

**Supplementary material to: Pérez-Aparicio et al. 2021: A cheap electronic sensor automated trap for monitoring the flight activity period of moths. — *Eur. J. Entomol.* 118: 315–321.**

Alicia PÉREZ-APARICIO; Jordi LLORENS; Joan Ramón ROSELL; Jordi MARTÍ;  
César GEMENO

**Supplementary Protocol for Setting up the Traps**

**Content**

1. Introduction .....	2
2. Preparing the SD card for the Raspberry and downloading the Raspberry Pi OS (operating system) .....	5
3. Starting the Raspberry for the first time .....	5
3.1. Backup Raspberry Pi Image.....	6
4. Wireless connection between PC and Raspberry .....	7
4.1. First time wireless connection of a Raspberry Pi with a given PC.....	7
4.2. Subsequent wireless connections with the PC.....	8
4.3. Problem solving: wireless connection when there is more than one network .....	8
5. Terminal window: "Talking" with the Raspberry .....	9
5.1. Installing and updating software.....	9
6. VNC Viewer: Remote access to the Raspberry and file transfer .....	10
6.1. Remote access.....	10
6.2. File transfer.....	12
6.3. Example for downloading pictures .....	12
6.4. Problem solving: VNC does not start .....	14
7. Video streaming with VLC .....	15
8. Python code .....	15
9. Capture images .....	17

## 1. Introduction

It is somewhat challenging to get started with a single-board computer, such as a Raspberry Pi, for those not particularly driven by computers and electronics (like some of the authors in this study!). There are excellent internet tutorials to get started with a Raspberry Pi Zero Wifi (see below), but even so, there are many different steps to get the Raspberry started and to operate with it (like connecting the peripherals, installing the operating system, establishing wireless connection and remote access, downloading software, transferring files and writing code), and being exposed to all of this simultaneously and for the first time may be baffling. The following guide is intended to illustrate these initial steps and the ensuing process needed to set the Raspberry Pi vision traps running, capture images and download them for analysis.

The Raspberry board comes with no operating system. The **operating system** (we used Raspbian, but now it is called **Raspberry Pi OS**) is **downloaded free from the internet** into the PC. Then it is installed in the SD card (new **SD cards** have to be **formatted** first). The SD card is installed in the Raspberry and the Raspberry is powered by connecting it via USB to a power charge (like a computer USB or a telephone charger). **At this point the Raspberry is on, but the user cannot "communicate" with the Raspberry because it has no screen, keyboard or mouse.** These are connected to the Raspberry using its USB and HDMI ports. Raspberry Pi uses "micro" USB and HDMI connectors, and there is only one USB for both mouse and keyboard. Inexperience people may be put off by this part of the process.

Wireless connection is necessary for remote access to the Raspberry, which allows running it from the PC without needing the peripherals. **Wireless** recognition between a given Raspberry Pi card and a PC using their IPs has to be established. **Remote access** is possible with free software (**VNC**). **Files can be transferred** wireless between Raspberry and PC using remote access.

The way to communicate with a Raspberry is by typing **computing code** in the Raspberry Pi OS **console**. Using the console is a hurdle for the computer novices, but it is really very easy once it is done a couple of times, and the code remains in the memory so it just has to be recalled for subsequent use.

Wireless communication requires **internet**. Indoors it may be provided by a local server, but in the field a **mobile telephone hotspot** may be required. We have encountered problems when the Raspberry detects several networks, and a tiny bit of code is required to make the Raspberry connect to the network of your choice (see 4.3 below).

The next step is to learn how to do **video streaming** to set the camera position and focus. This is done with the free software package **VLC** which needs to be installed both in the computer and the Raspberry.

The next step is to write a very simple script in **Python code** to instruct the Raspberry on how **to run the camera** (image resolution, picture frequency, etc.), and finally very simple code to ask the Raspberry run the Python code.

The trap is now ready to take pictures. Finally, image captures must be ended to download them from the SD card into the computer. After that, cleaning up the SD card is necessary to free space for further captures.

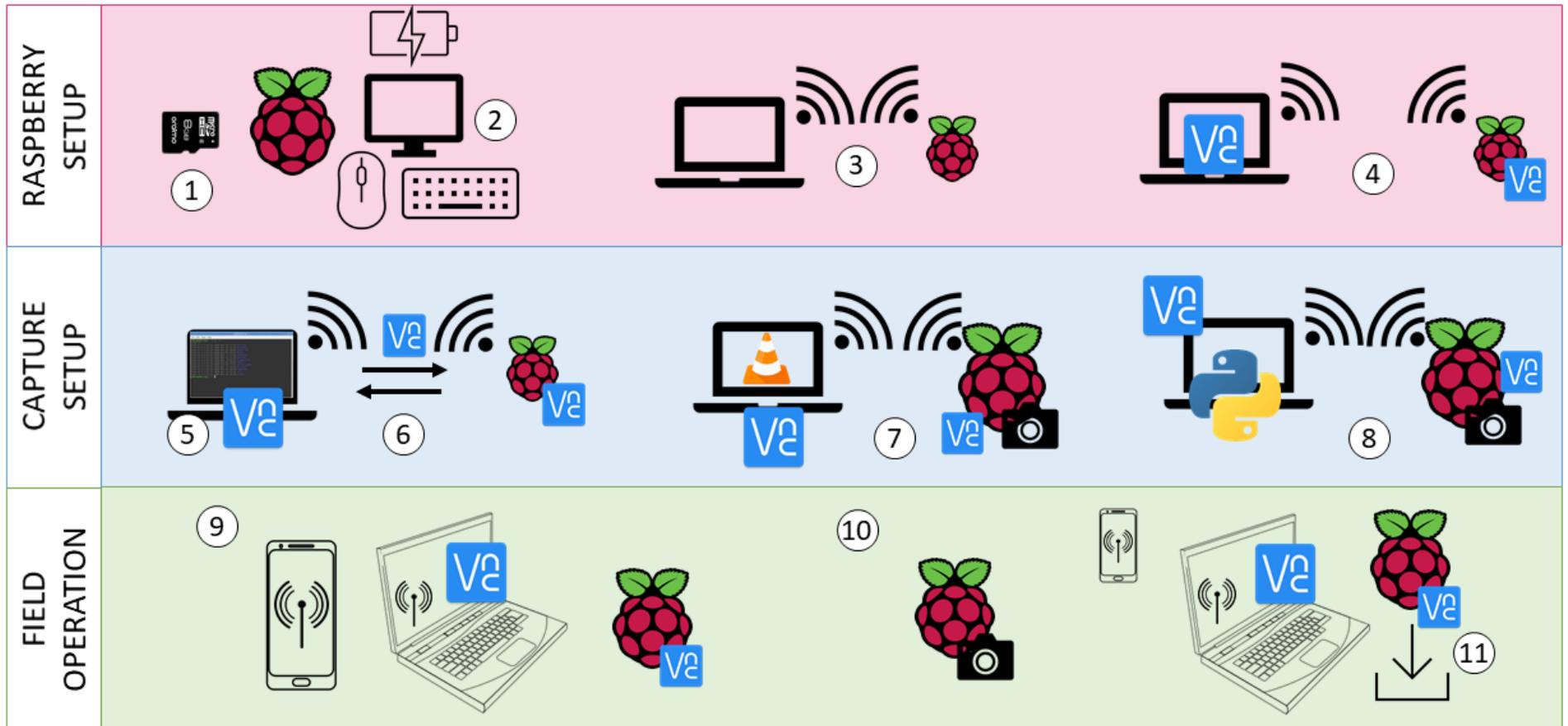
The first time that all of this is done by an amateur it seems like a daunting task and it may take several days and some frustration to complete the process, but after the first run is completed it all gets relatively easy.

### **References**

<https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up>

Short, M. Getting Started with the Raspberry Pi Zero Wireless.

<https://learn.sparkfun.com/tutorials/getting-started-with-the-Raspberry-pi-zero-wireless/all>



1. Format SD card and install OS
2. Power RB and connect peripherals
3. Stablisch Wireless connection PC-RB
4. Remote Access using VNC
5. RB OS terminal
6. Transfer files
7. VLC video streaming
8. Python image capture
9. Mobile hotspot and Remote Access
10. RB image capture
11. Download pictures

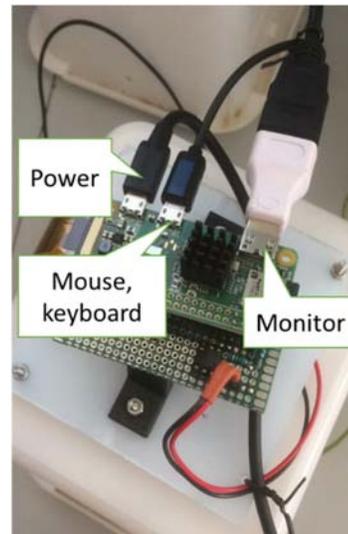
## 2. Preparing the SD card for the Raspberry and downloading the Raspberry Pi OS (operating system)

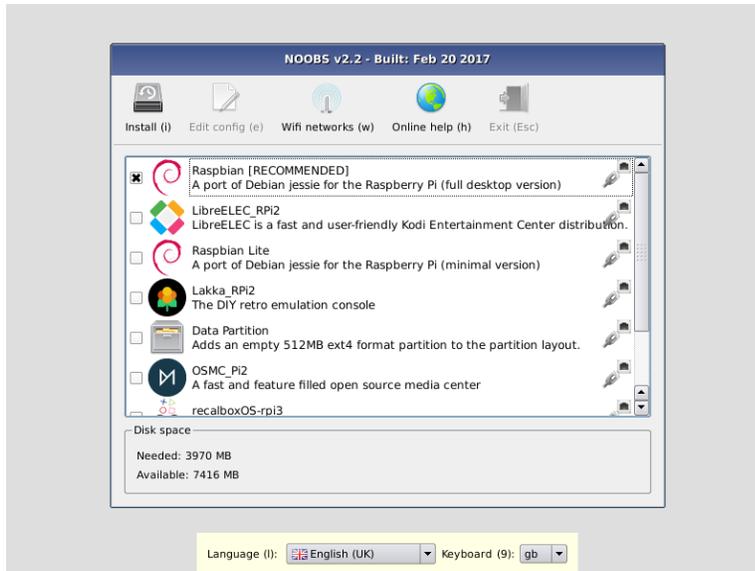
**\*NOTE:** At the time of writing this protocol the Raspberry Pi operating system has changed name: before it was called "Raspbian", now it is called "Raspberry Pi OS". Our protocol is based on Raspbian. Instructions on how to install Raspberry Pi OS in the SD card can be found in the following link. (<https://www.raspberrypi.org/software/>)

1. Insert the micro SD into a card reader to read it in the PC
2. Format the SD card. This is best done with a specific card formatting program (e.g., SD Card Formatter, <https://mygpstools.com/sd-card-formatter-how-to-use>)
3. Download Raspberry-Linux's NOOBS (not NOOBS lite) software from Raspberry page to the PC (<https://www.Raspberrypi.org/downloads/noobs/>)
4. Decompress the ZIP file and copy it to the SD card. Now the Raspberry operating system (=Raspbian) is installed in the card.

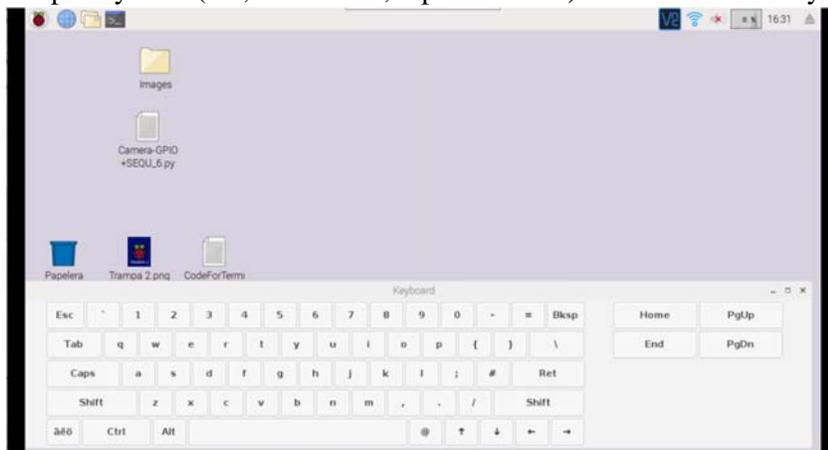
## 3. Starting the Raspberry for the first time

1. Install the SD card in the Raspberry
2. Connect monitor and mouse to the Raspberry
  - a. Mouse and monitor share micro-USB port. A USB splitter would allow connecting both of them. Another option is to activate the Raspbian's keyboard (see below). Until then, it may be required to unplug one to use the other.
3. Power the Raspberry. This is done by plugging it to a power source (like a telephone charger or a PC USB).
  - a. **Caution:** powering the Raspberry is like turning on a computer, and like a computer, once turned on it is advisable to turn it off using the program shutdown (see below), instead of unplugging it. Unplugging the Raspberry repeatedly without shutting off the operating system may cause malfunctions.
4. Follow the prompts and install Raspbian (the operating system in NOOBS)
  - a. Raspbian's user ID and password are "pi" and "raspberrypi" by default (they can be modified)

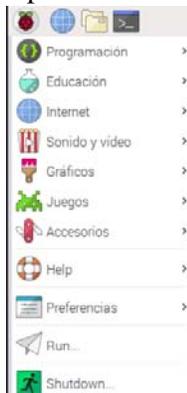




- At this point you may want to enable Raspbian's keyboard by clicking in the Raspberry icon (i.e., start menu, top left corner) → accessories → keyboard → OK.



- Shutdown. Click on the Raspberry icon and hit the shutdown icon (the figure of a person in black running on a green background)



### 3.1. Backup Raspberry Pi Image

<https://howchoo.com/g/nmexndnlmdb/how-to-back-up-a-raspberry-pi-on-windows>

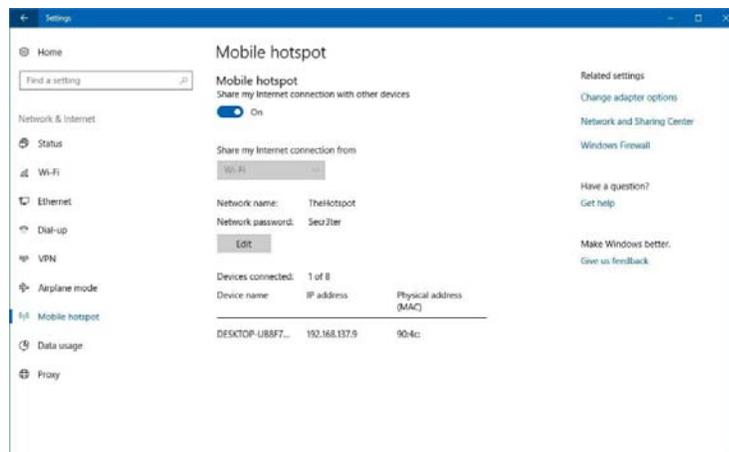
1. Download USB Imagetool.exe
2. Create an Image with the **Backup** tab (e.g., image8GB.trampa4.img)
3. Transfer an Image with the **Restore** tab
  - a. NOTE that the SD card needs to be formatted before
  - b. NOTE that if the SD card already had Linux, when in the PC WINDOWS will show as if it has several directories (F, G, etc) when in fact there is only one
  - c. NOTE that the disc size in which the image was created has to be maintained (can't put a 8GB image in an 16GB card)

## 4. Wireless connection between PC and Raspberry

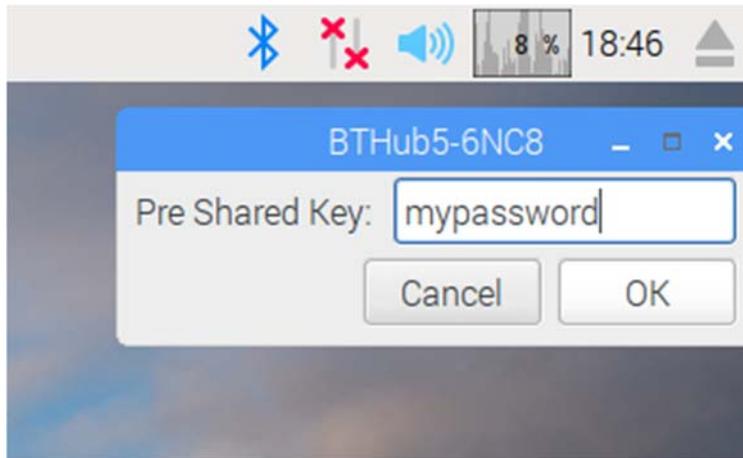
### 4.1. First time wireless connection of a Raspberry Pi with a given PC

This is necessary for remote access to the Raspberry. Notice that this exact procedure has to be done every time that the Raspberry is connected with a new computer but only for the very first time. Afterwards it is not necessary to type in the computer's IP password in Raspbian.

1. Create a hotspot with the PC. Enable Windows wireless internet access ("Mobile hotspot") in the PC by clicking on the "on/off" switch. The computer IP should show up in the bottom (Devices connected). If not, then can't continue.
  - a. Make sure that your computer is connected to an internet source, either wifi or, under field conditions, a mobile hotspot



2. Click on the Raspbian's icon with two red-color "X" symbols located on the top right corner of the window. Then, it should show the name of your computer in the blue tab, hit it and an empty cell asking for the shared key will open. Type in the computer's IP password (use Raspbian's keyboard to type it in if you have



activated it) and then hit OK.

3. After a few seconds the Raspberry should show up among the "Devices connected" section of the "Mobile hotspot" of the computer. In the Raspberry desktop the "two-red-arrows" symbol should have changed to the "blue-internet" symbol.

#### 4.2. Subsequent wireless connections with the PC

1. Make sure that you have internet access
2. Turn on Raspberry and computer as before
3. Enable Windows wireless internet access in the PC
4. Now the Raspberry should automatically show up among the devices connected

#### 4.3. Problem solving: wireless connection when there is more than one network

The Raspberry cannot connect to more than one network simultaneously. So, if there are several networks available (e.g., an open public network and the PC you want to connect the Raspberry to), the Raspberry will connect to the first one it finds and won't be able to connect to the other. In order to make the Raspberry ignore some networks and connect preferentially to others, the following commands should be typed in the terminal:

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

A file opens, type in the following:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1
```

```
network={  
    ssid="YOUR_NETWORK_NAME"  
    psk="YOUR_NETWORK_PASSWORD"
```

```
    key_mgmt=WPA-PSK
    priority=1
}

network={
    ssid="THE_NETWORK YOU WANT TO DISABLE_NAME"
    disabled=1
}
```

Several levels of priority can be set (higher numbers have higher priority) so that a network with priority=2 will be connected before a network with priority=1, if both are on reach.

- <https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>
- <https://unix.stackexchange.com/questions/80550/how-to-prevent-wpa-suplicant-from-connecting-to-a-network>

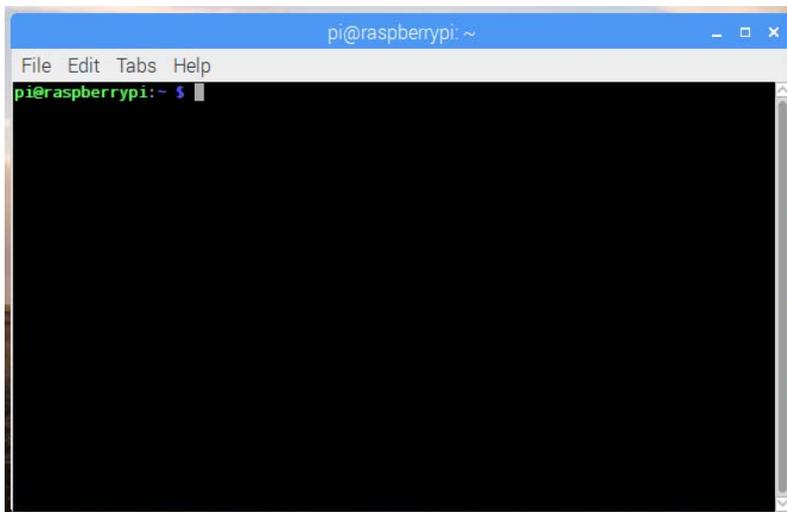
## 5. Terminal window: "Talking" with the Raspberry

This step may put off users that have never used any form of computer code. It can be quite intimidating at first, but there is not much to it. The basic idea is to instruct the Raspberry to "do things" with commands that the user writes in the terminal window. The commands are used for almost anything: downloading and updating software, starting and ending image captures and video streaming, etc.

There is an excellent online guide about this in the Raspberry's webpage:

<https://www.raspberrypi.org/documentation/usage/terminal/>

It is strongly recommended that first-time Raspberry users get acquainted with the console before moving on.



### 5.1. Installing and updating software

The command window is used to install and update software, using simple commands:

```
sudo apt-get update
```

`sudo apt-get upgrade`

`sudo reboot`

These commands will update everything and then restart the Raspberry Pi. The first command, `sudo apt-get update`, will update all the package indexes. The next command, `sudo apt-get upgrade`, actually performs upgrades of software on your Raspberry Pi. The "sudo reboot" command reboots your Pi so it can start up with the shiny new firmware ( if it has been updated ), you can wait until you next turn off your Pi naturally but you need to reboot before everything will be properly up to date.

**Source:** <https://learn.pimoroni.com/tutorial/raspberry-pi/keeping-your-raspberry-pi-updated>

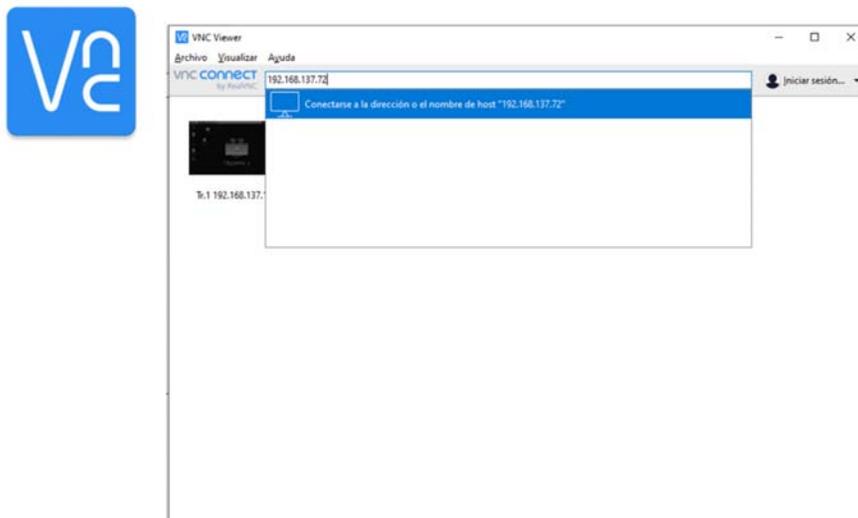
**Note:** It is a good idea to save the commands in a text file and keep it in the Raspberry's desktop for use as needed

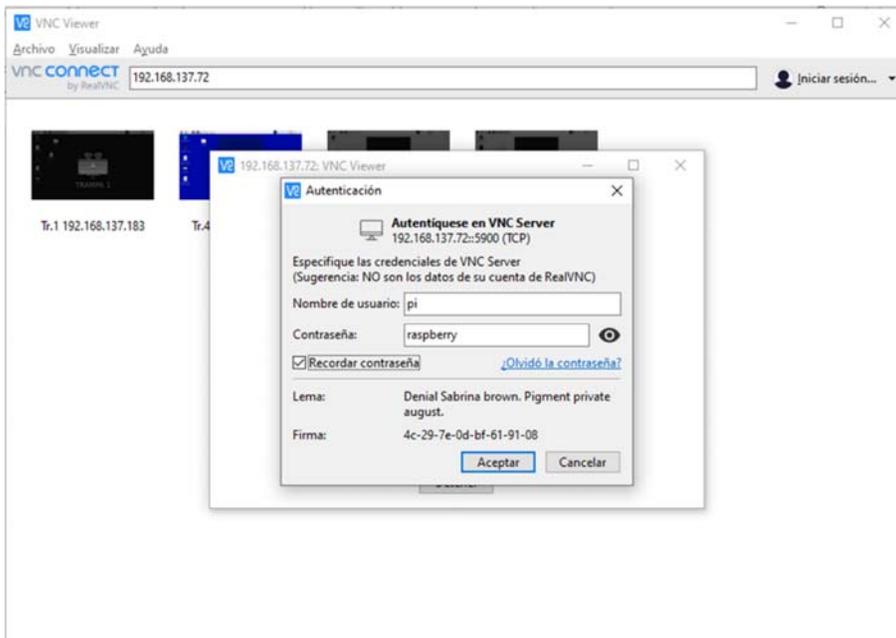
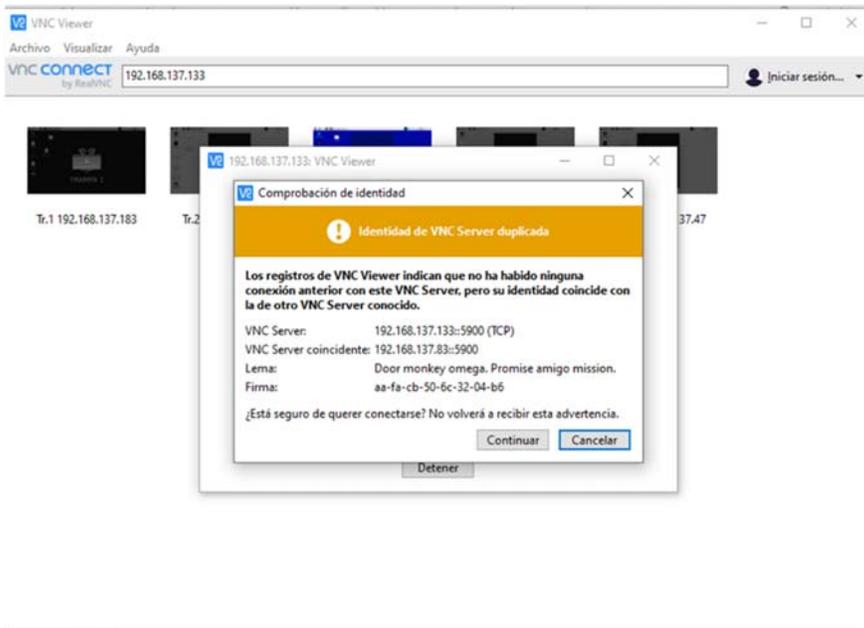
## 6. VNC Viewer: Remote access to the Raspberry and file transfer

Remote access allows working with the Raspberry from the computer, without the need of any peripherals. Once connected remotely, the Raspberry shows up on the PC's screen and it can be operated with the PC's mouse and keyboard. It is an essential function in order to program the Raspberry and download images from Raspberry to PC in the field.

### 6.1. Remote access

1. Install VNC Viewer in the PC  
(<https://www.realvnc.com/en/connect/download/viewer/>)
  - a. Raspbian already has VNC installed.
2. Open VNC viewer and type in the IP of the Raspberry
  - a. The IP shows up in the Mobile Hotspot page of the PC.
  - b. Notice that the IP of the Raspberry *may* change from day to day
3. It will ask the username ("pi") and password ("raspberr").





At this point the Raspbian's desktop will show up in the computer screen, and the computer's keyboard and mouse will operate the Raspbian's desktop.

5. To exit the remote access VNC just move the mouse to the top of the window until a tab expands. Then press the "X" symbol.
  - a. **Caution:** exiting the remote access does not shutdown the Raspberry. These are two separate processes and it is easy to get confused and exit VNC thinking that the Raspberry has been shutdown.
    - i. To let the Raspberry running: 1) exit VNC, 2) **DO NOT shutdown** the Raspberry
    - ii. To shutdown the Raspberry: 1) **DO shutdown** the Raspberry, 2) exit VNC

## 6.2. File transfer

It is necessary to transfer files containing code from the computer to the Raspberry Pi and pictures from the Raspberry Pi to the PC. To transfer files from the PC to the Raspberry Pi and the other way around the VNC Viewer application is used

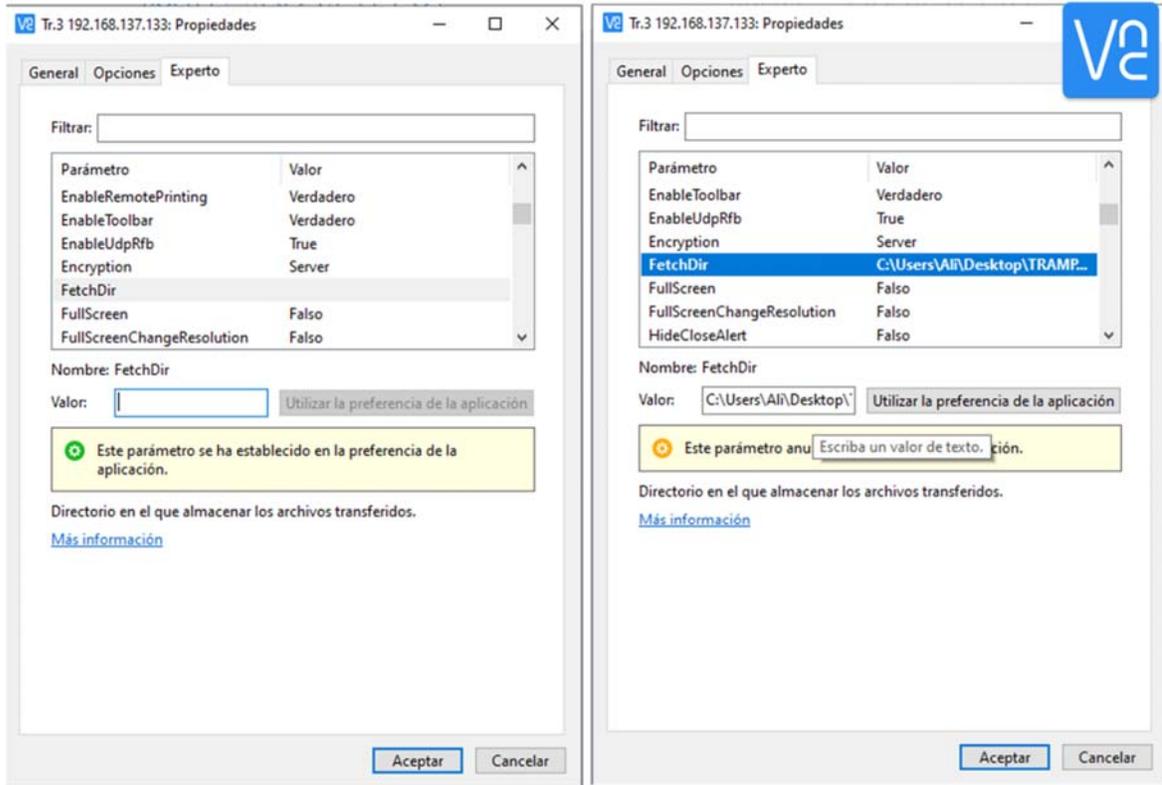
1. To transfer files **from the PC to the Raspberry Pi** click in the VNC Viewer icon which is at the top of the window: and follow the instructions 
  - a. **Source information and Tutorial:** <https://help.realvnc.com/hc/en-us/articles/360002250477#sending-files-to-vnc-server-0-0>
2. To transfer files **from the Raspberry Pi to the PC**, use VNC Viewer to open the VNC Server dialog remotely, select **Menu > File transfer**, and follow the instructions. 
  - a. **Source information and tutorial:** <https://help.realvnc.com/hc/en-us/articles/360002250477#fetching-files-from-vnc-server-0-1>
3. By default, files are downloaded to the desktop. To change this for future file transfer operations, select an option from the **Fetch:**
  - a. Show the hidden menu on the top of the screen
  - b. Click on options icon (the gear symbol)
  - c. Go to "expert" on the top right tab
  - d. Click on Fetch directory on the list of options
  - e. Change the destination directory

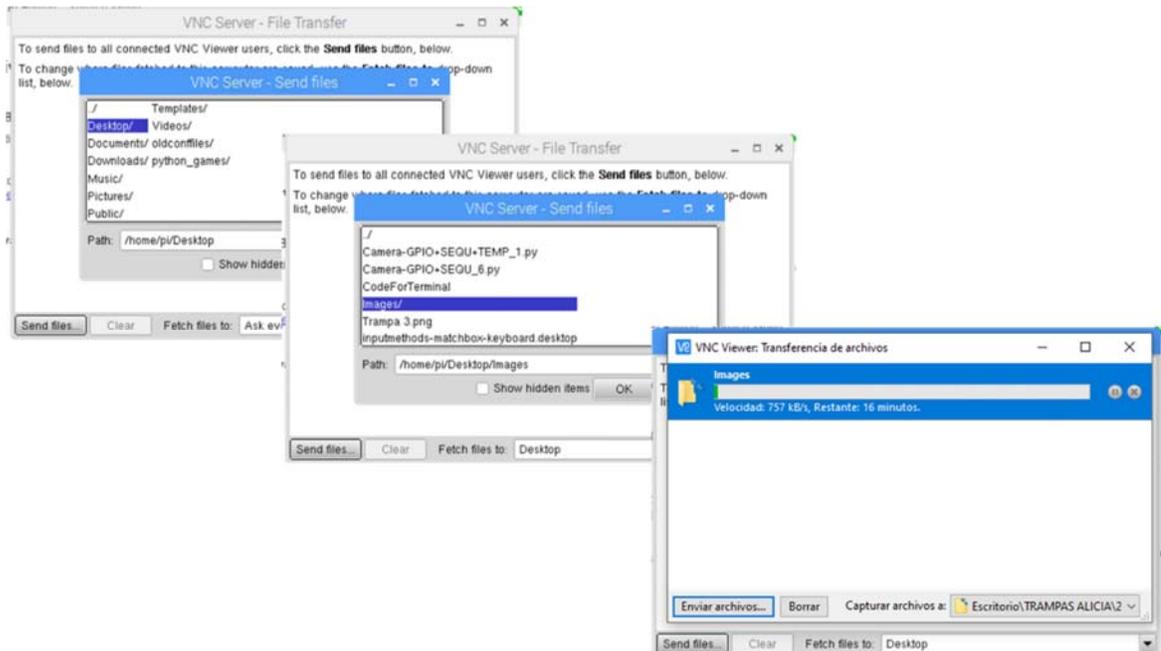
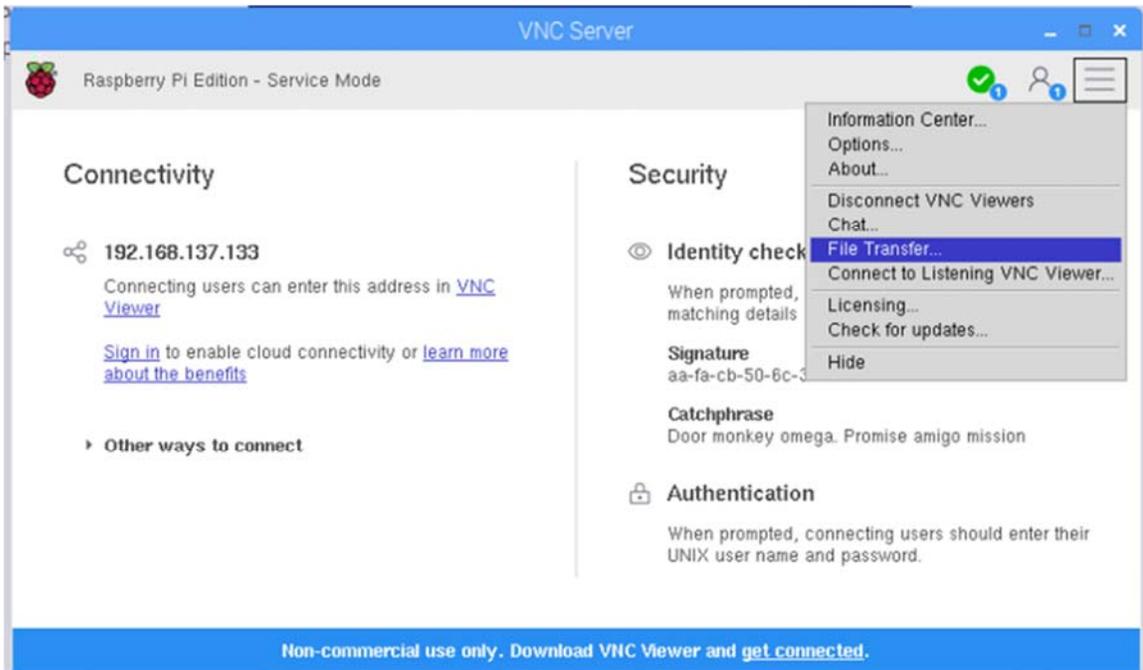
## 6.3. Example for downloading pictures

(Alicia PPT presentation)

1. VNC viewer/server, RIGHT-click on the black icon with blue letters "VNC" 
2. Go to **options** in **tools**
3. In the **expert** tab look for **FetchDir**
4. In the PC, go to the folder where you want to save the images, copy the address and paste it in FetchDir "value". The pictures will be then downloaded directly in the selected folder instead of the default one
5. Click on "File transfer"
6. Click on "Send files"
7. Choose the path where the picture files are located in the Raspberry (Perhaps: /home/pi/Desktop/Images/)

## 8. Send





#### 6.4. Problem solving: VNC does not start

We had this problem and it was related to having or not peripherals connected to the Raspberry Pi. If there was something connected, then VNC started alright, but if there was nothing connected to the USBs, the VNC did not start. The solution is to "fool" the Raspberry with a "virtual mouse" by typing the following command in the console:

`sudo apt install haveged`

- Source:  
<https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=230779&p=1413648#p1413648>

## 7. Video streaming with VLC

This step is needed only for real-time adjustment of focus and position of the camera when they are first installed on the traps, or when they are reinstalled for trap maintenance.

1. Power the Raspberry Pi
2. Activate wireless hotspot in the PC
3. Open VNC Viewer and connect with the Raspberry
4. Type the following code in the terminal:
  - a. `sudo apt-get install vlc`
5. Type the following code in the terminal:
  - a. `raspivid -o - -t 0 -hf -w 800 -h 800 -fps 24 |cvlc -vvv stream:///dev/stdin -sout '#standard{access=http,mux=ts,dst=:8160}' :demux=h264`
6. At this point the terminal looks like this:

```

b3a00520 main input debug: Buffering 0%
b2a00b20 main input debug: Buffering 6%
01910a90 main stream output debug: adding a new sout input for 'h264' (sout_in
out: 0xb3a02e30)
0191bae8 main mux debug: adding a new input
0191bae8 mux_ts mux debug: adding input codec=h264 pid=100
0191bae8 mux_ts mux debug: new PCR PID is 100
b3a00520 main input debug: Buffering 13%
b3a00520 main input debug: Buffering 20%
b3a00520 main input debug: Buffering 26%
b3a00520 main input debug: Buffering 33%
b3a00520 main input debug: Buffering 40%
b3a00520 main input debug: Buffering 46%
b3a00520 main input debug: Buffering 53%
b3a00520 main input debug: Buffering 60%
b3a00520 main input debug: Buffering 66%
b3a00520 main input debug: Buffering 73%
b3a00520 main input debug: Buffering 80%
b3a00520 main input debug: Buffering 86%
b3a00520 main input debug: Buffering 93%
b3a00520 main input debug: Buffering 100%
b3a00520 main input debug: Stream buffering done (320 ms in 72 ms)
b3a00520 main input debug: Decoder wait done in 0 ms
  
```

7. Open VLC media player in the PC (free download form the web)
8. Media --> Open Network Stream
9. Introduce the URL
  - a. `http://raspberrypi IP number:8160` (the raspberrypi IP number can be seen in the VNC window or in the wireless hotspot PC window)
10. At this point the image captured by the camera should show up in the PC monitor
11. Adjust the camera focus and position to your needs
12. Type "control + c" in the Raspberry's console to stop video streaming

## 8. Python code

This code sets the camera functions and instructs the Raspberry on how to capture pictures. Write the code in .txt format and save it in the Raspberry's desktop. Change the file extension to ".py". This code will be run using the commands shown in the following section ("capture images"). **Note:** do not modify the Python code while the Raspberry is capturing.

```

from time import sleep
from picamera import PiCamera
from datetime import datetime, timedelta
import RPi.GPIO as GPIO
import time
import os

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(3,GPIO.OUT)

def wait():
    # Calculate the delay to the start of the next hour
    next_hour = (datetime.now() + timedelta(minutes=10))
    ## replace(hours=1, minutes=0, seconds=0, microseconds=0)
    delay = (next_hour - datetime.now()).seconds
    sleep(delay)

def measure_temp(): ##Function to measure CORE temperature.
    temp = os.popen("vcgencmd measure_temp").readline()
    return (temp.replace("temp=", ""))

camera = PiCamera()
camera.resolution = (1500, 1500)
#camera.resolution = (3280, 2464)
camera.ISO=0 # Range (0..1600) 0="auto". Luz natural=100.
#camera.framerate = 30 #video
#camera.exposure_compensation=0 # Range (-25..25) Millor no forsar la
comensacio
camera.exposure_mode='auto' #('off' 'auto' 'night' 'snow')
camera.meter_mode='average' #('average' 'spot' 'backlit' 'matrix')
camera.shutter_speed=0 # 0=AUTO, Velocitat d'opturacio. Shutter=1000*1/segons
camera.awb_mode='auto' # Get or set the auto-white-balance gains of the camera.
('off' 'auto' 'sunlight' 'cloudy' 'shade' 'tungsten' 'fluorescent' 'incandescent' 'flash'
'horizon')
#camera.awb_gains=1.2 # Range(0-8)

camera.brightness=60 # Range (0..100). Son correccions de color a posteriori. Ajuda
a veure millor la imatge.
#camera.sharpness=0 # Range (-100..100). Son correccions de color a posteriori.
#camera.contrast=10 # Range (-100..100)
#camera.saturation=0
#camera.awb_mode='auto' #('off' 'auto' 'sunlight' 'cloudy' 'shade' 'tungsten'
'fluorescent' 'incandescent' 'flash' 'horizon')
camera.image_effect='none' # All options: http://picamera.readthedocs.io/en/release-1.10/api\_camera.html
camera.color_effects= None # Gray scale image (128,128) or normal color mode:
None

counter=0

```

```

while counter < 10000000:
    #value=(counter*0.1)+1
    #camera.exposure_compensation=value
    #print(value)
    ##print({ timestamp:% Y-%m-%d-%H-%M-%S })
    ##camera.start_preview()
    ##sleep(3)
    GPIO.output(3,GPIO.HIGH) # Encen el LED 18. Controlem tots els LEDS (4) de 3
mm.
    #print('LEDS ON')
    sleep(2)
    timestamp=time.gmtime()
    ##filename=('/home/pi/Desktop/Other images/img_2_{ timestamp:% Y-%m-%d-%H-
%M-%S }.jpg')
    present_time =time.strftime("% Y-%m-%d-%H-%M-%S",timestamp)
    print(present_time)
    temp = os.popen("vcgencmd measure_temp").readline()
    temp=temp.rstrip("\n")
    camera.capture('/home/pi/Desktop/Images/R_' + present_time + '+temp+'.jpg')
    GPIO.output(3,GPIO.LOW)
    print(temp)
    print('LEDS OFF')
    ##camera.stop_preview()
    counter= counter+1
    print(counter)
    wait()

```

## 9. Capture images

1. The console shows: "pi@Raspberrypi: ~"
2. Type "cd/home/pi/Desktop"
3. Enter
4. It shows "pi@Raspberrypi: ~/Desktop S"
5. Type "name of the python code file.txt" (the name that you gave to the
6. Enter. The Raspberry starts taking pictures with the settings of the python file
7. Next
  - a. To see the captures (and make sure that it is capturing correctly): Python code instructs that the pictures are saved in a folder called "Images" located in the desktop. To see captures open the "Images" folder in Raspberry's desktop
  - b. To continue capturing away from the computer: close VNC viewer (on the hidden tab that is on the top of the screen, hit the "X"), but do not shut down Raspberry and do not close Raspberry's terminal program window!!!
  - c. To stop captures: control + c
  - d. To shut down Raspberry: **first** disconnect the Raspberry (top left tab with Raspberry logo, green tab with running guy "shutdown") and **then** VNC viewer (on the hidden tab that is on the top of the screen, hit the "X")