

Manual de usuario para extracción de señales EEG con marcadores de eventos

Creado por: Alejandro Jesús Perdomo Cely

Preliminares:

Los siguientes son los links a los programas a instalar para el correcto funcionamiento del manual, de donde cada uno tendrá su paso a paso interno de instalación:

- FTDI Para Windows 7 a 11: Drivers que permiten la correcta conexión del dispositivo USB del casco con el computador.

https://ftdichip.com/wp-content/uploads/2021/08/CDM212364_Setup.zip

- OpenBCI: Interfaz cerebro-computador que permite visualizar las señales electroencefalográficas captadas por el casco.

<https://openbci.com/downloads>

- LabRecorder: Programa que entrelaza las cadenas de datos entre diferentes transmisores de tipo LSL.

<https://github.com/labstreaminglayer/App-LabRecorder/releases>

- Psychopy: Aplicación que genera estímulos visuales y auditivos, al igual que recibe información de periféricos como mouse, teclado o hasta mandos de consolas de videojuegos.

<https://www.psychopy.org/download.html>

- Eeglab de Matlab: Programa de Matlab que permite obtener las señales EEG junto con marcadores a partir de programas con extensión .xdf, aquellos generados por el programa LabRecorder.

<https://la.mathworks.com/matlabcentral/fileexchange/56415-eeglab>

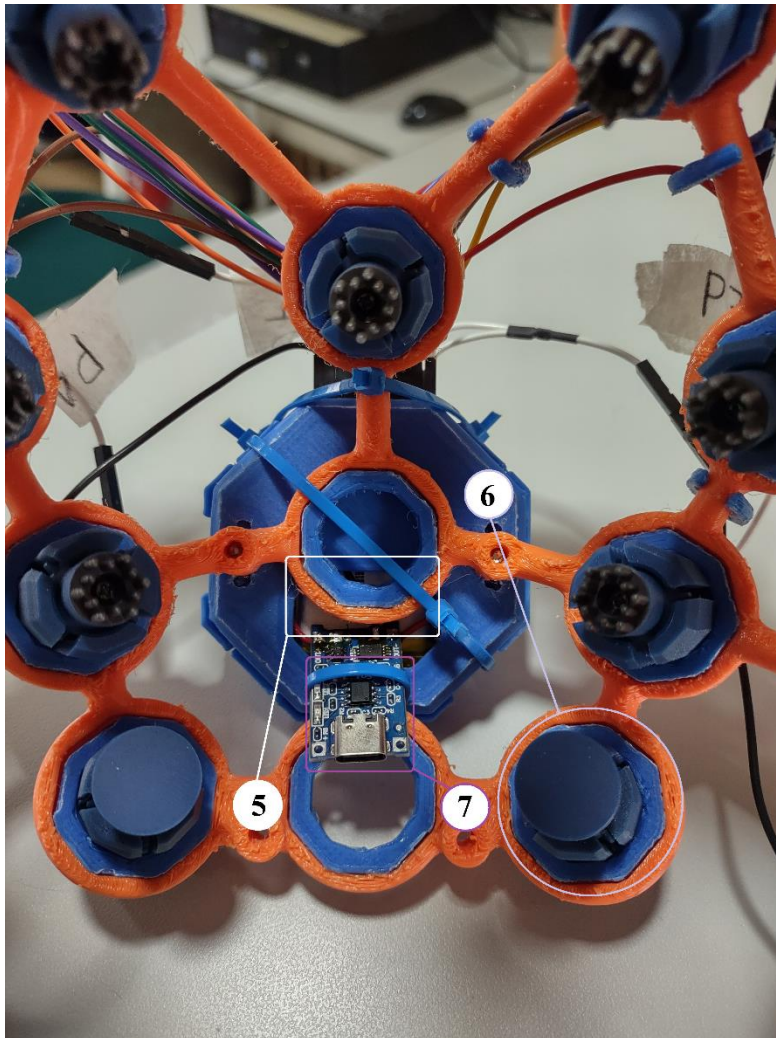
Una vez se han instalado correctamente todos los programas, se procede con el paso a paso.

1. Sacar de la caja:

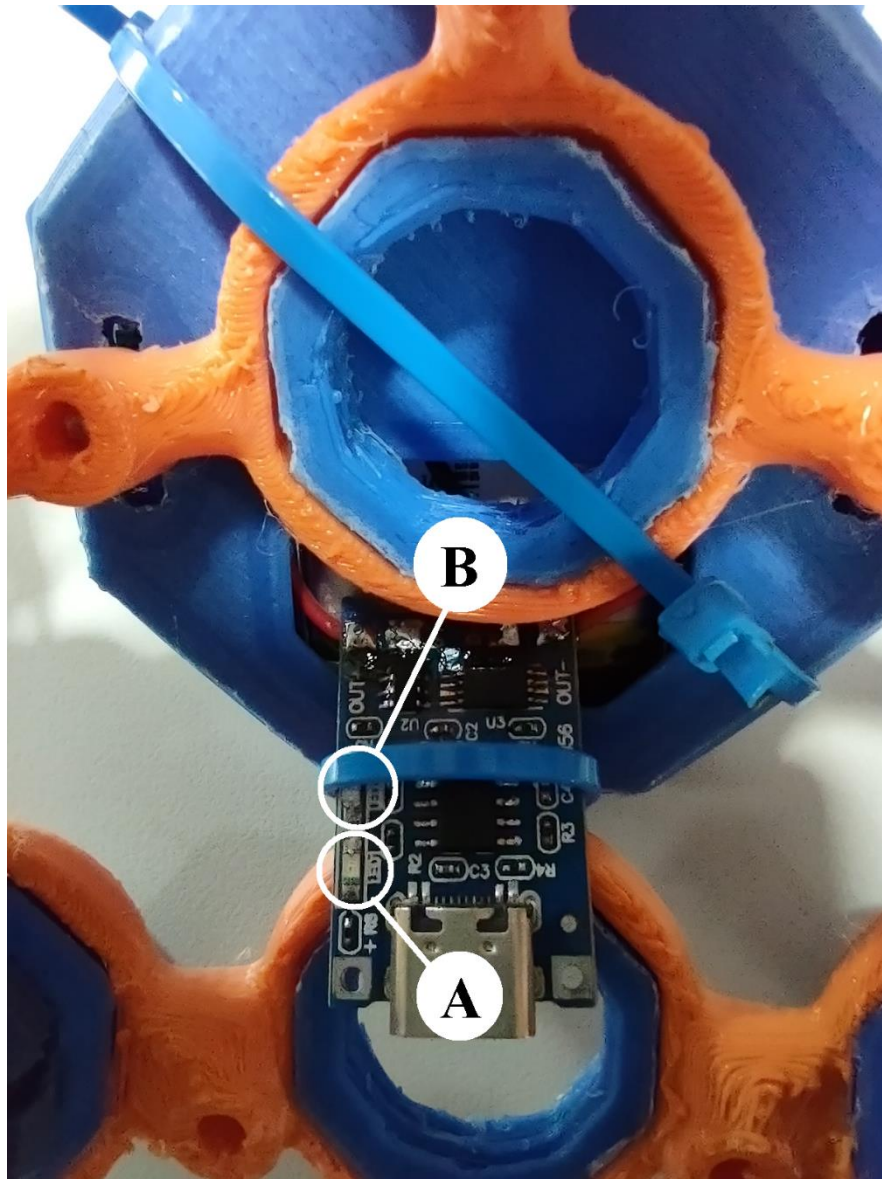
Localice la caja del casco en el inventario de la universidad conocido como “Casco EEG de OpenBCI”, luego remueva la tapa y cuidadosamente retire el casco, apoyándose de las correas de color negro que están sujetas al dispositivo. Proceda a sacar la pequeña caja de cartón, el cual contiene los elementos complementarios del casco, como el conector USB de comunicación, y repuestos de electrodos sobre cuero cabelludo y sobre piel directa, al igual que clips para los cables.

2. Cargar batería con conector tipo C:

Si el casco no ha sido usado en un tiempo, es recomendable cargar la batería antes de iniciar a usarlo, ya que con niveles bajos de carga el casco no opera correctamente, perdiendo información transmitida y alterando datos. Conecte un cargador tipo C de cinco voltios a la parte trasera interna del casco (7):

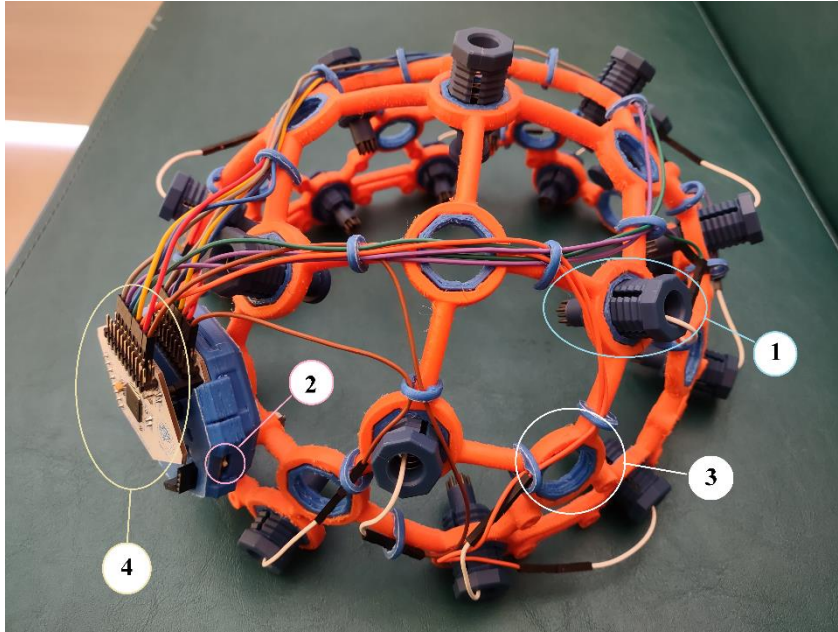


En el cual se iluminará un LED de color rojo indicando que está cargando (A), y se encenderá el LED de color azul cuando haya terminado de cargar (B):



3. Mover interruptor derecho:

Una vez se encuentre cargado la batería del casco, puede proceder a encenderlo, a través del interruptor que se localiza en la parte derecha exterior del encapsulado de circuitos (2):



Este tiene tres posiciones, el encendido sin almacenar en Micro SD (arriba), apagado (centro) y encendido almacenando en Micro SD (abajo). Si el usuario desea, puede insertarle una tarjeta Micro SD a la placa, a través de la ranura que se encuentra en la diagonal izquierda inferior exterior del encapsulado, para de esta forma encender el casco con almacenamiento en la Micro SD a través del interruptor posicionado hacia abajo. Si no se desea almacenar la información en una Micro SD, se mueve la posición del interruptor hacia arriba. En cualquier caso de encendido, deberá alumbrar un LED azul indicando que el casco está encendido.

4. Poner el casco sobre la cabeza:

Posicione el casco en la orientación adecuada antes de ponerlo sobre la cabeza, donde el encapsulado de circuitos de color azul debe ir en la parte posterior, y las correas negras deben descender por gravedad.



Si las correas están unidas, presione el pequeño botón de plástico en la chapa que las une, para soltarlas. Ahora, sujete el casco de cualquier parte de estructura de color naranja, y póngalo sobre la cabeza, y en el momento en el que los electrodos inicien a hacer contacto sobre el cabello, empiece a realizar movimientos circulares ligeros con el eje de rotación paralelo a la columna vertebral, mientras sigue bajando y reposando el casco sobre la cabeza, hasta el punto de ajustar considerablemente, pero sin presionar demasiado, ya que esto último puede causar dolor de cabeza si se usa el dispositivo durante mucho tiempo.

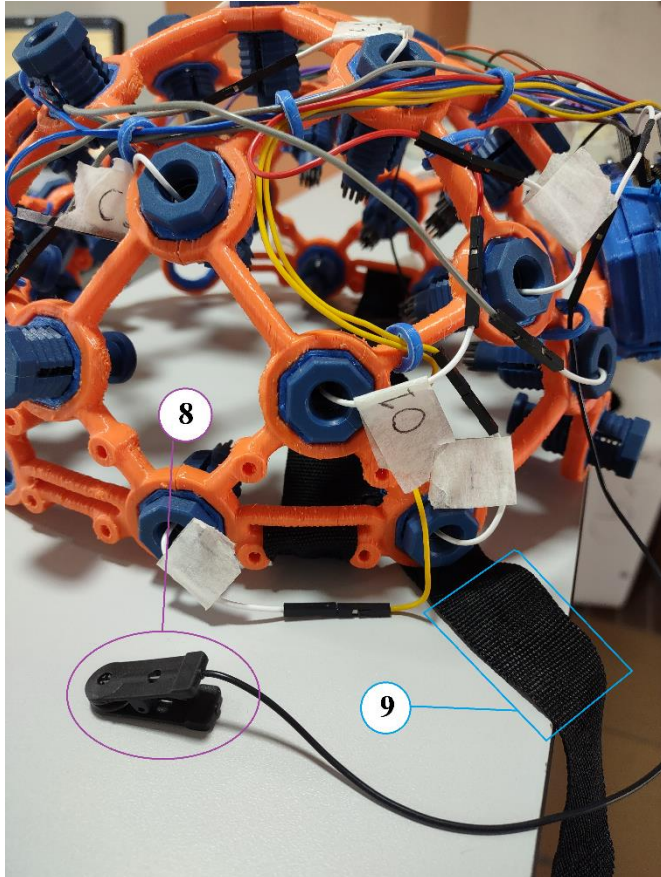


Una vez esté puesto el casco en su totalidad, abróchelo con las correas alrededor de la mandíbula, manteniendo la orientación adecuada para que reposen planamente sobre las orejas y debajo del

mentón. Es recomendable remover aretes y/o piercings durante las pruebas con el fin de reducir capacitancias o inductancias parásitas.

5. Pinzas de tierra y referencia en las orejas:

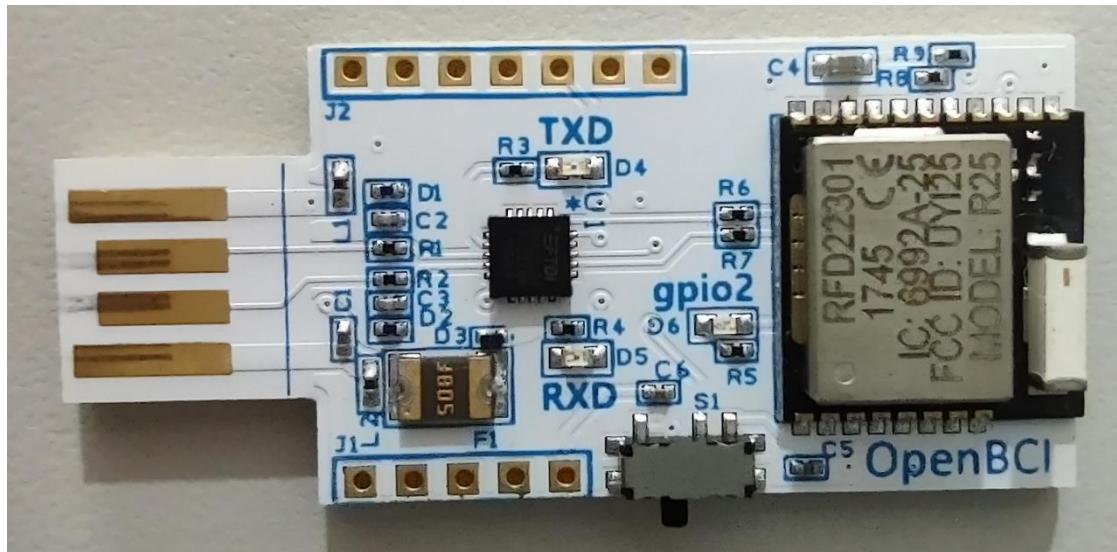
El casco contiene dos pequeñas pinzas de color negro, uno en cada lado (8).



Abra una pinza a la vez y ciérrelo en cada oreja del paciente, en el que la pinza esté más cerca de la respectiva oreja, donde es recomendable ponerlo en el lóbulo de la oreja, para mayor comodidad y resultados más replicables. Estos tienen el fin de generar los puntos de tierra y referencia del circuito de electrodos, obteniendo así las lecturas correctas de cada zona exterior del cráneo.

6. Dongle en PC:

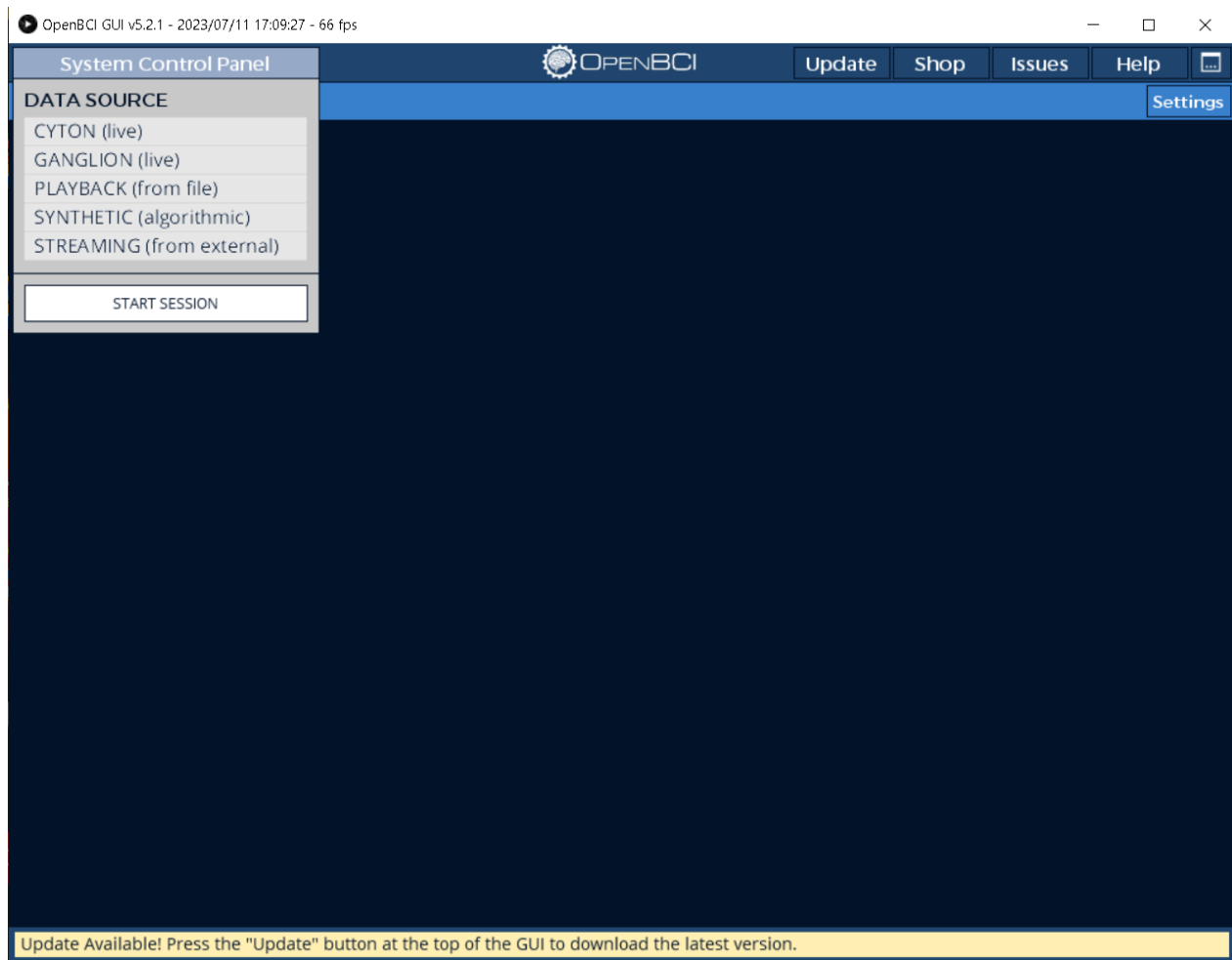
Dentro de la caja de cartón encontrará una pequeña bolsa antiestática conteniendo el dispositivo USB, también conocido como “Dongle”, encargado de recibir la información transmitida del casco e ingresarla a su computador.



Inserte el dispositivo en cualquier puerto del computador a usar, del cual se iluminará un LED azul, indicando su conexión.

7. **Iniciar OpenBCI en el PC:**

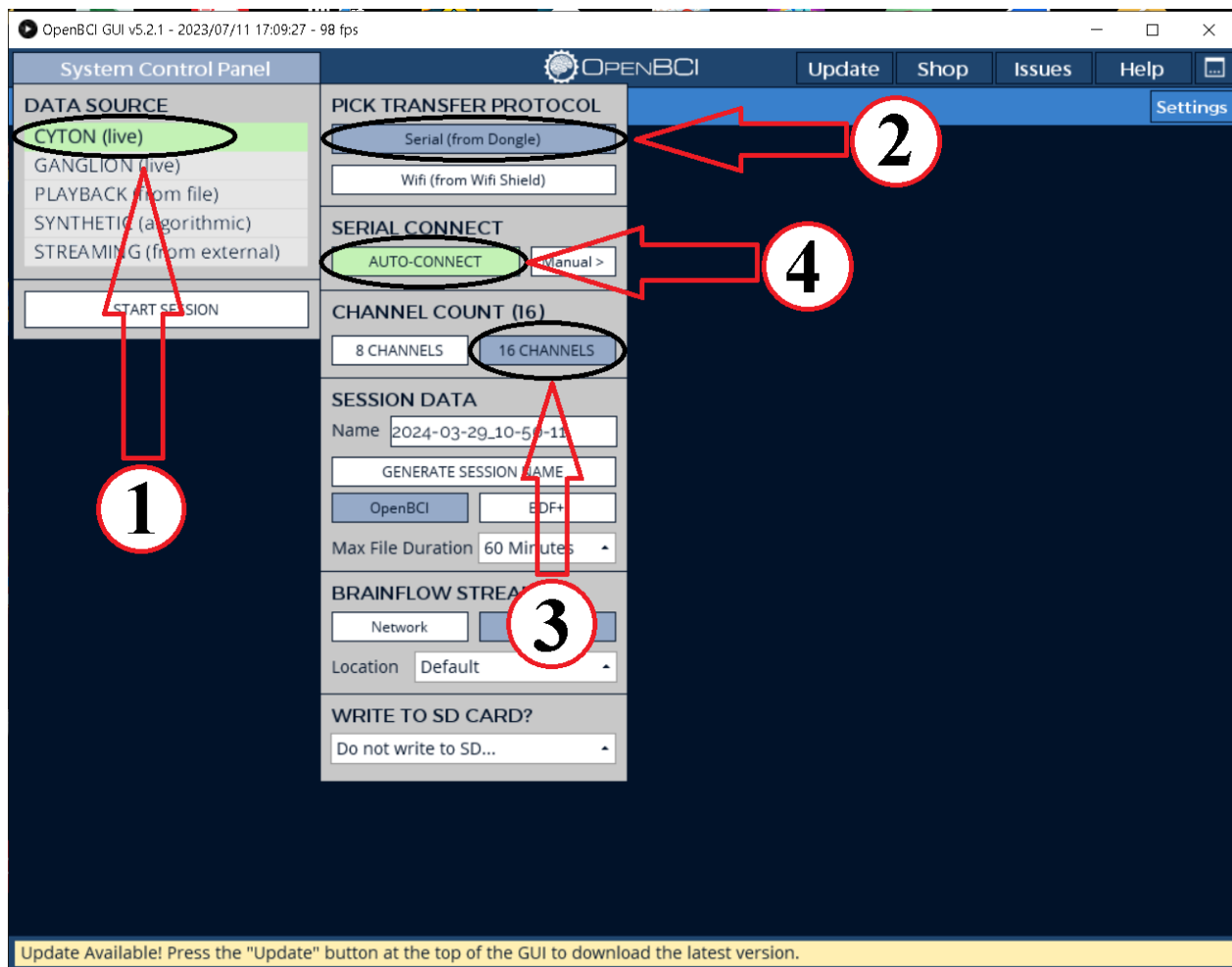
Presione click derecho sobre el icono del programa de OpenBCI, y seleccione la opción “Ejecutar como administrador”, luego presione en “Sí” y espere a que se ejecute el programa. Se le presentará una interfaz de color azul oscuro, donde en la parte inferior izquierda se le informará si existen actualizaciones disponibles para la aplicación.



Además, esta ubicación de información le mostrará más adelante información relevante sobre los diferentes procesos que se llevarán a cabo en la aplicación.

8. Conectar con casco a 16 electrodos:

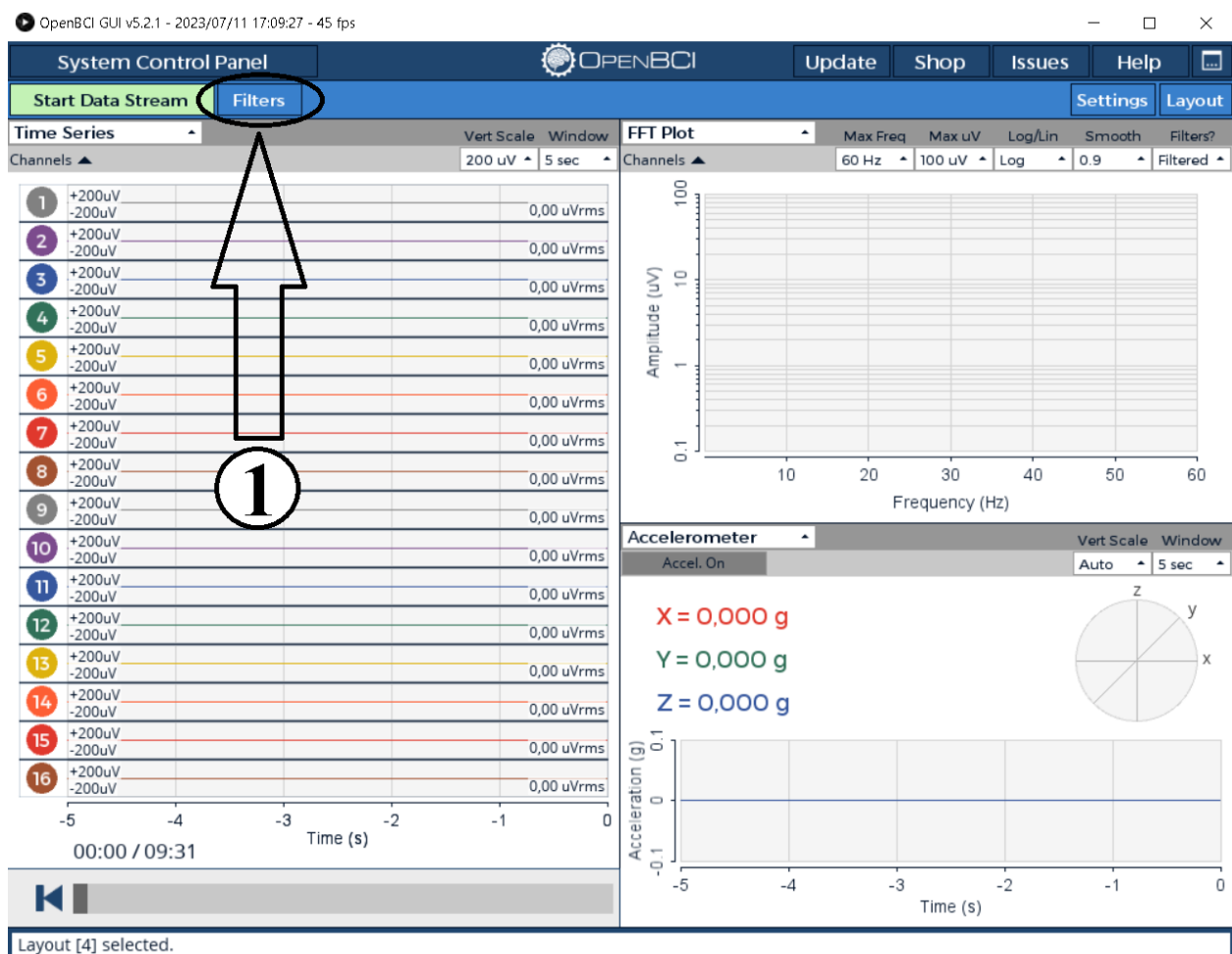
Bajo el cuadro con texto “System Control Panel” localizado en la parte superior izquierda, presione el que dice “CYTON (live)” (1), luego bajo “PICK TRANSFER PROTOCOL” seleccione “Serial (from Dongle)” (2) y bajo “CHANNEL COUNT (8)” presione “16 CHANNELS” (3) por el hecho de que el casco tiene la ampliación Daisy, duplicando la cantidad original de electrodos. Si lo desea, puede cambiar el nombre de la sesión de datos, la ubicación de almacenamiento, y si desea escribir la información o no en la Micro SD (esto último si se tiene insertado una tarjeta de almacenamiento en el casco). Es recomendable no alterar los demás ítems.



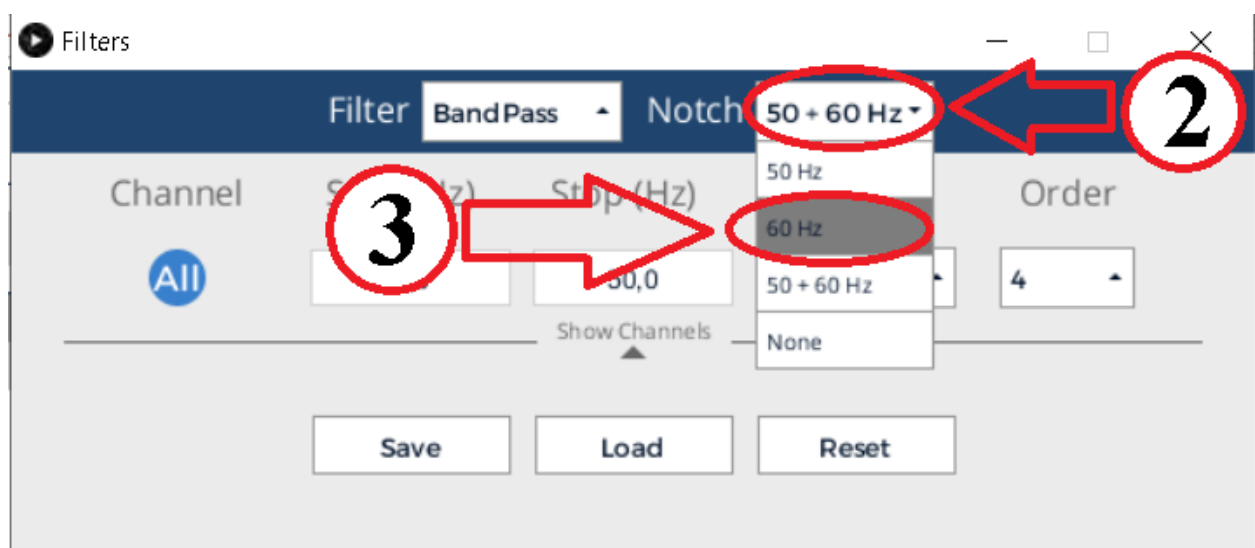
Una vez esté listo, presione el botón verde que dice “AUTO-CONNECT” (4) y espere unos segundos a que el sistema detecte y se conecte al casco de forma inalámbrica. Si se va conectando de manera exitosa, se verá que parpadean los LEDs verde y rojo en el Dongle, los cuales son los indicadores de transmisión y recepción de datos respectivamente.

9. Filtros y opciones de visualización:

Al ingresar a la interfaz gráfica de OpenBCI, se observará una ventana con los 16 electrodos en tiempo, en frecuencia, y el acelerómetro en los tres ejes, integrado dentro de la placa de circuitos. Sin embargo, inicialmente se encuentra todo pausado, así que se alterarán unas configuraciones antes de proseguir.

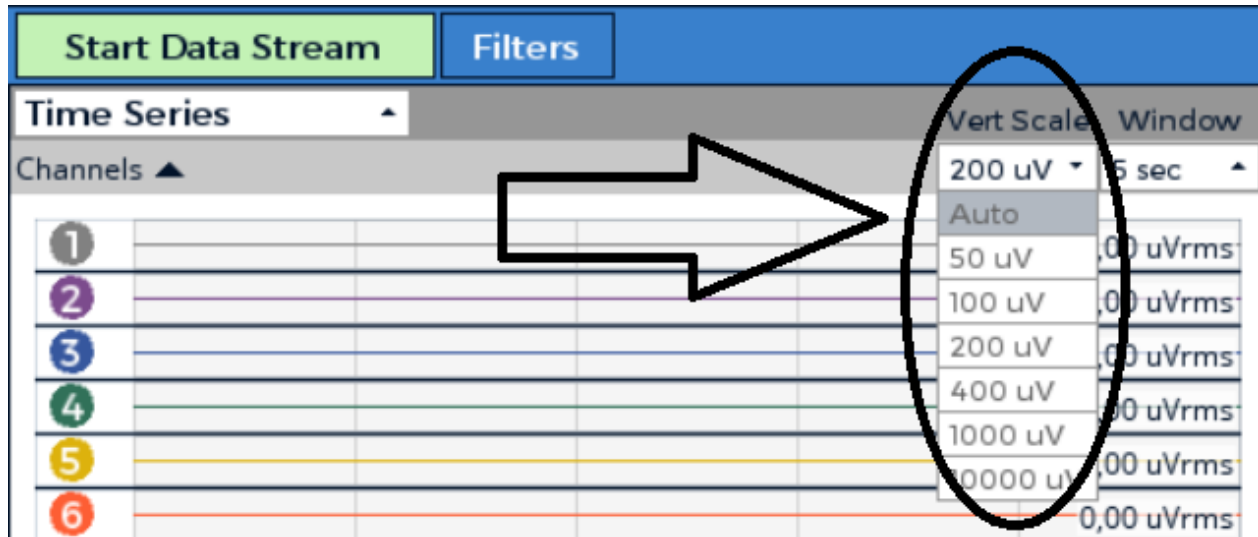


Primero, en la parte superior izquierda, se presionará en la opción “Filters” (1), y se abrirá una ventana para ajustar los filtros rechaza banda y su orden y tipo de filtro:



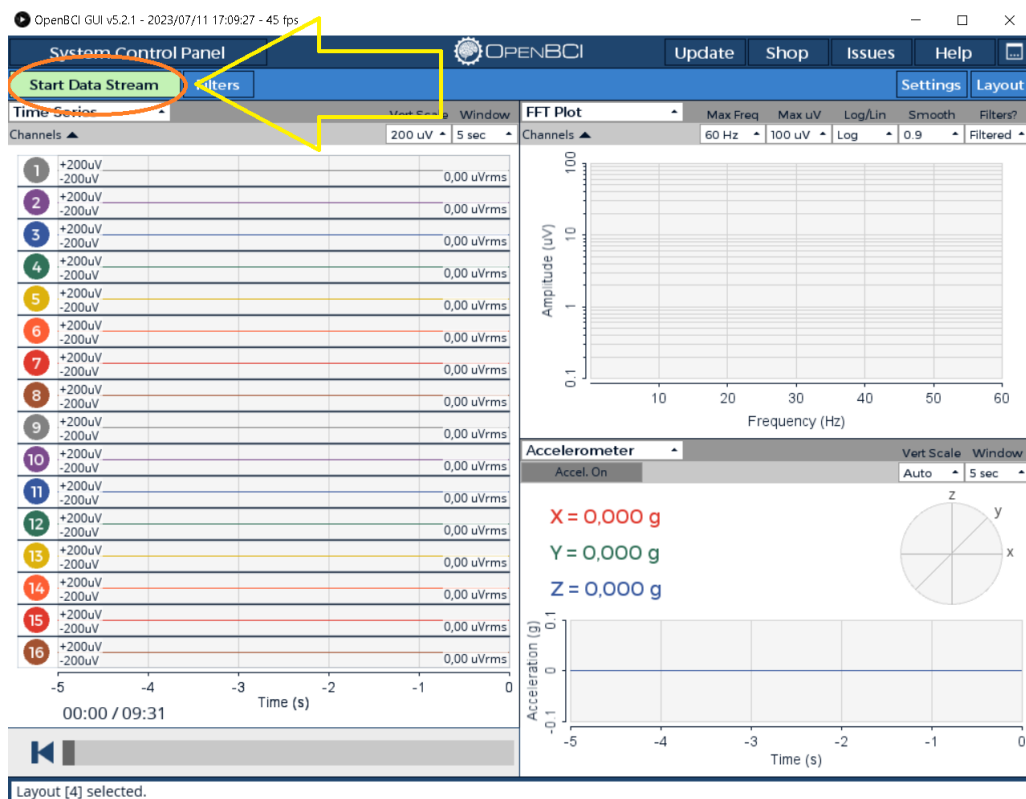
Se presiona el botón al lado derecho de la palabra “Notch” (2) para cambiar las frecuencias de rechazo a solamente “60 Hz” (3) y luego se cierra la ventana. Los cambios se guardan automáticamente.

Después, para visualizar de una mejor forma las señales que se captarán, al lado derecho del botón de “Filters” bajo el texto “Vert Scale” se le da click en el botón para seleccionar “Auto” para que las señales se observen sin importar su alto o bajo voltaje.



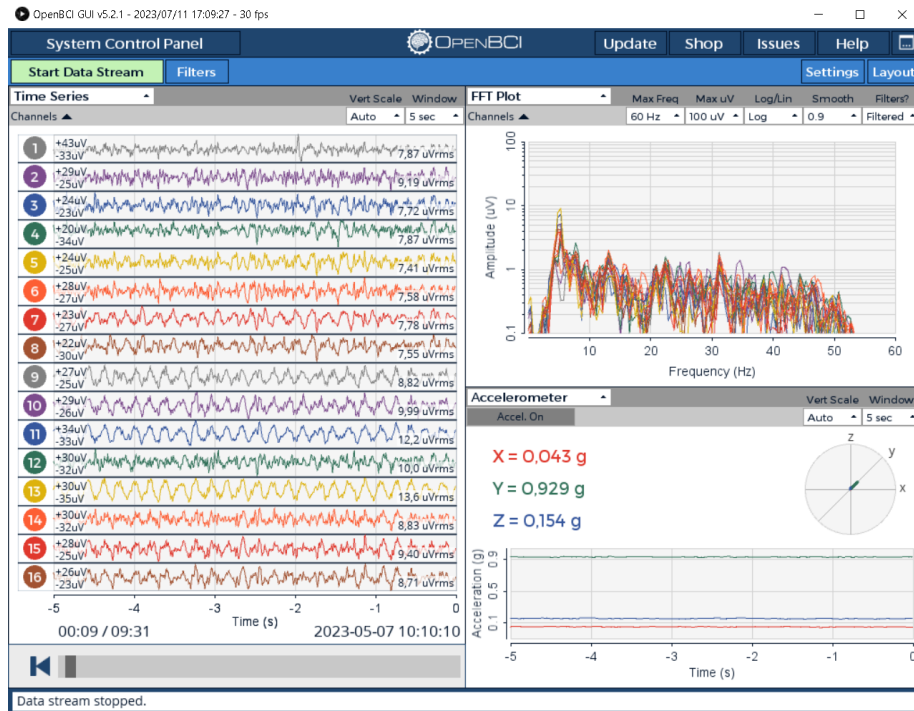
10. Iniciar recepción de datos:

Una vez se hayan terminado de realizar los ajustes, se presiona el botón verde de la parte superior izquierda que dice “Start Data Stream”:



Si todo funciona de manera correcta, se iniciará a recibir la información del casco, en el que comienza con señales en tiempo con altos valores de voltaje, y se estabilizarán, al igual que altas potencias en frecuencia hasta descender y mantenerse estable.

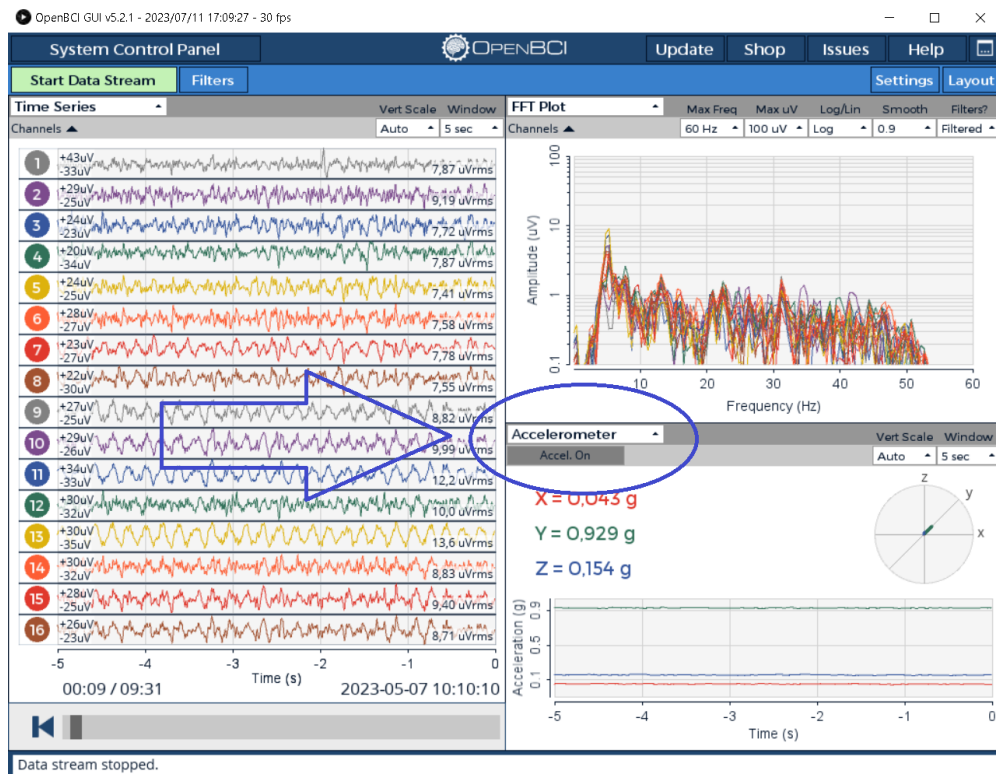




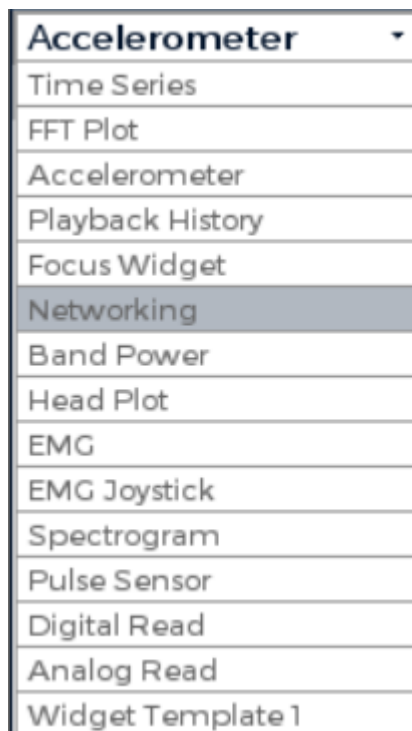
El LED rojo del Dongle conectado al PC se mantendrá iluminado, indicando la continua recepción de datos de parte del casco. Para mejores resultados de transmisión de datos, es recomendable mantener una distancia aproximada de medio metro como mínimo entre el PC y el paciente.

11. Inicio de transmisión por LSL:

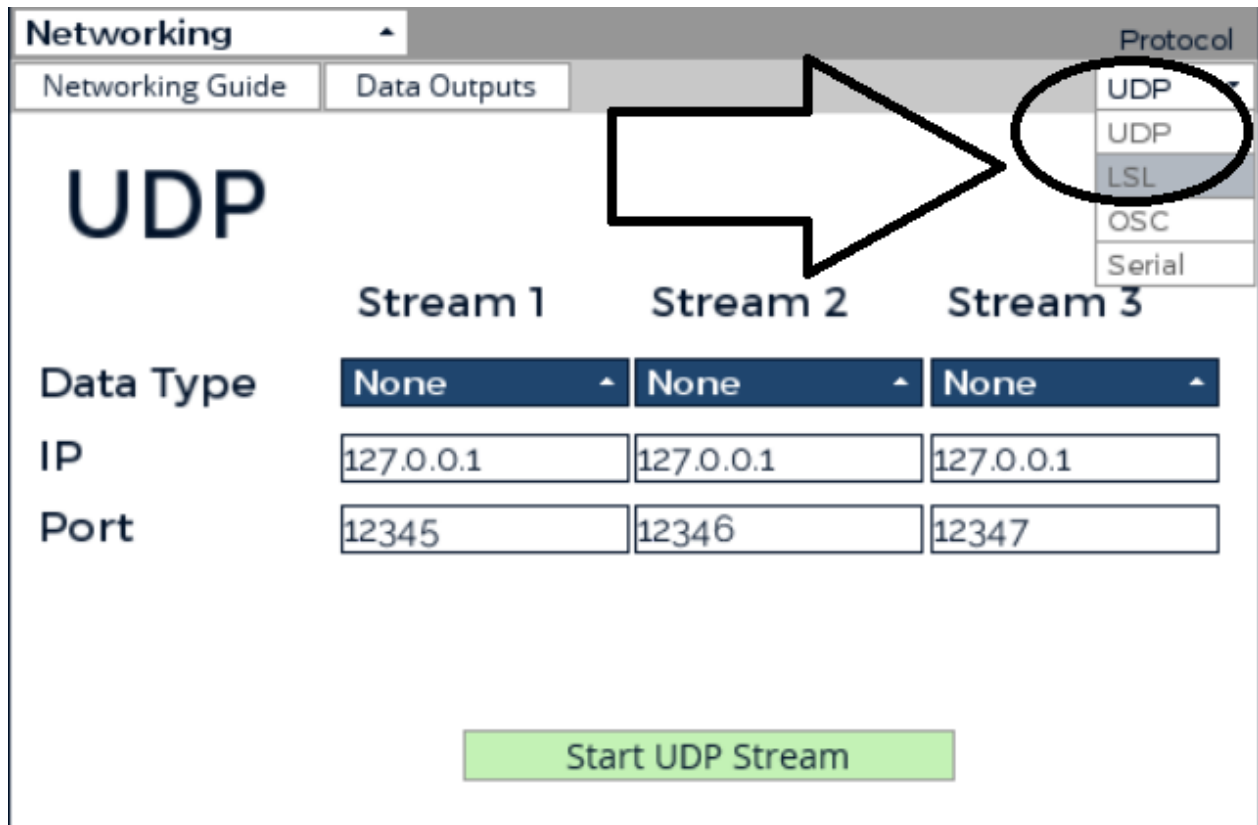
Ahora, para poder realizar el procedimiento de transmisión de datos desde OpenBCI hasta un sistema que procese la información como Matlab, se hará a través de la comunicación LSL, que significa “Lab Streaming Layer”. Se inicia por seleccionar cualquiera de los tres tipos de datos que nos muestra OpenBCI, ya sea “Time Series”, “FFT Plot”, o el recomendable, aquel que dice “Accelerometer”:



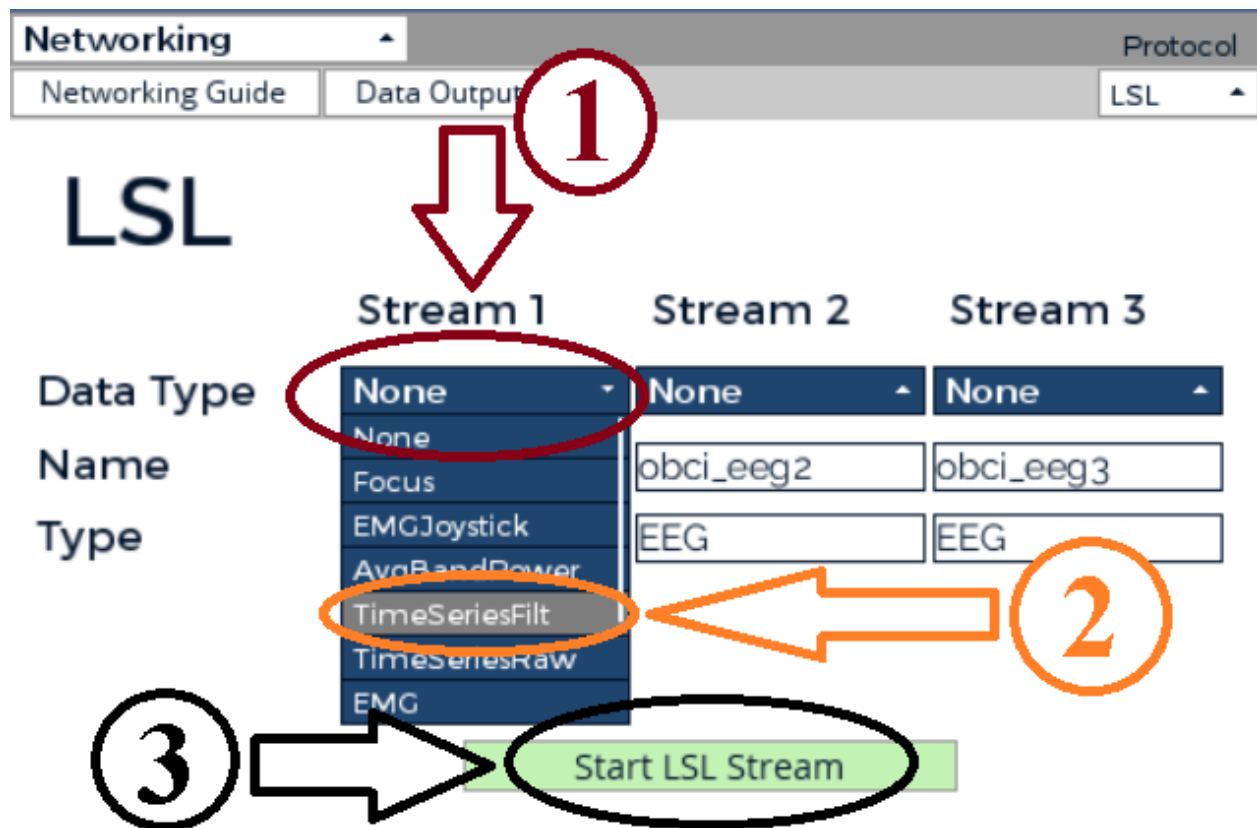
Donde al darle click se desplegará la pestaña mostrando múltiples opciones. Se selecciona aquel que dice Networking:



Luego iniciará por mostrar el protocolo de comunicación UDP, el cual para este estudio se cambiará a LSL, así que, en la parte superior derecha de la sección, debajo de “Protocol” se presiona para cambiar a LSL:



Después, se ajusta el Stream 1 para cambiar el tipo de datos a transmitir (1) para seleccionar “TimeSeriesFilt” (2), con el fin de transmitir a través de LSL las señales en tiempo de los 16 electrodos, con los filtros que le hemos aplicado anteriormente.

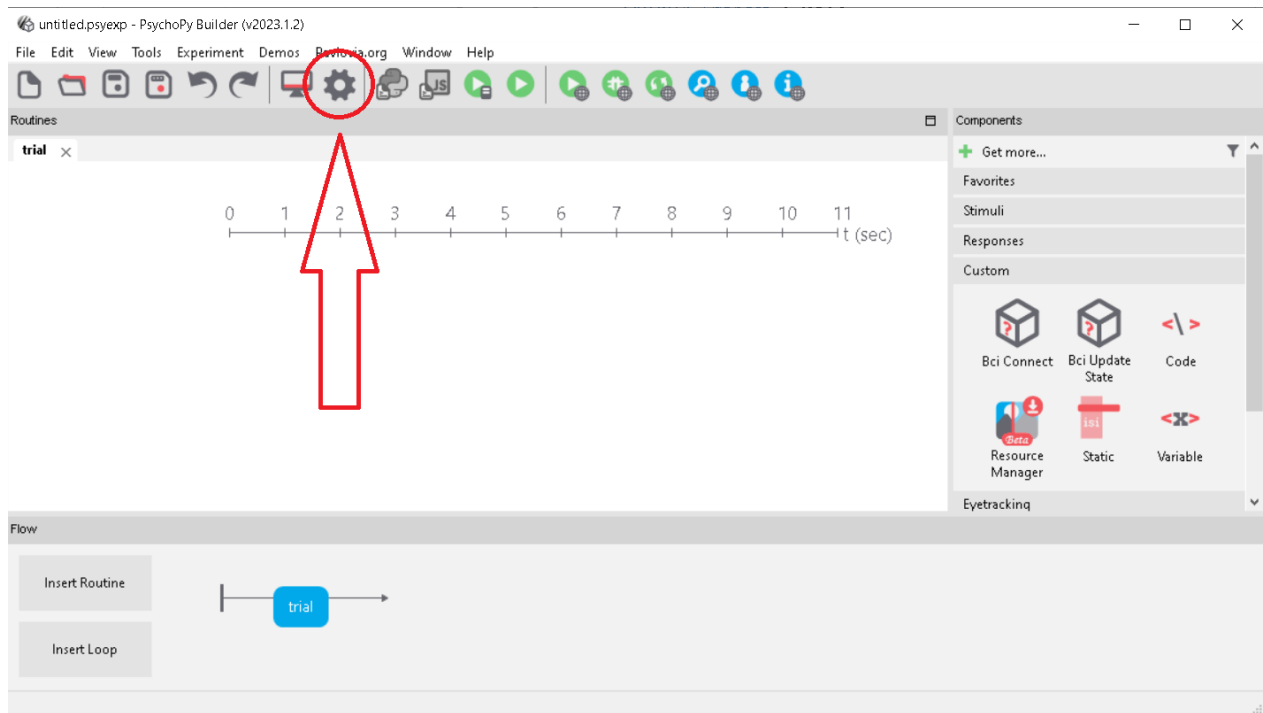


Y con esto se presiona el botón verde de la parte central inferior de la sección que dice “Start LSL Stream” (3) para activar la transmisión de datos. Tenga en cuenta que es posible transmitir hasta tres tipos de datos simultáneamente, pero para este estudio que se basa en señales en tiempo solo se activará el primer puerto de datos.

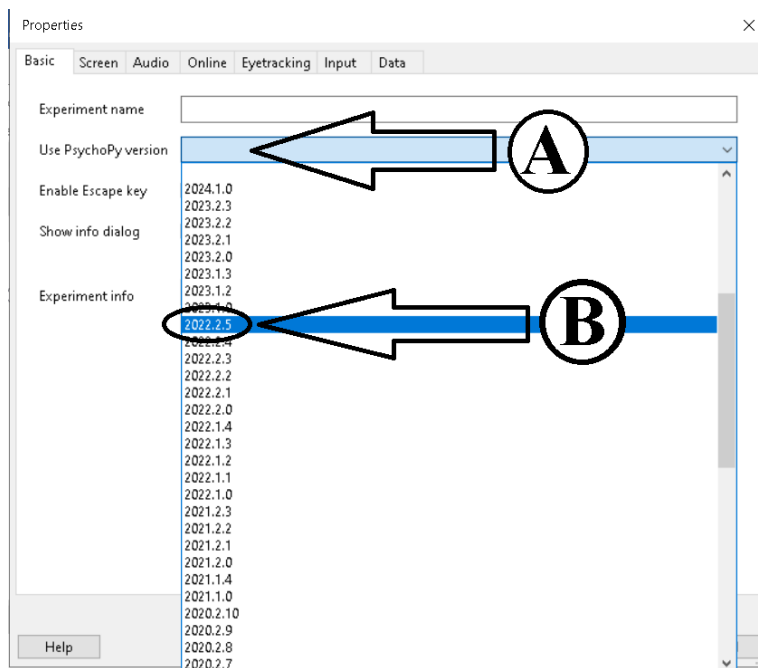
12. Marcadores en PsychoPy:

Para esta sección, se suministrarán los fragmentos de código, clave para generar los marcadores que establecerán las ubicaciones de los estímulos visuales y auditivos, los cuales, al unirlos más adelante con las señales en tiempo captadas por el casco, permitirán crear el archivo con marcadores y señales unidos. Además, adjunto se encuentra el programa creado para el proyecto, por si se desea modificar, o tener de base para el estudio. Si se desea conocer en profundidad el funcionamiento de Psychopy, es recomendable consultar tutoriales acerca de las herramientas básicas para crear un generador de estímulos.

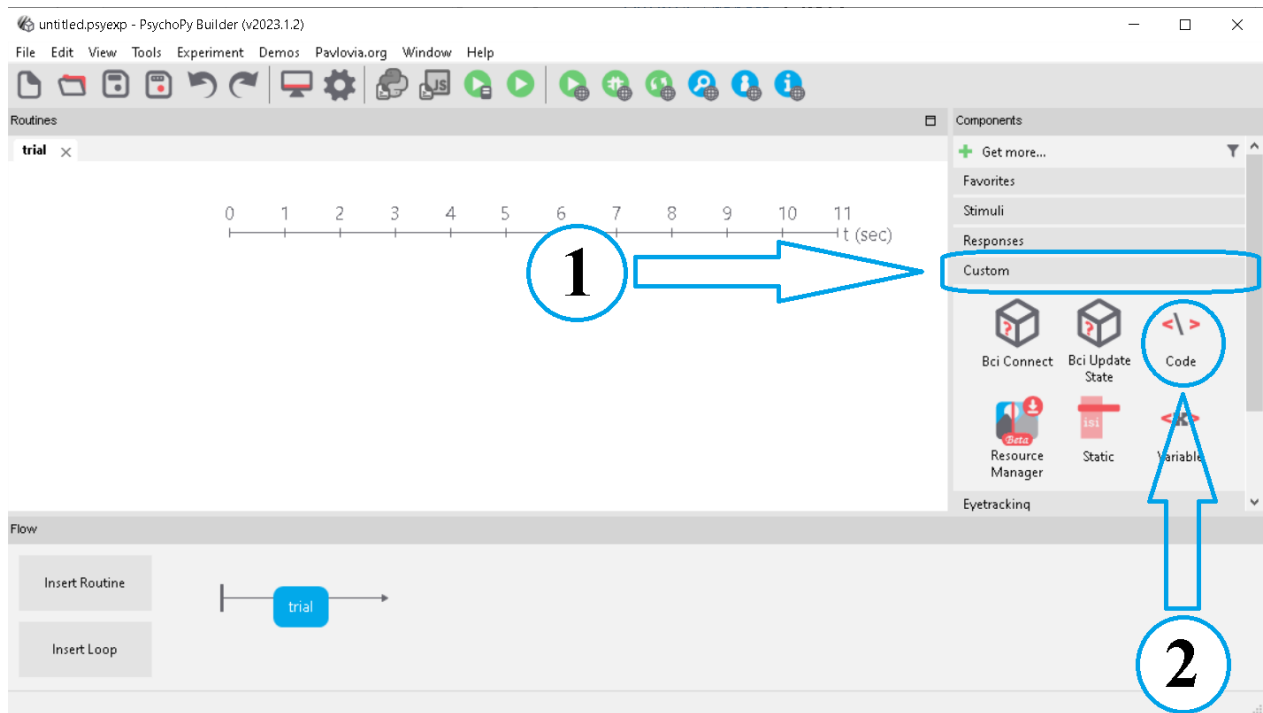
Debido a que Psychopy presenta inconvenientes con la emisión de tonos de sonido con las versiones más recientes, se iniciará por configurar la versión apta para pruebas auditivas (2022.2.5). Primero se ingresa en la herramienta de configuración de experimento:



En “Use PsychoPy version” (A) se da click y luego se busca la versión 2022.2.5 (B):



Ya con esto, para crear un nuevo generador de estímulos visuales y auditivos, se abrirá la aplicación de PsychoPy, al lado derecho en el cuadro “Custom” (1), en el primer bloque de rutina, se insertará un segmento de código del bloque “Code” (2):



El cual en la parte de “Before Experiment” el código que está a continuación:

Name
code
Code Type
Py
☐ disabled

Before Experiment*
Begin Experiment
Begin Routine
Each Frame
End Routine
End Experiment

```

1 from pylsl import StreamInfo, StreamOutlet
2 import random
3 info = StreamInfo(name = 'TriggerStream', type = 'Markers', channel_count = 1,
4 outlet = StreamOutlet(info)

```

El código completo es el siguiente:

from pylsl import StreamInfo, StreamOutlet

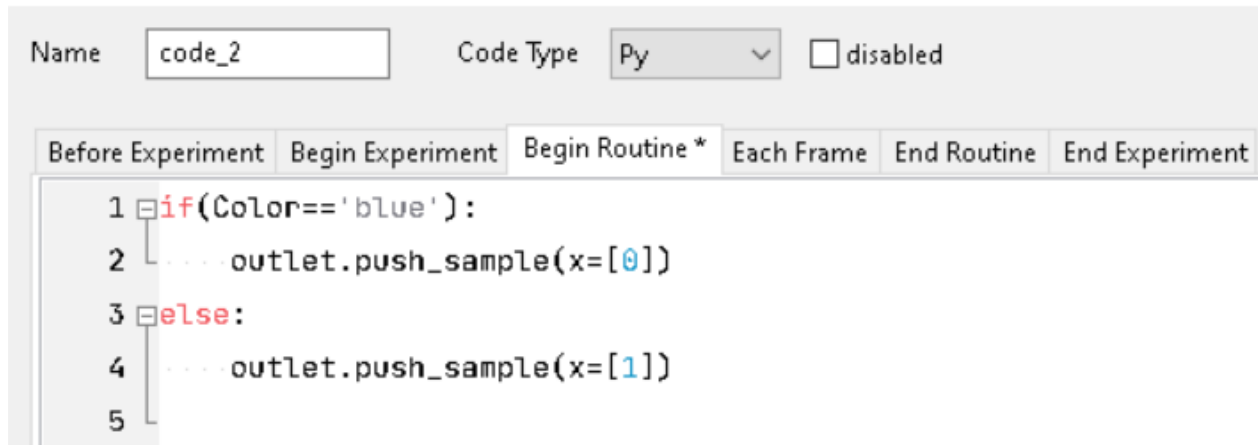
import random

info = StreamInfo(name = 'TriggerStream', type = 'Markers', channel_count = 1, channel_format = 'int32', source_id = 'example')

outlet = StreamOutlet(info)

Este primer código permite inicializar la conexión a través de LSL, y es obligatorio en todo inicio de flujo del programa generador de estímulos. Desde ahí, el usuario puede crear los estímulos que desee, visuales y/o auditivos, lo importante es que use un archivo de Excel para generar las características de los estímulos, para mantener las variables de los marcadores. En el momento de crear el segmento que contendrá la aparición de los estímulos, se insertará otro bloque que contendrá en la parte de “Begin Routine” el siguiente código:

code_2 Properties



Con el siguiente código:

```
if(Color=='blue'):
```

```
    outlet.push_sample(x=[0])
```

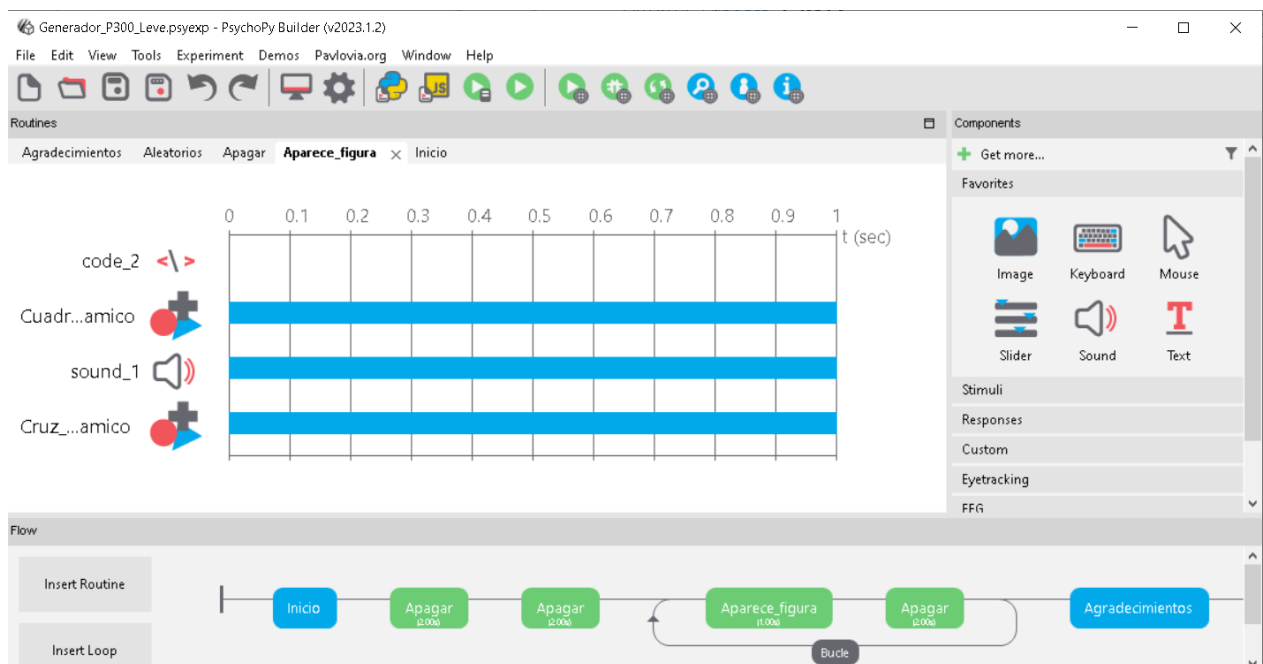
```
else:
```

```
    outlet.push_sample(x=[1])
```

El color del estímulo que se define al inicio está basado en la variable llamada “Color” de la hoja de Excel que se crea desde un inicio (también adjunto) que establece el color de los cuadrados de forma aleatoria, los cuales se basan en los nombres de los colores en inglés. Con esto, cada vez que en pantalla aparezca un cuadrado azul, se creará un marcador de tipo cero, mientras que si aparece un cuadrado de color diferente al azul se creará un marcador de tipo uno. Las enumeraciones de los marcadores y la cantidad que se desee puede cambiarse según el usuario, pero en este estudio se establece un marcador cero para ausencia de estímulo, y marcador uno para presencia de estímulo.



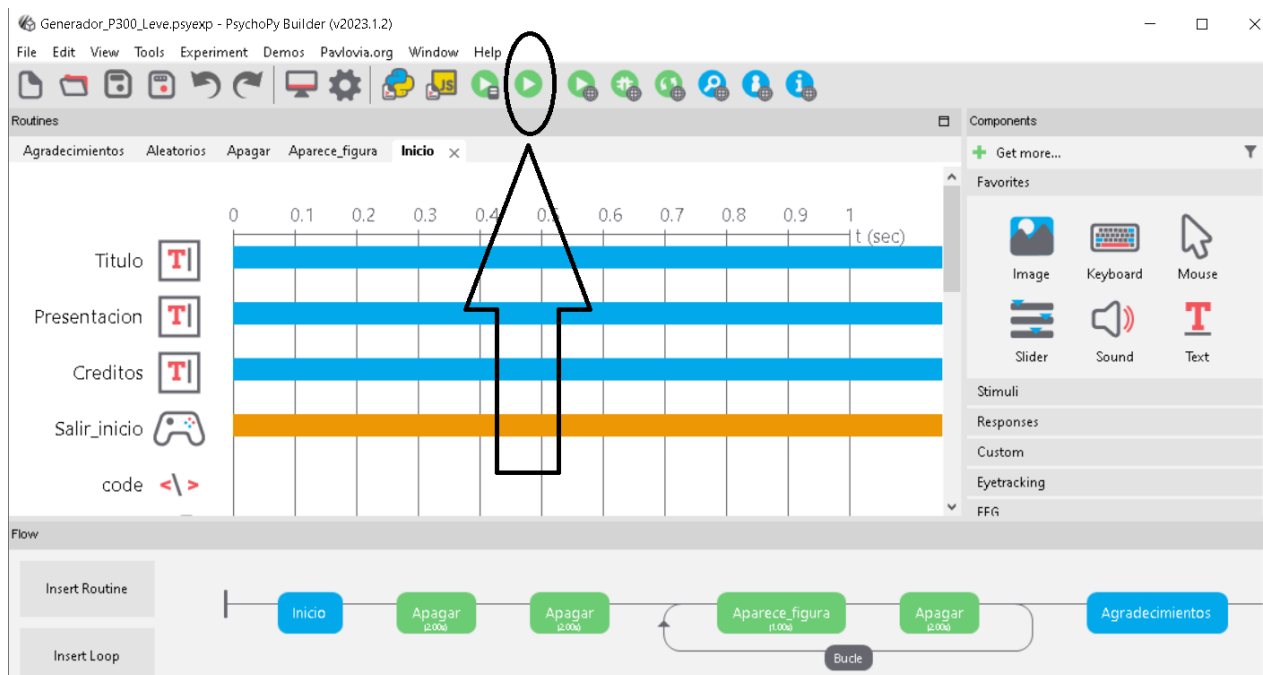
Este es el programa creado para el proyecto, con los segmentos de código al inicio (llamado Inicio) y en cada estímulo (llamado Aparece_figura).



Y la ubicación de la variable de la que se basa el código para crear los marcadores es el siguiente:

	A	B	C	D
1	Color	tano	Volumen	Cruz
2	green	A		1 green
3	blue	A		0 grey
4	white	B		1 white
5	blue	A		0 grey
6	red	C		1 red
7	blue	A		0 grey
8	yellow	D		1 yellow
9	blue	A		0 grey
10	cyan	E		1 cyan
11	blue	A		0 grey
12	aqua	F		1 aqua

Donde la primera columna es la variable que define marcadores 1 o 0 según el color del cuadrado. Se finaliza con el iniciar el programa de Psychopy en la parte central superior:



Donde se tendrá que esperar unos segundos, y luego se escribe el nombre del paciente y el número de la sesión y se presiona en “Ok”.

 Generador_P300_Leve

participant

241831

session

001

OK

Cancel

13. Grabar con LabRecorder:

Una vez inicia el programa de Psychopy, comienza la comunicación LSL, de la cual se debe anclar a la transmisión también LSL que está actualmente el casco con las señales EEG. El programa de LabRecorder permite unir ambas transmisiones en un solo archivo .xdf, por lo que se inicia por abrir el programa, donde se verá lo siguiente:

Lab Recorder

File Help

Recording Control

Start

Stop

☒ Enable RCS

RCS Port: 22345

Record from Streams

Select All

Select None

Update

Saving to...

C:\Users\ASUS-PC\Documents\CurrentStudy
sub-P001\ses-S001\eeg\sub-P001_ses-S001_task-Default_run-001_eeg.xdf

Study Root

C:\Users\ASUS-PC\Documents\CurrentStudy

Browse...

File Name/Template

-%p_ses-%s_task-%b_run-%r_%m.xdf

☒ BIDS

Block/Task (%b):

Default

Run (%r)

1

Participant (%p)

P001

Session (%s)

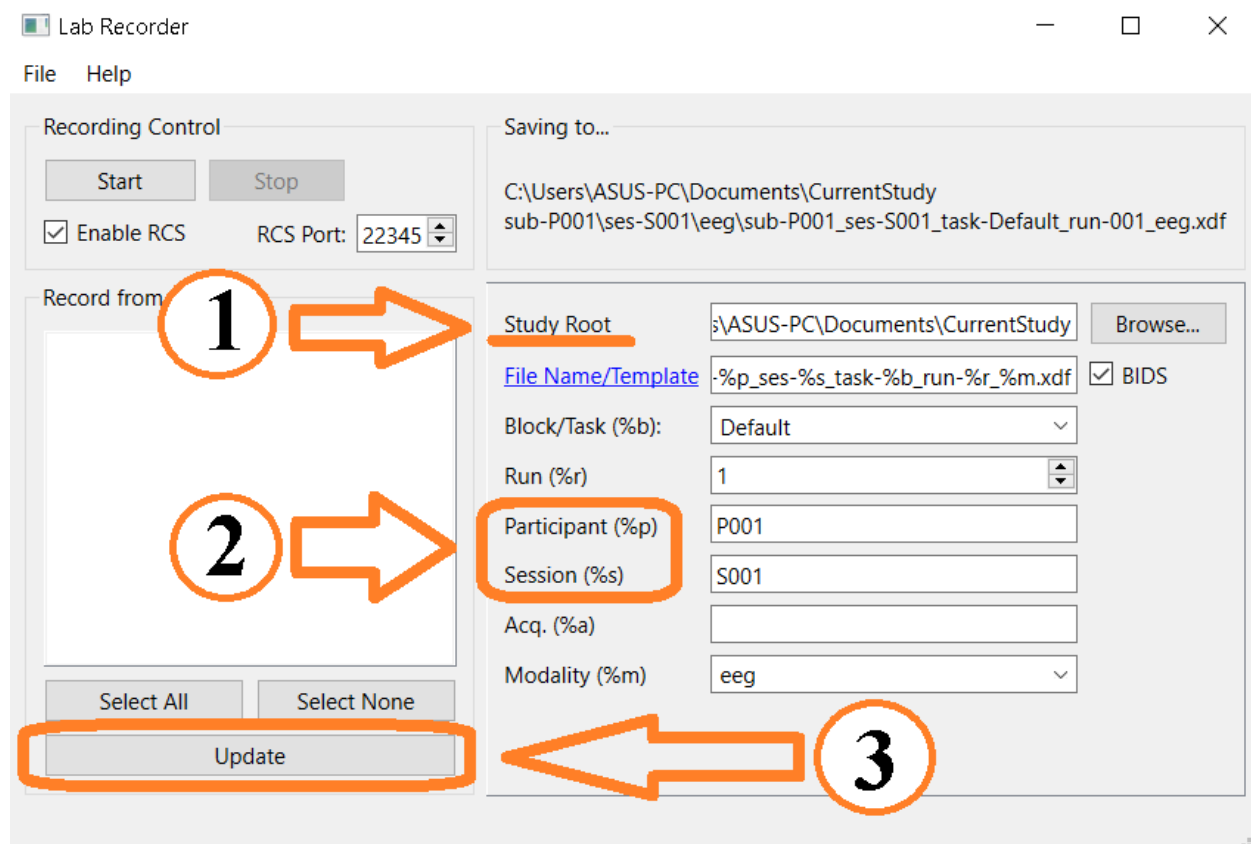
S001

Acq. (%a)

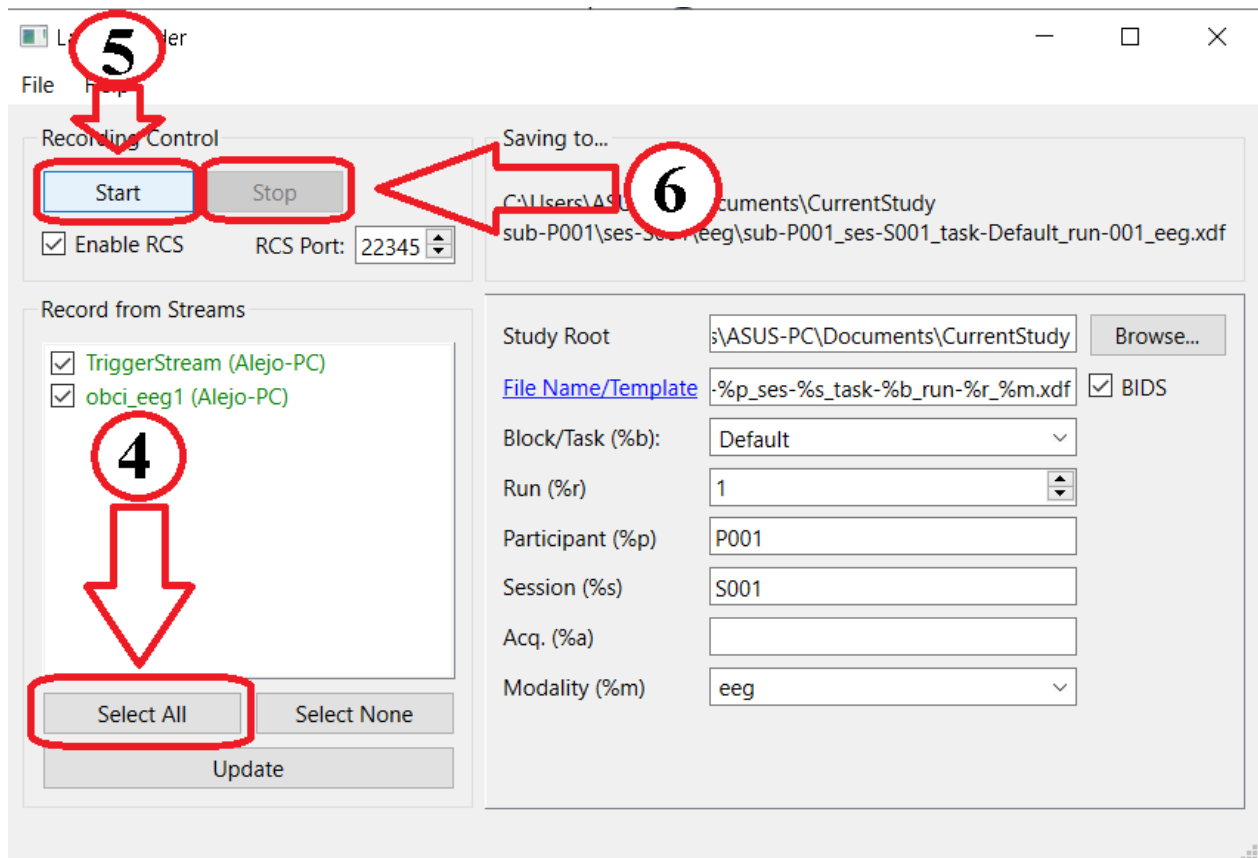
Modality (%m)

eeg

En “Study Root” (1) se puede modificar la dirección de almacenamiento, y el nombre del participante y el número de la sesión más abajo (2). Se conserva la modalidad eeg y luego se actualiza en el botón “Update” (3) para observar las comunicaciones LSL actuales:



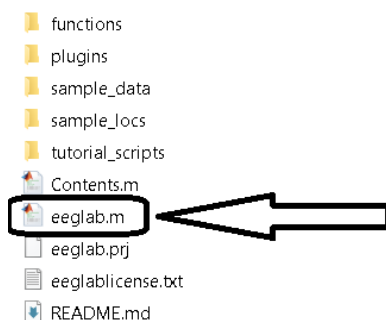
Con esto, se presionan las casillas de la izquierda de ambas transmisiones, para marcar las dos, de EEG y marcadores, o puede presionarse el botón “Select All” (4) solamente si se ha activado 1 solo puerto LSL en OpenBCI:



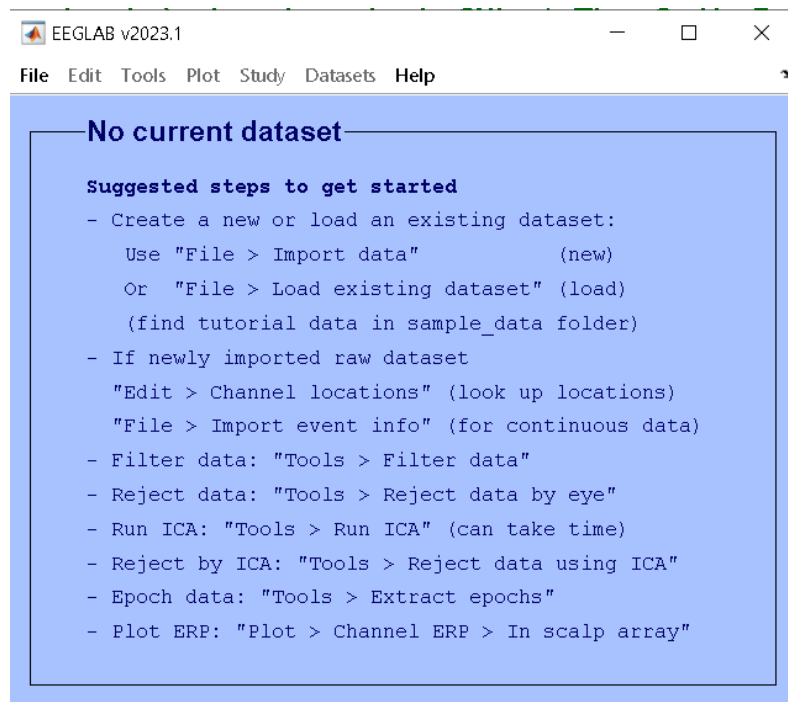
Y por último se presiona en la parte superior izquierda en “Start” (5) para iniciar a grabar las señales EEG junto con los marcadores que Psychopy crea. Luego se inicia el generador de estímulos de Psychopy, y en el momento de finalizar, se presiona en “Stop” (6) de al lado del botón de inicio de LabRecorder para finalizar la grabación de datos.

14. Importar archivo .xdf en EEGLab en Matlab:

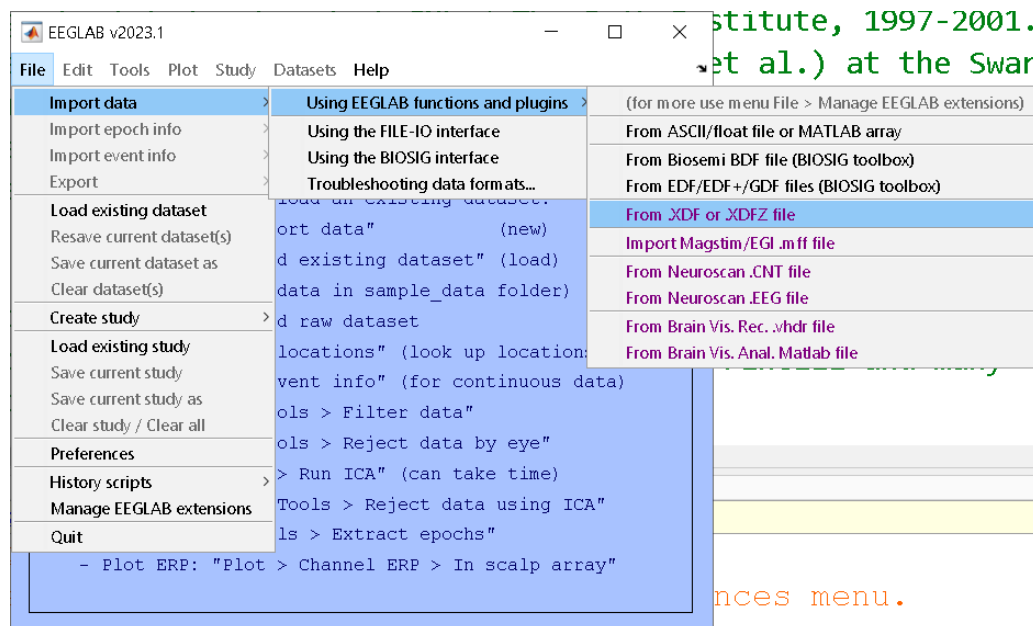
Ingresando a Matlab, se abrirá el programa llamado “eeglab.m” del cual se ejecutará en su respectiva carpeta.



Con eso se abrirá la ventana principal de EEGLAB:



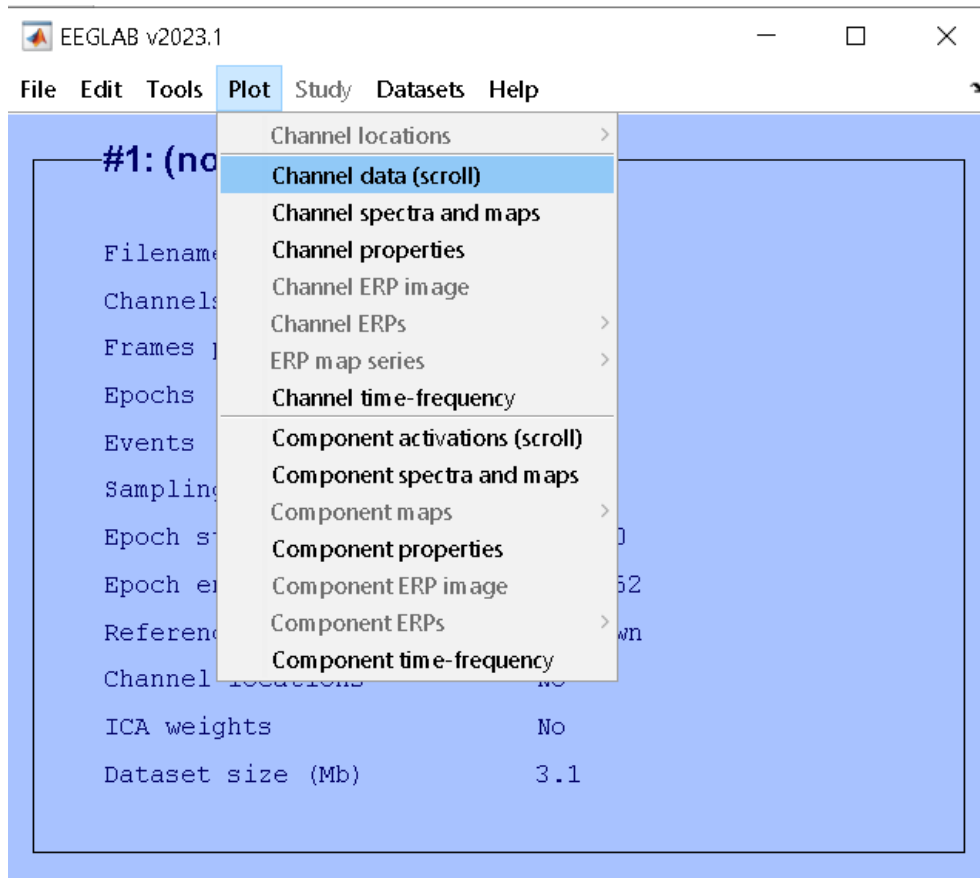
En el cual se importará el archivo creado, presionando en la parte superior izquierda en “File”, el primero que dice “Import data”, después en el primero que dice “Using EEGLAB functions and plugins” y por último en la primera opción de color violeta llamado “From .XDF or .XDFZ file”, el cual se ilustra a continuación:



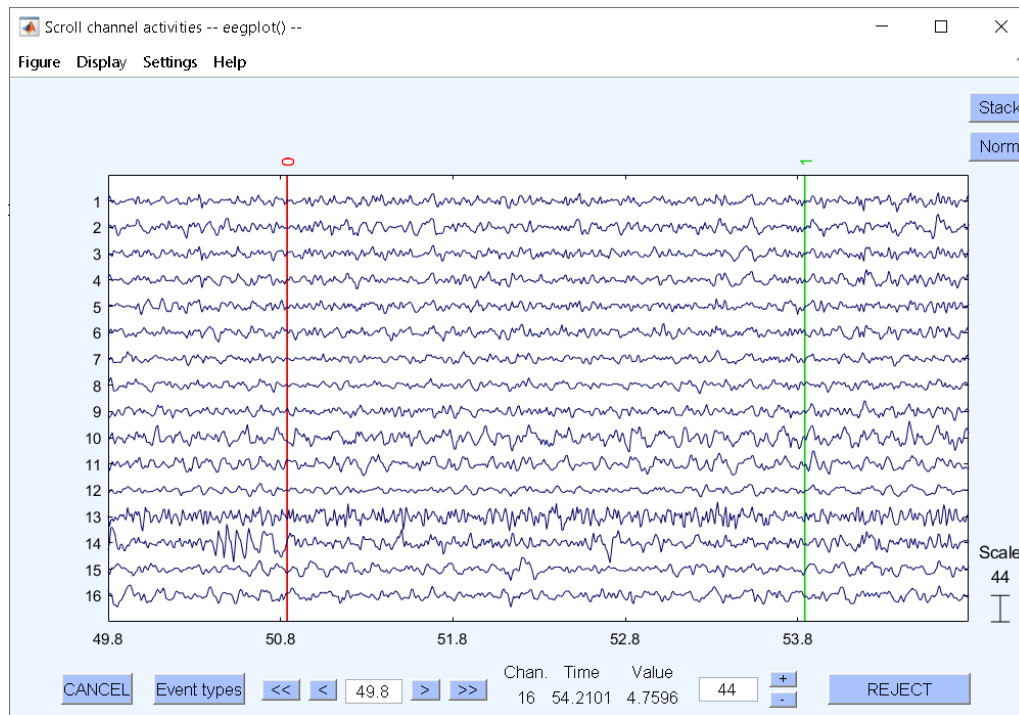
A partir de ahí se seleccionará el archivo creado con LabRecorder buscándolo en la ubicación que se seleccionó guardar en el paso anterior. Después se presiona “Ok” y es opcional ponerle un nombre al archivo, para luego presionar de nuevo “Ok”.

15. Visualizar señales EEG con marcadores:

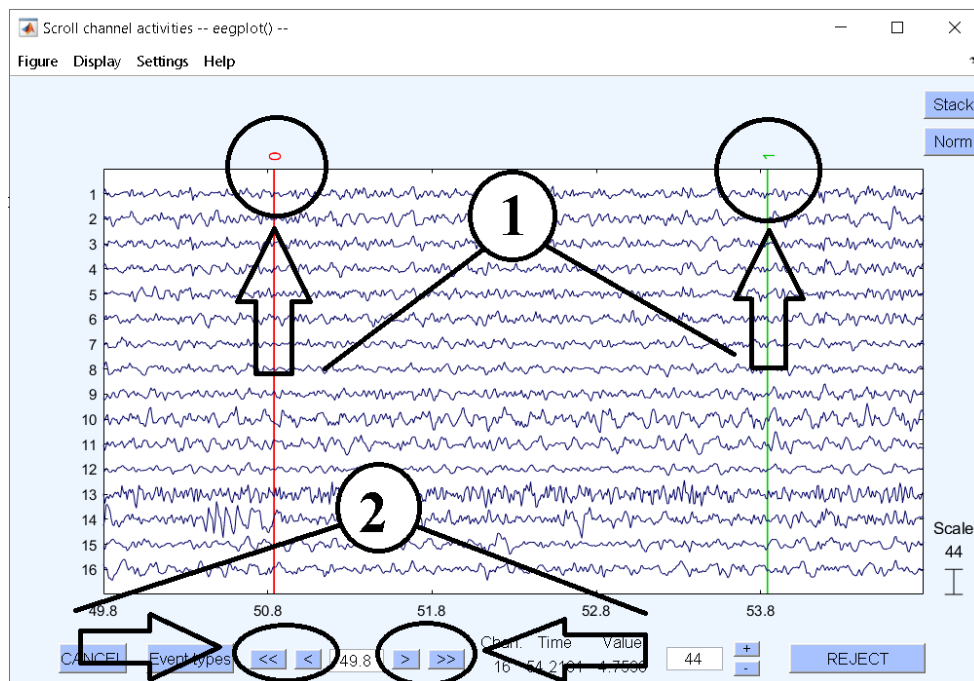
Ya habiéndose importado el archivo .xdf, en la parte izquierda superior de la ventana se selecciona en “Plot” y luego en “Channel data (scroll)”:



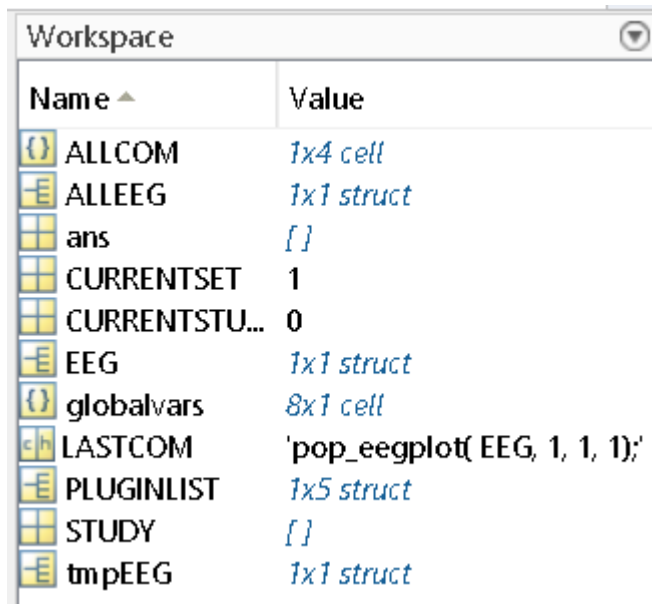
Donde nos arrojará esta ventana:



En la cual se podrá observar los marcadores creados con Psychopy (1), unidos con las señales EEG captadas con el casco y el programa OpenBCI. Las flechas de la parte inferior cercanos al centro permiten desplazar la visualización de las señales y marcadores (2).

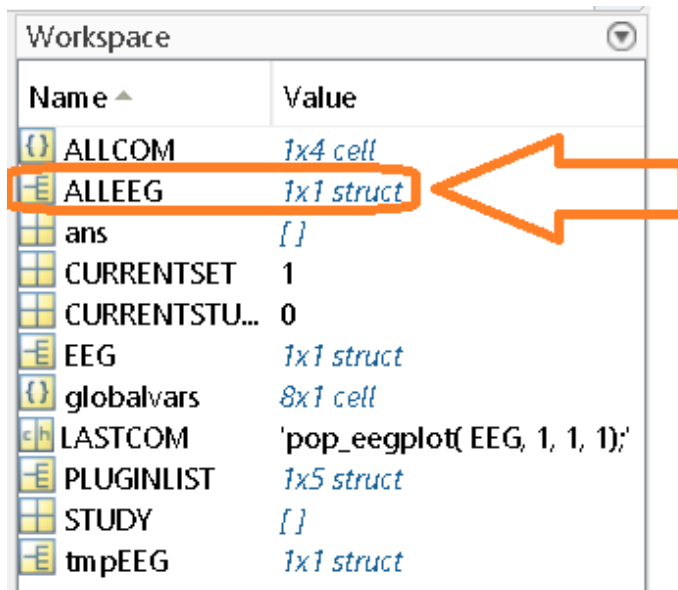


Con esto, se crean automáticamente las variables que pueden ser usadas para cualquier siguiente proyecto que involucre la extracción de las señales con sus respectivos marcadores en Matlab:



Name ^	Value
ALLCOM	1x4 cell
ALLEEG	1x1 struct
ans	[]
CURRENTSET	1
CURRENTSTU...	0
EEG	1x1 struct
globalvars	8x1 cell
LASTCOM	'pop_eegplot(EEG, 1, 1, 1);'
PLUGINLIST	1x5 struct
STUDY	[]
tmpEEG	1x1 struct

Donde en el segundo ítem, el que tiene de nombre “ALLEEG”:



Name ^	Value
ALLCOM	1x4 cell
ALLEEG	1x1 struct
ans	[]
CURRENTSET	1
CURRENTSTU...	0
EEG	1x1 struct
globalvars	8x1 cell
LASTCOM	'pop_eegplot(EEG, 1, 1, 1);'
PLUGINLIST	1x5 struct
STUDY	[]
tmpEEG	1x1 struct

Poseerá todas las señales electroencefalográficas, de todos los 16 electrodos, al igual que la información de los marcadores:

ALLEEG	
1x1 struct with 43 fields	
Field	Value
xmin	0
xmax	337.1520
times	1x42145 double
data	16x42145 single
icaact	[]
icawinv	[]
icasphere	[]
icaweights	[]
icachansind	[]
chanlocs	[]
urchanlocs	[]
chaninfo	1x1 struct
ref	'common'
event	1x100 struct
urevent	[]
eventdescription	1x3 cell
epoch	[]

La matriz “data” (A) contiene en cada fila cada lectura de electrodo, con el total de 16 electrodos y los datos en cantidad de muestras, del cual son 125 Hz de muestreo. Mientras la estructura “event” (B) contiene los tipos de marcadores generados, ya sean 1 o 0 en este proyecto, junto con el número de muestra total en el que apareció en toda la lectura de datos muestreados:

ALLEEG	
1x1 struct with 43 fields	
Field	Value
xmin	0
xmax	337.1520
times	1x42145 double
data	16x42145 single
icaact	[]
icawinv	[]
icasphere	[]
icaweights	[]
icachansind	[]
chanlocs	[]
urchanlocs	[]
chaninfo	1x1 struct
ref	'common'
event	1x100 struct
urevent	[]
eventdescription	1x3 cell
epoch	[]

Obteniendo como resultado final las señales recolectados de los 16 electrodos:

ALIEEG													
ALIEEG.data													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-8.6060	-1.2552	8.9317	11.7751	7.4623	-1.1018	1.2913	-7.3538	-11.7357	6.6603	2.3634	-3.4280	7.2129
2	-1.1119	-3.2638	4.5980	9.8776	-0.6710	-1.5435	4.3543	-0.0040	-3.2272	1.8965	1.8256	-5.2079	-4.3621
3	-5.7219	-2.4314	9.0637	9.3897	1.7531	-4.8468	-0.0873	-6.3201	-8.3629	9.8905	5.8910	0.3772	6.4548
4	-8.3597	-3.8111	6.1268	3.9531	-10.1629	-17.5818	-4.4186	-2.5152	3.1451	25.1034	23.5199	15.8923	15.2886
5	-2.7277	1.0146	2.0999	4.5316	4.4668	2.6713	2.0278	-4.5463	-8.0148	5.0022	9.2173	-0.5955	2.6495
6	-4.5883	-3.9843	4.6239	7.8999	5.0309	1.7418	0.0652	-6.2439	-6.3436	5.9676	6.0083	-2.4607	0.4554
7	-3.6915	-4.2882	-4.9088	1.7666	3.1187	9.0766	9.3035	5.3548	2.8475	-3.6127	-3.2376	-8.5716	-14.8183
8	0.2225	-0.3151	0.9993	7.3974	5.3125	8.2560	9.3293	7.5691	3.0991	-6.0958	-2.6105	-3.0165	-10.5121
9	-13.3447	-6.1228	-12.0522	-9.9869	-5.7050	-3.8659	13.6542	14.9724	20.5759	22.2991	1.0356	2.9336	2.3576
10	-2.8990	6.1599	-1.9400	-0.5615	5.2047	-0.4824	5.6590	2.7491	6.2586	5.0911	-10.2021	-4.2361	-11.8882
11	1.8187	-1.7524	0.2639	2.2882	-6.4187	-12.2918	-8.4156	-10.0359	-6.9646	-3.2733	-2.0367	5.1800	8.7448
12	-4.7945	-2.2188	-0.5928	4.0932	5.1796	3.5032	4.5725	0.8878	-3.6016	-1.4592	0.8658	-1.7628	-2.7964
13	-2.8738	7.6846	11.0575	-6.1546	-9.0638	4.9852	-5.0135	0.6149	-0.3559	-11.8214	9.6599	8.6535	-4.3909
14	-27.1430	-61.6361	-82.0963	-39.0584	6.4458	45.6534	86.4910	54.7711	-1.8200	-10.0100	-21.7125	-38.1151	-37.5730
15	-1.0117	-1.6378	-0.4940	0.3696	-4.8422	-1.2713	0.7858	-3.1117	-1.0856	-0.8960	3.3354	4.5428	-0.6503
16	-10.6820	-5.7221	0.2698	10.9827	16.6742	13.8974	11.6483	7.9532	1.8914	0.0278	-1.5450	-4.5862	-2.3271
17													

Y las ubicaciones de los marcadores con sus respectivas referencias:

ALIEEG			
ALIEEG.event			
Fields	type	latency	duration
1	'0'	3760	1
2	'1'	4102	1
3	'0'	4478	1
4	'0'	4853	1
5	'0'	5229	1
6	'0'	5604	1
7	'0'	5980	1
8	'0'	6356	1
9	'1'	6732	1
10	'0'	7107	1
11	'1'	7483	1
12	'1'	7859	1
13	'1'	8235	1
14	'1'	8610	1
15	'0'	8986	1
16	'0'	9361	1
17	'0'	9737	1