DIY Neuroscience Kit Basic

Upside Down Labs

Apr 02, 2026

CONTENTS

CHAPTER

ONE

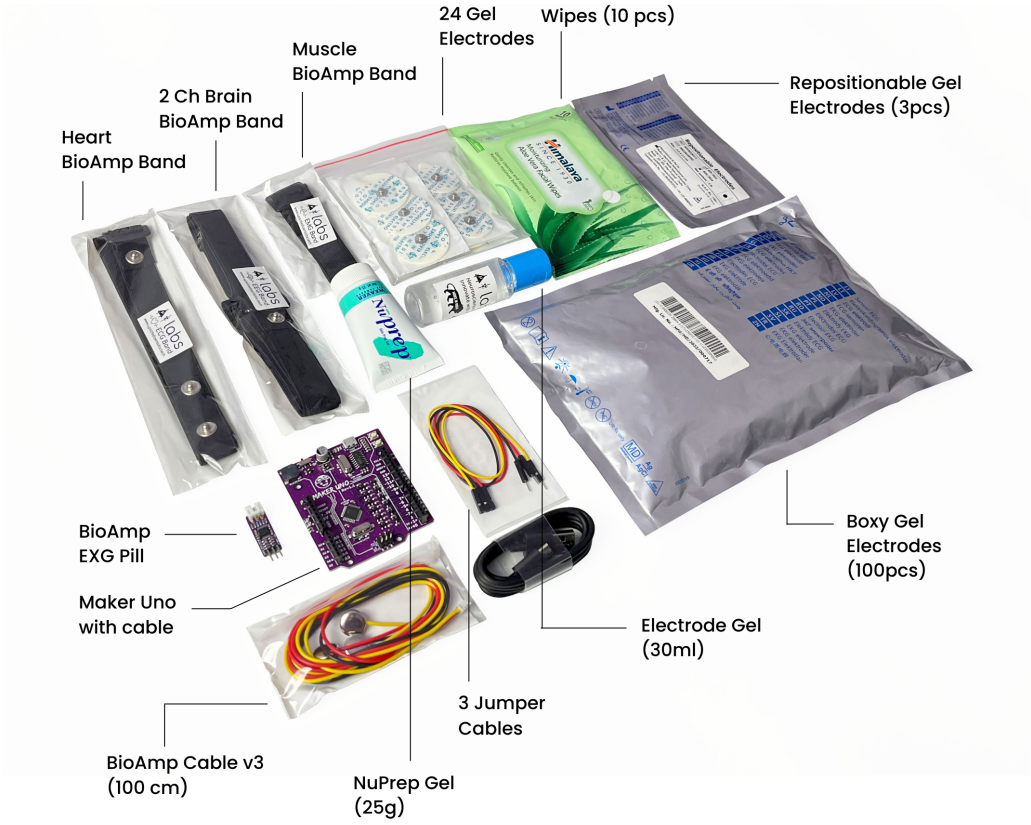
DIY NEUROSCIENCE KIT BASIC

1.1 Overview

This kit is perfect for `students`, `researchers` and `hobbyists` alike who want to start exploring neuroscience. Whether it's studying brain waves, monitoring heart rhythms, analyzing muscle movements, or tracking eye movements, this DIY kit provides an accessible and educational platform for understanding the complexities of human physiology and developing practical applications in the fields of human-computer interaction, and beyond.

1.2 Contents of the kit

From development board (Maker UNO), BioAmp EXG Pill, BioAmp cable v3, jumper cables, gel electrodes, dry electrode-based BioAmp bands to skin preparation kit, this includes everything that you need to get started with your awesome HCI/BCI project.



DIY Neuroscience Kit – Basic

Unboxing the kit:

https://youtu.be/7O9Bw8y5fQs

1.3 Software requirements

To use your DIY Neuroscience Kit Basic, you will need the softwares mentioned below. Instructions on how to use them are provided later in the guide.

- Arduino IDE (Download this to upload Chords arduino firmware to your development board)
- Chords Web (Use this open-source web application to visualize your biopotential signals)
- Visual Studio Code (or any other Code editor of your choice)
- Python (To run Chords-Python script)
- Chords Python (Use this open-source python script designed to record and visualize biopotential signals)

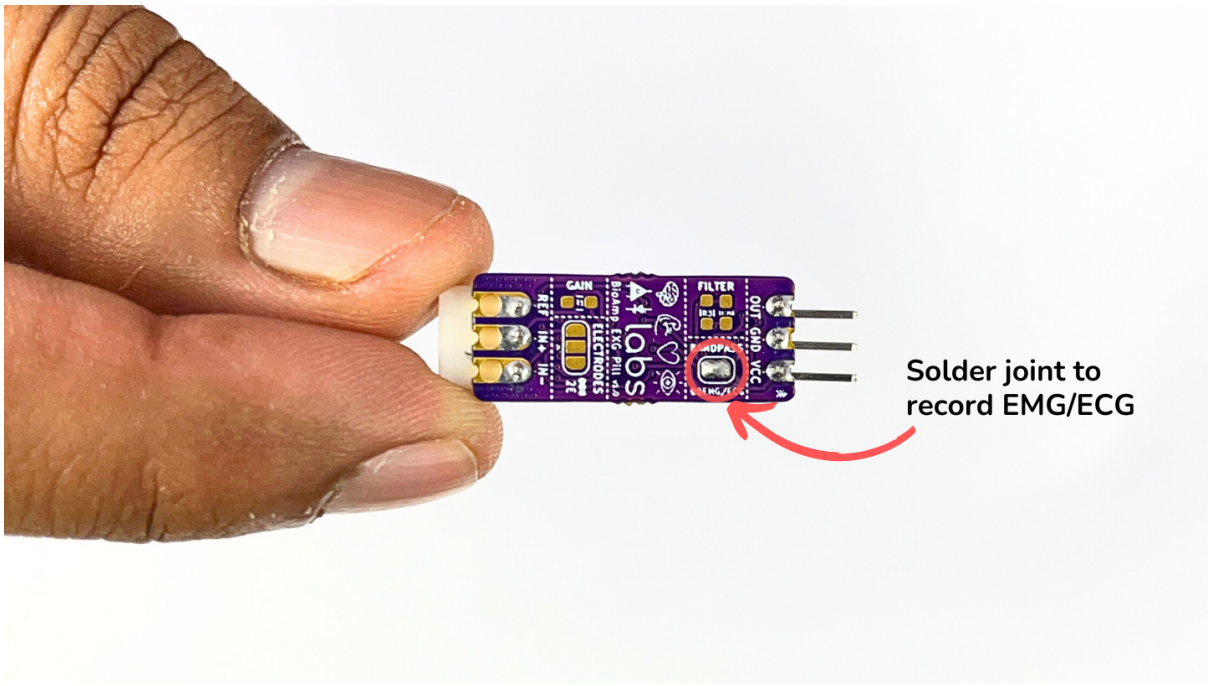
Note

1. The Chords Arduino firmware is identical for both Chords Web and Chords Python, so you only need to upload the code once, and you're all set.
2. You can choose either Chords Web or Chords Python to record and visualize your biopotential signals based on your needs. If you're curious, you can try both one at a time.

1.4 Using the kit

This kit is made in a way so that even beginners can use it and get started with recording biopotential signals from their body to explore the field of neuroscience by making HCI/BCI projects.

1.4.1 Step 1 (optional): Configure for EMG/ECG

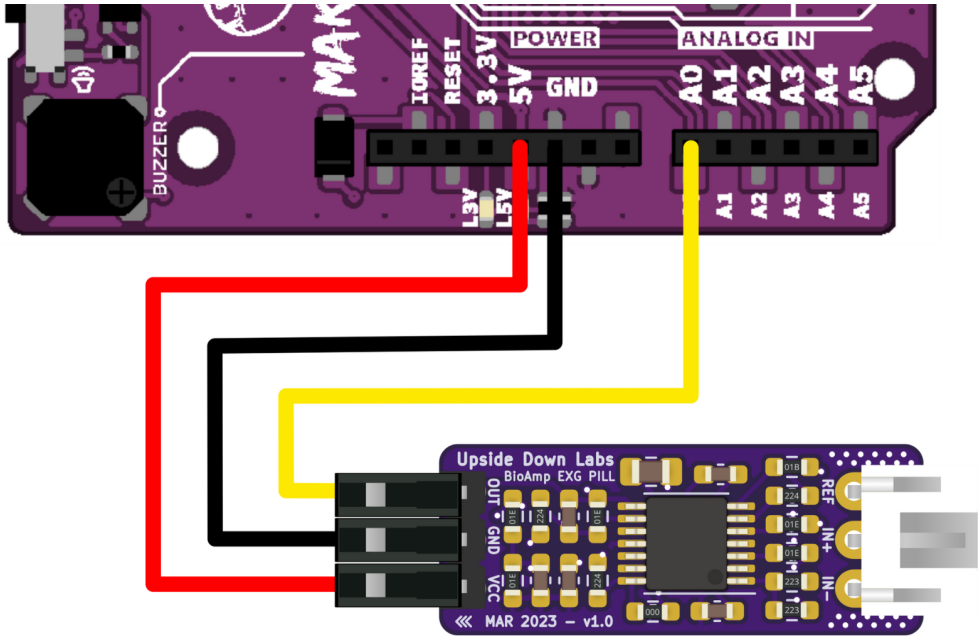


BioAmp EXG Pill is by default configured for recording EEG or EOG, so if you are recording any of the two signals you can skip this step. But if you want to record good quality ECG or EMG, then it is recommended to configure it by making a solder joint as shown in the image above.

Note

Even without making the solder joint the BioAmp EXG Pill is capable of recording ECG or EMG as well but the signals would be more accurate if you configure it.

1.4.2 Step 2: Connect Maker UNO



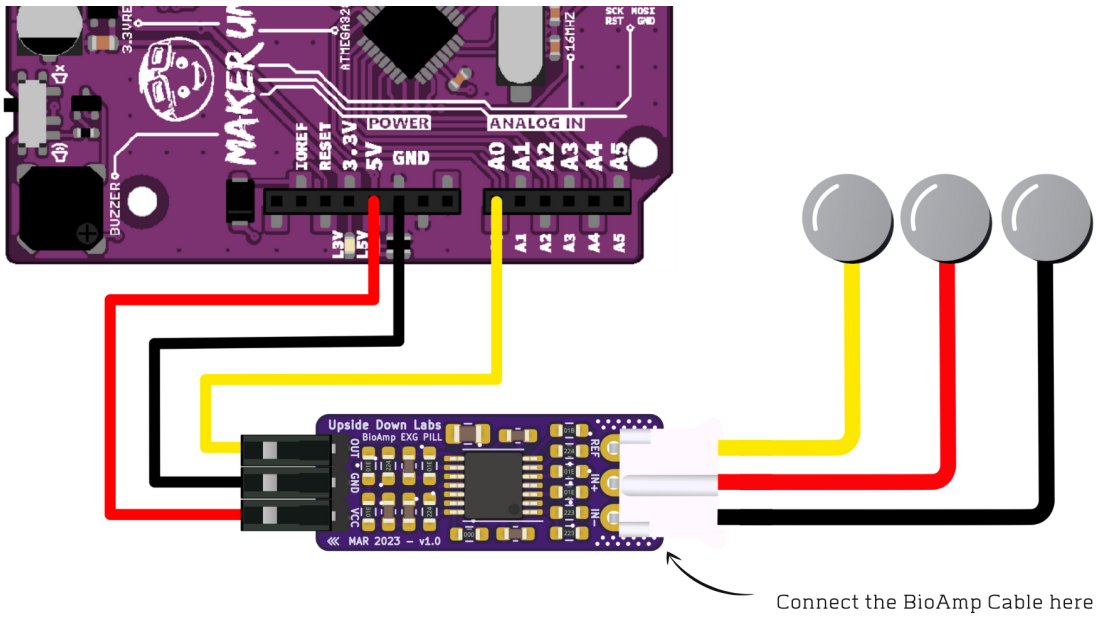
Connect `VCC` to `5V`, `GND` to `GND`, and `OUT` to `Analog pin A0` of your Maker UNO via jumper cables provided by us. If you are connecting OUT to any other analog pin, then you will have to change the INPUT PIN in the arduino sketch accordingly.

Warning

Take precautions while connecting to power, if power pins (GND & VCC) are to be swapped, your BioAmp EXG Pill will be fried and it'll become unusable (DIE).

1.4.3 Step 3: Connecting electrode cable

Connect the BioAmp cable to BioAmp EXG Pill by inserting the cable end in the JST PH connector as shown above.



1.4.4 Step 4: Skin Preparation

Apply Nuprep Skin Preparation Gel on the skin surface where electrodes would be placed to remove dead skin cells and clean the skin from dirt. After rubbing the skin surface thoroughly, clean it with an alcohol wipe or a wet wipe.

For more information, please check out detailed step by step *Skin Preparation Guide*.

1.4.5 Step 5: Electrode placement

We have 2 options to measure the signals, either using the gel electrodes or using dry electrode based BioAmp Bands. You can try both of them one by one.

1. *Using Gel electrodes guide*
2. *Assembling & using BioAmp Bands guide*

Once you have made the connections, return here to proceed to the next steps.

1.4.6 Step 6: Uploading the code

1. Connect the Maker Uno to your laptop using the USB cable (Type A to Type B). Go to Chords Arduino Firmware github repository, open `AVR-NANO-UNO-MEGA` folder and copy paste the arduino sketch in Arduino IDE that you downloaded earlier.

   Link for the arduino sketch: Chords Arduino Firmware for Maker Uno

2. Uncomment `#define BOARD_MAKER_UNO` in the code.
3. Go to `tools > board > Arduino AVR boards` and select Arduino UNO. In the same menu, select the COM port on which your Maker Uno is connected. To find out the right COM port, disconnect your Maker UNO board and reopen the menu. The entry that disappears should be the right COM port. Now click on the upload button.

Warning

Make sure your laptop is not connected to a charger and sit 5m away from any AC appliances for best signal acquisition.

1.4.7 Step 7: Setting up Chords Web

1. Visit chords.upsidedownlabs.tech

1.4.8 Step 8: Setting up Chords Python

Since you have uploaded the firmware already to your Maker UNO, use our python script and follow the steps given in the *Chords-Python documentation* for lsl streaming, CSV data logging, verbose output with detailed statistics and error reporting. Not only that, you get a complete web interface to access various applications (like ECG with heart rate, EMG with envelope, GUI of channels, CSV plotter, etc.) that you can use to further analyse your signals and create HCI/BCI projects.

1.5 Some Project Ideas

You can find step-by-step tutorials for various HCI/BCI projects on our Instructables.

Here are some project ideas that you can try making at your home. Click on the links below to get the step by step guides to build the projects.

1. Controlling video game using EEG signals
2. Recording EEG from visual cortex
3. Record publication-grade ECG signals
4. Measuring heart rate
5. Detecting heart beats
6. Creating a drowsiness detector
7. Detecting eye blinks
8. Detecting up and down movement of eyes
9. Recording publication-grade EMG signals

These are some of the project ideas but the possibilities are endless. So create your own Human Computer Interface (HCI) and Brain Computer Interface (BCI) projects and share them with us at contact@upsidedownlabs.tech

CHAPTER

TWO

SKIN PREPARATION GUIDE

2.1 Why skin preparation is important?

Proper skin preparation is crucial before recording any biopotential signal be it Electrocardiography (ECG), Electromyography (EMG), Electroencephalography (EEG), or Electrooculography (EOG).

- `Clean skin surface:` Removes dead skin cells, oils, & other substances that increases skin impedance.
- `Improve impedance:` Improves the conduction of electrical signals from the body to the recording equipment and minimizes impedance.
- `Electrode-skin contact:` Ensures optimal contact between the electrodes and the skin surface.
- `Signal quality:` Enhances the overall quality of recorded signals, providing clear & reliable data for analysis & improves the ability to capture subtle variations in biopotential signals.
- `Consistency in recordings:` Reduces variability in signal quality, making it easier to make any Human-Computer Interface (HCI), Brain-Computer Interface (BCI) project or a real-world application.
- `Long term adhesion:` Facilitates long-term adhesion & stable placement of electrodes to the skin during extended signal monitoring.

2.2 Kit Contents

Nuprep gel	Mildly abrasive, highly conductive gel that should be applied before placing the electrodes on the skin to improve signal quality & enhances the performance of monitoring electrodes.
Electrode Gel	Highly conductive gel that acts as a coupling agent between dry electrodes and the skin to aid the transmission of biopotential signals like ECG, EMG, EOG, or EEG.
Ten20 paste	Contains the right balance of adhesiveness and conductivity, enabling the dry electrodes to remain in place while allowing the transmittance of biopotential signals.
Alcohol Swabs/Wet wipes	Soft & non-woven pads that helps in cleaning the skin surface and does not leave any residue.
Cotton Swabs	Useful while applying nuprep gel or ten20 paste.

2.3 Steps to follow

You can follow the steps given below to do the skin preparation properly:

2.3.1 Step 1: Identify the targeted area

Identify the target area where the gel electrodes or BioAmp Bands will be placed for recording the biopotential signals.

2.3.2 Step 2: Apply NuPrep gel

Take a small amount of NuPrep gel using a cotton swab and apply it on your targeted area.

2.3.3 Step 3: Clean the skin surface

Use gentle, circular motions to rub the gel on the skin surface. This removes all the dead skin cells & improves conductivity.

Warning

Do not rub the gel for too long as it has abrasive properties and may cause skin redness and irritation.

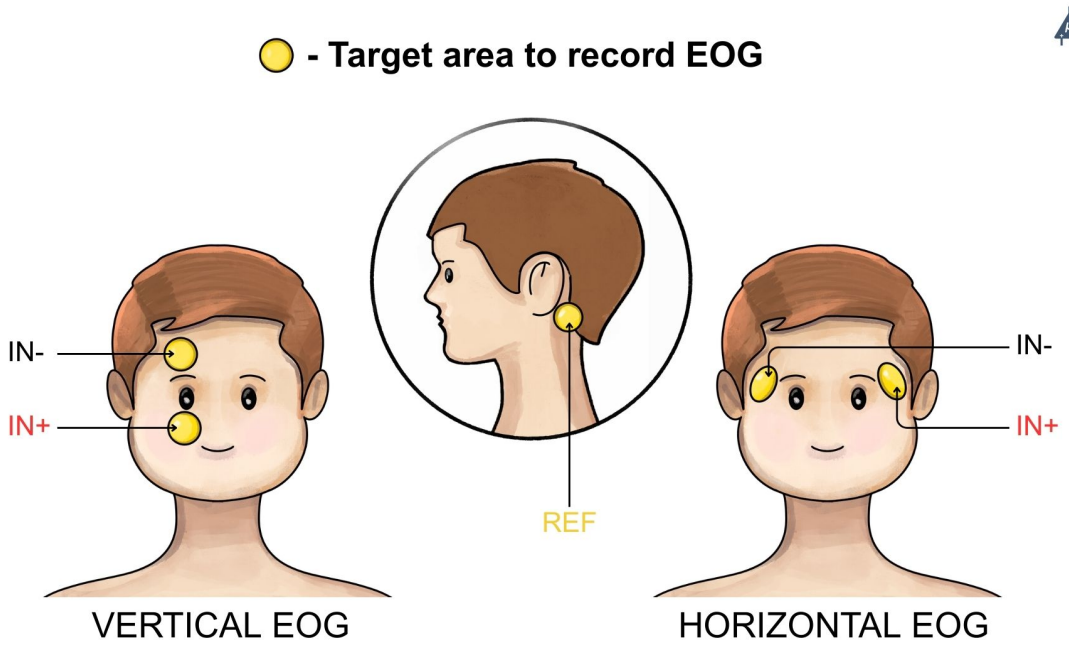


Fig. 1: Target area to record EOG

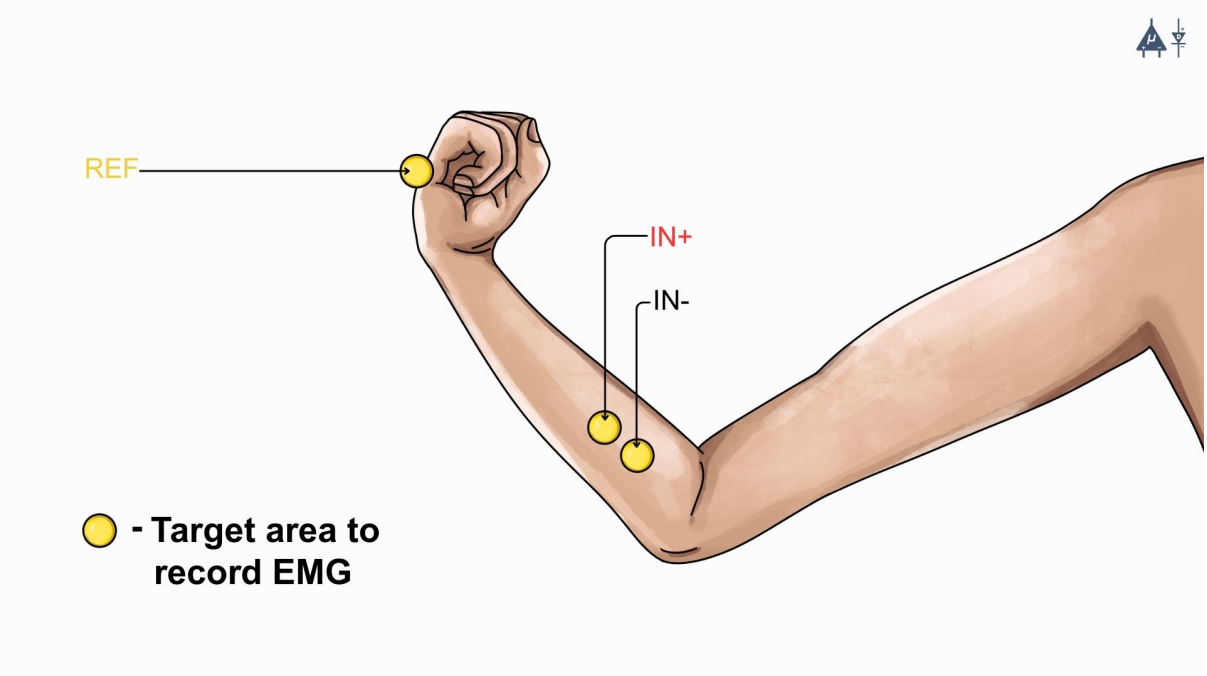


Fig. 2: Target area to record EMG

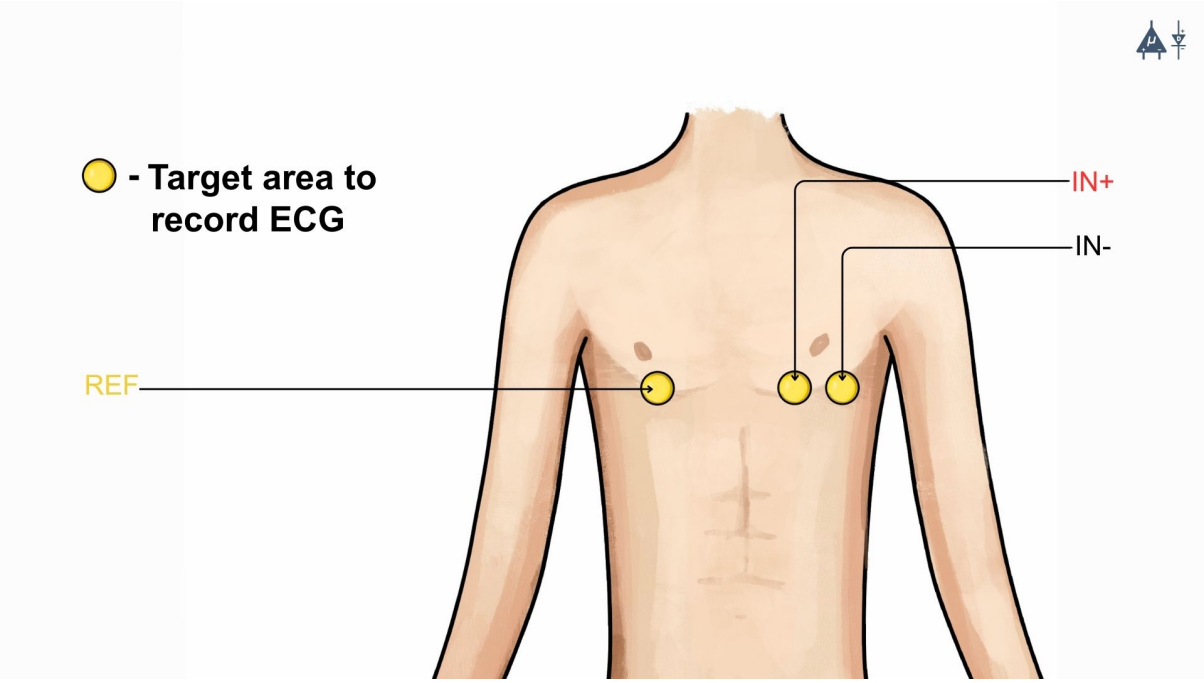


Fig. 3: Target area to record ECG

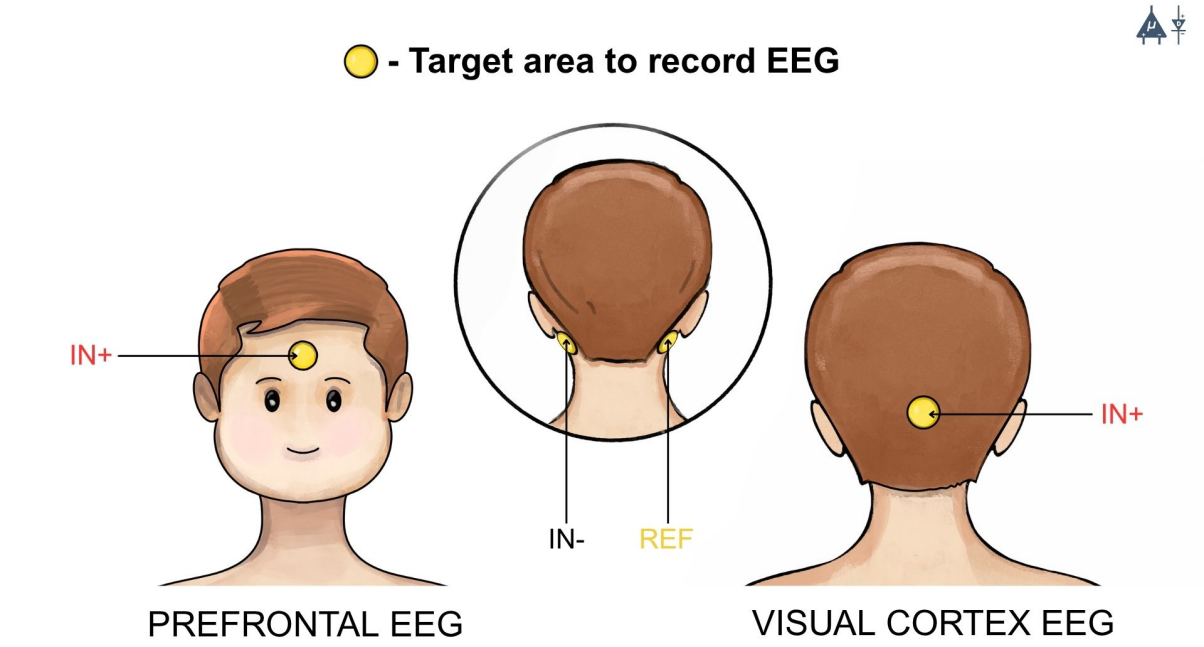


Fig. 4: Target area to record EEG

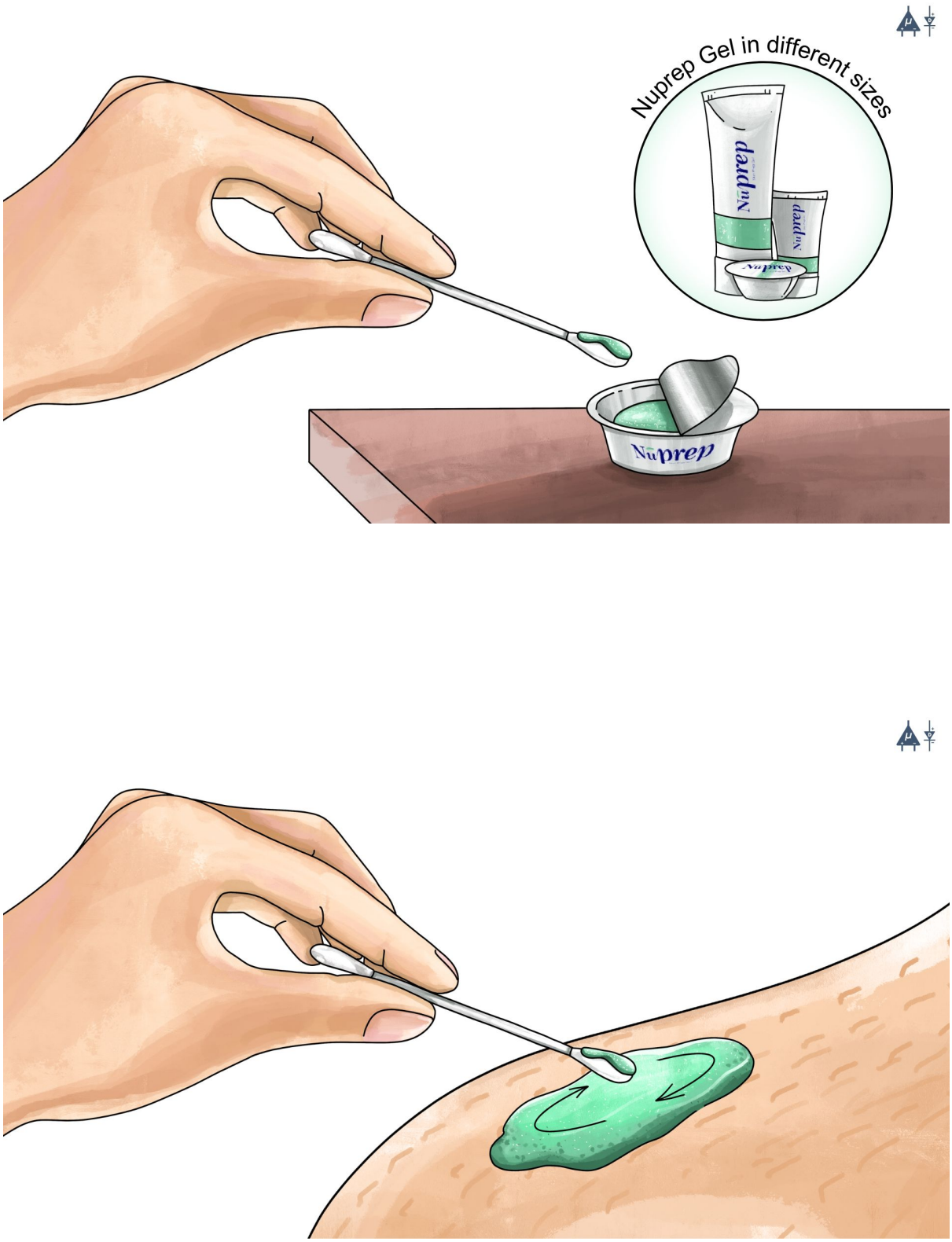


Fig. 5: Rub the gel gently using the cotton swab

2.3.4 Step 4: Wipe off the gel

Wipe away excess gel with alcohol swabs or wet wipes.

Fig. 6: Wipe away excess gel

Warning

- Using alcohol swabs can dry out the skin, so don't use them if your skin is already dry.
- Close your eyes while using the alcohol swabs for EOG recording else it may cause eye redness & irritation.

2.3.5 Step 5: Measuring the signals

Now you can either use gel electrodes or BioAmp bands for the signal recording.

Using gel electrodes

Connect the BioAmp cable to gel electrodes, peel the plastic backing from electrodes and place the IN+, IN-, REF cables according to your specific biopotential recording.

Note

While placing the gel electrodes on the skin, make sure to place the non-sticky tab of the electrode in the direction opposite to your hair growth. This allows you to remove the electrodes easily without pulling off much body hair.

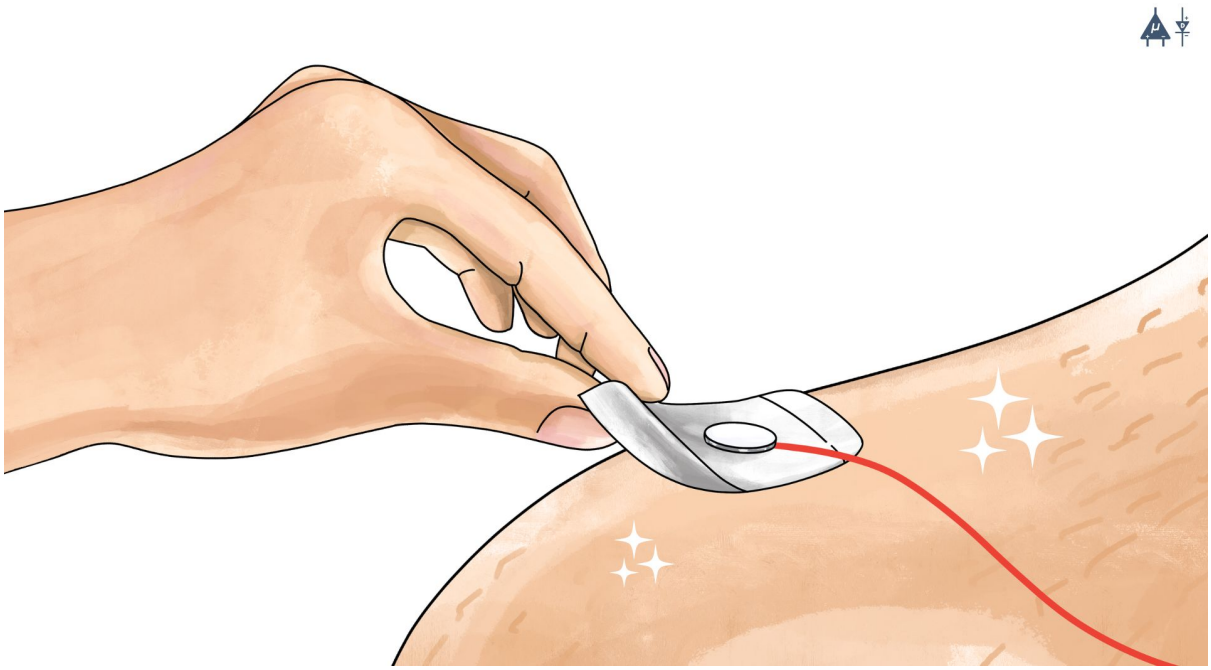


Fig. 7: Placing gel electrodes on skin surface

Using BioAmp bands

Connect the BioAmp cable to your BioAmp band. Now apply a small amount of `electrode gel` or `Ten20 conductive paste` on the dry electrodes between the skin and metallic part of BioAmp cable. This improves the signal conductivity, enhancing overall signal quality.

Note

The above graphics demonstrates the use of electrode gel/Ten20 paste with Muscle BioAmp Band. Similarly you can use Brain BioAmp Band and Heart BioAmp Band. Refer to *Using BioAmp Bands* guide to assemble and use all the BioAmp Bands correctly.

Now you are all set! Make all the connections correctly and start recording your biopotential signals.

Warning

NuPrep gel, Ten20 paste and the alcohol swabs shouldn't be used if you have a history of skin allergies to lotions and cosmetics.

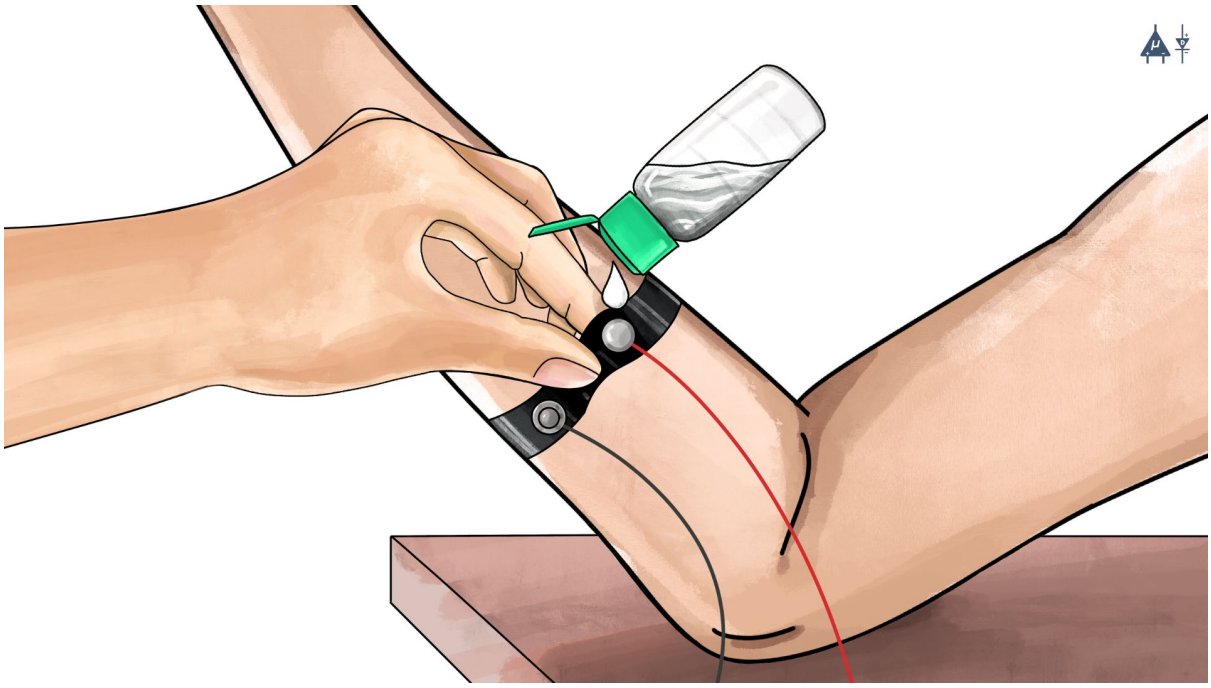


Fig. 8: Method 1: Using Electrode gel

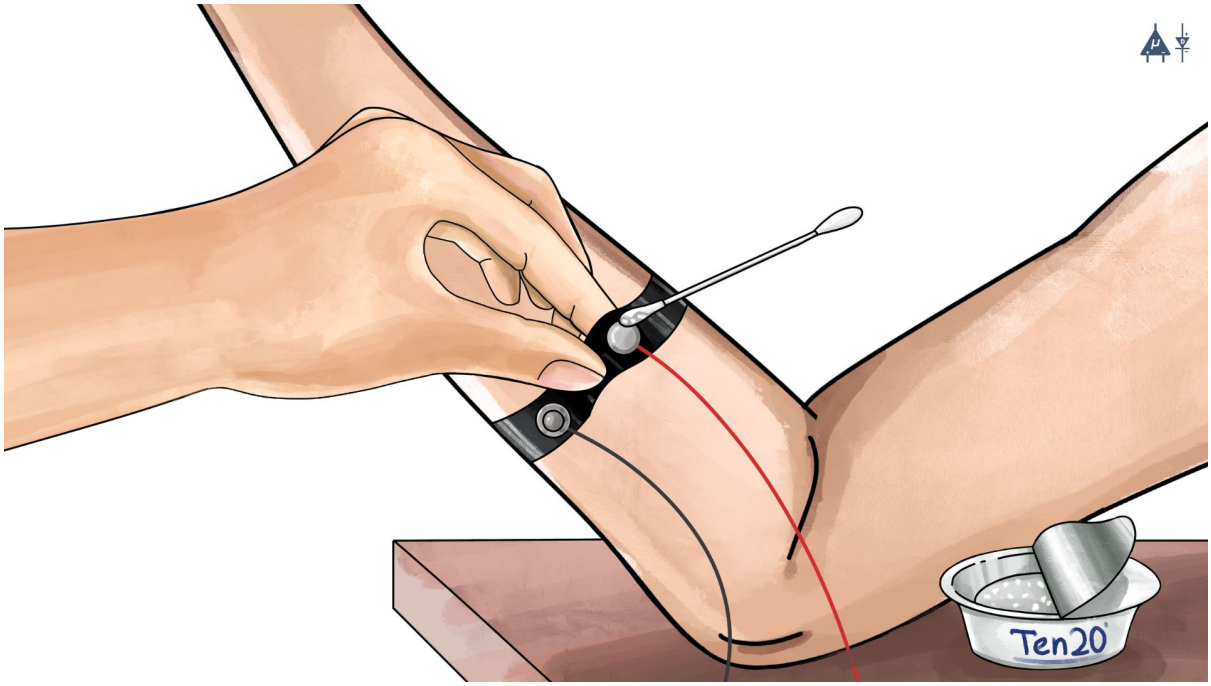


Fig. 9: Method 2: Using Ten20 paste

CHAPTER

THREE

USING GEL ELECTRODES

3.1 Overview

Gel electrodes are type of electrodes that are applied on the skin surface to transmit the electrical signals present at the skin surface to a recording or monitoring device. These electrodes use a conductive hydrogel to improve the interface between the Ag-AgCl electrode and the skin, enhancing signal quality by reducing impedance and improving electrical contact. These are often used to record biopotential signals like Electrocardiography (ECG), Electroencephalography (EEG), Electromyography (EMG), and Electrooculography (EOG).

3.2 Key components in a gel electrode

`Electrode material`: The electrode itself is typically made of a conductive material such as Ag-AgCl, which is commonly used due to its stability and biocompatibility.

`Hydrogel`: A hydrogel is applied to the electrode or embedded within it. This gel helps maintain a consistent electrical connection with the skin by reducing the impedance at the interface.

`Backing`: Gel electrodes have various types of backing like foam, cloth, and cleartape.

`Adhesive`: Gel electrodes has a sticky glue like interface called adhesive that allows it to firmly attach to your skin, it can be of different types like hydrogel (where hydrogel acts both as a conductive gel and adhesive), standard, and aggressive.

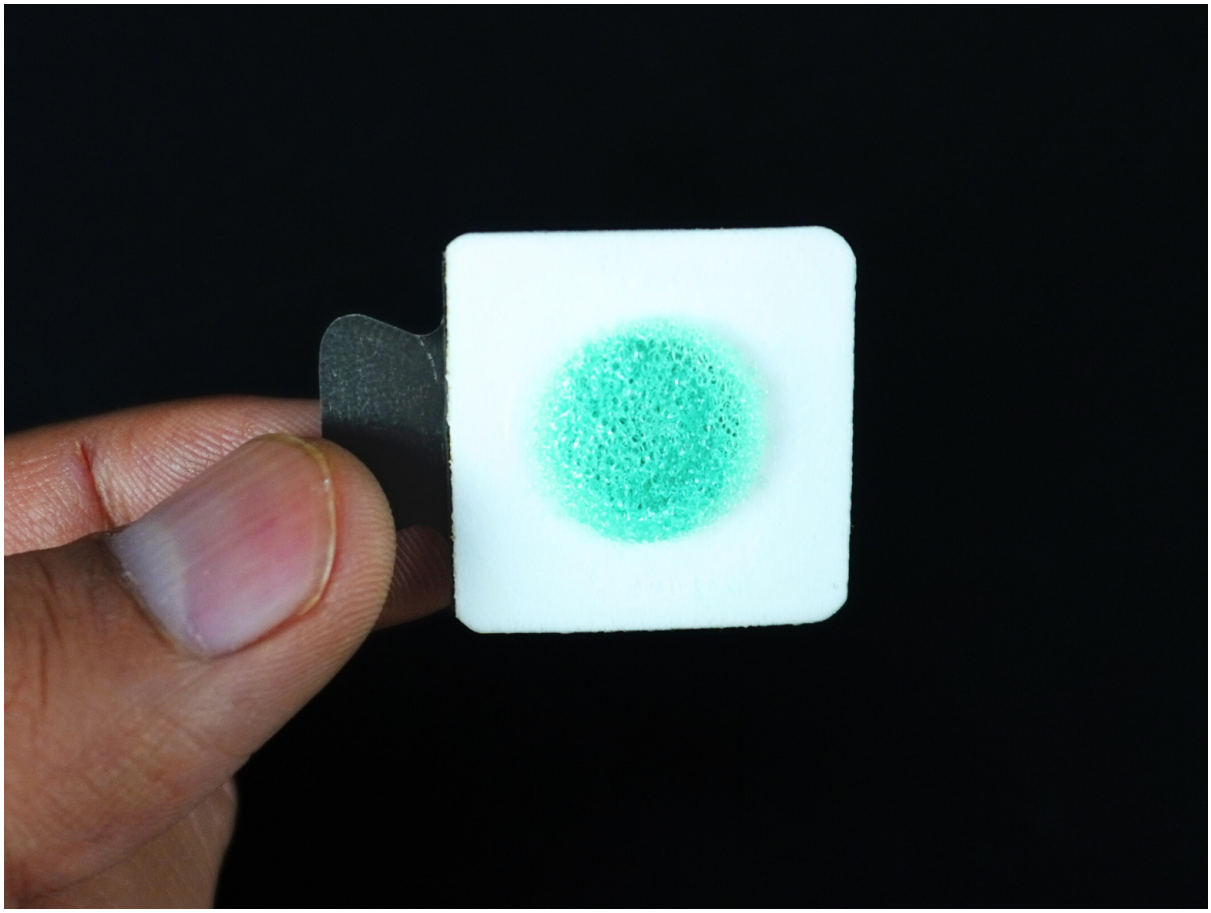
3.3 Types of gel electrodes

3.3.1 1. Classification on basis of water-content

Below are the gel electrodes categorised on the basis of the water content present in the hydrogel:

a. Liquid gel electrodes

These electrodes have hydrogel with higher water content which provide excellent conductivity but can dry out over time, reducing their effectiveness. They are preferred to be used if you have a dry skin and/or you want to record the biopotential data for a short time duration.



b. Solid gel electrodes

These electrodes contain hydrogel that is more solid or jelly-like. They are less prone to drying out as compared to liquid gel electrodes and have a strong adhesion.

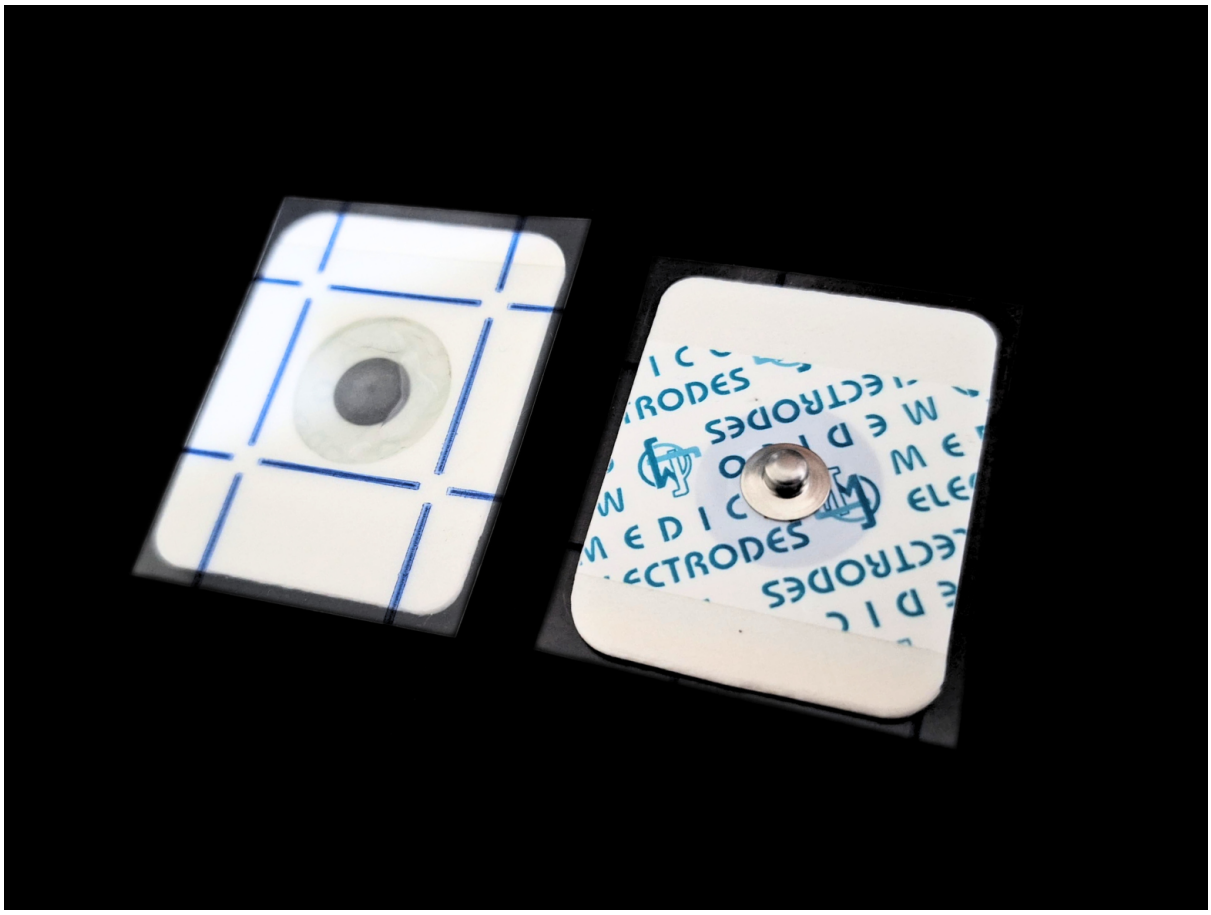
c. Full-surface hydrogel electrodes

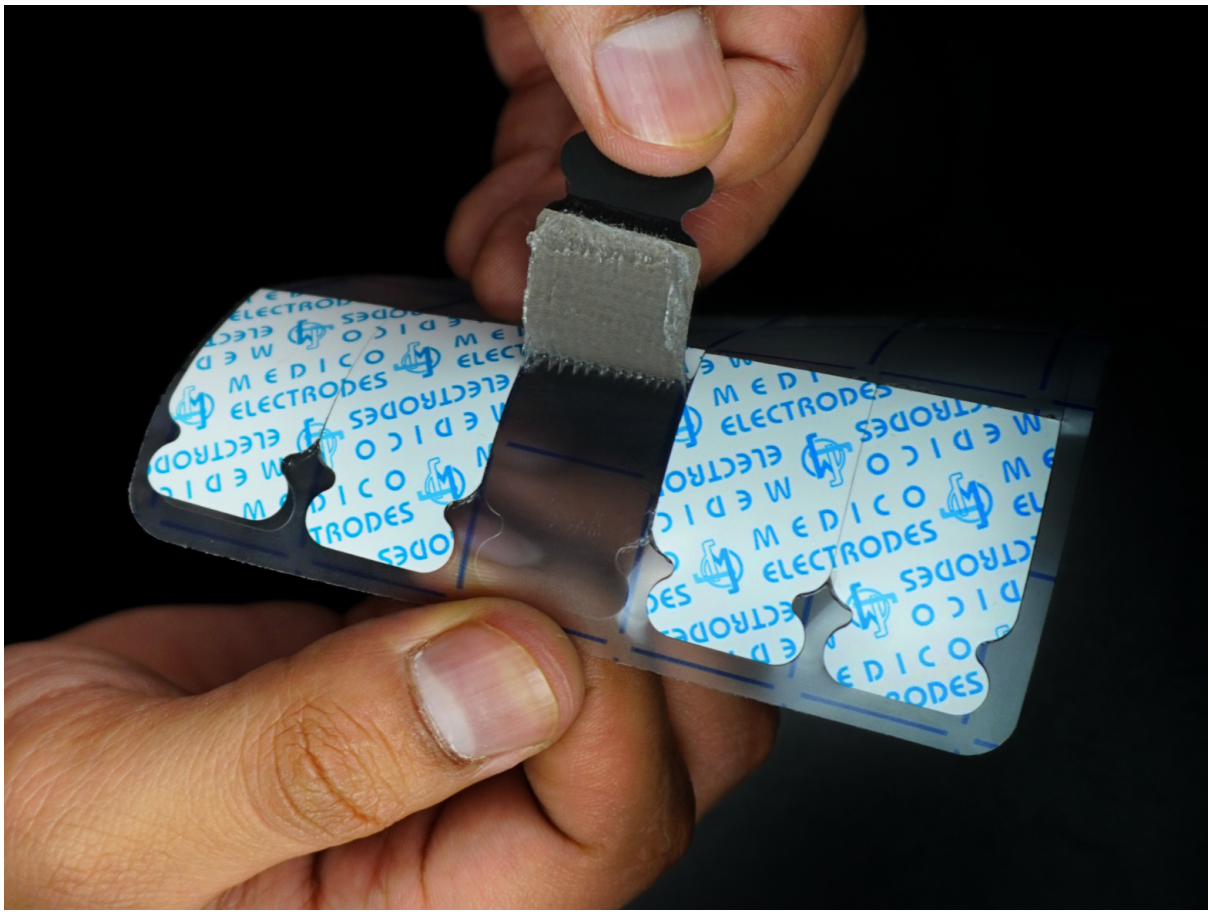
This is a type of solid gel electodes where hydrogel has minimal water content, covers the entire electode surface and acts as an adhesive. These electrodes are more comfortable for long-term monitoring. They offer good adhesion, conductivity and are less likely to cause skin irritation.

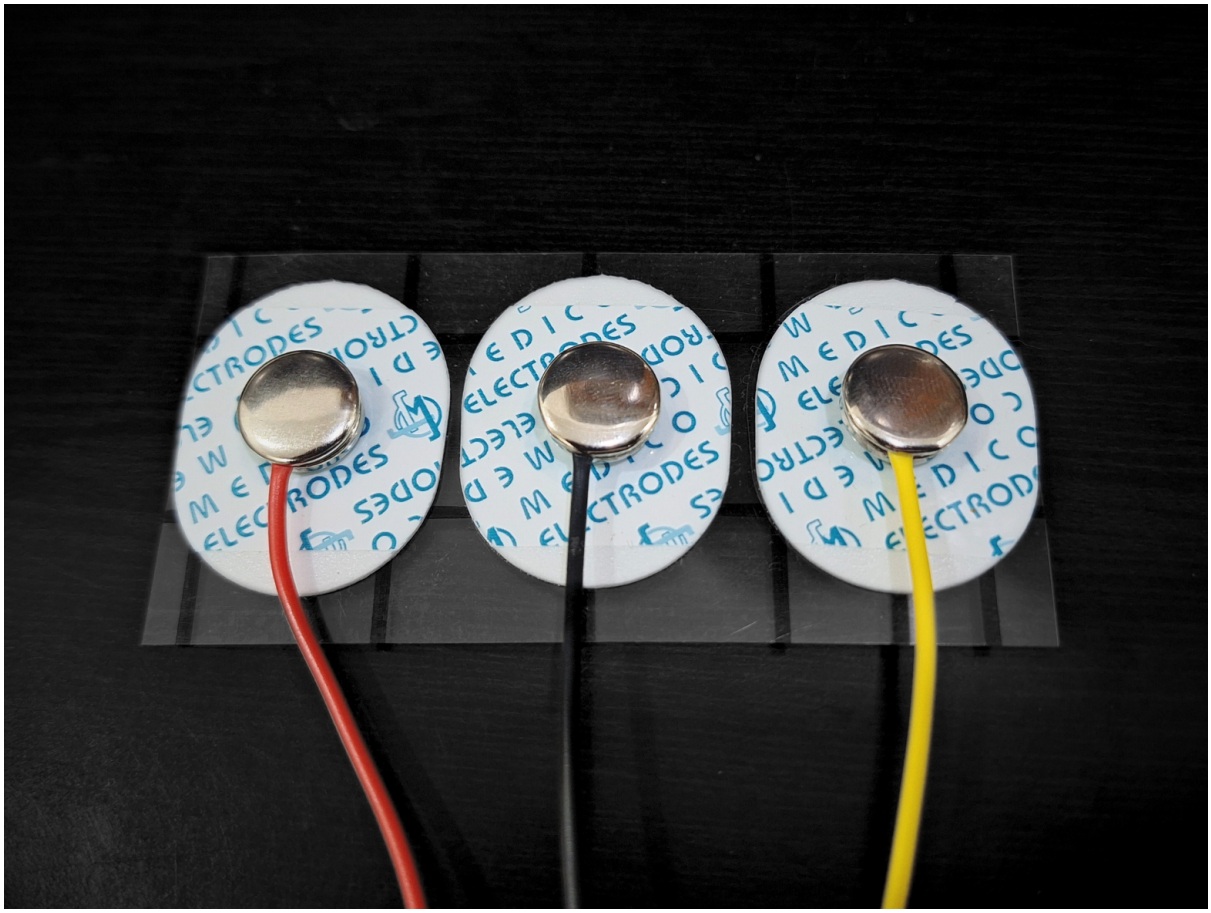
3.3.2 2. Classification on the basis of connectivity

a. Connecting via snap

Some electrodes have a metallic snap connector on them which enables you to connect BioAmp Snap Cable by snapping the dry electrode part of the cable onto the electrode.




MEDICO
ELECTRODES



b. Connecting via tab

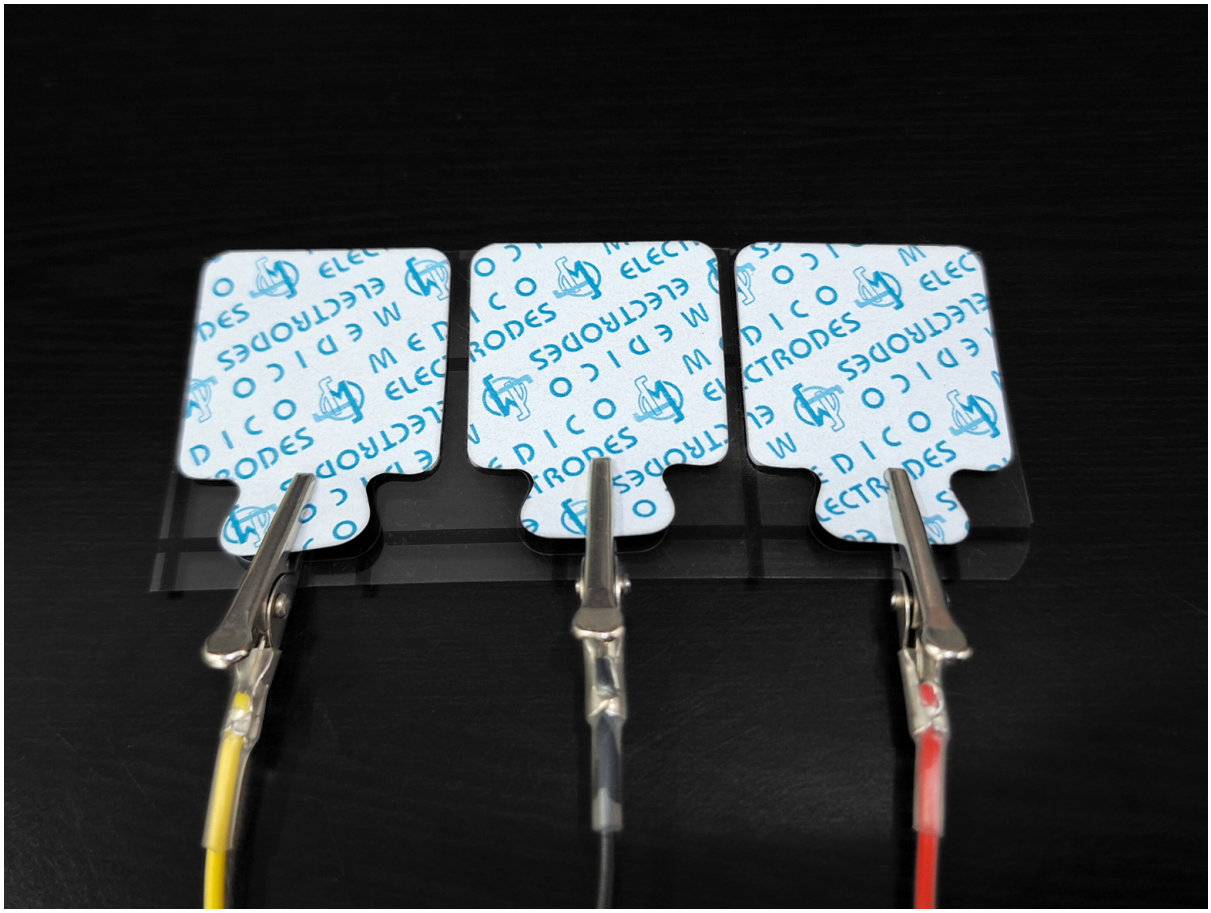
Some electrodes do not have the metallic snap connector on them, instead they have a conductive tab extending from the surface of the electrode. You can directly use BioAmp Alligator Cable to connect to the tab.

3.4 Using the electrodes

Determine the target area from where you want to record the biopotential signals.

Warning

For people having sensitive skin, it is recommended to use either gel electrodes with hydrogel/standard adhesive or *use BioAmp Bands*.



3.4.1 1. Skin Preparation

Remove any excessive hair on the targeted area. Apply Nuprep skin preparation gel on the skin surface where electrodes would be placed to remove dead skin cells and clean the skin from dirt. After rubbing the skin surface thoroughly, clean it with an alcohol wipe or a wet wipe.

For more information, please check out detailed step by step *Skin Preparation Guide*.

Note

Always ensure that the prepared skin area is dry prior to applying the gel electrodes.

3.4.2 2. Connecting the cable

Connect the BioAmp Snap/Alligator Cable on the gel electrodes.

3.4.3 3. Electrodes placement

Peel of the plastic backing and place the electrodes on your targeted skin surface as per the connections directed with every BioAmp hardware.

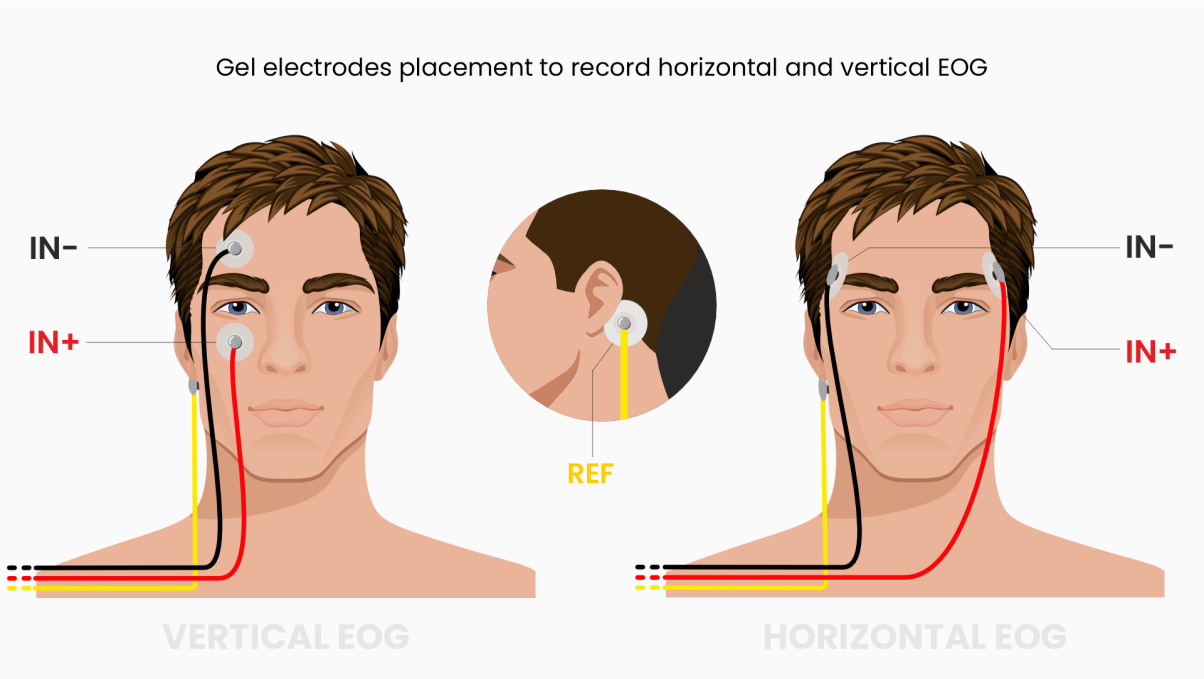


Fig. 1: Gel electrodes placement for EOG signal recording

Tip

While placing the gel electrodes on the skin, make sure to place the non-sticky tab of the electrode in the direction opposite to your hair growth. This allows you to remove the electrodes easily without pulling off much body hair.

3.5 Removing the electrodes

Once you are done with your signal recording, pull the non-sticky tab gently in the direction of your body hair growth to remove the gel electrode from the skin.

- To remove the gel residue with a wet wipe/alcohol swab.
- To remove the adhesive residue apply some oil to your skin and rub it gently for a couple of minutes to dissolve the adhesive, and then wipe it off with a dry cloth.

Important

You should not use water & soap to clean the adhesive residue, it is oil soluble and should be dissolved with oil before removing it with dry cloth. Using water and soap can make it stick harder and irritate your skin.

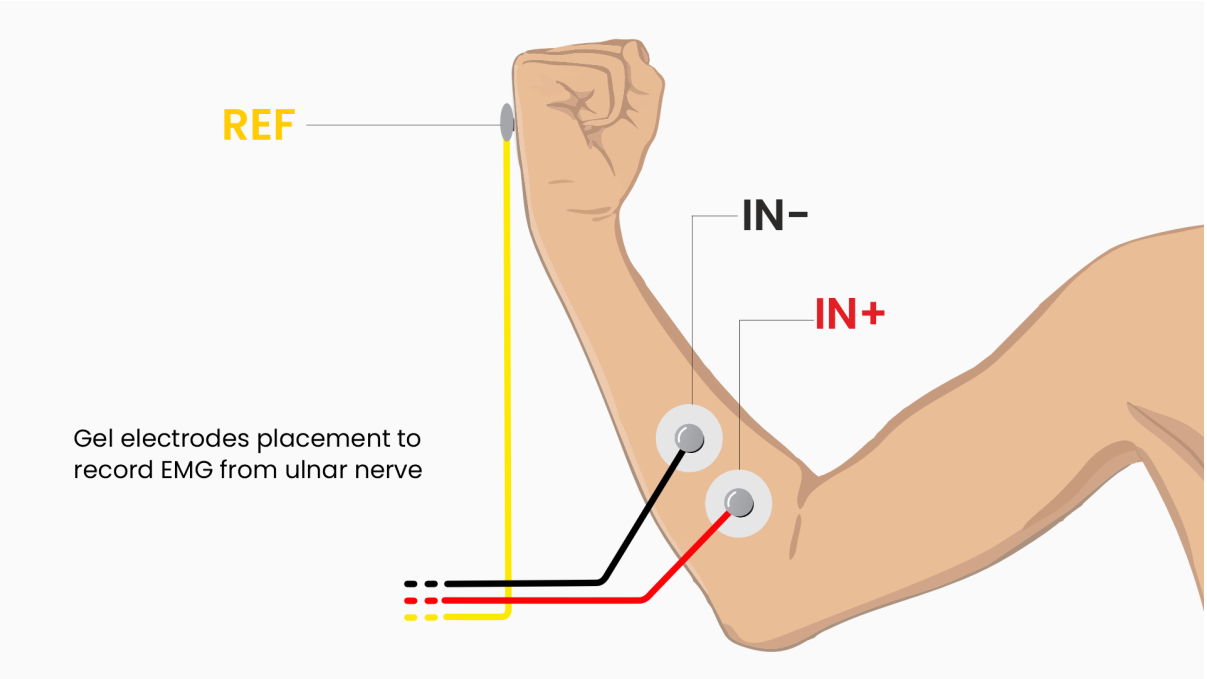


Fig. 2: Gel electrodes placement for EMG signal recording

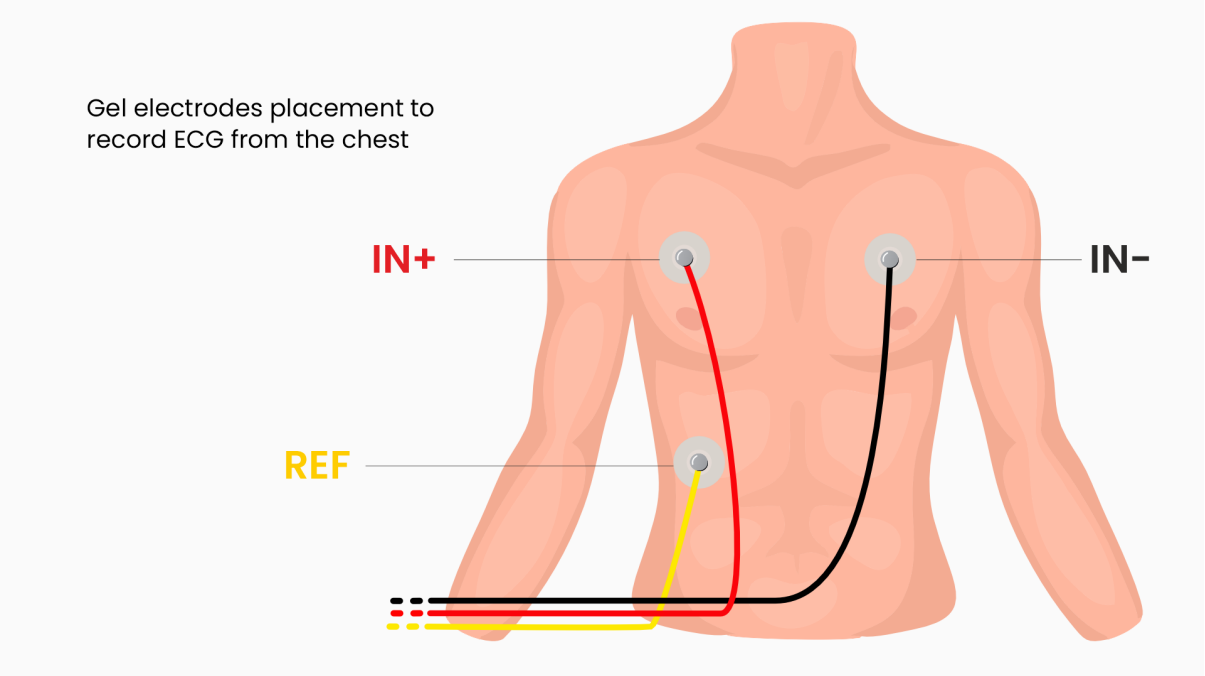


Fig. 3: Gel electrodes placement for ECG signal recording

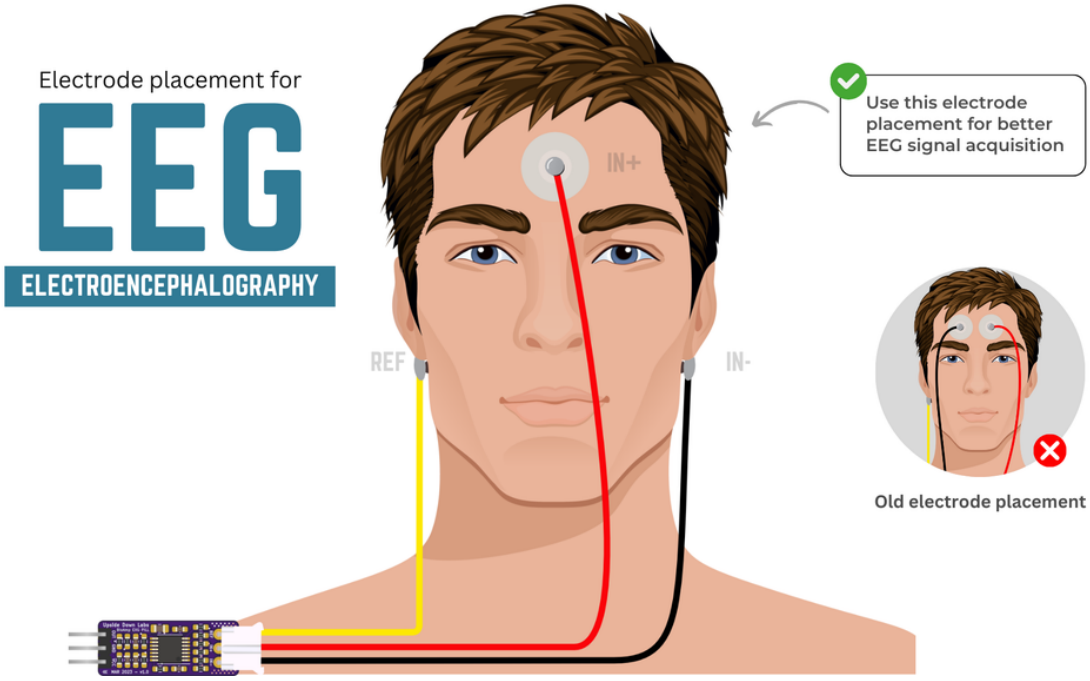


Fig. 4: Gel electrodes placement for EEG signal recording

CHAPTER

FOUR

USING BIOAMP BANDS

4.1 Overview

BioAmp Bands are dry electrode-based stretchable bands that allows you to record biopotential signals from your body be it from brain (EEG), muscles (EMG) or heart (ECG). These bands can only be used with our BioAmp Hardware by making the connections using BioAmp Cable.

4.2 Why use BioAmp Bands?

Usually, people use gel electrodes to record biopotential signals from the skin surface. But, it has its own disadvantages. So we came up with these BioAmp Bands using which users can enjoy a more comfortable, cost-effective, and hassle-free experience while recording biopotential signals.

- Comfort: BioAmp Bands are generally more comfortable to wear than gel electrodes, especially for long-term recordings. They conform to the body's shape and avoid the sticky, sometimes irritating sensation of gel electrodes.
- Reusability: Unlike gel electrodes, which are often single-use and need to be replaced frequently, BioAmp Bands can be reused multiple times. This makes them more cost-effective and environmentally friendly.
- Ease of Use: These bands are easy to wear and adjust, reducing the hassle of setup and ensuring consistent placement.
- Hygiene: They can be easily cleaned and sanitized between uses, reducing the risk of skin irritation and infections. Gel electrodes, on the other hand, can leave residue on the skin surface.
- Performance: The bands can provide stable and reliable signal recordings depending on your environment conditions. For hot/humid conditions, the bands usually perform better while recording the signals. But if the weather is cold causing dry skin, then it is recommended to prepare the skin properly and apply electrode gel between the metallic part of cable and skin surface. If you feel that the skin impedence is increasing, then reapply electrode gel frequently. The other option is to use gel electrodes after preparing the skin properly.

4.3 Types of BioAmp Bands

There are 3 types of BioAmp Bands and all these bands offer targeted and efficient solutions for recording biopotential signals from the muscles, heart, and brain, making them versatile tools for a wide range of HCI/BCI applications.

4.3.1 1. Muscle BioAmp Band

Muscle BioAmp Band (EMG Band) is a stretchable band that can be connected to any of our Muscle BioAmp Hardware or any EXG sensor using a BioAmp Cable. It allows you to record your muscle signals hassle-free.

Length	13 inches
Stretchability	2X (Upto 26 inches)
Usability	Reusable as it comes with washable fabric
Interface	Snap electrodes
Compatible Hardware	Muscle BioAmp Hardware or any EXG sensor
BioPotentials	EMG
No. of channels	1
Wearable	Yes

4.3.2 2. Heart BioAmp Band

Heart BioAmp Band (ECG Band) is a stretchable band that can be connected to any of our Heart BioAmp Hardware or any EXG sensor using BioAmp Cable. It allows you to record your ECG signals hassle-free.

Length	37 inches
Stretchability	2X (Upto 74 inches)
Usability	Reusable as it comes with washable fabric
Interface	Snap electrodes
Compatible Hardware	Heart BioAmp Hardware or any EXG sensor
BioPotentials	ECG
No. of channels	1
Wearable	Yes

4.3.3 3. Brain BioAmp Band

Brain BioAmp Band (EEG Band) is a stretchable band that can be connected to any of our Brain BioAmp Hardware or any EXG sensor using BioAmp Cable to record signals from the brain hassle-free.

Length	15.5 inches
Stretchability	2X (Upto 31 inches)
Usability	Reusable as it comes with washable fabric
Interface	Snap electrodes
Compatible Hardware	Brain BioAmp Hardware or any EXG sensor
BioPotentials	EEG
No. of channels	2 or 6
Wearable	Yes

You can get either a 2-channel or a 6-channel Brain BioAmp Band according to your project or research requirements:

2-Channel Brain BioAmp Band

It can be used to record EEG signals up to 2 channels either from the visual cortex (back of your head) or the prefrontal cortex part of brain.

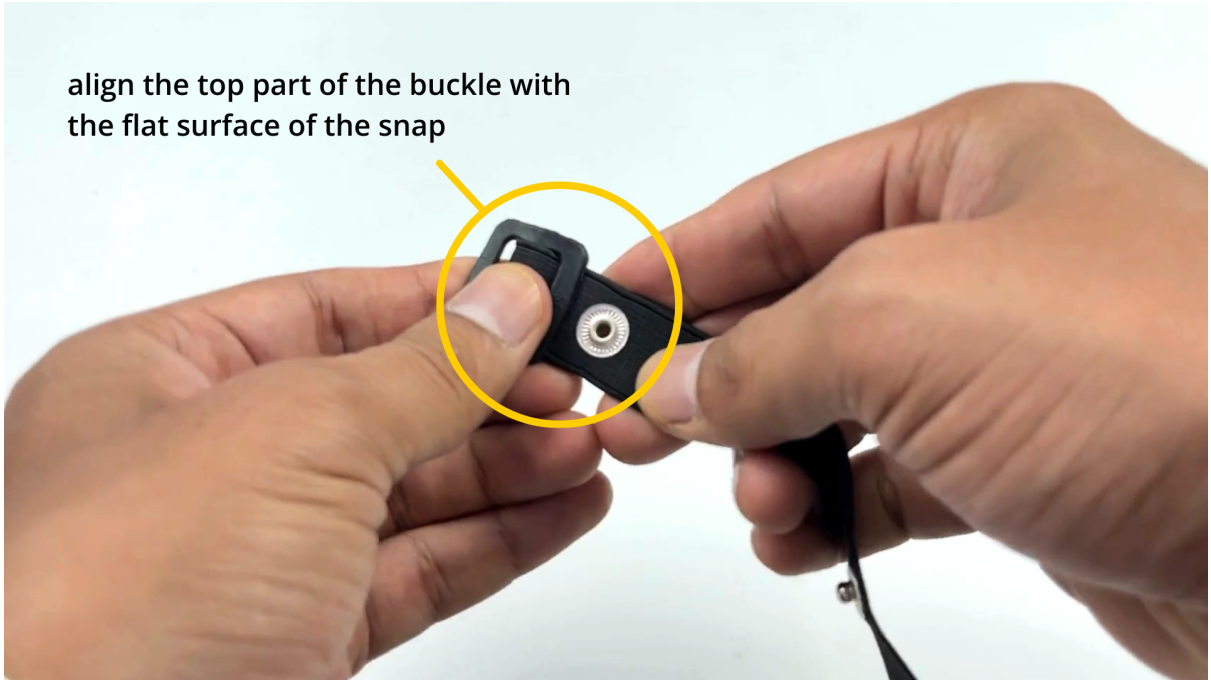
6-Channel Brain BioAmp Band

It can be used to record EEG signals up to 2 channels either from the visual cortex (back of your head) or the prefrontal cortex part of brain.

4.4 Using Muscle BioAmp Band

4.4.1 Assembly

1. Take your Muscle BioAmp Band, hold the side of the band that has buckle on it and align the top part of the buckle with the flat surface of the snap.



1. Take the other end of the band and insert it in the buckle.
3. Your band is now ready to use. You can also adjust the size of the band according to your targeted muscle.


Take other end of the band and
insert it in the buckle


Insert this end of the band into the buckle
as shown and pull it to finish the assembly

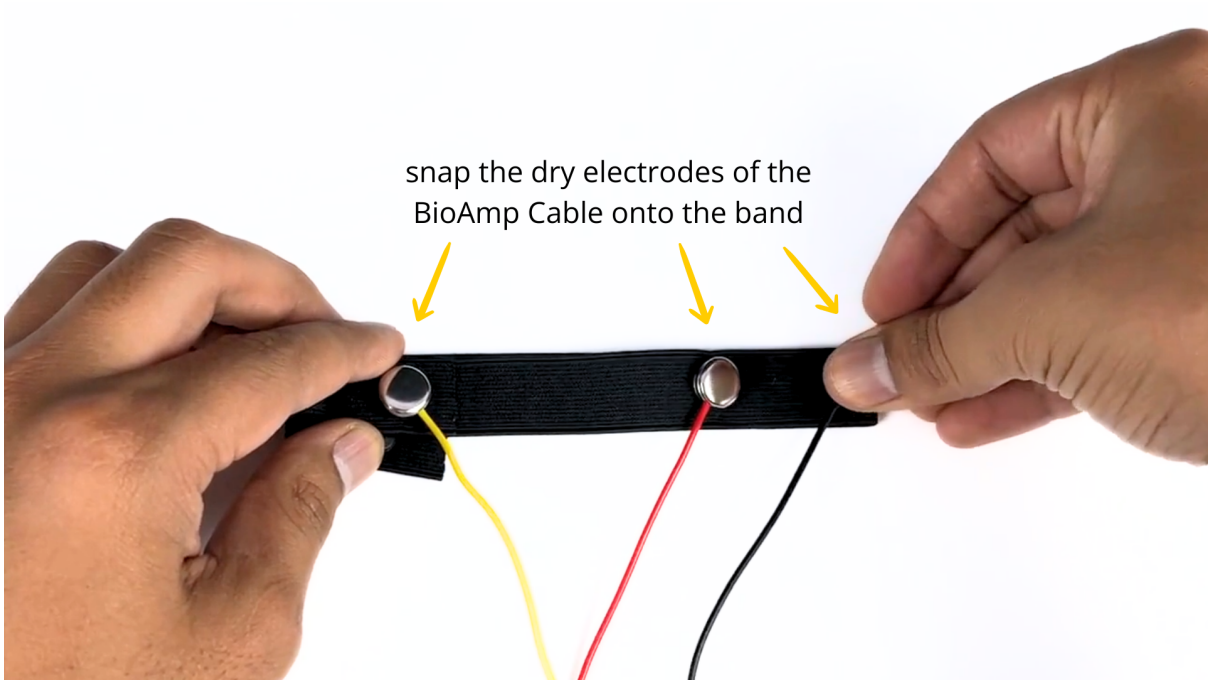
4.4.2 Skin Preparation

Apply Nuprep Skin Preparation Gel on the skin surface where dry electrodes would be placed to remove dead skin cells and clean the skin from dirt. After rubbing the skin surface thoroughly, clean it with an alcohol wipe or a wet wipe.

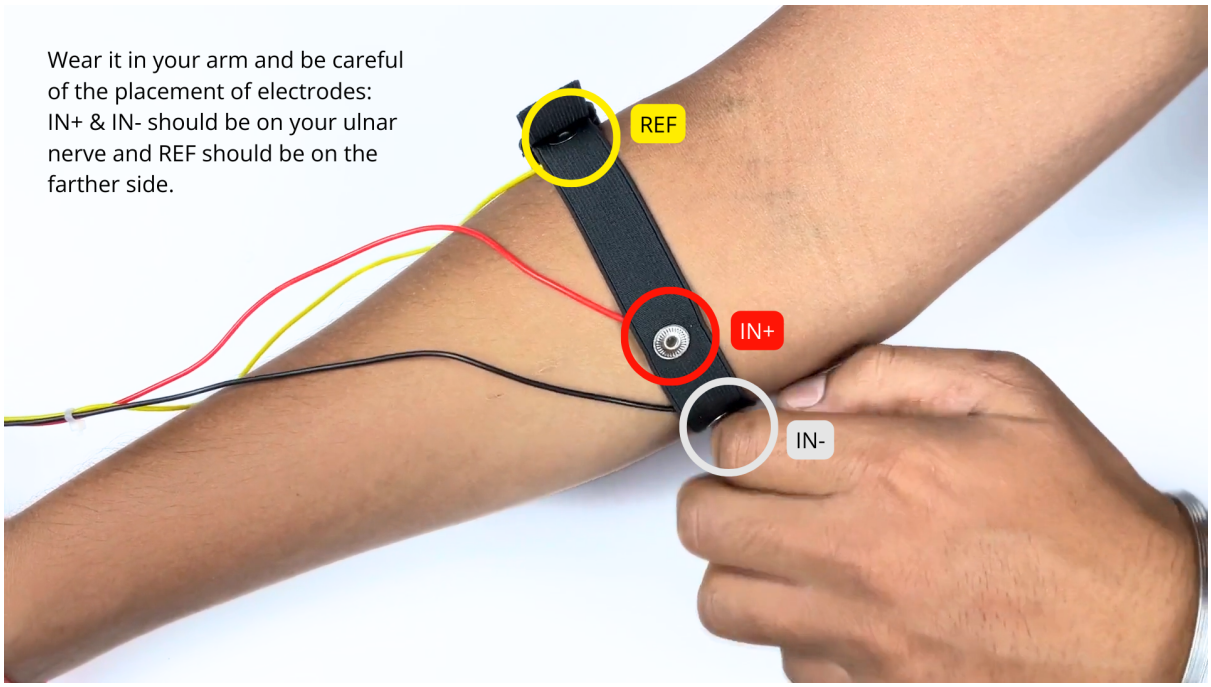
For more information, please check out detailed step by step *Skin Preparation Guide*.

4.4.3 Measure EMG

1. Flip the band and snap the dry electrodes of the BioAmp Cable on it as shown below.



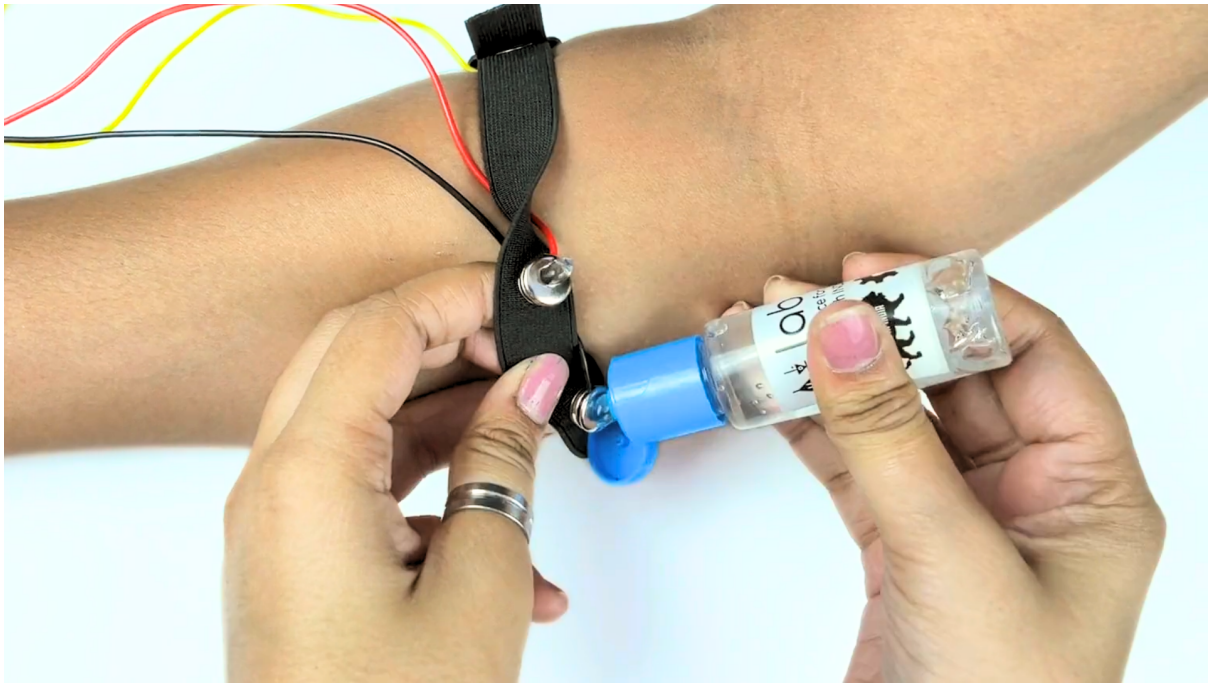
1. Flip the band again and wear it on your arm in such a way that IN+ and IN- are placed on the arm near the ulnar nerve and REF (reference) on the far side of the band.



Note

Make sure the dry electrodes (shiny parts of the BioAmp Cable) are in direct contact with the skin.

3. Now put a small amount of electrode gel or Ten20 paste between the skin and dry electrodes to get the best signal acquisition.



Note

- After using the band, don't leave the gel residue on the dry electrodes longer than an hour as it may corrode them over a period of time.
- Wash the band with liquid soap and rinse it properly after every use. Use it again only when it is completely dry.

4.5 Using Heart BioAmp Band

4.5.1 Skin Preparation

Apply Nuprep Skin Preparation Gel on your chest where dry electrodes would be placed to remove dead skin cells and clean the skin from dirt. After rubbing the skin surface thoroughly, clean it with an alcohol wipe or a wet wipe.

For more information, please check out detailed step by step *Skin Preparation Guide*.

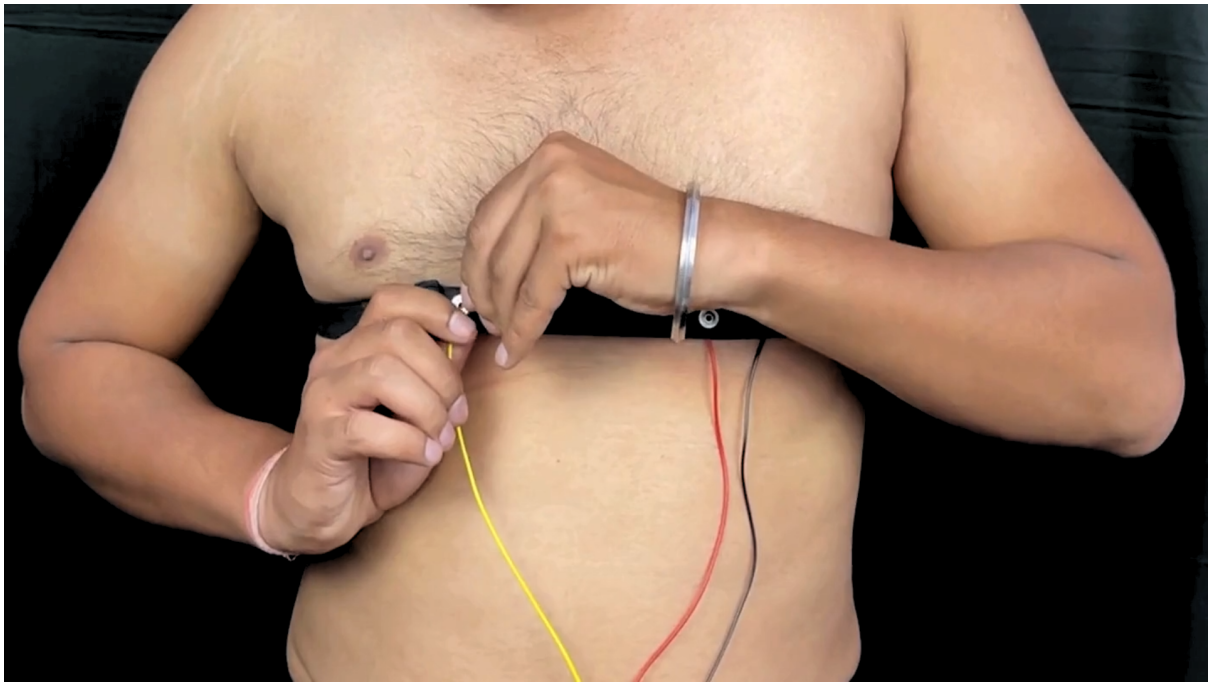
4.5.2 Assembly

1. Take your Heart BioAmp Band and wrap the band around your chest in such a way that the pointy part of the snap touches your chest and the flat part is on the outer side.

2. Now insert the loose end of the band into the buckle and tighten it by pulling the strap.
3. Your band is now ready to use. You can also adjust the size of the band according to your chest size.

4.5.3 Measure ECG

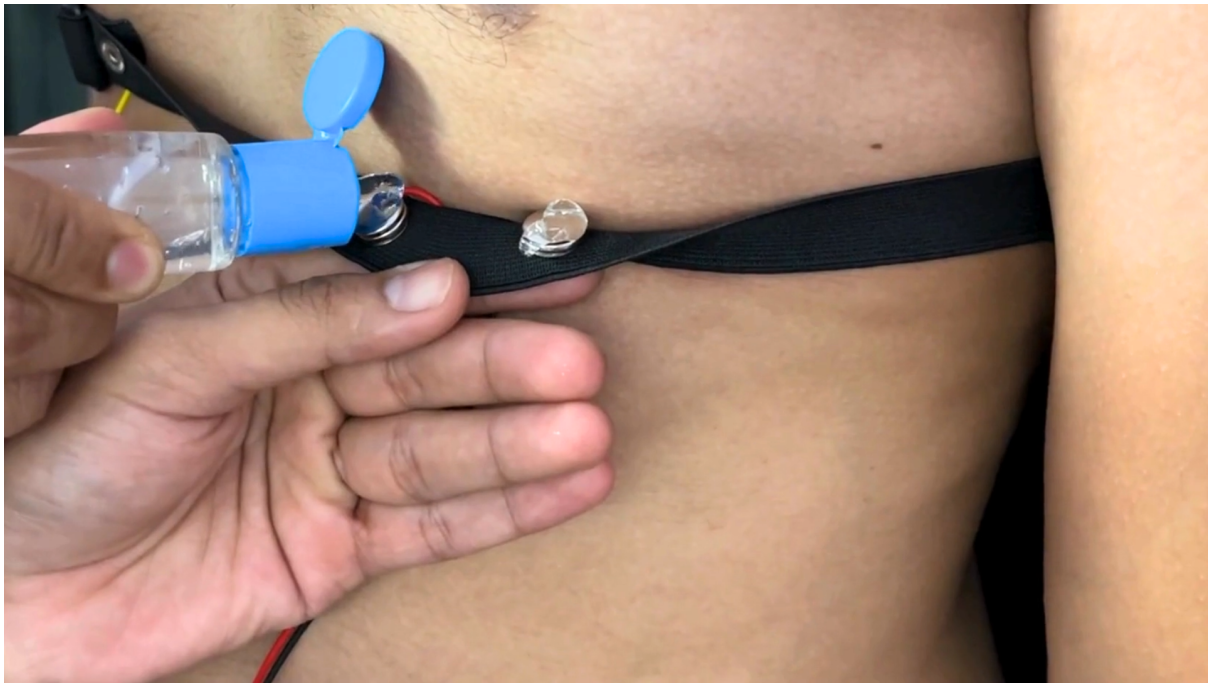
1. Snap the IN- cable on the left most side of the band, IN+ cable in the middle, and REF cable on the right side as shown below.



Note

Make sure the dry electrodes (shiny parts of the BioAmp Cable) are in direct contact with the skin.

2. Now put a small amount of electrode gel or Ten20 paste between the skin and dry electrodes to get the best signal acquisition.



Note

- After using the band, don't leave the gel residue on the dry electrodes longer than an hour as it may corrode them over a period of time.
- Wash the band with liquid soap and rinse it properly after every use. Use it again only when it is completely dry.

4.6 Using Brain BioAmp Band

4.6.1 Assembly

You get the band in two parts - the longer part consists of buckles at both ends and the shorter one has loose ends on both sides.

1. Hold one end of the longer band and align the top part of the buckle with the flat surface of the snap.
2. Now take the shorter band and insert it into the buckle of longer band.
3. Repeat step 1 and 2 for the other buckle on the longer band.
4. Your band is now ready to use. You can also adjust the size of the band according to your head size.

4.6.2 Skin Preparation

Apply Nuprep Skin Preparation Gel on your targeted area (visual cortex or prefrontal cortex) where dry electrodes would be placed to remove dead skin cells and clean the skin from dirt. After rubbing the skin surface thoroughly, clean it with an alcohol wipe or a wet wipe.

For more information, please check out detailed step by step *Skin Preparation Guide*.

4.6.3 Measure 1-channel EEG

1. Flip the band, take your BioAmp Cable, and snap the REF cable on a gel electrode. Now snap the IN- and IN+ cable on:

- Fp1 and Fp2 positions for recording EEG from prefrontal cortex
- O1 and O2 positions for recording EEG from visual cortex

Note

The electrode positions mentioned above are according to International 10-20 sytem for recording EEG.

2. Flip the band again and wear it in a way so that the dry electrodes (shiny parts of the cable) are in contact with:

- skin surface on the forehead (if recording from prefrontal cortex)
- scalp surface on the back side of your head (if recording from visual cortex)

3. Peel of the plastic backing of the gel electrode and place it on the bony part behind your earlobe.

Note

While placing the gel electrodes on the skin, make sure to place the non-sticky tab of the electrode in the direction opposite to your hair growth. This allows you to remove the electrodes easily without pulling off much body hair.

4. Now put a small amount of electrode gel or Ten20 paste between the skin/scalp and dry electrodes to get the best signal acquisition.

CHAPTER

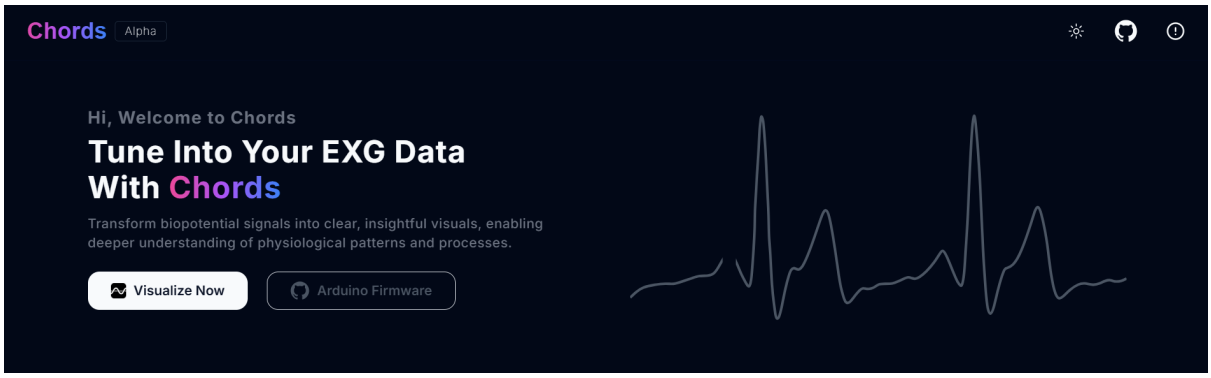
FIVE

CHORDS-WEB

https://youtu.be/fM0c6JVh3uE

5.1 Overview

Chords-Web is an open-source web application designed for real-time signal visualization, particularly tailored for bio-potential signals like EEG, EMG, ECG and EOG. This tool serves as an advanced alternative to the standard Arduino IDE serial plotter, offering enhanced functionality for students, researchers and hobbyists alike who want to record biopotential data using BioAmp hardware.



5.2 Browser Compatibility

The web application is compatible with the **Web Serial API**, which is essential for its functionality. Supported browsers include the latest versions of:

- **Google Chrome**
- **Microsoft Edge**
- **Opera**

If the user's browser does not support the Web Serial API, a message will inform them of the incompatibility, recommending the use of a supported browser.

For more information, refer to MDN Web Docs on the Web Serial API.

5.3 Software Requirements

- Arduino Firmware (Use this software to upload Chords Arduino firmware to your development board).
- NPG Lite Flasher (Use this software to upload NPG Lite firmware to your NPG Lite board or any ESP32 based board).
- Chromium-based browsers (Know more about *Browser Compatibility*).

5.4 Hardware Requirements

- A development board (see the list of Compatible Development Boards)
- A USB cable (type depends on board)
- BioAmp Hardware and its accessories

5.5 Setting up the hardware

Make all the connections according to the hardware you are using, including sensor connections with the development board, body connections with the sensor, and connections from the development board to your laptop.

5.5.1 Uploading the code

1. Connect your Arduino board to the laptop using a USB cable.
2. Open the Arduino Firmware for your board in the Arduino IDE.
3. Navigate to *Tools* > *Board* and select your board.
4. In the tools menu, choose the correct COM port (the one that disappears when the board is disconnected).
5. Upload the code and open Chords-Web in your web browser.

Once ready, upload the firmware to your development board. Visit the Chords Arduino Firmware GitHub repository, scroll to the supported boards table, locate your board, and click on the corresponding Arduino sketch.

Copy the sketch, paste it into the Arduino IDE, go to *Tools*, select your board and the correct COM port, then click the upload button.

To learn about how to flash the code on the NPG Lite Board, checkout:

5.6 Opening Chords-web

1. Open a Chromium-based browser like Google Chrome, Microsoft Edge, Opera, brave, etc.
2. Search chords.upsidedownlabs.tech .
3. Click on visualize now.
4. The images in the center show an overview of your hardware connections so far.
5. At the bottom, you can see buttons to access various applications, which are discussed below:

5.7 Applications

5.7.1 Chords Visualizer

The Chords Visualizer is a powerful web-based tool designed for seamless real-time biopotential signal acquisition and analysis. Designed for researchers, developers, and enthusiasts, this app provides an intuitive interface for monitoring multiple channels, applying advanced filters, and managing recorded data efficiently. Whether you're analyzing EMG, ECG, EOG, or EEG signals, the app ensures a smooth and interactive experience, simplifying data acquisition and enhancing signal interpretation.

Features

Feature	Description
Effortless Connectivity	Instantly detects **BioAmp Hardware** running **Chords-Web Arduino Firmware**, simplifying setup and ensuring smooth workflow from data acquisition to visualization.
Real-time Visualization	Experience smooth, real-time data rendering powered by WebGL-Plot. Ensures uninterrupted signal display for seamless data monitoring.
Frame Buffer & Snapshots	Stores the **last five snapshots**, with left/right controls for navigation. Changing the **channel count resets snapshots** for accuracy. Users can **zoom in and out** for detailed or broad views.
Recording & Data Management	Record data in **CSV format** indefinitely or with a timer. Manage files via a popover menu and **download or delete** them individually or as a ZIP with one click.
Zoom & Time Base	Use **Zoom-In & Zoom-Out** controls to focus on signal details. Adjust the **Time Base Slider** from **1 to 10 seconds per frame** for flexible data visualization.
Filter Options	Enhance biopotential signals using the **notch filter** and **bandpass filters**. Apply them individually per channel or collectively via the **master filter**.
Channel Support	Supports **real-time multi-channel plotting** with color-coded streams. Accommodates **up to 16 channels**, depending on the connected device, for flexible biopotential signal applications.
One-Click Disconnect	Easily disconnect from the development board with a single click, ensuring a hassle-free disconnection process after data collection or visualization.

Chords-Web Icons

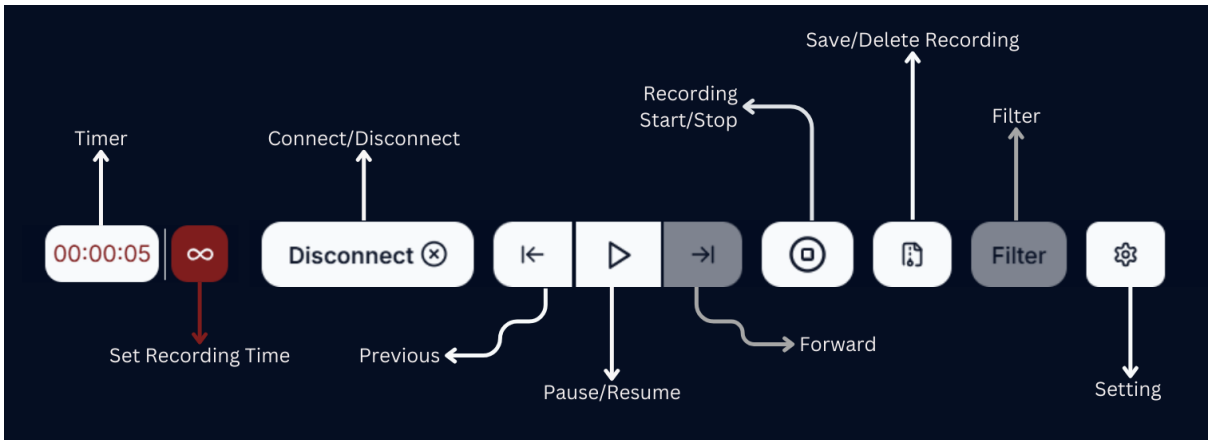


Fig. 1: Chords-Web Icons

Play/Pause Data Stream

- Clicking the pause button displays the last saved frame.
- You can view and save up to the last five snapshots of your data.
- Snapshots are automatically captured per frame.
- Navigate snapshots using the left and right buttons.

Setting Channel Count

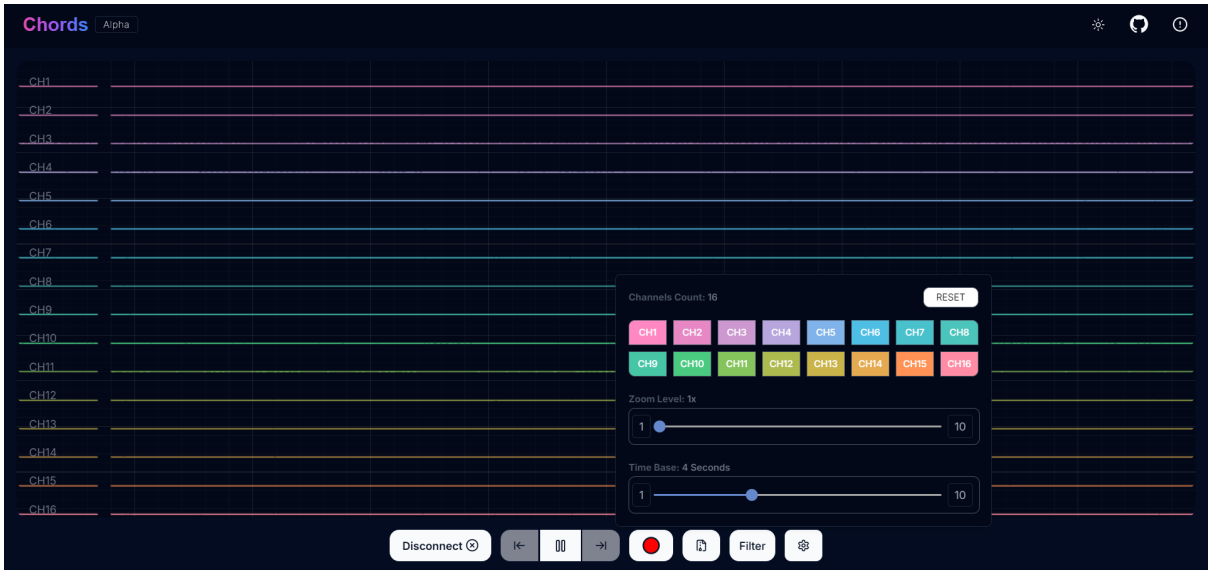


Fig. 2: Chords-Web Channel Support

- The number of available channels depends on the development board in use.
- Select a specific channel by clicking the channel button.
- Use the “Select All” button to choose all available channels at once.
- Click the reset button to revert to your previously selected channels.

Recording the Data

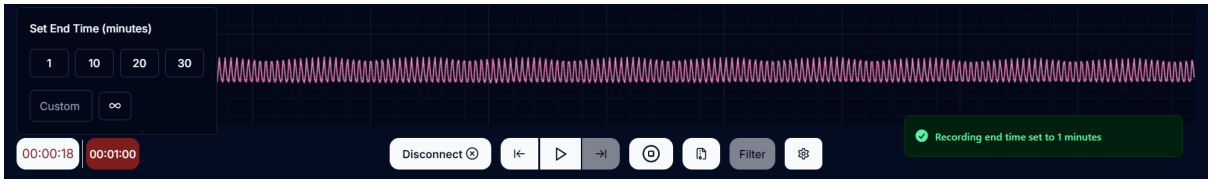


Fig. 3: Recording Time

- **Record** data in **CSV format** for a set duration or indefinitely until manually stopped.
- Start recording with a set time limit or record freely and stop anytime using the stop icon.
- Efficiently **download** or **delete** recorded files through the popover menu.
- Files are securely stored in **IndexedDB** for seamless management.
- Manage individual files by downloading specific files and removing them as needed.
- Easily download all files as a ZIP or delete them with a single click for seamless file management.

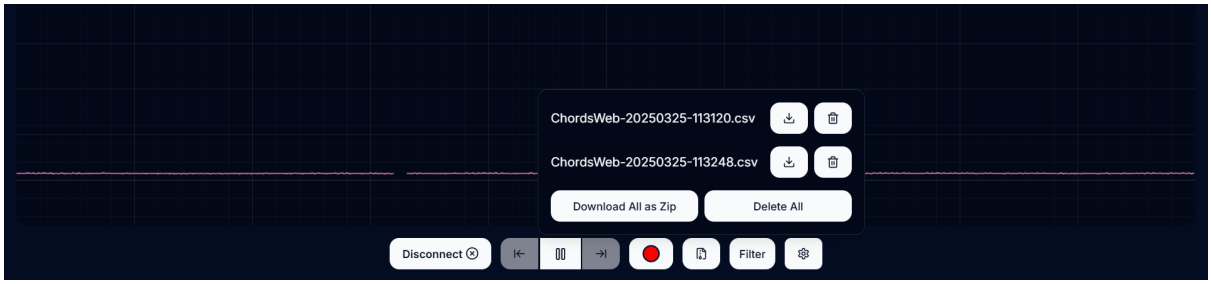


Fig. 4: Save and Delete Option

Visualizing EMG (Electromyography) signal

EMG captures the electrical activity produced by skeletal muscles.

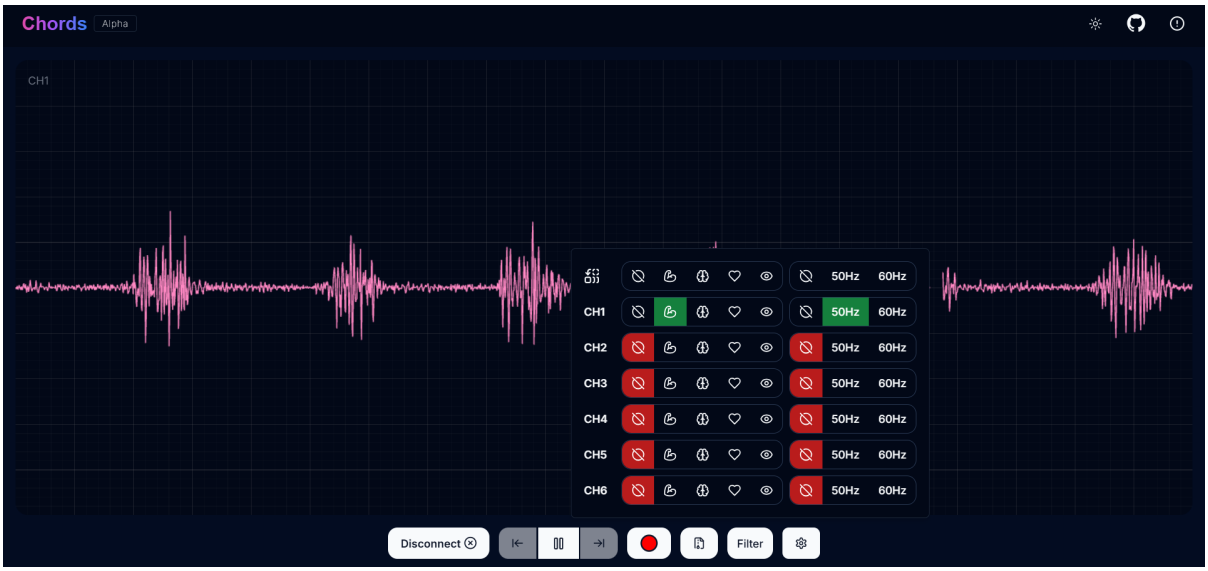


Fig. 5: EMG Signal Example

Visualizing EEG (Electroencephalography) signals

EEG records the electrical activity of the brain and is commonly used for diagnosing neurological conditions and studying brain activity.

Tip

To ensure you're recording a high-quality signal, refer to the detailed guide here: Troubleshooting EEG Signal Quality.

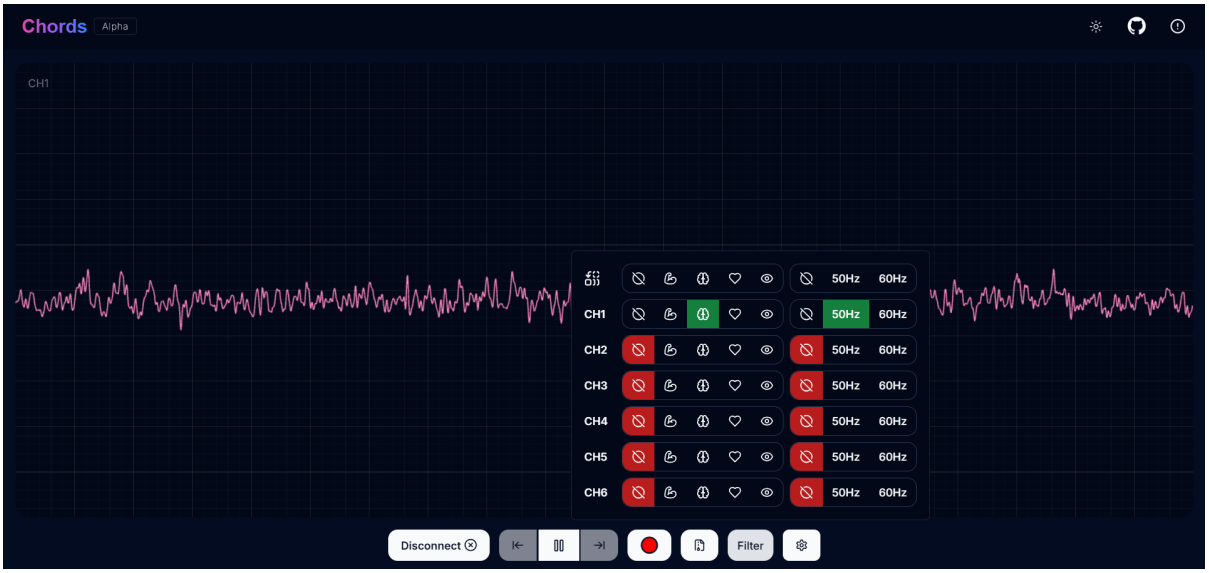


Fig. 6: EEG Signal Example

Visualizing EOG (Electrooculography) signals

EOG measures the electrical potential generated by eye movements.

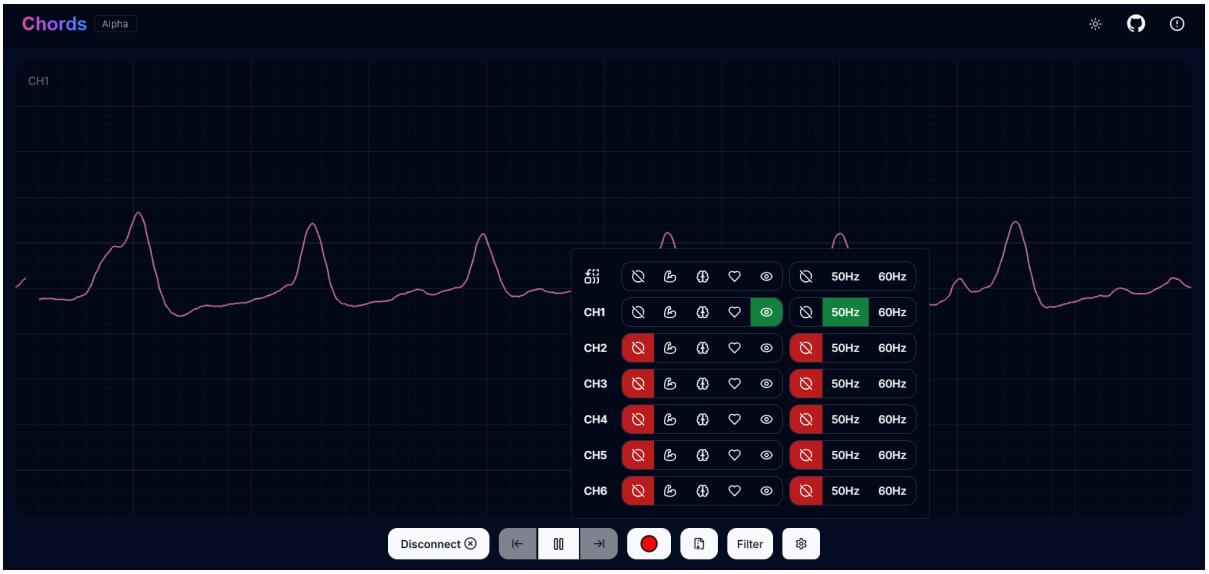


Fig. 7: EOG Signal Example

Visualizing ECG (Electrocardiography) signals

The ECG (Electrocardiography) signal represents the electrical activity of the heart. This custom ECG signal is used both in clinical practice and research to evaluate heart rhythm, detect abnormalities, and assess cardiac health.

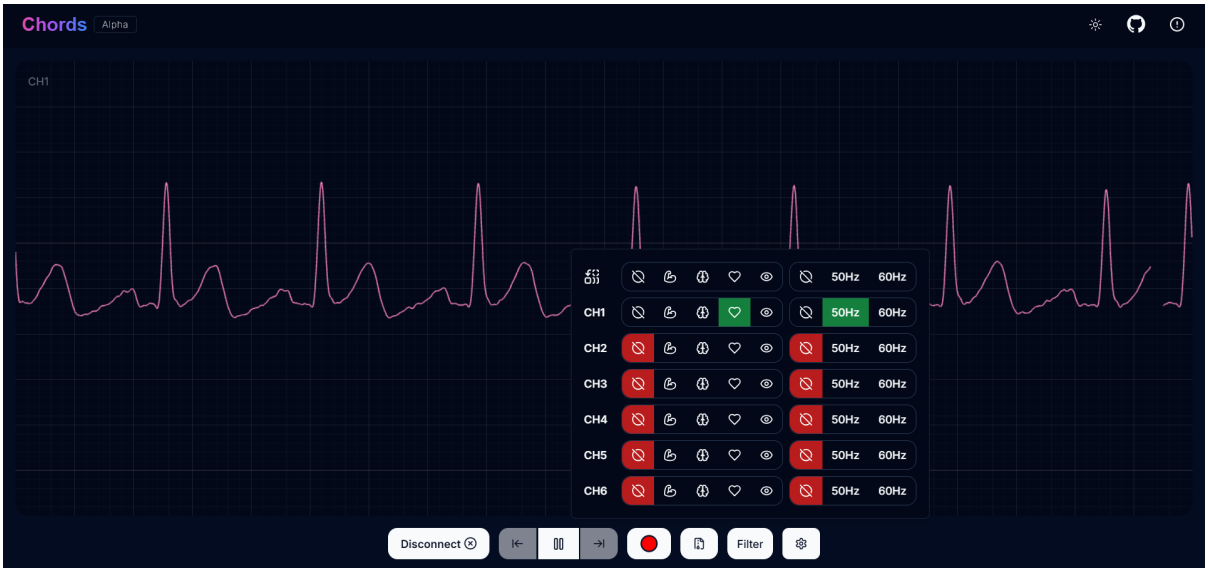


Fig. 8: ECG Signal Example

Other Options to Explore

- **Switch Theme** Quickly switch between light and dark modes using the theme button in the navigation bar.
- **Visit the GitHub Repository** Access the Chords Web GitHub repository via the link in the navigation bar.
- **Contributors** View the list of contributors using the link in the navigation bar's top-right corner.

Running the Application

1. Click the `Visualize Now` button to navigate to the applications page. Here, you will find two options.
2. Click the `Chords Visualizer` button to establish a connection with the Arduino and start streaming data.
3. Use the `ZoomIn/ZoomOut` buttons to adjust data visualization.
4. Use the `Play/Pause` button to control the data stream. Navigate the last five snapshots with the `Left/Right` buttons in the **Frame Buffer** feature.
5. Click the `Record` button to start recording data into a CSV file.
6. Click the `Download` button to save the recorded data.
7. Click the `Delete` button to remove recorded data.
8. Click the `Filter` button to apply filters for EMG, ECG, EOG, and EEG signals: - `Muscle` (70Hz high-pass for EMG) - `Heart` (30Hz low-pass for ECG) - `Eye` (10Hz low-pass for EOG) - `Brain` (45Hz low-pass for EEG) - Use the **Master button** to apply filters across all channels. - Apply **50Hz or 60Hz** filters to individual or all channels.
9. Select channels via the `Channels` button in the settings popover.
10. Adjust zoom using the `Zoom` slider for a detailed or overall view.

5.7.2 Serial Wizard Plotter & Monitor

Overview

The **Serial Wizard Plotter & Monitor** is a standalone feature within Chords-Web that provides real-time serial data visualization.

Fig. 9: Chords-Web Filter

Features

Feature	Description
Dual View Modes	This tool allows you to toggle between the **Plotter**, **Monitor**, or a combined view for comprehensive visualization.
Optimized Data Rendering	In newer Arduino versions, fast data plotting can lead to cluttered displays. The Serial Plotter & Monitor is optimized to handle high-frequency data, ensuring clear and accurate visual representation.
Footer Button Bar	Easily switch between different viewing modes using an intuitive footer button bar.
Baud Rate Selection	Choose from multiple baud rates to optimize serial communication based on your device's requirements.
Navigation Bar	Access features such as theme switching (light/dark), visit the GitHub repository, view contributor details, or return to the previous page.

Running the Application

1. Click the **Serial Wizard** button to launch the Serial Plotter & Monitor.
2. Click on Connect button select board.
3. Use the footer button bar to toggle between the Plotter, Monitor, or a combined view.
4. Navigate using the top bar to switch themes, visit the GitHub repository, view contributors, or return to the previous page.

Note

Checkout our YouTube video for more information:

https://youtu.be/-C_QUpwcEJQ

5.7.3 FFT Analysis and EEG Band Spectrum Plotting

Overview

We have introduced **FFT (Fast Fourier Transform) analysis** and **EEG band spectrum plotting** to improve real-time signal processing. These features enable you to visualize and analyze EEG frequency bands, providing deeper insights into brain activity.

Fig. 10: Chords-Web FFT Visualiser

Features

Feature	Description
Download EEG Data	Save recorded channel data as a **CSV file** for further analysis, storage, or external visualization.
Live EEG Band Monitoring	View real-time plots of EEG band values for better brain activity tracking.
Channel Selection	Toggle individual channels (CH0-CH2) on or off to customize which electrode inputs are displayed and recorded.

Supported EEG Bands

- **Delta (0.5 - 4 Hz)** → Associated with deep sleep and unconscious states.
- **Theta (4 - 8 Hz)** → Linked to relaxation, meditation, and light sleep.
- **Alpha (8 - 13 Hz)** → Reflects calm, wakeful relaxation, often seen during closed-eye rest.
- **Beta (13 - 30 Hz)** → Related to active thinking, problem-solving, and focus.
- **Gamma (30 - 45 Hz)** → Involved in high-level cognitive functioning, attention, and perception.

Running the Application

1. Select **"FFT Visualizer"** to see your brainwaves in real time.
2. You will get two option, select the appropriate option based on how your device is connected:
   - Serial
   - Bluetooth
3. The **top segment** displays filtered EEG data using a **45Hz low-pass filter** to remove noise.
4. The **bottom segment** is divided into two sections:
   - **Left side** → Shows EEG frequency values in Hz.
   - **Right side** → Offers two interactive modes:
     - **Band Power Mode** → Displays real-time EEG band power values.
     - **Beta Candle Mode** → A unique visualization where a glowing candle represents your focus level.
       * **Brighter candle** = Higher beta waves = Strong focus.
       * **Dim candle** = Lower beta waves = Distraction.

Note

Checkout our YouTube video for more information:

https://youtu.be/zkPGzX3GKnk

5.7.4 NPG Lite

Overview

We have added support for NPG Lite, enabling real-time visualization of signals directly from the onboard 3-channel BioAmp. Powered by the ESP32-C6 with a built-in 12-bit ADC, setup is seamless: simply upload the firmware, power on the board, and begin streaming instantly.

Fig. 11: Chords-Web NPG Lite

Features

Feature	Description
Wireless Bluetooth LE	Stream up to 3 channels of biopotential data over BLE - no cables after initial power-on.
Built-in BioAmp & 12-bit ADC	On-board amplifier and ESP32-C6 ADC ensure high-quality signal capture.
Interactive Controls	Select channels, play/pause live stream, apply bandpass & 50/60 Hz notch filters, and record to CSV.

Running the application

1. **Flash the correct firmware** - Visit the NPG-Lite Flasher and upload the **BLE firmware** to your board.
2. **Open Chords-Web** - Navigate to https://chords.upsidedownlabs.tech and click **Visualize Now**.
3. **Select "NPG-Lite"** - From the application list, choose **NPG-Lite**.
4. **Enable Bluetooth on your computer** - Turn on your system's Bluetooth, then click **Connect** in Chords-Web.
5. **Choose your device** - Select your NPG-Lite from the list of available devices.
6. **Begin streaming** - Visualize your biopotential signals in real time, select 1-3 channels, toggle play/pause, apply filters, or record to CSV.

Use Cases

- **Neurofeedback & Focus Training**: Monitor alpha/beta power to track attention.
- **Rehabilitation & Sports Science**: Quantify muscle (EMG) activity during exercises.
- **Research & Education**: Capture synchronized EEG/ECG/EOG data for analysis.

Note

Checkout our YouTube video for more information:

https://youtu.be/3YCioyc4uKs

5.7.5 Rep Forge

Overview

We've added **Rep-Forge** - a real-time, 3-channel EMG visualization tool for the NPG Lite. Rep-Forge lets you monitor your muscle strength as you work out or rehabilitate, directly in your browser. No external ADC or development board required: simply flash your firmware, apply electrodes, and start streaming.

Features

Feature	Description
3-Channel EMG Streaming	Stream up to three simultaneous EMG signals from your NPG Lite's onboard BioAmp.
Intelligent Channel Detection	Automatically highlights the active muscle channel so you can quickly identify which muscle you're engaging.
Live Strength Bars	Dynamic bar graphs update in real time to show relative contraction levels for each channel.
Noise-Reducing Filters	Built-in signal filters remove 50/60 Hz mains interference and high-frequency artifacts for a cleaner EMG trace.
Wireless BLE Connectivity	Stream data over Bluetooth LE—no cables needed once your board is powered on.

Running the application

1. **Power on your NPG Lite** by flipping its on/off switch.
2. **Connect** via USB-C and flash the **Rep-Forge** firmware using the NPG Lite Flasher.
3. **Unplug** USB and enable Bluetooth on your computer (or use Serial/Wi-Fi as needed).
4. Open your browser to **chords.upsidedownlabs.tech**, click **Visualize Now**, then choose **Rep-Forge**.
5. Click **Connect**, select your NPG Lite device from the list, and wait for the live EMG bars to appear.
6. **Place electrodes** on the target muscle group; watch Rep-Forge dynamically highlight and plot the active channel's strength.

Use Cases

- **Strength Training**: Quantify muscle activation during lifts or repetitions.
- **Rehabilitation**: Monitor recovery progress in injured muscle groups.
- **Research**: Capture high-quality EMG data for neuromuscular studies.

5.8 Technologies Used

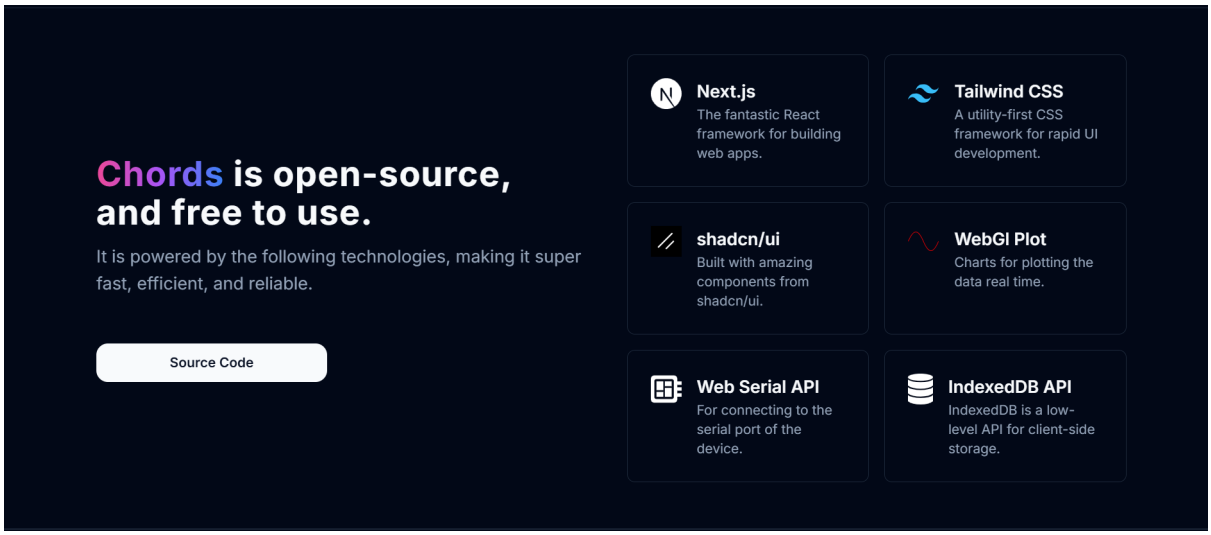


Fig. 12: Chords-Web Tech Stack

CHAPTER

SIX

CHORDS-PYTHON

https://youtu.be/bj7exKRsuZ8

6.1 Overview

Chords-Python is an open-source bag of tools for recording biopotential signals like **ECG**, **EMG**, **EEG**, or **EOG**, along with visualization using BioAmp hardware. It's ideal for educational purposes as it promotes DIY Neuroscience and makes biopotential signal exploration more accessible for students and researchers.

6.2 Features

Feature	Description
Connection	Supports both wired and wireless connections via Wi-Fi, Bluetooth, or USB (Serial).
Data Reading	Reads data packets from development board in real-time, efficiently processing them to prevent data loss and ensure accurate signal representation for analysis and visualization.
CSV Logging	Optionally logs incoming data to a CSV file with columns for counter and channel data, enabling easy storage, analysis, and sharing.
LSL Streaming	Streams data via Lab Streaming Layer (LSL), a protocol for time-synchronized data sharing. Enables real-time analysis, visualization, or integration with tools like BrainVision.
Applications	Provides an interface to run multiple applications simultaneously via the LSL stream.
Create Custom Application	Allows users to develop and integrate their own applications with the system.

6.3 Software Requirements

- Arduino IDE – Required to upload the Chords Arduino firmware to your development board.
- Python – Ensure you have the most recent version installed.
- VS Code or any other code editor (Alternatively, you can use the Command Prompt).

6.4 Hardware Requirements

To use Chords-Python, you need:

- A development board (Compatible Boards)
- A USB cable (type depends on the board)
- BioAmp Hardware and its accessories

6.5 Setting up the hardware

Make all the connections according to the hardware you are using , including sensor connections with the development board, body connections with the sensor, and connections from the development board to your laptop.

6.6 Uploading the code

Once you are all set, it is time to upload the code to your development board. Go to Chords Arduino Firmware GitHub repo, scroll down to the supported boards table, find your board name, and click on the Arduino sketch corresponding to that row. Copy the sketch and paste it into Arduino IDE. Go to tools, select your board, and the correct COM port. Now, hit the upload button.

6.7 Opening Chords-Python

There are two ways to use Chords-Python:

A. Using the `chordspy` Python Package (Recommended)

B. Running Scripts Manually from the Repository

6.7.1 A. Using the chordspy Package

This is the smoothest way to get started. Follow the steps below:

1. **Install Python** Make sure latest version of Python is installed.
2. **Create and Activate a Virtual Environment**:
   - **On Windows**:

     ```
     python -m venv .venv
     ```

     ```
     .venv\Scripts\activate
     ```

   - **On macOS/Linux**:

     ```
     python3 -m venv .venv
     ```

     ```
     source .venv/bin/activate
     ```

3. **Install the Package**:

   ```
   pip install chordspy
   ```

4. **Launch the Web Interface**:

```
chordspy
```

A web interface will open where you can connect your device and access applications.

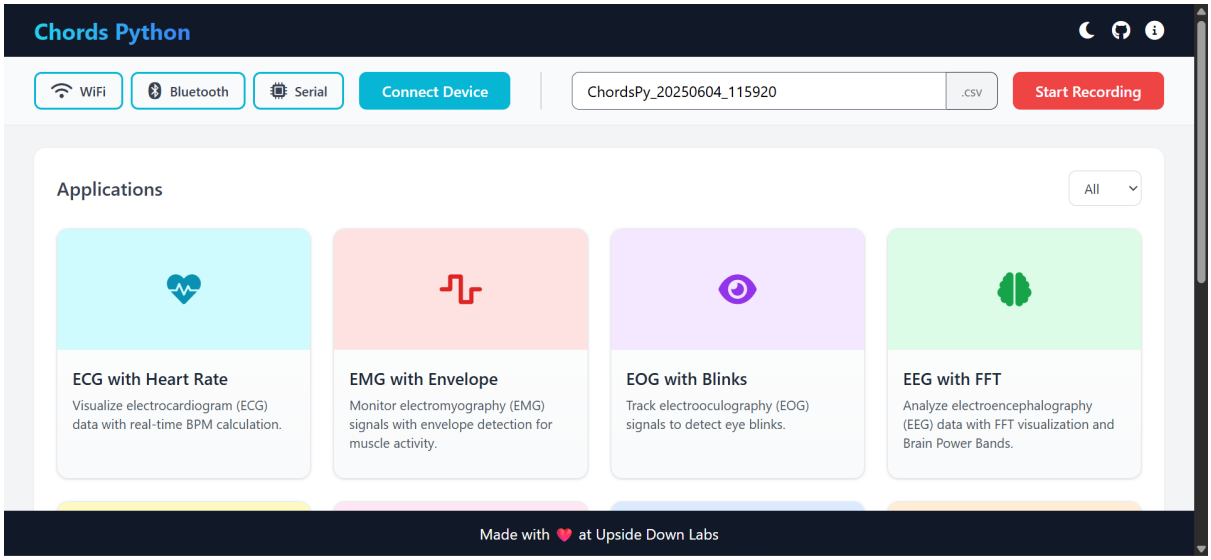


Fig. 1: Interface in Light Mode

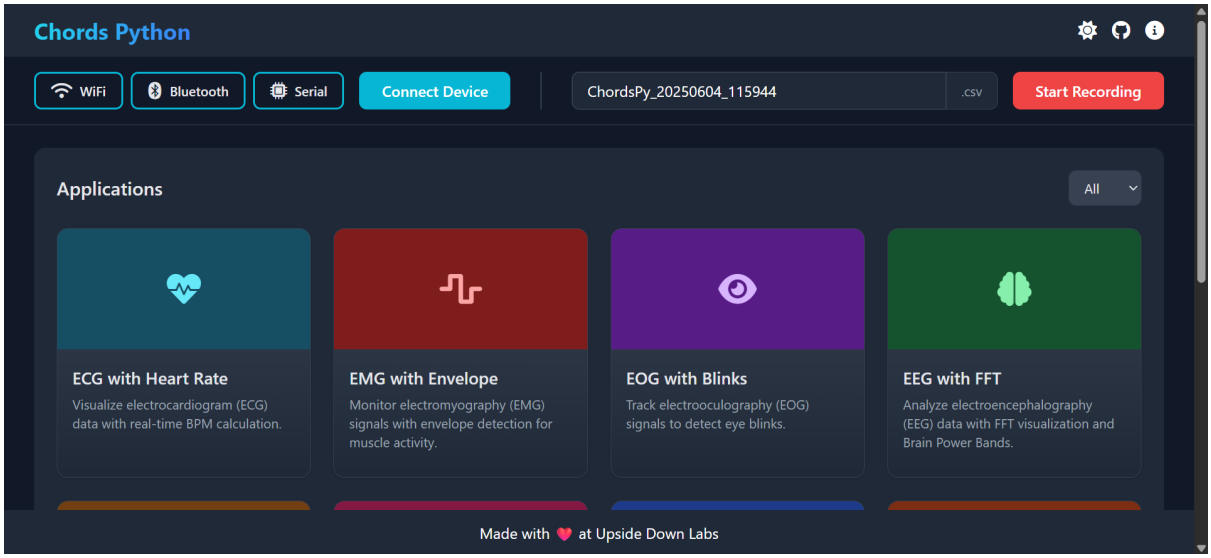


Fig. 2: Interface in Dark Mode

6.7.2 B. Running Scripts Manually (Alternative)

If you prefer running scripts directly (for development, debugging, or customization):

1. **Download the Repository**:
   - You can download the Chords-Python repository from GitHub by visiting the following link: Chords-Python.
   - Or, you can clone the repository using Git by running the following command:

```
git clone https://github.com/upsidedownlabs/Chords-Python.git
```

2. **Create and Activate a Virtual Environment** (if not already):
   - **On Windows**:

     ```
     python -m venv .venv
     ```

     ```
     .venv\Scripts\activate
     ```

   - **On macOS/Linux**:

     ```
     python3 -m venv .venv
     ```

     ```
     source .venv/bin/activate
     ```

3. **Install Requirements**:

   ```
   pip install -r requirements.txt
   ```

4. **Run the Application**:

   Navigate to the *chordspy* folder and run:

   ```
   python -m chordspy.app                          # To launch the web interface
   python -m chordspy.connection --protocol usb    # To start LSL stream via USB
   python -m chordspy.connection --protocol ble    # To start LSL stream via BLE
   python -m chordspy.connection --protocol wifi   # To start LSL stream via WiFi
   ```

   To run any application, open a new terminal:

   ```
   python chordspy.gui.py          # GUI Application
   python chordspy.ffteeg.py       # EEG with FFT Analysis
   ```

6.8 Connection

The first step is to establish a connection with your device and start the stream.

There are three connection options available:

- Wi-Fi
- Bluetooth
- Serial (USB)

6.8.1 Wi-Fi

1. Upload the Wi-Fi firmware through the `Chords-Arduino-Firmware` repository
2. Turn on the device and connect to the access point created by the device (e.g., `npg-lite-2`)
3. In the web interface:
   - Click the **Wi-Fi** button
   - Click the **Connect** button

A pop-up notification will appear indicating a successful connection.

6.8.2 Bluetooth

1. Upload the Bluetooth firmware through the `Chords-Arduino-Firmware` repository
2. Turn on the device and enable Bluetooth on your computer
3. In the web interface:
    - Click the **Bluetooth** button
    - Select your device from the list of available devices
    - Hit the **Connect** button

A pop-up notification will appear indicating a successful connection.

6.8.3 Serial (USB)

1. Upload the Serial firmware through the `Chords-Arduino-Firmware` repository
2. Connect the device to your computer using a USB cable
3. In the web interface:
    - Click the **Serial** button
    - Click the **Connect** button

A pop-up notification will appear indicating a successful connection.

Note

The connection step is essential as it initiates the LSL Stream, which is required for running applications.

6.9 CSV Logging

The raw data received from the device can be logged to a CSV file for further analysis or record-keeping. This optional feature can be enabled or disabled in the web interface.

To use CSV logging:

1. Click the **Start recording** button to begin logging - A file with name is created `ChordsPy_{timestamp}.csv` in the same folder. - File includes columns for counter and channel data
2. Click the **Stop recording** button to end logging - File will be saved in the same folder

Fig. 3: CSV Logging

6.10 Applications

There are many applications available that stream the LSL and can be run for various purposes.

List of available applications:

6.10.1 1. *ECG with Heart Rate*

https://youtu.be/tZud2tc-TGI

Overview

The **ECG with Heart Rate** is a real-time application designed to visualize and analyze Electrocardiogram (ECG) data using the Lab Streaming Layer (LSL) protocol. Built with Python and PyQt5, this application provides a graphical interface for monitoring ECG signals, detecting R-peaks (heartbeats), and calculating the heart rate in real time. It applies signal processing techniques and utilizes the *neurokit2* library to estimate R-peak detection and heart rate.

Features

Features	Description
1. Real-Time ECG Visualization	• Displays real-time ECG signals in a dynamic plot using *pyqtgraph*. • Supports adjustable y-axis scaling based on the sampling rate (e.g. 250 Hz or 500 Hz).
2. R-Peak Detection	• Utilizes the *neurokit2* library to detect R-peaks in the ECG signal. • Highlights detected R-peaks as red dots on the plot for easy visualization.
3. Heart Rate Calculation	• Computes heart rate (in BPM) using the time intervals between consecutive R-peaks. • Implements a moving average filter to smooth heart rate values and reduce noise.
4. Interactive GUI	• Built with PyQt5, providing a user-friendly interface with a real-time ECG plot and heart rate display. • Allows double-click to reset the plot's zoom level to default settings.
5. Signal Filtering	• Applies a low-pass Butterworth filter to remove high-frequency noise from the ECG signal. • Helps in enhancing signal clarity, which can assist in identifying R-peaks.

A GUI window will appear, displaying the real-time ECG signal along with the calculated heart rate.

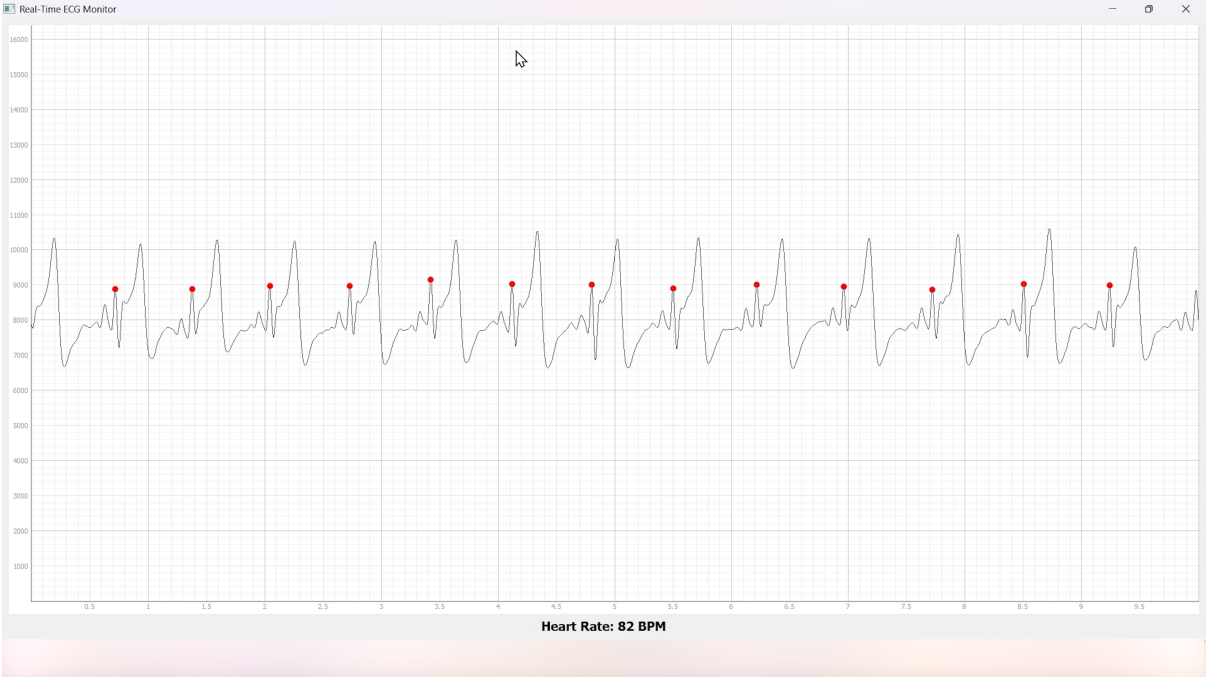


Fig. 4: Heart Rate with ECG

6.10.2 2. *EMG with Envelope*

https://youtu.be/TiDwSQEY2eY

Overview

The **EMG with Envelope** is a Python-based application designed to visualize and analyze Electromyography (EMG) signals in real-time. It connects to an EMG data stream using the Lab Streaming Layer (LSL) protocol, processes the signal to extract the EMG envelope, and displays both the filtered EMG signal and its envelope in a user-friendly graphical interface. Built with *PyQt5* and *pyqtgraph*, the application provides a responsive and interactive visualization tool for students, researchers, or developers working with EMG data.

Features

Features	Description
1. Real-Time EMG Signal Visualization	• Connects to an LSL stream to acquire real-time EMG data. • Displays the EMG signal after applying a high-pass filter (70 Hz cutoff) to remove low-frequency noise.
2. EMG Envelope Extraction	• Computes the Root Mean Square (RMS) envelope of the filtered EMG signal using a moving window. • Applies convolution with a uniform window and pads the result to align with the original signal length.
3. Interactive and Responsive GUI	• Built using *PyQt5* for a modern and intuitive user interface. • Features two synchronized plots: one for the filtered EMG signal and one for the EMG envelope. • Disables zoom and pan for a clean, fixed-axis visualization.
4. Customizable Signal Processing	• Implements a high-pass Butterworth filter to remove baseline drift and noise. • Adjusts the RMS window size dynamically based on the sampling rate (e.g., 25 samples for 250 Hz, 50 samples for 500 Hz).
5. Dynamic Plot Updates	• Updates the plots in real-time using a fixed-size circular buffer for efficient data handling. • Refreshes the display every 15 milliseconds for smooth and responsive visualization.

A GUI window will appear, displaying the real-time EMG signal along with the calculated EMG Envelope.

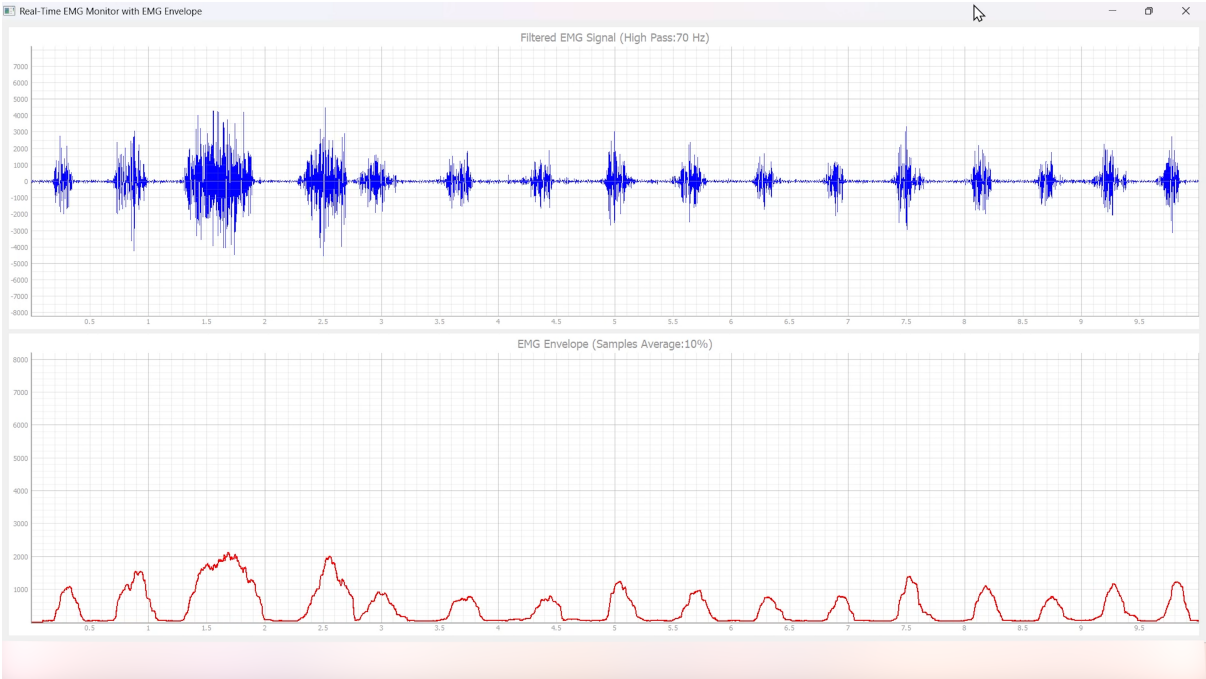


Fig. 5: EMG with Envelope

6.10.3 3. *EOG with Blinks*

The **EOG with Blinks** is a Python-based application designed to visualize and detect eye blinks in real-time using Electrooculography (EOG) signals. Built with the PyQt5 framework and PyQtGraph for plotting, the application connects to an LSL (Lab Streaming Layer) stream to acquire EOG data, processes the signal using a low-pass filter, and detects blinks based on dynamic thresholds. The application provides a dual-plot interface to display the filtered EOG signal and detected blinks, making it a useful tool for real-time monitoring and analysis of EOG data.

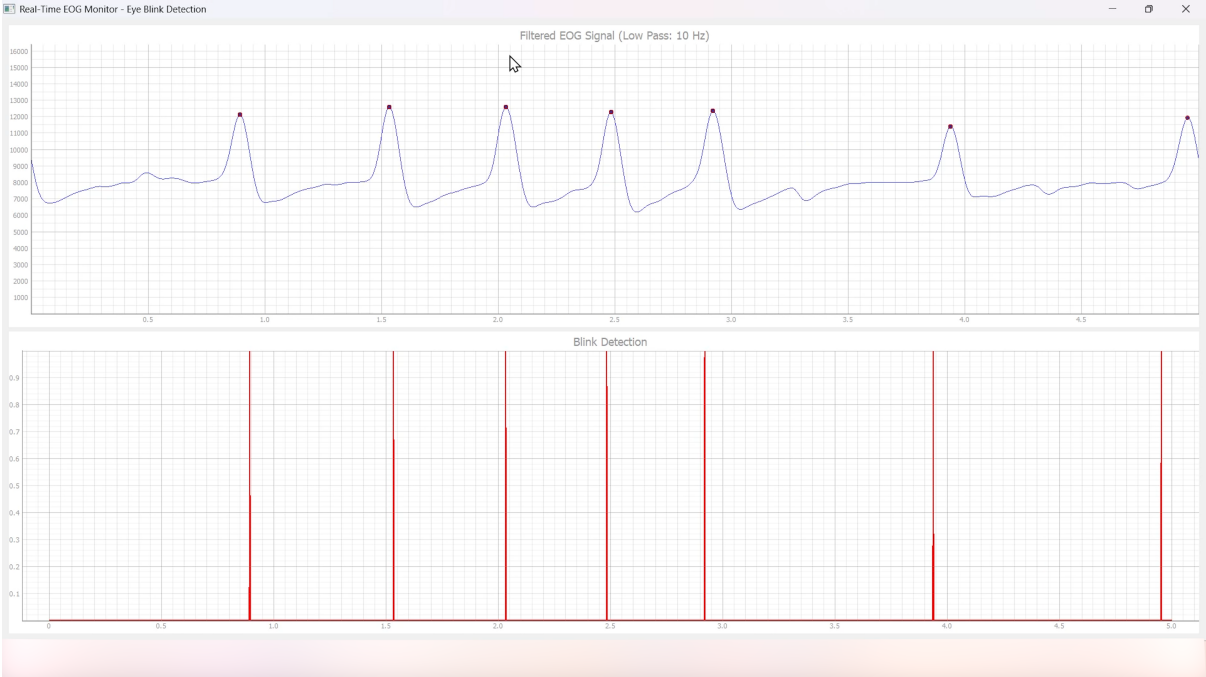


Fig. 6: EOG with Blinks

6.10.4 Features

Features	Description
1. Real-Time EOG Signal Visualization	• Displays the filtered EOG signal in real-time using a low-pass filter (10 Hz cutoff). • Dynamically updates the plot with a 5-second rolling window for continuous monitoring.
2. Dual-Plot Interface	• EOG Signal Plot: Displays the filtered EOG signal with detected peaks marked in red. • Blink Detection Plot: Shows a binary representation of detected blinks (1 for blink, 0 for no blink).
3. Blink Detection	• Detects blinks by identifying peaks in the filtered EOG signal. • Uses a dynamic threshold based on the mean and standard deviation of the signal to distinguish blinks from noise. • Implements a minimum time gap (0.1 seconds) between detected blinks to avoid false positives.
4. User-Friendly GUI	• Built with PyQt5 for a responsive and intuitive interface. • Includes features like grid lines, auto-scaling, and zoom disablement for better usability.

A GUI window will appear, displaying the real-time EOG signal along with the Blinks marked as Red dot.

6.10.5 4. *EEG with FFT*

https://youtu.be/yVD9KmyZgxA

Overview

The **EEG with FFT and Brainwave Power** is a Python-based application designed to visualize and analyze Electroencephalography (EEG) signals in real-time. It connects to an EEG data stream using the Lab Streaming Layer (LSL) protocol, processes the signal to remove noise, and performs Fast Fourier Transform (FFT) to compute the power of different brainwave frequency bands (Delta, Theta, Alpha, Beta, and Gamma). The application provides a graphical user interface (GUI) built with *PyQt5* and *pyqtgraph* for real-time visualization of raw EEG signals, FFT results, and brainwave power distribution.

Features

Features	Description
1. Multi-Channel EEG Visualization	• Displays raw EEG signals from all available channels in real-time. • Each channel shown in a scrolling plot with 500-sample moving window.
2. Multi-Channel FFT Analysis	• Computes and displays FFT for all EEG channels simultaneously. • Visualizes the FFT results in a separate plot, focusing on the 0-50 Hz range.
3. Signal Processing	• Applies a notch filter to remove 50 Hz powerline interference. • Uses a bandpass filter (0.5-48 Hz) to isolate relevant EEG frequencies. • Implements a Hanning window for FFT computation to reduce spectral leakage.
4. Single-Channel Brainwave Power Analysis	• **Calculates the power of five brainwave frequency bands:** – Delta (0.5-4 Hz) – Theta (4-8 Hz) – Alpha (8-13 Hz) – Beta (13-30 Hz) – Gamma (30-45 Hz) • Displays the power of each band in a bar chart for easy comparison.
5. User-Friendly GUI	• **Provides a clean and intuitive interface with Three-panels:** – Top-left: Multi-channel EEG waveforms – Top-right: Multi-channel FFT results – Bottom-right: Single-channel brainwave power analysis • Allows users to monitor multi-channel EEG data and its frequency components simultaneously.

A GUI window will appear, displaying the real-time EEG signal along with the calculated FFT and Brainwave power distribution.

Tip

To ensure you're recording a high-quality signal, refer to the detailed guide here: Troubleshooting EEG Signal Quality.

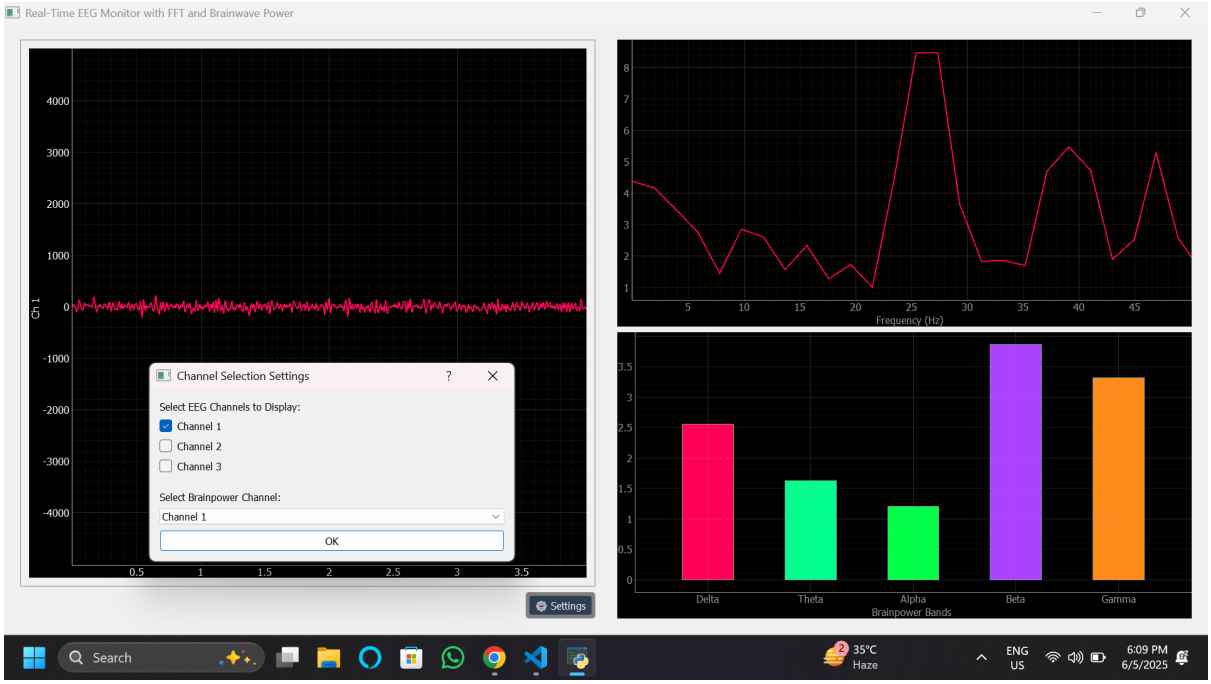


Fig. 7: EEG with FFT

6.10.6 5. *EEG Tug of War Game*

https://youtu.be/XAhcYg1J_7k

Overview

The **EEG Tug of War Game** is a Python-based application that leverages Electroencephalography (EEG) signals to create an interactive two-player game. Players control the movement of a ball on the screen by modulating their brain activity, specifically the Alpha and Beta frequency bands. The game uses the Lab Streaming Layer (LSL) protocol to acquire real-time EEG data, processes the signals to calculate relative power in the Alpha and Beta bands, and translates these into forces that move the ball. The first player aims to push the ball onto the opponent's side to score and win the game. The application is built using the *pygame* library for the graphical interface and integrates with *pylsl* for EEG data acquisition.

Features

Features	Description
1. Real-Time EEG Signal Visualization	• Connects to an LSL stream to acquire real-time EEG data. • Computes the power spectral density (PSD) of Alpha (8-13 Hz) and Beta (13-30 Hz) frequency bands using Welch's method. • Calculates the relative power ratio (Beta/Alpha) to determine player force.
2. Interactive Gameplay	• Two players compete to move a ball to the opponent's side using their brain activity. • The ball's movement is determined by the net force derived from the players' EEG signals.
3. Dynamic Thresholding	• Uses a moving average of the last 10 data points to smooth the force calculations. • Applies a threshold to prevent small fluctuations from affecting the ball's movement.
4. User-Friendly GUI	• Features a full-screen graphical interface with a central ball and two player paddles. • Displays real-time updates of the ball's position and forces applied by each player. • Includes buttons for starting, pausing, resuming, and exiting the game.
5. Win Condition and Feedback	• Declares a winner when the ball reaches either side of the screen. • Plays a sound effect to celebrate the winner. • Automatically pauses the game upon a win and allows for a restart.

The game window will open, featuring buttons for **START/RESTART**, **PLAY/PAUSE**, and **EXIT**. These buttons offer intuitive control, allowing players to easily start, pause, resume, or exit the game as needed.

For detailed instructions, check out the EEG Tug of War Game Instructable.

6.10.7 6. *EEG Beetle Game*

Overview

The **EEG Beetle Game** is a Python-based application that uses Electroencephalography (EEG) signals to control a beetle's movement in a 2D game environment. The game leverages the Lab Streaming Layer (LSL) protocol to acquire real-time EEG data, processes the signal to detect the user's focus level, and translates it into upward or downward movement of the beetle. The application is built using the *pygame* library for the game interface and integrates signal processing techniques to analyze EEG data in real-time.

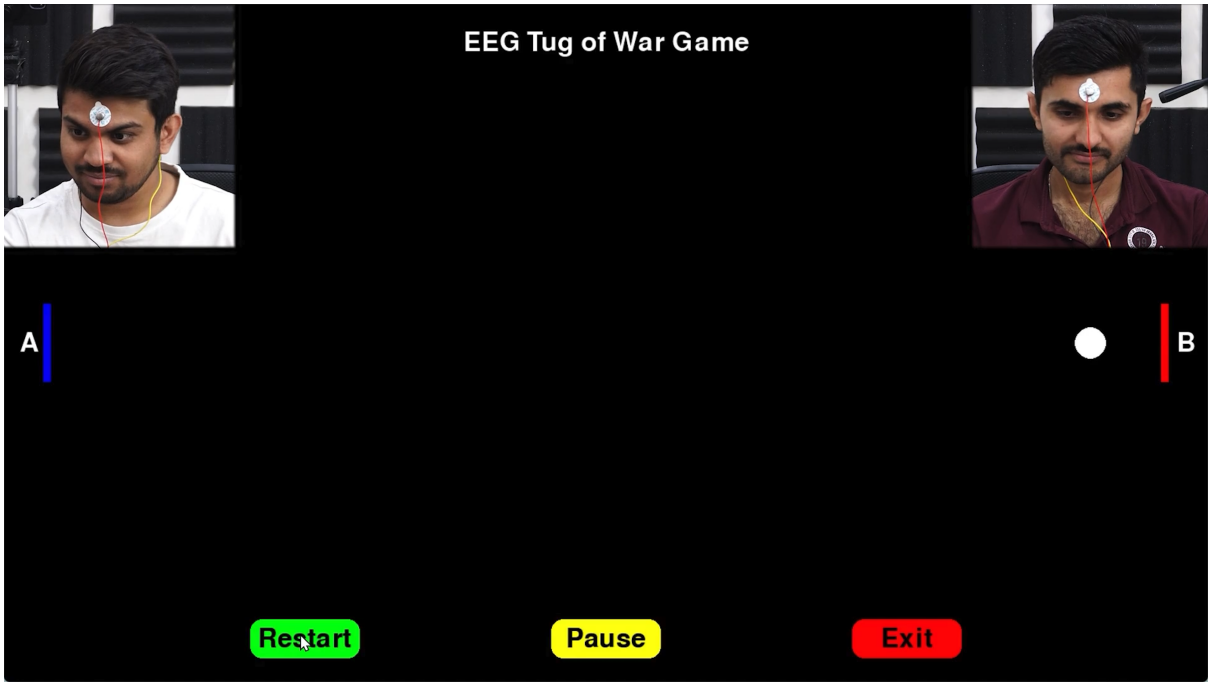


Fig. 8: EEG Tug of War

Features

Features	Description
1. Real-Time EEG Signal Visualization	• Connects to an LSL stream to acquire real-time EEG data. • Implements a notch filter to remove 50 Hz power line interference and a bandpass filter to isolate relevant EEG frequency bands (0.5–48 Hz).
2. Focus Level Calculation	• Computes the user's focus level by analyzing the power spectral density of the EEG signal. • Focus level is calculated using the ratio of high-frequency (beta and gamma) to low-frequency (delta, theta, and alpha) power bands.
3. Calibration System	• Includes a calibration phase to establish a baseline focus level for the user. • Dynamically sets a focus threshold based on the user's EEG data during calibration.
4. Beetle Movement Control	• Moves the beetle upward when the user's focus level exceeds the threshold. • Moves the beetle downward when the focus level is below the threshold. • Implements smooth animation and boundary constraints to ensure the beetle stays within the game window.
5. Interactive Game Interface	• Features a 2D game environment with a beetle sprite that responds to the user's focus level. • Displays real-time feedback on the beetle's position and focus level.
6. Dynamic Animation	• Uses a sequence of beetle sprites to create smooth animations. • Adjusts animation speed based on the game's frame rate.

A GUI window will appear, showing all calibration messages, followed by the game starting, and finally displaying the game with the beetle.

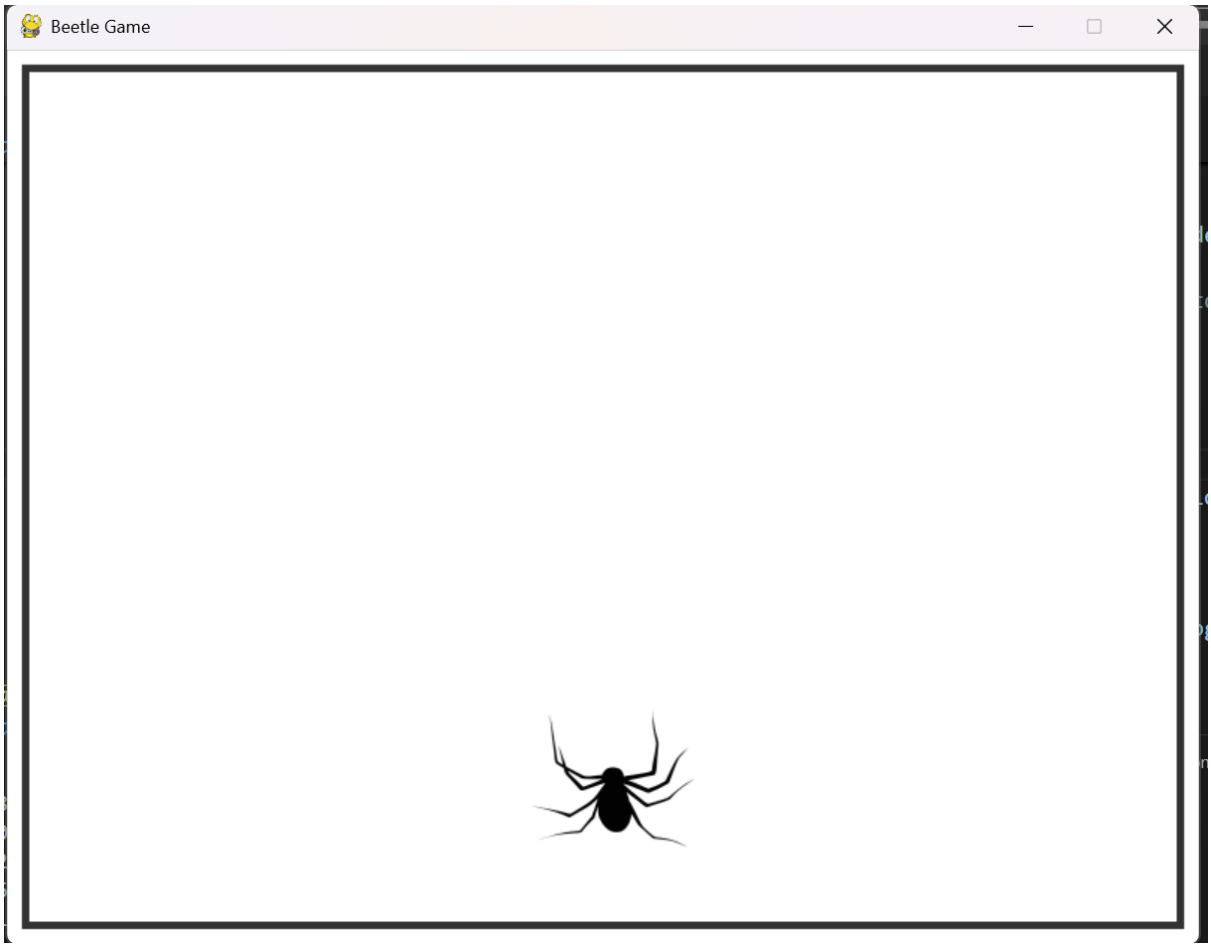


Fig. 9: EEG Beetle Game

6.10.8 7. *GUI*

https://youtu.be/BseTIdoimws

Overview

The **GUI** application is a Python-based tool designed to visualize real-time data streams from an Arduino device using the Lab Streaming Layer (LSL) protocol. The application connects to an LSL stream, retrieves multi-channel data, and plots it in real-time using the *pyqtgraph* library.

Features

Features	Description
1. LSL Stream Integration	• Automatically searches for and connects to available LSL streams. • Supports dynamic detection of the number of channels in the stream. • Displays connection status and channel count in the GUI.
2. Real-Time Data Visualization	• Plots real-time data for each channel in separate graphs. • Updates plots at a high frequency for smooth visualization.
3. Customizable GUI	• Built using *PyQt* and *pyqtgraph* for a responsive and interactive interface. • Features a clean layout with individual plots for each channel. • Includes a status bar to display LSL connection details.

A GUI window will appear that shows the data in real-time.

6.10.9 8. *EOG Keystroke Emulator*

https://youtu.be/ZJmUUtHJj08

Overview

The **EOG Keystroke Emulator** is a Python-based application designed to detect eye blinks using Electrooculography (EOG) signals and translate them into keystrokes. The application leverages the Lab Streaming Layer (LSL) protocol to acquire real-time EOG data, processes the signal to detect blinks, and simulates a spacebar press whenever a blink is detected. The application is built using the *tkinter* library for the graphical user interface (GUI) and integrates with *pyautogui* for keystroke emulation.


Real-Time Arduino Data
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
LSL Status: Connected

Fig. 10: GUI

Features

Features	Description
1. Real-Time EOG Signal Processing	• Connects to an LSL stream to acquire real-time EOG data. • Implements a low-pass filter to smooth the EOG signal for accurate blink detection.
2. Blink Detection	• Detects blinks by identifying peaks in the filtered EOG signal. • Uses a dynamic threshold based on the mean and standard deviation of the signal to distinguish blinks from noise. • Incorporates a refractory period to prevent multiple detections from a single blink.
3. Keystroke Emulation	• Simulates a spacebar press (*pyautogui.press('space')*) whenever a blink is detected. • Provides visual feedback by updating the GUI button color upon blink detection.
4. User-Friendly GUI	• Features a compact, movable popup window with a clean and intuitive interface. • Includes buttons for connecting to the LSL stream, starting/stopping blink detection, and quitting the application. • Displays an eye icon to represent the blink detection status.

A small window appears in the corner, displaying a *Connect* button. Once connected, a *Start* button becomes visible. Pressing the *Start* button initiates blink detection, and each detected blink triggers a spacebar key press.

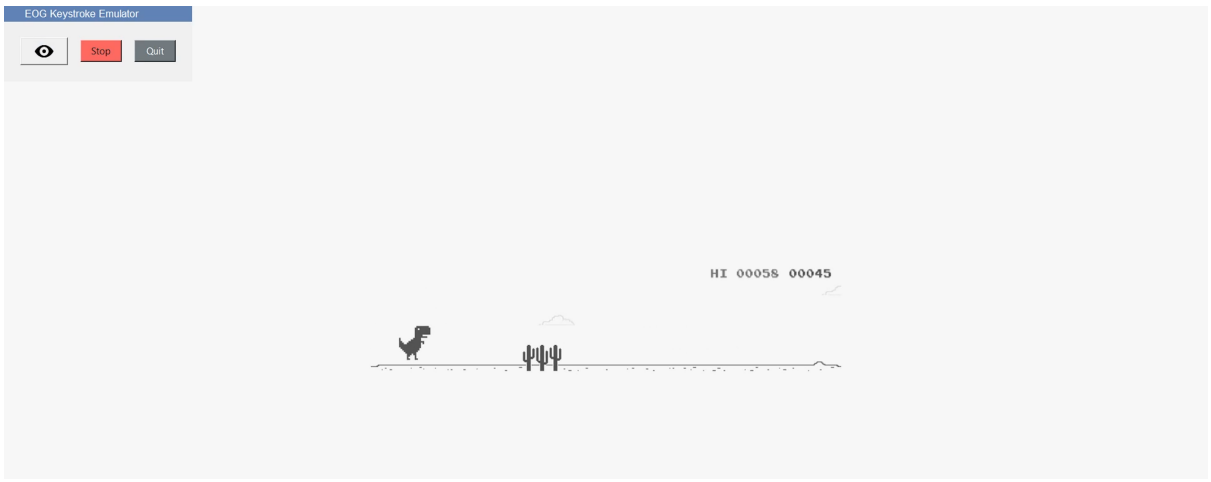


Fig. 11: Keystroke Emulator

6.10.10 9. *CSV Plotter*

https://youtu.be/wMnCOprRpZo

Overview

The **CSV Plotter** is a Python-based application designed to visualize data from CSV files. Built using the *tkinter* library for the graphical user interface (GUI) and *plotly* for data visualization, this tool allows users to load CSV files, select specific data channels, and generate interactive line plots.

Features

Features	Description
1. Load CSV Files	• Users can load CSV files containing data with a *Counter* column and multiple channels (e.g., *Channel1*, *Channel2*, etc.). • The application automatically detects the header row and skips any metadata above it.
2. Channel Selection	• A dropdown menu dynamically populates with available channels (e.g., *Channel1*, *Channel2*, etc.) from the loaded CSV file. • Users can select a specific channel to plot.
3. Interactive Data Visualization	• Utilizes *plotly* to generate interactive line plots for the selected channel. • **Plots include advanced features such as:** – *Zoom*: Zoom in to inspect specific data ranges. – *Pan*: Move across the plot to explore different sections. – *Autoscale*: Automatically adjust the plot scale to fit the data. – *Download Plot as PNG*: Save the generated plot as a high-quality PNG image. – *Hover-to-View Data Points*: Hover over the plot to view precise data values.
4. User-Friendly Interface	• Simple and intuitive GUI with buttons for loading files and plotting data. • Displays the name of the loaded CSV file for easy reference.

A small pop-up will appear, providing options to load the file, select the channel, and plot the data.

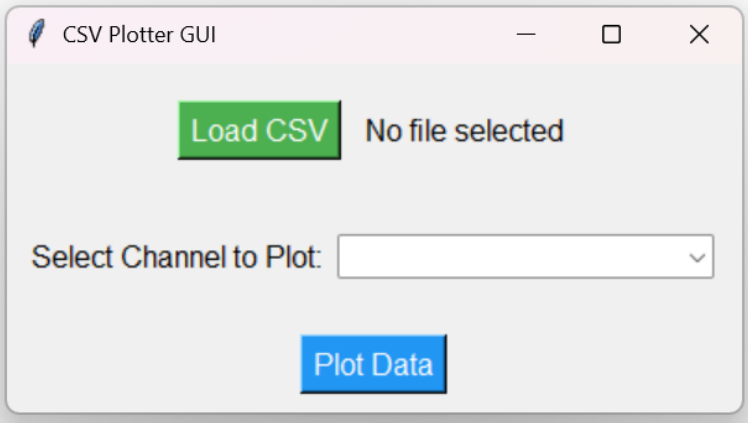


Fig. 12: CSV Plotter

6.10.11 10. *EOG Morse Decoder*

Overview

The **EOG Morse Decoder** is a Python-based application that enables users to input Morse code using eye movements detected via Electrooculography (EOG) signals. By moving your eyes left or right, you can generate dots and dashes, and by performing double or triple blinks, you can send characters or backspace. This application is ideal for hands-free communication and accessibility research.

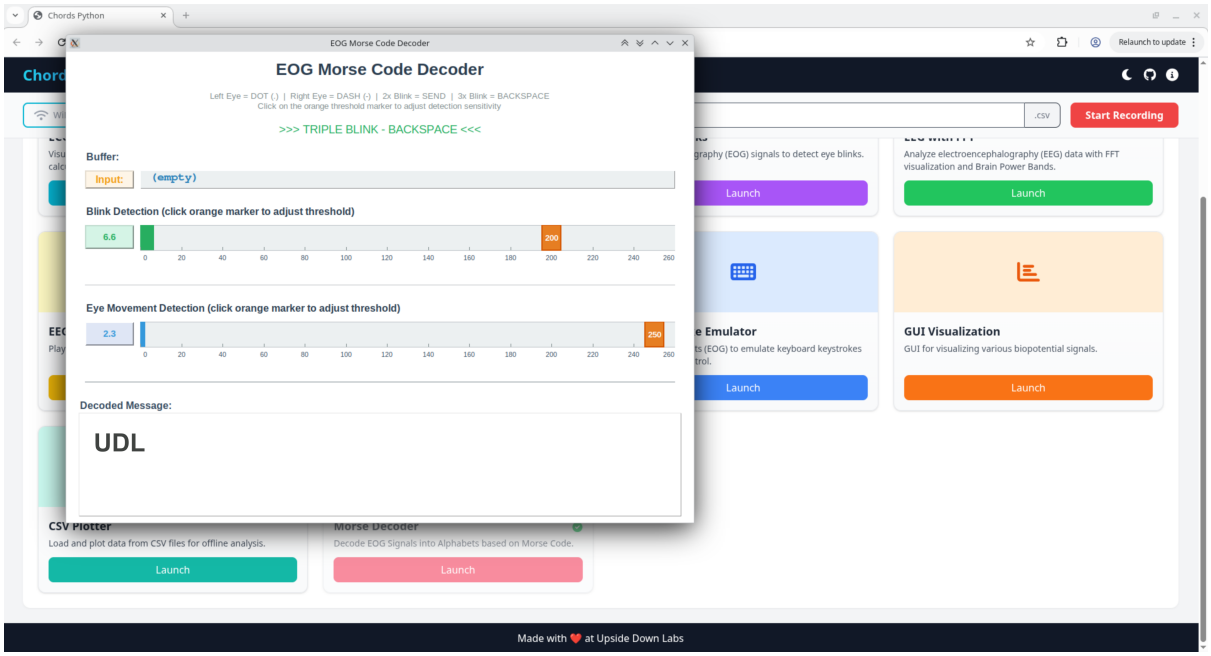


Fig. 13: Morse Decoder

6.10.12 Features

Features	Description
1. Real-Time EOG Signal Processing	• Connects to an LSL stream to acquire real-time EOG data. • Applies notch and bandpass filters to remove noise and isolate eye movement signals.
2. Eye Movement Detection	• Detects left and right eye movements by analyzing deviations from a dynamically calculated baseline. • Uses adjustable thresholds with interactive GUI sliders for real-time sensitivity adjustment.
3. Morse Code Input	• Left eye movement generates a dot (.) • Right eye movement generates a dash (-) • Double blink sends the current Morse sequence as a letter. • Triple blink acts as backspace to delete the last character.
4. Interactive GUI with Real-Time Visualization	• Displays the decoded message in a large, clear font using a Tkinter GUI. • Real-time bar graphs showing blink envelope and eye movement deviation. • Click-and-drag threshold adjustment on visual indicators. • Shows current buffer contents and detection status.
5. Adjustable Parameters	• Interactive threshold adjustment via clickable/draggable markers on visualization bars. • Separate thresholds for blink detection and eye movement detection. • Configurable debounce times for preventing false detections.

A GUI window will appear that allows you to input Morse code using eye movements and displays the decoded text in real-time.

Electrode Placement

Proper electrode placement is crucial for accurate EOG signal detection. The table below shows the recommended electrode configuration:

Electrode Pin	Description	Placement
A0P	Channel 1 Positive	Below eye
A0N	Channel 1 Negative	Above eye
A1P	Channel 2 Positive	On the left of left eye
A1N	Channel 2 Negative	On the right of right eye
REF	Reference	Bony part behind the ear

Note

- Channel 1 (A0P/A0N) detects vertical eye movements for blink detection
- Channel 2 (A1P/A1N) detects horizontal eye movements for left/right detection
- Proper skin preparation and electrode contact quality are essential for optimal performance

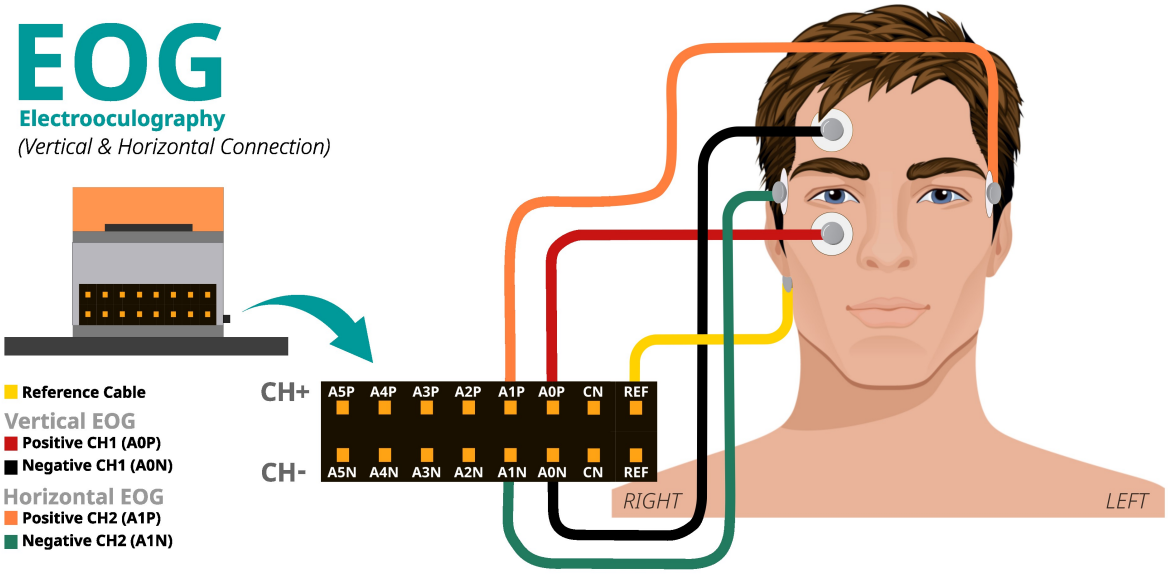


Fig. 14: Electrode Placement Diagram

How It Works

- **Signal Acquisition**: The application connects to an LSL stream and continuously reads vertical and horizontal EOG data from two channels.
- **Filtering**: Notch filters (50Hz), high-pass filters (1Hz), and low-pass filters (10Hz) are applied to remove powerline interference and isolate relevant frequencies.
- **Baseline Tracking**: A rolling baseline is maintained for horizontal EOG to adapt to individual signal characteristics.
- **Envelope Detection**: Vertical EOG signal envelope is computed for reliable blink detection.
- **Movement Detection**: Horizontal EOG deviations from baseline are analyzed to classify left and right eye movements.
- **Morse Code Input**:
  - Left eye movement → Dot (.)
  - Right eye movement → Dash (-)
  - Double blink → Sends the current Morse sequence as a letter
  - Triple blink → Backspace (deletes last character)
- **GUI Display**: Real-time visualization shows detection thresholds, signal levels, buffer contents, and decoded message.

Adjusting Detection for Your Neural Signals

The application provides an intuitive GUI for adjusting detection sensitivity in real-time:

Interactive Threshold Adjustment:

- **Blink Detection Threshold**: Click or drag the orange marker on the blink detection bar to adjust the threshold (range: 40-260). Higher values require stronger blinks.
- **Eye Movement Threshold**: Click or drag the orange marker on the eye movement bar to adjust the threshold (range: 40-260). Higher values require larger eye movements.

Visual Feedback:

- The blink detection bar shows the current envelope value (green when below threshold, red when above)
- The eye movement bar shows deviation from baseline (blue when below threshold, purple for left movement, red for right movement)
- Threshold markers display the current threshold value and can be dragged for adjustment

Timing Parameters (adjustable in code if needed):

- `BLINK_DEBOUNCE_MS` (default: 200ms): Minimum time between detecting separate blinks
- `DOUBLE_BLINK_MS` (default: 1000ms): Time window for double blink detection
- `TRIPLE_BLINK_MS` (default: 1500ms): Time window for triple blink detection
- `EYE_MOVEMENT_DEBOUNCE_MS` (default: 750ms): Minimum time between detecting separate eye movements

Tips for Optimal Performance:

- Make sure NPG Lite is charged and not connected to a charging laptop or an AC powered device
- Sit away from AC powered appliances
- Ensure proper skin preparation and electrode contact quality. Read the *Skin Preparation Guide* for detailed instructions.
- Start with default threshold values and adjust using the GUI sliders based on your signal strength
- Ensure proper electrode placement for clean vertical and horizontal EOG signals
- Maintain a neutral gaze position during baseline calculation at startup
- Use the real-time visualization bars to monitor signal quality and adjust thresholds accordingly

Troubleshooting

- If movements are not detected reliably, use the GUI to adjust thresholds in real-time by clicking/dragging the orange markers
- Ensure electrodes are placed correctly: vertical EOG above/below one eye, horizontal EOG at outer corners of both eyes
- Remain relaxed and maintain a neutral gaze position during the initial baseline calculation
- Check the real-time visualization bars to ensure signals are being detected properly
- If blinks or movements are too sensitive, increase the respective threshold using the GUI sliders
- If blinks or movements are not sensitive enough, decrease the threshold using the GUI sliders

6.11 Create Custom application

You can create custom applications using the provided framework by following these steps:

1. Configure Application Metadata:

Edit the `apps.yaml` file in the `config` folder with your application details:

```
- title: "Your Application Title"
  icon: "path/to/your/icon.png"
  color: "your_hex_color"
  script: "path/to/{app_name}.py"
  description: "Brief description of your application"
  category: "Your Category"
```

Add this as a new entry in the YAML list. Replace all placeholders with your actual application details.

Note

- The `icon` path should be relative to the application root directory
- `color` should be in HEX format (e.g., "#FF5733")
- The `script` path should point to your Python file

2. Create application script:

Create a new Python script in the main directory with your application name. The script should contain:

- LSL stream connection handling to receive device data
- User interface components using PyQt5/PyQtGraph
- Data processing logic for incoming signals

Tip

Use the existing applications in the repository as reference implementations for: - lsl setup and data acquisition - Advanced UI layouts - Signal processing examples - Performance optimization