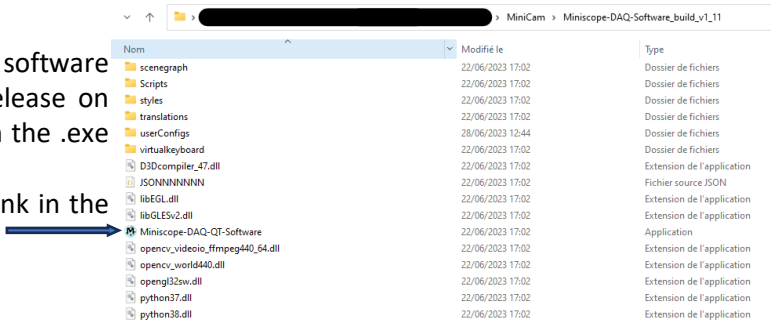
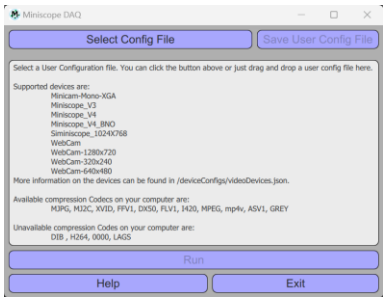
MiniCam User Guide

1) Getting started

Prerequisite : The Miniscope-DAQ-QT-Software is the software needed alongside the camera. Download the last release on this link. Once the zip file is downloaded, unzip it, run the .exe file and follow the instructions.
Once ready, open the application by clicking on the link in the folder created :



You will find the following interface :



The camera needs a User Config file to get working with the software. Written in JSON format, this file sets up all the parameters concerning the camera. Click on "**Select Config File**" and load the provided file.

You will find an interface where you can edit various settings for the camera and recordings. You can find details on how to edit the camera settings below.

Click on "**Run**" to get the camera working.

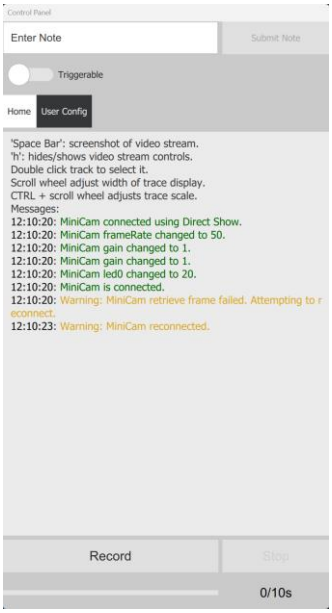
2) Working interface, recording

The GUI and data connections will be created based on the loaded User Configuration File. Use the sliders and buttons to adjust various parameters (framerate, LED intensity…).

You will also find on another window the Control Panel.

To record a clip, click and hold "Record". duration of the video and file directory are chosen in the Config File (see below).

Note : if you have important frame losses, try this: click "Exit" on the UI, plug off ad back on the camera, and restart the Software



3)Settings

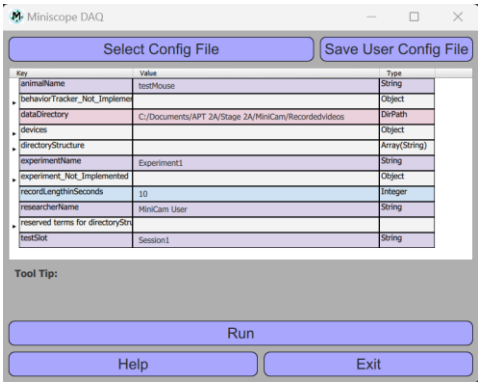
Settings are managed by the User Config File. You can edit it through the User interface after loading it in the app, or by directly editing the raw file.

a) With the User interface

You can edit the boxes by clicking on it and typing what you want. You can notably edit the recording length and data directory on this window.

b) Editing the raw file

Another way is opening independently the config file and editing it directly. Further instructions are indicated directly on it.



4)Data management

Recorded videos will be stored as 60 FPS videos no matter what you choose on the interface. To retrieve the corrected data, you must execute a Python program provided with the config file. Instructions are on it. Edit the path of the video to correct and the one of the corrected video that will be created and execute it.

Note : If you record long videos (more than a few minutes), ensure that the total expected frames does not exceed the "framesperfile" figure in the ConfigFile, otherwise update it for it to remain bigger.

VIDEODATA ANALYSIS

Throughout this document are mentioned other documents, such as python programs, csv files or else. You can find all these documents in the "MACHINE_LEARNING_TOOLS" GitHub repository.

1) Preparing MiniCam raw videos

The MiniCam provides us videos with two issues to address.

First issue :
The GUI allows you to choose on what framerate you are willing to record. No matter what you choose, you will have a slight offset between the chosen value and the actual framerate of recorded videos.

Second issue :
No matter what framerate you pick; the video you will effectively store on your computer will be a 60 fps (frames per second) video. There is actually not any loss of information, it is just an issue with the video compression. What's wrong is the time delay between each frame. Example : if you record a 30 fps 10 seconds clip, you we find yourself with a 60 fps 5 seconds video, so the video is in that case sped up by two.

I have created a **python program to address those two issues**, taking the actual video as a parameter and using it to create a new corrected video.

What do you need to run this program ?
- The absolute path of the video you want to edit
- The real framerate of the video you recorded (addressing the first issue)

How to get the framerate of your recorded video ?
The video recorded with the MiniCam comes out with a TimeStamps CSV file. Open this file and follow these steps to retrieve the actual framerate of your video :

<u>1- Edit the display of information of the file BAD ENGLISH ???? :</u>
Select all the data by clicking on the column A, go to "Data", "Text to Columns" : a pop-up window will show up. Select "Delimited", then "Comma", and "Finish". The change must look like this :

	A	B	C	D	E
1	Frame Number,Time Stamp (ms),Buffer Index				
2	0,-4,0				
3	1,16,0				
4	2,39,0				
5	3,59,0				
6	4,81,0				
7	5,102,0				
8	6,123,0				
9	7,144,0				
10	8,166,0				

	A	B	C
1	Frame Number	Time Stamp (ms)	Buffer Index
2	0	-4	0
3	1	16	0
4	2	39	0
5	3	59	0
6	4	81	0
7	5	102	0
8	6	123	0
9	7	144	0
10	8	166	0

<u>2- Edit the actual data :</u>
- In column D, enter "=B3-B2", and drag all the way down to extrapolate this formula for all lines. This gets you the inter-frame interval for each frame, in milliseconds.
- In column E, enter "=1000/D2" (admitting the first inter-frame interval value is in this place) and drag all the way down as well. You now have the instantaneous fps value for each frame.
- In column F, enter"=SUM(E:E)/(total number of frames)" to get the real average FPS value of your video !

(You can get the total number of frames in your clip by looking at the last value of the "Frame number" column of your file)
The edit should look like this :

	A	B	C
1	Frame Numb	Time Stamp	Buffer Index
2	0	-8	0
3	1	14	0
4	2	34	0
5	3	56	0
6	4	77	0
7	5	98	0
8	6	120	0
9	7	141	0
10	8	162	0

	A	B	C	D	E	F
1	Frame Numb	Time Stamp	Buffer Index	Inter-frame interval	Instant FPS	Av. FPS
2	0	-8	0	22	45,4545455	47,3010474
3	1	14	0	20	50	
4	2	34	0	22	45,4545455	
5	3	56	0	21	47,6190476	
6	4	77	0	21	47,6190476	
7	5	98	0	22	45,4545455	
8	6	120	0	21	47,6190476	
9	7	141	0	21	47,6190476	
10	8	162	0	22	45,4545455	

You can now go to the corresponding python program "framerate correction" and run it with the correct parameters. Further explanations are on it.

2) Actual use of DLC

You can find specific information about DLC on its GitHub repository.

Create a dedicated environment as explained on GitHub and install DeepLabCut within it. Run the dedicated notebook and follow along.

Labelling step :

When you run the labelling function, a window will appear.

Open the folder containing the frames to label by selecting file → open folder, select the correct folder.

Before starting to label the mouse, use the interface to store the coordinates of the edges : (edges of the plus maze or of the open field depending on the experiment). You will find the **"Retrieving coordinates data" diagram** explaining more precisely the data needed to be stored at the end of this document or on our GitHub repository.

You only need to write them down on the first frame, as the device isn't supposed to move throughout the experiment.

You can then start labelling the frames one by one. Don't forget to save your data at the end using Ctrl S or file → save selected layers.

All the other steps are displayed in the example jupyter notebooks available on GitHub.

3)DLC-processed files analysis

You will be provided with a CSV file for each video you have analysed with DeepLabCut. You can find an example of this file on GitHub.

Two Python programs are available on GitHub to calculate how much time a mouse spends on the opened and closed arms of the maze, or in a specific place on the open field depending on the experiment.

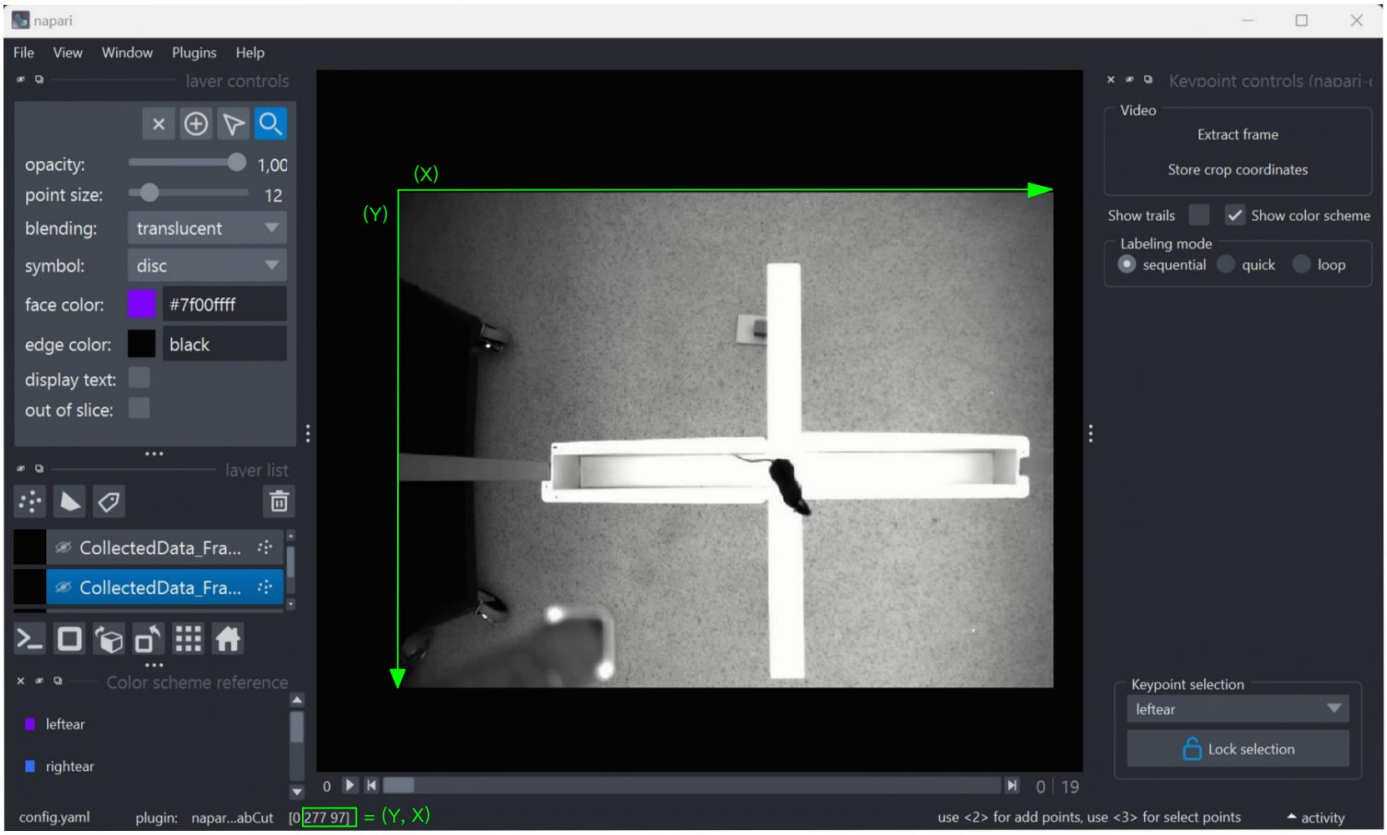
These Python programs are available on GitHub as **"program time spent in a square"** and **"program time spent in the maze"**.

These programs take as parameters the CSV file created with the coordinates of the mouse for each frame, and return the figures of time spent in each place in each area.

Further information is available within the programs.

RETRIEVING COORDINATES DATA

This window will appear when you run the labelling function :



The two numbers framed in green in the image above are the coordinates of the pixel your cursor is pointing at, updated in real time. First framed number is the Y coordinate, second is the X coordinate. (Corresponding axes are added in green on the image)
We need to retrieve some coordinates data on the structure we are working on for our programs to work.
These green framed numbers will be the numbers we look at when acquiring the coordinates throughout this procedure

1) Maze coordinates :

For the + maze program, we need to retrieve all these coordinates :

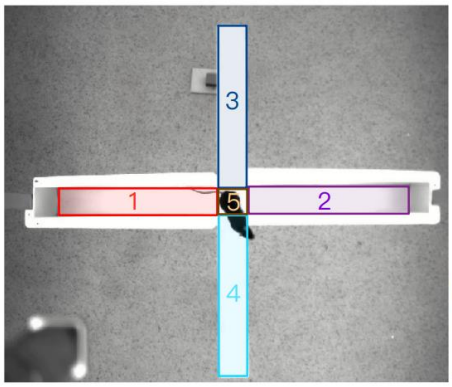
```
# +maze made by five rectangles : leftRectangle, RightRectangle, TopRectangle; BottomRectangle, plus the Middle.
# (leftRectangle, RightRectangle) = closed arms
# (TopRectangle; BottomRectangle) = Opened arms
# we want to define a square of interest : if the mouse is in it, count it, for each frame.
#coordinates of these rectangles :
    Lxmin = 284
    Lxmax = 576
    Lymin = 409
    Lymax = 456

    Rxmin = 628
    Rxmax = 929
    Rymin = 403
    Rymax = 451

    Txmin = 577
    Txmax = 628
    Tymin = 110
    Tymax = 404

    Bxmin = 579
    Bxmax = 632
    Bymin = 454
    Bymax = 753
```

To do so, we divide the maze into 5 separate areas :



1 = Left arm
2 = Right arm
3 = Top arm
4 = Bottom arm

As you can see, you have to retrieve **4 numbers for each arm** : two x coordinates, and two y coordinates.
Lets take the left arm as an example. There are four data to acquire : Lxmin, Lxmax, Lymin, and Lymax.
For the x coordinates, put your cursor on the left edge of the mouse walking area on the left arm, and retrieve the X coordinate : this will be the Lxmin coordinate.
Put your pointer on the right edge of the area, retrieve the X coordinate, you will get the Lxmax coordinate.
Same goes for the Y coordinates. This time, put your cursor on the top edge of the area, and retrieve the Y coordinate : this is your Lymin coordinate.
Do the same thing for the bottom edge of the area to get the Lymax coordinate.

Keep in mind throughout the procedure that you need to look at the left number for Y coordinates and the right number for X coordinates !

Repeat this for each arm to obtain all the coordinates needed for the + Maze program.

2) OpenField coordinates :

For OpenField experiments, you will need to define a square of interest to measure the time spent by a mouse in the center of the field. You will therefore only need 4 coordinates. Repeat the steps above as if it were only one arm of the maze, and retrieve the coordinates needed for our program.