HOW TO BUILD AND USE “PICT”

A user-friendly practical guide

Droissart V., Azandi L., Onguene E. R., Savignac M., Smith T.B., Deblauwe V.*

This guide is a supplement to the peer-reviewed paper: **PICT: A low cost, modular, open-source camera trap system to study plant–insect interactions** *Methods in Ecology and Evolution* (Droissart V., Azandi L., Onguene E. R., Savignac M., Smith T.B., Deblauwe V., 2021). https://doi.org/10.1111/2041-210X.13618

Disclaimer. Every effort has been made to check the content of this guide for accuracy and completeness. However, the authors cannot be held responsible for any errors or omissions. The readers remain responsible for their safety when building, mounting, and using the equipment and software products described in this guide. The readers must also comply with all applicable laws and regulations regarding the activities they intend to do with it. The authors of this guide cannot be held responsible for the results of action, inaction, or otherwise taken as a consequence of the information provided in this guide. The brands cited in the text are mentioned for illustration only and do not denote a commercial relationship between the authors and the commercial brands.

* Corresponding author: Vincent Deblauwe, BP 2008 (Messa), Yaounde, Cameroon, v.deblauwe@cgiar.org.

Document version 2.0.0.

Contact us

Found a bug? Have a question or want to discuss new idea?

Visit our GitHub Page https://github.com/Plant-insect-Interactions-Camera-Trap/pict/discussions

What is new in version 2.0.0 (this version)?

- Add support for Raspberry Pi Zero 2 W (Green led deactivation issue fixed)
- Add support for Raspberry Pi OS Bullseye (Picamera installation bug fixed)
- Add information on compatible lens
- More troubleshooting info
- Minor edits, typos, etc.

This guide is a work in progress

Look for the latest version at: https://doi.org/10.5281/zenodo.4139838

Table of contents

List of acronyms

FTP	File Transfer Protocol
IR	Infrared
IR CUT	Infrared cut-off filter
LED	Light-Emitting Diode
NoIR	No Infrared filter
PICT	Plant insects Interaction Camera Trap
SFTP	Secure File Transfer Protocol
SSH	Secure Shell
SSID	Service Set IDentifier
OS	Operation System

I. Foreword

1. What is PICT?

The **P**lant insect **I**nteractions **C**amera **T**rap (PICT), is a DIY, low-cost, weather-resistant camera designed to be autonomous, energy efficient and modular. Its architecture is based on a Raspberry Pi single board computer. The focus distance of PICT can be manually adjusted to under 5 cm. PICT is primarily designed for the study of plant-insect interactions, but it can be used for a variety of purposes including, and not limited to, bird feeding or nesting, amphibian nocturnal activities, predation by reptiles, insect behaviour, etc. It is the best alternative in any observational settings requiring a cheap, autonomous, remotely controlled and portable system to record images or videos for long periods of time.

Figure 1. *Frames extracted from video clips captured with PICT using a 3.6mm lens without IR filter and the Raspberry Pi v.1 camera. Clockwise from bottom left: A. Honeybee,* Apis mellifera *Linn., on a flower of African ebony (*Diospyros crassiflora *Hiern); B. Sphingidae on a flower of African ebony; C.* Ceratina *sp. on a flower of African ebony; D.* Xanthopan morganii *Walker on a flower of* Cyrtorchis chailluana *(Hook. F.) Schltr. orchid. (A-C) Mbalmayo Forest Reserve, Cameroon, credit Vincent Deblauwe. D. Yaounde, Cameroon, credit Vincent Droissart.*

2. How to use this guide

The purpose of this guide is to provide a detailed explanation of PICT construction, set-up, and use. It is intended to be accessible to anyone with minimal handiwork or computer skills. Advanced skills such as electronic and programming language are not required to use this manual. However, **we advise you to read this guide in its entirety before starting to build and use the camera.**

PICT is controlled using either a smartphone, a computer or a tablet. To provide maximal flexibility, we have successfully tested each step on devices running Windows (7 and 10), Android (7-10), macOS and iOS. Using Linux based OS is also possible and even easier, in particular, to access to your recordings. In this guide, we give you the choice between different software for each step in order to try a different way if you get stuck with one.

Inside this guide you are going to find boxes providing extra information, following this colour code:

Light orange boxes: hardware needed.

Green boxes: tips and frequent mistakes.

Pink boxes: command lines to be typed in.

Dark Orange boxes: warnings.

Yellow boxes: additional information for a deeper understanding.

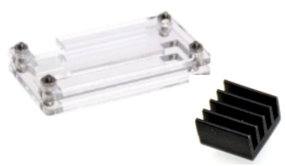
II. Hardware

All the material you need to build a PICT can be purchased at a local computer and hardware store. Attractive prices can be obtained from online sellers, but one needs to take into account delivery times depending on country of residence. At the time of writing (December 2020), the total cost of one camera without accessories (the battery, memory card, and camera mount) was less than USD 100.

1. Tools

✓ Box cutter or rotary drill	→ To cut a hole in the waterproof container
✓ Sandpaper	→ To make the surface rougher before applying glue
✓ Hot glue gun + glue sticks or silicon	→ To glue the protective lens to the container
✓ Epoxy adhesives	→ To glue the Velcro® strips and the case mount

2. PICT parts

What do we need it for?	Components	Details
Recording	**Raspberry Pi Zero W** 	The model “Zero” is the world's smallest, cheapest, and most energy-efficient single-board computer that runs a Linux operating system. The « W » model includes an on-board chip antenna for Wi-Fi and Bluetooth. Visit https://www.raspberrypi.org/ to locate an official seller for your country.
	Raspberry Pi Zero 2 W 	The second version of Raspberry Pi Zero was released in 2021. Due to the CPU upgrade it use more power from the battery and can dissipate more heat. We discourage you to choose this version 2 because it will drain the batteries faster than the older, less powerful, model. We calculated that PICT with this model uses ~10% more power at night (LED on) and 16% more power during daytime (LED off).
	Raspberry Pi Zero (W) Protecting case + heatsink 	Your Raspberry needs a protective case to avoid physical or electronic damage when handled. We recommend choosing a model that comes with a heatsink to improve heat dissipation in the sealed food container. Avoid models like the official raspberry case, that have to be opened each time you need to remove the SD card.
	NoIR (i.e. night vision) camera module with IR LED	The Raspberry Pi camera v. 1 without IR filter (NoIR). The camera is equipped with a 3.6 mm adjustable focus lens. The field of view is ca. 10 x 7 cm at 10 cm from the top of the lens. The lens is interchangeable with any M12 socket (S-mount) lens. The lack of IR filter improves signal to noise ratio in low light conditions and allows the use of an IR LED to

illuminate the scene. The images produced appear pinkish because IR light is not filtered out like in conventional cameras.

IR-CUT NoIR (i.e. night vision) camera module with IR LED

This is optional. Note that camera modules with an IR-cut filter are available. In these cameras the IR filter is toggled in and out according to ambient light intensity, allowing you to make true colour recording during the day and IR light recording at night. We do not recommend this type of camera because the coordination of the LED and IR-filter by independent photosensitive resistors can be problematic, thus resulting in completely black recordings on some occasions (e.g. at sunset and sunrise).

Extra M12 socket (S-mount) lens

This is optional. The 3.6mm lens above is quite polyvalent and ideal to record- insect > 1cm long at a distance of 10 cm. If you need to observe anatomic details of smaller insects, then consider a larger focal length. An 8 mm lens would be appropriate for insects 0.5—2 cm long. With a 16 mm lens you can observe insects 0.1—1 cm long. Note that the depth of field is inversely proportional to the focal length of the lens.

Camera module ribbon cable for Raspberry Pi Zero and Zero W

Raspberry camera modules currently available on the market are compatible with all Raspberry Pi models. However, the ribbon cable that is provided with the camera is not. The camera cable port of the Pi Zero is 1.1 cm wide, while the port of other Raspberry Pi version and the port on the camera are 1.6 cm wide. The yellow ribbon cable has one narrow end for the port on the Raspberry and one wide end for the camera. It is needed to connect the camera module to the Raspberry Pi Zero (W). The white ribbon cable with blue stripes that is often provided with the camera is 1.6 cm wide at both ends and is not suited for the Pi Zero camera port. You should either purchase a camera with the cable compatible with Raspberry Pi Zero or purchase the right cable separately.

Micro SDXC card + Micro SD to SD adapter

We recommend choosing a micro SD card with 64Go capacity or more. Class 10 (or UHS-1) speed is enough for the applications described in this guide. On a 64GB micro SD card, once the operating system is installed, 57GB will remain free for the recording of data. One-hour of video at default compression, recommended resolution (1296x972) and FPS (15) is around 700MB. Note that the actual size of the video may vary with the complexity of the scene and the amplitude of motion in the background and foreground. At night the background appears black and one hour of video will often require less than 400MB. An adapter might be required to insert the micro SD card into your computer. Note that using a unique size of SD card will reduce your workload during operation.

Powering	**Power banks** 	It is the battery that will power the Raspberry Pi Zero (W) and the camera module. If you plan to travel by plane with your PICT, we recommend choosing power banks up to 26 500 mAh (100 Wh) as anything above that limit is restricted by most airline companies at the time of writing. Otherwise, choose a capacity that suits your needs following the consumption displayed in the supplementary material of the paper. To operate one PICT continuously, you need at least two power banks, to have one charging while the other is powering PICT. The specifications to look for: • Best storage to price ratio. • No need to press a button to switch the power bank on. This kind of model is prone to unwanted interruption of power. • If you want to use a solar panel as a power source during recording, you want a power bank that can be charged while powering a device. This technology is called “pass-through charging”. • Fit into the food container while the USB cable is plugged in.
	A-Male to micro B USB cable 	To connect the Raspberry Pi Zero (W) to the power bank (it is the typical cable of Android phones). This cable is sometimes provided with the power bank. We suggest purchasing a cable with a “L” shaped USB connector to minimize the space occupied by the connector in the food container.
	Multi-port USB charger with high output amperage 	To charge multiple power banks simultaneously. Note that some batteries can use two cables and/or high output amperage technology (e.g. Qualcomm® Quick Charge™ 3) to charge at their maximum speed. A 20 000 mAh battery will usually need at least 8 hours to be fully charged.
Housing	**Action sports camera protective lens** 	The protective lens can sometimes be a UV filter. UV filtering does not alter image quality. The lens can be of any shape but its diameter needs to be at least 3 cm.
	Rectangular food storage container with a volume around 1000 ml (33.81 oz)	The food container should be the smallest container able to enclose the Raspberry Pi, the camera module, and the power bank with the USB cable plugged in. The lid should be translucent to transmit the IR light emitted by the LED, and completely waterproof. We used Lock & Lock model HPL817 (20.5*13.4*6.9 cm) or HPL842 (27.8*11.5*6 cm).

		
	Velcro® strip (~30 cm of for one PICT) 	To non-permanently assemble together the camera, the Raspberry Pi Zero (W), the power bank and the food container. Avoid strips with "adhesive" on the back surface because the adhesive is not strong enough to withstand the tensions occurring during use. We recommend instead to glue the Velcro® strip with 2 part epoxy adhesives (e.g. Araldite®).
Mounting	**Mount with standard ¼" screw** 	Any tablet or smartphone holder that can easily be glued to the food container (e.g. Rollei tablet holder).
	"Magic arm with clamp" or "Multi-function ball head clamp" 	To mount the PICT on small branches from 1.5 to 6 cm in diameter.
	Slate River Camera Arm with T-Handle extension 	This type of mount is to be screwed in a trunk or branch from 5 cm in diameter onwards.
	Slate River E-Aim Ratchet Strap Camera Mount 	A strip to mount the PICT on a tree trunk from 10 cm onwards.

Controlling	**A tablet or a smartphone**	The PICT can be entirely set up and operated from an Android device. A properly functioning low end device with long battery life and bright screen for outdoor use should be preferred. Water and dust resistance (such as IP68 rating) is a plus. We have tested the compatibility of suggested apps on Android versions 7, 8, 9, 10 and 12.
	A computer	A computer is required to install the operating system of the Raspberry Pi Zero (W) on the SD card and to post-process and analyse your recordings. The procedures described in this guide were successfully performed on computers running on Windows 10 and 7, as well as on macOS 10.13.

III. Building PICT

In the following section, we will go through the construction of the PICT step by step, starting with the electronic components, the case, and finally the mounting system.

1. PICT's electronics

In this section, we will assemble the electronic components to obtain the setup shown in Figure 1.

- You will first need to assemble the protective case of the Raspberry Pi Zero (W) following the instructions of the case manufacturer. If a heatsink was provided with the case, you can attach it on top of the Raspberry Pi processor (optional).
- We suggest using Velcro® strips to non-permanently fix the camera module and the Raspberry Pi on the power bank. Glue a strip of Velcro under the Raspberry Pi protective case and on top of the power bank with 2-part epoxy adhesives (see Fig. 1.c). Be careful to use the matching side of the Velcro strip for each couple of objects.

Tip: before the application of the glue, rub the surfaces with sandpaper to maximise adhesion.

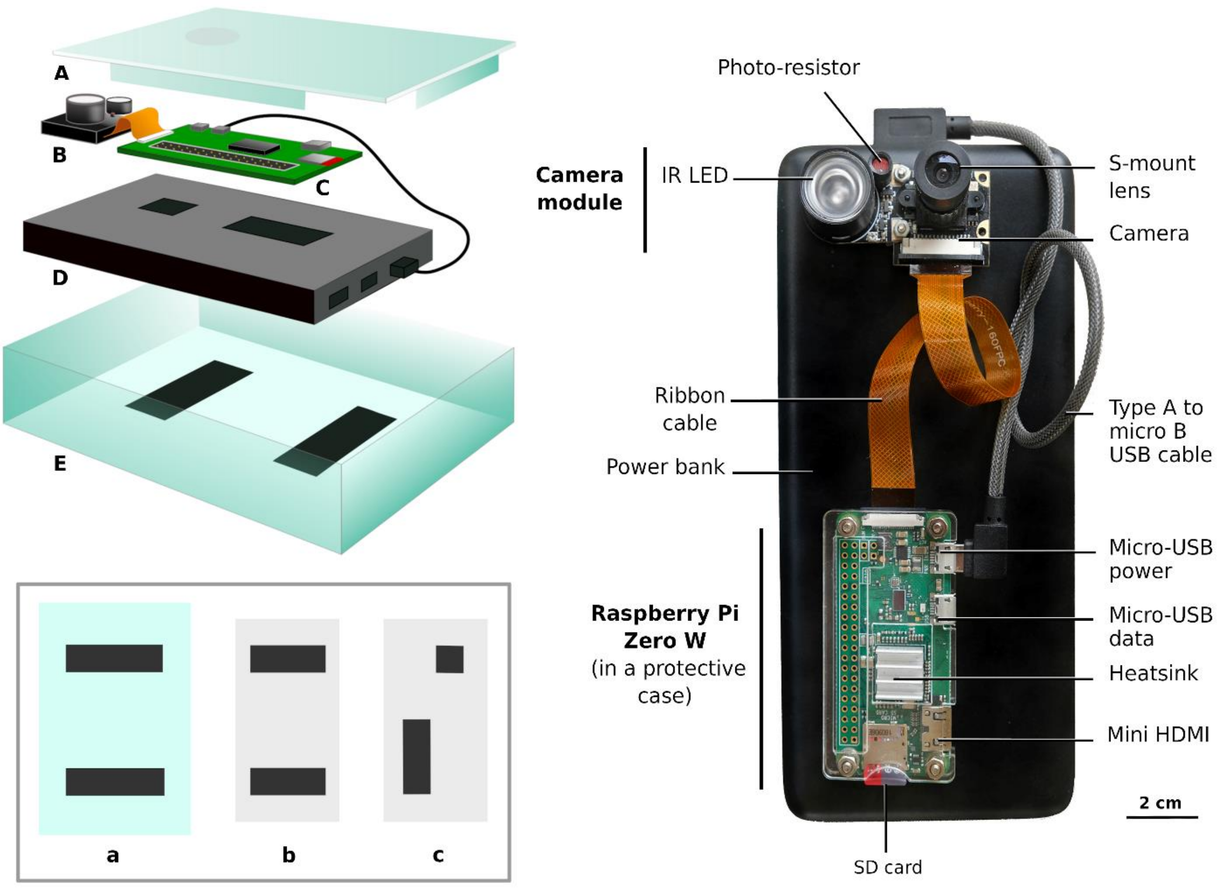


***Figure 2.** PICT components and assembly. **A.** Food container (lid), **B.** Camera module, **C.** Raspberry Pi Zero (W), **D.** Power bank, **E.** Food container (bottom). **Framed:** Location of the epoxy-glued Velcro® strips. **a.** Location of the Velcro strips on the inside bottom of the food container, **b.** location of the Velcro strips on the lower side of the Power bank (which is going to be in contact with the inside bottom of the food container), **c**. location of the Velcro strips on the upper side of the Power bank for the camera (up) and the Raspberry Pi Zero (W) (down).*

Tip: be careful to choose the same Velcro® strips configuration (side, position, length) for each PICT if you plan to build several of them. This will allow you to exchange parts between PICTs.

- Attach one of the LEDs to the camera with bolts and nuts provided by the manufacturer. Two LEDs are provided but one is enough for the application presented here. A second LED would increase the power drawn from the power bank.
- Add a square piece of Velcro® strip at the back of the camera module and the matching piece of Velcro strip to the power bank at the position shown in Fig. 1.c.
- Attach the yellow ribbon cable to the camera module and the Raspberry Pi Zero (W). To insert the cable into the port, you need to gently pull up on the edges of the port's plastic clip on the Raspberry Pi Zero (W) and the camera module. Make sure that the metallic connectors of the ribbon cable are facing the conductive pins of the port. Once the cable is fully inserted, push the plastic clip back into place.
- Attach the camera module and the Raspberry Pi Zero (W) to the power bank with the matching Velcro strips. The result should look like Figure 1 except that the USB cable is not plugged in yet.

2. PICT housing

- Choose the location of the power bank inside the food container so that there will be enough room for the USB cable once plugged in (Fig.1.a). Glue the two Velcro® strips inside the food container. Glue the matching side of Velcro on the power bank (Fig.1.b). We advise installing two parallel strips of Velcro, 10 cm apart.
- Once the power bank, the USB cable, the Raspberry Pi and the camera module are in place in the box, close it, and draw with a pen the future location of the translucent protective lens on the plastic lid. The protective lens should be placed in front of the camera. The IR LED will illuminate the scene through the translucent plastic lid.
- Remove the lid and cut a disk of plastic off with a box cutter (or with sharp chisels) to make room for the insertion of the protective lens. Be careful to apply gentle pressure so as not to cut yourself or to break the lid during this process by applying too much pressure on it.
- Place the protective lens in the lid and glue it with the glue gun or with silicone. Apply enough glue or silicone so that the cover remains waterproof.

Tip: *Humidity can damage electronic devices so do not touch your PICT with wet hands and to the extent possible, avoid opening the food container (PICT case) in wet environments. We also recommend the use of silica gel wrapped in a small paper bag to keep the interior of the container dry and avoid condensation. You should replace the silica gel after each use. Check the state of the food container gasket regularly and ensure there is no dirt on it as this might make it less waterproof.*

3. PICT mount

Once the PICT parts are assembled, you need to add a fixation on the container to mount the camera on a tree, a pole, or a tripod and to accurately point the camera towards the subject of your study. To achieve this, we recommend gluing on the container a piece of plastic or metal with a standard ¼ in screw (see Fig. 2). Be careful to fix it on an area where it would not prevent the lid from closing hermetically.

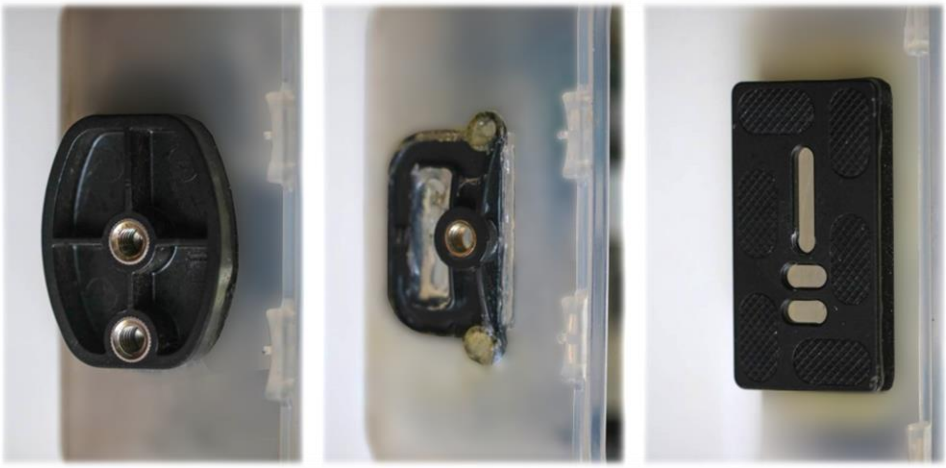


Figure 3. *Example of different mounts with standard ¼ in screw that can be glued on the case of the PICT. Two part epoxy adhesive is recommended to permanently fix the mount on the food container.*

Figure 4. *Example of various camera mounts that we successfully tested in the field.*

IV. Getting started

1. PICT operating system (Raspberry Pi OS) installation

The micro SD card is the hard drive of your PICT. Raspberry Pi OS (previously called Raspbian) is the Raspberry Pi Foundation's officially supported open-source Linux based operating system. For our project, we recommend **Raspberry Pi OS 32-bit Lite** because it uses fewer system resources and occupies less space on the micro SD card than other versions.

- ✓ Micro SD card + Micro SD to SD adapter
- ✓ Computer running Windows 7 or 10 or macOS
- ✓ Software:
  - SD Memory Card Formatter (compatible Windows/macOS)
  - Win32diskimager (Windows only) or BalenaEtcher (compatible Windows/macOS)
- ✓ Raspberry PI OS 32-bit Lite

- Download the image of the latest version of **Raspberry Pi OS 32-bit Lite** from https://www.raspberrypi.org/downloads/raspberry-pi-os/.

Note that we tested successfully Raspberry Pi OS 32-bit Lite bullseye version of January 28th 2022 and the Lite legacy buster version of January 28th 2022. Do not choose a 64bit version, it will not work.

- Download and install the freeware **SD Memory Card Formatter** available at https://www.sdcard.org/downloads/formatter/.
- Insert the micro SD card into the micro SD to SD card adapter and plug the adapter into your computer SD slot. If your computer doesn't have an integrated card reader, you can use an USB SD card reader. Start SD Memory Card Formatter. Keep default settings and select the letter of your SD card reader in the "Select card" field (Fig. 5.1). In our example, it is the letter "E", but it might be different on your computer.

Warning: If you have several external disks plugged into your computer, be sure to select and format the right one! The safest way is to disconnect every other external disk and USB key.

.

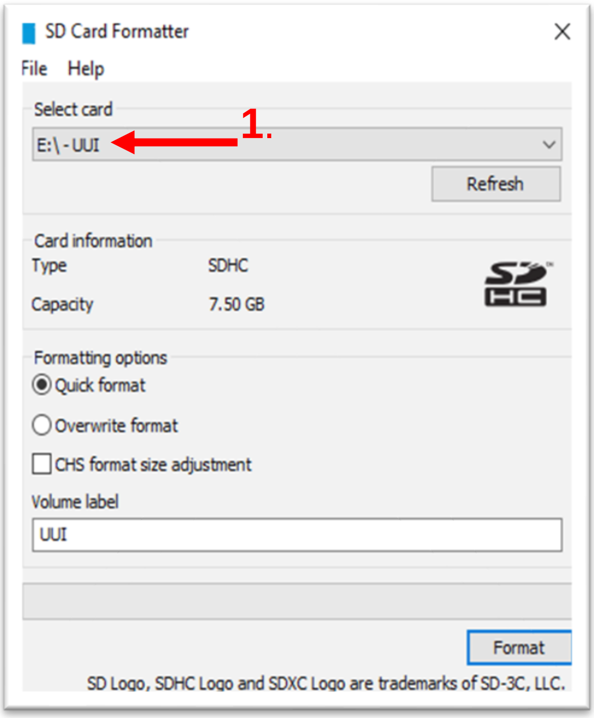


Figure 5. *The SD Card Formatter interface*

- The Raspberry Pi OS image is available for download as a ZIP archive in ZIP64 format (about 400MB) that you need to unzip. To uncompress the archive, an unzip tool that supports ZIP64 is required. This can be achieved using **7-Zip** (https://www.7-zip.org/) a free and open-source software running on Windows, or **The Unarchiver** on macOS (https://theunarchiver.com). The uncompressed image is over 1.7GB in size.
- Download the open-source software **Win32diskimager**, freely available at https://sourceforge.net/projects/win32diskimager/. If you meet problems with this software, you can also use the software **balena Etcher** (Windows and macOS), freely available at https://www.balena.io/etcher/.
- Run **Win32diskimager**. Select the xxx.img file you previously unzipped in the "Image File" field (Fig.6.1). Select the letter of your SD card reader in the "Device" field (Fig. 6.2).
- Leave all other settings at their default values. Click on the "Write" button. After a few minutes, the card will be written. Two partitions will be created on the SD card. The small one is called "boot" and is formatted in FAT32. It can be read by windows or other exploitation systems and we will use it in the next step. The second one is a large Linux partition. Ignore the request of Windows to format that partition. This is because Windows cannot handle Linux partitions without a third-party software.

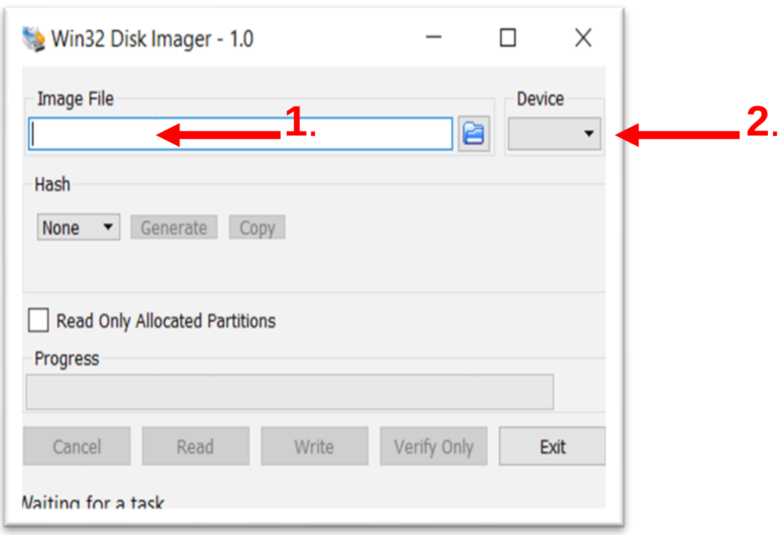


Figure 6*. The Win32 Disk Imager interface*

2. Configuration of PICT

- ✓ Internet access
- ✓ Computer running on Windows or macOS
- ✓ Computer software:
  - Termius (all OS)/PuTTY (Windows only)
- ✓ Smartphone software:
  - Termius (all OS)/JuiceSSH (Android only)

Tip: If you want to build several PICTs, you can also use Win32 Disk Imager *to create an image file of your micro SD card including the operating system and its final configuration and then write it on multiple other micro SD cards.*

1.1. Configure the PICT Wi-Fi and SSH connection

To be able to communicate with your PICT and to modify its configuration, you will need to set up its Wi-Fi parameters and activate the SSH protocol.

- Open **Notepad** by using the Start Menu of Windows (or **TextEdit** on macOS).
- You will now enter the list of Wi-Fi networks that your PICT will connect to. You can either create the list using the **Raspberry Pi Wi-Fi Config Generator** available at https://steveedson.co.uk/tools/wpa/ or you can use our example below. Note that the Generator will not provide the first line of the file. You will have to add it manually. Our example asks the PICT to connect to either one of two networks named *Wi-Fi network name one* and *Wi-Fi network name two* with passwords *Wi-Fi network password one* and *Wi-Fi network password two*, respectively.

  Adapt the text to match the name and password of your Wi-Fi network and copy-paste the lines in the blank screen of **Notepad**:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

network={
   ssid="Wi-Fi network name one"
   psk="Wi-Fi network password one"
   key_mgmt=WPA-PSK
   priority=2
   scan_ssid=1
}

network={
   ssid="Wi-Fi network name two"
   psk="Wi-Fi network password two"
   key_mgmt=WPA-PSK
   priority=1
   scan_ssid=1
}
```

  Note that the Raspberry Pi Zero W can only connect to 2.4GHz Wi-Fi bands (otherwise known as 802.11b/g/n standard) and not to 5Ghz bands (known as 802.11ac).

- Your PICT will connect to the available Wi-Fi network with the highest priority. By default, the priority of all networks is 0. If you set up only one network in the file then the priority command is no longer needed. Note that “scan_ssid=1” is an optional command which allows the PICT to connect to a network that is not broadcasting its SSID name.

- Double-check the SSID and the password. Be careful not to add unwanted white space or use wrong letter case. Any error will render the PICT unreachable through Wi-Fi.
- Go to **File > Save as**, name it “**wpa_supplicant.conf**” and in **Save as type**, choose: **All Files (*.*)** as shown in Figure 7. Be sure to set **Save as type** to **All Files (*.*)** (so the file is NOT saved with a .txt extension).
- Click on “**Save**”

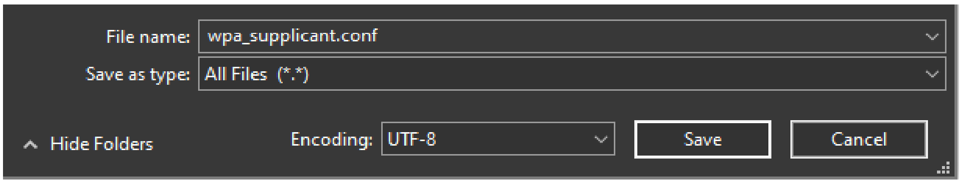


***Figure 7**. How to save the wpa_supplicant.conf file*

__Warning__: Be careful about the extension of the file. To make sure the extension is .conf (and not .txt) you can right click on the file and check in the properties.

***Figure 8**. The ssh and wpa_supplicant files copied on the SD card*

- Open Notepad again to create a new empty text file in the boot partition of the SD card. This file must be saved without extension: go to **Save as**, name it **“ssh”,** __with__ the “”, and in **Save as type**, choose **All Files (*.*).** This operation will enable SSH protocol on your PICT (Fig. 8).

- Once the two text files are copied on the SD card, remove it carefully and insert it into the Raspberry Pi Zero (W). Bootup the Raspberry by attaching the USB cable to the micro-USB power port (Fig. 9) and the power bank. Note that the Raspberry has two micro-USB connectors. The power connector is located just next to the camera connector (Camera module slot) on the short edge of the Pi. The other micro-USB connector (Micro-USB data) is used for accessories and should not be connected to the power outlet. Once powered on, the green activity LEDs will blink. After a few seconds the Raspberry will automatically connect to one of the Wi-Fi networks that were specified in the wpa_supplicant.conf file (see above).

- Note that you will be able to edit the file wpa_supplicant.conf later, provided that the Pi have connected to a WiFi network listed in the list you just created. See the section named Useful commands. If the Pi does not connect to a network, it means you have made a mistake and need to write the OS image file again.

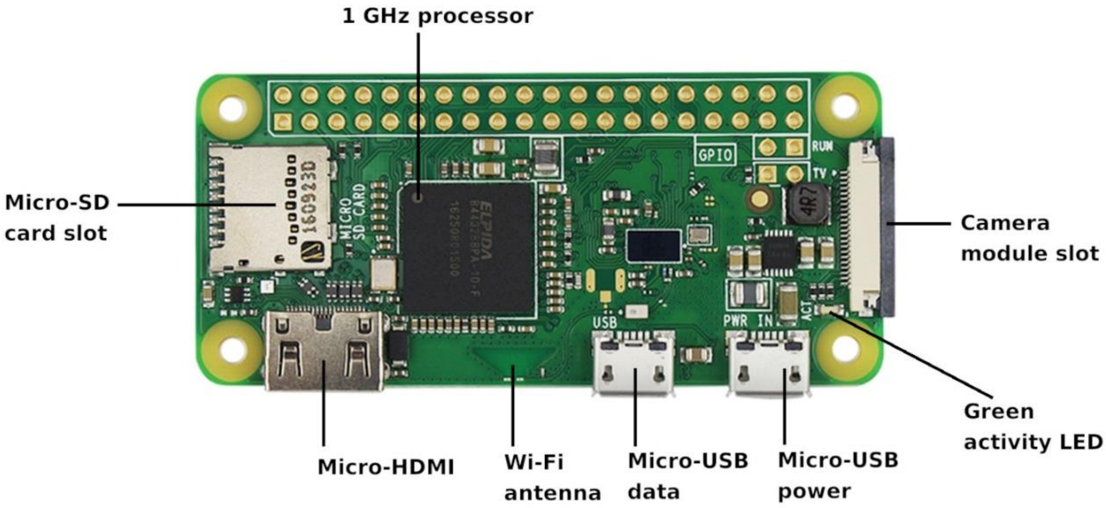


Figure 9. *Raspberry Pi Zero (W) v1 connectivity*

1.2. Several ways to connect to your PICT through SSH protocol

There are different ways to connect through SSH (see yellow box page 18 for additional information) to your PICT, but you will always need a device (computer, smartphone, tablet, or Wi-Fi router) emitting a Wi-Fi network with a name and password matching one of the networks described in the wpa_supplicant.conf file. The PICT will automatically connect to this network a few seconds after bootup. If the device broadcasting the Wi-Fi network is a smartphone, tablet or a computer (i.e. Wi-Fi hotspot), you can use that device to communicate with the PICT (Fig. 10.1). In all cases, you can communicate with the PICT using a smartphone, tablet or computer that is connected to the same Wi-Fi network as the PICT (Fig. 10.2).

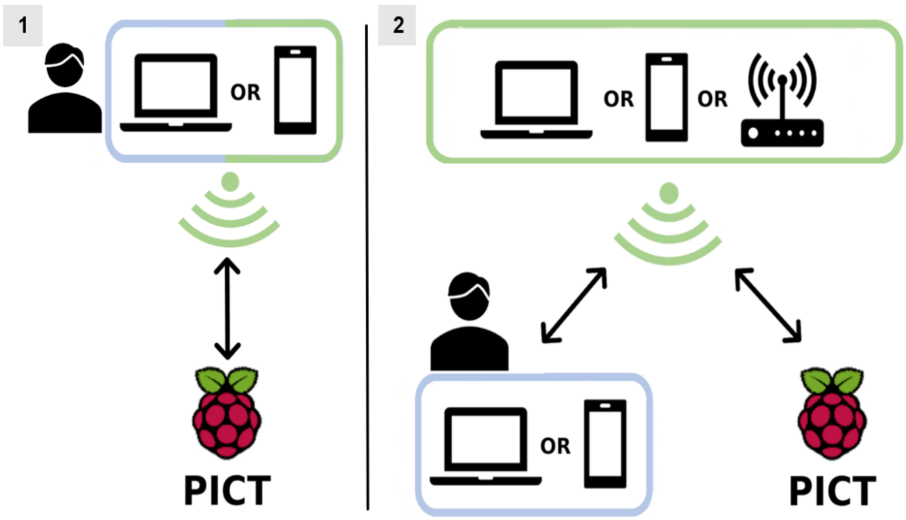


Figure 10. *The different ways to control your PICT with a remote device through a Wi-Fi network.* ***1.*** *The device that broadcasts the Wi-Fi is the controlling device;* ***2***. *The device that broadcasts the Wi-Fi and the controlling device are different. Light blue = controlling device; light green = device broadcasting network*

1.3. Create a Wi-Fi hotspot

The easiest way to communicate with your PICT is to create a Wi-Fi network with the device that will communicate with it (Fig. 10.1). The Wi-Fi network created with a portable device is commonly known as a **"Wi-Fi hotspot"**. Note that the Raspberry Pi Zero W can only connect to 2.4GHz Wi-Fi bands (otherwise known as 802.11b/g/n standard) and not to 5Ghz bands (known as 802.11ac).

a. Create a Wi-Fi hotspot on your Windows device

To create a Wi-Fi hotspot on your Windows computer, go to **Windows Settings > Network & Internet > Mobile hotspot** and turn on the **“Share my Internet connection with other devices”** option (Fig. 11).

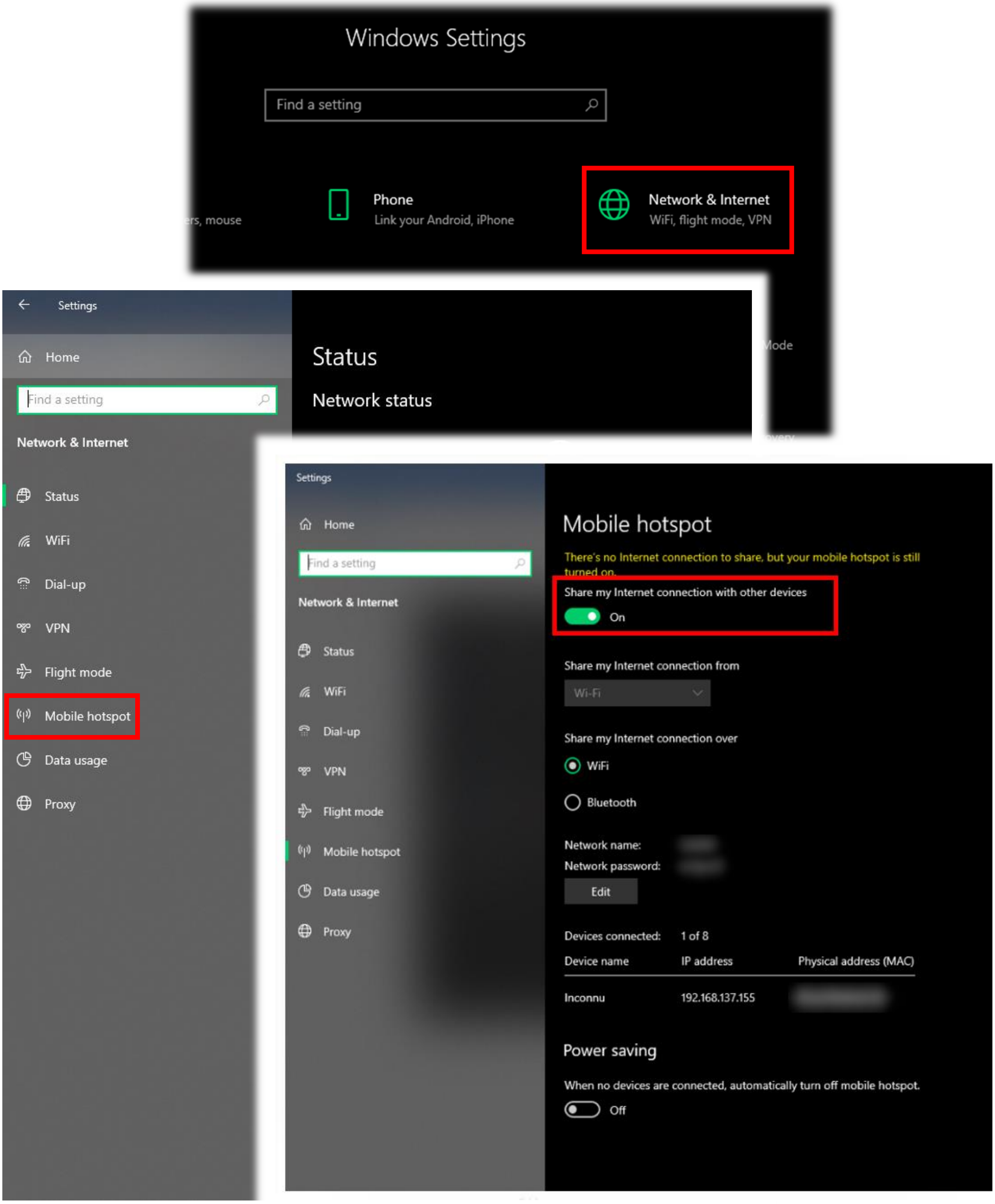


Figure 11. Create a Wi-Fi hotspot on your Windows device

b. Create a Wi-Fi hotspot on your iOS device

To set up a Hotspot on an iOS device you can follow the instructions available on the Apple support page (https://support.apple.com/en-us/HT204023).

c. Create a Wi-Fi hotspot on your Android device

Each version of Android differs slightly in the location of the Wi-Fi hotspot options but the general idea remains the same whatever the version.

For instance, on Android version 9, go to **Settings app > Network & Internet > Hotspot & tethering > Wi-Fi hotspot** and then click on **"On"** (Fig. 12). Edit the name (or SSID) and password of the Wi-Fi hotspot to make sure they match what you wrote in the wpa_supplicant.conf file.

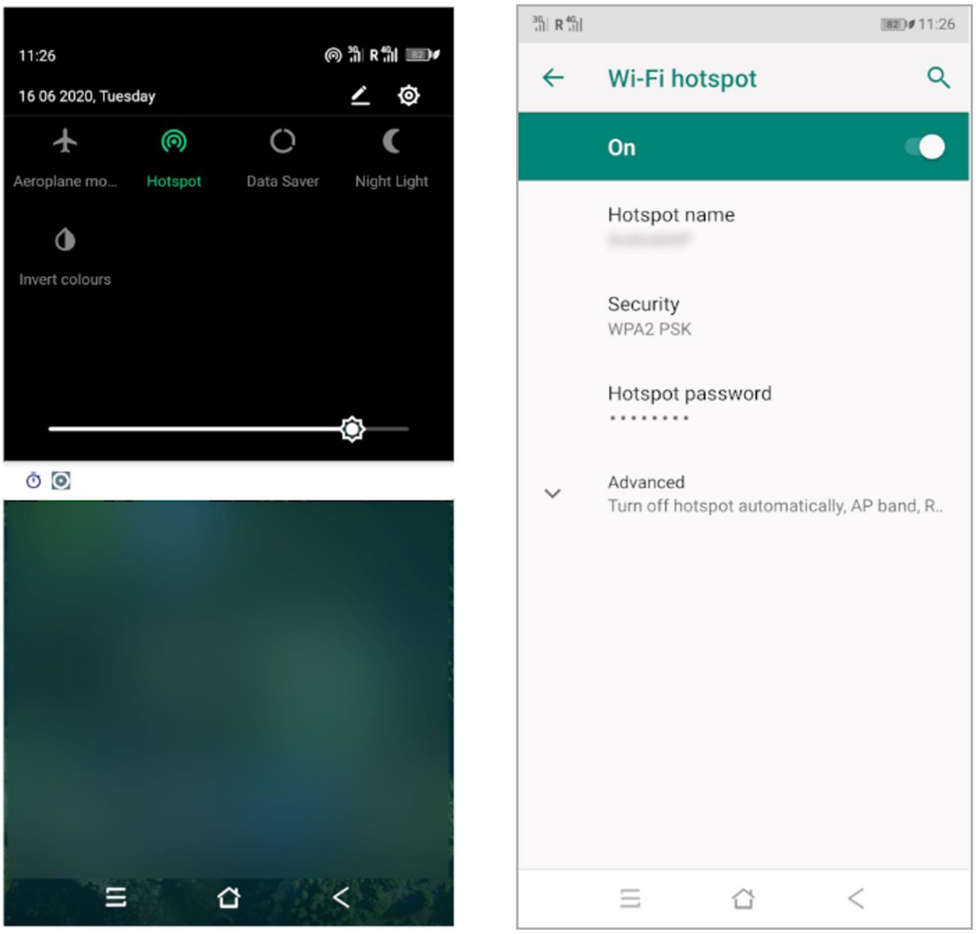


***Figure 12**. Create a Wi-Fi hotspot on your Android device*

1.4. Find your PICT IP address

When your PICT is connected to your Wi-Fi network you need to find the IP address assigned to it to be able to communicate.

If the Wi-Fi network is emitted by your local Wi-Fi router you can find the IP of every connected device in the Wi-Fi settings in the configuration page of the router. This page is displayed by entering your router IP in any web browser (usually 192.168.1.1). The IP is often written at the back of the router. Most router vendors explain the procedure in their online manuals. See for instance the official page for TP-Link (https://www.tp-link.com/us/support/faq/2392/) and Linksys (https://www.linksys.com/us/support-article?articleNum=137818) routers.

The procedures for Windows and Android devices are presented in the following section as well as some convenient software applications that scan the network and provide the IP address of all the connected devices.

Find your PICT IP address on a Windows 10 device's Hotspot

Go to **Windows Setting > Network & Internet > Mobile hotspot** to find the IP address of your PICT and any other devices connected to your computer hotspot (Fig. 13).

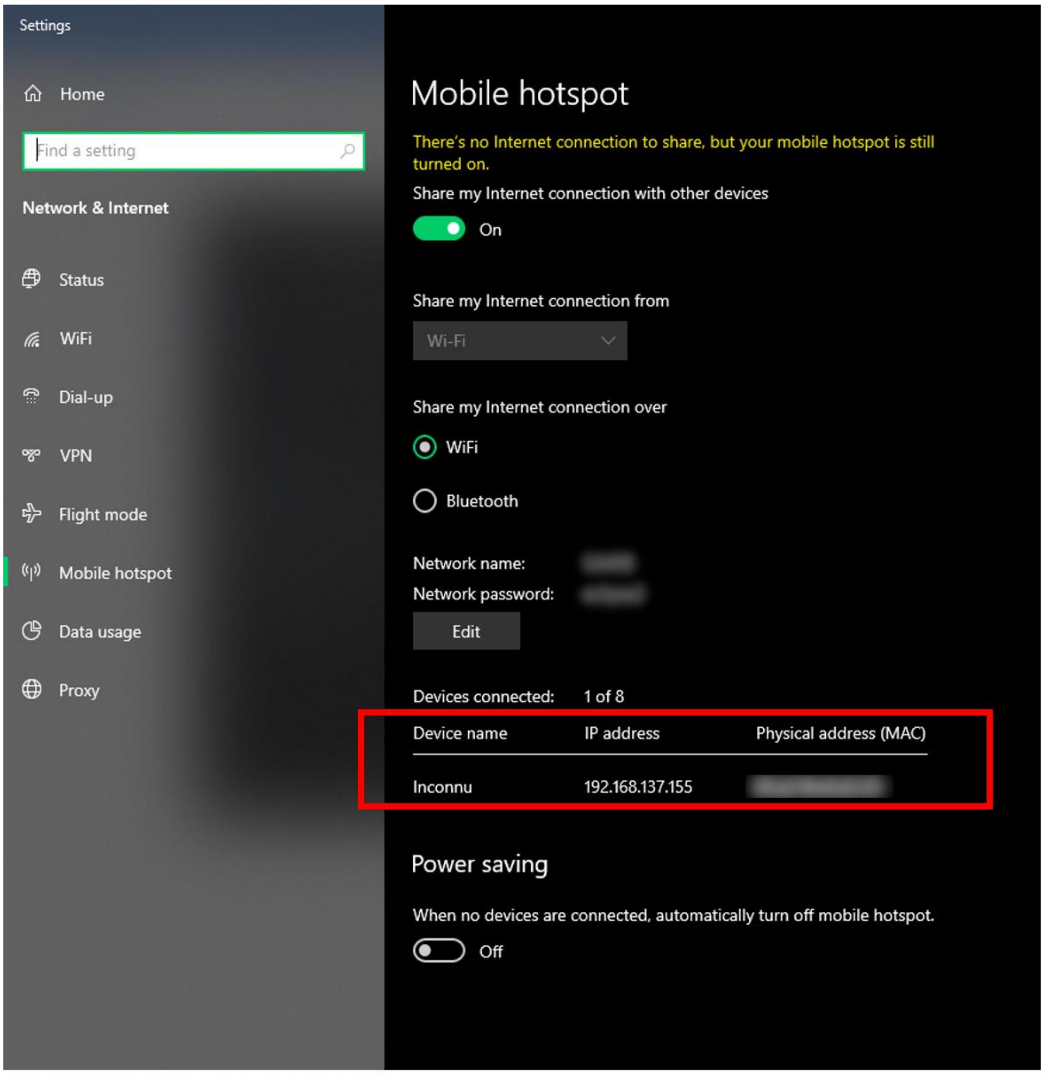


Figure 13. *Find your PICT IP address on a Windows 10 device*

Find your PICT IP address on an Android device’s Hotspot

Go to **Settings > Network and Internet > Hotspot & tethering > Wi-Fi hotspot**. Click on the **“Advanced”** button and on **“Connected users**”, it will show a list of devices connected to the hotspot with their IP addresses (Fig.14).

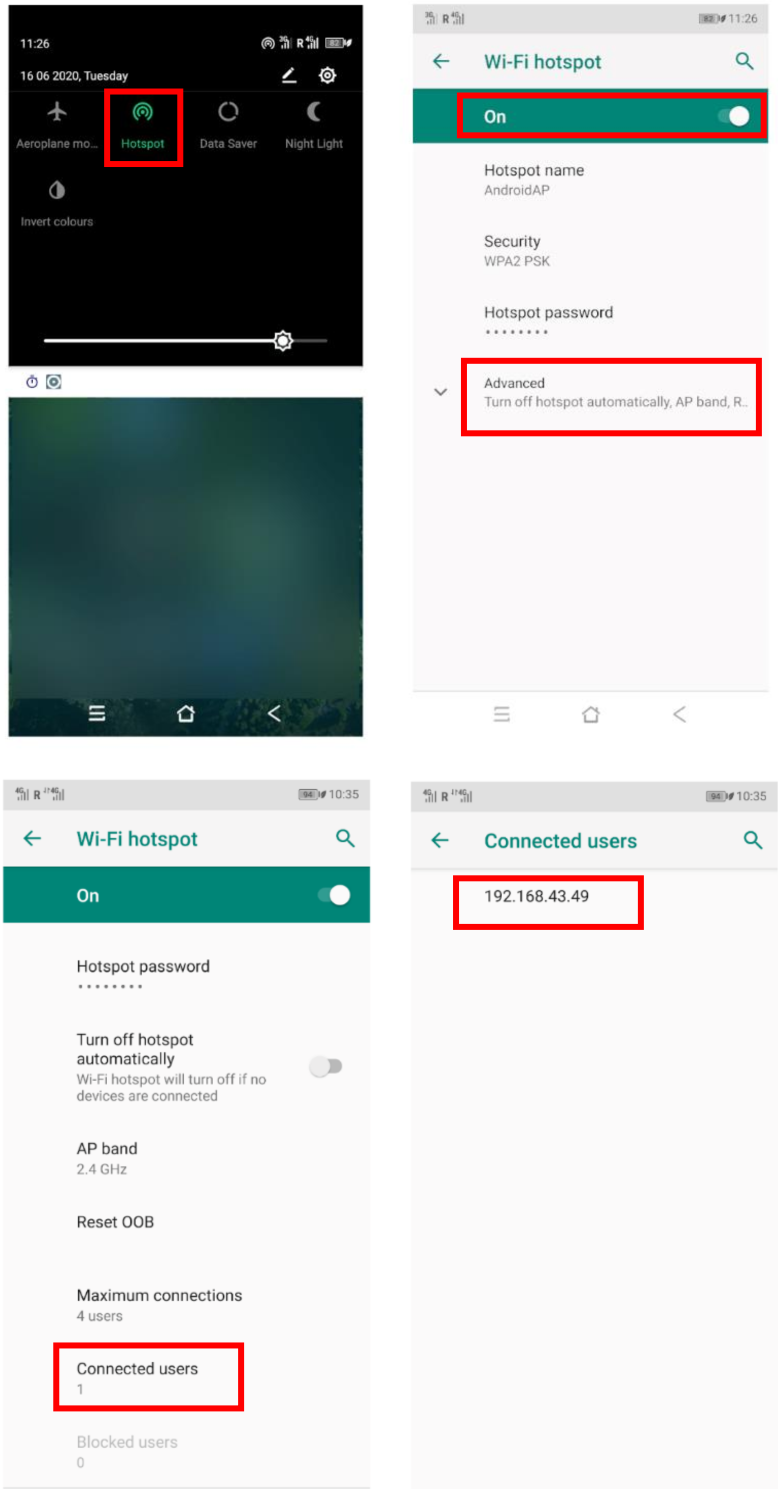


Figure 14. *Find the IP address of your PICT on an Android device*

In recent versions of android, you cannot see the IP of devices that are connected to your mobile hotspot using WiFi. In that case you have to use a third-party app to display details about connected devices. Try for instance **Ksiri Tools** (available at http://karthiksiriapps.blogspot.com/p/ksiri-android-tools.html).

Find your PICT IP address on any device

It is not always possible to find the IP address with the methods detailed above. For instance, iOS doesn't offer any way to find out exactly who is connected to your Personal Hotspot. An alternative solution is to use a software on a device that is connected to the same Wi-Fi network than the PICT to scan all the IP address of the network. We have tested successfully **Angry IP Scanner** on Windows and macOS, **Network Analyser** on iOS, **IP Scanner** on macOS or **Fing** on Windows, iOS and macOS.

As an illustration, we detail below the procedure with Fing. Fing has the advantage of being available on all platforms but has however some limitations. Fing can be used to scan the network that your device is connected to. But it won't scan the hotspot broadcasted by the device on which Fing is running.

- Install and launch **Fing** from the Play Store (Android) or App Store (Apple).
- Fing will propose to create an account. This is mandatory and you can't continue without an account.
- Click on **"Scan for devices"** (Fig.15.1).
- After a few seconds Fing will show the names, IP addresses, MAC addresses, and operating systems of all the devices connected to the Wi-Fi network. Look for the IP address of the "Raspberry Pi" device. Note that on iOS the app is unable to resolve the name of the Raspberry Pi. You will have to try every IP address with your SSH client (Termius/JuiceSSH). So we advise you not to use Fing on iOS if you use several PICTs at the same time.

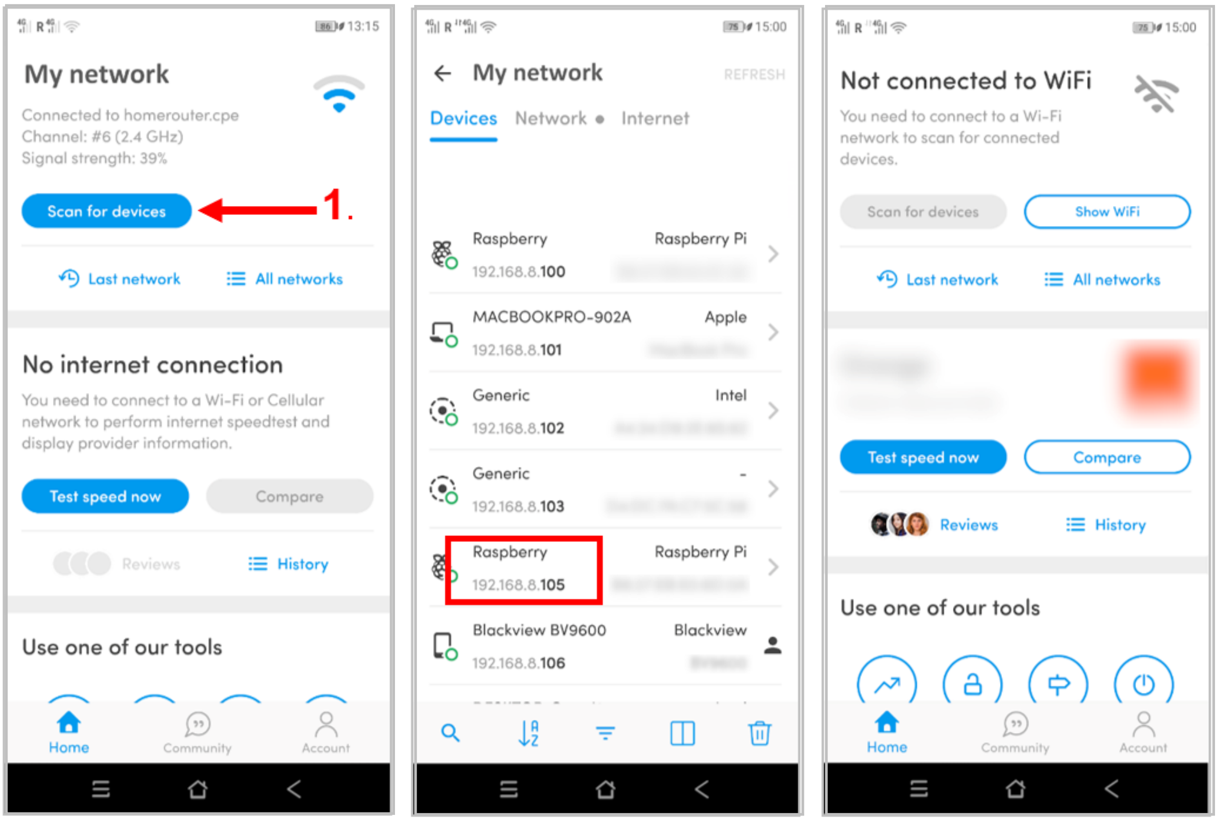


***Figure 15**. The Fing interface*

1.5. Access the terminal of your PICT

Raspberry Pi OS Lite does not include a graphical interface. It is operated through the terminal (or shell) (Fig. 16). The terminal is a text-based interface that accepts and interprets the text commands typed in by the user. You can use terminal commands in Raspberry Pi OS to run programs, execute scripts, manipulate files, etc. All the commands you need and their meaning are given below.

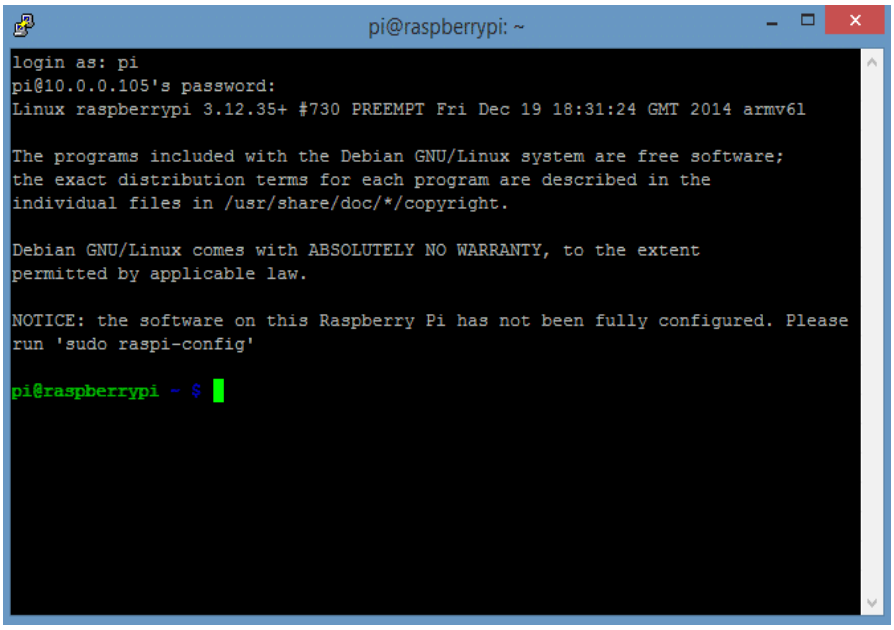
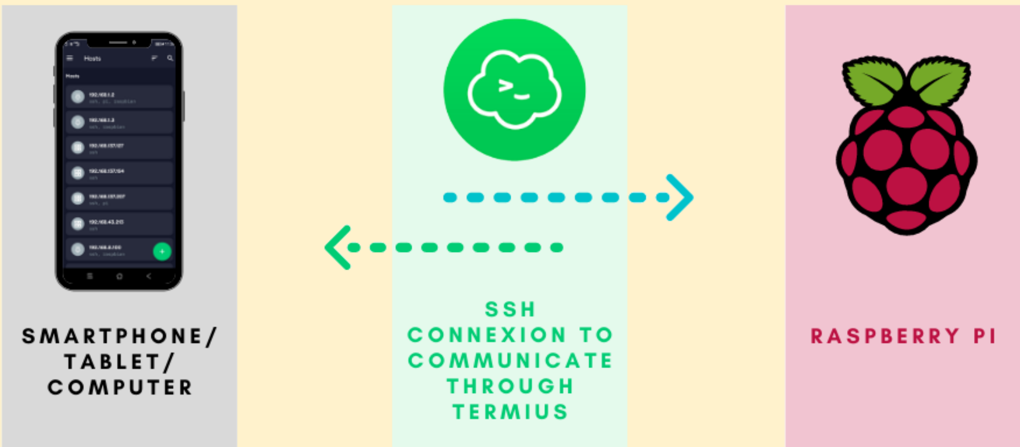


***Figure 16**. The terminal or command-line interface*

As your PICT has no screen or keyboard, you will use the screen of your computer or phone. To access the terminal from a separate device, you will establish a SSH connection using a dedicated application and the IP address given to your PICT by the Wi-Fi hotspot. We recommend **Termius as it is a cross-platform solution** (Windows, Android, iOS and macOS). Alternatively, you can also use **PuTTY** (Windows) or **JuiceSSH** (Android). If you plan to work in a remote area with no Internet connection, we recommend to install several of these software applications in case you have trouble with the one you usually use.

What is SSH?

SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users a secure way to access a computer over an unsecured network. SSH provides authentication and encrypted data communications between two devices connecting over an open network. Among other things, SSH is widely used for managing systems and applications remotely, enabling you to log in to another device over a network, execute commands and move files from one device to another.



The Termius interface differs slightly across platforms. Below we detail the procedure to access the PICT terminal using Termius on Android.

- Browse the Google Play Store for **Termius** app and install it. The interface is shown on Fig. 17.
- Click on the "+" button (Fig.17.1) and fill the **"Address"** line with the IP address of your PICT and click on **"Save"**. when asked, give the **login "pi"** and **password "raspberry".**
- Click on your PICT in the host list (Fig. 17.2).
- The terminal of your PICT and a keyboard will open after a few seconds (Fig.17.3).

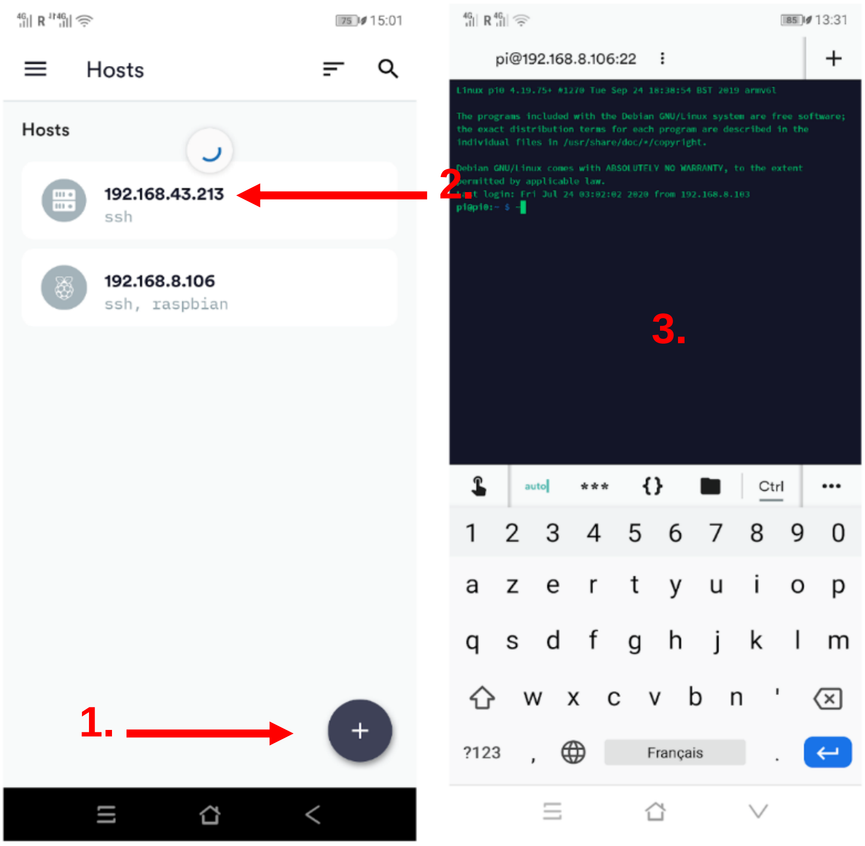


Figure 17. *The Termius interface on Android*

1.6. PICT general configuration

You now have a blank terminal opened in Termius (or similar applications), either on your computer or on your portable device. To execute a command, you type in the text followed by "enter" on your computer's keyboard or your device's virtual keyboard. Every line you enter is an individual command. The character "#" indicates that the subsequent text on the same line is a comment that will not be executed.

- Type in the following command line to start the Raspberry Pi Software Configuration Tool:

```
sudo raspi-config
```

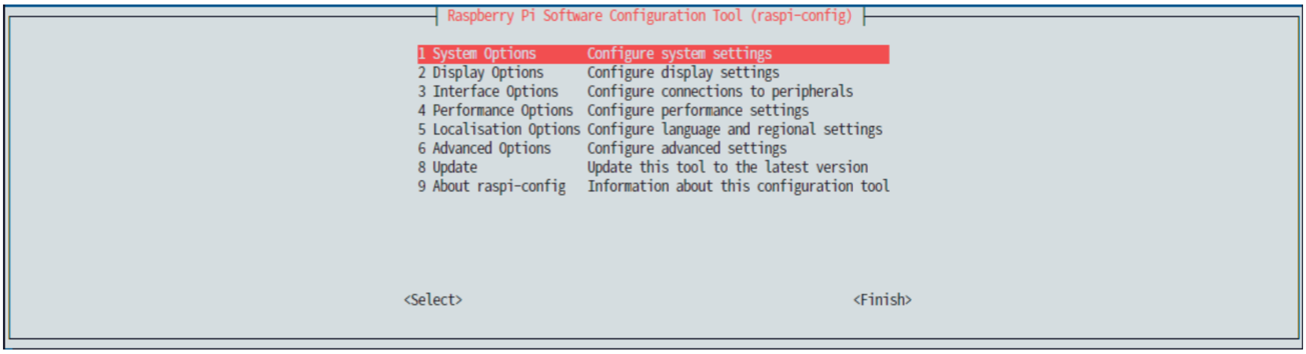


***Figure 18**. The Raspberry Pi Software Configuration Tool*

- The configuration tool menu will appear (Fig. 17).

- Go to **"1 System Options**" and then to "**S3 Password**" and change the password to "pi" (you will have to actualize it in Termius). This is not a mandatory step, but it will make your future connections easier as you will have to enter your PICT password several times.
- In the case you will use several PICTs on the same field site it is strongly recommended to give them different network names. These names will be visible in front of the IP address in your Wi-Fi device list and will be hard encoded in the video you will record. Go to **"1 System Options"** then to **"S4 Hostname"** and change the default name. For instance you can use "Pi01" or so.
- Go to **"3 Interface Options"** and
  - If you installed Bullseye or any later version of the OS (as suggested in this guide), go to **"I1 Legacy Camera"** and make sure that the legacy camera support is enabled.
  - If you installed Buster version of the OS (Legacy version), go to **"P1 Camera"** and make sure that the camera is activated.
- Go to **"6 Advanced Options"** and choose "**A1 Expand Filesystem**" to make sure that the entire micro SD card memory is available for writing.
- Go to <Finish> and accept to restart the Raspberry Pi Zero (W) when asked.
- Connect again to the Raspberry Pi using SSH.

Tip: You can restart the Raspberry Pi Zero (W) at any time with the command `sudo reboot` *and shut it down with* `sudo halt`

1.7. Update and install a Python package to manage the camera

What is Python?

Python is an interpreted programming language. A python interpreter was installed on your micro SD card with the Raspberry Pi OS.

Picamera is an open-source package that provides a pure Python interface to the Raspberry Pi camera module. We will use it to control the camera. To download and install Picamera, the Wi-Fi network to which you PICT is connected (see here) must have access to an internet connection. This will allow us to update the operating system and to download the package. The full package documentation can be found at https://picamera.readthedocs.io/

- Set up the actual date (important) with the following command:

```
sudo date 051512302022 # set the date to May 15th, 12:30, 2022
```

- Enter (or copy-paste) the following commands in the terminal:

```
sudo apt-get update # update the operating system
sudo apt-get install python3-picamera # download and install
the latest version of package Picamera
sudo apt-get dist-upgrade # update all the packages
sudo reboot # reboot the system
```

Tip: To copy-paste in Termius, use CTRL+ C and CTRL+V

What does # mean?

All the text typed in after the symbol "#" is considered a comment and will not be executed. It is common to add comments to explain the purpose of the command to other users.

```
sudo halt # turn off the Raspberry Pi
```

Example : "sudo halt" is the command executed by the Raspberry Pi, "turn off the Raspberry Pi" is the comment explaining the action done by this script. You can omit the comments when you copy-paste the command lines.

1.8. Reduce the power consumption of your PICT

One of the benefits of using a PICT is its large autonomy compared to other solutions. To achieve the best energy efficiency possible we will deactivate unnecessary components of the Raspberry Pi. This will help you to save around 0.13 W.

Deactivate the HDMI port

Since you are not using a screen you do not need the HDMI port.

- To open rc.local file with the nano editor, type in

```
sudo nano /etc/rc.local
```

- This will open the rc.local text file with the nano text editor. Add the following line just above “exit 0”.

```
/usr/bin/tvservice -o # disable HDMI (-p to re-enable)
```

Disable the activity LED

Contrarily to other Pi versions, the Pi Zero has only one LED: the activity LED. By default, this green LED is constantly activated when the Pi is running. You can save power and better conceal the PICT by disabling this LED.

- Add the following line above “exit 0”.

```
echo 0  | sudo tee /sys/class/leds/led0/brightness # disable ACT LED
```

Be sure to leave the line exit 0 at the end. Note that the activity LED will still blink on boot-up (2 times) and shutdown (10 times)

- Save your edits by pressing “CTRL+o” and return to the terminal using “CTRL+x”. Answer yes when asked and press “ENTER”.

Deactivate the Bluetooth

- Open config.txt with the command:

```
sudo nano /boot/config.txt
```

- Add the following line at the end of the text file that just opened:

```
dtoverlay=pi3-disable-bt # disable Bluetooth
```

You can save more power and make your PICT completely invisible by disabling the red blinking LED that is present on most Raspberry camera modules v. 1. To do so, add this line at the end of the same text file:

```
disable_camera_led=1 # disable camera LED
```

- Save your edits by pressing “CTRL+o” and return to the terminal using “CTRL+x”. Answer by yes when asked and press “ENTER”.
- Now reboot with

```
sudo reboot # reboot the system
```

Periodically disable the Wi-Fi module

You can save more power by turning off the Wi-Fi when it is not needed, for instance at night. The activation and deactivation time can be set up in Cron. This step is optional as the power saved is relatively small compared to the power used by other components like the LED light and processor (see the breakdown of components' power need in the MEE publication supplemented by this guide).

What is Cron?

Cron is a scheduling tool. You can use it to configure tasks or scripts that will run periodically at fixed intervals. The layout for a cron entry is made up of six components, respectively: minute (0 - 59), hour (0 - 23), day (1 - 31), month (1 - 12), day of the week (0 - 7), and the command to be executed. * stands for « any ».

To edit the Cron table type in:

```
crontab -e
```

The first time you run crontab you'll be prompted to select an editor; if you are not sure which one to use, choose nano by pressing Enter.

- Enter the following lines at the bottom of the page:

```
0 19 * * * sudo ifconfig wlan0 down # switch off Wi-Fi at dusk (7 PM)
0 7 * * * sudo ifconfig wlan0 up # switch on Wi-Fi at dawn (7 AM)
```

- Type CTRL + x to exit the Cron table. Press « y » followed by « Enter » to save your edits. You can see your scheduled tasks with:

```
crontab -l
```

Tip: Cron can be used to define hours at which the recording will take place:

```
0 19 * * * nohup sudo python camera.py & # start video recording script at dusk (at 7 PM)
0 7 * * * sudo killall python # stop video recording at dawn by shutting down every python processes(at 7 AM)
```

1.9. Setting up the camera

a. Install Rpi Cam-Web Interface to preview the camera stream

To set up the framing, focal distance, and time of your PICT you will use **Rpi Cam-Web Interface**. Note that the RaspiCAM Remote app is an alternate solution for Android devices. But we recommend the use of Rpi Cam-Web Interface as it is more responsive.

- First make sure that a camera is firmly connected to the camera slot of your Pi. On both end of the ribbon cable, the golden/silver connectors should face down (toward the Pi board), the black part up (towards you) (Fig. 2).
- To install the **Rpi Cam-Web Interface**, enter the following commands one by one in the terminal of your PICT:

```
sudo apt-get install git # install git
git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git # clone the code from github
sudo apt-get update
RPi_Cam_Web_Interface/./install.sh # run the install script
```

- The install wizard will show up after the last command (Fig. 19).

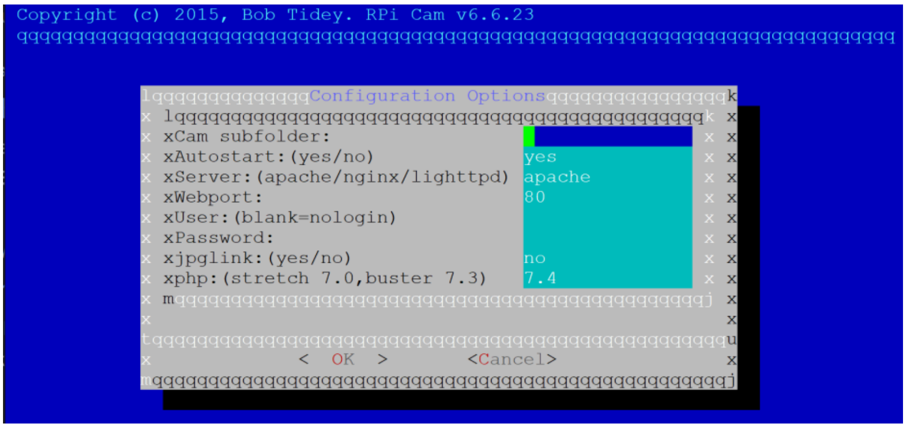


Figure 19. *The Rpi-Cam-Web-Interface wizard*

- Leave the Cam subfolder blank (delete « html »).
- Set autostart to yes.
- Leave all other options to their default setting and select OK (press on the tab key to reach OK or Cancel).
- After a several minutes of download and package install, **RPi-Cam-Web-Interface** will ask you if you want to start it now. Select « Yes ».

Open up a web browser (we recommend Google Chrome as it seems more responsive in this task but other browsers will work as well) on a device that is connected to the same Wi-Fi network and enter in the address field the IP address of the PICT followed by "ENTER".

An HTML page will open up with a preview of the video stream from your PICT (Fig. 20).

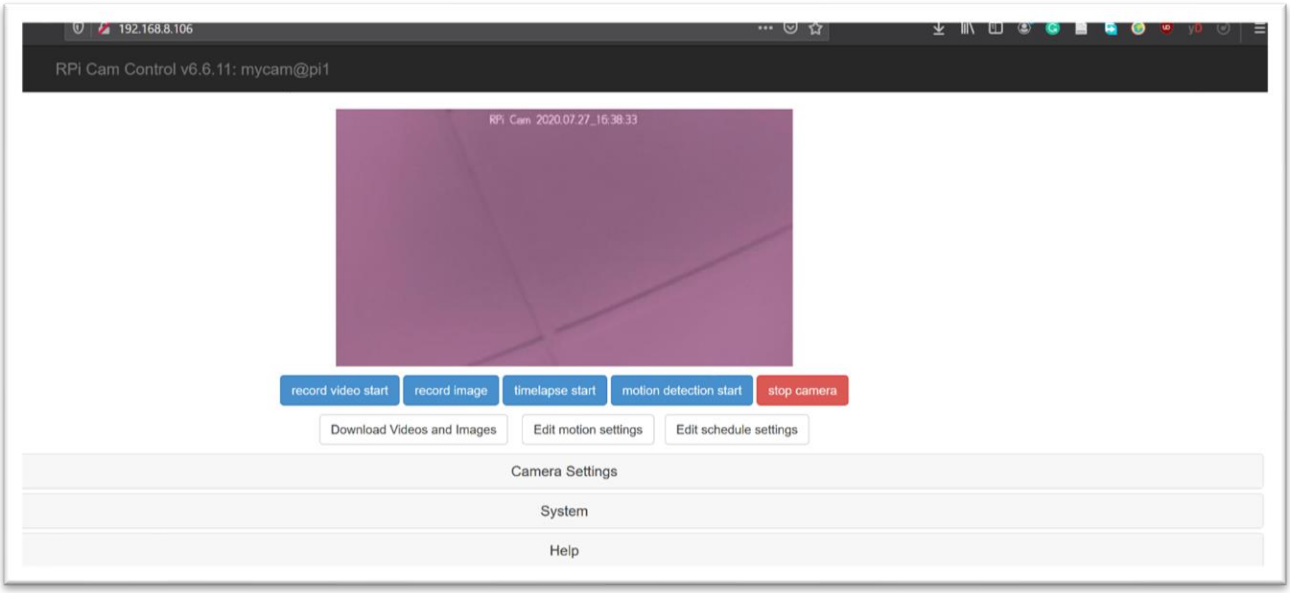


Figure 20. *The Rpi-Cam-Web-Interface*

- Click on the video to enlarge.

To facilitate the tuning of the focal distance, we recommend changing the following options in the camera settings at the bottom of the page:

- Set Resolutions/load preset to Max view 972p 4:3.
- Set Preview quality (1...100) to 15.
- Set Width (128...1024), to 1024.
- Click on Ok each time to save changes permanently.

From now on you can start the RPi-Cam-Web-Interface by typing in:

```
RPi_Cam_Web_Interface/start.sh
```

And stop it with:

```
RPi_Cam_Web_Interface/stop.sh
```

b. Create the python script that will control your camera module

First, create a folder in which your recordings will be saved. Use the following command line:

```
mkdir record # it will create a folder named “record” in the
main directory (i.e. home/pi/)
```

- Create and open an empty text file in /home/pi/ with the python extension (*.py):

```
sudo nano camera.py # it will create a text file called
“camera.py”, which will automatically open
```

- Copy-paste the script that corresponds to your needs (see below). Be careful to copy the indentation of lines (multiple space character) as well. Incorrect indentation will prevent the script from running.
- Save your edits by pressing “CTRL+o” and return to the terminal using “CTRL+x”. Answer yes when asked and press “ENTER”.
- To edit option(s) in a python script you’ve already created, use the same command line (`sudo nano camera.py`), modify the script, and save it as in the previous bullet point.

c. *Example script to continuously record video*

The following script will start the recording of **1000 videos lasting 1 hour each.** It is available as a text file attached to this guide to facilitate copy/pasting of indentation. We recommend using this script to monitor insect activity. The camera sensor’s native resolution is 2592x1944 pixels for the Raspberry Pi V1 camera, but we recommend 1296x972, which is one of the native outputs of the camera. It is obtained by 2 by 2 pixel binning which reduces noise in low light conditions. We found that higher resolutions exceed the capability of the lens we used to resolve small objects. Note that 1664 x 1248 is the largest resolution at 4:3 ratio for the h.264 encoder of picamera which implement level 4.1. The script below is available as a text file attached to this guide to facilitate copy/pasting of indentation.

```
import picamera
import socket
import uuid
from datetime import datetime as dt

qual=22 # level of image quality between 1 (highest quality, largest size)
and 40 (lowest quality, smallest size), with typical values 20 to 25,
default is 0.
video_duration = 3600 # video duration in seconds
video_number = 1000 # number of video sequences to shoot
UID = uuid.uuid4().hex[:4].upper()+'_'+dt.now().strftime('%Y-%m-%d_%H-%M')
# generate random unique ID that will be used in video filename
HostName=socket.gethostname()

with picamera.PiCamera() as camera:
    camera.resolution = (1296, 972) # max is 1664, 1248
    camera.framerate = 15 # recommended are 12, 15, 24, 30. Max fps is 42
at 1296x972
    camera.annotate_frame_num = True
    camera.annotate_text_size = int(round(camera.resolution[0]/64))
    camera.annotate_background = picamera.Color('black') # text background
colour
    camera.annotate_foreground = picamera.Color('white') # text colour

    for filename in camera.record_sequence([
        '/home/pi/record/'+HostName+'_'+UID+'_%03d.h264' % (h + 1)
        for h in range(video_number)
        ], quality=qual):

        start = dt.now() # get the current date and time
        while (dt.now() - start).seconds < video_duration: # run until
video_duration is reached
            camera.annotate_text = HostName+', '+str(camera.framerate)+'
fps, Q='+str(qual)+', '+dt.now().strftime('%Y-%m-%d %H:%M:%S') # tag the
video with a custom text
            camera.wait_recording(0.2) # pause the script for a short
interval to save power
```

d. Example script to record images at regular interval

If instead of recording videos you would like to take pictures at regular intervals (typically less than 1 frame per second) you can use the following **photography script**. The script below is available as a text file attached to this guide to facilitate copy/pasting of indentation. The maximal resolution of the sensor is 2592 x 1944 but we recommend 1296 x 972 (see the paragraph above).

```
from time import sleep
import picamera
import datetime as dt
import socket
import uuid

interval_duration = 10 # time lapse between two successive pictures in
seconds
UID = uuid.uuid4().hex[:4].upper()+'_'+dt.datetime.now().strftime('%Y-
%m-%d_%H-%M') # generate random unique ID that will be used in pictures
filenames
HostName=socket.gethostname()

with picamera.PiCamera() as camera:
  camera.resolution = (1296, 972) # set picture resolution
  camera.annotate_text_size = int(round(camera.resolution[0]/64))
  camera.annotate_text = HostName+', '+dt.datetime.now().strftime('%Y-
%m-%d %H:%M:%S') # tag the picture with a custom text
  camera.annotate_background = picamera.Color('black') # text background
colour
  camera.annotate_foreground = picamera.Color('white') # text colour

  for filename in camera.capture_continuous('/home/pi/record/'+ UID +
'_' +'{counter:05d}.jpg'): # counter:05d = file name with 5-digit
incrementor that starts at 1 and increases by 1 for each image taken
     sleep(interval_duration) # wait for next picture
     camera.annotate_text = HostName+',
'+dt.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
```

Tip: If you want to build several PICTs, you can use Win32 Disk Imager *to save a copy of your operating system and its final configuration and then write it on other SD cards. This procedure is also highly recommended to recover corrupted micro-SD card.*

e. Example script to record video only when a motion is detected

We advocate the application of motion detection algorithms as a postprocessing stage rather than doing this in situ because the processing of the video stream to filter out still sequences is computationally expensive. The increased power consumption will drastically reduce the autonomy of PIC. For instance, the motion detection algorithms implemented in RPi Cam Web Interface (https://github.com/silvanmelchior/RPi_Cam_Web_Interface) and MotionEyeOS (https://github.com/ccrisan/motioneye) would increase power draw over 24h (with 12h of daylight) by ca. 67% and 43% respectively.

You will find a great tutorial here to apply on the fly motion detection: https://github.com/roblanf/raspberrytrap

V. V. PICT use in the field

The procedure to operate your PICT in the field might seems complex at first, but after some practice, you will realize that it is pretty simple. We recommend first testing every step of the procedure in the office or at home.

You can remotely control your PICT from any Windows, Mac, iPhone or Android device. Different software is going to be used at each step, but the logic of the process is the same for all device types. Below we present the recommended general approach based on a cross-platforms software application. Alternatives applications are also given in Figure 21. Whatever OS and SSH client you are using, the commands to use remain the same.

Tip : We advise you to use a smartphone or tablet rather than a computer for outdoor recordings as they have better autonomy, can be protected from water and are easier to transport and handle in difficult field conditions. Remember that the first configuration of your PICT requires internet access to download/update the apps used to control the camera.

1. Quick Start

1) **Start the Wi-Fi network** on your portable device (see here), and power on your PICT by plugging in the USB cable to the power bank.

2) Find the IP address of your PICT (see here).

3) Select the subject to record/survey and firmly attach the PICT on a solid support that is able to withstand its weight (0.4 to 1kg, depending on battery capacity). Be mindful of the sun's trajectory across the sky to avoid backlight. You also need to take into account possible motion of the subject and the support in case of wind or rainfall.

4) Open your favourite web browser on your portable device (whether Chrome, Firefox, Safari or any other), Type in your PICT IP address in the address bar and press enter. You will see the **Rpi-Cam-Web-Interface showing a live stream** of the output of the camera module (see here how to install and use the Rpi-Cam-Web-Interface). Use this screen as a viewfinder to adjust the framing (by moving your PICT) and focus distance (by manually rotating the lens clockwise to focus more distant objects and counter-clockwise for closer objects).

5) When the framing and focusing are perfectly adjusted **stop the Rpi-Cam-Web-Interface**. This is required to let the python script access the camera module. To stop the Rpi-Cam-Web-Interface, click on the red button "stop camera" in the Rpi-Cam-Web-Interface page showing the live stream or type in the following command in a Termius terminal (see here how to use and install it):

```
RPi_Cam_Web_Interface/stop.sh
```

6) Set up the time and date of your PICT. Note that this can also be done in the **Rpi-Cam-Web-Interface.**

```
sudo date 051512302022 # set the date to May 15th, 12:30, 2022
```

7) Start the python script that will launch the recordings.

```
nohup sudo python camera.py &
```

8) When the observation period is finished you can stop the script:

```
sudo killall python # kill all python processes
```

Alternatively, you can shut down the PICT and the script at the same time:

```
sudo halt
```

Note that the micro SD card can get corrupted when the PICT shut down automatically due to power failure. In this case you need to re-write the SD card with a fresh image of your operating system and configuration to prevent possible failure.

2. Useful commands

1) To stop the python script, and hence the recording:

```
sudo killall python # kill all python processes
```

2) To shut down the PICT (and stop the recording):

```
sudo halt
```

3) To reboot the PICT (and stop the recording):

```
sudo reboot
```

To start the Rpi-Cam-Web-Interface. In case you inadvertently stopped it.

```
RPi_Cam_Web_Interface/start.sh
```

4) To check if the recording started properly, you can list the file(s) encoded in the 'record' folder you created previously (see here) using the command:

```
ls record -l # list all files in the 'record' folder. The '-l' flag allows the display of additional information such as file size.
```

By using this command repeatedly, you can see the currently recorded video file getting progressively bigger. The file size is indicated just after "`-rw-r—r—l root root`" (see figure 21). The above command can also be used to verify the volume of data that has been recorded so far.

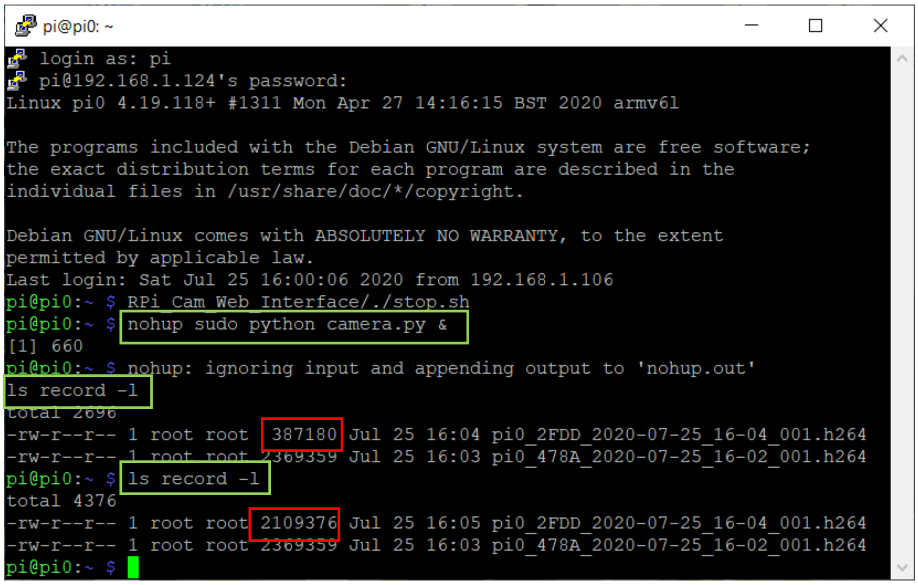


Figure 21. Monitoring your recordings

5) To clear all the files stored in the 'record' folder before starting a new recording session you can use the command:

```
rm record/* -f # delete all files in the 'record' folder. The '-f' flag prevents the system from asking for confirmation (y/n) for the deletion of each file.
```

6) To check disk space:

```
df -h # The ‘-h’ flag stands for human readable format.
```

7) To add, remove or edit Wi-Fi networks that PICT can connect to you have to change the content of wpa_supplicant.conf file:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

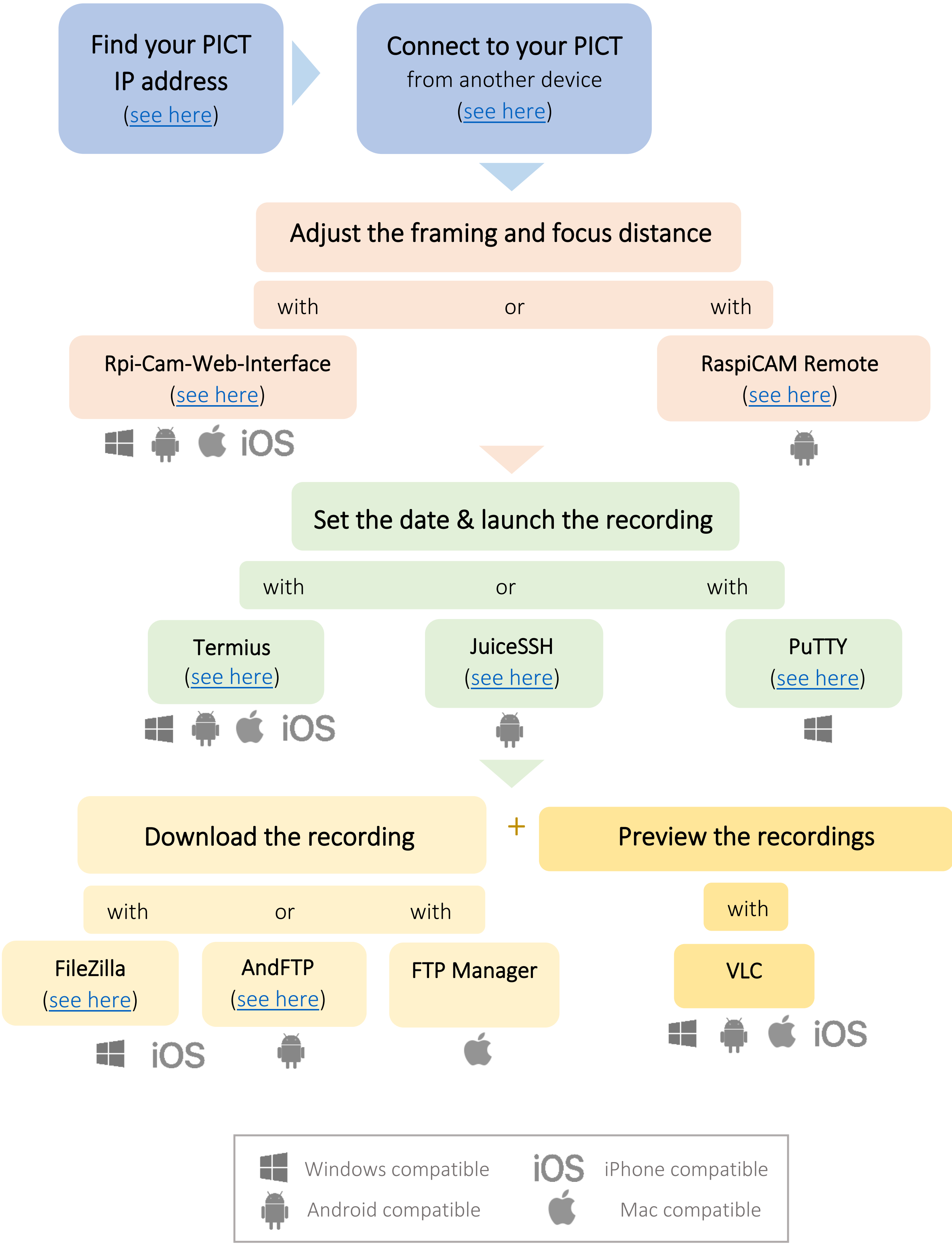


Figure 22. Steps to start a recording with your PICT and software/OS compatibility

VI. Processing and analysis of the recording

In this chapter we will see how to copy the recordings from the SD card to a computer and analyse them with a motion-detection algorithm. The general framework is shown on Fig. 23.

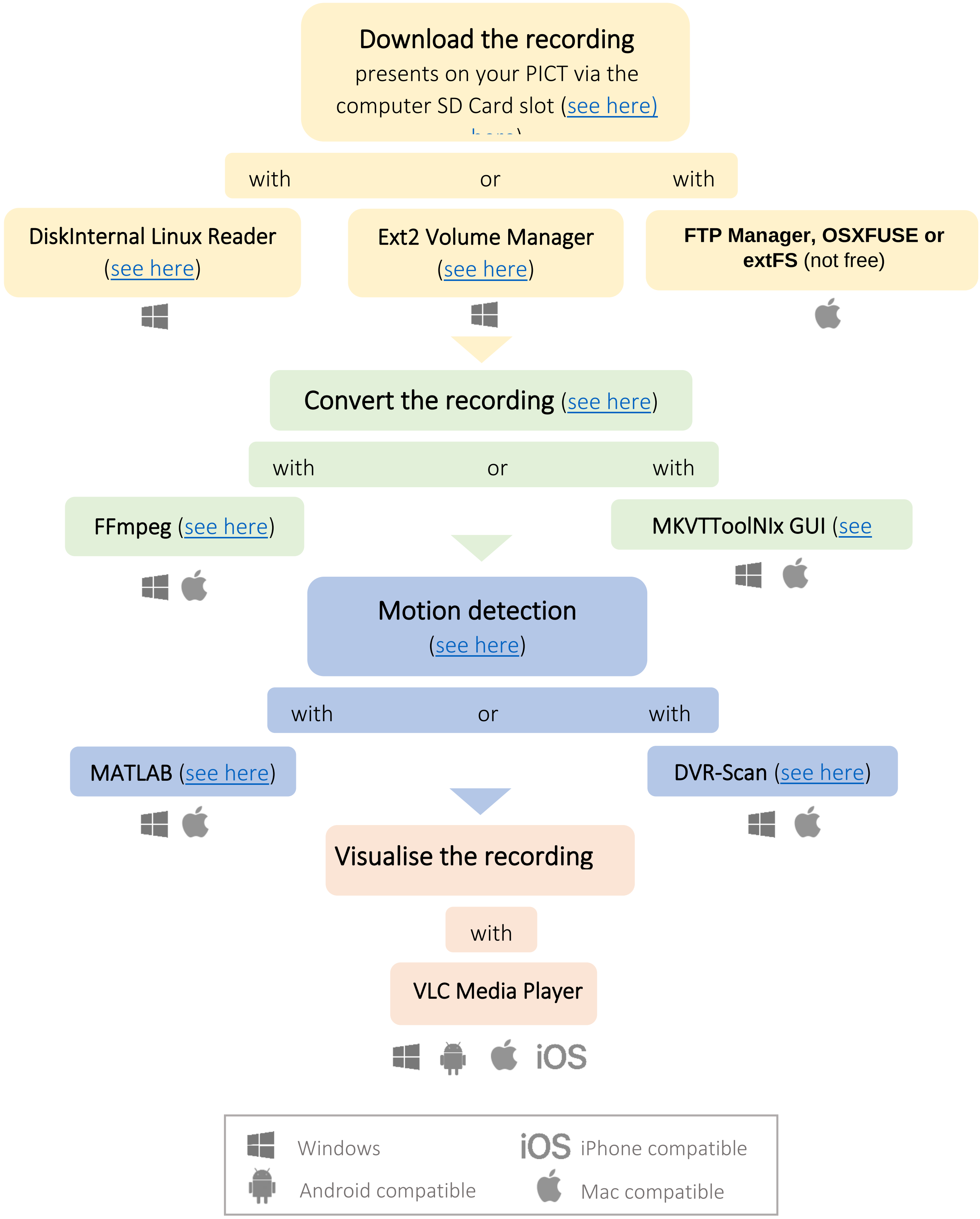


Figure 23. Steps to download and analyse your recording and software/OS compatibility

2. Copy your recordings from the micro SD card to your computer

You can download your recordings through Wi-Fi onto your portable device. This is done using a FTP client like **FileZilla** (Windows and iOS), **AndFTP** (Android) or **FTP Manager** (macOS) installed on the device where the files have to be sent. This method provides the convenience of remotely watching the photos/videos on a smartphone even when the camera is still recording. Download and install one of the FTP clients. Set up the host address as your PICT IP (see here) and the protocol as sFTP. The video files downloaded on your device can then be watched with the cross-platform VLC player. Note that because the data will be transferred through the Wi-Fi network the maximum rate of data transfer will be about 2 MB per second.

To efficiently copy all of your recordings you need a higher transfer rate. Reading files using the micro SD card reader of a computer can be done typically between 30 and 80 MB per second, depending on your hardware.

When you installed Raspberry Pi OS on the micro SD card you created two partitions. The smallest one is the boot partition and is formatted in FAT32. This is readable by any operating system including Windows and macOS. The largest one is the filesystem (data) partition that contains the Raspberry Pi OS and your recording. It is formatted in Ext3 Linux partition and is not readable by standard installation of Windows or macOS. In order to read the recordings on the Linux partition using a Windows device you will have to install a dedicated software.

- Install the freeware **DiskInternals Linux Reader™** available at https://www.diskinternals.com/linux-reader/
- Launch the DiskInternals Linux Reader to see and copy the files (Fig. 24).

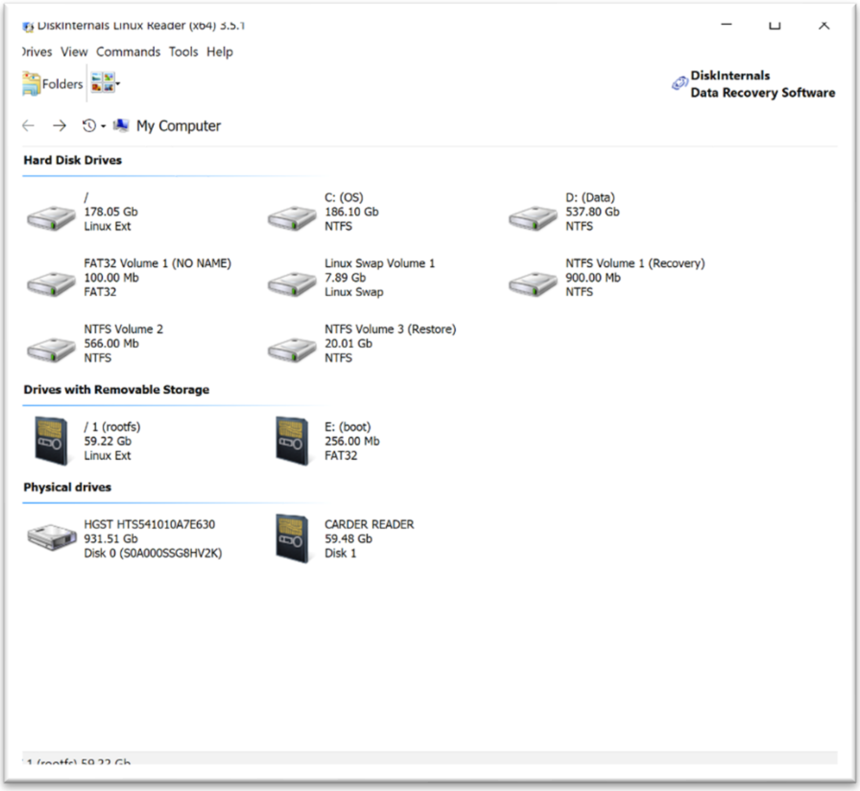


Figure 24. *The DiskInternals Linux Reader Interface*

An alternative free open source solution is Ext2Fsd available at http://www.ext2fsd.com. Once the software is running, the data stored on the SD card will appear in the Windows file explorer as for any compatible external storage drive. On the first start, the software presents itself like this shown on Fig. 25.

To avoid the need to start the software each time you need to read a Linux formated SD card, go to **File>Enable Ext2 Volume Manager autostart,** close the software and restart your computer.

If you have a computer running on macOS, you can use **OSXFUSE or extFS** (by Paragon Software, not free) to read the Linux formatted SD card.

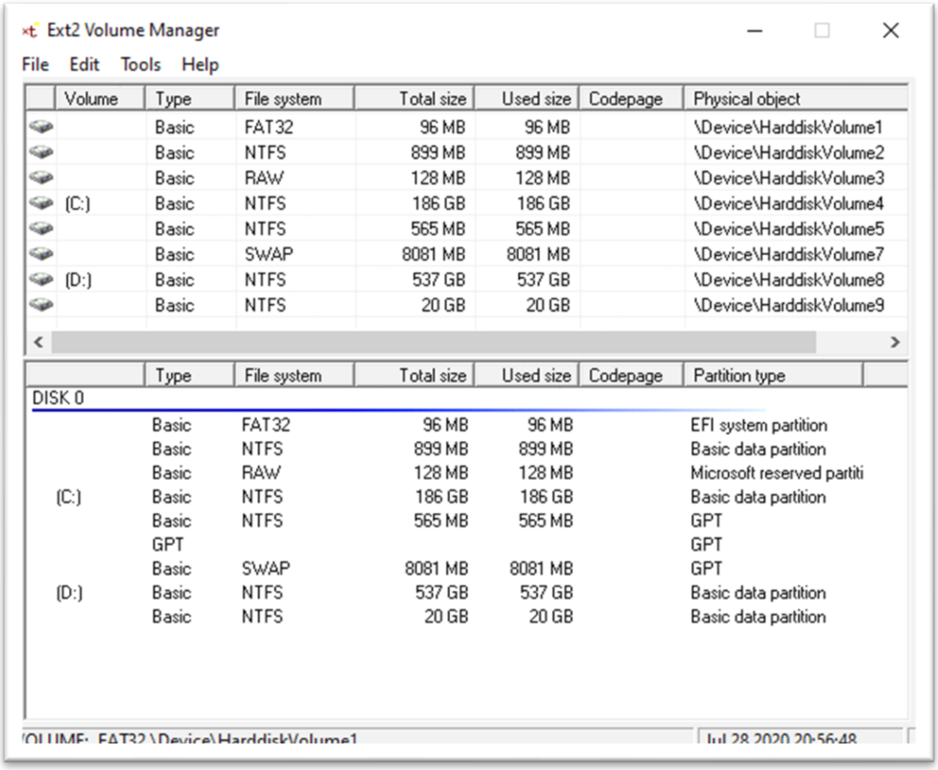


Figure 25. *The Ext2 Volume Manager interface*

3. Encapsulate the H.264 video in an mp4 container

VLC media player is able to read the raw .h264 format videos produced by the PICT. However, at this stage, there is no time stamp or FPS value in the metadata of the recordings. Thus VLC will read your video as it was recorded at 24 FPS (which might be incorrect) and most other media players will be unable to read the files. You first need to convert your recordings (i.e. encapsulate) into a common file format like .mp4 (container). This operation does not alter or modify the data in any way.

We recommend two different cross-platform (Windows, macOS and Linux) software applications to encapsulate your recordings. The first, **FFmpeg** is an open source solution supported by a wide community. It takes more time to install but achieves better speed when converting the videos (~5 seconds/1 hour recording, on an Intel® Core i7 2.00GHz, HDD). The second, **MKVmerge**, is slightly slower (~13 seconds/1 hour recording). Note that encapsulation time mostly depends on the speed of your hard drive. The use of an SSD drive will greatly reduce the processing time. Below we describe the procedure to install both applications on a Windows computer. On macOS the procedure is slightly different and we encourage Mac users to look for tutorials on the applications' respective websites.

3.1. Using FFmpeg

- Download and unzip **FFmpeg** (https://ffmpeg.org).
- Rename the uncompressed folder to « ffmpeg ».
- Place the folder on your OS main Disk (C:) so its path is C:\ffmpeg.
- Type in "Environment Variables" in Windows search field, next to the Start menu, select "Edit the system environment variables".
- In the windows that will open, click on "Environment Variables", a new window will open, select the line "Path" and click on "Edit". It will open another window, click on "New" and add the path of your ffmpeg executable. If you placed the FFmpeg files into C:\ffmpeg, your path should be "C:\ffmpeg\bin". Then click on "OK".
- **FFmpeg** is now properly installed !

With **FFmpeg** you can process your video one by one, but it would be incredibly time consuming. Therefore we recommend automating the process by creating a batch file.

- Create a new text file using **Notepad** in the folder containing your recording. Copy-paste the following lines in the new file.

```
mkdir newfiles
for %%a in ("*.h264") do "C:\ffmpeg\bin\ffmpeg" -
framerate 15 -i "%%a" -c:v copy "newfiles\%%~na.mp4"
pause
```

- Make sure that the installation path ("C:\ffmpeg\bin\ffmpeg" in our example) matches the actual installation path.
- The encapsuler will add a frame rate to the metadata of the new file. In the example above we specify a frame rate of 15 FPS. Change this value to match the actual frame rate of your recordings (as set up in the camera.py file and hard encoded in the video). If the frame rate is incorrect your movie player software will display the movie at the wrong speed and the duration of the movie will be erroneous.
- Click on the **"file"** tab and click **"Save as"**. In the **"Save as type"** drop-down menu, select **"All Files (*.*)"**. In the File name field type in a name, for instance "FFmpegBatch.bat" without quotation marks. Click on the "Save" button.
- Double click on the batch.bat file to start the conversion.
- The batch file will automatically create a folder named "newfiles" where it will place the converted videos as they are created.

What is a batch file ?

A batch file consists of a script stored in a plain text file. It is used to automatize an action, such as repeatedly convert files, as we have done above, without any action from the user after launching the process.

3.2. Using MKVmerge

- Download and install the free open source software **MKVToolNIx GUI** at https://mkvtoolnix.download.
- Create a new text file using **Notepad** in the folder containing your recording. Copy-paste the following lines in the new file. Note that the following example assumes that the video was recorded at 15 FPS. You have to change that value according to your data in order to get a correct video length.

```
for %%a in ("*.h264") do "C:\Program Files\MKVToolNix\mkvmerge" -o
"newfiles\%%~na.mp4" --default-duration 0:15fps "%%a"
pause
```

  - Make sure that the installation path ("C:\Program Files\MKVToolNix\mkvmerge" in our example) matches the actual installation path.
  - The encapsuler will add a frame rate to the metadata of the new file. In the example above we specify a frame rate of 15 FPS. Change this value to match the actual frame rate of your recordings as set up in the camera.py file and hard encoded in the video). If the frame rate is incorrect your movie player software will display the movie at the wrong speed and the duration of the movie will be erroneous.
  - Click on the **"file"** tab and click **"Save as"**. In the **"Save as type"** drop-down menu, select **"All Files (*.*)"**. In the File name field type in a name, for instance "MKVmergeBatch.bat" without quotation marks. Click on the "Save" button.
  - Double click on the batch.bat file to start the conversion.
  - The batch file will automatically create a folder named "newfiles" where it will place the converted videos as they are created.

Now that you have converted all the movies into mp4 files you can watch them on your favourite media player. With **VLC media player** (https://www.videolan.org) you can watch the movies at an accelerated speed to search for specific events related to your study. Alternatively, you can automatically detect the scenes where there is motion using specific algorithms.

3. Apply motion detection to your sequences

In this section we present two solutions for efficient post-processing of files via a motion detection algorithm. Motion detection can save time when reviewing a long recording to look for rare or brief events. However, automatic motion detection comes with a trade-off between false positive and false negative detection rates. The former is time consuming to review and the latter may lead one to underestimate the frequency of certain events. It is therefore important to adjust the detection threshold carefully to achieve your objectives. There are many automated motion detection methods available and the field of computer vision is advancing rapidly. If one would like to pursue a motion detection approach, we encourage reviewing the most recent literature with a search keyword combination such as “automated motion detection”.

3.3. Using DVR-Scan freeware

- Download and install **DVR-Scan** available at https://github.com/Breakthrough/DVR-Scan.
- Create a new text file using **Notepad** in the folder containing your recording. Copy-paste the following lines in the new file.

```
mkdir Motion
for %%a in ("*.mp4") do dvr-scan -q --time-before-event 1s --time-post-event 1s --threshold 1 -i "%%a" -o "Motion\%%~na_motion_only.avi"
pause
```

- Click on the **“file”** tab and click **“Save as”**. In the **“Save as type”** drop-down menu, select **“All Files (*.*)”**. In the File name field type in a name, for instance “DVRscanBatch.bat” without quotation marks. Click on the **“Save”** button.
- Place the file in the folder with.mp4 files. Double click on the file to execute it.
- It will create a folder named "Motion" and copy inside all your videos but with all the still sequences removed. You can fine-tune the threshold a frame's motion score must meet to trigger a motion event. The default threshold is 0.15, lower values will be more sensitive to changes, whereas higher values will be less sensitive. We found the value of 1 best fit our purpose but it might be different for your specific study. If the output contains scenes without any motion, or if background movement causes false events to be detected, try to adjust the threshold value.

3.4. Using MATLAB software

MATLAB isn't a free open source software, so this solution is only recommended for researchers who have access to a license. The advantage of the simple MATLAB motion detection function given below is the ability to choose a subset area to focus on a region of interest in the video frame. The function below is available as a text file attached to this guide to facilitate copy/pasting.

- Save the following function in your computer and make sure the containing folder location is listed in **MATLAB** search path:

```
function GetMotion(PathVideo)
% Filter out all still sequences from a set of video files and create a
% summary video including only the detected motion sequences.
%
% PathVideo The path to the folder which contain the mp4 video files to
% process.
%
% Citation:
% Droissart V., Azandi L., Onguene E. R., Savignac M.,  Smith T.B.,
Deblauwe
% V. 2021. A low cost, modular, open-source camera trap system to study
% plant-insect interactions. Methods in Ecology and Evolution
%

VideoList = dir(fullfile(PathVideo,'\*.mp4')); % list all mp4 files in
PathVideo
Tresh=3; % rms treshold above which a frame is detected as motion (6 with
red band)
Buffer=1; % number of seconds to keep before and after detection.

% Get area to inspect
ThisVideoID=min(10,size(VideoList,1));
ThisVideo = VideoReader(fullfile(VideoList(ThisVideoID).name));
FrameRate=ThisVideo.FrameRate;
disp(['Video Framerate is ',num2str(FrameRate),' frames/s']);
tmp = readFrame(ThisVideo);
h=figure('position',1.0e+03 *[0.0314 0.1058 1.2360 0.6560]);
imagesc(tmp),axis image,title('Please, left-click on two corner of the
area of analysis')
[x,y] = ginput(2); % require user input
x=round(sort(x));y=round(sort(y));
close(h);
AOI=false(ThisVideo.Height,ThisVideo.Width); % Area of analysis (user
defined)
AOI(y(1):y(2),x(1):x(2))=true;
AOIn=sum(AOI(:)); % number of elements in AOI

% Initialize the final video file
if exist(fullfile(PathVideo,'SummaryVideo.mp4'), 'file') == 2 % if summary
video file already exists
    delete(fullfile(PathVideo,'SummaryVideo.mp4')) % remove the file
end
MergedVideo = VideoWriter(fullfile(PathVideo,'SummaryVideo'),'MPEG-4'); %
mp4 is both the fastest to write and the highest compression rate
MergedVideo.FrameRate=FrameRate;
open(MergedVideo);
FrameID3=0; % number of frames recorded in MergedVideo

for ThisVideoID=1:size(VideoList,1) % for each video
```

```
  disp([datestr(now),' Loading ',VideoList(ThisVideoID).name,'
',num2str(ThisVideoID),'/',num2str(size(VideoList,1)),' Frame
',num2str(FrameID3)])

  % Read the first frame
  FrameID=1; % this is the frame ID of tmp
  ThisVideo = VideoReader(fullfile(VideoList(ThisVideoID).name));
  BufferFrames=round(Buffer*ThisVideo.FrameRate); % number of frames to
take before and after detection
  tmp = readFrame(ThisVideo);
  tmp = insertText(tmp,[0 0],[VideoList(ThisVideoID).name,'
',datestr(datetime(0,0,0,0,0,FrameID/ThisVideo.FrameRate),'HH:MM:SS')],'Te
xtColor','white','FontSize',15,'BoxOpacity',0.2);

  % Initialize variables
  FrameID2=BufferFrames*2; % frame ID since last detected motion

BufferVideo=zeros([ThisVideo.Height,ThisVideo.Width,3,BufferFrames],'uint8
'); % buffer: the last BufferFrames frames

  % Read the video
  while hasFrame(ThisVideo)
    % write the frame in the buffer
    id=mod(FrameID-1,BufferFrames)+1; % linear index of the frame tmp
    BufferVideo(:,:,:,id)=tmp;
    BufferVideoInd=[id:-1:1,BufferFrames:-1:id+1];
    % read the new frame
    tmp2 = readFrame(ThisVideo);
    tmp2 = insertText(tmp2,[0 0],[VideoList(ThisVideoID).name,'
',datestr(datetime(0,0,0,0,0,FrameID/ThisVideo.FrameRate),'HH:MM:SS')],'Te
xtColor','white','FontSize',15,'BoxOpacity',0.2);

    if FrameID2<BufferFrames % if a motion has been detected recently
      writeVideo(MergedVideo,tmp2);
      FrameID3=FrameID3+1;
      % Compute diference
      diff1=tmp2(:,:,3)-tmp(:,:,3);
      diff1=diff1(AOI); % only keep user sellected area
      diff2=rms(diff1);
      % Detect motion
      if diff2>Tresh && FrameID>ThisVideo.FrameRate
        FrameID2=1;
      else
        FrameID2=FrameID2+1;
      end
    elseif FrameID2==BufferFrames*2 % if a motion has not been detected
recently
      % Compute diference
      diff1=tmp2(:,:,3)-tmp(:,:,3);
      diff1=diff1(AOI); % only keep user sellected area
      diff2=rms(diff1);
      % Detect motion
      if diff2>Tresh && FrameID>ThisVideo.FrameRate % if a motion is
detected
        disp(['Frame ',num2str(FrameID),', Amplitude = ',num2str(diff2)])
        % record previous frames
        if FrameID<=BufferFrames % if the buffer is not yet full
          writeVideo(MergedVideo,BufferVideo(:,:,:,fliplr(id:-1:1)));
          FrameID3=FrameID3+id;
        else

writeVideo(MergedVideo,BufferVideo(:,:,:,fliplr(BufferVideoInd)));
```

```
          FrameID3=FrameID3+length(BufferVideoInd);
        end
        FrameID2=1;
      end
    else
      FrameID2=FrameID2+1;
    end
    tmp=tmp2;
    FrameID=FrameID+1;
  end
end
close(MergedVideo)
disp([datestr(now),' Done - Frame ',num2str(FrameID3)])
```

- The function requires a single argument: the path of the recordings you want to analyse.
- Click on two opposite corners of the rectangular subset of the frame that you are interested in (a window will open to allow you to do so).
- Wait for the analysis to finish.
- Watch the summary video created in the same folder as your recordings.

VII. Troubleshooting

1. The PICT is not showing up in the list of devices connected to the Wi-Fi network

Check that the Wi-Fi network name and password match what you specified in the wpa_supplicant.conf file (see here).

Check that your Wi-Fi network is set to a 2.4GHz Wi-Fi bands (otherwise known as 802.11b/g/n standard) and not to 5Ghz bands (known as 802.11ac).

Check that you do not have several devices broadcasting a Wi-Fi network (hotspot) with the same name.

Check that the PICT is properly powered (disfunction of the USB cable, the power bank is empty, …) and the micro SD card is correctly inserted in the reader slot of the Raspberry Pi.

2. The recording stopped unexpectedly

If your recording starts but suddenly stops after a few minutes or a few hours, check that your power supply and USB power cable are working properly.

Check that enough space is available in the micro SD card (see here). You can look for available space by typing "`df -h`" in the terminal or by using an FTP client. For instance, AndFTP on Android devices. Alternatively, you can insert the micro SD card in the SD slot of a Windows computer with Disk Internal Linux Reader installed.

Check that you did add "`nohup`" at the beginning of your command line to start the recording. If it was omitted, the recording process has stopped when you were automatically logged out of the terminal once the PICT got out of reach of the Wi-Fi network.

If the problem persists while the power supply works properly, there is enough space on your micro SD card and you used the nohup command, you should re-write the micro SD card with a fresh image of your operating system and configuration. This is sometimes needed after the PICT have shut down automatically due to power failure.

3. The movies are entirely black, only the date stamp and other metadata are shown

You should check that the ribbon cable of the camera module is properly connected to the Raspberry Pi and the camera. If the issues arise only during the night, try replacing the IR LED with a new one.

4. The Rpi-Cam-Web-Interface issues the error "error in RaspiMJPEG"

Make sure that a camera is connected to the Pi. Check the insertion of the ribbon cable into the slot of the Pi and the camera. The golden/silver connectors must face down, towards the Pi and not you. The black side must face up (Fig. 2).

5. The IR LED light is visible as a reflection into the camera module at night

When the distance between the LED and the translucent food container lid is too long, the light from the LED can be reflected on the lid. This creates visual artefacts on the recordings. You can solve this problem by placing a plastic band around the LED as shown on Fig. 26.

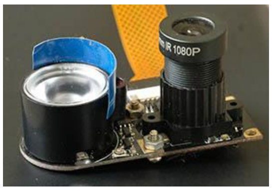


Figure 26. *Avoiding reflection on your recordings using a plastic band*

VIII. Annex

List of the software used in this guide

- SD Card Formatter v5.0.1
- Win32diskimager v1.0
- Notepad++ v7.7.1 (32-bit)
- MKVToolNIx GUI v46/v48
- Disk Internals Linux Reader v3.5.1 (64-bit)
- DVR-Scan v1.1 (or MATLAB)
- FileZilla v3.48.1 (64-bit)
- PuTTY v0.73 (64-bit)
- Termius v6.2.2
- Rpi-Cam-Web-Interface
- 7-Zip v20.00 Alpha (64-bit)

List of smartphone application

- JuiceSSH v2.1.4
- Fing v9.1.0
- Termius v4.6.2
- RaspiCAM Remote v1.17
- AndFTP v5.4

List of operating systems used in this guide

- Windows 7 & 10 (64-bit)
- Raspberry Pi Lite (32-bit)
- Android v7, 8, 9 and 10