

PiSCOPE case study

SAT Lap 2024

[Jean-Marc Teulé](#)

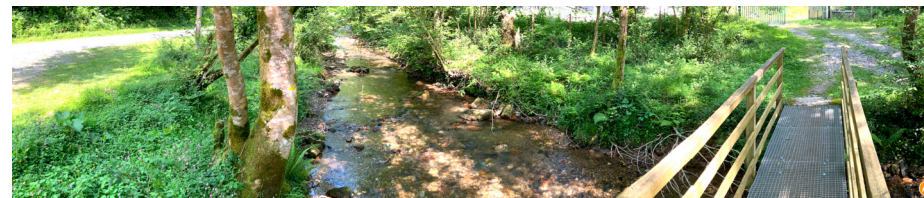
The entire **PiSCOPE** device, and the new cameras waterproofed by us, was deployed at the **Lapitxuri** site as part of the following trial :

SAT Lapitxuri 2024 Refuges Trial ([Eva Haristoy](#), [Cédric Tentelier](#))

- Does a low flow rate reduce the fitness of salmon juveniles in a selective way (on a particular gene or trait)?
 - If so, does the presence of refuges modulate this selection?
- Does a low flow rate lead to phenotypic plasticity?
 - If so, does the presence of refuges modulate this plasticity?
- And finally, does a low flow rate alter the ecological functions of the ecosystem?
- Does the presence of heterogeneous habitats (refuges) modulate this alteration?

This article is a summary of the feedback on the use and adequacy of this device in the context of this experiment.

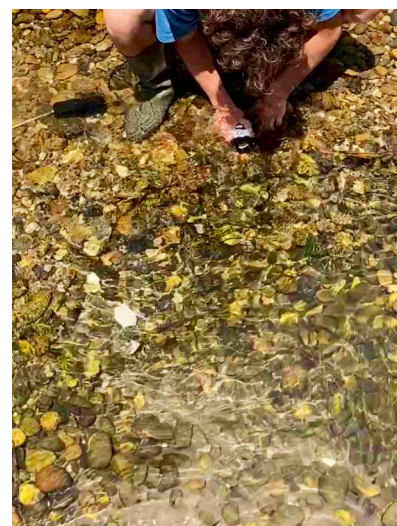
PiSCOPES installed at the Lapitxuri for the experimentation Refuges SAT Lapitxuri 2024



The experimental site of Lapitxuri (LAP) - IE-ECP platform is located after this bridge and borders the little river of the same name that flows along the Franco-Spanish border, in the municipality of [Ainhoa](#).

Our artificial river alongside the [Lapitxuri](#) reproduces the substrate and the course of a river while being divided into 11 biefs, including 6 of them, alternating with the others, will also integrate 3 refuge areas each, these refuges being under video recording via the 18 waterproof cameras to be positioned.

1. WEDNESDAY, JUNE 5: PLACEMENT OF CAMERAS



Refuge areas are holes arranged in the substrate (large gravel) of the bief at the edge of the water current.

Each camera is focused on its own designated refuge area **to record all activity there from dawn to dusk, in natural light..**





The validation of each camera's focus, controlled by Cédric Tentelier himself from our superb on-site video control room!

It was important to place the cameras before the arrival of the salmon juveniles and not to disturb them in their new habitat.

Welcome!



Underwater camera view of the refuge hole

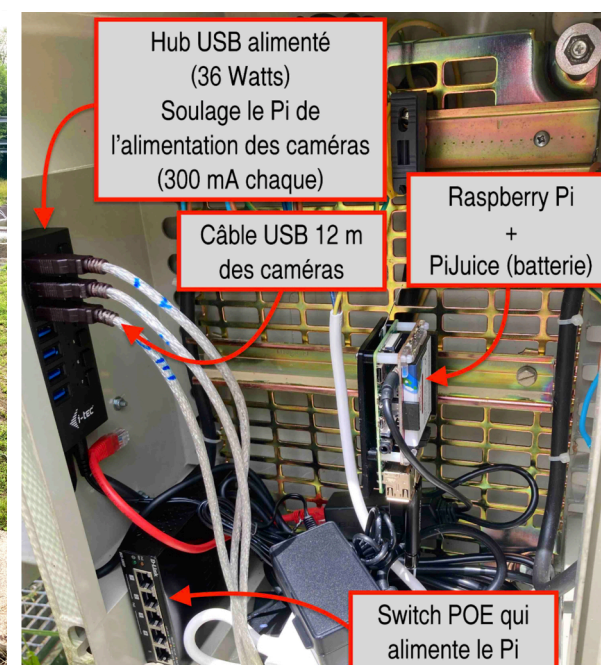
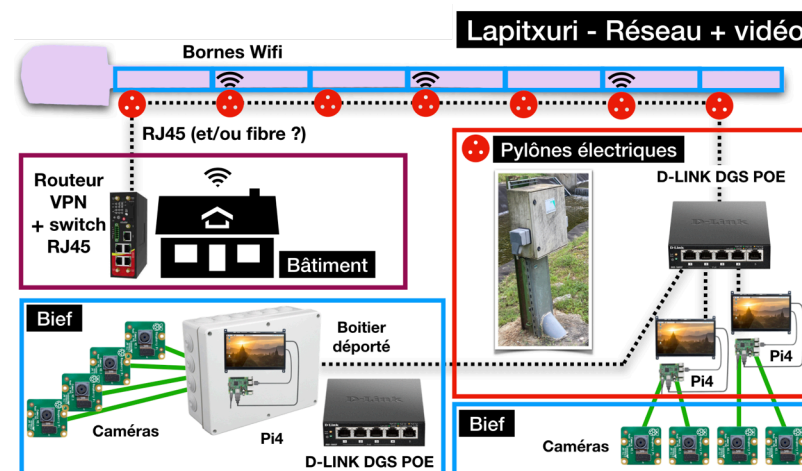
Refuges SAT Lapitxuri 2024 - Eva Haristoy, Cedric Tentelier
Ecobiop / UPPA / ISC-ECP 2024

Jean-Marc Teulé 08/2024

Video: short presentation on the positioning of the cameras to obtain an underwater view of the refuge holes

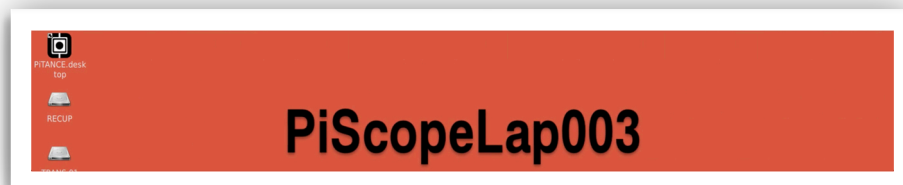
2. FRIDAY, JUNE 7: INSTALLATION OF THE 6 PISCOPES

PiSCOPES devices can fit into the electrical pylons that run along the artificial river or possibly in remote electrical boxes.

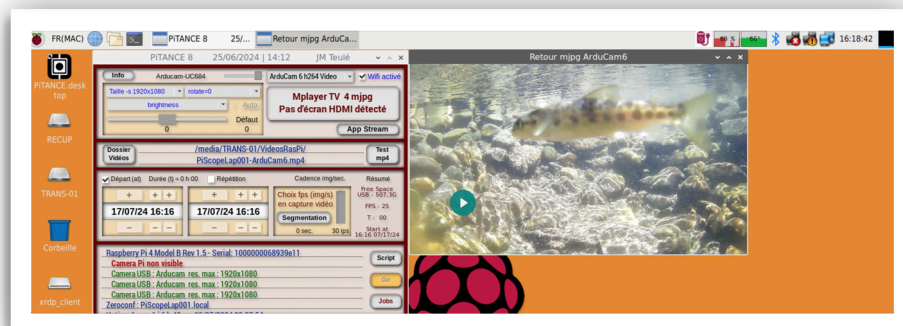


The **PiSCOPES** are connected to each other, from switch to switch, by RJ45 cables circulating in the duct connecting the pylons.

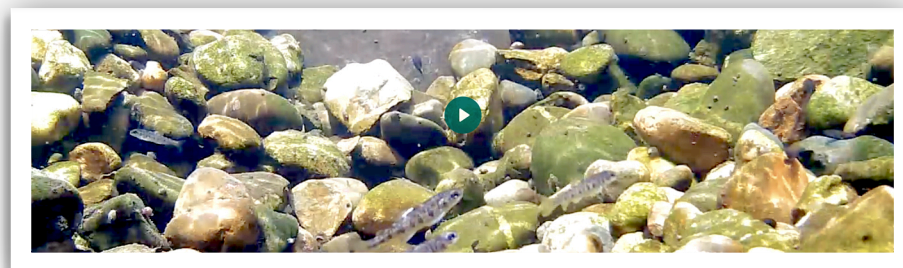
PiSCOPES use xRDP, so they are individually accessible via the RDP protocol with any Windows, Windows App on Mac, or Remmina on Linux.



Video: access to all remote PiSCOPE with a simple trackpad swipe



PiTANCE: near-real-time remote video feedback via RDP



Video: 10-second test recording PiSCOPELap005 Arducam2

Once the **PiSCOPES** have been installed, we can start checking the state of the cameras already in the wide bath of the **LAP**.

Do all the cameras still work when the **PiSCOPES** are started, after these 2 days of immersion? Alas, no...

It was the most stressful period of all, two months of anxiety:

- how many will remain watertight for 2 months?
- how will the PLA of cases behave in a natural environment, under the sun and over time?
- will the USB extension also remain sealed?
- what will happen to the lens fogging with this long immersion?
- how will the **PiSCOPES** components survive in their unventilated pylons in the full summer sun? Fortunately, the channel and pylons are shaded to protect the juvenile salmon from excessive sunlight and predators. Is that enough to protect the **PiSCOPES**?



3. A CATASTROPHE IS COMING.

2 days after the immersion of the cameras, the prototype returns a blurred image. Already dirty? No.

As a prototype, this camera has been through several baths; it is the only one to have had any testing, in fact.

And after 2 days of immersion in the natural world, it was struck by wet cataracts. It wasn't drowned, it still works. But its lens is obscured by condensation.



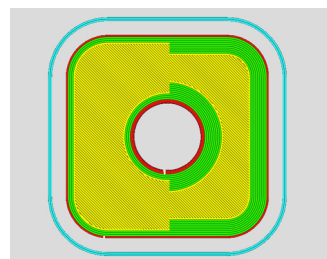
The start of a bad series?



This video camera was the first one assembled, and all the others were designed to the same model, so they are identical, with 2 or 3 differences.

Firstly, this camera fell onto hard ground from desk height, rolling its 13 metre cable, i.e. with some élan, a few days before being plunged into the LAP.

The impacted corner was siliconed for prevention. It was tested in immersion immediately afterwards, but probably not long enough to detect anything.



The second difference lies in the 3D printing of the parts. To print the parts in the series faster, I increased the number of walls (look green alongside), which may have improved the rigidity and watertightness of the PLA in the other cameras.

The third difference is that I was a bit heavier-handed with the silicone when sealing the front and rear covers of the series cameras.

It is impossible to know how this moisture infiltrated and is it really an infiltration? Perhaps the air supposed to be dry was not really dry when sealing the optics. This paradoxically could be good news for others.

4. THE DELIVERANCE

The other 17 cameras in the series did not experience any condensation.

They were removed from the **LAP** between August 1 and 2, 2024.

Either after an uninterrupted immersion of almost 2 months



The recovered electronics smell like old fish, even rinsed!

The **PiSCOPES** worked throughout this period, without any particular alert. They endured the confinement in the Pylons well.

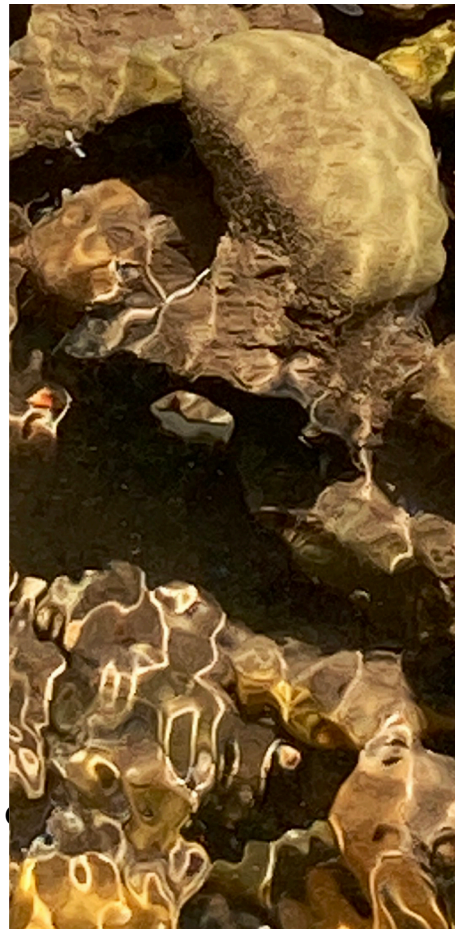
Only intriguing downside: some untriggered video recordings on a camera of a pylon over a short period. The camera was recognised but impossible to record it, no signal.

Powering the USB hub on/off had no effect. Ditto for the Raspberry Pi.

Only physically disconnecting/reconnecting the USB plug of the camera extension cable in the hub allowed the recording signal to be restored. Curiously, this capricious USB connector wore a viscous substance... A false contact from biological origin? Still no answer.

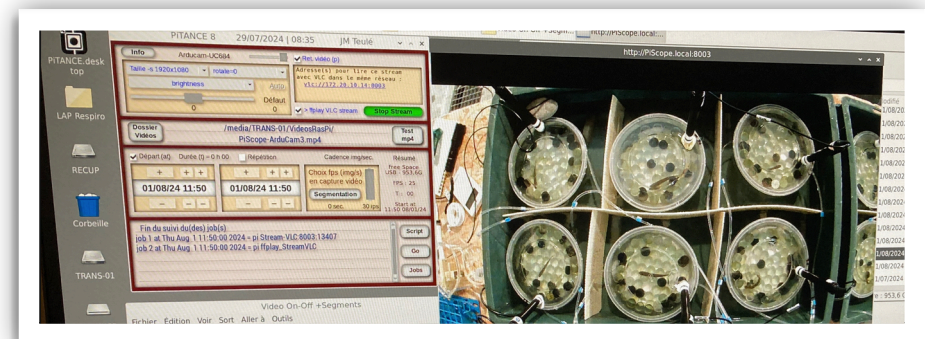
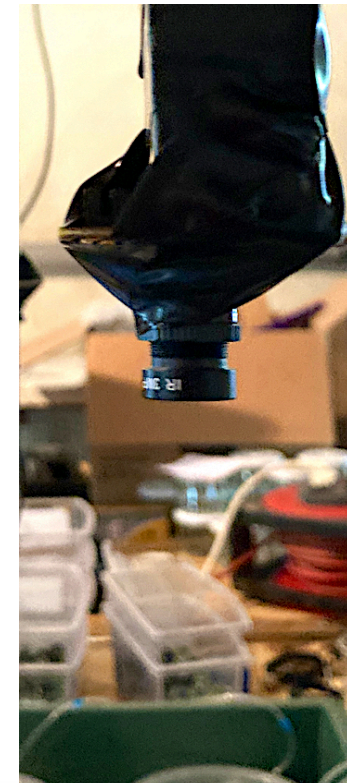
Some cameras suffered more severe silting than others, and logically it was the cameras in the biefs upstream that were affected by this massive silting.

Can you see the camera under the stone?



At the end of their experiment at the **LAP**, Eva and Cédric mobilised 1 **PISCOPE** with 2 naked B0200 cameras and their USB extension to simultaneously record the behaviour of surviving juveniles during 2 days of oximetry manipulation.

The 2 cameras are positioned at the ends of the wood cleats, which are fixed to the beams in the roof.



Video: oximetry - behaviour recording (excerpt)

6. PISCOPE REVIEW OF THE REFUGES LAPITXURI 2024 TRIAL

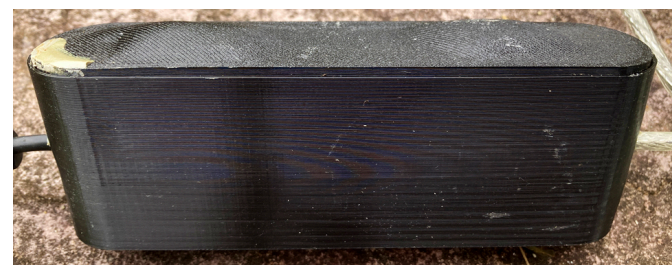
A - Waterproof cameras and their USB extensions

The camera housings held up. The USB extension casings held up to more or less, depending on the type of exposure. A mud and dried seaweed paradise.



The plastic cable ties on the camera housings that held the front and rear covers together were completely useless. They came loose during immersion and were no longer providing any tension after just two days, when the prototype camera failed.

It was tempting to think that they would provide additional protection to prevent the covers from being torn off. The covers did not move from the camera body, and the silicone seal remained intact.

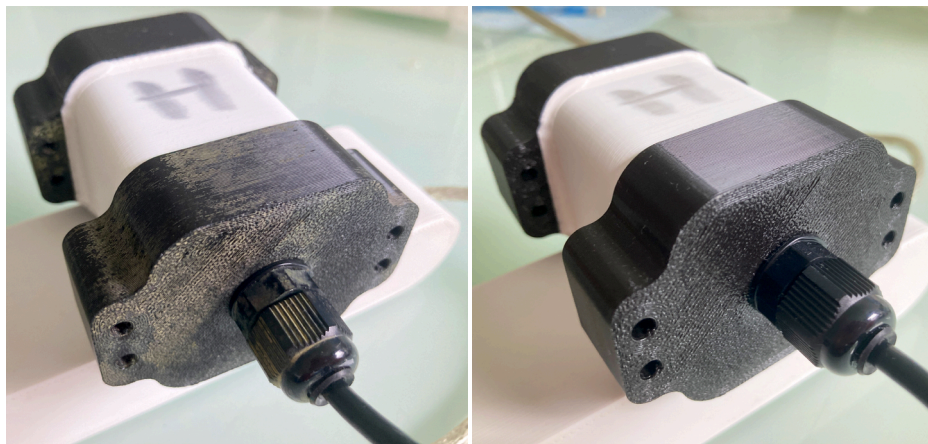


Some of the USB extension boxes suffered a little in the sun, but this had no effect on the cameras operation.

Printing black PLA for USB extensions finally exposed in the sun on the pebbles was not the best idea, but it was the result of not having enough time to change the filament while printing the series... What's more, the connectors were designed to remain underwater.



The 12m USB extension cables that have become inseparable from cameras remain a mess.



Camera before and after cleaning

There were doubts about how PLA would clean up. Algae sticks to black PLA, but white PLA is easier to wipe clean the first time (with a certain pink paste, for example). Mais au final, on arrive à éliminer les algues avec un spray anti-moisissure.

The goal of creating a waterproof, completely sealed, and therefore “maintenance-free” camera—meaning that the lens does not need to be cleaned regularly to prevent fogging—appears to have been achieved, but only under the conditions of this test, after two months of immersion in summer temperatures and low water conditions.

It would be necessary to test in winter to validate the principle.



The electronic board of these cameras has a radiator, so there is a certain heat release in video acquisition.

All this was coated in a dielectric gel. Here too, it was a bet, assuming that continuous immersion in temperate or cool water would be sufficient for heat dissipation. To be seen over time.

This camera casing is not designed for aerial use, let alone in sunlight. These cameras are only dedicated to immersion, in circulating baths.

B - PiTANCE 8 software: record activity in the refuges

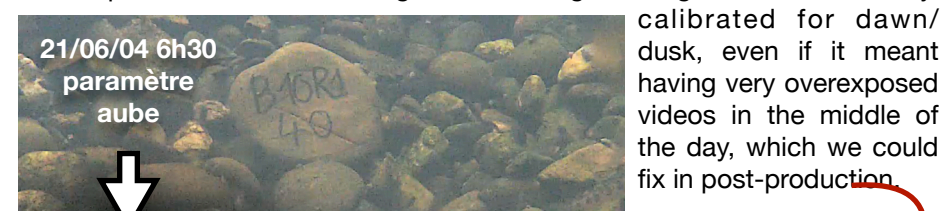
The ability to record multiple videos simultaneously using multiple cameras on a single **PISCOPE**, by using ffmpeg, has completely outdated the **PiTANCE 8** interface.

We needed to program the triggering of all these cameras for identical recording sequences and at the same time, but over heterogeneous durations and times in a day:

5 minutes every 2 hours during the day and 30 minutes at dawn and dusk while knowing that dawn and dusk times vary enough in two months to take them into account almost daily.

The **PiTANCE 8** interface can only schedule each camera one after the other, in terms of duration and image settings, and, if necessary, with regular repetition of this schedule. It thus appears currently poorly designed for this situation.

To compensate, we tried a single set of image settings for the entire day,



calibrated for dawn/dusk, even if it meant having very overexposed videos in the middle of the day, which we could fix in post-production.



Post-production : l'image n'est pas jolie mais l'information contenue dans l'image brûlée est récupérée

But given the anticipated volume of video, this post-production correction was finally no longer feasible.

Indeed, the quantity obtained a posteriori will represent on average an effective recording volumetry per day of:

- 21.75 GB for a **PiSCOPE** including 3 cameras recording in 1920x1080 at 30 frames/second.

So I had to bypass the interface to quickly adapt a semi-manual programme of two batch files grouping together:

- first, all the controls with image settings for the whole day, for each of the 3 cameras of each **PiSCOPE**
- second, all commands with image settings for dawn or dusk, for each of the 3 cameras on each **PiSCOPE**

The Night and Day batch files have been programmed via the AT command

```
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchNuit.sh" | at -M 06:48 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 08:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 10:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 12:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 14:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 16:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 18:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 20:00 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchNuit.sh" | at -M 21:25 07/11/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchNuit.sh" | at -M 06:49 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 08:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 10:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 12:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 14:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 16:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 18:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchJour.sh" | at -M 20:00 07/12/24
echo "/home/pi/...../PiTANCE_Application/DossierRepet/Repet-ArduCamBatchNuit.sh" | at -M 21:24 07/12/24
```

Even in this case, at the end of the day (dawn and dusk), the variation in brightness is quite fast.

On a 5-minute video, fixed settings may become inappropriate. An example of this brightness variation below in accelerated

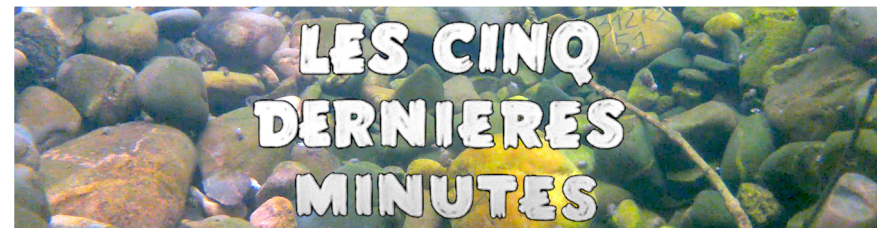


Video: natural light dawn setting from 06/21 from 6:30 am to 6:35 am accelerated x20: 5 min in 7 seconds

It should be possible to program a continuous variation of the image parameters for these transient phases, during video recording.

It's technically doable since the image settings can change manually during recording.

But determining the amplitude of variation desired during a recording is a complex job to interface.

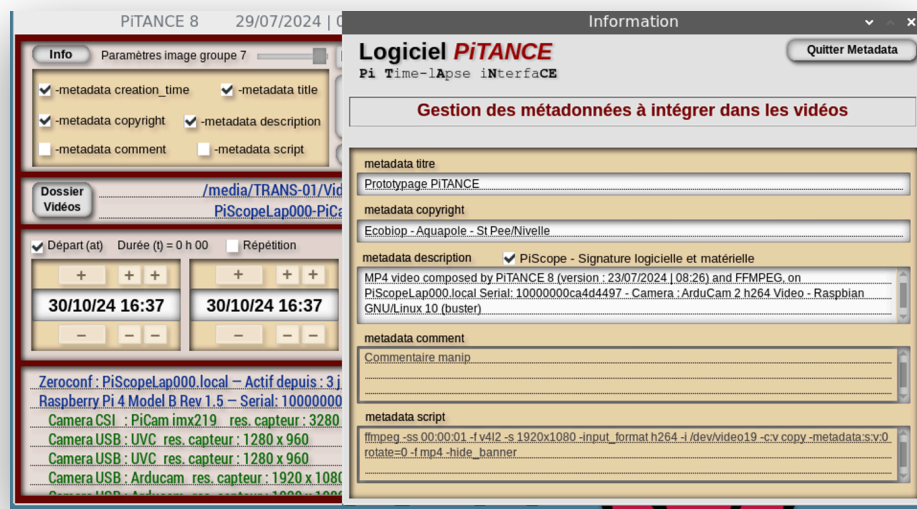


Video: last 5 minutes of the dusk recorded until 9:58 pm on 04/07/24

Natural light - without retouching

C - Test to save personalised metadata in videos

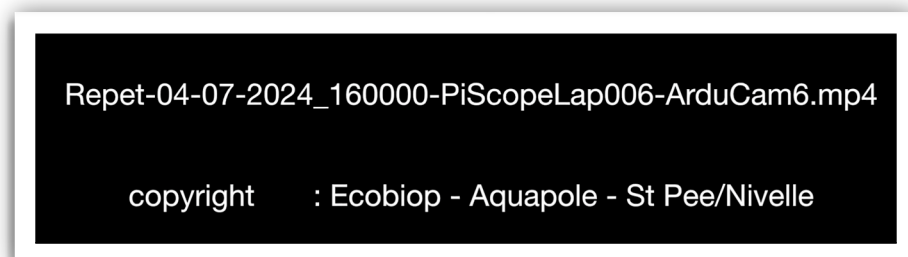
PiTANCE 8 allows you to enter information that ffmpeg can integrate into each video.



You can enter this data to integrate it into the camera programming.

This data is added to the automatic global metadata of ffmpeg that can be retrieved later, from the video, via ffprobe.

In the example of the video below, the added credits were composed only with the metadata contained in the video file:



Video: real-time day parameters + generic displaying part of the

D - Make the Time code and other apparent information in the videos

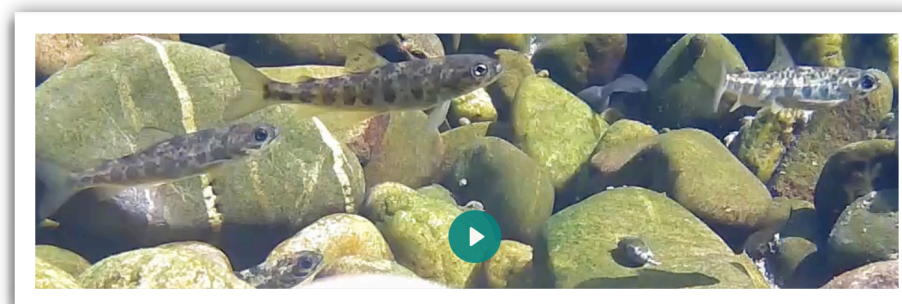
The raspivid command for **Raspberry PI** allows the overprinting of the date and time code of the video while recording it from the PiCAM camera. This overprint has no impact on the processor since it is the encoding of the camera that executes it.



Recording USB cameras does not allow this overlay.

While it is possible in theory to re-encode the camera stream during recording to add time code, processing a single camera would completely saturate the Raspberry Pi's processor. However, the time code of a video can be displayed in overlay and re-encoded afterwards with a video editor.

In the following example, I used DaVinci Resolve to generate an overlay of the file name (date, time, and location of recording) over an unused area of the video, then above that, its source time code, and on top of that, the frame numbering.



Video: Time code of the video and other information added with DaVinci Resolve in post-production

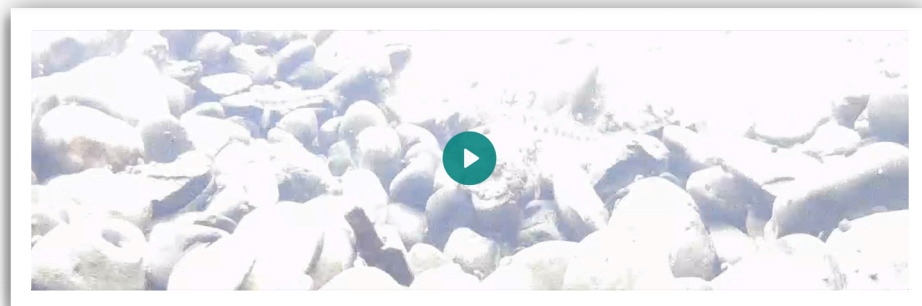
The advantage of post-editing is that you can compose this information and place it in the unused areas of the image. As here, the large pebbles, whose brightness and blur saturate the foreground..

D - ArduCam B0200 camera and automatic brightness adjustment

Sometimes, the automatic brightness adjustment system on **Arducam** cameras seems to hesitate for a while. This is annoying because it is relatively unpredictable, but it seems to be limited to transitional scenes with strong contrasts or scenes that exceed the camera's sensitivity limits.

Important: this is totally independent of the 12-meter active USB extension cable connected between the **Raspberry Pi** and camera.

Warning: video not recommended in case of risk of epilepsy!



Video: 2 examples of the automatic brightness selector problem of the Arducam B0200

E - Unique identifier for USB cameras and ArduCam B0200 in particular

USB device manufacturers can do whatever they want to integrate useful information into the microcode of their devices. This is particularly true for **ArduCam B0200** cameras.

Each has a serial number field but all with an identical reference: UC684 linked to the camera model!

In the case of a pool of identical cameras, as tested at Lapitxuri as part of this experiment, it is annoying not to be able to identify each camera and integrate this identifier into the video metadata.

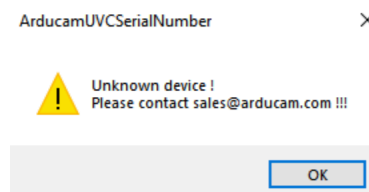
It's a hell of a lack of traceability.

```
Media Driver Info:
Driver name       : uvcvideo
Model            : Arducam USB Camera: Arducam USB
Serial           : UC684
Bus info         : usb-0000:01:00.0-1.2.4
Media version    : 5.10.103
Hardware revision: 0x00000100 (256)
Driver version   : 5.10.103
```

In my experience to date, only Logitech webcams provide a valid serial number. However, neither the brand nor the model name are clearly displayed.

```
Media Driver Info:
Driver name       : uvcvideo
Model            : UVC Camera (046d:0825)
Serial           : 479FE2E0
Bus info         : usb-0000:01:00.0-1.2.2
Media version    : 5.10.103
Hardware revision: 0x00000012 (18)
Driver version   : 5.10.103
```

If you receive the **unknown device** error, please contact [Arducam Support](#) for camera confirmation and technical support.



The **ArduCam** support provides a tool for creating a serial number as you want in the camera's microcode, but alas, it did not work at all as expected for this model.



See article [PiSCOPE Device and Software.pdf](#) for more information.

CONTACT

For any request, explanations, project that can use this device contact me directly : jean-marc.teule@inrae.fr

INTELLECTUAL PROPERTY OF THE CONTENTS:



Attribution (BY)
No Commercial Use (NC)
Sharing on the Same Terms (SA)

Crédit :

- Photos PiSCOPE / PiTANCE / field situations : Jean-Marc Teulé / INRAE
- Videos PiSCOPE / PiTANCE / field situations : Jean-Marc Teulé / INRAE
- Demos, animations, illustrations PiSCOPE / PiTANCE : Jean-Marc Teulé / INRAE

Logiciel PiTANCE : Licence GPL-3.0