed gain<br>1: Automatic Gain |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.60. setaTipoShutter( id : int, tipo : int ) : int

| Description | |
|---|---|
| Assigns the ITSCAM shutter type. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: Fixed Shutter relative to the active profile<br>1: Automatic Shutter relative to the active profile<br>2: Fixed Shutter in Day mode and Automatic in Night mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.61. setaTriggerAlternativo( id : int, tipo : int ) : int

| Description | |
|---|---|
| Assigns the status of the alternative/differentiated trigger of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: Same trigger in Visible and Infrared mode<br>1: Alternative/differentiated trigger enabled for Visible mode<br>2: Alternative/differentiated trigger enabled for Infrared mode |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.62. setaValorBalancoBrancoAlternativo( id : int, bb : int* ) : int

| Description | |
|---|---|
| Assigns the values for the alternative/differentiated white balance. | |
| **Parameters** | |
| id | ITSCAM connection index |
| bb | Vector with three 32-bit integer values, the first corresponding to red, the second to green, and the third to blue. For each component, it is possible to assign a value from 0 (ITSCAM controls the values) to 255. |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.63. setaValorGammaAlternativo( id : int, gamma : int ) : int

| Description | |
|---|---|
| Assigns the alternative/differentiated gamma value. | |
| **Parameters** | |
| id | ITSCAM connection index |
| Gamma | 0, 1, or from 70 to 255 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.64. setaValorGanhoAlternativo( id : int, ganho : int ) : int

| Description | |
|---|---|
| Assigns the alternative/differentiated gain value. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ganho | 0 to 72 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.2.65. setaValorSaida( id : int, saida : int ) : int

| Description | |
|---|---|
| Assigns the value of the output signal (pins 3, 4, 7, and 8 of the rear connector) of the ITSCAM. This function only has an effect for output 1 (OUT1) if it is set for the I/O type. | |
| **Parameters** | |
| id | ITSCAM connection index |
| saida | 0: Outputs disabled<br>1: Output 1 enabled and 2 disabled<br>2: Output 1 disabled and 2 enabled<br>3: Both outputs enabled |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

### 7.2.66. setaValorTriggerAlternativo( id : int, trigger : int ) : int

| Description | |
|---|---|
| Assigns the settings of the alternative/differentiated trigger of ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| trigger | As described in the *leTrigger* function |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

### 7.2.67. setaZoom( id : int, zoom : int ) : int

| Description | |
|---|---|
| On ITSCAMs with motorized lenses, it moves the lens zoom. | |
| **Parameters** | |
| id | ITSCAM connection index |
| zoom | 1 to 999: Move the zoom to open the field of view<br>1000: Does not move the lens<br>1001 to 1999: Move the zoom to narrow the field of view |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3. Commands Accepted with Adaptations

### 7.3.1. leBrilho( id : int ) : int

| Description | |
|---|---|
| Returns the brightness (or black level) value of ITSCAM relative to the active profile. The read range from 0 to 255 corresponds linearly to the minimum and maximum limits of the device (-100 to 100%). | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 255 | Brightness |

### 7.3.2. leContraste( id : int ) : int

| Description | |
|---|---|
| Returns the contrast value (or digital gain) of the ITSCAM relative to the active profile. The read range from 0 to 255 corresponds linearly to the minimum and maximum limits of the device (-100 to 100%). | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 255 | Contrast |

### 7.3.3. leFormatoFotoIO( id : int ) : int

| Description | |
|---|---|
| Returns the photo format of requests via I/O. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Photo submitted in JPEG format |

### 7.3.4. leFotoColorida( id : int ) : int

| Description | |
|---|---|
| Returns the status of colored photos in ITSCAM's Night Profile. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Grayscale photos in Night mode |
| 1 | Color photo in Night mode |

### 7.3.5. leGanhoFixo( id : int ) : int

| Description | |
|---|---|
| Returns the fixed gain value of ITSCAM relative to the active profile. The reading range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 72 | Gain value |

### 7.3.6. leGanhoMaximo( id : int ) : int

| Description | |
|---|---|
| Returns the maximum gain value of ITSCAM. The reading range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 72 | Maximum gain value |

### 7.3.7. leGanhoSegundaFotoLuzVisivel( id : int ) : int

| Description | |
|---|---|
| Returns the gain of the second photo relative to the active profile, when the predominant light is visible and the multiple photos per request functions are being used. The reading range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 72 | Gain value |

## 7.3.8. leSaturacao( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM color saturation value for the active profile. The read range from 0 to 255 corresponds linearly to the minimum and maximum limits of the device (-100 to 100%). | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 255 | Saturation |

## 7.3.9. leShutterFixo( id : int ) : int

| Description | |
|---|---|
| Returns the fixed shutter value of ITSCAM. The reading range from 0 to 2047 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Fixed shutter |

## 7.3.10. leShutterMaximo( id : int ) : int

| Description | |
|---|---|
| Returns the maximum shutter value of ITSCAM. The reading range from 0 to 2047 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Maximum Shutter |

## 7.3.11. leValorGanho( id : int ) : int

| Description | |
|---|---|
| Returns the current gain value of ITSCAM relative to the active profile. The reading range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 72 | Current gain value |

## 7.3.12. leValorOcrAlternativo( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM OCR mode to the differentiated settings. There is no differentiation in OCR settings for day and night. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | OCR disabled |
| 1 | OCR in fast mode (faster execution and lower recognition) |
| 2 | OCR in normal mode |
| 3 | OCR in slow mode |
| 4 | OCR in very slow mode (longer execution time and higher recognition rate) |

## 7.3.13. leValorShutter( id : int ) : int

| Description | |
|---|---|
| Returns the current shutter value of ITSCAM. The reading range from 0 to 2047 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 to the limit of the ITSCAM model | Each ITSCAM model has a shutter limit value. Refer to the Integration Manual for the value of the model used. |

## 7.3.14. requisitaFoto( id : int, buf : unsigned char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requests a photo from ITSCAM defined by the identifier ID. If ITSCAM is set to use flash, the photo will be synchronized with the shutter release. | |
| **Parameters** | |
| id | ITSCAM connection index |
| buf | Pre-allocated byte vector, large enough to contain the image (RGB*resolution+header). For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 Bytes (3*752*480+54), which corresponds to the size of a 752x480 pixel color BMP file. |
| formato | 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.15. requisitaFotoIO( id : int, foto : unsigned char*, formato : int, qualidade : int) : int

| Description | |
|---|---|
| Requests a photo from a trigger event and I/O. This function returns when the event occurs, or when the timeout is reached, whichever comes first. The request must be set as an exhibition only. | |
| **Parameters** | |
| id | ITSCAM connection index |
| photo | Vector with the size of the image in bytes |
| formato | 0: BMP image request<br>1: JPEG image request |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.16. requisitaFotoTriggerContinuo( id : int, buf : unsigned char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requests a photo generated by ITSCAM when the selected trigger type is continuous. If ITSCAM has embedded OCR and it is enabled, this function may take some time to return, as it will wait until a license plate is recognized or until the timeout is reached.<br>Requests a photo from a trigger event and I/O. The request must be set as an exhibition only. | |
| **Parameters** | |
| id | ITSCAM connection index |
| buf | Pre-allocated byte vector, large enough to contain the image (RGB*resolution+header). For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| formato | 0: BMP image request<br>1: JPEG image request |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.17. requisitaMultiplasFotos( id : int, buf : unsigned char**, nfotos : int, tamanho : int*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requires a sequence of photos. This function only returns when all images have been collected. If ITSCAM is set to use flash, the photos will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| buf | Vector of previously allocated byte vectors, each large enough to hold the image. For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| nfotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| tamanho | Vector with the size in bytes of each image |
| formato | 0: BMP 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.18. requisitaMultiplasFotosIO( id : int, foto : unsigned char**, nFotos : int, tamFotos : int*, formato : int, qualidade : int) : int

| Description | |
|---|---|
| Requests a sequence of photos from ITSCAM as a result of a trigger or I/O event, i.e., after calling this function, ITSCAM waits until an event occurs or until the timeout is reached. In the event of an I/O event, the function returns only after all photos have been captured.<br>If ITSCAM is set to use flash, the photos will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| foto | Vector of previously allocated byte vectors, each large enough to allocate the requested images. For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| nFotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| tamFotos | Vector with the sizes in bytes of the images created |
| formato | 0: BMP; 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.19. requisitaMultiplasFotosIOSemEspera( id : int, nfotos : int, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requests a sequence of photos via I/O or trigger to ITSCAM. This function returns when the event occurs or if the timeout value is reached. This identifier must be used later by the *requisitaFotoId* function to collect the photos individually.<br>If ITSCAM is set to use flash, the photos will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| nfotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| formato | 0: BMP image request<br>1: JPEG image request |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Photo package identifier. This identifier must be used later by the *requisitaFotoId* function to collect the photos individually. |

### 7.3.20. requisitaMultiplasFotosSemEspera( id : int, nfotos : int, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requires a sequence of photos. This function returns immediately. The return value is an identifier that must be used later by the *requisitaFotoId* function to collect the photos individually. If ITSCAM is set to use flash, the photos will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| nfotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| formato | 0: BMP 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Photo package identifier. This identifier must be used later by the *requisitaFotoId* function to collect the photos individually. |

### 7.3.21. requisitaQuadroVideo( id : int, buf : unsigned char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Requests a photo from ITSCAM defined by the identifier id without flash synchronization. | |
| **Parameters** | |
| id | ITSCAM connection index |
| buf | Pre-allocated byte vector, large enough to contain the image (RGB*resolution+header). For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| formato | 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.22. salvarFoto( id : int, arquivo : char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves a photo from ITSCAM to disk. If ITSCAM is set to use flash, the photo will be synchronized with the shutter release. | |
| **Parameters** | |
| id | ITSCAM connection index |
| arquivo | Character vector corresponding to the name of the file to be created |
| formato | 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.23. salvarFotoIO( id : int, arquivo : char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves an ITSCAM photo to disk after a trigger or I/O event occurs. The request must be set as an exhibition only. | |
| **Parameters** | |
| id | ITSCAM connection index |
| arquivo | Vector of byte vectors previously allocated, with sufficient size to allocate the requested images (RGB*resolution+header). For models with a resolution of 752x480 pixels, we recommend using a vector of at least 1,082,934 bytes, which corresponds to the size of a 752x480 pixel color BMP file. |
| formato | 0: BMP; 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.24. salvarFotoOcr( id : int, diretorio : char*, qualidade : int ) : int

| Description | |
|---|---|
| Saves a JPEG image from ITSCAM, placing the OCR information in the file name. | |
| **Parameters** | |
| id | ITSCAM connection index |
| diretorio | Directory where the photo will be saved. The file name has the following format: AAMMDD_HHmmSSsss_PPPPPPP.jpg, where AA = year, MM = month, DD = day, HH = hour, mm = minute, SS = second, sss = millisecond, PPPPPPP = vehicle license plate. |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.25. salvarFotoOcrIO( id : int, diretorio : char*, qualidade : int ) : int

| Description | |
|---|---|
| Saves a JPEG image from ITSCAM when a trigger or I/O event occurs. The request must be set as an exhibition only. This function does not work if the trigger type is continuous. Use salvarFotoOcrTriggerContinuo in this case. | |
| **Parameters** | |
| id | ITSCAM connection index |
| diretorio | Directory where the photo will be saved. The file name has the following format: AAMMDD_HHmmSSsss_PPPPPPP.jpg, where AA = year, MM = month, DD = day, HH = hour, mm = minute, SS = second, sss = millisecond, PPPPPPP = vehicle license plate. |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.26. salvarFotoOcrTriggerContinuo( id : int, diretorio : char*, qualidade : int ) : int

| Description | |
|---|---|
| Saves a JPEG image from ITSCAM when the specified trigger or I/O event is continuous. The recognized card will be in the name of the file generated by ITSCAM. This function may take a while to return, as it waits until ITSCAM performs recognition, or until the timeout is reached. The request must be set as an exhibition only. | |
| **Parameters** | |
| id | ITSCAM connection index |
| diretorio | Directory where the photo will be saved. The file name has the following format: AAMMDD_HHmmSSsss_PPPPPPP.jpg, where AA = year, MM = month, DD = day, HH = hour, mm = minute, SS = second, sss = millisecond, PPPPPPP = vehicle license plate. |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

## 7.3.27. salvarFotoTriggerContinuo( id : int, arquivo : char*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves an ITSCAM photo to disk when the specified trigger or I/O event is continuous. If ITSCAM is set to use flash, the photo will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. The request must be set as an exhibition only. | |
| **Parameters** | |
| id | ITSCAM connection index |
| arquivo | Character vector corresponding to the name of the file to be created |
| formato | 0: BMP; 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.28. salvarMultiplasFotos( id : int, filename : char**, nFotos : int, tamFotos : int*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves a sequence of photos from ITSCAM to disk. If ITSCAM is set to use flash, the photo will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| filename | String vector with the names of the files to be created |
| nFotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| tamFotos | Vector with the sizes in bytes of each image created |
| formato | 0: BMP; 1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in Bytes |

### 7.3.29. salvarMultiplasFotosIO( id : int, filename : char**, nFotos : int, tamFotos : int*, formato : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves a sequence of ITSCAM photos to disk after a trigger or I/O event occurs. If ITSCAM is set to use flash, the photo will be synchronized with the shutter release. Check the technical specifications of the flash to ensure that it can meet all requirements. | |
| **Parameters** | |
| id | ITSCAM connection index |
| filename | String vector with the names of the files to be created |
| nFotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| tamFotos | Vector with the sizes in bytes of the images created |
| formato | 0: BMP<br>1: JPEG |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.30. salvarMultiplasFotosOcr( id : int, diretorio : char*, nFotos : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves a sequence of JPEG images from ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| diretorio | Directory where the photos will be saved. The file name has the following format: AAMMDD_HHmmSSsss_ii_PPPPPPP.jpg, where AA = year, MM = month, DD = day, HH = hour, mm = minute, SS = second, sss = millisecond, ii = photo index, PPPPPPP = vehicle license plate. |
| nFotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

### 7.3.31. salvarMultiplasFotosOcrIO( id : int, diretorio : char*, nFotos : int, qualidade : int ) : int

| Description | |
|---|---|
| Saves a sequence of JPEG images from ITSCAM when a trigger or I/O event occurs. This function does not work if the trigger type is continuous. Use *salvarFotoOcrTriggerContinuo* in this case. | |
| **Parameters** | |
| id | ITSCAM connection index |
| diretorio | Directory where the photos will be saved. The file name has the following format: AAMMDD_HHmmSSsss_ii_PPPPPPP.jpg, where AA = year, MM = month, DD = day, HH = hour, mm = minute, SS = second, sss = millisecond, ii = photo index, PPPPPPP = vehicle license plate. |
| nFotos | 1 to the maximum value of the ITSCAM model (refer to the Integration Manual for limits). |
| qualidade | 0 to 100 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| Positive value | Success – image size in bytes |

### 7.3.32. setaAutoFoco( id : int, foco : int ) : int

| Description | |
|---|---|
| Triggers an autofocus routine when requested. | |
| **Parameters** | |
| id | ITSCAM connection index |
| foco | 2: Performs ITSCAM autofocus (once) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

### 7.3.33. setaBrilho( id : int, brilho : int ) : int

| Description | |
|---|---|
| Assigns the brightness or black level of photos generated by ITSCAM relative to the active profile. The written range from 0 to 255 corresponds linearly to the minimum and maximum limits of the device (-100 to 100%). | |
| **Parameters** | |
| id | ITSCAM connection index |
| brilho | 0 to 255 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.34. setaContraste( id : int, contraste : int ) : int

| Description | |
|---|---|
| Assigns the contrast of the photos generated by ITSCAM relative to the active profile. The contrast is equivalent to the digital gain and can range from 0 to 255, which correspond linearly to the minimum and maximum limits of the device (-100 to 100%). The default value is 100 and corresponds to multiplying all pixels in the image by 1, after the shutter and gain have made their automatic adjustments. The value 0 causes the pixels in the captured photo to be multiplied by 0, resulting in a black image. | |
| **Parameters** | |
| id | ITSCAM connection index |
| contraste | 0 to 255 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.35. setaFocoDayNight( id : int, daynight : int ) : int

| Description | |
|---|---|
| Saves the current focus setting to flash memory. | |
| **Parameters** | |
| id | ITSCAM connection index |
| daynight | 50: changes the zoom and focus of the night profile to the same values as the day profile |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.36. setaGanhoFixo( id : int, ganho : int ) : int

| Description | |
|---|---|
| Assigns the fixed gain value for ITSCAM related to the active profile. The written range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ganho | 0 to 72 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.37. setaGanhoMaximo( id : int, ganho : int) : int

| Description | |
|---|---|
| Assigns the maximum gain value for ITSCAM related to the active profile. The written range from 0 to 72 corresponds linearly to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| ganho | 0 to 72 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.38. setaPadrao( id : int) : int

| Description | |
|---|---|
| Restores ITSCAM factory settings: changing image and server values. The network, date, and time settings are not modified. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.39. setaPortaServidor( id : int, porta : int ) : int

| Description | |
|---|---|
| Assigns the port of the server selected in ITSCAM, when FTP or ITSCAMPRO | |
| **Parameters** | |
| id | ITSCAM connection index |
| porta | 1 to 65535 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.40. setaSaturacao( id : int, saturacao : int ) : int

| Description | |
|---|---|
| Assigns the color saturation of the color photos generated by ITSCAM relative to the active profile. The written range from 0 to 255 corresponds linearly to the minimum and maximum limits of the device (-100 to 100%). | |
| **Parameters** | |
| id | ITSCAM connection index |
| saturação | 0 to 255 |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.41. setaShutterFixo( id : int, shutter : int ) : int

| Description | |
|---|---|
| Assigns the fixed shutter value to ITSCAM. The written range from 0 to 2047 is linearly transformed to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| shutter | 1 the limit of the ITSCAM model (see the Integration Manual for limits) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.42. setaShutterMaximo( id : int, shutter : int ) : int

| Description | |
|---|---|
| Assigns the maximum shutter value to ITSCAM. The written range from 0 to 2047 is linearly transformed to the minimum and maximum limits of the sensor. | |
| **Parameters** | |
| id | ITSCAM connection index |
| shutter | 1 the limit of the ITSCAM model (see the Integration Manual for limits) |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.43. setaTipoServidor( id : int, tipo : int ) : int

| Description | |
|---|---|
| Assigns the ITSCAM server type. | |
| **Parameters** | |
| id | ITSCAM connection index |
| tipo | 0: None; 1: FTP; 2: ITSCAMPRO |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Success |

## 7.3.44. leTipoFlash( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM flash settings. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Disabled - Flash is never triggered |
| 2 | Single mode - Flash is instantly activated when an image is requested with the "Photo" command. |
| 3 | Continuous mode - Flash is triggered on every frame captured internally by the device. This mode is recommended only for LED lighting devices. |
| 4 | Single mode with delay - Flash is triggered moments before the shutter exposure, according to the set time (only for the "Photo" command) |
| 5 | Auto - Flash is activated only when the environment is dark, preventing flash from triggering during the day. Generates energy savings for the system |
| 6 | Auto with delay - Uses the same principle as Auto Flash, but uses a delay to optimize lighting at the moment of capture. |

## 7.3.45. leTrigger( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM trigger setting. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Disabled - Photo requests are not made via I/O |
| 2 | Rising edge - Images are sent when there is a rising edge on the I/O. |
| 3 | Descent edge - Images are sent when there is a descent edge on the I/O. |
| 4 | Rising and descent edge - Images are sent when there is a rising or descent edge on the I/O. |
| 8 | Continuous – ITSCAM captures images continuously. If OCR is embedded, only recognized images are transmitted. If there is no embedded OCR, or if it is disabled, all images are transmitted. |
| 10 | High level - Images are sent while the logical I/O level is high |
| 11 | Low level - Images are sent while the logical I/O level is low |
| 12 | Rising and approach edge. In this mode, ITSCAM receives photo requests both physically via I/O, at the rising edge, and via digital image processing. |

## 7.3.46. leTipoSaida( id : int ) : int

| Description | |
|---|---|
| Returns the ITSCAM output setting (pins 3 and 4 of the rear connector). | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 1 | Output set to use flash |
| 2 | Output set for I/O |

## 7.3.47. leValorSaida( id : int ) : int

| Description | |
|---|---|
| Returns output values of ITSCAM. The value for Output 1 returned by this function has no meaning when the output is set to flash. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Output disabled |
| 1 | Output 1 enabled and 2 disabled |
| 2 | Output 1 disabled and 2 enabled |
| 3 | Outputs enabled |

## 7.3.48. leDelay( id : int ) : int

| Description | |
|---|---|
| Returns the flash delay value of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 100 to 25000 | Delay in microseconds between flash activation and ITSCAM shutter exposure |

## 7.3.49. leHdr( id : int ) : int

| Description | | | |
|---|---|---|---|
| Returns the ITSCAM HDR (High Dynamic Range) setting. | | | |
| **Parameters** | | | |
| id | ITSCAM connection index | | |
| **Return** | | | |
| Negative value | Error (see Dynamic Library Error Code Table) | 0 | Disabled |
| | | 1 | Enabled |

## 7.3.50. leModoTeste( id : int ) : int

| Description | | | |
|---|---|---|---|
| Returns the ITSCAM operating mode. | | | |
| **Parameters** | | | |
| id | ITSCAM connection index | | |
| **Return** | | | |
| Negative value | Error (see Dynamic Library Error Code Table) | 2 | Horizontal test standard |
| 0 | Image captured from the sensor | 3 | Diagonal test standard |
| 1 | Vertical test standard | | |

## 7.3.51. leGanhoAlternativo( id : int ) : int

| Description | |
|---|---|
| Returns the status of the alternative/differentiated maximum gain of the ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 | Equal gain in Day and Night mode |
| 1 | Maximum alternative/differentiated gain enabled for Day mode |
| 2 | Alternative/differentiated maximum gain enabled for Night mode |

## 7.3.52. leValorGanhoAlternativo( id : int ) : int

| Description | |
|---|---|
| Returns the value of the alternative/differential gain from ITSCAM. | |
| **Parameters** | |
| id | ITSCAM connection index |
| **Return** | |
| Negative value | Error (see Dynamic Library Error Code Table) |
| 0 to 72 | Value of alternative/differentiated gain |


PUMATRONIX
www.pumatronix.com