

PiSCOPE device & software

[Jean-Marc Teulé](#)

The PiSCOPE device (literally “instrument to see with the Pi”) is an electronic assembly featuring a Raspberry Pi nano-computer (models 3, 4 or 0 depending on the cases, or models tested) and managing one or more cameras. It can be battery-powered, undulated, organized with complementary electronics and storage peripherals, all in an electrical box or any other appropriate container.

The PiSCOPE is powered (to date) by the Raspbian Buster OS, plus the prerequisites for use of the PiTANCE - Pi Time Lapse InterfaCE software, to provide a dedicated, turnkey solution, ready to run via a simple flash of an SD card, with various Raspberry Pi (up to 4).

This article summarizes the continuous evolution of this device in terms of hardware and software.

O rigins of the device: program timelapse under infrared



PiSCOPE began to be established within the Agronomy UMR (Unit of Mixed Research) of Versailles Research Center, my unit at that time, following a request from Antoine Gardarin, on 03/08/2016.



He needed an alternative to hunting cameras for easily programmable night-time images under 940 nanometer infrared illumination, for trials on sentinel prey predation in open fields.

The Raspberry Pi nano-computer and its dedicated Pi-NoIR camera specialized for infrared acquisition, seemed perfect for this work. With the advantage of being cheap and reusable.

It is on this basis that the PiSCOPE prototype was built and then developed to carry out Antoine Gardarin's experiments.

The scientific article documenting this work on sentinel prey was published in the journal **Basic and Applied Ecology Year: 2023**

[Assessing ground predation by invertebrates in crops: Camera observations clarify the relevance of aphid cards and coleopteran sentinel prey.](#)

For more details on the successive realizations of the PiSCOPE device and the use of its PiTANCE software at the time, see (in french only) :

[Dispositif de visualisation au champ PiScope et matériel vidéo](#) - site Agronomie

[Une caméra pour mettre en évidence la prédation des insectes à la surface du sol](#) Le Cahier des Techniques - Inra 2019 (97)

E volution from photo to video

At the time, the PiTANCE application could only program still image recordings (raspistill up to 1 frame / second), and it was not possible to go beyond this (several frames / second) with still images. To get past this stage, it was necessary to use the Raspberry Pi's video capture function (raspidvid) and therefore modify the PiTANCE interface.

It happened when the UMR Agroecology - GestAd Pole of the Dijon Research Center - France, contacted me on 04/03/2019, following Antoine Gardarin's work.

They had started using PiSCOPE but wanted a higher photo capture frequency for their own trials.

Here is an excerpt of their request and feedback:

On Apr. 3, 2019 at 12:38, B..... F..... <b.....f.....@inra.fr> wrote:

Good morning,

I would like to ask if there is a possibility to adjust more than one frame per second with the PiTANCE interface?

Hello, Thank you for the quick answer.

B..... F.....

I would like to test up to 6 frames per second.

B..... F.....

b.....f.....@inra.fr Doctoral student at INRA - UMR 1347 Agroecology - Pôle GestAd17 rue Sully - BP 86510; 21065 Dijon cedex (France)

Following this request, the interface was quickly modified to integrate video programming and all the options for setting up images.

Dijon seems to have appreciated the evolution of the product despite some problems with hard disk formatting, but these were totally independent of the PiSCOPE software:

May 2, 2019 at 4:23 pm, T..... F..... <t.....f.....@inra.fr> wrote:

Hello,

I'm one of the trainees working on the PiScope field cameras in Dijon, at the UMR Agroecology, and I'm having a problem storing the images taken by these cameras. To give you a bit of background, I want to identify predators and quantify weed seed predation, according to cropping systems. I had therefore planned to make films every two days for a week (to have a unity of time and season of fauna development) with 12 cameras, i.e. 36 films in one week. This means I have to dump all 12 cameras in a relatively short space of time. To do this, we provided 2 external hard drives (Seagate Backup Plus Hub Desktop 6Tb). Everything worked very well in the preliminary tests, with large film/photo sessions lasting 24 hours at 1 frame/s, thanks to a write speed on the disks of 30MB/s, i.e. around 1GB/min. But now that I've started my first real film session, the write rate on the hard disks has dropped sharply and is now down to 1.5MB/s. So I was wondering if you've had a similar problem as well as how you dump your memory cards and store your data?

On May 3, 2019 at 5:37 pm, T..... F..... <t.....f.....@inra.fr> wrote:

Hello,

Thank you very much for the reply.

And yes, Dijon is the new Hollywood of agriculture!

Yes that's indeed how we do it. Following your advice we're going to invest in usb sticks that we'll dump into the hard drives. However, my problem seems to be with the hard disks. Transferring the cards to another hard disk works fine, but when I tested the hard disks they seemed very slow. I've tried cleaning them up, analysing if there are any corrupted directories and there doesn't seem to be anything so I've put one in to be formatted, we'll see on Monday. If you have any other ideas, don't be afraid to suggest them.

May 10, 2019 at 10:30 am, T..... F..... <t.....f.....@inra.fr> wrote:

Hello,

Thank you very much for all this info which seems to have solved my problems. Formatting in exFAT seems to have solved the problem of writing to the hard disk (and just for info, exfat-fuse and exfat-utils are installed by default under ubuntu 18.04, which I use on my computer and on the one with which I read the Piscope cards). To avoid having to read the cards, I'm going to put usb sticks in the bugscope and write the photos to these sticks (one advantage is that I won't have to go through linux to read them, and I won't use the cards for nothing).

Now it's ready for the Cannes Film Festival.

Have a nice day!

Sincerely T.....

Evolution from aerial to aquatic

Since 2020, I have integrated the Ecobiop UMR which, until then, used a fairly complete analog video acquisition system:

1. **aerial IP cameras** with a signal transmitted via optical fiber to a powerful control system, controlled by Linux software developed by a third-party company.
2. **analog aquatic cameras** system (shown on right), used mainly in natural environments such as rivers.



I proposed to Ecobiop to see if my PiSCOPE could meet the challenge of doing as well or even better under the same conditions.

Aerial camera rather than having a monolithic, centralized control system, use Raspberry Pis as many localized video acquisition systems, controllable remotely (Wifi, RJ45...).

Aquatic camera the analog system has serious advantages:

1. Analog video has the advantage of allowing the video signal to be transported over a greater length, with a thinner cable between the camera and the system that digitizes the signal. This allows you to restrict the submersible part to the bare minimum, i.e. the camera.
2. Digital signals cannot be transported over long distances. So we need to find a way of extending the distance between the camera and the Raspberry Pi, otherwise we'd have to make the whole Raspberry Pi + camera + battery + data storage submersible.

To replace this analog set with the 100% digital PiSCOPE system, there are three topics to deal with:

Topic 1 is it possible to extend the distance of the CSI camera without any loss of quality?

- if the CSI camera can't be used, can we use a USB camera and extend its distance as well?
- if each type of camera (CSI and USB) can be extended, can both be used at the same time?

Topic 2 which camera models to choose from the existing variety, and how to make at least one of them waterproof, while avoiding the formation of internal condensation on the camera's optics, which remains a recurring problem during several days' immersion in a river?

Topic 3 complete the PiSCOPE and its PITANCE software, for optimum control of video recordings under these new conditions, and to adapt the interface to the needs of users for ease of use in the field.

Topic 1 Commercial proposals for extensions between CSI and USB cameras and the Raspberry Pi

The current length of the CSI ribbon that connects the Pi camera to the Raspberry Pi is 15 centimeters, most often.

Different lengths of CSI ribbons
© photo iuniker

The largest length available on the market is 2 meters.

Any extension of this ribbon beyond 15 cm could expose the signal to electro-magnetic noise and signal loss, thus affecting the quality of the videos recorded.

It also remains insufficient for our extension needs (beyond 10 meters if possible).

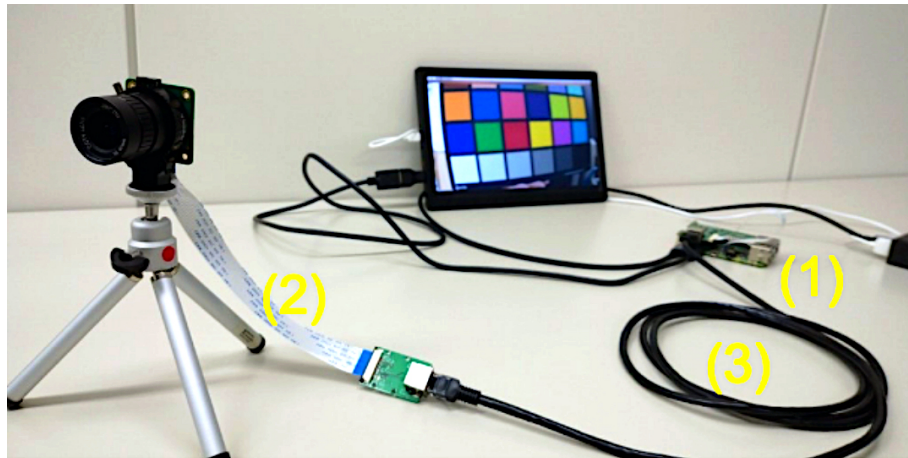


Solutions longer than 2 meters are available, with an RJ45 or HDMI cable.

HDMI-CSI extension
© photo Arducam



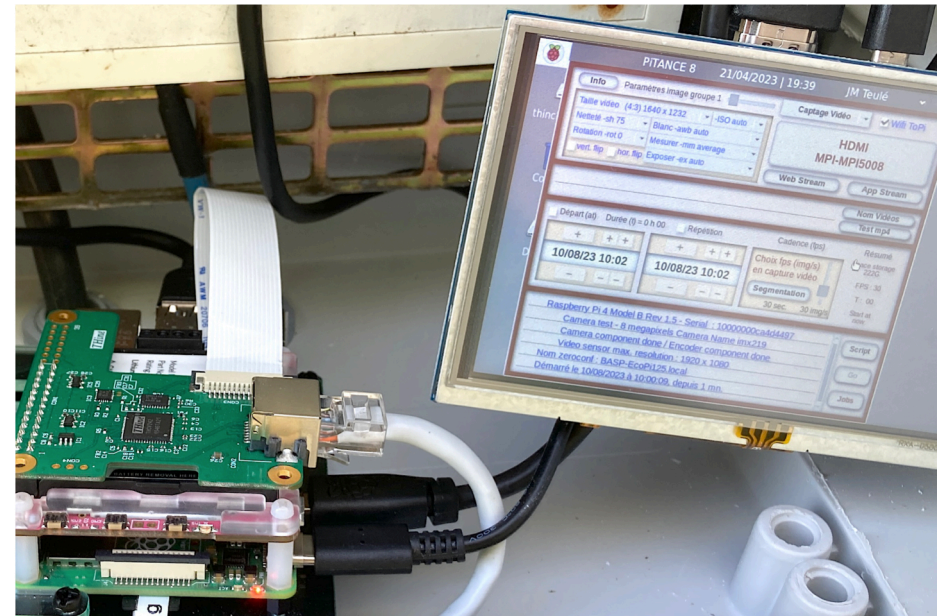
- **The HDMI-CSI solution** The HDMI-CSI solution seems passive (no additional power consumption or heating), but use is limited to 10 meters, and longer HDMI cables incorporating fiber optics are very expensive. This option is not currently validated.
- **The RJ45 solution from THine** “serializes” the camera signal, so it's not passive (power consumption, slight heating of the camera converter). However, results are very good up to 15 or 20 meters, depending on the quality of the RJ45 Cat6 cable used. RJ45 connectors are easy to crimp. This is important to facilitate the passage of bare cable in a waterproof box before crimping these connectors.



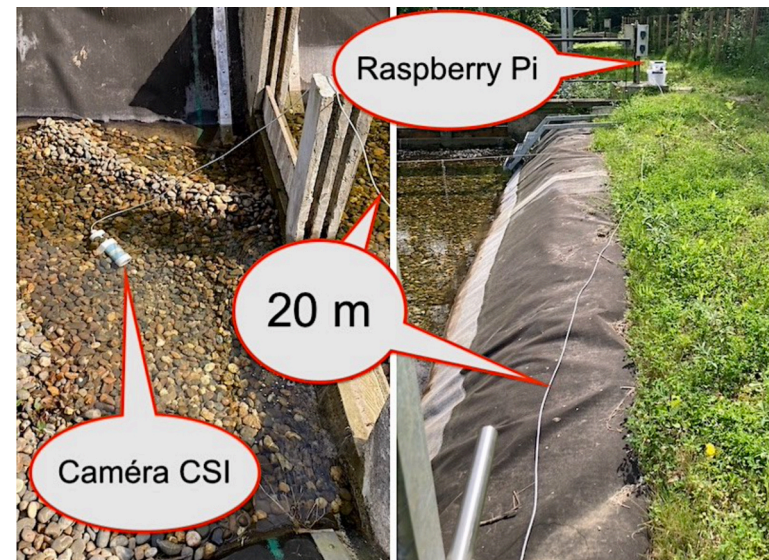
1. Plug Rx Board to Raspberry Pi Computer
2. Plug Tx Board to Raspberry Pi Camera
3. Connect Rx and Tx Boards with LAN Cable

© photo THine

This THine solution has finally been chosen for the signal extension between the CSI cameras and the Raspberry Pi: [validation test in a reach of the Lapitxuri artificial river from 08/10/2023](#) with an nearly 20 meter RJ45 cable between the CSI camera and the Raspberry Pi.



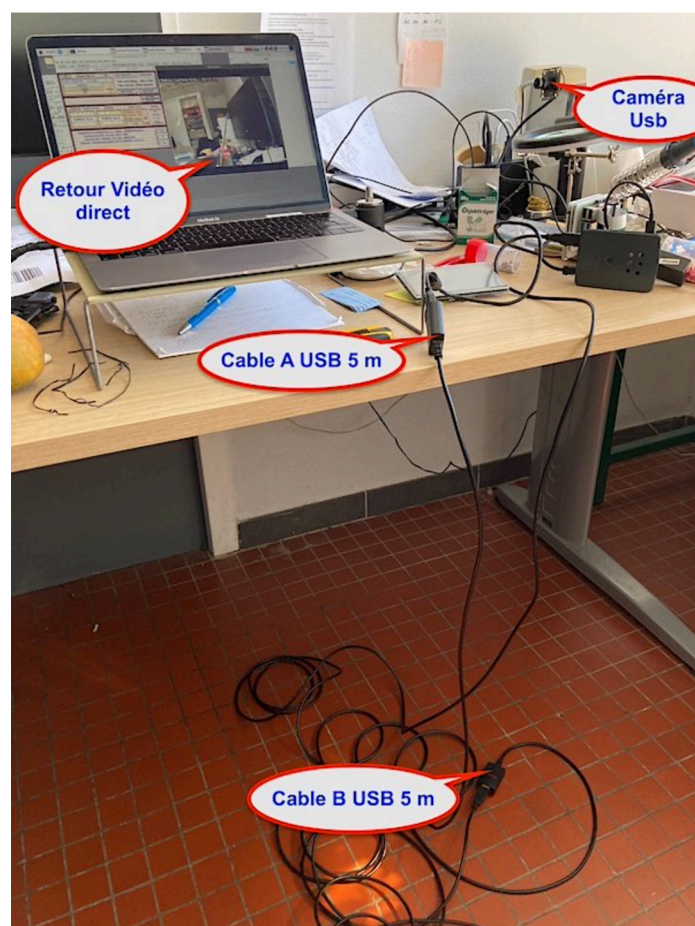
PiSCOPE and its PiTANCE 8 software (5" screen) just installed in a Lapitxuri electricity pylon with the 20 m THine extension (RJ45 cable) to the CSI camera in a Lapitxuri reach.



Concerning the extension of USB cameras, I tested several lengths of active USB extension cables (regenerating the signal) from different brands, considering that USB cameras already come with a fixed 1 m cable:

- Length 05 m : **OK** - Length 05 m x 2 (10 m) : **OK**
- Length 10 m : **OK** - Length 10 m x 2 (20 m) : **KO**
- Length 12 m : **OK** - Length 12 m x 2 (24 m) : **KO**
- Length 20 m : **OK but has a non-waterproof connection in the middle of the cable**

Test: 2 x 5 m (10 m) USB cables between CSI camera and Pi



We opted for the 12 meter extension, with no intermediate connection to seal.

Considered sufficiently long for current needs.

Inexpensive (length/price ratio).

It is rated for a maximum range of 36 m (3 x 12 m), but obviously not for our video transmission tests.



[USB Repeater Cable 2.0 12 m](#)

Topic 2 Construction of a waterproof camera and if possible, capable of avoiding internal condensation in prolonged immersion

BIBLIO AND INFORMATION

A very short but interesting preliminary documentary research...

1. [DEEPi: A miniaturized, robust, and economical camera and computer system for deep-sea exploration](#)

The DEEPi camera is an extremely compact remote head unit (~16 ml volume), can operate to depths of at least 5500 m, and uses a photopolymer 3D-printed shell partially filled with epoxy as a pressure housing

2. [Waterproofing your Electronics Project](#)

This post will focus on options available with a modest budget and also include a few interesting methods we haven't tried yet for reference. To put all this in context; we deploy our DIY loggers to typical sport diving depths and usually get solid multi-year operation from our underwater units.

PROTOTYPE FOR OUR FIRST WATERPROOF CAMERA

We chose the [Arducam B0200 USB camera](#) for its low-light sensitivity, optical barrel large enough to build a waterproof cell around it, 100° field of view and good value for money.



ArduCam

USB Low Light Wide Angle Camera Module

2MP Sony IMX291
(SKU: B0200)

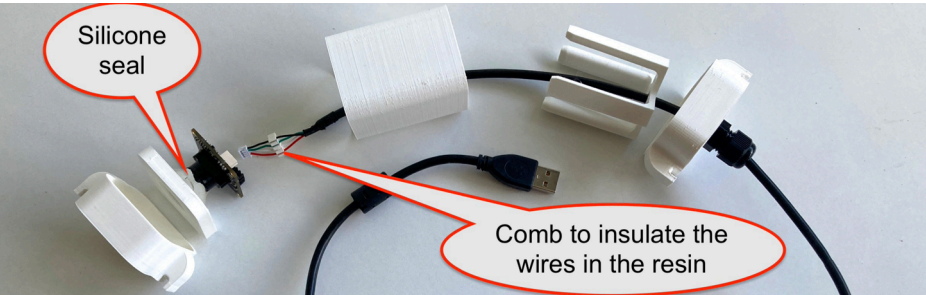
Camera	
Sensor	1/2.8" IMX291
Resolution	2MP 1945H x 1109V
Data Format	MJPEG/YUY2/H.264
Frame Rate	H.264 30fps@1920 x 1080; MJPEG 30fps@1920 x 1080; YUY2 30fps@640 x 360.
Dynamic Range	80DB
Minimum Illumination	0.001Lux
Lens	
Field of View (FOV)	D= 100°
Lens Mount	M12
Focusing Range	3.3ft (1M) to infinity
IR Sensitivity	Integral IR filter, visible light only
Functionality and Compatibility	
Auto Controlled Features	Saturation, Contrast, Acutance, White balance, Exposure.
Audio	Single microphone (optional dual channel)
System Compatibility	Win7/8/10, Linux and Mac with UVC
Power	
Power Supply	DC5V
Working Current	MAX 300mA
Physical	
Operating Temp.	-4°F~158°F (-20°C~+70°C)
Board Size	38x38mm (Hole pitch 28x28mm, 34x34mm)
Cable Length	3.3ft (1M)

After several iterations of case modeling to find a simple yet effective fit, especially around the optics, all the parts were 3D printed locally using PLA filament deposition.

However, FDM printing and PLA material are not particularly well-suited to the construction of objects for use in the wilderness, and for being weathered or, worse still, immersed in water.

This probably won't be a problem, since the housing is a simple shell, in contact only with the outer perimeter of the optics and the 4 legs of the

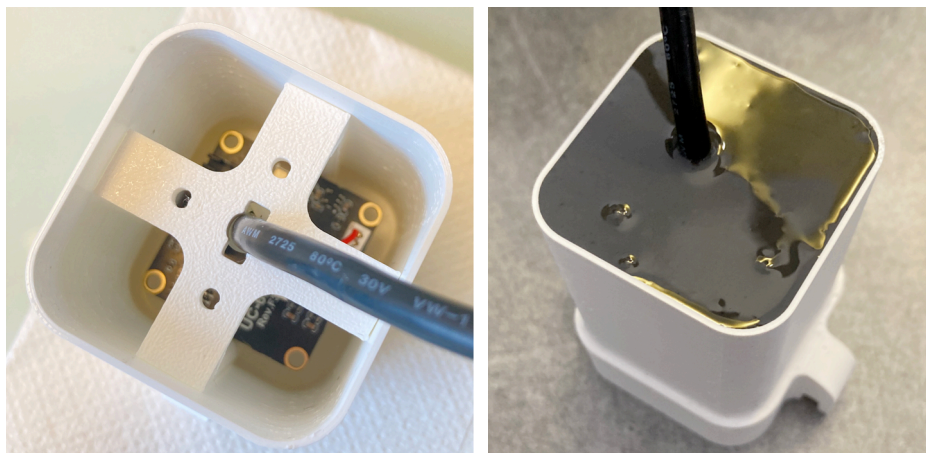
spacer that secures the empty space by centering the camera in it. This void will be filled with a dielectric resin.



Principle: after several hours (overnight) in an oven at 50°C, the optics and optical cradle are sealed with silicone sealant, leaving an air gap $\leq 1\text{mm}$ between the optic and the front glass.

Purpose: to leave as little air (assumed to be dry) as possible between the lens and the front glass in an attempt to avoid internal condensation on the lens during prolonged immersion.





The rest of the camera is completely flooded with a flexible gel for electrical boxes (right image above) in the hope that its features are true:

Non-toxic and safe | Non-spillable | Flame-retardant and non-propagating | Low emission of toxic and corrosive fumes and gases | Flexible and elastic over time | Remains accessible | Rapid cross-linking at room temperature <10 min | High dielectric properties | High mechanical, elastic and thermal properties | IP68 protection degree | Highly conformable and enveloping | For installation up to 1kV | Density: approx. 1.22 | Dielectric strength: >20 kV/mm | Volume resistivity: > 2 . 10¹⁵ Ω cm.



A first bath for 8 hours for the new Ecobiop B0200-001 camera

MAY 3, 2024: REAL-LIFE TEST IN A REACH OF THE LAPITXURI RIVER

The aim of the trial was to test the prototype camera in low-light conditions and validate its waterproofing.

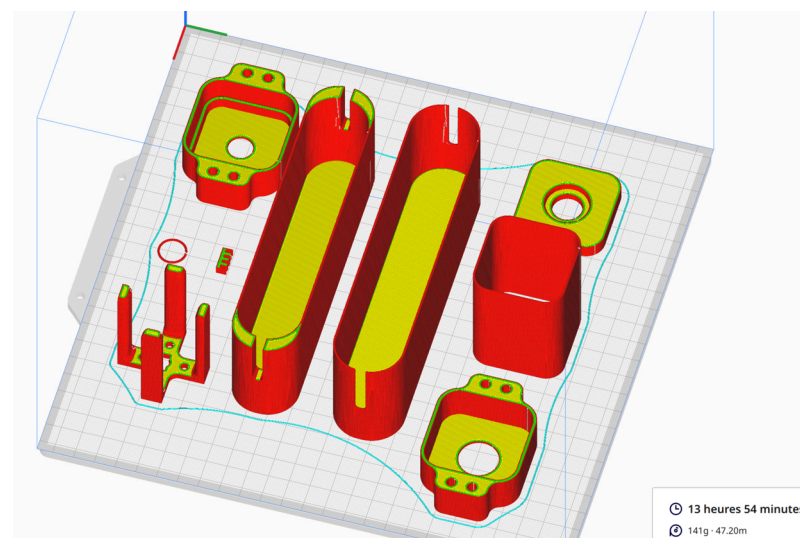
Visit : [Ecobiop B0200-001 twilight bath at Lapitxuri](#) (slideshow in french)

After this test, the prototype was validated with its 12 meter USB extension cable for the immediate construction of a small series of 18 cameras, the number required for a trial about to start less than a month later at [Lapitxuri](#) and entitled: Use of refuges and adaptation of salmon juveniles to low water levels (Refuges SAT Lapitxuri 2024 - Eva Haristoy, Cedric Tentelier).

CONSTRUCTION OF THE SMALL SERIES

Small batch production was hampered by a very tight schedule, but we managed to assemble 17 cameras out of the 18 planned. We had to innovate a little for good reproducibility, and also buy a 2nd 3D printer and use a 3rd (personal) printer to finish on time.

View of printed parts for one camera

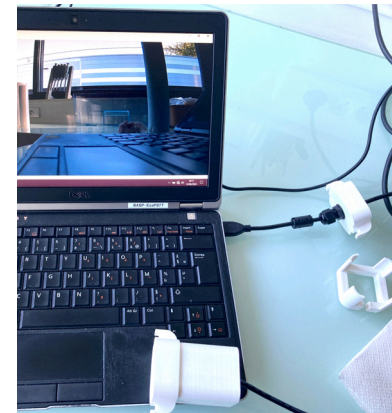


1. Creation of an applicator for the gluing of the external glass



2. Sealing the camera in its cradle - the trickiest part!

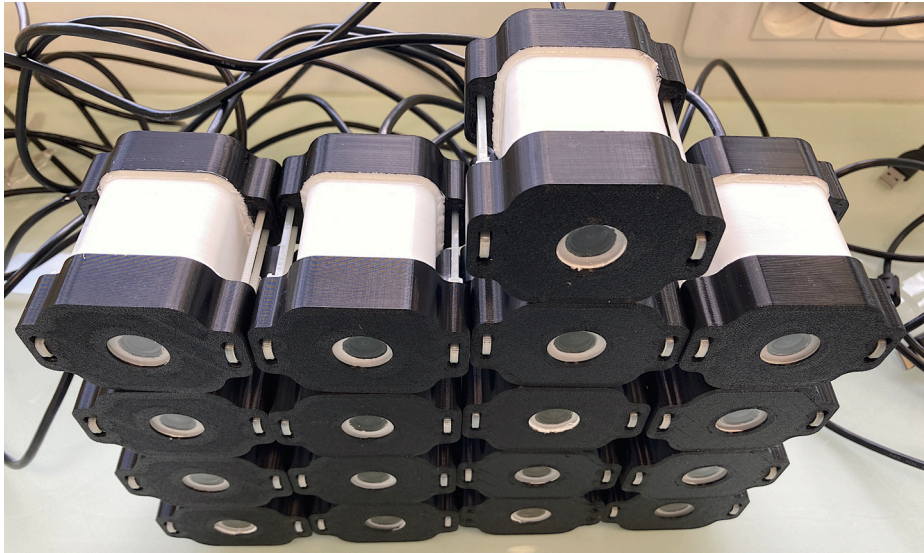
The optic are sealed against the 0.5 mm washer and the front glass with silicone, after these parts have spent a night in an oven at 50°C.



Goal: avoid optical condensation when immersed in water, via a minimum amount of sealed dry air between lens and glass.

We validate the camera aiming (video test) before filling the casing with dielectric gel and sealing the rear cover with silicone.





June 3, 2024: 17 out of 18 cameras are built. Prototype will make the 18th, time is running out!

3. Sealing connections: USB camera / 12-metre extension cable



© photo MCAD

We must add the 12 meter USB and amplified extension cable to the wire of each camera and make this pack waterproof by placing it in the resin-coated oval cases.



4 juin 2024 : les raccords USB deviennent étanches

Unfortunately, it's not the same resin as the one used for the camera (temporary out of stock). This resin hardens completely as it dries, so the seal is definitive. The brown adhesive tape is there just long enough to seal the cable passages, until they dry.

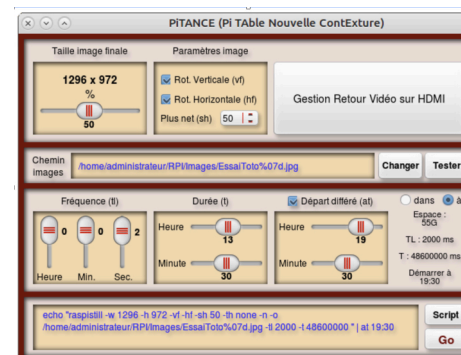
The volume of the printed oval cases was estimated so that one mixing bag of this resin fills 2 of these cases.

Topic 3 Evolution of PiTANCE software and PiSCOPE device to enable multi-camera management

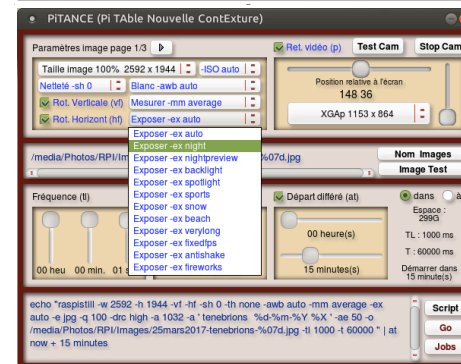
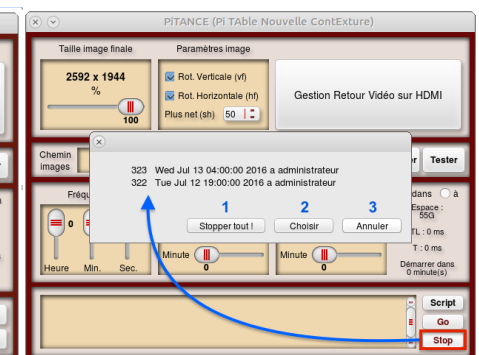
The PiSCOPE system is powered by the PiTANCE video recording programming software under GPL V3 license.

Evolutions since the first versions, the global form remains unchanged. Until version 8, PiTANCE's interface is in French only. Version 9 is under construction.

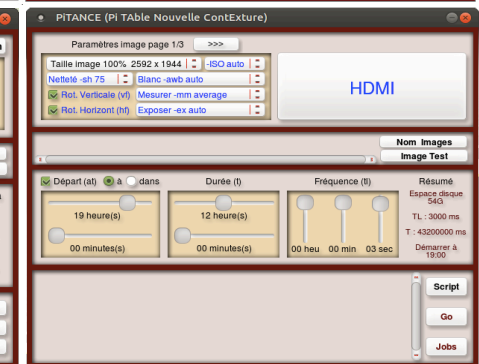
PiTANCE 1 Timelapse to the simplest



PiTANCE 2 Elimination of jobs

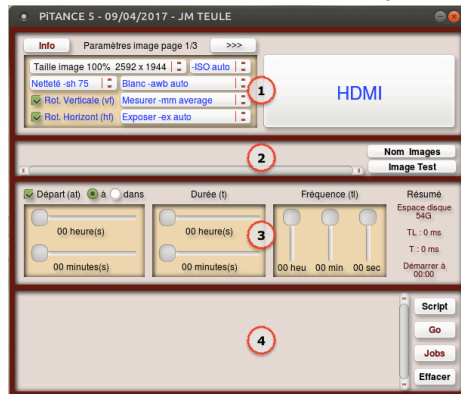


PiTANCE 3 Adding image settings

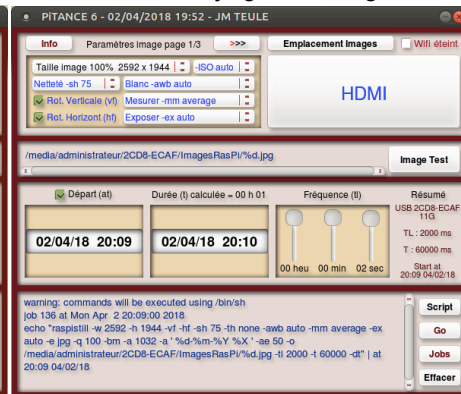


PiTANCE 4 Modifications interface

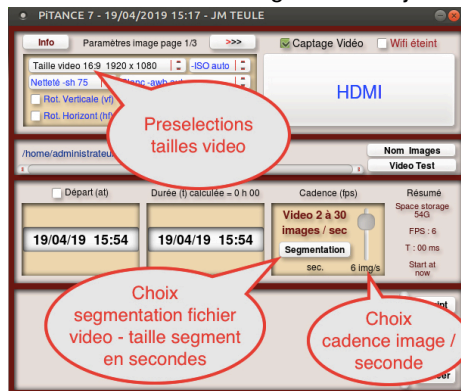
PiTANCE 5 Wifi chip and display



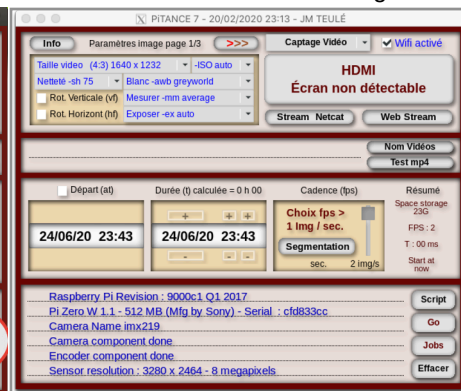
PiTANCE 6 Modifying time management



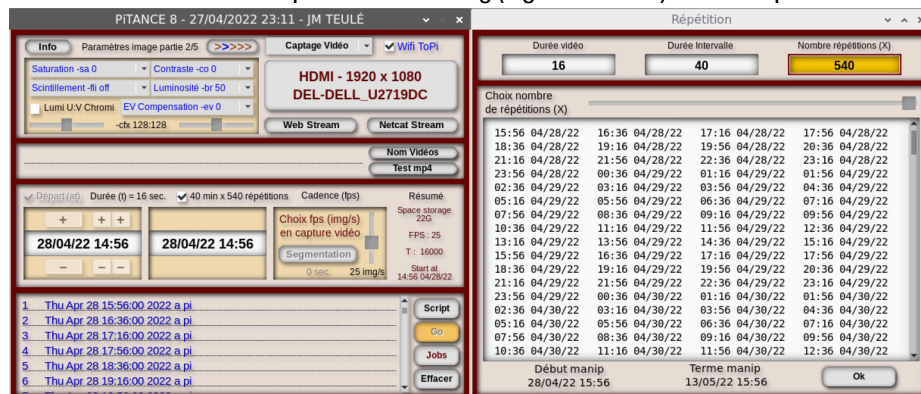
PiTANCE 7 Video management for Dijon



PiTANCE 7 MJPG and VLC streaming control

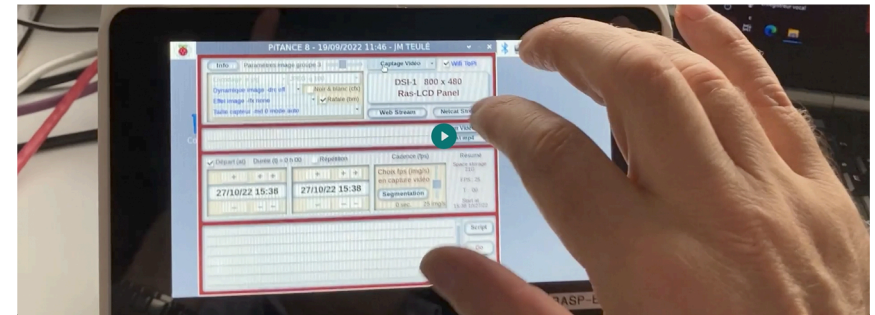


PiTANCE 8 Repetitions scheduling (regular intervals) for Ecobiop



A few videos of PiTANCE 8 in action

PiTANCE is a constantly evolving interface, and differences are inevitable between test and demo videos from different periods.



Video: PiTANCE 8 direct video feedback on 7" touch screen

PiTANCE Pi4 + Stream VLC
et **QRcode** pour **VLC** distant

Video: VLC stream playback from iPhone via QRcode

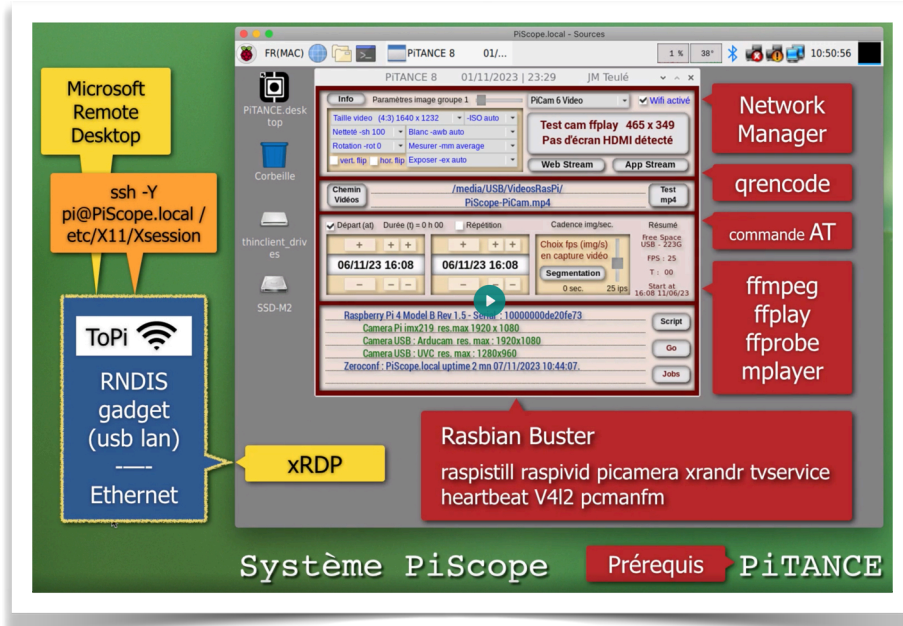
Stream MJPEG caméra **CSI Pi4**
Web Service Python + Picamera

Video: playback stream MJPEG Picamera

PREREQUISITES FOR PiTANCE 8 TO RUN WELL

Warning :

- PiTANCE 8 still works with old libraries **raspistill** et **raspivid**.
- The PiSCOPE system still works with the OS **Raspbian BUSTER** for Raspberry Pi up to 4.



PiSCOPE system = set of additions to Raspbian BUSTER and prerequisites for PiTANCE

MULTI-CAMERA MANAGEMENT

The video encoding of the CSI cameras dedicated to the Raspberry Pi is managed by its GPU, resulting in zero impact on the system.

To use the other cameras, I experimented with recording the signal from a USB camera via ffmpeg **without encoding it** on the Raspberry Pi 4 (retrieving the h.264 or MJPEG stream from the camera and encapsulating it directly in an MPEG4 container, i.e. an .mp4 file).

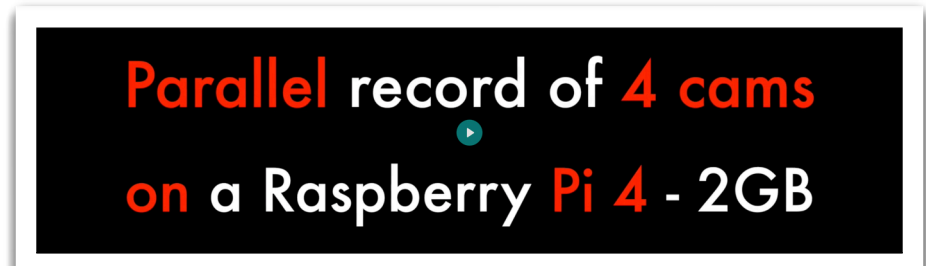
I noticed that the rate of CPU occupation remained very low.

Immediately the question arose: can a second USB camera be recorded simultaneously, and if so, how much more?

I couldn't find any similar examples on the Internet.

My knowledge of the arcana of ffmpeg remains very partial, such is its obscure clarity. But after a manual proof-of-concept, I've integrated these possibilities into PiTANCE 8.

Here's a video, a bit long, which validates these steps, including the simultaneous recording of 4 videos at 1920x1080 (the Pi's CSI camera + 3 Arducam USB cameras) on a Raspberry Pi 4 - 2 GB (RAM) in repeat recording mode, and then, a summary of an earlier test with 5 synchronous cameras, but not all at full resolution, the maximum memory of the Raspberry Pi allocated to the USB bus having been reached.



Video: Synchronous recording of 4 cameras 1920x1080 then 5 cameras (one at 1920x1080 + four at 1280x720)

We can also see the limits of PiTANCE 8's interface, which can now manage several USB cameras, but is still based on a camera-by-camera approach to setting their parameters.

Its structure has become unsuitable for this evolution. This calls for a profound redesign of the software interface (PiTANCE 9 project).

This multiplication of cameras has also triggered a reorganization of the equipment featured on PiSCOPE.

PISCOPE COMPONENTS

1. USB CAMERA POWER SUPPLY

Using several USB cameras for simultaneous recording requires powering them outside the Raspberry Pi, which will otherwise be unable to sustain indefinitely the multiplication of power drain on its USB bus. The maximum power requirement given for the Arducam B0200 is 300 mA max on 5 Volts.

That's why you have to use [a so-called active USB hub](#), i.e. self-powered and not a passive USB hub.



HUB 7 port USB 3.0 Charging + Power Adapter 36 W - © photo I-Tec

This model features a switch on each port with its own activity LED, making it easy to switch on/off or reset each USB device without disturbing the others or manipulating the plugs.

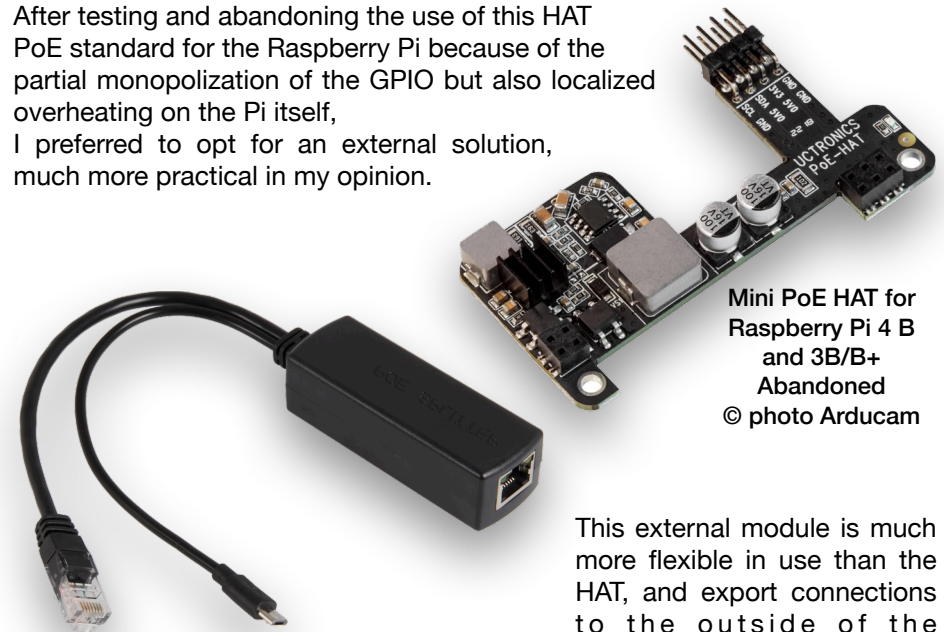
The status of the switch remains permanent even after a power cut!

Caution: many USB hubs are only active on 1 or 2 USB sockets dedicated to charging smartphones, for example.

Important: you'll need a backup power supply for the USB hub, otherwise the cameras will be out of action if the main power fails.

2. POE POWER SUPPLY AND ELECTRICAL RESILIENCE OF THE RASPBERRY PI

After testing and abandoning the use of this HAT PoE standard for the Raspberry Pi because of the partial monopolization of the GPIO but also localized overheating on the Pi itself, I preferred to opt for an external solution, much more practical in my opinion.



Mini PoE HAT for Raspberry Pi 4 B and 3B/B+ Abandoned
© photo Arducam

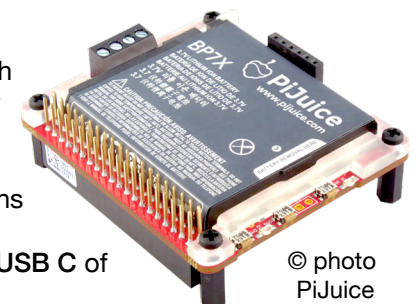
PoE separator module chosen for powering a Raspberry Pi © photo Adafruit

This external module is much more flexible in use than the HAT, and export connections to the outside of the Raspberry Pi. As this module is inserted between the PoE network cable and the Raspberry Pi, you can decide

whether or not to use it, without having to tear everything down.

What's more, it's ideal for use with the [PiJuice, the UPS uninterruptible power supply dedicated to the Raspberry Pi](#).

1. Totally uninterrupted power supply
2. Integrated smart on/off switch
3. Low-power deep sleep state with wake-up during a calendar interruption/event
4. Built-in RTC real-time clock
5. Buttons (x3) with super simple options that can be modified by the user
6. Charging via built-in micro USB or via USB C of Raspberry Pi



© photo PiJuice

7. Complete Energy Management API
8. Improved graphical user interface (GUI) available for easy installation (via APT)

Each Raspberry Pi 4 + PiJuice assembly is positioned on a [3D-printed plate to fit onto a DIN rail](#). It's an economical mounting system that fits perfectly into electrical boxes.



The USB key is a 1 TB SSD for storing videos.

Raspberry Pi + PiJuice are powered by the network distributed by POE switches.

The PiJuice authorizes the direct power supply on the USB C of the Raspberry Pi and recharges its battery via the GPIO.

But in this case, the current received is not undulated by the PiJUICE! And that was a problem for me at the Lapitxuri.



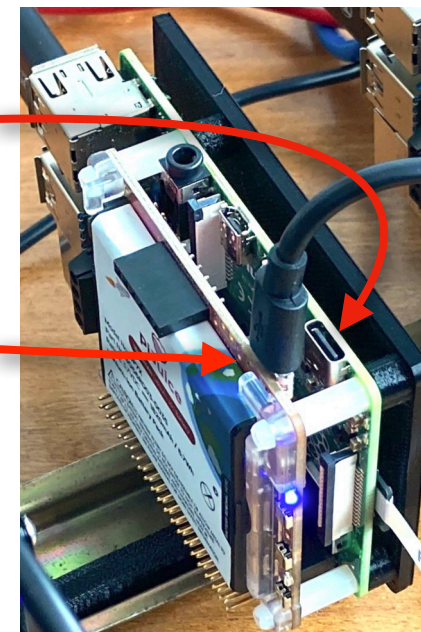
Video: POE assembly powering Raspberry Pi / PiJuice

As experienced at the [Lapitxuri](#) :

- The Raspberry Pi, connected directly to the AC main with original USB C power supply, freezes or crashes repeatedly. In short, an unusable Raspberry Pi.
- The same, once connected to the same current through the PiJuice's micro USB, no malfunctions. A very stable Raspberry Pi.

A current voltage that's not very clean or fluctuating is not always easy to spot.

Then maybe it's better to power the Raspberry Pi via the PiJuice's micro USB input to obtain a undulated current.. Even if the current is already filtered by the PoE switch, I opted for the micro USB version of the POE separator module. But it might be interesting to have both versions.



An important point - does not concern the Pi Zero:

The Raspberry Pi uses an SD card for its operating system. This card must be fragile since the forums are full of bitter comments on the subject. However, since 2016, I've only burned out 1 SD card, but no one more.

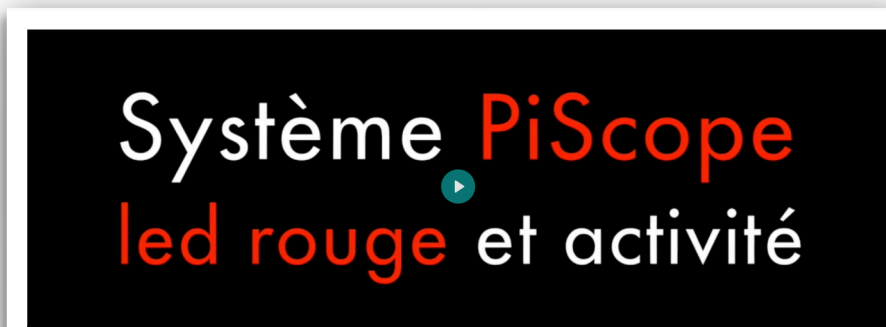


Why do I do this? Probably because I make sure that the Raspberry Pi's shutdown sequence is completed before powering it off, to avoid bricking the system.

But it's almost impossible without keeping your eyes on the Pi: its red LED, always fixed, only indicates the power supply status, and its green LED only furtively indicates read/write operations.

Impossible to know if writings are still pending when unplugging the power! I therefore forced the status of the red led in the PiSCOPE settings on the only

other option available: the beating heart (heartbeat) that rhythms the activity of the system according to its load. As soon as the system is stopped, the red LED becomes fixed at the very moment when the green LED flickers with its ultimate writings.

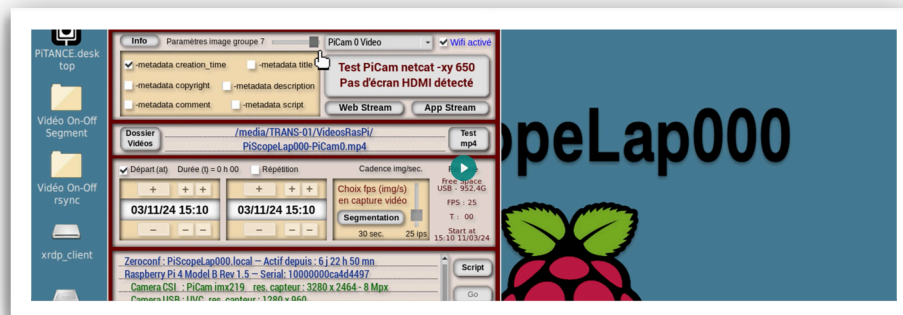


Video: of to beat, his heart stopped...

This way, you can take your eyes off the Raspberry Pi after a shutdown request: if the red LED remains fixed, you can calmly unplug the Pi while being sure that the system is well stopped.

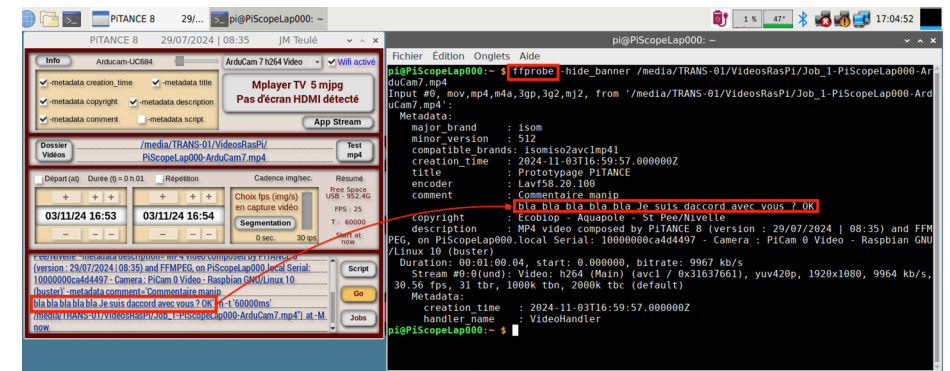
METADATA, TIME CODE OF USB CAMERAS

PiTANCE allows you to insert your own informations into each video.

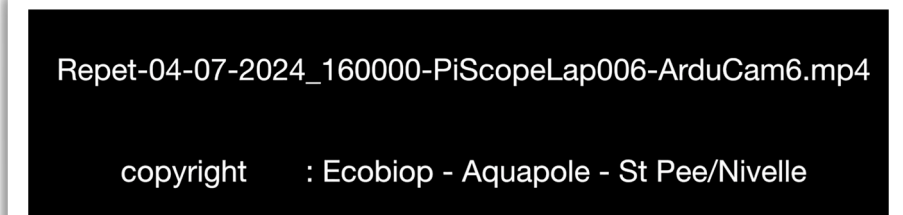


Video: enable/disable or modify metadata

To exploit these metadata it will be necessary to extract them from video with fprobe (part of ffmpeg installation).

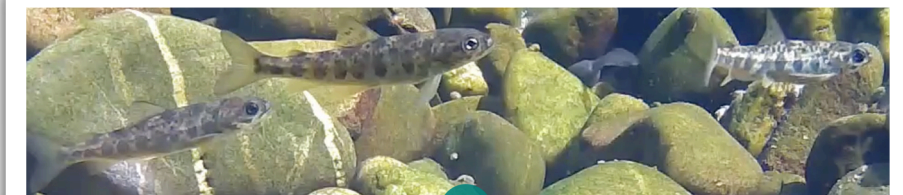


Example: you can compose a video introduction with this metadata.



Video: introduction displaying day's parameters in real time + part of the metadata recorded by ffmpeg

To make the time code of the USB cameras visible on the video, post-production with a third-party editor is necessary.



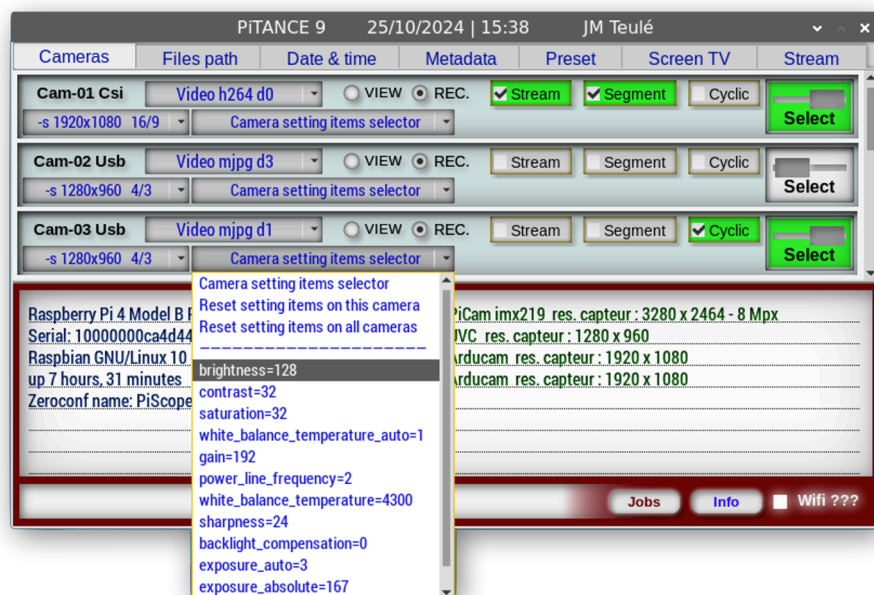
Video: overprinting of the video time code and other information with DaVinci Resolve (no other image editing)

PiTANCE and PiSCOPE system post LAP 2024 evolution and prospects

PiTANCE SOFTWARE

Following this field experience, it appears that PiTANCE 8 is simply no longer up to the job to program simultaneously and easily several cameras...

PiTANCE 9 is already under construction.



But it's a complete complete rewriting of the software that begins!

- handle up to 8 cameras per Pi in parallel for simultaneous video acquisition (or not), depending on the memory limits of the Raspberry Pi used and the power supply to the cameras,
- programming of complex time and duration and their repetitions,
- try to keep a readable and usable interface on a 5" screen,
- make PiTANCE a more international application (interface and display of dates and times currently only in French format), more compatible with citations in future publications,

- split the programming of USB camera recording and their image parameters to program a progressive variation in image quality during dawns and twilights or similar situations,
- consider the simultaneous programming of several PiSCOPEs with identical cameras on the same network,
- treat, also, IP cameras.

PiSCOPE SYSTEM

The present official operating system for Raspberry Pi has abandoned the historic raspivid (video) and raspistill (photo) libraries that were usable until Raspbian BUSTER, replacing them with new ones..

For reasons of stability and lack of time to evaluate them as they came along, the PiSCOPEs remained with an operating system based on Raspbian BUSTER, until the new software libraries became stable in use and third-party applications became compatible with them.

It's now high time to test these new rpicas-apps libraries for the PiSCOPE, and PiTANCE 9 should gradually make the transition to the latest Raspberry Pi 5 and Zero 2.

A significant adaptation in all.

But which could be useful to many people:

- ✓ You're a research unit, a laboratory, a researcher, or anyone after all...
 - ✓ You have one or more Raspberry Pi(s), dedicated to various uses or just waiting in a drawer... or a small budget...
 - ✓ You have a few accessible USB webcams, possibly an active USB hub and USB storage...
 - ✓ You'd like to simultaneously record videos of several laboratory experiments, or of one experiment from several angles, or do photogrammetry, or anything else with several cameras at the same time, or in delayed time...
- ➡ You download the full image of the PiSCOPE system (coming soon)
- ➡ You flash it onto an SD memory card, and your Raspberry Pi is transformed into a ready-to-use multi-video acquisition system...

ARDUCAM B0200 CAMERAS USED FOR LAPITXURI

These cameras do not have a unique identifier although a Serial line is recoverable with the following shell command:

```
pi@PiScopeLap000:~$ v4l2-ctl --all --device /dev/video19
```

`/dev/video19` is an arbitrary non-permanent device number assigned by the system to each USB connection. We have the name of this camera (Model line) but no unique serial number, just the reference to its generic series: UC684.

```
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
```

Logitech webcams, on the other hand, give a unique serial number on the same line.

It's a shame not to be able to individually identify each camera in a whole group of identical models.

However, ArduCam offers software to customize the serial number and even the name of their cameras.

After installing and launching it, it returns the following message:

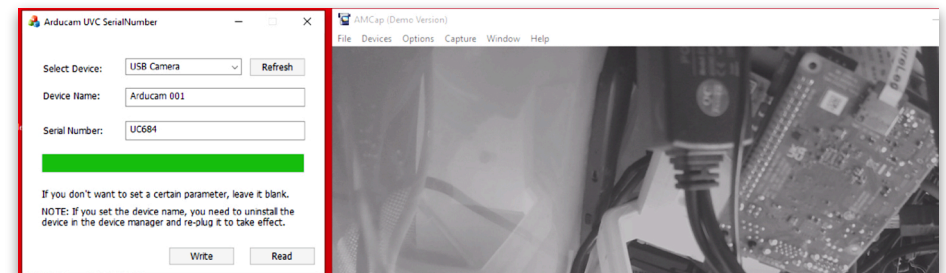
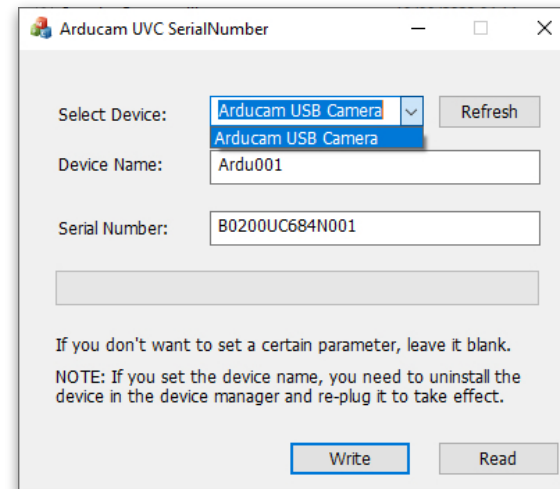
"Unknown Device Error Please Contact Arducam Support for Camera Confirmation and Technical Support."

"Some cameras in the table below only support modifying the serial number with the latest firmware version."

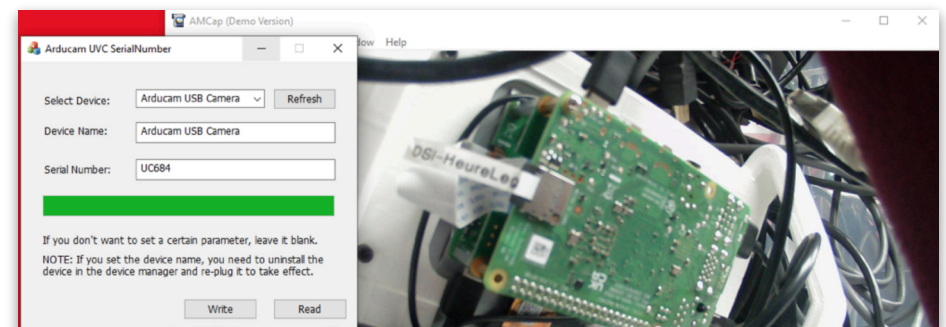
On February 13, 2024, an exchange begins with Arducam customersupport, in China.

They proposed a new compatible version with the B0200. Only on Windows.

Result: whether we change the name or serial number of the Arducam B0200, once this camera reconnected to the PC, it only works in black and white mode on Windows and is no longer recognized by the V4L2 driver on the Raspberry Pi's Raspbian Buster.



Fortunately, by putting back exactly his original name and serial number of, all work normally again.



Their answer to this issue:

Hi,

Oh, sorry for the the bad experience we brought.

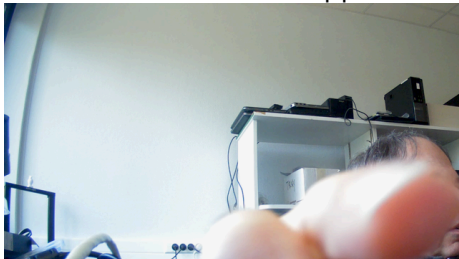
Fortunately, we have find the reason whichs causes the problem. Please burn the new firmware to your camera modules by following the guide video (I have put the burnApp、new firmware and guide video in the link below)

Besides, may you try a new program to modify your device names?

With their new program, renaming the camera or changing the serial number finally works, but the camera loses a very large number of factory resolutions, which is a pity for my usage...

I particularly regret the loss of the highest resolutions in 4:3...

But the worst thing is that with the 1280x720 resolution, the image is cropped and decentralized to the upper left corner



1280x720 of the modified B0200 compared to the same factory resolution.



1280x720 of an original B0200

And here the situation is no longer reversible.

I am kindly ending up with a partially bricked camera!...

**Their answer:**

Hi,

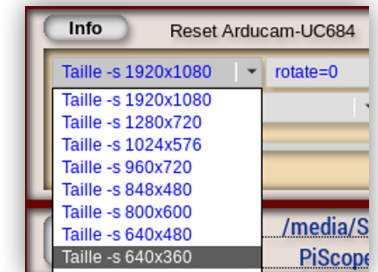
Got you.

Considering the cross-platform compatibility, we delete the non-standard resolution from the new firmware so that you can just see the four types of relution in the new firmware. Can you tell us which relutions do you need? May you list all the resolutions and FOV you need and we could make a specifich firmware for you? Ideally, the specifich firmware will also solve the off -center problem :-)

Mine:

Hi

I am interested in possible video resolutions in 16:9 and 4:3 from 1920x1080 to 640x360, but not those below ...Can you provide me, in the meantime, the original firmware that I can put the camera back in the factory configuration?

**Their answer:**

Hi,

Sorry to have kept you waiting :-)

We updated the firmware but there are only four types of stable resolution,so that we just kept 1920*1080 /1280*720 /640*480 / 320*240. Besides, we do not know why the 720p is not same as other resolutions in MJPG format, though we tried many ways in the firmware tool. We are still working on it and will inform you in time if any progress. BTW, in YUV format, every resolution is normal as expected.

My last answer, tired of all this despite their obvious good will...

Hello,

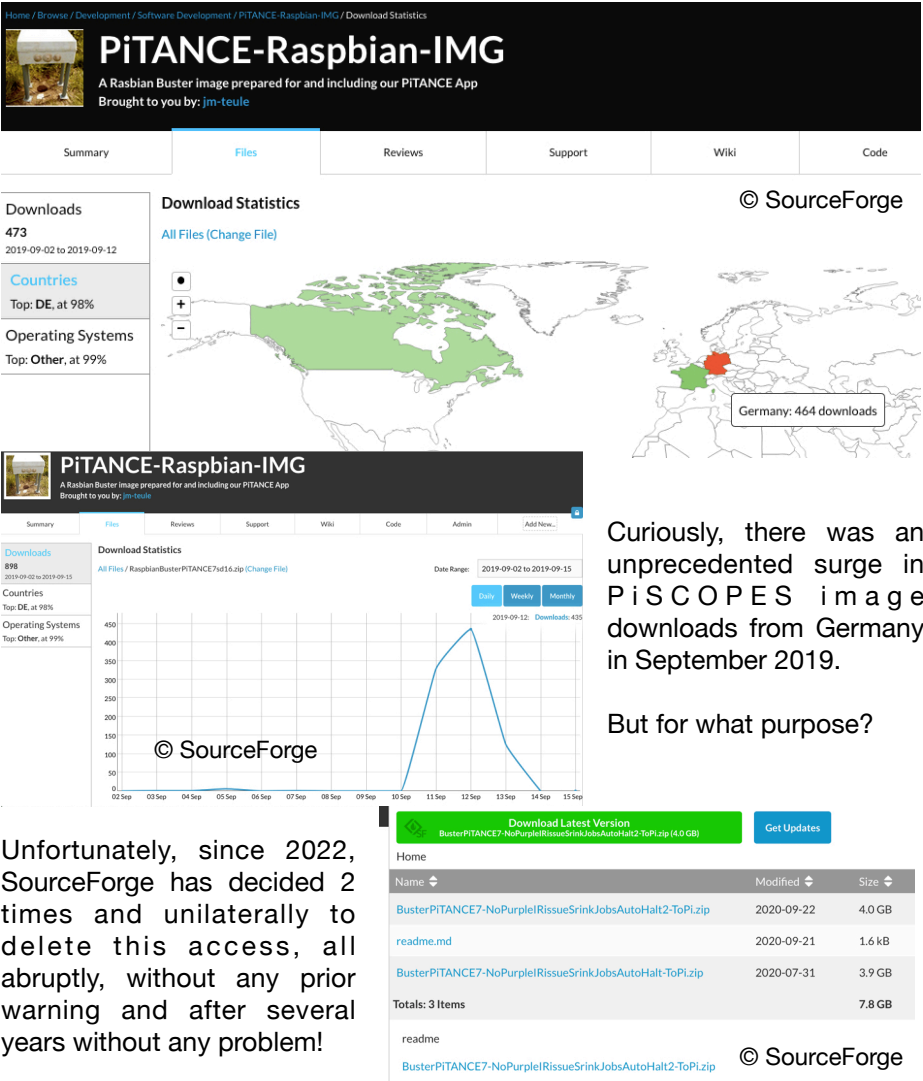
Ok, it seems to me that we are in a kind of impasse. I need to work without encoding to obtain mp4 video files directly and it seems to me that this is not possible with the raw format (which I'm not in the habit of using)...

I haven't contacted them since their return on March 18, 2024!

The question of giving these cameras a unique identifier remains unresolved.

HOSTING AND DISTRIBUTION

Until version 7 of PiTANCE, software resources and a complete, ready-to-use PiSCOPE image were hosted by SourceForge.



Unfortunately, since 2022, SourceForge has decided 2 times and unilaterally to delete this access, all abruptly, without any prior warning and after several years without any problem!

Their response to my request for explanation:

The project was using a large amount of disk space, had no recent downloads, and no source code detected. If there is source code you can publish, then we can restore it for you to upload the source but you will also have to reduce the file sizes. Our terms of use require source code to be provided on all projects. Sincerely,

SourceForge Support

I've cancelled a second attempt to restore this space, even though it's annoying for past publications with dead links.

The development of PiTANCE 8 therefore remained localized in the Ecobiop unit, which was not too annoying in a phase of reconstruction and adaptation of this system to the needs of the unit.

Now we need a sustainable hosting and/or a forge for public dissemination and to version the new iterations of PiTANCE.

The implementation of the GitLAB INRAE forge should meet this need. A deployment stage on this tool is planned for the first quarter of 2025.

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

INTELLECTUAL PROPERTY OF CONTENT:



Attribution (BY)
No Commercial Use (NC)
Share in the Same Conditions (SA)

Credit:

- photos PiSCOPE / PiTANCE / Field situations : Jean-Marc Teulé / INRAE
- vidéos PiSCOPE / PiTANCE / Field situations : Jean-Marc Teulé / INRAE
- demos, animations, illustrations PiSCOPE / PiTANCE : Jean-Marc Teulé / INRAE

Software PiTANCE : Licence GPL-3.0