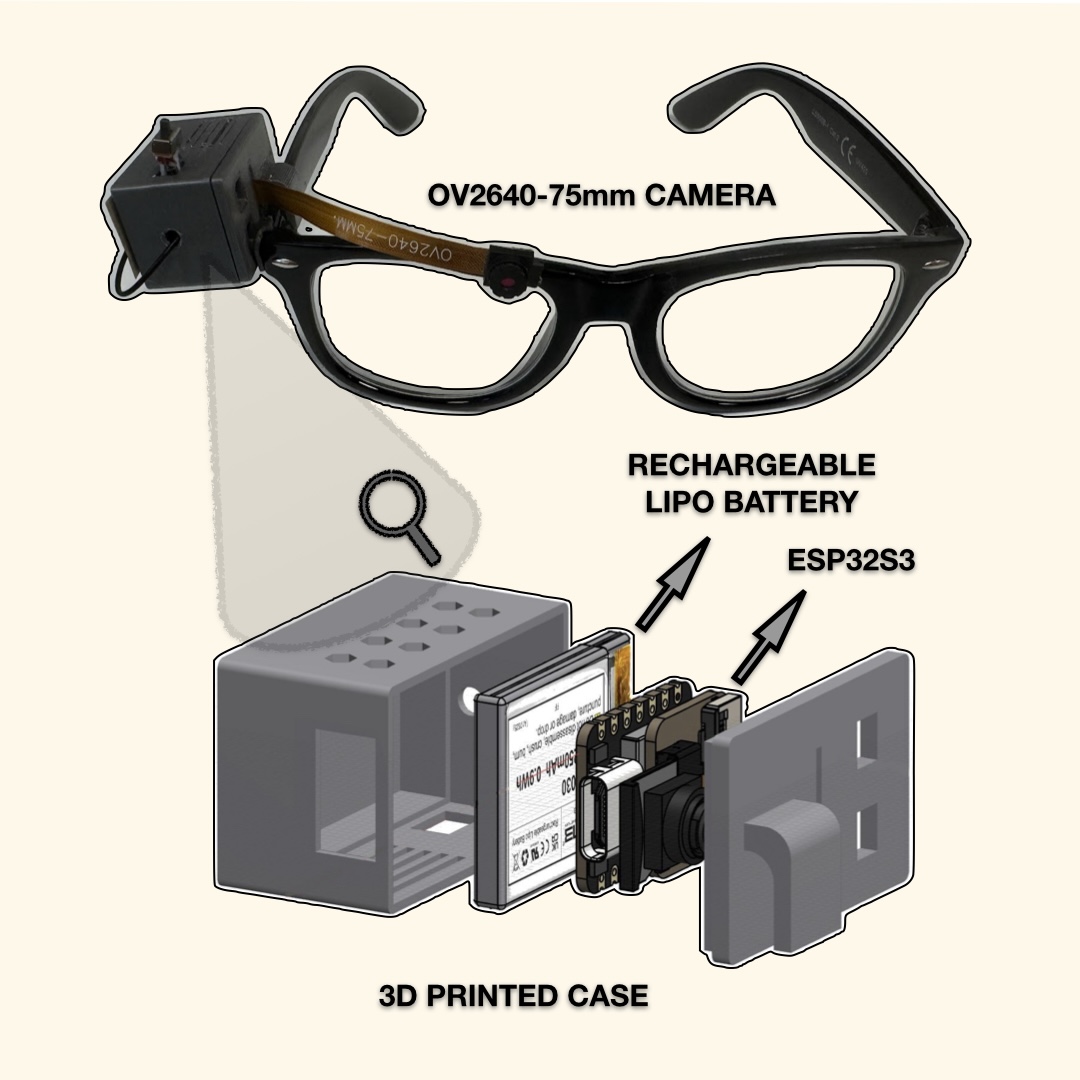
USER GUIDE CAMERA GLASS

# Description

This Guide show the fabrication and the functionalities of our camera glass device. This device works with Bluetooth (with links with IM-TWIN and ECHO Android Appication, some important informations can be found in the Echo repo) and the firmware is available at the following GitHub page <https://github.com/giampierobartolomei/Camera_Glass>.

 FOV 60 degrees (in this photo the distance from the subject is 1 meter)

The code used is partially modified from [Original Code Repo](https://github.com/s60sc/ESP32-CAM_MJPEG2SD). This repo is costantly updated, so in the future can be used for troubleshooting and for adding new features.

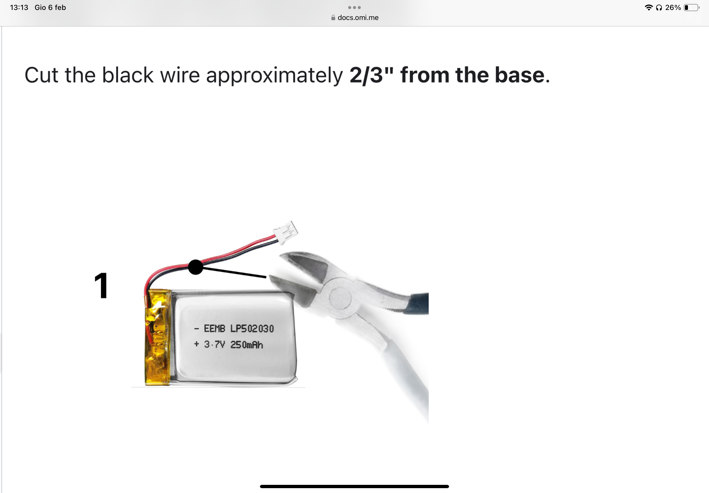
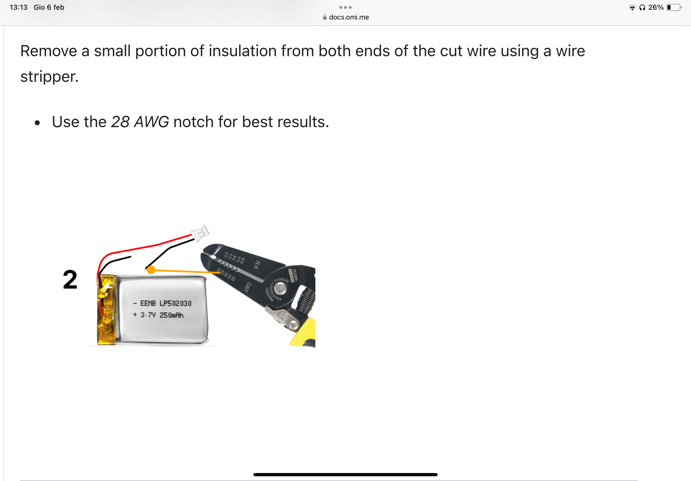
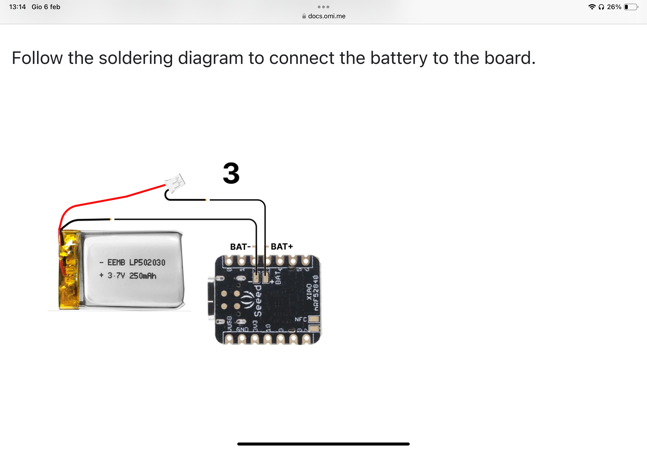
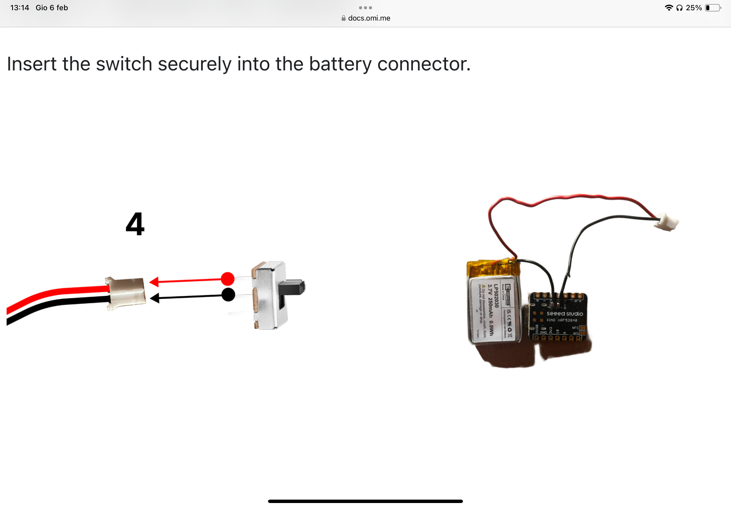
# Bill of Materials

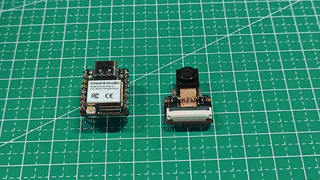
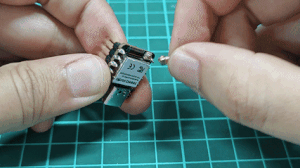
|  |  |  |
| --- | --- | --- |
| Material | Cost | Link |
| LP502030-PCM-LD/3.7V 250mAh LI-ION POLYMER BATTERY | 8.99 Euro | <https://amzn.eu/d/fRJ3vzB> |
| 3D printed case | N/A | [GitHub](https://github.com/giampierobartolomei/Camera_Glass) |
| 20CM length OV2640 2 million pixels 66 degrees camera module | 4.69 Euro | [AliExpress](https://it.aliexpress.com/item/1005004670289539.html?src=google&pdp_npi=4@dis!EUR!4.69!4.69!!!!!@!12000030047068880!ppc!!!&src=google&albch=shopping&acnt=272-267-0231&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=UneMJZVf&gclsrc=aw.ds&&albagn=888888&&ds_e_adid=&ds_e_matchtype=&ds_e_device=c&ds_e_network=x&ds_e_product_group_id=&ds_e_product_id=it1005004670289539&ds_e_product_merchant_id=109211432&ds_e_product_country=IT&ds_e_product_language=it&ds_e_product_channel=online&ds_e_product_store_id=&ds_url_v=2&albcp=20705056984&albag=&isSmbAutoCall=false&needSmbHouyi=false&gad_source=1&gbraid=0AAAAAoukdWNoL17o7fAJplJ7qB8a5TdBJ&gclid=CjwKCAiA2JG9BhAuEiwAH_zf3rouvPF3KxE00QSjF76YsPCjw2kG5WVqqPyXI3o1WDTSkx5Pw9ibFxoCiVIQAvD_BwE) |
| Solder, Wires, 1 switch, bioadhesive tape 3M and super attack | N/A | N/A |
| Xiao ESP32S3 Sense Module | 15 Euro | [AliExpress](https://it.aliexpress.com/item/1005007598922725.html?pdp_npi=4@dis!EUR!13.69!13.69!!!100.88!100.88!@212a65a717387057013297085d111f!12000041451018849!affd!!!&dp=CjwKCAiA2JG9BhAuEiwAH_zf3ogYDZnkIgsZxtqDq7Icw-uv2r_1mToMyWSG9MXS6uxvhKoLJdHcNhoC3xwQAvD_BwE&gad_source=1&gbraid=0AAAAADihhqU4HcLVNh9mD8oWdHLseg43X&gclid=CjwKCAiA2JG9BhAuEiwAH_zf3ogYDZnkIgsZxtqDq7Icw-uv2r_1mToMyWSG9MXS6uxvhKoLJdHcNhoC3xwQAvD_BwE&aff_fcid=8b16b4f852b04af4ada2ad1faa423efa-1738843874940-09276&aff_fsk&aff_platform=api-new-product-query&sk&aff_trace_key=8b16b4f852b04af4ada2ad1faa423efa-1738843874940-09276&terminal_id=f4742427721942f1ab15691ba183b4d8&afSmartRedirect=y) |

# Device Fabrication

Follow these simple steps:

1. Download And 3D print stl or gcode from the GitHub repo.
2. Manage the Wires as follows in the pictures



1. Now take the module and install the antenna, the espansion board and the right ov2640 camera (200 mm). Then insert a microSD (FAT32!). These steps and some of the next are also explained [Here](https://wiki.seeedstudio.com/xiao_esp32s3_getting_started/). This page shows the complete guide for first use and can be useful for troubleshooting.
2. Clone the repository and copy the ‘data’ folder (CameraGlass/ESP32-CAM\_MJPEG2SD/data) on the microSD. Then Connect the device to a computer and load the .ino code with Arduino Uno taking care of these requirements: Espressif Core version 3.1.1 ([Installation Guide](https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html)), PSRAM active (Arduino uno tool bar -> Tools -> PSRAM > "OPI\_PSRAM", Tools -> Partition Scheme -> 3 MB APP. If you have some trouble uploading the code (check loading blink Example from Arduino) disconnect the device, press and hold Boot button and while doing it reconnect the device to the computer and finally release the button.
3. Use a bioadhesive tape to glue the center of a glass and the retro of the ov2640 camera and use attack to glue these two piece of tapes. Then use the same method as before tu glue the cap of the case to the stick, without closing the hole for camera strip. Put the battery on the case, take out from the hole antenna and switch cables and secure the antenna with biohadesive tape (without using extra glue). Finally insert the camera module, with usb-port next to the biggest hole and close by pressing the case on the cap.

# First Use

OLD: On first installation, the application will start in wifi AP mode - connect to SSID: ESP-CAM\_MJPEG\_..., to allow router to be selected and router password entered via the web page on 192.168.4.1. The configuration data file (except passwords) is automatically created, and the application web pages automatically downloaded from GitHub to the SD card /data folder when an internet connection is available.

At this point the interface allows you without modify the code to: decide Video Settings, DISABLE MOTION DETECT!! (this setting autorecord when recognize a movement, disable it in first use and save settings), Start a stream to check the video recording, start manually the recording and finally have access to SD memory in order to see or delete files. In settings, you can also put the SSID and password of your Wi-Fi and a new IP address for connecting from your Wi-Fi is created.

NEW: Now the device works only with bt functionalities, and the settings (SVGA resolution, 20 fps, motion detect disabled and bt connection). Just compile the .ino file, and download the eco app to manage the camera recording.

# Computer vision data settings – Linked with computer vision repo

In order to process the recorded video with computer vision program follow these steps:

1. settings: SVGA (600X800) 20 fps. ALREADY SAVED.
2. Record the video, and it will be saved in the microSD card.
3. Save the file on the computer and run this command on the same directory: 'ffmpeg -i input.avi -c:v libx264 -crf 23 -preset medium -c:a aac -b:a 128k output.mp4'. This Line convert .avi file to .mp4 mantaining quality and fps.
4. If you mount the camera rotated, you need to rotate the video with this line: ‘ffmpeg -i output.mp4 -vf "transpose=2" -c:a copy output\_r.mp4’
5. A test with a 3 minutes video, where manual labelling were compared with our CNN output reported a K-Cohen of 0.785. You can find the experiment on the repo.
6. If you want to test the sync with echo log file: save echo log file, save it in .csv format. Then pass it in downsampling.py and then in plotanimation.py (echo repo) and compare this video with the processed one.

(UPDATE! NOW THIS IS ALL INCLUDED IN PLATFORM APPLICATION REPO, SEE THE PLATFORM REPO TO SEE ALL SYNCHRONIZED DATA, on that code are included the downsapling to 20 Hz of the echo log test and the data plotting)

1. In order to save processed video run (ALSO FOR THE PLATFORM):

‘sudo chmod 777 folder\_name’

‘sudo chown yourname:yourname folder\_name’

when in the gui\_eye\_contact dir. Then open both video on kdenlive. OR LOAD ON THE PLATFORM APPLICATION.

# How to record videos

There are three ways to record and save video:

1. Via smartphone: Download nRF Connect, Connect to ‘CameraModule’ and send these UTF8 string to respectively start and stop recordings: ‘rec’, ‘stp’.
2. Echo or IM-TWIN Android App: After connecting the TWC device, click on ‘Camera’ button. Once the device is connected, this button will become green. Then use the Rec button to start/stop a video. If camera is connected, Start log button will simultaneosly start the recording of the toy data and the video recording from the Camera Glasses. The camera will blink orange while recording. Use this as debug.

# Charging the Device

In order to charge the device, connect a usb-c cable to the device and turn it ON!. When charging, inside the cable hole you can see a red led blinking. When the device is charged red light will turn off. (Orange Led turns on with the device supply.)

# 8. Device Updates

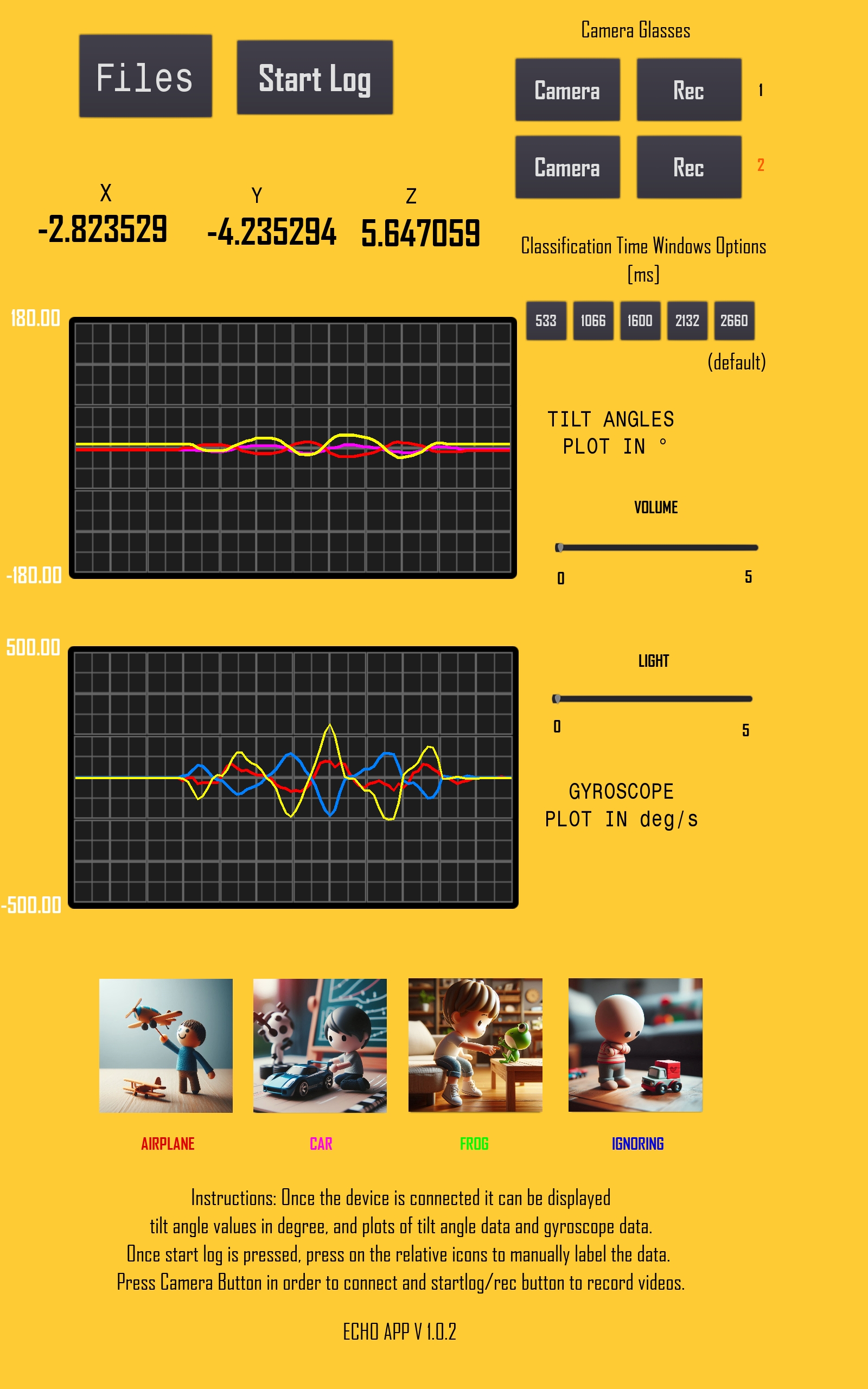
Since the Wifi connection shows a lot of problems, now device works with only BT functionalities, and you can also read the battery voltage from the command characteristic sended by the device every 10 seconds! (use nRf connect and set UTF-8 datatype). For doing this you need to connect d4 pin 1) to gnd with a 100k resistor; 2) to battery+ with a 100k resistor. (divider by 2 partitor).

# 9. Platform instructions

1) MOUNT CAMERA GLASSES & ECHO. LOAD THE FIRMWARE, AND INSTALL ECHO APP.

2) TURN ON ECHO (you will see a red light), AND THE GLASSESS (orange led on).

3) CONNECT ECHO AND THE GUI WILL OPEN.



4) click on camera to connect the camera glasses (repeat two times for the 2 glasses) once both are connected both camera button will become green.

5) Start LOG will start the recording of all the data (both echo and glasses). Use the orange led blink ad debug (it will blink only during the recording).

6) for new recording and to have the correct time stamp, close and reopen the app repeating the previous points.

7) save echo log file, by google drive(save it on csv). Take the SD and save the 2 videos.

8) Save the files on the computer and run this command on the same directory: 'ffmpeg -i input.avi -c:v libx264 -crf 23 -preset medium -c:a aac -b:a 128k output.mp4'. This Line convert .avi file to .mp4 mantaining quality and fps.

9) if necessary rotate the video: If you mount the camera rotated, you need to rotate the video with this line: ‘ffmpeg -i output.mp4 -vf "transpose=2" -c:a copy output\_r.mp4’

10) process the video con computer vision repository, go to the output folder and give permissions to save them:

‘sudo chmod 777 folder\_name’

‘sudo chown yourname:yourname folder\_name’

when in the gui\_eye\_contact dir.

11)Now you have the processed video, the second video, echo log file, and computer vision ouput (save it in .csv!). Load all on the platform application (just run it with a pyhton compiler.